

Suite Chapitre 3 : jeux d'instruction

- 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

1- 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.

2- 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 : $Ac \leftarrow Ac - data - C$:

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 pointé 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 pointé 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 : $Ac \longleftarrow Ac \text{ OR } reg$

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 : $Ac \longleftarrow Ac \text{ OR } M(HL)$

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

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

Exp : MVI A, 68
XRI FF

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

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 pointé par HL :
 $Ac \longleftarrow Ac \text{ XOR } M(HL)$

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

3- Les instructions Incrémentation et Décrémentation :

a) Les instructions d'incrémentations :

- INR reg ; increment the content of register reg : $reg \longleftarrow reg+1$
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: $M(HL) \longleftarrow M(HL)+1$
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
Etiq INX B

JNZ etiq

b) Les instructions décrémentation :

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

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

- DAD reg-pair
- DAA

4- Les instructions comparaison :

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

Exp : LXI H, 8000

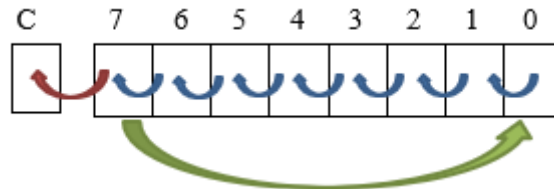
```

MVI A, 00
etiq INR A
    CMP M
    JNZ etiq
    RST 1

```

5- Les instructions de rotation et de décalage :

➤ RLC (Rotate Left with Carry):



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

Exp : MVA 08

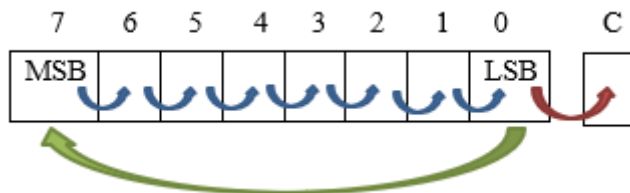
```

RLC
RLC

```

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

➤ RRC (Rotate with Carry) :



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

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

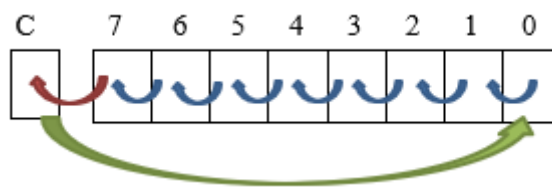
```

MVI A, 24      A=24
RRC           A=12
RRC           A=9

```

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

➤ RAL (Rotate All Left) :

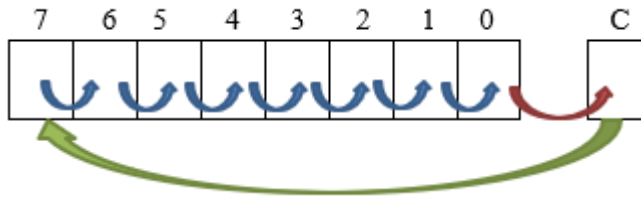


$$A_{i+1} \leftarrow A_i ; A_0 \leftarrow C \text{ Et } C \leftarrow A_7$$

Exp : MVI A, 24

RAL; A = 12
 RAL; A=9

➤ RAR (Rotate All Right):



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

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

```
MVI A, 40
RAR ; A=20
RAR ; A=10
```

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

6- Les instructions de branchement et test :

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

Exp : INST1 INST1

