

# EGC220

## Class Notes

### 3/10/2022

**Baback Izadi**

Division of Engineering Programs

[bai@enr.newpaltz.edu](mailto:bai@enr.newpaltz.edu)



# Test 1

## Spring 2022

Average	#####	<b>83</b>
Median	#####	<b>83</b>
MAX	#####	<b>100</b>
Minimum	#####	<b>42</b>

## Spring 2023

Average				67
Median				71
MAX				96
Minimum				0

# Gray code

0  
1  
2  
3

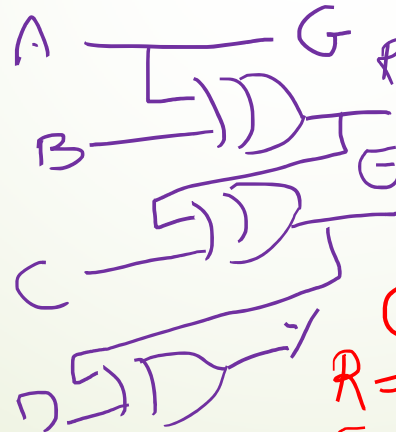
00  
01  
11  
10

2 bit Gray code

$$R = G \oplus B$$

$$E = R \oplus C$$

$$Y = E \oplus D$$



0  
1  
2  
3  
4  
5  
6  
7

000  
001  
011  
010  
110  
111  
101  
100

3 bit Gray code

ABCD	G	R	E	Y
0000	0	0	0	0
0001	0	0	0	1
0011	0	0	1	1
0010	0	0	1	0
0110	0	1	1	0
0111	0	1	0	1
0101	0	1	0	1
0100	0	1	0	0
1100	1	1	0	0
1101	1	1	0	1
1111	1	1	1	1
1110	1	1	1	0
1010	1	0	1	0
1011	1	0	1	1
1001	1	0	0	1
1000	1	0	0	0

$$G = A$$

$$R = \sum m(4, 5, 6, 7, 8, 9, 10, 11)$$

$$E = \sum m(2, 3, 4, 5, 10, 11, 12, 13)$$

$$Y = \sum m(1, 2, 5, 6, 9, 10, 13, 14)$$

Design a circuit that can convert a BCD code into a Gray. a. Write the truth table for this circuit. b. Find the minimized logic equations in SOP and POS for each output c. Draw the corresponding all NAND and all NOR gates logic diagram for this circuit. Label all inputs and outputs.

	W	X	Y	Z
0	0	0	0	0
1	0	0	0	1
2	0	0	1	1
3	0	0	1	0
4	0	1	1	0
5	0	1	1	1
6	0	1	0	1
7	0	1	0	0

$$Z = \sum m(1, 2, 5, 6) + d(10-15)$$

$$W = \sum m(8, 9) + d(10-15)$$

$$X = \sum m(4, 5, 6, 7, 8, 9) + d(10-15)$$

$$Y = \sum m(2, 3, 4, 5) + d(10-15)$$

8	1	1	0	0		
9	1	1	0	1		
10	1	1	1	1		x x x x
11	1	1	1	0		/ / / /
12	1	0	1	0		/ / / /
13	1	0	1	1		/ / / /
14	1	0	0	1		/ / / /
15	1	0	0	0		x x x x

