CSec15233
Malicious Software Analysis

Addressing Modes

Qasem Abu Al-Haija, PhD

8086 Addressing Modes

- Instructions operate on a data➔ Operands may be contained in :
  - Registers, Immediate (Instruction op-code), Memory Locations, I/O ports.
  - These different sources of operands are known as addressing modes.

- 8086 has 12 modes to access operands, classified into 5 groups:
  - Register and immediate addressing modes (two modes) .
  - Data Memory addressing modes (six modes) .
  - Port addressing mode (two modes) .
  - Relative addressing mode (one mode) .
  - Implied addressing mode (one mode) .

8086 Addressing Modes

1. **Register Addressing**
2. **Immediate Addressing**

Group I : Addressing modes for register and immediate data

3. **Direct Addressing**
4. **Register Indirect Addressing**
5. **Based Addressing**
6. **Indexed Addressing**
7. **Based Index Addressing**
8. **String Addressing**

Group II : Addressing modes for memory data

9. **Direct I/O port Addressing**
10. **Indirect I/O port Addressing**

Group III : Addressing modes for I/O ports

11. **Relative Addressing**

Group IV : Relative Addressing mode

12. **Implied Addressing**

Group V : Implied Addressing mode

8086 Addressing Modes

Group I : Addressing modes for register and immediate data

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

The instruction will specify the name of the register which holds the data to be operated by the instruction.

Example:

MOV CL, DH

The content of 8-bit register DH is moved to another 8-bit register CL

(CL) ← (DH)

Examples:

MOV CX,BX ; **Move content of BX to CX**

ADD CL,BL ; **Add content of CL and BL and store result in CL**

ADC BX,DX ; **Add content of BX, carry flag and DX, and store result in BX**

8086 Addressing Modes

Group I : Addressing modes for register and immediate data

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

In immediate addressing mode, an 8-bit or 16-bit data is specified as part of the instruction

Example:

