2711

Code : 20EC11T

Register Number □□□□□□□□□□

I Semester Diploma Examination, March/April-2022

DIGITAL ELECTRONICS

Time : 3 Hours] **[Max. Marks : 100**

Instructions : (1) Answer **one** full question from each section.
(2) **One** full question carries **20** marks.

SECTION – I

1. (a) (i) Compare analog and digital signals. **5**

(ii) $(10101)_2 - (10111)_2$, calculate using 2's complement method. **5**

(b) Perform the following operations : **10**

(i) Convert the binary number $(11110101011.0011)_2$ to octal.

(ii) Convert the hexadecimal number $(152A.25)_{16}$ to decimal.

(iii) Convert Gray Code 100111 into Binary number.

2. (a) (i) Give the BCD equivalent for the decimal number 589. **10**

(ii) Give the decimal equivalent for the Excess-3 number 010110001001.

(iii) Find the BCD addition between two BCD numbers 0101 and 0110.

(b) (i) State and explain De Morgan's theorem. **5**

(ii) Simplify the logical expression using Boolean laws (A + B) (A + C). **5**

SECTION – II

3. (a) Define logic gate. Write Symbol, Truth-table and Logic Expression of OR and NAND gates. **10**

(b) Realise AND, NOT, OR, NAND gates using only NOR gates. **10**

4. (a) (i) Find the SOP minterm expression for canonical form $f = \Sigma_1 (m_1, m_2, m_3, m_5)$ and write the truth-table. **5**

(ii) For the following POS expression, write the truth-table : **5**

$$Y = \bar{A}\bar{B}C + \bar{A}BC + AB\bar{C}$$

(b) (i) Minimize the following Boolean function using K-map : 5

$$Y = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$$

(ii) Minimize the following Boolean function using K-map :

$F(A, B, C, D) = \Sigma_1 m\,(0, 1, 3, 5, 7, 8, 9, 11, 13, 15)$ 5

SECTION – III

5. (a) (i) Explain half-adder with truth-table. 5

(ii) Explain two bit magnitude comparator with truth-table. 5

(b) Explain working of a full-subtractor with logic diagram and truth-table. 10

6. (a) (i) Give the comparison between serial and parallel adder. 5

(ii) Mention any 5 applications of comparators. 5

(b) Explain working of serial binary adder. 10

SECTION – IV

7. (a) Explain the working of 4 : 1 multiplexer with logic circuit, symbol and truth-table. 10

(b) (i) List the applications of demultiplexer. 5

(ii) Explain the operation of 1 : 2 demultiplexer. 5

8. (a) Implement AND and OR Gates using 2 : 1 Multiplexer. 10

(b) Describe what input conditions would be required to generate the code for the following decimal number in decimal to BCD encoder : 10

(i) 2

(ii) 4

(iii) 8

(iv) 3

(v) 7

SECTION – V

9. (a) Build a BCD-to-Seven segment decoder circuit with truth-table. 10

(b) (i) Explain the working of 4-line to 2-line Encoder. 5

(ii) Mention any 5 applications of Decoder. 5

10. (a) Show what would be the output condition for a two input TTL NAND gate for all the input conditions. 10

(b) (i) List the applications of Integrated Circuits. 5

(ii) Define the following parameters of Logic families : 5

(1) Speed (2) Fan in (3) Fan out (4) Power dissipation (5) Noise Margin

I Semester Diploma Examination March/ April - 2022
DIGITAL ELECTRONICS CODE: 20EC11T
SCHEME OF VALUATION

SECTION -I

1 (a) (i) Any 5 comparison, each comparison 1 mark, total 1x5 = 5 marks.
(ii) Taking 2's complement of subtrahend (2 marks) + Addition (2 marks) +
Final result (1 mark) = 5 marks.

1 (b) (i) Octal number = 3 marks.
(ii) Decimal number = 3 marks.
(iii) Binary number = 4 marks.

2 (a) (i) BCD equivalent = 3 marks.
(ii) Decimal equivalent = 3 marks.
(iii) Addition (2 marks) + Valid BCD number (2 marks) = 4 marks.

2 (b) (i) Statement of De Morgan's theorem (2 marks) + Verification (3 marks) = 5 marks.
[Any one theorem]
(ii) Simplification using Boolean laws = 5 marks.

SECTION -II

3 (a) Definition of logic gate = 2 marks.
OR Gate: Symbol (1 mark) + Truth-table (2 marks) + Logic expression (1 mark) = 4 marks.
NAND Gate: Symbol (1 mark) + Truth-table (2 marks) + Logic expression (1 mark) = 4 marks.

3 (b) Realization of AND gate using NOR gate:
Logic Diagram (1.5 marks) + truth-table (1 mark) = 2.5 marks.
Realization of NOT gate using NOR gate:
Logic Diagram (1.5 marks) + truth-table (1 mark) = 2.5 marks.
Realization of OR gate using NOR gate:
Logic Diagram (1.5 marks) + truth-table (1mark) = 2.5 marks.
Realization of NAND gate using NOR gate:
Logic Diagram (1.5 marks) + truth-table (1 mark) = 2.5 marks.

4(a) (i) Expression (2 marks) + Truth-table (3 marks) = 5 marks.
(ii) SOP Truth- table (or) POS Truth-table = 5marks.

4(b) (i) K map (3 marks) + Simplified Expression (2 marks) = 5 marks.
(ii) K map (3 marks) + Simplified Expression (2 marks) = 5 marks.

SECTION -III

5 (a) (i) Half adder: Symbol (1 mark) + Truth-table (1 mark) + Boolean expression (1 mark) + Logic Circuit (2 marks) = 5 marks.
(ii) Two-bit magnitude comparator: Symbol (2 marks) + Truth-table (2 marks) + Explanation (1 mark) = 5 marks.
5(b) Full subtractor: Symbol (1 mark) + Truth-table (3 marks) + Logic expression (2 marks) + K map (2 marks) + Logic Circuit (2 marks) = 10 marks.

6 (a) (i) Any 5 comparison, each comparison 1 mark, total 1x5 = 5 marks.
(ii) Any 5 applications, each application 1 mark, total 1x5 = 5 marks.

6 (b) Serial binary adder: Block diagram (5 marks) + Explanation (5 marks) = 10 marks.

SECTION -IV

7 (a) 4:1 multiplexer: Logic Symbol (2 marks) + Truth-table (2 marks) + Boolean expression (4 marks) + Logic Circuit (2 marks) = 10 marks.

7(b) (i) Any 5 applications, each application 1 mark, total 1x5 = 5 marks.
(ii) 1: 2 Demultiplexer: Logic Symbol (1 mark) + Truth-table (1 mark) + Boolean expression (1 mark) + Logic Circuit (2 marks) = 5 marks.

8 (a) AND gate: Logic Symbol (2.5 marks) + Truth-table (2.5 marks) = 5 marks.
OR gate: Logic Symbol (2.5 marks) + Truth-table (2.5 marks) = 5 marks.

8 (b) Each decimal number to BCD code calculation 2 marks, 2x5=10 marks.

SECTION -V

9 (a) BCD to seven segment decoder:
Diagram (4 marks) + Truth-table (4 marks) + Explanation (2 marks) = 10 marks.

9(b) (i) 4-line to 2-line encoder: Logic Symbol (1 mark) + Truth-table (2 marks) + Boolean expression (1 mark) + Logic Circuit (1 mark) = 5 marks.
(ii) Any 5 applications, each application 1 mark, total 1x5 = 5 marks.

10(a) Diagram (4 marks) + Explanation (4 marks) + Truth-table (2 marks) = 10 marks.

10(b) (i) Any 5 applications, each application 1 mark, total 1x5 = 5 marks.
(ii) Definition of each parameter 1 mark, total 1x5 = 5 marks.

I Semester Diploma Examination March/April - 2022
DIGITAL ELECTRONICS CODE: 20EC11T

SECTION –I

1(a) (i) Compare analog and digital signals. **5 Marks**

Ans.

