Gestion des Demandes d'Interruptions Matérielles
Cas du Contrôleur Programmable d'Interruptions
(PIC 8259 A)

Resp. Dr. Mohamed Feredj
Courriel : archiFeredj@gmail.com

0) Introduction

Comment identifier la source d'une demande d'interruption ?
Il existe différentes méthodes de détection de la source d'une interruption :

a) Par Polling (par sondage) : Solution logicielle
Cette technique utilise un programme défini au préalable pour de déterminer les priorités des interruptions.

- Est adéquate pour un petit nombre de sources d'interruption, sinon le temps d'identification des sources devient très important.

b) Par Daisy Chain (Interruption chaînées ou priorités chaînées) : Solution matérielle
Cette technique utilise un composant matériel (contenant des portes logiques) pour identifier la source d'interruption.

Avantages :
- Pas de programmation pour déterminer la priorité d'une interruption ;
- Réalisation très simple : Nécessite des bloc logiques très simples à concevoir ;
- Ajout et suppression d'autres sources d'interruption est très simple.

Inconvénients :
- On ne peut pas modifier l'ordre de priorité des interruptions par programmation.

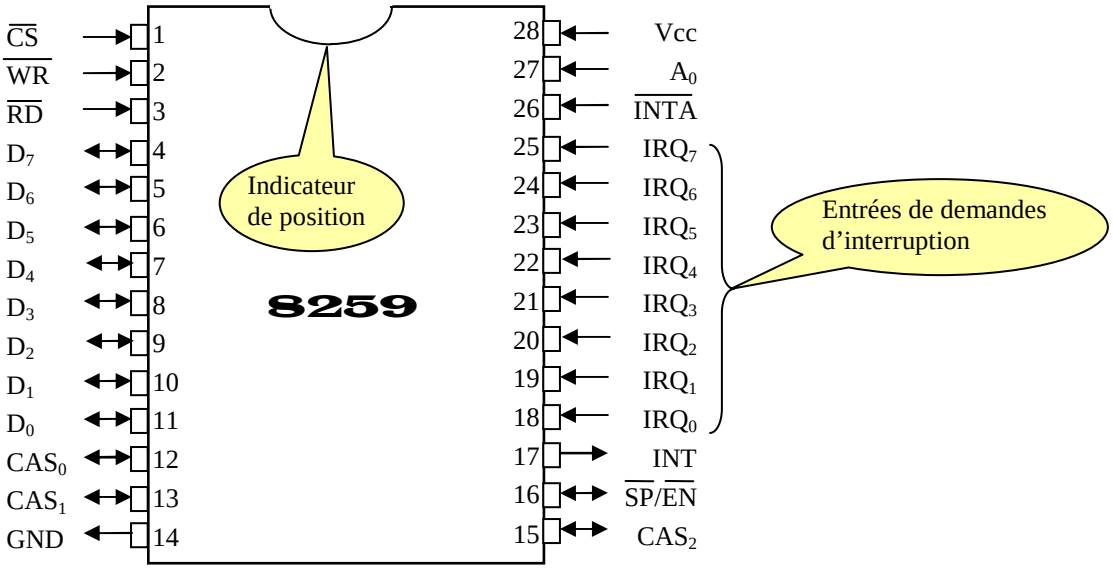
Dans ce chapitre, nous présentons le PIC (Programmable Interrupt Controller) 8259 d'Intel. C

1) Définition

Le PIC est un circuit qui décharge le µP de la gestion des demandes d'interruption matérielle provenant de l'extérieur. En effet, Sa fonction se résume par :

1. Réception des demandes d'interruption sur ses lignes d'entrée (IRQ0, … , IRQ7) ;
2. Détermination de l'interruption la plus prioritaire ;
3. Informer le µP.

Le PIC gère au maximum 8 demandes d'interruption. Cependant, il est cascadable, ce qui permet la gestion de 64 demandes. C'est-à-dire un PIC maître peut recevoir les sorties de 8 autres PICs esclaves.



