CODE CONVERTERS

Code converters

- A code converter circuit will convert coded information in one form to a different coding form.
- Coded representation for 10 decimal symbols is known as binary coded decimal (or BCD) or decimal codes.
- Minimum 4-bits are required to represent decimal symbol.
- Out of 16 , 4-bit combinations, only 10 combinations are used to represent 10 decimal symbols and remaining 6 will not be used (don't cares)

Binary and Binary coded Decimal(BCD)

Decimal	Binary	BCD
0	0000	0000
..	"	"
9	1001	1001
10	1010	0001 0000
11	1011	0001 0001
12	1100	0001 0010
13	1101	0001 0011
67		0110 0111
90		1001 0000
23		0010 0011

10

0001 0000

8421 BCD code

8 4 2 1

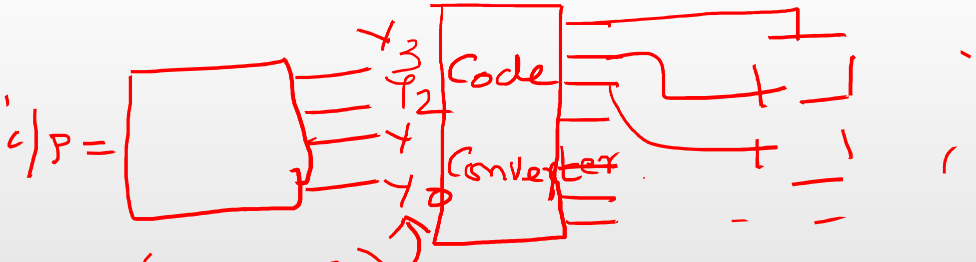
$1001 = (9)_{10}$

Difference between binary and BCD representation

- $(28)_{10}$

  Binary representation : $(11100)_2$

  8421 BCD representation : $(0010\ 1000)_2$



$(0-F)_{16}$

$(0000-1111)_2$

Introduction to BCD codes (4-bit)

Weighted codes = 8421, 84-2-1, 2421

Self-Complementary codes
→ Excess-3
84-2-1
2421

Decimal digit	8421 (BCD)	Excess 3	84-2-1	2421		Gray code
0	0000	0011	0000	0000		0000
1	0001	0100	0111	0001		0001
2	0010	0101	0110	0010	1000	0011
3	0011	0110	0101	0011	1001	0010
4	0100	0111	0100	0100	1010	0110
5	0101	1000	1011	1011	0101	~~0100~~ = 0111
6	0110	1001	1010	1100	0110	0101
7	0111	1010	1001	1101	0111	0100
8	1000	1011	1000	1110		1100
9	1001	1100	1111	1111		1101
Don't cares	1010, 1011, 1100, 1101, 1110, 1111	0000, 0001, 0010, 1101, 1110, 1111	0001, 0010, 0011, 1100, 1101, 1110	1000, 1001, 1010, 0101, 0110, 0111	0010, 0011, 0100, 1011, 1100, 1101	1110, 1111, 1010, 1000, 1011, 1001

Complements

Are used for simplifying the subtraction operation and for logical manipulation.

There are two complements for each base:

- (R-1)'s complement (Diminished radix complement)
- R's complement (Radix complement)

R = 2 8 10 16

1's 2's 7's 8's 9's 10's 15's 16's

- (R-1)'s complement:

  (R-1)'s complement of a number is $(R^n -1) – N$ single-digit no: $n = 1$

  $R = 2$ $N = 0$ (R-1)'s = 1's compl of N = 0

  Where R → base

  $(R^1 - 1) - N$

  N → number who's complement is to be taken $1 - 0 = 1$

  n → number of digits/bits in the number N $1 - 1 = 0$

- R’s complement

  R’s complement of a number is $\underline{\underline{R^n - N}}$

  Where R → base

  N → number who's complement is to be taken

  n → number of digits/bits in the number N

Examples: $R=2, \; N=0,$ 2's compl of a no

R's

$R^n - N \Rightarrow$ single bit $= n=1$

$2^1 - N \Rightarrow N=0,$

$\underline{(R^n-1) - N} + 1 = \underline{\underline{R^n - N}}$

Ex: $R=10, \; \underline{\underline{n=1}}$

$(R-1)'s = 9's$

$(R^1-1) - N = (9-N) = 9's$ compl. of N

$N=8 \Rightarrow 9-8=1$

9's compl of 8

9's comp. of $1 = 8$

10's comp of $\underline{1} = 10-1$

$= 9$

Code converter design steps:

- 1. Write the truth table
- 2. Identify the don’t care inputs from input code
- 3. Write the minterms/maxterms for every output variable
- 4. Simplify the expressions for output variables
- 5. Draw the circuit using the specified gates.

