Chapitre 2 – Bases de numération

I. Introduction

L'homme a dû s'inventer, depuis les temps ancestraux, des moyens pour compter, dénombrer les objets de son entourage. Des cailloux, coquillages, de tailles plus ou moins grandes, ont servi à déterminer les quantités, nombres, grandeur des choses afin de servir de système d'échange équitable.

Pour transmettre, stocker de l'information, d'un individu à un autre, on utilise en permanence des systèmes de codage : langues orales, langues écrites, chiffres (arabes, romains, etc..), signes (gestuels, dessins, idéogrammes, etc..).

Pour un ordinateur, le problème est le même, il faut un système de codage, on dit aussi un langage.

Un **langage**, c'est :

- un **alphabet** : ensemble des symboles utilisés.
- des **mots**, des **phrases** : combinaisons des éléments (des lettres) de l'alphabet.
- une **syntaxe** : ensemble de règles qui définissent comment construire ces mots, ces phrases.

On peut dire que le dénombrement a nécessité la mise en œuvre d'une forme simplifiée de langage pour déterminer des grandeurs.

La globalisation des échanges a nécessité la construction de systèmes de mesure connus de tous, normalisés. Les systèmes de numération sont apparus pour représenter des grandeurs, des nombres.

II. Bases de numération

A. Introduction, forme polynomiale

La **numération** désigne les **techniques de représentation** des nombres.

Une base de numération est **un système de représentation des nombres** :
_ Un **ensemble de symboles** (chiffres, en général), qui détermine la base
_ Des règles d'agencement de ces symboles (position, rang)
_ La grandeur exprimée par ce nombre (poids)

Le **rang** est la **position d'un chiffre dans un nombre** ; le rang se compte en partant de la droite, à partir de 0.

Le **poids** exprime **l'importance du chiffre dans un nombre**, en fonction de son rang. Il est exprimé comme un coefficient multiplicateur du chiffre pour obtenir sa grandeur, son importance.

$$\textbf{Poids} = \textbf{Base}^{\textbf{Rang}}$$

La **base** correspond aux nombre de symboles utilisés dans un système de numération et dont l'élévation à la puissance du rang définit la grandeur du nombre.

Le système décimal est celui que nous utilisons chaque jour. Cependant, dans des contextes particuliers, d'autres systèmes de numérations ont été conçus, basés sur un principe identique mais utilisant un jeu de symboles différent.

Un **nombre dans une base B** peut être représenté comme une **succession de chiffres** :

$$(\ C_n\ \ C_{n-1}\ \ .\ .\ .\ C_2\ \ C_1\ \ C_0)_B$$

La valeur de ce nombre dans la base que nous connaissons (base 10) sera obtenue en utilisant la forme polynomiale (plusieurs termes, plusieurs parties) :

$$(\ C_n\ \ C_{n-1}\ \ C_{n-2}\ .\ .\ .\ C_2\ \ C_1\ \ C_0)_B$$

$$= (\ C_n \ X\ B^n\ +\ C_{n-1}\ X\ B^{n-1}\ +\ .\ .\ .\ C_2\ X\ B^2\ +\ C_1\ X\ B^1\ +\ C_0\ X\ B^0\ +)_{10}$$

soit la somme pour n de 0 à rangMax de C_n*Base^n i

B. Système décimal, base 10

<u>10 symboles : { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9}</u>

à chaque rang, le symbole est associé à un poids : 10 élevé à la puissance du rang.

Rang	2	1	0

Poids	10^2 (centaines)	10^1 (dizaines)	10^0 (unités)
Symboles du nombre	4	6	8
Valeur	$4 \times 10^2 = 400$	$6 \times 10^1 = 60$	$8 \times 10^0 = 8$
Soit	400 + 60 + 8 = 468		

C. Système hexadécimal, base 16

16 symboles : { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

à chaque rang, le symbole est associé à un poids : 16 élevé à la puissance du rang.

Les symboles hexadécimaux A, B, C, D, E, F ont une valeur exprimée en base 10 de :
- 10 pour A, 11 pour B, 12 pour C, 13 pour D, 14 pour E, 15 pour F.

