جامعة بيرزيت

Birzeit University
Faculty of Engineering and Technology
Department of Electrical and Computer Engineering
First Semester – 2023/2024
ENCS2340 - Digital Systems
Homework # 2

Student name: Mohammed Jamil Saada

Student ID: 1221972

Section : 1

Notes:

1- Use this page as a cover for your homework.
2- Late homeworks will not be accepted (the system will not allow it).
3- Due date is Wednesday January 17, 2024at 11:59 pm on ritaj.
4- Organize your solution for each question (Q1, Q2, etc.) and add them to one file. Then, name you file as (Assign2_LastName_FirstName_StudetnsID.pdf).

Q1 **(10 points):** Design a combinational circuit with three inputs, x, y and z, and the three outputs, A, B, and C. when the binary input is 0, 1, 2, or 3, the binary output is one greater than the input. When the binary input is 4, 5, 6, or 7, the binary output is one less than the input.

① specification :-

1) There is 3-inputs(x,y,z) and 3-outputs (A,B,C)

2) if the Binary input is [0,1,2,3], then the Binary output is one greater than the input.

3) if the Binary input is [4,5,6,7], then the Binary output is one less than the input

② Formulation : By truth table

x	y	z	A	B	C
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	0	1	1
1	0	1	1	0	0
1	1	0	1	0	1
1	1	1	1	1	0

→ $A = \sum(3,5,6,7)$

$B = \sum(1,2,4,7)$

$C = \sum(0,2,4,6)$

③ Logic Minimization :

K-map For A

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

$A = XZ + XY + YZ$

K-map For B

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

$$B = x'y'z + x'yz' + xy'z' + xyz$$
$$= x'(y'z + yz') + x(y'z' + yz)$$
$$= x'(y \oplus z) + x(y \oplus z)'$$
$$= x \oplus y \oplus z$$

K-map for C

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

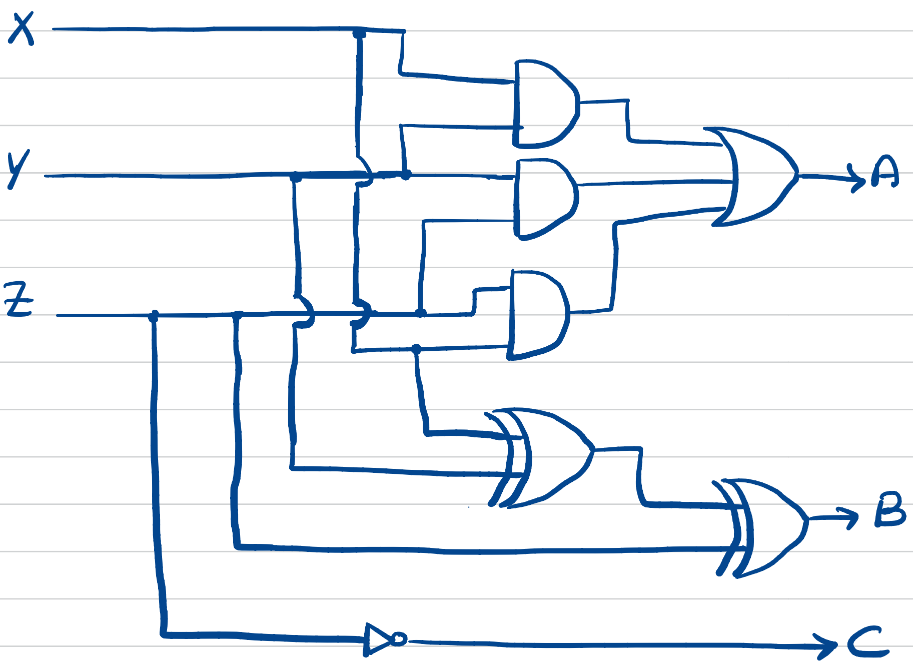
$C = z'$

④ Technology Mapping :-

$A = xy + xz + yz$

$B = x \oplus y \oplus z$

$C = z'$



Q2 **(5 points):** Implement the Boolean function **F(A,B,C) = AB + A'C + A'B'** Using a single 4x1 multiplexer.

