# Chapitre I : Représentation et codification des nombres

## I.1. Objectif de ce chapitre

A l'issu de ce chapitre, l'apprenant sera capable de :

- Se familiariser avec l'environnement informatique
- Identifier les principaux composants d'un ordinateur
- Comprendre les systèmes de numération
- Convertir entre les bases

## I.2. Définition de l'Informatique

L'informatique est la science qui s'occupe du traitement automatique de l'information. Le terme "*informatique*" vient de la contraction des mots "*information*" et "*automatique*".

### I.2.1. Rôle de l'informatique

L'informatique a pour rôle :

- La conception et la construction des ordinateurs;
- Le fonctionnement et la maintenance des ordinateurs;
- Leur exploitation (utilisation des ordinateurs dans les différents domaines d'activités).

## I.3. Architecture et fonctionnement d'un ordinateur

L'ordinateur est composé de deux parties essentielles : le matériel (hardware) et le logiciel (software). Dans ce cours, nous nous focalisons sur la partie matérielle de l'ordinateur et expliquant son architecture et sa structure logique. Par la suite, nous présentons une structure physique simplifie.

La Figure I.1 représente l'architecture générale d'un ordinateur

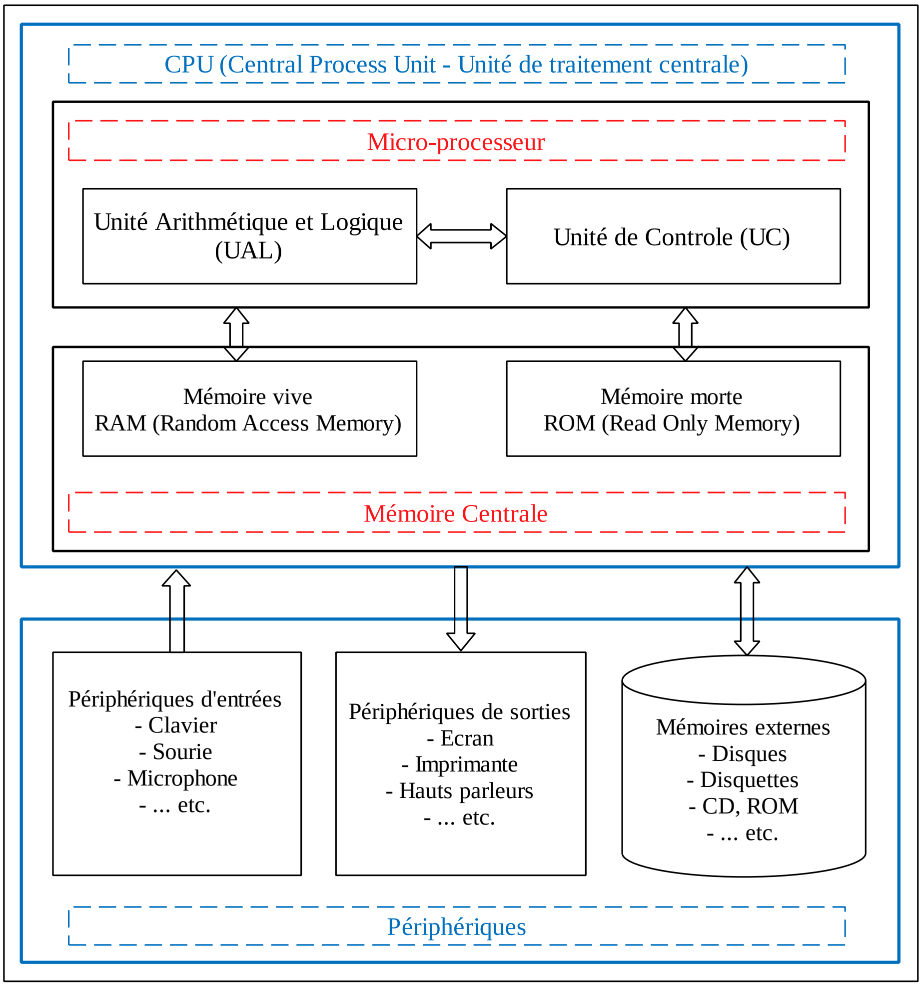



**Figure I.1 :** Architecture générale d'un ordinateur

**CPU :** Central Process Unit (Unité de traitement centrale)
**UAL :** Unité Arithmétique et Logique
**UC :** Unité de Contrôle ou de commande
**MC :** Mémoire Centrale

Un ordinateur est constitué, d'une manière générale, d'une unité centrale de traitement, d'une mémoire centrale et d'un ensemble de périphériques.

