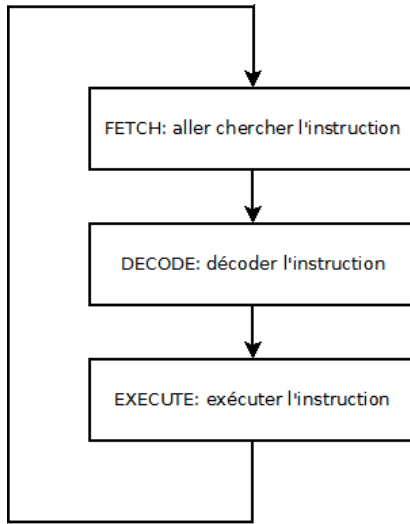


Microprocesseurs (MIC)

Chapitre 3 : Interruptions

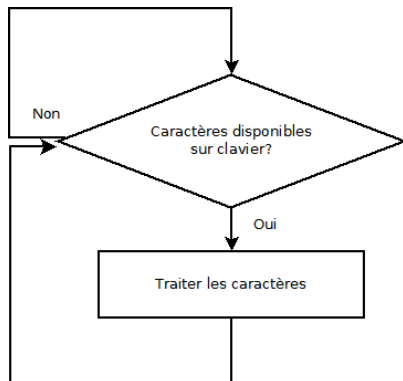
Rappel : le cycle du processeur



Présentation du problème

- Le cycle du processeur vu exécute tout un processus en une fois
 - Quid si un événement extérieur ? (frappe au clavier, click souris. . .)
 - Quid si erreur d'exécution ? (instruction inconnue, division par 0. . .)
 - Quid du dialogue avec le système ? (appel système, scheduling de processus. . .)

Le polling (scrutation)



- Exemple : traitement de texte
- Comment détecter une frappe au clavier sans interruption ?
- Seule solution : le *polling*
- Problème : attente active

- Interruption automatique du processus en cours
- Exécution d'une « routine de traitement d'interruption » (*interrupt handler*)
- En général, reprise ultérieure du programme interrompu

Types d'interruptions

- Interruptions matérielles
 - Provoquées par un matériel extérieur
- Exceptions
 - Provoquées par un programme, généralement suite à une erreur
- Appels-système
 - Provoquées par un programme, pour demander un service à l'O.S.

- Exemples :
 - Frappe sur le clavier
 - Arrivée d'un paquet sur le réseau
 - Click sur la souris
 - Fin de lecture sur disque
 - ...

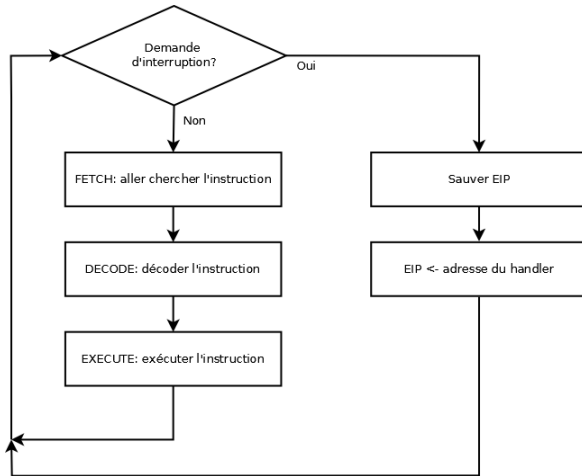
- Exemples :
 - Division par 0
 - Instruction inconnue
 - ...

- Exemples :
 - Demande de lecture sur fichier
 - Demande d'écriture sur fichier
 - ...

Numéros d'interruptions (80386)

- Chaque interruption a un numéro (« vecteur »)
- 256 types différents
- 0-31 : réservés pour des exceptions
 - 0 : division par 0
 - 6 : opcode non-défini
 - ...
- 32-255 :
 - Définis par l'OS
 - Programmés dans le PIC (interruptions matérielles)

