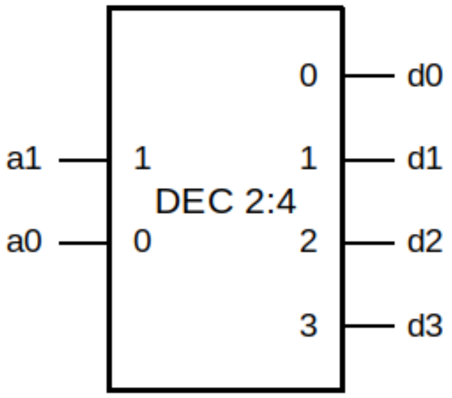
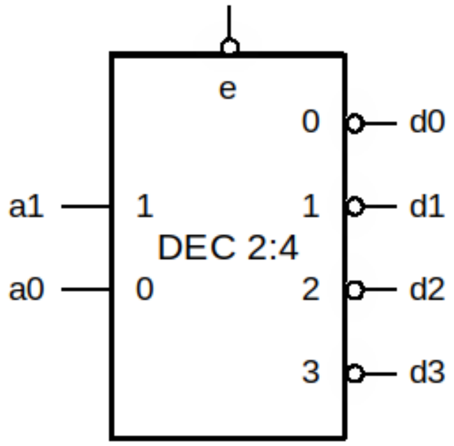
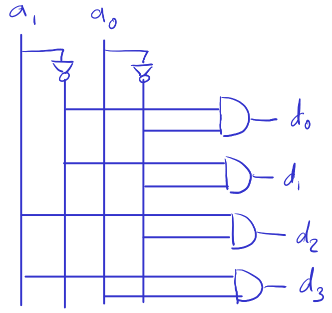
Example 1: design these decoders using logic gates

- DEC 2:4
- DEC 2:4, active low with active low enable





$d_0 = m_0 = \overline{a_1}\,\overline{a_0}$

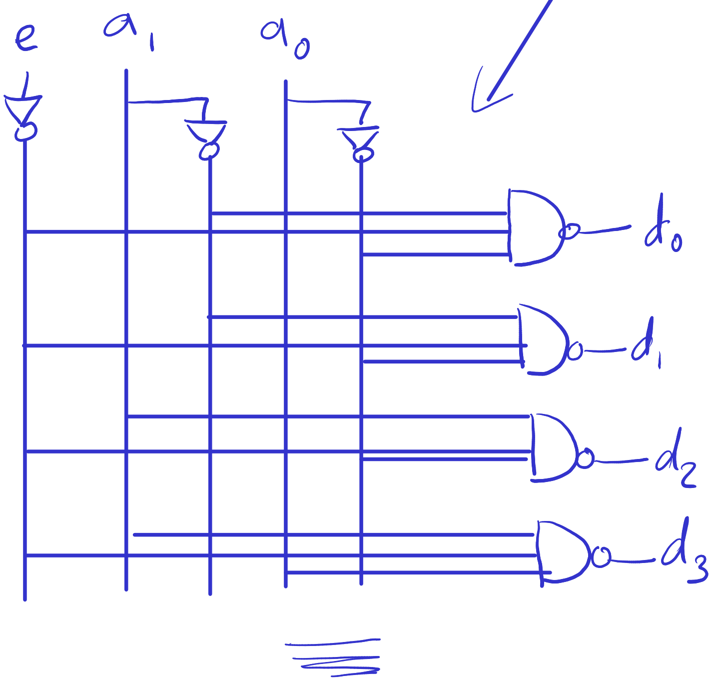


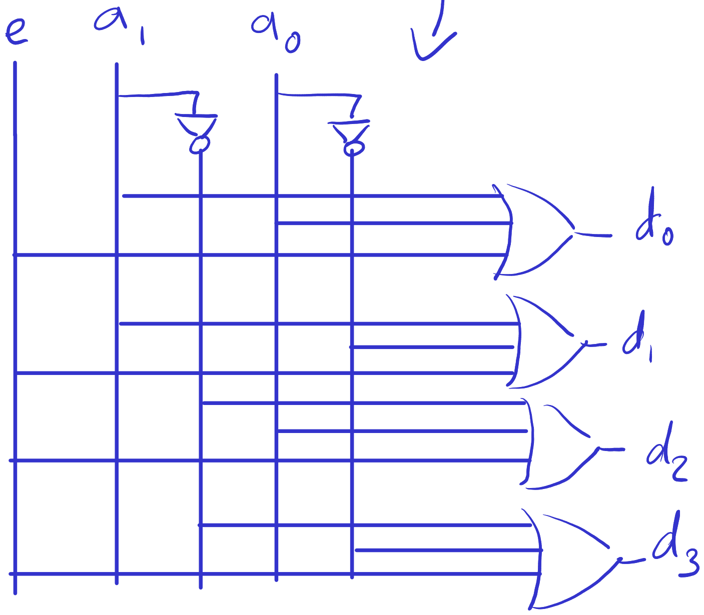
~~$d_0 = \overline{m_0} = \overline{\bar{a}_1 \bar{a}_0}$~~

