CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test - I

| Sub: | Analog and Digital Electronics | | | | | | | Code: | 15CS32 |
|---|---|---|---|---|---|---|---|---|---|
| Date: | 18 / 09 / 2017 | Duration: | 90 mins | Max Marks: | 50 | Sem: | 3 A,B,C | Branch: | CSE |

- PART A is compulsary
- (PART B-PART E) Answer **FOUR** FULL questions selecting AT LEAST ONE question from **each** part

| | | Marks | OBE CO | OBE RBT |
|---|---|---|---|---|
| | **PART A** | | | |
| **1** | Simplify the Boolean expression using Quine Mc-cluskey method. F(A,B,C,D)=Σm(1,3,6,7,8,9,10,12,14,15)+d(11,13) | [10] | CO2 | L3 |
| | **PART B** | | CO2 | L2 |
| 2a. | Describe positive and negative logic .Prove 'Positive OR' is equal to 'Negative AND' logic. | [4] | | |
| 2b. | Reduce the following functions using K-map technique and draw the logic diagram for the minimal sum obtained using only NAND gates. F(A,B,C,D)=Σm(0,1,2,4,5,12,14)+d(8,10) | [6] | CO2 | L3 |
| | **OR** | | | |
| 3a. | Simplify the following Boolean function using K-map in POS form and draw the logic diagram for minimal product obtained. F(A,B,C,D)=Σm(1,3,5,6,7,8,9,12,13) | [6] | CO2 | L3 |
| 3b. | Write a note on Static Hazards | [4] | CO2 | L2 |
| | **PART C** | | CO3 | L3 |
| 4a. | Design a 4:1 MUX and write its function table. | [5] | | |
| 4b. | Design 8:1 MUX using only 4:1 MUX | [5] | CO3 | L3 |
| | **OR** | | | |
| 5 a. | Explain 1 : 4 DEMUX with the following<br>• Block diagram<br>• Functional table<br>• Circuit diagram | [4] | CO3 | L3 |
| 5b. | Implement a full adder using 3 to 8 decoder | [6] | CO3 | L3 |
| | **PART D** | | CO3 | L3 |
| 6a. | Design a 4 bit priority encoder using basic gates. | [6] | | |
| 6b. | Design a 8 bit comparator using two 7485 ICs | [4] | CO3 | L3 |
| | **OR** | | | |
| | . | | | |

| | | | | |
|---|---|---|---|---|
| 7a. | Design a 5 bit magnitude comparator using IC7485. | [5] | CO3 | L3 |
| 7b. | With a neat diagram explain decimal to BCD encoder. | [5] | CO3 | L3 |
| | _____ | | | |
| | **PART E** | [10] | CO3 | L3 |
| 8 | A combinational circuit is defined by the following functions<br>F1= Σm(1,3,5) F2= Σm(5,6,7).implement the circuit using 3x3x2 PLA | | | |
| | OR | | | |
| 9 | Implement the following functions using PAL<br>F1(a,b,c,d)= Σm(0,2,6,7,8,9,12,13)<br>F2(a,b,c,d)= Σm(0,2,6,7,8,9,12,13,14)<br>F3(a,b,c,d)= Σm(2,3,8,9,10,12,13)<br>F4(a,b,c,d)= Σm(1,3,4,6,9,12,14) | [10] | CO3 | L3 |

----------ALL THE BEST----------

CMR INSTITUTE OF TECHNOLOGY

USN | | | | | | | | | | |

Internal Assesment Test - I

| Sub: | Analog And Digital Electronics | | | | | | | Code: | 15CS32 |
|---|---|---|---|---|---|---|---|---|---|
| Date: | 18 / 09 / 2017 | Duration: | 90 mins | Max Marks: | 50 | Sem: | 3 A,B,C | Branch: | CSE |

- PART A is compulsary
- (PART B-PART E) Answer **FOUR** FULL questions selecting AT LEAST ONE question from **each** part

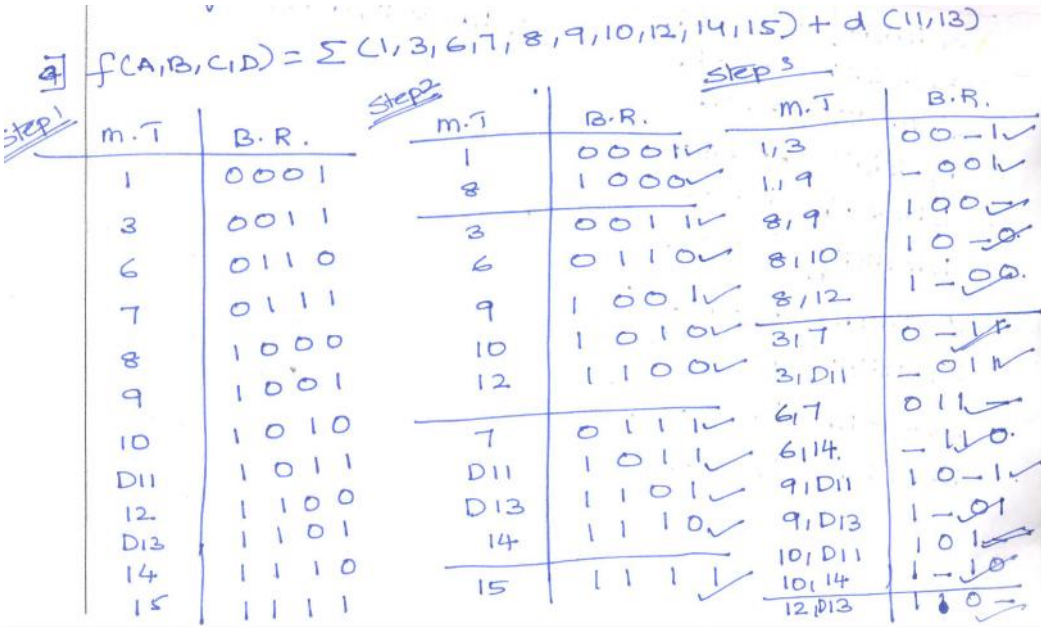
| | | Marks | OBE | |
|---|---|---|---|---|
| | | | CO | RBT |

