# Chapitre 1

# Codification et représentation des nombres

## Introduction

L'homme a besoin d'un système de codage pour identifier, quantifier, qualifier les objets, les lieux, les évènements…Ces codes lui permettent de mémoriser, traiter et communiquer les informations. Ils peuvent être : le langage, l'écriture, numérations, le graphisme, etc.. Chaque code respecte des règles bien définies.

Pour un ordinateur, le codage de l'information permet d'établir une correspondance qui permet sans ambiguïté de passer d'une représentation d'une information dite externe (une image, un son, un texte..) à une autre représentation de la même information (dite interne: sous forme binaire, n'utilisant que des 0 et des 1) suivant un ensemble de règles précises.
La raison pour laquelle les ordinateurs manipulent des données binaires est liée au fonctionnement de leurs composants physiques. Les transistors et les condensateurs, qui sont les éléments de base d'un ordinateur, possèdent deux états stables : activé/désactivé ou chargé/déchargé. Ainsi, un transistor dans l'état activé va stocker l'information 1 (ou 0 s'il est dans l'état désactivé).
Le terme bit (b minuscule dans les notations) signifie « binary digit », c'est-à-dire 0 ou 1 en numérotation binaire est la plus petite unité d'information manipulable par une machine numérique.

L'octet (en anglais byte ou B majuscule dans les notations) est une unité d'information composée de 8 bits. Il permet par exemple de stocker un caractère comme une lettre ou un chiffre.

Une unité d'information composée de 16 bits est généralement appelée mot (en anglais word). Une unité d'information de 32 bits de longueur est appelée mot double (en anglais double word, d'où l'appellation dword). Beaucoup d'informaticiens ont appris que 1 kilooctet valait 1024 octets.

. Voici les unités standardisées :

•Un kilooctet (ko) = $2^{10}$ octets =1024 octets

•Un mégaoctet (Mo) = $2^{20}$ octets =1024 koctets

•Un gigaoctet (Go) = $2^{30}$ octets = 1024 Moctets

•Un téraoctet (To) = $2^{40}$ octets=1024 Goctets

•Un pétaoctet (Po) = $2^{50}$ octets=1024 Toctets

L'objectif de ce chapitre est de prendre connaissance des notions de base du codage de l'information,

## 1- Les systèmes de numérations

Pour qu'une information numérique soit traitée par un circuit, elle doit être mise sous forme adaptée à celui-ci. Pour cela Il faut choisir un système de numération, de nombreux systèmes de numération sont utilisés en technologie numérique ; tels que les systèmes : Décimal (base 10), Binaire (base 2), Octal (base 8) et Hexadécimal (base 16).

Dans ce qui suit on va étudier les systèmes décimal, binaire, octal et hexadécimal ainsi que les conversions entre ces différentes bases de codage.

### 1-1- Le système décimal

Les nombres que nous utilisons habituellement sont ceux de la base 10 (système décimal). Nous disposons de dix chiffres différents de 0 à 9 pour écrire tous les nombres. D'une manière générale, **toute base N est composée de N chiffre de 0 à N-1.**

Soit un nombre décimal N = 2348. Ce nombre est la somme de 8 unités, 4 dizaines, 3 centaines et 2 milliers. Nous pouvons écrire

$$N = (2 \text{ x } 1000) + (3 \text{ x } 100) + (4 \text{ x } 10) + (8 \text{ x } 1)$$

$$\mathbf{2348 = (2 \times 10^3) + (3 \times 10^2) + (4 \times 10^1) + (8 \times 10^0)}$$

10 représente la base et les puissances de 0 à 3 le rang de chaque chiffre. Quelque soit la base, le chiffre de droite est celui des unités. Celui de gauche est celui qui a le poids le plus élevé. Cette écriture s'appelle **forme polynomiale.**

### 1-2 - Le système binaire

Dans les domaines de l'automatisme, de l'électronique et de l'informatique, nous utilisons la base 2. Tous les nombres s'écrivent avec deux chiffres uniquement (0 et 1). De même que nous utilisons le système décimal parce que nous avons commencé à compter avec nos dix doigts, nous utilisons le binaire car les systèmes technologiques ont souvent deux états stables.

