

Chapitre 5 : Les interruptions

1 Définition d'une interruption

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

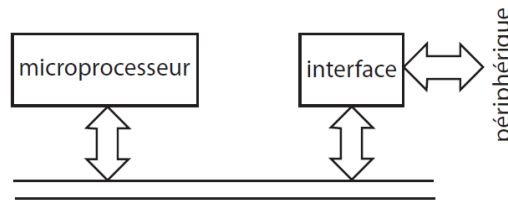


Figure 5.1. Échange entre microprocesseur et périphérique

Il y a deux façons 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)

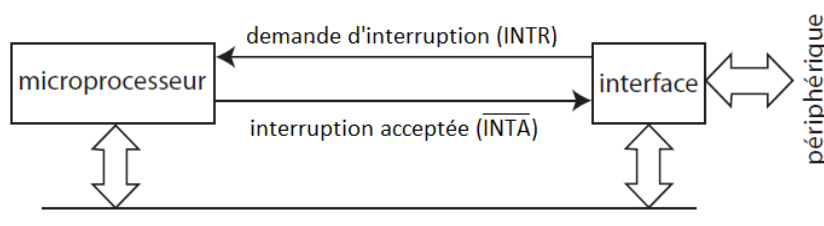


Figure 5.2. Schéma de demande d'interruption matérielle

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 ou une division par zéro.

La figure ci-dessous montre les types d'interruptions pris en charge par le microprocesseur 8086

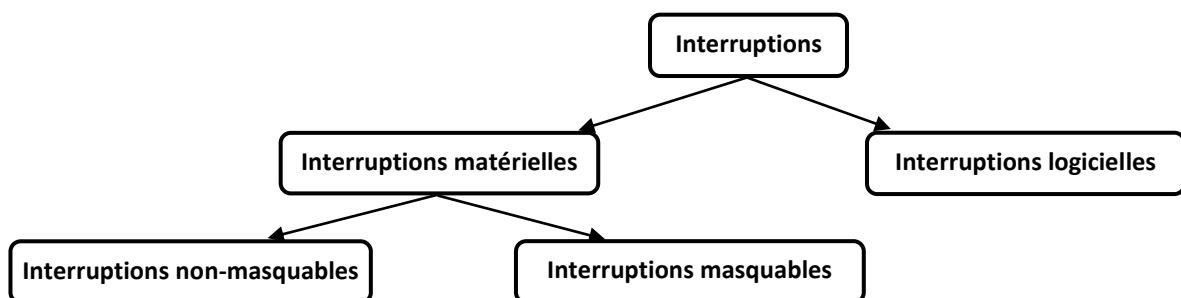


Figure 5.3. Types d'interruptions pris en charge par le 8086

2 Types d'interruptions

On distingue **trois** types d'interruptions d'après la façon dont elles sont appelées. 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 (exception)** : générées par le microprocesseur (**interruptions internes**), ces interruptions sont appelées par le microprocesseur en réponse à une erreur (en cas de dépassement, de division par zéro ou lors de l'exécution pas à pas d'un programme).

3 Table des vecteurs d'interruptions

Les interruptions du 8086 sont vectorisées en utilisant la **table des vecteurs d'interruptions** appelée en anglais **Interrupt Descriptor Table (IDT)**, qui 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.

La vectorisation des interruptions a pour avantage de placer une ISR n'importe où dans la mémoire, il suffit de spécifier le vecteur d'interruption correspondant.

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 :

Si le numéro du périphérique demandeur d'interruption est égal à 20H, l'adresse du vecteur d'interruption sera égale à $4 \times 20H = 80H$.

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).

Drapeau d'interruption IF :

- Si **IF = 1**, alors le processeur accepte les demandes d'interruptions masquables, c'est-à-dire qu'il les traite immédiatement.
- Si **IF = 0**, alors le processeur ignore ces interruptions.

L'état de l'indicateur IF peut être modifié par le programmeur à l'aide des deux instructions suivantes :

- **CLI** (Clear IF : $IF \leftarrow 0$)
- **STI** (SeT IF : $IF \leftarrow 1$).


