

# Chapitre $(00000110)_2$ : Architecture d'un ordinateur.

## Table des matières

<b>1</b>	<b>Introduction : Qu'est ce qu'un ordinateur ?</b>	<b>2</b>
1.1	Charles Babbage . . . . .	2
1.2	Ada Lovelace . . . . .	3
<b>2</b>	<b>Machine M-10</b>	<b>4</b>
<b>3</b>	<b>Les différents types de mémoires</b>	<b>6</b>
3.1	Caractéristiques d'une mémoire . . . . .	6
3.2	Qu'est-ce qu'un registre ? . . . . .	7
<b>4</b>	<b>Quelques interrogations ...</b>	<b>7</b>
<b>5</b>	<b>Architecture de Von Neumann</b>	<b>7</b>
<b>6</b>	<b>Travaux pratiques</b>	<b>9</b>
6.1	Le processeur . . . . .	9
6.2	Les mémoires . . . . .	10
6.3	Entrées-Sorties (E/S ou I/O) . . . . .	11
6.4	Les BUS . . . . .	12
6.5	Bilan . . . . .	12

# 1 Introduction : Qu'est ce qu'un ordinateur ?

## 1.1 Charles Babbage

*Machine automatique de traitement de l'information, obéissant à des programmes formés par des suites d'opérations arithmétiques et logiques. (d'après Larousse)*

En 1834, le mathématicien anglais Charles Babbage fut l'un des précurseurs de l'informatique.

Il fut le premier à énoncer le principe d'un ordinateur. Il a travaillé sur le principe d'une machine à calculer qui incorpore les cartes perforées des machines à tisser Jacquard (inventées en 1801 par le lyonnais Joseph Marie Jacquard).

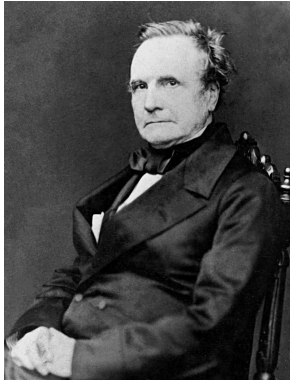


FIGURE 1 – Charles Babbage  
Image du domaine public USA



FIGURE 2 – Métier Jacquard  
Musée des Arts et Métiers [CC BY-SA]

Partant de ce principe, Charles Babbage imagine un calculateur mécanique qui recevrait instructions et données à partir de la lecture des perforations sur les fiches. Son idée était de produire des tables nautiques, astronomiques et mathématiques exactes. A l'époque, beaucoup de tables comportaient des erreurs à cause d'erreurs de recopies ou d'erreurs de calcul humain. Il passa le reste de sa vie à travailler sur les détails de cette machine analytique, tentant de la faire fonctionner.

Même s'il ne parvint pas à la construire de son vivant, elle fût bien construite à partir des plans de Babbage mais en 1991 (avec l'équivalent des technologies d'époque). Avec un peu plus de chance, réussir à construire sa machine. La théorie était bonne et fonctionnelle. L'Histoire aurait sans doute été bien différente si la machine avait vu le jour. La Machine à différences, un roman de science-fiction est construit autour de cette uchronie : que ce serait-il passé si Charles Babbage avait produit une machine à différence ?

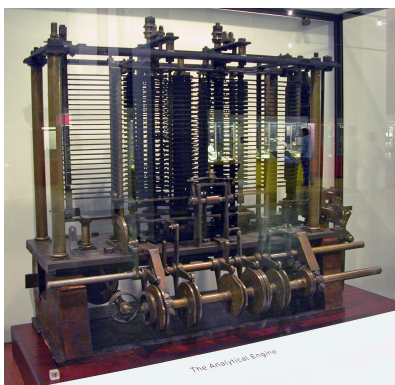


FIGURE 3 – Une partie de la machine analytique, construit par Charles Babbage, exposée au Science Museum de Londres. Musée des Arts et Métiers (photo Bruno Barral) [CC BY-SA]

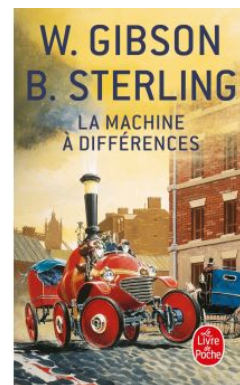


FIGURE 4 – Couverture de la Machine à Différences de William Gibson et Bruce Sterling

