Introduction à l'architecture des ordinateurs

Georges-André Silber
Centre de recherche en informatique
Mines de Paris

septembre 2009

Ordinateur, n.m. : machine automatique de traitement de l'information permettant de conserver, d'élaborer et de restituer des données sans intervention humaine en effectuant sous le contrôle de programmes enregistrés des opérations arithmétiques et logiques.

Nom "*ordinateur*", proposé par Jacques Perret (un latiniste de la Sorbonne) en 1956 à la demande d'IBM.

Cher Monsieur,

Que diriez-vous d'ordinateur ? C'est un mot correctement formé, qui se trouve même dans le Littré comme adjectif désignant Dieu qui met de l'ordre dans le monde. Un mot de ce genre a l'avantage de donner aisément un verbe ordiner, un nom d'action ordination. L'inconvénient est que ordination désigne une cérémonie religieuse ; mais les deux champs de signification (religion et comptabilité) sont si éloignés (...) que l'inconvénient est peut-être mineur. D'ailleurs, votre machine serait ordinateur (et non ordination) et ce mot est tout à fait sorti de l'usage théologique.
(...)
“Congesteur”, “digesteur”, évoquent trop “congestion” et “digestion”. “Synthétiseur” ne me paraît pas un mot assez neuf pour désigner un objet spécifique, déterminé, comme votre machine.

IBM 650

Un demi-million de dollars, faible encombrement.

12500 tours par minute, temps d'accès moyen 2,5 ms.
2000 mots de mémoire (5 chiffres base 10), @ de 0 à 1999.

"Loi de Hofstadter : cela prend toujours plus de temps que vous ne le pensez, même si vous tenez compte de la loi de Hofstadter."

— Douglas Hofstadter.

“Le travail s'étire jusqu'à remplir tout le temps disponible pour son accomplissement.”

— Cyril Northcote Parkinson.

“Le logiciel est comme un gaz, il se répand autant que le permet son support. Corollaire, le logiciel se développe jusqu'à ce qu'il soit limité par la loi de Moore. Il n'atteindra jamais un stade de maturité industrielle. L'industrie du logiciel est et restera toujours en état de crise.”

— Nathan Myhrvold (Microsoft).

"In fact, this is why there is a market for faster processors — software people have always consumed new capability as fast or faster than the chip people could make it available."

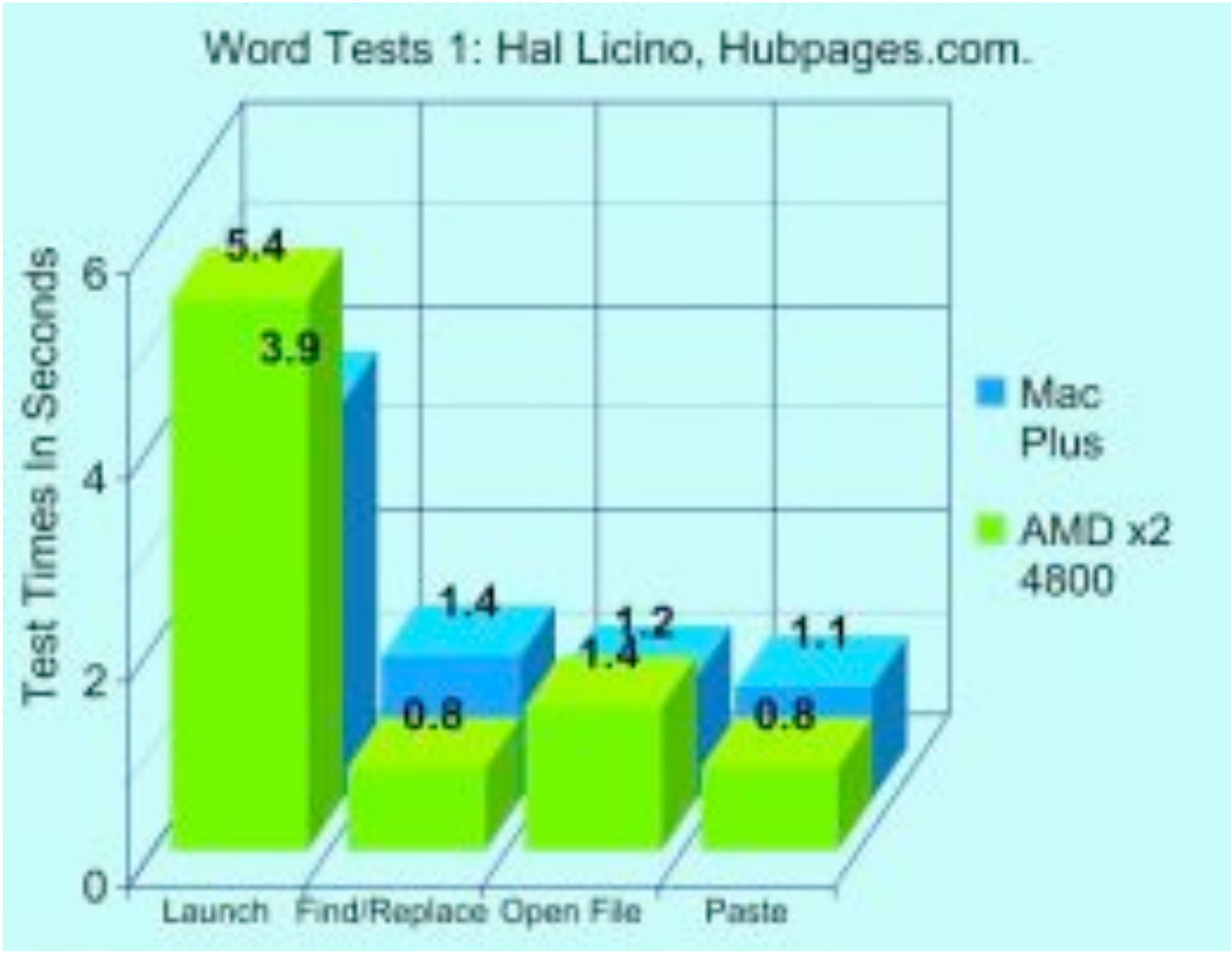
— Nathan Myhrvold (Microsoft).

“Si c’est digital, c’est que quelqu’un essaie de vous le vendre.”

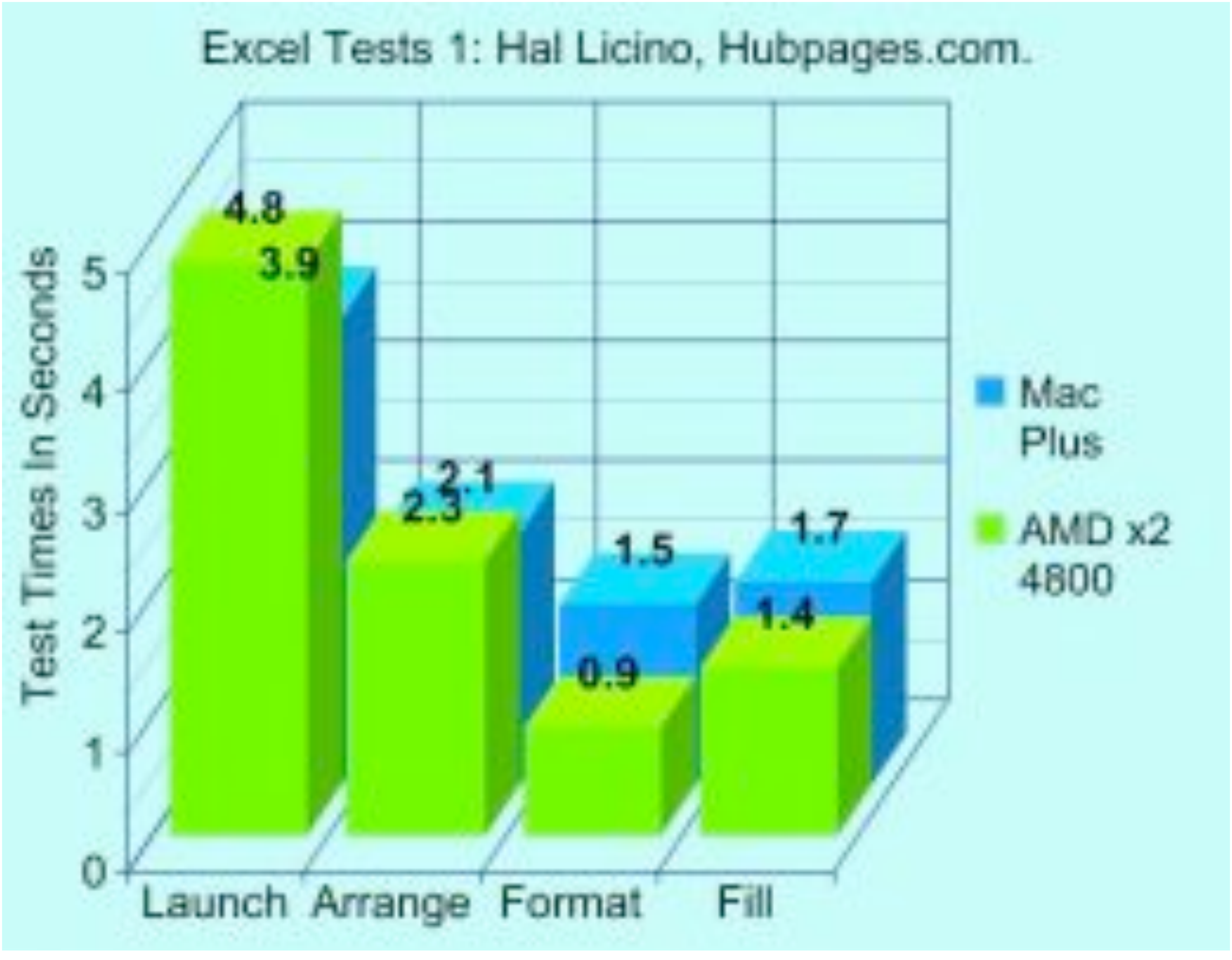
— Lincoln Spector.

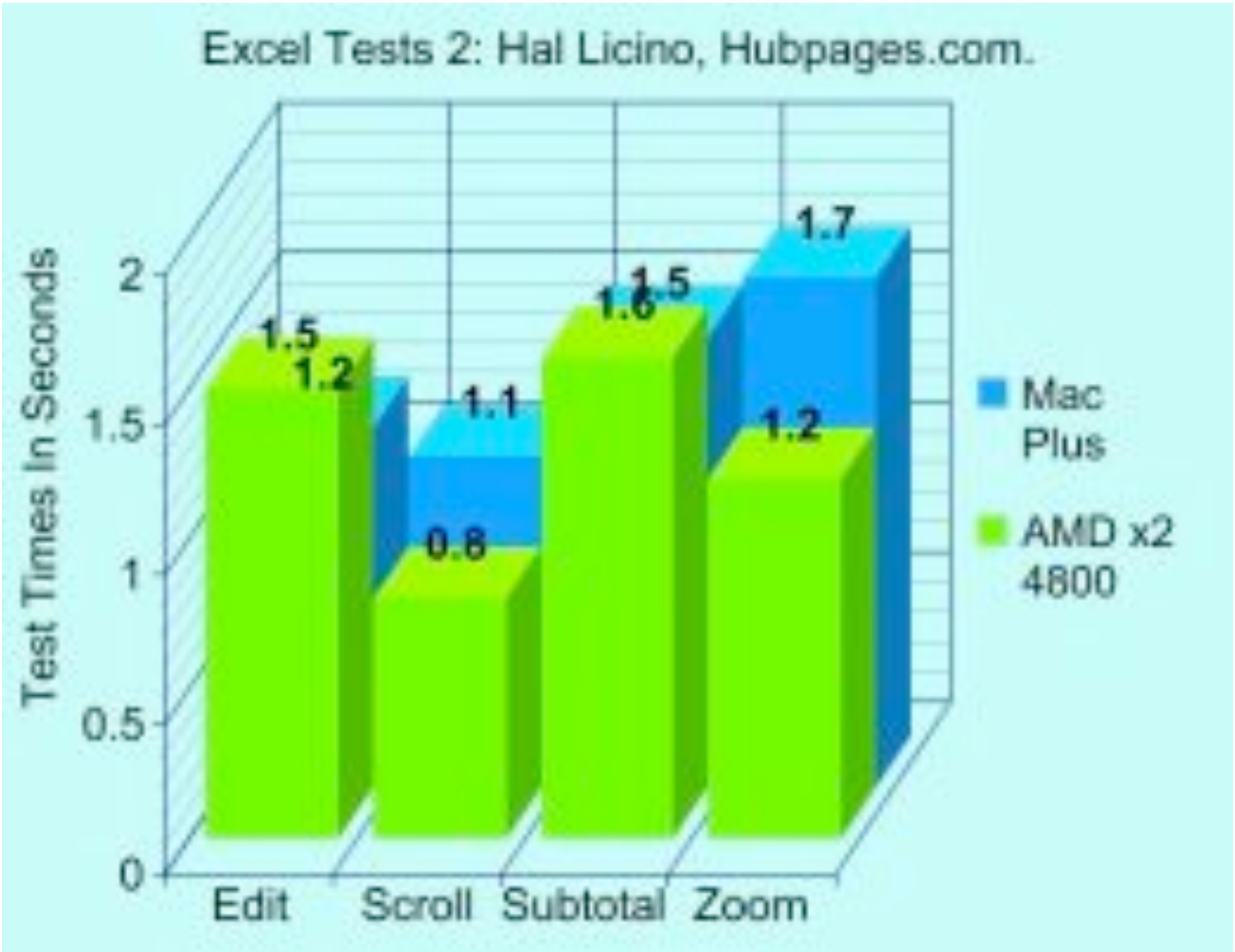
http://hubpages.com/hub/_86_Mac_Plus_Vs_07_AMD_DualCore_You_Wont_Believe_Who_Wins

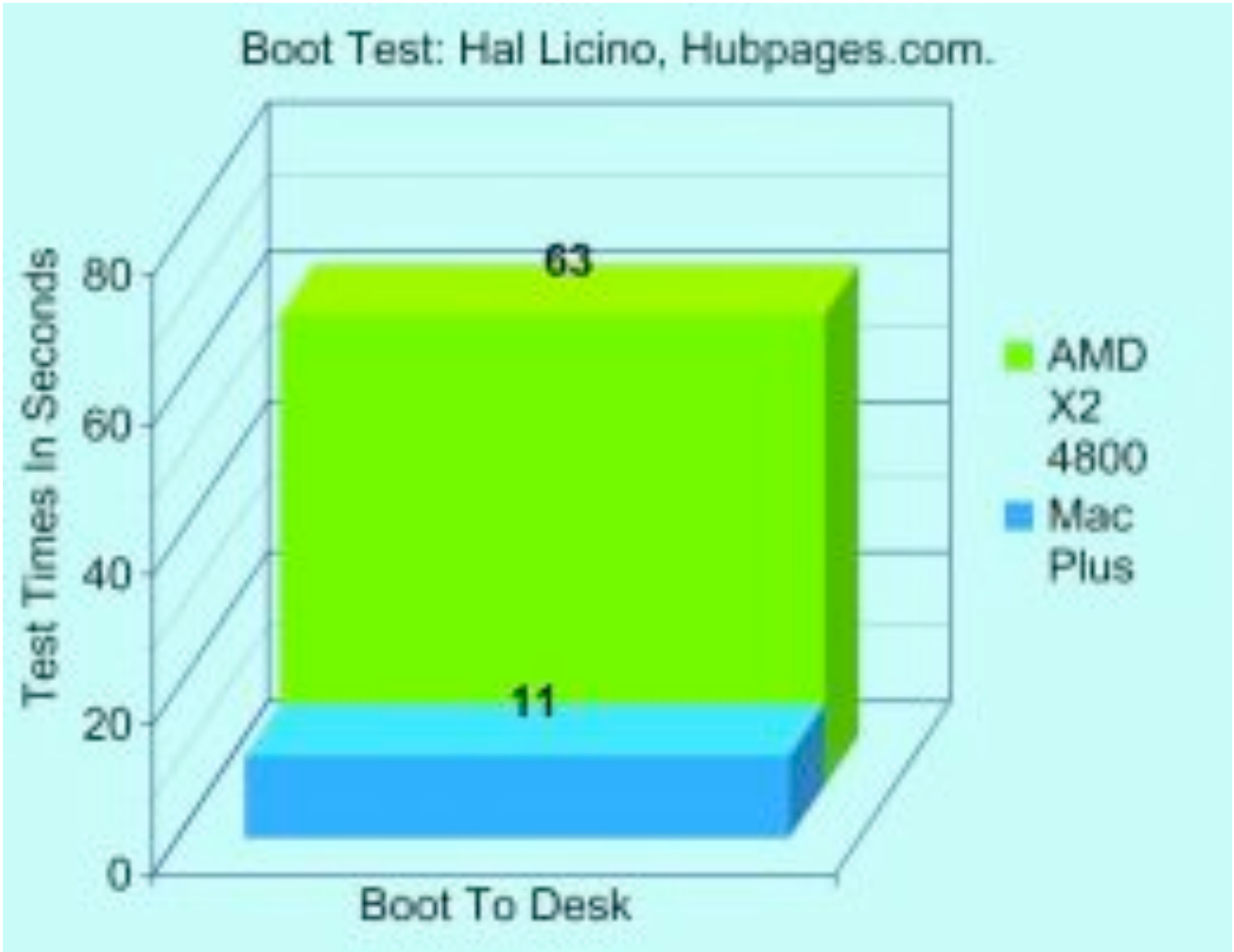
Mac Plus Motorola 68000 1986	Modern PC AMD Dual core 2007
8 MHz	2x2,4 GHz
4 Mo	1 Go
40 Mo DD	120 Go DD
—	2x1 Mo cache
60 W (+DD)	500 W + 40 W

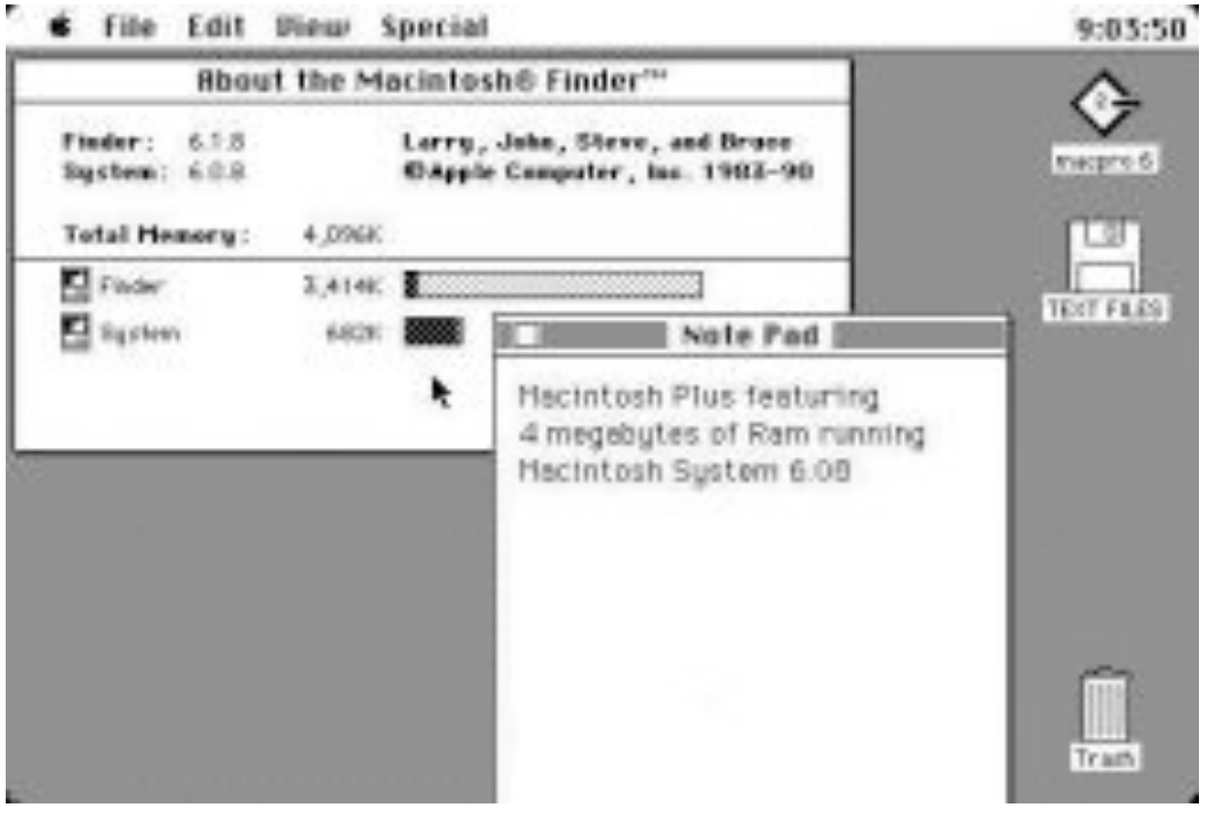

Word Tests 1: Hal Licino, Hubpages.com.
Test Times In Seconds
0
2
4
6
5.4
3.9
1.4
0.8
1.2
1.4
1.1
0.8
Launch
Find/Replace
Open File
Paste
Mac Plus
AMD x2 4800


Word Tests 2: Hal Licino, Hubpages.com.
Test Times In Seconds
1.25
1
0.75
0.5
0.25
0
1.2
0.9
1.1
0.9
1.1
0.8
0.2
0.1
Save
Scroll
Type Lag
Word Count
Mac Plus
AMD x2 4800


Excel Tests 1: Hal Licino, Hubpages.com.
Test Times In Seconds
0
1
2
3
4
5
4.8
3.9
2.3
2.1
1.5
0.9
1.7
1.4
Launch
Arrange
Format
Fill
Mac Plus
AMD x2 4800


Excel Tests 2: Hal Licino, Hubpages.com.
Test Times In Seconds
2
1.5
1
0.5
0
1.5
1.2
1.1
0.8
1.6
1.5
1.7
1.2
Edit
Scroll
Subtotal
Zoom
Mac Plus
AMD x2 4800


Boot Test: Hal Licino, Hubpages.com.
Test Times In Seconds
80
60
40
20
0
63
11
Boot To Desk
AMD X2 4800
Mac Plus


File Edit View Special
9:03:50
About the Macintosh® Finder™
Finder: 6.1.8
System: 6.0.8
Larry, John, Steve, and Bruce
©Apple Computer, Inc. 1983-90
Total Memory: 4,096K
Finder 3,414K
System 682K
Note Pad
Macintosh Plus featuring
4 megabytes of Ram running
Macintosh System 6.08
macpro 6
TEXT FILES
Trash

Sur les 17 tests, le vieil ordinateur gagne 53% du temps.

http://fr.wikipedia.org/wiki/Image:Gordon_Moore.jpg

Gordon Moore
Cofondateur d'Intel
Empereur de l'empirisme

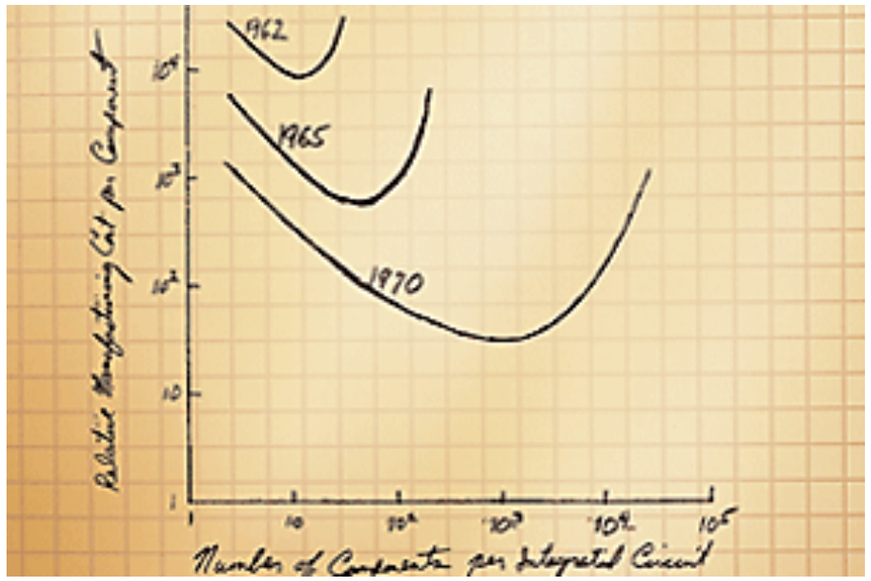
Les Lois de Moore

Première Loi de Moore

1965

“Cramming more components into integrated circuits” Electronics Magazine, 19 avril 1965

“The complexity for minimum component costs has increased at a rate of roughly a **factor of two per year** ... Certainly over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000. I believe that such a large circuit can be built on a single wafer.”

