

# Informatique Industrielle

## Le développement du calcul

Les différentes évolutions des instruments et des méthodes de calcul

- Boulier chinois
- Algèbre Al-Khawarizmi
- Pascaline
- Calculateur de Babbage (déjà des programmes dessus)
- Z3 Allemand pendant la 2ème guerre mondiale (Premier Ordinateur)
- ENIAC (fonctionne avec tubes a vide)
- Modèle de Van Neumann, remplacé par le modèle d'Harvard.

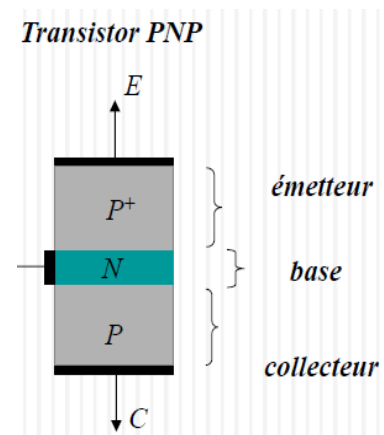
## Le développement de l'électronique

L'un des éléments les plus important dans l'évolution de l'électronique reste le transistor. Le transistor était composé de deux jonctions PN (qui signifient Positif, Négatif). Ces jonctions PN sont également connues sous le nom de "diode". La mise en place côte a côte de deux jonctions donne lieu a "l'effet transistor", une barrière de potentiel va se créer. Il y'a toujours une symétrie dans la jonction, on ne retrouvera jamais deux parties positives côte a côte, il faudra toujours la séparer par une partie du signe inverse.

p.e = NPN ou PNP sont des transistor valables. Par contre, PPN ou PNN ne sont pas des transistor.

Un transistor est toujours séparé en trois parties. L'émetteur, la base et le collecteur. La base est la partie de signe inverse qui séparera les deux parties de même signe P ou N. Ces deux parties seront nommées émetteur et récepteur (en fonction du mouvement des électrons). C'est ce mouvement des électrons qui produira le courant.

Le transistor est souvent composé soit de silicium, soit de germanium. Il est également important de noter que le transistor est un semi-conducteur. Un semi-conducteur est un matériau ayant les caractéristique d'un isolant, mais pour lequel la probabilité qu'il puisse contribuer a un courant électrique est suffisamment importante. La conductivité est a mi-chemin entre l'isolant est le conducteur.



Les transistors servent à faire des portes logique ( OU, ET ...). Ces mêmes portes serviront à faire des circuits intégrés qui pourront appliquer des fonctions logique. Au final, c'est ce principe qui nous permettra d'obtenir un **micro-processeur**.

Les portes logique sont donc composées de micro-processeur et qui fonctionnent comme des interrupteur. Il y'a deux types de portes, les portes, **MOS** et les portes **CMOS**. C'est à partir de ces portes qu'on créera les fonctions logique comme OR, AND, NOT, XOR, etc... Ce sont ces fonctions de base qui permettront de former les autres fonctions comme l'addition, la soustraction etc...

Le premier micro-processeur a été un micro-processeur de 4 bits qui tourne à 108 KHz. C'est à dire qu'il effectue une instruction en un centième de milliseconde. Il ne comportait que 2300 transistor sur une taille de 10 microns. En 2000, le micro-processeur Pentium 4 sort en ayant 42.000.000 de transistor, l'avancée a été rapide.

Pour ce qui est de la composition, le micro-processeur est composé par une unité de commande, une unité arithmétique et logique (ALU) et des registres, ce sont les trois grands éléments qui la composent.

C'est ces inventions qui permettront en grande partie l'évolution pour en arriver aux PC d'aujourd'hui, bien qu'ils soient accompagnés d'autres éléments comme la RAM, la souris, le clavier, l'interface graphique ou le disque dur. D'année en année, toutes ces technologies évoluent et s'améliorent.

### **Le schéma interne d'un micro-processeur**

Le micro-processeur est composé de trois **bus**.

