# Les ordinateurs calculent-ils juste ?

Christophe TROESTLER
Christophe.Troestler@umh.ac.be

Institut de Mathématique
Université de Mons-Hainaut
Mons, Belgium

# Les ordinateurs calculent-ils juste ?

1. Représentation binaire des nombres entiers
   — *deux symboles en valent bien dix !*
2. Codage des nombres entiers par une machine
   — *ou comment les ordinateurs parlent de* 0, 1, 2, 3,...
3. L'explosion d'Ariane 5
   — *quelques millions de dollars s'envoient en l'air...*
4. Codage machine des nombres réels
   — *ou le règne des nombres avec virgule,* 2,7, $\sqrt{2}$, $\pi$,...
5. Le défaut des missiles « patriot »
   — *quand une petite erreur a de graves conséquences*
6. L'arithmétique mise à mal
   — *où l'on découvre que les écoliers calculent mieux...*

# Les ordinateurs calculent-ils juste ?

1. Représentation binaire des nombres entiers
   — *deux symboles en valent bien dix !*
2. Codage des nombres entiers par une machine
   — *ou comment les ordinateurs parlent de* 0, 1, 2, 3,...
3. L'explosion d'Ariane 5
   — *quelques millions de dollars s'envoient en l'air...*
4. Codage machine des nombres réels
   — *ou le règne des nombres avec virgule,* 2,7, $\sqrt{2}$, $\pi$,...
5. Le défaut des missiles « patriot »
   — *quand une petite erreur a de graves conséquences*
6. L'arithmétique mise à mal
   — *où l'on découvre que les écoliers calculent mieux...*

# Les ordinateurs calculent-ils juste ?

1. Représentation binaire des nombres entiers
   — *deux symboles en valent bien dix !*
2. Codage des nombres entiers par une machine
   — *ou comment les ordinateurs parlent de* 0, 1, 2, 3,...
3. L'explosion d'Ariane 5
   — *quelques millions de dollars s'envoient en l'air...*
4. Codage machine des nombres réels
   — *ou le règne des nombres avec virgule,* 2,7, $\sqrt{2}$, $\pi$,...
5. Le défaut des missiles « patriot »
   — *quand une petite erreur a de graves conséquences*
6. L'arithmétique mise à mal
   — *où l'on découvre que les écoliers calculent mieux...*

# Les ordinateurs calculent-ils juste ?

1. Représentation binaire des nombres entiers
   — *deux symboles en valent bien dix !*
2. Codage des nombres entiers par une machine
   — *ou comment les ordinateurs parlent de* 0, 1, 2, 3,...
3. L'explosion d'Ariane 5
   — *quelques millions de dollars s'envoient en l'air...*
4. Codage machine des nombres réels
   — *ou le règne des nombres avec virgule,* 2,7, $\sqrt{2}$, $\pi$,...
5. Le défaut des missiles « patriot »
   — *quand une petite erreur a de graves conséquences*
6. L'arithmétique mise à mal
   — *où l'on découvre que les écoliers calculent mieux...*

# Les ordinateurs calculent-ils juste ?

1. Représentation binaire des nombres entiers
   — *deux symboles en valent bien dix !*
2. Codage des nombres entiers par une machine
   — *ou comment les ordinateurs parlent de* 0, 1, 2, 3,...
3. L'explosion d'Ariane 5
   — *quelques millions de dollars s'envoient en l'air...*
4. Codage machine des nombres réels
   — *ou le règne des nombres avec virgule,* 2,7, $\sqrt{2}$, $\pi$,...
5. Le défaut des missiles « patriot »
   — *quand une petite erreur a de graves conséquences*
6. L'arithmétique mise à mal
   — *où l'on découvre que les écoliers calculent mieux...*

