

Chapitre 2

Architecture des ordinateurs

2.1 Une brève histoire des ordinateurs

2.1.1 Machines à programmes externes

D’abord, des machines électromécaniques

L’ingénieur allemand Konrad Zuse développa entre 1936 et 1938 le Z1, un ordinateur mécanique utilisant le système binaire qui ne fonctionna jamais complètement. Parallèlement, il développa entre 1937 et 1941 et Z3, premier ordinateur électromécanique complètement automatique lisant son programme sur une bande perforée. Celui-ci utilisait déjà le calcul en virgule flottante et réalisait 3 à 4 additions par seconde. Il fut utilisé par l’institut de recherche aéronautique allemand pendant la seconde guerre mondiale afin de réaliser des analyses statistiques sur les vibrations des ailes d’avions.

L’université de Harvard et l’entreprise IBM conçurent et construisirent respectivement en 1944 l’ordinateur électromécanique Mark I. Ce dernier pesait environ 5 tonnes et était composé de plus de 750 000 pièces. Les mémoires de données et de programmes étaient séparées, chacune sur sa propre carte perforée ; cela donna le nom d’architecture de Havard.

Puis des machines électroniques

L’invention des tubes à vide permit des gains de vitesse et entraîna le remplacement le remplacement des machines électromécaniques par des machines électroniques, marquant ainsi le début de l’électronique moderne. La construction de la première machine à base de composants électroniques, l’ABC, est initiée par Djon Atanasoff entre 1937 et 1942.

Entre 1943 et 1945, une équipe dirigée par Tomy Flowers met en service les Colossus. Ces ordinateurs sont utilisés pour déchiffrer le code de Lorenz employé par l’état-major allemand pour ses communications durant la seconde guerre mondiale (à ne pas confondre avec la Bombe, machine électromécanique utilisée par l’équipe d’Alan Turing pour déchiffrer le code de l’Enigma).

Entre 1943 et 1944, les américains John Mauchly et John Eckert développe l’ENIAC afin d’effectuer des analyses balistiques pour l’armée.

★ Vidéos d’e-penser sur Alan Turing et Enigma et Passe Sciences sur la machine de Turing.

2.1.2 Machines à programmes internes

Se basant principalement sur les travaux de Mauchly, Eckert, John von Neumann et Grace Hopper, les machines à programmes internes sont peu à peu développées à la fin des années 40 et dans les années 50, notamment par IBM. Dans ce type de machines, les données et les programmes résident dans la mémoire contrairement à leurs ancêtres qui utilisaient des cartes perforées. Il s'agit d'un nouveau pas vers l'informatique moderne.

2.1.3 Émergence de l'informatique moderne

Miniaturisation et ouverture

L'invention du transistor en 1947 et son industrialisation dans les années 50 marque la fin des tubes à vide et la miniaturisation des ordinateurs. Cette miniaturisation permet en 1971 la création du microprocesseur par Intel. Les gains de vitesses, de coûts, de fiabilité, de consommation énergétique et de taille des machines qui s'en suivirent contribuèrent grandement à l'informatique telle que nous la connaissons aujourd'hui. Dans la foulée, les premiers ordinateurs grand public sont conçus : l'Altair 8800 en 1975 ; l'Apple 2 en 1977, etc. Ces machines n'ont pas toujours clavier ou écran et il faudra attendre le Macintosh en 1984 pour avoir une souris et interface graphique.

★ Vidéo de C'est pas sorcier sur la miniaturisation des composants électroniques.

Langages et systèmes d'exploitation

Après la naissance des premiers compilateurs conçus Grace Hopper à partir de 1951, John Backus et IBM achèvent l'élaboration du langage Fortran en 1956. Il est suivi par le Lisp, le Cobol et enfin le Basic en 1964. Les grands principes des langages de programmation étant formulés, de nombreux langages voient le jour entre les années 1970 et 2000 : le C (1972), ML (1973) dont est issu Caml, Ada (1983) et C++ en 1986. Le langage Python verra le jour quant à lui en 1991 et JavaScript en 1995.

