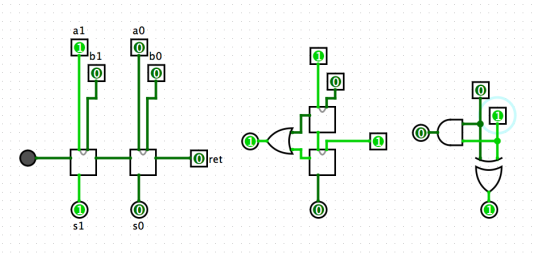
Principes de fonctionnement des machines binaires

2020–2021

Matthieu Picantin

- numération et arithmétique
- numération et arithmétique en machine
- numérisation et codage (texte, images)
- compression, cryptographie, contrôle derreur
- logique et calcul propositionnel
- circuits numériques



Système d'Avižienis pour une base $b > 2$

- exactement $2n+1$ chiffres $\overline{n}, \ldots, 0, \ldots, n$ avec $\frac{b}{2} < n < b$
- système *complet* : tout entier est représentable
- système *redondant* : certains entiers ont plusieurs représentations
- nouvelle méthode d'addition de $(a_{p-1} \cdots a_0)_b$ et $(c_{p-1} \cdots c_0)_b$
  - on calcule la retenue

$$r_{i+1} = \begin{cases} \overline{1} & \text{pour} \quad a_i + c_i \leq -n \\ 0 & \text{pour} \quad -n < a_i + c_i < n \\ 1 & \text{pour} \quad n \leq a_i + c_i \end{cases}$$

  - on pose $t_i = a_i + c_i - br_{i+1}$ avec $t_p = r_0 = 0$
  - on obtient la somme $(s_p \cdots s_0)_b$ avec $s_i = t_i + r_i$

Système d'Avižienis pour une base $b > 2$

- exactement $2n + 1$ chiffres $\overline{n}, \ldots, 0, \ldots, n$ avec $\frac{b}{2} < n < b$
- système *complet* : tout entier est représentable
- système *redondant* : certains entiers ont plusieurs représentations
- nouvelle méthode d'addition de $(a_{p-1} \cdots a_0)_b$ et $(c_{p-1} \cdots c_0)_b$
  - on calcule la retenue

$$r_{i+1} = \begin{cases} \overline{1} & \text{pour} \quad a_i + c_i \leq -n \\ 0 & \text{pour} \quad -n < a_i + c_i < n \\ 1 & \text{pour} \quad n \leq a_i + c_i \end{cases}$$

  - on pose $t_i = a_i + c_i - br_{i+1}$ avec $t_p = r_0 = 0$
  - on obtient la somme $(s_p \cdots s_0)_b$ avec $s_i = t_i + r_i$

Système d'Avižienis pour une base $b > 2$

- exactement $2n+1$ chiffres $\overline{n}, \ldots, 0, \ldots, n$ avec $\frac{b}{2} < n < b$
- système *complet* : tout entier est représentable
- système *redondant* : certains entiers ont plusieurs représentations
- nouvelle méthode d'addition de $(a_{p-1} \cdots a_0)_b$ et $(c_{p-1} \cdots c_0)_b$
  - on calcule la retenue

$$r_{i+1} = \begin{cases} \overline{1} & \text{pour} \quad a_i + c_i \leq -n \\ 0 & \text{pour} \quad -n < a_i + c_i < n \\ 1 & \text{pour} \quad n \leq a_i + c_i \end{cases}$$

  - on pose $t_i = a_i + c_i - br_{i+1}$ avec $t_p = r_0 = 0$
  - on obtient la somme $(s_p \cdots s_0)_b$ avec $s_i = t_i + r_i$

Système d'Avižienis pour une base $b > 2$

- exactement $2n+1$ chiffres $\overline{n}, \ldots, 0, \ldots, n$ avec $\frac{b}{2} < n < b$
- système *complet* : tout entier est représentable
- système *redondant* : certains entiers ont plusieurs représentations
- nouvelle méthode d'addition de $(a_{p-1} \cdots a_0)_b$ et $(c_{p-1} \cdots c_0)_b$
  - on calcule la retenue

$$r_{i+1} = \begin{cases} \overline{1} & \text{pour} \qquad a_i + c_i \leq -n \\ 0 & \text{pour} \; -n < a_i + c_i < n \\ 1 & \text{pour} \quad n \leq a_i + c_i \end{cases}$$

