

CHAPITRE 1

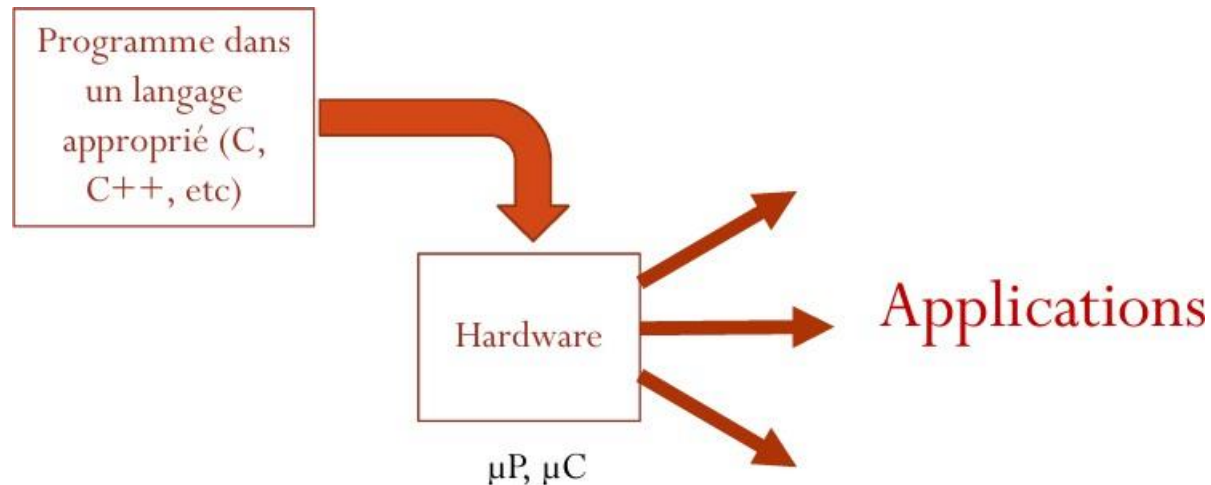
Introduction aux système embarqués (ordinateur)

Plan du Chapitre

- Qu'est ce qu'un système embarqué ?
- Architecture d'un ordinateur
- Architectures des microprocesseurs (HW)
- Architectures de jeux d'instructions (SW)

Système embarqué

Quand vous avez un programme informatique (**SW**) et que vous le mettez dans un circuit matériel compatible (**HW**) pour effectuer une application particulière, vous avez créé un système embarqué.



