RÉPUBLIQUE ALGÉRIÈNNE DÉMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA
FACULTE DE TECHNOLOGIE

DEPARTEMENT DE SCOLE COMMUN

TP Informatique 01

Dr. Bilal BOUDJELLAL

Professeur à l'université Mohamed Boudiaf – M'sila

Version 1 – Octobre 2018

bilal.boudjellal@gmail.com

Avant-propos

Les lignes qui suivent et qui concernent principalement les étudiants de 1ère année socle commun ST de l'université Mohamed Boudiaf - M'sila. Le but de ce document est d'aider les étudiants à mieux comprendre la matière "***TP - Informatique 1***".

N'hésitez pas à me contacter en cas de besoin Bon travail !

Enseignant : Dr. Bilal BOUDJELLAL

Contact : par mail au Bilal.Boudjellal@gmail.com

Disponibilité :

- **Au bureau :** Lundi, Mardi de 10h00 -11h00
- **Par émail :** toute question en relation avec le cours doit être envoie par émail, je m'engage à répondre aux questions par mail dans un délai de 48 heures qui suivent la réception du message, sauf en cas des imprévus.

Informations sur le TP :

- **Faculté :** Technologie
- **Département :** Socle Commun ST
- **Public cible :** 1ère année Licence
- **Intitulé du cours :** Informatique 01
- **Crédit :** 04
- **Coefficient :** 02
- **Durée :** 14 semaines
- **Horaire :** 01h30
- **Salle :** SI2

Chapitre I. Les systèmes de numération, codage

I.1 Introduction

La création de la numération est un des faits les plus marquants de l'histoire de l'humanité. Si la plupart des civilisations ont adopté le système décimal, c'est qu'il a toujours été naturel de compter sur ses doigts. L'utilisation des phalanges et des articulations permit même d'améliorer ce simple procédé connu de tous.

Aujourd'hui nos ordinateurs, 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** (**BI**nary degi**T**). 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***).

Mais comment représenter, au sein d'un système numérique, cette diversité des objets du monde réel ou virtuel ? Quelles sont les techniques utilisées pour représenter numériquement les grandeurs qui nous entourent ?

I.2 Systèmes de numération

On utilise les "*systèmes de numération*" pour compter des objets et de les représenter par des nombres.

Trois notions interviennent dans un système :

- La base **B** du système, c'est un nombre entier quelconque.
- Les digits du système sont des caractères tous différents et représentent chacun un élément de la base; il y en a donc **B** au total. Alors, toute *base* ***B*** est composée de ***B*** chiffre de ***0*** à ***B-1***.
- Poids du digit selon son rang.

En informatique, outre le système **décimal** (*base **10***), on utilise très fréquemment le système **binaire** (*base **2***) puisque l'algèbre booléenne est à la base de l'électronique numérique (une impulsion électrique). On utilise aussi très souvent le système **octal** (*base **8***) et le système **hexadécimal** (*base **16***) du fait de sa simplicité d'utilisation et de représentation pour les mots machines (il est bien plus simple d'utilisation que le binaire).

Il existe également d'autres bases de numérotation. Mais, toutes ces bases étant des puissances de deux:

$$\mathbf{B} = \mathbf{2}^{\mathbf{N}} \tag{I.1}$$

Avec $N \in \mathbb{N}^*$.

I.2.1 Le système décimal

Nous utilisons le système décimal dans nos activités quotidiennes. Pour ce numération, les notions interviennent dans cette système sont :

- La base du système décimal est '**10**'.
- Les digits du système binaire sont {***0, 1, 2, 3, 4, 5, 6, 7, 8, 9***}; alors, il y en a donc **10** chiffres au total.

Par exemple, le nombre 1011 exprimé en binaire signifie:

Nous disposons de dix chiffres différents de 0 à 9 pour écrire tous les nombres.

I.2.2 Le système binaire

Dans la numération binaire, les notions interviennent dans cette système sont :

- La base du système binaire est '**2**'.
- Les digits du système binaire sont {***0, 1***}; alors, il y en a donc **2** chiffres au total, appelés dans ce cas "**BIT**" (**B**inary dig**IT**).

