Systèmes de nombres

Représentation des informations

- Donnée de base manipulée par la machine physique est le bit (Binary Digit) qui ne peut prendre que deux valeurs: 0 et 1
- Au niveau physique, toutes les informations (nombres, caractères, instructions, images …) ne peuvent être représentées que par une combinaison de 0 et 1

sous forme d'une chaîne binaire

Base 2

- Codage de l'information se fait dans une base de représentation qui est la base 2
  - **système de numération positionnel**
- Un système de numération positionnel est un système d'écriture des nombres à l'aide de symboles appelés chiffres où la valeur des symboles ne dépend pas seulement de la forme des chiffres mais également de leur position dans le nombre
- Base 10:

10^3	10^2	10^1	10^0
1	**3**	**4**	**5**

Base X

- **Dans un système en base X, il faut X symboles différents pour représenter les chiffres de 0 à X-1**

Base 2:	0, 1
Base 5:	0, 1, 2, 3, 4
Base 8:	0, 1, 2, 3, 4, 5, 6, 7
Base 10:	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Base 16:	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

Systèmes de numération

Système	*Base*	*Symboles*
Décimal	**10**	**0, 1, 2, 3, 4, 5, 6, 7, 8, 9**
Binaire	**2**	**0, 1**
Octal	**8**	**0, 1, 2, 3, 4, 5, 6, 7**
Hexadécimal	**16**	**0,1, …,9, A, B, C, D, E, F**

Quantité/Comptage; Base 10

		9	8	8
+				1
		9	8	9

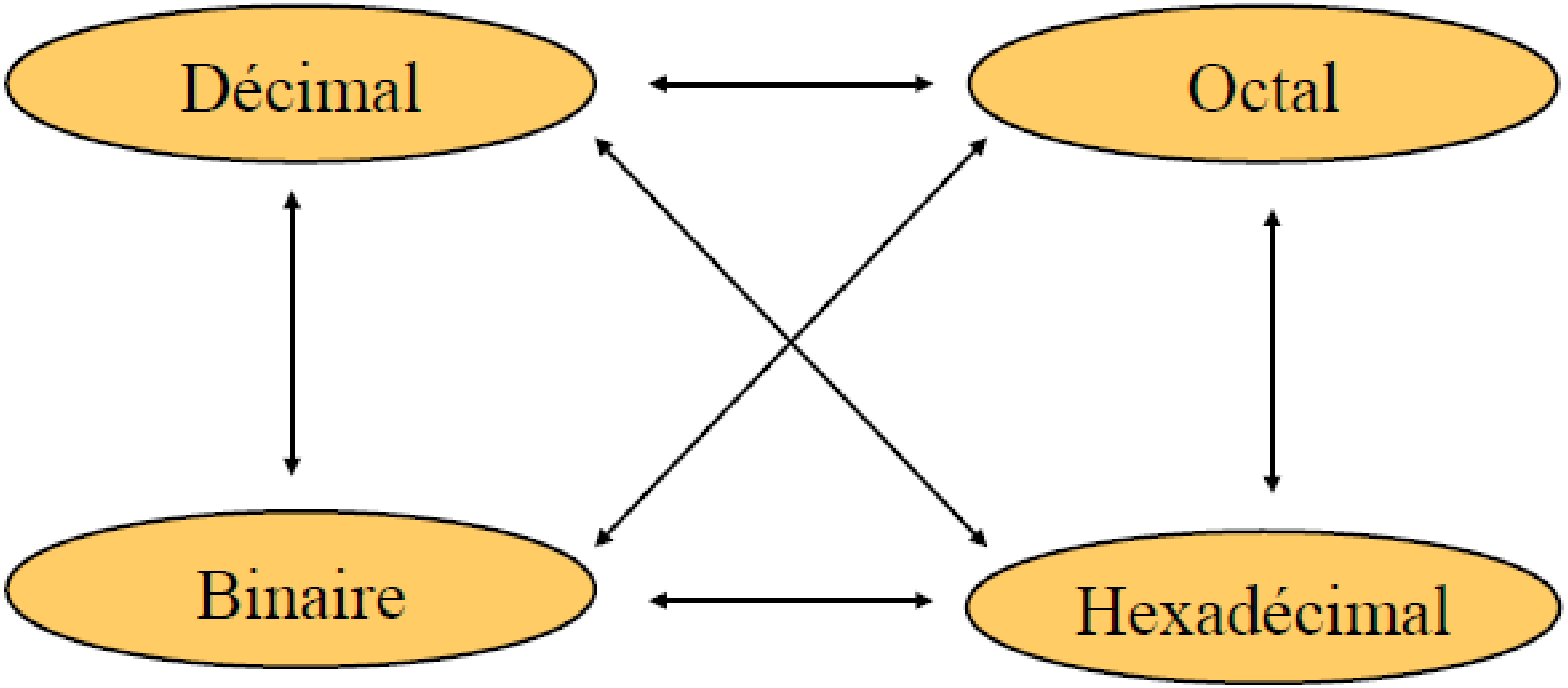
		5	3	1
+			2	8
		5	5	9

		1	1	
	0	9	9	9
+	0	0	0	1
	1	0	0	0

Quantité/Comptage

Décimal	*Binaire*	*Octal*	Hexadécimal
0	0	0	0
1	1	1	1
2	01+1 = $0^10 = 10$	2	2
3	10+1 = 11	3	3
4	11+1 = 1^10 = 0^100 = 100	4	4
5	100+1 = 101	5	5
6	101+1 = 110	6	6
7	110+1 = 111	7	7
8	111+1 = 1000	7+1 = 10	8
9	1000+1 = 1001	11	9
9+1 = $0^10 = 10$	1010	12	A
11	1011	13	B

