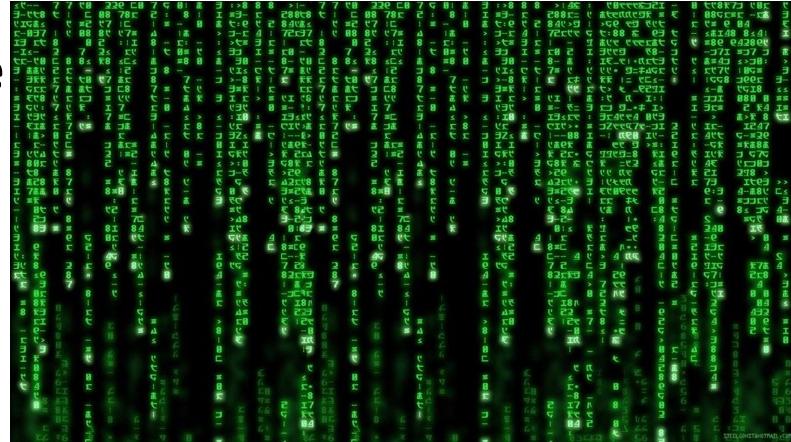


# R1.08 Informatique

**Julien GABIOT**  
**2022-2023**



## Sommaire S1

- 1 – Introduction – Architecture des ordinateurs – Algorithme
- 2 – Base du C
- 3 – Les fonction
- 4 – Les conditions
- 5 – Les boucles
- 6 – Les tableaux
- 6.5 – Les pointeurs
- 7 – Introduction au langage objet C++
- 8 - Introduction au langage objet C++ (suite)



## Volume Horaire

50h total

8h CM

18h TD

24h TP (6 ou 7 TP)



IUT de Vélizy-Rambouillet  
CAMPUS DE VÉLIZY-VILLACOUBLAY




---

R1.08 Informatique – Julien Gabiot – 2022/2023

3

## Descriptif officiel

Descriptif :

- concevoir l'algorithme d'un programme simple à partir d'un cahier des charges,
  - coder dans un langage évolué en respectant les bonnes pratiques (indentation, commentaires,...)
  - vérifier le fonctionnement.
  
  - Transmettre les bases de la programmation dans un langage évolué.
- => Choix du C / C++



IUT de Vélizy-Rambouillet  
CAMPUS DE VÉLIZY-VILLACOUBLAY




---

R1.08 Informatique – Julien Gabiot – 2022/2023

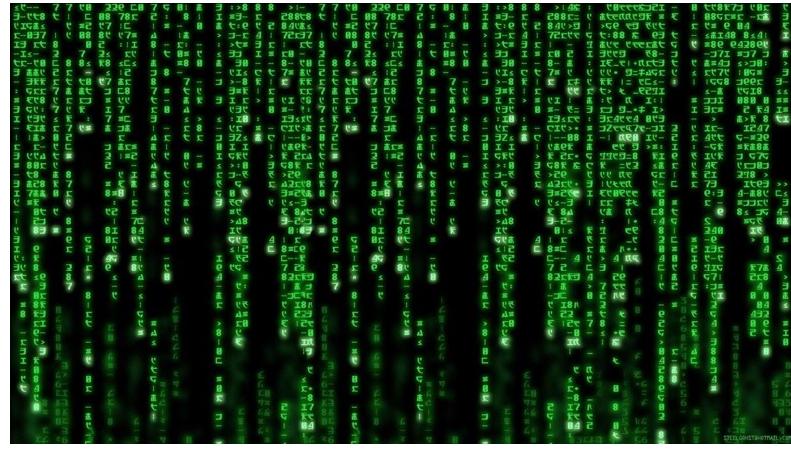
4

4

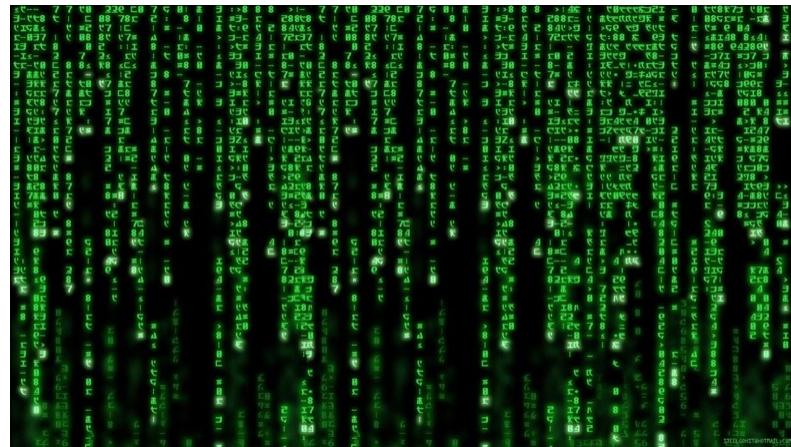
# Introduction – Architecture des ordinateurs – Algorithme

**Julien GABIOT**  
**2022-2023**

**Sources :** David Aubert  
 Charly Pecqueux-Guezene



## Historique



## Historique

1936 – 1956 : Première génération



Harvard Mark I : construit chez IBM puis envoyé à Harvard (1944).

