

**INTRODUCTION AUX
SYSTÈMES INFORMATIQUES**
REPRÉSENTATION DES DONNÉES
ENTIÈRES

Victor Ostromoukhov

Département d'Informatique et de Recherche Opérationnelle
Http : [//www.iro.umontreal.ca/~ostrom/](http://www.iro.umontreal.ca/~ostrom/)
E-mail : ostrom@iro.umontreal.ca

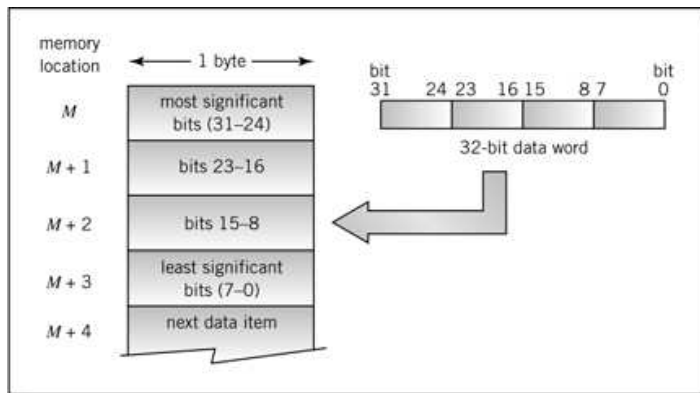
**REPRÉSENTATION DES DONNÉES
ENTIÈRES**
SOMMAIRE

Entiers Positifs	2
Entiers Signés -Signe et Magnitude	4
Entiers Signés -Complément Logique	4
Entiers Signés -Complément Arithmétique ...	12
Soustraction Binaire	17
Division D'Entiers	18

REPRÉSENTATION DES DONNÉES ENTIÈRES ENTIERS POSITIFS

Représentation des Entiers positifs

Un approche évidente ..

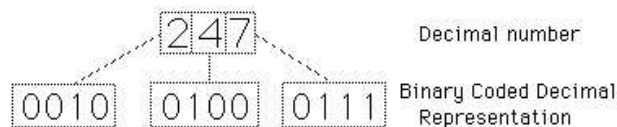


Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 4, Figure 04-01

Codage en Binaire

8 bits \Leftrightarrow 256 valeurs possibles
32 bits \Leftrightarrow 4294967296 valeurs possibles

Codage en BCD



8 bits \Leftrightarrow 99 valeurs possibles

REPRÉSENTATION DES DONNÉES ENTIÈRES ENTIERS POSITIFS

Value Range for Binary Versus Binary-coded Decimal

No. of Bits	BCD range		Binary range	
4	0-9	1 digit	0-15	1+ digit
8	0-99	2 digits	0-255	2+ digits
12	0-999	3 digits	0-4,095	3+ digits
16	0-9,999	4 digits	0-65,535	4+ digits
20	0-99,999	5 digits	0-1 Million	6 digits
24	0-999,999	6 digits	0-16 Million	7+ digits
32	0-99,999,999	8 digits	0-4 Billion	9+ digits
64	0-(10 ¹⁶ -1)	16 digits	0-16 Quintillion	19+ digits

- Le nombre de valeur codable en BCD est moins important qu'en binaire
- L'arithmétique en BCD est plus difficile en BCD qu'en binaire et plus lente (alternative BCD \triangleright Binaire \triangleright calcul \triangleright conversion BCD)

$$\begin{array}{r}
 76 \rightarrow 0111\ 0110_{\text{bcd}} \\
 \times 7 \rightarrow \quad \quad 0111_{\text{bcd}} \\
 \hline
 42 \rightarrow \quad \quad 101010_{\text{bin}} \rightarrow 0100\ 0010_{\text{bcd}} \\
 49 \rightarrow 110001_{\text{bin}} \rightarrow + 0100\ 1001_{\text{bcd}} \\
 \hline
 432 \rightarrow \quad \quad \quad \quad \quad \quad 0100\ 1101\ 0010 \\
 13 \leftarrow \text{adjust carry} \quad \quad \quad \quad \quad \quad + 0001\ 0011 \\
 \hline
 532 \rightarrow \quad \quad \quad \quad \quad \quad 0101\ 0011\ 0010 \\
 \quad \quad \quad \quad \quad \quad \quad \quad \quad = 532 \text{ in BCD}
 \end{array}$$

