Department of ISE
Internal Assessment Test 1 – Dec. 2023
Evaluation scheme

Sub:	**Digital Design and Computer Organization**					Sub Code:	**BCS302**	Branch:	**ISE**
Date:	**18.12.2023**	Duration:	90 min's	Max Marks:	50	Sem/Sec	03/ A,B,C		OBE

	Answer any FIVE FULL QUESTIONS	MARKS	CO	RBT
1(a)	Demonstrate the validity of the following identities by means of truth tables (i)x + yz = (x + y)(x + z) **Truth Table (2)** (ii) x(y + z) = xy + xz **Truth Table (2)**	4	CO1	L3
1(b)	Solve the following Boolean expressions to a minimum number of literals. Also draw logic diagrams of the circuits that implement the original and simplified expressions (i)(x + y) (x + y') **Simplification (2)** **Logic Diagram (1)** (ii)xyz + x'y + xyz' **Simplification (2)** **Logic Diagram (1)**	6	CO1	L3
2	Explain the Boolean theorems and postulates in detail **10 properties with truth table (each property carries 1 mark)**	10	CO1	L2
3(a)	Simplify the following Boolean expressions, using four-variable maps: A'B'C'D + AB'D + A'BC' + ABCD + AB'C **K-map (4)** **Simplified expression (1)**	5	CO1	L3
3(b)	Solve the following Boolean function F , together with the don't-care conditions d , and then express the simplified function in SOP and POS form. F(A,B,C,D) = Σ(0, 6,8,13,14) d(A,B,C,D) = Σ(2, 4, 10) **K-map (3)** **Simplified SOP (1)** **Simplified POS (1)**	5	CO1	L3
4(a)	Model the multiple-level NOR circuit for the following expression: CD(B + C)A + (BC' + DE') **Logic Diagram with basic gates (2)** **Logic Diagram with only NOR universal gate(3)**	5	CO1	L3
4(b)	Present the multiple-level NAND circuit for the following expression: w(x + y + z) + xyz **Logic Diagram with basic gates (2)** **Logic Diagram with only NAND universal gate(3)**	5	CO1	L3
5	Design a 3-bit subtractor combinational circuit using 2 bit subtractors **Truth Table of Full subtractor (3)**	10	CO2	L3

Dr.Ciyamala Kushbu S, AP/ISE

	K-Map for Difference and Borrow (2) **Simplified expression for Difference and Borrow (2)** **Logic Diagram using Half subtractor(3)**			
6(a)	Discuss the combinational circuit that decodes 3 input variables? **Truth Table of 3:8 decoder (2)** **K-Map for 3:8 decoder (2)** **Logic Diagram (1)**	5	CO2	L2
6(b)	Discuss the combinational circuit that encodes 4 input variables? **Truth Table of 4:2 encoder (2)** **K-Map for 4:2 encoder (2)** **Logic Diagram (1)**	5	CO2	L2

CCI **CI** **HOD**

Dr.Ciyamala Kushbu S, AP/ISE

Department of ISE
Internal Assessment Test 1 – Dec. 2023
IAT-1 solution

1(a) Demonstrate the validity of the following identities by means of truth tables

(i)x + yz = (x + y)(x + z)

$x\ y\ z$	$x + yz$	$(x + y)$	$(x + z)$	$(x + y)(x + z)$
0 0 0	0	0	0	0
0 0 1	0	0	1	0
0 1 0	0	1	0	0
0 1 1	1	1	1	1
1 0 0	1	1	1	1
1 0 1	1	1	1	1
1 1 0	1	1	1	1
1 1 1	1	1	1	1

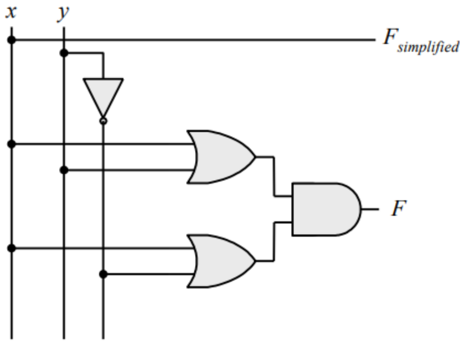
(ii) x(y + z) = xy + xz

$x\ y\ z$	$x(y + z)$	xy	xz	$xy + xz$
0 0 0	0	0	0	0
0 0 1	0	0	0	0
0 1 0	0	0	0	0
0 1 1	0	0	0	0
1 0 0	0	0	0	0
1 0 1	1	0	1	1
1 1 0	1	1	0	1
1 1 1	1	1	1	1

