

Chapitre 1 : Architecture d'un microprocesseur

I.1 Introduction aux systèmes à base d'un μ Processeur

Le système à μ Processeur est constitué :

- De l'Unité Centrale de Traitement (UCT), en anglais « Central Processing Unit » (CPU). L'UCT est constituée de leur tour par :
 - D'une Unité Arithmétique et Logique (UAL), en anglais « Arithmetic and Logic Unit » (ALU) : c'est l'organe de calcul d'un système à μ Processeur.
 - De registres : zones de stockage de données de travail de l'UAL (opérandes, résultats intermédiaires) ;
 - D'une Unité de Contrôle (UC), en anglais « Control Unit » (CU) : elle envoie les ordres ou (commandes) à tous les autres éléments du système à μ Processeur afin d'exécuter un programme.
- D'une mémoire centrale qui contient :
 - Le programme à exécuter : suite d'instructions élémentaires.
 - Les données à traiter.
- D'une unité d'entrées/sorties (E/S) : est un intermédiaire entre le système à μ Processeur et le monde extérieur.
- L'unité de transfert : est le support matériel de la circulation des données. Les échanges d'ordres et de données dans le système à μ Processeur sont synchronisés par une horloge qui délivre des impulsions (signal d'horloge) à des intervalles de temps fixes.

Par définition le μ Processeur est l'unité centrale de traitement (UAL + registres + unité de contrôle) entièrement contenue dans un seul circuit intégré. **Un système à μ Processeur est un microcalculateur ou un micro-ordinateur.** La figure I.1 représente une structure d'un système à μ Processeur.

Remarque : un circuit intégré qui inclut une UCT, de la mémoire et des périphériques est un microcontrôleur.

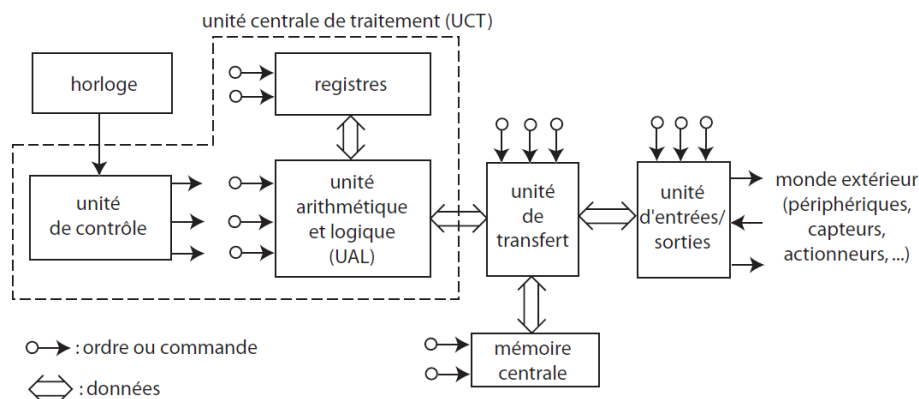


Fig. I.1 structure d'un système à μ Processeur

I.1.1 Circulation de l'information dans un système à μ Processeur

La réalisation matérielle des ordinateurs est généralement basée sur l'architecture de Von Neumann (figure I.2)

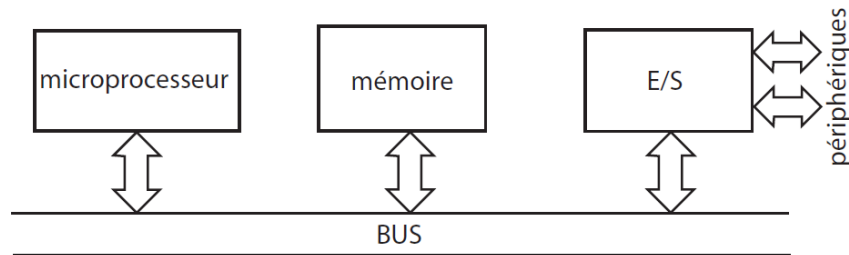


Fig. I.2 Architecture de Von Neumann

Le microprocesseur échange des informations avec la mémoire et l'unité d'E/S, sous forme de mots binaires, au moyen d'un ensemble de connexions appelé bus. Un bus permet de transférer des données sous forme parallèle, c'est-à-dire en faisant circuler n bits simultanément.