Un interrupteur est ouvert ou fermé, une diode est allumée ou éteinte, une tension est présente ou absente, une surface est réfléchissante ou pas (CD), un champ magnétique est orienté Nord-Sud ou Sud-Nord (disque dur). A chaque état du système technologique, on associe un état logique binaire.

La présence d'une tension sera par exemple notée 1 et l'absence 0. Le chiffre binaire qui peut prendre ces deux états est nommé "Bit" (Binary digit).

Avec un bit nous pouvons coder deux états

| |
|---|
| 0 |
| 1 |

Avec deux bits nous pouvons coder quatre états

| | |
|---|---|
| 0 | 0 |
| 0 | 1 |
| 1 | 0 |
| 1 | 1 |

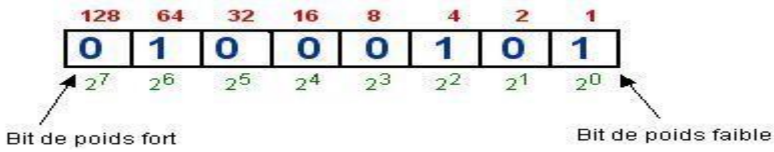
Avec trois bits nous pouvons coder huit états

| | | |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |
| 1 | 1 | 1 |

A chaque nouveau bit, le nombre de combinaisons possibles est doublé. Ce nombre est égal à **2 puissance N** (N étant le nombre de bits).

Un groupe de bits est appelé un mot, un mot de huit bits est nommé un octet (byte). Avec un octet, nous pouvons écrire 2 puissance 8 = **256 nombres binaires de 0 à 255**

Description d'un octet.




Un 1 dans une case représente la valeur décimale qui est au dessus.

**Correspondance entre binaire et décimal.**

<u>Conversion d'un nombre binaire en décimal</u>.

Il suffit de faire la somme des poids de chaque bit à 1, c'est-à-dire écrire le nombre sous forme polynomiale. Le nombre ci dessus est égal à

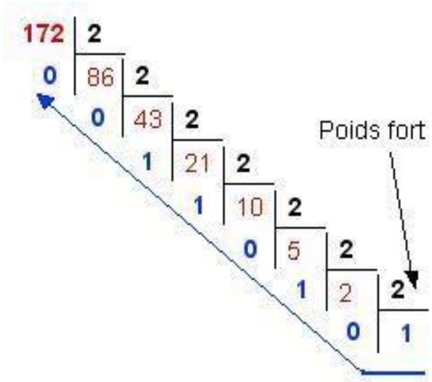
$(01000101)_2 = ( 1\times 2^0 + 1\times 2^2 + 1\times 2^6 )_{10} = (69)_{10}$

$(111011 )_2 = 1\times2^5 + 1\times2^4 + 1\times2^3 +0\times2^2 + 1\times2^1 + 1\times2^0 = (61)_{10}$

$(10011.1101 )_2=1\times2^4+0\times2^3 +0\times2^2 + 1\times2^1 + 1\times2^0 + 1\times2^{-1} +1\times2^{-2} +0\times2^{-3} +1\times2^4 =(19.75)_{10}$

Conversion d'un nombre décimal en binaire (exemple : N = 172).

Méthode par divisions




$172_{(10)} = 10101100_{(2)}$

Conversion d’un nombre décimal à virgule

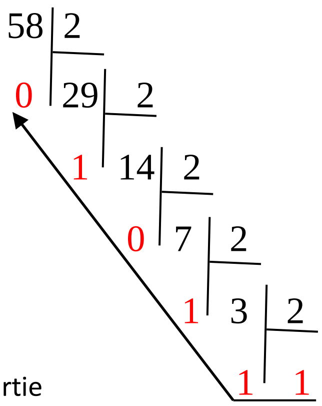
Pour convertir un nombre décimal à virgule dans une base B quelconque, il faut :

1-Convertir la partie entière en effectuant des divisions successives par **2** (comme nous l’avons vu précédemment).