Vecteur 0	Type 00H Division par zéro	00000H 00003H	<table border="1"> <tr> <td>IP</td> <td>2 octets</td> </tr> <tr> <td>CS</td> <td>2 octets</td> </tr> </table>	IP	2 octets	CS	2 octets
	IP	2 octets					
CS	2 octets						
Vecteur 1	Type 01H Pas à pas	00004H 00007H					
	Type 02H NMI	00008H 0000BH					
	Type 03H Instruction int sur 1 octet	0000CH 0000FH					
	Type 04H Overflow	00010H 00013H					
	Type 05H Réservé par Intel	00014H 00017H					
Interruptions Réservées par Intel	.						
	.						
	.						
	Type 1FH Réservé par Intel	0007CH 0007FH					
Interruptions Réservées par MS-DOS	Type 20H Réservé par Microsoft	00080H 00083H					
	.						
	.						
	Type 3FH Réservé par Microsoft	000FCH 000FFH					
Interruptions libres : disponibles pour l'utilisateur	Type 40H Libre	00100H 00103H					
	.						
	.						
	Type FFH Libre	003FCH 003FFH					

Figure 5.4. Table des vecteurs d'interruptions

4 Prise en charge d'une interruption par le microprocesseur

4.1 Les Interruptions Matérielles

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

1. Un signal de demande d'interruption est émis par un périphérique.
2. Le microprocesseur prend en compte le signal sur sa borne **INTR** après avoir achevé l'exécution de l'instruction en cours. Si l'indicateur du registre d'état **IF = 0**, le signal est ignoré, sinon, la demande d'interruption est acceptée.
3. Si la demande est acceptée, le microprocesseur met sa sortie le signal **INTA** au niveau 0 pendant 2 cycles d'horloge, pour indiquer au contrôleur qu'il prend en compte sa demande.
4. En réponse, le microprocesseur **reçoit le numéro de l'interruption** associé sur le bus de données.
5. Le numéro de l'interruption est utilisé pour trouver le vecteur d'interruption sur la table des vecteurs d'interruptions. Ensuite, le processeur effectue les taches suivante :
 - a. Sauvegarde les indicateurs du registre d'état sur la pile. (En utilisant l'instruction **PUSHF**)

- b. Sauvegarde **CS** et **IP** sur la pile.
 - c. Met les indicateurs **TF (Trap Flag)** et **IF (Interrupt Flag)** à zéro (pour masquer les autres interruptions).
 - d. Cherche dans la table des vecteurs d'interruptions l'adresse du sous-programme d'interruption, qu'il charge dans **CS : IP** (pour calculer l'adresse physique du sous-programme d'interruption $CS \times 10H + IP$).
6. Le microprocesseur exécute le sous-programme de service de l'interruption (ISR : Interrupt Service Routine). Pendant ce temps, les interruptions sont masquées ($IF=0$).
 7. Le sous-programme de service de l'interruption se termine par l'instruction **IRET** (Interrupt return), dès que cette instruction est exécutée, le contenu du registre d'état, **CS** et **IP** sont restaurés à partir de la pile et le microprocesseur reprend l'exécution du programme qu'il avait interrompu.