  - on pose $t_i = a_i + c_i - br_{i+1}$ avec $t_p = r_0 = 0$
  - on obtient la somme $(s_p \cdots s_0)_b$ avec $s_i = t_i + r_i$

Système d'Avižienis pour une base $b > 2$

- exactement $2n+1$ chiffres $\overline{n}, \ldots, 0, \ldots, n$ avec $\frac{b}{2} < n < b$
- système *complet* : tout entier est représentable
- système *redondant* : certains entiers ont plusieurs représentations
- nouvelle méthode d'addition de $(a_{p-1} \cdots a_0)_b$ et $(c_{p-1} \cdots c_0)_b$
  - on calcule la retenue

$$r_{i+1} = \begin{cases} \overline{1} & \text{pour} \quad a_i + c_i \leq -n \\ 0 & \text{pour} \quad -n < a_i + c_i < n \\ 1 & \text{pour} \quad n \leq a_i + c_i \end{cases}$$

garantie
sans propagation
de retenue!

  - on pose $t_i = a_i + c_i - br_{i+1}$ avec $t_p = r_0 = 0$
  - on obtient la somme $(s_p \cdots s_0)_b$ avec $s_i = t_i + r_i$

Numération positionnelle en base $b > 1$

- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(a_p \cdots a_0, a_{-1} \cdots a_{-q})_b$ représente le nombre réel $\displaystyle\sum_{k=-q}^{p} a_k b^k$
- $(a_p \cdots a_0, a_{-1} a_{-2} \cdots)_b$ représente le nombre réel $\displaystyle\sum_{k=-\infty}^{p} a_k b^k$

Non-unicité des représentations des réels

- certains nombres réels admettent plusieurs représentations dans une même base

$$(1)_{10} = (0,9999\cdots)_{10}$$

$$(1)_{d+1} = (0,dddd\cdots)_{d+1}$$

Numération positionnelle en base $b > 1$

- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(a_p \cdots a_0{,}a_{-1} \cdots a_{-q})_b$ représente le nombre réel $\displaystyle\sum_{k=-q}^{p} a_k b^k$
- $(a_p \cdots a_0{,}a_{-1} a_{-2} \cdots)_b$ représente le nombre réel $\displaystyle\sum_{k=-\infty}^{p} a_k b^k$

Non-unicité des représentations des réels

- certains nombres réels admettent
  plusieurs représentations dans une même base

$$(1)_{10} = (0,9999\cdots)_{10}$$

$$(1)_{d+1} = (0,dddd\cdots)_{d+1}$$

Numération positionnelle en base $b > 1$

- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(a_p \cdots a_0{,}a_{-1} \cdots a_{-q})_b$ représente le nombre réel $\sum_{k=-q}^{p} a_k b^k$
- $(a_p \cdots a_0{,}a_{-1}a_{-2} \cdots)_b$ représente le nombre réel $\sum_{k=-\infty}^{p} a_k b^k$

Non-unicité des représentations des réels

- certains nombres réels admettent plusieurs représentations dans une même base

$$(1)_{10} = (0,9999\cdots)_{10}$$

$$(1)_{d+1} = (0,dddd\cdots)_{d+1}$$

Numération positionnelle en base $b > 1$

- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(a_p \cdots a_0{,}a_{-1} \cdots a_{-q})_b$ représente le nombre réel $\displaystyle\sum_{k=-q}^{p} a_k b^k$
- $(a_p \cdots a_0{,}a_{-1} a_{-2} \cdots)_b$ représente le nombre réel $\displaystyle\sum_{k=-\infty}^{p} a_k b^k$

Non-unicité des représentations des réels

- certains nombres réels admettent
plusieurs représentations dans une même base

$$(1)_{10} = (0,9999\cdots)_{10}$$

$$(1)_{d+1} = (0,dddd\cdots)_{d+1}$$

Les nombres b-adiques pour une base $b > 1$

- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(\cdots a_2 a_1 a_0, a_{-1} a_{-2} \cdots a_{-q})_b$ représente le nombre b-adique $\sum_{k=-q}^{+\infty} a_k b^k$
- tout nombre b-adique admet un opposé b-adique (ou *complément*)