# Les ordinateurs calculent-ils juste ?

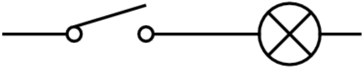
1. Représentation binaire des nombres entiers
   — *deux symboles en valent bien dix !*
2. Codage des nombres entiers par une machine
   — *ou comment les ordinateurs parlent de* 0, 1, 2, 3,...
3. L'explosion d'Ariane 5
   — *quelques millions de dollars s'envoient en l'air...*
4. Codage machine des nombres réels
   — *ou le règne des nombres avec virgule,* 2,7, $\sqrt{2}$, $\pi$,...
5. Le défaut des missiles « patriot »
   — *quand une petite erreur a de graves conséquences*
6. L'arithmétique mise à mal
   — *où l'on découvre que les écoliers calculent mieux...*

# Représentation binaire des nombres entiers

## Vous avez dit binaire ?

Binaire veut dire qu'il y a deux états que nous noterons 0 et 1. Pour un ordinateur, cela correspond respectivement à l'absence et au passage du courant.

0



1

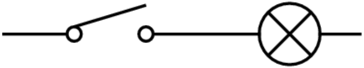
Représenter les nombres avec seulement deux symboles ? Comment écrivons nous habituellement les nombres avec les dix symboles 0, 1, 2,..., 9 ?

# Représentation binaire des nombres entiers

## Vous avez dit binaire ?

Binaire veut dire qu'il y a deux états que nous noterons 0 et 1. Pour un ordinateur, cela correspond respectivement à l'absence et au passage du courant.

0



1

Représenter les nombres avec seulement deux symboles ? Comment écrivons nous habituellement les nombres avec les dix symboles 0, 1, 2,..., 9 ?

# Écriture décimale des nombres entiers

$12 = 2$ unités et $1$ dizaine $= 1 \times 10 + 2$.

$$
\begin{aligned}
1256 &= 1 \times 1000 + 2 \times 100 + 5 \times 10 + 6 \\
&= 1 \cdot 10^3 + 2 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0
\end{aligned}
\quad \rightarrow
$$

| 1 | 2 | 5 | 6 |
|---|---|---|---|
| ↑ | ↑ | ↑ | ↑ |
| 1000 | 100 | 10 | 1 |

Pourquoi 10, 100, 1000,... ? C'est parce qu'on a 10 symboles !

$\boxed{0}, \boxed{1}, \dots, \boxed{9}$ → utiliser la *position* pour continuer

# Écriture décimale des nombres entiers

$12 = 2$ unités et $1$ dizaine $= 1 \times 10 + 2$.

$$\begin{aligned} 1256 &= 1 \times 1000 + 2 \times 100 + 5 \times 10 + 6 \\ &= 1 \cdot 10^3 + 2 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0 \end{aligned} \quad \to \quad \begin{array}{|c|c|c|c|} \hline 1 & 2 & 5 & 6 \\ \hline \end{array}$$

$$\begin{array}{cccc} \uparrow & \uparrow & \uparrow & \uparrow \\ 1000 & 100 & 10 & 1 \end{array}$$

Pourquoi 10, 100, 1000,... ? C'est parce qu'on a 10 symboles !

$\boxed{0}, \boxed{1}, \ldots, \boxed{9}$ $\Rightarrow$ utiliser la *position* pour continuer : $\boxed{1}\boxed{0}$

$\Rightarrow$ on peut écrire $\boxed{0}\boxed{0}, \ldots, \boxed{9}\boxed{9}$

càd 100 nombres car : $\boxed{\phantom{0}}\boxed{\phantom{0}}$

dix possibilités

dix possibilités

# Écriture décimale des nombres entiers

$12 = 2$ unités et $1$ dizaine $= 1 \times 10 + 2$.

$$1256 = 1 \times 1000 + 2 \times 100 + 5 \times 10 + 6 \rightarrow \boxed{1}\boxed{2}\boxed{5}\boxed{6}$$

$$= 1 \cdot 10^3 + 2 \cdot 10^2 + 5 \cdot 10^1 + 6 \cdot 10^0$$

$$\begin{array}{cccc} 1 & 2 & 5 & 6 \\ \uparrow & \uparrow & \uparrow & \uparrow \\ 1000 & 100 & 10 & 1 \end{array}$$

Pourquoi 10, 100, 1000,... ? C'est parce qu'on a 10 symboles !

