1790

Code : 20EC11T

Register Number | | | | | | | | | | |

I Semester Diploma Examination, Oct./Nov.-2021

DIGITAL ELECTRONICS

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

Special Note : Students can answer for max. of **100** marks, selecting any subsection from any main section.

SECTION – I

1. (A) (i) What is an ASCII code ? List any two features of ASCII code. **3**

(ii) (a) Represent $(1011100110)_2$ by its octal equivalent. **7**

(b) Convert $(423)_{10}$ to hexadecimal.

(c) Convert $(24.6)_8$ to decimal.

(B) (i) Explain Octal and Hexadecimal number system. **5**

(ii) State and prove Demorgan's theorem. **5**

2. (A) (i) List different types of number system. **3**

(ii) Solve the following : **7**

(a) $(1101)_2 - (111)_2 = (?)_2$

(b) $(4C)_{16} + (3A)_{16} = (?)_{16}$

(c) $(111)_2 * (101)_2 = (?)_2$

(B) (i) Illustrate the conversion from binary to gray code with one example. **5**

(ii) Show the realization of AND, OR and NOT gates using NOR gates. **5**

SatheeshaKM

SECTION – II

3. (A) (i) Define K-Map. Why it is used ? 3

(ii) Simplify the given Boolean equation by using K-map and realize using logic gates. 7

(a) $f = AB + \bar{A}B + A\bar{B}$

(b) $f = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + \bar{A}B\bar{C} + A\bar{B}\bar{C} + AB\bar{C}$

(B) Explain following gates with symbol, expression and truth table :

AND, OR, NOR, NAND and EXOR. 10

4. (A) (i) Define Duality theorem and give example. 3

(ii) Simplify following Boolean expression using Boolean Algebra and realize using logic gates. 7

(a) $Y = AB + \bar{A}B + A\bar{B}$

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

(B) (i) Illustrate SOP to POS conversion : 10

$F(A, B, C) = \Sigma(0, 2, 3, 5, 7)$

(ii) Illustrate POS to SOP conversion :

$F(A, B, C) = \Pi(2, 3, 5)$

SECTION – III

5. (A) (i) Define Half Adder. Write truth table, logical expression and gate level implementation of Half Adder. 5

(ii) Compare a Series Adders and Parallel Adders. 5

(B) Develop a 2-bit magnitude comparator and implement it using gates. 10

6. (A) (i) Define Full Adder. Write the truth table and logical expression of full adder. 5

(ii) Explain working of 3-bit Parallel Adder with block diagram. 7

(B) Construct AND, OR and EXOR gate using 2 : 1 Multiplexer. 8

SECTION – IV

7. (A) (i) Define Multiplexer and list any five applications. 5

(ii) Explain 2 : 1 Multiplexer with block diagram and truth table. 5

(B) (i) Construct 4 : 1 Multiplexer using 2 : 1 Multiplexers. 4

(ii) Explain 1 : 8 Demultiplexer operation with the help of truth table. 6

8. (A) (i) Define De-multiplexer and list any five applications. 5

(ii) Explain 1 : 2 De-Multiplexer with block diagram and truth table. 5

(B) Design the function 10

$F(A, B, C) = \Sigma(1, 2, 5, 7)$ using

(a) 8 to 1 MUX

(b) 4 to 1 MUX

SECTION – V

9. (A) Sketch and explain decimal to BCD encoder with logic symbol, truth table, Boolean Expression, logic diagram using gates. 10

(B) (i) List IC classification based on scale of integration. 5

(ii) Define Fun-out with respect to logic family specification. 2

(iii) Describe the features of TTL family. 3

[Turn over

SatheeshaKM

10. (A) Sketch and explain BCD to seven segment decoder (IC 7447) with logic circuit and truth table. **10**

(B) (i) List the features of CMOS logic family. **3**

(ii) Define Fan-in with respect to logic family specification. **2**

(iii) Describe the interfacing between CMOS to TTL. **5**

Satheesha K M

Scheme of valuation

Section-1

1. A. i) Definition of ASCII (1 Mark) + Listing any two features (1 x 2 =2 Marks)
1. A. ii) a) Octal equivalent (2 Marks) b) Hexadecimal equivalent (2 Marks) c) decimal equivalent (3 Mark)
1. B. i. a) Explanation of Octal system (2 Marks) b) Explanation of Hexadecimal system (3 Marks)
1. B. ii. Statement of Demorgan's theorem (2 Marks) + Verification (3 Marks) [Any one theorem]
2. A. i) Listing any three-number system (1 x 3 =3 Marks)
2. A. ii) a) Subtraction (2 Marks) b) Addition (2 Marks) c) Multiplication (3 Marks)
2. B. i. Conversion steps (3 Marks) + Example (2 Marks)
2. B. ii. Realization of AND (2 Marks) + OR (2 Marks) + NOT (1 Mark)

Section-2

3. A. i) Definition of K-Map (1 Mark) + Need of K-Map (2 Marks)
3. A. ii) a) Kmap (1 Mark) +Simplified expression (1 Mark) + realization (1 Mark) b) Kmap (2 Mark) +Simplified expression (1 Mark) + realization (1 Mark)
3. B. a) (2 x 5 =10 Marks) - Each gate realization carries 2 Marks each. [Symbol/Expression (1 Mark) + Truth table (1 Mark)]
4. A. i) Definition (1 Mark) + One example (2 Marks)
4. A. ii) a) For applying algebraic rule and simplifying carries 2 marks + realization (1 Mark) b) For applying algebraic rule and simplifying carries 3 marks + realization (1 Mark)
4. B. i. SOP to POS Conversion using example problem – Step-1-(1 Mark) + Step-2-(2 Marks) + Final Answer -(2 Marks)
4. B. ii. POS to SOP Conversion using example problem – Step-1-(1 Mark) + Step-2-(2 Marks) + Final Answer -(2 Marks)

Section-3

5. A. i) Definition of Half Adder (1 Mark) + Truth table (1 Mark) + Logical expression (1 Mark) + Gate level implement (2 Marks)
5. A. ii) Any 5 points. Each point carries 1 Mark each.
5. B. Block diagram (2 Mark) + Truth Table (3 Marks) + K-Map (A=B,A>B,A<B) (3 Marks) + realization (2 Mark)

Satheesha K M

6. A. i) Definition of Full Adder (1 Mark) + Truth table (2 Mark) + Logical expression (2 Mark)
6. A. ii) a) Block diagram (4 Marks) + Explanation (3 Marks)
6. B. Realization of AND (3 Marks) + OR (2 Marks) + EX-OR (3 Marks)

Section-4

7. A. i) Definition of Multiplexer (2 Mark) + Listing of any five applications carries 3 Marks.
7. A. ii) Block Diagram (3 Mark) + Truth Table (2 Marks)
7. B. a) Block diagram (4 Marks)
7. B. b) Block diagram (2 Mark) + Explanation (2 Marks) + Truth Table (2 Marks)
8. A. i) Definition of De-Multiplexer (2 Mark) + Listing of any five applications carries 3 Marks.
8. A. ii) Block Diagram (3 Mark) + Truth Table (2 Marks)
8. B. a) Final solution block diagram (4 Marks)
8. B. b) K-Map/Truth table (2 Marks) + Final solution block diagram (4 Marks)

Section-5

9. A. Logic Symbol (2 Marks) + Truth Table (3 Marks) + Boolean Expressions (2 Marks) + Gate level implement (3 Marks)
9. B. i) Any 5 classification. Each carry 1 Mark each.
9. B. ii) Definition (1 Mark) + Diagram representation (1 Mark)
9. B. iii) Any three feature each carries 1 Mark each.
10. A. i) Logic circuit (2 Marks) + Truth Table (4 Marks) + Explanation (4 Marks)
10. B. i) Any 3 feature. Each carry 1 Mark each.
10. B. ii) Definition (1 Mark) + Diagram representation (1 Mark)
10. B. iii) Diagram (3 Marks) + Explanation (2 Marks)

1) What is ASCII Code? List the any two features of ASCII code.

ASCII (AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE) CODE:

The standard binary code for the alphanumeric characters is the American Standard Code for Information Interchange (ASCII), which uses seven bits to code 128 characters, It allows manufacturers to standardize computer hardware such as keyboards, printers and video displays.

Features:

1. It has a total of 2^7(= 128) possible combinations to represent letters in the alphabet, punctuation marks and numbers.
2. The code is divided into two groups as 3 + 4 bits. The first three bits in the code are used to identify whether the remaining four bits represent letters, numerals, or punctuation marks. For example, 100 and 101 represent upper -case letters. Upper-case letter A is represented by 100 0001 (4116), B is represented by 100 0001 (4216) and so on. 110 and 111 represent lower-case letters. Lower-case letter a is represented by 110 0001 (6116), b is represented by 110 0001 (6216) and so on.
3. Redundant bits can be added for error-detection and correction.

1. ii)

a) Represent $(1011100110)_2$ by its octal equivalent

Solution: Add 0s so that the given binary number can be arranged in groups of 3 bits.

$1011100110_2 = 001\ 011\ 100\ 110_2$

Binary number:	001	011	100	110
	↓	↓	↓	↓
Octal equivalent:	1	3	4	6

$1011100110_2 = 1346_8$

Satheesha K M

1. ii) b) Convert $(423)_{10}$ to hexadecimal.