$$(\cdots 001)_{10} + (\cdots 999)_{10} = 0$$

$$(\cdots 001)_2 + (\cdots 111)_2 = 0$$

$$(\cdots 001)_{d+1} + (\cdots dddd)_{d+1} = 0$$

Les nombres b-adiques pour une base $b > 1$

- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(\cdots a_2 a_1 a_0, a_{-1} a_{-2} \cdots a_{-q})_b$ représente le nombre b-adique $\sum_{k=-q}^{+\infty} a_k b^k$
- tout nombre b-adique admet un opposé b-adique (ou *complément*)

$$(\cdots 001)_{10} + (\cdots 999)_{10} = 0$$

$$(\cdots 001)_2 + (\cdots 111)_2 = 0$$

$$(\cdots 001)_{d+1} + (\cdots dddd)_{d+1} = 0$$

Les nombres b-adiques pour une base $b > 1$

- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(\cdots a_2 a_1 a_0, a_{-1} a_{-2} \cdots a_{-q})_b$ représente le nombre b-adique $\sum_{k=-q}^{+\infty} a_k b^k$
- tout nombre b-adique admet un opposé b-adique (ou *complément*)

$$(\cdots 001)_{10} + (\cdots 999)_{10} = 0$$

$$(\cdots 001)_2 + (\cdots 111)_2 = 0$$

$$(\cdots 001)_{d+1} + (\cdots dddd)_{d+1} = 0$$

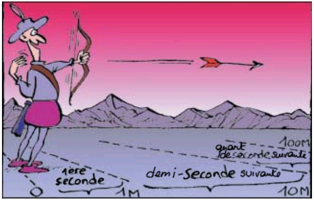
Les nombres b-adiques pour une base $b > 1$

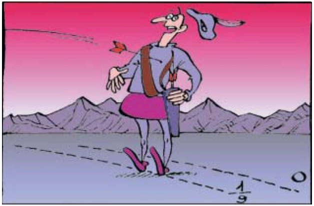
- exactement b chiffres, disons $\{0, 1, \cdots, b-1\}$
- $(\cdots a_2 a_1 a_0, a_{-1} a_{-2} \cdots a_{-q})_b$ représente le nombre b-adique $\sum_{k=-q}^{+\infty} a_k b^k$
- tout nombre b-adique admet un opposé b-adique (ou *complément*)

$$(\cdots 001)_{10} + (\cdots 999)_{10} = 0$$

$$(\cdots 001)_2 + (\cdots 111)_2 = 0$$

$$(\cdots 001)_{d+1} + (\cdots dddd)_{d+1} = 0$$





(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits

$$2^{32} \approx 4,3 \times 10^9 \qquad 2^{64} \approx 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des entiers!

Un choix parmi les $2^{32}!$ choix pour représenter les entiers de 0 à $2^{32}-1$

mots sur $\{0,1\}$ dans l'ordre lexicographique	base 2	base 10
00000000 00000000 00000000 00000000	0	0
00000000 00000000 00000000 00000001		
00000000 00000000 00000000 00000010		
00000000 00000000 00000000 00000011		
...		
11111111 11111111 11111111 11111110		
11111111 11111111 11111111 11111111		

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits

binary digit

$$2^{32} \sim 4,3 \times 10^{9} \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des entiers!

Un choix parmi les $2^{32}!$ choix pour représenter les entiers de 0 à $2^{32}-1$

mots sur $\{0,1\}$ dans l'ordre lexicographique	base 2	base 10
00000000 00000000 00000000 00000000	0	0
00000000 00000000 00000000 00000001		
00000000 00000000 00000000 00000010		
00000000 00000000 00000000 00000011		
...		
11111111 11111111 11111111 11111110		
11111111 11111111 11111111 11111111		

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits
  binary digit

$$2^{32} \sim 4,3 \times 10^9 \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des entiers!

Un choix parmi les 2^{32}! choix pour représenter les entiers de 0 à $2^{32}-1$

mots sur $\{0,1\}$ dans l'ordre lexicographique	base 2	base 10
00000000 00000000 00000000 00000000	0	0
00000000 00000000 00000000 00000001		
00000000 00000000 00000000 00000010		
00000000 00000000 00000000 00000011		
...		
11111111 11111111 11111111 11111110		
11111111 11111111 11111111 11111111		

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits
  binary digit

$$2^{32} \sim 4,3 \times 10^9 \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des entiers!

Un choix parmi les $2^{32}!$ choix pour représenter les entiers de 0 à $2^{32} - 1$

mots sur $\{0, 1\}$ dans l'ordre lexicographique	base 2	base 10
00000000 00000000 00000000 00000000	0	0
00000000 00000000 00000000 00000001		
00000000 00000000 00000000 00000010		
00000000 00000000 00000000 00000011		
...		
11111111 11111111 11111111 11111110		
11111111 11111111 11111111 11111111		

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits
  binary digit

$$2^{32} \sim 4,3 \times 10^9 \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des entiers!