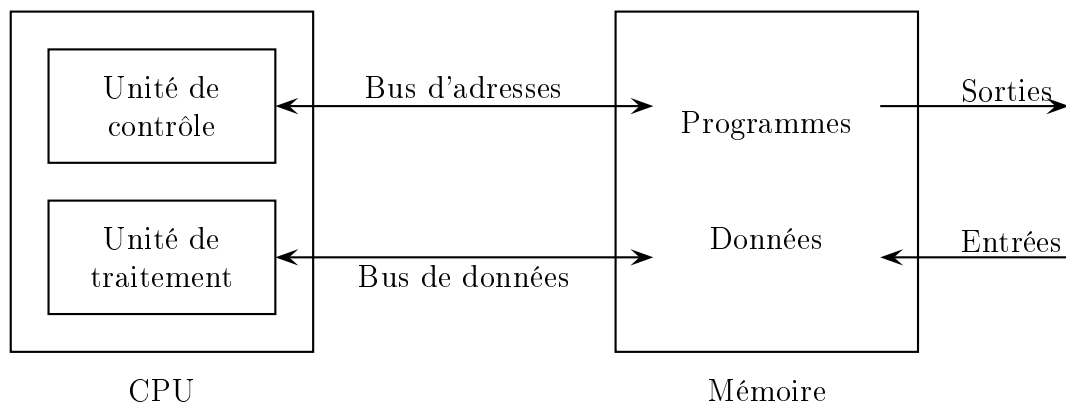
Au milieu des années 1960, chaque constructeur développe son propre système d'exploitation : OS 360 puis MVS chez IBM, le système UNIX (1970) chez AT&T... En 1984, Richard Stallman, chercheur au MIT, entame la création du système GNU (GNU's Not Unix) et promeut le mouvement du logiciel libre. Par la suite, c'est finalement MS-DOS, écrit par Microsoft pour IBM qui s'imposera sur les micro-ordinateurs grand public, suivi par Windows en 1985.

A l'heure actuelle on distingue trois grands types d'OS (Operating System) équipant les ordinateurs modernes : Windows, Mac OS et GNU Linux créé par Linus Torvalds en 1991.

2.2 Architecture de Von Neumann

John Von Neumann était un mathématicien et physicien de la première moitié du XX^e siècle. Il a grandement contribué à l'analyse fonctionnelle et à la théorie des ensembles en mathématiques ; à la mécanique quantique en physique ; au développement des ordinateurs tels que nous les connaissons aujourd'hui ; aux sciences économiques ; au développement de l'arme atomique.

En 1945, dans le rapport *First Draft of a Report on the EDVAC*, il décrit un modèle de calculateur à programme qui possède une unique mémoire qui sert à conserver les logiciels et les données. Auparavant, il fallait câbler le calculateur en fonction du programme à exécuter. Depuis les années 1940, la quasi totalité des ordinateurs sont organisés selon l'architecture dite de Von Neumann schématisée ainsi :



Remarque : CPU signifie Central Processing Unit ; en français, le processeur.

Le modèle de Von Neumann décompose l'ordinateur en 4 parties distinctes :

- **la mémoire** qui contient à la fois les données et le programme (suite d'instructions) à exécuter ;
- **les dispositifs d'entrée/sortie** qui assurent la communication entre le processeur et les périphériques ;
- dans le processeur :
 - **l'unité de contrôle** ou unité de commande, qui lit l'instruction à exécuter et présente à l'unité de traitement les opérations correspondantes avant de passer à l'instruction suivante ;
 - **l'unité de traitement**, qui effectue les calculs et écrit les résultats en mémoire ou vers l'interface d'entrées / sorties.

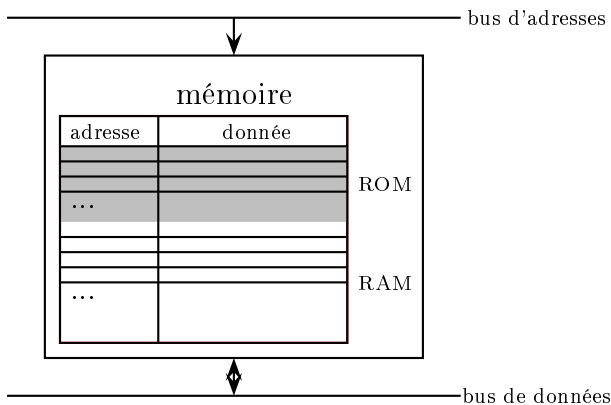
Le tout est rythmé par une horloge interne qui détermine la fréquence du processeur. Ces différents éléments sont reliés par des voies de communication appelées bus.

