Responsable du module CRI

Dr. Bahri M.R

Chapitre I

Codification et représentation des nombres

1- Introduction

2- Les entiers positifs

2-1 Représentation des entiers positifs dans une base B
(Transformation de bases).

2-2Les bases classiques

2-3Arithmétique

3- Les entiers négatifs

3-1 Représentation des nombres entiers négatifs en **SVA**

3-2 Représentation des nombres entiers négatifs en **CP1**

3-3 Représentation des nombres entiers négatifs en **CP2**

3-4 Arithmétique

4- Les nombres Réels

4-1Représentation des nombres réels en virgule fixe

4-2 Représentation des nombres réels en virgule flottante

4-3 Arithmétique

1- Introduction :

L'informatique se trouve de nos jours dans toute activité humaine. L'ordinateur, se trouve alors comme l'outil le plus répondu au monde. Pour comprendre comment fonctionne un ordinateur, il faut d'abord comprendre comment il représente et traite l'information de manière générale. Dans ce chapitre nous allons focaliser sur la codification et la représentation des nombres.

La machine travaille uniquement avec deux valeurs binaires (0 et 1), pour la manipulation de l'information et notamment des nombres.

2- Les entiers positifs :

L'ordinateur calcule en binaire, seuls les symboles 0 et 1 sont utilisés pour représenter et manipule les nombres. Par ailleurs la taille des nombres manipuler étant limitée, les cases de stockage ont une taille limitée ceci limite donc les capacités de l'ordinateur à exprimer les nombres.

2-1 Représentation des entiers positifs dans une base B

De même qu'en décimal on utilise dix chiffres pour représenter les nombres, dans une base B (B est un entier et B ≥ 2) on utilisé B chiffres pour la représentation des nombres :

Soit une base B dont l'ensemble des chiffres est : E=$\{X_0, X_1, X_3, \dots\dots\dots\dots\dots X_{n-1}\}$

Chaque nombre X exprimé en base B sera représenté par une concaténation des symboles X_i de l'ensemble E.

Exemples :

1- en base 10 $\boldsymbol{X = 19372}$ E=$\{0,1,3,4,5,6,7,8,9\}$
le nombre X est représenté par les chiffres **1,9,3,7** et **2.**

2- en base 2 $\boldsymbol{X = 10111}$ E=$\{0,1\}$

le nombre X est représenté par les chiffres **1** et **0.**

NB. Pour ne pas avoir une ambigüité dans la représentation des nombres il faut toujours préciser la base utilisée pour la représentation du nombre.

Exemple :

$X = 101$ en base 2 est différent de $X = 101$ en base 10.

Donc, $X = 101$ en base 2 sera exprimé **$X=(101)_2$**

a- Représentation d'un entier positif exprimé en base 10 dans une base B

Soit X un nombre entier positif et soit B une base (**B**ϵ **N**et **B ≥ 2).**
Pour obtenir la représentation de X dans la base B on applique le principe de la division euclidienne : **X = q.B + r**.

- on divisé à chaque fois le **q** par **B** et on note le reste**r.**
- La division et répétée jusqu'à ce que le**q**soit inférieureà**B.**
- Les restesobtenusrespectivement dans l'opération de la division.
  Sont notés de droite à gauche, dans l'expression du nombre.

Exemple

Soit le nombre entier X= 43 exprimer en base 10 ,
Pour exprimer X en base 3 on réalise la division successive par 3 :

44 = 3 * 14 + **2**
14 = 3 * 4 + **2**
4 = 3 * 1 + **1**

Le dernier reste est 1, 1 est inférieur à 3, donc on arrête le processus de la division.
Ainsi le nombre décimal 44 exprimer en base 10 sera exprimé **(122)** en base 3

b- Valeur décimale d'un nombre exprimer en base B

La valeur décimale d'un nombre X exprimer en base B et obtenue au multipliant chaque chiffre de la base B,de droite à gauche dans la représentation de X, par une puissance croissante (commençant par zéro) de la base.
Les valeurs ainsi obtenues seront additionnées :

Exemples

1- Soit le nombre X exprimer enbase 7**X=(3425)$_7$**
La valeur décimale de x est :

$$\boldsymbol{X = 5.7^0 + 2.7^1 + 4.7^2 + 37^3}, \text{ d'où} X = 1244 \quad en\ base\ \mathbf{10}$$

