



# M2-SEM

## Concepts avancés d'architecture

### Cours 4.1

# Interruptions

Année 2020-2021

Pr. R. BOUDOUR

# Processeur 8086 / 8088

2

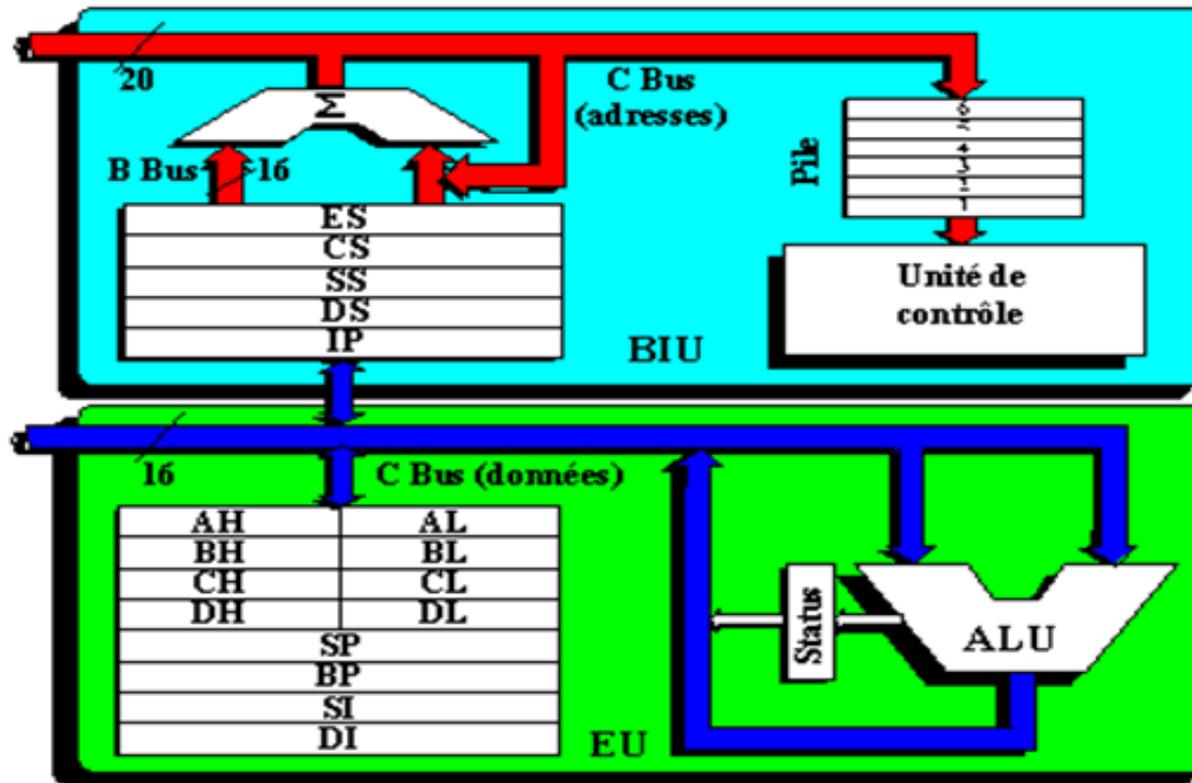
- Le processeur 8086 d'Intel est à la base des processeurs récents actuels.
- Les processeurs successifs de PC se sont en effet construits petit à petit en ajoutant à chaque processeur des instructions et des fonctionnalités supplémentaires mais en conservant à chaque fois les spécificités du processeur précédent.
- C'est cette façon d'adapter les processeurs à chaque étape qui permet qu'un ancien programme écrit pour un 8086 fonctionne toujours sur un nouvel ordinateur équipé d'un nouveau processeur.