MOV DL, 08H

The 8-bit data (08_H) given in the instruction is moved to DL

(DL) ← 08_H

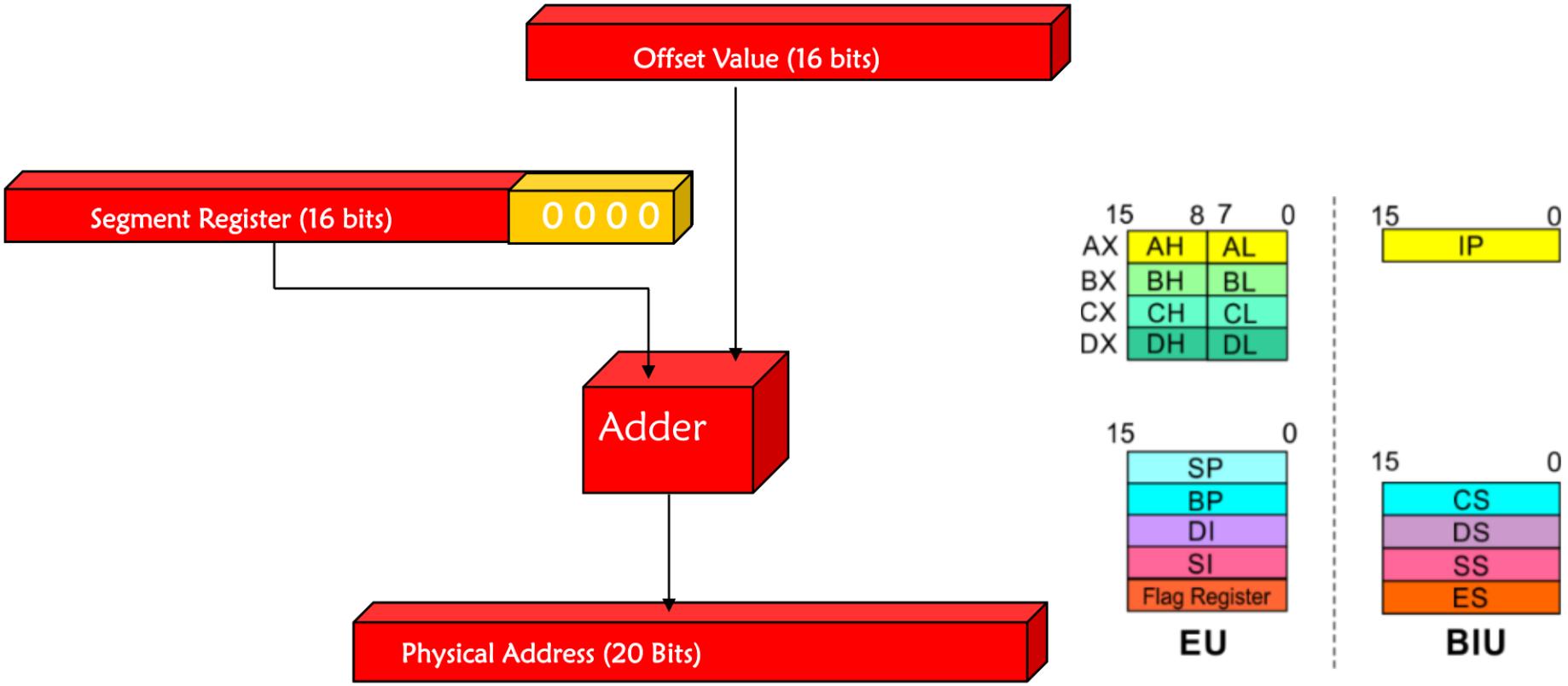
MOV AX, 0A9FH

The 16-bit data ($0A9F_H$) given in the instruction is moved to AX register

(AX) ← $0A9F_H$

8086 Addressing Modes

Group II : Addressing modes for Memory Data



8086 Addressing Modes

Group II : Addressing modes for Memory Data

- 20 Address lines ⇒ 8086 can address up to 2^{20} = 1M bytes of memory
- However, the largest register is only 16 bits.
- Physical Address will have to be calculated. Physical Address : Actual address of a byte in memory. i.e. the value which goes out onto the address bus.
- Memory Address represented in the form – Seg : Offset (E.g. 89AB:F012)
- Each time the processor wants to access memory, it takes the contents of a segment register, shifts it one hexadecimal place to the left (same as multiplying by 16_{10}), then add the required offset to form the 20- bit address.
