Rappels et Définitions

1 Généralités : Système informatique, l'information, le traitement.

2 Architecture d'un micro-ordinateur

1 Introduction au Microprocesseur 6809

Partie matérielle : **Architectures et fonctionnement**

1.1 Architecture externe : signaux et bus

1.2 Architecture interne : registres internes

1.3 Fonctionnement en régime établi

Partie Logiciel **Jeu d'instructions**

1.4 Définitions du jeu d'instruction

1.5 Définitions du code mnémonique

1.6 Définitions de L'adressage

Rappel et Définitions

1 Généralités

1.1 Système informatique

Un système informatique est une machine qui **traite** une **information**

1.2 L'information

L'information : est une grandeur qui doit être ***numérisée*** et ***codée*** pour pouvoir être traitée par la machine informatique

L'information : signe, message ou ensemble de connaissances à caractère **immatériel**. Ne pas confondre information et sa représentation, son support et son codage.

1.3 Le traitement

Exécution d'un programme constitué d'une suite organisée d'*instructions*. Ce programme peut être écrit dans des langages divers, *symboliques* ou non, et de différents *niveaux*

Exemple d'un programme qui additionne la valeur 1 à une *variable* en mémoire, sauvegarde le résultat en mémoire et l'affiche

2 Architecture d'un micro-ordinateur

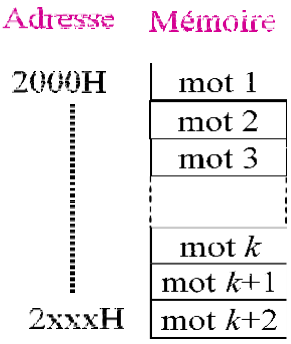
2.1 Généralités :

Seul, le micro-processeur ne peut fonctionner. Il faut aussi :

- de la mémoire pour stocker le programme à exécuter + les résultats intermédiaires
- des dispositifs d'entrée fournissant l'info à traiter (donnée)
- des dispositifs de sortie pour sortir les commandes vers l'extérieur du système

2.2 La mémoire

Organisée en mots (octets) de format identique, repérés par leur adresse.

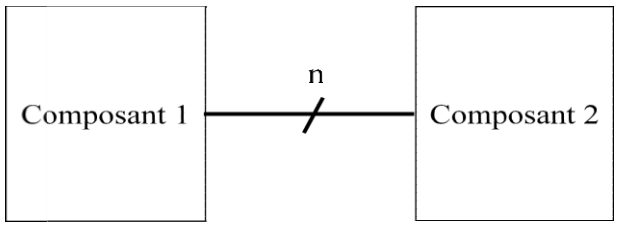


. **ROM** (*Read Only Memory*) programmée par le fabriquant (**PROM** programmable une fois par l'utilisateur, **EPROM** reprogrammable)

. **RAM** (*Random Access Memory*) lecture-écriture mais volatile, stockage des données provisoires

2.3 Les Bus

Le regroupement de *n* lignes (fils) permettant l'envoi en parallèle d'un mot de *n* bits entre deux composants 1 et 2

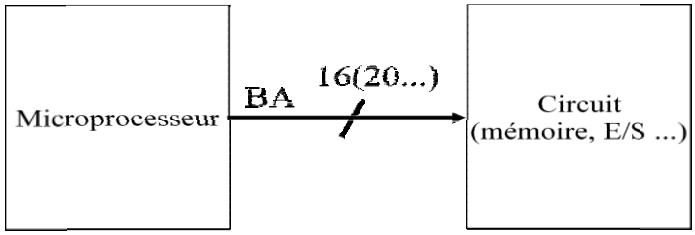


. Bus d'Adresse

Relie le microprocesseur à tout circuit adressable (mémoire, interfaces d'entrée/sortie ...),il est unidirectionnel.