Conversion d'une base à une autre



Exemple

$$25_{10} = 11001_2 = 31_8 = 19_{16}$$

Base

Rappel, système décimal

- Le nombre 125_{10} signifie:
  1 groupe de 100 (100 = 10^2)
  2 groupes de 10 (10 = 10^1)
  5 groupes de 1 (1 = 10^0)

Placer les valeurs

Système décimal

125

1 groupe de 100 (100 = 10^2)

2 groupes de 10 (10 = 10^1)

5 groupes de 1 (1 = 10^0)

Représentation d'un nombre N en base X

Chiffre de poids fort

Chiffre de poids faible

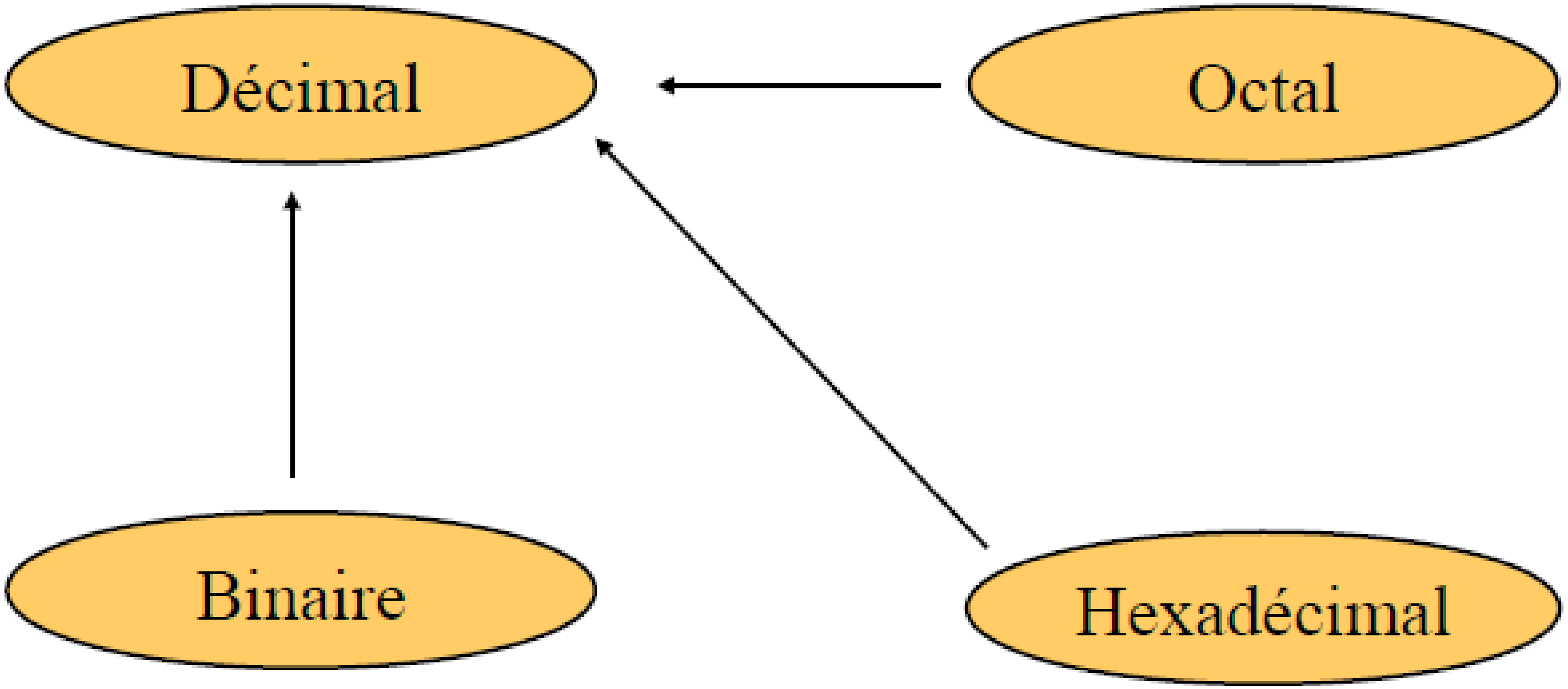
Poids

$$Nombre_x = \sum chiffre_i * X^i$$

$$
\begin{aligned}
125_{10} \Rightarrow \quad 5 \times 10^0 &= 5 \\
2 \times 10^1 &= 20 \\
1 \times 10^2 &= \underline{100} \\
125 &= 1 \times 10^2 + 2 \times 10^1 + 5 \times 10^0
\end{aligned}
$$

Base

Conversion du nombre N exprimé en base X vers la base 10



Conversion du nombre N exprimé en base X vers la base 10 (X $\longrightarrow$ 10)

- Technique $Nombre_x = chiffre_n \ldots chiffre_0$
  - Multiplier chaque chiffre par la base X^n, ou *n* est le “poids” (de 0 à n) de ce chiffre pour un nombre exprimé par **n+1** chiffres
  - Additionner les résultats

$$Nombre_{10} = \sum chiffre_i * X^i \ (i \ de \ 0 \ \text{à} \ n)$$

Exemple, (X $\rightarrow$ 10)

