Architecture des ordinateurs

Représentation des nombres binaires signés

Nombres avec un signe en binaire

- Il faut préciser le nombre de bits utilisés pour la représentation
- Plusieurs représentations possibles :
  - Utiliser un bit pour indiquer le signe
  - Excédent : Considérer qu'on enlève une constante à chaque nombre
    $\rightarrow$ utilisé pour la représentation des nombres flottants
  - Complément à 2 : Le bit le plus à gauche représente la partie négative
    $\rightarrow$ représentation la plus courante pour les entiers

Bit de signe

- Utiliser le bit de gauche pour le signe :
  1 pour – et 0 pour $+$
- Exemple : Sur 8 bits, on représente
  $+53_{10}$ avec 00110101 et -53_{10} avec 10110101.

- Problème : deux représentations distinctes pour une même valeur
- L'addition et la soustraction sont un peu compliquées :
  signe identique : on additionne les valeurs absolues
  signe différent : on soustrait la plus petite valeur absolue de la plus grande
- La comparaison de deux nombres est aussi compliquée.

Représentation en excédent

- Le nombre n représente la valeur $n - E$ en excédent E
- Le nombre 0 représente la valeur $-E$
- Exemple sur 8 bits en excédent 127 :

valeur binaire	valeur décimale	valeur représentée =
0000 0000	0	$0 - 127 = -127$
0000 0001	1	$1 - 127 = -126$
0111 1101	125	$125 - 127 = -2$
0111 1110	126	$126 - 127 = -1$
0111 1111	127	$127 - 127 = 0$
1000 0000	128	$128 - 127 = 1$
1000 0001	129	$129 - 127 = 2$
1111 1111	255	$255 - 127 = 128$

Représentation en excédent

- Le nombre n représente la valeur $n - E$ en excédent E
- Exemple sur 8 bits en excédent 127 :

valeur binaire	valeur décimale	valeur représentée =
0000 0000	0	$0 - 127 = -127$
0000 0001	1	$1 - 127 = -126$
0111 1101	125	$125 - 127 = -2$
0111 1110	126	$126 - 127 = -1$
0111 1111	127	$127 - 127 = 0$
1000 0000	128	$128 - 127 = 1$
1000 0001	129	$129 - 127 = 2$
1111 1111	255	$255 - 127 = 128$

- En choisissant bien E le bit de poids fort permet de connaitre le signe du nombre
- représentation des flottants

Exercice

- A partir de deux nombres représentés en excédent E, comment obtenir avec le moins d'opérations possible la représentation de leur somme en excédent E ?

Représentation des nombres signés en complément à 2 (1/2)

- On peut additionner les nombres de manière "classique" :
  le résultat aura le bon signe
- Chaque nombre ne possède qu'une représentation

- Pour un nombre $a_{n-1}a_{n-2}\cdots a_0$
- Partie négative a_{n-1} / Partie positive $a_{n-2}\cdots a_0$
- Valeur représentée : $-a_{n-1}2^{n-1} + \sum_{n-1>i\geq 0} a_i 2^i$

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur *n* bits ?

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\rightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\rightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

- Le plus petit nombre négatif représentable sur n bits ?

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\rightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

- Le plus petit nombre négatif représentable sur n bits ?

$\rightarrow$ -2^{n-1} : un bit à 1 à gauche et tous les autres à 0

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\rightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

- Le plus petit nombre négatif représentable sur n bits ?

$\rightarrow$ -2^{n-1} : un bit à 1 à gauche et tous les autres à 0

- Comment représenter 0 ?

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\rightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

- Le plus petit nombre négatif représentable sur n bits ?

$\rightarrow$ -2^{n-1} : un bit à 1 à gauche et tous les autres à 0

- Comment représenter 0 ?

$\rightarrow$ Avec tous les bits à 0

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\rightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

- Le plus petit nombre négatif représentable sur n bits ?

$\rightarrow$ -2^{n-1} : un bit à 1 à gauche et tous les autres à 0

- Comment représenter 0 ?

$\rightarrow$ Avec tous les bits à 0

- Comment représenter -1 sur n bits ?

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\rightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

- Le plus petit nombre négatif représentable sur n bits ?

$\rightarrow$ -2^{n-1} : un bit à 1 à gauche et tous les autres à 0

- Comment représenter 0 ?

$\rightarrow$ Avec tous les bits à 0

- Comment représenter -1 sur n bits ?

$\rightarrow$ Avec n bits à 1

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\longrightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

- Le plus petit nombre négatif représentable sur n bits ?

$\longrightarrow$ -2^{n-1} : un bit à 1 à gauche et tous les autres à 0

- Comment représenter 0 ?

$\longrightarrow$ Avec tous les bits à 0

- Comment représenter -1 sur n bits ?

