Addressing modes:

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 as follow:

1. Immediate addressing mode.
2. Register addressing mode.
3. Direct addressing mode.
4. Register indirect addressing mode.
5. Based addressing mode.
6. Indexed addressing mode.
7. Based indexed addressing mode
8. String addressing mode.
9. Input / Output mode.

Addressing modes:

1. **Immediate addressing mode:** In this type of addressing, immediate data is a part of instruction instead of the contents of a register or memory location, and it may be 8-bit or 16-bit in size.

- Ex: MOV AL, 015H

2. **Register addressing mode:** In register addressing mode, the data is stored in a register and is referred using the particular register. All the registers, except IP, may be used in this mode.

- Ex: MOV AX, BX

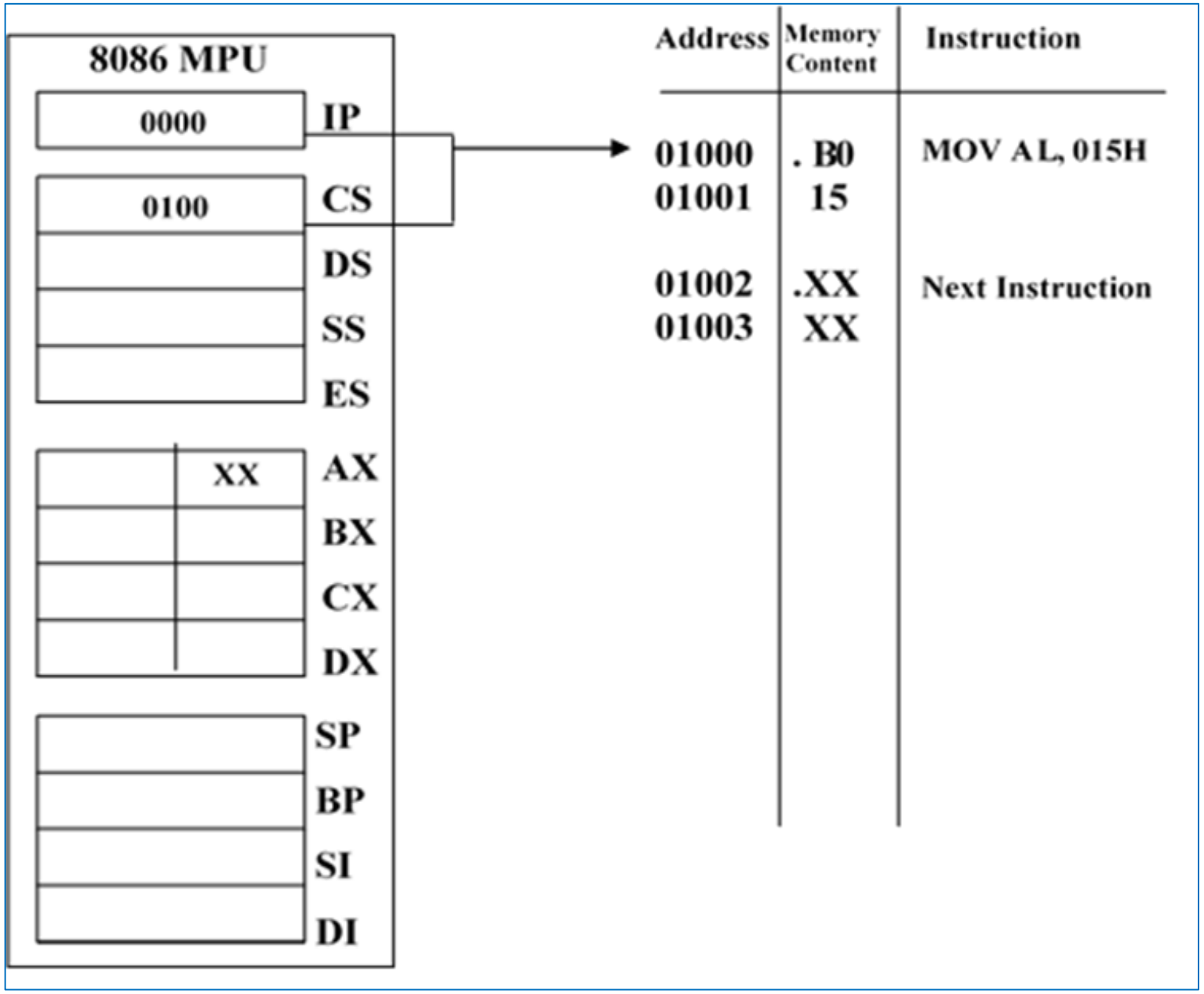

Fig. 8 (a): Immediate addressing mode before execution.

