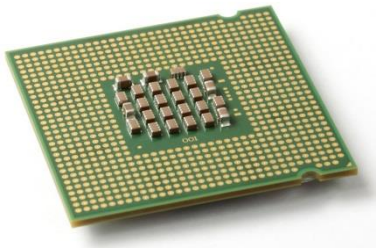
0001 1011

Le langage de l'ordinateur
Représentation des informations

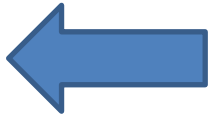
0001 1011



Rôle d'un ordinateur

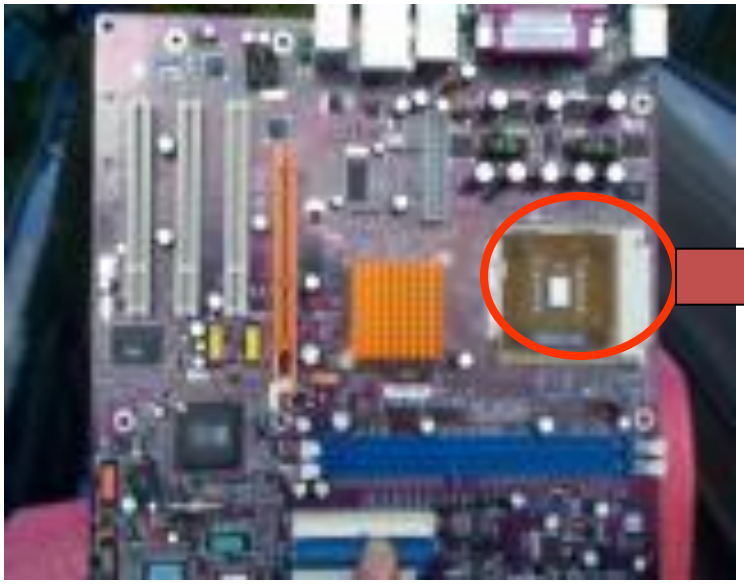
- Exécuter un programme de traitement sur des données en vue de résoudre un problème.

- Deux aspects :
  - L'expression du problème à résoudre, de sa solution dans un **langage compréhensible par l'ordinateur**



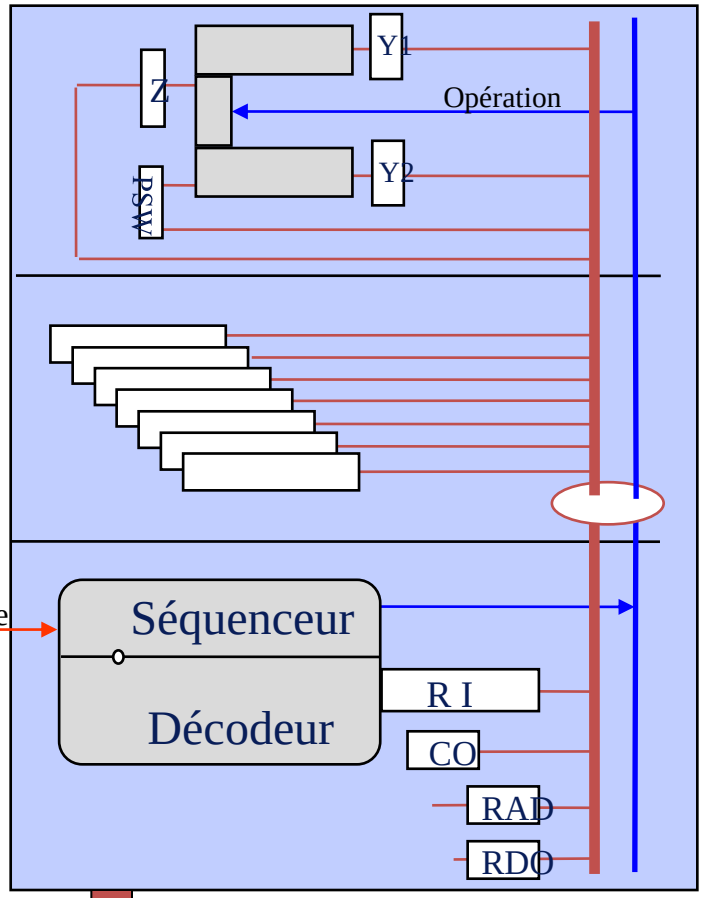
  - La **structure de l'ordinateur** qui doit être composé d'éléments permettant le stockage, le traitement , la lecture ou l'écriture des données

Composants de l'ordinateur

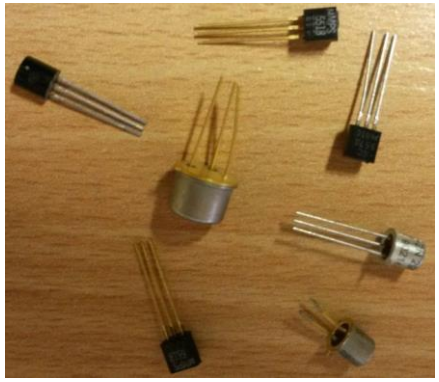


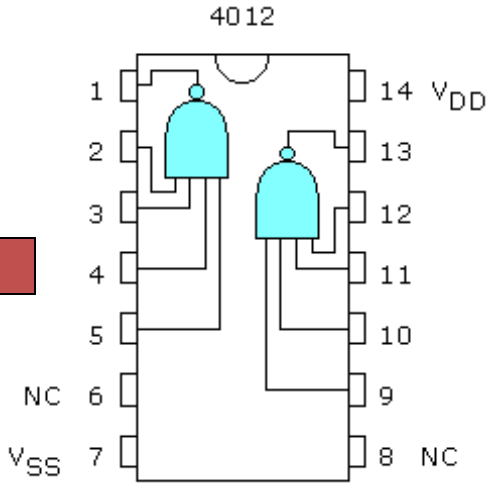
Unité Arithmétique et logique

Registres



TRANSISTORS





CIRCUITS LOGIQUES OU INTEGRES (PORTES)

La représentation des informations sur la machine physique

- La donnée de base manipulée par la machine physique est le **bit** (*Binary Digit*) qui ne peut prendre que deux valeurs : 0 et 1

- Ce 0 et 1 correspondent aux deux niveaux de voltage (0-1 et 2-5 volts) admis pour les signaux électriques issus des composants électroniques (transistors) qui constituent les circuits physiques de la machine

- Toutes les informations (nombres, caractères et instructions) ne peuvent être représentées que par une combinaison de 0 et 1 : **chaine binaire**. Un **octet** est une chaîne de 8 bits.

