Chapter Three

Addressing Modes of 8086

Introduction

- The different ways in which a processor can access data are called addressing modes.
- It is the way of specifying data to be operated by an instruction.
- It is the method to specify the operand of an instruction.
- The job of a microprocessor is to execute a set of instructions stored in memory to perform a specific task.
- Operations require the following:
    1. The operator or opcode which determines what will be done
    2. The operands which define the data to be used in the operation
- 8086 assembly language instructions can be used to illustrate the addressing modes.
- Format of **MOV** instruction:

```
MOV destination, source
```

Cont'd...

- When the 8086 executes an instruction, it performs the specified function on data.
- The data are called its **operands** and may be part of the instruction reside in one of the internal registers of the 8086, stored at an address in memory, or held at an I/O port.
- To access these different types of operands, the 8086 is provided with various addressing modes.

Source of data can be

- Immediate data
- A specified register
- A memory location

Destination of data can be

- A specified register
- A memory location

Cont'd...

- The addressing mode specifies a rule for interpreting or modifying the address field of the instruction before the operand is executed.
- Addressing modes for 8086 instructions are divided into two categories:
    1. Addressing modes for data
    2. Addressing modes for branch
- The 8086 memory addressing modes provide flexible access to memory, allowing you to easily access variables, arrays, records, pointers, and other complex data types.
- The key to good assembly language programming is the proper use of memory addressing modes.
- An assembly language program instruction consists of two parts.

Cont'd...

- The memory address of an operand consists of **two** components:
  - **Starting address** of memory segment.
  - **Effective address or Offset**: An offset is determined by adding any combination of three address elements: **displacement, base and index.**
    - **Displacement:** It is an 8-bit or 16-bit immediate value given in the instruction.
    - **Base**: Contents of base register, BX or BP.
    - **Index**: Content of index register SI or DI.

Cont'd...

Classification of 8086 Addressing Modes

- Immediate addressing mode
- Direct addressing mode
- Register addressing mode
- Register Indirect addressing mode
- Based addressing mode
- Indexed addressing mode
- Register relative addressing mode
- Based indexed addressing mode
- Relative based indexed addressing mode
- Implied Addressing Mode
- Port Addressing Mode

Immediate Addressing Modes

- In this type of addressing mode the source operand is an 8-bit or 16-bit data.
- Immediate data is a part of instruction.
- Destination operand can never be immediate data.

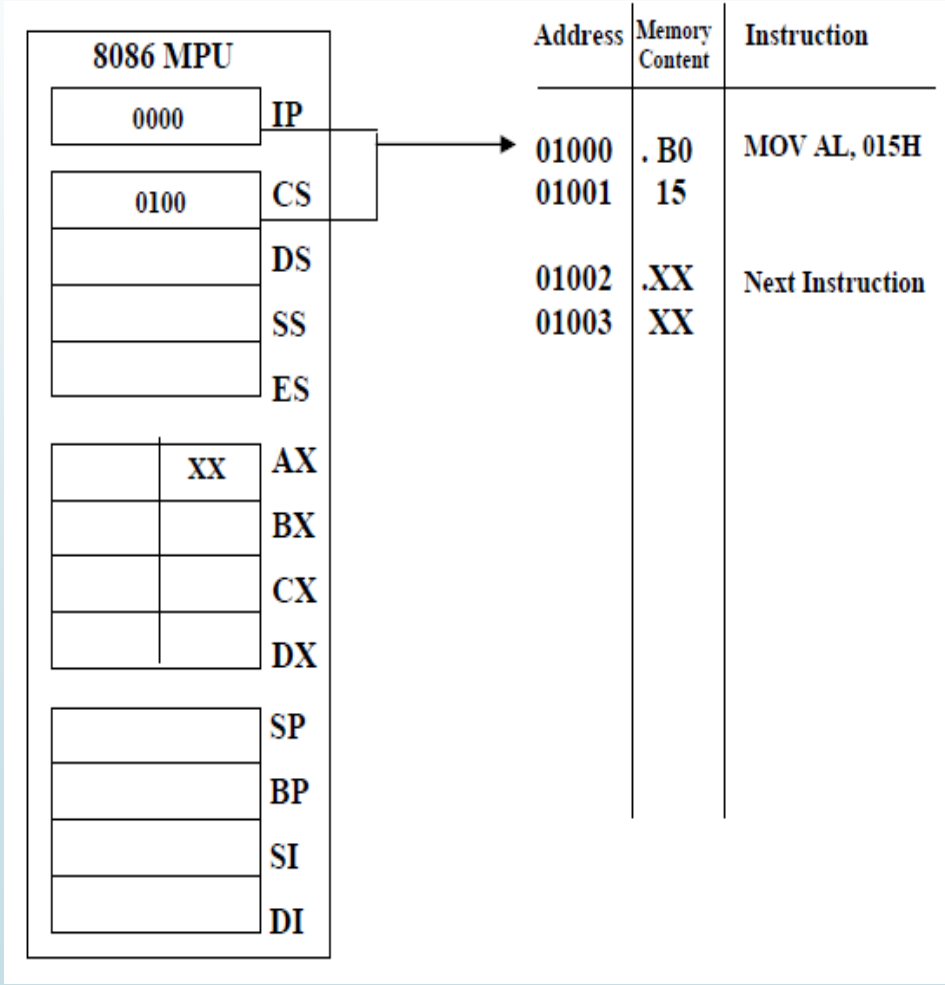
Example:

```
MOV AX, 2000H
MOV CL, 0AH
ADD AL, 45H
AND AX, 0000H
```

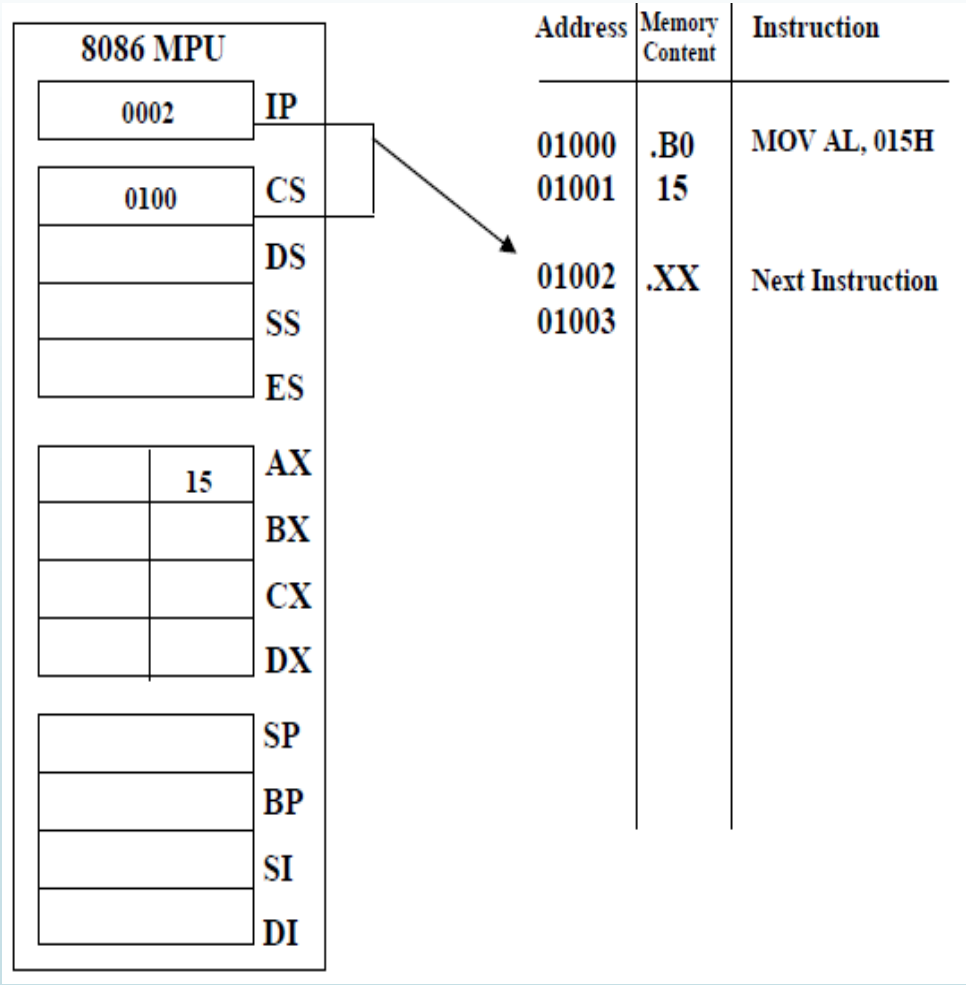
- In the above example, 2000H, 0AH, 45H, and 0000H are the immediate data.

