Department of Computer Engineering Techniques (Stage: 4)
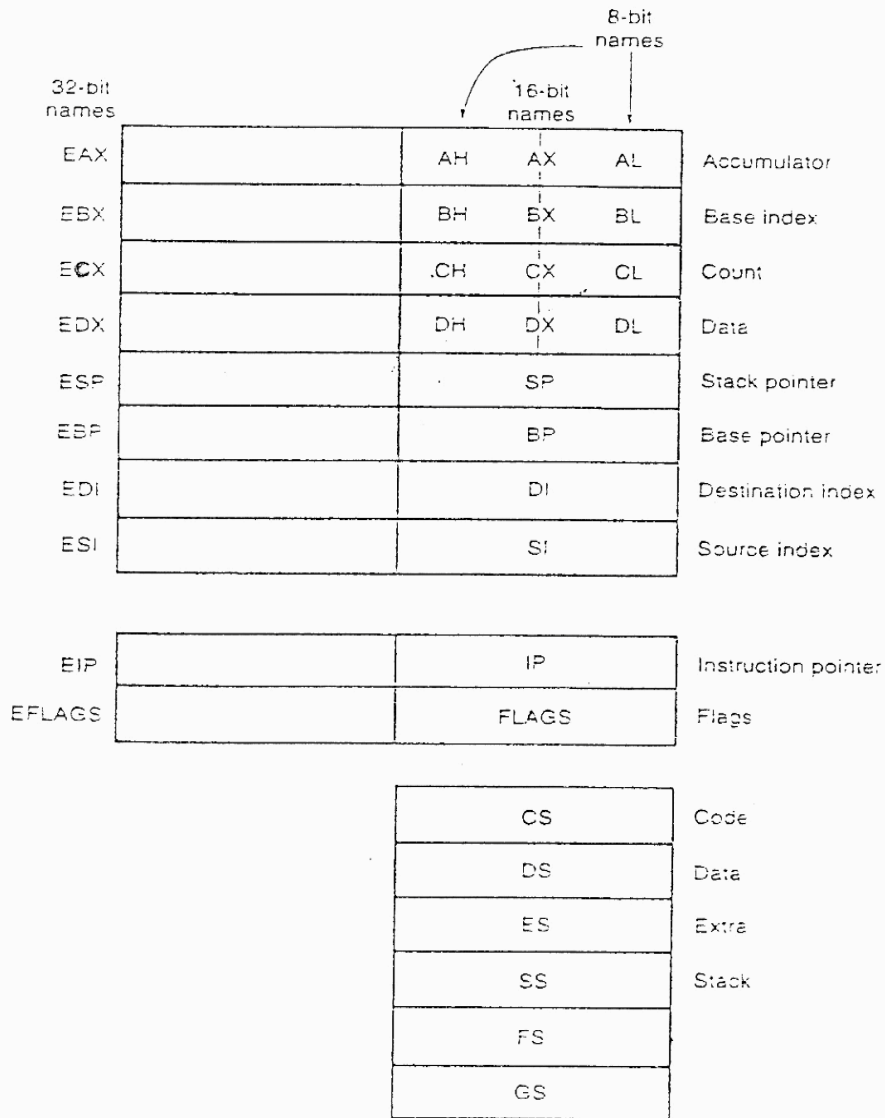
Advance Computer Technologies

Dr.: Mayas Aljibawi

The µP and its Architecture

| 32-bit names		16-bit names	8-bit names			
EAX		AH	AX	AL		Accumulator
EBX		BH	BX	BL		Base index
ECX		.CH	CX	CL		Count
EDX		DH	DX	DL		Data
ESP			SP			Stack pointer
EBP			BP			Base pointer
EDI			DI			Destination index
ESI			SI			Source index
EIP			IP			Instruction pointer
EFLAGS			FLAGS			Flags
			CS			Code
			DS			Data
			ES			Extra
			SS			Stack
			FS			
			GS			

Notes.
   1. The shaded areas registers exist only on the 80386 through the Pentium II

   2. The FS and GS register have no special names.

Addressing Modes

FIGURE 3–1 The MOV
instruction showing the source,
destination, and direction of
data flow.

MOV AX,BX

Source

Destination
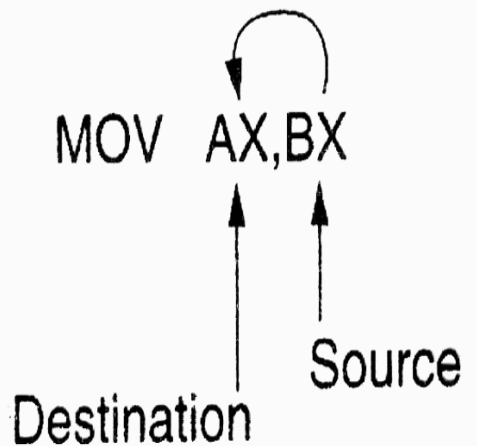
1. Register addressing
2. Immediate addressing

3. Direct addressing
4. Register indirect addressing
5. Base-plus-index addressing
6. Register relative addressing
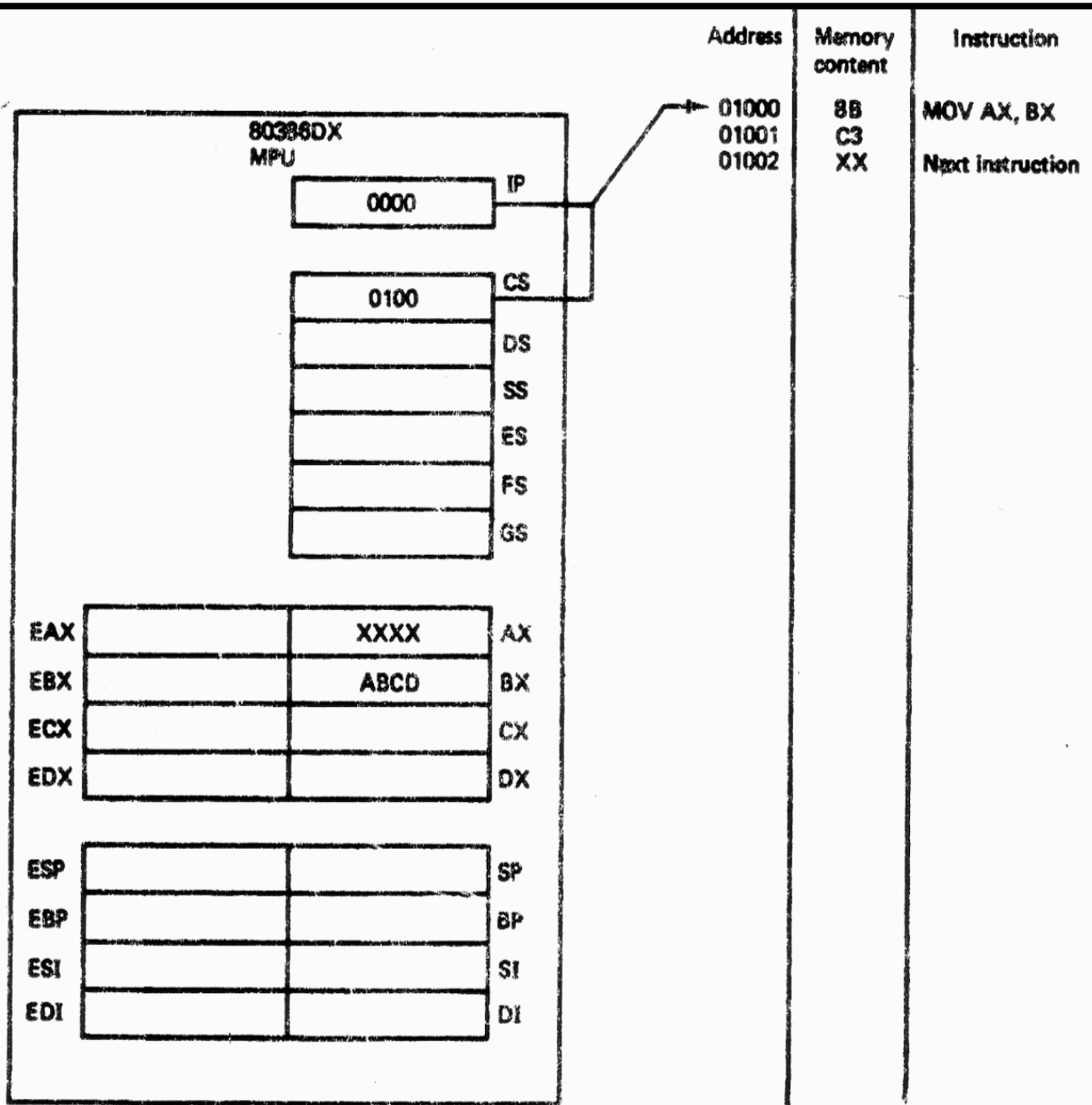7. Base relative-plus-index addressing

