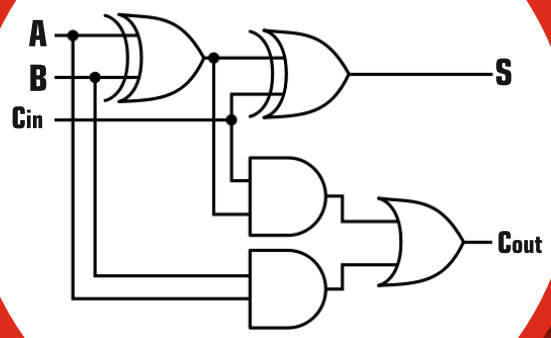
ACE ENGINEERING PUBLICATIONS
Since 2011
ACE

ESE | GATE | PSUs



ELECTRICAL ENGINEERING

DIGITAL ELECTRONICS & MICROPROCESSORS

Text Book : Theory with worked out Examples and Practice Questions

Chapter 6 Digital & Microprocessors

Solutions for Text Book Practice Questions

1. Number Systems

Solutions for Objective Practice Questions

01. Ans: (d)

Sol: $135_x + 144_x = 323_x$

$(1\times x^2 + 3 \times x^1 + 5 \times x^0)+(1\times x^2+4\times x^1+ 4\times x^0)$

$= 3x^2 + 2x^1 + 3x^0$

$\Rightarrow x^2+3x+5+x^2+4x+4 = 3x^2+ 2x + 3$

$x^2 - 5x - 6 = 0$

$(x-6)(x+1) = 0$ (Base cannot be negative)

Hence $x = 6$.

(OR)

As per the given number x must be greater than 5. Let consider $x = 6$

$(135)_6 = (59)_{10}$

$(144)_6 = (64)_{10}$

$(323)_6 = (123)_{10}$

$(59)_{10} + (64)_{10} = (123)_{10}$

So that $x = 6$

02. Ans: (a)

Sol: 8-bit representation of

$+127_{10} = 01111111_{(2)}$

1's complement representation of

$-127 = 10000000$.

2's complement representation of

$-127 = 10000001$.

No. of 1's in 2's complement of

$-127 = m = 2$

No. of 1's in 1's complement of

$-127 = n = 1$

$\therefore$ m: n = 2:1

03. Ans: (c)

Sol: In 2's complement representation the sign bit can be extended towards left any number of times without changing the value. In given number the sign bit is 'X_3', hence it can be extended left any number of times.

04. Ans: (c)

Sol: Binary representation of $+(539)_{10}$:

2	539	
2	269	–1
2	134	–1
2	67	–0
2	33	–1
2	16	–1
2	8	–0
2	4	–0
2	2	–0
	1	–0

$(+539)_{10} = (10000\ 11\ 0\ 11)_2 = (00100\ 0011011)_2$

2's complement $\rightarrow$ 110111100101

Hexadecimal equivalent $\rightarrow (DE5)_H$

05. Ans: 5

Sol: Symbols used in this equation are 0,1,2,3 Hence base or radix can be 4 or higher

$(312)_x = (20)_x\ (13.1)_x$

$3x^2 + 1x^1 + 2x^0 = (2x^1+0)\ (x+3x^0+x^{-1})$

$3x^2 + x + 2 = (2x)\left(x + 3 + \frac{1}{x}\right)$

$3x^2 + x + 2 = 2x^2 + 6x + 2$

$x^2 - 5x = 0$

$x(x-5) = 0$

$x = 0$ (or) $x = 5$

x must be $x > 3$, So $x = 5$

06. Ans: 3

Sol: $123_5 = x8_y$

$1 \times 5^2 + 2 \times 5^1 + 3 \times 5^0 = x.y^1 + 8 \times y^0$

$25 + 10 + 3 = xy + 8$

$\therefore xy = 30$

Possible solutions:

i. $x = 1, y = 30$

ii. $x = 2, y = 15$

iii. $x = 3, y = 10$

$\therefore$ 3 possible solutions exists.

07. Ans: 1

Sol: The range (or) distinct values

For 2's complement $\Rightarrow -(2^{n-1})$ to $+(2^{n-1}-1)$

For sign magnitude

$\Rightarrow -(2^{n-1}-1)$ to $+(2^{n-1}-1)$

Let $n = 2 \Rightarrow$ in 2's complement

$-(2^{2-1})$ to $+(2^{2-1}-1)$

-2 to $+1 \Rightarrow -2, -1, 0, +1 \Rightarrow X = 4$

$n = 2$ in sign magnitude $\Rightarrow -1$ to $+1 \Rightarrow Y = 3$

$X - Y = 1$

08. Ans: (c)

Sol: (a) $(68)_{16} = (001\ 101\ 000)_2$

$= (1\ \ 5\ \ 0)_8$

(b) $(8C)_{16} = (010\ \ 001\ 100)_2$

$= (2\ \ 1\ \ 4)_8$

(c) $(4F)_{16} = (001\ 001\ 111)_2$

$= (1\ \ 1\ \ 7)_8$

(d) $(5D)_{16} = (001\ 011\ 101)_2$

$= (1\ \ 3\ \ 5)_8$

09. Ans: (b)

Sol: A.7 5 B.6 5

(111 101) (110 101)

C. 3 7 D. 2 6

(011 111) (010 110)

10. Ans: (a)

Sol: 2's complement arithmetic is preferred in digital computers because it is efficient and one representation for zero.

11. Ans: (a)

Sol: $(11X1Y)_8 = (12C9)_{16}$

$8^4 + 8^3 + 8^2X + 8 + Y$

$= 16^3 + (2 \times 16^2) + (12 \times 16) + 9$

$4096 + 512 + 64X + 8 + Y$

$= 4096 + 512 + 192 + 9$

$\therefore 4616 + 64X + Y = 4809$

$64X + Y = 193$

By verification option (a) is correct.

12. Ans: (d)

Sol: 2's comp no:

a_3	a_2	a_1	a_0

2's comp no. using 6 bits

→

a_3	a_3	a_3	a_2	a_1	a_0

(2's comp no)× 2 + 1

→

a_3	a_3	a_2	a_1	a_0	1

Solutions for Conventional Practice Questions

01.

Sol:

1) 110.01 + 1.011

$= 110.010$

$\underline{\quad 1.011}$

111.101

$= 111.101$

2) $(11101.01)_2$

$= (1\times2^4 + 1\times2^3 + 1\times2^2 + 0\times2^1 + 1\times2^0)$

$+ (0 \times 2^{-1}) + (1 \times 2^{-2})$

$= 16 + 8 + 4 + 1 + 0.25$

$= (29.25)_{10}$

3) 11100.101 – 101.01

11100.101

11010.110 (2's complement of 101.01 is)

10111.011

4) Convert $(111000)_2$ to octal

$= \underline{111}\ \underline{000}$

$= (70)_8$

02.

Sol: First convert Hexadecimal to binary and then binary to octal number

$(A5F1)_{16} = (1010\ 0101\ 1111\ 0001)_2$

$= (001\ 010\ \ 010\ \ 111\ 110\ \ 001)_2$

1 2 2 7 6 1

$= (1\ 2\ 2\ 7\ 6\ 1)_8$

03.

Sol:

i) $(1A53)_{16} = (1\times16^3) + (10\times16^2)$

$+ (5\times16) + (3\times16°)$

$= 4096 + 2560 + 80 + 3$

$= (6739)_{10}$

ii) $(93)_{16} = (147)_{10,}\ (DE)_{16} = (222)_{10}$

$(93)_{16} + (DE)_{16} = 147 + 222 = (369)_{10}$

$= (171)_{16}$

iii) $(11010)_2$

$= 1\times2^4 + 1\times2^3 + 0\times2^2 + 1\times2 + 0\times2^0$

$= 16 + 8 + 0 + 2 + 0 = 26$

2. Logic Gates and Boolean Algebra

Solutions for Objective Practice Questions

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. ∴ Overflow is indicated by $= \bar{x}\,\bar{y}\,z + x\,y\,\bar{z}$

Examples

1. A = +7 0111
   B = +7 0111
   14 1110 $\Rightarrow \bar{x}\,\bar{y}\,z$
2. A = +7 0111
   B = +5 0101
   12 1100 $\Rightarrow \bar{x}\,\bar{y}\,z$
3. A = –7 1001
   B = –7 1001
   –14 10010 $\Rightarrow x\,y\,\bar{z}$
4. A = –7 1001
   B = –5 1011
   –12 10100 $\Rightarrow x\,y\,\bar{z}$

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	X	o/p	
1	0	1	$= \overline{X}$
1	1	0	$= \overline{X}$

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

Stage 2:

$\overline{X}$	X	o/p
0	1	1
1	0	1

For second XOR gate o/p = 1.

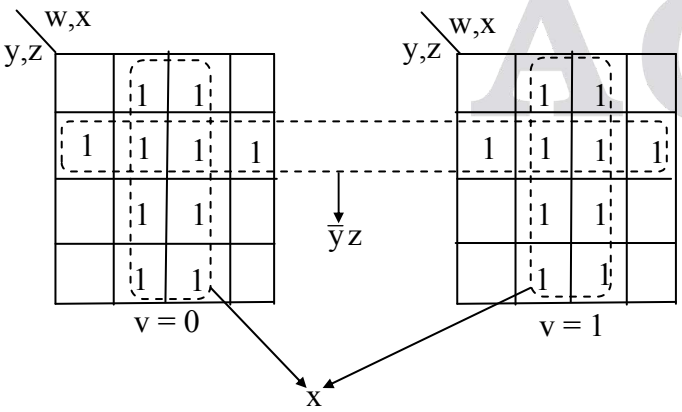
Similarly for third XOR gate o/p = $\overline{X}$ & for fourth o/p = 1

For Even number of XOR gates o/p = 1

For 20 XOR gates cascaded o/p = 1.

03. Ans: (b)

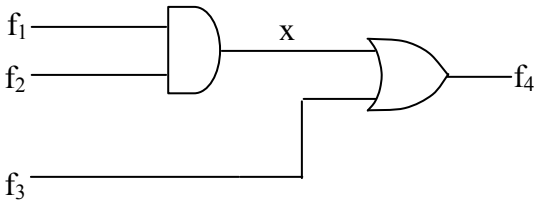
Sol:



Number of min terms = 20

04. Ans: (c)

Sol:



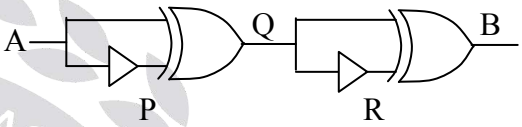
$x = f_1 f_2$

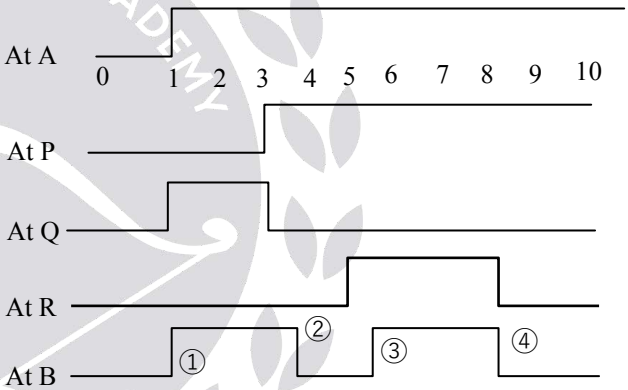
$f_4 = f_1.f_2 + f_3$

$f_2 = \sum m(6, 8)$

05. Ans: (d)

Sol:





06. Ans: (c)

Sol: $\overline{x_1} \oplus \overline{x_3} = \overline{x_1}\, x_3 + x_1 \overline{x_3} = y$

$\overline{x_2} \oplus \overline{x_4} = \overline{x_2}\, x_4 + x_2 \overline{x_4} = z$

$\left(\overline{x_1} \oplus \overline{x_3}\right) \oplus \left(\overline{x_2} + \overline{x_4}\right)$

$= y \oplus z = 0$, when y = z

$\therefore$ option (c) is true

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

07. Ans: (b)

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

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

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

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

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

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

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

$+ (\bar{b}\bar{c} + \bar{a}\bar{b} + \bar{a}\,\bar{c})(c) + abc$

$= a\bar{b}\bar{c} + \bar{a}b\bar{c} + abc + \bar{a}\bar{b}c$

$= \bar{c}[a\bar{b} + \bar{a}b] + c[ab + \bar{a}\bar{b}]$

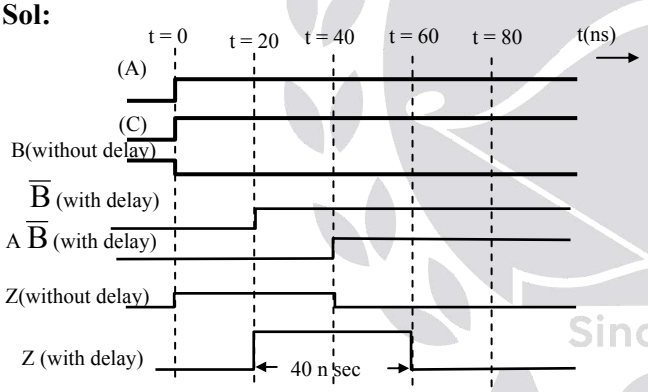
$= \sum m(1,2,4,7)$

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

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

08. Ans: 40

Sol:



$\therefore$ Z is 1 for 40 nsec

09. Ans: (c)

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

Where $Y_1 = \bar{Y}$

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

10. Ans: (d)

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

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

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

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

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.

12. Ans: (b)

Sol: (A) $A \oplus B = 0$ only when A = B

(B) $\overline{A + B} = \bar{A}.\bar{B} = 0$ only when A = 1 and B = 1

(C) $\bar{A}.B = 0$ only when A = 1 and B = 0

(D) $A \oplus B = 1$ only when $A \neq B$

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

Inverse function

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

$= abc + \bar{a}\,bc + ab\,\bar{c}$

14. Ans: (c)

Sol: AND gate : Boolean multiplication
OR gate : Boolean addition
NOT gate : Boolean complementation

15. Ans: (a)

Sol: When all inputs of a NAND-gate are shorted to get a one input, one output gate, it becomes an inverter.
When all inputs of a NAND-gate are at logic '0' level, the output is at logic '1' level.
Both statements are true and statement-II is the correct explanation of statement-I.

