



DIRO
IFT 1215

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

Max Mignotte

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

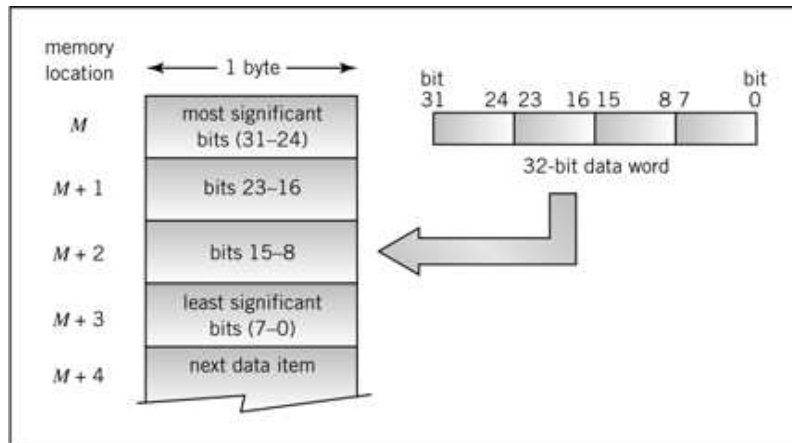
**REPRÉSENTATION DES DONNÉES
ENTIÈRES
SOMMAIRE**

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 ..

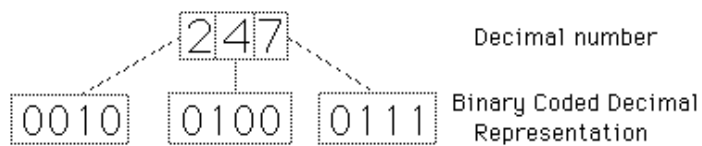


Englander: The Architecture of Computer
Hardware and Systems Software, 2nd edition
Chapter 4, Figure 04-01

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 ▷ Binaire ▷ calcul ▷ conversion BCD)

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

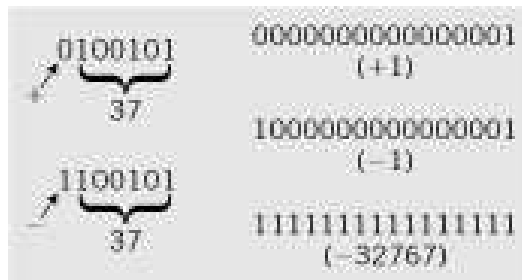
- Préféré pour certaines application (business) où il est nécessaire d'avoir une représentation exacte du nombre décimal et conversion BCD ▷ caractère facile

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,

$$\text{Min: } -(2^{n-1} - 1) \quad \text{Max: } (2^{n-1} - 1)$$

- 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 - COMPLÉMENT LOGIQUE

2. Complément logique

En Base 10

representation	500	999	0	499
number being represented	-499	-000	0	499
	— ——— increasing value ———> +			

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 -COMPLÈMENT LOGIQUE

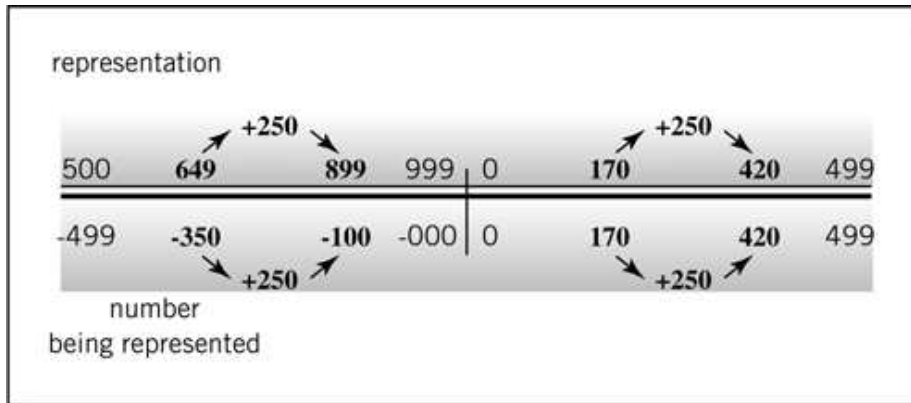
Quel est la valeur du signe et la magnitude de 9990 lorsque celui ci est représenté en complément à 9 ?