8. Scaled-index addressing
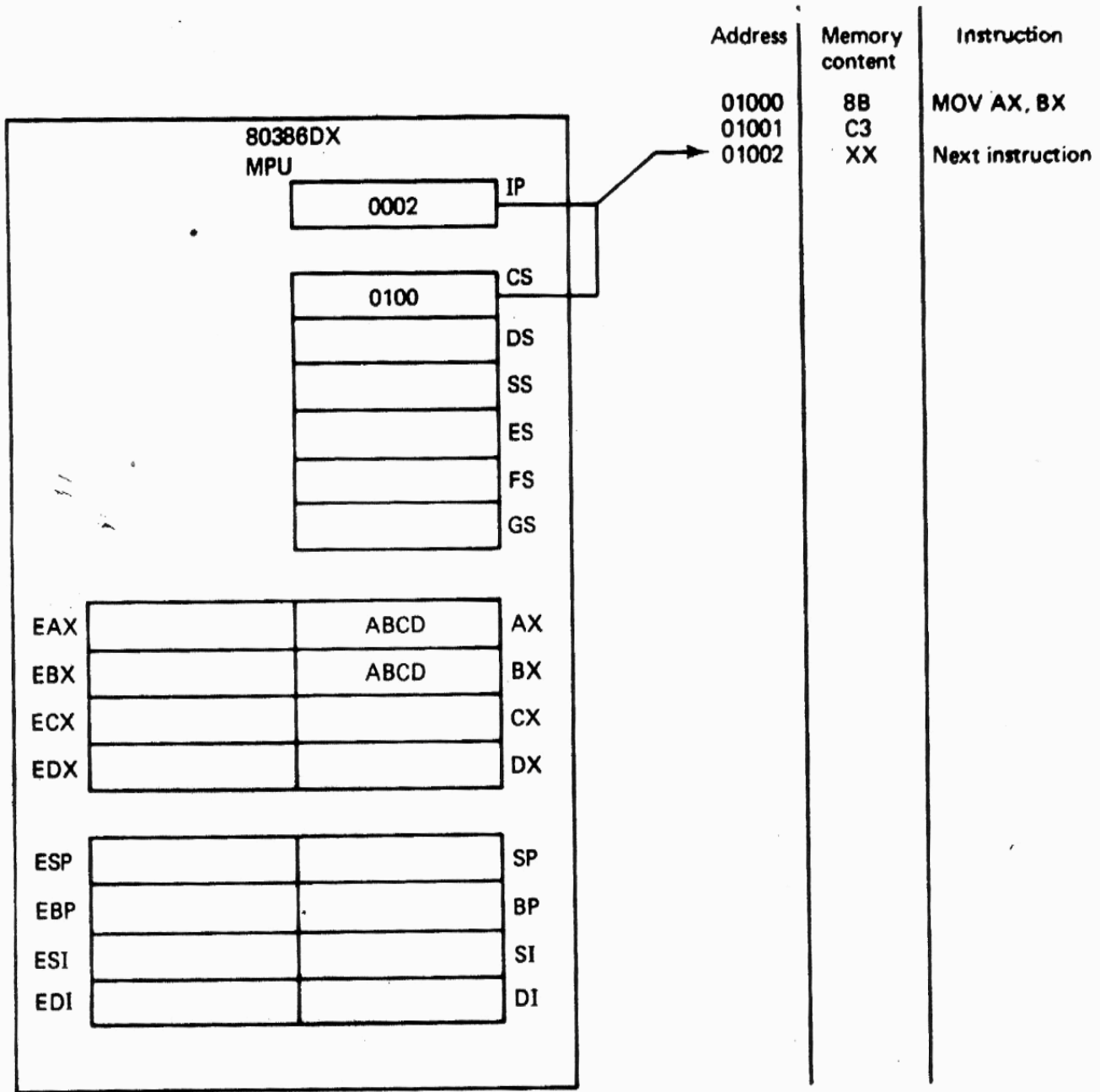(80386 – P4)

1.Register Operand Addressing Mode

MOV   AX, BX

Register	Operand size		
	Byte (Reg8)	Word (Reg16)	Double word (Reg32)
Accumulator	AL, AH	AX	EAX
Base	BL, BH	BX	EBX
Count	CL, CH	CX	ECX
Data	DL, DH	DX	EDX
Stack pointer	–	SP	ESP
Base pointer	–	BP	EBP
Source index	–	SI	ESI
Destination index	–	DI	EDI
Code segment	–	CS	–
Data segment	–	DS	–
Stack segment	–	SS	–
E data segment	–	ES	–
F data segment	–	FS	–
G data segment	–	GS	–

Figure 3.7   Direct addressing registers and operand sizes.

Address	Memory content	Instruction
01000	8B	MOV AX, BX
01001	C3	
01002	XX	Next instruction

80386DX
MPU

IP	0000
CS	0100
DS	
SS	
ES	
FS	
GS	

EAX		XXXX	AX
EBX		ABCD	BX
ECX			CX
EDX			DX

ESP			SP
EBP			BP
ESI			SI
EDI			DI

(a)

Figure 3.8 (a) Register addressing mode instruction before fetch and execution. (b) After execution.

Address	Memory content	Instruction
01000	8B	MOV AX, BX
01001	C3	
01002	XX	Next instruction

80386DX
MPU

IP 0002

CS 0100
DS
SS
ES
FS
GS

EAX		ABCD	AX
EBX		ABCD	BX
ECX			CX
EDX			DX

ESP			SP
EBP			BP
ESI			SI
EDI			DI

(b)

Figure 3.8 (Continued)

2.Immediate Operand Addressing Mode

Address	Memory content	Instruction
01000	B0	MOV AL, 15H
01001	15	
01002	XX	Next instruction
01003	XX	

80386DX
MPU

IP 0000

CS 0100

DS

SS

ES

FS

GS

EAX		XX	AX
EBX			BX
ECX			CX
EDX			DX

ESP		SP
EBP		BP
ESI		SI
EDI		DI

(a)
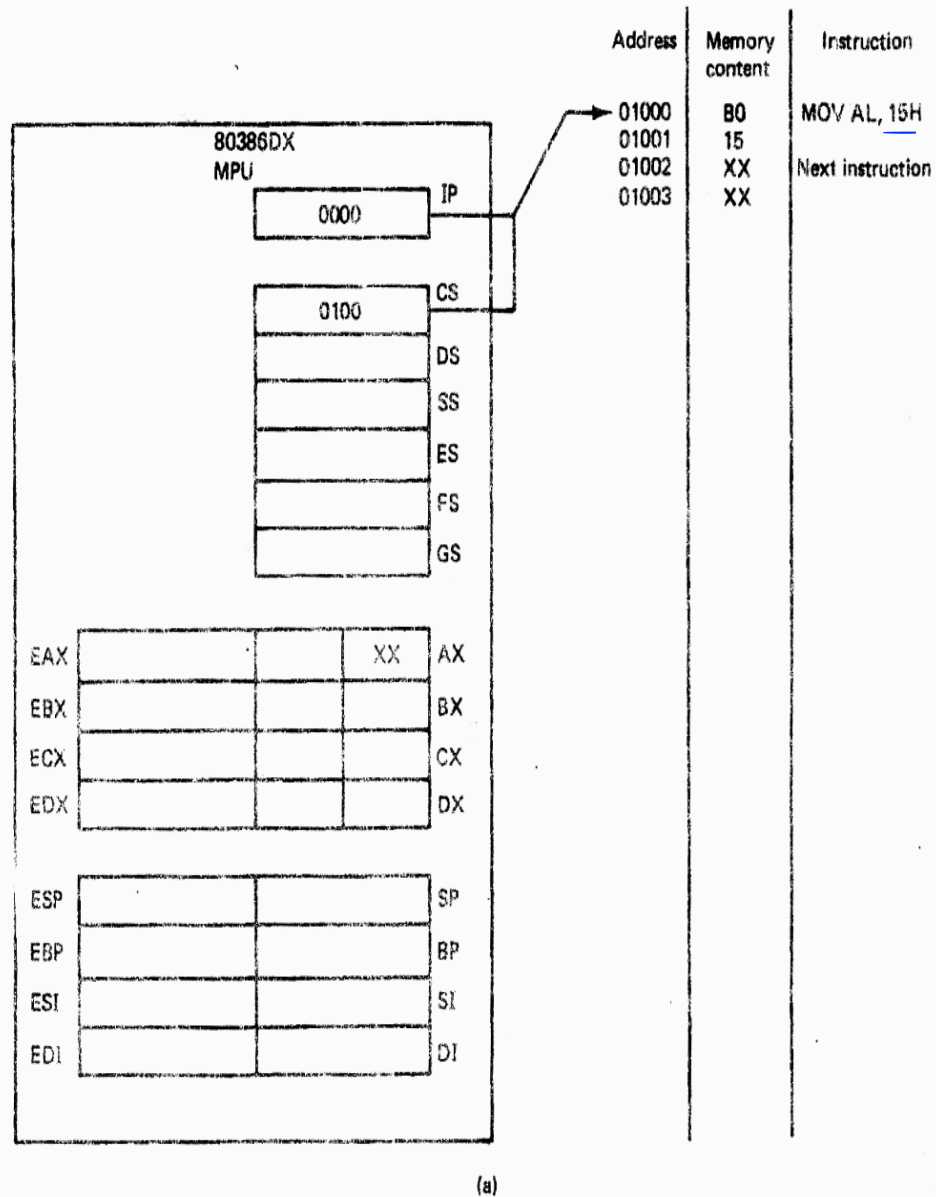
Figure 3.10 (a) Immediate addressing mode instruction before fetch and execution. (b) After execution.