2-Convertir la partie fractionnaire en effectuent des multiplications successives par **2** et en conservant à chaque fois le chiffre devenant entier.

Exemple : Conversion du nombre (58,625) en base 2

**Conversion de la partie entière**




Lecture du résultat de la partie entière

**Conversion de la partie fractionnaire**

$0.625 \times 2 = \mathbf{1}\ .25$

$0.25 \times 2 = 0\ .5$

$0.5 \times 2 = 1\ .0$

Lecture du résultat De la partie fractionnaire

$$\mathbf{(58.625)_{10} = (111010.101)_2}$$

Parfois en multipliant la partie fractionnaire par la base 2 ou n’importe quelle base B on n’arrive pas à convertir toute la partie fractionnaire. Ceci est dû essentiellement au fait que le nombre à convertir n’a pas un équivalent exacte dans la base B et sa partie fractionnaire est cyclique ou infinie.

Exemple : $(0.15)_{10}=(\ ?\ )_2$

$0.15 \times 2 = 0.3$

$0.3 \times 2 = 0.6$

$0.6 \times 2 = 1.2$

$0.2 \times 2 = 0.4$

$0.4 \times 2 = 0.8$

$0.8 \times 2 = 1.6$

$0.6 \times 2 = 1.2$

$0.2 \times 2 = 0.4$

$0.4 \times 2 = 0.8$

$0.8 \times 2 = 1.6$

$(0.15)_{10} = (0.00\underline{\mathbf{1001}}\ \underline{\mathbf{1001}}\ )_2$

On dit que le nombre $(0.15)_{10}$ est cyclique dans la base 2 de période **1001** .

### 1-3- <u>Le système octal (base 8) :</u>

Le système octal ou base 8 comprend huit chiffres qui sont {0, 1, 2, 3, 4, 5, 6, 7}. Les chiffres 8 et 9 n'existent pas dans cette base. Ecrivons à titre d'exemple, les nombres 45278 et 1274.6328 :

$(4527)_8 = 4*8^3 + 5*8^2 + 2*8^1 + 7*8^0 = (2391)_{10}$

$(1274.632)_8 = 1*8^3 + 2*8^2 + 7*8^1 + 4*8^0 + 6*8^{-1} + 3*8^{-2} + 2*8^{-3} = (700.8007813)_{10}$

<u>Conversion d'un nombre décimal en base 8</u>

$(110.625)_{10}=(\ ?\ )_8$

Partie entière

$$\begin{array}{r|l} 110 & 8 \\ \hline 6 & 13 \end{array} \qquad \begin{array}{r|l} 13 & 8 \\ \hline 5 & 1 \end{array}$$

partie fractionnaire

$0.625 \times 8 = 5.0$

$(110.625)_{10} = (156.5)_8$

## 1-4- Le système hexadécimal

La manipulation des nombres écrits en binaire est difficile pour l'être humain et la conversion en décimal n'est pas simple. On utilise aussi très souvent *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).

Pour écrire les nombres en base 16 nous devons disposer de 16 chiffres, pour les dix premiers, nous utilisons les chiffres de la base 10, pour les suivant nous utiliserons des lettres de l'alphabet.

| Décimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Hexadécimal | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |

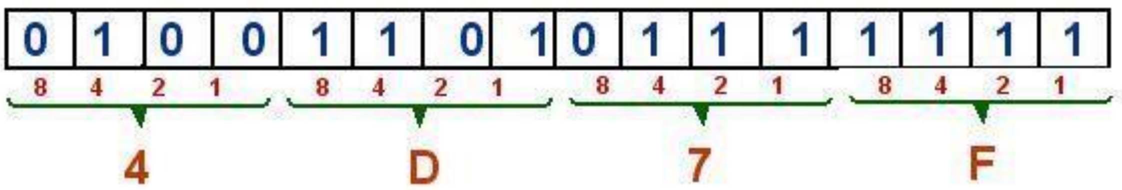
Les règles sont ici aussi les mêmes que pour le décimal.

