

Les registres interne du μP :

- 1- Accumulateur Ac (registre de travail) :registre à 8bits très utilisés dans le μP , il est exploité dans toutes les opérations arithmétiques et logiques.
Exp : LDA Adresse ; $\text{Acc} \longrightarrow \text{M(Adresse)} : \text{LDA } 8000$
 STA Adresse ; $\text{Acc} \longleftarrow \text{M(Adresse)} : \text{STA } 8005$
ADI data ; $\text{Acc} \text{Acc} + \text{Data} : \text{ADI} \longleftarrow 25$
- 2- Le compteur programme (PC: Prog Counter) :registre 16 bits, il pointe à l'adresse de la prochaine instruction à exécuter. Il permet d'adresser 2^{16} cases mémoires.
- 3- Le pointeur PILE (SP : Stack Pointer) :registre à 16 bits, utilisé dans l'exploitation d'une partie de la mémoire RAM en mémoire de type LIFO (Last In First Out) appelée PILE, il pointe au top de cette pile (le haut de la mémoire LIFO) il est incrémenté et décrémenté automatiquement à chaque opération de chargement et stockage dans la pile.
- 4- Le registre Index (i.d.R : index reg) :registre à 8bits, ou 16 bits il permet d'utiliser un mode d'adressage indexé. Dans le cas du μP 8085 le registre HL (16 bits) est utilisé comme registre d'index
- 5- Le registre d'état (SR : StatusRegister : Flag) : registre à 8bits indique l'état du μP après exécution d'une instruction, il fournit l'information concernant la dernière opération exécutée. Dans le cas du μP 8085 les bits du Flag sont comme suit :



- a) Le bit R (C) Carry : indique une retenue après l'opération :

$C=0 \longrightarrow$ pas de retenue

$C=1 \longrightarrow$ retenue

Exp : MVI A, F0

ADI 25 $A \longleftarrow F0+25$

- b) Le bit P (Parity) : indique si le résultat est pair/impair :

$P=1 \longrightarrow$ résultat pair

$P=0 \longrightarrow$ résultat impair

- c) Le bit Ac (Axxillary Carry) :indique un demi retenu dans l'opération

- d) Le bit Z (zéro) : indique si le résultat est nul :

$Z=1 \longrightarrow$ résultat nul

$Z=0 \longrightarrow$ résultat $\neq 0$

Exp : MVI A,1C

Les modes d'adressage :

- Les modes d'adressage caractérisent la façon dont est obtenue l'adresse du 2eme opérande d'une instruction. Pour un μ P avec différents modes d'adressage la programmation sera puissante parmi les modes d'adressages le plus répondu dans le μ P :
 - 1- Le mode implicite : l'opération n'a pas besoin de données ou adresse pour être exécuter.
Exp : INR A, RST 1, DCR B, NOP...
 - 2- Le mode immédiat : la donnée suit immédiatement l'opération dans une instruction.
Exp : MVI A,data
LXI H, adresse
 - 3- Le mode direct : l'adresse de l'opérande (donné) suit directement l'opération.
Exp : LDA Adresse ; LDA 8000 ; A \longrightarrow M(8000)
STA Adresse ; STA 8010 ; M(8010) \longrightarrow A
- 25.02.2018

Adressage direct :

- 1- Direct page zéro : Dans le mode d'espace mémoire est partagé en pages de 256 octets donc on peut avoir 256 pages. Les cases mémoires adressable en page '0' sont accédé avec le mode « Direct page zéro »

Exp : LDA 50 ; l'adresse effective : 0050.

- 2- Direct étendu : Dans ce mode l'adresse est définie par 2 octets qui suivent le code opération.

Exp : LDA 1240 ; AcM(1240)

Adressage Indirect :

L'adresse effective = le contenu du registre ou de l'adresse mémoire.

Syntaxe : code opératoire [reg ou adresse]

Exp : LDA [1240]

- Cette technique d'adressage donne lieu à différents modes :
 - Adressage indirect par mémoire.
 - Adressage indirect par registre.
 - Adressage indirect par registre avec déplacement.

➤ Code opération [reg], D

Adresse effective = (contenu du registre) + déplacement

- Code opération [reg, D] ;

Adresse effective= (contenue (du registre+ déplacement)

Exp : LDA [HL], 40

- Adressage indirect par registre post incrémentation :

- Code opération [reg]+

Exp : LDA [HL]+

- Adressage indirect pré-décrément par registre :

- Code opération [reg]-

Exp : LDA [HL]-

Adressage indexé :

Dans ce mode le μP utilise des registres internes spéciaux qu'on appelle registre d'index.

Dans le cas de I-8085 le registre d'index est le registre pair 'HL'.

Ce mode permet de manipuler un bloc facilement :

- Code opération R_{index}

Adresse effective : contenue (R_{index})

Exp : LDA R_{index} ; $Ac \rightarrow M(R_{index})$

MOV A,M ; $A \leftarrow M(HL)$

- Charger l'Acc par le contenu de la case mémoire pointé par HL.

- **Exercice :**

- Ecrire un programme qui permet d'effacer l'espace mémoire 8000 à 8010.

- Solution :

Adresse :	Code op :	Assembleur :	Commentaires :
7000	2E 00	MVI L, 00	HL ← 8000
7002	26 80	MVI H, 80	
7004	06 10	MVI B, 10	B=10
7006	36 00	MVI M, 00 etiq	M(HL) ← 00 etiq
7008	2C	INR L	L ← L+1
7009	05	DCR B	B ← B-1
700A	C2 06 70	JNZ etiq	Si B≠0 va à etiq
700D	CF	RST 1	FIN

- A voir sur net :
 - L'adressage indirect-indexé.
 - L'adressage indexé-indirect.

Jeux d'instruction :

L'ensemble des instructions du μP 8085 est partagé en 5 groupes principales :

- Instructions chargement et stockage : LDA, STA, MOV reg1,reg2....
- Instructions arithmétique et logique : ADD, SUB, ORA....
- Instructions rotation et décalage : RLC,RRC,RAL,RAR
- Instructions de branchement : JMP, JNZ,JC, JM,JP
- Instructions spéciales : sous programmes, PILE, interruptions

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, SP.

Charger le registre pair par la donnée 16 bits immédiat.

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

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

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

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...

$M(rp) \longrightarrow A$ note rp: BC,DE,HL, XP

Exp : LXI B, 8000

LDAX B

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

$A \longrightarrow M(rp)$

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 B	$A \longleftarrow M(BC)$
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≠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) ←L—

M(8201) ←H—

- A voir sur net :
- XCHG

2- Les instructions arithmétiques et logiques :

a) Arithmétique :

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

Ac ←— Ac+data

Exp : MVI A, 10

ADI 25

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

Ac ←— Ac+reg

Exp : MVI A,25

MVI B,48

ADD B

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

Ac ←— 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 ←— 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 ←— 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.