

Cours

1.1 La contribution de Turing à l'avènement des ordinateurs

1.1.1 Le binaire

Contrairement à Babbage, Turing se rend compte que les opérations arithmétiques ordinaires peuvent être traduites dans le cadre de la *logique booléenne*, sous forme de circuits logiques (booléens). Il conçoit comme première machine un *multiplicateur booléen* électrique.

Il prolonge l'idée de Gödel de coder les opérateurs, les instructions, et les fonctions (des *propositions*) par des nombres entiers.

Traiter des *instructions* revient à calculer sur des *nombres entiers*, écrits en *binaires*.

1.1.2 Les algorithmes

Le raisonnement mathématique est une combinaison :

- d'intuition
- d'ingéniosité, de procédures

L'idée de Turing est de diminuer la part d'intuition et d'augmenter celle de procédure.

Par exemple, lorsque l'on réalise une opération de tri, on le fait d'abord de manière intuitive. Pour qu'une machine soit capable de le réaliser, il faudra en dégager une *procédure*.

1.1.3 Machine de Turing

C'est une machine mathématique permettant la manipulation réglée de signes. Son rôle est de transformer les symboles fournis en entrée, en symboles lisibles en sortie.

Cette machine utilise les instructions d'un programme (une table d'états) et repère un état d'un registre, états qui définissent les étapes de l'algorithme.

C'est une machine conceptuelle, qui a permis de définir ce qu'est un problème *calculable*.

Une machine de Turing est conçue pour résoudre une classe de problèmes donnés : si une machine de Turing est programmée pour résoudre une opération d'addition, alors elle pourra résoudre TOUTES les opérations d'addition, quelles que soient les données en entrée.

Une machine de Turing particulière contient donc, par avance, un programme.

Une machine de **Turing universelle** est capable de *résoudre TOUS les problèmes possibles* de toutes les machines de Turing possibles (contient tous les programmes existants ou à venir).

John Von Neumann fait référence de manière explicite à ce concept lorsqu'il présente l'architecture à laquelle doivent se conformer les *machines universelles* :

Un ordinateur se définit comme une instance matérielle, concrète, d'une Machine de Turing Universelle

1.2 Architecture Von Neumann

Definition : Un **ordinateur** est un **appareil programmable** qui stocke son **programme** dans une **mémoire modifiable**.

John Von Neumann (1945) est le *concepteur* de l'architecture de la *machine universelle* et de ses différents dispositifs :

- processeur
- mémoires
- dispositifs d'entrée/sortie

Il est à l'origine du concept de **programme enregistré** : le stockage des données et des instructions sous forme d'impulsions électriques à l'intérieur même de la machine qui pourra les consulter à l'instant et à la vitesse qui lui convient. En remplacement du traitement mécanique alors utilisé, ce qui nécessitait une vaste manipulation de câbles et de boutons à placer sur la bonne position pour spécifier le nouveau problème. Charles Babbage avait déjà eut cette idée un siècle plus tôt.

Cette architecture s'oppose à celle de Harvard (mémoires différentes pour les données et les programmes) du Mark1.

Von Neumann est également le concepteur de l'ordinateur **IAS** à Princeton.

Le stockage était alors réalisé par des tubes cathodiques, chacun de 1024 bits.

Dans l'architecture de Von Neumann, un ordinateur est constitué de quatre parties distinctes :

1. Le **CPU** : Central Processing Unit (unité centrale de traitement) appelé aussi processeur ;
2. La **mémoire** où sont stockés les données et les programmes. La mémoire se divise entre :
 - une mémoire volatile (**RAM** ou mémoire vive) qui gère les programmes et les données
 - une mémoire permanente (**ROM**) qui gère les programmes de base de la machine.
3. des **bus** qui sont des fils conduisant des impulsions électriques et qui relient les différents composants ;
4. des **entrées-sorties** (E/S ou I/O input/Output) pour échanger avec l'extérieur.

Les échanges entre la mémoire et les registres du CPU se font via les bus selon une chronologie séquencée par l'horloge et suivant la nature des échanges : données ou adresse. Cette horloge est externe au processeur.

Un **programme** est une suite d'instructions codées sous forme binaire. Il précise la suite d'instructions exécutées par l'ordinateur. Ces instructions, codées en binaire peuvent être exprimées pour un humain en langage *assembleur*, le langage de *plus bas niveau* (une traduction mnémorique du binaire).

1.2.1 Le processeur CPU

L'unité de traitement est un circuit qui s'occupe de faire des calculs et de manipuler l'information provenant des entrées-sorties ou récupérée dans la mémoire. Dans les ordinateurs, l'unité de traitement porte le nom de processeur, ou encore de Central Processing Unit, abrégé en CPU. Tout processeur est conçu pour effectuer un nombre limité d'opérations bien précises, comme des calculs, des échanges de données avec la mémoire, etc.

Les processeurs multicœurs : regroupement de plusieurs processeurs dans le même boîtier. Un processeur multicœurs, contient plusieurs cœurs, chaque cœur pouvant exécuter un programme tout seul. Chaque cœur contient un séquenceur d'instruction, des registres, une unité de calcul. Les systèmes d'exploitation se sont adaptés à cette nouvelle architecture. Un processus peut lancer plusieurs *threads*, partagés sur les différents cœurs.

