

UNE BRÈVE HISTOIRE DE L'INFORMATIQUE

I SYSTÈMES INFORMATIQUES

II ARCHITECTURES DES ORDINATEURS

III REPRÉSENTATION DES NOMBRES

I Les systèmes informatiques

Un ordinateur : qu'est-ce à dire ???

On pensait (en France) l'appeler ordonnateur : Pb existait déjà et était très connoté.
ex : ordonnateur de pompe funèbre...

ORDINATEUR, -Trice, adj

Qui ordonne, dispose, met en ordre.

ORDINATEUR, subst. masc.

INFORMAT.

Machine algorithmique composée d'un assemblage de matériels correspondant à des fonctions spécifiques, capable de recevoir de l'information, dotée de mémoires à grande capacité et de moyens de traitement à grande vitesse, pouvant restituer tout ou partie des éléments traités, ayant la possibilité de résoudre des problèmes mathématiques et logiques complexes, et nécessitant pour son fonctionnement la mise en oeuvre et l'exploitation automatique d'un ensemble de programmes enregistrés.

Éléments périphériques d'un ordinateur; adresse, console, pupitre d'un ordinateur; imprimante, mémoire d'un ordinateur;

En Anglais : **COMPUTER** ---> TO COMPUTE ---> Latin COMPUTARE
« combiner des opérations » => beaucoup plus général que calculator

Un ordinateur : qu'est-ce à dire ???

Se focalise davantage sur la nature du processus mis en oeuvre : --> programmation impérative

ORDINATEUR, -Trice, adj

Qui ordonne, dispose, met en ordre.

ORDINATEUR, subst. masc.

INFORMAT.

Machine algorithmique composée d'un assemblage de matériels correspondant à des fonctions spécifiques, capable de recevoir de l'information, dotée de mémoires à grande capacité et de moyens de traitement à grande vitesse, pouvant restituer tout ou partie des éléments traités, ayant la possibilité de résoudre des problèmes mathématiques et logiques complexes, et nécessitant pour son fonctionnement la mise en oeuvre et l'exploitation automatique d'un ensemble de programmes enregistrés.

Éléments périphériques d'un ordinateur; adresse, console, pupitre d'un ordinateur; imprimante, mémoire d'un ordinateur;

Se focalise davantage sur le résultat (et peut-être plus générale)

En Anglais : **COMPUTER** ---> TO COMPUTE ---> Latin COMPUTARE
«combiner des opérations» => beaucoup plus général que calculator

Origine de l'ordinateur

Tout ce qui répond à la définition précédente est un ordinateur

Qui ordonne, dispose, met en ordre.

Pour simplifier :

Tout ce qui permet de remplacer une succession de tâches simples individuellement mais complexe dans son ensemble, par un automatisme qui va le faire pour nous !

Exemple en musique : l'orgue de barbarie

Le papier entre l'information qui est traduite en son par l'orgue

↑
programme

↑
computer

Papier = 0

Trou = 1



000010101001100001110101010

Exemple en calcul : la Pascaline

→ Première calculatrice
1642



Fonctionnement mécanique
à la main :

très long, surtout pour les
multiplications et les soustractions

http://perso.telecom-paristech.fr/~blanchet/SIP_UE_INF227/histoire/

Son principe de fonctionnement est totalement analogue à celui d'un microprocesseur
[à ceci près qu'elle fonctionne en base 10 et non en binaire]

pour faire des additions, soustractions et multiplication :

Systeme mécanique de
«propagation de retenue»

Améliorée par Huygens : Multiplication/division

Exemple industriel :

Le métier à tisser

- Tâche longue, très simple et répétitive
- Enjeu économique de premier ordre : faire du tissu [au XVIIIème siècle et tjrs. aujourd'hui]

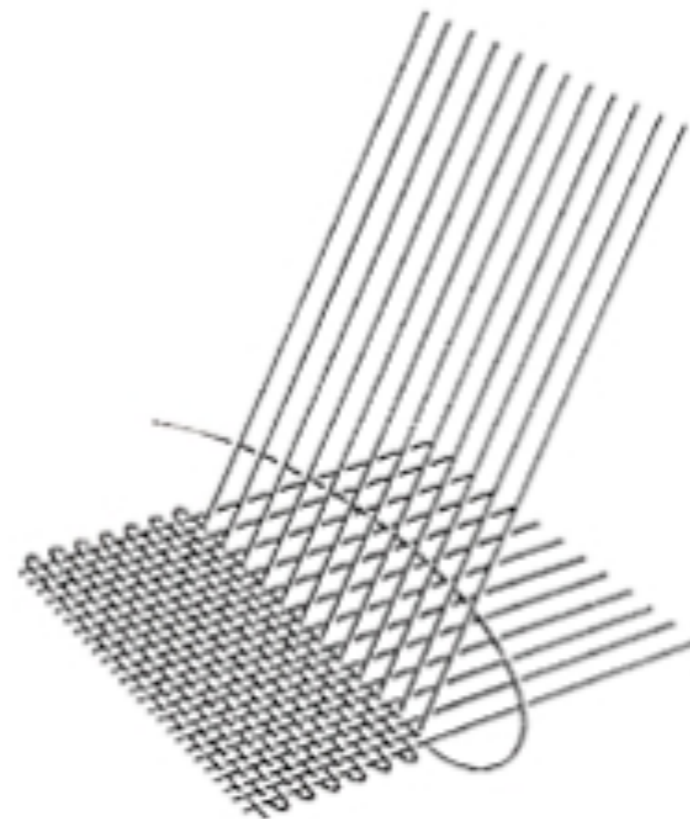
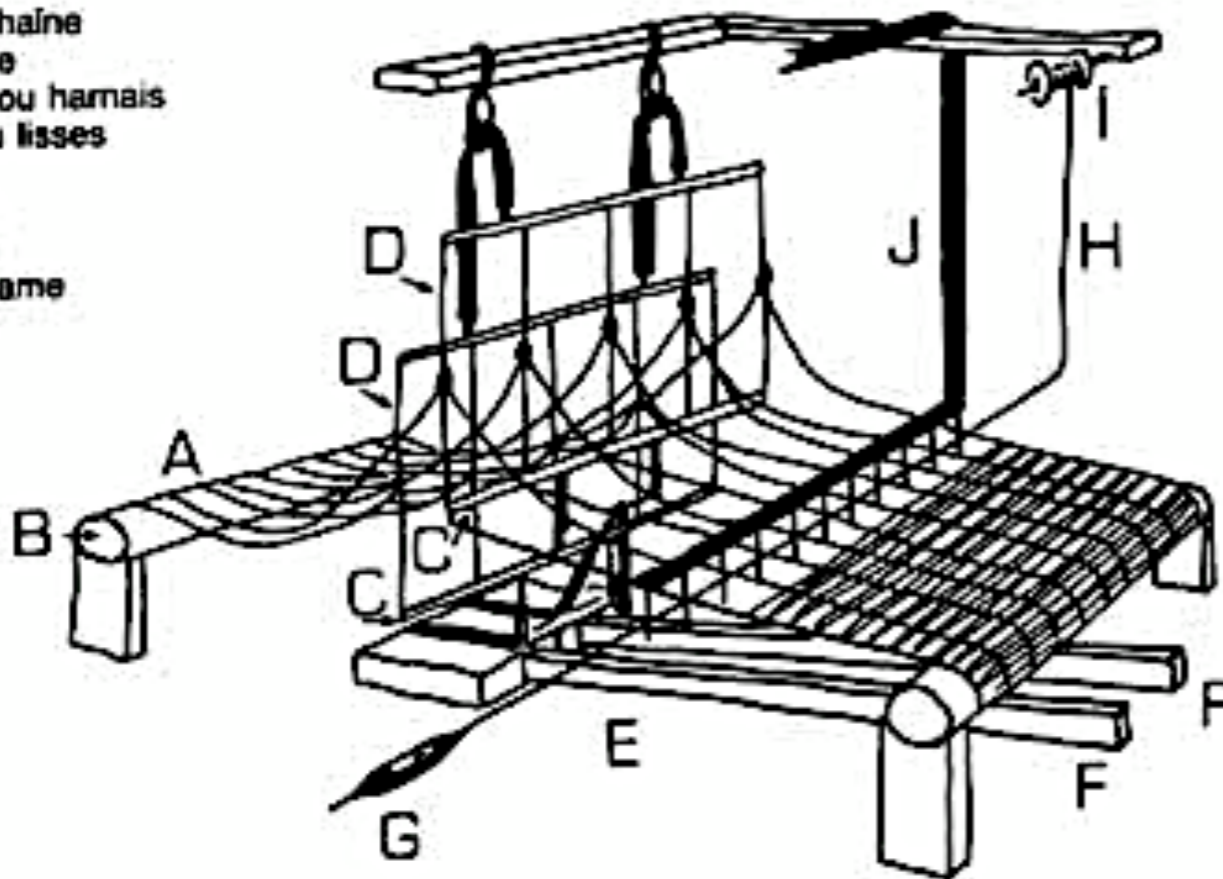
=> on veut l'automatiser !

