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

Internal Assessment Test 1 – December 2021

Sub:	Analog and Digital Electronics					Sub Code:	18CS33	Branch:	ISE	
Date:	17/12/2021	Duration:	90 min's	Max Marks:	50	Sem/Sec:	III / A,B and C		OBE	
	Answer any FIVE FULL Questions							MARKS	CO	RBT
1	Define Implicants, Combinational Circuits. Explain prime and essential prime implicants with example. Find the minterm & maxterm expansion of F(a,b,c,d)=a'(b'+d)+acd'. Show that a'c + b'c' + ab = a'b' + bc + ac'							10	CO3	L2
2	What is K-map & explain limitations of K-map? Solve F(A,B,C,D) = ΠM(0,3,4,7,810,12,14).ΠD(2,6) using K map to get minimum POS expression and implement using basic, NAND only and NOR only gates.							10	CO3	L2
3	What is Quine- McCuskey Method? Find the minimum SOP for the function. F (A, B, C, D) = Σ m(0,4,5,10,11,13,15) + Σ d(1, 10, 15) using Quine McCuskey method.							10	CO3	L1
4	Consider the following logic function F(A, B, C, D) = Σ m(0,4,5,10,11,13,14,15). a) Find two different minimum circuits which implements F using AND and OR gates. Identify two hazards in each circuit. b) Find an AND-OR circuit for F which as no hazards. c) Find an OR- AND circuit for F which as no hazards.							10	CO3	L1
5	What is Multiplexer? Build 8: 1 Mux using basic gates.							10	CO3	L3
6	What is Buffer? Explain different three state buffers. Construct 8-to-1 MUX using two 4-to-1 MUXes, two three-state buffers, and one inverter.							10	CO3	L3

Faculty Signature CCI Signature HOD Signature

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

Internal Assessment Test 1 – December 2021

Sub:	Analog and Digital Electronics					Sub Code:	18CS33	Branch:	ISE	
Date:	17/12/2021	Duration:	90 min's	Max Marks:	50	Sem/Sec:	III / A,B and C		OBE	
	Answer any FIVE FULL Questions							MARKS	CO	RBT
1	Define Implicants, Combinational Circuits. Explain prime and essential prime implicants with example. Find the minterm & maxterm expansion of F(a,b,c,d)=a'(b'+d)+acd'. Show that a'c + b'c' + ab = a'b' + bc + ac'							10	CO3	L2
2	What is K-map & explain limitations of K-map? Solve F(A,B,C,D) = ΠM(0,3,4,7,810,12,14).ΠD(2,6) using K map to get minimum POS expression and implement using basic, NAND only and NOR only gates.							10	CO3	L2
3	What is Quine- McCuskey Method? Find the minimum SOP for the function. F (A, B, C, D) = Σ m(0,4,5,10,11,13,15) + Σ d(1, 10, 15) using Quine McCuskey method.							10	CO3	L1
4	Consider the following logic function F(A, B, C, D) = Σ m(0,4,5,10,11,13,14,15). a) Find two different minimum circuits which implements F using AND and OR gates. Identify two hazards in each circuit. b) Find an AND-OR circuit for F which as no hazards. c) Find an OR- AND circuit for F which as no hazards.							10	CO3	L1
5	What is Multiplexer? Build 8: 1 Mux using basic gates.							10	CO3	L3
6	What is Buffer? Explain different three state buffers. Construct 8-to-1 MUX using two 4-to-1 MUXes, two three-state buffers, and one inverter.							10	CO3	L3

Faculty Signature CCI Signature HOD Signature

Scheme of Evaluation

Internal Assessment Test 1 – December 2021

Sub:	Analog and Digital Electronics							Code:	18CS33
Date:	17/12/2021	Duration:	90mins	Max Marks:	50	Sem:	III	Branch:	ISE

