

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Ziane Achour de Djelfa

جامعة الجلفة

Faculté des Sciences et de la Technologie

كلية العلوم و التكنولوجيا

Département de Génie Electrique

قسم الهندسة الكهربائية



Semestre: 1

Matières: Microprocesseurs et Microcontrôleurs

1^{er} année Master ELT(Electrotechnique)

Chapitre 3

Les interruptions et les interfaces d'entrées/sorties

Contenu de la matière :

Définition d'une interruption, Prise en charge d'une interruption par le microprocesseur, Adressages des sous programmes d'interruptions,

Adressages des ports d'E/S, Gestion des ports d'E/S

Réalisé et présenté par :

Dr. Obeidi Thameur

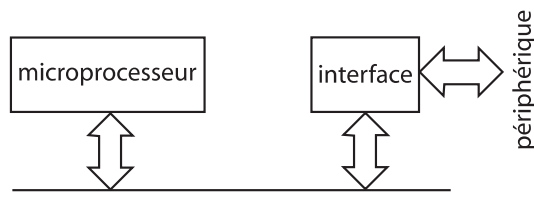
Année Universitaire : 2023 / 2024

Chapitre 3

Les interruptions et les interfaces d'entrées/sorties

1. Définition d'une interruption

Soit un microprocesseur qui doit échanger des informations avec un périphérique :



Il y a deux méthodes possibles pour recevoir les données provenant des périphériques :

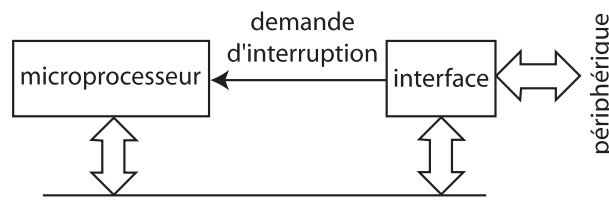
- **scrutation périodique** (ou **polling**) : le programme principal contient des instructions qui lisent cycliquement l'état des ports d'E/S.

Avantage : facilité de programmation.

Inconvénients :

- perte de temps s'il y a de nombreux périphériques à interroger ;
- de nouvelles données ne sont pas toujours présentes ;
- des données peuvent être perdues si elles changent rapidement.

- **interruption** : lorsqu'une donnée apparaît sur un périphérique, le circuit d'E/S le signale au microprocesseur pour que celui-ci effectue la lecture de la donnée : c'est une **demande d'interruption** (IRQ : Interrupt Request) :



Avantage : le microprocesseur effectue une lecture des ports d'E/S seulement lorsqu'une donnée est disponible, ce qui permet de gagner du temps et d'éviter de perdre des données.

Exemples de périphériques utilisant les interruptions :

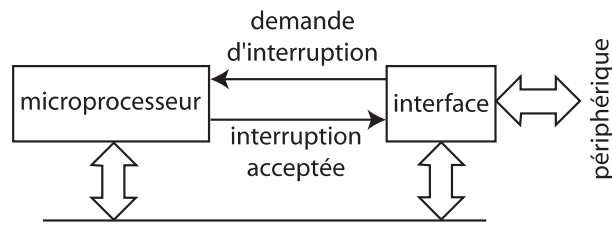
- clavier : demande d'interruption lorsqu'une touche est enfoncée ;
- port série : demande d'interruption lors de l'arrivée d'un caractère sur la ligne de transmission.

Remarque : les interruptions peuvent être générées par le microprocesseur lui-même en cas de problèmes tels qu'une erreur d'alimentation, une division par zéro ou un circuit mémoire défectueux (erreurs fatales). Dans ce cas, la demande d'interruption conduit à l'arrêt du microprocesseur.