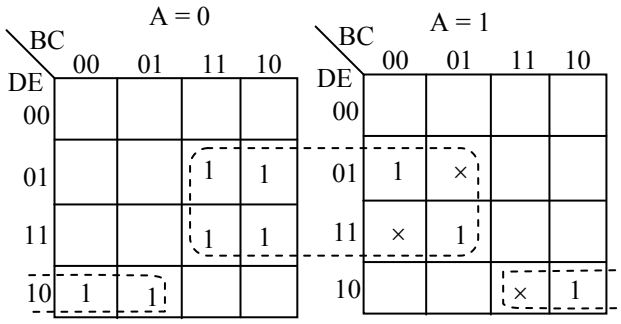
16. Ans: (c)

Sol: A NAND gate represents a universal logic family.
Only two NAND gates are sufficient to accomplish any of the basic gates.
Statement-I is true but statement-II is false.

Solutions for Conventional Practice Questions

01.

Sol: The given don't care & expression can be realized as follows:



The required don't care combination is

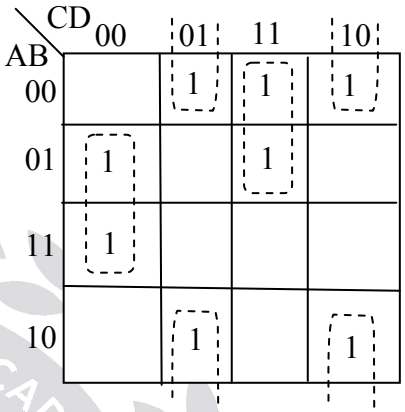
$= ABC\,\bar{D}\,E + AB\,\bar{C}\,DE + A\,\bar{B}\,CD\,\bar{E}$

02.

Sol:

Using K - map

$f(A, B, C, D) = \Sigma m(1, 2, 3, 4, 7, 9, 10, 12)$



$f(A, B, C, D) = B\bar{C}\,\bar{D} + \bar{A}CD + \bar{B}\bar{C}D + \bar{B}C\bar{D}$

Using the K-map, the minimal form of the given minterms is found.

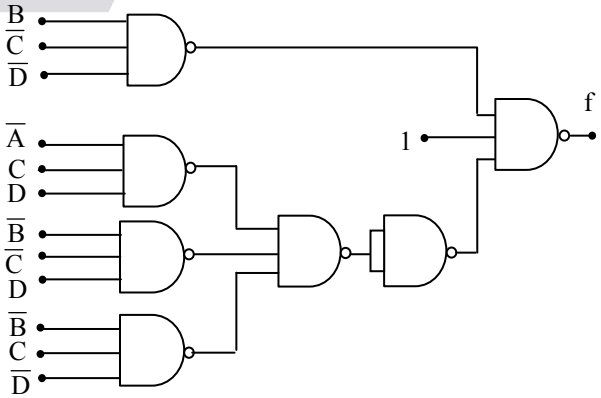
This function can be implemented using seven number of 3 input NAND gates.

$f = B\bar{C}\,\bar{D} + \bar{A}CD + \bar{B}\bar{C}D + \bar{B}C\bar{D}$

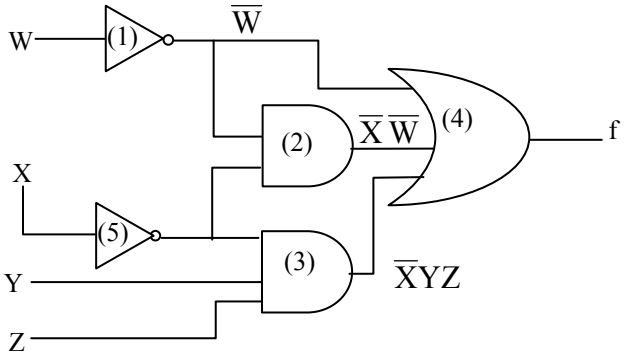
$$f = \overline{B\bar{C}\,\bar{D} + \bar{A}CD + \bar{B}\bar{C}D + \bar{B}C\bar{D}}$$

(∴ using De-Morgan's theorem)

The circuitry for above minimal expression using NAND gates as follows. Let us assume that variables are available in complement form also.



03.
Sol:



$$f = \overline{W} + \overline{X}\,\overline{W} + \overline{X}YZ$$

$$= \overline{W}[1 + \overline{X}] + \overline{X}YZ$$

$$f = \overline{W} + \overline{X}YZ$$

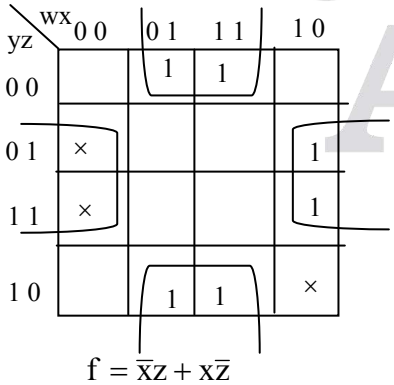
Thus from expression for the output f, WE can conclude that gate no.(2) is redundant and even if the gate is removed from the circuit the output expression is

$$f = \overline{W} + \overline{X}YZ$$

3. K - Maps

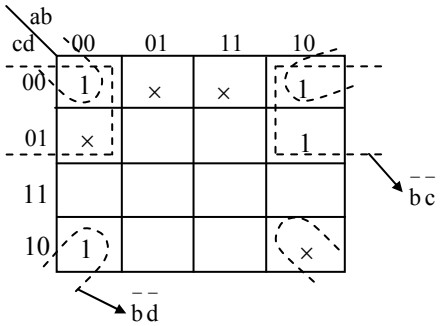
Solutions for Objective Practice Questions

01. Ans: (b)
Sol:



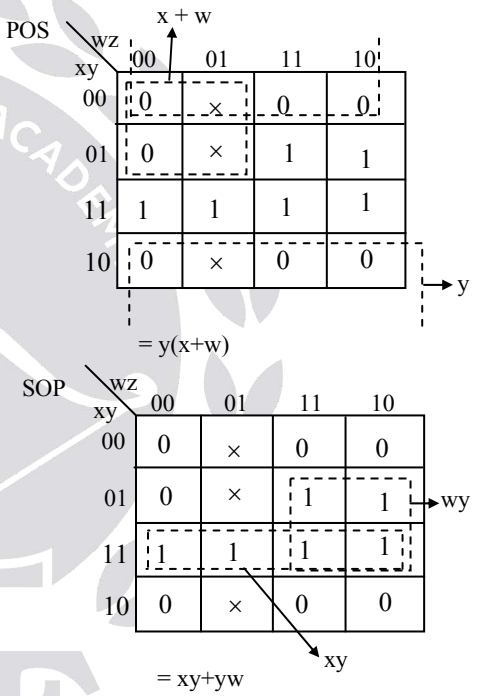
$$f = \overline{x}z + x\overline{z}$$

02. Ans: (b)
Sol:



$$f = \overline{b}\,\overline{d} + \overline{b}\,\overline{c}$$

03.
Sol:



SOP: x y + y w

POS: y(x + w)

04. Ans: (a)
Sol: For n-variable Boolean expression,

Maximum number of minterms = 2^n

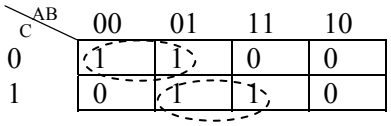
Maximum number of implicants = 2^n

Maximum number of prime implicants = $\frac{2^n}{2}$

$= 2^{n-1}$

05. Ans: (c)

Sol:



$F(A, B, C) = \overline{A}\,\overline{C} + BC$

06. Ans: 1

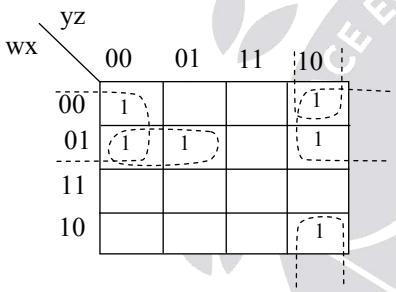
Sol: After minimization = $\left(\overline{\overline{A} + \overline{B} + \overline{C} + \overline{D}}\right)$

$= ABCD$

$\therefore$ only one minterm.

07. Ans: 3

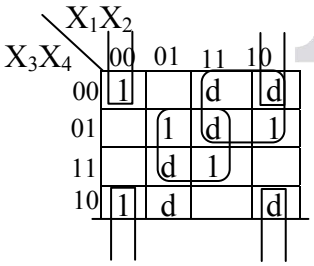
Sol: $\overline{w}\,\overline{z} + \overline{w}\,x\overline{y} + \overline{x}\,y\overline{z}$



$\therefore$ Total number of prime implicants of the function 'f' is 3.

08. Ans: (c)

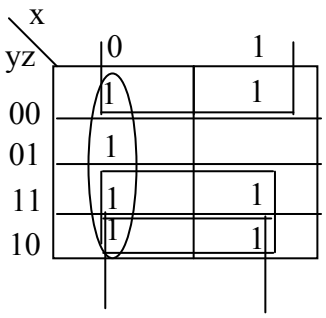
Sol: Given K-map is



Output = $\overline{X_2}\ \overline{X_4} + X_1\overline{X_3} + X_2X_4$

09. Ans: (a)

Sol:



The minimal form is

$F = \overline{x} + y + \overline{z}$

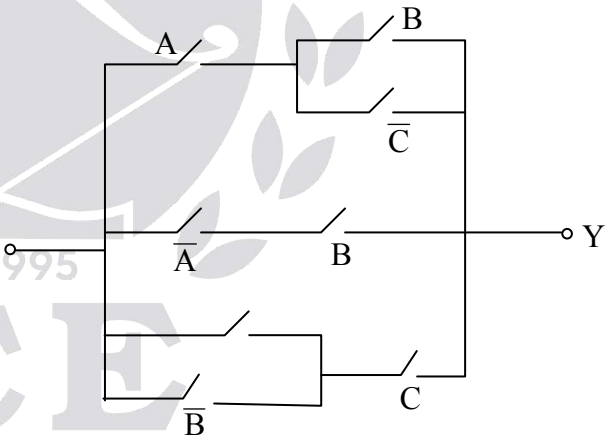
10. Ans: (a)

Sol: Given K-map

Solutions for Conventional Practice Questions

01.

Sol: Given circuit diagram is



Series combination: AND gate

Parallel combination: OR gate

$Y = A(B + \overline{C}) + \overline{A}B + (A + \overline{B})C$

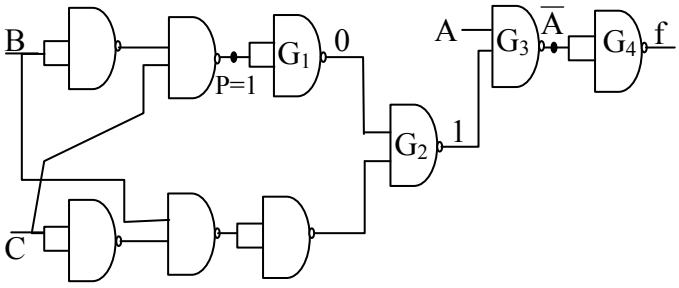
$= AB + A\overline{C} + \overline{A}B + AC + \overline{B}C$

$= B(A + \overline{A}) + A\overline{C} + AC + \overline{B}C$

$= B + A + \overline{B}C = A + B + C$

02.

Sol: The circuit diagram gives in the question is redrawn as



Output of gate G_1 will definitely be 0. If any one of the input of G_2 is 0, output of G_2 is definitely 1.

Output of gate $G_3 = \overline{A}$

Output of gate G_4, $f = \overline{\overline{A}} = A$

$\therefore f = A$

03.

Sol: Given K-map is

xy \ zw	00	01	11	10
00	d	1	0	1
01	0	1	d	0
11	1	d	d	0
10	d	0	0	d

The minimized SOP expression from the given k map is

$Y = \left[\overline{y}\,\overline{w} + \overline{x}\,\overline{z}w + xy\overline{z}\right]$ ------ (1)

for the expression in equation (1) the Literal count = 8

xy \ zw	00	01	11	10
00	d	1	0	1
01	0	1	d	0
11	1	d	d	0
10	d	0	0	d

The minimized POS expression is

$y = (\overline{x} + y)(\overline{z} + \overline{w})(\overline{y} + \overline{z})(x + z + w)$ ----- (2)

For the expression in equation (2) the Literal count = 9

4. Combinational Circuits

Solutions for Objective Practice Questions

01. Ans: (d)

Sol: Let the output of first MUX is "F_1"

$F_1 = AI_0 + AI_1$

Where A is selection line, I_0, I_1 = MUX Inputs

$F_1 = \overline{S}_1.W + S_1.\overline{W} = S_1 \oplus W$

Output of second MUX is

$F = \overline{A}.I_0 + A.I_1$

$F = \overline{S}_2.F_1 + S_2.\overline{F}_1$

$F = S_2 \oplus F_1$

But $F_1 = S_1 \oplus W$

$F = S_2 \oplus S_1 \oplus W$

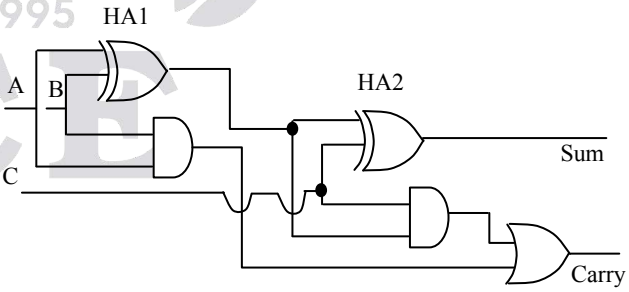
i.e., $F = W \oplus S_1 \oplus S_2$

02. Ans: 19.2

Sol: One AND/OR gate delay = 1.2 μs

One XOR gate delay = 2.4 μs

Full Adder with 2 Half Adder



In one F.A; Sum delay = 4.8 μs

Carry delay = 2.4 + 1.2 + 1.2μs = 4.8 μs

$\therefore$ RippleCarry waiting time

$= 4.8 \times 3 = 14.4$ μs

Final Result time = 14.4 + 4.8 = 19.2 μsec

03. Ans: (a)

Sol: The given circuit is binary parallel adder/subtractor circuit. It performs A+B, A–B but not A + 1 operations.