$d_0 = \overline{\bar{e}\,\bar{a}_1 \bar{a}_0} = e + a_1 + a_0$

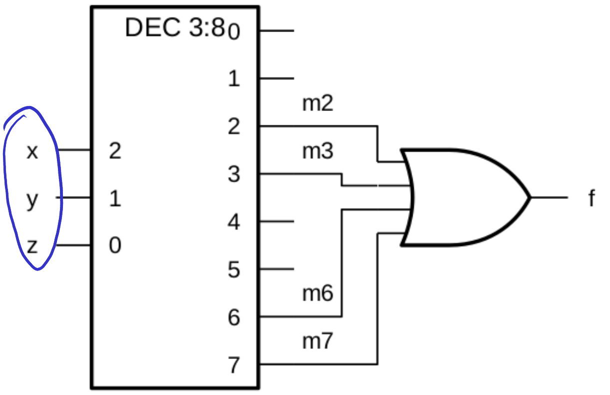
$d_0 =$ ~~$M_0 = a_1$~~ $+ a_0$

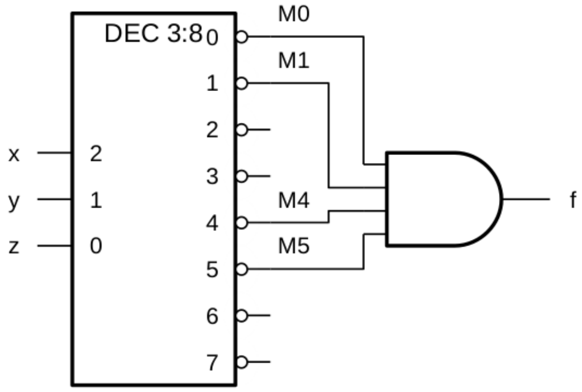
$d_0 = e + a_1 + a_0$





Example 2: f(x, y, z) = ∑(2, 3, 6, 7) = ∏(0, 1, 4, 5)
Design f using a DEC 3:8 (active-high output) and an OR gate
Design f using a DEC 3:8 (active-low output) and an AND gate.

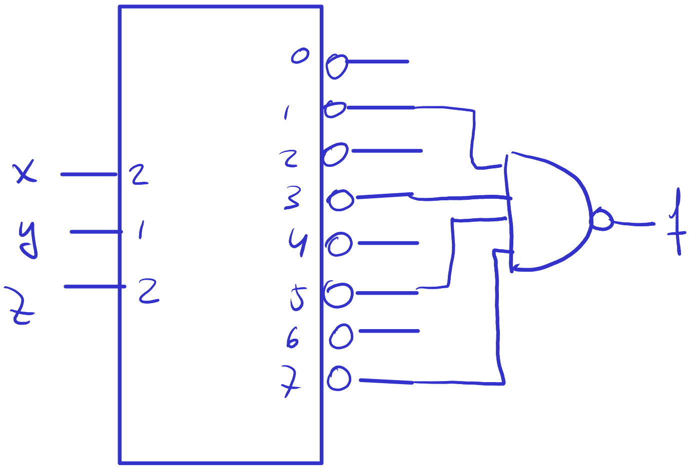


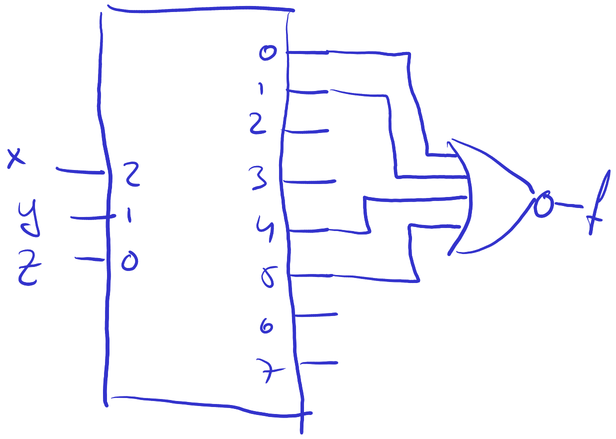


Example 3: f(x, y, z) = ∑(2, 3, 6, 7) = ∏(0, 1, 4, 5)
a) Design f using a DEC 3:8 (active-low output) and a NAND gate
b) Design f using a DEC 3:8 (active-high output) and a NOR gate.

AND – OR ≡ NAND – NAND

OR – AND
$\shortparallel$
NOR – NOR





$f(x,y,z) = \Sigma(1,3,5,7)$

Example 4

a) Design a MUX 8:1

b) Design a MUX 4:1 with an active-low enable input

$$z = \underset{1}{\overline{s1}}\,\underset{1}{\overline{s0}}\,d0 + \underset{1}{\overline{s1}}\,\underset{0}{\cancel{s0}}\,d1 + \underset{0}{s1}\,\underset{1}{\cancel{\overline{s0}}}\,d2 + \underset{0}{s1}\,\underset{0}{\cancel{s0}}\,d3$$

$s_1, s_0 = 0, 0 \rightarrow z = d_0$

$$z = (m_0 d_0 + m_1 d_1 + m_2 d_2 + m_3 d_3)\,\overline{e} =$$

$$\bar{e}\, m_0 d_0 + \bar{e}\, m_1 d_1 + \bar{e}\, m_2 d_2 + \bar{e}\, m_3 d_3$$