2) Description des signaux (broches) du PIC

VCC : Broche d'alimentation électrique à 5 volts ;

GND : La masse ;

$\overline{CS}$ ***(Chip Select) :*** Boîtier Sélectionne qui réveille le circuit (PIC). Donc, si le µP veut s'adresser au PIC, il doit envoyer sur cette broche un 0 ;

Donc, $\overline{CS} = \begin{cases} 0 : \text{PIC sélectionné} \\ 1 : \text{PIC non sélectionné} \end{cases}$

Si par exemple on a CS0 et $\overline{CS}1$ ➔ on doit envoyer 1 et 0 pour sélectionner un Circuit

$\overline{WR}$ ***(Write) :*** Ecriture (ligne de commande) ;

$\overline{RD}$ ***(Read) :*** Lecture (ligne de commande) ;

D0-D7 : Bus de données bidirectionnel ;

Exemple : Si @ du PIC est 20H alors
OUT 20H, AL ; ➔ $\overline{CS} = 0$ et $\overline{WR} = 0$
Si AL = 03H ➔ D0=D1=1 et D2=0…=D7=0;

CAS0-CAS2 : Pour cascader les PICs et servent comme moyen de communication ;

IRQ0-IRQ7 : Pour connecter 8 périphériques pouvant demander des interruptions. Si on veut plus on doit connecter sur chaque IRQi (maximum 8) un PIC esclave.
Le PIC maître peut accéder aux PICs esclaves par les lignes CAS0-CAS2. Exemple : 000=PIC0, 001=PIC1, … 111=PIC7.

$\overline{SP} / \overline{EN}$ **:** (SP=Slave Program/EN=Enable Buffer) : Est une broche à double fonction :

1) Si $\overline{SP} / \overline{EN}$ programmée en entrée : $\begin{cases} 0 : \text{PIC esclave} \\ 1 : \text{PIC maître} \end{cases}$
Donc, on branche $\overline{SP} / \overline{EN}$ à la masse ou à 5 volts

2) Si $\overline{SP} / \overline{EN}$ programmée en sortie, cela pour commander des transmetteurs de type 8286

INT : C'est la ligne par laquelle le PIC transmet la demande d'interruption au µP, sur l'entrée INTR ;

$\overline{INTR}$ **:** Permet au PIC de recevoir les acquittements du µP. Dés que ce signal est reçu, le PIC saura qu'il doit placer sur le BD les informations servant au µP à se brancher à la table des vecteurs.

A0 : Est utilisée en conjonction avec $\overline{CS}, \overline{RD}$ et $\overline{WR}$. Cette broche est connectée sur A0 du BA et permet de sélectionner un des registres internes du PIC.

Exemple : La première utilisation des PICs en cascade est réalisée avec le µP 80286 :