Bit “poids 0”

$$
\begin{array}{lrcl}
101011_2 \Rightarrow & 1 \times 2^0 & = & 1 \\
 & 1 \times 2^1 & = & 2 \\
 & 0 \times 2^2 & = & 0 \\
 & 1 \times 2^3 & = & 8 \\
 & 0 \times 2^4 & = & 0 \\
 & 1 \times 2^5 & = & \underline{32} \\
 & & & 43_{10}
\end{array}
$$

Fractions

- Décimal (rappel)

10^0	10^{-1}	10^{-2}
3.	1	4

3.14 →

$$4 * 10^{-2} = 0.04$$
$$1 * 10^{-1} = 0.1$$
$$3 * 10^{0} = 3$$

3.14

Fractions, (X → 10)

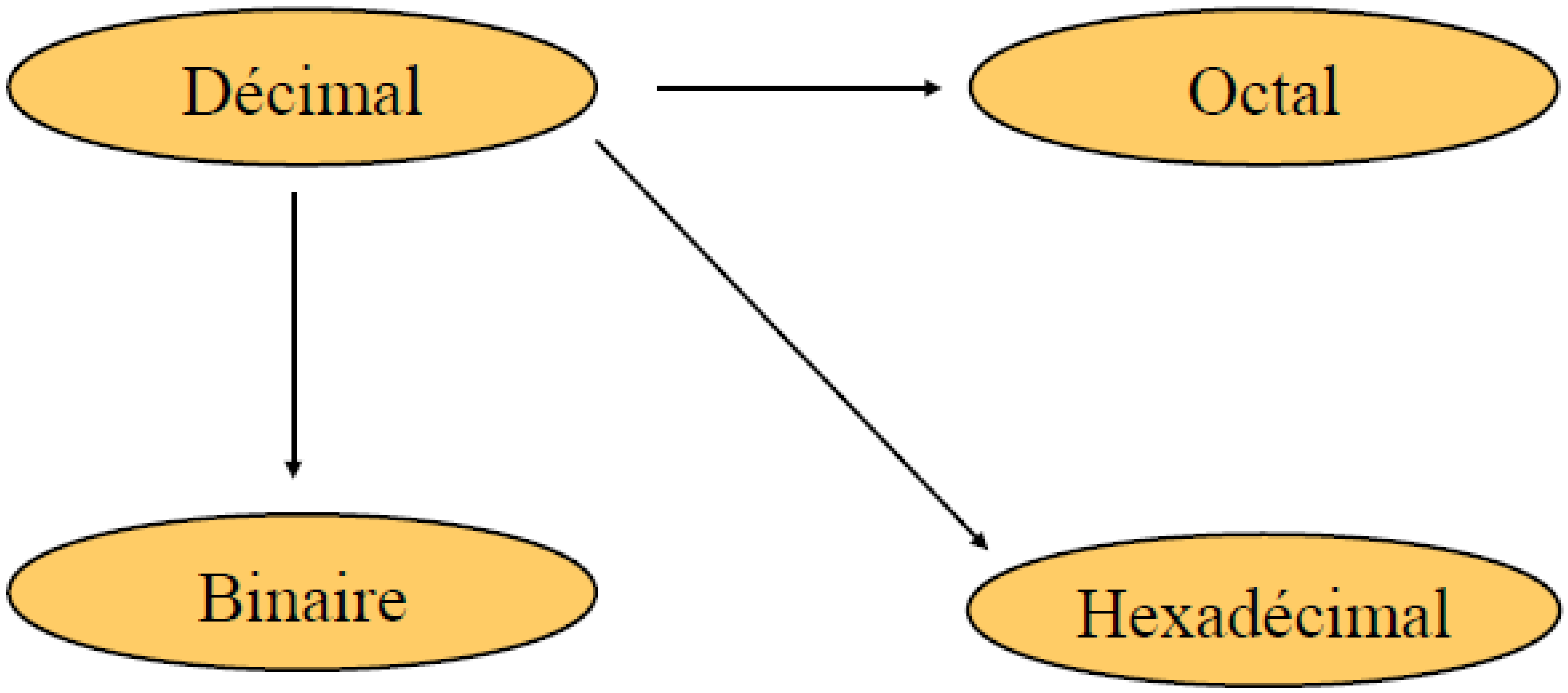
2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
1	0.	1	0	1	1

- Binaire vers décimal

$$\mathbf{10.1011_2}$$

$$\begin{aligned}
1 * 2^1 &= 2 \\
0 * 2^0 &= 0 \\
1 * 2^{-1} &= 0.5 \\
+ \quad 0 * 2^{-2} &= 0 \\
1 * 2^{-3} &= 0.125 \\
1 * 2^{-4} &= 0.0625 \\
\hline
& \mathbf{2.6875}
\end{aligned}$$

Conversion du nombre N exprimé en base 10 vers une base X (10 → X)



Conversion du nombre N exprimé en base 10 vers une base X (10 $\longrightarrow$ X)

- Conversion d'un nombre entier
  - Méthode des divisions successives
  - Méthode des soustractions successives