$\boxed{0}, \boxed{1}, \ldots, \boxed{9} \Rightarrow$ utiliser la *position* pour continuer : $\boxed{1}\boxed{0}$

$\Rightarrow$ on peut écrire $\boxed{0}\boxed{0}, \ldots, \boxed{9}\boxed{9}$

càd 100 nombres car : $\boxed{\phantom{0}}\boxed{\phantom{0}}$

dix possibilités

dix possibilités

# Écriture binaire des nombres entiers

On ne dispose que de 0 et 1.

$$\boxed{0}\,,\ \boxed{1} \Rightarrow \text{utiliser la } \textit{position} \text{ pour continuer : } \boxed{1\,|\,0} = 2$$

$$\Rightarrow \boxed{0\,|\,0}\,,\ \boxed{0\,|\,1}\,,\ \underset{\substack{\uparrow\ \ \uparrow\\ 2\ \ 1}}{\boxed{1\,|\,0}}\,,\ \underset{\substack{\uparrow\ \ \uparrow\\ 2\ \ 1}}{\boxed{1\,|\,1}} = 1 \times 2 + 1 = 3$$

$$\Rightarrow \boxed{1\,|\,0\,|\,0} = 4, \ldots$$

Chaque fois qu'on ajoute un *digit* on multiplie par 2 au lieu de 10 :

$$\underset{\substack{\uparrow\ \ \uparrow\ \ \uparrow\ \ \uparrow\\ 8\ \ 4\ \ 2\ \ 1}}{\boxed{1\,|\,1\,|\,0\,|\,1}} \quad \rightarrow \quad 1101 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 = 13$$

# Écriture binaire des nombres entiers

On ne dispose que de 0 et 1.

$$\boxed{0}\,,\ \boxed{1} \Rightarrow \text{utiliser la } \textit{position} \text{ pour continuer : } \boxed{1\,|\,0} = 2$$

$$\Rightarrow \boxed{0\,|\,0}\,,\ \boxed{0\,|\,1}\,,\ \underset{\substack{\uparrow\ \uparrow\\ 2\ \ 1}}{\boxed{1\,|\,0}}\,,\ \underset{\substack{\uparrow\ \uparrow\\ 2\ \ 1}}{\boxed{1\,|\,1}} = 1 \times 2 + 1 = 3$$

$$\Rightarrow \boxed{1\,|\,0\,|\,0} = 4, \ldots$$

Chaque fois qu'on ajoute un *digit* on multiplie par 2 au lieu de 10 :

$$\underset{\substack{\uparrow\ \uparrow\ \uparrow\ \uparrow\\ 8\ \ 4\ \ 2\ \ 1}}{\boxed{1\,|\,1\,|\,0\,|\,1}} \quad \to \quad 1101 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 = 13$$

# Comment fait-on des calculs ?

Les principes de calcul avec les représentations binaires sont les mêmes qu'avec les représentations décimales. Comparons l'addition de $13 = 1101$ et de $9 = 1001$ en décimal et en binaire :

| *décimal* | *binaire* |
|---|---|
| $\begin{array}{cc} 1 & \\ 1 & 3 \\ & 9 \\ \hline 2 & 2 \end{array}$ | $\begin{array}{ccccc} & & & 1 & \\ & 1 & 1 & 0 & 1 \\ & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$ |

De la même manière on peut faire des soustractions, des multiplications, des divisions,...

$\Rightarrow$ Une machine peut faire des calculs tout à fait exacts...

# Comment fait-on des calculs ?

Les principes de calcul avec les représentations binaires sont les mêmes qu'avec les représentations décimales. Comparons l'addition de $13 = 1101$ et de $9 = 1001$ en décimal et en binaire :

*décimal*

$$\begin{array}{cc} 1 & \\ 1 & 3 \\ & 9 \\ \hline 2 & 2 \end{array}$$

*binaire*

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

De la même manière on peut faire des soustractions, des multiplications, des divisions,...