- Bus d'adresses
- Bus de données
- Bus de commande

Un **bus** est un moyen d'acheminer les données entre divers composants. Il peut être uni-directionnel (bus d'adresse) ou bidirectionnel (bus de donnée).

Le micro-processeur est également composé de **registres** :

- Le registre d'instruction
- Le registre d'adresse
- Le registre auxiliaire
- Le registre d'état
- Le registre de données
- L'accumulateur
- le SP ou Stack Pointeur

Un **registre** est un emplacement de mémoire interne au processeur.

Le **PC**, aussi connu sous le nom de "compilateur de programme" se charge de garder en mémoire l'adresse de la première instruction du programme exécuté via un registre interne.

L'**ALU**, aussi connue sous le nom "Unité Arithmétique et Logique" est un circuit qui se charge d'assurer les fonctions arithmétique, logique, etc...

Le **Bloc Logique de Commande** organise les instruction au rythme d'une horloge et se charge de la synchronisation de toutes ces instructions.

Le **SP**, ou **Stack Pointeur** est la pile dans laquelle on va sauvegarder les données des différents registres lors de l'appel d'un sous-programme ou lors d'une interruption. Le pointeur sert à déterminer le début de le commencement et la fin des données enregistrés sur la pile.

L'**Accumulateur** sert à stocker le résultat de fin d'un calcul ou des données temporaire provenant de l'extérieur du processeur ou encore à stocker des données qui doivent être envoyés vers l'extérieur du processeur.

Le **registre d'état** contient différents bits dont chacun est un indicateur concernant l'opération effectuée. Ces **indicateur** peuvent être :

- **la retenue** (retenir la valeur à ajouter à l'unité supérieure)
- **le dépassement** (permet de décompter si les valeurs ne sont pas supportées, le dépassement ajoutera ce qu'il faut à la réponse finale pour la rendre vraie)
- **le zéro** (vérifie si les termes sont égaux ou nuls pour ne pas devoir faire de calcul)
- **la parité** (définir la parité ou pas de la réponse)

## Systeme minimum

Le microprocesseur est donc puissant, mais seul, il ne peut pas concevoir un programme. Il lui faut au minimum quelques composants supplémentaires. La RAM, la ROM ainsi que l'I/O (interface d'entrée/sorties). Entre ces différents éléments, il doit y avoir un échange de données qui passera via les bus de commande, de données et d'adresse.

## Les microcontrôleurs

L'évolution des technologies mène directement souvent à une facette intéressante, la miniaturisation, mais elle dépend de l'avancée technologie des composants et de l'intégration de ces composants dans un seul composant.

C'est ce principe qui est exploité par les microcontrôleurs, on intègre le système

minimum dans un seul circuit.

### Quelles sont les différences entre le système à microprocesseur et le microcontrôleur ?

Les système à microprocesseur prennent plus de place matérielle et demande une bonne connectique, les microcontrôleurs s'affranchissent de ces contraintes car ils intègrent tout en un seul circuit.

#### Architecture de Von Neumann

C'est l'architecture utilisée par la plupart des système microprogramées. Le traitement d'une instruction nécessite la lecture d'au moins deux cases mémoire (ou 3 si l'opérande est codé sur deux octet), ceci correspond à deux ou trois cycle machine.

#### Architecture Harvard

C'est l'architecture utilisée pour les microcontrôleur. La différence réside dans le fait que le traitement d'une instruction ne nécessite que la lecture d'une case mémoire, ce qui correspond à un seul cycle machine.

#### Les registres dans les microcontrôleurs

Ils sont nombreux et leurs tâches varient :

- **Le registre de travail** : C'est un registre 8 bits destiné à la manipulation générale des données. Il peut contenir une donnée de 8 bits nommée littéral. Les microcontrôleurs doivent transmettre et recevoir des données, ces registres sont donc utilisés pour écrire et lire dans le registre correspondant.
- **Le registre de direction TRIS** : Leur rôle est de programmer chacune des lignes des ports, soit en entrée, soit en sortie. Les différentes broches d'un port peuvent donc transmettre ou recevoir des données.
- **Le registre d'état STATUS** : Son rôle est de donner des infos concernant le fonctionnement ou la réponse d'une opération à travers les indicateur Z (zéro) et C (carry = retenue).
- **Le registre PC** : c'est un compteur ordinal qui contient l'adresse de la prochaine instruction à exécuter. Lorsqu'une instruction est exécuter la suivante est déjà pointée.

