Un ordinateur, comment ça marche ?

Arnaud Tisserand

CNRS, Lab-STICC

Octobre 2018

Plan de l'exposé

- Introduction
- Processeurs
- Langages de programmation
- Systèmes d'exploitation
- Conclusion

Introduction

Ordinateur : définition du dictionnaire

Petit Larousse 2015 :

> N. m.: Machine automatique de traitement de l'information, obéissant à des programmes formés par des suites d'opérations arithmétiques et logiques.

Wikipédia (3 oct. 2018) :

> Système de traitement de l'information programmable tel que défini par Turing et qui fonctionne par la lecture séquentielle d'un ensemble d'instructions, organisées en programmes, qui lui font exécuter des opérations logiques et arithmétiques.

Précurseurs mécaniques

Une des premières calculatrices mécaniques :

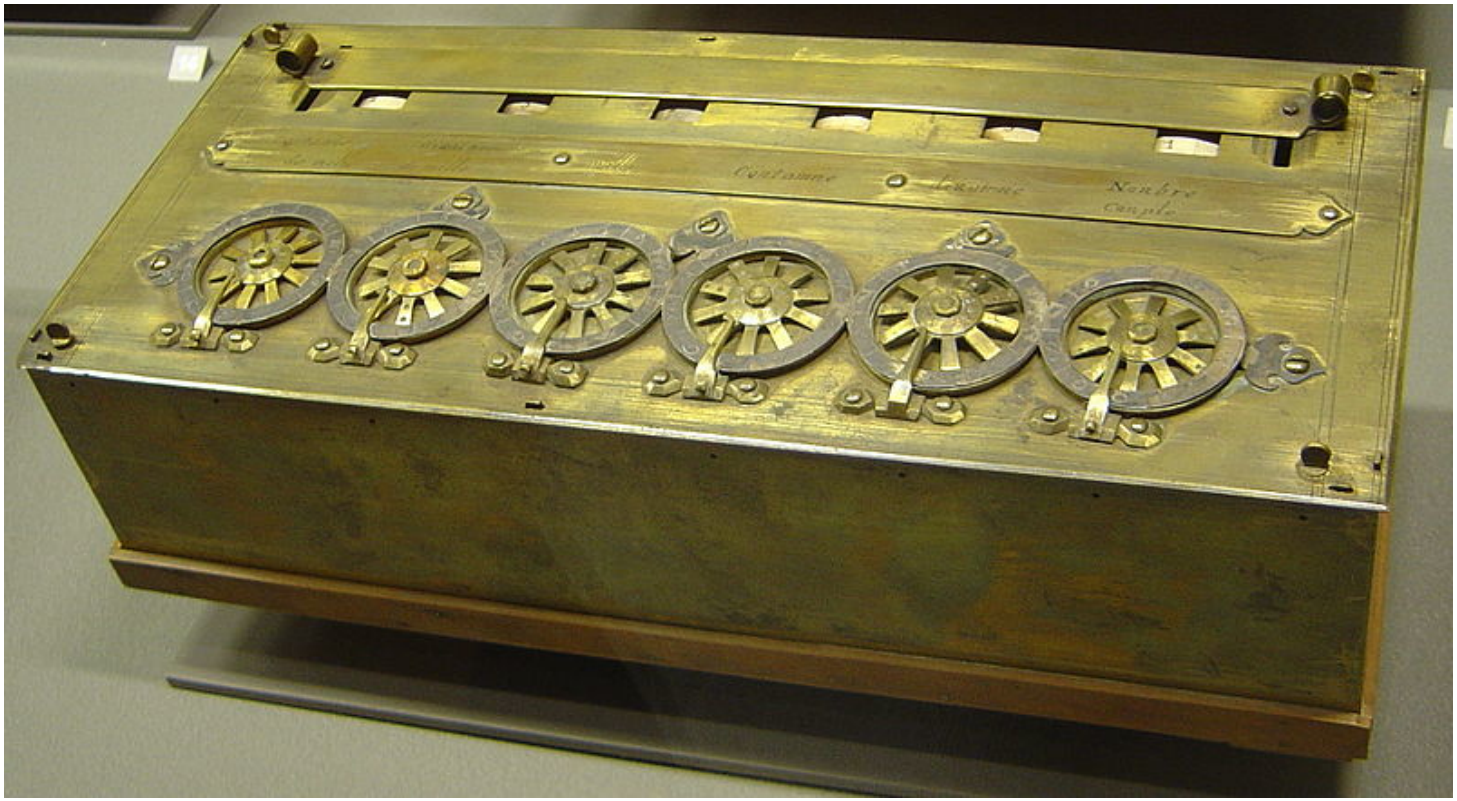

FIGURE 1 – Pascaline imaginée en 1642 par Blaise Pascal (1623–1662) (image : Wikimedia)

Opérations : $+$ et $-$ directement ($\times$ et $\div$ par itérations)

Précurseurs mécaniques (suite)

Un des premiers calculateurs mécaniques programmables :

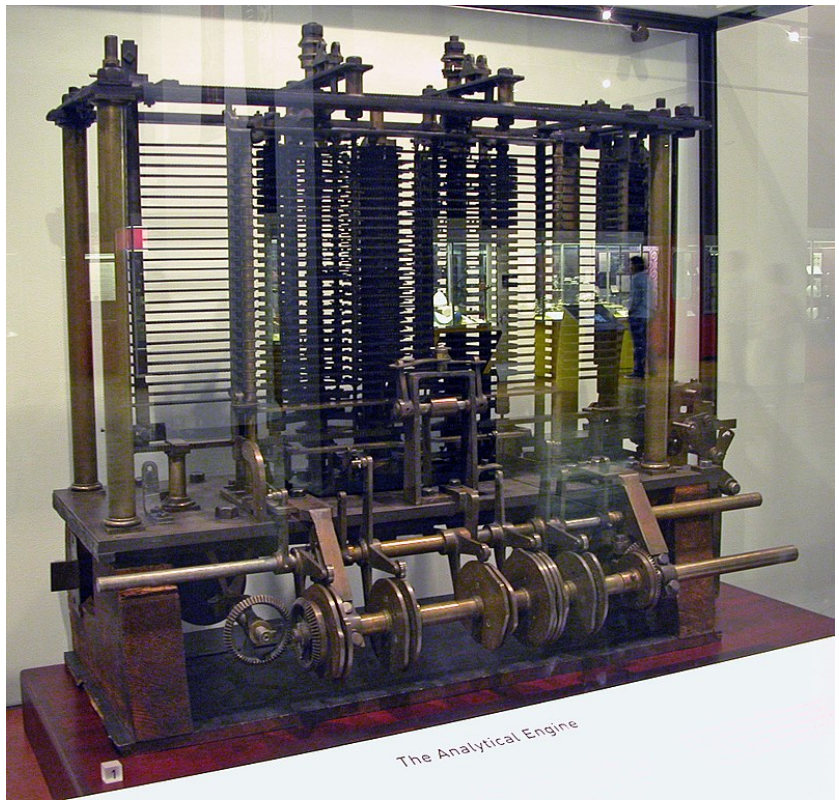

FIGURE 2 – Prototype partiel de la machine analytique imaginée en 1834 par Charles Babbage (1791–1871) (image : Wikimedia)

Machine à faire des calculs utilisant des cartes perforées inspirées des métiers à tisser Jacquard pour stocker les instructions à exécuter.