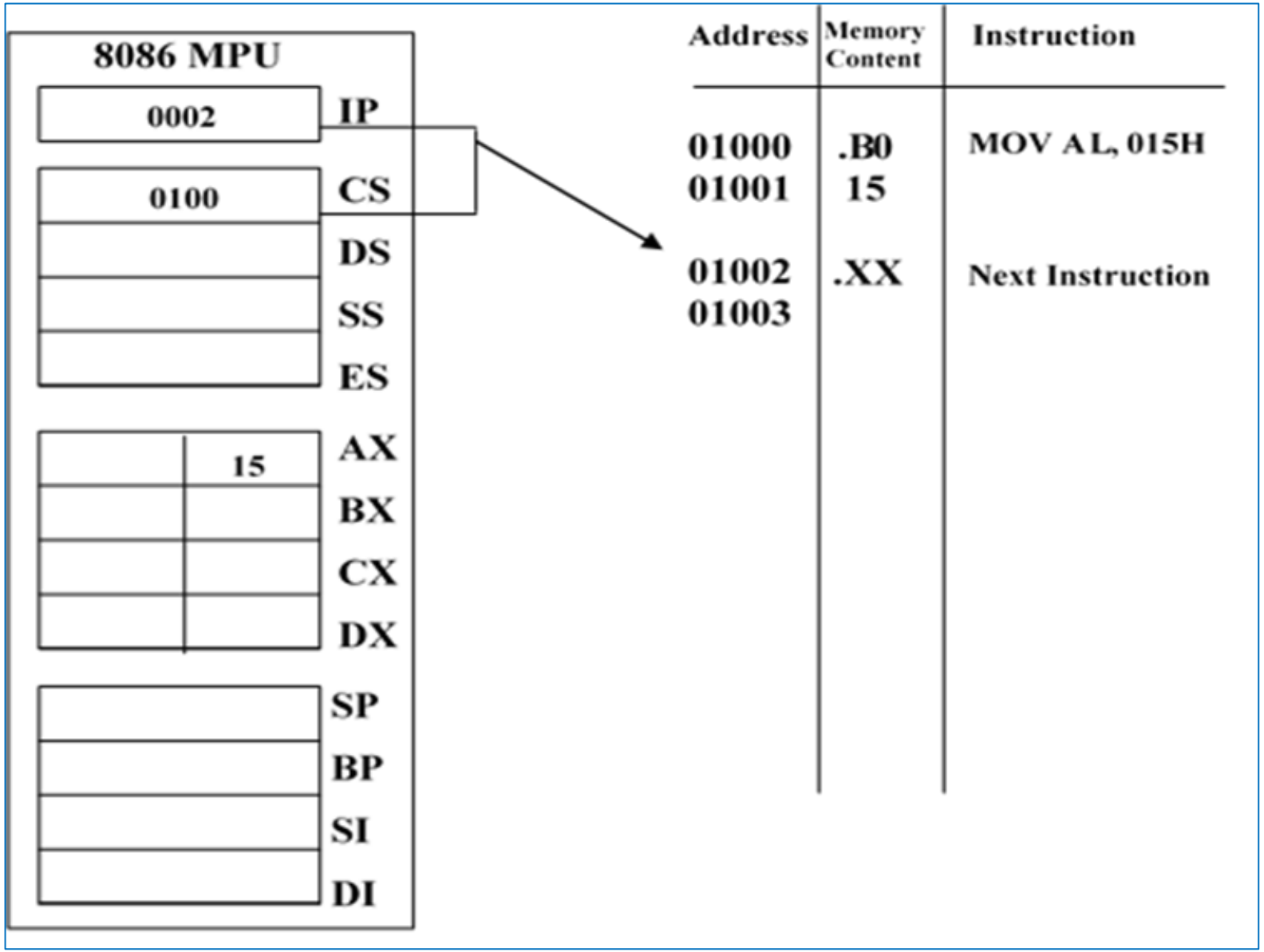

Fig. 8 (b): Immediate addressing mode after execution.

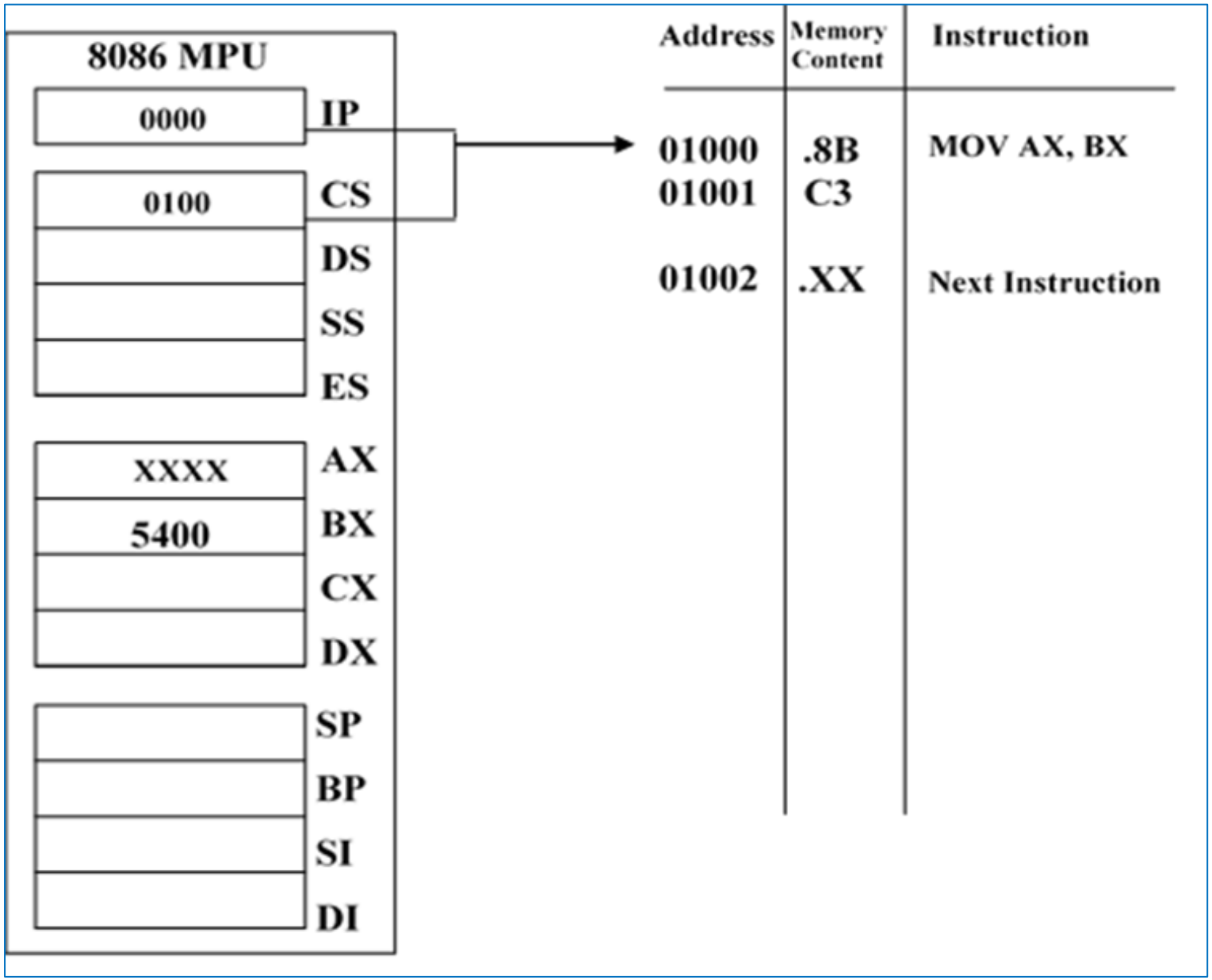

Fig. 9 (a): Register addressing mode before execution.