Un choix parmi les 2^{32}! choix pour représenter les entiers de 0 à $2^{32}-1$

mots sur $\{0,1\}$ dans l'ordre lexicographique	base 2	base 10
00000000 00000000 00000000 00000000	0	0
00000000 00000000 00000000 00000001	1	1
00000000 00000000 00000000 00000010	10	2
00000000 00000000 00000000 00000011	11	3
...		...
11111111 11111111 11111111 11111110		4 294 967 294
11111111 11111111 11111111 11111111		4 294 967 295

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits
  binary digit

$$2^{32} \sim 4,3 \times 10^9 \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des entiers!

Un choix parmi les $2^{32}!$ choix pour représenter les entiers de 0 à $2^{32}-1$

mots sur $\{0,1\}$ dans l'ordre lexicographique	base 2	base 10
00000000 00000000 00000000 00000000	0	0
00000000 00000000 00000000 00000001	1	1
00000000 00000000 00000000 00000010	10	2
00000000 00000000 00000000 00000011	11	3
...		...
11111111 11111111 11111111 11111110		4 294 967 294
11111111 11111111 11111111 11111111		4 294 967 295

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits
  binary digit

$$2^{32} \sim 4,3 \times 10^{9} \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des entiers!

Un choix parmi les $2^{32}!$ choix pour représenter les entiers de 0 à $2^{32} - 1$

mots sur $\{0, 1\}$ dans l'ordre lexicographique	base 2	base 10
00000000 00000000 00000000 00000000	0	0
00000000 00000000 00000000 00000001	1	1
00000000 00000000 00000000 00000010	10	2
00000000 00000000 00000000 00000011	11	3
...		...
11111111 11111111 11111111 11111110		4 294 967 294
11111111 11111111 11111111 11111111		4 294 967 295

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits
  binary digit

  $$2^{32} \sim 4,3 \times 10^9 \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des entiers!

Un choix parmi les 2^{32}! choix pour représenter les entiers de 0 à $2^{32} - 1$

mots sur $\{0, 1\}$ dans l'ordre lexicographique	base 2	base 10
00000000 00000000 00000000 00000000	0	0
00000000 00000000 00000000 00000001	1	1
00000000 00000000 00000000 00000010	10	2
00000000 00000000 00000000 00000011	11	3
...		...
11111111 11111111 11111111 11111110		4 294 967 294
11111111 11111111 11111111 11111111		4 294 967 295

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits
  binary digit

$$2^{32} \sim 4,3 \times 10^9 \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des entiers!

Un choix parmi les $2^{32}!$ choix pour représenter les entiers de 0 à $2^{32}-1$

mots sur $\{0,1\}$ dans l'ordre lexicographique	base 2	base 10
00000000 00000000 00000000 00000000	0	0
00000000 00000000 00000000 00000001	1	1
00000000 00000000 00000000 00000010	10	2
00000000 00000000 00000000 00000011	11	3
...		...
11111111 11111111 11111111 11111110		4 294 967 294
11111111 11111111 11111111 11111111		4 294 967 295

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits
  binary digit

  $$2^{32} \sim 4,3 \times 10^9 \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des entiers!

Un choix parmi les 2^{32}! choix pour représenter les entiers de 0 à $2^{32} - 1$

mots sur $\{0, 1\}$ dans l'ordre lexicographique	base 2	base 10
00000000 00000000 00000000 00000000	0	0
00000000 00000000 00000000 00000001	1	1
00000000 00000000 00000000 00000010	10	2
00000000 00000000 00000000 00000011	11	3
...		...
11111111 11111111 11111111 11111110		4 294 967 294
11111111 11111111 11111111 11111111		4 294 967 295

représentation non-signée

Un parmi les $2^{32}!$ choix pour représenter les entiers de $-2^{31}+1$ à $2^{31}-1$

mots sur $\{0,1\}$	base 10
11111111 11111111 11111111 11111111	$-2\,147\,483\,647$
11111111 11111111 11111111 11111110	$-2\,147\,483\,646$
...	...
10000000 00000000 00000000 00000010	-2
10000000 00000000 00000000 00000001	-1
10000000 00000000 00000000 00000000	-0
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