1.Design a 3 bit binary to gray code converter.

a

3-bit Binary B2 B1 B0	Gray G2 G1 G0
0 0 0	0 0 0
0 0 1	0 0 1
0 1 0	0 1 1
0 1 1	0 1 0
1 0 0	1 1 0
1 0 1	1 1 1
1 1 0	1 0 1
1 1 1	1 0 0

b

$G_2 = \Sigma m(4, 5, 6, 7)$

$G_1 = \Sigma m(2, 3, 4, 5)$

$G_0 = \Sigma m(1, 2, 5, 6)$

c

B_2 \ B_1B_0				
0	0	0	0	0
1	1	1	1	1

$G_2 = B_2$

B_2 \ B_1B_0				
0	0	0	1	1
1	1	1	0	0

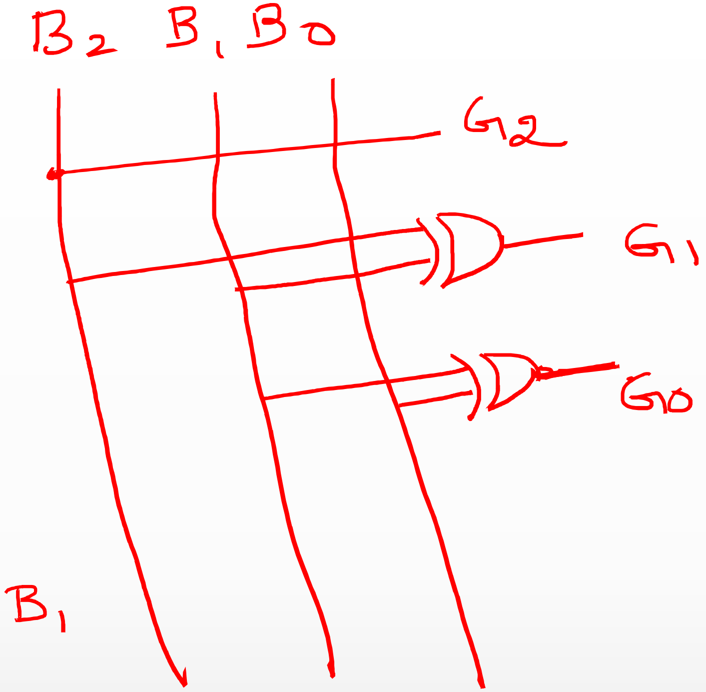
$G_1 = \overline{B_2}B_1 + B_2\overline{B_1} = B_2 \oplus B_1$

G_2 \ G_1G_0		01		10
		1		1
		1		1

$G_0 = \overline{B_1}B_0 + B_1\overline{B_0}$

$= B_1 \oplus B_0$

d



$B_2B_1B_0$ $G_2G_1G_0$

1 1 1 ⇒ 1 0 0

$(10110110)_2$

$\oplus$

$= (11101101)$ Gray code

Gray binary.

1. Design a 3 bit ~~binary~~ to ~~gray~~ code converter ~~contd~~.. ← Assignment

2. Design a code converter to convert a decimal digit represented in 8421 code to a decimal digit represented in Excess 3 code. ⇒ NAND gates

(a)

Decimal digit	8 4 2 1 A B C D	Excess 3 code E3 E2 E1 E0
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100
Don't cares	1010,1011, 1100,1101,1110, 1111	--------------

8421 → Excess-3

↳ simplified SOP → grouping '1's

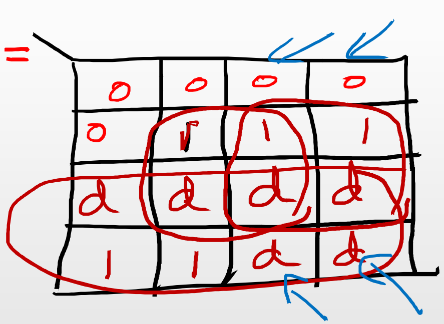
$E_3 = \Sigma m(5,6,7,8,9) + d(10,11,12,13,14,15)$

