

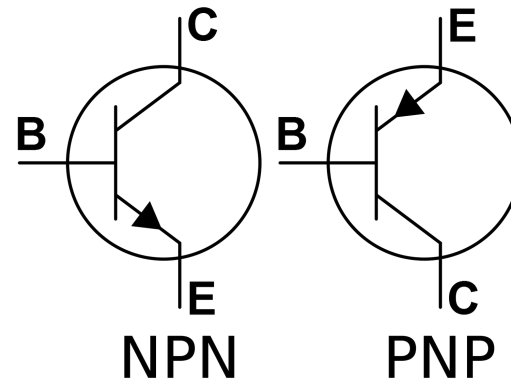
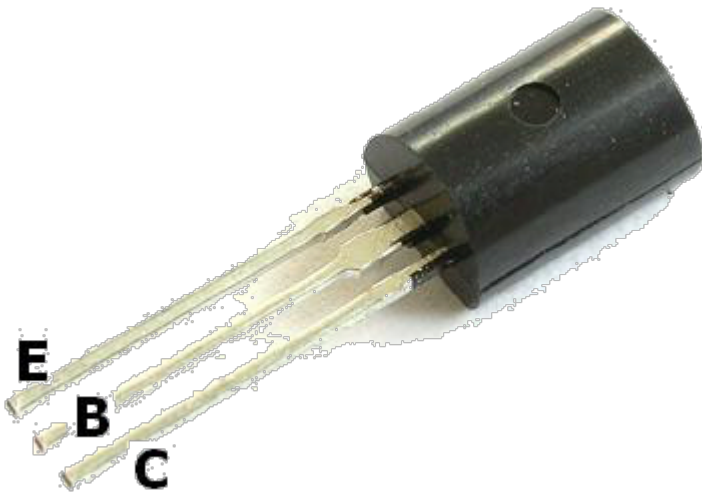
Principe de fonctionnement d'un ordinateur, codage de l'information

**Raymond Namyst
Marc Zeitoun
Université de Bordeaux**

Au niveau électronique

Des transistors... beaucoup de transistors

- Transistor = télerupteur miniature
 - Sorte de robinet de courant électrique
 - On peut commander le passage du courant



Représentation de l'information

Des histoires de 0 et de 1

- Au niveau électrique, on distingue deux états
 - Courant qui passe / courant qui ne passe pas
- Représentation *binnaire* de l'information
 - Deux chiffres seulement: 0 et 1
 - 0 : absence de courant
 - 1 : présence de courant
 - **BIT** = *Binary Digit*
- Un ordinateur est capable
 - De stocker une grande quantité de 0 et de 1
 - Dont certains sont reçus depuis l'extérieur
CDROM, clé USB, clavier, etc.
 - D'effectuer très rapidement des calculs sur des ensembles de bits

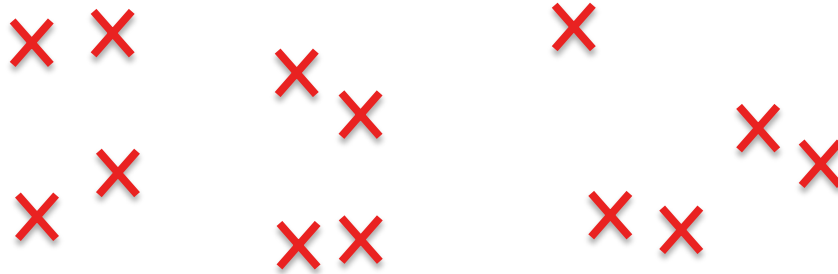
Représentation de l'information

Du binaire au codage en base 2

- Pour manipuler des objets plus complexes, les ordinateurs *regroupent* les BITS par paquets indivisibles
 - **Octet** = paquet de 8 BITS
 - **Mot** = ensemble de plusieurs octets
 - Exemple : « un ordinateur 32 bits » est un ordinateur qui manipule des mots de 4 octets ($4 \times 8 = 32$)
 - Aujourd'hui, la mode est plutôt aux ordinateurs « 64 bits »
- Représentation des entiers positifs
 - 0, 1, 10, 11, 100, 101, 110, 111, 1000, 1001, 1010, 1011, 1100, etc.
 - Exemple : le nombre décimal 13 est « codé » 1101 en binaire (donc 00000000 00000000 00000000 00001101 sur une machine 32 bits)

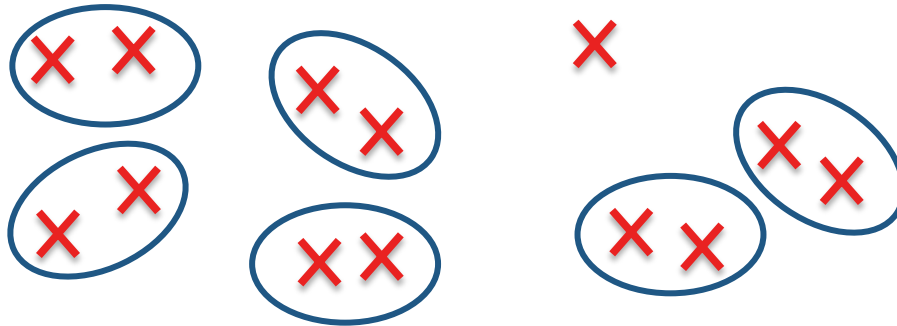
Représentation de l'information

À une époque, à l'école primaire...



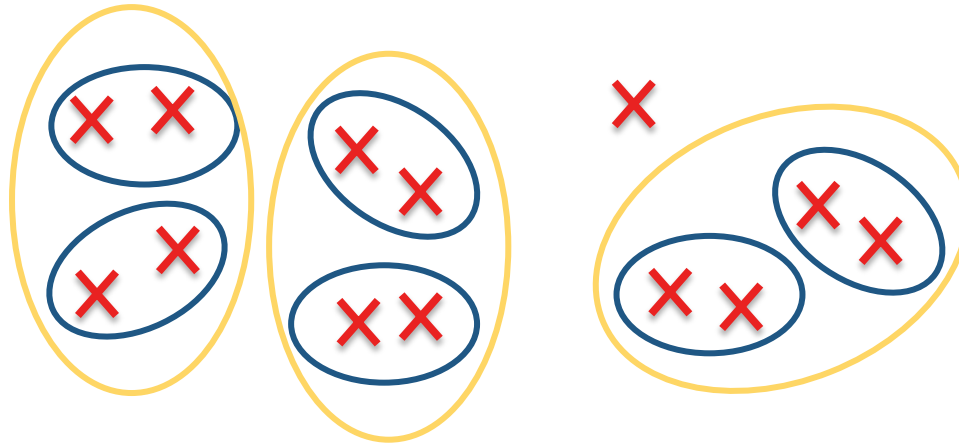
Représentation de l'information

À une époque, à l'école primaire...



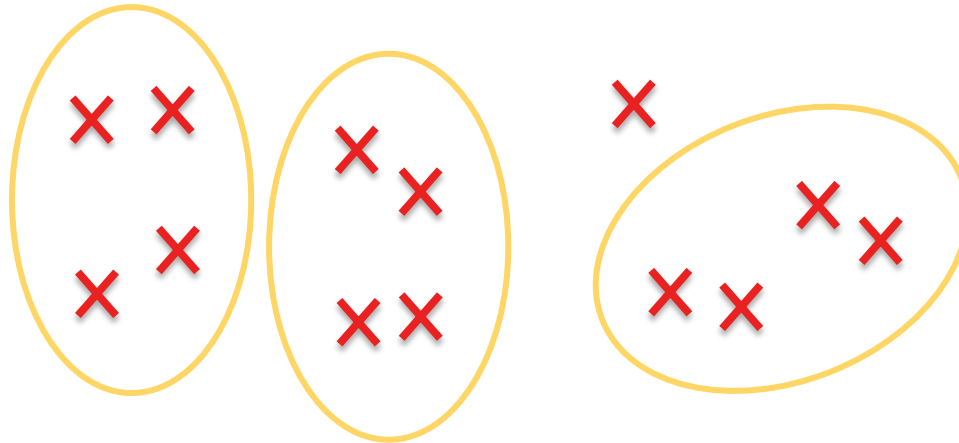
Représentation de l'information

À une époque, à l'école primaire...



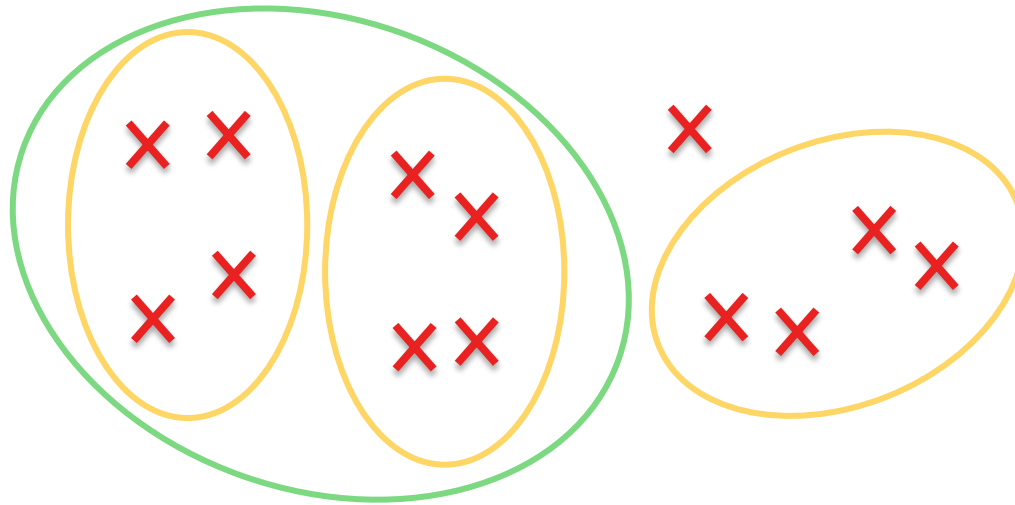
Représentation de l'information

À une époque, à l'école primaire...