Ex : - un microprocesseur 8 bits comporte 16 bits d'adresse 65536 adresses

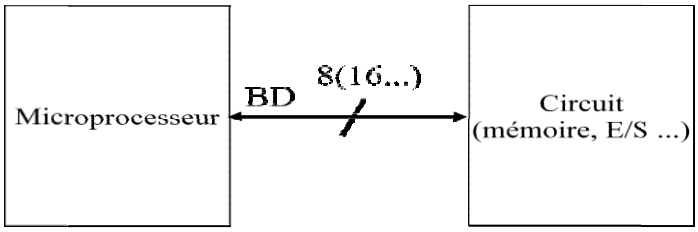
- un microprocesseur 16 bits comporte 20 bits d'adresse 1024000 adresses



. Bus de Donnée

Transmet les données entre le microprocesseur et le circuit adres. Il est bidirectionnel

Ex : Un système 8 bits (16, 32,64 ...) possède un BD 8 bits (16, 32, 64 ...) = taille de l'info que le processeur peut traiter en une opération élémentaire

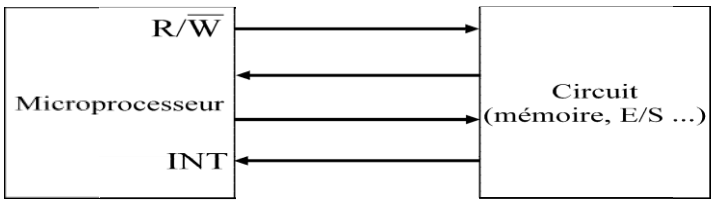


. Bus de Controle

Ensemble de lignes transmettant des signaux permettant le fonctionnement du microprocesseur, des circuits mémoire, des circuit d'interface ...

Exemple : **R/W** (µ-processeur circuit) : sens de transmission du bus de donnée

INT (circuit µ-processeur) : interruption



2.4 Horloge

Fournit des signaux périodiques de 1 a X000 MHz destinés à séquencer le travail du µprocesseur

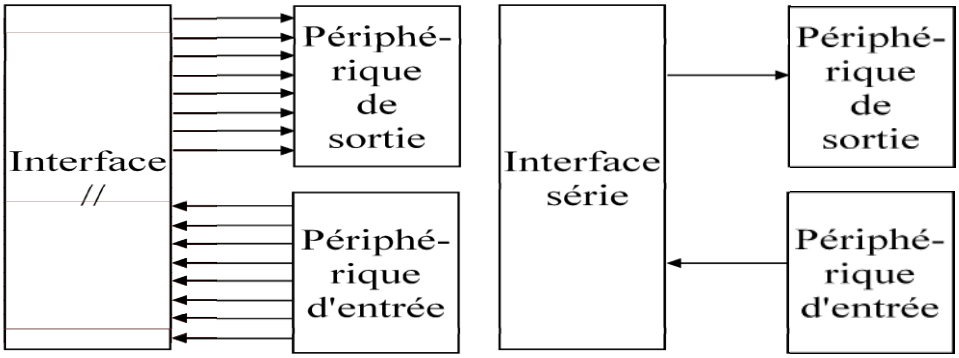
définit les cycles de base

Exemple : écrire la donnée 26H à l'adresse 3540H

- le µprocesseur compose la valeur 3540H sur le bus d 'adresse
- met la ligne R/W à 0
- émet la valeur 26H sur le bus de donnée

2.5 Les Circuits d'interface d'Entrée- Sorties

Permettent au µprocesseur de dialoguer avec l'extérieur.



