Experiment 5

Arithmetic and Logical Instructions

Introduction:

In this experiment, you will be introduced to the arithmetic and logic instructions of the 8086 family of processors.

Objectives:

- Use logic instructions in assembly language
- Use arithmetic instructions in assembly language
- Use Shift and Rotate instructions in assembly language

Arithmetic Instructions:

The following table (Table 5.1) summarizes the arithmetic instructions used in the 8086 microprocessor. It also shows the effect of each instruction, a brief example, and the flags affected by the instruction. The "*" in the table means that the corresponding flag may change as a result of executing the instruction. The "-" means that the corresponding flag is not affected by the instruction, whereas the "?" means that the flag is undefined after executing the instruction.

Type	**Instruction**	**Example**	**Meaning**	**Flags Affected**					
				OF	SF	**ZF**	**AF**	PF	**CF**
Addition	ADD	ADD AX,7BH	AX ← AX + 7B	*	*	*	*	*	*
	ADC	ADC AX,7BH	AX ← AX + 7B +CF	*	*	*	*	*	*
	INC	INC [BX]	[BX]←[BX]+1	*	*	*	*	*	-
Subtraction	SUB	SUB CL,AH	CL ← CL – AH	*	*	*	*	*	*
	SBB	SBB CL,AH	CL ← CL – AH – CF	*	*	*	*	*	*
	DEC	DEC DAT	[DAT] ← [DAT] – 1	*	*	*	*	*	-
	NEG	NEG CX	CX ← 0 – CX	*	*	*	*	*	*
Multiplication	MUL	MUL CL MUL CX	AX ← AL * CL (DX,AX) ← AX* CX	*	?	?	?	?	*
	IMUL	IMUL BYTE PTR X IMUL WORD PTR X	AX ← AL * [X] (DX,AX) ← AX* [X]	*	?	?	?	?	*
Division	DIV	DIV WORD PTR X	AX ← (([DX,AX])/[X]) DX ←R(([DX,AX])/[X])	?	?	?	?	?	?
	IDIV	IDIV BH	AL ← Q(AX/BH) AH ←R(AX/BH)	?	?	?	?	?	?
Sign Extension	CBW	CBW	AH ← MSB(AL)	-	-	-	-	-	-
	CWD	CWD	DX ← MSB(AX)	-	-	-	-	-	-

5.1: Summary of Arithmetic Instructions of the 8086 microprocessor

ASSEMBLER PROGRAM

ADDITION:	**MULTIPLICATION**	
MOV AX , 4000 MOV BX , 0006 MOV CX , 8 ADC AX , BX LOOP 0009 INT 7	(8-bit) MOV AX , FF MOV CL , 6 MUL CL INT 7	(16-bit) MOV AX , FFFF MOV CX , 0200 MUL CX INT 7
SUBTRACTION	**DIVISION**	
MOV AX , 4000 MOV BX , 0006 MOV CX , 8 SBB AX , BX LOOP 0009 INT 7	(8-bit) MOV AX , 0400 MOV CL , 6 DIV CL INT 7	(16-bit) MOV DX , 23 MOV AX , 4 MOV CX , 300 DIV CX INT 7

OBSERVATIONS

• Using single stepping record the contents of AX register until CX becomes zero

Addition:

CX	AX	CX	AX	CX	AX
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____
_____	_____	_____	_____	_____	_____

–
_____ _____ _____ _____ _____ _____
–
_____ _____ _____ _____ _____ _____
–
_____ _____ _____ _____ _____ _____
–

Subtraction:

CX	AX	CX	AX	CX	AX
_____	_____	_____	_____	_____	_____
–					
_____	_____	_____	_____	_____	_____
–					
_____	_____	_____	_____	_____	_____
–					
_____	_____	_____	_____	_____	_____
–					
_____	_____	_____	_____	_____	_____
–					
_____	_____	_____	_____	_____	_____
–					

- Record the values of AX, BX, CX and DX before and after the execution of MUL/DIV instruction.