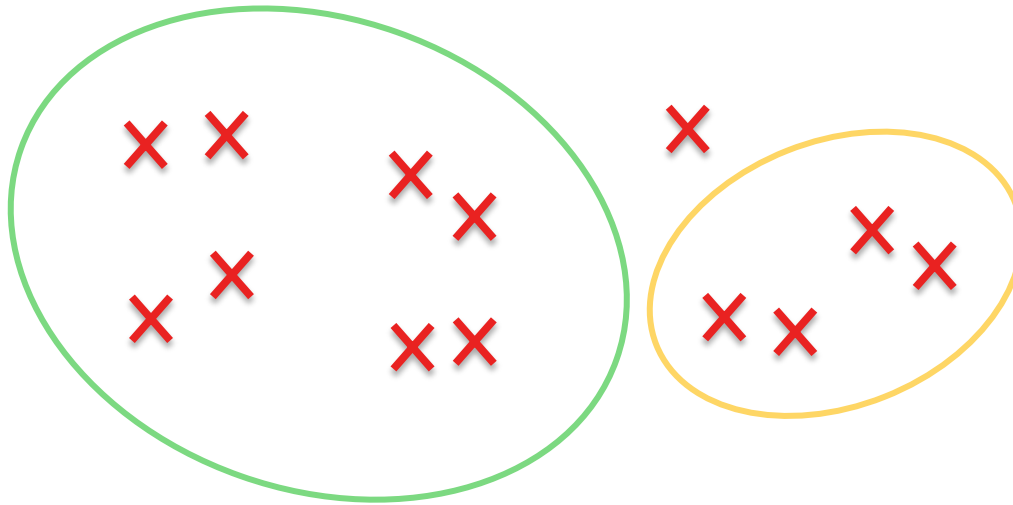
Représentation de l'information

À une époque, à l'école primaire...



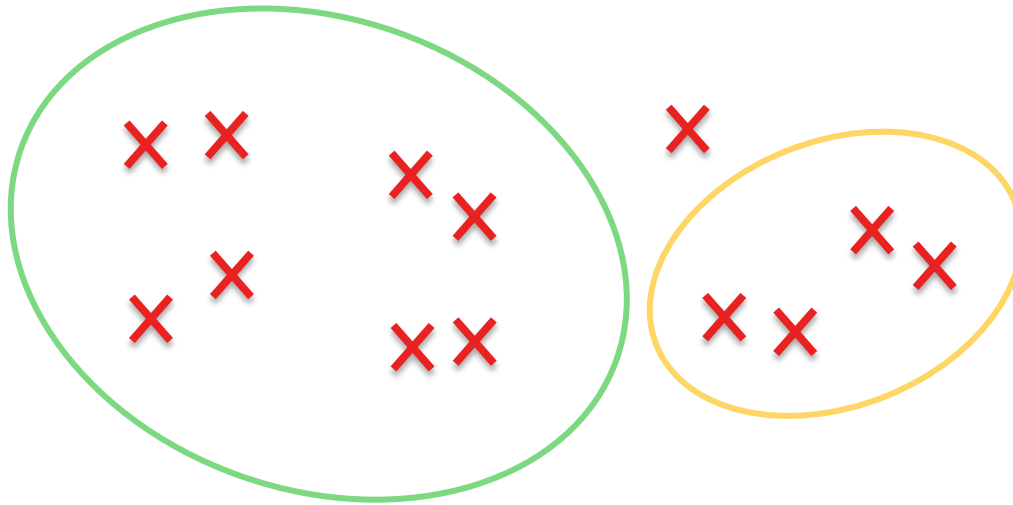
Représentation de l'information

À une époque, à l'école primaire...



Représentation de l'information

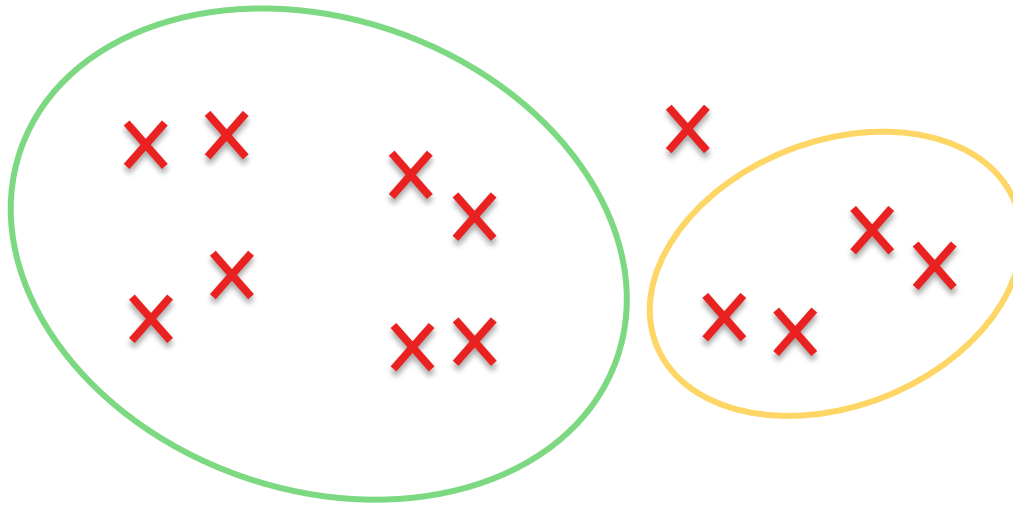
À une époque, à l'école primaire...



1101

Représentation de l'information

À une époque, à l'école primaire...



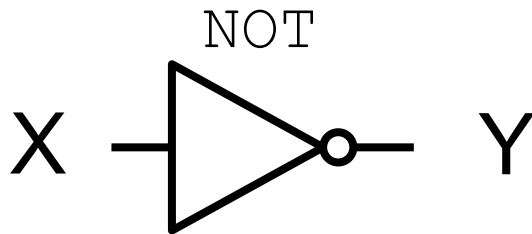
$$\begin{array}{r} 1101 \\ = 1000 \quad (8) \\ + 100 \quad (4) \\ + 1 \quad (1) \end{array}$$

**Principe de fonctionnement d'un
ordinateur :
les circuits élémentaires**

Au niveau électronique

Des transistors... aux portes logiques

- En combinant plusieurs transistors, on peut former des circuits logiques élémentaires : les *portes logiques*
- Exemple : la porte « NON »
 - Inverseur

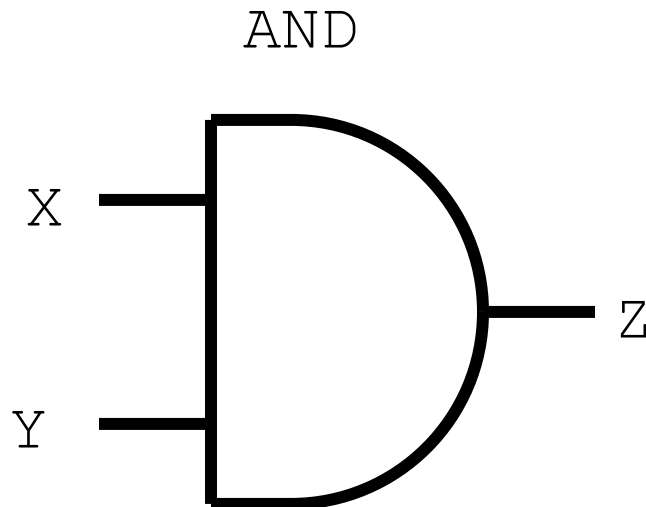


X	Y
0	1
1	0

Au niveau électronique (suite)

Des transistors... aux portes logiques

- La porte « ET »
 - La sortie vaut 1 *si et seulement si* les deux entrées valent 1

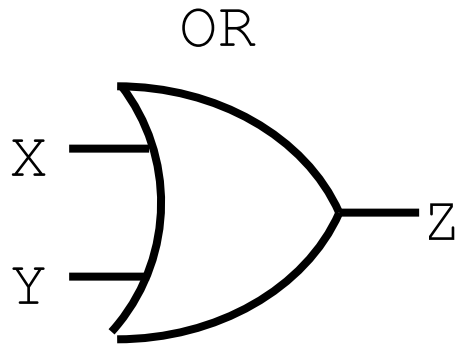


X	Y	Z
0	0	0
0	1	0
1	0	0
1	1	1

Au niveau électronique (suite)

Des transistors... aux portes logiques

- La porte « OU »
 - La sortie vaut 1 si l'une des deux entrées vaut 1

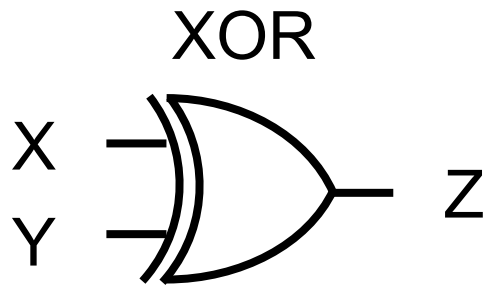


X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	1

Au niveau électronique (suite)

Des transistors... aux portes logiques

- La porte « OU EXCLUSIF »
 - La sortie vaut 1 si et seulement si uniquement l'une des deux entrées vaut 1