Address	Memory content	Instruction
01000	B0	MOV AL, 15H
01001	15	
01002	XX	Next instruction
01003	XX	

80386DX
MPU

IP — 0002

CS — 0100

DS

SS

ES

FS

GS

EAX — 15 — AX

EBX — BX

ECX — CX

EDX — DX

ESP — SP

EBP — BP

ESI — SI

EDI — DI
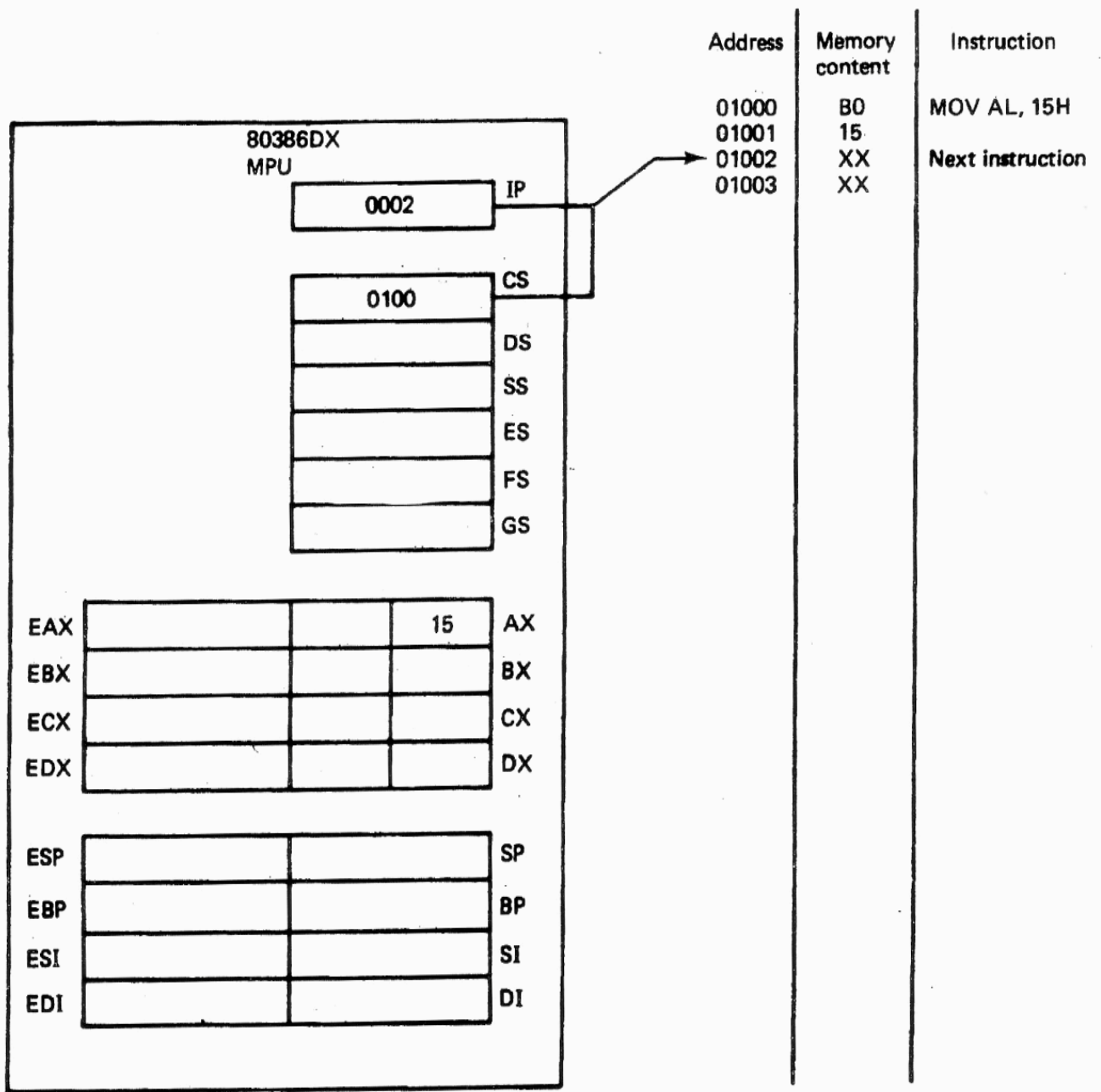
(b)

Figure 3.10 (Continued)

16-Bit Memory Operand Addressing Modes

$$EA = base + index + displacement$$

effective address

$$PA = SBA : EA$$

$$PA = \text{Segment base} : \text{Base} + \text{Index} + \text{Displacement}$$

$$PA = \begin{Bmatrix} CS \\ SS \\ DS \\ ES \end{Bmatrix} : \begin{Bmatrix} BX \\ BP \end{Bmatrix} + \begin{Bmatrix} SI \\ DI \end{Bmatrix} + \begin{Bmatrix} \text{8-bit displacement} \\ \text{16-bit displacement} \end{Bmatrix}$$

3.Direct Addressing Mode

Address	Memory content	Instruction
01000	8B	MOV CX, [1234H]
01001	0E	
01002	34	
01003	12	
01004	XX	Next instruction

80386DX
MPU

IP 0000

CS 0100
DS 0200
SS
ES
FS
GS

EAX — AX
EBX — BX
ECX XXXX CX
EDX — DX

ESP — SP
EBP — BP
ESI — SI
EDI — DI

| 02000 | XX |
| 02001 | XX |

| 03234 | ED | Source operand |
| 03235 | BE | |

(a)
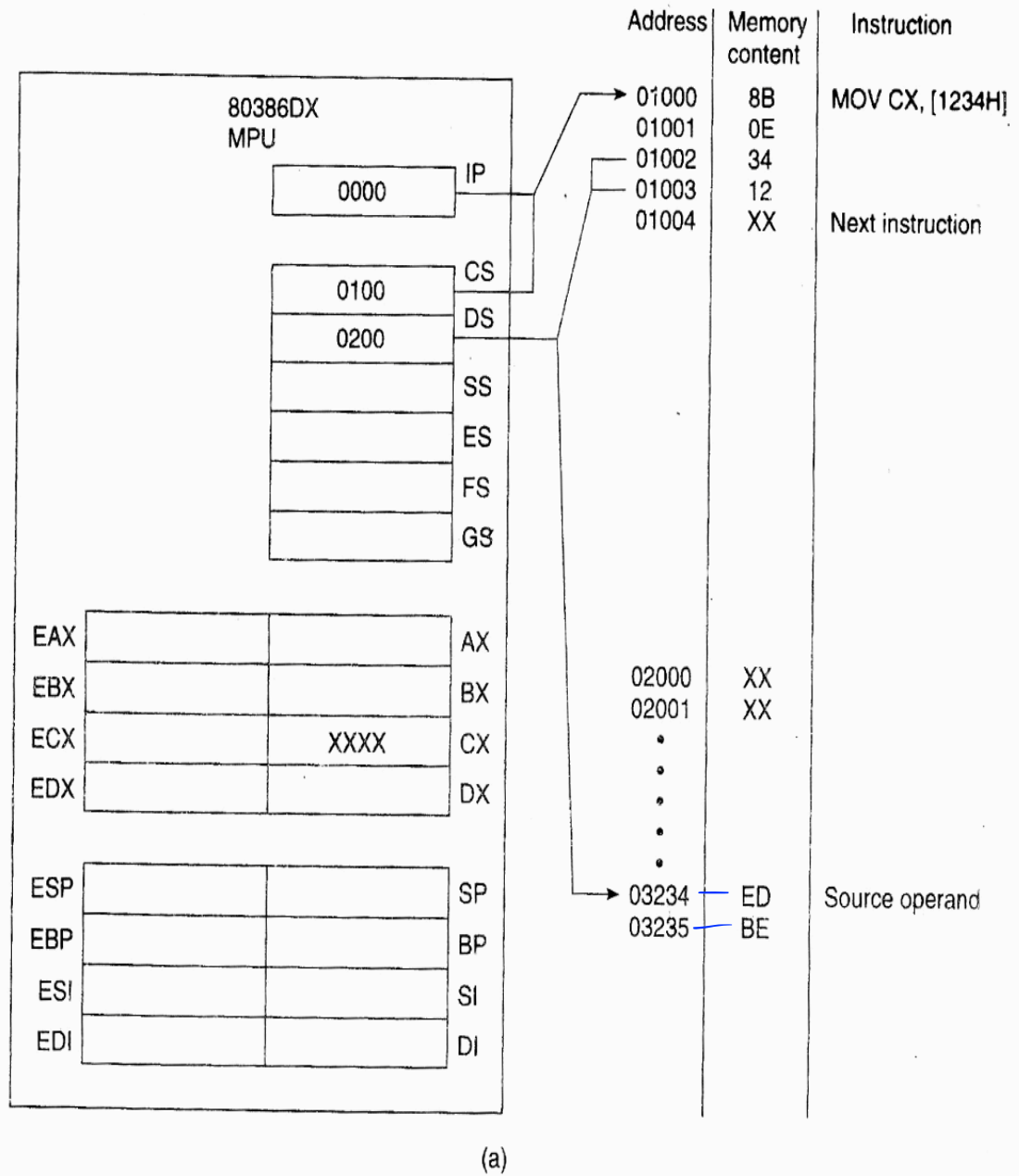
Figure 3.13 (a) Direct addressing mode instruction before fetch and execution. (b) After execution.

Address	Memory content	Instruction
01000	8B	MOV CX, [1234H]
01001	0E	
01002	34	
01003	12	
01004	XX	Next instruction
02000	XX	
02001	XX	
03234	ED	
03235	BE	