2- Soit le nombre X exprimer enbase 2**X=(1101)$_2$**
La valeur décimale de x est :

$$\boldsymbol{X = 1.2^0 + 0.2^1 + 1.2^2 + 1.2^3}\ , \text{ d'où } X = 13 \quad en\ base\ \mathbf{10}$$

2.2 Les bases classiques :

L'informatique généralement se sert de quelques bases classiques notamment : la base **2(binaire**), La base **8** (**octal**), la base **10** (**décimal**) et la base **16** (**Hexadécimale**)

Dans cette section nous présentons la conversion des nombres binaires en octales et en hexadécimal car ceci joue un rôle prépondérant dans les systèmes digitaux.

Sachant que chaque symbole binaire occupe un bit(0 ou 1), un nombre en binaire et donc une collection de bits.

Sachant aussi que le nombre 8 est une puissance 3 de 2 et que le nombre 16 et une puissance 4 de 2.
Donc, chaque chiffre octal correspond à 3 bits et chaque chiffre hexadécimal à 4 bits.

3-1 Conversion binaire Octal

La conversion du binaire en octal se fait par un groupement de 3 bits de droite à gauche dans la représentation binaire du nombre, chaque groupement sera remplacé par le chiffre équivalent en base octale.

Binaire	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
octale	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**

Exemples :

Soit le nombre entier X exprimer en binaire $\mathbf{X=(1101001)_2}$
La conversion de X en octal se fait par groupement de trois bits de droite à gauche, de la manière suivante :

X = **001101 001** donc $\mathbf{x=(151)_8}$

1 5 1

NB : les bits manquants sont remplis par des zéros.

3-2 Conversion binaire hexadécimal

La conversion du binaire en hexadécimal se fait par un groupement de 4 bits de droite à gauche, chaque collection sera remplacée par le chiffre équivalent en base hexadécimal.

Binaire	0000	0001	0010	0011	0100	0101	0110	0111
Hexadécimale	0	1	2	3	4	5	6	7
Binaire	1000	1001	1010	1011	1100	1101	1110	1111
Hexadécimale	8	9	A	B	C	D	E	F

NB : les chiffres de la base 16sont :$\{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$

Exemple :

Soit le nombre entier X exprimer en binaire **$X=(1101111011)_2$**

La conversion de X en hexadécimal se fait par groupement de quatre bits de droite à gauche, de la manière suivante :

X=<u>001101111011</u> donc**$X= (37B)_{16}$**

3 7 B

NB : les bits manquants à gauche sont remplis par des zéros.

Remarque : La conversion directe d'une Base **B** vers une Base **K** est possible si l'une des deux base est une puissanceentière de l'autre (ex : **B=3, K= 9**)

2-3 Arithmétique

Les opérations arithmétiques classiques,pour le autres bases, se réalisent de la même manière qu'en décimal. Dans la suite de ce cours nous exposons les opérations arithmétiques binaires à titre d'exemples illustratifs. Les détails seront exposés en travaux dirigés.

a- L'addition binaire :

L'addition de deux bits binaire se réalise selon la spécification suivante :

bit1		bit2		résultat	retenue
0	+	0	=	0	0
0	+	1	=	1	0
1	+	0	=	1	0
1	+	1	=	0	1

L'addition binaire de deux nombres s'effectue bit à bit de droite à gauche, en reportant les retenues

Exemple :

Soit les deux nombres **$X = (1101101)_2$**et **$y=(10101011)_2$**

L'opération d'addition se réalise de la mêmemanière que dans la base 10

$^{1}0^{1}11^{1}0^{1}1^{1}1^{1}01$
\+ **$X+Y = (100011000)_2$**
10101011

100011000

b- la soustraction binaire :

La soustraction de deux bits binaire se réalise selon la spécification suivante :

bit1		bit2		résultat		retenue
0	-	0	=	0	;	0
0	-	1	=	1	;	1
1	-	0	=	1	;	0
1	-	1	=	0	;	0

La soustraction binaire de deux nombres s'effectue bit à bit de droite à gauche, en reportant les retenues

NB : plus de détaille sera effectué en travaux dirigés.

c- La multiplication binaire :

La multiplication de deux bits binaire se réalise selon la spécification suivante :

bit1		**bit2**		**résultat**
0	x	0	=	0
0	x	1	=	0
1	x	0	=	0
1	x	1	=	1

Exemple :