$\longrightarrow$ Avec n bits à 1

- Représenter -84 en complément à 2 sur 8 bits ?

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\rightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

- Le plus petit nombre négatif représentable sur n bits ?

$\rightarrow$ -2^{n-1} : un bit à 1 à gauche et tous les autres à 0

- Comment représenter 0 ?

$\rightarrow$ Avec tous les bits à 0

- Comment représenter -1 sur n bits ?

$\rightarrow$ Avec n bits à 1

- Représenter -84 en complément à 2 sur 8 bits ?

$\rightarrow$ $-128_{10} + 44_{10} = 10101100$

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\longrightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

- Le plus petit nombre négatif représentable sur n bits ?

$\longrightarrow$ -2^{n-1} : un bit à 1 à gauche et tous les autres à 0

- Comment représenter 0 ?

$\longrightarrow$ Avec tous les bits à 0

- Comment représenter -1 sur n bits ?

$\longrightarrow$ Avec n bits à 1

- Représenter -84 en complément à 2 sur 8 bits ?

$\longrightarrow$ $-128_{10} + 44_{10} = 10101100$

- Valeur en décimal de 11000011 en complément à 2 ?

Représentation des nombres signés en complément à 2 (2/2)

- Le plus grand nombre positif représentable sur n bits ?

$\longrightarrow$ $2^{n-1} - 1$: un bit à 0 à gauche et tous les autres bits à 1

- Le plus petit nombre négatif représentable sur n bits ?

$\longrightarrow$ -2^{n-1} : un bit à 1 à gauche et tous les autres à 0

- Comment représenter 0 ?

$\longrightarrow$ Avec tous les bits à 0

- Comment représenter -1 sur n bits ?

$\longrightarrow$ Avec n bits à 1

- Représenter -84 en complément à 2 sur 8 bits ?

$\longrightarrow$ $-128_{10} + 44_{10} = 10101100$

- Valeur en décimal de 11000011 en complément à 2 ?

$\longrightarrow$ $-128_{10} + 67_{10} = -61_{10}$

Opposé d'un nombre en complément à 2

- Pour trouver la représentation d'un entier négatif : on prend la représentation du nombre positif et on calcule l'opposé
- Basculer tous les bits et ajouter 1
- $C_1(a)$: complément à 1 => Basculer tous les bits de a
- $C_2(a)$: complément à 2 => $C_1(a) + 1$

-84_{10} en complément à 2 sur 8 bits :

$$84_{10} = (1 \times 8 + 2) \times 8 + 4 = 124_8 = 1\,010\,100_2$$

$84_{10} =$	0101 0100	
	1010 1011	Complément à 1
	+1	
$-84_{10} =$	1010 1100	Complément à 2

Nombre négatif du complément à 2 vers décimal

- Pour trouver la valeur d'un nombre négatif en complément à 2 :
  on calcule l'opposé, on convertit en décimal et on ajoute un signe –.

Représentation en décimal du nombre 11 000 011 en complément à 2 sur 8 bits :

$$\begin{array}{rl} a & 11\ 000\ 011 \\ C_1(a) & 00\ 111\ 100 \\ & \qquad\quad +\ 1 \\ \hline C_2(a) & 00\ 111\ 101 \end{array}$$

$111\,101 = 75_8 = 61_{10}$

Avec le signe : -61_{10}

Calcul rapide du complément à 2

```
i ← 0
```

Première partie : copier jusqu'au premier 1

tant que $a_i \neq 1$ faire

  $b_i \leftarrow a_i$

  i ← i+1

fin tant que

  $b_i \leftarrow a_i$

  i ← i+1

Deuxième partie : inverser jusqu'à la fin

tant que $i \leq n$ faire

  $b_i \leftarrow 1 - a_i$

  i ← i+1

fin tant que

Tester sur : 10100111 et 10101000

Addition en complément à 2

	10 101 100	-84
+	10 111	$+23$
=	11 000 011	$= -61$

- L'addition peut produire un bit à 1 supplémentaire
- Attention à ne conserver que le nombre de bits utilisés pour la représentation !

		11 000 011	-61
+		11 101 001	-23
=	~~1~~	10 101 100	= -84

Débordement de capacité

- Si on additionne deux nombres positifs on peut avoir un débordement de capacité.
- Exemple sur 8 bits :

	01 111 111	127
+	00 000 001	1
=	10 000 000	= -128 !

	01 011 101	93
+	00 101 011	136
=	10 001 000	= -120 !

Supplément

- On peut faire la multiplication des nombre en complément à 2 avec l'algorithme de *Booth*
- Sinon on peut évaluer le signe, multiplier les valeurs absolue, puis prendre le complément à 2 si le résultat doit être négatif
- Sur une calculatrice en mode "programmeur" les touches $1's$ et $2's$ calculent respectivement le complément à 1 et à 2