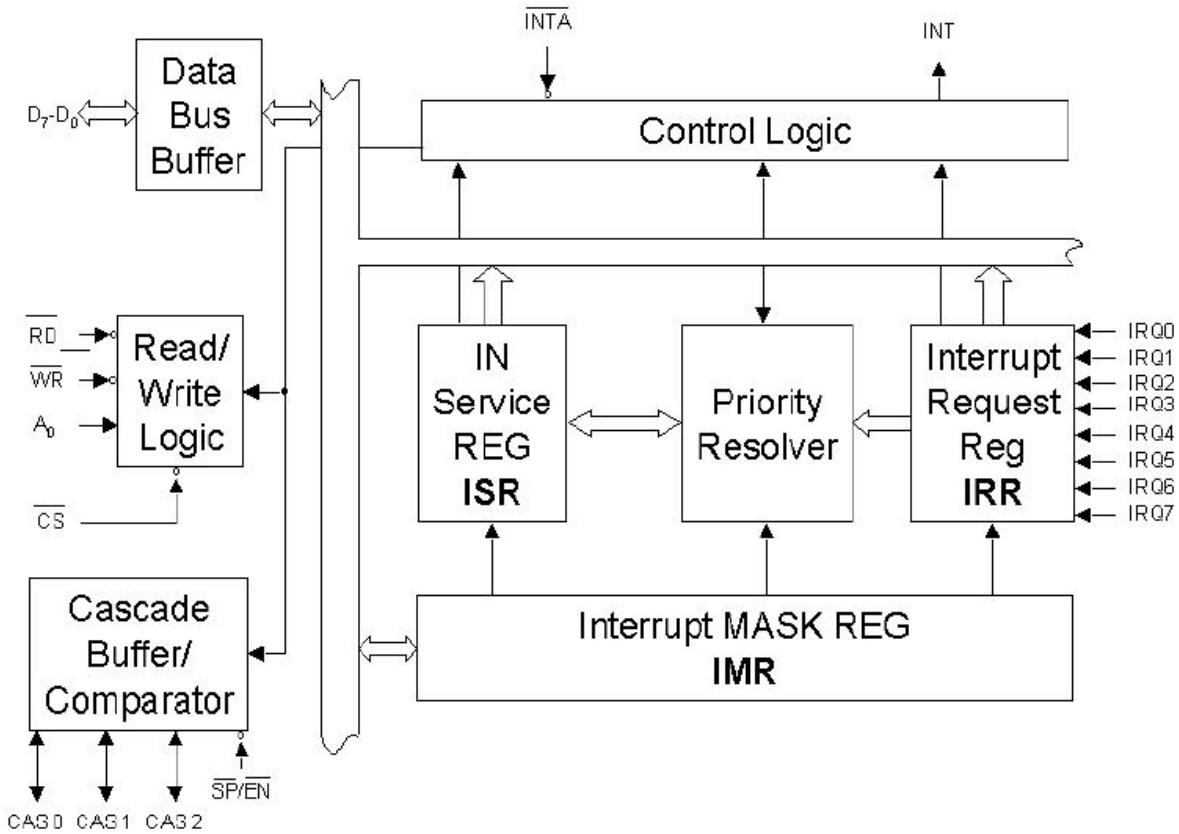
PIC Maître :
- IRQ0 → Timer
- IRQ1 → Clavier
- IRQ2 → PIC esclave
- IRQ3 → COM2

IRQ4 → COM1
IRQ5 → LPT2
IRQ6 → Lecteur Disquette
IRQ7 → LPT1

PIC Esclave :

IRQ23 → Coprocesseur Mathématique
IRQ24 → Disque Dur

3) Architecture Interne du PIC



3.1) Registres Internes du PIC :

1. *IRR (Interrupt Request Register)* : Registre de demandes d'interruption ;
2. *ISR(Interrupt Service Register)* : Registre d'Interruption en Service ;
3. *PR (Priority Resolver)* : Le résolveur de priorité ;
4. *IMR (Interrupt Mask Register)* : Registre de Masquage des Interruptions ;

5. RWL (Read/Write Logic) : La logique de Lecture/Ecriture. Ce bloc contient :
   a) 4 ICW(1-4): Initialization Command Words (Mots de Commande d'Initialisation)
   b) 3 OCW(1-3) : Operation Command Words (Mots de Commande des Opérations)

IRR (Interrupt Request Register) : (Registre de demandes d'interruption)

Est un registre sur 8 bits qui reçoit les 8 lignes de demandes d'interruption et mémorise chacune d'elles (demandes interruption), en positionnant à 1 le bit correspondant :

$$IRRi = \begin{cases} 1 : \text{Présence demande d'interruption sur IRQi} \\ 0 : \text{Pas de demande d'interruption sur IRQi} \end{cases}$$

ISR (Interrupt Service Register) : (Registre d'Interruption en Service)

Est un registre sur 8bits marquant les interruptions en service, en positionnant à 1 le bit correspondant :

$$ISRi = \begin{cases} 1 : \text{Interruption de IRQi en service} \\ 0 : \text{Sinon} \end{cases}$$

<u>Remarque :</u> *Une fois qu'une interruption est mise en service, son bit sur IRR passe automatiquement à zéro.*

<u>*Exemple :*</u>

ISR7	ISR6	ISR5	ISR4	ISR3	ISR2	ISR1	ISR0
0	0	1	0	0	1	0	0

Les interruptions correspondent aux IRQ5 et IRQ2 sont en cours de traitement

IMR (Interrupt Mask Register) : Registre de Masquage des Interruptions

Est un registre sur 8 bits permettant de masquer les demandes d'interruption :

$$IMRi = \begin{cases} 1 : \text{Demandes sur IRQi sont masquées} \\ 0 : \text{Sinon} \end{cases}$$

PR (Priority Resolver) : (Résolveur de Priorité)

Il sélectionne parmi les demandes d'interruption non masquées mémorisées dans IRR quelle est celle possédant la plus haute priorité.