Solution:

			Remainder	
16	423	→	7	(LSD)
16	26	→	A	
16	1	→	1	(MSD)
	0			

$423_{10} = 1A7_{16}$

1. ii) c) Convert $(24.6)_8$ to decimal

Solution:

Octal number:	2	4	.	6
Octal weight:	8^1	8^0	.	8^{-1}
Decimal value:	8	1	.	0.125

$24.6_8 = (8 \times 2 + 1 \times 4) . (0.125 \times 6)$
$= (16 + 4) . (0.75)$
$= 20.75_{10}$
$24.6_8 = 20.75_{10}$

1. B. i) Explain Octal and Hexadecimal number system.

OCTAL NUMBER SYSTEM:
The octal number system has a base of 8, meaning that it has eight possible digits: 0, 1, 2, 3, 4, 5, 6 and 7. Thus, each digit of an octal number can have any value from 0 to 7. The octal system is a positional-value system, wherein each digit has its own value or weight expressed as a power of 8. The digit positions in an octal number have weights as shown in below table.

Octal weight:	8^4	8^3	8^2	8^1	8^0	.	8^{-1}	8^{-2}	8^{-3}
Decimal value:	4096	512	64	8	1	.	0.125	0.016	0.002
	MSD					Octalpoint			LSD

Examples of Octal Numbers : $(256)_8$, $(432.35)_8$, (250.06)8, $(125.56)_8$,etc

HEXADECIMAL NUMBER SYSTEM:

The hexadecimal number system has a base of 16, meaning that it has 16 possible digits 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F.. Thus, each digit of an hexadecimal number can have any value

from 0 to F. This system is a positional-value system, wherein each digit has its own weight expressed as a power of 16. The digit positions in a hexadecimal number have weights as shown

Hexadecimal weight:	16^3	16^2	16^1	16^0	.	16^{-1}	16^{-2}
Decimal value:	4096	256	16	1	.	0.0625	0.004
	MSD				Hexadecimal point		LSD

Hexadecimal numbers are used extensively in microprocessor based systems and computers. They are shorter than binary numbers.

Examples of Hexadecimal Numbers : $(24A6)_{16}$, $(345F.50)_{16}$, etc

1. B. ii. State and prove Demorgan's theorem

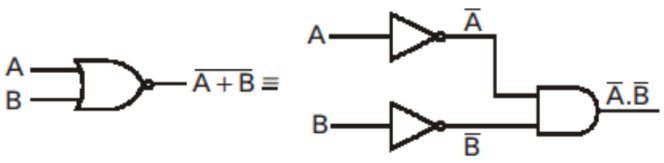
De Morgan's First Theorem

DeMorgan's first theorem is stated as follows in words:

The complement of a sum is equal to the product of the individual complements.

It says that the complement of two or more variables OR ed is the same as the AND of the complements of each individual variable.

DeMorgan's First Theorem



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

DeMorgan's first theorem is stated as follows in equation form:

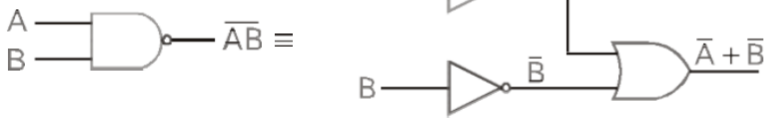
$$\overline{A+B} = \overline{A}.\overline{B}$$

DeMorgan's second theorem is stated as follows in words:

The complement of a product is equal to the sum of the individual complements.

It says that the complement of two or more variables ANDed is the same as the OR of the complements of each individual variable.

Satheesha K M



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

DeMorgan's second theorem is stated as follows in equation form

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

2. A. i) List different types of number system.

In general there are four different number system they are ;

1) Decimal Number System
2) Binary Number System
3) Octal Number System
4) Hexadecimal Number System

2. A. ii. a)$(1101)_2$ - $(111)_2$ = $(?)_2$

$$\begin{array}{rl} 1101 & (13)_{10} \\ 111 & (07)_{10} \\ \hline 0110 & (06)_{10} \end{array}$$

2. A. ii. b) $(4C)_{16}$ + $(3A)_{16}$ = $(?)_{16}$

$$\begin{array}{r} 4C_{16} \\ +\ 3A_{16} \\ \hline 86_{16} \end{array}$$

Right Column: $C_{16} + A_{16} = 12_{10} + 10_{10} = 22_{10} = 16_{16} = 6_{16}$ (1 Carry)
Left Column: $4_{16} + 3_{16} + 1_{16}$ (Carry) = $4_{10} + 3_{10} + 1_{10} = 8_{10} = 8_{16}$

Satheesha K M

2. A. ii. c) $(111)_2 * (101)_2 = (?)_2$

$$
\begin{array}{ll}
111 & (7)_{10} \\
\times 101 & (5)_{10} \\
\hline
111 & \\
000+ & \\
111++ & \\
\hline
100011 & (35)_{10}
\end{array}
$$

2. B. i. Illustrate the conversion from binary to gray code.

BINARY TO GRAY CONVERSION

In the conversion from binary to gray the following rules apply:

1. The most significant digit (left most) in the Gray code is same as the corresponding digit in the binary number.
2. Going from left to right, add each adjacent pair of binary digits to get the next Gray code digit, discard carry.
3. Let the binary number be B3B2B1B0 with B3being the MSB and B0 the LSB.
4. Let the corresponding Gray code be G3G2G1G0 with G3 being the MSB and G0 the LSB.
5. G3in the Gray code is the same as the corresponding digit B3in the binary number. That is,

G3= B3

The other bits in the Gray code are given by the following logical expressions:

G2= B3+ B2

G1= B2+ B1

G0= B1+ B0

Example 1. Convert binary 1011 to Gray code**.**
Solution:

$B_3\ B_2\ B_1\ B_0 = 1011$

$G_3 = B_3 = 1$ (MSB)

$G_2 = B_3 + B_2 = 1 + 0 = 1$

$G_1 = B_2 + B_1 = 0 + 1 = 1$

$G_0 = B_1 + B_0 = 1 + 1 = 0$ (LSB) (Discard carry)

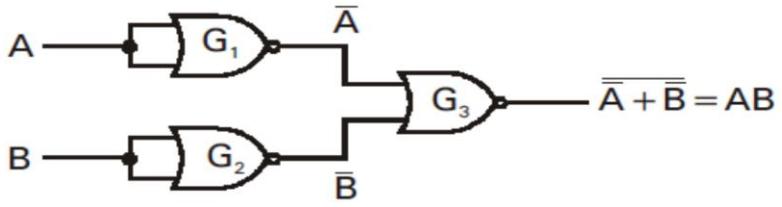
1011 (Binary) = 1110 (Gray)

2. B. ii. Show realization of AND, OR and NOT gates using NOR gate.

AND GATE.

Two NOR gates are used to invert the two input variables before they are applied to another NOR gate.

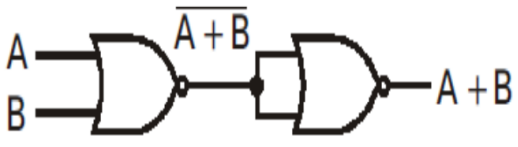
The AND output is derived as follows: $Y = \overline{\overline{AB}} = AB$



OR GATE:

A NOR gate is used to invert (complement) a NOR output to form the OR function.

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



Satheesha K M

NOT GATE

A NOR gate can be used a NOT gate by tying all its input terminals together and feeding the signal to be inverted to the common terminal as shown.

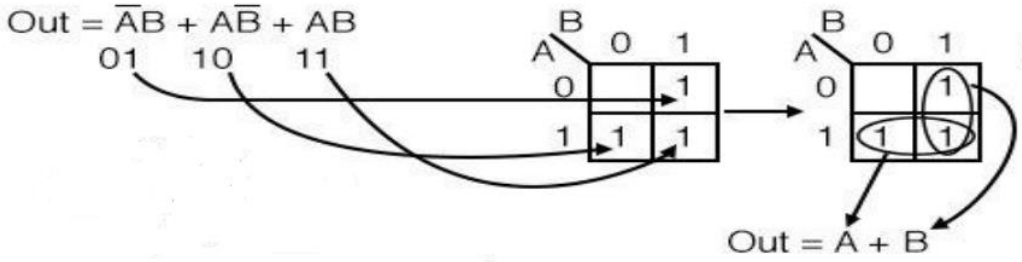
3. A. i. Define K-Map. Why it is used?

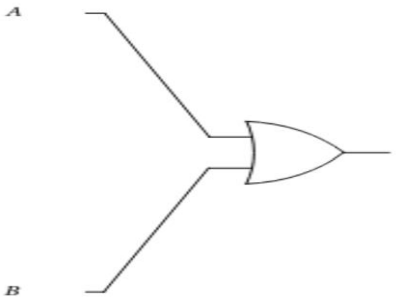
"Karnaugh Map is a graphical method, which consists of 2^n cells for n variables. The adjacent cells are differed only in single bit position."

Simplification of the Boolean functions having more than 4 variables using Boolean postulates and theorems is a lengthy and time-consuming process. We must re-write the simplified expressions after each step which may add error to the process. To overcome this difficulty, American physicist Maurice Karnaugh introduced a method for simplification of Boolean functions in an easy way.