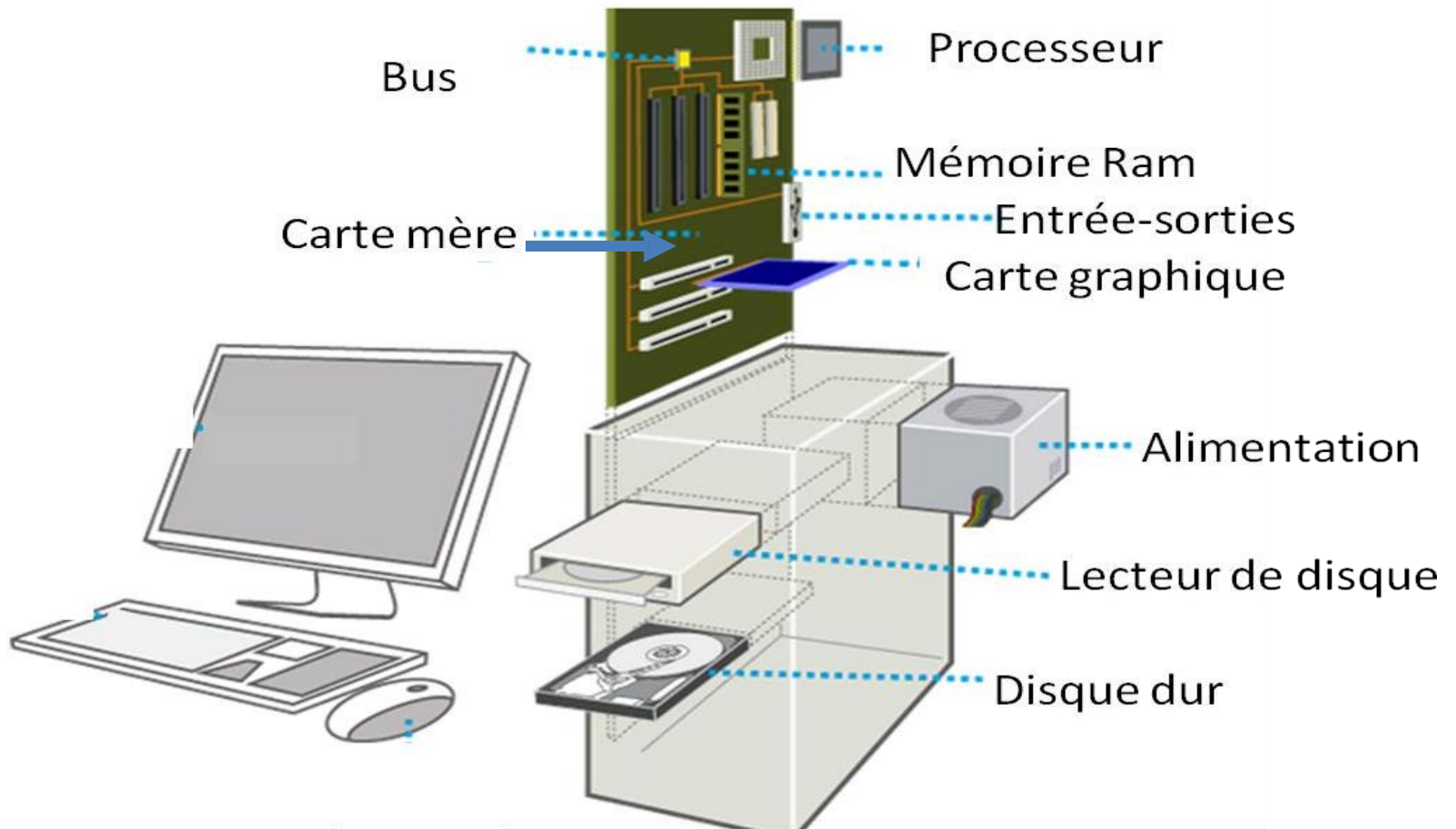


Système embarqué



Architecture d'un ordinateur

Les éléments connectés à la carte mère et les constituant de l'ordinateur



Les composants d'un ordinateur

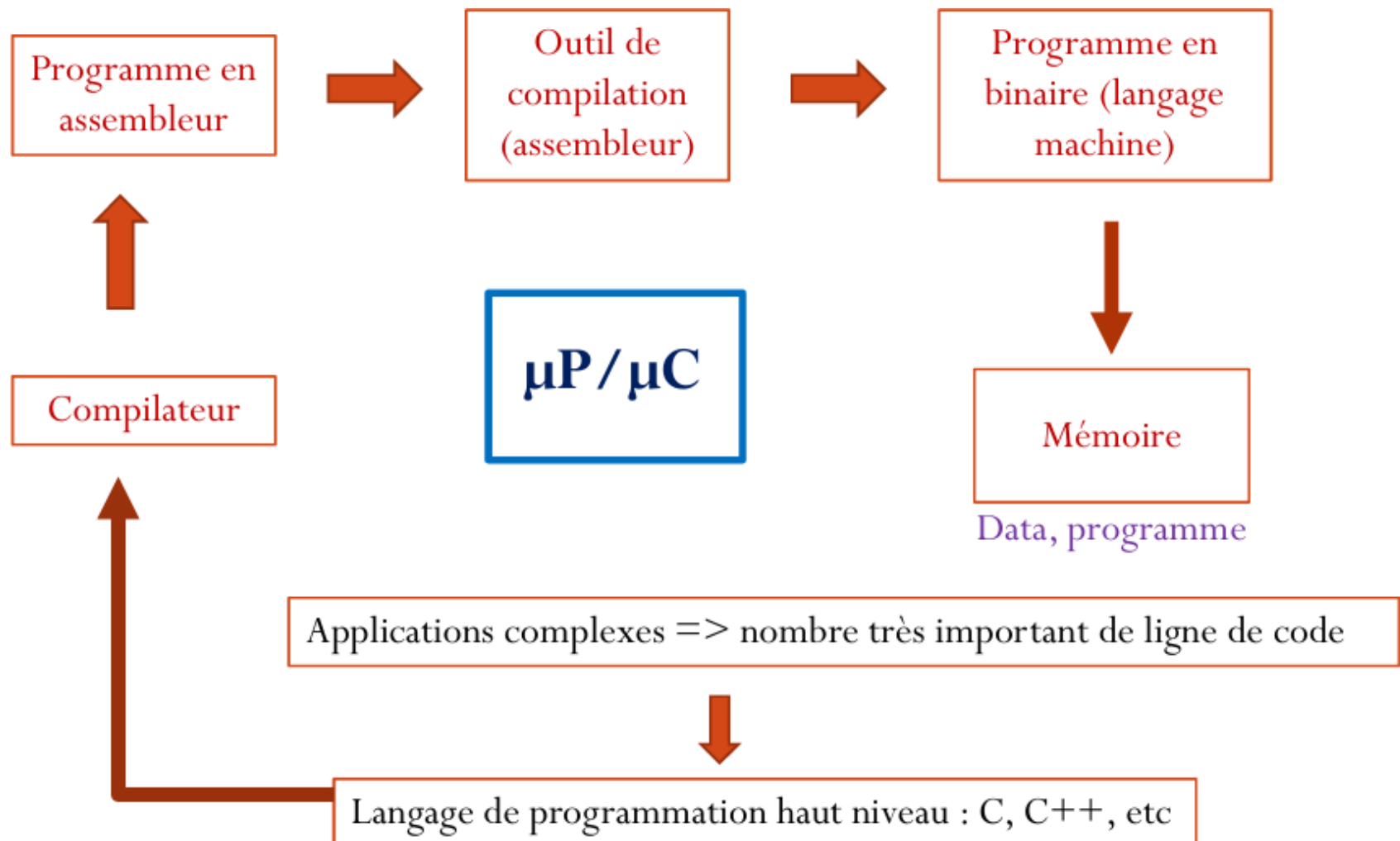
Lorsqu'on ouvre un ordinateur, quelle que soit sa marque, nous retrouvons en général les mêmes éléments:

- La **carte mère** relie tous les éléments constituant un ordinateur.
- Le **processeur ou microprocesseur** permet à l'ordinateur d'effectuer les opérations (calculs) demandés.
- Les **bus** sont des systèmes de communication permettent de connecter les différentes parties fonctionnelles de l'ordinateur entre elles.
- La **mémoire vive** (ou **RAM** « Random Access Memory ») stocke les informations des programmes et données en cours de fonctionnement.
- La **carte graphique** ou carte vidéo permet de produire une image pouvant être affichée sur un écran.
- Les **entrées-sorties** permettent de communiquer avec le monde extérieur comme les clés USB.
- Le **disque dur** stocke les informations des programmes et les données de base de la machine.
- Le **lecteur de disque** peut être lecteur et graveur en même temps.
- L'**alimentation électrique**.

D'autres éléments peuvent aussi être considérés comme principaux comme la **carte son** qui permet d'acquérir le son extérieur ou de restituer les sons venant de l'ordinateur ou bien la **carte réseau** qui elle permet de se connecter sur un réseau.

