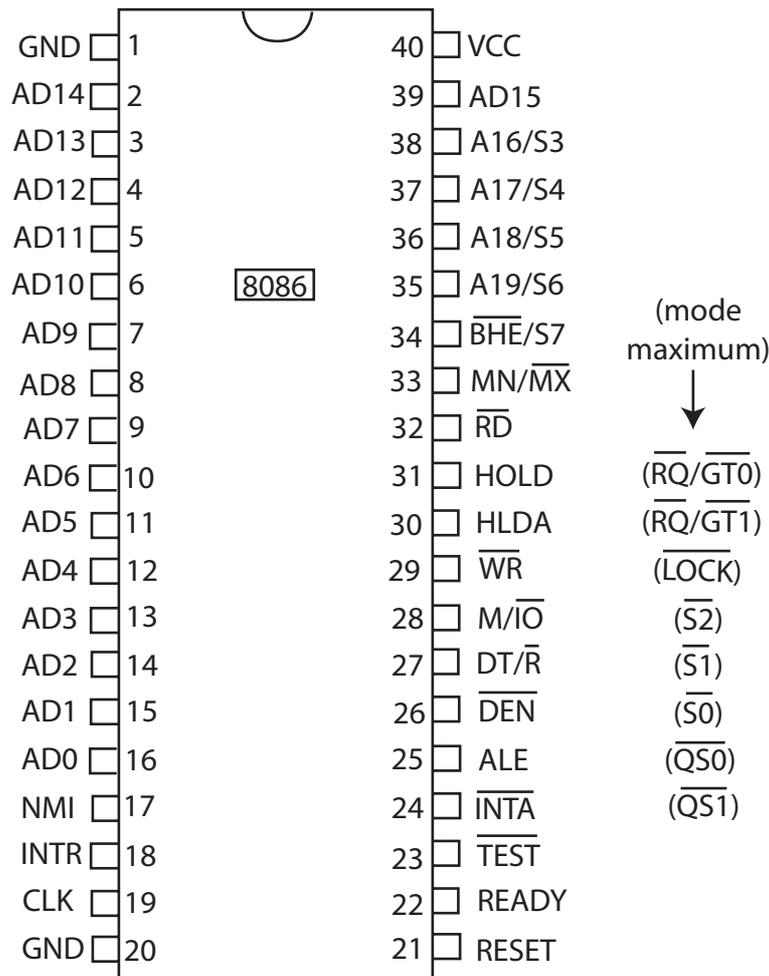


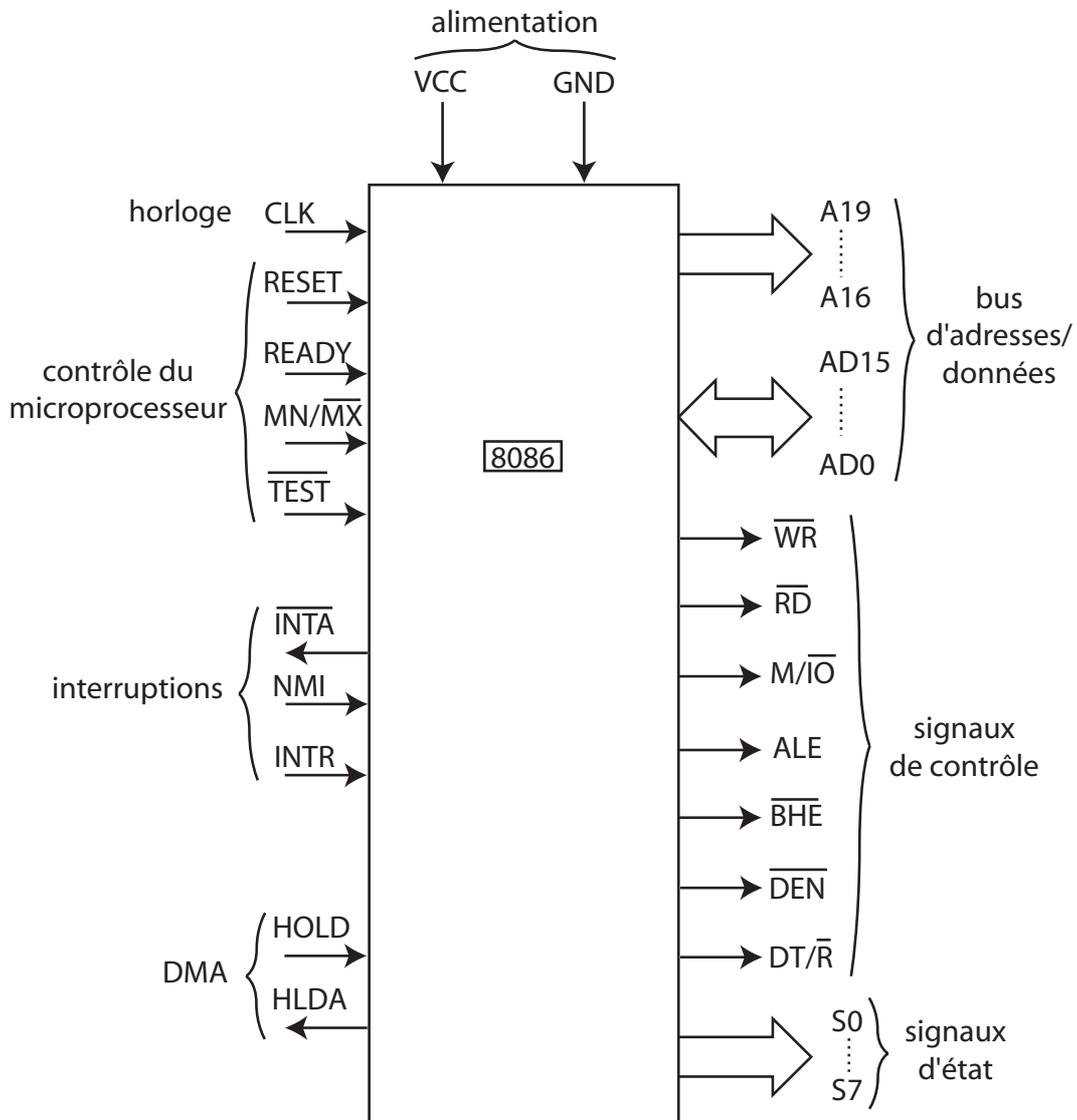
Le microprocesseur Intel 8086

4.1 Description physique du 8086

Le microprocesseur 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 :

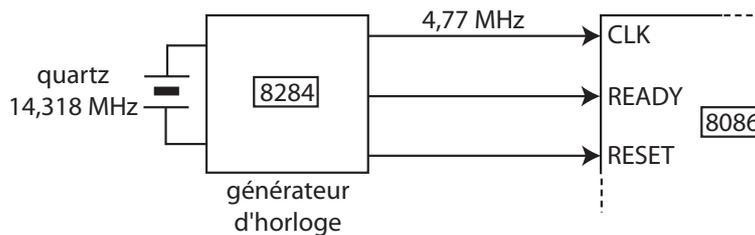


4.2 Schéma fonctionnel du 8086



4.3 Description et utilisation des signaux du 8086

CLK : entrée du signal d'horloge qui cadence le fonctionnement du microprocesseur. Ce signal provient d'un **générateur d'horloge** : le 8284.