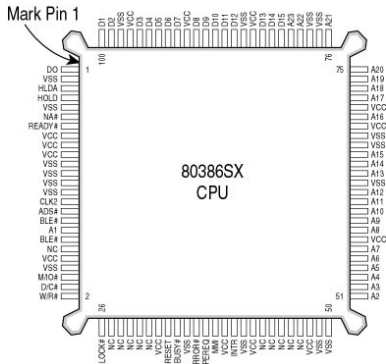
Cycle du processeur avec interruptions





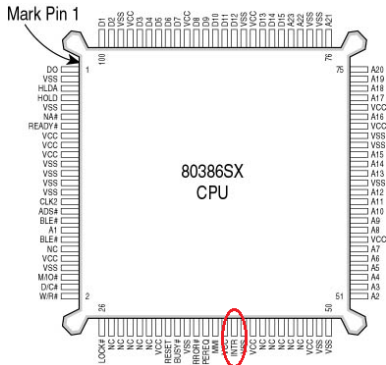
- INTR = INTerrupt Request
- Signale au processeur l'arrivée d'une interruption matérielle

La broche INTR



- INTR = INTerrupt Request
- Signale au processeur l'arrivée d'une interruption matérielle

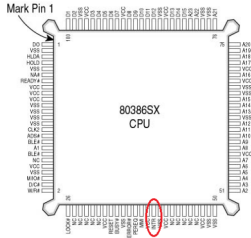
La broche INTR



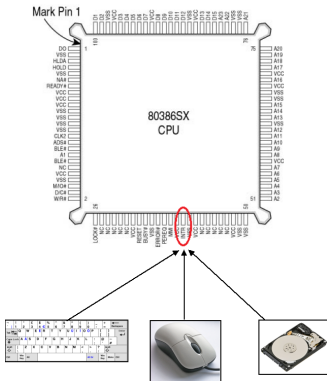
- INTR = INTerrupt Request
- Signale au processeur l'arrivée d'une interruption matérielle

Le contrôleur d'interruptions

- La broche INTR est unique

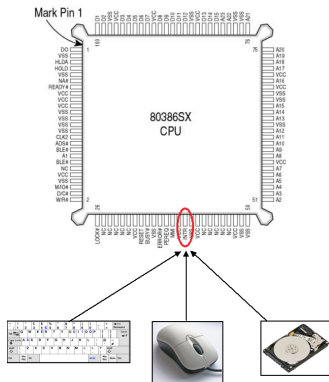


Le contrôleur d'interruptions



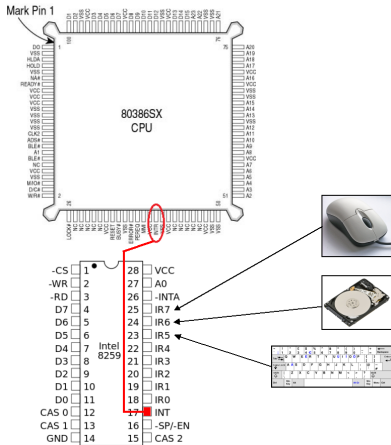
- La broche INTR est **unique**
- Or plusieurs sources potentielles d'interruptions

Le contrôleur d'interruptions



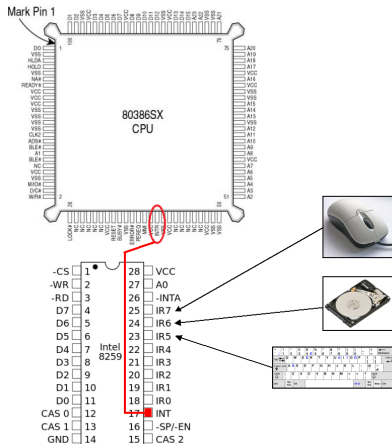
- La broche INTR est **unique**
- Or plusieurs sources potentielles d'interruptions
- \Rightarrow Il faut sérialiser les interruptions

Le contrôleur d'interruptions



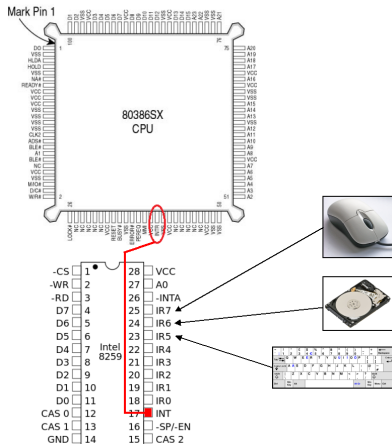
- La broche INTR est **unique**
- Or plusieurs sources potentielles d'interruptions
- \Rightarrow Il faut sérialiser les interruptions
- Rôle du **Contrôleur d'interruptions** (PIC : *Programmable Interrupt Controller*)