3. A. ii. Simplify following Boolean expression using K-Map and draw the logic diagram of simplified expression.

a) Out = $A\,B + \bar{A}\,B + A\,\bar{B}$



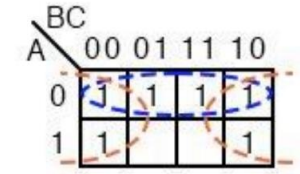


Satheesha K M

b) Out = $\bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + \bar{A}\,B\,C + \bar{A}\,B\,\bar{C} + A\,\bar{B}\,\bar{C} + A\,B\,\bar{C}$

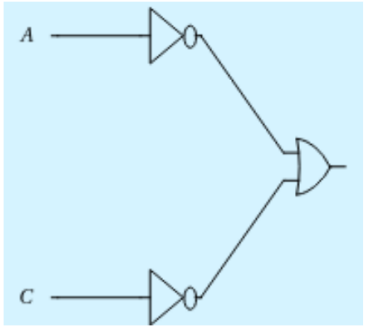
Solution:

Out = $\bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + \bar{A}\,B\,C + \bar{A}\,B\,\bar{C} + A\,\bar{B}\,\bar{C} + A\,B\,\bar{C}$



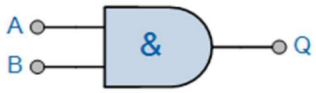
Out = $\bar{A} + \bar{C}$

The six cells above from the unsimplified equation can be organized into two groups of four. These two groups should give us two p-terms in our simplified result of $\bar{A} + \bar{C}$



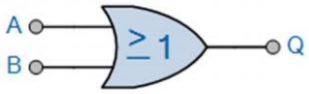
4. B. Explain following gates with symbol expression and truth table: AND, OR, NOR, NAND and EXOR

AND Gate: The Symbol, expression, and the truth table of AND gate

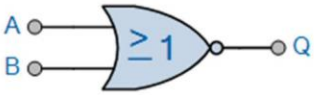
Symbol	Truth Table		
 2-input AND Gate	B	A	Q
	0	0	0
	0	1	0
	1	0	0
	1	1	1
Boolean Expression Q = A.B	Read as A AND B gives Q		

Satheesha K M

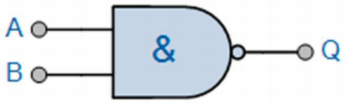
OR Gate: The Symbol, expression, and the truth table of OR gate

Symbol	Truth Table		
 2-input OR Gate	B	A	Q
	0	0	0
	0	1	1
	1	0	1
	1	1	1
Boolean Expression **Q = A+B**	Read as A **OR** B gives Q		

NOR Gate: The Symbol, expression and the truth table of NOR gate

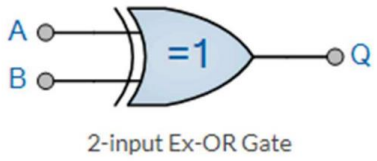
Symbol	Truth Table		
 2-input NOR Gate	B	A	Q
	0	0	1
	0	1	0
	1	0	0
	1	1	0
Boolean Expression $Q = \overline{A+B}$	Read as A **OR** B gives **NOT** Q		

NAND Gate: The Symbol, expression and the truth table of NAND gate

Symbol	Truth Table		
 2-input NAND Gate	B	A	Q
	0	0	1
	0	1	1
	1	0	1
	1	1	0
Boolean Expression $Q = \overline{A.B}$	Read as A **AND** B gives **NOT** Q		

Satheesha K M

EX-OR Gate: The Symbol, expression, and the truth table of EX-OR gate

<table>
<tr><th colspan="2">Symbol</th><th colspan="3">Truth Table</th></tr>
<tr><td colspan="2" rowspan="5"> 

2-input Ex-OR Gate</td><td>B</td><td>A</td><td>Q</td></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>1</td></tr>
<tr><td>1</td><td>1</td><td>0</td></tr>
<tr><td colspan="2">Boolean Expression Q = A ⊕ B</td><td colspan="3">A OR B but NOT BOTH gives Q</td></tr>
</table>

4. A. i) Define Duality theorem and give examples

The duality theorem is one of those elegant theorems proved in advanced mathematics. We will state the theorem without proof. Here is what the duality theorem says.

Starting with a Boolean relation, you can derive another Boolean relation by

1. Changing each OR sign to an AND sign.
2. Changing each AND sign to an OR sign.
3. Complementing any 0 or 1 appearing in the expression.

1) F or instance, Eq. says that

A+0=A .

The dual relation is A· 1 =A.

This dual property is obtained by changing the OR sign to an AND sign, and by complementing the 0 to get a 1. The duality theorem is useful because it sometimes produces a new Boolean relation.

4. B. Simplify following Boolean expression using Boolean Algebra and realize using logic gates.

a) Out = $A\,B + \bar{A}\,B + A\,\bar{B}$

Out = $A\,B + \bar{A}\,B + A\,\bar{B}$

= $B\,(A + \bar{A}) + A\,\bar{B}$ $\quad (A + \bar{A} = 1)$

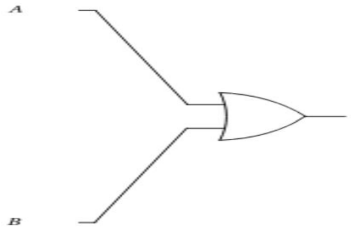
Satheesha K M

$= B\,(1) + A\,\bar{B}$ $\qquad B\,(1) = B$

$= B + A\,\bar{B}$ $\qquad B + A\,\bar{B} = B + A$

Out $= B + A$

Out $= A + B$



b) Out $= \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + \bar{A}\,B\,C + \bar{A}\,B\,\bar{C} + A\,\bar{B}\,\bar{C} + A\,B\,\bar{C}$

Out $= \bar{A}\,\bar{B}\,\bar{C} + \bar{A}\,\bar{B}\,C + \bar{A}\,B\,C + \bar{A}\,B\,\bar{C} + A\,\bar{B}\,\bar{C} + A\,B\,\bar{C}$

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

$= \bar{A}\,\bar{B}\,(1) + \bar{A}\,B\,(1) + A\,\bar{C}\,(1)$

$= \bar{A}\,\bar{B} + \bar{A}\,B + A\,\bar{C}$

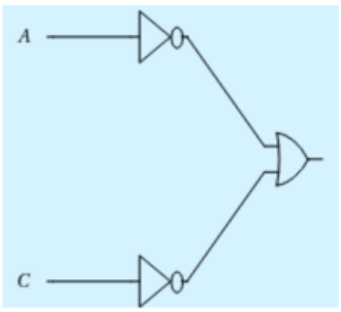
$= \bar{A}\,(\bar{B} + B) + A\,\bar{C}$

$= \bar{A}\,(1) + A\,\bar{C}$

$= \bar{A} + A\,\bar{C}$

Out $= \bar{A} + \bar{C}$

Satheesha K M



4. B a) Illustrate SOP to POS conversion: F (A, B, C) = ∑ (0, 2, 3, 5, 7).

To convert the SOP form into POS form, follow the steps below

Step 1: we should change the Σ to Π.

Step 2: write the numeric indexes of missing variables of the given Boolean function.

Example 1: Let us consider the SOP function $F(A,B,C) = \sum(0, 2, 3, 5, 7) = \bar{A}\bar{B}\bar{C} + \bar{A}B\bar{C} + \bar{A}BC + A\bar{B}C + ABC$ is written in POS form by

Step 1: changing the operational sign to Π

Step 2: writing the missing indexes of the terms, 001(1), 100(4) and 110(6). Now write the sum form for these noted terms.

$001 = (A + B + \bar{C})$ $100 = (\bar{A} + B + C)$ $110 = (\bar{A} + \bar{B} + C)$

Writing down the new equation in the form of POS form,

F(A,B,C) = Π (1, 4, 6) $= (A + B + \bar{C}) \bullet (\bar{A} + B + C) \bullet (\bar{A} + \bar{B} + C)$

4. B. b) Illustrate POS to SOP conversion: F (A, B, C) =Π (2, 3, 5)

To convert the POS form into SOP form, follow the steps below

Step 1: we should change the Π to Σ

Step 2: write the numeric indexes of missing terms of the given Boolean function.

Example 3: Let us consider the POS function F(A,B,C) =Π(2,3,5) = $(A + \bar{B} + C) \bullet (A + \bar{B} + \bar{C}) \bullet (\bar{A} + B + \bar{C})$ is written in SOP form by

Step 1: Changing the operational sign to Σ

Step 2: Writing the missing indexes of the terms, 000(0), 001(1), 100(4), 110(6) and 111(7). Now write the product form for these noted terms.

Satheesha K M

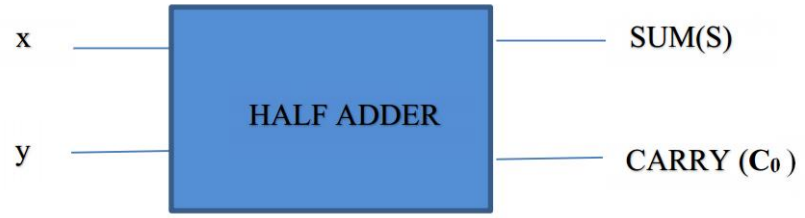
$000 = \bar{A}\bar{B}\bar{C}$ $001 = \bar{A}\bar{B}C$ $100 = A\bar{B}\bar{C}$

$110 = AB\bar{C}$ $111 = ABC$

Writing down the new equation in the form of SOP form,

$$\mathbf{F(A,B,C) = \Sigma(0, 1, 4, 6, 7)} = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB\bar{C} + ABC$$

5. **A. i) Define Half Adder. Write the truth table, logical expression, and gate level implementation of Half Adder.**
   Half Adder is a combinational circuit that adds two binary digits (addend and augend) to produce sum (S) and carry (C0).



