Architecture des ordinateurs

Note de cours

T.Dumartin

1 Généralités

Chapitre
1

1.1 Introduction

L'informatique, contraction d'information et automatique, est la science du traitement de l'information. Apparue au milieu du 20ème siècle, elle a connu une évolution extrêmement rapide. A sa motivation initiale qui était de faciliter et d'accélérer le calcul, se sont ajoutées de nombreuses fonctionnalités, comme l'automatisation, le contrôle et la commande de processus, la communication ou le partage de l'information.

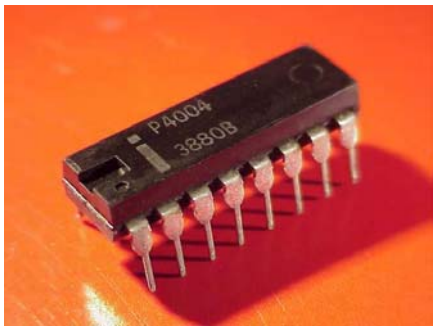
Le cours d'architecture des systèmes à microprocesseurs expose les principes de base du traitement programmé de l'information. La mise en œuvre de ces systèmes s'appuie sur deux modes de réalisation distincts, le matériel et le logiciel. Le matériel (**hardware**) correspond à l'aspect concret du système : unité centrale, mémoire, organes d'entrées-sorties, etc… Le logiciel (**software**) correspond à un ensemble d'instructions , appelé programme, qui sont contenues dans les différentes mémoires du système et qui définissent les actions effectuées par le matériel.

1.2 Qu'entend-t-on par architecture ?

L'architecture d'un système à microprocesseur représente l'organisation de ses différentes unités et de leurs interconnexions. Le choix d'une architecture est toujours le résultat d'un compromis :

- entre performances et coûts
- entre efficacité et facilité de construction
- entre performances d'ensemble et facilité de programmation
- etc …

1.3 Qu'est ce qu'un microprocesseur ?



Un microprocesseur est un circuit intégré complexe. Il résulte de l'intégration sur une puce de fonctions logiques combinatoires (logiques et/ou arithmétique) et séquentielles (registres, compteur, etc…). Il est capable d'interpréter et d'exécuter les instructions d'un programme. Son domaine d'utilisation est donc presque illimité.

Le concept de microprocesseur a été créé par la Société Intel. Cette Société, créée en 1968, était spécialisée dans la conception et la fabrication de puces mémoire. À la demande de deux de ses clients — fabricants de calculatrices et de terminaux — Intel étudia une unité de calcul implémentée sur une seule puce. Ceci donna naissance, en 1971, au premier microprocesseur, le 4004, qui était une unité de calcul 4 bits fonctionnant à 108 kHz. Il résultait de l'intégration d'environ 2300 transistors.

Remarques :

La réalisation de circuits intégrés de plus en plus complexe a été rendue possible par l'apparition du transistor en 1947. Il a fallu attendre 1958 pour voir apparaître le 1^{ier} circuit intégré réalisé par Texas Instrument.

1.4 Rappels

Les informations traitées par un microprocesseur sont de différents types (nombres, instructions, images, vidéo, etc…) mais elles sont toujours représentées sous un format binaire. Seul le codage changera suivant les différents types de données à traiter. Elles sont représentées physiquement par 2 niveaux de tensions différents.

En binaire, une information élémentaire est appelé **bit** et ne peut prendre que deux valeurs différentes : **0** ou **1**.

Une information plus complexe sera codée sur plusieurs bit. On appelle cet ensemble un **mot**. Un mot de 8 bits est appelé un **octet**.

Représentation d'un nombre entier en binaire :

Les nombres sont exprimés par des chiffres pouvant prendre deux valeurs 0 ou 1. A chaque chiffre est affecté un poids exprimé en puissance de 2.

Ex : $(101)_2$ <> $1x\,2^2 + 0x2^1 + 1x2^0 = (5)_{10}$

Représentation d'un nombre entier en hexadécimal :

Lorsqu'une donnée est représentée sur plus de 4 bits, on préfère souvent l'exprimer en hexadécimal. Les nombres sont exprimés par des chiffres et des lettres pouvant prendre 16 valeurs :

0 1 2 3 4 5 6 7 8 9 A B C D E F

A chaque chiffre est affecté un poids exprimé en puissance de 16.

Ex : $(9A)_{16}$ <> $9x16^1 + Ax16^0 = 9x16^1 + 10x16^0 = (154)_{10}$

Attention !! :

1 kilobit = 2^{10} bit = 1024 bit
1 mégabit = 2^{10} kbit = 1024 kbit
1 gigabit = 2^{10} Mbit = 1024 Mbit

1.5 Où trouve-t-on des systèmes à microprocesseur ?

Les applications des systèmes à microprocesseurs sont multiples et variées :

- Ordinateur, PDA
- console de jeux
- calculatrice
- télévision
- téléphone portable
- distributeur automatique d'argent
- robotique
- lecteur carte à puce, code barre
- automobile
- instrumentation
- etc…








2 Architecture de base

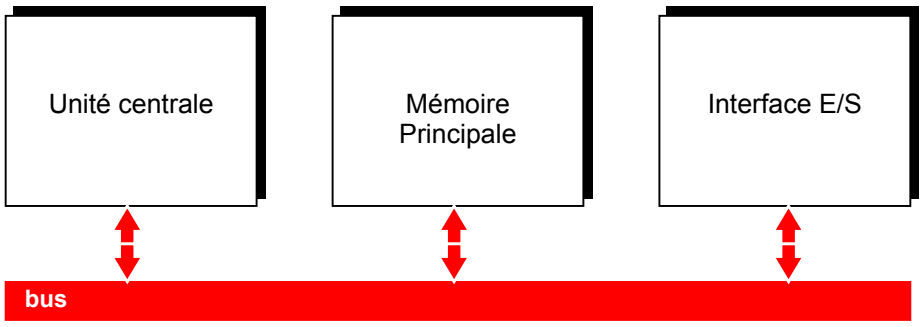
Chapitre 2

2.1 Modèle de von Neumann

Pour traiter une information, un microprocesseur seul ne suffit pas, il faut l'insérer au sein d'un système minimum de traitement programmé de l'information. John Von Neumann est à l'origine d'un modèle de machine universelle de traitement programmé de l'information (1946). Cette architecture sert de base à la plupart des systèmes à microprocesseur actuel. Elle est composé des éléments suivants :

- une unité centrale
- une mémoire principale
- des interfaces d'entrées/sorties

Les différents organes du système sont reliés par des voies de communication appelées **bus**.



2.2 L'unité centrale

Elle est composée par le microprocesseur qui est chargé d'interpréter et d'exécuter les instructions d'un programme, de lire ou de sauvegarder les résultats dans la mémoire et de communiquer avec les unités d'échange. Toutes les activités du microprocesseur sont cadencées par une horloge.

On caractérise le microprocesseur par :

- sa fréquence d'horloge : en MHz ou GHz
- le nombre d'instructions par secondes qu'il est capable d'exécuter : en MIPS
- la taille des données qu'il est capable de traiter : en bits

2.3 La mémoire principale

Elle contient les instructions du ou des programmes en cours d'exécution et les données associées à ce programme. Physiquement, elle se décompose souvent en :

- une mémoire morte (**ROM** = Read Only Memory) chargée de stocker le programme. C'est une mémoire à lecture seule.

- une mémoire vive (**RAM** = Random Access Memory) chargée de stocker les données intermédiaires ou les résultats de calculs. On peut lire ou écrire des données dedans, ces données sont perdues à la mise hors tension.

Remarque :
Les disques durs, disquettes, CDROM, etc… sont des périphériques de stockage et sont considérés comme des mémoires secondaires.

2.4 Les interfaces d'entrées/sorties

Elles permettent d'assurer la communication entre le microprocesseur et les périphériques. (capteur, clavier, moniteur ou afficheur, imprimante, modem, etc…).

2.5 Les bus

Un bus est un ensemble de fils qui assure la transmission du même type d'information. On retrouve trois types de bus véhiculant des informations en parallèle dans un système de traitement programmé de l'information :

- **un bus de données** : bidirectionnel qui assure le transfert des informations entre le microprocesseur et son environnement, et inversement. Son nombre de lignes est égal à la capacité de traitement du microprocesseur.

- **un bus d'adresses**: unidirectionnel qui permet la sélection des informations à traiter dans un *espace mémoire* (ou *espace adressable*) qui peut avoir 2^n emplacements, avec n = nombre de conducteurs du bus d'adresses.

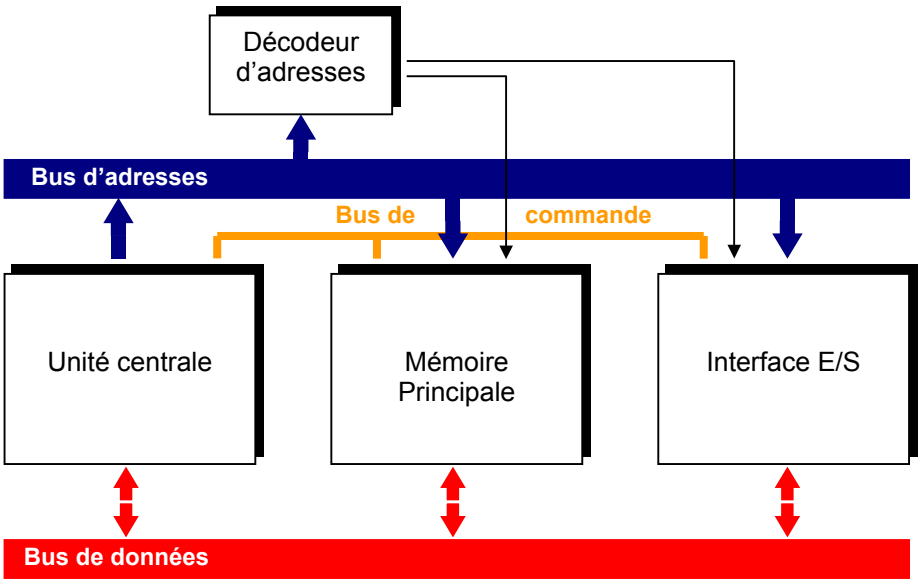
- **un bus de commande**: constitué par quelques conducteurs qui assurent la synchronisation des flux d'informations sur les bus des données et des adresses.

2.6 Décodage d'adresses

La multiplication des périphériques autour du microprocesseur oblige la présence d'un **décodeur d'adresse** chargé d'aiguiller les données présentes sur le bus de données.

En effet, le microprocesseur peut communiquer avec les différentes mémoires et les différents boîtier d'interface. Ceux-ci sont tous reliés sur le même bus de données et afin d'éviter des conflits, un seul composant doit être sélectionné à la fois.

Lorsqu'on réalise un système microprogrammé, on attribue donc à chaque périphérique une zone d'adresse et une fonction « décodage d'adresse » est donc nécessaire afin de fournir les signaux de sélection de chacun des composants.



Remarque : lorsqu'un composant n'est pas sélectionné, ses sorties sont mises à l'état « haute impédance » afin de ne pas perturber les données circulant sur le bus. (elle présente une impédance de sortie très élevée = circuit ouvert).

3 Les mémoires

Chapitre 3

Une mémoire est un circuit à semi-conducteur permettant d'enregistrer, de conserver et de restituer des informations (instructions et variables). C'est cette capacité de mémorisation qui explique la polyvalence des systèmes numériques et leur adaptabilité à de nombreuses situations. Les informations peuvent être écrites ou lues. Il y a écriture lorsqu'on enregistre des informations en mémoire, lecture lorsqu'on récupère des informations précédemment enregistrées.

3.1 Organisation d'une mémoire

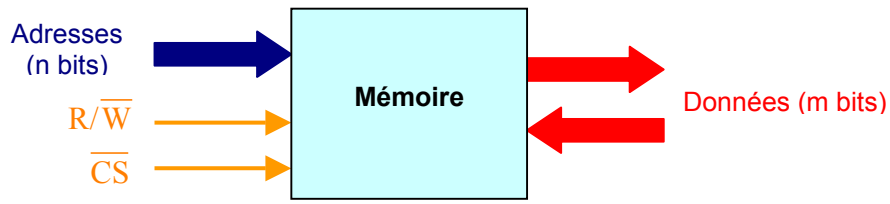
Une mémoire peut être représentée comme une armoire de rangement constituée de différents tiroirs. Chaque tiroir représente alors une case mémoire qui peut contenir un seul élément : des **données**. Le nombre de cases mémoires pouvant être très élevé, il est alors nécessaire de pouvoir les identifier par un numéro. Ce numéro est appelé **adresse**. Chaque donnée devient alors accessible grâce à son adresse

Adresse	Case mémoire
7 = 111	
6 = 110	
5 = 101	
4 = 100	
3 = 011	
2 = 010	
1 = 001	
0 = 000	0001 1010

Avec une adresse de *n* bits il est possible de référencer au plus 2^n cases mémoire. Chaque case est remplie par un mot de données (sa longueur *m* est toujours une puissance de 2). Le nombre de fils d'adresses d'un boîtier mémoire définit donc le nombre de cases mémoire que comprend le boîtier. Le nombre de fils de données définit la taille des données que l'on peut sauvegarder dans chaque case mémoire.

En plus du bus d'adresses et du bus de données, un boîtier mémoire comprend une entrée de commande qui permet de définir le type d'action que l'on effectue avec la mémoire (lecture/écriture) et une entrée de sélection qui permet de mettre les entrées/sorties du boîtier en haute impédance.

On peut donc schématiser un circuit mémoire par la figure suivante où l'on peut distinguer :



- les entrées d'adresses
- les entrées de données
- les sorties de données
- les entrées de commandes :
  - une entrée de sélection de lecture ou d'écriture. ($R/\overline{W}$)
  - une entrée de sélection du circuit. ($\overline{CS}$)

Une opération de lecture ou d'écriture de la mémoire suit toujours le même cycle :

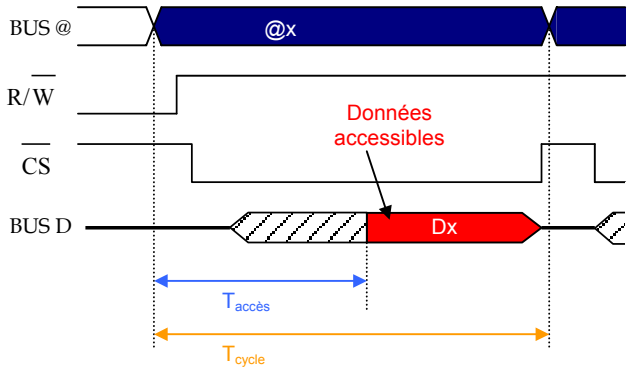
1. sélection de l'adresse
2. choix de l'opération à effectuer (R/$\overline{W}$)
3. sélection de la mémoire ($\overline{CS}$ = 0)
4. lecture ou écriture la donnée

Remarque :
Les entrées et sorties de données sont très souvent regroupées sur des bornes bidirectionnelles.

3.2 Caractéristiques d'une mémoire

- **La capacité :** c'est le nombre total de bits que contient la mémoire. Elle s'exprime aussi souvent en octet.
- **Le format des données :** c'est le nombre de bits que l'on peut mémoriser par case mémoire. On dit aussi que c'est la largeur du mot mémorisable.
- **Le temps d'accès :** c'est le 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.
- **Le temps de cycle :** il représente l'intervalle minimum qui doit séparer deux demandes successives de lecture ou d'écriture.
- **Le débit :** c'est le nombre maximum d'informations lues ou écrites par seconde.
- **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.

Exemple : Chronogramme d'un cycle de lecture



Remarque :
Les mémoires utilisées pour réaliser la mémoire principale d'un système à microprocesseur sont des mémoires à semi-conducteur. On a vu que dans ce type de mémoire, on accède directement à n'importe quelle information dont on connaît l'adresse et que le temps mis pour obtenir cette information ne dépend pas de l'adresse. On dira que l'accès à une telle mémoire est aléatoire ou direct.

A l'inverse, pour accéder à une information sur bande magnétique, il faut dérouler la bande en repérant tous les enregistrements jusqu'à ce que l'on trouve celui que l'on désire. On dit alors que l'accès à l'information est séquentiel. Le temps d'accès est variable selon la position de l'information recherchée. L'accès peut encore être semi-séquentiel : combinaison des accès direct et séquentiel. Pour un disque magnétique par exemple l'accès à la piste est direct, puis l'accès au secteur est séquentiel.

3.3 Différents types de mémoire

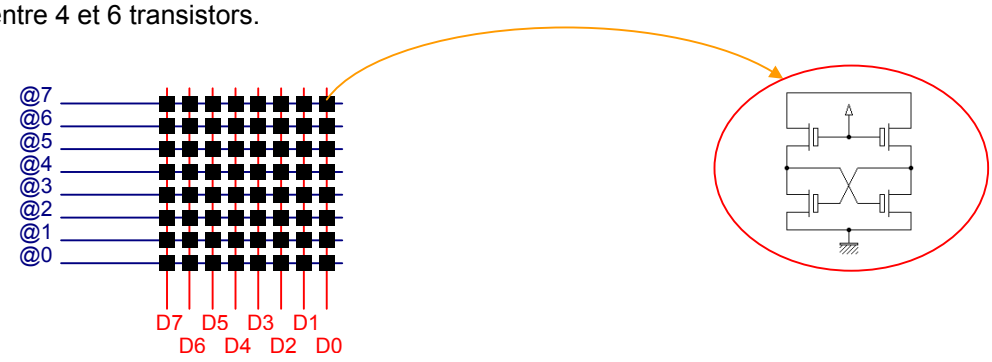
3.3.1 Les mémoires vives (RAM)

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 Acces Memory : mémoire à accès aléatoire) :

- Les RAM statiques
- Les RAM dynamiques

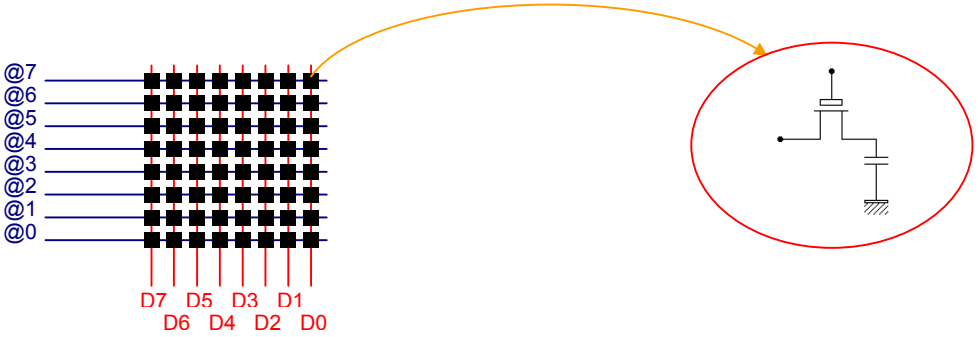
3.3.1.1 Les RAM statiques

Le bit mémoire d'une RAM statique (SRAM) est composé d'une bascule. Chaque bascule contient entre 4 et 6 transistors.



3.3.1.2 Les RAM dynamiques

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



- **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.

- **Inconvénients :**

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. Ce rafraîchissement indispensable a plusieurs conséquences :

- il complique la gestion des mémoires dynamiques car il faut tenir compte des actions de rafraîchissement qui sont prioritaires.
- la durée de ces actions augmente le temps d'accès aux informations.

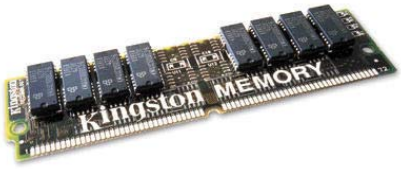
D'autre part, la lecture de l'information est destructive. En effet, elle se fait par décharge de la capacité du point mémoire lorsque celle-ci est chargée. Donc toute lecture doit être suivie d'une réécriture.

3.3.1.3 Conclusions

En général les mémoires dynamiques, qui offrent une plus grande densité d'information et un coût par bit plus faible, sont utilisées pour la mémoire centrale, alors que les mémoires statiques, plus rapides, sont utilisées lorsque le facteur vitesse est critique, notamment pour des mémoires de petite taille comme les caches et les registres.