- The term Effective Address (EA) represents the offset address of the data within a segment which is obtained by different methods, depending upon the addressing mode that is used in the instruction.
- Let us assume that the various registers in 8086 have the following values stored in them.

89AB : F012 → 89AB → 89AB0 (Paragraph to byte → 89AB x 10 = 89AB0)
F012 → 0F012 (Offset is already in byte unit)
+ -------
98AC2 (The absolute address)

16 bytes of contiguous memory

8086 Addressing Modes

Group II : Addressing modes for Memory Data

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

Here, the effective address of the memory location at which the data operand is stored is given in the instruction.

The effective address is just a 16-bit number written directly in the instruction.

Example:

```
MOV  BX, [1354H]
MOV  BL, [0400H]
```

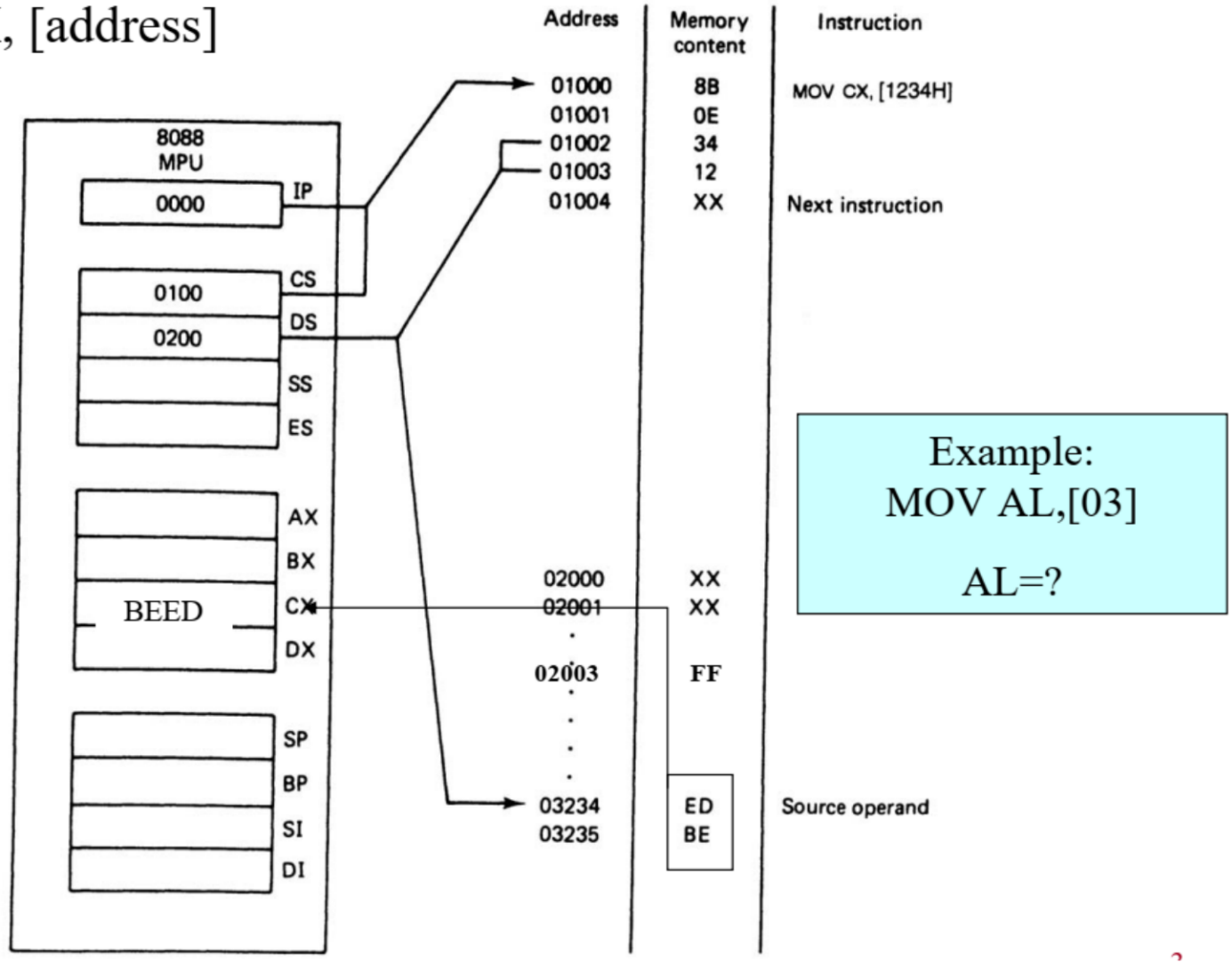
The square brackets around the 1354_H denotes the contents of the memory location. When executed, this instruction will copy the contents of the memory location into BX register.