$\Rightarrow$ Une machine peut faire des calculs tout à fait exacts...

# Comment fait-on des calculs ?

Les principes de calcul avec les représentations binaires sont les mêmes qu'avec les représentations décimales. Comparons l'addition de $13 = 1101$ et de $9 = 1001$ en décimal et en binaire :

| *décimal* | *binaire* |
|---|---|
| $$\begin{array}{cc} 1 & \\ 1 & 3 \\ & 9 \\ \hline 2 & 2 \end{array}$$ | $$\begin{array}{ccccc} & & & 1 & \\ & 1 & 1 & 0 & 1 \\ & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$$ |

De la même manière on peut faire des soustractions, des multiplications, des divisions,...

$\Rightarrow$ Une machine peut faire des calculs tout à fait exacts...

# Comment fait-on des calculs ?

Les principes de calcul avec les représentations binaires sont les mêmes qu'avec les représentations décimales. Comparons l'addition de $13 = 1101$ et de $9 = 1001$ en décimal et en binaire :

*décimal*

$$\begin{array}{cc} 1 & \\ 1 & 3 \\ & 9 \\ \hline 2 & 2 \end{array}$$

*binaire*

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

De la même manière on peut faire des soustractions, des multiplications, des divisions,...

$\Rightarrow$ Une machine peut faire des calculs tout à fait exacts...

# Comment fait-on des calculs ?

Les principes de calcul avec les représentations binaires sont les mêmes qu'avec les représentations décimales. Comparons l'addition de $13 = 1101$ et de $9 = 1001$ en décimal et en binaire :

| *décimal* | *binaire* |
| --- | --- |
| $\begin{array}{cc} 1 & \\ 1 & 3 \\ & 9 \\ \hline 2 & 2 \end{array}$ | $\begin{array}{ccccc} & & & 1 & \\ & 1 & 1 & 0 & 1 \\ & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$ |

De la même manière on peut faire des soustractions, des multiplications, des divisions,...

$\Rightarrow$ Une machine peut faire des calculs tout à fait exacts...

# Comment fait-on des calculs ?

Les principes de calcul avec les représentations binaires sont les mêmes qu'avec les représentations décimales. Comparons l'addition de $13 = 1101$ et de $9 = 1001$ en décimal et en binaire :

*décimal*

$$\begin{array}{cc}
1 & \\
1 & 3 \\
 & 9 \\
\hline
2 & 2
\end{array}$$

*binaire*

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

De la même manière on peut faire des soustractions, des multiplications, des divisions,...

$\Rightarrow$ Une machine peut faire des calculs tout à fait exacts...

# Comment fait-on des calculs ?

Les principes de calcul avec les représentations binaires sont les mêmes qu'avec les représentations décimales. Comparons l'addition de $13 = 1101$ et de $9 = 1001$ en décimal et en binaire :

| *décimal* | *binaire* |
|---|---|
| $$\begin{array}{cc} 1 & \\ 1 & 3 \\ & 9 \\ \hline 2 & 2 \end{array}$$ | $$\begin{array}{ccccc} & & & 1 & \\ & 1 & 1 & 0 & 1 \\ & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$$ |

De la même manière on peut faire des soustractions, des multiplications, des divisions,...

$\Rightarrow$ Une machine peut faire des calculs tout à fait exacts...

# Comment fait-on des calculs ?

Les principes de calcul avec les représentations binaires sont les mêmes qu'avec les représentations décimales. Comparons l'addition de $13 = 1101$ et de $9 = 1001$ en décimal et en binaire :

| *décimal* | *binaire* |
|---|---|
| $\begin{array}{cc} 1 & \\ 1 & 3 \\ & 9 \\ \hline 2 & 2 \end{array}$ | $\begin{array}{ccccc} & & & 1 & \\ & 1 & 1 & 0 & 1 \\ & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$ |

De la même manière on peut faire des soustractions, des multiplications, des divisions,...