Rang	2	1	0
Poids	$16^2 = 256$	$16^1 = 16$	$16^0 = 1$
Symboles du nombre	1	F	2
Valeur	$1 \times 16^2 = 256$	$F \times 16^1$	$2 \times 16^0 = 2$
		Soit : $15 \times 16^1 = 240$	
Soit	256 + 240 + 2 = 498		

D. Système octal, base 8

8 symboles : { 0, 1, 2, 3, 4, 5, 6, 7}

à chaque rang, le symbole est associé à un poids : 8 élevé à la puissance du rang.

Rang	2	1	0
Poids	$8^2 = 64$	$8^1 = 8$	$8^0 = 1$
Symboles du nombre	2	7	1
Valeur	$2 \times 8^2 = 128$	$7 \times 8^1 = 56$	$1 \times 8^0 = 1$
Soit	128 + 56 + 1 = 185		

III.Le système binaire, une représentation pour l'ordinateur

Un ordinateur est une machine électronique, et dans ce domaine on connaît surtout des phénomènes physiques à 2 états :

- circuit électrique ouvert ou fermé
- transistor conducteur ou saturé
- aimantation Nord-Sud ou Sud-Nord, etc.

On en déduit que l'alphabet utilisé aura 2 éléments : 0 et 1 (Par convention). C'est ce que l'on appelle le langage binaire
Le système binaire a été retenu pour représenter les nombres au sein de l'ordinateur. Il utilise en effet 2 symboles qu'on peut faire correspondre à 2 états électriques d'une machine électronique.

A. Système binaire, base 2

2 symboles : { 0, 1}

à chaque rang, le symbole est associé à un poids : 2 élevé à la puissance du rang.

Rang	2	1	0
Poids	$2^2 = 4$	$2^1 = 2$	$2^0 = 1$
Symboles du nombre	1	0	1
Valeur	$1 \times 2^2 = 4$	$0 \times 2^1 = 0$	$1 \times 2^0 = 1$
Soit	5		

IV. Conversions entre bases de numération

A. Base B vers base 10

On utilise la forme polynomiale d'un nombre : celle-ci permet d'obtenir une grandeur exprimée en base 10.

B. Base 10 vers base B

Deux techniques :

- Divisions entières par la base pour la partie entière et multiplication par la base pour la partie fractionnaire (décimale)
- Utilisation d'un tableau des poids et recherche de la valeur à convertir

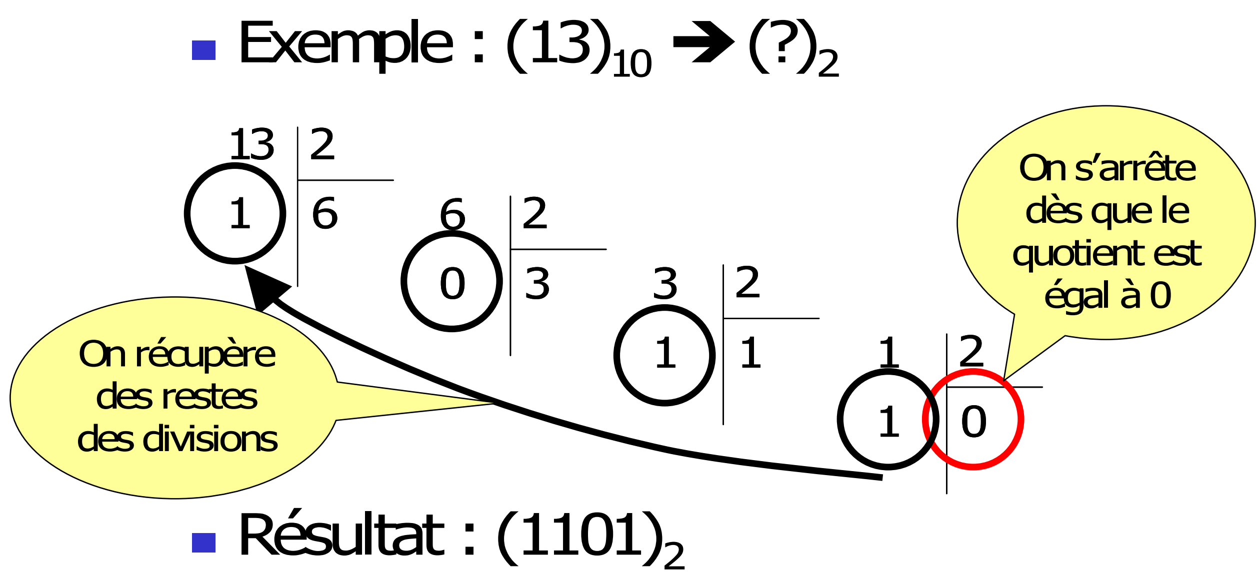


Figure 1 - conversion de la partie entière (exemple décimal vers binaire)