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

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

Donc 9990 en complément à 9 représente -9

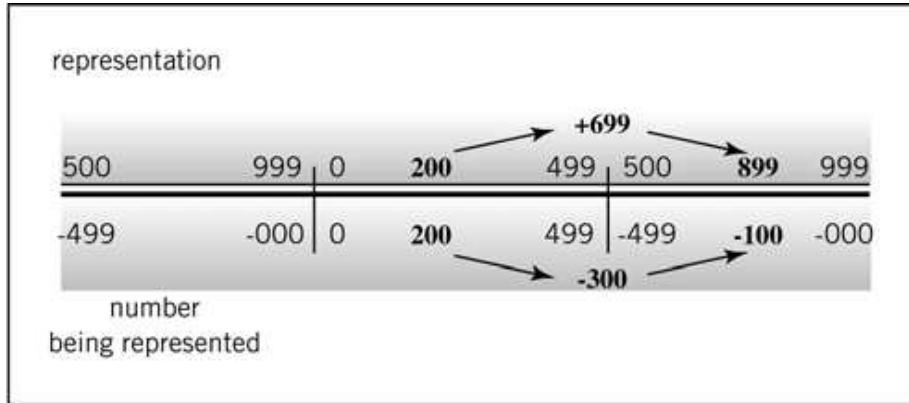
Addition



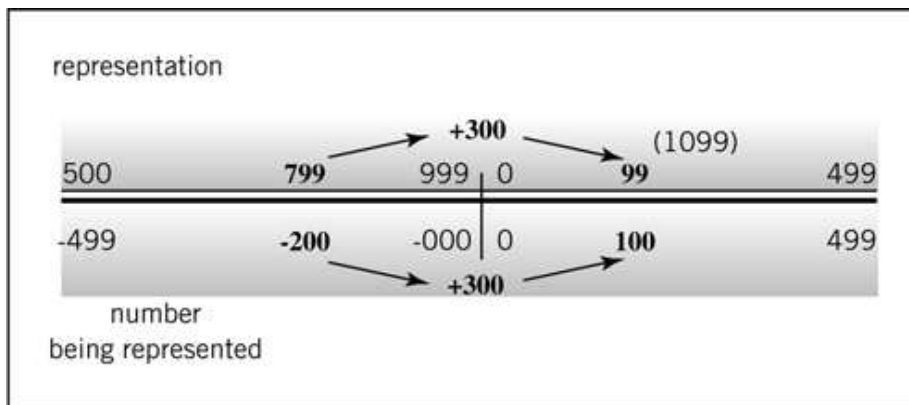
Englander: The Architecture of Computer
Hardware and Systems Software, 2nd edition
Chapter 4, Figure 04-06

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIERS 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 -COMPLÉMENT 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

$\begin{array}{r} 799 \\ 100 \\ \hline 899 \end{array}$	$\begin{array}{r} 799 \\ 300 \\ \hline 1099 \\ \hline \leftarrow 1 \\ 100 \end{array}$
No end-around carry	End-around carry

- 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 - COMPLÉMENT LOGIQUE

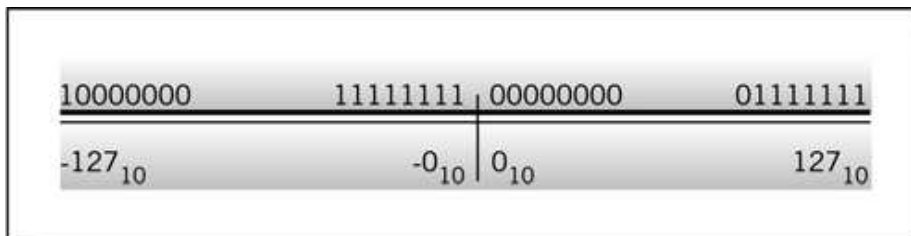
En Base 2

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	<i>add two binary numbers</i>
+7	+ 1 1 1	
53	1 1 0 1 0 1	

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



Englander: The Architecture of Computer Hardware and Systems Software, 2nd edition
Chapter 4, Figure 04-10

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

Add

$$00101101 = 45$$

$$\underline{00111010} = \underline{58}$$

$$01100111 = 103$$

■ ■ ■

Add the 16-bit numbers

$$0000000000101101 = 45$$

$$\underline{1111111111000101} = \underline{-58}$$

$$111111111110010 = -13$$

Add

$$01000000 = 64$$

$$\underline{+01000001} = \underline{65}$$

$$10000001 = -126$$

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIERS SIGNÉS - COMPLÉMENT LOGIQUE

