I.C.E.™

Inspiring Creativity & Endeavour

IES | GATE | PSU

DIGITAL LOGIC

For

COMPUTER SCIENCE

DIGITAL LOGIC

SYLLABUS

Logic Functions, Minimization, Design and synthesis of combinational and sequential circuits ; Number representation and computer arithmetic (fixed and floating point)

ANALYSIS OF GATE PAPERS

Exam Year	1 Mark Ques.	2 Mark Ques.	Total
2003	1	4	9
2004	4	5	14
2005	4	4	12
2006	1	5	11
2007	3	5	13
2008	4	1	9
2009	2	-	2
2010	3	2	7
2011	2	3	8
2012	2	-	2
2013	3	-	3
2014 Set-1	2	1	4
2014 Set-2	3	1	5
2014 Set-3	2	2	6
2015 Set-1	1	2	5
2015 Set-2	1	2	5
2015 Set-3	-	3	6
2016 Set-1	3	2	7
2016 Set-2	3	-	3
2017 Set-1	2	1	4
2017 Set-2	3	4	11

CONTENTS

Topics		Page No
1.	**NUMBER SYSTEMS**	
1.1	Introduction	01
1.2	Binary Number System	01
1.3	Octal Number System	03
1.4	Hexadecimal Number System	04
1.5	Binary Codes	06
1.6	Complements	07
2.	**BOOLEAN ALGEBRA**	
2.1	Introduction	11
2.2	SOP & POS expressions	13
2.3	Karnaugh Map	14
3.	**LOGIC GATES**	
3.1	Introduction	17
3.2	Basic Gates	17
3.3	Universal Gates	18
3.4	Special Purpose Gates	19
3.5	Implement of SOP & POS	20
4.	**LOGIC FAMILIES**	
4.1	Introduction	22
4.2	Characteristics of Digital ICs	22
4.3	Transistor-Transistor Logic	24
4.4	Emitter-Coupled Logic	26
4.5	MOS Logic	27
4.6	CMOS Logic	28
4.7	COMPARISON BETWEEN LOGIC FAMILIES	29
5.	**COMBINATIONAL CIRCUITS**	
5.1	Introduction	30
5.2	Adders	30
5.3	Subtractors	32
5.4	Multiplexer	33
5.5	Demultiplexer	34
5.6	Decoder	34

5.7 Encoder 35

6. **SEQUENTIAL CIRCUITS**

6.1 Introduction 36
6.2 Flip-Flops 36
6.3 Registers 40
6.4 Counters 41

7. **CONVERTERS**

7.1 Introduction 45
7.2 Digital to Analog Converter 45
7.3 Binary-Weighted Resistors DAC 46
7.4 R-2R Ladder DAC 46
7.5 Analog to digital converters 46
7.6 Counter types ADC 47
7.7 Successive Approximation ADC 47
7.8 Dual Slope ADC 48
7.9 Flash ADC 49

8. **SEMICONDUCTOR MEMORIES**

8.1 Introduction 50
8.2 RAM 50
8.3 ROM 50
8.4 Programmable Logic Devices 51

9. **GATE QUESTIONS** 53

10. **ASSIGENMENT QUESTIONS** 93

1 NUMBER SYSTEMS

1.1 INTRODUCTION

Whenever we use numbers in an everyday way we use certain conventions. For example we all understand that the number 1234 is a combination of symbols which means one thousand, two hundreds, three tens and four units. We also accept that the symbols are chosen from a set of ten symbols: 0,1, 2, 3, 4, 5, 6, 7, 8 and 9. This is the decimal number system; it is part of the language of dealing with quantity.

A number has a base or radix. These terms both mean: how symbols or digits are used to express a number. A base 10 number has ten symbols; the base 10 number system is the decimal number system. Sometimes the base of a number is shown as a subscript: $(1234)_{10}$. Here the 10 is a subscript which indicates that the number is a base 10 number.

