

Chapitre 5

Pilotage des périphériques par interruption matérielle

Nous avons vu la notion d'entrée-sortie d'un point de vue logiciel et sa réalisation matérielle de base. Nous avons vu comment construire les ports d'entrées-sorties. Nous avons vu également comment contrôler le transfert des données entre ces ports et le microprocesseur en utilisant la méthode de l'interrogation. Le problème avec cette méthode est que le microprocesseur va passer son temps à interroger alors que rien ne se passe. Nous allons donc voir une autre technique : celle des **entrées-sorties pilotées par interruption matérielle** dans laquelle le processeur peut vaquer à ses occupations ; on le prévient lorsqu'un périphérique a besoin de lui.

5.1 Rappels sur l'interruption du microprocesseur

5.1.1 Principe de l'interruption du microprocesseur

Principe.- Un microprocesseur acceptant d'être interrompu possède une broche spéciale, disons INTR pour *INTErruption*, accessible en entrée seulement. À la fin de chaque instruction, le microprocesseur vérifie l'entrée de la broche d'interruption INTR. En général elle n'est pas active, aussi le microprocesseur passe-t-il à l'instruction suivante. Mais lorsqu'elle est active, par contre, le contrôle est transféré à une *routine de service d'interruption* (ISR pour *Interrupt Service Routine*) conformément au schéma ci-dessous :

1. Processus normal.
2. Apparition d'une interruption.
3. Termine l'instruction en cours.
4. Empile le registre des drapeaux, CS et IP sur la pile.
5. Se reporte à l'ISR.
6. Exécute l'ISR, se terminant par IRET (*Interrupt RETurn*).
7. Dépile IP, CS et le registre des drapeaux.
8. Revient au processus normal.

Interruptions masquables.- On distingue quelquefois deux types d'interruptions matérielles : celles du premier type, les **interruption matérielle non masquables**, doivent nécessairement être prise en compte par le processeur alors que celles du second type, les **interruptions matérielles masquables**, peuvent ne pas être prise en compte si un indicateur est positionné.

Les interruptions non masquables sont en général réservées aux fonctions critiques du système, par exemple sauver l'état du microprocesseur en cas de coupure imminente du courant.

5.1.2 Cas du 8086

Broches impliquées.- Le microprocesseur 8086 possède deux broches spécialisées pour les interruptions matérielles : INTR pour les interruptions masquables et NMI (pour *NonMaskable Interrupt*) pour les interruptions non masquables. Les interruptions masquables sont masquées lorsque l'indicateur d'interruption IF est positionné.

Instructions de masquage.- Le positionnement de l'indicateur IF s'effectue de façon logicielle grâce à l'instruction :

STI

(pour *SeT Interrupt flag*). On le met à zéro grâce à l'instruction :

CLI

(pour *CLear Interrupt flag*).

On ne positionne pas IF en général lorsqu'on effectue déjà une interruption.

