



**OBJECTIFS du TP :**

Dans ce TP, on abordera les points suivants :

- 1- ADDITION DE DEUX NOMBRES DE 8 BITS
- 2- SOUSTRACTION DE DEUX NOMBRES DE 8 BITS
- 3- MULTIPLICATION DE DEUX NOMBRES DE 8 BITS
- 4- ADDITION DE DEUX NOMBRES DE 16 BITS

**ACTIVITE 0:**(Préparation) Explorez les petits morceaux de code assembleur pour le uP8085 en examinant le contenu des registres et case mémoire utilisés à chaque pas d'exécution et expliquez que fait le programme

Address	Symbolic
2000H	MVI A,49H
2001H	
2002H	MVI B,4AH
2003H	
2004H	MVI C,4BH
2005H	
2006H	STA 6285H
2007H	
2008H	
2009H	HLT

Address	Contents
0H	LDA 9H
1H	ADD AH
2H	ADD BH
3H	SUB CH
4H	OUT
5H	HLT

Mnemonics
MVI A,17H
MVI B,2DH
ADD B
STA 5600H
INR A
MOV C,A
HLT

MOV A,B
MOV A,C
MOV B,A
MOV B,C
MOV C,A
MOV C,B

```
MVI C,03H
DCR C
JZ 0009H
JMP 0002H
HLT
```

Label	Instruction	Comment
	MVI C,03H	;Load counter with decimal.3
REPEAT:	DCR C	;Decrement counter
	JZ END	;Test for zero
	JMP REPEAT	;Do it again
END:	HLT	

Label	Mnemonic	Comment
	MVI A,00H	;Clear accumulator
	MVI B,0CH	;Load decimal 12 into B
	MVI C,08H	;Preset counter with 8
REPEAT:	ADD B	;Add decimal 12
	DCR C	;Decrement the counter
	JNZ REPEAT	;Test for zero
	HLT	;Stop it

Address	Symbolic
2000H	MVI A,00H
2001H	
2002H	MVI B,0CH
2003H	
2004H	MVI C,08H
2005H	
2006H	ADD B
2007H	DCR C
2008H	JNZ 2006H
2009H	
200AH	
200BH	HLT

```
MVI A, 38 H
ADI 87 H
DAA
HLT
```

Label	Mnemonic	Comment
	IN 02H	
	ANI 01H	
	JNZ YES	
	MVI A,4EH	
	JMP DONE	
YES:	MVI A,59H	
DONE:	MVI C,08H	;Load counter with 8
AGAIN:	OUT 04H	;Send LSB to port 4
	RAR	;Position next bit
	DCR C	;Decrement count
	JNZ AGAIN	;Test count
	HLT	

Label	Mnemonic	Comment
	IN 02H	
	ANI 01H	
	JNZ YES	
	MVI A,4EH	
	JMP DONE	
YES:	MVI A,59H	
DONE:	MVI C,08H	;Load counter with 8
AGAIN:	OUT 04H	;Send LSB to port 4
	RAR	;Position next bit
	DCR C	;Decrement count
	JNZ AGAIN	;Test count
	HLT	

```
.ORG 1800h
START: LXI B,1234h
LXI D,5678h
PUSH B
PUSH D
POP B
POP D
HLT
```

**ACTIVITE 1:** Fournir un organigramme pour le programme suivant puis exécuter sur le simulateur en mode pas à pas en indiquant l'état des différents drapeaux Z, AC, CY, ... ETC et le contenu des registres et cases mémoires utilisés dans chaque pas.

**ADDITION:** Effectuer l'addition de deux nombres de 8 bits en utilisant le uP8085.

## ALGORITHMHE

- 1) Démarrez le programme en chargeant les premières données dans l'Accumulateur.
- 2) Copiez les données vers un registre (registre B).
- 3) Obtenez les deuxièmes données et chargez-les dans l'accumulateur.
- 4) Ajoutez les deux contenus du registre.
- 5) Vérifiez la retenue.
- 6) Stockez la valeur de la somme et conservez-la dans un emplacement mémoire.
- 7) Terminez le programme.

