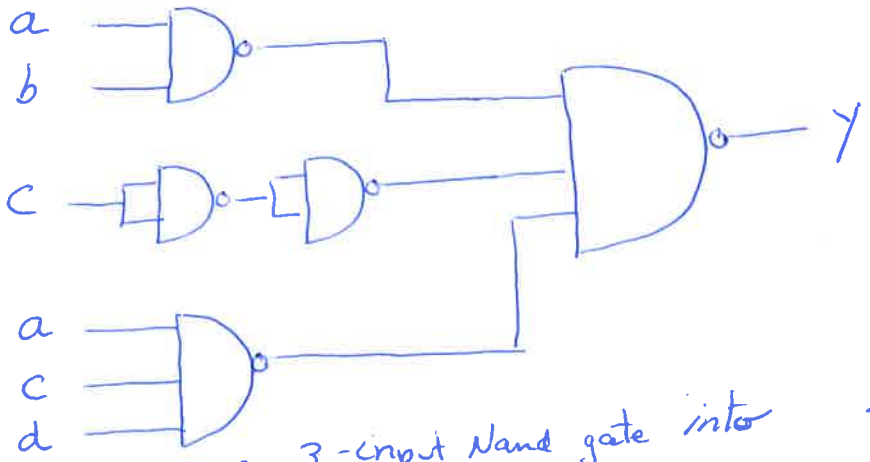
Problem 1 (option A)

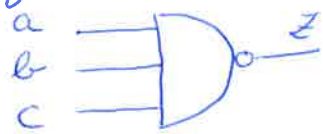
a) $Y = ab + c' + acd$

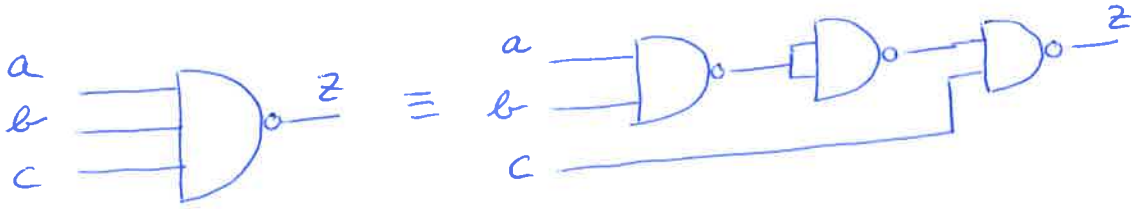
$Y = (ab + c' + acd)''$

$Y = [(ab)' \cdot c'' \cdot (acd)']'$

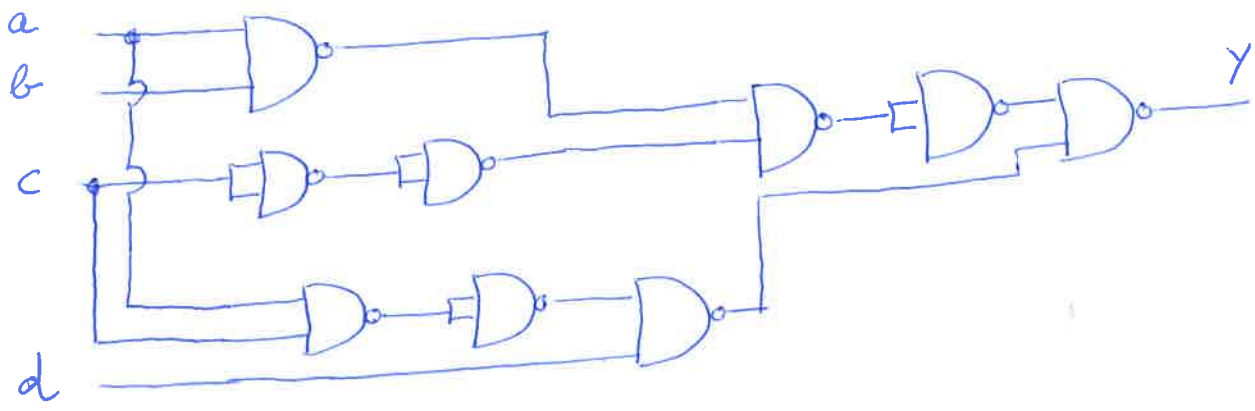


We transform a 3-input Nand gate into 2-input Nand gates.





Finally:



b)

$$y = ab + c' + acd$$

$ab = ab(c+c') = abc + abc'$

$ab = abc(d+d') + abc'(d+d') =$

$abcd + abcd' + abc'd + abc'd'$

m_{1111} m_{1110} m_{1101} m_{1100}

$c' = c'(a+a') = c'a + c'a' = ac' + a'c'$

$ac' = ac'(b+b') = abc' + ab'c'$

$ac' = abc'(d+d') + ab'c'(d+d') =$

$abc'd + abc'd' + ab'c'd + ab'c'd'$

$a'c' = a'c'(b+b') = a'bc' + a'b'c'$

$a'c' = a'bc'(d+d') + a'b'c'(d+d') =$

$a'bc'd + a'bc'd' + a'b'c'd + a'b'c'd'$

m_{1101} m_{1100} m_{1001} m_{1000}

$c' = abc'd + abc'd' + ab'c'd + ab'c'd' +$

$+ a'bc'd + a'bc'd' + a'b'c'd + a'b'c'd'$

m_{0101} m_{0100} m_{0001} m_{0000}

$acd = acd(b+b') = abcd + ab'cd$

m_{1011}

$$y = m_{15} + m_{14} + m_{13} + m_{12} + m_{11} + m_9 + m_8 + m_5 + m_4 + m_1 + m_0$$