La représentation des informations sur la machine physique

1 bit	0, 1	2^1 états
2 bits	00, 01, 10, 11	2^2 états
3 bits	000,001,010,011,100,101,110, 111	2^3 états
n bits		2^n états

1 octet = 8 bits (byte)	Avant 1998	Après 1998
Kilooctet (Ko)	2^{10} octets = 1024 octets	1000 octets
Mégaoctet (Mo)	2^{20} octets = 1024 Koctets	1000 Koctets
Gigaoctet (Go)	2^{30} octets = 1024 Moctets	1000 Moctets

La représentation des informations sur la machine physique

$(N)_X = a_n \ldots\ldots a_1\ a_0$ exprimé dans la base $X = a_0 \times X^0 + a_1 \times X^1 + \ldots + a_n \times X^n$

Poids fort

Poids faible

base	symboles	valeur
X = 2	0, 1	$110 = 0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2$
X = 10	0,1,2,3,4,5,6,7,8,9	$110 = 0 \times 10^0 + 1 \times 10^1 + 1 \times 10^2$
X = 16	0,1,2,3,4,5,6,7,8,9 A B C D E F	$110 = 0 \times 16^0 + 1 \times 16^1 + 1 \times 16^2$

La représentation des informations sur la machine physique

base	chiffres	Écriture en base 2
10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9	0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001
16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9 A (10), B (11), C(12), D(13), E(14), F(15)	0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, 1001, 1010, 1011, 1100, 1101, 1110, 1111
8	0, 1, 2, 3, 4, 5, 6, 7	000, 001, 010, 011, 100, 101,110, 111

$(AF2)_{16} = (1010\ 1111\ 0010)_2$ $(72)_8 = (111\ 010)_2$

La représentation des informations sur la machine physique

- Les bases d'expression : conversions utiles

Base 10 → base X

- **Méthode des divisions** :
- Partie entière : on divise PE(N) par X jusqu'à obtenir un quotient égal à 0

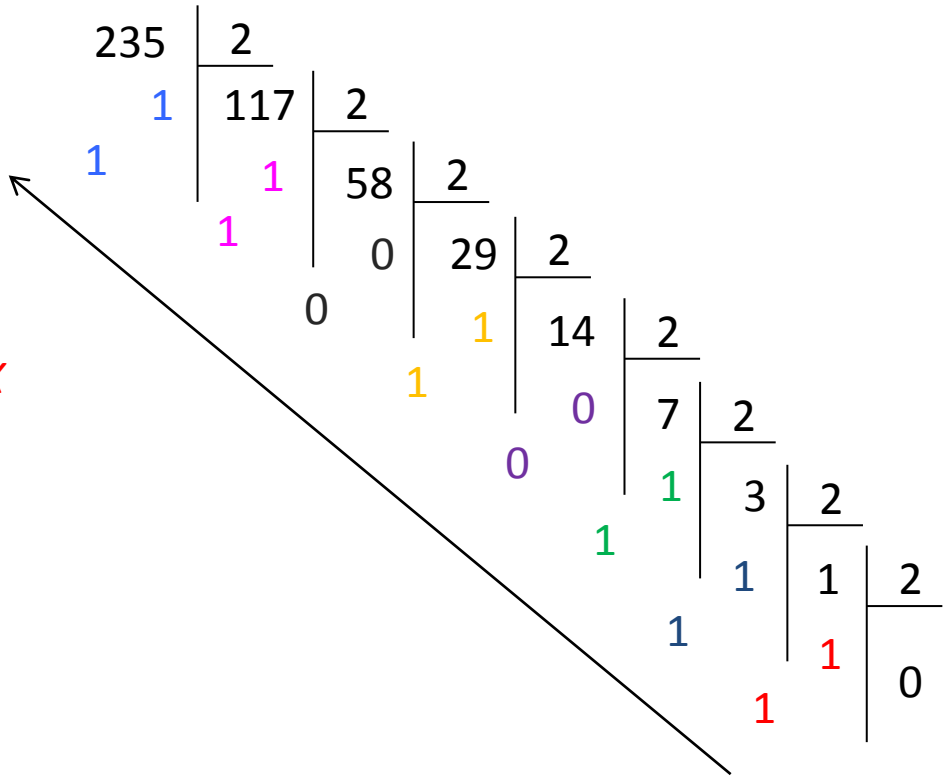
Partie fractionnaire : on multiplie PF(N) par X

0,625 * 2 = 1,25
0,25 * 2 = 0,5
0,5 * 2 = 1,0
0 * 2 = 0,0
…..

$(0,625)_{10} = (0, 1010000)_2$

Exemple : Convertir $(235,625)_{10}$ en base 2

$(235,625)_{10} = (11101011, 1010000)_2$



La représentation des informations sur la machine physique

- Les bases d'expression : conversions utiles

Base 10 → base X

- **Méthode des soustractions** : on soustrait à N la plus grande puissance de X qui lui est inférieure ou égale, puis on répète le processus sur la différence résultante jusqu'à obtenir 0.

Exemple : Convertir $(235,625)_{10}$ en base 2

On sait

$2^0 = 1$, $2^1 = 2$, $2^2 = 4$, $2^3 = 8$, $2^4 = 16$, $2^5 = 32$, $2^6 = 64$, $2^7 = 128$, $2^8 = 256$, $2^9 = 512$, $2^{10} = 1024$

$2^{-1} = ½ = 0,5$, $2^{-2} = ¼ = 0,25$, $2^{-3} = 1/8 = 0,125$, $2^{-4} = 1/16 = 0,0625$

$235 - 2^7 = 107$; $107 – 2^6 = 43$; $43 – 2^5 = 11$; $11 – 2^3 = 3$; $3 – 2^1 = 1$; $1 – 2^0 = 0$