Un parmi les $2^{32}!$ choix pour représenter les entiers de $-2^{31}+1$ à $2^{31}-1$

mots sur $\{0, 1\}$	base 10
11111111 11111111 11111111 11111111	$-2\,147\,483\,647$
11111111 11111111 11111111 11111110	$-2\,147\,483\,646$
...	...
10000000 00000000 00000000 00000010	-2
10000000 00000000 00000000 00000001	-1
10000000 00000000 00000000 00000000	-0
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	$2\,147\,483\,646$
01111111 11111111 11111111 11111111	$2\,147\,483\,647$

bit de signe

Un parmi les $2^{32}!$ choix pour représenter les entiers de $-2^{31}+1$ à $2^{31}-1$

mots sur $\{0,1\}$	base 10
11111111 11111111 11111111 11111111	$-2\,147\,483\,647$
11111111 11111111 11111111 11111110	$-2\,147\,483\,646$
...	...
10000000 00000000 00000000 00000010	-2
10000000 00000000 00000000 00000001	-1
10000000 00000000 00000000 00000000	-0
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	$2\,147\,483\,646$
01111111 11111111 11111111 11111111	$2\,147\,483\,647$

bit de signe

inadapté au calcul

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	$-$ 2 147 483 648
10000000 00000000 00000000 00000001	$-$ 2 147 483 647
...	...
11111111 11111111 11111111 11111101	$-$ 3
11111111 11111111 11111111 11111110	$-$ 2
11111111 11111111 11111111 11111111	$-$ 1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0, 1\}$	base 10
10000000 00000000 00000000 00000000	– 2 147 483 648
10000000 00000000 00000000 00000001	– 2 147 483 647
...	...
11111111 11111111 11111111 11111101	– 3
11111111 11111111 11111111 11111110	– 2
11111111 11111111 11111111 11111111	– 1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	− 2 147 483 648
10000000 00000000 00000000 00000001	− 2 147 483 647
...	...
11111111 11111111 11111111 11111101	− 3
11111111 11111111 11111111 11111110	− 2
11111111 11111111 11111111 11111111	− 1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	− 2 147 483 648
10000000 00000000 00000000 00000001	− 2 147 483 647
...	...
11111111 11111111 11111111 11111101	− 3
11111111 11111111 11111111 11111110	− 2
11111111 11111111 11111111 11111111	− 1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	$-2\,147\,483\,648$
10000000 00000000 00000000 00000001	$-2\,147\,483\,647$
...	...
11111111 11111111 11111111 11111101	-3
11111111 11111111 11111111 11111110	-2
11111111 11111111 11111111 11111111	-1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	– 2 147 483 648
10000000 00000000 00000000 00000001	– 2 147 483 647
...	...
11111111 11111111 11111111 11111101	– 3
11111111 11111111 11111111 11111110	– 2
11111111 11111111 11111111 11111111	– 1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	− 2 147 483 648
10000000 00000000 00000000 00000001	− 2 147 483 647
...	...
11111111 11111111 11111111 11111101	− 3
11111111 11111111 11111111 11111110	− 2
11111111 11111111 11111111 11111111	− 1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

bit de signe

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0, 1\}$	base 10
10000000 00000000 00000000 00000000	$-2\,147\,483\,648$
10000000 00000000 00000000 00000001	$-2\,147\,483\,647$
$\cdots$	
11111111 11111111 11111111 11111101	-3
11111111 11111111 11111111 11111110	-2
11111111 11111111 11111111 11111111	-1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
$\cdots$	$\cdots$
01111111 11111111 11111111 11111110	$2\,147\,483\,646$
01111111 11111111 11111111 11111111	$2\,147\,483\,647$

bit de signe

complément à 2

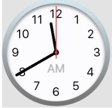
Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0, 1\}$	base 10
10000000 00000000 00000000 00000000	− 2 147 483 648
10000000 00000000 00000000 00000001	− 2 147 483 647
. . .	
11111111 11111111 11111111 11111101	− 3
11111111 11111111 11111111 11111110	− 2
11111111 11111111 11111111 11111111	− 1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
. . .	. . .
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

bit de signe

complément à 2

midi **moins** vingt
twenty **to** noon



midi (**et**) vingt
twenty **past** noon

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31} - 1$

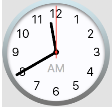
mots sur $\{0, 1\}$	base 10
10000000 00000000 00000000 00000000	$-2\,147\,483\,648$
10000000 00000000 00000000 00000001	$-2\,147\,483\,647$
...	
11111111 11111111 11111111 11111101	-3
11111111 11111111 11111111 11111110	-2
11111111 11111111 11111111 11111111	-1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

