

**OBJECTIFS du TP :**

Dans ce TP, on abordera les points suivants :

- 1-Architecture d'un microprocesseur en générale
- 2-Modes d'adressage 8085 direct, indirect, immédiat, implicite, registre ...etc.
- 3- L'architecture du microprocesseur 8085.( Discutez les fonctions du compteur de programme, du pointeur de pile et des indicateurs d'état , des fonctions des signaux ALE,WR, RD, S0 et S1du microprocesseur 8085).
- 4-Brochage du uP 8085
- 5-Langage assembleur pour 8085
- 6-Intiation à un logiciel assembleur pour le 8085 et exploration de quelques instructions de base.

**Question 1:** Discuter avec l'enseignant le morceau de code en assembleur suivant écrit pour un type de microprocesseur.

Mnémonique Code Op	Code Machine (Binaire)	Opération
LDA 5H	0000 0101	Il charge l'accumulateur du contenu à partir de l'emplacement adressé.
ADD CH	0001 1100	Il ajoute le contenu de l'emplacement mémoire au contenu de l'accumulateur.
SUB BH	0010 1011	Il soustrait le contenu de l'emplacement mémoire du contenu de l'accumulateur.
OUT 03H	11100111	Il transfère le contenu de l'accumulateur vers le port de sortie d'adresse 03.
HLT	1111xxxx	Il arrête l' exécution.

**Question 2 :** Quel sera le contenu de l'accumulateur et des drapeaux (CY, S, P et Z), après l'exécution de l'instruction ADD D si A = C3 H and D = 3D H.

**Question 3:** Quel sera le contenu de l'accumulateur et des drapeaux (CY, S, P et Z), après l'exécution de l'instruction SUB D si A = C3 H and D = 3D H.

---

**Question 4:** Quel sera le contenu de l'accumulateur et du registre des drapeaux, après l'exécution du programme suivant

```
MVI A, 47 H
MVI B, 37 H
ADD B
DAA
HLT
```

**Question 5:** Quelle sera la valeur des drapeaux accumulateur, CY et AC après l'exécution du programme suivant :

---

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

---

**Question 6 :** Écrire un programme pour additionner deux nombres stockés dans les emplacements mémoire 2100 H et 2201 H sans utiliser les instructions LDAX et STAX. La réponse est à charger dans l'emplacement mémoire 2100 H. Vérifier par le résultat par simulation.

---

```
LXI H, 2201 H ; Loads H = 22 H and L = 01 H
LXI D, 2100 H ; Loads D = 21 H and E = 00 H
MOV A, M      ;  $[A] \leftarrow [M_{2201H}]$ 
XCHG          ; H = 21 H, L = 00 H and D = 22 H, E = 01 H
ADD M         ;  $[A] \leftarrow [A] + [M_{2100H}]$ 
MOV M, A      ;  $[M_{2100H}] \leftarrow [A]$ 
HLT
```

**Question 7 :** Quel sera le contenu de l'emplacement mémoire 2500 H et des drapeaux (CY, S, P et Z), après l'exécution du programme suivant

---

```
MVI C, C8 H
MVI A, 11 H
ADD C
STA 2500 H
HLT
```

**Question 8 :** Quelle sera la valeur du pointeur de pile, une fois le programme suivant exécuté par le **uP8085**.

---

```
LXI SP, 27FF H
PUSH D
CALL SUBROUT
POP D
ADD D
PUSH D
HLT
```

**Question 9 :** Quel sera le contenu du registre B, après l'exécution du programme en langage assembleur **uP8085** suivant ?

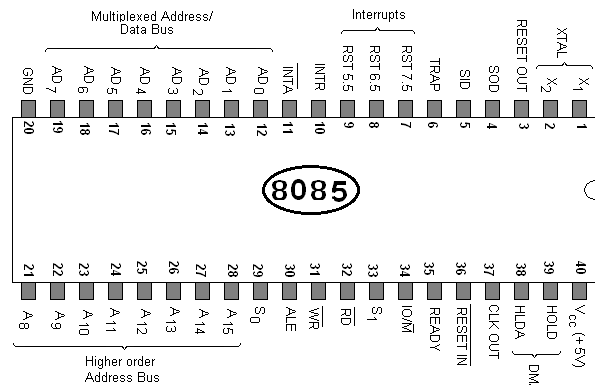
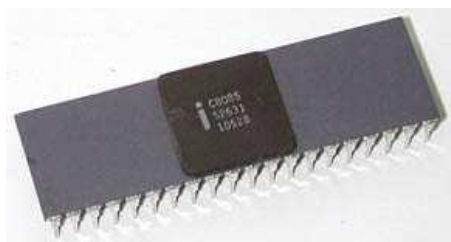
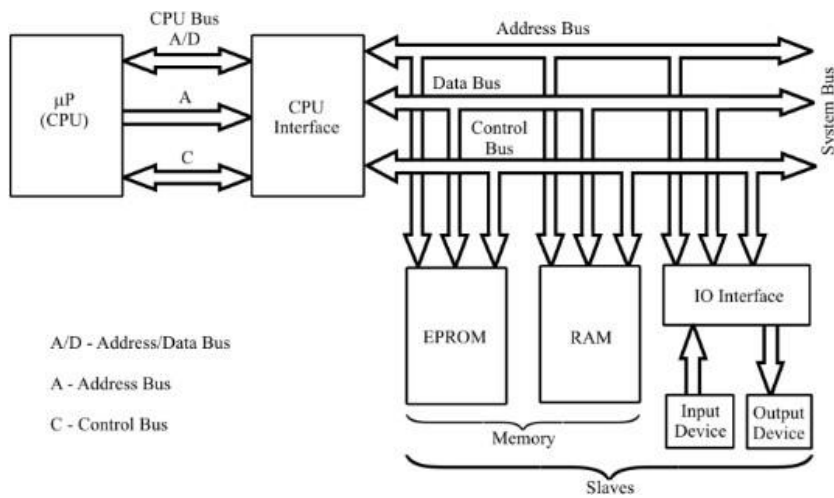
---

```
MVI A, 08 H
STC
CMC
RAL
MOV B, A
HLT
```