# Architecture interne du microprocesseur 8086

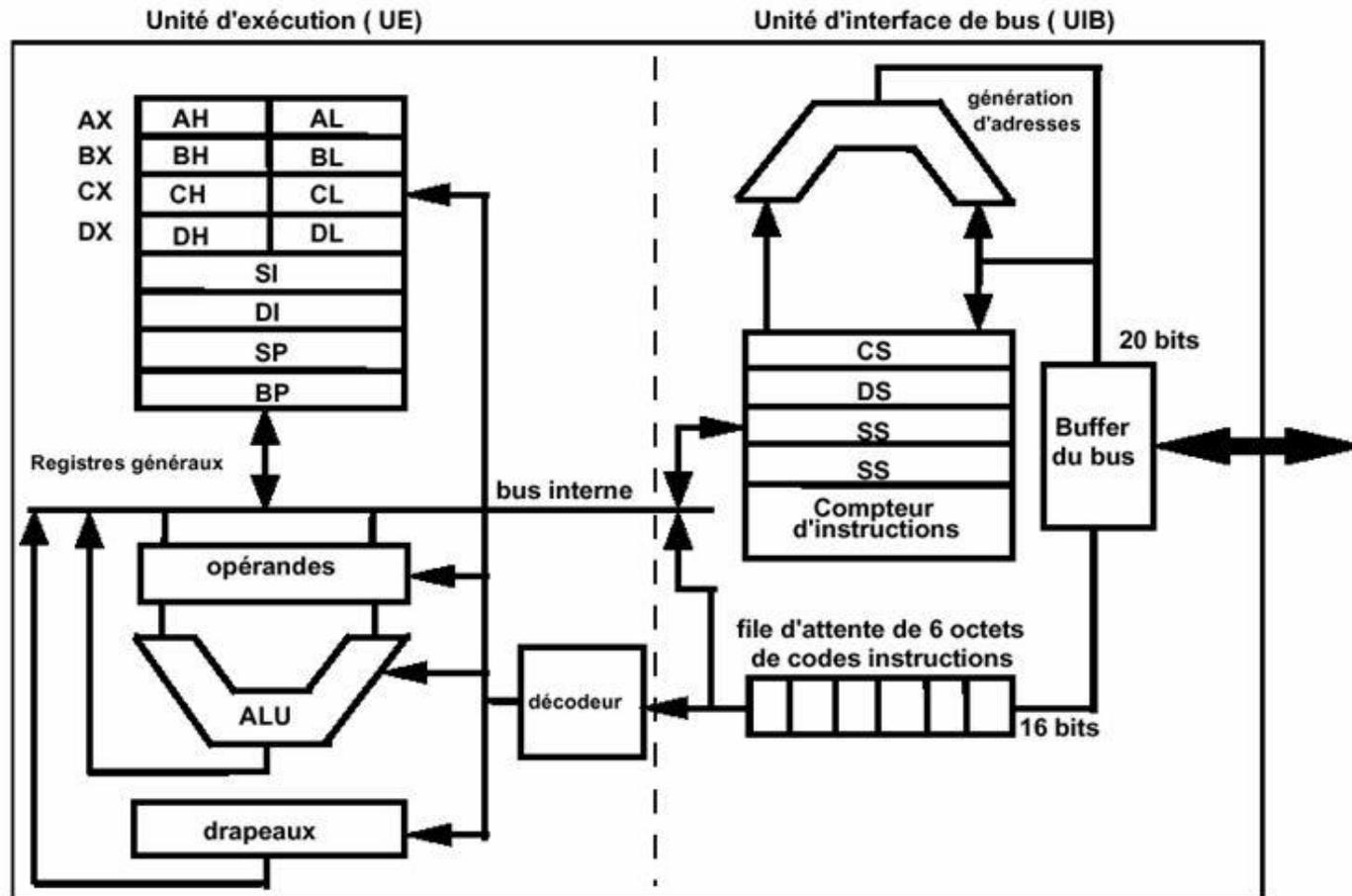
3

□  $\mu$ p 16 bits (Intel 1978) : 8086



# Architecture interne du microprocesseur 8086

4



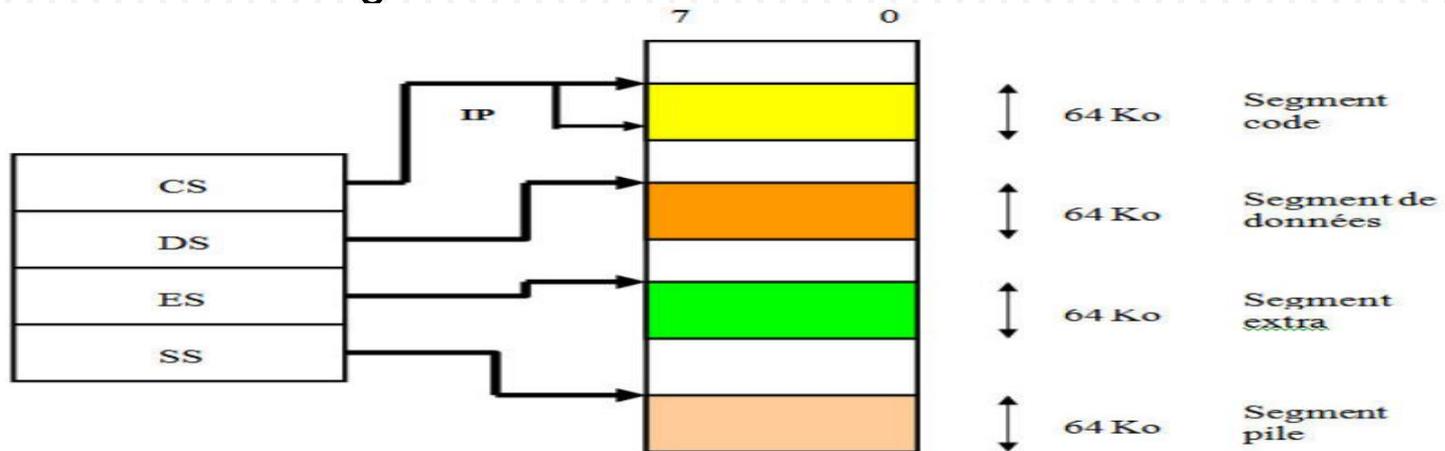
# Registres du 8086/8088

- La taille des registres du 8086 est de 16 bits. Certains d'entre eux (**AX**, **BX**, **CX** et **DX** peuvent être considérés comme formés de 2 registres de 8 bits.
  - Par exemple, **AX** peut être décomposé en **AH** et **AL**, **AH** étant formé de la partie haute (high) de **AX** et **AL** de la partie basse (low).
- Les registres **AX**, **BX**, **CX** et **DX** sont des registres généraux même si certaines instructions du microprocesseur leur donnent une signification particulière.
- **SI** et **DI** sont associés à des opérations de traitement de chaînes de caractères (**Source Index** ou index d'origine et **Destination Index** ou index de destination) mais peuvent également être considérés comme registres à usage général.
- **SP** (**Stack Pointer** ou indicateur de sommet de pile) ne peut être utilisé comme registre à usage général. Mais **BP** (**Base Pointer** ou registre de base) peut l'être.

# Registres du 8086/8088

6

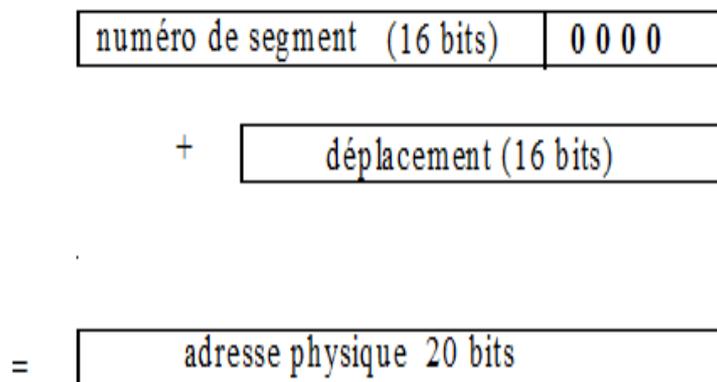
- Quatre registres jouent un rôle fondamental lors des calculs d'adresse :
  - **CS** : Code Segment
  - **DS** : Data Segment (ou segment de données)
  - **SS** : Stack Segment (ou segment de pile)
  - **ES** : Extra Segment



# Calcul de l'adresse physique

7

- ❑ Le bus d'adresses du 8086 possède 20 bits.
- ❑ L'adresse de 20 bits est formée par la juxtaposition d'un registre segment (16 bits de poids fort) et d'un déplacement (*offset*, 16 bits de poids faible).
- ❑ **Adresse physique = Base \* 16 + offset**



*Exemple :*

Pour CS=1000 et IP = 2006

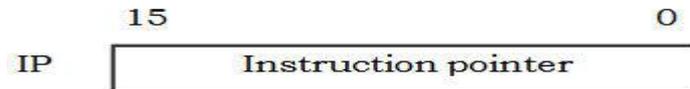
Adresse physique =

$$\begin{array}{r} 10000 \\ + \quad 2006 \\ \hline = 12006 \end{array}$$

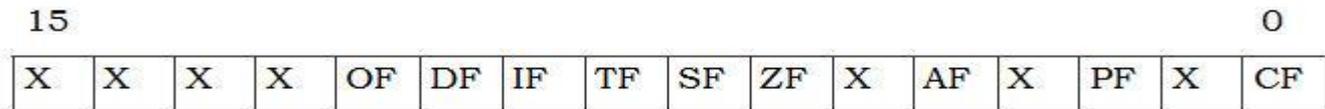
# Registres du 8086/8088

8

- Le registre **IP** (*Instruction Pointer* « ou compteur ordinal) contient l'adresse de la prochaine instruction à effectuer.



- Le registre **STATUS** contient les indicateurs d'état du microprocesseur. Il a le format suivant :