RESET : entrée de remise à zéro du microprocesseur. Lorsque cette entrée est mise à l'état haut pendant au moins 4 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.

$\overline{\text{TEST}}$: entrée de mise en attente du microprocesseur d'un événement extérieur.

$\text{MN}/\overline{\text{MX}}$: entrée de choix du mode de fonctionnement du microprocesseur :

- mode minimum ($\text{MN}/\overline{\text{MX}} = 1$) : le 8086 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 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).

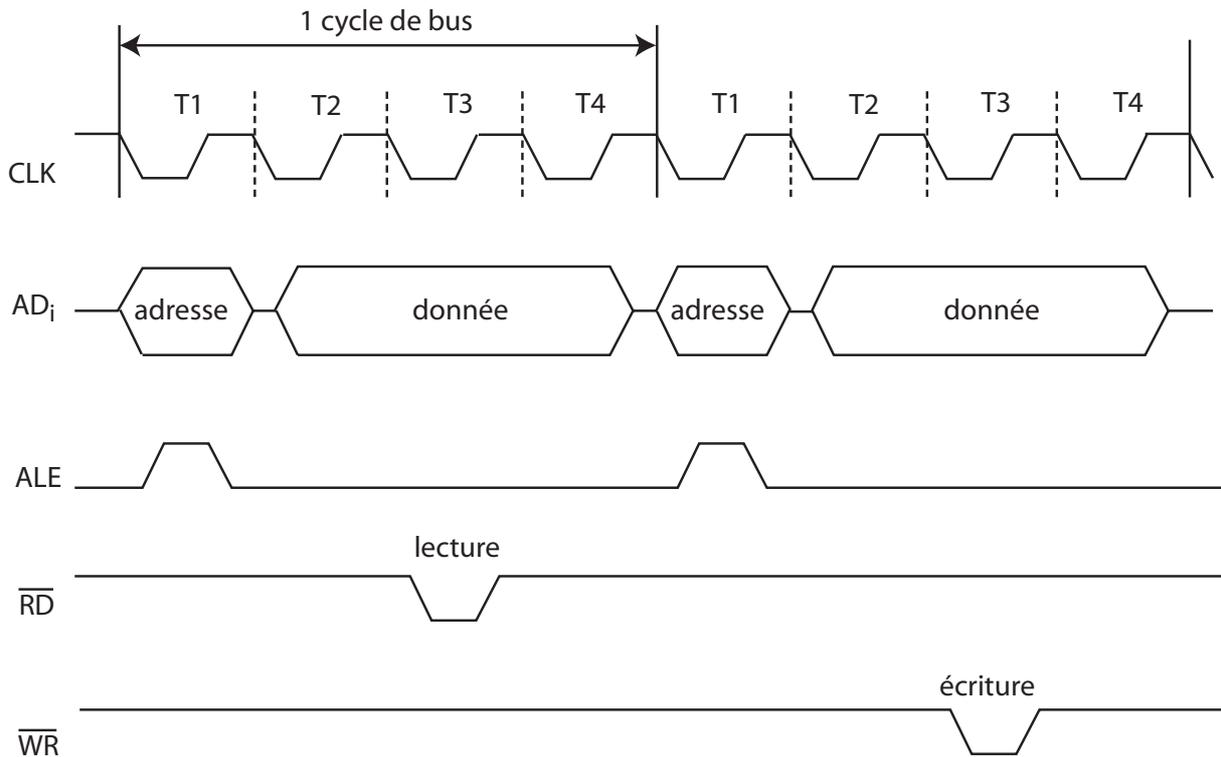
S0 à **S7** : signaux d'état indiquant le type d'opération en cours sur le bus.