Interface parallèle :
- rapide
- convenable sur quelques mètres

Interface série :
- lent
- convenable sur environ 25 m

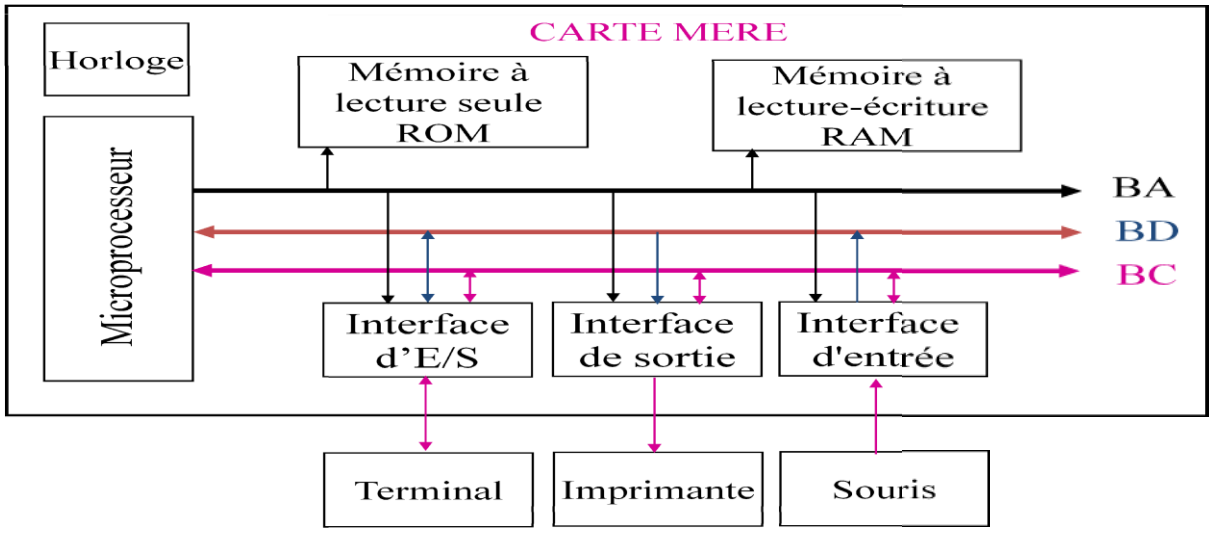


Figure 2.1 Structure général d'un calculateur

1 Introduction au Microprocesseur (MP 6809)

Le microprocesseur c'est Quoi ?

Le microprocesseur est une conséquence de la miniaturisation des circuits intégrés c'est un circuit LSI (Large Scale Integration), c'est à dire résultant d'une intégration à très grande échelle de plusieurs milliers de transistors sur un carré de quelques millimètres de côté et appelé « puce» (wafer en anglais). La technologie MOS, qui se prête à une forte densité d'intégration, a permis d'intégrer sur une seule puce, l'unité centrale d'un ordinateur, appelée aussi processeur d'où le nom donné à ce nouveau composant : MICROPROCESSEUR.

Le microprocesseur 6809 est un processeur 8 bits dont l'organisation interne est orientée 16 bits. Il est fabriqué en technologie **MOS** (***Metal Oxyde Semiconductor***)canal N et se présente sous la forme d'un boîtier **DIL** (***Dual In Line***) 40 broches. Il est monotension(5V). Il existe deux versions différenciées par l'horloge.

1.1 Partie Matériel : Architectures et Fonctionnement

1.1.1 Architecture externe

Quels sont les signaux usuels du micropresseur ?

Bien entendu ces signaux porteront des appellations différentes d'un microprocesseur à un autre, mais ils se retrouveront presque toujours. Les quatre signaux de lecture et d'écriture seront souvent codés sur deux bits pour laisser disponible le maximum de broches. L'entrée RESET initialise le microprocesseur, c'est à dire remet généralement à zéro le contenu des registres ; le compteur ordinal, quant à lui, se trouve généralement chargé à une valeur fixée par le constructeur du microprocesseur, par exemple 0000 en hexadécimal pour le 8080 A et le 8085 A, de Intel.