Remarque : Le µP envoie toujours 2 signaux consécutifs sur $\overline{\text{INTA}}$ *:*

1. *Dés la réception du 1er Signal, le PIC effectue :*
   ➔ *ISRi = 1 (au front descendant) et IRRi = 0 (au front montant)*
2. *Dés la réception du 2ème Signal, le PIC effectue :*
   ➔ *Emettre le Type(n°) d'It sur son bus de données au µP (au front descendant) et ISRi = 0 (au front montant). Pour la remise de ISRi à 0, il est préférable de le faire dans la routine d'interruption juste avant IRET.*

4) Programmation du PIC

Programmer le PIC, c'est de lui transmettre par le µP une séquence d'octets qui sera rangée dans les mots ICWi (i=1-4) et OCWi (i=1-3) :

ICWi pour l'initialisation ;
OCWi pour le contrôle de certaines fonctionnalités.

Remarque : La différence entre les ICWi et les OCWi est fondamentale :

1. *Les ICW précèdent tout usage du PIC ;*
2. *Les OCW peuvent intervenir au cours du déroulement du programme.*

4.1) La séquence d'initialisation :

L'initialisation du PIC consiste à émettre une séquence de 4 octets. Les 2 premiers sont obligatoires dans tous les cas.

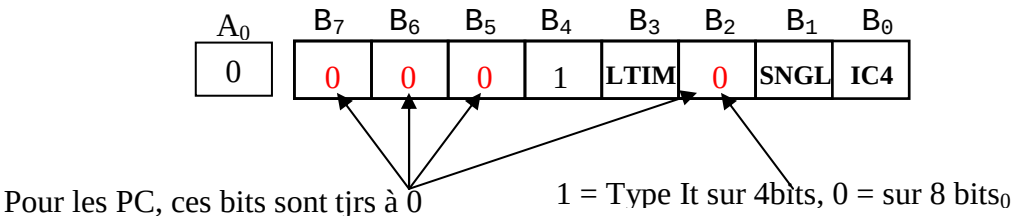
Remarque :

1. *L'ordre d'initialisation des ICWi est important. Par contre celui des OCWi non ;*

2. *L'initialisation des ICWi se fait une fois (au démarrage de la machine) et on ne peut pas le faire car le BIOS nous précède. Pour le réinitialiser, on doit passer au mode protégé.*

ICW1

- Le PIC est il seul ou cascadé?
- Détection des demandes d'It se fait sur les fronts ou sur les niveaux ?
- ICW4 sera il utilisé ou non ?



$$IC4 = \begin{cases} 1 : \text{Présence avec ICW4} \\ 0 : \text{Absence de ICW4} \end{cases}$$

$$\texttt{SNGL(Single)} = \begin{cases} \texttt{1:PIC seul} \\ \texttt{0:PIC cascadé} \end{cases}$$

$$\text{LTIM (Level Triggered Input Mode)} = \begin{cases} 1 : \text{Déclenchement par niveau sur IRQi} \\ 0 : \text{Déclenchement par front sur IRQi} \end{cases}$$

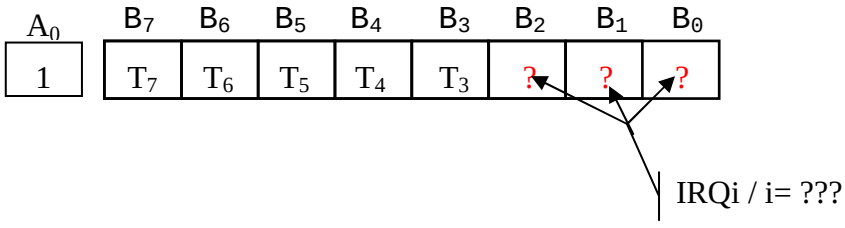
Exemple :

```
MOV AL, 11h
OUT  20H, AL
```

ICW2

- Comment va-t-on constituer le n° (type) d'It pour accéder à la table de vecteurs ?

ICW2 permet de générer le N° de l'interruption correspond à la ligne IRQi qui a provoqué l'interruption.



