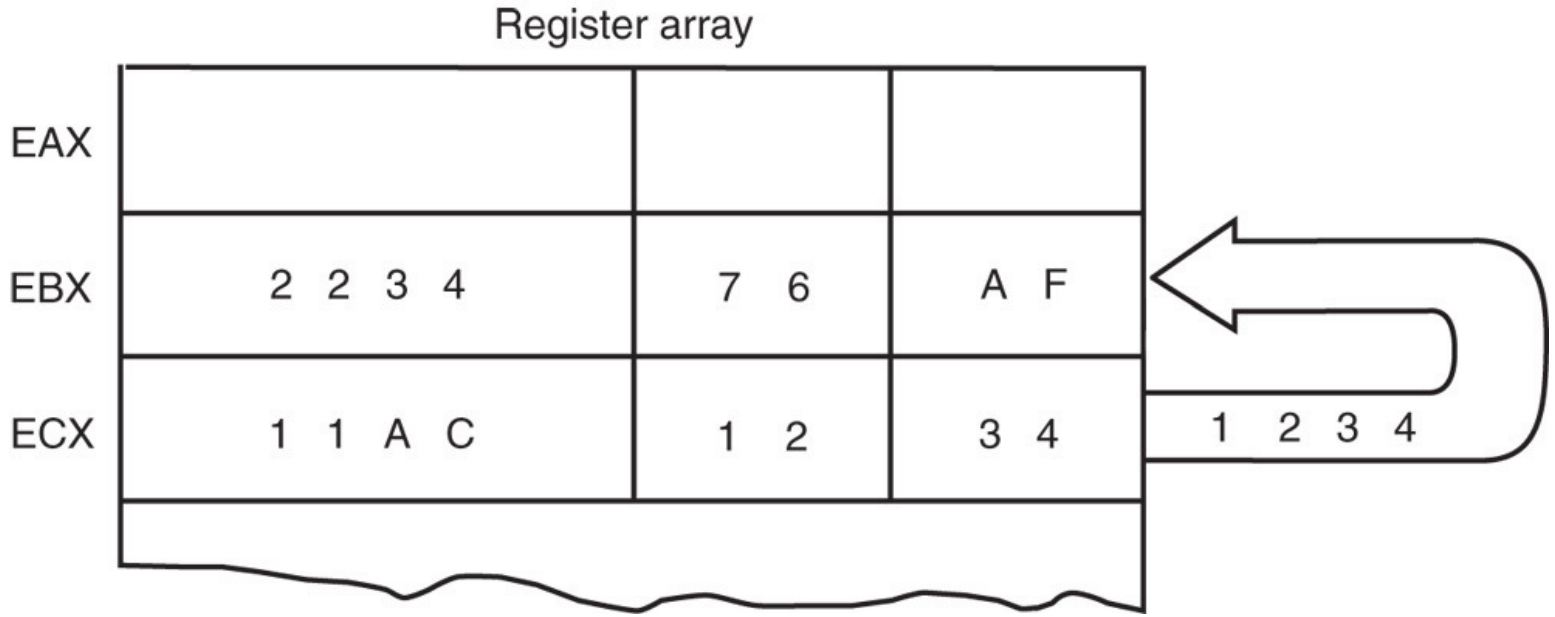
ADDRESSING MODES/8086

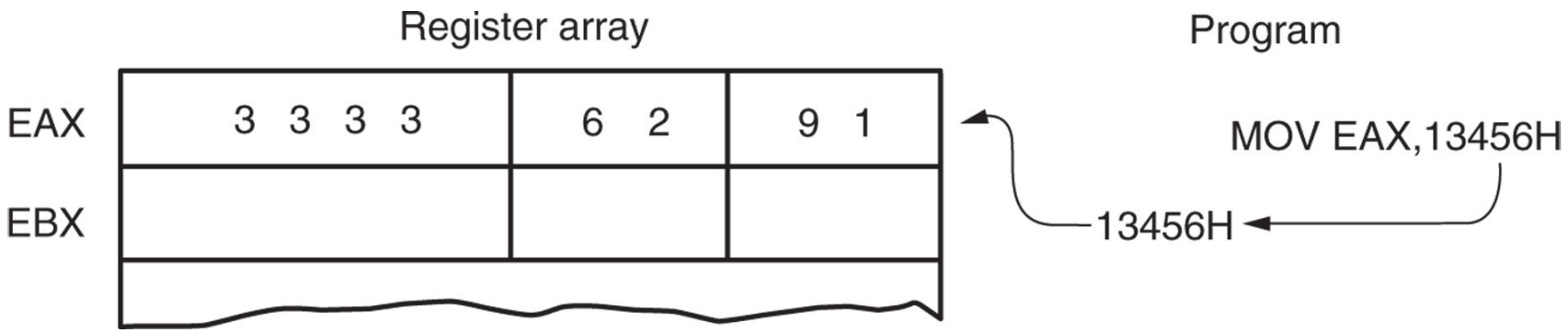
-Mrs. JYOTSNA A. KULKARNI