$\Rightarrow$ Une machine peut faire des calculs tout à fait exacts...

# Comment fait-on des calculs ?

Les principes de calcul avec les représentations binaires sont les mêmes qu'avec les représentations décimales. Comparons l'addition de $13 = 1101$ et de $9 = 1001$ en décimal et en binaire :

| *décimal* | *binaire* |
|---|---|
| $\begin{array}{rr} 1 & \\ 1 & 3 \\ & 9 \\ \hline 2 & 2 \end{array}$ | $\begin{array}{rrrrr} & & & 1 & \\ & 1 & 1 & 0 & 1 \\ & 1 & 0 & 0 & 1 \\ \hline 1 & 0 & 1 & 1 & 0 \end{array}$ |

De la même manière on peut faire des soustractions, des multiplications, des divisions,...

$\Rightarrow$ Une machine peut faire des calculs tout à fait exacts...

# Les nombres entiers machine (1/2)

Une machine ne possède que des ressources *limitées*
$\Rightarrow$ on ne peut pas stocker de très grands nombres.

Si on dispose de 8 *bits* :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| $\pm$ | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | $2^6$ | | | $2^3$ | $2^2$ | $2$ | $1$ |

$\Rightarrow$ nombres de $-1111111 = -127$ à $1111111 = 127 = 2^7 - 1$.

# Les nombres entiers machine (1/2)

Une machine ne possède que des ressources *limitées*
$\Rightarrow$ on ne peut pas stocker de très grands nombres.

Si on dispose de 8 *bits* :

$$\begin{array}{|c|c|c|c|c|c|c|c|}\hline \pm & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ \hline \end{array}$$

$$\begin{array}{cccccccc} & \uparrow & & & \uparrow & \uparrow & \uparrow & \uparrow \\ & 2^6 & & & 2^3 & 2^2 & 2 & 1 \end{array}$$

$\Rightarrow$ nombres de $-1111111 = -127$ à $1111111 = 127 = 2^7 - 1$.

# Les nombres entiers machine (2/2)

| Type | C | bits | nombres accessibles |
|---|---|---|---|
| petits entiers | `int` | 16 | $-32767$ à $32767$ |
| entiers longs | `long int` | 32 | $-2147483647$ à $2147483647$ |

Que se passe-t-il lorsqu'un entier est trop grand pour être stocké ? Par exemple, si on veut stocker 130 avec 8 bits ?

| + | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | ↑ $2^6$ | | | | | | ↑ 1 |

Ce n'est pas possible : il y a *dépassement de capacité* (*overflow*). Dans ce cas, un « drapeau » est levé et le programme doit suivre une route de calcul différente...

# Les nombres entiers machine (2/2)

| Type | C | bits | nombres accessibles |
|---|---|---|---|
| petits entiers | `int` | 16 | $-32767$ à $32767$ |
| entiers longs | `long int` | 32 | $-2147483647$ à $2147483647$ |

Que se passe-t-il lorsqu'un entier est trop grand pour être stocké ? Par exemple, si on veut stocker 130 avec 8 bits ?

| + | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | ↑ $2^6$ | | | | | | ↑ $1$ |

Ce n'est pas possible : il y a *dépassement de capacité* (*overflow*). Dans ce cas, un « drapeau » est levé et le programme doit suivre une route de calcul différente...

# Les nombres entiers machine (2/2)

| Type | C | bits | nombres accessibles |
|---|---|---|---|
| petits entiers | `int` | 16 | $-32767$ à $32767$ |
| entiers longs | `long int` | 32 | $-2147483647$ à $2147483647$ |

Que se passe-t-il lorsqu'un entier est trop grand pour être stocké ? Par exemple, si on veut stocker 130 avec 8 bits ?

| + | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | | ↑ $2^6$ | | | | | | ↑ 1 |

Ce n'est pas possible : il y a *dépassement de capacité* (*overflow*). Dans ce cas, un « drapeau » est levé et le programme doit suivre une route de calcul différente...

