

R103 — Architecture des ordinateurs

Chapitre 1 — Codage de l'information

Clément PAGÈS
clement.pages@u-pec.fr

Université Paris-Est Créteil
BUT Informatique 1^{re} année

2025-2026

Architecture des ordinateurs

- S'intéresse aux **composants** des machines et à leur **structure matérielle**.
- Points abordés dans cette ressource
 - ① **Éléments de base d'architecture**
 - ② **Codage de l'information**
 - ③ Algèbre booléenne.
 - ④ Circuits logiques élémentaires.
- Informatique : science du traitement automatisé de l'information.
Contraction de « information » et « automatique ».
- Ordinateur : appareil concret permettant de manipuler de l'information de manière automatisée.

Sommaire

- 1 Éléments de base d'architecture
- 2 Codage de l'information
- 3 Représentation en mémoire des nombres

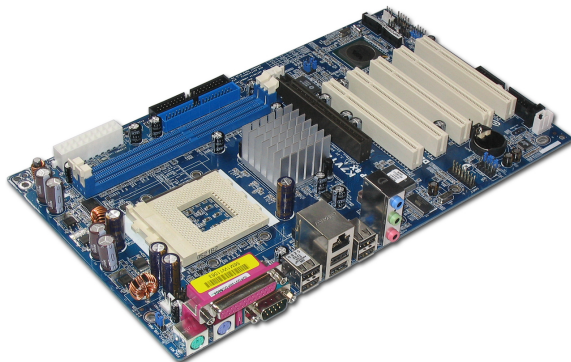
Composants d'un ordinateur

Carte mère

Processeur (ou *CPU*, *Central Process Unit*)

Mémoires

Entrée-sorties



Source : Wikipedia, *Carte mère*

Composants d'un ordinateur

Carte mère Carte principale sur laquelle se branchent tous les autres composants. Elle gère la communication entre tous les éléments.

Processeur (ou *CPU*, *Central Process Unit*)

Mémoires

Entrée-sorties

Composants d'un ordinateur

Carte mère

Processeur (ou *CPU*, *Central Process Unit*) Il effectue les calculs, exécute les programmes. Contient quatre parties principales :

- l'unité arithmétique et logique;
- l'unité de contrôle, décode les instructions à exécuter;
- les registres sont des petites mémoires temporaires;
- signaux de commandes communiquent avec les autres composants.

Mémoires

Entrée-sorties

Composants d'un ordinateur

Carte mère

Processeur (ou *CPU*, *Central Process Unit*)

Mémoires Il en existe plusieurs sortes.

- Mémoire morte — *ROM*, *read-only memory*
- Mémoire vive — *RAM*, *random access memory*
- Mémoire cache — Utilisée par le processeur
- Mémoires de masse — disques durs, CD, ...
- Mémoires flash — Clés USB, SSD, ...

Entrée-sorties

Composants d'un ordinateur

Carte mère

Processeur (ou *CPU*, *Central Process Unit*)

Mémoires

Entrée-sorties

- Périphériques d'entrée : claviers, souris, webcams, micros, etc.
- Périphériques de sortie : écrans, imprimantes, haut-parleurs, etc.,
- Périphériques d'entrée-sortie : écran tactile, puce bluetooth, etc.

Sommaire

- ① Éléments de base d'architecture
- ② Codage de l'information
 - Données numériques et données textuelles
 - Bases de numération
- ③ Représentation en mémoire des nombres

Information dans les circuits électroniques

- Un ordinateur est avant tout un **circuit électronique** (extrêmement complexe).
- On ne peut représenter que deux situations : « présence de courant électrique » et « absence de courant électrique ».
 - Absence de courant $\rightarrow 0$
 - Présence de courant $\rightarrow 1$

Information dans les circuits électroniques

- Un ordinateur est avant tout un **circuit électronique** (extrêmement complexe).
- On ne peut représenter que deux situations : « présence de courant électrique » et « absence de courant électrique ».
 - Absence de courant $\rightarrow 0$
 - Présence de courant $\rightarrow 1$
- Système **binaire** : on ne manipule que des 0 et des 1.