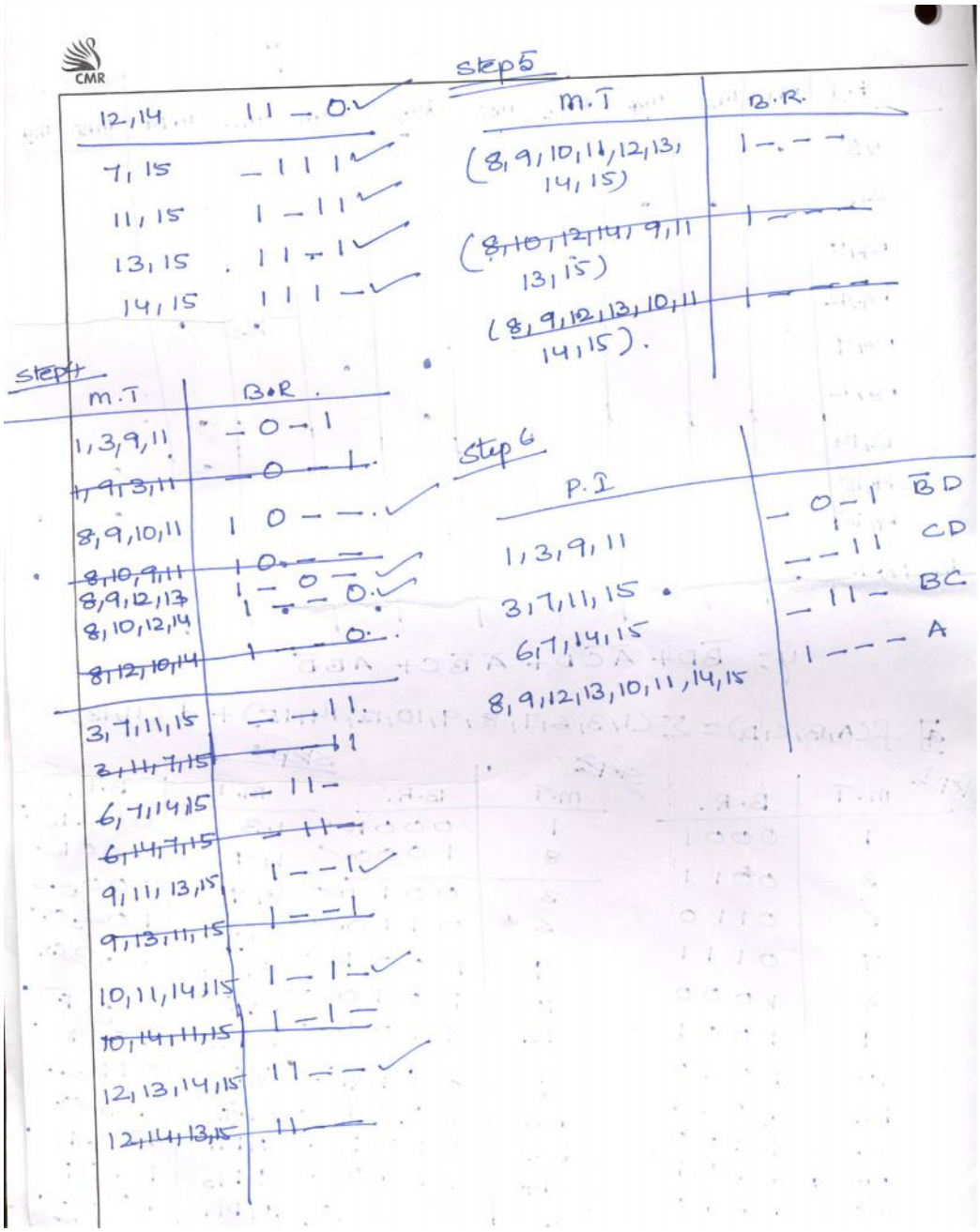
**PART A**

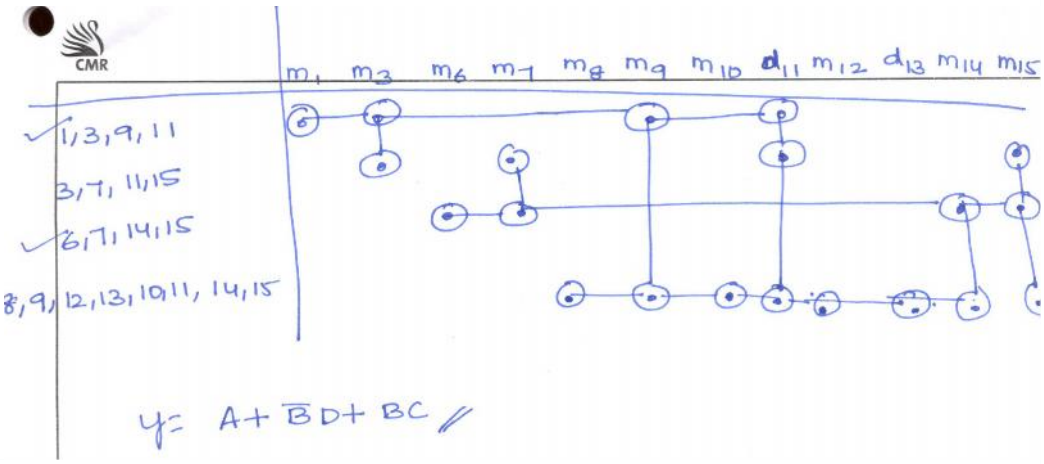
**1** Simplify the Boolean expression using Quine Mc-cluskey method. F(A,B,C,D)=Σm(1,3,6,7,8,9,10,12,14,15)+d(11,13) [10] CO2 L3

**Scheme:**
Step1 to Step 5 each 1 mark
Step 6 – 2 marks (prime implicants)
Essential prime implicant table – 2 marks
Boolean expression – 1mark