Les microprocesseurs peuvent être classés selon la longueur maximale des mots binaires qu'ils peuvent échanger avec la mémoire et les E/S : microprocesseurs 8 bits, 16 bits, 32 bits, ...

Le bus peut être décomposé en trois bus distincts :

- le bus d'adresses : permet au microprocesseur de spécifier l'adresse de la case mémoire à lire ou à écrire.
- le bus de données : permet les transferts entre le microprocesseur et la mémoire ou les E/S.
- le bus de commande transmet les ordres de lecture et d'écriture de la mémoire et des E/S. La figure I.3 illustre le système les différents bus d'un système à μ Processeur

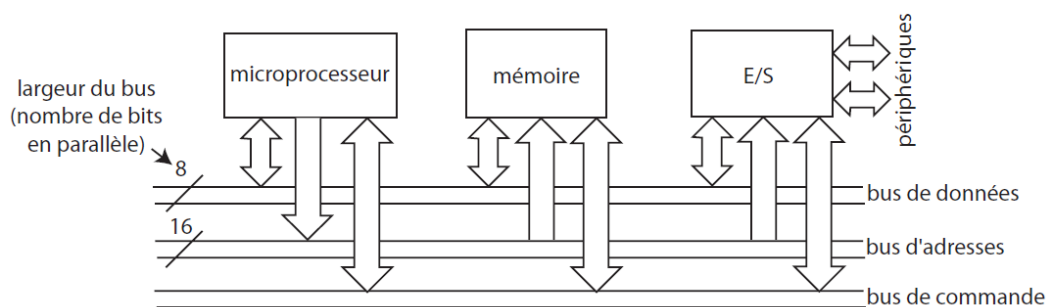


Fig. I.3 différents bus d'un système à μ Processeur

Remarque : les bus de données et de commande sont bidirectionnels, le bus d'adresse est unidirectionnel : seul le microprocesseur peut délivrer des adresses (il existe une dérogation pour les circuits d'accès direct à la mémoire, DMA).

I.1.2 Description matérielle d'un μ Processeur

Un μ processeur se présente sous la forme d'un circuit intégré muni d'un nombre généralement important de broches. Exemples :

- Intel 8085, 8086, Zilog Z80 : 40 broches, DIP (Dual In-line Package) ;
- Motorola 68000 : 64 broches, DIP ;
- Intel 80386: 196 broches, PGA (Pin Grid Array).

Technologies de fabrication : NMOS, PMOS, CMOS.

On peut représenter un μ processeur par son schéma fonctionnel (figure I.4)

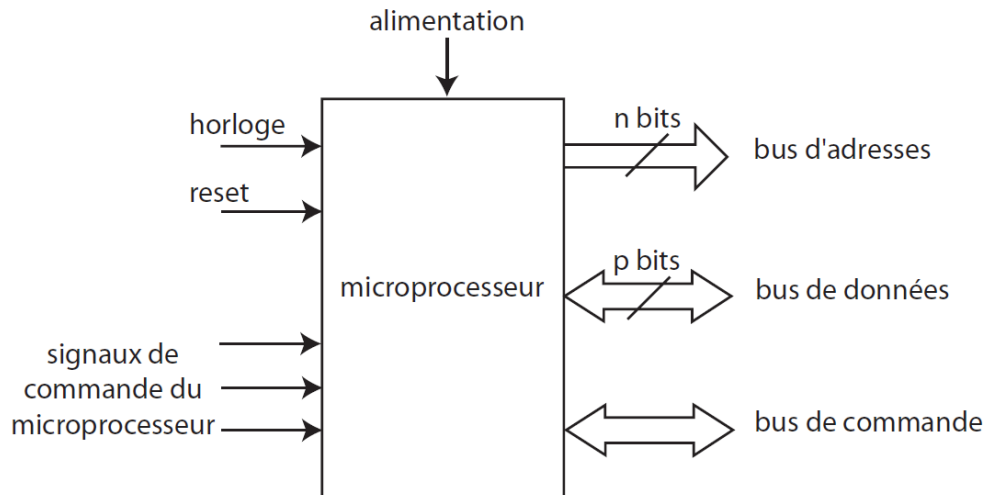
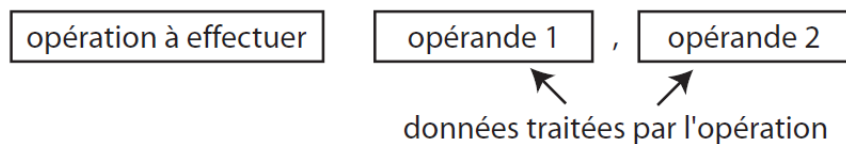