Soit les deux nombres **$X=(110\ 10)_2$**et **$y= (1010)_2$**

L'opération de multiplication **X * Y**se réalise de la même manière que dans la base 10

```
      11010
x 1010
  ____________
 =00000
+11010.
+00000..
+11010  . . .
_____________
```

=100000100

NB :

a- Le produit de deux nombres positifs est toujours plus grand que les nombres multipliées, ceci pose un problème pour la représentation interne des nombres (taille et capacité).

b- La division binaire est une opération assez complexe généralement on utilise le même principe que le décimal (division par soustraction).

c- **Le développement d'arithmétique dans les différentes bases sera exposé dans les séances des travaux dirigés.**

3- Les entiers négatifs

Dans le calcul binaire les nombres entiers positifs sont représentés uniquement en 0 et 1. Le signe négatif précédant les valeurs entières pose un problème de représentation sur machine.

Plusieurs méthodes sont utilisées pour représenter les nombres négatifs dans un ordinateur,parmilesquelles nous citons : la représentation en signe et valeur absolue **(SVA),** le complément à 1(**CP1)**et le complément à 2 (**CP2)**.

3-1 Représentation des nombres entiers négatifs en SVA (Signe et valeur absolue)

a- Principe

- Les nombres entiers en SVA seront codés en binaire par **±**valeur absolue, le bit le plus fort du nombre représenté, sera réservé pour le signe (**1** pour le signe '**-**' et **0** pour le signe '**+**').
- Un nombre entier codé sur n bits aura une valeur absolue **V** :

$$\mathbf{0} \leq \boldsymbol{V} \leq \mathbf{2}^{\boldsymbol{n}-\mathbf{1}} - \mathbf{1}.$$

b- Propriété

Soit un nombre entier X codé en SVA sur n bits, la plage de la représentation de X est :– $(\mathbf{2^{n-1}-1}) \leq \boldsymbol{X} \leq +(\mathbf{2^{n-1}-1})$.

Sachant que sur **n** bits nous avons $\mathbf{2^n}$ représentation, et de **0** à $\mathbf{2^{n-1}-1}$ nous avons $\mathbf{2^{n-1}}$ nombre.Donc, le **zéro** compte 2 fois **(+0)** et **(-0)**.
Le**zéro** possède alors deux représentations : (-(2^{n-1}-1),……-1,-0,+0,1,2,…,(2^{n-1}-1)).

Exemple1 :

Sur **4** bits nous aurons **16** nombres entiers :

(-7,-6,-5,-4,-3,-2,-1,-0,+1,+1,+2,+3,+4,+5,+6,+7).

NB : pour un nombre codé mot sur n bits nous aurons réellement 2^n-1 nombredifférents (le zéro occupe deux codes)

Exemple2 :

Le nombre entier **X=-20** sera codé en SVA sur **6 bits** minimum comme suit :
20 = $(10100)_2$
X sera codé par : 110100.

Exemple3 :

Le nombre entier **X=+ 20** sera codé en SVA sur **6 bits** comme suit :
20 = $(10100)_2$
X sera codé par :010100.

Exemple3:

Le nombre entier **X=-20** sera codé en SVA sur **8 bits** comme suit :
20 = $(10100)_2$
X sera codé par : 10010100.

c- Arithmétique en SVA

En **SVA**, les opérations d'addition et de soustraction sont compliquées car le bit de signe doit être traité à part.Cette méthode n'est pas utilisée par les constructeurs de calculateur pour la réalisation des opérations d'addition et de soustraction. Les applications d'arithmétique SVA serons plus détaillée en travaux dirigées.

3-2 Représentation des nombres entiers négatifs en complément à 1 (CP1)

a- Principe :

Un nombre entier négatif est obtenu en complément à 1(complément logique) par remplacement de chaque bit à 0 par 1 et Vic versa, dans le code binaire du nombre positif.Le bit le plus fort dans le code du nombre est réservé pour le signe (**1** pour le signe '**-**' et **0** pour le signe '**+**').

Exemple :

Soit le nombre entier **X = - 5** codé sur 4bites ;

Nous avons : +5 =**$(0101)_2$**en binaire (le bit le plus fort est un bit de signe, 0 pour le signe positif).