Truth table of Half Adder:

INPUTS		OUTPUTS	
x	**y**	**C_0**	**S**
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Logic Equation:

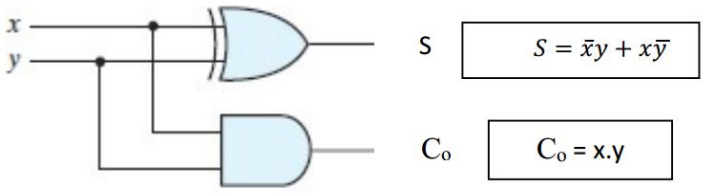
From the above truth table, the simplified Boolean functions

$$S = \bar{x}y + x\bar{y} = x \oplus y$$

C_o=x.y

Therefore, S can be implemented using an EX -OR and Co can be implemented using AND gate

Satheesha K M



5. A. ii. Compare: Serial Adders and Parallel Adders

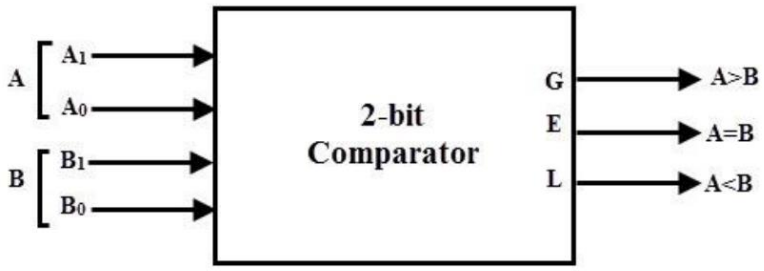
SERIAL ADDERS	PARALLEL ADDERS
Adds only one bit at a time	All bits are added at a time
Time required for addition depends on number of bits	Time required for addition does not depends on number of bits
Speed of response is slower	Speed of response is faster
Requires only one full adder	Number of full adders is equal to number of bits in the binary number
They use shift registers	They use registers with parallel load capacity
They are cheaper	They are expensive

5. B. Develop a 2-bit magnitude comparator, and implement it using gates.

Two-Bit magnitude comparator:

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.

The figure below shows the block diagram of a 2 - bit magnitude comparator.



SatheeshaKM

Truth table of 2-bit magnitude comparator:

INPUTS				OUTPUTS		
A_1	A_0	B_1	B_0	A>B	A=B	A<B
0	0	0	0	0	1	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	0	1
0	1	0	0	1	0	0
0	1	0	1	0	1	0
0	1	1	0	0	0	1
0	1	1	1	0	0	1
1	0	0	0	1	0	0
1	0	0	1	1	0	0
1	0	1	0	0	1	0
1	0	1	1	0	0	1
1	1	0	0	1	0	0
1	1	0	1	1	0	0
1	1	1	0	1	0	0
1	1	1	1	0	1	0

K-map Simplification:

For A>B:

A_1A_0 \ B_1B_0	00	01	11	10
00	0	0	0	0
01	1	0	0	0
11	1	1	0	1
10	1	1	0	0

$=A_1B_1' + A_0B_1'B_0' + A_1A_0B_0'$

Satheesha K M

For A=B:

A_1A_0 \ B_1B_0	00	01	11	10
00	1	0	0	0
01	0	1	0	0
11	0	0	1	0
10	0	0	0	1

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

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

$=(A_0B_0 + A_0' B_0') (A_1B1 + A_1' B_1')$

$=(A_0 \odot B_0) (A_1 \odot B_1)$

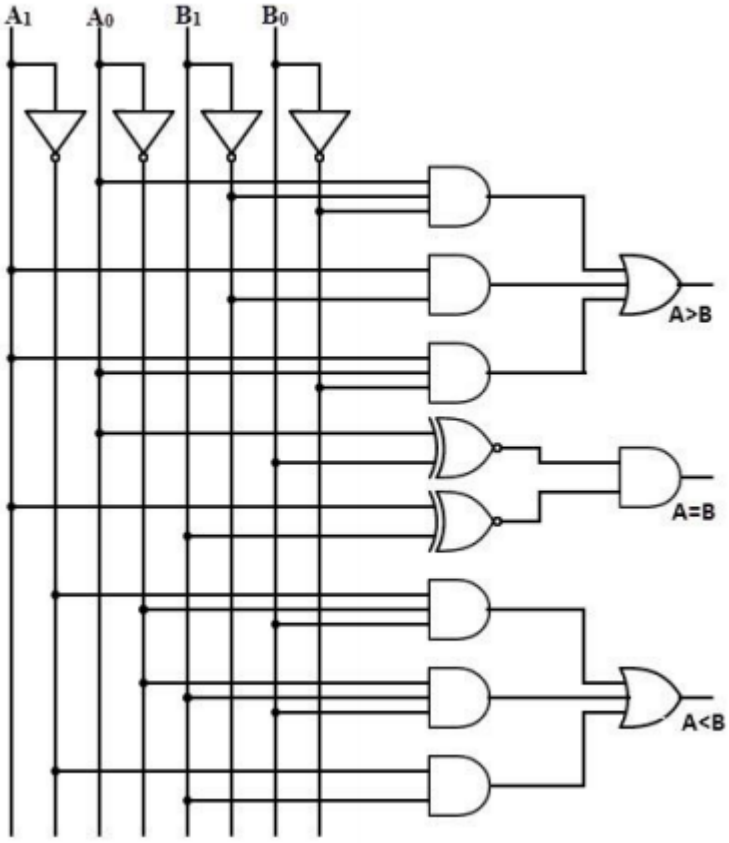
For A< B:

A_1A_0 \ B_1B_0	00	01	11	10
00	0	1	1	1
01	0	0	1	1
11	0	0	0	0
10	0	0	1	0

$Y=A_1'B_1 + A_0'B_1B_0 + A_1'A_0'B_0$

Satheesha K M

GATE LEVEL IMPLEMENTATION



6. A. i. Define Full Adder. Write the truth table, logical expression, and gate level implementation of Full Adder.

A **FULL-ADDER** is a combinational arithmetic logic circuit that performs addition of three binary digits and generates two outputs. It is used for adding two input bits and an input carry and generates a sum and carry output. The two inputs are taken as A and B, they represent the two significant bits to be added. The third input Cin, represents the carry from the previous lower significant bit (LSB) position.

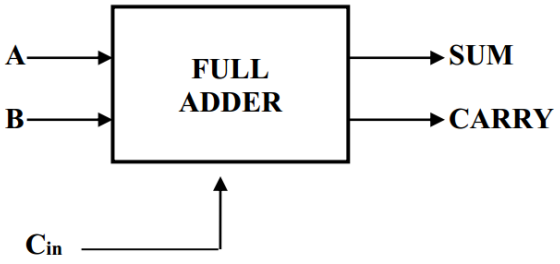


Figure below shows the block schematic of FULL-ADDER.

TRUTH TABLE

Satheesha K M

INPUTS			OUTPUTS	
A	B	C_{in}	SUM	CARRY
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

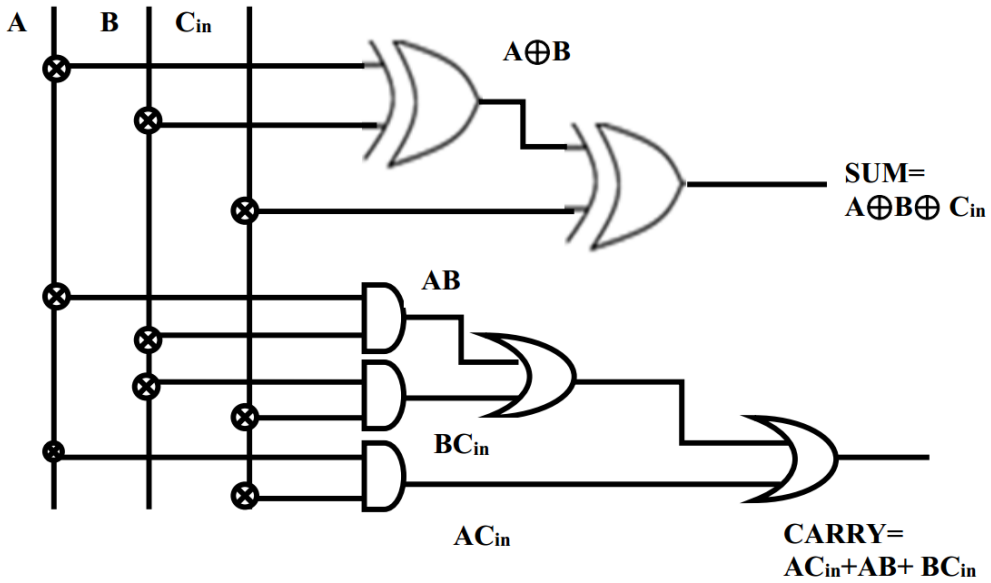
Logic Equation:

From the above truth table, the simplified Boolean functions

$$\mathbf{SUM} = A \oplus B \oplus C_{in}$$

$$\mathbf{CARRY} = AC_{in} + AB + BC_{in}$$

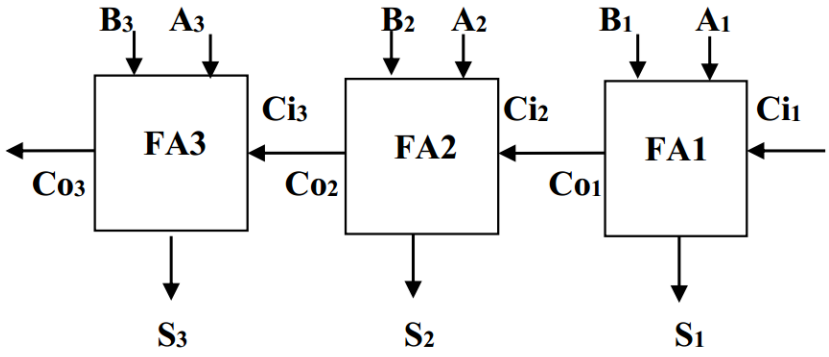
Logic Diagram:



Satheesha K M

6. A. ii) Explain working of 3-bit Parallel Adder with block diagram.

A binary parallel adder is a digital circuit that produces the arithmetic sum of two binary numbers in parallel. It consists of full adders connected in a chain, with the output carry from each full adder connected to the input carry of next full adder in the chain. Thus several full adders are connected to form adders that add several bits at one time. A three bit parallel adder consists of three full adders, with the output carry from each full adder connected to the input carry of next full adder in the chain.