Binary ↓      ↑ Gray

CKT to convert 2 bit binary to Gray

A	B	X	Y
0	0	0	0
0	1	0	1
1	0	1	1
1	1	1	0

$X = A$   
 $Y = A \oplus B$

Gray code to binary

X	Y	A	B
0	0	0	0
0	1	0	1
1	1	1	1
1	0	1	0

$A = X$   
 $B = X \oplus Y$

Binary to Gray

Binary	Gray
000	000
001	001
010	011
011	010
100	110
101	111
110	101
111	100

# GRAY Code

00  
01  
—  
11  
10  
0

000  
001  
011  
010  
—  
110  
111  
101  
100

$$Y = \sum m(1, 2, 5, 6) + d(9)$$

$$G(A, B, C, D) = \sum m(8, 9) + d(10, 11, 12, 13, 14, 15)$$

$$R(A, B, C, D) = \sum m(4, 5, 6, 7, 8, 9) + d( \quad )$$

$$E(A, B, C, D) = \sum m(2, 3, 4, 5) + d( \quad )$$

BCD → GRAY

BCD

A	B	C	D	GRAY	
0	0	0	0	0000	0
0	0	0	1	0001	1
0	0	1	0	0011	2
0	0	1	1	0010	
0	1	0	0	0110	
0	1	0	1	0111	
0	1	1	0	0101	
0	1	1	1	0100	
1	0	0	0	1100	
1	0	0	1	1101	
1	0	1	0	1111	
1	0	1	1	1110	
1	1	0	0	1010	
1	1	0	1	1011	
1	1	1	0	1001	
1	1	1	1	1000	15

1111  
1110  
1010  
1011  
1001  
1000

don't  
 Care  
 ← 15

Bcd → GRAY

GRAY → BCD

ABCD  
 0000  
 0001  
 0010  
 0011  
 0100  
 0101  
 0110  
 0111  
 1000  
 1001  
 1010  
 1011  
 1100  
 1101  
 1110  
 1111

GREY  
 0000  
 0001  
 0011  
 0010  
 0110  
 0111  
 0101  
 0100  
 1100  
 1101  
 1111  
 1110  
 1010  
 1011  
 1001  
 1000

GREY

0000  
 0001  
 0010  
 0011  
 0100  
 0101  
 0110  
 0111  
 1000  
 1001  
 1010  
 1011  
 1100  
 1101  
 1110  
 1111

ABCD

0000  
 0001  
 0011  
 0010  
 0110  
 0111  
 0101  
 0100  
 1100  
 1101  
 1110  
 1111  
 XXXX  
 XXXX

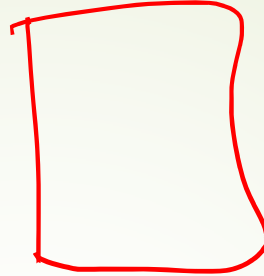
$A = \sum m(13, \dots)$

$B = \sum m(\dots)$

ABCD | wxyz

0000 0000  
0001 0001  
0010 0011  
0011 0010  
0100 0110  
0101 0111  
0110 0101  
0111 0100  
1000 1100  
1001 1101  
1010 1111  
1011 1110  
1100 1010  
1101 1011  
1110 1001  
1111 1000

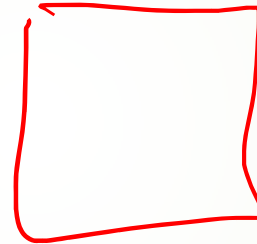
w



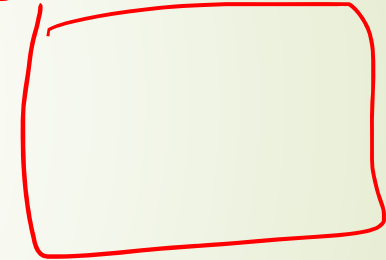
x



y



z





## Problem 1

There is an integrated circuit called a BCD-seven segment decoder that takes 4 inputs and has seven outputs. The inputs represent a number between 0 and 9, and each of the seven outputs corresponds to one of seven LED's in a seven-segment display. A typical seven segment display is shown below.

- Write the truth table for each segment "a, b, c, d, e, f, g" with inputs A, B, C, and D. Make sure to adhere to the indicated segment notations.
- Simplify each output in Minimum S.O.P.
- Implement each output using all NAND gates.

