

## Série d'exercices 5: Les Interruptions du 8085

### Rappel théorique préalable

Le 8085 possède 5 lignes d'interruption :

- **TRAP** (masquable uniquement par reset, priorité la plus haute)
- **RST 7.5** (déclenchée par front montant)
- **RST 6.5** (déclenchée par niveau haut)
- **RST 5.5** (déclenchée par niveau haut)
- **INTR** (priorité la plus faible, nécessite une instruction CALL)

### Exercice 1 : Priorités des interruptions

Si les interruptions suivantes arrivent simultanément : TRAP, RST 7.5, RST 5.5, INTR.  
Laquelle sera traitée en premier ? Justifiez.

#### Solution :

**TRAP** sera traitée en premier.

Justification : La priorité (de la plus haute à la plus basse) est :  
TRAP > RST 7.5 > RST 6.5 > RST 5.5 > INTR.

### Exercice 2 : Masquage des interruptions

Quelles instructions permettent de masquer/démasquer RST 7.5, RST 6.5, RST 5.5 ?  
Donnez un exemple de code qui masque RST 6.5 et RST 5.5 mais autorise RST 7.5.

#### Solution :

- **SIM** (Set Interrupt Mask) – permet de masquer RST 7.5, 6.5, 5.5.
- **RIM** (Read Interrupt Mask) – lit le masque actuel.

#### Code exemple :

assembly

**MVI A, 00001010B** ; *bit1=1 masque RST 5.5, bit2=1 masque RST 6.5, bit4=0 (RST 7.5 non masqué)*

**SIM** ; *applique le masque*

(Remarque : bits SID, SOD non utilisés ici)

### Exercice 3 : Gestion de l'interruption INTR

Lors d'une interruption INTR, le 8085 doit recevoir un code d'instruction (opcode) sur le bus de données.

Quelle est l'instruction typique envoyée par le contrôleur d'interruption ? Que se passe-t-il après ?

**Solution :**

L'instruction typique est **CALL** (opcode 0xCD) suivi de l'adresse de la routine de service.

Ou plus simplement, on envoie un **RST n** (n=0 à 7) : opcode 0xCF (RST 1) par exemple.

Après exécution :

1. Sauvegarde du PC dans la pile.
2. Saut à l'adresse  $8 * n$  (pour RST n).
3. À la fin de l'ISR, RET restaure le PC.

### Exercice 4 : Différence entre TRAP et RST 7.5

Expliquez deux différences fondamentales entre TRAP et RST 7.5.

**Solution :**

1. **Masquabilité** : TRAP est non masquable (sauf par reset), RST 7.5 est masquable via SIM.
2. **Déclenchement** : TRAP est sensible au niveau (et verrouillé), RST 7.5 est sensible au front montant.

### Exercice 5 : Écrire une routine de service pour RST 5.5

Écrivez une routine d'interruption pour RST 5.5 qui incrémente un compteur mémoire à l'adresse 2500H, puis réactive globalement les interruptions.

### Solution :

assembly

**ORG 002CH** ; Adresse vectorielle de RST 5.5 ( $8 \times 5.5$  ? Non :  $RST\ 5.5 = 8 \times 5 = 40$  décimal =  $0x28$ , mais le 8085 utilise  $0x2C$  pour RST 5.5 ? Attention : en réalité RST 5.5 -> vecteur  $0x2C$ , RST 6.5 ->  $0x34$ , RST 7.5 ->  $0x3C$ )  
; Correction : vecteurs réels : RST 5.5 =  $0x2C$ , RST 6.5 =  $0x34$ , RST 7.5 =  $0x3C$

**ISR\_RST55:**

**PUSH PSW** ; sauve A et flags

**LDA 2500H**

**INR A**

**STA 2500H**

**EI** ; réactive interruptions globales (après sauvegarde)

**POP PSW**

**RET**

Remarque : L'adresse exacte d'exécution après RST 5.5 est **0x002C**.

## Exercice 6 : Programme complet avec interruption RST 7.5

On veut qu'à chaque appui sur un bouton générant RST 7.5, le 8085 lise un port d'entrée (adresse 0F0H) et l'affiche sur un port de sortie (adresse 0F1H). Écrivez le programme principal et l'ISR.

### Solution :

assembly

; Programme principal

**MAIN:**

**MVI A, 00001000B** ; bit3=1 pour réinitialiser le flip-flop RST 7.5

**SIM** ; permet de réarmer RST 7.5 après traitement

**EI** ; active globalement les interruptions

**HERE:**

**JMP HERE** ; boucle infinie

; ISR pour RST 7.5 (vecteur  $0x3C$ )

**ORG 003CH**

**ISR\_RST75:**

**PUSH PSW**

**IN 0F0H** ; lecture port entrée

**OUT 0F1H** ; écriture port sortie

Université Djilali BOUNAAMA, Khemis Miliana

مليانة خميس بونعامه جيلالي جامعة

Faculté des Sciences et de la Technologie

والتكنولوجيا العلوم كلية



**MVI A, 00001000B** ; réinitialise RST 7.5 (obligatoire pour la prochaine)

**SIM**

**POP PSW**

**EI**

**RET**

### Tableau récapitulatif des vecteurs

Interruption	Adresse vecteur
TRAP	0x0024
RST 5.5	0x002C
RST 6.5	0x0034
RST 7.5	0x003C
INTR + RST n	8 * n