En, effet, le PIC positionne que les bits B0, B1 et B2 du ICW2 correspondent à l'IRQi. Cependant, les bits T3-T7 correspondent aux 5 derniers bits du N° de l'It sont fixés par l'utilisateur.

Exemple :

```
MOV AL, 10h
OUT  21H, AL
```

ICW3

- Si le PIC est maître, comment peut-on savoir sur quelle IRQi les PICs esclaves sont connectés?
- Si le PIC est esclave, comment peut-on savoir sur quelle IRQi du maître est connecté ?

1) ICW3 du PIC Maître

A0	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
1	S_7	S_6	S_5	S_4	S_3	S_2	S_1	S_0

$$Si = \begin{cases} Si = 1 : \text{Un PIC esclave est connecté à IRQi} \\ Si = 0 : \text{Sinon} \end{cases}$$

1) ICW3 du PIC Esclave

A0	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
1	0	0	0	0	0	ID_2	ID_1	ID_0

ID2ID1ID0 = Code d'identification du PIC esclave. Ce code (entre 0 et 7) représente le numéro de l'IRQ du PIC maître sur laquelle est connecté le PIC esclave.

Remarque :

Le PIC esclave sera activé dés qu'il reçoit ce code sur ses lignes CAS0-CAS2.

ICW4

- Avec quel µP fonctionne le PIC ?
- Comment sera géré le signal de fin d'It ?
- S'agit-t-il d'un maître ou d'un esclave ?
- Comment est programmée la ligne $\overline{SP}/\overline{EN}$?

A0	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
1	0	0	0	SFNM	BUF	M/S	AEOI	µP

$$\mu P = \begin{cases} 1 : \mu P\ 8086 \text{ ou plus} \\ 0 : 8080/8085 \end{cases}$$

$$\text{AEOI} = \begin{cases} 1 : \text{Fin d'interruption automatique} \\ 0 : \text{Fin d'interruption normal} \end{cases}$$

$$\text{BUF} = \begin{cases} 1 : \overline{\text{SP}}/\overline{\text{EN}} \text{ en sortie (commande des transmetteurs)} \\ 0 : \overline{\text{SP}}/\overline{\text{EN}} \text{ en entrée (selectionne le mode maitre/esclave)} \end{cases}$$

$$\text{M/S (avec BUF = 1)} = \begin{cases} 1 : \text{PIC maitre} \\ 0 : \text{PIC esclave} \end{cases}$$

avec BUF = 0 ==> M/S est ignoré

$$\text{SFNM(Special Fully Nested Mode} = \begin{cases} 1 : \text{Mode emboitable complet} \\ 0 : \text{Mode emboitable partiel} \end{cases}$$

Remarque :

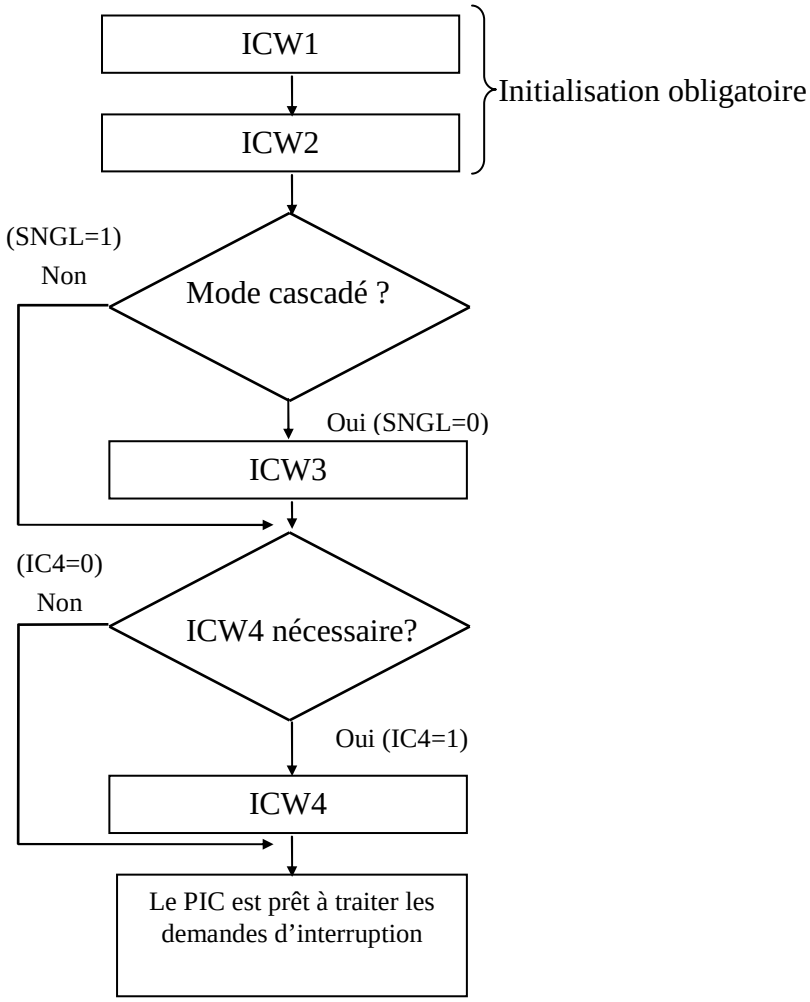
1) ***SFNM*** *est utile dans le cas d'existence de PIC esclave. Donc,*

