

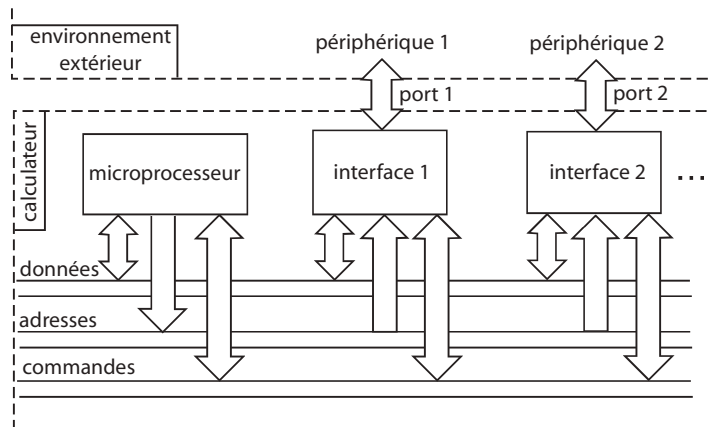
Chapitre 6

Les interfaces d'entrées/sorties

6.1 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.

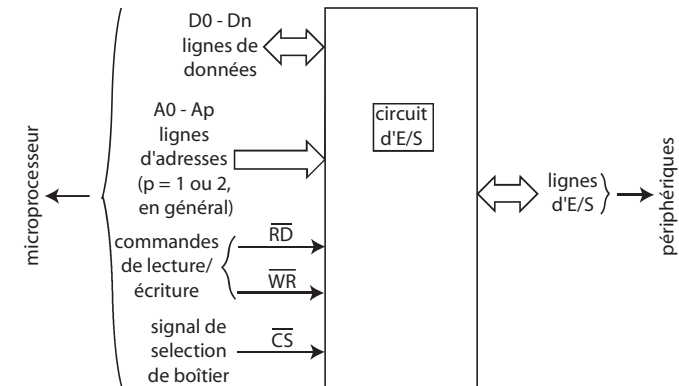


Les points d'accès aux interfaces sont appelés **ports**.

Exemples :

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

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



6.2 Adressage des ports d'E/S

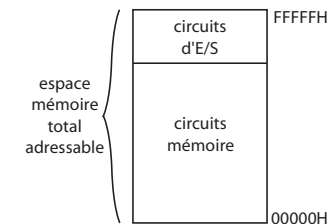
Un circuit d'E/S possède des registres pour gérer les échanges avec les périphériques :

- 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**) :



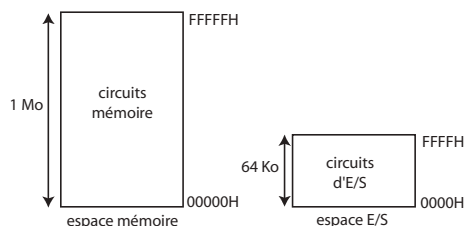
Conséquences :

- 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.

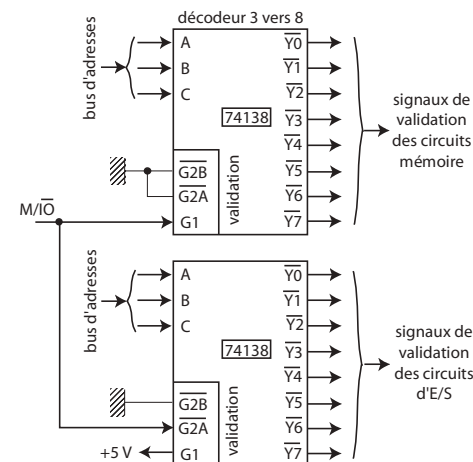
6.3 Gestion des ports d'E/S par le 8086

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 $M/\overline{IO}$ 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 :