I.2.3 Le système octale

Pour la numération octale, les notions interviennent dans cette système sont :

- La base du système binaire est '**8**'.
- Les digits du système binaire sont {***0, 1, 2, 3, 4, 5, 6, 7***}; alors, il y en a donc **8** chiffres au total.

I.2.4 Le système hexadécimale

Pour la numération hexadécimale, les notions interviennent dans cette système sont :

- La base du système binaire est '**16**'.
- Les digits du système binaire sont {***0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F***}; alors, il y en a donc **16** chiffres au total.

Avec :

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

I.3 Les systèmes de codage des informations

Nous utilisons le *système décimal* (*base 10*) dans nos activités quotidiennes. Ce système est basé sur dix symboles, de *0 à 9*, avec une unité supérieure (dizaine, centaine, etc.) à chaque fois que dix unités sont comptabilisées. C'est un *système positionnel*, c'est-à-dire que l'endroit où se trouve le symbole définit sa valeur.

D'une manière générale, dans une *base* ***B***, l'entier naturel **X** composée de **N** chiffres s'écrit sous la forme polynomiale suivante:

$$(X)_B = (X_N \times B^{N-1}) + \ldots + (X_3 \times B^2) + (X_2 \times B^1) + (X_1 \times B^0) \qquad (I.2)$$

Avec X_N est un des chiffres de la base **B**.

Par exemple, soit un nombre décimal ***X*** = ***2569***. Ce nombre est la somme de **9** unités, **6** dizaines, **5** centaines et **2** milliers.

On peut écrire alors:

$$X = (2 \times 1000) + (5 \times 100) + (6 \times 10) + (9 \times 1)$$
$$X = (2 \times 10^3) + (5 \times 10^2) + (6 \times 10^1) + (9 \times 10^0)$$

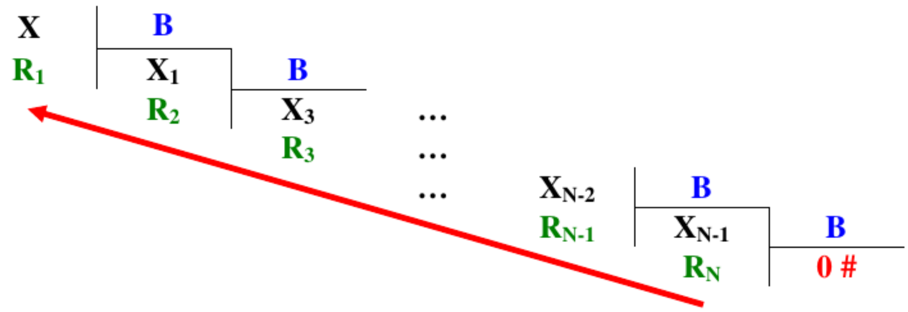
Avec :

- La base de ce nombre est '**10**'.
- Ce nombre est composé de **4** chiffres qui sont {***2, 5, 6 et 9***}. Le chiffre de droite est celui des unités. Celui de gauche est celui qui a le poids le plus élevé.
- Les puissances {***0, 1, 2, 3***} sont le rang de chaque chiffre.

I.3.1 Conversion décimal vers base B

La conversion **décimale** vers ***base B*** est obtenu par divisions euclidiennes successives par **B** et on prend les restes dans le sens inverse. (Le dernier reste est le poids fors, c'est à dire le plus à gauche, et le premier reste est le poids faible, c'est à dire le plus à droite).

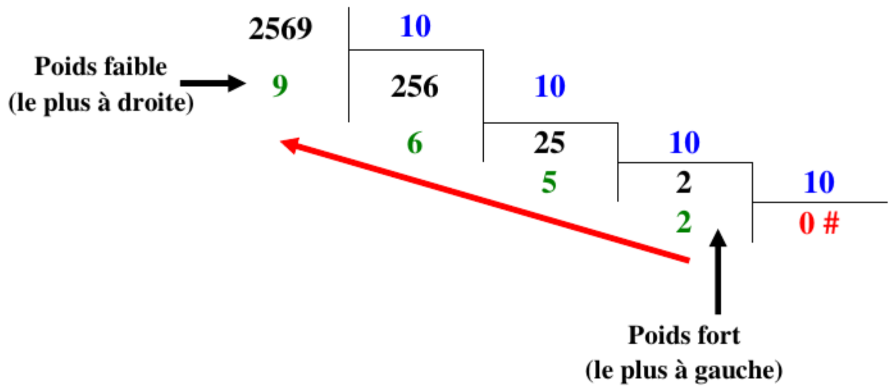
Le schéma ci-dessous explique la méthode mieux qu'un long discours :