Prise en charge des interruptions.- La figure ci-dessous

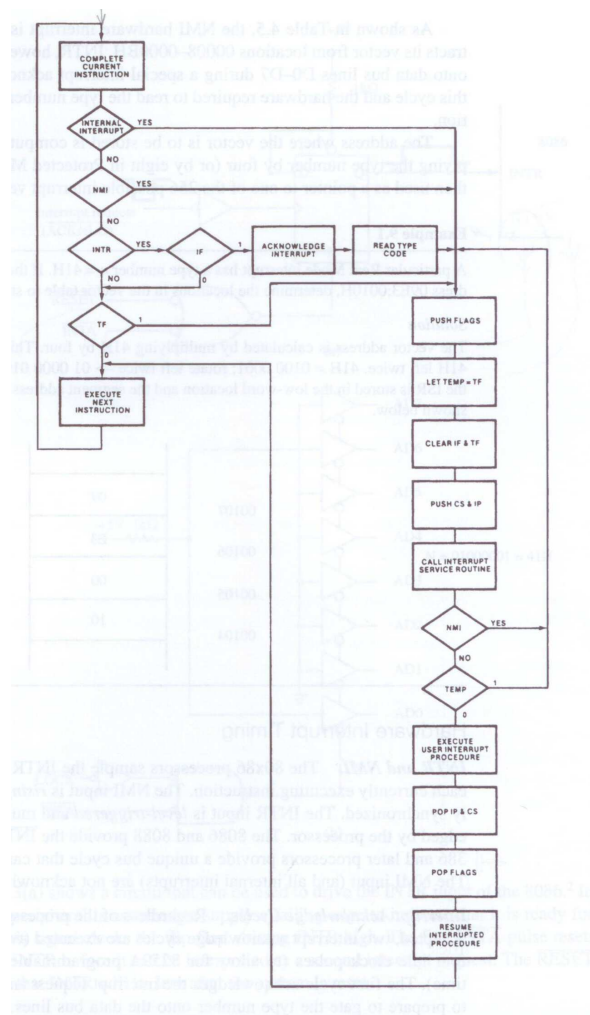


FIGURE 5.1 – Prise en charge des interruptions

indique le comportement du microprocesseur à l'égard des interruptions. On remarquera que les interruptions internes (par exemple division par zéro) ont la plus grande priorité, c'est-à-dire qu'elles sont traitées en premier en cas de plusieurs demandes d'interruption. Le microprocesseur met automatiquement l'indicateur IF à zéro lorsqu'une demande d'interruption est reçue. Il est mis à nouveau à un lors de l'instruction IRET. On peut évidemment jouer sur STI et CLI lorsqu'on n'est pas satisfait de ce comportement par défaut.

Réponse à une demande d'interruption matérielle.- À la fin de chaque instruction, le microprocesseur 8086 teste les entrées INTR et NMI. L'entrée NMI est *rising-edge-triggered* et synchronisée de façon interne. Par contre l'entrée INTR est *level-triggered* et doit donc être à niveau haut jusqu'à ce que le signal \overline{INTA} (pour *INTerrupt Acknowledge*) du microprocesseur indique qu'il en a pris connaissance. Il n'y a pas d'accusé de réception pour les interruptions internes et pour NMI.

Lorsque la demande d'interruption $INTR$ est acceptée par le microprocesseur, celui-ci envoie deux impulsions \overline{INTA} (deux cycles d'accusé de réception de l'interruption sont exécutés), séparés par quatre impulsions d'horloge (ceci permet au 8259A, que nous verrons après, de se repositionner). La première impulsion accuse réception de la requête d'interruption et indique au matériel externe de se préparer à mettre le numéro du type d'interruption sur le bus des données. Pendant le second cycle, le microprocesseur envoie le contenu de $D0$ à $D7$, ce qui lui permet de décider entre 256 numéros d'interruptions.

Interfaçage.- La figure :

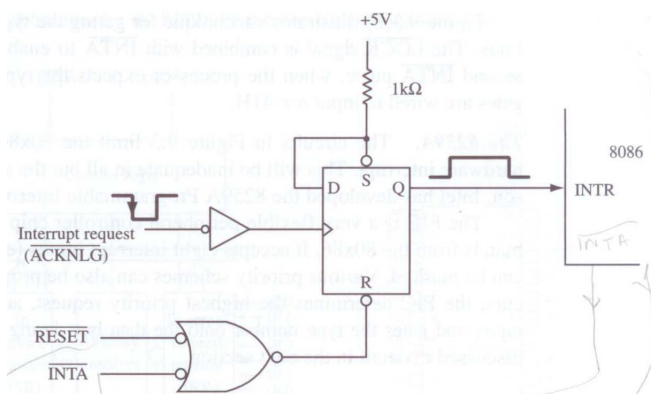


FIGURE 5.2 – Interfaçage des interruptions (1)

montre un circuit qui pouvant être utilisé pour piloter l'entrée $INTR$ du 8088/86. Le signal de demande d'interruption (ici supposé être un signal à niveau bas) entre dans la bascule, ce qui donne un signal $INTR$ à niveau haut. La première impulsion \overline{INTA} remet Q à zéro, ce qui remet $INTR$ à zéro avant qu'il ne soit interprété comme une seconde demande d'interruption. L'entrée \overline{RESET} nous assure que $INTR$ sera à niveau bas lorsque le système démarre.

La figure :

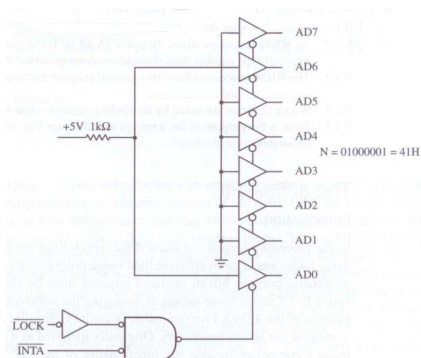


FIGURE 5.3 – Interfaçage des interruptions (2)

illustre une technique indiquant le numéro de l'interruption. Le signal \overline{LOCK} est associé à \overline{INTA} pour positionner les portes à trois états pendant la seconde impulsion \overline{INTA} .