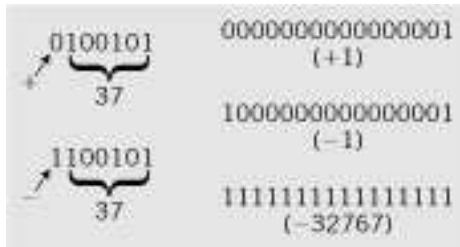
- Préféré pour certaines application (business) où il est nécessaire d'avoir une représentation exacte du nombre décimal et conversion BCD \triangleright caractère facile

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIERS SIGNÉS -SIGNE ET MAGNITUDE

Représentation des Entiers Signés

1. Représentation Signe et magnitude



• la moitié des codes est affecté au nombres positives et l'autre moitié au nombre négatives

- 2 valeurs pour le zéro
 $+0$ (00000000) et -0 (10000000)

Pour un codage n bits,
 Min : $-(2^{n-1} - 1)$ Max : $(2^{n-1} - 1)$

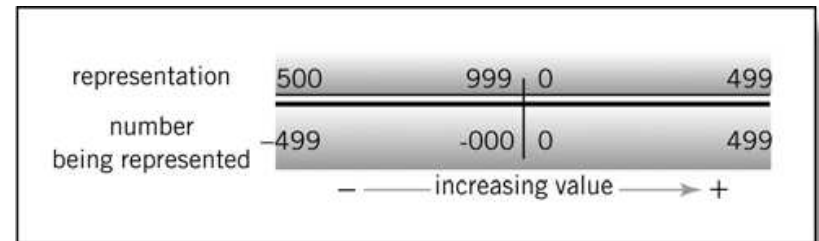
- Les algorithmes de calcul sont plus difficiles a implémenter en hardware
 - ▷ Le système doit tester à la fin de chaque calcul pour assurer qu'il n'y a qu'un seul zéro
 - ▷ Deux techniques de calcul différents lorsque les deux nombres sont de même signe ou de signe différents

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIER SIGNÉS -COMPLÈMENT LOGIQUE

2. Complément logique

En Base 10



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition
 Chapter 4, Figure 04-05

- En base 10, on dit que l'on prend la représentation en complément à 9 et on peut l'obtenir :

Représenter -467_{10} (3 digits) en complément à 9 ?

$$\begin{array}{r} 999 \\ - 467 \\ \hline 532 \end{array}$$

Représenter -467_{10} (4 digits) en complément à 9 ?

$$\begin{array}{r} 9999 \\ - 467 \\ \hline 9532 \end{array}$$

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIER SIGNÉS -COMPLÉMENT LOGIQUE

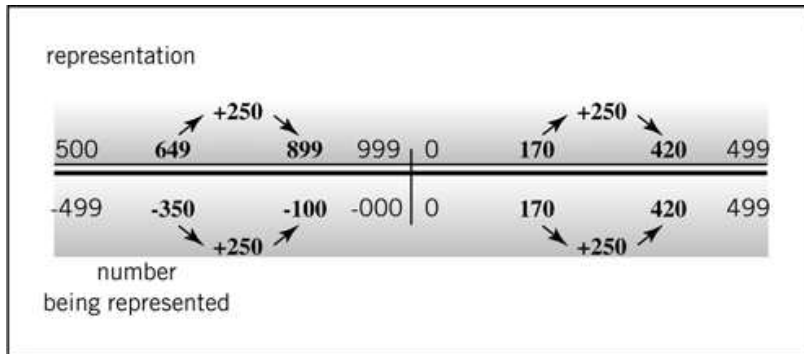
Quel est la valeur du signe et la magnitude de 9990 lorsque celui ci est représenté en complément à 9 ?

