

TP N°2

Opérations arithmétiques et logiques.

I. Objectifs du TP

- Maitriser les instructions logiques et arithmétiques.
- Maitriser les instructions de rotation.
- Ecrire des programmes en assembleur utilisant ces instructions.

II. Rappels

1. Registre d'état

On mentionne ici les indicateurs (flags) affectés par une instruction arithmétique ou logique, on utilise les notations suivantes :

| D ₇ | D ₆ | D ₅ | D ₄ | D ₃ | D ₂ | D ₁ | D ₀ |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| S | Z | | AC | | P | | CY |

Les flags sont des drapeaux ou des indicateurs d'état (registre d'état). Ils indiquent l'état d'une opération effectuée par l'unité arithmétique et logique. Ce registre contient 5 bits S, Z, AC, P et CY.

S: Sign-flag (Bit de signe). C'est le bit D₇ du résultat de l'opération.

Z: Zero-flag. Si le résultat de l'opération égale à 0 ce bit est mis à 1. Autrement il est mis à 0.

AC : Auxilliary Carry-flag. Dans une opération arithmétique, quand une retenue est générée par le bit D₃ à D₄ le AC est mis à 1 sinon il est mis à 0. Il est utilisé dans les codes BCD et il n'est pas utilisé par le programmeur.

P : Parity-flag. Après une opération arithmétique ou logique, si le résultat contient un nombre pair de 1, le drapeau P est affecté par 1. Si le résultat contient un nombre impair de 1, le drapeau P est affecté par 0.

CY : Carry-flag. Dans une opération arithmétique, quand une retenue est générée par les huit bits le CY est mis à 1 sinon il est mis à 0. Ce bit est accessible par le programmeur.

2. Opérations arithmétiques

Les instructions arithmétiques de base sont l'addition, la soustraction, l'incrémentement et la décrémentement. Pour ADD, ADI, SUB et INR, le résultat de l'opération sera stocker dans l'accumulateur A.

| | |
|-------------------------------------|---|
| ADD : Addition | Additionner le contenu de l'accumulateur A avec le contenu du registre ou case mémoire. |
| ADI : Addition Immédiate | Additionner le contenu de l'accumulateur A avec une valeur |
| SUB : Soustraction | Soustraire le contenu du registre du contenu de l'accumulateur A ou case mémoire. |
| SUI : Soustraction Immédiate | Soustraire une valeur du contenu de l'accumulateur A |
| INR : Incrémentation | Incrémente le contenu d'un registre ou case mémoire par 1 (+ 1) |
| DCR : Décrémentation | Décrémente le contenu d'un registre ou case mémoire par 1 (- 1) |
| INX : Incrémentation 16bits | Incrémenter un registre pair ou SP (16bits) |
| DCX : Décrémentation 16bits | Décrémenter un registre pair ou SP (16bits) |

3. Opérations logiques

Les opérateurs logiques de base sont AND, OR, XOR et NOT. Le résultat de l'opération sera stocker dans l'accumulateur A.

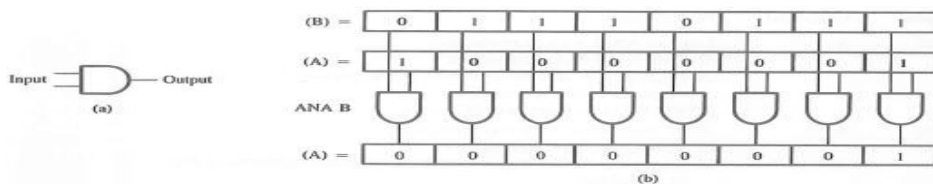
| | |
|-----------------------------|---|
| ANA : And | And logique entre l'accumulateur A et le contenu du registre. |
| ANI : And Immédiate | And logique entre le contenu de l'accumulateur A et une valeur 8bits. |
| ORA : Or | Or logique entre l'accumulateur A et le contenu du registre. |
| ORI : Or Immédiate | Or logique entre le contenu de l'accumulateur A et une valeur 8bits. |
| XRA : Xor | Ou exclusif entre l'accumulateur A et le contenu du registre. |
| XRI : Xor Immédiate | Ou exclusif entre le contenu de l'accumulateur A et une valeur 8bits. |
| CMA : Complémenter A | Complémenter de l'accumulateur A (inverser tous les bits de A). |

Exemple : Instruction ANI

MVI B,77H

MVI A, 81H

ANI B



4. Opérations de rotation

Les opérations de rotation sont RLC, RAL, RRC et RAR. Le résultat de l'opération sera stocker dans l'accumulateur A.

RLC : Rotation à gauche de A.

RAL : Rotation à gauche de A par CY.

RRC : Rotation à droite de A.

RAR : Rotation à droite de A par CY.

III. Travail demandé

1. Première Partie

1- Spécifier le contenu des registres A et B et le registre d'état en exécutant les instructions suivantes (Step By Step). Aussi, donner le contenu du port de sortie 00H.

| | | | | | |
|-------------------|-----|-----|---|---|----|
| Contenu initial : | A | B | S | Z | CY |
| | 00H | FFH | 0 | 0 | 0 |

MVI A, F2H

MVI B, 7AH

ADD B

OUT 00H

HLT

2- Quelle est l'opération qui peut être effectuée en exécutant : ADD A

3- Quelle est l'opération qui peut être effectuée en exécutant : SUB A. Donner l'état de Z et CY dans ce cas.

4- Spécifier le contenu des registres A et C et le registre d'état en exécutant les instructions suivantes (Step By Step).

| | | | | | |
|-------------------|----|----|---|---|----|
| Contenu initial : | A | C | S | Z | CY |
| | XX | XX | 0 | 0 | 0 |

MVI A, 5EH

ADI A2H

MOV C, A

HLT

2. Deuxième Partie

1- Quelle est l'opération qui peut être effectuée par l'exécution de l'instruction XRA A. Donner l'état de Z et CY.

2- Trouver le contenu des registres et l'état des flags S, Z et CY après l'exécution de l'instruction ORA A

MVI A,A9H

MVI B,57H

ADD B

ORA A

3- Trouver le contenu des registres et l'état des flags S, Z et CY pour le programme suivant.

| | | | | | |
|-------------------|----|----|---|---|----|
| Contenu initial : | A | B | S | Z | CY |
| | XX | XX | X | X | X |

XRA A

MVI B,4AH

SUI 4FH

ANA B

HLT

3. Troisième Partie

1- L'accumulateur pour l'ensemble des instructions suivantes contient un nombre BCD.

Expliquer le rôle de ces instructions

MVI A,79H

ANI F0H

RRC

RRC

RRC

RRC

2- Ecrire l'équation mathématique réalisé par le programme suivant :

MVI A,07H

RLC

MOV B,A

RLC

RLC

ADD B

4. Quatrième Partie

Ecrire un programme assembleur qui permet de réaliser la fonction mathématique suivante.

Sachant que X est un nombre inférieur à 7 et est stocké dans la case mémoire 0100H et Y sera stocké dans la case 0101H.

$$Y=(16(2X+1)+X-1)/8$$