

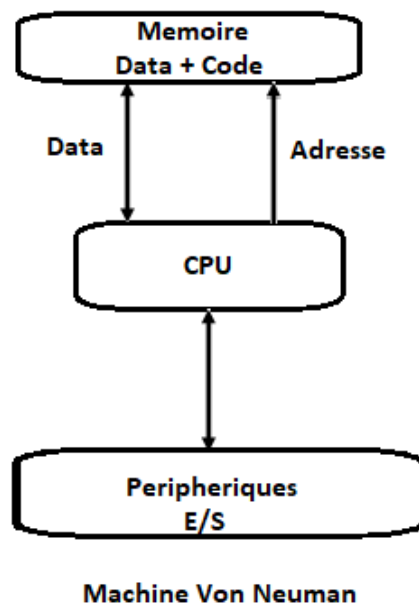
Chapitre 2

Architecture et classes des processeurs

Il existe deux principales Architectures des processeurs classées selon leur utilisation de la mémoire.

II.1 Architecture de 'Von Neuman'

C'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.

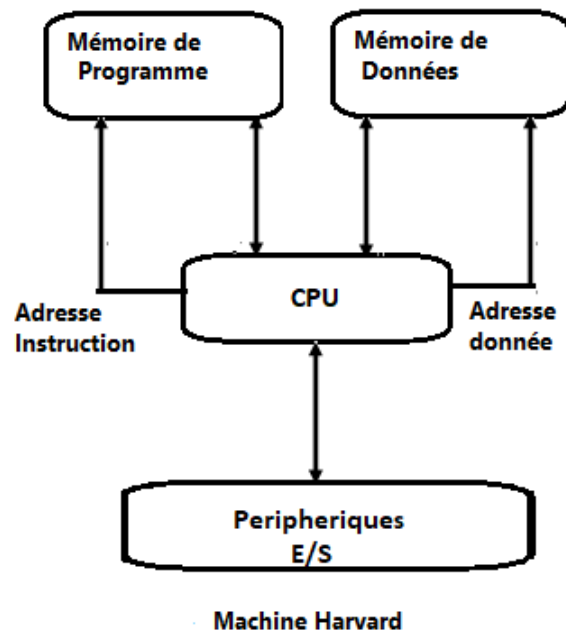


L'architecture de *von Neumann* décompose l'ordinateur en 4 parties distinctes :

- 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.

II.2 Architecture 'Harvard'

L'architecture Harvard stocke les instructions et les données de la machine dans des unités de mémoire distinctes qui sont connectées par différents bus. Dans ce cas, il y a au moins deux espaces d'adressage mémoire avec lesquels travailler, il y a donc un registre de mémoire pour les instructions de la machine et un autre registre de mémoire pour les données. Les ordinateurs conçus avec l'architecture Harvard sont capables d'exécuter un programme et d'accéder aux données indépendamment, et donc simultanément. L'architecture Harvard a une séparation stricte entre les données et le code. Ainsi, l'architecture Harvard est plus compliquée, mais des pipelines séparés éliminent le goulot d'étranglement créé par Von Neumann.



II.2 Jeu d'instructions des processeurs

Le *jeu d'instruction* (Instruction Set Architecture: ISA) a une importance capitale. Il détermine les instructions élémentaires exécutées par le CPU. C'est un équilibre entre la complexité matérielle du CPU et la facilité d'exprimer les actions requises. On le représente de manière symbolique (ex: MSP, code sur 16 bits):

`mov r5, @r8 ; commentaire [R8]<-R5`

`ADD r4, r5 ; R5<-R5+R4`

Le jeu d'instructions des processeurs se subdivise en deux classes ;

- CISC: Complex Instruction Set Computer
- RISC: Reduce Instruction Set Computer

II.2.1 CISC: Complex Instruction Set Computer

- Une instruction peut designer plusieurs opérations élémentaires.
Ex: un load, une opération arithmétique et un store,
Ex: calculer une interpolation linéaire de plusieurs valeurs en mémoire.
- Accélération par des mécanismes matériels complexes
- Grandes variations de taille et de temps d'exécution pour les instructions
- Résulte en un code compact mais complexe à générer.
- Exemple des processeurs CISC ; Motorola 68000, Intel x86/Pentium

II.2.2 RISC: Reduce Instruction Set Computer

- Petites instructions simples, toutes de même taille, ayant toutes (presque) le même temps d'exécution
- Pas d'instruction complexe
- Accélération en pipeline l'exécution (entre 3 et 7 étages de pipeline pour une instruction) augmentation de la vitesse d'horloge
- Code plus simple à générer, mais moins compact
- Tous les microprocesseurs modernes utilisent ce paradigme:
Microchip PICs, SPARC, MIPS, ARM, PowerPC, etc.