$$Y = \sum_4 m(0, 2, 4, 5, 8, 9, 11, 12, 13, 14, 15)$$

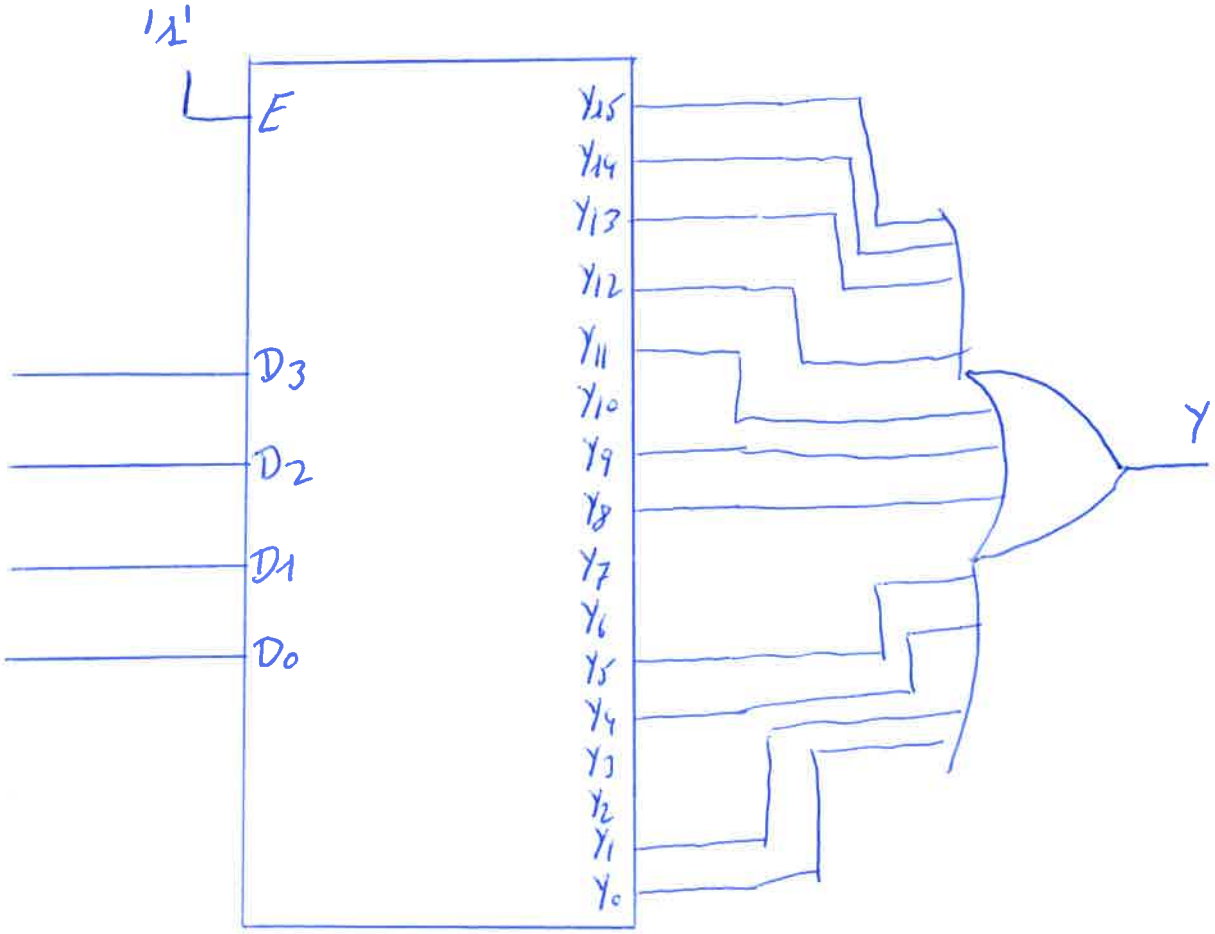
$$Y = \prod_4 M(2, 3, 6, 7, 10)$$

$$Y = (a+b+c'+d)(a+b+c'+d')(a+b'+c'+d)(a+b'+c'+d')(a'+b+c'+d)$$

Truth table:

a	b	c	d	Y
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	0
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	1
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