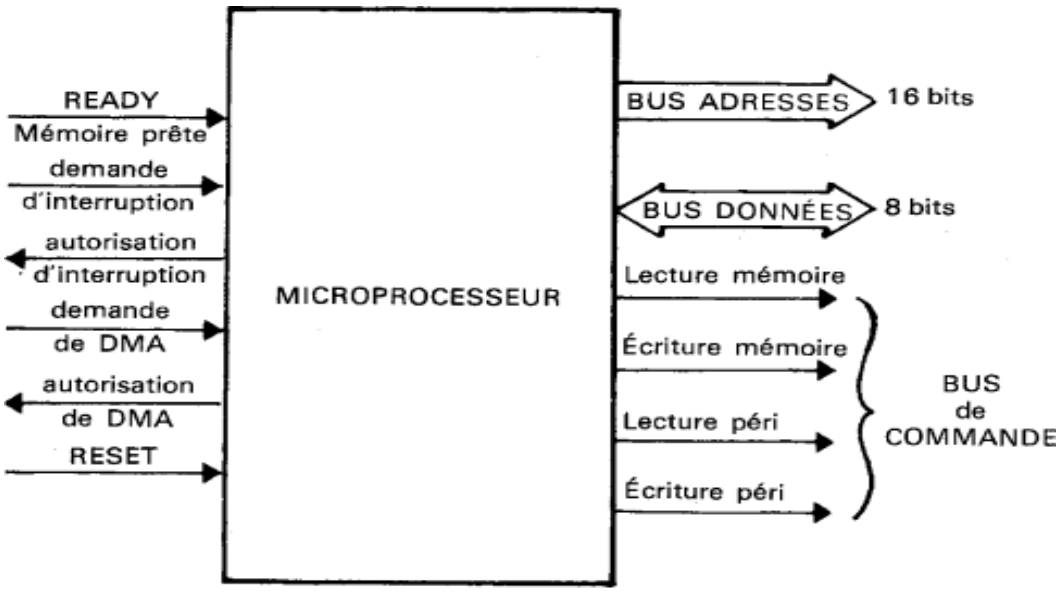
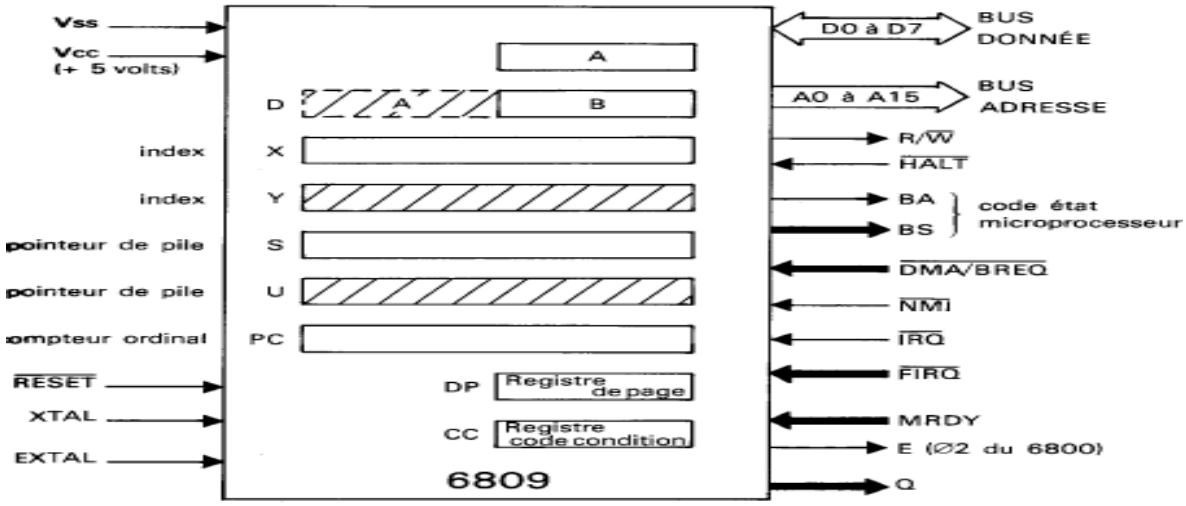
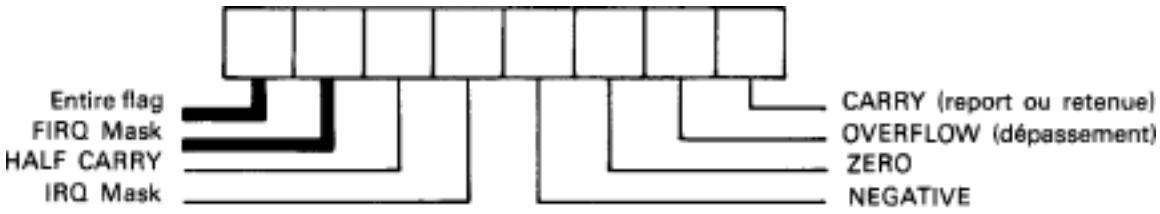