**a) Unité Centrale de Traitement (UCT/CPU) ou bien Processeur (Microprocesseur) :** Le processeur est le cœur de l'ordinateur, chargé de traiter et d'exécuter les programmes (les instructions) de l'ordinateur. Il comporte, essentiellement, deux unités : l'Unité de Contrôle (UC) et l'Unité Arithmétique et Logique (UAL).

- **Unité de Contrôle (UC) :** Elle est responsable de la lecture en mémoire centrale et du décodage des instructions ;

- **Unité Arithmétique et Logique (UAL) :** Aussi appelée unité de traitement, cette dernière exécute les instructions. Elle se charge d'effectuer toutes les opérations arithmétiques, logiques et relationnelles contenues dans l'instruction et effectue aussi des échanges de données (réception/transmission) avec la mémoire vive.

Ces deux unités communiquent, non seulement, avec la mémoire centrale, mais également avec les différents périphériques.

**b) Mémoire Centrale (ou Principale) :** Elle est constituée d'une mémoire vive (RAM : Random Access Memory) et d'une mémoire morte (ROM : Read Only Memory).

- **RAM :** Une RAM, ou mémoire vive, sert au stockage temporaire des données et du programme à exécuter. Les mémoires vives sont en général volatiles : elles perdent leurs informations en cas de coupure du courant électrique.
- **ROM :** Une ROM ou mémoire morte, par opposition à la RAM, ne s'efface pas à la coupure du courant. Elle peut être lue mais pas (ou peu de fois) écrite. Elle conserve des programmes nécessaires au fonctionnement du matériel, surtout lors du démarrage (avant le chargement du système d'exploitation dans la RAM).

**c) Périphériques :** Les périphériques sont composés de périphériques d'entrée, de sortie et des périphériques d'entrée/sortie.

- **Périphériques d'entrée :** C'est l'ensemble de périphériques qui permet de transmettre des données à l'ordinateur : Le clavier, la souris, le microphone, la webcam, etc.).
- **Périphériques de sortie :** C'est l'ensemble de périphériques qui permet de recevoir des données de l'ordinateur et de les renvoyer vers l'extérieur : L'écran, l'imprimante, les haut-parleurs, etc.).
- **Périphériques d'entrée/sortie :** C'est l'ensemble de périphériques qui permet, à la fois, de transmettre et de recevoir des données. Ils sont, également, appelés périphériques de stockage ou mémoires externes (mémoires auxiliaires) : La disquette, le CD-ROM, la clé USB, le disque dure externe, etc.).