Note: **Answer Any five full questions.**

Question No.		Description	Marks Distribution		Max Marks
1		**Implicants**: A group of one or more 1's which are adjacent a group of one or more 1s which are adjacent and can be combined on a Karnaugh Map is called an **implicants**. **Combinational Circuit:** The output of the combinational circuit depends on the values at the input at any given time. **Definition of Prime and Essential Prime Implicants:** **Problem Solution**	1M 1M 2M 6M	10M	10M
2		***Karnaugh map/K map*** is a method simplifying and manipulating switching functions. K map method is faster and easier to apply than other simplification methods. **Limitations of K map:** ➢ Complexity of **K-map** simplification process increases with the increase in the number of variables ➢ K map is manual technique and simplification process heavily depends on the human ability. **Solving the function using K map** **Logic circuit diagram using NAND and NOR Gates**	1M 2M 3M 4M	10M	10M
3		The Quine-McCluskey method reduces the minterm expansion (standard sum-of-products form) of a function to obtain a minimum sum of products. **Steps** **Prime Implicate Chart** **Final expression**	2M 5M 3M	10M	10M

4		**Finding Static 1 and Static 0 Hazard** **Hazard free equation in terms of AND-OR** **Hazard free equation in terms of OR - AND**	4M 2M 2M	10M	10M
5		The multiplexer is a device that has multiple inputs and single line output. The select lines determine which input is connected to the output. **Implementation of 8:1 MUX using basic gates** **Block Diagram** **Truth Table** **Expression** **Logic Diagram**	2M 8M	10M	10M
6		A **buffer** has only a single input and a single output with behavior that is the opposite of an NOT gate. It simply passes its input, unchanged, to its output. **Different types of three state buffers** **Implementing 8:1 MUX using 4:1 MUX and Buffers**	1M 4M 5M	10M	10M

Scheme Of Evaluation Internal Assessment Test 1 – NOV 2021

Sub:	Analog and Digital Electronics							Code:	18CS33
Date:	17/12/2021	Duration:	90mins	Max Marks:	50	Sem:	III	Branch:	ISE

Note: **Answer Any full five questions**

Q. 1 Define Implicants, Combinational Circuits. Explain prime and essential prime implicants with example. Find the minterm & maxterm expansion of F(a,b,c,d)=a'(b'+d)+acd'. Show that a'c + b'c' + ab = a'b' + bc + ac'.

Implicants:

A group of one or more 1's which are adjacent a group of one or more 1s which are adjacent and can be combined on a Karnaugh Map is called an **implicants**.

Combinational Circuits:

Combinational Circuits are circuits made up of different types of logic gates. The output of the combinational circuit depends on the values at the input at any given time.

Prime and Essential Prime Implicants:

A group of square or rectangle made up of bunch of adjacent minterms which is allowed by definition of K-Map are **called prime** implicants **(PI).** Consider a Boolean function, F = AB + ABC + BC. Implicants. are AB, ABC and BC. Prime implicants.

Essential prime implicants (EPI) are those prime implicants which always appear in final solution.

Duplicate terms have been crossed out, because $X + X = X$. This expression can then be converted to decimal notation:

$$f = a'b'c'd' + a'b'c'd + a'b'cd' + a'b'cd + a'bc'd + a'bcd + abcd' + ab'cd'$$
$$0\,0\,0\,0 \quad 0\,0\,0\,1 \quad 0\,0\,1\,0 \quad 0\,0\,1\,1 \quad 0\,1\,0\,1 \quad 0\,1\,1\,1 \quad 1\,1\,1\,0 \quad 1\,0\,1\,0$$
$$f = \Sigma\, m(0, 1, 2, 3, 5, 7, 10, 14)$$

The maxterm expansion for f can then be obtained by listing the decimal integers (in the range 0 to 15) which do not correspond to minterms of f:

$$f = \Pi\, M(4, 6, 8, 9, 11, 12, 13, 15)$$

Show that $a'c + b'c' + ab = a'b' + bc + ac'$.

We will find the minterm expansion of each side by supplying the missing variables. For the left side,

$$a'c(b + b') + b'c'(a + a') + ab(c + c')$$
$$= a'bc + a'b'c + ab'c' + a'b'c' + abc + abc'$$
$$= m_3 + m_1 + m_4 + m_0 + m_7 + m_6$$

For the right side,

$$a'b'(c + c') + bc(a + a') + ac'(b + b')$$
$$= a'b'c + a'b'c + abc + a'bc + abc' + ab'c'$$
$$= m_1 + m_0 + m_7 + m_3 + m_6 + m_4$$

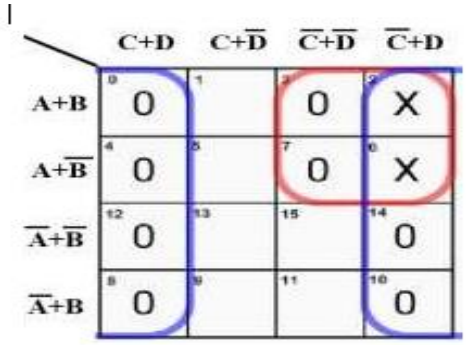
Because the two minterm expansions are the same, the equation is valid.

