SCHOOL OF ELECTRONICS ENGINEERING

Fall Semester 2024-2025

Continuous Assessment Test – I

Programme Name & Branch: B.Tech **Slot: D2+TD2**

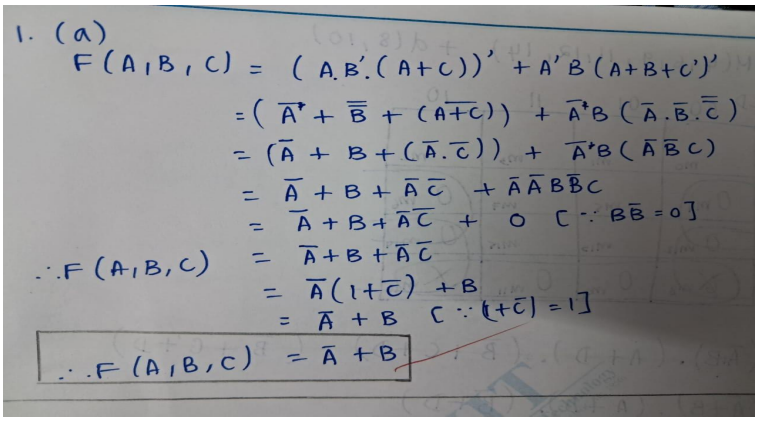
Course Name & code: Digital Systems Design (BECE102L)

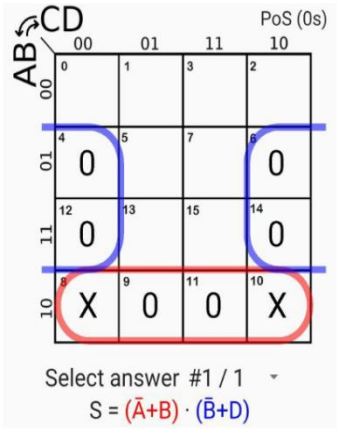
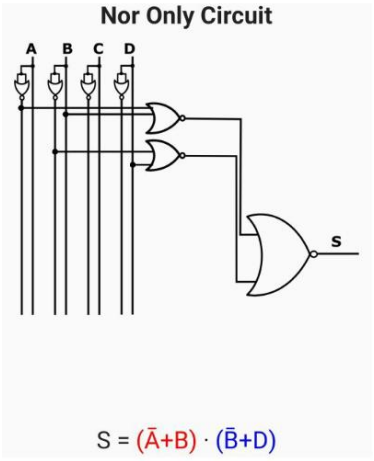
Class Number (s): VL2024250103667, 3679, 3674, 4051, 4054

Faculty Name (s): (Nithish Kumar V, Dhanabal R, Tanmaya Kumar Das, Jayakrishnan P, Shilpi Ruchi Kerketta)

Exam Duration: 90 Min. **Maximum Marks: 50**

General instruction(s): **ANSWER ALL QUESTIONS**

Q.No.	Question	Max Marks	CO	BL
1.	(a) Simplify the given Boolean function using theorems, postulates and De Morgan's theorems. $F(A,B,C) = (A.B'.(A + C))' + A'.B.(A + B + C')'$ (b) Express the simplified function of (a) in canonical form.  A'+B A'(B+B')(C+C')+(A+A')B(C+C') A'(BC+BC'+B'C+B'C')+B(AC+AC'+A'C+A'C') (A'BC+A'BC'+A'B'C+A'B'C')+(ABC+ABC'+A'BC+A'BC') f(A,B,C)=∑m(0,1,2,3,6,7)	10	CO1	BL3
2.	Using K-maps simplify the Boolean function F = ∏M(4,6,9,11,12,14) + d(8, 10) and sketch the minimized expression with NOR gates only.	10	CO1	BL3