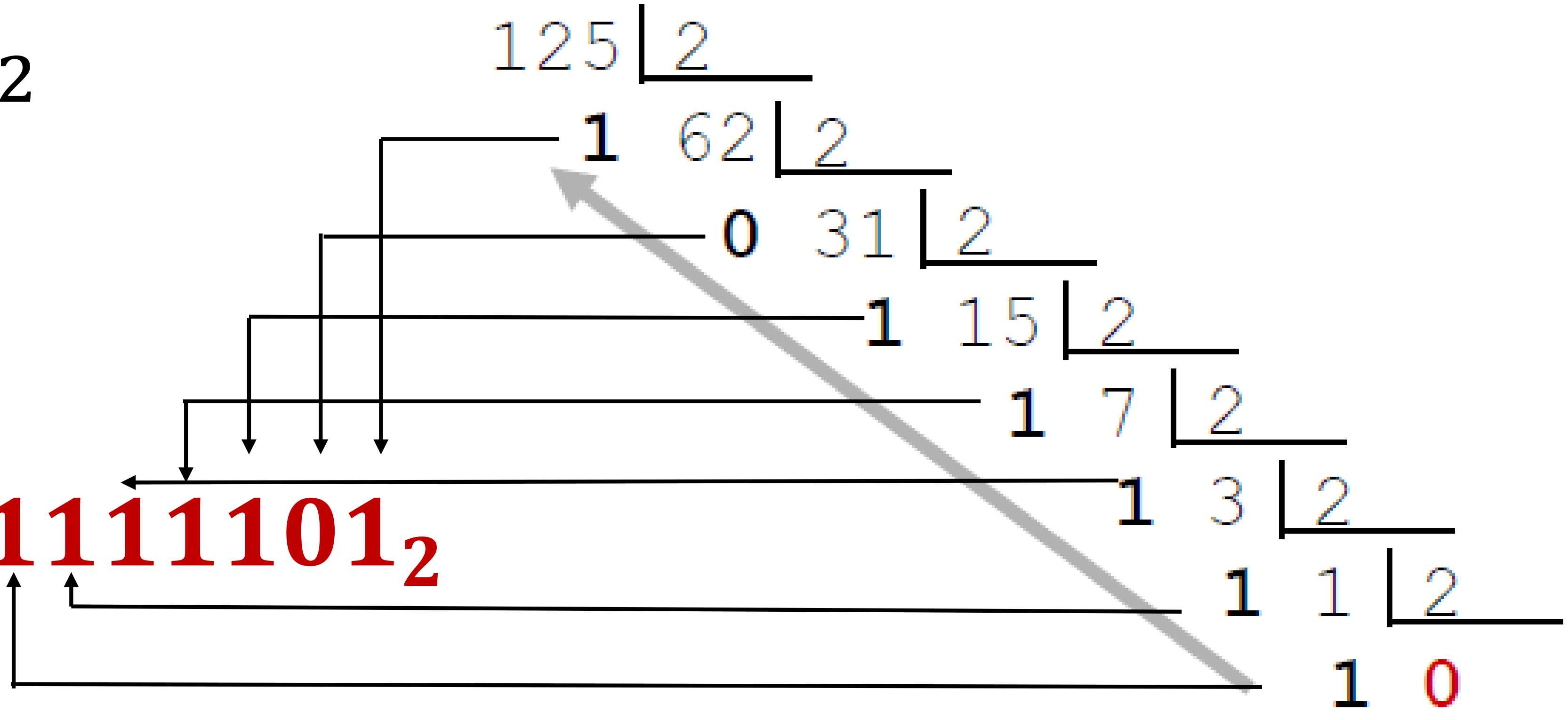
Conversion du nombre N exprimé en base 10 vers une base X (10 $\rightarrow$ X)

- Conversion d'un nombre entier
  - Méthode des divisions successives
    - N est itérativement divisé par X jusqu'à obtenir un quotient égal à 0
    - La conversion du nombre N dans la base X est obtenue en notant les restes de chacune des divisions effectuées depuis la dernière division jusqu'à la première

Conversion du nombre N exprimé en base 10 vers une base X (10 → X)

- Conversion d'un nombre entier
  - Méthode des divisions successives

$125_{10} = ?_2$



$125_{10} = \mathbf{1111101_2}$

Conversion du nombre N exprimé en base 10 vers une base X (10 → X)

- Conversion d'un nombre entier
  - Méthode des soustractions successives
    - La plus grande puissance de X qui est inferieure ou égale à N est soustraite à N.
    - Répéter jusqu'à obtenir un résultat égale à 0
    - Le nombre N exprimé en base X est obtenu en notant le nombre de fois où une même puissance de X a été retirée et ce pour chaque puissance depuis la plus grande apparaissant dans l'ordre décroissant des puissances.

Conversion du nombre N exprimé en base 10 vers une base X ($10 \longrightarrow X$)

- Conversion d'un nombre entier
  - Méthode des soustractions successives

$235_{10} = ?_8$ $\mathbf{8^0 = 1;\ 8^1 = 8;\ 8^2 = 64;\ 8^3 = 512;}$

235 – 64 = 171; 171 – 64 = 107; 107 – 64 = 43; => **3 x 64**

43 – 8 = 35; 35 – 8 = 27; 27 – 8 = 19; 19 - 8 = 11; 11 - 8 = 3

$\Rightarrow$**5 x 8**

3 – 1 = 2; 2 – 1 = 1; 1 – 1 = **0**; => **3 x 1**

$$235_{10} = 3 * 64 + 5 * 8 + 3 * 1 = 353_8$$

Conversion du nombre N exprimé en base 10 vers une base X (10 $\longrightarrow$ X)