Sl. No	Analog	Digital
1	An analog signal is a continuous signal that represents physical measurements.	Digital signals are time separated signals which are generated using digital modulation.
2	It is denoted by sine waves.	It is denoted by square waves
3	It uses a continuous range of values that help you to represent information.	Digital signal uses discrete 0 and 1 to represent information.
4	Temperature sensors, FM radio signals, Photocells, Light sensor, Resistive touch screen are examples of Analog signals.	Computers, CDs, DVDs are some examples of Digital signal.
5	The analog signal bandwidth is low.	The digital signal bandwidth is high.

1(a)(ii) $(10101)_2$-$(10111)_2$,Calculate using 2's complement method. **5 Marks**

Ans.

Here, 10101 – Minuend
10111 – Subtrahend

Step 1: We take 2's complement of subtrahend

$10111 \longrightarrow 01000 + 1 = 01001$

Step 2: Add 2's complement of subtrahend with minuend

$$\begin{array}{r} 10101 \\ \underline{+\ 01001} \\ =11110 \end{array}$$

Step 3: In the above result, we didn't get the carry bit. So calculate the 2's complement of the result, i.e., So,

$11110 \longrightarrow 00001 + 1 = 00010$

It is the negative number and the final result value.

1(b) Perform the following operations **3+3+4=10 Marks**

(i) Convert the binary number $(11110101011.0011)_2$ to octal.

Ans. Firstly, we make pairs of three bits on both sides of the binary point.

11 110 101 011 . 001 1

On the right side of the binary point, the last pair has only one bit. To make it a complete pair of three bits, we added two zeros on the extreme side. In the same way we add one zero on left extreme side.

011 110 101 011 . 001 100

Then, we wrote the octal digits, which correspond to each pair.

011	110	101	011	.	001	100
3	6	5	3	.	1	4

Hence, $(11110101011.0011)_2 = (3653.14)_8$

(ii) Convert the hexadecimal number $(152A.25)_{16}$ to decimal.

Ans. $(152A.25)_{16} = (1x16^3)+ (5x16^2)+ (2x16^1)+ (Ax16^0)+ (2x16^{-1})+ (5x16^{-2})$

$(152A.25)_{16} = (1x4096)+ (5x256)+ (2x16)+ (Ax1)+ (2x16^{-1})+ (5x16^{-2})$

$(152A.25)_{16} = 4096+1280+32+10+(2x1/16)+ (5x1/256)$

$(152A.25)_{16} = 5418+0.125+0.125$

$(152A.25)_{16} = 5418.14453125$

So, the decimal number of the hexadecimal number 152A.25 is 5418.14453125.

(iii) Convert Gray code 100111 into Binary number.

Ans. For a 5-bit operation, B_4 B_3 B_2 B_1 B_0 represents a 5-bit binary code and G_4 G_3 G_2 G_1 G_0 represents a 5-bit gray code.

B_5=G_5=1=1

B_4= $B_5 \oplus G_4$=1⊕0=1

B_3= $B_4 \oplus G_3$=1⊕0=1

B_2= $B_3 \oplus G_2$=1⊕1=0

B_1= $B_2 \oplus G_1$=0⊕1=1

B_0= $B_1 \oplus G_0$=1⊕1=0

So, Binary number will be 111010.

2(a) (i) Give the BCD equivalent for the decimal number 589. **3+3+4=10 Marks**

Ans. The decimal number is 589.

BCD code is 0101 1000 1001

Hence, $(589)_{10}= (010110001001)_{BCD}$

(ii) Give the decimal equivalent for the Excess-3 number 010110001001.

Ans.

Step 1: Frame each four digit into a group

0101 1000 1001

Step 2: Get decimal equivalent for each group

0101	1000	1001
5	8	9

Step 3: Subtract 3 from each digit

$$\begin{array}{cccc} & 5 & 8 & 9 \\ - & \underline{3} & \underline{3} & \underline{3} \\ & 2 & 5 & 6 \end{array}$$

Hence, (010110001001) = 256

(iii) Find the BCD addition between two BCD numbers 0101 and 0110.

Ans.

$$\begin{array}{rrcl} & 0101 & & \\ & \underline{+0110} & & \\ & 1011 & \longrightarrow & \text{Invalid BCD number} \\ \underline{} & \underline{+0110} & \longrightarrow & \text{Add 6} \\ 0001 & 0001 & \longrightarrow & \text{Valid BCD number} \end{array}$$

2(b) (i) State and Explain De Morgan’s theorems **5 Marks**

Ans. De Morgan’s First theorem states that, “The complement of product of variables is equal to the sum of the complements of individual variables”.

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

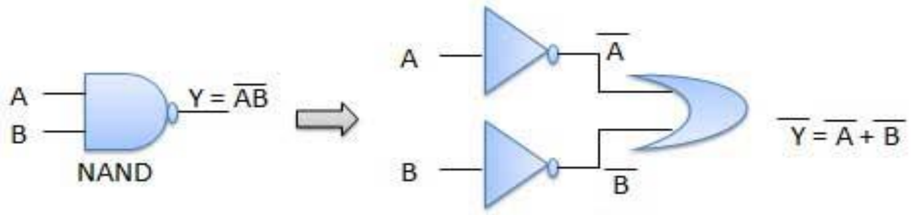
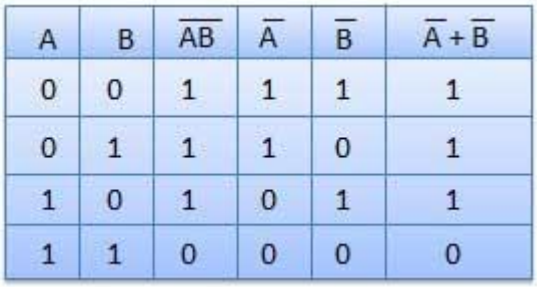


Table showing verification of the De Morgan's first theorem –



A	B	$\overline{AB}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

De Morgan’s Second theorem states that, “The complement of sum of variables is equal to the product of the complements of individual variables”.

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

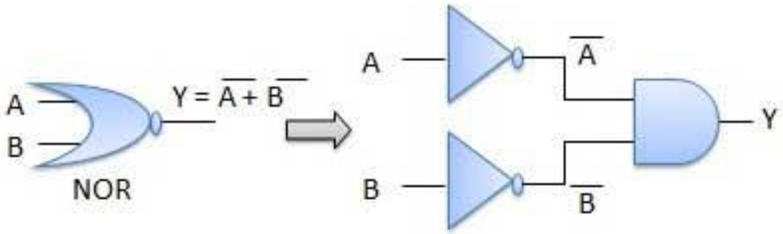
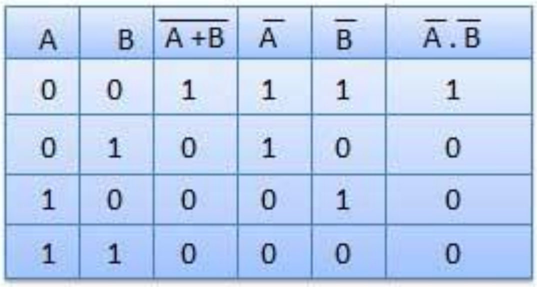


Table showing verification of the De Morgan's second theorem –



A	B	$\overline{A+B}$	$\overline{A}$	$\overline{B}$	$\overline{A}.\overline{B}$
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	0

2 (b) (ii) Simplify the logical expression using Boolean laws (A+B)(A+C) 5 Marks

Ans.

(A+B)(A+C) = AA + AC + AB + BC

= A + AC + AB + BC {A.A = A, Identity AND law}

= A (1 + C) + AB + BC

= A + AB + BC {1 + C = 1, Identity OR law}

= A (1 + B) + BC

= A + BC {{1 + B = 1, Identity OR law}

SECTION –II

3 (a) Define logic gate. Write symbol, Truth-table and Logic Expression of OR and NAND gates. **10 Marks**

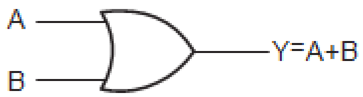
Ans.