2. Prise en charge d'une interruption par le microprocesseur

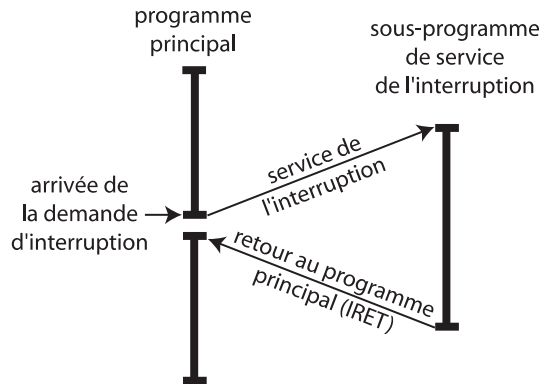
A la suite d'une demande d'interruption par un périphérique :

- le microprocesseur termine l'exécution de l'instruction en cours ;
- il range le contenu des principaux registres sur la pile de sauvegarde : pointeur d'instruction, flags, ...
- il émet un **accusé de réception** de demande d'interruption (Interrupt Acknowledge) indiquant au circuit d'E/S que la demande d'interruption est acceptée :



Remarque : le microprocesseur peut refuser la demande d'interruption : celle-ci est alors **masquée**. Le masquage d'une interruption se fait généralement en positionnant un flag dans le registre des indicateurs d'état. Il existe cependant des interruptions **non masquables** qui sont toujours prises en compte par le microprocesseur.

- il abandonne l'exécution du programme en cours et va exécuter un **sous-programme de service de l'interruption** (ISR : Interrupt Service Routine) ;
- après l'exécution de l'ISR, les registres sont restaurés à partir de la pile et le microprocesseur reprend l'exécution du programme qu'il avait abandonné :

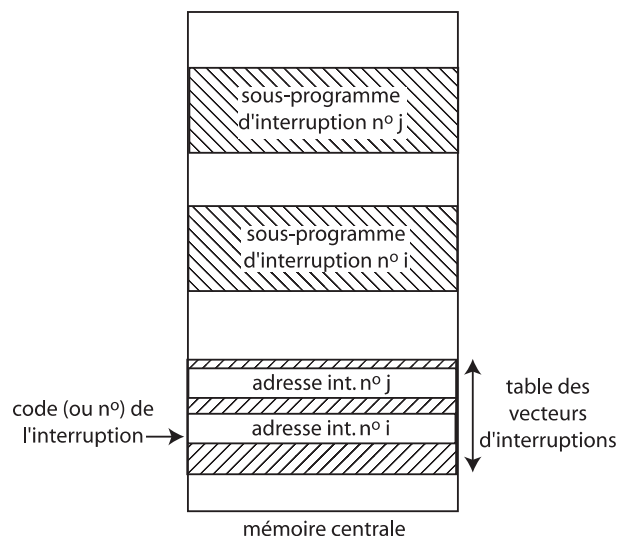


Remarque : la dernière instruction d'un sous-programme de service d'interruption doit être l'instruction **IRET** : retour d'interruption.

Si plusieurs interruptions peuvent se produire en même temps, on doit leur affecter une **priorité** pour que le microprocesseur sache dans quel ordre il doit servir chacune d'entre elle.

3. Adresses des sous-programmes d'interruptions

Lorsqu'une interruption survient, le microprocesseur a besoin de connaître l'adresse du sous-programme de service de cette interruption. Pour cela, la source d'interruption place sur le bus de données un code numérique indiquant la nature de l'interruption. Le microprocesseur utilise ce code pour rechercher dans une table en mémoire centrale l'adresse du sous-programme d'interruption à exécuter. Chaque élément de cette table s'appelle un **vecteur d'interruption** :



Lorsque les adresses des sous-programmes d'interruptions sont gérées de cette manière, on dit que les interruptions sont **vectorisées**.

Avantage de la vectorisation des interruptions : l'emplacement d'une ISR peut être n'importe où dans la mémoire, il suffit de spécifier le vecteur d'interruption correspondant.