80386DX
MPU

IP 0004

CS 0100
DS 0200
SS
ES
FS
GS

EAX — AX
EBX — BX
ECX BEED CX
EDX — DX

ESP — SP
EBP — BP
ESI — SI
EDI — DI
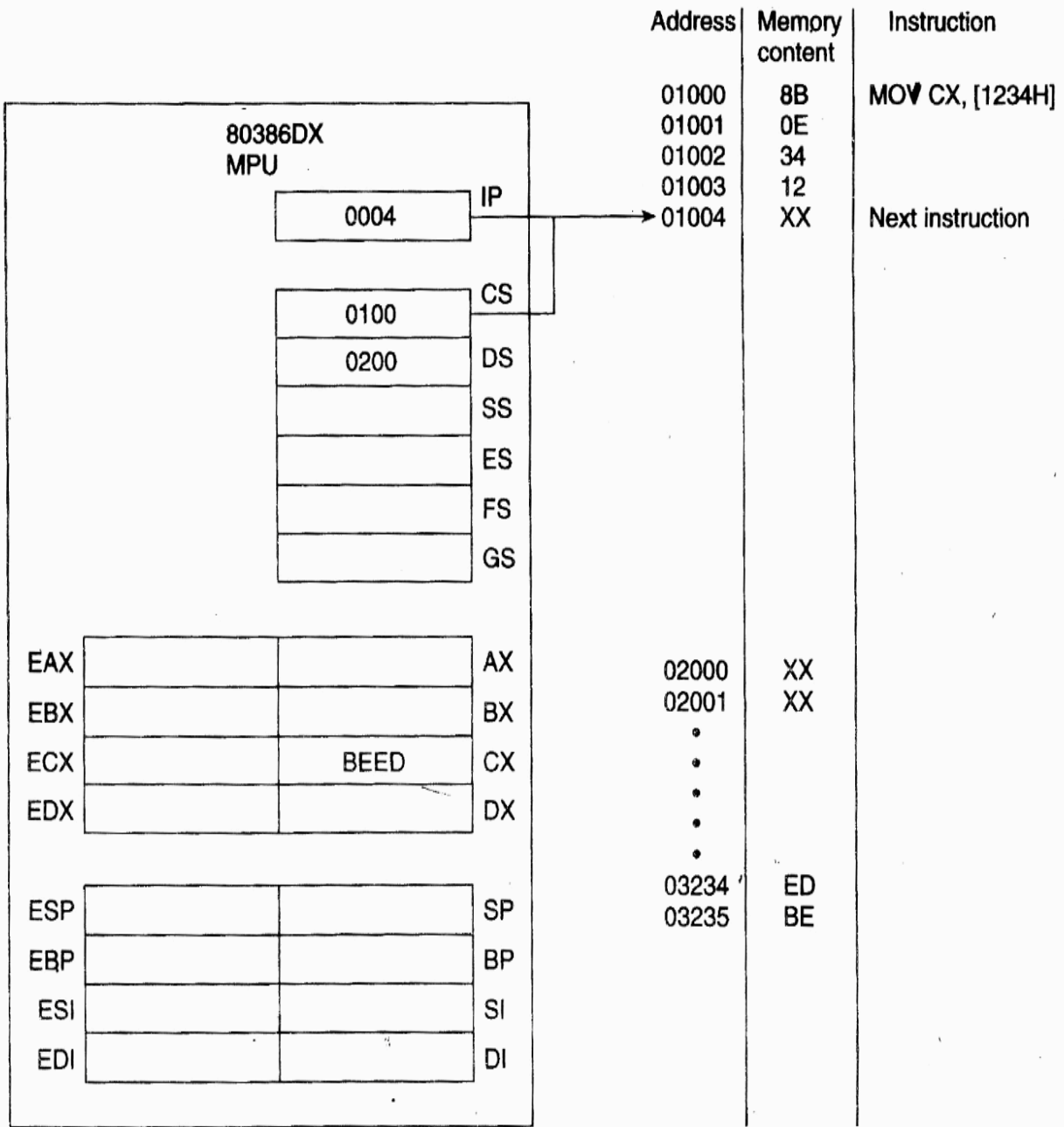
(b)

Figure 3.13   (Continued)

Example 1-15

Find the physical address of the memory location and its contents after the execution of the following, assuming that DS = 1512H.

```
MOV     AL,99H
MOV     [3518],AL
```

Solution:

First AL is initialized to 99H, then in line two, the contents of AL are moved to logical address DS:3518 which is 1512:3518. Shifting DS left and adding it to the offset gives the physical address of 18638H (15120H + 3518H = 18638H). That means after the execution of the second instruction, the memory location with address 18638H will contain the value 99H.
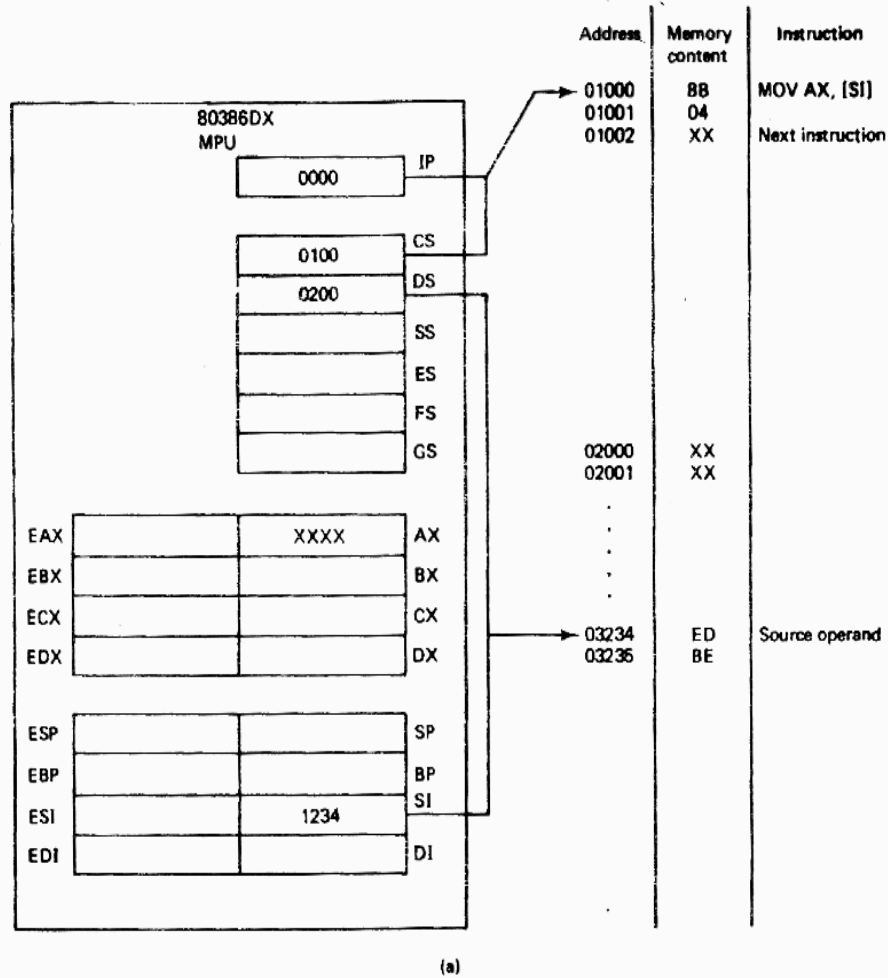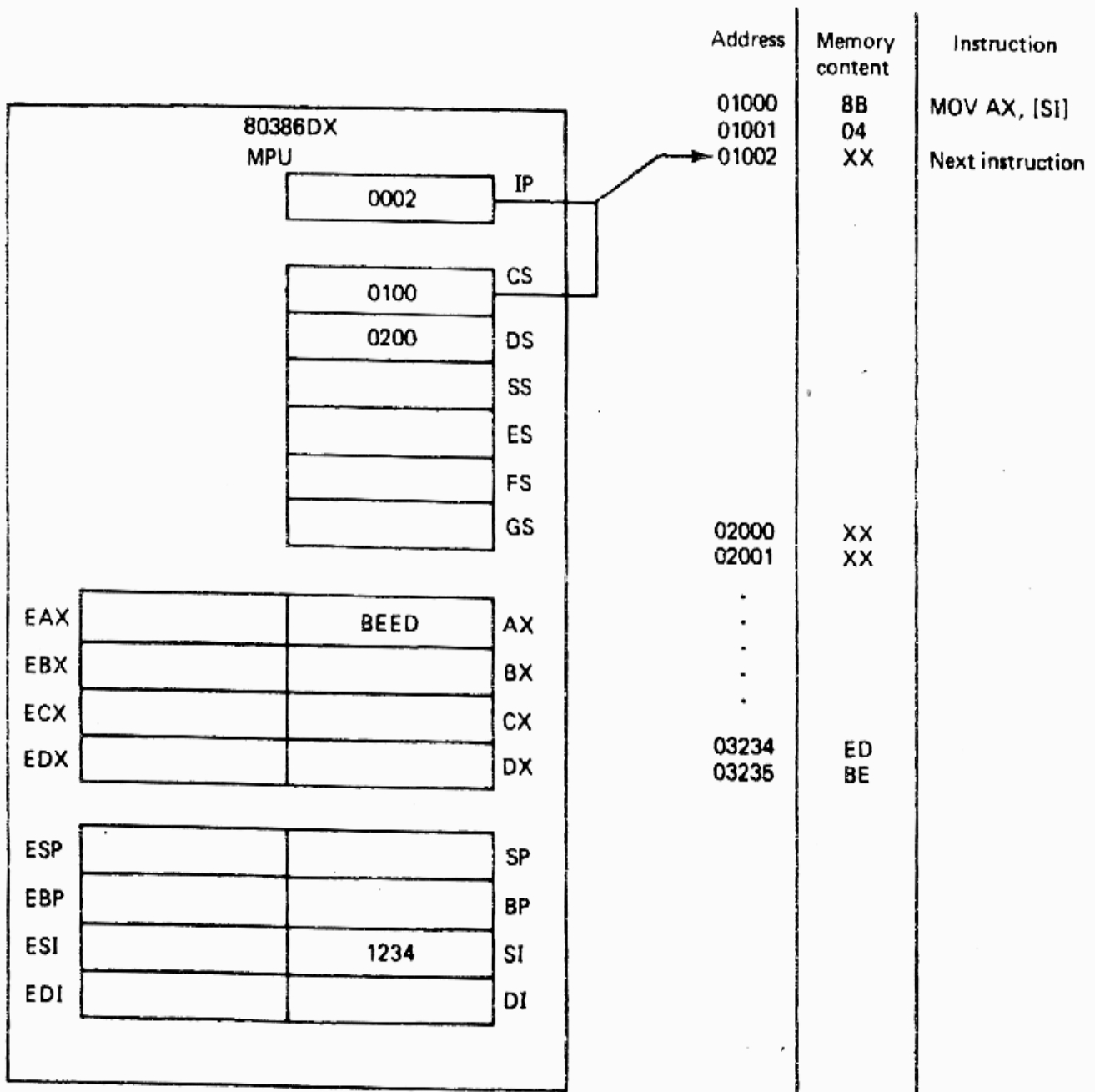
4. Register Indirect Addressing Mode

Address	Memory content	Instruction
01000	8B	MOV AX, [SI]
01001	04	
01002	XX	Next instruction
02000	XX	
02001	XX	
03234	ED	Source operand
03235	BE	

80386DX MPU

IP 0000
CS 0100
DS 0200
SS
ES
FS
GS

EAX — XXXX AX
EBX — BX
ECX — CX
EDX — DX

ESP — SP
EBP — BP
ESI — 1234 SI
EDI — DI

(a)

Figure 3.15 (a) Instruction using register indirect addressing mode before fetch and execution. (b) After execution.