Le **CP1(+5)= - 5**, donc **X = 1010** en**CP1.**

b- Propriétés :

- En complément à 1 ,le**CP1(X)** et le symétrique de **(X).**
- Donc, les nombres entiers codés sur**n** bits en CP1seront aux nombres de **2^{n-1}**nombre positif et **2^{n-1}** nombre négatif.
- la plage de la représentation d'un entier X codé sur n bits est :– $(\mathbf{2^{n-1}-1}) \leq \boldsymbol{X} \leq +(\mathbf{2^{n-1}-1})$.
- Sachant que sur **n** bits nous avons **2^n** représentation, et de **0** à **2^{n-1}-1** nous avons **2^{n-1}** nombre. Donc, le **zéro** compte 2 fois **(+0)** et **(-0)**.

NB : la représentation en CP1 est facile à réaliser électroniquement.

c- ARITHMETIQUE

- **L'Addition en CP1 :**

Principe :

- dans l'addition une retenue générée par le bit de signe doit être ajoutée au résultat obtenue car on effectue l'addition des compléments y compris le bit de signe (report de la retenue). Une condition nécessaire pour ajouter la retenue, est que une retenue doit être aussi générer par le bit juste avant signe.
- Si nue retenue est générée exclusivement par les bits de signe ou par le bit juste avant signe, l'addition est impossible a réalisée. une erreur de dépassement de capacités est déclarée.

Exemple1 : soit les deux entiers **X= 7** et **Y = 6**

On veut réaliser L'opération **X-Y** sur **4 bits** en **CP1**

7-6 = (7)+(-6) donc, 7- 6 = 7 + CP1(6).

7= $(0111)_2$ et **6= $(0110)_2$ avec CP1(6) = 1001**

Alors, $^{1}\,0^{1}\,1^{1}\,1\,1$

\+

$1\;0\;0\;1$

= **1** 0 0 0 0 une retenue est générée par le bit de signe et le bit avant signe

= 0 0 0 1 le résultat est correcte 7 – 6 = 1

Exemple2 : soit les deux entiers **X= 7** et **Y = 6**
On veut réaliser L'opération **X+Y** sur **4 bits** en **CP1**
7= $(0111)_2$ et **6= $(0110)_2$**

Alors, $^{1}\,0^{1}\,1^{1}\,1\,1$

\+

$0\;1\;1\;0$

= 1 1 0 1 **Résultat incorrecte**

Une retenue est générée uniquement par le bit avant signe.
L'addition est incorrecte (7 + 6 = 13, le nombre signé 13 nécessite 5 bits pour la codification. Un dépassement de capacités doit êtresignalé dans ce cas).

Remarque : En CP1 la soustraction d'un nombre se réduit à l'addition de son complément à 1.(le circuit de soustraction et inutile). En plus le CP1 ne traite pas le bit de signe à part.

3-3Représentation en complément à 2(CP_2)

a- Principe :

- Le complément à 2 (complément arithmétique) d'un nombre entier négatif est obtenu en rajoutant +1 à sa représentation CP1.
- Cette représentation donne une configuration de plus car le zéro aura une seul représentation.

Exemple :

Soit le nombre entier **X = - 5** codé sur **4**bits ;
Nous avons : **X = 1010** en **CP1.**
Le **CP2(+5)= CP1(+5) +1, donc X = 1011 en CP2.**

b-propriétés

- la plage de la représentation d'un entier X codé sur n bits en CP2 est : $-(\mathbf{2}^{n-1}) \leq \boldsymbol{X} \leq +(\mathbf{2}^{n-1}-\mathbf{1})$
- sur **n**bits on a : **X+CP2(X)=2^{n}** (d'où le nom complément à 2)
- la valeur d'un nombre représenter en CP2 revient à lire un nombre binaire $X_{n-1}X_{n-2}\ldots\ldots X_0$de la façon suivante :

$$\boldsymbol{X = X_{n-1}2^{n-1} + \cdots 2^{'}X_1 + 2^{\circ}X_0 - X_{n-1}2^{n-1}}$$

Donc,$\boldsymbol{X} = (\sum_{i=0}^{n-2} \boldsymbol{X_i 2^i})$ **-** $\boldsymbol{X_{n-1}2^{n-1}}$

- Le CP2 (CP2(X)) = X pour toute valeur entière X.

c-ARITHMETIQUE

- **L'Addition en CP2 :**

Principe :

- dans l'addition une retenue générée par le bit de signe sera ignorée. Une condition nécessaire pour accepter le résultat ainsi obtenu, est qu'une retenue doit être aussi générer par le bit juste avant signe.
- Si nue retenue est générée exclusivement par les bits de signe ou par le bit juste avant signe, l'addition est impossible a réalisée. une erreur de dépassement de capacités est déclarée.

