Chapter 5

8086 Microprocessor

Addressing Modes

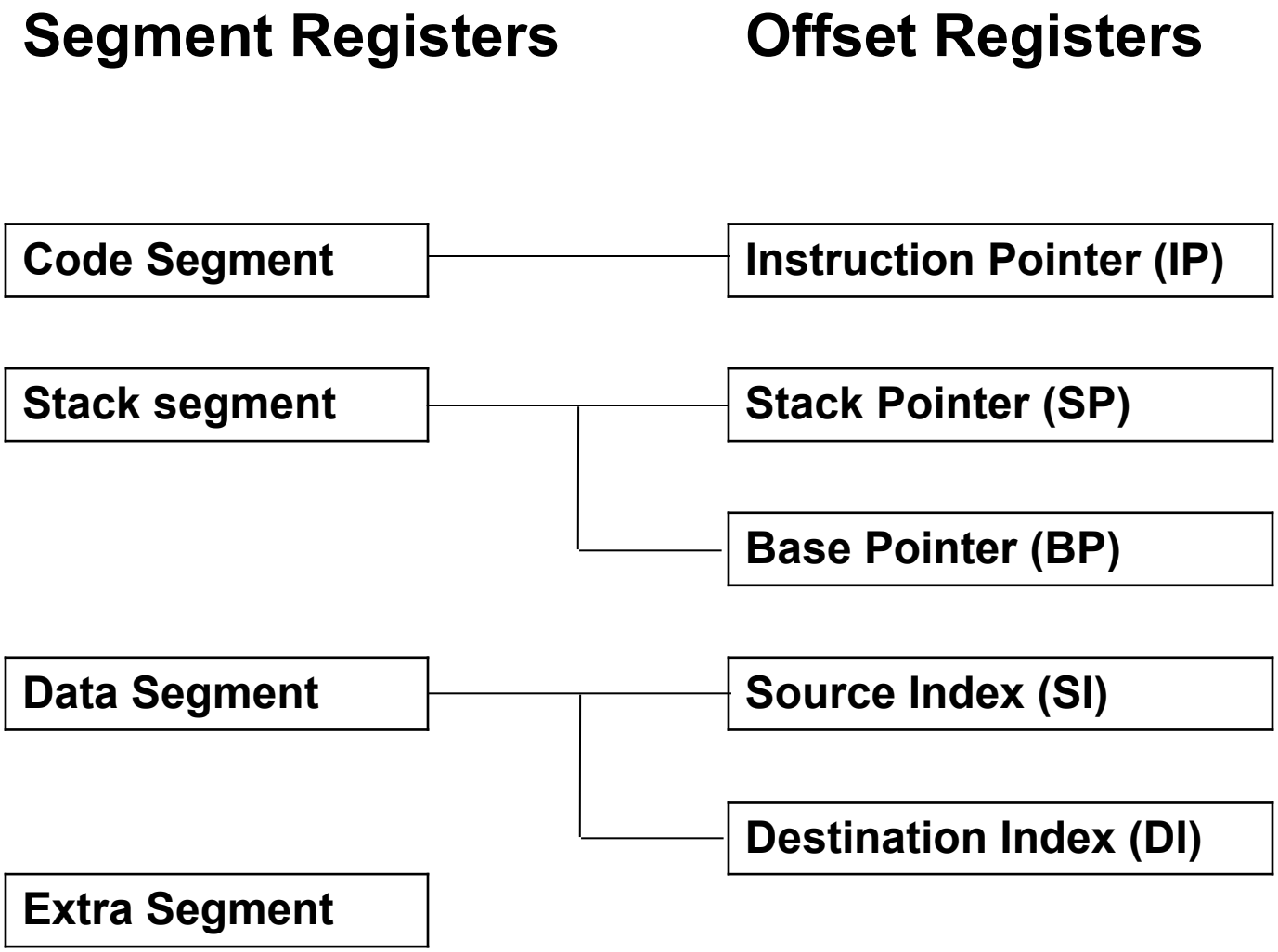
Bharati Ingale (bharati_ingale@yahoo.com)

8086 – Addressing Modes

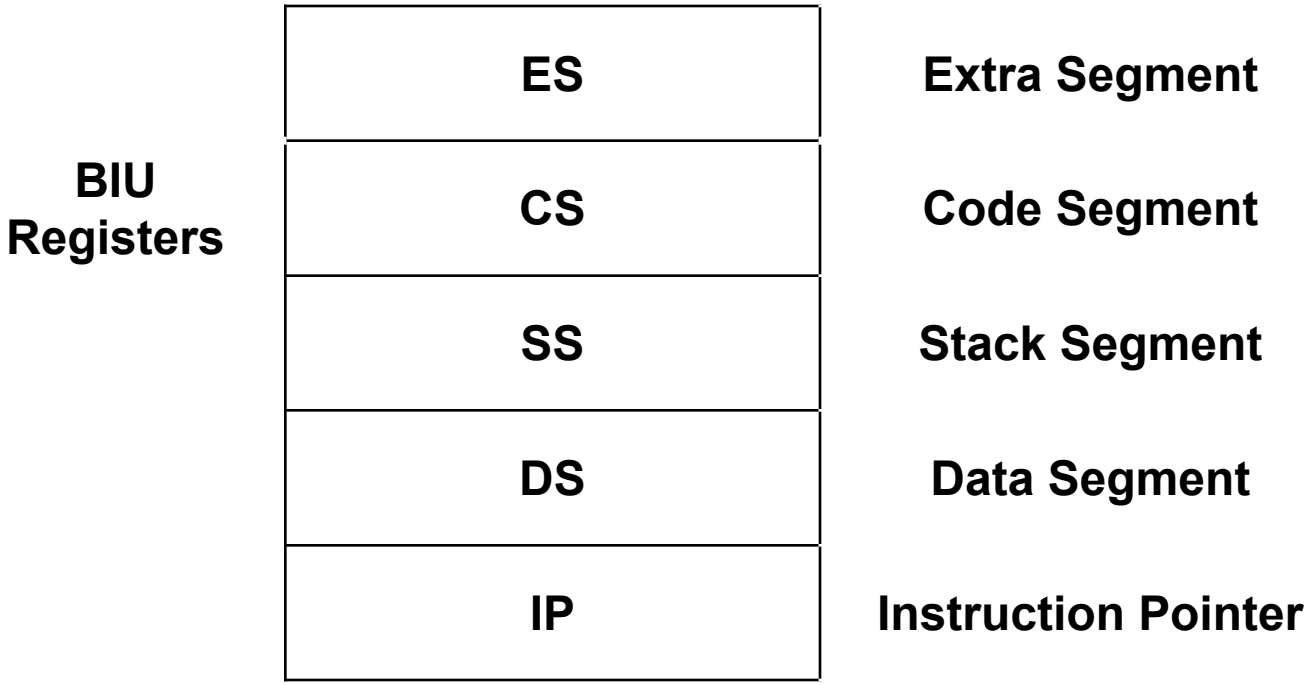
- Addressing mode indicates a way of locating data or operands.

- The different ways that a processor accesses the data is referred to as addressing mode.