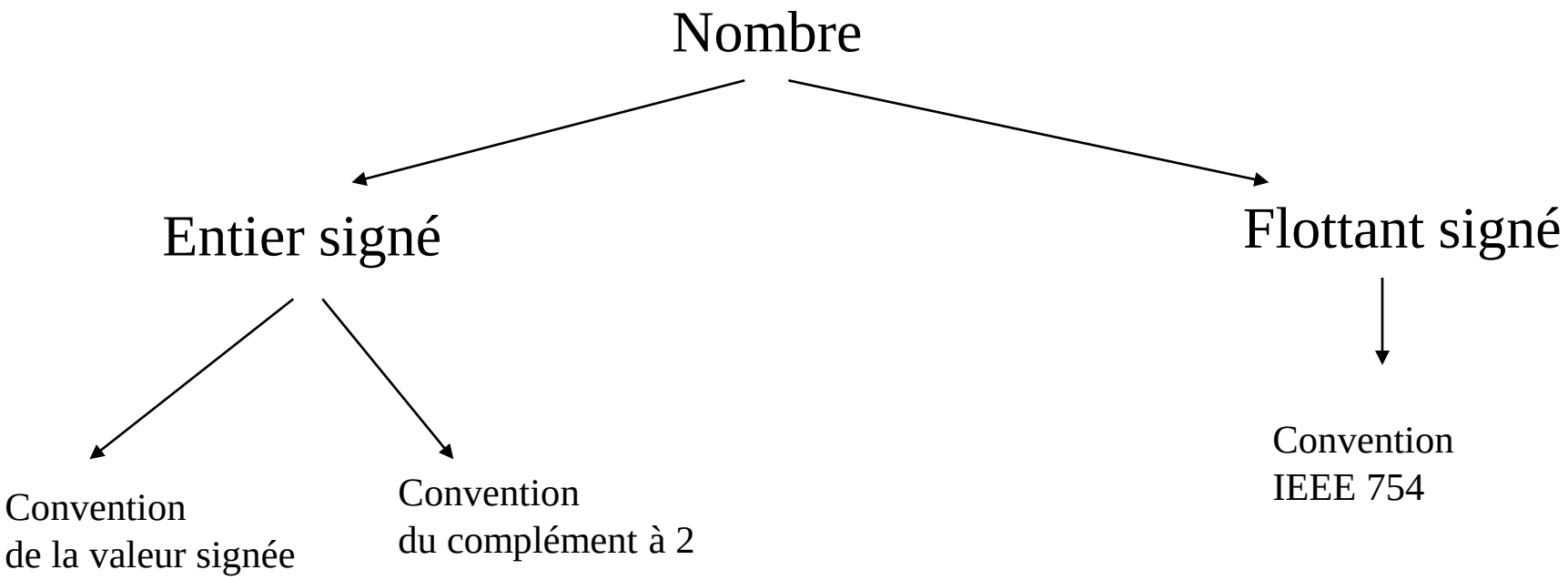
$235 = 2^7 + 2^6 + 2^5 + 2^3 + 2^1 + 2^0 = (11101011)_2$

$0,625 – 2^{-1} = 0,125$; $0,125 – 2^{-3} = 0$

$0,625 = (0,101)_2$

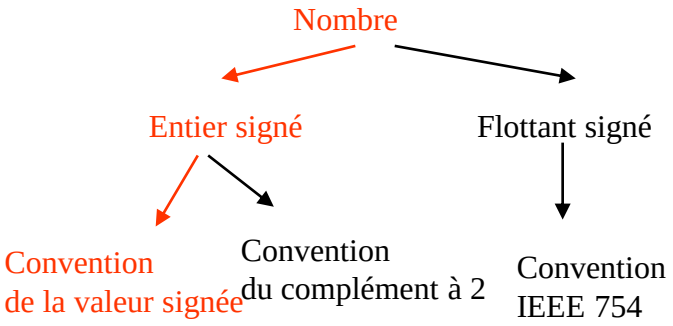
$(235,625)_{10} = (11101011,101)_2$

La représentation des informations sur la machine physique



La valeur d'un nombre signé est représentée par une chaine binaire de taille fixe (n bits).
en utilisant une convention de représentation

La représentation des informations sur la machine physique



Le bit de poids fort sert de bit de signe et vaut 1 si le nombre est négatif, 0 sinon.
Les autres bits codent la valeur absolue du nombre en binaire (base 2)

Exemple : représenter + 124 sur 8 bits en valeur signée

$124 = 64 + 32 + 16 + 8 + 4 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 = 01111100$

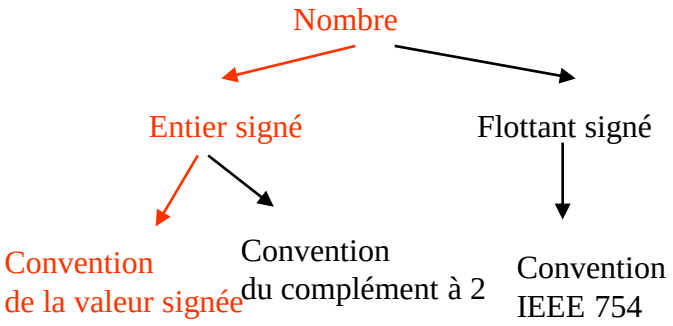
+124 = 01111100

Exemple : représenter - 124 sur 8 bits en valeur signée

$124 = 64 + 32 + 16 + 8 + 4 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 = 01111100$

- 124 = 11111100

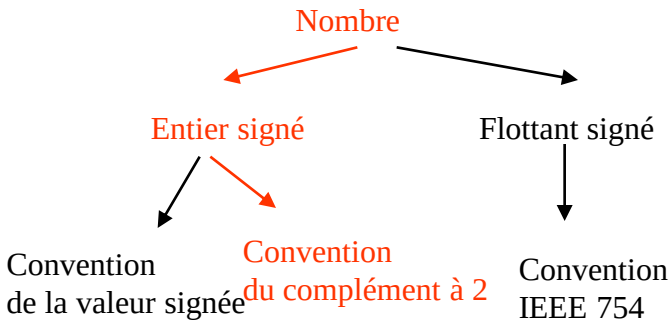
La représentation des informations sur la machine physique



Toute chaine binaire commençant par un 1 est un nombre négatif : sur 8 bits ce sont les chaines 11111111 (-127) à 10000000 (- 0)
Toute chaine binaire commençant par un 0 est un nombre positif : sur 8 bits ce sont les chaines 01111111 (+127) à 00000000 (+ 0)

! L'arithmétique des ordinateurs est différente de l'arithmétique des humains : l'ensemble des nombres représentables est contraint.
Ici il y a deux représentations du 0 !

La représentation des informations sur la machine physique



Un nombre positif est représenté par son équivalent binaire sur n bits.
Un nombre négatif est représenté en prenant le complément à 2 de son équivalent positif.