Précurseurs électromécaniques

Premier processeur en notation scientifique (représentation flottante) :

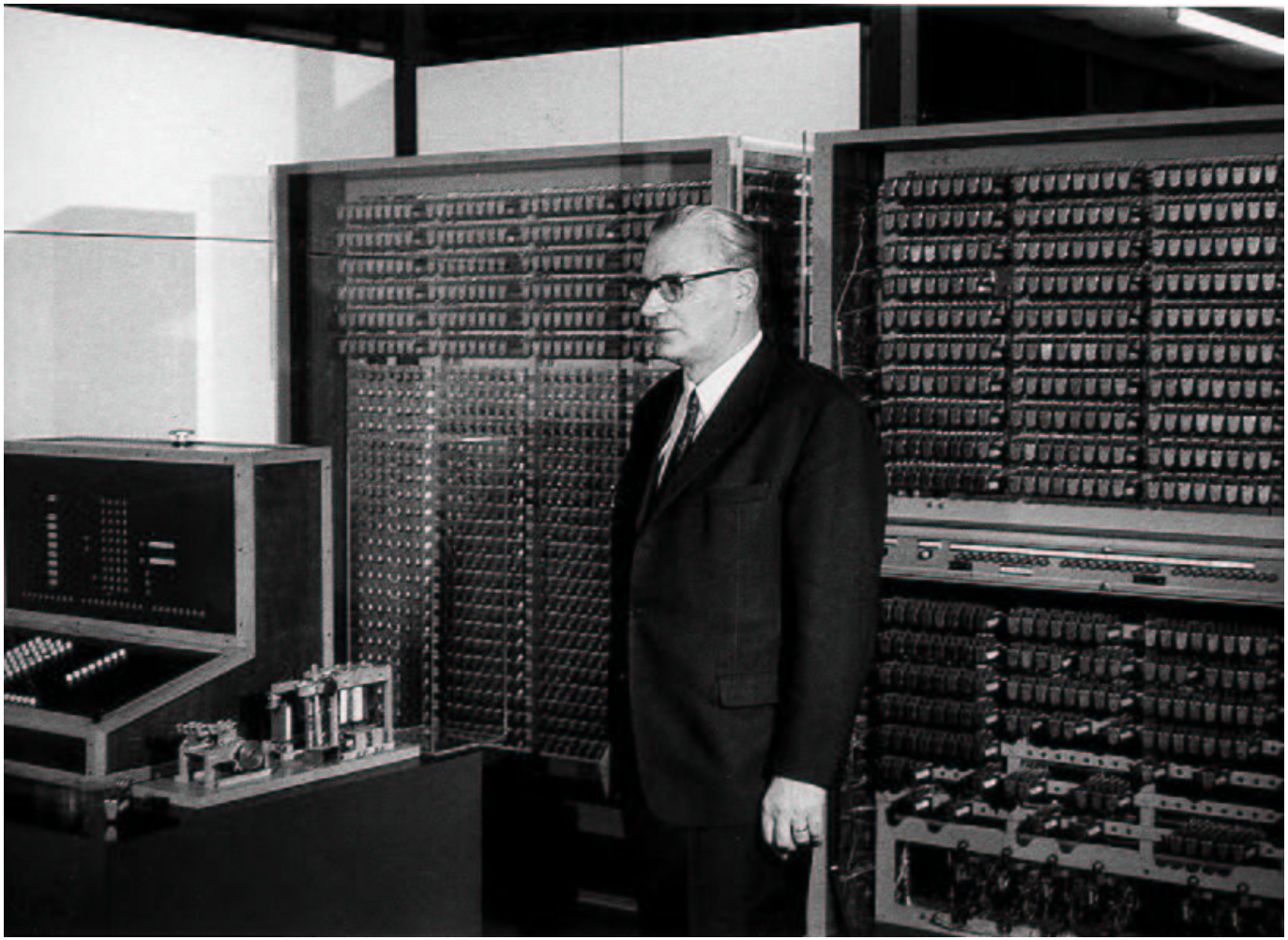

FIGURE 3 – Z3 créé en 1941 par Konrad Zuse (1910–1995) à Berlin (photographie de la version reconstruite en 1961, source http://www.epemag.com/zuse/)

Architecture et caractéristiques du Z3

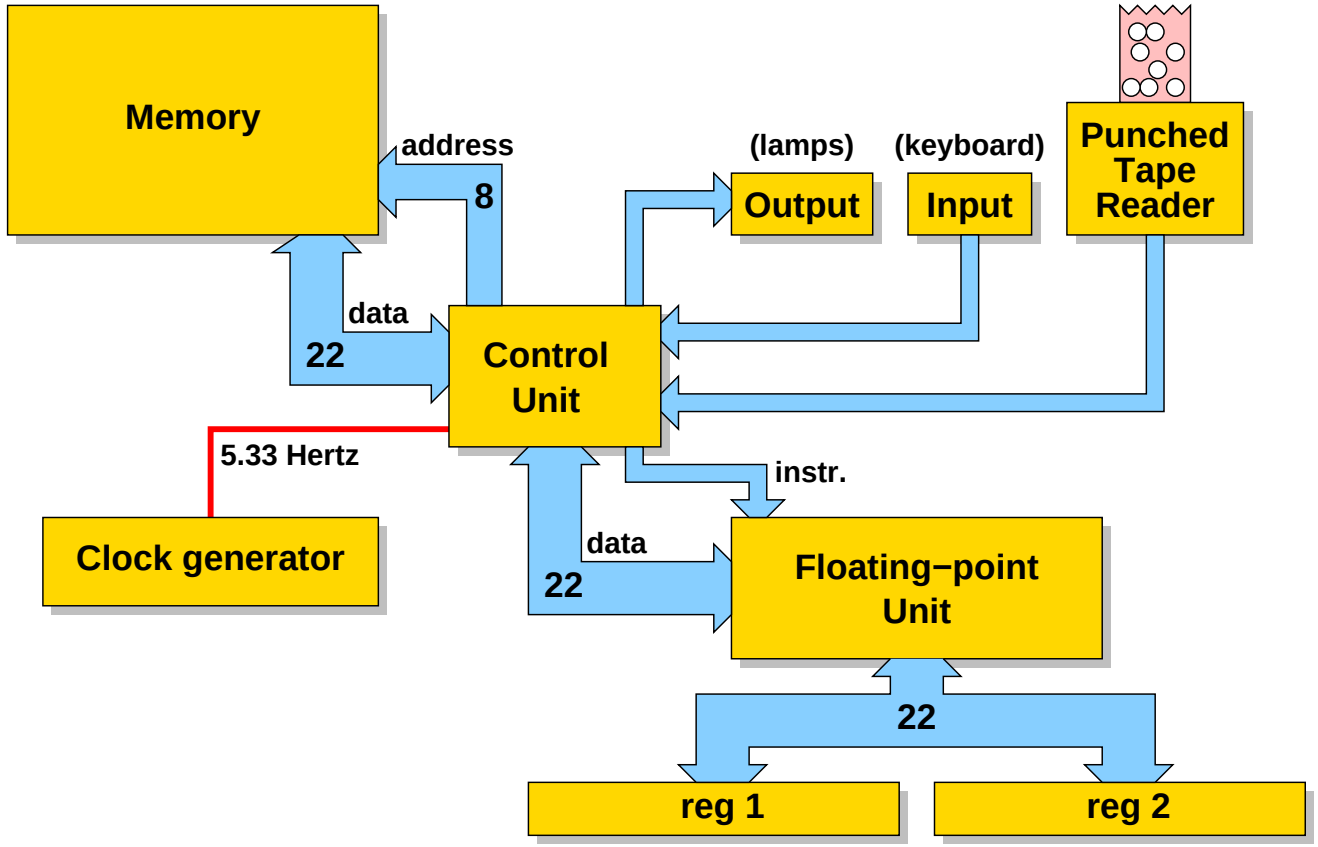

FIGURE 4 – Z3

- Taille et poids : $5 \times 2 \times 0.8$ m, ≈ 1000 kg
- Fréquence : 5.33 Hz (opérations par seconde)
- Technologie : électrique à relais (num. : 600, mém. : 1400)
- Consommation électrique : ≈ 4000 W

Précurseurs électroniques

Electronic Numerical Integrator And Computer (Univ. Pennsylvanie) :