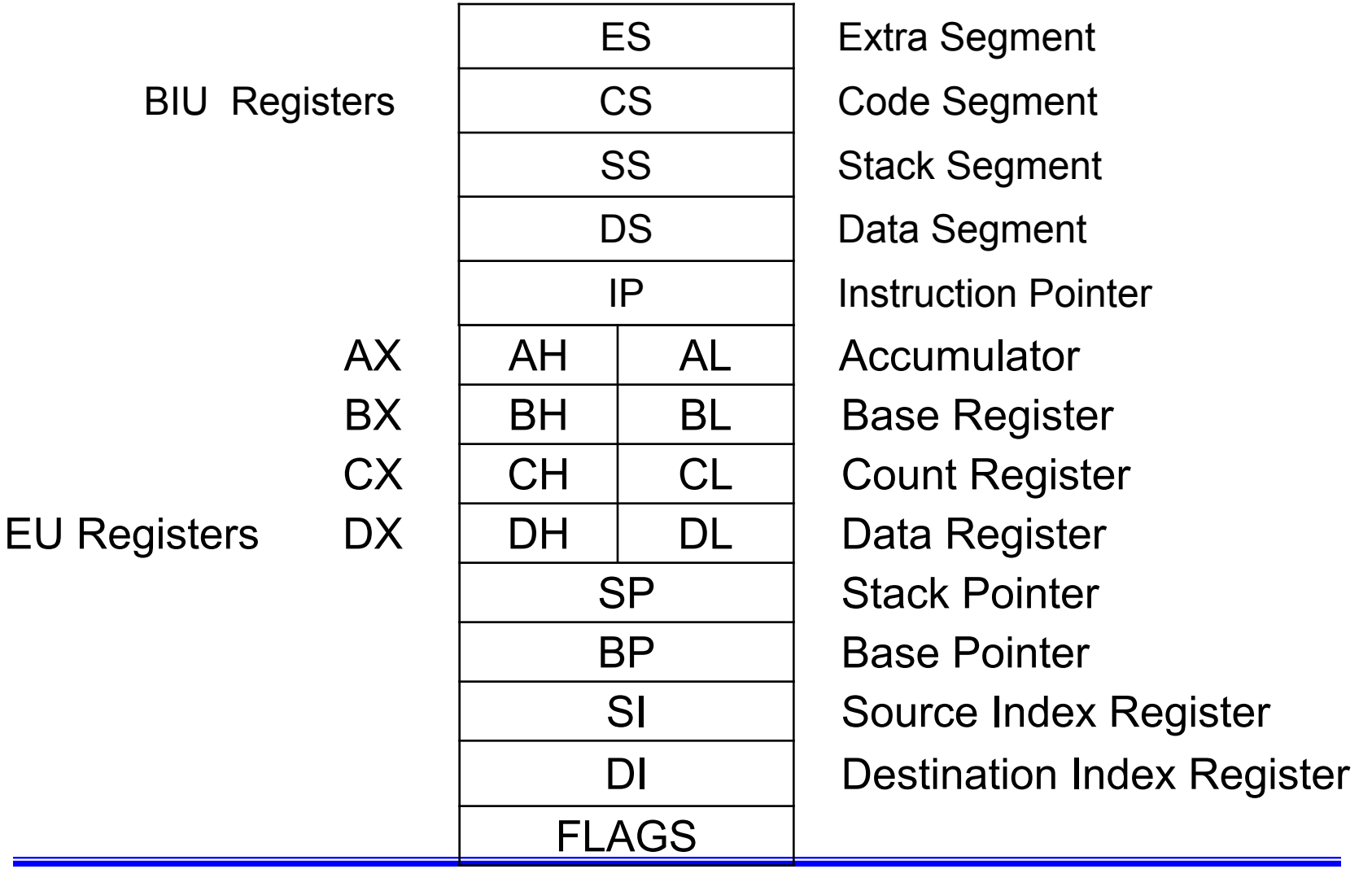
8086 – Addressing Modes



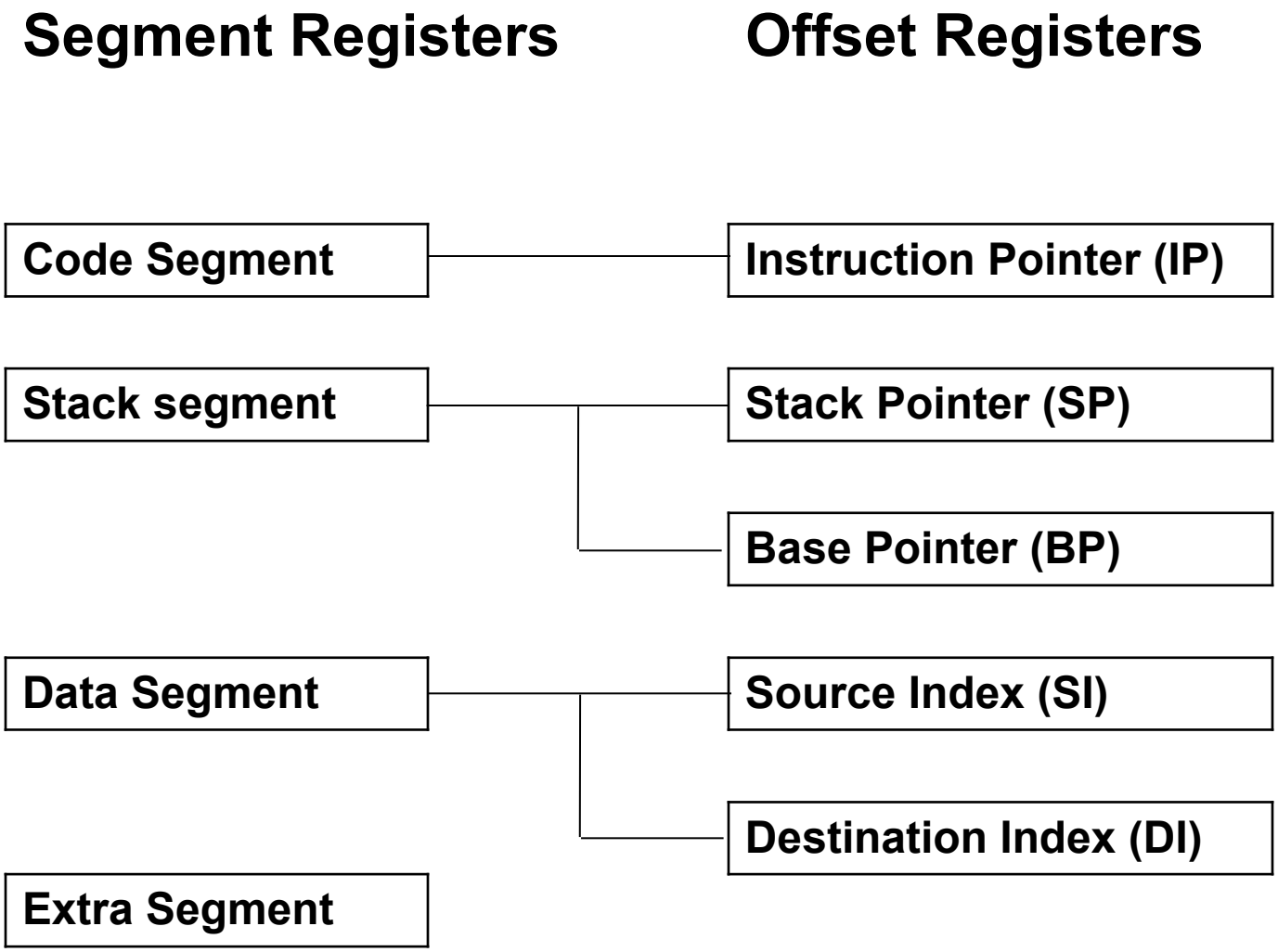
8086 – Programmer's Model of 8086



8086 – Programmer's Model of 8086



8086 – Addressing Modes



8086- Architecture

Flag Register

- Flag is a flipflop
- 9 flags
- 2 types: 6-Status & 3-Control

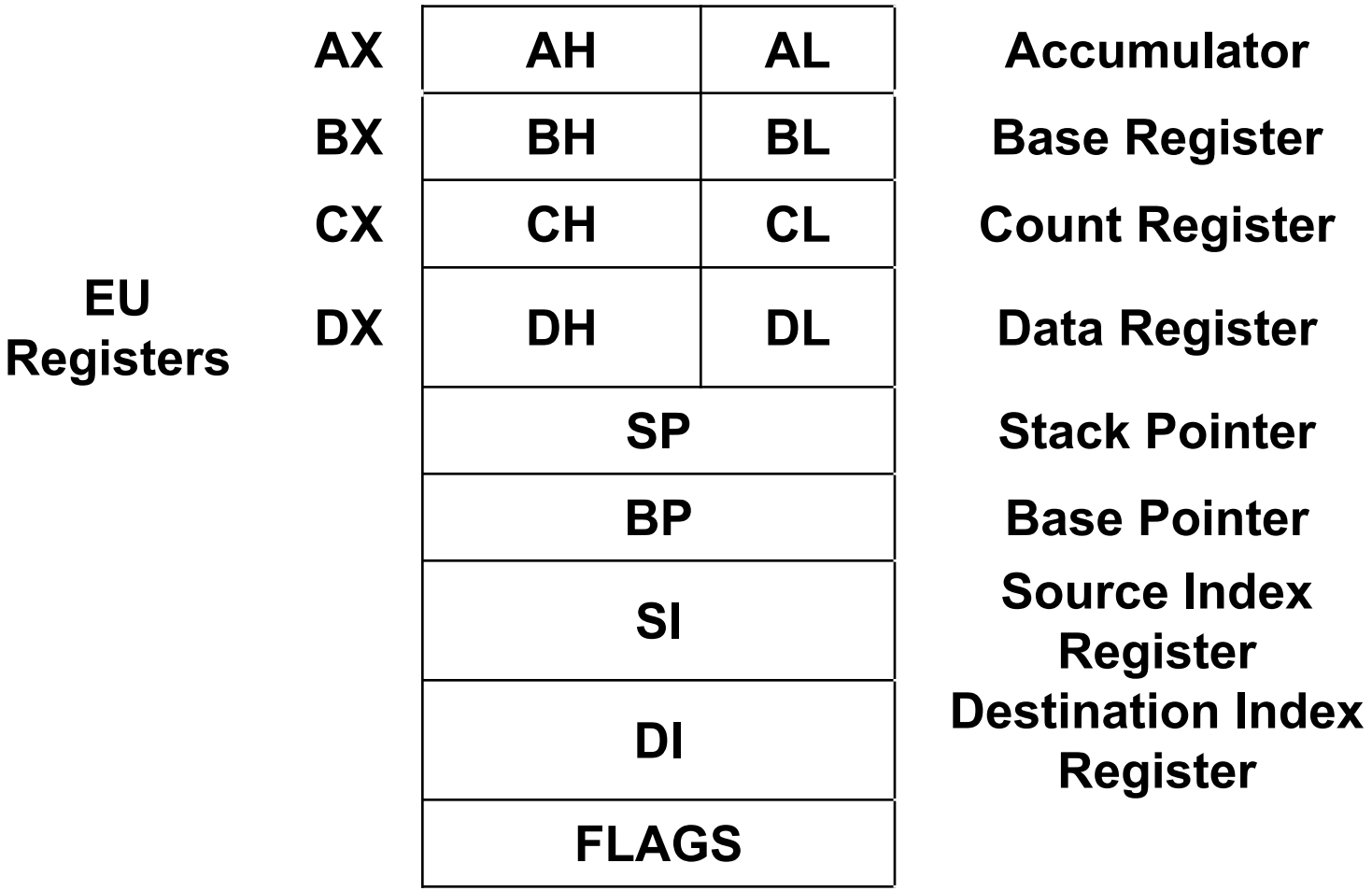
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X	X	X	X	OF	DF	IF	TF	SF	ZF	X	AF	X	PF	X	CF

Control Flag		Status Flag	
1	DF – Direction	1	SF - Sign
2	IF – interrupt Enable	2	ZF – Zero
3	TF – Trap	3	AF – Auxillary Carry
		4	PF – Parity
		5	CF – Carry
		6	OF - Overflow

- Stat
- Con

D15 D14 D13 D12 D11D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0

8086 – Programmer's Model of 8086



8086 – Addressing Modes

- An instruction may belong to one or more addressing modes, **depending upon the data types used in the instruction and the memory addressing modes,**

8086 – Addressing Modes

- Thus the addressing modes

  i. describe the types of operands and

  ii. the way they are accessed for executing an instruction

8086 – Addressing Modes

- Addressing modes depends on the **types of the instructions.**

- According to the flow of instruction execution, the instructions may be categorized as

i. Sequential control flow instructions and

ii. Control transfer instructions.

8086 – Addressing Modes

- **Sequential control flow instructions**

  When the Instructions,

i. after execution,

ii. transfers the control to the next instruction

  appearing immediately after it (in the sequence)

  in the program.

Eg.: the arithmetic, logical, data transfer and processor control instructions.