Exemple1 : soit les deux entiers **X= 7** et **Y = 6**
On veut réaliser L'opération **X-Y** sur **4 bits** en **CP2**
7- 6 = (7)+(-6) donc, 7- 6 = 7 + CP2(6).
7= (0111)$_2$ et **6= (0110)$_2$ avec CP1(6) = 1001** donc**CP2(6) = 1010**

Alors,
$$\begin{array}{r} {}^{1}\,0^{1}\,1\,1\,1 \\ + \\ 1\,0\,1\,0 \\ \hline = \quad \mathbf{1}\,\mathbf{0}\,0\,0\,1 \end{array}$$

Une retenue est générée par le bit designe et le bit avant signe. Elle sera ignorée

Le résultat est correcte (7 – 6 = 1)

Exemple2 : soit les deux entiers **X= 7** et **Y = 6**

On veut réaliser L'opération **X+Y** sur **4 bits** en **CP2**

7= $(0111)_2$ et **6= $(0110)_2$**

Alors, $^{1}\,0^{1}\,1\,1\,1$

\+

0 1 1 0

———————————

= 1 1 0 1 **Résultat incorrecte**

Une retenue est générée uniquement par le bit avant signe.

L'addition est incorrecte (7 + 6 = 13, le nombre signé 13 nécessite 5 bits pour la codification. Un dépassement de capacités doit être signalé dans ce cas).

Remarque :

- soustraire un nombre reviens à additionner son opposé c.-à-d son complément à 2
- la multiplication de deux nombres revient à multiplier les valeurs absolus des 2 nombres et de prendre l'opposé du résultat si les nombres sont de signe différent
- pour diviser, il faut passer par la division des valeurs absolues et repositionner les signes du quotidien est du reste si nécessaires

4 Les Nombres réels

L'ordinateur manipule uniquement que des bits, il faut donc définir une représentation des nombres décimaux adaptée aux calculs. En plus l'espace de stockage d'un nombre sur une machine est limité, ce qui induit à une limite de la précision des calculs.

L'idée de base pour la représentation des nombres décimaux est de définir l'emplacement d'une virgule séparant la partie entière de la partie décimale et de considérer que les chiffres à droite de la virgule correspondent aux puissances négatives de la base (comme pour les décimaux).

4-1 Représentation des nombres réels en virgule fixe :

L'objectif du calcul fractionnaire est d'avoir le maximum possible de précision dans l'écriture des nombres fractionnaires. L'emplacement de la virgule dans l'écriture du nombre permet de déterminer cette précision. Malheureusement, dans un ordinateur la taille de la représentation est limitée, ainsi la gestion de la position de la virgule pose un réel problème.

a- Conversion base B base décimale

Dans la représentation d'un nombre ou virgule fixe, on choisit une position dans les chiffres du nombre pour séparer la partie fractionnaire à droite de la position choisie, et la partie entière partant de la position choisie vers la gauche.
La conversion de la partie fractionnaire est obtenue par multiplications successives par des puissances négatives et décroissantes la base B utilisée en commençant par (-1).La partie entière est obtenue selon le principe des Nombres entiers.

Ainsi, pour coder un nombre réel sur n positions, on peut par exemple définir une virgule après les K chiffres de poids faible, donc,

Pour X= $(a_{n-K-1}\ldots a_1 a_0 , a_{-1} \ldots a_{1-k}\ a_{-k})_B$

La valeurdécimale de X sera donnée par la formule :$X = \sum_{i=-k}^{n-k-1} \mathrm{ak}2^k$

Exemple :

Soit le nombre réel x codé en binaire en virgule fixe :**$X = (1101 , 01)_2$**
La valeur décimale de x est obtenue comme suit :
$X= (1* 2^{-2}) + (0* 2^{-1}) + (1* 2^0) + (0* 2^1) + (1* 2^2) + (1* 2^3)$
Donc, X= 0.25 + 0 + 1 + 4 + 8.
D'où **X = 13.25.**

b- Conversion base décimale base B

Pour transformer un nombre décimal fractionnaire en base B, la partie entière est transformée selon le même principe des nombres entiers. La partie fractionnaire est obtenue par multiplication successive de la partie fractionnaire par la base B. on note dans chaque opération la partie entièrerésultante de la multiplication. Le processus de multiplication et répété jusqu'à disparition de la partie fractionnaire ou apparition d'unepériode de chiffres.