Add

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

■■■

Subtract

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

Changing the sign of the addend by inverting

$$\begin{array}{r}
 01101010 \\
 \underline{10100101} \\
 00000111 \\
 \text{(end-around carry) } \xrightarrow{+1} \\
 \hline
 00010000 = 16
 \end{array}$$

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIERS SIGNÉS - COMPLÉMENT ARITHMÉTIQUE

3. Complément arithmétique

En Base 10

Representation	500	999	10	499
Number being represented	500	-500	0	499

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

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

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

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

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

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIER SIGNED - COMPLÉMENT 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**

$$\begin{array}{r} 10000 \\ - \quad 9990 \\ \hline 000 \\ 000 \\ 000 \\ 000 \\ \hline 10 \end{array}$$

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

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

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

↑
*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 - COMPLÉMENT 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) <u>39</u>	00100111	->	11011000	
2) <u>19</u>	1			+1
2) <u>9</u>	1		11011001	
2) <u>4</u>	1			
2) <u>2</u>	0			
2) <u>1</u>	0			
	0	1		

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIERS SIGNÉS - COMPLÉMENT 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*

<p>(+4) + (+2)</p> $\begin{array}{r} 0100 \\ 0010 \\ \hline 0110 \end{array} = (+6)$ <p style="margin-left: 20px;">no overflow, no carry</p>	<p>(+4) + (+6)</p> $\begin{array}{r} 0100 \\ 0110 \\ \hline 1010 \end{array} = (-6)$ <p style="margin-left: 20px;">overflow, no carry the result is incorrect</p>
--	---

<p>(-4) + (-2)</p> $\begin{array}{r} 1100 \\ 1110 \\ \hline 11010 \end{array} = (-6)$ <p style="margin-left: 20px;">no overflow, carry ignoring the carry, the result is correct</p>	<p>(-4) + (-6)</p> $\begin{array}{r} 1100 \\ 1010 \\ \hline 10110 \end{array} =$ <p style="margin-left: 20px;">overflow carry ignoring the carry, the result is incorrect</p>
--	---

REPRÉSENTATION DES DONNÉES ENTIÈRES

ENTIERS SIGNÉS - COMPLÉMENT ARITHMÉTIQUE

Table de Multiplication

0 . 0 = 0
0 . 1 = 0
1 . 0 = 0
1 . 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	

<pre style="font-family: monospace; font-size: 1.2em;"> 1011 x1101 ----- 1011 0000 1011 1011 ----- 10001111</pre>	<p>Multiplicand (11)</p> <p>Multiplier (13)</p> <p style="font-size: 2em;">}</p> <p>Partial products</p> <p>Product (143)</p>
---	---

REPRÉSENTATION DES DONNÉES ENTIÈRES

SOUSTRACTION BINAIRE

Soustraction binaire

$$\begin{array}{r}
 10110_2 \text{ Minuend} \\
 - 1100_2 \text{ Subtrahend} \\
 \hline
 ?010_2 \text{ Difference}
 \end{array}$$

$\uparrow \uparrow \uparrow$
 Rule 1: $0_2 - 0_2 = 0_2$
 Rule 2: $1_2 - 0_2 = 1_2$
 Rule 3: $1_2 - 1_2 = 0_2$
 Rule 4:

$$\begin{array}{r}
 10 \text{ Borrow} \\
 0 \text{ After borrow} \\
 \cancel{1}0110_2 \text{ Minuend} \\
 - 1100_2 \text{ Subtrahend} \\
 \hline
 1010_2 \text{ Difference}
 \end{array}$$

\uparrow
 Rule 4: $(0_2 - 1_2)$

REPRÉSENTATION DES DONNÉES ENTIÈRES

DIVISION D'ENTIERS

Division d'Entiers

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

	00001101	← Quotient
Divisor →	1011 / 10010011	← Dividend
	$\begin{array}{r} 1011 \\ \underline{1011} \\ 001110 \\ \underline{1011} \\ 001111 \\ \underline{1011} \\ 100 \end{array}$	
Partial remainders	<div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> \swarrow 001110 </div> <div style="text-align: center;"> \searrow 001111 </div> </div>	
	100	← Remainder