Fig. 3.2 : Les signaux usuels d'un microprocesseur



a) *Registres et signaux du 6809*

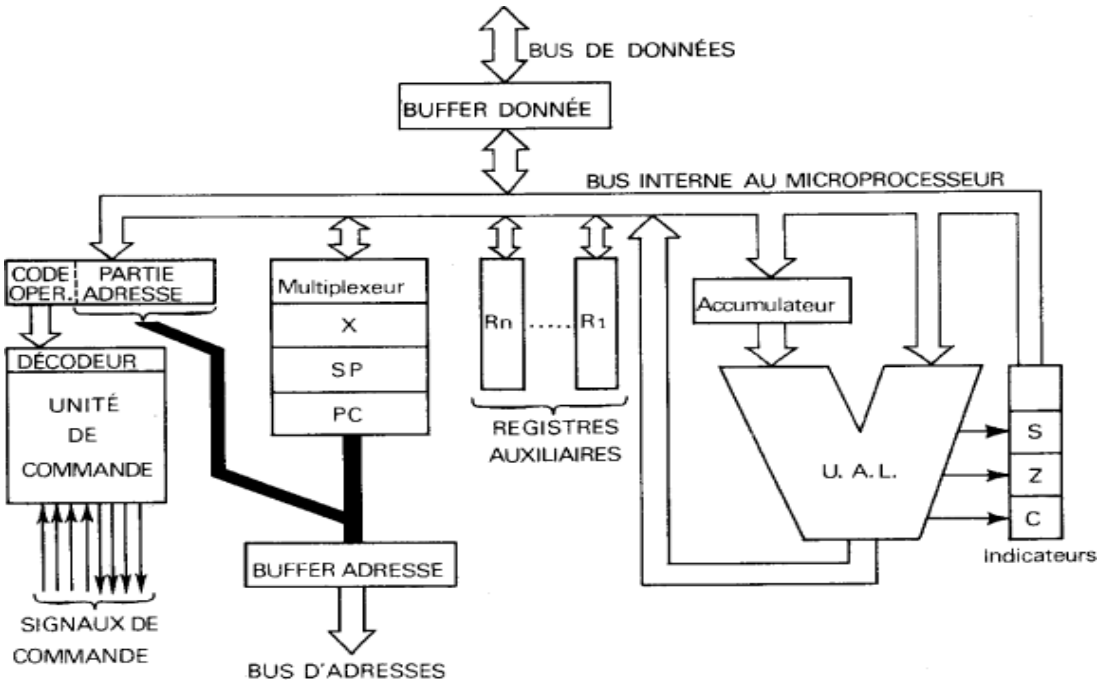


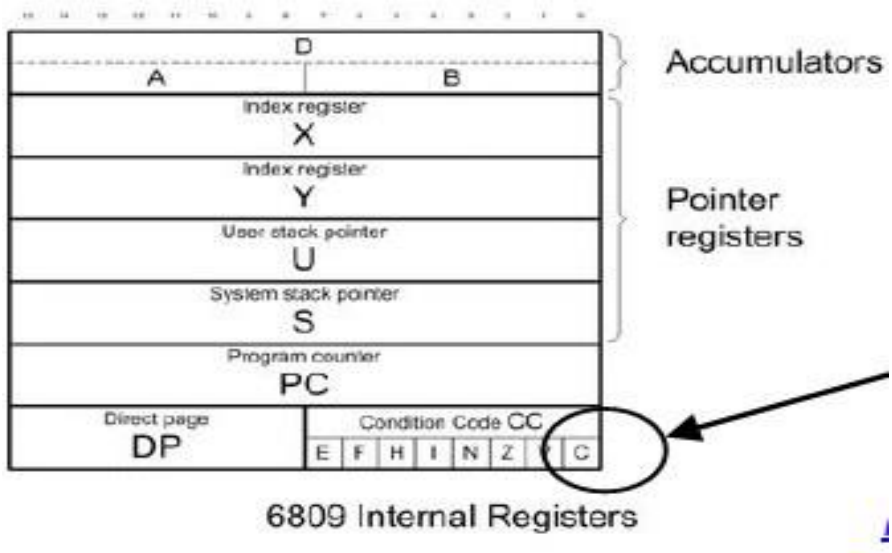
b) *Registre de conditions du 6809*

Figure 2.3 : Les registres internes du 6809

1.1.2 Structure interne

L'architecture standard d'un microprocesseur comprend, comme le montre la figure, une unité arithmétiue et logiue U.A.L, une unité de controle U.C et des registres. Les registres sont en fait des mémoires temporaires ui ont une tache bien précise, ils sont utilisés pour le stockage provisoire d'informations. Certains de ces registres sont accessibles à l'utilisateur par la programmation.





1.2.1 L'accumulateur

Ce registre de 8 bits

- servira pour toutes les opérations arithmétiques et la plupart des opérations logiques. Il contiendra un opérande au début de l'opération puis le résultat à la fin de l'opération ;
- recevra les données en provenance de l'extérieur, ces données étant ensuite reprises de l'accumulateur pour être rangées en mémoire ;
- recevra en général les données en provenance de la mémoire et destinées à être dirigées vers un périphérique. Autrement dit, tout transfert E/S nécessitera un transfert Accumulateur/Port du périphérique concerné ;
