**2.3 Représentation binaire des nombres signés**

Les systèmes numériques doivent être capables de traiter des nombres positifs et négatifs. L’utilisation d’une représentation signée suppose l’utilisation d’un **format** (nombre de bits) fixé au préalable.

**2.3.1 Représentation en complément à 2**

Le **complément à 2** est le mode de représentation le plus utilisé en arithmétique binaire et donc dans les ordinateurs pour coder les nombres entiers.Dans cette représentation, les nombres positifs se représentent par leur valeur binaire naturelle.

Par exemple **+6** est représenté par **0000 0110** sur un format de 8 bits.

La représentation des nombres négatifs s’obtient comme suit :

• On part de la représentation binaire naturelle de l’opposé arithmétique du nombre à coder

(nombre positif),

• On calcule son **complément à 1** (CA1) ou **complément restreint**. Celui-ci est obtenu en

Inversant tous ses bits,

• On en déduit son **complément à 2** (CA2) ou **complément vrai** en ajoutant 1 au niveau du LSB.

**Exemple :** Représentation de **-5 en CA2** sur un format de **8 bits**

• Représentation binaire naturelle de **+5 : 5 = 0000 0101,**

**• CA1 de +5 : 5 = 1111 1010,**

**• CA2 de +5 : −5 = 1111 1011.**

**On identifie le CA2 d’un nombre** à son opposé arithmétique car ***A* + (− *A*) = 2*n* = 0** **mod 2*n***

si ***n***est le format de représentation du nombre ***A*.** En effet, soit ***A* = *an*−1K*a*1*a*0**, alors

***A* = *an*−1K *a*1 *a* 0**, et donc :

***A* + *A* = 11K11, soit *A* + *A* = 2*n* −1, et − *A* = *A* +1 .**

La représentation en complément à **2** présente les caractéristiques suivantes :

**•** Le principe d’obtention de l’opposé d’un nombre négatif est le même que celui permettant

d’obtenir l’opposé d’un nombre positif,

**•** Le nombre **0** a une représentation unique,

**•** Un format sur ***n***bits permet de coder en **CA2** les nombres *N* vérifiant

**−2*n*−1 ≤ *N* ≤ +2*n*−1 −1**

**Par exemple, pour *n* = 4,**

***N*(10) *N*(2) *N*(2) −*N*(2) −*N*(10)**

0 0000 1111 0000 0

1 0001 1110 1111 -1

2 0010 1101 1110 -2

3 0011 1100 1101 -3

4 0100 1011 1100 -4

5 0101 1010 1011 -5

6 0110 1001 1010 -6

7 0111 1000 1001 -7

1000 -8

**Tableau 1 :** représentation en complément à 2 sur 4 bits

On peut ainsi représenter des nombres compris entre -4 et +3 sur un format de 4 bits,

entre **-16** et **+15** sur un format de 5 bits, entre **-32 et +31** sur un format de 6 bits, entre **-64 et +63** sur unformat de 7 bits, etc.

• Le bit de poids fort (MSB) est représentatif du bit de signe, mais il est traité comme les autres bits dans les opérations arithmétiques : si **MSB = 0** le nombre est positif, si **MSB = 1** le nombre est négatif.

• Le nombre **+2*n*−1** n’est pas représenté. En effet, dans le cas où ***n* = 4**, le calcul du CA2 de 1000 donne **-(-8) = 0111 + 1 = 1000 = -8.** Ce qui est arithmétiquement incorrect, car 0 est le seul entier à être son propre opposé. On a donc choisi de supprimer la représentation du nombre **+2*n*−1** , un MSB à 1 étant représentatif d’un nombre négatif.

**N. B. Extension d’un nombre codé en CA2**

L’extension d’un nombre codé sur *n* bits à un format sur ***n+k***bits est réalisé comme suit :

• Si le nombre est positif, on complète les *k* bits de poids forts par des **0**. Par exemple,

0110 000110

6 10

( ) (

( )

CA2

Codé sur 4 bits 6 bits

14243 CA2)

• Si le nombre est négatif, on complète les *k* bits de poids forts avec des 1. Par exemple,

1010 111010

6 10

( ) ( )

( )

CA2

Codé sur 4 bits 6 bits

CA2

−

→

14243

.

**2.3.2 Représentation module + signe**

Il s’agit d’une représentation parfois utilisée car plus simple que celle du CA2, mais qui est moins bien adaptée aux opérations arithmétiques. Dans cette représentation, le bit de poids le plus fort représente le signe (**MSB = 0** => nombre positif, **SB = 1** => nombre négatif), et les autres bits la valeur absolue du nombre. Ainsi, un format de *n* bits permet de coder les nombres compris entre **−(2*n*−1 −1) et 2*n*−1 −1**. Dans cette représentation, le zéro possède deux notations possibles.

**Par exemple, pour *n* = 4,**

*N*(10) *N*(2) −*N*(2) −*N*(10)

0 0000 1000 0

1 0001 1001 -1

2 0010 1010 -2

3 0011 1011 -3

4 0100 1100 -4

5 0101 1101 -5

6 0110 1110 -6

7 0111 1111 -7

**Tableau 2** : Représentation "module + signe" sur 4 bits

**Nota Biné : Extension d’un nombre en représentation "module + signe"**

L’extension d’un nombre codé sur ***n***bits à un format sur **n+k**bits consiste à décaler le bit de signe à la position du MSB et à compléter les autres positions par des **0**, que le nombre soit positif ou négatif.

**Par exemple**

0110 000110

6 10

( ) )

( )

M S

Codé sur 4 bits 6 bits

+ → (M+S

14243 {

*S*

, et 1110 100110

6 10

( ) ( )

( )

M S

Codé sur 4 bits 6 bits

+ M S

−

→ +

14243 {

*S*