

## Jeux d'ins du microprocesseur 8085

### II.5 Jeu d'instructions du 8085 [16]

Les instructions du 8085 sont divisées en trois groupes :

- Instructions de transfert
- Instruction arithmétiques et logiques
- Instructions de branchement

#### II.5.1 Instructions de transfert

Ce premier groupe d'instructions permet de transférer (copier) des données d'un endroit un autre (d'un registre ou mémoire ou interface d'E/S) appelé source vers un autre registre (ou E/S ou mémoire) appelée destination. En effet, le contenu de la source n'est pas déplacé, mais copié dans la destination sans modifier le contenu de la source. Les instructions de transfert sont résumées comme suit:

**1. MOV :** Move : Copier un octet de données depuis un registre ou adresse mémoire

**Syntaxe :** MOV Rd,Rs ou MOV M,Rs ou MOV Rd,M

Tel que Rd : Registre destination Rs: Registre source et M adresse spécifiée par HL

**Exemple :**

MOV C,D ou MOV M,B

**2. MVI :** Move Immédiate : Charger un octet de données

**Syntaxe :** MVI Rd, donnée 8 bits

Tel que Rd : Registre destination

**Exemple :**

MVI A,14H

**3. OUT :** Output to port : Envoyer un octet de données depuis A vers une interface de sortie

**Syntaxe :** OUT adresse du port de sortie

**Exemple :**

OUT 08H

**4. IN :** Input from Port : Recevoir un octet de données d'une interface d'entrée dans A

**Syntaxe :** IN adresse du port d'entrée

**Exemple :**

IN 09H

**5. LXI :** Load register pair Immediate : Charger un mot (16bits) de données

**Syntaxe :** LXI Rp, donnée 16 bits

Tel que Rp : Registre pair

**Exemple :**

LXI B,1400H ; (B)=14H et (C)=00H

**6. LDA :** Load Accumulator : Charger l'accumulateur par le contenu d'une adresse mémoire

**Syntaxe :** LDA Adresse 16 bits

**Exemple :**

LDA 2000H

**7. LDAX :** Load Accumulator : Charger l'accumulateur par le contenu d'une adresse mémoire spécifié dans un registre B ou D

**Syntaxe :** LDAX Rp

Tel que Rp : Registre pair B ou D

**Exemple :**

LDAX B

**8. STA :** Store Accumulator : Stocker l'accumulateur dans une case mémoire

**Syntaxe :** STA Adresse 16 bits

**Exemple :**

STA 3000H

**9. STAX :** Store Accumulator : Stocker l'accumulateur dans une case mémoire dont l'adresse est spécifié dans un registre B ou D

**Syntaxe :** STAX Rp

Tel que Rp : Registre pair B ou D

**Exemple :**

STAX D

**10. LHLD :** Load HL registers direct : Charger le registre pair H par le contenu de la case mémoire.

**Syntaxe :** LHLD Adresse 16 bits

**Exemple :**

LHLD 2000H

**11. SHLD :** Store HL registers direct : Charger le contenu de la case mémoire par le contenu du registre pair H.

**Syntaxe :** SHLD Adresse 16 bits

**Exemple :**

SHLD 3000H

**12. XCHG** : Exchange the contents of HL with DE : Echange les contenus de HL et DE.

**Syntaxe** : XCHG

**13. XTHL** : Exchange the top of the stack with HL : Echange le top de la pile avec HL.

**Syntaxe** : XTHL

**14. SPHL** : Copy HL registers into the stack pointer : Copie les contenus des registres HL dans le registre SP.

**Syntaxe** : SPHL

**15. PCHL** : Copy HL registers into the program counter : Copie les contenus des registres HL dans le registre CP.

**Syntaxe** : PCHL

## **VII.2. Instructions arithmétiques**

Le microprocesseur 8085 effectue plusieurs opérations arithmétiques à savoir : l'addition (ADD ou ADI), la soustraction (SUB ou SUI), l'incréméntation 8 bits (INR), l'incréméntation 16 bits (INX), la décrémentation 8 bits (DCR) et la décrémentation 16 bits (DCX).

Pour les instructions ADD, ADI, SUB et SUI, le microprocesseur assume que l'accumulateur A est par défaut l'un des deux opérandes et le résultat sera stocker dans l'accumulateur A. Les flags seront affectés.

Les instructions d'incréméntation et de décrémentation affectent le contenu du registre spécifié et affectent tout les flags à l'exception du CY.