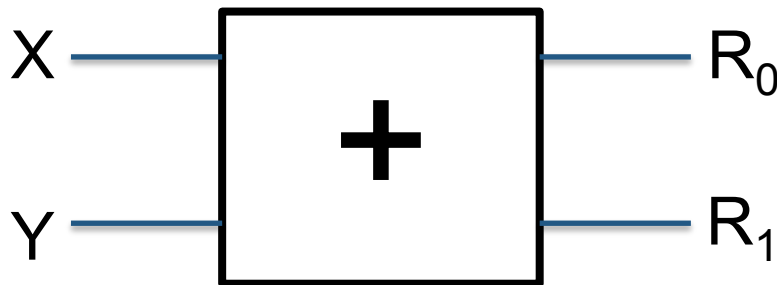


X	Y	Z
0	0	0
0	1	1
1	0	1
1	1	0

Calculer en binaire

Des portes logiques aux circuits

- En construisant des assemblages de portes logiques, on peut imaginer toutes sortes de circuits !
- Par exemple, un additionneur « 1 bit »
 - On veut un circuit à deux entrées (X et Y) et à deux sorties (les deux bits du résultat)

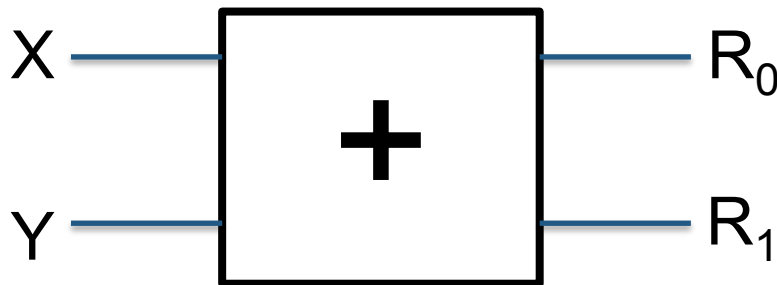


X	Y	R ₁	R ₀
0	0	0	0
0	1		
1	0		
1	1		

Calculer en binaire

Des portes logiques aux circuits

- En construisant des assemblages de portes logiques, on peut imaginer toutes sortes de circuits !
- Par exemple, un additionneur de 2 bits X et Y
 - On veut un circuit à deux entrées (X et Y) et à deux sorties (les deux bits du résultat)

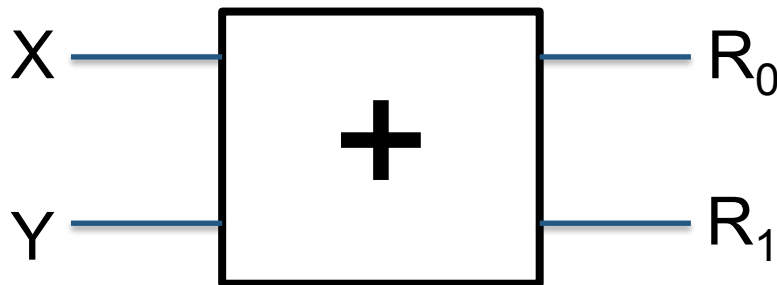


X	Y	R ₁	R ₀
0	0	0	0
0	1	0	1
1	0		
1	1		

Calculer en binaire

Des portes logiques aux circuits

- En construisant des assemblages de portes logiques, on peut imaginer toutes sortes de circuits !
- Par exemple, un additionneur « 1 bit »
 - On veut un circuit à deux entrées (X et Y) et à deux sorties (les deux bits du résultat)

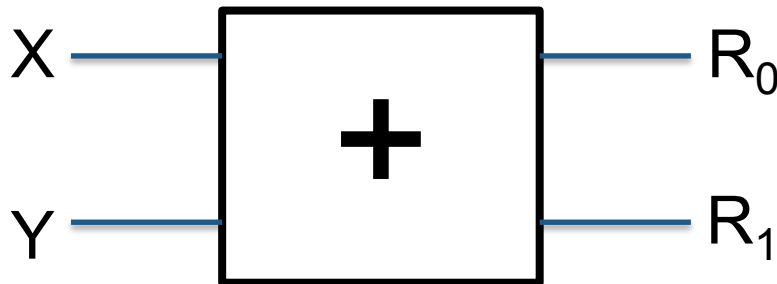


X	Y	R ₁	R ₀
0	0	0	0
0	1	0	1
1	0	0	1
1	1		

Calculer en binaire

Des portes logiques aux circuits

- En construisant des assemblages de portes logiques, on peut imaginer toutes sortes de circuits !
- Par exemple, un additionneur « 1 bit »
 - On veut un circuit à deux entrées (X et Y) et à deux sorties (les deux bits du résultat)



X	Y	R ₁	R ₀
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Calculer en binaire

Des portes logiques aux circuits

- Un additionneur « 1 bit » = ?

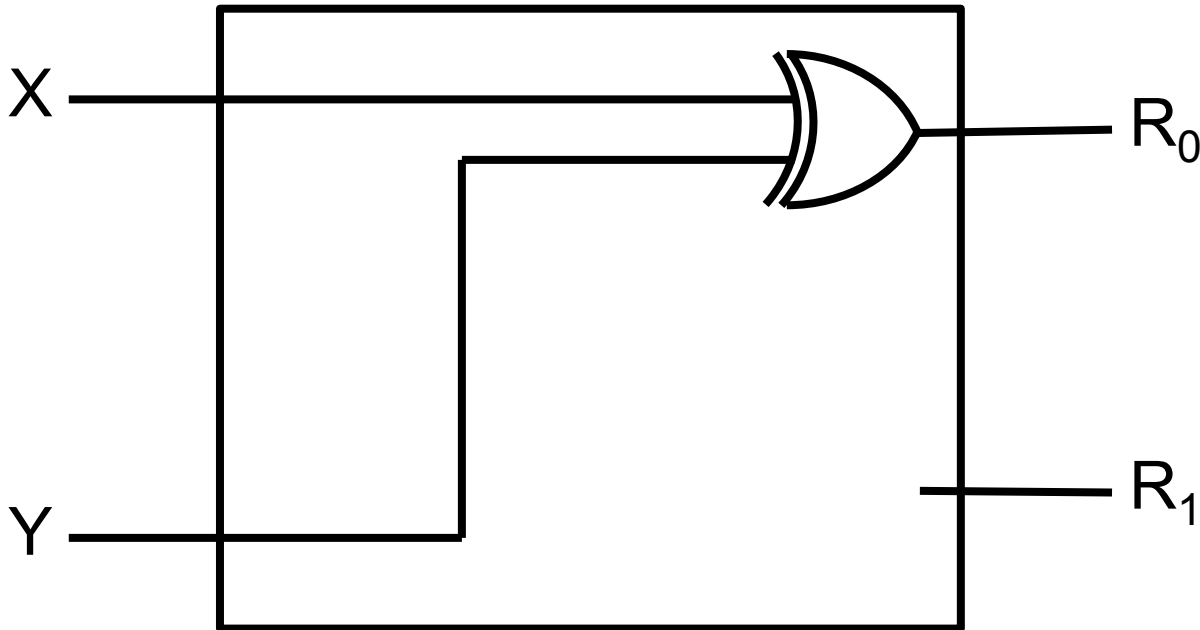


X	Y	R ₁	R ₀
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Calculer en binaire

Des portes logiques aux circuits

- Un additionneur « 1 bit » = une porte OU EXCLUSIF + ?

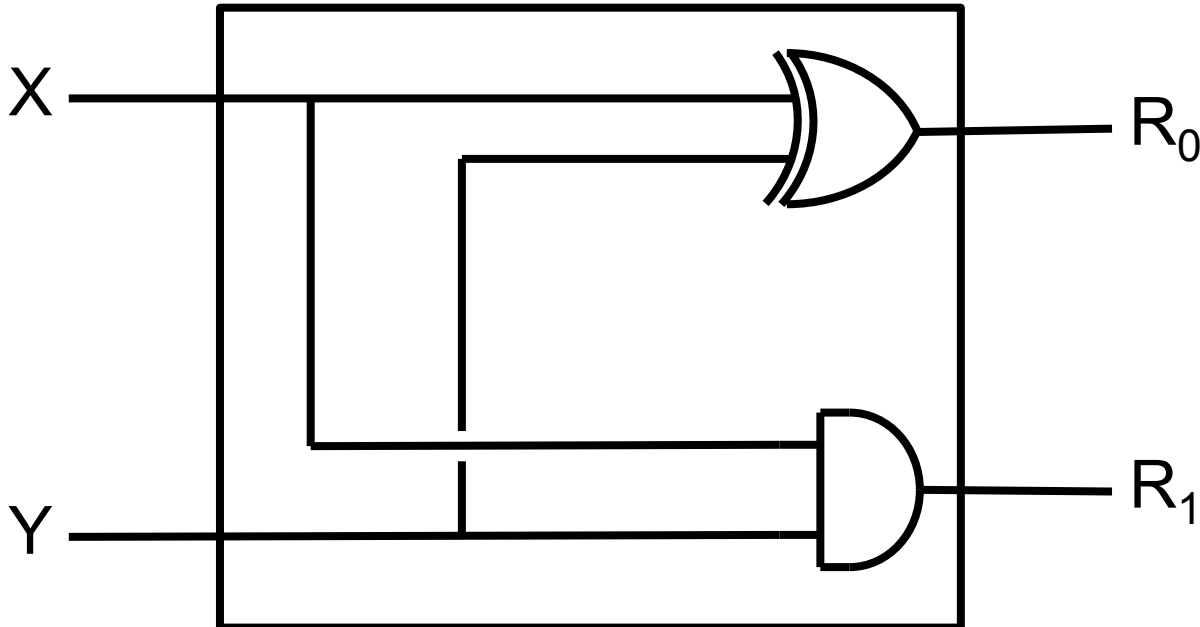


X	Y	R ₁	R ₀
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Calculer en binaire