Calculs : *Dispositif électro-mécanique*

Mémoire à tores de ferrite

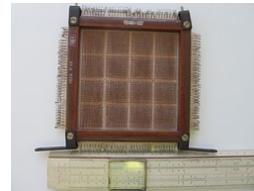
→ A donné son nom à l'architecture Harvard



Tube à vide (triode ici)



ENIAC : commandé en 1945 par l'armée américaine pour les calculs balistiques



Mémoire à tores de ferrite du BESM-6 soviétique



## Historique

1957-1965 : Deuxième génération



Premier transistor (1947)

Bell Labs

**Remplacement des tubes par des transistors**

→ Beaucoup plus légers, petits, robustes, consomment moins d'énergie, fonctionnent instantanément à la mise sous tension...



IUT de Vélizy-Rambouillet  
CAMPUS DE VÉLIZY-VILLACOUBLAY



## Historique

1957-1965 : Deuxième génération

PDP-1 produit en 1959.

Mémoire équivalente à 9,216 octets

"clock speed" à 187 kilohertz

730 kg

**Vendu pour seulement 120 000 \$ par la société**

*Digital Equipment Corporation (DEC)*

« Spacewar »



R1.08 Informatique – Julien Gabiot – 2022/2023

9

## Historique

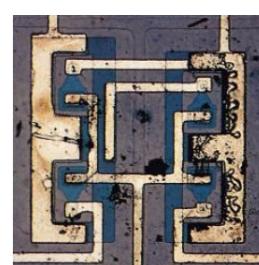
1963-1971 : Troisième génération

### Ordinateurs à circuits intégrés

Invention des circuits intégrés par *Jack Kilby* en 1958



Double bascule avec 4 transistors. (1.5mm de largeur)



Le mini-ordinateur HP 3000 fait son apparition...  
(1972)



R1.08 Informatique – Julien Gabiot – 2022/2023

10

10

## Historique

1971- fin des années 1980: Quatrième génération

### Apparition du microprocesseur

→ Beaucoup plus légers, petits, robustes, consomment moins d'énergie, fonctionnent instantanément à la mise sous tension...



Microprocesseur AMD

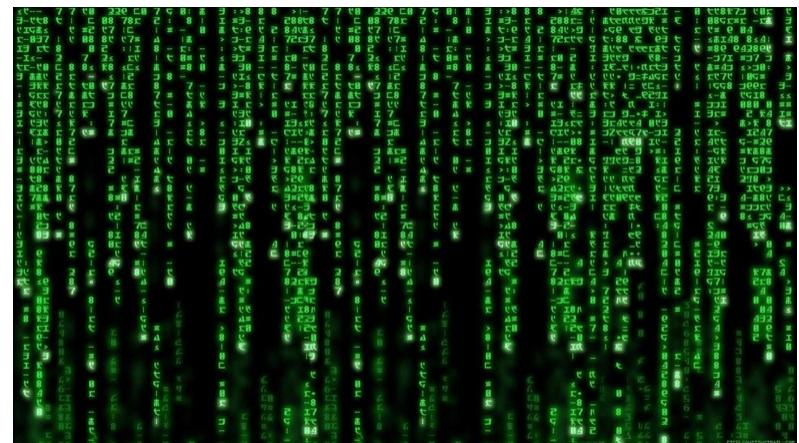


Supercalculateur



11

## Architecture

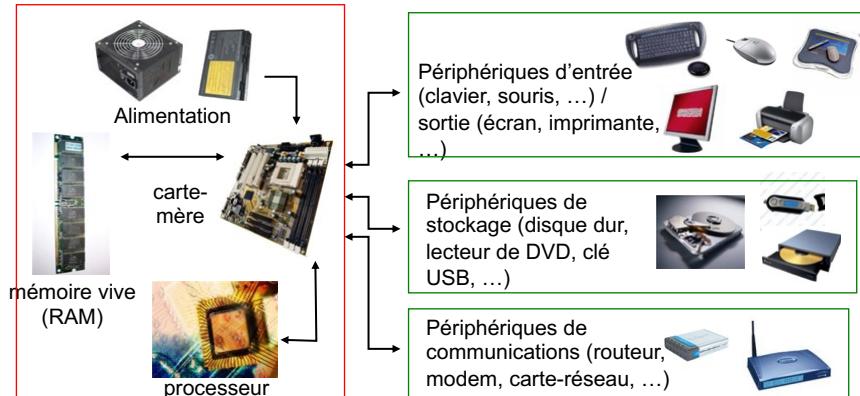


12

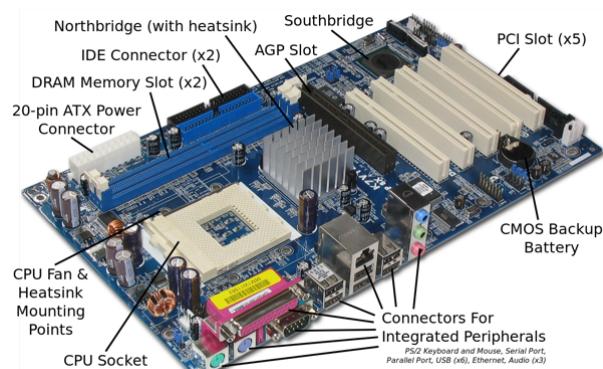
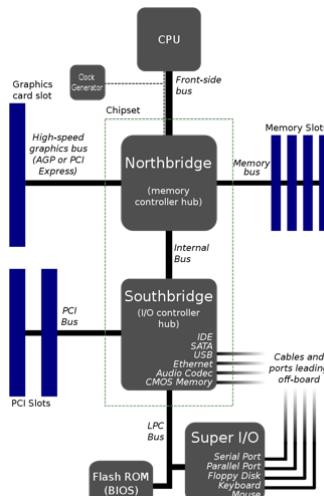
## La partie physique : l'ordinateur (*hardware*)

*Les différents composants de l'ordinateur*

Schéma très simplifié d'un ordinateur :



## La carte mère



**Northbridge et Southbridge** forment le **Chipset**.

Ils peuvent être séparés (comme ci-dessus) ou réunis dans un seul circuit intégré

## La carte mère

La carte mère est un **circuit imprimé** qui assure la liaison des différents composants et périphériques de l'ordinateur (*compatible PC*).