Working of 3-bit parallel Adder:

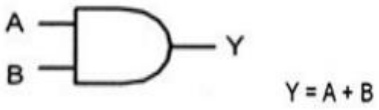
1) As shown in the above figure, the first full adder FA1 adds A1 and B1 along with the carry Ci1 to generate the sum S1 (the first bit of the output sum) and the carry Co1 which is connected to the next adder in chain.

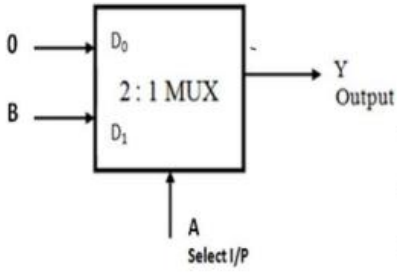
2) The second full adder FA2 uses this carry bit as an input carry bit Ci2 to add with the input bits A2 and B2 to generate the sum S2(the second bit of the output sum) and the carry Co2 which is again further connected to the next adder in chain

3) The third full adder FA3 uses this carry bit as an input carry bit Ci3 to add with the input bits A3 and B3 to generate the sum S3(the third bit of the output sum) and the carry Co3 .

6. B. Construct AND, OR and EXOR gate using 2:1 Multiplexer.

Implementation of AND gate using 2 : 1 Mux

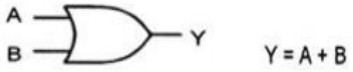


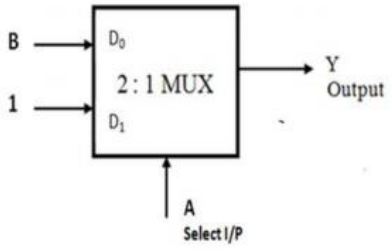


	D_0	D_1
$\bar{B}$	0	1
B	2	3
	0	B

B	A	Y =A.B
0	0	0
0	1	0
1	0	0
1	1	1

Implementation of OR gate using 2 : 1 Mux

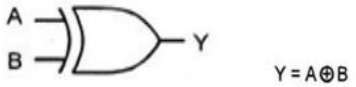


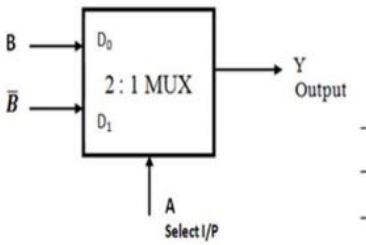


	D_0	D_1
$\bar{B}$	0	1
B	2	3
	B	1

B	A	Y =A+B
0	0	0
0	1	1
1	0	1
1	1	1

Implementation of EX-OR gate using 2 : 1 Mux





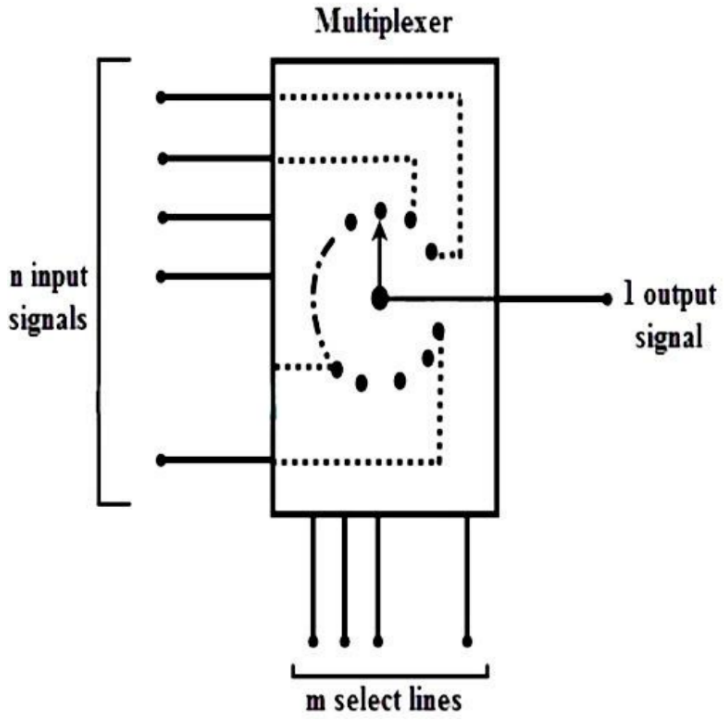
	D_0	D_1
$\bar{B}$	0	1
B	2	3
	B	$\bar{B}$

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

SatheeshaKM

7. A. i. Define Multiplexer and List any five applications.

A multiplexer (MUX), also known as a data selector, is a device that selects between several analog or digital input signals and forwards the selected input to a single output line. The selection is directed a separate set of digital inputs known as select lines. A multiplexer of inputs has select lines, which are used to select which input line to send to the output.



APPLICATIONS OF MULTIPLEXERS.

A Multiplexer is used in numerous applications like, where multiple data can be transmitted using a single line.

Communication System: A Multiplexer is used in communication systems, which has a transmission system and a communication network. A Multiplexer is used to increase the efficiency of the communication system by allowing the transmission of data such as audio & video data from different channels via cables and single lines.

Computer Memory: A Multiplexer is used in computer memory to keep up a vast amount of memory in the computers, and to decrease the number of copper lines necessary to connect the memory to other parts of the computer.

Telephone Network: A multiplexer is used in telephone networks to integrate the multiple audio signals on a single line of transmission.

Transmission from the Computer System of a Satellite: A Multiplexer is used to transmit the data signals from the computer system of a satellite to the ground system by using a GSM communication.

Data Routing: Multiplexers are extensively used in data routing applications to route the data to a one particular destination from one of several sources.

Logic Function Generator: In place of logic gates, a logical expression can be generated by using a multiplexer. It is possible to connect the multiplexer such that it duplicates the logic of any truth table. In such cases it can generate the Boolean algebraic function of a set of input variables.

Parallel to Serial Conversion: A multiplexer circuit can be used to convert the parallel data to serial data, so as to reduce parallel buses to serial signals. This type of conversion is needed in telecommunication, test and measurement, military/aerospace, data communications applications.

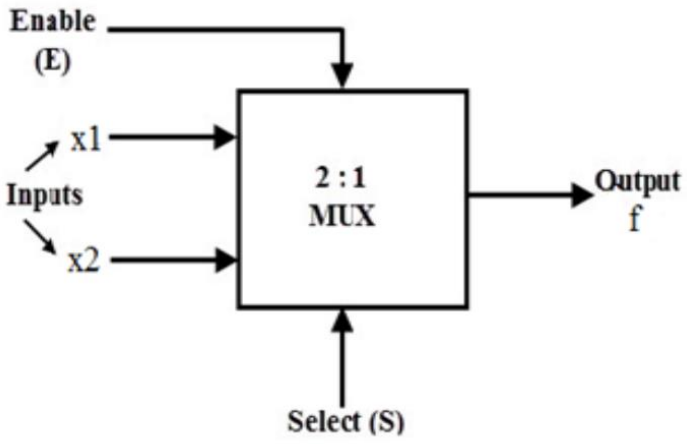
Operation sequencing: Sequence of operation can be done by using multiplexer. Wave form generation: Wave form can be generating by using multiplexer for different applications.

7. A. ii) Explain 2:1 Multiplexer with block diagram and truth table.

2:1 MULTIPLEXER

A 2-to-1 multiplexer is the digital multiplexer circuit that has two data inputs x1 and x2, one selects line S and one output f. To implement a 2-to-1 multiplexer circuit we need 2 AND gates, an OR gate, and a NOT gate.

The block diagram, logic symbol and switching circuit analogy of 2-to-1 multiplexer is shown in the figure below.



