



**Fonctionnement du processeur :  
exécution des instructions machine**



**RUPTURE DE PROGRAMME : les  
INTERRUPTIONS**

# Notion d'interruptions

- L'exécution d'un programme s'effectue instruction après instruction.
  - Une **interruption** est un mécanisme permettant de stopper l'exécution du programme en cours afin d'aller exécuter une tâche jugée plus prioritaire.
  - Une **interruption** permet de signaler un événement survenu sur la machine et d'exécuter un traitement spécifique (programme de service ou **routine d'interruption**) lié à cet événement .



Elle travaille



DRING !



Elle répond  
au téléphone



Elle reprend son  
travail

# Notion d'interruptions

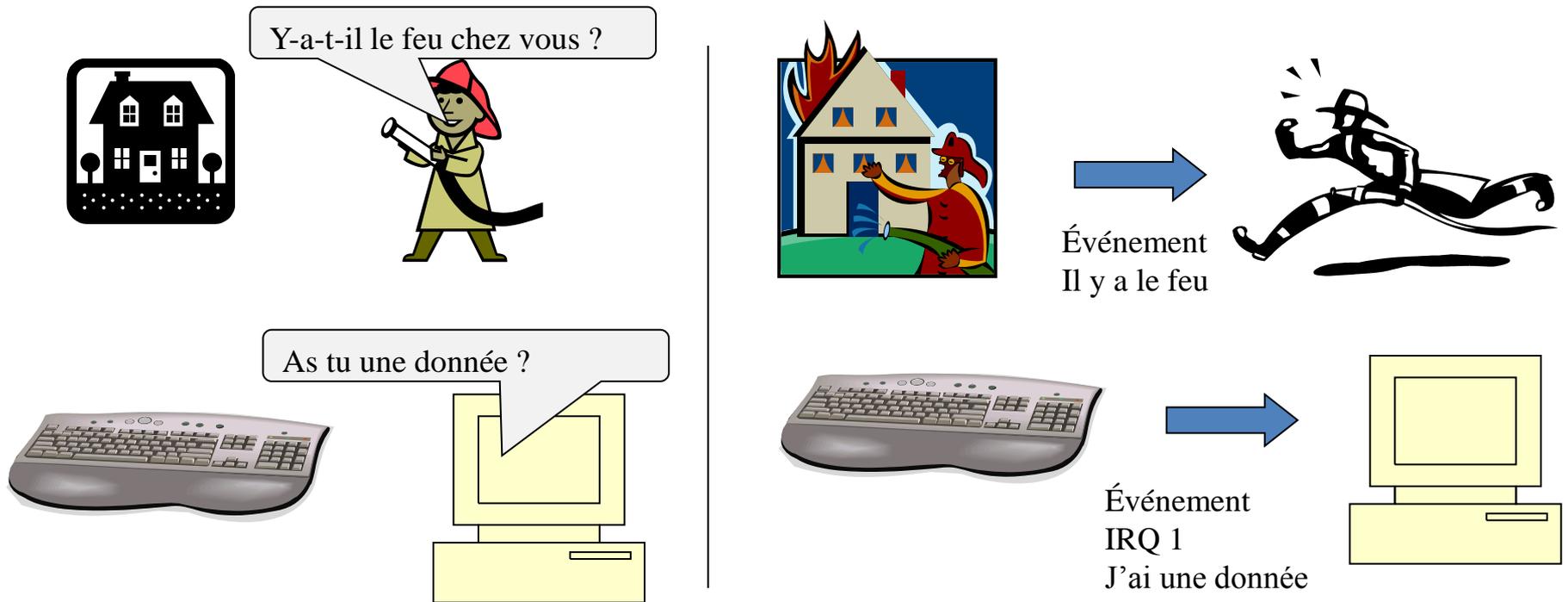
- Une **interruption** est un mécanisme permettant de stopper l'exécution du programme en cours afin d'aller exécuter une tâche jugée plus prioritaire.
- Elle est caractérisée par un **numéro** et un traitement associé (la routine ou traitant d'interruption - **ISR - Interrupt Service Routine**)

On distingue principalement deux types d'événements :

- Les interruptions externes ou matérielles sont émises par les périphériques du processeur (fin d'écriture disques, plus de papier imprimante...). Ce sont les **IRQs**.
- Les interruptions internes ou logicielles sont émises par le processeur lui-même lorsqu'il rencontre une erreur dans l'exécution du programme (division par zéro, accès mémoire illégal). Ce sont les **trappes**.

# Notion d'interruptions

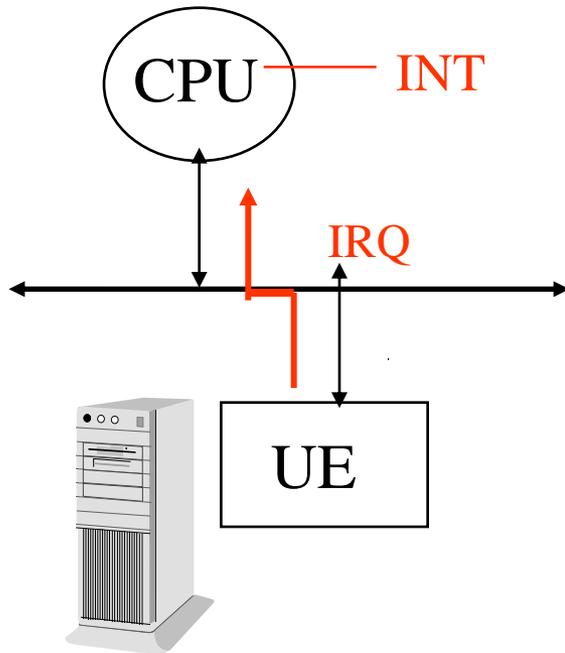
- Une **interruption** est un mécanisme permettant de stopper l'exécution du programme en cours afin d'aller exécuter une tâche jugée plus prioritaire.
- Elle évite au processeur de **scruter** les périphériques



SCRUTATION : Toutes les 5 minutes

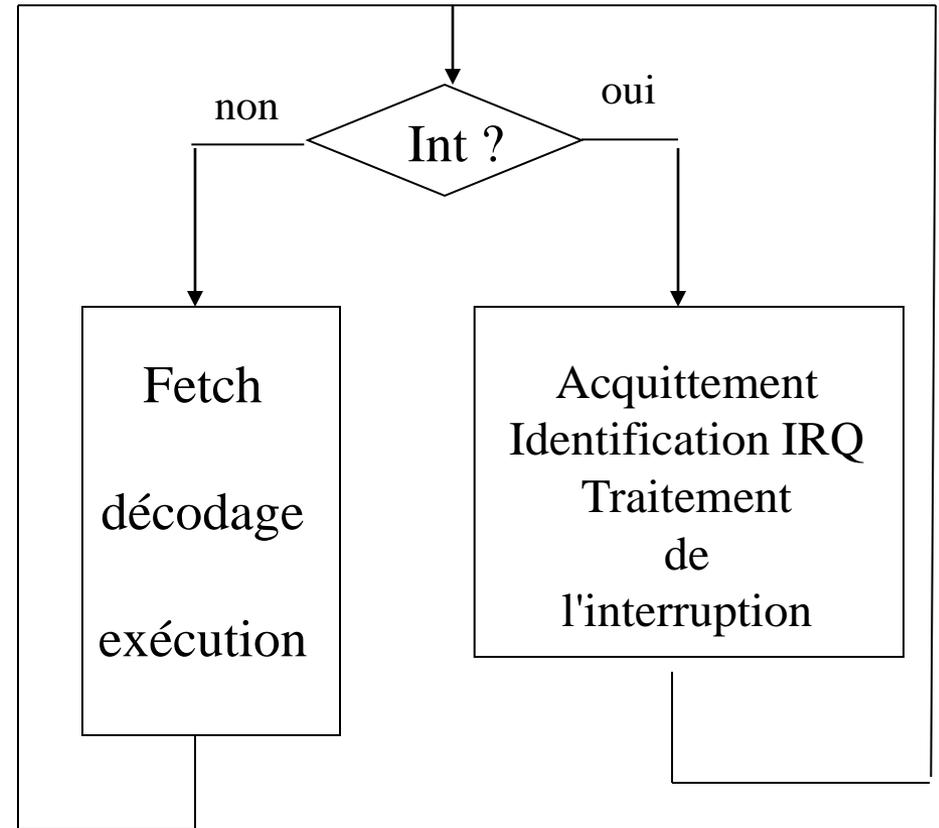
INTERRUPTION : quand l'événement survient