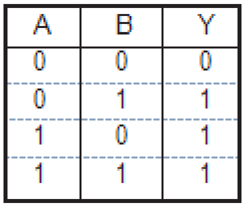
Logic gate:

The logic gate is the most basic building block of any digital system, including computers. Each one of the basic logic gates is a piece of hardware or an electronic circuit that can be used to implement some basic logic expression.

OR gate:

- An OR gate performs an ORing operation on two or more than two logic variables.
- The OR operation on two independent logic A and *B* is written as Y = A + B and reads as *Y* equals *A* OR *B*. An OR gate is a logic circuit with two or more inputs and one output.
- The output of an OR gate is LOW only when all of its inputs are LOW. For all other possible input combinations, the output is HIGH.
- The operation of a two-input OR gate is explained by the logic expression $Y = A + B$. Figure shows the circuit symbol and the truth table of a two-input OR gate.



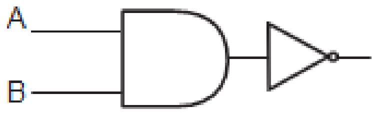


A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

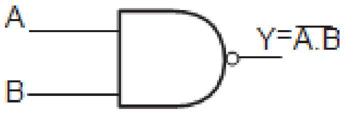
NAND gate:

- NAND stands for NOT AND.
- An AND gate followed by a NOT circuit makes it a NAND gate (fig.a)
- (fig.b) shows the circuit symbol of a two-input NAND gate.
- The truth table of a NAND gate is obtained from the truth table of an AND gate by complementing the output entries (Fig. c).
- The output of a NAND gate is a logic ‘0’ when all its inputs are a logic ‘1’. For all other input combinations, the output is a logic ‘1’. NAND gate operation is logically expressed as

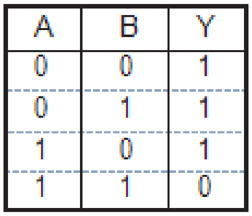
$$Y = \overline{A.B}$$



(a)



(b)



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

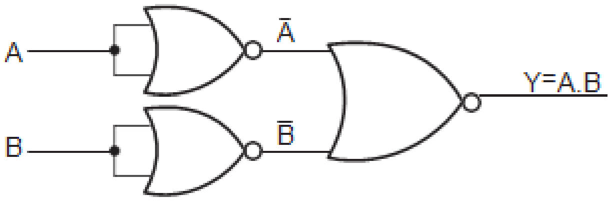
(c)

3 (b) Realise AND, NOT, OR and NAND gates using only NOR gates **10 Marks**

Ans.

Realisation of AND function using only NOR gates

- Connect two NOT using NORs at the inputs of a NOR to get AND logic.



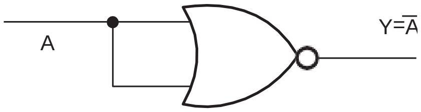
- From De Morgan’s theorems: (A+B)’ = A’B’

(A’+B’)’= A’’B’’ = AB

Truth- Table:

Input		Output
A	**B**	**Y=A.B**
0	0	0
0	1	0
1	0	0
1	1	1

Realisation of NOT function using only NOR gates:



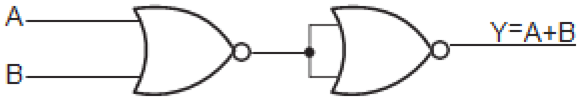
- Its output is Y = A'

Truth- Table:

Input	**Output**
A	**Y=A'**
0	1
1	0

Realisation of OR function using only NOR gates:

- Connect a NOT using NORs at the output of the NOR to invert it and get OR logic.



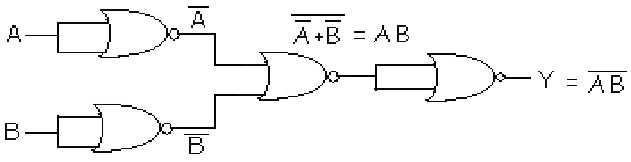
- Its output is Y = ((A+B)')'
  Y = (A+B)

Truth- Table:

Input		**Output**
A	**B**	**Y=A+B**
0	0	0
0	1	1
1	0	1
1	1	1

Realisation of NAND function using only NOR gates

LOGIC DIAGRAM



TRUTH TABLE

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

- Its output is Y = (A.B)'

https://mathswithme.in/

4 (a) (i) Find the SOP minterm expression for canonical form F=∑ (m1, m2, m3, m5) and write the truth-table. **5 Marks**

Ans.

By expanding the above summation we can get the following function.

F = m1 + m2 + m3 + m5

By substituting the minterms in the above equation we can get the below expression

F = A’B’C + A’BC’ + A’BC + AB’C

Input Variables			**Min terms**	**Output**
A	**B**	**C**	**m_i**	**F**
0	0	0	A’B’C’ = m_o	0
0	0	1	A’B’C = m_1	1
0	1	0	A’BC’ = m_2	1
0	1	1	A’BC = m_3	1
1	0	0	AB’C’ = m_4	0
1	0	1	AB’C= m_5	1
1	1	0	ABC’ = m_6	0
1	1	1	ABC = m_7	0

4 (a) (ii) For the below POS expression write the truth-table.

Y=A’B’C + A’BC + ABC’ **5 Marks**

Ans.

Note:- We write the Truth-table for the SOP expression Y= A’B’C + A’BC + ABC’

F(A,B,C)= ∑m(1,3,6)= m1 + m3 + m6

(OR)

We write the Truth-table for the POS expression Y = π A, B, C (0, 2, 4, 5, 7)

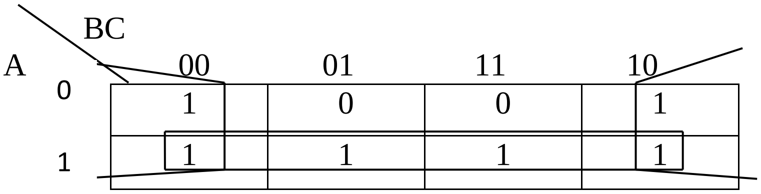
= (A + B + C) * (A + B’ + C) * (A’ + B + C) * (A’ + B + C’) * (A’ + B’ + C’)

Input Variables			**SOP terms (Minterms)**	**POS terms Max terms**	**Output**
A	**B**	**C**	**m_i**	**M_i**	**Y**
0	0	0	m0 = A’.B’.C’	M0 = (A + B +C)	0
0	0	1	m1 = A’.B’.C	M1 = (A + B +C’)	1
0	1	0	m2 = A’.B.C’	M2 = (A + B’ +C)	0
0	1	1	m3 = A’.B.C	M3 = (A + B’ +C’)	1
1	0	0	m4 = A.B’.C’	M4 = (A’ + B +C)	0
1	0	1	m5= A.B’.C	M5 = (A’ + B +C’)	0
1	1	0	m6 = A.B.C’	M6 = (A’ + B’ +C)	1
1	1	1	m7 = A.B.C	M7 = (A’ + B’+C’)	0

4 (b) (i) Minimize the following Boolean function using K – map
Y = A’B’C’ + A’BC’ + AB’C’ + AB’C + ABC’ + ABC **5 Marks**

Ans.

- Since the given Boolean expression has 3 variables, so we draw a 2 x 3 K Map.
- We fill the cells of K Map in accordance with the given Boolean function.
- Then, we form the groups in accordance with the above rules and find the simplified expression.



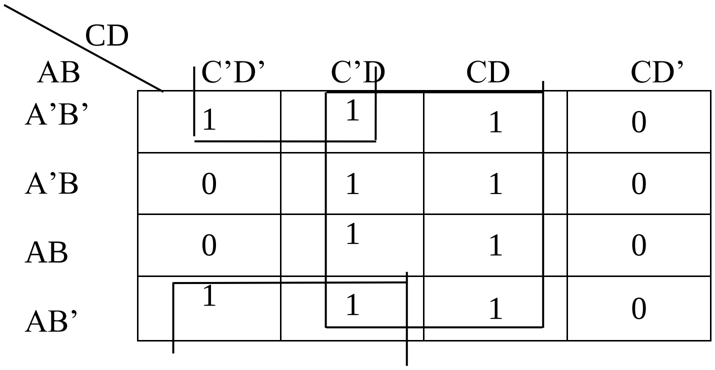
A \ BC	00	01	11	10
0	1	0	0	1
1	1	1	1	1

Simplified expression : Y = A + C’

4 (b) (ii) Minimize the following Boolean function using K – map
F (A,B,C,D) = ∑ m (0, 1, 3, 5, 7, 8, 9, 11, 13, 15) **5 Marks**

Ans.