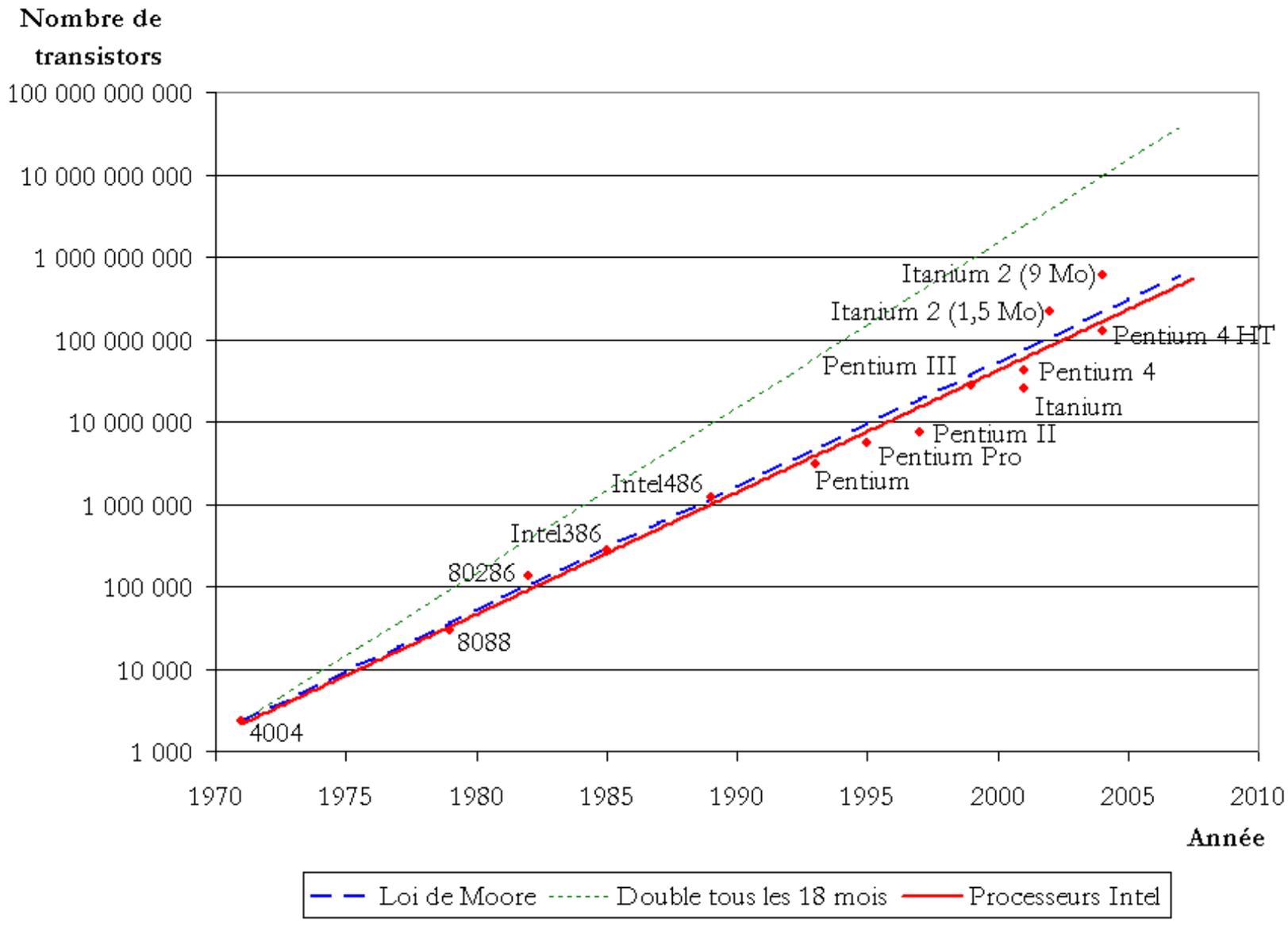


http://en.wikipedia.org/wiki/Moore's_law

Deuxième Loi de Moore

1975

Le nombre de transistors des **microprocesseurs** sur une puce de silicium double tous les deux ans.



http://en.wikipedia.org/wiki/Moore's_law

année	**processeur**	**largeur** (μm)	**transistors** (millions)	**fréquence** (MHz)	**tension** (V)	**cache** (Ko)
1971	4004	10	0.0023	0.108	12	0
1978	8086	3.0	0.029	5-10	5	0
1982	80286	1.5	0.134	6-12	5	0
1985	80386	1.5	0.275	16-33	5	0
1989	80468	1.0	1.2	25-50	5	0
1993	Pentium	0.8	3.1	60-66	5	0
1997	Pentium2	0.35	7.5	233-300	2.8	0
1999	Pentium3	0.18	28	500-733	1.65	256
2000	Pentium4	0.18	42	1400-2000	1.7	256
2002	Pentium4	0.13	55	2000-3000	1.5	512
2002	Itanium2	0.13	220	900-1000	-	256/1500
2003	Pentium4	0.09	>55	>3000	1.2	1024
2006	Core 2 duo	0.065	291	2930	1.3	2x2048

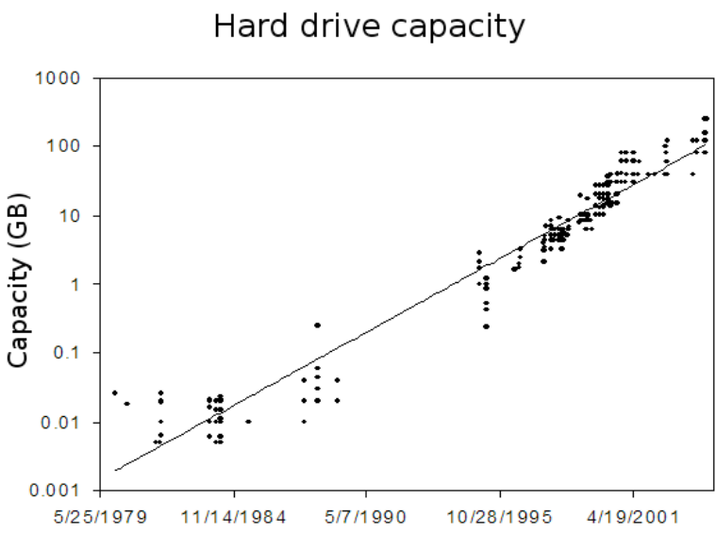
Barrières technologiques

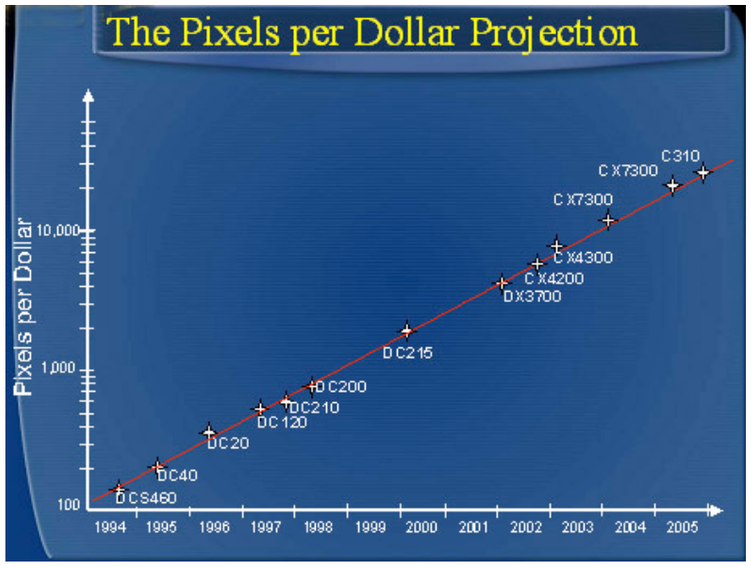
- Depuis 2004, problèmes de dissipation thermique
- Plusieurs processeurs par puce
- Asynchronisme (transmission du signal d'horloge très coûteuse en espace et en énergie)
- The Wall : 20 nm (photolitographie)
- Quantique, ADN, nano-informatique ?

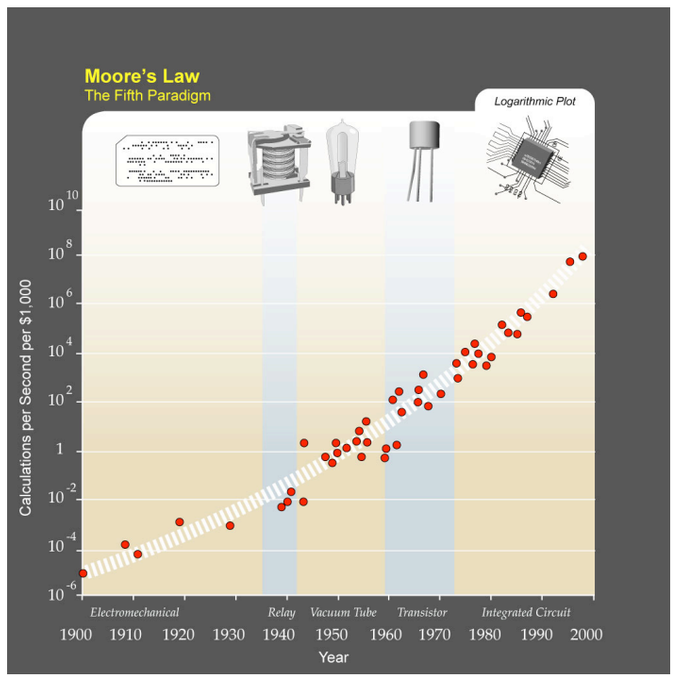
Autres Lois de Moore

L'exponentiel semble être la règle pour l'industrie numérique, c'est un moteur.

Prophéties auto-réalisatrices ?







http://en.wikipedia.org/wiki/Moore's_law

Le monde de demain ?

"*Si l'industrie automobile avait avancé aussi vite que l'industrie des semi-conducteurs, une Rolls Royce pourrait rouler plusieurs millions de km avec un litre d'essence et il serait moins coûteux de la jeter que de la garer.*"

— Gordon Moore.

Analogie avec le transport aérien ?

- 1978 : Paris - New York, 900 €, 7h
- 2003 : Paris - New York, 1 centime, 1/4 de seconde

Le logiciel

La technologie logicielle a une importance cruciale.

Les avancées scientifiques et technologiques au niveau du logiciel sont beaucoup plus lentes et ne suivent pas la loi de Moore.

Loi de Wirth:
«*Le logiciel ralentit plus vite que le matériel accélère.*»

Obésiciels

- Office 2000 : 293 Mo
- Office 2007 : 3 Go (*10 fois plus de fonctionnalités ?)*
- Windows Vista nécessite 10 fois plus de mémoire vive et 3 fois plus de puissance processeur pour fonctionner que Windows XP

Exemple : écrire un programme pour **trier** un million d'entiers ($n = 1000000$).

Ordinateur A

1 milliard d'instructions par seconde,
algorithme du tri par insertion,
codé en $2n^2$ instructions en assembleur.

Ordinateur B

10 millions d'instructions par seconde,
algorithme du tri par fusion,
codé en $50n\lg n$ instructions dans un langage de haut niveau.

- Ordinateur A

$$\frac{2.(10^6)^2 \text{ instructions}}{10^9 \text{ instructions/seconde}} = 2000 \text{ secondes}$$

- Ordinateur B

$$\frac{50.10^6 \lg 10^6 \text{ instructions}}{10^7 \text{ instructions/seconde}} \approx 100 \text{ secondes}$$

- Le choix du **langage** et la performance des **compilateurs** jouent un rôle crucial dans la **vitesse** d'un programme et son **occupation mémoire**.
- Exemple : somme des 10000 premiers entiers sur un Pentium 4 à 2.4 GHz.
- En **Java** : 0.1 seconde, 3.3 Mo de mémoire occupée.
- En **Python** : 0.03 seconde, 2.2 Mo de mémoire occupée.
- En **C** : inférieur à 0.001 seconde, 300 Ko de mémoire occupée.

```
class somme {
  public static void main(String args[]) {
    int i, somme;
    i = 0;
    somme = 0;
    for (i = 0; i <= 10000; i++)
      somme = somme + i;
    System.out.println(somme);
  }
}
```

JAVA

```
i = 0
somme = 0
while i <= 10000:
    somme = somme + i
    i = i + 1
print somme
```

PYTHON

```c
int main() {
   int i;
   int somme;
   i = 0;
   somme = 0;
   for (i = 0; i <= 10000; i++)
     somme = somme + i;
   printf("%d", somme);
}
```

Et le **compilateur** ?

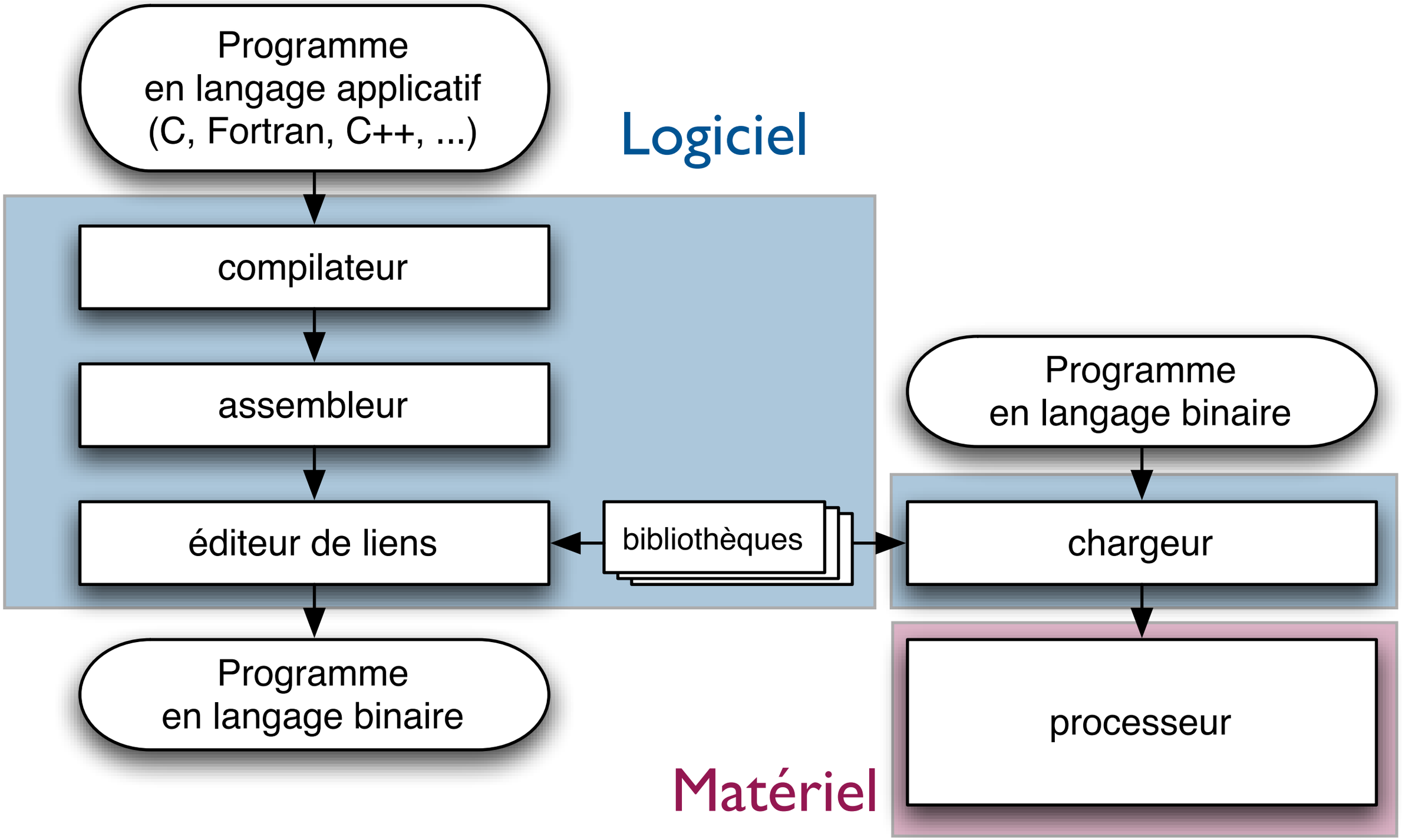
Le **type** de compilateur utilisé et les **options** utilisées lors de la compilation jouent également un rôle important dans la vitesse d'exécution des programmes.

Le programme C "idiot" suivant, compilé avec ou sans les optimisations activées dans le compilateur, donne un programme dont le temps d'exécution est de :

- 17 secondes **sans** les optimisations activées,
- 7 secondes **avec**.

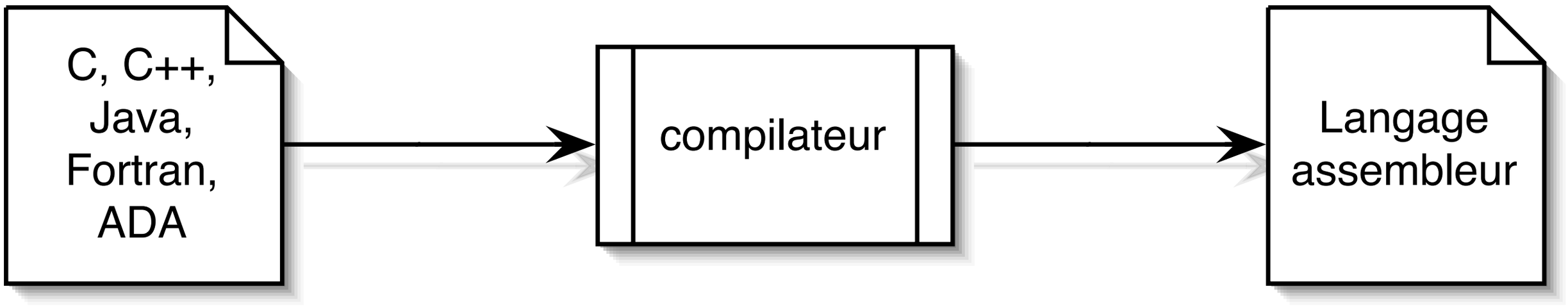
```
int main() {
   long i;
   long somme;
   i = 0;
   somme = 0;
   for (i = 0; i <= 1000000000L; i++)
      somme = (somme + i)%34;
   printf("%d", somme);
}
```

Compilation / Exécution



Compilation

- Les programmes écrits dans un langage applicatif (C, C++, Java, Fortran, ADA, etc.) sont traduits en langage d'assemblage par un compilateur
- Le langage d'assemblage est propre à un processeur donné
- Habituellement un compilateur produit un seul type de langage d'assemblage




```c
int main() {
  int i;
  int somme;
  i = 0;
  somme = 0;
  for (i = 0; i <= 100; i++)
    somme = somme + i;
  printf("%d", somme);
}
```

Langage C

```
main:
        pushl   %ebp
        movl    %esp, %ebp
        subl    $24, %esp
        [...]
.L2:
        cmpl    $100, -4(%ebp)
        jle     .L5
        jmp     .L3
.L5:
        [...]
        addl    %eax, (%edx)
        leal    -4(%ebp), %eax
        incl    (%eax)
        jmp     .L2
.L3:
        movl    -8(%ebp), %eax
        movl    %eax, 4(%esp)
        call    printf
```

Assembleur x86

Assemblage

- Le programme en langage d'assemblage (ou langage assembleur) est traduit en binaire par un programme appelé assembleur
- Le binaire est une suite de mots de taille fixe ou variable, chaque mot étant une instruction du processeur
- Une instruction se compose d'un code opération (opcode) et d'opérandes (constantes ou numéros de registre)

Exemple: PowerPC

- Une instruction PowerPC: 32 bits
- Un ‘opcode’ sur 5 bits (32 instructions)

Table 2. PowerPC instruction formats

Format	0	6	11	16	21	26	30	31
D-form	opcd	tgt/src	src/tgt	immediate				
X-form	opcd	tgt/src	src/tgt	src	extended opcd			
A-form	opcd	tgt/src	src/tgt	src	src	extended opcd		Rc
BD-form	opcd	BO	BI	BD			AA	LK
I-form	opcd	LI					AA	LK

Instructions PPC

```
li r0,0
```

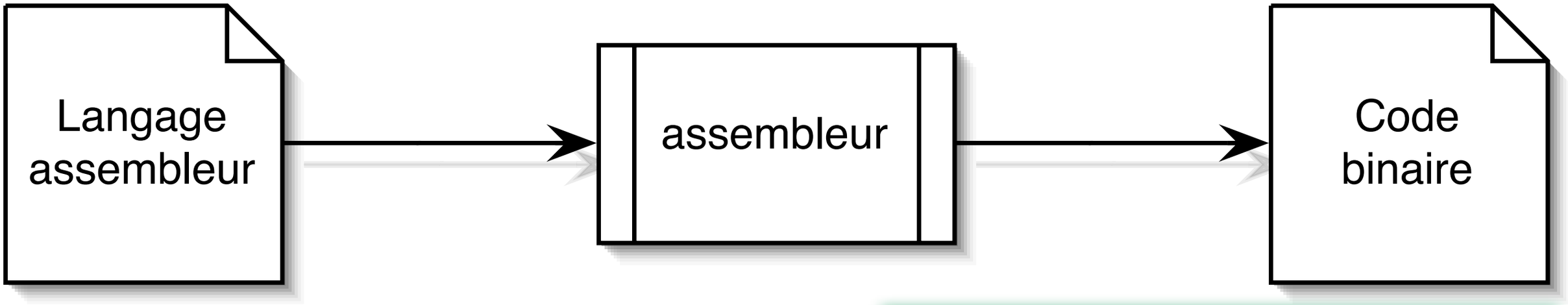
Charger (load) l'entier 0 dans le registre r0

```
add r0,r2,r0
```

Ajouter (add) le contenu de r0 à celui de r2 et stocker le résultat dans r0

```
stw r0,56(r30)
```

Stocker (store) le contenu de r0 en RAM à l'adresse contenue dans r30+56




```
main:
        pushl   %ebp
        movl    %esp, %ebp
        subl    $24, %esp
        [...]
.L2:
        cmpl    $100, -4(%ebp)
        jle     .L5
        jmp     .L3
.L5:
        [...]
        addl    %eax, (%edx)
        leal    -4(%ebp), %eax
        incl    (%eax)
        jmp     .L2
.L3:
        movl    -8(%ebp), %eax
        movl    %eax, 4(%esp)
        call    printf
```

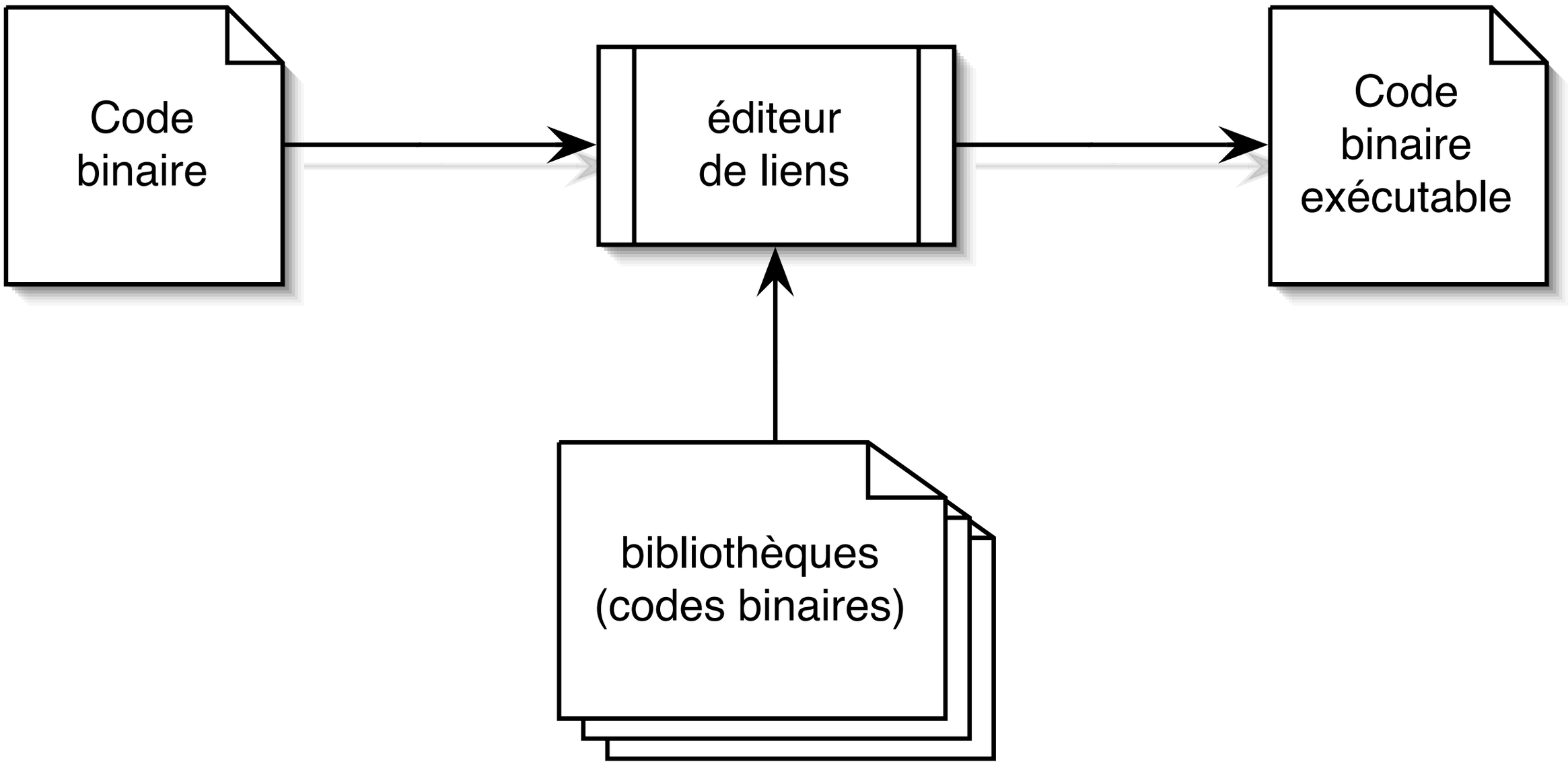
Assembleur x86

```
0000000 feed face 0000 0012 0000 0000 0000 0001
0000020 0000 0003 0000 01b0 0000 2000 0000 0001
0000040 0000 0148 0000 0000 0000 0000 0000 0000
0000060 0000 0000 0000 0000 00c8 0000 01cc
0000100 0000 00c8 0000 0007 0000 0007 0000 0004
0000120 0000 0000 5f5f 7465 7874 0000 0000 0000
0000140 0000 0000 5f5f 5445 5854 0000 0000 0000
0000160 0000 0000 0000 0000 0084 0000 01cc
0000200 0000 0002 0000 0294 0000 0005 8000 0400
0000220 0000 0000 0000 0000 5f5f 7069 6373 796d
0000240 626f 6c73 7475 6231 5f5f 5445 5854 0000
0000260 0000 0000 0000 0000 00a0 0000 0020
0000300 0000 026c 0000 0005 0000 02bc 0000 0004
0000320 8000 0408 0000 0000 0000 0020 5f5f 6373
0000340 7472 696e 6700 0000 0000 0000 5f5f 5445
0000360 5854 0000 0000 0000 0000 0000 0000 00c0
0000400 0000 0003 0000 028c 0000 0002 0000 0000
0000420 0000 0000 0000 0002 0000 0000 0000 0000
0000600 0000 0050 0000 0000 0000 0000 0000 0000
0000620 0000 0001 0000 0001 0000 0002 0000 0000
```

