

TP N°4

Ecriture et simulation des programmes en assembleur.

I. Objectifs du TP

- Maitriser les instructions de branchement et boucles.
- Ecrire des programmes en assembleur utilisant ces instructions.

II. Rappels

Les instructions de branchement (ou saut) permettent de modifier l'ordre d'exécution des instructions du programme en fonction de certaines conditions.

Il existe 3 types de saut :

- Appels de sous-programmes.
- Sauts inconditionnel : JMP
- Sauts conditionnels :

JC : Saut si CY=1

JNC : Saut si CY=0

JZ : Saut le résultat est nul (Z=1)

JNZ : Saut le résultat est différent de zéro (Z=0)

JP : Saut si S=0

JM : Saut si S=1

JPE : Saut si P=1

JPO : Saut si P=0

Ces instructions sont souvent utilisés avec les opérations arithmétiques et logiques avec en plus les instructions de comparaison avec l'accumulateur CMP R/M et CMI donnée de 8bits. La comparaison est une soustraction qui ne modifie pas le contenu de l'accumulateur ou registre/case mémoire, contrairement aux autres opérations arithmétiques et logique.

Tableau

Un **tableau** est un ensemble de valeurs situés dans une zone mémoire. Sa déclaration dans le simulateur se fait de la manière suivante **après** l'instruction **HLT**:

Exemple d'un tableau de trois cases 56H, 97H, 3AH et qui commence à l'adresse 0050H.

```
# ORG 0050
```

```
# DB 56H, 97H, 3AH
```

III. Travail demandé

Exercice N°1

1. Effectuer la multiplication des deux opérandes **05H** et **06H** situés aux adresses mémoire **0050H** et **0051H** respectivement en utilisant la notion de boucles : $A \times B = A + A + \dots + A$ B fois.

Le résultat sera stocké dans la case **0052H**.

Exercice N°2

2. Effectuer la division des deux opérandes **14H=20₁₀** et **06H=6₁₀** ($20/6$) situés aux adresses mémoire **0053H** et **0054H** respectivement en utilisant la notion de boucles. Le quotient sera stocker dans la case mémoire **0055H** et le reste dans la case **0056H**.

Exercice N°3

Ecrire un programme en assembleur qui permet de compter le nombre de valeurs négatives appartenant à un tableau de **10** octets commençant à l'adresse **0200H**, puis stocker le résultat dans la case mémoire d'adresse **020AH**.

Exercice N°4

Ecrire un programme en assembleur qui permet d'écrire les nombres de **30** à **56** en mémoire à partir de l'adresse **0100H**.

Exercice N°5

Ecrire un programme en assembleur qui permet de compter le nombre de **0** dans un octet. Le résultat sera stocké dans la case mémoire d'adresse **0100H**.

Exercice N°6

Permuter les **10** éléments d'un tableau placé en mémoire à l'adresse **0100H**.

Exemple :