Détermination de l'adresse de l'ISR.- Comme nous l'avons déjà vu à propos des interruptions logicielles, l'adresse de la routine du service d'interruption doit être stockée dans quatre emplacements de mémoire successifs dans la *table de vecteur d'interruption* commençant à l'adresse 00000h. Lorsqu'une interruption survient, un numéro d'interruption de huit bits est fourni au microprocesseur, identifiant l'entrée appropriée de la table. La méthode pour déterminer le numéro de l'interruption dépend de la nature de l'interruption :

- les interruptions logicielles fournissent leur numéro grâce à l'opérande de l'instruction INT n , autrement dit n est le numéro de l'interruption ;
- les interruptions internes ont des numéros prédéfinis par les concepteurs du microprocesseur, par exemple 0 pour une division par zéro ;
- l'interruption NMI a pour numéro prédéfini 2 ;
- l'interruption INTR spécifie son numéro grâce aux broches $D0$ à $D7$ durant un cycle spécial de lecture du numéro.

L'adresse à laquelle le vecteur est stocké est calculée par le microprocesseur en multipliant le numéro par quatre.

Le problème de la priorité.-

5.2 Contrôleur d'interruptions programmable

Il n'est pas très facile de dessiner le circuit indiquant le numéro d'interruption matérielle ni de décider quel périphérique est prioritaire par rapport à tel autre.

Pour faciliter ce travail, *Intel* a conçu un **contrôleur d'interruptions programmable** (PIC pour *Programmable Interrupt Controller*) appelé 8259A, permettant de traiter jusqu'à huit interruptions.

5.2.1 Brochage

Le 8259A est un circuit intégré à 28 broches dont les noms sont indiqués à droite de la figure ci-dessous :

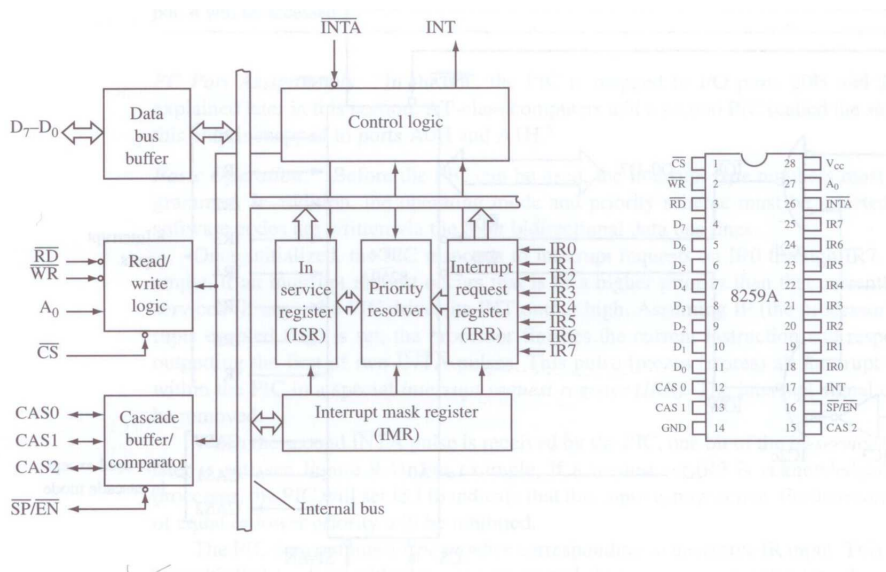


FIGURE 5.4 – Le PIC 8259

Son équivalent logique apparaît à gauche de la figure ci-dessus.

— **CAS0 - CAS1 - CAS2** (*CAS*cade interface)

Ces broches 12, 13 et 15 sont utilisées pour relier plusieurs 8259 ensemble de façon à porter le nombre d'interruptions programmables à 64 avec un mode maître/esclave.

Sur l'IBM-PC d'origine, il n'y a qu'un seul 8259 (utilisé en mode esclave) ; nous ne nous intéresserons donc qu'au mode esclave ici. Pour utiliser le 8259 en mode esclave, le contrôleur doit être programmé pour cela (nous verrons comment plus tard), les broches **CAS0** à **CAS1** sont alors ignorées.

— **SP/EN** (*Slave Programming/EN*able)

La broche 16 est un signal de sortie qui, en mode tampon, permet d'activer les tampons du bus des données du système. En mode non tampon, on a $SP = 1$ pour le maître et $SP = 0$ pour l'esclave.