Remarques :

Voici un historique de quelques DRAM qui ont ou sont utilisées dans les PC :



- **La DRAM FPM** (Fast Page Mode, 1987) : Elle permet d'accéder plus rapidement à des données en introduisant la notion de page mémoire. (33 à 50 Mhz)

- **La DRAM EDO** (Extended Data Out, 1995) : Les composants de cette mémoire permettent de conserver plus longtemps l'information, on peut donc ainsi espacer les cycles de rafraîchissement. Elle apporte aussi la possibilité d'anticiper sur le prochain cycle mémoire. (33 à 50 Mhz)

- **La DRAM BEDO** (Bursted EDO) : On n'adresse plus chaque unité de mémoire individuellement lorsqu'il faut y lire ou y écrire des données. On se contente de transmettre l'adresse de départ du processus de lecture/écriture et la longueur du bloc de données (Burst). Ce procédé permet de gagner beaucoup de temps, notamment avec les grands paquets de données tels qu'on en manipule avec les applications modernes. (66 Mhz)

- **La Synchronous DRAM** (SDRAM, 1997) : La mémoire SDRAM a pour particularité de se synchroniser sur une horloge. Les mémoires FPM, EDO étaient des mémoires asynchrones et elle induisaient des temps d'attentes lors de la synchronisation. Elle se compose en interne de deux bancs de mémoire et des données peuvent être lues alternativement sur l'un puis sur l'autre de ces bancs grâce à un procédé d'entrelacement spécial. Le protocole d'attente devient donc tout à fait inutile. Cela lui permet de supporter des fréquences plus élevées qu'avant (100 Mhz).

- **La DDR-I ou DDR-SDRAM** (Double Data Rate Synchronous DRAM, 2000) : La DDR-SDRAM permet de recevoir ou d'envoyer des données lors du front montant et du front descendant de l'horloge. (133 à 200 MHz)

3.3.2 Les mémoires mortes (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 lue. 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 :

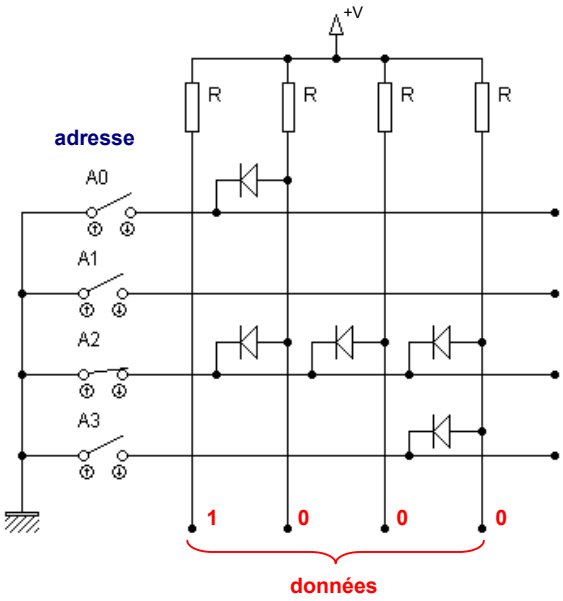
- ROM
- PROM
- EPROM
- EEPROM
- FLASH EPROM.

3.3.2.1 LA ROM

Elle est programmée par le fabricant et son contenu ne peut plus être ni modifié., ni effacé par l'utilisateur.

- **Structure :**

Cette mémoire est composée d'une matrice dont la programmation s'effectue en reliant les lignes aux colonnes par des diodes. L'adresse permet de sélectionner une ligne de la matrice et les données sont alors reçues sur les colonnes (le nombre de colonnes fixant la taille des mots mémoire).



- **Programmation :**

L'utilisateur doit fournir au constructeur un *masque* indiquant les emplacements des diode dans matrice.

- **Avantages :**

► Densité élevée
► Non volatile
► Mémoire rapide

- **Inconvénients :**

► Écriture impossible
► Modification impossible (toute erreur est fatale).
► Délai de fabrication .(3 à 6 semaines)
► Obligation de grandes quantités en raison du coût élevé qu'entraîne la production du masque et le processus de fabrication.

3.3.2.2 La PROM

C'est une ROM qui peut être programmée une seule fois par l'utilisateur (Programmable ROM). La programmation est réalisée à partir d'un programmateur spécifique.

- **Structure :**

Les liaisons à diodes de la ROM sont remplacées par des fusibles pouvant être détruits ou des jonctions pouvant être court-circuitées.

- **Programmation :**

Les PROM à fusible sont livrées avec toutes les lignes connectées aux colonnes (0 en chaque point mémoire). Le processus de programmation consiste donc à programmer les emplacements des ''1'' en générant des impulsions de courants par l'intermédiaire du programmateur ; les fusibles situés aux points mémoires sélectionnés se retrouvant donc détruits.

Le principe est identique dans les PROM à jonctions sauf que les lignes et les colonnes sont déconnectées (1 en chaque point mémoire). Le processus de programmation consiste donc à programmer les emplacements des ''0'' en générant des impulsions de courants par l'intermédiaire du programmateur ; les jonctions situées aux points mémoires sélectionnés se retrouvant court-circuitées par effet d'avalanche.

- **Avantages :**

► idem ROM
► Claquage en quelques minutes
► Coût relativement faible

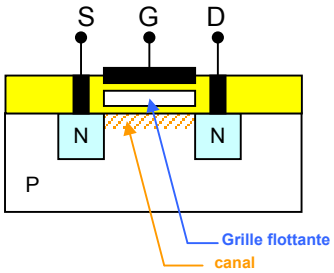
- **Inconvénients :**

► Modification impossible (toute erreur est fatale).

3.3.2.3 L'EPROM ou UV-EPROM

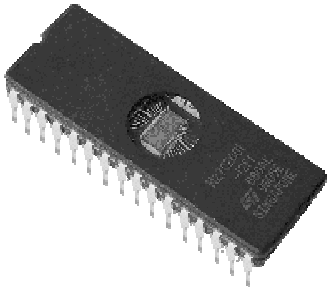
Pour faciliter la mise au point d'un programme ou tout simplement permettre une erreur de programmation, il est intéressant de pouvoir reprogrammer une PROM. La technique de claquage utilisée dans celles-ci ne le permet évidemment pas. L'EPROM *(*Erasable Programmable ROM*)* est une PROM qui peut être effacée.

- **Structure**



Dans une EPROM, le point mémoire est réalisé à partir d'un transistor FAMOS (Floating gate Avalanche injection Metal Oxyde Silicium). Ce transistor MOS a été introduit par Intel en 1971 et a la particularité de posséder une grille flottante.

- **Programmation**



La programmation consiste à piéger des charges dans la grille flottante. Pour cela, il faut tout d'abord appliquer une très forte tension entre Grille et Source. Si l'on applique ensuite une tension entre D et S, la canal devient conducteur. Mais comme la tension Grille-Source est très importante, les électrons sont déviés du canal vers la grille flottante et capturés par celle-ci. Cette charge se maintient une dizaine d'années en condition normale.

L'exposition d'une vingtaine de minutes à un rayonnement ultra-violet permet d'annuler la charge stockée dans la grille flottante. Cet effacement est reproductible plus d'un millier de fois. Les boîtiers des EPROM se caractérise donc par la présence d'une petite fenêtre transparente en quartz qui assure le passage des UV. Afin d'éviter toute perte accidentelle de l'information, il faut obturer la fenêtre d'effacement lors de l'utilisation.

- **Avantages :**

► Reprogrammable et non Volatile

- **Inconvénients :**

► Impossible de sélectionner une seule cellule à effacer
► Impossible d'effacer la mémoire in-situ.
► l'écriture est beaucoup plus lente que sur une RAM. (environ 1000x)

3.3.2.4 L'EEPROM

L'EEPROM (Electically EPROM) est une mémoire programmable et effaçable électriquement. Elle répond ainsi à l'inconvénient principal de l'EPROM et peut être programmée in situ.

- **Structure**

Dans une EEPROM, le point mémoire est réalisé à partir d'un transistor SAMOS reprenant le même principe que le FAMOS sauf que l'épaisseur entre les deux grilles est beaucoup plus faible.

- **Programmation**

Une forte tension électrique appliquée entre grille et source conduit à la programmation de la mémoire. Une forte tension inverse provoquera la libération des électrons et donc l'effacement de la mémoire.

- **Avantages :**

► Comportement d'une RAM non Volatile.
► Programmation et effacement mot par mot possible.

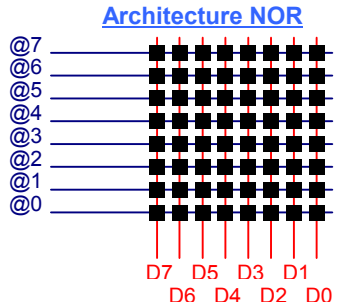
- **Inconvénients :**

► Très lente pour une utilisation en RAM.
► Coût de réalisation.

3.3.2.5 La FLASH EPROM

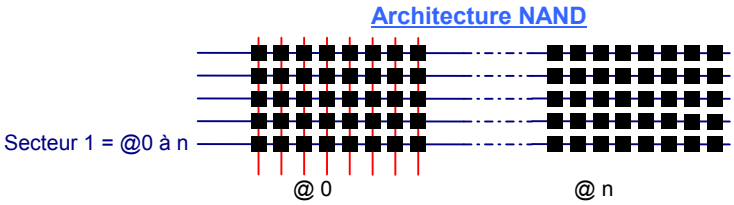
La mémoire Flash s'apparente à la technologie de l'EEPROM. Elle est programmable et effaçable électriquement comme les EEPROM.

- **Structure**



Il existe deux technologie différentes qui se différencient par l'organisation de leurs réseaux mémoire : l'architecture NOR et NAND. L'architecture NOR propose un assemblage des cellules élémentaires de mémorisation en parallèle avec les lignes de sélection comme dans une EEPROM classique. L'architecture NAND propose un assemblage en série de ces mêmes cellules avec les lignes de sélection. D'un point de vue pratique, la différence majeure entre NOR et NAND tient à leurs interfaces. Alors qu'une NOR dispose de bus d'adresses et de données dédiés, la NAND est dotée d'une interface d'E/S indirecte. . Par contre, la structure NAND autorise une implantation plus dense grâce à une taille de cellule approximativement 40 % plus petite que la structure NOR.

- **Programmation**



Si NOR et NAND exploitent toutes deux le même principe de stockage de charges dans la grille flottante d'un transistor, l'organisation de leur réseau mémoire n'offre pas la même souplesse d'utilisation. Les Flash NOR autorise un adressage aléatoire qui permet de la programmer octet par octet alors que la Flash NAND autorise un accès séquentiel aux données et permettra seulement une programmation par secteur comme sur un disque dur.

- **Avantages**

Flash NOR :

► Comportement d'une RAM non Volatile.
► Programmation et effacement mot par mot possible.
► Temps d'accès faible.

Flash NAND :

- Comportement d'une RAM non Volatile.
- Forte densité d'intégration ➡ coût réduit.
- Rapidité de l'écriture/lecture par paquet
- Consommation réduite.

  - **Inconvénients**

Flash NOR :

- Lenteur de l'écriture/lecture par paquet.
- coût.

Flash NAND :
- Ecriture/lecture par octet impossible.
- Interface E/S indirecte

La Flash EPROM a connu un essor très important ces dernières années avec le boom de la téléphonie portable et des appareils multimédia (PDA, appareil photo numérique, lecteur MP3, etc...).

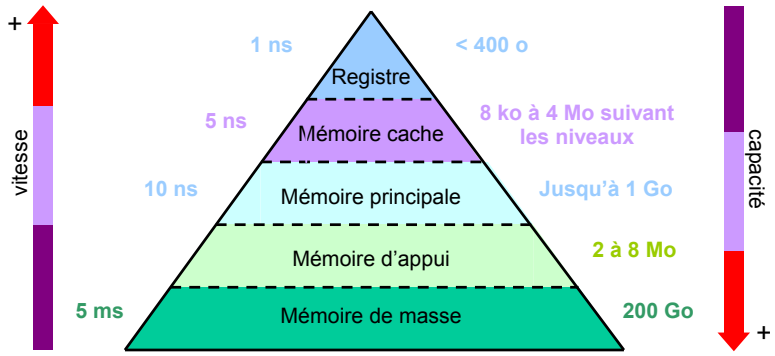
3.4 Critères de choix d'une mémoire

Les principaux critères à retenir sont :

- capacité
- vitesse
- consommation
- coût

3.5 Notion de hiérarchie mémoire

Une mémoire idéale serait une mémoire de grande capacité, capable de stocker un maximum d'informations et possédant un temps d'accès très faible afin de pouvoir travailler rapidement sur ces informations. Mais il se trouve que les mémoires de grande capacité sont souvent très lente et que les mémoire rapides sont très chères. Et pourtant, la vitesse d'accès à la mémoire conditionne dans une large mesure les performances d'un système. En effet, c'est là que se trouve le *goulot d'étranglement* entre un microprocesseur capable de traiter des informations très rapidement et une mémoire beaucoup plus lente (ex : processeur actuel à 3Ghz et mémoire à 400MHz). Or, on n'a jamais besoin de toutes les informations au même moment. Afin d'obtenir le meilleur compromis coût-performance, on définie donc une hiérarchie mémoire. On utilise des mémoires de faible capacité mais très rapide pour stocker les informations dont le microprocesseur se sert le plus et on utilise des mémoires de capacité importante mais beaucoup plus lente pour stocker les informations dont le microprocesseur se sert le moins. Ainsi, plus on s'éloigne du microprocesseur et plus la capacité et le temps d'accès des mémoire vont augmenter.



- **Les registres** sont les éléments de mémoire les plus rapides. Ils sont situés au niveau du processeur et servent au stockage des opérandes et des résultats intermédiaires.
- **La mémoire cache** est une mémoire rapide de faible capacité destinée à accélérer l'accès à la mémoire centrale en stockant les données les plus utilisées.
- **La mémoire principale** est l'organe principal de rangement des informations. Elle contient les programmes (instructions et données) et est plus lente que les deux mémoires précédentes.
- **La mémoire d'appui** sert de mémoire intermédiaire entre la mémoire centrale et les mémoires de masse. Elle joue le même rôle que la mémoire cache.
- **La mémoire de masse** est une mémoire périphérique de grande capacité utilisée pour le stockage permanent ou la sauvegarde des informations. Elle utilise pour cela des supports magnétiques (disque dur, ZIP) ou optiques (CDROM, DVDROM).

4 Le microprocesseur

Chapitre 4

Un microprocesseur est un circuit intégré complexe caractérisé par une très grande intégration et doté des facultés d'interprétation et d'exécution des instructions d'un programme. Il est chargé d'organiser les tâches précisées par le programme et d'assurer leur exécution. Il doit aussi prendre en compte les informations extérieures au système et assurer leur traitement. C'est le cerveau du système.

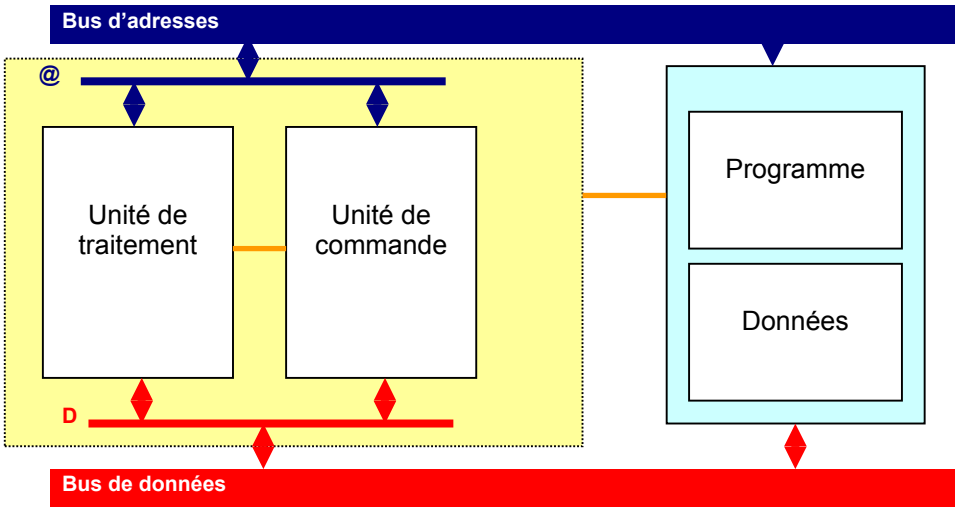
A l'heure actuelle, un microprocesseur regroupe sur quelques millimètre carré des fonctionnalités toujours plus complexes. Leur puissance continue de s'accroître et leur encombrement diminue régulièrement respectant toujours, pour le moment, la fameuse *loi de Moore* (1).

4.1 Architecture de base d'un microprocesseur

Un microprocesseur est construit autour de deux éléments principaux :

- Une unité de commande
- Une unité de traitement

associés à des registres chargées de stocker les différentes informations à traiter. Ces trois éléments sont reliés entre eux par des bus interne permettant les échanges d'informations.



Remarques :
Il existe deux types de registres :

- les registres d'usage général permettent à l'unité de traitement de manipuler des données à vitesse élevée. Ils sont connectés au bus données interne au microprocesseur.
- les registres d'adresses (pointeurs) connectés sur le bus adresses.

4.1.1 L'unité de commande

Elle permet de séquencer le déroulement des instructions. Elle effectue la recherche en mémoire de l'instruction. Comme chaque instruction est codée sous forme binaire, elle en assure le décodage pour enfin réaliser son exécution puis effectue la préparation de l'instruction suivante. Pour cela, elle est composée par :

- **le compteur de programme** constitué par un registre dont le contenu est initialisé avec l'adresse de la première instruction du programme. Il contient toujours l'adresse de l'instruction à exécuter.

(1) Moore (un des co-fondateurs de la société Intel) a émis l'hypothèse que les capacités technologiques permettraient de multiplier par 2 tous les 18 mois le nombre de transistors intégrés sur les circuits.

- **le registre d'instruction et le décodeur d'instruction :** chacune des instructions à exécuter est rangée dans le registre instruction puis est décodée par le décodeur d'instruction.

- **Bloc logique de commande (ou séquenceur) :** Il organise l'exécution des instructions au rythme d'une horloge. Il élabore tous les signaux de synchronisation internes ou externes (bus de commande) du microprocesseur en fonction des divers signaux de commande provenant du décodeur d'instruction ou du registre d'état par exemple. Il s'agit d'un automate réalisé soit de façon câblée (obsolète), soit de façon micro-programmée, on parle alors de micro-microprocesseur.

4.1.2 L'unité de traitement

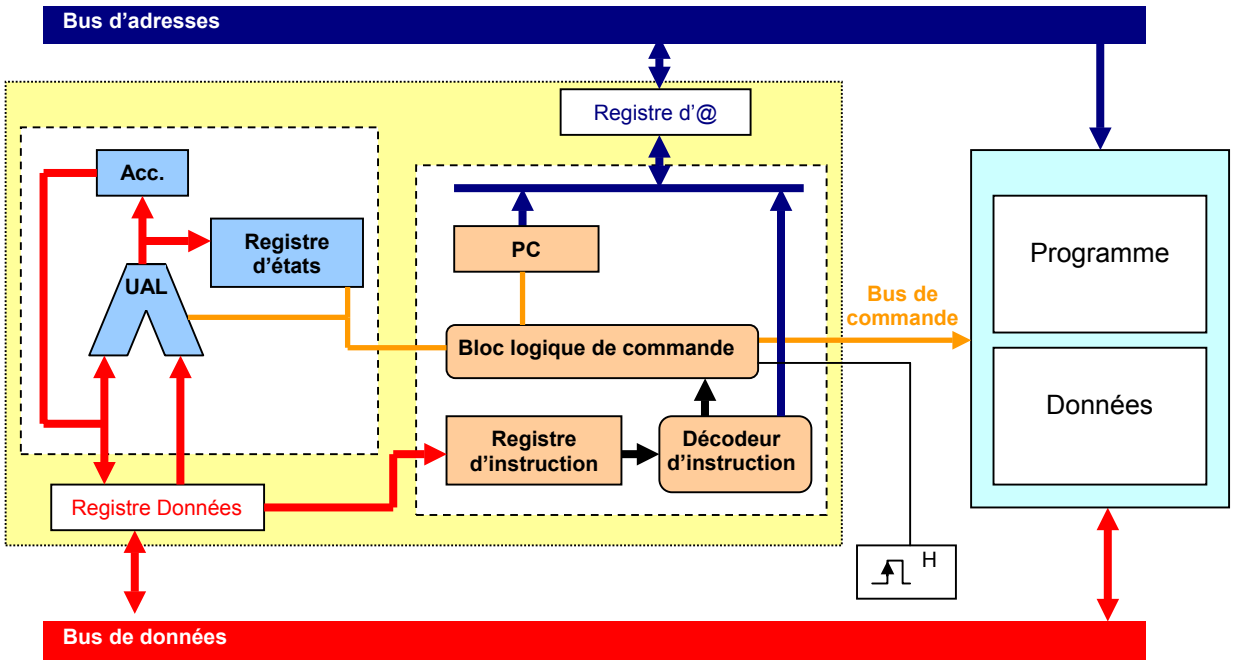
C'est le cœur du microprocesseur. Elle regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions :

- **L'Unité Arithmétique et Logique (UAL)** est un circuit complexe qui assure les fonctions logiques (ET, OU, Comparaison, Décalage , etc…) ou arithmétique (Addition, soustraction).