- pourra servir de compteur ou de décompteur car il peut être incrémenté ou décrémenté par des instructions.

Certains microprocesseurs comme le 6800 et le 6502 ont deux accumulateurs.

1.1.2.2 Le registre de conditions ou d'état

Ce registre rassemble des bits d'état ou indicateurs, qui sont de précieuses informations résultant de l'exécution d'une instruction. L'utilisation de ces indicateurs relève du logiciel ; aussi nous les étudierons avec le logiciel d'un microprocesseur.

1.1.2.3 Le compteur ordinal

Le compteur ordinal contient l'adresse de la prochaine instruction à exécuter. Son contenu est donc envoyé au registre adresse de la mémoire par l'intermédiaire du bus adresses. C'est un compteur car les instructions sont normalement rangées dans leur ordre d'exécution. L'exécution d'une instruction incrémente automatiquement le compteur ordinal désigné dans la littérature anglo-saxonne par *program counter* et en abréviation P.C. Toutefois, il est possible, dans certains cas, d'aller se brancher directement à la première adresse d'un sous-programme ou à une adresse bien définie (cas d'une boucle). Pour ce faire, la nouvelle adresse devra être chargée dans le compteur ordinal, soit par programme, soit automatiquement selon le type d'instruction et le type de microprocesseur.