- Since the given Boolean expression has 4 variables, so we draw a 4 x 4 K Map.
- We fill the cells of K Map in accordance with the given Boolean function.
- Then, we form the groups in accordance with the above rules and find the simplified expression.



AB \ CD	C’D’	C’D	CD	CD’
A’B’	1	1	1	0
A’B	0	1	1	0
AB	0	1	1	0
AB’	1	1	1	0

Thus, minimized Boolean expression is F (A, B, C, D) = B’C’ + D

SECTION –III

5 (a) (i) Explain half-adder with truth table. **5 Marks**

Ans.

A half-adder is an arithmetic circuit block that can be used to add two bits. Such a circuit thus has two inputs that represent the two bits to be added and two outputs, with one producing the SUM output and the other producing the CARRY. Figure shows the truth table of a half-adder, showing all possible input combinations and the corresponding outputs.

The Boolean expressions for the SUM and CARRY outputs are given by the equations

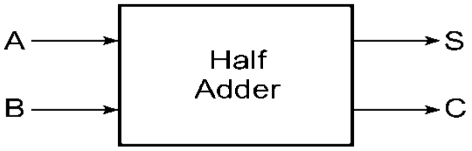
SUM S = AB’ + A’B

CARRY $C = A.B$

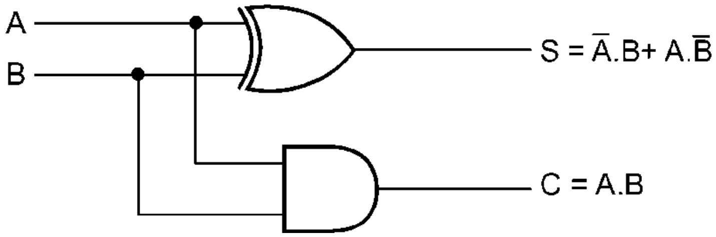
While the first one representing the SUM output is that of an EX-OR gate, the second one representing the CARRY output is that of an AND gate.

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Truth table of a half-adder.



Logic Symbol of Half-adder

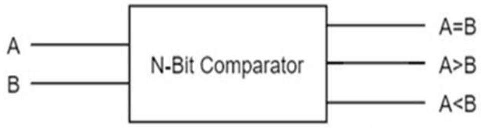


Logic implementation of a half-adder.

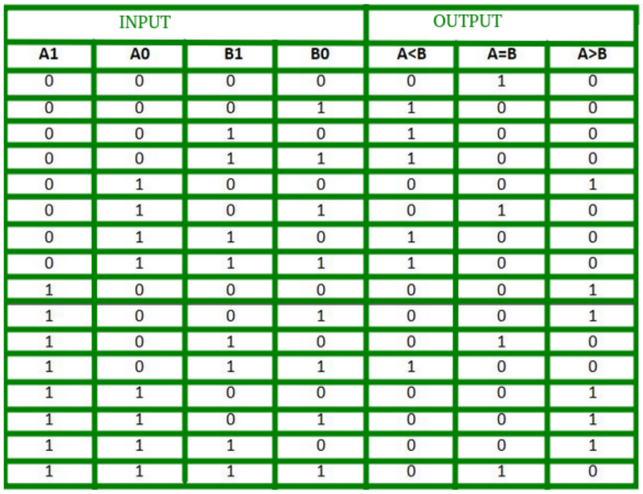
5 (a) (ii) Explain two bit magnitude comparator with truth table. **5 Marks**

Ans.

In digital system, comparison of two numbers is an arithmetic operation that determines if one number is greater than, equal to, or less than the other number. So comparator is used for this purpose. Magnitude comparator is a combinational circuit that compares two numbers, A and B, and determines their relative magnitudes (Fig.) The outcome of comparison is specified by three binary variables that indicate whether A>B, A=B, or A<B.



A comparator used to compare two binary numbers each of two bits is called a 2-bit magnitude comparator. It consists of four inputs and three outputs to generate less than, equal to and greater than between two binary numbers. The truth table for a 2-bit comparator is given below:



INPUT				OUTPUT		
A1	A0	B1	B0	A<B	A=B	A>B
0	0	0	0	0	1	0
0	0	0	1	1	0	0
0	0	1	0	1	0	0
0	0	1	1	1	0	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	1	0	0
0	1	1	1	1	0	0
1	0	0	0	0	0	1
1	0	0	1	0	0	1
1	0	1	0	0	1	0
1	0	1	1	1	0	0
1	1	0	0	0	0	1
1	1	0	1	0	0	1
1	1	1	0	0	0	1
1	1	1	1	0	1	0

Figure: Truth Table of 2-Bit Comparator

From the above truth table logical expressions for each output can be expressed as follows:

$A > B : A_1B_1' + A_0B_1'B_0' + A_1A_0B_0'$

$A = B : A_1'A_0'B_1'B_0' + A_1'A_0B_1'B_0 + A_1A_0B_1B_0 + A_1A_0'B_1B_0'$

$: A_1'B_1' (A_0'B_0' + A_0B_0) + A_1B_1 (A_0B_0 + A_0'B_0')$

$: (A_0B_0 + A_0'B_0') (A_1B_1 + A_1'B_1')$

$: (A_0 \text{ Ex-Nor } B_0) (A_1 \text{ Ex-Nor } B_1)$

$A < B : A_1'B_1 + A_0'B_1B_0 + A_1'A0'B0$

By using these Boolean expressions, we can implement a logic circuit for this comparator as given below :

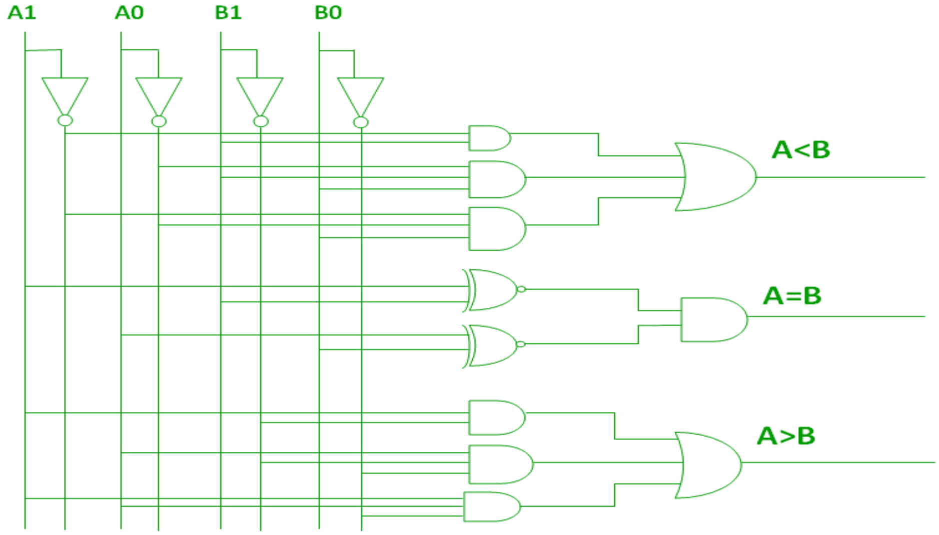


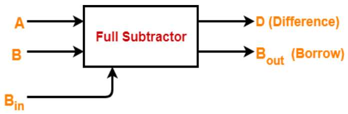
Figure-: Logic Circuit of 2-Bit Comparator

5 (b) Explain working of a full-subtractor with logic diagram and truth table. **10 Marks**

Ans.

Full Subtractor is a combinational logic circuit.

