



**i**INFO

**ILIT**  
GRAND OUEST  
NORMANDIE

**R 1.03**

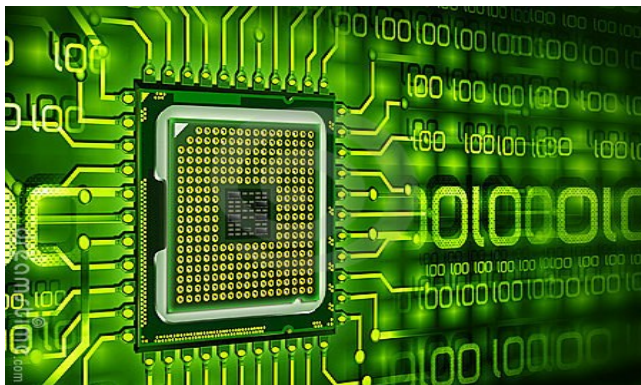
**2025 - 2026**

# **Introduction à l'architecture des ordinateurs**

**Corrigé du TD n° 2**

**Codage de l'information**

**Les décimaux**



**ANNE Jean-François**

***D'après le cours de M. JEANPIERRE***

Le but de ce TD est de se familiariser avec les opérations et les nombres « réels » en binaire.

## A. Entiers relatifs, virgule fixe :

### 1°) Codez sur 8 bits en « signe + valeur absolue » les nombres suivants :

On prend la valeur absolue du nombre en base 10 que l'on code en base 2 sur 8 bits. Et on change le MSB par un 1 si le nombre de départ est négatif.

- 55<sub>10</sub> => 0011 0111<sub>2</sub> => 1011 0111<sub>2s</sub> ;
- 0<sub>10</sub> => 0000 0000<sub>2</sub> => 1000 0000<sub>2s</sub> ;
- + 127<sub>10</sub> => 0111 1111<sub>2</sub> => 0111 1111<sub>2s</sub> ;
- 128<sub>10</sub> => 1000 0000<sub>2</sub> => 1000 0000<sub>2s</sub> . **Attention -128 = +128 !**

### 2°) Codez sur 8 bits en complément à 2 les nombres suivants :

On prend la valeur absolue du nombre en base 10 que l'on code en base 2 sur 8 bits. Et si le nombre de départ est négatif, on inverse les 0 en 1 et inversement, puis on ajoute 1 au nombre obtenu.

- 55<sub>10</sub> => 0011 0111<sub>2</sub> => 1100 1000<sub>2C1</sub> => 1100 1001<sub>2C2</sub> ;
- 0<sub>10</sub> => 0000 0000<sub>2</sub> => 1111 1111<sub>2C1</sub> => 1000 0000<sub>2C2</sub> ;
- + 127<sub>10</sub> => 0111 1111<sub>2</sub> => 0111 1111<sub>2C2</sub> ;
- 128<sub>10</sub> => 1000 0000<sub>2</sub> => 0111 1111<sub>2C1</sub> => 1000 0000<sub>2C2</sub> **Attention -128 = +128 !**

### 3°) Convertissez les nombres suivants en base 2, en utilisant le complément à 2 sur 8,3 bits :

- 2,5<sub>10</sub> => 00000010,100<sub>2</sub> => 11111101,011<sub>2C1</sub> => 11111101,100<sub>2C2</sub> ;
- 35,25<sub>10</sub> => 00100011,010<sub>2</sub> => 11011100,101<sub>2C1</sub> => 11011100,110<sub>2C2</sub> ;
- 2,3<sub>10</sub> => 00000010,010<sub>2</sub> => 11111101,101<sub>2C1</sub> => 11111101,110<sub>2C2</sub> ;

### 4°) Codez sur 4, 8, 10 et 16 bits en complément à 2 les nombres suivants :

Nombre	Sur 4 bits	Sur 8 bits	Sur 10 bits	Sur 16 bits
- 1	0001 => 1110 => 1111	0000 0001 => 1111 1110 => 1111 1111	00 0000 0001 => 11 1111 1110 => 11 1111 1111	0000 0000 0000 0001 => 1111 1111 1111 1110 => 1111 1111 1111 1111
- 2	0010 => 1101 => 1110	0000 0010 => 1111 1101 => 1111 1110	00 0000 0010 => 11 1111 1101 => 11 1111 1110	0000 0000 0000 0010 => 1111 1111 1111 1101 => 1111 1111 1111 1110
- 3	0011 => 1100 => 1101	0000 0011 => 1111 1100 => 1111 1101	00 0000 0011 => 11 1111 1100 => 11 1111 1101	0000 0000 0000 0011 => 1111 1111 1111 1100 => 1111 1111 1111 1101

## B. Arithmétique :

Il s'agit ici de faire les opérations dans la base demandée. **PAS** de convertir le résultat !

