

# Chap. I : Architecture de base d'un ordinateur

Laurent Poinot

UMR 7030 - Université Paris 13 - Institut Galilée

Cours “Architecture et Système”

Dans cette partie, nous décrivons rapidement l'architecture de base d'un ordinateur et les principes de son fonctionnement. Un ordinateur est une machine de traitement de l'information. Il est capable d'acquérir de l'information, de la stocker, de la transformer en effectuant des traitements quelconques, puis de la restituer sous une autre forme. Le mot " informatique " vient de la contraction des mots " information " et " automatique ". Nous appelons **information** tout ensemble de données. On distingue généralement différents types d'informations : textes, nombres, sons, images, etc., mais aussi les instructions composant un programme. Comme on l'a vu dans la première partie, toute information est manipulée sous forme binaire (ou numérique) par l'ordinateur.

# Plan du chapitre

- ① Principes généraux de fonctionnement ;
- ② La mémoire principale ;
- ③ Le processeur central ;
- ④ Liaisons Processeur-Mémoire : les bus.

# Mémoire principale & processeur

Les deux principaux constituants d'un ordinateur sont la **mémoire principale** et le **processeur** ou **CPU** (Central Processing Unit). La mémoire principale permet de stocker de l'information (programmes et données), tandis que le processeur exécute pas à pas les instructions composant les programmes.

# Architecture de von Neumann

L'architecture, dite **architecture de von Neumann** décompose l'ordinateur en quatre parties distinctes :

- ① Le processeur est composé d'une **unité arithmétique et logique** (UAL ou ALU en anglais) ou unité de traitement : son rôle est d'effectuer les opérations de base et d'une **unité de contrôle**, chargée du séquençage des opérations ;
- ② La **mémoire** qui contient à la fois les données et le programme exécuté par l'unité de contrôle. La mémoire se divise entre **mémoire volatile** ou **RAM** (Random Access Memory) qui contient programmes et données en cours de traitement, et **mémoire permanente** ou **ROM** (Read Only Memory) qui stocke programmes et données de base de la machine ;
- ③ Les dispositifs d'**entrée-sortie**, qui permettent de communiquer avec le monde extérieur.

Les différents composants sont reliés par des **bus**.

# Les programmes

Un **programme** est une suite d'instructions élémentaires, qui vont être exécutées dans l'ordre par le processeur. Ces instructions correspondent à des actions très simples, telles qu'additionner deux nombres, lire ou écrire une case mémoire, etc. Chaque instruction est codée (physiquement câblée) en mémoire sur quelques octets. Le processeur est capable d'exécuter des programmes en **langage machine**, c'est-à-dire composés d'instructions très élémentaires suivant un codage précis. Chaque type de processeurs est capable d'exécuter un certain ensemble d'instructions, son **jeu d'instructions**. Pour écrire un programme en langage machine, il faut donc connaître les détails du fonctionnement du processeur qui va être utilisé. Les langages de programmation utilisés de nos jours sont très largement plus évolués que le langage machine. Néanmoins ils reposent sur ce dernier : après compilation ils sont transformés (traduits) en langage machine exécutable par le processeur.

# Le processeur

Le **processeur**, (ou CPU, Central Processing Unit, “ Unité centrale de traitement ” en français) est le composant essentiel d’un ordinateur qui interprète les instructions et traite les données d’un programme. Le processeur est un circuit électronique complexe (circuit intégré) qui exécute chaque instruction très rapidement, en quelques cycles d’**horloges**. Toute l’activité de l’ordinateur est cadencée par une horloge unique, de façon à ce que tous les circuits électroniques travaillent tous ensemble de façon synchronisée. La fréquence de cette horloge s’exprime en MHz (millions de cycles par seconde) ou GHz (milliards de cycles par secondes). Par exemple, un processeur “ Intel Core 2 Duo P8400 ” possède une horloge cadencée à 2,26 GHz.

# Principe de fonctionnement du processeur

Pour chaque instruction, le processeur effectue schématiquement les opérations suivantes :

- ① lire dans la mémoire principale l'instruction à exécuter ;
- ② effectuer le traitement correspondant à cette instruction ;
- ③ passer à l'instruction suivante.



