Activité 4 : Représentation des entiers relatifs

Pour représenter les entiers relatifs en notation binaire, on doit étendre la représentation en binaire naturel aux nombres négatifs. Une solution pourrait être de réserver le bit de poids fort (MSB) pour le signe de l'entier à représenter et d'utiliser les autres pour représenter sa valeur absolue : 0 indiquerait un entier positif et 1 un entier négatif.
Ainsi, avec des mots de 16 bits, en utilisant 1 bit pour le signe et 15 bits pour la valeur absolue, on pourrait représenter les entiers relatifs de

$$1111\ 1111\ 1111\ 1111 = -(2^{15} - 1) = -32\ 767$$

à

$$0111\ 1111\ 1111\ 1111 = 2^{15} - 1 = 32\ 767.$$

Cependant, cette méthode a plusieurs inconvénients, l'un d'eux étant qu'il y a deux zéros, l'un positif et l'autre négatif.
Ensuite, il complique les opérations arithmétiques. Par exemple pour additionner deux entiers relatifs encodés de cette manière, il faudrait faire une addition ou une soustraction selon que les entiers sont du même signe ou non. Ainsi l'addition binaire de 0101 (5_{10}) et 1101 (-5_{10}) donne :

	0	1	0	1	5_{10}
+	1	1	0	1	-5_{10}
	0	0	1	0	2_{10}

Retenue non conservée (résultat sur 4 bits)

Le résultat de cette opération est différent de 0, ce qui complique la mise en œuvre d'opérations.

1. Ecriture des nombres signés

La solution pour résoudre ces problèmes est d'utiliser un encodage particulier nommé **binaire signé en complément à 2**.
Dans cet encodage :

- ➔ le bit de poids fort représente le signe des entiers (0 pour un entier positif et 1 pour un entier négatif),
- ➔ la représentation des nombres positifs ou nul est inchangée
- ➔ les nombres négatifs sont encodés par la méthode du complément à deux.

Si on utilise des mots de 4 bits, on peut représenter les entiers relatifs compris entre - 8 et 7. Les entiers relatif x strictement négatifs sont représentés comme l'entier naturel $x + 2^4 = x + 16$. Ainsi, les entiers naturels de 0 à 7 servent à représenter les entiers relatifs positifs ou nuls et les entiers naturels de 8 à 15 décrivent les entiers relatifs strictement négatifs.

1. **Compléter** la table suivante en déterminant pour chaque quartet sont équivalent décimal dans le cas d'un encodage d'un entier relatif par la méthode du complément à 2.

Nombre binaire MSB			LSB	Valeur en binaire naturel	Valeur en binaire signé en complément à 2
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	**2**
0	0	1	1	3	**3**
0	1	0	0	4	**4**
0	1	0	1	5	**5**
0	1	1	0	6	**6**
0	1	1	1	7	**7**
1	0	0	0	8	**-8**
1	0	0	1	9	**-7**
1	0	1	0	10	**-6**
1	0	1	1	11	**-5**
1	1	0	0	12	**-4**
1	1	0	1	13	**-3**
1	1	1	0	14	**-2**
1	1	1	1	15	-1

2. **Lister** l'intervalle des entiers relatifs codables avec un octet. Même question avec des mots de 32 bits et 64 bits.

	Valeur mini	Valeur maxi
1 octet	$-(2^8 / 2) = -128$	$(2^8 / 2) - 1 = 127$
32 bits	$-(2^{32} / 2) = 2\ 147\ 483\ 648$	$(2^{32} / 2) - 1 = 2\ 147\ 483\ 647$
64 bits	$-(2^{64} / 2)$	$(2^{64} / 2) - 1$

2.Calcul du complément à 2

Le complément à deux d'un mot binaire m de k bits s'obtient en inversant les k bits et en ajoutant 1.

Le complément à deux de 44_{10} = 0101100_2 est donc :

	0	1	0	1	1	0	0
Complémenter ↓	1	0	1	0	0	1	1
Ajouter 1 ↓	1	0	1	0	1	0	0

Le complément à 2 d'un nombre signé transforme un nombre positif en un nombre négatif et vis versa.

Le résultat du complément à 2 de 101100_2 (44_{10}) représente donc le nombre -44_{10} :

$$-44_{10} = 1010100_2$$

3. **Convertir** les entiers naturels 7 et -7 en des nombres binaires signés de 4 bits dans la notation en complément à 2. Même question pour des nombres binaires signés sur 8 bits.

7 → 0111_2 → Complément à 2 : 1000 + 1 = 1001_2

	7	-7
Sur 4 bits	0111	1001
Sur 8 bits	0000 0111	1111 1001

4. **Convertir** en décimal les nombres binaires signés dans la notation en complément à 2 suivants :

➔ 11011001 → 00100110 + 1 = 00100111_2 (39_{10}) donc 11011001 = -39_{10}