Cont'd...

Immediate addressing mode before execution.



Immediate addressing mode after execution.



Direct Addressing Mode

- In this type of addressing mode the effective address is directly given in the instruction as displacement.
- Memory address (offset) is directly specified in the instruction.
- This addressing mode sometimes also called **displacement** addressing mode

Example:

```
MOV AX, [DISP]
MOV AX, [0500H]
MOV AX, [5000H]
```

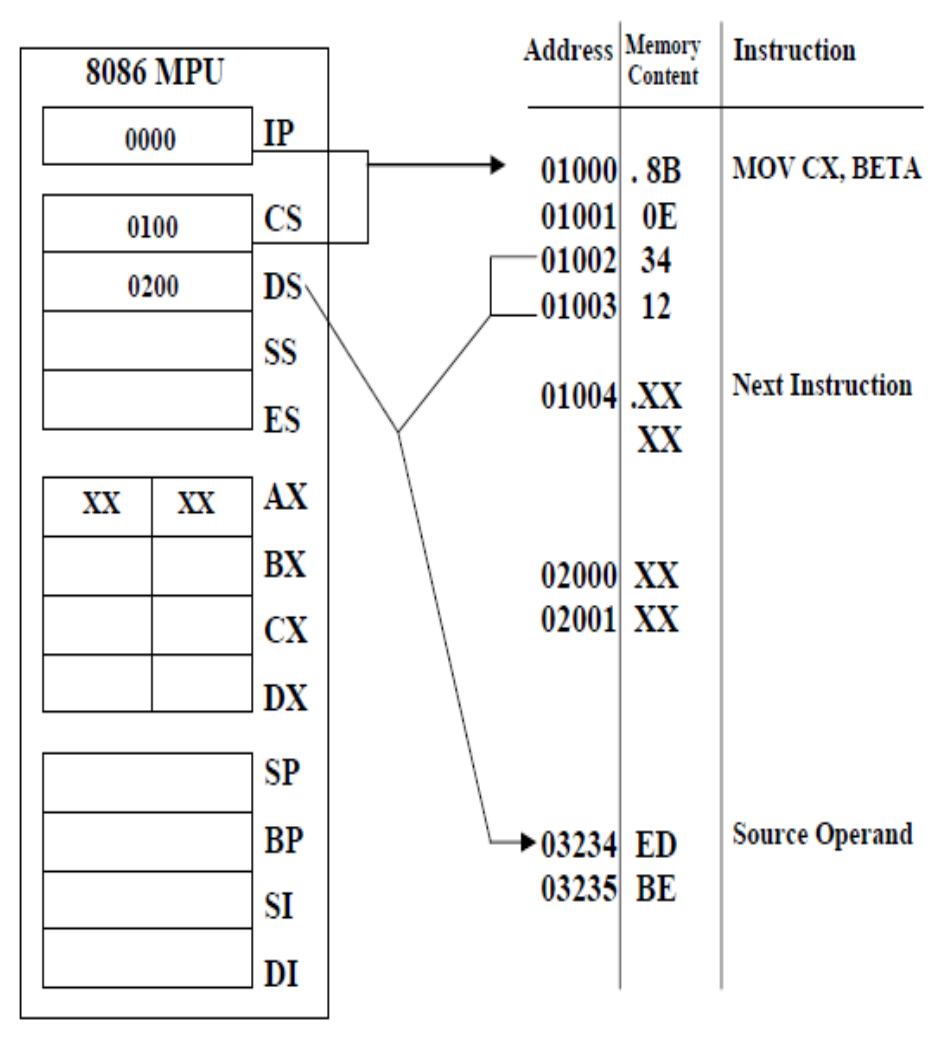
Example:
DS=0200H (Content of DS)
Offset=5000H

Physical Address (PA)
=
10H*DS + Offset=7000H

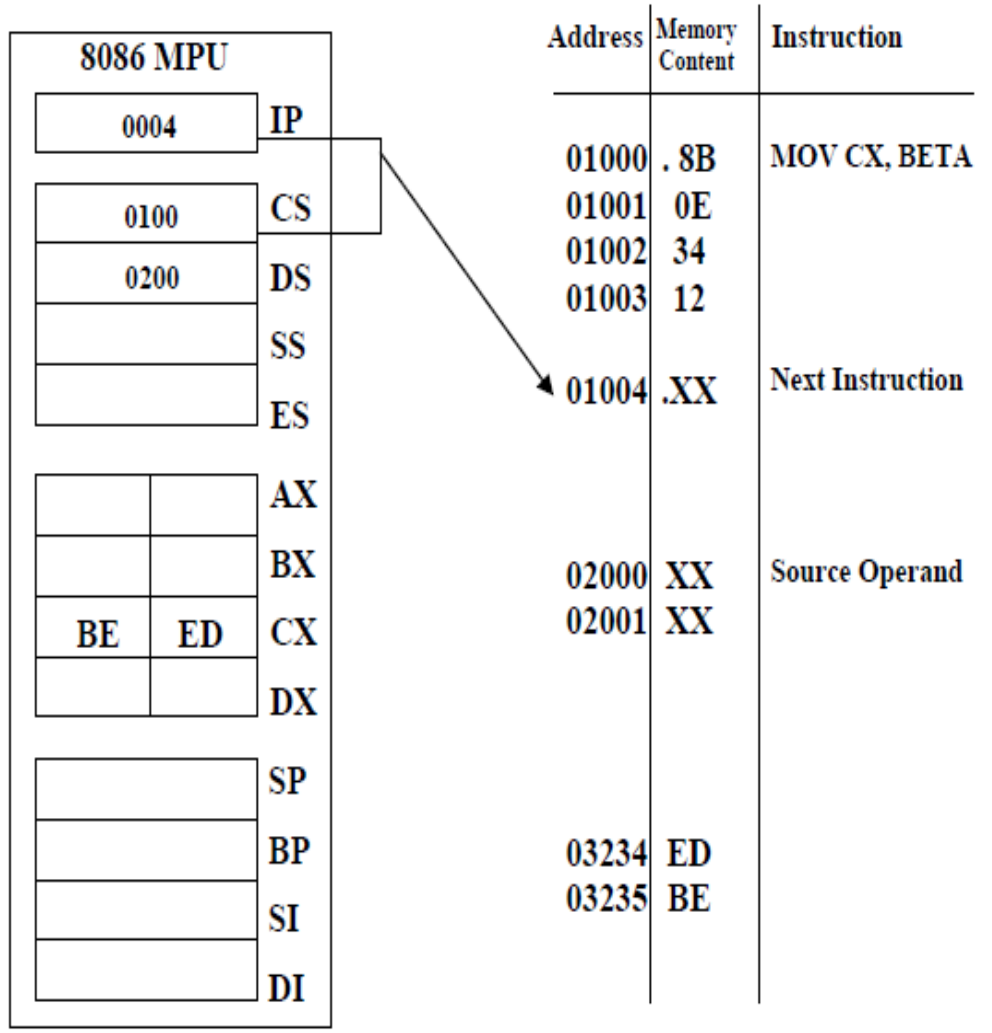
- Here, data resides in a memory location in the data segment.
- Physical address of memory location is calculated using DS and offset values DISP, 0500H and 5000H in each instructions above.