Principe de base du métier à tisser :

LE TISSAGE ET LA TOILE

Le tissage

- A : Fils de chaîne
- B : EnsoUPLE
- C : Cadres ou hamais
- D : Lices ou lisses
- E : Lames
- F : Pédales
- G : Navette
- H : Fil de trame
- I : Bobine
- J : Battant



Le passage au binaire est élémentaire :

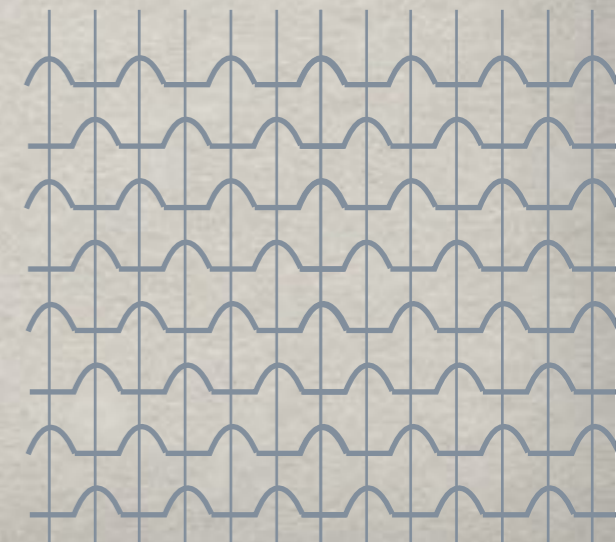
Basile Bouchon inventa en 1725 un rouleau de papier perforé, qui automatise le tissage des étoffes à motifs.

---> Un système de tringleries passe dans les trous pour soulever les fils.

Ce dispositif sera perfectionné par Falcon (1728)

Puis par Jacquard, en 1790, qui révolutionnera l'industrie lyonnaise. Invention déclarée d'utilité nationale malgré les pressions des lobby du textiles menacés par l'invention.

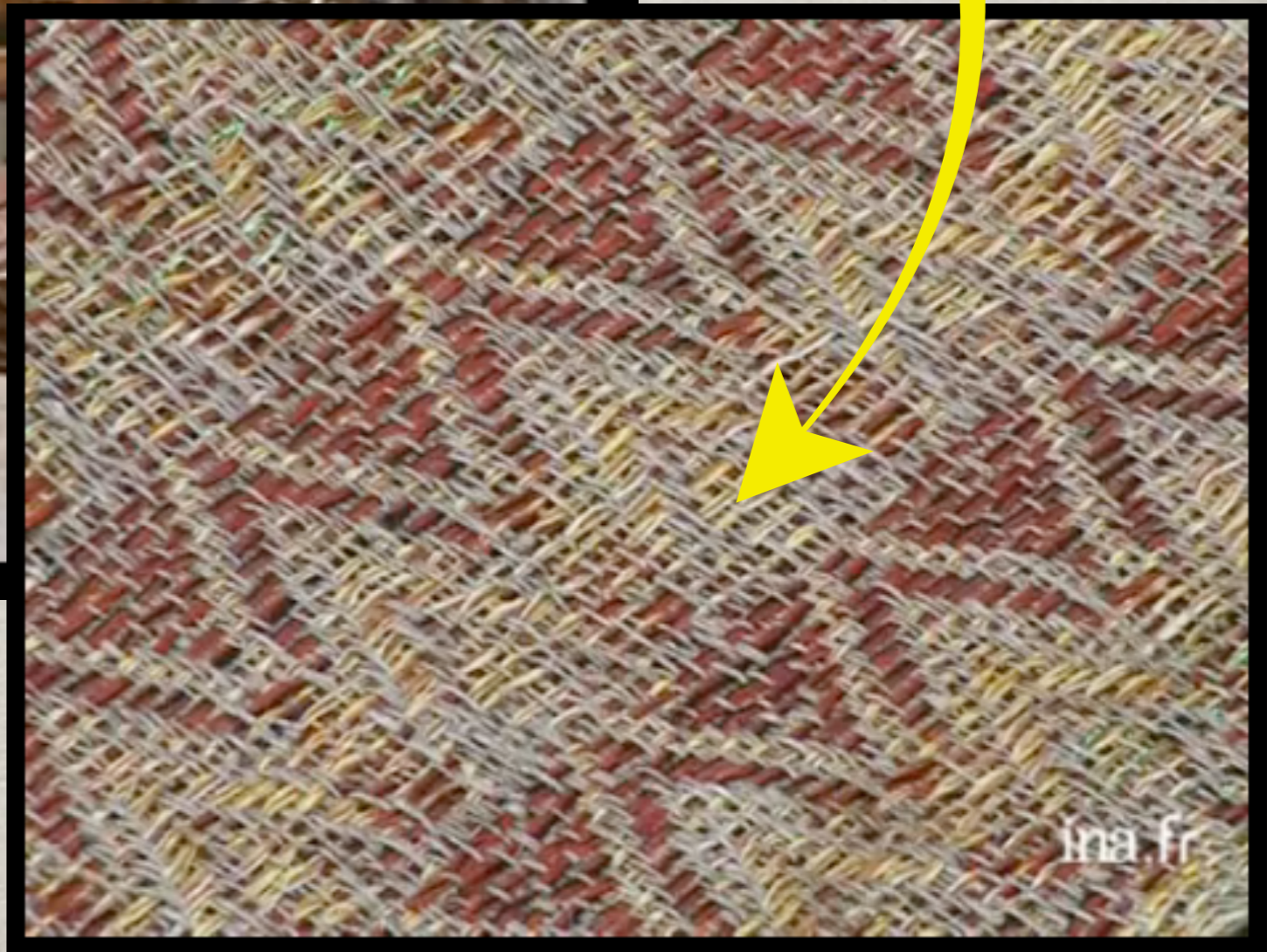
fil en bas = 0
fil en haut = 1
:
:
0 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 1
0 1 0 1 0 1 0 1 0 1
1 0 1 0 1 0 1 0 1 0
0 1 0 1 0 1 0 1 0 1
:
:
:





Programme

Métier à
Tisser



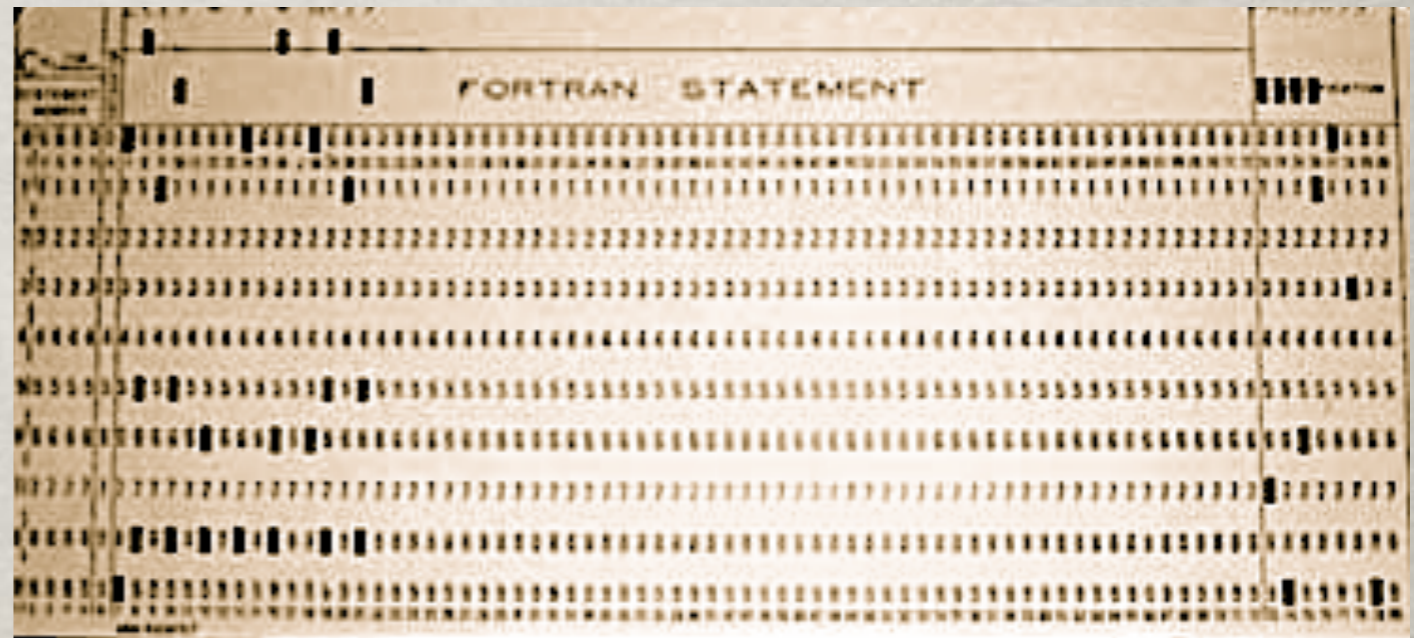
Tissu



Métier à tisser Jacquard (1790)
Musée des Arts et Métiers, Paris.
Photo : © David Monniaux

Les cartes perforées :

→ une idée bicentenaire.