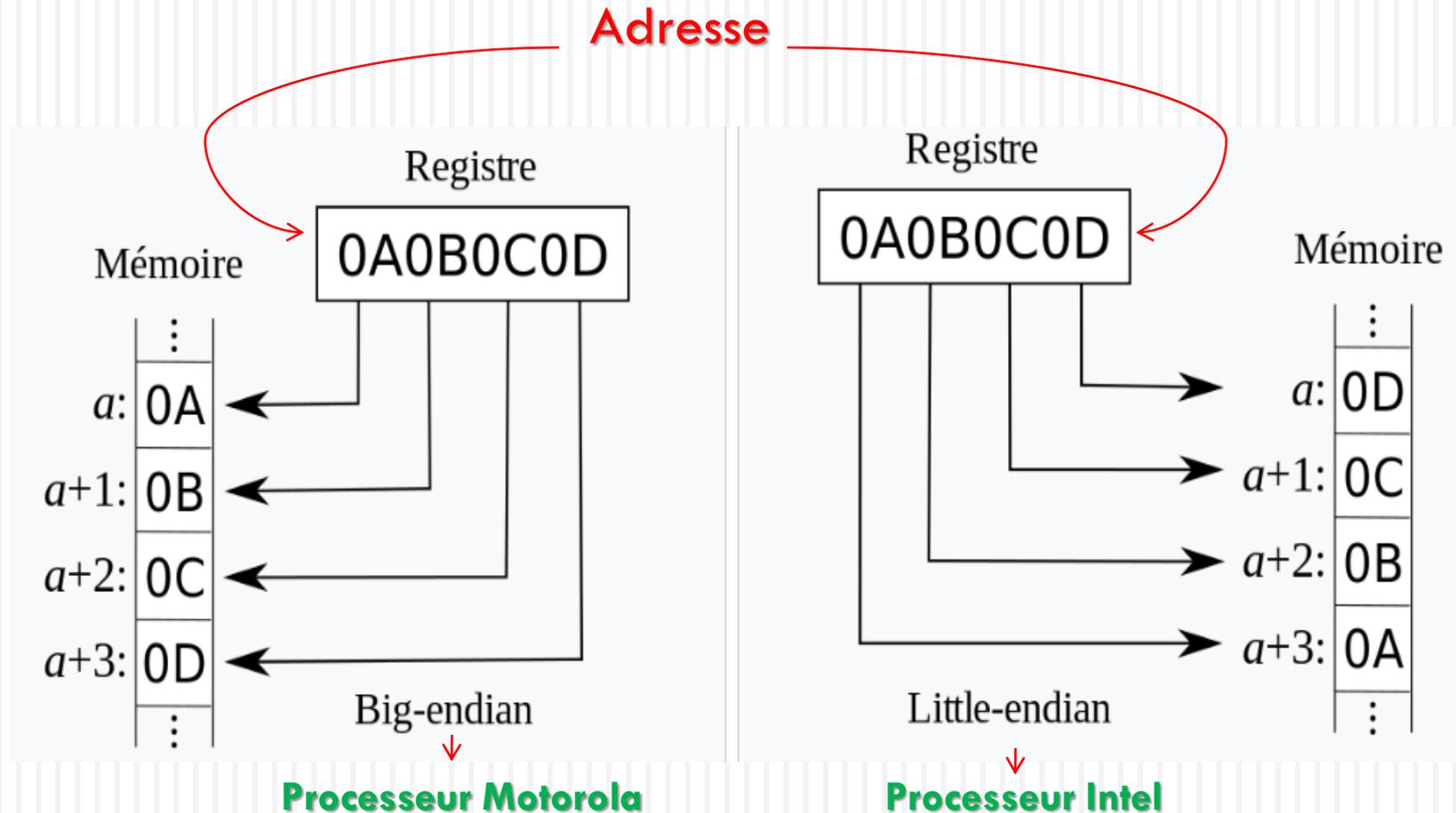


avec :

**O** : débordement (overflow)  
**D** : direction  
**I** : interruption  
**T** : fonctionnement pas à pas  
**S** : signe  
**Z** : zéro  
**A** : auxiliary carry  
**P** : parité  
**C** : retenue (carry)