— **INT**

La broche 17 est une sortie devant être reliée à la broche **INTR** du 8086.

- $\overline{\text{INTA}}$
La broche 26 est une entrée devant être reliée à la broche INTA du 8086.
- **IR0 - IR1 - IR2 - IR3 - IR4 - IR5 - IR6 - IR7** *Interrupt Request*
Les broches 18 à 25 sont utilisées comme interruptions matérielles. Lorsqu'on a un niveau haut sur l'une de ces broches, le microprocesseur va à un emplacement mémoire. Pour chacune de ces broches, il existe un emplacement dans la table des vecteurs d'interruption.
- V_{CC} et **GND**
L'alimentation électrique de + 5V se fait grâce aux broches 28 et 14.
- **D0 - D1 - D2 - D3 - D4 - D5 - D6 - D7**
Le 8259 est programmé *via* un bus des données bidirectionnel correspondant aux broches 11 à 4.
- **A0**
La broche 27 permet au 8259 d'apparaître en tant que deux ports de huit bits au microprocesseur.
- $\overline{\text{WR}}$ et $\overline{\text{RD}}$
Les broches 2 et 3 permettent de contrôler le sens des données.
- $\overline{\text{CS}}$ (*Chip Select*)
La broche d'entrée 1 permet à la sortie du décodeur d'adresse d'activer le PIC.

5.2.2 Fonctionnement du PIC

Le 8259 comporte trois registres internes, nommés **ISR** (pour *In Service Register*), **IRR** (pour *Interrupt Request Register*) et **IMR** (pour *Interrupt Mask Register*).

Avant que le PIC ne puisse être utilisé, les numéros d'interruption doivent être programmés. De plus, un mode opératoire et un protocole de priorité doivent être choisis. Ceci se fait grâce au bus des données bidirectionnel.

Une fois initialisé, le PIC répond aux requêtes d'interruption IR0 à IR7. Par exemple si une requête d'interruption survient et qu'elle est de priorité plus grande que celle qui est en train d'être effectuée, le PIC l'exécute. En supposant que l'indicateur IF est positionné, le microprocesseur termine l'instruction en cours et répond par la première des deux impulsions $\overline{\text{INTA}}$. Cette impulsion gèle (stocke) toutes les requêtes d'interruption dans le PIC dans le registre spécial IRR. Le signal d'interruption peut alors être retiré.

Lorsque la seconde impulsion $\overline{\text{INTA}}$ est reçue par le PIC, un bit du registre ISR est positionné. Par exemple si le microprocesseur accuse réception d'une requête IR3, le PIC positionne le bit IS3 de ISR pour indiquer que cette entrée est active. De plus les entrées de priorité inférieure ou égale sont inhibées.

Le PIC sort alors le **numéro de type** correspondant à l'entrée IR active. Ce nombre est multiplié par quatre par le microprocesseur et utilisé comme pointeur dans la table des vecteurs d'interruption localisée aux adresses 00000h à 003FFh.

Avant de transférer le contrôle à l'adresse du vecteur, le microprocesseur place CS, IP et les indicateurs sur la pile. La routine de service de l'interruption ISR est alors exécutée. Lorsqu'elle est terminée, l'ISR doit exécuter une commande spéciale **EOI** (pour *End-Of-Interrupt*) du PIC. Celle-ci positionne le bit correspondant du registre ISR pour l'entrée IR active. Si ceci n'était pas fait, toutes les interruptions de priorité égale ou inférieure resteraient inhibées par le PIC.

Le cycle d'interruption est terminé lorsque l'instruction IRET est exécutée. Celle-ci replace CS, IP et les indicateurs et transfère le contrôle au programme qui avait été interrompu.

5.2.3 Modes opératoires du PIC

Le 8259 peut être programmé dans l'un des six modes opératoires suivants :

- 1. **Fully Nested.**- C'est le mode par défaut du PIC, celui utilisé par l'IBM-PC. Dans ce mode, IR0 possède la priorité la plus élevée et IR7 la priorité la moins élevée.
- 2. **Special Fully Nested.**- On choisit ce mode pour obtenir un PIC maître dans un système en cascade. Ce mode est identique au mode *fully nested* mais il étend la règle de priorité aux PIC reliés en cascade. Par exemple dans le cas de la figure :

Uffenbeck, p. 394

IRQ8 a une priorité plus élevée que IRQ12 (bien qu'ils soient reliés tous les deux à IRQ2). De même IRQ12 a une priorité plus élevée que IRQ5 (car IRQ12 est relié à IRQ2).