**Solution**




CMR
step5
12,14 11 - 0
7,15 - 1 1 1
11,15 1 - 1 1
13,15 1 1 - 1
14,15 1 1 1 -
M.T
B.R
(8,9,10,11,12,13,14,15)
1 - - -
(8,10,12,14,9,11,13,15)
(8,9,12,13,10,11,14,15).
step4
M.T
B.R
1,3,9,11 - 0 - 1
8,9,10,11 1 0 - -
8,10,9,11
8,9,12,13 1 - 0 -
8,10,12,14 1 - - 0
8,12,10,14
3,7,11,15 - - 1 1
3,11,7,15
6,7,14,15 - 1 1 -
6,14,7,15
9,11,13,15 1 - - 1
9,13,11,15
10,11,14,15 1 - 1 -
10,14,11,15
12,13,14,15 1 1 - -
12,14,13,15
step6
P.I
1,3,9,11 - 0 - 1 B̄D
3,7,11,15 - - 1 1 CD
6,7,14,15 - 1 1 - BC
8,9,12,13,10,11,14,15 1 - - - A


CMR
m1 m3 m6 m7 m8 m9 m10 d11 m12 d13 m14 m15
1,3,9,11
3,7,11,15
6,7,14,15
8,9,12,13,10,11,14,15
Y= A + B̄D + BC

## PART B

| | | | | |
|---|---|---|---|---|
| 2a. | Describe positive and negative logic .prove 'positive OR' equal to 'Negative AND' logic . | [4] | CO2 | L2 |

Solution :-

Positive Logic :- (1 mark)

- If we use binary 0 for low voltage and binary 1 for high voltage, it is called positive logic.
- choosing H-1=T and L-0=F is called positive logic.

Negative Logic :- (1 mark)

- If we use binary 0 for high voltage and binary 1 for low voltage, it is called Negative Logic.
- choosing H-1=F and L-0=T is called Negative Logic.

Proof :- (2 marks)

OR Gate Truth Table with Positive Logic [0 = L, 1 = H]

| A | B | Y = A+B |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

⟺

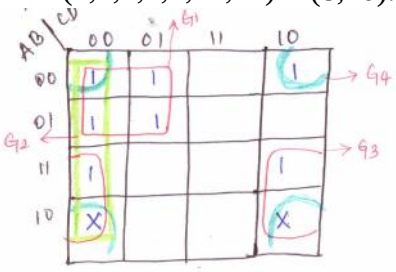
In Negative Logic [0-H, 1-L] same Truth Table is written as

| A | B | Y |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 0 |

This is AND Gate in Negative Logic.

| | | | | |
|---|---|---|---|---|
| 2b. | Reduce the following function using K-map techniques and draw its logic diagram using only NAND gates. F(A,B,C,D)=Σm(0,1,2,4,5,12,14)+d(8,10). | [6] | CO2 | L3 |

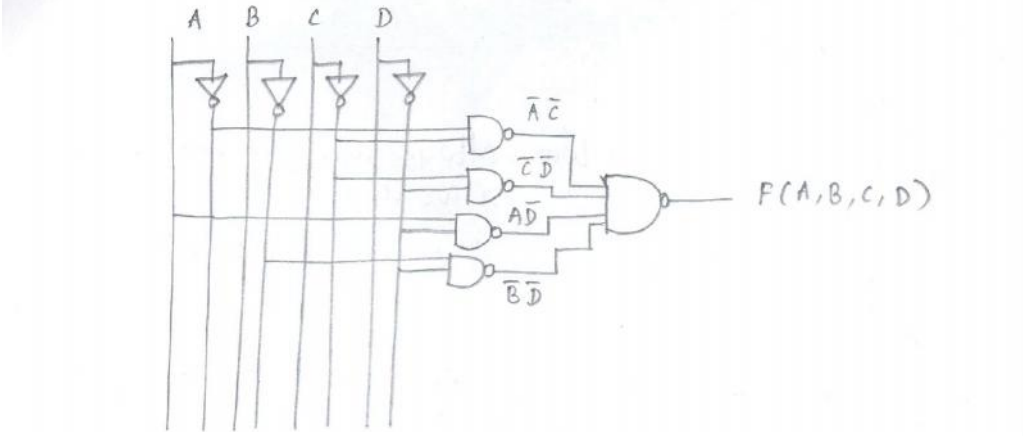
(4 quads)




Mapping - 1 mark
Grouping - 2 marks
Boolean Expression - 1 mark
Logic diagram - 2 marks

Boolean expression:-

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



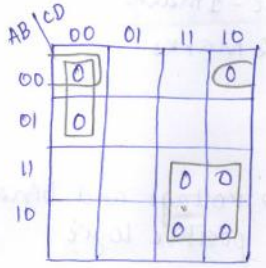

**OR**

| | | | | |
|---|---|---|---|---|
| 3a. | Simplify the following Boolean function using K-map in POS form and draw its logic diagram.<br>$F(A,B,C,D)=\Sigma m(1,3,5,6,7,8,9,12,13)$. | [6] | CO2 | L3 |

TIM conversion - 1 mark
Mapping - 1 mark
Grouping - 1 mark
Expression - 2 mark
Logic diagram - 1 mark.

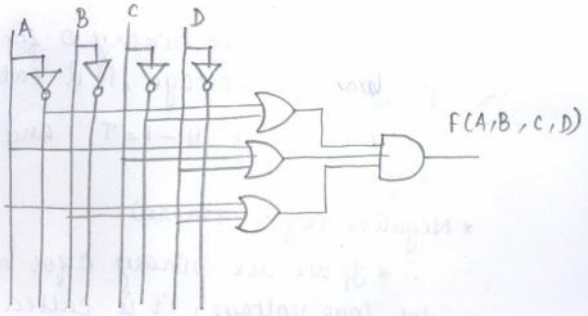
Sol:-

$F(A,B,C,D) = \Pi M(0,2,4,10,11,14,15)$




$\bar{F}(A,B,C,D) = AC + \bar{A}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{D}$

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




| | | | | |
|---|---|---|---|---|
| 3b. | Write a note on Static Hazards. | [4] | CO2 | L2 |

Static Hazards:-

There are two types of static Hazards.

Static - 0 Hazard: (2 marks)

→ This type of Hazard occurs when $Y = A \cdot A'$ type of situation appear for a logic circuit and when A makes a transition $0 \to 1$.