Enfin, le résultat sera la juxtaposition des restes et écrit comme suit:

$$(\mathbf{X})_{10} = (\mathbf{R_N}\ \ \mathbf{R_{N-1}}\ \ \ldots\ \ \mathbf{R_3}\ \ \mathbf{R_2}\ \ \mathbf{R_1})_{\mathbf{B}} \qquad \text{(I.3)}$$

Par exemple :



Alors :

$$(\mathbf{2569})_{10} = (\mathbf{2}\quad \mathbf{5}\quad \mathbf{6}\quad \mathbf{9})_{10}$$

I.3.2 Conversion base B vers décimal

Pour passer d'un nombre en ***base B*** à un nombre en ***base 10*** (**décimal**), on utilise l'écriture polynomiale décrite précédemment. Le diagramme ci-dessous explique la conversion du nombre **X** composé de **N** chiffres de la *base* ***B*** en décimal:

$$
\begin{array}{ccccccccccc}
X_1 & & X_2 & & \cdots & & X_{N-1} & & X_N & & \\
\times & & \times & & \cdots & & \times & & \times & & \\
B^{N-1} & & B^{N-2} & & \cdots & & B^{1} & & B^{0} & & \\
\downarrow & & \downarrow & & \cdots & & \downarrow & & \downarrow & & \\
X_1 \times B^{N-1} & + & X_2 \times B^{N-2} & + & \cdots & + & X_{N-1} \times B^{1} & + & X_N \times B^{0} & = & (Y)_{10}
\end{array}
$$

Par exemple :

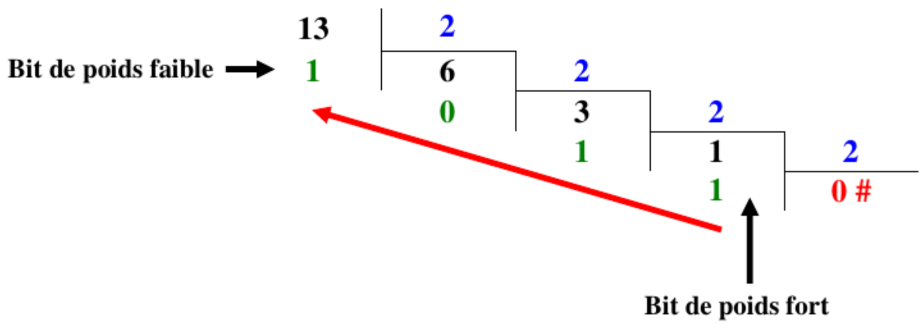
$$
\begin{array}{ccccccccc}
2 & & 5 & & 6 & & 9 & & \\
\times & & \times & & \times & & \times & & \\
10^{3} & & 10^{2} & & 10^{1} & & 10^{0} & & \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \\
2 \times 10^{3} & + & 5 \times 10^{2} & + & 6 \times 10^{1} & + & 9 \times 10^{0} & & \\
\parallel & & \parallel & & \parallel & & \parallel & & \\
2000 & + & 500 & + & 60 & + & 9 & = & (2569)_{10}
\end{array}
$$

Alors :

$$(2 \quad 5 \quad 6 \quad 9)_{10} = (2569)_{10}$$

I.3.3 Conversion décimal – binaire

Soit $(13)_{10}$ un nombre décimal à convertir en binaire (*base* 2). On utilise la division euclidienne, encore appelée division entière :

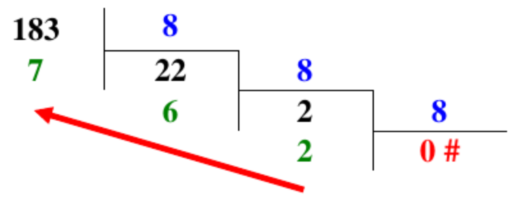


Le résultat est donc:

$$(13)_{10} = (1101)_2$$

I.3.4 Conversion décimal – octal

Soit **(183)**$_{10}$ un nombre décimal à convertir en octal (*base 8*). On utilise la division euclidienne:

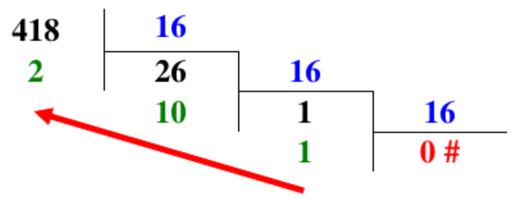


Le résultat est donc:

$$(183)_{10} = (267)_8$$

I.3.5 Conversion décimal – hexadécimal

Soit **(183)**$_{10}$ un nombre décimal à convertir en hexadécimal (*base 16*). On utilise la division euclidienne:



Le résultat est donc:

$$(418)_{10} = (1A2)_{16}$$

I.3.6 Conversion binaire – décimal

On peut facilement convertir un nombre binaire (*base 2*) vers décimal (*base 10*) en utilisant l'écriture polynomiale décrite précédemment. Par exemple soit $(1011)_2$ à convertir en décimal:

$$
\begin{array}{ccccccccc}
1 & & 0 & & 1 & & 1 & & \\
\times & & \times & & \times & & \times & & \\
2^3 & & 2^2 & & 2^1 & & 2^0 & & \\
\downarrow & & \downarrow & & \downarrow & & \downarrow & & \\
1 \times 2^3 & + & 0 \times 2^2 & + & 1 \times 2^1 & + & 1 \times 2^0 & & \\
\parallel & & \parallel & & \parallel & & \parallel & & \\
8 & + & 0 & + & 2 & + & 1 & & = (11)_{10}
\end{array}
$$

Le résultat est:

$$(1011)_2 = (11)_{10}$$

I.3.7 Conversion octal – décimal

Soit $(135)_8$ un nombre octal (*base 8*) à convertir en décimal (*base 10*). En utilisant l'écriture polynomiale décrite précédemment:

$$
\begin{array}{ccccccc}
1 & & 0 & & 5 & & \\
\times & & \times & & \times & & \\
8^2 & & 8^1 & & 8^0 & & \\
\downarrow & & \downarrow & & \downarrow & & \\
1 \times 8^2 & + & 0 \times 8^1 & + & 5 \times 8^0 & & \\
\parallel & & \parallel & & \parallel & & \\
64 & + & 0 & + & 5 & & = (69)_{10}
\end{array}
$$

Le résultat est:

$$(135)_8 = (69)_{10}$$

I.3.8 Conversion hexadécimal – décimal

Soit $(\mathbf{1AF})_8$ un nombre hexadécimal (*base* ***16***) à convertir en décimal (*base* ***10***). En utilisant l'écriture polynomiale décrite précédemment:

$$\begin{array}{ccccccc} 1 & & A & & F & & \\ \times & & \times & & \times & & \\ 16^{2} & & 16^{1} & & 16^{0} & & \\ \downarrow & & \downarrow & & \downarrow & & \\ 1 \times 16^{2} & + & 10 \times 16^{1} & + & 15 \times 16^{0} & & \\ \parallel & & \parallel & & \parallel & & \\ 256 & + & 160 & + & 15 & & = (431)_{10} \end{array}$$

Le résultat est:

$$(1AF)_8 = (431)_{10}$$

I.3.9 Conversion binaire – octal et vice versa

a) Conversion binaire – octal

Pour convertir un nombre binaire (*base* **2**) en octal (*base* ***8***), il suffit de faire des groupes de trois chiffres (***3 bits***) puisque :

$$8 = 2^3$$

Et en commençant depuis la droite vers la gauche. Chaque groupe de trois chiffres binaires est remplacé par un chiffre octal, comme présenté dans le tableau ci-dessous.

octal	binaire
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

Par exemple, convertissons $(\mathbf{1011001101})_2$:

$$\underbrace{001}_{1}\quad \underbrace{011}_{3}\quad \underbrace{001}_{1}\quad \underbrace{101}_{5}$$

Le résultat est:

$$(1011001101)_2 = (1315)_8$$

b) Conversion octal – binaire

Pour convertir un nombre octal (*base* ***8***) en binaire (*base* **2**), il suffit de faire des groupes de trois chiffres (***3 bits***) comme expliqué ci-dessus dans la partie précédente.

