**Lab Manual # 04**

**Implementation of basic Addressing modes using EMU8086**

**ADDRESSING MODES:**

An addressing mode specifies, how to calculate the effective memory address of an operand by using information held in registers and/or constants contained within a machine instruction or elsewhere

**Register Addressing**

In this mode the source operand, destination operand or both are to be contained in the 8086 register.

MOV DX, CX

MOV BL,DL

**Immediate Addressing**

Transfers the source-immediate byte or word of data into the destination register or memory location.

MOV CL, 03H

MOV DX, 0502H

**Direct Addressing**

Moves a byte or word between a memory location and a register. The instruction set does not support a memory-to-memory transfer, except for the MOVS instruction.

**Examples:**

MOV CX,START

MOV START,BL

START can be defined as an address by using the assembler DB(Define Byte) or DW (Define Word) pseudo instructions.

**Register Indirect Addressing**

Transfers a byte or word between a register and a memory location addressed by an index or base register. The index and base registers are BP, BX, DI, and SI.

**BX=FA09H**

**Example:**

MOV AX,[BX]

instruction copies the word-sized data from the data segment offset address indexed by BX into register AX.

**Load Effective Address:**

Computes the effective address of the second operand (the source operand) and stores it in the first operand (destination operand). The source operand is a memory address (offset part) specified with one of the processors addressing modes; the destination operand is a general- purpose register.

**Array:**

Arrays can be seen as chains of variables. A text string is an example of a byte array, each character is presented as an ASCII code value (0..255).

Here are some array definition examples:

Array1 DB 48h, 65h, 6Ch, 6Ch, 6Fh, 00h

Array2 DB 'Hello', 0

You can access the value of any element in array using square brackets, for example:

MOV AL, Array1[3]

You can also use any of the memory index registers BX, SI, DI, BP, for example:

MOV SI, 3

MOV AL, Array2[SI]

**Lab Tasks**

**Execute the following tasks CLO [1]**

**TASK 1:** Write a program to swap the contents of Ax and Bx registers using register addressing.

**TASK 2:** Declare two byte sized integer arrays num1 and num2 having 5 elements each, add them and store the result in a third array num3.

**TASK 3:** Write a program to swap the contents of two word sized arrays num1 and num2.

*For all programs, use single step and create report showing all the registers that are updated once each instruction is executed.*