Des portes logiques aux circuits

- Un additionneur « 1 bit » = une porte OU EXCLUSIF + une porte ET

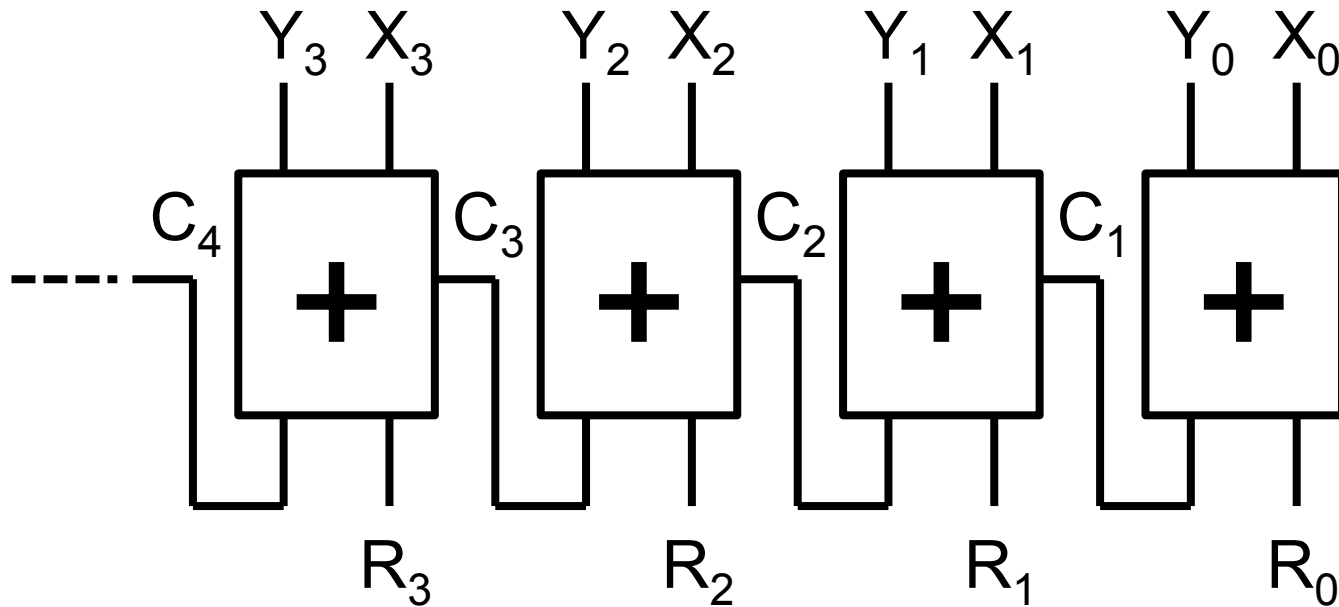


X	Y	R ₁	R ₀
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Calculer en binaire

Des portes logiques aux circuits

- Pour additionner des nombres de plusieurs bits, il faut cascader des additionneurs de 3 bits
 - Mais c'est une autre histoire...

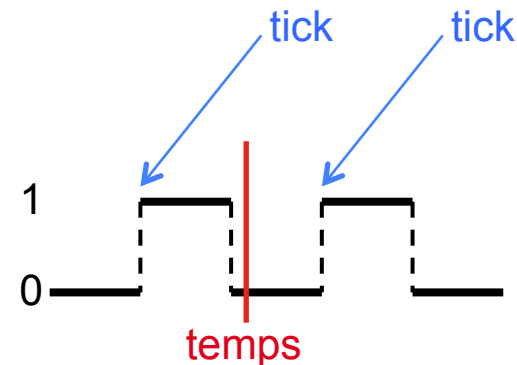
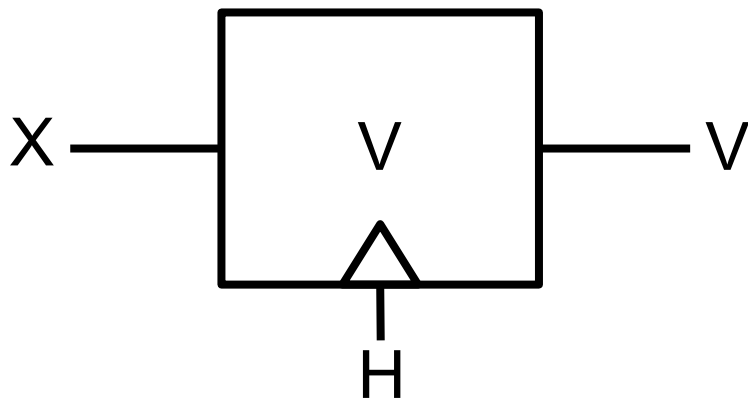


**Principe de fonctionnement d'un
ordinateur :
le microprocesseur**

Au niveau électronique (suite)

Des portes logiques aux circuits

- Par assemblage de portes logiques et de circuits, on parvient à construire des circuits de plus en plus complexes
 - Unités arithmétiques et logiques (addition, soustraction, multiplication, division, etc.)
 - Multiplexeurs et démultiplexeurs (*aiguillage* de l'information)
 - Y compris des « mémoires » qui ne changent d'état que lors d'un « tick d'horloge »



Au niveau électronique (suite)

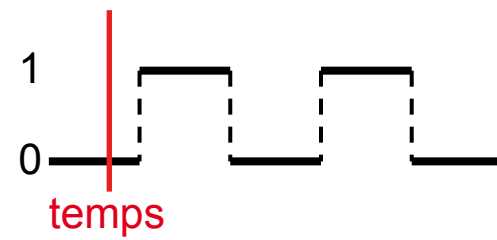
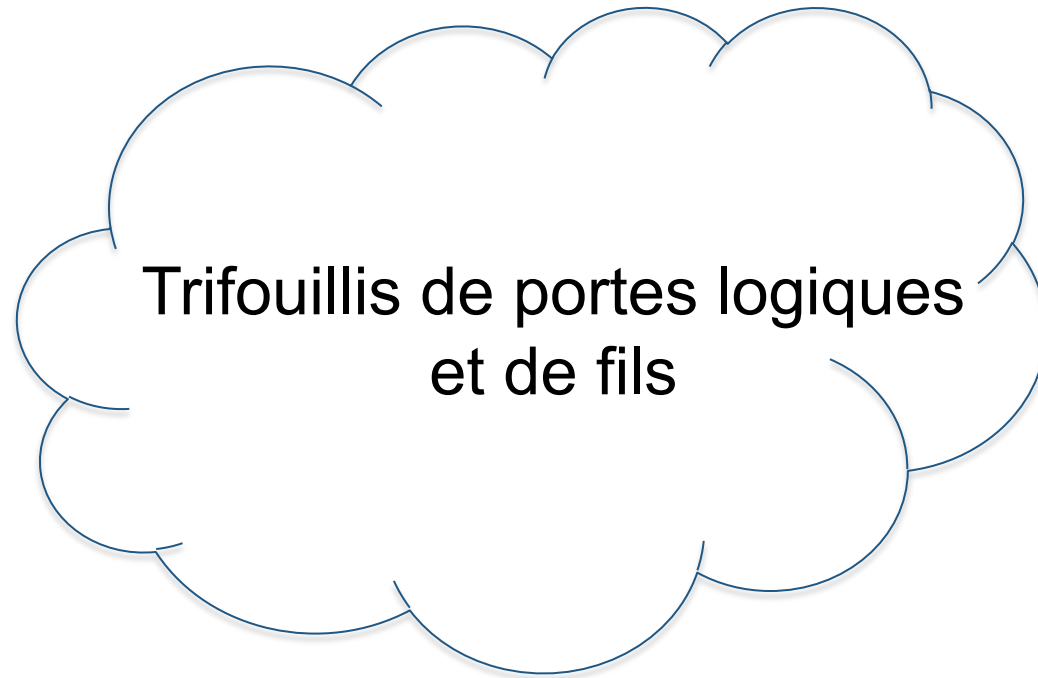
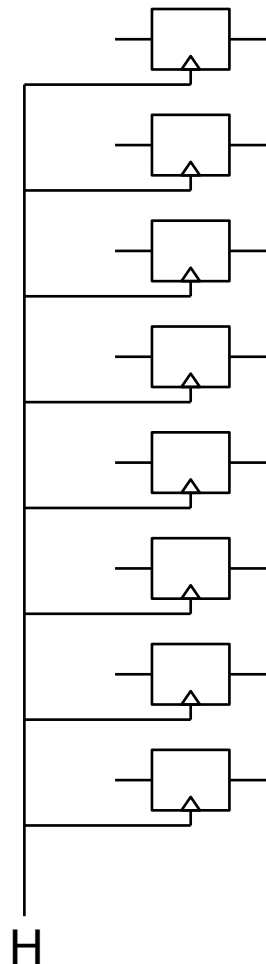
Notion de cycle

A cloud-shaped graphic with a blue outline and a light blue shadow, containing the text "Trifouillis de portes logiques et de fils".