- SFNM = 1 ➔ Ordre de priorité des Its du PIC esclave est géré par le PIC

maître. ➔ *PIC maître accepte plus d'une demande d'Its de l'esclave*

- SFNM = 0 ➔ *PIC maître accepte une seule demande d'Its de l'esclave à la fois*

Séquence d'initialisation :



<u>Exemple :</u>

Soit 2 PICs : PIC maître a les adresses 20H et 21H et PIC esclave a les adresses 0A0H et 0A1H. Le PIC esclave est connecté sur la ligne IRQ2 du PIC maître.
La détection des demandes d'Its se fait par front montant pour les deux PICs.

Ecrire la séquence d'instructions assembleur permettant l'initialisation des 2 Pics.

4.2) La manipulation des mots de commande d'opération (OCW) :

OCW1 :

Ce registre permet de positionner les bits du registre IMR

➔ Pour masquer démasquer les interruptions.

A_0	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
1	M_7	M_6	M_5	M_4	M_3	M_2	M_1	M_0

Avec,

$$Mi = \begin{cases} 1 : \text{Masquer les deemandes d'interruptions sur IRQi} \\ 0 : \text{Accepter les demandes d'interruptions sur IRQi} \end{cases}$$

OCW2 :

Ce registre permet :

- De positionner le ISRi à 0 dans le cas où IRET ne peut pas le faire ;
- De traiter l'ordre des priorités des Its ;

A_0	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
0	R	SL	EOI	0	**0**	L_2	L_1	L_0

Les bits :

R = Rotate : Pour commander les rotations
SL = Select Level : Si SL = 1 ➔ L2L1L0 sont pris en compte
EOI = End Of Interruption, Définit le mode de fin d'interruption

Sont utilisés ensemble.

L2L1L0 : indiquent le niveau d'interruption ciblé par l'opération. Ils sont actifs si le bit SL = 1

Voici toutes les combinaisons possibles :

	R	SL	EOI	DESCRIPTION
Fin interruption	0	0	1	Veut dire mettre le bit ISRi de la routine la plus prioritaire à 0 EOI non spécifique (l'ordre de priorité est le même)
	0	1	1	EOI spécifique (mettre le ISR de la routine du niveau L2L1L0 à 0)
Rotation Automatique	1	0	1	Mettre ISRi à 0 et lui donner la priorité la plus faible (Rotation de priorités sur EOI non spécifique)
	1	0	0	Rotation de priorité sur chaque EOI automatique
	0	0	0	Arrêt de rotation sur les EOI automatiques (EOI sans changement d'ordre des priorités)
Rotation spécifique	1	1	1	Mettre ISR de l'interruption du niveau L2L1L0 à 0 et lui donner la priorité la plus faible. (Rotation sur EOI spécifique)
	1	1	0	Etablissement d'un ordre de priorité
	0	1	0	Pas d'opération

Les plus utilisés

OCW3 :

Il permet :