BETA=[1234H]

Direct addressing mode before execution.



Direct addressing mode after execution.



Register Addressing Mode

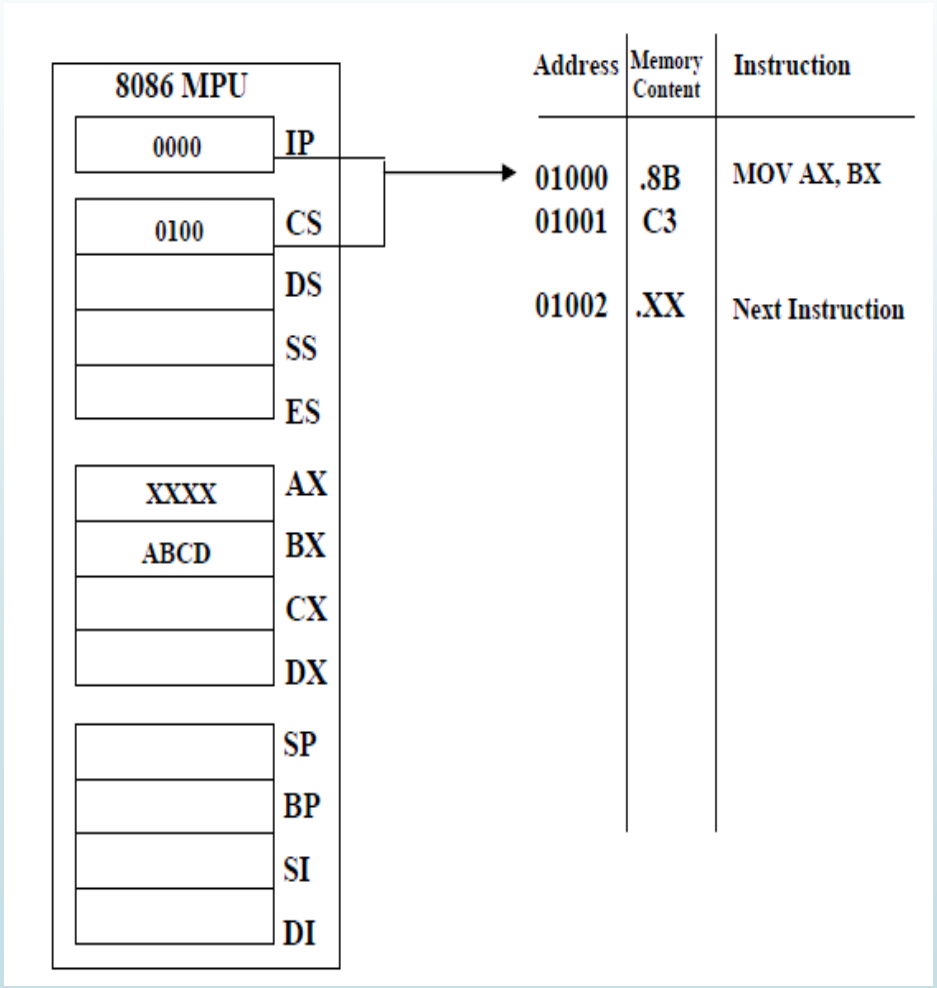
- With the register addressing mode, the operand to be accessed is specified as residing in an internal register of the 8086
  - In this type of addressing mode both the operands are registers.
  - The operand is placed in one of 8 bit or 16-bit general purpose registers.
  - The data is in the register that is specified by the instruction.
  - Source/destination can be one of the 8086 registers

Example:

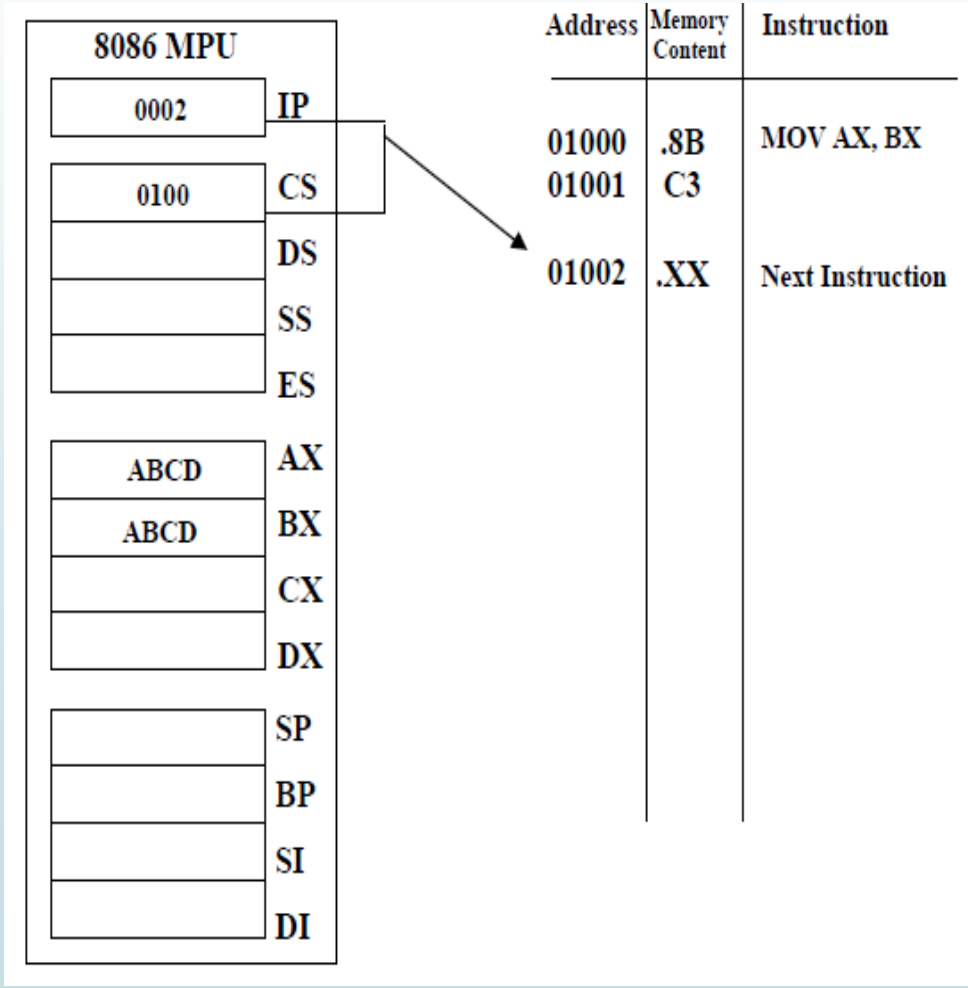
```
MOV AX, CX
MOV AX, BX
XOR AX, DX
ADD AL, BL
```

Cont'd...