Le compteur ordinal contient l'adresse de la prochaine instruction à exécuter. Son contenu est donc envoyé au registre adresse de la mémoire par l'intermédiaire du bus adresses. C'est un compteur car les instructions sont normalement rangées dans leur ordre d'exécution. L'exécution d'une instruction incrémente automatiquement le compteur ordinal désigné dans la littérature anglo-saxonne par *program counter* et en abréviation P.C. Toutefois, il est possible, dans certains cas, d'aller se brancher directement à la première adresse d'un sous-programme ou à une adresse bien définie (cas d'une boucle). Pour ce faire, la nouvelle adresse devra être chargée dans le compteur ordinal, soit par programme, soit automatiquement selon le type d'instruction et le type de microprocesseur.

1.1.2.4 Le registre instrunction

Toutes les instructions du programme sont en mémoire RAM ou ROM. Le processeur ira donc chercher l'instruction en mémoire pour la stocker dans un registre à partir duquel les instructions seront décodées (c'est le cycle recherche nécessaire pour toute instruction). Ce registre est appelé *registre instruction*. Le décodage permettra de reconnaître une addition, une soustraction, un décalage, etc., et des signaux de commande seront activés en conséquence pour permettre l'exécution de l'instruction. Si besoin est, le microprocesseur retournera dans la mémoire pour chercher un opérande.

1.1.2.5 Le pointeur de pile

La pile est une petite partie de la mémoire RAM utilisée pour sauvegarder les contenus des registres lorsque cela est nécessaire, par exemple lors d'un appel de sous programme ou d'une interruption. L'adresse de la prochaine position mémoire à utiliser pour

la sauvegarde de données est à tout moment fournie par le contenu d'un registre appelé « pointeur de pile > (Stack Pointer en anglais). Nous étudierons en détail le fonctionnement de la pile quand nous aborderons le logiciel.

1.1.2.6 Le registre Index

Ce registre permet l'adressage indexé. Il est destiné à recevoir une adresse aussi c'est un registre de 16 bits. Certains microprocesseurs peuvent avoir plusieurs index.

1.1.3 Fonctionnement en régime établi

Le microprocesseur réalise 2 opérations: • Lecture du code opératoire (le code de l'instruction à exécuter)

• Exécution de l'instruction

C'est le pointeur d'instruction qui repère l'instruction à exécuter. Le pointeur d'instruction est dans un registre qui contient l'adresse de la prochaine instruction à exécuter. Le microprocesseur réalise donc la gestion du pointeur d'instruction. Pour le µP 6809, la société MOTOROLA appelle ce registre le compteur ordinal ou le compteur de programme (P.C.). Fonctionnement détaillé de la recherche puis de l'exécution d'une instruction (le compteur ordinal est initialisé à l'adresse de l'instruction à exécuter).

- Le µP positionne sur son bus des adresses à l'adresse de l'instruction à exécuter.
- Le µP lit le premier octet de l'instruction à exécuter (une instruction peut être composée d'un, deux ou trois octets.
- Le µP décode le premier octet de l'instruction; à partir de ce moment le µP sait ce qu'il doit faire.
- Le µP incrémente le compteur ordinal d'un, deux ou trois. (Cette opération n'est possible qu'une fois que le µP a lu et décodé le premier octet de l'instruction; cet octet contient la taille de l'instruction: un, deux ou trois octets.
- Le µP lit éventuellement les deuxième et troisième octets de l'instruction à exécuter.
- Le µP exécute l'instruction.

1.1.4 Initialisation du compteur ordinal à la mise sous tension A la mise sous tension

Le compteur ordinal doit être initialisé pour pointer la première instruction du programme. La séquence de démarrage est la suivante:

- Positionnement sur le bus des adresses de l'adresse FFFE
- Lecture de la donnée présente à l'adresse FFFE puis transfert de cette donnée (8 bits) sur l'octet de poids fort du compteur ordinal (P.C.).
- Lecture de la donnée présente à l'adresse FFFF puis transfert de cette donnée (8 bits) sur l'octet de poids faible du compteur ordinal (P.C.).
- Recherche de la première instruction à exécuter.