Carte perforée FORTRAN utilisée en informatique dans les années 1960.
Photo : J.M. Favre

Beaucoup de Français veulent y voir l'ancêtre de l'ordinateur :

made in France

Le métier a tisser :

<http://www.ina.fr/video/VDD09031288>

Le **papier** entre l'information qui est traduite en son par **le métier**

↑
programme

↑
computer



0 = pas de trou

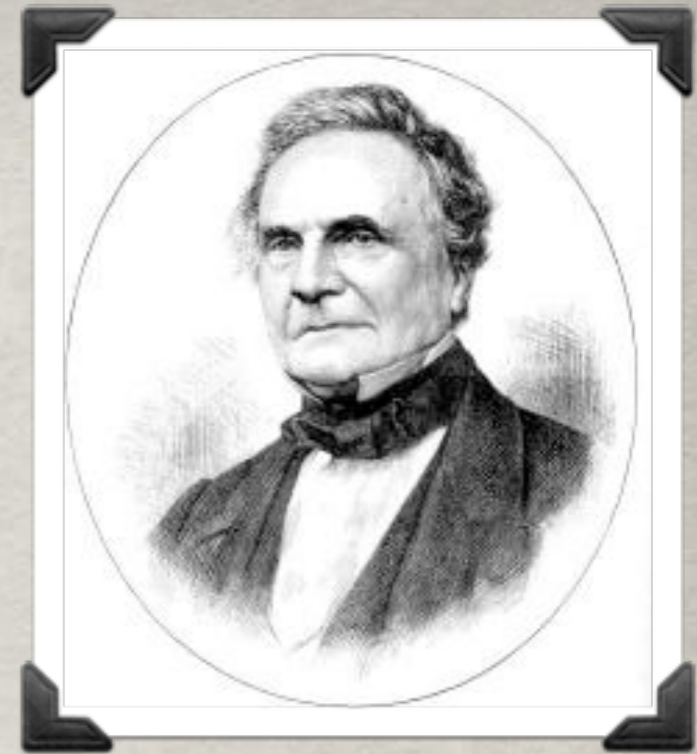
→ Fil bloqué en bas

1 = trou

→ Tige passe à travers pour lever le fil

La machine de Babbage :

The Analytical Engine



CHARLES BABBAGE
(1792-1871)

Le grenier [Stockage/mémoire]

- Stock de ~1000 nombres de 50 chiffres

Le moulin [unité de calcul : arithmétique et logique]

- Le moulin peut chercher un nombre dans le grenier et y placer le résultat d'une opération
- Il réalise les quatre opérations de base et des tests logiques : branchement conditionnel !
- réalisation de boucle conditionnel type While

Lecteur de cartes perforées d'après le modèle de Jacquard [entrée]

- C'est par ce biais que l'on envoyait le programme dans la machine

Les résultats était gravés sur des feuillets métalliques [sortie]

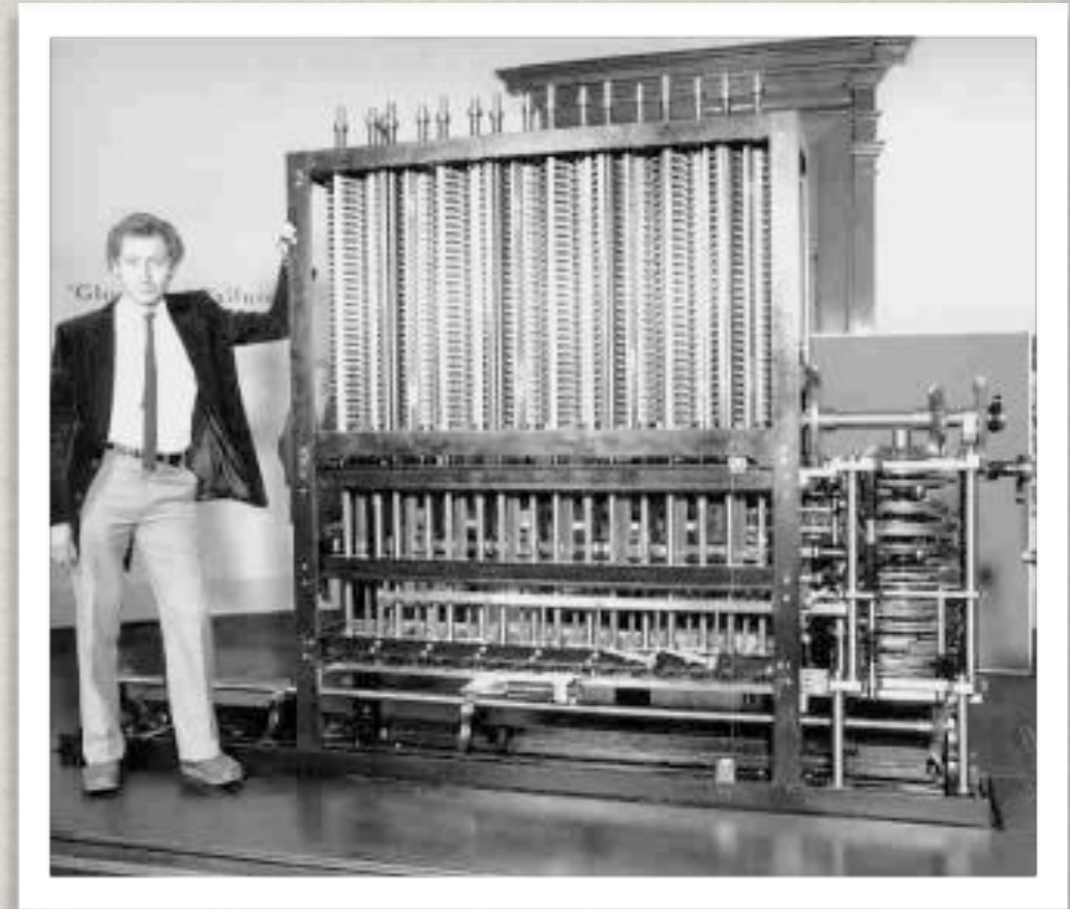
- On obtenait ainsi le résultat des calculs.

Le tout fonctionnant bien sûr

... à la vapeur !!!

Les machines Babbage, n'ont pas abouti de son vivant à une réalisation concrète, par manque de financement.

Son premier calculateur : Difference Engine a finalement été réalisé par le musée des sciences de Londres et fonctionne !



L'**Analytical Engine** a servi de modèle à IBM pour concevoir le plus gros calculateur électromécanique au monde : **Harvard Mark I**
[3 additions à la seconde, multiplication en 6 secondes !!!]

Les branchements conditionnels et les boucles : idées pour le moins visionnaire du temps de Babbage n'ont été réalisées qu'après la seconde guerre mondiale avec des calculateurs électroniques à transistors [IBM].

L'ordinateur n'est pas nécessairement une machine électrique ?



Mais actuellement l'électricité est le moyen le plus :

- pratique
- économique
- rapide

RQ: De plus, tout n'est pas 100% électrique :

- Stockage : disque dur et bande --> magnétique
CD - DVD ---> lecture optique sur surface sérigraphiée
carte perforée : papier.....
- Transport / communication : Fibre optique, onde radio.

Autre idée ???

Ordinateur quantique....
Ordinateur à eau !

Reste théorique

L'électricité est un moyen technologique, pas une fin en soi !

Vers l'ordinateur actuel :

L'ordinateur d'aujourd'hui est le résultat d'avancées
aussi bien conceptuelles que techniques

Evolutions conceptuelles

GEORGE BOOLE (1815-1864)

1854 : L'algèbre booléenne définit un ensemble à deux éléments contenant *vrai* et *faux* ainsi que des tables d'opérations pouvant toutes se ramener à deux opérations de base : OU , ET auxquelles on ajoute la négation NON

OU	V	F
V	V	V
F	V	F

ET	V	F
V	V	F
F	F	F

A	NON(A)
V	F
F	V

Dans l'ordinateur le Vrai prend la valeur 1, et le Faux la valeur 0

CLAUDE SHANNON (1916-2001)

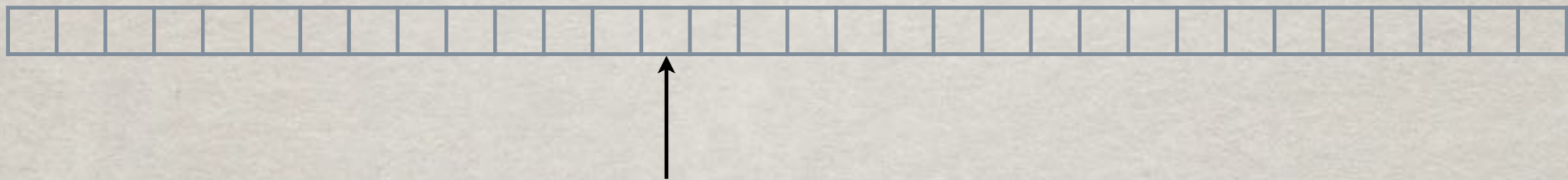
Dans sa thèse, il fit le rapprochement entre les nombres binaires, l'algèbre de Boole et les circuits électriques. Il réalise lui même les premiers circuits électroniques a commutation, ce qui a permis ensuite la création des portes logiques. [MIT-BELL Laboratories]

On comprendra ensuite que les opérations mathématiques peuvent être réalisées aussi bien en binaire qu'en décimal. [cf représentation des nombres]

Alan Turing : Invente la notion de «machine de Turing» :

C'est une définition purement théorique, extrêmement simple et générale de tout processus de calcul qui peut-être réalisé par une machine.

Il pose ainsi les bases de la lecture-écriture dans un registre de mémoire de taille infini.



La machine de Turing lit un programme qui lui indique à chaque étape comment elle doit :

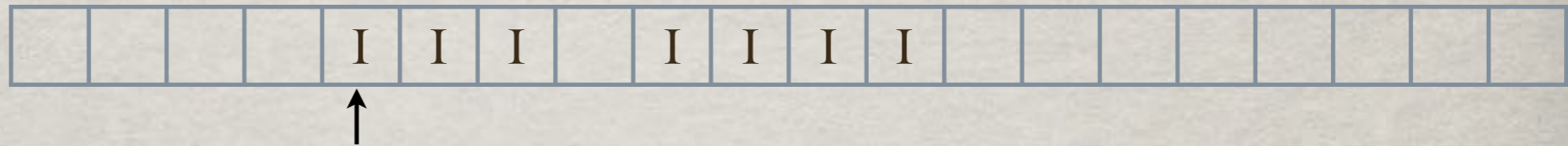
- lire l'état de la mémoire et en fonction de sa valeur :
- se déplacer dans la mémoire
- [éventuellement] modifier l'état de la mémoire

Tout algorithme, tout calcul peut s'écrire comme une succession d'opérations triviales réalisées par une machine de Turing.

