

# PHY4501 — MANIPULATION DE MOTS

## 1. ADDITION

Un mot est une donnée codée sur 2 octets (16 bits). Le programme *W\_Add* doit réaliser l'addition de deux mots *W1* et *W2*. Chaque mot occupe deux adresses de la mémoire de données : *W1\_L* = 7F, *W1\_H* = 7E, *W2\_L* = 7D et *W2\_H* = 7C. Le résultat *RES* est stocké dans les registres R0 (octet de poids faible, *RES\_L*) et R1 (octet de poids fort, *RES\_H*).

**Q1.** Calculer *W1* + *W2* lorsque (valeurs exprimées en hexadécimal) :

1. *W1* = 1F25 et *W2* = 4212 ;
2. *W1* = 28F2 et *W2* = 3748 ;
3. *W1* = FFFF et *W2* = 0001.

**Q2.** Réaliser l'étude complète du programme :

1. Faire le tableau de description des données.
2. Écrire, en pseudo-codes, l'algorithme du programme.
3. Sous Ride, écrire le programme en langage assembleur AT89C51.
4. En mode débogage, vérifier le bon fonctionnement du programme.

## 2. SOUSTRACTION

Le programme *W\_Sub* doit réaliser la soustraction de deux mots : *W1* - *W2*.

**Q1.** Calculer *W1* - *W2* lorsque (valeurs exprimées en hexadécimal) :

1. *W1* = 0503 et *W2* = 0201 ;
2. *W1* = 0403 et *W2* = 020F ;
3. *W1* = 0203 et *W2* = 020F.

**Q2.** Réaliser l'étude complète du programme.

## 3. COMPARAISON

Le programme *W\_Comp* doit réaliser la comparaison de deux mots (*W1* et *W2*). Le résultat *RES* (bit P3.7) sera égal à 1 lorsque *W1* est strictement inférieur à *W2*.

**Q1.** Donner la valeur théorique de *RES* lorsque :

- |   |   |
|---|---|
| 1. <i>W1</i> = 0000 et <i>W2</i> = 0000 ; | 4. <i>W1</i> = 0203 et <i>W2</i> = 0302 ; |
| 2. <i>W1</i> = 0000 et <i>W2</i> = 0102 ; | 5. <i>W1</i> = 0203 et <i>W2</i> = 0102.  |
| 3. <i>W1</i> = 0200 et <i>W2</i> = 0102 ; |   |

**Q2.** Réaliser l'étude complète du programme.

## 4. COMPARAISON PAR ROUTINE

Le programme `W_Comp_Rt` est une variante du programme `W_Comp` dans laquelle la comparaison s'effectue à l'aide d'une routine (sous-programme). Les paramètres passés de cette routine sont les adresses hautes des deux mots, `W1_H` et `W2_H`, qui seront passés respectivement à l'aide des registres R0 et R1. Le résultat de la comparaison (`Res_Rt`) sera retourné par la routine à l'aide du bit F0 du PSW.

- Q1.** Dans l'appel à une routine, donner le rôle de la pile.
- Q2.** Donner le nom du registre associé à la pile. Donner sa valeur au *reset*.
- Q3.** On propose d'initialiser la valeur de ce registre à 30h. Justifier le choix de cette valeur et des adresses utilisées pour les variables enregistrées en mémoire programme.
- Q4.** Réaliser l'étude complète du programme.
- Q5.** Le passage des paramètres se fait à l'aide des registres R0 et R1, proposer des solutions dans le cas où ces registres sont déjà utilisés par le programme appelant (sans changer les registres de la routine).

## 5. COMPARAISON AVEC ENTRÉES/SORTIES

Le programme `W_Comp_In` est une variante du programme `W_Comp_Rt` dans laquelle les mots `W1` et `W2` sont entrées à l'aide d'interrupteurs (`W_IN`) et visualisés à l'aide d'afficheurs sept segments. Afin de limiter le câblage, seuls les deux bits de poids faible de chaque adresse de chaque mot sont modifiables, comme illustré en figure 1. Deux DEL permettent de visualiser si  $W1 < W2$  (*INF*) ou si  $W1 \geq W2$  (*SUP*).