Ce circuit comporte **plusieurs couches**

Prise en compte du **temps de propagation des signaux**

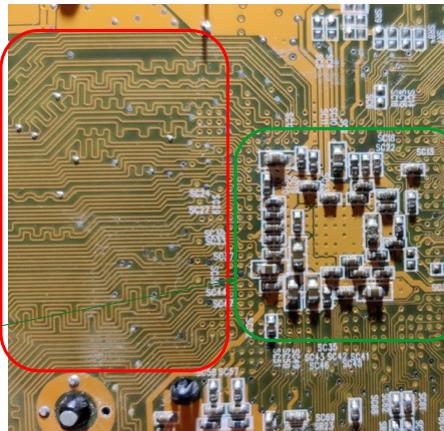
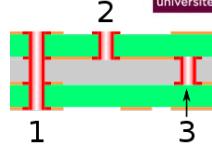
→ Grande complexité qui exclut toute production non manufacturée.

« Ziggags » permettant d'adapter les temps de propagation des signaux les uns par rapport aux autres

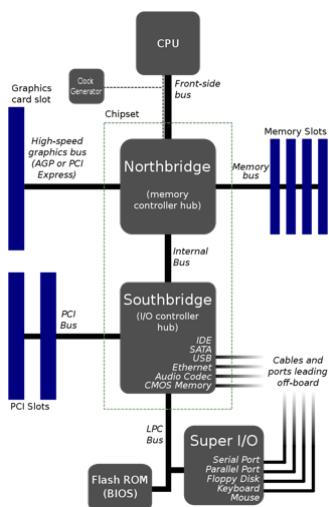
→ Ils doivent arriver en même temps dans le processeur

Dessous du processeur

→ Condensateurs pour filtrer le bruit et les perturbations sur les pistes



## La carte mère



### Northbridge (« pont nord »):

→ Directement relié au *microprocesseur*

→ Gère les périphériques « rapides » (mémoire, processeur graphique [GPU], southbridge)

### Southbridge (« pont sud »):

→ Contrôle tous les bus et périphériques plus lents que ceux du Northbridge.



## Les BUS de communication



Définition : Un **bus** est un dispositif de transmission de données partagé entre plusieurs composants d'un système numérique.

Différent d'une liaison → communication point à point

Bus parallèle : Les bits sont envoyés *en parallèle*. Chaque ligne (fil ou piste) du bus correspond à un bit.

→ coûteux

→ peu fiables sur des distances importantes

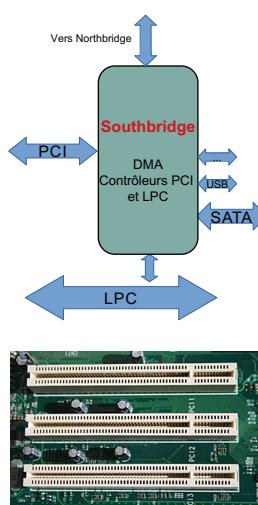
Bus série : Les bits sont envoyés les uns à la suite des autres sur une même ligne (fil ou piste).

→ permet des communications sur de « grandes » distances.

Bus interne : Permet de connecter les composants internes à l'ordinateur (comme le processeur et la mémoire)

Bus externe : Permet de connecter des composants externes, comme des périphériques, à la carte mère de l'ordinateur.

## Le BUS PCI (Peripheral Component Interconnect)



→ Bus local **synchrone** cadencé à une fréquence de 33 ou 66 MHz (version 2.1)

→ **Parallèle de largeur** de 32 ou 64 bits

→ **Débit** de 132 Mo/s ou 264 Mo/s

→ **Communication maître-esclave**

Exemple : Transaction de lecture

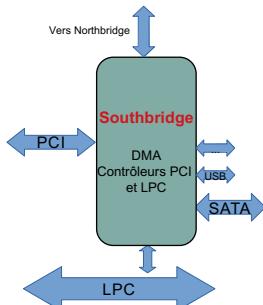
- Le maître place l'adresse des données à lire sur le bus
- Le maître passe la main à l'esclave
- L'esclave place les données à lire sur le bus



→ Utilisé pour les cartes d'extensions (réseau, son, éventuellement graphiques).

→ Deux cartes peuvent communiquer indépendamment du processeur

## Les bus ISA / et LPC



Ports ISA

### ISA (Industry Standard Architecture):

→ Bus **synchrone parallèle** de *largeur* 16 bits et cadencé à 8 MHz (moins rapide que le PCI)

→ Développé 1981 par IBM dans le cadre de la création du PC, il est devenu *presque obsolète*, remplacé par le PCI et le LPC.