# L'explosion d'Ariane 5

On June 4, 1996 an unmanned Ariane 5 rocket launched by the European Space Agency exploded just forty seconds after its lift-off from Kourou, French Guiana. The rocket was on its first voyage, after a decade of development costing $ 7 billion. The destroyed rocket and its cargo were valued at $ 500 million.

The failure of the Ariane 501 was caused by the complete loss of guidance and attitude information 37 seconds after start of the main engine ignition sequence (30 seconds after lift-off). This loss of information was due to specification and design errors in the software of the inertial reference system.

The internal SRI software exception was caused during execution of a data conversion from 64-bit floating point to 16-bit signed integer value. The floating point number which was converted had a value greater than what could be represented by a 16-bit signed integer.

1. Entiers en binaire
   - Écriture décimale
   - Écriture binaire
   - Calculs
2. Entiers machine
3. Ariane 5
4. **Réels en binaire**
5. Missiles « patriot »
6. L'arithmétique mise à mal

# Codage des nombres réels

## Représentation binaire

Décimal :

$$0{,}1 = \frac{1}{10}\,; \qquad 1{,}25 = 1 + 2\frac{1}{10} + 5\frac{1}{100} = \begin{array}{|c|c|c|} \hline 1, & 2 & 5 \\ \hline \end{array}$$

$$\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ 1 & \frac{1}{10} & \frac{1}{100} \end{array}$$

Binaire :

$$0{,}1 = \frac{1}{2}\,; \qquad 1{,}01 = 1 + 0\frac{1}{2} + 1\frac{1}{4} = \begin{array}{|c|c|c|} \hline 1, & 0 & 1 \\ \hline \end{array} = 1{,}25$$

$$\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ 1 & \frac{1}{2} & \frac{1}{4} \end{array}$$

# Codage des nombres réels

## Représentation binaire

Décimal :

$$0{,}1 = \frac{1}{10}\,; \qquad 1{,}25 = 1 + 2\frac{1}{10} + 5\frac{1}{100} = \begin{array}{|c|c|c|} \hline 1, & 2 & 5 \\ \hline \end{array}$$

$$\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ 1 & \frac{1}{10} & \frac{1}{100} \end{array}$$

Binaire :

$$0{,}1 = \frac{1}{2}\,; \qquad 1{,}01 = 1 + 0\frac{1}{2} + 1\frac{1}{4} = \begin{array}{|c|c|c|} \hline 1, & 0 & 1 \\ \hline \end{array} = 1{,}25$$

$$\begin{array}{ccc} \uparrow & \uparrow & \uparrow \\ 1 & \frac{1}{2} & \frac{1}{4} \end{array}$$

# Nombres en « virgule flottante »

## Notation scientifique

Exemple sur 8 bits :

$$+\ 1,\ 0\ 0\ 1\ 0\ 0\ 1$$

$$+\ 0,\ \underbrace{0\ 0\ 0\ 0\ 0\ 0}_{7}\ 1\ 0\ 0\ 1\ 0\ 0\ 1$$

Typiquement :

| Précision | C | bits |
|---|---|---|
| simple | `float` | 24 |
| double | `double` | 53 |

# Nombres en « virgule flottante »

## Notation scientifique

Exemple sur 8 bits :

| | | | | | | |
|---|---|---|---|---|---|---|
| + | 1, | 0 | 0 | 1 | 0 | 0 | 1 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| + | 0, | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

$$\xrightarrow[7]{\qquad\qquad\qquad}$$

Typiquement :

| Précision | C | bits |
|---|---|---|
| simple | `float` | 24 |
| double | `double` | 53 |

# Nombres en « virgule flottante »

## Notation scientifique

Exemple sur 8 bits :

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| + | 1, | 0 | 0 | 1 | 0 | 0 | 1 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| + | 0, | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 |

$$\xrightarrow[7]{\qquad\qquad\qquad}$$

Typiquement :

| Précision | C | bits |
|---|---|---|
| simple | `float` | 24 |
| double | `double` | 53 |

# Le défaut des missiles « patriot » (1/2)

On February 25, 1991, during the Gulf War, an American Patriot Missile battery in Dharan, Saudi Arabia, failed to track and intercept an incoming Iraqi Scud missile. The Scud struck an American Army barracks, killing 28 soldiers and injuring around 100 other people.