Note:

- In a number system, the digit (number) used cannot be equal to or greater than its base number. E.g. In base 10, the largest number that is used is 10-1=9.
- When we count in Base Ten, we count starting with zero and going up to nine in order 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 …Once we reach the last symbol, we create a new placement in front of the first and count that up. 8, 9, 10, 11, 12…19, 20… This continues when we run out of symbols for that placement. So, after 99, we go to 100.

1.1.1 POSITIONAL NUMBER SYSTEM

Positional numbering system uses a set of symbols. The value that each symbol represents, however, depends on its face value and its place value, the value associated with the position it occupies in the number. In other words, we have the following.

$$\text{Symbolvalue} = \text{Facevalue} \times \text{Placevalue}$$

e.g The symbol value of 2 in $(26)_{10}$ is $2 \times 10 = 20$

1.2 BINARY NUMBER SYSTEM

The binary number system is a numbering system that represents numeric values using two unique digits (0 and 1). This is also known as the base-2 number system. A number in binary form can be written as $(1011)_2$. In this representation, the first digit '1' is called most significant bit (MSB) & the last bit '1' is called least significant bit (LSB).

1.2.1 DECIMAL TO BINARY CONVERSION

1) For numbers greater than 1

- Write the decimal number as the dividend. Write the base of the destination system (in our case, "2" for binary) as the divisor on the left side.
- Write the integer answer (quotient) under the dividend, and write the remainder (0 or 1) to the right of the quotient.
- Continue downwards, dividing each new quotient by two and writing the remainders to the right of each dividend. Stop when the quotient is 0.
- Starting with the bottom remainder, read the sequence of remainders upwards to the top.

Example: Convert $(156)_{10}$ to binary (base-2).

Base	Quotient	Remainder
2	156	
2	78	0
2	39	0
2	19	1
2	9	1
2	4	1
2	2	0
2	1	0
2	0	1
2		

Writing the remainders from bottom to top 10011100 we get
$(156)_{10} = (10011100)_2$

Note: This method can be modified to convert from decimal to any base. The divisor is 2 because the desired destination is base 2 (binary). If the desired destination is a different base, replace the 2 in the method with the desired base. For example, if the desired destination is base 9, replace the 2 with 9. The final result will then be in the desired base.

2) For the numbers less than 1

- Multiply the decimal number by 2. The integer part of the result is kept aside as a carry.
- Again multiply the fraction part of the result until we get only integer part in the result (no fraction part).

Example: Convert $(0.75)_{10}$ into its binary equivalent.

Multiplication	Carry
0.125 × 2 = 0.25	0
0.25 × 2 = 0.5	0
0.5 × 2 = 1.0	1

Writing carry from top to bottom, we get $(0.75)_{10} = (.001)_2$.

1.2.2 BINARY TO DECIMAL CONVERSION

- Define a variable n whose value whose value 0, 1, 2, 3… on the left side of decimal point &−1, −2, −3, −4… on the right side of decimal point.
- Multiply each binary digit with 2^n by taking corresponding value of n & add all the multiplications.

Example: Convert $(10011100.001)_2$ to its decimal equivalent.

Solution:

10011100.001

n = 7 6 5 4 3 2 1 0 -1 -2 -3

Now,

$$(10011100.001)_2 = 1\times 2^7 + 0\times 2^6 + 0\times 2^5 + 1\times 2^4 + 1\times 2^3$$
$$+1\times 2^2 + 0\times 2^1 + 0\times 2^0 + 0\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3} = (156.125)_{10}$$

1.2.3 BINARY ADDITION

Rules for Binary Addition:.

1) 0+0=0
2) 0+1=1
3) 1+0=1
4) 1+1=10 (=decimal 2) first '1' will go as a carry
5) 1+1+1=11 (=decimal 3) first '1' will go as a carry

Example: Add $(1101)_2 + (0101)_2$

Solution:

$$\begin{array}{r} 1\ \ 1 \\ 1101 \\ +0101 \\ \hline 10010 \end{array}$$

$\therefore (1101)_2 + (0101)_2 = (10010)_2$

1.2.4 BINARY SUBTRACTION

Rules for Binary Subtraction:

1) 0-0=0
2) 0-1=1 (It is not possible to subtract 1 from 0 hence we take a borrow equal to the base of number system (for binary it is 2)).
3) 1-0=1
4) 1-0=0

Example: Subtract $(1100)_2 - (0011)_2$.

$$\begin{array}{r} 22 \\ 11\cancel{0}\cancel{0} \\ -0011 \\ 11 \\ \hline 1001 \end{array}$$

$\therefore (1100)_2 - (0011)_2 = (1001)_2$

1.2.5 BINARY MULTIPLICATION

Rules for Binary Multiplication:

1) 0×0=0
2) 0×1=0
3) 1×0=0
4) 1×1=1

Example: Multiply $(1100)_2 \times (101)_2$.