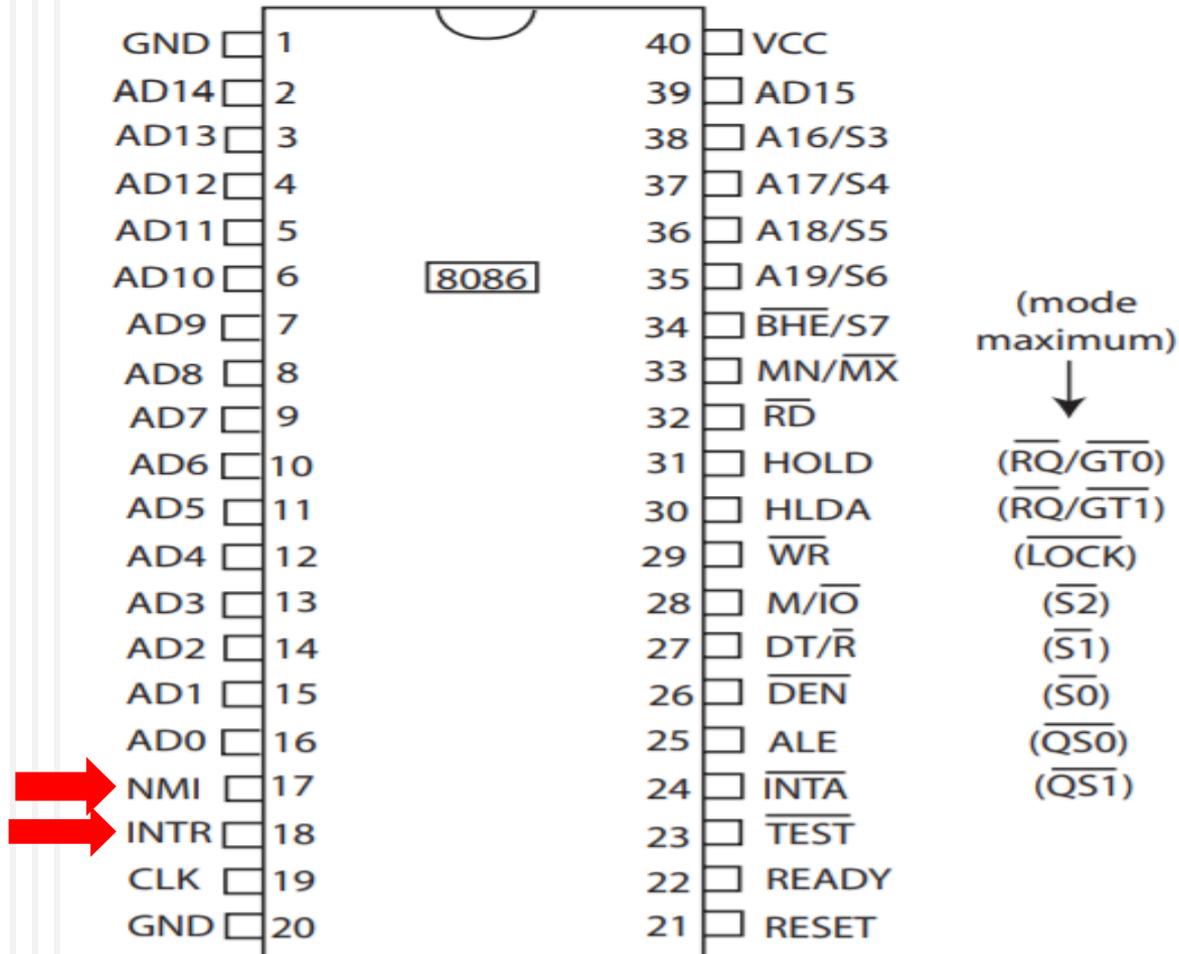
# Petit boutiste – Gros boutiste

9



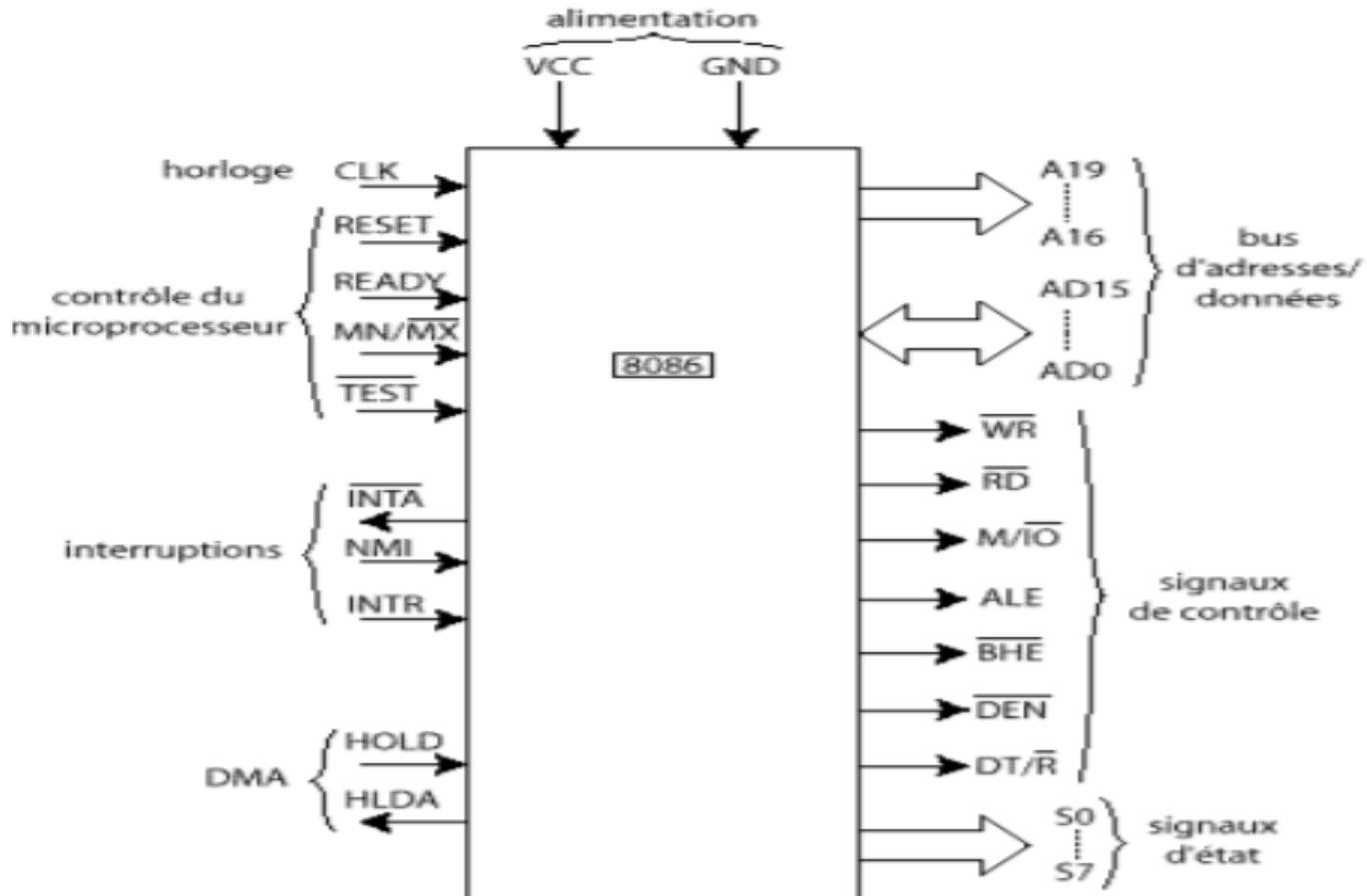
# Brochage du 8086

10



# Schéma fonctionnel du 8086

11



# Un petit programme réalisé avec DEBUG

12

```
C:\DOS>debug
```

```
-a
```

```
24A2 : 0100 ORG 200
```

```
24A2 : 0200 DB "Bien le bonjour de la part de DEBUG!$"
```

```
24A2 : 0225 ORG 100
```

```
24A2 : 0100 MOV AH,09
```

```
24A2 : 0102 MOV DX,200
```

```
24A2 : 0105 INT 21
```

```
24A2 : 0107 MOV AX,4C00
```

```
24A2 : 010A INT 21
```

```
24A2 : 010C [Return]
```

```
-g
```

# Le même programme écrit pour MASM/TASM

13

```
DOSSEG
.MODEL SMALL
.STACK 50
.DATA
LIGNE DB "Bien le bonjour de la part de MASM/TASM!$"
.CODE
MOV AX,@DATA
MOV DS,AX
MOV AH,09H
MOV DX,OFFSET LIGNE
INT 21H
MOV AX,4C00H
INT 21H
END
```

# Besoin d'interruption

14

- Un OS est un programme réactif
- Quand on lui soumet quelques entrées
- Il effectue des traitements
- Il produit un résultat MAIS
- pendant ce temps on peut interagir par interruption du processus en cours ou
- on peut arrêter ou démarrer un autre processus.
- Cette réactivité est due aux interruptions.
- Les OS modernes sont orientés interruption

# Notion d'interruption

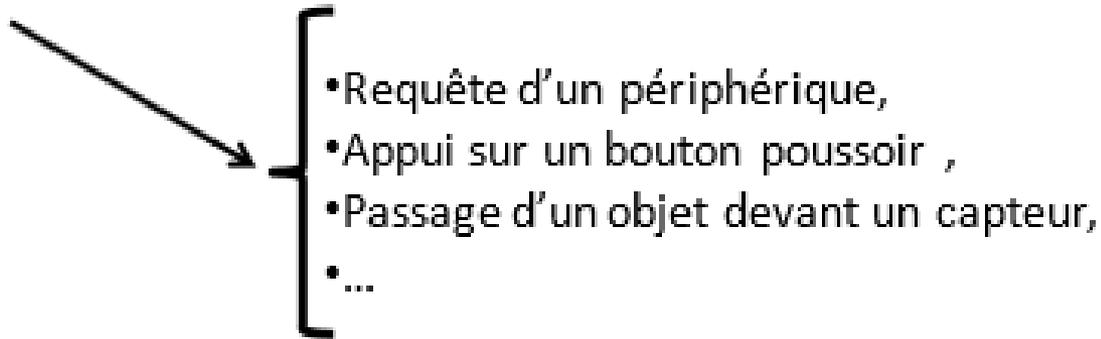
15

- Principe :
  - interrompre un programme en cours pour traiter une tâche plus urgente
  - prise en compte d'événements asynchrones
- Objectif :
  - Détecter un événement imprévu  
alarme, coupure d'alimentation ...
  - Sans avoir à faire une scrutation permanente  
analogie avec une sonnerie de téléphone
  - Pour exécuter un sous programme appelé sous-programme d'interruption.

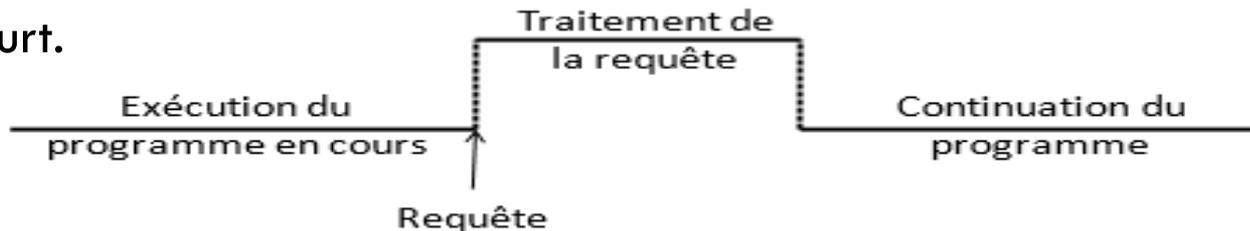
# Notion d'interruption

16

- ❑ Problème : à quel moment le microprocesseur va prendre en compte les **événements** extérieurs à la séquence d'instructions qu'il exécute ?



- ❑ A chacun de ces événements correspond une tâche à exécuter par le microprocesseur. Cette tâche est codée sous forme d'une procédure. Pour pouvoir exécuter cette procédure il faut que se produise une rupture de séquence. Cette rupture doit avoir lieu dans un délai assez court.