→ An $A \cdot A'$ condition should always generate 0 at the output (Static - 0) - But, the NOT gate output takes finite time to become 0 following $0 \to 1$ transition of A.

Static - 1 Hazard :- (2 marks)

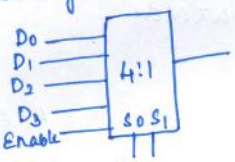
→ This type of Hazard occurs when $Y = A + A'$ type of situation occurs for a logic circuit and when A makes a transition from 1 to 0.

→ An $A + A'$ condition always generate 1 at the output side (static-1), but the NOT gate output takes finite time to become 1 following 1 to 0 transition of A

## PART C

| | | | | |
|---|---|---|---|---|
| 4a. | Design a 4:1 MUX and write its function table. | [5] | CO3 | L3 |

Block diagram:-



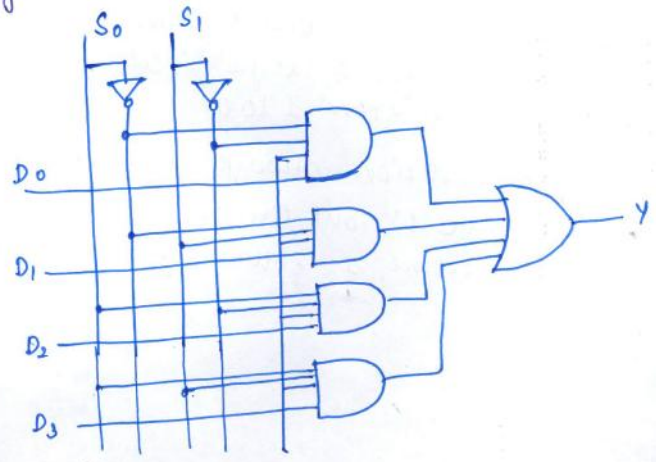

Logic symbol - 1
Function table - 1
Exp - 1
Logic diagram - 2.

function table:-

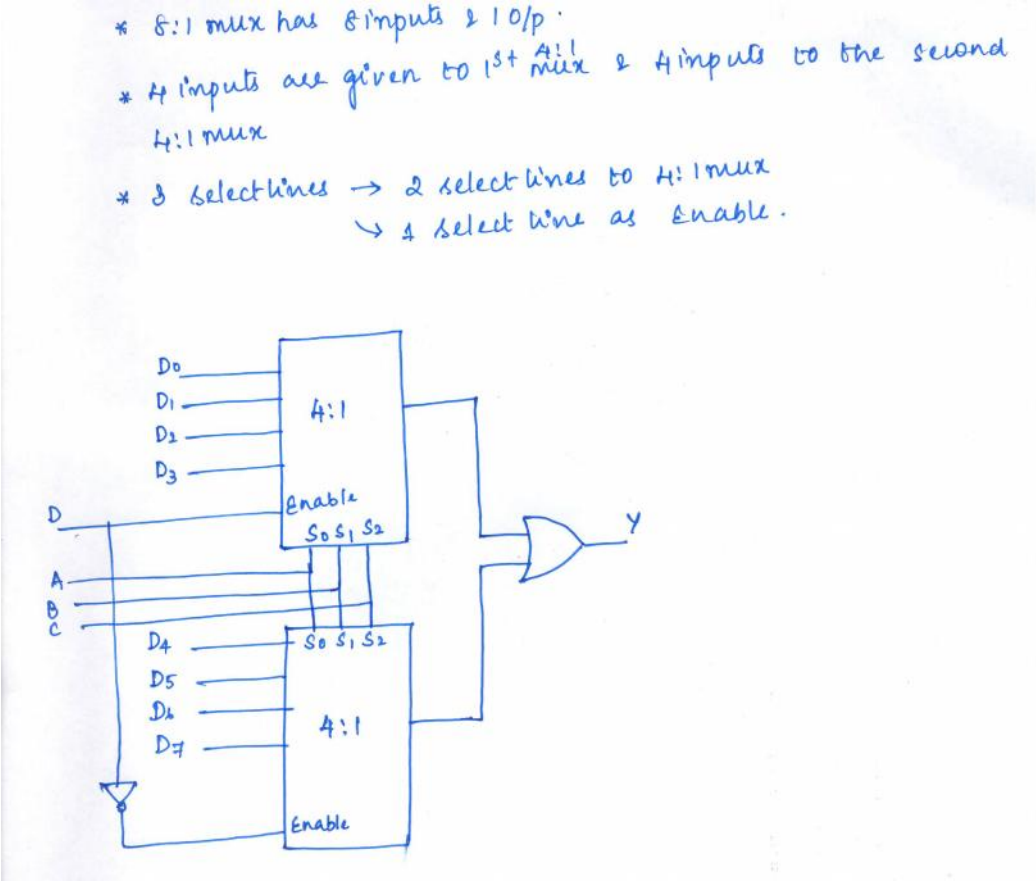
| E | $S_0$ | $S_1$ | Y |
|---|---|---|---|
| 1 | 0 | 0 | $D_0$ |
| 1 | 0 | 1 | $D_1$ |
| 1 | 1 | 0 | $D_2$ |
| 1 | 1 | 1 | $D_3$ |
| 0 | x | x | 0 |

$Y = E\bar{S_0}\bar{S_1}D_0 + E\bar{S_0}S_1D_1 + ES_0\bar{S_1}D_2 + ES_0S_1D_3$

Logic diagram:-



| | | | | |
|---|---|---|---|---|
| 4b. | Design 8:1 MUX using only 4:1 MUX | [5] | CO3 | L3 |




**OR**

| | | | | |
|---|---|---|---|---|
| **5 a.** | Explain 1 to 4 DEMUX with the following<br>• Block diagram<br>• Functional table<br>• Circuit diagram<br><br>Scheme:<br>• Block diagram -1 mark<br>• Functional table- 2 mark<br>• Circuit diagram – 1 mark | [4] | CO3 | L3 |

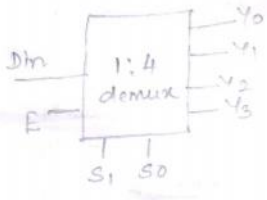
1:4 Demultiplexer:- No of i/ps = 1 ; No of outputs = 4.

