

Chapitre 3

Jeux d'instruction du 8085:

L'ensemble des instructions du μP 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 ; $A \leftarrow M(\text{Adresse})$
Exp : LDA 8000
- STA Adresse ; $A \rightarrow M(\text{Adresse})$
Exp : STA 8002

Exp : LDA 8000

STA 8004

RST 1

Exp : MVI A,05

STA 8000

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

Rp:register pair : BC, DE, HL.

Charger le registre pair par la donnée 16 bits immédiat.

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

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

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

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 pointée par le registre pair rp.

A $\rightarrow M(rp)$

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	BG $\longrightarrow$ 8000
7005	11 00 81	LXI D, 8100	DE $\longrightarrow$ 8100
7008	0A	LDAX Betiq	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 pointée 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 : $Ac \leftarrow Ac - data - C$:

Exp : MVI A, 25
SBI 1E

➤ SBB reg; soustraire le contenu du registre reg de l'accumulateur avec empreinte :
 $Ac \leftarrow Ac - reg - C$

Exp : MVI A, 35
MVI B, 1F
SBB B

➤ SBB M ; soustraire de l'accumulateur le contenu de la case mémoire pointée par HL avec empreinte :
 $Ac \leftarrow Ac - M(HL) - C$

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
 $Ac \leftarrow Ac \text{ AND data}$ (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 : $Ac \leftarrow Ac \text{ AND reg}$

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 pointée par HL : $Ac \leftarrow Ac \text{ AND } M(HL)$

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

- ORI data ; OR-logique du contenu de l'accumulateur avec data : $Ac \leftarrow Ac \text{ OR } data$

Exp : MVI A, 55
ORI 0F

- ORA reg ; OR-logique de l'accumulateur avec registre reg : $Ac \leftarrow Ac \text{ OR } reg$

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

- ORA M ; OR-logique du contenu de l'accumulateur avec le contenu de la case mémoire pointée par HL : $Ac \leftarrow Ac \text{ OR } M(HL)$

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

- XRI data ; XOR-logique du contenu de l'accumulateur avec data : $Ac \leftarrow Ac \text{ XOR } data$

Exp : MVI A, 68
XRI FF

- XRA reg ; XOR-logique du contenu de l'accumulateur avec le contenu du registre reg : $Ac \leftarrow Ac \text{ XOR } reg$

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 pointée par HL : $Ac \leftarrow Ac \text{ XOR } M(HL)$

Exp : MVI A, 35
LXI H, 8000 M(8000)= F0
XRA M A=C5

4- Les instructions Incrémentation et Décrémentation :

a) Les instructions d'incrémentation :

- INR reg ; increment the content of register reg : $reg \leftarrow reg + 1$

Reg : A, B, C, D, E, H, L.
Exp : MVI A, 00
eti INR A
 JNZ eti
 RST 1

- INR M; increment the content of memory pointed of register HL: $M(HL) \leftarrow M(HL) + 1$

Exp : LXI H, 8000 M(8000)=36
 INR H

- INX reg-pair ; incrémenter le registre pair :

Reg-pair : BC, DE, HL, SP
 Exp : LXI B,0000
 Etiquette INX B
 JNZ etiq

b) Les instructions décrémentation :

- DCR reg ; Decrement the content of register reg : $\text{reg} \leftarrow \text{reg} - 1$
 Exp : MVI A, 50
 DCR A
 JNZ etiq
 RST 1
- DCR M ; Decrement the content of memory pointed by register HL : $M(\text{HL}) \leftarrow M(\text{HL}) - 1$
 Exp : LXI H, 8000 $M(8000) = 00$
 DCR M $M(8000) = \text{FF}$
- DCX reg-pair : Décrémenter le registre reg- pair : $\text{reg-pair} \leftarrow \text{reg-pair} - 1$
 Exp : LXI D, 0000
 DCX D

- Travail à la maison To see in Internet the instructions below:

- DAD reg-pair
- DAA

5- Les instructions comparaison :

- CPI data ; comparer immédiatement le contenu de l'accumulateur avec data :
 $\text{Ac} - \text{data} = ? \Rightarrow \text{Si } \text{Ac} = \text{data} \longrightarrow Z=1$
 $\text{Si } \text{Ac} < \text{data} \longrightarrow C=1$
 Exp : MVI A, 25
 CPI 25
 Exp 2 : MVI A,00
 Etiquette Traitement (ADD ou SUB ou)
 CPI 30
 JNZ etiq
- CMP reg ; comparer le contenu de l'accumulateur avec le contenu du registre reg :
 $\text{Ac} - \text{reg} \longrightarrow \text{Si } \text{Ac} \neq \text{reg} \Rightarrow Z=1$
 $\text{Si } \text{Ac} < \text{reg} \Rightarrow C=1$
 Exp : MVI A, 00
 MVI B, 40
 Etiquette INR A
 CMP B
 JNZ etiq
- CMP M ; comparer le contenu de l'accumulateur avec le contenu de la case mémoire pointée par HL : $\text{Ac} - M(\text{HL}) \longrightarrow \text{Si } \text{Ac} = M(\text{HL}) \Rightarrow Z=1$
 $\text{Si } \text{Ac} < M(\text{HL}) \Rightarrow C=1$

```

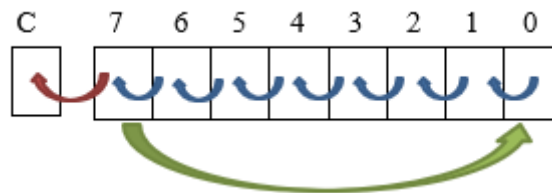
Exp : LXI H, 8000
      MVI A, 00
      etiq INR A
          CMP M
          JNZ etiq
          RST 1