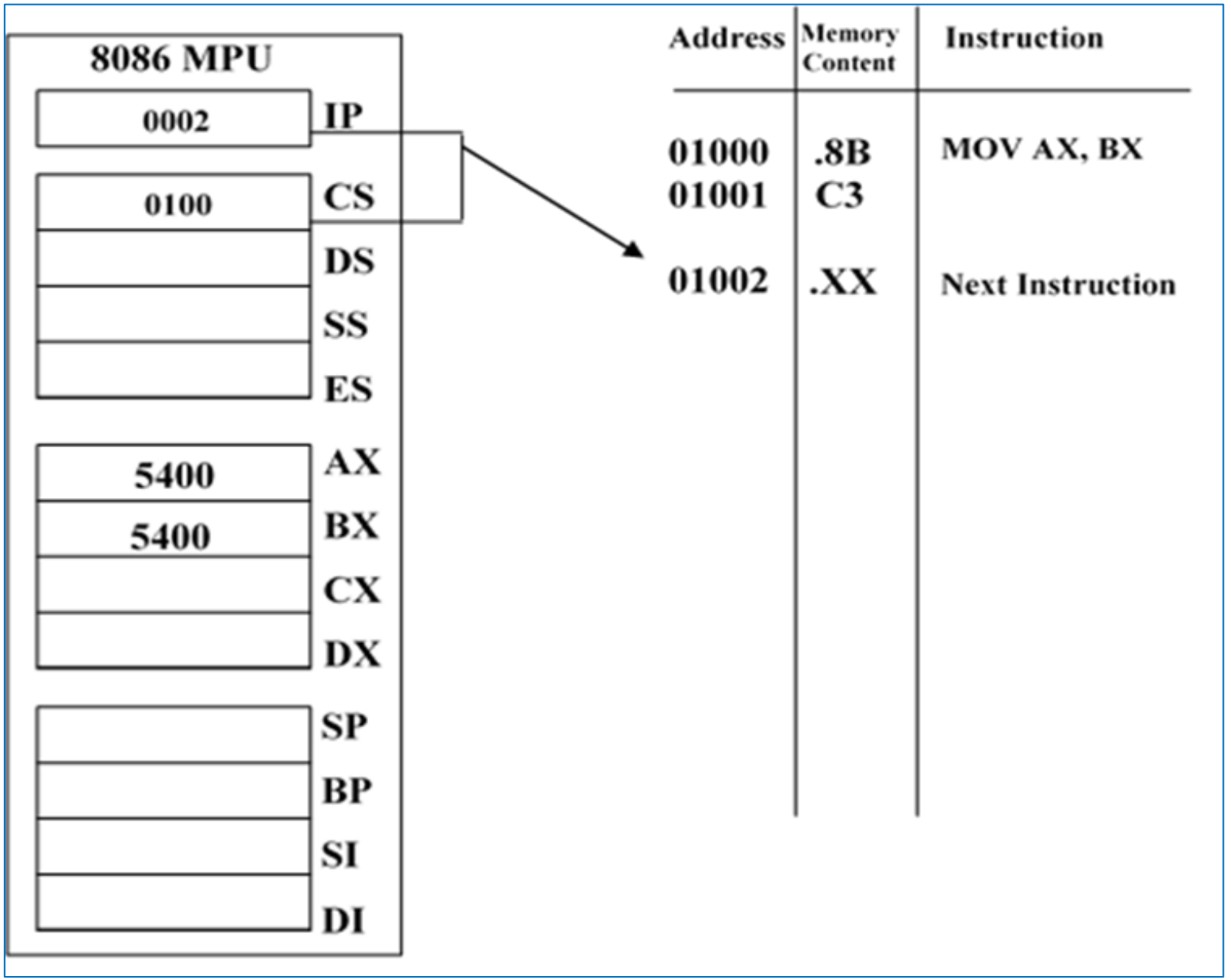

Fig. 9 (b): Register addressing mode after execution.

Addressing modes:

3. **Direct addressing mode:** In the direct addressing mode a 16-bit memory address (offset) is directly specified in the instruction as a part of it.

- Ex: MOV CX, [1234H]
- Here, the operand resides in a memory location in the data segment, whose effective address may be completed using 1234H as the offset address and content of DS as segment address. The effective address here, is 10H * DS + 1234H.

Addressing modes:

4. **Register indirect addressing mode:** Sometimes, the address of the memory location, which contains data or operand, is determined in an indirect way, using the offset register. This mode of addressing is known as register indirect mode. In this addressing mode, the offset address of data is in either BX or SI or DI or BP registers. The default segment is either DS or ES or SS. The data is supposed to be available at the address pointed to by the content of any of the above registers in the default data segment.

- Ex: MOV AX, [SI]
- Here, data is present in a memory location in DS whose offset address is in SI. The effective address of the data is given as 10H * DS + SI.

