

# Architecture des systèmes informatiques

## Cours 1

*“Un bon informaticien, c'est comme un pilote de formule 1 ; si il ne maîtrise pas sa machine, il va dans le décor !”*

# Un peu d'histoire

- De tout temps, les hommes ont cherché à calculer. Le mot “*calcul*” vient d'ailleurs de “*calculus*”, caillou en latin
- A chaque époque, un saut technologique
  - Machines mécaniques (ex : la pascaline en 1642)
  - Machines électromécaniques avec l'apparition de l'électricité (machine de hermann et ses cartes perforées en 1890)
  - Machines électroniques
    - Lampes (ABC en 1941 et ENIAC en 1945)
    - Transistor (TRADIC en 1955 et le premier disque dur)
    - Circuits intégrés depuis les années 70

# Vers la banalisation de l'architecture

- On programme avec des langages sophistiqués sur des systèmes d'exploitation intuitifs
- Mais le concept de base reste la logique binaire, vrai/faux, courant/pas courant
- De fait, de multiples couches se superposent
  - Langages évolués
  - assembleur
  - Système d'exploitation
  - Instructions machines au niveau du processeur
  - Micro-instructions à l'intérieur du Circuit
  - Commandes électroniques

# Langages machine et d'assemblage

- Le processeur exécute des instructions qui lui sont propres : c'est son langage ; on parle de langage machine
  - Suite de 0 et de 1 difficile à lire par un homme
- Le langage d'assemblage d'un processeur représente son langage machine sous une forme lisible par l'homme
- Un assembleur est un applicatif permettant de convertir un langage d'assemblage en langage machine
  - 1 processeur, 1 Langage d'ass, 1 langage machine

# Langages évolués

- Un langage évolué est un langage informatique de haut niveau indépendant vis-à-vis d'une architecture
  - C, C++, Java, Python, ....
- On utilise des compilateurs pour convertir ces langages vers le langage machine du processeur qui exécutera ce programme
  - Ex : gcc
- On peut aussi s'arrêter au langage d'assemblage
  - Ex : *gcc -S fichier.c* produira *fichier.s*
  - Il faut alors utiliser *as fichier.s*

# Chaîne de compilation

- Une chaîne de compilation est un ensemble de conversion des différentes parties de code pour arriver à un programme exécutable
  - Compilation des différents langages contenus dans différents fichiers
  - Assemblage des parties en langage d'assemblage
  - Edition de liens pour rassembler tous les morceaux de code machine en un seul programme

# Objectif du cours

- Comprendre le lien entre langage évolué et sa conversion en langage cible pour un processeur donné
- Apprendre les grands principes de fonctionnement d'un processeur, des périphériques associés et les mécanismes sous-jacent (interruptions, pile d'exécution, etc)
- Utiliser un environnement de développement adapté (plate-forme arduino)

# Architecture de Von Neumann

- Principes
  - séparer l'unité de commande, l'unité de traitement et le programme
  - Le programme est enregistré dans une mémoire
  - Le code et les données sont séparés mais dans la même mémoire
  - La machine communique avec des dispositifs d'entrées/sorties
- Les fondements des machines modernes dont le premier prototype est l'EDVAC en 1945

# Architecture de Von Neumann (2)

L'architecture de von Neumann décomposait l'ordinateur en 4 parties distinctes

1. **L'unité arithmétique et logique (UAL)** ou unité de traitement : son rôle est d'effectuer les opérations de base, un peu comme le ferait une calculatrice ;

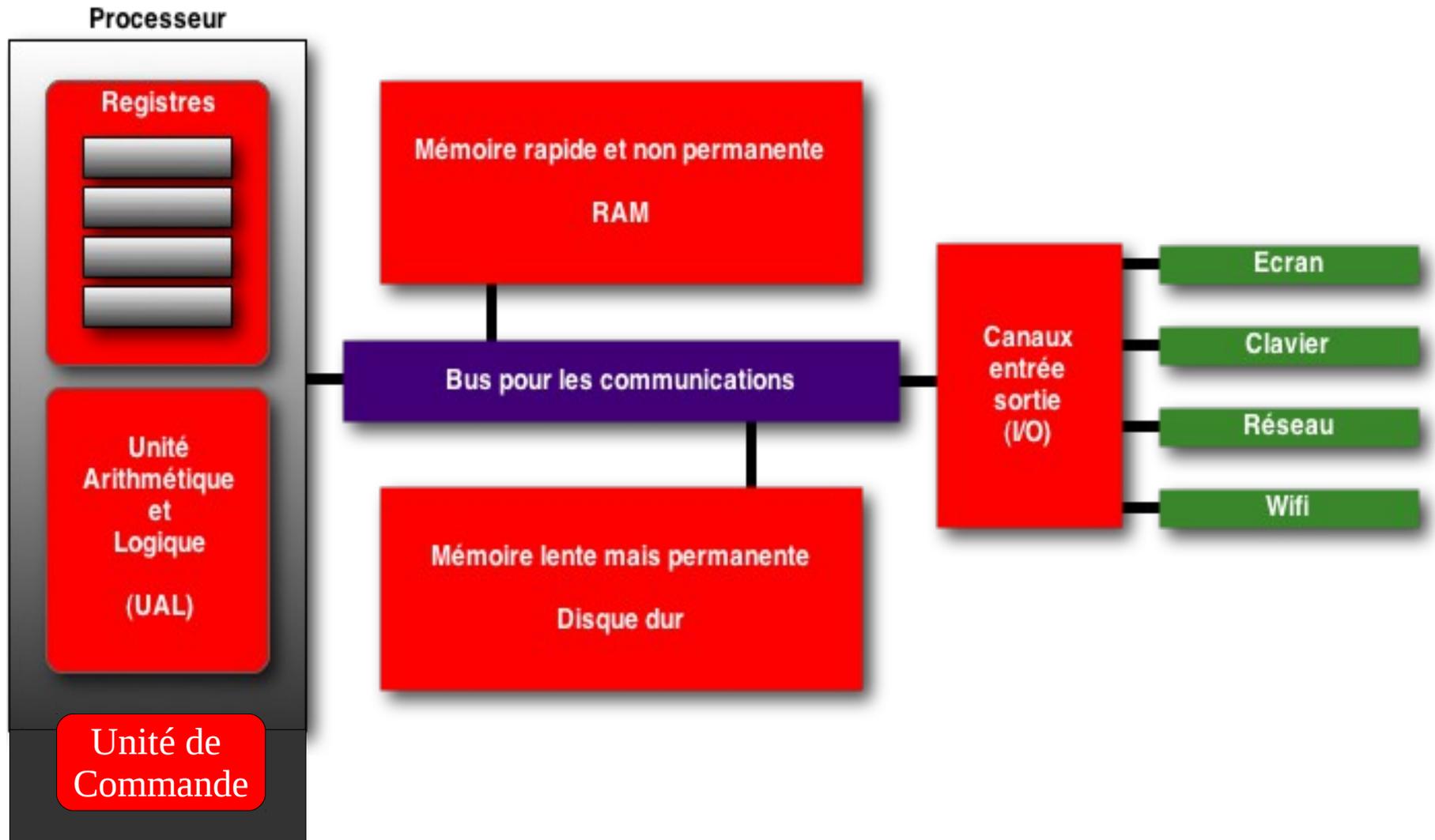
2. **L'unité de contrôle** : c'est l'équivalent des doigts qui actionneraient la calculatrice ;