→ Utilisé pour connecter des cartes d'extensions travaillant à faible cadence (cartes son [années 80-90] capteurs de température ou tension [aujourd'hui]). On le retrouve également dans des applications industrielles.



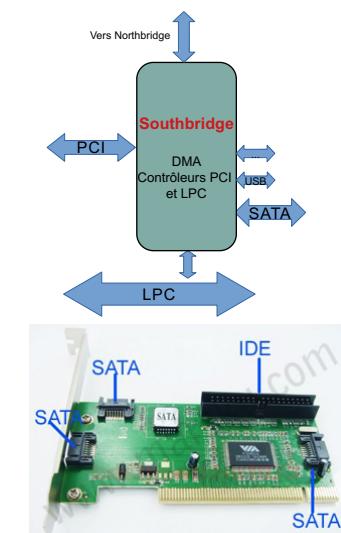
### LPC (Low Pin Count):

→ Bus **synchrone parallèle** de *largeur* 4 bits et cadencé à 33MHz (comme le PCI)

=> Limite l'encombrement sur la carte mère

→ Remplaçant du ISA pour les périphériques travaillant à faible vitesse.

## Les bus IDE et SATA



### IDE / PATA (Integrated Drive Electronics / Parallel Advanced Technology Attachment):

→ Bus **parallèle** de *largeur* 16 bits

→ Débit allant de 33 Mo/s à 133 Mo/s

→ Half-Duplex

→ Désavantages du parallèle : encombrement et perturbation sur les lignes



### SATA (Serial Advanced Technology Attachment):

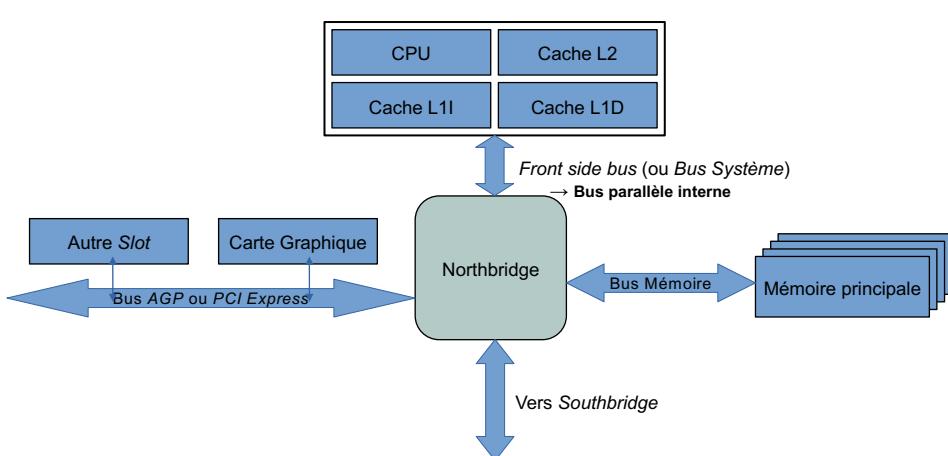
→ Bus **série** de *largeur* 4 bits

→ Débit allant de 1,5 Go/s à 6 Go/s

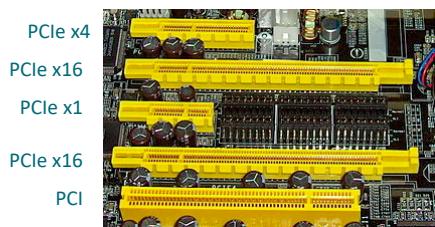
→ Half-Duplex

→ A remplacé le PATA pour remédier aux désavantages de ce dernier.

## Le Northbridge



## Les bus AGP et PCI Express



Génération	x1	x4	x8	x16
PCIe 1.1	250 Mo/s	1 Go/s	2 Go/s	4 Go/s
PCIe 2.0	500 Mo/s	2 Go/s	4 Go/s	8 Go/s
PCIe 3.0	1 Go/s	4 Go/s	8 Go/s	16 Go/s
PCIe 4.0	2 Go/s	8 Go/s	16 Go/s	32 Go/s

### Bus AGP (Accelerated Graphics Port) :

→ Bus créé en 1997 par Intel. Remplacé par le PCI Express à partir de 2004.

→ Bus **synchrone** cadencé à une fréquence de 66 MHz (basé sur le PCI version 2.1)

→ **Parallèle de largeur de 32 bits**

→ **Débit** de 266 Mo/s à 2133 Mo/s suivant les versions (de 1.0 à 3.5).

→ Nécessité d'un bus rapide et de haut débit pour réaliser des calculs graphiques. D'où sa séparation du côté sud.



### Bus PCI Express :

→ Dérive de la norme PCI => les constructeurs peuvent adapter simplement leurs cartes d'extension.

→ **Interfaces séries en parallèle** : 1, 4, 8 ou 16 lignes bidirectionnelles (*full-duplex*) en parallèle.

## Les Mémoires

Qu'est-ce qui caractérise une mémoire ?