# Mécanisme des interruptions

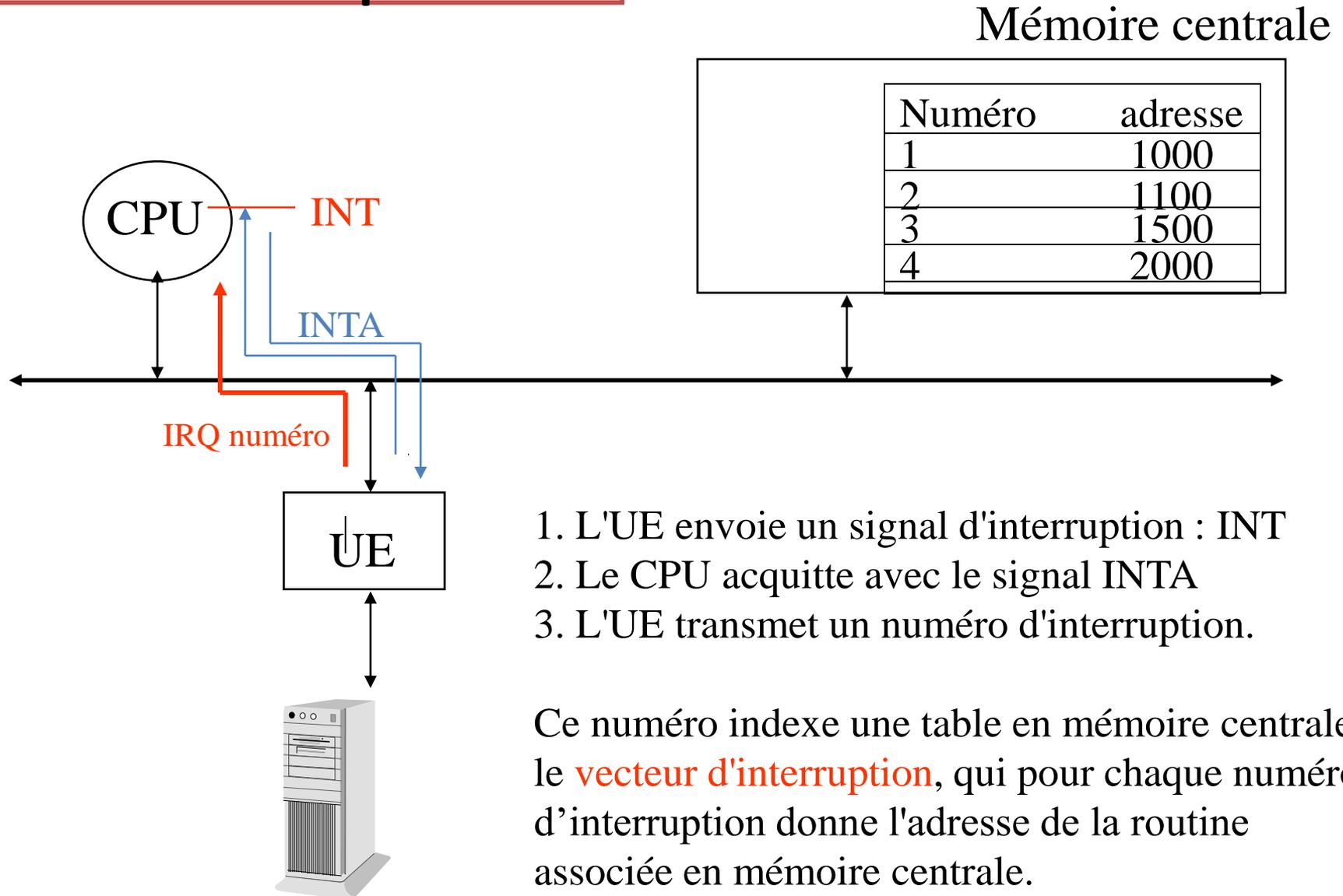


Un périphérique signale un événement au processeur en émettant une interruption matérielle.

Le processeur teste sa ligne d'entrée d'interruption (INT) avant de commencer le traitement de l'instruction suivante du programme qu'il exécute.



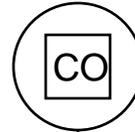
# Les interruptions



# Les Interruptions

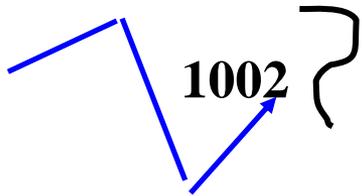
Table des vecteurs d'interruptions (adresse 00000)

IRQ 3	0017
traitant IRQ3	
Programme	

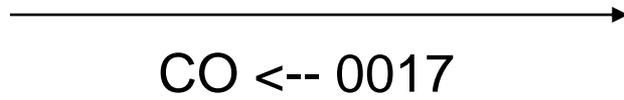


0017  
1002

Programme



Interruption n°3

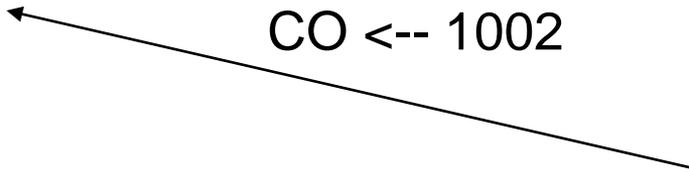


Traitant d'IRQ n°3

0017

Traitement IRQ

1002



✓ La table des vecteurs d'interruptions est à l'adresse 00000 sur un pc; Elle occupe 1 Ko (256 entrées de 4 octets)

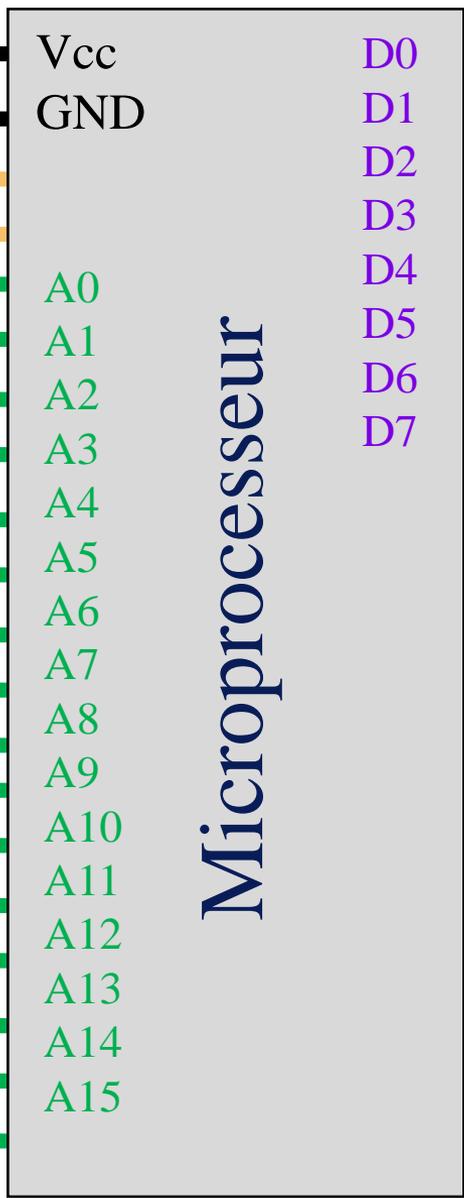
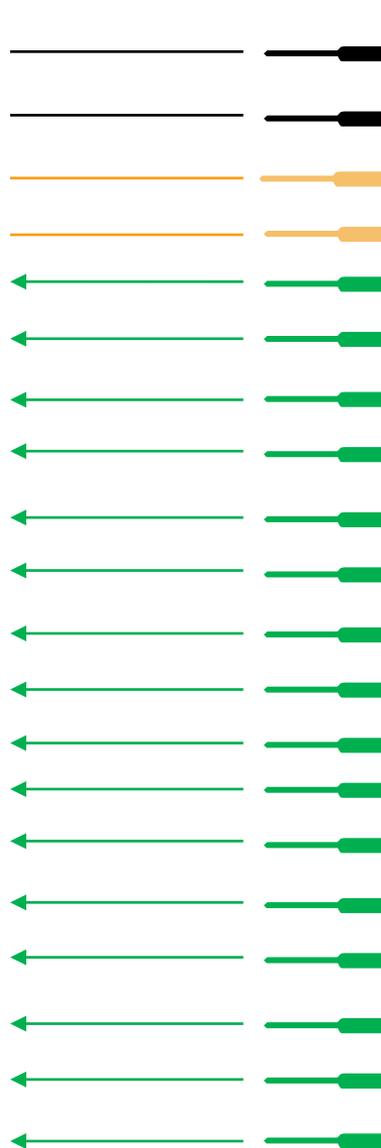
# Processeur (Brochage)



