

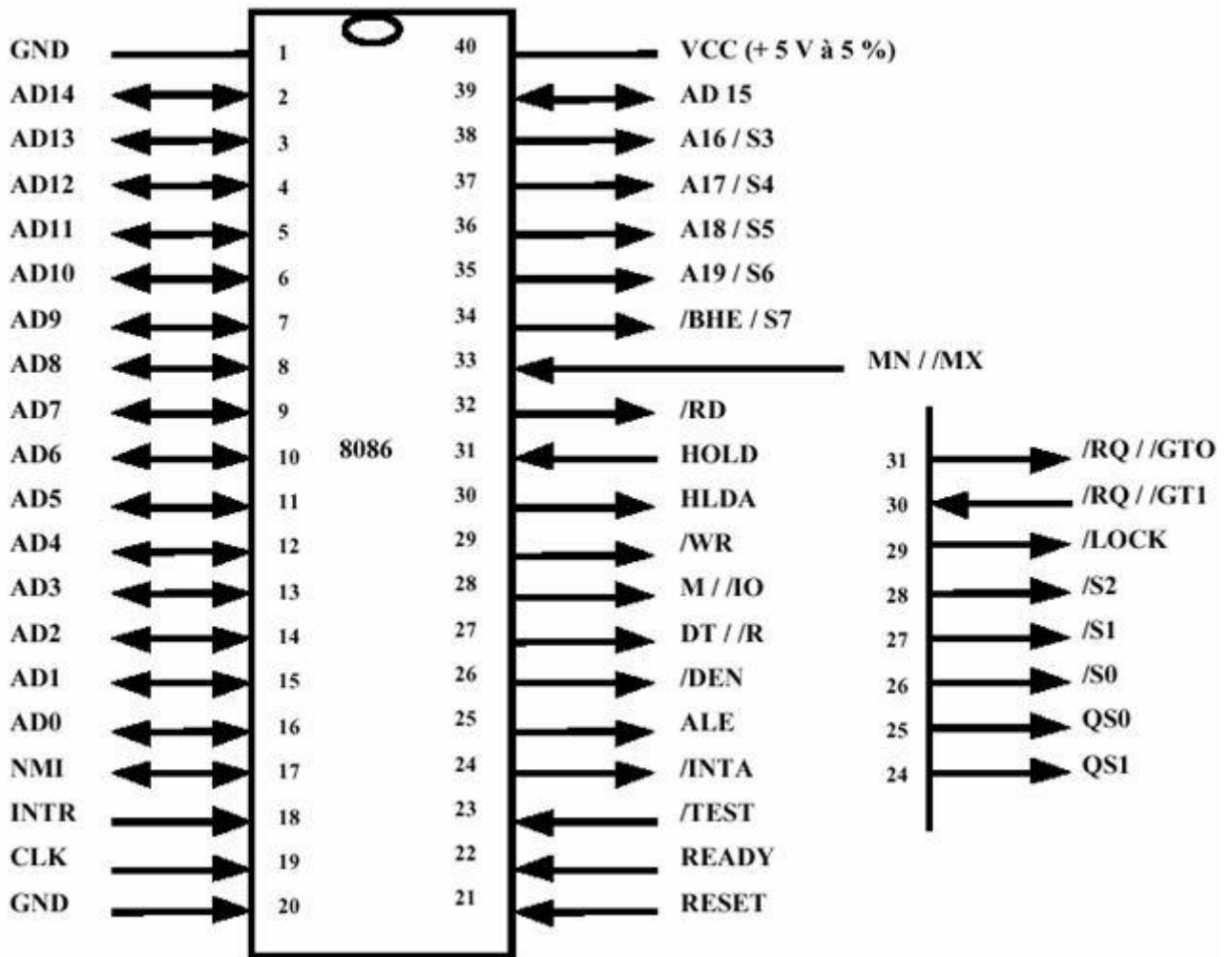
Le microprocesseur 8086 / 8088

I) Introduction :

Le processeur 8086 d'Intel est à la base des processeurs Pentium actuels. Les processeurs successifs (de PC) se sont en effet construits petit à petit en ajoutant à chaque processeurs des instructions et des fonctionnalités supplémentaires, mais en conservant à chaque fois les spécificités du processeur précédent. C'est cette façon d'adapter les processeurs à chaque étape qui permet qu'un ancien programme écrit pour un 8086 fonctionne toujours sur un nouvel ordinateur équipé d'un Pentium IV.