## 1.2 Ada Lovelace

Pendant ces nombreuses années de recherche, Charles Babbage entre en contact avec **Ada Lovelace**, de son nom complet Augusta Ada King, comtesse de Lovelace, née Ada Byron. Il s'agit de la fille du poète Lord Byron, l'un des poètes anglais les plus connus de cette époque. Elle ne connut pas son père puisqu'il quitta l'Angleterre pour toujours alors qu'elle n'avait qu'un an. Sa mère demanda la séparation et la garde de l'enfant. Éduquée par sa mère, elle-même cultivée et intelligente, Ada Lovelace parvient à devenir une jeune fille particulièrement douée en mathématiques, ce qui est était très rare à l'époque : la maîtrise des sciences étant un symbole de puissance et de contrôle, on écartait les femmes de cette connaissance. Qu'en est t-il à l'heure actuelle ?



FIGURE 5 – Ada Lovelace

Ada Lovelace est immédiatement fascinée par la machine théorique de Charles Babbage. Même après son mariage, elle continuera ses études de mathématiques, parvenant même à obtenir comme tuteur le célèbre mathématicien Auguste De Morgan. Elle étudiera la machine avec passion et parviendra à créer le premier programme de l'histoire ! En collaboration avec Charles Babbage, elle mettra au point un algorithme permettant de lancer un calcul très complexe sur la machine. Elle est aujourd'hui considérée comme la première programmeuse de l'histoire. Aucun homme avant elle n'avait conçu de programme potentiellement exécutable sur une machine réelle. Son programme comportait d'ailleurs la première boucle non bornée (TANT QUE). Son programme était donc non purement séquentiel, contrairement aux instructions envoyées aux machines Jacquard. Une autre idée révolutionnaire qu'on lui attribue : l'idée d'encoder autre chose que des nombres avec des nombres ! Maintenant que vous avez vu qu'on pouvait encoder un texte ou une image à l'aide d'une suite de nombres, cela vous paraît peut-être évident, mais comme toujours, tant que personne n'en a eu l'idée, une invention n'existe pas. Elle aurait ainsi eu l'idée d'utiliser la machine pour créer des morceaux de musique. Alternant entre haut et bas, elle tentera de créer des algorithmes permettant de gagner aux courses hippiques (pour subventionner la machine !) mais dilapida malheureusement son argent. Elle mourut jeune, à l'âge de 36 ans. Ci-dessous une illustration de son programme : image du programme (Cette œuvre est dans le domaine public dans tous les pays pour lesquels le copyright a une durée de vie de 100 ans ou moins après la mort de l'auteur). Ada Lovelace est considérée par les historiens de l'informatique comme la première personne à avoir programmé de l'histoire. On peut voir notamment son portrait sur les hologrammes d'authentification des produits Microsoft.



FIGURE 6 – hologrammes d'authentification



FIGURE 7 – Premier programme informatique non séquentiel

## 2 Machine M-10

Aujourd'hui, nous allons vous offrir un ordinateur. Un vrai. Mais en papier. M-10 est une petite machine informatique constituée :

1. d'une mémoire qui contient des données,
2. d'une unité arithmétique et logique (UAL) capable de réaliser une opération telle une addition,
3. d'un registre nommé accumulateur qui recevra le résultat des opérations réalisées par l'UAL.

La mémoire est composée de 10 « cases ». Ces cases sont notées de @0 à @9 ce qui correspond à l'adresse mémoire de la case.

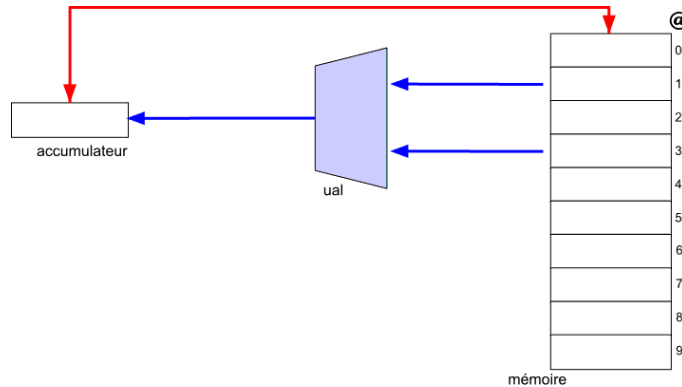


FIGURE 8 – La structure de la machine M10  
Auteur : G. Coste CC-BY-SA

Notre machine possède deux types d'instructions :

1. les instructions arithmétiques et logiques réalisées par l'UAL (+, -, ×, ET, OU, ...)
2. les instructions de transfert entre la mémoire et l'accumulateur.