Pour calculer plus rapidement, le processeur utilise une mémoire plus rapide que la RAM : *Les caches*, qui sont des mémoires intermédiaires entre les registres internes du micro-processeur et la mémoire vive.

Pour un processeur multicoeur, certains caches sont dédié à chaque "coeur", d'autres sont partagés.

La puissance de la machine augmente significativement en élevant la fréquence de calcul, supérieure à celle d'un processeur monocœur. Mais cette méthode a ses limites : En effet, l'augmentation de fréquence d'un processeur cause rapidement des problèmes de surchauffe.

1.2.2 mémoires ROM et RWM

Les mémoires ROM stockent des programmes à exécuter et sont lues directement par le processeur. Pour les mémoires RWM ou RAM, on peut y accéder en lecture et écriture.

Les RWM stockent les variables du programme à exécuter, des données que le programme va manipuler. Selon l'architecture de la machine, le programme peut être entièrement stocké dans la ROM, ou bien être partagé entre la ROM et la RWM.

1.2.3 Les bus

Les données doivent circuler entre les différentes parties d'un ordinateur, notamment entre la mémoire vive et le CPU. Le système permettant cette circulation est appelé bus. Il existe, sans entrer dans les détails, 3 grands types de bus : adresses, données, de contrôle.

1.2.4 Les périphériques

Les périphériques sont très divers. On les classe en périphérique d'entrée, sortie, entrée-sortie.

Toutes les entrées-sorties contiennent des *registres d'interfaçage*, qui permettent de faire l'intermédiaire entre le périphérique et le reste de l'ordinateur.

part 2

Loi de Moore

Des années 1970 aux années 2000, la miniaturisation des circuits a suivi la Loi de Moore qui prédit un doublement du nombre de transistors par cm² tous les 18 mois. Cette miniaturisation et l'augmentation des fréquences d'horloge (qui ont aussi doublé environ tous les 18 mois) ont permis d'augmenter la puissance des processeurs de façon exponentielle pendant près de 40 ans.

La figure suivante montre l'évolution du nombre de transistors pour un même élément de surface : [source-wikipedia](#)

```
{{< img src="../../images/processeur3.png" caption="illustration de la loi de Moore - source wikipedia" >}}
```

L'image ci-dessous illustre, elle, l'évolution de la finesse de gravure

```
{{< img src="../../images/evol_transistors.png" caption="illustration de la finesse de gravure dans le temps - source eskool.gitlab.io/" >}}
```

part 3

Architecture Von Neumann vs Harvard

La différence de ces architectures repose sur la gestion des mémoires pour les programmes et données :

- Von Neumann : un même support (RWM) contient programme et données pour les processus en cours. Les instructions des programmes sont dupliquées dans la RAM. Elles existent alors à 2 endroits distincts de la machine : dans la mémoire de masse (persistante) du système, mais aussi dans la RAM. Il serait assez inefficace d'exécuter le programme depuis la mémoire de masse, car celle-ci est beaucoup plus lente.

- Harvard : Données dans une mémoire de travail différente de la RAM (mémoire flash) utilisée pour les programmes. Pas de duplication du programme. Economie des *supports de stockage*, et *economie d'énergie*.

Voir la video expliquant la différence entre ces 2 architectures :

{{}}

L'architecture Harvard est utilisée pour les microcontrôleurs, pour lesquels le besoin de miniaturisation amène à intégrer tous les composants internes dans une même puce (onchip). Les microcontrôleurs embarquent une mémoire de type EEPROM (*electrically erasable programmable read only memory 64-256ko*), suffisamment rapide d'accès pour l'utiliser directement, et sans avoir besoin de charger le programme dans une nouvelle RAM.

La machine avec architecture Harvard peut aussi gérer une mémoire de masse (persistante) comme un disque dur, ou carte flash externe. Il s'agit alors d'une extension, utile pour augmenter la capacité de la mémoire flash interne, qui est souvent réduite.

part 4

SoC : System on Chip

Un SoC - System on Chip Système sur Puce est un circuit intégré qui accueille sur une même puce, dans une taille très réduite, comprenant de la mémoire, un ou plusieurs microprocesseurs, des périphériques d'interface (processeur graphique GPU...), et autres composants nécessaires à la composition du SoC complet (modules), parmi lesquels :

- un (ou plusieurs) microprocesseur CPU - Central Processing Unit
- de la mémoire :
 - vive RAM - Random Access Memory :
 - La mémoire LPDDR - Low Power Double Data Rate (Vitesse de données double à faible consommation) / LPDDR SDRAM / DDR Mobile / mDDR
- des contrôleurs pour diverses fonctions :
 - accélération des algorithmes de Machine Learning
 - Image Processing Unit pour la Photo
 - DSP - Digital Signal Processing pour le Multimedia
 - puces de connectivité
 - GPS
 - gestion de l'alimentation
 - ...

La consommation électrique, et l'échauffement sont beaucoup inférieurs.

Ces systèmes permettent de réduire l'encombrement, la consommation d'énergie (courants faibles), et de diminuer le coût de production (gros volumes de fabrication). L'inconvénient principal est qu'il est n'est pas possible de modifier soi-même ses composants, et de l'upgrader.