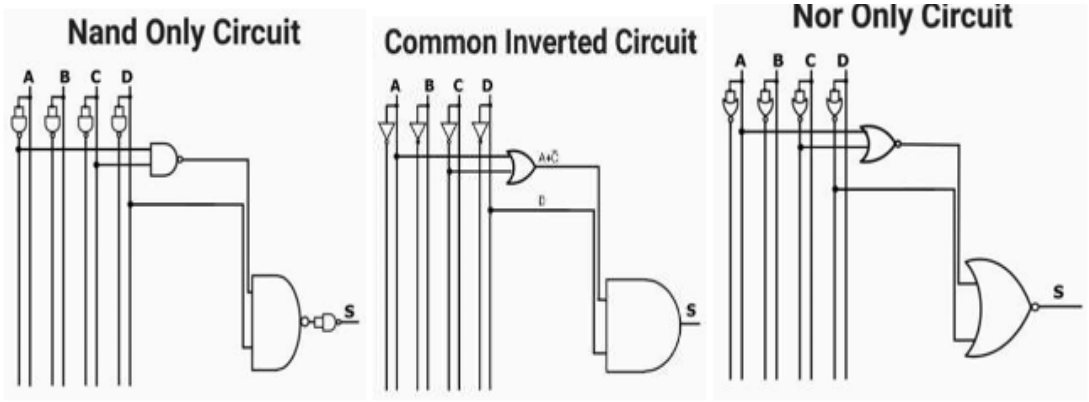
Q.2 What is K-map & explain limitations of K-map? Solve F(A,B,C,D) = ΠM(0,3,4,7,810,12,14).ΠD(2,6) using K map to get minimum POS expression and implement using basic, NAND only and NOR only gates.

Karnaugh map/K map is a method simplifying and manipulating switching functions. K map method is faster and easier to apply than other simplification methods.

Limitations of K map:

- Complexity of **K-map** simplification process increases with the increase in the number of variables
- K map is manual technique and simplification process heavily depends on the human ability.





Q.3 What is Quine- McCuskey Method? Find the minimum SOP for the function. F (A, B, C, D) = Σ m(0,4,5,10,11,13,15) + Σ d(1, 10, 15) using Quine McCuskey method.

The Quine-McCluskey method provides a systematic simplification procedure which can be readily programmed for a digital computer. The Quine-McCluskey method reduces the minterm expansion (standard sum-of-products form) of a function to obtain a minimum sum of products.

Petrick's method is a technique for determining all minimum sum-of-products solutions from a prime implicant chart

F(A, B, C, D) = Σ m(2, 3, 7, 9, 11, 13) + Σ d(1, 10, 15) using Quine McCuskey method.

Q. 3: QM - Mtd:

$F(A,B,C,D) = \Sigma m(0,4,5,10,11,13,15) + \Sigma d(1,10,15)$

Sop: ?

Solution:

Stage - I		Stage II		Stage - III	
	ABCD		abcd		
0	0000 ✓	(0,1) ✓	000 –	(0,1,4,5)	0 – 0 –
1	0001 ✓	(0,4) ✓	0 – 00	(0,4,1,5)	0 – 0 – ✗
4	0100 ✓	(1,5) ✓	0 – 01		
5	0101 ✓	(4,5) ✓	010 –		
10	1010 ✓	(5,13)	– 101		
11	1011 ✓	(10,11)	101 –		
13	1101 ✓	(11,15)	1 – 11		
15	1111 ✓	(13,15)	11 – 1		

	0,	4,	5,	10,	11,	13,	15
(5,13) $b\bar{c}d$			×	⊗		×	
(10,11) $a\bar{b}c$ ✓				×	×		
(11,15) acd					×		×
(13,15) abc ✓						×	×
(0,1,4,5) $\bar{a}\bar{c}$ ✓	×	⊗	×				

PI = $b\bar{c}d$, $a\bar{b}c$, acd, abc, $\bar{a}\bar{c}$

EPI = $F = a\bar{b}c + abc + \bar{a}\bar{c}$ Sop.