#### La base de temps

Tout les circuit microprogrammés fonctionnent à partir d'une base de temps (une

horloge). Son rôle est de cadencer les différentes opérations effectuées et notamment l'exécution des instructions. Un cycle machine correspond à quatre période d'horloge.

### Jeu d'instruction et mode d'adressage

le jeu d'instruction est composé de différents mnémonique (série de caractère correspondant à une instruction) qui correspondent à un littéral sur 8 bit, une correspondance dans un registre, le numéro du bit concerné ou encore le registre dans lequel la réponse doit être stockée.

Quand au mode d'adressage il peut différer en fonction de la manière utilisée et être immédiat, direct, bit à bit ou indirect.

### L'automate programmable

Un automate programmable est un appareil qui commande un processus et qui est très utilisé dans le milieu industriel. Les instructions sont stockés dans la mémoire de l'appareil et lui permet donc de faire des tâches qui varieront d'un automate à l'autre.

L'automate programmable industriel (**API**) est la machine qu'on rencontrera dans tout les systèmes automatisés dans le milieu industriel. Son invention est intervenue lorsqu'il a fallu remplacer la logique câblée (relais électronique) par la logique programmée.

L'API est composé de beaucoup de composant qu'on retrouve dans l'ordinateur :

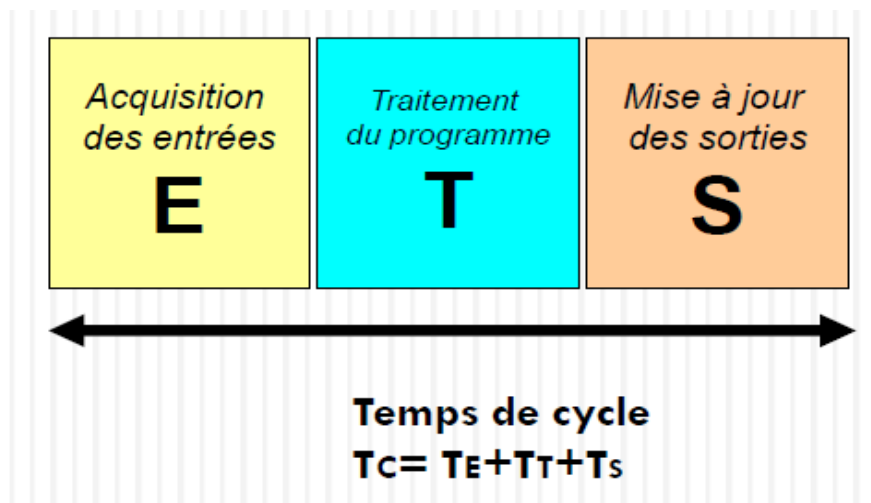
- ROM
- RAM
- Alimentation
- Unité de traitement
- Horloge
- Cartes entrée
- Cartes sortie
- Cartes spécifique

### Le fonctionnement de l'API

L'API fonctionnera toujours de la manière suivante pour traiter une tâche :

- Il procède à l'écriture en mémoire de l'état des informations présente dans les cartes entrée. On appelle cette partie **E** (Acquisition des **E**ntrées).
- Il exécute le programme écrit par l'utilisateur. On appelle cette partie **T** (Traitement du programme).
- Il met à jour les cartes sorties en procédant à une écriture, à noter que ces écritures peuvent être des instructions à envoyer, comme le fait de fermer une vanne. On appelle cette partie **S** (Mise à jour des **S**orties).