This addressing mode is called direct because the displacement of the operand from the segment base is specified directly in the instruction. (its taken with DS as segment register)

8086 Addressing Modes

Group II : Addressing modes for Memory Data

MOV CX, [address]



8086 Addressing Modes

Group II : Addressing modes for Memory Data

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

In Register indirect addressing, name of the register which holds the effective address (EA) will be specified in the instruction.

Registers used to hold EA are any of the following registers:

BX, BP, DI and SI.

Content of the DS register is used for base address calculation.

Example:

MOV CX, [BX]

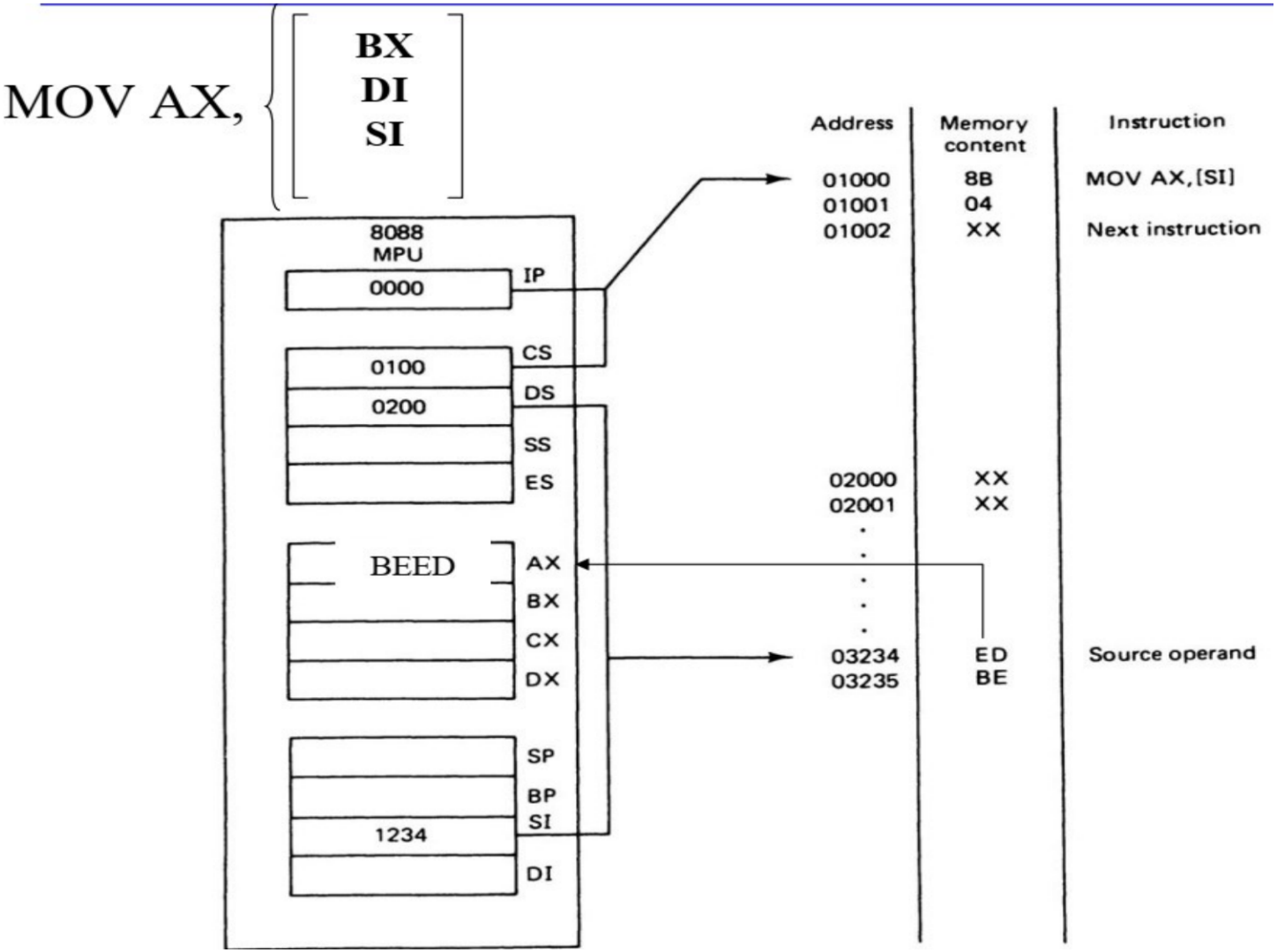
Note : Register/ memory enclosed in brackets refer to content of register/ memory