## CODE SOURCE

```
MVI C, 00
LDA 1962
MOV B, A
LDA 1954
ADD B
JNC LOOP
INR C
LOOP: STA 2023
MOV A, C
STA 2024
HLT
```

**ACTIVITE 2:** Fournir un organigramme pour le programme suivant puis exécuter sur le simulateur en mode pas à pas en indiquant l'état des différents drapeaux Z, AC, CY, ... ETC et le contenu des registres et cases mémoires utilisés dans chaque pas.

**SOUSTRACTION :** Effectuer la soustraction de deux nombres de 8 bits en utilisant 8085.

## ALGORITHMHE

1. Démarrez le programme en chargeant les premières données dans l'accumulateur.
2. Déplacez les données vers un registre (registre B).
3. Obtenez les deuxièmes données et chargez-les dans l'accumulateur.
4. Soustrayez les deux contenus du registre.
5. Vérifiez la retenue.
6. Si la retenue est présente, prenez le complément à 2 de l'accumulateur.
7. Stockez la valeur de l'emprunt dans un emplacement mémoire.
8. Stockez la valeur de différence (présente dans l'accumulateur) dans un emplacement mémoire.
9. Terminez le programme.

## CODE SOURCE

```
MVI C, 00
LDA 1830 ; Input 06
MOV B, A
LDA 1971 ; Input 02
SUB B
JNC LOOP
CMA
INR A
INR C
LOOP: STA 2023
MOV A, C
STA 2027
HLT
```

**MULTIPLICATION :** Effectuer la multiplication de deux nombres de 8 bits en utilisant le uP8085.

## ALGORITHMHE

- 1) Démarrez le programme en chargeant la paire de registres HL avec l'adresse de l'emplacement mémoire.

- 2) Déplacez les données vers un registre (registre B).
- 3) Obtenez les deuxièmes données et chargez-les dans l'accumulateur.
- 4) Ajoutez les deux contenus du registre.
- 5) Vérifiez la retenue(carry).
- 6) Augmentez la valeur du carry.
- 7) Vérifiez si l'ajout répété est terminé et stockez la valeur du produit et conservez-le dans l'emplacement mémoire.
- 8) Terminez le programme.

#### CODE SOURCE

```

MVI D, 00
MVI A, 00
LXI H, 1969
MOV B, M ; Input FF
INX H
MOV C, M ; Input FF
LOOP: ADD B
JNC NEXT
INR D
NEXT: DCR C
JNZ LOOP
STA 1969
MOV A, D
STA 1970
HLT

```

#### QUESTIONS RELATIVES À LA PROCHAINE EXPÉRIENCE :

1. Qu'est-ce que l'instruction XCHG ?
2. Qu'est-ce que l'enseignement DAD ?
3. Expliquez les instructions des CFF.
4. Expliquez l'instruction SPHL.
5. Différence entre SHLD et STA.

**ACTIVITE 3:** Fournir un organigramme pour le programme suivant puis exécuter sur le simulateur en mode pas à pas en indiquant l'état des différents drapeaux Z ,AC,CY,...ETC et le contenu des registres et cases mémoires utilisés dans chaque pas.

ADDITION 16bits: Effectuer l'addition de deux nombres de 16 bits en utilisant le uP8085.

#### ALGORITHMME

1. Initialisez les MSB de la somme à 0
2. Obtenez le premier nombre.
3. Ajoutez le deuxième nombre au premier nombre.
4. S'il y a un report, incrémentez les MSB de la somme de 1.
5. Stockez les LSB de la somme.
6. Stocker les MSB de la somme

#### CODE SOURCE

```

LHLD 2001H ; Input 7766
XCHG
LHLD 2003H ; Input 4422
MVI C, 00H
DAD D
JNC LOOP
INR C
LOOP: MOV A, C
STA 7502H
SHLD 3000H
HLT

```