$F(A,B,C) = AB + A'C + A'B'$

truth table:

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

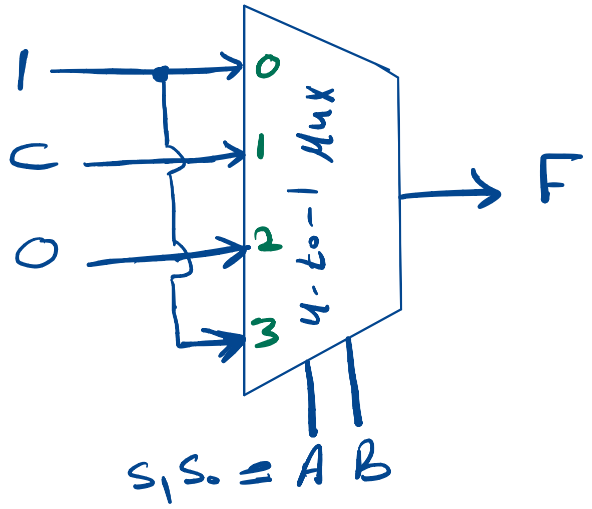
$F=1$

$F=C$

$F=0$

$F=1$

→ For (4 x 1) multiplexer we need 2-select then A is s_1, B is s_0 and C is input



Q3 **(5 points):** Implement the same function in **Q2** using the minimum number of 2x4 decoders with enable and a single NOR gate.

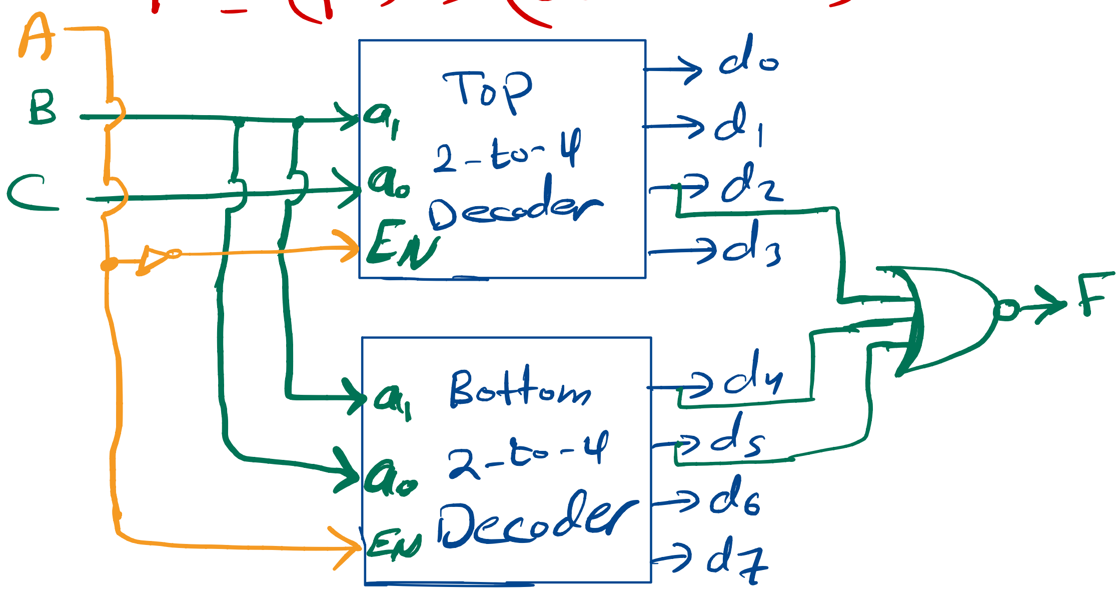
$F(A,B,C) = AB + A'C + A'B' = \sum(0,1,3,6,7)$

truth table:

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

$F' = \sum(2,4,5) = d_2 + d_4 + d_5$

$F = (F')' = (d_2 + d_4 + d_5)'$



Q4 **(10 points):** Implement the following function F(A,B,C,D) = ∑(0, 2, 4, 6, 8, 10) using

a. Mux 4×1
b. Decoders 3-to-8
c. AND-OR
d. NAND-NAND

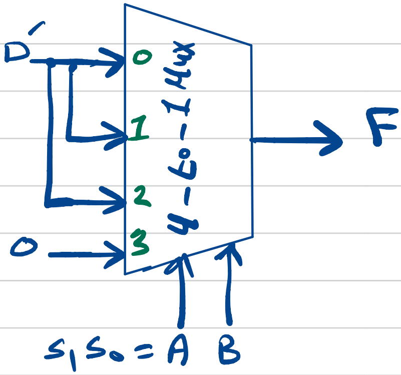
$F(A,B,C,D) = \sum(0,2,4,6,8,10) = d_0+d_2+d_4+d_6+d_8+d_{10}$

truth table :