Alimentation  
et masse

INT  
INTA

Adresses



D0  
D1  
D2  
D3  
D4  
D5  
D6  
D7

Données  
Numéro IRQ

Read  
Write

Entrée horloge  
ou cristal

# Hiérarchisation des interruptions

- Le processeur dispose d'une seule broche INT pour recevoir les interruptions.
- Que faire :
  - Si deux interruptions arrivent en même temps ?
  - Si une interruption survient durant le traitement d'une autre interruption ?
- On introduit une notion de priorité (niveau) entre interruptions. Les interruptions sont dites **hiérarchisées**.
  - Exemple : 8 niveaux

Niveau 0 1 2 3 4 5 6 7

Plus forte priorité



Plus faible priorité

# Hiérarchisation des interruptions

- Le processeur dispose d'une seule broche INT pour recevoir les interruptions.
- Que faire :
  - Si deux interruptions arrivent en même temps ?
    - On traite d'abord la plus prioritaire
  - Si une interruption survient durant le traitement d'une autre interruption ?
    - Elle interrompt le traitement en cours seulement si elle est de priorité supérieure
- On introduit une notion de priorité (niveau) entre interruptions.
  - Exemple : 8 niveaux

Niveau 0 1 2 3 4 5 6 7



Plus forte priorité

Plus faible priorité

# Hiérarchisation des interruptions

Arrivée IRQ 5, 6

5  
Traitement IRQ 5,  
IRQ 6 en attente

Arrivée IRQ 7; IRQ 6, 7 en attente

Arrivée IRQ 2  
On interrompt le  
traitement de IRQ5

2  
Traitement IRQ 2,  
IRQ 6, 7 , 5 en attente

Arrivée IRQ 1  
On interrompt le  
traitement de IRQ2

1  
Traitement IRQ 1,  
IRQ 6, 2, 7 , 5  
en attente

5  
Fin traitement IRQ 2,  
Reprise traitement IRQ 5

Fin traitement IRQ 1,  
Reprise traitement IRQ 2

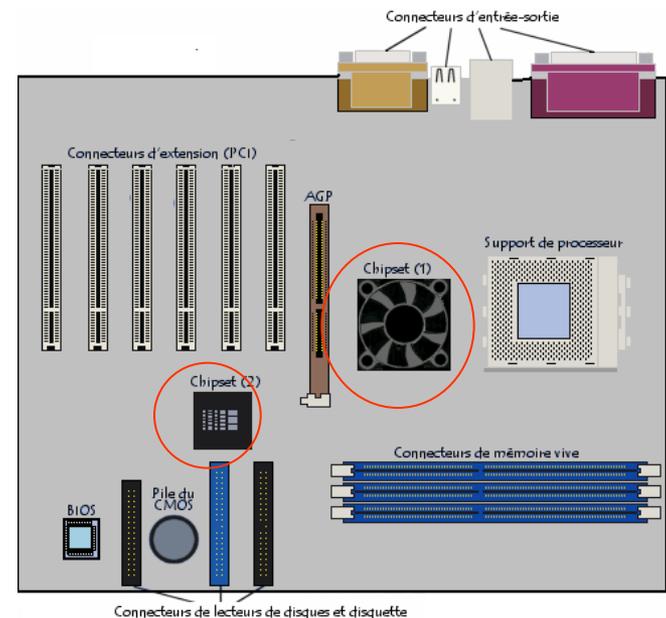
6,7  
Fin traitement IRQ 5,  
Traitement IRQ 6, puis IRQ 7

# Le contrôleur d'interruption

- La gestion des priorités entre interruptions matérielles est à la charge d'un circuit spécifique : le **contrôleur d'interruption**.
- Il reçoit les priorités délivrées par les périphériques du processeur, mémorise les interruptions reçues, arbitre les priorités et délivre l'interruption la plus prioritaire au processeur.

Les ponts gèrent et intègrent :

- L'accès aux bus;
- L'adressage des unités d'échanges;
- Le contrôleur d'interruptions
- Le contrôleur DMA
- Le contrôleur de mémoire centrale



Arrivée IRQ 5, 6

Traitement IRQ 5,  
IRQ 6 en attente



5

AR\_SP5

CO\_RT2 CO

Traitement IRQ 2,  
IRQ 6, 7, 5 en attente

AR\_SP2



1

Traitement IRQ 1,  
IRQ 6, 2, 7, 5  
en attente

CO

Fin traitement IRQ 1,  
Reprise traitement IRQ 2



5

Fin traitement IRQ 2,  
Reprise traitement IRQ 5

Fin traitement IRQ 5,  
Traitement IRQ 6, puis IRQ 7

6,7

num	Adresse routine
1	Adresse CO_RT1
2	Adresse CO_RT2
3	Adresse CO_RT3
4	Adresse CO_RT4
5	Adresse CO_RT5

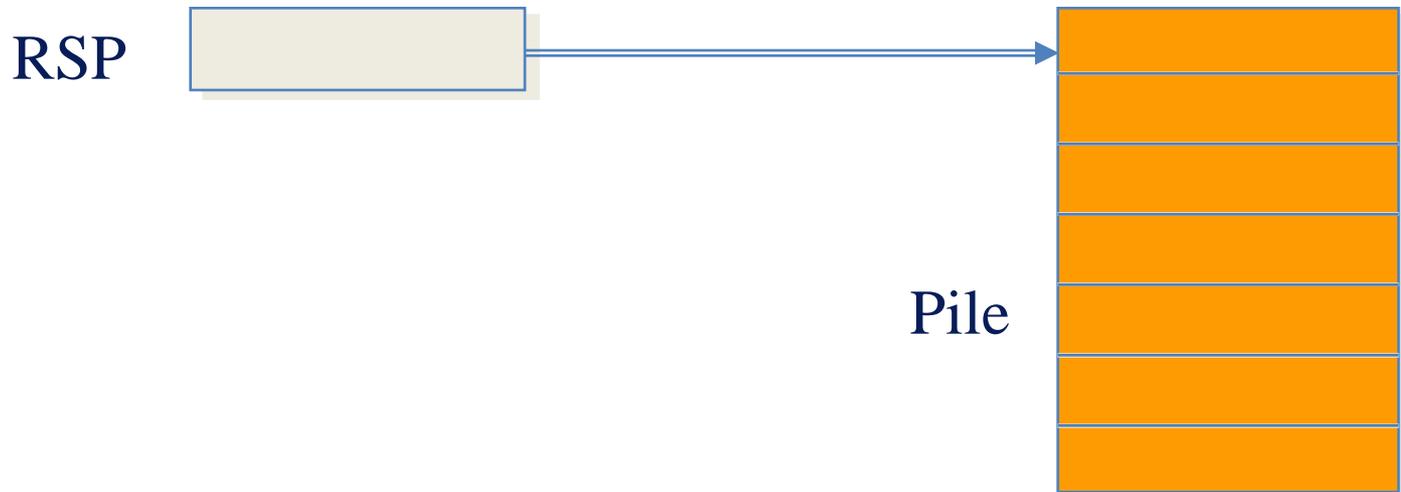
Il faut mémoriser les adresses AR\_SP5, AR\_SP2

▪ Registres (nombre ?)