Exemple :

X=7,125 on veut exprimer X en base binaire :

- Conversion de la partie entière de X en base 2 : **$7=(111)_2$**

- Conversion de la partie fractionnaire de X en base 2 :

0,125 x 2 = 0,25 = 0 + 0,25
0.25 x 2 = 0,5 = 0 + 0,5

0,5 x 2 = 1,0 = 1 + 0,0

$0,125 = (0,001)_2$

Ainsi : **$7,125 = (111,001)_2$**

c- Arithmétique

Les opérationsarithmétiques en virgule fixe dans une base B, se réalise de la même manière qu'en base 10.

NB : Plus de détails seront exposées en travaux dirigés.

4-2 Représentation des nombres Réels en virgule flottante :

La gestion de la position de la virgule fixe étant difficile, en général, on fait recours à la représentation en virgule flottante [FLOATING POINT].

a- Principe :

La représentation à virgule flottante consiste à représenter les nombres réels sous la forme suivante : $\mathbf{X = \pm M.B^E}$,Avec :

B=base (2, 8, 10, 16, …)

M=mantisse (un nombre purement fractionnaire)

E=exposant (un nombre entier)

La mantisse est normalisée (elle comporte le maximum de chiffres significatifs) donc :

$(\mathbf{0,1})_{\mathbf{2}} \leq |\boldsymbol{M}| < (\mathbf{1})_{\mathbf{2}}$ **soit** $(\mathbf{0,5})_{\mathbf{10}} \leq |\boldsymbol{M}| < (\mathbf{1})_{\mathbf{10}}$

Exposant et mantisse doivent pouvoir représenter des nombres positifs et négatifs. Ils pourraient être codés en SVA, CP1 ou CP2, souvent M et sous forme SVA, E est sans signe mais décalé (Biaisé).

La représentation interne d'un nombre réel normalisé sur n bits aura la forme suivante :

SM	E	M

Signe de la maitrise **Exposant** **Mantisse**

b- Cas d'étude : représentation des nombre réels en virgule flottante selon la norme IEEE-754 en simple précision.

Par souci de normalisation de représentation des nombres en vigueur flottante, plusieurs normes ont été proposées.

Dans la suite de ce cours nous abordons la norme IEEE-754 établit par le comité IEEE (Institut of Electrical and Electronics Engénners), en simple précision sur 32 bits.

- **Principe :**

Dans cette norme un nombre flottant X s'écrit sous la forme $\boldsymbol{X = \mp 1, M.2^{E}}$, où :

M : représentelapseudo mantisse car le 1 entier n'est pas représenté. La mantisse est codée sur 23 bits (correspondant aux puissances négatives de 2^{-1} à 2^{-23}).

E :est un exposant codé sur 8 bits, l'exposant est décalé de +127 car il sera représenté sans signe sur les 8 bits(**Ed = E + 127**)

- un bit de signe. Selon le schéma suivant :

S	Ed	M

1 bit de signe 8 bits pour l'exposant 23 bits pour la pseudo mantisse

- **Description** :

1- le bit le plus fort est un bit de signe, représentant le signe de la mantisse (0 pour le signe négatif et 1 pour le signe positif).

2- l'exposant sur **8** bit est décalé de **127**pour représenter les puissances positives et négatives. La plage de représentation de l'exposant est :
$-(\boldsymbol{2^{8-1}}) \leq \boldsymbol{E} \leq +(\boldsymbol{2^{8-1}-1})$**d'où**$-\boldsymbol{127} \leq \boldsymbol{E} \leq +\boldsymbol{127}$

- l'exposant décalé sera représenté sur 8 bits non signé, il sera dans la plage $\boldsymbol{0 \leq Ed \leq 255}$.
- Les valeurs acceptées pour **Ed** sont l'intervalle$\boldsymbol{1 \leq Ed \leq 254}$**.**Les valeurszéro et 255 serons réservées à d'autre fins .
- De même, Les valeurs acceptées pour **E** sont $\boldsymbol{-126 \leq Ed \leq +127.}$ Les valeurs -127 et +128 serons réservées à d'autre fins .

