

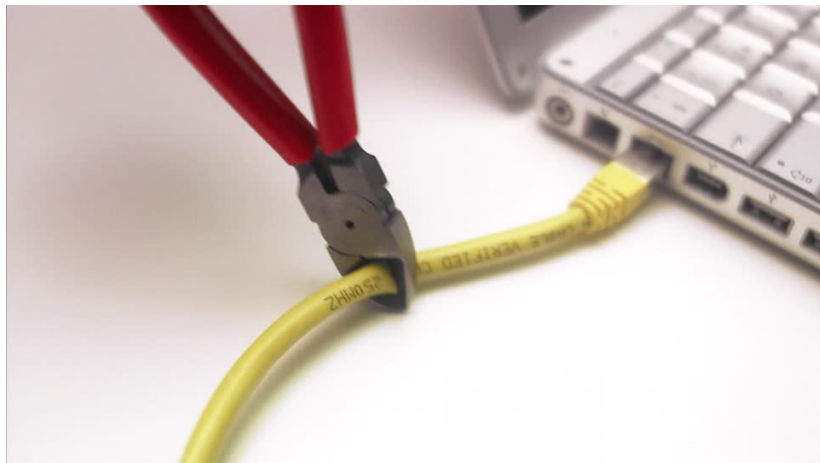
# Architecture des ordinateurs VIII

Frédéric Boulanger

CentraleSupélec



# Interruptions



# Interruptions

## Entrées-sorties

```
@loop    ldr r0, 0x9234    % lecture de l'état des boutons
         cmp r0, #1      % bouton 0 enfoncé ?
         beq on          % brancher vers l'allumage
         cmp r0, #2      % bouton 1 enfoncé ?
         beq off         % brancher vers l'extinction
         b  loop         % boucler

@on      mov r0, #1      % allumage = mettre la bascule à 1
         b  output      % brancher vers l'écriture

@off     mov r0, #0      % extinction = mettre la bascule à 0

@output  str r0, 0xABCD  % écriture dans la bascule
         b  loop         % boucler
```

- ▶ Le processeur passe son temps à scruter les entrées



# Interruptions

## Entrées-sorties

```
@loop    ldr r0, 0x9234    % lecture de l'état des boutons
         cmp r0, #1      % bouton 0 enfoncé ?
         beq on          % brancher vers l'allumage
         cmp r0, #2      % bouton 1 enfoncé ?
         beq off         % brancher vers l'extinction
         b loop          % boucler

@on      mov r0, #1      % allumage = mettre la bascule à 1
         b output       % brancher vers l'écriture

@off     mov r0, #0      % extinction = mettre la bascule à 0

@output  str r0, 0xABCD  % écriture dans la bascule
         b loop          % boucler
```

- ▶ Le processeur passe son temps à scruter les entrées
- ▶ Il peut rater un appui sur un bouton



# Interruptions

## Scrutation (polling)

- ▶ Consultation périodique des entrées



# Interruptions

## Scrutation (polling)

- ▶ Consultation périodique des entrées
- ▶ Prendrait plus de 100% du temps avec quelques périphériques



# Interruptions

## Scrutation (polling)

- ▶ Consultation périodique des entrées
- ▶ Prendrait plus de 100% du temps avec quelques périphériques

## Exemple

# Interruptions

## Scrutation (polling)

- ▶ Consultation périodique des entrées
- ▶ Prendrait plus de 100% du temps avec quelques périphériques

## Exemple

- ▶ vous attendez un appel téléphonique



# Interruptions

## Scrutation (polling)

- ▶ Consultation périodique des entrées
- ▶ Prendrait plus de 100% du temps avec quelques périphériques

## Exemple

- ▶ vous attendez un appel téléphonique
- ▶ votre portable est en mode silencieux

# Interruptions

## Scrutation (polling)

- ▶ Consultation périodique des entrées
- ▶ Prendrait plus de 100% du temps avec quelques périphériques

## Exemple

- ▶ vous attendez un appel téléphonique
- ▶ votre portable est en mode silencieux
  - ▶ vous devez regarder l'écran assez souvent

# Interruptions

## Scrutation (polling)

- ▶ Consultation périodique des entrées
- ▶ Prendrait plus de 100% du temps avec quelques périphériques

## Exemple

- ▶ vous attendez un appel téléphonique
- ▶ votre portable est en mode silencieux
  - ▶ vous devez regarder l'écran assez souvent
  - ▶ vous risquez de rater l'appel

# Interruptions

## Interruptions

- ▶ vous attendez un appel téléphonique

# Interruptions

## Interruptions

- ▶ vous attendez un appel téléphonique
- ▶ votre portable sonne ou vibre

# Interruptions

## Interruptions

- ▶ vous attendez un appel téléphonique
- ▶ votre portable sonne ou vibre
  - ▶ vous pouvez faire autre chose en attendant



# Interruptions

## Interruptions

- ▶ vous attendez un appel téléphonique
- ▶ votre portable sonne ou vibre
  - ▶ vous pouvez faire autre chose en attendant
  - ▶ vous ne risquez pas de rater l'appel



# Interruptions

## Interruptions

- ▶ vous attendez un appel téléphonique
- ▶ votre portable sonne ou vibre
  - ▶ vous pouvez faire autre chose en attendant
  - ▶ vous ne risquez pas de rater l'appel

## Avec un ordinateur



# Interruptions

## Interruptions

- ▶ vous attendez un appel téléphonique
- ▶ votre portable sonne ou vibre
  - ▶ vous pouvez faire autre chose en attendant
  - ▶ vous ne risquez pas de rater l'appel

## Avec un ordinateur

- ▶ lors d'une demande d'interruption :

# Interruptions

## Interruptions

- ▶ vous attendez un appel téléphonique
- ▶ votre portable sonne ou vibre
  - ▶ vous pouvez faire autre chose en attendant
  - ▶ vous ne risquez pas de rater l'appel

## Avec un ordinateur

- ▶ lors d'une demande d'interruption :
  - ▶ l'exécution du programme est interrompue

# Interruptions

## Interruptions

- ▶ vous attendez un appel téléphonique
- ▶ votre portable sonne ou vibre
  - ▶ vous pouvez faire autre chose en attendant
  - ▶ vous ne risquez pas de rater l'appel

## Avec un ordinateur

- ▶ lors d'une demande d'interruption :
  - ▶ l'exécution du programme est interrompue
  - ▶ le processeur saute à une adresse particulière pour traiter l'interruption



# Interruptions

## Conditions

- ▶ le traitement de l'interruption doit être transparent pour le programme interrompu



# Interruptions

## Conditions

- ▶ le traitement de l'interruption doit être transparent pour le programme interrompu
  - ▶ il faut sauvegarder le compteur ordinal



# Interruptions

## Conditions

- ▶ le traitement de l'interruption doit être transparent pour le programme interrompu
  - ▶ il faut sauvegarder le compteur ordinal
  - ▶ ainsi que le registre d'état



# Interruptions

## Conditions

- ▶ le traitement de l'interruption doit être transparent pour le programme interrompu
  - ▶ il faut sauvegarder le compteur ordinal
  - ▶ ainsi que le registre d'état
  - ▶ et les registres modifiés par le traitement de l'interruption



# Interruptions

## Conditions

- ▶ le traitement de l'interruption doit être transparent pour le programme interrompu
  - ▶ il faut sauvegarder le compteur ordinal
  - ▶ ainsi que le registre d'état
  - ▶ et les registres modifiés par le traitement de l'interruption

## Exemple

```
cmp r1, r2      % SR <- N et Z en fonction de r1-r2  
beq ok
```



# Interruptions

## Conditions

- ▶ le traitement de l'interruption doit être transparent pour le programme interrompu
  - ▶ il faut sauvegarder le compteur ordinal
  - ▶ ainsi que le registre d'état
  - ▶ et les registres modifiés par le traitement de l'interruption

## Exemple

```
cmp r1, r2      % SR <- N et Z en fonction de r1-r2  
beq ok
```

- ▶ si une interruption se produit entre ces instructions :

# Interruptions

## Conditions

- ▶ le traitement de l'interruption doit être transparent pour le programme interrompu
  - ▶ il faut sauvegarder le compteur ordinal
  - ▶ ainsi que le registre d'état
  - ▶ et les registres modifiés par le traitement de l'interruption

## Exemple

```
cmp r1, r2      % SR <- N et Z en fonction de r1-r2  
beq ok
```

- ▶ si une interruption se produit entre ces instructions :
  - ▶ si le SR est modifié, le branchement ne sera pas correct

# Interruptions

## Conditions

- ▶ le traitement de l'interruption doit être transparent pour le programme interrompu
  - ▶ il faut sauvegarder le compteur ordinal
  - ▶ ainsi que le registre d'état
  - ▶ et les registres modifiés par le traitement de l'interruption

## Exemple

```
cmp r1, r2      % SR <- N et Z en fonction de r1-r2  
beq ok
```

- ▶ si une interruption se produit entre ces instructions :
  - ▶ si le SR est modifié, le branchement ne sera pas correct
  - ▶ si r1 ou r2 sont modifiés, le programme n'aura pas le bon comportement



# Interruptions

## Détection

- ▶ une interruption est provoquée par changement d'état d'un périphérique



# Interruptions

## Détection

- ▶ une interruption est provoquée par changement d'état d'un périphérique
  - ▶ passage d'un signal de 0 à 1 ou vice-versa



# Interruptions

## Détection

- ▶ une interruption est provoquée par changement d'état d'un périphérique
  - ▶ passage d'un signal de 0 à 1 ou vice-versa
  - ▶ si le périphérique est configuré pour générer une demande d'interruption



# Interruptions

## Détection

- ▶ une interruption est provoquée par changement d'état d'un périphérique
  - ▶ passage d'un signal de 0 à 1 ou vice-versa
  - ▶ si le périphérique est configuré pour générer une demande d'interruption
  - ▶ il rend actif un signal d'entrée du processeur (patte IRQ)



# Interruptions

## Détection

- ▶ une interruption est provoquée par changement d'état d'un périphérique
  - ▶ passage d'un signal de 0 à 1 ou vice-versa
  - ▶ si le périphérique est configuré pour générer une demande d'interruption
  - ▶ il rend actif un signal d'entrée du processeur (patte IRQ)

## Traitement



# Interruptions

## Détection

- ▶ une interruption est provoquée par changement d'état d'un périphérique
  - ▶ passage d'un signal de 0 à 1 ou vice-versa
  - ▶ si le périphérique est configuré pour générer une demande d'interruption
  - ▶ il rend actif un signal d'entrée du processeur (patte IRQ)

## Traitement

- ▶ à la fin de l'exécution de chaque instruction, le séquenceur vérifie le signal IRQ

# Interruptions

## Détection

- ▶ une interruption est provoquée par changement d'état d'un périphérique
  - ▶ passage d'un signal de 0 à 1 ou vice-versa
  - ▶ si le périphérique est configuré pour générer une demande d'interruption
  - ▶ il rend actif un signal d'entrée du processeur (patte IRQ)

## Traitement

- ▶ à la fin de l'exécution de chaque instruction, le séquenceur vérifie le signal IRQ
  - ▶ s'il est actif et que les interruptions sont autorisées

# Interruptions

## Détection

- ▶ une interruption est provoquée par changement d'état d'un périphérique
  - ▶ passage d'un signal de 0 à 1 ou vice-versa
  - ▶ si le périphérique est configuré pour générer une demande d'interruption
  - ▶ il rend actif un signal d'entrée du processeur (patte IRQ)

## Traitement

- ▶ à la fin de l'exécution de chaque instruction, le séquenceur vérifie le signal IRQ
  - ▶ s'il est actif et que les interruptions sont autorisées
  - ▶ il sauvegarde le PC et le SR



# Interruptions

## Détection

- ▶ une interruption est provoquée par changement d'état d'un périphérique
  - ▶ passage d'un signal de 0 à 1 ou vice-versa
  - ▶ si le périphérique est configuré pour générer une demande d'interruption
  - ▶ il rend actif un signal d'entrée du processeur (patte IRQ)

## Traitement

- ▶ à la fin de l'exécution de chaque instruction, le séquenceur vérifie le signal IRQ
  - ▶ s'il est actif et que les interruptions sont autorisées
  - ▶ il sauvegarde le PC et le SR
  - ▶ il charge le PC avec une adresse prédéterminée (vecteur d'interruption)



# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu



# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu
- ▶ on utilise généralement un système d'exploitation



# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu
- ▶ on utilise généralement un système d'exploitation
- ▶ ou un *moniteur* qui fournit des services :



# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu
- ▶ on utilise généralement un système d'exploitation
- ▶ ou un *moniteur* qui fournit des services :
  - ▶ pour activer ou désactiver les interruptions





# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu
- ▶ on utilise généralement un système d'exploitation
- ▶ ou un *moniteur* qui fournit des services :
  - ▶ pour activer ou désactiver les interruptions
  - ▶ pour configurer la génération d'une demande d'interruption quand une patte change de niveau (ligne d'interruption)



# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu
- ▶ on utilise généralement un système d'exploitation
- ▶ ou un *moniteur* qui fournit des services :
  - ▶ pour activer ou désactiver les interruptions
  - ▶ pour configurer la génération d'une demande d'interruption quand une patte change de niveau (ligne d'interruption)
  - ▶ pour indiquer le code qui traitera cette interruption



# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu
- ▶ on utilise généralement un système d'exploitation
- ▶ ou un *moniteur* qui fournit des services :
  - ▶ pour activer ou désactiver les interruptions
  - ▶ pour configurer la génération d'une demande d'interruption quand une patte change de niveau (ligne d'interruption)
  - ▶ pour indiquer le code qui traitera cette interruption
- ▶ le système se charge de sauvegarder tout ce qui doit l'être

# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu
- ▶ on utilise généralement un système d'exploitation
- ▶ ou un *moniteur* qui fournit des services :
  - ▶ pour activer ou désactiver les interruptions
  - ▶ pour configurer la génération d'une demande d'interruption quand une patte change de niveau (ligne d'interruption)
  - ▶ pour indiquer le code qui traitera cette interruption
- ▶ le système se charge de sauvegarder tout ce qui doit l'être

## Avec MicroPython



# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu
- ▶ on utilise généralement un système d'exploitation
- ▶ ou un *moniteur* qui fournit des services :
  - ▶ pour activer ou désactiver les interruptions
  - ▶ pour configurer la génération d'une demande d'interruption quand une patte change de niveau (ligne d'interruption)
  - ▶ pour indiquer le code qui traitera cette interruption
- ▶ le système se charge de sauvegarder tout ce qui doit l'être

## Avec MicroPython

- ▶ on dispose de 22 lignes d'interruptions



# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu
- ▶ on utilise généralement un système d'exploitation
- ▶ ou un *moniteur* qui fournit des services :
  - ▶ pour activer ou désactiver les interruptions
  - ▶ pour configurer la génération d'une demande d'interruption quand une patte change de niveau (ligne d'interruption)
  - ▶ pour indiquer le code qui traitera cette interruption
- ▶ le système se charge de sauvegarder tout ce qui doit l'être

## Avec MicroPython

- ▶ on dispose de 22 lignes d'interruptions
  - ▶ 16 proviennent des pattes d'entrées-sorties



# Interruptions

## Dans la pratique

- ▶ il est rare de travailler sur un processeur nu
- ▶ on utilise généralement un système d'exploitation
- ▶ ou un *moniteur* qui fournit des services :
  - ▶ pour activer ou désactiver les interruptions
  - ▶ pour configurer la génération d'une demande d'interruption quand une patte change de niveau (ligne d'interruption)
  - ▶ pour indiquer le code qui traitera cette interruption
- ▶ le système se charge de sauvegarder tout ce qui doit l'être

## Avec MicroPython

- ▶ on dispose de 22 lignes d'interruptions
  - ▶ 16 proviennent des pattes d'entrées-sorties
  - ▶ 6 sont liées à des sources internes à la carte



# Interruptions

## Exemple avec MicroPython

```
import pyb
def traitement(l):
    print("Interruption sur ", l)

ligne = pyb.ExtInt(pyb.Pin("X17"), # bouton USR
                  pyb.ExtInt.IRQ_FALLING,
                  pyb.Pin.PULL_UP,
                  traitement)

pyb.enable_irq(True)
```





# Interruptions

## Exemple avec MicroPython

```
import pyb
def traitement(l):
    print("Interruption sur ", l)

ligne = pyb.ExtInt(pyb.Pin("X17"), # bouton USB
                  pyb.ExtInt.IRQ_FALLING,
                  pyb.Pin.PULL_UP,
                  traitement)

pyb.enable_irq(True)
```

À chaque passage à 0 de la patte X17, la fonction `traitement` sera appelée et affichera un message.



# Timers



# Timers



- ▶ Un timer est un périphérique muni d'un compteur qui est décrémenté à chaque impulsion d'une horloge.

# Timers



- ▶ Un timer est un périphérique muni d'un compteur qui est décrémenté à chaque impulsion d'une horloge.
- ▶ Lorsque que le compteur atteint 0, il génère une demande d'interruption

# Timers



- ▶ Un timer est un périphérique muni d'un compteur qui est décrémenté à chaque impulsion d'une horloge.
- ▶ Lorsque que le compteur atteint 0, il génère une demande d'interruption
- ▶ En chargeant le compteur avec une valeur adéquate, on peut être interrompu lorsqu'un certain délai expire.

# Timers

## Avec MicroPython

```
import pyb
def clignote(t):
    pyb.LED(1).toggle()

timer = pyb.Timer(4, freq=1, callback=clignote)
```



# Timers

## Avec MicroPython

```
import pyb
def clignote(t):
    pyb.LED(1).toggle()

timer = pyb.Timer(4, freq=1, callback=clignote)
```

- ▶ La LED 1 de la carte (rouge) s'allumera et s'éteindra chaque seconde

# Timers

## Avec MicroPython

```
import pyb
def clignote(t):
    pyb.LED(1).toggle()

timer = pyb.Timer(4, freq=1, callback=clignote)
```

- ▶ La LED 1 de la carte (rouge) s'allumera et s'éteindra chaque seconde
- ▶ MicroPython se charge de calculer la valeur à charger dans le compteur pour obtenir une fréquence donnée



# Timers

## Avec MicroPython

```
import pyb
def clignote(t):
    pyb.LED(1).toggle()

timer = pyb.Timer(4, freq=1, callback=clignote)
```

- ▶ La LED 1 de la carte (rouge) s'allumera et s'éteindra chaque seconde
- ▶ MicroPython se charge de calculer la valeur à charger dans le compteur pour obtenir une fréquence donnée
- ▶ il se charge également d'activer la génération d'interruption pour le timer et d'enregistrer notre fonction pour traiter cette interruption.



# Timers

## Avec MicroPython

```
import pyb
def clignote(t):
    pyb.LED(1).toggle()

timer = pyb.Timer(4, freq=1, callback=clignote)
```

- ▶ La LED 1 de la carte (rouge) s'allumera et s'éteindra chaque seconde
- ▶ MicroPython se charge de calculer la valeur à charger dans le compteur pour obtenir une fréquence donnée
- ▶ il se charge également d'activer la génération d'interruption pour le timer et d'enregistrer notre fonction pour traiter cette interruption.

À faire en BE !

Suite...

Systeme d'exploitation