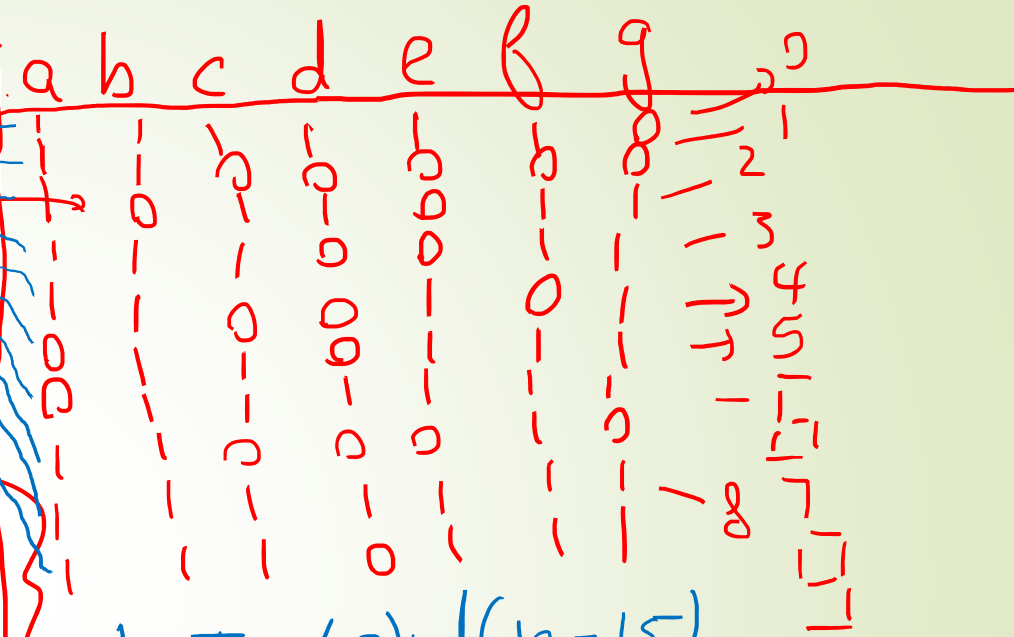
Digit	Inputs				Output (7 Segments)
	D	C	B	A	
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	2
3	0	0	1	1	3
4	0	1	0	0	4
5	0	1	0	1	5
6	0	1	1	0	6
7	0	1	1	1	7
8	1	0	0	0	8
9	1	0	0	1	9



~~V. I. T. X~~

Input

D	C	B	A
0	0	0	0
0	0	0	1
0	0	1	0
0	0	1	1
0	1	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	0	0	0
1	0	0	1



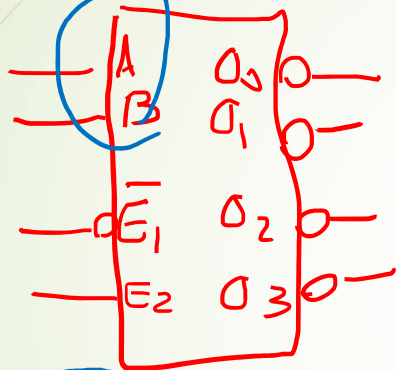
a  
b  
c  
d  
e  
f  
g  
h  
i  
j  
k  
l  
m  
n  
o

$b = TTM(2) + d(10-15)$   
 $c = TTM(1,4,7) + d(10-15)$   
 $g = TTM(0,1,7) + d(10-15)$

$a = TTM(5,6) + d(10,11,12,13,14,15)$

### Problem 2

Design a 1 out of 4 decoder with active low outputs and two enable lines, one active low and one active high.



$E_1$	$E_2$	A	B	$O_3$	$O_2$	$O_1$	$O_0$
1	X	X	X	1	1	1	1
X	0	X	X	1	1	1	1
0	1	0	0	1	1	1	0
0	1	0	1	1	1	0	1
0	1	1	0	1	0	1	1
0	1	1	1	0	1	1	1

inactive

active

$$\begin{aligned} \bar{O}_0 &= \bar{E}_1 \bar{E}_2 \bar{A} \bar{B} \quad m_0 \\ \bar{O}_1 &= \bar{E}_1 \bar{E}_2 \bar{A} B \quad m_1 \\ \bar{O}_2 &= \bar{E}_1 \bar{E}_2 A \bar{B} \quad m_2 \\ \bar{O}_3 &= \bar{E}_1 \bar{E}_2 A B \quad m_3 \end{aligned}$$



Problem 2

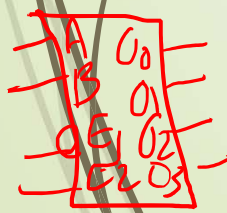
Design a 1 out of 4 decoder with active low outputs and two enable lines, one active low and one active high.