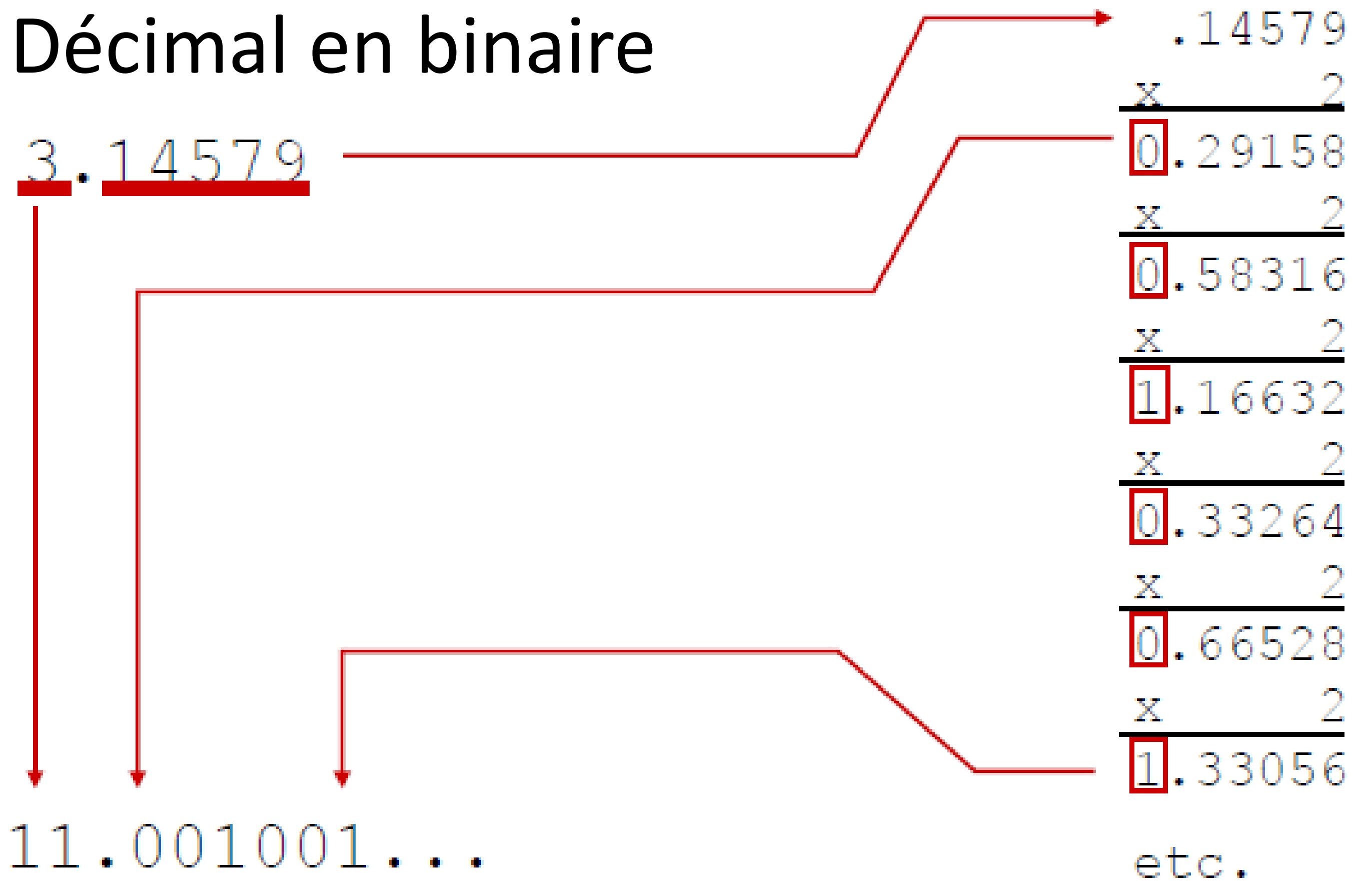
- Conversion d'un nombre **fractionnaire N**
  - Sa partie entière vers une base X
    - Méthode des division successives
    - Méthode des soustractions
  - Partie fractionnaire
    - Multiplier cette partie fractionnaire par la base X
    - La multiplication est itérée sur la partie fractionnaire du résultat obtenu
    - Prendre des parties entières de chacun des résultats des multiplications effectuées

Conversion du nombre N exprimé en base 10 vers une base X (10 ⟶ X)

- Conversion d’un nombre **fractionnaire N**

Le développement s’arrête lorsque la précision voulue est obtenue

Décimal en binaire



Conversion du nombre N exprimé en base 10 vers une base X ($10 \longrightarrow X$)

- Exemple: Convertir 0.45_{10} en base 2:

0.45*2 = 0.9 = 0+0.9

0.9*2 = 1.8 = 1+0.8

0.8*2 = 1.6 = 1+0.6

0.6*2 = 1.2 = 1+0.2

0.2*2 = 0.4 = 0+0.4

0.4*2 = 0.8 = 0+0.8

0.8*2 = 1.6

$$0.45_{10} = 0.01\overline{1100}$$

Conversion du nombre N exprimé en base 10 vers une base X (10 $\longrightarrow$ X)