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 :

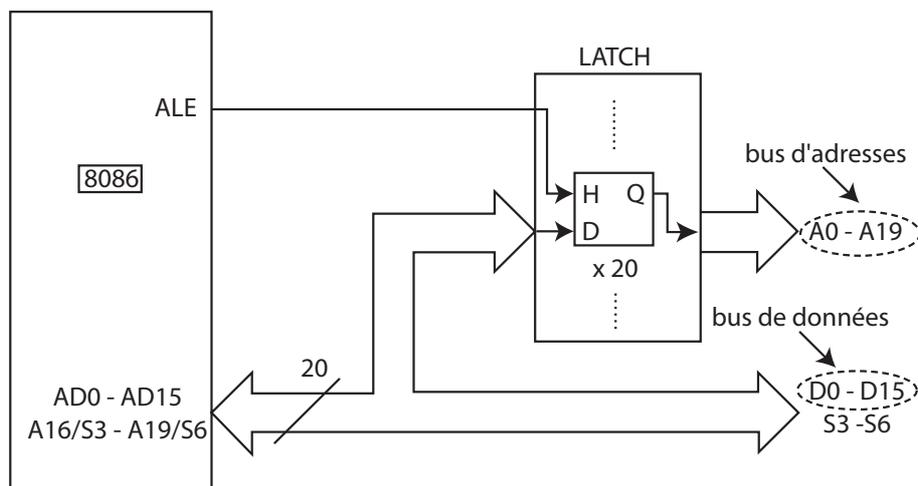
- 16 bits de données (microprocesseur 16 bits);
- 20 bits d'adresses, d'où $2^{20} = 1$ Mo d'espace mémoire adressable par le 8086.

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.

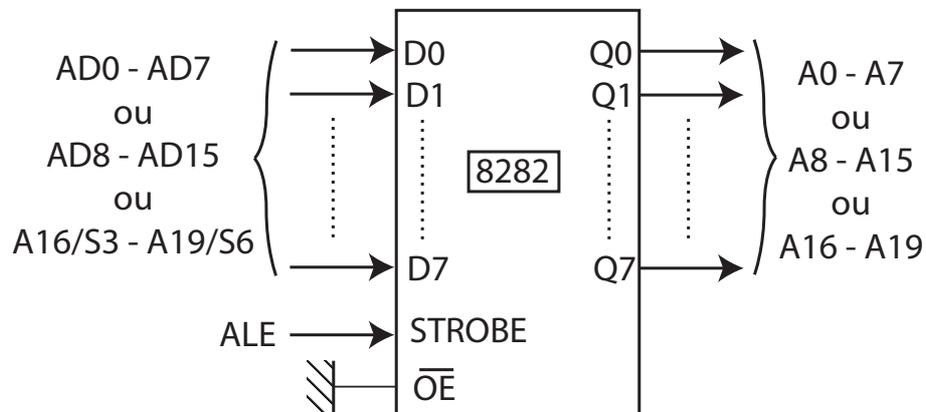
Circuit de démultiplexage A/D :



Fonctionnement :

- 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.

Exemples de bascules D : circuits 8282, 74373, 74573.



\overline{RD} : Read, signal de lecture d'une donnée.

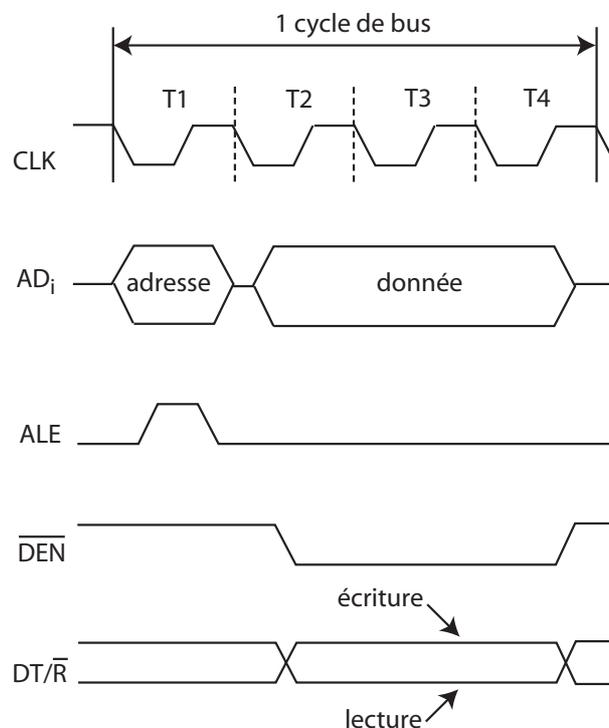
\overline{WR} : Write, signal d'écriture d'une donnée.

M/\overline{IO} : Memory/Input-Output, indique si le 8086 adresse la mémoire ($M/\overline{IO} = 1$) ou les entrées/sorties ($M/\overline{IO} = 0$).

\overline{DEN} : Data Enable, indique que des données sont en train de circuler sur le bus A/D (équivalent de ALE pour les données).