1(b) Solve the following Boolean expressions to a minimum number of literals. Also draw logic diagrams of the circuits that implement the original and simplified expressions

(i)(x + y) (x + y')

= x + yy'
= x(x +y') + y(x + y')
= xx + xy' + xy + yy'
= x



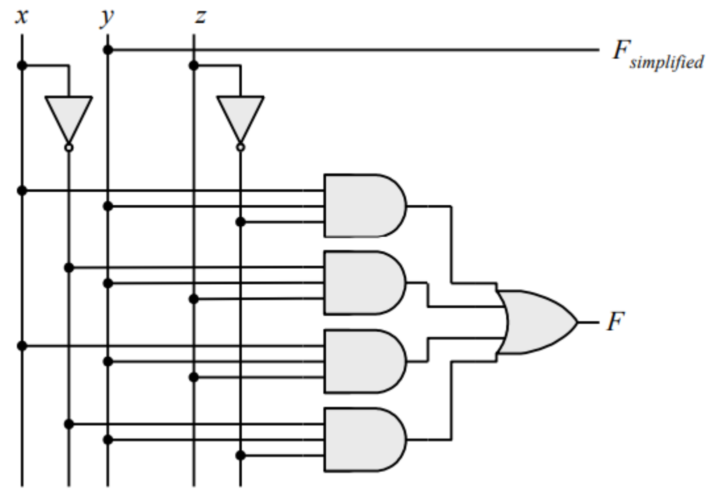
Dr.Ciyamala Kushbu S, AP/ISE

(ii)xyz + x’y + xyz’

= xy(z + z') + x'y

= xy + x'y

= y



2 Explain the Boolean theorems and postulates in detail

<table>
<tr><th>S.No</th><th>Laws Names</th><th>AND property</th><th>OR Property</th></tr>
<tr><td>1</td><td>Identity Law</td><td>A.1 = A<table><tr><th>A</th><th>1</th><th>$A \cdot 1$</th></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table></td><td>A+0 = A<table><tr><th>A</th><th>0</th><th>$A + 0$</th></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
<tr><td>2</td><td>Zero And One Law</td><td>A.0 = 0<table><tr><th>A</th><th>0</th><th>$A \cdot 0$</th></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table></td><td>A+1 = 1<table><tr><th>A</th><th>1</th><th>$A + 1$</th></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>3</td><td>Inverse Law</td><td>A.A' = 0<table><tr><th>A</th><th>A'</th><th>$A \cdot A'$</th></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr></table></td><td>A+A' = 1<table><tr><th>A</th><th>A'</th><th>$A + A'$</th></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr></table></td></tr>
<tr><td>4</td><td>Idempotent Law</td><td>A.A = A<table><tr><th>A</th><th>A</th><th>$A \cdot A$</th></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table></td><td>A + A = A<table><tr><th>A</th><th>A</th><th>$A + A$</th></tr><tr><td>1</td><td>1</td><td>1</td></tr><tr><td>0</td><td>0</td><td>0</td></tr></table></td></tr>
</table>

5 Commutative Law

A.B = B.A

A	B	$A\cdot B$	$B\cdot A$
1	1	1	1
1	0	0	0
0	1	0	0
0	0	0	0

A+B = B+A

A	B	$A+B$	$B+A$
1	1	1	1
1	0	1	1
0	1	1	1
0	0	0	0

6 Associative Law

(A.B).C = A.(B.C)

A	B	C	$(A\cdot B)\cdot C$	$A\cdot(B\cdot C)$
1	1	1	1	1
1	1	0	0	0
1	0	1	0	0
1	0	0	0	0
0	1	1	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	0	0

(A+B)+C = A+(B+C)

A	B	C	$(A+B)+C$	$A+(B+C)$
1	1	1	1	1
1	1	0	1	1
1	0	1	1	1
1	0	0	1	1
0	1	1	1	1
0	1	0	1	1
0	0	1	1	1
0	0	0	0	0

7 Absorption Law

A(A+B) = A

A	B	$A+B$	$A\cdot(A+B)$
1	1	1	1
1	0	1	1
0	1	1	0
0	0	0	0

A + A.B = A

A	B	$A\cdot B$	$A+(A\cdot B)$
1	1	1	1
1	0	0	1
0	1	0	0
0	0	0	0

8 Distribution Law

$A\cdot(B+C)=(A\cdot B)+(A\cdot C)$

A	B	C	$B+C$	$A\cdot(B+C)$	$(A\cdot B)+(A\cdot C)$
1	1	1	1	1	1
1	1	0	1	1	1
1	0	1	1	1	1
1	0	0	0	0	0
0	1	1	1	0	0
0	1	0	1	0	0
0	0	1	1	0	0
0	0	0	0	0	0