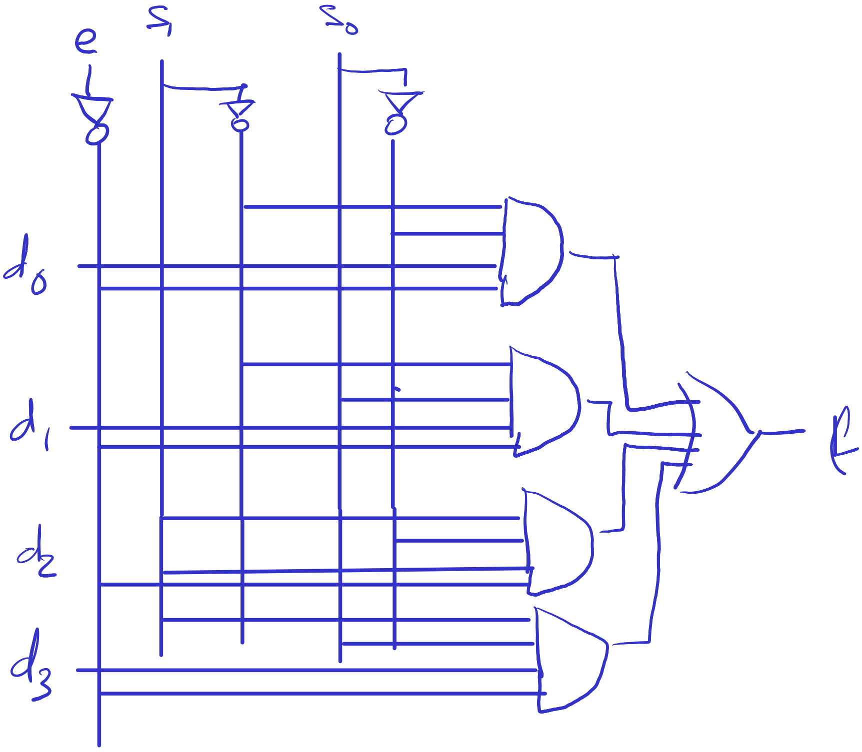
a) $z =$

8 data inputs

3 selection inputs.

$$z = m_0 d_0 + m_1 d_1 + \dots + m_7 d_7 =$$

$$= \bar{s}_2 \bar{s}_1 \bar{s}_0 d_0 + \bar{s}_2 \bar{s}_1 s_0 d_1 + \bar{s}_2 s_1 \bar{s}_0 d_2 + \dots + s_2 s_1 s_0 d_7$$



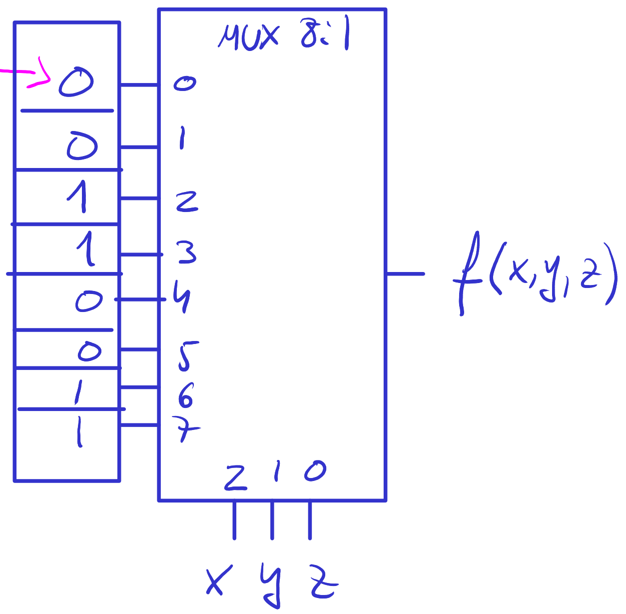
Example 5

Design f(x, y, z) = ∑(2, 3, 6, 7) with MUX 8:1

z \ xy	00	01	11	10
0	0	1	1	0
1	0	1	1	0

f

f(0,0,0) = 0



CLB

LUT

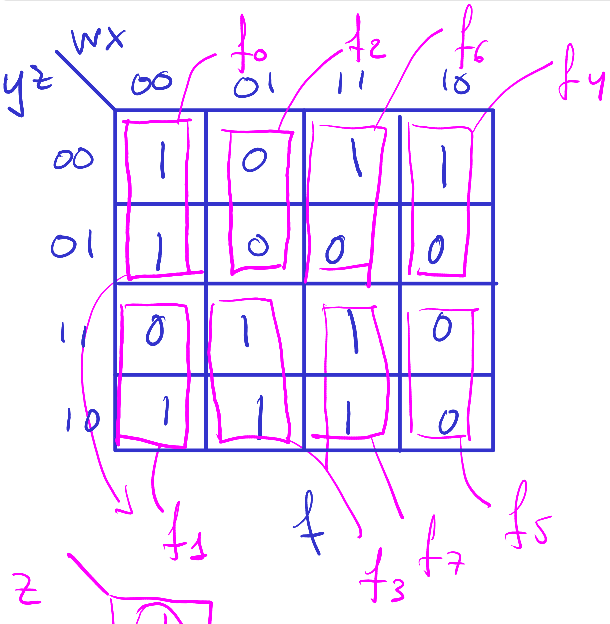
Example 6

Design f(w, x, y, z) = ∑(0, 1, 2, 6, 7, 8, 12, 14, 15)

a) with MUX 8:1

b) with MUX 4:1

a)