4. Les interruptions du 8086

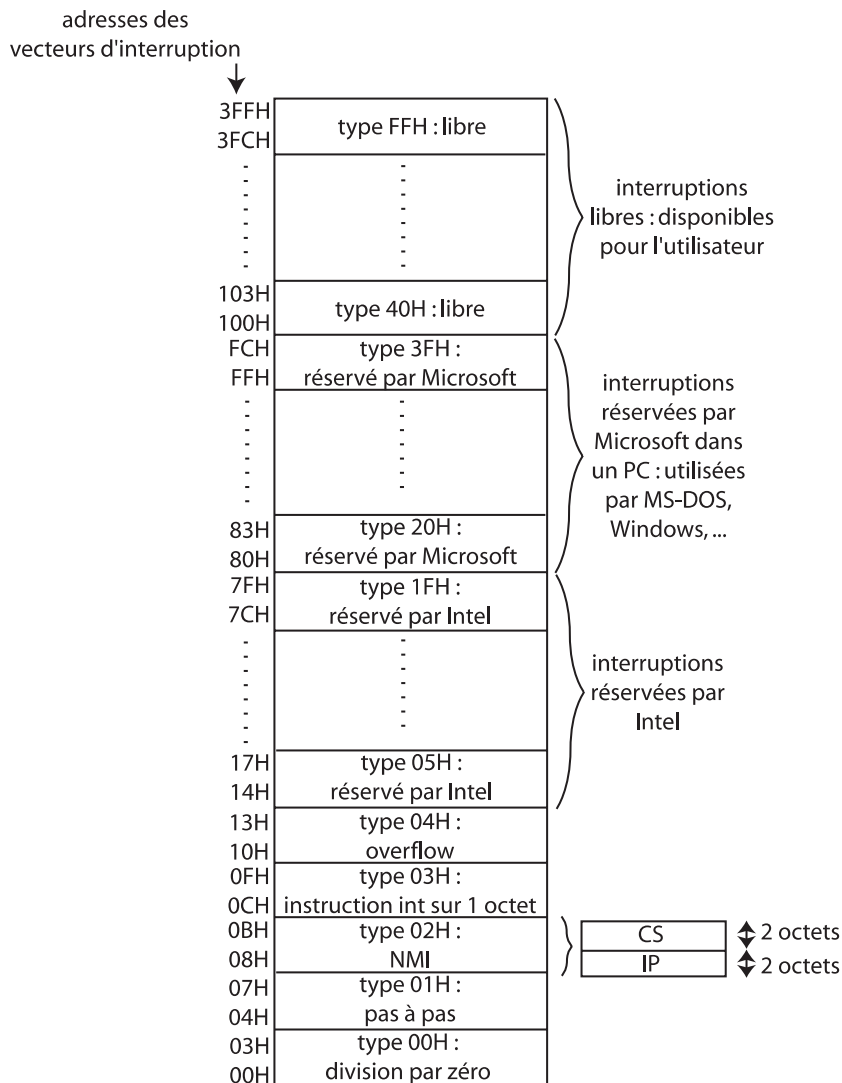
Le microprocesseur 8086 peut gérer jusqu'à 256 interruptions. Chaque interruption reçoit un numéro compris entre 0 et 255 appelé **type** de l'interruption.

Trois sortes d'interruptions sont reconnues par le 8086 :

- interruptions **matérielles** produites par l'activation des lignes INTR et NMI du microprocesseur ;
- interruptions **logicielles** produites par l'instruction INT n, où n est le type de l'interruption ;
- interruptions **processeur** générées par le microprocesseur en cas de dépassement, de division par zéro ou lors de l'exécution pas à pas d'un programme.

Les interruptions du 8086 sont vectorisées. La table des vecteurs d'interruptions doit obligatoirement commencer à l'adresse physique 00000H dans la mémoire centrale.

Chaque vecteur d'interruption est constitué de 4 octets représentant une adresse logique du type **CS : IP**.



Remarque : correspondance entre le type de l'interruption et l'adresse du vecteur correspondant :

$$\text{adresse vecteur d'interruption} = 4 \times \text{type de l'interruption}$$

Exemple : interruption 20H, adresse du vecteur = $4 \times 20H = 80H$.

La table des vecteurs d'interruptions est chargée par le programme principal (carte à microprocesseur) ou par le système d'exploitation (ordinateur) au démarrage du système. Elle peut être modifiée en cours de fonctionnement (détournement des vecteurs d'interruptions).

5. Le contrôleur programmable d'interruptions 8259

Le microprocesseur 8086 ne dispose que de deux lignes de demandes d'interruptions matérielles (NMI et INTR). Pour pouvoir connecter plusieurs périphériques utilisant des interruptions, on peut utiliser le contrôleur programmable d'interruptions 8259 dont le rôle est de :

- recevoir des demandes d'interruptions des périphériques ;
- résoudre les priorités des interruptions ;
- générer le signal INTR pour le 8086 ;
- émettre le numéro de l'interruption sur le bus de données.