Voyons une première instruction qu'on notera **ADD** et qui permet de faire une addition.

Un tout petit programme : **ADD @1 @2**. Que fait-il en détail ?

1. Il commande à l'UAL de faire une addition. Pour cela :
2. Il place le contenu de la case-mémoire 1 à l'entrée de l'UAL.
3. Il place le contenu de la case-mémoire 2 sur l'autre entrée de l'UAL.
4. Il additionne les deux entrées
5. Il place le résultat dans l'accumulateur

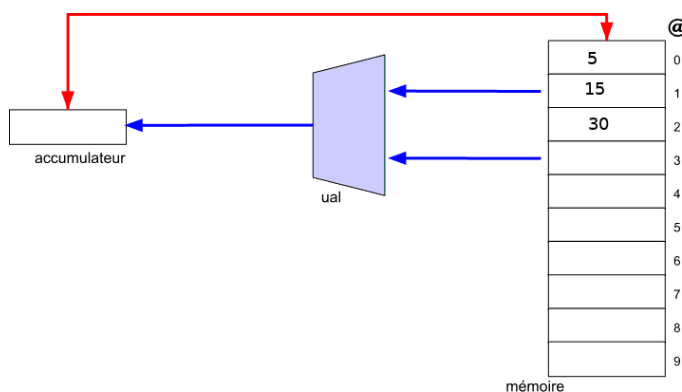


FIGURE 9 – Auteur : G. Coste CC-BY-SA

### Exercice n° 1

Que contient l'accumulateur après exécution du programme (**ADD @1 @2**) ?

Une fois le résultat dans l'accumulateur, comment parvenir à le placer en mémoire ?

Et bien imaginons une instruction **STR @3** (STR pour STORE from REGISTER) qui permet de placer le contenu de l'accumulateur dans la case-mémoire 3.

D'ailleurs, imaginons aussi l'inverse : **LDR @5** (LDR pour LOAD to REGISTER) qui permet de placer le contenu de la case-mémoire 5 dans l'accumulateur.

### Exercice n°2

Exécuter le programme suivant en utilisant la mémoire initiale représentée ci-dessous. Que contiennent les mémoires à la fin ?

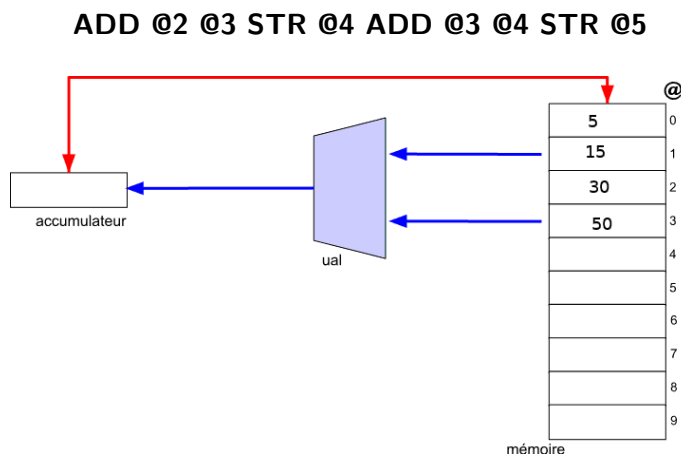


FIGURE 10 – Auteur : G. Coste CC-BY-SA

### Exercice n°3

Faisons une liste des instructions disponibles : 1. **ADD @2 @3** : additionne les contenus des zones-mémoires et place le résultat dans l'accumulateur 2. **SUB @2 @3** : Effectue la soustraction @2 - @3 et place le résultat dans l'accumulateur 3. **STR @5** : copie le contenu du registre-accumulateur en zone-mémoire 5. 4. **LDR @5** : copie le contenu de la zone-mémoire 5 dans l'accumulateur.