**Le temps de cycle** sera défini par l'addition des trois temps que prend l'automate pour exécuter ces actions. L'ensemble de ces actions défini **une tâche**. Il est important de noter que ces trois actions se passeront à l'intérieur du CPU, l'action réelle se verra sur l'actionneur (les vannes qui se ferment, le mécanisme qui s'enclenche).



Il est également important de noter que l'automate a besoin d'une horloge. Cette horloge sert à cadencer un fonctionnement cyclique pour enchaîner les cycles **ETS** les uns après les autres.

### Les avantages des API

- Les API sont très évolutifs, assurant plusieurs fonctions, on peut passer d'une simple application à un programme bien plus complexe.
- La vitesse (le temps de cycle) est très rapide, de l'ordre de quelques millisecondes, et cela tend à s'améliorer.
- Il y'a une grande modularité, l'automate est facilement modifiable et cela se montre par un grand nombre d'entrées sorties.
- Le développement d'une application pour l'automate est de plus en plus simple grâce aux outils de programmations correspondant.
- L'architecture de commande permet de régler l'interface de manière centralisée ou à distance, beaucoup plus de facilité grâce aux offres de réseaux, aux bus de

terrain et aux I/O.

- La mise en oeuvre est plus simple, possibilité de faire des simulation avant de passer à la réalité.
- Maintenance simple grâce a l'aide intégrée et diagnostic plus simple.
- Les applications sont plus portable d'une machine à l'autre.

## Langages de programations pour les automates

Les langages sont divers et variés et divergent de langages plus écrits. Il y'a par exemple **FBD** (Function Blocks Diagram) qui fonctionne avec une programmation de blocs. D'autres langages de programmations comme le **Ladder Diagram** qui expliquent une logique.

L'**IL** (Instruction List) est un langage plus écrit, qui ressemble au langage connu, tout comme le **ST** (Structured Text). Le **GRAFCET** fonctionne avec un système de condition.

## Applications des automates

Il faut savoir qu'en Europe, les principaux fabricants d'automates sont Siemens et Schneider Electric.

Les applications sont diverses. Un automate peut gérer les commande d'un moteur, commander des procédés industriels... Il est important de savoir que les automates peuvent communiquer entre eux et se laisser des tâches en fonction de ce qui reste a accomplir. On comprends donc que dans des applications comme la fabrications de bouteille d'eau, la communication est essentielle.

## La communication

La communication se fait entre un émetteur et un récepteur via un **médium** (qui va permettre à un émetteur de devenir un récepteur et vice-versa). La communication implique le fait que des données vont être échangées.

Ces données peuvent être sous forme **analogique** ou **numérique**. Dans le cas des données analogiques la valeur va évoluer de manière continue, dans le cas de la valeur numérique, l'évolution sera discontinue (0 ou 1). Il est important de noter qu'une information analogique doit être **numérisée** pour pouvoir être lue par des **convertisseurs**.

## Types de transmission

Il y'a trois type de transmissions :

- La transmission **simplex**, qui vas dans un seul sens.

- La transmission **half duplex**, qui fait alterner émission et réception a chacun des deux émetteur/récepteur.
- La transmission **full duplex** qui est une communication continue dans les deux sens. Les émetteur/récepteur sont utilisés a leur maximum.

La transmission peut être **en série**, elle nécessite alors trois fils pour l'émission, la réception et la masse. Ici les bits d'un octet sont transmis les uns a la suite des autres. C'est la transmission la plus utilisée. La transmission peut également être **en parallèle**. Ici, les bits d'un octet sont transmis simultanément. La transmission se fait sur des courtes distance, car la qualité est faible et se dégrade rapidement.

Il y'a deux type de transmission en série. La transmission **série synchrone** et la transmission **série asynchrone**. Pour la transmission **série synchrone**, les informations sont transmises de façon continue. Un signal de synchronisation est transmis en parallèle aux signaux de données pour ne pas perdre d'informations. Pour la transmission **série asynchrone**, les informations sont transmises de manière irrégulière, mais l'intervalle entre deux transmission est fixe. On utilise le système de **bit de START** et **bit de STOP** pour indiquer quand commence et quand termine l'information.