$$A3F_{(16)} = (A \times 16^2) + (3 \times 16^1) + (F \times 16^0)$$
$$A3F_{(16)} = (10 \times 256) + (3 \times 16) + (15 \times 1)$$
$$A3F_{(16)} = 2560 + 48 + 15 = 2623_{(10)}$$

**Correspondance entre binaire et hexadécimal.**

La conversion du binaire en hexadécimal est très simple, c'est d'ailleurs la raison pour laquelle nous utilisons cette base.

Il suffit de faire correspondre un mot de quatre bits (quartet) à chaque chiffre hexadécimal.

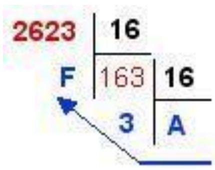



$$4D7F_{(16)} = 0100110101111111_{(2)}$$

Conversion d'un mot de 16 bits entre binaire et hexadécimal

**Correspondance entre décimal et hexadécimal.**

La méthodes par divisions s'applique comme en binaire (exemple : N = 2623).



$2623_{(10)}=A3F_{(16)}$

En général pour faire La conversion d'un nombre d'une base quelconque B1 vers une autre base B2 il faut passer par la base 10. Mais si la base B1 et B2 s'écrivent respectivement sous la forme d'une puissance de 2 on peut passer par la base 2 (binaire)

Base octale (base 8) : $8=2^3$ chaque chiffre octal se convertit tout seul sur 3 bits.

Base hexadécimale (base 16) : $16=2^4$ chaque chiffre hexadécimal se convertit tout seul sur 4 bits.

**Exemples :**

$(6\ 5\ 3\ 0.7)_8 = (110\ 101\ 011\ 000.111)_2$

$(101\ 010\ 100\ 111\ 000)_2 =(5\ 2\ 4\ 7\ 0)_8$

$(1101\ 1000\ 1011\ 0110.011)_2 =(D\ 8\ B\ 6.6)_{16}$

## 1-5- Les nombres signés

Nous avons jusqu'à présent parlé de nombres entiers naturels. Ils ne peuvent par nature qu'être positifs ou nuls. Envisageons maintenant les nombres entiers relatifs ou autrement dit, munis d'un signe '+' ou '-'. En décimal,

+1, +2, +3 etc. sont des nombres positifs. Ils sont supérieurs à 0 ( $n >0$ )

-1, -2, -3 etc. sont des nombres négatifs. Ils sont inférieurs à 0 ( $n < 0$ )

De même en binaire,

+1, +10, +11, +100, +101 etc. sont des nombres binaires positifs,

-1, -10, -11, -100, -101 etc. sont des nombres binaires négatifs.

Le problème est que les circuits électroniques digitaux ne peuvent enregistrer que des 0 ou des 1 mais pas de signes + ou -. Le seul moyen est alors de convenir que si un nombre est susceptible d'être négatif on lui réserver un bit pour indiquer le signe.

Plusieurs méthodes sont utilisées pour représenter les nombres négatifs dans un ordinateur, parmi lesquelles nous citons : la représentation en signe et valeur absolue **(SVA),** le complément à 1 (**CP1)** et le complément à 2 (**CP2)**. On cherche une représentation binaire des entiers négatifs pour que l'addition de deux nombres entiers relatifs fonctionne, c'est la représentation complément à 2 (CP2).

## Comment calculer les codes des nombres négatifs en CP2 sur 8 bits ?

Le calcul se fait en trois étapes :

1°calcul du code binaire du nombre sur 8 bits

2° Calcul du complément à 1 = Remplacer tous les 0 par des 1 et tous les 1 par des 0.

3° Calcul du complément à 2 = Ajouter 1 au complément à 1

Exemples : 1-comment écrire – 4 en CP2 ?