2.3 Architecture détaillée

2.3.1 La mémoire

Organisation

On peut considérer la mémoire comme un tableau à deux colonnes où la deuxième colonne contient la donnée et la première colonne indique son adresse. Elle est constituée de la ROM et de la RAM.



- La **ROM** (Read Only Memory), ou mémoire morte, est accessible en lecture et non en écriture. Elle conserve les données en l'absence de courant électrique. Elle sert au démarrage de l'ordinateur.
- La **RAM** (Random Access Memory), ou mémoire vive, stocke temporairement (en présence continue de courant électrique) les données et la suite d'instructions du programme à exécuter.

Le processeur possède également en interne un petit nombre de cases mémoires appelées **registres**. Cette mémoire est la plus rapide d'un ordinateur mais elle a un coût de fabrication très élevé.

Remarque : la mémoire ROM contient notamment le BIOS (Basic Input Output System) qu'il est possible, sur les machines dotés de carte mère récente, de mettre à jour (flashage du BIOS).

Les registres

Un registre est un emplacement mémoire interne au processeur. Il sert à stocker des opérandes et des résultats intermédiaires lors des opérations effectuées dans l'UAL. Leur capacité, leur nombre et leurs rôles varient selon les processeurs ; actuellement, les registres sont majoritairement de taille 64 bits. Ils sont accessibles via un jeu d'instructions.

Mémoires centrale et cachée

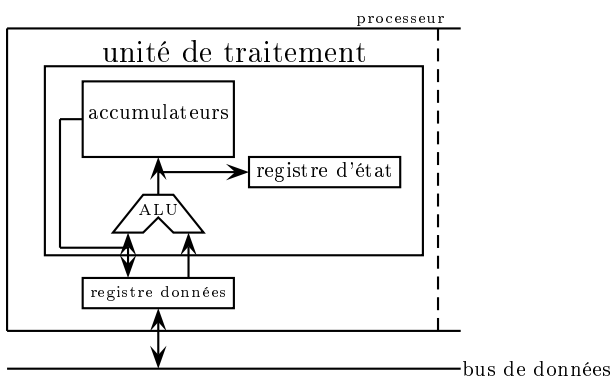
La mémoire centrale est une mémoire vive qui contient les programmes en cours et les données qu'ils manipulent. Elle est de taille importante (plusieurs Go). Elle est organisée en cellules appelées « cases mémoires » qui contiennent chacune une donnée ou une instruction repérées par une adresse qui est un nombre entier. Le temps d'accès à chaque cellule est le même : on parle de mémoire à accès aléatoire : la RAM, bien qu'il soit plus judicieux de parler de mémoire à accès direct.

Afin de pouvoir adapter la très grande vitesse du processeur à celle plus faible de la mémoire centrale, on place entre les deux une mémoire plus rapide, la mémoire cache qui contient les

instructions et les données en cours d'utilisation car, la plupart du temps, les données qui viennent d'être utilisées ont une probabilité plus grande d'être réutilisées que d'autres. La mémoire cache (de l'ordre de quelques Mo) est souvent constitué de mémoire de type statique SRAM plus rapide mais plus chère que celle de type RAM dynamique (SDRAM, DDR ...) utilisée dans la mémoire centrale.

2.3.2 L'unité de traitement

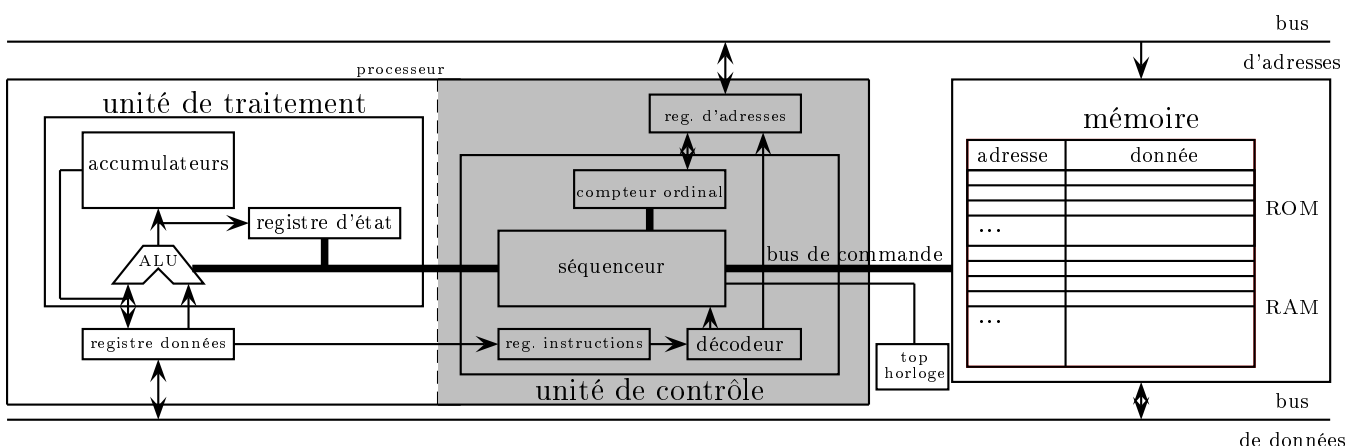
L'unité de traitement regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions.