Trifouillis de portes logiques
et de fils

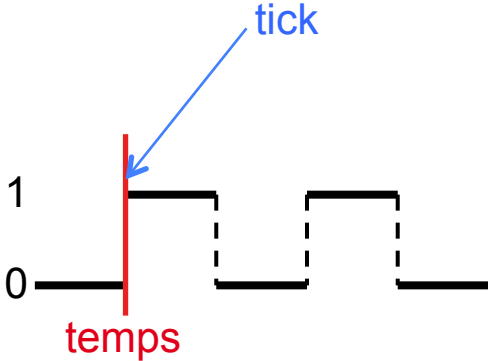
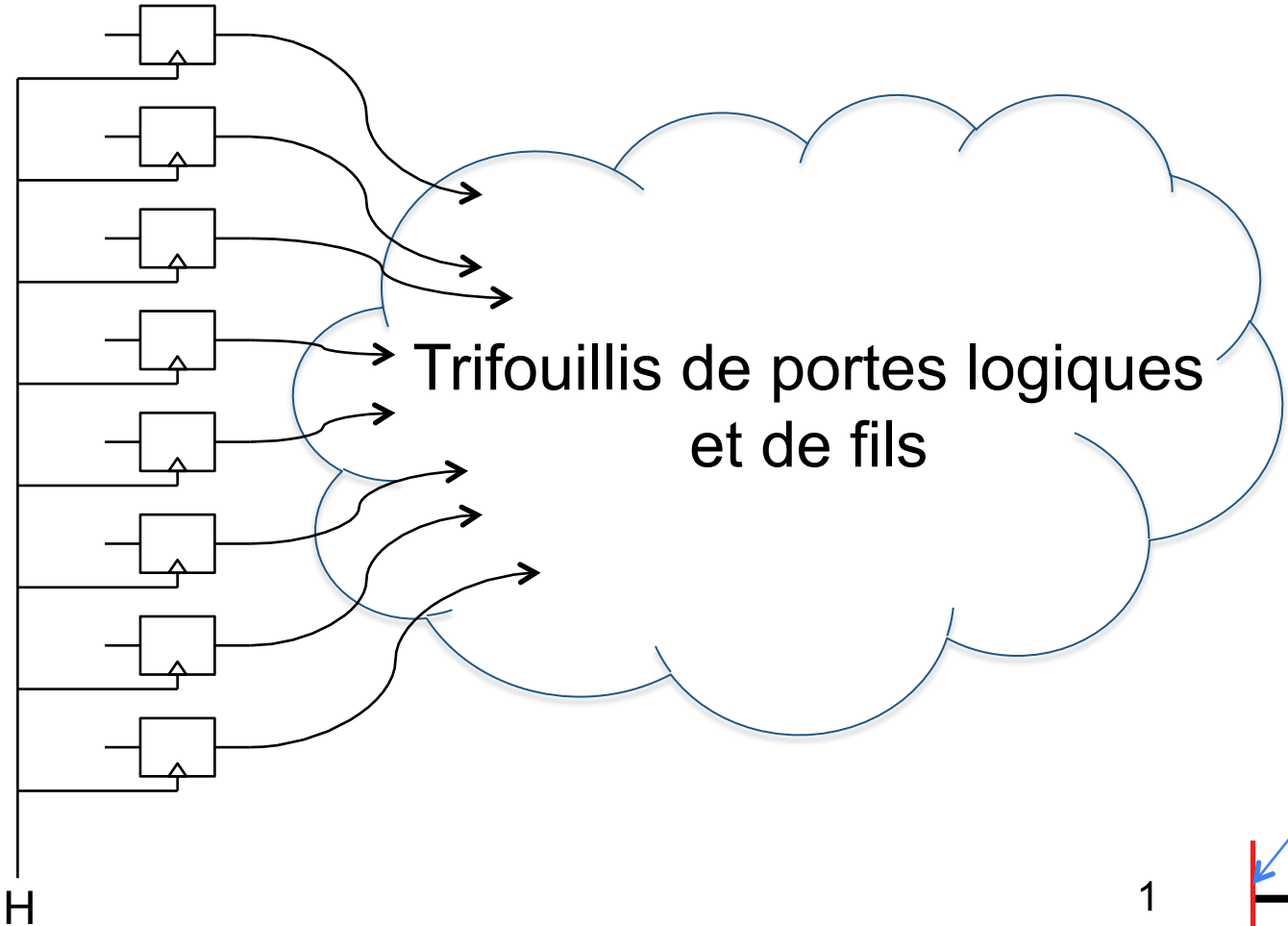
Au niveau électronique (suite)

Notion de cycle



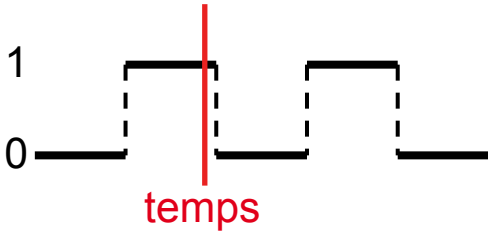
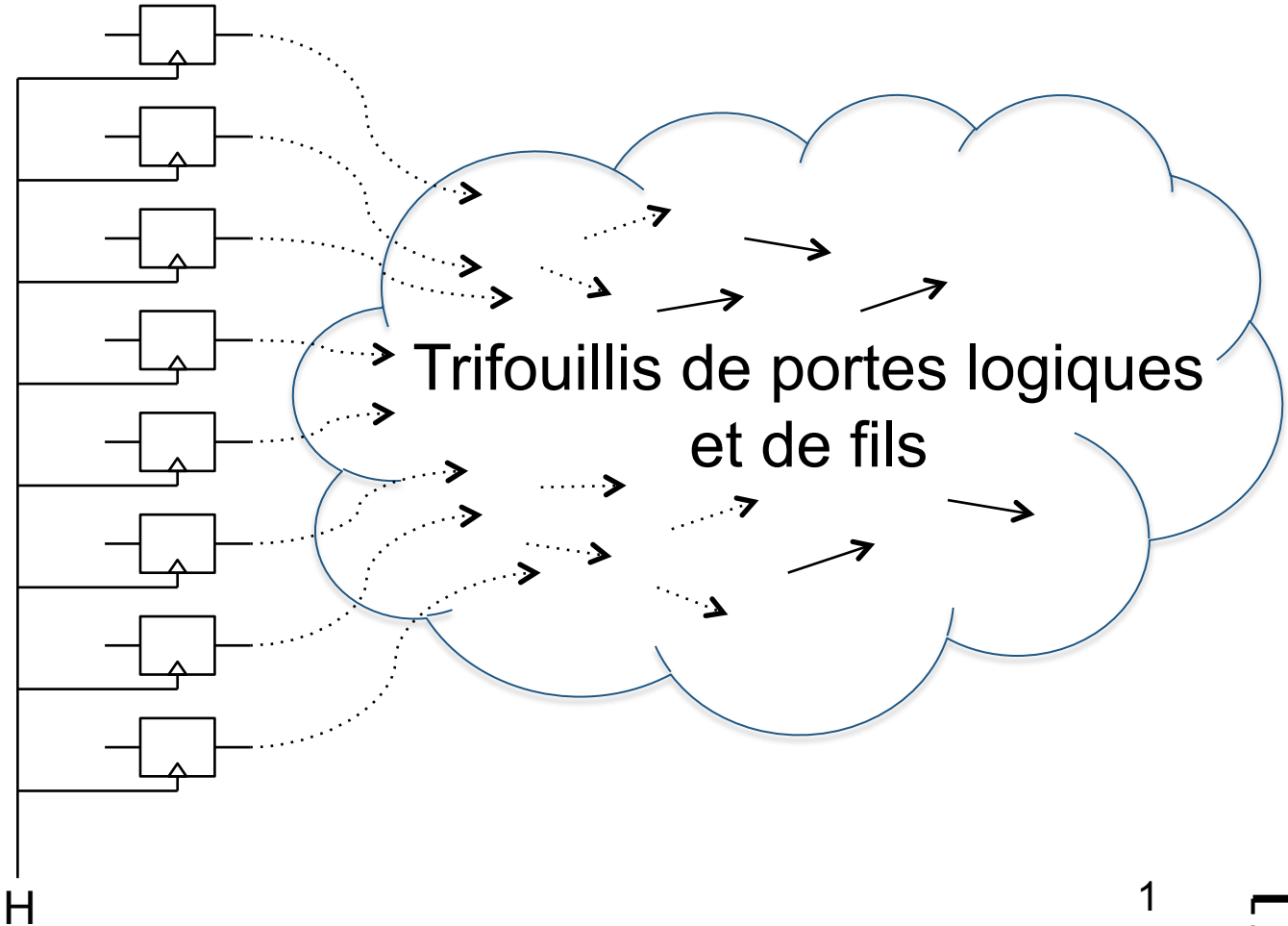
Au niveau électronique (suite)

Notion de cycle



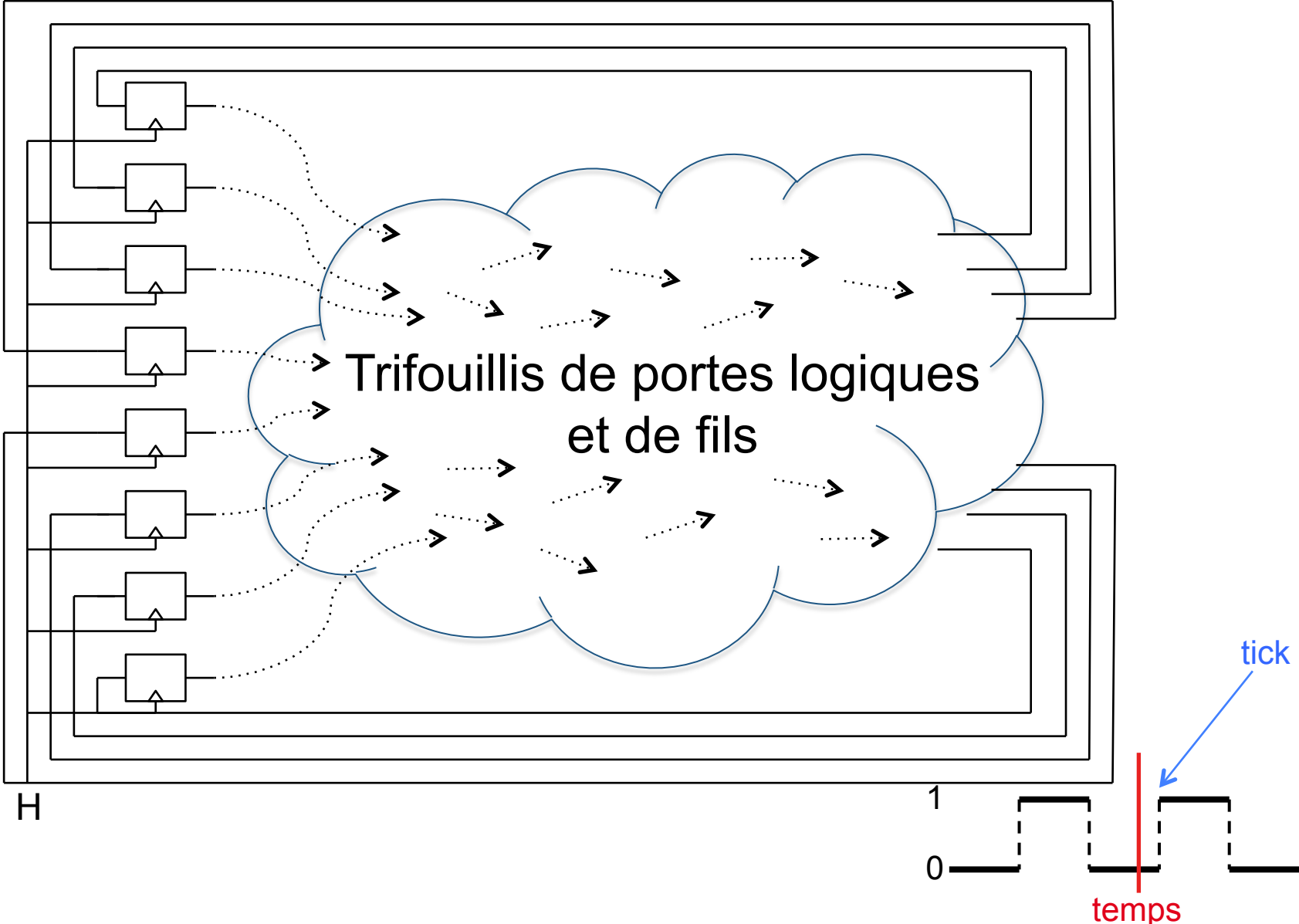
Au niveau électronique (suite)