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

$f_0(z) = f(0,0,0,z)$

$f_1(z) = f(0,0,1,z)$

$f_2(z) = f(0,1,0,z)$

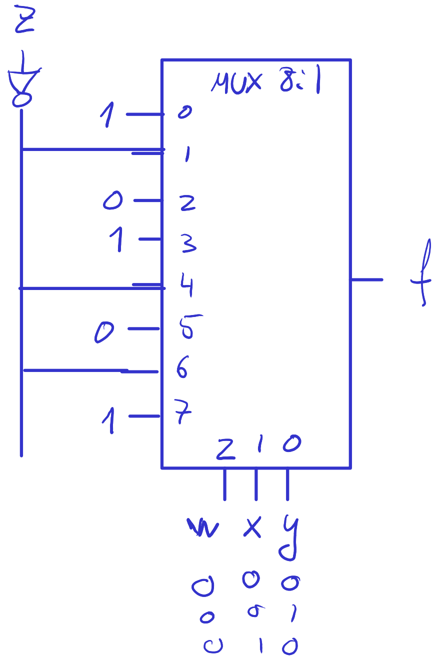
$f(0,0,0,z) = 1$

$f_0 = 1;\ f_1 = \bar{z}$

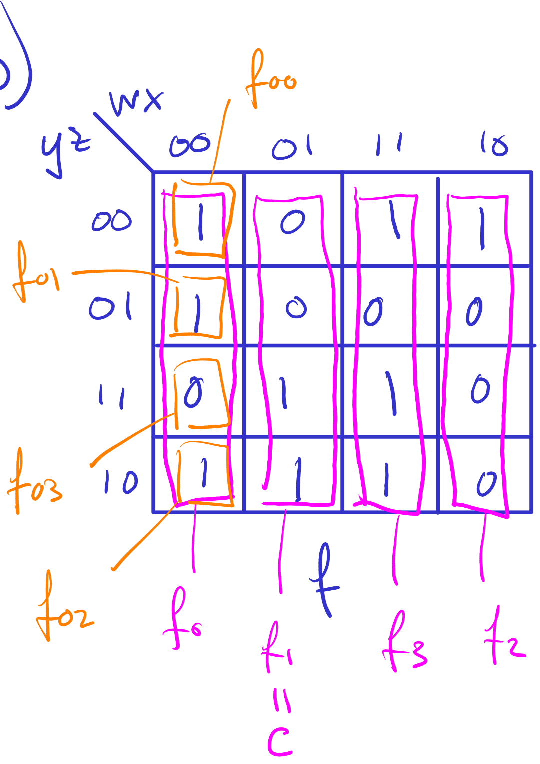
$f_2 = 0;\ f_3 = 1$

$f_4 = \bar{z};\ f_5 = 0$

$f_6 = \bar{z};\ f_7 = 1$



b)



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

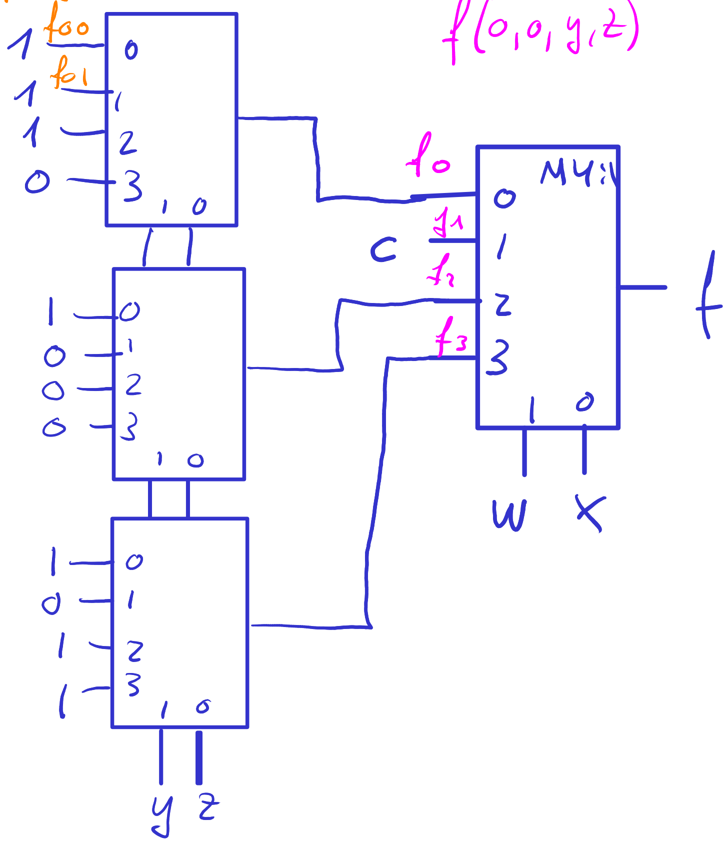
$f(0,\underset{||}{0},0,0)$

$f_0(y=0,z=0)$

$f(w=0,\ x=0)$

$||$

$f(0,0,y,z)$



Example 7

Design the following encoders using K-maps:

a) 4-bit binary encoder

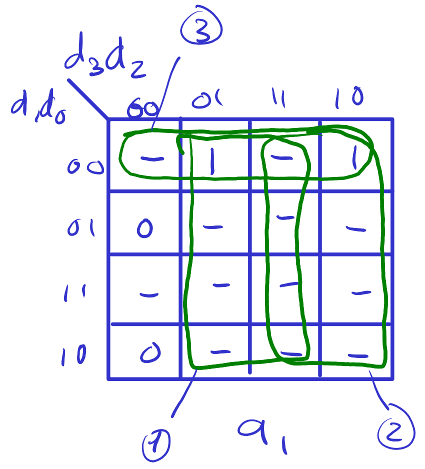
b) 4-bit Gray encoder

Add an enable input signal after completing the design.

a)

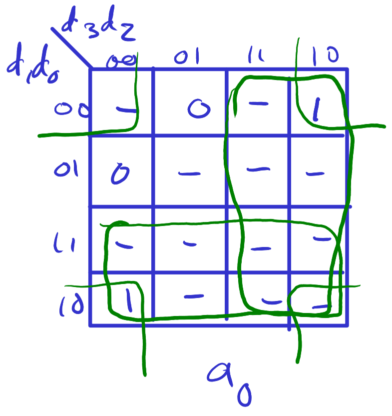
d_1d_0 \ d_3d_2	00	01	11	10
00	--	10	--	11
01	00	--	--	--
11	--	--	--	--
10	01	--	--	--

