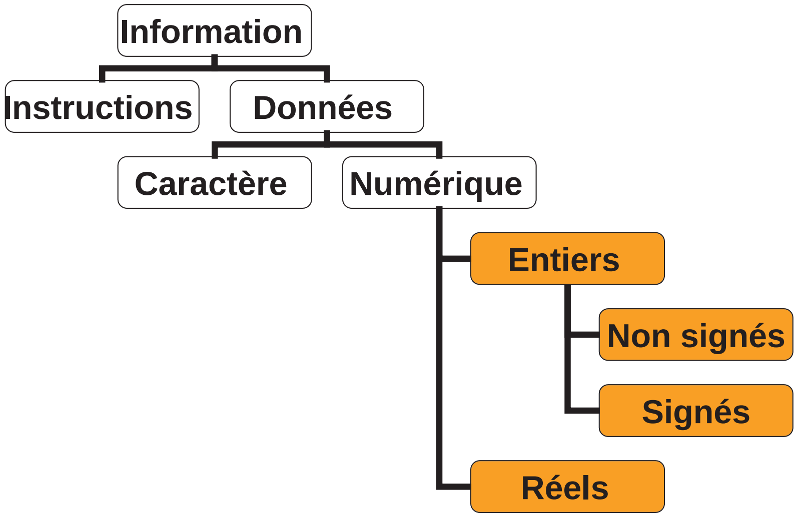
Représentation de l'information dans la machine

- Introduction
- Représentation des nombres négatifs
  - Signe / valeur absolue
  - Complément à 1
  - Complément à 2
- Représentation des nombres réels
  - Virgule fixe
  - Virgule flottante
- Le codage BCD et EXCESS3
- Représentation des caractères

Introduction



Représentation des nombres entiers

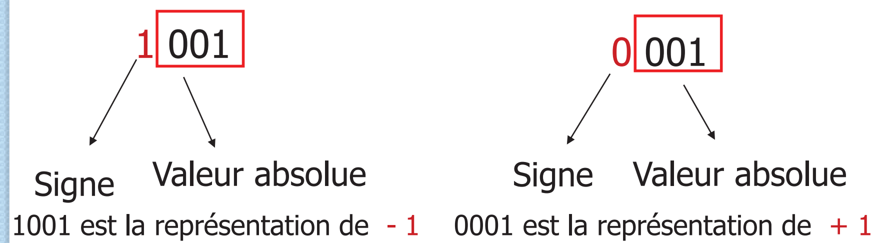
- Il existe deux types d'entiers :
  - les entiers non signés (positifs)
  - et les entiers signés (positifs ou négatifs)
- **Problème :** Comment indiquer à la machine qu'un nombre est négatif ou positif ?

- Il existe 3 méthodes pour représenter les nombres négatifs :
  - Signe/ valeur absolue
  - Complément à 1(complément restreint)
  - Complément à 2 (complément à vrai)

Représentation signe / valeur absolue (S/VA)

- Si on travail sur n bits , alors le bit du poids fort est utilisé pour indiquer le signe :
  - 1 : signe négatif
  - 0 : signe positif
- Les autres bits (n -1) désignent la valeur absolue du nombre.

- Exemple : Si on travail sur 4 bits.



Sur 3 bits on obtient :

signe	VA	valeur
0	00	+ 0
0	01	+ 1
0	10	+ 2
0	11	+ 3
1	00	- 0
1	01	- 1
1	10	- 2
1	11	- 3

- Les valeurs sont comprises entre -3 et +3

$$-3 \leq N \leq +3$$
$$-(4-1) \leq N \leq +(4-1)$$
$$-(2^2-1) \leq N \leq +(2^2-1)$$
$$-(2^{(3-1)}-1) \leq N \leq +(2^{(3-1)}-1)$$

Si on travail sur **n** bits , l'intervalle des valeurs qu'on peut représenter en S/VA :

$$-(2^{(n-1)}-1) \leq N \leq +(2^{(n-1)}-1)$$

Représentation en complément à un

- On appel complément à un d'un nombre N un autre nombre N' tel que :

$$N+N'=2^n-1$$

n : est le nombre de bits de la représentation du nombre N .