$E_2 = \Sigma m(1,2,3,4,9) + d(10,11,12,13,14,15)$

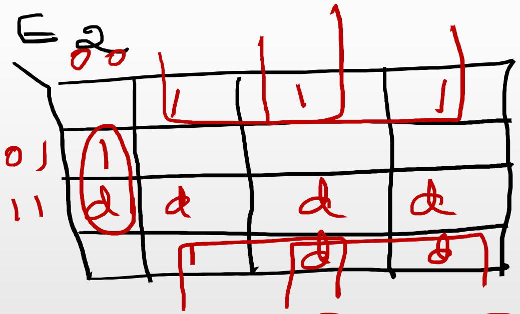
$E_1 = \Sigma m(0,3,4,7,8) + d(\quad " \quad)$

From truth table: $E_0 = \overline{D}$

$E_3 =$

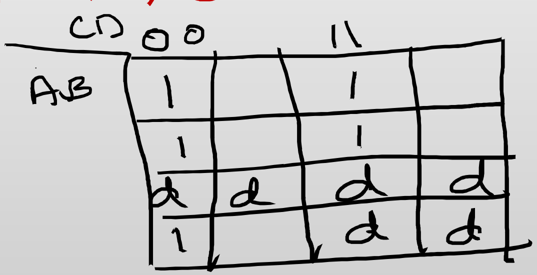


E_2



$E_3 = A + BC + BD$

$E_2 = \overline{B}C + \overline{B}D + B\overline{C}\overline{D}$

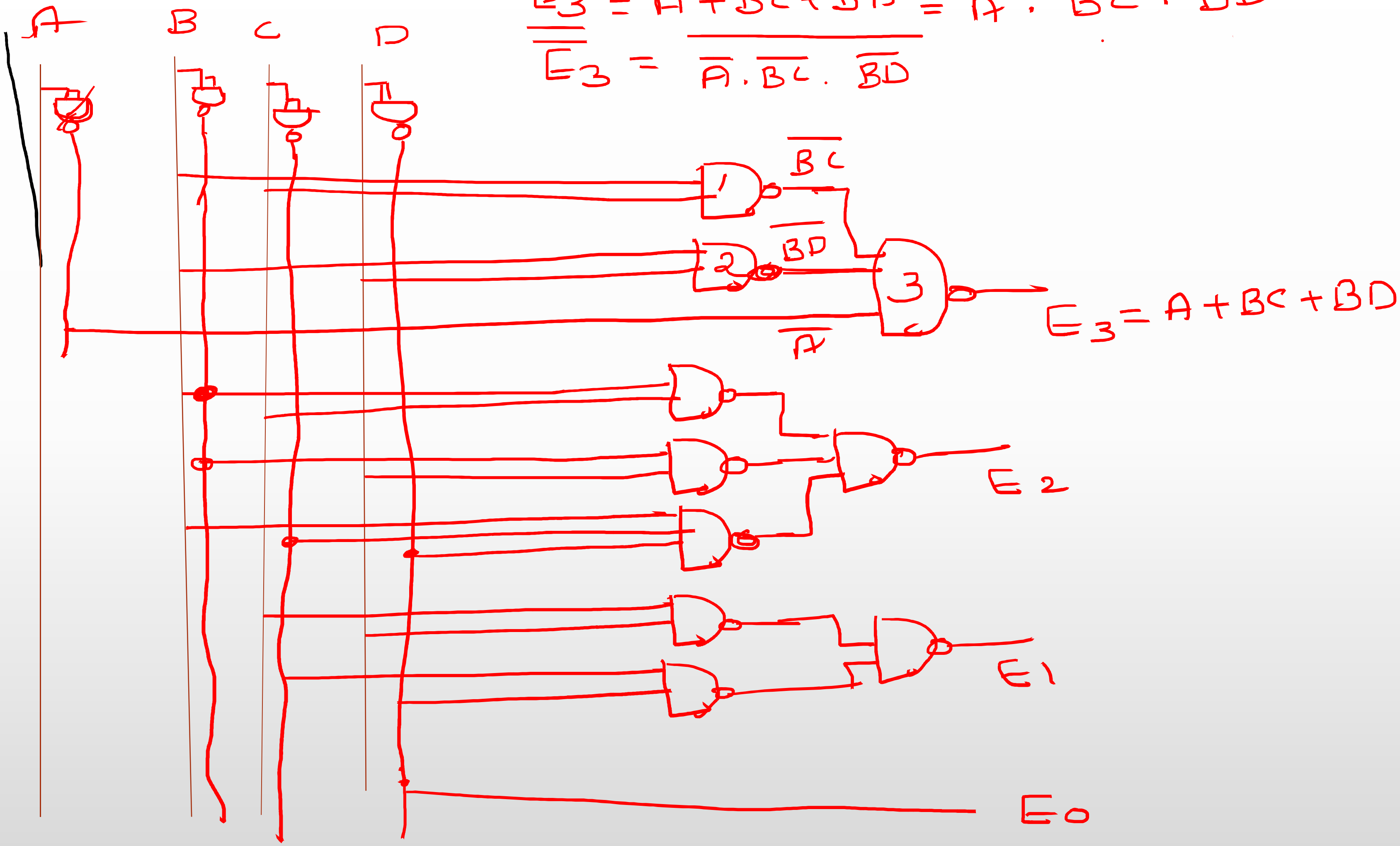


$E_1 = CD + \overline{C}\overline{D}$

2. Design a code converter to convert a decimal digit represented in 8421 code to a decimal digit represented in Excess 3 code contd.

$$\overline{E_3} = \overline{A + BC + BD} = \overline{A} \cdot \overline{BC} \cdot \overline{BD}$$

$$\overline{\overline{E_3}} = \overline{\overline{A} \cdot \overline{BC} \cdot \overline{BD}}$$



2. Design a code converter to convert a decimal digit represented in 8421 code to a decimal digit represented in Excess 3 code.

3. Design a code converter to convert a decimal digit represented in 8 4 2 1 code to a decimal digit represented in 8 4 -2 -1 code. → NOR gates → POS → grouping 0's