### Les supports utilisés

Les supports les plus utilisés sont :

- La paire de **fils torsadés**. C'est la plus simple a mettre en oeuvre et la moins chère.
- La **câble coaxial** qui dispose de grandes qualités électriques et permet d'effectuer des transmissions a grande vitesse. Il est très bien isolé.
- La **fibres optiques**, c'est un câble qui transmet des signaux lumineux, conviens pour les transmissions sûres et les longues distances. C'est également le médium le plus cher.

Le choix de l'un ou l'autre va influencer sur la vitesse, la distance ainsi que l'immunité électro-magnétique.

### Les différentes topologies

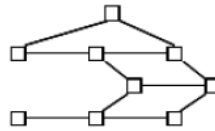
Lorsqu'il faut communiquer dans un réseau, on peut imaginer différentes topologies pour communiquer, les schémas peuvent être très différents les uns des autres :



TOPOLOGIE **POINT A POINT** (entre 2 unités en communication)

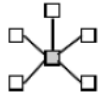


TOPOLOGIE **MAILLEE**



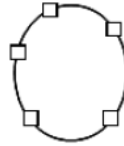
(les équipements sont reliés entre eux pour former une toile d'araignée. Pour atteindre un noeud, plusieurs chemins sont possibles)

TOPOLOGIE EN **ETOILE**



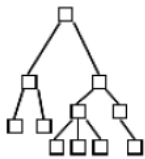
(plusieurs unités communiquent par leur propre ligne avec une unité dite Centrale)

TOPOLOGIE EN **ANNEAU**



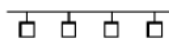
(toutes les unités sont montées en série dans une boucle fermée. ⇒ les communications doivent traverser toutes les unités pour arriver au récepteur)

TOPOLOGIE EN **ARBRE**



(c'est une variante de la topologie en étoile)

TOPOLOGIE **BUS**



(le réseau se compose d'une ligne principale à laquelle toutes les unités sont connectées)

## Les réseaux locaux industriels

C'est une conception pour l'informatisation dans une industrie, cette conception se nomme **Pyramide CIM** (Computer Integrated Manufacturing). Elle explique le fait qu'il y'a 4 parties qui définiront cette industrie. L'**Usine** (la gestion de l'entreprise), l'**Atelier** (le contrôle de la production), les **Machines** (l'automatisation) et les **I/O** (les données qui transitent). On remarque qu'il y'a une divergence d'exécution dans le temps des différentes tâches en fonction de l'endroit du système. Une tâche prendra nettement moins de temps à se faire sur le terrain qu'à l'usine elle-même. Cela s'explique par le fait qu'au niveau de la gestion de l'entreprise, les actes ne doivent pas se faire à la milliseconde, contrairement au terrain où une instruction doit être effectuée en très peu de temps.

## La connectique

En fonction de la logistique utilisée, la connectique sera différente. Les connecteurs sont définis en 3 catégories définissant leur brochage (le nombre de points). On retrouve la **connectique à 9 points**, le **RJ45** ainsi que l'**Open Style**. La connectique est la partie physique de la liaison entre le système et les machines.

## Caractéristique des bus

Le bus le plus répandu est le BUS ASI (Actual Sensor Interface), il est simple d'utilisation, permet de réduire le coût des câblages, il est rapide à mettre en service et facile à intégrer dans les architectures d'automatismes.

## Transmission d'un octet

La transmission d'un octet ne consiste pas seulement à envoyer 8 bits de données mais d'encadrer cette information avec un bit de START et un bit de STOP pour indiquer le début et la fin de l'information. Le **bit de parité** pourra aider à déterminer si le message est erroné ou pas en vérifiant si la parité est respectée. On travaille avec deux types de parité. Le **PAIRE** qui vérifiera que le nombre de bit de donnée à 1 soit impair et la parité **IMPAIR** qui vérifiera si le nombre de bit de donnée soit impair. Il est important de noter que si deux bits sont erronés, le bit de parité ne détectera aucune erreur.