output paths = 4

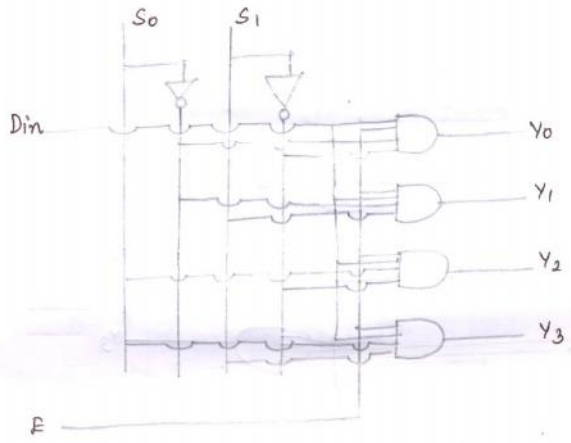
∴ Select lines = 2.

Function table:-

| E | $S_0$ | $S_1$ | output $Y_0$ | $Y_1$ | $Y_2$ | $Y_3$ |
|---|---|---|---|---|---|---|
| 0 | X | X | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | Din | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 | Din | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 | Din | 0 |
| 1 | 1 | 1 | 0 | 0 | 0 | Din |




$Y_0 = E\bar{S}_0\bar{S}_1 Din$ $Y_1 = ES_0\bar{S}_1 Din$ $Y_2 = ES_0\bar{S}_1 Din$ $Y_3 = E S_0 S_1$




| | | | | | |
|---|---|---|---|---|---|
| **5b.** | Implement a full adder using 3 to 8 decoder | [6] | CO3 | L3 |
| | **PART D** | | CO3 | L3 |
| **6a.** | Design a 4 bit priority encoder using basic gates. | [6] | | |

Scheme:
Truth table 2 marks
Expression – 2 marks
Logic diagram – 2 mark

| Inputs | | | | Outputs | |
|---|---|---|---|---|---|
| S | $X_1$ | $X_2$ | $X_3$ | A | B |
| 0 | x | x | x | 0 | 0 |
| 1 | 1 | x | x | 0 | 1 |
| 1 | 0 | 1 | x | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 | 0 |

K map:

A

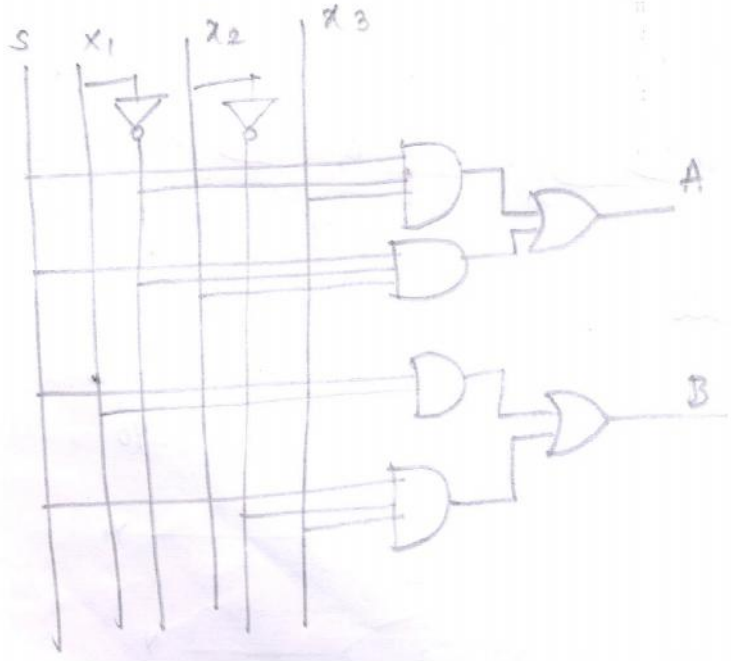
| $SX_1$ \ $X_2X_3$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 | 0 |
| 10 | 0 | 1 | 1 | 1 |

$A = S\bar{X}_1X_3 + S\bar{X}_1X_2$

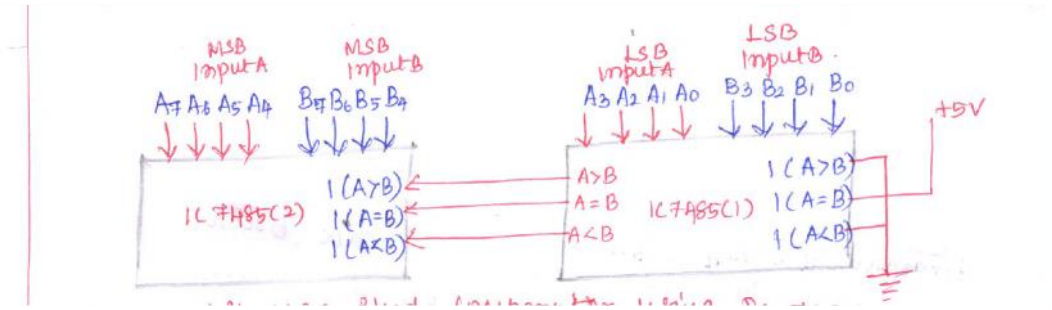
B

| $SX_1$ \ $X_2X_3$ | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 0 | 0 | 0 | 0 |
| 01 | 0 | 0 | 0 | 0 |
| 11 | 1 | 1 | 1 | 1 |
| 10 | 0 | 1 | 0 | 0 |