8086 – Addressing Modes

- **Control Transfer Instructions**,

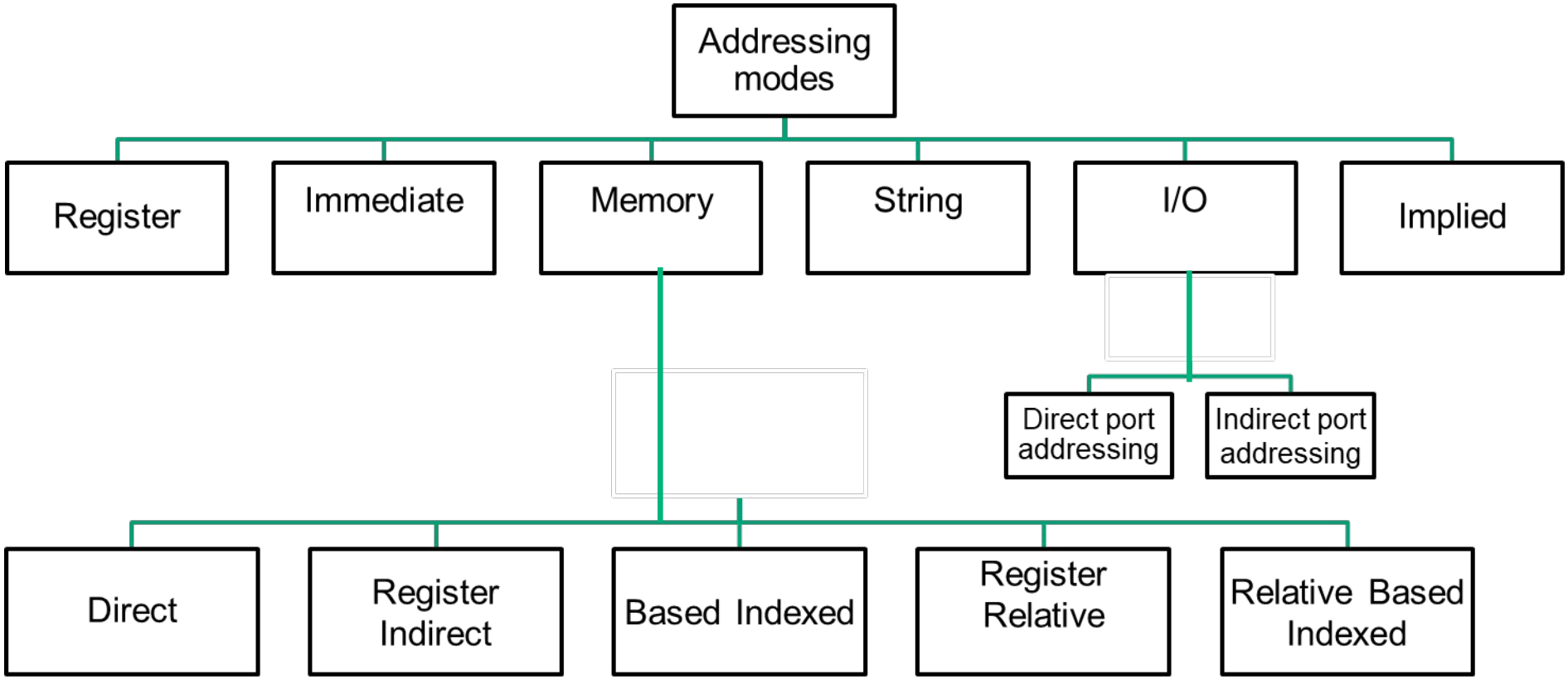
  Instructions transfers the control to some predefined address specified in the instruction after its execution.

Eg.: INT, CALL, RET and JUMP instructions .

Addressing modes

1. Immediate Addressing mode
2. Register Addressing mode
3. Memory Addressing mode
4. String Addressing mode
5. I/O Addressing mode
6. Implied Addressing mode

8086


Addressing modes
Register
Immediate
Memory
String
I/O
Implied
Direct port addressing
Indirect port addressing
Direct
Register Indirect
Based Indexed
Register Relative
Relative Based Indexed

8086 - Addressing modes for Data Memory

1. Immediate Addressing Mode:

- In this type of addressing, immediate data is a part of instruction, and appears in the form of successive byte or bytes.
- Example: MOV AX, 0005H
- The immediate data 0005H is moved into reg AX.
- The immediate data may be 8-bit or 16-bit in size.

OPCODE	Immediate Operand

.

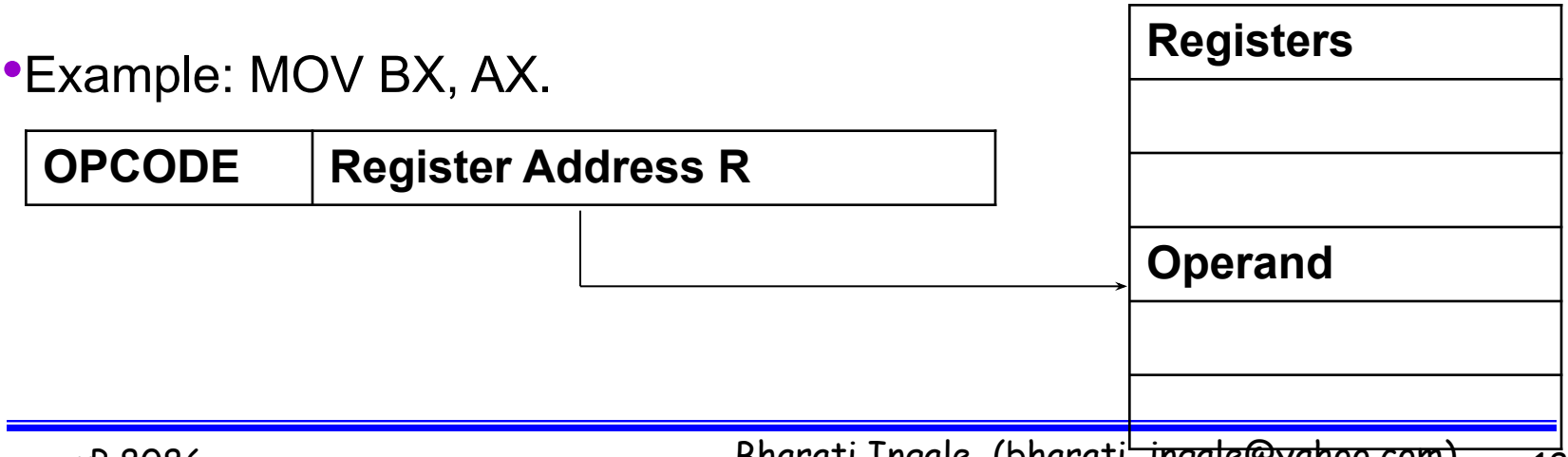
8086 - Addressing modes for Data Memory

2. Register Addressing Mode:

- In register addressing mode, the data is stored in a register and it is referred using the particular register.

- All the registers, except IP, may be used in this mode.

