

Formation Automatique et Informatique Industrielle

Master 1 S2

Matière : Systèmes Embarqués et Systèmes
Temps Réel SE-STR

Par : ATOUI Hamza

Plan de TD-TP

- Exemples avec solutions.
- Exercices.

Exemples & exercices

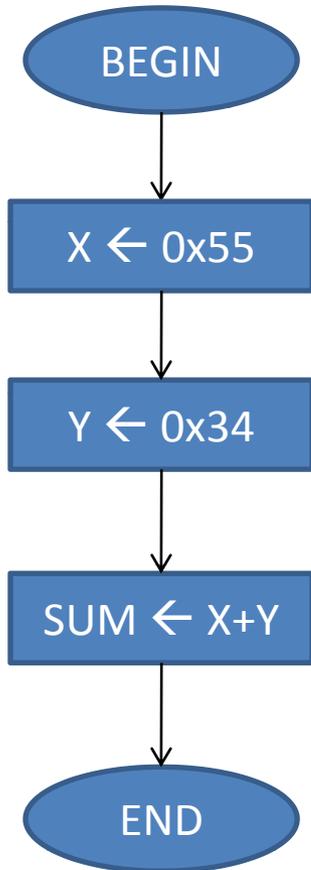
- Exemple 1:
 - Ecrire un programme qui fait le travail suivant:
 - Ranger la valeur 0x55 dans la case mémoire d'offset 0x33 par rapport à la BANK0.
 - Ranger la valeur 0x34 dans la case mémoire d'offset 0x40 par rapport à la BANK1.
 - Calculer la somme des deux cases mémoires initialisées précédemment puis ranger le résultat de la somme dans la case mémoire d'offset 0x50 par rapport à la BANK3.

Solution de l'exemple 1

- `; X ← 0x55 (X dans la BANK0)`
- `BCF STATUS,RP0 ;`
- `BCF STATUS,RP1 ; nous sommes dans la BANK0`
- `MOVLW 0x55`
- `MOVWF 0x33`

- `; Y ← 0x34 (Y dans la BANK1)`
- `BSF STATUS,RP0 ; nous sommes dans la BANK1`
- `MOVLW 0x34`
- `MOVWF 0x40`

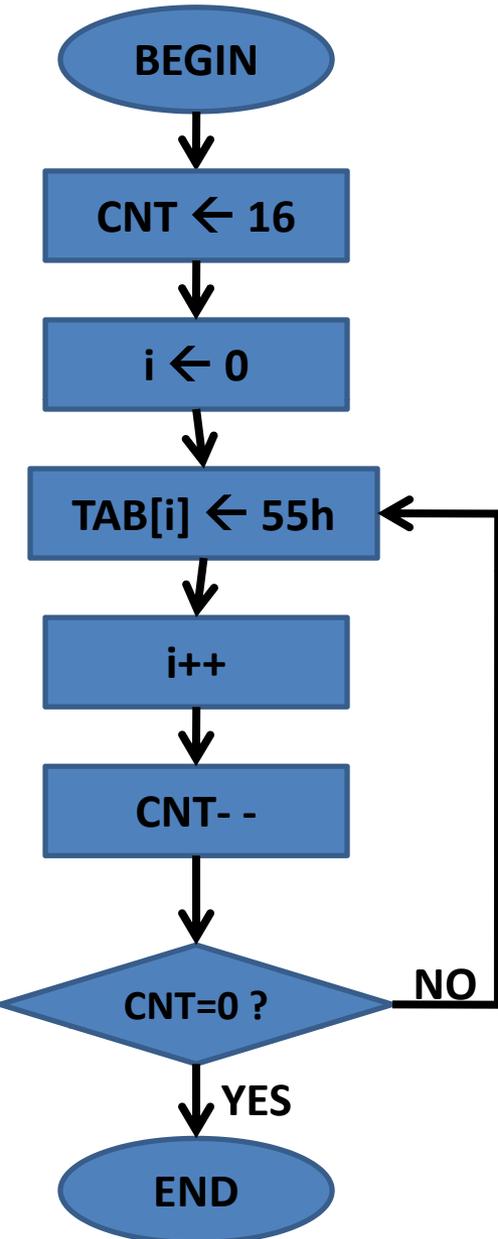
- `; SUM ← X+Y (NB: WREG contient Y donc`
- `; on passe vers la BANK0 pour sommer`
- `; avec X)`
- `BCF STATUS,RP0`
- `ADDWF 0x33,W`
- `; SUM dans la BANK3`
- `BSF STATUS,RP0`
- `BSF STATUS,RP1`
- `MOVWF 0x50`
- `GOTO $; END`



Exemples & exercices

- Exemple 2:
 - Ecrire un programme qui initialise les 16 octets débute à l'offset 0x30 dans la BANK3 par 0x55.

Solution de l'exemple 2



- `; CNT ← 16` (CNT existe dans la BANK3)
- `BSF STATUS,IRP ; SELECT B2-3 indirect mode`
- `BSF STATUS,RP0`
- `BSF STATUS,RP1 ; SELECT B3 direct mode`
- `MOVLW D'16'`
- `MOVWF CNT`
- `; i ← 0`
- `MOVLW 0xB0`
- `MOVWF FSR ; (IRP:FSR) pointe vers l'adresse TAB`
- `; TAB[i] ← 0x55`
- `MOVLW 0x55`
- `ENCORE : MOVWF INDF`
- `; i++`
- `INCF FSR,F ; pointe vers une autre case MEM`
- `; CNT-- avec le TEST`
- `DECFSZ CNT,F`
- `GOTO ENCORE`
- `GOTO $; END`

Exercices

- Exercice 1:
 - Ecrire un programme qui calcul la parité arithmétique de la case mémoire d'offset 0x50 par rapport à la BANK1, s'elle est paire, on met le caractère « P » dans la case mémoire d'offset 0x60 par rapport à la BANK3, sinon on met le caractère « I ».

Exercices

- Exercice 2:
 - Ecrire un programme qui calcul la somme MODULO_256 d'un tableau d'offset 0x20 par rapport à la BANK0 de taille 16 octets, et ranger le résultat dans la case mémoire d'offset 0x20 par rapport à la BANK3.

Exercices

- Exercice 3:
 - Ecrire un programme qui fait la division par 2 de toutes les cases mémoires de taille 32 octets placées dans la BANK0 à partir de l'offset 0x10.

Exercices

- Exercice 4:
 - Ecrire un programme qui fait la recherche de la première occurrence du caractère « H » dans une chaîne de caractères de longueur égal à 64 octets, dont l'adresse de la chaîne de caractères est 0x190
 - Si le cas ranger l'index de cette case mémoire dans la case mémoire d'offset 0x0C/BANK0.

Exercices

- Exercice 5:
 - Ecrire un programme qui fait le transfert d'une table d'offset 0x10 par rapport à la bank0 vers une autre table d'offset 0x20 par rapport à la bank3 dont le nombre d'octets à transférer est 16.
 - **(Problème de multi-pointeurs)**

Exercices

- Exercice 6:
 - Ecrire un programme qui réalise la multiplication entre 2 cases mémoires de 8 bits ($M1 * M2$) et ranger le résultat dans une autre case mémoire de 16 bits ($RM \leftrightarrow H:L$). (**selon l'algorithme suivant**)

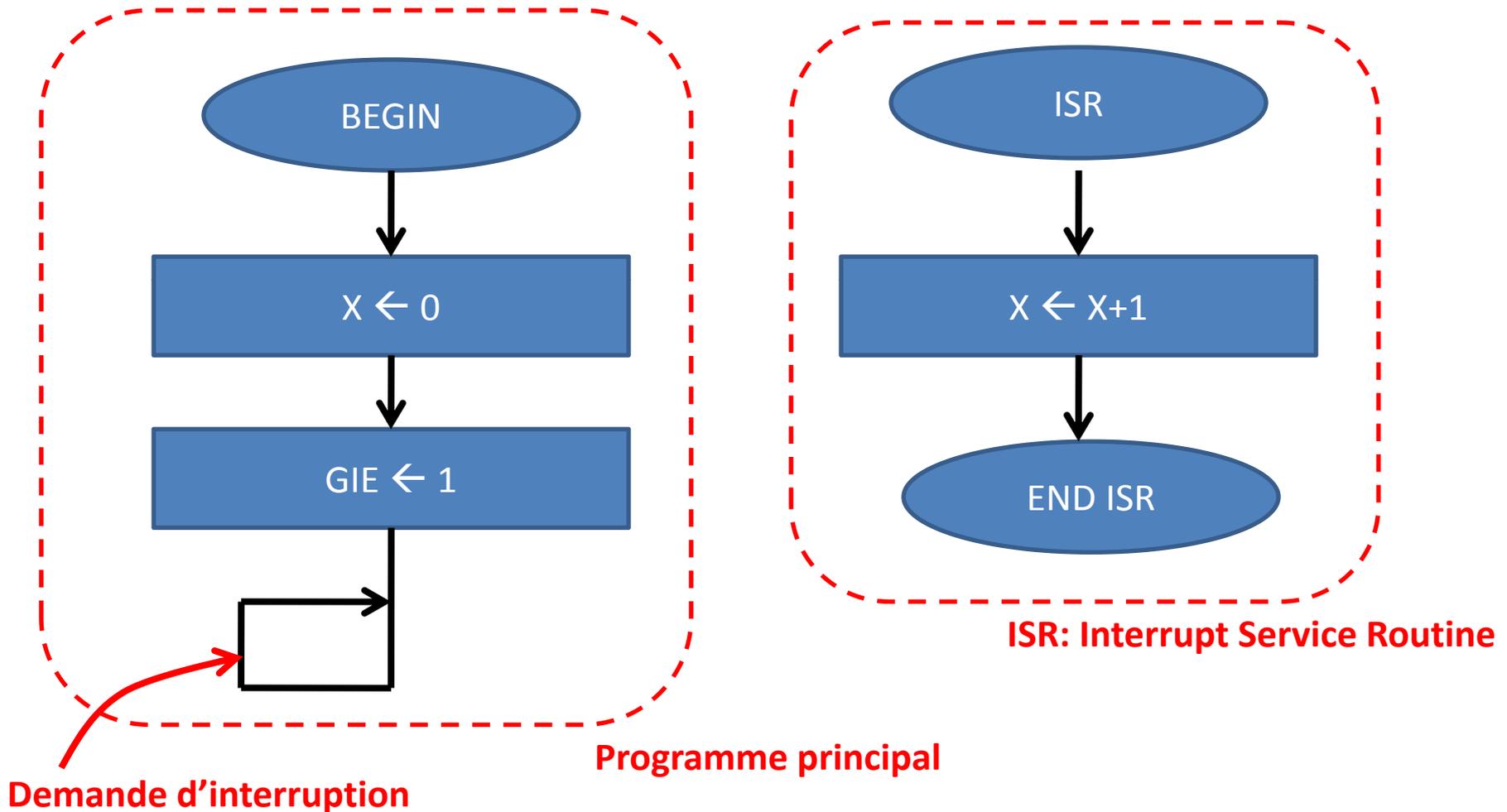
```
H ← 0;
L ← M1;
CNT ← 8;
While (CNT > 0)
    If (FIRST BIT OF L is 1) then
        H ← H+M2;
    End If
    Logical Shift Right (H:L);
    CNT ← CNT-1;
End While
```

Exercices

- L'exemple du cours sur les interruptions :
- Le code suivant ne fait rien (boucler infiniment dans la même adresse [**il plante**]), mais à chaque demande d'interruption le ISR incrémente la case mémoire d'offset 0x0C par rapport à la BANK0.
- L'organigramme suivant présente la solution de l'exemple :

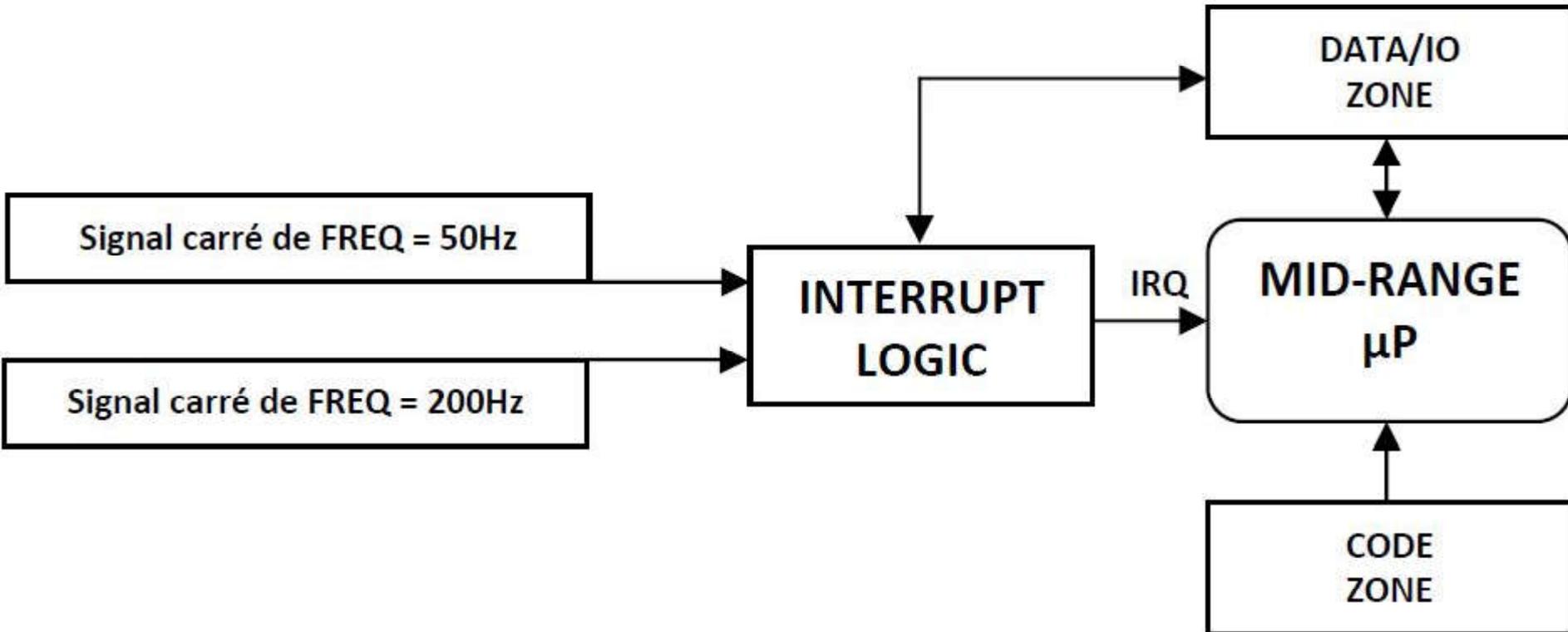
Exercices

- On demande de traduire en assembleur, puis simuler le



Exercices

- Exercice sur la gestion des interruptions:
- Soit le système suivant :



Exercices

- Ce système gère deux sources d'interruptions à travers la **logique d'interruption IRQ** avec la topologie de registre **INTCON** de **MID-RANGE** pour notre système est la suivante :
 - **Bit 0 (E50)** : présente l'activation de demande **d'interruption** de signal 50Hz (ENABLE BIT) [la mise à 1 de ce bit autorise les demandes d'interruption de signal 50Hz].
 - **Bit 1 (E200)** : présente l'activation de demande d'interruption de signal 200Hz (ENABLE BIT) [la mise à 1 de ce bit autorise les demandes d'interruption de signal 200Hz].
 - **Bit 2 (F50)** : présente le flag d'indication de présence d'une demande d'interruption par le signal 50Hz [si ce bit positionne à 1, donc il indique la présence d'une demande d'interruption].
 - **Bit 3 (F200)** : présente le flag d'indication de présence d'une demande d'interruption par le signal 200Hz [si ce bit positionne à 1, donc il indique la présence d'une demande d'interruption].
 - **Bits 4 jusqu'à 6** : non utilisées.
 - **Bit 7 (GIE)** : présente le bit d'autorisation des interruptions globales (GIE) [la mise à 1 de ce bit autorise les demandes d'interruption par la ligne IRQ, sinon sont masquées].

Exercices

- *La tâche d'interruption (ISR) gère les deux demandes d'interruption par :*
 - Si la source d'interruption est le signal 50Hz alors le traitement de cette interruption fait l'inversion de bit 1 de la case mémoire PORT/BANK0.
 - Si la source d'interruption est le signal 200Hz alors le traitement de cette interruption fait la décrémentation de la case mémoire CNT/BANK0 puis, si cette case mémoire tombe à zéro la case mémoire FLAG/BANK0 positionne à 1.
 - La priorité est donnée à la source de signal 50Hz.

Exercices

- *La tâche principale (MAIN) fait le travail suivant :*
 - Configure les interruptions (autorisation des demandes d'interruption de signal 50Hz et 200Hz).
 - Initialise les flags des interruptions (remise à zéro des deux flags).
 - Initialise les cases mémoires PORT/BANK0 et FLAG/BANK0 par zéro et la case mémoire CNT/BANK0 par 100 (100 en décimal et 0x64 en hexadécimal).
 - Autorise les demandes d'interruption globales (GIE ← 1).
 - Puis cette tâche se boucle infiniment sur le traitement suivant :
 - Si la case mémoire FLAG/BANK0 est à 1 alors :
 - CNT/BANK0 ← 100.
 - FLAG/BANK0 ← 0.
 - INFO_OUT/BANK0 ← INFO_IN/BANK3.

Exercices

- Questions:
 1. Réaliser les organigrammes des deux tâches (ISR et MAIN).
 2. Ecrire le code assembleur de chaque tâche (ISR et MAIN).
 3. Quelle est la période de la partie qui boucle infiniment de la tâche MAIN.

```
Save context:
    MOVWF TMP1
    SWAPF TMP1, F
    MOVF STATUS, W
    MOVWF TMP2
Load context:
    MOVF TMP2, W
    MOVWF STATUS
    SWAPF TMP1, W
```