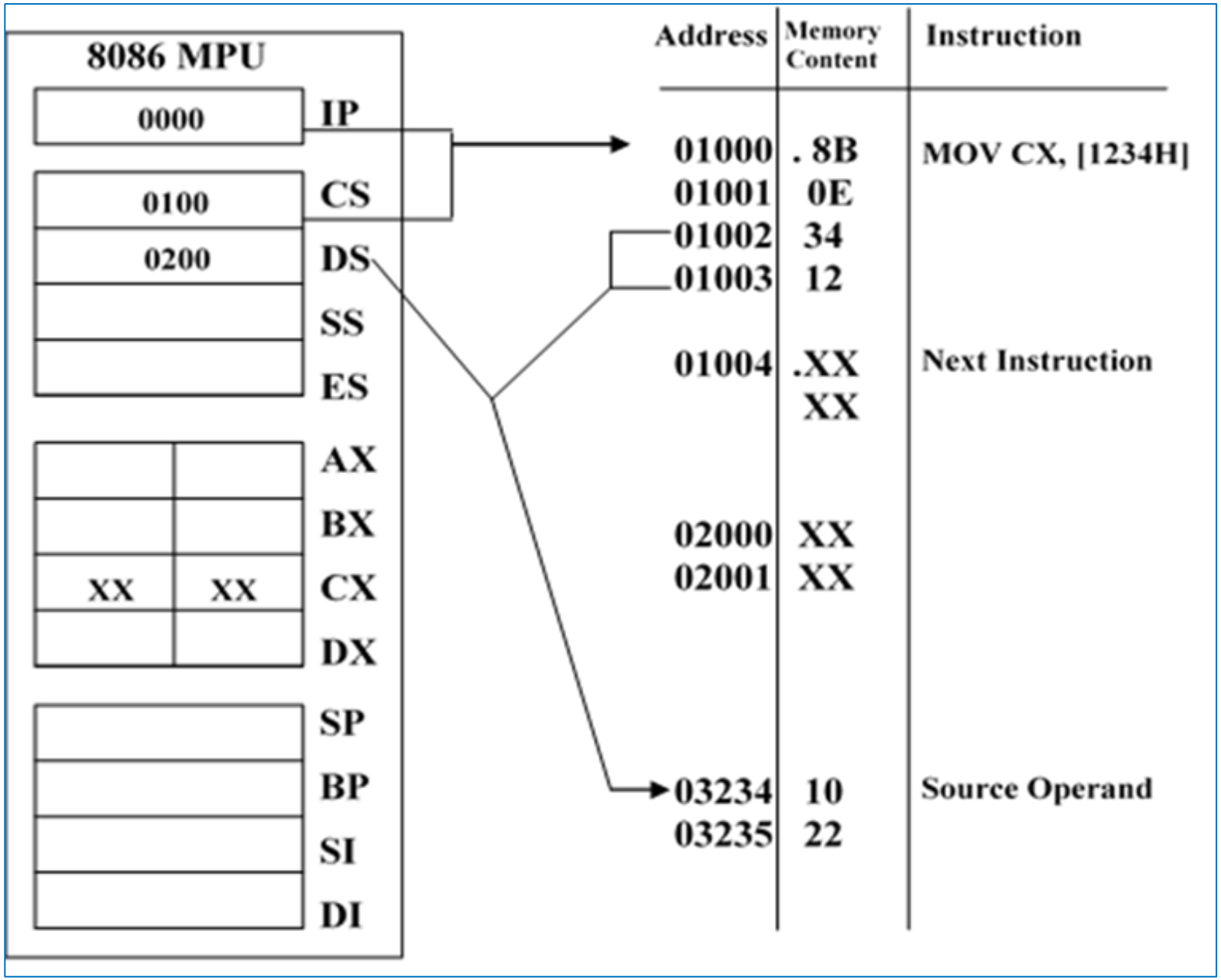

Fig. 10 (a): Direct Addressing mode before execution.

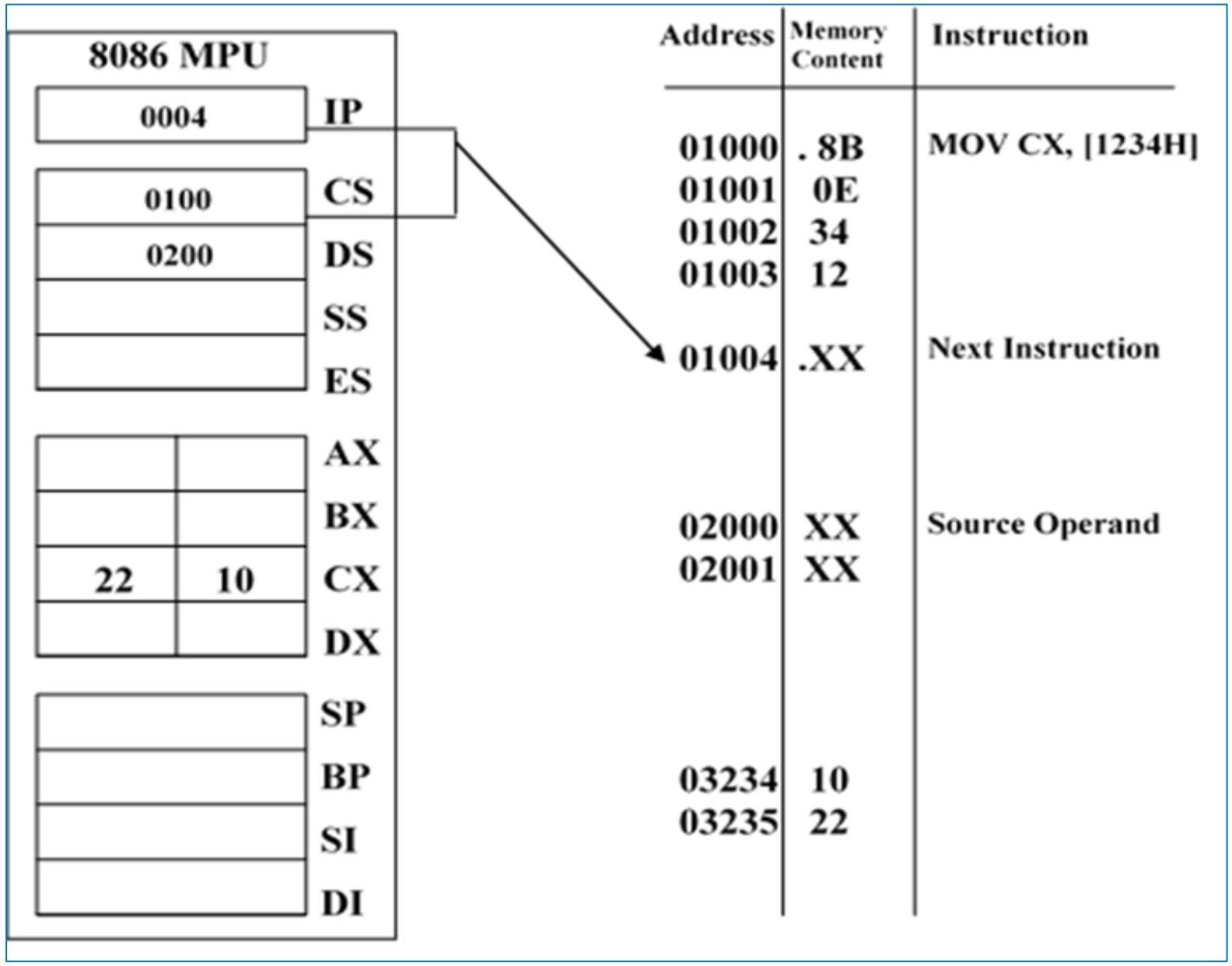

Fig. 10 (b): Direct Addressing mode after execution.