La figure I.2 résume la partie matériel de l'ordinateur

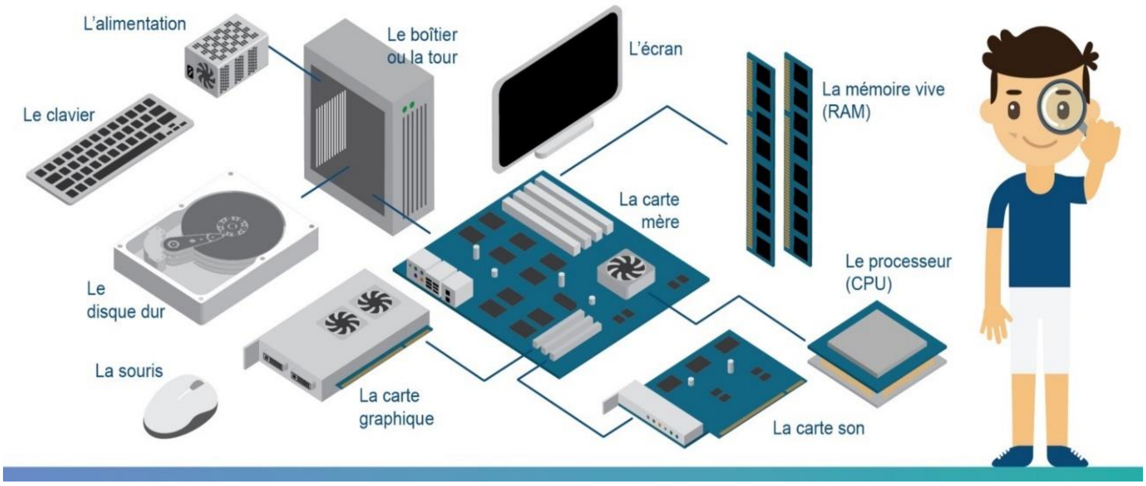



**Figure I.2 :** Partie matériel de l'ordinateur

## I.4. Les systèmes de codage des informations

Toute sorte d'information manipulées par un ordinateurs (numériques, textuelles, images, sons, vidéos, etc.) est représentée par des séquences de deux chiffres : 0 et 1. Ces deux chiffres sont désignés par **BIT** (BInary degiT).

Donc un **bit** est soit 0 ou bien 1 qui est représenté par l'ordinateur par deux états électroniques: soit il y a présence d'une impulsion électrique (c'est l'état 1), soit il y a absence d'impulsion électrique (c'est l'état 0).

**Les systèmes de numération ou le codage** d'une information consiste à établir une correspondance entre la représentation externe (habituelle) de l'information (le caractère A ou le nombre 36 par exemple), et sa représentation interne dans la machine, qui est une suite de bits.

Un système de numération se définit par deux éléments: la **base** du système et les **symboles** utilisés. Les systèmes les plus utilisés sont:

- ***Système Binaire (Base 2)*** **:** Les symboles utilisés **{ 0,1}**
- ***Système Décimale (Base10)*** **:** Les symboles utilisés **{ 0,1,2, 3,4,5,6,7,8,9}**
- **Système *Octal (Base 8)*** **:** Les symboles utilisés **{ 0,1,2,3,4,5,6,7}**
- *Système* ***Hexadécimal (Base16)*** **:** Les symboles utilisés **{0,1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, *C, D, E, F}***

  *Avec A=10, B=11, C=12, D=13, E=14, F=15*

### I.4.1. Notation

Un nombre quelconque Nb exprimé dans une base b sera noté comme suit :

$$Nb = (a_{n-1}\, a_{n-2} \dots\ a_1\, a_0)_b$$

*Avec :*

$b$ : la base du système de numérotation

$a_i$ : symbole du système de numérotation, $i=0,\dots,n-1$ *et* $a_i < b$

Exemples :

$Nb_1 = (1995)_{10}\ avec\ a_{3=}1, a_2 = 9, a_1 = 9, a_0 = 5$

$Nb_2 = (243)_8\ avec\ \ a_2 = 2, a_1 = 4, a_0 = 3$

$Nb_3 = (1011010)_2\ avec\ a_6 = 1, a_5 = 0, a_4 = 1, a_{3=}1, a_2 = 0, a_1 = 1, a_0 = 0$

$Nb_4 = (BAC)_8\ avec\ \ a_2 = B, a_1 = A, a_0 = C$

$Nb_5 = (248)_8$ Cette notation est erronée, car le nombre contient un symbole supérieur ou égale à la base.

❖ **Remarques :**

- Quand la base n'est pas mentionnée, on considère qu'on est en base 10.
- Dans une base b, nous avons b chiffres : 0, 1, 2, …., (b-1).