- It is used for the purpose of subtracting two single bit numbers.
- It also takes into consideration borrow of the lower significant stage.
- Thus, full subtractor has the ability to perform the subtraction of three bits.
- Full subtractor contains 3 inputs and 2 outputs (Difference and Borrow)



Designing a Full Subtractor- Full subtractor is designed in the following steps-

Step-01: Identify the input and output variables-

- Input variables = A, B, B_{in} (either 0 or 1)
- Output variables = D, B_{out} where D = Difference and B_{out} = Borrow

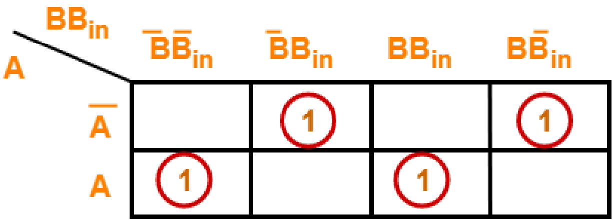
Step-02: Truth Table

Inputs			**Outputs**	
A	**B**	**B_{in}**	**B_{out} (Borrow)**	**D (Difference)**
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

https://mathswithme.in/

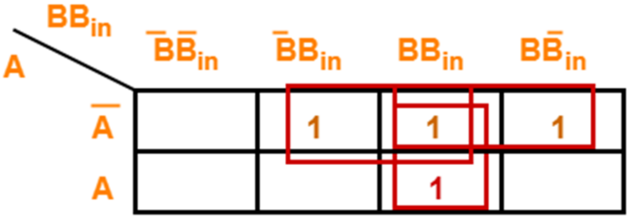
Step-03:

Draw K-maps using the above truth table and determine the simplified Boolean expressions-



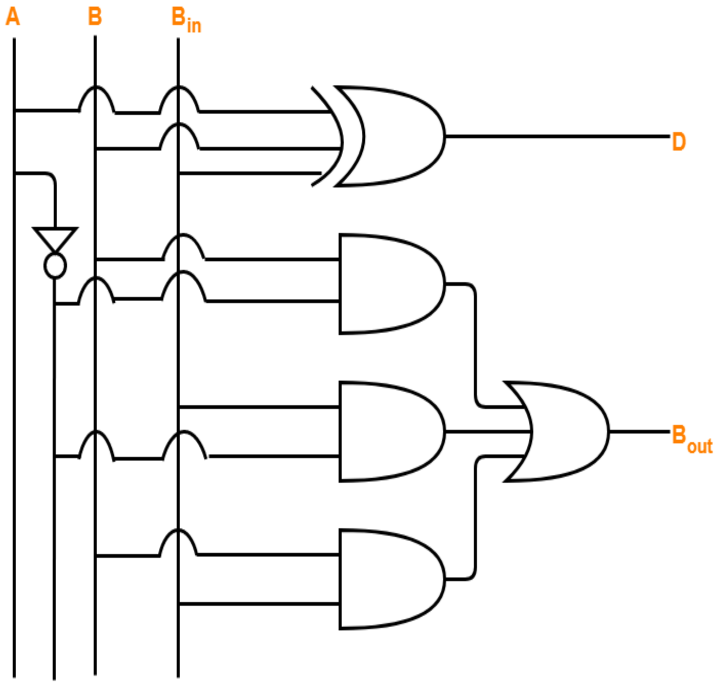
A \ BB_{in}	$\bar{B}\bar{B}_{in}$	$\bar{B}B_{in}$	BB_{in}	$B\bar{B}_{in}$
$\bar{A}$		1		1
A	1		1	

$$\boxed{D = A \oplus B \oplus B_{in}}$$



A \ BB_{in}	$\bar{B}\bar{B}_{in}$	$\bar{B}B_{in}$	BB_{in}	$B\bar{B}_{in}$
$\bar{A}$		1	1	1
A			1	

$$\boxed{B_{out} = \bar{A}B + (\bar{A} + B)B_{in}}$$



6 (a) (i) Give the comparison between serial and parallel adder. **5 Marks**

Ans

Sl. No	Serial Adder	Parallel Adder
1	It is used to add two binary binary numbers in serial form.	It is used to add two binary numbers in parallel form.
2	A serial adder uses shift registers.	A parallel adder uses registers with parallel loads.
3	It requires single full adder.	It requires multiple full adders.
4	Carry flip-flop is used in serial adder.	Ripple carry adder is used in parallel adder.
5	Serial adder is a sequential circuit.	Parallel adder is a combinational circuit.

6 (a) (ii) Mention any 5 applications of comparators. **5 Marks**

Ans.

- Comparators are used in central processing unit.
- They are used in microcontrollers.
- They are used in control applications.
- They are used as process controllers and for servo motor control.
- Used in password verification and biometric applications.

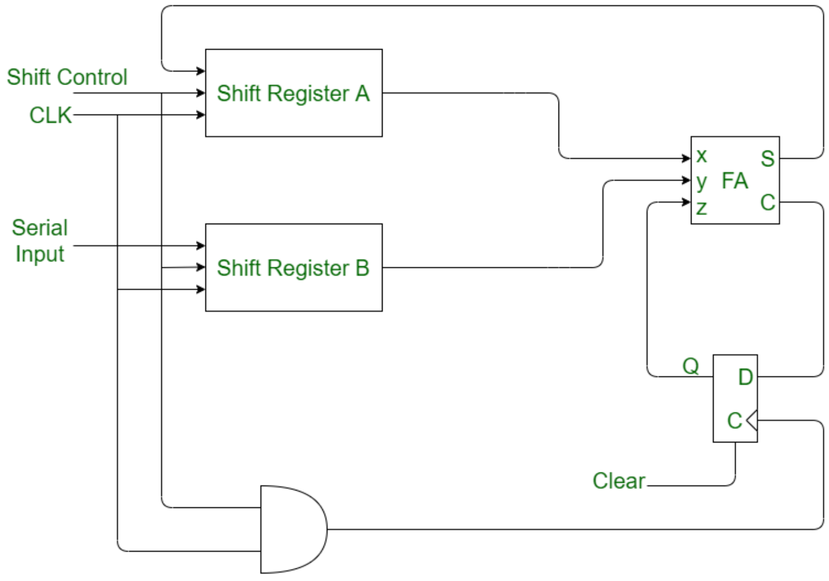
6 (b) Explain working of serial binary adder. **10 Marks**

Ans.

- **Serial binary adder** is a combinational logic circuit that performs the addition of two binary numbers in serial form. Serial binary adder performs bit by bit addition. Two shift registers are used to store the binary numbers that are to be added.

- A single full adder is used to add one pair of bits at a time along with the carry. The carry output from the full adder is applied to a D flip-flop. After that output is used as carry for next significant bits. The sum bit from the output of the full adder can be transferred into a third shift register.

Block diagram of Serial Binary Adder:



Block Diagram of Serial Binary Adder

- ➢ Shift Register is a group of flip flops used to store multiple bits of data. There are two shift registers used in the serial binary adder. In one shift register augend is stored and in other shift register addend is stored.
- ➢ Full adder is the combinational circuit which takes three inputs and gives two outputs as sum and carry. The circuit adds one pair at a time with the help of it.
- ➢ The carry output from the full adder is applied on the D flip-flop. Further, the output of D flip-flop is used as a carry input for the next pair of significant bits.

Working Process:

Following is the procedure of addition using serial binary adder:

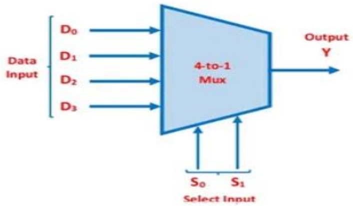
- **Step-1:**
  The two shift registers A and B are used to store the numbers to be added.
- **Step-2:**
  A single full adder is used to add one pair of bits at a time along with the carry.
- **Step-3:**
  The contents of the shift registers shift from left to right and their output starting from a and b are fed into a single full adder along with the output of the carry flip-flop upon application of each clock pulse.
- **Step-4:**
  The sum output of the full adder is fed to the most significant bit of the sum register.
- **Step-5:**
  The content of sum register is also shifted to right when clock pulse is applied.
- **Step-6:**
  After applying four clock pulse the addition of two registers (A & B) contents are stored in sum register.

SECTION –IV

7 (a) Explain the working of 4:1 multiplexer with logic circuit, symbol and truth table. 10 Marks

Ans.