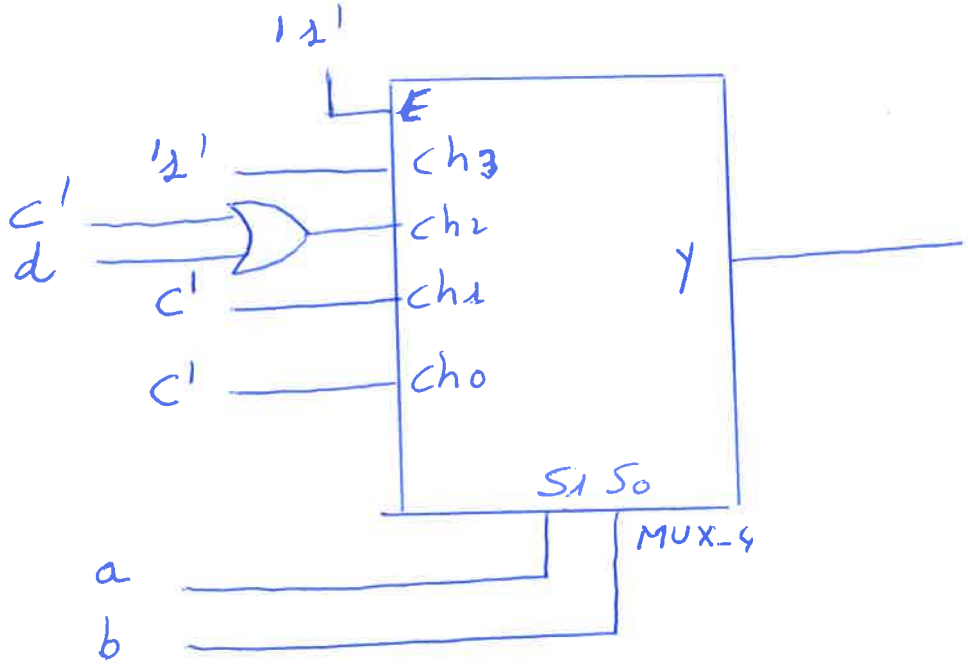
c) With a decoder 4-16 we implement the minterms of the logic function.



DEC-4-16

d)

	a	b	c	d	Y	
ch0	0	0	0	0	1	c'
	0	0	0	1	1	
	0	0	1	0	0	
	0	0	1	1	0	
ch1	0	1	0	0	1	c'
	0	1	0	1	1	
	0	1	1	0	0	
	0	1	1	1	0	
ch2	1	0	0	0	1	c'+d
	1	0	0	1	1	
	1	0	1	0	0	
	1	0	1	1	1	
ch3	1	1	0	0	1	1
	1	1	0	1	1	
	1	1	1	0	1	
	1	1	1	1	1	

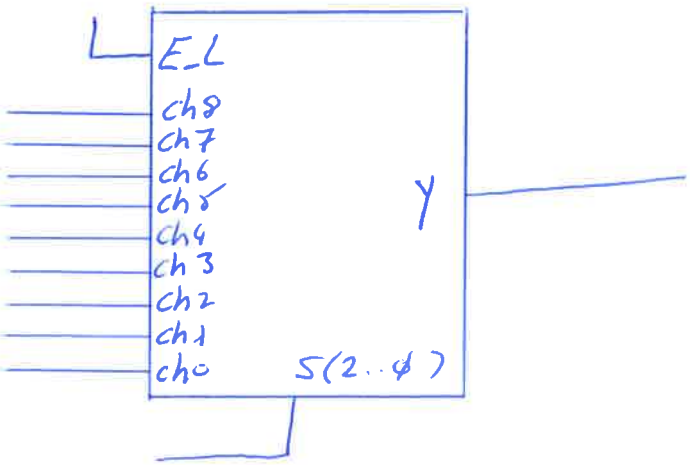


Problem 2

a) It is a hierarchical design because the circuit contains several components.

The project contains 3 VHDL files: The top (MUX-8), and MUX-4 and MUX-2.

b) The symbol of the top entity corresponds to a MUX-8.



c) The truth table of the top entity is:

E_L	S_2	S_1	S_0	ch_7	ch_6	ch_5	ch_4	ch_3	ch_2	ch_1	ch_0	Y
1	–	–	–	–	–	–	–	–	–	–	–	0
0	0	0	0	–	–	–	–	–	–	–	0	0
0	0	0	0	–	–	–	–	–	–	–	1	1
0	0	0	1	–	–	–	–	–	–	0	–	0
0	0	0	1	–	–	–	–	–	–	1	–	1
0	0	1	0	–	–	–	–	–	0	–	–	0
0	0	1	0	–	–	–	–	–	1	–	–	1
0	0	1	1	–	–	–	–	0	–	–	–	0
0	0	1	1	–	–	–	–	1	–	–	–	1
0	1	0	0	–	–	–	0	–	–	–	–	0
0	1	0	0	–	–	–	1	–	–	–	–	1
0	1	0	1	–	–	0	–	–	–	–	–	0
0	1	0	1	–	–	1	–	–	–	–	–	1
0	1	1	0	–	0	–	–	–	–	–	–	0
0	1	1	0	–	1	–	–	–	–	–	–	1
0	1	1	1	0	–	–	–	–	–	–	–	0
0	1	1	1	1	–	–	–	–	–	–	–	1

Number of Minterms = $8 \cdot 2^7 = 1024$ Minterms.