Register addressing mode before execution.



Register addressing mode after execution.



Register Indirect Addressing Mode

- Register indirect addressing is similar to direct addressing in that an effective address is combined with the contents of DS to obtain a physical address.
- However, it differs in the way the offset is specified. This time effective address (EA) resides in either a pointer register, or index register within the 8086.
- The pointer register can be either **BX** or **BP** and the index register can be **SI** or **DI**.

Example1:

MOV AX, [BX].

- Here, Data is present in a memory location in DS whose offset address is in BX.

Cont'd…

- The default segment can be either DS Or ES.
- **Example 2**:

  MOV AX, [SI]

- This instruction moves the contents of the memory location offset by the value of EA in SI from the current value in DS to the AX register.
- **Assume**:

  SI contains 1234H and DS contains 0200H

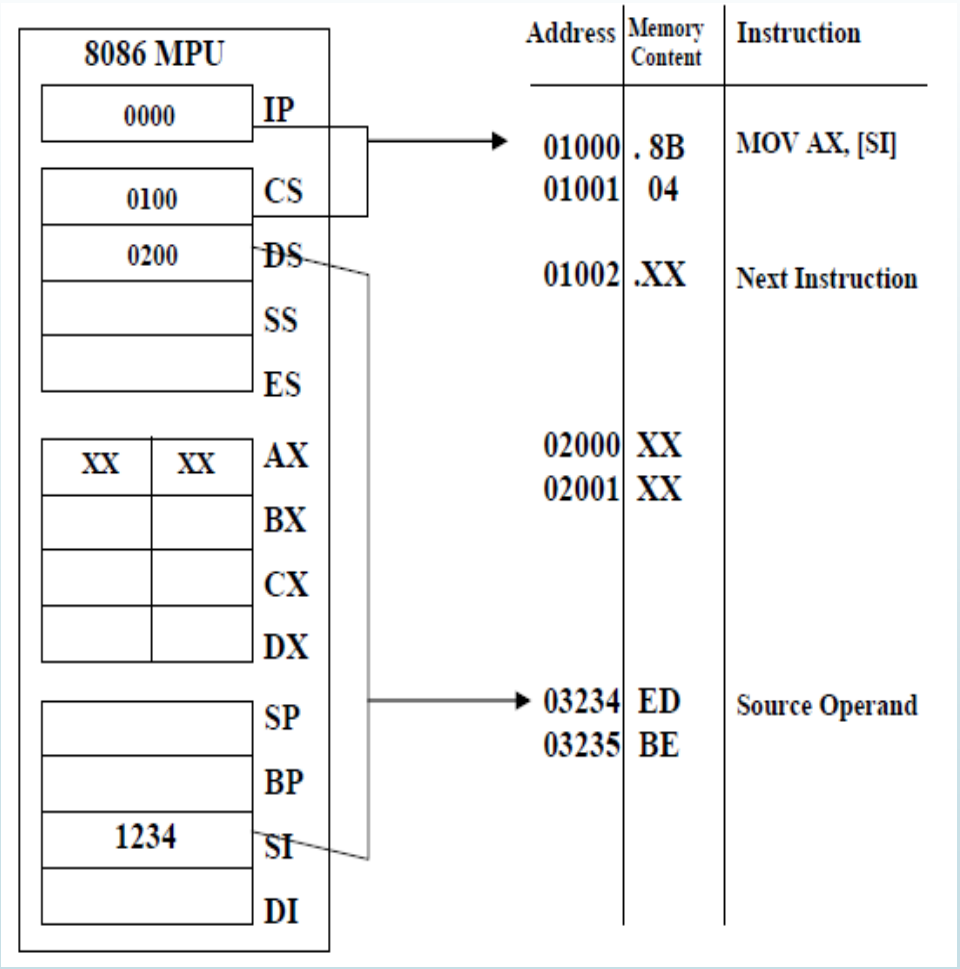
Physical Address (PA)=10H*DS + [SI]

PA=10H*0200H + 1234H

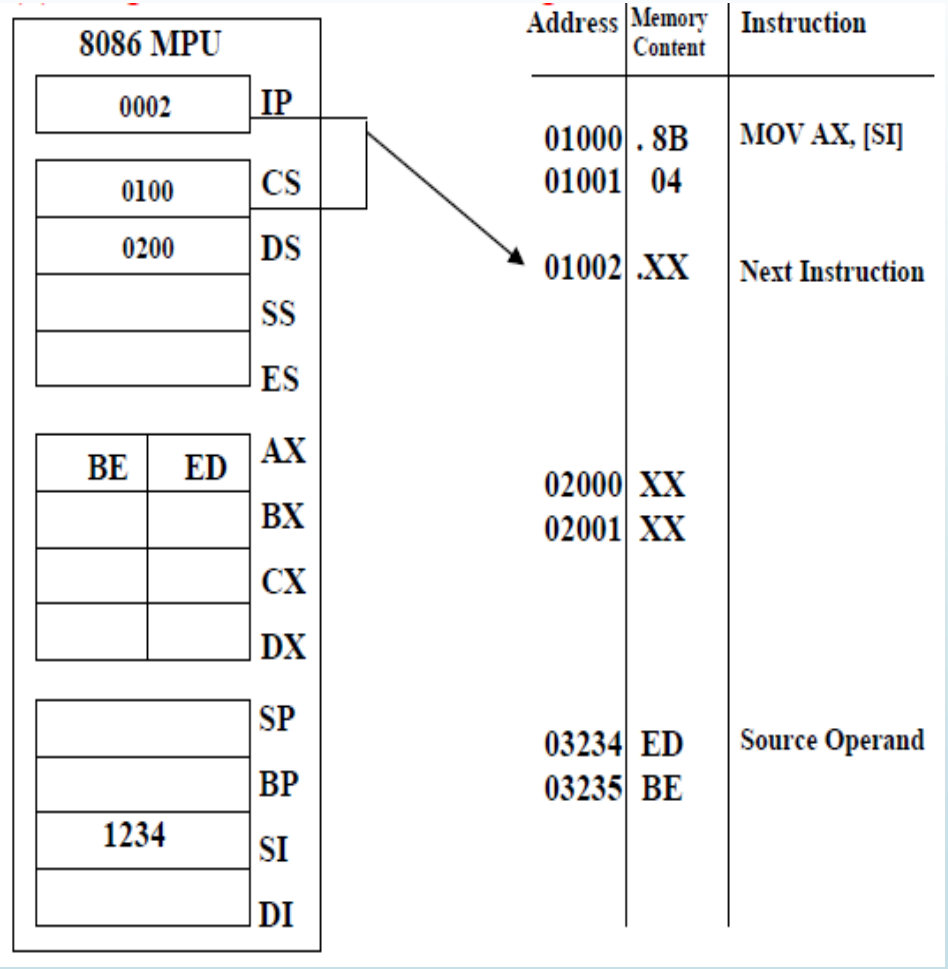
PA=3234H

Cont'd...