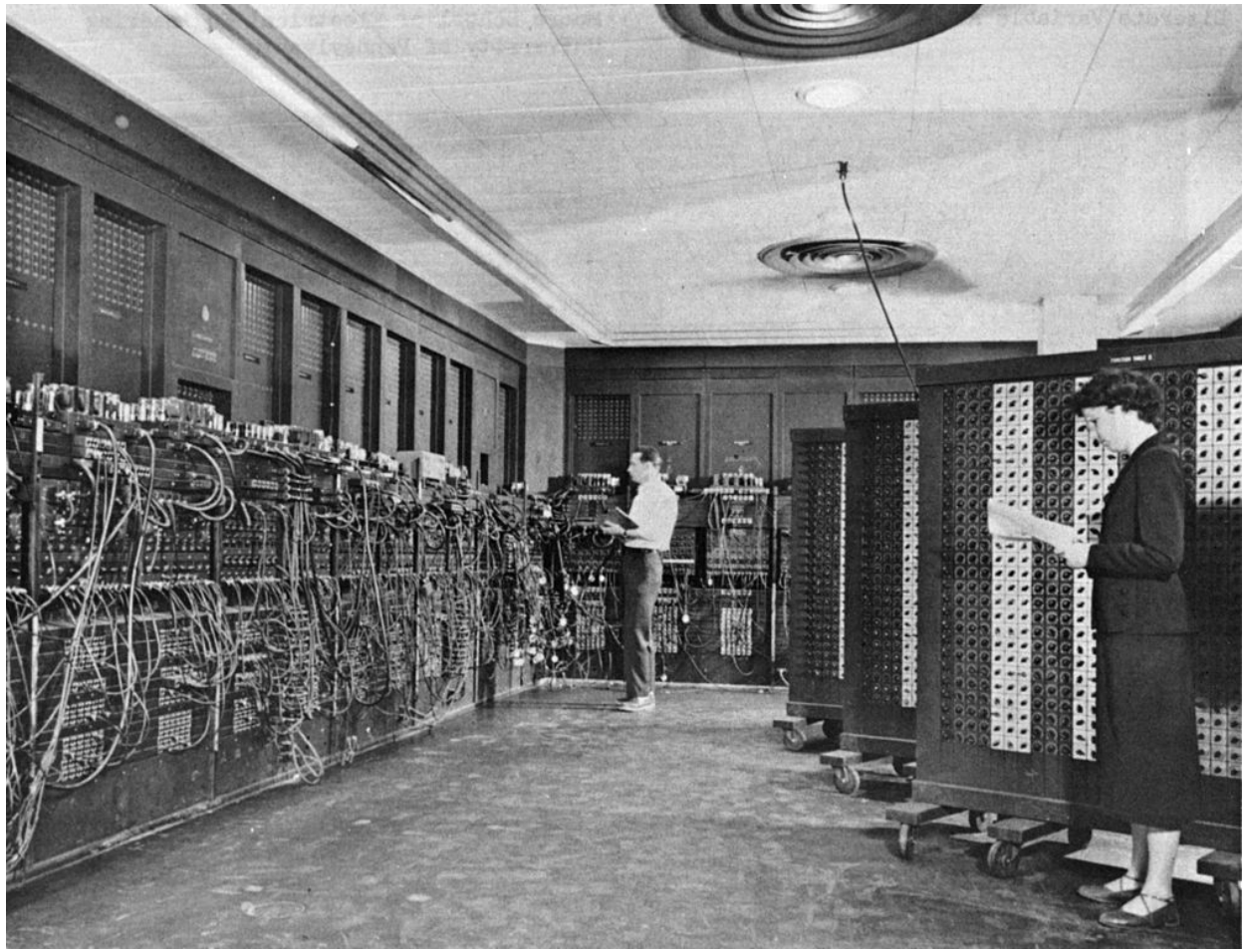

FIGURE 5 – ENIAC conçu en 1943 (source image Wikimedia)

Caractéristiques : 18000 tubes à vide, 5000 op/s, 150 kW

Premier microprocesseur : Intel 4004 en 1971

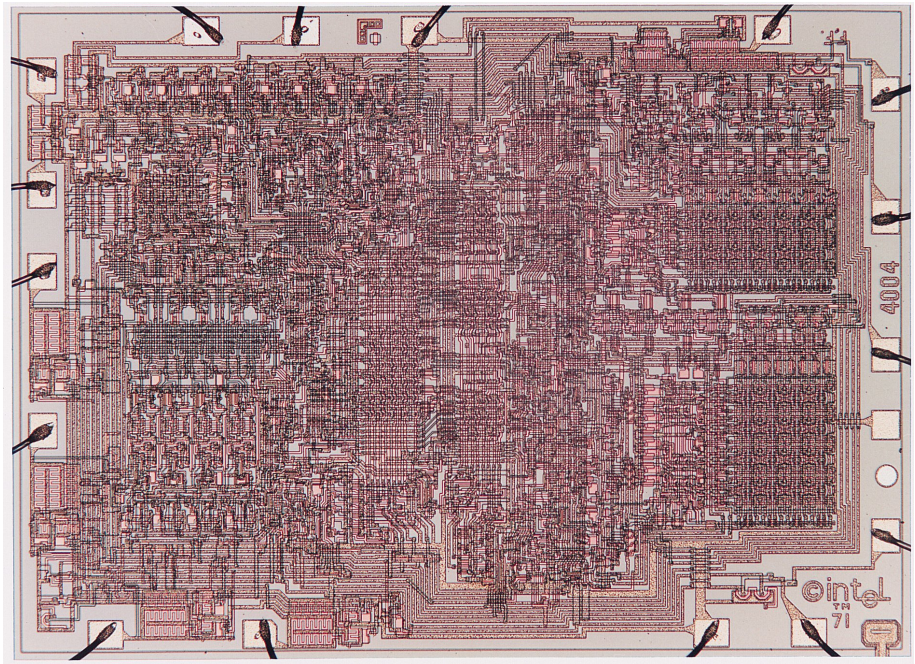

FIGURE 6 – Vue interne du circuit (source : Intel)

- Premier microprocesseur commercialisé en seul circuit intégré (200 $)
- Données de 4 bits, 46 instructions de 8 bits (programme $\leq$ 4096 octets), mémoire 640 octets
- 2300 transistors sur 3.81 $\times$ 2.79 mm, technologie 10 μm ($\sim$ ENIAC)
- Horloge à 740 kHZ, 8 cycles par instruction (92 500 inst./s)
- Boîtier DIP 16 broches céramique, consommation 1 W sous 15 V

Codage de l'information en binaire

Le plus petit élément d'information est le bit (de l'anglais *binary digit* pour chiffre binaire) qui peut être 0 ou 1.

Chaque donnée est composée d'un mot de plusieurs bits. Par exemple :

- un octet = 8 bits
- un mot de 32 bits
- un mot de 64 bits

Représentation des nombres entiers (0, 1, 2, 3, ...) :

- en décimal (base 10) : d_0 est le chiffre des unités, d_1 celui des dizaines, d_2 celui des centaines et d_3 celui des milliers

$$d_3 \times 1000 + d_2 \times 100 + d_1 \times 10 + d_0 \times 1$$

- en binaire (base 2) : les chiffres binaires $b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, \ldots$ sont associés aux valeurs 1, 2, 4, 8, 16, 32, 64, 128, ...

$$b_7 \times 128 + b_6 \times 64 + b_5 \times 32 + b_4 \times 16 + b_3 \times 8 + b_2 \times 4 + b_1 \times 2 + b_0 \times 1$$

Quelques entiers en décimal et en binaire

décimal	binaire
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

décimal	binaire
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111

décimal	binaire
254	11111110
255	11111111
2018	11111100010
65535	1111111111111111

Les calculs binaires sont très anciens. Exemple : article de 1703 par G. Leibniz (1646–1716) : “Explication de l’arithmétique binaire”

Exemples de représentation des caractères en binaire

ASCII : *American Standard Code for Information Interchange*

A	65	1000001	a	97	1100001	0	48	110000
B	66	1000010	b	98	1100010	1	49	110001
C	67	1000011	c	99	1100011	2	50	110010
D	68	1000100	d	100	1100100	3	51	110011
E	69	1000101	e	101	1100101	4	52	110100
F	70	1000110	f	102	1100110	8	56	111000
G	71	1000111	g	103	1100111	9	57	111001
H	72	1001000	h	104	1101000		32	100000
I	73	1001001	i	105	1101001	@	64	1000000
J	74	1001010	j	106	1101010	-	45	101101
X	88	1011000	x	120	1111000	.	46	101110
Y	89	1011001	y	121	1111001	,	44	101100
Z	90	1011010	z	122	1111010	;	59	111011

bonjour $\rightsquigarrow$ 110001011011111101110110101011011111101011110010

Processeurs

Processeur : définition du dictionnaire

Petit Larousse 2015 :

> N. m. : Organe d'un ordinateur qui assure l'interprétation et l'exécution des instructions.

Wikipédia (29 septembre 2018) :

> Composant présent dans de nombreux dispositifs électroniques qui exécute les instructions machine des programmes informatiques.

Composants dans un processeur

- Transistors
- Cellules logiques
- Unités de calcul
- Aiguillages
- Horloge
- Registres
- Mémoires

Transistors

Interrupteur élémentaire piloté par un signal de commande (C) qui connecte ou déconnecte les points A et B.