Operations:

EA = (BX)
BA = (DS) x 16_{10}
MA = BA + EA
(CX) ← (MA) or,
(CL) ← (MA)
(CH) ← (MA +1)

8086 Addressing Modes

Group II : Addressing modes for Memory Data



8086 Addressing Modes

Group II : Addressing modes for Memory Data

- Assume that DS=1120, SI=2498 and AX=17FE show the memory locations after the execution of:

MOV [SI],AX

DS (Shifted Left) + SI = 13698.

With little endian convention:

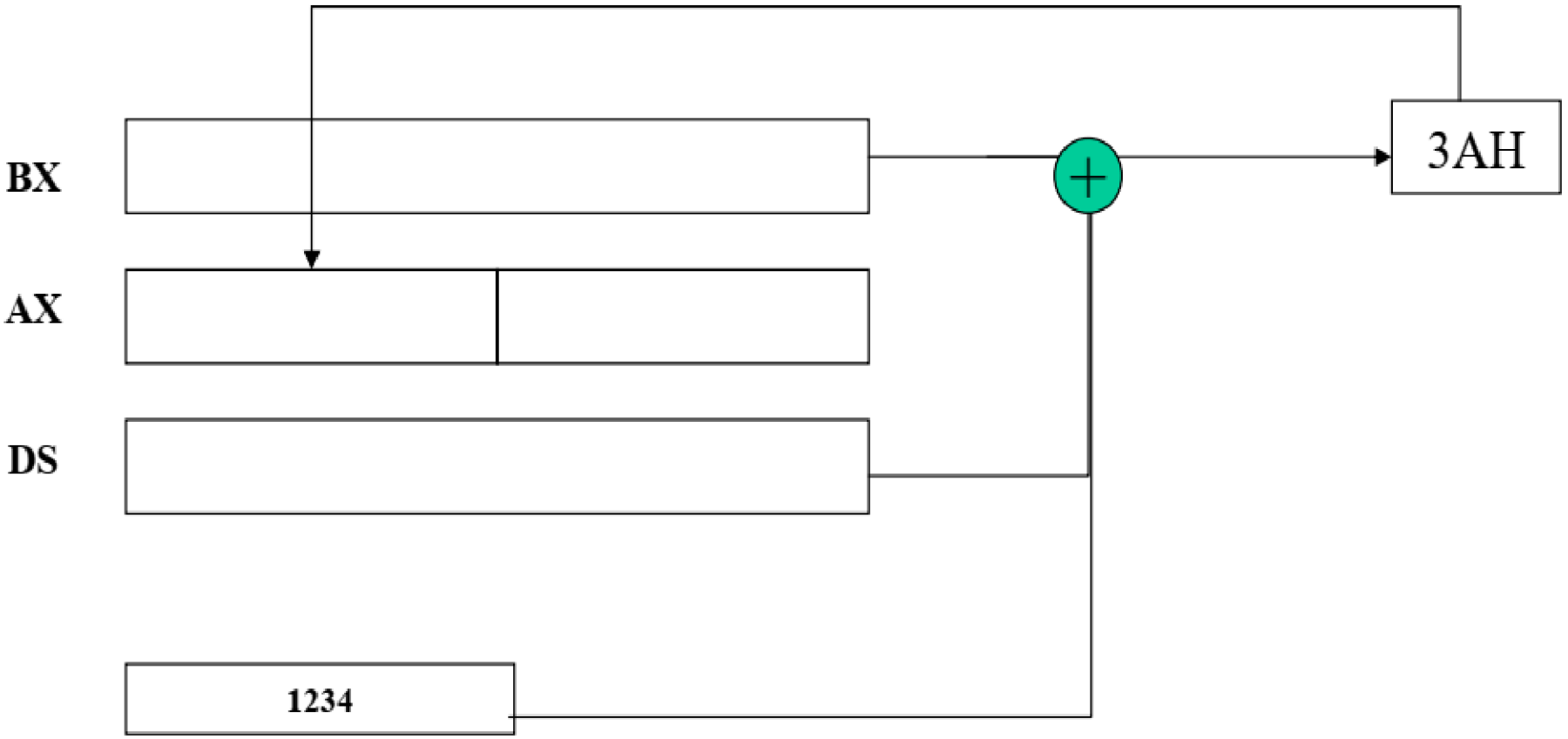
Low address 13698 → FE

High Address 13699 → 17

8086 Addressing Modes

Group II : Addressing modes for Memory Data

MOV AH, [DS:BX / SS:BP] + 1234h



8086 Addressing Modes

Group II : Addressing modes for Memory Data

- To access memory we use these four registers: BX, SI, DI, BP
- Combining these registers inside [ ] symbols, we can get different memory locations (Effective Address, EA)
- Supported combinations:

[BX + SI] [BX + DI] [BP + SI] [BP + DI]	[SI] [DI] d16 (variable offset only) [BX]	[BX + SI + d8] [BX + DI + d8] [BP + SI + d8] [BP + DI + d8]
[SI + d8] [DI + d8] [BP + d8] [BX + d8]	[BX + SI + d16] [BX + DI + d16] [BP + SI + d16] [BP + DI + d16]	[SI + d16] [DI + d16] [BP + d16] [BX + d16]

BX BP	SI DI	+ disp

8086 Addressing Modes

Group II : Addressing modes for Memory Data

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

In Based Addressing, BX or BP is used to hold the base value for effective address and a signed 8-bit or unsigned 16-bit displacement will be specified in the instruction.

In case of 8-bit displacement, it is sign extended to 16-bit before adding to the base value.

When BX holds the base value of EA, 20-bit physical address is calculated from BX and DS.

When BP holds the base value of EA, BP and SS is used.

Example:

MOV AX, [BX + 08H]

Operations:

$0008_H \leftarrow 08_H$ (Sign extended)
EA = (BX) + 0008_H
BA = (DS) x 16_{10}
MA = BA + EA
(AX) ← (MA) or,
(AL) ← (MA)
(AH) ← (MA + 1)

8086 Addressing Modes

Group II : Addressing modes for Memory Data

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

SI or DI register is used to hold an index value for memory data and a signed 8-bit or unsigned 16-bit displacement will be specified in the instruction.

Displacement is added to the index value in SI or DI register to obtain the EA.

In case of 8-bit displacement, it is sign extended to 16-bit before adding to the base value.

Example:

MOV CX, [SI + 0A2H]

Operations:

$FFA2_H \leftarrow A2_H$ **(Sign extended)**

$EA = (SI) + FFA2_H$
$BA = (DS) \times 16_{10}$
$MA = BA + EA$

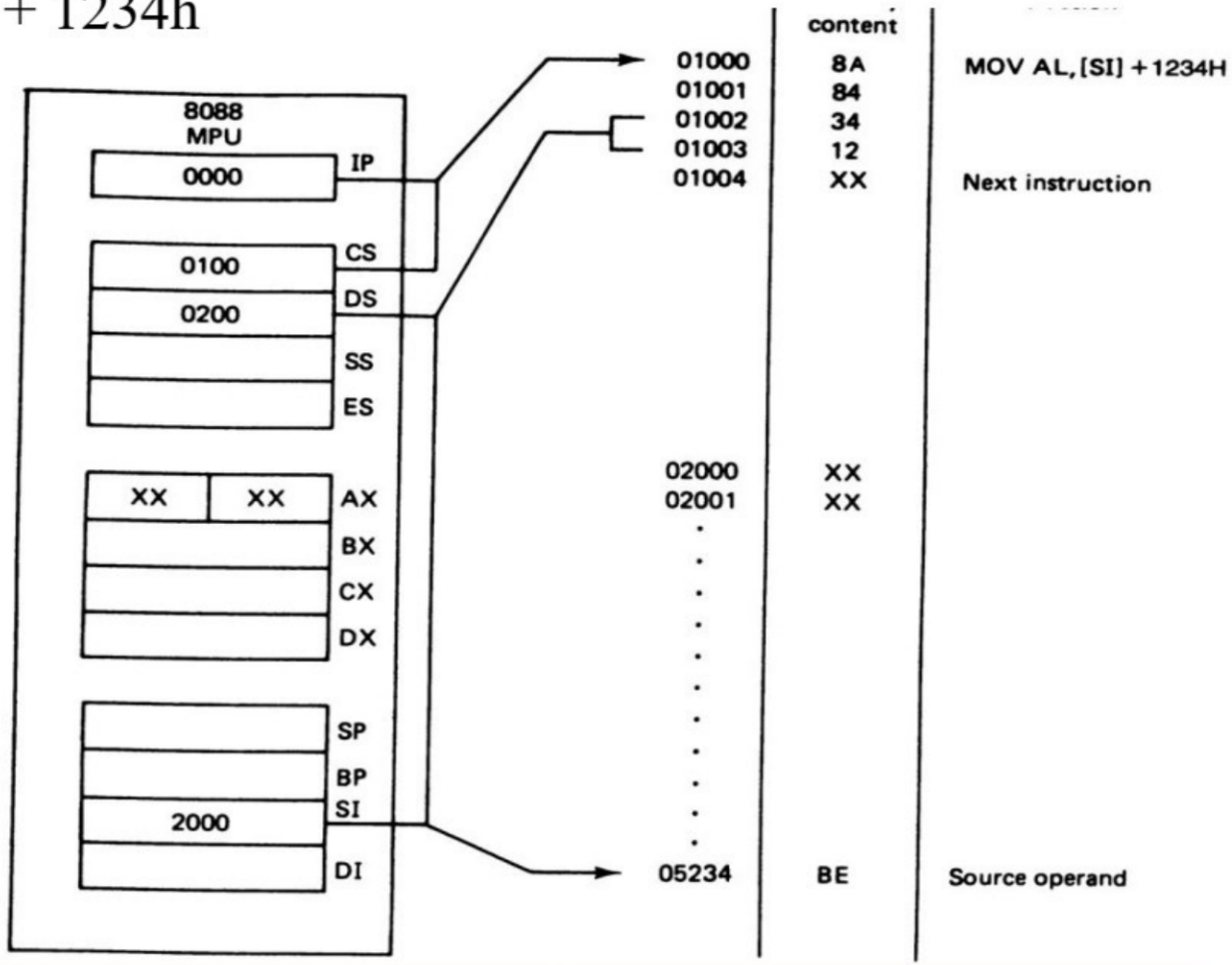
$(CX) \leftarrow (MA)$ **or,**

$(CL) \leftarrow (MA)$
$(CH) \leftarrow (MA + 1)$

8086 Addressing Modes

Group II : Addressing modes for Memory Data

MOV AH, [$\frac{SI}{DI}$] + 1234h



Example: What is the physical address MOV [DI-8],BL if DS=200 & DI=30h ?
DS:200 shift left once 2000 + DI + -8 = 2028

8086 Addressing Modes

Group II : Addressing modes for Memory Data

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

In Based Index Addressing, the effective address is computed from the sum of a base register (BX or BP), an index register (SI or DI) and a displacement.

Example:

MOV DX, [BX + SI + 0AH]

Operations:

$000A_H \leftarrow 0A_H$ **(Sign extended)**

$EA = (BX) + (SI) + 000A_H$
$BA = (DS) \times 16_{10}$
$MA = BA + EA$

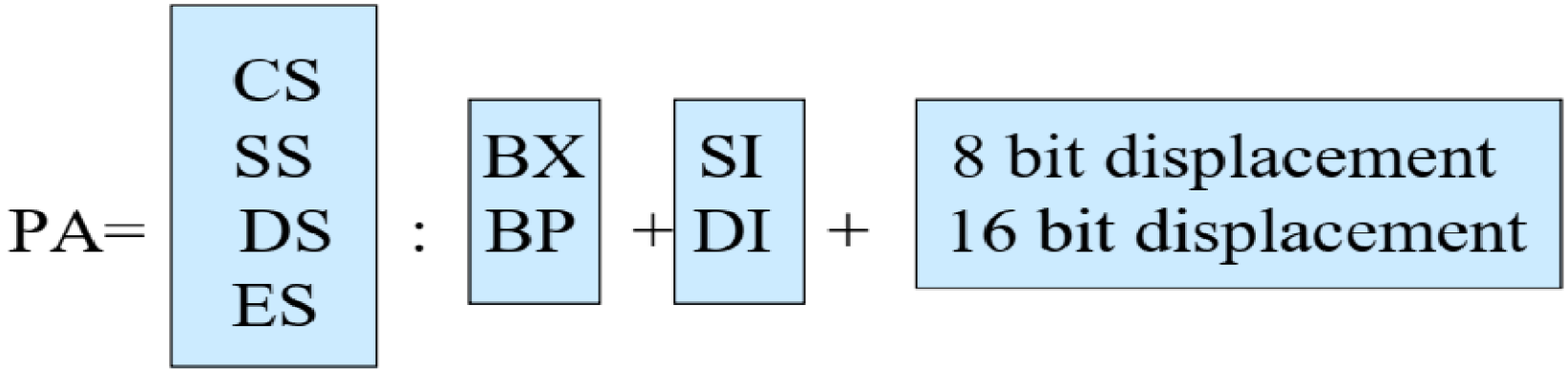
$(DX) \leftarrow (MA)$ **or,**

$(DL) \leftarrow (MA)$
$(DH) \leftarrow (MA + 1)$

8086 Addressing Modes

Group II : Addressing modes for Memory Data

- Based Relative + Indexed Relative
- We must calculate the PA (physical address)