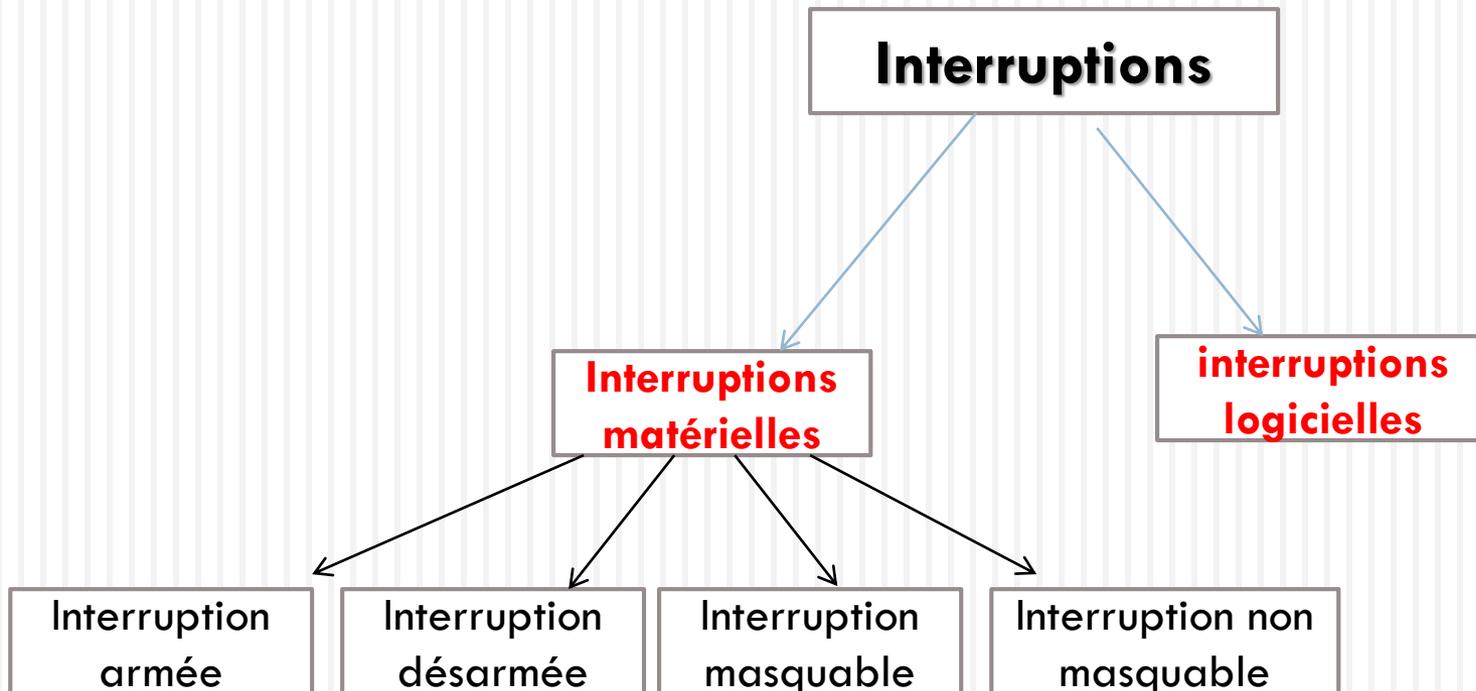


# Notion d'interruption

- ❑ **Une interruption** est un mécanisme qui permet d'interrompre l'exécution d'un processus suite à un événement extérieur ou intérieur et de passer le contrôle à une routine dite routine d'interruption ou traitement d'interruption.
- ❑ **Le système d'interruption** est un dispositif, incorporé au niveau du séquenceur, qui enregistre et traite les signaux d'interruption envoyés au microprocesseur :
  - ❑ Arrêter le processus en cours
  - ❑ Sauvegarder le contexte du processus interrompu
  - ❑ Exécuter le programme de routine d'interruption
  - ❑ Restaurer le contexte du processus interrompu
  - ❑ Reprendre l'exécution du programme interrompu

# Actions d'une interruption

18



# Actions d'une interruption

## **Masquer et démasquer une interruption :**

- Certaines interruptions présentent tellement d'importance qu'il ne doit pas être possible d'interrompre leur traitement. On masquera alors les autres interruptions pour empêcher leur prise en compte.
- Certaines interruptions sont non-masquables : on les prend obligatoirement en compte.
- Une interruption masquée n'est pas ignorée : elle est prise en compte dès qu'elle est démasquée.

## **Armer et désarmer une interruption :**

- Au contraire, une interruption peut être désarmée ou inhibée : elle sera ignorée et donc perdue.
- Par défaut, les interruptions sont armées.

# Propriétés du système d'interruptions

20

- Le système d'interruptions élaboré se trouve dans l'ordinateur orienté vers les applications "temps réel" de conduite de processus industriels ou d'acquisition de données.
- **Un bon système** doit permettre au programmeur de pouvoir :
  - activer/invalider le système d'interruption dans son ensemble;
  - armer/désarmer chacune des interruptions individuellement : une interruption désarmée est ignorée;
  - masquer/démasquer individuellement chaque interruption : une interruption masquée n'est pas ignorée, elle est mémorisée, mais elle n'est prise en compte que lorsqu'elle est démasquée;
  - établir une hiérarchie entre les sources d'interruption avec plusieurs niveaux de priorité, si possible de façon dynamique;
  - associer un programme spécifique à chaque interruption.

# Mécanisme d'interruption

- IT externe et matérielle
  - provoquée par un périphérique (clavier, port ES, imprimante ...)
  - permet de gérer les conflits d'accès au processeur
- IT externe logicielle
  - IT est générée par un programme. L'instruction assembleur INT
- IT interne trap ou exception
  - IT est générée par le processeur lui même.  
Division/0, overflow
- Priorités des interruptions
  - hiérarchisation des IT: classement par ordre de priorités.
  - Priorité IT interne > Priorité IT matérielles > Priorité IT logicielles

# Détection d'une interruption

22

- **Interruptions matérielles:**

- Détection sur une ligne du processeur
- Ligne active => déroutement du programme pour traiter l'IT
- le microprocesseur termine l'instruction en cours avant de traiter l'IT
- événement asynchrone

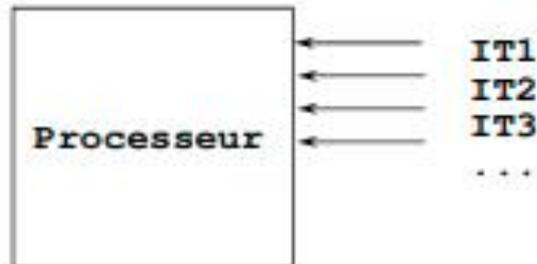
- **Interruptions logicielles**

- invoquée par un processus à un moment précis de son exécution (instruction INT)
- événement synchrone
- peut être assimilé à un appel de sous programme

# Reconnaissance d'une interruption

23

- Différents moyens physique pour déterminer la source d'une IT
- Interruptions **multi-niveaux**:
  - Chaque équipement est relié à une entrée d'IT particulière sur le micro.

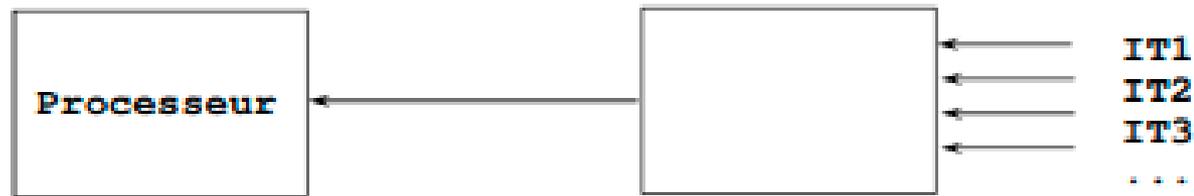


- **Avantage:** solution techniquement simple
- **Inconvénients:** coûteuse en broches d'entrée du processeur, pas très portable

# Reconnaissance d'une interruption

24

- Interruption **ligne unique** :



- » **Avantage** : une seule ligne d'IT sur le processeur
- » **Inconvénient** : scrutation des périphériques pour déterminer le générateur de l'interruption