II) Architecture externe du 8086 :

Le 8086 est un circuit intégré de forme DIL de 40 pattes comme le montre la figure suivante :



Le 8086 (développé en 1978) est le premier microprocesseur de type x86. Il est équipé d'un bus de données de 16 bits et un bus d'adresses de 20 bits et fonctionne à des fréquences diverses selon plusieurs variantes: 5, 8 ou 10 MHz.

III) Architecture interne du 8086 :

Il existe deux unités internes distinctes: l'UE (Unité d'Exécution) et l'UIB (Unité d'Interfaçage avec le Bus). Le rôle de l'UIB est de récupérer et stocker les informations à traiter, et d'établir les transmissions avec les bus du système. L'UE exécute les instructions qui lui sont transmises par l'UIB. L'image ci-dessous résume les notions présentées ici. Le microprocesseur pris comme exemple est le 8086/8088. Les processeurs actuels de la famille x86 traitent les informations de la même façon.

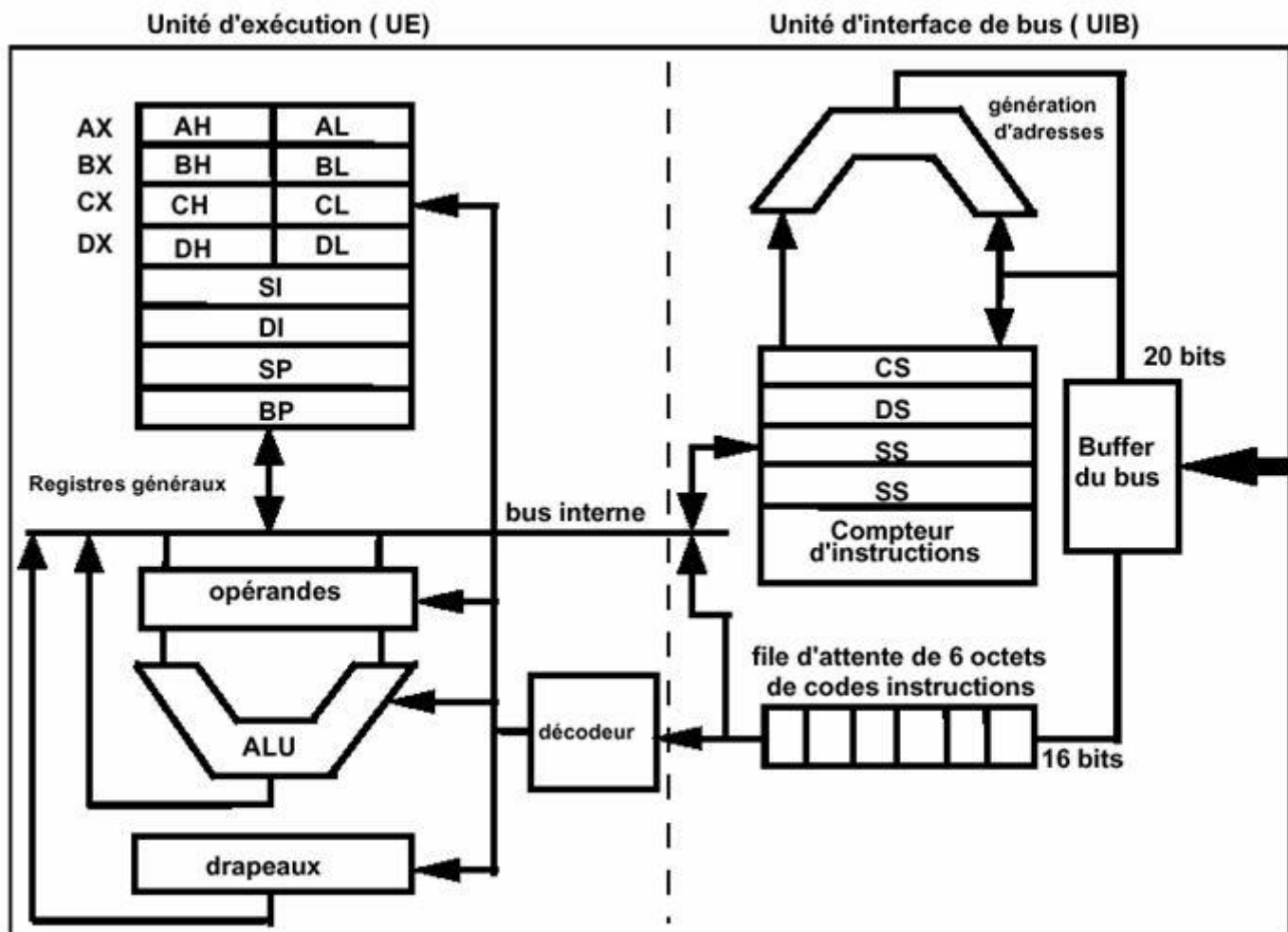
Nous pouvons à présent examiner plus en détail le traitement des instructions par l'UE et l'UIB. Avec le microprocesseur 8085, le traitement des instructions se passait comme suit:

-Extraction des instructions par l'UIB

- Exécution des instructions

- Extraction des nouvelles instructions

Lorsque l'exécution d'une instruction est terminée, l'UE reste inactif un court instant, pendant que l'UIB extrait l'instruction suivante. Pour remédier à ce temps d'attente, le *prétraitement* ou *traitement pipeline* a été introduit dans le 8086/8088. Pendant que l'UE exécute les informations qui lui sont transmises, l'instruction suivante est chargée dans l'UIB. Les instructions qui suivront sont placées dans une file d'attente. Lorsque l'UE a fini de traiter une instruction l'UIB lui transmet instantanément l'instruction suivante, et charge la troisième instruction en vue de la transmettre à l'UE. De cette façon, l'UE est continuellement en activité. Dans la figure suivante nous pouvons observer un schéma plus détaillé de l'UE et l'UIB. Nous y retrouvons les éléments dont il a été question précédemment



Donc en conclusion on peut dire que le 8086/8088 se compose essentiellement de deux unités : la BIU qui fournit l'interface physique entre le microprocesseur et le monde extérieur et l'EU qui comporte essentiellement l'UAL de 16 bits qui manipule les registre généraux de 16 bits aussi .

Remarque :

La file d'attente d'instructions contient des informations qui attendent d'être traitées par l'UE. La file d'attente est parfois appelée *capacité de traitement*. Le microprocesseur 8086 est capable de mémoriser jusqu'à six octets. Les microprocesseurs actuels sont bien entendu équipés d'une file d'attente plus rapide et plus large, c'est à dire capable d'emmagasiner plus d'informations.

IV) Les registres du 8086/8088 :

IV-1) Introduction :

Le jeu de registres contient l'ensemble des registres du microprocesseur. Un registre est une petite partie de mémoire intégrée au microprocesseur, dans le but de recevoir des informations spécifiques, notamment des adresses et des données stockées durant l'exécution d'un programme. Il existe plusieurs types

de registres. Certains d'entre eux sont affectés à des opérations d'ordre général et sont accessibles au programmeur à tout moment. Nous disons alors qu'il s'agit de registres généraux. D'autres registres ont des rôles bien plus spécifiques et ne peuvent pas servir à un usage non spécialisé.