1. Exécuter le programme suivant en utilisant la mémoire initiale représentée ci-dessous. Que contiennent les mémoires à la fin ?

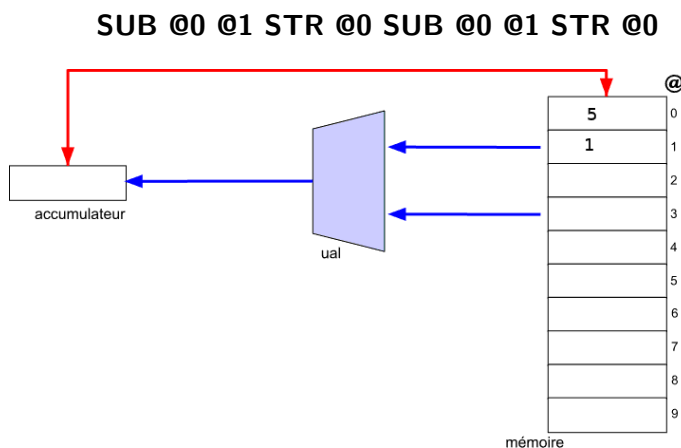


FIGURE 11 – Auteur : G. Coste CC-BY-SA

2. Le problème est que l'accès à la mémoire centrale est plutôt lent. Or ici, on ne fait que stocker le résultat pour aller ensuite le reprendre... On pourrait au contraire utiliser une instruction utilisant le résultat contenu dans l'accumulateur (identifié avec une adresse du type @A ?) pour aller directement le placer à l'entrée de l'UAL.

Exécuter le programme suivant en utilisant la mémoire initiale représentée ci-dessous. Que contiennent les mémoires à la fin ? Vous devriez constater qu'on effectue une somme, mais en ne faisant qu'un seul appel à la mémoire centrale à chaque étape.

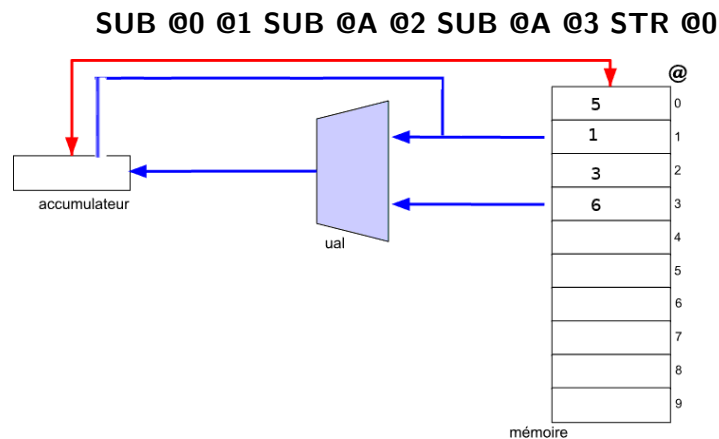


FIGURE 12 – Auteur : G. Coste CC-BY-SA

### 3 Les différents types de mémoires

#### 3.1 Caractéristiques d'une mémoire

Une « mémoire » est un composant électronique capable de stocker temporairement des informations  
 Une mémoire est caractérisée par :

1. Sa capacité, représentant le volume global d'informations (en bits) que la mémoire peut stocker.
2. Son temps d'accès, correspondant à l'intervalle de temps entre la demande de lecture/écriture et la disponibilité de la donnée.
3. Sa persistance :
  - **Mémoire volatile** : le contenu de la mémoire n'existe que si il y a une alimentation électrique (typiquement les mémoires caches et mémoire centrale)
  - **Mémoire permanente**, de masse : mémoire de grande capacité dont le contenu demeure même sans alimentation électrique (typiquement le disque dur)