- La lecture des registre IRR et ISR
- L'établissement du mode Polling
- L'établissement du mode spécial de masquage

A_0	B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
0	0	ESMM	SMM	0	**1**	P	RR	RIS

<u>Remarque :</u>

1. *C'est le bit B3 qui fait la distinction entre OCW2 et OCW3. Donc*
   - *B_3=1 ➔ OCW3*
   - *B_3=0 ➔ OCW2*

RR (Read Register Command) :

Si RR = 1 ➔ La lecture du registre IRR ou ISR est autorisée.
Le choix du registre à lire dépend de la valeur du bit RIS

RIS (Read In Service) :

Ce bit est utilisé pour choisir le registre ISR ou IRR.

$$RIS = \begin{cases} 1 : \text{ISR est sélectionné} \\ 0 : \text{IRR est sélectionné} \end{cases}$$

Voici les combinaisons possibles de RR et RIS :

RR	RIS	Description
0	0	Pas de lecture
0	1	Pas de lecture
1	0	Lecture IRR
1	1	Lecture ISR

Exemple : Pour lire ISR

```
MOV AL, 0BH ;  Pour lire ISR
OUT 20H, AL
IN AL, 20H ;  Lecture de ISR
```

Remarque :

1. *Si $\overline{RD}$ = 0, $\overline{CS}$ = 0 et A0 = 0 alors le PIC renvoie le contenu du registre sélectionné (IRR ou ISR) sur le BD ;*
2. *Un choix du registre à lire restera effectif jusqu'à l'envoi d'un autre mot de commande d'opération.*

P (Poll Command) : Pour qu'elle prenne effet, il faut que RR soit à 0

C'est une commande qui permet la prise en charge des interruptions par balayage. Dans ce mode, c'est le µP qui a l'initiative de déclencher les interruptions, celles-ci sont traitées par le PIC mais il ne peut interrompre le microprocesseur (Bit IF à 0).

Ainsi à l'initiative du microprocesseur, un mot de commande OCW3, ayant **P = 1**, est émis, le PIC traite alors la prochaine lecture ($\overline{RD}$ =0 et $\overline{CS}$=0) comme une reconnaissance d'interruption. Durant cette lecture, le PIC envoi sur le bus de données le mot suivant :

B_7	B_6	B_5	B_4	B_3	B_2	B_1	B_0
I	X	X	X	X	W_2	W_1	W_0

Avec ,

$$I = \begin{cases} 1 : \text{Présence d'une demande d'interruption} \\ 0 : \text{Pas de demande d'interruption} \end{cases}$$

Et

W2W1W0 = N° de l'IRQ

<u>Remarque :</u>

1. *Pour ce mode, on peut avoir plusieurs PIC connectés (mais non cascadés) autant qu'adresses existes ;*
2. *On peut gérer plus de 64 niveaux d'interruption.*

ESMM (Enable SMM) :

Si ESMM = 1 alors prendre en compte le bit **SMM**
Sinon SMM est ignoré.

SMM (Special Mask Mode) : Mode Masque Spécial d'interruption

Si ESMM = 1 ➔ SMM prend effet.

En effet, les bits sont utilisés ensemble.

Voici toutes les combinaisons possibles :

ESMM	SMM	Description
1	0	Annuler le masque spécial d'interruption
1	1	Etablir le masque spécial d'interruption
0	0	Aucun effet
0	1	Aucun effet

Ça sert à quoi le mode masque spécial ?

Permet de résoudre le problème d'occupation abusive du microprocesseur par une routine d'interruption particulière.

Solution ➔ le SMM sert à autoriser les interruptions de tout niveau différent de celui de l'interruption en service à intervenir. Ainsi, ce ne seront plus seulement les interruptions les plus prioritaires qui seront prises en compte mais également les moins prioritaires.

Comment établir le mode masque spécial ?