Fig. I.4 schéma fonctionnel d'un μ processeur

I.1.2 Fonctionnement d'un μ Processeur

- Un microprocesseur exécute un programme. Le programme est une suite d'instructions stockées dans la mémoire. Une instruction peut être codée sur un ou plusieurs octets. Format d'une instruction est donné par :



Exemple :

Additionner case mémoire1, case mémoire2
opération *opérandes*

La figure I.5 représente le rangement d'un programme (suite d'instructions) dans une mémoire.

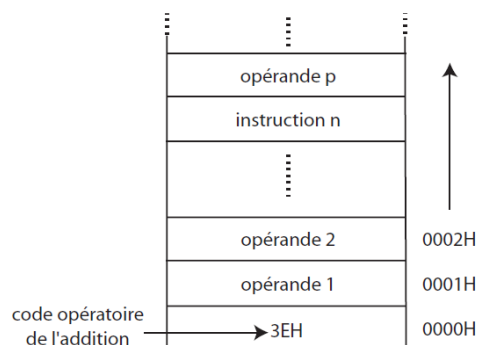


Fig. I.5 suite d'instructions stockée dans la mémoire

- Pour exécuter les instructions dans l'ordre établi par le programme, le microprocesseur doit savoir à chaque instant l'adresse de la prochaine instruction à exécuter. Le microprocesseur utilise un registre contenant cette information. Ce registre est appelé pointeur d'instruction (IP : Instruction Pointer) ou compteur d'instructions ou compteur ordinal (voir figure I.6)

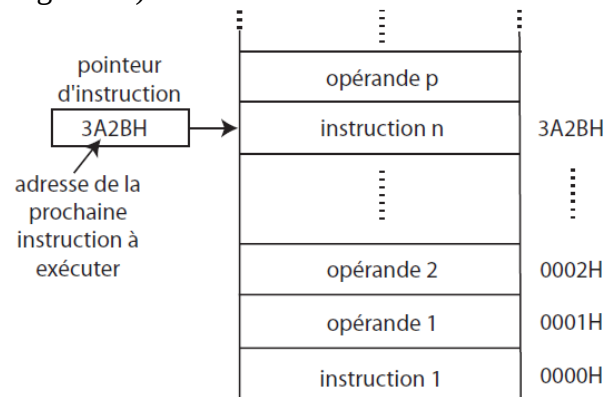


Fig. I.5 contenu du registre pointeur d'instruction

- la valeur initiale du pointeur d'instruction est fixée par le constructeur du microprocesseur. Elle vaut une valeur bien définie à chaque mise sous tension du microprocesseur ou bien lors d'une remise à zéro (reset).
- Pour savoir quel type d'opération doit être exécuté (addition, soustraction, ...), le microprocesseur lit le premier octet de l'instruction pointée par le pointeur d'instruction (code opératoire) et le range dans un registre appelé registre d'instruction. Le code opératoire est décodé par des circuits de décodage contenus dans le microprocesseur. Des signaux de commande pour l'UAL sont produits en fonction de l'opération demandée qui est alors exécutée.
- Pendant que l'instruction est décodée, le pointeur d'instruction est incrémenté de façon à pointer vers l'instruction suivante, puis le processus de lecture et de décodage des instructions recommence.
- A la suite de chaque instruction, un registre du microprocesseur est actualisé en fonction du dernier résultat : c'est le registre d'état du microprocesseur. Chacun des bits du registre d'état est un indicateur d'état ou flag (drapeau). La figure I.6 représente le registre d'état du μ processeur Z80.

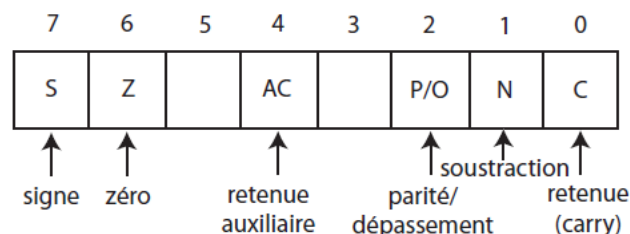


Fig. I.6 le registre d'état du μ processeur Z80.