Address	Memory content	Instruction
01000	8B	MOV AX, [SI]
01001	04	
01002	XX	Next instruction
02000	XX	
02001	XX	
.		
.		
.		
.		
03234	ED	
03235	BE	

80386DX
MPU

IP 0002

CS 0100
DS 0200
SS
ES
FS
GS

EAX		BEED	AX
EBX			BX
ECX			CX
EDX			DX

ESP			SP
EBP			BP
ESI		1234	SI
EDI			DI

(b)

Figure 3.15 (Continued)

Example 1-16

Assume that DS = 1120, SI = 2498, and AX = 17FE. Show the contents of memory locations after the execution of

                    MOV [SI],AX

Solution:

The contents of AX are moved into memory locations with logical address DS:SI and DS:SI + 1; therefore, the physical address starts at DS (shifted left) + SI = 13698. According to the little endian convention, low address 13698H contains FE, the low byte, and high address 13699H will contain 17, the high byte.

5. Based addressing mode

$$PA = \begin{Bmatrix} CS \\ DS \\ SS \\ ES \\ FS \\ GS \end{Bmatrix} : \begin{Bmatrix} BX \\ BP \end{Bmatrix} + \begin{Bmatrix} \text{8-bit displacement} \\ \text{16-bit displacement} \end{Bmatrix}$$

(a)

Memory

| Element n |
| Element $n-1$ |
| · · · |
| Data structure |
| · · · |
| Element 2 |
| Element 1 |
| Element 0 |

Displacement

Base register
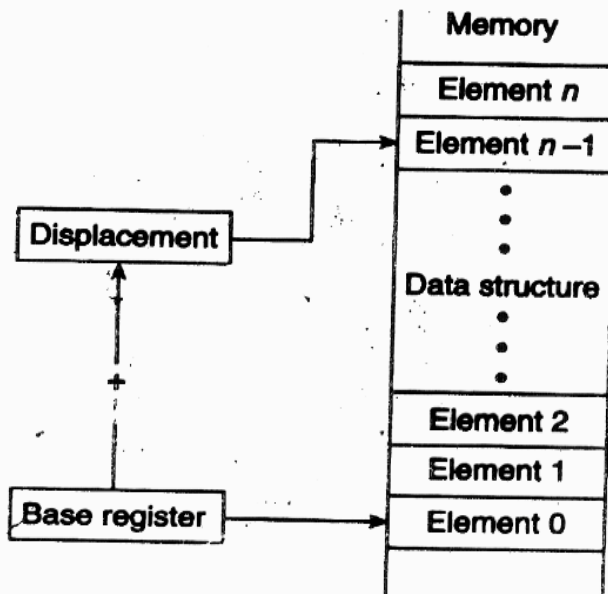
(b)

Figure 3.16 (a) Computation of a based address. (b) Based addressing of a structure of data.

Address	Memory content	Instruction
01000	88	MOV [BX] + 1234H, AL
01001	07	
01002	34	
01003	12	
01004	XX	Next instruction
02000	XX	
02001	XX	
04234	XX	Destination operand
04235	XX	

80386DX MPU

IP 0000

CS 0100
DS 0200
SS
ES
FS
GS

EAX | BE | ED | AX
EBX | 1000 | BX
ECX | | CX
EDX | | DX

ESP | | SP
EBP | | BP
ESI | | SI
EDI | | DI

(a)
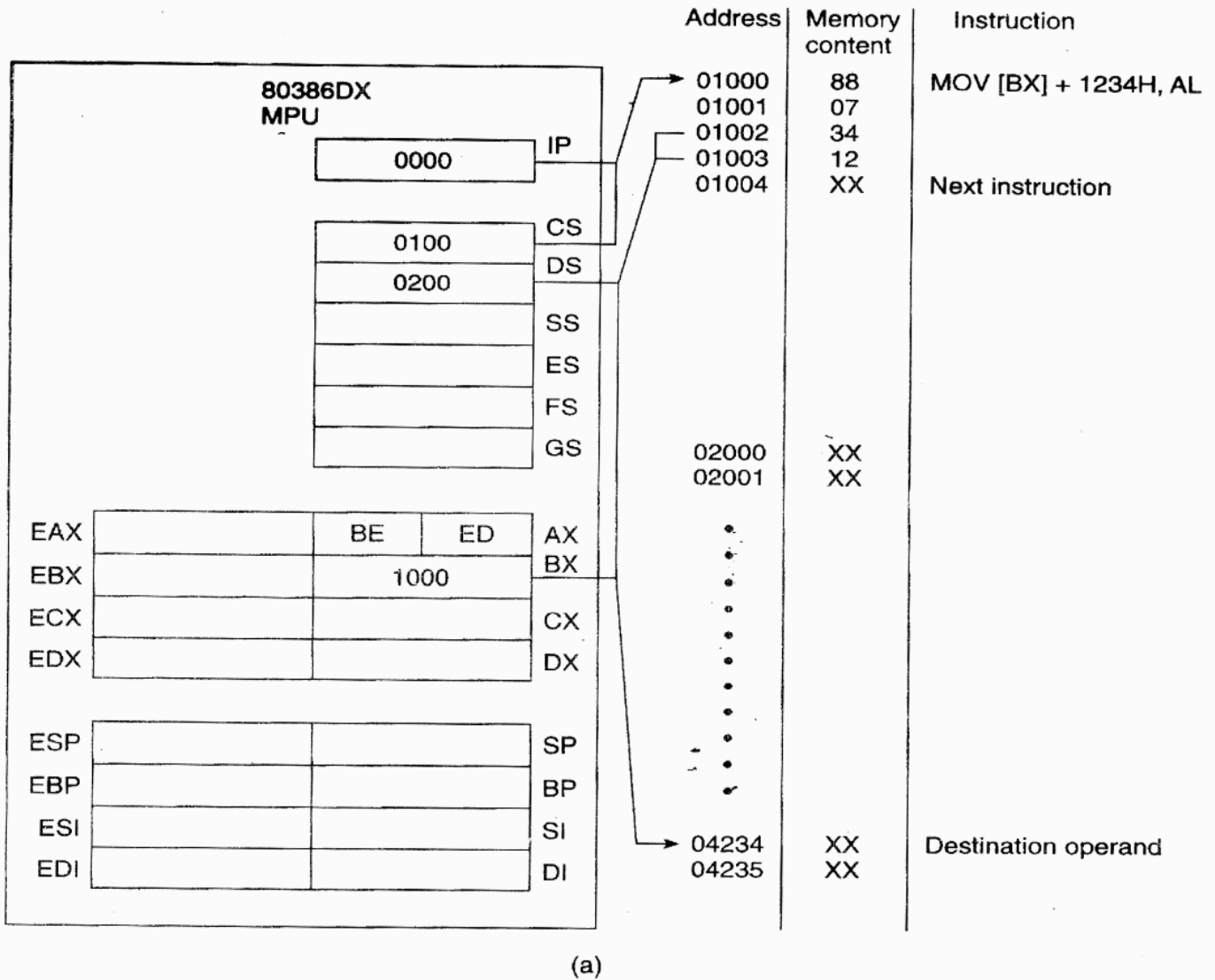
Figure 3.17 (a) Instruction using direct base pointer addressing mode before fetch and execution. (b) After execution.

Address	Memory content	Instruction
01000	88	MOV [BX] + 1234H, AL
01001	07	
01002	34	
01003	12	
01004	XX	Next instruction