Exemple : représenter + 124 sur 8 bits en complément à 2

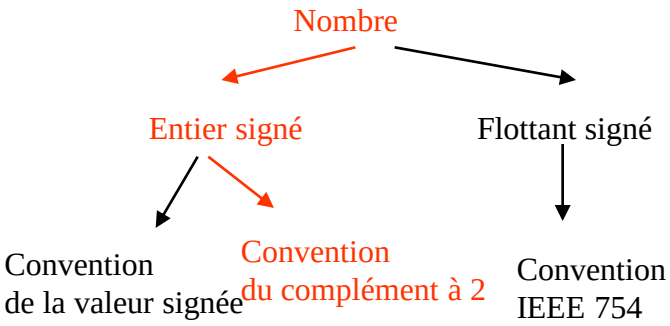
$124 = 64 + 32 + 16 + 8 + 4 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 = 01111100$

Exemple : représenter - 124 sur 8 bits en complément à 2

$124 = 64 + 32 + 16 + 8 + 4 = 2^6 + 2^5 + 2^4 + 2^3 + 2^2 = 01111100$

```
  01111100
  10000011   inversion des bits
+        1   ajout de 1
  ----------
  10000100  = - 124
```

La représentation des informations sur la machine physique



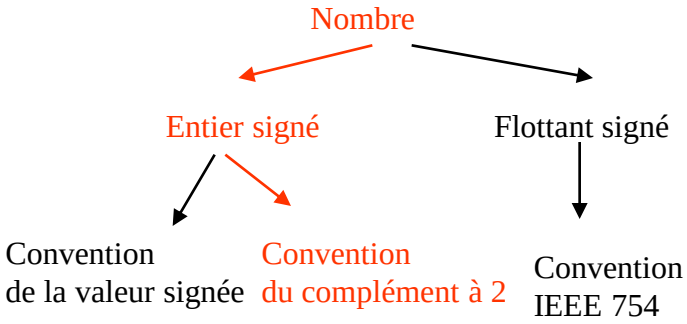
Toute chaine binaire commençant par un 1 est un nombre négatif : sur 8 bits ce sont les chaines 11111111 (-1) à 10000000 (- 128)
Toute chaine binaire commençant par un 0 est un nombre positif : sur 8 bits ce sont les chaines 01111111 (+127) à 00000000 (+ 0)

```
 11111111
 00000000
+        1
----------
 00000001
```

```
 11111111
-        1
----------
 11111110
 00000001
```

```
 10000000
 01111111
+        1
----------
 10000000
```

La représentation des informations sur la machine physique



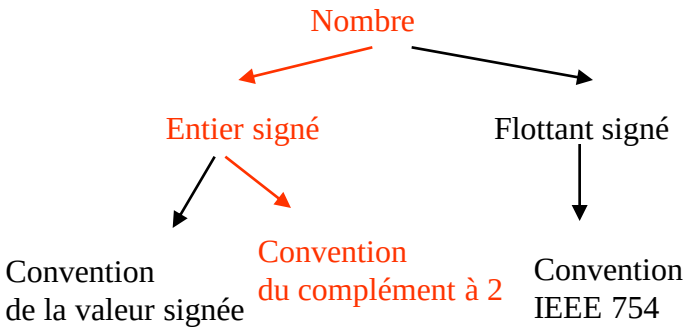
Exemple : additionner + 127 et + 2 avec une représentation en complément à 2 sur 8 bits

```
  127        01111111
+   2      + 00000010
-------    ----------
  129        10000001   → ce résultat est un nombre négatif (-127) et non la valeur 129
                          le résultat est trop grand pour la machine : il y a OVERFLOW
```

Le résultat est faux !

Éventail des nombres représentables [- 128, 127]

La représentation des informations sur la machine physique



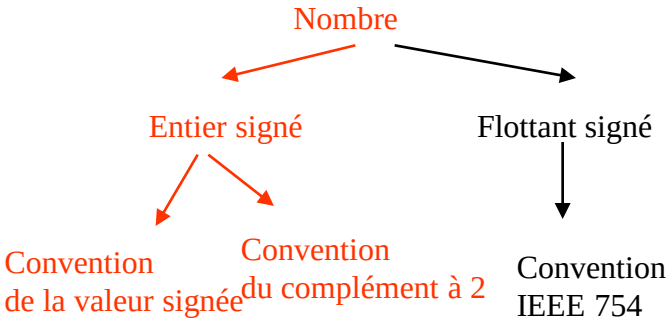
Exemple : additionner + 127 et - 2 avec une représentation en complément à 2 sur 8 bits

```
  127        01111111
-   2    + 11111110
-------   ---------------
  125      101111101  ---> le résultat ne tient pas sur 8 bits : il y a CARRY
```

Le résultat est sur 9 bits !

Éventail des nombres représentables [- 128, 127]

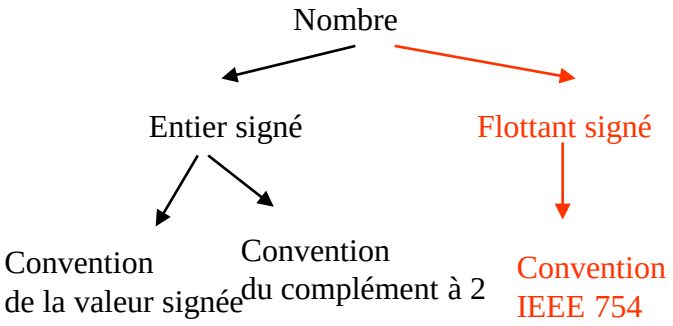
La représentation des informations sur la machine physique



- Nombres entiers signés sur n bits

	8 bits	p bits
Valeur signée	[-127, + 127]	$[– 2^{p-1} – 1, + 2^{p-1} – 1]$
Complément à 2	[-128, + 127]	$[– 2^{p-1}, + 2^{p-1} – 1]$

La représentation des informations sur la machine physique



- <u>Représentation en virgule flottante</u>
- Un nombre est représenté en virgule flottante dans la base B s'il est sous la forme

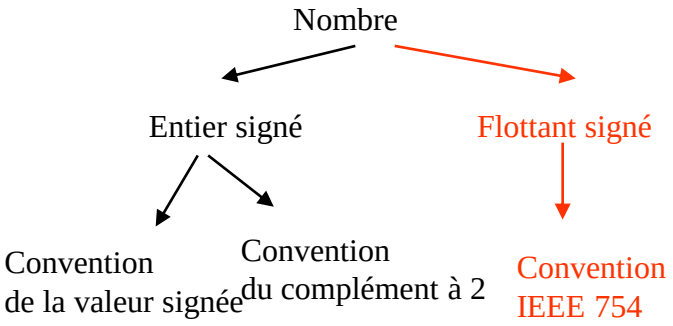
$$\pm M1,M2 \times B^{\pm c}$$

M1,M2 est la mantisse; B est la base; c est l'exposant.