# Caractéristiques d'un processeur

Un processeur est défini par :

- ① la largeur de ses registres internes de manipulation de données (8, 16, 32, 64, 128 bits) ;
- ② la cadence de son horloge exprimée en MHz ou GHz ;
- ③ le nombre de noyaux de calcul (core) ;
- ④ son jeu d'instructions (ISA en anglais, Instructions Set Architecture) dépendant de la famille (CISC, RISC, etc) ;
- ⑤ sa finesse de gravure exprimée en nm (nanomètres,  $10^{-9}$  mètres, soit un milliardième de mètre).

# Caractéristiques d'un processeur (suite)

Ce qui caractérise principalement un processeur est la famille à laquelle, il appartient :

- ① CISC (Complex Instruction Set Computer) : choix d'instructions aussi proches que possible d'un langage de haut niveau ;
- ② RISC ( Reduced Instruction Set Computer) : choix d'instructions plus simples et d'une structure permettant une exécution très rapide ;
- ③ VLIW (Very Long Instruction Word) ;
- ④ DSP (Digital Signal Processor). Même si cette dernière famille (DSP) est relativement spécifique. En effet un processeur est un composant programmable et est donc a priori capable de réaliser tout type de programmes. Les DSP sont des processeurs spécialisés pour les calculs liés au traitement de signaux.

# Structure d'un processeur

Le processeur est grossièrement divisé en deux parties, l'**unité de contrôle** et l'**unité de traitement** :

- ① l'unité de contrôle est responsable de la lecture en mémoire principale et du décodage des instructions ;
- ② l'unité de traitement, aussi appelée **Unité Arithmétique et Logique** (U.A.L.), exécute les instructions qui manipulent les données.

Ces deux unités communiquent avec la mémoire principale, la première pour lire les instructions, la seconde pour recevoir/transmettre des données binaires, mais ils communiquent également avec les différents périphériques (clavier, souris, écran, etc.).

# Structure de la mémoire principale

La mémoire est divisée en emplacements (des cases mémoires contiguës) de taille fixe (par exemple huit bits) utilisés pour stocker instructions et données. En principe, la taille d'un emplacement mémoire pourrait être quelconque ; en fait, la plupart des ordinateurs en service aujourd'hui utilisent des emplacements mémoire d'un octet (" byte " en anglais, soit huit bits, unité pratique pour coder un caractère par exemple).

# Adresse mémoire

Dans une mémoire de taille  $N$ , on a  $N$  emplacements mémoires, numérotés (ou adressés) de 0 à  $N - 1$ . Chaque emplacement est repéré par un numéro unique, appelé **adresse**. L'adresse est le plus souvent écrite en hexadécimal.

# Opérations sur la mémoire

Seul le processeur peut modifier l'état de la mémoire. Chaque emplacement mémoire conserve les informations que le processeur y écrit jusqu'à coupure de l'alimentation électrique, où tout le contenu est perdu (contrairement au contenu des mémoires externes comme les disquettes et disques durs). On parle de **mémoire vive**. Les seules opérations possibles sur la mémoire sont :

- ① **écriture** d'un emplacement : le processeur donne une valeur et une adresse, et la mémoire range la valeur à l'emplacement indiqué par l'adresse ;
- ② **lecture** d'un emplacement : le processeur demande à la mémoire la valeur contenue à l'emplacement dont il indique l'adresse. Le contenu de l'emplacement auquel le processeur accède en lecture demeure inchangé.

# Caractéristiques d'une mémoire

- ① La **capacité** : nombre total de bits que contient la mémoire. Elle s'exprime aussi souvent en octet ;
- ② Le **format des données** : nombre de bits que l'on peut mémoriser par case mémoire. On parle de la **largeur** du mot mémorisable ;
- ③ Le **temps d'accès** : temps qui s'écoule entre l'instant où a été lancée une opération de lecture/écriture en mémoire et l'instant où la première information est disponible sur le bus de données ;
- ④ Le **temps de cycle** : il représente l'intervalle minimum qui doit séparer deux demandes successives de lecture ou d'écriture ;
- ⑤ Le **débit** : nombre maximum d'informations lues ou écrites par seconde ;
- ⑥ La **volatilité** : elle caractérise la permanence des informations dans la mémoire. L'information stockée est volatile si elle risque d'être altérée par un défaut d'alimentation électrique et non volatile dans le cas contraire.

# RAM

Nous savons qu'il existe deux types distincts de mémoire : les mémoires vives et les mémoires mortes.