- Toutes les étapes (lecture de l'instruction, décodage, exécution) sont synchronisées par un séquenceur qui assure le bon déroulement des opérations, ce dernier exécute lui-même un programme appelé microcode, contenu dans une mémoire morte à l'intérieur du microprocesseur.

- Le séquenceur est dirigé par une horloge qui délivre un signal de fréquence donnée permettant d'enchaîner les différentes étapes de l'exécution d'une instruction. Chaque instruction est caractérisée par le nombre de périodes d'horloge (ou microcycles) qu'elle utilise (donnée fournie par le fabricant du microprocesseur). La figure I.7 représente le signal d'horloge nécessaire l'exécution d'une instruction.

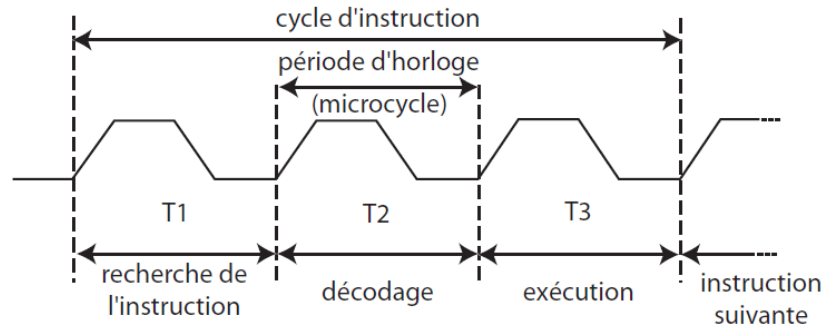


Fig. I.7 le signal d'horloge.

- Structure complète d'un microprocesseur simple : pour fonctionner, un microprocesseur nécessite donc au minimum les éléments suivants (figure I.8)

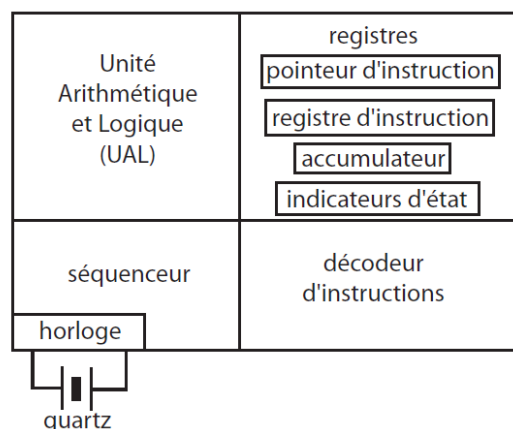


Fig. I.8 structure simple d'un μ Processeur

I.2 Architecture externe d'un μ Processeur (Intel 8086)

Le μ Processeur Intel 8086 est un microprocesseur 16 bits, apparu en 1978. C'est le premier microprocesseur de la famille Intel 80x86 (8086, 80186, 80286, 80386, 80486, Pentium, ...). Il se présente sous la forme d'un boîtier DIP (Dual In-line Package) à 40 broches (voir figure I.9)

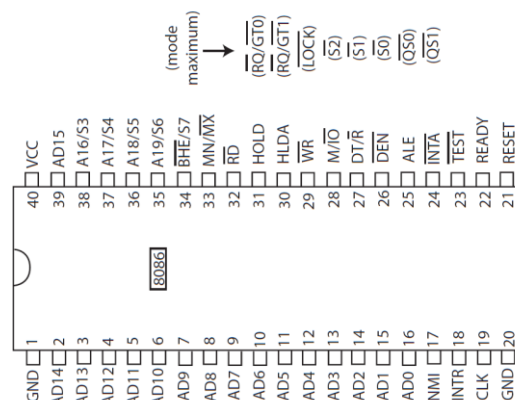


Fig. I.9 boîtier DIP du μ Processeur Intel 80x86

I.2.1 Description et utilisation des signaux 80x86

- **CLK** : entrée du signal d'horloge qui cadence le fonctionnement du microprocesseur. Ce signal provient d'un générateur d'horloge : c'est le circuit 8224 (voir figure I.10)