+ 4 = $0000\ 0100_{(2)}$

Le complément à 1 de ce code est CP1(+4)=$1111\ 1011_{(2)}$

Ajoutons 1 à ce code pour obtenir son complément à 2

$1111\ 1011_{(2)} + 1 = 1111\ 1100_{(2)} = (-4)_{CP2}$

2-comment écrire – 24 en CP2 ?

Le nombre +24 = $(11000)_2 = (0001\ \mathbf{1000})_{\mathbf{CP2}}$

Le complément à 1 de ce code est CP1(+24)= 1110 0111

Ajoutons 1 à ce code pour obtenir son complément à 2

$1110\ 0111_{(2)} + 1 = 1110\ 1000_{(2)} =$

$(24)_{CP2}$ ainsi $(-24)_{10} = (1110\ \mathbf{1000})_{CP2}$

Le bit le plus à gauche du code CP2 est celui qui va représenter le signe. Signe négatif si ce bit vaut 1, signe positif quand ce bit vaut 0.

Le plus grand nombre signé sur 8 bits est **+127 (01111111 )**

Le plus petit nombre signé sur 8 bits est **-128 ( 10000000 )**

**-128 à +127 => 256 combinaisons** (2 puissance 8)

Ce qui s’applique sur 8 bits s’applique aussi sur 4, 16, … bits.

## 2- Les opérations Arithmétique dans les différentes bases

### 2-1- Binaire

- **L'addition**

L'algorithme de l'addition des nombres binaires est le même que celui de l'addition des nombres décimaux [3].

Dressons la table d'addition

0 + 0 = 0

0 + 1 = 1

1 + 0 = 1

1 + 1 = 0 et retenue de 1

**Exemple :**

```
      1   1
  10001101
+     1001
 ---------
 =10010110
```

**Remarque** : 1 + 1 + 1 = 1 et retenue de 1.

- **La soustraction**

Dressons la table de soustraction

0 - 0 = 0

1 - 0 = 1

1 - 1 = 0

0 - 1 = 1 avec retenue

Comme dans le cas du système décimal, la soustraction en binaire est également basée sur la notion de retenue

**Exemple**

```
     1 ₁0 1 1 0
-    ₁1 1 0 0
------ ---
=    01 010
```

## ➢ La multiplication

Toujours comme dans le cas du système décimal, chaque nombre de la ligne du bas sera distribué à la ligne du haut. Ensuite, chaque ligne sera sommée (ne pas oublier les retenues) et le résultat sera alors obtenu.

**Exemple**

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | 1 | 0 | 1 | 0 |
| x | | | | 1 | 0 | 1 | 0 | 1 | 0 |
| | | | | | | 0 | 0 | 0 | 0 |
| + | | | | $_1$ | 1 | 0 | 1 | 0 | 0 |
| + | | | | 0 | 0 | 0 | 0 | 0 | 0 |
| + | | $_1$ | 1 | 0 | 1 | 0 | 0 | 0 | 0 |
| + | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| + | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| = | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

## ➢ La division

Pour la division en binaire, c’est le même algorithme classique de la division décimale enseigné dans les classes élémentaires. Elle s'effectue à l'aide de soustractions et de décalages, Les chiffres du quotient, plus simples, sont 0 ou 1 ; 1 lorsque le diviseur est plus petit que le dividende et 0 sinon.

**Exemple**

```
 1 0 1 1      | 100
-1 0 0        |--------
 0 0 1 1      | 10,11
 - 0 0 0      |
     1 1 0    |
    -1 0 0    |
       1 0 0  |
      -1 0 0  |
           0  |
```

## 2-2- Les opérations arithmétiques en Octal :

### ➢ L'addition

Comme pour le système binaire, on applique les mêmes règles pour les nombres octaux. Toutefois, dans ce cas, on aura la retenue **«1»** à gauche à chaque fois que la somme dépasse la valeur **7** car **$7_{(8)}+1_{(8)}=10_{(8)}$.** (Voir le tableau 3)

