Computer Organization and Architecture (EET2211)

LAB I: Analyze the Arithmetic and Logical operationsusing different Addressing Modes of the 8086 Microprocessor.

Siksha 'O' Anusandhan(Deemed to be University), Bhubaneswar

Branch: CSE	Section: 2241026		
S. No.	**Name**	**Registration No.**	**Signature**
6	Dinanath Dash	2241004161	Dinanath Dash

Marks: _____/10

Remarks:

Teacher's Signature

I. OBJECTIVE:

1. Perform Addition, Subtraction, Multiplication, and Division of two 16-bit numbers using immediate addressing mode and store the results using direct addressing mode.
2. Perform the following operations on two 8-bit data (**data1, data2**) given in memory locations and store the result in another memory location using indirect addressing mode.
   i. Swapping of nibble of **data1**
   ii. Y= (**data1** and **data2**) or (**data1**xor**data2**)
3. Find the Gray code of an 8-bit binary number.
4. Find the 2's complement of an 8-bit number.

II. PRE-LAB

- Explain the addressing modes involved in instructions.

Ans- Addressing modes in computer instructions define how the CPU access operands, which typically data or address in memory or registers.

i) Immediate Addressing: In this mode, the operand is directly specified in the instruct

ii) Direct Addressing: The operand's memory location is directly specified in the instruction.

iii) Register Addressing: ~~The operand's mem~~. The operand's ~~memory~~ is in a process register.

iv) Indirect addressing: The instruction contains the address of a memory location that contains the actual address of the operand.

For each objective in prelab describe the following points:

- Write the assembly code with a description (ex. Mov ax,3000h – ax<-3000h)

- Examine and analyze the input/output of assembly code.

III. LAB

Note: For each objective do the following job and assessment:

- Screenshots of the Assembly language program (ALP)

For Obj. 1:

```
;Dinanath Dash
;2241004161

;Addition of two 16-bit numbers
mov ax, 3456h; value stored at ax
mov cx, ax; value of ax stored in cx
add ax, 2345h; value added in ax
mov [2000h], ax; value stored at memory location

;Subtraction of two 16-bit numbers
mov ax, cx; value of cx stored in ax
sub ax, 2345h; value subtracted from ax
mov [2002h], ax; value stored at memory location

;Multiplication of two 16-bit numbers
mov ax, cx; value of cx stored in ax
mov bx, 2345h; value added in bx
mul bx; value multiplied at bx
mov [2004], ax; value stored in memory location
mov [2006], dx; value stored in memory location

;Division of two 16-bit numbers
mov dx, 0000h; value stored at dx
mov ax, cx; value of cx stored at ax
div bx; value divided at bx
mov [2008h], ax; value stored at memory location
mov [200ah], dx; value stored at memory location
hlt; execution halted
```

For Obj 2:

a)

```
;Dinanath Dash
;2241004161

;Swapping of nibble of data1
mov al, [1000h]; value stored at memory location
rol al, 4; al rotated left by 4
ror al, 8; al rotated right by 8
hlt; execution halted
```

b)

```
;Dinanath Dash
;2241004161

;Y= (data1 and data2) or (data1 xor data2)
mov al, [1000h]; value stored at memory location
mov bl, [1010h]; value stored at memory location
mov ah, al; value of al stored at ah
and ah, bl; and operation on ah and bl
xor al, bl; xor opearation on al and bl
or ah, bl; or operation  ah and bl
hlt; execution halted
```

For Obj. 3:

```
;Dinanath Dash
;2241004161

;Gray code of an 8-bit binary number

mov al, [1000h]; value stored at memory location
mov bl, al; value of al stored in bl
shr al, 01; value of al shifted right by 1
xor bl, al; xor operation performed on bl and al
mov [1001h], bl; value stored at memory location
hlt; execution halted
```

For Obj. 4:

```
;Dinanath Dash
;2241004161

;2's complement of a 8-bit number

mov al, [1000h]; value stored at memory location
not al; complemennt of al
add al, 01h; value added at al
mov [1001h], al; value added to memory location
hlt; execution halted
```