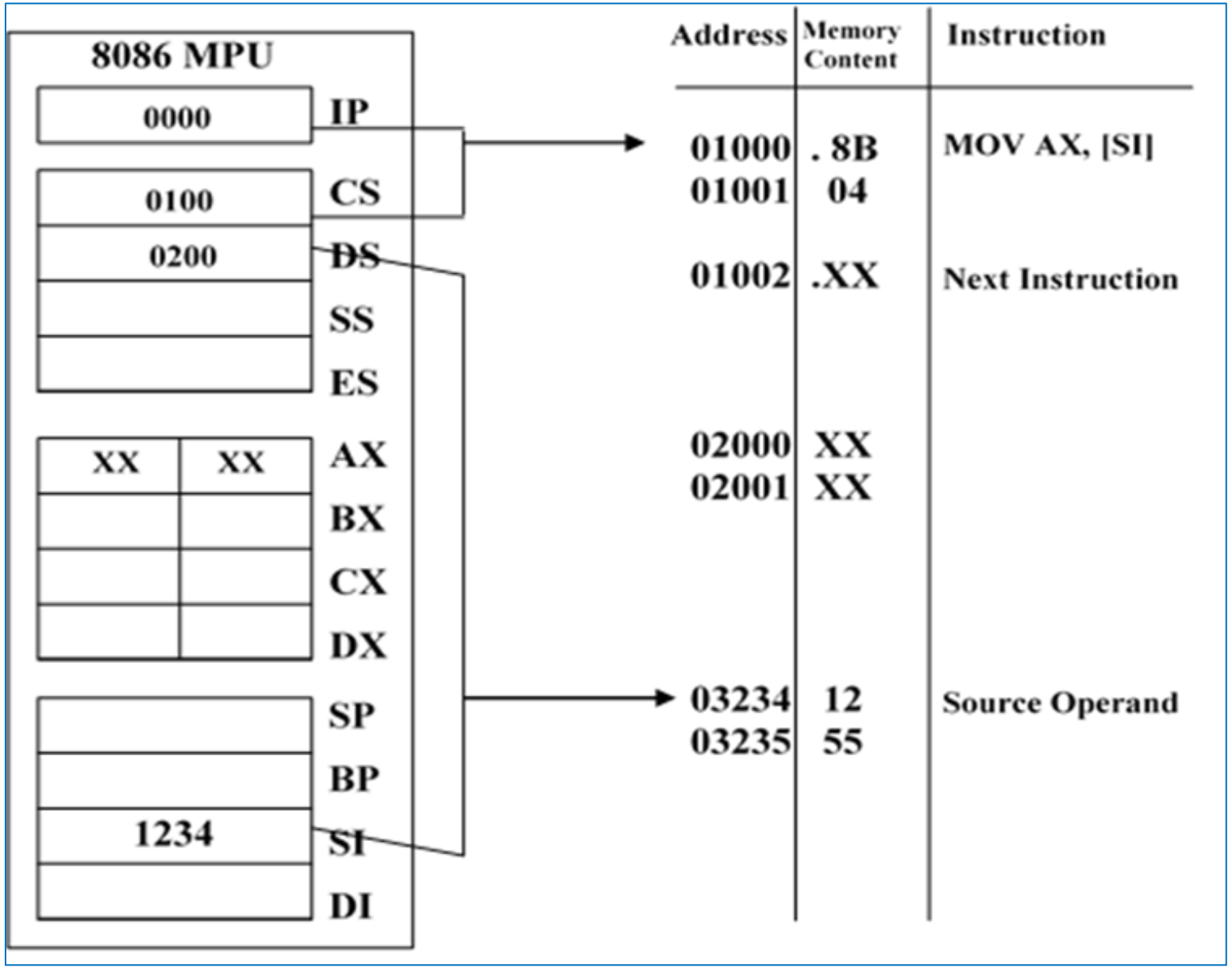

Fig. 11 (a): Register Indirect Addressing before execution.