a_1a_0



d_1d_0 \ d_3d_2	00	01	11	10
00	-	1	-	1
01	0	-	-	-
11	-	-	-	-
10	0	-	-	-

a_1

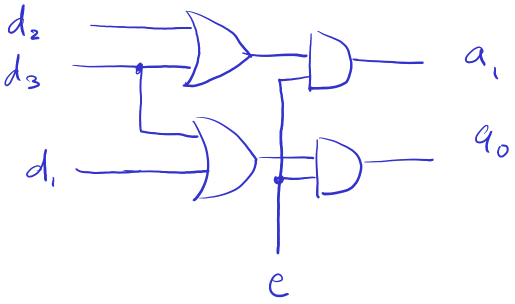


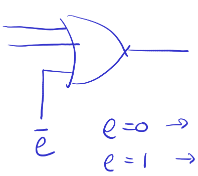
d_1d_0 \ d_3d_2	00	01	11	10
00	-	0	-	1
01	0	-	-	-
11	-	-	-	-
10	1	-	-	-

a_0

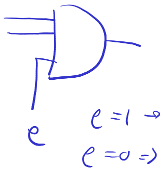
$$\boxed{a_1 = d_2 + d_3 \quad (2) \qquad a_0 = d_3 + d_1 \quad (2)}$$

$$a_1 = \overline{d_1}\,\overline{d_0} \quad (2+2) \qquad a_0 = \overline{d_2}\,\overline{d_0} \quad (2+2)$$





$e=0 \rightarrow$ enable
$e=1 \rightarrow$ "1"



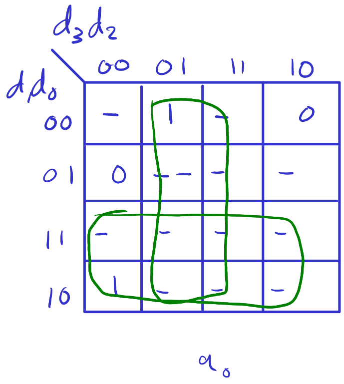
$e=1 \rightarrow$ enabled
$e=0 \Rightarrow$ "0"

n	bin	Gray
0	00	00
1	01	01
2	10	11
3	11	10

b)

d_1d_0 \ d_3d_2	00	01	11	10
00	--	11	--	10
01	00	--	--	--
11	--	--	--	--
10	01	--	--	--

a_1a_0

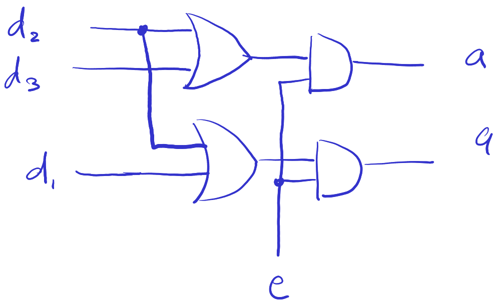


d_1d_0 \ d_3d_2	00	01	11	10
00	-	1	-	0
01	0	--	-	-
11	-	-	-	-
10	1	-	-	-

a_0

$$a_0 = d_2 + d_1$$

$$a_1 = d_2 + d_3$$

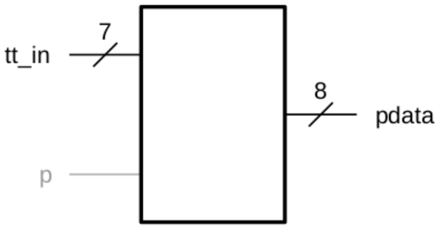


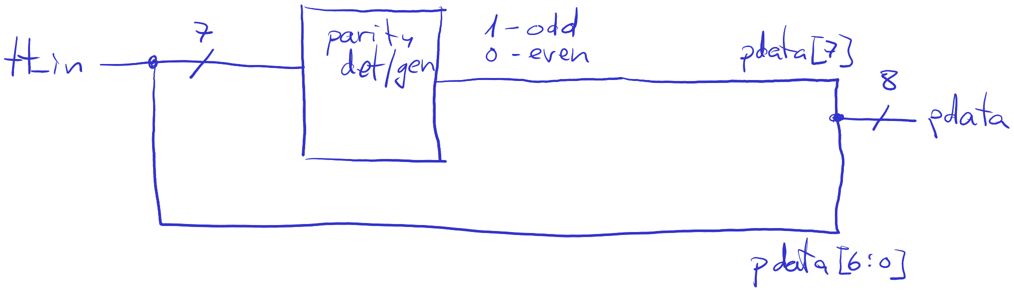
Example 8

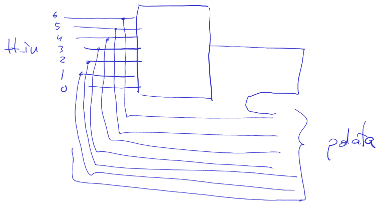
We need to design an interface module for an old teletype system in order to connect it to a standard serial computer port. The teletype generates 7-bit ASCII character codes while our computer can only receive 8-bit words (bytes) that must have even parity.

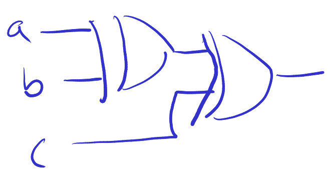
a) Design a module that reads 7-bit words through a 'tt_in' input signal and generates 8-bit words with leading even parity bit in output signal 'pdata' (bit 7 in pdata is the parity bit and the rest of the bits are copied from tt_in)

b) Add a control signal 'p' to the system to select the parity of the generated output so that the output will be even when p=0 and odd when p=1.