Binaire x86 (hexadécimal)

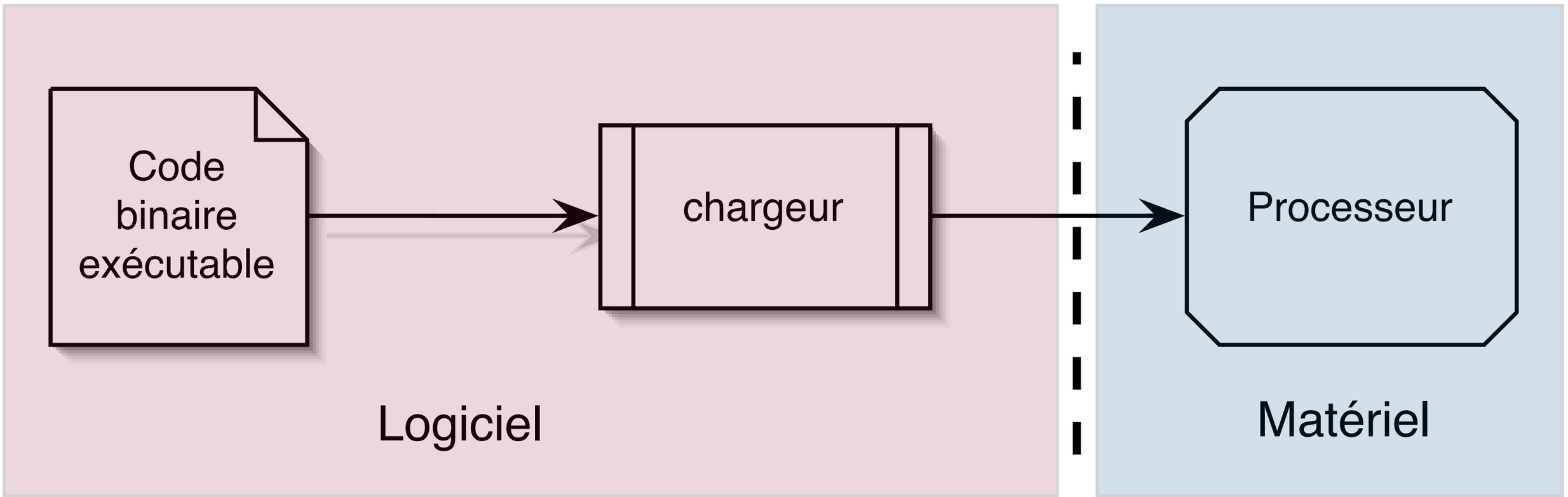
Edition de liens

- Le code binaire doit être composé avec d'autres code binaires déjà compilés/ assemblés (bibliothèques)
- Exemple: affichage sur écran
- Cette phase s'appelle l'édition de liens et est effectuée par un programme appelé l'éditeur de liens

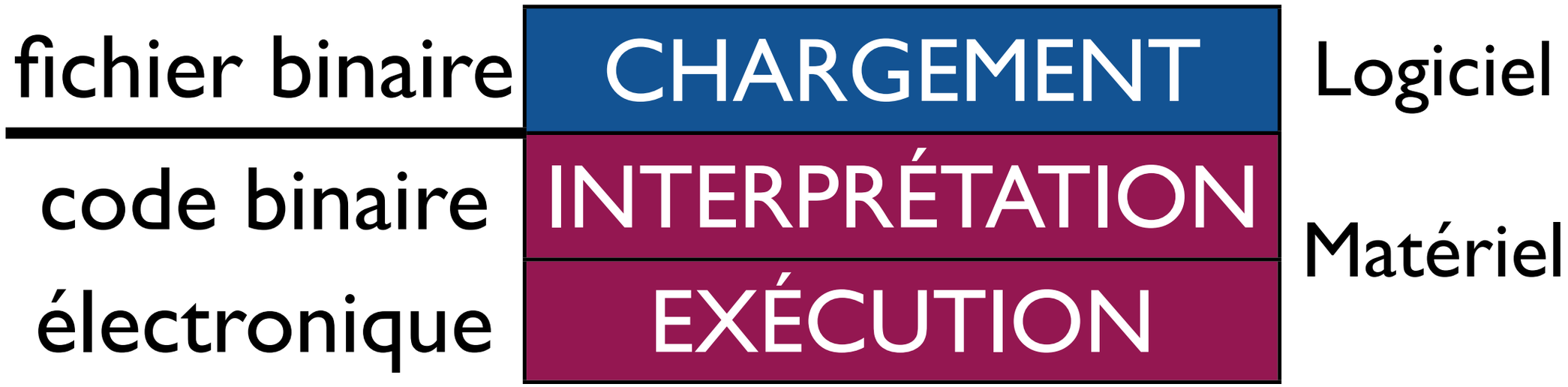

Code
binaire
éditeur
de liens
Code
binaire
exécutable
bibliothèques
(codes binaires)

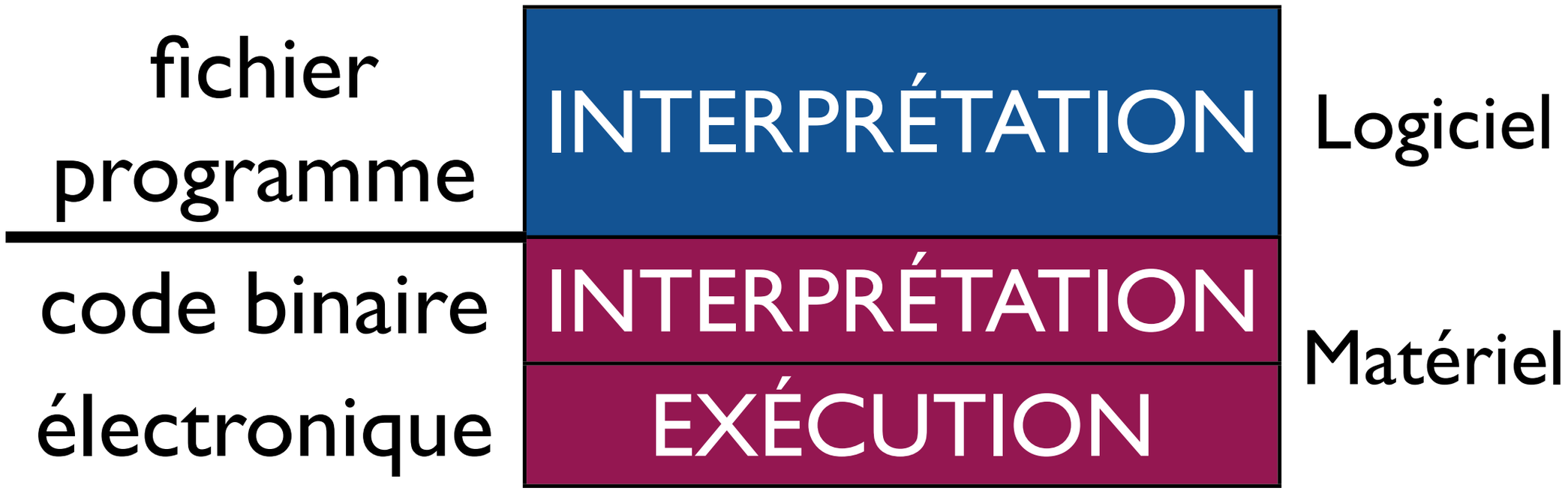
Exécution

- Le code binaire exécutable doit être transféré sur le processeur
- Cet aspect est pris en charge par un programme appelé le chargeur et qui est intégré au système d'exploitation
- un programme en cours d'exécution sur le processeur est appelé un processus


Code
binaire
exécutable
chargeur
Processeur
Logiciel
Matériel

Virtualisation





Problème du “bootstrap”

- Le compilateur est lui-même un programme écrit dans un langage applicatif qui a été compilé par un autre compilateur.
- Comment démarrer ?

Problème de la poule et de l’oeuf

L’inventeur du “bootstrap”
Le baron de Münchhausen

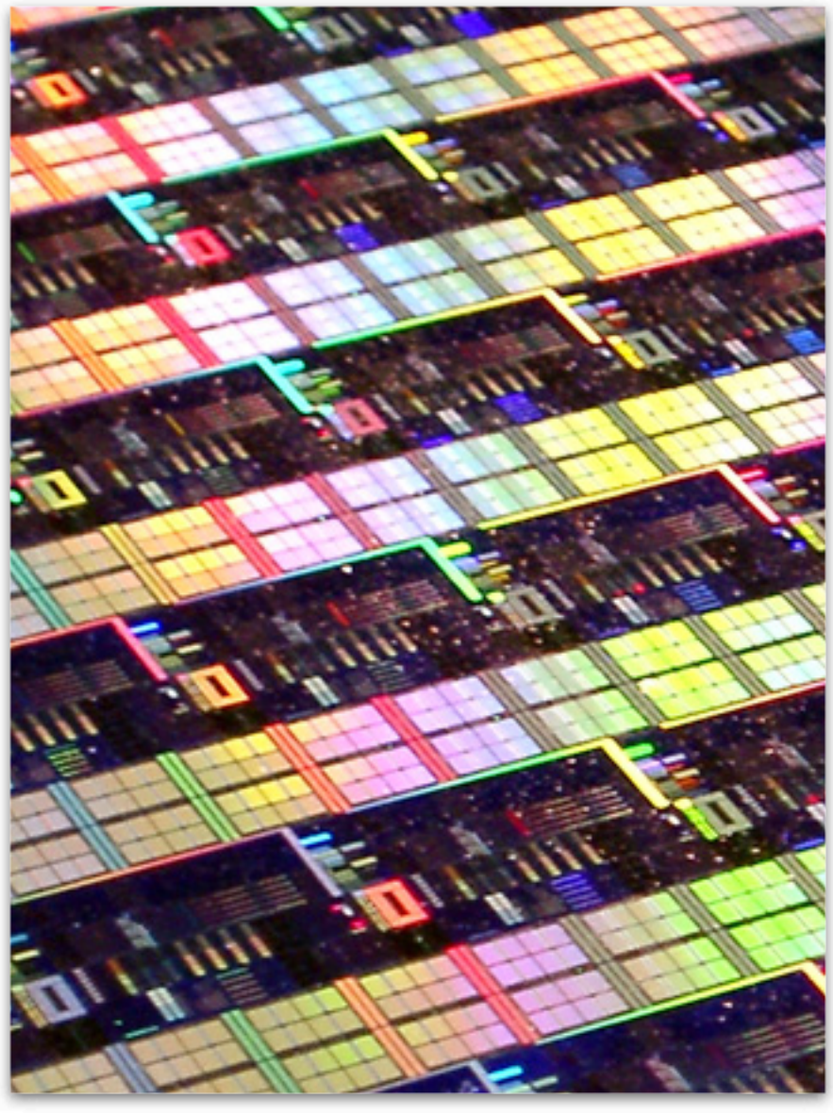
Modes d'exécution

- Trois grands modes d'exécution pour les programmes
  - Langages compilés (C, Fortran, C++, ...)
  - Langages interprétés (Python, Perl, PHP, ...)
  - Langages avec machine virtuelle (Java, C#, ...)

Couche Matérielle

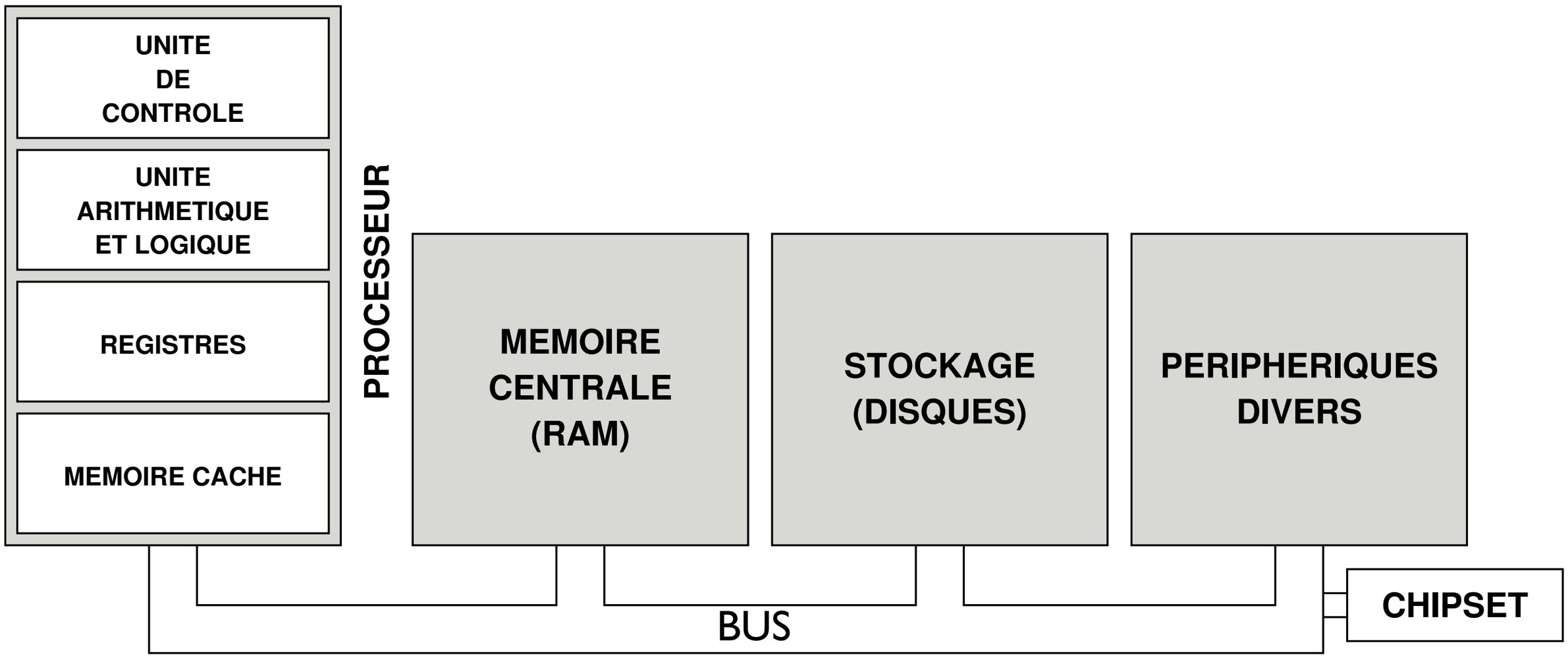
Couche matérielle

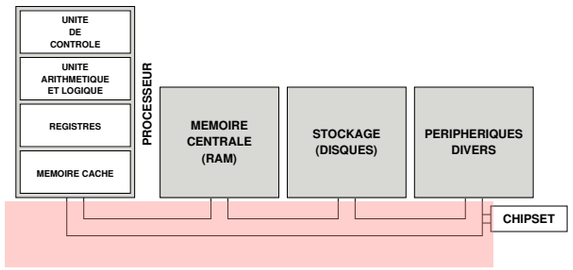
- Compilation/exécution
- Vue globale d'un ordinateur
- Chemin des données
- Couche physique
- Mémoire



circuit intégré

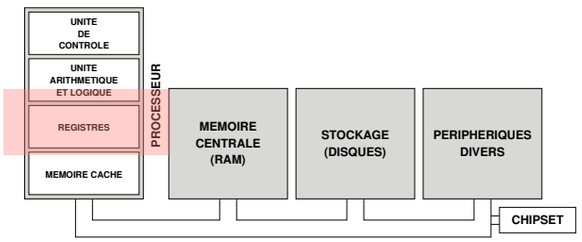
Vue globale d'un ordinateur





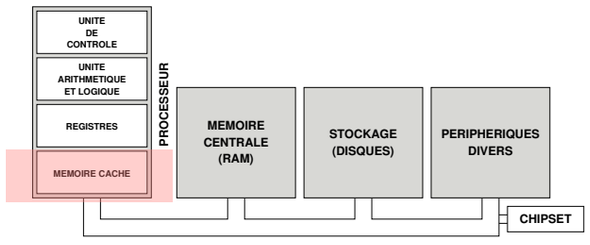
BUS

- Ensemble de fils électriques
- Signaux d'adresse, de données, de commande
- Interconnexion de plusieurs périphériques avec le même ensemble de fils (≠ point à point)
- Connexions parallèle (PCI) ou série (USB)
- Les bus sont partout dans un ordinateur



Registres

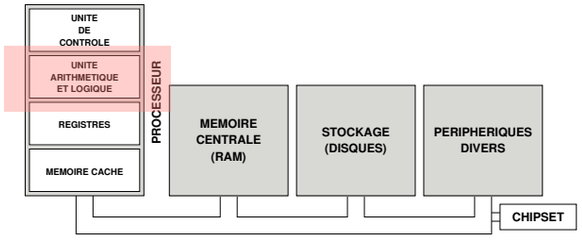
- Cases mémoire internes au processeur
  - de petite taille (32, 64, 128 bits)
  - très rapides (technologie SRAM)
- Registres contiennent
  - des données
  - des instructions (dont RI, le registre d'instruction)
  - des adresses (dont CO, le compteur ordinal)



Mémoire cache

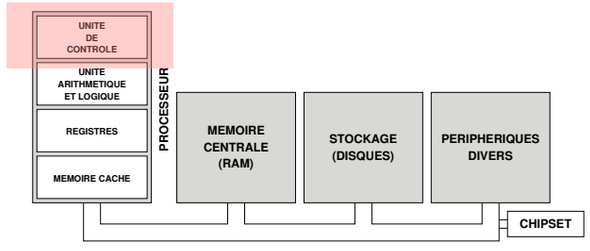
Antémémoire

- Mémoire de petite taille intégrée au processeur
- Entre 512 Ko et 8 Mo
- Très rapide (technologie SRAM)
- Toutes les données traitées par le processeur passent par le cache
- Amélioration des performances (voir plus loin)



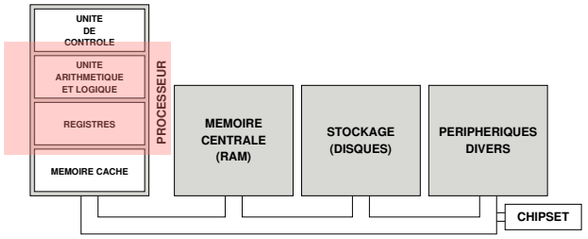
Unité Arithmétique et Logique

- La machine à calculer de l'ordinateur
- Instructions arithmétiques
  - Entiers
  - Nombres à virgule flottante
- Instructions logiques sur les bits
- Instructions de comparaison



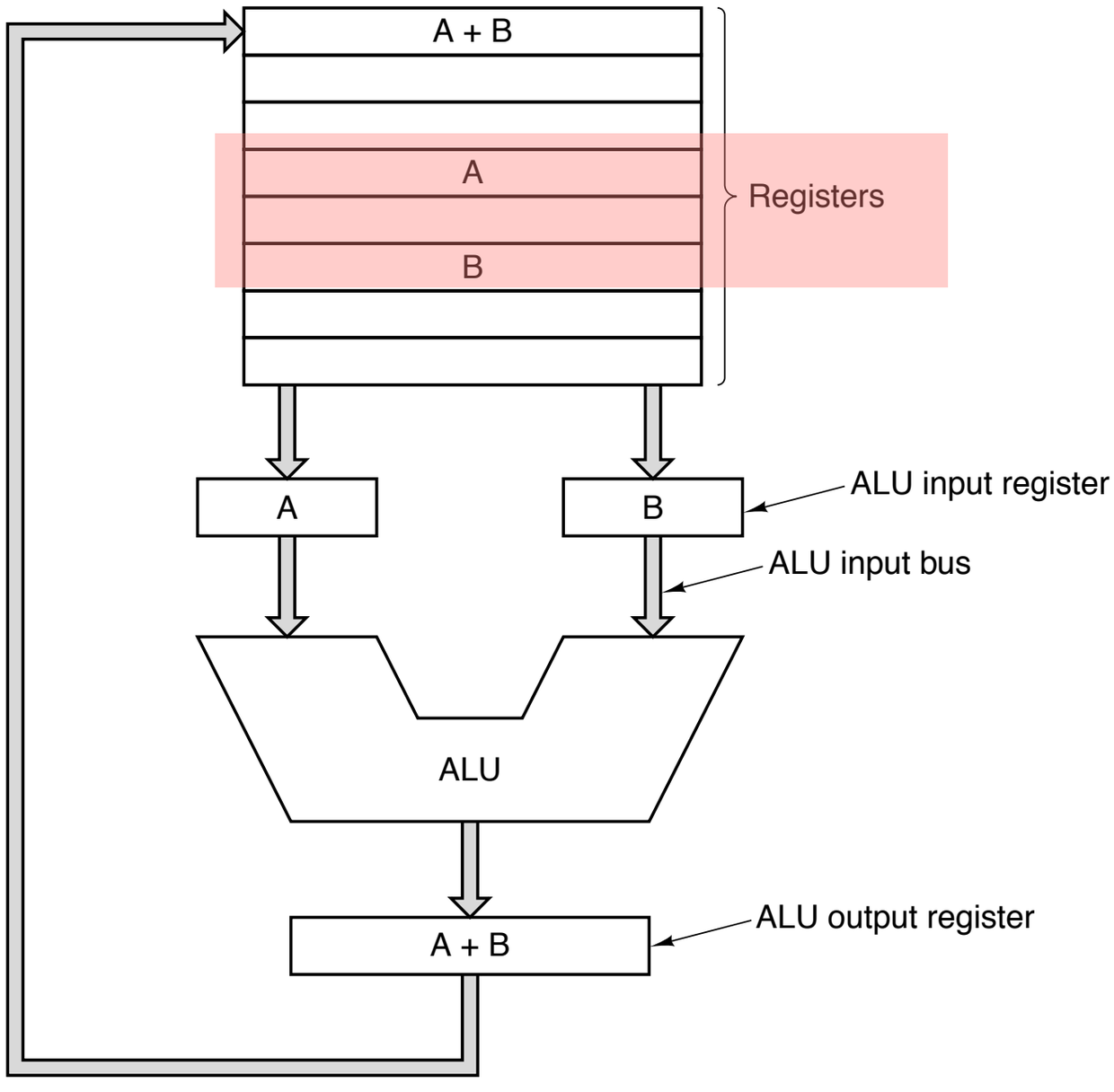
Unité de contrôle

- *chargement* des instructions
- *décodage* des instructions
- Instructions du processeur (3 familles)
  I. Transferts registres / mémoire
  II. Branchements
  III. Opérations arithmétiques et logiques sur des registres