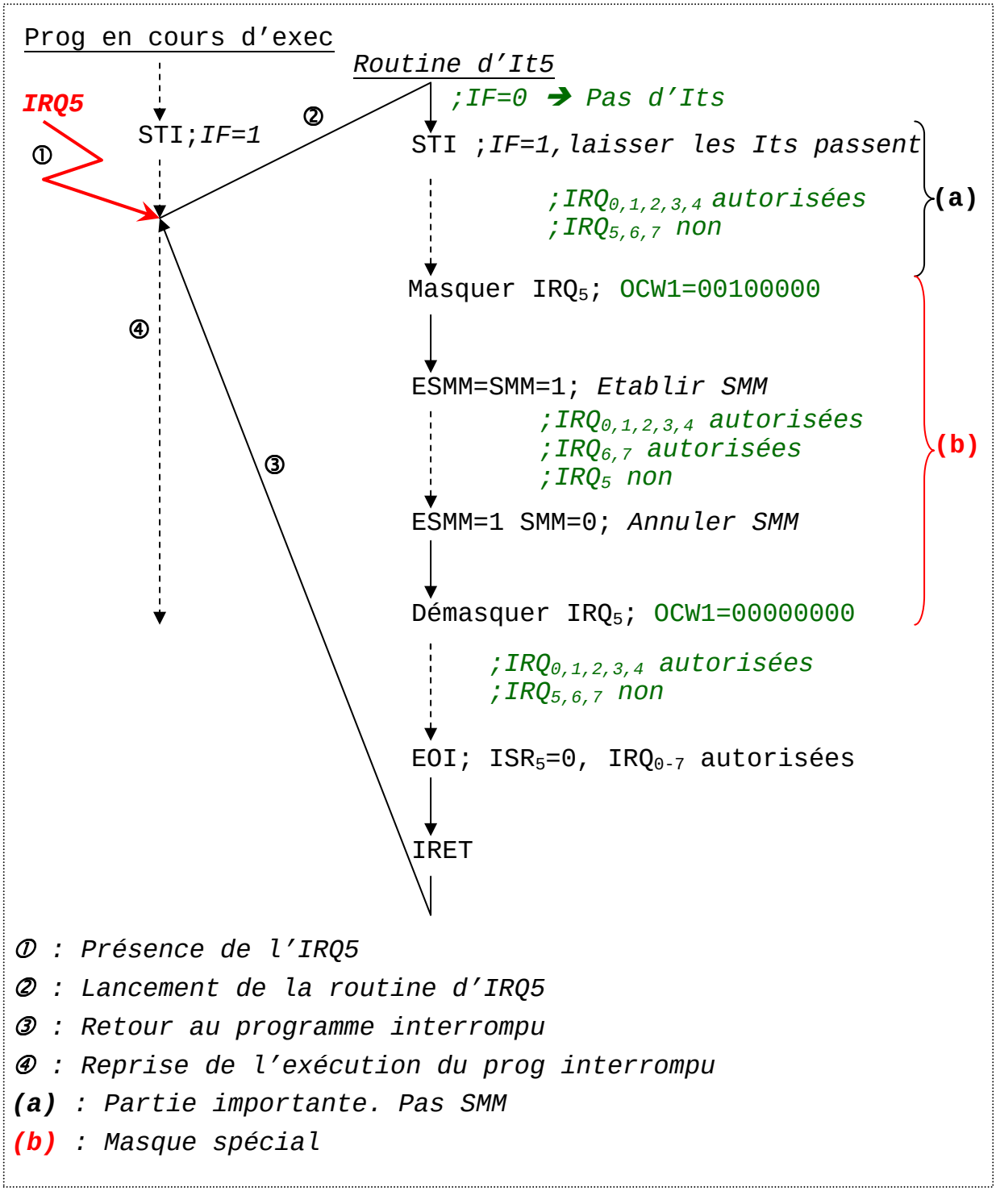
Il faut exécuter la séquence d'opérations suivante:

- Masquer uniquement le bit correspondant à la routine d'interruption en cours d'exécution. Pour cela, On doit utiliser OCW1 ;
- Mettre le PIC en mode spécial de masquage à l'aide du registre OCW3 ;
- Autoriser les interruptions (IF=1).

Remarque : *Juste avant la fin de la routine d'interruption, il faut :*

1. *Annuler le mode masque spécial*

Exemple :



① : Présence de l'IRQ5

② : Lancement de la routine d'IRQ5

③ : Retour au programme interrompu

④ : Reprise de l'exécution du prog interrompu

***(a)** : Partie importante. Pas SMM*

***(b)** : Masque spécial*