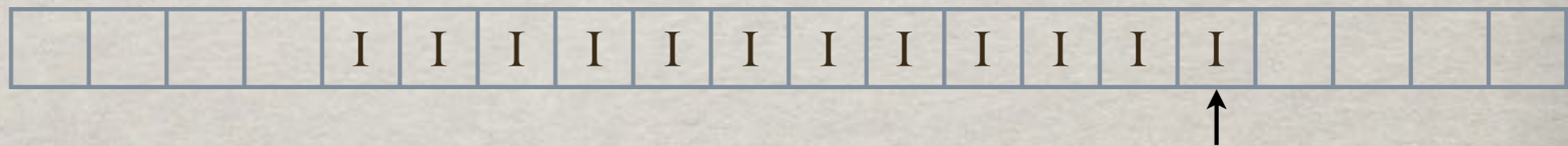
Exemple : programmer toute multiplication dans un alphabet a un seul caractère : {I} «bâton»

```
B,I, ,1
1, ,>,2
2,I, ,3
2, ,>,17
3, ,>,4
4,I,>,4
4, ,>,5
5,I,>,6
6, ,<,19
6,I, ,7
7, ,>,8
8,I,>,8
8, ,>,26
26, ,I,9
26,I,>,26
9,I,<,9
9, ,<,10
10,I,<,24
24,I,<,24
24, ,I,25
25,I,>,6
10, ,I,11
11,I,<,11
11, ,<,12
12,I,<,16
16,I,<,16
16, ,>,2
12, ,>,13
13, ,>,14
14,I, ,15
14, ,I,E
15, ,>,14
17,I, ,18
18, ,>,17
17, ,I,E
19,I, ,20
20, ,<,21
21, ,<,22
22,I, ,23
23, ,<,22
22, ,I,E
```

Etat initial de la mémoire :



Etat final de la mémoire :



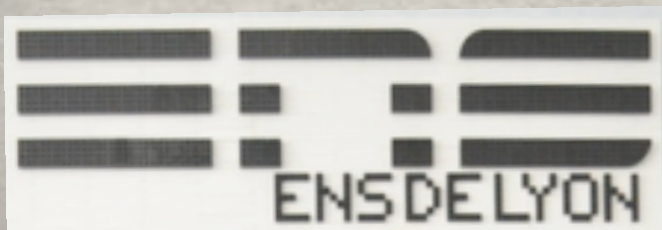
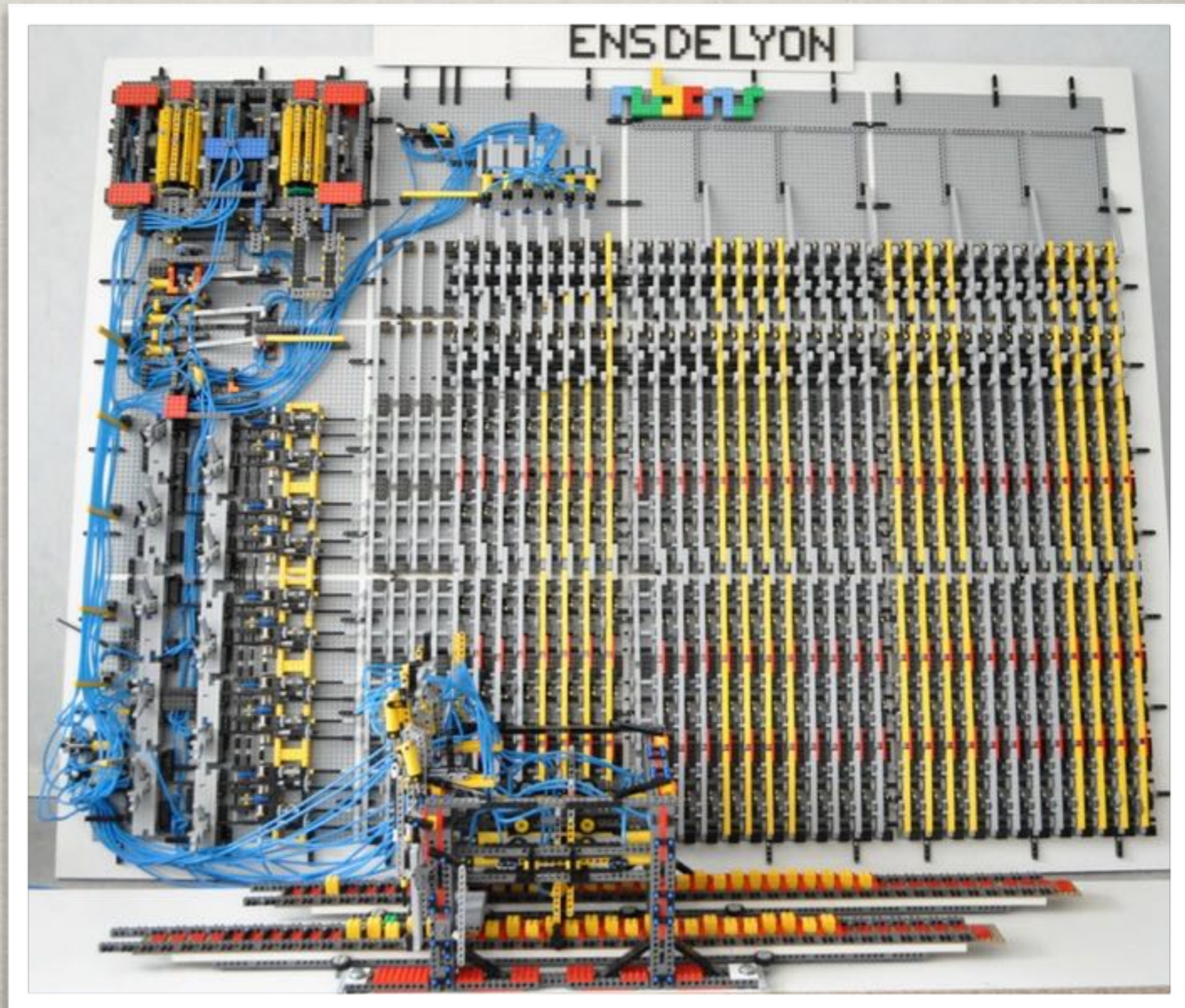
Le programme est une stratégie de «manipulation» bien concrète de la mémoire, qui vise à restituer l'opération abstraite étudiée : ici la multiplication

Réalisation d'une machine de Turing binaire : alphabet = {0,1}

C'est l'équivalent de notre «processeur» ou du moins de l'« unité de calcul » mais ici :

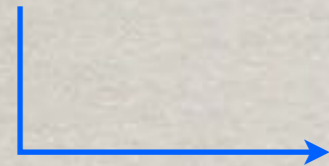
- En LEGO
- a air comprimé
(avec une pompe à vélo)

Laboratoire informatique
de l'ENS-Lyon



Evolutions techniques

L'évolution s'est faite en répondant à deux contraintes antagonistes :

 Miniaturisation vs Fiabilité des composants

Engrenages :

Systemes purement mécaniques, chaque pièce commande le déplacement de la suivante.

Le relai électromécanique :

Circuit doté d'un électro-aimant agissant comme un interrupteur sous l'action d'un autre courant

Le tube à vide : amplificateur de courant que l'on peut également utiliser comme interrupteur

Le transistor : semi-conducteur dopé [NPN ou PNP] miniaturisé à l'extrême [MQ]

C'est à la base un amplificateur électronique qui peut-être utilisé comme interrupteur commandé

Tous ont en commun d'être des interrupteurs commandés

La guerre des machines

ENIGMA-COLOSSUS : Machine «à écrire»
(1944) engrenages cryptage/décryptage
des communications nazis

Z3 [militaire] : 1 tonne, 600 relais UAL,
All 1600 mémoire détruite par les alliés
opération en 3 secondes

MARK I [IBM] : 5 tonnes de relais
opération en 6 secondes - calcul des logarithmes
cartes perforées
pas de branchement conditionnel

ENIAC [militaire] : 30 tonnes - 20.000 tubes à vides
(1946) US 1 opération par seconde calculs de la bombe H
programmation par câblage...

John Von Neumann :

Durant la deuxième guerre mondiale, Von Neumann, mathématicien génial, est le spécialiste en Balistique et en explosifs de l'armée US. C'est donc un acteur clef du projet Manhattan à Los Alamos.

Il voit tout de suite l'intérêt des ordinateurs dans la modélisation numérique.
de la diffusion de la chaleur. Processus complexe mais essentiel lors de l'explosion :

Von Neumann a calculé l'altitude d'explosion produisant des dégâts maximums

EDVAC (1949-1961)

Ballistics Research Laboratory

US ARMY

8 tonnes - 4000 tubes à vides 10.000 diodes

1 opération par ms, opérations sur les flottants
cartes perforées.