Register Indirect addressing mode before execution.



Register Indirect addressing mode after execution.



Based Addressing Mode

- In the based addressing mode, the physical address of the operand is obtained by adding a direct or indirect displacement to the contents of either BX or BP and the current value in DS and SS respectively.
- The offset address of the operand is given by the sum of contents of the BX/BP registers and 8-bit/16-bit displacement.
- The physical memory address is calculated according to the base register.

  Example: MOV AX, [BX + 1234H]

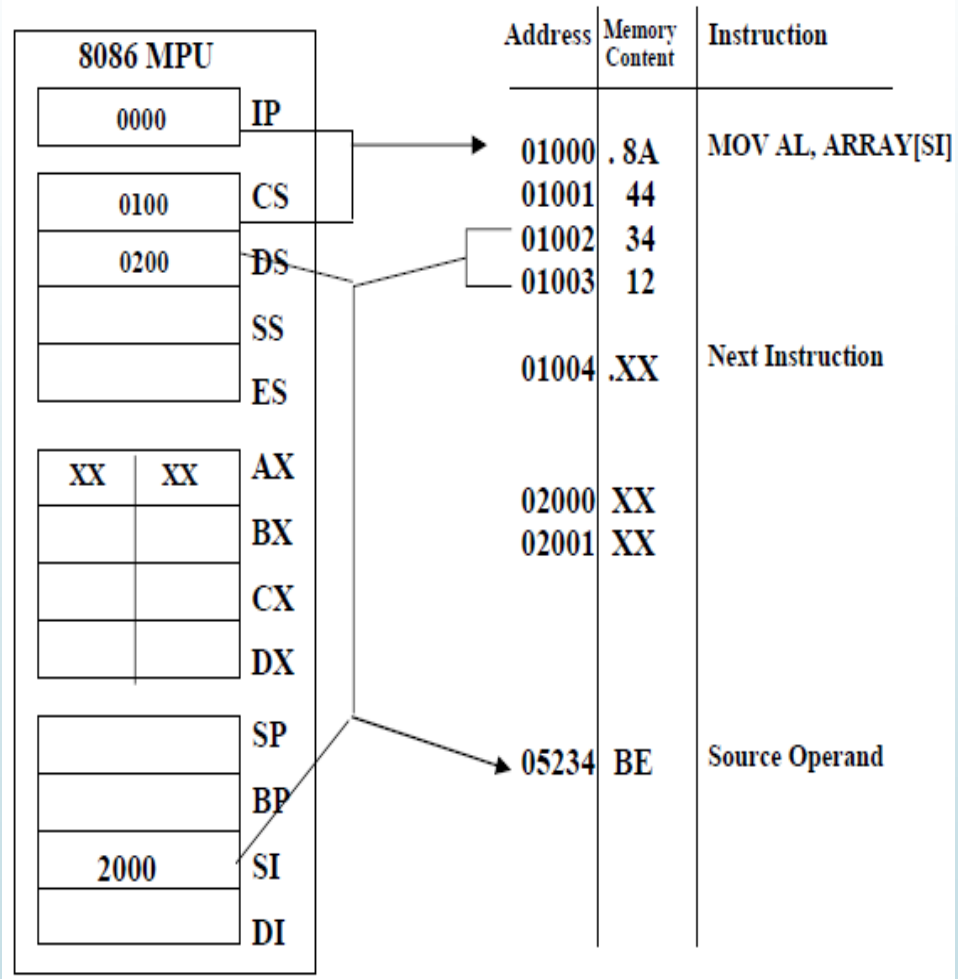
- The physical address (PA) is computed as: 10H*DS+[BX or BP]+Displacement.
  Assume:
  - DS=0200H, BX=1000H and DISP=1234H
  - PA=10H*0200H + 1000H + 1234H
  - PA=04234H

Indexed Addressing Mode

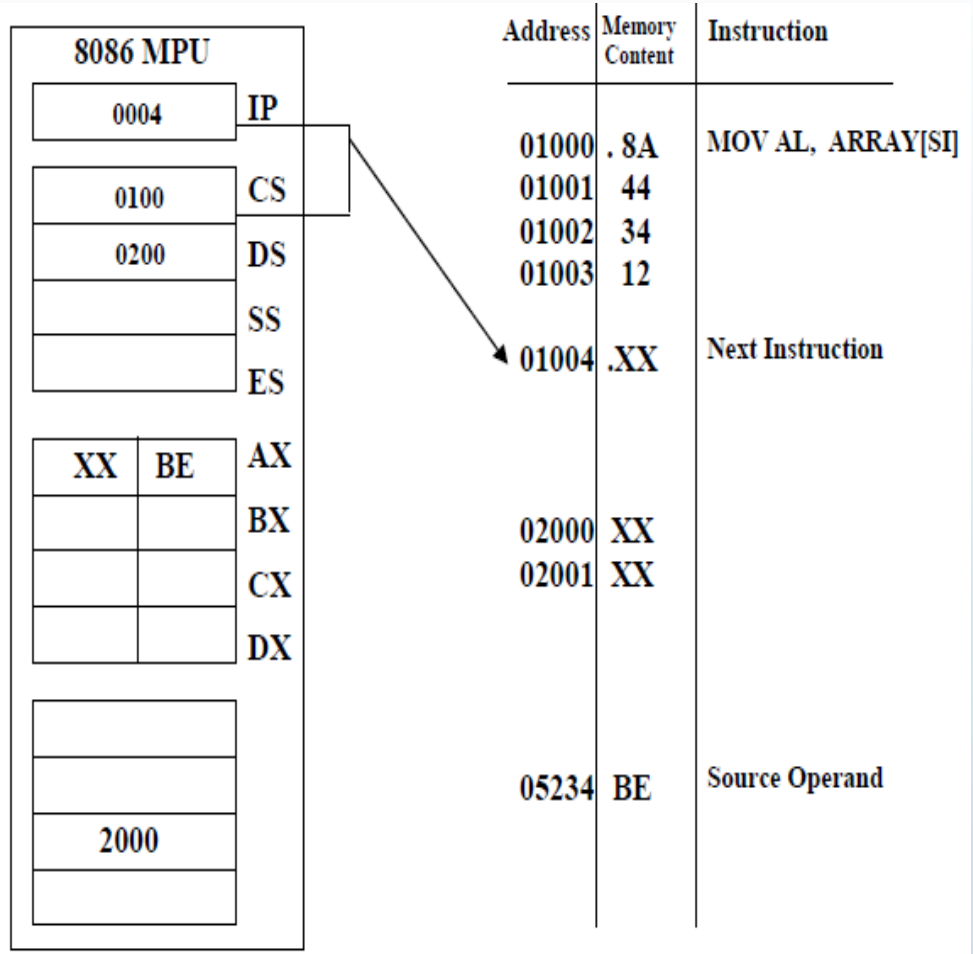
- Indexed addressing works identically to the based addressing, it uses the contents of one of the index registers, instead of BX or BP, in the generation of the physical address.
- The operands offset address is found by adding the contents of SI or DI register and 8-bit/16-bit displacements.
- In this type of addressing mode the effective address is sum of index register and displacement.
- Example:
  - MOV AX, [SI+2000]
  - MOV AL, [DI+3000]
- The physical address for the source operand is calculated from DS, SI, and the direct displacement. (PA=10H*DS + [SI or DI] + DISP
- Assume DS=0200H, SI=1234H, DISP=2000H
  - PA=10H*0200H + 1234H + 2000H= **05234H**

ARRAY=[1234H]

Indexed addressing mode before execution.