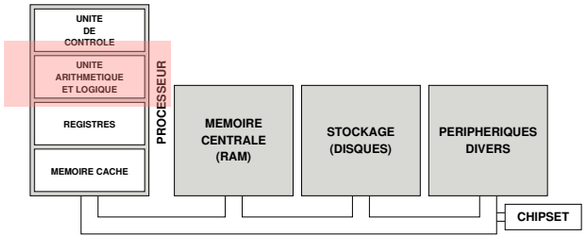


ALU and datapath

UAL et chemin des données

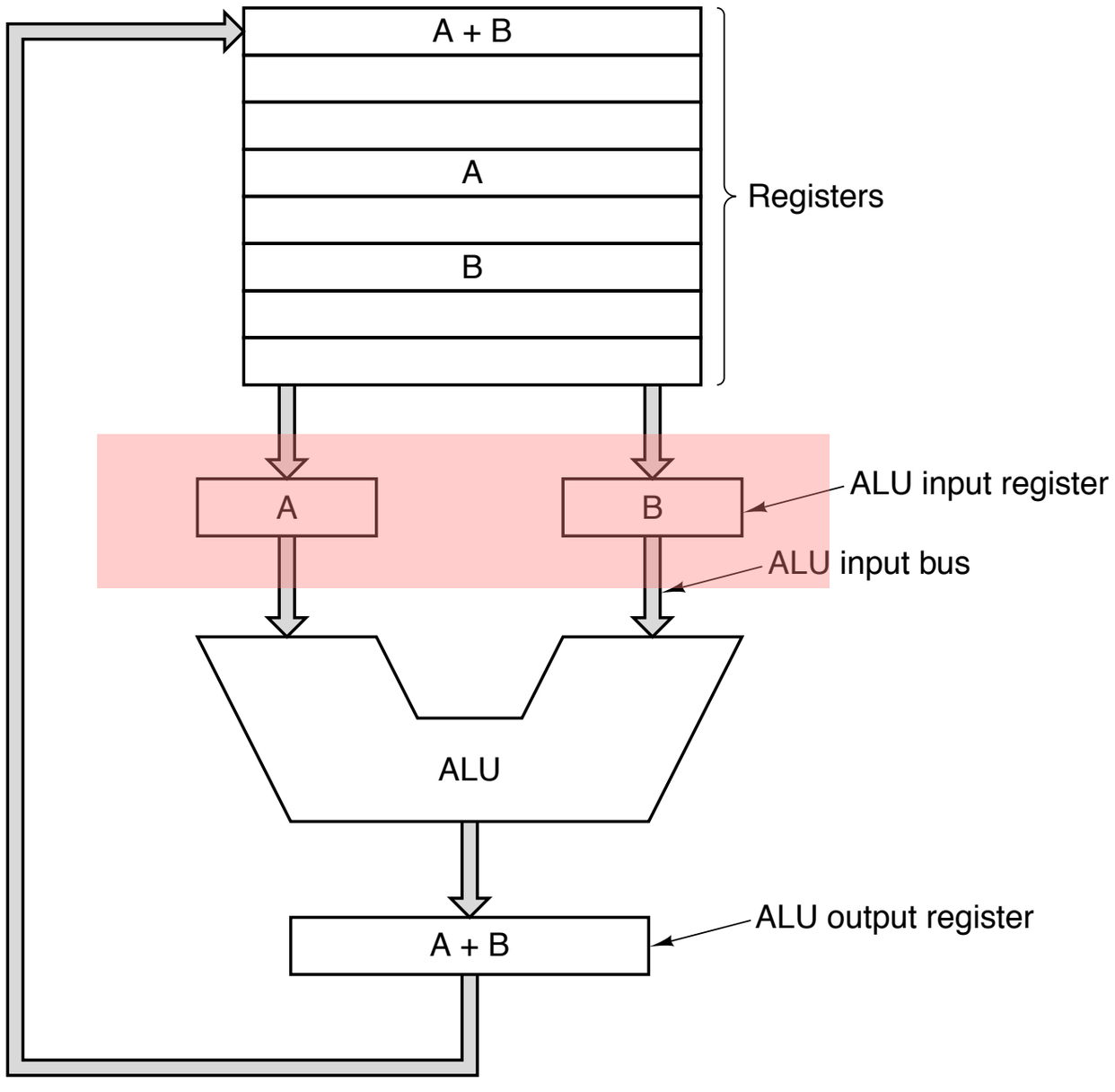


- Exemple
  - calcul de A+B
- Au départ, A et B sont dans des registres

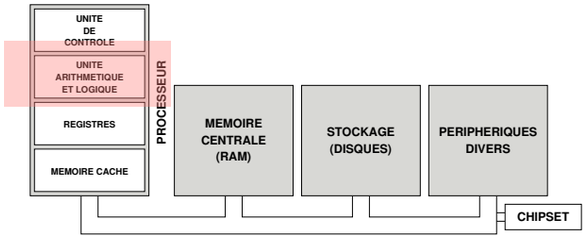


ALU and datapath

UAL et chemin des données

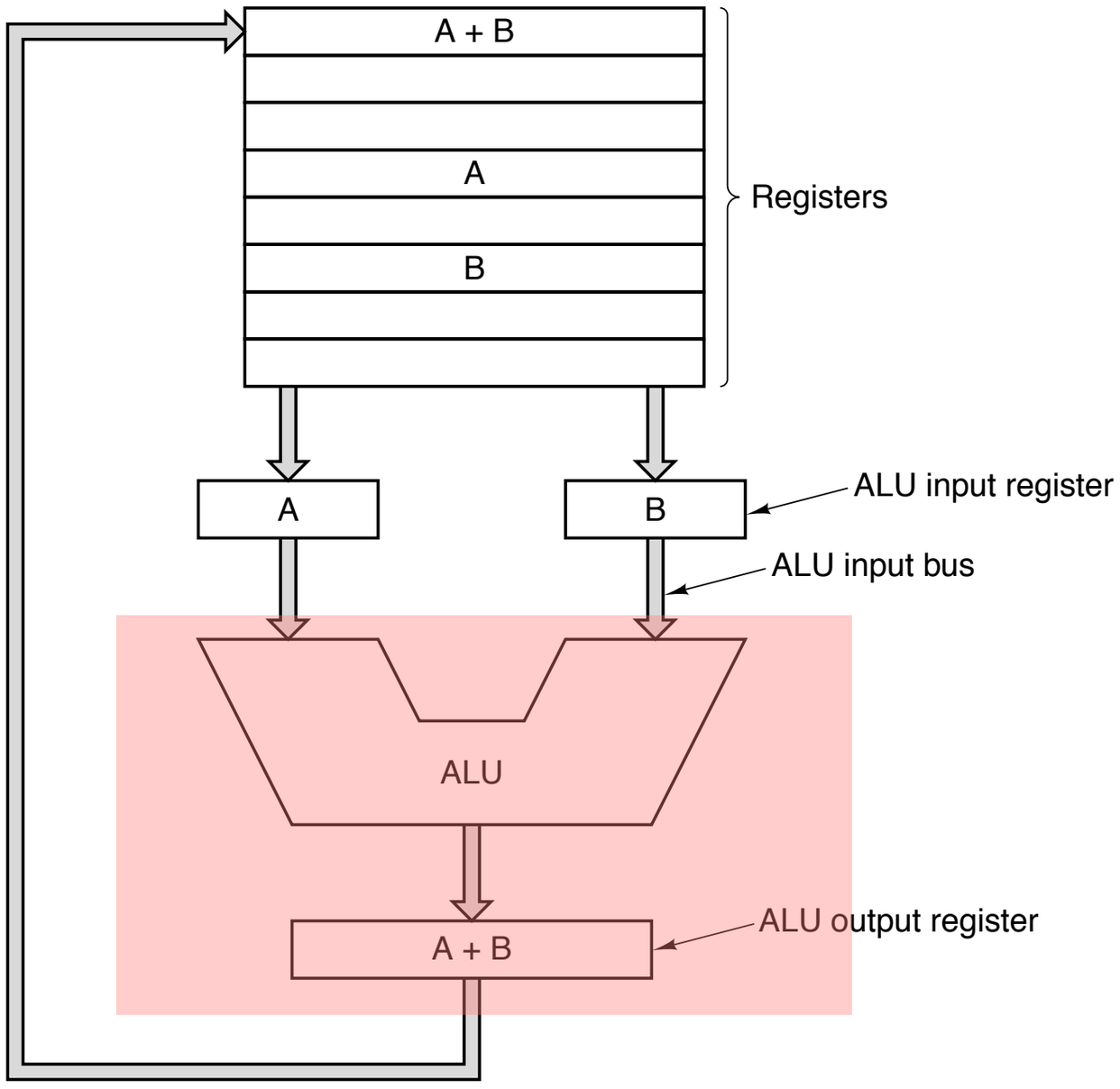


- A et B sont déplacés dans des registres d'entrée de l'UAL

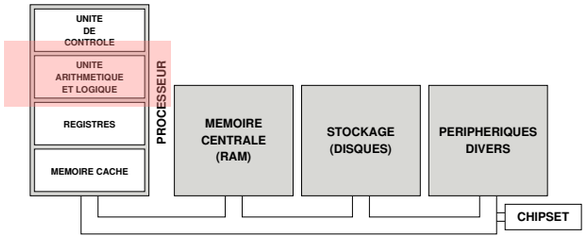


ALU and datapath

UAL et chemin des données

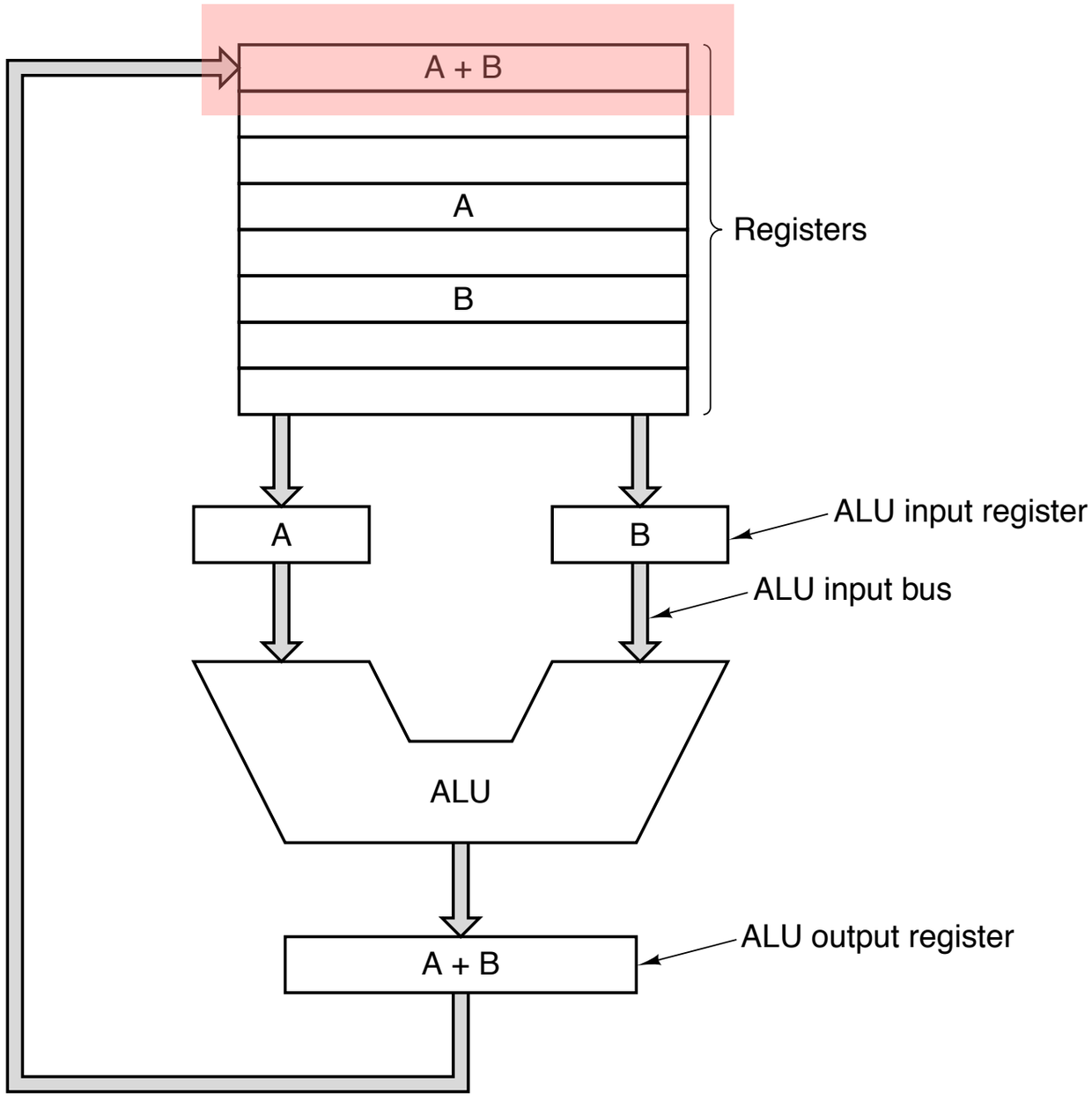


- Calcul de A+B par l'UAL
- Stockage du résultat dans le registre de sortie de l'UAL

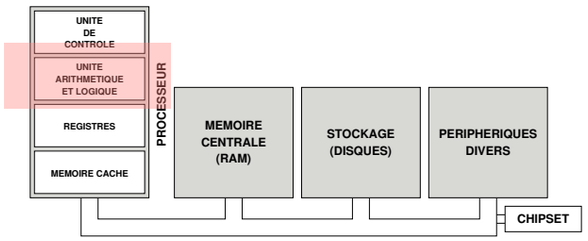


ALU and datapath

UAL et chemin des données

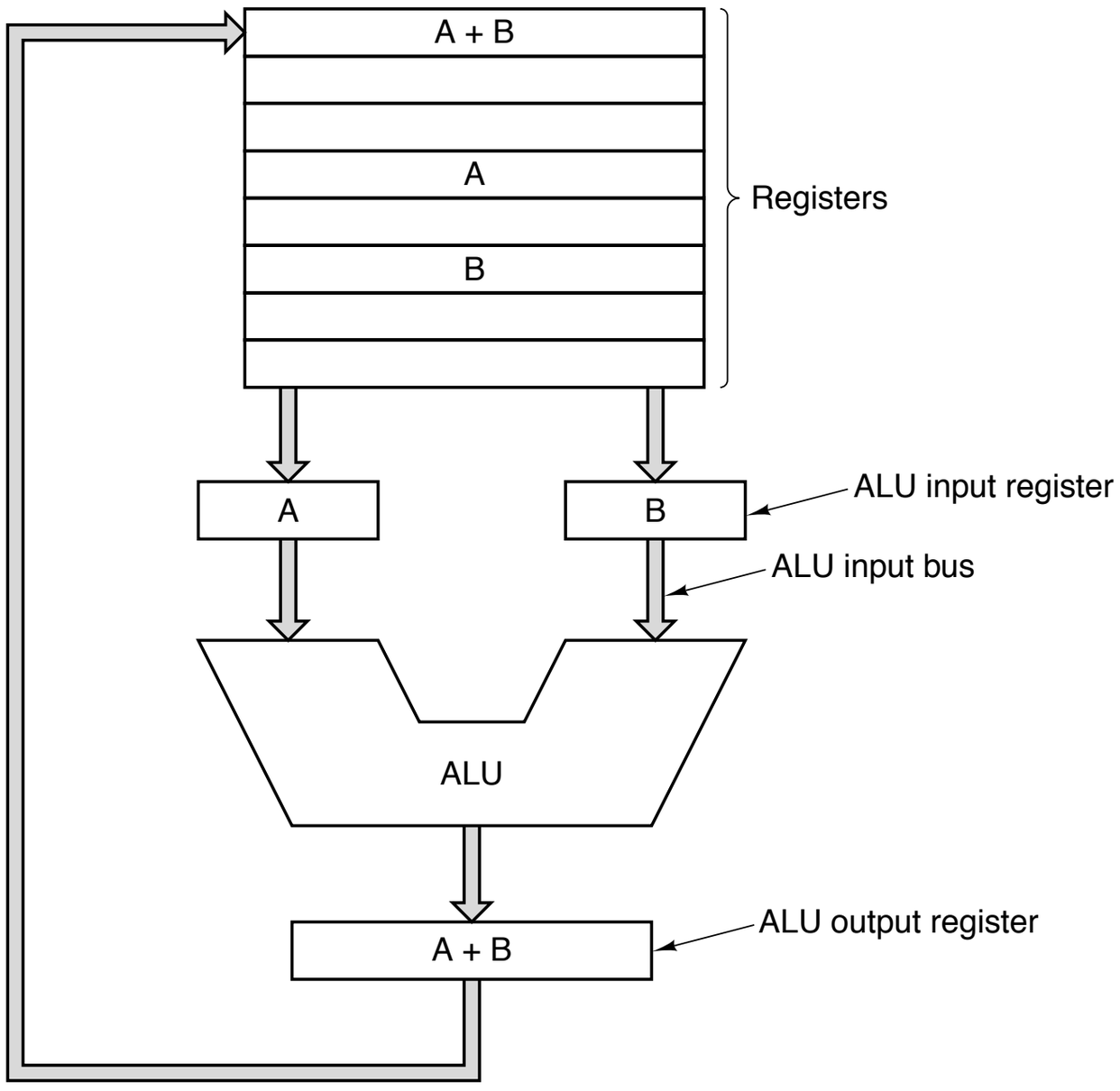


- Stockage du résultat dans un registre

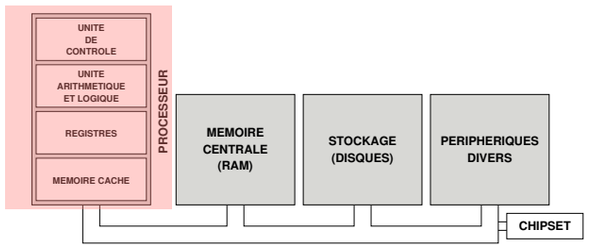


ALU and datapath

UAL et chemin des données



- Le temps que mettent les données à faire un cycle donne une indication fondamentale sur la vitesse du processeur



Exécution d'une Instruction

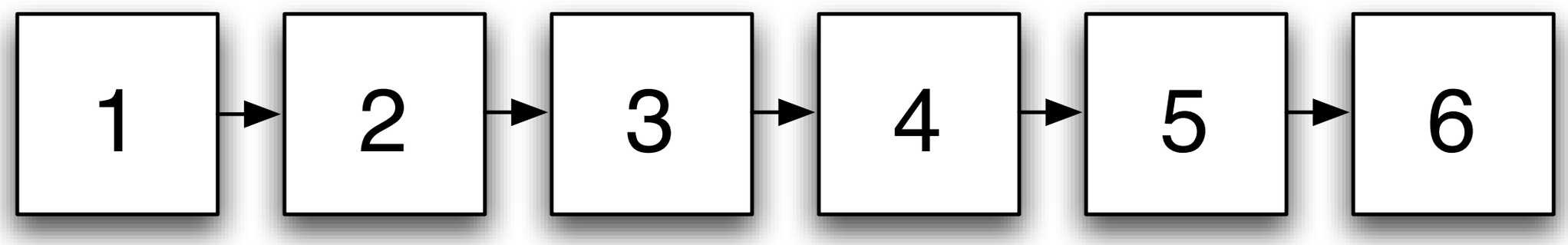
1. Charger la prochaine instruction se trouvant à l'adresse contenue dans CO dans le registre RI
2. Décoder l'instruction dans RI
3. Charger les opérandes de l'instruction
4. Exécuter l'instruction (UAL ou directe)
5. Ecrire le résultat
6. Modifier CO et retourner en 1.

Datapath

Cycle FETCH → DECODE → EXECUTE

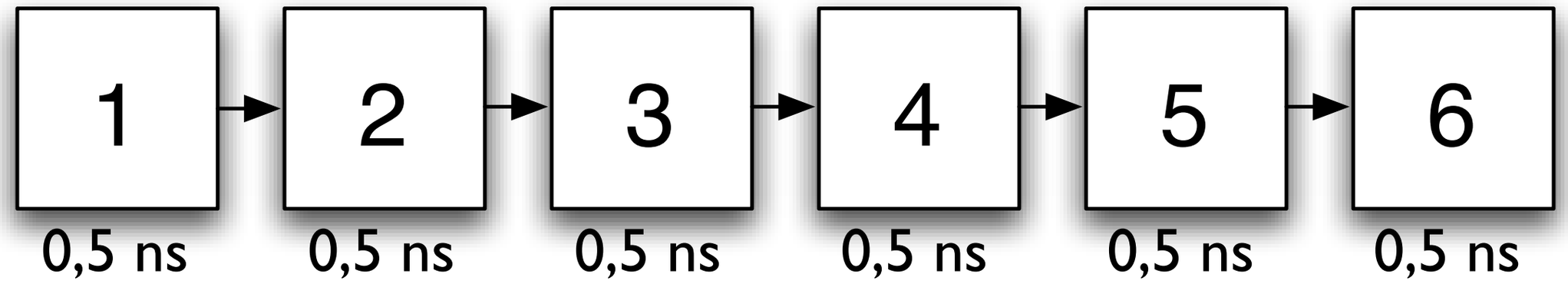
Pipeline

- Chaque étape de l'algorithme précédent est indépendante
- Chaque étape peut-être confiée à une partie distincte du processeur
- Travail à la chaîne



Pipeline (2)

Cycle d'horloge de 0,5 ns (2 GHz)

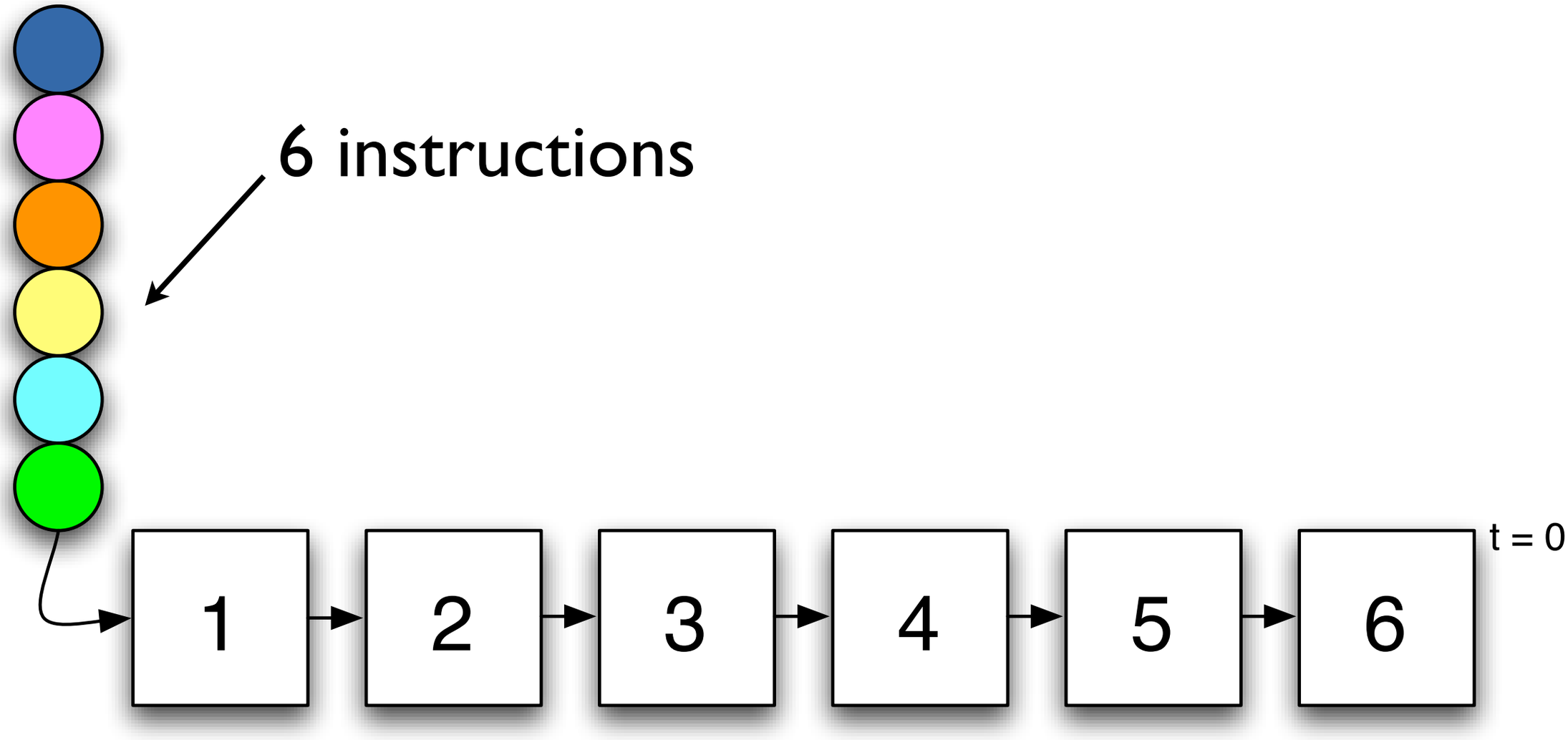


3 ns par instruction

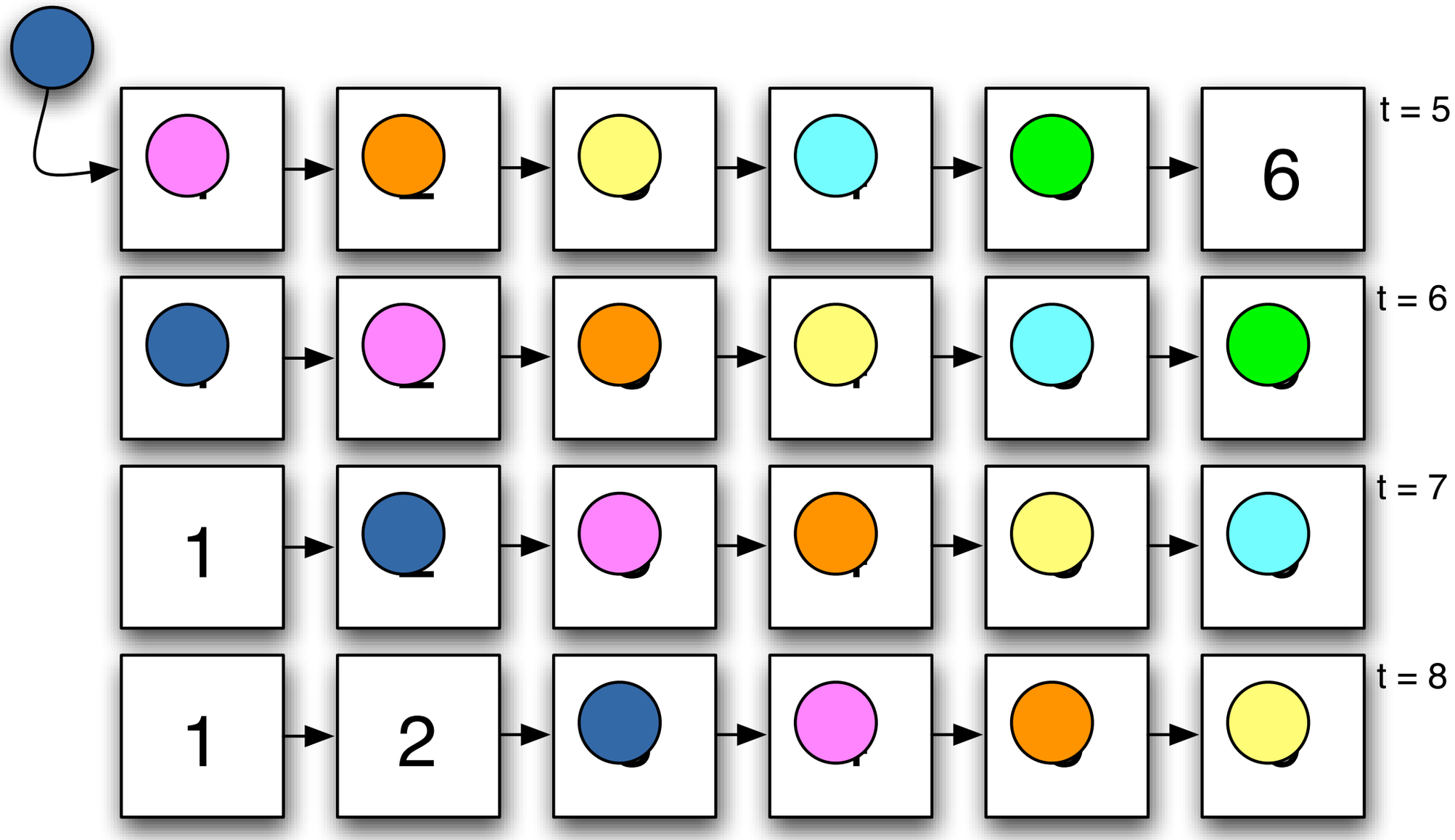
333 millions d'instructions par seconde (MIPS)

Pipeline (3)

En fait, on peut faire mieux...