Notion de cycle



Au niveau électronique (suite)

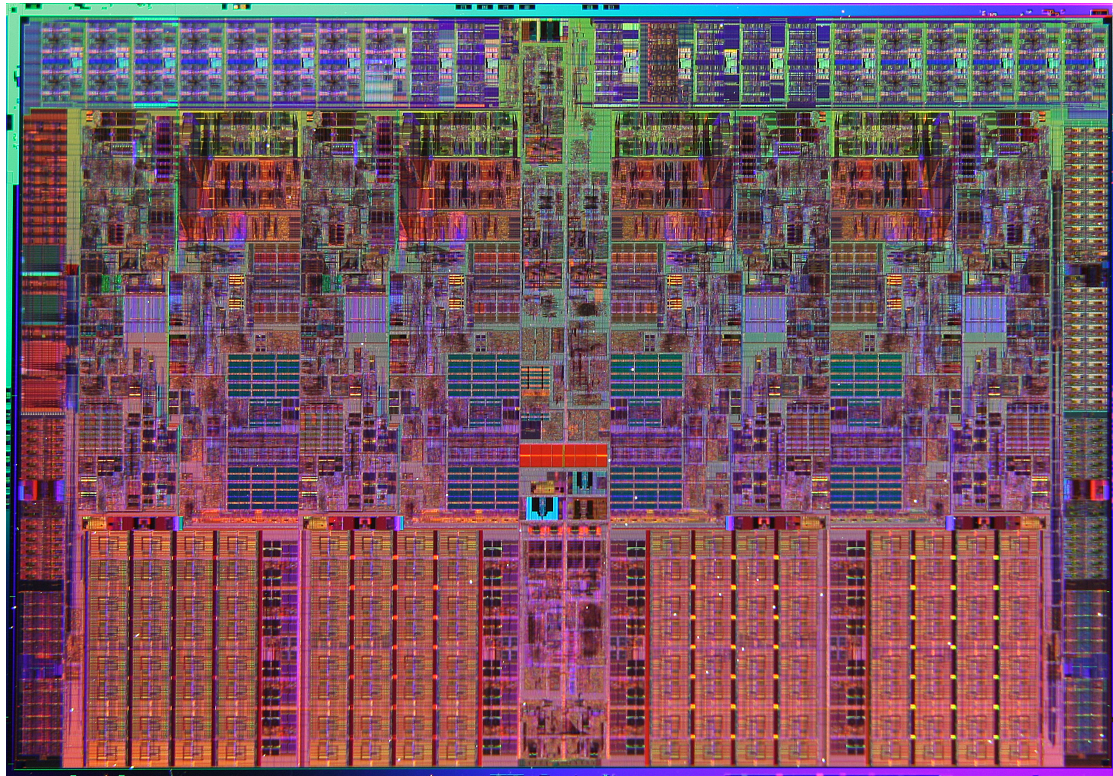
Notion de cycle



Au niveau électronique (suite)

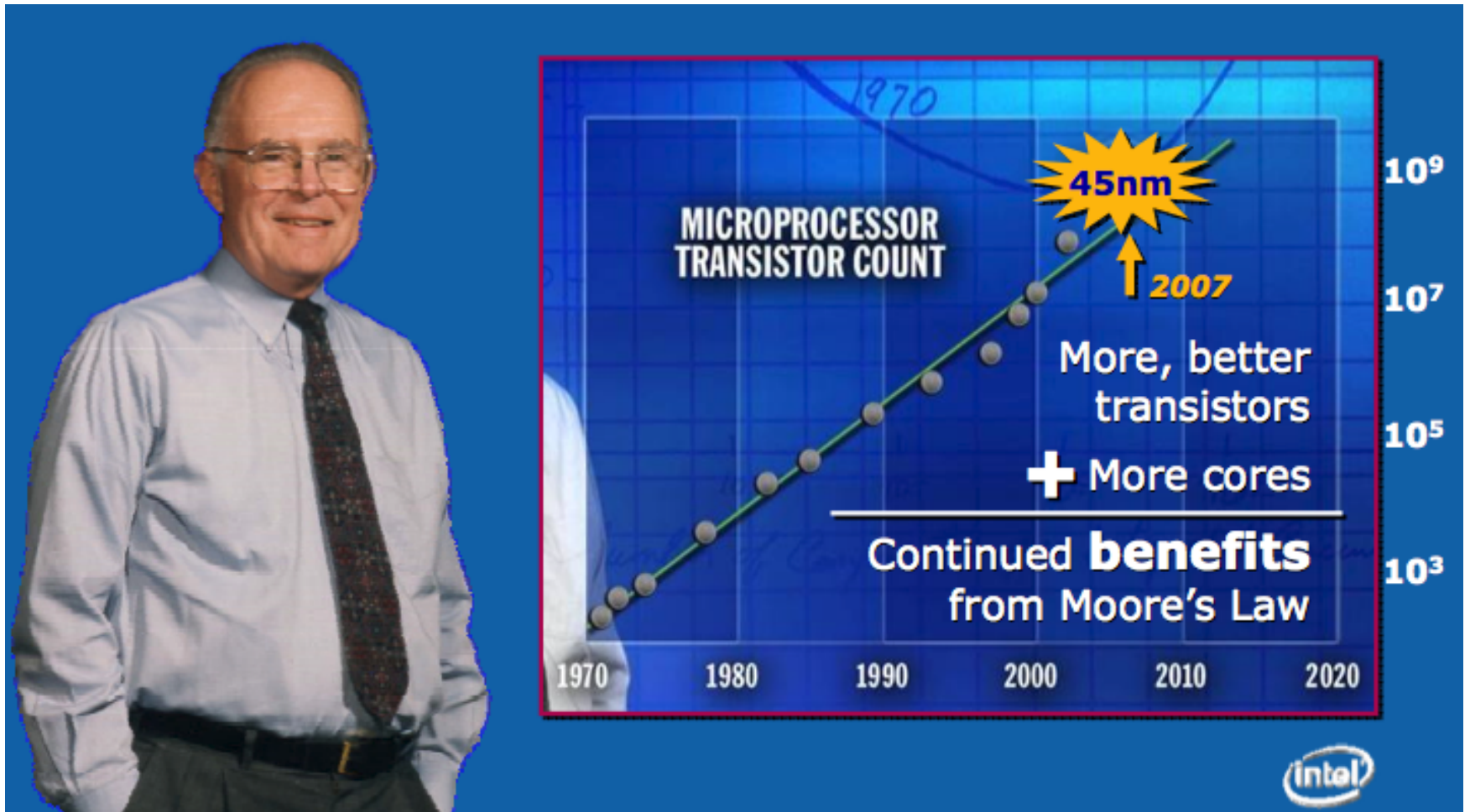
En définitive

- Aujourd'hui, les microprocesseurs sont constitués de plusieurs milliards de transistors
 - Plus personne n'est capable d'examiner leur architecture en détail ;)
 - Ça n'est pas grave : on procède par assemblage de boîtes !