- Exemple: Convertir $\frac{2}{3}_{10}$ en base 2:

$$\frac{2}{3} \times 2 = \frac{4}{3} = 1 + \frac{1}{3}$$

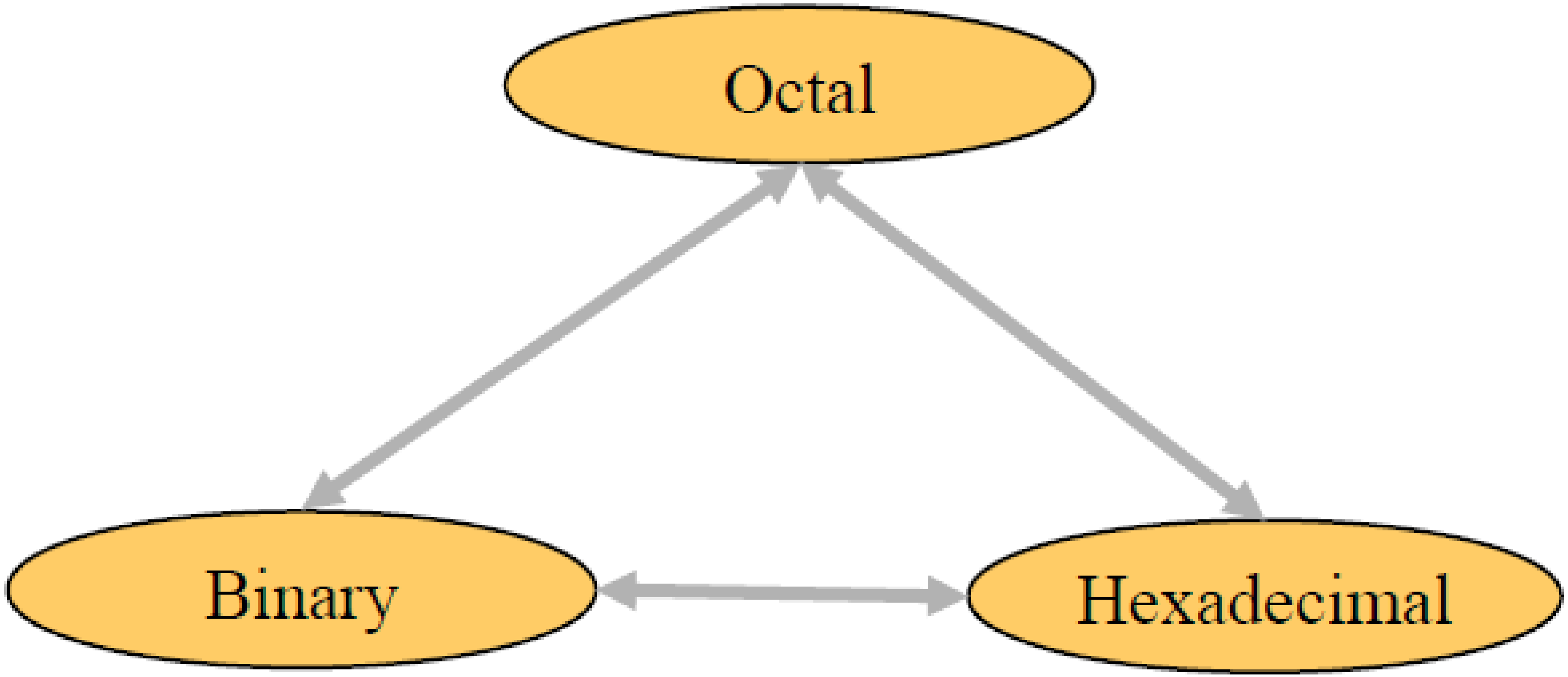
$$\frac{1}{3} \times 2 = \frac{2}{3} = 0 + \frac{2}{3}$$

$$\frac{2}{3}_{10} = 0.\overline{10}$$

Conversion du nombre N exprimé dans la base 8, 16 vers la base 2 et vice versa

- Toutes les informations sont représentées dans un ordinateur sous forme d'une chaîne binaire
- Base de représentation – base 2
- Chaînes binaires ne sont pas aisément manipulables par l'esprit humain
- Deux autres bases sont très souvent utilisées
- La base 8 (système octal) $2^3 = 8$
- La base 16 (système hexadécimal) $2^4 = 16$

Conversion du nombre N exprimé dans la base 8, 16 vers la base 2 et vice versa



Conversion du nombre N exprimé dans la base 8 vers la base 2 et vice versa

- Octal vers binaire, 8 $\longrightarrow$ 2
- Convertir un nombre N exprimé en base 8 vers la base 2 s'effectue en remplaçant chacun des chiffres du nombre N par leur équivalent binaire sur 3 bits car $2^3 = 8$

$$\mathbf{705.14_8} = ?_2$$

$$\begin{array}{ccccccc} \mathbf{7} & \mathbf{0} & \mathbf{5} & . & \mathbf{1} & \mathbf{4} \\ \downarrow & \downarrow & \downarrow & & \downarrow & \downarrow \\ 111 & 000 & 101 & . & 001 & 100 \end{array}$$

Conversion du nombre N exprimé dans la base 8 vers la base 2 et vice versa

- Binaire vers octal, $2 \longrightarrow 8$
- Convertir un nombre N exprimé en base 2 vers la base 8 s'effectue en découpant la chaîne binaire N en paquet de 3 bits depuis le bit de poids faible jusqu'au bit de poids fort pour la partie entière et depuis le bit de poids fort jusqu'au bit de poids faible pour la partie fractionnaire

Digit de poids fort

Digit de poids faible

$$\mathbf{1011010111.11_2} = ?_8$$

Digit de poids faible

$$\underbrace{001}_{1}\,\underbrace{011}_{3}\,\underbrace{010}_{2}\,\underbrace{111}_{7}\,.\,\underbrace{110}_{6}{}_2$$