Un 8259 peut gérer jusqu'à 8 demandes d'interruptions matérielles.

Brochage du 8259 :

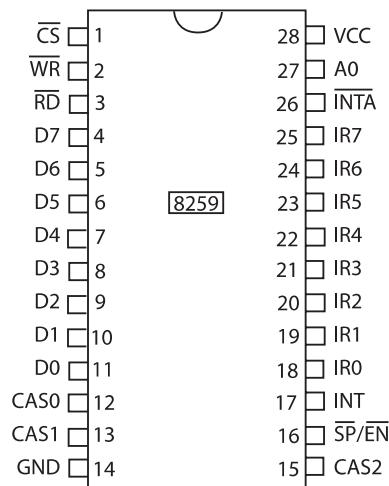
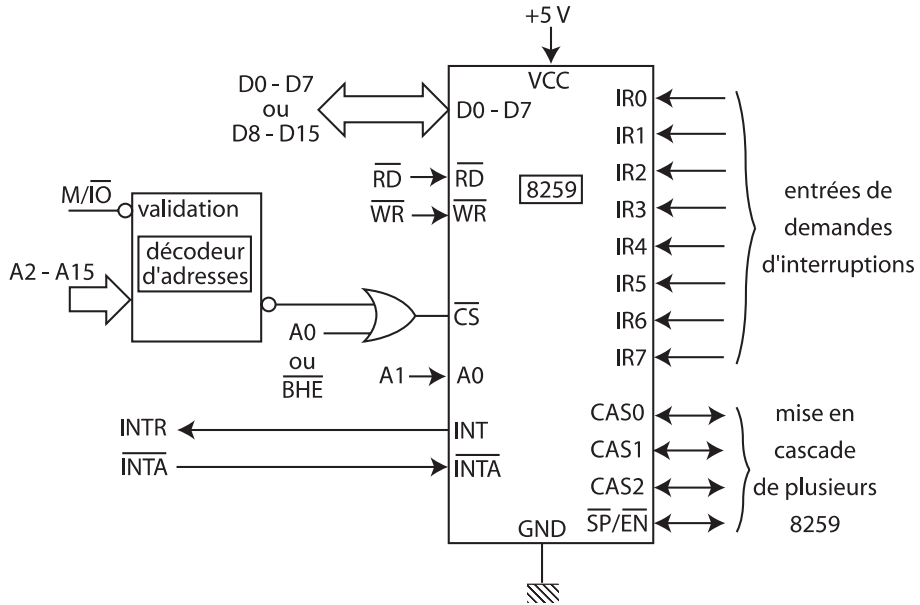
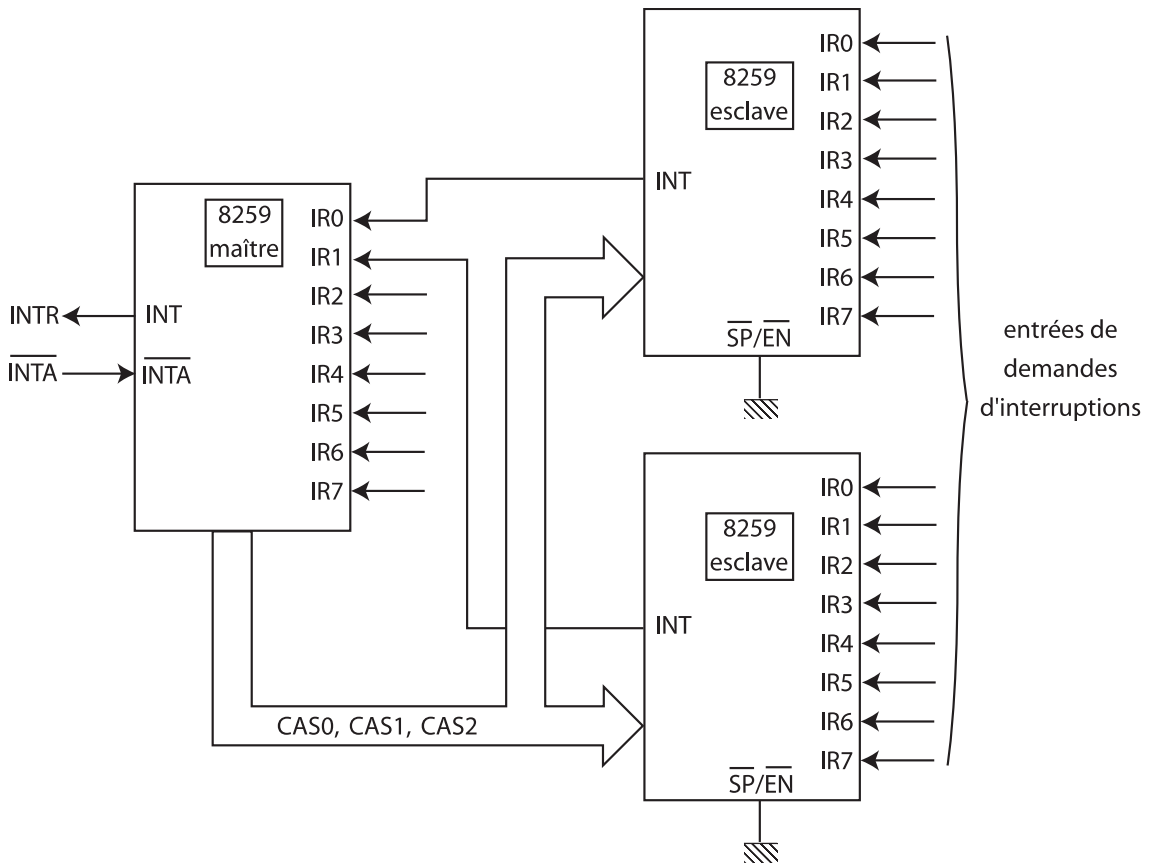