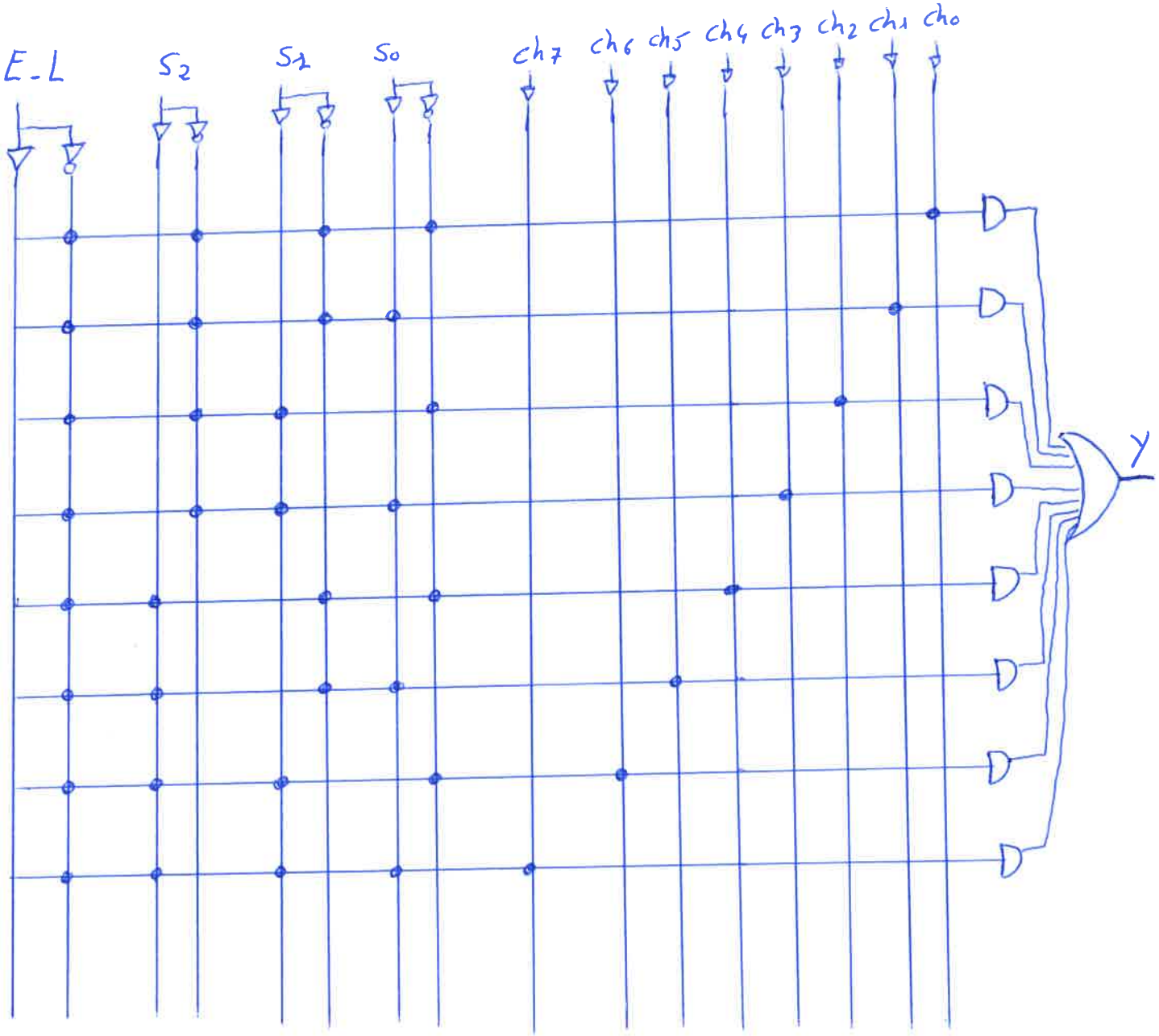
Number of Maxterms = $2^{11} + 8 \cdot 2^7 = 3072$ maxterms.

d)

$$Y = E\_L' S_2' S_1' S_0' ch0 + E\_L' S_2' S_1' S_0 ch1 + E\_L' S_2' S_1 S_0' ch2 +$$

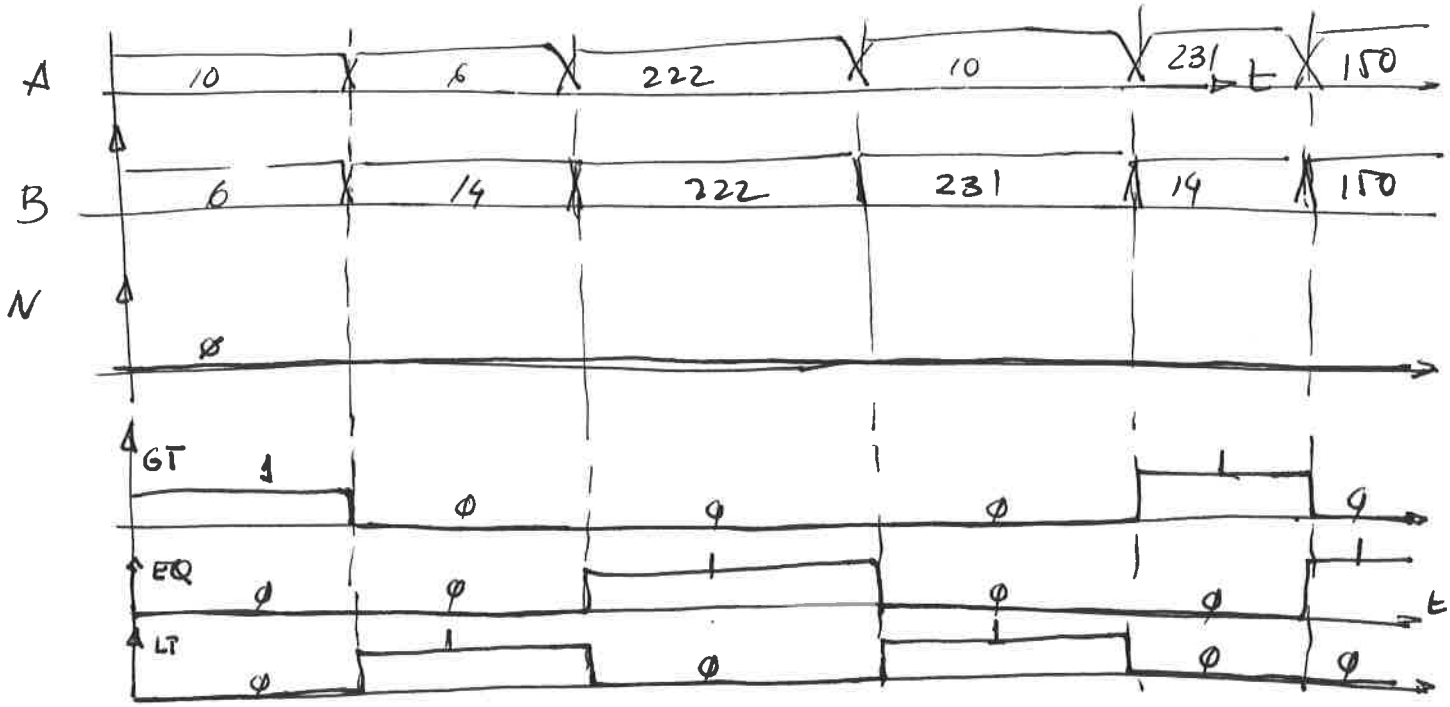
$$+ E\_L' S_2' S_1 S_0 ch3 + E\_L' S_2 S_1' S_0' ch4 + E\_L' S_2 S_1' S_0 ch5 +$$