SatheeshaKM

ENABLE	SELECT	OUTPUT
E	S	f
0	X	0
1	0	X1
1	1	X2

7. B. ii) a) Construct 4:1 Multiplexer using 2:1 Multiplexers.

REALIZATION OF 4:1 MUX USING 2:1 MUX

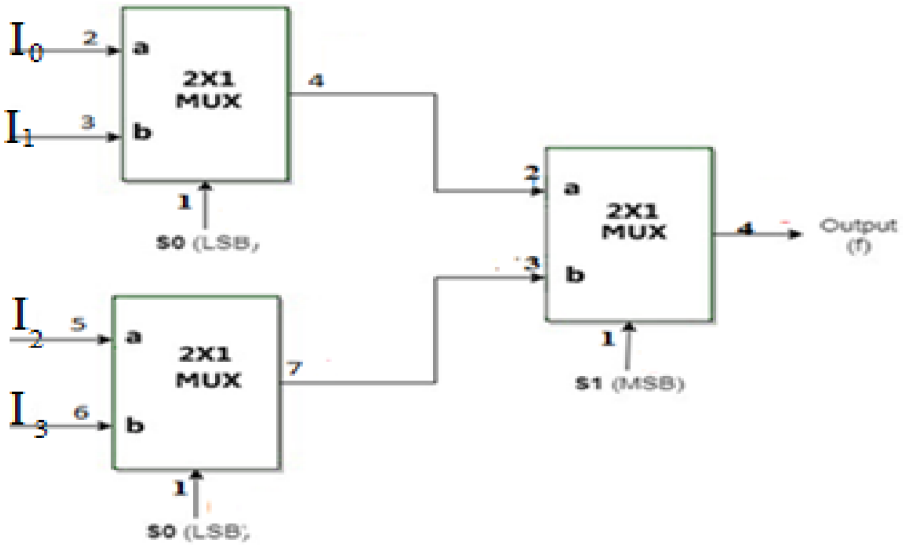
Let us implement 4x1 Multiplexer using 2x1 Multiplexers. We know that 2x1 Multiplexer has 2 data inputs, 1 selection lines and one output. Whereas, 4x1 Multiplexer has 4 data inputs, 2 selection lines and one output.

So, we require two 2x1 Multiplexers in first stage to get the 4 data inputs. Since, each 2x1 Multiplexer produces one output, we require a2x1 Multiplexer in second stage by considering the outputs of first stage as inputs and to produce the final output.

Let the 4x1 Multiplexer has four data inputs I3 to I0, two selection lines S1 &S0 and one output Y. When we select selection lines, makes sure that order is correct. I.e. First stage mux with selection line S0 and second stage mux S1.

The Pin diagram, circuit connection, Truth table, and logical expression of 4x1 Multiplexer

is shown below.

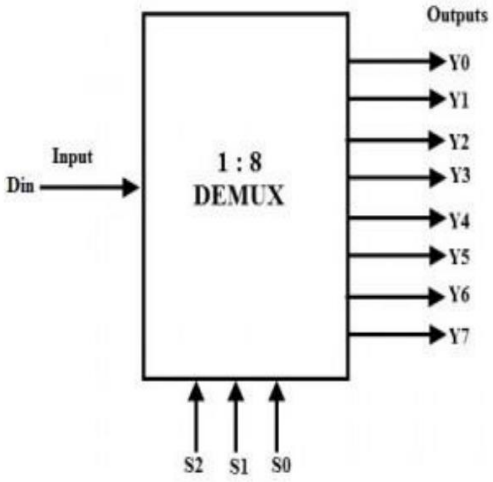


Satheesha K M

7. B. ii) b) Explain 1:8 Demultiplexer using gates with the help of truth table.

1-to-8 Demultiplexer

1) A 1 to 8 demultiplexer consists of one input line, 8 output lines and 3 select lines. Let the input be D, S1 and S2 are two select lines and eight outputs from Y0 to Y7.
2) The below figure shows the block diagram of a 1-to-8 demultiplexer that consists of single input D, three select inputs S2, S1 and S0 and eight outputs from Y0 to Y7.
3) It distributes one input line to one of 8 output lines depending on the combination of select inputs.



S2	S1	S0	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
0	0	0	D	0	0	0	0	0	0	0
0	0	1	0	D	0	0	0	0	0	0
0	1	0	0	0	D	0	0	0	0	0
0	1	1	0	0	0	D	0	0	0	0
1	0	0	0	0	0	0	D	0	0	0
1	0	1	0	0	0	0	0	D	0	0
1	1	0	0	0	0	0	0	0	D	0
1	1	1	0	0	0	0	0	0	0	D

The truth table of 1 to 8 demux is shown with 3 select inputs S0, S1, S2 & 8 outputs Y0 to Y7. From truth table, the Boolean expressions for all the outputs can be written as-

$Y0 = D\,\overline{S2}\,\overline{S1}\,\overline{S0}$

$Y1 = D\,\overline{S2}\,\overline{S1}\,S0$

$Y2 = D\,\overline{S2}\,S1\,\overline{S0}$

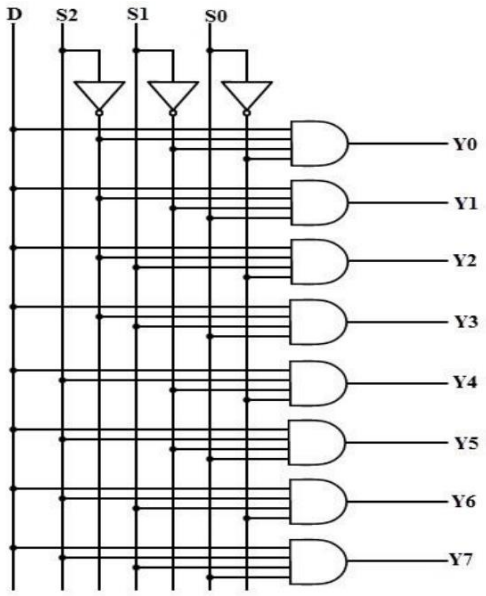
$Y3 = D\,\overline{S2}\,S1\,S0$

$Y4 = D\,S2\,\overline{S1}\,\overline{S0}$

$Y5 = D\,S2\,\overline{S1}\,S0$

$Y6 = D\,S2\,S1\,\overline{S0}$

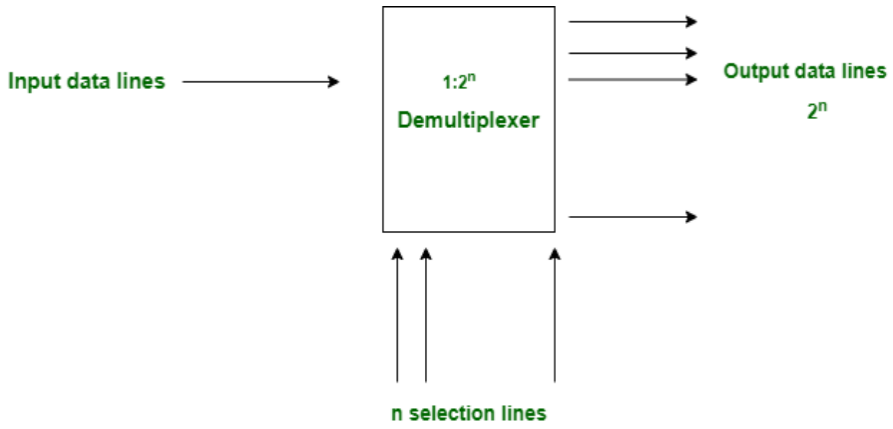
$Y7 = D\,S2\,S1\,S0$



SatheeshaKM

8) A. i) **Define De-Multiplexer and List any five applications.**

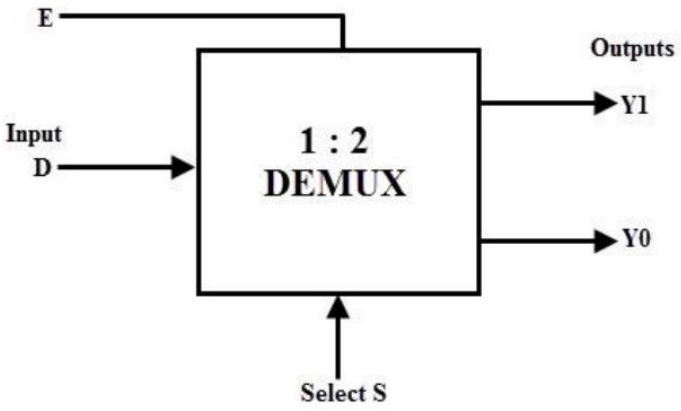
A demultiplexer is a combinational logic circuit that receives the information on a single input and transmits the same information over one of 2^n possible output lines.



Applications of Demux

1) A demultiplexer is used to connect a single source to multiple destinations. Demultiplexers are mainly used in the field of the communication system.
2) A serial to parallel converter is used for reconstructing the parallel data from the incoming serial data stream. In this technique, serial data from the incoming serial data stream is given as the input to the DEMUX at regular intervals. When all data signals have been stored. The output of the demux can be retrieved and read out in parallel.
3) Demultiplexer helps to store the output of the ALU in multiple registers and storage units in an ALU circuit. The output of the data the ALU is fed as data input to the DEMUX. Each output of the DEMUX is connected to the multiple registers which can be stored in the register.
4) Boolean function implementation
5) Security monitoring system (Mux used for selecting a particular surveillance camera at a time)
6) Extensively used in microprocessor or computer control systems

8) A. ii) Explain 1:2 De-Multiplexer with block diagram and truth table.