### I.4.2. Conversion d'un nombre d'un système à un autre

**a) Conversion de la base 2, 8, 16 → base 10**

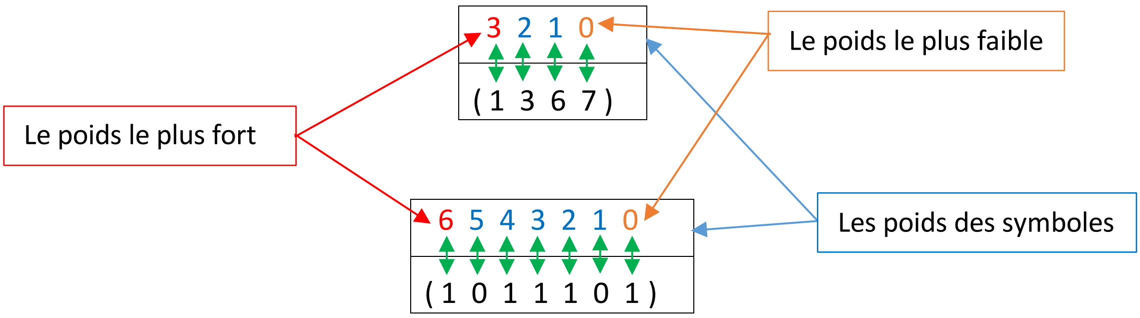
Pour convertir un nombre $Nb = (a_{n-1}\, a_{n-2} \dots\ a_1\, a_0)_b$ de la base b vers la base 10, on effectue le calcul suivant :

$$(Nb)_b = (a_{n-1}\ * b^{n-1} + a_{n-2} * b^{n-2} \dots\ + a_1\ * b^1 + a_0 * b^0)_{10}$$

$$(Nb)_b = \sum_{i=0}^{n-1} a_i * b^i$$

*Elle correspond à la somme des produits de chaque symbole du nombre par le poids correspondant.*

**Le poids des symboles :** il s'obtient en numérotant les symboles à partir de la droite vers la gauche, en commençant du 0.




Exemple 1 :

$(1367)_8 = (?)_{10}$

$(1367)_8 = (1 * 8^3 + 3 * 8^2 + 6 * 8^1 + 7 * 8^0)_{10}$

$(1367)_8 = (759)_{10}$

Exemple 2 :

$(1011101)_2 = (?)_{10}$

$(1011101)_2 = (1 * 2^6 + 0 * 2^5 + 1 * 2^4 + 1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0)_{10}$

$(1011101)_2 = (93)_{10}$

Exemple 3 :

