

# Numération de base

---

# Numération de base

---

## *Plan de la phase*

*Introduction*  
*Language binaire*  
*Langage hexadécimal*  
*Exercices*

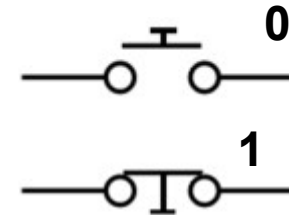
# Numération de base

## Introduction

Pour **passer** d'un **langage A** à un **langage B**, il faut un **systeme** de **codification** (ou de codage).

*Il existe plusieurs possibilités de codage de l'information :*

- BINAIRE
- HEXADECIMAL
- BCD
- ASCII
- ...



Les **systemes** informatiques actuels ne **fonctionnent** actuellement que selon **une logique à deux états**, le courant passe ou ne passe pas. Ces deux états binaires notés **0** et **1** déterminent cette logique binaire.

Toute **information** à traiter devra être **représentée** sous une **forme binaire**, que ce soit en interne ou sur les « fils » reliant les composants de l'ordinateur.

# Numération de base

## *LE LANGAGE BINAIRE*

Alphabet : Symboles **0, 1**

Ces symboles combinés, doivent permettre de définir toute information à traiter,  
La base de numérotation est 2 (car on utilise 2 symboles), et les calculs se font en base 2.

Un nombre base 2 est noté  $n_2$

**$110_2$**

En base 10 un nombre devrait être noté  $n_{10}$

**$110_2 \neq 110_{10}$**

# Numération de base

## LE LANGAGE BINAIRE - Conversion

On **pass**e d'un nombre en **base 10** à un nombre en **base 2** par divisions successives.

Soit  $135_{10}$  à convertir en base 2

135 / 2	=	67	reste	1
67 / 2	=	33	reste	1
33 / 2	=	16	reste	1
16 / 2	=	8	reste	0
8 / 2	=	4	reste	0
4 / 2	=	2	reste	0
2 / 2	=	1	reste	0
1 / 2	=	0	reste	1



**Sens de lecture**

$$135_{10} = (10000111)_2$$

Chaque élément binaire **0** ou **1** est appelé un digit binaire ou **bit**

Une suite de **quatre bits** : un **quartet**

Une suite de **huit bits** : un **octet**



En anglais **octet** se traduit par **byte** et non par **bit**

# Numération de base

## LE LANGAGE BINAIRE - Conversion

On **pass**e d'un nombre en **base 2** à un nombre en **base 10** par **multiplications successives**.

Soit  $10011_2$  à convertir en décimal

1	0	0	1	1
x	x	x	x	x
$2^4$	$2^3$	$2^2$	$2^1$	$2^0$
(16)	(8)	(4)	(2)	(1)
=	=	=	=	=
16	0	0	2	1

dont la somme donne  $19_{10}$

On peut en déduire une autre manière de convertir un nombre décimal en binaire :

$2^7$	$2^6$	$2^5$	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	
128	64	32	16	8	4	2	1	
135	7	7	7	7	7	3	1	
1	0	0	0	0	1	1	1	

MSB → ← LSB

Poids binaire

# Numération de base

## *LE LANGAGE BINAIRE - Conversion*

Base 10	Base 2	Base 10	Base 2
1	1	6	110
2	10	7	111
3	11	8	1000
4	100	9	1001
5	101	10	1010

*Les dix premiers nombres binaires*

Le poids binaire : puissance à laquelle est élevé le bit lors de la conversion binaire-décimal :

$$135_{10} = (10000111)_2 = 1 \times 2^7 + 0 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

**MSB** : Le bit de poids fort (*Most Significant Bit*), c'est le bit situés le plus à gauche.

**LSB** : Le bit de poids faible (*Less Significant Bit*), c'est le bit situés le plus à droite.

# Numération de base

## LE LANGAGE BINAIRE – Opération binaires

### Addition

a	b	c	S	R
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

```

  1 1 1 1 1 (carried digits)
    0 1 1 0 1
+   1 0 1 1 1
-----
= 1 0 0 1 0 0

```

### Soustraction

a	b	c	S	R
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

```

      *   * * * (starred columns are borrowed from)
    1 1 0 1 1 1 0
-   1 0 1 1 1
-----
= 1 0 1 0 1 1 1