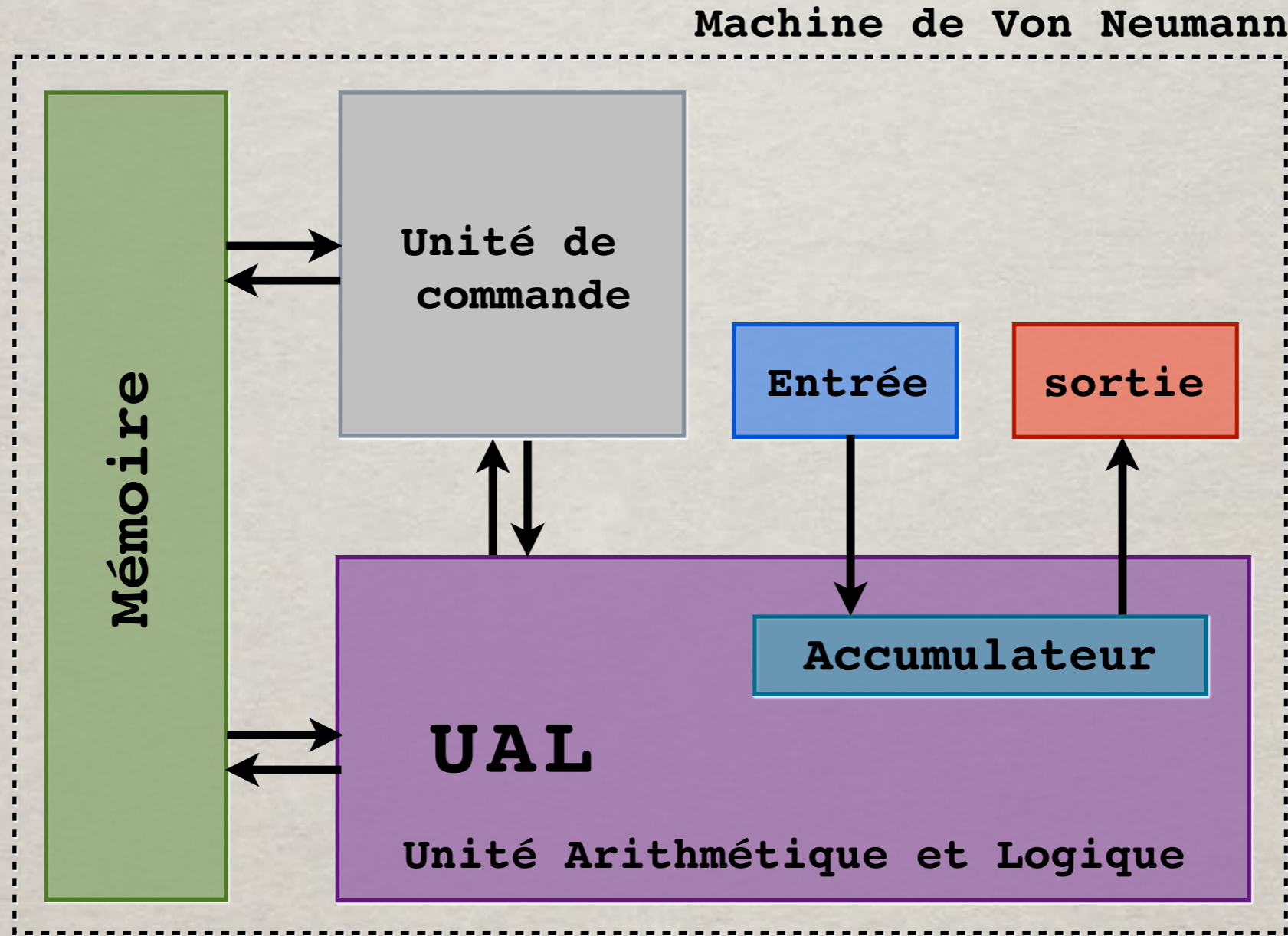
Beaucoup moins de pannes que l'ENIAC

Il combine l'Analytical Engine de Babbage avec l'idée de Machine de Turing, pour proposer ce qui est toujours l'architecture de base d'un processeur.

C'est en particulier Von Neumann qui a l'idée de stocker le programme dans la mémoire

Architecture de Von Neumann :

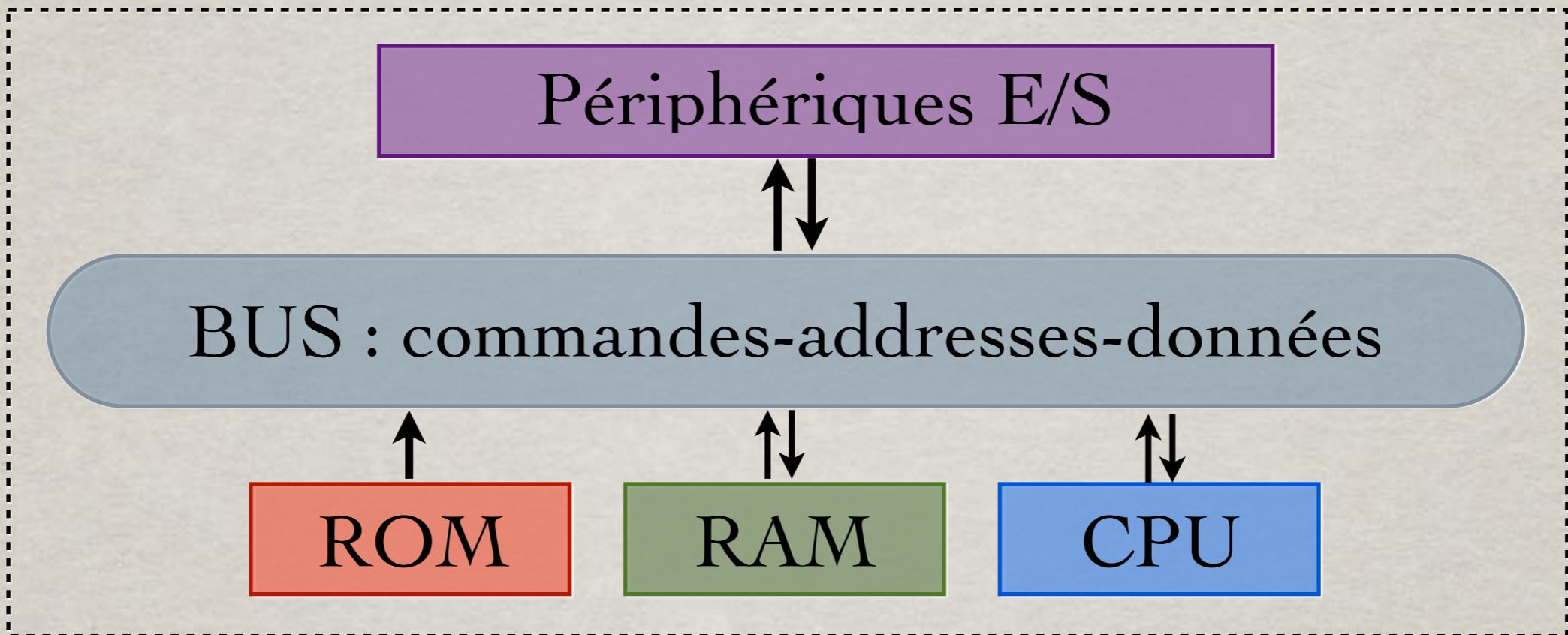
(attribuée à J.V.Neumann en tout cas)



Rq - On distingue ici :

- UC : qui dirige co-ordonne les opérations et les accès mémoires
- UAL : qui réalise véritablement les calculs

Structure d'une architecture monoprocesseur actuelle



on retrouve la machine de Von Neumann

CPU - Central Processing Unit :

Élément de l'ordinateur qui interprète et exécute les instructions d'un programme.

[CPU n bits les informations sont codées sur n bits]. Mémoire(s) Cache(s) [au sein du CPU]

ROM - Read Only Memory : mémoire « morte » que l'on ne peut que lire : par extension elle désigne la mémoire qui persiste lorsque la machine est éteinte, bien que modifiable.

RAM : Random Access Memory : mémoire vive (lecture et écriture) et volatile (se vide quand il n'y a plus de courant) ; c'est (généralement) une suite de cases contenant 8 bits (soit un **octet**), chaque case ayant sa propre adresse.

Périphériques : [tous ont leur propre système d'adressage]

disque dur (E/S), lecteur disque (E/S), clavier (E), souris (E), écran (S), imprimante (S), accès réseau (E/S), aux formes multiples [Câble Ethernet LAN, Wifi, Bluetooth].