- **Le registre d'état** est généralement composé de 8 bits à considérer individuellement. Chacun de ces bits est un indicateur dont l'état dépend du résultat de la dernière opération effectuée par l'UAL. On les appelle *indicateur d'état* ou *flag* ou *drapeaux*. Dans un programme le résultat du test de leur état conditionne souvent le déroulement de la suite du programme. On peut citer par exemple les indicateurs de :
  - retenue **(carry : C)**
  - retenue intermédiaire **(Auxiliary-Carry : AC)**
  - signe **(Sign : S)**
  - débordement **(overflow : OV ou V)**
  - zéro **(Z)**
  - parité **(Parity : P)**

- **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.

4.1.3 Schéma fonctionnel

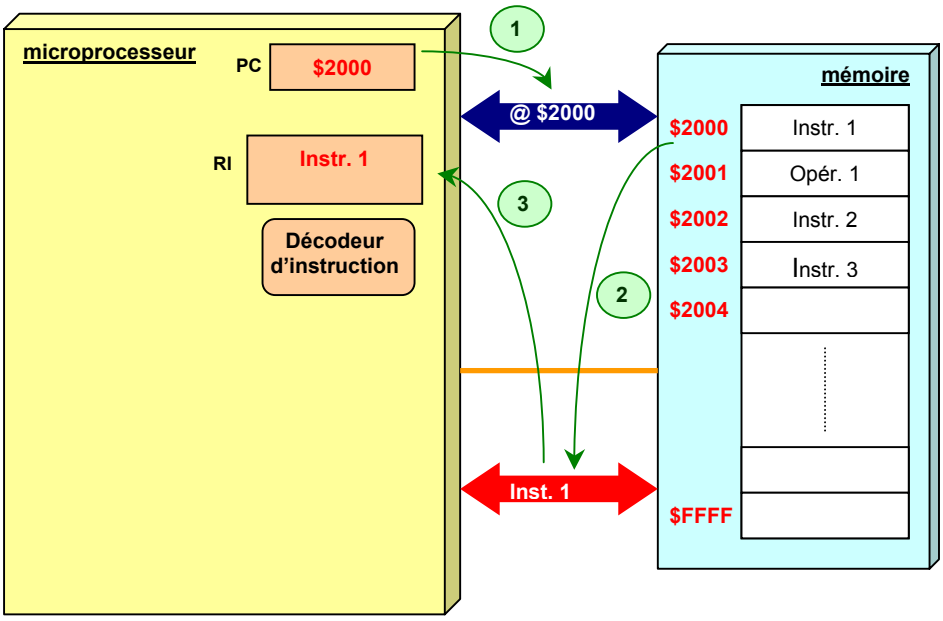


4.2 Cycle d'exécution d'une instruction

Le microprocesseur ne comprend qu'un certain nombre d'instructions qui sont codées en binaire. Le traitement d'une instruction peut être décomposé en trois phases.

- **Phase 1:** *Recherche de l'instruction à traiter*

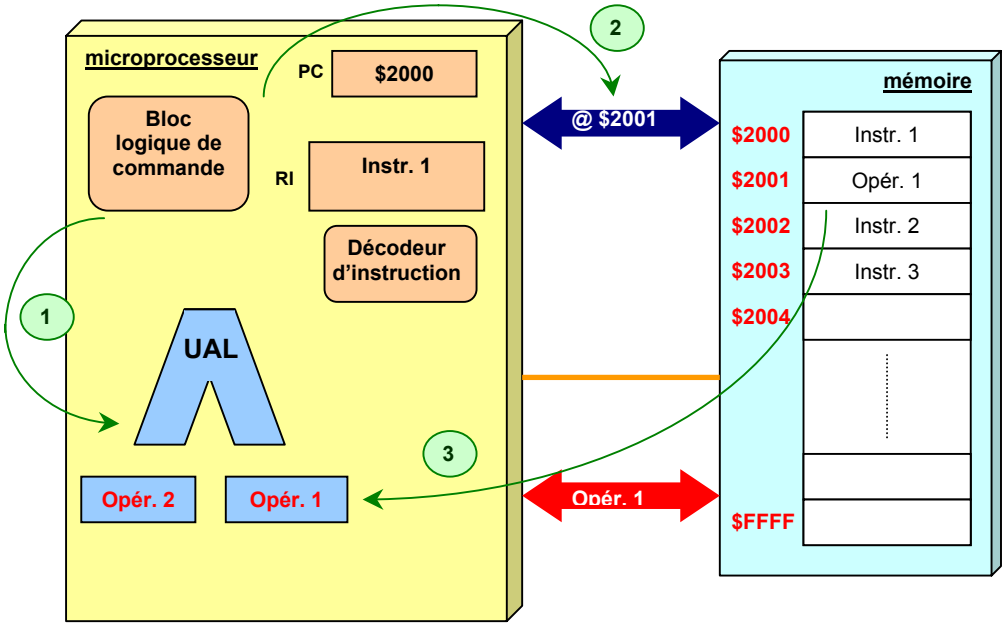
1. Le PC contient l'adresse de l'instruction suivante du programme. Cette valeur est placée sur le bus d'adresses par l'unité de commande qui émet un ordre de lecture.
2. Au bout d'un certain temps (temps d'accès à la mémoire), le contenu de la case mémoire sélectionnée est disponible sur le bus des données.
3. L'instruction est stockée dans le registre instruction du processeur.



- **Phase 2 :** *Décodage de l'instruction et recherche de l'opérande*

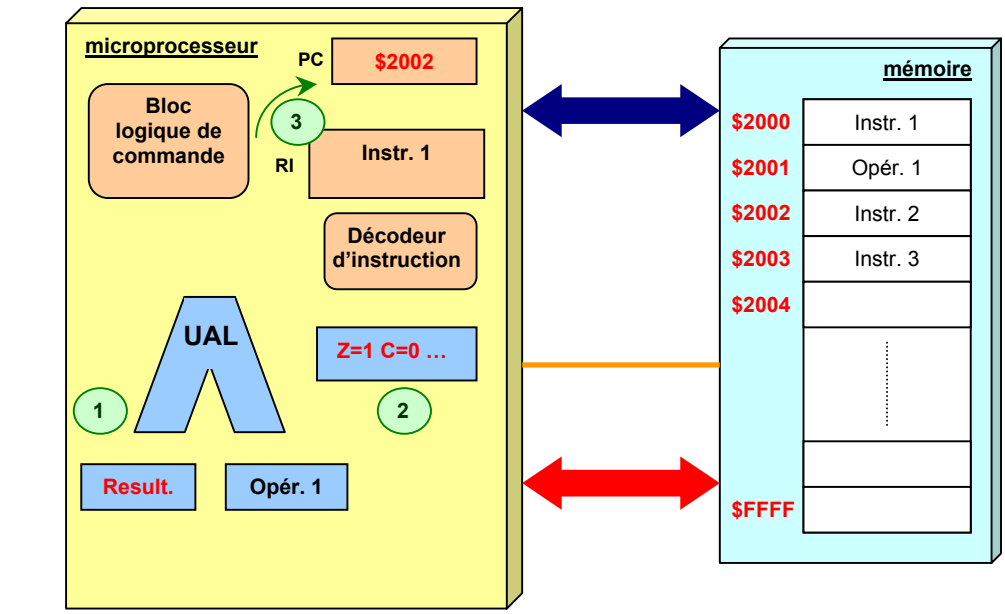
Le registre d'instruction contient maintenant le premier mot de l'instruction qui peut être codée sur plusieurs mots. Ce premier mot contient le code opératoire qui définit la nature de l'opération à effectuer (addition, rotation,...) et le nombre de mots de l'instruction.

1. L'unité de commande transforme l'instruction en une suite de commandes élémentaires nécessaires au traitement de l'instruction.
2. Si l'instruction nécessite une donnée en provenance de la mémoire, l'unité de commande récupère sa valeur sur le bus de données.
3. L'opérande est stockée dans un registre.



- **Phase 3 :** *Exécution de l'instruction*

1. Le micro-programme réalisant l'instruction est exécuté.
2. Les drapeaux sont positionnés (*registre d'état*).
3. L'unité de commande positionne le PC pour l'instruction suivante.



4.3 Jeu d'instructions

4.3.1 Définition

La première étape de la conception d'un microprocesseur est la définition de son jeu d'instructions. Le jeu d'instructions décrit l'ensemble des opérations élémentaires que le microprocesseur pourra exécuter. Il va donc en partie déterminer l'architecture du microprocesseur à réaliser et notamment celle du séquenceur. A un même jeu d'instructions peut correspondre un grand nombre d'implémentations différentes du microprocesseur.

4.3.2 Type d'instructions

Les instructions que l'on retrouve dans chaque microprocesseur peuvent être classées en 4 groupes :

- **Transfert de données** pour charger ou sauver en mémoire, effectuer des transferts de registre à registre, etc…
- **Opérations arithmétiques** : addition, soustraction, division, multiplication
- **Opérations logiques :** ET, OU, NON, NAND, comparaison, test, etc…
- **Contrôle de séquence :** branchement, test, etc…

4.3.3 Codage

Les instructions et leurs opérandes (paramètres) sont stockés en mémoire principale. La taille totale d'une instruction (nombre de bits nécessaires pour la représenter en mémoire) dépend du type d'instruction et aussi du type d'opérande. Chaque instruction est toujours codée sur un nombre entier d'octets afin de faciliter son décodage par le processeur. Une instruction est composée de deux champs :

- le code instruction, qui indique au processeur quelle instruction réaliser
- le champ opérande qui contient la donnée, ou la référence à une donnée en mémoire (son adresse).

Exemple :

Code instruction	Code opérande
1001 0011	0011 1110

Le nombre d'instructions du jeu d'instructions est directement lié au format du code instruction. Ainsi un octet permet de distinguer au maximum 256 instructions différentes.

4.3.4 Mode d'adressage

Un mode d'adressage définit la manière dont le microprocesseur va accéder à l'opérande. Les différents modes d'adressage dépendent des microprocesseurs mais on retrouve en général :

- l'adressage de registre où l'on traite la données contenue dans un registre
- l'adressage immédiat où l'on définit immédiatement la valeur de la donnée
- l'adressage direct où l'on traite une données en mémoire

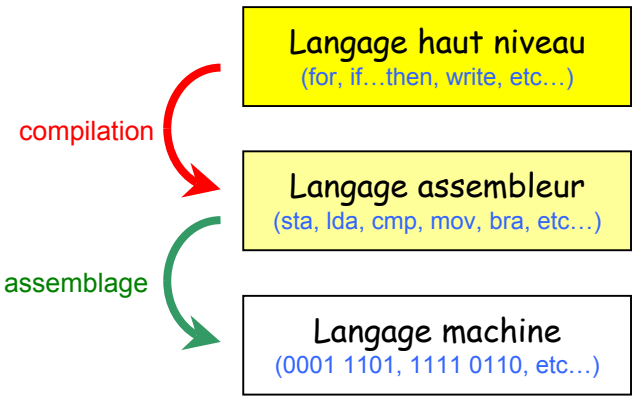
Selon le mode d'adressage de la donnée, une instruction sera codée par 1 ou plusieurs octets.

4.3.5 Temps d'exécution

Chaque instruction nécessite un certain nombre de cycles d'horloges pour s'effectuer. Le nombre de cycles dépend de la complexité de l'instruction et aussi du mode d'adressage. Il est plus long d'accéder à la mémoire principale qu'à un registre du processeur. La durée d'un cycle dépend de la fréquence d'horloge du séquenceur.

4.4 Langage de programmation

Le langage **machine** est le langage compris par le microprocesseur. Ce langage est difficile à maîtriser puisque chaque instruction est codée par une séquence propre de bits. Afin de faciliter la tâche du programmeur, on a créé différents langages plus ou moins évolués.



Le langage **assembleur** est le langage le plus « proche » du langage machine. Il est composé par des instructions en général assez rudimentaires que l'on appelle des **mnémoniques**. Ce sont essentiellement des opérations de transfert de données entre les registres et l'extérieur du microprocesseur (mémoire ou périphérique), ou des opérations arithmétiques ou logiques. Chaque instruction représente un code machine différent. Chaque microprocesseur peut posséder un assembleur différent.

La difficulté de mise en œuvre de ce type de langage, et leur forte dépendance avec la machine a nécessité la conception de langages de **haut niveau**, plus adaptés à l'homme, et aux applications qu'il cherchait à développer. Faisant abstraction de toute architecture de machine, ces langages permettent l'expression d'algorithmes sous une forme plus facile à apprendre, et à dominer (C, Pascal, Java, etc…). Chaque instruction en langage de haut niveau correspondra à une succession d'instructions en langage assembleur. Une fois développé, le programme en langage de haut niveau n'est donc pas compréhensible par le microprocesseur. Il faut le **compiler** pour le traduire en assembleur puis l'**assembler** pour le convertir en code machine compréhensible par le microprocesseur. Ces opérations sont réalisées à partir de logiciels spécialisés appelés *compilateur* et *assembleur*.

Exemple de programme :

Code machine (68HC11)	Assembleur (68HC11)	Langage C
@00 C6 64 @01 B6 00 @03 1B @04 5A @05 26 03	LDAB #100 LDAA #0 ret ABA DECB BNE ret	A=0 ; for (i=1 ; i<101 ; i++) A=A+i ;

4.5 Performances d'un microprocesseur

On peut caractériser la puissance d'un microprocesseur par le nombre d'instructions qu'il est capable de traiter par seconde. Pour cela, on définit :

- le **CPI** (Cycle Par Instruction) qui représente le nombre moyen de cycles d'horloge nécessaire pour l'exécution d'une instruction pour un microprocesseur donné.
- le **MIPS** (Millions d'Instructions Par Seconde) qui représente la puissance de traitement du microprocesseur.

$$MIPS = \frac{F_H}{CPI}$$ avec F_H en MHz

Pour augmenter les performances d'un microprocesseur, on peut donc soit augmenter la fréquence d'horloge (limitation matérielle), soit diminuer le CPI (choix d'un jeu d'instruction adapté).

4.6 Notion d'architecture RISC et CISC

Actuellement l'architecture des microprocesseurs se composent de deux grandes familles :

- L' architecture CISC (Complex Instruction Set Computer)
- L'architecture RISC (Reduced Instruction Set Computer)

4.6.1 L'architecture CISC

4.6.1.1 Pourquoi

Par le passé la conception de machines CISC était la seule envisageable. En effet, vue que la mémoire travaillait très lentement par rapport au processeur, on pensait qu'il était plus intéressant de soumettre au microprocesseur des instructions complexes. Ainsi, plutôt que de coder une opération complexe par plusieurs instructions plus petites (qui demanderaient autant d'accès mémoire très lent), il semblait préférable d'ajouter au jeu d'instructions du microprocesseur une instruction complexe qui se chargerait de réaliser cette opération. De plus, le développement des langages de haut niveau posa de nombreux problèmes quant à la conception de compilateurs. On a donc eu tendance à incorporer au niveau processeur des instructions plus proches de la structure de ces langages.

4.6.1.2 Comment

C'est donc une architecture avec un grand nombre d'instructions où le microprocesseur doit exécuter des tâches complexes par instruction unique. Pour une tâche donnée, une machine CISC exécute ainsi un petit nombre d'instructions mais chacune nécessite un plus grand nombre de cycles d'horloge. Le code machine de ces instructions varie d'une instruction à l'autre et nécessite donc un décodeur complexe (micro-code)

4.6.2 L'architecture RISC

4.6.2.1 Pourquoi

Des études statistiques menées au cours des années 70 ont clairement montré que les programmes générés par les compilateurs se contentaient le plus souvent d'affectations, d'additions et de multiplications par des constantes. Ainsi, 80% des traitements des langages de haut niveau faisaient appel à seulement 20% des instructions du microprocesseur. D'où l'idée de réduire le jeu d'instructions à celles le plus couramment utilisées et d'en améliorer la vitesse de traitement.

4.6.2.2 Comment

C'est donc une architecture dans laquelle les instructions sont en nombre réduit (chargement, branchement, appel sous-programme). Les architectures RISC peuvent donc être réalisées à partir de séquenceur câblé. Leur réalisation libère de la surface permettant d'augmenter le nombres de registres ou d'unités de traitement par exemple. Chacune de ces instructions s'exécutent ainsi en un cycle d'horloge. Bien souvent, ces instructions ne disposent que d'un seul mode d'adressage. Les accès à la mémoire s'effectue seulement à partir de deux instructions (Load et Store). Par contre, les instructions complexes doivent être réalisées à partir de séquences basées sur les instructions élémentaires, ce qui nécessite un compilateur très évolué dans le cas de programmation en langage de haut niveau.

4.6.3 Comparaison

Le choix dépendra des applications visées. En effet, si on diminue le nombre d'instructions, on crée des instructions complexes (CISC) qui nécessitent plus de cycles pour être décodées et si on diminue le nombre de cycles par instruction, on crée des instructions simples (RISC) mais on augmente alors le nombre d'instructions nécessaires pour réaliser le même traitement.

Architecture RISC	Architecture CISC
instructions simples ne prenant qu'un seul cycle 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	instructions complexes prenant plusieurs cycles 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

4.7 Améliorations de l'architecture de base

L'ensemble des améliorations des microprocesseurs visent à diminuer le temps d'exécution du programme.

La première idée qui vient à l'esprit est d'augmenter tout simplement la fréquence de l'horloge du microprocesseur. Mais l'accélération des fréquences provoque un surcroît de consommation ce qui entraîne une élévation de température. On est alors amené à équiper les processeurs de systèmes de refroidissement ou à diminuer la tension d'alimentation.

Une autre possibilité d'augmenter la puissance de traitement d'un microprocesseur est de diminuer le nombre moyen de cycles d'horloge nécessaire à l'exécution d'une instruction. Dans le cas d'une programmation en langage de haut niveau, cette amélioration peut se faire en optimisant le compilateur. Il faut qu'il soit capable de sélectionner les séquences d'instructions minimisant le nombre moyen de cycles par instructions. Une autre solution est d'utiliser une architecture de microprocesseur qui réduise le nombre de cycles par instruction.

4.7.1 Architecture pipeline

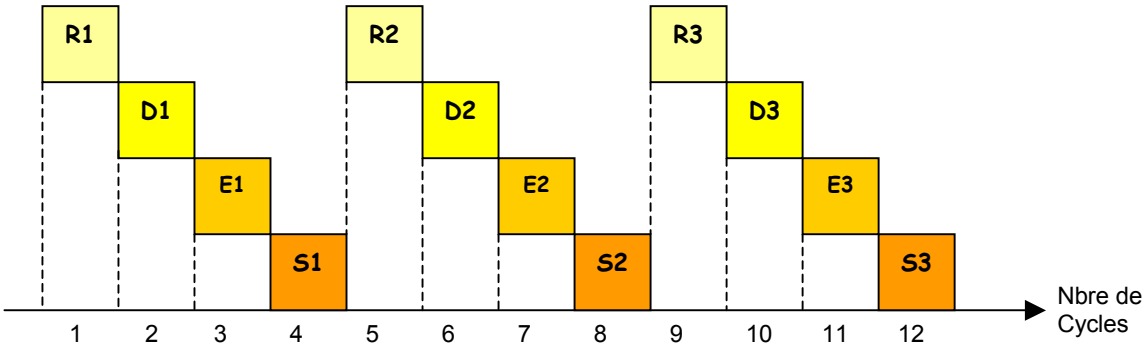
4.7.1.1 Principe

L'exécution d'une instruction est décomposée en une succession d'étapes et chaque étape correspond à l'utilisation d'une des fonctions du microprocesseur. Lorsqu'une instruction se trouve dans l'une des étapes, les composants associés aux autres étapes ne sont pas utilisés. Le fonctionnement d'un microprocesseur simple n'est donc pas efficace.

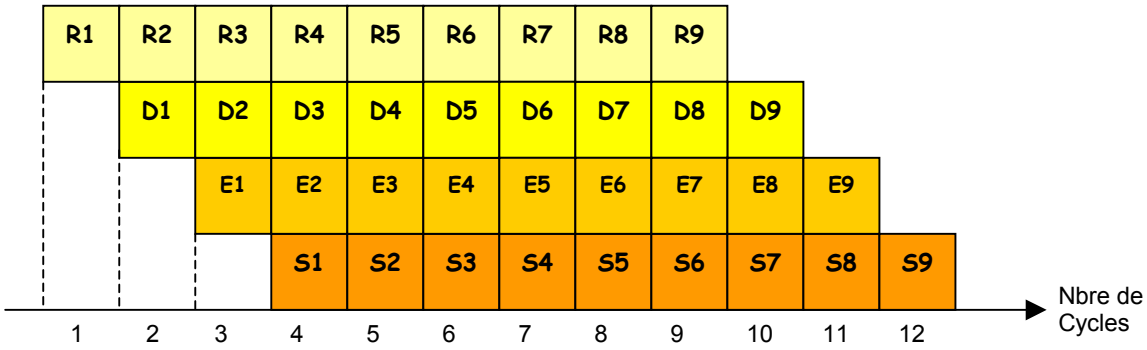
L'architecture pipeline permet d'améliorer l'efficacité du microprocesseur. En effet, lorsque la première étape de l'exécution d'une instruction est achevée, l'instruction entre dans la seconde étape de son exécution et la première phase de l'exécution de l'instruction suivante débute. Il peut donc y avoir une instruction en cours d'exécution dans chacune des étapes et chacun des composants du microprocesseur peut être utilisé à chaque cycle d'horloge. L'efficacité est maximale. Le temps d'exécution d'une instruction n'est pas réduit mais le débit d'exécution des instructions est considérablement augmenté. Une machine pipeline se caractérise par le nombre d'étapes utilisées pour l'exécution d'une instruction, on appelle aussi ce nombre d'étapes le nombre d'étages du pipeline.