Indexed addressing mode after execution.



Based Indexed Addressing Mode

- In the based indexed addressing mode, the physical address of the operand is obtained by adding a direct or indirect displacement to the contents of either BX or BP and the current value in DS and SS respectively.
- The offset address of the operand is computed by summing the Base register to the contents of an Index register.
- In this the effective address is sum of base register and index register.

  Base register: BX, BP

  Index register: SI, DI

- The physical memory address is calculated according to the base register.

Cont'd…

Example:

- MOV AX, [BX + SI].
- MOV AL, [BP + SI]
- MOV AX, [BX + DI]
- The physical address (PA) is computed as: 10H*DS+[BX or BP]+[SI or DI].

  Assume:

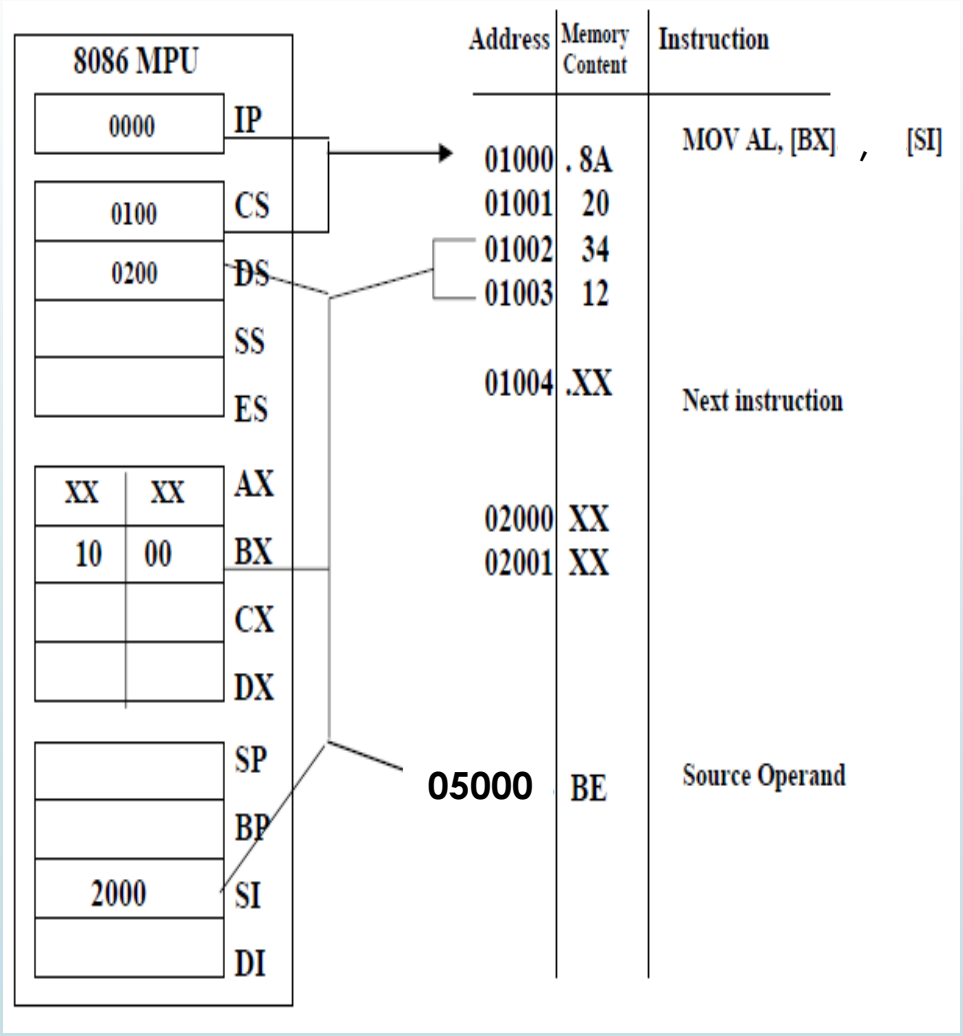
  DS=0200H, BX=1000H and SI=1234H

  PA=10H*0200H + 1000H + 1234H

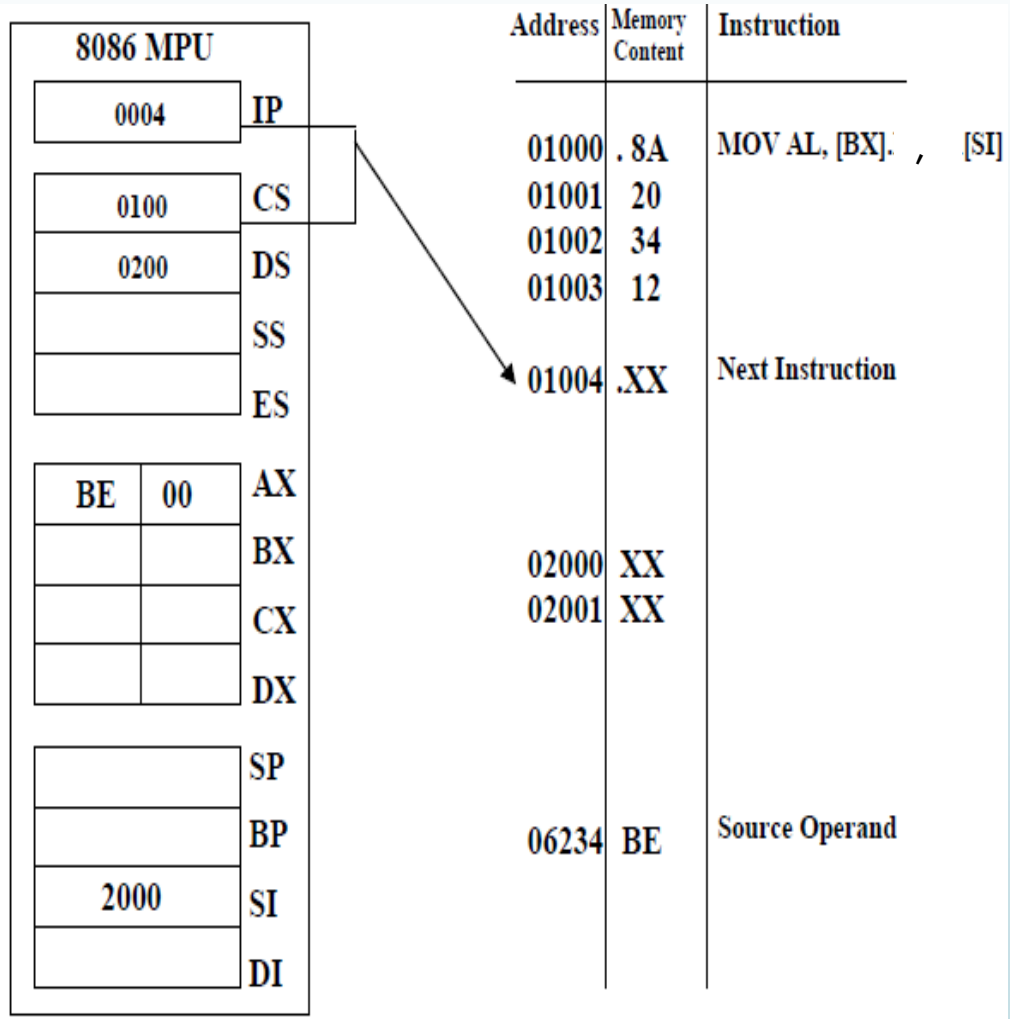
  PA=04234H

Cont'd...