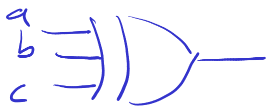




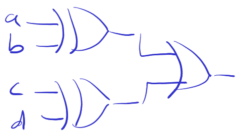


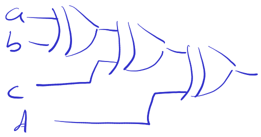


$\overset{?}{\equiv}$



$x \oplus y = \bar{x}y + x\bar{y}$

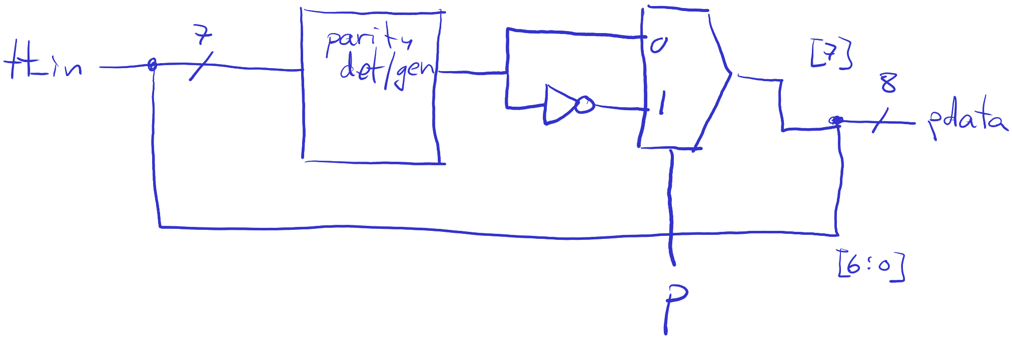


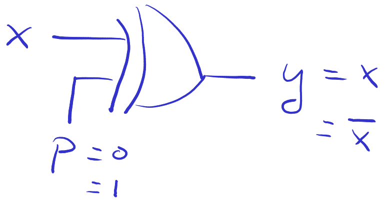


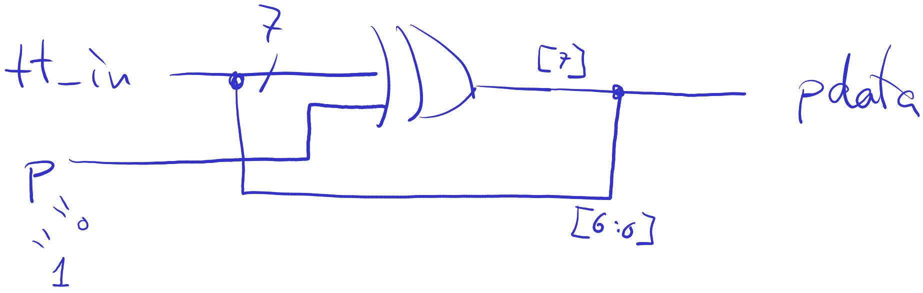
$(a\oplus b)\oplus c \overset{?\checkmark}{=} a\oplus(b\oplus c)$

$a\oplus b\oplus c$

$x \oplus y = \bar{x}y + x\bar{y}$





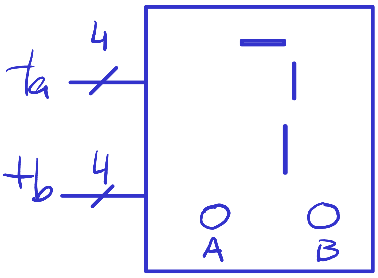


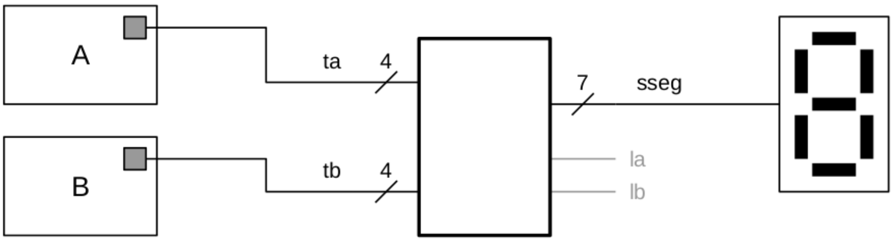
Example 9

Two experiments are carried out in two rooms A and B. It is important to know the maximum temperature achieved in both rooms. Temperature sensors provide digital reading of the temperature through 4-bit signals 'ta' and 'tb' that range from 0 to 9. We need a circuit that displays the temperature achieved in the room with the highest temperature.

a) Design a digital circuit using standard combinational subsystems that generates the 7-segment code 'sseg' for the temperature in the room with the highest temperature.

b) Add two additional output to the system 'la' and 'lb' to control two LEDs. 'la' should be '1' when ta>=tb, and 'lb should be '1' when tb>=ta.