Solution:

$$\begin{array}{r} 1100 \\ \times 101 \\ \hline 1100 \\ +\ 00000 \\ +\ 110000 \\ \hline 111100 \end{array}$$

$\therefore (1100)_2 \times (101)_2 = (111100)_2$

1.2.6 BINARY DIVISION

Binary division is the repeated process of subtraction, just as in decimal division.

Example: Perform $(1100)_2 \div (100)_2$.

$$\begin{array}{l} 100\overline{)1100}\,(1 \\ \quad -100 \\ \hline \quad\ 0100\,(1 \\ \quad\ -100 \\ \hline \quad\ \ 0000 \end{array}$$

$\therefore (1100)_2 \div (100)_2 = (11)_2$

1.2.7 BINARY EQUIVALENTS

1) 1 Nybble (or nibble)=4 bits
2) 1 Byte=2 nybbles =8 bits
3) 1 Kilobyte (KB)=1024 bytes
4) 1Megabyte(MB)=1024kilobytes= 1,048,576 bytes
5) 1 Gigabyte (GB) =1024 megabytes = 1,073,741,824 bytes

1.3 OCTAL NUMBER SYSTEM

Octal is another number system with less symbols to use than our conventional number system. Octal is fancy for Base Eight meaning eight symbols are used to represent all the quantities. They are 0, 1, 2, 3, 4, 5, 6, and 7.
When we count up one from the 7, we need a new placement to represent what we call 8 since an 8 doesn't exist in Octal. So, after 7 is 10. A number can be represented in Octal as $(526)_8$.

1.3.1 DECIMAL TO OCTAL CONVERSION

The procedure to convert decimal to octal is exactly same as to convert decimal to binary.

Example: Convert $(63.625)_8$ into decimal.

Base	Quotient	Remainder
8	63	
8	7	7
	0	7

Writing the remainder from bottom to top, the octal conversion of $(63)_{10}$ is $(77)_8$.

Multiplication	Carry
0.625 × 8 = 5.0	5

$\therefore (63.625)_{10} = (77.5)_8$

1.3.2 OCTAL TO DECIMAL CONVERSION

The procedure to convert octal to decimal is exactly same as to convert binary to decimal.

Example: Convert $(77.5)_8$ to decimal.
77.5
N = 10-1
Now,

$(77.5)_8 = 7\times8^1 + 7\times8^0 + 5\times8^{-1}$

$= (63.625)_{10}$

1.3.3 OCTAL TO BINARY CONVERSION

An octal number can be converted into binary by representing each octal digit into its bit binary equivalent.

Example: Convert $(24.53)_8$ into binary.
Solution:

$$\begin{array}{cccc} 2 & 4 & .\ 5 & 3 \\ \swarrow & \downarrow & \downarrow & \searrow \\ 010 & 100 & 101 & 011 \end{array}$$

$\therefore (24.53)_8 = (010100.101011)_2$

Note: A binary number can be converted into octal by grouping 3 binary bits & converting each group into its octal equivalent.

Example: Convert $(10001.0101)_2$ into octal.

$$\begin{array}{cccc} 010 & 001\ . & 010 & 100 \\ \searrow & \downarrow & \downarrow & \swarrow \\ 2 & 1\ . & 2 & 4 \end{array}$$

$\therefore (10001.0101)_2 = (21.24)_8$

1.3.4 OCTAL ADDITION

The addition of octal numbers is not difficult provided you remember that anytime the sum of two digits exceeds 7, a carry is produced.