Schéma fonctionnel :



Remarque : si le nombre de demandes d'interruptions est supérieur à 8, on peut placer plusieurs 8259 en cascade :

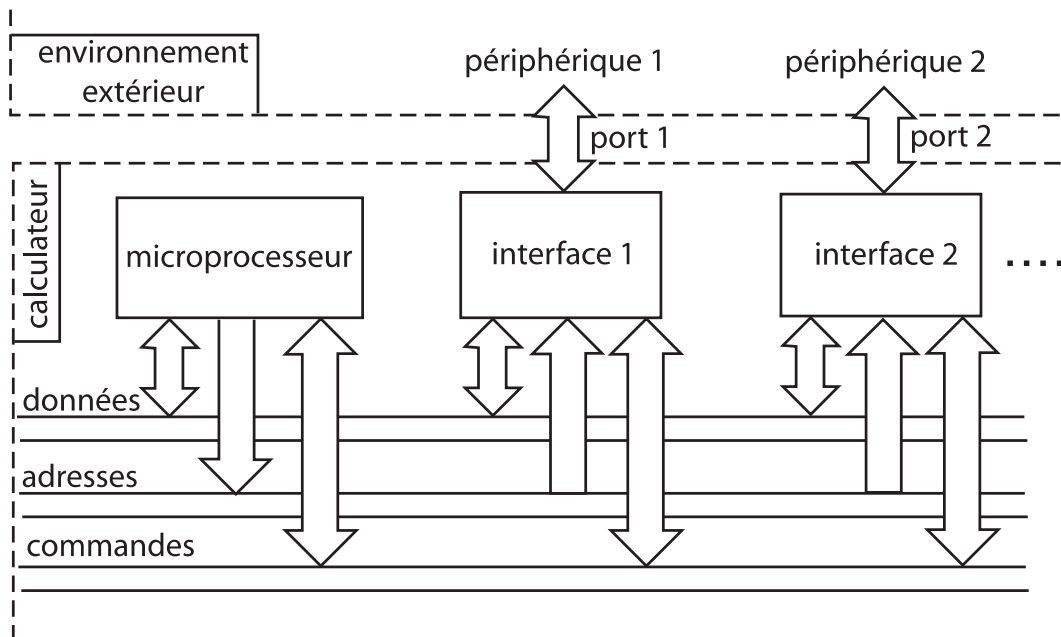


Les interfaces d'entrées/sorties

1. Définitions

Une **interface d'entrées/sorties** est un circuit intégré permettant au microprocesseur de communiquer avec l'environnement extérieur (périphériques) : clavier, écran, imprimante, modem, disques, processus industriel, ...

