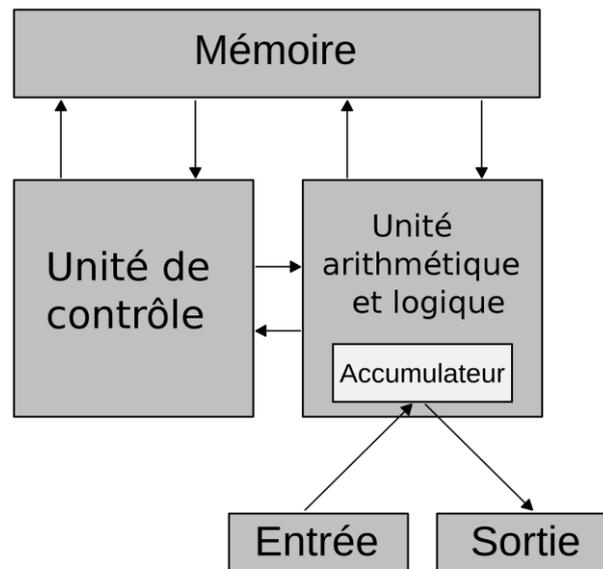


Chapitre 1: Introduction à ASD (Algorithmique et structure de données 1)

Dans ce chapitre on va représenter brièvement l'architecture de la première machine (machine de **Von Neumann**), comme on va faire une généralité sur l'algorithmique et la programmation en C.

1. L'architecture de Von Neumann

L'**architecture de Von Neumann** est un modèle structurel d'ordinateur dans lequel une unité de stockage (mémoire) unique sert à conserver à la fois les instructions et les données demandées ou produites par le calcul. Les ordinateurs actuels sont tous basés sur des versions améliorées de cette architecture.



L'architecture de *Von Neumann* décompose l'ordinateur en 4 parties distinctes :

- l'**unité arithmétique et logique** (UAL ou ALU en anglais) ou **unité de traitement** : son rôle est d'effectuer les opérations de base ;
- l'**unité de contrôle** ou **de commande** (*control unit*), chargée du « séquençage » des opérations ;
- la **mémoire** qui contient à la fois les données et le programme qui indiquera à l'*unité de contrôle* quels sont les calculs à faire sur ces données ;
- les **dispositifs d'entrée-sortie**, qui permettent de communiquer avec le monde extérieur.

1.1. Principaux composants

Le microprocesseur



Intel CPU Core i7 6700K

Le **microprocesseur** (ou unité centrale de traitement, UCT, en anglais *Central Processing Unit*, **CPU**) est un composant essentiel qui exécute les instructions machine des programmes informatiques.

Remarque : il est le plus souvent amovible, placé sur un support appelé **socket**, et équipé d'un **radiateur** et d'un **ventilateur** (c'est le composant de la carte mère le plus gourmand en énergie).

Il est schématiquement constitué de 3 parties :

- l'**unité arithmétique et logique** (ALU) est chargée de l'exécution de tous les calculs que peut réaliser le microprocesseur :
 - opération arithmétiques (sur les nombres entiers ou flottants)
 - opérations logiques (sur les bits) ;
- les **registres** permettent de mémoriser de l'information (donnée ou instruction) au sein même du CPU, en très petite quantité ;
- l'**unité de contrôle** permet d'exécuter les instructions (les programmes) et fait la récupération des instructions et des données pour les envoyer à l'ALU.

La mémoire

La **mémoire** permet de stocker des données et des programmes.

La mémoire se divise entre :

- mémoire **volatile (RAM)** contenant les programmes et données en cours de fonctionnement) : cette mémoire perd son contenu lorsqu'elle n'est plus alimentée.
- mémoire **permanente (HDD, SSD, ...)** contenant les programmes et données de base de la machine : cette mémoire ne le disparaît pas à la mise hors tension.

Dans la plupart des mémoires, les informations sont classées par **adresses** : chaque octet est accessible par une adresse unique.

Chapitre 1: Introduction à ASD (Algorithmique et structure de données 1)

Pour des raisons économiques, les mémoires sont en général divisées en plusieurs familles :

Une mémoire de masse ou mémoire de stockage



HDD



SSD

sert à stocker à **long terme** des **grandes quantités d'informations**. Les technologies les plus courantes de mémoires de masse sont *électromécaniques* (disques durs – HDD) ou à *semi-conducteurs* (SSD, clefs USB, ...), elles visent à obtenir une capacité de stockage élevée à faible coût et ont généralement une **vitesse inférieure** aux autres mémoires.

ordres de grandeur :

- *capacité : jusqu'à 10 To (HDD)*
- *vitesse : jusqu'à 500 Mo/s (SSD)*

La mémoire vive ou RAM



SDRAM

C'est l'espace principal de stockage du microprocesseur, mais il s'agit d'une mémoire volatile.

Appelée **RAM** (*Random Access Memory*) car le temps d'accès aux données est faible et ne dépend pas de l'emplacement.

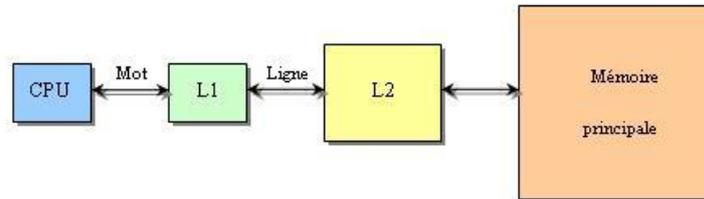
ordres de grandeur :

Chapitre 1: Introduction à ASD (Algorithmique et structure de données 1)

- *capacité : jusqu'à 32 Go*
- *vitesse : jusqu'à 2 Go/s*

Une mémoire cache

La RAM est rapide, mais le microprocesseur l'est encore plus ! Afin de ne pas limiter ses performances en l'obligeant à attendre (on parle de goulot d'étranglement), on utilise de petites unités de mémoires, beaucoup plus rapides, mais nettement plus chères !