bit de signe

complément à 2

1. inverser tous les bits

midi **moins** vingt
twenty **to** noon



midi (**et**) vingt
twenty **past** noon

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

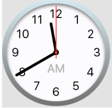
mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	−2 147 483 648
10000000 00000000 00000000 00000001	−2 147 483 647
…	…
11111111 11111111 11111111 11111101	−3
11111111 11111111 11111111 11111110	−2
11111111 11111111 11111111 11111111	−1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
…	…
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

bit de signe

complément à 2

1. inverser tous les bits
2. ajouter 1

midi **moins** vingt
twenty **to** noon



midi (**et**) vingt
twenty **past** noon

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

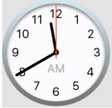
mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	− 2 147 483 648
10000000 00000000 00000000 00000001	− 2 147 483 647
...	
11111111 11111111 11111111 11111101	3
11111111 11111111 11111111 11111110	− 2
11111111 11111111 11111111 11111111	− 1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

bit de signe

complément à 2

1. inverser tous les bits
2. ajouter 1

midi **moins** vingt
twenty **to** noon



midi (**et**) vingt
twenty **past** noon

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

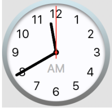
mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	− 2 147 483 648
10000000 00000000 00000000 00000001	− 2 147 483 647
…	…
11111111 11111111 11111111 11111101	− 3
11111111 11111111 11111111 11111110	− 2
11111111 11111111 11111111 11111111	− 1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
…	…
01111111 11111111 11111111 11111110	2 147 483 646
01111111 11111111 11111111 11111111	2 147 483 647

bit de signe

complément à 2

1. inverser tous les bits
2. ajouter 1

midi **moins** vingt
twenty **to** noon



midi (**et**) vingt
twenty **past** noon

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

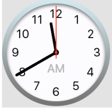
mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	$-2\,147\,483\,648$
10000000 00000000 00000000 00000001	$-2\,147\,483\,647$
...	
11111111 11111111 11111111 11111101	-3
11111111 11111111 11111111 11111110	-2
11111111 11111111 11111111 11111111	-1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
...	...
01111111 11111111 11111111 11111110	$2\,147\,483\,646$
01111111 11111111 11111111 11111111	$2\,147\,483\,647$

bit de signe

complément à 2

1. inverser tous les bits
2. ajouter 1

midi **moins** vingt
twenty **to** noon



midi (**et**) vingt
twenty **past** noon

Un parmi les $2^{32}!$ choix pour représenter les entiers de -2^{31} à $2^{31}-1$

mots sur $\{0,1\}$	base 10
10000000 00000000 00000000 00000000	$-2\,147\,483\,648$
10000000 00000000 00000000 00000001	$-2\,147\,483\,647$
…	
11111111 11111111 11111111 11111101	-3
11111111 11111111 11111111 11111110	-2
11111111 11111111 11111111 11111111	-1
00000000 00000000 00000000 00000000	0
00000000 00000000 00000000 00000001	1
00000000 00000000 00000000 00000010	2
…	…
01111111 11111111 11111111 11111110	$2\,147\,483\,646$
01111111 11111111 11111111 11111111	$2\,147\,483\,647$

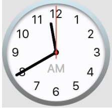
bit de signe

complément à 2

1. inverser tous les bits
2. ajouter 1

$$2^{32}-(2^{32}-x)=x$$

midi **moins** vingt
twenty **to** noon



midi (**et**) vingt
twenty **past** noon

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits

$$2^{32} \sim 4,3 \times 10^{9} \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des réels!

Normes IEEE 754 `binary32`

1	11000110	10010011110000111000000
s	e	m

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits

$$2^{32} \sim 4,3 \times 10^{9} \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des réels!

Normes IEEE 754 `binary32`

$\boxed{1}$		$\boxed{11000110}$		$\boxed{10010011110000111000000}$
s		e		m
$\downarrow$		$\downarrow$		$\downarrow$
$(-1)^{s}$	$\times$	2^{e-B}	$\times$	$1 \bullet m$
$(-1)^{1}$	$\times$	$2^{198-127}$	$\times$	$1.10010011110000111000000_2$

environ $-3.7240626.10^{21}$

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits

$$2^{32} \sim 4,3 \times 10^{9} \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des réels!