Pipeline (4)

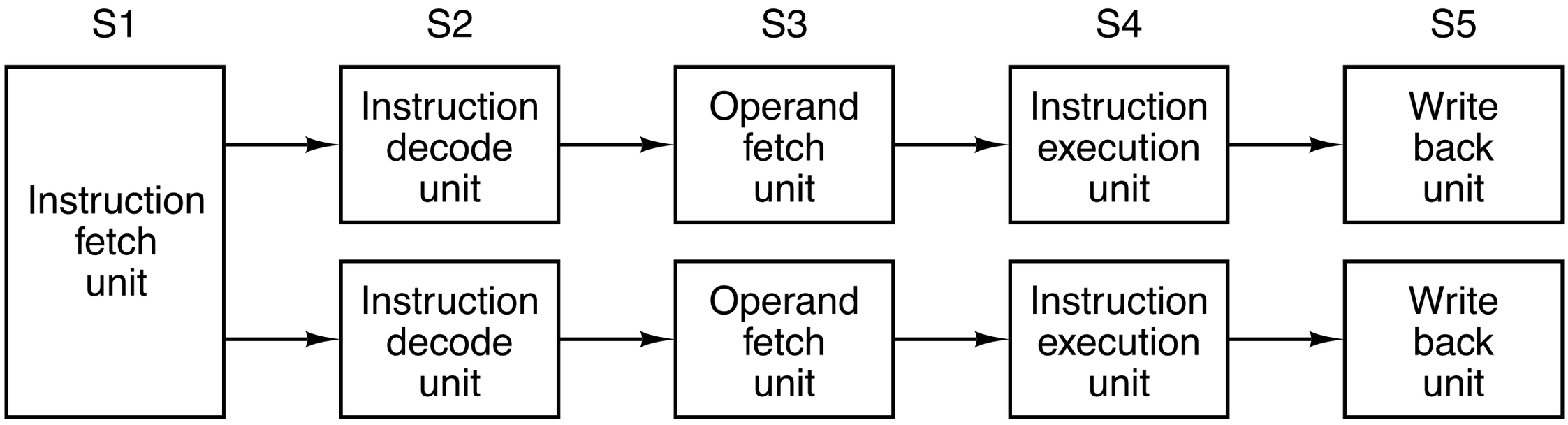


Première instruction : 6 étapes, 3 ns (latence)

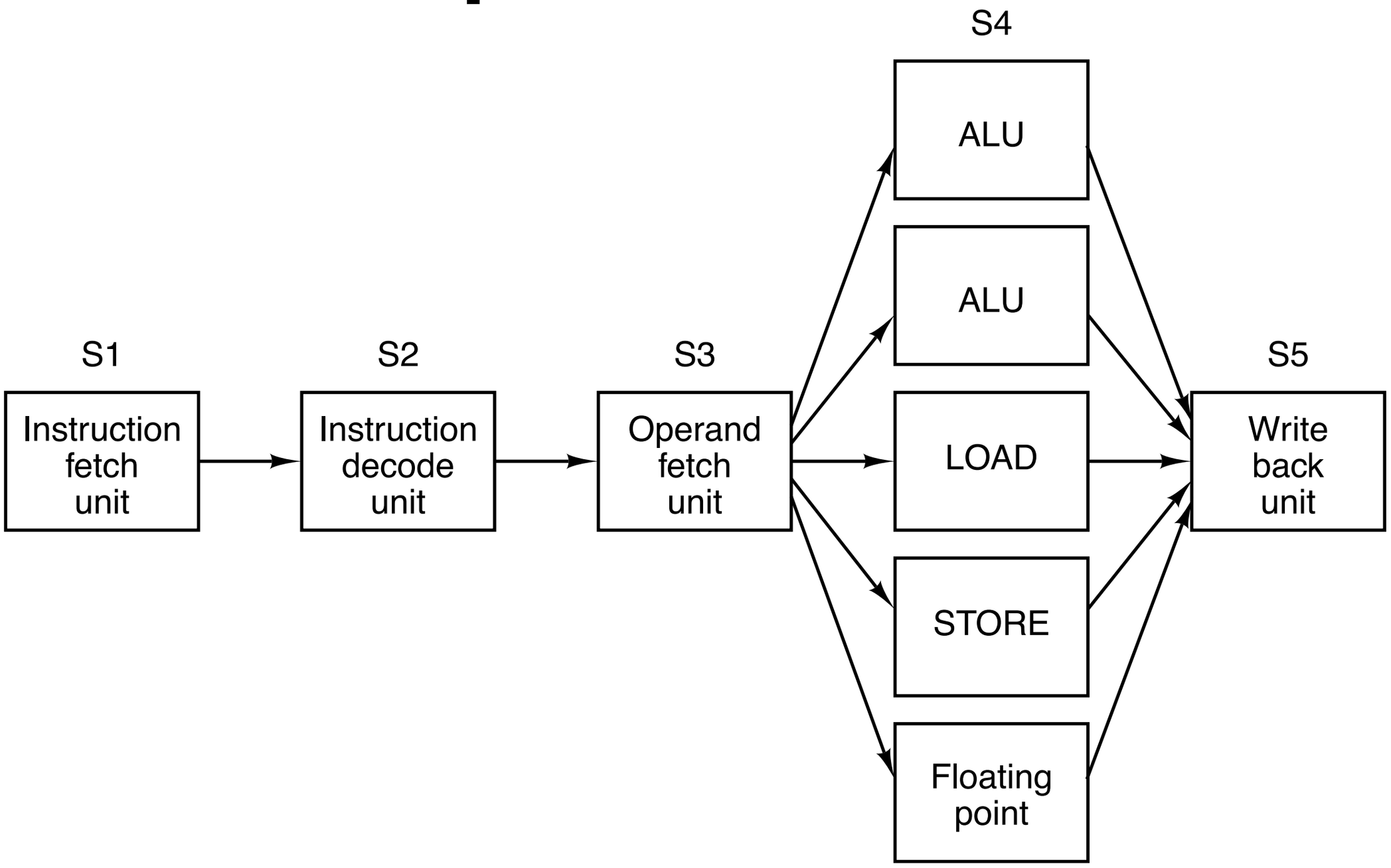
Ensuite, une instruction se termine à chaque cycle

Débit : 2000 MIPS (bande passante)

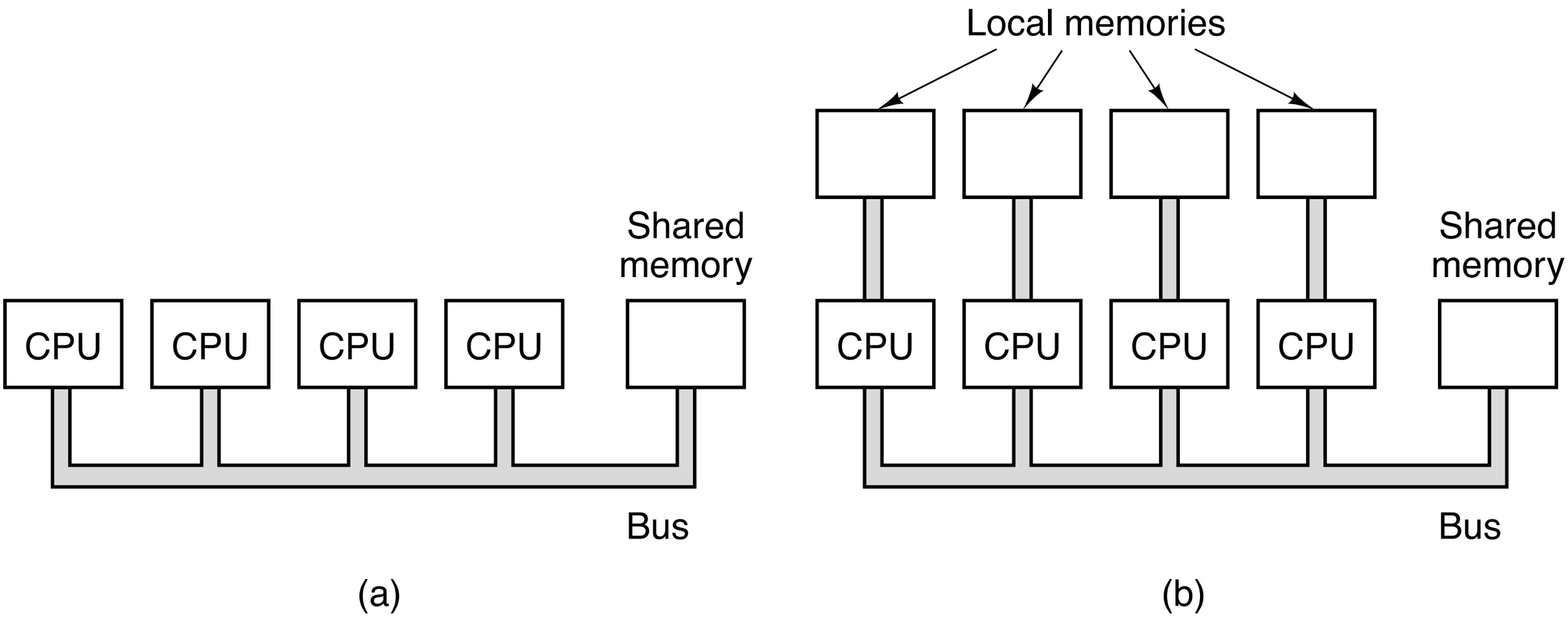
Multi-pipelines



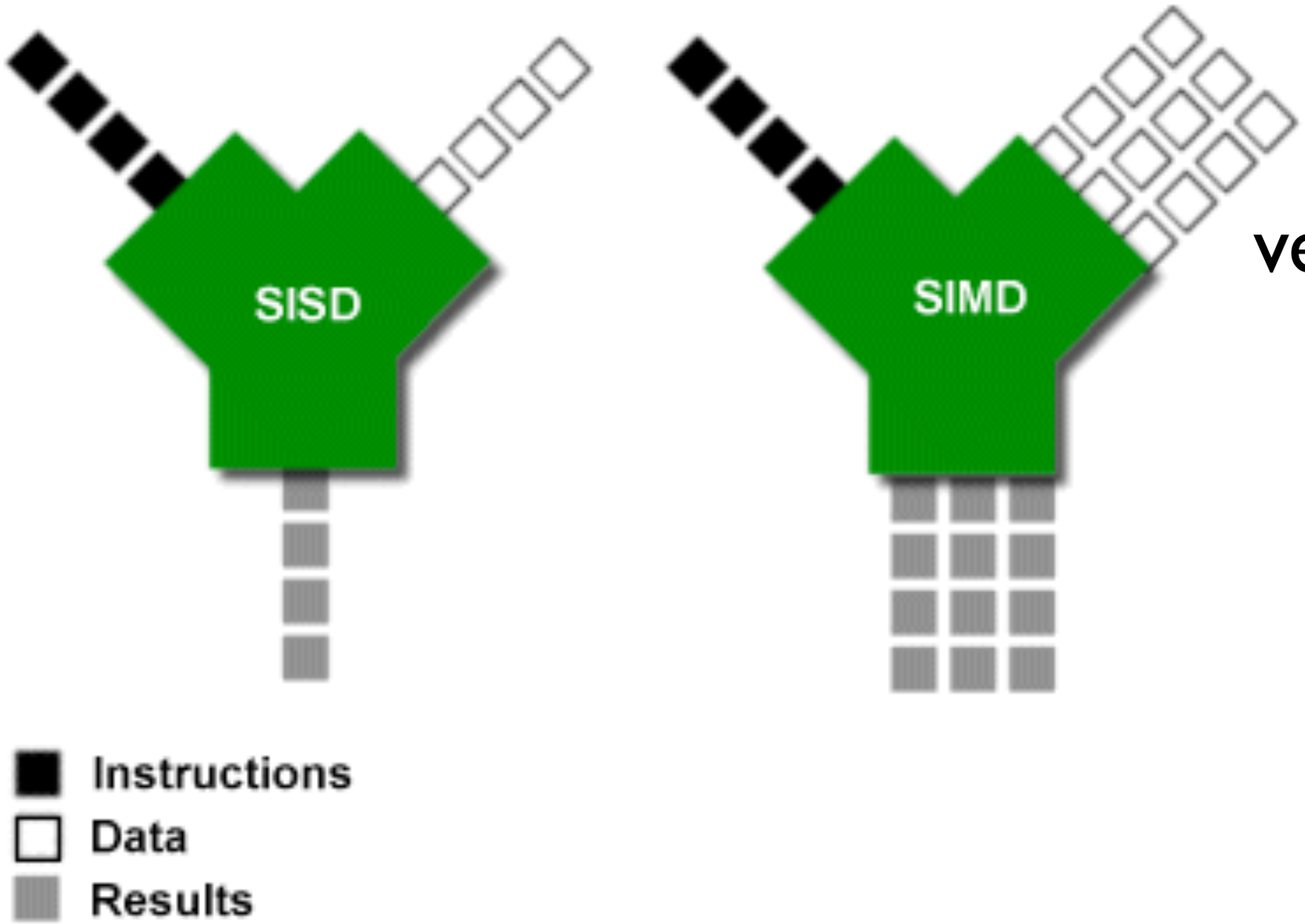
Superscalaire



Multiprocesseurs

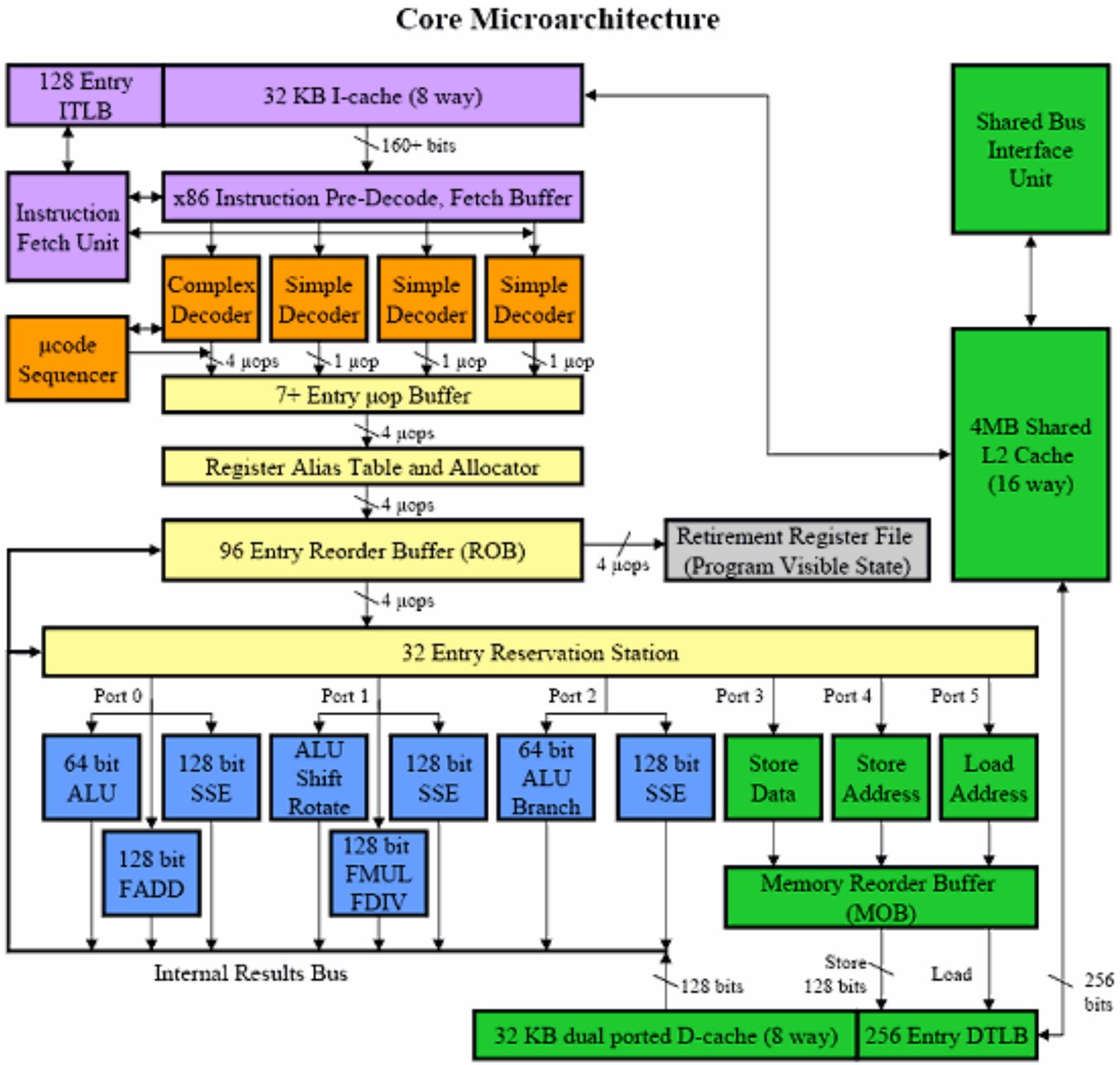


Parallélisme de données

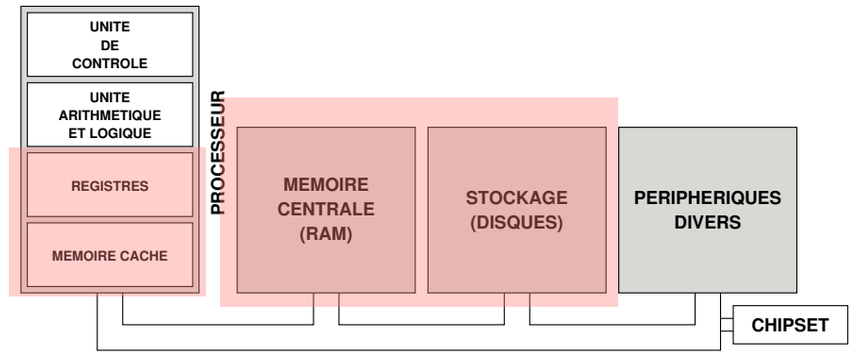


vecteurs

Micro-architecture

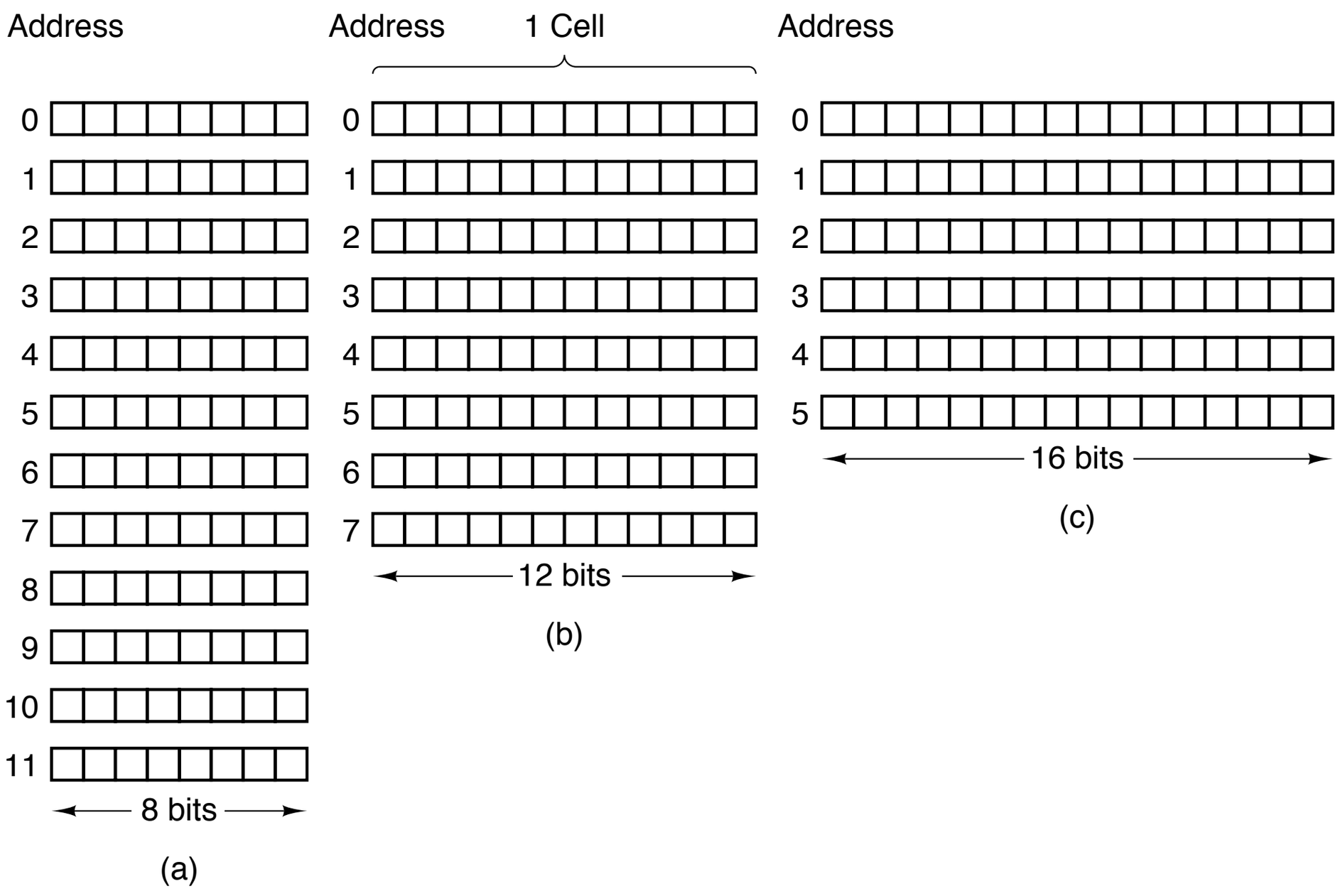


Mémoire


UNITE
DE
CONTROLE
UNITE
ARITHMETIQUE
ET LOGIQUE
REGISTRES
MEMOIRE CACHE
PROCESSEUR
MEMOIRE
CENTRALE
(RAM)
STOCKAGE
(DISQUES)
PERIPHERIQUES
DIVERS
CHIPSET