**Capacité :** Quantité d'information qu'une mémoire peut stocker (en octets).

**Temps d'accès :** Temps nécessaire pour obtenir une information en mémoire (en nanosecondes).

**Bandé passante :** Produit de la largeur du bus de données et de la fréquence de la mémoire.

[ pour un bus de données de 64 bits et une fréquence de 100 MHz, on a une BP de 800 Mo/s ]

**Volatilité :** Temps pendant lequel une information est disponible en mémoire.

[ RAM volatile // ROM non volatile ]

Encombrement

Coût

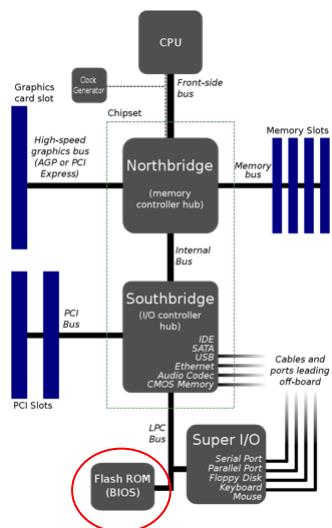
→ Besoin de mémoires aux caractéristiques très différentes. => Différentes mémoires sur un ordinateur.

## Les Mémoires

Mesurer la capacité d'une mémoire

Terminologie	Abréviation	Capacité en octets	
Kilo-octet	Ko	1 024 octets	$2^{10}$ octets
Méga-octet	Mo	1 024 Ko	$2^{20}$ octets
Giga-octet	Go	1 024 Mo	$2^{30}$ octets
Téra-octet	To	1 024 Go	$2^{40}$ octets

## Les Mémoires : Mémoires mortes



Une **mémoire morte** (*Read Only Memory [ROM]* en anglais) est une mémoire informatique **non-volatile** (qui ne s'efface pas lorsqu'elle n'est plus alimentée en électricité), dont le contenu est fixé à sa programmation et ne pouvant être modifiée.

Historiquement, le **BIOS** était stocké dans une mémoire morte qui sur la carte mère. Il s'agit d'un firmware, un programme très proche du matériel.

Il permet :

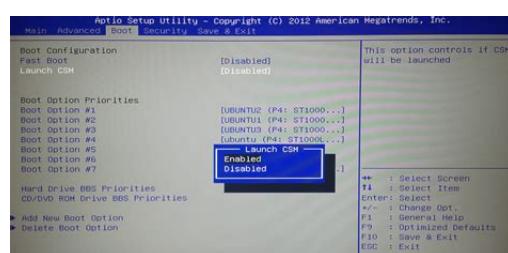
- L'initialisation des composants de la carte mère
- L'identification des périphériques connectés
- L'amorçage le système d'exploitation.

Aujourd'hui le BIOS est stocké sur une **mémoire flash**. Il s'agit d'une **mémoire de masse à semi-conducteurs réinscriptible** (comme une clé USB).

La réinscriptibilité permet la mise à jour du BIOS (rare).



## Mémoire Morte : le BIOS



### Écran de BIOS

→ Modification des paramètres de d'amorçage

→ Utile pour faire du *Multiboot*

### « CMOS Battery »

Alimente une **mémoire RAM CMOS** de quelques centaines d'octets permettant la sauvegarde des paramètres du BIOS (ordre d'amorçage, date, etc.).



## Les mémoires

### Mémoires de travail

Ce sont les mémoires qui sont actives dans l'exécution d'un programme.

- Mémoire principale (mémoire vive)
- Mémoire morte (BIOS)
- Mémoire cache (mémoire vive)
- Registres du processeur

« barrette DDR SDRAM »



### Mémoires de stockage

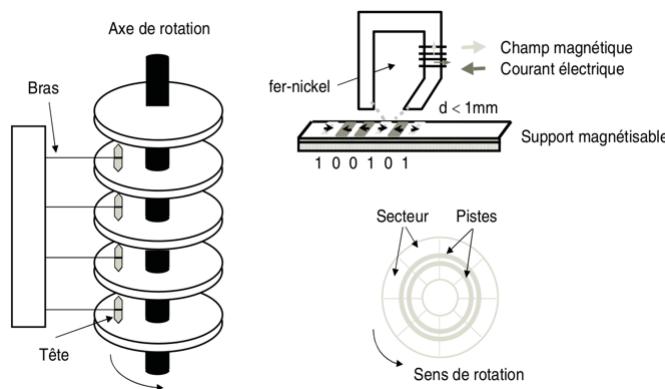
Ces mémoires permettent de stocker de manière permanente de grandes quantité de données qui ne participent pas directement à l'exécution d'un programme.

- Un programme stocké sur le disque dur est chargé en mémoire principale avant d'être exécuté.



## La mémoire de stockage

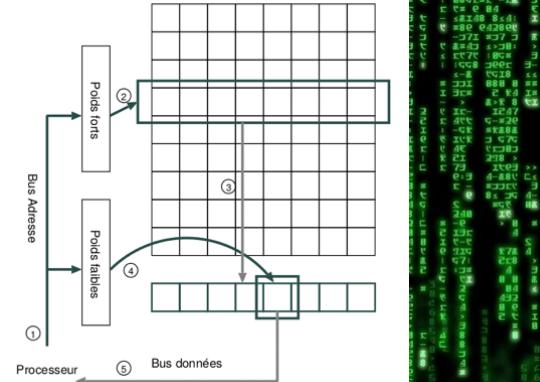
### Principe de fonctionnement d'un disque dur



## La mémoire de travail

### Fonctionnement :

- 1 : Envoi de l'adresse par le processeur
- 2 : Sélection d'une ligne avec la partie haute de l'adresse
- 3 : Lecture de la ligne
- 4 : Sélection du mot (64 bits si processeur 64 bits) dans la ligne avec la partie basse de l'adresse
- 5 : Envoi du mot au processeur via le bus de données



## Mémoire de travail VS Mémoire de stockage



VS



## Mémoire de travail : mémoire vive

### Technologie : DDR SDRAM

*Double Data Rate Synchronous Dynamic Random Access Memory*

→ Utilisé depuis les années 2000

→ Double Data Rate : Transmet les données sur front montant et descendant d'horloge (contrairement à la SRAM classique).