3- La pseudo mantisse et la partie fractionnaire elle occupe les 23 bits les plus faible dans la représentation interne.

Exemple

Représentation du nombre **X= 25,127** en virgule flottante **IEEE-754** en **simple** précision :

1- Mettre X sous Forme $\boldsymbol{X = \mp 1, M.2^E}$

$25 = (11001)_2$, $0.125 = (0,001)_2$

$25,125 = +1,1001001 . 2^4$

2- La pseudo mantisse **$M=(0,1001001)_2$**

3- L'exposant biaiser **Ed = 4+127= 131**

$131 = (10000011)_2$

4- le bit de signe =**0**

Donc sur 32 bit en norme IEEE-754
Le nombre X=25,125 sera représenté en interne par :

0	10000011	10010010000000000000000
S	Ed	M

Remarque

La représentation interne du nombre réel peut être exprimée en hexadécimale, octal, etc…

Dans l'exemple précédant si en procède par groupement de 4 bits de droite à gauche, en suite nous remplacent les groupements par leurs équivalents en binaire, nous obtiendrons la représentation interne hexadécimale du nombre réel :

X= 41C90000

NB : L'arithmétique des réels en virgule flottantes ainsi que des applications plus détaillées de la norme IEEE-754 seront exposées en travaux dirigés.

Remarque : le tableau ci-dessous récapitule les plages des valeurs desnombres réels représentés en normes IEEE-754 simple précision.

Signe	Exposant décalé	Exposant	Mantisse	valeur
0	**0**	**-127**	**0**	**Zéro**
1	**0**	**-127**	**0**	**Zéro**
0	**0**	**-127**	$\neq$ **0**	**Nombre dé normalisé** $X = +0, M.2^{-127}$
1	**0**	**-127**	$\neq$ **0**	**Nombre dé normalisé** $X = -0, M.2^{-127}$
0	$1 \leq Ed \leq 254$	$-126 \leq Ed \leq +127$	**Quelconque**	**Nombre normalisé** $X = +1, M.2^{-E}$
1	$1 \leq Ed \leq 254$	$-126 \leq Ed \leq +127$	**Quelconque**	**Nombre normalisé** $X = -1, M.2^{-E}$
0	**255**	**128**	**0**	$+\infty$
1	**255**	**128**	**0**	$-\infty$
0	**255**	**128**	$\neq$ **0**	**NaN**
1	**255**	**128**	$\neq$ **0**	**NaN**

Annexe : série1 des travaux dirigés

Université Constantine 2 - Abdelhamid Mehri **Octobre 2015**

Faculté des NTIC

Tronc commun MI

Module : CRI

TD1

Systèmes de Numération

Exercice 1 :

1-Exprimer les nombres suivant en base Décimale :

$(472)_8$ $(3132)_4$ $(560)_7$ $(ABDF)_{16}$

2-Exprimer lenombre décimal X= 327 en base 2, 3, 7, 8, et 16.

3-Faire la conversion Binaire/Décimale des nombres suivants :

101, 11101, 111101101, 11111111.

Exercice 2 :

1- Soit le nombre décimal $X= 4a^5 + 2a^3 + a + 5$ tel que a est un entier (a>5).

Exprimer X en base a

2- exprimer les nombres décimaux X, Y, Z en base a (a est un entier/ a>1)

$X= a$, $Y= a^2$, $Z=a^3$

Exercice 3:

Déterminer les couples des entiers (x,y) tel que :

$(x\ y)_7 = (yx)_{10}$

Exercice 4:

1- **Soit le nombre décimal X= 512, exprimer X en base 2, 4, 8 et 16.**
2- **Soit le nombre Y = (11010110101)$_2$**

   Exprimer directement et sans passer par la base 10 le nombre Y en base 4, 8, 16.
3- **Exprimer directement en base 2 et sans passer par la procédure de division les nombres : $X = (1323)_4$, $Y= (3765)_8$, $Z= (AB1F9)_{16}$**

Exercice 5 :

1- **Faire l'addition des nombres binaires suivants : $X=(101101)_2$ et $Y=(110110)_2$**
2- **Réaliser en base 8 l'addition des deux nombres $X=(735)_8$ et $Y=(132)_8$**
3- **Réaliser en hexadécimale l'addition des deux nombres $X=(A1F)_{16}$ et $Y=(9BC)_{16}$**

Exercice 6:

Soit les nombres entiers X et Y tel que : X= 18 et Y= 30.