The effect of executing the MOV BX, CX.



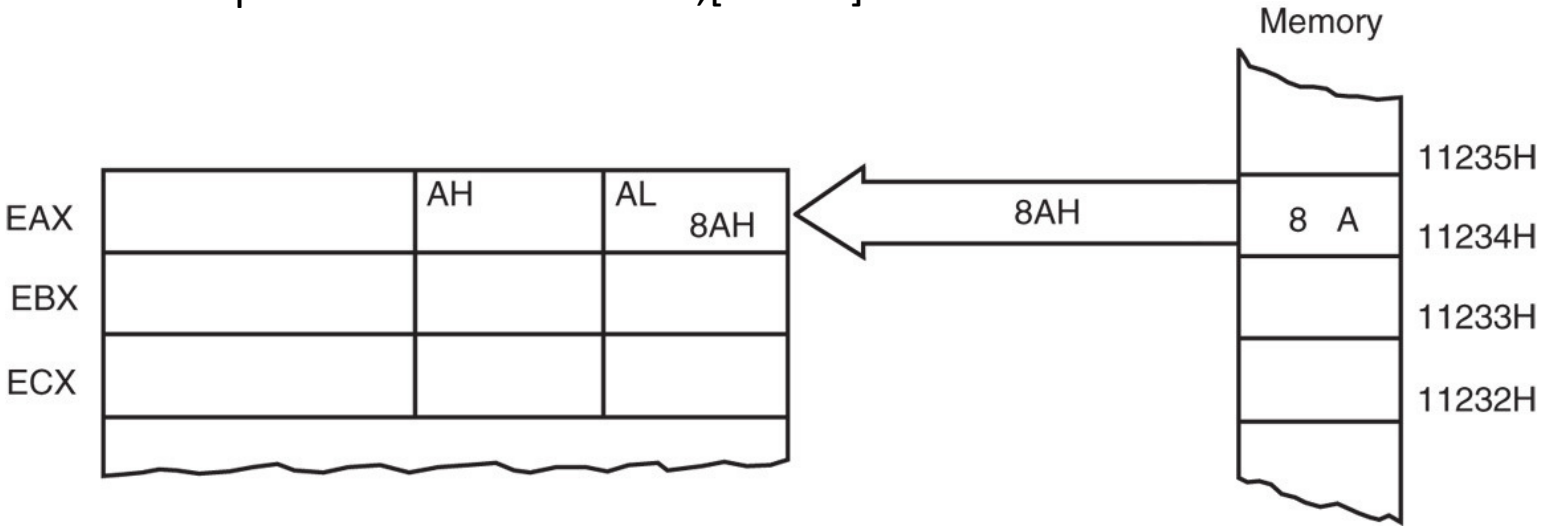
Register Addressing?

The operation of the MOV AX,3456H instruction.