A 4-to-1 multiplexer is a digital multiplexer that has four data inputs, two select lines, and one output. To implement a 4-to-1 multiplexer circuit we need 4 AND gates, an OR gate, and a 2 NOT gate. In a 4-to-1 multiplexer, four inputs D_0, D_1, D_2, and D_3, two data select lines that are S_0 and S_1 as 4-inputs represent $2^m = 2^2 = 2$ data control lines. One of these four inputs will be connected to the output based on the combination of the inputs at the selection lines.



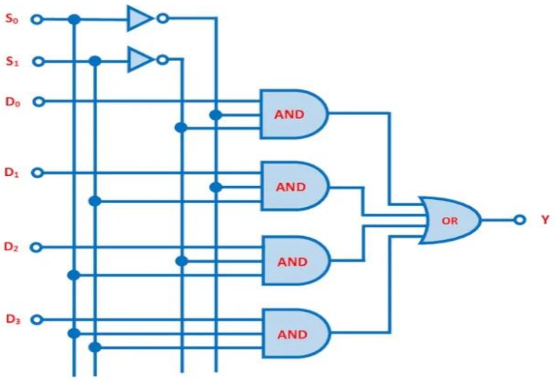
Logic Symbol

4 to 1 Multiplexer Working Principle:

- If both the select inputs S0=0, S1=0 then the topmost AND gate is enabled and all other AND gate is disabled. So, the data input D0 is selected and transmitted as output. Hence, we get the output Y = D0.
- If both the select inputs S0=1, S1=1 then the bottom-most AND gate is enabled and all other AND gate is disabled. So, the data input D3 is selected and transmitted as output. Hence, we get output Y=D3.

4 to 1 Multiplexer Truth Table:

Select Data Inputs		**Output**
S_1	**S_0**	**Y**
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3



A 4-to-1 Mux Logic Circuit

The final Boolean expression of this 4:1 multiplexer is given as

$$Y = D_0S_1'\ S_0' + D_1S_1'\ S_0 + D_2S_1\ S_0' + D_3S_1\ S_0$$

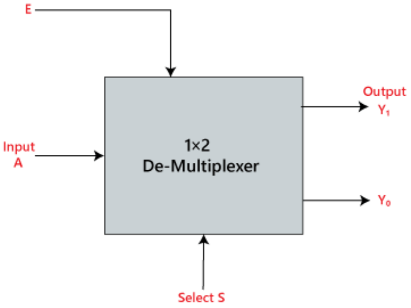
7 (b) (i) List the applications of demultiplexer. **5 Marks**

Ans.

- They are used in microprocessor or computer control systems.
- Synchronous data transmission systems.
- Boolean function implementation.
- Data acquisition systems.
- Combinational circuit design.
- Automatic test equipment systems.
- Security monitoring systems.

7 (b) (ii) Explain the operation of 1:2 demultiplexer. **5 Marks**

Ans. In the 1 to 2 De-multiplexer, there are only two outputs, i.e., Y_0, and Y_1, 1 selection lines, i.e., S_0, and single input, i.e., A. On the basis of the selection value, the input will be connected to one of the outputs.



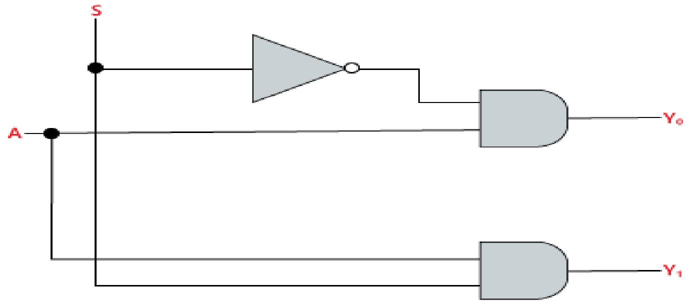
Block diagram of 1:2 demultiplexer

Truth Table:

INPUTS	Output	
S_0	Y_1	Y_0
0	0	A
1	A	0

The logical expression of the term Y is as follows: $Y_0=S_0'.A$ and $Y_1=S_0.A$

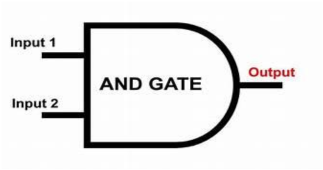
Logical circuit of the above expressions is given below:



8 (a) Implement AND and OR Gates using 2:1 Multiplexer **10 Marks**

Ans **Implementation of AND gate using 2 : 1 Mux**

AND GATE :



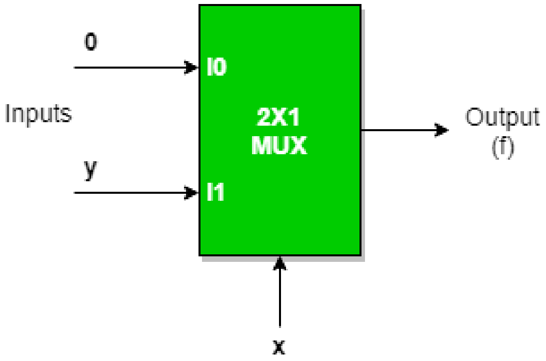
Here X and Y are inputs. Output of AND gate is X.Y i.e. in canonical SOP form it is F(m3).

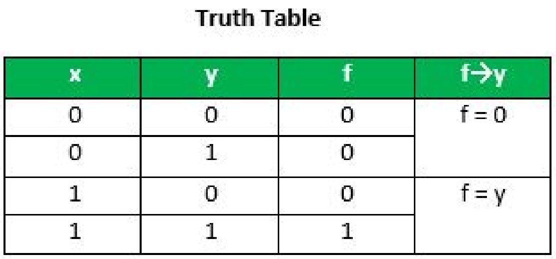
We take X as the select line.

Now the implementation table will be as below:

	I0	I1
Y’	0	1
Y	2	$\boxed{3}$
	0	Y

From the above implementation table AND gate can be implemented using 2x1 mux as shown below figure. The input I0 is connected to GND (0) and I1 is connected to input Y. The selected input S is connected to input X.



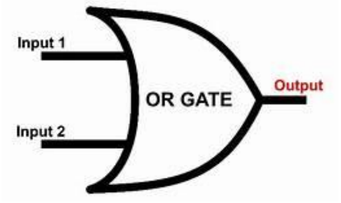


Truth Table

x	y	f	f→y
0	0	0	f = 0
0	1	0	
1	0	0	f = y
1	1	1	

Implementation of OR gate using 2 : 1 Mux.

OR GATE :



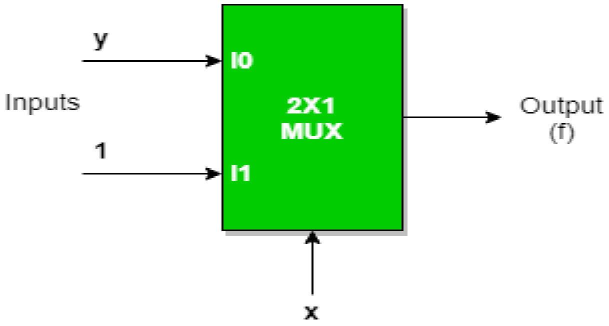
Here X and Y are inputs. Output of OR gate is X+Y i.e. in canonical SOP form it is F(m1,m2,m3).

We take X as the select line.

Now the implementation table will be as below:

	I0	I1
Y'	0	$\boxed{1}$
Y	$\boxed{2}$	$\boxed{3}$
	Y	1

From the above implementation table OR gate can be implemented using 2x1 mux as shown below figure. The input I0 is connected to Y and I1 is connected to input Vcc (1). The selected input S is connected to input X.