- Le premier digit est supérieur à 4, donc **signe négative**

$$\begin{array}{r}
 9999 \\
 - 9990 \\
 \hline
 \text{---} \\
 9
 \end{array}$$

Donc 9990 en complément à 9 représente -9

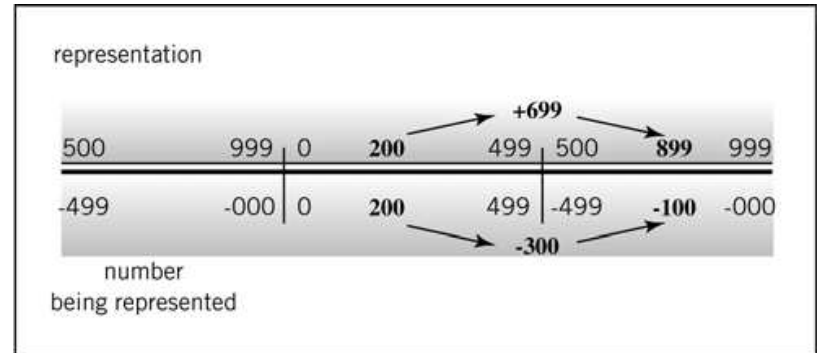
Addition



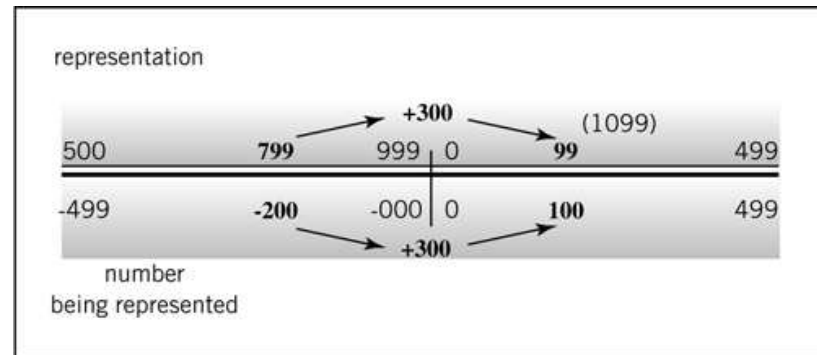
Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 4, Figure 04-06

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIER SIGNÉS -COMPLÉMENT LOGIQUE



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 4, Figure 04-07



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition Chapter 4, Figure 04-08

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIER SIGNÉS -COMPLÈMENT LOGIQUE

- En conséquence, une procédure pour additionner 2 chiffres dans le cas où le résultat s'étend au delà du nombre maximum de digits consiste à ajouter la dernière retenue

$-200_{10} + 100_{10}$ en complément à 9 sur 3 digits

$-200_{10} + 300_{10}$ en complément à 9 sur 3 digits

$\begin{array}{r} 799 \\ 100 \\ \hline 899 \end{array}$ <p>No end-around carry</p>	$\begin{array}{r} 799 \\ 300 \\ \hline 1099 \\ \hline \rightarrow 1 \\ 100 \end{array}$ <p>End-around carry</p>
--	---

- Pour soustraire, on prend le complément du chiffre que l'on doit soustraire et on réalise l'addition
- Possibilité de débordement [*overflow*]
Exemple : $300 + 300 = 600$ (-399) ?