→ Synchronous : Attend un front d'horloge pour prendre en compte l'état des signaux d'entrées (def électronique numérique).

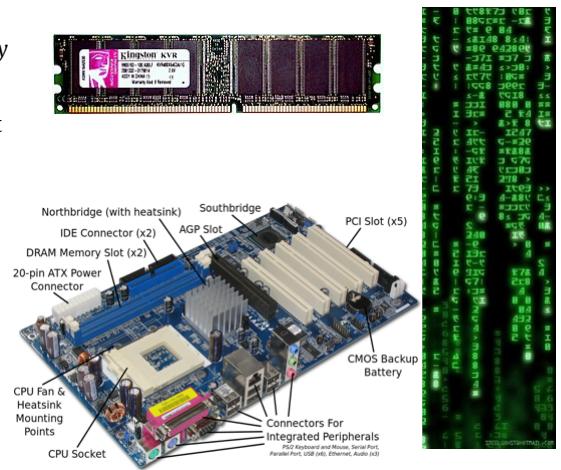
### Type DDR-200 :

→ Fréquence 100 MHz / Bande passante de 1,6 Go/s

Type DDR-600 :

→ Fréquence 300 MHz / Bande passante de 4,8 Go/s

=> Logique qu'elle soit reliée au processeur via le **Northbridge**



## Les mémoires du processeur : Registres

### Registres de processeur

→ Mémoire interne au processeur

→ Coût de fabrication élevé car la place dans un microprocesseur est limitée

→ Un processeur est en partie caractérisé par la taille de ses registres :

Processeur 32 bits => registres 32 bits

Différents registres :

→ La plupart stockent les opérandes et résultats des opérations réalisées

→ Registres d'instructions

→ Pointeur de pile

→ Compteur de programme



## Les mémoires du processeur : Caches

### Mémoire cache : Principe de localité

#### Localité temporelle :

Si une information (instruction ou donnée) est manipulée par le processeur, il est très probable qu'elle sera également utilisée peu de temps après.

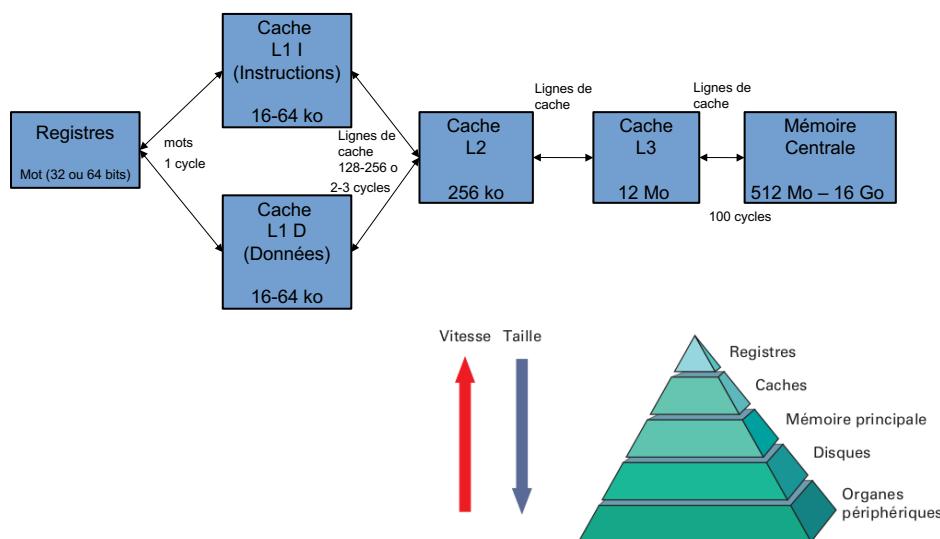
#### Localité spatiale :

Si une information est manipulée par le processeur, il est très probable qu'une information située dans un emplacement mémoire proche sera également utilisée peu de temps après.

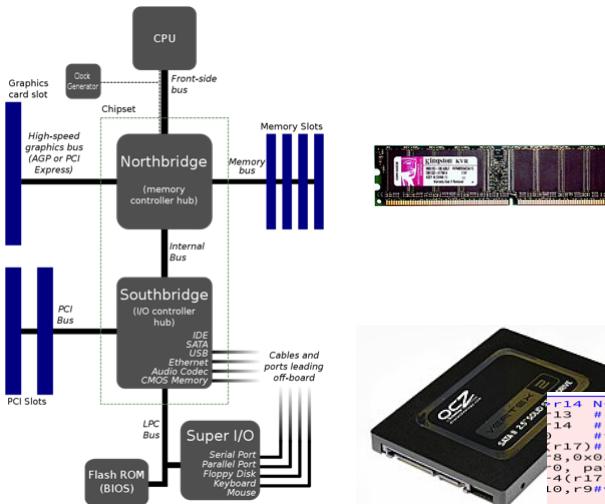
→ La mémoire cache contient des lignes de données copiées depuis la mémoire centrale.



