

## Chapitre 3

### Jeux d'instruction du 8085:

L'ensemble des instructions du  $\mu$ P 8085 est partagé en 5 groupes principaux :

- Instructions chargement et stockage : LDA, STA, MOV reg1,reg2....
- Instructions arithmétique et logique : ADD, SUB, ORA....
- Instructions rotation et décalage :RAL,
- Instructions de branchement : JMP, JNZ,...
- Instructions spéciales sous programmes, PILE... : CALL , RET, PUSH, POP.

#### 1- Instructions chargement et stockage :

- LDA Adresse ;  $A \longleftarrow M(\text{Adresse})$   
Exp : LDA 8000
- STA Adresse ;  $A \longrightarrow M(\text{Adresse})$   
Exp : STA 8002

Exp : LDA 8000

STA 8004

RST 1

Exp : MVI A,05

STA 8000

- LXI rp,Data 16 ;  $rp \longleftarrow \text{data 16}$

Rp:register pair : BC, DE, HL.

Charger le registre pair par la donnée 16 bits immédiat.

Exp :LXI B,1234 ;  $\longrightarrow B=12, C=34$

LXI D,5678 ;  $\longrightarrow D=56, E=78$

LXI H,9ABC ;  $\longrightarrow H=9A, L=BC$

LXI SP 6800

- LDAX rp : charger l'acc par le contenu dont l'adresse est dans le registre pair, c'est un adressage Indirect : LDAX B, LDAX D...

Exp : LXI B, 8000

LDAX B

- STAX rp : stocker le contenu de l'accumulateur dans la case mémoire pointé par le registre pair rp.

A  $\longleftarrow$  M(rp)

Exp : LXI D,8000

STAX D

- Exercice:

- Soit à transférer le bloc mémoire 8000-8010 vers 8100-8110 :

Adresse :	Code op :	Assembleur :	Commentaires :
7000	26 10	MVI H, 10	H $\longrightarrow$ 10
7002	01 00 80	LXI B, 8000	BC $\longrightarrow$ 8000
7005	11 00 81	LXI D,8100	DE $\longrightarrow$ 8100
7008	0A	LDAX Btiq	A $\longleftarrow$ M(BC)
7009	12	STAX D	M(DE) $\longleftarrow$ A
700A	0C	INR C	C $\longrightarrow$ C+1
700B	1C	INR E	E $\longrightarrow$ E+1
700C	25	DCR H	H $\longrightarrow$ H-1
700D	C2 08 70	JNZ etiq	Si H $\neq$ 0 alors va à etiq
7010	CF	RST 1	FIN

- LHLD,Adr ; charger le reg L par contenu de l'Adr et le reg H par le contenu de Adr+1 :

L  $\longrightarrow$  M(Adr)

H  $\longrightarrow$  M(adr+1)

- SHLD, Adr : stocker le contenu de reg L dans l'Adr et le contenu de reg H dans Adr+1 :

Exp : SHLD 8200

M(8200)  $\longleftarrow$  L

M(8201)  $\longleftarrow$  H

- A voir sur net :
  - XCHG

## 2- Les instructions arithmétiques et logiques :

### a) Arithmétique :

- ADI data : additionner à l'accumulateur la donnée data :

$Ac \longleftarrow Ac + data$

Exp : MVI A, 10

ADI 25

- ADD reg : additionner à l'accumulateur le contenu du registre reg :

$Ac \longleftarrow Ac + reg$

Exp : MVI A, 25

MVI B, 48

ADD B

- ADD M: additionner à l'accumulateur le contenu de la case mémoire pointée par le registre pair HL :

$Ac \longleftarrow Ac + M(HL)$

Exp: LXI H, 8000

ADD M; ajouter à l'Acc le contenu de la case mémoire 8000

- ACI data : additionner à l'accumulateur la donnée data en tenant compte de la retenue :

$Ac \longleftarrow Ac + data + C$

Exp: MVI A, 45

ACI E3

- ADC reg: additionner à l'accumulateur le contenu du registre reg en tenant compte de la retenue

$Ac \longleftarrow Ac + reg + C$

Exp : MVI A, 25

MVI B, 48

ADC B

### • Exercice :

- Réaliser l'addition de deux mots à 16 bits stockés en mémoire 8000, 8001 et 8010, 8011 et ranger le résultat dans 8020 et 8021.
- Réaliser l'addition de trois cases mémoires : 8000, 8001, 8002 et ranger le résultat dans la case 8010.