Les interfaces d'E/S sont connectées au microprocesseur à travers les bus d'adresses, de données et de commandes.

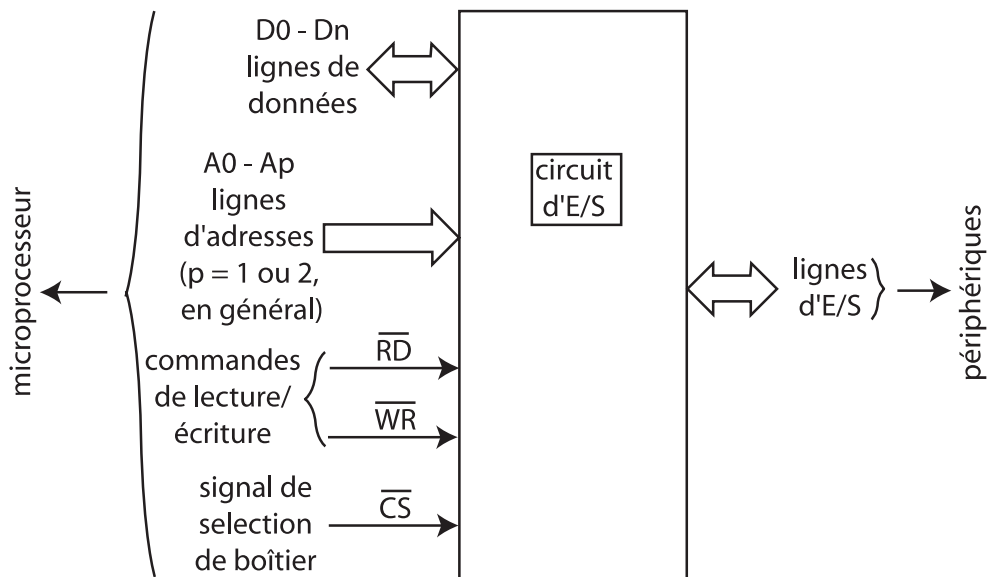


Les points d'accès aux interfaces sont appelés **ports**.

Exemples :

interface	port	exemple de périphérique
interface parallèle	port parallèle	imprimante
interface série	port série	modem

Schéma synoptique d'un circuit d'E/S :



2. Adressage des ports d'E/S

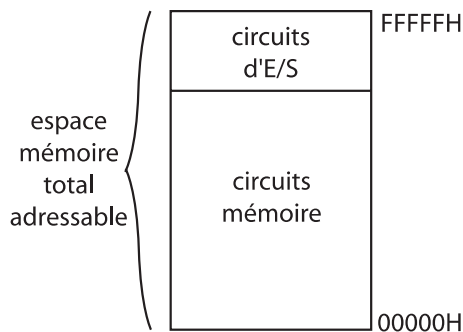
Un circuit d'E/S possède des registres pour gérer les échanges avec les périphériques :

- registres de configuration ;
- registres de données.

A chaque registre est assigné une adresse : le microprocesseur accède à un port d'E/S en spécifiant l'adresse de l'un de ses registres.

Le microprocesseur peut voir les adresses des ports d'E/S de deux manières :

- **adressage cartographique** : les adresses des ports d'E/S appartiennent au même espace mémoire que les circuits mémoire (on dit que les E/S sont **mappées en mémoire**) :



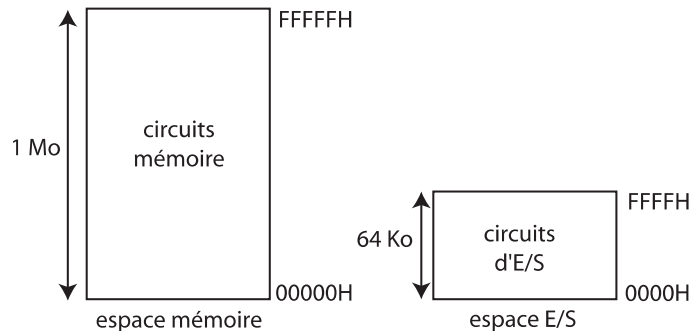
Conséquences :

- l'espace d'adressage des mémoires diminue ;
- l'adressage des ports d'E/S se fait avec une adresse de même longueur (même nombre de bits) que pour les cases mémoires ;

- toutes les instructions employées avec des cases mémoires peuvent être appliquées aux ports d'E/S : les mêmes instructions permettent de lire et écrire dans la mémoire et les ports d'E/S, tous les modes d'adressage étant valables pour les E/S.