$(2B3)_{16} = (?)_{10}$

$(2B3)_{16} = (2 * 16^2 + B * 16^1 + 3 * 16^0)_{10}$

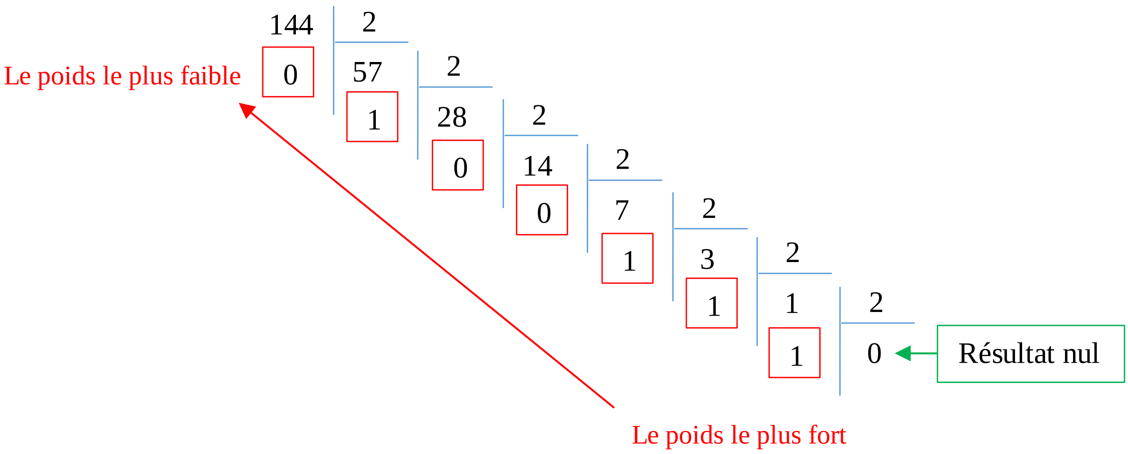
$(2B3)_{16} = (691)_{10}$

**b) Conversion de la base 10 → base 2, 8, 16**

Soit Nb un nombre exprimé dans la base 10, pour trouver son équivalent en base b, on applique la méthode des divisions successives sur b, jusqu'à l'obtention d'un résultat nul. Puis, on récupère les restes des divisions dans le sens inverse, i.e. le dernier reste trouvé représentera le poids le plus fort et le premier reste trouvé sera le poids le plus faible.

**Exemple 1 :**

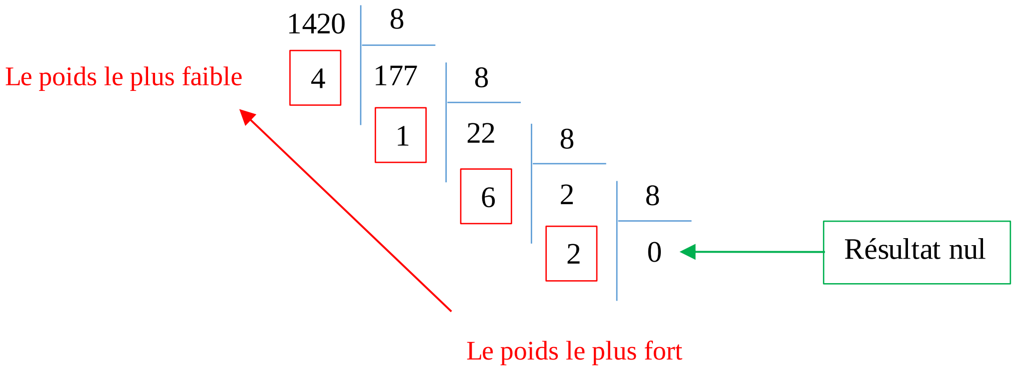
- $(144)_{10} = (?)_2$




- $(144)_{10} = (1110010)_2$

**Exemple 2 :**

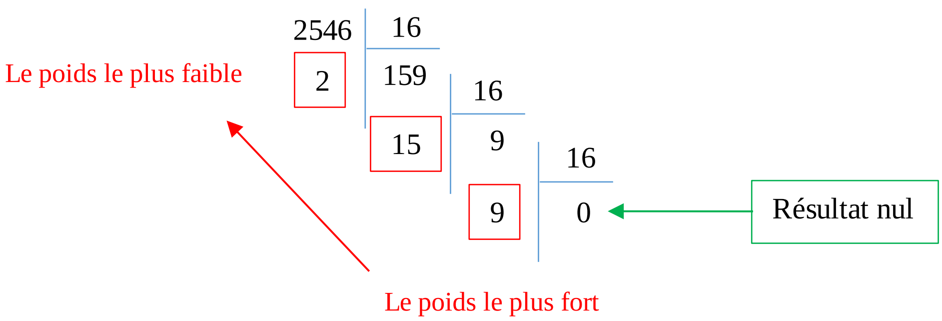
- $(1420)_{10} = (?)_8$




- $(1420)_{10} = (2614)_8$

**Exemple 3 :**

- $(2546)_{10} = (?)_{16}$




- $(2546)_{10} = (9F2)_{16}$

### c) Conversion de la base 8 → base 2

Pour convertir un nombre Nb exprimé en base 8 vers la base 2, nous procédons comme suit:

$$8 = 2^3$$

Il faut donc utiliser **3** bits pour exprimer un seul chiffre octal en binaire.