Exemple de l'exécution en 4 phases d'une instruction :

Modèle classique :



Modèle pipeliné :



4.7.1.2 Gain de performance

Dans cette structure, la machine débute l'exécution d'une instruction à chaque cycle et le pipeline est pleinement occupé à partir du quatrième cycle. Le gain obtenu dépend donc du nombre d'étages du pipeline. En effet, pour exécuter **n** instructions, en supposant que chaque instruction s'exécute en **k** cycles d'horloge, il faut :

- **n.k** cycles d'horloge pour une exécution séquentielle.
- **k** cycles d'horloge pour exécuter la première instruction puis **n-1** cycles pour les **n-1** instructions suivantes si on utilise un pipeline de **k** étages

Le gain obtenu est donc de :

$$G=\frac{n \bullet k}{k+n-1}$$

Donc lorsque le nombre **n** d'instructions à exécuter est grand par rapport à **k**, on peut admettre qu'on divise le temps d'exécution par **k**.

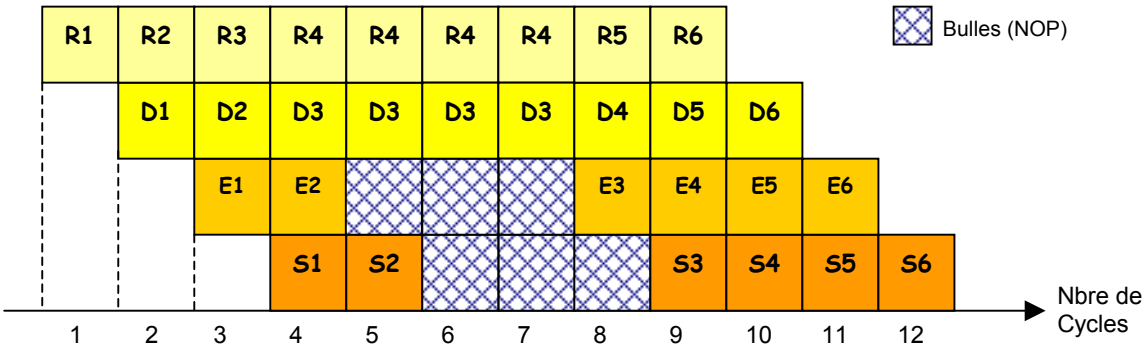
Remarques :
Le temps de traitement dans chaque unité doit être à peu près égal sinon les unités rapides doivent attendre les unités lentes.

Exemples :
L'Athlon d'AMD comprend un pipeline de 11 étages.
Les Pentium 2, 3 et 4 d'Intel comprennent respectivement un pipeline de 12, 10 et 20 étages.

4.7.1.3 Problèmes

La mise en place d'un pipeline pose plusieurs problèmes. En fait, plus le pipeline est long, plus le nombre de cas où il n'est pas possible d'atteindre la performance maximale est élevé. Il existe 3 principaux cas où la performance d'un processeur pipeliné peut être dégradé ; ces cas de dégradations de performances sont appelés des **aléas** :

- **aléa structurel** qui correspond au cas où deux instructions ont besoin d'utiliser la même ressource du processeur (conflit de dépendance),
- **aléa de données** qui intervient lorsqu'une instruction produit un résultat et que l'instruction suivante utilise ce résultat avant qu'il n'ait pu être écrit dans un registre,
- **aléa de contrôle** qui se produit chaque fois qu'une instruction de branchement est exécutée. Lorsqu'une instruction de branchement est chargée, il faut normalement attendre de connaître l'adresse de destination du branchement pour pouvoir charger l'instruction suivante. Les instructions qui suivent le saut et qui sont en train d'être traitées dans les étages inférieurs le sont en général pour rien, il faudra alors vider le pipeline. Pour atténuer l'effet des branchements, on peut spécifier après le branchement des instructions qui seront toujours exécutées. On fait aussi appel à la **prédiction de branchement** qui a pour but de recenser lors de branchements le comportement le plus probable. Les mécanismes de prédiction de branchement permettent d'atteindre une fiabilité de prédiction de l'ordre de 90 à 95 %.



Lorsqu'un aléa se produit, cela signifie qu'une instruction ne peut continuer à progresse dans le pipeline. Pendant un ou plusieurs cycles, l'instruction va rester bloquée dans un étage du pipeline, mais les instructions situées plus en avant pourront continuer à s'exécuter jusqu'à ce que l'aléa ait disparu. Plus le pipeline possède d'étages, plus la pénalité est grande. Les compilateurs s'efforcent d'engendrer des séquences d'instructions permettant de maximiser le remplissage du pipeline. Les étages vacants du pipeline sont appelés des « bulles » de pipeline, en pratique une bulle correspond en fait à une instruction NOP (No OPeration) émise à la place de l'instruction bloquée.

4.7.2 Notion de cache mémoire

4.7.2.1 Problème posé

L'écart de performance entre le microprocesseur et la mémoire ne cesse de s'accroître. En effet, les composants mémoire bénéficient des mêmes progrès technologique que les microprocesseurs mais le décodage des adresses et la lecture/écriture d'une données sont des étapes difficiles à accélérer. Ainsi, le temps de cycle processeur décroît plus vite que le temps d'accès mémoire entraînant un goulot d'étranglement. La mémoire n'est plus en mesure de délivrer des informations aussi rapidement que le processeur est capable de les traiter. Il existe donc une latence d'accès entre ces deux organes.

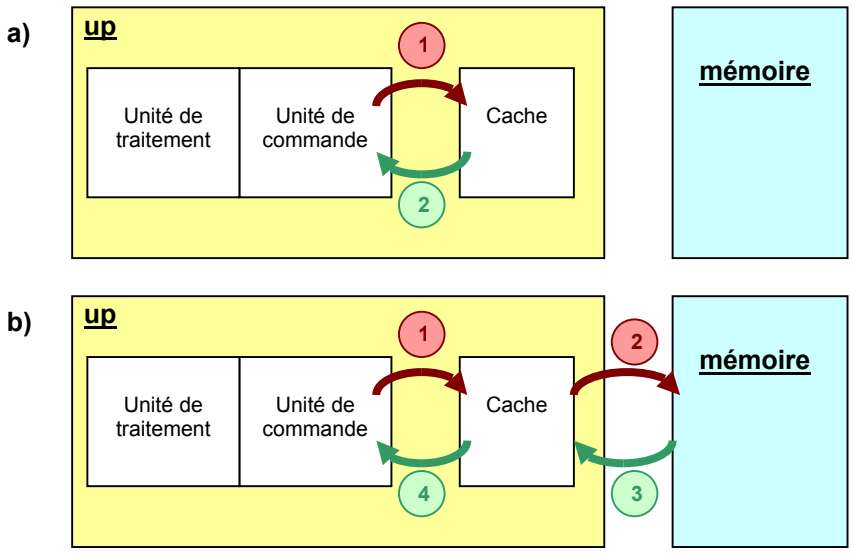
4.7.2.2 Principe

Depuis le début des années 80, une des solutions utilisées pour masquer cette latence est de disposer une mémoire très rapide entre le microprocesseur et la mémoire. Elle est appelée **cache mémoire**. On compense ainsi la faible vitesse relative de la mémoire en permettant au microprocesseur d'acquérir les données à sa vitesse propre. On la réalise à partir de cellule SRAM de taille réduite (à cause du coût). Sa capacité mémoire est donc très inférieure à celle de la mémoire principale et sa fonction est de stocker les informations les plus récentes ou les plus souvent utilisées par le microprocesseur. Au départ cette mémoire était intégrée en dehors du microprocesseur mais elle fait maintenant partie intégrante du microprocesseur et se décline même sur plusieurs niveaux.

Le principe de cache est très simple : le microprocesseur n'a pas conscience de sa présence et lui envoie toutes ses requêtes comme s'il agissait de la mémoire principale :

- Soit la donnée ou l'instruction requise est présente dans le cache et elle est alors envoyée directement au microprocesseur. On parle de **succès** de cache. (a)

- soit la donnée ou l'instruction n'est pas dans le cache, et le contrôleur de cache envoie alors une requête à la mémoire principale. Une fois l'information récupérée, il la renvoie au microprocesseur tout en la stockant dans le cache. On parle de **défaut** de cache. (b)



Bien entendu, le cache mémoire n'apporte un gain de performance que dans le premier cas. Sa performance est donc entièrement liée à son taux de succès. Il est courant de rencontrer des taux de succès moyen de l'ordre de 80 à 90%.

Remarques :

❶ Un cache utilisera une carte pour savoir quels sont les mots de la mémoire principale dont il possède une copie. Cette carte devra avoir une structure simple.

❷ Il existe dans le système deux copies de la même information : l'originale dans la mémoire principale et la copie dans le cache. Si le microprocesseur modifie la donnée présente dans le cache, il faudra prévoir une mise à jour de la mémoire principale.

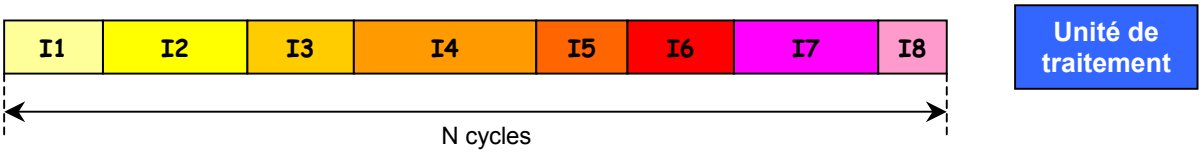
❸ Lorsque le cache doit stocker une donnée, il est amené à en effacer une autre. Il existe donc un contrôleur permettant de savoir quand les données ont été utilisées pour la dernière fois. La plus ancienne non utilisée est alors remplacée par la nouvelle.

❹ A noter que l'on peut reprendre le même principe pour les disques durs et CD/DVD.

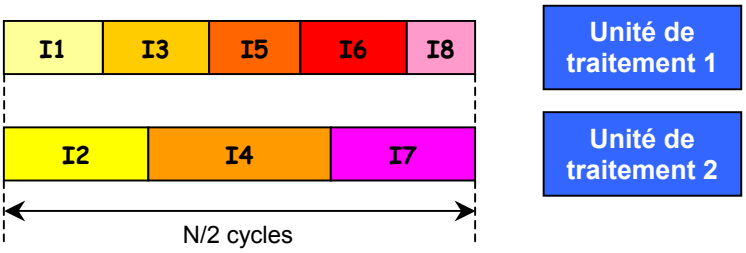
4.7.3 Architecture superscalaire

Une autre façon de gagner en performance est d'exécuter plusieurs instructions en même temps. L'approche superscalaire consiste à doter le microprocesseur de plusieurs unités de traitement travaillant en parallèle. Les instructions sont alors réparties entre les différentes unités d'exécution. Il faut donc pouvoir soutenir un flot important d'instructions et pour cela disposer d'un cache performant.

Architecture scalaire :



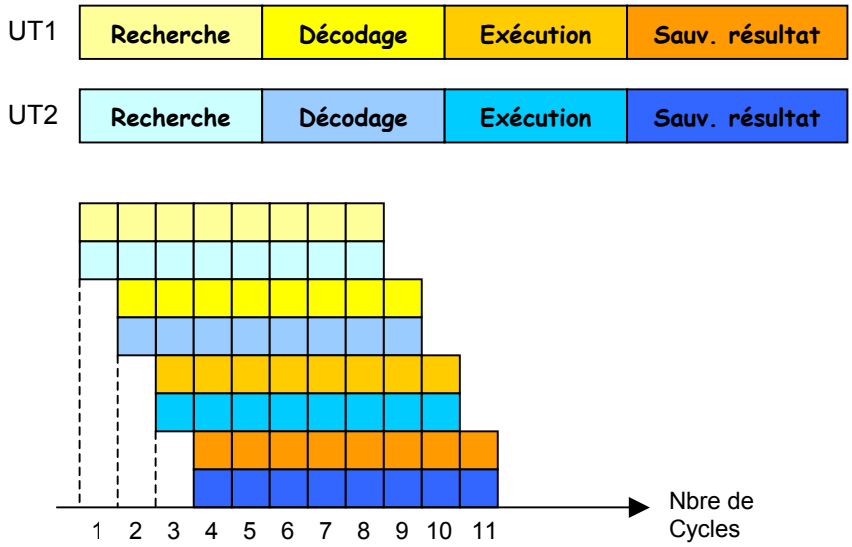
Architecture superscalaire :



Remarque :
C'est le type d'architecture mise en oeuvre dans les premiers Pentium d'Intel apparus en 1993.

4.7.4 Architecture pipeline et superscalaire

Le principe est de d'exécuter les instructions de façon pipelinée dans chacune des unités de traitement travaillant en parallèle.



4.8 Processeurs spéciaux

4.8.1 Le microcontrôleur

Ce sont des systèmes minimum sur une seule puce. Ils contiennent un CPU, de la RAM, de la ROM et des ports d'Entrée/Sorties (parallèles, séries, I2C, etc..). Ils comportent aussi des fonctions spécifiques comme des compteurs programmables pour effectuer des mesures de durées, des CAN voir des CNA pour s'insérer au sein de chaînes d'acquisition, des interfaces pour réseaux de terrain, etc ...

Il est adapté pour répondre au mieux aux besoin des applications embarquées (appareil électroménagers, chaîne d'acquisition, lecteur carte à puce, etc...). Il est par contre généralement moins puissant en terme de rapidité, de taille de données traitables ou de taille de mémoire adressable qu'un microprocesseur.

4.8.2 Le processeur de signal

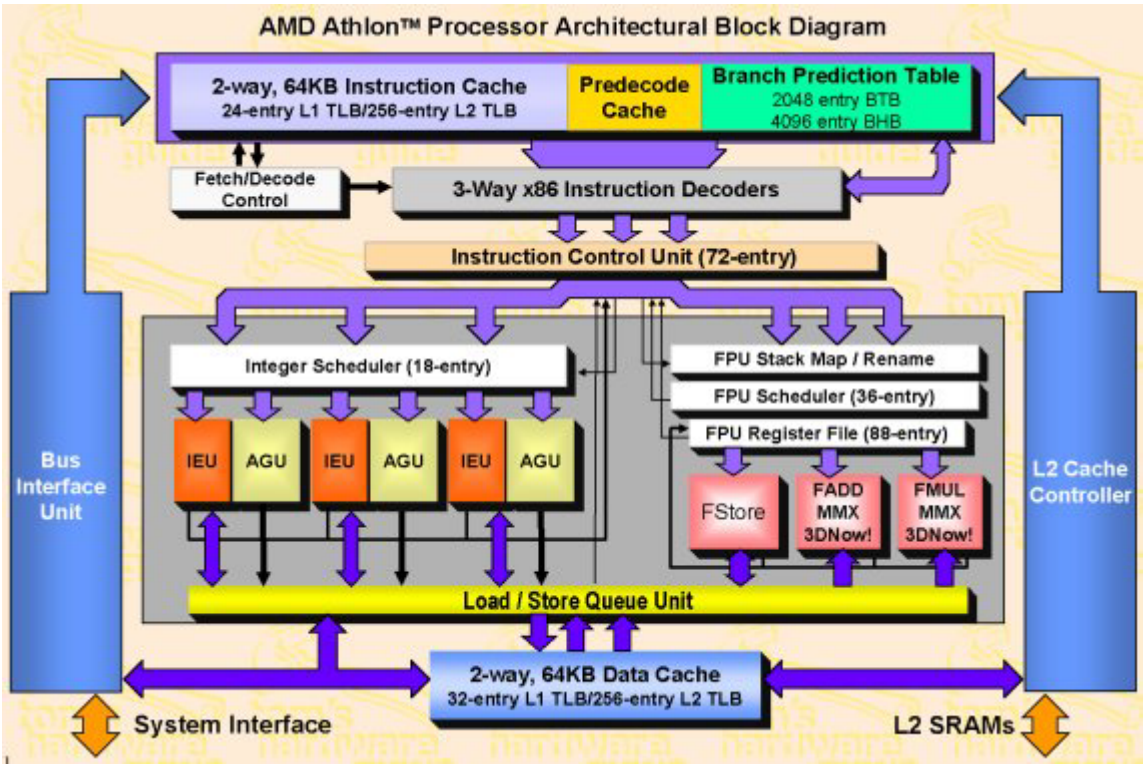
Le processeur de signal est beaucoup plus spécialisé. Alors qu'un microprocesseur n'est pas conçu pour une application spécifique, le processeur DSP (Digital Signal Processor) est optimisé pour effectuer du traitement numérique du signal (calcul de FFT, convolution, filtrage numérique, etc...).

Les domaines d'application des D.S.P étaient à l'origine les télécommunications et le secteur militaire. Aujourd'hui, les applications se sont diversifiées vers le multimédia (lecteur CD, MP3, etc..) l'électronique grand public (télévision numérique, téléphone portable, etc…), l'automatique, l'instrumentation, l'électronique automobile, etc…

4.9 Exemples

Voici deux exemples d'architecture de deux processeurs qui tenaient le haut du pavé lors de leur sortie en 1999 : l'Athlon d'AMD et le Pentium III d'Intel. (f ≈500MHz)

4.9.1 AMD Athlon :

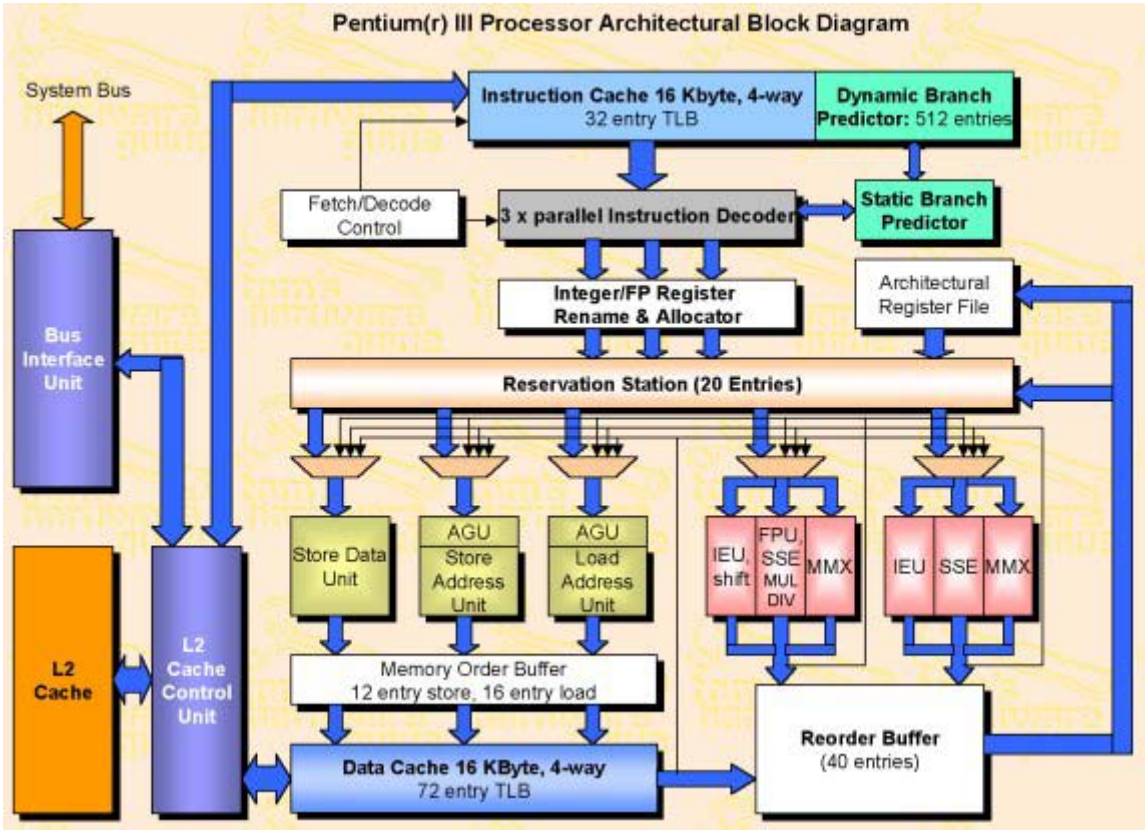


AGU : Adress Generation Unit
IEU : Integer Execution Unit
BTB : Branch Target Buffer
BHB : Branch History Buffer

Caractéristiques:

- 9 unité de traitement se composant de :
  - 1 ALU (traitement entier) comprenant 6 unités de traitement :
    - 3 unités pour le traitement des données (IEU)
    - 3 unités pour l'adressage des données (AGU)
  - 1 FPU (traitement réel) comprenant 3 unités :
    - 1 FPU store
    - 1 Fadd / MMX / 3Dnow !
    - 1 Fmul /MMX / 3Dnow !
- Pipeline entier : 10 étages, pipeline flottant : 15 étages.
- Prédiction dynamique et exécution du traitement "dans le désordre" (out-of-order)
- 6 unités de décodage parallèles (3 micro-programmées, 3 câblées) mais seules 3 peuvent fonctionner en même temps.
- Cache mémoire de niveau 1 (L1) : 128 Ko
  - 64 Ko pour les données
  - 64 Ko pour les instructions
- Contrôleur de cache L2 supportant de 512Ko à 8Mo avec vitesse programmable (1/2 ou 1/3 de la vitesse CPU)
- 22 millions de transistors

4.9.2 Intel Pentium III



Caractéristiques :

- Plusieurs unités de traitement mais au 5 instructions exécutées en même temps sur 5 ports :
    - Port 0 : ALU, FPU, AGU MMX et SSE
    - Port 1 : ALU, SSE, MMX
    - Port 2 : AGU (load)
    - Port 3 : AGU (store)
    - Port 4 : Store Data Unit
- Pipeline entier : 12 à 17 étages, pipeline flottant : environ 25 étages
- Prédiction dynamique et exécution du traitement "dans le désordre" (out-of-order)
- 3 unités de décodage parallèles : 1 micro-programmée, 2 câblées.
- 5 pipelines de 10 étages
- Cache mémoire de niveau 1 (L1) : 32 Ko
    - 16 Ko pour les données
    - 16 Ko pour les instructions
- Contrôleur de cache L2 supportant jusqu'à 512 Ko à ½ de la vitesse CPU
- 9.5 millions de transistors

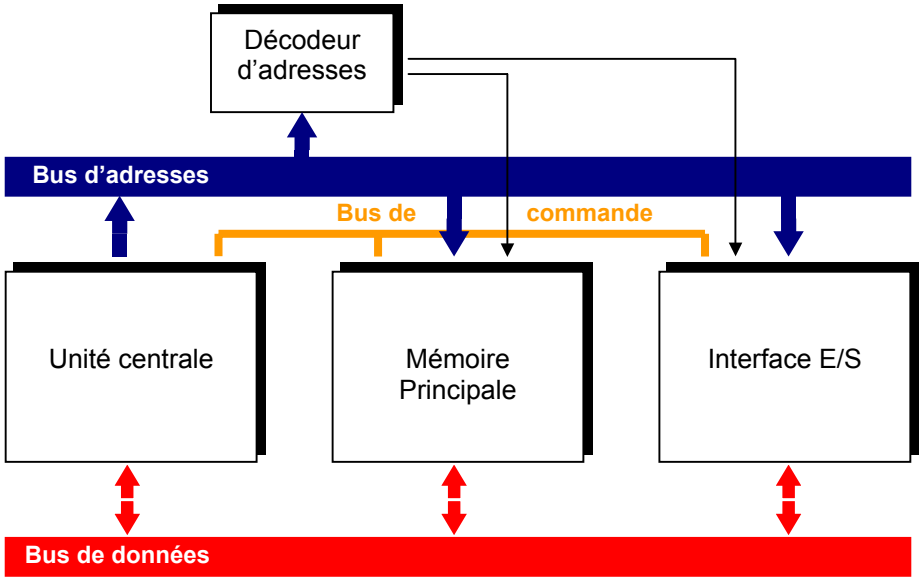
.

5 Les échanges de données

Chapitre 5

La fonction d'un système à microprocesseurs, quel qu'il soit, est le traitement de l'information. Il est donc évident qu'il doit acquérir l'information fournie par son environnement et restituer les résultats de ses traitements. Chaque système est donc équipé d'une ou plusieurs interfaces d'entrées/sorties permettant d'assurer la communication entre le microprocesseur et le monde extérieur.

Les techniques d'entrées/sorties sont très importantes pour les performances du système. Rien ne sert d'avoir un microprocesseur calculant très rapidement s'il doit souvent perdre son temps pour lire des données ou écrire ses résultats. Durant une opération d'entrée/sortie, l'information est échangée entre la mémoire principale et un périphérique relié au système. Cet échange nécessite une *interface* (ou *contrôleur*) pour gérer la connexion. Plusieurs techniques sont employées pour effectuer ces échanges.



5.1 L'interface d'entrée/sortie

5.1.1 Rôle

Chaque périphérique sera relié au système par l'intermédiaire d'une interface (ou contrôleur) dont le rôle est de :

- **Connecter** le périphérique au bus de données
- **Gérer** les échanges entre le microprocesseur et le périphérique

5.1.2 Constitution

Pour cela, l'interface est constituée par :

- Un **registre de commande** dans lequel le processeur décrit le travail à effectuer (sens de transfert, mode de transfert).
- Un ou plusieurs **registres de données** qui contiennent les mots à échanger entre le périphérique et la mémoire
- Un **registre d'état** qui indique si l'unité d'échange est prête, si l'échange s'est bien déroulé, etc…

On accède aux données de l'interface par le biais d'un espace d'adresses d'entrées/sorties.

5.2 Techniques d'échange de données

Avant d'envoyer ou de recevoir des informations, le microprocesseur doit connaître l'état du périphérique. En effet, le microprocesseur doit savoir si un périphérique est prêt à recevoir ou à transmettre une information pour que la transmission se fasse correctement. Il existe 2 modes d'échange d'information :

- Le mode programmé par **scrutation** ou **interruption** où le microprocesseur sert d'intermédiaire entre la mémoire et le périphérique
- Le mode en accès direct à la mémoire (**DMA**) où le microprocesseur ne se charge pas de l'échange de données.

5.2.1 Echange programmé

5.2.1.1 Scrutation

Dans la version la plus rudimentaire, le microprocesseur interroge l'interface pour savoir si des transferts sont prêts. Tant que des transferts ne sont pas prêts, le microprocesseur attend.

L'inconvénient majeur est que le microprocesseur se retrouve souvent en phase d'attente. Il est complètement occupé par l'interface d'entrée/sortie. De plus, l'initiative de l'échange de données est dépendante du programme exécuté par le microprocesseur. Il peut donc arriver que des requêtes d'échange ne soient pas traitées immédiatement car le microprocesseur ne se trouve pas encore dans la boucle de scrutation.

Ce type d'échange est très lent.

5.2.1.2 Interruption

Une interruption est un signal, généralement asynchrone au programme en cours, pouvant être émis par tout dispositif externe au microprocesseur. Le microprocesseur possède une ou plusieurs entrées réservées à cet effet. Sous réserve de certaines conditions, elle peut interrompre le travail courant du microprocesseur pour forcer l'exécution d'un programme traitant la cause de l'interruption.

Dans un échange de données par interruption, le microprocesseur exécute donc son programme principal jusqu'à ce qu'il reçoive un signal sur sa ligne de requête d'interruption. Il se charge alors d'effectuer le transfert de données entre l'interface et la mémoire.

Principe de fonctionnement d'une interruption :

Avant chaque exécution d'instructions, le microprocesseur examine si il y a eu une **requête** sur sa ligne d'interruption. Si c'est le cas, il interrompt toutes ces activités et sauvegarde l'état présent (registres, PC, accumulateurs, registre d'état) dans un registre particulier appelé **pile**. Les données y sont ''entassées'' comme on empile des livres (la première donnée sauvegardée sera donc la dernière à être restituée). Ensuite, il exécute le programme d'interruption puis restitue l'état sauvegardé avant de reprendre le programme principale.

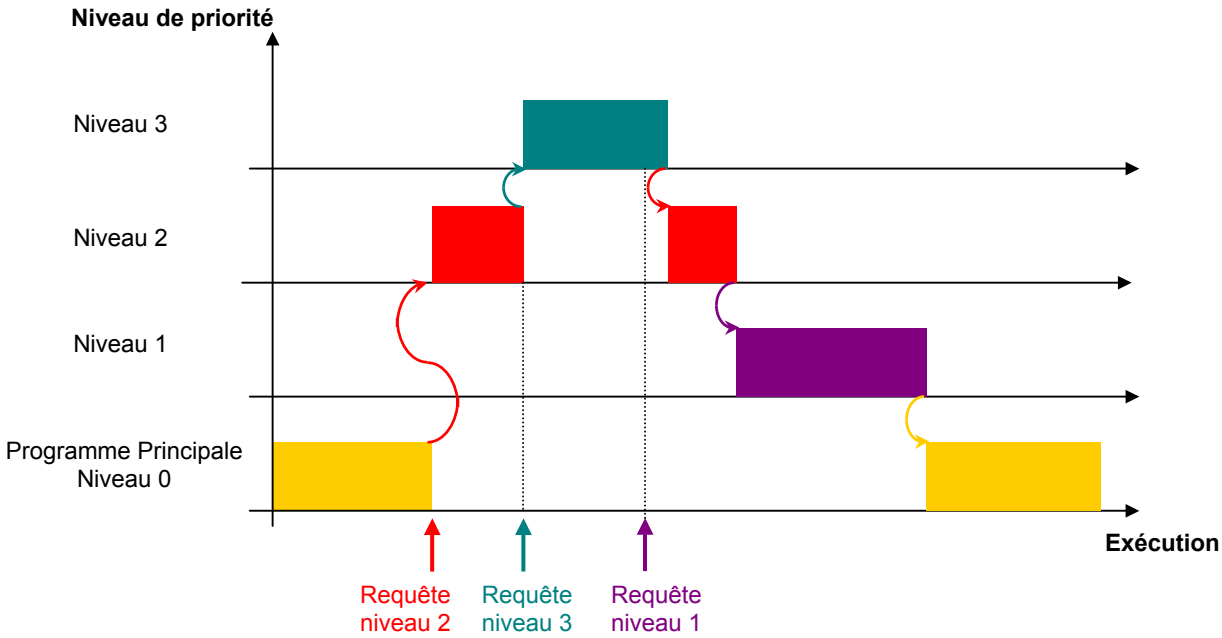
Remarques :

❶ Certaine source d'interruption possède leur propre autorisation de fonctionnement sous la forme d'un bit à positionner, on l'appelle le masque d'interruption.

❷ On peut donc interdire ou autoriser certaines sources d'interruptions, on les appelle les interruptions masquables.

❸ Chaque source d'interruption possède un vecteur d'interruption où est sauvegardé l'adresse de départ du programme à exécuter.

❹ Les interruptions sont classées par ordre de priorité. Dans le cas où plusieurs interruptions se présentent en même temps, le microprocesseur traite d'abord celle avec la priorité la plus élevée.



5.2.2 Echange direct avec la mémoire

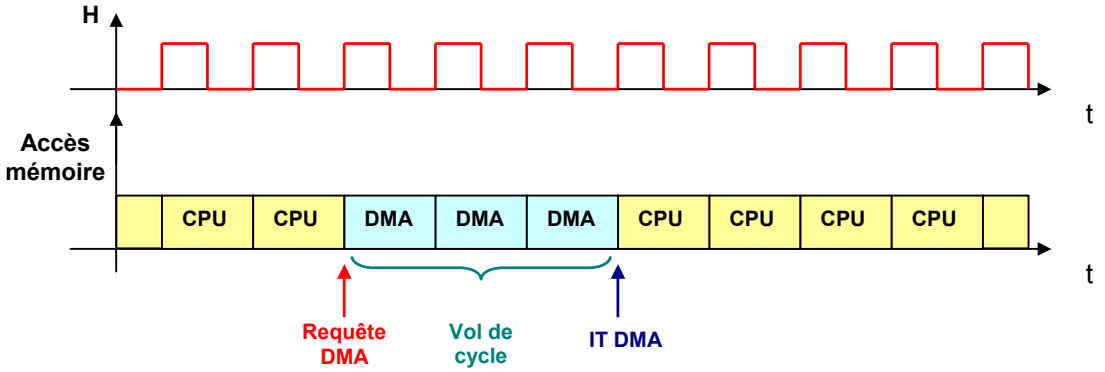
Ce mode permet le transfert de blocs de données entre la mémoire et un périphérique sans passer par le microprocesseur. Pour cela, un circuit appelé **contrôleur de DMA** (Direct Memory Access) prend en charge les différentes opérations.

Le DMA se charge entièrement du transfert d'un bloc de données. Le microprocesseur doit tout de même :

- initialiser l'échange en donnant au DMA l'identification du périphérique concerné
- donner le sens du transfert
- fournir l'adresse du premier et du dernier mot concernés par le transfert

Un contrôleur de DMA est doté d'un registre d'adresse, d'un registre de donnée, d'un compteur et d'un dispositif de commande (logique câblée). Pour chaque mot échangée, le DMA demande au microprocesseur le contrôle du bus, effectue la lecture ou l'écriture mémoire à l'adresse contenue dans son registre et libère le bus. Il incrémente ensuite cette adresse et décrémente son compteur. Lorsque le compteur atteint zéro, le dispositif informe le processeur de la fin du transfert par une ligne d'interruption.

Le principal avantage est que pendant toute la durée du transfert, le processeur est libre d'effectuer un traitement quelconque. La seule contrainte est une limitation de ses propres accès mémoire pendant toute la durée de l'opération, puisqu'il doit parfois retarder certains de ses accès pour permettre au dispositif d'accès direct à la mémoire d'effectuer les siens : il y a apparition de vols de cycle.



5.3 Types de liaisons

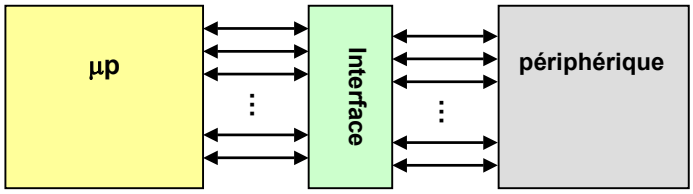
Les systèmes à microprocesseur utilisent deux types de liaison différentes pour se connecter à des périphériques :

- liaison parallèle
- liaison série

On caractérise un type de liaison par sa **vitesse de transmission** ou **débit** (en bit/s).

5.3.1 Liaison parallèle

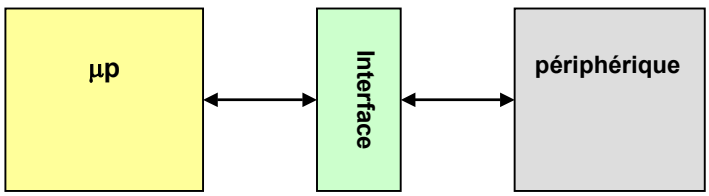
Dans ce type de liaison, tous les bits d'un mot sont transmis simultanément. Ce type de transmission permet des transferts rapides mais reste limitée à de faibles distances de transmission à cause du nombre important de lignes nécessaires (coût et encombrement) et des problèmes d'interférence électromagnétique entre chaque ligne (fiabilité). La transmission est cadencée par une horloge



Exemple :
Bus PCI, AGP dans un PC.

5.3.2 Liaison série

Dans ce type de liaison, les bits constitutifs d'un mot sont transmis les uns après les autres sur un seul fil. Les distances de transmission peuvent donc être plus beaucoup plus importantes mais la vitesse de transmission est plus faible. Sur des distance supérieures à quelques dizaines de mètres, on utilisera des modems aux extrémités de la liaison.



La transmission de données en série peut se concevoir de deux façons différentes :

- en mode synchrone, l'émetteur et le récepteur possède une horloge synchronisée qui cadence la transmission. Le flot de données peut être ininterrompu.

- en mode asynchrone, la transmission s'effectue au rythme de la présence des données. Les caractères envoyés sont encadrés par un signal *start* et un signal *stop*.

Principe de base d'une liaison série asynchrone :

Afin que les éléments communicants puissent se comprendre, il est nécessaire d'établir un protocole de transmission. Ce protocole devra être le même pour chaque élément.

Paramètres rentrant en jeu :

- **longueur des mots transmis** : 7 bits (code ASCII) ou 8 bits

- **vitesse de transmission** : les vitesses varient de 110 bit/s à 128000 bit/s et détermine les fréquences d'horloge de l'émetteur et du récepteur.

- **parité** : le mot transmis peut être suivis ou non d'un bit de parité qui sert à détecter les erreurs éventuelles de transmission. Il existe deux types de parité : paire ou impaire. Si on fixe une parité paire, le nombre total de bits à 1 transmis (bit de parité inclus) doit être paire. C'est l'inverse pour une parité impaire.

- **bit de start** : la ligne au repos est à l'état 1 (permet de tester une coupure de la ligne). Le passage à l'état bas de la ligne va indiquer qu'un transfert va commencer. Cela permet de synchroniser l'horloge de réception.

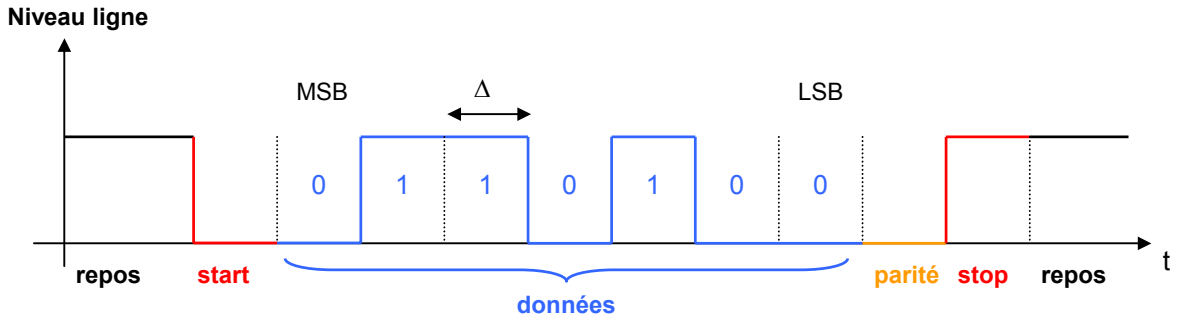
- **bit de stop** : après la transmission, la ligne est positionnée à un niveau 1 pendant un certains nombre de bit afin de spécifier la fin du transfert. En principe, on transmet un, un et demi ou 2 bits de stop.

Déroulement d'une transmission :

Les paramètres du protocole de transmission doivent toujours être fixés avant la transmission. En l'absence de transmission, la liaison est au repos au niveau haut pour détecter une éventuelle coupure sur le support de transmission. Une transmission s'effectue de la manière suivante :

❶ L'émetteur positionne la ligne à l'état bas : c'est le bit de **start**.
❷ Les bits sont transmis les un après les autres, en commençant par le bit de poids fort.
❸ Le bit de parité est éventuellement transmis.
❹ L'émetteur positionne la ligne à l'état haut : c'est le bit de **stop**.

Exemple : transmission d'un mot de 7 bits $(0110100)_2$ – Parité impaire – 1 bit de Stop



Horloge = $F=\frac{1}{\Delta}$ Hz

Vitesse de transmission = $v=\frac{1}{\Delta}$ bits/s

Contrôle de flux :

Le contrôle de flux permet d'envoyer des informations seulement si le récepteur est prêt (modem ayant pris la ligne, tampon d'une imprimante vide, etc…). Il peut être réalisé de manière logiciel ou matériel.

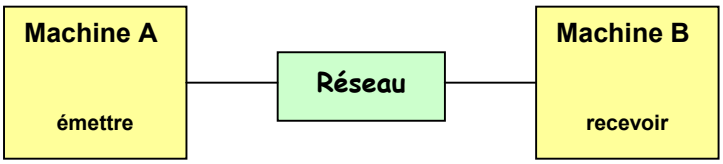
Pour contrôler le flux de données matériellement, il faudra utiliser des lignes de contrôle supplémentaire permettant à l'émetteur et au récepteur de s'informer mutuellement de leur état respectif (prêt ou non).

Dans un contrôle de type logiciel, l'émetteur envoie des données et lorsque le récepteur ne peut plus les recevoir (registre plein), il envoie une information à l'émetteur pour le prévenir, via la liaison série. L'émetteur doit donc toujours être à l'écoute du récepteur avant d'envoyer une donnée sur la ligne.

5.4 Notion de réseau

5.4.1 Introduction

Pour des raisons d'efficacité, on essaie de plus en plus de connecter des systèmes indépendants entre eux par l'intermédiaire d'un réseau. On permet ainsi aux utilisateurs ou aux applications de partager et d'échanger les mêmes informations.



Pour faire circuler une information sur un réseau, on peut utiliser principalement deux stratégies. Soit l'information est envoyée de façon complète, soit elle décomposées en petits morceaux (paquets). Les paquets sont alors envoyés séparément sur le réseau puis réassemblés par la machine destinataire. On parle souvent de réseaux à *commutations de paquets*… La première stratégie n'est que très rarement utilisés en informatique car les risques d'erreurs sont trop importants.

