Solution TD1

1/ Exercice 1 : 8 bits → 1 Byte → KB → MB → GB → TB

→ 256 B = 256 × 8 = 2048 bits

→ 4 KB = 4 × 1024 = 4096 bits

→ 3 GB = 3 × 1024 × 1024 = 3145728 KB

→ 1 TB = 1024 × 1024 × 1024 × 1024 $= 2^{40}$ B

($\downarrow 2^{10}$; $\downarrow 2^{10}$; $\downarrow 2^{10}$; $\downarrow 2^{10}$)

$$\begin{cases} 1\,KB = 2^{10}\,B = 1024\,B \\ 1\,MB = 2^{10}\,KB = 1024\,KB \\ 1\,GB = 2^{10}\,MB = 1024\,MB \\ 1\,TB = 2^{10}\,GB = 1024\,GB \end{cases}$$

word 8 bits = 1 Byte
double-word = 16 bits
= 2 Bytes

2/ An 8bit code can have each bit either 0 or 1

Total number of different combination
$= 2^{\text{number of bits}} = 2^8 = 256$ different instruction

→ Days of the week : There are 7 days in a week

$2^n \geq N$ (number of elements to encode)

$2^n \geq 7 \Rightarrow 2^{\boxed{3}} = 8 \geq 7$

Minimum bits needed : 3 bits

→ Days of month : Usually 30 or 31 days (to be 31 (max))

$2^n \geq 31$

$2^4 = 16 < 31$ ✗ , $2^{\boxed{5}} = 32 > 31$ ✓

Minimum bits needed : 5 bits

Exercice 2:

→ $(1101.01001)_2 = (?)_{10}$

$= 1\times2^8 + 1\times2^7 + 0\times2^6 + 1\times2^5 + 0\times2^4 + 1\times2^3 + 0\times2^2 + 0\times2 + 1$

$256 \quad 128 \quad 32 \quad 8$

$= (425)_{10}$

→ $(65)_8 = (?)_{16}$ we convert octal to binary

step 1 : $(6)_8 = (110)_2$ $(5)_8 = (101)_2$

$(65)_8 = (110\ 101)_2$

step 2: $(110101)_2 = (?)_{16}$

$(0011\ 0101)_2$

No, we convert each 4-bits group to hex:

$(0011)_2 = (3)_{16}$

$(0101)_2 = (5)_{16}$ ⇒ $(35)_{16}$

→ $(25)_{10} = (?)_2$

25 | 2
1 | 12 | 2
0 | 6 | 2
0 | 3 | 2
1 | 1 | 2
1 | 0

$= (11001)_2$

→ $(1907)_{10} = (?)_{16}$

1907 | 16
3 | 119 | 16
7 | 7 | 16
7 | 0

$= (773)_{16}$

→ $(A5B2)_{16} = (?)_2$ we convert each digit to 4 binary bits

$A_{16} = (1010)_2$, $5_{16} = (0101)_2$, $B_{16} = (1011)_2$, $2_{16} = (0010)_2$

$= (1010\ 0101\ 1011\ 0010)_2$

→ $(1111\ 1111)_2 = (?)_{10} = (?)_{16}$

$(255)_{10} = (ff)_{16}$

$(1111)_2 = F_{16}$ { $1\ 1\ 1\ 1$; $2^3\ 2^2\ 2^1\ 2^0$; $8+4+2+1 = 15$ in Hexa : F

→ $(8473)_{10} = (?)_2 = (?)_{16}$

8473	16
9	529
	16
1	33
	16
1	2
	16
2	0

$= (2119)_{16} = (?)_2$

$(0010\ 0001\ 0001\ 1001)_2$

→ $(EF1)_{16} = (?)_2 = (?)_{10}$

$E_{16} = 1110_2$
$F_{16} = 1111_2$
$1_{16} = 0001_2$
} $= (1110\ 1111\ 0001)_2$

$E_{16} = 14$, $F_{16} = 15$

$= 14 \times 16^2 + 15 \times 16^1 + 1 \times 16^0$

$= 3584 + 240 + 1$

$= (3825)_{10}$

→ $(208)_{10} = (?)_8 = (?)_{16}$

Decimal to octal (we divise by 8)

208	8
0	26
	8
2	3
	8
3	0

$= (320)_{10}$

208	16
0	13
	16
13	0

D

$= (D0)_{16}$

→ $(15,3125)_{10} = (?)_2$

15 | 2
1 | 7 | 2
1 | 3 | 2
1 | 1 | 2
1 | 0

$= (1111)_2$

0,3125 (convert the fractional part)