$$+ E\_L' S_2 S_1 S_0' ch6 + E\_L' S_2 S_1 S_0 ch7.$$



Problem 3

a)



A	B	N	GT	EQ	LT
10	6	0	1	0	0
6	14	0	0	0	1
222	222	0	0	1	0
10	231	0	0	0	1
231	14	0	1	0	0
150	150	0	0	1	0
+10	+6	1	1	0	0
+6	+14	1	0	0	1
-34	-34	1	0	1	0
+10	-25	1	1	0	0
-25	+14	1	0	0	1
-106	-106	1	0	1	0

b)

etc.

unsigned decimal

$10 \rightarrow (0000\,1010)_2$

$14 \rightarrow (0000\,1110)_2$

$6 \rightarrow (0000\,0110)_2$

$222 \rightarrow (1101\,1110)_2$

$231 \rightarrow (1110\,0111)_2$

$150 \rightarrow (1001\,0110)_2$

signed decimal

$+10 \leftrightarrow [0]\,000\,1010$

$+14 \leftrightarrow [0]\,000\,1110$

$+6 \leftrightarrow [0]\,000\,0110$

$-34 \leftarrow [1]\,101\,1110$

$-25 \leftarrow [1]\,101\,0111$

$-106 \leftarrow [1]\,001\,0110$

$[0]\,110\,1001$

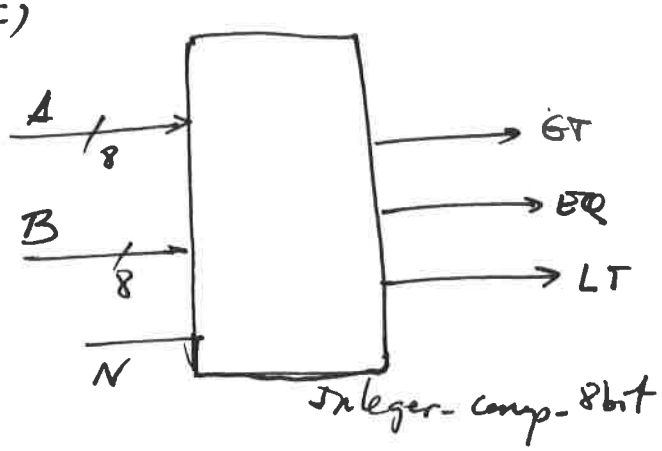
$+1$

$[0]\,110\,1010$

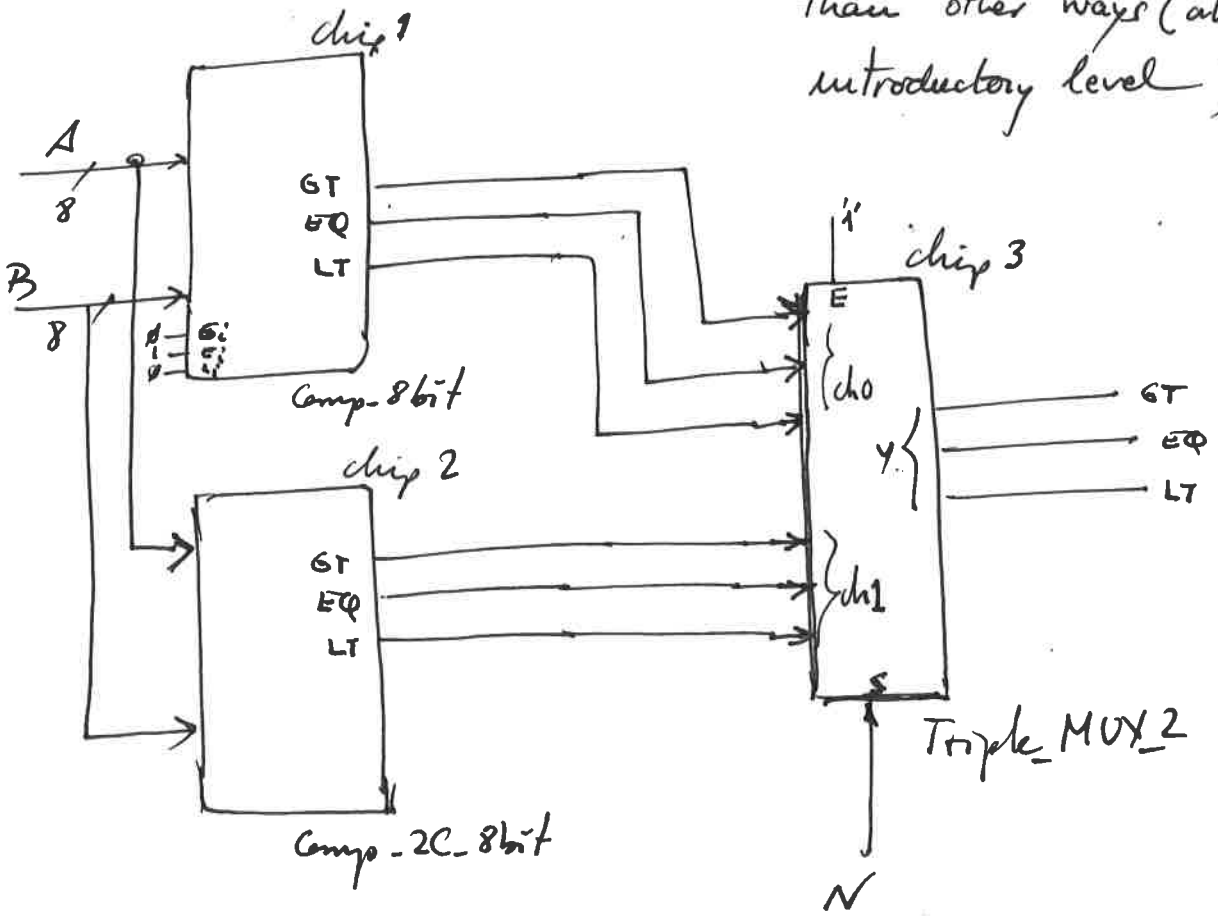
$+106$ — -106

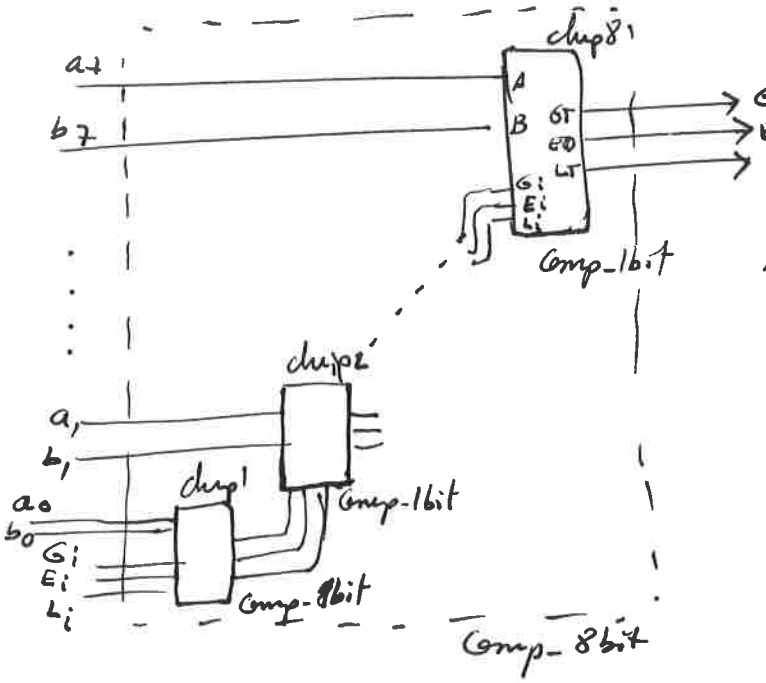
→ The idea is that with a $N=8$ it is only possible to represent signed integers in 2C from $-128 \leq A, B, \leq +127$ and unsigned integers from $0 \leq A, B \leq 255$

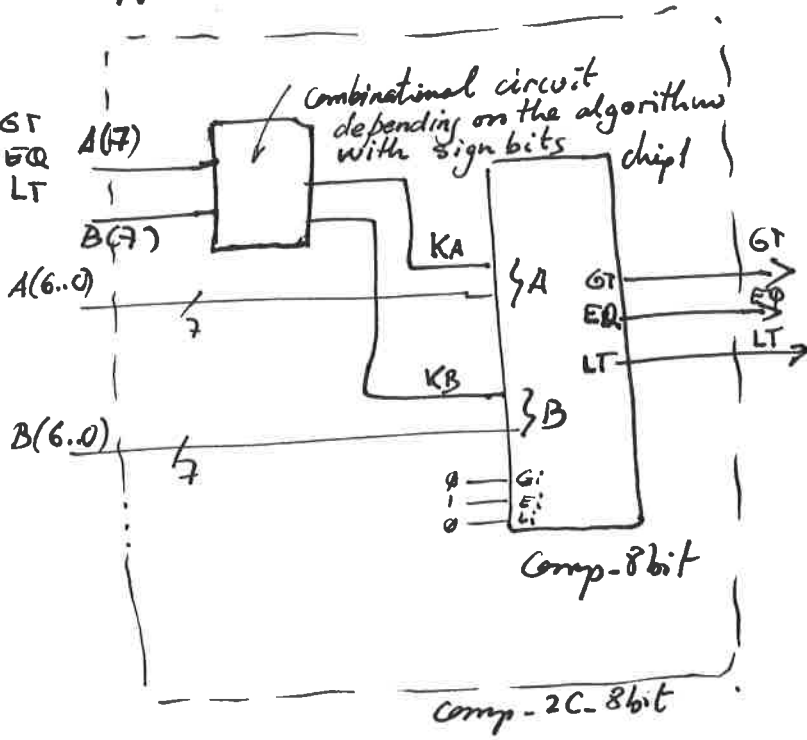
c)