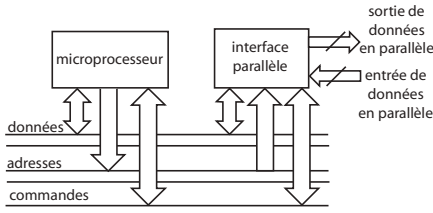
- si l'adresse du port d'E/S est sur un octet :
 - OUT adresse, AL** : écriture d'un port sur 8 bits ;
 - OUT adresse, AX** : écriture d'un port sur 16 bits.
 - si l'adresse du port d'E/S est sur deux octets :
 - OUT DX, AL** : écriture d'un port sur 8 bits ;
 - OUT DX, AX** : écriture d'un port sur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

6.4 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.

Brochage du 8255 :

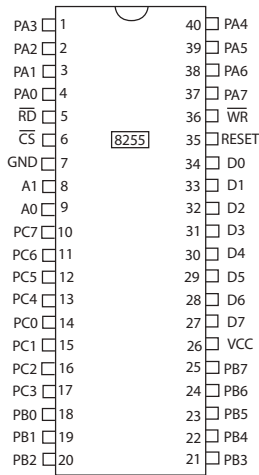
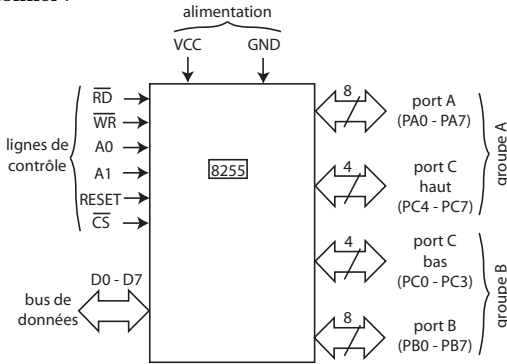


Schéma fonctionnel :



Le 8255 contient 4 registres :

- trois registres contenant les données présentes sur les ports A, B et C ;
- un registre de commande pour la configuration des port A, B et C en entrées et/ou en sorties.

Accès aux registres du 8255 : les lignes d'adresses A0 et A1 définissent les adresses des registres du 8255 :

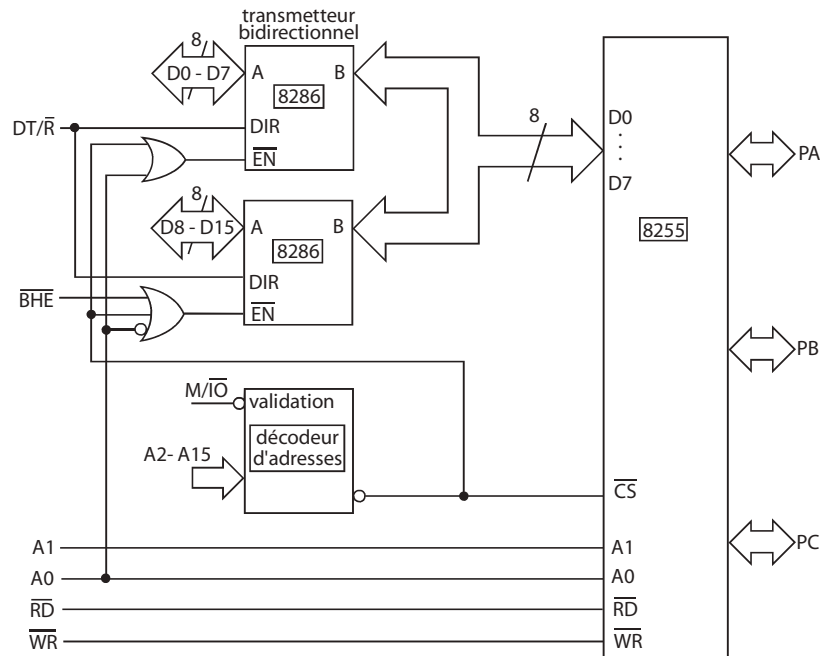
A1	A0	RD	WR	CS	opération
0	0	0	1	0	lecture du port A
0	1	0	1	0	lecture du port B
1	0	0	1	0	lecture du port C
0	0	1	0	0	écriture du port A
0	1	1	0	0	écriture du port B
1	0	1	0	0	écriture du port C
1	1	1	0	0	écriture du registre de commande
X	X	X	X	1	pas de transaction
1	1	0	1	0	illégal
X	X	1	1	0	pas de transaction

Remarque : le registre de commande est accessible uniquement en écriture, la lecture de ce registre n'est pas autorisée.