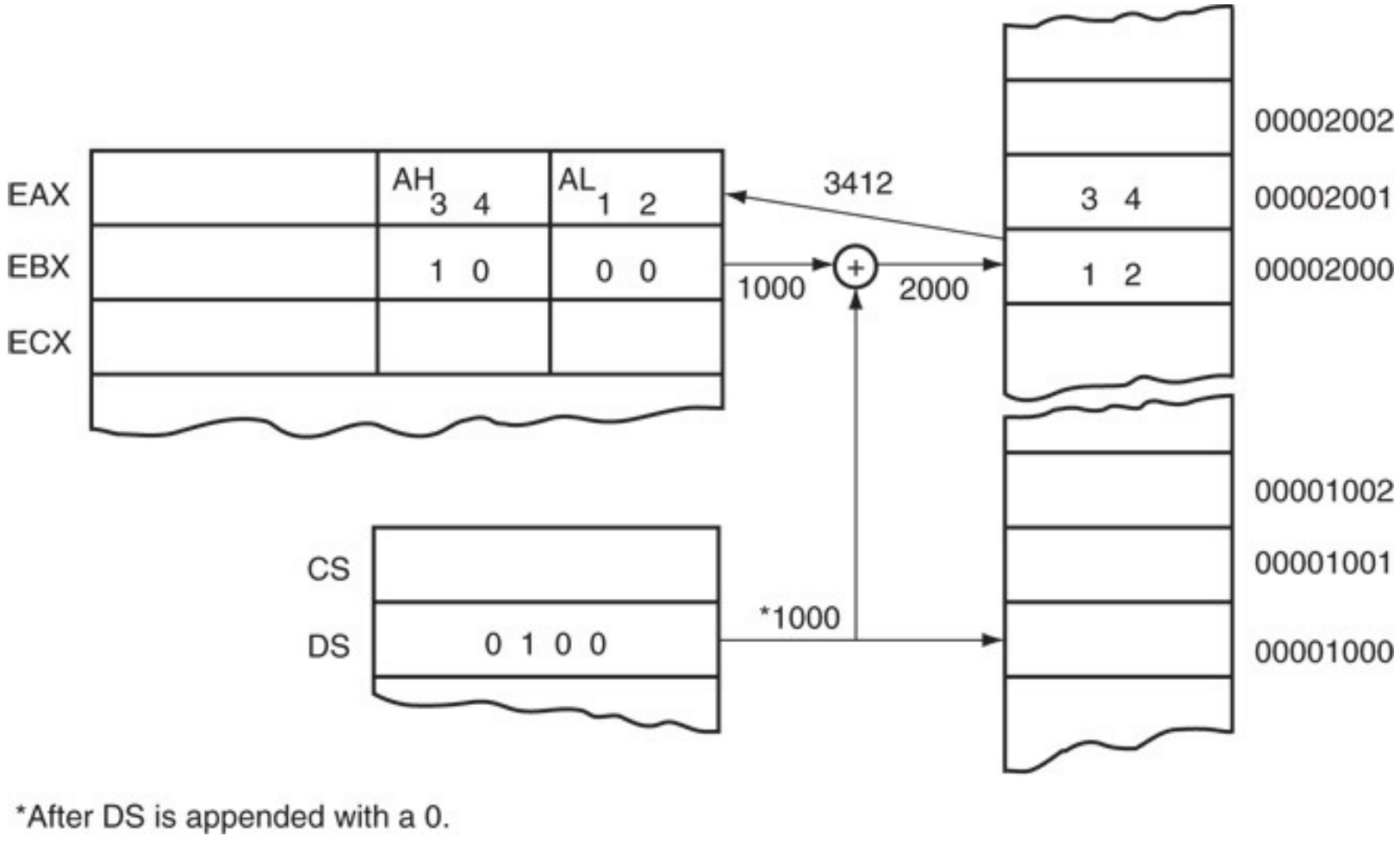
- Immediate addressing?

The operation of the MOV AL,[1234H] instruction when DS=1000H .