80386DX
MPU

IP   0004

CS   0100
DS   0200
SS
ES
FS
GS

EAX		BE	ED	AX
EBX		1000		BX
ECX				CX
EDX				DX

ESP			SP
EBP			BP
ESI			SI
EDI			DI

| 02000 | XX |
| 02001 | XX |

| 04234 | ED |
| 04235 | XX |
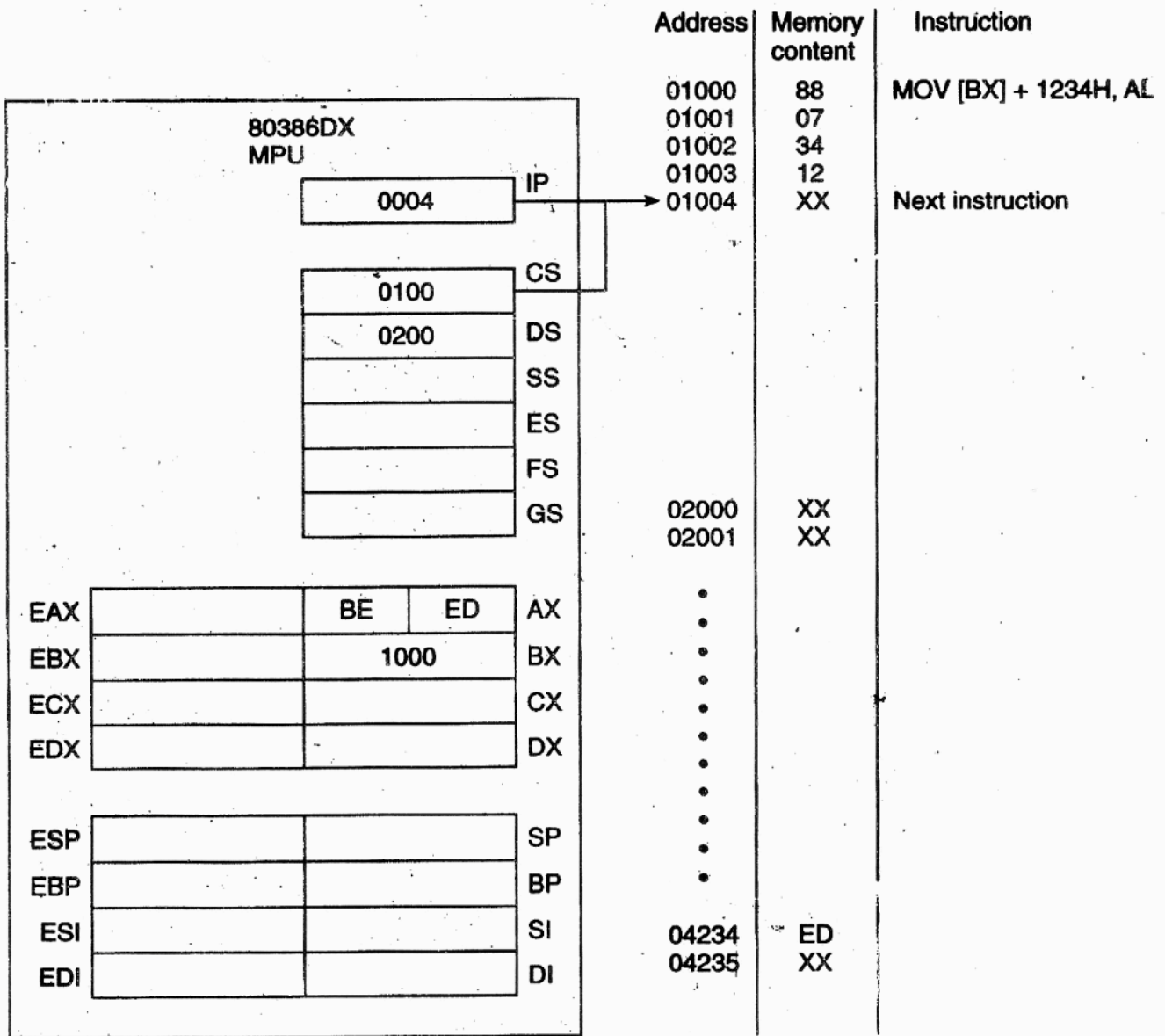
(b)

Figure 3.17 (Continued)

Based relative addressing mode

In the based relative addressing mode, base registers BX and BP, as well as a displacement value, are used to calculate what is called the effective address. The default segments used for the calculation of the physical address (PA) are DS for BX and SS for BP. For example:

```
MOV    CX,[BX]+10      ;move DS:BX+10 and DS:BX+10+1 into CX
                       ;PA = DS (shifted left) + BX + 10
```

Alternative codings are "MOV CX,[BX+10]" or "MOV CX,10[BX]". Again the low address contents will go into CL and the high address contents into CH. In the case of the BP register,

```
MOV    AL,[BP]+5       ;PA = SS (shifted left) + BP + 5
```

Again, alternative codings are "MOV AL,[BP+5]" or "MOV AL,5[BP]".

6. Indexed addressing mode
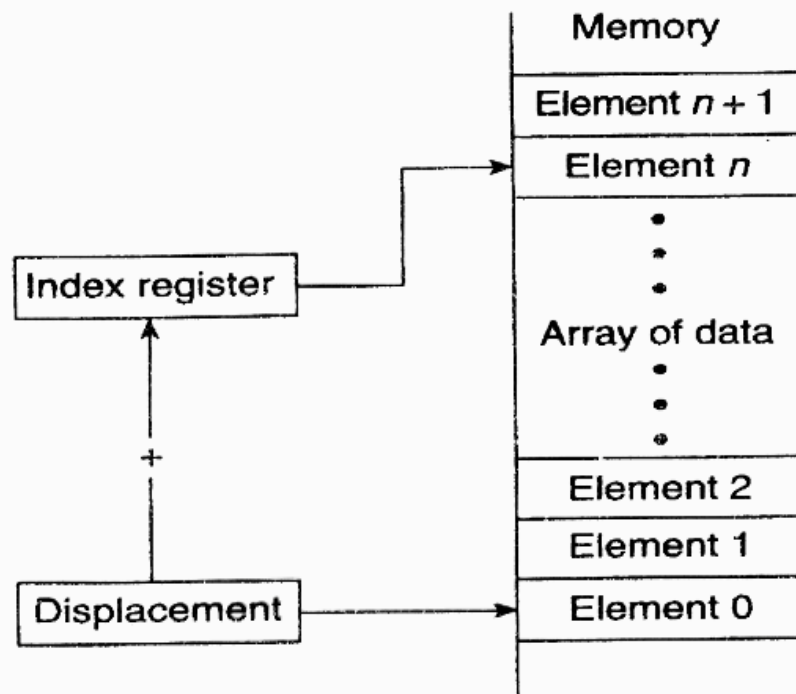
(a)

PA = Segment base: index + displacement

$$PA = \begin{Bmatrix} CS \\ DS \\ SS \\ ES \\ FS \\ GS \end{Bmatrix} : \begin{Bmatrix} SI \\ DI \end{Bmatrix} + \begin{Bmatrix} \text{8-bit displacement} \\ \text{16-bit displacement} \end{Bmatrix}$$

Figure 3.19 (a) Instruction using direct indexed addressing mode before fetch and execution. (b) After execution.

Address	Memory content	Instruction
01000	8A	MOV AL, [SI] + 1234
01001	44	
01002	34	
01003	12	
01004	XX	Next instruction
02000	XX	
02001	XX	
05234	BE	

80386DX
MPU

IP   0004

CS   0100
DS   0200
SS
ES
FS
GS

EAX   XX   BE   AX
EBX   BX
ECX   CX
EDX   DX

ESP   SP
EBP   BP
ESI   2000   SI
EDI   DI
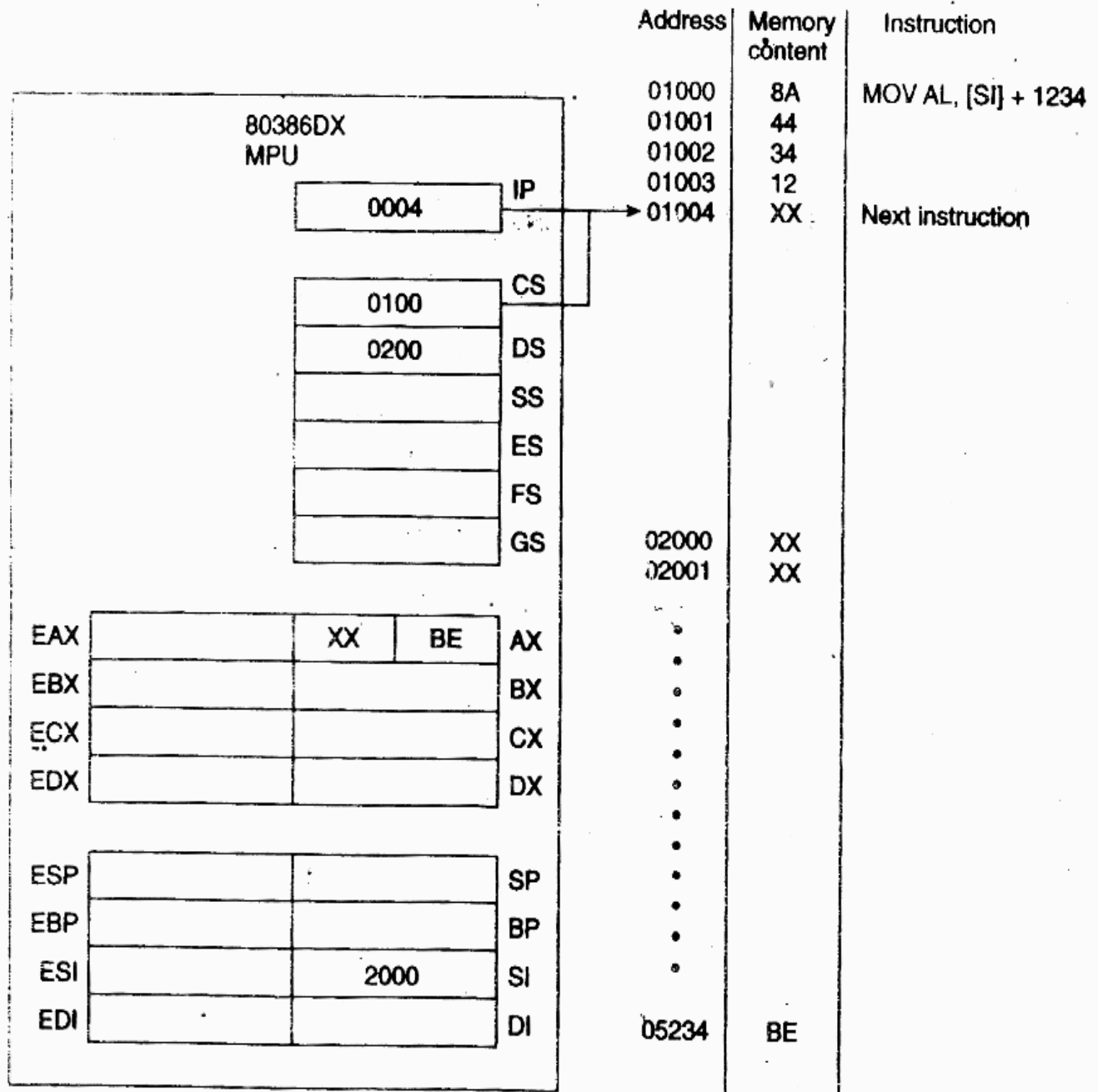
(b)

Figure 3.19   (Continued)

Example 1-17

Assume that DS = 4500, SS = 2000, BX = 2100, SI = 1486, DI = 8500, BP = 7814, and AX = 2512. Show the exact physical memory location where AX is stored in each of the following. All values are in hex.