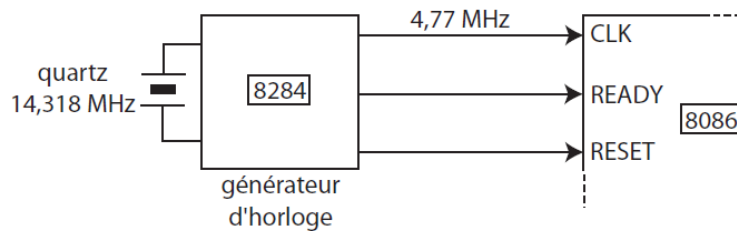


Fig. I.10 circuit générateur d'horloge : le circuit 8284

- **RESET** : entrée de remise à zéro du microprocesseur. Lorsque cette entrée est mise à l'état haut pendant au moins quatre périodes d'horloge, le microprocesseur est réinitialisé : il va exécuter l'instruction se trouvant à l'adresse FFFF0H (adresse de bootstrap). Le signal de RESET est fourni par le générateur d'horloge.
- **READY** : entrée de synchronisation avec la mémoire. Ce signal provient également du générateur d'horloge.
- **TEST** : entrée de mise en attente du microprocesseur d'un événement extérieur.
- **MN/ $\overline{\text{MX}}$** : entrée de choix du mode de fonctionnement du microprocesseur :
 - Mode minimum ($\text{MN}/\overline{\text{MX}} = 1$) : le 80x86 fonctionne de manière autonome, il génère lui-même le bus de commande ($\overline{\text{RD}}$, $\overline{\text{WR}}$, ...) ;
 - Mode maximum ($\text{MN}/\overline{\text{MX}} = 0$) : ces signaux de commande sont produits par un contrôleur de bus, le circuit 8288. Ce mode permet de réaliser des systèmes multiprocesseurs.
- **NMI et INTR** : entrées de demande d'interruption. INTR : interruption normale, NMI (Non Maskable Interrupt) : interruption prioritaire.
- **$\overline{\text{INTA}}$** : Interrupt Acknowledge, indique que le microprocesseur accepte l'interruption.
- **HOLD et HLDA** : signaux de demande d'accord d'accès direct à la mémoire (DMA).
- **S₀ à S₇** : signaux d'état indiquant le type d'opération en cours sur le bus.
- **A₁₆/S₃ à A₁₉/S₆** : 4 bits de poids fort du bus d'adresses, multiplexés avec 4 bits d'état.
- **AD₀ à AD₁₅** : 16 bits de poids faible du bus d'adresses, multiplexés avec 16 bits de données. Le bus A/D est multiplexé (multiplexage temporel) d'où la nécessité d'un démultiplexage pour obtenir séparément les bus d'adresses et de données :
 - 16 bits de données (microprocesseur 16 bits) ;
 - 20 bits d'adresses, d'où $2^{20} = 1\text{M}_0$ d'espace mémoire adressable par le 8086.

La figure I.11 illustre le chronogramme du bus **Adress/Data**

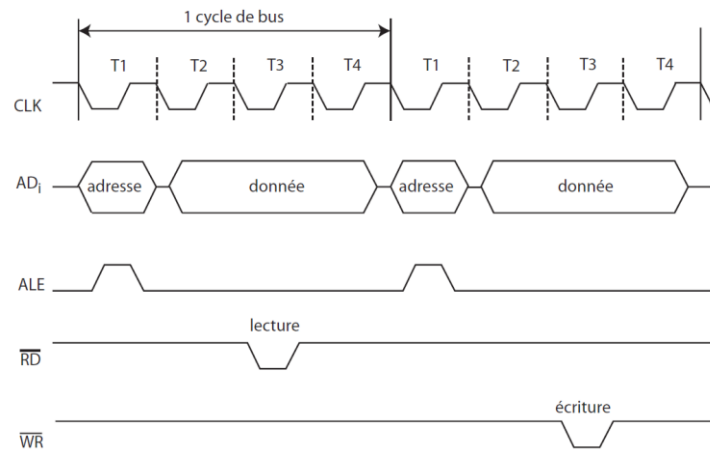


Fig. I.11 chronogramme du bus A/D

Le démultiplexage des signaux AD_0 à AD_{15} (ou A_{16}/S_3 à A_{19}/S_6) se fait en mémorisant l'adresse lorsque celle-ci est présente sur le bus A/D, à l'aide d'un verrou (latch), ensemble de bascules D. La commande de mémorisation de l'adresse est générée par le microprocesseur : c'est le signal ALE, Address Latch Enable. Circuit de démultiplexage A/D est donné par la figure (I.12).