Si les deux entrées de l'addition ont le même signe et le signe du résultat est différent alors on a un problème de débordement

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIER SIGNÉS -COMPLÈMENT LOGIQUE

En Base 2

Table d'Addition

0+0=0
 1+0=0+1=1
 1+1=0 carry 1
 1+1+1=1 carry 1

46	$1\ 0\ 1\ 1\ 1\ 0$
$+7$	$+ \quad \quad \quad 1\ 1\ 1$
53	$1\ 1\ 0\ 1\ 0\ 1$

↙ add two binary numbers

Le complément à 1 est simplement réalisé en changeant les zéros en un et l'inverse (\triangleright **inversion**)

10000000	11111111	00000000	01111111
-127_{10}	-0_{10}	0_{10}	127_{10}

**REPRÉSENTATION DES DONNÉES
ENTIÈRES**
ENTIERS SIGNÉS -COMPLÉMENT LOGIQUE

Add

$$\begin{array}{r} 00101101 = 45 \\ \underline{00111010} = \underline{58} \\ 01100111 = 103 \end{array}$$

■■■

Add the 16-bit numbers

$$\begin{array}{r} 0000000000101101 = 45 \\ \underline{1111111111000101} = \underline{-58} \\ 111111111110010 = -13 \end{array}$$

Add

$$\begin{array}{r} 01000000 = 64 \\ \underline{+01000001} = \underline{65} \\ 10000001 = -126 \end{array}$$

**REPRÉSENTATION DES DONNÉES
ENTIÈRES**
ENTIERS SIGNÉS -COMPLÉMENT LOGIQUE

Add

$$\begin{array}{r} 01101010 = 106 \\ \underline{11111101} = \underline{-2} \\ \textcircled{0}01100111 \\ \text{(end-around carry)} \quad \xrightarrow{+1} \\ 01101000 = 104 \end{array}$$

■■■

Subtract

$$\begin{array}{r} 01101010 = 106 \\ \underline{-01011010} = \underline{90} \end{array}$$

Changing the sign of the addend by inverting

$$\begin{array}{r} 01101010 \\ \underline{10100101} \\ \textcircled{0}00001111 \\ \text{(end-around carry)} \quad \xrightarrow{+1} \\ 00010000 = 16 \end{array}$$

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIERS SIGNÉS -COMPLÈMENT ARITHMÉTIQUE

3. Complément arithmétique

En Base 10

Representation	500	999	0	499
Number being represented	-500	-001	0	499
	-	-	+	

- En base 10, la représentation en complément à 10 peut être obtenu

Représenter -467_{10} (3 digits) en complément à 10 ?

$$\begin{array}{r} 1000 \\ - 467 \\ \hline 533 \end{array}$$

Représenter -467_{10} (4 digits) en complément à 10 ?

$$\begin{array}{r} 10000 \\ - 467 \\ \hline 9533 \end{array}$$

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIERS SIGNÉS -COMPLÈMENT ARITHMÉTIQUE

Quel est la valeur du signe et la magnitude de 9990 lorsque celui-ci est une représentation complémentaire à 10 avec 4 digits ?

- Le premier digit est supérieur à 4, donc **signe négative**

$$\begin{array}{r} 10000 \\ - 9990 \\ \hline 10 \end{array}$$

Donc 9990 en complément à 10 représente -10

Complément à 10 = Complément à 9 + 1

- **Additions simples !**

$-200_{10} + 100_{10}$ en complément à 10 sur 3 digits

$-200_{10} + 300_{10}$ en complément à 10 sur 3 digits

$$\begin{array}{r} 800 \\ + 100 \\ \hline 900 \end{array} \qquad \begin{array}{r} 800 \\ + 300 \\ \hline 1\ 100 \end{array}$$

↑
*On laisse tomber
la retenue*

*Toute retenue au delà du nombre de digit
n'est pas prise en compte*

REPRÉSENTATION DES DONNÉES ENTIÈRES ENTIERS SIGNÉS -COMPLÈMENT ARITHMÉTIQUE

En Base 2

10000000	11111111	00000000	01111111
-128_{10}	-1_{10}	0_{10}	127_{10}

- Les nombres positifs sont représentés par eux-mêmes
- Un nombre négatif commencera par un "1"

Pour trouver le complément à deux d'un nombre :

- 1- Soustraire la valeur au modulus
- 2- Trouver le complément à un et ajouter 1