For Multiplication

8-bit: Before Execution of MUL: AX : _____ , BX : _____ CX : _____ , DX : _____	After Execution of MUL: AX : _____ , BX : _____ CX : _____ , DX : _____
16-bit: Before Execution of MUL: AX :_____ , BX :_____ CX : _____ , DX : _____	After Execution of MUL: AX : _____ , BX : _____ CX : _____ , DX : _____

For Division

8-bit: Before Execution of DIV: AX : _____ , BX : _____ CX : _____ , DX : _____	After Execution of DIV: AX : _____ , BX : _____ CX : _____ , DX : _____
16-bit: Before Execution of DIV: AX : _____ , BX : _____ CX : _____ , DX : _____	After Execution of DIV: AX : _____ , BX : _____ CX : _____ , DX : _____

EXERCISE 1

Write following program on the trainer and observe the changes in registers AX, BX, CX, DX, Flag and memory locations 0200 & 0201 by single stepping the program.

```
MOV AX , 2A34
MOV BX , 0200
```

```
MOV CX , 6A24
MOV WORD[BX] , D256
ADD AX , BX
SUB CX , [BX]
ADD WORD[BX] , 4829
SUB AX , 8245
ADD [BX] , AX
ADD CX , 32AD
INT 7
```

	After 1st Instruction	After 2nd Instruction	After 3rd Instruction	After 4th Instruction	After 5th Instruction
AX					
BX					
CX					
DX					
Flag					
WORD[0200]					
	After 6th Instruction	After 7th Instruction	After 8th Instruction	After 9th Instruction	After 10th Instruction
AX					
BX					
CX					
DX					
Flag					
WORD[0200]					

EXERCISE 2

Write a program which input ten 8-bit numbers as input from user and output their sum on LCD display

EXERCISE 3

Write a program, which calculate the factorial of any given number (the number may be used as an immediate operand in the instruction)

Logical Instructions:

Logic shift and rotate instructions are called bit manipulation operations. These operations are designed for low-level operations, and are commonly used for low- level control of input/output devices. The list of the logic operations of the 8086 is given in Table 5.1, along with examples, and the effect of these operations on the flags. The "*" in the table means that the corresponding flag may change as a result of executing the instruction. The "-" means that the corresponding flag is not affected by the instruction, whereas the "?" means that the flag is undefined after executing the instruction.

Instruction	Example	Meaning	Flags Affected				
			OF	SF	**ZF**	**AF**	PF
AND	AND AX, FFDFH	AX ← AX AND FFDFH	0	*	*	?*	*
OR	OR AL, 20H	AL ← AL OR 20H	0	*	*	?*	*
XOR	XOR NUM1, FF00	[NUM1] ← [NUM1] XOR FF00	0	*	*	?*	*
NOT	NOT NUM2	[NUM2] ← $\overline{[NUM2]}$	-	-	-	-	*

Table 5.2: Summary of the Logic Instructions of the 8086 Microprocessor

ASSEMBLER PROGRAM

```
MOV AX, 8A53
MOV BX, 0200
MOV CX, 692D
MOV DX, E6CB
MOV WORD [BX], 7B8A
AND AX, BX
AND CX, [BX]
OR [BX], CX
OR WORD [BX], 6F0C
XOR AX, 94D7
XOR DX, C4D1
 INT 7
```

OBSERVATIONS

By using single stepping record the contents of following registers:

Register	After 4th instruction	After 5th instruction	After 6th instruction	After 7th instruction	After 9th instruction	After 10th instruction
AX						
BX						
CX						
DX						
Flag						
Word[0200]						

EXERCISE 1

Write a program which mask the bits of AX register, by setting left-most 4 bits ,resetting right most 4 bits and complement bit position number 9 and 10.(Hint: Use AND,OR and XOR instructions for masking).