Based Indexed addressing mode before execution.



Based Indexed addressing mode after execution.



Register Relative Addressing Mode

- In Register relative addressing mode, data is available at an effective address formed by adding an 8 bit or 16-bit displacement with content, any one of the registers BX, BP, SI, DI in the default (DS or ES) segment.

Example:

MOV AX, 50H [BX].

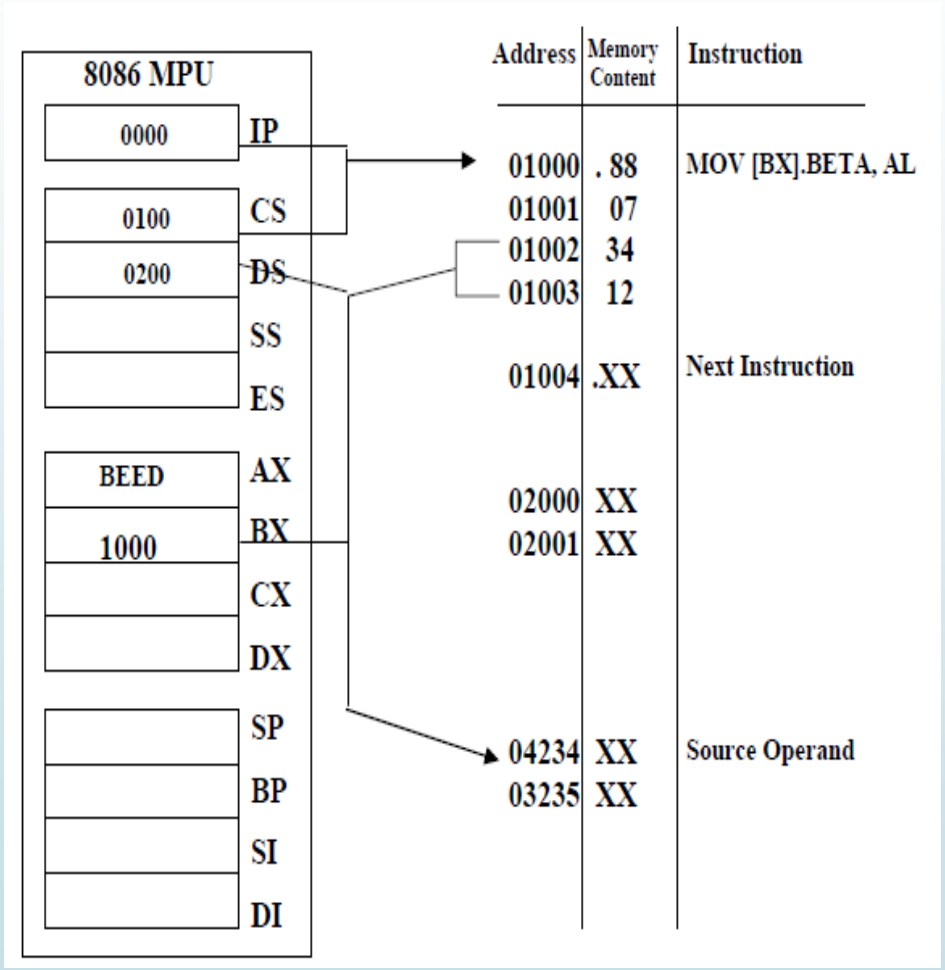
- Here, physical address is given as: **10H*DS+50H+[BX]**
- Assume DS=0200H, BX=1234H;

PA=10H*0200H + 50H + 1234H

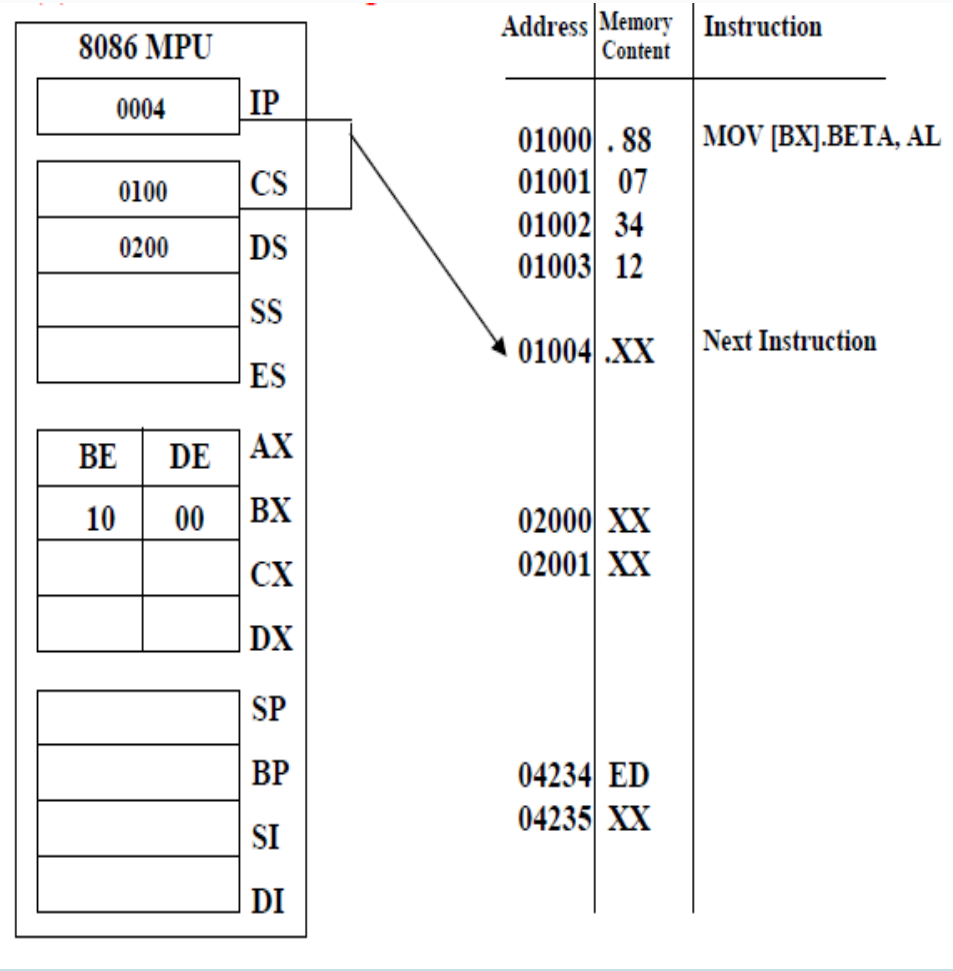
PA=3284H

BETA=[12345H]

Register Relative addressing mode before execution.



Register Relative addressing mode after execution.



Relative Based Indexed Addressing Mode

- Combining the based addressing mode and the indexed addressing mode together results in a new, more powerful mode known as relative based indexed addressing.
- Here the effective address is formed by adding an 8 bit or 16-bit **displacement** with the sum of the content of any one of the index registers in the default segment.