Par exemple, convertissons $(\mathbf{1273})_8$:

$$\begin{array}{cccc} \mathbf{1} & \mathbf{2} & \mathbf{7} & \mathbf{3} \\ \downarrow & \downarrow & \downarrow & \downarrow \\ \mathbf{001} & \mathbf{010} & \mathbf{111} & \mathbf{011} \end{array}$$

Le résultat est:

$$(\mathbf{1273})_8 = (\mathbf{001010111011})_2$$

I.3.10 Conversion binaire – hexadécimal et vice versa

c) Conversion binaire – hexadécimal

Pour convertir un nombre binaire (*base* **2**) en hexadécimal (*base* ***16***), il suffit de faire des groupes de quatre chiffres (***4 bits***) puisque :

$$\mathbf{16} = \mathbf{2}^{4}$$

Et en commençant depuis la droite vers la gauche. Chaque groupe de quatre chiffres binaires est remplacé par un chiffre hexadécimal, comme présenté dans le tableau ci-dessous.

octal	binaire
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
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

Par exemple, convertissons (**1011001101**)$_2$:

$$\underbrace{\mathbf{0010}}_{\mathbf{2}}\ \underbrace{\mathbf{1100}}_{\mathbf{C}}\ \underbrace{\mathbf{1101}}_{\mathbf{D}}$$

Le résultat est:

$$(\mathbf{1011001101})_2 = (\mathbf{2CD})_{16}$$

d) Conversion hexadécimal – binaire

Pour convertir un nombre hexadécimal (*base* ***16***) en binaire (*base* **2**), il suffit de faire des groupes de trois chiffres (***4 bits***) comme expliqué ci-dessus dans la partie précédente.

Par exemple, convertissons (**1F7**)$_{16}$:

$$\begin{array}{ccc} \mathbf{1} & \mathbf{F} & \mathbf{7} \\ \downarrow & \downarrow & \downarrow \\ \mathbf{0001} & \mathbf{1111} & \mathbf{0111} \end{array}$$

Le résultat est:

$$(\mathbf{1F7})_{16} = (\mathbf{000111110111})_2$$

I.4 Calculs dans une base B

Les opérations arithmétiques (addition, soustraction, multiplication, division) sont réalisables dans toute ***base B***, avec mêmes règles que pour la ***base décimale*** (*base 10*). Mêmes les retenues également mais dépendant de la base.

Alors, Quand on additionne deux chiffres ***a*** et ***b*** dans la ***base B***. Si la somme des valeurs décimales de ***a*** et ***b*** dépasse ou égale ***B*** alors il y a une retenue.

Par exemple pour l'addition ***binaire*** (*base 2*):

$(\mathbf{0+0})_2 = (\mathbf{0})_2$

$(\mathbf{0+1})_2 = (\mathbf{1})_2$

$(\mathbf{1+0})_2 = (\mathbf{1})_2$

$(\mathbf{1+1})_2 = (\mathbf{10})_2$ (soit ***0*** avec une retenue de ***1***)

$(\mathbf{1+1+1})_2 = (\mathbf{11})_2$ (soit ***1*** avec une retenue de ***1***)

$$\begin{array}{rcc} & \mathbf{1} & \\ & & \mathbf{1} \\ + & & \mathbf{1} \\ + & & \mathbf{1} \\ \hline & \mathbf{1} & \mathbf{1} \end{array} \qquad \begin{array}{rcc} & \mathbf{1} & \\ & & \mathbf{1} \\ + & & \mathbf{1} \\ \hline & \mathbf{1} & \mathbf{0} \end{array} \qquad \begin{array}{rc} & \mathbf{1} \\ + & \mathbf{0} \\ \hline & \mathbf{1} \end{array} \qquad \begin{array}{rc} & \mathbf{0} \\ + & \mathbf{1} \\ \hline & \mathbf{1} \end{array} \qquad \begin{array}{rc} & \mathbf{0} \\ + & \mathbf{0} \\ \hline & \mathbf{0} \end{array}$$