1 3 2 7 . 6

Conversion du nombre N exprimé dans la base 16 vers la base 2 et vice versa

- Hexadécimal vers binaire, 16 $\longrightarrow$ 2
- Convertir un nombre N exprimé en base 16 vers la base 2 s'effectue en remplaçant chacun des chiffres du nombre N par leur équivalent binaire sur 4 bits car $2^4 = 16$

$$\boldsymbol{7A5.1E_{16} =?_2}$$

$$\begin{array}{ccccc} \mathbf{7} & \boldsymbol{A} & \mathbf{5.} & \mathbf{1} & \boldsymbol{E_{16}} \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0111 & 1010 & 0101\,. & 0001 & 1110 \end{array}$$

$$\boldsymbol{7A5.1E_{16} = 11110100101.0001111_2}$$

Conversion du nombre N exprimé dans la base 16 vers la base 2 et vice versa

- Binaire vers hexadécimal, 2 → 16
- Convertir un nombre N exprimé en base 2 vers la base 16 s'effectue en découpant la chaîne binaire N en paquet de 4 bits depuis le bit de poids faible jusqu'au bit de poids fort pour la partie entière et depuis le bit de poids fort jusqu'au bit de poids faible pour la partie fractionnaire

Digit de poids fort

Digit de poids faible

$$1011010111.11_2 = ?_{16}$$

Digit de poids faible

$$\underbrace{0010}_{2}\,\underbrace{1101}_{D}\,\underbrace{0111}_{7}\,.\,\underbrace{1100}_{C}{}_2$$

2 D 7 . C

Conversion du nombre N exprimé dans la base 8 vers la base 16 et vice versa

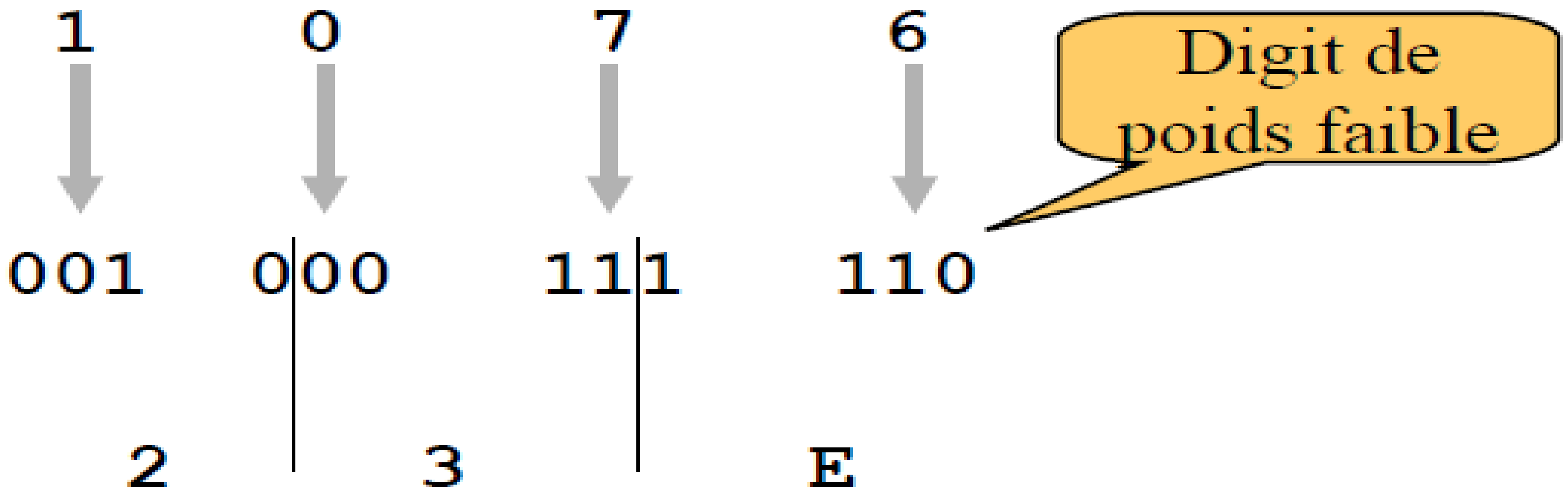
- Technique
- Utiliser système binaire comme un système intermédiaire

$$\text{Base } 8 \longrightarrow \text{Base } 2 \longrightarrow \text{Base } 16$$

$$\text{Base } 16 \longrightarrow \text{Base } 2 \longrightarrow \text{Base } 8$$