- *Exemple* : $+59,4151 * 10^{-5}$

- Il s'agit de représenter la mantisse et son signe, ainsi que l'exposant et son signe.

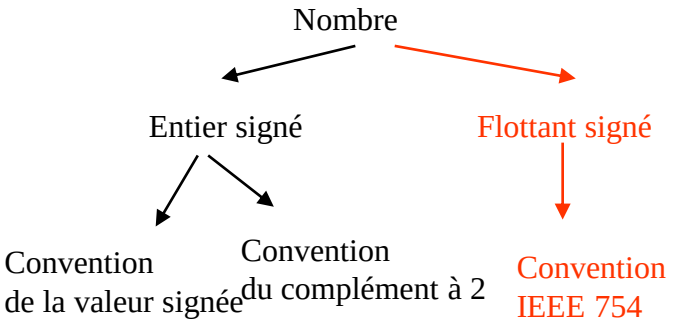
La représentation des informations sur la machine physique



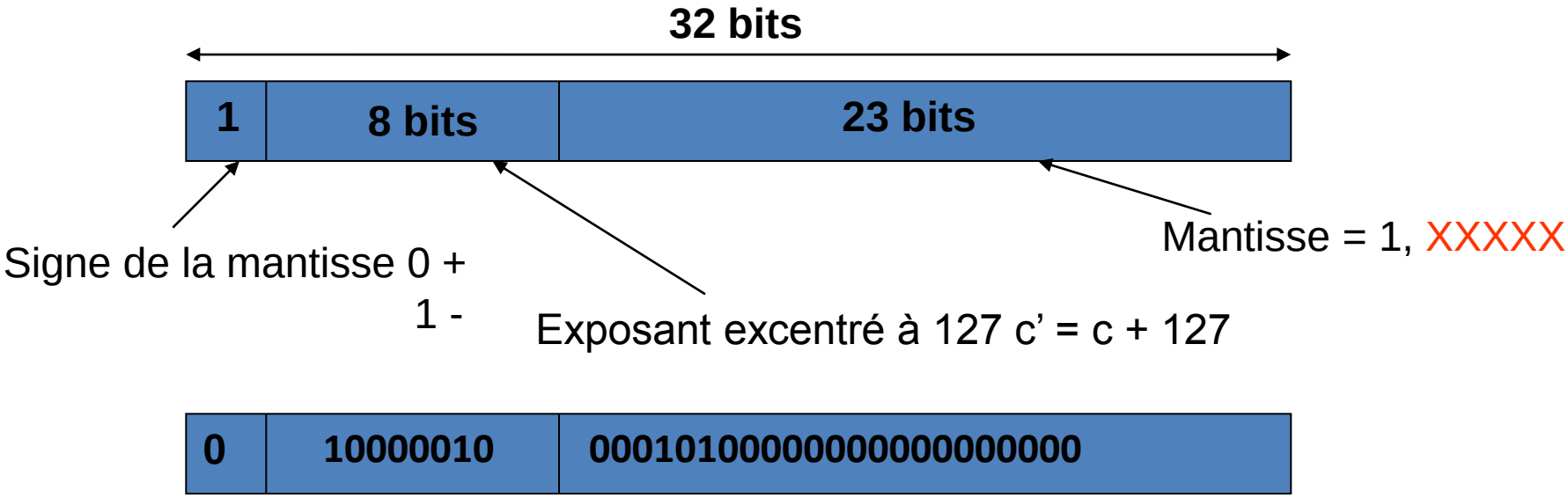
- Représentation IEEE 754 : $\pm 1,XXXX \times 2^c$

  - Deux formats de représentation : simple précision sur 32 bits; double précision sur 64 bits
  - La mantisse est normalisée sous la forme 1,XXXX (pseudo-mantisse) et le 1 n'est pas codé en machine (bit caché). Le signe est codé sur un bit valant 0 si la mantisse est positive, et 1 si elle est négative.
  - L'exposant est codé avec un excédant à 127 (SP) et 1023 (DP) de manière à coder une valeur positive. L'exposant c' codé est alors égal à c + 127 ou 1023.

La représentation des informations sur la machine physique



- Représentation IEEE 754 Simple Précision :



Représenter $(8,625)_{10}$

$(8,625)_{10} = (1000 + 0,101) = 1,000101 * 2^3$

exposant c' = 3 + 127 = 130 = $(10000010)^2$

Représenter $(8,625)_{10}$

$(8)_{10} = (1000)_2$

$(0, 625)_{10} = (0,101)_2$

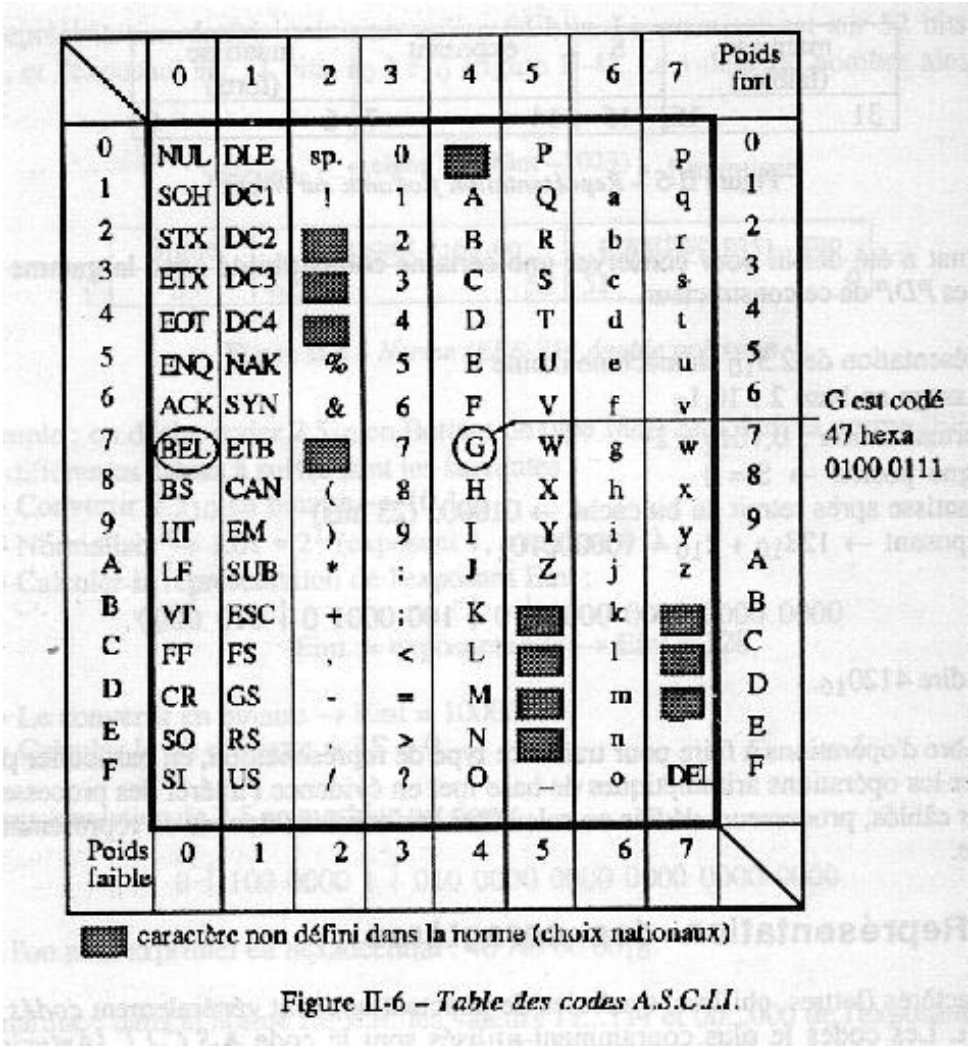
La représentation des caractères sur la machine physique