Le contrôleur d'interruptions



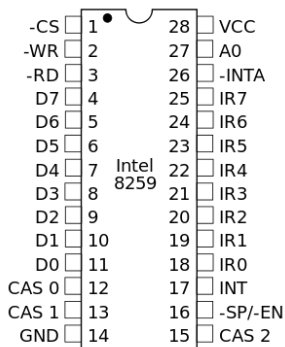
- La broche INTR est **unique**
- Or plusieurs sources potentielles d'interruptions
- \Rightarrow Il faut sérialiser les interruptions
- Rôle du **Contrôleur d'interruptions** (PIC : *Programmable Interrupt Controller*)
- Set pour les interruptions matérielles

Le contrôleur d'interruptions



- La broche INTR est **unique**
- Or plusieurs sources potentielles d'interruptions
- \Rightarrow Il faut sérialiser les interruptions
- Rôle du **Contrôleur d'interruptions** (PIC : *Programmable Interrupt Controller*)
- Set pour les interruptions matérielles
- Exemple : Intel 8259

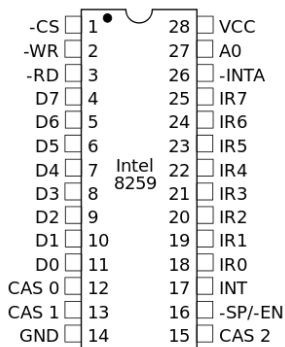
Le contrôleur d'interruptions



- Relié aux périphériques via bornes IR(Q)

Exemple : Intel 8259

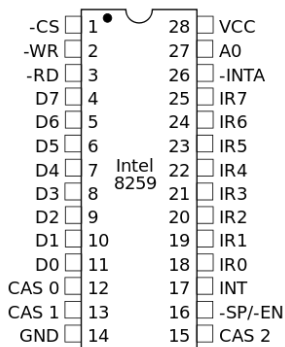
Le contrôleur d'interruptions



Exemple : Intel 8259

- Relié aux périphériques via bornes IR(Q)
- Envoie les demandes d'interruptions une par une au CPU (borne INT)

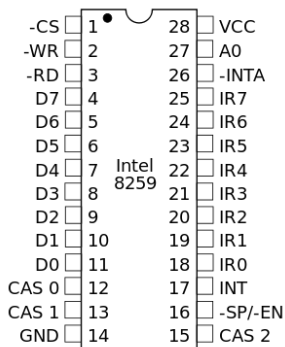
Le contrôleur d'interruptions



- Relié aux périphériques via bornes IR(Q)
- Envoie les demandes d'interruptions une par une au CPU (borne INT)
- Programmable pour donner des priorités différentes à chaque périphérique

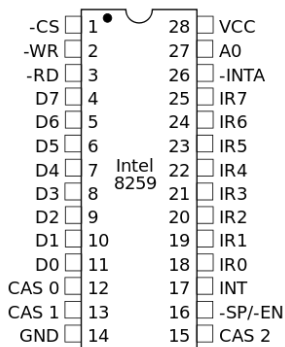
Exemple : Intel 8259

Le contrôleur d'interruptions



Exemple : Intel 8259

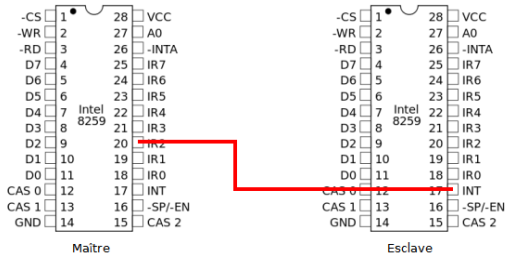
- Relié aux périphériques via bornes IR(Q)
- Envoie les demandes d'interruptions une par une au CPU (borne INT)
- Programmable pour donner des priorités différentes à chaque périphérique
- PIC = *Programmable Interrupt Controller*



Exemple : Intel 8259

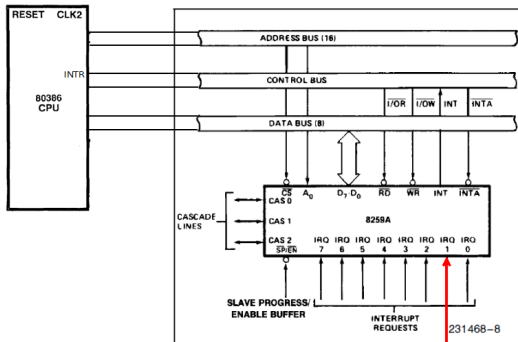
- IRQ0 – timer programmable
- IRQ1 – clavier
- IRQ2 – mise en cascade d'un 2^e PIC
- IRQ3 – port série
- IRQ4 – port série
- IRQ5 – port parallèle
- IRQ6 – disque floppy
- IRQ7 – port parallèle