K	C_0	Operation
0	0	A+B (addition)
0	1	A+B+1(addition with carry)
1	0	A+$\overline{B}$ (1's complement addition)
1	1	A+$\overline{B}$+1(2's complement subtraction)

04. Ans: (d)

Sol: It is expansion of 2:4 decoders to 1:8 demultiplexer A_1, A_0 must be connected to S_1, S_0 i.e.., $R = S_0$, $S = S_1$

Q must be connected to S_2 i.e., $Q = S_2$

P is serial input must be connected to D_{in}

05. Ans: 6

Sol: T = 0 → NOR → MUX 1 → MUX 2

2ns 1.5ns 1.5ns

Delay = 2ns + 1.5ns + 1.5ns = 5ns

T = 1 → NOT → MUX 1→NOR→ MUX 2

1ns 1.5ns 2ns 1.5ns

Delay = 1ns + 1.5ns + 2ns + 1.5ns = 6ns

Hence, the maximum delay of the circuit is 6ns.

06. Ans: –1

Sol: When all bits in 'B' register is '1', then only it gives highest delay.

∴ '–1' in 8 bit notation of 2's complement is 1111 1111.

07. Ans: (d)

Sol: The race hazard problem does not occur in combinational circuits.

The output of a combinational circuit depends upon present inputs only.

Statement-I is false but Statement-II is true.

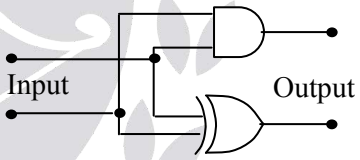
08. Ans: (b)

Sol: A de-multiplexer can be used as a decoder. A decoder with enable input acts as a de-multiplexer, while using Enable input as a data input line. De-multiplexer is realized using AND gates.

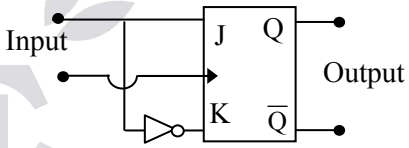
A	B	F
0	0	C
0	1	C
1	0	$\overline{C}$
1	1	$\overline{C}$

09. Ans: (b)

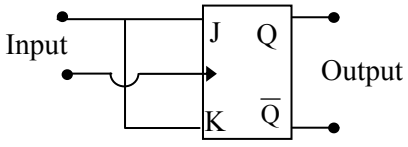
Sol: **Half Adder**



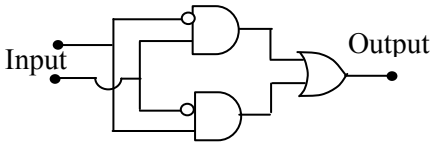
D-Flipflop



T-Flipflop

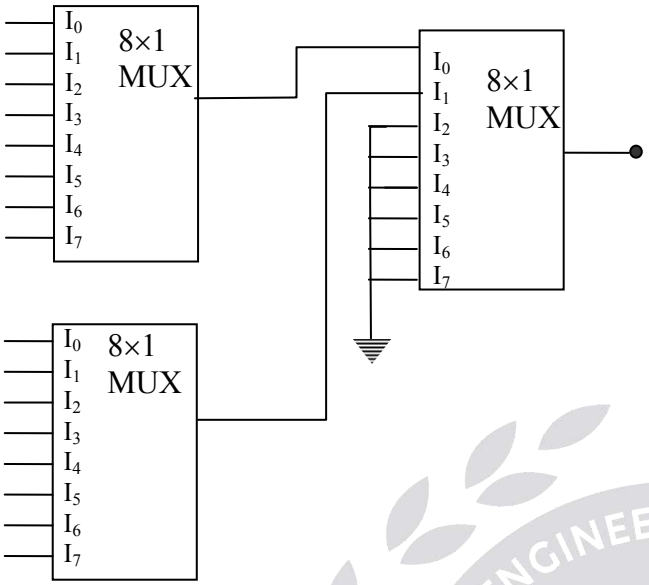


Exclusive - OR



10. Ans: (b)

Sol: →A 64 input MUX using 8-input MUX



→ A 6-variable function can be implemented using 6-input MUX.

11. Ans: 195

Sol: In a 16 bit parallel binary adder, the carry has to propagate 15 stages plus the maximum of time taken for producing sum and carry worst case Delay, T = 15×12 + 15

T = 180 + 15

T = 195ns.

12. Ans: (b)

Sol: Any Boolean function can be realized by using a suitable multiplexer.

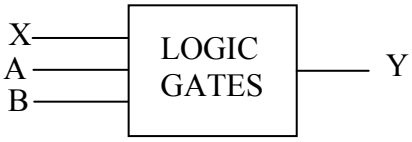
A multiplexer can be realized using NAND and NOR gates, which are universal gates.

Both statements are correct but statement-II is not a correct explanation for statement-I.

Solutions for Conventional Practice Questions

01.

Sol:



Truth table:

A	B	X	Y
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0

K map for the above truth table is

A \ BX	00	01	11	10
0	1	0	1	0
1	1	1	0	0

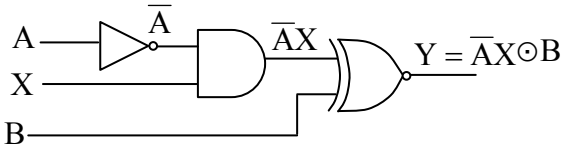
$$y = A\overline{B} + \overline{B}\overline{X} + \overline{A}BX$$

$$= \overline{B}[A + \overline{X}] + \overline{A}BX$$

$$y = \overline{B}[\overline{\overline{A}X}] + \overline{A}XB$$

$$y = [\overline{A}X \odot B]$$

If different logic gates are used then minimum number of gates required is 3



02.
Sol:
(a) Ex - 3 to 2421 code converter

Dec no.	Ex-3 Code	2 4 2 1 code
	$E_3\ E_2\ E_1\ E_0$	$Y_3\ Y_2\ Y_1\ Y_0$
0	0 0 1 1	0 0 0 0
1	0 1 0 0	0 0 0 1
2	0 1 0 1	0 0 1 0
3	0 1 1 0	0 0 1 1
4	0 1 1 1	0 1 0 0
5	1 0 0 0	1 0 1 1
6	1 0 0 1	1 1 0 0
7	1 0 1 0	1 1 0 1
8	1 0 1 1	1 1 1 0
9	1 1 0 0	1 1 1 1

K map for Y_3

E_3E_2 \ E_1E_0	00	01	11	10
00	d	d	0	d
01	0	0	0	0
11	1	d	d	d
10	1	1	1	1

$Y_3 = E_3$

K map for Y_2

E_3E_2 \ E_1E_0	00	01	11	10
00	d	d	0	d
01	0	0	1	0
11	1	d	d	d
10	0	1	1	1

$$Y_2 = E_3E_2 + E_2E_1E_0 + E_3E_0 + E_3E_1$$

K-Map for Y_1

E_3E_2 \ E_1E_0	00	01	11	10
00	d	d	0	d
01	0	1	0	1
11	1	d	d	d
10	1	0	1	0

$$Y_1 = E_3E_1E_0 + \overline{E}_3\overline{E}_1E_0 + E_3E_1E_0 + \overline{E}_3E_1\overline{E}_0 + E_3E_2$$

$$Y_1 = E_3E_2 + E_3 \oplus E_1 \oplus E_0$$

K-Map for Y_0

E_3E_2 \ E_1E_0	00	01	11	10
00	d	d	0	d
01	1	0	0	1
11	1	d	d	d
10	1	0	0	1

$$Y_0 = \overline{E}_0$$

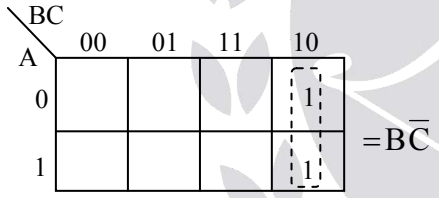
02.
Sol:
(b) The excess -3 code table

Dec	BCD Code	Ex–3 Code
0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 1 0 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 0 0
6	0 1 1 0	1 0 0 1
7	0 1 1 1	1 0 1 0
8	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 0 0

A self complementary code is a code in which the code of a number and code of 9's complement of that number are complementary to each other so from above table eg: if number is 3 its Ex-3 code is 0110. 9's complement of 3 in Ex-3 code is 1001 which is complementary to 0110. Thus Ex-3 code is a self complementary code.

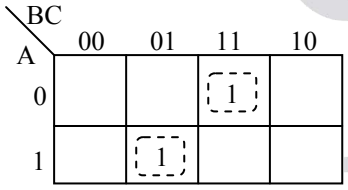
03.
Sol:

Input in Decimal	Input			Output						Output in Decimal
	A	B	C	F_0	F_1	F_2	F_3	F_4	F_5	
0	0	0	0	0	0	0	0	0	0	$0^2 = 0$
1	0	0	1	0	0	0	0	0	1	$1^2 = 1$
2	0	1	0	0	0	0	1	0	0	$2^2 = 4$
3	0	1	1	0	0	1	0	0	1	$3^2 = 9$
4	1	0	0	0	1	0	0	0	0	$4^2 = 16$
5	1	0	1	0	1	1	0	0	1	$5^2 = 25$
6	1	1	0	1	0	0	1	0	0	$6^2 = 36$
7	1	1	1	1	1	0	0	0	1	$7^2 = 49$



$= B\overline{C}$

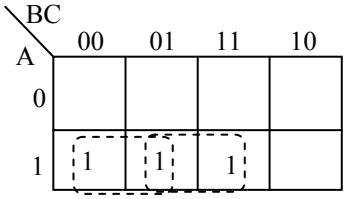
$F_3 = \sum m\ (2, 6)$



$= \overline{A}BC + A\overline{B}C$

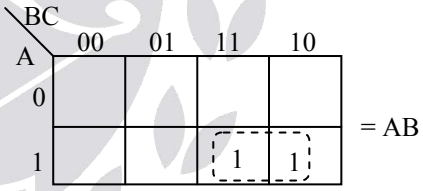
$= C\ (A \oplus B)$

$F_2 = \sum m\ (3, 5)$



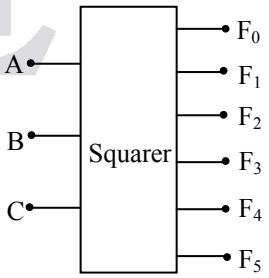
$= A\overline{B} + AC$

$F_1 = \sum m\ (4, 5, 7)$



$= AB$

$F_0 = \sum m\ (6, 7)$



04.
Sol:

(i)

Excess – 3 code of 38 = 0110 1011
Excess – 3 code of 37 = 0110 1010
1100 (1) 0101

(If there is carry out add 1101 0101
–0011 +0011
1010 1000

Corrected sum in Excess – 3 is = $(75)_{10}$

(ii)

Excess – 3 code of 129 is = 0100 0101 1100
Excess – 3 code of 131 is = 0100 0110 0100
1000 1100 0000
–0011 –0011 +0011
0101 1001 0011

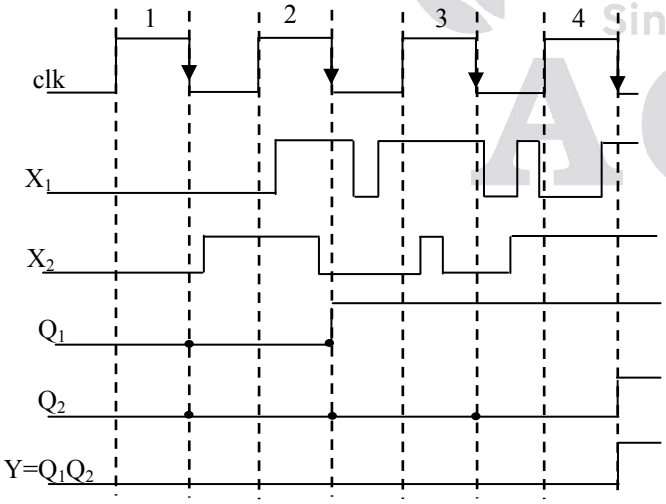
Corrected sum in Excess-3 code is = $(260)_{10}$

5. Sequential Circuits

Solutions for Objective Practice Questions

01. Ans: (c)

Sol: Given Clk, X_1, X_2
Output of First D-FF is Q_1
Output of Second D-FF is Q_2



02. Ans: 4

Sol: In the given first loop of states, zero has repeated 3 times. So, minimum 4 number of Flip-flops are needed.

03. Ans: 7

Sol: The counter is cleared when
$Q_DQ_CQ_BQ_A = 0110$

Clk	Q_D	Q_C	Q_B	Q_A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	0	0	0

As the clear input is given to be synchronous so it waits upto the next clock pulse to clear the counter & hence the counter get's cleared during the 7^{th} clock pulse.
$\therefore$ mod of counter = 7

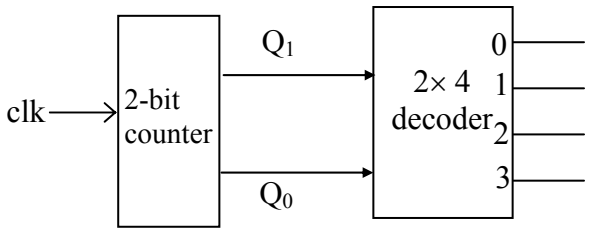
04. Ans: (b)

Sol: The given circuit is a mod 4 ripple down counter. Q_1 is coming to 1 after the delay of $2\Delta t$.

CLK	Q_1	Q_0
	0	0
1	1	1
2	1	0
3	0	1
4	0	0

05. Ans: (c)

Sol: Assume n = 2



Outputs of counter is connected to inputs of decoder

Counter outputs Q_1	Q_0	Decoder inputs a	b	Decoder outputs d_3	d_2	d_1 d_0	
0	0	0	0	0	0	0	1
0	1	0	1	0	0	1	0
1	0	1	0	0	1	0	0
1	1	1	1	1	0	0	0

The overall circuit acts as 4-bit ring counter n = 2

$\therefore$ $k = 2^2 = 4$, k-bit ring counter

06. Ans: (b)

Sol:

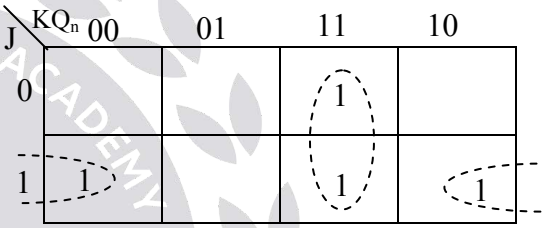
CLK	Serial in= B⊕C⊕D	A B C D
0		1 0 1 0
1	1	1 1 0 1
2	0	0 1 1 0
3	0	0 0 1 1
4	0	0 0 0 1
5	1	1 0 0 0
6	0	0 1 0 0
7	1	1 0 1 0

$\therefore$ After 7 clock pulses content of shift register become 1010 again.