Un prophète

Gordon Moore, co-fondateur d'Intel



« Le nombre de transistors par unité de surface double tous les deux ans »

Loi de Moore

- Si les transistors étaient des personnes

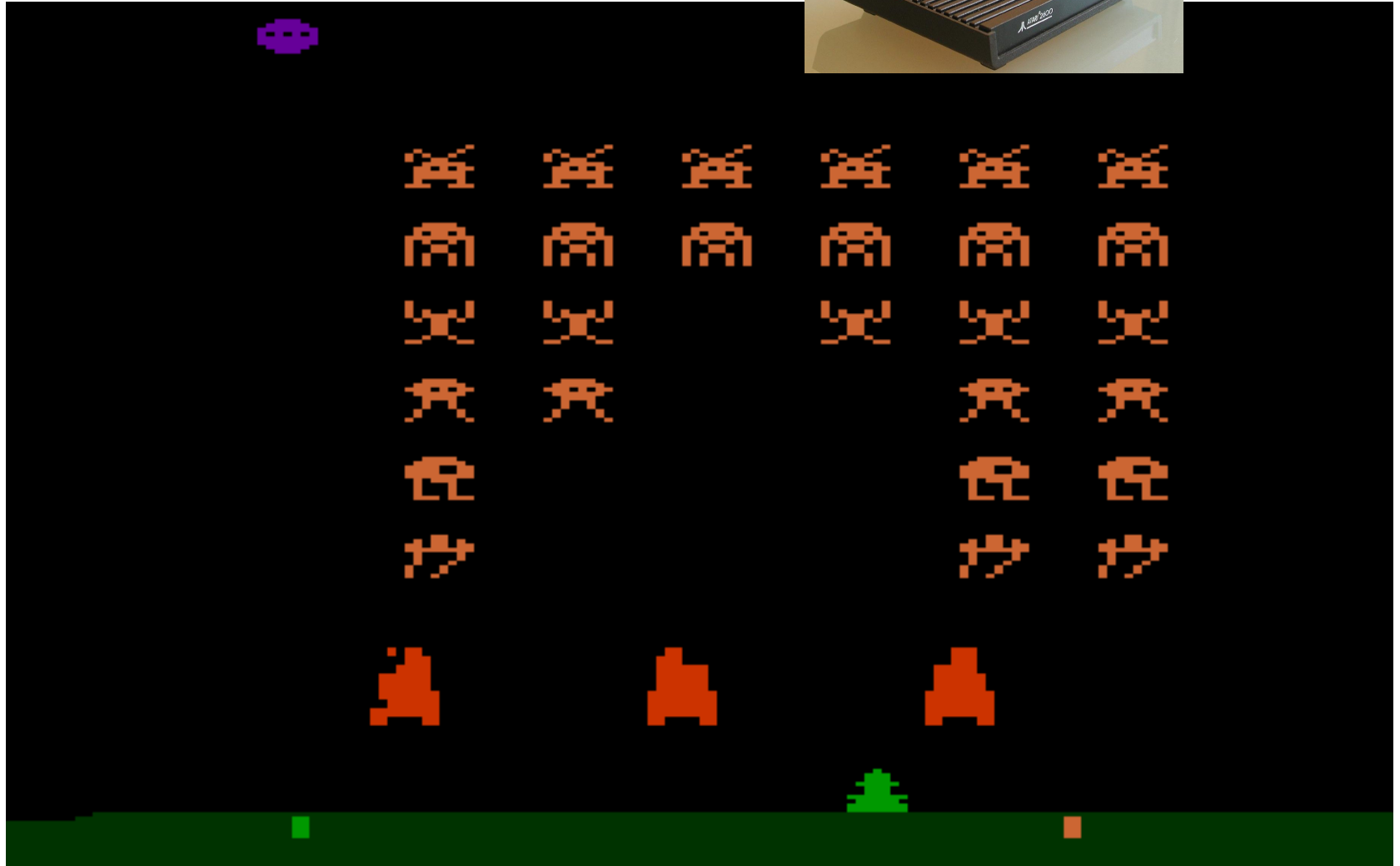


Toujours sur la surface d'un opéra...

1977 : Console Atari 2600



1,19 MHz



2006 : Sony PS3



Processeur Cell 3.2 GHz
Processeur graphique Nvidia RSX



2006 : Sony PS3



Processeur Cell 3.2 GHz
Processeur graphique Nvidia RSX

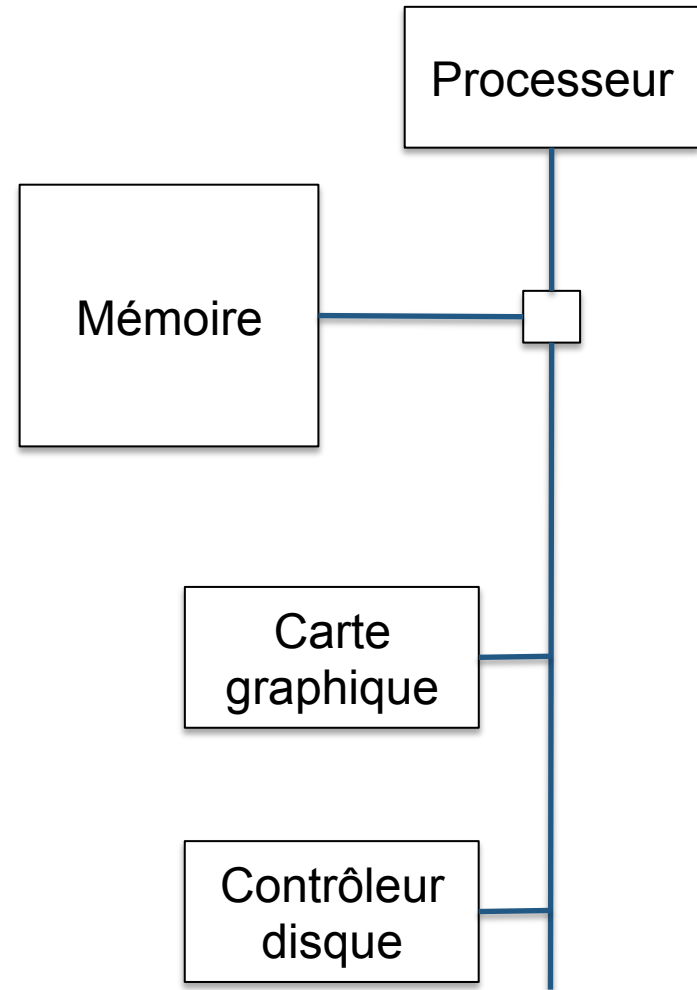
La PS3 calcule
des millions de fois
plus vite !



Architecture des ordinateurs

- Processeur
 - Unité arithmétique & logique
 - Registres
- Mémoire
- Périphériques d'Entrée-Sortie
 - Clavier, Souris
 - Contrôleur disque
 - Carte graphique
 - Carte réseau

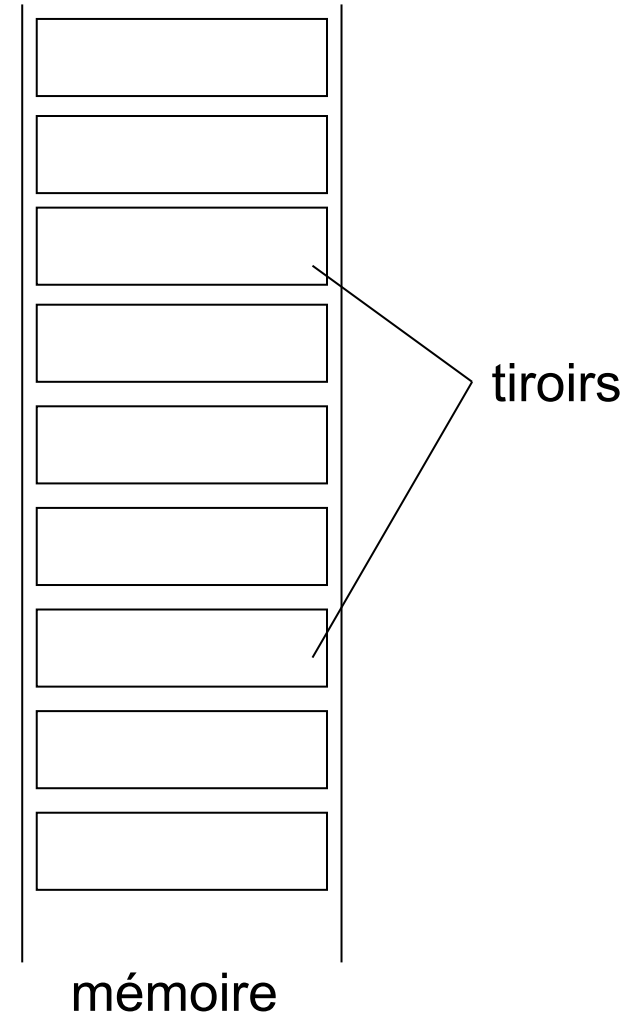
Vue d'ensemble



Principe de fonctionnement d'un ordinateur



microprocesseur



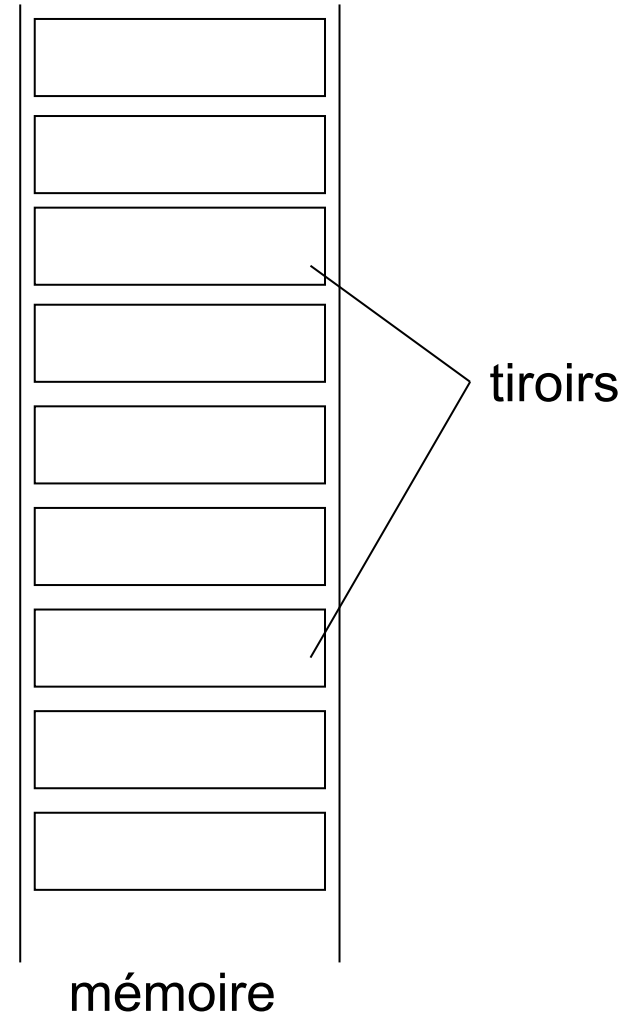
Principe de fonctionnement d'un ordinateur



microprocesseur

Il sait :