- 3. **Nonspecific Rotating.**- Ce mode est prévu pour les systèmes ayant plusieurs sources d'interruption, toutes de même niveau de priorité. Lors d'une commande EOI, le bit IS est mis à zéro et possède alors la priorité la plus basse. La priorité des autres entrées tourne alors. La figure :

Uffenbeck, p. 398

illustre cette technique.

- 4. **Specific Rotating.**- Ce mode est analogue au mode précédent, mais la commande EOI peut indiquer le bit IS particulier à positionner et donc qui doit avoir la priorité la plus grande. La figure :

Uffenbeck, p. 399

illustre l'utilisation de ce mode.

- 5. **Special Mask.**- Comme nous l'avons vu, le PIC inhibe les requêtes d'interruption de priorité égale ou inférieure à celle en cours. Dans le mode *special mask*, toutes les interruptions sont permises hormis celles ayant le même numéro que celle qui est en cours.
- 6. **Polling.**- Dans ce mode, la sortie INT du PIC est inhibée et le 8259 est utilisé comme un *interroateur à priorité (prioritized poller)*.

5.2.4 Programmation des registres de contrôle d'initialisation du PIC

Il y a quatre **mots de contrôle d'initialisation** (ICW1 à ICW4, pour *Initialization Control Word*).

Il existe seulement une broche d'adresse (A0) pour communiquer avec le 8259. La table suivante montre comment on accède à ces quatre mots :

CS	A0	Initialisation
0	0	ICW1
0	1	ICW2, ICW3, ICW4
1	x	Le 8289 n'est pas adressé

La séquence d'initialisation doit toujours débuter par ICW1, suivie de ICW2 puis d'un dernier mot (ICW3 ou ICW4 si nécessaire). Le mot ICW1 spécifie si on a besoin de ICW3 ou de ICW4.

5.2.4.1 Initialisation de ICW1

Format.- Le format de ICW1 est le suivant :

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
A_7	A_6	A_5	1	LTIM	ADI	SNGL	IC4

- Comme on le voit, il est obtenu lorsqu'on commande le PIC avec $A0 = 0$ et $D4 = 1$. Il doit nécessairement être le premier mot d'initialisation. Les bits de commandes doivent être sur le bus des données, seuls étant pris en compte $D0$ à $D3$ et $D5$ à $D7$.
- IC4 doit être à 1 si ICW4 est nécessaire, à 0 sinon.
Ce bit doit être à 1 pour un microprocesseur 8086.
- SNGL (pour *SiNGLe*) doit être à 1 si on n'utilise qu'un seul 8259 et à 0 si on est en mode cascade. Dans le mode cascade, on aura besoin de ICW3.
- ADI (pour *call ADdress Interval*) doit être à 1 pour un intervalle de 4 et à 0 pour un intervalle de 8.
- LTIM (pour *Level-TrIggered Mode*) doit être à 1 pour un mode déclenchement (*level-triggered mode*) par niveau et à 0 pour un mode de déclenchement par impulsion (*edge-triggered mode*) pour IR0 – IR7.
- Les bits A_5 à A_7 donnent l'adresse du vecteur d'interruption. Ils ne sont utiles que dans le cas du mode du microprocesseur MCS-80/85.
Ils valent tous 0 pour le 8086.

Exemple.- Les instructions suivantes programment le 8259A en maître pour le mode 8086, à déclenchement par impulsion et comme PIC unique :

```
mov AL, 00010011b    ; declenchement par impulsion,
                    ; PIC unique, mode 80x86
out 20h, AL          ; ICW1 (A0 = 0 et D4 = 1)
```

5.2.4.2 Initialisation de ICW2

Format.- Le format de ICW2 est le suivant :

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
A_{15}/T_7	A_{14}/T_6	A_{13}/T_5	A_{12}/T_4	A_{11}/T_3	A_{10}/T_2	A_9/T_1	A_8/T_0

- On spécifie les bits A_{15} à A_8 de l'adresse du vecteur d'interruption dans le cas du mode MCS-80/85 et les bits T_7 à T_3 dans le cas du mode 8086/8088.
- Dans le cas du mode 8086/8088, le PIC donne aux trois bits T_2 , T_1 et T_0 les valeurs 000 à 111 pour les entrées IR0 à IR7 ; le numéro de type de base du PIC doit donc se terminer par 000b.

Exemple.- Déterminer le numéro de type de base pour que les entrées IRQ0 à IRQ7 correspondent aux numéros de type 08h à 0Fh. À quels emplacements mémoire le microprocesseur ira-t-il chercher les vecteurs d'interruption correspondants ?