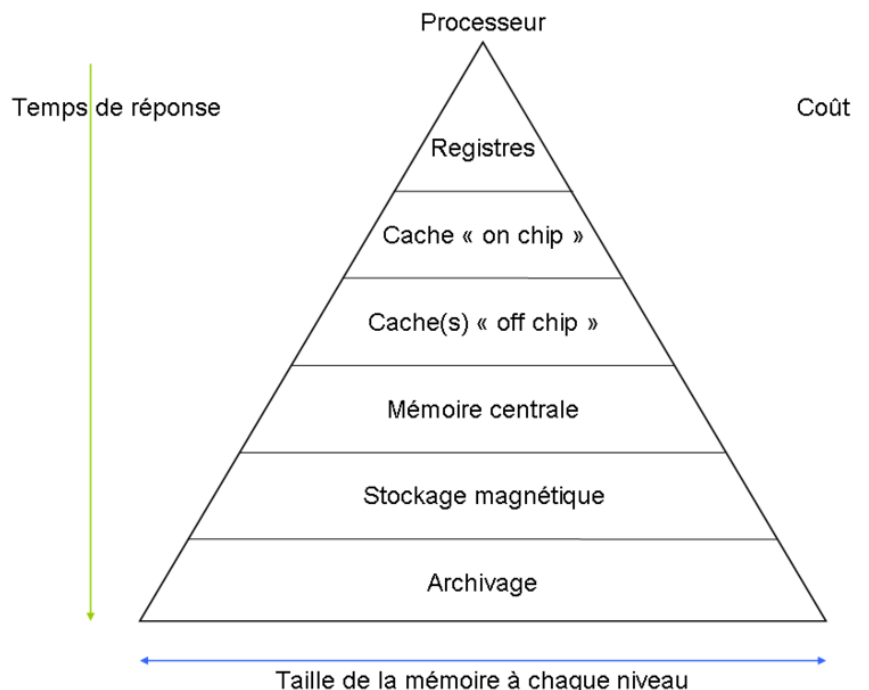


FIGURE 13 – Par Original téléversé par Poil sur Wikipédia français. Auteur : Stéphane 3 avr 2005  
 CC BY-SA 3.0

### 3.2 Qu'est-ce qu'un registre ?

Il s'agit d'une zone-mémoire interne au processeur. L'accès à cette mémoire est donc très rapide puisqu'elle est au sommet de la hiérarchie des mémoires. Il s'agit d'une zone-mémoire qui est physiquement très proche du processeur.

Dans les architectures qualifiées de load-store, les programmes transfèrent d'abord des données de la mémoire centrale (grand stockage mais peu rapide) vers des registres (faible stockage mais rapide), puis effectuent des opérations sur ces registres, et enfin transfèrent le résultat en mémoire centrale.

On voit donc apparaître plusieurs zones de stockage différentes :

- Registres : les fameuses mémoires proches du processeur
- Caches on-chip : des zones mémoires assez rapides situées directement sur la carte à puce (chip) du microprocesseur. On y stocke les résultats précédents de façon à pouvoir les retrouver sans avoir à tout recalculer.
- Caches off-chip : des zones mémoires assez rapides situées en dehors de la carte du processeur lui-même.
- Mémoire centrale : une mémoire bien plus grande mais également beaucoup plus longue en terme d'accès.

## 4 Quelques interrogations ...

Il reste toujours pas mal de questions en suspens. Quatre exemples :

1. Où sont stockés les programmes là-dedans ? Ça doit être forcément dedans puisque le programme fait partie de la machine. Mais où ?
2. Comment fait la machine pour faire une addition ?
3. Si on veut récupérer des données externes (clavier ?), on récupère à partir de quelle provenance ?
4. Si on veut envoyer des informations vers l'extérieur (écran ?), on envoie vers quelle destination ?

## 5 Architecture de Von Neumann

John Von Neumann (1903-1957) est un mathématicien et physicien américano-hongrois. Il a apporté de nombreux travaux autant en physique quantique, en mathématiques qu'en économie. Sa contribution à l'informatique est immense puisque tous les ordinateurs utilisent l'architecture qu'il a établie il y a plus de 70 ans.

Nous nous étions demandé où stocker les instructions de la machine M10. La réponse de Von Neumann est de placer les données et les programmes au même endroit. C'est pourquoi ce type d'ordinateur se nomme parfois aussi ordinateur à programme enregistré.



FIGURE 14 – Image de Los Alamos National Laboratory.

La société permet l'utilisation de cette photo sous condition d'en préciser le propriétaire.