MOV AH,[BP+SI+29]
or
MOV AH,[SI+29+BP]
or
MOV AH,[SI][BP]+29

} The register order does not matter

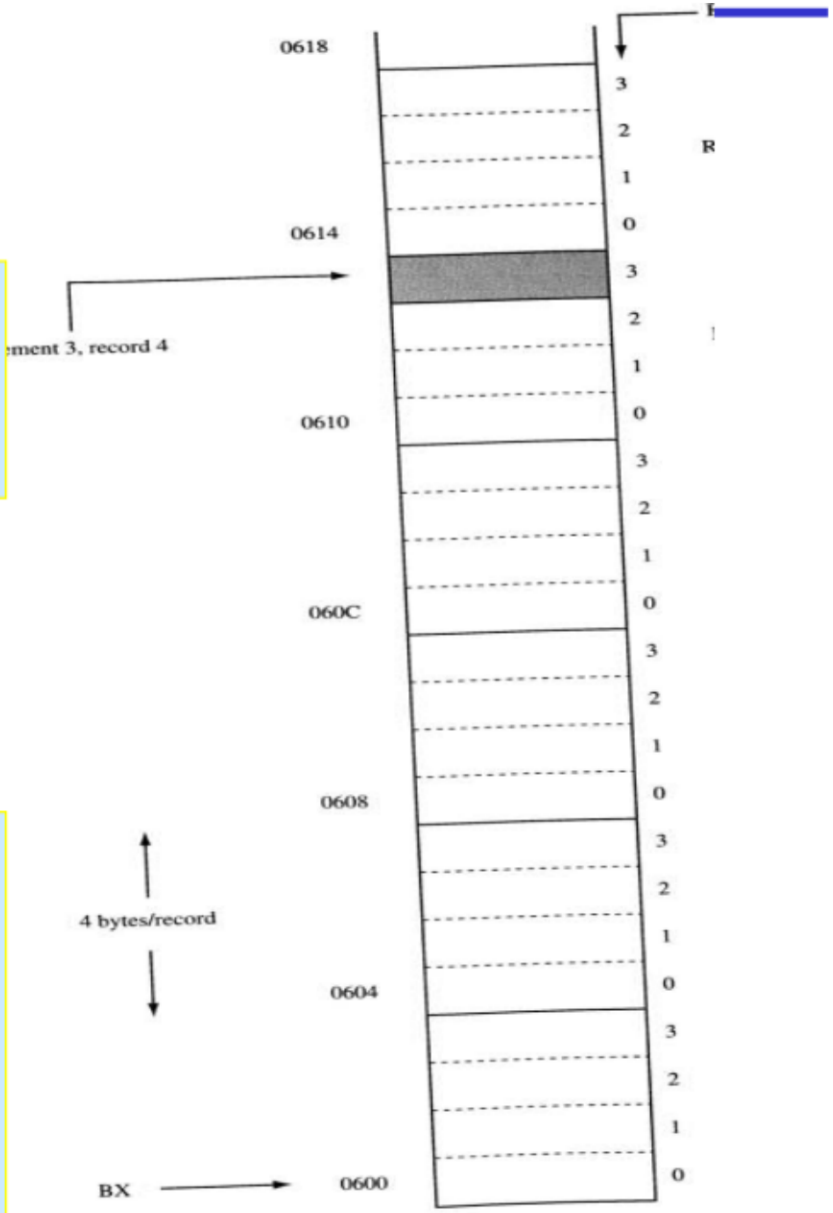
8086 Addressing Modes

ure 4–4.
e base + index +
placement
dressing mode can
used to access a
rticular element in a
rticular record of an
ray.

```
MOV BX, 0600h
MOV SI, 0010h ; 4 records, 4 elements each.
MOV AL, [BX + SI + 3]
```

OR

```
MOV BX, 0600h
MOV AX, 004h ;
MOV CX,04;
MUL CX
MOV SI, AX
MOV AL, [BX + SI + 3]
```



8086 Addressing Modes

Group II : Addressing modes for Memory Data

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

Note : Effective address of the Extra segment register

Employed in string operations to operate on string data.

The effective address (EA) of source data is stored in SI register and the EA of destination is stored in DI register.

Segment register for calculating base address of source data is DS and that of the destination data is ES

Example: MOVS BYTE

Operations:

Calculation of source memory location:

EA = (SI) BA = (DS) x 16_{10} MA = BA + EA

Calculation of destination memory location:

EA_E = (DI) BA_E = (ES) x 16_{10} $MA_E = BA_E + EA_E$

$(MA_E) \leftarrow (MA)$

If DF = 1, then (SI) ← (SI) – 1 and (DI) = (DI) - 1
If DF = 0, then (SI) ← (SI) +1 and (DI) = (DI) + 1

8086 Addressing Modes

Group II : Addressing modes for Memory Data

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

Example : MOVS WORD.