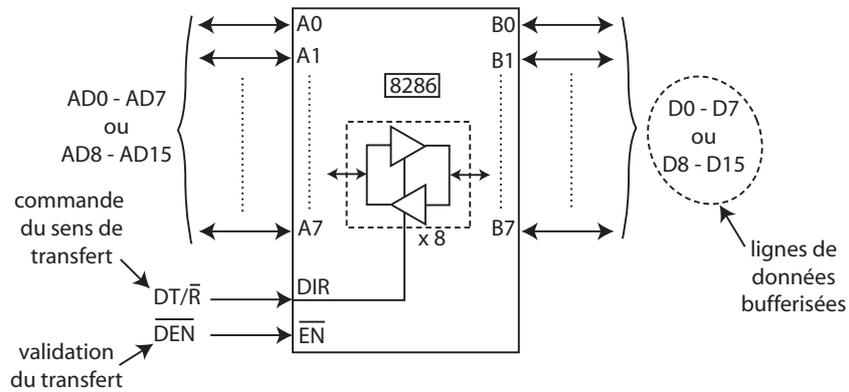
DT/\overline{R} : Data Transmit/Receive, indique le sens de transfert des données :

- $DT/\overline{R} = 1$: données émises par le microprocesseur (écriture) ;
- $DT/\overline{R} = 0$: données reçues par le microprocesseur (lecture).



Les signaux \overline{DEN} et DT/\overline{R} sont utilisés pour la commande de **tampons de bus** (buffers) permettant d'amplifier le courant fourni par le microprocesseur sur le bus de données.

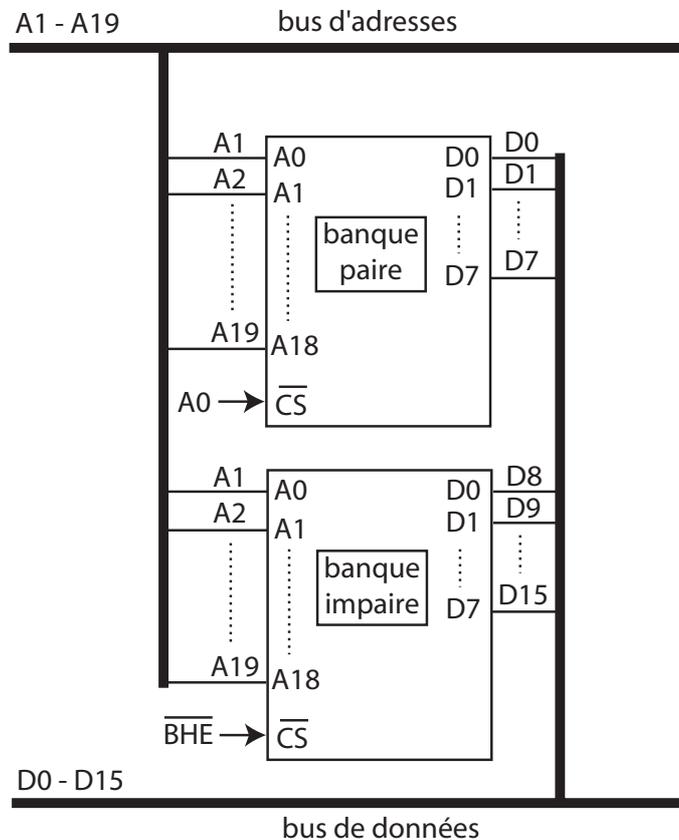
Exemples de tampons de bus : circuits transmetteurs bidirectionnels 8286 ou 74245.



\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 :

- A0 pour la banque paire qui contient les octets de poids faible ;
- \overline{BHE} pour la banque impaire qui contient les octets de poids fort.

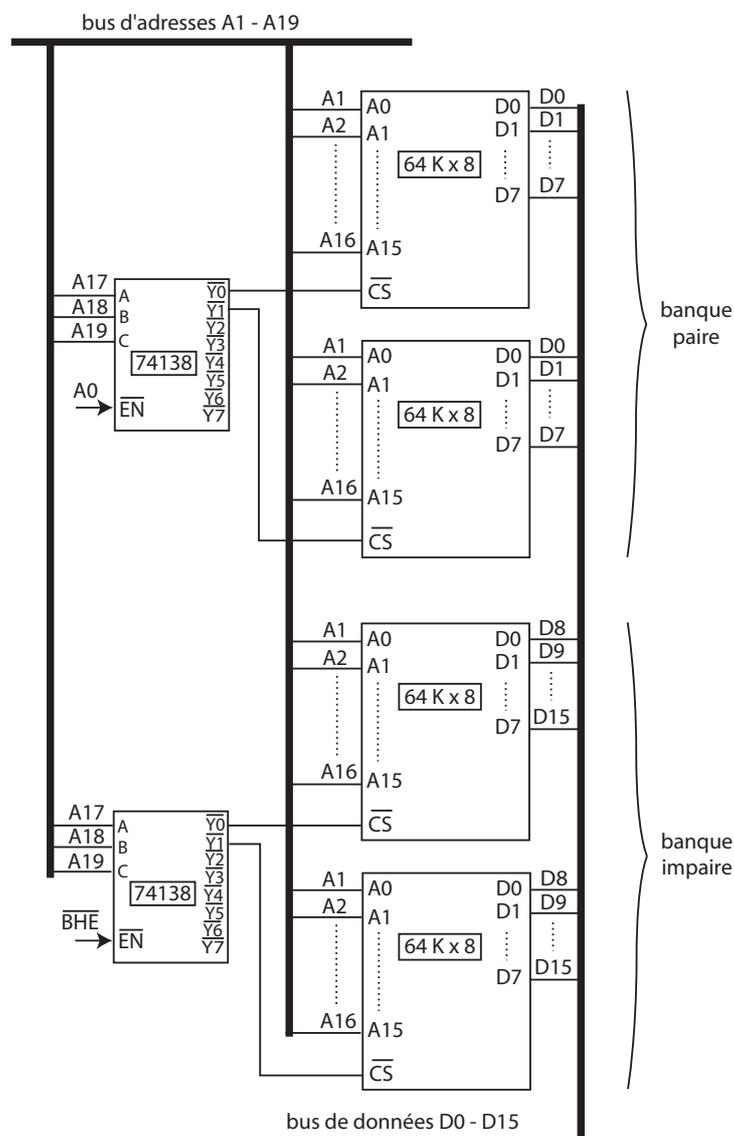


Seuls les bits A1 à A19 servent à désigner une case mémoire dans chaque banque de 512 Ko. Le microprocesseur peut ainsi lire et écrire des données sur 8 bits ou sur 16 bits :