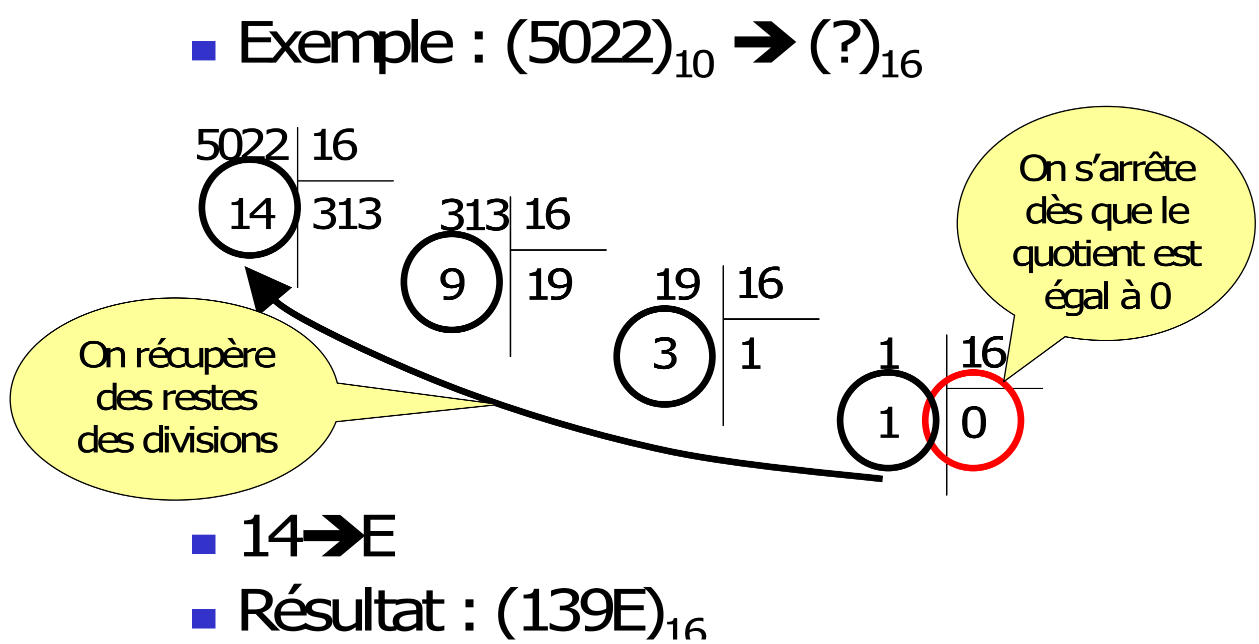


Figure 2 - conversion de la partie entière (exemple décimal vers hexadécimal)