- La valeur d'un caractère est représentée par une chaine binaire de taille fixe.

- La valeur binaire de chaque caractère est stockée dans une table.

- Il existe trois grands codages pour les caractères :
  - le code ASCII (*American Standard Code for Information Interchange*) sur 7 bits
  - le code EBCDIC (*Extended Binary-Coded Decimal Interchange Code*), **développé par IBM** sur 8 bits
  - le code UNICODE **sur 16 bits mis au point en 1991. Il est compatible avec le code ASCII.**

La représentation des caractères sur la machine physique



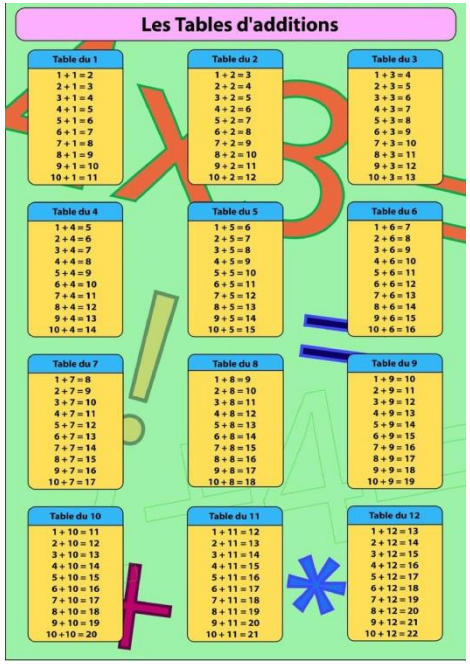
	0	1	2	3	4	5	6	7	Poids fort
0	NUL	DLE	sp.	0	■	P		p	0
1	SOH	DC1	!	1	A	Q	a	q	1
2	STX	DC2	■	2	B	R	b	r	2
3	ETX	DC3	■	3	C	S	c	s	3
4	EOT	DC4	■	4	D	T	d	t	4
5	ENQ	NAK	%	5	E	U	e	u	5
6	ACK	SYN	&	6	F	V	f	v	6
7	BEL	ETB	■	7	G	W	g	w	7
8	BS	CAN	(	8	H	X	h	x	8
9	HT	EM	)	9	I	Y	i	y	9
A	LF	SUB	*	:	J	Z	j	z	A
B	VT	ESC	+	;	K	■	k	■	B
C	FF	FS	,	<	L	■	l	■	C
D	CR	GS	-	=	M	■	m	■	D
E	SO	RS	.	>	N	■	n		E
F	SI	US	/	?	O	_	o	DEL	F
Poids faible	0	1	2	3	4	5	6	7	

G est codé 47 hexa 0100 0111

■ caractère non défini dans la norme (choix nationaux)

Figure II-6 – *Table des codes A.S.C.I.I.*

- Code ASCII
- Les codes 0 à 31 sont *caractères de contrôle. Ils* permettent de faire des actions telles que :
  - retour à la ligne (CR)
  - Bip sonore (BEL)
- Les codes 65 à 90 représentent les majuscules
- Les codes 97 à 122 représentent les minuscules



Le langage de l'ordinateur
Circuits logiques : un aperçu
Circuit additionneur n bits

r_{n-1} r_{n-2} ... r_2 r_1 r_0

a_{n-1} a_{n-2} ... a_2 a_1 a_0 Opérande A

b_{n-1} b_{n-2} ... b_2 b_1 b_0 Opérande B

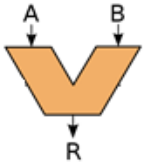
---

c_{n-1} c_{n-2} ... c_2 c_1 c_0 Résultat C

r_{n-1} r_{n-2} ... r_2 r_1 r_0 Retenues

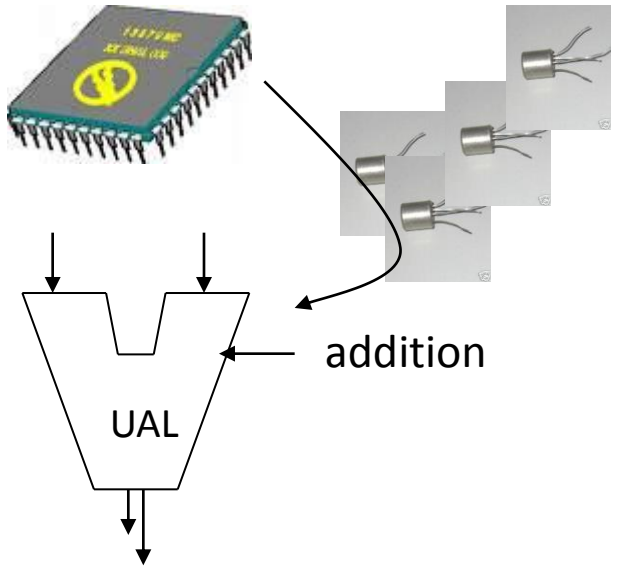
01001111
+ 10011010

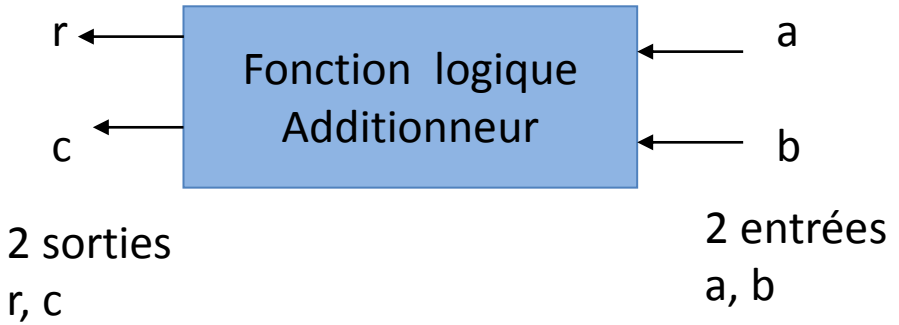
L'additionneur est un circuit de l'Unité Arithmétique et Logique (UAL)