Exemple :

Soit N=1010 sur 4 bits donc son complément à un de N :

$$N'=(2^4-1)-N$$

$N'=(16-1)-(1010)_2=(15)-(1010)_2=(1111)_2-(1010)_2$

$=0101$

$$\begin{array}{rcccc} & 1 & 0 & 1 & 0 \\ + & 0 & 1 & 0 & 1 \\ \hline & 1 & 1 & 1 & 1 \end{array}$$

Avantages et inconvénients de la représentation signe/valeur absolue

- C'est une représentation assez simple .
- On remarque que le zéro possède deux représentations +0 et -0 ce qui conduit à des difficultés au niveau des opérations arithmétiques.
- Pour les opérations arithmétiques il nous faut deux circuits : l'un pour l'addition et le deuxième pour la soustraction .

L'idéal est d'utiliser un seul circuit pour faire les deux opérations, puisque

$$a-b=a+(-b)$$

Remarque 1 :

- Pour trouver le complément à un d'un nombre, il suffit d'inverser tous les bits de ce nombre : si le bit est un 0 mettre à sa place un 1 et si c'est un 1 mettre à sa place un 0 .

Exemple :

Sur 4 Bits

$$\begin{array}{cccc} 1 & 0 & 1 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 1 & 0 & 1 \end{array}$$

Sur 5 Bits

$$\begin{array}{ccccc} 0 & 1 & 0 & 1 & 0 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 0 & 1 & 0 & 1 \end{array}$$

Remarque 2

- Dans cette représentation , le bit du poids fort nous indique le signe (0 : positif , 1 : négatif).
- Le complément à un du complément à un d'un nombre est égale au nombre lui même .

$$CA1(CA1(N)) = N$$

Exemple :

Quelle est la valeur décimale représentée par la valeur 101010 en complément à 1 sur 6 bits ?

- Le bit poids fort indique qu'il s'agit d'un nombre négatif.

$$\text{Valeur} = -\ CA1(101010)$$
$$= -\ (010101)_2 = -\ (21)_{10}$$

• Sur 3 bits on remarque que les valeurs sont comprises entre -3 et +3

$$-3 \leq N \leq +3$$
$$-(4-1) \leq N \leq +(4-1)$$
$$-(2^2-1) \leq N \leq +(2^2-1)$$
$$-(2^{(3-1)}-1) \leq N \leq +(2^{(3-1)}-1)$$

- Si on travail sur n bits , l'intervalle des valeurs qu'on peut représenter en CA1 :

$$-(2^{(n-1)}-1) \leq N \leq +(2^{(n-1)}-1)$$

Si on travail sur 3 bits :

Valeur en CA1	Valeur en binaire	Valeur décimal
000	000	+ 0
001	001	+ 1
010	010	+ 2
011	011	+ 3
100	- 011	- 3
101	- 010	- 2
110	- 001	- 1
111	- 000	- 0

•Dans cette représentation , le bit du poids fort nous indique le signe .

•On remarque que dans cette représentation le zéro possède aussi une double représentation (+0 et – 0) .

Complément à 2 (complément à vrai)

- Si on suppose que **a** est un nombre sur **n** bits alors :

$$a + 2^n = a \text{ modulo } 2^n$$

et si on prend le résultat sur n bits on va obtenir la même valeur que a .

$$\boxed{a + 2^n = a}$$

Exemple : soit a = 1001 sur 4 bits

$2^4 = 10000$

$$\begin{array}{rccccc} & & 1 & 0 & 0 & 1 \\ + & 1 & 0 & 0 & 0 & 0 \\ \hline & 1 & 1 & 0 & 0 & 1 \end{array}$$

Si on prend le résultat sur 4 bits on trouve la même valeur de

a = 1001

• Si on prend deux nombres entiers **a** et **b** sur **n** bits , on remarque que la soustraction peut être ramener à une addition :

$$a - b = a + (-b)$$

• Pour cela il suffit de trouver une valeur équivalente à **-b** ?

$a - b = a + 2^n - b = a + (2^n - 1) - b + 1$

On a $b + CA1(b) = 2^n - 1$ donc $CA1(b) = (2^n - 1) - b$

Si on remplace dans la première équation on obtient :

$a - b = a + CA1(b) + 1$

La valeur CA1(b)+1 s'appelle le complément à deux de b :

CA1(b)+1 = CA2(b)

Et enfin on va obtenir : a - b = a+ CA2(b) → transformer la soustraction en une addition .

Remarque 2

- Dans cette représentation , le bit du poids fort nous indique le signe (0 : positif , 1 : négatif).
- Le complément à deux du complément à deux d'un nombre est égale au nombre lui même .

CA2(CA2(N))= N

- Exemple :

  Quelle est la valeur décimale représentée par la valeur 101010 en complément à deux sur 6 bits ?
- Le bit poids fort indique qu'il s'agit d'un nombre négatif.
- Valeur = - CA2(101010)

  = - (010101 + 1)

  = - $(010110)_2$ = - (22)

Exemple

- Trouver le complément à vrai de : 01000101 sur 8 bits ?