- Observations (with screenshots)

For Obj. 1:

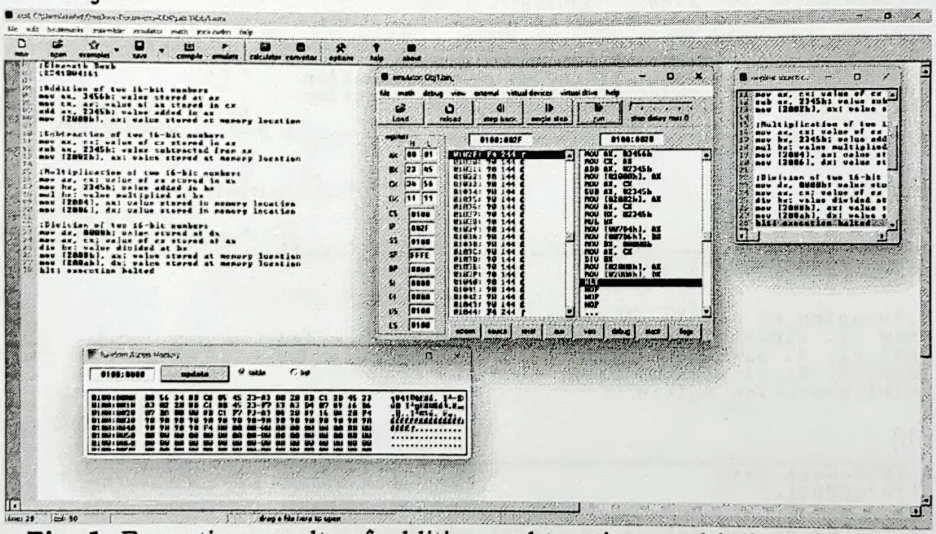

Fig. 1. Execution results of addition, subtraction, multiplication and division using immediate and direct addressing mode of 8086 emulator.

For Obj 2:

a)

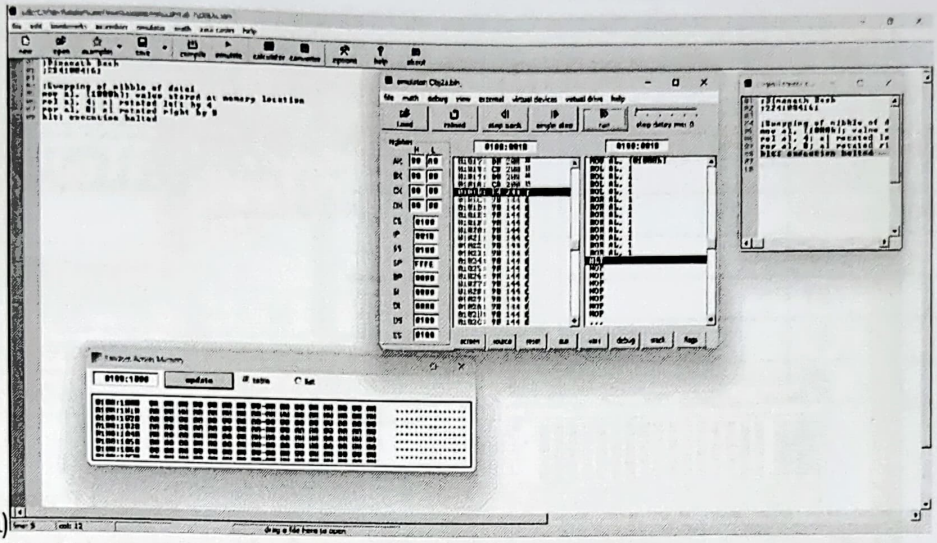

Fig. 2a. Execution results of swapping of nibble of **data1** using immediate and direct addressing mode of 8086 emulator.

b)