(a) MOV [BX]+20,AX   (b) MOV [SI]+10,AX

(c) MOV [DI]+4,AX   (d) MOV [BP]+12,AX

Solution:

In each case PA = segment register (shifted left) + offset register + displacement.

(a) DS:BX+20   location 47120 = (12) and 47121 = (25)

(b) DS:SI+10   location 46496 = (12) and 46497 = (25)

(c) DS:DI+4   location 4D504 = (12) and 4D505 = (25)

(d) SS:BP+12   location 27826 = (12) and 27827 = (25)

Base relative plus index addressing.

7. Based-Indexed Addressing Mode

PA = Segment base: base + index + displacement

$$PA = \begin{Bmatrix} CS \\ DS \\ SS \\ ES \\ FS \\ GS \end{Bmatrix} : \begin{Bmatrix} BX \\ BP \end{Bmatrix} + \begin{Bmatrix} SI \\ DI \end{Bmatrix} + \begin{Bmatrix} \text{8-bit displacement} \\ \text{16-bit displacement} \end{Bmatrix}$$

(b)

Memory

EDI [| Element]

REC C

REC B

EBX [| REC C]

REC A

Displacement

[FILE]

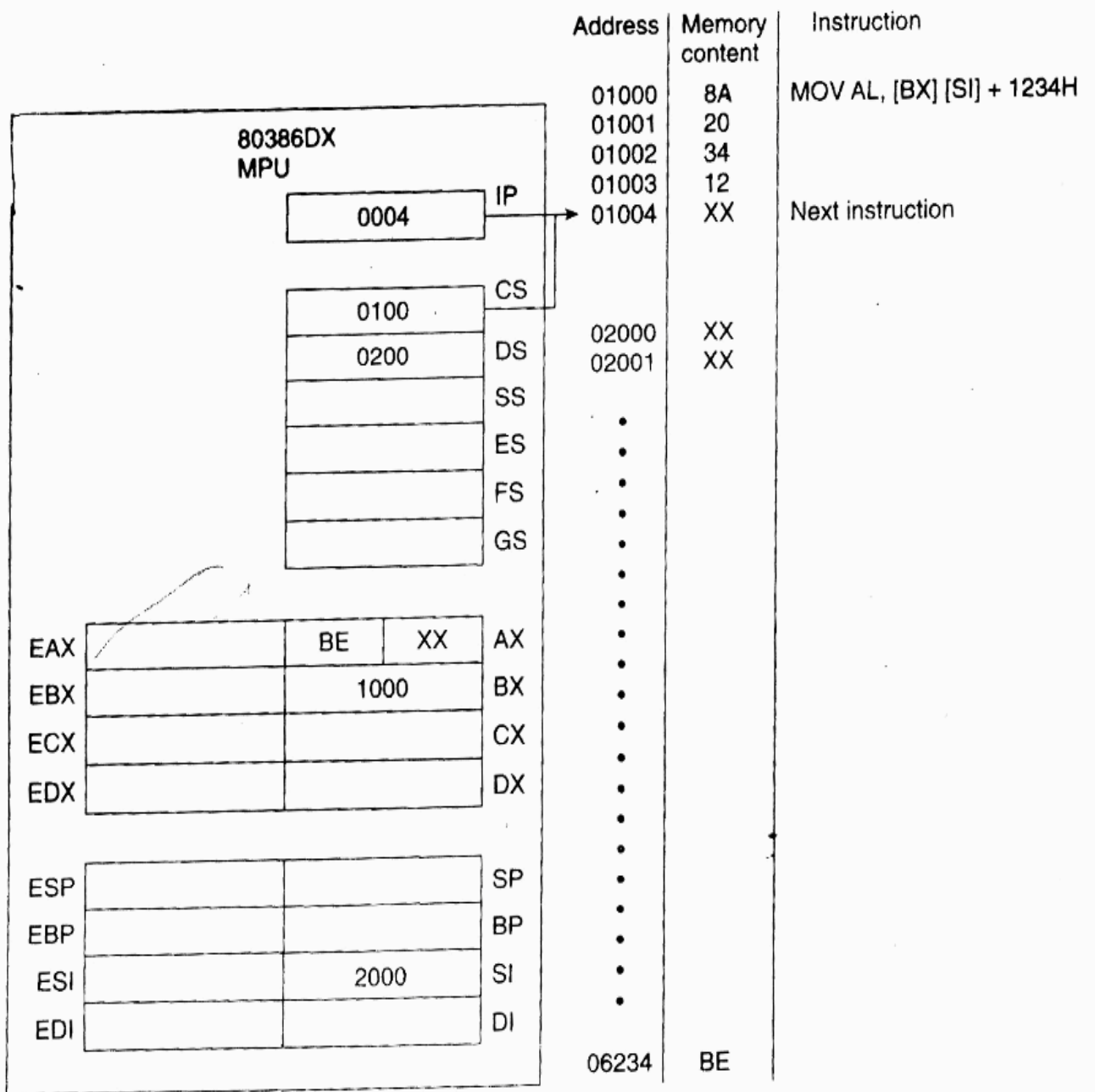| Address	Memory content	Instruction
01000	8A	MOV AH, [BX] [SI] + 1234H
01001	20	
01002	34	
01003	12	
01004	XX	Next instruction
02000	XX	
02001	XX	
06234	BE	Source operand

(a)
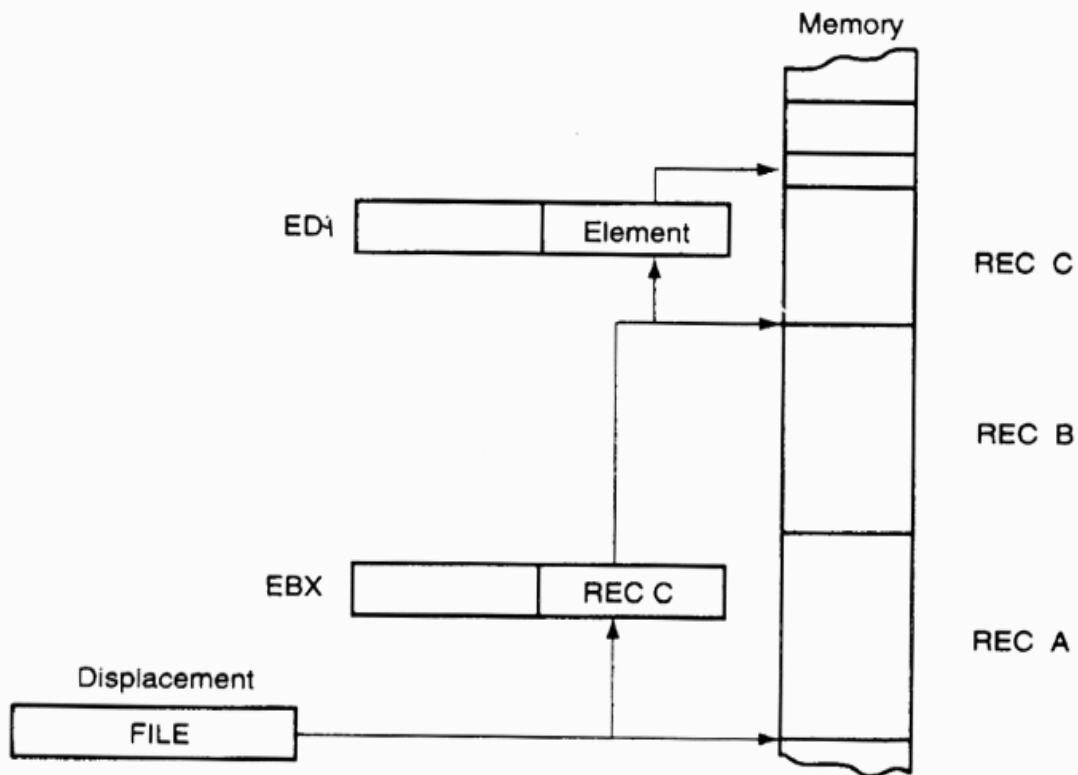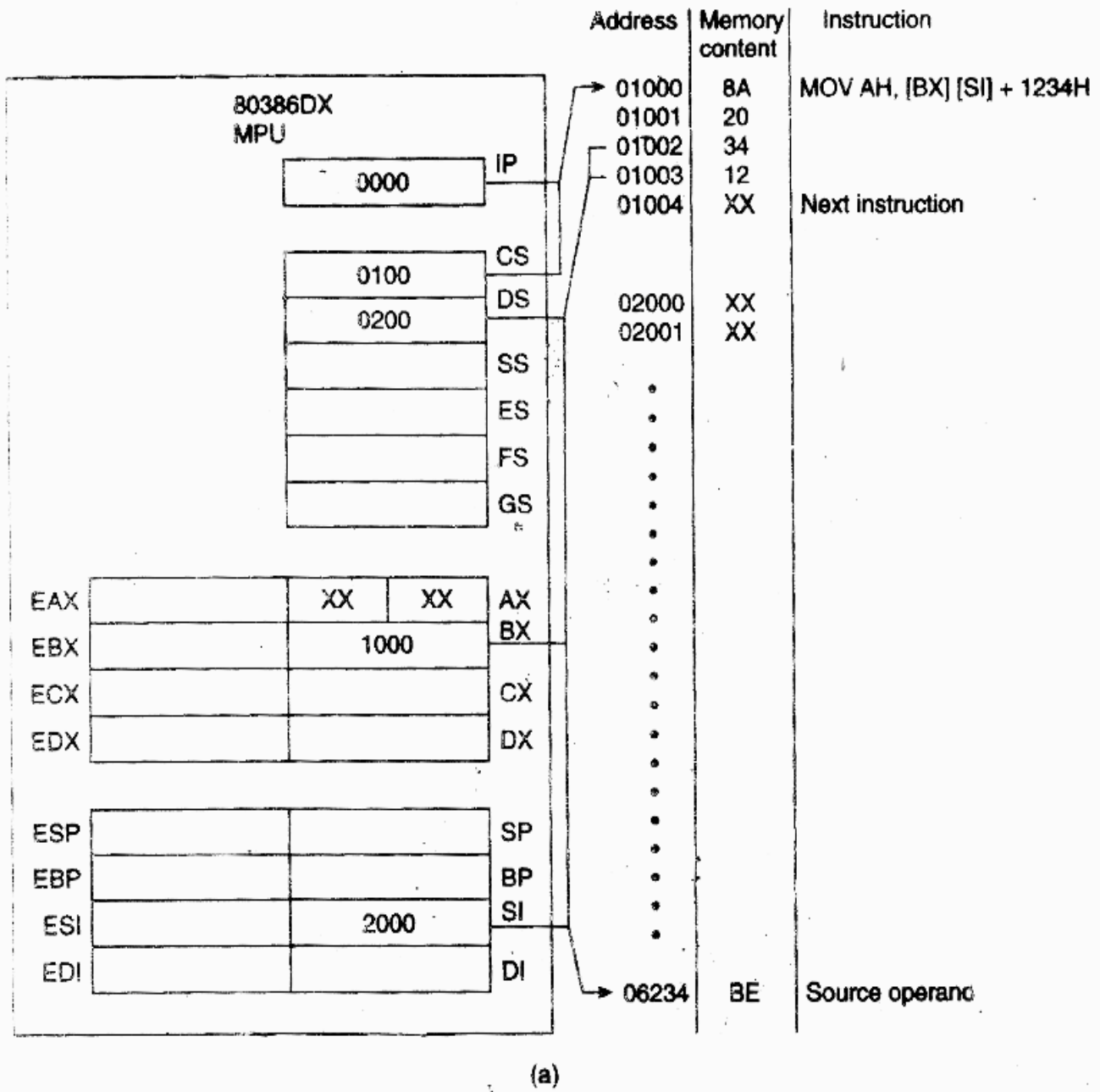
Figure 3.21 (a) Instruction using based-indexed addressing mode before fetch and execution. (b) After execution.