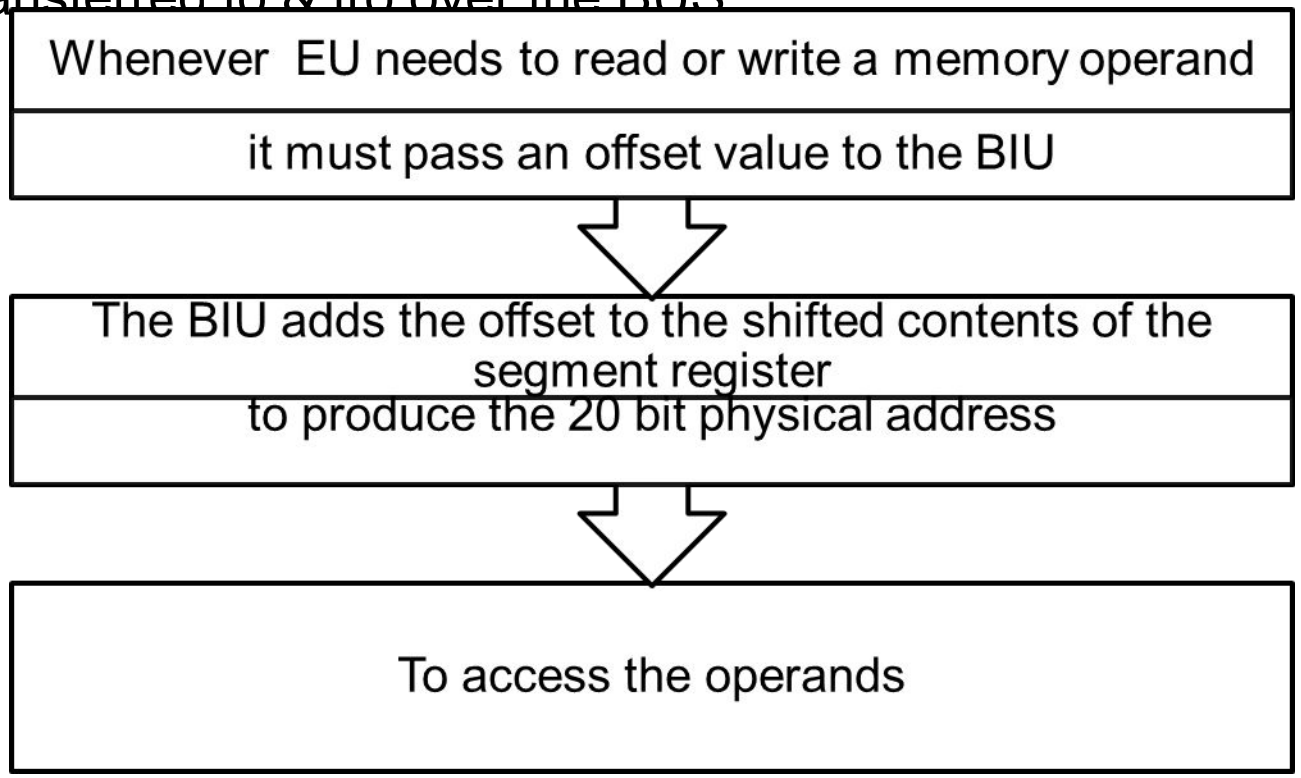
- Example: MOV BX, AX.



8086 - Addressing modes for Data Memory

3. Memory Addressing Mode:

- In this addressing mode the operands in the memory must be transferred to & fro over the BUS

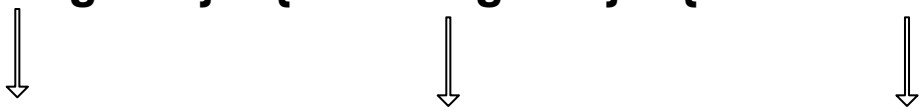


8086 - Addressing modes for Data Memory

3. Memory Addressing Mode:

- Offset for a memory operand is called the operand's effective address EA

EA = {Base register} + {Index register} + {8 or 16 bit displacement}



{BX,BP} **{SI,DI}** **{8 or 16 bit displacement}**

Any combination of these three components gives rise to variety of 8086 memory addressing mode.

8086 - Addressing modes for Data Memory

3. Memory Addressing Mode:

The physical address is then calculated once we get the Effective Address

PA = Segment : Offset

= Segment register : EA

PA = Segment register :Base + Index + displacement

{CS,SS,DS,ES} {BX,BP} + {SI,DI} + {8 or 16 bit displacement}

8086 - Addressing modes for Data Memory

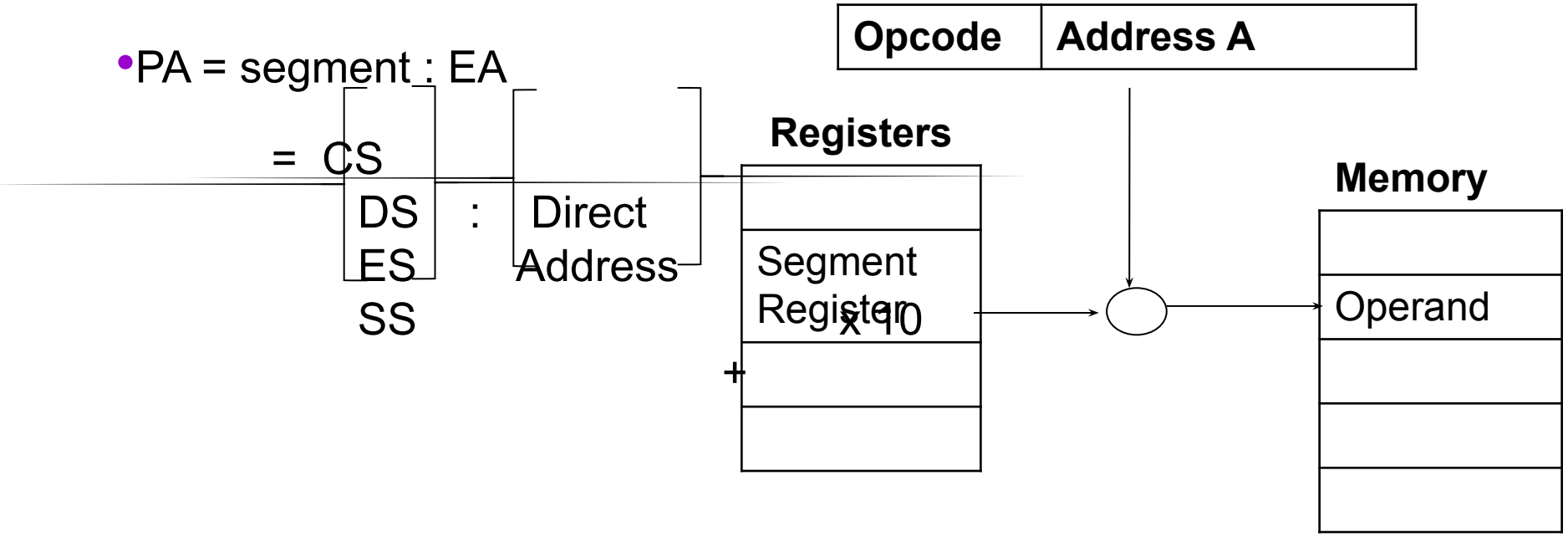
The different Memory addressing modes are:

1. Direct memory addressing mode
2. Register indirect addressing mode
3. Based indexed addressing mode
4. Register relative addressing mode
5. Relative based indexed addressing mode

8086 - Addressing modes for Data Memory

3.1. Direct Memory Addressing Mode:

- In this mode, a 16-bit memory address (offset) is directly specified in the instruction as a part of it.
- EA = 8/16 bit displacemnt
- PA = segment : EA



8086 - Addressing modes for Data Memory

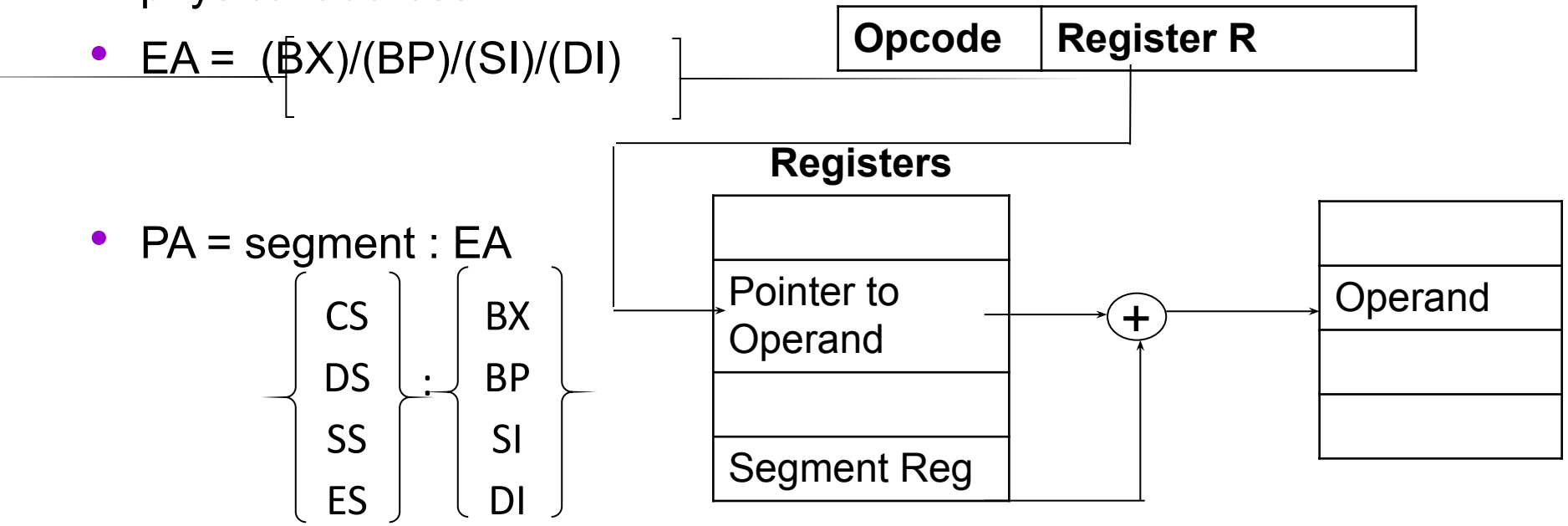
3.1. Direct Memory Addressing Mode:

- Example: MOV [5000H], AL
- The contents of AL are copied to memory location whose offset is [5000]
- The effective address, here, is **10H*DS+5000H.**
- **By default, DS segment register is used for direct addressing mode**

8086 - Addressing modes for Data Memory

3.2. Register Indirect Addressing Mode:

- The EA of the memory is taken directly from one of the base register or index register specified by the instruction.
- This address is added with the segment reg*10 to generate the 20 bit physical address.
- EA = (BX)/(BP)/(SI)/(DI)



- PA = segment : EA

CS, DS, SS, ES : BX, BP, SI, DI

- If reg is SI,DI,BX ⇒ DS default segment

8086 - Addressing modes for Data Memory

3.2. Register Indirect Addressing Mode:

- If reg is SI,DI,BX ⟹ DS default segment register
- If BP ⟹ SS default segment register

- Example: MOV AX, [BX]
- Here, data is present in a memory location in DS whose offset address is in BX.

The effective address of the data is given as 10H*DS+ [BX].

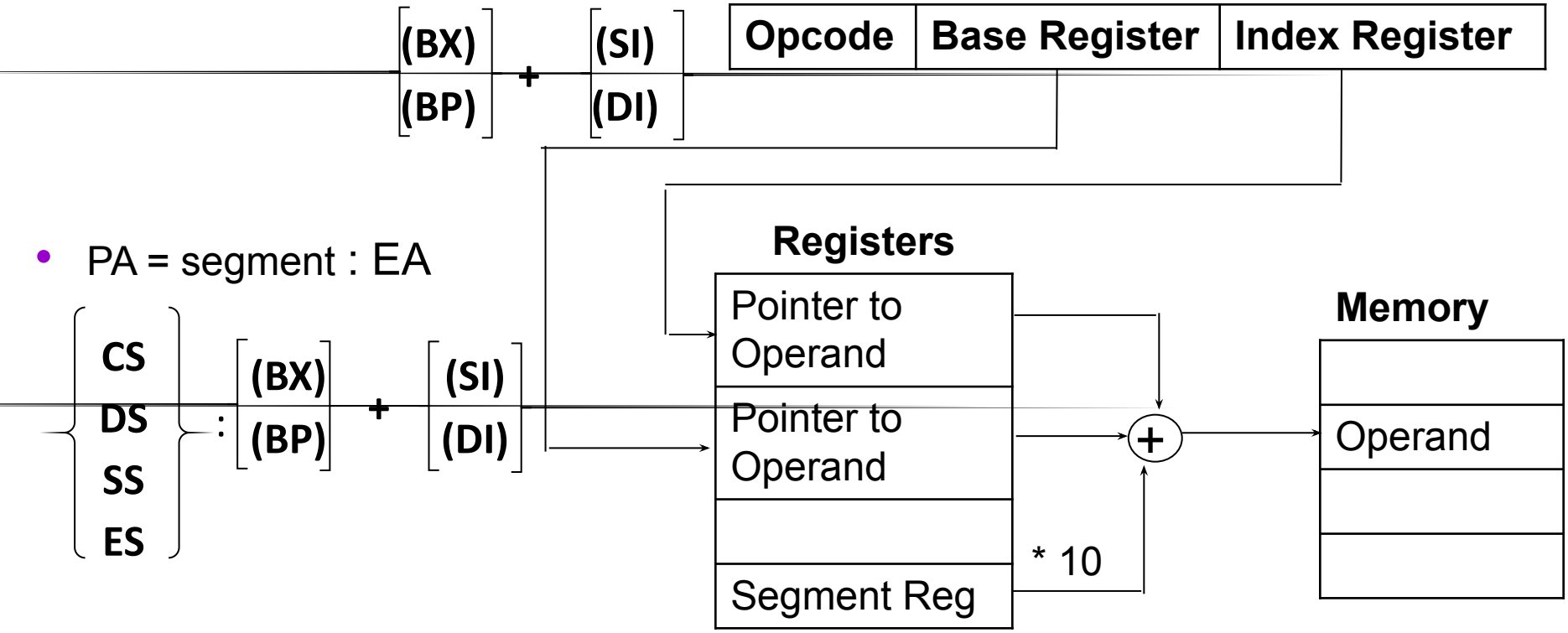
- Example: MOV [SI], AL
- Here, the contents of AL are copied to memory location whose offset address is in SI

The effective address of the data is given as 10H*DS+ [SI].

8086 - Addressing modes for Data Memory

3.3 Base Indexed Addressing Mode: a special case of the register indirect addressing mode.

- The offset of the operand is stored in one of the index registers.
- EA = {Base register} + {Index register}



- PA = segment : EA

8086 - Addressing modes for Data Memory

3.3 Base Indexed Addressing Mode:

- Example: MOV AX, [SI]

  Here, data is available at an offset address stored in SI in DS.

  The effective address is computed as 10H*DS+ [SI].

- Example: MOV DX, [BX+SI]

Moves a word from address pointed by BX+ SI in Data Seg to DX

Here, data is available at an offset address stored in BX+SI in DS.

The effective address is computed as 10H*DS+ [BX + SI].

- Example: MOV [BP+DI],BX

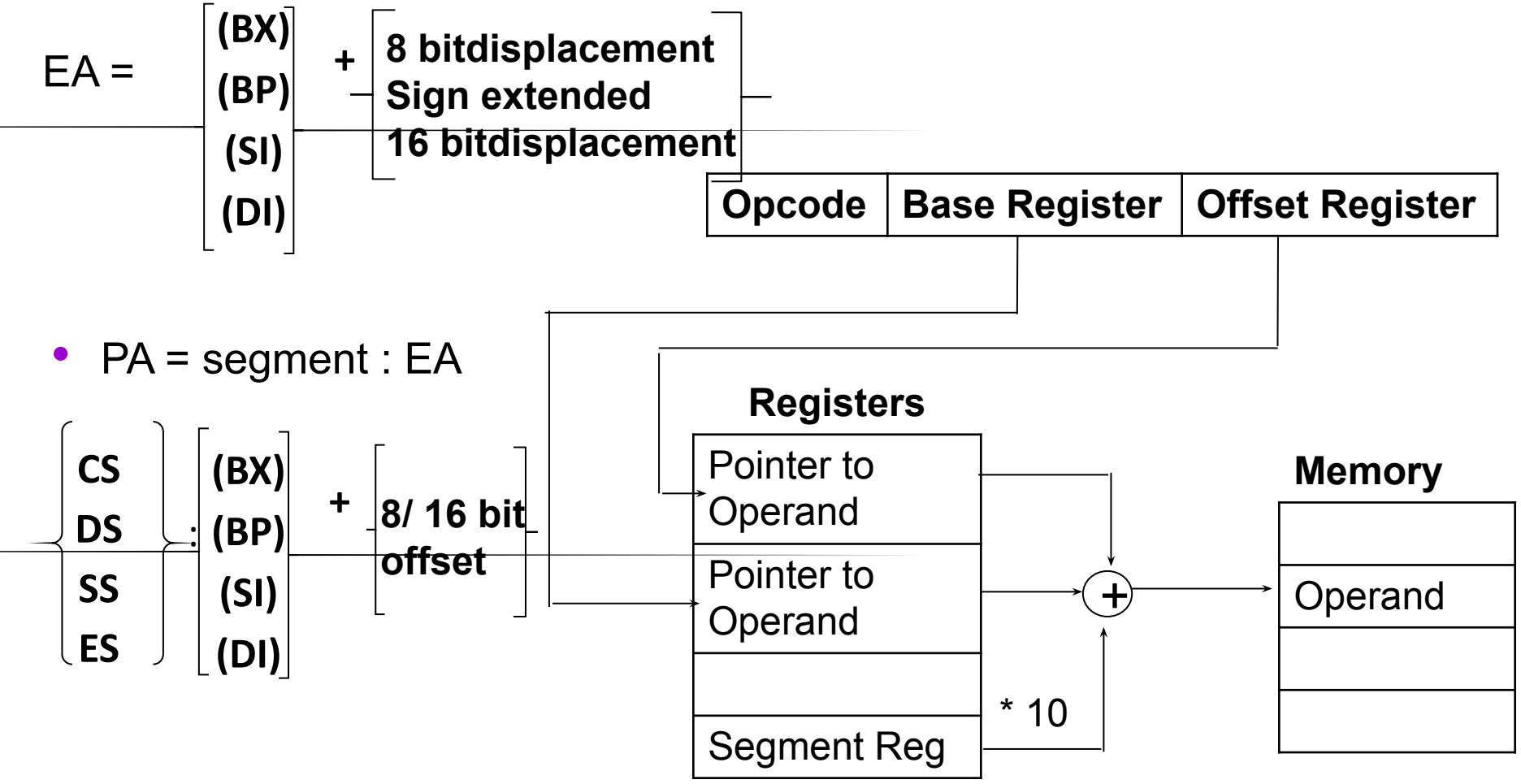
  The effective address is computed as 10H*SS+ [BP + DI].

3.4 Register Relative Addressing Mode:

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

8086 - Addressing modes for Data Memory

3.4 Register Relative Addressing Mode:



8086 - Addressing modes for Data Memory

3.4 Register Relative Addressing Mode:

- Example:

i. MOV AX, 50H [BX]: Copy the contents of memory location whose offset is given by **[BX]+ 50H displacement in DS to AX**

ii. MOV [BX + 1100],AL: Copy the contents of **AL register to a memory location whose offset** is given by [BX]+ 1100H displacement in DS

The effective address is given as 10H*DS+50H+ [BX] & SS*10H+BP+50H

8086 - Addressing modes for Data Memory

3.5 Relative Base Index Addressing Mode:

- The effective address of operand is calculated as an 8-bit or 16-bit displacement + the bases registers + index registers
- Base registers - BX or BP and any one of the index registers, in a default segment.

3.5 Relative Base Index Addressing Mode :

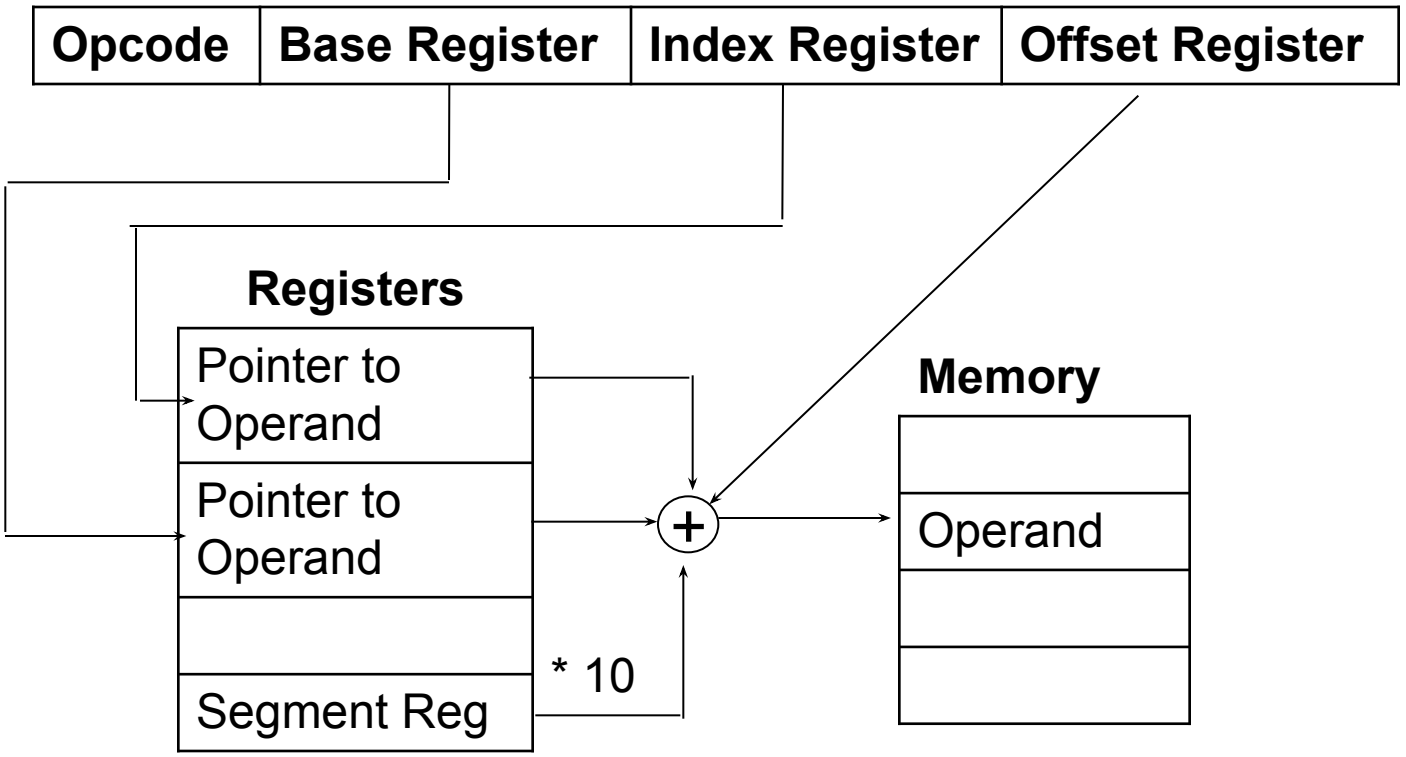
- EA = {Base register}+ {Index register}+{8-bit or 16-bit displacement }

$$\left[\begin{matrix}(BX)\\(BP)\end{matrix}\right] + \left[\begin{matrix}(SI)\\(DI)\end{matrix}\right] \pm \left[\begin{matrix}\text{8 /16 bit}\\\text{displacement}\end{matrix}\right]$$

- PA = segment : EA

$$\left\{\begin{matrix}CS\\DS\\SS\\ES\end{matrix}\right\} : \left[\begin{matrix}(BX)\\(BP)\end{matrix}\right] + \left[\begin{matrix}(SI)\\(DI)\end{matrix}\right] \pm \left[\begin{matrix}\text{8 /16 bit}\\\text{displacement}\end{matrix}\right]$$

3.5 Base Relative Addressing Mode:



3.5 Base Relative Addressing Mode:

Example:

- **MOV [BP] [SI] 2000H, AL:** Move the contents of AL into the memory location whose address is given by SI + BP+ [2000H] from the beginning of the SS to CX.

  The effective address of data is computed as 10H*SS+ [BP] + [SI] + 2000H.

3.5 Base Relative Addressing Mode:

Example:

- **MOV CX, [BX+SI + [0400]:** Move the contents of the memory location whose address is given by SI + BX+ [0400 H] from the beginning of the DS to CX.

  The effective address of data is computed as 10H*DS+ [BP] + [SI] + 0400 H.

4. String addressing

- SI points to 1st byte or word of source string & DS – default segment reg
- DI points to 1st byte or word of destination string & ES – default segment reg
- Do not use the normal memory addressing modes

8086 - Addressing modes for Data Memory

5. I/O Addressing modes:

- Basically used for Ios

- **2 types – Memory mapped I/O**

  - **I/O Mapped I/O: Direct port addressing**

    Indirect port addressing

6. Implied:

- In this addressing mode, the operands are implied and hence not specified in the instruction.

- Example: STC - Set Carry Flag,

  CLD – Clear Direction Flag

If [BX] = 0158H; [DI] = 10A5H; Displacement = 1B57H; [DS] = 2100H
Calculate EA and PA

1. Register AM
2. Immediate AM
3. Direct MAM: EA = 1B57H PA = DS *10 + DISPLACEMENT =21000H+1B57H= 22B57
4. Reg Indirect MAM : assume [BX] □ EA = [BX]=0158H
   PA = SEGMENT: OFFSET = DS: [BX] = 21158H
   and [DI] □ EA = [DI]=10A5H; PA = DS:[DI] = 220A5H
1. Reg Relative MAM: assume [BX] □ EA = DISPLACEMENT +[BX]
   = 1B57H+0158H =1CAFH; PA = DS: [[BX]+1B57H] = 22CAFH

   and [DI] EA = DIS +[DI]= 2BFCH PA =DS :[[DI]+1B57H]=23BFCH
1. Based Index MAM assume [BX] EA= [BX] +[DI]= 11FDH PA =DS: [[BX+[DI]]= 221FDH
1. Relative Base Index MAM assume [DI] EA =

If [BX] = 0158H; [DI] = 10A5H; Displacement = 1B57H; [DS] = 2100H
Calculate EA and PA

Based Index MAM assume [BX] EA= [BX] +[DI]= 11FDH PA =DS: [[BX+[DI]]= 221FDH

Relative Base Index MAM assume [DI] EA = [BX]+[DI]+ DISPLACEMENT = 2D54H

PA = DS: [[BX]+[DI]+ DISPLACEMENT] = 23D54H

MOV CX , SS: [BX]

Here, the concept of Segment Override Prefix is being used. Although the default segment for the offset BX is DS, as the SS is mentioned in the instruction, it is overriding the default segment. Hence, the Stack Segment (SS) register is being used here.

8086 - Addressing modes for Stack Memory

II. Addressing modes for Stack Memory

1. Register Addressing mode:

The operands are specified in reg (ONLY 16 bit reg.)

Eg: PUSH BX

Transfers [BH] at addr. location pointed by SP-1 & [BL] to SP-2 in Stack seg.

2. Register Indirect Addressing mode

The addr. of the operands is specified in reg (ONLY 16 bit reg.)

Eg: PUSH [BX]

Transfers a byte from the addr. location pointed by BX & BX+1 in Data Seg to SP-1 & SP-2 in Stack seg.

8086 - Addressing modes for Stack Memory

3. Flag Addressing mode

The contents of the flag reg are transferred to and from stack.

Eg: PUSH F

Transfers higher byte of the flag reg. to SP-1 and lower byte to SP-2 in Stack seg.

4. Segment Register Addressing mode:

The segment reg. (except CS) are transferrred to and from the Stack.

Eg: PUSH DS

Transfers higher byte of the DS reg to SP-1 and lower byte to SP-2 in Stack seg.

8086 - Addressing modes for Program Memory

1. **Intra-segment direct mode:**

- The address to where the control is transferred is specified directly in the instruction as an immediate displacement value.
- The address to which the control is to be transferred lies in the **same segment** in which the control transfer instruction lies
- Only **IP changes, CS does not change**
- Also called Relative addr.mode.
- The **effective address to which the control will be transferred is given by the sum of 8 or 16 bit displacement and current content of IP**.

Eg: JMP Label1

8086 - Addressing modes for Program Memory

2. Intra-segment indirect mode:

- The branch address is **specified in a register or a memory location** (in DS only)
- As intra seg only IP changes, CS does not change
- Eg: JMP Label1 [BX]
- IP ← {DS:[BX], DS:[BX+1]}

8086 - Addressing modes for Program Memory

3. Inter-segment direct mode:

- The new branch address is specified in the instruction.
- In this mode, the address to which the control is to be **transferred is in a different segment.**
- This addressing mode provides a means of branching from one code segment to another code segment.
- CS and IP of the destination address are specified directly in the instruction.
- Eg: JMP Label
- IP ← offset address of Label
- CS ← Segment address of Label

8086 - Addressing modes for Program Memory

4. Inter-segment indirect mode:

- The new branch address is **specified indirectly in a register or a memory location (in DS only)**
- Both **CS & IP get new values**.
- CS and IP of the destination address are specified directly in the instruction.
- Eg: JMP Label [BX]
- IP ← {DS:[BX], DS:[BX+1]}
- CS ← {DS:[BX+2], DS:[BX+3]}

8086 - Instruction Set