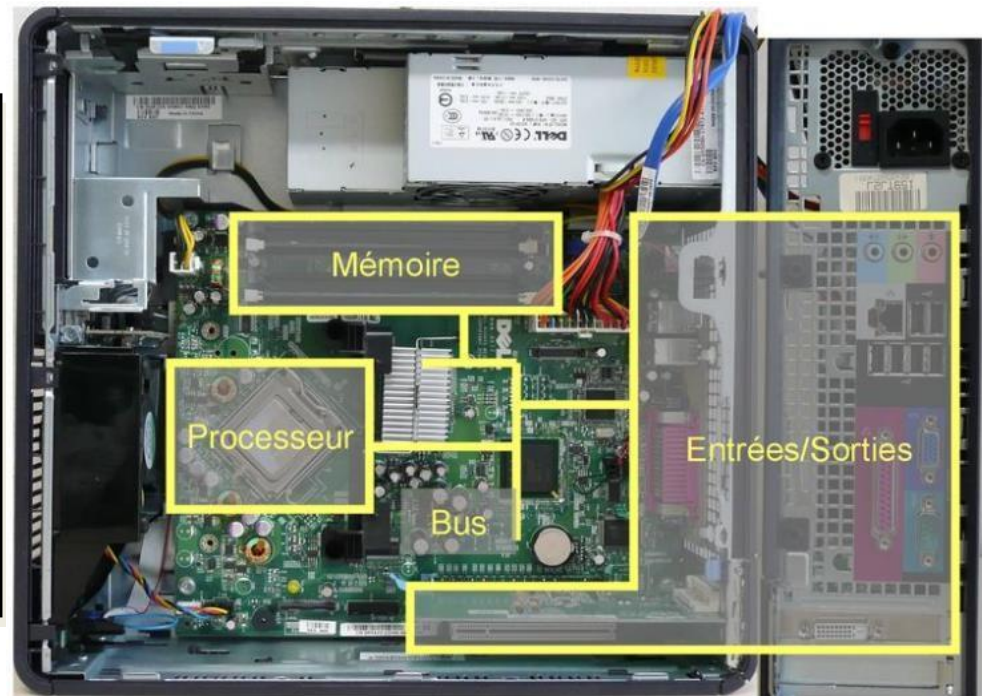
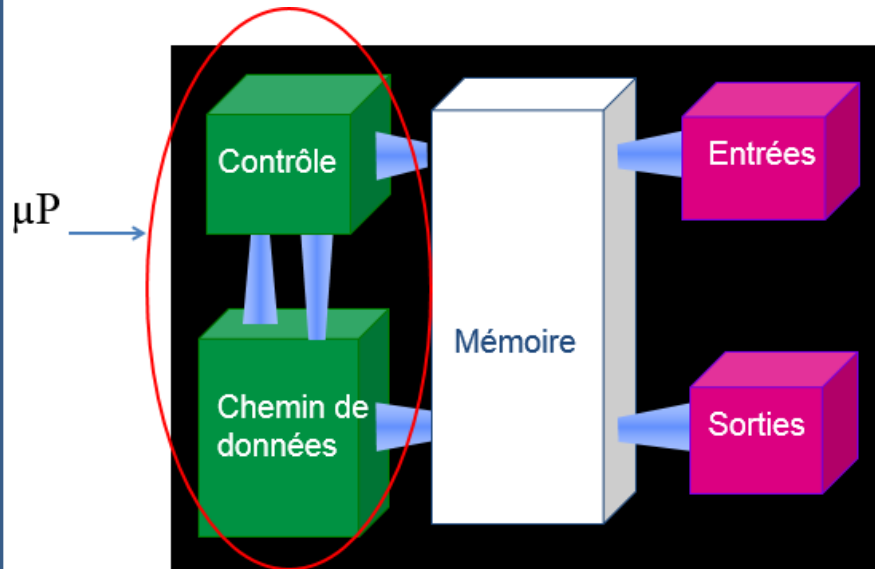
Tous ces éléments une fois connecté à la carte mère constituent ce qu'on appelle l'unité centrale de l'ordinateur.

Comment programmer un ordinateur ?



Architectures des microprocesseurs (HW)

D'une manière générale, l'organisation d'un ordinateur est donnée par un microprocesseur, une mémoire centrale et des périphériques d'entrées/sorties.