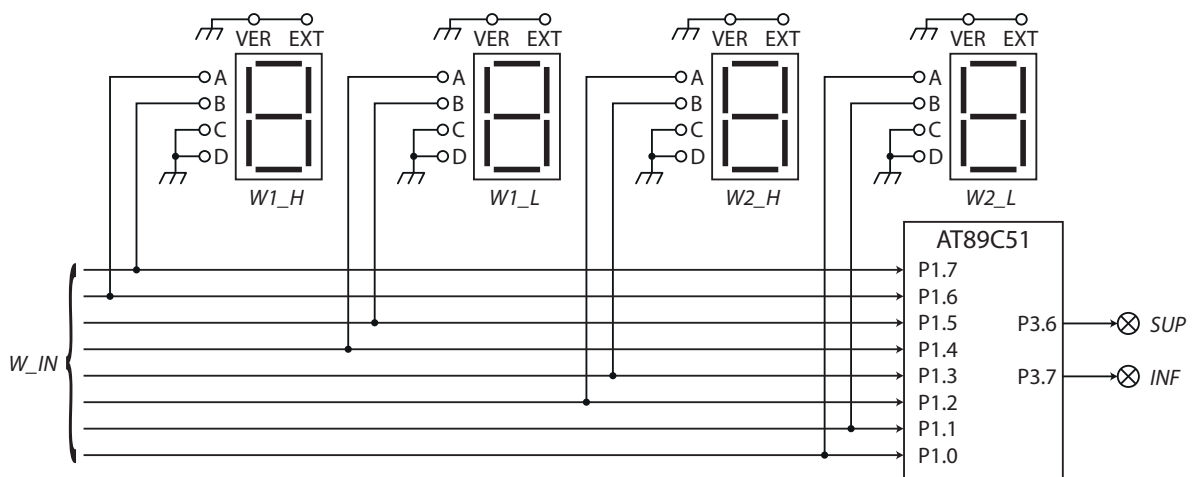


FIGURE 1 – Schéma de câblage du comparateur

- Q1.** Réaliser l'étude complète du programme.
- Q2.** Effectuer le câblage selon la figure 1, programmer le microcontrôleur et tester le programme.

## 6. COMPARAISON PAR INTERRUPTION

Le programme W\_Comp\_IT est une variante du programme W\_Comp\_In dans laquelle la comparaison s'effectue lors d'une interruption externe. Celle-ci sera provoquée par un front descendant de  $Bp$  ( $\overline{INT0}$ ), comme illustré en figure 2.

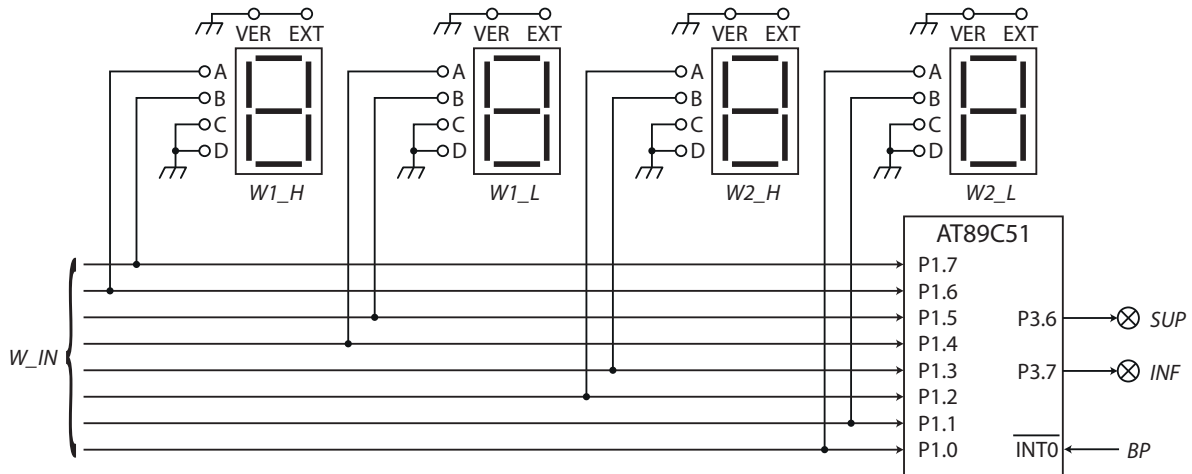


FIGURE 2 – Schéma de câblage du comparateur par interruption

- Q1.** Donner la valeur des registres IE et TCON pour obtenir la configuration demandée.
- Q2.** Donner l'adresse du vecteur d'interruption de  $\overline{\text{INT0}}$ . Donner le nombre d'octets séparant cette adresse à l'adresse du vecteur d'interruption suivant. Quelle conclusion en tirer ?
- Q3.** Réaliser l'étude complète du programme.
- Q4.** Effectuer le câblage selon la figure 1, programmer le microcontrôleur et tester le programme.

## 7. COMPARAISON AVEC PAUSE

Le programme `W_Comp_Pause` est une variante du programme `W_Comp_IT`. Ce dernier continue d'afficher *INF* ou *SUP* même si *W\_IN* change de valeur (et donc *W1* et *W2*). On souhaite donc que le résultat reste affiché uniquement quelques secondes. La temporisation peut être réalisée à l'aide de boucles de décomptage comme illustré en figure 3.

```

MOV      R0 , #27h
Rpt_t0:  MOV      R1 , #0FFh
Rpt_t1:  MOV      R2 , #0FFh
Rpt_t2:  DJNZ     R2 , Rpt_t2
         DJNZ     R1 , Rpt_t1
         DJNZ     R0 , Rpt_t0

```

FIGURE 3 – Réalisation d'une temporisation

- Q1.** À l'aide de la durée de l'instruction DJNZ, calculer la durée approximative de la temporisation proposée en figure 3.

- Q2.** Écrire le programme `W_Comp_Pause` dans lequel la temporisation proposée en figure 3 est utilisée sous la forme d'une routine nommée `Pause`.
- Q3.** Programmer le microcontrôleur et tester le programme.
- Q4.** Proposer une solution afin qu'aucun registre ne soit modifié par la routine `Pause` (entre son entrée et sa sortie, en utilisant toujours les registres `R0` à `R2`).