# Reconnaissance d'une interruption

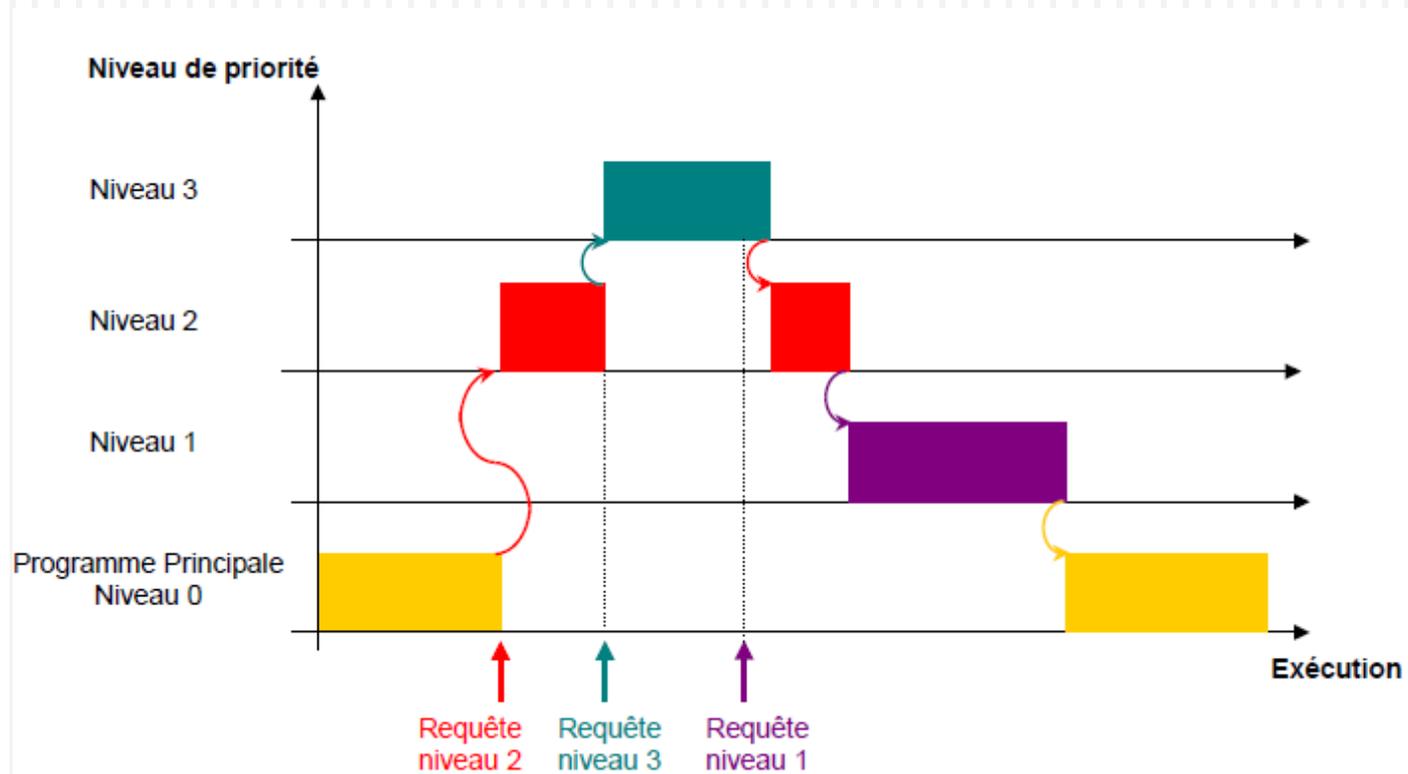
25

## Interruption vectorisée :

- 1 signal de demande
- un identificateur qui permet le branchement direct sur le sous programme d'IT
- le vecteur est déposé sur le bus de donnée
- il est fourni par un composant appelé Contrôleur d'IT »
  - Avantages : le microprocesseur reconnaît de suite le périphérique qui a déclenché l'IT »
  - Inconvénient : il est nécessaire de gérer des priorités (dépôts simultanés de 2 vecteurs sur le bus)

# Interruptions avec priorité

26



# Traitement d'une interruption

- Réception par l'UC d'une demande d'IT interne ou externe
- Acceptation ou rejet par l'UC de cette demande
- Fin de l'instruction en cours
- Sauvegarde de l'état du système
- Forçage du compteur ordinal qui prend l'adresse de la première instruction du SP associé à cette IT
- le SP une fois terminé provoque la restauration des registres et du micro

# Traitement d'une interruption

28

- a) Type = 14h
- b) Type \* 4 = 50h -> Offset dans la Table des vecteurs d'IT
- c) Table des vecteurs d'IT
- d) Flags --> pile

00054	XX	} ISR Adresse = 2000:3456 =23456
00053	20	
00052	00	
00051	34	
00050	56	
0004F	XX	
	⌞	
	⌞	
00002	XX	...
00001	XX	
00000	XX	

CS:IP--> pile  
0 --> IF  
2000:3456--> CS:IP

## e) Procédure ISR

```
23456 PUSH AX  
      PUSH BX
```

```
23874 IRET
```

- f) Pile --> CS:IP  
Pile --> Flags

# Interruptions 8086

- ❑ Le microprocesseur 8086 peut gérer jusqu'à 256 interruptions :
  - ❑ **Les interruptions matérielles** sont produites par l'activation des lignes INTR et NMI ;
  - ❑ **Les interruptions logicielles** sont produites par l'instruction INT n, où n est le type de l'interruption ;
  - ❑ **Les dérivements** sont générés par le processeur.

# Interruptions du 8086

30

- ❑ Toutes les interruptions ont un numéro de 0 à 255, sauf reset. Certains numéros sont fixés par le matériel, d'autres par les concepteurs du 8086 et le reste disponibles pour le programmeur.
- ❑ Les interruptions ont des priorités. Elles sont, du plus prioritaire au moins prioritaire :
  - Reset, Exceptions, INT, NMI et INTR.**
- ❑ A l'intérieur d'une de ces catégories, l'INT avec le numéro le plus bas est habituellement la plus prioritaire.
  - ❑ Une interruption de haute priorité peut interrompre une interruption de basse priorité.
  - ❑ Une interruption de basse priorité ne peut pas interrompre une interruption de priorité égale ou supérieure.
  - ❑ L'interruption Reset est la plus prioritaire.

# Indicateur IF

31

- A un instant donné, les interruptions sont soit masquées soit autorisées, suivant l'état d'un indicateur spécial du registre d'état, IF (Interrupt Flag).
  - 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 à l'aide de deux instructions,
  - CLI (CLear IF pour la mise à 0 de IF),
  - et STI (SeT IF, pour la mise à 1 de IF).

# Interruptions vectorisées

32

- ❑ La table des vecteurs d'interruption contient l'adresse des instructions à exécuter lorsqu'une interruption survient.
  - ❑ Pour chaque interruption, la table contient 4 bytes d'information : CS et IP.
  - ❑ La taille de la table des vecteurs d'interruption est donc :  
 $256 \text{ ints} * 4 \text{ bytes} = 1\text{K bytes ou } 1\text{Ko.}$
  - ❑ La table des vecteurs d'interruption commence à l'adresse 00000h et elle finit à l'adresse 003FFh. Ne pas mettre de segments à cet endroit !
  - ❑ Lorsqu'une interruption survient, le numéro de l'interruption permet de trouver l'emplacement des instructions à exécuter.
  
- ❑ Pour chaque interruption, il existe un programme associé, qui s'exécute pour effectuer un service donné (ISR).
- ❑ Les adresses de ISR sont données par IP et CS (2 octets chacun), stockées au bas de l'espace mémoire.

# Interruptions vectorisées

33

- ❑ Le numéro de l'interruption est multiplié par 4 afin de trouver l'adresse dans la table associée à CS et IP (IP est à l'adresse inférieure, suivi de CS). Puis un JMP à CS:IP est fait. CS:IP est un « vecteur » vers le code à exécuter afin de répondre à l'interruption.
  
- ❑ **Exemple** : Si le contenu de la mémoire, à partir de l'adresse 00000h est :  
00h, 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, etc.  
et que l'interruption 1 survient, alors CS:IP = 0000:0004 et la routine se trouvant à l'adresse 0706:0504 sera exécutée.
  