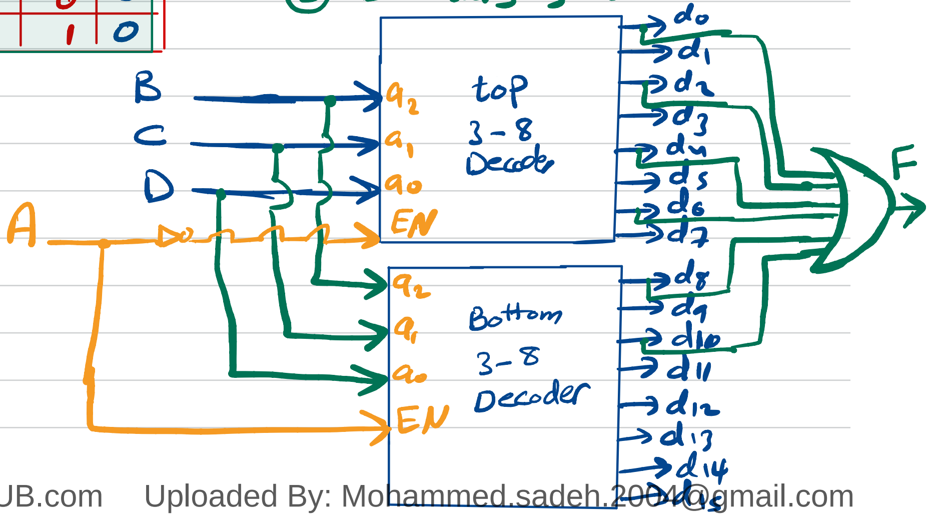
A	B	C	D	F	
0	0	0	0	1	$F=D'$
0	0	0	1	0	
0	0	1	0	1	
0	0	1	1	0	
0	1	0	0	1	$F=D'$
0	1	0	1	0	
0	1	1	0	1	
0	1	1	1	0	
1	0	0	0	1	$F=D'$
1	0	0	1	0	
1	0	1	0	1	
1	0	1	1	0	
1	1	0	0	0	$F=0$
1	1	0	1	0	
1	1	1	0	0	
1	1	1	1	0	

(a) Mux 4x1

For Mux 4x1 we need 2-select then A will Be S_1 and B S_0



(b) Decoders 3-to-8

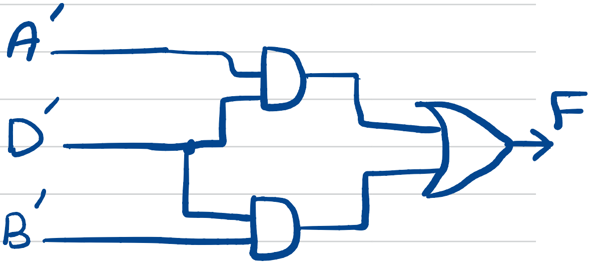


ⓒ AND — OR

$F(A,B,C,D) = \Sigma(0,2,4,6,8,10)$

AB \ CD	00	01	11	10
00	1			1
01	1			1
11				
10	1			1

$F = A'D' + B'D'$

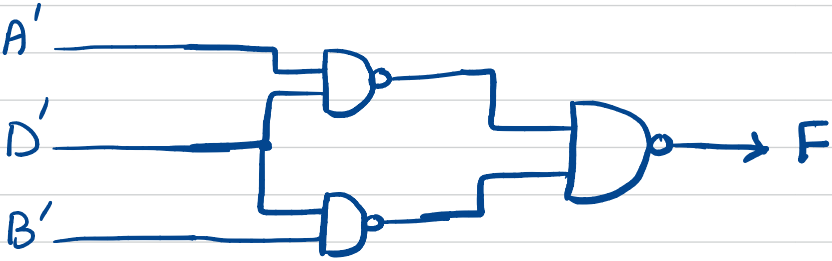


ⓓ NAND - NAND

$F = A'D' + B'D'$

$F' = (A'D' + B'D')' = (A'D')' \cdot (B'D')'$

$F = (F')' = ((A'D')' \cdot (B'D')')'$



Q5 **(6 points):** In the following function determine the Essential prime implicant

F(A,B,C,D) = Σ (0,2,5,7,6,8.9,10,11,13,14,15)

$F(A,B,C,D) = \sum(0,2,5,6,7,8,9,10,11,13,14,15)$

AB \ CD	00	01	11	10
00	1			1
01		1	1	1
11		1	1	1
10	1	1	1	1

→ All prim implicants :-

AC, CD', AB', BD, BC, AD, B'D'

→ Essential prime implicants:

BD, B'D'

Q6 **(4 points):** Explain the concept of odd parity generator?

Odd Parity: one of the two parities we use in generators and checkers to add a parity bit to the n-bit code in generators and then the checkors recieve (n+1) bit.

in odd parity:

→ Count of 1's in the (n+1)-bit code is odd.

→ use an even function to generate the odd parity bit

→ use an even function to check the (n+1)-bit code

Example: for 4-bit code (1101) use odd parity to generate parity bit and check the messge after recieve

we use even function to determine the parity bit → P=0

then the sender transmits (01101)

then the reciever check it by even function to determine if there is an error, if recieved as (01101) → E=0

But if one bit changes from 0 to 1 or 1 to 0 → E=1 and there is an error.