# Codage des nombres entiers relatifs

M. Combacau
combacau@laas.fr

UNIVERSITÉ TOULOUSE III PAUL SABATIER

NORTHEASTERN UNIVERSITY

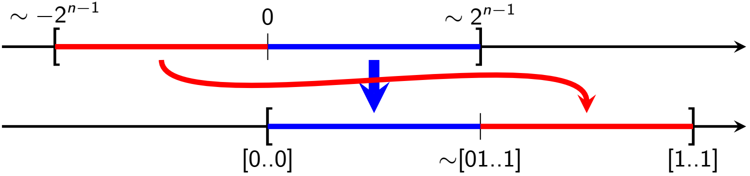
12 novembre 2024

## Objectif

Savoir coder un nombre entier positif ou négatif en base 2
Connaître les trois techniques de codage d'un nombre entier relatif

# Principe de la technique de codage d'un nombre négatif

- L'intervalle de valeur s'étend à certains nombres négatifs.
- Le plus naturel est de coder un intervalle centré sur 0 dans $\mathbb{Z}$
- Dans ce cas, la valeur absolue du nombre codé sur $n$ bits ne dépassera jamais $2^{n-1}$
- Les codes affectés aux valeurs positives doivent respecter si possible le codage binaire vu dans la séquence précédente



# Trois modes de codage (1)

**Signe + valeur absolue**

Le bit de poids fort sert de signe (=0 positif, =1 : négatif)

- Deux codes pour 0 : [0..0] (+0) et [10..0](-0) (peu optimal)
- Conserve le codage des nombres entiers positifs en binaire naturel
- Code l'intervalle $[-2^{n-1}+1, +2^{n-1}-1]$
- Valeur codée $= (-1)^{a_{n-1}} \times (a_{n-2}.2^{n-2} + \ldots + a_0)$
- En général seule la valeur [0..0] est utilisée pour le code de 0
- Semble très naturel, mais peu commode en informatique pour les opérations (sauf pour l'extraction de la valeur absolue)

# Trois modes de codage (2)

**Avec biais**

Codage de $A + Biais$ au lieu de $A$ (avec $Biais > 0$)

- Pour conserver la symétrie du codage le biais vaut $2^{n-1}$ ou $2^{n-1} - 1$
- Perd le codage des nombres entiers positifs en binaire naturel
- Code l'intervalle $[-Biais, +2^n - 1 - Biais]$
- Valeur codée $= (a_{n-1}.2^{n-1} + \ldots + a_0) - Biais$
- Conserve la distance entre les nombres dans les codes (soustraction)
- Le bit $a_{n-1}$ sert de bit de signe (=0 : négatif, =1 :positif)

# Trois modes de codage (3.1)

## En complément à la base (ca2)

$A \geq 0$ : codage de $A$

$A < 0$ : codage de $2^n - |A|$

- Délicat car fonction de codage non continue
- Conserve le codage des nombres entiers positifs en binaire naturel
- Code l'intervalle $[-2^{n-1}, +2^{n-1} - 1]$
- Le bit $a_{n-1}$ sert de bit de signe (=1 : négatif, =0 : positif)
- Valeur $= \begin{cases} (a_{n-2}.2^{n-2} + \ldots + a_0) & si\ A \geq 0\ (a_{n-1} = 0) \\ (a_{n-1}.2^{n-1} + \ldots + a_0) - 2^n & si\ A < 0\ (a_{n-1} = 1) \end{cases}$
- Possède une propriété intéressante pour la soustraction

# Trois modes de codage (3.2)

Propriété vis à vis de la soustraction

- Soit un nombre entier positif $A$ codé en binaire sur $n$ bits. La valeur codée est celle de $|A|$
- Quelle est la valeur codée pour le nombre $-A$ ?
  Par définition, c'est la valeur $2^n - |A|$
- Que vaut la somme $A + (-A)$ ?
  $A + (-A) = |A| + 2^n - |A| = 2^n$
- Ainsi, le résultat sur $n$ bits de la somme d'un nombre et de son opposé vaut 0. Le ca2 transforme la soustraction en addition.
- Pour effectuer $A$-$B$ (A et B tous deux positifs), on calcule le complément à 2 de $B$ et on additionne ce résultat à $A$.