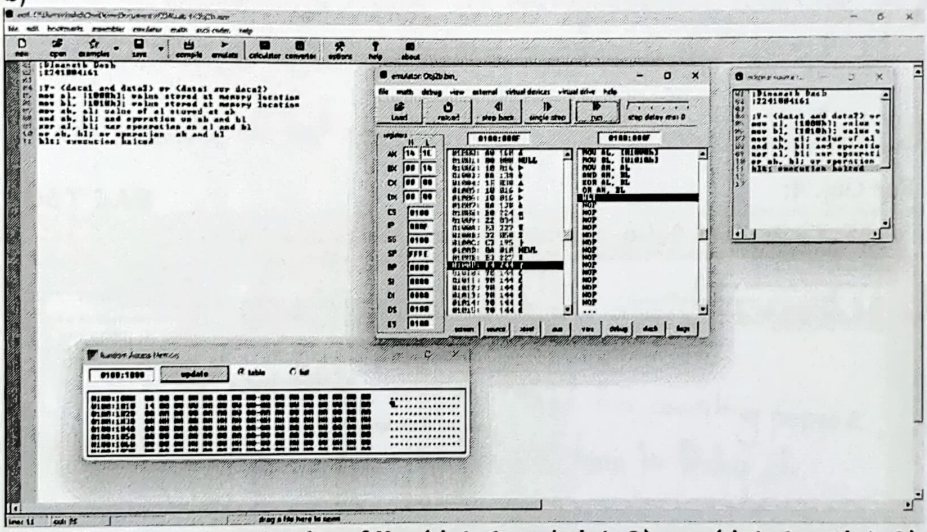

Fig. 2b. Execution results of *Y*= (**data1** and **data2**) or (**data1**xor**data2**) using immediate and direct addressing mode of 8086 emulator.

For Obj. 3:

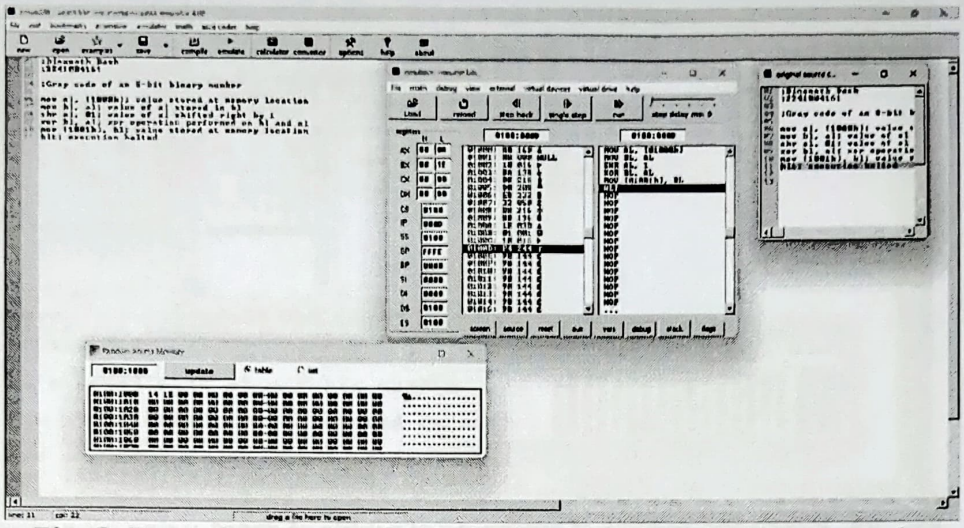

Fig. 3. Execution results of to find gray code of a 8-bit binary number using immediate and direct addressing mode of 8086 emulator.

For Obj. 4:

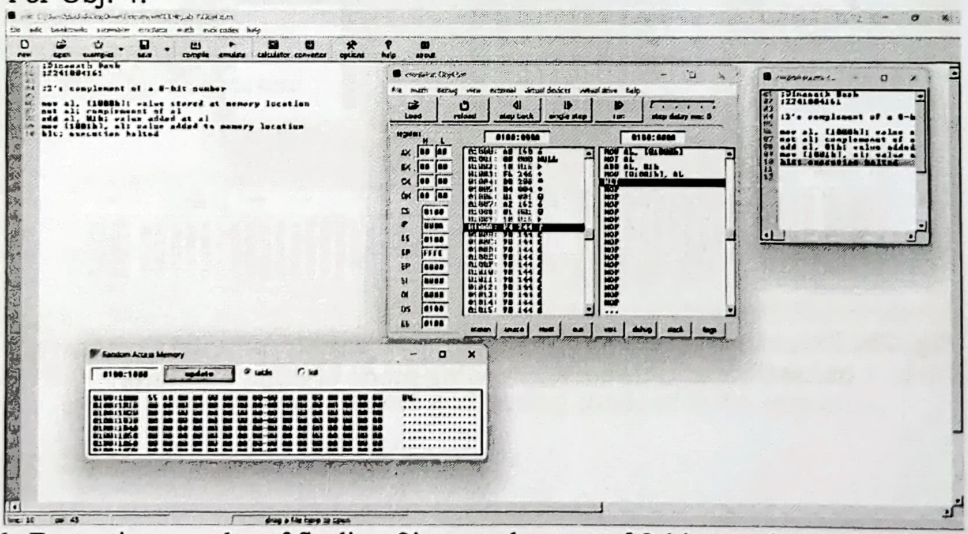

Fig. 4. Execution results of finding 2's complement of 8-bit number using immediate and direct addressing mode of 8086 emulator.

From this result, I have observed.....

Input:

Sl. No.	Memory Location	Operand (Data)

Output:

Sl. No.	Memory Location	Operand (Data)

1	-	Add (3456h, 2345h)
	-	Subtract (3456h, 2345h)
	-	Multiply (3456h, 2345h)
	-	Divide (3456h, 2345h)
2	1000h	al: A
	1010h	al: A, bl: 14
3	1000h	al: 14
4	1000h	al: 55

1	-	Addition: 579B
	-	Subtraction: 1111
	-	Multiplication: DD2E
	-	Division: 0001
2	-	A0
		1E
3	1001h	1E
4	1001h	AB

IV. CONCLUSION

In conclusion, the project demonstrated the feasibility of implementing a simple 8-bit microprocessor using an emulation platform. The microprocessor was able to perform basic arithmetic operations, bitwise operations, and control flow instructions. The implementation was verified using a series of test cases and the results were consistent with the expected results.

V. POST LAB

1. Discuss different general-purpose registers used in 8086 microprocessors.

Ans: The 8086 has 8-general purpose registers, each of which is 16 bit wide. They are ax, bx, cx, dx, si, di, bp and sp: used for counting pupose, like number of iterations while looping, number of characters in string, etc.

2. Explain the concept of segmented memory. What are its advantages?

Ans: Memory segmentation describes the system of segmenting process on a loading the information into non-contiguous spaces in memory. This allows for better efficiency in memory arrangement through the system incurs external fragmentation it allows for ~~chunks~~ smaller chunks of segmented process to be loaded.

3. Explain the physical address formation in 8086.

Ans: In the 8086 microprocessor, physical address generation involves combining the contents of the segment register with the offset address to form a 20-bit physical address. The segment register provides the base address, and the offset addr specifies the displacement from the base.

4. Write an assembly program to multiply 05H and 04H without using arithmetic instruction.

Ans:

```
;Dinanath Dash
;2241004161

;multiply 05H and 04H without using arithmetic instruction

mov ax, 0005h
mov bx, 0004h
mov cx, [2002h]
mov dx, 000h
mov si, 000fh
loop:
shr cx, 1
jc addition
addition:
add ax, bx
shr bx, 1
dec si
jnz loop
mov [2204h], ax
hlt
```

5. Write the function of the following logical instructions.

a) SHL/SAL b) SHR c) SAR d) ROR e) ROL

Ans:

a) SHL / SAL (Shift Left Logical/Arithmetic) : They shift the bits of th operand to the left.

b) SHR (Shift Right Logical): They shift the bits of the operand to the right.

c) SAR (Shift Right Arithmetic): It also shifts the bits of the operand to the right.

d) ROR (Rotate Right): It rotates the bits of the operand to the right.

e) ROL (Rotate Left): It rotates the bits of the operand to the left.