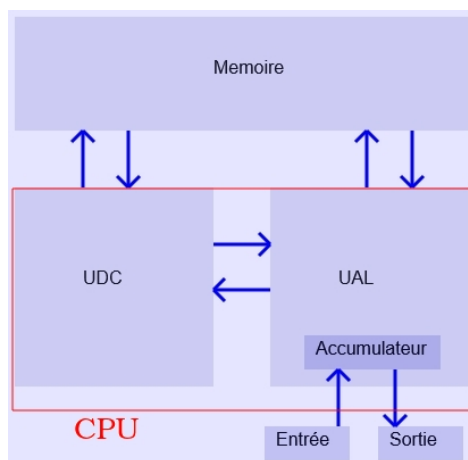


FIGURE 15 – premier schéma de principe  
Auteur : G. Coste CC-BY-SA

On voit donc que les zones-mémoires peuvent envoyer ou recevoir des informations à destination :

- de l'Unité Arithmétique Logique (UAL), comme avec notre M10
- mais également vers une Unité de Contrôle (UDC) qui va servir à interpréter les ordres.

Notre ordinateur dispose donc :

1. D'une zone mémoire dite mémoire vive (RAM) qui permet de stocker temporairement les données et les instructions. On dit qu'il s'agit d'une mémoire volatile (qui disparaît lorsqu'on coupe l'alimentation)
2. Le processeur également nommé UDT (Unité de Traitement) ou CPU (central processing unit) en anglais. Il est composé de l'UDC (unité de contrôle) et l'UAL (unité arithmétique et logique, qui contient au moins un registre nommé accumulateur)
3. Les flèches représentent, elles, ce qu'on nomme les bus parallèles : un bus de 64 bits est constitué de 64 « fils électriques » qui permettent de faire transiter 64 bits simultanément. Si on doit transporter 128 bits avec des bus de 64 bits, il faudra donc faire 2 opérations de transport, là où un seul transport suffit avec un bus de 128 bits. On dispose d'ailleurs de 3 types de bus :
  - Les bus d'adresses permettent de transmettre les adresses mémoires voulues
  - Les bus de données permettent de transférer les données situées qu'on trouve à partir de l'adresse fournie
  - Les bus de contrôle permettent de transférer les bits correspondants aux informations nécessaires à la gestion des instructions

#### Exercice n° 4

---

1. On part du principe que le système doit pouvoir transporter en une seule opération une adresse via son bus d'adresses
  - a. Combien d'adresses-mémoires RAM différentes peut-on avoir dans un ordinateur dont le bus d'adresse est un bus 16 bits ?
  - b. Si on considère que chaque case mémoire correspond à un octet, quelle est la mémoire vive maximale disponible sur ce système s'il ne disposant pas d'autres manières d'adresser sa mémoire ?
2. Faire de même pour un ordinateur muni d'un processeur 32 bits, et de bus d'adresses de 32 bits.

#### Exercice n° 5

---

Votre disque dur ou votre carte SD correspondent-ils à la mémoire vive ?

#### Exercice n° 6

---

1. Combien d'opérations pour stocker ou lire un grand entier stocké sur 4 octets avec un bus de données de 8 bits connaissant l'adresse mémoire du premier octet ?
2. Combien d'opérations pour stocker ou lire un grand entier stocké sur 4 octets avec un bus de données de 32 bits connaissant l'adresse mémoire du premier octet ?
3. La même lecture d'un entier sur 4 octets va-t-elle être plus rapide avec un bus de données 64 bits ?
4. Dans quel cas le bus de données 32 bits va-t-il alors être plus efficace ?



## 6 Travaux pratiques

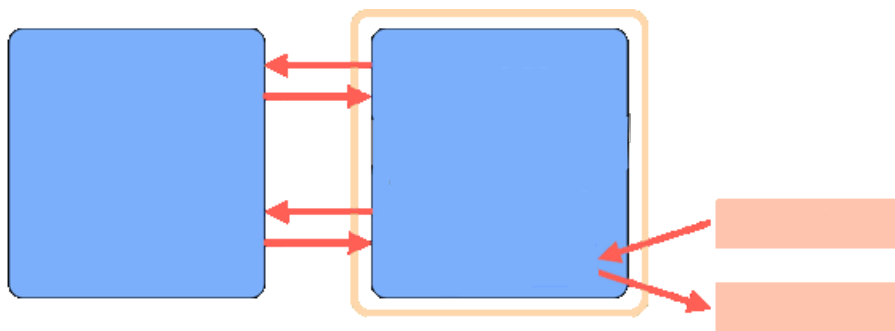
### Exercice n°7

---

Compléter le schéma ci-dessous représentant une architecture dite de « Von Neumann » :

On distingue trois éléments :

- Le processeur (CPU) : élément permettant d'exécuter les instructions d'un programme
- La mémoire : éléments permettant de stocker les données.
- Les entrées et sorties : éléments permettant la communication entre l'ordinateur et l'être humain appelés périphériques.