- ❑ Reset est la seule interruption sans numéro.  
Pour reset, CS:IP = FFFF0H

# Table des vecteurs d'interruption

34

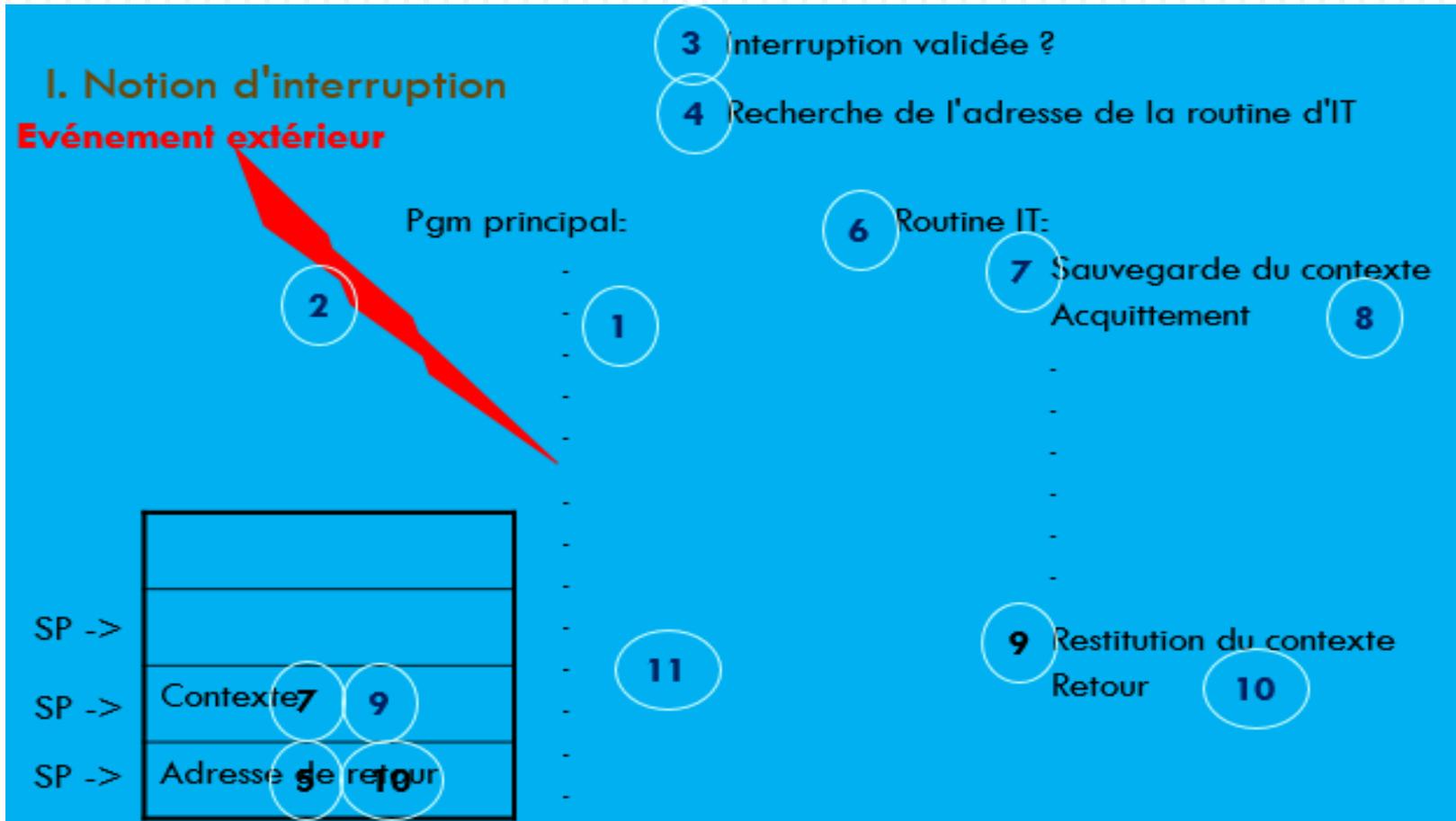
INT N	Adresse Logique
INT 00	0000:0000
INT 01	0000:0004
INT 02	0000:0008
INT NN	0000:[00NNx4]
INT FF	0000:03FC

Table des vecteurs d'interruptions

# Déroulement d'une interruption masquable

35

## I. Notion d'interruption Événement extérieur



# Déroulement d'une interruption masquable

36

- Reprenons les différents évènements liés à la réception et le traitement d'une interruption masquable :
  - Un signal INT est émis par un périphérique (ou plutôt par l'interface gérant celui-ci).
  - Le contrôleur d'interruptions reçoit ce signal sur une de ses bornes IRQi. Dès que cela est possible (suivant les autres interruptions en attente de traitement), le contrôleur envoie un signal au processeur sur sa borne INT.
  - Le processeur prend en compte le signal sur sa borne INTR après avoir achevé l'exécution de l'instruction en cours (ce qui peut prendre quelques cycles d'horloge). Si l'indicateur IF=0, le signal est ignoré, sinon, la demande d'interruption est acceptée.

# Déroulement d'une interruption masquable

- Si la demande est acceptée, le processeur met sa sortie INTA au niveau 0 pendant deux cycles d'horloge, pour indiquer au contrôleur qu'il prend en compte sa demande.
- En réponse, le contrôleur d'interruption place le numéro de l'interruption associé à la borne IRQ<sub>i</sub> sur le bus de données.
- Le processeur lit le numéro de l'interruption sur le bus de données et l'utilise pour trouver le vecteur d'interruption (afin de traiter l'interruption). Ensuite, tout se passe comme pour un appel système, c'est-à-dire que le processeur :
  - sauvegarde les indicateurs du registre d'état sur la pile (en d'autres termes il conserve l'état de la mémoire concernant le programme en cours d'exécution) ;
  - met l'indicateur IF à 0 pour masquer les interruptions suivantes ;
  - cherche dans la table des vecteurs d'interruptions l'adresse du traitant d'interruption.
- La procédure traitant l'interruption se déroule. Pendant ce temps, les interruptions sont masquées (IF=0).
- La procédure se termine par l'instruction IRET qui permet de reprendre le programme qui avait été interrompu.

# Séquence d'évènements après une interruption

- ❑ Lors d'une interruption logicielle ou matérielle, les évènements suivants se produisent :
  - ❑ Le 8086 termine l'instruction en cours ;
  - ❑ Le drapeau IF est testé si l'interruption provient de la ligne d'interruption matérielle (INTR). Si le drapeau est à 0, l'interruption est ignorée ;
  - ❑ Si une autre interruption de priorité supérieure ou égale est en cours, l'interruption est ignorée.
  - ❑ 3 registres sont mis sur la pile (PUSH) dans l'ordre qui suit : les drapeaux, CS actuel, puis IP actuel. Le drapeau IF est mis à 0 après le PUSH des flags ;