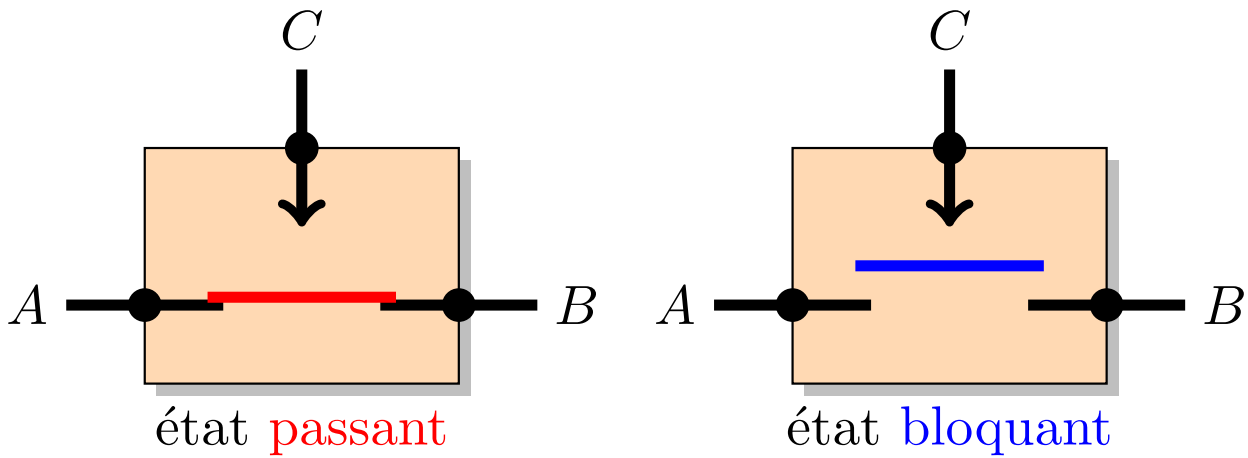

FIGURE 7 – Schéma de principe d'un transistor

Finesse de gravure : plus petite taille des éléments réalisables dans le circuit.

1 m	1 mm	1 μm	1 nm
1	0.00**1**	0.000 00**1**	0.000 000 00**1**

Vue interne d'une puce électronique

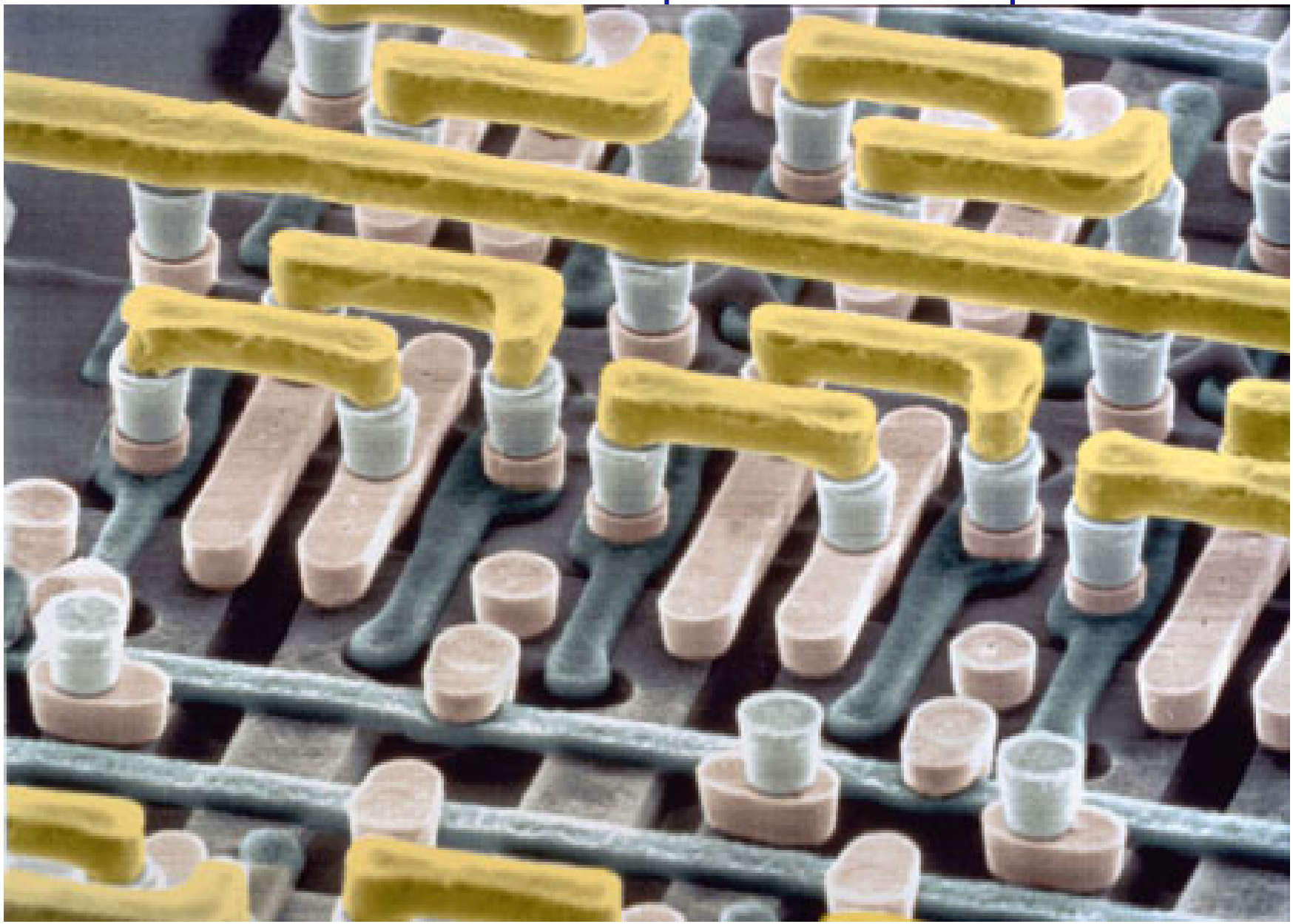

FIGURE 8 – Image prise au microscope électronique

Cellules logiques

Ensemble de transistors qui effectue un calcul simple sur quelques bits.

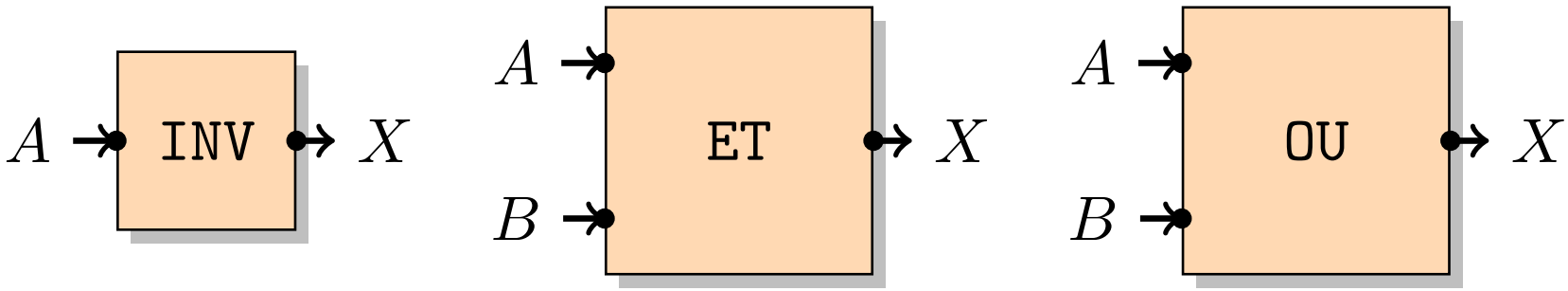

FIGURE 9 – Exemples de cellules logiques

A	B	INV(A)	ET(A, B)	OU(A, B)
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

On utilise des cellules logiques (comme INV, ET, OU), arithmétiques (addition sur 1 bit) et de mémoire (stocker un bit dans le temps).

Unité de calcul

Ensemble de cellules logiques qui effectue des calculs arithmétiques (p. ex. $+$, $-$, $\times$, $\div$) et logiques (inv, et, ou, décalage) sur des mots de plusieurs bits.