Différents niveaux de mémoire

Ses mémoires **cache** servent à conserver un court instant des informations fréquemment consultées. Elles ont une très grande vitesse, et un coût élevé pour une faible capacité de stockage.

Ordres de grandeur :

- *capacité : quelques ko (L1) à quelques Mo (L2)*
- *vitesse : jusqu'à 5 Go/s*

Le registre de processeur

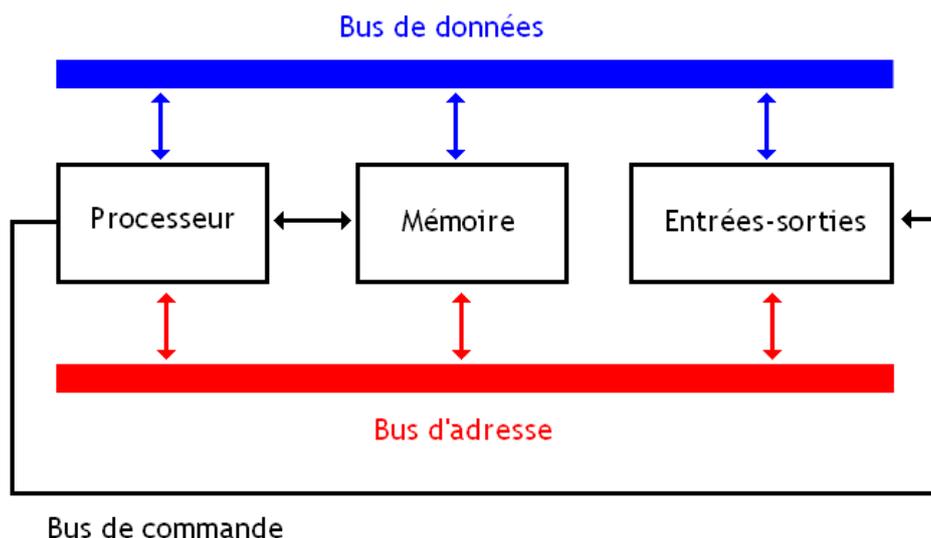
Il est intégré au processeur. Ce type de mémoire est très rapide mais aussi très cher et est donc réservé à une très faible quantité de données.

Ordres de grandeur :

- *capacité : quelques dizaines d'octets*
- *vitesse : jusqu'à 30 Go/s*

Les bus

Chapitre 1: Introduction à ASD (Algorithmique et structure de données 1)



Pour que les données circulent entre les différentes parties d'un ordinateur (mémoire, CPU et les entrées/sorties), il existe des systèmes de communication appelés **bus**. Il en existe de 3 grands types :

- Le **bus d'adresse** permet de faire circuler des adresses *par exemple l'adresse d'une donnée à aller chercher en mémoire* ;
- Le **bus de données** permet de faire circuler des données ;
- Le **bus de contrôle** permet de spécifier le type d'action *exemples : écriture d'une donnée en mémoire, lecture d'une donnée en mémoire*.

2.1. Les circuits logiques

Dans les **microprocesseurs**, les **opérations booléennes** sont réalisées par transistors formant ce que l'on appelle des **circuits logiques**.

Un circuit logique prend en entrée et donne en sortie des signaux électriques à deux états (HAUT et BAS). Il existe deux catégories de circuit logique :

- les circuits **combinatoires** (les états en sortie dépendent uniquement des états en entrée)
- les circuits **séquentiels** (les états en sortie dépendent des états en entrée ainsi que du temps et des états antérieurs)

On représente les circuits logiques à l'aide de schémas appelés **logigrammes**.

2. Algorithmique

Y a deux constructeurs pour mettre en œuvre d'un programme (Algorithme), le côté **matériel** ou **Hardware** comme on a vu précédemment, et le côté de la programmation ou **software**.

Il existe pas mal nombre des exemples pour présenter cette-là. Telles que : les étapes à suivre pour cuisiner quelque choses, la suite des opérations pour inscrire à la première année

Chapitre 1: Introduction à ASD (Algorithmique et structure de données 1)

universitaire, les travaux à faire (paperasse, construction et gestion de projet) pour construire une cité des bâtiments... etc.

Les instructions dans un programme doivent être simples et classer par un ordre exact, **par exemple** : si on va résoudre une équation de deuxième degré sous forme $ax^2+bx+c=0$, il faut savoir les valeurs de (a, b, c) puis calculer **Delta** puis chercher des solutions selon **Delta**. On ne peut pas modifier la séquence des opérations, sauf s'il n'y a pas de relations entre elles. Par exemple : pas de relation entre (râper d'oignons et râper de tomates) donc pas nécessaire si on râpe d'oignons premièrement ou deuxièmement.

Dans le deuxième chapitre on va voir comment construire un algorithme et un programme C, et comment créer une séquence des instructions pour faire des affichages des messages et des calculs.

Contents

1.	L'architecture de Von Neumann	1
1.1.	Principaux composants.....	2
	Le microprocesseur	2
	La mémoire	2
	Les bus	4
2.1.	Les circuits logiques.....	5
2.	Algorithmique	5