It turns out that the cause was an inaccurate calculation of the time since boot due to computer arithmetic errors.

Specifically, the time in tenths of second as measured by the system's internal clock was multiplied by 1/10 to produce the time in seconds. This calculation was performed using a 24 bit fixed point register [...] the Patriot battery had been up around 100 hours [...]

# Le défaut des missiles « patriot » (2/2)

$$\frac{1}{10} = 0{,}000110011001100110011001100\ldots =$$

| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0, | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

$$+\,0{,}00000000000000000000000011001100\ldots$$

L'erreur à chaque dixième de seconde est de

$$0{,}00000000000000000000000011001100\ldots \approx 0{,}000000095$$

et donc, après 100 heures,

$$0{,}000000095 \times 100 \times 60 \times 60 \times 10 = 0{,}34.$$

Un missile Scud parcourt environ 1 676 mètres par seconde
$\Rightarrow$ en 0,34 secondes il parcourt environ 570 mètres !

# Le défaut des missiles « patriot » (2/2)

$$\frac{1}{10} = 0{,}0001100110011001100110011001100\ldots =$$

| | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0, | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

$$+\ 0{,}0000000000000000000000011001100\ldots$$

L'erreur à chaque dixième de seconde est de

$$0{,}0000000000000000000000011001100\ldots \approx 0{,}000000095$$

et donc, après 100 heures,

$$0{,}000000095 \times 100 \times 60 \times 60 \times 10 = 0{,}34.$$

Un missile Scud parcourt environ 1 676 mètres par seconde
$\Rightarrow$ en 0,34 secondes il parcourt environ 570 mètres !

# Le défaut des missiles « patriot » (2/2)

$$\frac{1}{10} = 0{,}0001100110011001100110011001100\ldots =$$

| | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0, | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

$$+\,0{,}0000000000000000000000011001100\ldots$$

L'erreur à chaque dixième de seconde est de

$$0{,}0000000000000000000000011001100\ldots \approx 0{,}000000095$$

et donc, après 100 heures,

$$0{,}000000095 \times 100 \times 60 \times 60 \times 10 = 0{,}34.$$

Un missile Scud parcourt environ 1 676 mètres par seconde

$\Rightarrow$ en 0,34 secondes il parcourt environ 570 mètres !

# Le défaut des missiles « patriot » (2/2)

$$\frac{1}{10} = 0{,}0001100110011001100110011001100\ldots =$$

| | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0, | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |

$$+\,0{,}0000000000000000000000011001100\ldots$$

L'erreur à chaque dixième de seconde est de

$$0{,}0000000000000000000000011001100\ldots \approx 0{,}000000095$$

et donc, après 100 heures,

$$0{,}000000095 \times 100 \times 60 \times 60 \times 10 = 0{,}34.$$

Un missile Scud parcourt environ 1 676 mètres par seconde
$\Rightarrow$ en 0,34 secondes il parcourt environ 570 mètres !

# L'arithmétique mise à mal (1/2)

En arithmétique :

$$\frac{(1+x)-1}{x} = 1$$

Si on fait ce calcul sur un ordinateur :

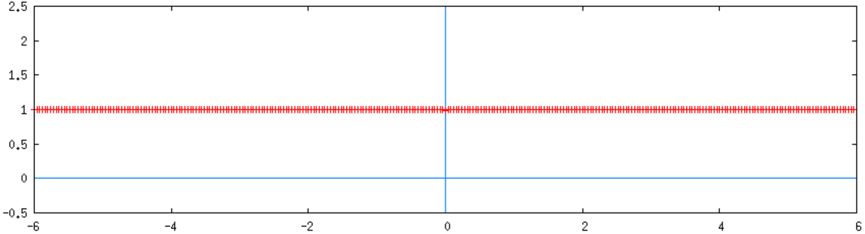
avec zoom horizontal de $10^{-16}$.