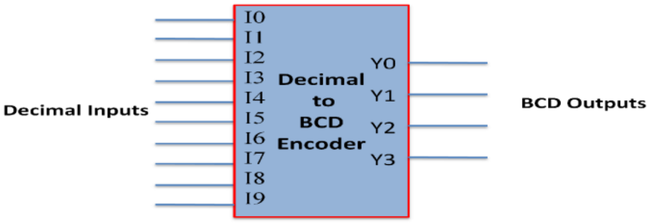
Truth Table

x	y	f	f→y
0	0	0	f = y
0	1	1	
1	0	1	f = 1
1	1	1	

8 (b) Describe what input conditions would be required to generate the code for the following decimal number in Decimal to BCD encoder:

(i) 2 (ii) 4 (iii) 8 (iv) 3 (v) 7 10 Marks

Ans A decimal to BCD (binary coded decimal) encoder is also known as 10-line to 4-line encoder. It accepts 10- inputs and produces a 4-bit output corresponding to the activated decimal input.



The truth table of Decimal to BCD encoder is shown in Table. There are ten inputs corresponding 10 decimal inputs (I0, I1, I2, I3, I4, I5, I6, I7, I8, I9) and 4 Outputs (Y0, Y1, Y2, Y3). Let us look at the table carefully. Note the encoder assumption that only one of the inputs is activated (logic 1) and other inputs are not activated (i.e. at logic 0).

TABLE: Truth – Table for Decimal to BCD Encoder

Decimal Inputs										BCD Outputs			
I0	I1	I2	I3	I4	I5	I6	I7	I8	I9	Y3	Y2	Y1	Y0
1	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0	0	0	0	1	0	1
0	0	0	0	0	0	1	0	0	0	0	1	1	0
0	0	0	0	0	0	0	1	0	0	0	1	1	1
0	0	0	0	0	0	0	0	1	0	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	0	0	1

In the truth table, the input variable I0 represents the least significant digit (LSD) and and I9 represents most significant digit (MSD). Similarly, in the outputs Y0 represents least significant bit (LSB) and Y3 is the most significant bit (MSB). The truth table includes only all valid combinations of the inputs. The valid combinations are those which have exactly one input equal to logic 1 while all other inputs are logic 0's. The Boolean expressions are as follows:

Y0 = I1 +I3 +I5 +I7 +I9

Y1 = I2 +I3 +I6 +I7

Y2 = I4 +I5 +I6 +I7
Y3 = I8 +I9

Given Decimal Numbers are 2, 4, 8, 3 and 7.
(i) $(2)_{10} = (0010)_2$

```
2 |2
   1  - 0
```

(ii) $(4)_{10} = (0100)_2$

```
2 |4
 2| 2 - 0
   1 - 0
```

(iii) $(8)_{10} = (1000)_2$

```
2 |8
 2| 4 - 0
2 |2 - 0
   1 - 0
```

(iv) $(3)_{10} = (0011)_2$

```
2 |3
   1  - 1
```

(iii) $(7)_{10} = (0111)_2$

```
2 |7
 2| 3 - 1
  1 - 1
```

Decimal Input	**BCD Code (8421Code)**			
	2^3	**2^2**	**2^1**	**2^0**
2	0	0	1	0
4	0	1	0	0
8	1	0	0	0
3	0	0	1	1
7	0	1	1	1

SECTION –V

9 (a) Build a BCD-to-Seven Segment decoder circuit with truth table. **10 Marks**

Ans. In **Binary Coded Decimal (BCD)** encoding scheme each of the decimal numbers(0-9) is represented by its equivalent binary pattern(which is generally of 4-bits).

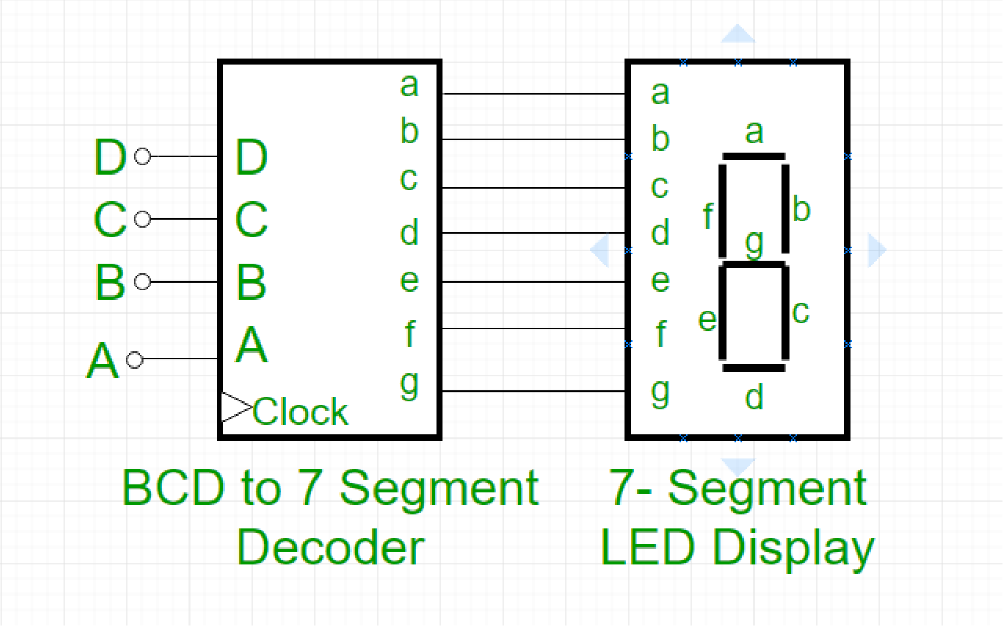
Whereas, **Seven segment** display is an electronic device which consists of seven Light Emitting Diodes (LEDs) arranged in a some definite pattern (common cathode or common anode type), which is used to display Hexadecimal numerals(in this case decimal numbers,as input is BCD i.e., 0-9).

Two types of seven segment LED display:

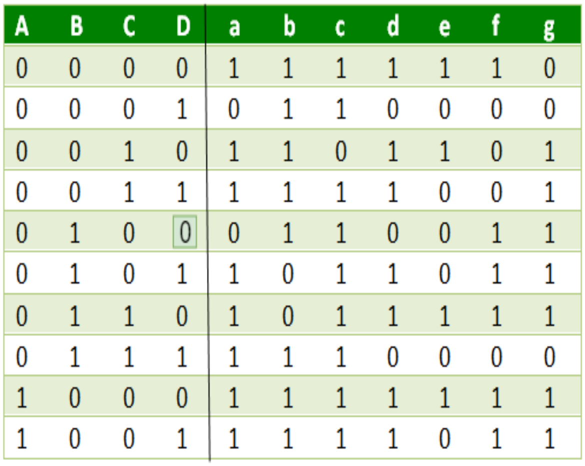
1. **Common Cathode Type:** In this type of display all cathodes of the seven LEDs are connected together to the ground or -Vcc(hence,common cathode) and LED displays digits when some 'HIGH' signal is supplied to the individual anodes.
2. **Common Anode Type:** In this type of display all the anodes of the seven LEDs are connected to battery or +Vcc and LED displays digits when some 'LOW' signal is supplied to the individual cathodes.

But, seven segment display does not work by directly supplying voltage to different segments of LEDs. First, our decimal number is changed to its BCD equivalent signal then BCD to seven segment decoder converts that signals to the form which is fed to seven segment display.

This BCD to seven segment decoder has four input lines (A, B, C and D) and 7 output lines (a, b, c, d, e, f and g), this output is given to seven segment LED display which displays the decimal number depending upon inputs.



Truth Table – For common cathode type BCD to seven segment decoder:



A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

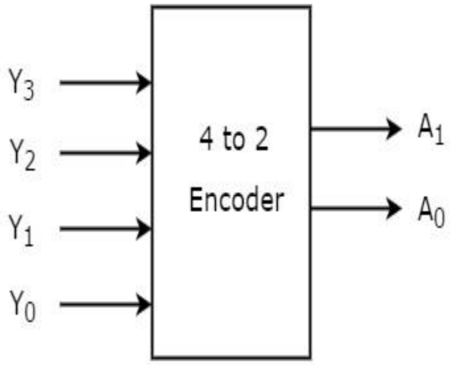
9 (b) (i) Explain the working of 4-line-to-2-line Encoder. 5 Marks

Ans

An **Encoder** is a combinational circuit that performs the reverse operation of Decoder. It has maximum of 2^n input lines and ‘n’ output lines. It will produce a binary code equivalent to the input, which is active High. Therefore, the encoder encodes 2^n input lines with ‘n’ bits. It is optional to represent the enable signal in encoders.