Une mémoire vive sert au stockage temporaire de données. Elle doit avoir un temps de cycle très court pour ne pas ralentir le microprocesseur. Les mémoires vives sont en général volatiles : elles perdent leurs informations en cas de coupure d'alimentation. (Certaines d'entre elles, ayant une faible consommation, peuvent être rendues non volatiles par l'adjonction d'une batterie.) Il existe deux grandes familles de mémoires RAM (Random Access Memory : mémoire à accès aléatoire) :

- ① Les RAM statiques ;
- ② Les RAM dynamiques.



# RAM statique

Le bit mémoire d'une **RAM statique** (SRAM) est composé d'une bascule (composant électronique élémentaire). Chaque bascule contient entre quatre et six transistors.

# RAM dynamique

Dans les **RAM dynamiques** (DRAM), l'information est mémorisée sous la forme d'une charge électrique stockée dans un condensateur.

- **Avantages** : Cette technique permet une plus grande densité d'intégration, car un point mémoire nécessite environ quatre fois moins de transistors que dans une mémoire statique. Sa consommation s'en retrouve donc aussi très réduite ;
- **Inconvénient** : La présence de courants de fuite dans le condensateur contribue à sa décharge. Ainsi, l'information est perdue si on ne la régénère pas périodiquement (charge du condensateur). Les RAM dynamiques doivent donc être rafraîchies régulièrement pour entretenir la mémorisation : il s'agit de lire l'information et de la recharger.

# ROM

Pour certaines applications, il est nécessaire de pouvoir conserver des informations de façon permanente même lorsque l'alimentation électrique est interrompue. On utilise alors des mémoires mortes ou mémoires à lecture seule (ROM : Read Only Memory). Ces mémoires sont non volatiles. Ces mémoires, contrairement aux RAM, ne peuvent être que lues. L'inscription en mémoire des données restent possible mais est appelée programmation. Suivant le type de ROM, la méthode de programmation changera. Il existe donc plusieurs types de ROM :

- ROM ;
- PROM ;
- EPROM ;
- EEPROM ;
- FLASH EPROM.

Le processeur est parfois appelé **CPU** (de l'anglais “ Central Processing Unit ”) ou encore **MPU** (Micro-Processing Unit) pour les microprocesseurs. Un microprocesseur n'est rien d'autre qu'un processeur dont tous les constituants sont réunis sur la même puce électronique (pastille de silicium), afin de réduire les coûts de fabrication et d'augmenter la vitesse de traitement. Les micro-ordinateurs (ordinateurs personnels) sont tous équipés de microprocesseurs. L'architecture de base des processeurs équipant les gros ordinateurs est la même que celle des microprocesseurs.

# Composition d'un processeur

Les parties essentielles d'un processeur sont :

- ① L'**Unité Arithmétique et Logique** (UAL, en anglais Arithmetic and Logical Unit - ALU), qui prend en charge les calculs arithmétiques élémentaires et les tests ;
- ② L'**unité de contrôle** ou séquenceur, qui permet de synchroniser les différents éléments du processeur. En particulier, il initialise les registres lors du démarrage de la machine et il gère les interruptions ;
- ③ Les **registres**, qui sont des mémoires de petite taille (quelques octets), suffisamment rapides pour que l'UAL puisse manipuler leur contenu à chaque cycle de l'horloge. Un certain nombre de registres sont communs à la plupart des processeurs.

# Types de registres

- **Compteur de programme** : ce registre contient l'adresse mémoire de l'instruction en cours d'exécution ;
- **Accumulateur** : ce registre est utilisé pour stocker les données en cours de traitement par l'UAL ;
- **Registre d'adresses** : il contient toujours l'adresse de la prochaine information à lire par l'UAL : soit la suite de l'instruction en cours, soit la prochaine instruction ;
- **Registre d'instructions** : il contient l'instruction en cours de traitement ;
- **Registre d'état** : il sert à stocker le contexte du processeur, ce qui veut dire que les différents bits de ce registre sont des “ drapeaux ” (flags) servant à stocker des informations concernant le résultat de la dernière instruction exécutée.

## Types de registres (suite)

- **Pointeurs de pile** : ce type de registre, dont le nombre varie en fonction du type de processeur, contient l'adresse du sommet de la pile (ou des piles) ;
- **Registres généraux** : ces registres sont disponibles pour les calculs ;
- L'**horloge** qui synchronise toutes les actions de l'unité centrale ;
- L'**unité d'entrée-sortie**, qui prend en charge la communication avec la mémoire de l'ordinateur, permettant au processeur d'accéder aux périphériques de l'ordinateur.