Puisque ICW2 stocke l'adresse de base, il doit être programmé avec 08h. La table suivante donne les numéros de type de sortie pour chaque entrée IRQ et l'emplacement du vecteur d'interruption correspondant.

Entrée	Numéro de type	Emplacement du vecteur d'interruption
IRQ0	08h	0000:0020-23h
IRQ1	09h	0000:0024-27h
IRQ2	0Ah	0000:0028-2Bh
IRQ3	0Bh	0000:002C-2Fh
IRQ4	0Ch	0000:0030-33h
IRQ5	0Dh	0000:0034-37h
IRQ6	0Eh	0000:0038-3Bh
IRQ7	0Fh	0000:003C-3Fh

5.2.4.3 Initialisation de ICW3

Si le bit D1 de ICW1 est à 0, le mode en cascade est spécifié. Dans ce cas, une seconde écriture du PIC avec $A_0 = 1$ est interprétée comme ICW3.

Ce mot ne nous intéresse pas ici puisqu'il n'y a qu'un 8259 sur le premier IBM-PC.

5.2.4.4 Initialisation de ICW4

Format.- Si le bit D0 de ICW1 est à 0, le mode 8086 est indiqué. Dans ce cas, une autre écriture du PIC avec A0 = 1 (seconde ou troisième, suivant la valeur du bit D1) est interprétée comme ICW4. Son format est le suivant.

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	0	SFNM	BUF	M/S	AEOI	μ PM

- μ PM indique le mode du microprocesseur, soit MCS-80/85 si D0 = 0, soit 8086/8088 si D0 = 1.
- Le bit D1 (AEOI pour *Automatic EOI*) à 1 active automatiquement l'instruction EOI utilisée dans les modes *fully nested* et *automatic rotating priority*. Ceci évite de placer une instruction EOI avant l'instruction IRET dans le code de l'ISR.
Lorsque D1 vaut 0, EOI doit être effectué en utilisant l'OCW (*Operation Command Word*).
- Les bits D2 (M/S pour *Master/Slave*) et D3 (BUF pour *BUFfer*) spécifient si le PIC est le maître ou l'esclave dans un environnement de CPU tamponné et contrôle la sortie $\overline{SP/EN}$.
- Le bit D4 (SFNM pour *Special Fully Nested Mode*) sélectionne ou non le mode *special fully*.

Exemple.-

5.2.5 Programmation des registres de contrôle des opérations du PIC

Que se passe-t-il si plusieurs broches parmi IR0–IR7 sont activées au même moment ? Les **mots de contrôle des opérations (OCW pour *Operation Control Words*)** y répondent. Une fois le PIC initialisé avec les ICW, les données en écriture suivantes vers le 8289A sont interprétées comme des OCW. Ces octets OCW1, OCW2 et OCW3 spécifient les modes de priorité de rotation, le mode de masquage spécial, le mode d'interrogation, le masque d'interruption et les commandes EOI.

Comme pour les ICW, la table suivante montre comment on accède à ces trois mots :

CS	A0	Operation Command Word
0	0	OCW2, OCW3
0	1	OCW1
1	x	Le 8289 n'est pas adressé

5.2.5.1 Initialisation de OCW1 (masquage)

Format.- On peut lire ou écrire dans ce registre. Le format est le suivant :

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
M7	M6	M5	M4	M3	M2	M1	M0

où $M_i = 1$ indique que l'entrée IR est masquée, c'est-à-dire qu'il n'y aura pas de réponse à une requête de sa part.

Exemple.- Pour l'IBM-PC, IRQ0 est associé au minuteur, pilotant en particulier l'horloge interne.

En utilisant `debug`, écrivons un programme `TIMEROFF.COM` permettant de désactiver l'interruption du minuteur du PC :

```
>debug
-a 100
1785:0100 MOV AL,01
1785:0102 OUT 21,AL
1785:0104 INT 20
1785:0106
-n timeroff.com
-r cx
cx 0000
: 6
-w
Ecriture de 00006 octets
```

Écrivons de même un programme `TIMERON.COM` permettant d'activer l'interruption du minuteur du PC :

```
1785:0100 B000 MOV AL,00 ; active IRQ0-IRQ7
1785:0102 E621 OUT 21,AL ; sort OCW1
1785:0104 CD20 INT 20 ; retour a MS-DOS
```

Écrivons d'autre part, grâce à un éditeur de texte, un fichier `TEST.BAT` contenant :

```
@echo.off
:start
time
cls
goto start
```

Pour tester les programmes, il suffit de taper `TIMEROFF` puis `TEST`, en mode MS-DOS et non dans une fenêtre MS-DOS de Windows. On voit alors l'affichage de l'heure sans que les secondes ne changent, puisque DOS ne reçoit plus les tics d'horloge.