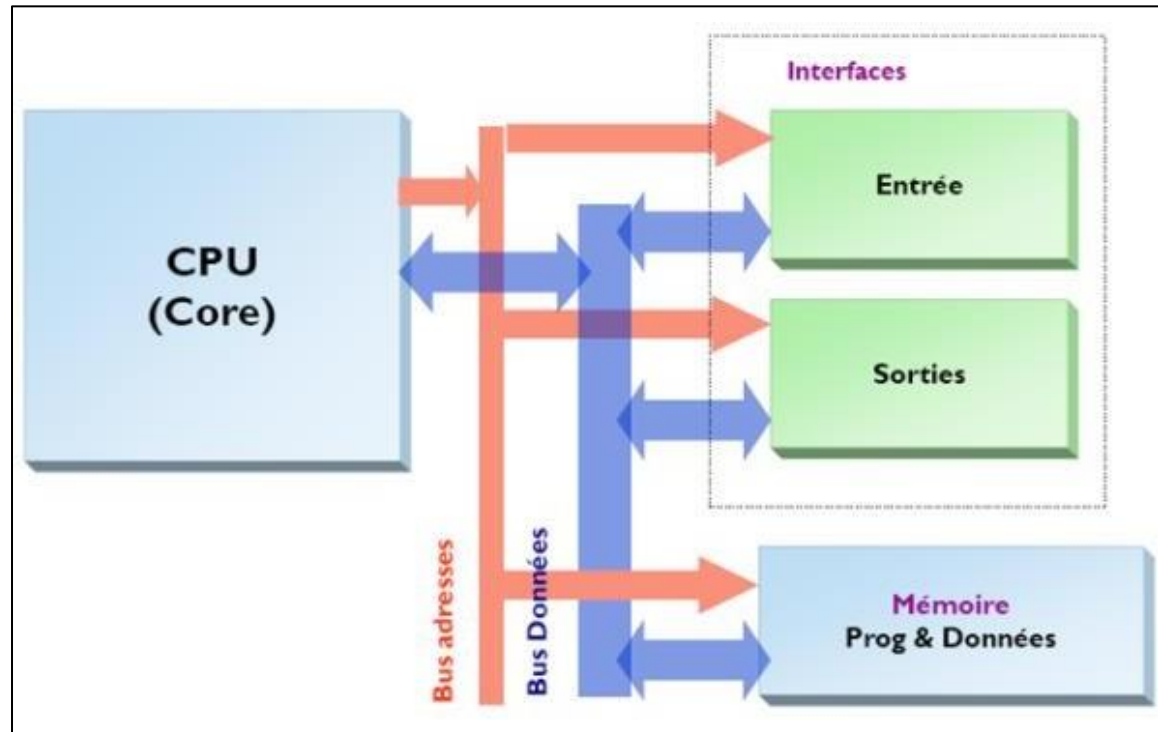


Les microprocesseurs utilisent, généralement, deux types d'architecture :

- ❖ Architecture Von Neumann
- ❖ Architecture Harvard

Architecture de Von Neumann

Architecture de Von Neumann: c'est une architecture d'ordinateur numérique dont la conception est basée sur le concept d'ordinateur inclut un seul espace de stockage (mémoire) pour stocker les données ainsi que le programme à exécuter. Cette architecture a été conçue par le célèbre mathématicien et physicien **John Von Neumann** en 1945.

