Elément de module :

# Architecture des ordinateurs


Enseigné par:

- CHERIF Walid


Année universitaire 2023/2024

# Chapitre 2:

## Systèmes de numération

1 - Bases usuelles

2 - Nombres signés

3 - Opérations binaires

4 - Complément à 1 et complément à 2

5 - Codage des réels

# 1- Bases usuelles

## 1.1- Base binaire :

L'arithmétique binaire est utilisée par les ordinateurs car les deux chiffres 0 et 1 s'y traduisent facilement par la tension ou le passage d'un courant : 0 représentant l'état bas (tension ou courant nul) et 1 l'état haut (tension non nulle, courant qui passe).
Tout nombre peut être représenté par des bits {0,1} :

$$37_{10} = 1*2^5 + 0*2^4 + 0*2^3 + 1*2^2 + 0*2^1 + 1*2^0$$

Ainsi, l'écriture du nombre 37 en binaire est :

$$37_{10} = 100101_2$$

# 1- Bases usuelles

## 1.2- Autres bases :

L’écriture dans les autres bases est similaire ; parmi les bases les plus courantes :

- La base octale : (8) {0, 1, 2, 3, 4, 5, 6, 7}
- La base décimale : (10) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- La base hexadécimale : (16) {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

## Généralisation :

Pour une base $B$ à - $k$ éléments- : $\{a_0\ , a_1\ , a_2, \dots\ a_{k-1}\}$
L’écriture de tout nombre dans cette base prend la forme :

$$N_m N_{m-1} \dots N_1 N_0\ _B$$

avec $N_i \in B$

# 1- Bases usuelles

## 1.3- Conversion :

**De $B$ vers la base décimale :**

On numérote les positions des $N_i$ de droite à gauche –à partir de $0$ -
On multiplie chaque $N_i$ par $k$ à la puissance la position du $N_i$ et on somme :

$$N_m N_{m-1} \dots N_1 N_0 \,_B = N_0 * k^0 + N_1 * k^1 \dots N_m * k^m$$

**De la base décimale vers $B$ :**

On effectue des divisions successives par $k$, jusqu'à obtenir un zéro, et on lie les restes du bat vers le haut :

# 1- Bases usuelles

## 1.4. Exercice :

On définit la Base 7 : {0, 1, 2, 3, 4, 5, 6} $\quad k = 2^m + 1$

Complétez le tableau suivant :

| | | | | | |
|---|---|---|---|---|---|
| Décimale | 121 | | | | |
| Binaire | | 1101011 | | | |
| Octale | | | | | |
| Hexadécimale | | | B7C | | |
| Base 7 | | | | 603 | |
| Base k | | | | | (k-1) 6 (k-3) 2 |

# 2- Nombres signés

Il suffit d'affecter un bit pour le signe et d'attribuer par convention la valeur 0 au signe + et la valeur 1 au signe -.

Ainsi le nombre +32 s'écrira dans le système binaire : 0100000

Et le nombre -32 s'écrit: 1100000

Exemple :

Le tableau suivant donne un aperçu des différentes représentations pour un nombre compris entre -128 et + 127.

| S | $2^6$ | $2^5$ | $2^4$ | $2^3$ | $2^2$ | $2^1$ | $2^0$ | S | valeur absolue | S | complément à 1 | S | complément à 2 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | - | 127 | - | 0 | - | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | - | 126 | - | 1 | - | 2 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | - | 125 | - | 2 | - | 3 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | - | 1 | - | 126 | - | 127 |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | - | 0 | - | 127 | - | 128 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | + | 127 | + | 127 | + | 127 |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | + | 126 | + | 126 | + | 126 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | + | 1 | + | 1 | + | 1 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | + | 0 | + | 0 | + | 0 |
| Posibilité de représentation | | | | | | | | | de – 127 à + 127 | | de – 127 à + 127 | | de – 128 à + 127 |

# 3- Opérations binaires

Réaliser les opérations binaires suivantes

1001000 + 1011001 Que se passe-t-il si on ne dispose que de 7 bits ?

101101 – 10100

1100000001 x 1011

1100101010/1001

101010/101

# 4- Complément à 1 et Complément à 2

Représentation du complément vrai = complément à 2
Représentation du complément restreint

```
   1111
-  1001
   ----
   0110     => complément restreint
+     1
   ----
   0111     => complément vrai
```

il n'est même pas nécessaire d'exécuter une opération de soustraction pour obtenir ce complément restreint, on s'aperçoit qu'il suffit de transformer tous les 1 en 0 et vice versa pour l'obtenir. $100110_{(2)}$ à pour complément restreint : 011001

Le complément vrai d'un nombre est la valeur qu'il faut ajouter à ce nombre pour obtenir la valeur maximale + 1 que l'on peut exprimer
On peut aussi obtenir le complément vrai d'un nombre en calculant d'abord son complément restreint et en ajoutant ensuite 1.

# 4- Complément à 1 et Complément à 2

## Récapitulons:

On part du bit de poids le plus faible (bit de droite) :
===> si c'est un zéro, on recopie 0 jusqu'au premier 1 rencontré,
===> si c' est un "1", on garde ce premier 1.