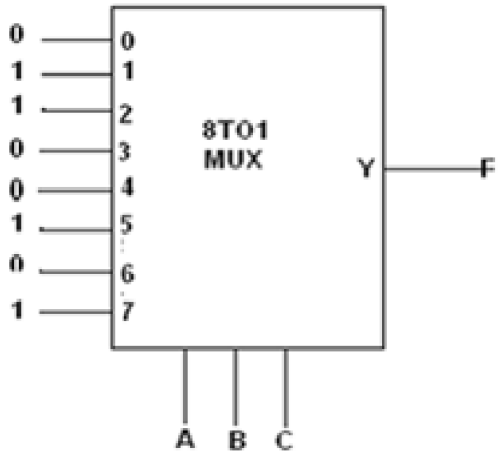


In this demux, there are only two possible ways to connect the input to output lines by using one select signal. When the select input is low, then the input is routed to Y0 When select input is high, then the input is routed to Y1. The truth table of 1 : 2 demux is shown below:

Select	Input	Outputs	
S	D	Y_1	Y_0
0	0	0	0
0	1	0	1
1	0	0	0
1	1	1	0

8) B. a) Design the function F (A, B, C) = Σ (1, 2, 5, 7) using 8 to 1 MUX

We can implement it using all three variables at selection lines. We put 1 on the min term lines which are present in functions and 0 on the rest.



A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

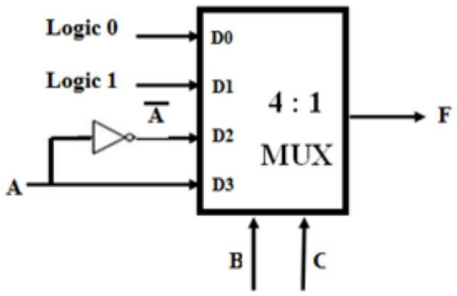
7) B. b) Design the function F (A, B, C) = Σ (1, 2, 5, 7) using 4 to 1 MUX

Solution: - Method 1:- Analyzing truth table

- As N=3, so we use $2^{N-1} = 2^2 = 4$ i.e 4 to 1 MUX.
- Suppose we have B, C on the selection lines. And when we have B=0, C=0 in the truth table, we see output of the function will be 0 hence we connect LOGIC 0 to Do input line.
- When B C = 0 1, then output of the function should be A'+ A = 1. Hence we connect LOGIC 1 to D1 line.
- When B C = 1 0, then output of the function should be A'. Hence we connect A' to D2 line.
- When BC=11, then output of the function should be A. Hence we connect A to D3 line.
- Hence we have the circuit as:

Truth Table

A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

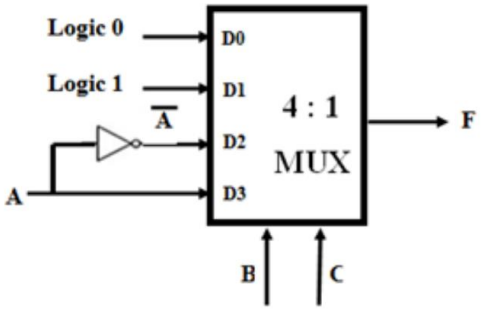


Satheesha K M

Method 2:- Using implementation table

- For 3 variable Logic function [A,B,C] , any 2 variables will be assigned for select inputs & 1 variable will be selected for i/p lines.
- List min terms of complimented variable selected in 1st row and min terms of un-complimented variable selected in 2nd row.
- Now encircle the min terms which are present in the function F(A, B, C)= Σ (1, 2, 5, 7) .
- No circled minterms in a column, put 0 for corresponding I/P line.
- For both circled minterms we put 1 on the line.
- If anyone minterm is circled in a column, then variable of that corresponding row is considered as input line.

	Do	D1	D2	D3
$\overline{A}$	0	(1)	(2)	3
A	4	(5)	6	(7)
	0	1	$\overline{A}$	A

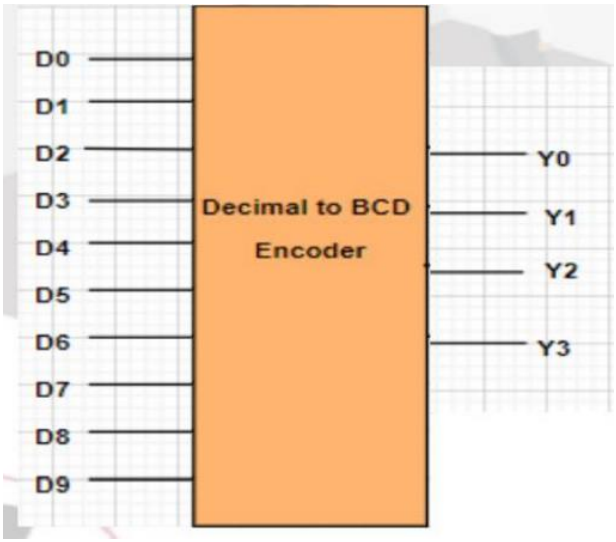


9. A. Sketch and Explain Decimal to BCD Encoder with Logic symbol, truth-table Boolean expression, logic diagram using gates.

Decimal to BCD Encoder

1) The decimal to binary encoder usually consists of 10 input lines and 4 output lines.
2) Each input line corresponds to the each decimal digit(0 to 9) and 4 outputs correspond to the BCD code.
3) This encoder accepts the decimal data as an input and encodes it to the BCD output which is available on the output lines.

Logic Symbol



Truth table

Input										Output			
D_9	D_8	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Y_3	Y_2	Y_1	Y_0
0	0	0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	1	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	1	0
0	0	0	0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	0	1	0	0	0	0	0	1	0	0
0	0	0	0	1	0	0	0	0	0	0	1	0	1
0	0	0	1	0	0	0	0	0	0	0	1	1	0
0	0	1	0	0	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0	1	0	0	1

Boolean Expressions:

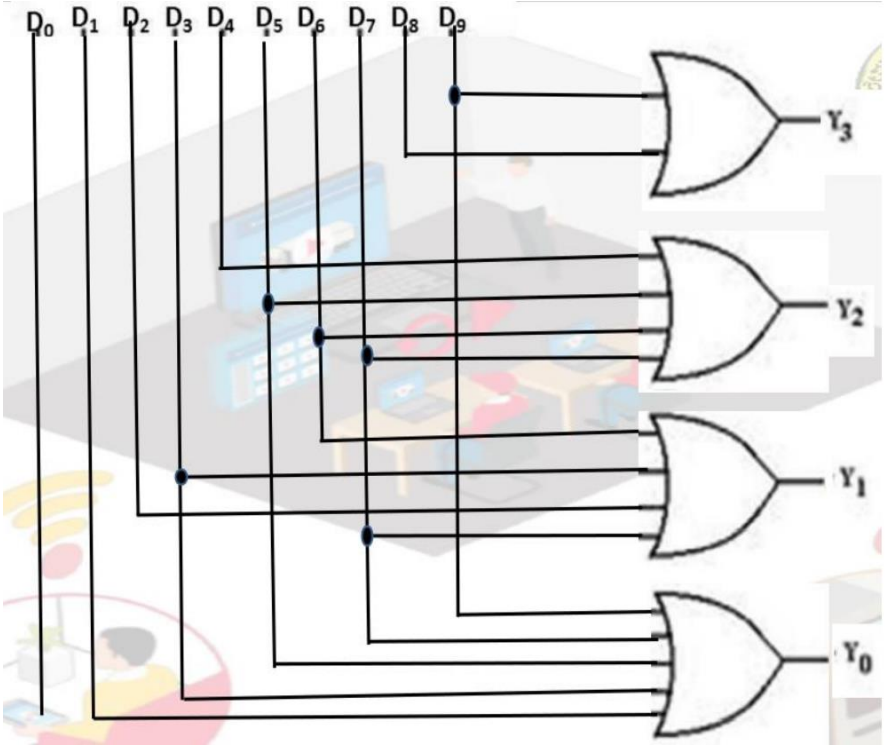
Y0 = D1 + D3 + D5 + D7 + D9

Y1 = D2 + D3 + D6 + D7

Y2 = D4 + D5 + D6 + D7

Y3 = D8 + D9

Logic Diagram



9. B. i) List IC classification based on scale of integration

IC classification based on scale of integration:

As the technology is improving day by day, the number of transistors incorporated in a single IC chip is also increasing. Depending uponthe number of transistors incorporated in a single chip the ICs are categorized in five groups. Namely

a. Small Scale Integration (SSI)where the number of transistors incorporated in a single IC chip is up to 100
b. Medium Scale Integration (MSI)where the number of transistors incorporated in a single IC chip is from 100 to 1000
c. Large Scale Integration (LSI)where the number of transistors incorporated in a single IC chip is from 1000 to 20,000.
d. Very Large-Scale Integration (VLSI)where the number of transistors incorporated in a single IC chip is from 20,000 to 10,00,000.
e. Ultra-Large-Scale Integration (ULSI)where the number of transistors incorporated in a single IC chip is from 10,00,000 to 1,00,00,000

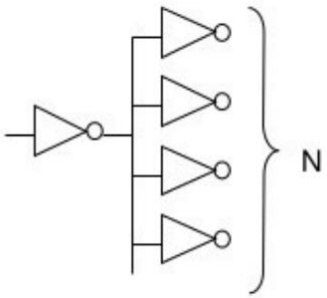
9. B. ii) Explain Fan-out with respect to logic family specification.

Fan out:

The fan-out is defined as the maximum number of inputs (load) that can be connected to the output of a gate without degrading the normal operation.

Fan Out is calculated from the amount of current available in the output of a gate and the amount of current needed in each input of the connecting gate. It is specified by manufacturer and is provided in the data sheet. Exceeding the specified maximum load may cause a malfunction because the circuit will not be able supply the demanded power.