```

06.03.2018

6- Les instructions de rotation et de décalage :

➤ RLC (Rotate Left with Carry):



$A_{i+1} \leftarrow A_i$; $A_0 \leftarrow A_7$ Et $C \leftarrow A_7$

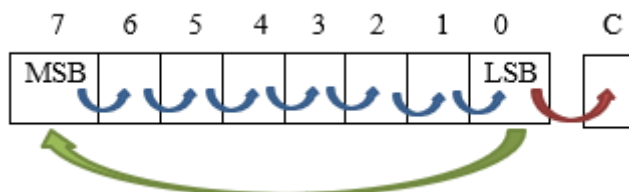
Exp : MVA 08

RLC

RLC

- Cette instruction peut être utiliser pour la multiplication d'un nombre par 2, et aussi pour tester le bit MSB.

➤ RRC (Rotate with Carry) :



$A_{i+1} \rightarrow A_i$; $A_0 \rightarrow A_7$ Et $C \leftarrow A_0$

Exp : soit à diviser le contenu de l'accumulateur par 4 :

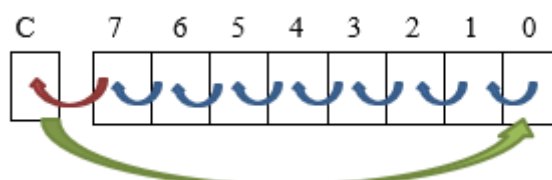
MVI A, 24 A=24

RRC A=12

RRC A=9

- Cette instruction peut être utiliser pour division d'un multiple de 2 par 2 et aussi pour tester le bit LSB.

➤ RAL (Rotate All Left) :



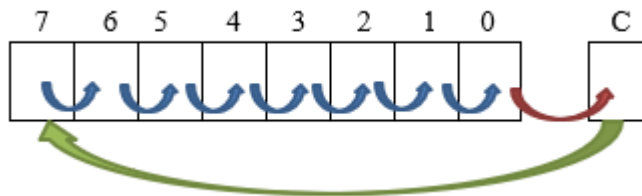
$A_{i+1} \leftarrow A_i ; A_0 \leftarrow C \quad \text{Et} \quad C \leftarrow A_7$

Exp : MVI A, 24

RAL; A = 12

RAL; A=9

➤ RAR (Rotate All Right):



$A_{i+1} \rightarrow A_i ; C \rightarrow A_7 \quad \text{Et} \quad C \leftarrow A_0$

Exp : soit diviser le contenu de l'accumulateur Ac par 4 :

MVI A, 40

RAR ; A=20

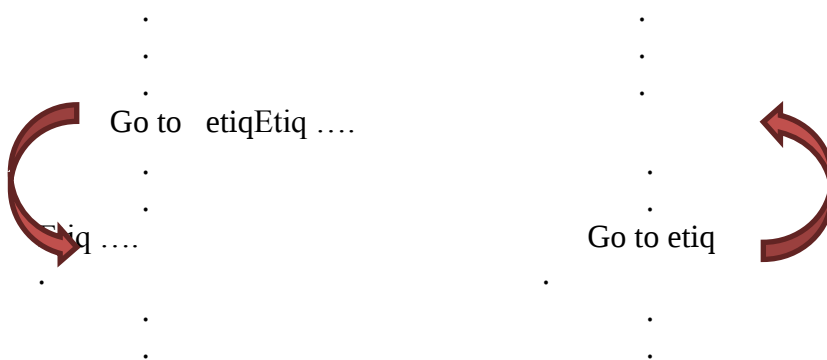
RAR ; A=10

- Cette instruction peut être utilisée pour diviser d'un multiple de 2 par 2 et aussi pour tester le bit LSB.

7- Les instructions de branchement et test :

a) Branchement inconditionnel : se brancher à une ligne de prog sans condition :

Exp : INST1 INST1



- Le branchement inconditionnel est utilisé pour sauter un ensemble d'instruction ou répéter l'exécution d'un ensemble d'instruction.

➤ JMP adresse ; saut sans condition à l'adresse :

Exp 1 : MVI A, 15

MVI B, 25

JMP etiq

ADD B

Etq RST 1



Exp2: Boucler un prog :

MVI A, 20

MVI B, 30

 etiq ADD B
 JMP etiq
 RST 1

Exp3: NOP

Etiqu traitement

JMP etiq

RST 1

b) Branchement conditionnel : se brancher à une ligne de programme sous condition :

Exp :INST1

.

.

 If condition thenGo to etiq

.

.

INST

.

Etiqu

.

.

.

- Ce type de branchement permet d'exécuter un traitement dans le cas où la condition est vraie.

➤ J Condition Adr ; saut sous condition à Adr.

- Condition : NZ : non Zéro : résultat ≠ 0

 Z : Zéro : résultat = 0

 NC : non Carry : retenue = 0

 C : Carry : retenue = 1

 PO : ParityOdd : Parité Impair (P=0 ??)

 PE : ParityEven : Parité pair (P=1 ??)

 P : Plus : S= 0

 M : Minus : S=1

Exp1 : MVI A, 08

Exp2: MVI A, 08

Etq1 DCR A
JNZ etiq
RST 1

etiq2 DCR A
JZ etiq 1
JMP etiq 2
RST 1 etiq 1