Conversion du nombre N exprimé dans la base 8 vers la base 16 et vice versa

$1076_8 = ?_{16}$



$1076_8 = 23E_{16}$

Conversion du nombre N exprimé dans la base 8 vers la base 16 et vice versa

$1F0C_{16} = ?_8$

1	F	0	C
↓	↓	↓	↓
0001	1111	0000	1100

0 | 001 | 111 | 1 0 | 000 | 1 | 100

1 | 7 | 4 | 1 | 4

$1F0C_{16} = 17414_8$

Mesure de la quantité d'information

Base 10

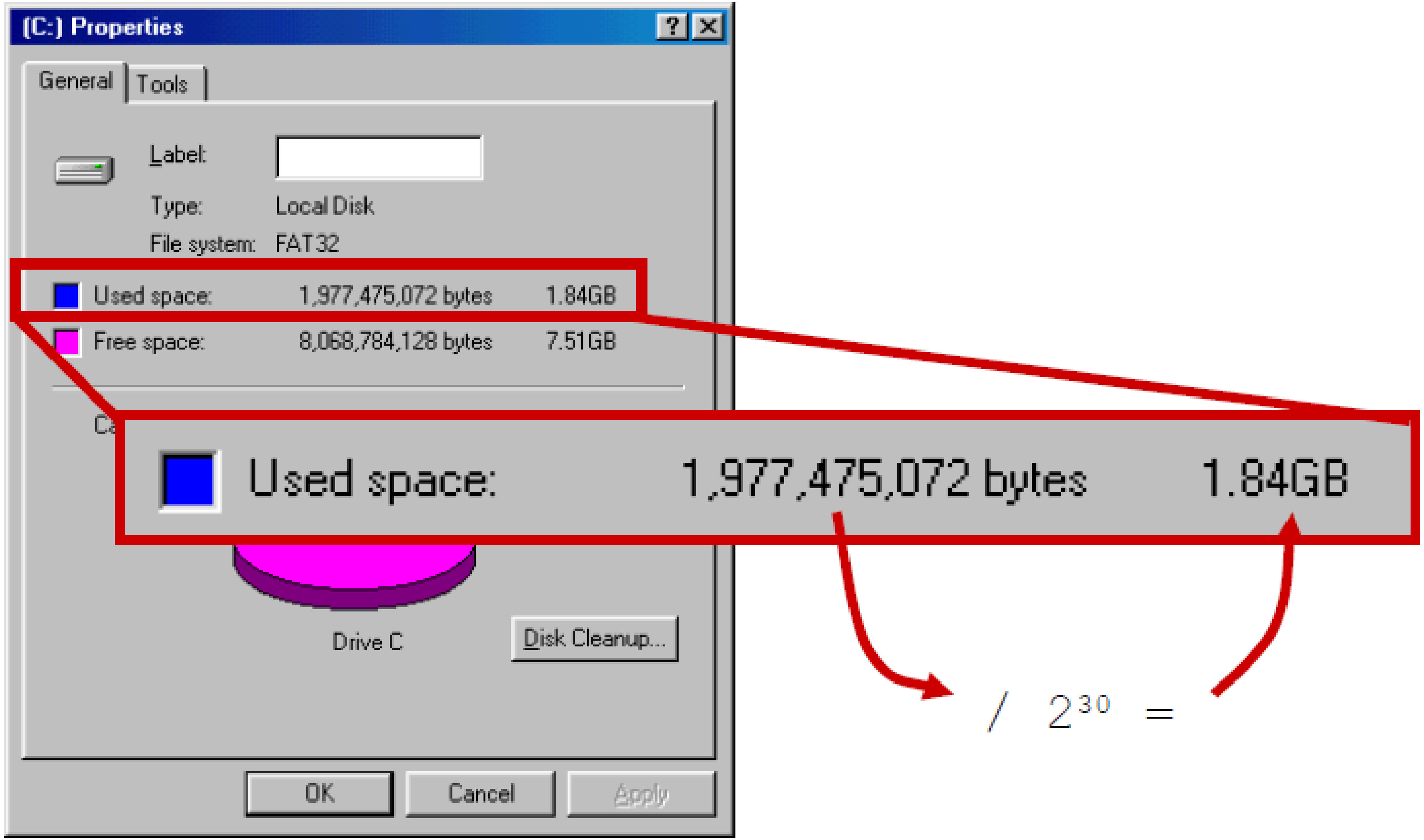
Puissance	Nom	Symbole	Valeur
10^{-12}	pico	p	.000000000001
10^{-9}	nano	n	.000000001
10^{-6}	micro	µ	.000001
10^{-3}	milli	m	.001
10^{3}	kilo	k	1000
10^{6}	mega	M	1000000
10^{9}	giga	G	1000000000
10^{12}	tera	T	1000000000000

Mesure de la quantité d'information

Base 2

Puissance	Nom	Symbole	Valeur
2^{10}	kilo	k	1024
2^{20}	mega	M	1048576
2^{30}	Giga	G	1073741824

Exemple



(C:) Properties
General
Tools
Label:
Type:
Local Disk
File system:
FAT32
Used space:
1,977,475,072 bytes
1.84GB
Free space:
8,068,784,128 bytes
7.51GB
Drive C
Disk Cleanup...
OK
Cancel
Apply
Used space:
1,977,475,072 bytes
1.84GB
/ 2^30 =

Addition binaire

- Deux valeurs de 1 bit

A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	10

“deux”

Addition binaire

- 2 valeurs de *n*-bits
- Additionner les bits dans chaque position
- Propager les retenues

```
 1   1
  10101        21
+ 11001      + 25
 101110        46
```

Multiplication

- Décimal (rappel)
- On place d'abord les deux nombres l'un sous l'autre.
- On prend le chiffre des unités du nombre du bas et on le multiplie avec tous les chiffres du nombre du haut en commençant par la droite.
- On multiplie de la même façon les chiffres du nombre du haut avec le chiffre des dizaines du nombre du bas et on continue de la même manière avec les autres chiffres. On inscrit les réponses de ces multiplications l'un au-dessous de l'autre.
- On additionne des réponses des multiplications, ainsi on obtiendra la réponse de la grande multiplication.

$$
\begin{array}{r}
35 \\
\times\ 105 \\
\hline
175 \\
000 \\
35 \\
\hline
3675
\end{array}
$$

Multiplication

- 2 valeurs de 1-bit

A	B	$A \times B$
0	0	0
0	1	0
1	0	0
1	1	1

Multiplication

- 2 valeurs de *n*-bits
- Comme les valeurs décimales

```
    1110
  x 1011
--------
    1110
   1110
  0000
 1110
--------
10011010
```