### 3- Les instructions arithmétiques et logiques :

#### b) Arithmétique (la suite) :

➤ SUI Data ; soustraire data de l'accumulateur :  $Ac \longleftarrow Ac - data$

Exp : MVI A, 45

SUI 20

➤ SUB reg ; soustraire de l'accumulateur le contenu du registre reg :  $Ac \longleftarrow Ac - reg$

Exp : MVI A, 55

MVI B, 25

SUB B

➤ SUB M ; soustraire de l'accumulateur le contenu de la case mémoire M pointé par HL :

Exp : LXI H, 8000                      M(8000)= 25

MVI A, 60

SUB M

Algorithme pour réaliser des soustractions :

The detailed procedure for subtracting multi-digit decimal numbers is as follows:

1. Set the carry flag = 1 to indicate no borrow.
2. Load the accumulator with 99H, representing the number 99 decimal.
3. Add zero to the accumulator with carry, producing either 99H or 9AH, and resetting the carry flag.
4. Subtract the subtrahend digits from the accumulator. producing either the nines or tens complement.
5. Add the minuend digits to the accumulator.
6. Use the DAA instruction to make sure the result in the accumulator is in decimal format, and to indicate a borrow in the carry flag if one occurred.
7. If there are more digits to subtract, go to step 2. Otherwise. stop.

➤ SBI Data ; soustraire data de l'accumulateur avec empreinte :  $Ac \longleftarrow Ac - data - C$  :

Exp : MVI A, 25

SBI 1E

➤ SBB reg; soustraire le contenu du registre reg de l'accumulateur avec empreinte :

$Ac \longleftarrow Ac - reg - C$

Exp : MVI A, 35

MVI B, 1F

SBB B

➤ SBB M ; soustraire de l'accumulateur le contenu de la case mémoire pointé par HL avec empreinte :

$Ac \longleftarrow Ac - M(HL) - C$

Exp : MVI A, 46  
LXI H, 8000  
SBB M

c) Les instructions logiques : AND, OR, XOR :

- ANI data, ET-logique du contenu de l'accumulateur avec data :  
L'opérateur AND permet de masquer certains bits  
Ac ← Ac AND data (utilisé généralement comme masque)

Exp : MVI A, 55  
ANI 0F

Exp1: pour tester le bit 4 du registre C

MVI C,15  
MOV A,C  
ANI 10  
MOV C,A

- ANA reg ; ET-logique de l'accumulateur avec registre reg : Ac ← Ac AND reg

Exp : MVI A, 55  
MVI B, AA  
ANA B

- ANA M ; ET-logique du contenu de l'accumulateur avec le contenu de la case mémoire pointé par HL : Ac ← Ac AND M(HL)

Exp : MVI A, 35  
LXI H, 8000            M(8000) = F0  
ANA M

- ORI data ; OR-logique du contenu de l'accumulateur avec data : Ac ← Ac OR data

Exp : MVI A, 55  
ORI 0F

- ORA reg ; OR-logique de l'accumulateur avec registre reg : Ac ← Ac OR reg

Exp : MVI A, 55  
MVI B, AA  
ORA B

- ORA M ; OR-logique de contenu de l'accumulateur avec le contenu de la case mémoire pointé par HL : Ac ← Ac OR HL

Exp : MVI A, 35  
LXI H, 8000            M(8000)= F0  
ORA M

- XRI data ; XOR-logique du contenu de l'accumulateur avec data :  
Ac ← Ac XOR data

Exp : MVI A, 68  
XRI FF

- XRA reg ; XOR-logique du contenu de l'accumulateur avec le contenu du registre reg :

Ac ← Ac XOR reg

Exp : MVI A, 55

MVI B, FF

XRA B            A = AA

- XRA M ; XOR-logique de l'accumulateur avec le contenu de la case mémoire pointé par HL :

Ac ← Ac XOR M(HL)

Exp : MVI A, 35

LXI H, 8000    M(8000)= F0

XRA M            A=C5

⋮