07. Ans: (b)

Sol:

J	K	Q_n	$\overline{Q}_n$	$T = (J + Q_n)(K + \overline{Q}_n)$	Q_{n+1}
0	0	0	1	0.1 = 0	0
0	0	1	0	1.0 = 0	1 } Q_n
0	1	0	1	0.1 = 0	0
0	1	1	0	1.1 = 1	0 } 0
1	0	0	1	1.1 = 1	1
1	0	1	0	1.0 = 0	1 } 1
1	1	0	1	1.1 = 1	1
1	1	1	0	1.1 = 1	0 } $\overline{Q}_n$



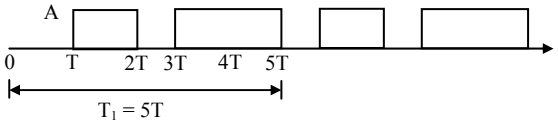
$T = J\,\overline{Q_n} + KQ_n = (J+Q_n)(K+\overline{Q_n})$

08. Ans: 1.5

Sol:

Clk	Q_1	Q_2	Q_3	Q_4	Q_5	$Y = Q_3 + Q_5$
0	0	1	0	1	0	0
1	0	0	1	0	1	1
2	1	0	0	1	0	0
3	0	1	0	0	1	1
4	1	0	1	0	0	1
5	0	1	0	1	0	0

The waveform at OR gate output, Y is [A = +5V]



Average power

$$P = \frac{V_{Ao}^2}{R} = \frac{1}{R}\left[\lim_{T_1\to\infty}\frac{1}{T_1}\int_o^{T_1} y^2(t)\,dt\right]$$

$$= \frac{1}{RT_1}\left[\int_T^{2T} A^2 dt + \int_{3T}^{5T} A^2 dt\right]$$

$$= \frac{A^2}{RT_1}\left[(2T - T) + (5T - 3T)\right]$$

$$= \frac{A^2.3T}{R(5T)} = \frac{5^2.3}{10\times 5} = 1.5\,mW$$

09. Ans: (b)

Sol:

Present State	Next State		Output (Y)	
	X = 0	X = 1	X = 0	X = 1
A	A	E	0	0
B	C	A	1	0
C	B	A	1	0
D	A	B	0	1
E	A	C	0	1

Step (1):

By replacing state B as state C then state B, C are equal.

Reducing state table		
Present state	Next state	
	X = 0	X = 1
A	A	E
B	B	A
B	B	A
D	A	B
E	A	B

Step (2):

Reducing state table		
Present state	Next state	
	X = 0	X = 1
A	A	E
B	B	A
D	A	B
E	A	B

State D, E are equal, remove state E and replace E with D in next state.

Reducing state table		
Present state	Next state	
	X = 0	X = 1
A	A	D
B	B	A
D	A	B
D	A	B

Finally reduced state table is

Reduced state table		
Present state	Next state	
	X = 0	X = 1
A	A	D
B	B	A
D	A	B

∴ 3 states are present in the reduced state table.

10. Ans: (c)

Sol: State table for the given state diagram

State	Input	Output
S_0	0	1

S_0	1	0
S_1	0	1
S_1	1	0

Output is 1's complement of input.

11. Ans: (c)

Sol: In state (C), when XYZ = 111, then Ambiguity occurs

Because, from state (C)

$\Rightarrow$ When X = 1, Z = 1

$\Rightarrow$ N.S is (A)

When Y = 1, Z = 1 $\Rightarrow$ N.S is (B)

12. Ans: (c)

Sol: For Asynchronous sequential circuits clock is applied at one flip flop and the next stage receives clock from previous stage output.

13. Ans: (d)

Sol: Master slave JK flip flop is a edge triggered flip flop.

14. Ans: (b)

Sol: Divider

: Bi stable multivibrator

Clips input voltage at Two predetermined levels

: Schmitt trigger

Square wave generator

: Astable multivibrator

Narrow current pulse generator

: Blocking oscillator

15. Ans: (a)

Sol: A flip-flop is a bistable multivibrator.

A flip-flop remains in one stable state indefinitely until it is directed by an input signal to switch over to the other stable state.

Both statements are correct and statement-II is correct explanation of statement-I.

16. Ans: (a)

Sol: The collection of all state variables (memory element stored values) at any time, contain all the information about the past, necessary to account for the circuit's future behaviour.

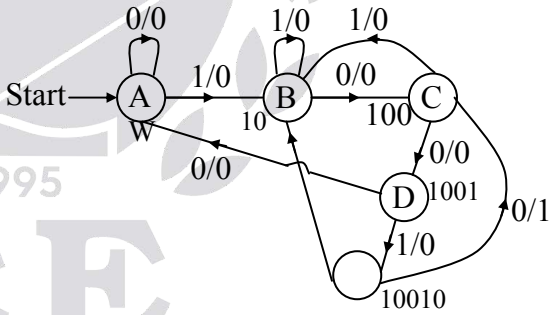
A change in the stored values in memory elements changes the sequential circuit from one state to another.

Both statements are correct and statement - II is correct explanation of statement-I.

Solutions for Conventional Practice Questions

01.

Sol: State diagram:



State table:

Input	Present state	Next state	Output
0	A	A	0
1	A	B	0
0	B	C	0
1	B	B	0
0	C	D	0
1	C	B	0
0	D	A	0
1	D	E	0
0	E	C	0
1	E	B	0

Implementing the given sequence detector using D-flipflops:

Input S	Present state Q_2 Q_1 Q_0	Next state D_2 D_1 D_0	Output
0	0 0 0	0 0 0	0
0	0 0 1	0 1 0	0
0	0 1 0	0 1 1	0
0	0 1 1	0 0 0	0
0	1 0 0	0 1 0	1
0	1 0 1	× × ×	0
0	1 1 0	× × ×	0
0	1 1 1	× × ×	0
1	0 0 0	0 0 1	0
1	0 0 1	0 0 1	0
1	0 1 0	0 0 1	0
1	0 1 1	1 0 0	0
1	1 0 0	0 0 1	0
1	1 0 1	× × ×	0
1	1 1 0	× × ×	0
1	1 1 1	× × ×	0

K-maps:

SQ_2 \ Q_1Q_0	00	01	11	10
00				
01		×	×	×
11		×	×	×
10			1	

$$D_2 = SQ_1Q_0$$

SQ_2 \ Q_1Q_0	00	01	11	10
00		1		1
01	1	×	×	×
11		×	×	×
10				

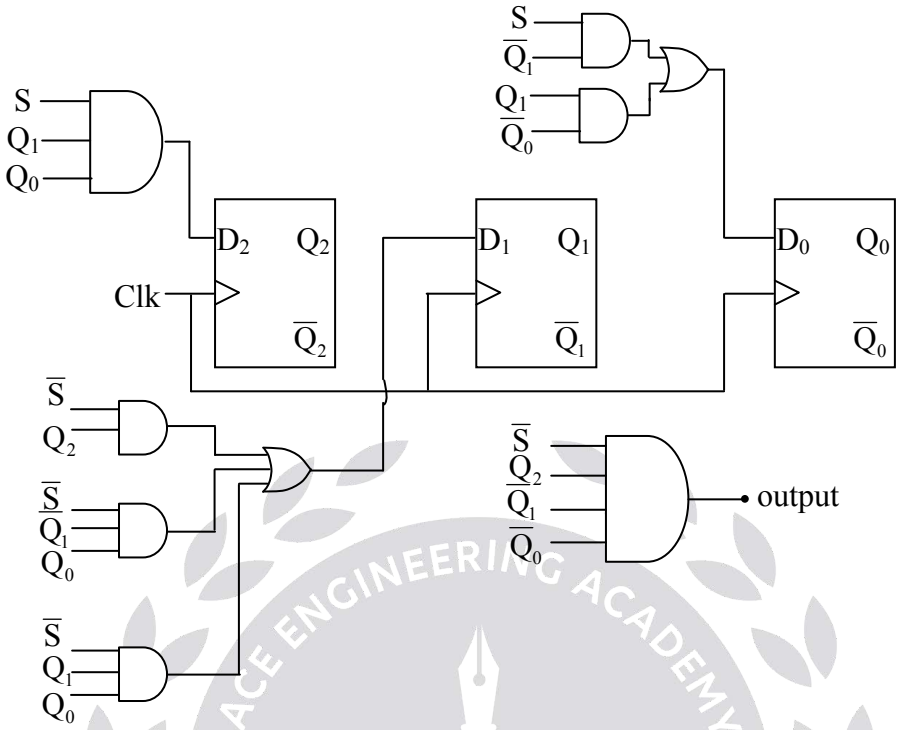
$$D_1 = \overline{S}Q_2 + \overline{S}Q_1Q_0 + \overline{S}Q_1Q_0$$

SQ_2 \ Q_1Q_0	00	01	11	10
00				1
01		×	×	×
11	1	×	×	×
10	1	1		1

$$D_0 = S\overline{Q_1} + Q_1\overline{Q_0}$$

$$\text{Output} = \overline{S}Q_2\overline{Q_1}\overline{Q_0}$$

Circuit diagram:



02.
Sol:

Present State			Next State			Flip Flop Inputs					
Q	**Q**	**Q**	**Q**	**Q**	**Q**	**J**	**K**	**J**	**K**	**J**	**K**
0	0	0	0	1	0	0	×	1	×	0	×
0	1	0	1	0	1	1	×	×	1	1	×
1	0	1	1	1	0	×	0	1	×	×	1
1	1	0	0	0	0	×	1	×	1	0	×

Excitation Table

Q_n	Q_{n+1}	J	K
0	0	0	×
0	1	1	×
1	0	×	1
1	1	×	0

1, 3, 4, 7 are minterms taken as don't cares for this problem.

$K_2 = \sum(6) + \sum d(0,1,2,3,4,7)$

Q_2 \ Q_1Q_0	00	01	11	10
0	×	×	×	×
1	×		×	1

$K_2 = Q_1$

$K_1 = \sum m(2,6) + \sum d(0,1,3,4,5,7)$

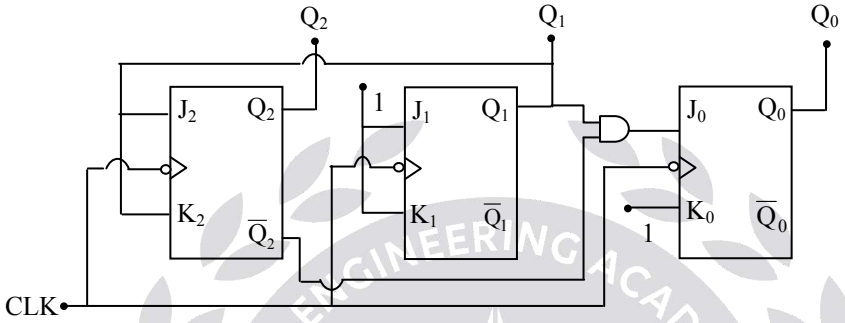
Q_2 \ Q_1Q_0	00	01	11	10
0	×	×	×	1
1	×	1	×	1

$K_1 = 1$

$K_0 = \sum m(5) + \sum d(0,1,2,3,4,6,7)$

Q_2 \ Q_1Q_0	00	01	11	10
0	×	×	×	×
1	×	1	×	×

$K_0 = 1$



03.

Sol:

Present State			Next State			FF Inputs		
Q_2	Q_1	Q_0	Q_2^+	Q_1^+	Q_0^+	$D_2 = O_2$	$D_1 = O_1$	$D_0 = O_0$
1	0	1	0	1	0	0	1	0
0	1	0	1	1	0	1	1	0
1	1	0	1	0	1	1	0	1

K – map for O_2:

Q_2 \ Q_1Q_0	00	01	11	10
0	× 0	× 1	× 3	1 2
1	× 4	0 5	× 7	1 6

$\Rightarrow O_2 = Q_1$

K-map for O_1:

Q_2 \ Q_1Q_0	00	01	11	10
0	× 0	× 1	× 3	1 2
1	× 4	1 5	× 7	0 6

$O_1 = \overline{Q_2} + \overline{Q_1} = \overline{Q_2 Q_1}$

K-map for O_0:

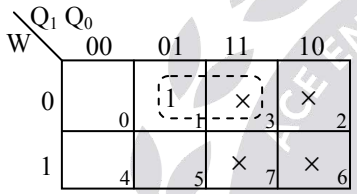
Q_2 \ Q_1Q_0	00	01	11	10
0	X 0	X 1	X 3	0 2
1	X 4	0 5	× 7	1 6

$O_0 = Q_2 Q_1$

04.
Sol:

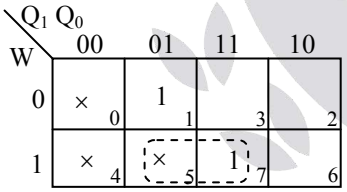
I/P	Present State		Next State		FF I/Ps			
W	Q_1	Q_0	Q_1^+	Q_0^+	J_1	K_1	J_0	K_0
0	0	0	0	0	0	×	0	×
0	0	1	1	0	1	×	×	1
0	1	0	1	0	×	0	0	×
0	1	1	1	1	×	0	×	0
1	0	0	0	1	0	×	1	×
1	0	1	0	1	0	×	×	0
1	1	0	1	1	×	0	1	×
1	1	1	0	0	×	1	×	1

K- map for J_1:



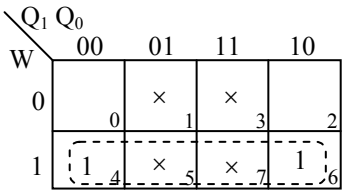
$J_1 = \overline{W}Q_0$

K- map for K_1:



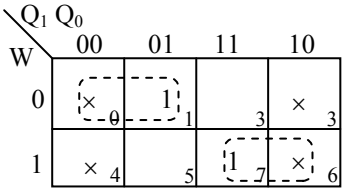
$K_1 = WQ_0$

K- map for J_0:

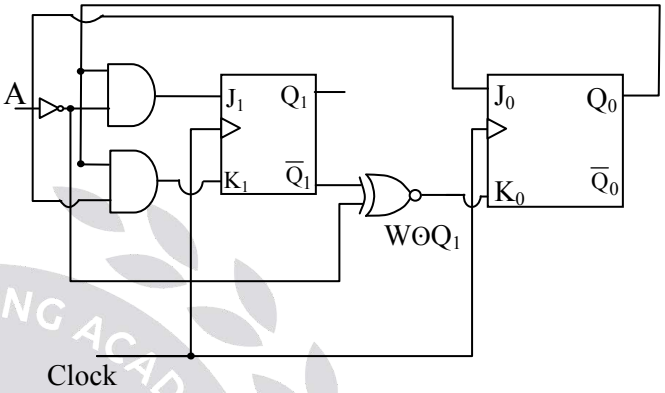


$J_0 = W$

K- map for K_0:



$K_0 = \overline{W}\,\overline{Q_1} + WQ_1$



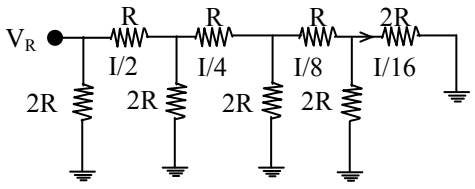
6. AD and DA Converters

Solutions for Objective Practice Questions

01. Ans: (b)
Sol:

CLK	Counter			Decoder				V_0
	Q_2	Q_1	Q_0	D_3	D_2	D_1	D_0	
1	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	1	1
3	0	1	0	0	0	1	0	2
4	0	1	1	0	0	1	1	3
5	1	0	0	1	0	0	0	8
6	1	0	1	1	0	0	1	9
7	1	1	0	1	0	1	0	10
8	1	1	1					11
				1	0	1	1	

02. Ans: (b)
Sol:



R_{equ} = (((((2R||2R)+R)||2R)+R)||2R)+R)||2R)

$R_{equ} = R = 10k\Omega$.

$$I = \frac{V_R}{R} = \frac{10V}{10k} = 1mA.$$

Current division at $\frac{I}{16}$

$$= \frac{1\times10^{-3}}{16} = 62.5\,\mu A$$

03. Ans: (c)

Sol: Net current at inverting terminal,

$$I_i = \frac{I}{4} + \frac{I}{16} = \frac{5I}{16}$$

$$V_0 = -I_i R = -\frac{5I}{16}\times 10k\Omega$$

$$= \frac{-5\times1\times10^{-3}\times10\times10^{3}}{16} = -3.125V$$

04. Ans: (d)

Sol: Given that $V_{DAC} = \sum_{n=0}^{3} 2^{n-1} b_n$ Volts

$$V_{DAC} = 2^{-1}b_0 + 2^0 b_1 + 2^1 b_2 + 2^2 b_3$$

$\Rightarrow V_{DAC} = 0.5b_0 + b_1 + 2b_2 + 4b_3$

Initially counter is in 0000 state

Up counter o/p $b_3\ b_2\ b_1\ b_0$	V_{DAC}(V)	o/p of comparator
0 0 0 0	0	1
0 0 0 1	0.5	1
0 0 1 0	1	1
0 0 1 1	1.5	1
0 1 0 0	2	1
0 1 0 1	2.5	1
0 1 1 0	3	1
0 1 1 1	3.5	1
1 0 0 0	4	1
1 0 0 1	4.5	1
1 0 1 0	5	1
1 0 1 1	5.5	1
1 1 0 0	6	1
1 1 0 1	6.5	0

When V_{DAC} = 6.5 V, the o/p of comparator is '0'. At this instant, the clock pulses to the counter are stopped and the counter remains in 1101 state.

$\therefore$ The stable reading of the LED display is 13.

05. Ans: (b)

Sol: The magnitude of error between V_{DAC} & V_{in} at steady state is $|V_{DAC} - V_{in}| = |6.5 - 6.2|$

= 0.3 V

06. Ans: (a)

Sol: In Dual slope

ADC $\Rightarrow V_{in}T_1 = V_R.T_2$

$$\Rightarrow V_{in} = \frac{V_R T_2}{T_1}$$

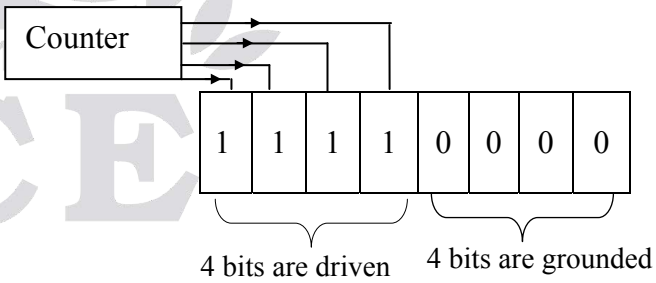
$$= \frac{100\,mV\times370.2\,ms}{300\ ms}$$

DVM indicates = 123.4

07. Ans: (d)

Sol: No. of bits = 8,

Reference voltage = 8V



Maximum peak to peak amplitude of the waveform at the output of the digital to analog converter is

$$V_{max} = \frac{V_{ref}}{2^n}\left(d_n 2^n\right)$$

$$= \frac{8}{256}\times240 = 7.5V$$

08. Ans: (d)

Sol: Ex: $f_{in} = 1$ kHz $\rightarrow f_s = 2$ kHz

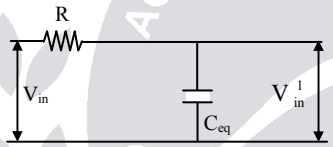
$f_{in} = 25$ kHz $\leftarrow f_s = 50$ kHz

1. Max conversion time $= 2^{N+1}T = 2^{11}.1\ \mu s$
$= 2048\ \mu s$
2. Sampling period $= T_s \geq$ maximum conversion time
$T_s \geq 2048\ \mu s$
3. Sampling rate $f_s = \frac{1}{T_s} \leq \frac{1}{2048\times10^{-6}}$
$f_s \leq 488 \quad f_s \leq 500$ Hz
4. $f_{in} = \frac{f_s}{2} = 250\,Hz$

09. Ans: (b)

Sol:

$$V_{in}^{1} = \frac{V_{in}}{RC_{eq}}T$$



V_{in}^{1} has to settle down within $\frac{1}{2}$ LSB of full scale value.

i.e $\frac{509}{510}V_{in} = \frac{V_{in}.T}{75\times(255\times8\times10^{-12})}$

$\Rightarrow T = (75\times255\times8\times10^{-12})\times\frac{509}{510}$

$T \approx 0.15$ µsec

Thus sample period $T_s \geq T$

$T_s \geq 0.15$ m sec

$$f_s\ max = \frac{1}{Ts_{,min}}$$

$$= \frac{1}{0.15\times10^{-6}}\,Hz$$

≈ 6 Megasamples

10. Ans: (a)

Sol: Dual-slope A/D converter is the most preferred A/D conversion approach in digital multimeters. Dual-slope A/D converter provides high accuracy in A/D conversion, while at the same time suppressing the hum effect on the input signal.

Both Statements are true and statement-II is the correct explanation of statement-I.

11. Ans: (d)

Sol: SAR type ADC : Settling time for n-bits is (n+2) T clock pulses

Flash ADC : (2^n-1) comparators required for n-bit dual

Dual slope ADC : Works well even in noisy environment

Counter DAD : Settling time dependent on the input

12. Ans: (c)

Sol: Dual slope ADC : Hum rejection approximation

Counter-ramp ADC : Conversion time dependent on single amplitude

Successive ADC : Fixed conversion time, depends on the number of bits

Simultaneous ADC : High speed operation

13. Ans: (a)

Sol: The output of an 8-bit A to D converter is 40H for an input of 2.5V.

ADC has an output range of 00 to FFH for an input range of –5V to +5V.

Both Statements are true and statement-II is the correct explanation of statement-I.

14. Ans: (c)

Sol: Digital ramp converter is the slowest ADC.

Conversion time for digital ramp ADC is not N^2T.

15. Ans: (b)

Sol: Resolution for n-bit A/D converter in percentage.

$$= \frac{1}{2^n - 1} \times 100$$

$$= \frac{1}{2^{12} - 1} \times 100$$

$$= 2.442 \times 10^{-4} \times 100 = 0.02442$$

Solutions for Conventional Practice Questions

01.

Sol: We know that (from superposition theorem)

$V_{01} = -V_{Ref}\,(b_0 + b_1\,2^1 + b_2\,2^2 + b_3\,2^3 + b_4\,2^4 + b_5\,2^5 + b_6\,2^6 + b_7\,2^7)$

$V_{02} = -V_{Ref}\,(b_8 + b_9\,2^1 + b_{10}\,2^2 + b_{11}\,2^3 + b_{12}\,2^4 + b_{13}2^5 + b_{14}\,2^6 + b_{15}\,2^7)$

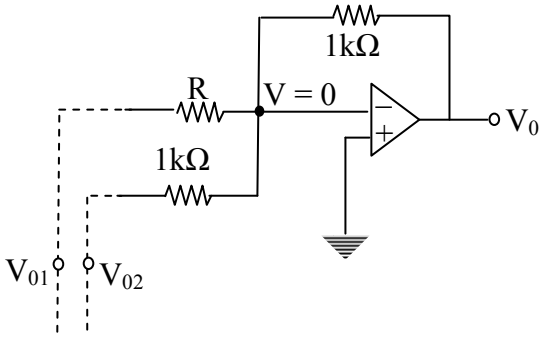
The correct value corresponding to an 16-bit DAC is,

$V_0 = -V_{Ref}\,[\,b_0 + b_1\,2^1 + \text{- - - -} + b_{15}\,2^{15}]$

from virtual ground concept

$$\frac{0 - V_{01}}{R} + \frac{0 - V_{02}}{1k} + \frac{0 - V_0'}{1k} = 0$$

$$\therefore \frac{V_{01}}{1} + \frac{V_{02}}{R} = \frac{-V_0'}{1}$$



Analog output $V_0 = -V_0'$

$$\frac{V_0}{1} = \frac{-V_{Ref}(b_0 + b_1\,2^1 + \ldots + b_7\,2^7)}{1} + \frac{V_{Ref}}{R}[b_8 + \ldots\ldots + b_{15}2^7]$$

$V_0 = -V_{Ref}[b_0 + b_1\,2^1 + \ldots + b_{15}\,2^{15}]$

Comparing, we have

$R = 0.5\ k\Omega$

02.

Sol: a) Given $f = 100$ kHz

$$\tau = \frac{1}{f} = \frac{1}{100k} = 10^{-5}\ \text{sec}$$

N = number of bits = 8

Maximum conversion time of an 8 bit digital ramp ADC is $2^n\,\tau$

$\tau_{Ramp} = 2^n\tau = 2^8 \times 10^{-5} = 2.56$ m sec

$= 2560\ \mu$ sec

Maximum conversion time of successive approximation type counter of 8 bit is $n\,\tau$.

$\tau_{Successive} = n\,\tau = 8 \times 10^{-5} = 80\ \mu$ sec

Maximum conversion time of flash type ADC

$\tau_{Flash} = \tau = 10^{-5}$

$= 10\ \mu$ sec

$$\frac{\tau_{ramp}}{\tau_{successive}} = \frac{2560}{80} = 128;\quad \frac{\tau_{successive}}{\tau_{Flash}} = \frac{80}{10} = 8$$

$$\frac{\tau_{ramp}}{\tau_{Flash}} = \frac{2560}{10} = 256$$

b) 3 - bit Flash type ADC:

The Flash converter is the highest – speed ADC. The Flash converter in figure has a 3-bit resolution and a step size of 1V. The voltage divider sets up reference levels for each comparator so that there are seven levels

corresponding to 1V, 2V…….7V. The V_A is connected to another input of each comparator. With V_A< 1V all the comparator outputs C_1 to C_7 will be "HIGH". With V_A > 1V, one or more of the comparator outputs will be low.

The comparator outputs are fed into an active - low priority encoder that generates a binary output corresponding to the highest - numbered comparator output that is "Low".

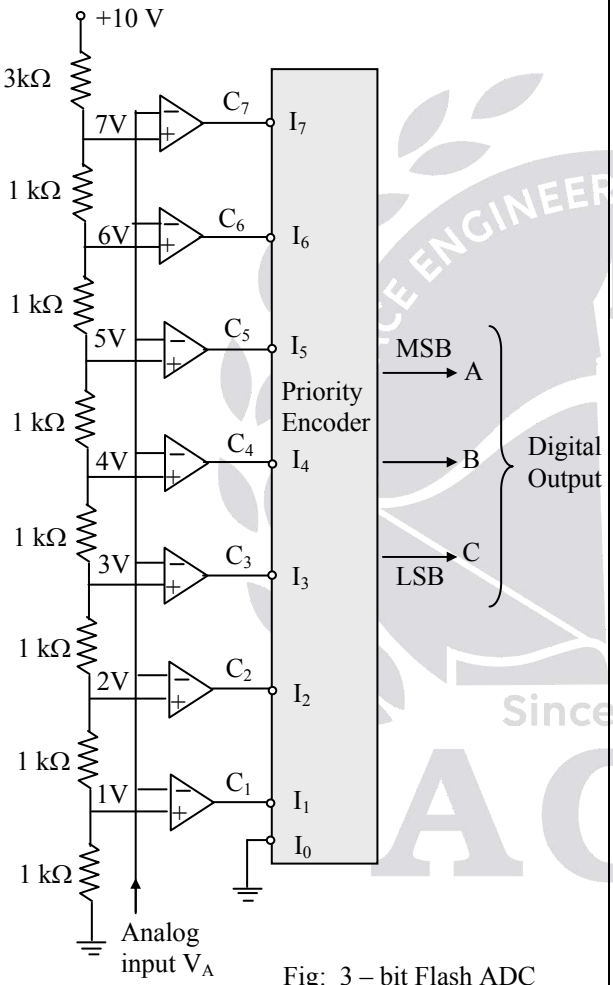
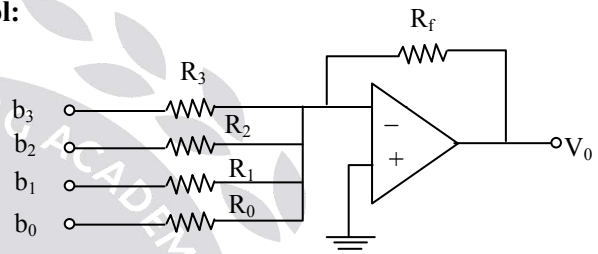


Fig: 3 – bit Flash ADC

Analog input	Comparator outputs							Digital Outputs		
V_A	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C	B	A
0 - 1 V	1	1	1	1	1	1	1	0	0	0
1 - 2 V	0	1	1	1	1	1	1	0	0	1
2 - 3 V	0	0	1	1	1	1	1	0	1	0
3 - 4 V	0	0	0	1	1	1	1	0	1	1
4 - 5 V	0	0	0	0	1	1	1	1	0	0
5 - 6 V	0	0	0	0	0	1	1	1	0	1
6 - 7 V	0	0	0	0	0	0	1	1	1	0
> 7 V	0	0	0	0	0	0	0	1	1	1