1.2 Partie Logicielle

1.2.1 Jeu d'instructions

En lagage évolué symboliue de haut niveau (C, pascal, Fortran)

- var1 = var2+1 *en C ou Fortran*
- var1 := var2+1 *en Pascal*
- printf("Résultat = %d",var1) *en C*
- writeln('Résultat = ',var1) *en Pascal*
- write (1,200) var1

Chaque ligne représente une *instruction* dans un langage dit *évolué* ou *haut niveau* et qui est un langage *symbolique* proche de l'anglais

Chaque ligne représente une *instruction* dans un langage dit *évolué* ou *haut niveau* et qui est un langage *symbolique* proche de l'anglais

- **A chaque instruction en langage d'assemblage correspond une instruction en langage machine** (=opération élémentaire du microprocesseur). Il en résulte que :
  - l'assembleur n'est jamais un logiciel très encombrant
  - le langage d 'assemblage est le langage symbolique le plus proche de la machine
- **Pour qu'un programme écrit en langage évolué devienne exécutable, il faut le traduire dans le langage machine du microprocesseur** du système informatique utilise a son exécution (*compilateur*)
- Un langage évolué est un langage puissant : une instruction en langage évolué sera traduite par le compilateur à l'aide de plusieurs instructions machine → un compilateur est un logiciel de taille plus importante qu'un assembleur

Un microprocesseur est capable d'exécuter des instructions (en langage machine/microcode) :

- arithmétiques : +, -, *, parfois /
- logiques : ET, OU, OU exclusif, négation, les décalages, les rotations
- de prises de décision en fonction du résultat d'un traitement précédent

Le jeu d'instructions qui en résulte est donc restreint et pourtant il permet de tout faire

1.2.2 L'adressage

- L'adressage registre

  Il n'y a ni opérande, ni adresse.

  Exemple : MOV A, B soit transfert du contenu de B dans A.

- L'adressage immédiat

  Il est défini par la dernière lettre du code instruction ; soit I.

  Exemple : MVI A, 08 qui signifie initialiser A à 08.

- L'adressage direct

  L'adresse est exprimée sur 16 bits dans l'instruction.

  Exemple : LDA 1000 H qui signifie lire la position-mémoire d'adresse 1000 en hexadécimal et transférer son contenu dans l'accumulateur.

• L'adressage indirect par registre

Les registres B, C, D, E, H et L sont organisés pour pouvoir être associés par deux, ce qui crée trois registres de 16 bits, appelés « paires de registres » : BC, DE et HL. Les deux premières paires BC et DE sont utilisées comme registres d'adressage indirect pour deux instructions :

- une instruction de lecture LDAX rp dans laquelle rp, qui signifie « register pair », désigne la paire de registres concernée, et qui est désignée par son premier registre, soit B ou D. Exemple : LDAX B qui signifie charger dans l'accumulateur le contenu de la position-mémoire dont l'adresse est le contenu de BC,

- une instruction d'écriture STAX rp.

Exemple : STAX D qui signifie écrire le contenu de l'accumulateur dans la position-mémoire dont l'adresse est le contenu de DE.

La paire HL est prévue pour remplacer le registre index qui n'existe pas. Elle joue le rôle d'un index avec un déplacement toujours nul. Toutes les instructions utilisant la paire HL pour registre d'adressage indirect sont reconnaissables par la lettre M dans le code instruction, cette lettre symbolisant une position-mémoire.

Exemple : MOV A, M qui signifie transférer dans l'accumulateur A, le contenu de la position-mémoire dont l'adresse est le contenu de HL, ce que nous représentons par (M) → A.

REMARQUE:

Pour les mots de 16 bits, l'octet poids faible est traité le premier ; il correspond au deuxième registre de la paire de registres.

Exemple : LHLD addr, qui signifie transférer dans la paire HL les contenus des position-mémoires d'adresses addr et addr + 1. Les transferts sont les suivants :