3.	(a) List the error in the following code and rewrite the error free code in gate level to act as a 2x1 multiplexer.
	Module m21-str(Y, D0, D1, S); //line 1
	inputs D0, D1, S //line 2
	Output Y, S; //line 3
	wires T1, T2, Sbar //line 4
	and A1(T1; D1; S); //line 5
	and A2(T2 D0 Sbar) //line 6
	not (Sbar S) //line 7
	OR (Y T1 T2) //line 8
	EndModule //line 9
	
	module m21_str(Y,S,D0,D1); //line 1
	input D0, D1, S; //line 2
	output Y; //line 3
	wire T1, T2, Sbar; //line 4
	and A1(T1,D1, S); //line 5
	and A2(T2,D0,Sbar); //line 6
	not (Sbar,S); //line 7
	or (Y,T1,T2); //line 8
	endmodule //line 9
		10	CO2	BL3

	(b) Execute the following statements and display the output values. (i) &X, where X=4'b1010. Ans. 0 (ii) (A && B), where A=4'b0010 and B=4'b0100. Ans. 1 (iii) A?B:C where A = 3'b101, B=3'b110 and C=3'b01 Ans B=3'b110 (iv) {X,2{Y}}, where X=4'b1011 and Y= 4'b1100.			

<table>
<tr><td></td><td>Ans 8'b11001100
(v) (X = = =Y), where X=4’b1010 and Y= 4’b10x0.
Ans 0000</td><td></td><td></td><td></td></tr>
<tr><td>4.</td><td>Design a 3-bit combinational circuit whose output is ‘1’ if the input variables have more 0’s than 1’s. The output is ‘0’ otherwise. Write a Verilog gate-level model of the circuit along with its testbench code.
<table><tr><th>A</th><th>B</th><th>C</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td><td>0</td></tr></table>=∑m(0,1,2,4)

F=A’B’C’+A’B’C+A’BC’+AB’C’

module minterm_logic_gate_level (input A, B, C, output F);
// Intermediate signals
wire A_not, B_not, C_not;
wire term1, term2, term3, term4;
// Generate negations
not (A_not, A);
not (B_not, B);
not (C_not, C);
// Implement each product term
and (term1, A_not, B_not, C_not); // A'B'C'
and (term2, A_not, B_not, C); // A'B'C
and (term3, A_not, B, C_not); // A'BC'
and (term4, A, B_not, C_not); // AB'C’
// Combine terms with OR gate
or (F, term1, term2, term3, term4);
endmodule</td><td>10</td><td>CO3</td><td>BL3</td></tr>
<tr><td>5.</td><td>(a) Design a 3-bit EVEN Parity Generator Circuit using 4x1 multiplexer</td><td>10</td><td>CO3</td><td>BL3</td></tr>
</table>

Q. 5)(a) 3 bit even parity generator

i/ps: A, B, C

o/p: P

i/p			o/p
A	B	C	P
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

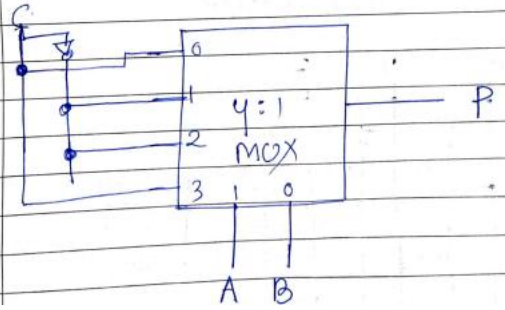
$P = \sum m(1, 2, 4, 7)$

Even parity generator using 4×1 MUX

Let A, B : selection i/ps

C : i/p

Input			o/p	
A	B	C	P	
0	0	0	0	$P = C$
0	0	1	1	
0	1	0	1	$P = \bar{C}$
0	1	1	0	
1	0	0	1	$P = \bar{C}$
1	0	1	0	
1	1	0	0	$P = C$
1	1	1	1	



(b) Implement a full subtractor circuit using 2 to 4 decoders

A	B	B_{in}	D	B_{out}
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

Figure-5:Truth Table of Full subtractor