EPI = $a\bar{b}c$, abc, $\bar{a}\bar{c}$

PI = $b\bar{c}d$, $a\bar{b}c$, acd, abc, $\bar{a}\bar{c}$

Q.4 Consider the following logic function F(A, B, C, D) = Σ m(0,4,5,10,11,13,14,15).

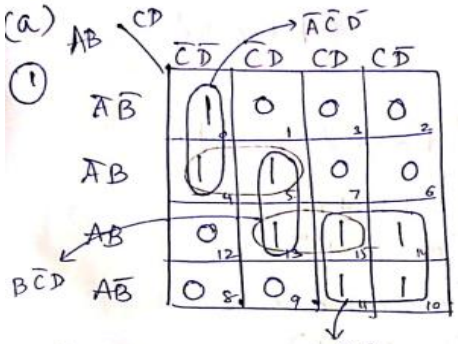
a) Find two different minimum circuits which implements F using AND and OR gates. Identify two hazards in each circuit.

b) Find an AND-OR circuit for F which as no hazards.

c) Find an OR- AND circuit for F which as no hazards.

$F(A,B,C,D) = \Sigma m(0,4,5,10,11,13,14,15)$

(a)



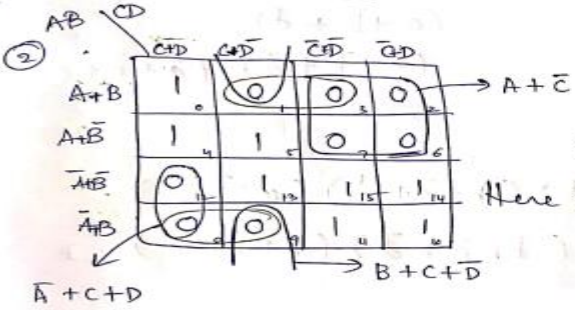
1st minimum solution is

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

The hazards are $\bar{A}B\bar{C}$ and ABD.

↳ Static-1-hazard.

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



2nd minimum solution is

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

Here the hazards are $(A+B+\bar{D})$ and $(\bar{A}+B+C)$

↳ Static-0-hazard.

(b) AND-OR ckt for F which has no hazards:

$F = AC + B\bar{C}D + \bar{A}\bar{C}\bar{D} + \bar{A}B\bar{C} + ABD$

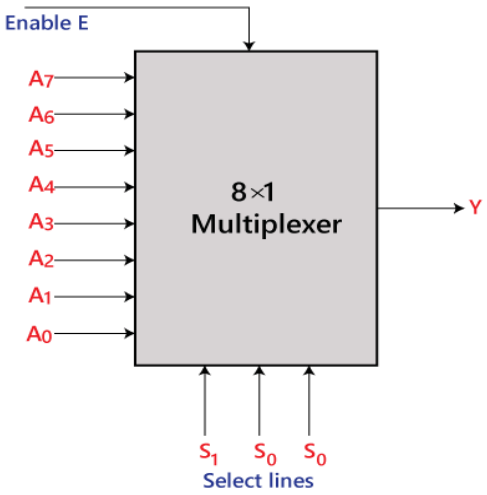
(c) OR-AND ckt for F which has no hazards:

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

Q.5 What is Multiplexer? Build 8: 1 Mux using basic gates.

The multiplexer is a device that has multiple inputs and single line output. The select lines determine which input is connected to the output.