Vitesse, prix
Registers
Cache
Main memory
Magnetic disk
Tape
Optical disk
Capacité

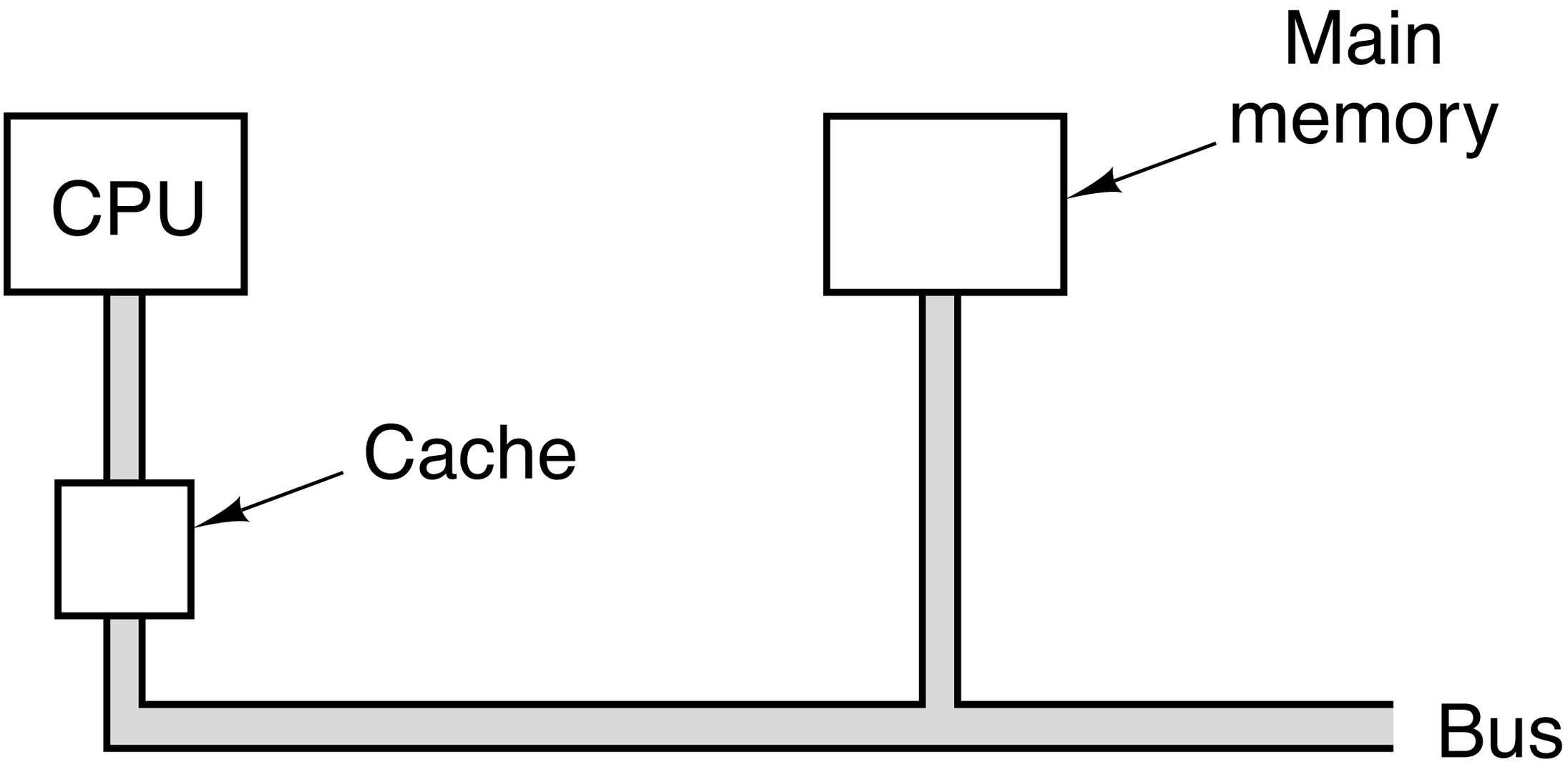
Organisation de la RAM



Types de RAM

- RAM statique (SRAM)
  - Registres, caches
    - 6 transistors par bit
- RAM dynamique (DRAM)
  - Mémoire centrale, caches L3
  - 1 transistor + 1 condensateur par bit

Mémoire cache



Localité temporelle

Exemple de programme en pseudo-code

...
Stocker 4 à l'adresse mémoire 120
Stocker 12 à l'adresse mémoire 234
Additionner le contenu de l'adresse 120 avec le contenu de l'adresse 234 et stocker le tout à l'adresse 345
Ajouter 3 au contenu de l'adresse 345
...

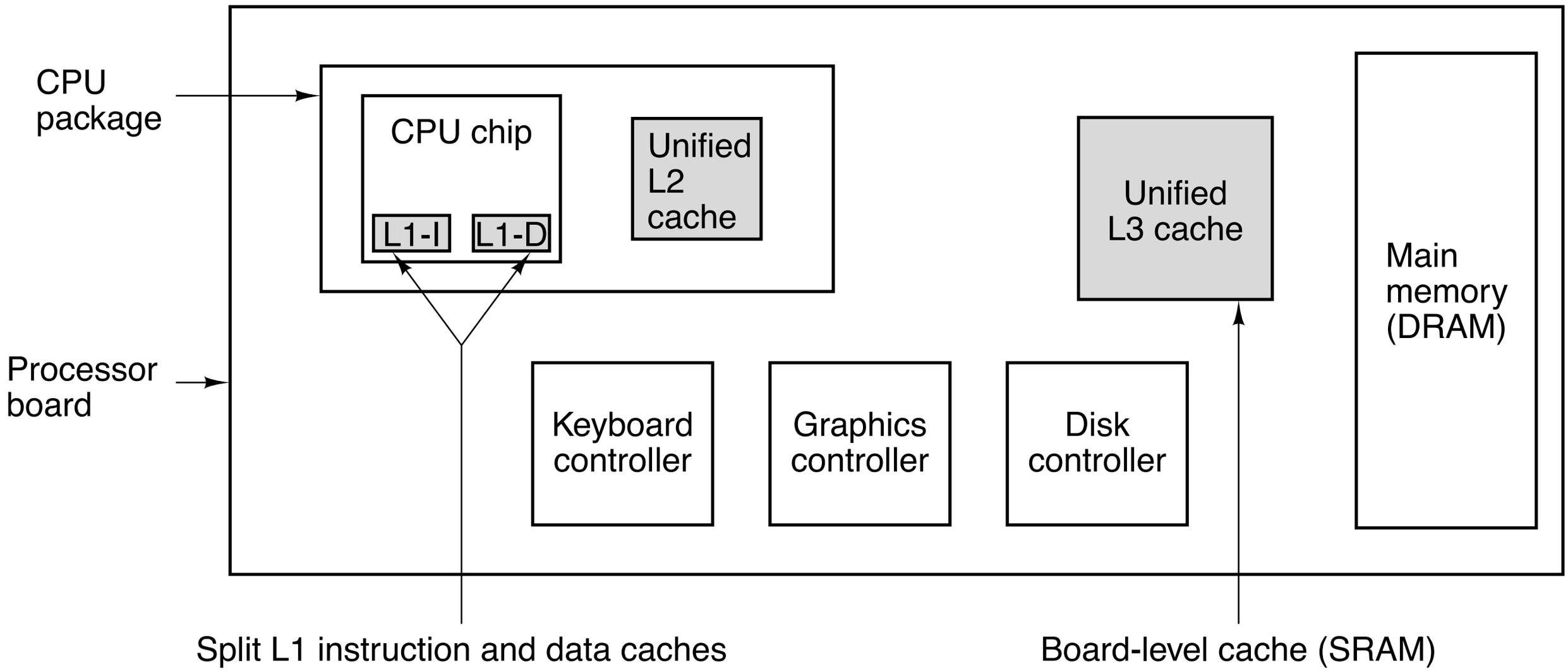
Localité spatiale

Les programmes manipulent des “blocs” de données.
Images, sons, films, matrices, etc...

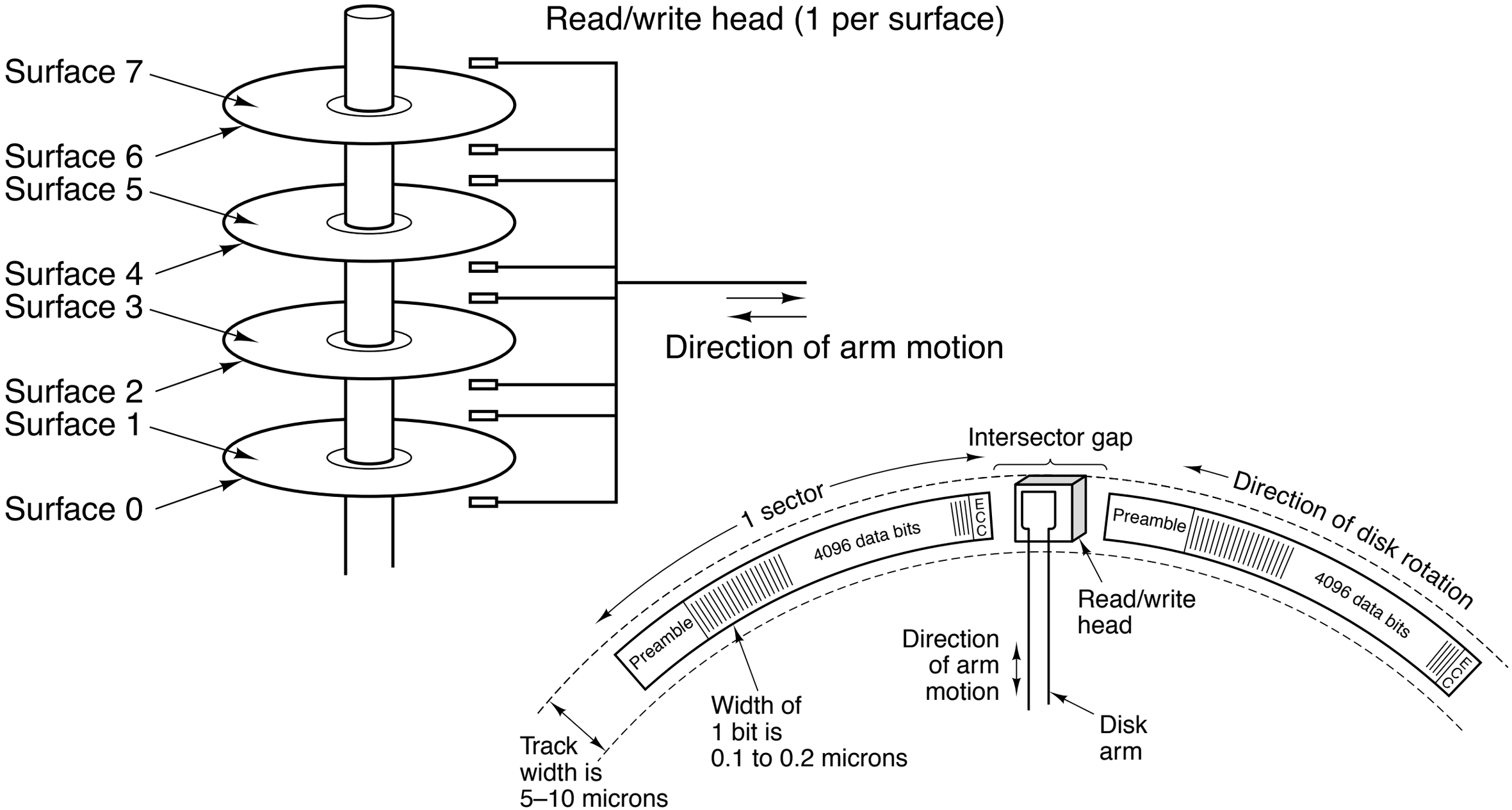
```
void transfo (int w, int h, unsigned char *bytes, unsigned char *dest,
              unsigned char *lut, unsigned char val)
{
  int i,j;

  for (j = 0; j < h; j++)
    for (i = 0; i < w; i++)
      {
        unsigned char current = bytes[j*w + i];
        current = lut[current];
        current = (((int)current + val)>255)?255:current+val;
        dest[j*w + i] = current;
      }
}
```

Caches

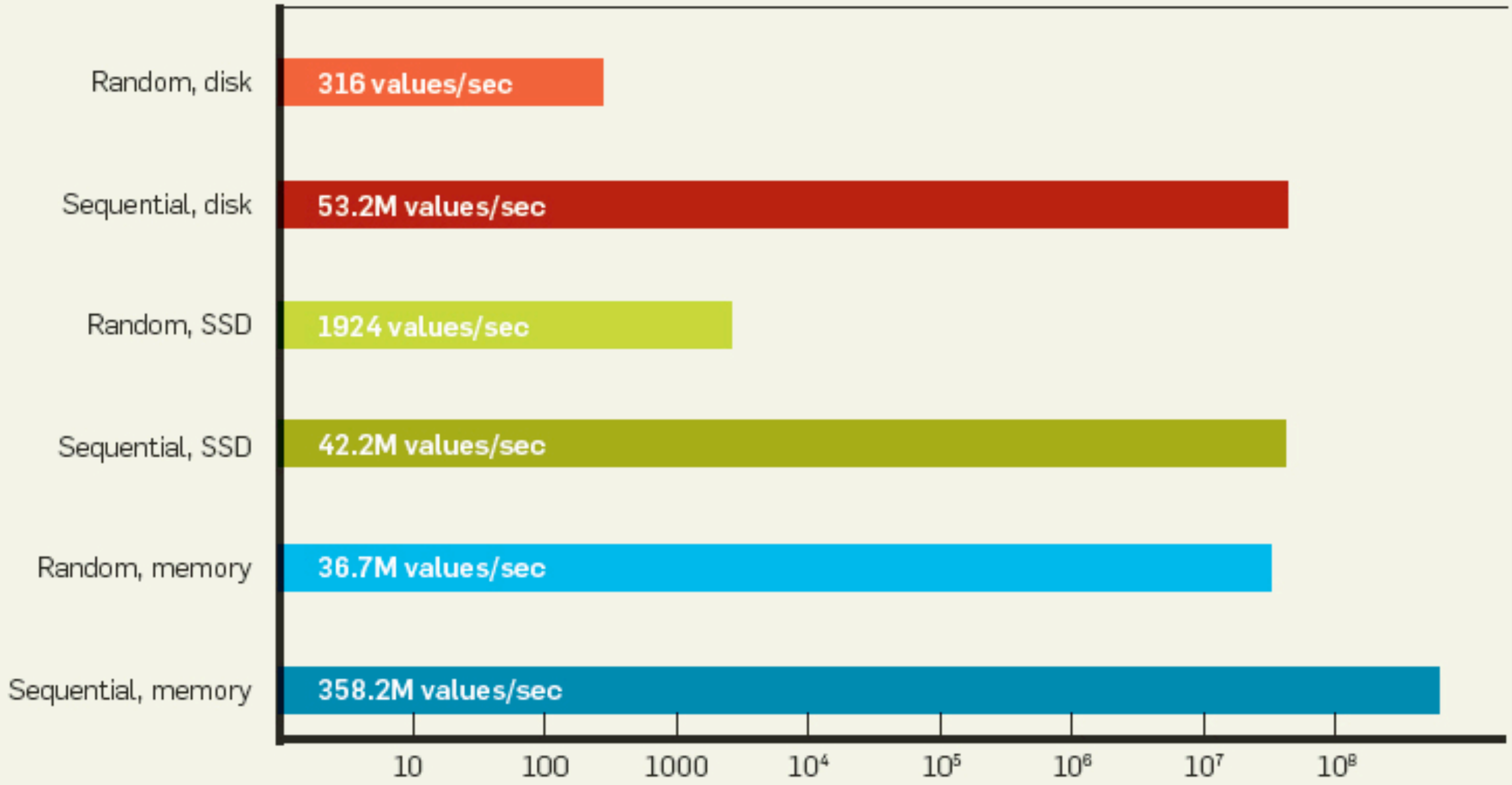

CPU
package
CPU chip
L1-I
L1-D
Unified
L2
cache
Unified
L3 cache
Main
memory
(DRAM)
Processor
board
Keyboard
controller
Graphics
controller
Disk
controller
Split L1 instruction and data caches
Board-level cache (SRAM)

Disques


Read/write head (1 per surface)
Surface 7
Surface 6
Surface 5
Surface 4
Surface 3
Surface 2
Surface 1
Surface 0
Direction of arm motion
Intersector gap
1 sector
Direction of disk rotation
Preamble
4096 data bits
ECC
Read/write head
Direction of arm motion
Disk arm
Width of 1 bit is 0.1 to 0.2 microns
Track width is 5–10 microns

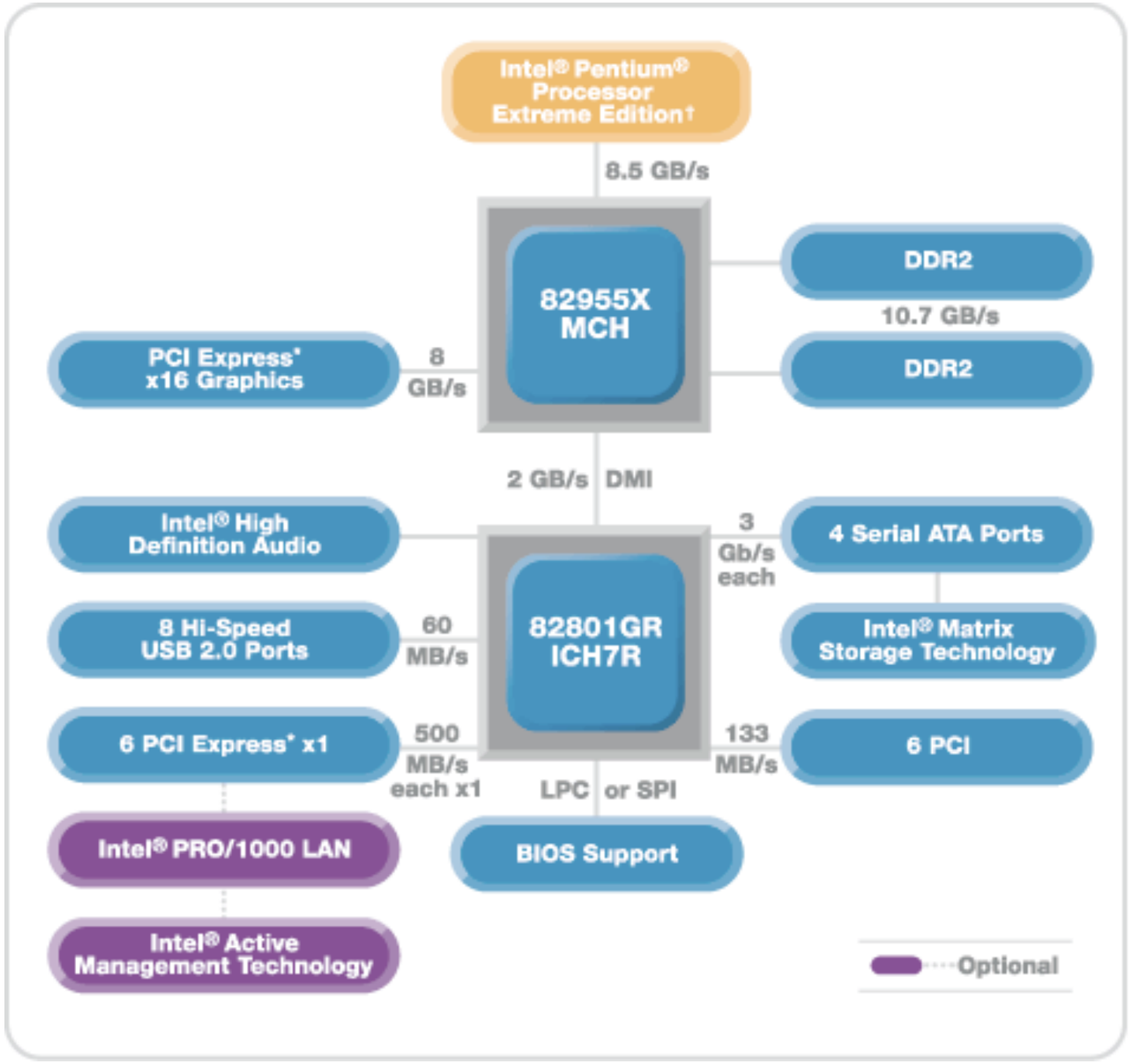
Disques (2)

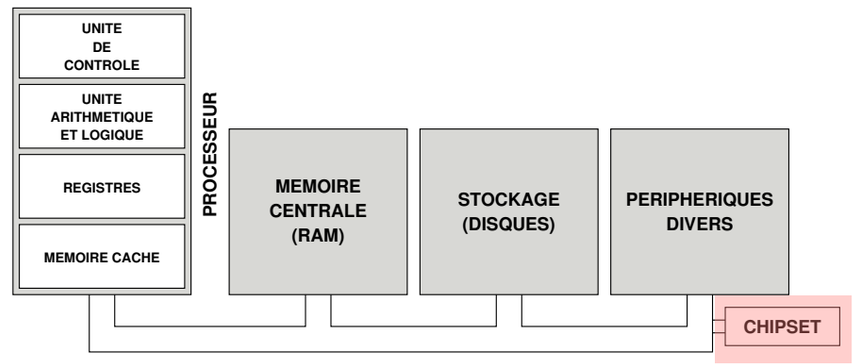
Néanmoins...



* Disk tests were carried out on a freshly booted machine (a Windows 2003 server with 64GB RAM and eight 15,000RPM SAS disks in RAID5 configuration) to eliminate the effect of operating-system disk caching. SSD test used a latest generation Intel high-performance SATA SSD.