1°) Addition :

a) En base 2 :

0111011 + 1110001 <hr/> 10101100	11101110 + 01110101 <hr/> 10110111 + 1000011010	11001100 + 10011001 + 11011101 <hr/> 10011011 + 1011011101	10001010 + 01110111 + 10101010 + 11111111 <hr/> 00010101 + 1010111111
--	--	--	--

b) En base 16 :

2E + 3F <hr/> 6D	3FD + 978 <hr/> D75	1F2D + FABC <hr/> 119E9	FED + CBA + 012 + 345 <hr/> 1FFE
------------------------	---------------------------	-------------------------------	--

2°) Soustraction :

a) En base 2 :

1110011 - 0111011 <hr/> 111000	11101110 - 01000101 <hr/> 10100101 - 100 <hr/> 100	11111111 - 00011001 - 00111001 <hr/> 01011001 - 1010100 <hr/> 1010100	11111111 - 01000010 - 10001000 - 00100100 <hr/> 00010001 - 00000000 <hr/> 00000000
--------------------------------------	--	--	--

b) En base 16 :

BA - AB <hr/> 0F	F7A - DC5 <hr/> 1B5	94AB - 27FC <hr/> 6CAF	FE3ABC - 5FAD2F <hr/> 9E8D8D
------------------------	---------------------------	------------------------------	------------------------------------

C. Virgule flottante

$$N = (-1)^S \cdot 1, M \cdot 2^E$$

1°) Codez sur 32 bits en norme IEEE754 (float) :

$5_{10} \Rightarrow$  **01000000 10100000 00000000 00000000** <sub>IEEE754</sub> ;

$5_{10} \Rightarrow 101_2 \Rightarrow + 1,01 \times 2^{+2} \Rightarrow S=+ ; M = 01 ; E = 2+127 = 129 = 1000\ 0001_2 ;$

$S \Rightarrow + \Rightarrow 0$

$E \Rightarrow 129 \Rightarrow 1000\ 0001$

$M \Rightarrow 010\ 0000\ 0000\ 0000\ 0000\ 0000$

$5_{10} \Rightarrow 0\ 1000\ 0001\ 010\ 0000\ 0000\ 0000\ 0000\ 0000$  IEEE754

$2,5_{10} \Rightarrow$  **01000000 00100000 00000000 00000000** IEEE754;

$2,5_{10} \Rightarrow 10,1_2 \Rightarrow + 1,01 \times 2^1 \Rightarrow S=+ ; M = 01 ; E = 1+127 = 128 = 1000\ 0000_2 ; ;$

$S \Rightarrow + \Rightarrow 0$

$E \Rightarrow 128 \Rightarrow 1000\ 0000$

$M \Rightarrow 010\ 0000\ 0000\ 0000\ 0000\ 0000$

$2,5_{10} \Rightarrow 0\ 1000\ 0000\ 010\ 0000\ 0000\ 0000\ 0000$  IEEE754

$-2,25_{10} \Rightarrow$  **11000000 00100000 00000000 00000000** IEEE754;

$-2,25_{10} \Rightarrow 10,01_2 \Rightarrow - 1,001 \times 2^1 \Rightarrow S=- ; M = 001 ; E = 1+127 = 128 = 1000\ 0000_2 ; ;$

$S \Rightarrow - \Rightarrow 1$

$E \Rightarrow 128 \Rightarrow 1000\ 0000$

$M \Rightarrow 001\ 0000\ 0000\ 0000\ 0000\ 0000$

$-2,25_{10} \Rightarrow 1\ 1000\ 0000\ 001\ 0000\ 0000\ 0000\ 0000$  IEEE754

**2°) Décodez les nombres suivants codés en 32 bits en norme IEEE754 (float) :**

**+ 12.75** <sub>10</sub>  $\Rightarrow$  **01000001 01001100 00000000 00000000** IEEE754;

0 1000 0010 1001100 00000000 00000000 IEEE754

$S \Rightarrow 0 \Rightarrow +$

$E \Rightarrow 10000010 \Rightarrow 130 \Rightarrow 130-127 = 3$

$M \Rightarrow 100110000000000000000000$

$S = + ;$

$M = 100110000000000000000000 ;$

$E = 3 ;$

$(-1)^0 \cdot 1,10011 \cdot 2^3 \Rightarrow 1100,11_2 \Rightarrow 12,75_{10}$

**-4096** <sub>10</sub>  $\Rightarrow$  **11000101 10000000 00000000 00000000** IEEE754;

1 10001011 000000000000000000000000 IEEE754

$S \Rightarrow 1 \Rightarrow -$

$E \Rightarrow 10001011 \Rightarrow 139 \Rightarrow 139-127 = 12$

M => 000000000000000000000000

S = - ;

