Représentation des entiers relatifs

1ère NSI

Plan du cours

Addition binaire

- L'addition binaire répond aux règles suivantes :

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0 + 0 = 0 \qquad 1 + 0 = 0 + 1 = 1 \qquad 1 + 1 = 10$$

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0 + 0 = 0 \qquad 1 + 0 = 0 + 1 = 1 \qquad 1 + 1 = 10$$

- Exemple (sans retenue) :

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

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0+0=0 \qquad 1+0=0+1=1 \qquad 1+1=10$$

- Exemple (sans retenue) :

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

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0+0=0 \qquad 1+0=0+1=1 \qquad 1+1=10$$

- Exemple (sans retenue) :

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

- Exemple (avec retenue) :

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

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0 + 0 = 0 \qquad 1 + 0 = 0 + 1 = 1 \qquad 1 + 1 = 10$$

- Exemple (sans retenue) :

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

- Exemple (avec retenue) :

$$\begin{array}{ccccc} \color{red}{1} & & & \color{red}{1} & \\ & 1 & 1 & 0 & 1 \\ + & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$$

Addition binaire

- L'addition binaire répond aux règles suivantes :

$$0+0=0 \qquad 1+0=0+1=1 \qquad 1+1=10$$

- Exemple (sans retenue) :

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

- Exemple (avec retenue) :

$$\begin{array}{ccccc} \color{red}{1} & & & \color{red}{1} & \\ & 1 & 1 & 0 & 1 \\ + & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$$

$\boxed{\text{☞ } \textit{Exercices 1,2}}$

Overflow (dépassement)

Si les nombres sont stockés sur un **nombre fixe de bits**, il peut se produire un **dépassement** (*overflow* en anglais).

Overflow (dépassement)

Si les nombres sont stockés sur un **nombre fixe de bits**, il peut se produire un **dépassement** (*overflow* en anglais).

- Exemple (sur 4 bits) :

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

Overflow (dépassement)

Si les nombres sont stockés sur un **nombre fixe de bits**, il peut se produire un **dépassement** (*overflow* en anglais).

- Exemple (sur 4 bits) :

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

- Le 5ème bit n'est pas pris en compte

Overflow (dépassement)

Si les nombres sont stockés sur un **nombre fixe de bits**, il peut se produire un **dépassement** (*overflow* en anglais).

- Exemple (sur 4 bits) :

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

- Le 5ème bit n'est pas pris en compte
- Équivalent à : $13 + 9 = 6$ (!)

Overflow (dépassement)

Si les nombres sont stockés sur un **nombre fixe de bits**, il peut se produire un **dépassement** (*overflow* en anglais).

- Exemple (sur 4 bits) :

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

- Le 5ème bit n'est pas pris en compte
- Équivalent à : $13 + 9 = 6$ (!)

$\boxed{☞ \textit{Exercice 3}}$

Cas particulier

- Exemple (sur 8 bits) :

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

Cas particulier

- Exemple (sur 8 bits) :

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

Cas particulier

- Exemple (sur 8 bits) :

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

- Équivalent à $37 + 219 = 0$

Cas particulier

- Exemple (sur 8 bits) :

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

- Équivalent à $37 + 219 = 0$
- 219 équivaut donc à -37 (car $37 + (-37) = 0$)...

Cas particulier

- Exemple (sur 8 bits) :

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

- Équivalent à 37 + 219 = 0
- 219 équivaut donc à -37 (car 37 + (-37) = 0)...
- ... ou bien 37 équivaut à -219 (car (-219) + 219 = 0)

Cas particulier

- Exemple (sur 8 bits) : $$\begin{array}{ccccccccc} & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ + & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array}$$
- Équivalent à $37 + 219 = 0$
- 219 équivaut donc à -37 (car 37 + (-37) = 0)...
- ... ou bien 37 équivaut à -219 (car (-219) + 219 = 0)

Règle

*Dans la représentation des entiers relatifs, le bit de poids fort est un **bit de signe** :*