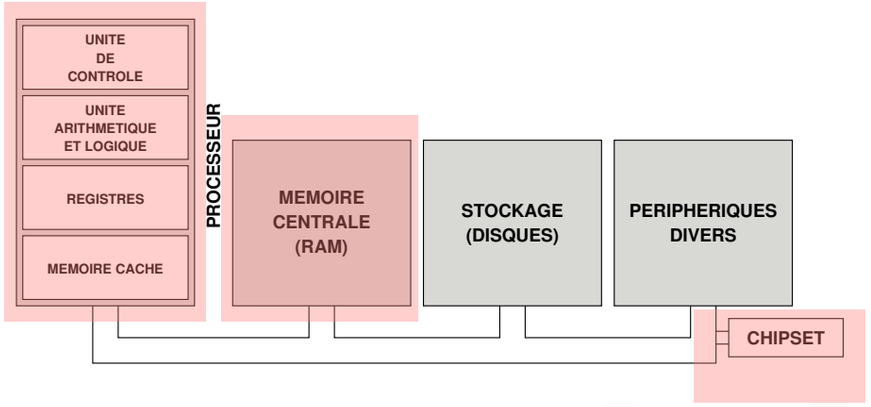
Chipset

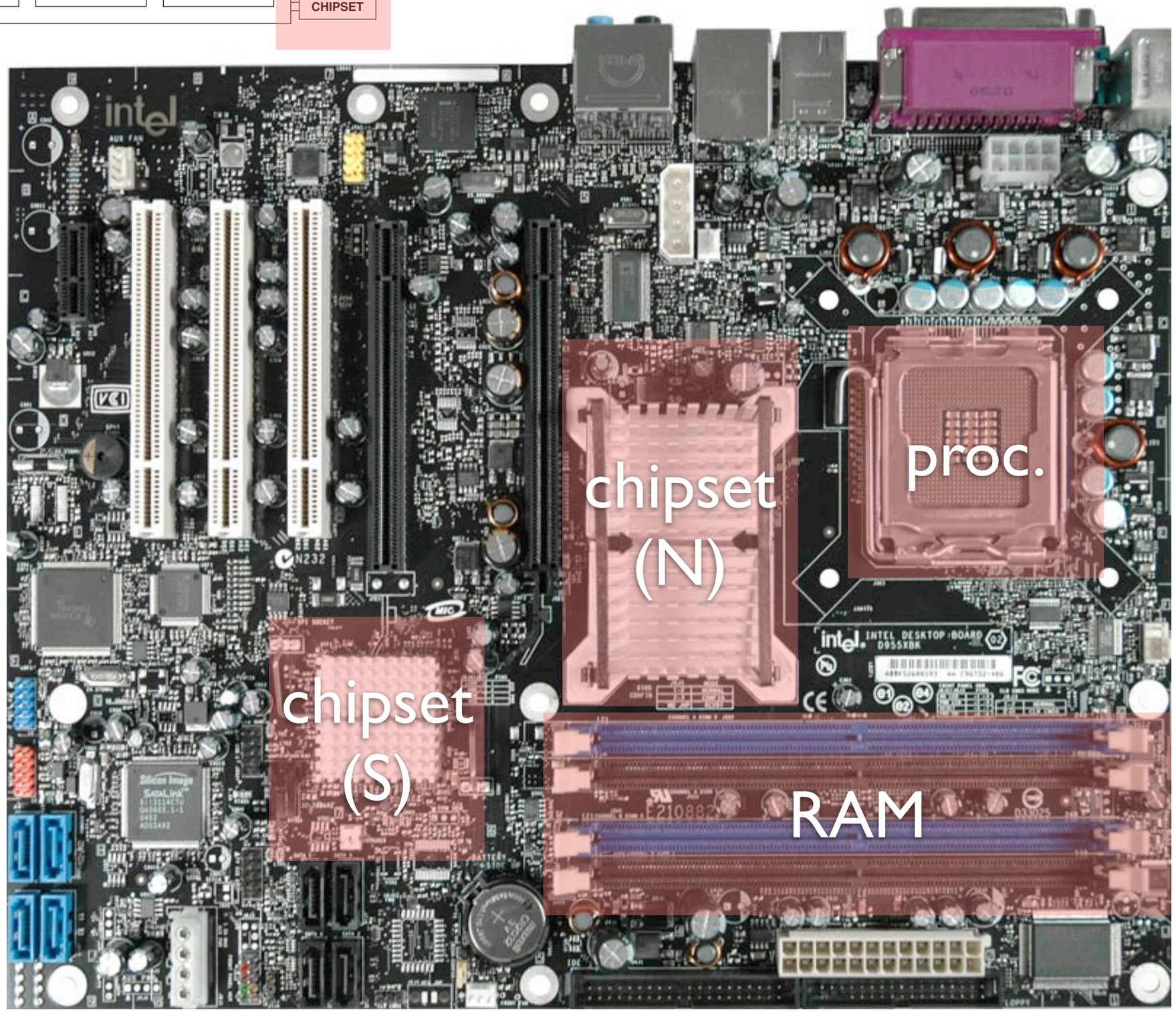




Chef d’orchestre de l’ordinateur

Carte mère

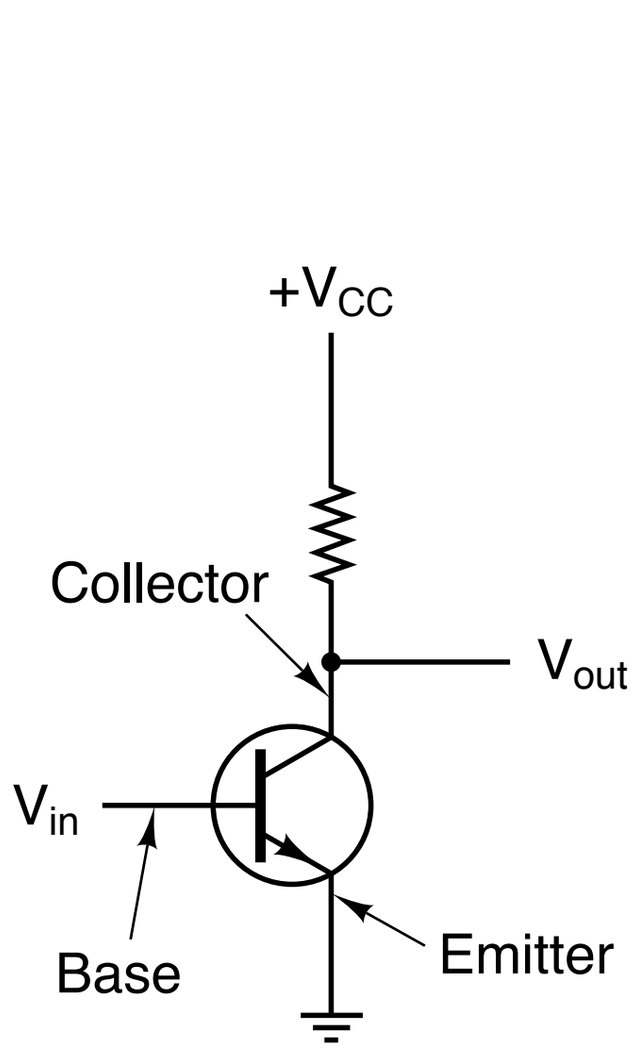

UNITE
DE
CONTROLE
UNITE
ARITHMETIQUE
ET LOGIQUE
REGISTRES
MEMOIRE CACHE
PROCESSEUR
MEMOIRE
CENTRALE
(RAM)
STOCKAGE
(DISQUES)
PERIPHERIQUES
DIVERS
CHIPSET


chipset
(N)
proc.
chipset
(S)
RAM

Un peu de technologie

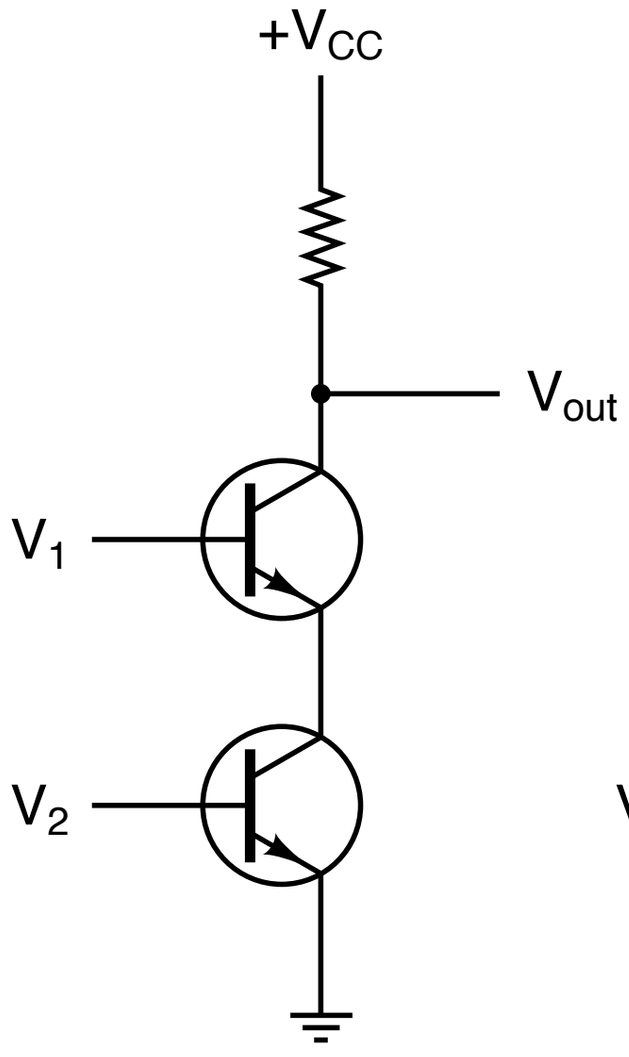
- George Boole (1815-1864)
- Claude Shannon (M.I.T 1938)
  - *A symbolic analysis of relay and switching circuits (1938)*
  - *The mathematical theory of communication (1948)*
- Shockley, Bardeen et Brattain (Nobel 1956)
  - Transistor (semiconducteur dopé)

Transistors (NPN)



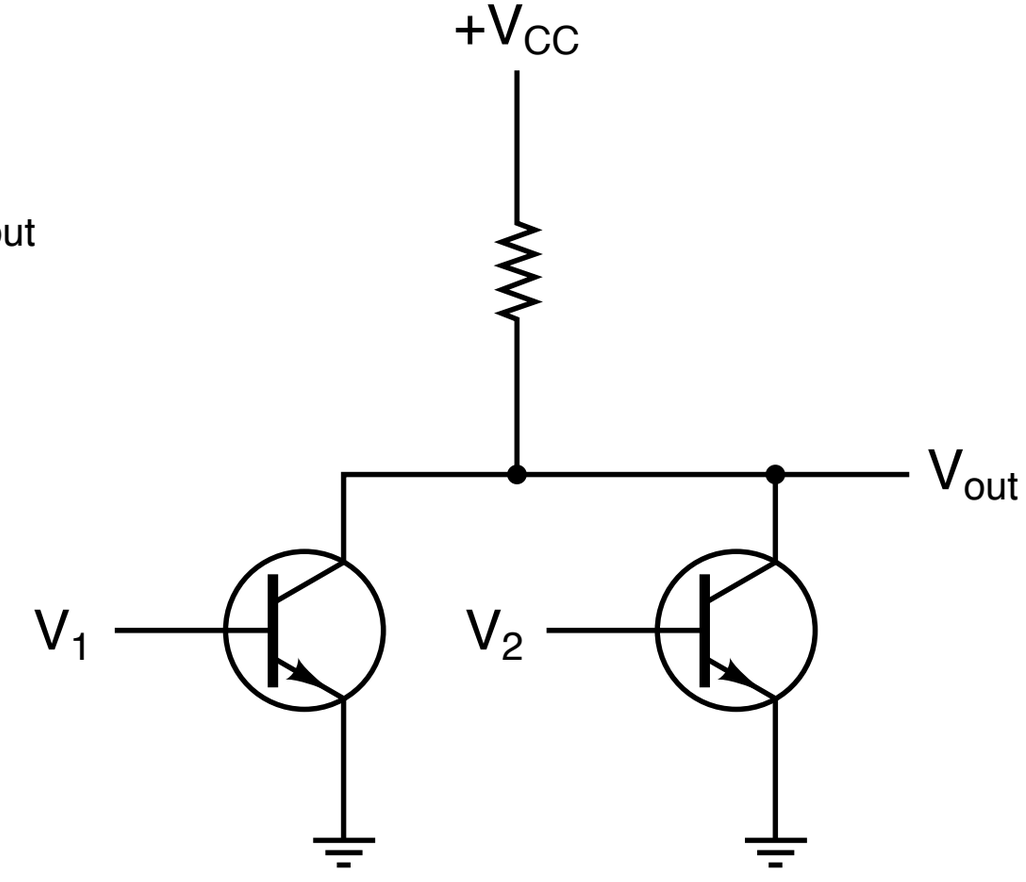
(a)

NOT



(b)

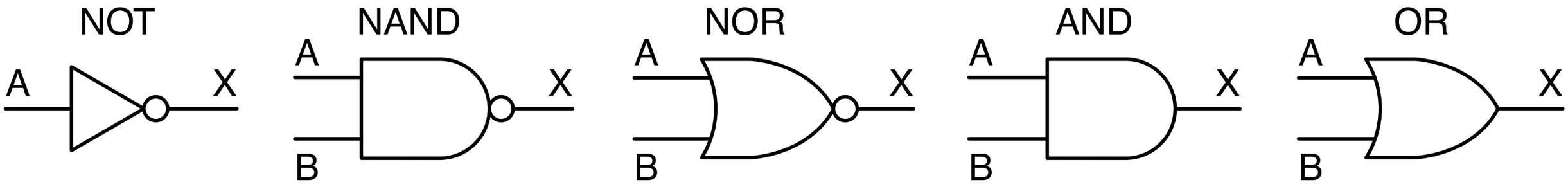
NAND



(c)

NOR

Portes



A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(d)

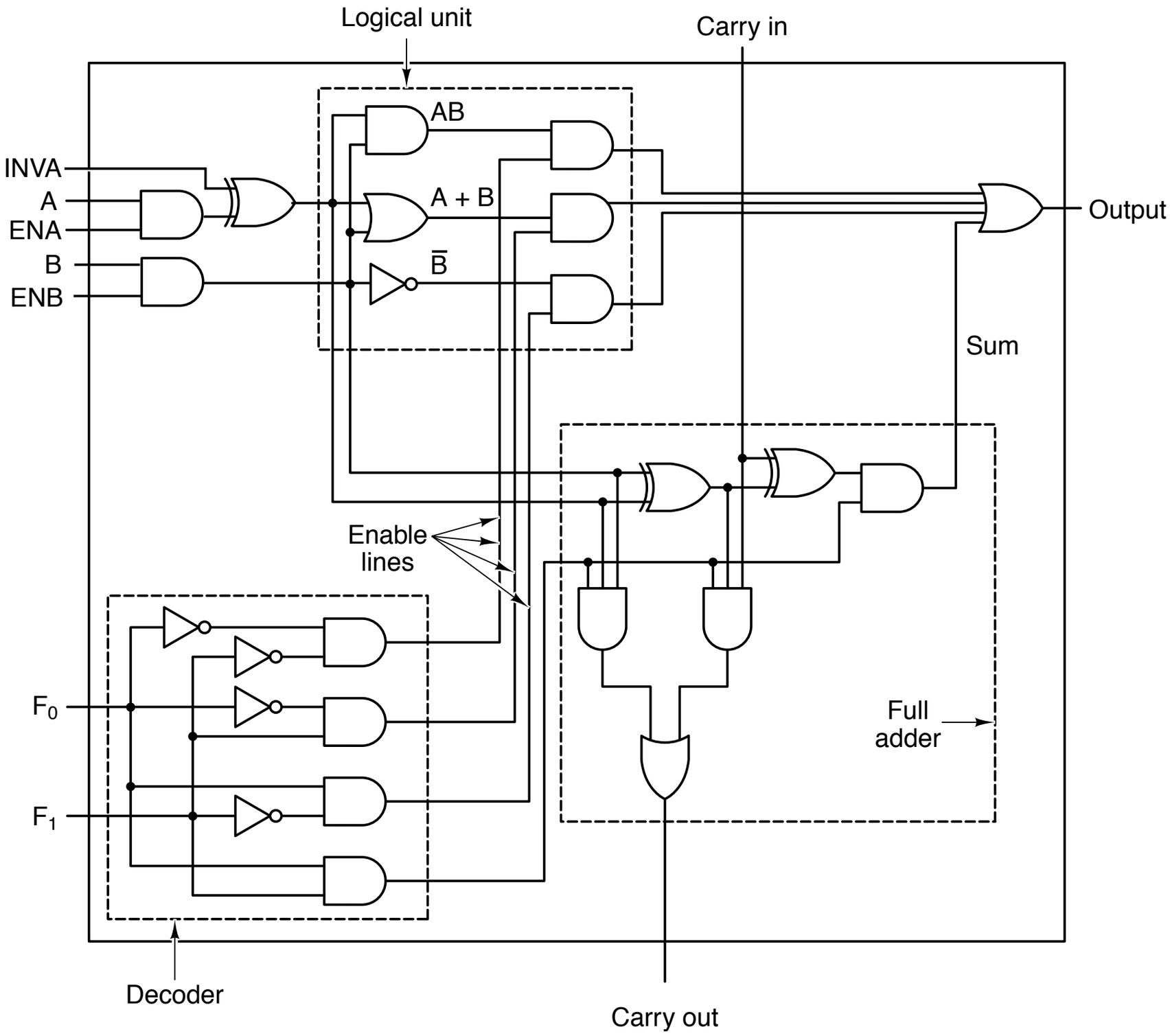
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

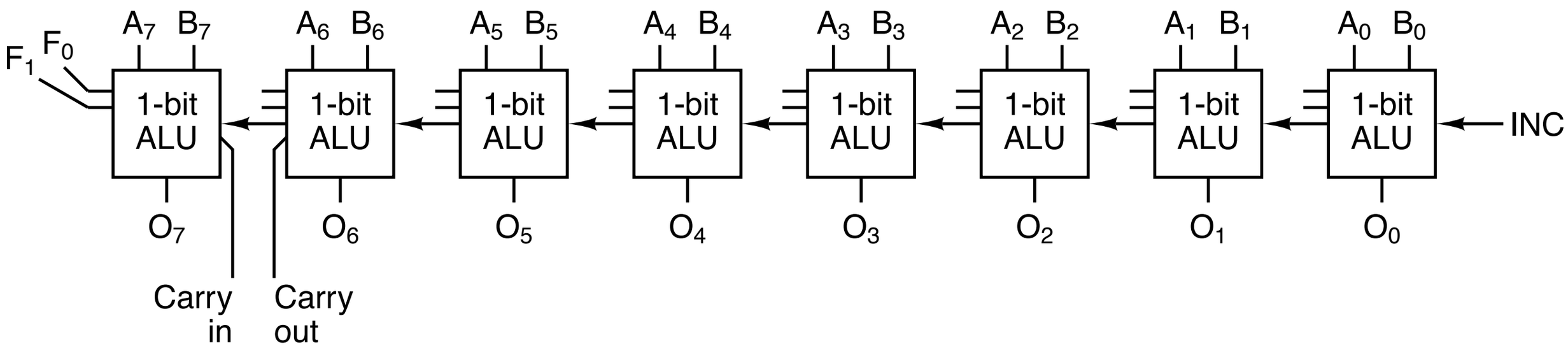
Additionneur 1 bit

A	B	$R_{Entrée}$	A + B + $R_{Entrée}$	R_{Sortie}
0	0	0	**0**	**0**
0	0	1	**1**	**0**
0	1	0	**1**	**0**
0	1	1	**0**	**1**
1	0	0	**1**	**0**
1	0	1	**0**	**1**
1	1	0	**0**	**1**
1	1	1	**1**	**1**

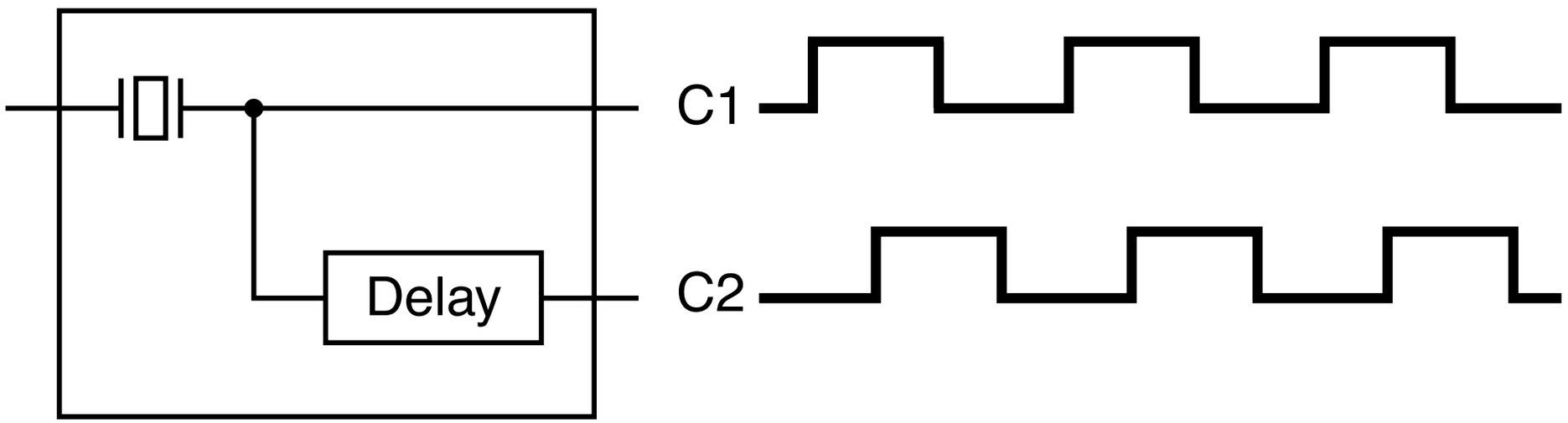
Additionneur 1 bit



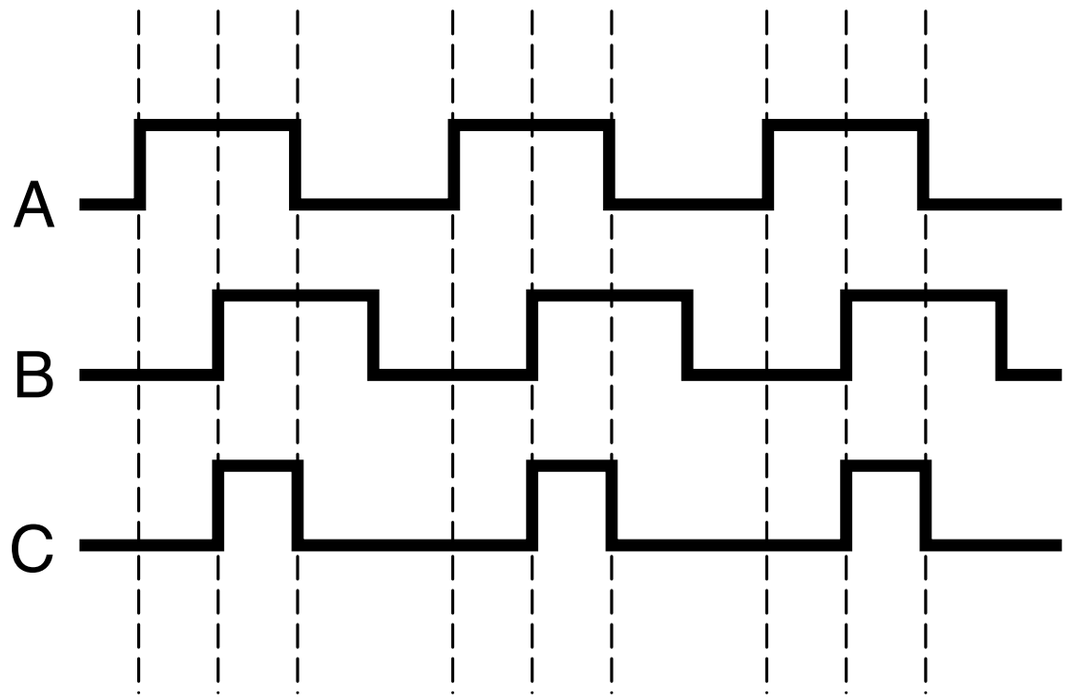
Additionneur 8 bits



Horloge



(a) (b)



(c)

Circuits intégrés

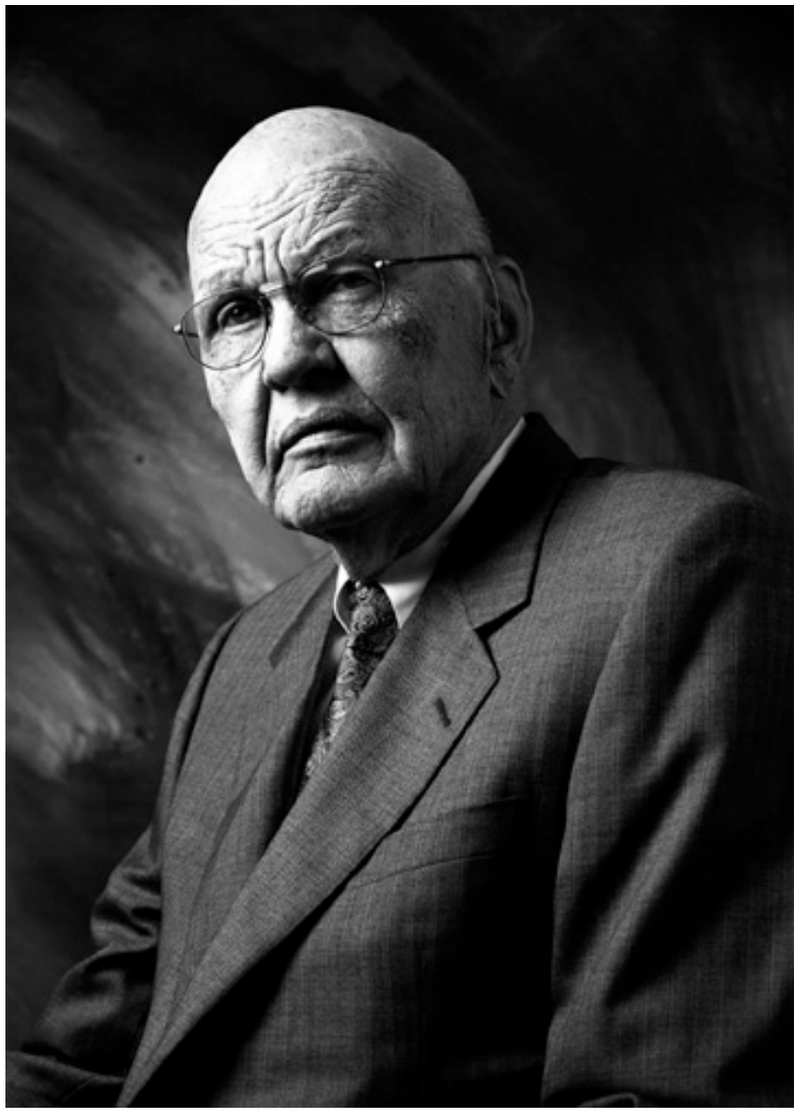


Photo Texas Instruments

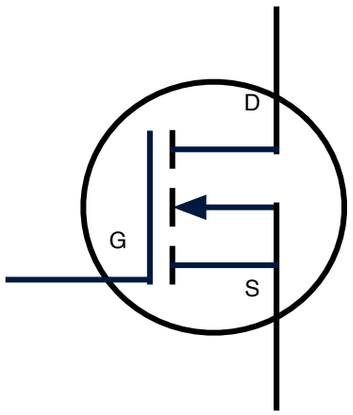
Photo Texas Instruments

Photo Texas Instruments

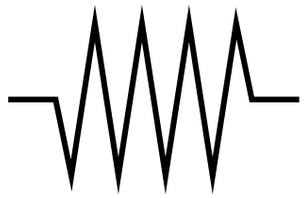
Jack Kilby (1923-2005) — Prix Nobel de physique 2000