La représentation des chiffres de la base 8 vers le binaire est comme suit :

- $(7)_8 = (1 * 2^2 + 1 * 2^1 + 1 * 2^0)_2 = (111)_2$
- $(4)_8 = (1 * 2^2 + 0 * 2^1 + 0 * 2^0)_2 = (100)_2$
- $(3)_8 = (0 * 2^2 + 1 * 2^1 + 1 * 2^0)_2 = (011)_2$

| Chiffre en octal | Chiffre équivalent en binaire ($2^2$ $2^1$ $2^0$) |
|---|---|
| 0 | 0 0 0 |
| 1 | 0 0 1 |
| 2 | 0 1 0 |
| 3 | 0 1 1 |
| 4 | 1 0 0 |
| 5 | 1 0 1 |
| 6 | 1 1 0 |
| 7 | 1 1 1 |

Exemple :

- $(743)_8 = (111\ \ 100\ \ 011)_2$

### d) Conversion de la base 16 → base 2

Pour convertir un nombre Nb exprimé en base 16 vers la base 2, nous procédons comme suit:

$$16 = 2^4$$

Il faut donc utiliser **4** bits pour exprimer un seul chiffre hexadécimal en binaire.

La représentation des chiffres de la base 16 vers le binaire est comme suit :

| Chiffre en hexadécimal | Chiffre équivalent en binaire ($2^3$ $2^2$ $2^1$ $2^0$) |
|---|---|
| 0 | 0 0 0 0 |
| 1 | 0 0 0 1 |
| 2 | 0 0 1 0 |
| 3 | 0 0 1 1 |
| 4 | 0 1 0 0 |
| 5 | 0 1 0 1 |
| 6 | 0 1 1 0 |
| 7 | 0 1 1 1 |

| Chiffre en hexadécimal | Chiffre équivalent en binaire ($2^3$ $2^2$ $2^1$ $2^0$) |
|---|---|
| 8 | 1 0 0 0 |
| 9 | 1 0 0 1 |
| A | 1 0 1 0 |
| B | 1 0 1 1 |
| C | 1 1 0 0 |
| D | 1 1 0 1 |
| E | 1 1 1 0 |
| F | 1 1 1 1 |

- $(7)_{16} = (0 * 2^3 + 1 * 2^2 + 1 * 2^1 + 1 * 2^0)_2 = (0111)_2$
- $(B)_{16} = (1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0)_2 = (1011)_2$
- $(3)_{16} = (0 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0)_2 = (0011)_2$
- $(A)_{16} = (1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 0 * 2^0)_2 = (1010)_2$

Exemple :

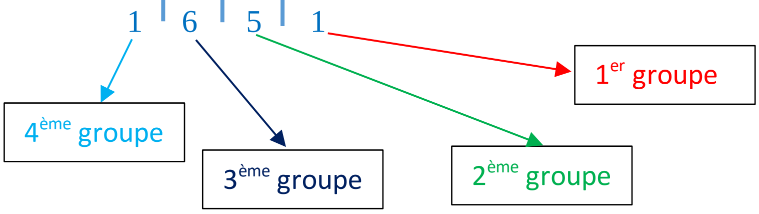
- $(7B3A)_{16} = (0111\ \ 1011\ \ 0011\ \ 1010)_2$

**e) Conversion de la base 2 → base 8**

Pour trouver l'équivalent d'un nombre binaire en octal, il suffit de former des **groupes de 3 bits** chacun ( Puisque **8 = 2$^3$** ), en commençant du poids le plus faible ( à partir de la droite), si le dernier groupe formé possède moins de 3 bits, il suffit de rajouter des 0, puis calculer l'équivalent en octal de chaque groupe.

Exemple :

- $(10110101010)_2 = (?)_8$
- $(010|110|101|010)_2 = (1651)_8$

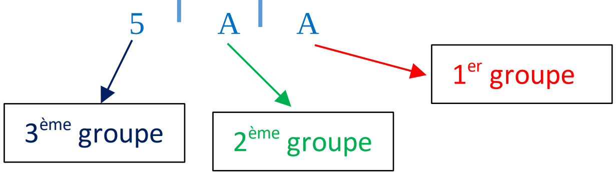



**f) Conversion de la base 2 → base 16**

Pour trouver l'équivalent d'un nombre binaire en Hexadécimal, il suffit de former des **groupes de 4 bits** chacun ( Puisque **16 = 2$^4$** ), en commençant du poids le plus faible ( à partir de la droite), si le dernier groupe formé possède moins de 4 bits, il suffit de rajouter des 0, puis calculer l'équivalent en Hexadécimal de chaque groupe.