03.
Sol:



$$V_0 = \left[\frac{-R_f b_3}{R_3} - \frac{R_f b_2}{R_2} - \frac{R_f b_1}{R_1} - \frac{R_f b_0}{R_0}\right] V_{ref}$$

Comparing fig (i) with the figure given in the question $b_3\, b_2\, b_1\, b_0 = 1011$,

$R_f = 100k\Omega$; $R_3 = 100k\Omega$; $R_2 = 200k\Omega$; $R_1 = 400k\Omega$; $R_0 = 800k\Omega$; $V_{ref} = 5V$

$$\therefore V_0 = \frac{-100}{100}\times 5 - \frac{100}{200}\times 0 - \frac{100}{400}\times 5 - \frac{100}{800}\times 5$$

$$= -5 - \frac{1}{4}\times 5 - \frac{5}{8} = -6.875V$$

$V_0 = -6.875V$

7. Architecture, Pin Details of 8085 & Interfacing with 8085

Solutions for Objective Practice Questions

01. Ans: (a)
Sol: chip select is an active low signal for $\overline{\text{chipselect}} = 0$; the inputs for NAND gate must

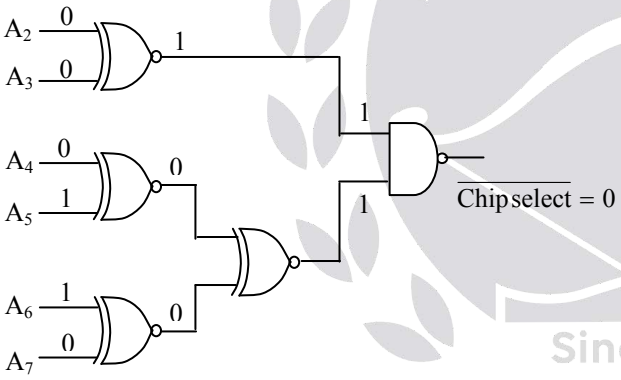
be let us see all possible cases for $\overline{\text{chipselect}} = 0$ condition

A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
0	0	0	0	0	0	X	X	
0	0	1	1	0	0	X	X	
0	1	0	1	0	0	X	X	
0	1	1	0	0	0	X	X	→ 60H (A_1A_0=00)
1	0	0	1	0	0	X	X	
1	0	1	0	0	0	X	X	
0	0	0	0	1	1	X	X	
0	0	1	1	1	1	X	X	
0	1	0	1	0	0	X	X	
0	1	1	0	0	0	X	X	→63H(A_1A_0=11)
1	0	0	1	0	0	X	X	
1	0	0	0	0	0	X	X	

The only option that suits hare is option(a)

A_0 & A_1 are used for line selection

A_2 to A_7 are used for chip selection



∴ Address space is 60H to 63H

02. Ans: (d)

Sol:
- Both the chips have active high chip select inputs.
- Chip 1 is selected when $A_8 = 1$, $A_9 = 0$
  Chip 2 is selected when $A_8 = 0$, $A_9 = 1$
- Chips are not selected for combination of 00 & 11 of A_8 & A_9
- Upon observing A_8 & A_9 of given address Ranges, F800 to F9FF is not represented

03. Ans: (d)

Sol: The I/O device is interfaced using "Memory Mapped I/O" technique.

The address of the Input device is

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	
1	1	1	1	1	0	0	0	1	1	1	1	1	0	0	0	$=F8F8_H$

The Instruction for correct data transfer is = LDA F8F8H

04. Ans: (b)

Sol:
- Output 2 of 3×8 Decoder is used for selecting the output port. ∴Select code is 010

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	----	A_0
0	1	0	1	0	0	----	0

$\Rightarrow$ 5000H

- This mapping is memory mapped I/O

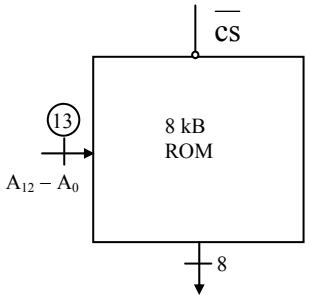
05. Ans: (d)

Sol:

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9 - - - - A_0	
0	0	0	0	1	0	0 - - - - 0	=0800H
0	0	0	0	1	0	1 - - - - 1	=0BFFH
0	0	0	1	1	0	0 - - - - 0	=1800H
0	0	0	1	1	0	1 - - - - 1	=1BFFH
0	0	1	0	1	0	0 - - - - 0	=2800H
0	0	1	0	1	0	1 - - - - 1	=2BFFH
0	0	1	1	1	0	0 - - - - 0	=3800H
0	0	1	1	1	0	1 - - - - 1	=3BFFH

06. Ans: (a)

Sol: Address Range given is



	A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0
1000H →	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
2FFFH →	0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1

To provide $\overline{cs}$ as low, The condition is

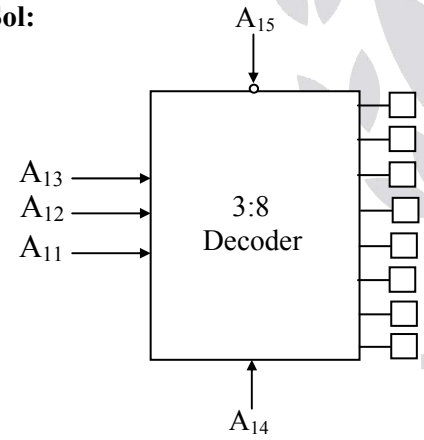
$A_{15} = A_{14} = 0$ and $A_{13}\ A_{12} = 01$ (or) (10)

i.e $A_{15} = A_{14} = 0$ and $A_{13}\ A_{12}$ shouldn't be 00, 11.

Thus it is $A_{15} + A_{14} + [A_{13}A_{12} + \overline{A_{13}}\ \overline{A_{12}}]$

07. Ans: (a)

Sol:



A_{15}, A_{14} are used for chip selection
A_{13}, A_{12}, A_{11} are used for input of decoder

A_{15} A_{14}	A_{13} A_{12} A_{11}	A_{10} ------ A_0
Enable of decoder	Input of decoder	Address of chip

Size of each memory block = 2^{11} = 2K

08. Ans: (a)

Sol: The data path contains all the circuits to process data within the CPU with the help of which data is suitably transformed.

It is the responsibility of the control path to generate control and timing signals as required by the opcode.

Both Statements are true and statement-II is the correct explanation of statement-I.

09. Ans: (b)

Sol: Program counter is a register that contains the address of the next instruction to be executed.

IR (Instruction Register) is not accessible to programmer.

Both Statements are true but statement-II is not correct explanation of statement-I.

10. Ans: (a)

Sol: A processor can reference a memory stack without specifying an address.

The address is always available and automatically updated in the stack pointer.

Both Statements are true and statement-II is the correct explanation of statement-I.

11. Ans: (c)

Sol: The programmer has to initialize the stack pointer based on design requirements.

12. Ans: (b)

Sol: The DMA technique is more efficient than the Interrupt-driven technique for high volume I/O data transfer.

The DMA technique does not make use of the Interrupt mechanism.

Both Statements are true but statement-II is not correct explanation of statement-I.

13. Ans: (c)

Sol: A microcontroller has onchip (inbuilt) memory, where as a microprocessor has no such internal memory.

The program to be run by microprocessor is to be store in separate memory (E^2PROM) chip and to be interfaced microprocessor.

14. Ans: (d)

Sol: INTR is a non vectored interrupt. As such external hardware is required to supply vector address. SIM is not used with respect to INTR. The SIM is used for selective local masking of three hardware maskable vectored interrupts (RST 7.5, RST 6.5, RST 5.5)

Solutions for Conventional Practice Questions

01.

Sol: Given that, microprocessor has

Number of address lines (m) = 20

Number of data lines (n) = 16

(i) Addressing capacity = $2^m = 2^{20}$

Data handling capacity = -2^{n-1} to $(2^{n-1} - 1)$

$= -2^{15}$ to $(2^{15} - 1)$

(ii) Number of memory ICs required

$$= \frac{2^{20} \times 16}{2^{16} \times 8} = 32$$

02.

Sol:

- Interrupt enable flipflop gets disabled by 8085 when it vectors to an ISR after recognizing occurance of an interrupt. As such, all the maskable interrupts are disabled automatically to avoid re-entrance.
- At the end of ISR, the programmer has to include EI instruction which sets the interrupt enable flipflop and enables the maskable interrupts.
- It is necessary to enable all the maskable interrupts before coming out of ISS.

03.

Sol:

- M_1 is program memory (ROM)
- M_2 & M_3 are Data memories (RAM)
- A_{11} to A_{15} of 8085 are used for chip selection for each memory.

  A_0 to A_{10} of 8085 are used for line selection within each memory.
- Size of each memory is 2KB since 11 Address lines are used for line selection

Memory Address map

A_{15}	A_{14}	A_{13}	A_{12}	A_{11}	A_{10}	A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0		
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	=0000H	M_1
			.							.							
			.							.							
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	=07FFH	
1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	=8800H	M_2
			.							.							
			.							.							
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	=8FFFH	
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	=A000H	M_3
			.							.							
1	0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	=A7FFH	

04.

Sol: **1.** Two 2732 A (4 kB) EPROM ICs are interfaced to 8085
4 KB = 2^{12} B has 12 Address input pins ($A_{11} - A_0$)
Byte difference for 4 kB is FFFH
EPROM1 - Starting address is 0000H
End address is 0000H
0000H + FFFH = OFFFH
EPROM2 - Starting address is 1000H
End address is 1000H
1000H + FFFH = 1FFFH

2. Two 6116 (2kB) RAM ICs are also interfaced to 8085
2 kB = 2^{11}B has 11 address input Pins ($A_{10} - A_0$)
Byte difference for 2 kB is 7FFH
RAM1 – Starting address is 2000H
End address is 2000H +
7FFH = 27FFH
RAM2 – starting address is 3000H (assuming discontinuous address mapping)
End address is 3000H +
7FFH = 37FFH

3. Two 8255 ICs are also interfaced to 8085,
8255 IC_1 - 4000H to 4003H
8255IC_2 - 5000H to 5003H
Assuming discontinuous address mapping

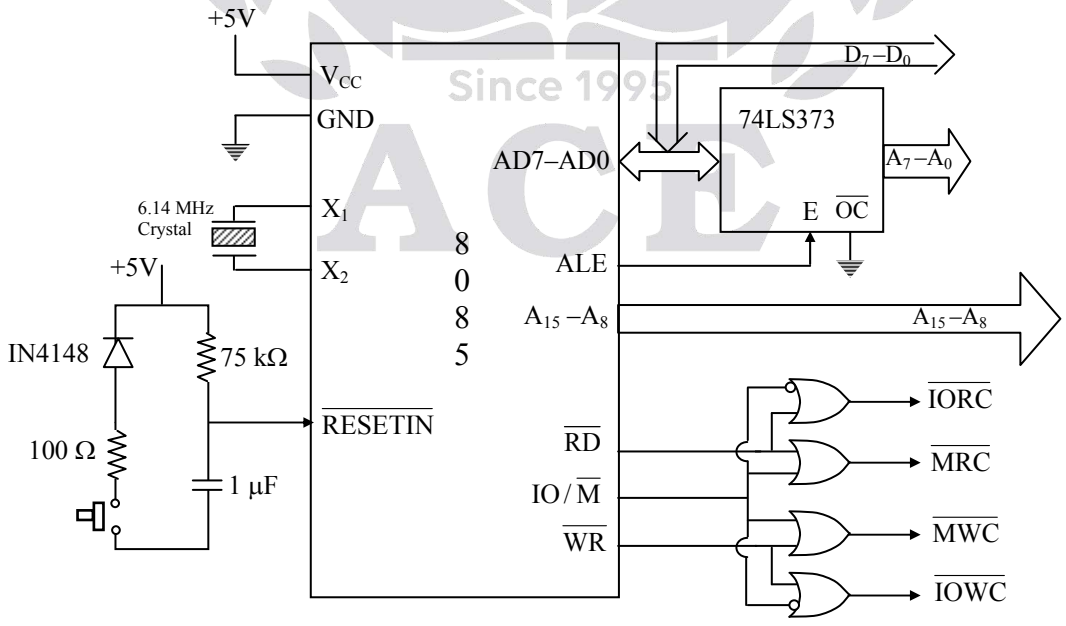
Address Map:

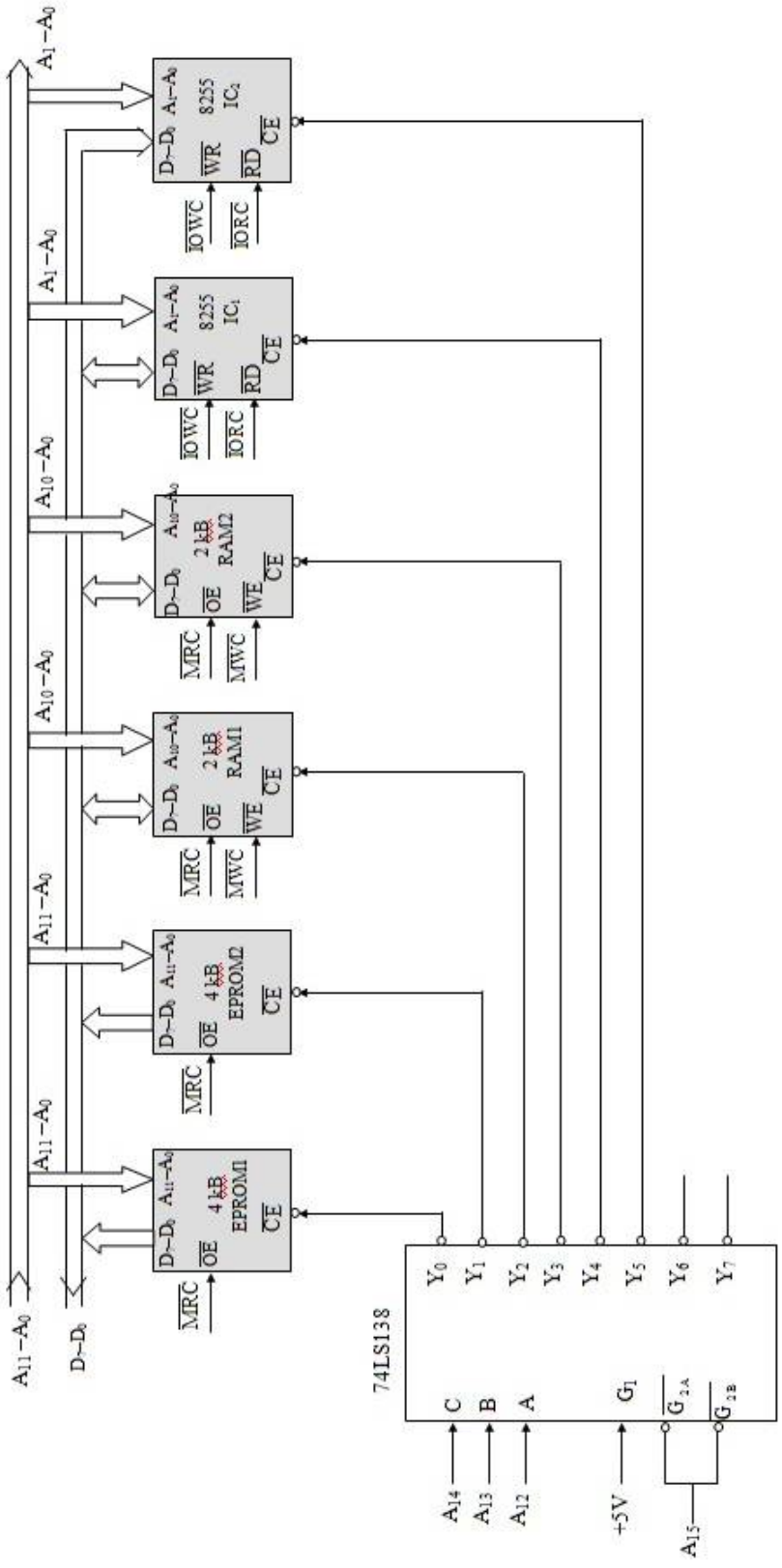
		$A_{15}A_{14}A_{13}A_{12}$	$A_{11}A_{10}A_9A_8$	$A_7A_6A_5A_4$	$A_3A_2A_1A_0$
4 KB EPROM1	0000H	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0
	0FFFH	0 0 0 0	1 1 1 1	1 1 1 1	1 1 1 1
4 KB EPROM2	1000H	0 0 0 1	0 0 0 0	0 0 0 0	0 0 0 0
	1FFFH	0 0 0 1	1 1 1 1	1 1 1 1	1 1 1 1
2 KB RAM1	2000H	0 0 1 0	0 0 0 0	0 0 0 0	0 0 0 0
	27FFH	0 0 1 0	0 1 1 1	1 1 1 1	1 1 1 1
2 KB RAM2	3000H	0 0 1 1	0 0 0 0	0 0 0 0	0 0 0 0
	37FFH	0 0 1 1	0 1 1 1	1 1 1 1	1 1 1 1
8255 IC1	4000H	0 1 0 0	0 0 0 0	0 0 0 0	0 0 | 0 0
	4003H	0 1 0 0	0 0 0 0	0 0 0 0	0 0 | 1 1
8255 IC2	5000H	0 1 0 1	0 0 0 0	0 0 0 0	0 0 | 0 0
	5003H	0 1 0 1	0 0 0 0	0 0 0 0	0 0 | 1 1

- It can be observed from above address map that A_{14} - A_{12} of 8085 can be used for selecting output lines of 74LS138 (3 × 8 decoder)
- For each EPROM IC, A_{11} - A_0 of 8085 can be used for byte selection
- For each RAM IC, A_{10} - A_0 of 8085 can be used for byte selection
- For each 8255 IC, A_1 - A_0 of 8085 can be used for port selection
- Decoding logic of 74LS138 is given below

A_{15}=0 ($\overline{G_{2B}}$ $\overline{G_{2A}}$)	1 (G1)	A_{14} A_{13} A_{12} (C B A)	Y_0 Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7	
0 0	1	0 0 0	0 1 1 1 1 1 1 1	→ Output Y_0 connected to $\overline{CE}$ of EPROM1
0 0	1	0 0 1	1 0 1 1 1 1 1 1	→ Output Y_1 connected to $\overline{CE}$ of EPROM2
0 0	1	0 1 0	1 1 0 1 1 1 1 1	→ Output Y_2 connected to $\overline{CE}$ of RAM1
0 0	1	0 1 1	1 1 1 0 1 1 1 1	→ Output Y_3 connected to $\overline{CE}$ of RAM2
0 0	1	1 0 0	1 1 1 1 0 1 1 1	→ Output Y_4 connected to $\overline{CE}$ of 8255 IC_1
0 0	1	1 0 1	1 1 1 1 1 0 1 1	→ Output Y_5 connected to $\overline{CE}$ of 8255 IC_2

4. The below given figure shows demultiplexing the time multiplexed address/data bus of 8085, power on reset circuit for 8085, generation of required control signals.




A11–A0
D–D0
MRC
OE
CE
4 kB EPROM1
4 kB EPROM2
2 kB RAM1
2 kB RAM2
WE
MWC
IOWC
IORC
WR
RD
8255 IC1
8255 IC2
74LS138
Y0
Y1
Y2
Y3
Y4
Y5
Y6
Y7
C
B
A
G1
G2A
G2B
A14
A13
A12
+5V
A15

8. Instruction set of 8085 & Programming with 8085

Solutions for Objective Practice Questions

01. Ans: (c)

Sol:

6010H : LXI H,8A79H ; (HL) = 8A79H
6013H : MOV A, L ; (A)←(L) = 79
6014H : ADD H ; (A) = 0111 1001
+
; (H) = 1000 1010
; (A) = 0000 0011
CY = 1, AC = 1
6015H : DAA ; 66 Added to (A) since CY = 1 & AC = 1
; (A) = 69H
6016H : MOV H,A ; (H) ← (A) = 69H
6017H : PCHL ; (PC)←(HL) = 6979H

02. Ans: (c)

Sol: 0100H : LXI SP, 00FFH ; (SP) = 00FFH
0103H : LXI H, 0107 H ; (HL) = 0107H
0106H : MVI A, 20H ; (A) = 20H
0108H : SUB M ; (A)←(A)-(0107)
; (0107) = 20H
; (A) = 00H
The contents of Accumulator is 00H

03. Ans: (c)

Sol: LXI SP, 00FFH ; (SP) = 00FFH
LXI H, 0107 H ; (HL) = 0107H
MVI A, 20H ; (A) = 20H
SUB M ; (A) ← (A) – (0107)
; (0107) = 20H = M
; (A) = 00H
ORI 40H ; A ∀ 40H
A = 40H
ADD M ; 40H + 20H = 60H

04. Ans: (c)

Sol: SUB1 : MVI A, 00H A← 00H
CALL SUB2 → program will shifted to SUB 2 address location

A

SUB 2 : INR A → | 01H |

RET → returned to the main program
∴ The contents of Accumulator after execution of the above SUB2 is 02H

05. Ans: (c)

Sol: The loop will be executed until the value in register equals to zero, then,
Execution time
=9(7T+4T+4T+10T)+(7T+4T+4T+7T)+7T
= 254T

06. Ans: (d)

Sol: H = 255 : L = 255, 254, 253, ------0
H = 254 : L = 0, 255, 254, -------0
|
H = 1 : L = 0, 255, 254, 253, -----0
H = 0 : ——

→ In first iteration (with H = 255), the value in L is decremented from 255 to 0 i.e., 255 times

→ In further remaining 254 iterations, the value in L is decremented from 0 to 0 i.e., 256 times
∴ 'DCRL' instruction gets executed for
$\Rightarrow [255 + (254 \times 256)]$
$\Rightarrow 65279$ times

07. Ans: (a)

Sol: "STA 1234H" is a 3-Byte Instruction and it requires 4 Machine cycles (Opcode fetch, Operand1 Read, Operand2 Read, Memory write). The Higher order Address (A_{15} – A_8) sent in 4 machine cycles is as follows

Given "STA 1234" is stored at 1FFEH

i.e., Address Instruction

1FFE, 1FFF, 2000 : STA 1234H

Machine cycle	Address (A_{15}-A_0)	Higher order address (A_{15}-A_8)
1. Opcode fetch	1FFEH	1FH
2. Operand1 Read	1FFFH	1FH
3. Operand2 Read	2000H	20H
4. Memory Write	1234H	12H

i.e. Higher order Address sent on A_{15}-A_8 for 4 Machine Cycles are 1FH, 1FH, 20H, 12H.

08. Ans: (d)

Sol: The operation SBI BE_H indicates A-BE $\rightarrow$ A where A indicates accumulator

Thus the result of the subtraction operation is stored in the accumulator and the contents of accumulator are changed.

09. Ans: (c)

Sol: If the content in register B is to be multiplied with the content in register C, the contents of register B is added to the accumulator (initial value of accumulator is 0) for C times.

10. Ans: (c)

Sol: Push takes 12T states due to pre decrement and pop takes 10T states.

11. Ans: (d)

Sol:

CY

Given A = $A7_H$ = 1 0100111 0

After executing RLC $\Rightarrow$ A = 0 1001111 1

A = $4F_H$ and CY = 1

12. Ans: (b)

Sol: **OUT**: output data from accumulator to a port with 8-bit addresses. The contents of the accumulator are copied into the I/O ports specified by the operand.

IN: Input data to accumulator from a port with 8-bit address. The contents of the input port designated in the operand are read and loaded into the accumulator.

13. Ans: (a)

Sol: When RET instruction is executed by any subroutine then the top of the stack will be popped out and assigned to the PC.

14. Ans: (b)

Sol: PUSH PSW $\Rightarrow$ 1 Byte instruction

$\Rightarrow$ OPFC + 2T + MW1C + MW2C

$\Rightarrow$ Special OPFC + MW1C + MW2C

$\Rightarrow$ 3 Machine cycles

15. Ans: (c)

Sol: Flags are not affected for execution of data transfer instructions since there is no involvement of ALU.

16. Ans: (a)

Sol: Immediate addressing : LXI H, 2050H
Implied addressing : RRC
Register addressing : MOV A, B
Direct addressing : LDA 30FF

17. Ans: (c)

Sol: 'DAD' instruction adds contents of HL register pair with specified register pair contents and stored in HL register pair.

18. Ans: (a)

Sol: Format of instruction Template:-

Label	Mnemonics	operand	comments

19. Ans: (b)

Sol: Implicit addressing mode
: RAL
Register-indirect addressing mode
: MOV A, M
Immediate addressing mode
: JMP 3FAOH
Direct addressing mode :
LDA 03FCH

20. Ans: (a)

Sol: Total no. of machine cycles in CALL instruction is 18.

1. Opcode fetch = 6T
2. Two memory READ machine cycles to read subroutine address = 3T + 3T = 6T
3. Two memory WRITE machine cycles on the stack = 3T + 3T = 6T

$\therefore$ I/O was not used in CALL instruction.

21. Ans: (d)

Sol: PCHL : Transfer the contents of HL to the program counter.
SPHL : Transfer the contents of HL to the stack pointer
XTHL : Exchange the top of the stack with the contents of HL pair
XCHG : Exchange the contains of HL with those of DE pair

Solutions for Conventional Practice Questions

01.

Sol: i) DAD H; (HL) + (HL)
This instruction doubles the 16 bit number in HL pair which is equivalent to shifting that 16 bit number to left by 1 bit.

ii) First instruction should be initialization instruction for loading stack pointer (SP) with required 16 bit RAM address.

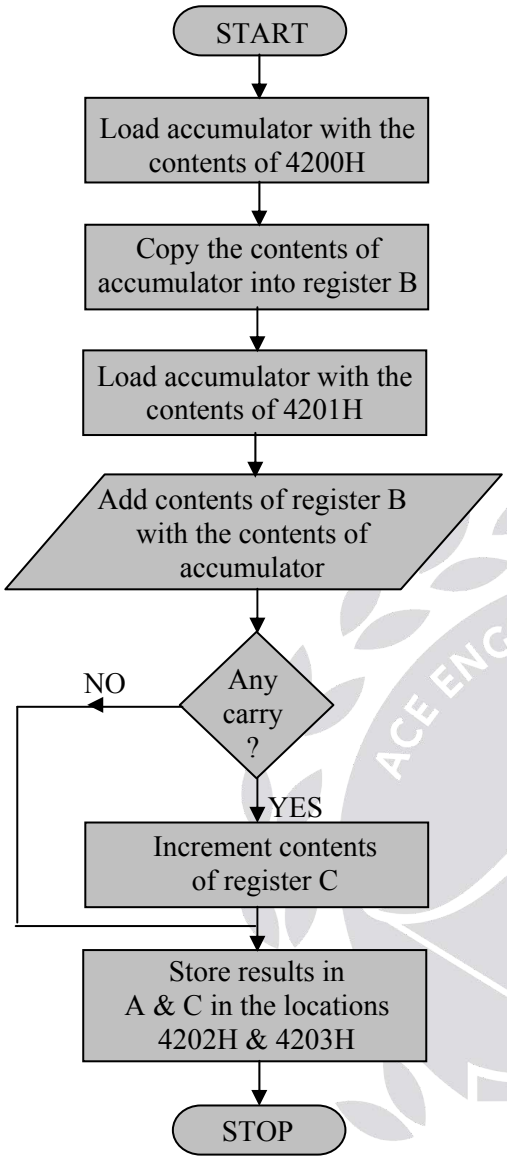
Ex: LXI SP, 2500H
i.e., stack should be properly initialized.

02.

Sol:

```
        ORG   0000H
        JMP   MAIN
        ORG   0100H
MAIN:   XRA   A
        MOV   C,A
        LDA   4200H
        MOV   B,A
        LDA   4201H
        ADD   B
        JNC   END
        INR   C

END:    STA   4202H
        MOV   A,C
        STA   4203H
        HLT
```



03.

Sol:

```
        ORG   0000H
        JMP   MAIN
        ORG   0100H
MAIN:   LDA   2040H
        MOV   B, A
        ANI   0FH
        STA   2041H
        MOV   A, B
        ANI   F0H
        RRC
        RRC
        RRC
        RRC
        STA   2042H
        HLT
```

04.

Sol:

```
        ORG   0000H
        JMP   MAIN
        ORG   2501H
NUM     DB    96H
        ORG   2100H
MAIN:   LDA   2501H
        CMA
        ADI   01H
        STA   2502H
        HLT
```

06.