Exemple :

$$39_{10} = 00100111_2$$

$$-39_{10} = 11011001_2$$

2) 39	00100111	->	11011000	
2) 19	1			+1
2) 9	1			
2) 4	1			
2) 2	0			
2) 1	0			
0	1			

convert a decimal number to binary (arrow pointing to the first column)
change the sign bit: 1 toggle bit, 2 add 1 (arrow pointing to the sign bit)

REPRÉSENTATION DES DONNÉES ENTIÈRES ENTIERS SIGNÉS -COMPLÈMENT ARITHMÉTIQUE

- Possibilité de débordement [*overflow*]

Si les deux entrées de l'addition ont le même signe et le signe du résultat est différent alors on a un problème de débordement

$(+4) + (+2)$ $\begin{array}{r} 0100 \\ 0010 \\ \hline 0110 \end{array} = (+6)$ <p style="font-size: small;">no overflow, no carry</p>	$(+4) + (+6)$ $\begin{array}{r} 0100 \\ 0110 \\ \hline 1010 \end{array} = (-6)$ <p style="font-size: small;">overflow, no carry the result is incorrect</p>
---	--

$(-4) + (-2)$ $\begin{array}{r} 1100 \\ 1110 \\ \hline 11010 \end{array} = (-6)$ <p style="font-size: small;">no overflow, carry ignoring the carry, the result is correct</p>	$(-4) + (-6)$ $\begin{array}{r} 1100 \\ 1010 \\ \hline 10110 \end{array} = (+3)$ <p style="font-size: small;">overflow, carry ignoring the carry, the result is incorrect</p>
---	--

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIERS SIGNÉS - COMPLÉMENT ARITHMÉTIQUE

Table de Multiplication

$0 \cdot 0 = 0$
$0 \cdot 1 = 0$
$1 \cdot 0 = 0$
$1 \cdot 1 = 1$

Multiplication d'Entiers

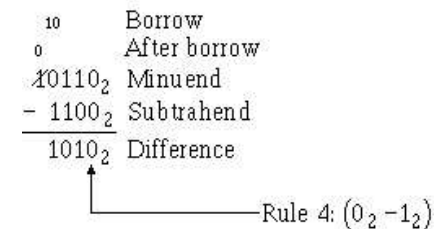
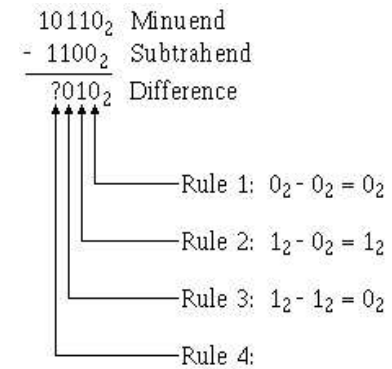
Base 2 Place	2^4	2^3	2^2	2^1	2^0	
		1	0	0	1	9
		x		1	0	x 2
Carry		0	0	0	0	18
		1	0	0	1	
		1	0	0	1	0

1011	Multiplicand (11)
x1101	Multiplier (13)
1011	} Partial products
0000	
1011	
1011	
10001111	Product (143)

REPRÉSENTATION DES DONNÉES ENTIÈRES

SOUSTRACTION BINAIRE

Soustraction binaire



REPRÉSENTATION DES DONNÉES ENTIÈRES

DIVISION D'ENTIERS

Division d'Entiers

(Dividende)	147	11 (Diviseur)
	4	13
	(Reste)	(Quotient)

	00001101	← Quotient
Divisor →	1011 / 10010011	← Dividend
	$\begin{array}{r} 1011 \\ \underline{1011} \\ 001110 \end{array}$	
Partial remainders →	$\begin{array}{r} 001110 \\ 1011 \\ \underline{001111} \end{array}$	
	$\begin{array}{r} 1011 \\ \underline{1011} \\ 100 \end{array}$	← Remainder