Ensuite on inverse tous les bits après le premier 1 rencontré à partir de la droite.
Attention si le bit le plus à droite est un 1, c'est aussi le premier 1 rencontré !

# 4- Complément à 1 et Complément à 2

Exemple 1 :

(42)10 = (101010)2 ===> le bit le plus à droite est un 0

0 ==> 0   on conserve le zéro
1 ==> 1   premier 1 rencontré est conservé
0 ==> 1   inversion des bits après le premier 1 rencontré
1 ==> 0
0 ==> 1
1 ==> 0

Le nombre (42)10 = (101010)2 s'écrit en complément vrai : 010110 .

# 4- Complément à 1 et Complément à 2

En utilisant la méthode du complément restreint + 1 :

| | | | |
|---|---|---|---|
| Valeur de départ | | 101010 | $(42)_{10}$ |
| Calcul complément restreint | => | 010101 | on inverse tous les bits |
| | + | 1 | on ajoute 1 |
| Soit un complément vrai | => | 010110 | |

# 4- Complément à 1 et Complément à 2

Exemple 2 :

(59)10 = (111011)2 ===> le bit le plus à droite est un 1

1 ==> 1  premier 1 rencontré est conservé
1 ==> 0  inversion des bits après le premier 1 rencontré
0 ==> 1
1 ==> 0
1 ==> 0
1 ==> 0

Le nombre (59)10 = (111011)2 s'écrit en complément vrai : 000101 .

# 4- Complément à 1 et Complément à 2

En utilisant la méthode du complément restreint + 1 :

| | | | |
|---|---|---|---|
| Valeur de départ | | 111011 | $(59)_{10}$ |
| Calcul complément restreint | => | 000100 | on inverse tous les bits |
| | + | 1 | on ajoute 1 |
| Soit un complément vrai | => | 000101 | |

Complément vrai = complément restreint + 1

# 4- Complément à 1 et Complément à 2

## Opération en complément à 2

0111 codage de 7 en binaire
1000 complément à un
1001 on ajoute 1 : représentation de -7 en complément à deux.

| | | |
|---|---|---|
| -7 | | 1001 |
| +9 | | 1001 |
| __ | | ____ |
| 2 | (1) | 0010 (on 'ignore' la retenue) |

# 4- Complément à 1 et Complément à 2

## Exercice

Compléter le tableau suivant:

| | 0000 0000 | 0000 0010 | 1011 0001 | 0001 1110 |
|---|---|---|---|---|
| Signé sur 8bits | | | | |
| Compl. à 2 sur 8bits | | | | |

Donnez les représentations binaires de: +77, -77, +15 et -15 sur 8bits suivant les 2 conventions.

# 5- Codage des réels

## IEEE 754 (32 bits)

Etapes :

*1^er bit donne le signe,*
*Ecrire le nombre en binaire,*
*Décaler la virgule jusqu'au dernier 1,*
*1,M* $\times 2^c$
*Compléter par des zéros à droite.*

Sens inverse :

*Ecrire sous la forme: 1,M* $\times 2^c$*,*
*Décaler la virgule de c positions,*
*Convertir.*

Exercice : -10,125

- *Signe négatif: 1^er bit = 1.*
- *Binaire: 1010,001*
- *Décalage: 1,010001* $\times 2^3$
- *M= 010001*
- *E= 3+127=130=*$10000010_2$

*c/c: 1 10000010 010001 0000…*

# 5- Codage des réels

## IEEE 754 (64 bits)

Principe :

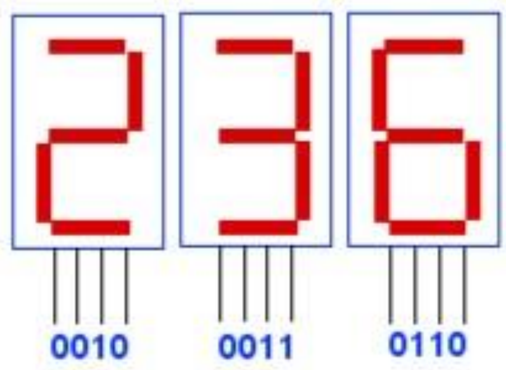
*1$^{er}$ bit donne le signe,*
*11 bits suivants: la mantisse,*
*Les 52 derniers bits: l'exposant.*

Ecrire sous cette norme:

129,6375

# Le décimal codé binaire

Ce codage est destiné à l'affichage de valeurs décimales, chaque digit doit être codé en binaire sur 4
bits (unités, dizaines, centaines ...).
Ce codage ne permet aucun calcul, il est uniquement destiné à la saisie et à l'affichage de données



# Le codage ASCII

Le binaire permet de coder les nombres que les systèmes informatiques peuvent manipuler.
Cependant, l'ordinateur doit aussi utiliser des caractères alphanumériques pour mémoriser et
transmettre des textes. Pour coder ces caractères, on associe à chacun d'entre eux un code binaire,
c'est le codage ASCII (American Standard Code for Information Interchange).
Le caractère A par exemple à pour code 65 soit : 01000001 en binaire.

| | |
|---|---|
| Le caractère f : | code 102 |
| le point d'interrogation ? : | code 63 |
| Le chiffre 2 : | code 50 |