Example: Perform

i) $(5)_8 + (1)_8$

ii) $(5)_8 + (6)_8$

iii) $(23)_8 + (11)_8$

iv) $(23)_8 + (64)_8$

Solution:

i)

$$\begin{array}{r} 5 \\ +1 \\ \hline 6 \end{array}$$

$\therefore (5)_8 + (1)_8 = (6)_8$

ii)

$$\begin{array}{r} 5 \\ +6 \\ \hline 11 \end{array}$$

As we cannot represent a number greater than 8, 11 is not a valid octal number.

$11 = 1(\text{carry}) \times 8(\text{base}) + 3$

Hence 1 will go to carry & the addition will be 13 i.e. $(5)_8 + (6)_8 = (13)_8$

iii)

$$\begin{array}{r} 23 \\ +11 \\ \hline 34 \end{array}$$

$\therefore (23)_8 + (11)_8 = (34)_8$

iv)

$$\begin{array}{r} 23 \\ +64 \\ \hline 107 \end{array}$$

Here, $6+2=8$ is not a valid octal number.

$8 = 1 \times 8 + 0$ hence 1 will go to carry.

1.3.5 OCTAL SUBTRACTION

The subtraction in octal follow the same rules as in case of decimal. The only difference is that when we are subtracting a larger number from a smaller one, we have to take 8 as borrow instead of 10 as in case of decimal number system.

Example: Perform $(46)_8 - (7)_8$.

Solution:

$$\begin{array}{rr} & 8+6 \\ 3 & =14 \\ \cancel{4} & \cancel{6} \\ - & 7 \\ \hline 3 & 7 \end{array}$$

In the octal example $(7)_8$ cannot be subtracted from $(6)_8$, so you must borrow from the 4. Reduce the 4 by 1 and add base (i.e.8) to the $(6)_8$. By subtracting $(7)_8$ from 14 you get a difference of $(7)_8$. Write this number in the difference line and bring down the 3.

$\therefore (46)_8 - (7)_8 = (37)_8$

1.4 HEXADECIMAL NUMBER SYSTEM

The hexadecimal system is Base Sixteen. As its base implies, this number system uses sixteen symbols to represent numbers. Unlike binary and octal, hexadecimal has six additional symbols that it uses beyond the conventional ones found in decimal. But what comes after 9? 10 is not a single digit but two… Fortunately, the convention is that once additional symbols are needed beyond the normal ten, letters are to be used. So, in hexadecimal, the total list of symbols to use is 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F.

1.4.1 DECIMAL TO HEXADECIMAL CONVERSION

The procedure to convert decimal to hexadecimal is exactly same as to convert decimal to binary only we have replace base 2 with 16.

Example: Convert $(55.03125)_{10}$ to its hexadecimal equivalent.

Solution:

Base	Quotient	Remainder
16	55	
16	3	7
	0	3

Writing remainders from bottom to top we get,

$(55)_{10} = (37)_{16}$

Now,

Multiplication	carry
0.03125×16=0.5	0
0.5×16=8	8

Writing carry from top to bottom we get,

$(0.03125)_{10} = (0.08)_{16}$

$\therefore (55.03125)_{10} = (37.08)_{16}$

1.4.2 HEXADECIMAL TO DECIMAL CONVERSION

Like binary to decimal conversion here also define a variable n & conversion can be done by following same procedure.

Example: Convert $(A6)_{16}$ to decimal.

Solution:

A 6

n = 1 0

Now,

$(A6)_{16} = 10\times16^1 + 6\times16^0 = (166)_{10}$

1.4.3 HEXADECIMAL TO BINARY CONVERSION

A number in base 16 can be converted into base 2 by representing each hexadecimal bit into its 4 bit binary equivalent.

Example: Convert $(2FD.B61)_{16}$ into binary.

Solution:

2 F D . B 6 1

0010 1111 1101 1011 0110 0001

$\therefore (2FD.B61)_{16} = (001011111101101101100001)_2$

1.4.4 HEXADECIMAL ADDITION

While adding two hexadecimal numbers if the result exceeds 15, a carry is generated.