Circuits logiques et algèbre de boole

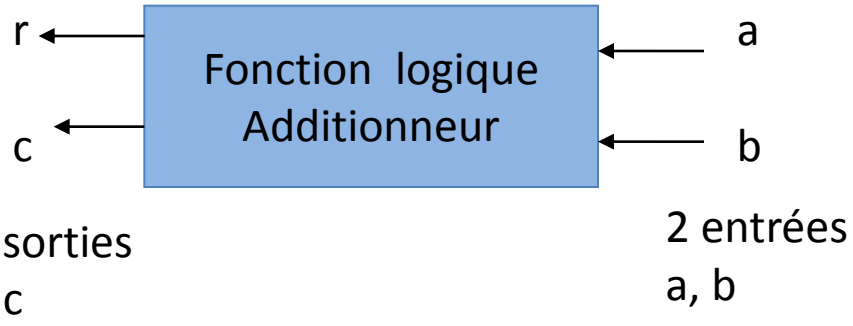
- Circuits intégrés = ensemble de transistors à 2 états : 0, 1
- On associe à une donnée binaire une variable logique.
- Les transistors assemblés constituent des traitements sur les variables logiques : le traitement est décrit par une fonction logique.
- Une fonction logique est une boite noire admettant en entrée 1 à n variables logiques et rendant en sortie 1 à n variables de sortie.





Circuits logiques et algèbre de boole

- Une fonction logique est décrite par une table de vérité.



a	b	c	r
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

La table de vérité précise les états de sorties en fonction des entrées.

Toute fonction logique peut être décrite à partir de quatre fonctions de base :

- OU logique (+)
- ET logique (.)
- NON ($\overline{A}$)
- OU exclusif ($\oplus$)

Circuits logiques et algèbre de boole

- Fonction logique de base :

  A chaque fonction correspond un circuit logique encore appelé porte logique constitué de transistors.

	a b s 0 0 0 0 1 1 1 0 1 1 1 0	XOR s est à 1 si a ou b sont à 1. $s = a \oplus b$

	a b s 0 0 0 0 1 0 1 0 0 1 1 1	ET s est à 1 si a et b sont tous deux à 1. s = a.b
	a s 0 1 1 0	NON L'état de s est opposé à celui de a s = $\overline{a}$
	a b s 0 0 0 0 1 1 1 0 1 1 1 1	OU s est à 1 si a ou b sont à 1 ou a et b sont tous deux à 1. s = a + b

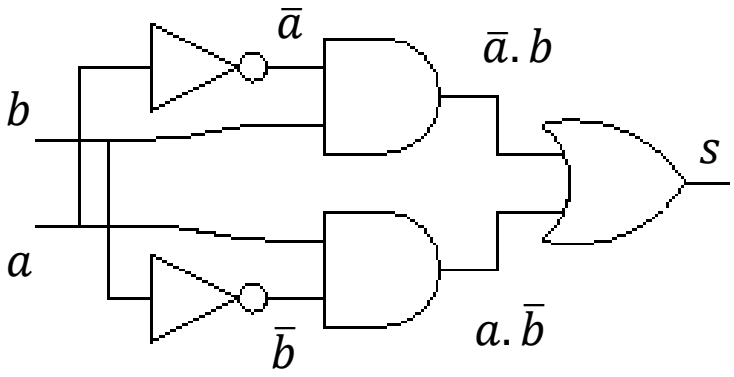
Circuits logiques et algèbre de boole

- Un petit exemple : la porte XOR

a	b	s
0	0	0
0	1	1
1	0	1
1	1	0

XOR
s est à 1 si a ou b sont à 1.
$s = a \oplus b$

$s = \bar{a}.b + a.\bar{b}$



a	b	s
0	0	0
0	1	0
1	0	0
1	1	1

ET
s est à 1 si a et b sont tous deux à 1.
s = a.b

a	s
0	1
1	0

NON
L'état de s est opposé à celui de a
s = $\bar{a}$

a	b	s
0	0	0
0	1	1
1	0	1
1	1	1

OU
s est à 1 si a ou b sont à 1 ou a et b sont tous deux à 1.
s = a + b

L'additionneur poids faible

a_0	b_0	c_0	r_0
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

a	b	s
0	0	0
0	1	1
1	0	1
1	1	0

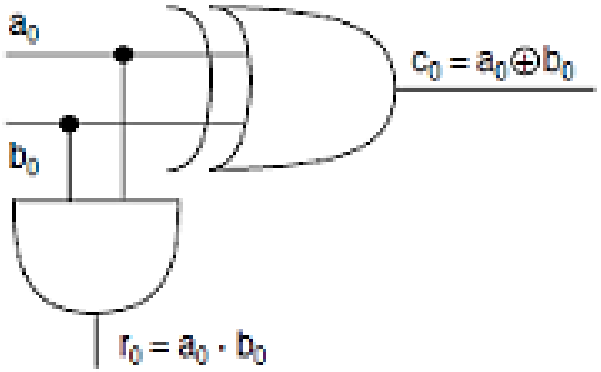
XOR
s est à 1 si a ou b sont à 1.