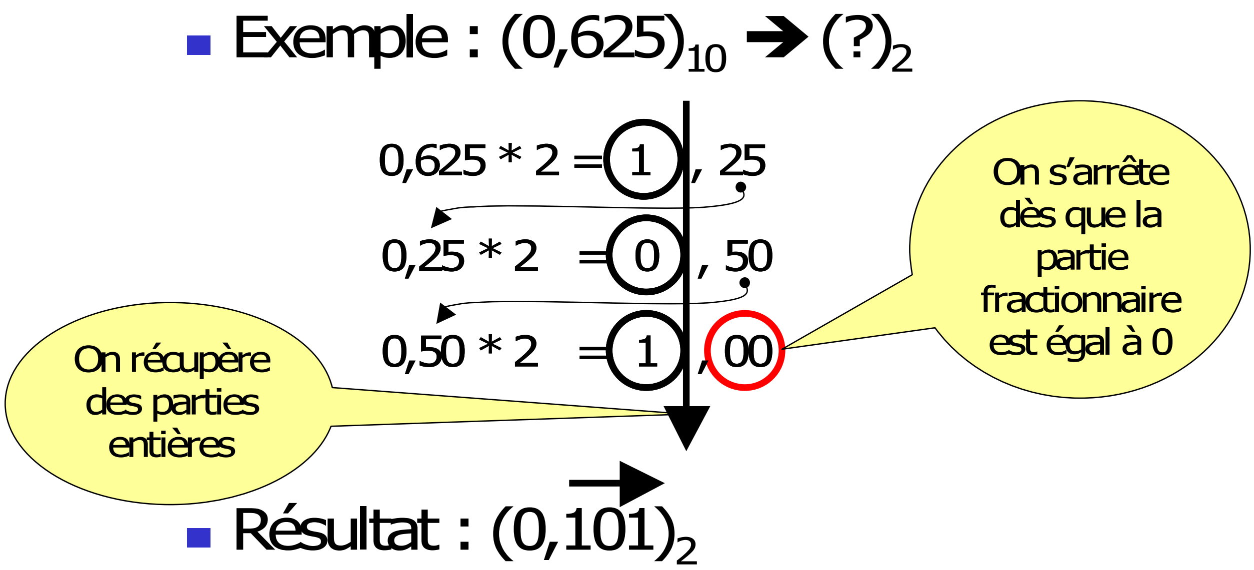


Figure 3 - conversion de la partie fractionnaire (exemple décimal vers binaire)