Information dans les circuits électroniques

- Un ordinateur est avant tout un **circuit électronique** (extrêmement complexe).
- On ne peut représenter que deux situations : « présence de courant électrique » et « absence de courant électrique ».
 - Absence de courant $\rightarrow 0$
 - Présence de courant $\rightarrow 1$
- Système **binaire** : on ne manipule que des 0 et des 1.
- Bases de numération
 - Décimal** base 10. Les chiffres de 0 à 9.
 - Binaire** base 2. Uniquement 0 et 1.
 - Octal** base 8. Les chiffres de 0 à 7.
 - Hexadécimal** base 16. Les chiffres 0 1 2 3 4 5 6 7 8 9 A B C D E F

Les données numériques

- Nombres entiers

Les données numériques

- Nombres entiers
 - Non signés (absence de signe, considérés comme positifs).

Les données numériques

- Nombres entiers
 - Non signés (absence de signe, considérés comme positifs).
 - Signés (positifs ou négatifs).

Les données numériques

- Nombres entiers
 - Non signés (absence de signe, considérés comme positifs).
 - Signés (positifs ou négatifs).
- Nombres **flottants** (décimaux)

Les données numériques

- Nombres entiers
 - Non signés (absence de signe, considérés comme positifs).
 - Signés (positifs ou négatifs).
- Nombres **flottants** (décimaux)
 - En écriture décimale (avec une virgule) : 1,487 – 9,0454

Les données numériques

- Nombres entiers
 - Non signés (absence de signe, considérés comme positifs).
 - Signés (positifs ou négatifs).
- Nombres **flottants** (décimaux)
 - En écriture décimale (avec une virgule) : 1,487 – 9,0454
 - En écriture scientifique $1,4587 \times 10^{-8}$.

Les données numériques

- Nombres entiers
 - Non signés (absence de signe, considérés comme positifs).
 - Signés (positifs ou négatifs).
- Nombres **flottants** (décimaux)
 - En écriture décimale (avec une virgule) : 1,487 – 9,0454
 - En écriture scientifique $1,4587 \times 10^{-8}$.
- Certains nombres réels ne peuvent pas être représentés.

Les données numériques

- Nombres entiers
 - Non signés (absence de signe, considérés comme positifs).
 - Signés (positifs ou négatifs).
- Nombres **flottants** (décimaux)
 - En écriture décimale (avec une virgule) : 1,487 – 9,0454
 - En écriture scientifique $1,4587 \times 10^{-8}$.
- Certains nombres réels ne peuvent pas être représentés.
 - Les nombres irrationnels (exemple : $\sqrt{2}$)

Les données numériques

- Nombres entiers
 - Non signés (absence de signe, considérés comme positifs).
 - Signés (positifs ou négatifs).
- Nombres **flottants** (décimaux)
 - En écriture décimale (avec une virgule) : 1,487 – 9,0454
 - En écriture scientifique $1,4587 \times 10^{-8}$.
- Certains nombres réels ne peuvent pas être représentés.
 - Les nombres irrationnels (exemple : $\sqrt{2}$)
 - Certains nombres décimaux (exemple : 0,925)

Les données numériques

- Nombres entiers
 - Non signés (absence de signe, considérés comme positifs).
 - Signés (positifs ou négatifs).
- Nombres **flottants** (décimaux)
 - En écriture décimale (avec une virgule) : 1,487 – 9,0454
 - En écriture scientifique $1,4587 \times 10^{-8}$.
- Certains nombres réels ne peuvent pas être représentés.
 - Les nombres irrationnels (exemple : $\sqrt{2}$)
 - Certains nombres décimaux (exemple : 0,925)
- L'ensemble des nombres qui admettent une décomposition en base 2 est nommé **ensemble des nombres binaires**. Dans ce cours, on le note $\mathbb{B}$.

Données non numériques

Principe

- Principalement les caractères.
- Table de codage caractère → valeur numérique → écriture en binaire.
- L'**encodage** n'est pas standard (mais il le devient progressivement).
 - ASCII (*American Standard Code for Information Interchange*) **7 bits**
 - Unicode **16 bits**
 - ISO/IEC 10646 (*Intern. Standards Organization/Intern. Electronic Commission*), sur **32 bits**.