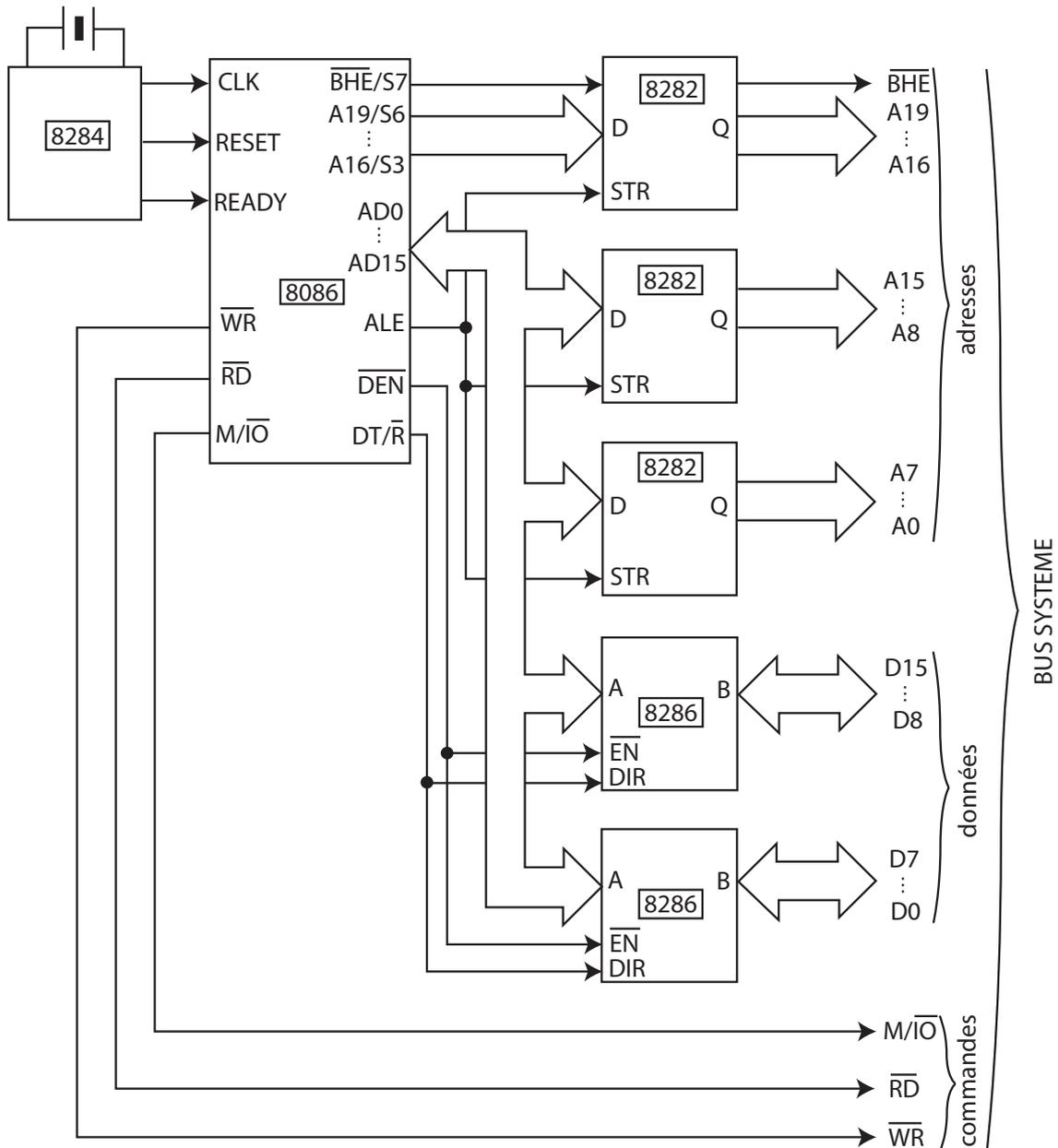
$\overline{\text{BHE}}$	A0	octets transférés
0	0	les deux octets (mot complet)
0	1	octet fort (adresse impaire)
1	0	octet faible (adresse paire)
1	1	aucun octet

Remarque : le 8086 ne peut lire une donnée sur 16 bits en une seule fois, uniquement si l'octet de poids fort de cette donnée est rangé à une adresse impaire et l'octet de poids faible à une adresse paire (alignement sur les adresses paires), sinon la lecture de cette donnée doit se faire en deux opérations successives, d'où une augmentation du temps d'exécution du transfert dû à un mauvais alignement des données.