M = 000000000000000000000000 ;

E = 12 ;

$(-1)^1 \cdot 1,0 \cdot 2^{+12} \Rightarrow 100000000000_2 \Rightarrow -4096_{10}$

**+ 8192.25** <sub>10</sub> => **01000110 00000000 00000001 00000000** IEEE754;

0 10001100 000000000000000100000000 IEEE754

S => 0 => +

E => 10001100 => 140 => 140-127 = 13

M => 000000000000000100000000

S = + ;

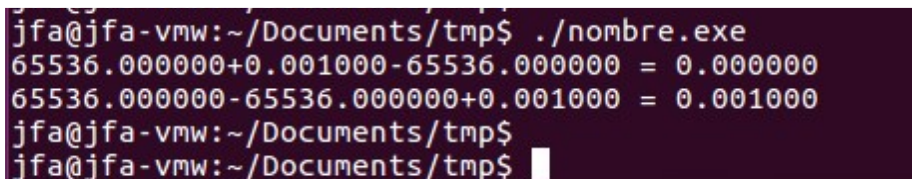
M = 000000000000000100000000 ;

E = 13 ;

$(-1)^0 \cdot 1,0000000000000001 \cdot 2^{+13} \Rightarrow 1,0000000000000001_2 \Rightarrow +8192,25_{10}$

**3°) Quel est le résultat du programme C suivant ?**

```
int main() {
    float x = 65536;
    float y = 0.001;
    float z = x+y-x;
    float t = x-x+y;
    printf("%f+%f-%f = %f\n", x, y, x, z);
    printf("%f-%f+%f = %f\n", x, x, y, t);
}
```



**D. Pour aller plus loin :**

**1°) Multiplication en base 2 :**

$$\begin{array}{r} 01110 \\ * 11100 \\ \hline 110001000 \end{array}$$

$$\begin{array}{r} 111111,10 \\ * 100100,01 \\ \hline 100011111101,1110 \end{array}$$

$$\begin{array}{r} 11001101 \\ * 10000000 \\ \hline 110011010000000 \end{array}$$

**2°) Multiplication en base 16 :**

Il faut la table de multiplication par 16 !

$$\begin{array}{r} 32 \\ * 56 \\ \hline 10CC \end{array}$$

$$\begin{array}{r} ABC \\ * DEF \\ \hline 959184 \end{array}$$

$$\begin{array}{r} 1234 \\ * FEDC \\ \hline 121F3CB0 \end{array}$$

$$\begin{array}{r} 1F3D \\ * F4BE \\ \hline 1E98C146 \end{array}$$

**3°) Division en base 2 :**

$$\begin{array}{r|l} 100011 & 111 \\ -111 & 101 \\ \hline 000111 & \\ -111 & \\ \hline 0 & \end{array}$$

$$\begin{array}{r|l} 1110101 & 1101 \\ -1101 & 1001 \\ \hline 0001101 & \\ -1101 & \\ \hline 0 & \end{array}$$

$$\begin{array}{r|l} 100011,10 & 101,01 \\ -10101 & 110,11 \\ \hline 0011101 & \\ -10101 & \\ \hline 0100000 & \\ -10101 & \\ \hline 010110 & \\ -10101 & \\ \hline 01 & \end{array}$$

**4°) Additions en base 16 :**

Effectuez l'addition en base 16 des nombres contenus dans les cases en dessous de la case concernée.

				82				
			38	4A				
		18	20	2A				
	A	E	12	18				
4	6	8	A	E				

Exemple : A = 4 + 6

**5°) Opérations en Base 2 :**

Effectuer les calculs suivants sachant que, en binaire :  
A = 111100, B = 1110, C = 101, D = 11111.

- A+B+D = 111100 + 1110 + 11111 = 110 1001
- A-B = 111100 - 1110 = 10 1110
- B×C = 1110 \* 101 = 100 0110
- A÷C = 111100 / 101 = 1100

## Table de multiplication hexadécimale

UND	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	0	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	0	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	0	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	0	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	0	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	0	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	0	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	0	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	0	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	0	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	0	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	0	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	0	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	0	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1