On arrête en utilisant `CTRL-C`.

Entrer alors `TIMERON` puis `TEST`. Cette fois-ci, les secondes devraient être incrémentées.

5.2.5.2 Initialisation de OCW2

Format.- On écrit dans OCW2 pour spécifier la commande EOI au PIC. Son format est le suivant :

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
R	SL	EOI	0	0	L2	L1	L0

Remarquons que le bit D_4 est nul, ainsi le PIC ne confondra-t-il pas cet octet avec ICW1 (pour lequel on a $DA = 1$). Les commandes qui peuvent être écrites dans ce registre sont données par le tableau suivant :

R	SL	EOI	Commande	Description
0	0	1	EOI non spécifique	Utilisée dans le mode <i>fully nested</i> pour mettre le bit IS à 1 ; si le bit 1 de ICW4 est à 1, cette commande est exécutée automatiquement lors de INTA.
0	1	1	EOI spécifique	Utilisée pour mettre à 1 un bit donné de IS, spécifié par D_0 – D_2 .
1	0	1	Rotate on EOI non spécifique	Utilisé pour opérer en mode rotation non particulière ; met à 1 le bit IS de numéro le plus bas.
1	0	0	Set rotate in auto EOI mode	Si le bit 1 de ICW4 est à 1, le PIC effectuera automatiquement une rotation sur la commande EOI non particulière lors des cycles INTA
0	0	0	Clear rotate in auto EOI mode	Utilisé pour annuler le mode d'auto-rotation
1	1	1	Rotate on specific EOI command	Utilisé pour opérer en mode de rotation particulière ; les bits D_0 – D_2 spécifient le bit IS à mettre à 1, donc de priorité la plus basse
1	1	0	Set priority command	Utilisé pour assigner une entrée IR particulière de priorité la plus basse

Exemple.- Décrivons la forme de la routine de service d'interruption pour que le PIC soit programmé dans le mode *fully nested* :

```

ISR PROC FAR
    ... ; debut de l'ISR
    ... ; fin de l'ISR
    MOV AL,00100000b ; EOI non particulier
    OUT 20h,AL ; OCW2
    IRET
ISR ENDP

```

Remarque.- On n'a pas besoin de se préoccuper du bit IS lorsqu'on opère en mode *fully nested*, puisque l'instruction EOI non particulière remplace toujours le bit IS de numéro le plus bas. Dans le mode *fully nested*, ceci correspondra toujours à la routine en cours.

5.2.5.3 Initialisation de OCW3

Accès.- On écrit sur OCW3 en utilisant le niveau bas de l'adresse du port ($A0 = 0$). On le distingue de ICW1 et de OCW2 par les bits D3 et D4, qui doivent être à 1 et à 0 respectivement.

Format.- Le format de OCW3 est le suivant :

A_0	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
0	0	ESMM	SMM	0	1	P	RR	RIS

- Les bits D5 et D6 permettent de programmer le mode de masquage spécial (SMM pour *Special Mask Mode* et ESMM pour *Enable Special Mask Mode*), conformément au tableau suivant :

SMM	ESMM	Action
1	1	active le masquage spécial
0	1	désactive le masquage spécial
1	0	pas d'action
0	0	pas d'action

- On utilise le bit D2 pour sélectionner le mode d'interrogation. Les bits D1 et D0 permettent de lire IRR ou IS conformément au tableau suivant :

RIS	RR	Action
1	1	Lire le registre IS lors de la prochaine impulsion \overline{RD}
0	1	Lire le registre IR lors de la prochaine impulsion \overline{RD}
1	0	pas d'action
0	0	pas d'action

Exemple.- Considérons les instructions suivantes se trouvant dans la routine de service du port COM1 du PC utilisant IRQ4 :

```
MOV AL,00010000b ; masque IRQ4
OUT 21h,AL       ; OCW1 (IMR)
MOV AL,01101000b ; mode de masquage special
OUT 20h,AL       ; OCW3
```

En se masquant lui-même et en sélectionnant le mode de masquage spécial, les interruptions IRQ5 à IRQ7 ne seront plus acceptées par le PIC (ainsi que celles de priorité plus élevée IRQ0 à IRQ3).