➤ Pile

# La structure de pile

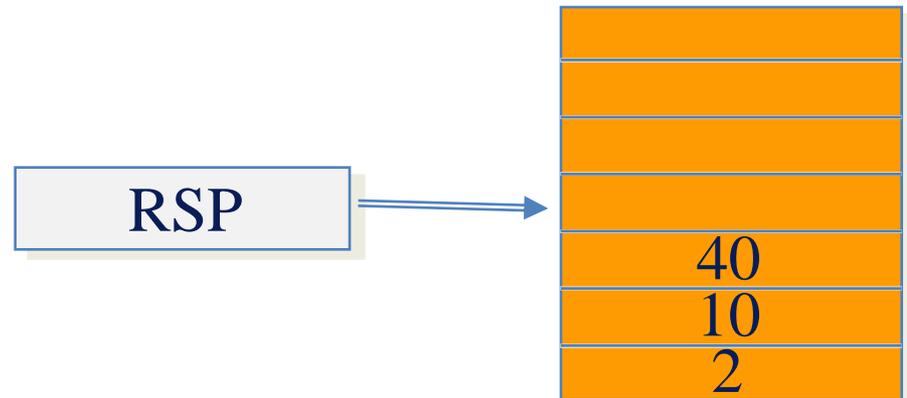
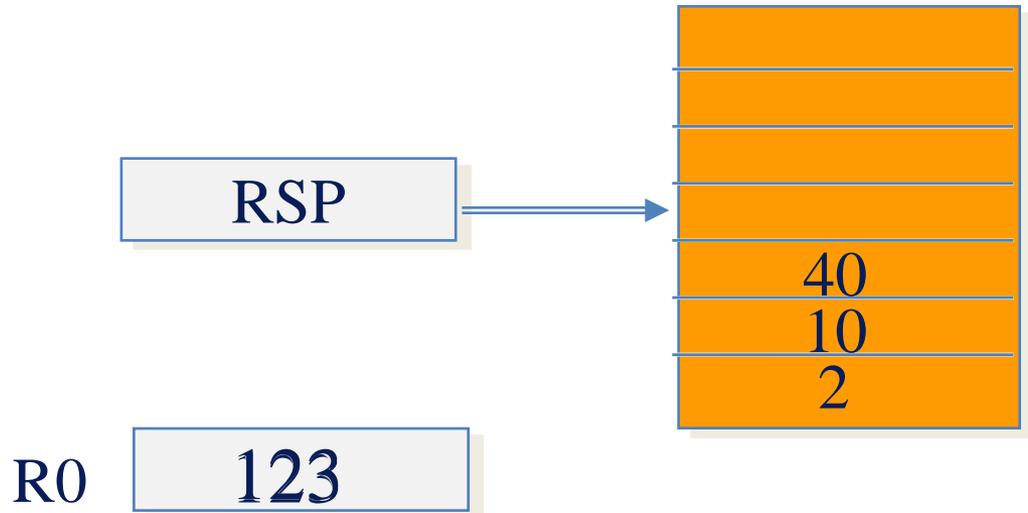
- La pile est une zone de la mémoire centrale constituée d'un ensemble de mots.
- Seul le mot au sommet de la pile est accessible. Son adresse est contenue dans le registre du processeur RSP
- Deux opérations sont seulement autorisées :
  - Ajouter un mot au sommet de la pile (PUSH)
  - Oter le mot au sommet de la pile (POP)



# La structure de pile

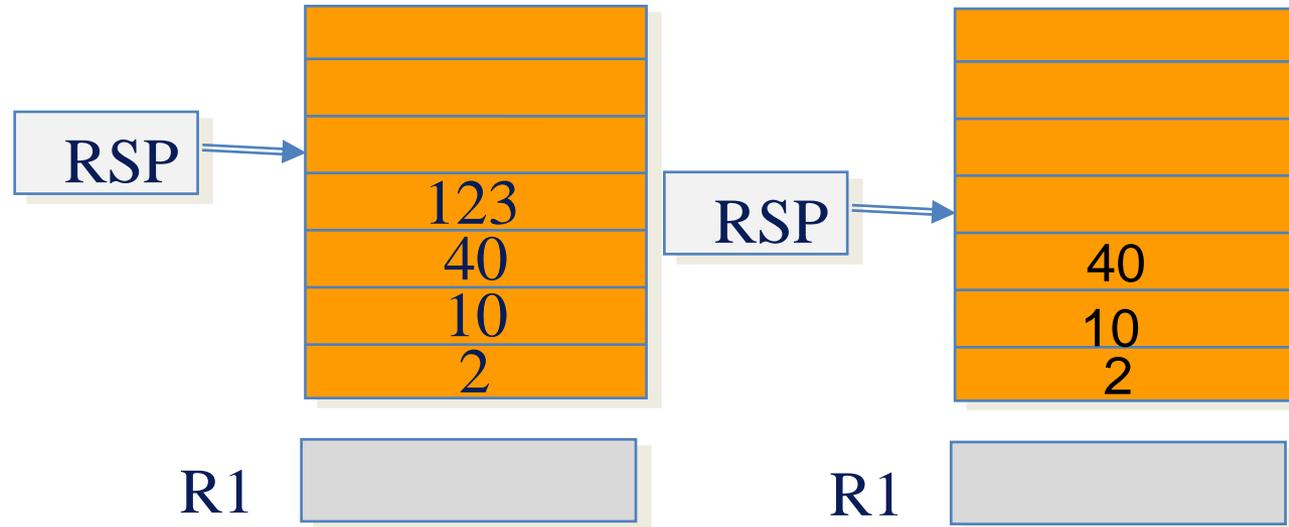
Ajouter un mot au  
sommet de la pile  
PUSH Rg1 R0

Le contenu du  
registre R0 est écrit  
dans le mot dont  
l'adresse est (RSP).  
RSP est incrémenté

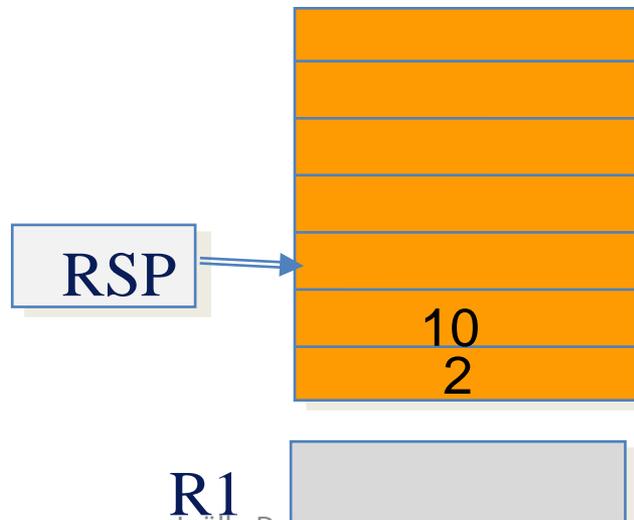


# La structure de pile

Ajouter un mot au  
sommet de la pile  
loop : POP Rg1 R1  
      JMP loop



RSP est décrémenté.  
Le contenu du mot  
dont l'adresse est  
(RSP) est mis dans le  
registre R1.



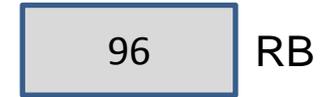
# La structure de pile

La pile est une structure de données régie par une politique **LIFO : Last In First Out**.