Address	Memory content	Instruction
01000	8A	MOV AL, [BX] [SI] + 1234H
01001	20	
01002	34	
01003	12	
01004	XX	Next instruction
02000	XX	
02001	XX	
06234	BE	

80386DX MPU

IP 0004

CS 0100
DS 0200
SS
ES
FS
GS

EAX		BE	XX	AX
EBX		1000		BX
ECX				CX
EDX				DX

ESP				SP
EBP				BP
ESI		2000		SI
EDI				DI

(b)

Figure 3.21 (Continued)

Based indexed addressing mode

By combining based and indexed addressing modes, a new addressing mode is derived called the based indexed addressing mode. In this mode, one base register and one index register are used. Examples:

```
MOV    CL,[BX][DI]+8          ;PA = DS (shifted left) + BX + DI + 8
MOV    CH,[BX][SI]+20         ;PA = DS (shifted left) + BX + SI + 20
MOV    AH,[BP][DI]+12         ;PA = SS (shifted left) + BP + DI + 12
MOV    AH,[BP][SI]+29         ;PA = SS (shifted left) + BP + SI + 29
```

The coding of the instructions above can vary; for example, the last example could have been written

```
MOV    AH,[BP+SI+29]
or
MOV    AH,[SI+BP+29]  ;the register order does not matter.
```

Note that "MOV AX,[SI][DI]+displacement" is illegal.

Example 21-4

Show how data is placed after execution of the following code.

```
        MOV   EAX,7698E39FH        ;EAX=7698E39F
        MOV   [4524],AX
        MOV   [8000],EAX
```

Solution:

For "MOV [4524],AX" we have

       DS:4524 =(9F)

       DS:4525 =(E3)

and for "MOV [8000],EAX" we have

       DS:8000 = (9F)

       DS:8001 = (E3)

       DS:8002 = (98)

       DS:8003 = (76)

8. Scaled-Index Addressing Mode

PA = Segment base: $\{$ Base + (Index × Scale factor) + Displacement $\}$

$$PA = \begin{Bmatrix} CS \\ SS \\ DS \\ ES \\ FS \\ GS \end{Bmatrix} : \begin{Bmatrix} AX \\ BX \\ CX \\ DX \\ SP \\ BP \\ SI \\ DI \end{Bmatrix} + \begin{Bmatrix} AX \\ BX \\ CX \\ DX \\ BP \\ SI \\ DI \end{Bmatrix} \times \begin{Bmatrix} 1 \\ 2 \\ 4 \\ 8 \end{Bmatrix} + \begin{Bmatrix} \text{8-bit displacement} \\ \text{16-bit displacement} \end{Bmatrix}$$

Figure 3.22 32-bit address mode physical and effective address computation for memory operands.

32-bit memory operand addressing mode

PA = Segment base: Indirect address

$$
PA = \begin{Bmatrix} CS \\ DS \\ SS \\ ES \\ FS \\ GS \end{Bmatrix} : \begin{Bmatrix} AX \\ BX \\ CX \\ DX \\ SP \\ BP \\ SI \\ DI \end{Bmatrix}
$$

(a)

PA = Segment base: Base + Displacement

$$
PA = \begin{Bmatrix} CS \\ DS \\ SS \\ ES \\ FS \\ GS \end{Bmatrix} : \begin{Bmatrix} AX \\ BX \\ CX \\ DX \\ SP \\ BP \\ SI \\ DI \end{Bmatrix} + \begin{Bmatrix} 8\text{-bit displacement} \\ 16\text{-bit displacement} \end{Bmatrix}
$$

(b)

Segment base: $\left(\text{Index} \times \dfrac{\textbf{Scale}}{\textbf{factor}} \right) + \textbf{Displacement}$

$$
\begin{Bmatrix} CS \\ DS \\ SS \\ ES \\ FS \\ GS \end{Bmatrix} : \begin{Bmatrix} AX & & 1 \\ BX & & \\ CX & & 2 \\ DX & \times & \\ BP & & 4 \\ SI & & \\ DI & & 8 \end{Bmatrix} + \begin{Bmatrix} 8\text{-bit displacement} \\ 16\text{-bit displacement} \end{Bmatrix}
$$

(c)

PA = Segment base: Base + $\left(\text{Index} \times \dfrac{\textbf{Scale}}{\textbf{factor}} \right) + \textbf{Displacement}$

$$
PA = \begin{Bmatrix} CS \\ DS \\ SS \\ ES \\ FS \\ GS \end{Bmatrix} : \begin{Bmatrix} AX \\ BX \\ CX \\ DX \\ SP \\ BP \\ SI \\ DI \end{Bmatrix} + \begin{Bmatrix} AX & & 1 \\ BX & & \\ CX & & 2 \\ DX & \times & \\ BP & & 4 \\ SI & & \\ DI & & 8 \end{Bmatrix} + \begin{Bmatrix} 8\text{-bit displacement} \\ 16\text{-bit displacement} \end{Bmatrix}
$$

(d)

Figure 3.23 (a) 32-bit mode indirect memory-address computation. (b) 32-bit mode-based memory-address computation. (c) 32-bit mode scaled-index memory-address computation. (d) 32-bit mode scaled-based-index memory-address computation.

Table 1-4: Sample Segment Overrides

Instruction	Segment Used	Default Segment
MOV AX,CS:[BP]	CS:BP	SS:BP
MOV DX,SS:[SI]	SS:SI	DS:SI
MOV AX,DS:[BP]	DS:BP	SS:BP
MOV CX,ES:[BX]+12	ES:BX+12	DS:BX+12
MOV SS:[BX][DI]+32,AX	SS:BX+DI+32	DS:BX+DI+32

Table 1-5: Summary of 80x86 Addressing Modes

Addressing Mode	Operand	Default Segment
Register	reg	none
Immediate	data	none
Direct	[offset]	DS
Register indirect	[BX]	DS
	[SI]	DS
	[DI]	DS
Based relative	[BX]+disp	DS
	[BP]+disp	SS
Indexed relative	[DI]+disp	DS
	[SI]+disp	DS
Based indexed relative	[BX][SI]+disp	DS
	[BX][DI]+disp	DS
	[BP][SI]+ disp	SS
	[BP][DI]+ disp	SS

Table 8-2: Addressing Modes for the 80386/486

Addressing Mode	Operand	Default Segment
Register	register	none
Immediate	data	none
Direct	[OFFSET]	DS
Register indirect	[BX]	DS
	[SI]	DS
	[DI]	DS
	[EAX]	DS
	[EBX]	DS
	[ECX]	DS
	[EDX]	DS
	[ESI]	DS
	[EDI]	DS
Based relative	[BX]+disp	DS
	[BP]+disp	SS
	[EAX]+disp	DS
	[EBX]+disp	DS
	[ECX]+disp	DS
	[EDX]+disp	DS
	[EBP]+disp	SS
Indexed relative	[DI]+disp	DS
	[SI]+disp	DS
	[EDI]+disp	DS
	[ESI]+disp	DS
Based indexed relative	[R1][R2]+disp where R1 and R2 are any of the above	If BP is used, segment is SS; otherwise, DS is the segment

Note: In based indexed relative addressing, disp is optional.

Example 21-5

Find the effective address in each of the following cases. Assume that ESI = 200H, ECX = 100H, EBX = 50H, and EDI = 100H.

(a) MOV AX,[2000+ESI*4]   (b) MOV AX,[5000+ECX*2]
(c) MOV ECX,[2400+EBX*4]   (d) MOV DX,[100+EDI*8]

Solution:

(a) EA (effective address) is $2000H + 200H \times 4 = 2000 + 800H = 2800H$. Therefore, the logical address of the operand moved into AX is DS:2800H.

(b) By the same token we have $EA = 5000H + 100H \times 2 = 5000H + 200 = 5200H$.

(c) $EA = 2400H + 4 \times 50H = 2400H + 140H = 2540H$.

(d) $100H + 8 \times 100H = 100H + 800H = 900H$.