- Les **accumulateurs** sont des registres de travail qui servent à stocker une opérande au début d'une opération arithmétique et le résultat à la fin de l'opération.
- L'**unité arithmétique et logique** (UAL ou ALU en anglais) est un circuit complexe qui assure les fonctions logiques (et, ou, comparaison ...) ou arithmétiques (addition, soustraction ...).
- Le **registre d'état** donne des informations sur l'état du processeur comme par exemple s'il y a une retenue ou un débordement des valeurs possibles.

2.3.3 L'unité de contrôle

L'unité de contrôle permet de séquencer le déroulement des instructions. Elle effectue la recherche en mémoire de l'instruction, le décodage de l'instruction. Enfin elle pilote l'exécution de l'instruction.



- Le **compteur ordinal** aussi appelé compteur de programme est un registre dont le contenu est initialisé avec l'adresse de la première instruction du programme. Il contient toujours l'adresse de la prochaine instruction à exécuter.
- Chacune des instructions à exécuter est transférée dans le **registre d'instructions** puis est décodée par le **décodeur d'instruction**.

- Le bloc logique de commande ou **séquenceur** est un automate qui, au rythme de l'horloge et en fonction des informations données par le décodeur, organise l'exécution des instructions.

2.3.4 Le rôle de l'horloge

Cadence d'un processeur

Le processeur dispose d'une horloge qui cadence l'accomplissement des instructions dont l'unité de temps est appelée cycle. Lorsqu'on l'on parle par exemple d'un processus cadencé à 3 GHz, cela signifie qu'il y a 3 milliards (3×10^9) cycles d'horloge par seconde.

Jusqu'en 2004 environ, la fréquence des processeurs a augmenté linéairement selon les lois de Moore. Depuis elle stagne... En effet, au-delà, la chaleur produite par effet Joule devient trop importante et pourrait perturber la lecture des tensions aux bornes des circuits de l'UAL, voir détériorer physiquement ses circuits électroniques.

L'augmentation de la finesse de gravure permet de diminuer l'effet Joule mais on estime qu'en dessous de 4 nm les effets quantiques (effet tunnel notamment) se feront sentir rendant le microprocesseur instable.

Cycles d'instructions

Dans un processeur, ce que l'on appelle un cycle correspond à l'exécution de chacune des cinq actions suivantes :

1. lire l'instruction (LI) ;
2. décoder l'instruction (DI) ;
3. exécuter l'opération dans l'UAL (EX) ;
4. accéder à la mémoire en lecture ou en écriture (M) ;
5. écrire le résultat dans le registre (ER).

Afin d'optimiser le temps global d'exécution, le processeur n'exécute pas les opérations de manière séquentielle mais exécute simultanément plusieurs instructions qui sont à des étapes différentes de leur traitement : c'est le principe du « pipeline d'instructions ». Le pipeline est un concept s'inspirant du fonctionnement d'une ligne de montage.

Exemple : dans le tableau ci-dessous, on a cinq instructions qui devraient demander $5 \times 5 = 25$ cycles pour être exécutées séparément. Ici, il en faut seulement 9 (charge maximale du processeur au cycle 5). Cette optimisation peut être ralentie par des branchements ou accélérée par une meilleure organisation de la mémoire.

Instructions	Cycles du processeur								
	1	2	3	4	5	6	7	8	9
A	LI	DI	EX	M	ER				
B		LI	DI	EX	M	ER			
C			LI	DI	EX	M	ER		
D				LI	DI	EX	M	ER	
E					LI	DI	EX	M	ER