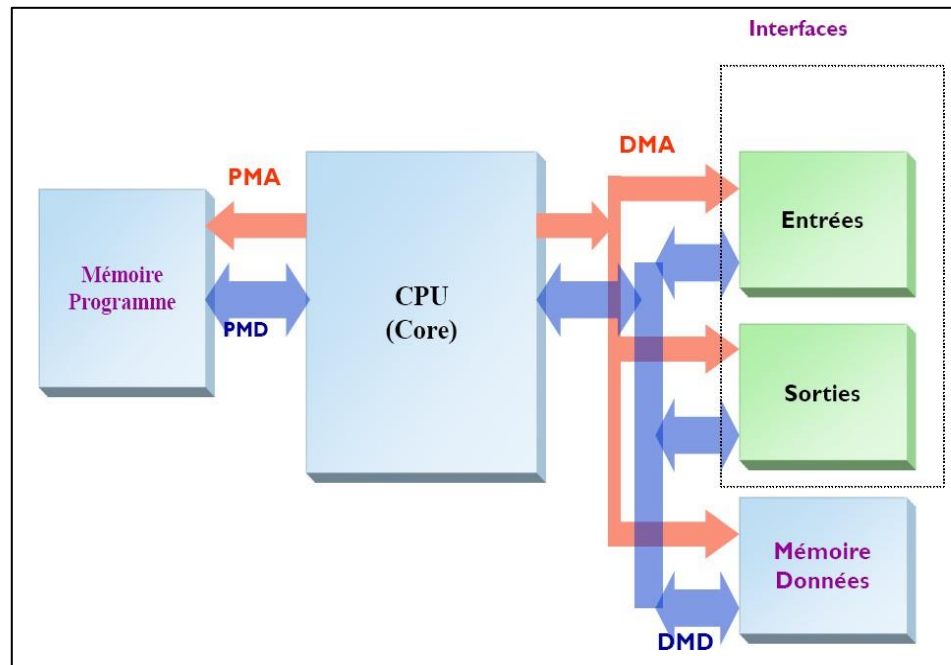


Architecture de VON NEUMANN

Architecture de Harvard

Architecture Harvard dispose de deux mémoires séparées pour stocker les données et le programme d'où l'existence de deux bus de données DMD (Data Memory Data) et PMD (Program Memory Data) ainsi que deux bus d'adresse (PMA et DMA). Le principal avantage d'avoir des bus séparés pour les instructions et les données est que le CPU peut accéder aux instructions et lire/écrire des données en même temps.

- Les exigences matérielles du Von Neumann sont moindres que celles de l'architecture Harvard alors que Harvard est plus rapide que l'architecture Von Neumann.



Architecture de HARVARD

Architectures de jeux d'instructions (SW)

La performance d'un microprocesseur est définie par trois facteurs clés :

- le nombre d'instructions I.
- le nombre de cycles d'horloge par instruction C.
- le temps d'un cycle d'horloge T.

Cette performance se calcule par : $I * C * T$. Les valeurs de I et C se distinguent suivant l'architecture des jeux d'instruction.

Nous distinguons, principalement, deux types:

- L'architecture **CISC (Complex Instruction Set Computer)** : I faible, C grand
- L'architecture **RISC (Reduced Instruction Set Computer)** : I élevé, C faible

Notion des architecture RISC et CISC

Architecture de jeu d'instructions Complexes CISC

L'approche CISC tente de minimiser le nombre d'instructions par programme, en sacrifiant le nombre de cycles par instruction. Les ordinateurs basés sur l'architecture CISC sont conçus pour diminuer le coût de la mémoire. En effet, les gros programmes ont besoin de plus de stockage, ce qui augmente le coût de la mémoire et les grosses mémoires deviennent plus chères. Pour résoudre ces problèmes, le nombre d'instructions par programme peut être réduit en intégrant le nombre d'opérations dans une seule instruction, ce qui rend les instructions plus complexes.(Exemple: 8086,68000...).

Architecture de jeu d'instructions réduites RISC

L'approche RISC est utilisée dans les appareils portables en raison de son efficacité énergétique. C'est un type d'architecture de microprocesseur qui utilise un ensemble d'instructions hautement optimisées. Le RISC fait l'inverse, en réduisant les cycles par instruction tout en augmentant le nombre d'instructions par programme. Il est réalisé en superposant l'exécution de plusieurs instructions à la manière d'un pipeline. Il présente un avantage de performance élevé par rapport au CISC.(Exemple : ARM, MIPS, SPARC...).

Notion des architectures RISC et CISC

Architecture RISC

- Instructions simples ne prenant qu'un seul cycle d'horloge.
- Instructions au format fixe.
- Décodeur simple (câblé).
- Beaucoup de registres.
- Seules les instructions LOAD et STORE ont accès à la mémoire.
- Peu de modes d'adressage.
- Compilateur complexe.

Architecture CISC

- Instructions complexes prenant plusieurs cycles d'horloge.
- Instructions au format variable décodeur complexe (microcode).
- Peu de registres.
- Toutes les instructions sont susceptibles d'accéder à la mémoire.
- Beaucoup de modes d'adressage.
- Compilateur simple.

Les processeurs **RISC** sont plus connus par leur **faible coût** et leur **efficacité énergétique**. Ce type d'architecture permet l'optimisation des performances d'un CPU puisque elle limite le nombre de cycles par instruction dans ses programmes.

Notion des architectures RISC et CISC

ARM

```
add r0, r0, r1
add r1, r1, #1
cmp r1, #11
bne loop
```

8086

```
Add [BX],300
```