

MNI TASK

Question#1:

Compare Register Addressing mode, Immediate Addressing mode, Direct Addressing mode and Register Indirect Addressing mode with the help of an example and draw conclusion that which addressing mode is fastest?

Solution:

Comparison with the help of an example

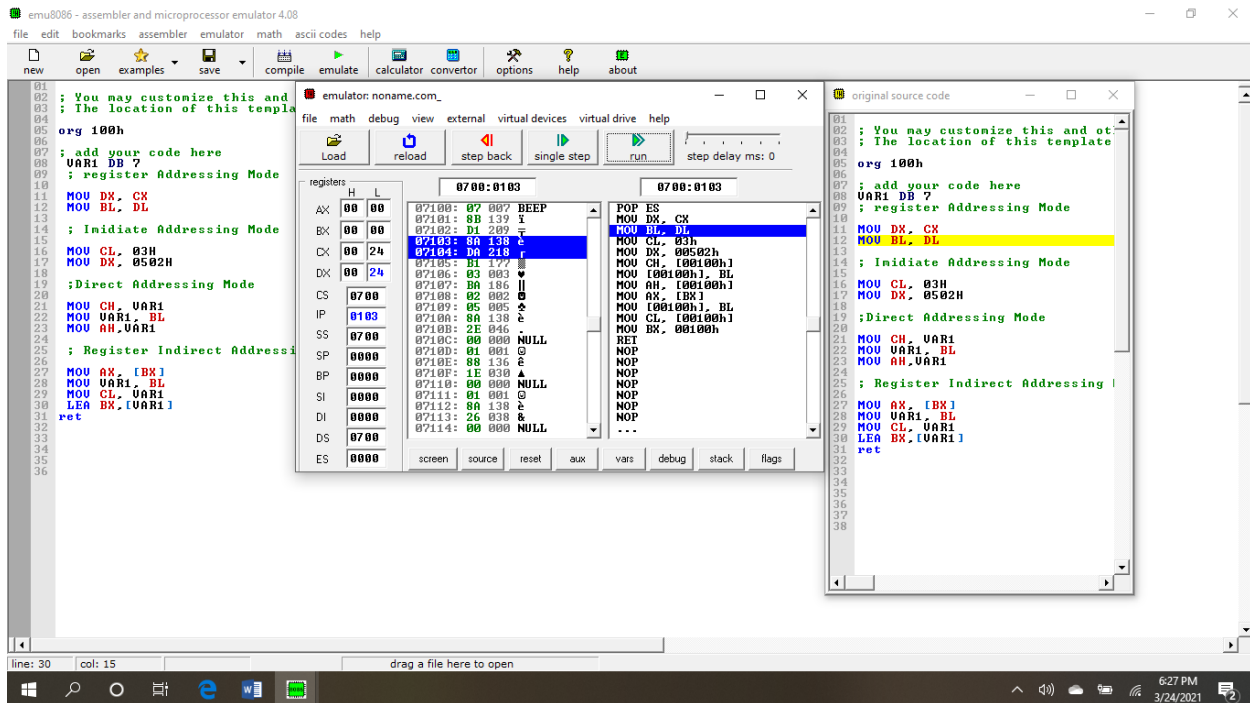
Let's compare these addressing modes with the help of an example in detail and draw conclusion that which addressing mode is the fastest addressing mode;

1) In case of Register addressing mode:

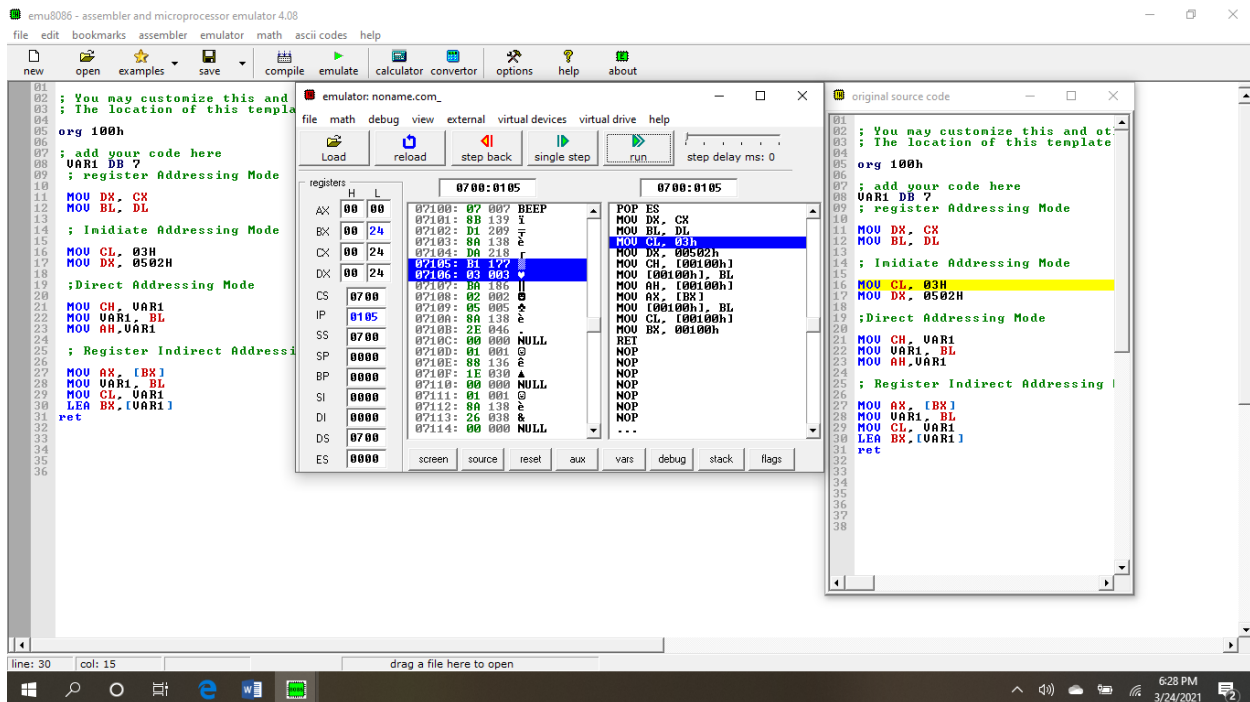
In case of register addressing mode cpu doesn't need to access the memory it can directly perform an operation through registers as in the following example we can see that:

Initially the CX register has value 24 in it will move in the DX register in a

Single step:



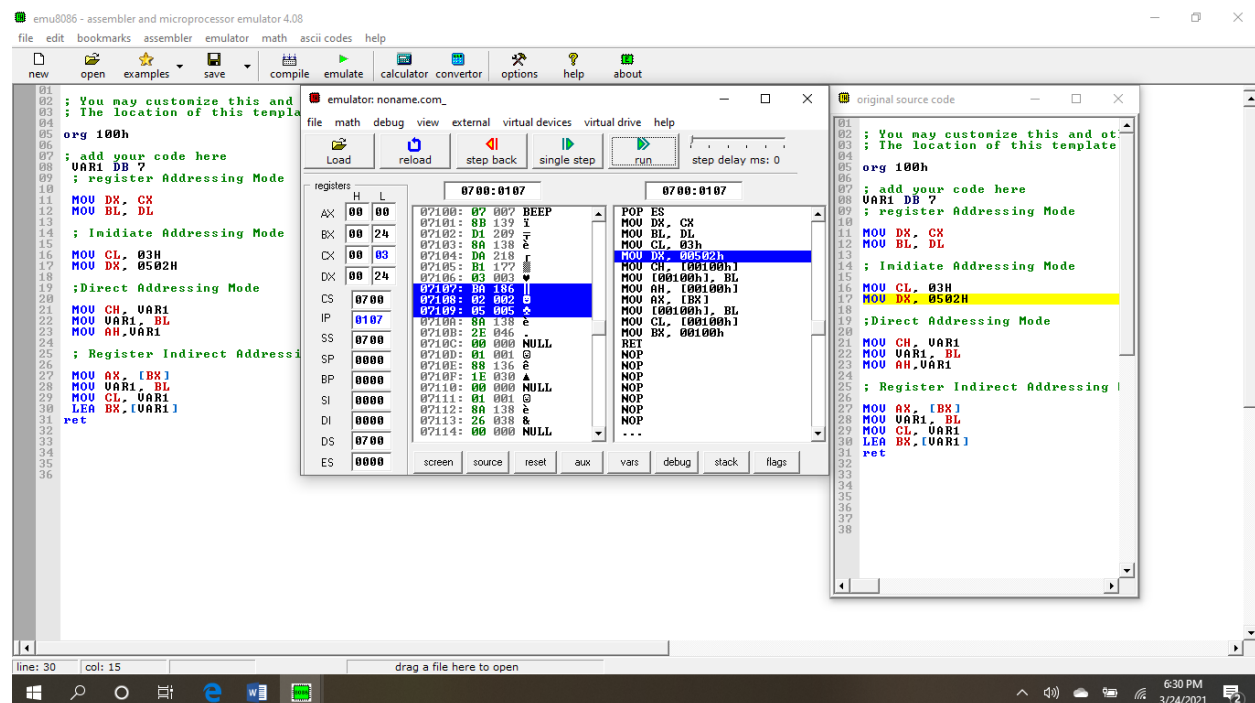
Now simply in this step the value contained by the lower part (DL) of DX register will move directly to the lower part (BL) of BX register as shown in following figure:



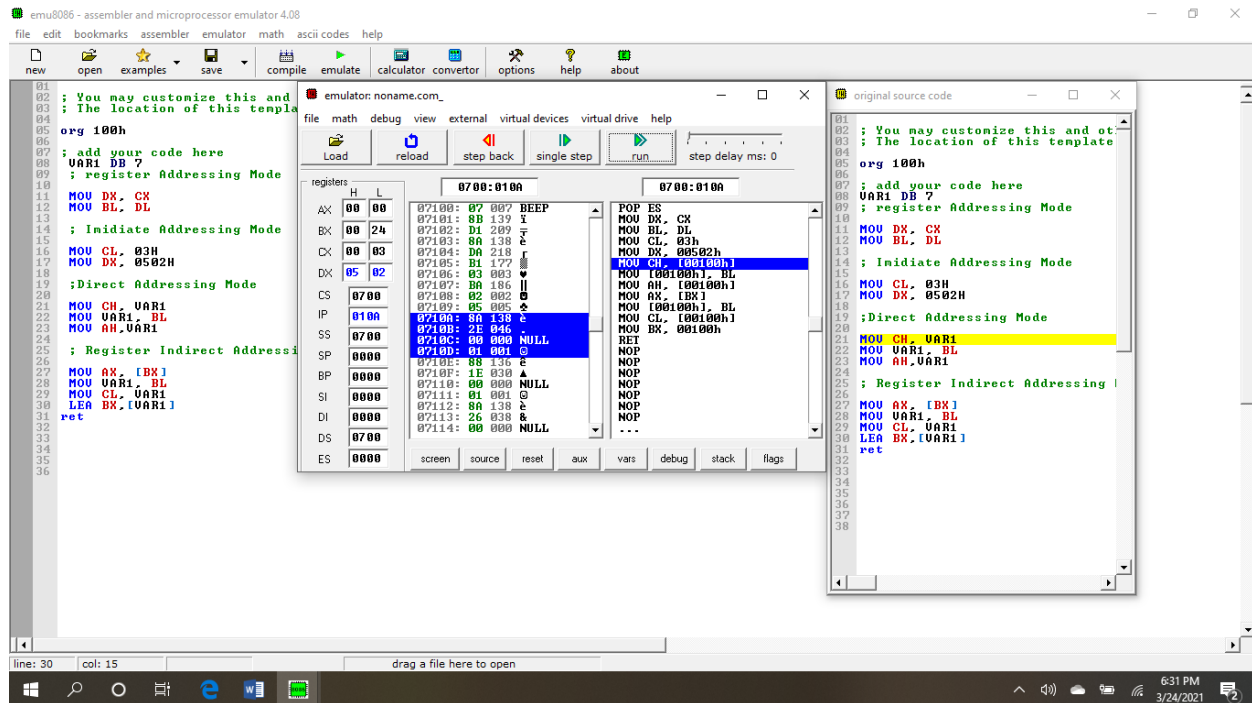
2) In case of immediate addressing mode:

In this mode, there are two operands. One is a register and the other is a constant value. The register comes quickly after the op code. Transfers the source-immediate byte or word of data into the destination register or memory location as shown in following example

In the first step the hexadecimal value 03H will simple load within the lower part of the CX register immediately as shown in following figure:

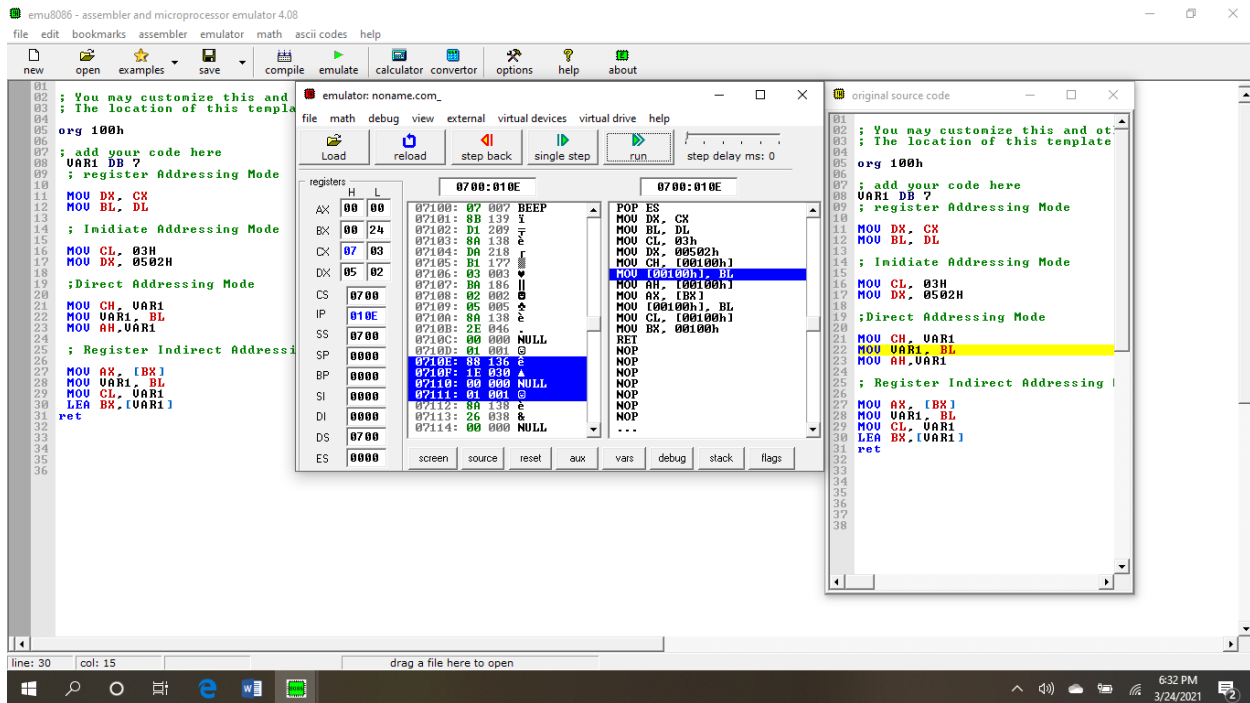


In this step of immediate addressing mode example the same thing will happen again , the hexadecimal value 0502H will move immediately within the DX register as shown in following figure:

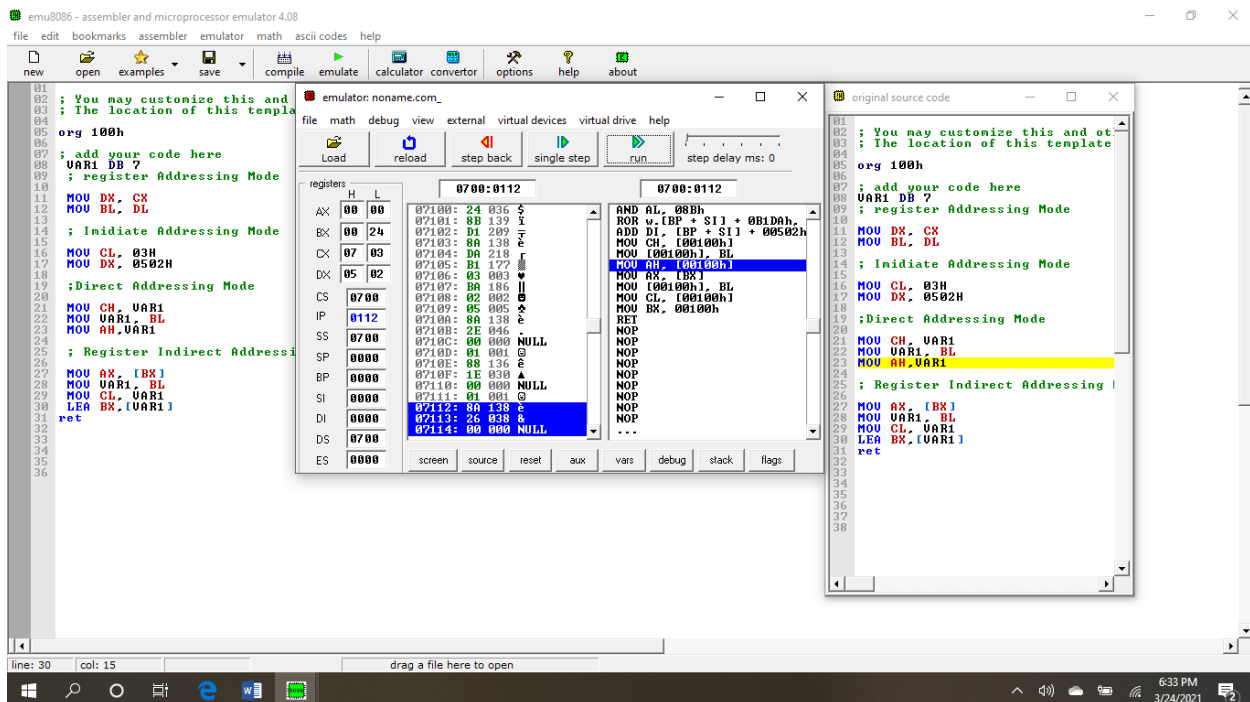


3) In case of direct addressing mode:

It loads or stores the data from memory to register and vice versa. The instruction consists of a register and an offset address. To compute physical address, shift left the DS register and add the offset address into it. . The instruction set does not support a memory-to-memory transfer, except for the MOVS instruction as shown in the following example: In the first step the I have to define the variable value with the help of command VAR DR 7 as I have inserted 7 data byte as shown in following figure now this VAR1 having 7 value this value will move in the register's CX higher part which is CH