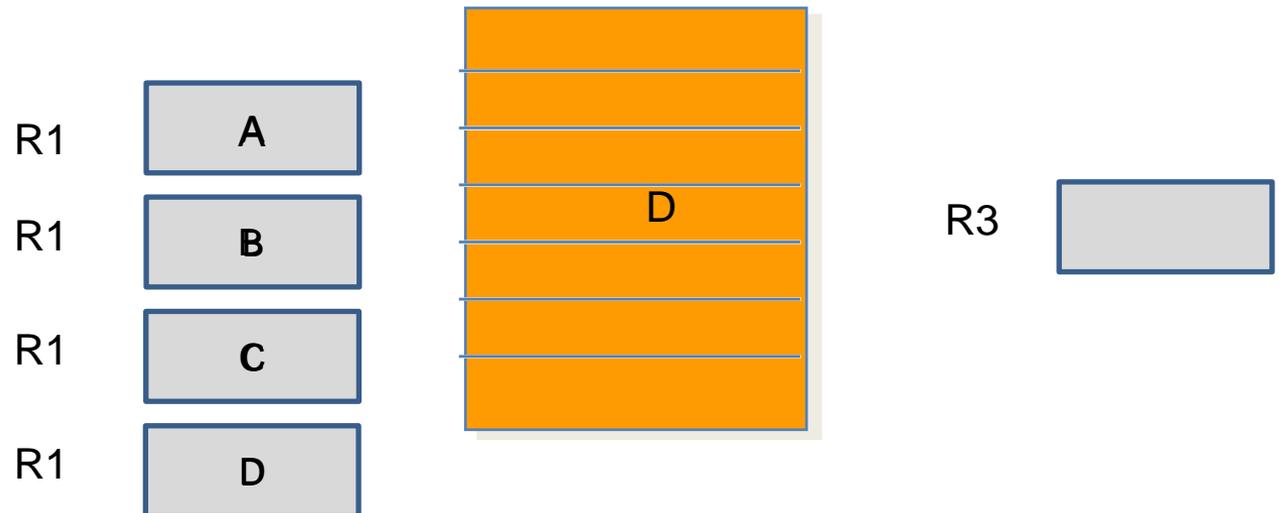
Le dernier élément ajouté à la pile est le premier à en être ôté

100	A
104	B
108	C
112	D
116	E

Mémoire centrale



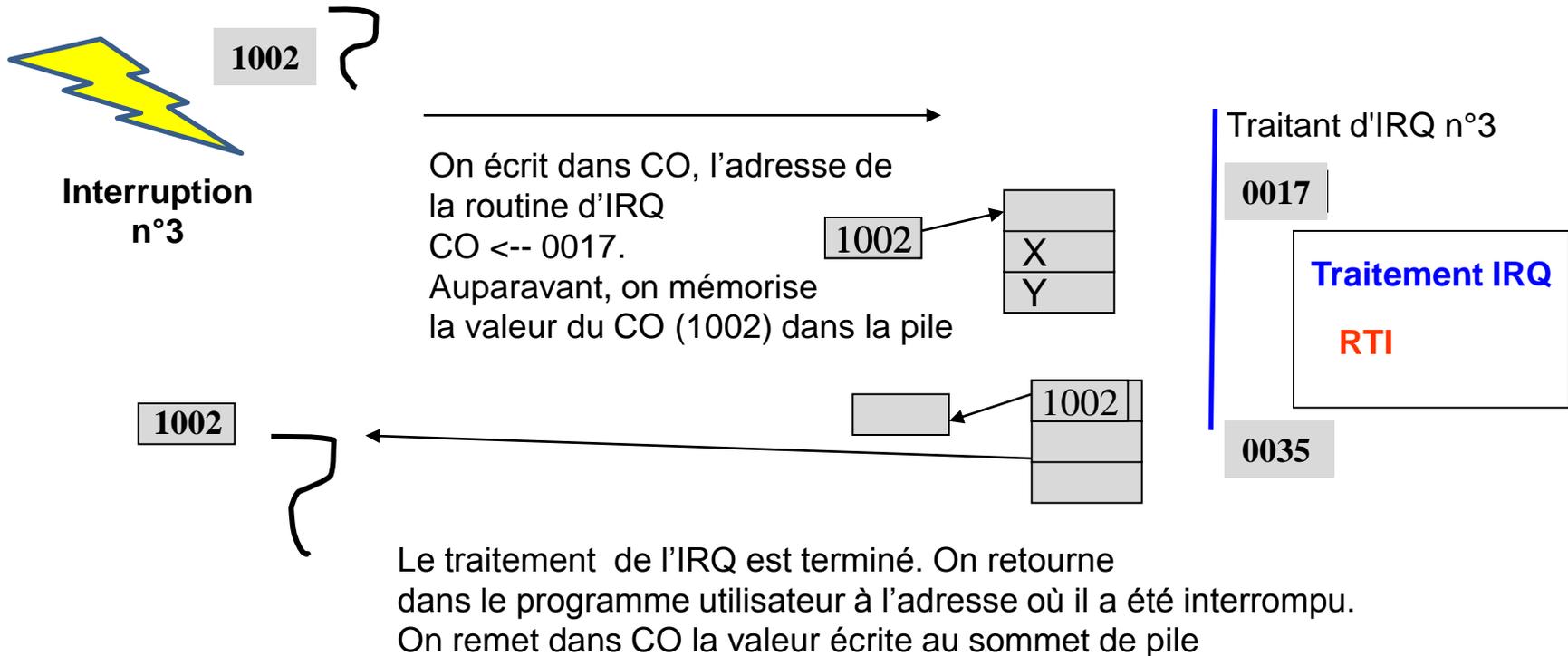
```
Loop : load Im R2 3  
        load B R1 4  
        push Rg1 R1  
        add Im RB 4  
        add Im R2 -1  
        jmp loop  
        pop Rg1 R3
```



# La structure de pile

- La pile est une zone de la mémoire centrale constituée d'un ensemble de mots
- Dans le cadre des interruptions, cette zone est utilisée pour sauvegarder les adresses contenues dans la CO, quand le processeur passe du programme utilisateur aux routines d'interruptions.

Programme



# Hiérarchisation des interruptions

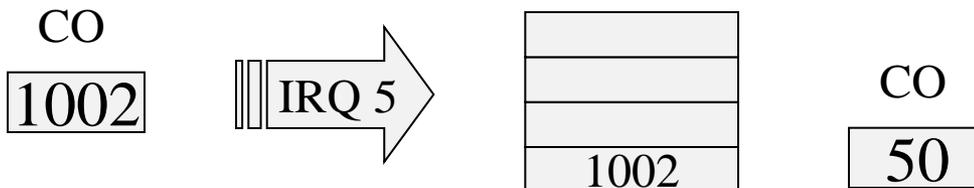
IRQ 1	10
IRQ 2	20
IRQ 3	30
IRQ 4	40
IRQ 5	50
IRQ 6	60
IRQ 7	70
IRQ 8	80

10	Routine irq1
20	Routine irq2
30	Routine irq3
40	Routine irq4
50	Routine irq5
60	Routine irq6
70	Routine irq7
80	Routine irq8

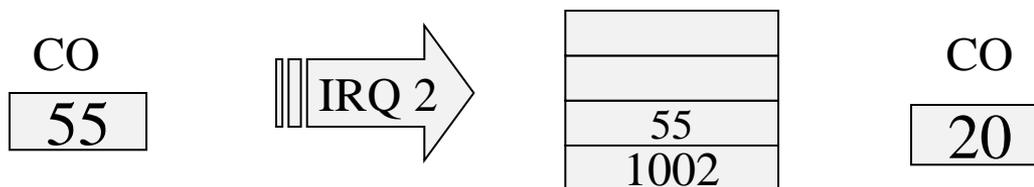
1000	I1
1001	I2
→ 1002	I3
1003	I4
1004	I5
1005	I6

Programme X

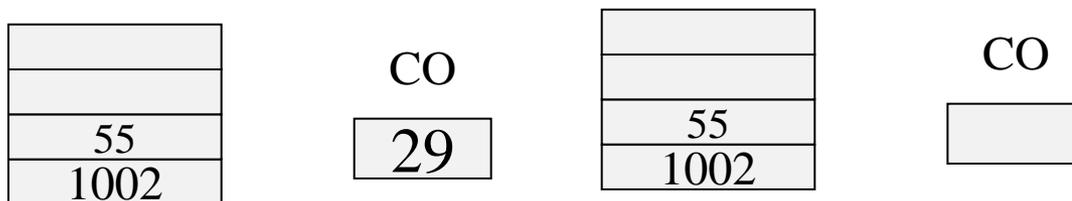
Le processeur exécute le programme X, arrivée de IRQ 5



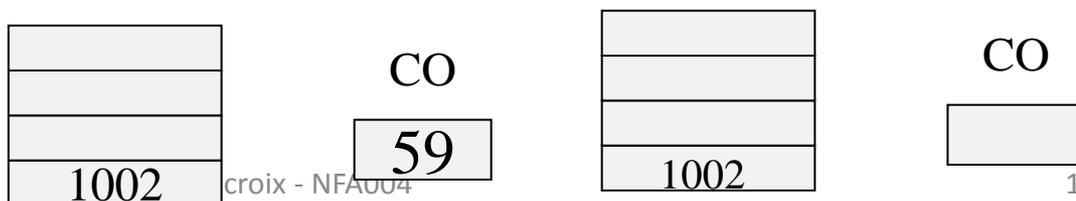
Le processeur exécute le traitant IRQ 5, arrivée de IRQ 2