Codage utilisé pour cette propriété !

# Trois modes de codage (3.3)

Calcul du code de $-|A| = Ca2(-|A|)$

- Solution 1 : coder en binaire naturel $2^n - |A|$ (définition sur 8 bits)
- Solution 2 : soit $cb(|A|)$ le code binaire de $|A|$
  - Complémentons les bits de $cb(|A|)$ et notons le résultat $\overline{cb(|A|)}$
  - $\overline{cb(|A|)}$ est le code de la valeur notée $\overline{|A|}$
  - La somme $cb(|A|) + \overline{cb(|A|)} = [1 \ldots 1]$ de valeur $2^n - 1$
  - On peut donc écrire : $|A| + \overline{|A|} + 1 = 2^n - 1 + 1 = 2^n$
  - Soit encore $\overline{|A|} + 1 = 2^n - |A|$
  - D'où $\overline{cb(|A|)} + 1 = ca2(|A|)$
- Ceci fournit un algorithme pour calculer le code binaire de $-|A|$ en ca2 à partir de son code binaire naturel $cb(|A|)$ que l'on sait calculer par divisions successives (entre autres)

# Trois modes de codage (3.4)

**Calcul de** $\overline{cb(|A|)} + 1$

$$
\begin{array}{lcccccccccl}
cb(|A|) & = & a_{n-1} & \dots & a_j & 1 & 0 & \dots & 0 & (1) \\
\overline{cb(|A|)} & = & \overline{a_{n-1}} & \dots & \overline{a_j} & 0 & 1 & \dots & 1 & (2) \\
 & & & & & & & + & 1 & (3) \\
\hline
\overline{cb(|A|)} + 1 & = & \overline{a_{n-1}} & \dots & \overline{a_j} & 1 & 0 & \dots & 0 & (4)
\end{array}
$$

- Après complément des bits du code binaire naturel de $|A|$ (2)
- On rajoute 1 ce qui propage une retenue sur tous les bits de poids faible à 1 consécutifs (en bleu dans (2))
- Le résultat (4) peut donc être construit à partir de (1) par :
  1. les bits de la partie "poids faible du code (1) de la forme $1[0 \dots 0]$ sont conservés (il peut ne pas y avoir de $[0 \dots 0]$)
  2. tous les autres bits sont complémentés

# Exercices de codage : signe + valeur absolue (sur 8 bits)

**Code de -6**

- le code de 6(=4+2) en binaire naturel est [00000110]
- le code de -6 est donc [10000110]

**Code de -45**

- le code de 45(=32+8+4+1) en binaire naturel est [00101101]
- le code de -45 est donc [10101101]

# Exercices de codage : biais de $2^{n-1}$ (=128 sur 8 bits)

**Code de +32**

- Par définition : code binaire de 128+32,
  soit [10100000]

**Code de -45**

- Par définition code binaire de 128-45 = 83=64+16+2+1,
  soit [01010011]
- Remarque :128-45 =127-44. avec 127 ($2^8 - 1$) codé par [01111111]
- 44 = 32+8+4 est codé [00101100]
- Soustraire 44 au code de 127 (les bits à 1 prennent la valeur 0)
  soit [01010011] qui est bien le code binaire de 83

# Exercices de codage : complément à 2 (sur 8 bits)

**Code de -14**

- Première solution (appliquer la définition)

Donc, codons 256-14= 242 = 255-13=255-8-4-1, soit[11110010]

- Deuxième solution (algorithmique)

Codons 14=8+4+2 : [00001110],

Dans le code en Ca2 de -14,

1. la partie en bleu est conservée
2. la partie en noir est complémentée

D'où $Ca2(-14) = [11110010]$

Nous retrouvons bien le code de 242 calculé plus haut

**Code de -61**

-Code de 61 = 63-2 :[00111101] ; d'où $Ca2(-61) = [11000011]$

Qui est bien le code binaire de 256-61=195