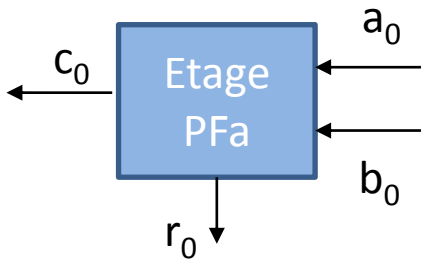
$$s = a \oplus b$$

a	b	s
0	0	0
0	1	0
1	0	0
1	1	1

ET
s est à 1 si a et b sont tous deux à 1.
s = a.b

- $c_0 = a_0$ XOR b_0
- $r_0 = a_0$ ET b_0



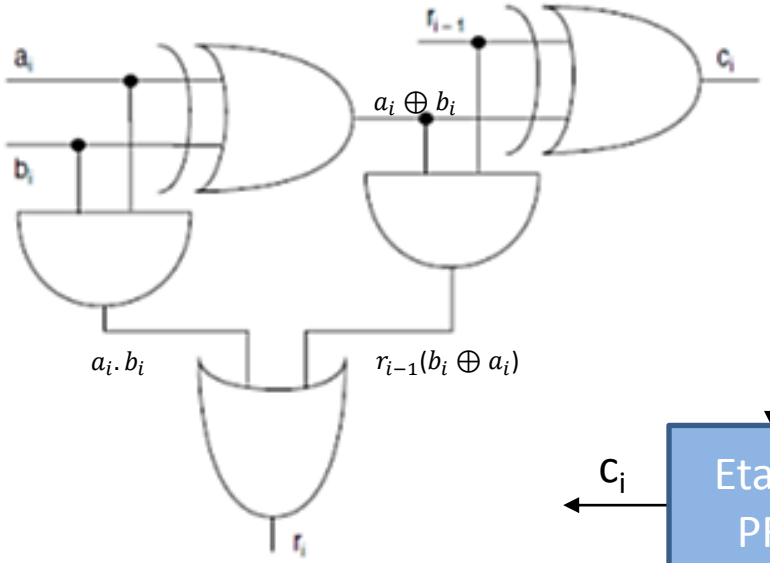


L'additionneur poids fort

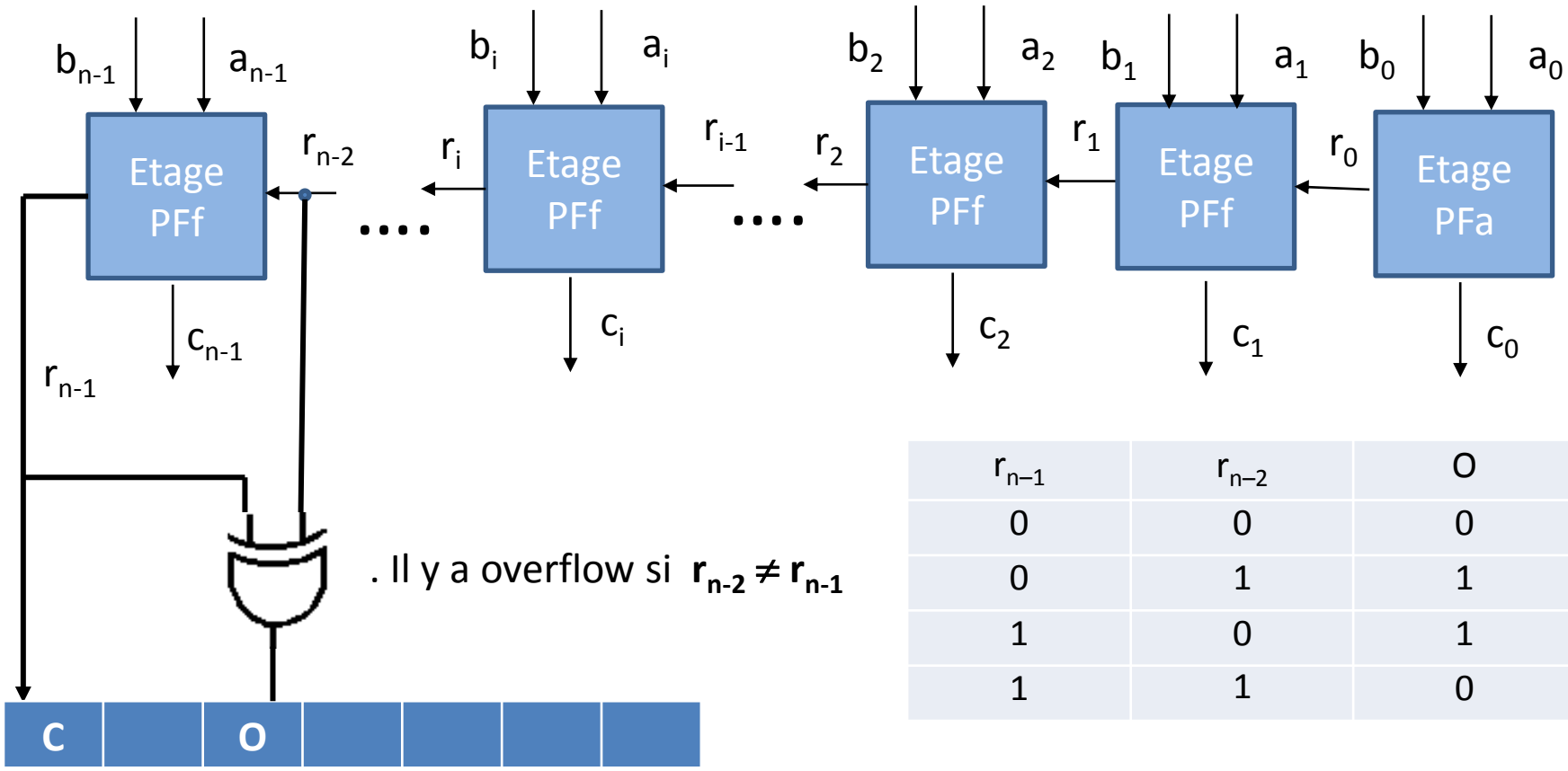
a_1	b_1	r_0	c_1	r_1
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

$$c_1 = \overline{a_1}.\overline{b_1}.r_0 + \overline{a_1}.\overline{r_0}.b_1 + \overline{b_1}.\overline{r_0}.a_1 + a_1.r_0.b_1$$
$$c_1 = a_1 \oplus b_1 \oplus r_0$$

$$r_1 = a_1.\overline{b_1}.r_0 + a_1.r_0.b_1 + b_1.\overline{a_1}.r_0 + a_1.\overline{r_0}.b_1$$
$$r_1 = a_1.b_1 + r_0(b_1 \oplus a_1)$$



L'additionneur complet n bits



. Il y a overflow si $\mathbf{r_{n-2} \neq r_{n-1}}$

r_{n-1}	r_{n-2}	O
0	0	0
0	1	1
1	0	1
1	1	0

Indicateurs de l'UAL

Table de vérité pour l'indicateur d'overflow O
Ce qui correspond à une porte XOR.
Donc $O = r_{n-2} \oplus r_{n-1}$.