Données non numériques

La table ASCII

	0_	1_	2_	3_	4_	5_	6_	7_
_0	NUL	DLE	SP	0	@	P	'	p
_1	SOH	DC1	!	1	A	Q	a	q
_2	STX	DC2	"	2	B	R	b	r
_3	ETX	DC3	#	3	C	S	c	s
_4	EOT	DC4	\$	4	D	T	d	t
_5	ENQ	NAK	%	5	E	U	e	u
_6	ACK	SYN	&	6	F	V	f	v
_7	BEL	ETB	'	7	G	W	g	w
_8	BS	CAN	(	8	H	X	h	x
_9	HT	EM	)	9	I	Y	i	y
_A	LF	SUB	*	:	J	Z	j	z
_B	VT	ESC	+	;	K	[	K	}
_C	FF	FS	,	<	L	\		
_D	CR	GS	-	=	M	]	m	}
_E	SO	RS	.	>	N	^	n	~
_F	SI	US	/	?	O	_	o	DEL

Écriture en binaire

Écriture binaire d'un nombre

Écrire un nombre entier naturel en base 2, c'est l'écrire comme une somme de termes de la forme $b \cdot 2^k$ où $b \in \{0, 1\}$ et $k \in \mathbb{N}$.

Écriture en binaire

Écriture binaire d'un nombre

Écrire un nombre entier naturel en base 2, c'est l'écrire comme une somme de termes de la forme $b \cdot 2^k$ où $b \in \{0, 1\}$ et $k \in \mathbb{N}$.

Écrire un nombre réel en base 2, c'est l'écrire comme une somme de termes de la forme $b \cdot 2^k$ où $b \in \{0, 1\}$ et $k \in \mathbb{Z}$.

Écriture en binaire

Écriture binaire d'un nombre

Écrire un nombre entier naturel en base 2, c'est l'écrire comme une somme de termes de la forme $b \cdot 2^k$ où $b \in \{0, 1\}$ et $k \in \mathbb{N}$.

Écrire un nombre réel en base 2, c'est l'écrire comme une somme de termes de la forme $b \cdot 2^k$ où $b \in \{0, 1\}$ et $k \in \mathbb{Z}$.

Écriture en binaire

Écriture binaire d'un nombre

Écrire un nombre entier naturel en base 2, c'est l'écrire comme une somme de termes de la forme $b \cdot 2^k$ où $b \in \{0, 1\}$ et $k \in \mathbb{N}$.

Écrire un nombre réel en base 2, c'est l'écrire comme une somme de termes de la forme $b \cdot 2^k$ où $b \in \{0, 1\}$ et $k \in \mathbb{Z}$.

Exemple : écriture binaire de $13_{(10)}$ et $51,5625_{(10)}$

$$13_{(10)} = 8 + 4 + 1 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 1101_{(2)}$$

$$\begin{aligned} 51,5625_{(10)} &= 32 + 16 + 2 + 1 + \frac{1}{2} + \frac{1}{16} \\ &= 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^1 + 1 \cdot 2^{-1} + 1 \cdot 2^{-4} \\ &= 110011,1001_{(2)} \end{aligned}$$

Passage de l'écriture décimale à l'écriture binaire

Pour les nombres entiers

- Deux méthodes
 - Décomposer mentalement en somme de puissances de deux.

Passage de l'écriture décimale à l'écriture binaire

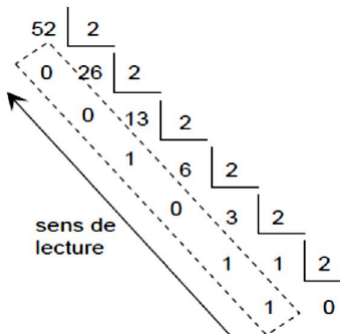
Pour les nombres entiers

- Deux méthodes
 - Décomposer mentalement en somme de puissances de deux.
 - Algorithme de la remontée : effectuer des divisions euclidiennes par 2 successives, jusqu'au premier quotient nul. Lire les restes du dernier au premier.

Passage de l'écriture décimale à l'écriture binaire

Pour les nombres entiers

- Deux méthodes
 - Décomposer mentalement en somme de puissances de deux.
 - Algorithme de la remontée : effectuer des divisions euclidiennes par 2 successives, jusqu'au premier quotient nul. Lire les restes du dernier au premier.
- Application : $52_{(10)} = 1101_{(2)}$



Passage de l'écriture décimale à l'écriture binaire

Pour les nombres non entiers

- On écrit la partie entière en binaire (cf. diapositive précédente).
- On écrit la partie décimale en binaire par multiplications successives par 2, jusqu'à obtenir un produit entier. L'écriture binaire s'obtient en lisant la partie entière des produits du premier au dernier.

$$0,8125_{(10)} \cdot 2 = 1,625_{(10)}$$

$$0,625 \cdot 2 = 1,25_{(10)}$$

$$0,25_{(10)} \cdot 2 = 0,5_{(10)}$$

$$0,5 \cdot 2 = 1$$

Donc $0,5128_{(10)} = 0,1101_{(2)}$

Deux autres bases utiles : octal et hexadécimal

Principe

- La base 2 est pénible car l'écriture des nombres est longue.