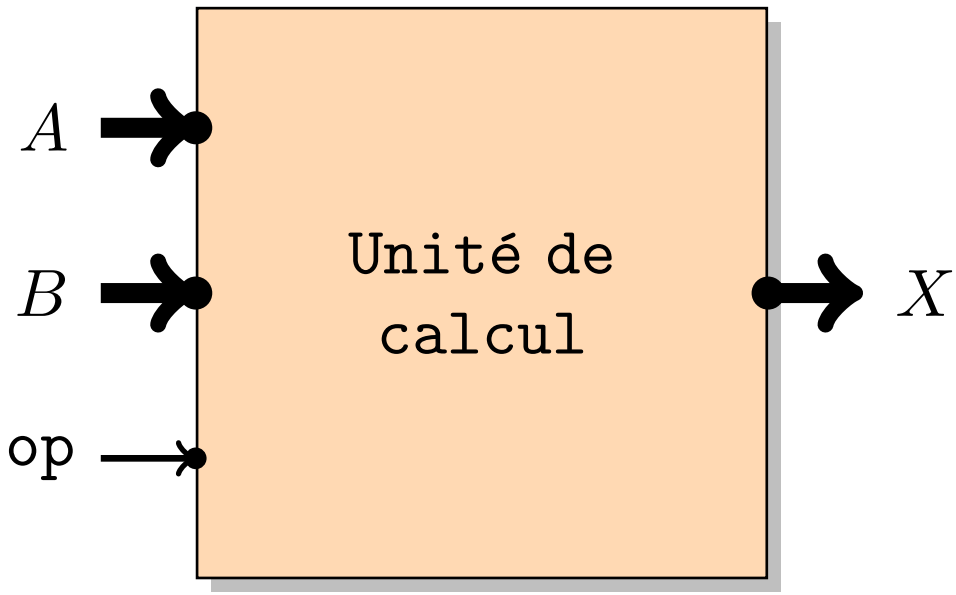

FIGURE 10 – Unité de calcul

La sortie de l'unité est le résultat X :

$$X \leftarrow A \text{ op } B$$

avec op qui peut être une des opérations possibles (p. ex. $+$, $-$, $\times$, $\div$).

Aiguillages

Cellules qui connectent différents points d'un circuit en fonction de la valeur d'un signal de contrôle (C sur un 1 bit).

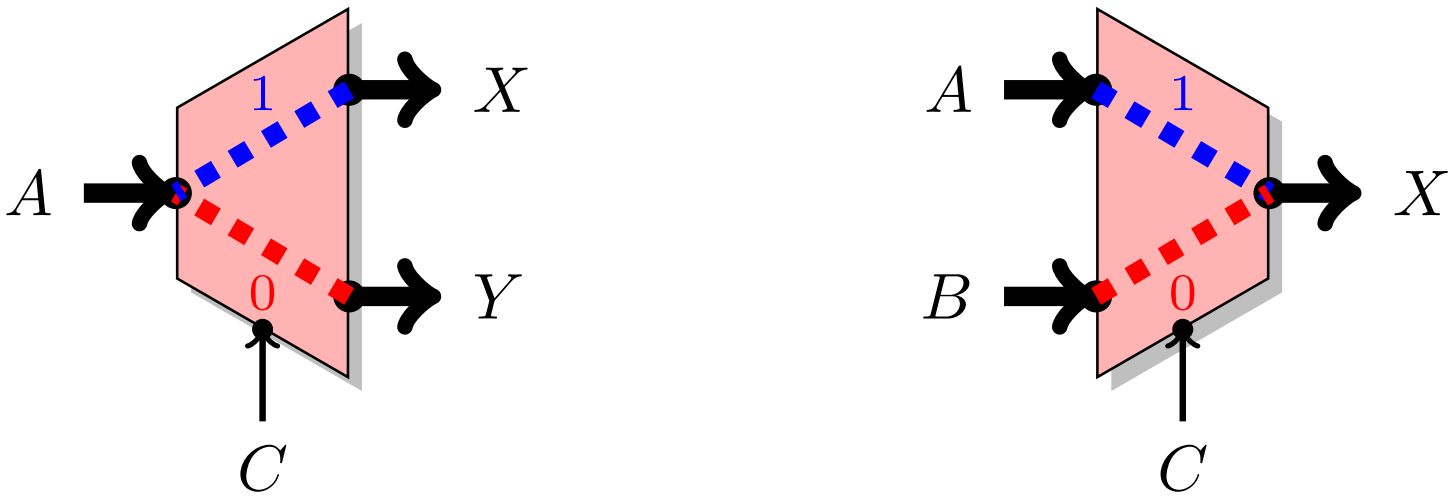

FIGURE 11 – Aiguillages $1 \to 2$ (gauche) et $2 \to 1$ (droite)

Fonctionnement des aiguillages :

- $1 \to 2$ (à gauche) : l'entrée A est connectée à la sortie X lorsque $C = 1$ ou à la sortie Y lorsque $C = 0$
- $2 \to 1$ (à droite) : la sortie X est connectée à l'entrée A lorsque $C = 1$ ou à l'entrée B lorsque $C = 0$

Horloge

Chef d'orchestre qui synchronise les composants de mémorisation pour passer d'une étape de calcul à la suivante. Chaque étape est appelée cycle horloge. Les flèches en bleu foncé indiquent les instants de passage d'un cycle au prochain appelés fronts d'horloge.

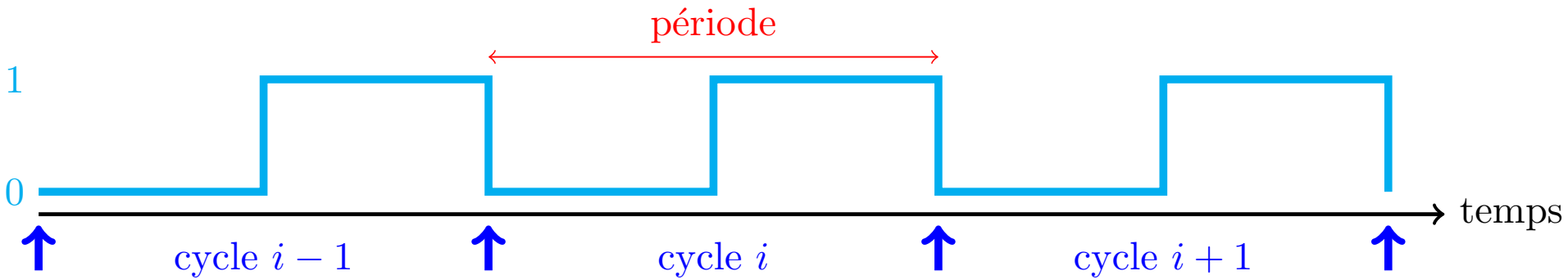

FIGURE 12 – Signal d'horloge

Fréquence de fonctionnement ($f = 1/p$) :

1 Hz	1 kHz	1 MHz	1 GHz
1	**1** 000	**1** 000 000	**1** 000 000 000

Registre

Toute petite mémoire qui stocke un seul mot de n bits pendant un cycle horloge. Au front d'horloge, la valeur sur la sortie X devient la valeur qui était présente sur l'entrée A au cycle précédent.

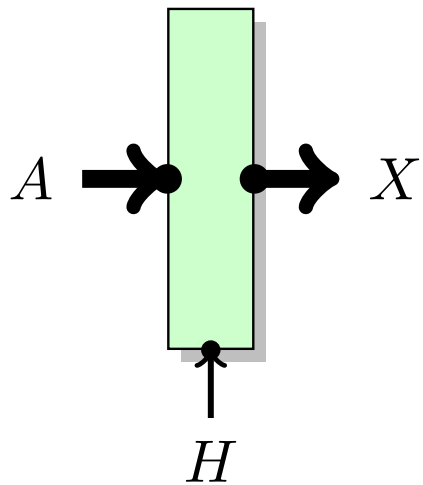

FIGURE 13 – Registre

Mémoire

Stocke un ensemble de mots, à différentes adresses, qui peuvent être lus ou écrits au rythme imposé par le signal d'horloge.