IV-2) Les registres généraux :

Les registres généraux peuvent être utilisés dans toutes les opérations arithmétiques et logiques que le programmeur insère dans le code assembleur. Un registre complet présente une grandeur de 16 bits. Comme le montre la figure 2, chaque registre est en réalité divisé en deux registres distincts de 8 bits. De cette façon, nous pouvons utiliser une partie du registre si nous désirons y stocker une valeur n'excédant pas 8 bits. Si, au contraire, la valeur que nous désirons y ranger excède 8 bits, nous utiliserons le registre complet, c'est à dire 16 bits. Nous verrons plus loin qu'il est possible de manipuler facilement les registres généraux.

Le programmeur dispose de 8 registres internes de 16 bits qu'on peut diviser en deux groupes comme le montre la figure 2 :

- groupe de données : formé par 4 registres de 16 bits (AX,BX,CX,et

DX) chaque registre peut être divisé en deux registres de 8 bits

(AH,AL,BH,BL,CH,CL,DH et DL)

- groupe de pointeur et indexe : formé de 4 registres de 16 bits (SI, DI, SP, BP) et font généralement référence à un emplacement en mémoire.

Groupe de données :

	15	8 7	0
AX	AH		AL
BX	BH		BL
CX	CH		CL
DX	DH		DL

Groupe de pointeur et indexe :

	15	0
Stack pointer	SP	
Base pointer	BP	
Source index	SI	
Destination index	DI	

IV-2-1) Groupe de données :

Registre AX : (Accumulateur)

Toutes les opérations de transferts de données avec les entrées-sorties ainsi que le traitement des chaînes de caractères se font dans ce registre, de même les opérations arithmétiques et logiques.

Les conversions en BCD du résultat d'une opération arithmétique (addition, soustraction, multiplication et la division) se font dans ce registre.

Registre BX : (registre de base)

Il est utilisé pour l'adressage de données dans une zone mémoire différente de la zone code : en général il contient une adresse de décalage par rapport à une adresse de référence.). (Par exemple, l'adresse de début d'un tableau). De plus il peut servir pour la conversion d'un code à un autre.

Registre CX : (Le compteur)

Lors de l'exécution d'une boucle on a souvent recours à un compteur de boucles pour compter le nombre d'itérations, le registre CX a été fait pour servir comme compteur lors des instructions de boucle.

Remarque :

Le registre CL sert en tant que compteur pour les opérations de décalage et de rotation, dans ce cas il va compter le nombre de décalages (rotation) de bits à droite ou à gauche.

Registre DX :

On utilise le registre DX pour les opérations de multiplication et de division mais surtout pour contenir le numéro d'un port d'entrée/sortie pour adresser les interfaces d'E/S.

IV-2-2) Groupe de pointeur et indice :

Ces registres sont plus spécialement adaptés au traitement des éléments dans la mémoire. Ils sont en général munis de propriétés d'incrémentation et de décrémentation.

Un cas particulier de pointeur est le pointeur de pile (Stack Pointer : SP). Ce registre permet de pointer la pile pour stocker des données ou des adresses selon le principe du "Dernier Entré Premier Sorti" ou "LIFO"

(Last In First Out).

L'indexe SI : (source index) :

Il permet de pointer la mémoire il forme en général un décalage (un offset) par rapport à une base fixe (le registre DS), il sert aussi pour les instructions de chaîne de caractères, en effet il pointe sur le caractère source

L'indexe DI : (Destination index) :

Il permet aussi de pointer la mémoire il présente un décalage par rapport à une base fixe (DS ou ES), il sert aussi pour les instructions de chaîne de caractères, il pointe alors sur la destination

Les pointeurs SP et BP : (Stack pointer et base pointer)

Ils pointent sur la zone pile (une zone mémoire qui stocke l'information avec le principe filo : voir plus loin), ils présentent un décalage par rapport à la base (le registre SS). Pour le registre BP il a un rôle proche de celui de BX, mais il est généralement utilisé avec le segment de pile.

IV -2- 3) Les registres segment:

	15	0
Code segment	CS	
Data segment	DS	
Stack segment	SS	
Extra segment	ES	