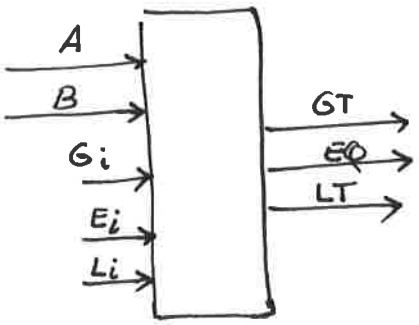
This circuit may be designed using the plan C2 because it is too large and basing it on smaller chips is simpler than other ways (at this introductory level)







d)



Comp_1bit

A	B	Gi	Ei	Li	GT	EQ	LT
1	Ø	-	-	-	1	Ø	Ø
Ø	1	-	-	-	Ø	Ø	1
1	1	1	Ø	Ø	1	Ø	Ø
1	1	Ø	1	Ø	Ø	1	Ø
1	1	Ø	Ø	1	Ø	Ø	1
Ø	Ø	1	Ø	Ø	1	Ø	Ø
Ø	Ø	Ø	1	Ø	Ø	1	Ø
Ø	Ø	Ø	Ø	1	Ø	Ø	1

$2^5 \rightarrow$ 32 different combinations

(Some of them don't care, like ØØ11Ø, ØØ101, ØØØ11, ⋮)

- Plan A: gate equations
- Plan B: truth table or algorithm
- Plan C2: using components

Plan A:
- → Canonical (maxterms / minterms)
- → SoP / PoS (using minilog)
- → any kind of equation

Thus, if Plan A has to be used, and canonical equations based on maxterms, we have to find how many maxterms GT, EQ and LT have.

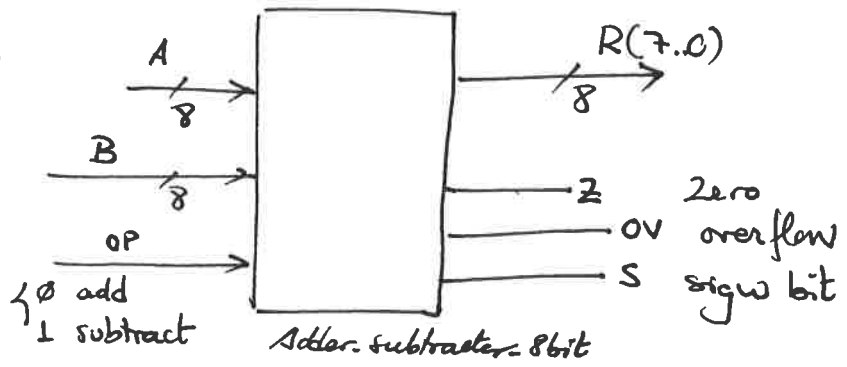
Ø 1 x x x	Ø	→ $M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}, M_{14}, M_{15}$
1 1 Ø 1 Ø	Ø	→ M_{26}
1 1 Ø Ø 1	Ø	→ M_{25}
Ø Ø Ø 1 Ø	Ø	→ M_2
Ø Ø Ø Ø 1	Ø	→ M_1

⇒ 12 maxterms

(assuming that other terms don't care are considered '1', like ØØ11Ø, ØØ101, ØØ111, ⋮)

Problem 4

a)



A, B, R ⇒ signed integers in 2C

$-128 \leq A, B, R \leq +127$

$A = \boxed{0}1010101$

$B = 0111\,0011$

OP = 0 → add

$$\begin{array}{r} 85 \\ +115 \\ \hline 200 \end{array}$$ ← out of range

$R = \boxed{1}1001000$ → overflow OV = 1 out of range

→ Z = 0

→ S = 1

a) A + B

---

b) $\boxed{0}1010101$

$\boxed{0}1110011$

OP = 1 subtract

$$\begin{array}{r} 85 \\ -115 \\ \hline -30 \end{array}$$

R = A − B = A + 2C(B)