$A+(B\cdot C)=(A+B)\cdot(A+C)$

A	B	C	$B\cdot C$	$A+(B\cdot C)$	$(A+B)\cdot(A+C)$
1	1	1	1	1	1
1	1	0	0	1	1
1	0	1	0	1	1
1	0	0	0	1	1
0	1	1	1	1	1
0	1	0	0	0	0
0	0	1	0	0	0
0	0	0	0	0	0

9 De Morgan's Law

(A.B)' = A'+B'

A	B	$A\cdot B$	$(A\cdot B)'$	A'	B'	$A'+B'$
1	1	1	0	0	0	0
1	0	0	1	0	1	1
0	1	0	1	1	0	1
0	0	0	1	1	1	1

(A+B)' = A'.B'

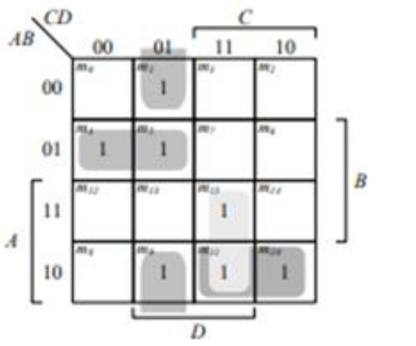
A	B	$A+B$	$(A+B)'$	A'	B'	$A'\cdot B'$
1	1	1	0	0	0	0
1	0	1	0	0	1	0
0	1	1	0	1	0	0
0	0	0	1	1	1	1

10 Double Complement Law

(A’)’ = A

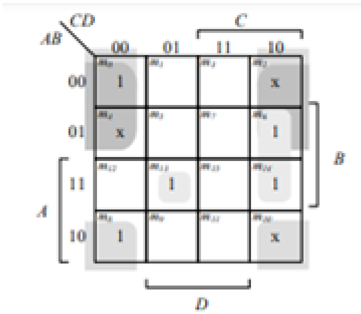
A	A'	$(A')'$
1	0	1
0	1	0

3(a) Simplify the following Boolean expressions, using four-variable maps:
A'B'C'D + AB'D + A'BC' + ABCD + AB'C



(c) $F = A'BC' + B'C'D + ACD + AB'C$

3(b) Solve the following Boolean function F, together with the don't-care conditions d , and then express the simplified function in SOP and POS form.
F(A,B,C,D) = Σ(0, 6,8,13,14)
d(A,B,C,D) = Σ(2, 4, 10)

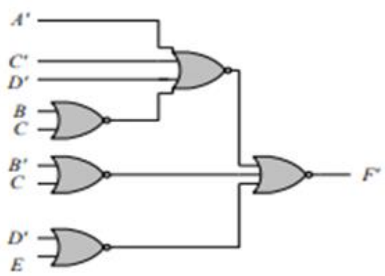


$F = A'D' + B'D' + BCD' + ABC'D$
$F = \Sigma(0, 2, 4, 6, 8, 10, 13, 14)$

4(a) Model the multiple-level NOR circuit for the following expression: CD(B + C)A + (BC' + DE')

Multi-level NOR:

$F = ACD(B + C) + (BC' + DE')$
$F' = [ACD(B + C) + (BC' + DE')]'$
$F' = [(A' + C' + D')'(B + C) + (B' + C)' + (D' + E)']'$
$F' = [((A' + C' + D') + (B + C)')' + (B' + C)' + (D' + E)']'$
$F' = [(A' + C' + D' + (B + C)')' + (B' + C)' + (D' + E)']'$

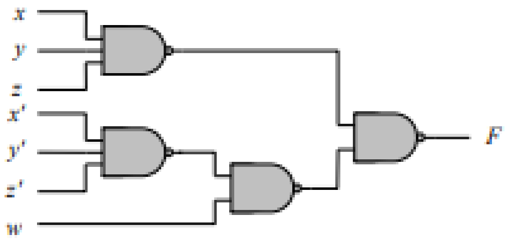


Dr.Ciyamala Kushbu S, AP/ISE

4(b) Present the multiple-level NAND circuit for the following expression: w(x + y + z) + xyz