High

High  
Low

$C_1$	$E_2$	A	B
0	0	0	0
0	1	0	1
0	1	1	0
0	1	1	1
1	X	X	X
X	0	X	X

active



$O_3$	$O_2$	$O_1$	$O_0$
0	0	0	1
0	0	1	0
0	1	0	0
1	0	0	0
0	0	0	0
0	0	0	0

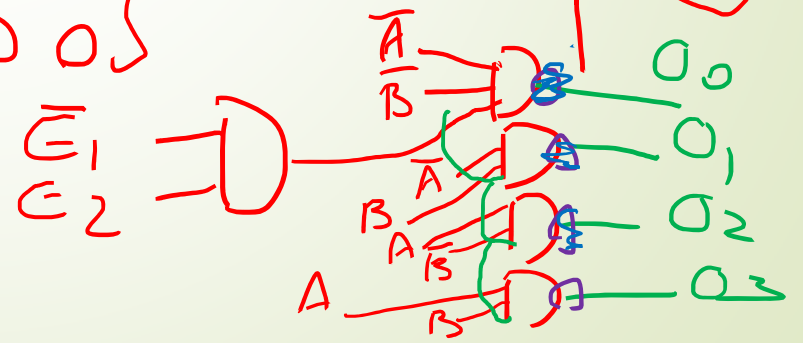
Inactive

$$O_0 = \bar{E}_1 E_2 \bar{A} B$$

$$O_1 = \bar{E}_1 E_2 A \bar{B}$$

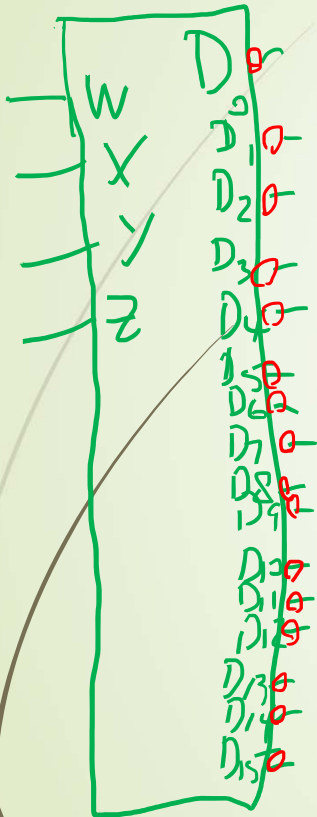
$$O_2 = \bar{E}_1 E_2 A B$$

$$O_3 = \bar{E}_1 E_2 \bar{A} \bar{B}$$



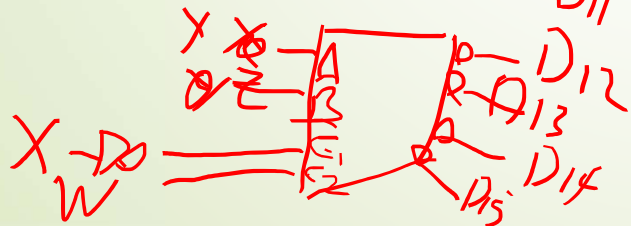
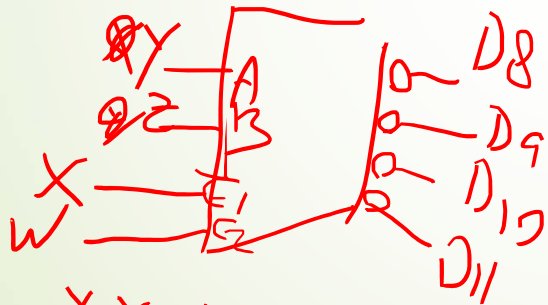
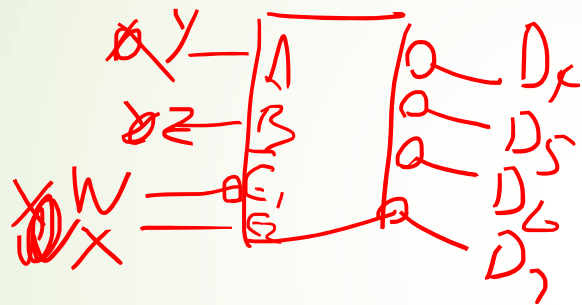
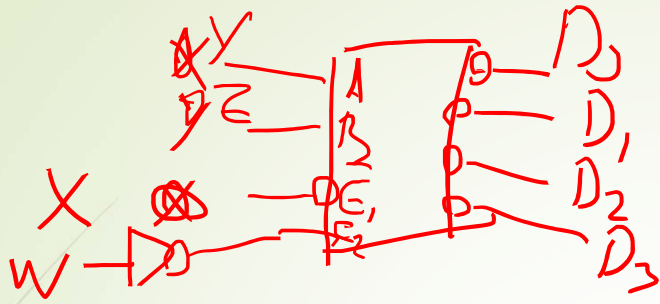
### Problem 3

Using the decoder in Problem 3, design a 1 out of 16 decoder with active low outputs.



W	X	Y	Z	$D_{15}$	$D_{14}$	$D_{13}$	$D_{12}$	$D_{11}$	$D_{10}$	$D_9$	$D_8$	$D_7$	$D_6$	$D_5$	$D_4$	$D_3$	$D_2$	$D_1$	$D_0$
0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Active  
inactive



## Problem 4

Write a Verilog code for the decoder in Problem 2: 1-out of 4 decoder active low outputs and two enable lines, one active low and one active high.

```
module decoder-1-out-4 (EN, E, A, B, O0, O1, O2, O3);  
input EN, E, A, B;  
output O0, O1, O2, O3;  
Assign ~O0 = ~EN & E & ~A & ~B;  