Le processeur achève le traitant IRQ 2, reprise du traitant IRQ 5



Le processeur achève le traitant IRQ 5, reprise du programme X



# Le masquage des interruptions

- **Une interruption masquée** n'est pas prise en compte par le processeur lorsque celle-ci survient.
- Deux niveaux de masquage :
  - Au niveau du processeur
  - Au niveau du contrôleur

# Le masquage des interruptions

## Registre PSW du processeur

S	O	C	Z	I	
---	---	---	---	---	--

I = 1, Interruptions masquées

I = 0, Interruptions démasquées

- O : positionné à 1 si Overflow, 0 sinon
- Z : positionné à 1 si résultat opération nul, 0 sinon
- C : positionné à 1 si carry, 0 sinon
- S : positionné à 0 si résultat opération positif, 1 sinon

- Au niveau du processeur, masquage, démasquage global
- EI : Enable Interrupt
  - les interruptions sont démasquées
- DI : Disable Interrupt
  - les interruptions sont masquées

```
LOAD Im R1 10
```

```
DI
```

```
ADD Im R1 20
```

```
ADD D R1 100
```

```
ADD I R1 20
```

```
EI
```

```
MUL R R1 50
```

Interruptions  
masquées

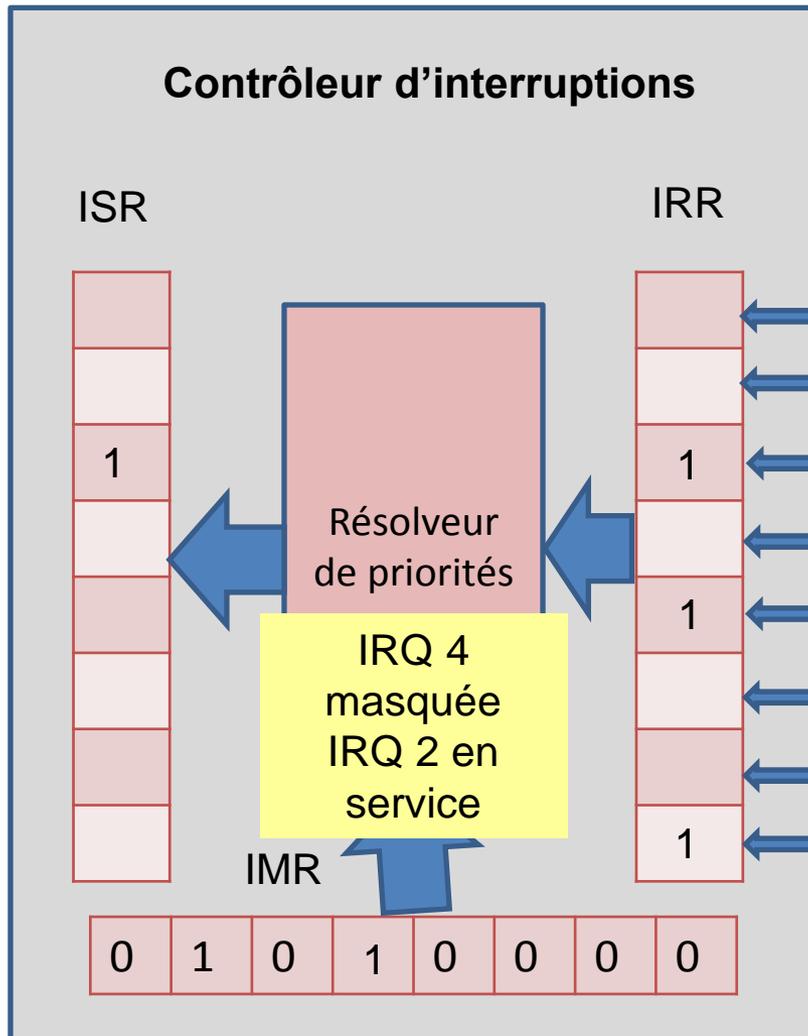


IRQ attente



IRQ Prise en compte

# Le masquage des interruptions



- Au niveau du contrôleur, masquage, démasquage par niveau

IRQ 0

Les interruptions 2, 4 et 7 sont délivrées au contrôleur. L'interruption 4 est masquée; elle est ignorée.