# L'arithmétique mise à mal (1/2)

En arithmétique :

$$\frac{(1+x)-1}{x} = 1$$

Si on fait ce calcul sur un ordinateur :



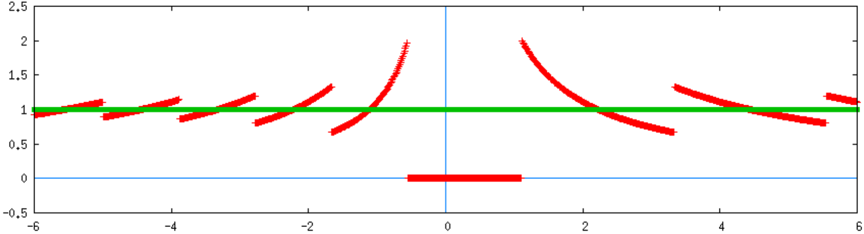

avec zoom horizontal de $10^{-16}$.

# L'arithmétique mise à mal (1/2)

En arithmétique :

$$\frac{(1+x)-1}{x} = 1$$

Si on fait ce calcul sur un ordinateur :




avec zoom horizontal de $10^{-16}$.

# L'arithmétique mise à mal (2/2)

Exemple avec 8 bits :

$$1 + x = \boxed{+}\boxed{1,}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}$$

$$\boxed{+}\ 0,\ \underbrace{0\ 0\ 0\ 0\ 0\ 0\ 0}_{\longrightarrow\ 8}\ \boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{0}\boxed{0}\boxed{1}$$

$$= \boxed{+}\boxed{1,}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0} \qquad \text{pour la machine}$$

$$= 1$$

D'où,

$$\frac{(1+x)-1}{x} = \frac{1-1}{x} = 0$$

# L'arithmétique mise à mal (2/2)

Exemple avec 8 bits :

$$1 + x = \boxed{+}\boxed{1,}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}$$

$$\boxed{+}\ 0,\ \underbrace{0\ 0\ 0\ 0\ 0\ 0\ 0}_{8}\ \boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{0}\boxed{0}\boxed{1}$$

$$= \boxed{+}\boxed{1,}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0} \qquad \text{pour la machine}$$

$$= 1$$

D'où,

$$\frac{(1+x)-1}{x} = \frac{1-1}{x} = 0$$

# L'arithmétique mise à mal (2/2)

Exemple avec 8 bits :

$$1 + x = \boxed{+}\,\boxed{1,}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}$$

$$\boxed{+}\; 0,\; 0\; 0\; 0\; 0\; 0\; 0\; 0\; \boxed{1}\boxed{0}\boxed{0}\boxed{1}\boxed{0}\boxed{0}\boxed{1}$$

$$\xrightarrow[8]{\qquad\qquad\qquad}$$

$$= \boxed{+}\,\boxed{1,}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0}\boxed{0} \qquad \text{pour la machine}$$

$$= 1$$

D'où,

$$\frac{(1 + x) - 1}{x} = \frac{1 - 1}{x} = 0$$

# Quelques liens...

## Pour en savoir plus

### Ariane 5

http://www.math.ufl.edu/~cws/3114/ariane-siam.html
http://www.esrin.esa.it/htdocs/tidc/Press/Press96/press33.html

### Missiles « patriot »

http://www.math.psu.edu/dna/disasters/patriot.html
http://catless.ncl.ac.uk/Risks/20.85.html#subj2.1

### Sur les calculs « avec virgule »

http://www.lahey.com/float.htm
http:
//wwwzenger.informatik.tu-muenchen.de/persons/huckle/bugse.html