Assign ~O1 = ~EN & E & ~A & B;  
Assign ~O2 = ~EN & E & A & ~B;  
Assign ~O3 = ~EN & E & A & B;  
endmodule
```

1-out-of-4

```
// 2-to-4-Line Decoder with Enable: Dataflow Verilog Desc. // 1  
// (See Example 3-16 for logic diagram) // 2  
module decoder_2_to_4_df_v(EN, A0, A1, D0, D1, D2, D3); // 3  
input EN, A0, A1; // 4  
output D0, D1, D2, D3; // 5  
// 6  
assign D0 = EN & ~A1 & ~A0; // 7  
assign D1 = EN & ~A1 & A0; // 8  
assign D2 = EN & A1 & ~A0; // 9  
assign D3 = EN & A1 & A0; // 10  
// 11  
endmodule // 12
```

# Structural Verilog Description of 2-to-4-Line Decoder

```
// 2-to-4-Line Decoder with Enable: Structural Verilog Desc. // 1
// (See Figure 3-16 for logic diagram) // 2
module decoder_2_to_4_st_v (EN, A0, A1, D0, D1, D2, D3); // 3
  input EN, A0, A1; // 4
  output D0, D1, D2, D3; // 5

  wire A0_n, A1_n, N0, N1, N2, N3; // 6

  not // 7
    g0(A0_n, A0), // 8
    g1(A1_n, A1); // 9
  and // 10
    g3(N0, A0_n, A1_n), // 11
    g4(N1, A0, A1_n), // 12
    g5(N2, A0_n, A1), // 13
    g6(N3, A0, A1), // 14
    g7(D0, N0, EN), // 15
    g8(D1, N1, EN), // 16
    g9(D2, N2, EN), // 17
    g10(D3, N3, EN); // 18
endmodule // 19 // 20
```