```



# Numération de base

## LE LANGAGE BINAIRE – Opération binaires

### Multiplication

La multiplication binaire s'effectue selon le principe de la multiplication décimale.

$$\begin{array}{r}
 \begin{array}{r}
 1011 \quad (A) \\
 \times 1010 \quad (B) \\
 \hline
 0000 \quad \leftarrow \text{Corresponds to a zero in B} \\
 + 1011 \quad \leftarrow \text{Corresponds to a one in B} \\
 + 0000 \\
 + 1011 \\
 \hline
 = 1101110
 \end{array}
 \end{array}$$

### Division

La division est basé sur une succession de soustraction et s'emploie de la même façon qu'une division décimale.

$$\begin{aligned}
 (11011)_2 &= (101)_2 \times (101)_2 + (10)_2 \\
 (27)_{10} &= (5)_{10} \times (5)_{10} + (2)_{10}
 \end{aligned}$$

$$\begin{array}{r}
 11011 \quad | \quad 101 \\
 - 101 \quad \leftarrow \\
 \hline
 0011 \\
 - 000 \quad \leftarrow \\
 \hline
 111 \\
 - 101 \quad \leftarrow \\
 \hline
 10
 \end{array}$$

# Numération de base

## *LE LANGAGE BINAIRE – Nombres fractionnaires*

La partie décimale d'un nombre se traduit en binaire en mettant en œuvre des puissances négatives de 2.

### Conversion de binaire / décimal

$$\begin{array}{rcl}
 \text{Ainsi} & 1\ 0\ 0 & .\ 0\ 1_2 \text{ sera équivalent à} \\
 (1 \times 2^2) + (0 \times 2^1) + (0 \times 2^0) & . & (0 \times 2^{-1}) + (1 \times 2^{-2}) \\
 (1 \times 4) + (0 \times 2) + (0 \times 1) & . & (0 \times 1/2) + (1 \times 1/4) \text{ soit } 4.25_{10}
 \end{array}$$

### Conversion de décimal / binaire

$$\begin{array}{rcl}
 \text{Soit } 0.625_{10} & & \\
 0.625 \times 2 = 1.250 & \text{Poids binaire} & 1 \times 2^{-1} \\
 0.250 \times 2 = 0.500 & \text{Poids binaire} & 0 \times 2^{-2} \\
 0.500 \times 2 = 1.000 & \text{Poids binaire} & 1 \times 2^{-3}
 \end{array}$$

On arrête le processus quand il n'y a plus de partie fractionnaire ou que la précision obtenue est jugée suffisante.

$$\text{Ainsi } (0.625)_{10} = (0.101)_2$$

# Numération de base

## LE LANGAGE HEXADECIMAL – Conversion

Le système hexadécimal permet un compromis entre l'utilisation d'un code binaire et une facilité de lecture des résultats.

On utilise un **alphabet** de **16** symboles.

Symboles hexadécimal : **0 1 2 3 4 5 6 7 8 9 A B C D E F**

Équivalent décimal : **0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15**

### Conversion décimal / hexadécimal

Soit à convertir en base 16 le nombre  $(728)_{10}$

$(728)_{10}$	:	16	=	45	Reste	8	 Sens de lecture
$(45)_{10}$	:	16	=	2	Reste	13	
$(2)_{10}$	:	16	=	0	Reste	2	

Les nombres supérieurs à 9 n'existent pas dans la notation hexadécimal, le « 13 » doit être remplacé par son équivalent hexadécimal « D »

Donc  $(728)_{10} = (2D8)_{16}$ , on notera les nombres en base 16 par **0x2D8** ou par **2D8H**

# Numération de base

## *LE LANGAGE HEXADECIMAL – Conversion*

### Conversion hexadécimal / décimal

Soit **0x13D** à convertir en base 10

$$\begin{array}{ccccccc} & & 0 & x & 1 & 3 & D \\ & & & & & & \downarrow \\ & 1 & x & 16^2 & + & 3 & x & 16^1 & + & 13 & x & 16^0 \\ 1 & x & 256 & + & 3 & x & 16 & + & 13 & x & 1 & = & 317 \end{array}$$

# Numération de base

## LE LANGAGE HEXADÉCIMAL – Conversion

### Conversion binaire / hexadécimal

La représentation des 16 symboles de l'alphabet hexadécimal en binaire, utilise au maximum 4 bits.

Décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binaire	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100	1101	1110	1111
Hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

On peut donc **directement** et facilement **passer** du **binaire** en **hexadécimal** en **décomposant** le nombre binaire en **blocs de 4 bits** – en partant de la droite (bits dits de poids faible) et en restituant sa valeur hexadécimal à chacun de ces blocs.

$$(732)_{10} = (\textcolor{blue}{10}\textcolor{red}{110}\textcolor{green}{1100})_2 = (\textcolor{blue}{0010} \textcolor{red}{1101} \textcolor{green}{1100})_2 = (\textcolor{blue}{2}\textcolor{red}{D}\textcolor{green}{C})_{16}$$

### Conversion hexadécimal / binaire

Pour cela, on convertit les chiffres qui composent le nombre hexadécimal en leur équivalent binaire.

$$\text{Ainsi : } (1AB)_{16} \quad \begin{array}{ccc} 1 & A & B \\ \textcolor{blue}{0001} & \textcolor{red}{1010} & \textcolor{green}{1011} \end{array} \quad (1AB)_{16} = (\textcolor{blue}{0001}\textcolor{red}{1010}\textcolor{green}{1011})_2$$

# Numération de base

## LE LANGAGE HEXADECIMAL – Addition

L'addition s'effectue à partir de la technique de l'addition et de la table d'addition suivante :

+	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

$$\begin{array}{r}
 \phantom{1}D5 \\
 + AC \\
 \hline
 181
 \end{array}$$

# Numération de base

## LE LANGAGE HEXADECIMAL – Multiplication

La multiplication s'effectue à partir de la technique de la multiplication par glissement par jalousies et en utilisant la table de multiplication suivante :

x	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

$$\begin{array}{r}
 25 \\
 \times A1 \\
 \hline
 325 \\
 + 142. \\
 \hline
 1745
 \end{array}$$

# Numération de base

## Exercices

**2.1** Convertir en binaire les nombres  $397_{10}$ ,  $133_{10}$  et  $110_{10}$  puis en décimal les nombres  $101_2$ ,  $0101_2$  et  $1101110_2$  et vérifier en convertissant pour revenir à la base d'origine.

**2.2** Effectuer les opérations suivantes et vérifier les résultats en procédant aux conversions nécessaires.

a)  $1100 + 1000 =$

b)  $1001 + 1011 =$

c)  $1100 - 1000 =$

d)  $1000 - 101 =$

e)  $1 + 1 + 1 + 1 =$

**2.3** Réaliser les opérations suivantes et vérifier les résultats en procédant aux conversions nécessaires.

a)  $1011 \times 11 =$

b)  $1100 \times 101 =$

c)  $100111 \times 0110 =$

**2.4** Réaliser les opérations suivantes et vérifier les résultats en procédant aux conversions nécessaires.

a)  $100100 / 11 =$

b)  $110000 / 110 =$

**2.5** Convertir en binaire  $127.75_{10}$  puis  $307.18_{10}$ . Vous pourrez constater, à la réalisation de ce dernier exercice, que la conversion du  $.18$  peut vous entraîner « assez loin ». C'est tout le problème de ce type de conversion et la longueur accordée à la partie fractionnaire dépendra de la précision souhaitée.



# Numération de base

## Exercices

- 
- 2.6** Convertir en hexadécimal    a)  $3167_{10}$     b)  $219_{10}$     c)  $6560_{10}$
- 
- 2.7** Convertir en décimal    a)  $0x3AE$     b)  $0xFFFF$     c)  $0x6AF$
- 
- 2.8** Convertir en base 16    a)  $128_{10}$     b)  $101_{10}$     c)  $256_{10}$   
d)  $110_2$     e)  $1001011_2$
- 
- 2.9** Convertir en base 10    a)  $0xC20$     b)  $0xA2E$
- 
- 2.10** Convertir en base 2    a)  $0xF0A$     b)  $0xC01$