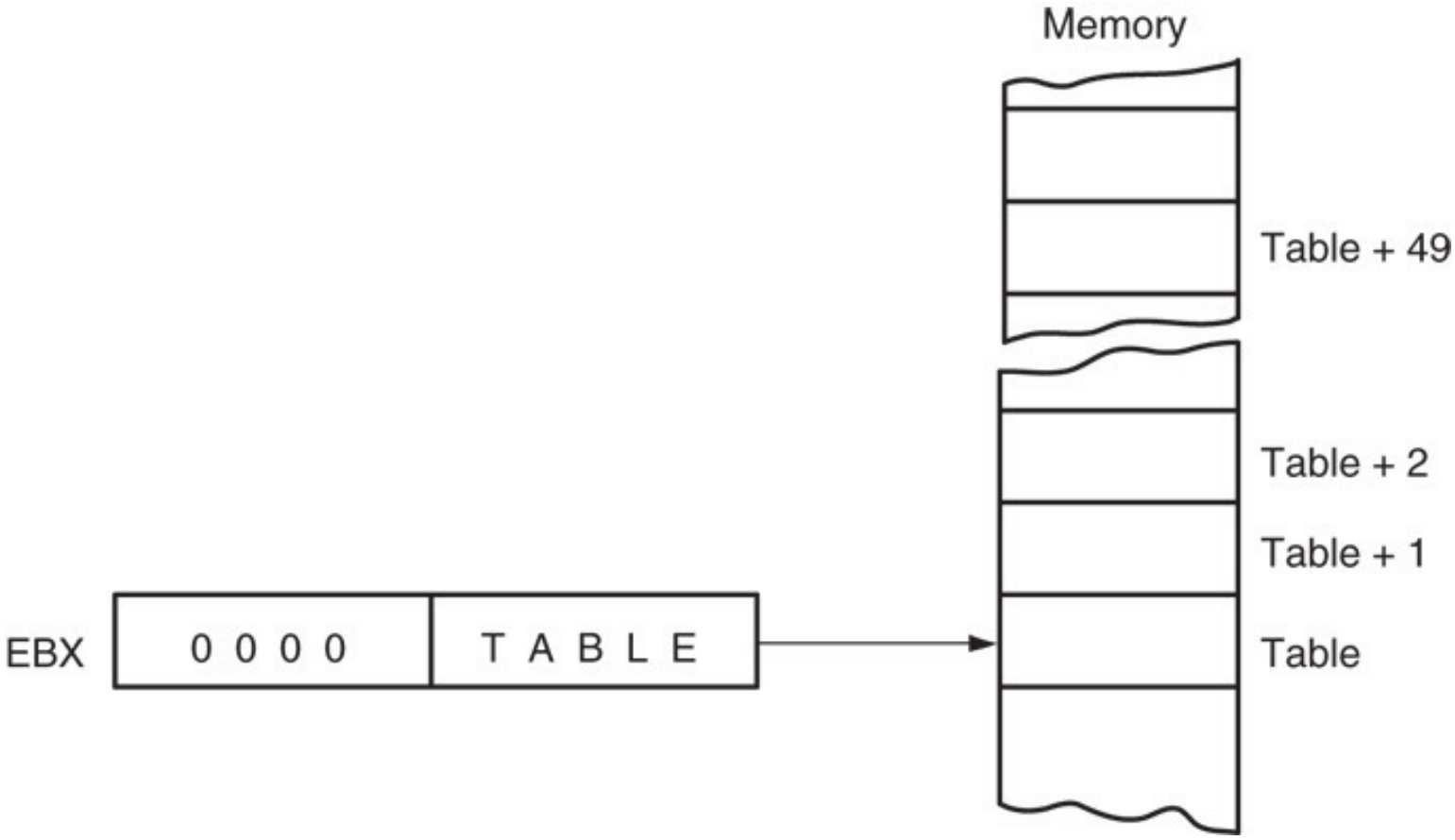
- Direct addressing?

The operation of the MOV AX,[BX] instruction when BX = 1000H and DS = 0100H. Note that this instruction is shown after the contents of memory are transferred to AX.