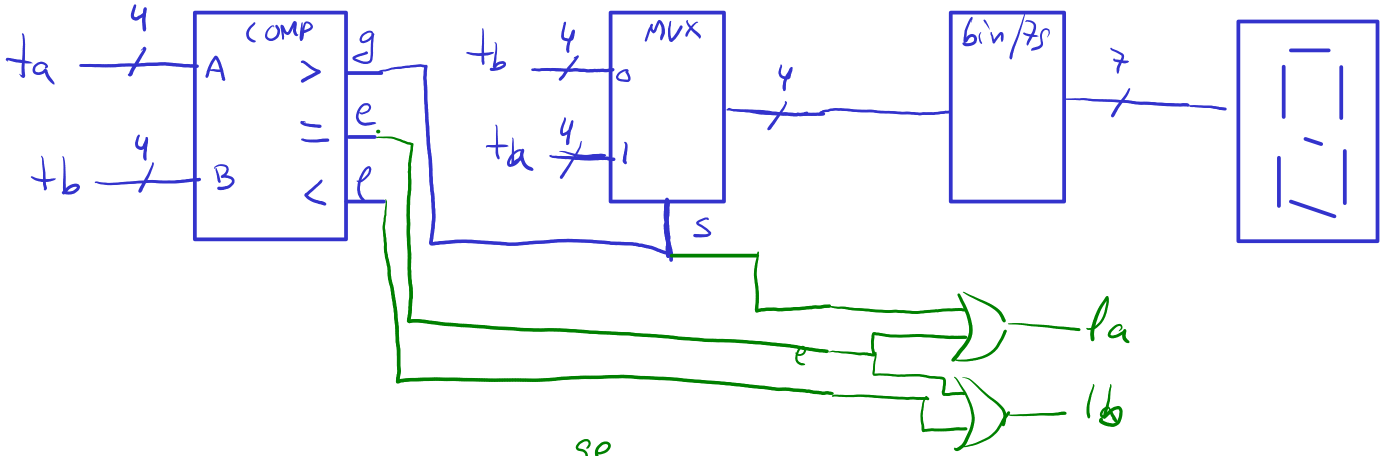


compare ta, tb → comparator

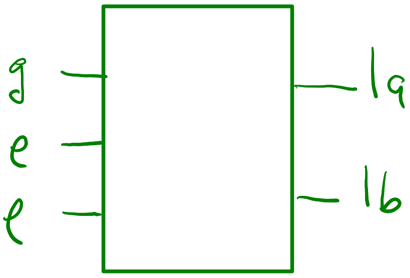
select ta or tb → multiplexer

convert from n.binary to 7s. → bin/7s converter

a)



b)



l \ ge	00	01	11	10
0		1	1	1
1		1	1	1

$la = g + e$

l \ ge	00	01	11	10
0		1	1	
1	1	1	1	1

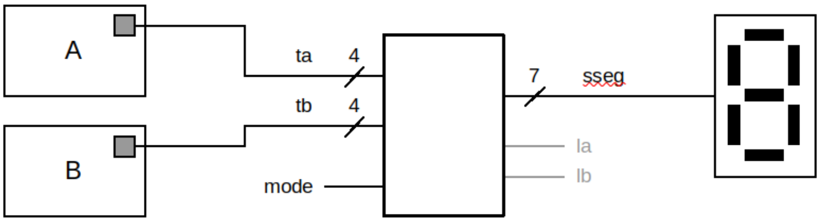
$lb = e + \ell$

Example 10

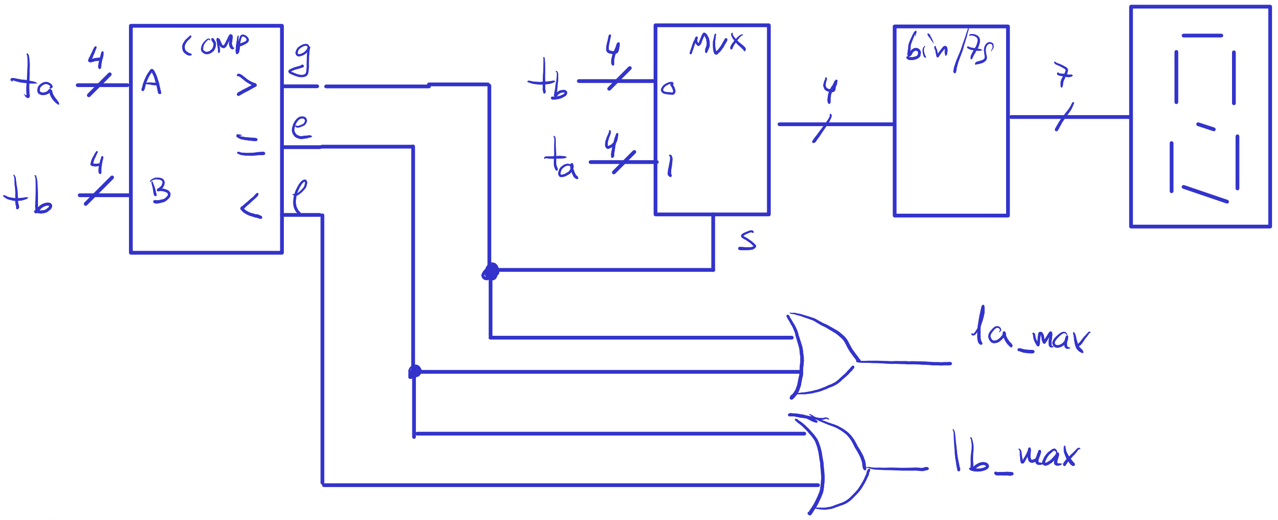
Add a control signal 'mode' to the circuit in the previous example so that:

- When mode=0, the temperature shown and the active LED correspond to the maximum (as in the previous example).
- When mode=1, the temperature shown and the active LED corresponds to the minimum.

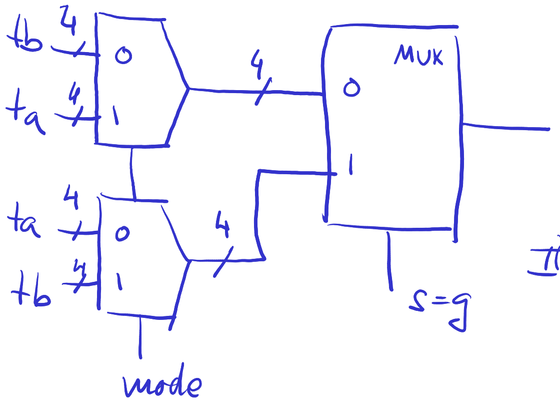
In both cases, both LEDs should be on when the temperatures in A and B are the same.