IRQ 1

L'arbitrage se fait entre les interruptions 2 et 7

IRQ 2

L'interruption 2 entre en service

IRQ 3

IRQ 4

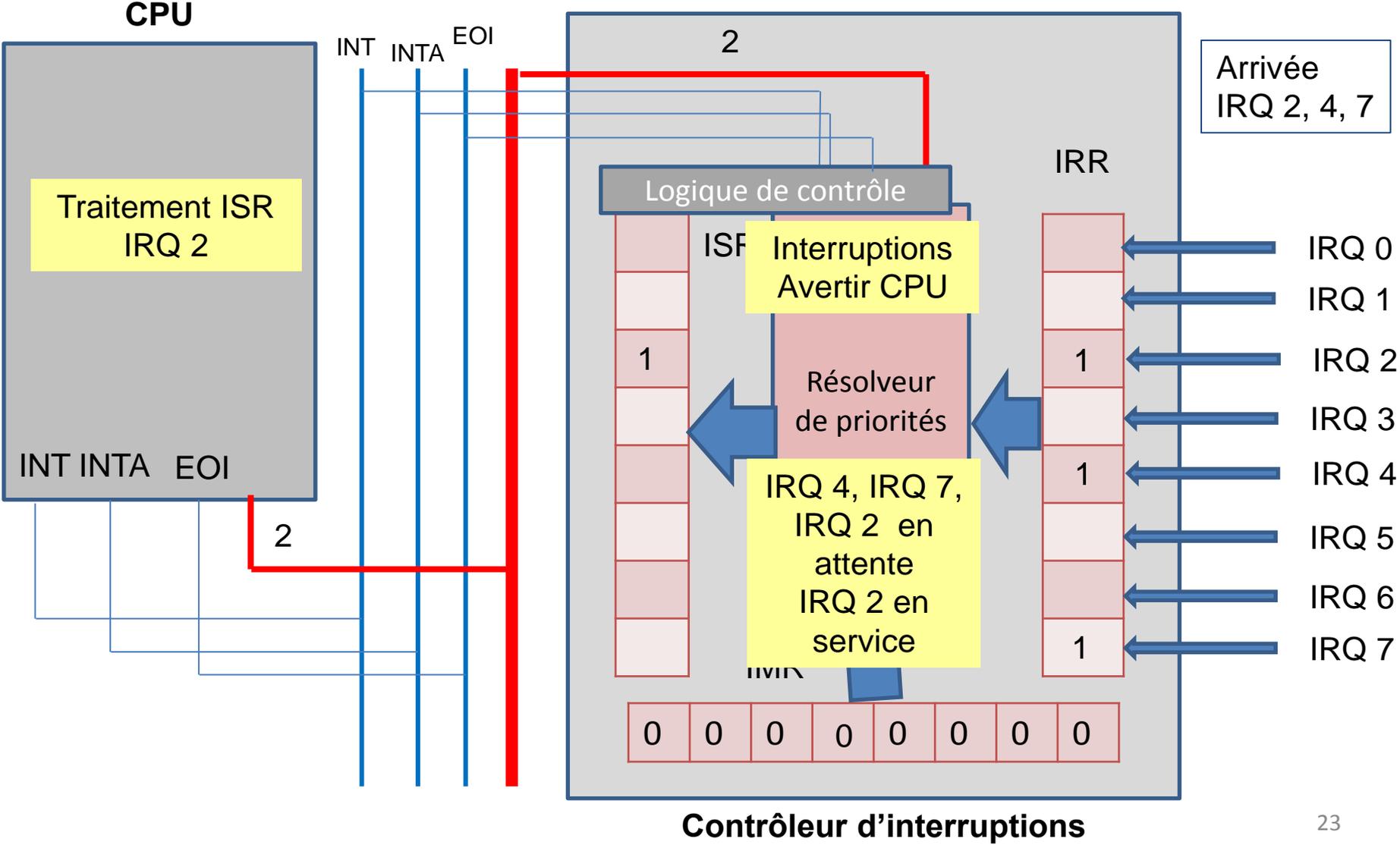
IRQ 5

IRQ 6

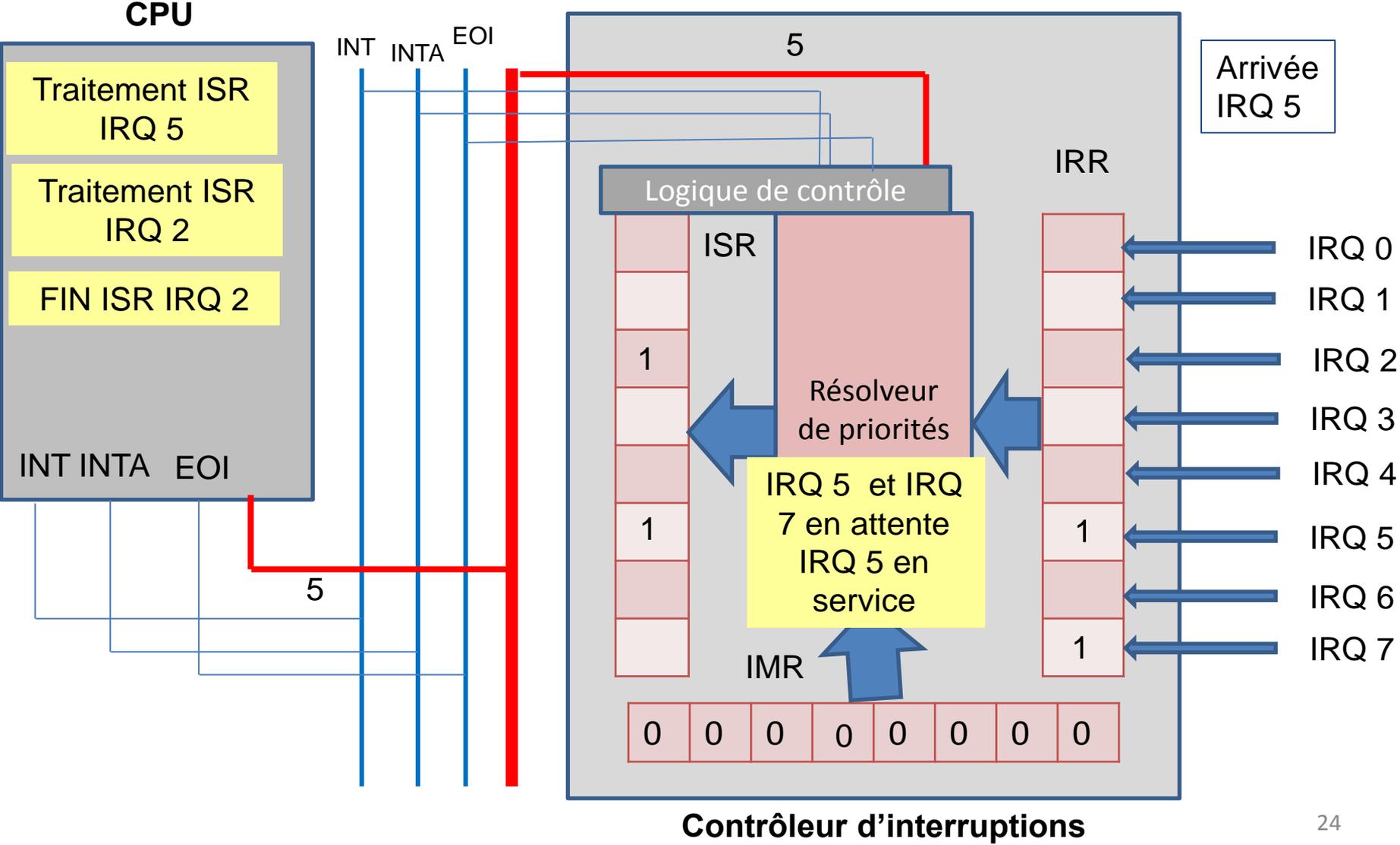
IRQ 7

IRR : registre des requêtes d'interruptions  
ISR : registre des interruptions en services  
IMR : registre des interruptions masquées