Les règles et les moyens mis en œuvre dans l'interconnexion et le dialogue des machines définissent le protocole et l'architecture du réseau. Toutes ces règles sont définies par des normes pour que des machines d'architecture différente puissent communiquer entre elles. Par exemple, pour l'envoie d'un fichier entre deux ordinateurs reliés par un réseau, il faut résoudre plusieurs problèmes :
Avant l'envoi des premiers octets du fichier, la machine source doit :

1) accéder au réseau,
2) s'assurer qu'elle peut atteindre la machine destination en donnant une adresse,
3) s'assurer que la machine destination est prête à recevoir des données,
4) contacter la bonne application sur la machine destination,
5) s'assurer que l'application est prête à accepter le fichier et à le stocker dans son système de fichier

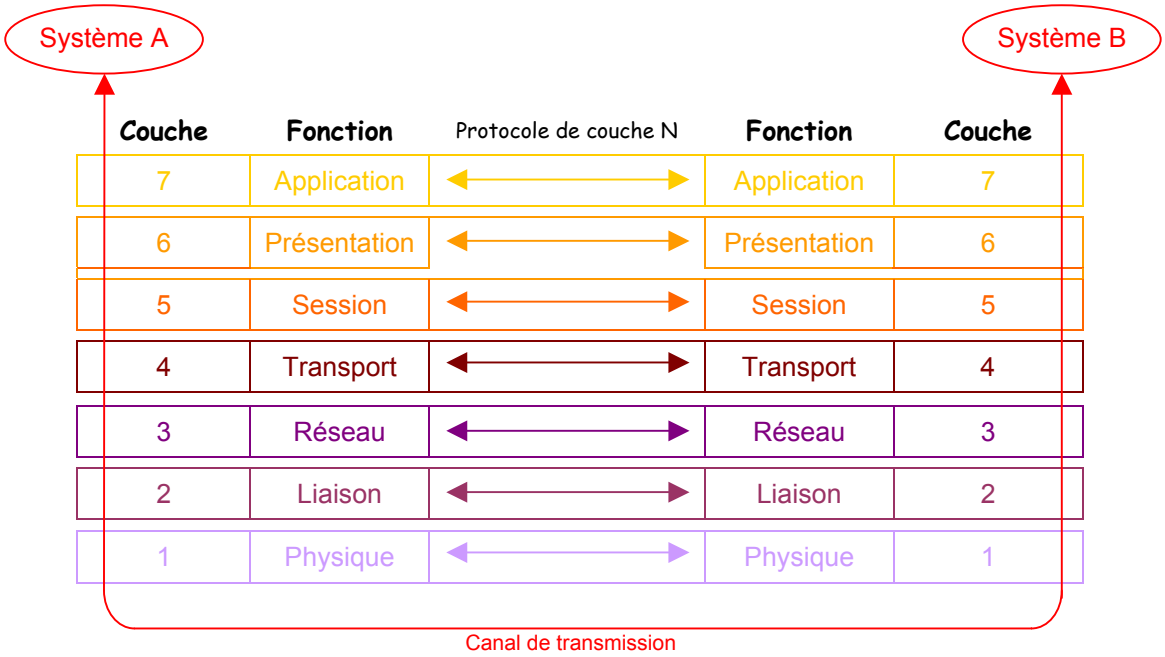
Pendant l'envoi, les 2 machines doivent :

6) envoyer les données dans un format compris par les 2 machines,
7) gérer l'envoi des commandes et des données et le rangement des données sur disque,
8) s'assurer que les commandes et les données sont échangées correctement et que l 'application destinataire reçoit toutes les données sans erreur et dans le bon ordre.

Au début des années 70, chaque constructeur a développé sa propre solution réseau autour d'architecture et de protocoles privés. Mais ils se sont vite rendu compte qu'il serait impossible d'interconnecter ces différents réseaux… Ils ont donc décidé de définir une norme commune. Ce modèle s'appelle **OSI** (Open System Interconnection) et comporte 7 couches qui ont toutes une fonctionnalité particulière. Il a été proposé par l'ISO, et il est aujourd'hui universellement adopté et utilisé.

5.4.2 Le modèle OSI

Le modèle OSI définit un modèle d'architecture décomposé en couche dont la numérotation commence par le bas. Il donne une description globale de la fonction de chaque couche. Chacune des couches de ce modèle représente une catégorie de problème que l'on peut rencontrer dans la conception d'un réseau. Ainsi, la mise en place d'un réseau revient à trouver une solution technologique pour chacune des couches composants le réseau. L'utilisation de couches permet également de changer une solution technique pour une seule couche sans pour autant être obligé de repenser toute l'architecture du réseau. De plus, chaque couche garantit à la couche supérieure qu'elle a réalisé son travail sans erreur.



Chaque couche est identifiée par son niveau N et réalise un sous-ensemble de fonctions nécessaire à la communication avec un autre système. Pour réaliser ces fonctions de communication, la couche N s 'appuie uniquement sur la couche immédiatement inférieure par l 'intermédiaire d'une interface. Le dialogue entre les deux systèmes s'établie forcément entre deux couches de niveau N identique mais l'échange ''physique'' de données s'effectue uniquement entre les couches de niveau 1. Les règles et conventions utilisées pour ce dialogue sont appelées **protocole de couche N**.

On appelle les couches 1, 2, 3 et 4 les couches ''*basses*'' et les couches 5, 6 et 7 les couches ''*hautes*''. Les couches ''basses'' sont concernées par la réalisation d'une communication fiable de bout en bout alors que les couches ''hautes'' offrent des services orientées vers les utilisateurs.

- **Couche 1 : Physique**

La couche physique se préoccupe de résoudre les problèmes matériels. Elle normalise les moyens mécaniques (nature et caractéristique du support : câble, voie hertzienne, fibre optique, etc…), électrique (transmission en bande de base, modulation, puissance, etc…) et fonctionnels (transmission synchrone/asynchrone, simplex, half/full duplex, etc..) nécessaires à l'activation, au maintient et la désactivation des connexions physiques destinées à la transmission de bits entre deux entités de liaison de données.

- **Couche 2 : Liaison**

La couche liaison de données détecte et corrige si possible les erreurs dues au support physique et signal à la couche réseau les erreurs irrécupérable. Elle supervise le fonctionnement de la transmission et définit la structure syntaxique des messages, la manière d'enchaîner les échanges selon un protocole normalisée ou non.

Cette couche reçoit les données brutes de la couche physique, les organise en trames, gère les erreurs, retransmet les trames erronées, gère les acquittements qui indiquent si les données ont bien été transmises puis transmet les données formatées à la couche réseau supérieure.

- **Couche 3 : Réseau**

La couche réseau est chargée de l'acheminement des informations vers le destinataire. Elle gère l'adressage, le routage, le contrôle de flux et la correction d'erreurs non réglées par la couche 2. A ce niveau là, il s'agit de faire transiter une information complète (ex : un fichier) d'une machine à une autre à travers un réseau de plusieurs ordinateurs. Elle permet donc de transmettre les trames reçues de la couche 2 en trouvant un chemin vers le destinataire.

- **La couche 4 : Transport**

Elle remplit le rôle de charnière entre les couches basses du modèle OSI et le monde des traitements supportés par les couches 5,6 et 7. Elle assure un transport de **bout en bout** entre les deux systèmes en assurant la segmentation des messages en paquets et en délivrant les informations dans l'ordre sans perte ni duplication. Elle doit acheminer les données du système source au système destination quelle que soit la topologie du réseau de communication entre les deux systèmes. Elle permet ainsi aux deux systèmes de dialoguer directement comme si le réseau n'existait pas. Elle remplit éventuellement le rôle de correction d'erreurs. Les critères de réalisation de la couche transport peuvent être le délai d'établissement de la connexion, sa probabilité d'échec, le débit souhaité, le temps de traversé, etc…

- **La couche 5 : Session**

Elle gère le dialogue entre 2 applications distantes (dialogue unidirectionnel/bidirectionnel, gestion du tour de parole, synchronisation, etc...).

- **La couche 6 : Présentation**

Cette couche s'occupe de la partie syntaxique et sémantique de la transmission de l'information afin d'affranchir la couche supérieure des contraintes syntaxiques. Elle effectue ainsi le codage des caractères pour permettre à deux systèmes hétérogènes de communiquer. C'est à ce niveau que peuvent être implantées des techniques de compression et de chiffrement de données.

- **La couche 7 : Application**

Elle gère les programmes utilisateurs et définit des standards pour les différents logiciels commercialisés adoptent les mêmes principes (fichier virtuel, messagerie, base de données, etc…).

5.4.3 Classification des réseaux

On peut établir une classification des réseaux à l'aide de leur taille. Les réseaux sont divisés en quatre grandes familles : les PAN, LAN, MAN et WAN.

- **PAN : Personal Area Network ou réseau personnel**

Ce type de réseau interconnecte des équipements personnels comme un ordinateur portable, un ordinateur fixe, une imprimante, etc… Il s'étend sur quelques dizaine de mètres. Les débits sont importants (qq Mb/s).

- **LAN : Local Area Network ou réseau local.**

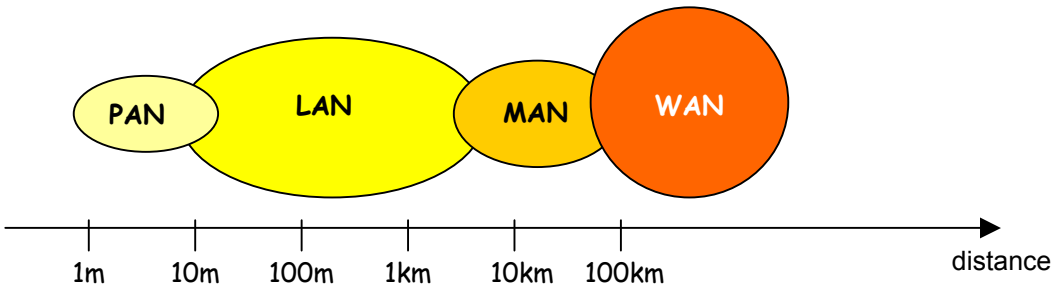
Ce type de réseau couvre une région géographique limitée (réseau intra-entreprise) et peut s'étendre sur plusieurs kilomètres. Les machines adjacentes sont directement et physiquement reliées entre elles. Les débits sont très importants (de qq Mb/s à qq Gb/s).

- **MAN : Metropolitan Area Network ou réseau métropolitain.**

Ce type de réseau possède une couverture qui peut s'étendre sur toute une ville et relie des composants appartenant à des organisations proches géographiquement(qq dizaines de kilomètres). Ils permettent ainsi la connexion de plusieurs LAN. Le débit courant varie jusqu'à 100 Mb/s.

- **WAN : Wide Area Network ou étendu.**

Ce type de réseau couvre une très vaste région géographique et permet de relier des systèmes dispersés à l'échelle planétaire (plusieurs milliers de km). Toutefois, étant donné la distance à parcourir, le débit est plus faible (de 50 b/s à qq Mb/s).



5.4.4 Topologie des réseaux

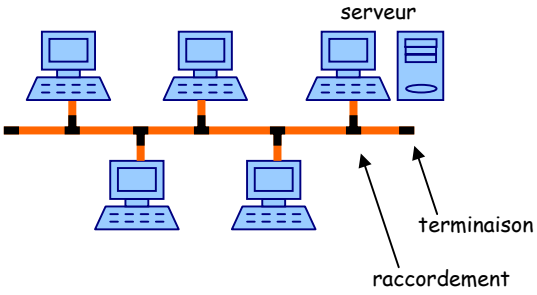
La topologie *physique* d'un réseau définie l'architecture du réseau. Elle peut être différente suivant le mode de transmission des informations.

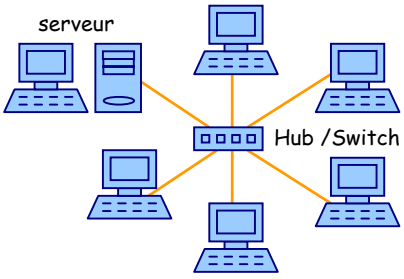
En mode de **diffusion**, chaque système partage le même support de transmission. Toute information envoyée par un système sur le réseau est reçue par tous les autres. A la réception, chaque système compare son adresse avec celle transmise dans le message pour savoir si il lui est destiné ou non. Il ne peut donc y avoir qu'un seul émetteur en même temps. Avec une telle architecture, la rupture du support provoque l'arrêt du réseau alors que la panne d'un des éléments ne provoque pas de panne globale du réseau.

Dans le mode **point à point**, le support physique relie les systèmes deux à deux seulement. Lorsque deux éléments non directement connectés entre eux veulent communiquer, ils doivent le faire par l'intermédiaire d'autres éléments. Les protocoles de routage sont alors très importants.

Topologie en bus

- Facile à mettre en œuvre.
- A tout moment un seule station a le droit d'envoyer un message.
- La rupture de la ligne provoque l'arrêt du réseau.
- La panne d'une station ne provoque pas de panne du réseau.
- mode de diffusion.



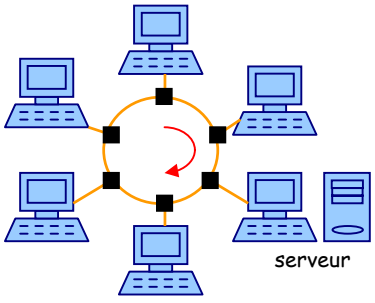


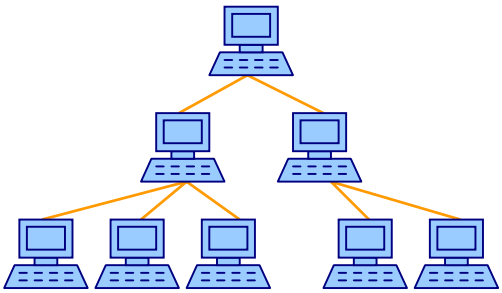
Topologie en étoile

- Le nœud central reçoit et renvoie tous les messages.
- Fonctionnement simple.
- Moins vulnérable sur rupture de ligne.
- La panne du nœud central paralyse tout le réseau.
- Mode point à point (avec *switch*) ou diffusion (avec *hub*).

Topologie en boucle

- Chaque station a tour à tour la possibilité de prendre la parole.
- Chaque station reçoit le message de son voisin en amont et le réexpédie à son voisin en aval.
- La station émettrice retire le message lorsqu'il lui revient.
- Si une station tombe en panne, il y a mise en place d'un système de contournement de la station.
- Si il y a rupture de ligne apparaît, tout s'arrête (sauf si on a prévu une 2^{ième} boucle).
- mode point à point.





Topologie en arbre

► Peut être considéré comme une topologie en étoile dans la quelle chaque station peut être une station centrale d'un sous ensemble de stations formant une structure en étoile.

► Complexe

► Mode point à point.

Topologie en réseau maillé

►Tous les ordinateurs du réseau sont reliés les uns aux autres par des câbles séparés.

► On a une meilleure fiabilité.

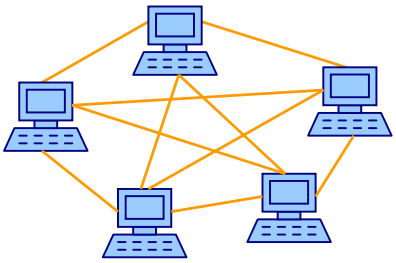
► Si une station tombe en panne, le réseau continue de fonctionner.

► Si il y a rupture de ligne apparaît, le réseau continue de fonctionner.

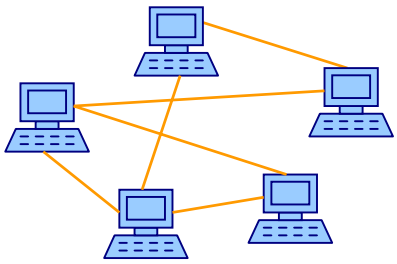
► Il est très coûteux.

► Il est possible de diminuer les coûts en utilisant un maillage partielle (irrégulier).

► Mode point à point.



Maillage régulier



Maillage irrégulier

Remarques :

Il existe deux modes de fonctionnement pour un réseau, quelque soit son architecture :

- avec connexion : l'émetteur demande un connexion au récepteur avant d'envoyer son message. La connexion s'effectue si et seulement si ce dernier accepte, c'est le principe du téléphone.
- sans connexion : l'émetteur envoie le message sur le réseau en spécifiant l'adresse du destinataire. La transmission s'effectue sans savoir si le destinataire est présent ou non, c'est le principe du courrier.

6 Un exemple - le PC

Chapitre
6

Le terme PC (**Personal Computer)** a été introduit en 1981 lorsque la firme IBM (Internal Business Machines) a commercialisé pour la première fois un ordinateur personnel destiné à une utilisation familiale. Depuis, les domaines d'application du PC ont énormément évolué…. De la gestion de production à la gestion de systèmes d'acquisition, en passant par la reconnaissance de forme ou le traitement de l'image, ses domaines d'utilisation sont extrêmement riches et variés. Pour cela, le PC est défini par une architecture minimale laissant la liberté à chacun de rajouter les périphériques d'entrée/sorties nécessaires à l'utilisation visée, qu'elle soit familiale ou professionnelle. Un PC est composé par une unité centrale associée à des périphériques (clavier, moniteur, carte d'acquisition, etc…)

6.1 L'unité centrale

Elle est composée par :

- La carte mère
- Le microprocesseur
- La mémoire
- La carte vidéo
- Les périphériques internes de stockage

6.1.1 La carte mère

La carte mère est l'un des éléments essentiels d'un ordinateur. Elle assure la connexion physique des différents composants (processeur, mémoire, carte d'entrées/sorties, ...) par l'intermédiaire de différents bus (adresses, données et commande). Plusieurs technologies de bus peuvent se côtoyer sur une même carte mère. La qualité de la carte mère est vitale puisque la performance de l'ordinateur dépend énormément d'elle. On retrouve toujours sur une carte mère :

- **le chipset :** c'est une interface d'entrée/sortie. Elle est constituée par un jeu de plusieurs composants chargé de gérer la communication entre le microprocesseur et les périphériques. C'est le lien entre les différents bus de la carte mère.

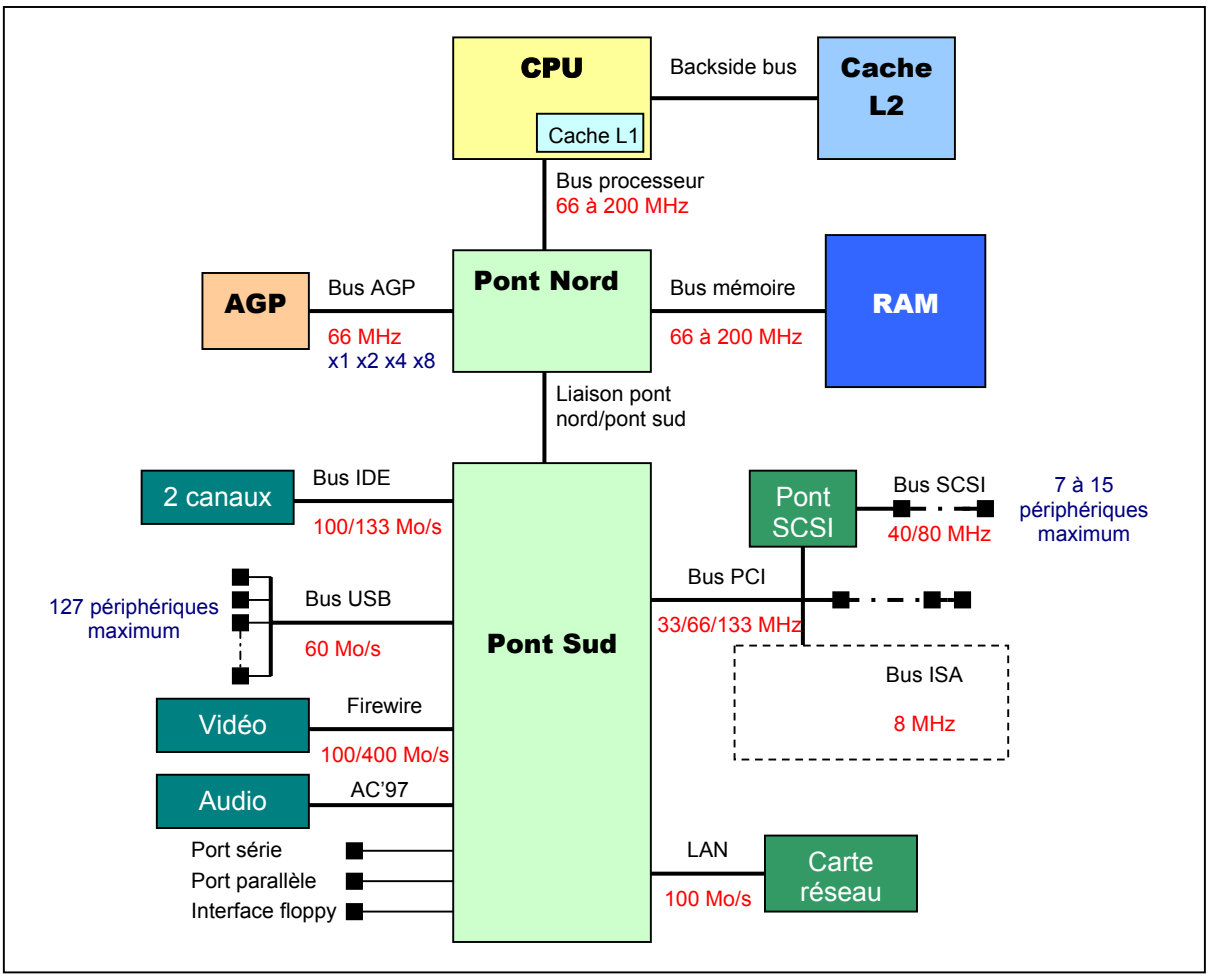
- **le BIOS** (Basic Input Ouput Service) **:** c'est un programme responsable de la gestion du matériel : clavier, écran, disques durs, liaisons séries et parallèles, etc... Il est sauvegardé dans une mémoire morte (EEPROM) et agit comme une interface entre le système d'exploitation et le matériel.