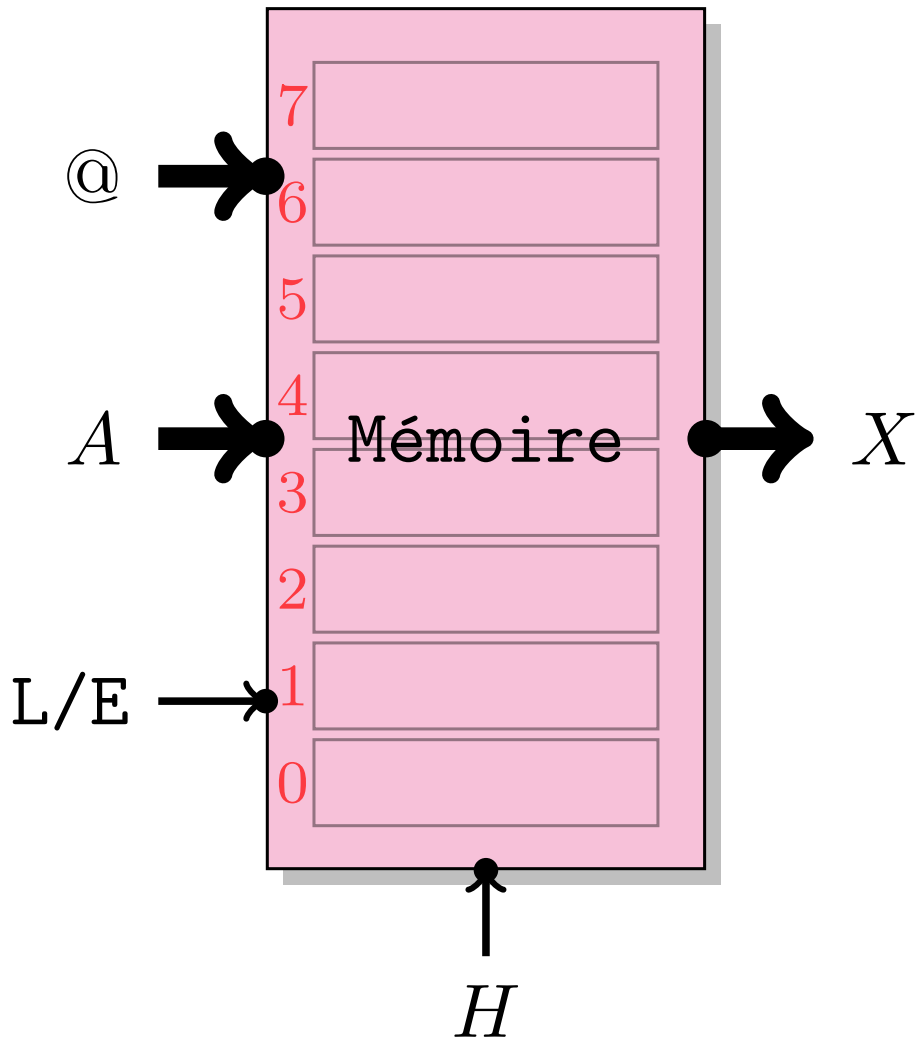

FIGURE 14 – Mémoire

Processeur

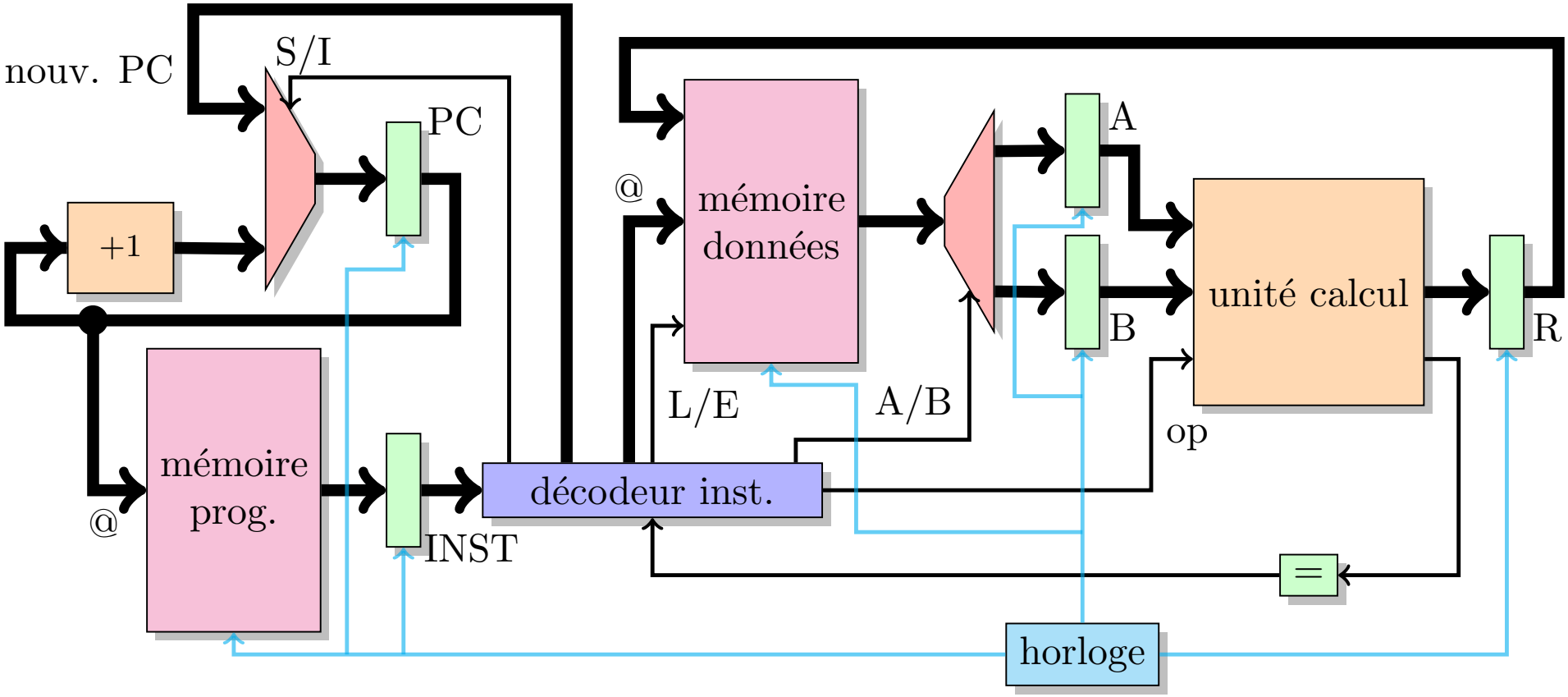

FIGURE 15 – Processeur

- PC : compteur de programme (adresse de l'instruction actuelle)
- S/I : saut ou incrément pour la prochaine instruction
- L/E : lecture ou écriture de donnée
- A/B : sélection du registre temporaire A ou B

Instructions et mémoire de programme

À chaque cycle horloge, une instruction est lue dans la mémoire de programme pour spécifier au processeur ce qu'il doit faire.

Jeu d'instructions du processeur :

instruction	comportement
`LEC @ A/B`	lecture de la donnée à l'adresse @ dans A ou B
`ECR @`	écrit le résultat à l'adresse @
`ADD`	fait l'addition des deux registres A et B
`MUL`	fait la multiplication des deux registres A et B
`SAU @`	saute à l'instruction à l'adresse @
`EG?`	test si les deux registres sont égaux

Codage des instructions

Il y a 6 instructions donc 3 bits sont suffisants pour dire quelle instruction effectuer. Ensuite il faut des bits pour l'adresse @ et un bit pour dire A ou B pour la destination des lectures.

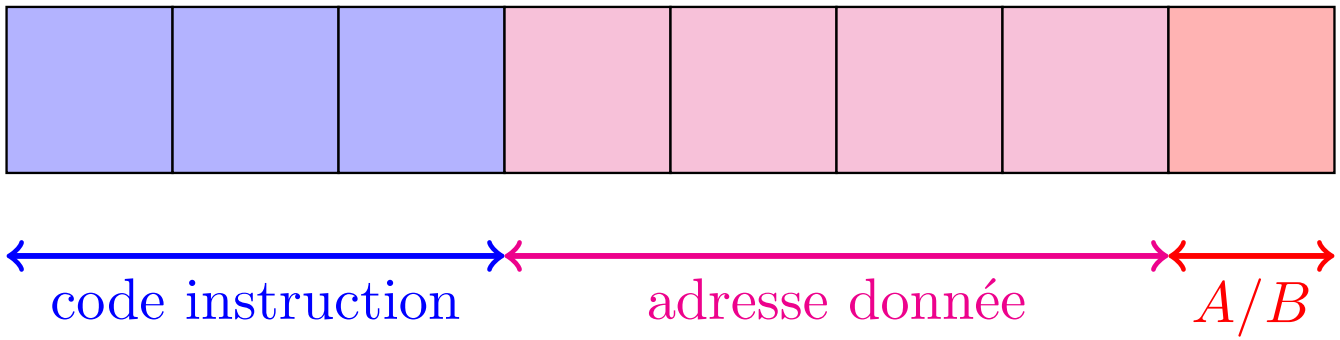

FIGURE 16 – Format des instructions

`LEC` : `000`, `ECR` : `001`, `ADD` : `010`, `MUL` : `011`, `SAU` : `100`, `EG?` : `101`

Exemples :

- 0 0 0 0 1 0 0 0 : LEC @=4 A
- 0 0 0 0 1 0 1 1 : LEC @=5 B
- 0 1 0 0 0 0 0 0 : ADD
- 0 0 1 1 0 0 0 0 : ECR @=8

Exemple de programme pour notre petit processeur

État de la mémoire avant le programme

0	1	2	3	4	5	6	7
0	10	20	0	0	0	0	0

```
0    LEC @=1 A
1    LEC @=2 B
2    ADD
3    ECR @=3
4    MUL
5    ECR @=4
6    LEC @=3 A
7    LEC @=4 B
8    ADD
9    ECR @=0
```

Quel sera l'état de la mémoire après le programme ?