Réalisation des deux banques avec plusieurs boîtiers mémoire :



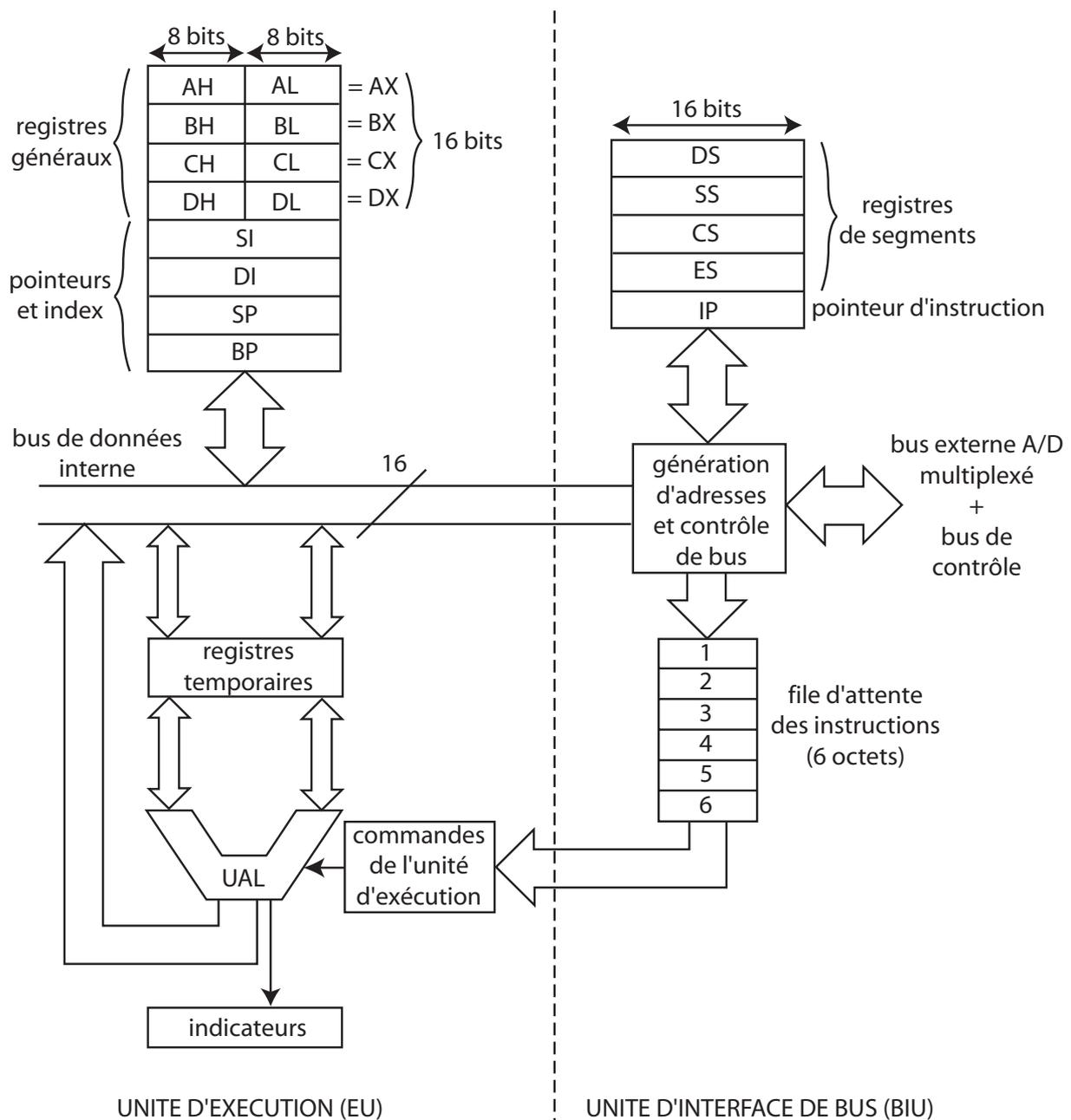
Création du bus système du 8086 :



4.4 Organisation interne du 8086

Le 8086 est constitué de deux unités fonctionnant en parallèle :

- l'unité d'exécution (EU : Execution Unit) ;
- l'unité d'interface de bus (BIU : Bus Interface Unit).



Rôle des deux unités :

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

Les deux unités fonctionnent simultanément, d'où une accélération du processus d'exécution d'un programme (fonctionnement selon le principe du **pipe-line**).

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) ;

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. Exemple : AX = accumulateur, CX = compteur.

- **Registres de pointeurs et d'index** : 4 registres sur 16 bits.

Pointeurs :

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.

Index :

SI : Source Index ;

DI : Destination Index.

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

Les pointeurs et les index contiennent des adresses de cases mémoire.

- **Pointeur d'instruction et indicateurs (flags)** : 2 registres sur 16 bits.

Pointeur d'instruction : **IP**, contient l'adresse de la prochaine instruction à exécuter.

Flags :

				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 ;

Les registres de segments, associés aux pointeurs et aux index, permettent au microprocesseur 8086 d'adresser l'ensemble de la mémoire.