Sol:

	12FFH = 4863d	Operation	No. of times of Execution
	LXI B, 12FFH	; (BC) ← 12FFH	1 time
DELAY:	DCX B	; (BC)←(BC)-1	4863 times
	XTHL	; (TOS)←(HL)	4863 times
	NOP	; No operation	4863 times
	NOP	; No operation	4863 times
	MOV A,C	; (A)←(C)	4863 times
	ORA B	; (A)←(A) ∨ (B)	4863 times
	JNZ DELAY	; Jump to DELAY, if Z = 0	4863 times

Total T-states = 1×(10T) + 4863 [6T + 16T + 4T + 4T + 4T + 4T + 10T] – 3T
= 10T + 23342T – 3T
= 233431T

Time = 233431T × 0.30μs
= 70029.3μS
= 70.029ms
≅ 70.03 ms

03. Ans: (c)

Sol:

- 8086 μp has 20 Address output lines. As such, a total of about 2^{20} i.e., 1MB memory can be directly addressed by 8086 μP
- The programming model of 8086 μP has the following registers
  AX, BX, CX, DX
  CS, DS, SS, ES
  Flag registers, SP, IP, BP, SI, DI i.e., a total no. of 14 registers
- There are total 9 flags in 8086 μp and the flag register is divided into two types.
  (a) Status flags: The six status flags are
    1. Sign flag (S)
    2. Zero flag (Z)
    3. Auxiliary carry flag (AC)
    4. Parity flag (P)
    5. Carry flag (CY)
    6. Overflow flag (O)
  (b) Control flags: The three control flags are
    1. Directional flag (D)
    2. Interrupt flag (I)
    3. Trap flag (T)

D_{15}	D_{14}	D_{13}	D_{12}	D_{11}	D_{10}	D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
				O	D	I	T	S	Z		AC		P		CY

Fig: Format of flag register

04. Ans: (c)

Sol: Setting Trap flag puts the processor into single mode for debugging. In single stepping microprocessor executes an instruction and enters into single step ISR. If TF = 1, the CPU automatically generates an internal interrupt after each instruction, allowing a program to be inspected as it executes instruction by instruction.

05. Ans: (b)

Sol: For 8086 μP, the jump distance in bytes for short jump range is forward 127 and backward 128.

06. Ans: (a)

Sol: Number of address lines in 8086 is 20. Address space is 2^{20} = 1MB.

07. Ans: (d)

Sol: The instruction queue length in 8086 is 6 bytes and in 8088 is 4 bytes.

08. Ans: (d)

Sol: 8086 microprocessor can be operated in multiprocessor configuration when MN/$\overline{MX}$ input connected to ground.

09. Ans: (d)

Sol: A 16 bit μP completes access of a word starting from even address in one bus cycle.

10. Ans: (b)

Sol: In relative base indexed Addressing mode, the 20 bit physical address of Data segment location is calculated as followed.

P.A = (D.S register)×10H + (B× register) + (DI register) + 16 bit displacement

= 2100H×10H + 0158H + 1045H + 1B57H

= 21000H + 2CF4H

= 23CF4H

11. Ans: (a)

Sol: Effective Address = (C.S reg)×10H + (IP reg)

= 1FABH × 10H + 10A1H

= 20B51H

12. Ans: (c)

Sol: SI is the source index, used as a pointer to the current character being read in a string instruction. It is also available as an offset to add to BX or BP when doing indirect addressing.

DI is the destination index, used as a pointer to the current character being written or compared in a string instruction. It is also available as an offset.

13. Ans: (b)

Sol: The intermediate wait states are always, inserted between the clock cycles T_2 and T_3.

14. Ans: (a)

Sol: For a type 0 interrupt, the 8086 pushes the flag register on the stack, resets IF and TF and pushes the return addresses on the stack.

15. Ans: (c)

Sol: The interrupt vector table IVT of 8086 contains the starting CS and IP values of the interrupt service routine.

16. Ans: (d)

Sol: The 8086 arithmetic instructions work on Signed and unsigned numbers Unpacked BCD data.

17. Ans: (c)

Sol: LOOP and ROTATE instructions of an 8086 μp uses the contents of a CX register as a counter.

18. Ans: (c)

Sol: In a multi-processor configuration, the two co-processor instruction sets must be disjoint.

19. Ans: (b)

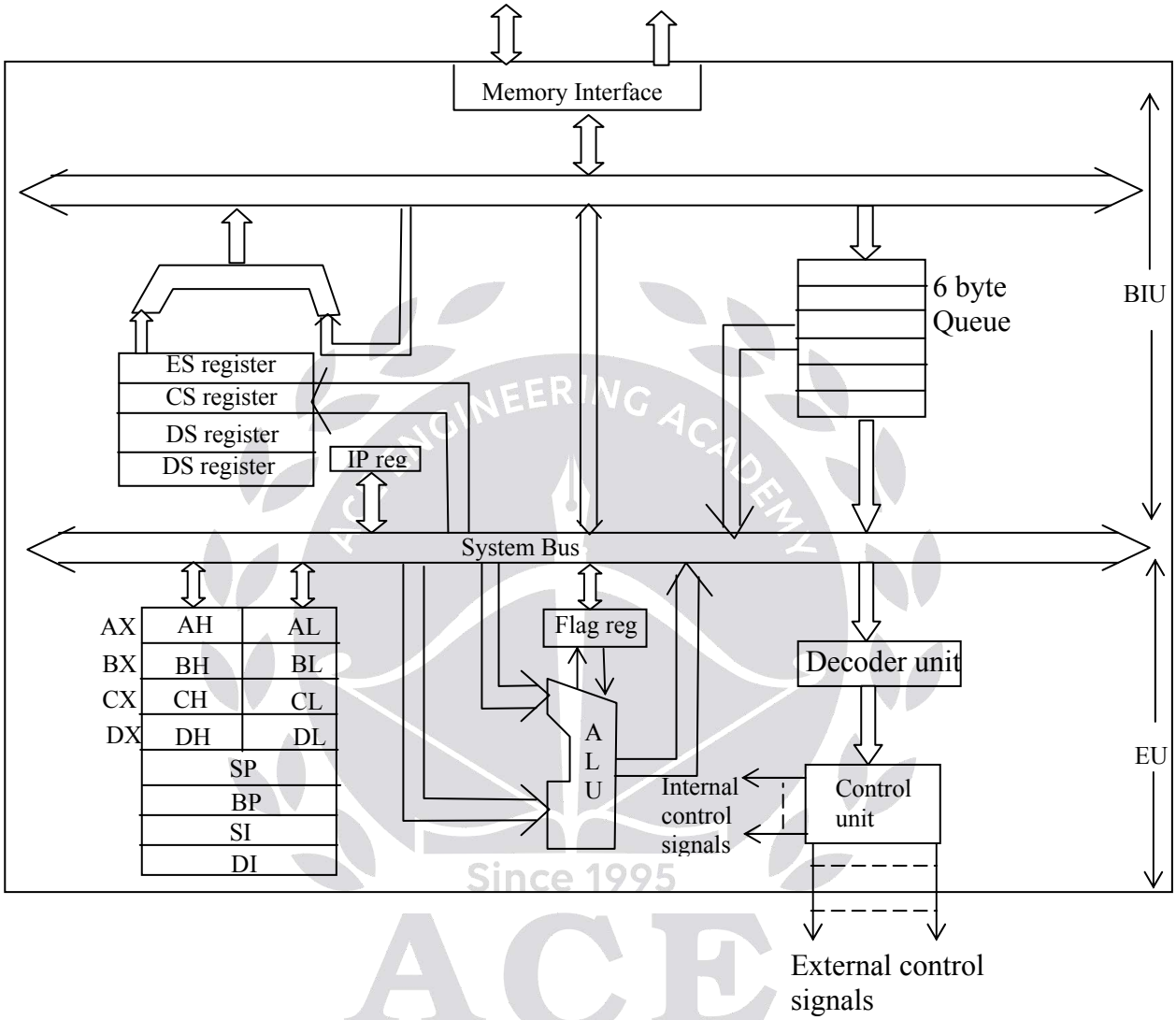
Sol: MOV [1234 H], AX

Move the contests of register AX to memory offset 1234 H and 1235 H

Solutions for Conventional Practice Questions

01.

Sol: ARCHITECTURE OF 8086 MICROPROCESSOR



SPECIAL FUNCTIONS OF GENERAL PURPOSE REGISTERS OF 8086

There are four 16bit general purpose Registers namely AX, BX, CX & DX registers in 8086μP.
There are some special functions assigned to each general purpose register as given below

- **Special functions of AX Register:**

i) For IO data transfer operations, AL register acts as either source or as destination. For data output operations, AL register as source. For data input operations, AL register as destination.

Ex:
```
IN    AL, F8H
IN    AL, DX
OUT   F9H, AL
OUT   DX, AL
```

ii) For few instructions, when operands are not specified in the instruction then machine implicitly assumes accumulator as the operand.

Ex: DAA
DAS
AAA

iii) For multiplication operation, accumulator acts as one of the source operands and also as destination for result-strong similarly, accumulator acts as one of the operands in division operation.

- **Special Functions of BX Register:**
  BX register is used to hold 16 bit offset Address in few indirect memory Addressing modes.
  Ex: MOV [BX] , AL
  MOV AX, 04[BX]
- **Special function of CX Register:**
  8 bit CL register or 16bit CX register can be used as counter register in few instructions.
  Ex: Loop instructions, rotate instructions, shift instructions
- **Special Functions of DX Register:**

i) in variable IO port addressing mode, DX register is used as IO pointer register to hold 16bit port Address.

Ex: IN AL, DX
OUT DX, AL

ii) For multiplication operation of two 16 bit numbers, the higher order 16bits of 32bit result will be stored in DX register for division operation of 32bit/16bit, the 16bits of remain will be stored in DX register

02.

Sol: Disadvantages of 8085:

- 16-bit processing is complicated.
- Instruction set is simple.
- Speed is low.
- Process of fetch and execution takes place instruction by instruction.
- Less number of registers.

Advantages of 8086 over 8085

- 8086μP is a 16bit microprocessor i.e., the processing capacity and Data Handling capacity of 8086μP is 16bit
- The addressing capacity is 1MB
- fetching and execution operations can be pipelined.
- powerful instructions are made available. Instruction set is rich with string manipulation instructions and bit manipulation instructions
- can perform more complicated arithmetic and logical operations.
- high speed. Standard operating speed is 5MHz

Limitations of 8086:

- Probably the most important difference between an 8086 and a modern PC processor is that the 8086 has no hardware support for virtual memory.
- An 8086 is only one part of a complete computer system. It requires at least several other chips to function.
- One distinctive and annoying feature that was unique to the 8086 was its segmented addressing scheme. It made it difficult for any one process to grow larger than a certain limit and it was designed to run programs that had less than 64k of code and less than 64k of data. In other words, it was

designed to support what PC programmers called "small model" programs.

03.

Sol: There are 2 types of unconditional CALL instructions available in ISA of 8086 namely Intrasegment CALL instructions (NEAR CALL instructions) and inter segment CALL instructions (FAR CALL instructions)

Intra segment CALL instructions:

- The current contents of IP register is pushed into stack and is initialized with specified target address, as specified below:

  ((SP)-1) ← (IPH)
  ((SP)-2) ← (IPL)
  (SP) ← (SP)–2
  (IP) ← target address
- Based on specification of target address, there are 2 types of intra segment CALL instructions namely direct near CALL instruction and indirect near CALL instruction.
- In direct near CALL instruction, target address is directly provided in instruction.

  Ex: CALL NEAR 3000H
  CALL NEAR DELAY
- In indirect near CALL instruction, target address is available either in a register or memory.

Ex: CALL NEAR BX
CALL NEAR wordptr[3000H]
CALL NEAR wordptr[BX]
CALL NEAR wordptr[SI]

Inter segment CALL instructions:

- The current contents of CS register and IP register are pushed into stack, and are initialized with target address as given below.

  ((SP)–1) ← (CSH)
  ((SP)–2) ← (CSL)
  ((SP)–3) ← (IPH)
  ((SP)–4) ← (IPL)
  (SP) ← (SP)–4
  (CS:IP) ← target address
- Based on specification of target address, there are 2 types of inter segment CALL instructions namely direct far call instruction and indirect far call instruction.
- In direct far call instruction, target address is directly provided in the instruction.

  Ex: CALL FAR 3000:1200
  CALL FAR DELAY
- In indirect far call instruction, target address is available in memory.

  Ex: CALL FAR dwordptr [1200]
  CALL FAR dwordptr [BX]
  CALL FAR dwordptr [SI]

04.

Sol i)

- The architecture of 8086μp is divided into two functional units namely Bus interfacing unit (BIU), and Execution unit (EU).
- BIU is the fetch unit & EU is the execute unit where the functional operations of both units are asynchronous, independent but overlapping.

Functions of BIU

- Provides Bus connectivity
- Fetches code of instructions from code segment and stores them in 6 byte Queue.
- Generates 20bit physical addresses of segment locations
- Sends data to RAM locations or output devices
- Receives data from RAM locations or input devices

Functions of EU:

- Gets the code from Queue & decodes
- Using decoded version, generates necessary control signals & required for execution
- Performs Arithmetic & logical operations

Functional working of 8086μP

- Upon application of reset pulse, the Queue will be empty BIU runs instruction-fetch machine cycle and fetches code from code segment and puts in Queue
- The EU gets the code decodes it and generates machine level information regarding timing & control signals. Based on the decoded version of code, EU completes execution.
- If EU requires external memory/IO access for execution, then it makes a request to EU. Such request will be honoured by BIU only after completion of currently running fetch operation.

ii) Elements of BIU to support its functions:

- Memory interface
- All four segment Registers (CS reg, DS reg, ES reg, SS reg) and Instruction pointer
- 6 Byte Queue whose working principle is FIFO
- Shifter and Adder circuit

Elements of EU to support its functions

- Decoder unit
- Control unit
- Arithmetic and Logical unit
- Flag register
- 4 general purpose registers (AX, BX, CX, DX)
- 4 offset registers (SP, BP, SI, DI)

Instruction pipelining:

In the 8086 there is a 6 byte instruction prefetch queue which is used to prefetch instruction bytes while the processor is working on processing earlier bytes. In this way, it is statistically possible that the next opcode can be fetched and available to the processor when it is done with the prior opcode and it wants the next opcode. This is called pipelining, or caching, and it can speed up processing. Of course, if the processor branches, the prefetched instruction bytes have to be discarded. Modern processors actually have branch prediction algorithms to help this issue.

Cases	BIU	EU
Case1: Queue is Empty	Runs Instruction fetch machine cycle, fetches the code from code segment and puts in Queue	Remains idle
Case 2: Queue is full	Remains idle	Gets the code from Queue for execution
Case 3: Queue is empty with few Bytes filled	Runs Instruction fetch machine cycle, fetches code of from code segment and puts in Queue	Gets the code from Queue for execution