Exemple :

- $(10110101010)_2 = (?)_{16}$
- $(0101|1010|1010)_2 = (5AA)_{16}$




**g) Conversion de la base 16 → base 8**

Pour convertir un nombre Nb exprimé en base 16 vers la base 8 ou vice versa, nous devons **passer par une base intermédiaire** tel que le **décimal ou le binaire**, mais le passage par le binaire est beaucoup plus simple.

Exemple :

- $(C9F)_{16} = (1100\ \ 1001\ \ 1111)_2$
- $(\underset{6}{110}|\underset{2}{010}|\underset{3}{011}|\underset{7}{111})_2 = (6237)_8$

**Exercice :** *(Systèmes de numérotation)*
Effectuer les conversions suivantes :

$2022 = (?)_2$ ;   $(1011001101)_2 = (?)_{10}$ ;   $(1011001101)_2 = (?)_8 = (?)_{16}$ ; $(32103)_4 = (?)_2$ ; $(37163)_8 = (?)_2$; $(379)_{10} = (?)_{16}$; $(3A2D)_{16} = (?)_{10}$; $(4D5B)_{16} = (?)_8$

**Corrigé :** *(Systèmes de numérotation)*

Effectuer les conversions suivantes :

$2022 = (11111100110)_2$ ;

$(1011001101)_2 = (717)_{10}$ ;

$(1011001101)_2 = (1315)_8 = (2CD)_{16}$ ;

$(32103)_4 = (1110010011)_2$ ;

$(37163)_8 = (011111001110011)_2$ oubien $(11111001110011)_2$ ;

$(379)_{10} = (17B)_{16}$ ;

$(3A2D)_{16} = (14893)_{10}$ ;

$(4D5B)_{16} = (46533)_8$

## I.5. Opérations arithmétiques de base

Dans ce qui suit, nous allons voir comment les opérations arithmétiques de base sont réalisé sur les systèmes de numération base 2, 8 et 16.

### I.5.1. Opérations arithmétique en binaire (Base 2)

#### a) Addition binaire

L'addition en base 2 fonctionne comme l'addition en décimal, mais attention car en binaire, $\boxed{\mathbf{1 + 1 = 10}}$ car $(2)_{10} = (10)_2$ : il faut donc placer 0 et mettre une retenue de 1 sur le bit suivant.

**Les règles de l'addition**

- 0 + 0 = 0
- 0 + 1 = 1
- 1 + 0=1
- 1 + 1 = 10, On écrit "0" et on reporte "1" sur le bit de rang supérieur (rang de gauche)

### b) Soustraction binaire

Dans la soustraction binaire, on procède comme en décimal. Quand la quantité à soustraire est supérieure à la quantité dont on soustrait, on emprunte 1 au voisin de gauche. En binaire, ce 1 ajoute 2 à la quantité dont on soustrait, tandis qu'en décimal il ajoute 10.

**Les règles de la soustraction**

- 0 - 0 = 0
- 0 - 1 = (on emprunte "1" ce qui fait 10-1, on écrit "1" et on retient 1)
- 1 - 0 = 1
- 1 - 1 = 0

### c) Multiplication binaire

La multiplication binaire s'effectue selon le principe des multiplications décimal, on multiplie donc le multiplicande par chacun des bits du multiplicateur. On décale les résultats intermédiaires obtenus et on effectue ensuite l'addition de ses résultats partiels.

**Les règles de la multiplication**

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

### d) Division binaire

La division binaire s'effectue à l'aide de soustractions et de décalages, comme la division décimale, sauf que les digits du quotient ne peuvent être que 1 ou 0. Le bit du quotient est 1 si on peut soustraire le diviseur, sinon il est 0.

**Les règles de la division**

- 0 : 0 = /
- 0 : 1 = 0
- 1 : 0 = /
- 1 : 1 = 1