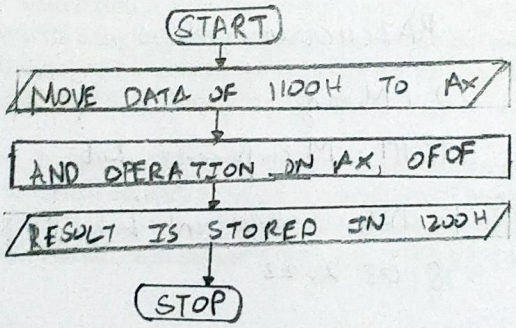
AND OPERATION



Ex.No.1 Data transfer and Logical operation using 8086

1.1 Introduction:

The purpose of this experiment is to learn about the registers, instruction sets, data transfer operation and logical operation of 8086 by using AND, OR in the given two 16-bit numbers and to store them in a memory location.

1.2 Hardware Requirement:

The 8086 Microprocessor kit, Power Supply.

1.3 Program Logic:

The logical AND instruction is used for masking off bits. The bits which have to be cleared are to be AND ed with a logical zero and other bits are to be one. Hence to achieve the above objective AND operation is performed between the data and the bits which has to be masked. Data AND with 0F0FH, will mask the bits 4, 5, 6, 7, 12, 13, 14 and 15.

In a similar manner, the logical OR has to be performed to set a particular bit. The bits which have to be set are to be OR ed with a logical one and other bits are to be zero. Hence to achieve the above objective OR operation is performed between the data and the bits which has to be set. Data OR with F0F0H, will set the bits 4, 5, 6, 7, 12, 13, 14 and 15.

1.4 Program:

Introduction of general purpose registers, data transfer instruction, logical instruction (AND, OR), immediate addressing, direct addressing:

ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV AX, [1100H]	8B06	Transfer Data from address 1100 to AX
1004		AND AX, 0F0FH	81E0	AND operation is Done
1008		MOV [1200H], AX	8906	Transfer Result to address 1200
100C		HLT	F4	Stop the program

Observation

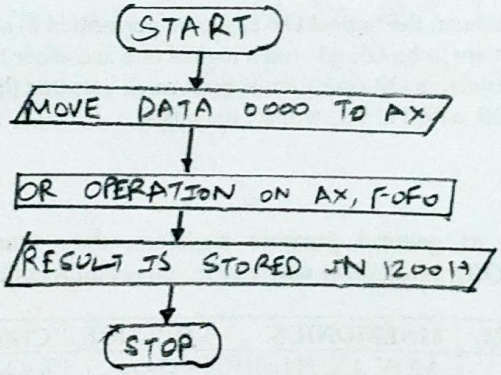
IN PUT ADDRESS	DATA
1100H	34
1101H	12

12 34
↓ ↓
H L

OUT PUT ADDRESS	DATA
1200H	04
1201H	02

0000 0010 0011 0100
0000 1111 0000 1111
0000 0010 0000 0100
0 2 0 4

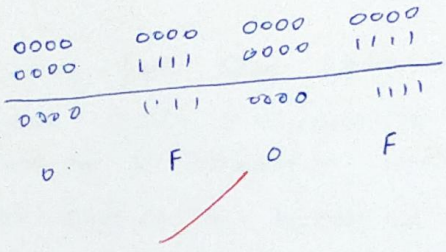
OR OPERATION



ADDRESS	LABEL	MNEMONICS	OPCODE	COMMENTS
1000		MOV AX, 0000H	C7C0	Transfer Data 0000 to AX
1004		OR AX, F0F0H	81C8	OR operation is Done with F0F0
1008		MOV [1200H], AX	8906	Transfer Result to address 1200
100C		HLT	F4	Terminate The Program

Observation

OUT PUT ADDRESS	DATA
1200H	0F
1201H	0F



1.5 Pre Lab Questions:

1. Mention the functions of BIU and EU.
2. Define BUS and give the classification of Buses
3. What is assembly level programming?

1.6 Post Lab Question:

1. Write an ALP to perform NAND and NOR operation.
2. Simulate the same using emulator 8086.

Result: Data transferres and logical operation is performed using 8086 and output is verified.

Microprocessor Lab

Experiment 1 : Data transfer and Logical operation using 8086

I The Lab Questions

1. Mention functions of BIU and EU.

Soln. BIU takes care of all data and addresses transfers on the buses for the EU like sending addresses, fetching instructions from the memory, reading data from the ports and the memory as well as writing data to the ports and the memory. EU has no direction connection with System Buses so this is possible with the BIU

2. Define BUS and give the Classification of Buses

Soln. A BUS is the electrically conducting path along which data is transmitted inside any digital electronic device. A Computer bus consists of a set of parallel conductors, which may be conventional wires, copper tracks, or micro strip aluminium trails on the surface of a silicon chip.

Data Bus: allows data to travel back and forth between the Microprocessor (CPU) and memory (RAM)

Address Bus: The address bus carries information about the location of data in memory.

Control Bus: The control bus carries the control signals that make sure everything is flowing smoothly from place to place.

Expansion Bus: Messages and information pass between Computer and the add-in boards plugged in over the expansion bus

3. What is assembly level programming?

Soln. Assembly language, is a low level programming language with a very strong correspondence between the instructions in the language and the architecture machine code instructions.

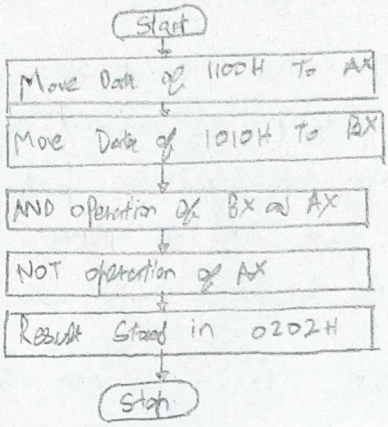
II Post Lab Questions

1. Write an ALP to perform NAND and NOR operation

Soln

ALP (NAND)

```
MOV AX, 1100H;
MOV BX, 1010H;
AND AX, BX;
NOT AX;
MOV [0202H], AX;
HLT;
```



Calculation:

NAND: AND op

$AX \cdot BX = 1100H \cdot 1010H$

$= 0001\ 0001\ 0000\ 0000$
$\quad 0001\ 0000\ 0001\ 0000$
$\quad 0001\ 0000\ 0000\ 0000$

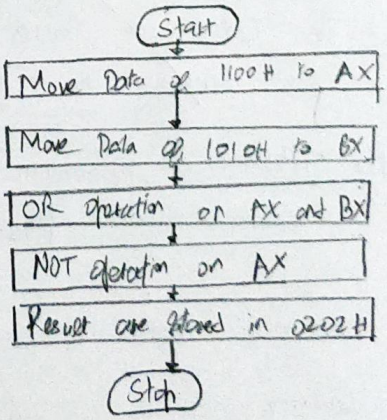
NOT op:

$\overline{AX} = 1110\ 1111\ 1111\ 1111$

$= FFFF \rightarrow$ O/P BIT.

ALP (NOR)

```
MOV AX, 1100H
MOV BX, 1010H
OR AX, BX
NOT AX
MOV [0202H], AX
HLT
```



Calculation:

NOR: OR op

$Ax + Bx = 1100H + 1010H$

$0001\ 0001\ 0000\ 0000$
$0001\ 0000\ 0001\ 0000$
$0001\ 0001\ 0001\ 0000$

NOT OP:

$\overline{AX} = 1110\ 1110\ 1110\ 1111 = FEEF \Rightarrow$ o/p BIT