Pour additionner deux chiffres octaux, on cherche ces chiffres sur le tableau 3**:** un en début de colonne, l'autre en début de ligne ; la somme se trouve dans la case correspondant au croisement entre colonne et ligne [3]. Par exemple, si l'on doit additionner les chiffres octaux **7** et 3**,** on cherche la somme dans la case située au croisement de la colonne **7** et de la ligne **3.** Dans cette case, on trouve **12.** Ainsi, dans le système octal, on aura**: 7 + 5 = 14.**

Exemple d'addition en base octale:

$$\begin{array}{rccc} & {}_{1}\mathbf{1} & {}_{1}\mathbf{2} & \mathbf{3}_{(8)} \\ + & \mathbf{4} & \mathbf{5} & \mathbf{7}_{(8)} \\ \hline & \mathbf{6} & \mathbf{0} & \mathbf{2}_{(8)} \end{array}$$

| ± | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 10 |
| 2 | 2 | 3 | 4 | 5 | 6 | 7 | 10 | 11 |
| 3 | 3 | 4 | 5 | 6 | 7 | 10 | 11 | 12 |
| 4 | 4 | 5 | 6 | 7 | 10 | 11 | 12 | 13 |
| 5 | 5 | 6 | 7 | 10 | 11 | 12 | 13 | 14 |
| 6 | 6 | 7 | 10 | 11 | 12 | 13 | 14 | 15 |
| 7 | 7 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |

**Tableau 3 :** Tableau d'addition et de soustraction dans le système octal

### ➢ La soustraction

Comme en décimal mais en se limitant à 7 (octal). Il faut juste apprendre de nouvelles tables. Pour faire la soustraction de deux chiffres octaux, on cherche ces chiffres sur le tableau 3**:** le diminuende dans la case correspondant au croisement entre colonne et ligne, le diminuteur en début de colonne (ou en début de ligne), le résultat de soustraction se trouve en début de ligne (ou en début de colonne).

Exemple de soustraction octale:

$$\begin{array}{cccc} & \mathbf{7} & {}^{1}\mathbf{2} & {}^{1}\mathbf{3}_{(8)} \\ - & {}_{1}\mathbf{4} & {}_{1}\mathbf{5} & \mathbf{7}_{(8)} \\ \hline & \mathbf{2} & \mathbf{4} & \mathbf{4}_{(8)} \end{array}$$

## 2-3 Les opérations arithmétiques en Hexadécimal :

- **L'addition**

Comme en décimal, l'addition s'effectue chiffre par chiffre. Toutefois, dans ce cas, on aura la retenue **«1»** à gauche à chaque fois que la somme dépasse la valeur **F** car **$F_{16)}+1_{16)}=10_{(16)}$.** (Voir le tableau 4)

Exemple d'addition Hexadécimale:

$$\begin{array}{cccc} & \mathbf{1} & {}^{1}\mathbf{2} & \mathbf{A}_{(16)} \\ + & \mathbf{E} & \mathbf{5} & \mathbf{7}_{(16)} \\ \hline & \mathbf{F} & \mathbf{8} & \mathbf{1}_{(16)} \end{array}$$

- **La soustraction**

En se basant le tableau 4, la soustraction de deux chiffres hexadécimaux, se fait en cherchant ces chiffres sur le tableau**:** le diminuende dans la case correspondant au croisement entre colonne et ligne, le diminuteur en début de colonne (ou en début de ligne), le résultat de soustraction se trouve en début de ligne (ou en début de colonne).

Exemple de soustraction Hexadécimale **:**

$$\begin{array}{cccc} & F & {}^{1}2 & A_{(16)} \\ - & {}_{1}E & 5 & 7_{(16)} \\ \hline & 0 & D & 3_{(16)} \end{array}$$

| ± | 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 | 20 |
| C | C | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
| D | D | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 |
| E | E | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 |
| F | F | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |

**Tableau 4 :** Tableau d'addition et de soustraction dans le système Hexadécimal