### 6.1 Le processeur

Le processeur (CPU, pour Central Processing Unit) est le cerveau de l'ordinateur. Il permet de manipuler, des données et des instructions codées sous forme binaires. Le processeur est composé de millions de transistors placés dans un boîtier comportant des connecteurs d'entrée-sortie, surmonté d'un ventilateur. C'est un circuit électronique cadencé au rythme d'une horloge interne qui envoie des impulsions.

**Remarque :** La fréquence d'horloge, correspond nombre d'impulsions par seconde. Elle s'exprime en **Hertz** (Hz). Pour un ordinateur à 2 GHz, l'horloge envoie 2 000 000 000 impulsions par seconde. Cette horloge permet notamment au processeur de basculer constamment d'un processus à l'autre sans que l'utilisateur ne s'en aperçoive.

### Exercice n°8

---

Tester les commandes shell :

- Quelques informations sur le processeur : `lscpu`
- Affiche en continu les informations relatives aux processus : `top`

**Remarques :** chaque processus est identifié par un numéro PID.

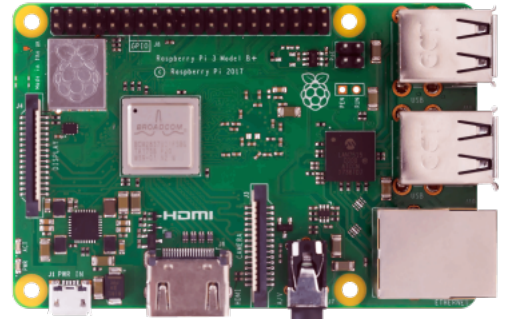
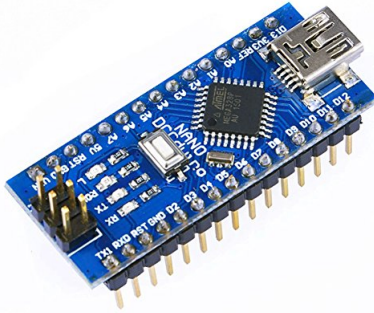
### Exercice n°9

---

1. Quel processeur équipe votre machine ?
2. Donner sa fréquence maximale.
3. A quelle fréquence est-il en train de travailler ?
4. A quoi sert l'horloge ?

## Exercice n° 10

Sur les photos ci-dessous, identifier le processeur.



## Exercice n° 11

La commande `lscpu` permet d'avoir quelques informations sur la fréquence du processeur. Répondre aux questions, à l'aide de la capture d'écran ci-contre.

1. Citer la marque du processeur
2. A quelle fréquence le processeur est-il en train de travailler ?
3. Combien d'impulsions par seconde le processeur est-il capable de faire ?

```
(base) fourlegnie@fourlegnie:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):      32-bit, 64-bit
Byte Order:          Little Endian
CPU(s):              2
On-line CPU(s) list: 0,1
Thread(s) per core: 1
Core(s) per socket: 2
Socket(s):           1
NUMA node(s):       1
Vendor ID:           GenuineIntel
CPU family:          6
Model:               58
Model name:          Intel(R) Pentium(R) CPU 2020M @ 2.40GHz
Stepping:            9
CPU MHz:             1197.210
CPU max MHz:         2400.0000
CPU min MHz:         1200.0000
BogoMIPS:            4788.78
Virtualization:      VT-x
L1d cache:           32K
```

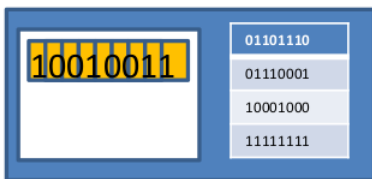
Sources :

- <https://interstices.info/le-modele-darchitecture-de-von-neumann/>
- <http://deptinfo.cnam.fr/new/spip.php?rubrique714>