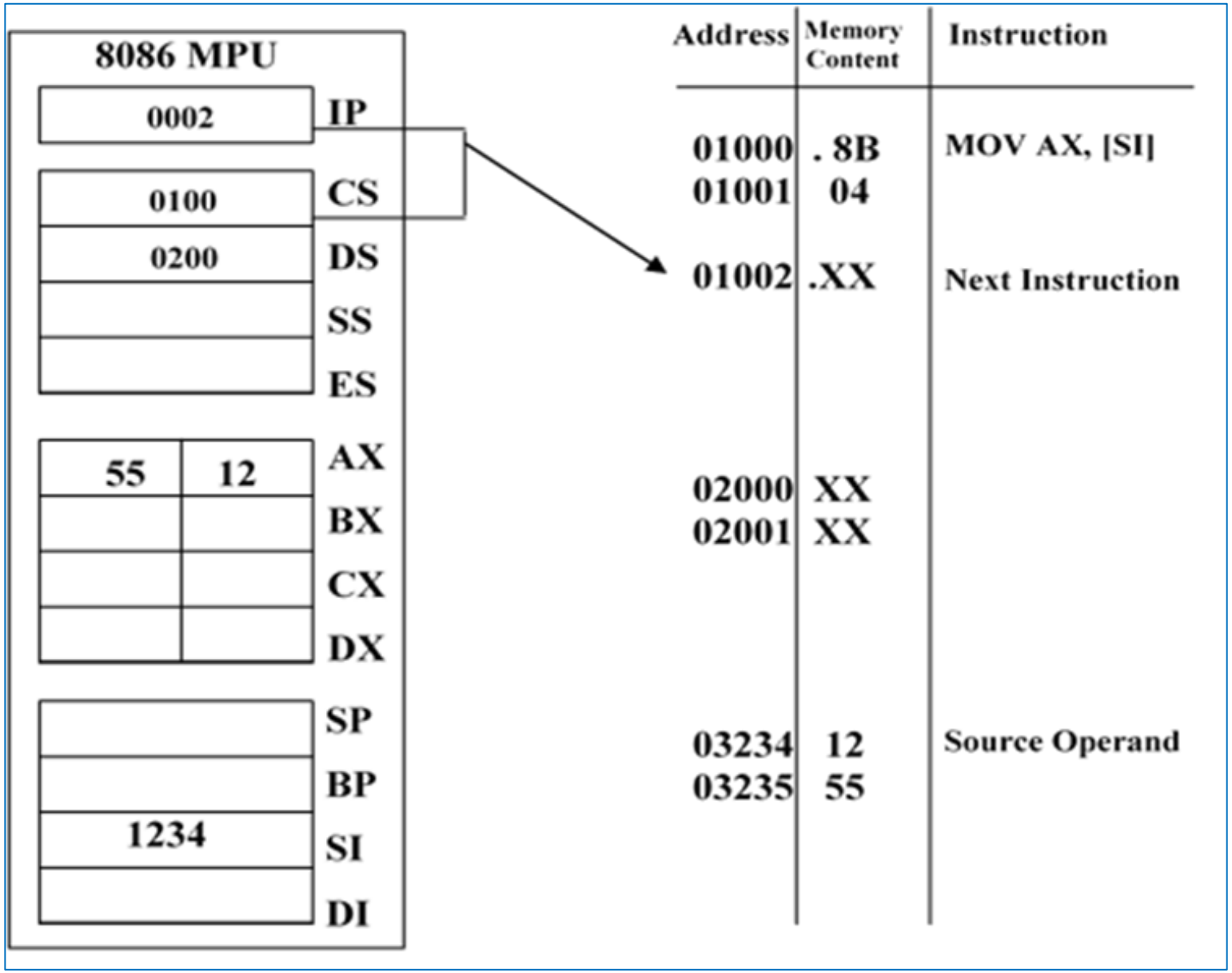

Fig. 11(b): Register Indirect Addressing mode after execution.

Addressing modes:

5. **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.

- Ex: MOV [BX] + 1234H, AL; EA=BX+1234H, PA=DS*10H+EA
- After execute the instruction the content of register AL is moved to memory location specified by PA (as shown in Fig. 12)

Addressing modes:

6. **Indexed Addressing mode:** In the Indexed addressing mode, the effective address of the operand is obtained by adding a direct or indirect displacement to the contents of either SI or DI register. 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.

- Ex: MOV BL, [SI]+1234H; EA=SI+1234H, PA=DS*10H+EA

- After execute the instruction the byte of data stored at this location (PA), is read into lower byte BX.

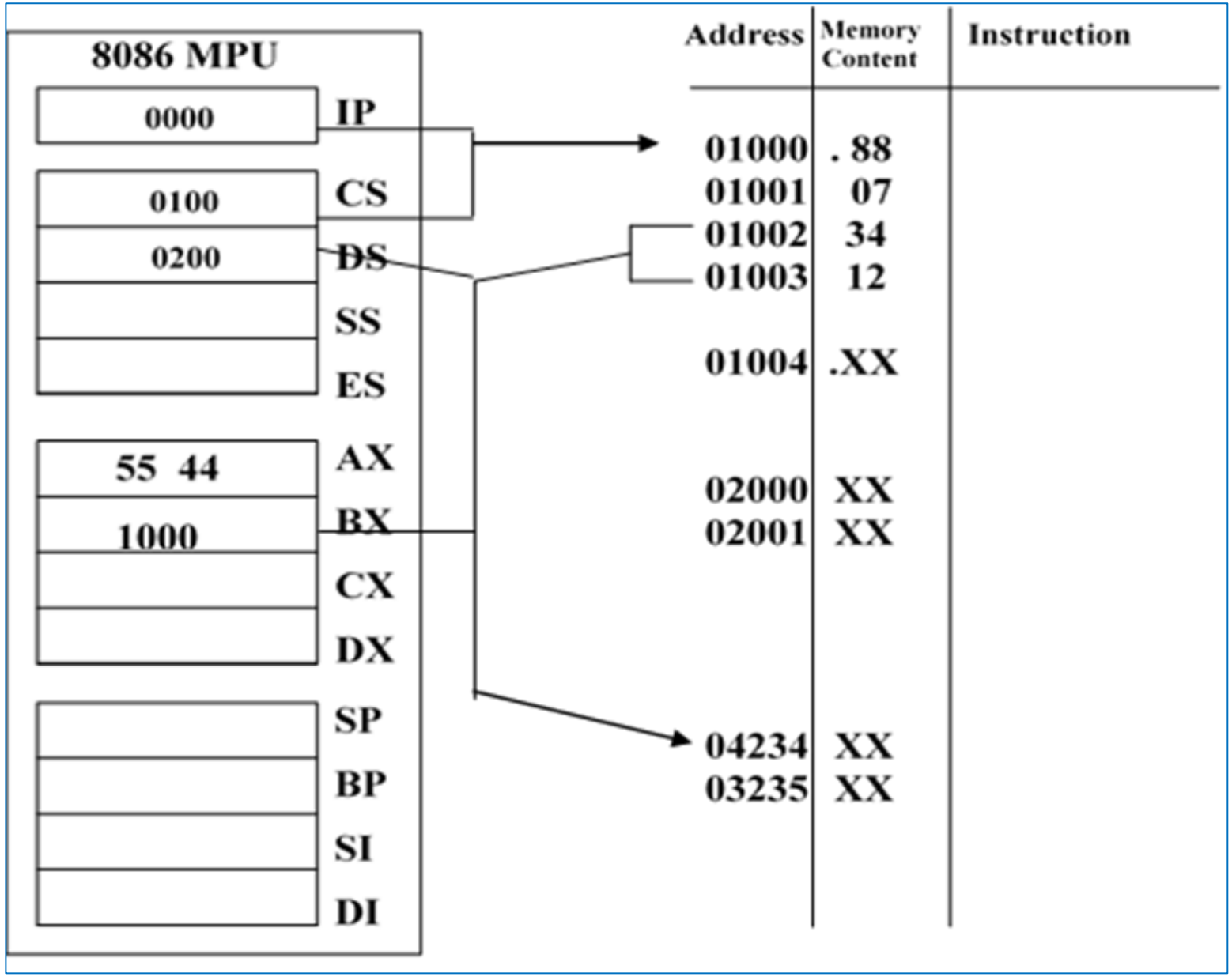

Fig. 12(a): Based Addressing before execution.