- **adressage indépendant** : le microprocesseur considère deux espaces distincts :
 - l'espace d'adressage des mémoires ;
 - l'espace d'adressage des ports d'E/S.

C'est le cas du microprocesseur 8086 :



Conséquences :

- contrairement à l'adressage cartographique, l'espace mémoire total adressable n'est pas diminué ;
- l'adressage des port d'E/S peut se faire avec une adresse plus courte (nombre de bits inférieur) que pour les circuits mémoires ;
- les instructions utilisées pour l'accès à la mémoire ne sont plus utilisables pour l'accès aux ports d'E/S : ceux-ci disposent d'instructions spécifiques ;
- une même adresse peut désigner soit une case mémoire, soit un port d'E/S : le microprocesseur doit donc fournir un signal permettant de différencier l'adressage de la mémoire de l'adressage des ports d'E/S.

Remarque : l'adressage indépendant des ports d'E/S n'est possible que pour les microprocesseurs possédant un signal permettant de différencier l'adressage de la mémoire de l'adressage des ports d'E/S ainsi que les instructions spécifiques pour l'accès aux ports d'E/S. Par contre, l'adressage cartographique est possible pour tous les microprocesseurs.

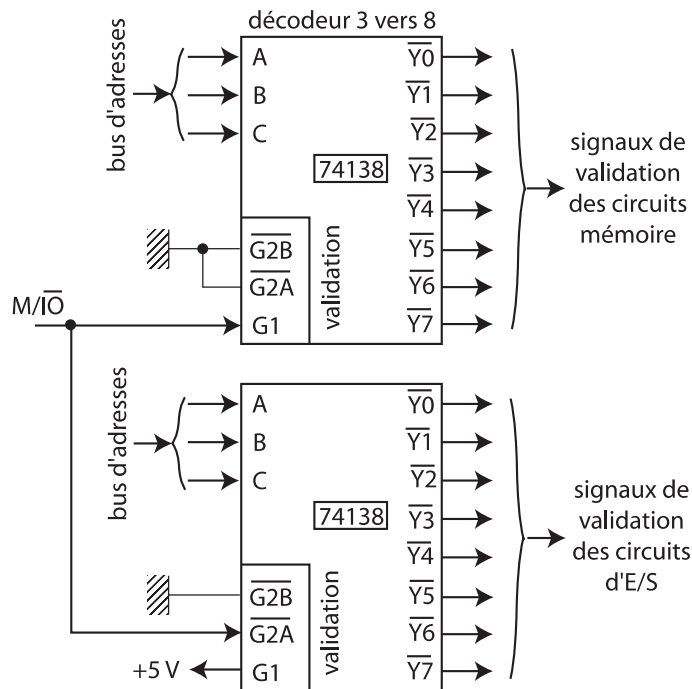
3. Gestion des ports d'E/S par le 8086

Le 8086 dispose d'un espace mémoire de 1 Mo (adresse d'une case mémoire sur 20 bits) et d'un espace d'E/S de 64 Ko (adresse d'un port d'E/S sur 16 bits).

Le signal permettant de différencier l'adressage de la mémoire de l'adressage des ports d'E/S est la ligne M/\overline{IO} :

- pour un accès à la mémoire, $M/\overline{IO} = 1$;
- pour un accès aux ports d'E/S, $M/\overline{IO} = 0$.

Ce signal est utilisé pour valider le décodage d'adresse dans les deux espaces :



Les instructions de lecture et d'écriture d'un port d'E/S sont respectivement les instructions **IN** et **OUT**. Elles placent la ligne M/I0 à 0 alors que l'instruction MOV place celle-ci à 1.