Example: Perform

i) $(5)_{16} + (1)_{16}$

ii) $(5)_{16} + (6)_{16}$

iii) $(A)_{16} + (F)_{16}$

iv) $(CC)_{16} + (58)_{16}$

Solution:

i)

$$\begin{array}{r} 5 \\ +1 \\ \hline 6 \end{array}$$

ii)

$$\begin{array}{r} 5 \\ +6 \\ \hline B \end{array}$$

iii)

$$\begin{array}{r} A \\ +F \\ \hline 19 \end{array}$$

Here $A(10) + F(15) = 25$ this is not a valid hexadecimal number.

As $25 = 1(\text{carry})\times16(\text{base}) + 9$

$\therefore (A)_{16} + (F)_{16} = (19)_{16}$

iv)

$$\begin{array}{r} {}^{1} \\ CC \\ +_{1}58 \\ \hline 124 \end{array}$$

$C + 8 = 20 = 1(\text{carry})\times16(\text{base}) + 4$

$C + 5 + 1 = 18 = 1(\text{carry})\times16(\text{base}) + 2$

1.4.5 HEXADECIMAL SUBTRACTION

The subtraction in hexadecimal follows the same rules as in case of decimal. The only difference is that when we are subtracting a larger number from a smaller one, we have to take 16 as borrow instead of 10 as in case of decimal number system.

Example: Perform $(46)_{16} - (D)_{16}$.

Solution:

$$\begin{array}{rr} & 16+6 \\ 3 & =22 \\ \cancel{4} & \cancel{6} \\ - & D \\ \hline 3 & 9 \end{array}$$

In the octal example $(D)_{16}$ cannot be subtracted from $(6)_{16}$, so we must borrow from the 4. Reduce the 4 by 1 and add base (i.e. 16) to the $(6)_{16}$. By subtracting $(D)_{16}$ from 22 we get a difference of $(9)_{16}$. Write this number in the difference line and bring down the 3.

$\therefore (46)_{16} - (D)_{16} = (39)_{16}$

Note:

GATE QUESTIONS

Topics **Page No**

1.	LOGIC FUNCTIONS AND MINIMIZATION	54
2.	COMBINATIONAL CIRCUITS	69
3.	SEQUENTIAL CIRCUITS	76
4.	NUMBER SYSTEMS	84

1 LOGIC FUNCTIONS AND MINIMIZATION

Q.1 The simultaneous equations on the Boolean variables x, y, z and w,

$x+y+z=1$

$xy=0$

$xz+w=1$

$xy+\overline{z}\overline{w}=0$

Have the following solution for x, y, z and w, respectively

a) 0100 b) 1101

c) 1011 d) 1000

[GATE-2000]

Q.2 Given the following Karnaugh map, which one of the following represents the minimal Sum-Of-Products of the map?

yz \ wx	00	01	11	10
00	0	x	0	x
01	x	1	x	1
11	0	x	1	0
10	0	1	x	0

a) $xy+y'z$

b) $wx'y'+xy+xz$

c) $w'x+y'z+xy$

d) $xz+y$

[GATE-2001]

Q.3 Minimum sum of products expression for f(w, x, y, z) shown in Karnaugh map below is

yz \ wx	00	01	11	10
00	0	1	1	0
01	x	0	0	1
11	x	0	0	1
10	0	1	1	x

a) $xz+y'z$

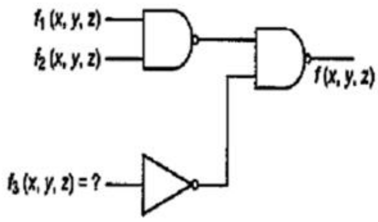
b) $xz'+zx'$

c) $x'y+zx'$

d) None of these

[GATE-2002]

Q.4 Consider the following logic circuit whose inputs are functions f_1, f_2, f_3 and output is f:



Give that

$f_1(x,y,z)=\sum(0,1,3,5)$

$f_2(x,y,z)=\sum(6,7)$, and

$f(x,y,z)=\sum(1,4,5)$ f_3 is

a) $\sum(1,4,5)$ b) $\sum(6,7)$

c) $\sum(0,1,3,5)$ d) None of the

[GATE-2002]