Normes IEEE 754 `binary32`

$$\begin{array}{ccccc} \boxed{1} & & \boxed{11000110} & & \boxed{10010011110000111000000} \\ s & & e & & m \\ \downarrow & & \downarrow & & \downarrow \\ (-1)^{s} & \times & 2^{e-B} & \times & 1 \bullet m \\ (-1)^{1} & \times & 2^{198-127} & \times & 1.10010011110000111000000_{2} \end{array}$$

environ $-3.7240626.10^{21}$

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits

$$2^{32} \sim 4,3 \times 10^9 \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des réels!

Normes IEEE 754 `binary32`

$$\begin{array}{ccccc}
\boxed{1} & & \boxed{11000110} & & \boxed{10010011110000111000000} \\
s & & e & & m \\
\downarrow & & \downarrow & & \downarrow \\
(-1)^s & \times & 2^{e-B} & \times & 1 \bullet m \\
(-1)^1 & \times & 2^{198-127} & \times & 1.10010011110000111000000_2
\end{array}$$

environ $-3.7240626.10^{21}$

(si) on manipule rarement les très très très graaaaaaaaaands nombres
(si) on manipule rarement les nombres avec une graaaaande précision

Choix de fixer la taille des représentations

- architectures communes 32 ou 64 bits: arithmétique sur des nombres représentés sur 32 ou 64 bits

$$2^{32} \sim 4,3 \times 10^9 \qquad 2^{64} \sim 18,4 \times 10^{18}$$

- une infinité de choix pour représenter des réels!

Normes IEEE 754 `binary32`

$$\begin{array}{ccccc}
\boxed{1} & & \boxed{11000110} & & \boxed{10010011110000111000000} \\
s & & e & & m \\
\downarrow & & \downarrow & & \downarrow \\
(-1)^s & \times & 2^{e-B} & \times & 1 \bullet m \\
(-1)^1 & \times & 2^{198-127} & \times & 1.10010011110000111000000_2
\end{array}$$

environ $-3.7240626.10^{21}$

Normes IEEE 754 `binary32`

$$\begin{array}{ccccc} \boxed{1} & & \boxed{11000110} & & \boxed{10010011110000111000000} \\ s & & e & & m \\ \downarrow & & \downarrow & & \downarrow \\ (-1)^s & \times & 2^{e-B} & \times & 1 \bullet m \\ (-1)^1 & \times & 2^{198-127} & \times & 1.10010011110000111000000_2 \end{array}$$

environ $-3.7240626.10^{21}$

- pour $0 < e < e_{max}$, nombre normal de valeur $(-1)^s \times 1.m \times 2^{e-B}$
- pour $e = 0$, nombre dénormalisé de valeur $(-1)^s \times 0.m \times 2^{1-B}$
- pour [illegible] et [illegible], deux valeurs [illegible] et [illegible]
- pour [illegible] et [illegible], valeur indéterminée [illegible] (Not-a-Number)

- mode arrondi au plus près (mode par défaut)
- mode arrondi vers zéro
- mode arrondi vers plus l'infini
- mode arrondi vers moins l'infini

Normes IEEE 754 `binary32`

$$
\begin{array}{ccccc}
\boxed{1} & & \boxed{11000110} & & \boxed{10010011110000111000000} \\
s & & e & & m \\
\downarrow & & \downarrow & & \downarrow \\
(-1)^s & \times & 2^{e-B} & \times & 1 \bullet m \\
(-1)^1 & \times & 2^{198-127} & \times & 1.10010011110000111000000_2
\end{array}
$$

environ $-3.7240626.10^{21}$

- pour $0 < e < e_{max}$, nombre normal de valeur $(-1)^s \times 1.m \times 2^{e-B}$
- pour $e = 0$, nombre dénormalisé de valeur $(-1)^s \times 0.m \times 2^{1-B}$
- pour $e = e_{max}$ et $m = 0$, deux valeurs $-\infty$ et $+\infty$
- pour $e = e_{max}$ et $m \neq 0$, valeur indéterminée `NaN` (Not-a-Number)

- mode arrondi au plus près (mode par défaut)
- mode arrondi vers zéro
- mode arrondi vers plus l'infini
- mode arrondi vers moins l'infini