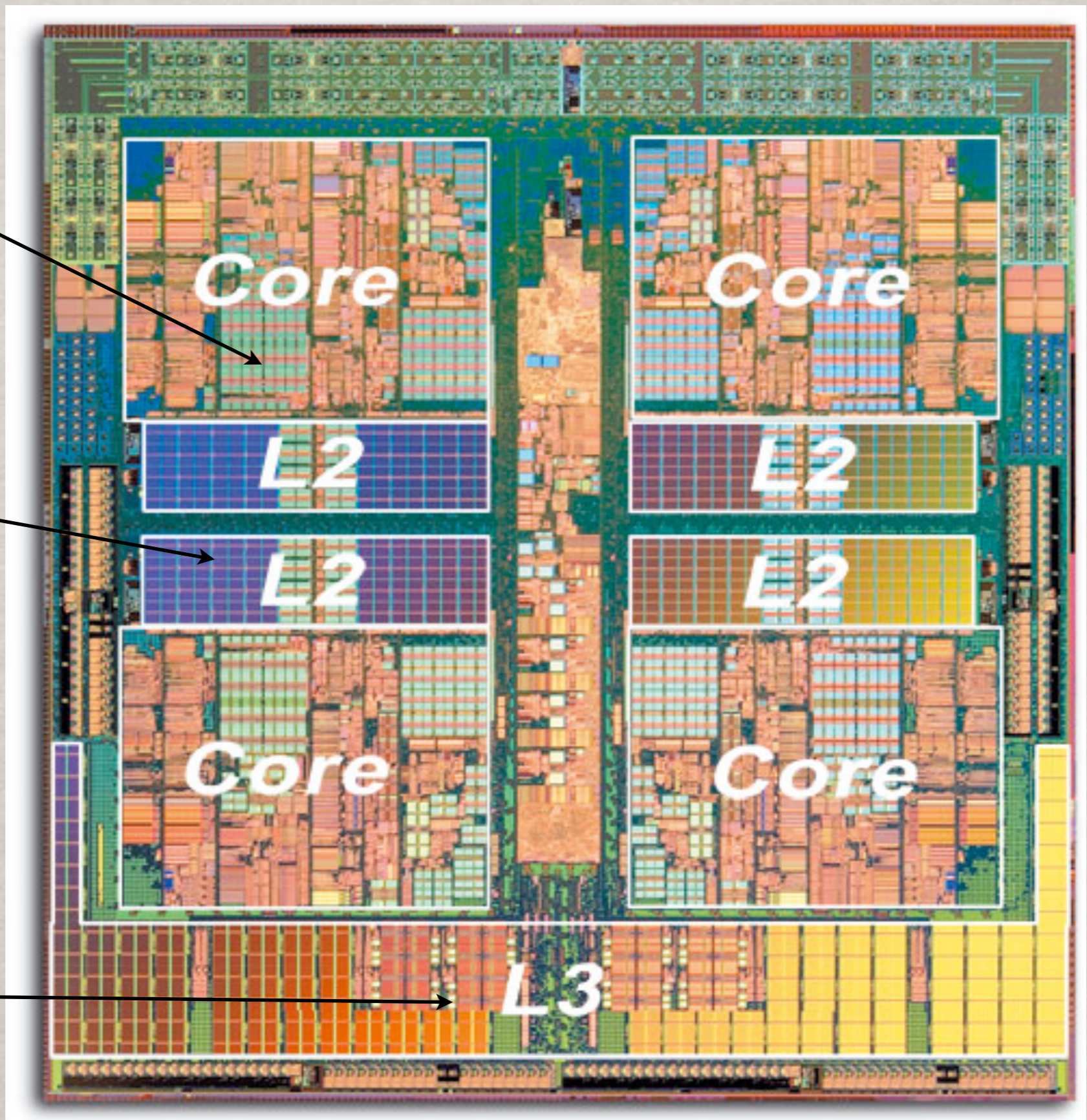
Un «vieux» processeur : 2007



L1
[128 ko]

L2
[512 ko]

L3
[2 Mo]



463 millions de Transistors !!!

Rq - Cette loi est amenée a ne plus être vérifiée :

- Effets quantiques liés à une sur-miniaturisation
- Plusieurs coeurs => diminution du nombre de transistors pour une même efficacité.

De plus elle est très empirique et a été ajustée a de nombreuses reprises.

TOP 500

été 2015

RANK	SITE	SYSTEM	CORES	RMAX (TFLOP/S)	RPEAK (TFLOP/S)	POWER (KW)
1	National Super Computer Center in Guangzhou China	Tianhe-2 (MilkyWay-2) - TH-IVB-FEP Cluster, Intel Xeon E5-2692 12C 2.200GHz, TH Express-2, Intel Xeon Phi 31S1P NUDT	3,120,000	33,862.7	54,902.4	17,808
2	DOE/SC/Oak Ridge National Laboratory United States	Titan - Cray XK7 , Opteron 6274 16C 2.200GHz, Cray Gemini interconnect, NVIDIA K20x Cray Inc.	560,640	17,590.0	27,112.5	8,209
3	DOE/NNSA/LLNL United States	Sequoia - BlueGene/Q, Power BQC 16C 1.60 GHz, Custom IBM	1,572,864	17,173.2	20,132.7	7,890
4	RIKEN Advanced Institute for Computational Science (AICS) Japan	K computer, SPARC64 VIIIfx 2.0GHz, Tofu interconnect Fujitsu	705,024	10,510.0	11,280.4	12,660
5	DOE/SC/Argonne National Laboratory United States	Mira - BlueGene/Q, Power BQC 16C 1.60GHz, Custom IBM	786,432	8,586.6	10,066.3	3,945