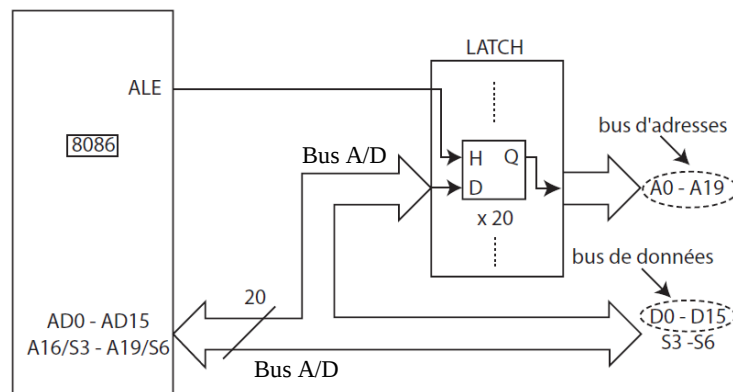


Fig. I.12 circuit de démultiplexage A/D

Le fonctionnement du circuit verrou (latch) :

- si $ALE = 1$, le verrou est transparent ($Q = D$) ;
- si $ALE = 0$, mémorisation de la dernière valeur de D sur les sorties Q ;
- les signaux de lecture ($\overline{RD}$) ou d'écriture ($\overline{WR}$) ne sont générés par le microprocesseur que lorsque les données sont présentes sur le bus A/D.

- $\overline{RD}$: Read, signal de lecture d'une donnée.
- $\overline{WR}$: Write, signal d'écriture d'une donnée.
- $M/\overline{IO}$: Memory/Input-Output, indique si le 8086 adresse la mémoire ($M/IO = 1$) ou les entrées/sorties ($M/IO = 0$).
- $\overline{DEN}$: Data Enable, indique que des données sont en train de circuler sur le bus A/D (équivalent de ALE pour les données).
- $DT/\overline{R}$: Data Transmit/Receive, indique le sens de transfert des données :

- $DT/\overline{R} = 1$: données émises par le microprocesseur (écriture) ;
- $DT/\overline{R} = 0$: données reçues par le microprocesseur (lecture).
- **$\overline{BHE}$** : Bus High Enable, signal de lecture de l'octet de poids fort du bus de données. Le 8086 possède un bus d'adresses sur 20 bits, d'où la capacité d'adressage de 1 Mo ou 512 Kmots de 16 bits (bus de données sur 16 bits). Le méga-octet adressable est divisé en deux banques de 512 Ko chacune : la banque inférieure (ou paire) et la banque supérieure (ou impaire). Ces deux banques sont sélectionnées par :
 - A_0 pour la banque paire qui contient les octets de poids faible ;
 - $\overline{BHE}$ pour la banque impaire qui contient les octets de poids fort.

Seuls les bits A_{19} à A_1 servent à désigner une case mémoire dans chaque banque de 512 Ko. Le microprocesseur peut ainsi lire et écrire des données sur 8 bits ou sur 16 bits suivant le tableau :

$\overline{BHE}$	A_0	octets transférés
0	0	les deux octets (mot complet)
0	1	octet fort (adresse impaire)
1	0	octet faible (adresse paire)
1	1	aucun octet

I.3 Architecture interne d'un μ Processeur (Intel 8086)

Le 8086 est constitué de deux unités fonctionnant en parallèle :

- l'unité d'exécution (EU : Execution Unit) : exécute les instructions contenues dans la file d'attente.
- l'unité d'interface de bus (BIU : Bus Interface Unit) : recherche les instructions en mémoire et les range dans une file d'attente.

Le microprocesseur 8086 contient 14 registres répartis en 4 groupes :

- **Registres généraux** : 4 registres sur 16 bits.

$AX = (AH, AL)$, $BX = (BH, BL)$, $CX = (CH, CL)$ et $DX = (DH, DL)$. Ils peuvent être également considérés comme 8 registres sur 8 bits. Ils servent à contenir temporairement des données. Ce sont des registres généraux mais ils peuvent être utilisés pour des opérations particulières.

- **Registres de pointeurs** : 2 registres sur 16 bits.