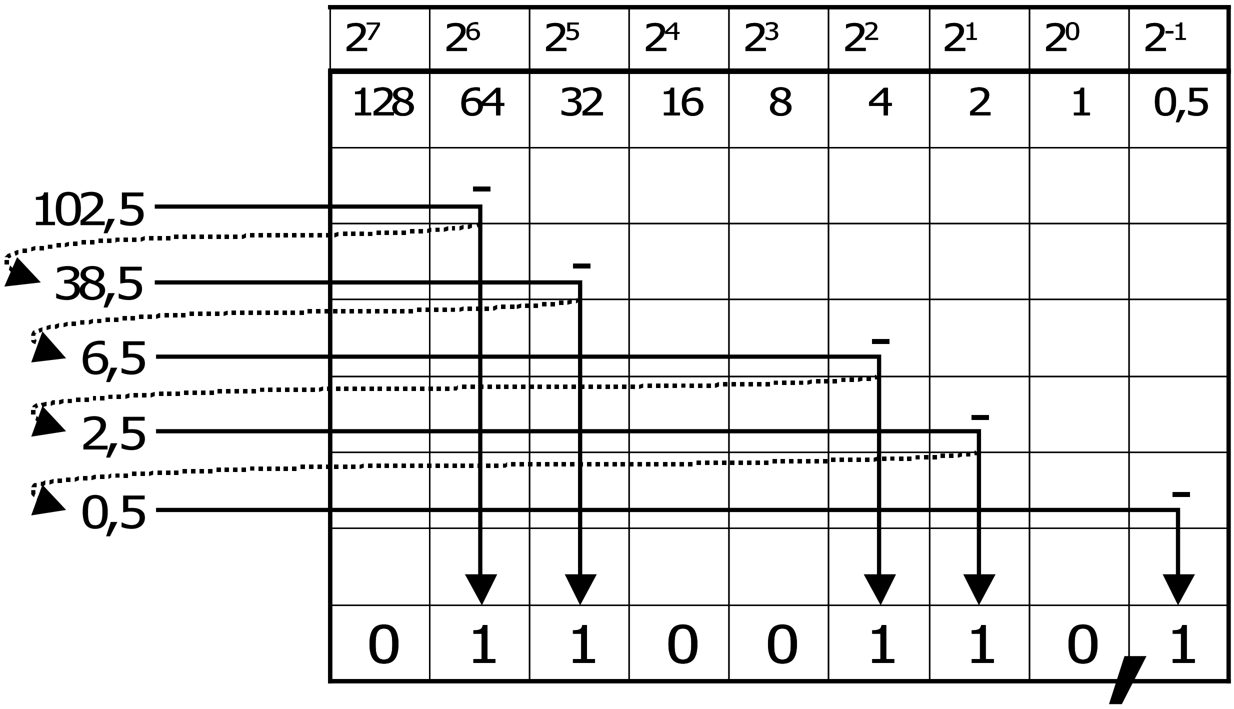


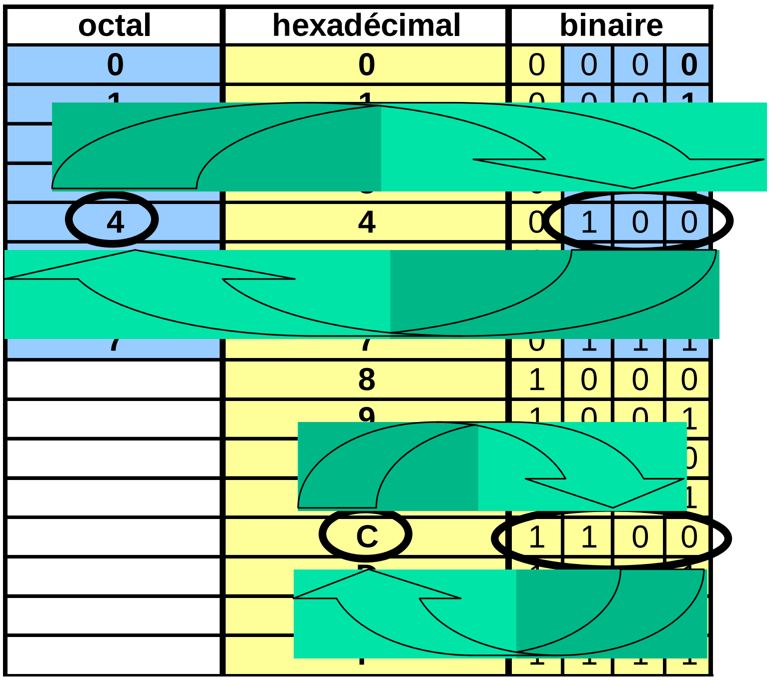
Figure 4 - tableau de poids

C. Base 2 vers bases 8 ou 16, et inversement

Un nombre en base 8 peut être codé sur 3 bits, un nombre en base 16 sur 4 bits. Il y a donc correspondance directe entre 1 chiffre octal ou hexadécimal et sa représentation en binaire et inversement.

Un nombre octal est constitué de chiffres (0 à 7) qui constituent le regroupement de 3 chiffres binaires ($2^3 = 8$). Le regroupement se fait en commençant par la droite.

Un nombre hexadécimal est constitué de chiffres (0 à F) qui constituent le regroupement de 4 chiffres binaires ($2^4=16$). Le regroupement se fait en commençant par la droite.



$(4)_8 \leftrightarrow (0100)_2$

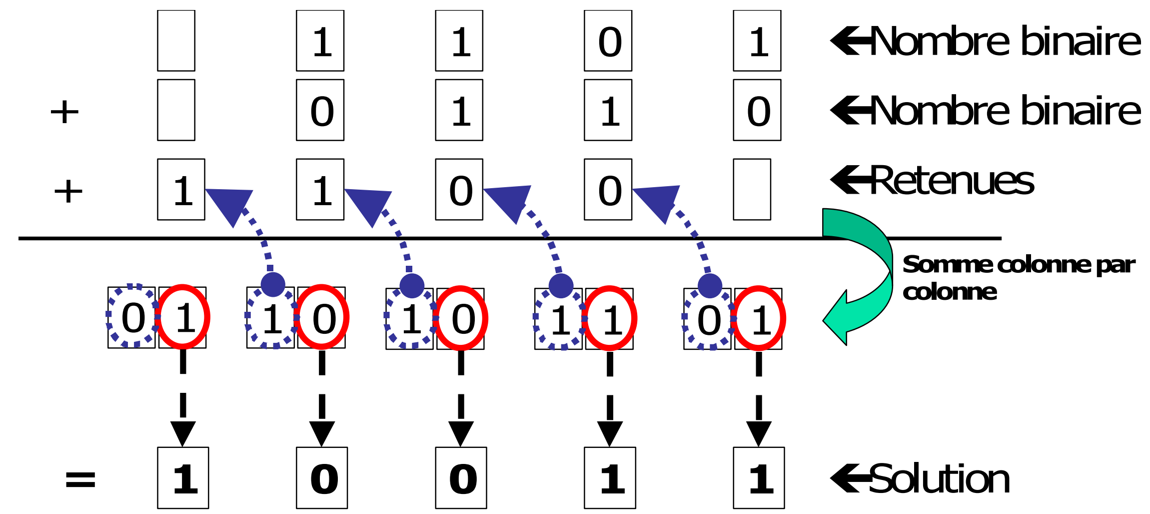
$(C)_{16} \leftrightarrow (1100)_2$

V. Arithmétique binaire

L'arithmétique binaire (ou pour tout autre base) se comporte de la même manière qu'en décimal.