## 6.2 Les mémoires

Mémoires vives : mémoires **volatiles** :

Mémoires de masse :  
mémoires **permanentes**



**REGISTRES**  
N bits (32, 64)  
1 nanoseconde

**Mémoires Caches**  
Koctets (L1,L2)  
5 nanosecondes

**Mémoires Centrales**  
Goctets  
10 nanosecondes

**Mémoires de masse**  
500 Goctets - Toctets  
5 millisecondes

Au plus près du cpu

Capacité, vitesse

Au plus loin du cpu

## Exercice n° 12

### Tester les commandes shell :

Tester les commandes suivantes et identifier la capacité des mémoires caches, centrales et de masse.

`free -b` ; `free -k` ; `free -g` ; `df -h`



## Exercice n° 13

Citer trois caractéristiques d'une mémoire.

## Exercice n° 14

1. Lire la page : <https://fr.wikipedia.org/wiki/Octet>

2. Montrer que l'affichage donné par Vista n'est pas correct :

Type :	Disque local		
Système de fichiers :	NTFS		
 Espace utilisé :	85 455 966 208 octets	79,5 Go	
 Espace libre :	65 989 451 776 octets	61,4 Go	
	151 445 417 984 octets	141 Go	

3. Tester les commandes : `free --kibi` ; `free --kilo`.

4. Justifier que les affichages encadrés sont cohérents.

```
(base) fourlegnie@fourlegnie:~$ free --kilo
              total        used        free      shared  buff/cache   available
Mem:          8235986        2515660        817790        451698        4902535        4955455
Swap:         2147479           11341        2136137
(base) fourlegnie@fourlegnie:~$ free --kibi
              total        used        free      shared  buff/cache   available
Mem:          8042956        2457692        797708        441028        4787556        4838404
Swap:         2097148           11076        2086072
(base) fourlegnie@fourlegnie:~$
```

## 6.3 Entrées-Sorties (E/S ou I/O)

Un périphérique est un matériel électronique pouvant être raccordé à un ordinateur par l'intermédiaire de l'une de ses interfaces d'entrée-sortie (interface VGA, HDMI, USB, RJ45.), le plus souvent par l'intermédiaire d'un connecteur. L'interface d'entrées-sorties est pilotée par un driver (pilote d'entrées-sorties)

On distingue habituellement les catégories de périphériques suivantes :

- périphériques de sortie : ce sont des périphériques permettant à l'ordinateur d'émettre des informations vers l'extérieur, tels qu'un écran, une imprimante..
- périphériques d'entrée : ce sont des périphériques capables uniquement d'envoyer des informations à l'ordinateur, par exemple la souris, le clavier, etc.
- périphériques d'entrée-sortie : ce sont des périphériques capables d'envoyer des informations à l'ordinateur et permettant également à l'ordinateur d'émettre des informations vers l'extérieur, par exemple la carte wifi, le bluetooth, le disque dur, etc

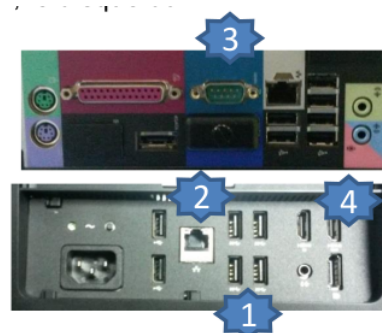
## Exercice n° 15

### Tester les commandes shell :

A l'aide de la commande `lshw -short` donner quatre périphériques.

## Exercice n° 16

Ci-contre, retrouver les interfaces RJ45, VGA, HDMI et USB.



## 6.4 Les BUS

Le bus informatique est la réunion des parties matérielles et immatérielles permettant à différents composants de l'ordinateur de s'échanger des informations.

Voici quelques exemples :

- Bus système (Front Side Bus FSB) permet la communication entre le processeur et la mémoire centrale.
- bus PCI, USB, bus série, ...

### Exercice n° 17

#### Tester les commandes shell :

A l'aide de la commande `lshw -short` identifier quelques bus.

## 6.5 Bilan

Placer les différents composants de l'ordinateur et expliquez leur rôle :

- Processeur
- Horloge
- Mémoire cache
- Mémoire centrale
- Bus
- Entrées
- Sorties