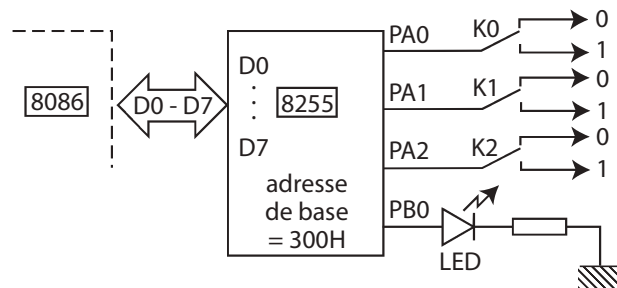
Configuration du 8255 : les ports peuvent être configurés en entrées ou en sorties selon le contenu du registre de commande. De plus le 8255 peut fonctionner selon 3 modes : mode 0, mode 1 ou mode 2.

Le mode 0 est le plus simple : les ports sont configurés en entrées/sorties de base. Les données écrites dans les registres correspondants sont mémorisées sur les lignes de sorties ; l'état des lignes d'entrées est recopié dans les registres correspondants et n'est pas mémorisé.

Les modes 1 et 2 sont plus complexes. Ils sont utilisés pour le dialogue avec des périphériques nécessitant un asservissement.



Exemple de programmation : soit le montage suivant :



On veut que la led s'allume lorsqu'on a la combinaison : $K0 = 1$ et $K1 = 0$ et $K2 = 1$.

Programme :

```

portA      equ 300H      ; adresses des registres du 8255
portB      equ 302H
portC      equ 304H
controle   equ 306H

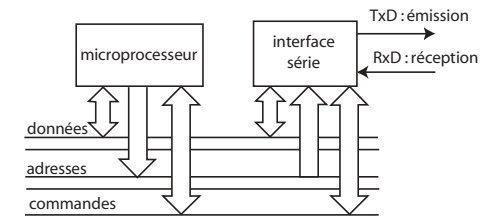
        mov dx,controle  ; initialisation du port A en entrée
        mov al,90H       ; et du port B en sortie (mode 0) :
        out dx,al        ; controle = 10010000B = 90H

boucle :  mov dx,portA    ; lecture du port A
        in  al,dx
        and al,00000111B ; masquage PA0, PA1 et PA2
        cmp al,00000101B ; test PA0 = 1, PA1 = 0 et PA2 = 1
        jne faux        ; non -> aller au label 'faux' ...
        mov al,00000001B ; oui -> mettre PB0 à 1
        jmp suite       ; et continuer au label 'suite'
faux :    mov al,00000000B ; ... mettre PB0 à 0
suite :   mov dx,portB    ; écriture du port B
        out dx,al
        jmp boucle       ; retourner lire le port A

```

6.5 L'interface série 8250

Une interface série permet d'échanger des données entre le microprocesseur et un périphérique bit par bit.



Avantage : diminution du nombre de connexions (1 fil pour l'émission, 1 fil pour la réception).

Inconvénient : vitesse de transmission plus faible que pour une interface parallèle.

Il existe deux types de transmissions séries :

- **asynchrone** : chaque octet peut être émis ou reçu sans durée déterminée entre un octet et le suivant ;
- **synchrone** : les octets successifs sont transmis par blocs séparés par des octets de synchronisation.

La transmission asynchrone la plus utilisée est celle qui est définie par la norme **RS232**.