SP : Stack Pointer, pointeur de pile (la pile est une zone de sauvegarde de données en cours d'exécution d'un programme).

BP : Base Pointer, pointeur de base, utilisé pour adresser des données sur la pile.

- **Registres d'index** : 2 registres sur 16 bits.

SI : Source Index.

DI : Destination Index.

Ils sont utilisés pour les transferts de chaînes d'octets entre deux zones mémoire.

- **Pointeur d'instruction** : IP, contient l'adresse de la prochaine instruction à exécuter.
- **Indicateurs (flags)** : indique le signe, la parité ou si le résultat après les opérations arithmétiques et logiques est nulle.

				O	D	I	T	S	Z		A		P		C
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

CF : indicateur de retenue (carry), PF : indicateur de parité, AF : indicateur de retenue auxiliaire, ZF : indicateur de zéro, SF : indicateur de signe, TF : indicateur d'exécution pas à pas (trap), IF : indicateur d'autorisation d'interruption, DF : indicateur de décrémentation, OF : indicateur de dépassement (overflow).

- **Registres de segments** : 4 registres sur 16 bits.

CS : Code Segment, registre de segment de code, DS : Data Segment, registre de segment de données, SS : Stack Segment, registre de segment de pile, ES : Extra Segment, registre de segment supplémentaire pour les données, la figure I.13 représente architecture interne d'un microprocesseur 8086.

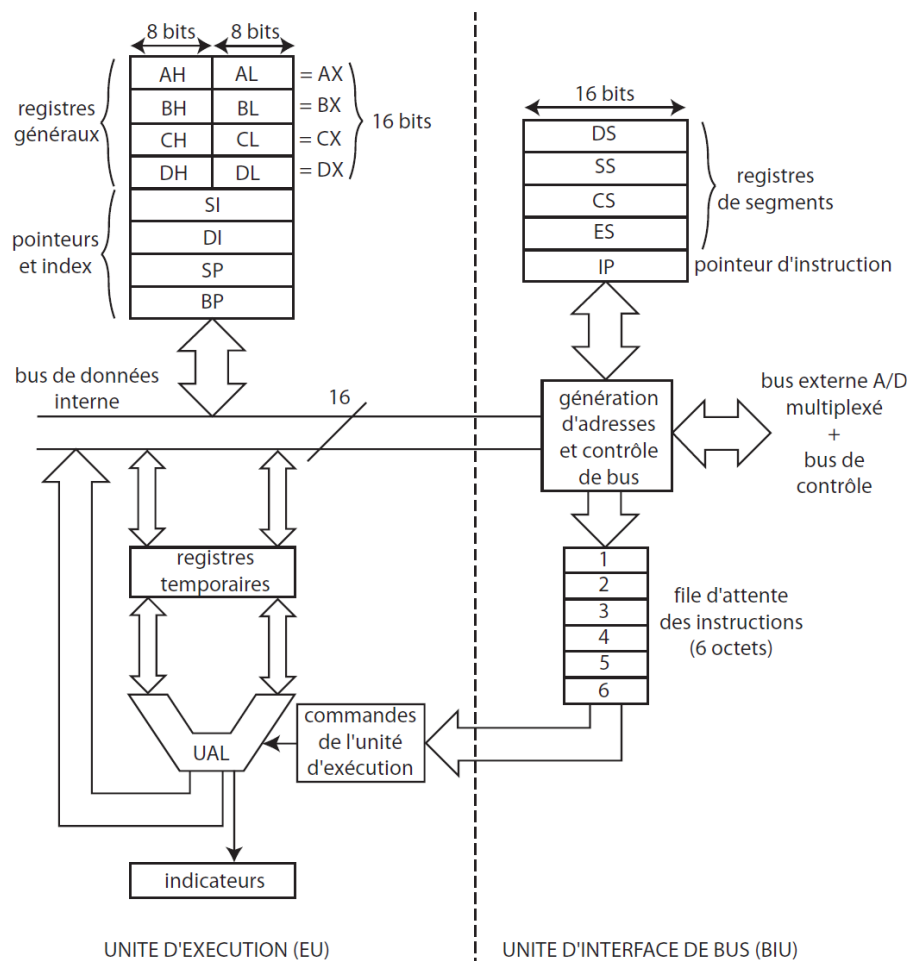


Fig. I.13 Architecture interne du microprocesseur 8086