Solution

0	1	2	3	4	5	6	7
0	10	20	0	0	0	0	0

```
0    LEC @=1 A
1    LEC @=2 B
2    ADD
3    ECR @=3
4    MUL
5    ECR @=4
6    LEC @=3 A
7    LEC @=4 B
8    ADD
9    ECR @=0
```

0	1	2	3	4	5	6	7
230	10	20	30	200	0	0	0

Exemples de processeurs du commerce

processeur	MSP 430	Xeon E7-8890 v2
type	micro-contrôleur faible conso.	processeur haut de gamme
fabricant	Texas Instruments	Intel
données	16 bits	64 bits
instructions	1 / cycle	30 / cycle (avec 15 cœurs)
fréquence	4 – 24 MHz	2.8 – 3.4 GHz
tension	1.8 – 3.6 V	0.6 – 1.3 V
puissance	3.5 mW à 10 MHz	155 W
c. sommeil	0.5 – 50 μA	n.c.
prix	1 – 7 €	$\approx$ 5 500 €
technologie	250 nm	22 nm

Langages de programmation

Programme

Petit Larousse 2015 :

> N. m. : Séquence d'instructions et de données enregistrée sur un support susceptible d'être traitée par un ordinateur.

Programme binaire ou langage machine

Séquence d'instructions en format binaire exécutée par le processeur.

Exemple pour notre petit processeur :

0 0 0 0 1 0 0 0

0 0 0 0 1 0 1 1

0 1 0 0 0 0 0 0

0 0 1 1 0 0 0 0

C'est pas vraiment lisible pour un humain !

Assembleur

Séquence d'instructions affichées avec noms, adresses et informations.

```
0    LEC @=1 A
1    LEC @=2 B
2    ADD
3    ECR @=3
4    MUL
5    ECR @=4
6    LEC @=3 A
7    LEC @=4 B
8    ADD
9    ECR @=0
```

C'est un peu mieux, mais cela reste très compliqué.

Et puis l'assembleur dépend totalement du jeu d'instructions et donc du processeur.

Langage de haut niveau

Ensemble de directives pour décrire, de façon indépendante du processeur :

- des créations, manipulations et destructions de structures de données
  - nombres
  - chaînes de caractères
  - conteneurs (listes et tableaux)
  - fichiers
- des algorithmes
  - opérations
  - test de condition
  - répétitions (boucles)
  - créer et utiliser des fonctions

Compilateur

Programme qui traduit automatiquement un programme source, écrit dans un langage de haut niveau, en un programme exécutable, écrit en binaire, pour une exécution dans le processeur directement.

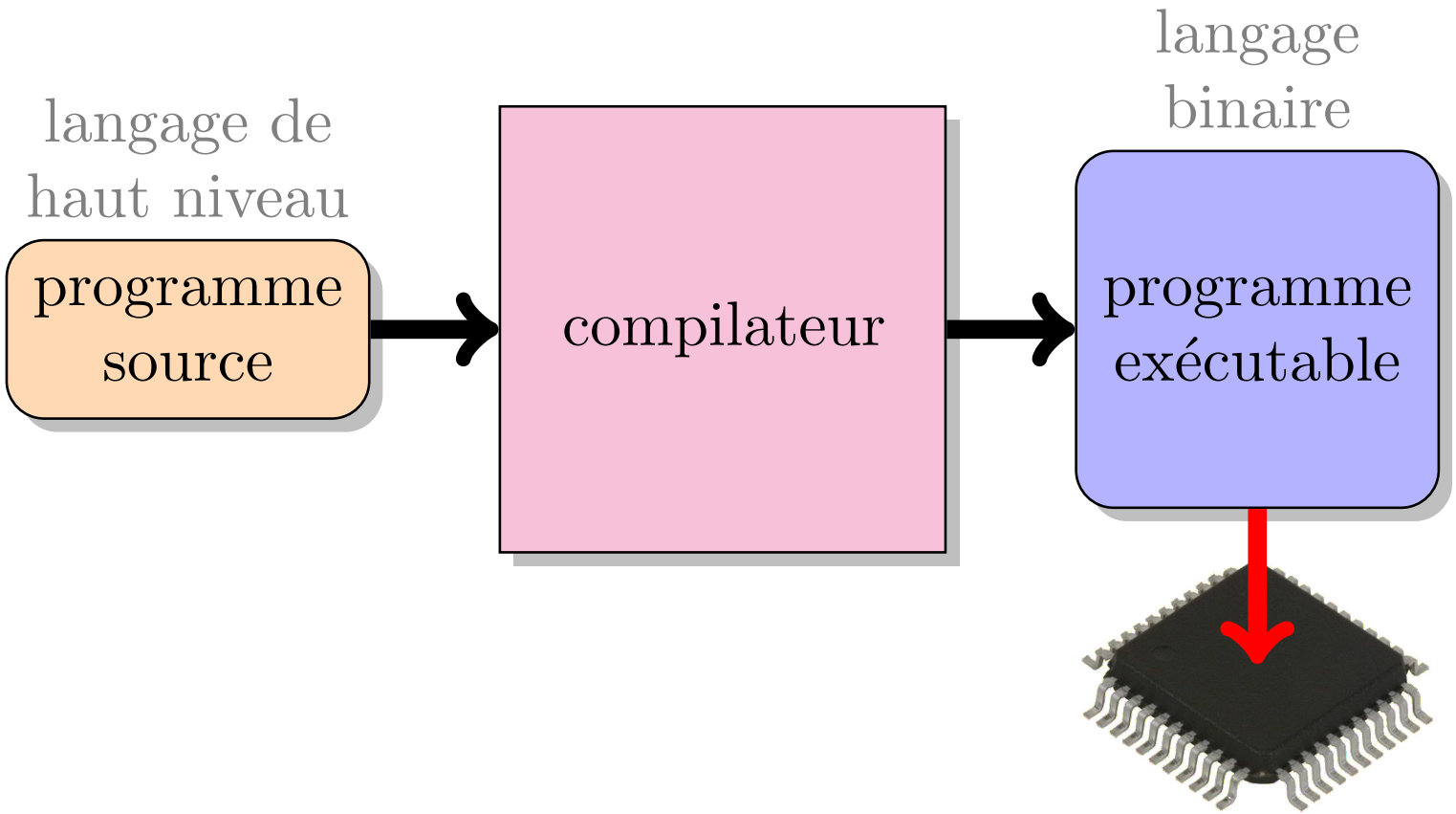

FIGURE 17 – Utilisation type d'un compilateur

Exemple en langage C

```c
#include <stdio.h>

int add(int a, int b)
{
    int r = a + b;
    return r;
}

int main(void)
{
    printf("%d\n", add(1, 4));
    printf("%d\n", add(10, 2));
    return 0;
}
```

- Voir la compilation et l'exécution
- Voir le fichier assembleur généré par le compilateur.

Interpréteur et machine virtuelle

Une machine virtuelle s'exécute sur le processeur. Elle interprète un programme intermédiaire (ou bytecode) produit par un compilateur à partir du programme source.