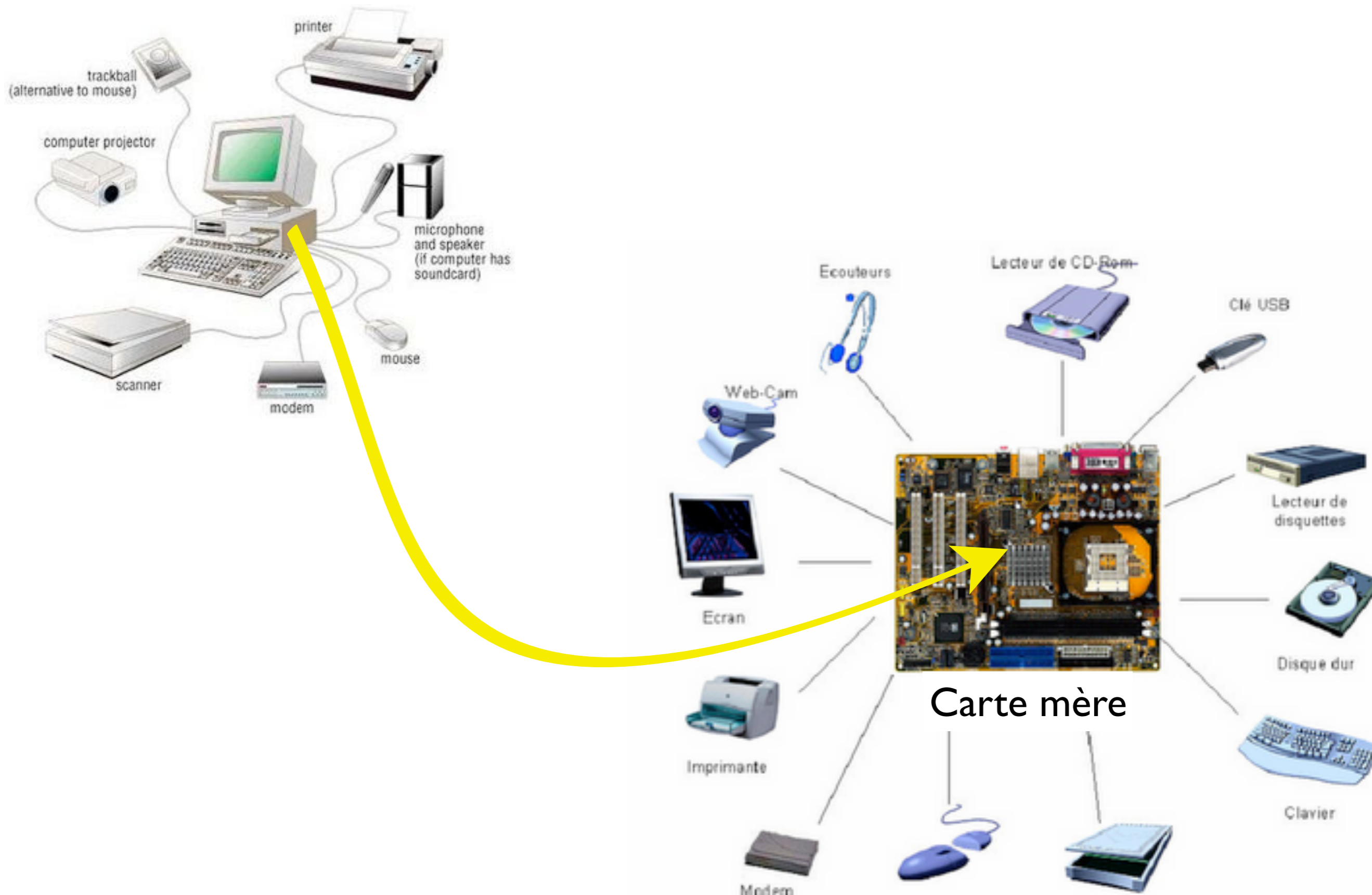
II Interfaces informatiques

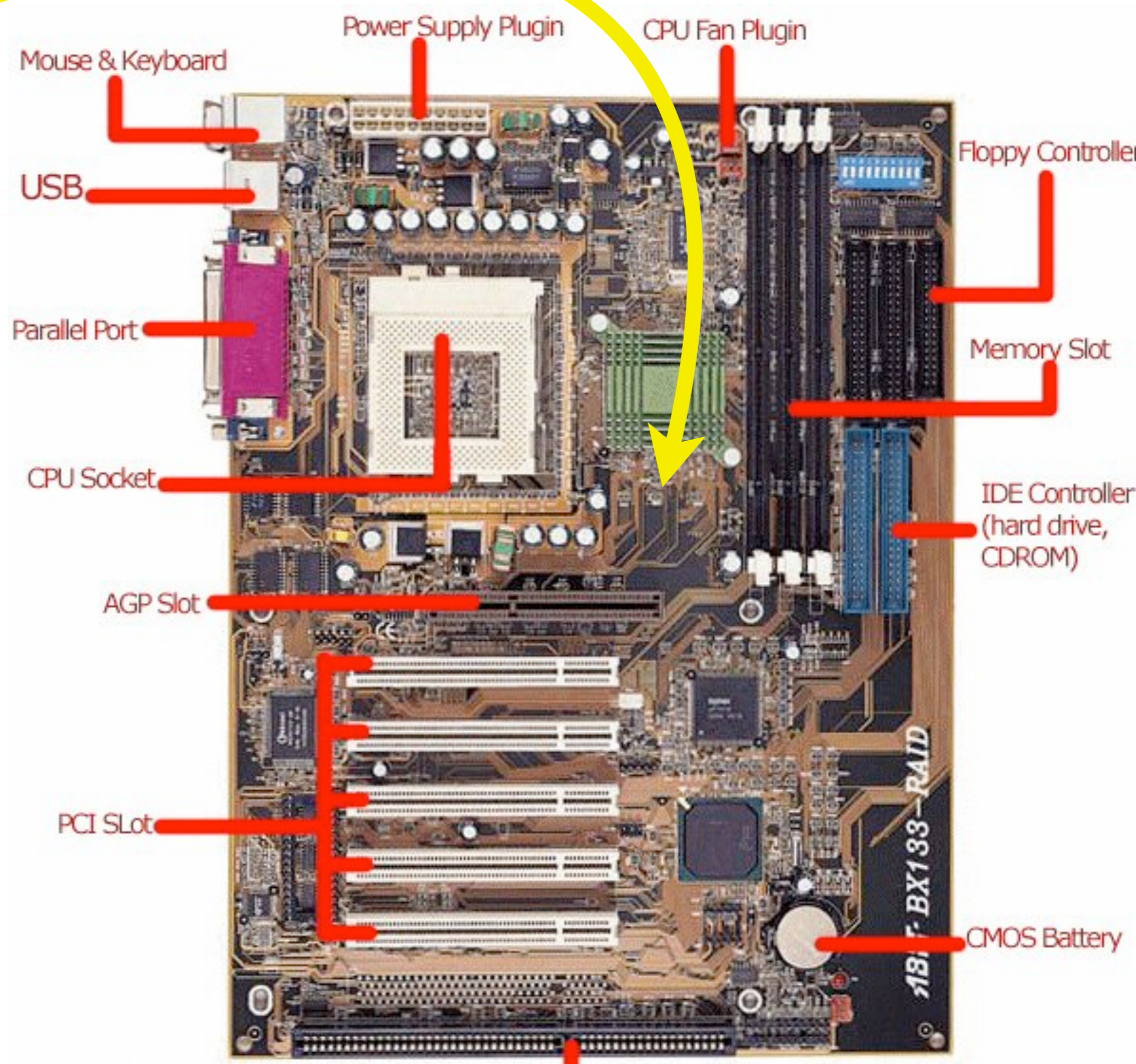
OS : Operating System

Programme informatique

Un ordinateur personnel : PC

et ses périphériques...





1 - OS : Operating System

Il s'agit d'une interface fondamentale permettant :

- à l'utilisateur de commander des programmes en mémoire
- aux programmes d'accéder aux ressources en calcul/mémoire de la machine
- de communiquer avec les périphériques

Au démarrage :

BIOS : Basic Input Output System



Ce n'est pas un OS car il est pré-programmée en ROM sur la carte mère mais c'est lui qui initialise le système et lance l'OS

Il existe des centaines d'OS, mais deux familles ressortent

Unix [Bell 1969]

Terminal de commandes UNIX (shell, bash, etc...)

- Recherche
- Université -> plus de 90% des super calculateurs
- Armée
- Réseaux

→ **Mac OS, iOS**

(versions graphiques)

Click ! Click !

→ **GNU-Linux** [Ubuntu, Debian, Raspberry Pi, Red Hat], **Android**

→ **Solaris [Sun]** et bien d'autres

DOS [Microsoft 1981] : Disk Operating System

(versions light de Unix pour les PC IBM)

→ **Windows** [version graphique]

- PC particuliers
- Entreprise/Public : bureautique de base -> 90% des machines => effet de monopôle
- Commerce

Unix shell

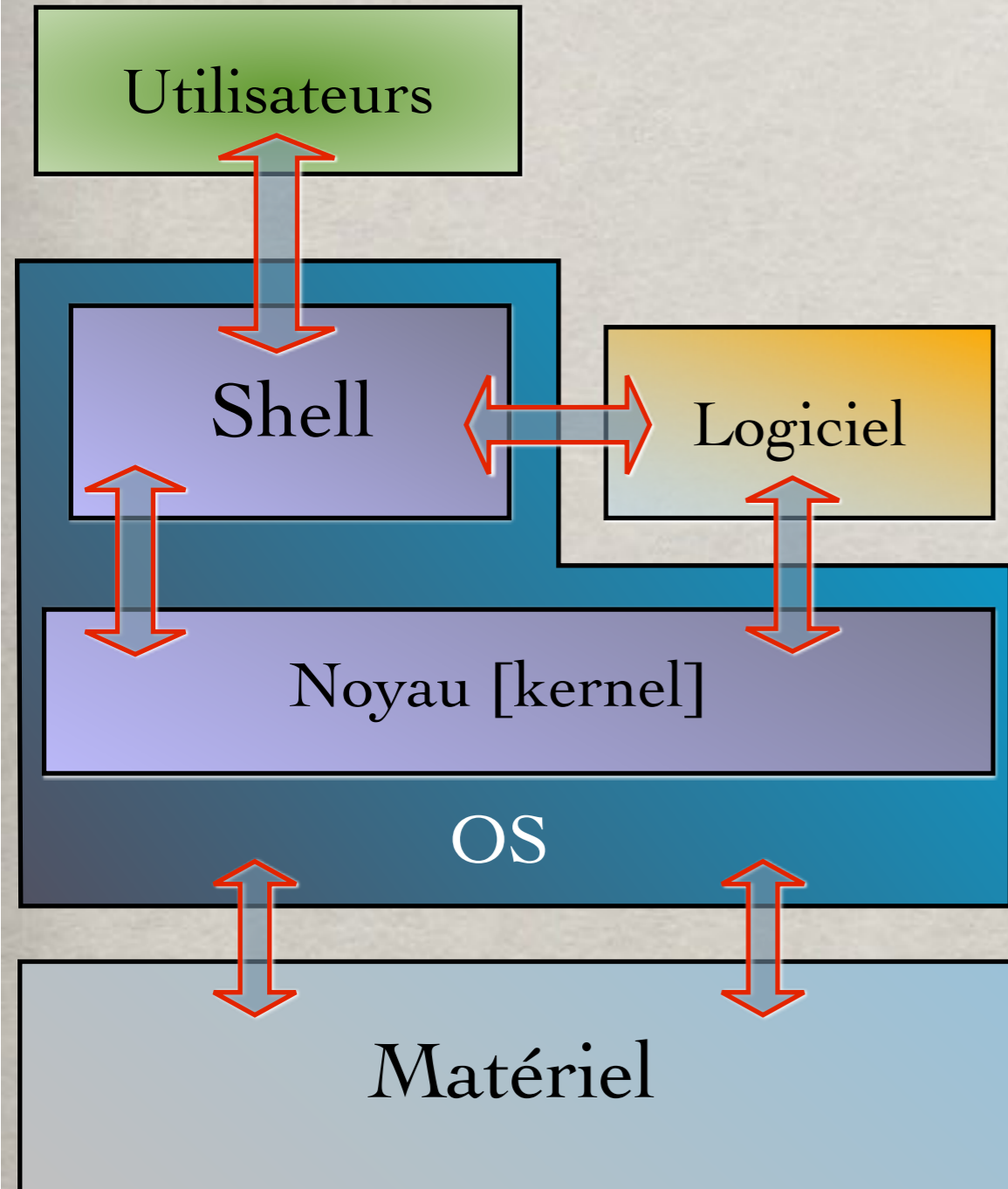
Terminal de commande :

- accès/gestion disques et fichiers très aboutis
- exécution/compilation programmes
- accès réseaux/messageries :
permet l'utilisation a distance de tout système Unix-like sur n'importe quelle machine.
- écriture de scripts shell [ex: C - shell]
=> programmation d'opérations système qui peut être très élaborée

ex :

Il faut crée 225 comptes et gérer les droits pour : des étudiants/professeurs/stagiaires d'une rentrée universitaire

=> on programme un script shell avec lecture d'une base de donnée



Anatomie du système Unix

2 - Programme informatique

Le programme informatique est un simple document textuel écrit dans un langage dit de programmation [ex : Python, C, Java] qui est proche de nos représentations abstraites des choses :

« $a = b + c$ » ; «si $x == 5$ alors faire ceci sinon faire cela».

Le programme est aussi un document à l'adresse d'un autre programmeur qui devra le modifier. Il doit donc être rédigé proprement, et être commenter : **Programmer c'est rédiger.**

Le programme est rédigé dans un **environnement de développement** qui offre :

- une certaine clarté [coloration syntaxique, indentation automatique, proposition des variables]
- des outils de débogage [interprétation, pas à pas, affichage des valeurs en direct].

Le programme en l'état ne veut rien dire pour le processeur qui ne connaît que : 0, 1, trouver une case mémoire, modifier la valeur en mémoire etc.

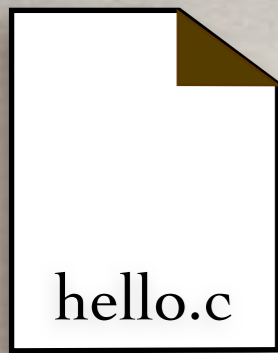
C'est le travail du compilateur de transcrire le langage abstrait en une succession d'opérations élémentaires qui pourront être accomplies très vite par le processeur.

Un bon compilateur comprend que certaines opérations sont inutiles [ex: invariant de boucle] et optimise les affectations, les boucles pour réduire les temps de calcul sans en changer le sens.

Enfin le compilateur, fait le lien entre un langage et un processeur :

Autant l'exécutable dépend peu du langage utilisé autant il dépend du processeur qui va le lire.