$B = SX_1 + S\bar{X}_2X_3$



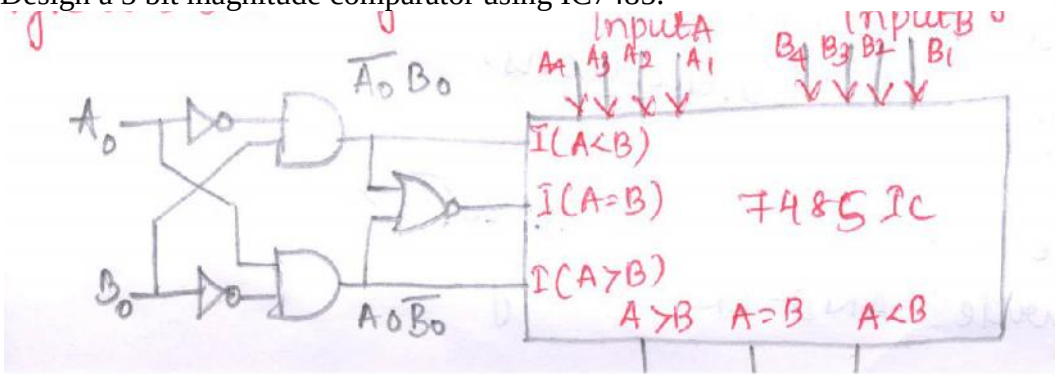

**6b.** Design a 8 bit comparator using two 7485 ICs [4] CO3 L3




**OR**

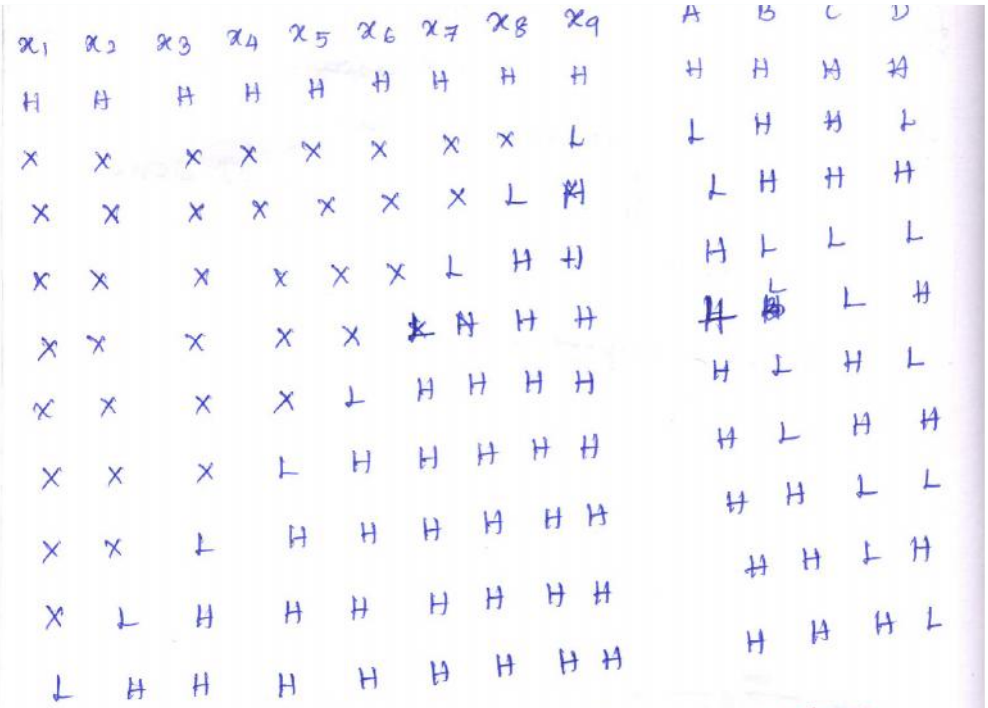
.

**7a.** Design a 5 bit magnitude comparator using IC7485. [5] CO3 L3



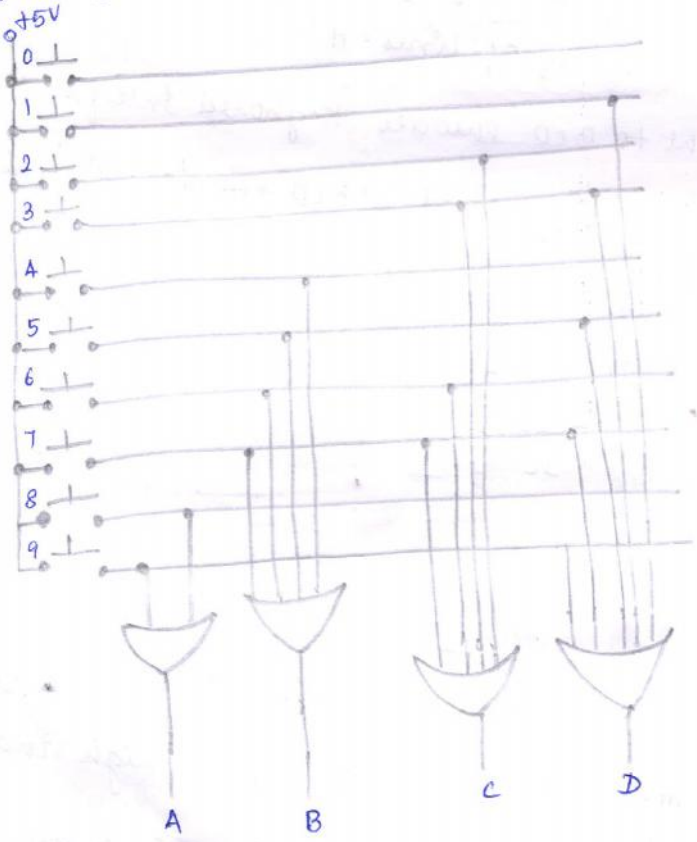

**b.** With a neat diagram explain decimal to BCD encoder. [5] CO3 L3

Scheme:
Truth table – 2.5 marks
Logic diagram- 2.5 marks



Ten Keypad interface for Decimal to BCD encoder:-

Priority is given to highest decimal.