$F = w(x + y + z) + xyz$

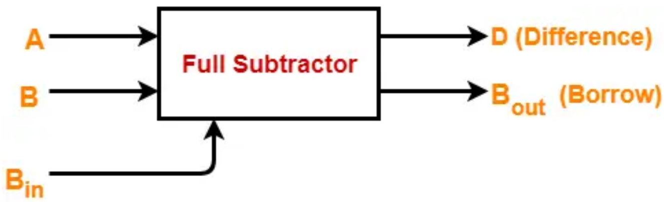
$F' = [w(x + y + z)]'[xyz]' = [w(x'y'z')']'(xyz)'$



5 Design a 3-bit subtractor combinational circuit using 2 bit subtractors

A 3-bit subtractor or full subtractor is a combinational circuit that performs subtraction involving three bits, namely A (minuend), B (subtrahend), and Bin (borrow-in) . It accepts three inputs: A (minuend), B (subtrahend) and a Bin (borrow bit) and it produces two outputs: D (difference) and Bout (borrow out). The logic symbol and truth table are shown below.

Logic Symbol of Full subtractor



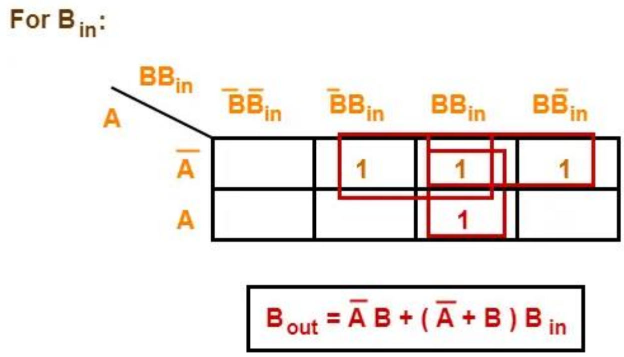
Truth Table of Full subtractor

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

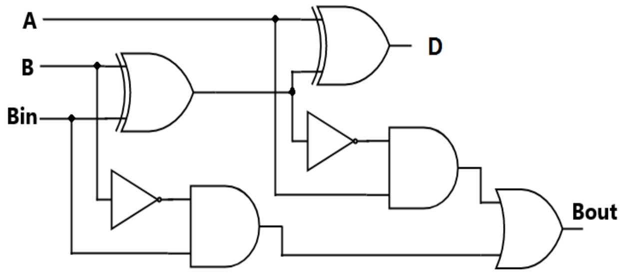
From the above truth table we can find the boolean expression using K-map

For D:

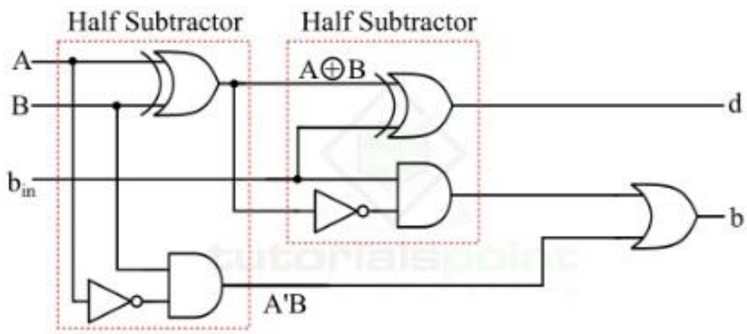
$D = A \oplus B \oplus B_{in}$



From the equation we can draw the Full-subtractor circuit as shown



(Or)



Circuit Diagram for Full subtractor using Half-subtractors

6(a) Discuss the combinational circuit that decodes 3 input variables?

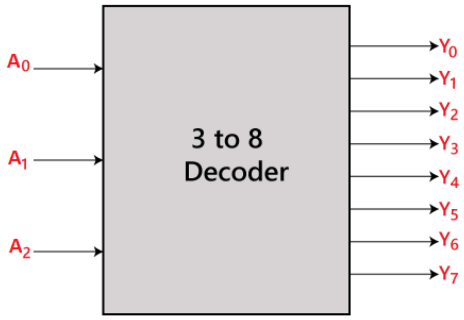
Decoder:

The combinational circuit that change the binary information into 2^N output lines is known as Decoders. The binary information is passed in the form of N input lines. The output lines define the 2^N-bit code for the binary information. At a time, only one input line is activated for simplicity. The produced 2^N-bit output code is equivalent to the binary information.