Σm

Decimal digit	8 4 2 1 A B C D	8 4 -2 -1 Y3 Y2 Y1 Y0
0	0000	0000
1	0001	0111
2	0010	0110
3	0011	0101
4	0100	0100
5	0101	1011
6	0110	1010
7	0111	1001
8	1000	1000
9	1001	1111
Don't cares	10, 11, 12, 13, 14, 15	

$Y_3 = \Pi M(0, 1, 2, 3, 4) \cdot d(10, 11, 12, 13, 14, 15)$

$Y_2 = \Pi M(0, 5, 6, 7, 8) \cdot d(10, 11, 12, 13, 14, 15)$

$Y_1 = \Pi M(0, 3, 4, 7, 8) \cdot d(\quad " \quad)$

$Y_0 = D$

③ only Simplified POS

④ Circuit using NOR gates

3. Design a code converter to convert a decimal digit represented in 8 4 2 1 code to a decimal digit represented in 8 4 -2 -1 code.

4

Design a code converter to convert a decimal digit represented in Excess 3 code to a decimal digit represented in 8 4 -2 -1 code. = NOR gates

Decimal digit	EXCESS -3 CODE A B C D	8 4 -2 -1 Y3 Y2 Y1 Y0
0	0011	0000
1	0100	0111
2	0101	0110
3	0110	0101
4	0111	0100
5	1000	1011
6	1001	1010
7	1010	1001
8	1011	1000
9	1100	1111
Don't cares	0000,0001,0010, 1101,1110,1111	------------