4.5 Gestion de la mémoire par le 8086

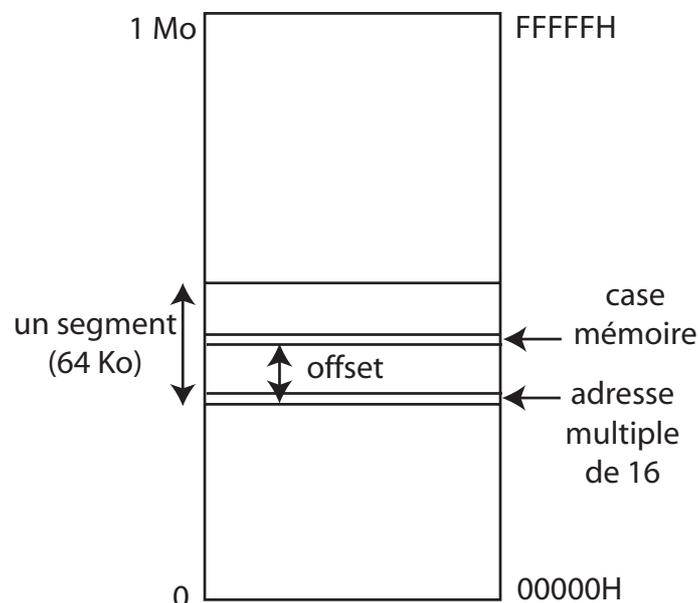
L'espace mémoire adressable par le 8086 est de $2^{20} = 1\,048\,576$ octets = 1 Mo (20 bits d'adresses). Cet espace est divisé en **segments**. Un segment est une zone mémoire de 64 Ko (65 536 octets) définie par son adresse de départ qui doit être un multiple de 16. Dans une telle adresse, les 4 bits de poids faible sont à zéro. On peut donc représenter l'adresse d'un segment avec seulement ses 16 bits de poids fort, les 4 bits de poids faible étant implicitement à 0.

Pour désigner une case mémoire parmi les $2^{16} = 65\,536$ contenues dans un segment, il suffit d'une valeur sur 16 bits.

Ainsi, une case mémoire est repérée par le 8086 au moyen de deux quantités sur 16 bits :

- l'adresse d'un segment ;
- un déplacement ou **offset** (appelé aussi **adresse effective**) dans ce segment.

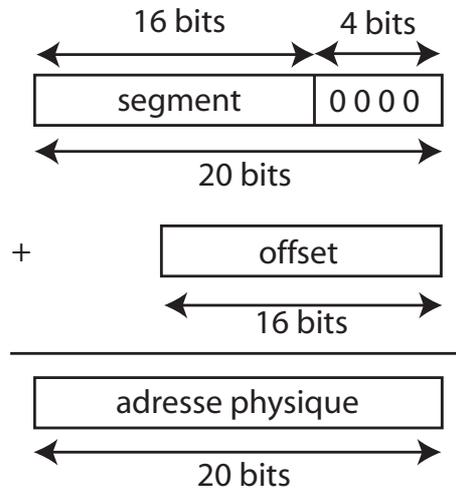
Cette méthode de gestion de la mémoire est appelée **segmentation de la mémoire**.



La donnée d'un couple (segment,offset) définit une **adresse logique**, notée sous la forme **segment : offset**.

L'adresse d'une case mémoire donnée sous la forme d'une quantité sur 20 bits (5 digits hexa) est appelée **adresse physique** car elle correspond à la valeur envoyée réellement sur le bus d'adresses A0 - A19.

Correspondance entre adresse logique et adresse physique :



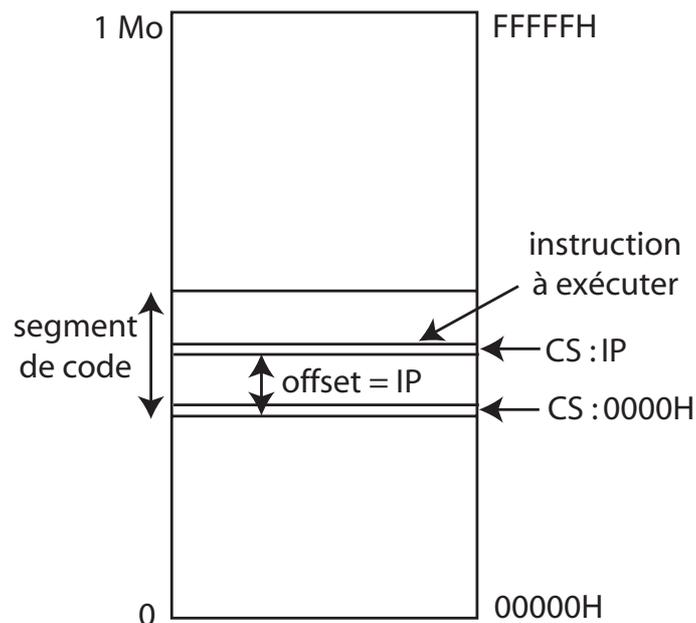
Ainsi, l'adresse physique se calcule par l'expression :

$$\text{adresse physique} = 16 \times \text{segment} + \text{offset}$$

car le fait d'injecter 4 zéros en poids faible du segment revient à effectuer un décalage de 4 positions vers la gauche, c'est à dire une multiplication par $2^4 = 16$.