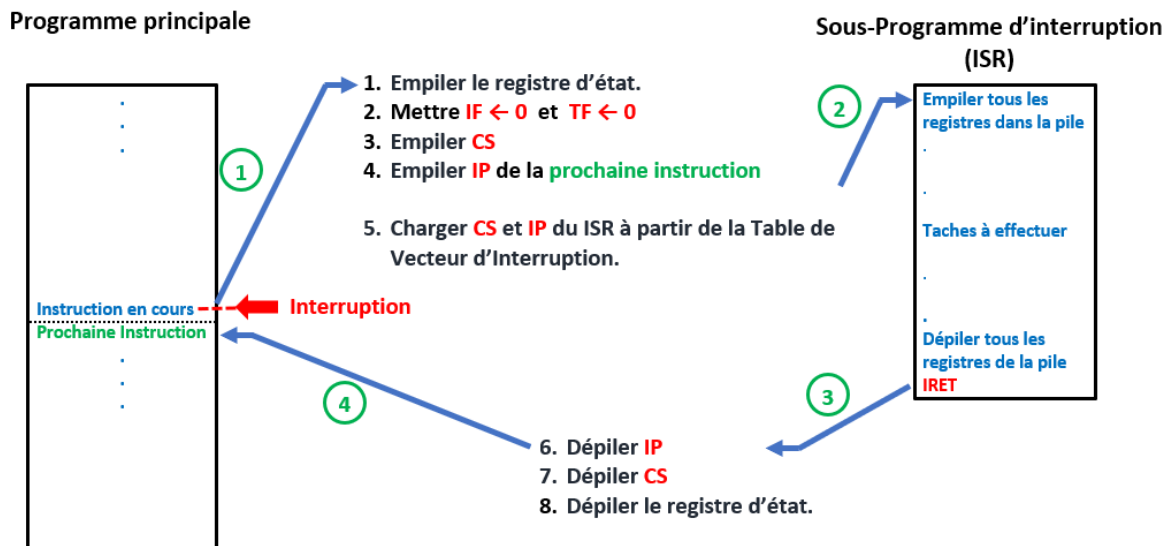


Figure 5.5. Organigramme d'appel d'une interruption par le 8086

Remarque : le microprocesseur peut refuser la demande d'interruption: celle-ci est alors masquée. Le masquage d'une interruption se fait généralement en positionnant à 0 le **IF** dans le registre d'état. Il existe cependant des interruptions **non masquables** qui sont toujours prises en compte par le microprocesseur. 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.

4.2 Les interruptions logicielles

Les interruptions logicielles sont semblables aux interruptions matérielles. L'unique différence réside dans le fait que les interruptions logicielles sont émises par des programmes. Les interruptions logicielles permettent d'appeler des fonctions du **BIOS** (Basic Input Output System) ou du système d'exploitation **DOS** (Disk Operating System). Elles sont émises par des programmes. Ces interruptions ont une fonction définie, par exemple la lecture d'une donnée à partir du clavier, l'écriture d'une donnée à l'écran...etc. D'autres interruptions sont définies par **Intel** (IRQ0 ... IRQ4 voir tableau ci-dessous).

La syntaxe de l'appel d'une interruption logicielle en langage assembleur pour le microprocesseur 8086/8088 est tout simplement : **INT constante** (Exemple : INT 21H)

Où N constante est un entier compris entre 00h et FFh. Il y a donc 256 interruptions possibles pour le microprocesseur 8086.

Exemple :

L'interruption logicielle (software) INT 21 du DOS, de fonction 02H (voir brochure du TP), permet d'afficher un caractère à l'écran.

```
MOV DL, 'a'
MOV AH, 02H
INT 21H
```

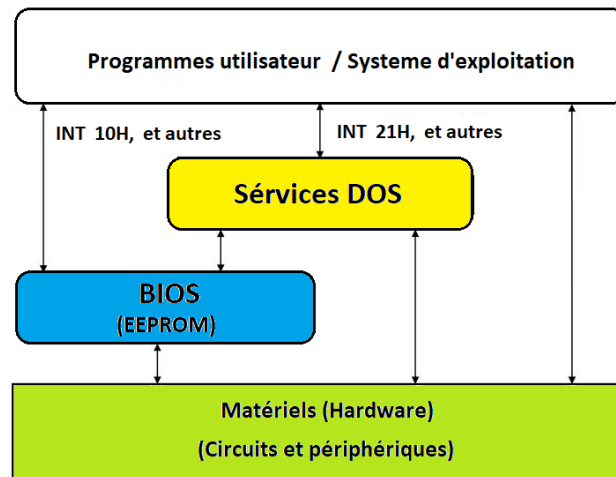


Figure 5.6. Schéma des appels d'interruptions logiciels

5 Différence entre les interruptions matérielles et les interruptions logiciel

Le tableau ci-dessous montre la différence entre les interruptions matérielles et les interruptions logiciels

Tableau 5.1. Différence entre les interruptions matérielles et les interruptions logiciels