- **l'horloge :** elle permet de cadencer le traitement des instructions par le microprocesseur ou la transmission des informations sur les différents bus.

- **les ports de connexion :** ils permettent de connecter des périphériques sur les différents bus de la carte mère. Il existe des ports « internes » pour connecter des cartes d'extension (PCI, ISA, AGP) ou des périphériques de stockage (SCSI, IDE, Serial ATA) et des ports « externes » pour connecter d'autres périphériques (série, parallèle, USB, firewire, etc …)

- **Le socket :** c'est le nom du connecteur destiner au microprocesseur. Il détermine le type de microprocesseur que l'on peut connecter.

Architecture d'une carte mère



Ici le chipset est composé par deux composants baptisé Pont Nord et Pont Sud. Le pont Nord s'occupe d'interfacer le microprocesseur avec les périphériques rapides (mémoire et carte graphique) nécessitant une bande passante élevée alors que le pont sud s'occupe d'interfacer le microprocesseur avec les périphériques plus lents (disque dur, CDROM, lecteur de disquette, réseau, etc…).

On voit apparaître différents bus chargés de transporter les informations entre le microprocesseur et la mémoire ou les périphériques :

- **Bus processeur :** on l'appelle aussi bus système ou FSB (Front Side Bus). Il relie le microprocesseur au pont nord puis à la mémoire. C'est un bus 64 bits.

- **Bus IDE :** il permet de relier au maximum 2 périphériques de stockage interne par canal (disque dur ou lecteur DVDROM/CDROM). Son débit est de 133 Mo/s. Lorsque 2 périphériques sont reliés sur le même canal, un doit être le maître (prioritaire sur la prise du bus) et l'autre l'esclave.

- **Bus PCI** (Peripheral Component Interconnect) **:** Il a été créé en 1991 par Intel. Il permet de connecter des périphériques internes. C'est le premier bus à avoir unifier l'interconnexion des systèmes d'entrée/sortie sur un PC et à introduire le système plug-and-play. Il autorise aussi le DMA. C'est un bus de 32 bits. On retrouve une révision du bus PCI sur les cartes mères de serveur ayant une largeur de bus de 64 bits et une fréquence de 133 MHz.

- **Bus AGP** (Accelered Graphic Port) **:**. Il a été créé en 1997 lors de l'explosion de l'utilisation des cartes 3D qui nécessitent toujours plus de bandes passantes pour

obtenir des rendus très réalistes. C'est une amélioration du bus PCI. Il autorise en plus le DIME (DIrect Memory Execution) qui permet au processeur graphique de travailler directement avec les données contenues dans la RAM sans passer par le microprocesseur à l'instar d'un DMA. C'est un bus 32 bits et son débit maximum est de 2 Go/s (en x8).

- **Bus ISA** (Industry Standard Architecture) **:** C'est l'ancêtre du bus PCI. On ne le retrouve plus sur les nouvelles générations de cartes mères.

- **Bus SCSI** (Small Computer System Interface) **:** c'est un bus d'entrée/sortie parallèle permettant de relier un maximum de 7 ou 15 périphériques par contrôleur suivant la révision du protocole utilisée. C'est une interface concurrente à l'IDE qui présente l'avantage de pouvoir connecter plus de périphériques pour des débits supérieurs. En outre, ces périphériques peuvent partager le bus lors d'un dialogue contrairement à l'IDE. Mais son coût reste très élevé… elle est utilisée pour les serveurs.

- **Bus USB** (Universal Serial Bus) **:** c'est un bus d'entrée/sortie plug-and-play série. Dans sa deuxième révision (USB 2.0), il atteint un débit de 60 Mo/s. Un de ces avantages est de pouvoir connecter théoriquement 127 périphériques. Il supporte de plus le hot plug-and-play (connexion ou déconnexion de périphériques alors que le PC fonctionne).

- **Bus firewire :** c'est un bus SCSI série. Il permet de connecter jusqu'à 63 périphériques à des débits très élevés (100 à 400 Mo/s). Ces applications sont tournées vers la transmission de vidéos numériques.

- **Liaison pont nord/pont sud :** ses caractéristiques dépendent du chipset utilisé. Chaque fabricant a en effet développé une solution propriétaire pour connecter les deux composants de leur chipset. Pour Intel, c'est Intel Hub Architecture (IHA) dont les débits atteignent 533 Mo/s. Pour Nvidia (en collaboration avec AMD), c'est l'HyperTransport qui atteint des débits de 800 Mo/s.

Remarques :

❶ Tous les bus « internes » (PCI, IDE, AGP) vont être amenés à disparaître très rapidement et seront remplacés par des bus série :

- Le **Serial Ata**, remplaçant du bus IDE, présente des débits de 150 Mo/s qui passeront bientôt à 300 Mo/s dans la prochaine révision du bus. Il permet de connecter des disques durs ou des lecteurs optiques.

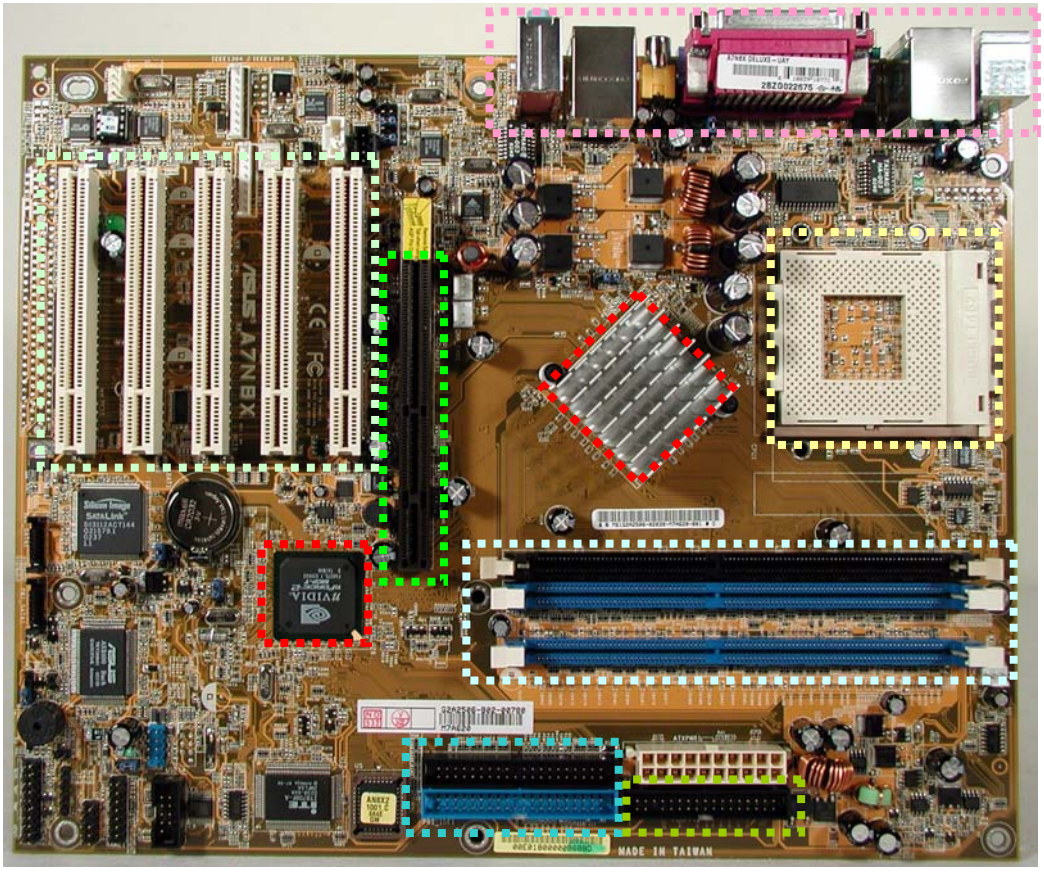
- Le **PCI Express**, remplaçant des bus PCI et AGP, permet d'atteindre des débits de 250 Mo/s dans sa version de base qui peuvent monter jusqu'à 8Go/s dans sa version x16 destinée à des périphériques nécessitant des bandes passantes très élevées (application graphique).

❷ Les bus de connexions filaires tendent à être remplacés par des systèmes de communications sans fils. A l'heure actuelle, il existe :

- le **Bluetooth** qui offre actuellement un débit de 1 Mb/s pour une portée d'une dizaine de mètre et qui va servir à connecter des périphériques nécessitant des bandes passantes faibles (clavier, souris, etc…).

- le **WIFI** (WIreless FIdelity Network) qui permet de connecter des ordinateurs en réseau. La dernière révision permet des débits de 54 Mb/s.

Exemple : Carte mère ASUS A7N8X

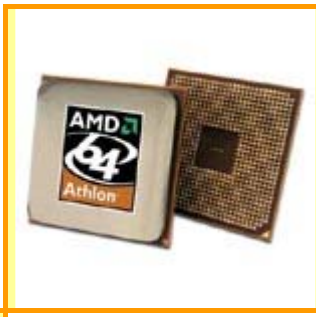
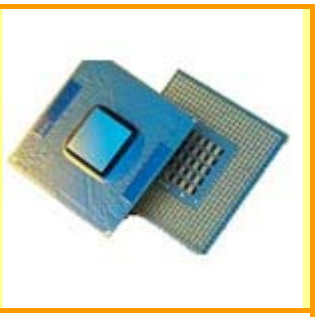


Connecteur PCI
Connecteur AGP
Connecteur RAM
Connecteur IDE
Chipset
Socket
Connecteurs Externes (port série, parallèle, firewire, USB, etc…)
Connecteur floppy

6.1.2 Le microprocesseur

Le microprocesseur est bien entendu l'élément essentiel du PC. Nous avons vu que les performances d'un microprocesseur étaient liées à son architecture et sa fréquence de fonctionnement. A l'heure actuelle, le marché des microprocesseurs pour PC est dominé par deux principaux constructeurs : Intel et AMD. Ceux-ci ont adopté deux stratégies différentes pour réaliser des microprocesseurs toujours plus performants. Intel, fort de son savoir faire, a choisi de fabriquer des microprocesseurs toujours plus rapide en terme de fréquence de fonctionnement alors qu'AMD essaie plutôt d'optimiser ses architectures afin qu'elles soient capables d'exécuter toujours plus d'instructions par cycle d'horloge. Ces deux optiques se retrouvent dans les références des microprocesseurs de chaque marque. Lorsqu'Intel désigne chaque nouveau microprocesseur par sa fréquence, AMD préfère utiliser un P-Rating se référant aux performances des microprocesseurs Intel. Chaque fondeur utilise des sockets et des chipsets différents pour leurs microprocesseurs. Ainsi, le choix d'un microprocesseur impose forcément un choix sur un type de carte mère. Pour connaître les performances d'un microprocesseur, il ne faut donc pas se fier à la seule valeur de sa fréquence de fonctionnement. Il faut prendre en compte toutes les caractéristiques liées à son architecture et ne pas oublier de l'entourer d'un chipset et d'une mémoire performants. La dernière chose à ne pas omettre lorsqu'on choisit un microprocesseur est son système de refroidissement. En effet, plus la fréquence augmente et plus la dissipation thermique sera importante. Un microprocesseur mal refroidit peut entraîner des dysfonctionnements au sein du PC voir même la destruction du microprocesseur lui même. Il faut prévoir un système d'air cooling (ventilateur + radiateur ou heat pipe) ou de water cooling (circuit de refroidissement à eau).

Exemple : Microprocesseur haut de gamme Décembre 2004

			
Référence	**Athlon 64 4000 +**	**Pentium 4 3.4GHz Extreme Edition**	**Pentium M 2GHz**
Support	Socket 939	Socket 478	Socket 478 (portable)
Fréquence	2400 MHz	3400 MHz	2000 MHz
Bus processeur	200 MHz	200 MHz quad pumped	100 MHz quad pumped
Finesse gravure	0.13 µm	0.13 µm	0.09 µm
Cache L1	128 ko	8 ko	32 ko
Cache L2	1024 ko	512 ko	2048 ko
Fréquence cache L2	2400 MHz	3400 MHz	2000 MHz
Architecture	AMD K8	Intel NetBurst	Intel Dothan

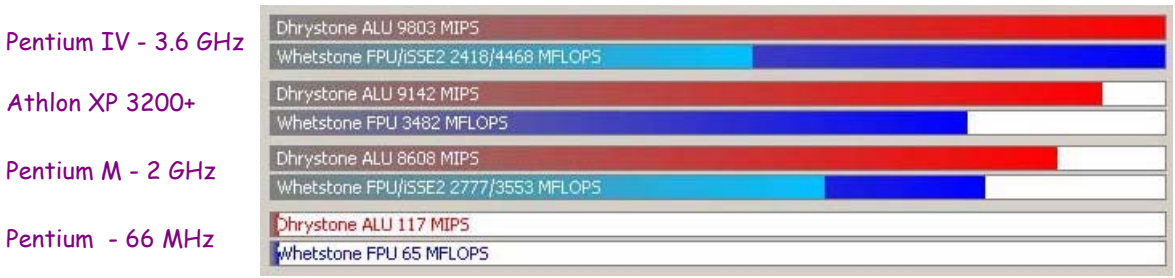
Exemple : Système de refroidissement

Heat Pipe

Ventirad

Kit Watercooling

Exemple : Performance en MIPS (traitement entier) et MFLOPS (traitement réel) de différents microprocesseurs.

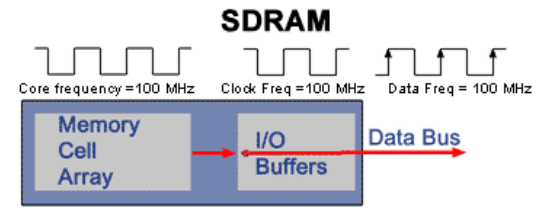


6.1.3 La mémoire

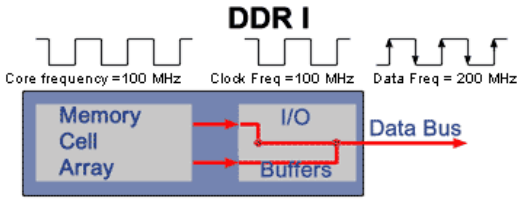
La qualité et la quantité de mémoire d'un PC vont permettre, au même titre que le microprocesseur, d'accroître les performances de celui-ci. Si on dispose d'un microprocesseur performant, encore faut-il que la mémoire puisse restituer ou sauvegarder des informations aussi rapidement qu'il le désire. La fréquence de fonctionnement de la mémoire est donc un paramètre essentiel. De même, si on veut réduire le nombre d'accès aux périphériques de stockage secondaire qui sont très lents (disque dur, CDROM, etc…), il faudra prévoir une quantité mémoire principale suffisante.

Aujourd'hui, toutes les mémoires que l'on retrouve sur les PC sont des RAM dynamiques (**DRAM**). Elles sont toutes synchronisées sur l'horloge du bus processeur (FSB). Un boîtier mémoire est constitué de 3 éléments fondamentaux qui sont :

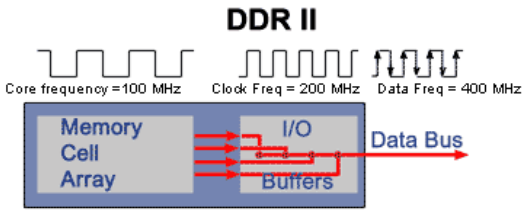
- La matrice de cellules mémoires
- Les buffers d'entrée/sortie
- Le bus de données



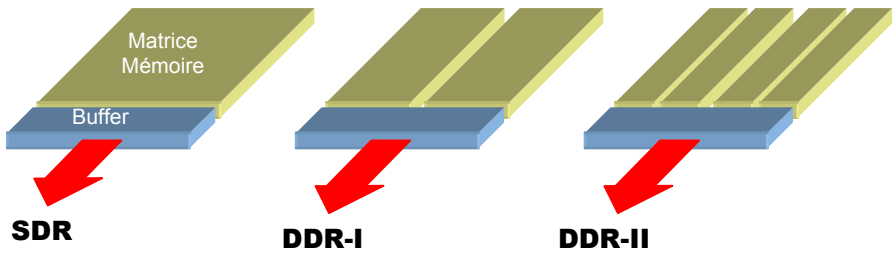
Dans les premières **SDRAM**, tous les ensembles fonctionnaient à 100 MHz. C'est à dire que la cellule mémoire fournissait une information mémoire toutes les 10 ns au buffer d'entré/sortie qui lui même la renvoyait sur le bus à une fréquence de 100 MHz. Comme les DRAM fonctionnent sur 64 bits, on avait une bande passante de 800 Mo/s. Les différentes évolutions de la SDRAM permirent d'atteindre une fréquence de 166 MHz.



Actuellement, les technologies de DRAM permettent d'effectuer des accès à la mémoire sur le front montant et descendant de l'horloge (**DDR-I SDRAM**) et ainsi de doubler la bande passante mémoire sans en modifier la fréquence de fonctionnement. Pour cela, il faut bien entendu que la matrice mémoire puisse délivrer 2 informations par cycle d'horloge. Les DDR les plus rapides permettent d'atteindre des fréquences de 200 MHz pour l'accès à la matrice de cellules. Néanmoins, on commence à approcher les limites de fonctionnement du cœur de la mémoire.



La prochaine technologie reviendra donc à une fréquence de 100 MHz pour la matrice de cellules mais doublera la fréquence du buffer d'entré/sortie pour compenser (**DDR II SDRAM**). Il faut donc que le cœur de la mémoire puisse délivrer 4 informations par cycle d'horloge. Tout ceci est rendu possible en divisant le nombre de matrices mémoire. Dans le cas de la SDRAM, la matrice de cellules mémoire est constituée d'un seul bloc physique contre deux pour la DDR-I puis quatre pour la DDR-II.



De plus, de nombreux chipsets permettent de gérer deux canaux d'accès à la mémoire et donc d'accéder simultanément à deux modules de mémoire différents. Le PC sera donc plus performant si il utilise, par exemple, deux barrette de 256 Mo plutôt qu'une seule de 512 Mo. (bus quad pumped d'Intel)

Exemple : Dénomination des mémoires SDRAM

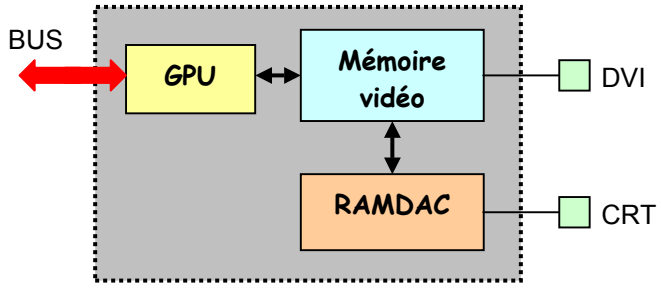
Désignation	Type	FSB	Vitesse	B.P.
PC 100	SDR	100 MHz	100 MHz	0,8 Go/s
PC-1600	DDR-I	100 MHz	200 MHz	1,6 Go/s
PC-2100	DDR-I	133 MHz	266 MHz	2.13 Go/s
PC-2700	DDR-I	166 MHz	333 MHz	2.66 Go/s
PC-3200	DDR-I	200 MHz	400 MHz	3,2 Go/s
PC-3500	DDR-I	216 MHz	432 MHz	3.5 Go/s
PC-3700	DDR-I	233 MHz	466 MHz	3.7 Go/s
PC-4000	DDR-I	250 MHz	500 MHz	4.0 Go/s
PC2-3200	DDR-II	100 MHz	400 MHz	3,2 Go/s
PC2-4300	DDR-II	133 MHz	533 MHz	4,3 Go/s
PC2-5300	DDR-II	166 MHz	667 MHz	5,3 Go/s

La bande passante est théoriquement doublée si les barrettes sont utilisées en “dual channel”.