~~$Y_3 = \Sigma m($~~

$Y_3 = A$

$Y_2 = B$

$Y_1 = \overline{C}$

$Y_0 = \overline{D}$

5

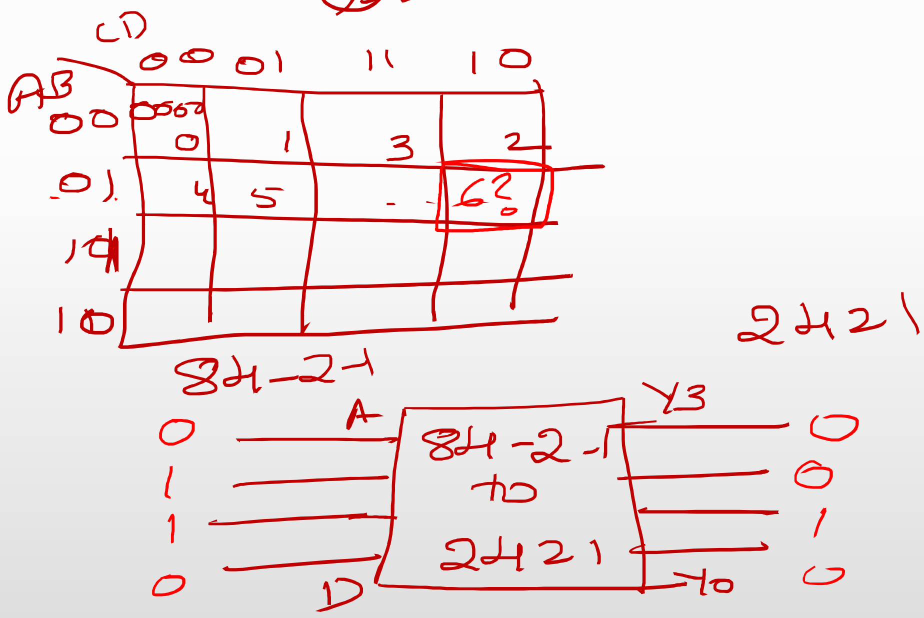
~~4~~. Design a code converter to convert a decimal digit represented in 8 4 -2 -1 code to a decimal digit represented in 2 4 2 1 code. = NOR gates

Decimal digit	8 4 -2 -1 A B C D	2 4 2 1 Y3 Y2 Y1 Y0
0	0000 (0)	0000
1	0111 (7)	0001
2	0110 (6)	0010
3	0101 (5)	0011
4	0100 (4)	0100
5	1011 (11)	1011
6	1010 (10)	1100
7	1001 (9)	1101
8	1000 (8)	1110
9	1111 (15)	1111
Don't cares	0001,0010,0011, 1100,1101,1110	______

$Y_3 = A$

$Y_2 = \Sigma m(4, 6, 7, 8, 9) + d(1, 2, 3, 12, 13, 14)$

$Y_1 = \Sigma m(2, 3, 5, 8, 9) + d(2, 3, 12, 13, 14)$



4. Design a code converter to convert a decimal digit represented in 8 4 -2 -1 code to a decimal digit represented in 2 4 2 1 code contd $\longrightarrow$ NOR

