Cours_representation_entier_relatif

March 23, 2021

Rappel : un entier relatif est un nombre entier (sans virgule) qui peut être positif ou négatif. (Ne pas croire que les entiers relatifs sont uniquement les entiers négatifs)

1 Le complément à 2^n

Les entiers relatifs s'écrivent en binaire en utilisant un système d'encodage appelé **complément à 2**.

> **IMPORTANT : L'encodage en complément à 2 n'a de sens que si on DEFINIT A L'AVANCE LE NOMBRE DE BITS sur lequel on souhaite écrire ce nombre en binaire**

Dans cet encodage, les entiers relatifs s'écrivent comme les entiers positifs à ceci près que le bit de poids fort (le plus à gauche) a un poids négatif. Bien sûr les nombres écrits en complément à 2 peuvent aussi s'écrire en base hexadécimale.

Exemple : Soient les nombres $1010_{(C2)}$ et $0110_{(C2)}$ écrits **sur 4 bits en complément à 2**

	1	0	1	0
poids	-2^3	2^2	2^1	2^0
poids	-8	4	2	1

On a donc $1010_{(C2)} = -8 + 2 = -6 = A(16)$

	0	1	1	0
poids	-2^3	2^2	2^1	2^0
poids	-8	4	2	1

On a donc $0110_{(C2)} = 4 + 2 = 6 = 6(16)$

2 Avantages du compléments à 2

1. Le MSB (bit le plus à gauche) permet de connaître le signe de l'entier :
   - MSB $= 1$: Entier négatif

- MSB = 0 : Entier positif

2. **Le complément à 2** respecte les opérations arithmétiques classiques (comme si on avait des entiers positifs), ce qui permet d'utiliser les mêmes structures matérielles dans l'**ALU** (voir TD)
3. Il suffit de duppliquer le MSB à gauche autant de fois que nécessaire pour écrire l'entier sur un nombres de bits plus importants.
4. L'entier nul possède une seule représentation en **complément à 2** : "Une suite de 0"
5. L'entier -1 est représenté par "Une suite de 1" en **complément à 2**

3 Le complément à 2 en machine.

Pour calculer l'opposé d'un entier, la machine (c'est-à-dire son **Unité Arithmétique et Logique**) réalise successivement ces 2 opérations basiques :

1. Inverser tous les bits de la représentation binaire de cet entier
2. Ajouter 1 à cet entier

Exemple sur 4 bits:

$6 = 0110_{(C2)} \xrightarrow{inversion} 1001_{(C2)} \xrightarrow{+1} 1010_{(C2)} = -6$

$-6 = 1010_{(C2)} \xrightarrow{inversion} 0101_{(C2)} \xrightarrow{+1} 0110_{(C2)} = 6$

Sur n bits en compléments à 2, une machine peut représenter 2^n nombres dont :

- le minimum = 100...00 = -2^{n-1}
- le maximum = 011...11 = $2^{n-1} - 1$

Exemple sur 4 bits (n =4) :

Complément à 2	décimal
0000	**0**
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	**maximum = 7 =** $2^{4-1} - 1$
1000	**minimum = -8 =** -2^{4-1}
1001	-7
1010	-6
1011	-5
1100	-4
1101	-3
1110	-2
1111	**-1**