## Les mémoires du processeur : Caches



## Chemin de l'information



```
r14 N->r15
13 #r16=@t=&t[0]
14 #r17=@v=&v[0]
9 #i=0
(r17) #r9=v[1]
8,0x01
0, pair
-4(r17) #r10=v[i-1]
lo,r9#v[i-1]-v[i]
,4(r17) #r10=v[i+1]
>,10#v[i]+v[i+1]
(r16) #range t[1]
3,1 #i++
-16,4#t++ (@t++4)
-17,4#v++ (@v++4)
15,bcl # i<n?
```

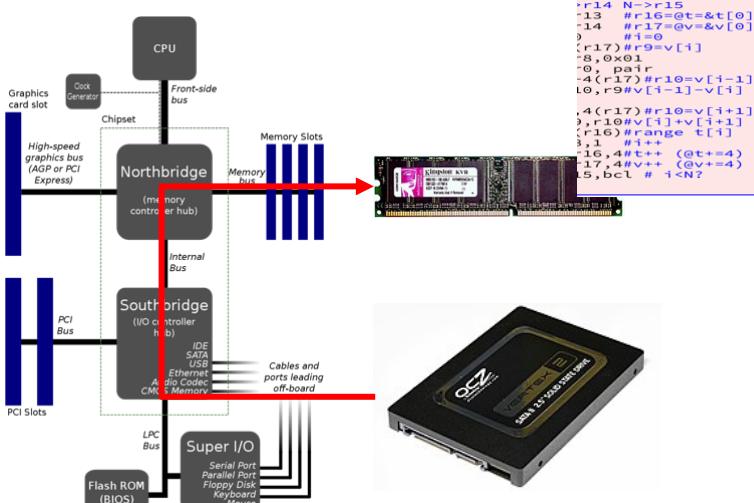


R1.08 Informatique – Julien Gabiot – 2022/2023

35

35

## Chemin de l'information



```
r14 N->r15
13 #r16=@t=&t[0]
14 #r17=@v=&v[0]
9 #i=0
(r17) #r9=v[1]
8,0x01
0, pair
-4(r17) #r10=v[i-1]
lo,r9#v[i-1]-v[i]
,4(r17) #r10=v[i+1]
>,10#v[i]+v[i+1]
(r16) #range t[1]
3,1 #i++
-16,4#t++ (@t++4)
-17,4#v++ (@v++4)
15,bcl # i<n?
```