Deux autres bases utiles : octal et hexadécimal

Principe

- La base 2 est pénible car l'écriture des nombres est longue.
- On écrit souvent les nombres en base 8 ou en base 16, plus pratique.

Deux autres bases utiles : octal et hexadécimal

Principe

- La base 2 est pénible car l'écriture des nombres est longue.
- On écrit souvent les nombres en base 8 ou en base 16, plus pratique.
- Méthode : on part de l'écriture binaire et on regroupe par blocs
 - de 3 bits pour l'octal ($8 = 2^3$).

Deux autres bases utiles : octal et hexadécimal

Principe

- La base 2 est pénible car l'écriture des nombres est longue.
- On écrit souvent les nombres en base 8 ou en base 16, plus pratique.
- Méthode : on part de l'écriture binaire et on regroupe par blocs
 - de 3 bits pour l'octal ($8 = 2^3$).
 - de 4 bits pour l'hexadécimal ($16 = 2^4$).

Deux autres bases utiles : octal et hexadécimal

Principe

- La base 2 est pénible car l'écriture des nombres est longue.
- On écrit souvent les nombres en base 8 ou en base 16, plus pratique.
- Méthode : on part de l'écriture binaire et on regroupe par blocs
 - de 3 bits pour l'octal ($8 = 2^3$).
 - de 4 bits pour l'hexadécimal ($16 = 2^4$).

Exemple

Écriture du nombre $1100111010101_{(2)}$ en octal et en hexadécimal.

$$1\ 100\ 111\ 010\ 101_{(2)} = 14725_{(8)}$$

$$1\ 1001\ 1101\ 0101_{(2)} = 19D5_{(16)}$$

Deux autres bases utiles : octal et hexadécimal

Tables à connaître

Décimal	0	1	2	3	4	5	6	7	8
Binaire	0	1	10	11	100	101	110	111	1000
Octal	0	1	2	3	4	5	6	7	10
Hexadécimal	0	1	2	3	4	5	6	7	8

Décimal	9		10		11		12		13		14		15	
Binaire	1001		1010		1011		1100		1101		1110		1111	
Octal	11		12		13		14		15		16		17	
Hexadécimal	9		A		B		C		D		E		F	

Sommaire

- 1 Éléments de base d'architecture
- 2 Codage de l'information
- 3 Représentation en mémoire des nombres

Nombres entiers signés

Différentes représentations

- Nous avons vu comment représenter des **nombres entiers** non signés.
- Comment représenter des *nombres signés*?

Nombres entiers signés

Différentes représentations

- Nous avons vu comment représenter des **nombres entiers** non signés.
- Comment représenter des *nombres signés* ?
- Il existe **trois représentations**
 - ① Signe et valeur absolue.
 - ② Représentation en complément à 1.
 - ③ Représentation en complément à 2.

Nombres entiers signés

Différentes représentations

- Nous avons vu comment représenter des **nombres entiers** non signés.
- Comment représenter des *nombres signés*?
- Il existe **trois représentations** (exemple sur 3 bits)
 - ① **Signe et valeur absolue.** Le bit de poids fort désigne le signe (0 pour positif, 1 pour négatif).

$$-1_{(10)} = -001_{(2)} = 101_{(2_{va})}$$

$$+1_{(10)} = +001_{(2)} = 001_{(2_{va})}$$

- ② Représentation en complément à 1.
- ③ Représentation en complément à 2.

Nombres entiers signés

Différentes représentations

- Nous avons vu comment représenter des **nombres entiers** non signés.
- Comment représenter des *nombres signés* ?
- Il existe **trois représentations** (exemple sur 3 bits)
 - 1 Signe et valeur absolue.
 - 2 **Représentation en complément à 1**. Le bit de poids fort code le signe, les autres sont complémentés si le nombre est négatif, sinon ils codent la valeur en binaire naturel :

$$-1_{(10)} = -001_{(2)} = 110_{(2_{c1})}$$

$$+1_{(10)} = +001_{(2)} = 001_{(2_{c1})}$$

- 3 Représentation en complément à 2.