- *0 indique un entier positif ou nul*
- *1 indique un entier négatif*

Cas particulier

- Exemple (sur 8 bits) :

	0	0	1	0	0	1	0	1
+	1	1	0	1	1	0	1	1
1	0	0	0	0	0	0	0	0

- Équivalent à 37 + 219 = 0
- 219 équivaut donc à -37 (car 37 + (-37) = 0)...
- ... ou bien 37 équivaut à -219 (car (-219) + 219 = 0)

Règle

Dans la représentation des entiers relatifs, le bit de poids fort est un ***bit de signe*** *:*

- *0 indique un entier positif ou nul*
- *1 indique un entier négatif*

☞ *D'après la règle, c'est donc 219 qui équivaut à -37*

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais 37 + 219 = 256 = 2^8
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais 37 + 219 = 256 = 2^8
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Méthode : complément à 2

Dans la représentation d'un entier relatif sur n bits, l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) est représenté par le nombre $2^n - a$.

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais $37 + 219 = 256 = 2^8$
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Méthode : complément à 2

Dans la représentation d'un entier relatif sur n bits, l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) est représenté par le nombre $2^n - a$.

- Exemple : opposé de 55 (sur 8 bits)

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais 37 + 219 = 256 = 2^8
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Méthode : complément à 2

Dans la représentation d'un entier relatif sur n bits, l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) est représenté par le nombre $2^n - a$.

- Exemple : opposé de 55 (sur 8 bits)
  - On calcule $2^8 - 55 = 256 - 55 = 201$

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais 37 + 219 = 256 = 2^8
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Méthode : complément à 2

Dans la représentation d'un entier relatif sur n bits, l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) est représenté par le nombre $2^n - a$.

- Exemple : opposé de 55 (sur 8 bits)
  - On calcule $2^8 - 55 = 256 - 55 = 201$
  - En binaire, $201 = (1100\ 1001)_2$

Complément à 2

- Naturellement, $37 + 219 \neq 0$ mais $37 + 219 = 256 = 2^8$
- Ainsi, 256 - 37 = 219 traduit le fait que 219 représente le nombre -37 sur 8 bits

Méthode : complément à 2

Dans la représentation d'un entier relatif sur n bits, l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) est représenté par le nombre $2^n - a$.

- Exemple : opposé de 55 (sur 8 bits)
  - On calcule $2^8 - 55 = 256 - 55 = 201$
  - En binaire, $201 = (1100\ 1001)_2$
  - -55 est donc représenté par $(1100\ 1001)_2$

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1}-1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)
  - $55 = (0011\ 0111)_2$

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)
  - $55 = (0011\ 0111)_2$
  - On inverse tous les bits : $(1100\ 1000)_2$

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)
  - $55 = (0011\ 0111)_2$
  - On inverse tous les bits : $(1100\ 1000)_2$
  - On ajoute 1 : $(1100\ 1001)_2$

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1}-1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)
  - $55 = (0011\ 0111)_2$
  - On inverse tous les bits : $(1100\ 1000)_2$
  - On ajoute 1 : $(1100\ 1001)_2$
- -55 est donc représenté par $(1100\ 1001)_2$

Complément à 2

Algorithme : complément à 2

Dans la représentation d'un entier relatif sur n bits, la représentation de l'opposé d'un entier naturel a (inférieur à $2^{n-1} - 1$) s'obtient de la façon suivante :

- On donne la représentation de a en binaire, sur n bits
- On **inverse** tous les bits (0 devient 1, 1 devient 0)
- On ajoute 1 au résultat

- Exemple : opposé de 55 (sur 8 bits)
  - $55 = (0011\ 0111)_2$
  - On inverse tous les bits : $(1100\ 1000)_2$
  - On ajoute 1 : $(1100\ 1001)_2$
- -55 est donc représenté par $(1100\ 1001)_2$

☞ *Exercices 4,5,6,7*