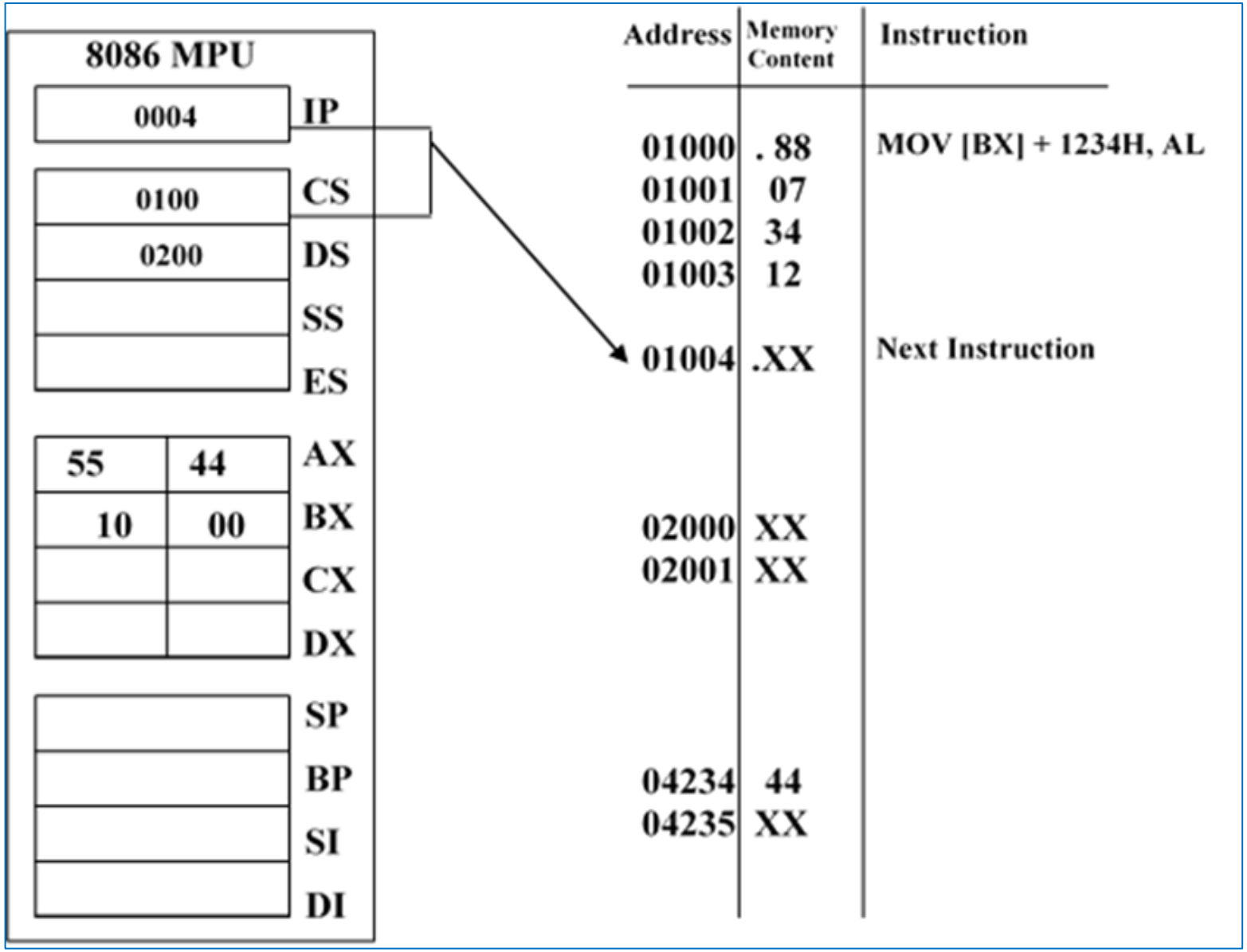

Fig. 12(b): Based Addressing mode after execution.