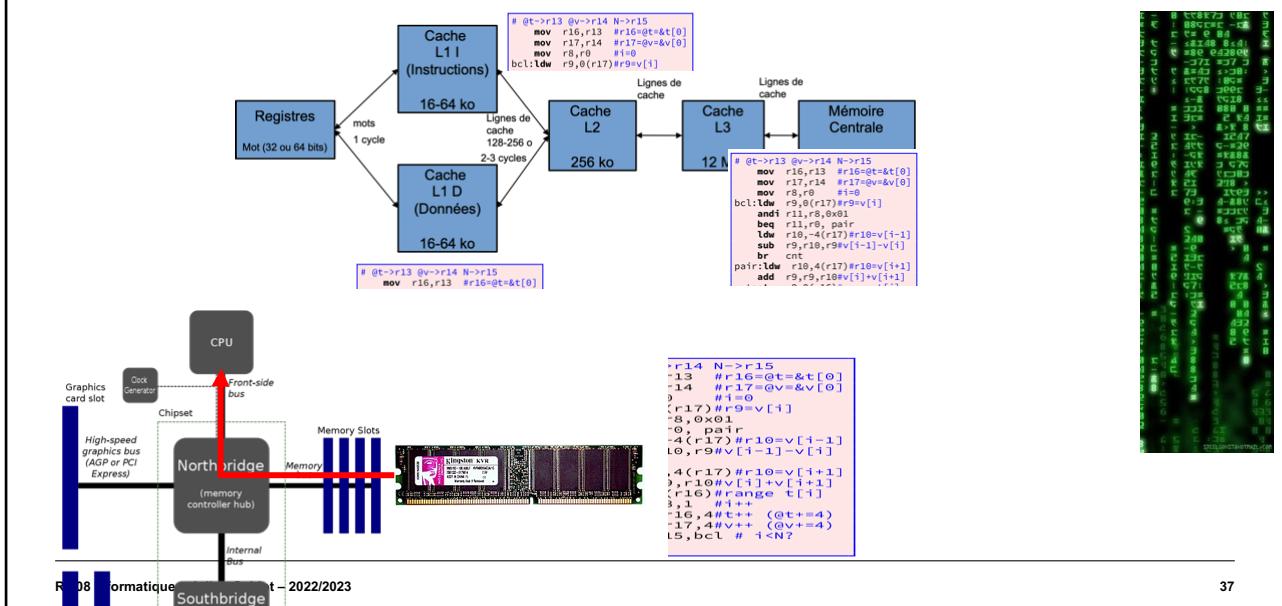


R1.08 Informatique – Julien Gabiot – 2022/2023

36

36

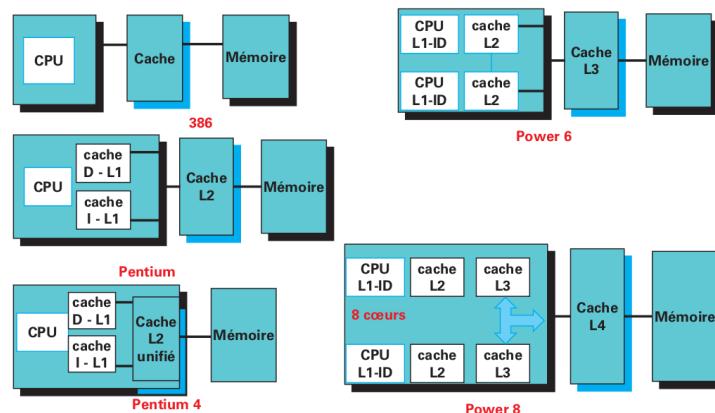
## Chemin de l'information



37

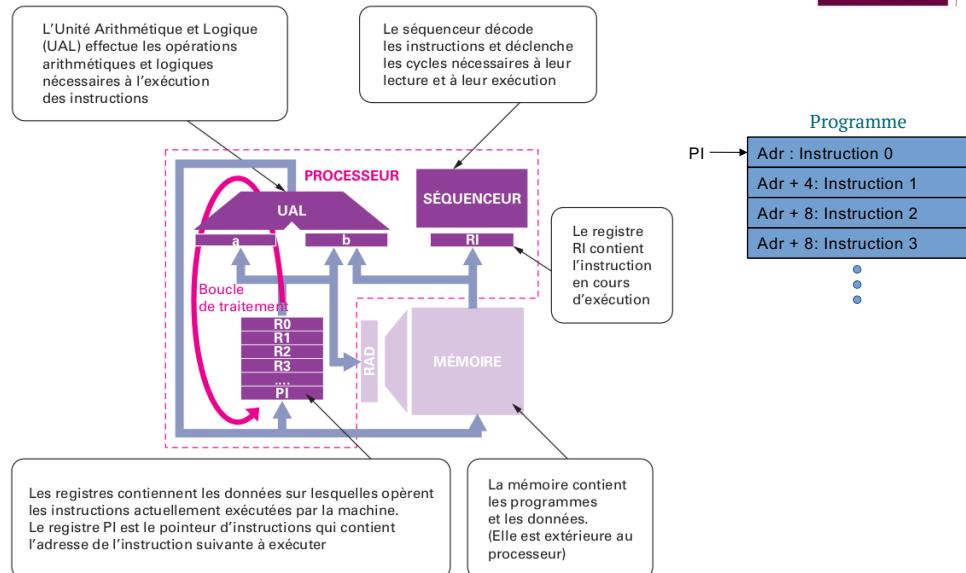
## Les processeurs : introduction

### Mémoire cache



38

## Les processeurs : introduction



R1.08 Informatique – Julien Gabiot – 2022/2023

39

39

## Langage Machine et langage Assembleur

Langage Machine : Le **contrôleur** (ou **séquenceur**) manipule les signaux de contrôle du processeur. Chaque configuration, ou chaque enchaînement de configurations, peut-être associée à un mot binaire. Ce mot est appelé une **instruction**.

→ Elle correspond à une **opération élémentaire** réalisée par le processeur (charger un mot mémoire dans un registre, ajouter 1 au contenu de ce registre, le ranger en mémoire, etc).

→ Une suite d'instruction est appelée **programme**.

Langage Assembleur : Le langage assembleur est une traduction littérale du langage machine. A chaque mot binaire est associé, de manière bijective, à un **symbole mnémoneque** rappelant de manière intelligible pour l'homme l'action réalisée.

Exemple :

- Charger un mot dans un registre (**ldw [load word]**)
- Ajouter 1 (**addi [add immediate]**)
- Stocker le contenu en mémoire (**str [store]**)

R1.08 Informatique – Julien Gabiot – 2022/2023

40

40

## Langage Machine et langage Assembleur

Boucle : ondelette de Haar

```
int N;
int t[N], v[N];

for(int i=0;i<N;i++){
    if (i%1) // i impair
        t[i]=v[i-1]-v[i];
    else      // i pair
        t[i]=v[i]+v[i+1];
}
```

```
# @t->r13 @v->r14 N->r15
    mov r16,r13 #r16=@t=&t[0]
    mov r17,r14 #r17=@v=&v[0]
    mov r8,r0 #i=0
bcl:ldw r9,0(r17)#r9=v[i]
    andi r11,r8,0x01
    beq r11,r0, pair
    ldw r10,-4(r17)#r10=v[i-1]
    sub r9,r10,r9#v[i-1]-v[i]
    br cnt
pair:ldw r10,4(r17)#r10=v[i+1]
    add r9,r9,r10#v[i]+v[i+1]
cnt:stw r9,0(r16)#range t[i]
    addi r8,r8,1 #i++
    addi r16,r16,4#t++ (@t+=4)
    addi r17,r17,4#v++ (@v+=4)
    blt r8,r15,bcl # i<N?
fin: ...
```



## Parenthèse : Espace d'échange (swap)

→ Partie de la mémoire de masse (disque dur ou SSD) utilisée par le système d'exploitation comme mémoire vive.

### Utilité :

- Lorsque la mémoire centrale est pleine
- Certaines données stockées en RAM sont peu souvent utilisées
- Certaines applications sont assez gourmandes en mémoire (ex : Libre Office)
- *Hibernation* : mise en veille prolongée. On utilise la propriété de stockage permanent de la mémoire de masse. Permet de reprendre son travail là où on en était sans consommer d'énergie via la batterie ou l'alimentation secteur.