Q.5 Let f(A, B) = A' + B. Simplified expression for function $f(f(x+y,y),z)$ is

a) x' + z b) xyz

c) xy' + z d) none of these

[GATE-2002]

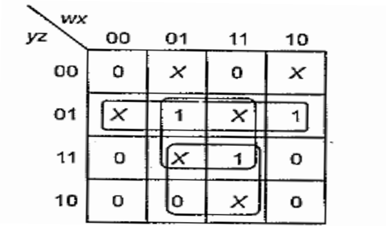
Q.6 The literal count of a Boolean expression is the sum of the number of times each literal appears in the expression. For example, he literal counts of the $(xy + xz')$ is 4. What are the minimum possible literal counts of the product-of-sums and sum-of-products representations respectively of the function given by the following Karnaugh map? Here, x denotes don't care.

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

EXPLANATIONS

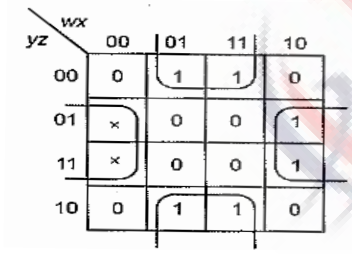
Q.1 **(c)**
This is solved by heuristics and by not lengthy computations.
Since, xy = 0 and from the last equation, we get z’w’ = 0
Therefore, only option (c) satisfies both these, i.e., xy = 0 and z’w’ = 0.

Q.2 **(a)**
The solution is obtained by the following:



By this minimal sum of product of the map comes out to be XY + Y’Z

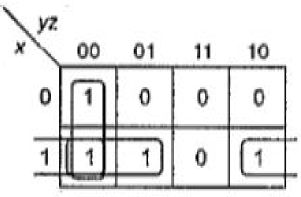
Q.3 **(b)**
Lets elaborate the given K map



With the described Kmap, following is concluded.
For minterm (SOP), f = xz’ + zx’

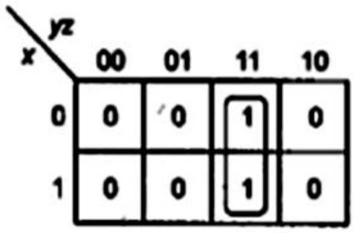
Q.4 **(a)**
As given,
We need to create 3 K maps for
f_1 (x, y, z), f_2 (x, y, z), f_3 (x, y, z)

1. Kmap for f_1 (x, y, z)



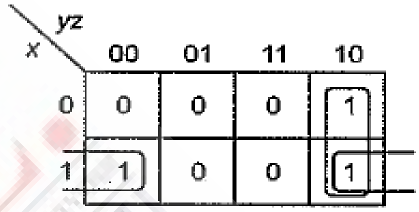
f_1 (x, y, z) = $\bar{y}z + x\bar{y} + x\bar{z}$

2. Kmap for f_2 (x, y, z)



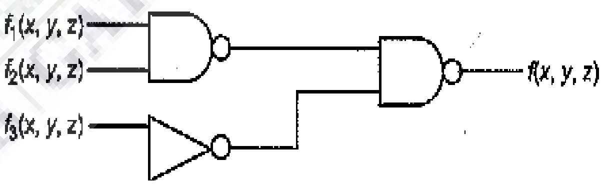
f_2 (x, y, z) = yz

3. Kmap for f_3 (x, y, z)



f_3 (x, y, z) = $\bar{y}z + x\bar{z}$

Considering the inputs and the output,



$\overline{\overline{f_1 f_2} . \bar{f_3}}$ = f(x, y, z)

$f_1 f_2 + f_3$ = f(x, y, z)

