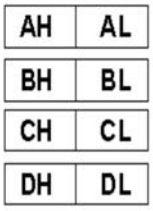
TP1

1-INTRODUCTION:

1-1-Le microprocesseur 8086:

- Microprocesseur des premiers PC et compatibles
- Compatibilité ascendante : un programme écrit pour le 8086 marche pour les processeurs suivants
- Caractéristiques :
  - – Bus de données : 16 bits
  - – Bus d'adresse : 20 bits
  - – Registres : 16 bits
  - » Accumulator (AX)
  - » Base (BX)
  - » Counter (CX)
  - » Data (DX)
  - » Pointeur d'instruction (IP)
  - » Registres segments code (CS), data (DS), extra (ES), stack (SS)
  - » Pointeur de pile (stack) (SP), de base (BP)
  - » Index source (SI), Index destination (DI)

Certains registres 16 bits peuvent être utilisés comme deux registres 8 bits



1-2-L'adressage par segment:

- Largeur du bus d 'adresse = 20 bits
  - ⇒Adressage mémoire = 2^{20} = 1 Mo
- Le pointeur d 'instruction fait 16 bits
  - ⇒Adressage = 2^{16} = 64 Ko cela ne couvre qu 'une partie de la mémoire
  - = 1 segment de mémoire
- Il faut 2 registres pour indiquer une adresse au processeur Ex: CS:IP

CS:IP, (index) DS:SI, DS:DI, ES:SI, ES:DI, (pile) SS:SP, SS:BP

1-3-Types d'adressage 8086:

- Adressage par registre (ou implicite)
  - Add AX, BX
  - Inc AX (pas d 'accès mémoire pour les opérandes)
- Adressage immédiat
  - Add AX, valeur (1 accès mémoire pour lire la valeur)
- Adressage direct
  - Add AX, [adresse] (2 accès mémoire : adresse puis valeur [adresse])
- Adressage indexé (ou relatif)
  - Add AX, [adresse+index] (2 accès mémoire)

1-4-Jeu d'instructions 8086:

- Instruction d'affectation MOV
- Instructions arithmétiques INC (incrémentation)

DEC (décrementation)
ADD (addition)
SUB (soustraction)
CMP (soustraction sans sauvegarde)

- Instructions logiques NOT, OR, XOR

AND, TEST (= AND sans sauvegarde)
SHL (SHR), SAL (SAR)
ROL (ROR), RCL (RCR)

NEG

2-MANIPULATION :

2-1- on complète les manquants du programme ci-dessous.:

On exécute le en mode pas à pas, le résultat dans le tableau suivant :

Code machine

	Binaire	Hexadécimale	Mnémonique	Description
0050:0100	101110001100110000000000	B8CC00	MOV AX, 00CC	; charger l'Acc par CCh
0050:0103	101110100011001100000000	BA3300	MOV DX, 0033	; charger DX par 33h
0050:0106	10010000	90	NOP	; aucun op à effectue

Etape	1	2	3	4
AX	0000	00CC	00CC	00CC
IP	0100	0103	0106	0009
DX	0000	0000	0033	0033

2-2- le programme en langage d'assemblage décrit ci-dessous:

L'exécution de mode pas à pas et décrire le processus d'exécution dans chaque étape (en se basant sur le changement des contenus des différents registres du CPU)

Code machine

	Binaire	Hexadécimale	Mnémonique	Description
0050:0010	101110000110011000000000	B86600	MOV AX, 0066	; charger l'Acc par 66h
0050:0013	101110110101010100000000	BB5500	MOV BX, 0055	; charger BX par 55h
0050:0016	01010000	50	PUSH AX	
0050:0017	01010011	53	PUSH BX	
0050:0018	101110110000000000000000	BB0000	MOV BX, 0000	
0050:001B	101110000000000000000000	B80000	MOV AX, 0000	
0050:001E	10000001111000101111101111111111	81E2FBFF	AND DX, FFFB	
0050:0022	01011011	5B	POP BX	
0050:0023	01011000	58	POP AX	

2-3- les Programmes en langage d' assemblage décrit ci –dessous :

		Binaire	Hexadécimale	Mnémonique	Description
Addition	0050:0100	10111000011001100000000	B80400	MOV AX, 0004	
	0050:0103	10111011000001010000000	BB0500	MOV BX, 0005	
	0050:0106	0000001111011000	03D8	ADD BX, AX	
Opération logique	0050:0108	10111000000001010000000	B80500	MOV AX, 0005	
	0050:010B	10111011001010110000000	BB2B00	MOV BX, 002B	
	0050:010E	1000101111010000	8BD0	MOV DX, AX	
	0050:0110	0010001111010011	23D3	AND DX, BX	
	0050:0112	1000101111010000	8BD0	MOV DX, AX	
	0050:0114	0000101111010011	0BD3	OR DX, BX	
	0050:0116	1000101111010000	8BD0	MOV DX, AX	
	0050:0118	0011001111010011	33D3	XOR DX, BX	
Soustraction	0050:011A	10111000000010000000000	B80800	MOV AX, 0008	
	0050:011D	00101101000001010000000	2D0500	SUB AX, 0005	
	0050:0120	00101101000010010000000	2D0900	SUB AX, 0009	
	0050:0123	00101101111111010000000	2DFA00	SUB AX, 00FA	

3- CONCLUSION :