# Opérations du processeur

Le rôle fondamental de la plupart des unités centrales de traitement, indépendamment de la forme physique qu'elles prennent, est d'exécuter une série d'instructions stockées appelées " programme ". Les instructions et les données transmises au processeur sont exprimées en mots binaires (code machine). Elles sont stockées dans la mémoire. Le séquenceur ordonne la lecture du contenu de la mémoire et la constitution des mots présentés à l'UAL qui les interprète. L'ensemble des instructions et des données constitue un programme. Le langage le plus proche du code machine tout en restant lisible par des humains est le **langage d'assemblage**, aussi appelé langage **assembleur** (forme francisée du mot anglais " assembler "). Toutefois, l'informatique a développé toute une série de langages, dits de **haut niveau** (comme le BASIC, Pascal, C, C++), destinés à simplifier l'écriture des programmes. Les opérations décrites ici sont conformes à l'architecture de von Neumann.



# Opérations du processeur (suite)

Le programme est représenté par une série d'instructions qui réalisent des opérations en liaison avec la mémoire vive de l'ordinateur. Il y a quatre étapes lors du traitement des instructions :

- ① **FETCH** : Recherche de l'instruction ;
- ② **DECODE** : Décodage de l'instruction ;
- ③ **EXECUTE** : Exécution des opérations ;
- ④ **WRITEBACK** : Écriture du résultats.

# FETCH

La première étape, FETCH (recherche), consiste à rechercher une instruction dans la mémoire vive de l'ordinateur. L'emplacement dans la mémoire est déterminé par le compteur de programme, qui stocke l'adresse de la prochaine instruction dans la mémoire de programme. Après qu'une instruction a été recherchée, le compteur de programme est incrémenté par la longueur du mot d'instruction. L'instruction que le processeur recherche en mémoire est utilisée pour déterminer ce que le CPU doit faire.

# DECODE

Dans l'étape DECODE (décodage), l'instruction est découpée en plusieurs parties telles qu'elles puissent être utilisées par d'autres parties du processeur. La façon dont la valeur de l'instruction est interprétée est définie par le jeu d'instructions du processeur. Souvent, une partie d'une instruction, appelée **opcode** (code d'opération), indique quelle opération est à faire, par exemple une addition. Les parties restantes de l'instruction comportent habituellement les autres informations nécessaires à l'exécution de l'instruction comme par exemple des valeurs pour l'addition.

# EXECUTE

Après les étapes de recherche et de décodage arrive l'étape EXECUTE (exécution) de l'instruction. Au cours de cette étape, différentes parties du processeur sont mises en relation pour réaliser l'opération souhaitée. Par exemple, pour une addition, l'unité arithmétique et logique (UAL) sera connectée à des entrées et des sorties. Les entrées présentent les nombres à additionner et les sorties contiennent la somme finale. L'UAL contient le circuit électronique pour réaliser des opérations d'arithmétique et de logique simples sur les entrées (addition, opération sur les bits). Si le résultat d'une addition est trop grand pour être codé par le processeur, un signal de débordement est positionné dans un registre d'état.

# WRITEBACK

La dernière étape WRITEBACK (écriture du résultat), écrit tout simplement les résultats de l'étape d'exécution en mémoire. Très souvent, les résultats sont écrits dans un registre interne au processeur pour bénéficier de temps d'accès très courts pour les instructions suivantes. Dans d'autres cas, les résultats sont écrits plus lentement dans des mémoires RAM, donc à moindre coût et acceptant des codages de nombres plus grands.

# Liaisons processeur-mémoire : les bus

Les informations échangées entre la mémoire et le processeur circulent sur des **bus**. Un bus est simplement un ensemble de  $n$  fils conducteurs, utilisés pour transporter  $n$  signaux binaires. Le **bus d'adresse** est un bus unidirectionnel : seul le processeur envoie des adresses. Il est composé de  $n$  fils ; on utilise donc des adresses de  $n$  bits. La mémoire peut posséder au maximum  $2^n$  emplacements (adresses 0 à  $2^n - 1$ ). Le **bus de données** est un bus bidirectionnel. Lors d'une lecture, c'est la mémoire qui envoie un mot sur le bus (le contenu de l'emplacement demandé) ; lors d'une écriture, c'est le processeur qui envoie la donnée.