Un peu d'histoire

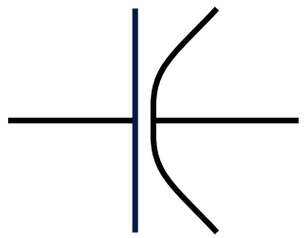
Circuit intégré = circuit électronique



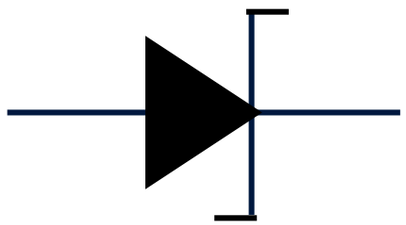
transistor



résistor

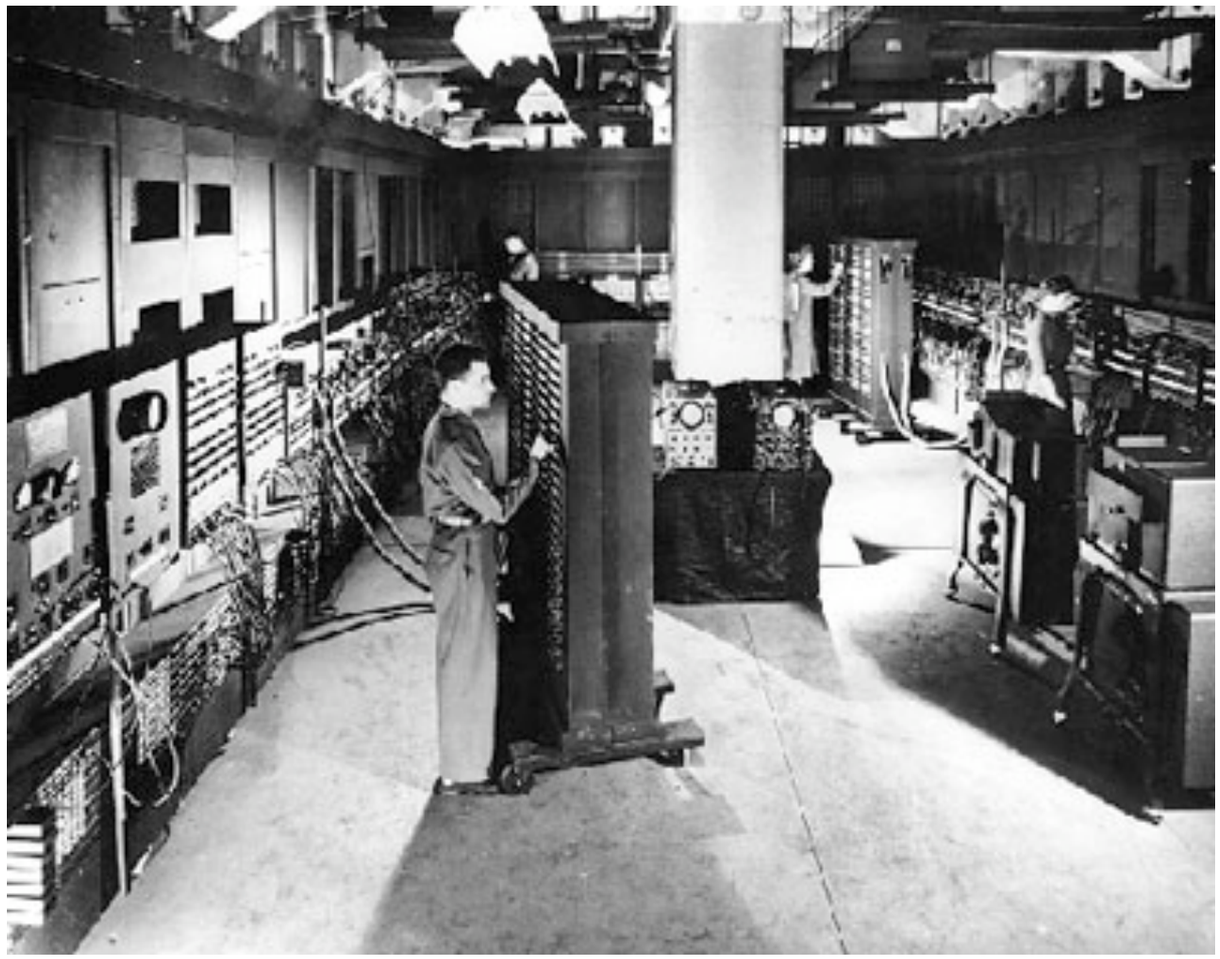


condensateur



diode

Avant le transistor



ENIAC (1945) — 18000 tubes — 200 000 W

Transistor (1947)

- Élément fondamental d'un ordinateur
- Révolution dans les années 50
- Circuits de plus en plus complexes
- "*Tyranny of numbers*"

Photo Nobelprize.org

Pas de vacances pour Kilby

1958 — Texas Instruments

- Bloc monolithique de **semiconducteur**
- Couche de métal ajoutée par dessus
- Plus de fils ou de composants ajoutés à la main

Première puce de Kilby

Photo Texas Instruments

Photo Texas Instruments

Robert Noyce

- Même idée (1959)
- Couche de métal ajoutée à la fin
- Supression à certains endroits pour établir les connexions
- Permet d'améliorer la production de masse

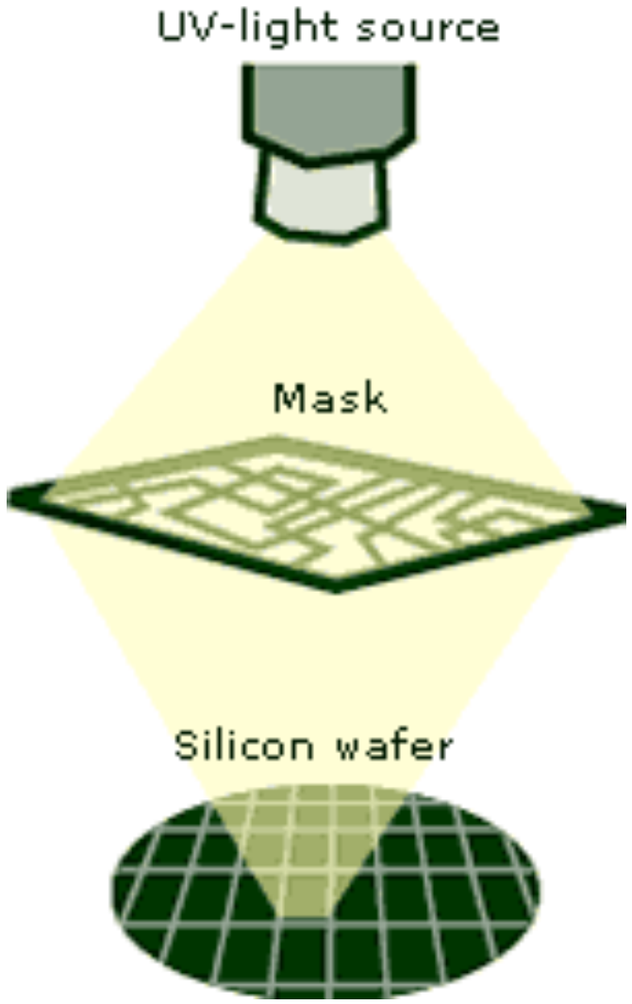
Photo Intel Corp.

Robert Noyce — Co-fondateur d'Intel avec Gordon Moore

Fabrication (1)

Photolithographie

- Source UV puissante
- Masque
- Film photosensible
- Silicium

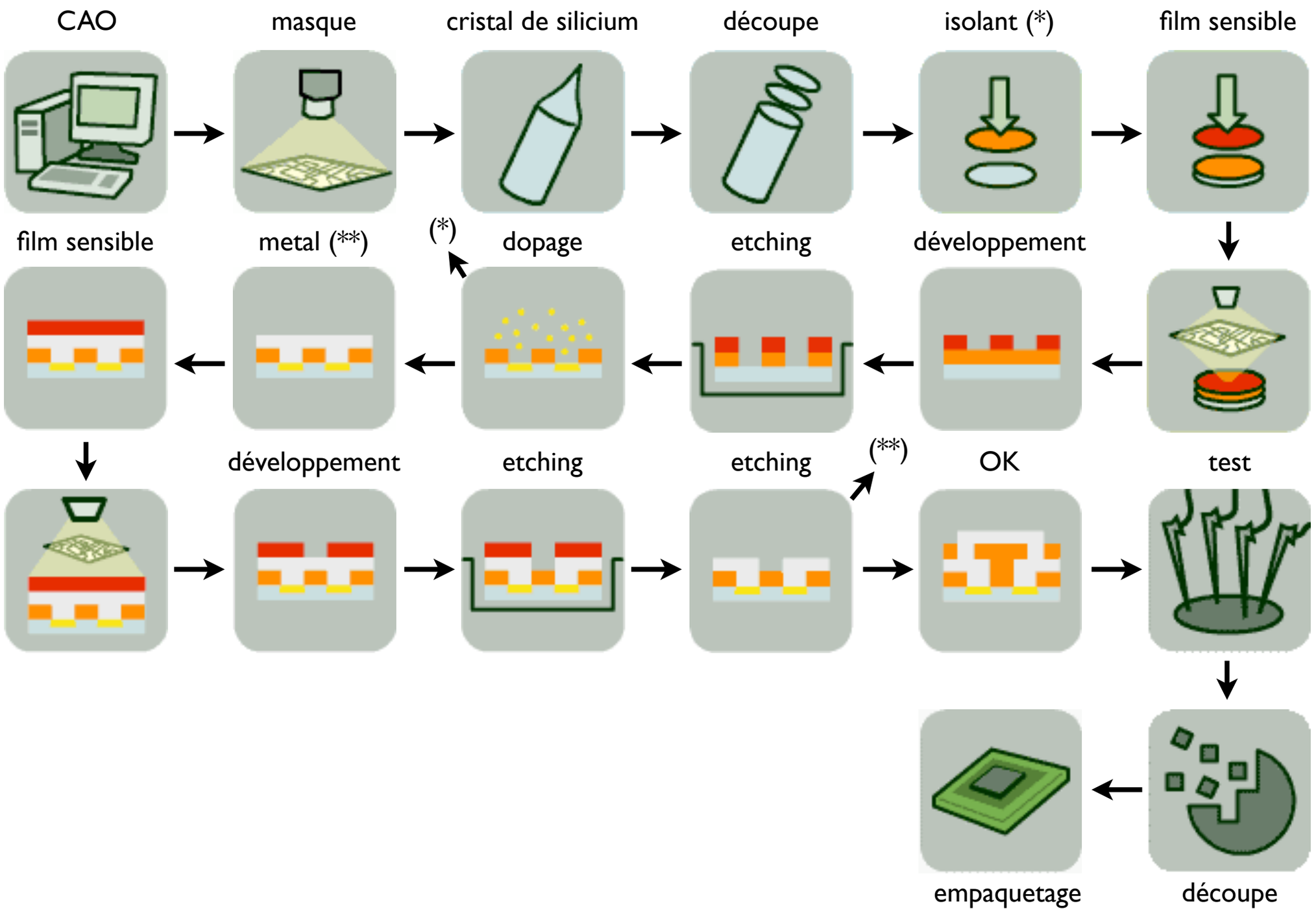


Fabrication (2)

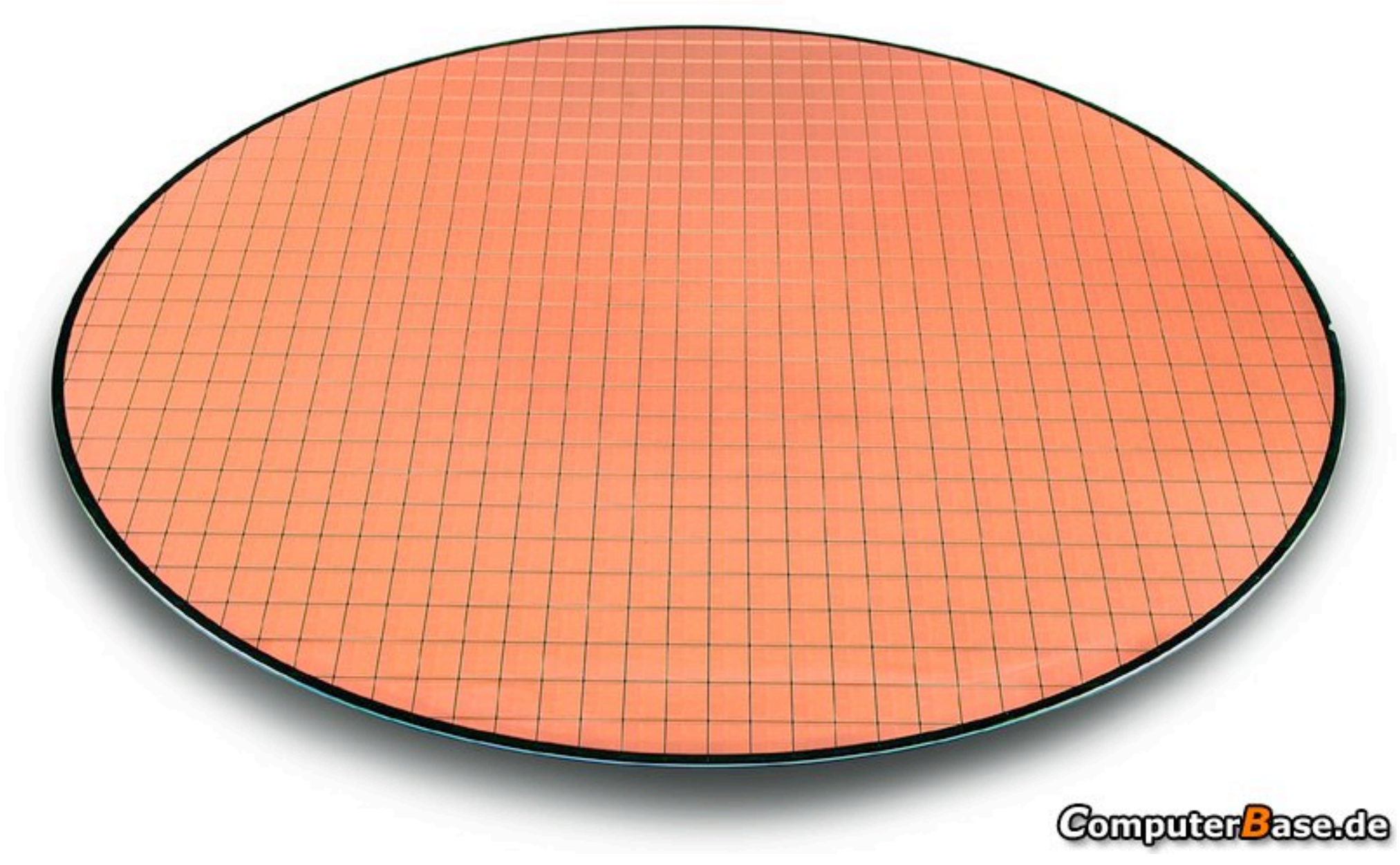
- Propreté drastique
- Combinaisons étanches

Photo Intel Corp.

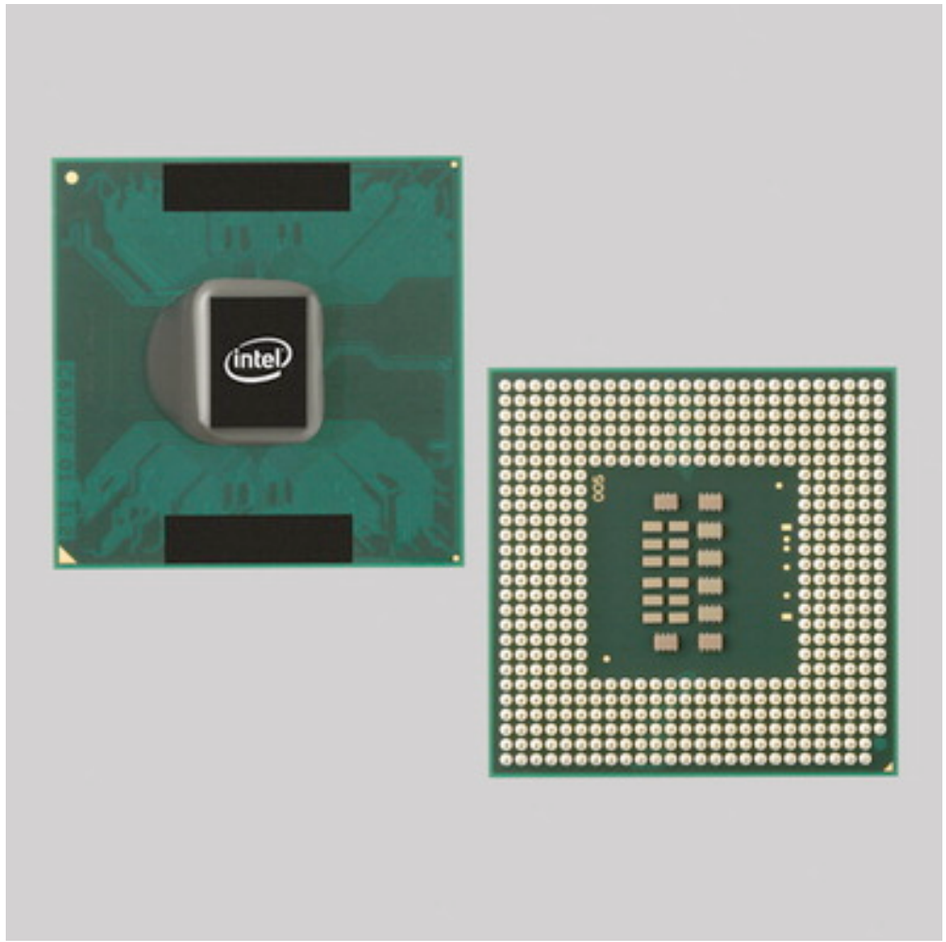
Fabrication (3)

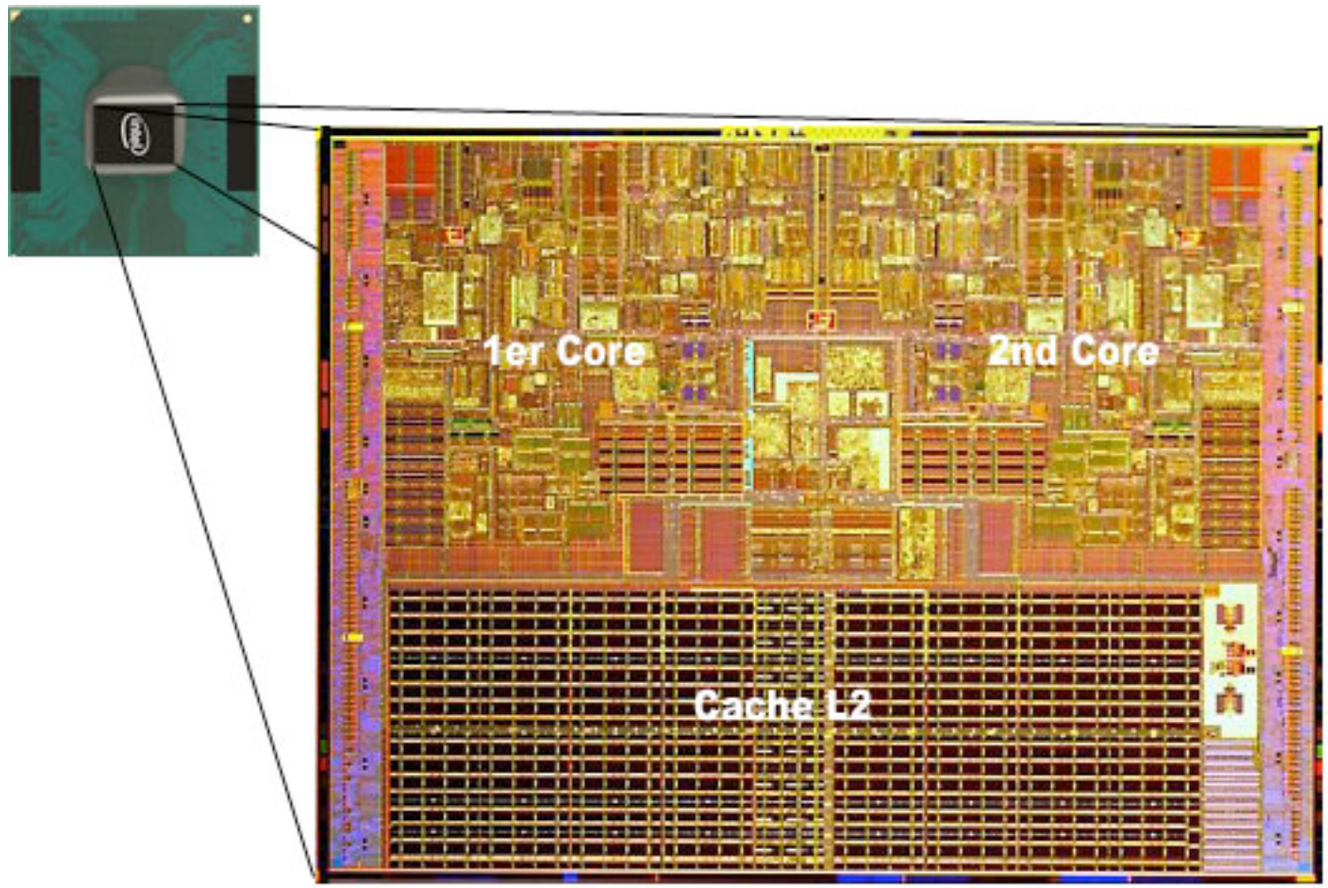


Fabrication (4)



Fabrication (5)





Évolution des circuits

- 3D
- VLSI
- Taille des transistors : 65nm, 45nm, 32nm
- SoC : System on a Chip
- 5 nanomètres : plus de transistor...
- Atome : 0,1 nm

Couche Logicielle

Microsoft Corporation, 1978

Logiciels de base

- BIOS (ROM)
- Boot loader (disque)
- Système d'exploitation (disque)

Séquence de démarrage PC x86

- Mise sous tension
- Chargement du code à l'adresse 0xFFFF0000 (BIOS)
- POST (Power On Self Test)
- Recherche d'un périphérique dans la liste des périphériques de boot
  - Disquette, disque dur, clé USB, réseau, CDROM, ...

Séquence de démarrage PC x86 (suite)

- Lecture du contenu du premier secteur (512 octets)
  - Appelé le MBR (Master Boot Record)
    - Doit se finir par 0xAA55
    - 64 octets pour la table des partitions
    - Code binaire de 446 octets au maximum
- Stockage à l'adresse mémoire 0x00007C00
- Exécution de ce code (saut)

Séquence de démarrage PC x86 (suite)

- Le code du MBR charge souvent un autre code
  - Au début d'une partition ou via le réseau
    - Exemple: LILO, GRUB, NTLDR
    - Choix de l'OS par l'utilisateur
- Linux: chargement du "kernel" (noyau)
- Linux: le noyau s'endort et passe la main au premier processus (Init) qui se multiplie.

Système d'exploitation

- Programme assurant la gestion de l'ordinateur et de ses périphériques
- Est-il nécessaire ?
  - Abstraction du matériel (machine virtuelle)
  - Optimiser l'utilisation des ressources
    - Matérielles et logicielles
  - Partager ces ressources (temps, espace)

Exigences

Evolutivité

Connectivité

Taille

Fiabilité

Efficacité

Sécurité

Beauté

Prix

Richesse

Tout système est un compromis entre ces différents critères non exhaustifs !

Processus

- Programme en cours d'exécution
- Contenu
  - Code du programme
  - Données courantes
  - Informations
    - Position dans le programme
    - Fichiers ouverts
    - Propriétaire, ...

Traitement par lot

- Batch processing
- L'utilisateur donne plusieurs commandes dans une queue d'exécution de programmes.
- Pipeline
- Chaque programme est exécuté sous la forme d'un processus monopolisant toutes les ressources jusqu'à sa fin.

Multi-tâches

- Plusieurs processus sont en cours d'exécution "en même temps"
- Il n'y a qu'un seul processus par ressource d'exécution du processeur à un instant *t*
- Les processus qui ne sont pas en cours d'exécution sont stockés en mémoire (RAM ou disque)

Approches du Multi-tâches

- L'utilisateur passe d'un processus à l'autre quand il le décide
- Chaque application décide quand elle laisse la main aux autres
- Le système gère les processus et les fait s'exécuter à tour de rôle sur les ressources d'exécution
  - Multi-tâches préemptif

Multi-utilisateurs

- Un système multi-tâches et nécessaire pour un système multi-utilisateurs en temps partagé (*time sharing*)
- Gestion des identités d'utilisateurs
- Cloisonnement des processus et des fichiers

Temps réel

- Doit garantir des temps de réaction bornés pour des signaux extérieurs urgent
- Sert pour le pilotage et le contrôle de déroulements externes (centrales nucléaires)
- La plupart des systèmes n’y arrivent pas

Systèmes distribués

- Un seul programme va s'exécuter sur des ressources d'exécution distribuées.
- Plusieurs *processus* éventuellement divisés en *fils d'exécution* (*threads*).
- Dans ce domaine, on en est encore à la préhistoire.

Couches d'un S. E.