Nombres entiers signés

Différentes représentations

- Nous avons vu comment représenter des **nombres entiers** non signés.
- Comment représenter des *nombres signés* ?
- Il existe **trois représentations** (exemple sur 3 bits)
 - ① Signe et valeur absolue.
 - ② Représentation en complément à 1.
 - ③ **Représentation en complément à 2**. Le bit de poids fort code le signe, les autres sont complémentés et incrémentés de 1 s'il est négatif, sinon ils codent la valeur en binaire naturel :

$$-1_{(10)} = -001_{(2)} = 111_{(2c2)}$$

$$+1_{(10)} = +001_{(2)} = 001_{(2c2)}$$

Nombres entiers signés

Exemple complet sur 3 bits

Signe et valeur absolue			
Sgn	b_1	b_0	
0	0	0	+0
0	0	1	+1
0	1	0	+2
0	1	1	+3
1	0	0	-0
1	0	1	-1
1	1	0	-2
1	1	1	-3

Complément à 1			
Sgn	b_1	b_0	
0	0	0	+0
0	0	1	+1
0	1	0	+2
0	1	1	+3
1	0	0	-3
1	0	1	-2
1	1	0	-1
1	1	1	-0

Complément à 2			
Sgn	b_1	b_0	
0	0	0	+0
0	0	1	+1
0	1	0	+2
0	1	1	+3
1	0	0	-4
1	0	1	-3
1	1	0	-2
1	1	1	-1

Les flottants

La norme IEEE 754

- Aussi appelés **représentation en virgule flottante**.
- Norme principale : IEEE 754 simple précision.
- Se base sur l'écriture scientifique... en base 2 ! Tout nombre binaire $b \in \mathbb{B}$ s'écrit de manière unique sous la forme $S \cdot M \cdot 2^E$ où $S = \pm 1$ (**signe** du nombre), $M \in \mathbb{N}$ est un nombre entier et E est l'exposant.

Les flottants

La norme IEEE 754

- Aussi appelés **représentation en virgule flottante**.
- Norme principale : IEEE 754 simple précision.
- Se base sur l'écriture scientifique... en base 2 ! Tout nombre binaire $b \in \mathbb{B}$ s'écrit de manière unique sous la forme $S \cdot M \cdot 2^E$ où $S = \pm 1$ (**signe** du nombre), $M \in \mathbb{N}$ est un nombre entier et E est l'exposant.
- En pratique
 - $S = 0$ ou 1 : 1 bit
 - On **biaise** l'exposant en encodant $E + 127$: 8 bits
 - Mantisse M : 23 bits
 - Au total : 32 bits

Les flottants

Exemple

On cherche à représenter $x = -6,531\,25_{(10)}$ avec la norme IEEE 754 simple précision.

Les flottants

Exemple

On cherche à représenter $x = -6,531\,25_{(10)}$ avec la norme IEEE 754 simple précision.

$$-6,53125_{(10)} = -110,10001_{(2)} = 1,1010001 \cdot 2^2$$

Les flottants

Exemple

On cherche à représenter $x = -6,531\,25_{(10)}$ avec la norme IEEE 754 simple précision.

$$-6,53125_{(10)} = -110,10001_{(2)} = 1,1010001 \cdot 2^2$$

Donc

$$\begin{cases} S = 1 \\ M = 1010001_{(2)} \\ E = 2 \quad \text{i.e.} \quad E + 127 = 129_{(10)} = 10000001_{(2)} \end{cases}$$

Les flottants

Exemple

On cherche à représenter $x = -6,531\,25_{(10)}$ avec la norme IEEE 754 simple précision.

$$-6,53125_{(10)} = -110,10001_{(2)} = 1,1010001 \cdot 2^2$$

Donc

$$\begin{cases} S = 1 \\ M = 1010001_{(2)} \\ E = 2 \quad \text{i.e.} \quad E + 127 = 129_{(10)} = 10000001_{(2)} \end{cases}$$

La représentation de x est :

S	$E + 127$	M
1	10000001	0100010000000000000000