If (DF)=0, (DS)=3000_{I6}, (SI)=0020_{I6},
(ES)=5000_{I6},(DI)= 0040_{I6},
(30020)=30_{I6}, (30021)= 05_{I6},
(50040)=06_{I6}, and (50041)=20_{I6},

then, after this MOVS:

(50040)=30_{I6}, (50041)=05_{I6},
(SI)=0022_{I6}, and (DI)=0042_{I6}

8086 Addressing Modes

Group III : Addressing modes for I/O Ports

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

These addressing modes are used to access data from standard I/O mapped devices or ports.

In direct port addressing mode, an 8-bit port address is directly specified in the instruction.

Example: **IN AL, [09H]**

Operations: $PORT_{addr} = 09_H$
$(AL) \leftarrow (PORT)$

Content of port with address 09_H is moved to AL register

In indirect port addressing mode, the instruction will specify the name of the register which holds the port address. In 8086, the 16-bit port address is stored in the DX register.

Example: **OUT [DX], AX**

Operations: $PORT_{addr} = (DX)$
$(PORT) \leftarrow (AX)$

The content of AX is moved to the port whose address is specified by the DX register.

8086 Addressing Modes

Group IV : Relative Addressing mode

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

In this addressing mode, the effective address of a program instruction is specified relative to Instruction Pointer (IP) by an 8-bit signed displacement.

Example: JZ 0AH

Operations:

$000A_H \leftarrow 0A_H$ **(sign extend)**

If ZF = 1, then

$EA = (IP) + 000A_H$
$BA = (CS) \times 16_{10}$
$MA = BA + EA$

If ZF = 1, then the program control jumps to new address calculated above.

If ZF = 0, then next instruction of the program is executed.

8086 Addressing Modes

Group V : Implied Addressing mode

1. Register Addressing
2. Immediate Addressing
3. Direct Addressing
4. Register Indirect Addressing
5. Based Addressing
6. Indexed Addressing
7. Based Index Addressing
8. String Addressing
9. Direct I/O port Addressing
10. Indirect I/O port Addressing
11. Relative Addressing
12. Implied Addressing

Instructions using this mode have no operands. The instruction itself will specify the data to be operated by the instruction.

Example: **CLC**

This clears the carry flag to zero.

More Examples

Example 1: MOV CL,[BX], Register Indirect Addressing

EA=(BX)=2000H, Assume DS = 3000H.

Memory address=DSx10H+(BX)=32000H. The byte from the memory address 32000H is read and stored in CL.

Example 2: MOV CH,[BX-100H], Based Addressing

EA=(BX)-100H, Assume DS = 3000H.

Memory address=DSx10H+(BX)-100H = 30000H+2000H-100H=31F00H

The byte from the memory address 31F00H is taken and stored in CH.

Example 3: MOV CX,[DI], Register Indirect Addressing

EA=(DI)=3000H, Assume DS = 3000H.

Memory address=DSx10H+(DI)=30000H+3000H=33000H

A word from the memory address 33000H is taken and stored in CX.

Example 4: MOV CL,[DI+10H], Indexed Addressing

EA=(DI)+10H, Assume DS = 3000H.

Memory address=DSx10H+(DI)+10H =30000H+3000H+10H = 33010H

A byte from the memory address 33010H is taken and stored in CL.

Example 5: MOV AX,[BX+SI], Based Index Addressing

EA=(BX)+(SI), Assume DS = 3000H.

Memory address=DSx10H+(BX)+(SI)=30000H+2000H+1000H =33000H

Example 6: MOV CX,[BX+SI+50H], Based Index Addressing

EA= (BX)+(SI)+50H, Assume DS = 3000H.

Memory address=DSx10H+(BX)+(SI)+50H=30000H+2000H+1000H+50H=33050H

Summary of the addressing modes

Addressing Mode	Operand	Default Segment
Register	Reg	None
Immediate	Data	None
Direct	[offset]	DS
Register Indirect	[BX] [SI] [DI]	DS DS DS
Based Relative	[BX]+disp [BP]+disp	DS SS
Indexed Relative	[DI]+disp [SI]+disp	DS DS
Based Indexed Relative	[BX][SI or DI]+disp [BP][SI or DI]+disp	DS SS

16 bit Segment Register Assignments

Type of Memory Reference	Default Segment	Alternate Segment	Offset
Instruction Fetch	CS	none	IP
Stack Operations	SS	none	SP,BP
General Data	DS	CS,ES,SS	BX, address
String Source	DS	CS,ES,SS	SI, DI, address
String Destination	ES	None	DI

Example for default segments

- The following registers are used as offsets. Assuming that the default segment used to get the logical address, give the segment register associated?

a) BP b)DI c)IP d)SI, e)SP, f) BX

- Show the contents of the related memory locations after the execution of this instruction

  MOV [BP][SI]+10,DX

  if DS=2000, SS=3000,CS=1000,SI=4000,BP=7000,DX=1299 (all hex)

Thank you