Lecture d'un port d'E/S :

- si l'adresse du port d'E/S est sur un octet :
`IN AL, adresse` : lecture d'un port sur 8 bits ;
`IN AX, adresse` : lecture d'un port sur 16 bits.

- si l'adresse du port d'E/S est sur deux octets :

`IN AL, DX` : lecture d'un port sur 8 bits ;
`IN AX, DX` : lecture d'un port sur 16 bits.

où le registre DX contient l'adresse du port d'E/S à lire.

Écriture d'un port d'E/S :

- si l'adresse du port d'E/S est sur un octet :
`OUT adresse, AL` : écriture d'un port sur 8 bits ;
`OUT adresse, AX` : écriture d'un port sur 16 bits.

- si l'adresse du port d'E/S est sur deux octets :

`OUT DX, AL` : écriture d'un port sur 8 bits ;
`OUT DX, AX` : écriture d'un port sur 16 bits.

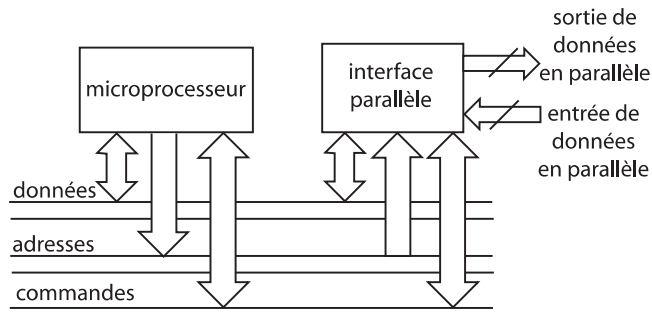
où le registre DX contient l'adresse du port d'E/S à écrire.

Exemples :

- lecture d'un port d'E/S sur 8 bits à l'adresse 300H :
`mov dx,300H`
`in al,dx`
- écriture de la valeur 1234H dans le port d'E/S sur 16 bits à l'adresse 49H :
`mov ax,1234H`
`out 49H,ax`

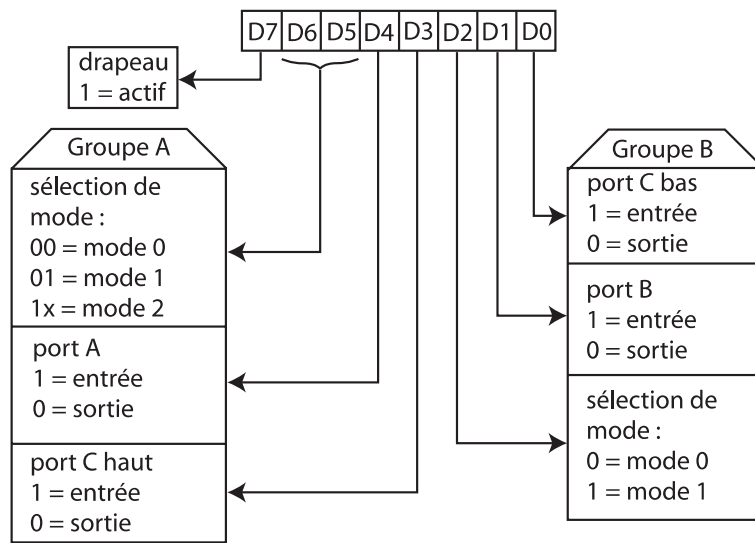
4 L'interface parallèle 8255

Le rôle d'une interface parallèle est de transférer des données du microprocesseur vers des périphériques et inversement, tous les bits de données étant envoyés ou reçus simultanément.



Le 8255 est une interface parallèle programmable : elle peut être configurée en entrée et/ou en sortie par programme.

Structure du registre de commande :



Connexion du 8255 sur les bus du 8086 : le bus de données du 8255 est sur 8 bits alors que celui du microprocesseur 8086 est sur 16 bits. On peut donc connecter le bus de données du 8255 sur les lignes de données de poids faible du 8086 (D0 - D7) ou sur celles de poids fort (D8 - D15).

Une donnée est envoyée (ou reçue) par le microprocesseur 8086 :

- sur la partie faible du bus de données lorsque l'adresse à écrire (ou à lire) est paire : validation par A0 ;
- sur la partie haute lorsque l'adresse est impaire : validation par $\overline{\text{BHE}}$.