

Experiment 2: Basic Instructions in Assembly Language.

Logical Instructions

Some of the commonly used logical instructions are:

The NOT command complements all the bits

MOV AX, 7030H; In binary AX = 0111 0000 0011 0000
NOT AX ; In binary AX = 1000 1111 1100 1111

The **AND** command performs bitwise logical and of the operands and stores the result in first operand.

MOV BL, 70H ; In binary BL= 0111 0000
MOV CL, 40H ; In binary CL= 0100 0000
AND CL,BL; In binary CL= 0100 0000

The **OR** command performs bitwise logical or of the operands and stores the result in first operand.

MOV BL, 70H ; In binary BL= 0111 0000
MOV CL, 40H ; In binary CL= 0100 0000
OR CL,BL; In binary CL =0111 0000

The **XOR** command performs bitwise logical xor of the operands and stores the result in first operand.

MOV BL, 70H ; In binary BL= 0111 0000
MOV CL, 40H ; In binary CL= 0100 0000
XOR CL,BL; In binary CL =0011 0000

The **TEST** command performs bitwise logical and of the operands but none of the operands are changed. The result is indicated in the flag register.

MOV BL, 70H ; In binary BL= 0111 0000
MOV CL, 42H ; In binary CL= 0100 0010
TEST CL,BL; In binary CL = 0100 0010

Assembly Language Program structure

The first line is generally the title. To indicate title line you must use the key word **TITLE** at the beginning. For example:

TITLE example 01

To indicate the size of code and data memory model is used. The syntax is :

.MODEL memory_model

memory model can be : SMALL, LARGE, MEDIUM, HUGE etc.

For example:

.MODEL SMALL

A programs data segment contains all the variable definitions and constants. To declare a data segment we use the directive ***.DATA***

The purpose of stack segment declaration is to set aside a block of memory to store the stack.

. STACK 100H

To indicate the code segment use the ***.CODE*** directive.

Inside the code segment the instructions are organized as procedures. The simplest procedure definition is

Name PROC

; body of procedure

Name ENDP

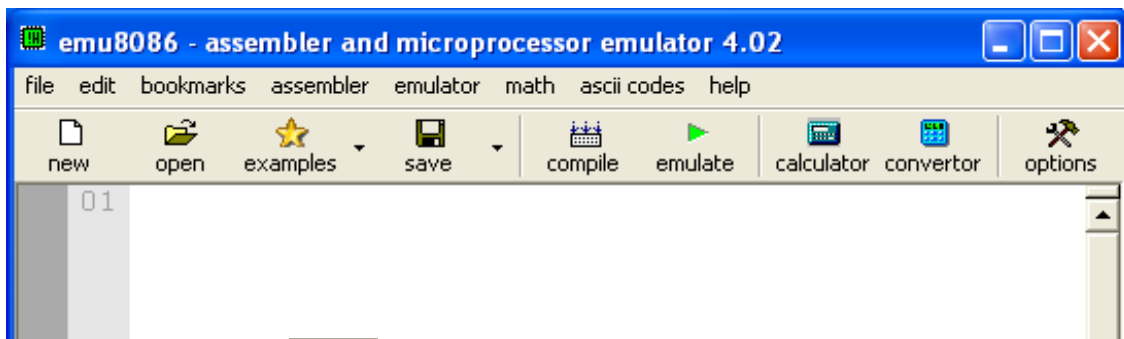
END MAIN

The last line should be the END directive followed by the name of main procedure.

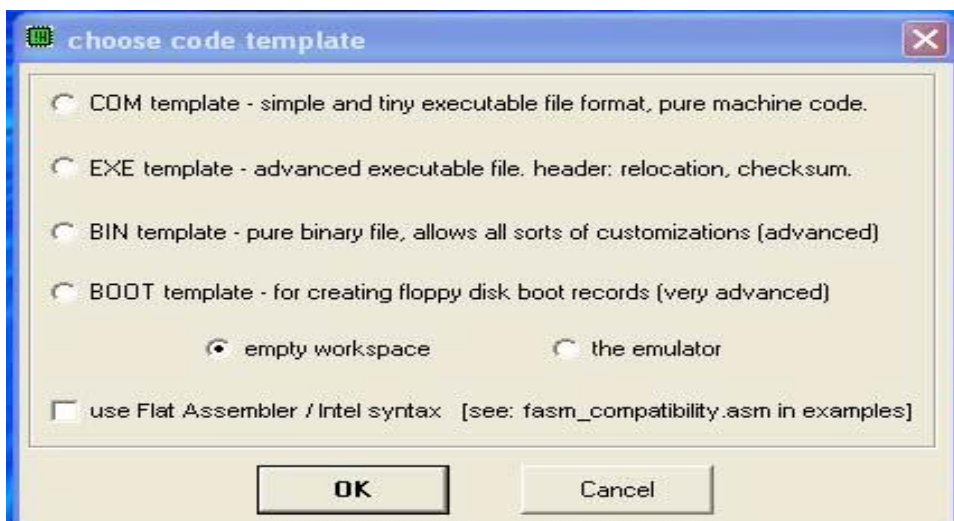
The 8086 Simulator:

In this LAB we will be using the *emu8086* assembler and microprocessor emulator.

Step 1: Open the emu8086 application. You'll see the following screen



Step 2: Click on the  button and select empty work space



Step 3: Write your code in the workspace.