Example:

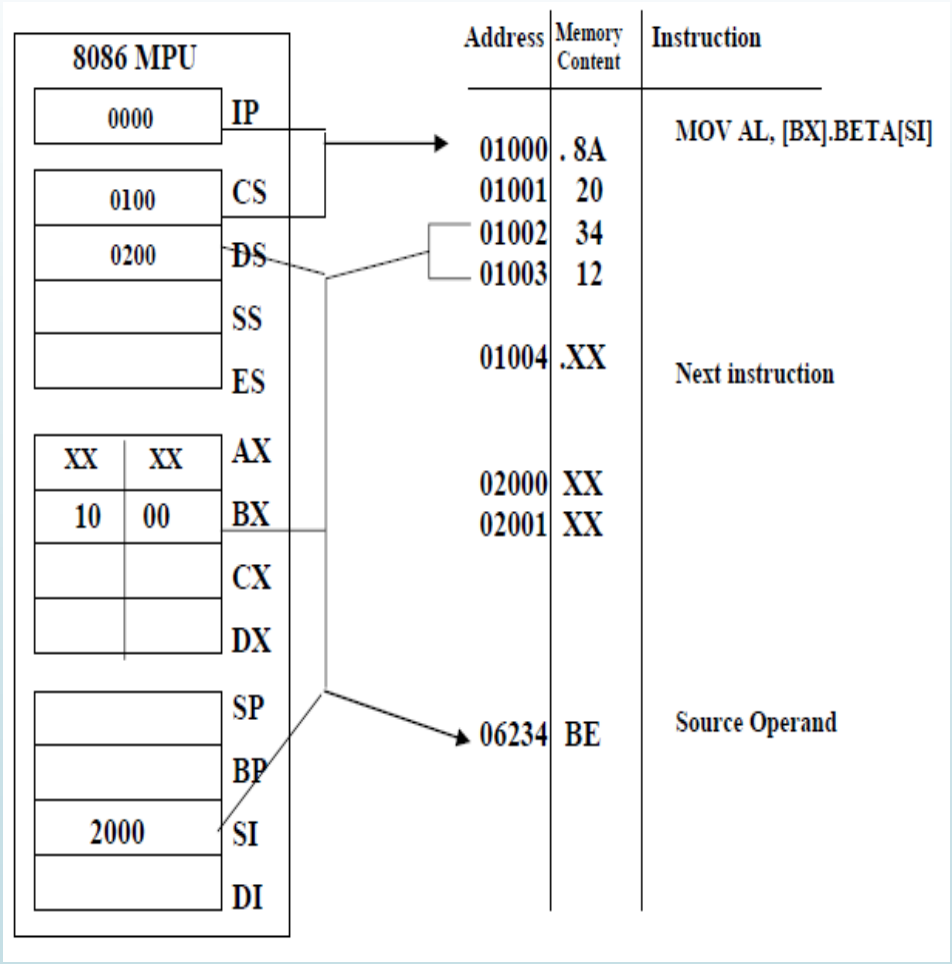
MOV AX, [BX] 50H [SI]

- The physical address of data is computed as:

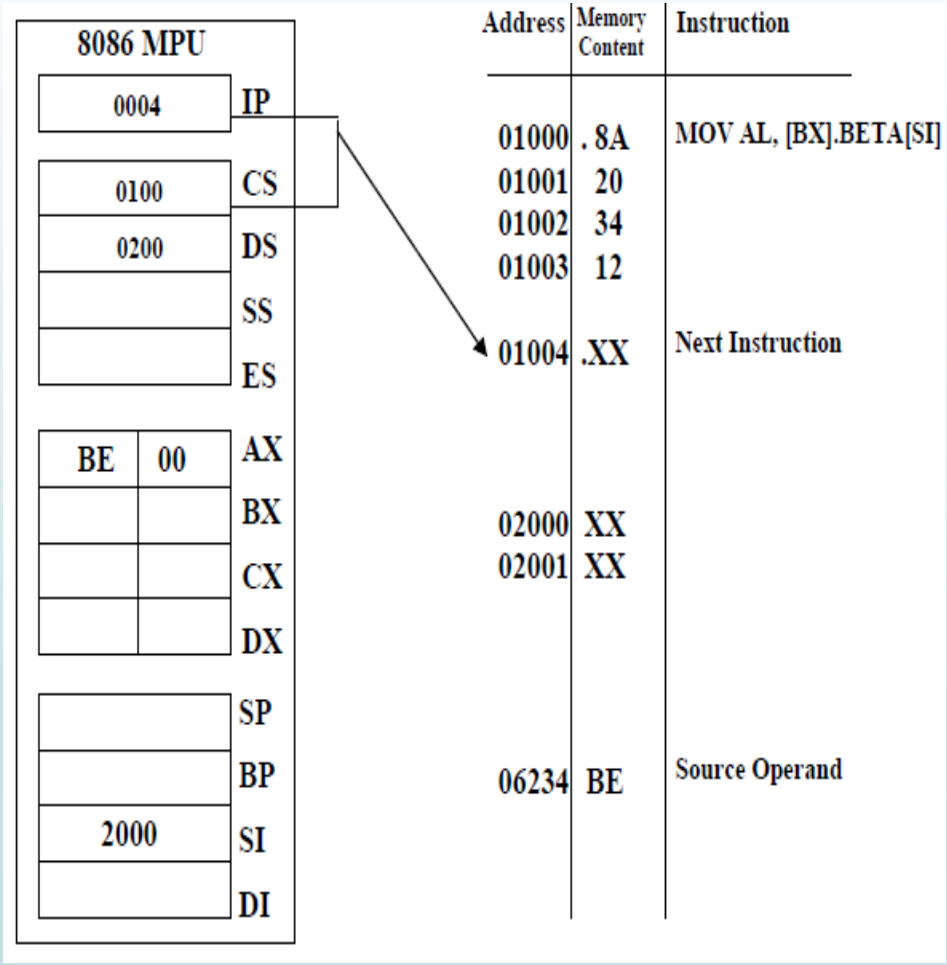
10H*DS+[BX] + [SI] + 50H.

BETA=[1234H]

Relative Based Indexed addressing mode before execution.



Relative Based Indexed addressing mode after execution.



Implied Addressing mode

- This type of addressing mode is also known as the implicit addressing mode (in the X86 assembly language).
- The implied addressing mode does not specify an effective address explicitly for either the destination or the source, or both sometimes.
- The opcode implies either the source or destination effective addresses or both sometimes.
- In implied addressing the operand is specified in the instruction itself.
- In this mode the data is 8 bits or 16 bits long and data is the part of instruction.
- In the implied mode, the operands are specified implicitly in the instruction definition.

Example:

```
STC   (used to set Carry flag to 1)
CLC      (used to reset Carry flag to 0)
STD   (used to set Direction flag to 1)
CLD   (used to reset Direction flag to 0)
```

Port Addressing Mode

- Port addressing is used in conjunction with the IN and OUT instructions to access input and output ports. Any of the memory addressing modes can be used for the port address for memory mapped ports.
- For ports in the I/O address space, only the **Direct addressing** mode and an **Indirect addressing mode** using DX are available.
- For example, **Direct addressing** of an input port is used in the instruction

  IN AL, 15H

- This stands for input the data from the byte wide input port at address 15H of the I/O address space to register AL.

Cont'd…

- Next, let us consider another example. Using Indirect port addressing for the source operand in an IN instruction, we get:

  IN AL, DX

- It means input the data from the byte wide input port whose address is specified by the contents of register DX.
- For instance, if DX equals 1234H the contents of the port at this I/O address are loaded into AL.

~~~~~~~~~~~ END~~~~~~~~~~
~~~~~~~~~~~