A. Addition

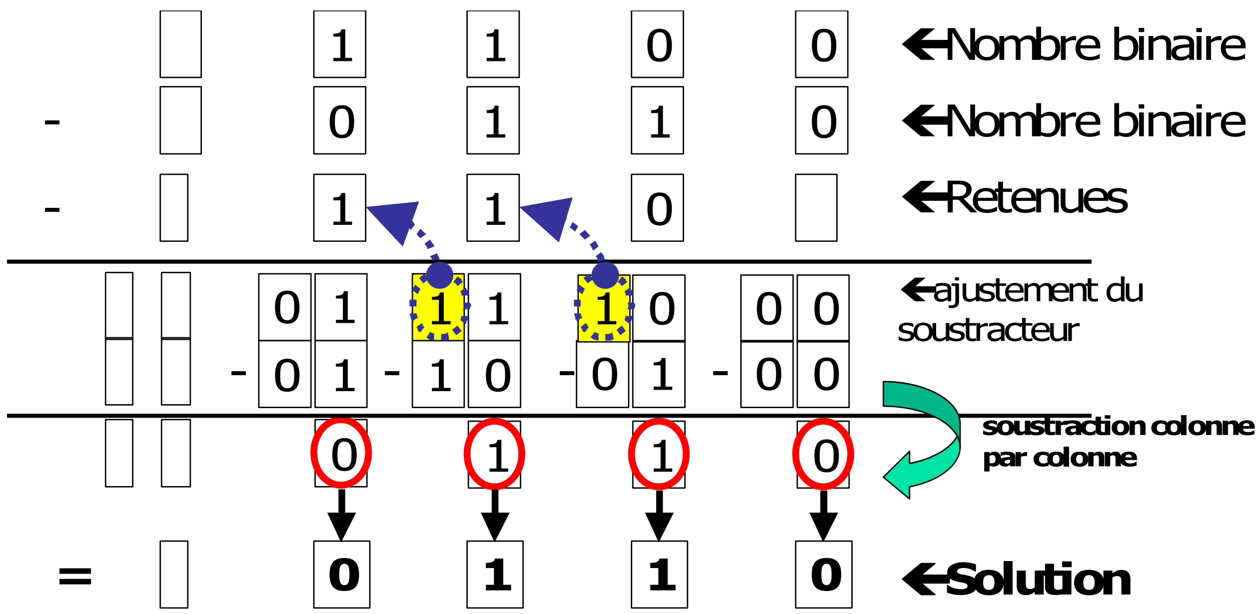
					Retenue additive
0	+	0	=	0	
0	+	1	=	1	
1	+	0	=	1	
1	+	1	=	0	1



(Soit en décimal 13+6=19)

B. Soustraction

					Retenue soustractive
0	-	0	=	0	
0	-	1	=	1	1
1	-	0	=	1	
1	-	1	=	0	



(Soit en décimal 12-6=6)