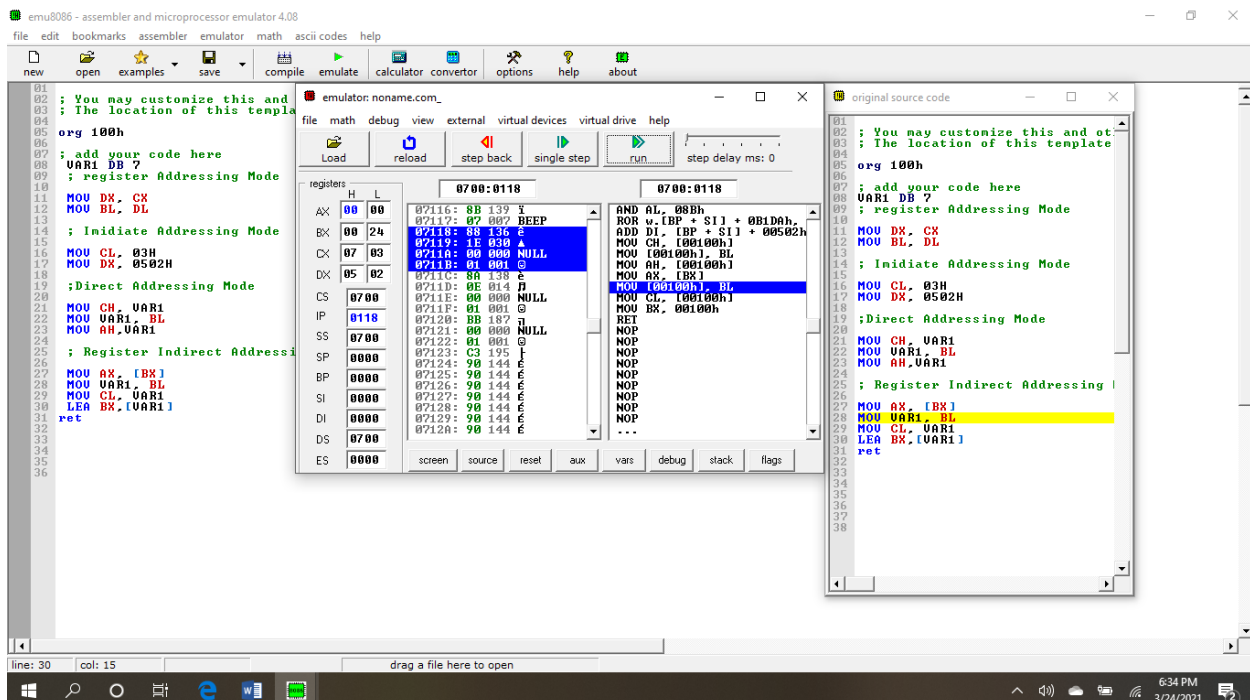
The variable value can also be updated with the already existing register value as in this step the value in BL part of BX register will be moved to the VAR1 as shown in following figure:



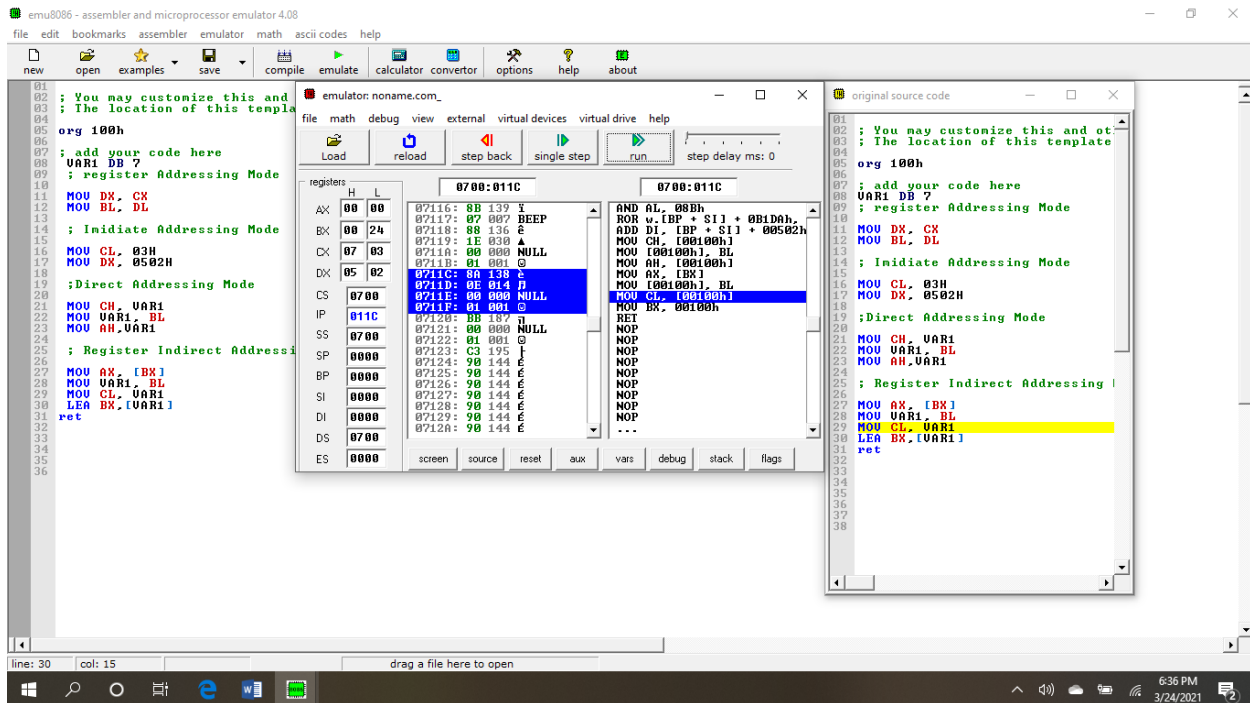
4) In case of Register Indirect Addressing:

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. It means that in this mode the memory location will be indirectly pointed with the help of register which contains the address of that memory location:

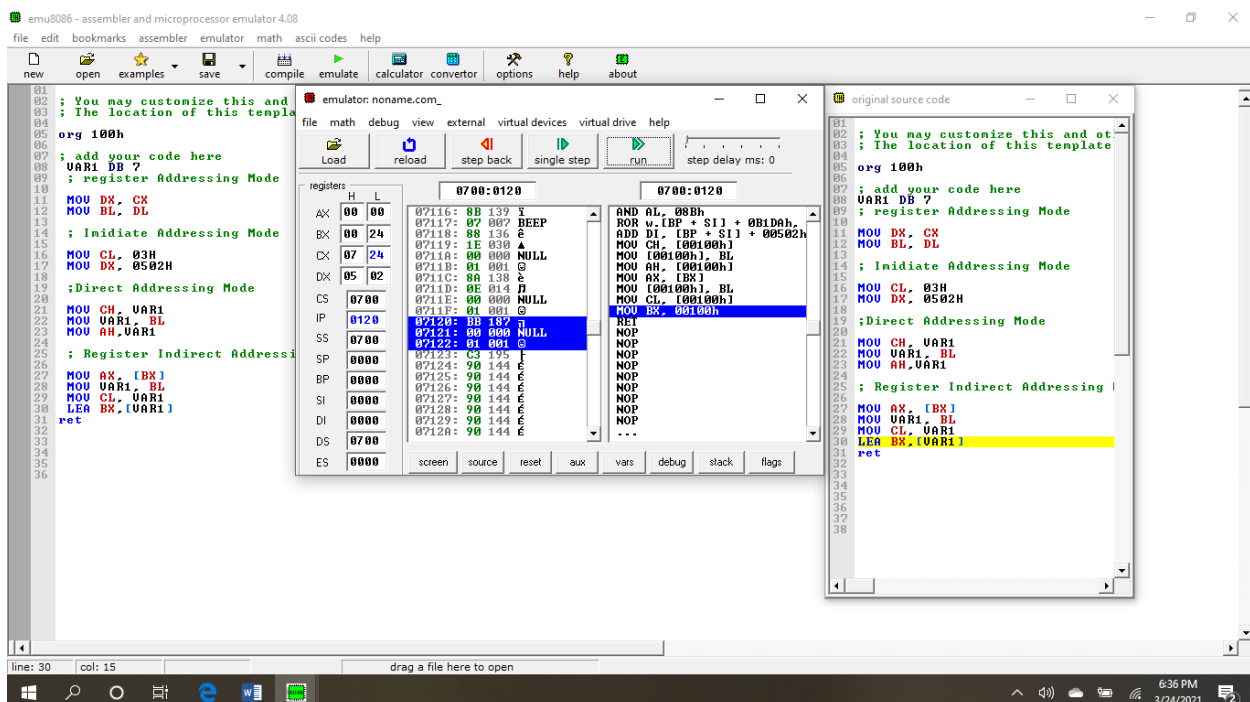
In this given example we simply want to describe the concept related to the indirect addressing mode as shown in the following figure in the first step the address of BX will be stored in the AX register:



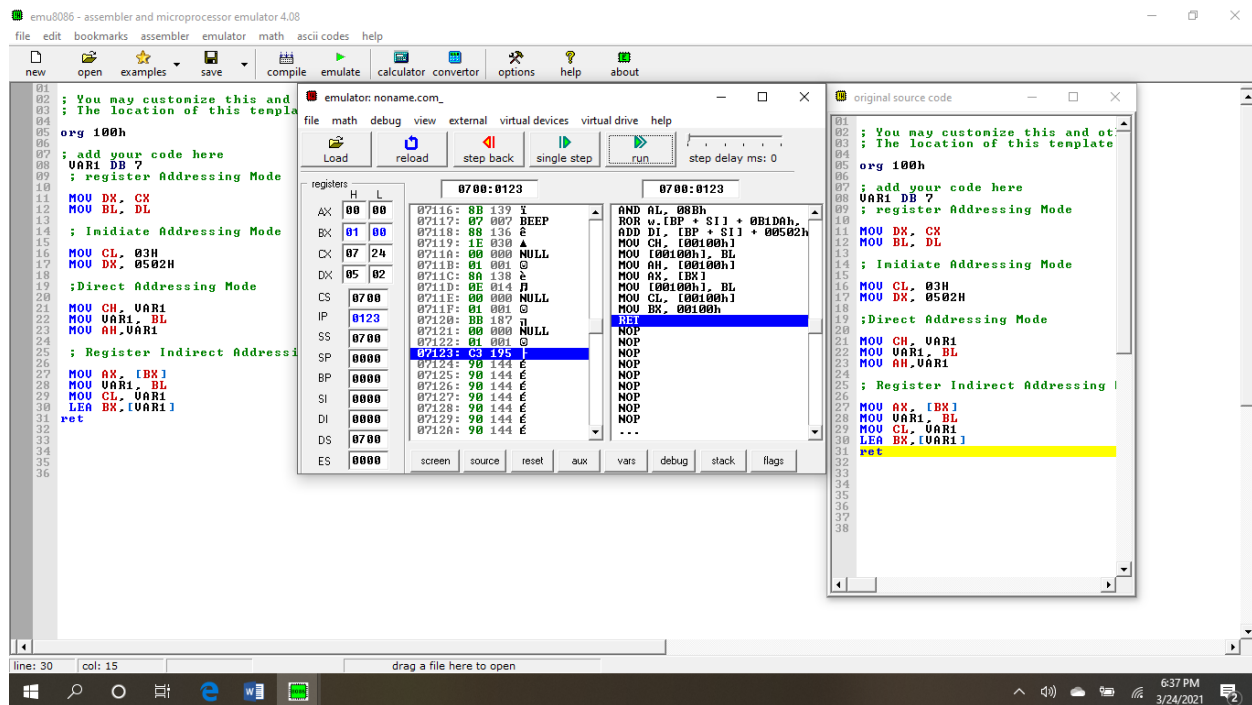
Now in the second step simply the VAR1 will be updated with the value of BL as shown in following figure:



Now in this step the CL register will be updated with the VAR1 value as shown in following figure:



In this step I have just used the command load effective address to load the memory address of VAR1 in the BX register as shown in following figure:



Conclusion:

From above comparison I concluded that the register addressing mode is fastest because the instructions are compact and fastest executing of all instruction forms. Registers may be used as source operands, destination operands or both. So the register addressing mode should be faster as register is the fastest memory location in the computer. However, the direct addressing mode should be fast but is always less fast than the register addressing mode if the memory data that you're accessing is already in the CPU's data cache.

-----THE END-----