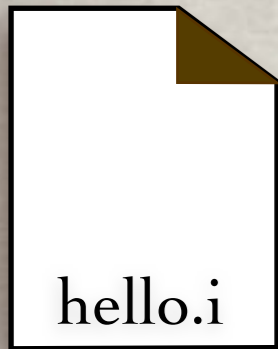
Création d'un exécutable



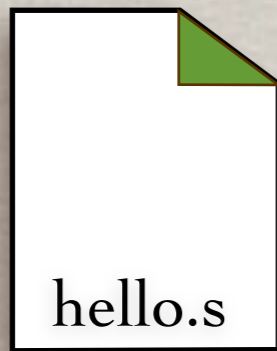
Code source :

c'est du texte

pré-processing
[vérifie la syntaxe]



compilation

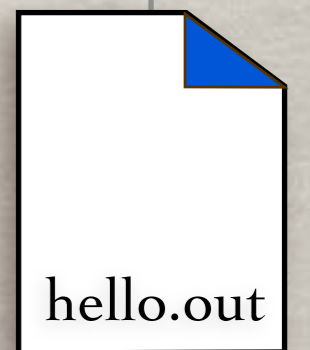


programme assembleur

assemblage
[langage machine]



reliage
[importe les bibliothèques]



Exécutable

Compilateur GNU

```
gcc hello.c -o hello.out
```

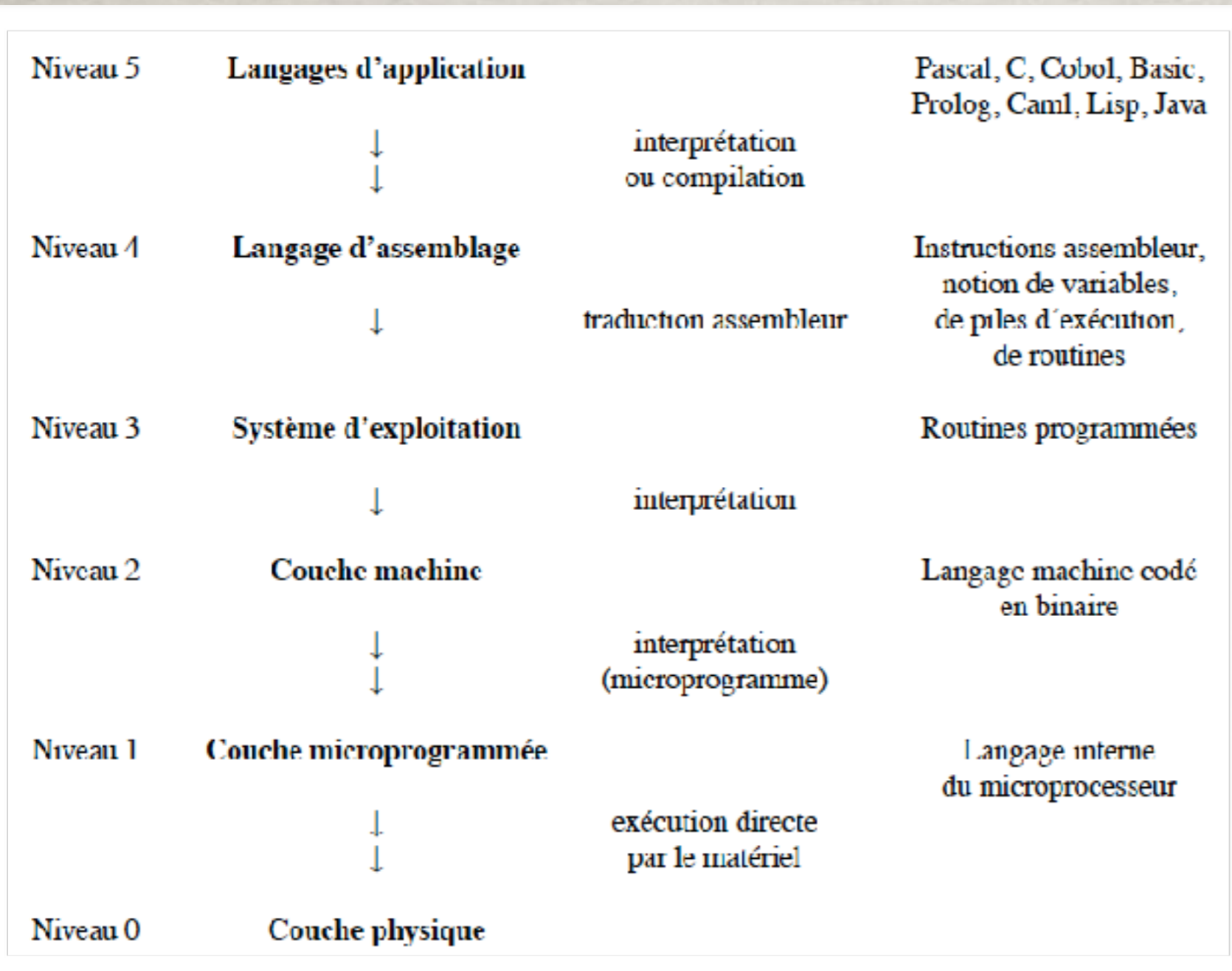
```
./hello.out
```

binaire : langage machine

Exemple concret :

«Hello World»

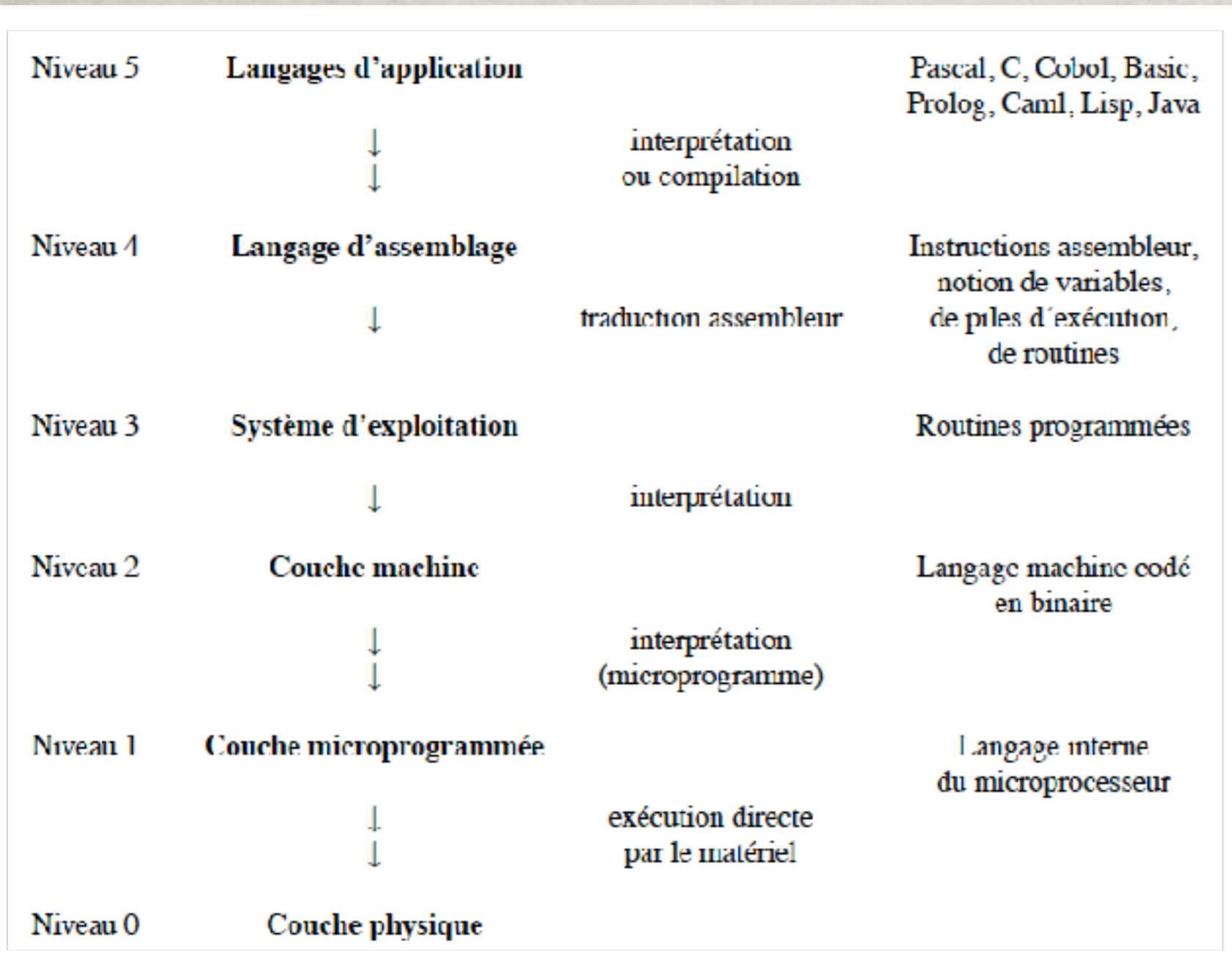
Dans toutes ses composantes l'informatique est un problème de couches !



Interprétation :

Chaque instruction du langage L_n est traduite en la séquence adéquate du langage L_{n-1}
À chaque exécution il y a ré-interprétation à travers toutes les couches.

Dans toutes ses composantes l'informatique est un problème de couches !



Compilation :

Le programme est traduit intégralement et une fois pour toute à travers toutes les couches logicielles.

On obtient un exécutable : programme sous forme d'instructions binaires, contenant aussi les bibliothèques et que l'OS envoie à la machine qui les interprète «électriquement» selon une stratégie pré-établie par le compilateur en fonction du processeur.

Interprétation ou compilation : telle est la question !

Trouver tous les nombres premiers inférieurs à N.

-> On utilise ici l'algorithme dit «Naïf»

N	C	Python
6.250	0.043 s	0.45 s
12.500	0.15 s	1.68 s
25.000	0.56 s	6.2 s
50.000	2.02 s	23.7 s
100.000	7.8 s	88.1 s
200.000	29.4 s	329.9 s
400.000	110.6 s	1245.3 s

On voit que C va 10 fois plus vite sur une double «boucle for»