$\Rightarrow \quad f_3 = f(x, y, z) - f_1 f_2$

$= \bar{y}z + x\bar{z} - (\bar{y}z + x\bar{y} + x\bar{z})(yz)$

$= \bar{y}z + x\bar{z} - 0 = \bar{y}z + x\bar{z}$

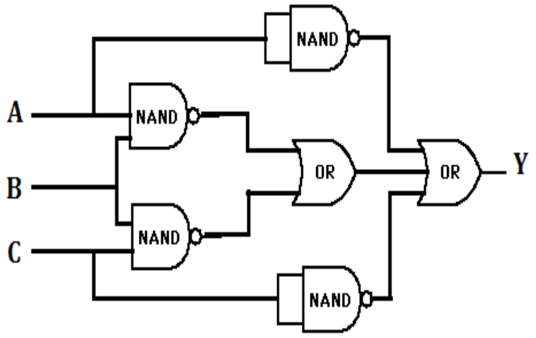
$\Rightarrow f_3 = \sum 1, 4, 5$

Q.5 **(c)**
Since, f(A, B) = A’ + B,
f(f(x + y, y), z) = f’(x + y, y) + z
= ((x + y)’ + y)’ + z
= (x + y) . y’ + z
= xy’ + z

Q.6 **(a)**

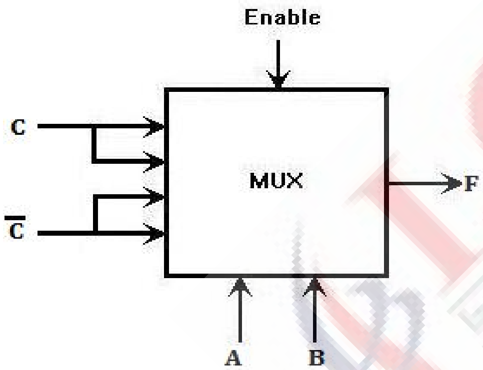
ASSIGNMENT QUESTIONS

Q.1 For the logic circuit shown in figure below, the output Y is equal to



a) $\overline{ABC}$ b) $\overline{A}+\overline{B}+\overline{C}$
c) $\overline{AB}+\overline{BC}+\overline{A}+\overline{C}$ d) $\overline{AB}+\overline{BC}$

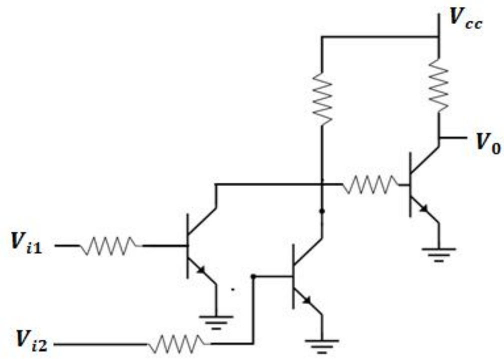
Q.2 The logic realized by the circuit shown in figure below, is



a) $F = A \odot C$ b) $F = A \oplus C$
c) $F = B \odot C$ d) $F = B \oplus C$

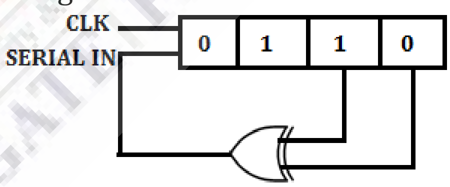
Q.3 Choose the correct statement (s) from the following
a) PROM contains a programmable AND array and a fixed OR array.
b) PLA contains a fixed AND array and a programmable OR array.
c) PROM contains a fixed AND array and programmable OR array
d) None of the above.

Q.4 In the figure shown below the circuit of a gate in the Resistor Transistor Logic (RTL) family. The circuit represents



a) NAND b) AND
c) NOR d)OR

Q.5 The initial contents of the 4-bit serial in parallel out, right shift, Shift Register shown in figure below, is 0110. After three clock pulses are applied, the contents of the Shift Register will be



a)0000 b)0101
c) 1010 d) 1111

Q.6 The binary representation of 5.375 is
a) 111.1011 b) 101.1101
c) 101.011 d) 111.001

Q.7 Dual slope integration type Analog-to-Digital converters provide
a) Higher speeds compared to all other types of A / D converters
b) Very good accuracy without putting extreme requirements on component stability
c) Poor rejection of power supply hums
d) Better resolution compared to all other types of A/D converters for the same number of bits.

Q.8 Data in the serial form can be converted into parallel form by using