# Architecture de Von Neumann (3)

3. La **mémoire** qui contient à la fois les données et le programme qui dira à l'unité de contrôle quels calculs faire sur ces données. La mémoire se divise entre mémoire volatile (programmes et données en cours de fonctionnement) et mémoire permanente (programmes et données de base de la machine) ;

4. Les **entrées-sorties** : dispositifs qui permettent de communiquer avec le monde extérieur.

# Architecture de Von Neumann (4)



*Image Inspirée du cours de Patrick BELLOT TELECOM PARIS*

# Processeur

Le processeur (Central Processing Unit) contient

- Une **Unité Arithmétique et Logique (UAL)** qui réalise des opérations élémentaires
  - Arithmétiques : addition, soustraction, ...
  - Logiques : ET, OU, comparaison ...
- Une **Unité de Commande (UC)**
  - Coordinateur général
  - Lit les instructions du programme en mémoire
  - Commande l'UAL pour exécuter ces instructions
- Des mémoires (**registres**) qui stockent des données

# Les registres

Outre des **registres de données** qui stockent les valeurs nécessaires au calcul et le résultat de ce calcul, il existe des registres particuliers. Citons :

- CO : le compteur ordinal qui contient l'adresse de la prochaine instruction à exécuter
- RI qui contient l'instruction à exécuter
- PSW qui contient les flags relatifs au calcul (retenue, dépassement, etc.)
- Pointeur de pile : la pile sert à l'exécution des programmes structurés

# Algorithme de l'UC

L'unité de commande n'a pour seul rôle que d'exécuter des instructions machine. Il suit sans fin l'algorithme suivant :

- Chargement dans RI de l'instruction pointé par CO
- Incrémentation de CO de la taille de l'instruction
- Décodage de l'instruction en micro-instructions
- Adressages des données, voire rappatriement dans les registres
- Exécution de ces micro-instructions
- Stockage du (des) résultat(s)

# L'horloge et ses cycles

- L'algorithme précédent est exécuté à une cadence définie par l'horloge
- On parle de cycle d'horloge
  - Un processeur fonctionnant à 1GHz aura un cycle de 1 nano seconde
  - Les instructions seront exécutées, selon leurs complexité, dans une durée multiple du cycle
- L'horloge est extérieure à l'UC et peut rythmer aussi les autres composants

# Le ou plutôt les bus

- Le BUS est le squelette de l'architecture
- Il relie les différentes parties entre elles pour qu'elles puissent communiquer, interagir
- Se sont des fils électriques sur lesquels circulent des signaux (Ex. RESET, WR, RD, ..)
- On parle en fait de 3 bus :
  - Bus d'adresses sur lequel est précisé l'adresse de la donnée ciblée
  - Bus de Données ou circule cette donnée
  - Bus de contrôle qui véhicule les ordres du processeur

# Mémoire

- Tout dispositif capable de stocker des informations (données ou instructions)
- Le programme à exécuter est stocké en mémoire dite vive (RAM) ; cette mémoire est volatile (perte de données si plus d'énergie)
- Le programme à “archiver” est stocké sur le disque sous forme de fichiers
- Il existe aussi des ROM qui sont non volatiles (pour le système d'exploitation ou des applications d'informatique embarquée)

# Mémoire (2)

- Ecriture => enregistrement d'une information du processeur vers la mémoire
- Lecture => récupération par le processeur d'une information précédemment enregistrée
- Une mémoire vive peut être vue comme une suite de cases numérotées de 0 à la taille de la mémoire
- Ces numéros sont les adresses
- Le contenu d'une case est une donnée
- Les données sont regroupées en mots adressables

# Résumons...

- De manière très générale et simpliste, que se passe-t-il quand on lance l'exécution d'un programme stocké sur disque ?