CA2(01000101)= CA1(01000101)+ 1
CA1(01000101)= (10111010)
CA2(01000101)=(10111010)+ 1 = (10111011)

Remarque 1 :

Pour trouver le compétemment à 2 d'un nombre : il faut parcourir les bits de ce nombre à partir du poids faible et garder tous les bits avant le premier 1 et inverser les autres bits qui viennent après.

0	1	0	0	0	1	0	1
↓	↓	↓	↓	↓	↓	↓	↓
1	0	1	1	1	0	1	1

1	1	0	1	0	1	0	0
↓	↓	↓	↓	↓	↓	↓	↓
0	0	1	0	1	1	0	0

Si on travail sur 3 bits :

Valeur en CA2	Valeur en binaire	valeur
000	000	+ 0
001	001	+ 1
010	010	+ 2
011	011	+ 3
100	- 100	- 4
101	- 011	- 3
110	- 010	- 2
111	- 001	- 1

• Dans cette représentation , le bit du poids fort nous indique le signe .
• On remarque que le zéro n'a pas une double représentation.

•Sur 3 bits on remarque que les valeurs sont comprises entre -4 et +3

$$-4 \le N \le +3$$

$$-4 \le N \le +(4-1)$$

$$-2^{2} \le N \le +(2^{2}-1)$$

$$-2^{(3-1)} \le N \le (2^{(3-1)}-1)$$

Si on travail sur n bits , l'intervalle des valeurs qu'on peut représenter en CA2 :

$$-(2^{(n-1)}) \le N \le +(2^{(n-1)}-1)$$

La représentation en complément à deux (complément à vrai) est la représentation la plus utilisée pour la représentation des nombres négatifs dans la machine.

Opérations arithmétiques en CA2

Effectuer les opérations suivantes sur 5 Bits , en utilisant la représentation en CA2

$$\begin{array}{r} +9 \\ +4 \\ \hline +13 \end{array} \qquad \begin{array}{rr} & 0\,1\,0\,0\,1 \\ + & 0\,0\,1\,0\,0 \\ \hline & 0\,1\,1\,0\,1 \end{array}$$

Le résultat est positif

$(01101)_2 = (13)_{10}$

$$\begin{array}{r} +9 \\ -4 \\ \hline +5 \end{array} \qquad \begin{array}{rr} & 0\,1\,0\,0\,1 \\ + & 1\,1\,1\,0\,0 \\ \hline & \boxed{1}\,0\,0\,1\,0\,1 \end{array}$$

Report

Le résultat est positif

$(00101)_2 = (5)_{10}$

$$\begin{array}{r} -9 \\ -4 \\ \hline -13 \end{array} \qquad \begin{array}{rr} & 1\,0\,1\,1\,1 \\ + & 1\,1\,1\,0\,0 \\ \hline & \boxed{1}\,1\,0\,0\,1\,1 \end{array}$$

Report

Le résultat est négatif :

Résultat = - CA2 (10011)

= -(01101)

= - 13

$$\begin{array}{r} -9 \\ +9 \\ \hline +0 \end{array} \qquad \begin{array}{rr} & 1\,0\,1\,1\,1 \\ + & 0\,1\,0\,0\,1 \\ \hline & \boxed{1}\,0\,0\,0\,0\,0 \end{array}$$

Report

Le résultat est positif

$(00000)_2 = (0)_{10}$

La retenue et le débordement

- On dit qu'il y a une retenue si une opération arithmétique génère un report .
- On dit qu'il y a un débordement (Over Flow) ou dépassement de capacité: si le résultat de l'opération sur n bits et faux .
  - Le nombre de bits utilisés est insuffisant pour contenir le résultat
  - Autrement dit le résultat dépasse l'intervalle des valeurs sur les n bits utilisés.

Cas de débordement

	0	1				
+ 9		0	1	0	0	1
+ 8	+	0	1	0	0	0
+ 17		1	0	0	0	1

Négatif

	1	0				
- 9		1	0	1	1	1
- 8	+	1	1	0	0	0
- 17		0	1	0	1	1

Positif

- Nous avons un débordement si la somme de deux nombres positifs donne un nombre négatif .
- Ou la somme de deux nombres négatifs donne un Nombre positif
- Il y a jamais un débordement si les deux nombres sont de signes différents.

Fin de la quatrième séance