Deux PIC en cascade



- Permet d'augmenter le nombre d'IRQ

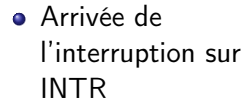
Déroulement d'une interruption



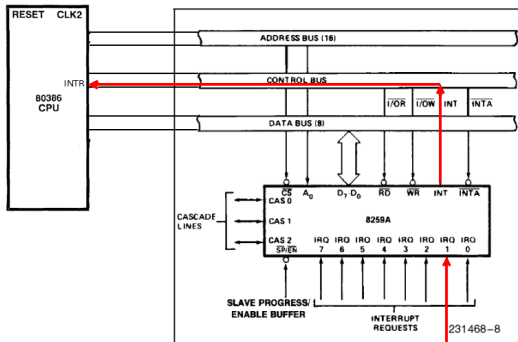
- Arrivée de l'interruption sur INTR



Déroulement d'une interruption



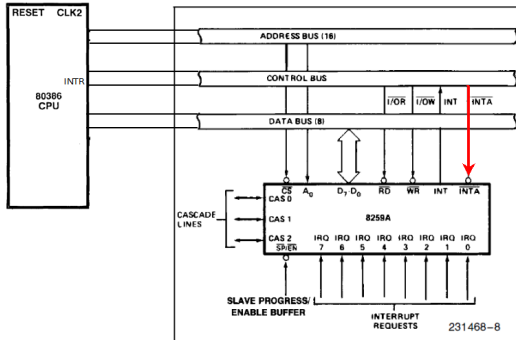
Déroulement d'une interruption



- Arrivée de l'interruption sur INTR



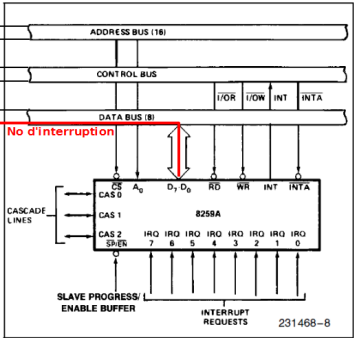
Déroulement d'une interruption



- Arrivée de l'interruption sur INTR
- Accusé de réception sur INTA



Déroulement d'une interruption



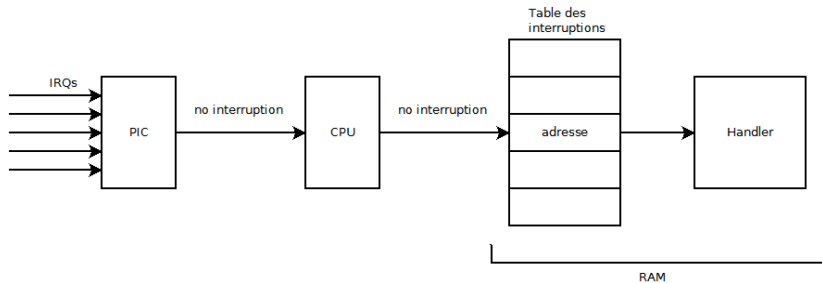
- Arrivée de l'interruption sur INTR
- Accusé de réception sur INTA
- Envoi du n° d'interruption sur bus de données



Table d'interruptions

- Contient les adresses des handlers d'interruptions
- Indexée par le numéro d'interruption
- Dans la RAM
- Gérée par l'OS
- Détails : leçon ultérieure et cours de Systèmes

Table d'interruptions : schéma général



Structure d'un handler (monoprogrammation)

- 1 Sauver registres utilisés (sur pile)
- 2 Traiter l'interruption
- 3 Restaurer registres utilisés (àpd pile)
- 4 IRET

L'instruction IRET

- Exécutée en fin de handler d'interruption
- Restaure automatiquement l'ancien EIP
- Permet de revenir au programme interrompu

Le flag IF

- Si IF=0, le CPU est non-interruptible
- Si IF=1, le CPU est interruptible
- CLI : instruction pour mettre IF à 0
- STI : instruction pour mettre IF à 1
- CLI/STI : utilisés par l'OS