3:8 Decoder:

The 3 to 8 line decoder is also known as Binary to Octal Decoder. In a 3 to 8 line decoder, there is a total of eight outputs, i.e., Y0, Y1, Y2, Y3, Y4, Y5, Y6, and Y7 and three outputs, i.e., A0, A1, and A2. This circuit has an enable input 'E'. Just like 2 to 4 line decoder, when enable 'E' is set to 1, one of these four outputs will be 1. The block diagram and the truth table of the 3 to 8 line encoder are given below.

Block Diagram:



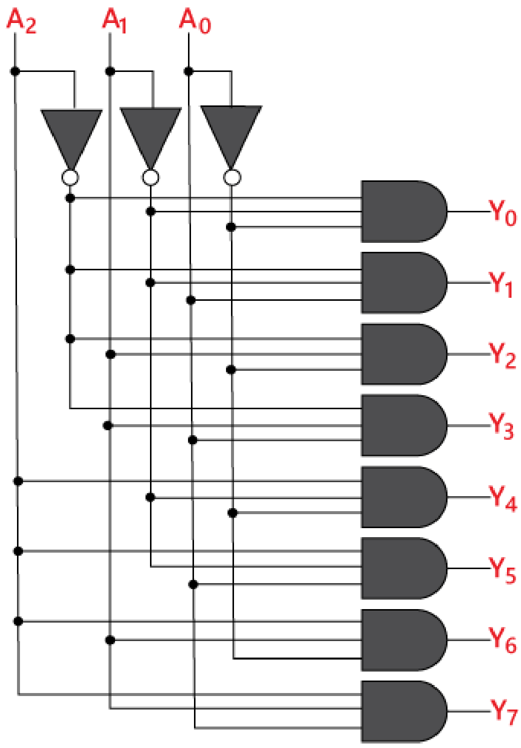
Truth Table:

Enable	INPUTS			Outputs							
E	A_2	A_1	A_0	Y_7	Y_6	Y_5	Y_4	Y_3	Y_2	Y_1	Y_0
0	x	x	x	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1	0	0
1	0	1	1	0	0	0	0	1	0	0	0
1	1	0	0	0	0	0	1	0	0	0	0
1	1	0	1	0	0	1	0	0	0	0	0
1	1	1	0	0	1	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0

The logical expression of the term Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6, and Y_7 is as follows: **(solve using k –map)**

$$Y_0=A_0'.A_1'.A_2'$$
$$Y_1=A_0.A_1'.A_2'$$
$$Y_2=A_0'.A_1.A_2'$$
$$Y_3=A_0.A_1.A_2'$$
$$Y_4=A_0'.A_1'.A_2$$
$$Y_5=A_0.A_1'.A_2$$
$$Y_6=A_0'.A_1.A_2$$
$$Y_7=A_0.A_1.A_2$$

Logical circuit of the above expressions is given below:



6(b) Discuss the combinational circuit that encodes 4 input variables?

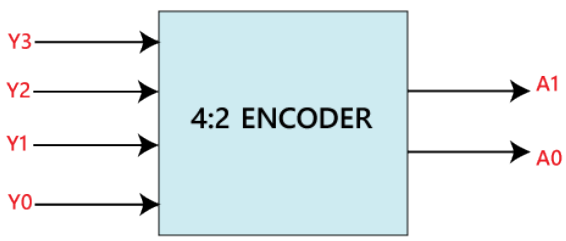
Encoders:

The combinational circuits that change the binary information into N output lines are known as **Encoders**. The binary information is passed in the form of 2^N input lines. The output lines define the N-bit code for the binary information. In simple words, the **Encoder** performs the reverse operation of the **Decoder**. At a time, only one input line is activated for simplicity. The produced N-bit output code is equivalent to the binary information.

4 to 2 line Encoder:

In 4 to 2 line encoder, there are total of four inputs, i.e., Y_0, Y_1, Y_2, and Y_3, and two outputs, i.e., A_0 and A_1. In 4-input lines, one input-line is set to true at a time to get the respective binary code in the output side. Below are the block diagram and the truth table of the 4 to 2 line encoder.

Block Diagram:



Truth Table:

INPUTS				OUTPUTS	
Y_3	Y_2	Y_1	Y_0	A_1	A_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1

The logical expression of the term A0 and A1 is as follows: (**solve using k –map**)

$A_1=Y_3+Y_2$
$A_0=Y_3+Y_1$

Logical circuit of the above expressions is given below: