Université Tunis El-Manar	**Année Universitaire** : 2018-2019
Faculté des Sciences de Tunis	**Section :** IF3
Module : Programmation Machine	
Enseignant : C.A. ABID	

Fiche TD/TP 2

A) Opérations de transfert

On rappelle les instructions suivantes :
MOV reg1,reg2 : reg1← reg2
MOV reg, m : reg ← Mémoire(HL)
MOV M, reg : Mémoire(HL) ← reg
MVI reg, value : reg← value
MVI M, value : Mémoire(HL) ← value
LXI reg_pair, data 16 bits
LDA adr : A← Mémoire(adr)
STA adr : Mémoire(adr) ← A
LHLD adr : L← Mémoire(adr) et H← Mémoire(adr+1)
SHLD adr : Mémoire(adr) ← L et Mémoire(adr+1)← H
XCHG : Permuter les contenus des registres DE et HL

ADD reg : A ← A + reg.
ADD M : A ← A + Mémoire(HL).
ADC reg : : A ← A + reg + CS
ADC M : : A ← A + Mémoire(HL) + CS
INR reg : reg ← reg + 1
DCR reg : reg ← reg - 1

Exercice 1.

Écrire un programme assembleur 8085 permettant d'additionner deux nombres codés sur 16 bits.

B) Les tests

En assembleur, pour réaliser des tests, on utilise les bits du registre FLAGS comme condition de test et une instruction de branchement conditionnelle (saut si certains bits du registre FLAGS valent 0 ou 1) pour déclencher la partie alors ou la partie sinon du test.
Les bits du registre FLAGS sont positionnés par les instructions que nous avons déjà vues (instructions arithmétiques, logiques, ...). Ils sont également positionnés par des instructions spécialement conçues pour réaliser des tests et qui n'ont d'autres effets que de positionner les bits de FLAGS en fonction de certaines conditions sur leurs opérandes telle que l'instruction CMP,
Une fois les indicateurs positionnés, une instruction dite de saut conditionnel teste un bit ou une combinaison de bits de FLAGS et, en fonction du résultat :
– effectue une rupture de séquence (un saut) vers un endroit précis dans le code où l'exécution se poursuit
normalement.
– continue en séquence si le test ne donne pas un résultat positif.

Les bits de FLAGS du processeur 8085

Le processeur 8085 dispose des bits de FLAGS suivants :
1. Carry (CS) : bit de retenu
2. ZERO (Z) : indique si le résultat de la dernière opération est nul ou non,
3. SIGN (S) : indique si le résultat de la dernière opération est négatif ou non,
4. PARITY (P) : indique si le résultat de la dernière opération est pair ou non,
5. AUXILIARY CARRY (AC) : bit de retenu auxiliaire pour les bits ayant la position 3.

L'instruction CMP
C’est l’instruction la plus utilisée pour positionner les FLAGS avant d’effectuer une instruction de saut conditionnel.
Syntaxe: CMP registre : comparer l’accumulateur avec le registre
CMP M : comparer l’accumulateur avec le mot mémoire adressé par HL,

CMP permet de comparer deux valeurs. Pour cela CMP soustrait le second opérande du premier, sans cependant modifier l’opérande destination, mais en positionnant les indicateurs en fonction du résultat. Ainsi, si le résultat de la soustraction est nul, donc l’indicateur Z a été positionné à 1, cela signifie que les deux valeurs comparées sont égales. En utilisant des raisonnements du même genre, on peut savoir si les deux valeurs sont différentes, ordonnées strictement ou non.

Les instructions de sauts conditionnels

1. JZ addr (label). (Saut si le résultat est nul)
2. JNZ addr (label) (Saut si le résultat est non nul)
3. JC addr (label). (Saut s’il y a un bit de retenu)
4. JNC addr (label). (Saut s’il n’y a pas un bit de retenu)
5. JP addr (label). (Saut si le résultat est positif)
6. JM addr (label). (Saut si le résultat est négatif)
7. JPE addr (label) (Saut si le résultat est pair)
8. JPO addr (label) (Saut si le résultat est impair)

Exercice 2.

Traduire les algorithmes suivants en assembleur 8085 :
a)
```
x:=0
tant que x<10 faire
        x:=x+1
fait
```

b)
```
x:=0
répéter
        x:=x+1
jusqu'à x=10
```
c)
```
j:=0
pour x de 1 à 10 faire
        j:=j+x
fpour
```

Exercice 3.

Écrire un programme assembleur 8085 qui sauvegarde le résultat de la somme $\sum_{i=1}^{n} i$ où est une valeur entière strictement positive spécifiée à travers une variable.