	Interruption matériel	Interruption logiciel
Description	L'interruption matérielle est générée par un périphérique externe ou l'apparition d'une défaillance matérielle .	Une interruption logicielle est générée par l'instruction INT .
Déclenchement	L'interruption matérielle est déclenchée par le matériel externe et est considérée comme l'un des moyens de communiquer avec les périphériques extérieurs.	L'interruption logicielle est déclenchée par instruction et considérée comme l'un des moyens de communiquer avec le noyau (DOS ou BIOS) ou de déclencher des appels système, en particulier lors de la gestion des erreurs ou des exceptions .
Catégorie	Les interruptions matérielles peuvent être classées en deux types : 1. Interruption masquable. 2. Interruption non masquable.	Les interruptions logicielles peuvent être classées en deux types : 1. Interruptions normales (instruction INT) 2. Interruption processeur (Exception)
Priorité	Elle a une priorité plus basse que les interruptions logicielles	Elle a une priorité plus élevée parmi toutes les interruptions.
Type de processus	Les interruptions matérielles sont des événements asynchrones.	Les interruptions logicielles sont des événements synchronisés.

6 Liste de quelques interruptions

Le tableau suivant décrit brièvement le rôle de quelques interruptions.

Type	Numéro	Déclenchement	Description
Processeur	00	Division par 0.	Traitement de l'erreur quand une division par 0 survient.
Processeur	01	Pas à pas : quand TF=1	Trap : Exécution pas à pas.
Processeur	02	NMI	appelée quand un problème matériel grave est détecté.
Processeur	03	INT 3	Breakpoint : Point d'arrêt dans le code.
Processeur	04	INTO	Débordement numérique.
BIOS	05	INT 5	Interruption appelée quand la touche "Impécr" est appuyée.
Processeur	06		Opcode invalide (Opcode non défini)
Processeur	07		Pas de Co-processeur mathématique
	08	IRQ0	Timer
	09	IRQ1	Clavier
	0A	IRQ2	PIC2
	0B	IRQ3	Port série 2 (COM2)
	0C	IRQ4	Port série 1 (COM1)

	0D	IRQ5	Disque dur
	0E	IRQ6	Disquette
	0F	IRQ7	Port parallèle (LPT1) : imprimante
DOS	21H	INT	Services généraux (contient plusieurs fonctions)

Note : Chaque **IRQ** correspond à une **interruption matérielle**. Les **interruptions logicielles** sont sollicitées par l'instruction **INT**.

Les cinq premières interruptions logicielles (IRQ0.. IRQ4) sont définies par Intel.

7 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 :

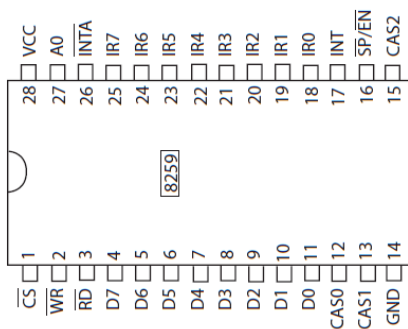


Schéma fonctionnel :

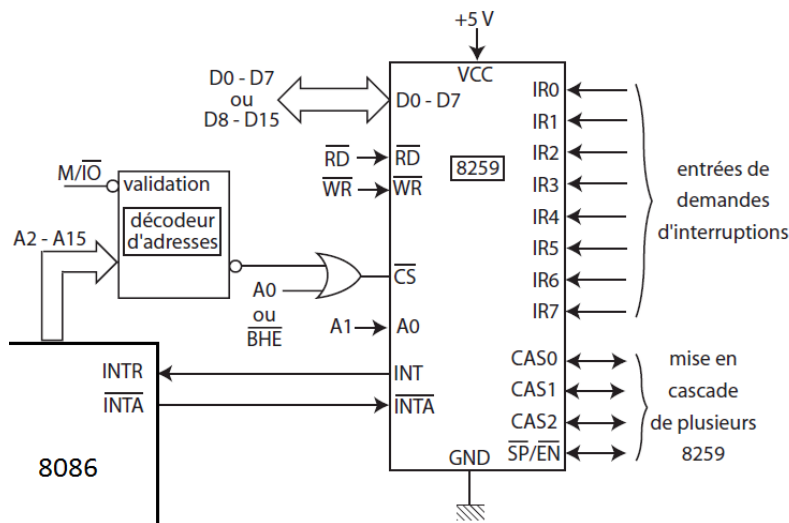


Figure 5.7. Schéma fonctionnel de gestion des interruptions matérielles