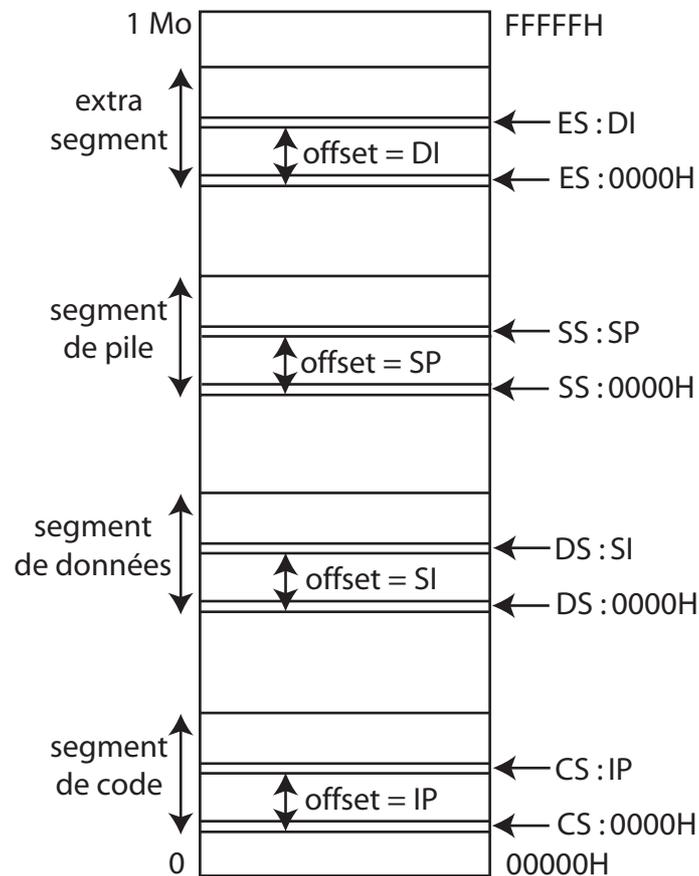
A un instant donné, le 8086 a accès à 4 segments dont les adresses se trouvent dans les registres de segment CS, DS, SS et ES. Le segment de code contient les instructions du programme, le segment de données contient les données manipulées par le programme, le segment de pile contient la pile de sauvegarde et le segment supplémentaire peut aussi contenir des données.

Le registre CS est associé au pointeur d'instruction IP, ainsi la prochaine instruction à exécuter se trouve à l'adresse logique CS : IP.

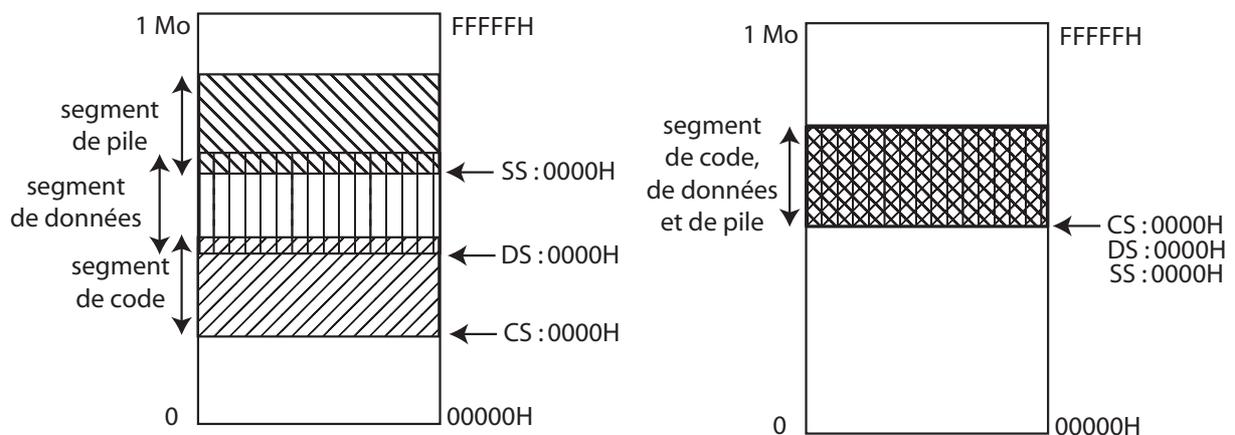


De même, les registres de segments DS et ES peuvent être associés à un registre d'index. Exemple : DS : SI, ES : DI. Le registre de segment de pile peut être associé aux registres de pointeurs : SS : SP ou SS : BP.

Mémoire accessible par le 8086 à un instant donné :



Remarque : les segments ne sont pas nécessairement distincts les uns des autres, ils peuvent se chevaucher ou se recouvrir complètement.



Le nombre de segments utilisé définit le **modèle mémoire** du programme.

Contenu des registres après un RESET du microprocesseur :

IP = 0000H

CS = FFFFH

DS = 0000H

ES = 0000H

SS = 0000H

Puisque CS contient la valeur FFFFH et IP la valeur 0000H, la première instruction exécutée par le 8086 se trouve donc à l'adresse logique FFFFH : 0000H, correspondant à l'adresse physique FFFF0H (bootstrap). Cette instruction est généralement un saut vers le programme principal qui initialise ensuite les autres registres de segment.

4.6 Le microprocesseur 8088

Le microprocesseur 8088 est identique au 8086 sauf que son bus de données externe est sur 8 bits au lieu de 16 bits, le bus de données interne restant sur 16 bits.

Le 8088 a été produit par Intel après le 8086 pour assurer la compatibilité avec des circuits périphériques déjà existant, fabriqués pour les microprocesseurs 8 bits 8080 et 8085.

Différences avec le 8086 :

- les broches AD8 à AD15 deviennent A8 à A15 (bus de données sur 8 bits) ;
- la broche $\overline{\text{BHE}}$ n'existe pas dans le 8088 car il n'y a pas d'octet de poids fort sur le bus de données ;
- la broche M/ $\overline{\text{IO}}$ devient IO/ $\overline{\text{M}}$ pour la compatibilité avec d'anciens circuits d'E/S.

Au niveau de l'architecture interne, pas de différences avec le 8086 sauf que la file d'attente des instructions passe de 6 à 4 octets.