1- **Exprimer sur six bits en SVA, CP1 et CP2 les nombres entiers X, -X, Y et -Y. Est-ce que cette codification est possible sur Cinque bits ? Justifier.**
2- **Réaliser, si possible, les opérations suivantes en SVA, CP1 et CP2 : X-Y, Y-X, -X-Y.**

Exercice 7:

Soit les nombres entiers : X= (1101011) en CP1 , Y=(11001100) en CP2,

Z= (0100111) en CP1 et T= (0101101) en CP2.

Donner le signe et la valeur décimale de chaque nombre.

Exercice 8 :

1- **Exprimer en binaire les nombres réels suivants : 112,125 ; 237,25 ; 128,75**
2- **Exprimer en décimal les nombres réels suivants : $(111,01101)_2$; $(101,10101)_2$**
3- **Réaliser les opérations arithmétiques suivantes : 112,125 + 237,25 et $(111,01101)_2+(101,10101)_2$**

Exercice 9 :

1- **Donner selon la norme IEEE-754 le code des nombres réels suivants :**

X= 27,25 ; Y= -13,5 ; Z= 0,375

2- **X est un nombre réel codé selon la norme IEEE-754, écrire X sous forme réelle.**

X=(11011000011010110000000000000000)

Annexe : série supplémentaire des travaux dirigés

INTERROGATION ECRITE(2014-2015)

Questions (15 Pts)

1- Soit les nombres entiers $X = (137)_8$ et $Y = (255)_6$

a- Exprimer X et Y en base hexadécimale (16). *(02 pts)*

b- Réaliser les opérations arithmétiques X + Y et X*Y en base hexadécimale.*(02,5 pts)*

2 -Soit les nombres entiers a et n tel que : $a= n^2+ 1$ et $n > 1$.
Exprimer les nombres suivants en base a : $A= n^2+ 2$, $B= (n^2+ 2)^2$ et $C = n(n^2+ 2)$. *(03 pts)*

3- Soit le nombre réel X = 61/8.
Exprimer X Sous la forme $X= (A, B)_2$ ensuite $X=(A, B)_{16}$*(02 pts)*
tel que: A représente la partie entière et B représente la partie fractionnaire.

4- Soit les nombres entiers X= 15 et y = - 32.

a- Coder si possible X et Y en SVA, CP1 et CP2 sur 6 bits *(03 pts)*

b- Réaliser si possible sur 6 bits les opérations : X+Y et X-Y en CP1 et en CP2. *(02,5 pts)*

Contrôle 1 (2014-2015)

Exercice1 :

Soit les nombres réels X , Y , et Z tel que : X = 79,25 , Y= 170 et Z = $(46)_{16}$

1/ Calculer en base 16 le nombre T tel que : T= Y- X (1,5 Pts)

2/ Représenter en SVA, CP1 et CP2 les nombre suivants :

(Z-Y), 2(Z-Y) est 16(Z-Y) (utiliser le nombre de bits nécessaire) (3 Pts)

3/ Nous considérons une machine dans laquelle les nombres réels sont représentés en virgule flottante sur 24 bits selon le format $X= \pm 1, M . 2^E$, et tel que :

- **Le bit le plus fort est réservé pour le signe (1 pour le signe négatif).**
- **Les 7 bits suivants représentent l'exposant E décalé avec la valeur (+ 63).**
- **Les derniers 16 bits sont réservés pour la mantisse M.**

1 bit pour le signe	*7 bits pour l'exposant*	*16 bits pour la mantisse*

a- Donner la représentation interne du nombre réel T selon la norme proposée. (03 pts)

b- Donner la représentation interne de T en Octale (base 8). (01 pts)

c- Donner la représentation interne en hexadécimale pour (- ∞) (01,5 pts)

4/ Simplifier Algébriquement F sous forme de Sop. (03Pts)