Block Diagram:



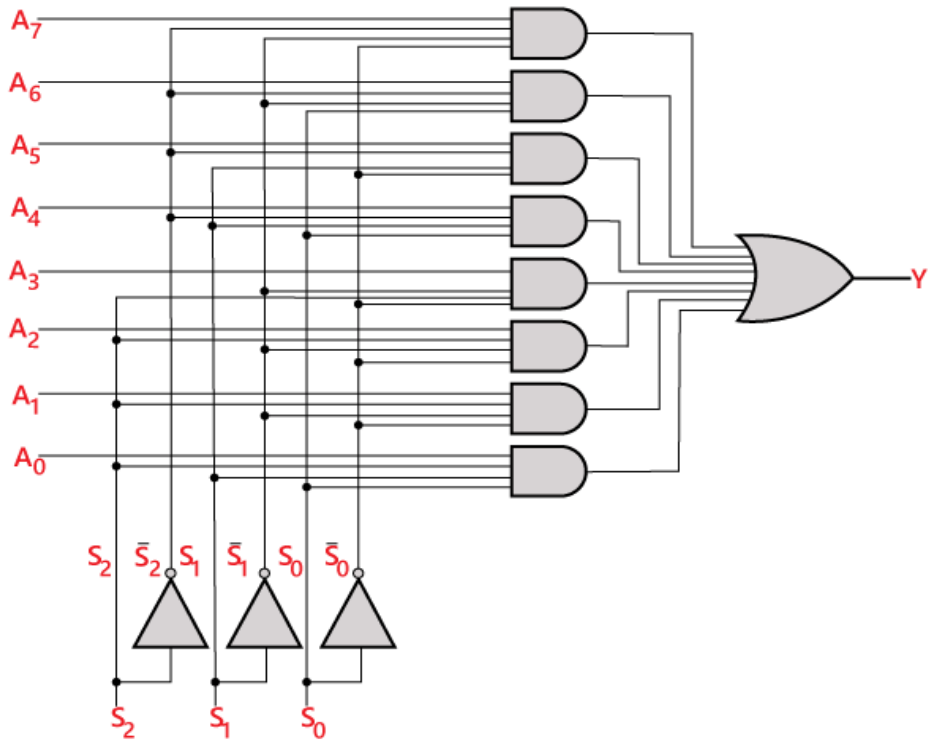
Truth Table:

INPUTS			Output
S_2	S_1	S_0	Y
0	0	0	A_0
0	0	1	A_1
0	1	0	A_2
0	1	1	A_3
1	0	0	A_4
1	0	1	A_5
1	1	0	A_6
1	1	1	A_7

The logical expression of the term Y is as follows:

$Y=S_0'.S_1'.S_2'.A_0+S_0.S_1'.S_2'.A_1+S_0'.S_1.S_2'.A_2+S_0.S_1.S_2'.A_3+S_0'.S_1'.S_2\ A_4+S_0.S_1'.S_2\ A_5+S_0'.S_1.S_2\ .A_6+S_0.S_1.S_3.A_7$

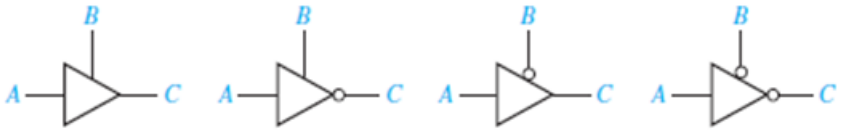
Logical circuit of the above expression is given below:



Q 6. What is Buffer? Explain different three state buffers. Construct 8-to-1 MUX using two 4-to-1 MUXes, two three-state buffers, and one inverter.

A **buffer** has only a single input and a single output with behavior that is the opposite of an NOT gate. It simply passes its input, unchanged, to its output.

The following Figure shows the truth tables for four types of three-state buffers.



B	A	C
0	0	Z
0	1	Z
1	0	0
1	1	1

(a)

B	A	C
0	0	Z
0	1	Z
1	0	1
1	1	0

(b)

B	A	C
0	0	0
0	1	1
1	0	Z
1	1	Z

(c)

B	A	C
0	0	1
0	1	0
1	0	Z
1	1	Z

(d)

Fig: Four kinds of three state buffers

In Figures (a) and (b), the enable input B is not inverted, so the buffer output is enabled when B = 1 and disabled when B = 0. That is, the buffer operates normally when B = 1, and the buffer output is effectively an open circuit when B = 0. We use the symbol Z to represent this high-impedance state.

In Figure (b), the buffer output is inverted so that C = A' when the buffer is enabled.

The buffers in Figures (c) and (d) operate the same as in (a) and (b) except that the enable input is inverted, so the buffer is enabled when B = 0.

In the following Figure, the outputs of two three-state buffers are tied together. When B = 0, the top buffer is enabled, so that D = A; when B = 1, the lower buffer is enabled, so that D = C.