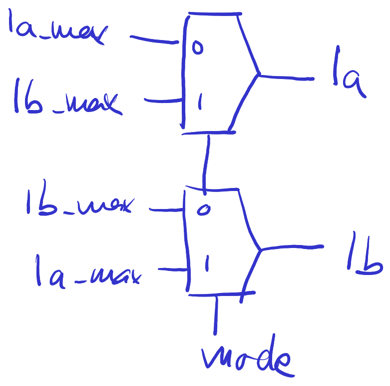


mode = 0 | mode = 1

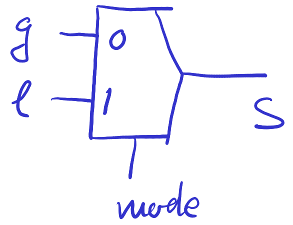


I)

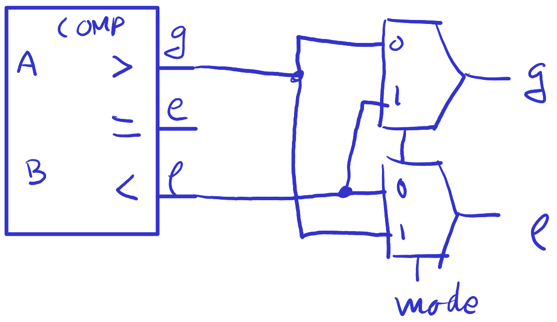




II)



III)

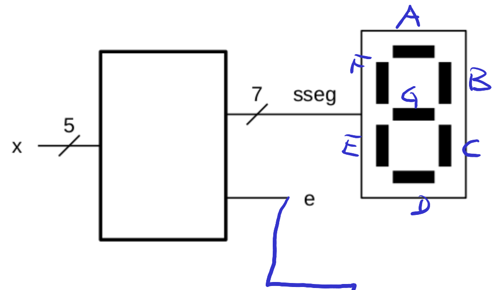


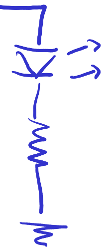
Example 11

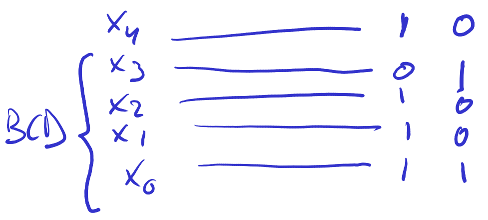
A computation system receives BCD digits through an input 'x' of 5 bits, where the most significant bit is an even parity bit.

a) Design a circuit that tests the parity of the input number and displays the number in a 7-segment display. An error output 'e' will be activated when the parity is not correct or the input number is not a BCD digit.

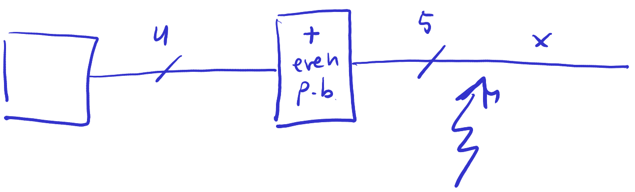
b) Modify the design so that when there is an error, the displays shows the symbol corresponding to number 14 ($1110_{(2} = E_{(16}$).

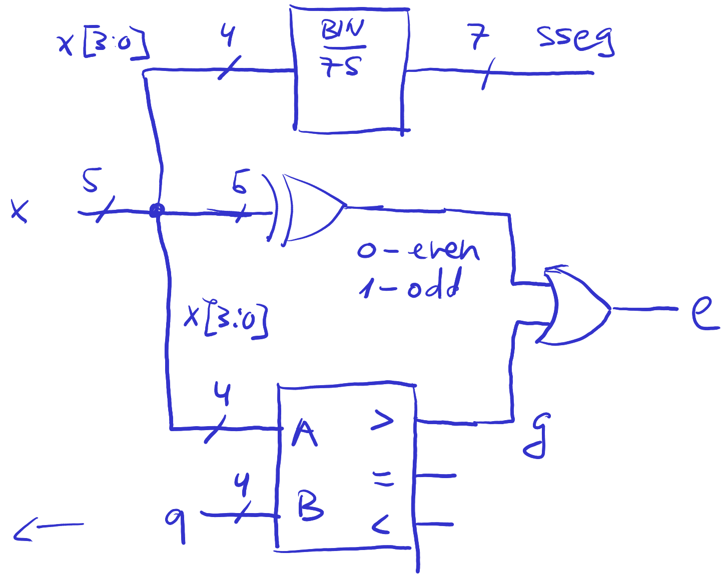


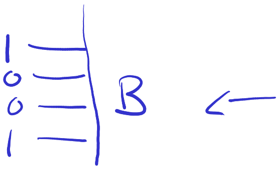




→ even parity bit







parity detector

x[3:0] > 9 ⇒ error

⇑

OR

parity of x is odd

x	sseg	e
10111	7	0
01001	9	0
10011	~~8~~	1
1 $\underbrace{1101}_{\cancel{13}}$	d	1

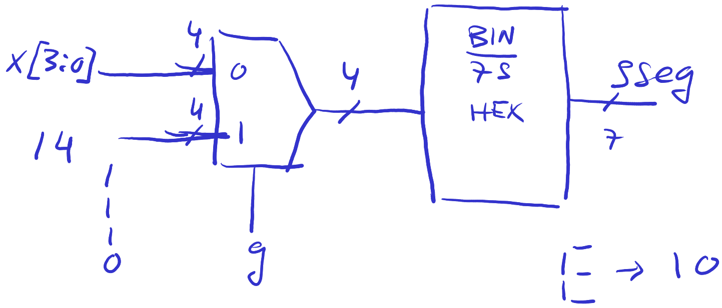
BCD

b) x = 11101 ⇒ e = 1 , E $14_{(10} = E_{(16}$

sseg : if x[3:0] > 9 ⇒ convert 14
else → convert x[3:0]

selection ⇒ MUX !

I)



E → 1001111

II)