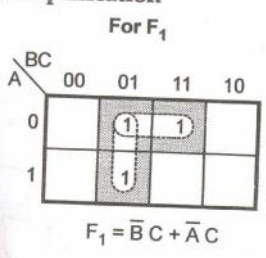

---

**PART E**

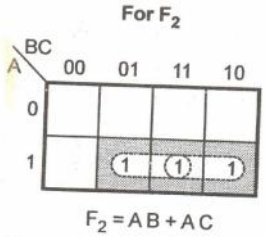
| | | | | |
|---|---|---|---|---|
| 8 | A combinational circuit is defined by the following functions F1= Σm(1,3,5) F2= Σm(5,6,7).implement the circuit using 3x3x2 PLA Kmap – 4 | [10] | CO3 | L3 |

PLA table -3
Fuse map- 3

**Solution : K-map simplification**



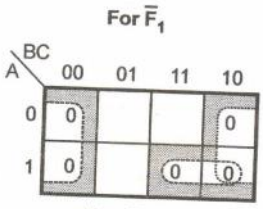

$F_1 = \overline{B}C + \overline{A}C$



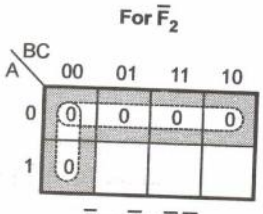

$F_2 = AB + AC$

**Fig. 6.8.11**

To implement functions $F_1$ and $F_2$ we require 3×4×2 PLA and we have to implement them using 3×3×2 PLA. There we have to examine product terms by grouping 0s instead of 1. That is product terms for complement of a function.




$\overline{F}_1 = \overline{C} + AB$




$\overline{F}_2 = \overline{A} + \overline{B}\,\overline{C}$

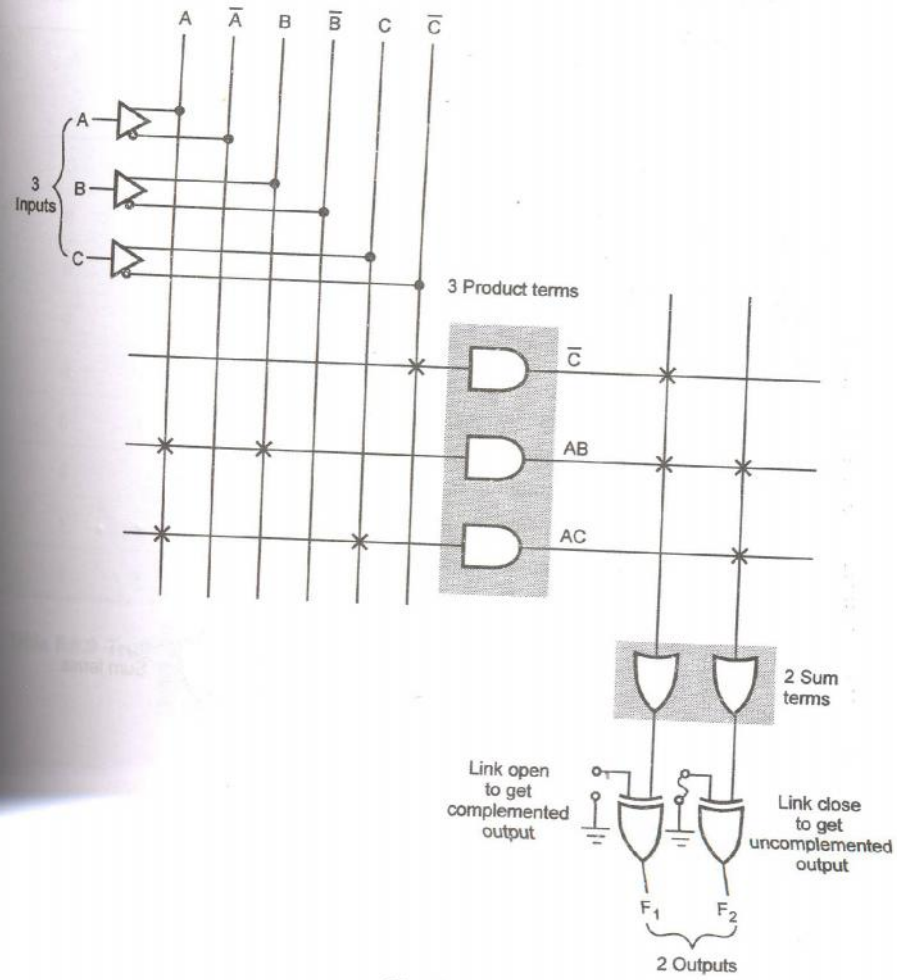
**Fig. 6.8.12**

PLA program table

| Product terms | Inputs | | | Outputs | |
|---|---|---|---|---|---|
| | A | B | C | $F_1$ | $F_2$ |
| $\overline{C}$ | - | - | 0 | 1 | - |
| AB | 1 | 1 | - | 1 | 1 |
| AC | 1 | - | 1 | - | 1 |
| | | | | C | T |

Table 6.8.8

Implementation



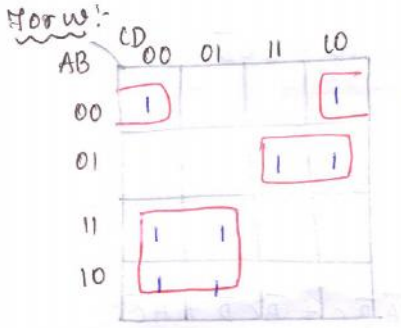

OR

| | | | | |
|---|---|---|---|---|
| **9** | Implement the following functions using PAL<br>F1(a,b,c,d)= Σm(0,2,6,7,8,9,12,13)<br>F2(a,b,c,d)= Σm(0,2,6,7,8,9,12,13,14)<br>F3(a,b,c,d)= Σm(2,3,8,9,10,12,13)<br>F4(a,b,c,d)= Σm(1,3,4,6,9,12,14)<br><br>Scheme:<br>Kmap – 4<br>PAL table -3<br>Fuse map- 3 | [10] | CO3 | L3 |