**1. ADD** : Addition : Additionner le contenu de l'accumulateur A avec le contenu du registre ou case mémoire.

**Syntaxe** : ADD R ou ADD M

Tel que R : Registre 8 bits et M : adresse d'une case mémoire

**Exemple** :

MVI A,04H

MVI B,03H

ADD B

Ce qui donne  $(A)=(A)+(B)=04H+03H=07H$  S=0 Z=0 CY=0

**2. ADI** : Addition Immédiate : Additionner le contenu de l'accumulateur A avec une valeur

**Syntaxe** : ADI donnée 8 bits

Tel que R : Registre 8 bits

**Exemple** :

MVI A,FFH

ADI 01H

Ce qui donne  $(A)=(A)+01H=FFH+01H=00H$  S=0 Z=1 CY=1

**3. SUB :** Soustraction : Soustraire le contenu du registre du contenu de l'accumulateur A ou case mémoire.

**Syntaxe :** SUB R ou SUB M

Tel que R : Registre 8 bits et M : adresse d'une case mémoire

**Exemple :**

MVI A,FFH

MVI B,03H

SUB B

Ce qui donne  $(A)=(A)-(B)=FFH-03H=FCH$  S=1 Z=0 CY=0

**4. SUI :** Soustraction Immédiate : Soustraire une valeur du contenu de l'accumulateur A

**Syntaxe :** SUI donnée 8 bits

Tel que R : Registre 8 bits

**Exemple :**

MVI A, 05H

SUI 02H

Ce qui donne  $(A)=(A)-02H=05H-02H=03H$  S=0 Z=0 CY=0

**5. INR :** Incrémentation : Incrémenter le contenu d'un registre ou case mémoire par 1 (+ 1)

**Syntaxe :** INR R ou INR M

Tel que R : Registre 8 bits et M : adresse d'une case mémoire

**Exemple :**

MVI A,05H

INR A

Ce qui donne  $(A)=(A)+1=05H-1=06H$  S=0 Z=0

**6. DCR :** Décrémentation : Décrémenter le contenu d'un registre ou case mémoire par 1 (- 1)

**Syntaxe :** DCR R ou DCR M

Tel que R : Registre 8 bits et M : adresse d'une case mémoire

**Exemple :**

MVI A,01H

DCR A

Ce qui donne  $(A)=(A)-1=01H-1=00H$  S=0 Z=1

**7. INX :** Incrémentation 16bits : Incrémenter un registre pair ou SP (16bits)

**Syntaxe :** INX Rp

Tel que Rp : Registre pair (B, D ou H) ou SP

**Exemple :**

LXI SP,2000H

INX SP

Ce qui donne  $(SP)=(SP)+1=2000H+1=2001H$  S=0 Z=1

**8. DCX :** Décrémentation 16bits : Décrémenter un registre pair ou SP

**Syntaxe :** DCX Rp

Tel que Rp : Registre pair (B, D ou H) ou SP

**Exemple :**

LXI B,3000H

DCX B

Ce qui donne  $(B)=(B)-1=3000H-1=2FFFH$  S=0 Z=0

**9. ADC :** Add register contents with carry : Additionner registre 8 bits ou case mémoire avec A et CY

**Syntaxe :** ADC R ou ADC M

Tel que R : Registre 8 bits et M: adresse d'une case mémoire

**Exemple :**

MVI A,30H

MVI B,F0H

ADD B ;  $(A)=30H+F0H=20H$  et CY=1

ADC B ;  $(A)=20H+F0H+1=11H$

**10. ACI :** Add immediate 8 bits data with carry : Additionner une donnée 8 bits avec A et CY

**Syntaxe :** ACI donnée 8 bits

**Exemple :**

MVI A,30H

MVI B,F0H

ADD B ;  $(A)=30H+F0H=20H$  et CY=1

ACI 30H ;  $(A)=20H+30H+1=51H$

**11. SBB :** Subtract register contents with borrow : Soustraire registre 8 bits ou case mémoire et report CY de A.

**Syntaxe :** SBB R ou SBB M

Tel que R : Registre 8 bits et M : adresse d'une case mémoire

**Exemple :**

MVI A,30H

MVI B,20H

ADI FFH ; (A)=30H+FFH=2FH et CY=1

SBB B ; (A)=2FH-20-1=0EH

**12. SBI : Subtract** immediate 8 bits data with borrow : Soustraire donnée 8 bits et report CY de A.

**Syntaxe :** SBI donnée 8 bits

**Exemple :**

MVI A,30H

ADI FFH ; (A)=30H+FFH=2FH et CY=1

SBI 20H ; (A)=2FH-20-1=0EH

**13. DAA : Decimal Adjust Accumulator :** Ajustement décimal de l'accumulateur

**Syntaxe :** DAA

**Exemple :**

MVI A,38H

MVI B,43H

ADD B ; A=7BH

DAA ; A= 81H

### VII.3. Instructions logiques

Le microprocesseur est un circuit logique programmable. L'ensemble des opérateurs logiques est composé de : AND, OR, XOR et NOT. Le résultat de l'opération sera stocker dans l'accumulateur A. Ces opérateurs engendrent une mise à zéro du flag CY. Ils modifient l'état de S, Z et P suivant le résultat à l'exception de CMA. L'opérateur AND sert au masquage à 0 et l'opérateur OR sert au masquage à 1. Les opcodes de ces opérateurs sont :

**1. ANA :** And : And logique entre l'accumulateur A et le contenu du registre ou case mémoire.

**Syntaxe :** ANA R ou ANA M

Tel que R : Registre 8 bits et M : adresse d'une case mémoire

**Exemple :**

MVI A,81H

MVI B,77H

ANA B

Ce qui donne (A)=01H.