- aller chercher un nombre depuis la mémoire
- faire des opérations simples avec le contenu de ses *registres*
- ranger des nombres en mémoire

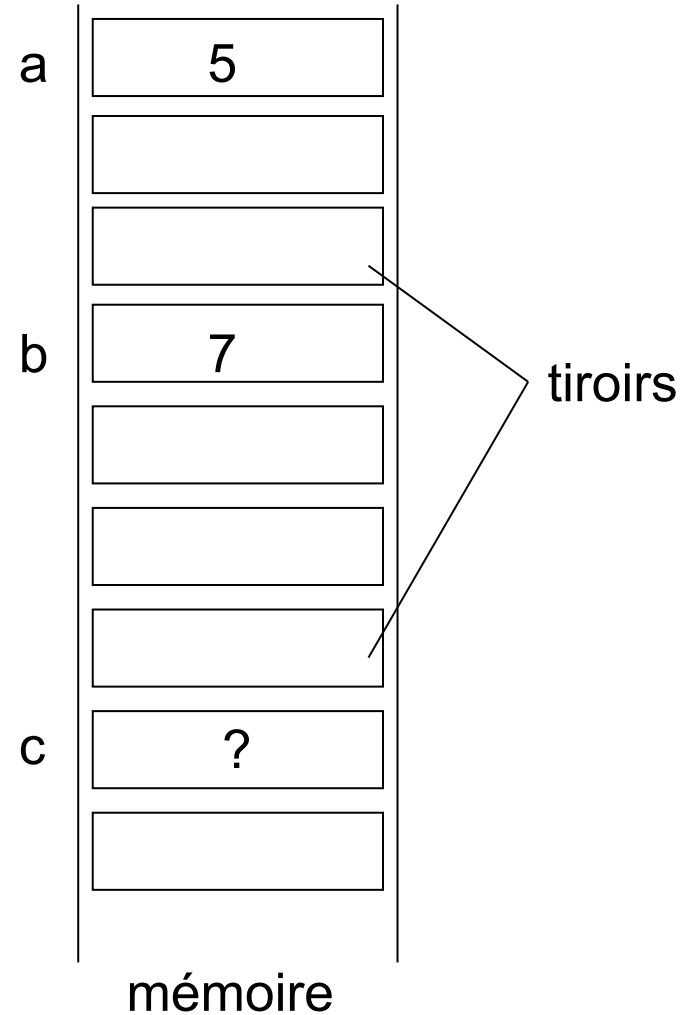


Principe de fonctionnement d'un ordinateur



microprocesseur

Pour calculer l'opération $c = 2 \times a + b$, il faut :



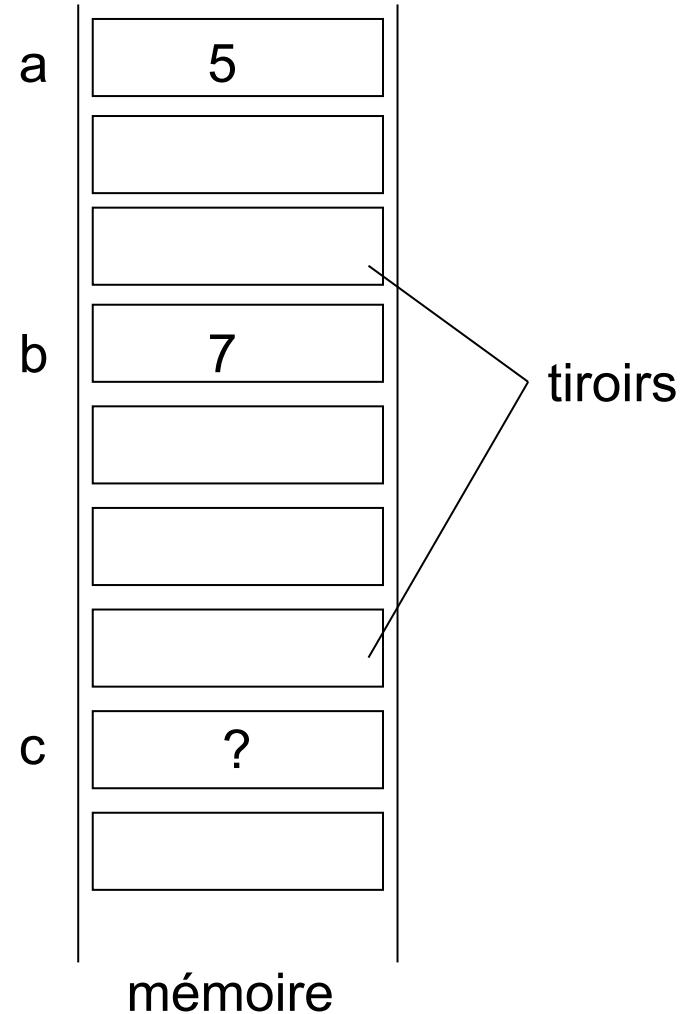
Principe de fonctionnement d'un ordinateur



microprocesseur

Pour calculer l'opération $c = 2 \times a + b$, il faut :

- charger a dans le registre r1
- calculer $r1 = r1 * 2$
- charger b dans le registre r2
- calculer $r1 = r1 + r2$
- ranger r1 dans c



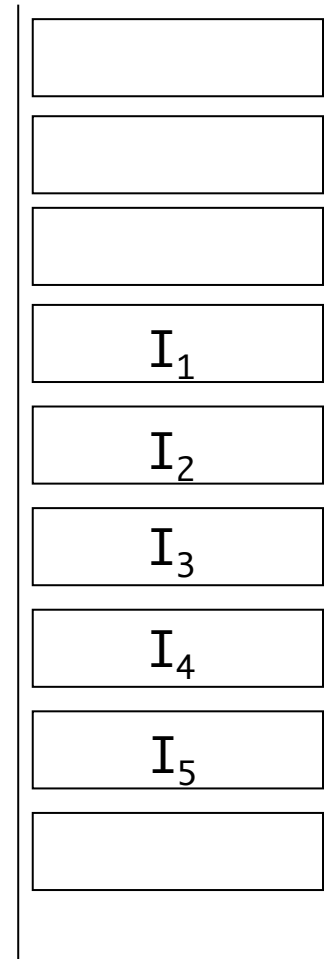
Principe de fonctionnement d'un ordinateur



microprocesseur

Il s'agit d'un programme :

- charger a dans le registre r1 **Instruction 1**
- calculer $r1 = r1 * 2$ **Instruction 2**
- charger b dans le registre r2 **Instruction 3**
- calculer $r1 = r1 + r2$ **Instruction 4**
- ranger r1 dans c **Instruction 5**

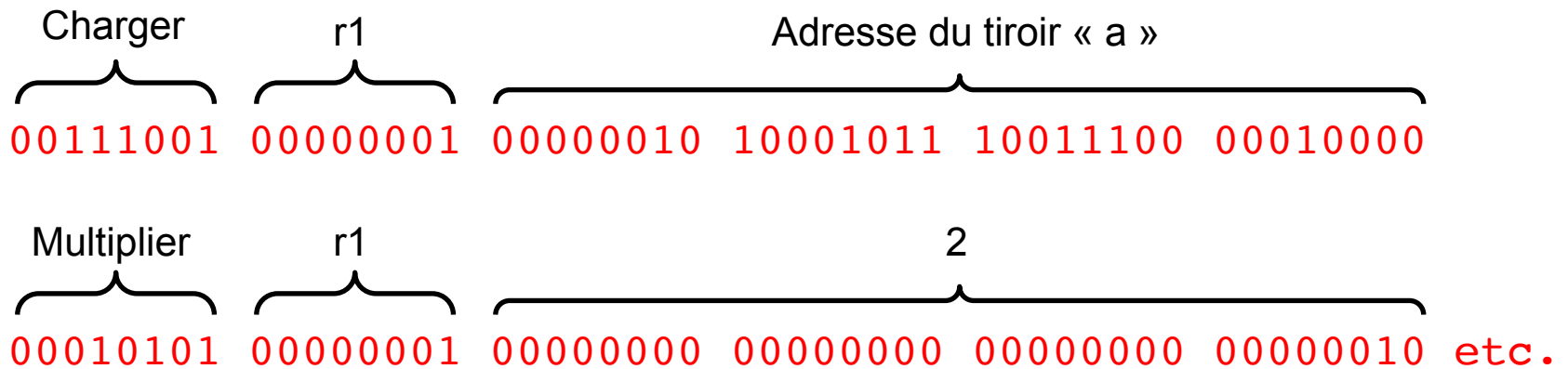


mémoire

Principe de fonctionnement d'un ordinateur

Notion de programme

- Vu que les instructions se trouvent en mémoire — qui ne contient que des nombres — chaque instruction est codée par un ou plusieurs nombres
 - Un programme est donc une suite de nombres binaires



- Cela peut rapidement devenir rébarbatif pour un être humain
 - D'où l'invention de langages de programmation de « haut niveau »

Architecture des ordinateurs

Modèle de Von Neumann

- Le microprocesseur exécute des instructions qu'il charge depuis la mémoire
 - Instructions de transfert
 - load/store
 - Instructions de calcul
 - Add/sub/..., and/or, ...
 - Instruction de branchement
 - Notion de bloc de code conditionnel
 - Boucles et traitements itératifs

```
move 9, r1
load @x, r2
move r2, r3
start:
mult r2, r3, r3
decr r1
jump_if_not_zero start
store r3, @y
halt
```

$$y = x^{10}$$