Sol:-

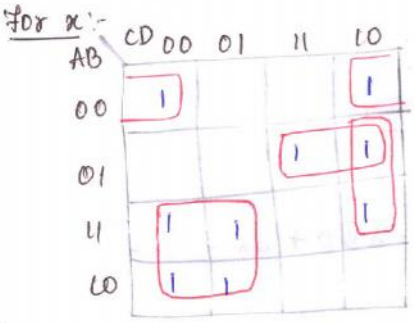
K-map:-

For w:-



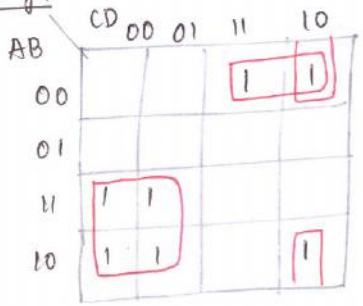

$w = \bar{A}\bar{B}\bar{D} + \bar{A}BC + A\bar{C}$

For x:-



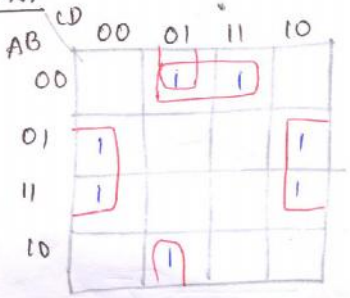

$x = \underbrace{\bar{A}\bar{B}\bar{D} + \bar{A}BC + A\bar{C}}_{w} + BC\bar{D}$

For y:-




$Y = \bar{A}\bar{B}C + \bar{B}C\bar{D} + A\bar{C}$

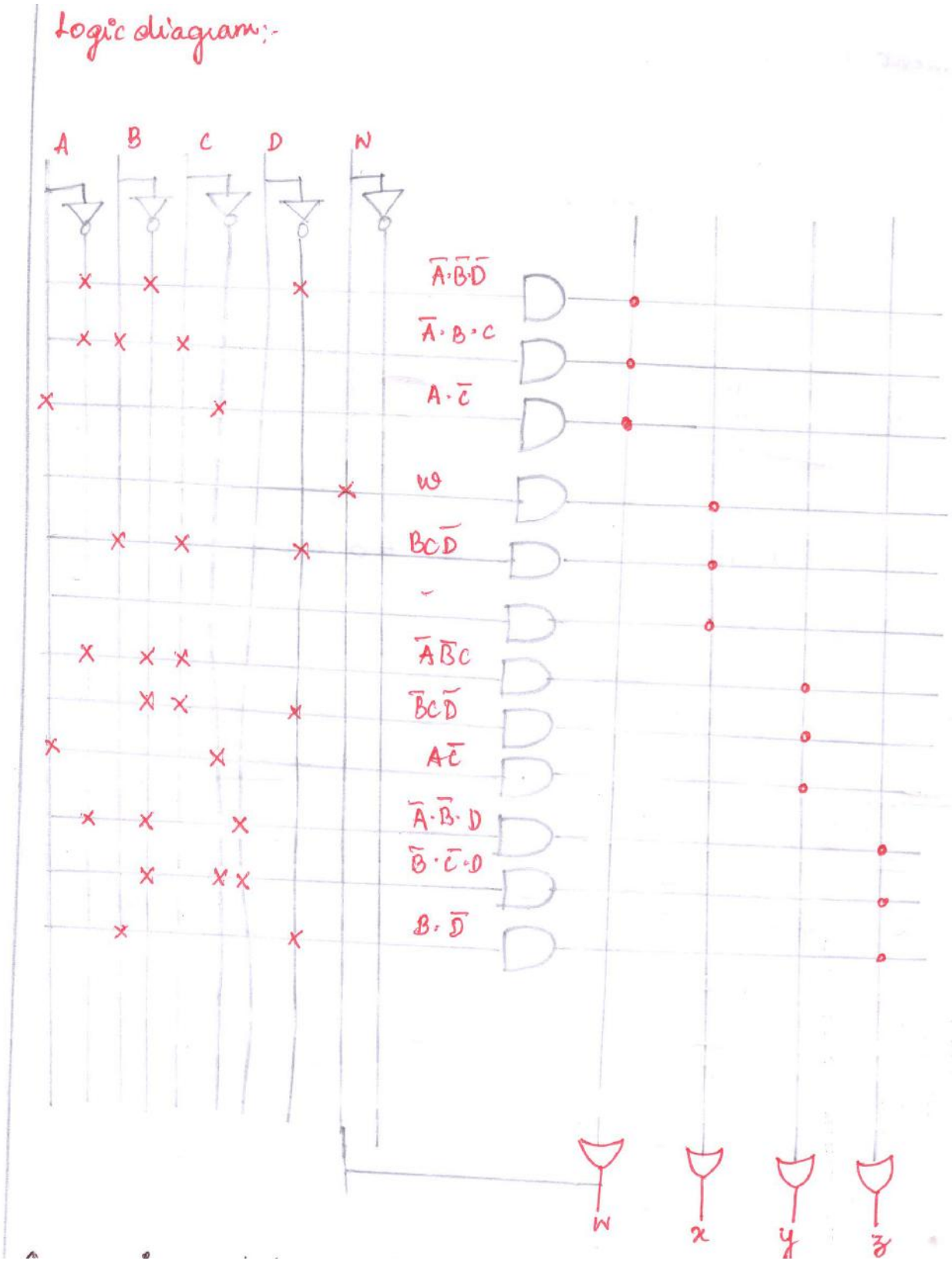
For z:-




$z = \bar{A}\bar{B}D + \bar{B}\bar{C}D + B\bar{D}$

| Product term | AND inputs | | | | O/ps | |
|---|---|---|---|---|---|---|
| | A | B | C | D | W | |
| 1) | 0 | 0 | – | 0 | – | $W = \bar{A}\bar{B}\bar{D} + \bar{A}BC + A\bar{C}$ |
| 2) | 0 | 1 | 1 | – | – | |
| 3) | 0 | – | 0 | – | – | |
| 4) | – | – | – | – | 1 | |
| 5) | – | 1 | 1 | 0 | – | $x = w + BC\bar{D}$ |
| 6) | – | – | – | – | – | |
| 7) | 0 | 0 | 1 | – | – | |
| 8) | – | 0 | 1 | 0 | – | $y = \bar{A}\bar{B}C + \bar{B}C\bar{D} + A\bar{C}$ |
| 9) | 1 | – | 0 | – | – | |
| 10) | 0 | 0 | – | 1 | – | $z = \bar{A}\bar{B}D + \bar{B}\bar{C}D + B\bar{D}$ |
| 11) | – | 0 | 0 | 1 | – | |
| 12) | – | 1 | – | 0 | – | |

Logic diagram:-




----------ALL THE BEST----------