$0,3125 \times 2 = 0,625 \rightarrow 0$
$0,625 \times 2 = 1,25 \rightarrow 1$
$0,25 \times 2 = 0,5 \rightarrow 0$
$0,5 \times 2 = 1,0 \rightarrow 1$
$= (0,0101)_2$

final result: $(1111, 0101)_2$

F 3 9
+ 6 8 B
= 15 12 4
$= (15\ C\ 4)_{16}$

2 C F 3
+ 1 2 B
= 2 13 1 4
$(2\ D\ 1\ E)_{16}$ directly

$(1\ 1\ 0\ 1\ 0)_2$
$+(1\ 1\ 1\ 1\ 1)_2$
$= (10\ 1001)_2$

$(11010 + 1111) + 1010 + 111 = 11010 + 1111 + 1010 + 111$
$+ \ldots + \ldots$

11010 ↷ +
1111 ↷ +
1010 ↷ + then
111 ↷ + Then
= 110110

$(1100101)_2 - (1001010)_2 = (11011)_2$

1 1 0 0 1 0 1
− 1 0 0 1 0 1 0
0 0 1 1 0 1 1

→ A C74 − B3F = (?)$_{16}$

A C 7 4
− 0 B 3 F
=

10 12 7 4 +16
− 0 11 3 15
$(A\ 1\ 3\ 5)_{16}$

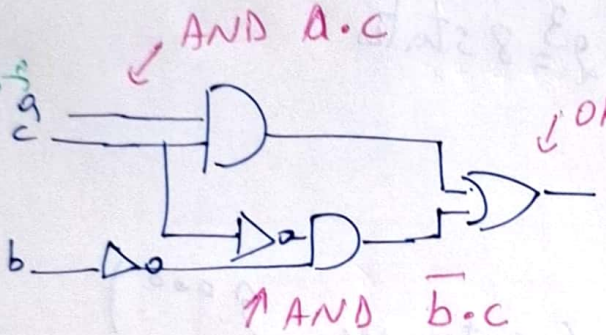
AND $\begin{cases} 1101 \\ 1011 \end{cases} \Rightarrow (1001)_2 = (9)_{10}$

OR $\begin{cases} 1101 \\ 1011 \end{cases} \Rightarrow (1111)_2 = (15)_{10}$

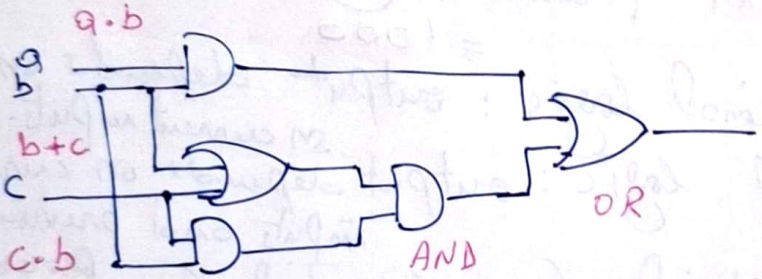
XOR $\begin{cases} 1101 \\ 1011 \end{cases} \Rightarrow (0110)_2 = (6)_{10}$

Exercice 3 :

1/



$= (a.c) + (\bar{b}.\bar{c})$

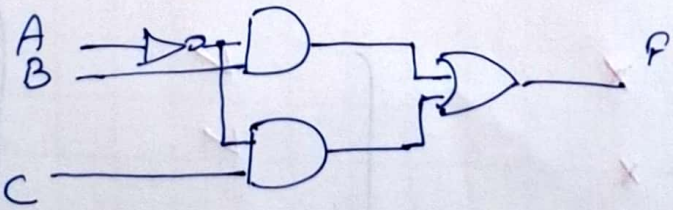


$= (a.b) + [(b+c)(b.c)]$

2/ $F(A,B,C) = \bar{A}B + A\bar{B}C + ABC$

Simplification $A\bar{B}C + ABC = Ac(\bar{B}+B) = (Ac)(1) = Ac$

$F(A,B,C) = \bar{A}B + Ac$



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

Exercise 5: 3 bit counter (Q_2 Q_1 Q_0)

Max value can count

$(111)_2 = (7)_{10}$

Number of different states $= 2^3 = 8$ states

Q_2	Q_1	Q_0
0	0	0
⋮	⋮	⋮
1	1	1

$1\times2^2 + 1\times2^1 + 1\times2^0 = 7$

counter

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

$2^3 = 8$ states

Next value after 111 (counter 111 → 000)

$= (000)$

Exercise 4: Combinational logic: output depends only on current inputs.

Sequential logic: output depends on current inputs and previous state

Component	Combinational	Sequential	Process binary data used
Video game	×	✓	✓
Pin code locking Σ	×	✓	✓
Speed radon sensn	✓	×	✓
Self-checkout Σ	×	✓	✓
classroom attendance	×	✓	✓
Auto traffic light	×	✓	×
Light switch	✓	×	×
Indus filling machine	×	✓	×
Computer Keyboard → (both depending of keyboard	✓	✓	✓