4-line- to- 2-line Encoder

Let 4 to 2 Encoder has four inputs Y_3, Y_2, Y_1 & Y_0 and two outputs A_1 & A_0. The block diagram of 4 to 2 Encoder is shown in the following figure.



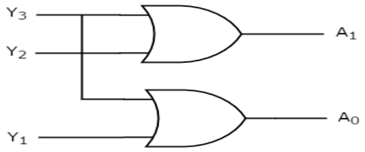
At any time, only one of these 4 inputs can be '1' in order to get the respective binary code at the output. The Truth table of 4 to 2 encoder is shown below.

Inputs				**Outputs**	
Y_3	**Y_2**	**Y_1**	**Y_0**	**A_1**	**A_0**
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

From Truth table, we can write the Boolean functions for each output as

$$A1=Y3+Y2 \text{ and } A0=Y3+Y1$$

We can implement the above two Boolean functions by using two input OR gates. The circuit diagram of 4 to 2 encoder is shown in the following figure.



The above circuit diagram contains two OR gates. These OR gates encode the four inputs with two bits.

9 (b) (ii) Mention any 5 applications of Decoder. 5 Marks

Ans

- It is used in code conversion. i.e analog to digital conversion in the analog decoder.
- It may also be used for data distribution.
- In a high-performance memory system, this decode can be used to minimize the effect of system decoding.
- The decoder is used as address decoders in CPU memory location identification.
- They are mainly used in logical circuits, data transfer.
- Microprocessor selecting different I/O devices.
- Microprocessor memory system selecting different banks of memory.

10 (a) Show what would be the output condition for a two input TTL NAND gate for all the input conditions. 10 Marks

Ans. The circuit diagram of a 2 input TTL NAND gate is as follows:

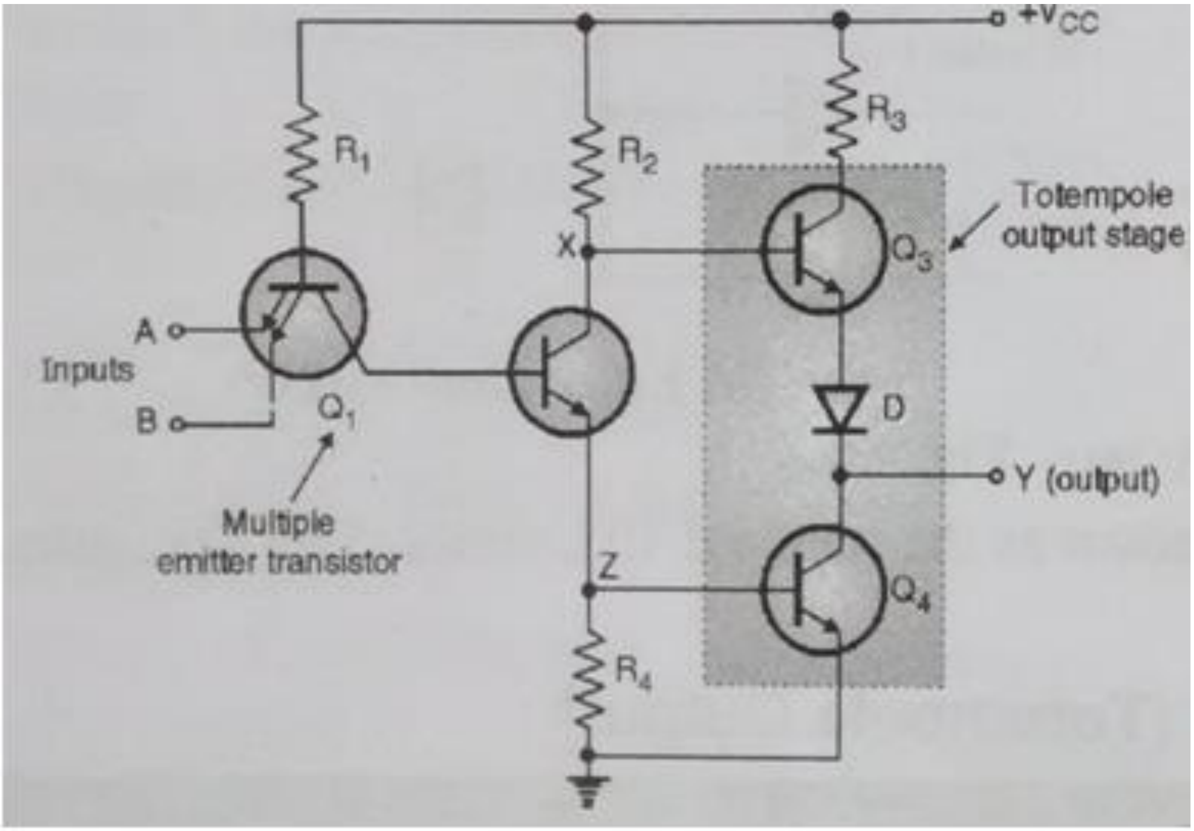


Fig3. Two input TTL NAND gate

- A two input TTL NAND is shown above. A and B are two inputs while Y is the output.

- Operation of the gate:

a) A and B both low: both B-E junctions of Q1 are forward biased. Hence D1 and D2 will conduct to force the voltage at point C to 0.7V. This voltage is insufficient to forward bias B-E junction of Q2. Hence Q2 remains OFF. Therefore its collector voltage rises to VCCVCC. As Q3 is operating in emitter follower mode, output Y will be pulled up to high voltage Y= 1

b) Either A or B low: If any one input is connected to ground with other left open or connected to VCCVCC the corresponding diode (D1 or D2) will conduct. This will pull down voltage at C o 0.7V. This voltage is insufficient to turn on Q2 so it remains OFF. So collector voltage of Q2 will be equal to VCC. This voltage acts as base voltage for Q3. As Q3 acts as an emitter follower, output Y will be pulled to VCCVCC. Y= 1

c) A and B both high: If both A and B are connected to then both diodes D1 and D2 will be reverse biased and do not conduct. Therefore D3 is forward biased and base current is supplied to transistor Q2 via R1 and D3. As Q2 conducts, the voltage at X will drop down and Q3 will be OFF, whereas voltage at Z will increase to turn ON Q4. As Q4 goes into saturation, the output voltage Y will be pulled down to low. Y = 0

The operation is clearly illustrated in the below table.

Inputs		**Transistors State**				**Output**
A	**B**	**Q_1**	**Q_2**	**Q_3**	**Q_4**	**Y**
0	0	On	Off	On	Off	1
0	1	On	Off	On	Off	1
1	0	On	Off	On	Off	1
1	1	Off	On	Off	On	1

10 (b) (i) List the applications of Integrated Circuits. 5 Marks

Ans. The applications of IC’s include:

- They are used in smart phones, mp3 players, laptops, computers.
- IC’s are also used in Television and cameras.
- They are widely used in aircraft and space craft systems.
- IC’s are the basic component used in scientific calculators and digital watches.
- They are used in control systems.

10 (b) (ii) Define the following parameters of Logic families.

1) Speed 2) Fan in 3) Fan out 4) Power Dissipation 5) Noise Margin. 5 Marks

Ans.

1. **Speed:** Speed of a logic circuit is determined by the time between the application of input and change in the output of the circuit.
2. **Fan in:** Fan in is the number of inputs connected to the gate without any degradation in the voltage level.
3. **Fan out:** Fan out specifies the number of standard loads that the output of the gate can drive without impairment of its normal operation.
4. **Power dissipation:** Power dissipation is a measure of power consumed by the gate when fully driven by all its inputs.
5. **Noise Margin:** It is the maximum noise voltage added to an input signal of a digital circuit that does not cause an undesirable change in the circuit output. It is expressed in volts.

CERTIFICATE

“I certify that the model answer script that is prepared by me for the subject CODE: 20EC11T are from prescribed text books and model answer script and scheme of valuation prepared by me are correct”.

Scrutinized by: