CSE2006
Microprocessor & Interfacing

Module – 1
Introduction to 8086 Microprocessor

Dr. E. Konguvel
Assistant Professor (Sr. Gr. 1),
Dept. of Embedded Technology,
School of Electronics Engineering (SENSE),
konguvel.e@vit.ac.in
9597812810

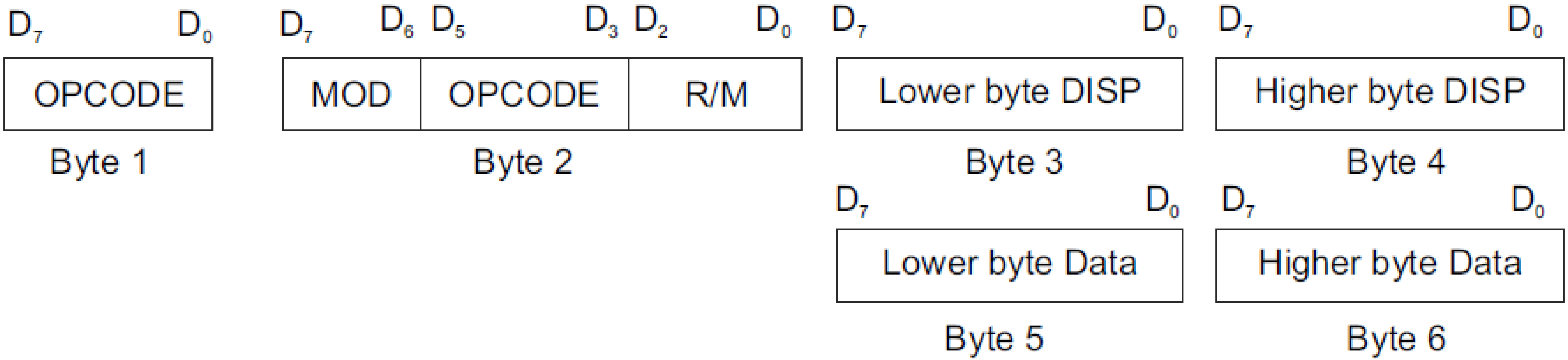
VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Addressing Modes

- An ***instruction:*** command given to a microprocessor to perform a specified operation with given data.
- Instruction = Opcode + Operand
- The addressing mode is used to locate the operand or data.
- Different types of addressing modes depends on the location of data in the 8086 processor.
- The operand can specify a register or a memory location in any one of the memory segments or I/O ports.

Addressing Modes

General Instruction Format



D_7 OPCODE D_0

Byte 1

D_7 MOD D_6 D_5 OPCODE D_3 D_2 R/M D_0

Byte 2

D_7 Lower byte DISP D_0

Byte 3

D_7 Higher byte DISP D_0

Byte 4

D_7 Lower byte Data D_0

Byte 5

D_7 Higher byte Data D_0

Byte 6

Addressing Modes

Types of Addressing Modes

1. Immediate addressing
2. Register addressing
3. Memory addressing
4. Branch addressing

Addressing Modes

1. Immediate Addressing

- 8-bit or 16-bit operand (data) is a part of the instruction.

Ex.: MOV AX, 4000H.

Data 4000H be loaded to the AX register immediately.

MOV BX, 7000H; load 7000H in BX register

MOV CX, 4500H; store 4500H in CX register

Addressing Modes

2. Register Addressing

- The data is in the register specified by the instruction.

  MOV Destination, Source

  Data copied from Source register to Destination register.

- Registers (General Purpose) used: 8-bit (AL, AH, BL, BH, CL, CH, DL, DH) & 16-bit (AX, BX, CX, DX, SI, DI, SP, BP).

- Both operands must be of the same length.

  Ex.:

  MOV AL, BL; Copies the value of BL into AL

  MOV AX, BX; Copies the contents of BX into AX

Addressing Modes

3. Memory Addressing

- Memory addressing requires determination of physical address.
- The 16-bit effective address can be made up of base, index and displacement.
- The basic formula for the 16-bit effective address (EA) and the 20-bit physical address (PA) is given below:
- 16-bit EA = Base + Index + Displacement
- 20-bit PA = Segment × 10 + Base + Index + Displacement

Addressing Modes

3. Memory Addressing

Memory addressing has the following combinations:

a. Direct addressing
b. Register indirect addressing
c. Based addressing
d. Indexed addressing
e. Based Indexed addressing
f. Based Indexed with displacement addressing

Addressing Modes

3a. Direct Addressing

- Most common addressing mode, Instruction operand specifies the memory address where data is located.
- The displacement-only addressing mode consists of an 8-bit or 16-bit constant that specifies the offset of the actual address of the target memory location.
- Ex.: MOV AX, [5000]

  copies 2 bytes of data starting from memory location DS × 10 + 5000H to the AX register. Lower byte: DS × 10 + 5000H and Higher byte: DS × 10 + 5001H.
- Ex.: MOV AL, DS: [5000H]

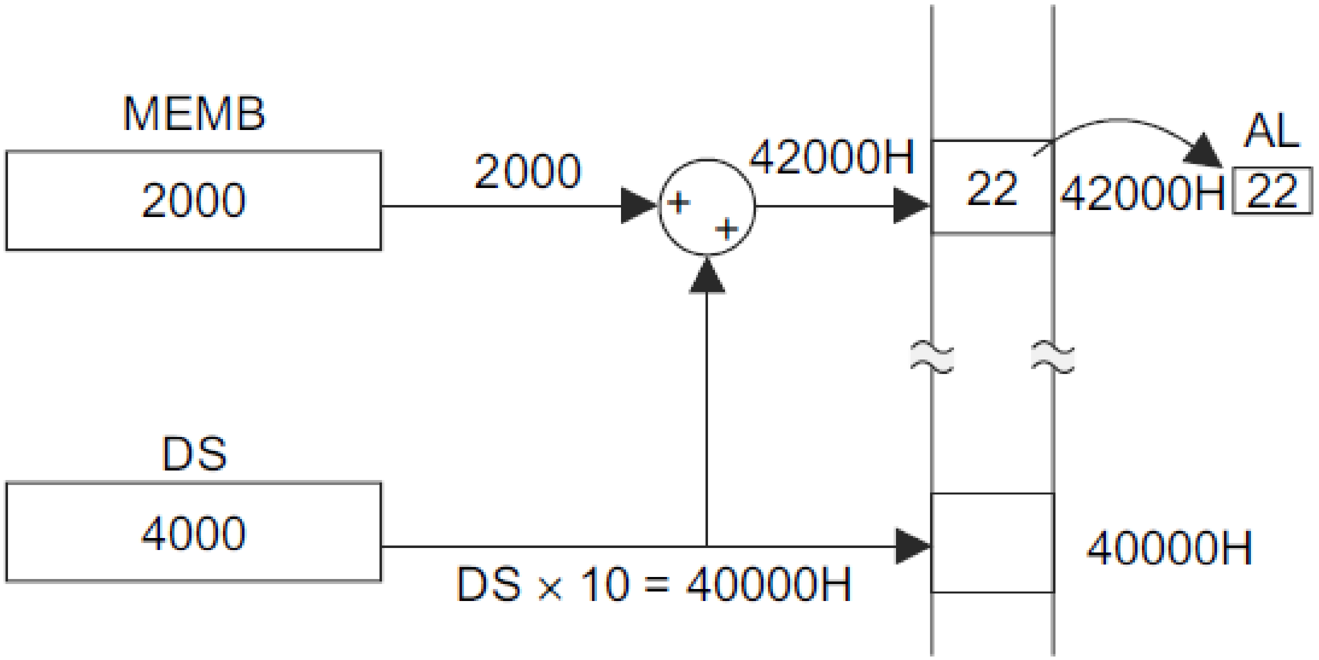
  content of the memory location DS × 10 + 5000H loads into the AL register.

Addressing Modes

3a. Direct Addressing

- MOV DS:[2000H], AL

  Content of the AL register will move to memory location DS × 10 + [2000H]



Addressing Modes

3b. Register Indirect Addressing

- Instruction specifies a register containing an address, where data is located.
- The effective address of the data is in the base register BX or an index register that is specified by the instruction.
- Works with index registers SI, DI, and base registers BX and BP registers.
- Ex.:

```
MOV AL, [BX]
MOV AH, [DI]
MOV AL, [SI]
MOV AH, [BP]
```

Addressing Modes

3b. Register Indirect Addressing

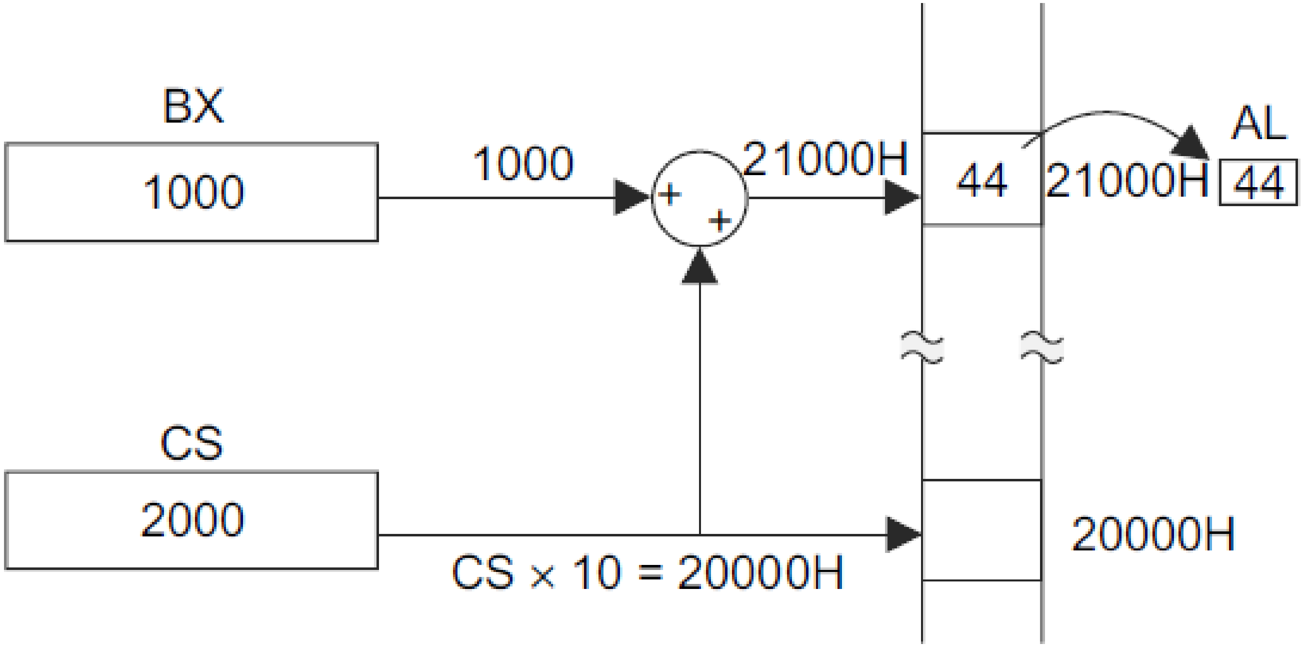
- The BX, BP, SI, or DI registers are using the DS segment by default.
- The base pointer uses stack segment by default.
- Segment Override Instructions

```
MOV AL, CS: [BX]
MOV AL, DS: [BP]
MOV AL, SS: [SI]
MOV AL, ES: [DI]
```

Addressing Modes

3b. Register Indirect Addressing

- BX = 1000H, CS = 2000. PA = 2000×10 + 1000 = 21000H.
- MOV AL, [BX] - contents of the memory location 21000H is 44H which will be stored in the AL register.



Addressing Modes

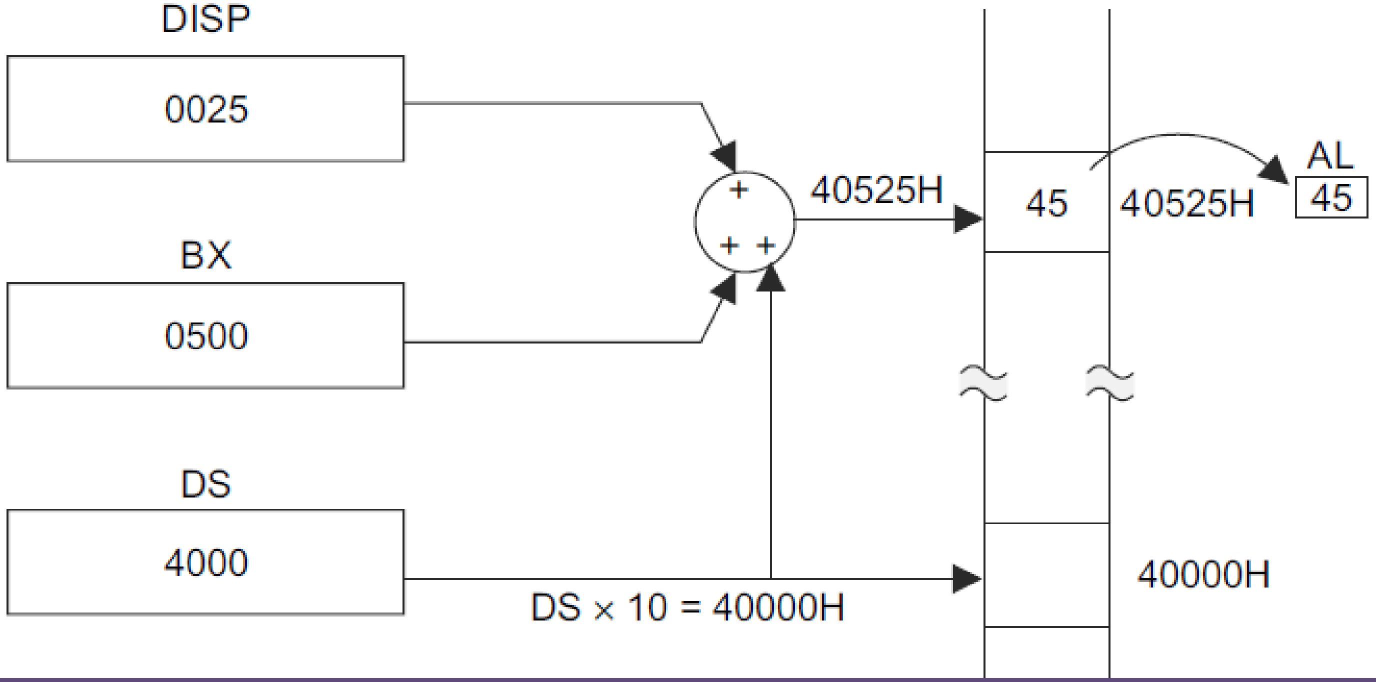
3c. Based Addressing

- The 8-bit or 16-bit instruction operand is added to the contents of a base register (BX or BP), the resulting value is a pointer to the location where the data resides.
- The effective address in based addressing mode is obtained by adding the direct or indirect displacement to the contents of either the base register BX or the base pointer BP.
- EA = BX + 8-bit displacement, EA = BP + 8-bit displacement
- EA = BX + 16-bit displacement, EA = BP + 16-bit displacement
- PA = Segment × 10 + BX + 8-bit displacement
- PA = Segment × 10 + BP + 8-bit displacement
- PA = Segment × 10 + BX + 16-bit displacement
- PA = Segment × 10 + BP + 16-bit displacement

Addressing Modes

3c. Based Addressing

- When 16-bit DISP = 0025H, BX = 0500H and DS = 4000H, PA = DS × 10 + BX + DISP = 4000H × 10 + 0500 + 0025 = 40525H.
- MOV AL, DS: [BX+DISP] – contents of memory location 40525H will be copied into the AL register.



Addressing Modes

3c. Based Addressing

- Examples:

```
MOV AL, [BX+8-bit DISP]
MOV AH, [BX+8-bit DISP]
MOV AL, [BP+8-bit DISP]
MOV AH, [BP+8-bit DISP]
```

Addressing Modes

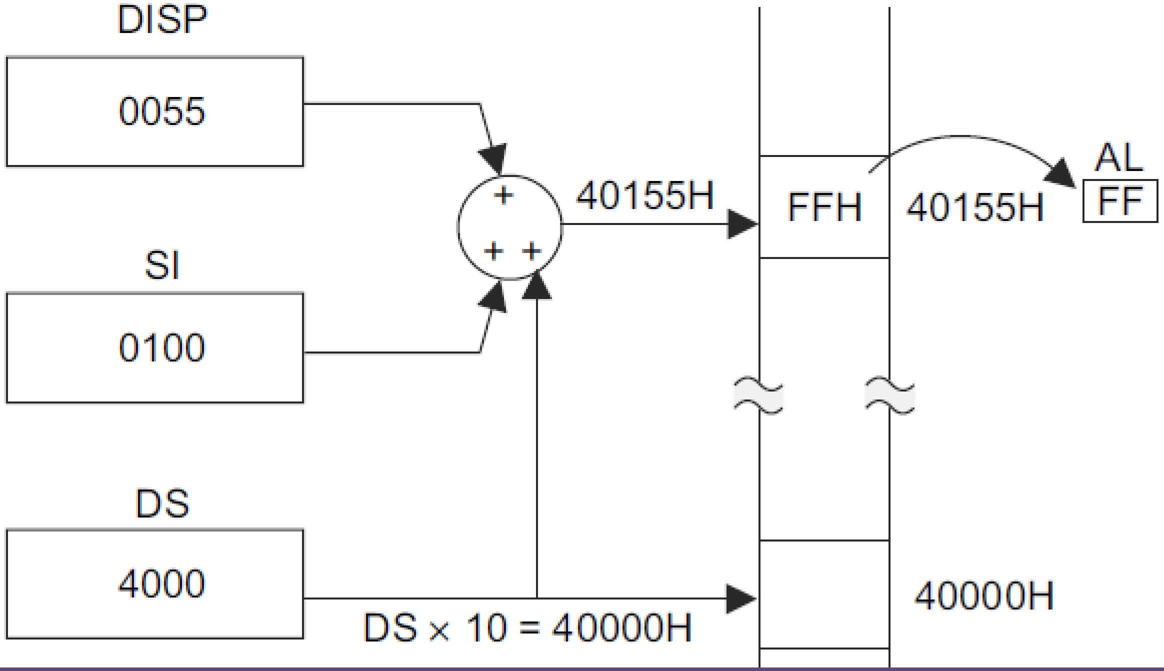
3d. Indexed Addressing

- The 8-bit or 16-bit instruction operand is added to the contents of an index register (SI or DI), and the resulting value is a pointer to the location where the data resides.
- The displacement value is used as a pointer to the starting point of an array of data in memory and the contents of the specified register is used as an index.
- EA = SI + 8-bit displacement, EA = DI + 8-bit displacement
- EA = SI + 16-bit displacement, EA = DI + 16-bit displacement
- PA = Segment × 10 + SI + 8 bit displacement
- PA = Segment × 10 + DI + 8 bit displacement
- PA = Segment × 10 + SI + 16 bit displacement
- PA = Segment × 10 + SI + 16 bit displacement

Addressing Modes

3d. Indexed Addressing

- 16-bit DISP = 0055H, SI = 0100H and DS = 4000H, PA = DS × 10 + SI + DISP = 4000H × 10 + 0100 + 0055 = 40155H.
- MOV AL, DS: [SI + DISP] - contents of the memory location 40155H, FF will be loaded into the AL register.



Addressing Modes

3d. Indexed Addressing

- The index addressing modes generally involve BX, SI, and DI registers with the data segment and [BP+DISP] uses the stack segment by default.
- In the register indirect addressing modes, the segment override prefixes can be used to specify different segments.
- Ex.:

```
MOV AL, SS: [BX + DISP]
MOV AL, ES: [BP + DISP]
MOV AL, CS: [SI + DISP]
MOV AL, SS: [DI + DISP]
```

Addressing Modes

3e. Based Indexed Addressing

- The contents of a base register (BX or BP) is added to the contents of an index register (SI or DI), the resulting value is a pointer to the location where the data resides.
- The effective address is the sum of a base register and an index register which are specified in the instruction.
- The based indexed addressing modes are simply combinations of the register indirect addressing modes.
- These addressing modes form the offset by adding together a base register (BX or BP) and an index register (SI or DI).

Addressing Modes

3e. Based Indexed Addressing

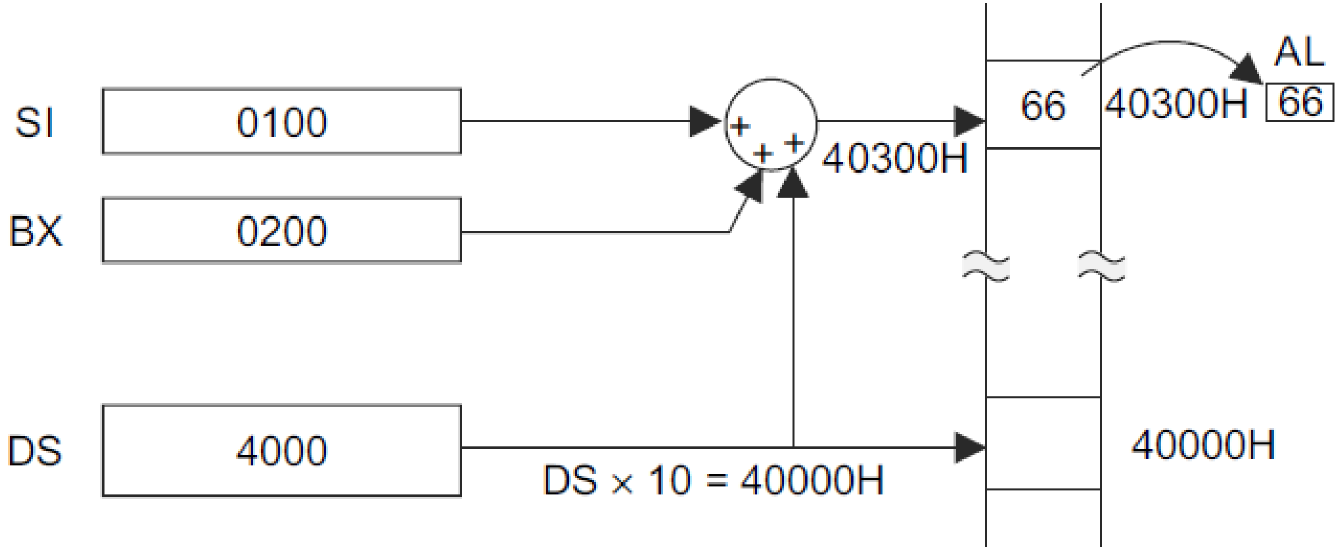
The EA and the PA computation are given below:

- EA = BX + SI,
- EA = BX + DI
- EA = BP + SI,
- EA = BP + DI
- PA = Segment × 10 + BX + SI
- PA = Segment × 10 + BX + DI
- PA = Segment × 10 + BP + SI
- PA = Segment × 10 + BP + DI

Addressing Modes

3e. Based Indexed Addressing

- If BX = 0200H and SI = 0100H, MOV AL, [BX + SI] loads the content of the memory location DS × 10 + BX + SI into the AH register.
- If DS = 4000H, the memory location address is 4000 × 10 + 0200 + 0100 = 40300H whose content 66H will be loaded into the AH register.



Addressing Modes

3e. Based Indexed Addressing

- Examples:

```
MOV AL, [BX + DI]
MOV AL, [BX + SI]
MOV AL, [BP + SI]
MOV AL, [BP + DI]
```

Addressing Modes

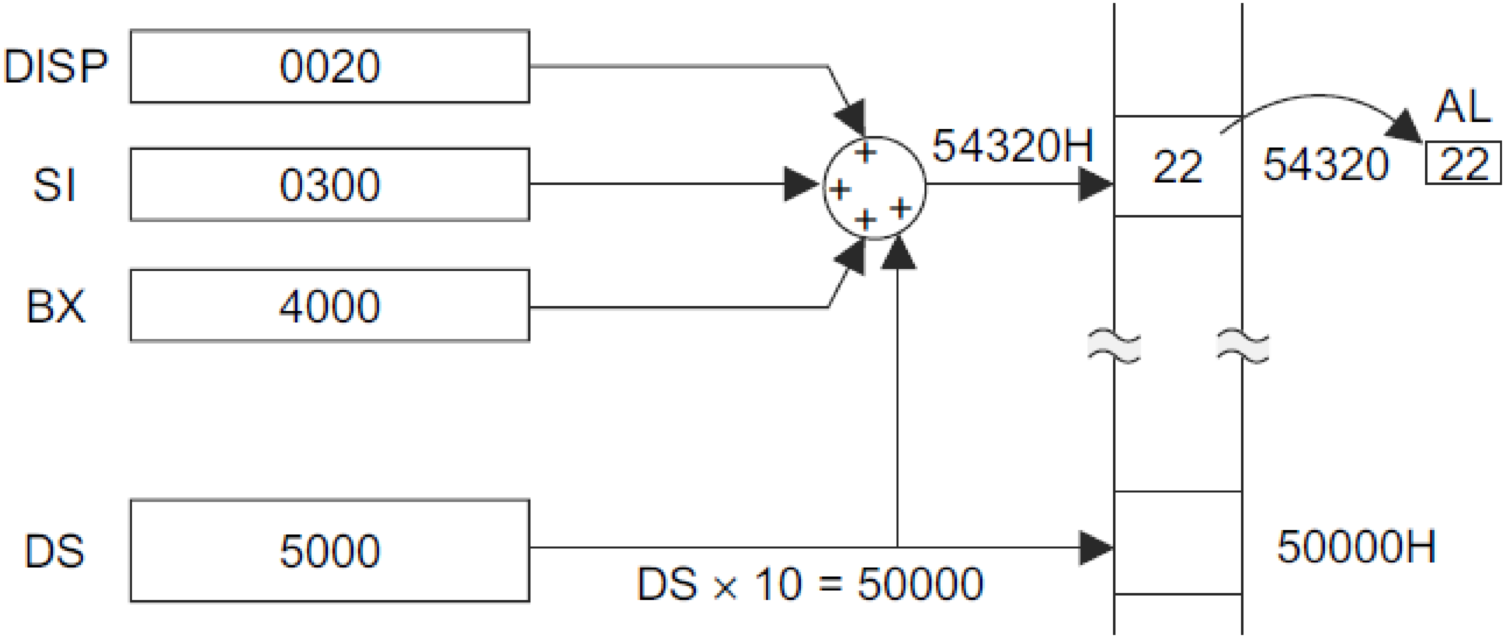
3f. Based Indexed with Displacement Addressing

- The 8-bit or 16-bit instruction operand is added to the contents of a base register (BX or BP) and index register (SI or DI), the resulting value is a pointer to the location where the data resides.
- The effective address is the sum of an 8-bit or 16-bit displacement and based index address.
- EA = BX + SI + 8-bit or 16-bit instruction
- EA = BX + DI 8-bit or 16-bit instruction
- EA = BP + SI + 8-bit or 16-bit instruction
- EA = BP + DI 8-bit or 16-bit instruction
- PA = Segment × 10 + BX + SI + 8-bit or 16-bit instruction
- PA = Segment × 10 + BX + DI + 8-bit or 16-bit instruction
- PA = Segment × 10 + BP + SI + 8-bit or 16-bit instruction
- PA = Segment × 10 + BP + DI+ 8-bit or 16-bit instruction

Addressing Modes

3f. Based Indexed with Displacement Addressing

- If DISP = 0020H, BX = 4000H, SI = 0300 and DS = 5000H, PA = DS × 10 + BX + SI + DISP = 5000H × 10 + 4000 + 0300 + 0020 = 54320H.
- MOV AL, DS:[BX + SI + DISP] - content of the memory location 54320H will be copied into the AL register.



Addressing Modes

3f. Based Indexed with Displacement Addressing

- Examples:

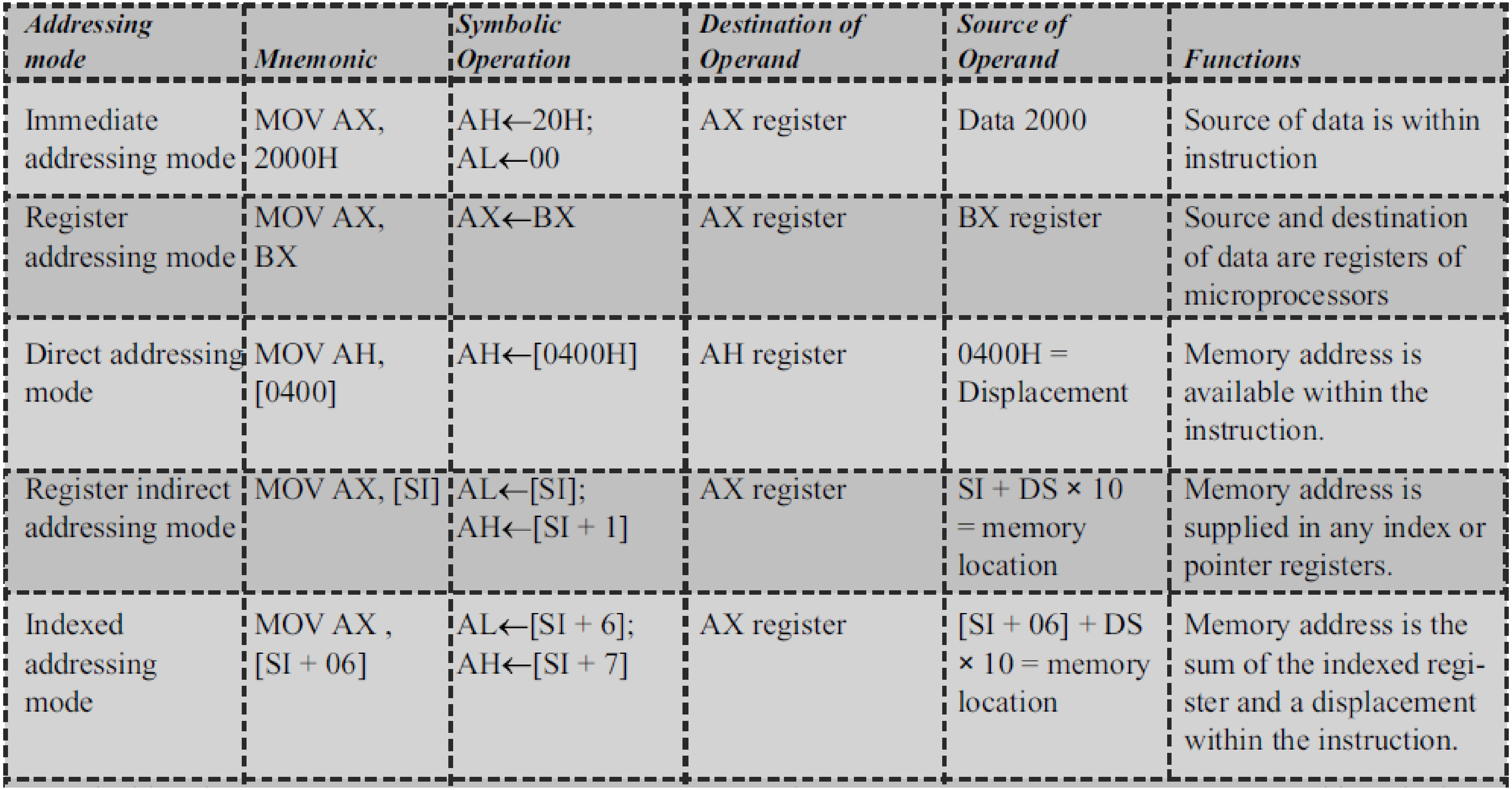
```
MOV AL, [BX + DI + DISP]
MOV AL, [BX + SI + DISP]
MOV AL, [BP + SI + DISP]
MOV AL, [BP + DI + DISP]
```

Addressing Modes

String Addressing Mode

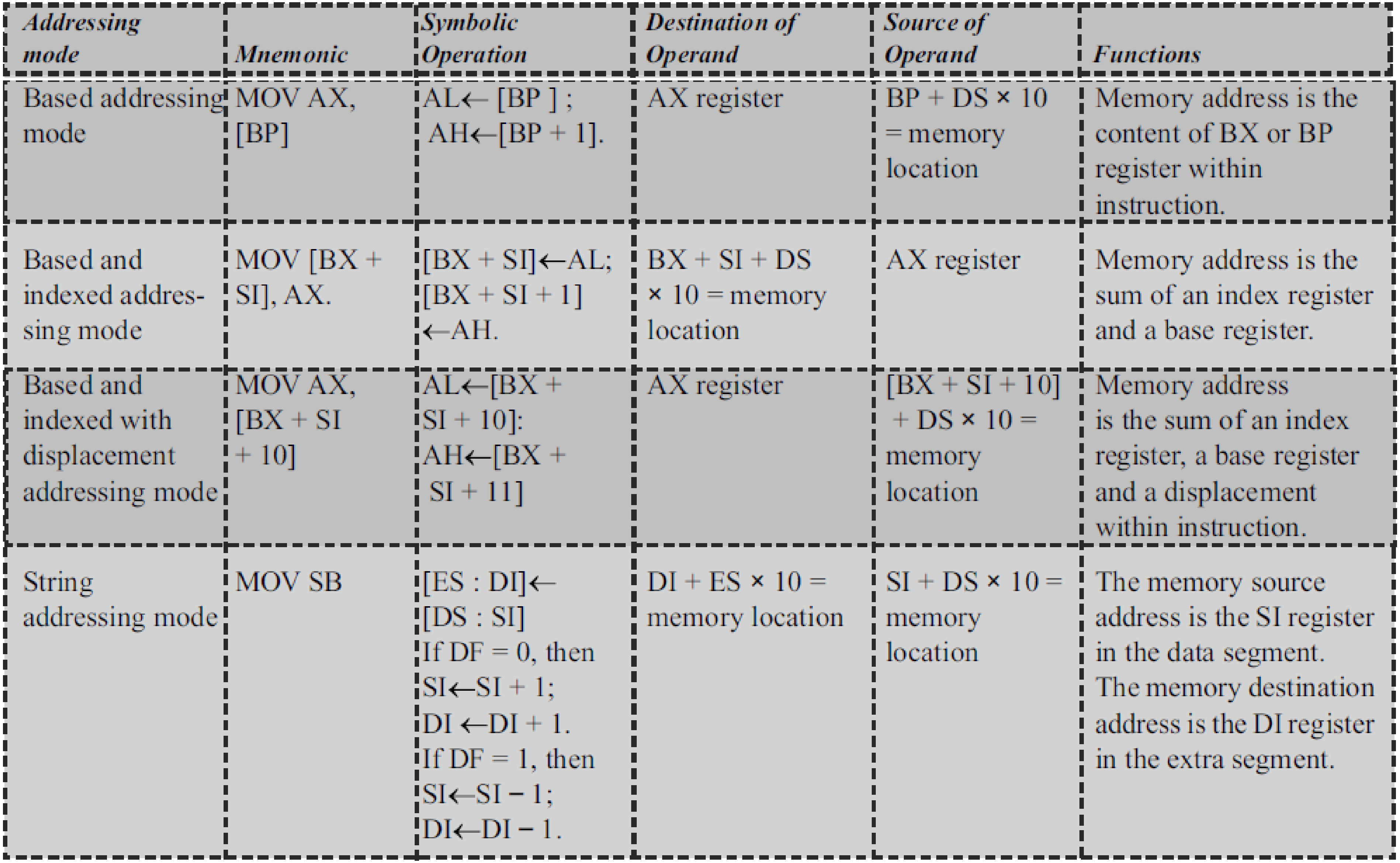
- String is a sequence of bytes or words which are stored in memory.
- Stored characters in word processors and data table are examples of string.
- Some instructions are designed to handle a string of characters or data.
- These instructions have a special addressing mode where DS : SI is used as a source of string and ES:DI is used to locate the destination address of the string.
- For example, MOV SB instruction is used to move a string of source data to the destination location.

Addressing Modes



Addressing mode	*Mnemonic*	*Symbolic Operation*	*Destination of Operand*	*Source of Operand*	*Functions*
Immediate addressing mode	MOV AX, 2000H	AH←20H; AL←00	AX register	Data 2000	Source of data is within instruction
Register addressing mode	MOV AX, BX	AX←BX	AX register	BX register	Source and destination of data are registers of microprocessors
Direct addressing mode	MOV AH, [0400]	AH←[0400H]	AH register	0400H = Displacement	Memory address is available within the instruction.
Register indirect addressing mode	MOV AX, [SI]	AL←[SI]; AH←[SI + 1]	AX register	SI + DS × 10 = memory location	Memory address is supplied in any index or pointer registers.
Indexed addressing mode	MOV AX , [SI + 06]	AL←[SI + 6]; AH←[SI + 7]	AX register	[SI + 06] + DS × 10 = memory location	Memory address is the sum of the indexed register and a displacement within the instruction.

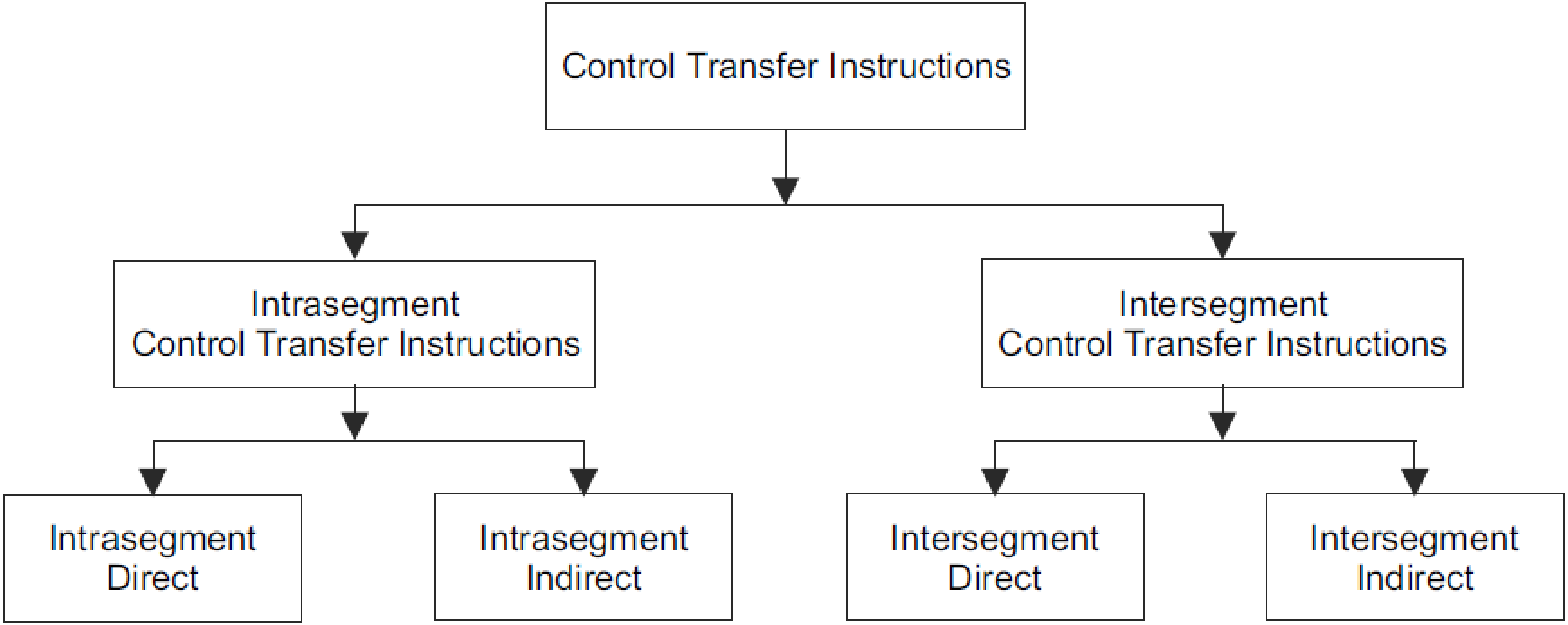
Addressing Modes



Addressing mode	*Mnemonic*	*Symbolic Operation*	*Destination of Operand*	*Source of Operand*	*Functions*
Based addressing mode	MOV AX, [BP]	AL← [BP] ; AH←[BP + 1].	AX register	BP + DS × 10 = memory location	Memory address is the content of BX or BP register within instruction.
Based and indexed addressing mode	MOV [BX + SI], AX.	[BX + SI]←AL; [BX + SI + 1] ←AH.	BX + SI + DS × 10 = memory location	AX register	Memory address is the sum of an index register and a base register.
Based and indexed with displacement addressing mode	MOV AX, [BX + SI + 10]	AL←[BX + SI + 10]: AH←[BX + SI + 11]	AX register	[BX + SI + 10] + DS × 10 = memory location	Memory address is the sum of an index register, a base register and a displacement within instruction.
String addressing mode	MOV SB	[ES : DI]← [DS : SI] If DF = 0, then SI←SI + 1; DI ←DI + 1. If DF = 1, then SI←SI − 1; DI←DI − 1.	DI + ES × 10 = memory location	SI + DS × 10 = memory location	The memory source address is the SI register in the data segment. The memory destination address is the DI register in the extra segment.

Addressing Modes

4. Branch Addressing


Control Transfer Instructions
Intrasegment Control Transfer Instructions
Intersegment Control Transfer Instructions
Intrasegment Direct
Intrasegment Indirect
Intersegment Direct
Intersegment Indirect

Addressing Modes

4. Branch Addressing

- **Intrasegment mode:** To transfer the control to a destination that lies in the same segment where the control transfer instruction itself resides.

- **Intersegment mode:** To transfer the control to a destination that lies in a different segment.

- Addressing modes depends on:
  - Destination location is within the same segment or in a different one.
  - Method of passing the destination address to the processor.

Addressing Modes

4a. Intrasegment Direct

- The effective branch address is sum of an 8-bit or 16-bit displacement and the current contents of IP.

- When the displacement is 8-bit long, it is referred to as a short jump.

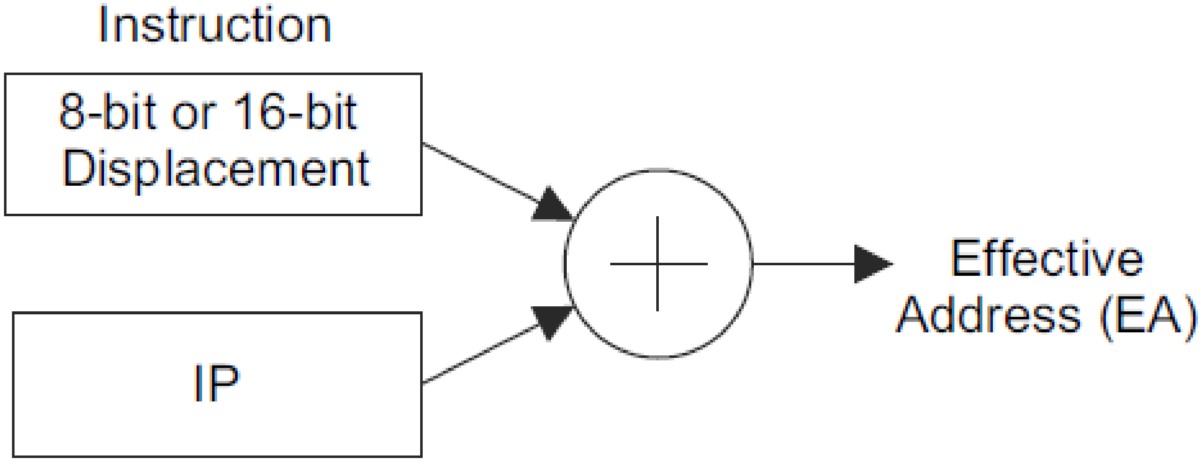
- Also referred as relative addressing because the displacement is computed ‘relative’ to the IP.

- Used with either conditional or unconditional branching, but a conditional branch instruction can have only 8-bit displacement.

Addressing Modes

4a. Intrasegment Direct

- Destination location to which the control is transferred lies in the same segment where the control-transfer instruction lies and appears directly in the instruction as an immediate displacement value.
- EA = Contents of IP + 8- or 16-bit displacement.



Addressing Modes

4b. Intrasegment Indirect

- The effective branch address is the contents of a register or memory location that is accessed using any of the above data-related addressing modes except the immediate mode.

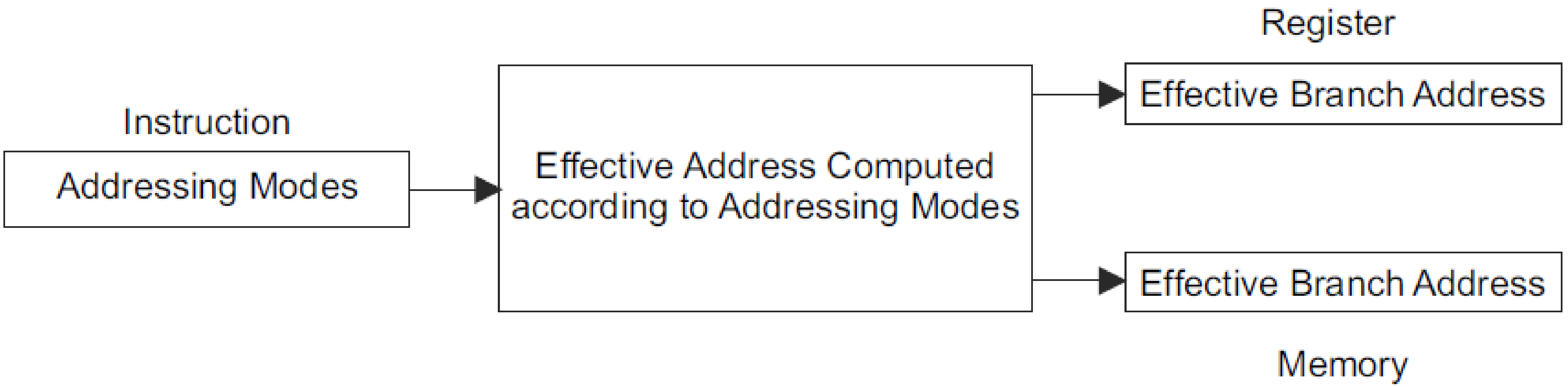
- The contents of IP are replaced by the effective branch address.

- Used only in unconditional branch instructions.

- Control to be transferred lies in the same segment where the control instruction lies and is passed indirectly to the instruction.

Addressing Modes

4b. Intrasegment Indirect



Register

Instruction

Addressing Modes

Effective Address Computed according to Addressing Modes

Effective Branch Address

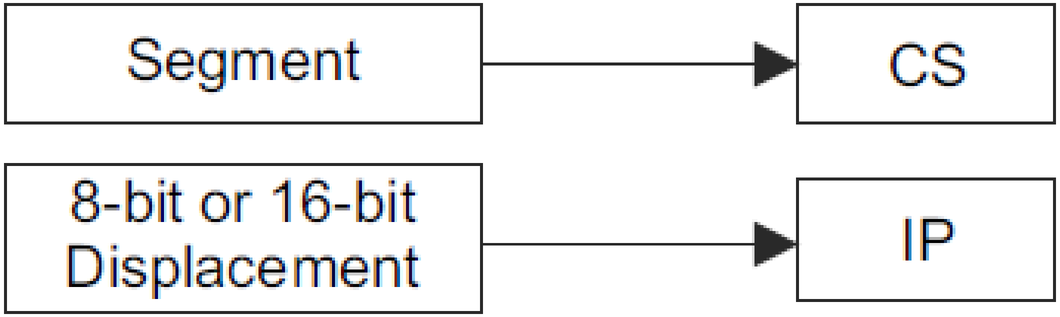
Effective Branch Address

Memory

Addressing Modes

4c. Intersegment Direct

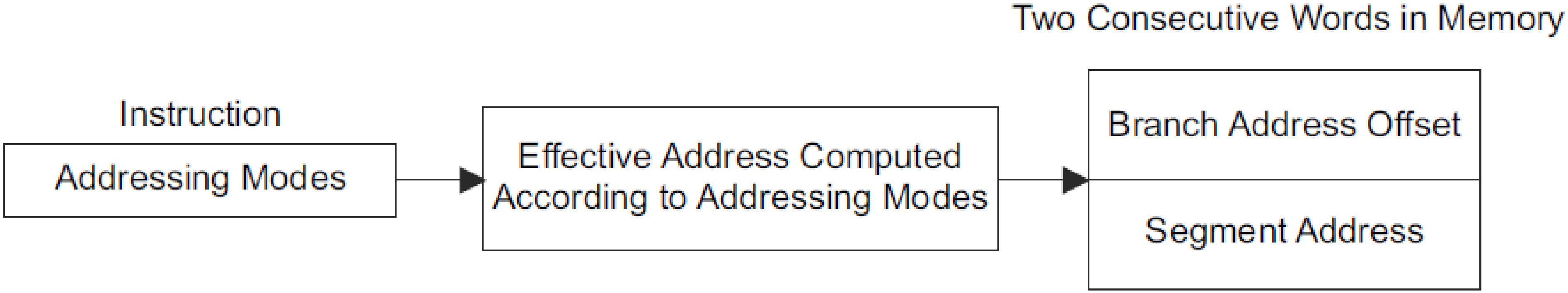
- This replaces the contents of IP with a part of the instruction and the contents of CS with another part of the instruction.
- The purpose is to provide a means of branching from one code segment to another.



Addressing Modes

4d. Intersegment Indirect

- This mode replaces the contents of IP and CS with the contents of two consecutive words in memory that are referenced using any of the above data-related addressing modes except the immediate and register modes.
- The starting address of the memory block may be referred using any of the addressing modes except immediate.


Two Consecutive Words in Memory
Instruction
Addressing Modes
Effective Address Computed According to Addressing Modes
Branch Address Offset
Segment Address

Addressing Modes

Problems

1. Find the addressing modes:

```
MOV CX, BX
MOV BX, 1234
MOV AX, [SI]
MOV [Offset Address], 2345
MOV CX, [BX+SI]
MOV AX, [BX+SI+1234]
```

Addressing Modes

Problems

1. Find the addressing modes:

MOV CX, BX	Register addressing
MOV BX, 1234	Immediate addressing
MOV AX, [SI]	Indexed addressing
MOV [Offset Address], 2345	Memory addressing
MOV CX, [BX+SI]	Based indexed addressing
MOV AX, [BX+SI+1234]	Based indexed with displacement addressing

Addressing Modes

Problems

2. Assume AX = 1000H, BX = 2000H, SI = 3000H, DI = 4000H, BP = 5000H, SP = 6000H, CS = 8000H, DS = 1000H, SS = 2000H, IP = 7000H, Offset (displacement) = 0500H. Determine the 16-bit effective addresses and 20-bit physical address for the following addressing modes:

a. Direct addressing mode - MOV AX, [0500H]

16-bit effective addresses = 0500

20-bit physical address = CS × 10 + 0500

= 8000 × 10 + 0500 = 80500H

Addressing Modes

Problems (2 contd.)

b. Register indirect addressing - MOV AX, [BX]

16-bit effective addresses = 2000

20-bit physical address = CS × 10 + BX

= 8000 × 10 + 2000 = 82000H

c. Based indexed addressing - MOV AX, DS:[BX + SI]

16-bit effective addresses = BX + SI = 2000 + 3000 = 5000H

20-bit physical address = DS × 10 + BX + SI

= 1000 × 10 + 2000 + 3000 = 15000H

Addressing Modes

Problems (2 contd.)

d. Based indexed with displacement addressing

MOV AX, DS:[BX + SI + DISP]

20-bit physical = DS × 10 + BX + SI + DISP

= 1000 × 10H + 2000H + 3000H + 0500H

= 15500H

Addressing Modes

Problems

3. Determine the starting and ending address for data segment and code segment. Assume DS = 5000H and CS = 7000H.

Solution:

Starting = DS x 10 + 0000H = 50000H

Ending = DS x 10 + FFFFH = 5FFFFH

Starting = CS x 10 + 0000H = 70000H

Ending = CS x 10 + FFFFH = 7FFFFH

Addressing Modes

Problems

4. What physical address can be accessed by the instruction MOV [BP], AL if BP = 2500H. Assume the content of stack segment register is 4578H.

Solution:

MOV [BP], AL

SS × 10 + BP = 4578 × 10 + 2500H = 47C80H.