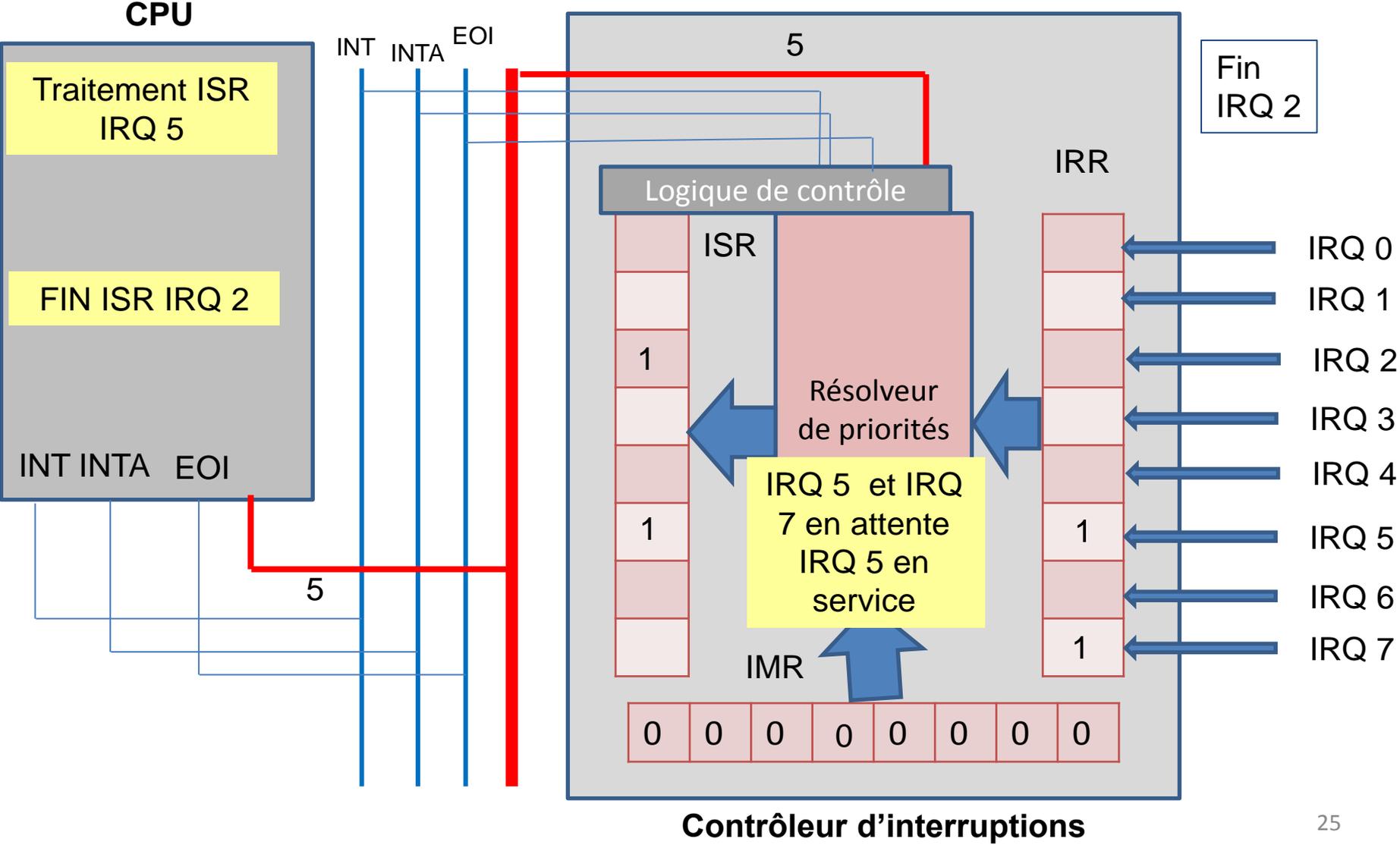
# Déroulement des interruptions



# Déroulement des interruptions



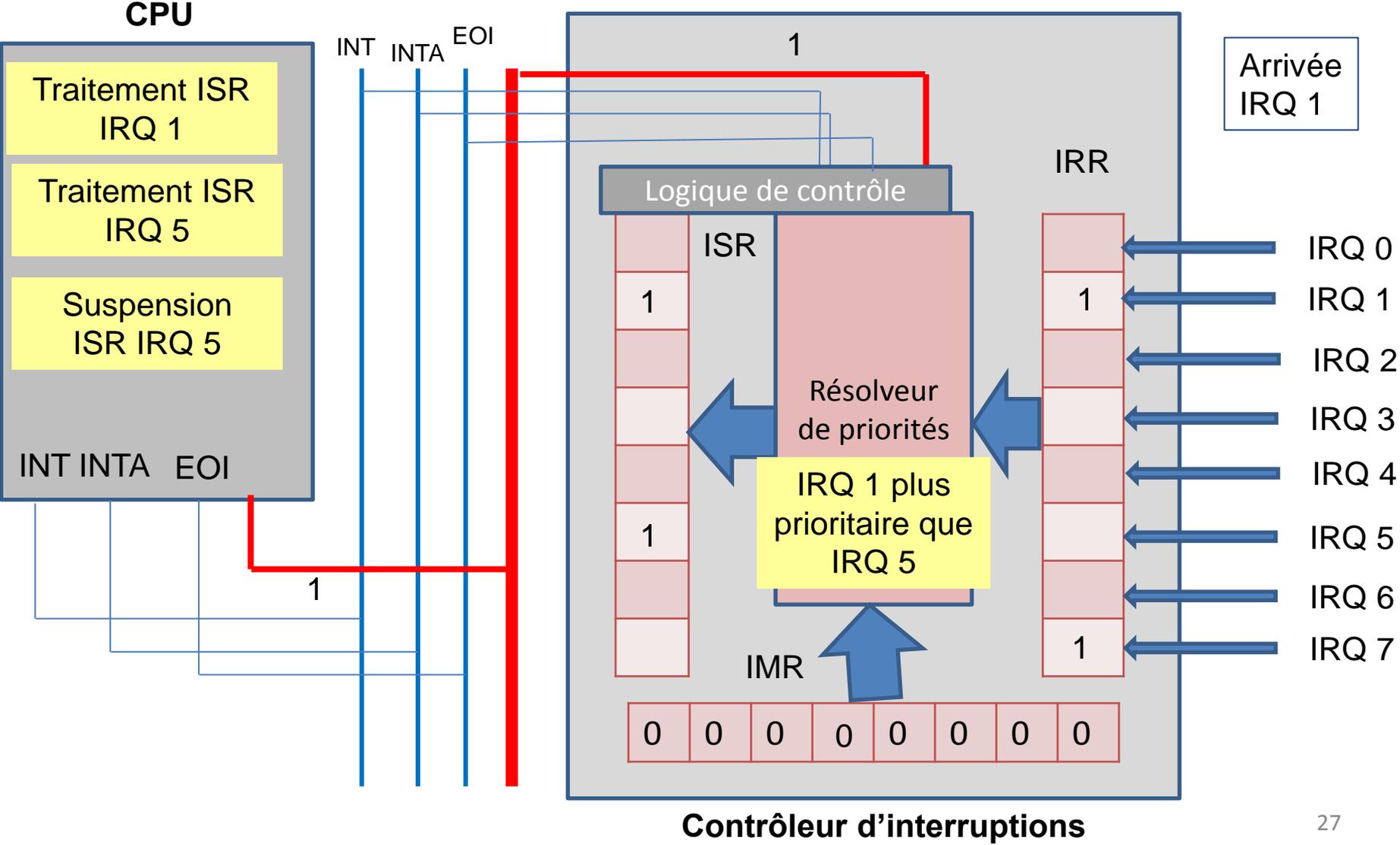
# Déroulement des interruptions



# Déroulement des interruptions

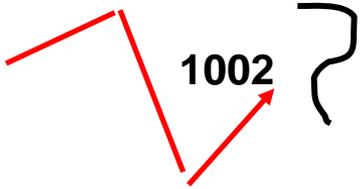
1. Le contrôleur d'interruptions reçoit les signaux  $IRQ_i$  émis par les périphériques sur ses bornes  $IRQ_i$ . Les interruptions reçues sont mémorisées dans le registre IRR. Le contrôleur active le signal INT
2. Le processeur prend en compte le signal. Si l'indicateur  $I=1$  au niveau du registre d'état, le signal est ignoré, sinon, la demande d'interruption est acceptée.
3. Si la demande est acceptée, le processeur met sa sortie INTA pour indiquer au contrôleur qu'il prend en compte sa demande.
4. En réponse, le contrôleur d'interruption arbitre les priorités entre les IRQs en attente et place le numéro de l'interruption  $IRQ_i$  de plus forte priorité sur le bus de données. Le bit correspondant dans ISR est mis à 1 et remis à 0 dans IRR.
5. 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). Le processeur sauvegarde le CO courant sur la pile et le charge avec l'adresse du traitant d'interruption.
6. La procédure traitant l'interruption se déroule. Lorsqu'elle est terminée, le processeur avertit le contrôleur en activant le signal EOI. Si des interruptions sont en attente, le process reprend.

# Déroulement des interruptions



# Les Interruptions matérielles → asynchrones

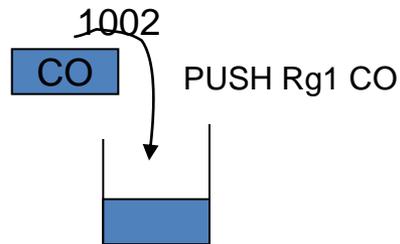
Mode utilisateur  
Programme



Interruption  
n°3

1002

Mode noyau



CO ← 0017

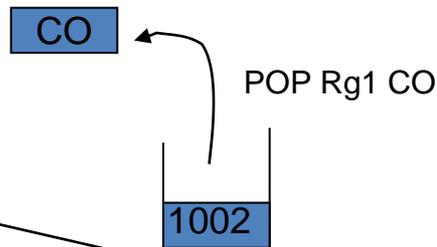
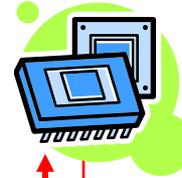


Table des vecteurs  
d'interruptions  
(adresse 00000)

IRQ 3	0017
traitant IRQ3	
Programme	

0017

1002



Irq 3



Traitant d'IRQ n°3

0017

Traitement IRQ

Les interruptions externes ou matérielles sont émises par les périphériques du processeur (fin d'écriture disques, plus de papier imprimante...). Ce sont les **IRQs**.

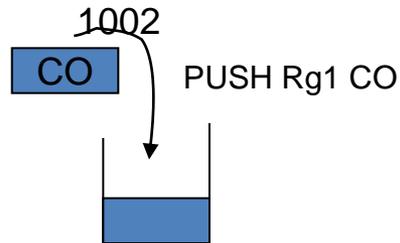
# Les Interruptions logicielles trappes → synchrones

Mode utilisateur  
Programme

I = 0;  
A = 12,  
A = A / I;

1002

Mode noyau



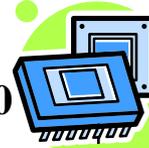
CO ← 0040

Levée exception 0

Table des vecteurs  
d'interruptions  
(adresse 00000)

0040  
EXCP 0  
1002

Div A, 0



EXCP 0

EXCP 0	0040
traitant EXCP 0	
Programme	



Traitant Exception 0

0040

Traitement Exception 0  
Arrêt du programme

Les interruptions internes ou logicielles sont émises par le processeur lui-même lorsqu'il rencontre une erreur dans l'exécution du programme (division par zéro, accès mémoire illégal). Ce sont les **trappes**.

# Instructions machine / assembleur

Instructions liées  
aux interruptions

EI : Autoriser les interruptions

DI : Masquer les interruptions

RTI : retour de traitement d'interruption

Instructions liées  
à la pile

POP Rg1 R1 : Le sommet de pile est mis dans R1

PUSH Rg1 R1 : R1 est placé au sommet de la pile