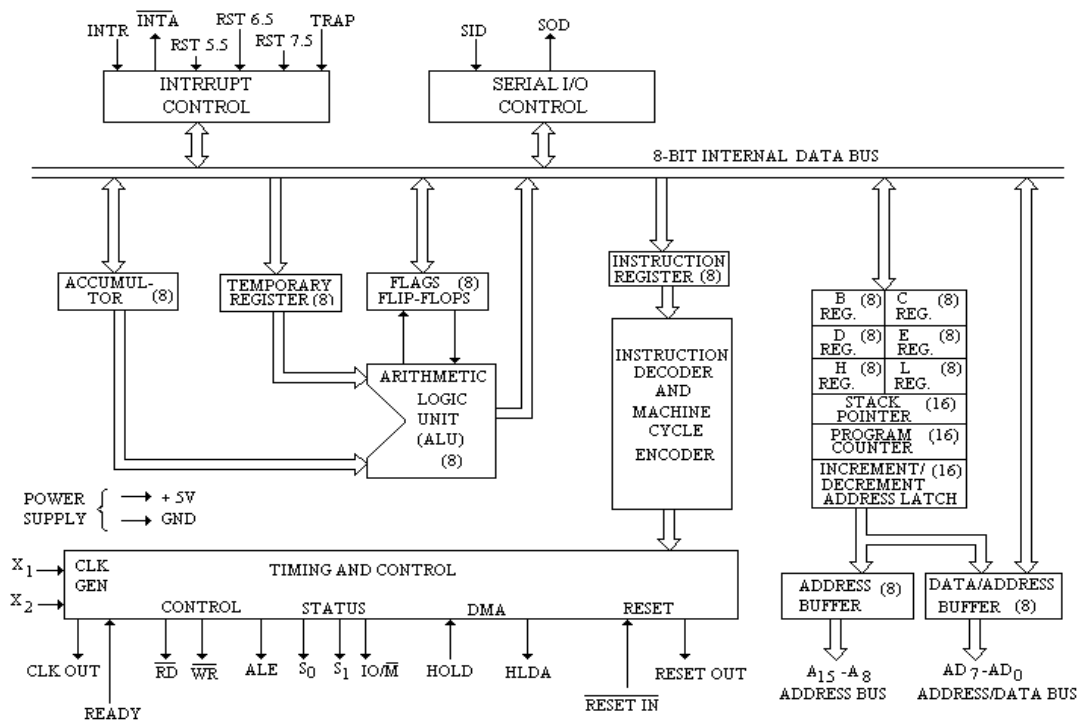
**Question 10 :** Écrivez un programme en langage assembleur de 8085 pour effacer les emplacements mémoire à partir de 2001 (c'est-à-dire que chaque emplacement doit avoir 00 H). La longueur des octets de données est donnée en 2000 H d'emplacement mémoire.

**Question 11 :** Écrivez un programme en langage assembleur de 8085 pour ajouter cinq octets de données stockées dans des emplacements mémoire à partir de 2000 H. Si la somme génère un report, arrêtez l'addition et stockez 01 H dans l'emplacement mémoire 2501 H ; sinon, continuez l'addition et stockez la somme à 2501 emplacements mémoire.

**Question 12 :** Écrivez un programme en langage assembleur 8085 pour tester le 6ème bit (bit D6) d'un octet stocké à l'emplacement mémoire 2500 H. Si le bit D6 est nul alors stockez 00 H à 2501 H sinon stockez le même numéro à 2501 H.



1.