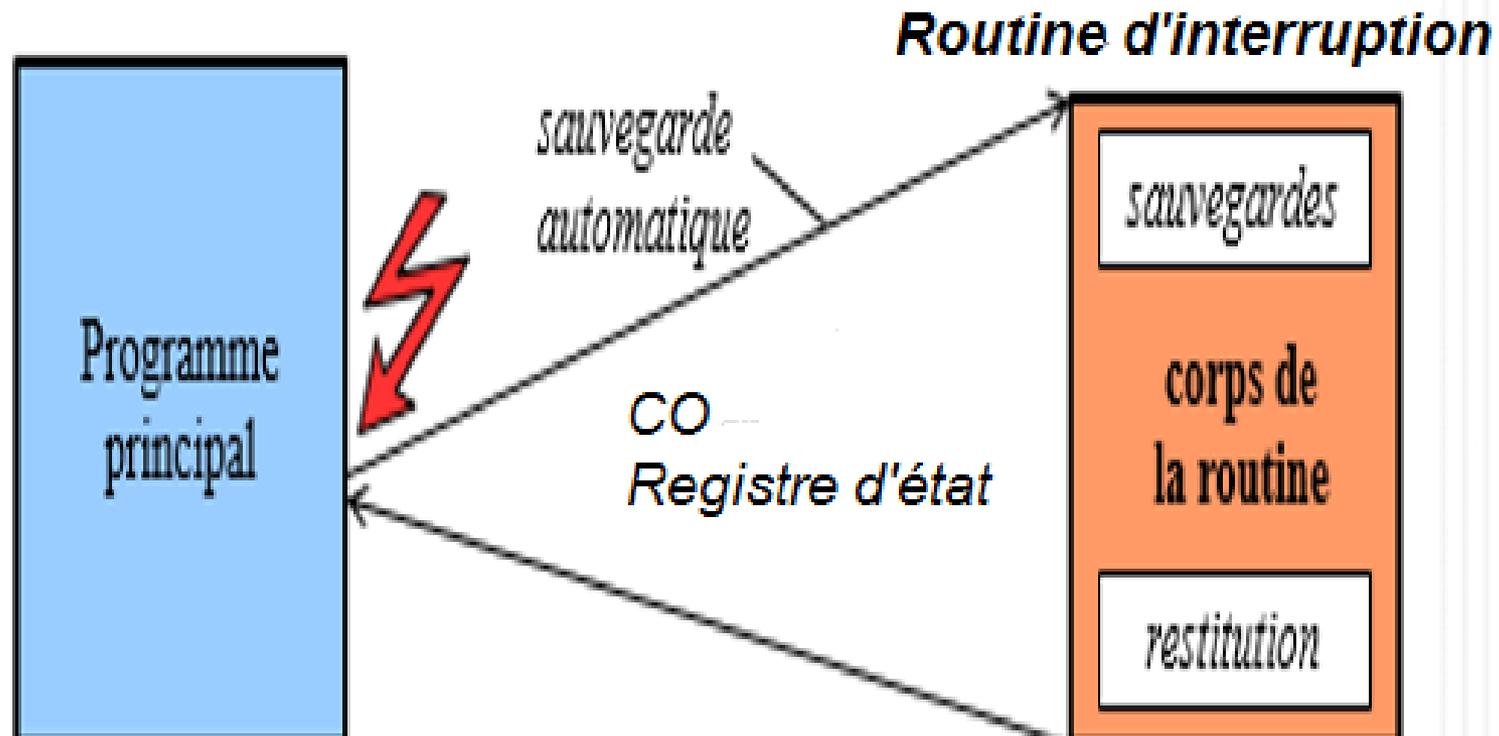
# Séquence d'évènements après une interruption

39

- ❑ Un JMP est exécuté à l'adresse CS:IP trouvée dans la table des vecteurs d'interruption à partir du numéro de l'interruption ;
- ❑ La routine de service de l'interruption (ISR ou Interrupt Service Routine) est exécutée ;
- ❑ Théoriquement, la dernière instruction de l'ISR est IRET (Interrupt RETurn). Un IRET est équivalent à trois POP dans cet ordre: IP, CS et drapeaux.

# Séquence d'évènements après une interruption

40



# ISR (Interrupt Service Routine)

- ❑ Pour chaque interruption, il existe un programme associé, qui s'exécute pour effectuer un service donné (ISR).
- ❑ Les adresses de ISR sont données par IP et CS (2 octets chacun), stockées au bas de l'espace Mémoire → 1024 octets sont réservés pour la table des vecteurs d'Interruptions

# Interruptions Internes (Exceptions)

42

- ❑ Les interruptions internes sont des interruptions qui peuvent se produire sans que l'instruction INT soit présente. Il s'agit d'exceptions qui surviennent quand un évènement logiciel spécial arrive (division par zéro par exemple)
  - ❑ L'interruption 0 survient lorsque le diviseur de DIV ou IDIV est 0.
  - ❑ L'interruption 4 survient lorsque le drapeau Overflow est 1 et que l'instruction INTO est exécutée.
  - ❑ Les interruptions 1 et 3 servent au debogage de programme.

*Note : Comme pour toutes les interruptions, les interruptions internes désactivent les interruptions matérielles.*

# Exemples

43

## INT 00 (Division / 0)

MOV	AL, 87h
XOR	CL, CL
DIV	CL

INT 00 → ISR(Message: DIVIDE ERROR)

## INT 01 (Single-Stepping)

1. TF du FR → 1
2. CPU → CS:IP = 00004 pour chercher l'adresse de ISR associé (DUMPer le contenu des registres sur l'écran)

## INT 02 (NMI)

Si le Pin NMI est '1' → CPU → CS:IP = 00008 → ISR associée

## INT 03 (BreakPoint)

Comme INT 01 mais arrêt après un bloc d'instructions → Debugage

## INT 04 (Dépassement)

Dépassement de capacité pour les nombres signés

L'instruction **INTO** → tester OF.  
Si '1' → INT04 activé (CS:IP=00010H) → Message affiché  
Sinon INTO ↔ NOP

# Interruptions Logicielles

44

- ❑ Les interruptions logicielles sont utilisées pour exécuter des tâches logicielles ayant une priorité sur le programme principal.
- ❑ Une interruption logicielle est appelée avec l'instruction INT xx où xx est le numéro de l'interruption.
- ❑ Chaque interruption peut avoir des sous-fonctions. Pour spécifier une sous-fonction :
  - ❑ Il faut placer dans un registre prédéterminé le numéro de la sous-fonction voulue avant d'appeler l'interruption (le registre utilisé est souvent AH).
  - ❑ La sous-routine de service de l'interruption testera la valeur du registre prédéterminé et aiguillera la sous-routine en conséquence.

# Interruptions Logicielles

45

□ Voici quelques exemples tirés de EMU8086 :

**INT 20h** - Quitte le système d'exploitation.

**INT 21h / AH=09h** - Sortie d'une chaîne de caractères depuis DS:DX.

**INT 21h / AH=0Ah** - Entrée d'une chaîne de caractères vers DS:DX, le premier octet indique la taille du buffer, le deuxième octet indique le nombre de caractères réellement lus.

**INT 21h / AH=4Ch** - Quitte le système d'exploitation.

**INT 21h / AH=31h** - Quitte le SE sans libération de l'espace mémoire

**INT 21h / AH=01h** - lit un caractère sur l'entrée standard, avec écho, le résultat est stocké dans AL.

**INT 21h / AH=02h** - écrit un caractère vers la sortie standard, DL = le caractère à écrire, après exécution, AL = DL.

# Exemples pour $\mu$ P 8088/8086

46

```
C> debug  
-D 0000:0000-002F
```

```
0000:0000 E8 56 2B 02 56 07 70 00  
          C3 E2 00 F0 56 07 70 00  
0000:0010 56 07 70 02 ....  
0000:0020 A5 FE 00 F0 ....
```

**ISR(INT 00)  $\rightarrow$  CS:IP = 022Bh:56E8h**

**ISR(NMI)  $\rightarrow$  CS:IP = F000h:E2C3h**

**ISR(INT08)  $\rightarrow$  CS:IP = F000h:FEA5h**

$8 \times 4 = 20H$

**NOTE:**

**INT00  $\rightarrow$  07998H  $\rightarrow$  Espace MS DOS**  
**NMI  $\rightarrow$  FE2C3H  $\rightarrow$  Espace BIOS ROM**

# Questions

47

- Q1. Supposons que les 256 premiers bytes de la mémoire valent de 00h à FFh. A quelle adresse se trouvera le code à exécuter lors de l'IRQ12 ?
- Q2. Quel est le temps minimal entre deux demandes d'interruptions de même type (sur front). En déterminer, quelle en sera la conséquence ? la fréquence maximale. Si ce temps n'est pas respecté
- Q3. Quelles différences y a-t-il entre un appel d'une procédure et un traitant ?

A 3D rendered white figure, resembling a stylized person or character, is holding a large, light gray rectangular sign. The sign is tilted slightly and features the text "THANKS FOR YOUR ATTENTION" in a bold, white, sans-serif font with a subtle drop shadow. The figure is standing on a light gray surface, and the background is a plain, light gray gradient.

**THANKS FOR YOUR  
ATTENTION**