6.1.4 La carte vidéo

Le rôle de la carte graphique est de convertir les données numériques à afficher en un signal compréhensible par un écran . Alors qu'à ses débuts, la carte vidéo se chargeait uniquement d'afficher une simple image formée de points colorées (pixel), les derniers modèles apparus se chargent d'afficher des images en 3D d'une grande complexité. C’est donc un système à microprocesseur à elle seule qui est composée par :

- Un GPU (Graphics Processor Unit)
- De la mémoire vidéo
- D’un dispositif de conversion analogique numérique : RAMDAC.
- D’entrées/sorties vidéo



La carte vidéo communique avec la mémoire centrale et le microprocesseur par l’intermédiaire d’un bus. Actuellement, c’est le bus AGP qui est le plus utilisé mais il va progressivement être remplacé par le PCI Express qui présente des débits beaucoup plus élevés (8 Go/s contre 2 Go/s).

6.1.4.1 Le GPU

Le GPU est le processeur central de la carte graphique. Il se charge du traitement des données vidéo, permettant ainsi de soulager le microprocesseur. Son rôle est de traiter les objets envoyés par le microprocesseur puis d'en déduire les pixels à afficher. En effet, dans le cas de l'affichage du scène 3D, le microprocesseur communique au GPU les données à afficher sous forme vectorielle. Les objets sont donc définis par une masse de points représentant leurs coordonnées dans l'espace. Pour afficher un objet à l'écran, le GPU procède en plusieurs étapes :

1. placer les objets dans le repère et leur appliquer des transformations (translation, rotation, etc…)
2. appliquer les effets de lumières sur chaque objet
3. décomposer les objets en petits triangles puis en fragments
4. appliquer des textures et des effets sur les fragments
5. afficher les pixel résultants de l'association des fragments

Pour cela, il est constitué d'un immense pipeline principal. Celui-ci comprend au moins un vertex shader (étape 1 et 2), un setup engine (étape 3) et un pixel shader (étape 4 et 5).

Remarques :

❶ toutes ces opérations doivent être effectuées pour tous les pixels de la scène à afficher. Pour une image en 1600x1200, cela fait 1 920 000 pixels à calculer, soit près de 6 millions de fragments !!!! D'autant plus que pour bien faire, le GPU doit être capable d'afficher 50 images/s soit calculer 300 millions de fragments par seconde… Ceci explique pourquoi les GPU des cartes 3D récentes sont plus complexes que les derniers microprocesseurs.

❷ Pour utiliser au mieux les capacités des cartes graphiques ont dispose d'API (Application Program Interface) qui sont des langages de description et de manipulation des objets :

- Direct3D de Microsoft
- OpenGL

6.1.4.2 La mémoire vidéo

Elle sert à stocker les images et les textures à afficher. Elle doit présenter des débits très important. Actuellement, la plupart des cartes graphiques sont dotées de DDR SDRAM.

6.1.4.3 Le RAMDAC

Le Ramdac (Random Access Memory Digital Analog Converter) convertit les signaux délivrés par la carte en signaux analogiques compatibles avec la norme VGA des moniteurs. Plus la fréquence du RAMDAC d'une carte graphique sera élevée, plus le rafraîchissement et la résolution de l'image pourront être élevés. Le confort visuel apparaît à partir d'un rafraîchissement de 72 Hz (fréquence à laquelle sont rafraîchies les lignes à afficher). En principe, la fréquence du RAMDAC est donc de l'ordre de :

Largeur écran x Hauteur écran x fréquence rafraîchissement x 1.32

On rajoute un coefficient de 1.32 à cause du temps perdu par le canon à électron lors de ces déplacements.

Exemple :
Pour une résolution de 1600x1200 à une fréquence de 85Hz, il faudra un RAMDAC de 1600x1200x85x1.32= 215 Mhz !!!

6.1.4.4 Les entrées/sorties vidéo

La sortie vers le moniteur se fait par l'intermédiaire d'une sortie au format VGA. Maintenant, la plupart des cartes disposent d'une sortie TV au format S-vidéo. Depuis l'explosion des écrans LCD, elles disposent aussi souvent d'un port DVI en plus du port VGA. Le port DVI est numérique et ne nécessite pas la traduction des données par le RAMDAC.

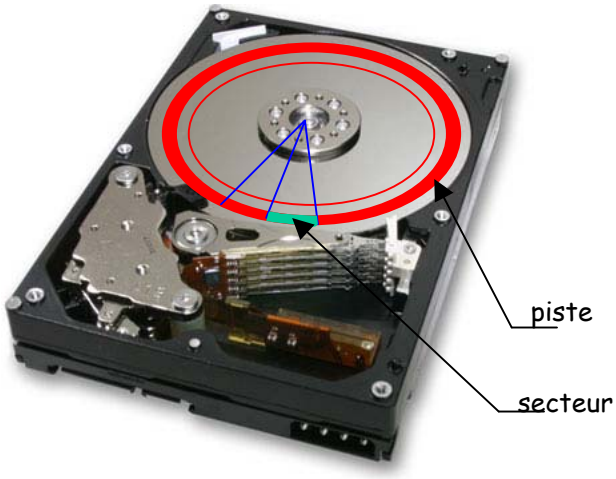
6.1.5 Les périphériques internes de stockage

Ce sont les périphériques de type mémoire de masse. On les appelle ainsi pour leur grande capacité de stockage permanent. Ces périphériques sont dotés d'un contrôleur permettant de les faire dialoguer avec le microprocesseur. Actuellement, les plus répandus sont l'IDE et le SCSI. Le SCSI présente des débits plus importants que l'IDE (160Mo/s contre 133Mo/s) et permet de connecter plus de périphériques sur le même contrôleur (7 contre 4). Néanmoins, cette technologie étant plus onéreuse, on la retrouve surtout sur des serveurs alors que l'IDE est présent dans tous les PC. A l'heure actuelle, ces deux types de contrôleur sont en fin de vie et sont progressivement remplacés par des contrôleur de type Serial ATA. Ce sont des contrôleurs série dérivés de l'interface IDE qui vont permettent d'atteindre des débits de 600 Mo/s.

Les périphériques internes de stockage sont principalement des périphériques utilisant des supports magnétiques (disque dur) ou optiques (CDROM, DVDROM).

6.1.5.1 Le disque dur

Les disques durs sont capables de stocker des quantités impressionnantes d'informations, et surtout de les ordonner et de les retrouver rapidement.

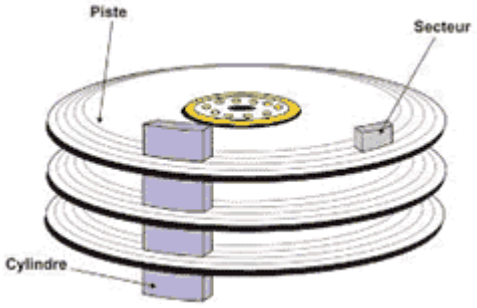


Principe :

Le disque dur est constitué de plusieurs plateaux empilés, entre lesquels se déplace un bras comptant plusieurs têtes de lecture. Chaque plateau est recouvert d'une surface magnétique sur ses deux faces et tourne à une vitesse comprise entre 4000 et 15000 tr/min. La tête de lecture/écriture est composée par un aimant autour duquel est enroulée une bobine.

Pour écrire, on fait passer un courant électrique dans la bobine ce qui crée un champ magnétique. Les lignes de champ magnétique traversent la couche d'oxyde et orientent celui-ci en créant de petits aimants dont le sens est donné par le sens du courant dans la bobine.

Pour lire, on fait passer la tête de lecture/écriture sur le support magnétisé qui crée un courant induit dans la bobine dont le sens indique s'il s'agit d'un 0 ou d'un 1.



Le formatage :

Le formatage de *bas niveau* permet d'organiser la surface du disque en éléments simples (**pistes** et **secteurs**) qui permettront de localiser l'information. Le nombre total de pistes dépend du type de disque. Il est effectué en usine lors de la fabrication du disque. Chaque piste est découpée en secteurs. Toutefois l'unité d'occupation d'un disque n'est pas le secteur, trop petit pour que le système puisse en tenir compte. On utilise alors un groupe d'un certain nombre de secteurs (de 1 à 16) comme unité de base. Ce groupe est appelé Bloc ou Cluster. C'est la taille minimale que peut occuper un fichier sur le disque. Pour accéder à un secteur

donné, il faudra donc déplacer l'ensemble des bras et attendre ensuite que ce secteur se positionne sous les têtes. L'accès à un bloc est aléatoire alors que l'accès à un secteur est séquentiel.

Une autre unité de lecture/écriture est le **cylindre**. Un cylindre est constitué par toutes les pistes superposées verticalement qui se présentent simultanément sous les têtes de lecture/écriture. En effet, il est plus simple d'écrire sur les mêmes pistes des plateaux superposés que de déplacer à nouveau l'ensemble des bras.

Le formatage de *haut niveau* permet de créer un système de fichiers gérable par un système d'exploitation (DOS, Windows, Linux, OS/2, etc ...).

La défragmentation :

A mesure que l'on stocke et supprime des fichiers, la répartition des fichiers sur les différents clusters est modifiée. L'idéal, pour accéder rapidement à un fichier, serait de pouvoir stocker un fichier sur des clusters contigus sur le même cylindre. La défragmentation permet de réorganiser le stockage des fichiers dans les clusters pour optimiser la lecture.

Les caractéristiques :

- capacité en Go
- vitesse de rotation en tours minutes
- temps d'accès exprimé en millisecondes
- interface (IDE, SCSI, SATA)
- taux de transfert moyen exprimé en Mo par seconde

A noter que les disques durs actuels sont équipés de cache mémoire afin de diminuer les temps d'accès.

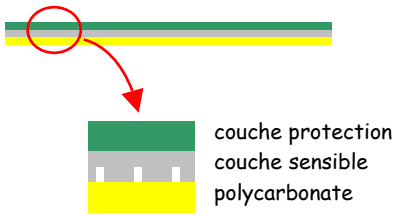
6.1.5.2 Les disques optiques

Le disque optique numérique résulte du travail mené par de nombreux constructeurs depuis 1970. La terminologie employée varie selon les technologies employées et l'on retrouve ainsi les abréviations de CD (Compact Disk), CDROM (CD Read Only Memory), CDR, (CD Recordable), DVD (Digital Video Disk), DVDROM (DVD Read Only Memory), etc… Le Compact Disc a été inventé par Sony et Philips en 1981 dans le but de fournir un support audio et vidéo de haute qualité. Les spécifications du Compact Disc ont été étendues en 1984 afin de permettre au CD de stocker des données numériques. En 1990 Kodak met au point le CD-R. Un CD est capable de stocker 650 ou 700 Mo de données et 74 ou 80 min de musique. Le taux de transfert d'un CD-ROM est de 150 ko/s, ce qui correspond au taux de transfert d'un lecteur de CD audio. On peut monter jusqu'à 7200 ko/s (48X) avec un lecteur de CDROM.

Principe CD-ROM:

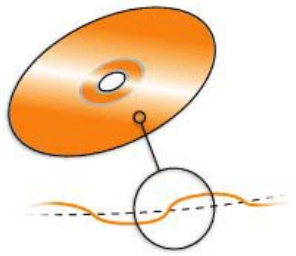
Un CD-ROM est un disque de 12 cm de diamètre composé de plusieurs couches superposées :

- une couche principale en polycarbonate, un plastique résistant et laissant passer la lumière
- une couche métallique réfléchissante composée de plats et de creux
- une couche de vernis protecteur qui vient protéger le métal de l'agression des UV



Plusieurs technologies différentes existent en fonction du type de CD : CD-ROM, CD-R, CD-RW. Le principe de lecture/écriture utilise un rayon infrarouge d'une longueur d'onde de 780 nm.

Lors de la lecture d'un CD, le faisceau laser traverse la couche de polycarbonate puis rencontre ou non un creux. Lors d'un passage devant un creux, la lumière du laser est fortement réfractée, de telle sorte que la quantité de lumière renvoyée par la couche réfléchissante est minime. Alors que pour un passage devant un plat, la lumière est pratiquement entièrement réfléchie. Lorsque le signal réfléchi change, la valeur binaire est 1. Lorsque la réflexion est constante, la valeur est 0. A noter que contrairement aux disques durs, un CD n'a qu'une seule piste organisée en spirale.



Cette piste n'est pas régulière mais oscille autour de sa courbe moyenne. La fréquence de ces oscillations est de 22,05 kHz. Cette oscillation permet à la tête de lecture de suivre la courbe et de réguler la vitesse de rotation du CD.

Pour l'écriture, il faut utiliser un graveur avec des supports adéquates (CD-R ou CD-RW). Les techniques sont assez similaires qu'il s'agisse d'un CD-R ou d'un CD-RW. Dans le cas d'un CR-R, on ajoute une couche de colorant organique pouvant être brûlé par un laser 10 fois plus puissant que le laser requit pour lire un CD. Cette couche de colorant est photosensible. Lorsqu'elle est soumise à une forte lumière, elle l'absorbe et sa température augmente à plus de 250°, ce qui fait qu'elle brûle localement, et crée des plages brûlées et non brûlées. Les creux et bosses du CD classique sont donc ici remplacés par le passage d'une zone brûlée à une zone non brûlée qui réfléchisse plus ou moins de lumière. Pour les CD-RW, on utilise un alliage métallique qui possède la particularité de pouvoir retrouver son état d'origine en utilisant un laser à 200 degrés (effacement).

Les méthodes d'écriture :

- Monosession : Cette méthode crée une seule session sur le disque et ne donne pas la possibilité de rajouter des données sur le CD.
- Multisession : Cette méthode permet de graver un CD en plusieurs fois, en créant une table des matières (TOC pour table of contents) de 14Mo pour chacune des sessions.
- Track At Once : Cette méthode permet de désactiver le laser entre deux pistes, afin de créer une pause de 2 secondes entre chaque pistes d'un CD audio.
- Disc At Once : Contrairement à la méthode précédente, le Disc At Once écrit sur le CD en une seule traite. Les musiques sont donc enchaînées.

Les techniques de gravures :

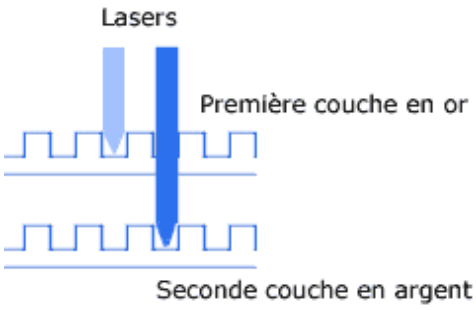
- Burn Proof ou Just Link : Le problème des graveurs était l'envoi des données à un rythme suffisant. Lorsque les données n'étaient plus présentes dans le buffer du graveur, il y avait une rupture de flux. Ceci entraînait l'arrêt de la gravure par manque de données et le CDR était inutilisable. Pour corriger ce type d'erreurs, les fabricants utilisent maintenant des techniques qui suspendent la gravure lorsque les données ne sont pas présentes, et la reprend dès que les données sont de nouveau présentes dans le buffer. Cette technique est appelée JUST LINK chez la majorité des fabricants, Burn-Proof chez Plextor.
- L'overburning : cette technique permet de dépasser légèrement la capacité du support vierge afin de stocker un peu de données supplémentaires. Pour ce faire, il faut que le logiciel de gravure, ainsi que le graveur, supportent cette technique.

Caractéristiques d'un lecteur/graveur :

- la vitesse maximum de gravage des CD-R
- la vitesse maximum de gravage des CD-RW
- la vitesse maximum de lecture des CD
- interface (IDE, SCSI, SATA)

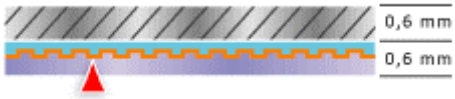
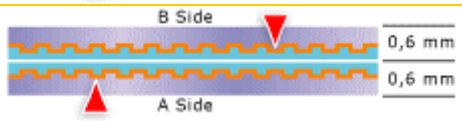
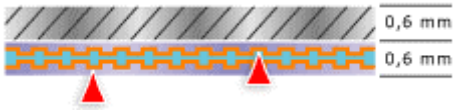
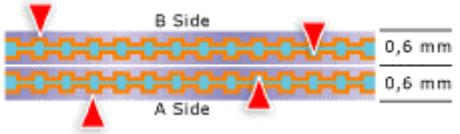
Principe DVDROM :

Le DVD-ROM (***D**igital **V**ersatile **D**isc - **R**ead **O**nly **M**emory*) est apparu en 1997 et est principalement dédié à la vidéo. C'est en fait un CD-ROM dont la capacité est bien plus grande. En effet, la lecture/écriture est effectuée à partir d'un laser rouge (650 et 635 nm) et permet d'obtenir des creux beaucoup plus petits et donc de stocker plus d'informatons. Les deux longueurs d'ondes utilisées permettent de lire/écrire sur des DVD "double couche". Ces disques sont constitués d'une couche transparente et d'une couche réflexive et permettent donc de stocker encore plus d'informations sur un seul CD.



Il existe 3 types de DVD réinscriptibles et incompatibles :

- DVD-RAM : le disque simple face permet de stocker 2.6 Go. Il n'est pas compatible avec les lecteurs de salon.
- DVD-RW de Sony, Philips et HP permet de stocker 4.7Go par face. Il est entièrement compatible avec les platines de salon.
- DVD+RW est le nouveau standard concurrent au DVD-RW. Il est entièrement compatible avec les platines de salon. Plusieurs marques ont formé une alliance et développent des graveurs DVD présentant des temps d'accès plus faible et des vitesses de gravure plus importante.

Type de support	Capacité	Nombre de CD	
CD	800 Mo	1	
DVD-RAM	2.6 Go	4	
DVD-RW/+RW simple face simple couche	4.7 Go	6	
DVD-RW/+RW double face simple couche	9.4 Go	12	
DVD-RW/+RW simple face double couche	8.5 Go	11	
DVD-RW/+RW double face double couche	17 Go	22	

Bibliographie

Cours Web :

A brief history of Intel and AMD microprocessors (cours DEUG Université Angers) – *Jean-Michel Richer*
Architecture Avancée des ordinateurs (cours Supelec Rennes) – *Jacques Weiss*
Architecture des ordinateurs (cours IUT GTR Montbéliard) – *Eric Garcia*
Architecture des ordinateurs (cours IUT SRC Marne la Vallée) – *Dominique Présent*
Architecture des ordinateurs (cours Université Franche Compté) – *Didier Teifreto*
Architecture des ordinateurs (cours IUP STRI Toulouse)
Architecture des ordinateurs (cours Université de Sherbrooke) – *Frédéric Mailhot*
Architecture des ordinateurs (cours Polytechnique) – *Olivier Temam*
Architecture des ordinateurs (cours IUT GTR Villetaneuse) – *Emmanuel Viennet*
Architecture des ordinateurs (cours DEUG MIAS) – *Frédéric Vivien*
Architecture des Ordinateurs (cours Licence Informatique USTL) – *David Simplot*
Architecture des machines et systèmes Informatiques – *Joëlle Delacroix*
Architectures des processeurs (cours DEUST Nancy) – *Yannick Chevalier*
Architecture des systèmes à microprocesseurs – *Maryam Siadat et Camille Diou*
Architecture des systèmes à microprocesseurs (cours IUT Mesures Physiques) – *Sébastien Pillement*
Architecture Systèmes et Réseaux (cours DEUG 2ième année) – *Fabrice Bouquet*
Carte graphique (ENIC) – *Julien Lenoir*
Cours de réseau (cours EISTI) – *Bruno Péant*
Cours de réseaux (cours Maîtrise Informatique Université Angers) – *Pascal Nicolas*
Du processeur au système d'exploitation (cours DEUST Nancy) – *Yannick Chevalier*
Introduction to computer architecture (cours DEUG Université Angers) – *Jean-Michel Richer*
Les réseaux : introduction (DESS DCISS) – *Emmanuel .Cecchet*
Les systèmes informatiques (cours CNAM) – *Christian Carrez*

Sites web :

Fonctionnement des composants du PC
http://www.vulgarisation-informatique.com/composants.php

Cours d'initiation aux microprocesseurs et aux microcontrôleurs
http://www.polytech-lille.fr/~rlitwak/Cours_MuP/sc00a.htm

Architecture des ordinateurs – Université Angers
http://www.info.univ-angers.fr/pub/richer/ens/deug2/ud44/

Les docs de Heissler Frédéric
http://worldserver.oleane.com/heissler/

X-86 secret
http://www.x86-secret.com/

Le cours hardware d'YBET informatique
http://www.ybet.be/hardware/hardware1.htm

Informa Tech
http://informatech.online.fr/articles/index.php

Articles presse :

Mémoire Flash – article Electronique Juillet 98
Les processeurs numériques de signal – article Electronique Janvier 2004
Fonctionnement d'un processeur et d'une carte graphique – article Hardware magazine Novembre 2003

Livres :

Architecture et technologie des ordinateurs (Dunod) – *Paolo Zanella et Yves Ligier*
Technologie des ordinateurs et des réseaux (Dunod) – *Pierre-Alain Goupille*
Les microprocesseurs, comment ça marche ? (Dunod) – *T. Hammerstrom et G. Wyant*