S0	S1	Encoded status of the bus cycle
0	0	HALT
0	1	WRITE
1	0	READ
1	1	FETCH

# 8085A CPU INSTRUCTIONS IN OPERATION CODE SEQUENCE

Table 5-2

OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC	OP CODE	MNEMONIC
00	NOP	2B	DCX H	56	MOV D,M	81	ADD C	AC	XRA H	D7	RST 2
01	LXI B,D16	2C	INR L	57	MOV D,A	82	ADD D	AD	XRA L	D8	RC
02	STAX B	2D	DCR L	58	MOV E,B	83	ADD E	AE	XRA M	D9	—
03	INX B	2E	MVI L,D8	59	MOV E,C	84	ADD H	AF	XRA A	DA	JC Adr
04	INR B	2F	CMA	5A	MOV E,D	85	ADD L	B0	ORA B	DB	IN D8
05	DCR B	30	SIM	5B	MOV E,E	86	ADD M	B1	ORA C	DC	CC Adr
06	MVI B,D8	31	LXI SP,D16	5C	MOV E,H	87	ADD A	B2	ORA D	DD	—
07	RLC	32	STA Adr	5D	MOV E,L	88	ADC B	B3	ORA E	DE	SBI D8
08	—	33	INX SP	5E	MOV E,M	89	ADC C	B4	ORA H	DF	RST 3
09	DAD B	34	INR M	5F	MOV E,A	8A	ADC D	B5	ORA L	E0	RPO
0A	LDAX B	35	DCR M	60	MOV H,B	8B	ADC E	B6	ORA M	E1	POP H
0B	DCX B	36	MVI M,D8	61	MOV H,C	8C	ADC H	B7	ORA A	E2	JPO Adr
0C	INR C	37	STC	62	MOV H,D	8D	ADC L	B8	CMP B	E3	XTHL
0D	DCR C	38	—	63	MOV H,E	8E	ADC M	B9	CMP C	E4	CPO Adr
0E	MVI C,D8	39	DAD SP	64	MOV H,H	8F	ADC A	BA	CMP D	E5	PUSH H
0F	RRC	3A	LDA Adr	65	MOV H,L	90	SUB B	BB	CMP E	E6	ANI D8
10	—	3B	DCX SP	66	MOV H,M	91	SUB C	BC	CMP H	E7	RST 4
11	LXI D,D16	3C	INR A	67	MOV H,A	92	SUB D	BD	CMP L	E8	RPE
12	STAX D	3D	DCR A	68	MOV L,B	93	SUB E	BE	CMP M	E9	PCHL
13	INX D	3E	MVI A,D8	69	MOV L,C	94	SUB H	BF	CMP A	EA	JPE Adr
14	INR D	3F	CMC	6A	MOV L,D	95	SUB L	C0	RNZ	EB	XCHG
15	DCR D	40	MOV B,B	6B	MOV L,E	96	SUB M	C1	POP B	EC	CPE Adr
16	MVI D,D8	41	MOV B,C	6C	MOV L,H	97	SUB A	C2	JNZ Adr	ED	—
17	RAL	42	MOV B,D	6D	MOV L,L	98	SBB B	C3	JMP Adr	EE	XRI D8
18	—	43	MOV B,E	6E	MOV L,M	99	SBB C	C4	CNZ Adr	EF	RST 5
19	DAD D	44	MOV B,H	6F	MOV L,A	9A	SBB D	C5	PUSH B	F0	RP
1A	LDAX D	45	MOV B,L	70	MOV M,B	9B	SBB E	C6	ADI D8	F1	POP PSW
1B	DCX D	46	MOV B,M	71	MOV M,C	9C	SBB H	C7	RST 0	F2	JP Adr
1C	INR E	47	MOV B,A	72	MOV M,D	9D	SBB L	C8	RZ	F3	DI
1D	DCR E	48	MOV C,B	73	MOV M,E	9E	SBB M	C9	RET Adr	F4	CP Adr
1E	MVI E,D8	49	MOV C,C	74	MOV M,H	9F	SBB A	CA	JZ	F5	PUSH PSW
1F	RAR	4A	MOV C,D	75	MOV M,L	A0	ANA B	CB	—	F6	ORI D8
20	RIM	4B	MOV C,E	76	HLT	A1	ANA C	CC	CZ Adr	F7	RST 6
21	LXI H,D16	4C	MOV C,H	77	MOV M,A	A2	ANA D	CD	CALL Adr	F8	RM
22	SHLD Adr	4D	MOV C,L	78	MOV A,B	A3	ANA E	CE	ACI D8	F9	SPHL
23	INX H	4E	MOV C,M	79	MOV A,C	A4	ANA H	CF	RST 1	FA	JM Adr
24	INR H	4F	MOV C,A	7A	MOV A,D	A5	ANA L	D0	RNC	FB	EI
25	DCR H	50	MOV D,B	7B	MOV A,E	A6	ANA M	D1	POP D	FC	CM Adr
26	MVI H,D8	51	MOV D,C	7C	MOV A,H	A7	ANA A	D2	JNC Adr	FD	—
27	DAA	52	MOV D,D	7D	MOV A,L	A8	XRA B	D3	OUT D8	FE	CPI D8
28	—	53	MOV D,E	7E	MOV A,M	A9	XRA C	D4	CNC Adr	FF	RST 7
29	DAD H	54	MOV D,H	7F	MOV A,A	AA	XRA D	D5	PUSH D		
2A	LHLD Adr	55	MOV D,L	80	ADD B	AB	XRA E	D6	SUI D8		

D8 = constant, or logical/arithmetic expression that evaluates to an 8-bit data quantity.

Adr = 16-bit address.

D16 = constant, or logical/arithmetic expression that evaluates to a 16-bit data quantity.