Normes IEEE 754 — binary32

$$\boxed{1} \quad \boxed{11000110} \quad \boxed{10010011110000111000000}$$

$$\begin{array}{ccccc} s & & e & & m \\ \downarrow & & \downarrow & & \downarrow \\ (-1)^s & \times & 2^{e-B} & \times & 1 \bullet m \\ (-1)^1 & \times & 2^{198-127} & \times & 1.10010011110000111000000_2 \end{array}$$

environ $-3.7240626.10^{21}$

- pour $0 < e < e_{max}$, nombre normal de valeur $(-1)^s \times 1.m \times 2^{e-B}$
- pour $e = 0$, nombre dénormalisé de valeur $(-1)^s \times 0.m \times 2^{1-B}$
- pour $e = e_{max}$ et $m = 0$, deux valeurs $-\infty$ et $+\infty$
- pour $e = e_{max}$ et $m \neq 0$, valeur indéterminée NaN (Not-a-Number)

- mode arrondi au plus près (mode par défaut)
- mode arrondi vers zéro
- mode arrondi vers plus l'infini
- mode arrondi vers moins l'infini

Normes IEEE 754 — binary32

$$\boxed{1} \qquad \boxed{11000110} \qquad \boxed{10010011110000111000000}$$

$$\begin{array}{ccccc} s & & e & & m \\ \downarrow & & \downarrow & & \downarrow \\ (-1)^s & \times & 2^{e-B} & \times & 1 \bullet m \\ (-1)^1 & \times & 2^{198-127} & \times & 1.10010011110000111000000_2 \end{array}$$

environ $-3.7240626.10^{21}$

- pour $0 < e < e_{max}$, nombre normal de valeur $(-1)^s \times 1.m \times 2^{e-B}$
- pour $e = 0$, nombre dénormalisé de valeur $(-1)^s \times 0.m \times 2^{1-B}$
- pour $e = e_{max}$ et $m = 0$, deux valeurs $-\infty$ et $+\infty$
- pour $e = e_{max}$ et $m \neq 0$, valeur indéterminée `NaN` (Not-a-Number)

- mode arrondi au plus près (mode par défaut)
- mode arrondi vers zéro
- mode arrondi vers plus l'infini
- mode arrondi vers moins l'infini

Normes IEEE 754 — `binary32`

$$\boxed{1} \qquad \boxed{11000110} \qquad \boxed{10010011110000111000000}$$

$$\begin{array}{ccccc} s & & e & & m \\ \downarrow & & \downarrow & & \downarrow \\ (-1)^s & \times & 2^{e-B} & \times & 1 \bullet m \\ (-1)^1 & \times & 2^{198-127} & \times & 1.10010011110000111000000_2 \end{array}$$

environ $-3.7240626.10^{21}$

- pour $0 < e < e_{max}$, nombre normal de valeur $(-1)^s \times 1.m \times 2^{e-B}$
- pour $e = 0$, nombre dénormalisé de valeur $(-1)^s \times 0.m \times 2^{1-B}$
- pour $e = e_{max}$ et $m = 0$, deux valeurs $-\infty$ et $+\infty$
- pour $e = e_{max}$ et $m \neq 0$, valeur indéterminée `NaN` (Not-a-Number)

- mode arrondi au plus près (mode par défaut)
- mode arrondi vers zéro
- mode arrondi vers plus l'infini
- mode arrondi vers moins l'infini

Normes IEEE 754 — `binary32`

$$\begin{array}{ccccc} \boxed{1} & & \boxed{11000110} & & \boxed{10010011110000111000000} \\ s & & e & & m \\ \downarrow & & \downarrow & & \downarrow \\ (-1)^s & \times & 2^{e-B} & \times & 1 \bullet m \\ (-1)^1 & \times & 2^{198-127} & \times & 1.10010011110000111000000_2 \end{array}$$

environ $-3.7240626.10^{21}$

- pour $0 < e < e_{max}$, nombre normal de valeur $(-1)^s \times 1.m \times 2^{e-B}$
- pour $e = 0$, nombre dénormalisé de valeur $(-1)^s \times 0.m \times 2^{1-B}$
- pour $e = e_{max}$ et $m = 0$, deux valeurs $-\infty$ et $+\infty$
- pour $e = e_{max}$ et $m \neq 0$, valeur indéterminée `NaN` (Not-a-Number)

- mode arrondi au plus près (mode par défaut)
- mode arrondi vers zéro
- mode arrondi vers plus l'infini
- mode arrondi vers moins l'infini