C. Multiplication

0	x	0	=	0

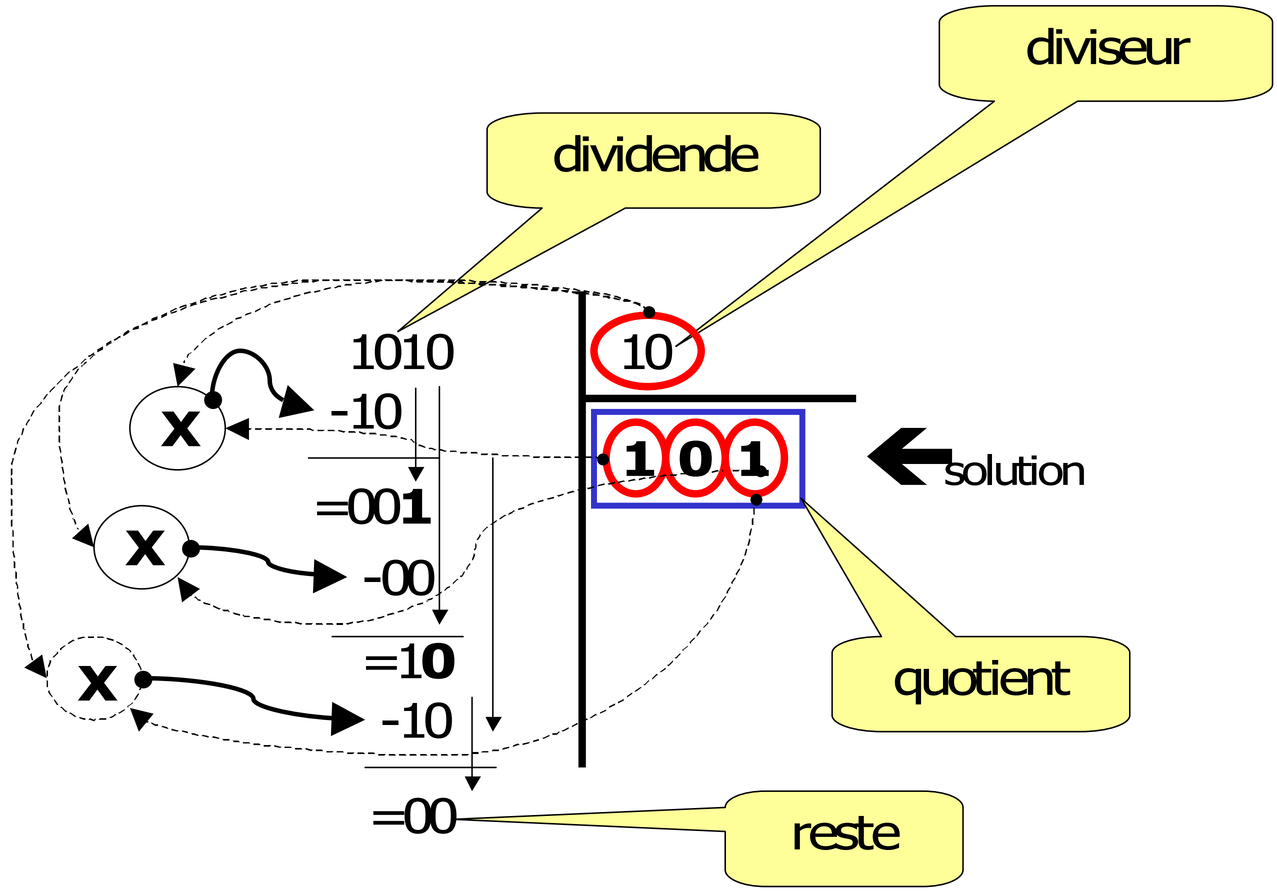
0	x	1	=	0
1	x	0	=	0
1	x	1	=	1

			1	0	0	1	
*				1	0	1	
			1	0	0	1	
+		0	0	0	0	.	
+	1	0	0	1	.	.	
	0	0	0	0	0	(0)	← **retenues**
=	1	0	1	1	0	1	

(Soit en décimal 9 X 5 = 45)

D. division

0	/	0	=	impossible
0	/	1	=	0
1	/	0	=	impossible
1	/	1	=	1



E. Compter en décimal, et dans les autres bases

Lorsqu'on compte dans une base, quand on a épuisé les chiffres de la base, on passe au rang supérieur . Par exemple :

- En décimal : après '9' (unités), on ajoute '1' au poids supérieur (dizaines) et on recommence à 0 unités.
- En hexadécimal : après 'F' (unités), on ajoute '1' au poids supérieur et on recommence à 0 unités
- En octal : après '7' (unités), on ajoute '1' au poids supérieur et on recommence à 0 unités
- En binaire : après '1' (unités), on ajoute '1' au poids supérieur et on recommence à 0 unités
- En base 4 (4 symboles : {0, 1, 2, 3} : après '3' (unités), on ajoute '1' au poids supérieur et on recommence à 0 unités

Décimal	Hexadécimal	Octal	binaire	Base 4
0	0	0	0	0
1	1	1	1	1
2	2	2	**10**	2
3	3	3	11	3
4	4	4	**100**	**10**
5	5	5	101	11
6	6	6	**110**	12
7	7	7	111	13
8	8	**10**	**1000**	**20**

9	9	11	1001	21
10	A	12	**1010**	22
11	B	13	1011	23
12	C	14	**1100**	**30**
13	D	15	1101	31
14	E	16	**1110**	32
15	F	17	1111	33
16	**10**	**20**	**10000**	**100**
17	11	21	10001	101
18	12	22	**10010**	102
19	13	23	10011	103
20	14	24	**10100**	**110**
21	15	25	10101	111
22	16	26	**10110**	112
23	17	27	10111	113
24	18	**30**	**11000**	**120**
25	19	31	11001	121
26	1A	32	**11010**	122
27	1B	33	11011	123
28	1C	34	**11100**	**130**