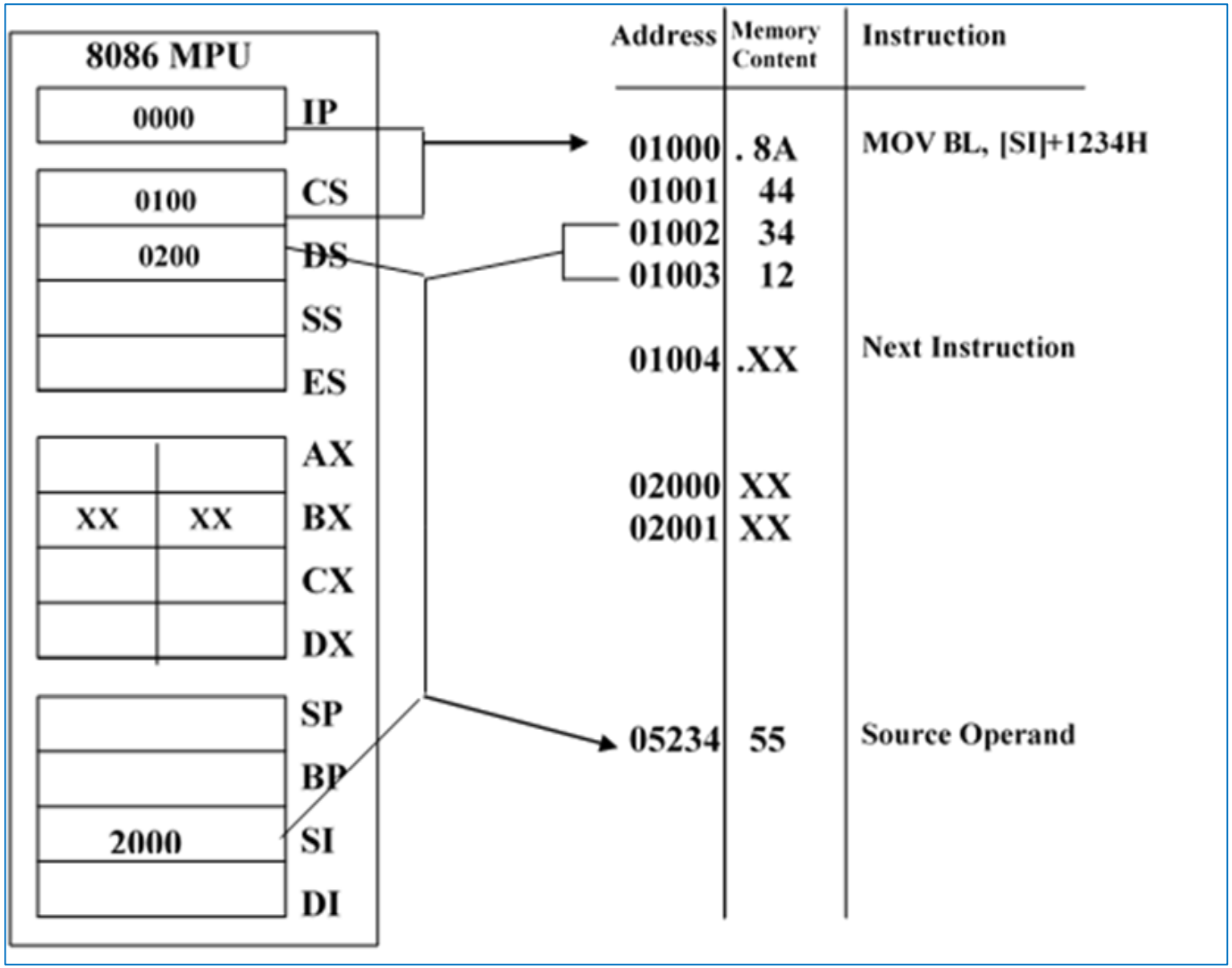

Fig. 13(a): Indexed Addressing before execution.

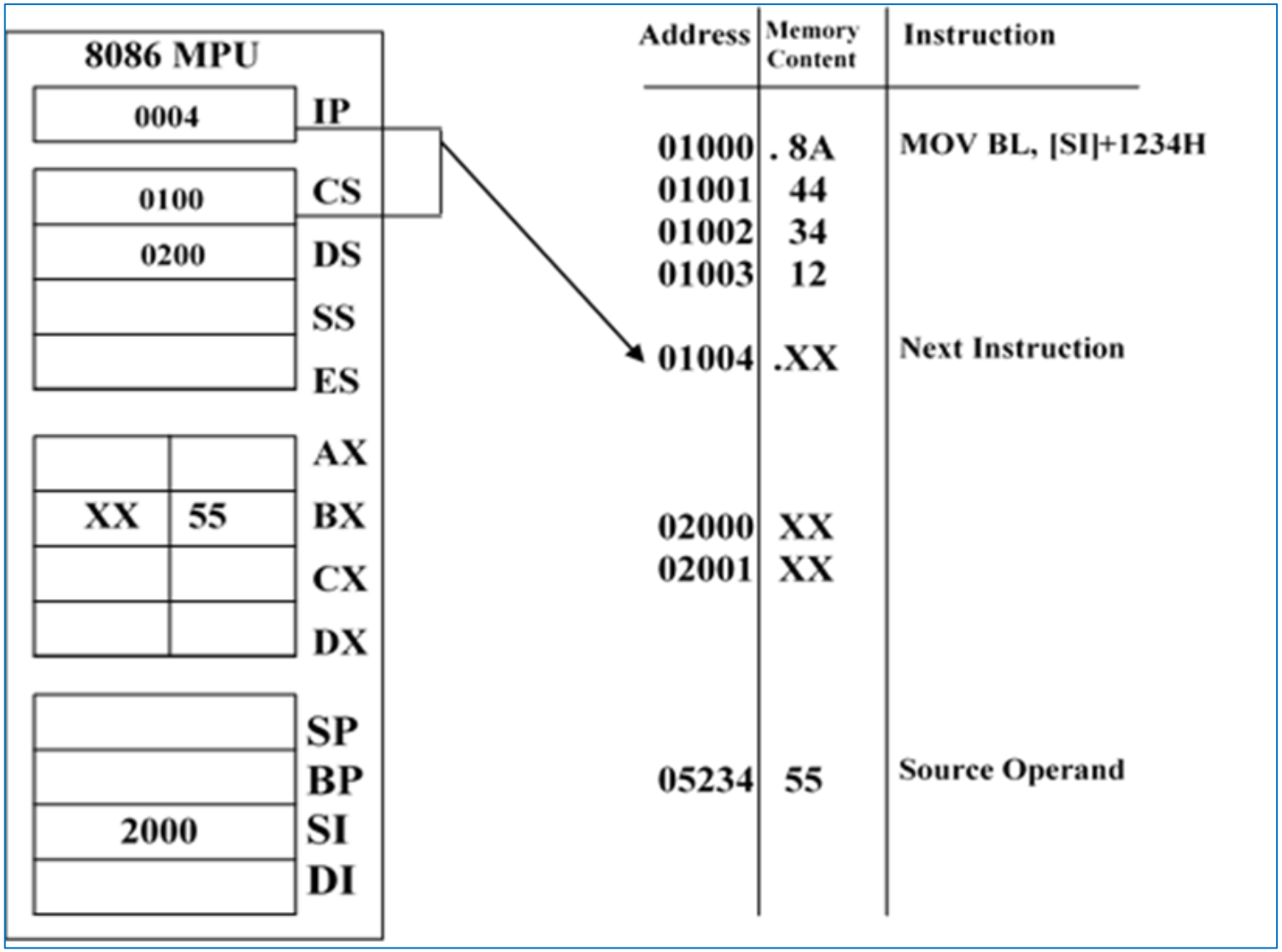

Fig. 13(b): Direct Indexed Addressing mode after execution.

Addressing modes:

7. **Based -Index addressing mode:** Combining the based addressing mode and the indexed addressing mode together results in a new, more powerful mode known as based indexed addressing. The effective address of data is formed, by adding an 8 or 16-bit displacement with the content of a base register (any one of BX or BP) and the content of an index register (any one of SI or DI). The default segment register may be ES or DS.

- Ex: MOV AX, 1234H [BX] [SI]
- Here, 1234H is an immediate displacement, BX is base register and SI is an index register the effective address of data is computed as

  EA= [BX] + [SI] + 1234H

  And the physical address

  PA=10H * DS + EA

Addressing modes:

8. **String Addressing Mode:** The string instructions of the 8086's instruction set automatically use the source and destination index registers to specify the effective addresses of the source and destination operands, respectively.

- Ex: MOVSB

The physical address for the source operand

10H * DS + [SI]

The physical address for the destination operand

10H * ES + [DI]

- Notice that neither SI nor DI appears in the string instruction, but both are used during its execution.

Addressing modes:

9. **Input / Output mode:** This addressing mode is related with input output operations.

- Ex:

  IN Al, 45

  OUT DX, AL