$Y_3 = \Sigma m(8,9,10,11,15) + d(1,2,3,12,13,14) = A$

$Y_2 = \Sigma m(4,8,9,10,15) + d(1,2,3,12,13,14) =$

$Y_1 = \Sigma m(5,6,8,11,15) + d(11)$

$Y_0 = D$

AB \ CD	00	01	11	10
00	0	d	d	d
01	1	0	0	0
11	d	d	1	d
10	1	1	0	1

3 Quads, 1 pair

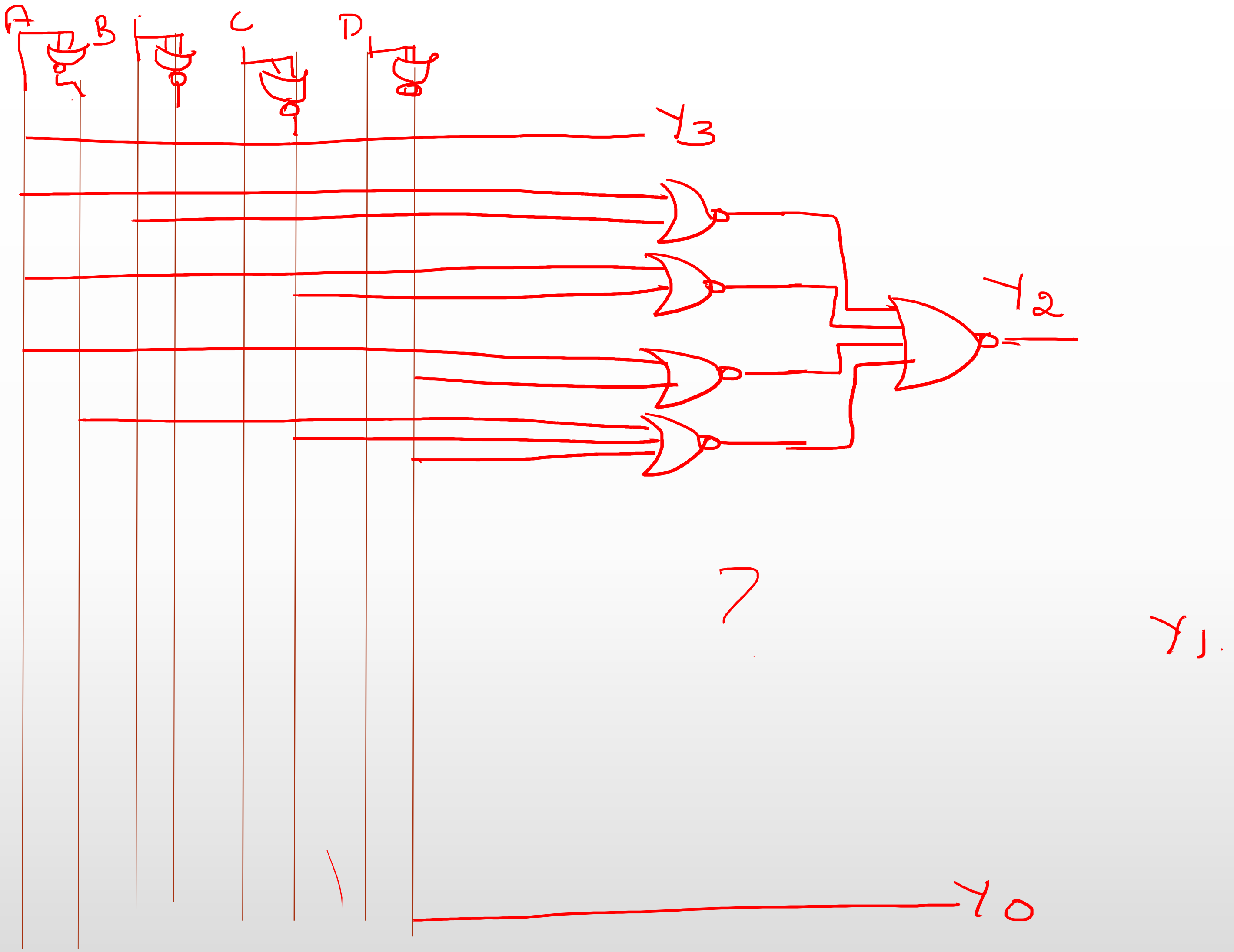
$Y_2 = (A+B)(A+\bar{C})(A+\bar{D})(B+\bar{C}+\bar{D})$

Y_1

				10
	0	d	d	d
01	0	1	0	1
11	d	d	1	d
10	1	0	1	0

$Y_1 = (A+B)(\bar{B}+C+D)(\bar{A}+C+\bar{D})(A+\bar{C}+\bar{D})(\bar{A}+\bar{B}+D)$

$\bar{A}+\bar{B}+D$


A
B
C
D
Y3
Y2
Y1
Y0

5. Design a code converter to convert a decimal digit represented in 2 4 2 1 code to a decimal digit represented in gray code.

Decimal digit	2 4 2 1 A B C D	Gray code G3 G2 G1 G0
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	1011	0111
6	1100	0101
7	1101	0100
8	1 1 10	1100
9	1 1 1 1	1101
	0101, 0110, 0111, 1000, 1001, 1010	

$G_3 = \Sigma m(14, 15) + d(5, 6, 7, 8, 9, 10)$

$G_2 = \Sigma m(4, 11, 12, 13, 14, 15) + d(5, 6, 7, 8, 9, 10)$

$G_1 = \Sigma m(2, 3, 4, 11) + d(\text{"})$

$G_0 = \Sigma m(1, 2, 11, 12, 15) + d(\text{"})$

5. Design a code converter to convert a decimal digit represented in 2 4 2 1 code to a decimal digit represented in gray code

- Any questions?

1 byte

① A = 257; cout<<A<<endl;   A = 1

1 $\boxed{0000\ 0001}$ => 1

256

0011
1100

1001

② A = -3 => ?

4 bits => (-3) = 1001

↳ 2 bytes = unsigned = 0000 - - - - 16 bits

16-bits

253

1111 - - - 1111

~~[illegible]~~ 1111 1111 1111 1101 unsigned

? 65,533