For example, a logic gate having fan out of 4 can drive at the maximum 4 logic inputs Only



9. B. iii) Describe the features of TTL family.

Features and Merits of TTL

a. High speed of operation.
b. Low cost of manufacturing.
c. Requires only one supply voltage.
d. Lesser immune to noise when compared to ECL, but more than CMOS.
e. Fastest saturation, when compared to other logic families.
f. Low output impedance for high/low states.
g. Good fan out.
h. Immune to static noise
i. Compatible with other logic families.
j. Good driving/Interfacing.
k. Moderate packing density.

10. A. Sketch and Explain BCD-to-seven segment decoder (IC7447) with Logic circuit and truth-table.

BCD-to-7 segment decoder

In most of the practical applications seven segment displays are used to give visual indications (characters and numbers) of the output states of digital IC's such as decade counters, latches etc. Thus outputs are usually in 4 bit BCD (Binary Coded Decimal) form and are thus notavailable for driving seven segment displays.

So, the special BCD-to-seven segment decoder/driver ICs' are used to convert BCD signal into a form suitable for driving these displays. Seven segment decoder / driver is a digital circuit that can decode a digital input to the seven-segment format and simultaneously drive a 7-segment LED display using the decoded information. What that will be displayed on the 7-segment

display is the numerical equivalent of the input data. For example, a BCD to seven segment decoder driver can decode a 4 line BCD (binary coded decimal) to 8 line seven segment format and can drive the display using this information.Generally, it has 4 input lines and 8 output lines as shown in figure

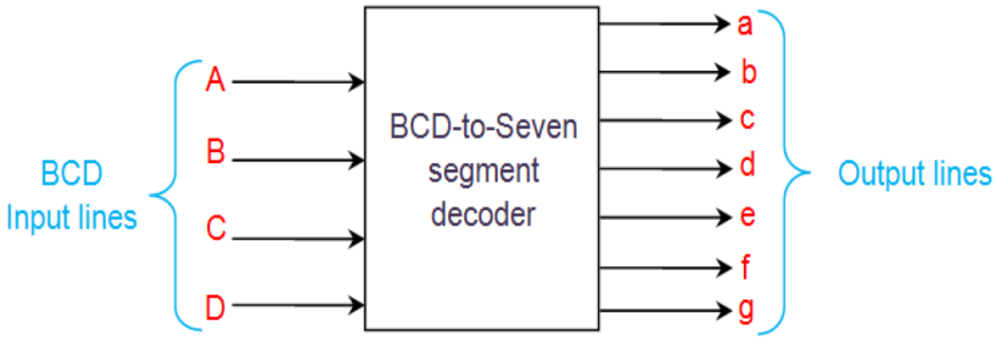


Fig 1. BCD-to-seven segment decoder

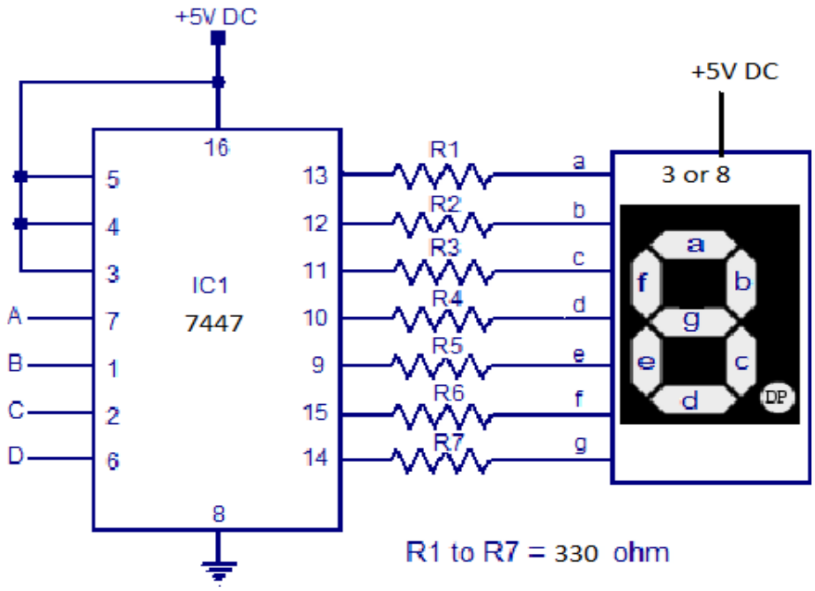
IC 7447 BCD-to-seven segment Decoder/Driver

IC 7447 is a BCD -to- seven segment Decoder which translates the 8421 BCD code to a code that lights the proper segments on the common -anode seven-segment LED display.

IC 7447 is one such IC with active low outputs. Seven segment displays make use of segments, and each segment contains and the seven LED are labelled from a to g. The digits from 0 to 9 can be displayed by forward biasing the different LEDS.

The IC is stand alone and requires no external components other than the LED current limiting resistors. The display used here must be a common anode type because the IC has active low outputs.

Logic Circuit



Common anode type '0' means ON

Truth Table

Decimal Digit	Input Lines				Output Lines								Display pattern
	D	C	B	A	a	b	c	d	e	f	g	dp	
0	0	0	0	0	0	0	0	0	0	0	1	1	
1	0	0	0	1	1	0	0	1	1	1	1	1	
2	0	0	1	0	0	0	1	0	0	1	0	1	
3	0	0	1	1	0	0	0	0	1	1	0	1	
4	0	1	0	0	1	0	0	1	1	0	0	1	
5	0	1	0	1	0	1	0	0	1	0	0	1	
6	0	1	1	0	0	1	0	0	0	0	0	1	
7	0	1	1	1	0	0	0	1	1	1	1	1	
8	1	0	0	0	0	0	0	0	0	0	0	1	
9	1	0	0	1	0	0	0	0	1	0	0	1	

10. B. i) List the features of CMOS Logic family.

Features and Merits of CMOS

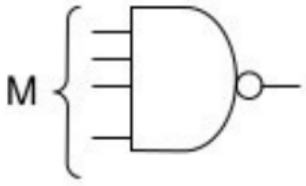
a. MOSFETS are active switching elements.
b. CMOS supports a very large fan-out, more than 50 transistors.
c. Low power consumption.
d. It has excellent noise immunity amongst all families.
e. Requires very less chip area and it has greater packing density.
f. Excellent temperature stability.
g. Works well over a wide range of temperature.
h. They are used in portable electronics.
i. Large logic swing.
j. Simpler and inexpensive to fabricate because of the reduction in the number of external connections.
k. Requires less chip area and it has greater packing density.
l. Higher reliability.

m. CMOS devices can work with a single power supply over a range of 3 to 15V (simple and inexpensive). The best compromise for speed, noise immunity, and overall performance is a supply voltage from 9V to 12V

10. B. ii) Explain Fan-in with respect to logic family specification.

Fan in:

The fan-in defined as the maximum number of inputs that a logic gate can accept. If number of inputs exceeds, the output will be undefined or incorrect. It is specified by manufacturer and is provided in the data sheet

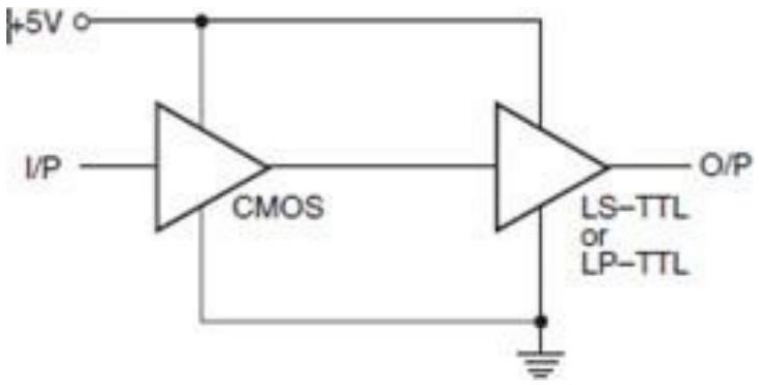


10. B. iii) Describe the interfacing between CMOS to TTL.

The first possible type of CMOS-to-TTL interface is the one where both ICs are operated from a common supply. The TTL family has a recommended supply voltage of 5 V, whereas the CMOS family devices can operate over a wide supply voltage range of 3–18 V. In the below figure both ICs operate from 5 V. As far as the voltage levels in the two logic states are concerned, the two have become compatible. The CMOS output has a V_{OH}(min.) of 4.95V (for V_{CC} =5 V) and a V_{OL}(max.) of 0.05 V, which is compatible with V_{IH}(min.) and V_{IL}(max.) requirements of approximately 2 and 0.8V respectively for TTL family devices. In fact, in a CMOS-to- TTL interface, with the two devices operating on the same V_{CC}, voltage level compatibility is always there. It is the current level compatibility that needs attention.

The below figure shows a CMOS-to-TTL interface with both devices operating from 5V supply and the CMOS IC driving a low-power TTL or a low-power Schottky TTL device.

Satheesha K M



Certificate

Certified that the model answers prepared by me for code no 20EC11T (Digital Electronics) are from prescribed textbooks and model answers and scheme of valuation prepared by me are correct.

SATHEESHA K M

LECTURER, E&C DEPT

KARNATAKA GOVT POLYTECHNIC,MANGALORE

Scrutinized by

(SHOBHARANI) (NETHRAVATHI H.R) Mallikarjun.R.R (DEVI D.T.)