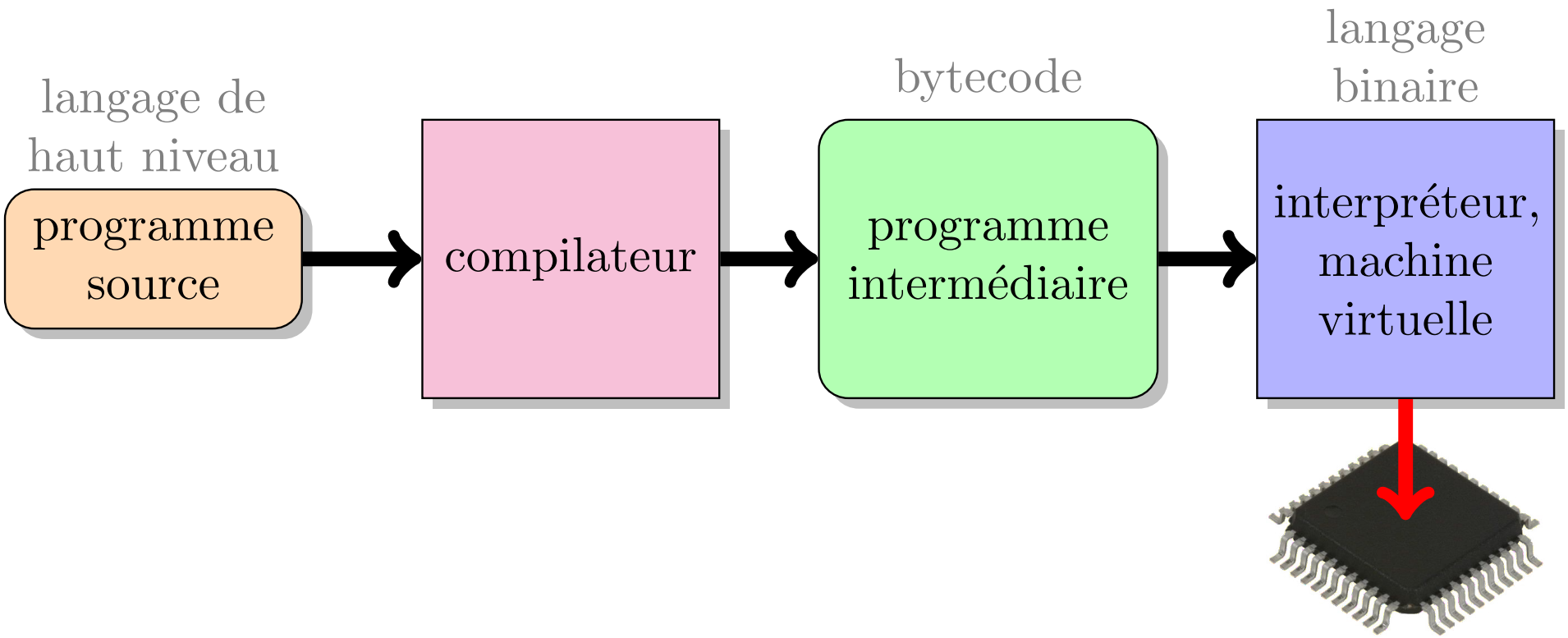

FIGURE 18 – Utilisation type d'un interpréteur et d'une machine virtuelle

Exemple en langage Python

```
def add(a, b):
  r = a + b
  return r

print add(1, 4)
print add(10, 2)
```

- Voir l'exécution
- Voir le bytecode

Systèmes d'exploitation

Organisation logicielle / matérielle d'un ordinateur

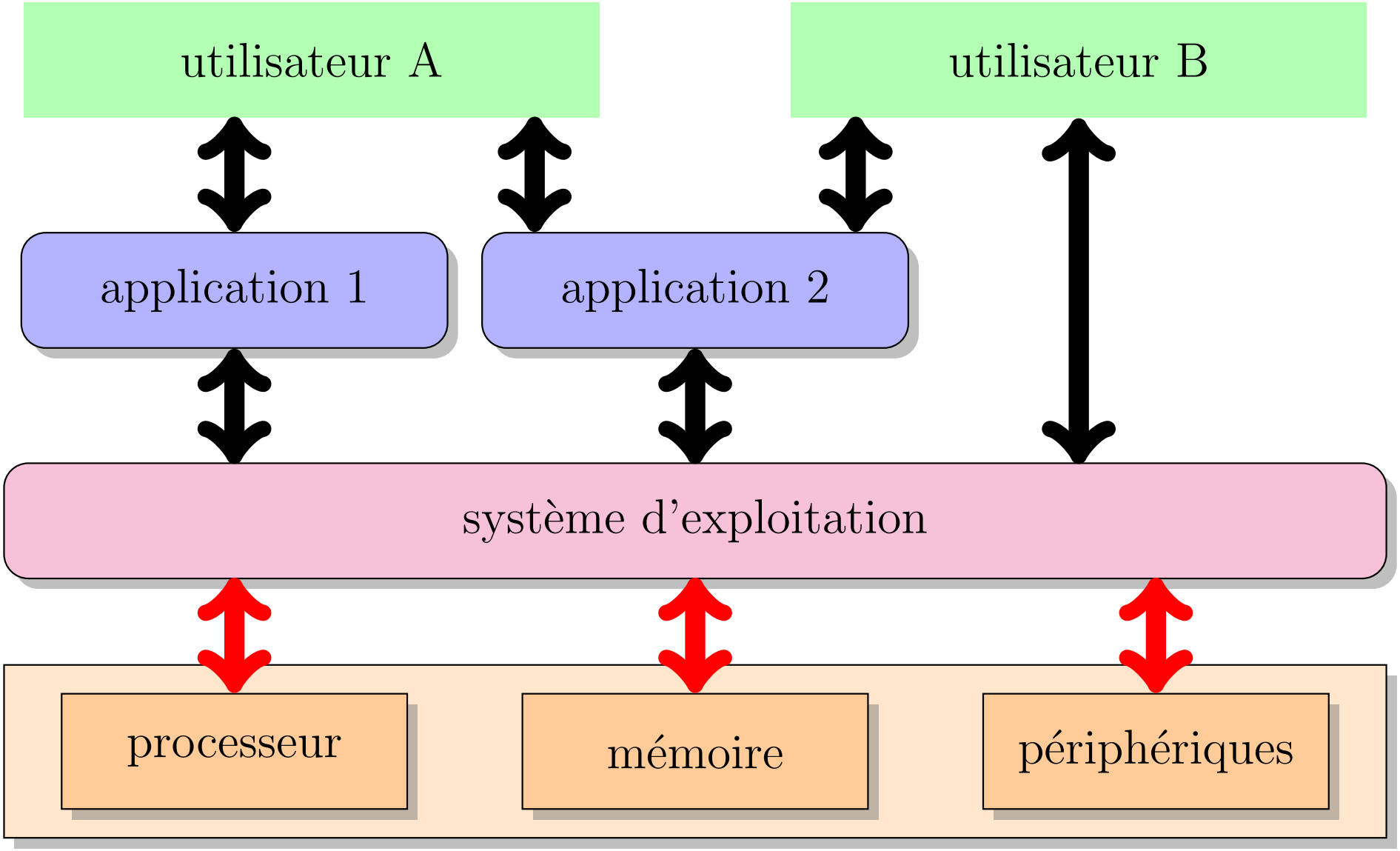

FIGURE 19 – Organisation de l'ordinateur avec un système d'exploitation

Système d'exploitation

Ensemble de programmes qui assure :

- le démarrage et l'arrêt de l'ordinateur
- l'interface et la gestion des différents éléments matériels :
  - processeur(s)
  - mémoire(s)
  - périphériques : écran, clavier, souris, disque dur, réseau, etc
- le bon fonctionnement de l'ordinateur (éviter les conflits d'accès)
- la gestion des processus (lancement et arrêt des programmes)
- la gestion des utilisateurs (droits d'accès)
- la gestion de l'énergie pour les portables
- la gestion des erreurs
- etc

Exemples

- Windows
- MacOS
- Linux
- Android
- etc

Conclusion

Tendances actuelles

- Systèmes sur puce : intégrer dans une seule puce électronique des processeurs, mémoires, accélérateurs graphiques, capteurs, etc
- Processeurs multicœurs (2, 4, 8, 10)
- Processeurs à beaucoup de cœurs *manycores* (100, 1000, 10000)
- Basse consommation d'énergie
- Sécurité contre les attaques
- Etc

Quelques liens

- https://interstices.info/ : Revue de culture scientifique en ligne sur les sciences du numérique.
- http://www.aconit.org/ : Association pour un conservatoire de l'informatique et de la télématique
- http://www.computerhistory.org/ : Computer history museum (en anglais)
- https://www.fetedelascience.fr/ : Site permanent de la FDS
- https://lejournal.cnrs.fr/ : Mensuel en ligne du CNRS
- http://www-labsticc.univ-ubs.fr/~tisseran/dissemination/ : Ma page web

Fin, des questions ?

Contact :

- mailto:arnaud.tisserand@univ-ubs.fr
- http://www-labsticc.univ-ubs.fr/~tisseran
- CNRS, Laboratoire Lab-STICC, Univ. Bretagne Sud, Centre de recherche C. Huygens, rue St Maudé, BP 92116, Lorient.

Merci