$\boxed{1}0001100$

$+\,1$

$\boxed{1}0001101$

$\boxed{0}1010101$

$+\ \boxed{1}0001101$

$R = \boxed{1}1100010$ →

III

−30 ✓

$0001110\,1$

$+\,1$

$0001111\,0$ → +30

OV = 0, Z = 0, S = 1

---

c) $\boxed{1}0101111$

$+\ \boxed{0}1111100$

OP = 0 →

$$\begin{array}{r} -81 \\ +124 \\ \hline +43 \end{array}$$

$R = \boxed{0}0101011$ → +43

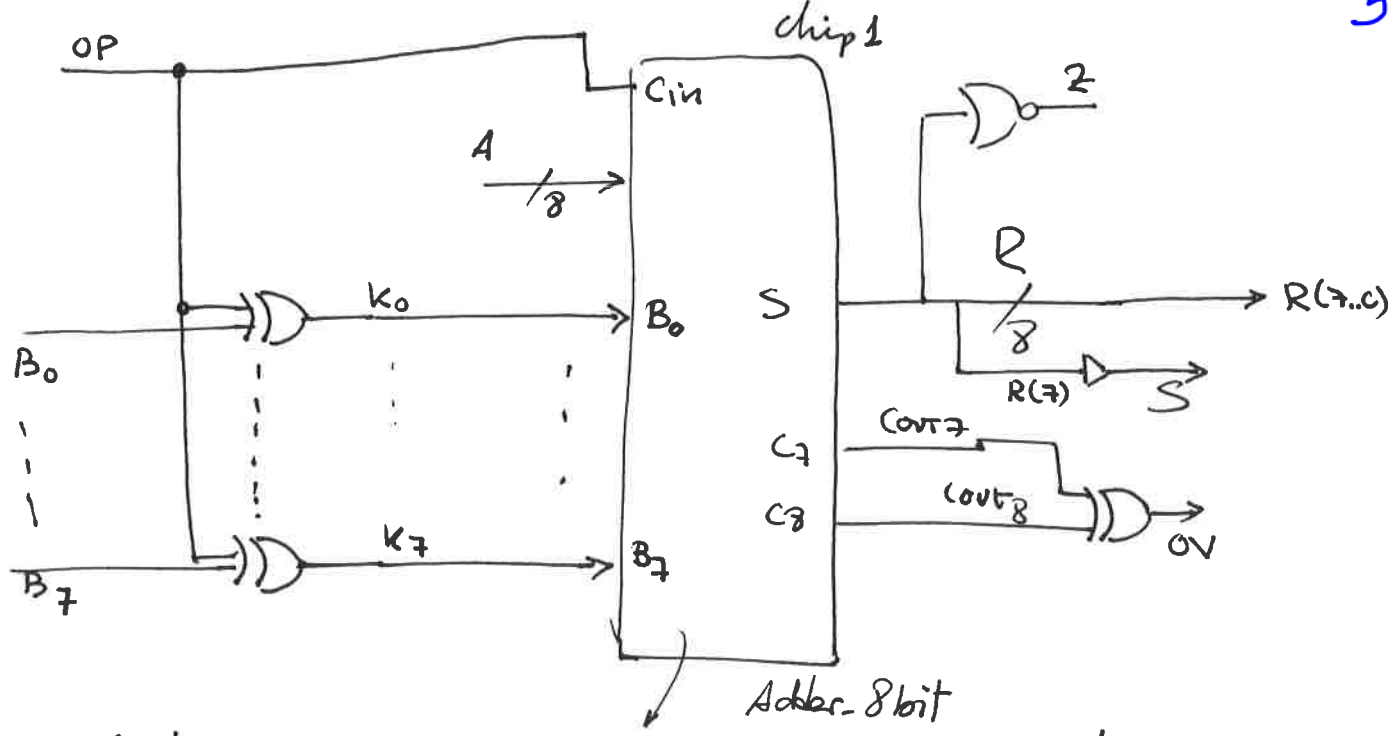
OV = 0, Z = 0, S = 0

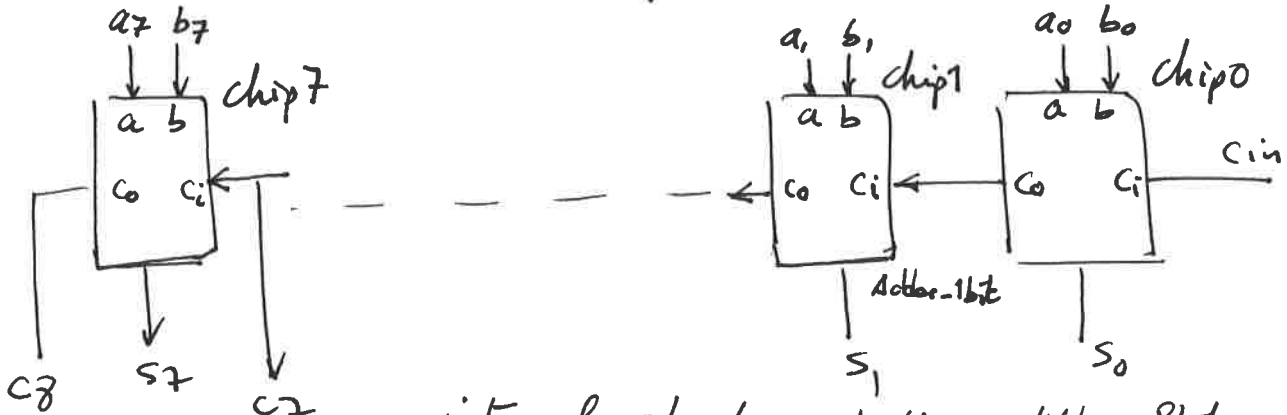
$\boxed{0}1010000$

$+\,1$

$\boxed{0}1010001$ →

b)





internal structure of the Adder_8bit based on ripple carry

c) overflow is detected when $C_7 \neq C_8$ which is a function XOR $OV = \underline{C_7 \text{ XOR } C_8}$

d) From the timing diagram: $t_p = 475878{,}963\,ns - 475869{,}983\,ns$

$t_p = 8{,}98\,ns$

$f_{max} < \frac{1}{t_p} = \underline{111{,}36\,MHz}$ millions of operations per second

{ Number of gate levels aproximately which affect the propagation delay
→ assuming a worst-case scenario

Adder_1bit → 3
Adder_8bit → 3 × 8 → 24
Adder_subtractor_8bit
from OP to OV → 1 + 24 + 1 → 26

$td_{1gate} \approx \frac{8.98\,ns}{26} \approx \underline{0.34\,ns}$