Exemple.- Écrivons une routine de service vérifiant que l'interruption de l'entrée IRQ7 du PC est « légitime ».

Rappelons que le PIC ne prend pas connaissance de IR7 si la requête d'interruption n'est pas maintenue jusqu'à accusé de réception par le microprocesseur. Le programme est le suivant :

```
ISR7  PROC FAR
      MOV  AL,00001011b ; lire lors de la prochaine IO
      OUT  20h,AL      ; OCW3
      IN   AL,20h      ; lire le statut IS
      TEST AL,80h      ; IS7 positionne ?
      JZ   FALSE      ; non : entree erronee
                        ; oui : processus d'interruption correcte
      ...
FALSE: IRET
ISR7  ENDP
```

5.2.6 Interfaçage

Nous avons vu que les adresses des ports du PIC doivent être consécutives.

Exemple.- Déterminer les adresses des ports d'entrées/sorties du PIC relié de la façon indiquée sur la figure ci-dessous :

Uffenbeck, p. 392

5.3 Cas de l'IBM-PC

Pour le PC et le PC-XT, IBM utilise un seul PIC dont les adresses d'entrées-sorties sont 20h et 21h. Les requêtes d'interruptions sont les suivantes :

IRQ0	Minuteur 0
IRQ1	Clavier
IRQ2	réservé
IRQ3	COM2
IRQ4	COM1
IRQ5	LPT2
IRQ6	Contrôleur du lecteur de disquettes
IRQ7	LPT1

Le PIC opère en mode *fully nested*.

5.4 Commentaire du BIOS : initialisation du PIC

Des noms sont donnés aux divers ports du 8259 au début du code du BIOS :

```
0020          19  INTA00    EQU    20H      ; 8259 PORT
0021          20  INTA01    EQU    21H      ; 8259 PORT
0020          21  EOI       EQU    20H
```

Le PIC est initialisé lors du démarrage. Le code afférent du BIOS commence ligne 562 :

```
562 ;-----
563 ;      INITIALIZE THE 8259 INTERRUPT CONTROLLER CHIP ;
564 ;-----
E1CE B013    565 C25:  MOV     AL,13H          ; ICW1 - EDGE, SNGL, ICW4
E1D0 E620    566      OUT     INTA00,AL
E1D2 B808    567      MOV     AL,8            ; SETUP ICW2 - INT TYPE 8 (8-F)
E1D4 E621    568      OUT     INTA01,AL
E1D6 B009    569      MOV     AL,9            ; SETUP ICW4 - BUFFRO,8086 MODE
E1D8 E621    570      OUT     INTA01,AL
E1DA B0FF    571      MOV     AL,0FFH         ; MASK ALL INTS. OFF
E1DC E621    572      OUT     INTA01,AL     ; (VIDEO ROUTINE ENABLE INTS.)
573
```

Commentaires.- 1°) On commence par initialiser ICW1 (lignes 565 et 566) en envoyant un octet de contrôle au port 20h. L'octet de contrôle est 13h = 0001 0011b, c'est-à-dire que IC4 = 1 pour dire qu'on a besoin de ICW4, SNGL = 1 pour dire qu'on n'utilise qu'un seul 8259, ADI = 0 pour spécifier un intervalle de 8, LTIM = 0 pour un mode de déclenchement par impulsion, D4 = 1 puisqu'il s'agit de ICW1, D0 = D1 = D3 = 0 mais ces bits n'ont pas d'importance pour un 8086.

- 2°) On envoie ensuite un octet à ICW2 (lignes 567 et 568). L'octet de contrôle stocke l'adresse de base ; nous avons déjà vu qu'il doit valoir 8h pour que les interruptions IR0 - IR7 soient 8h - Fh.

- 3°) Puisque ICW1 dit qu'on n'a pas besoin de ICW3 mais de ICW4, on envoie ensuite un octet à ICW4 (lignes 569 et 570). L'octet de contrôle est 9h = 0000 1001b, c'est-à-dire que μ PM = 1 pour un 8086, AEOI = 0 pour un EOI non automatique, M/S = 0 pour esclave (il n'y a qu'un seul 8259), BUF = 1 pour une sortie avec tampon, SFNM = 0 pour ne pas être dans le mode *special fully*, D7 = D6 = D5 = 0 comme toujours pour ICW4.

- 4°) On envoie ensuite le mot OCW1 de masquage (lignes 571 et 572). L'octet de masquage est FFh, c'est-à-dire que l'on masque toutes les interruptions matérielles.