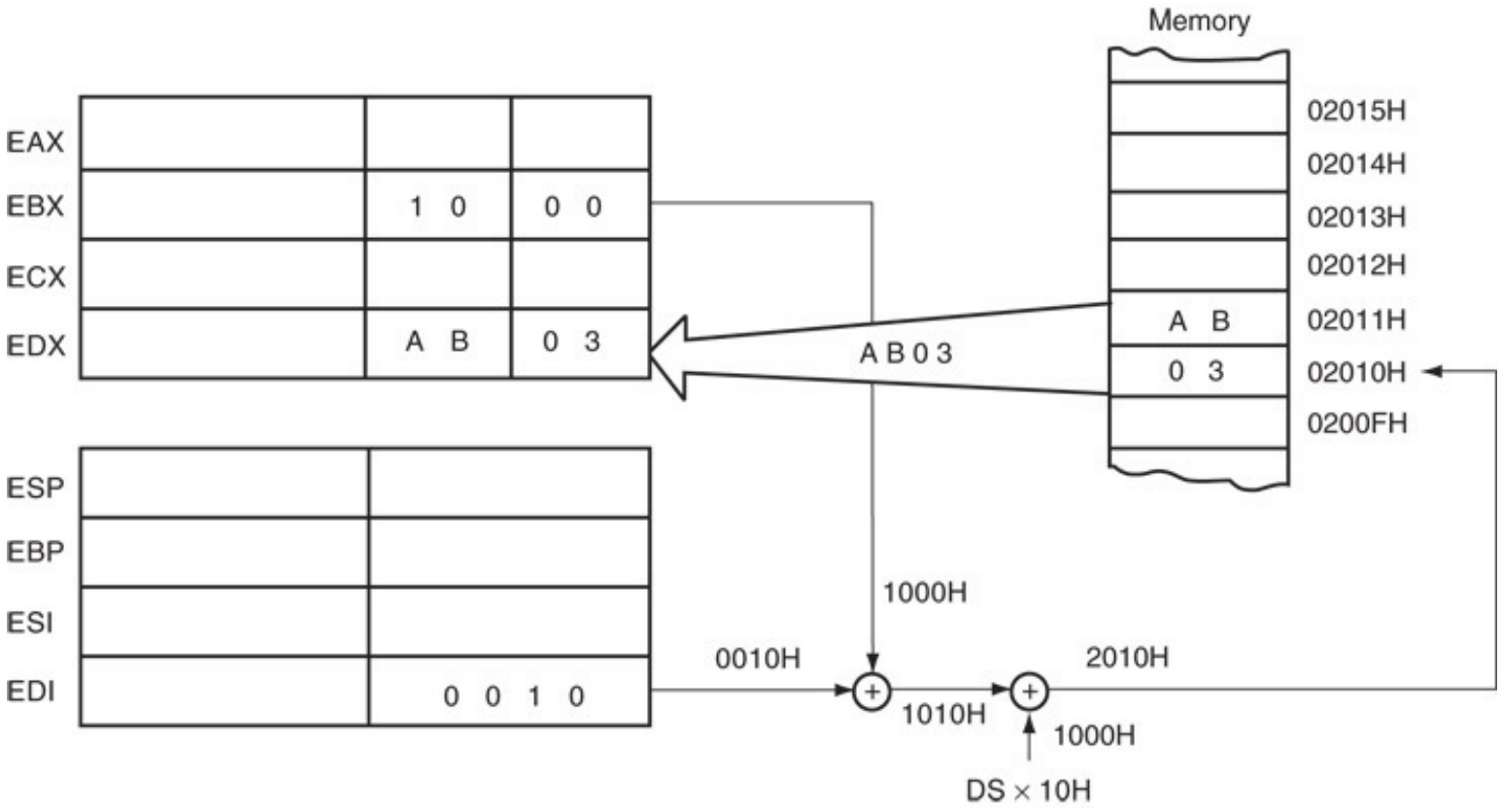
Register Indirect Addressing?

Figure 3–7 An array (TABLE) containing 50 bytes that are indirectly addressed through register BX.



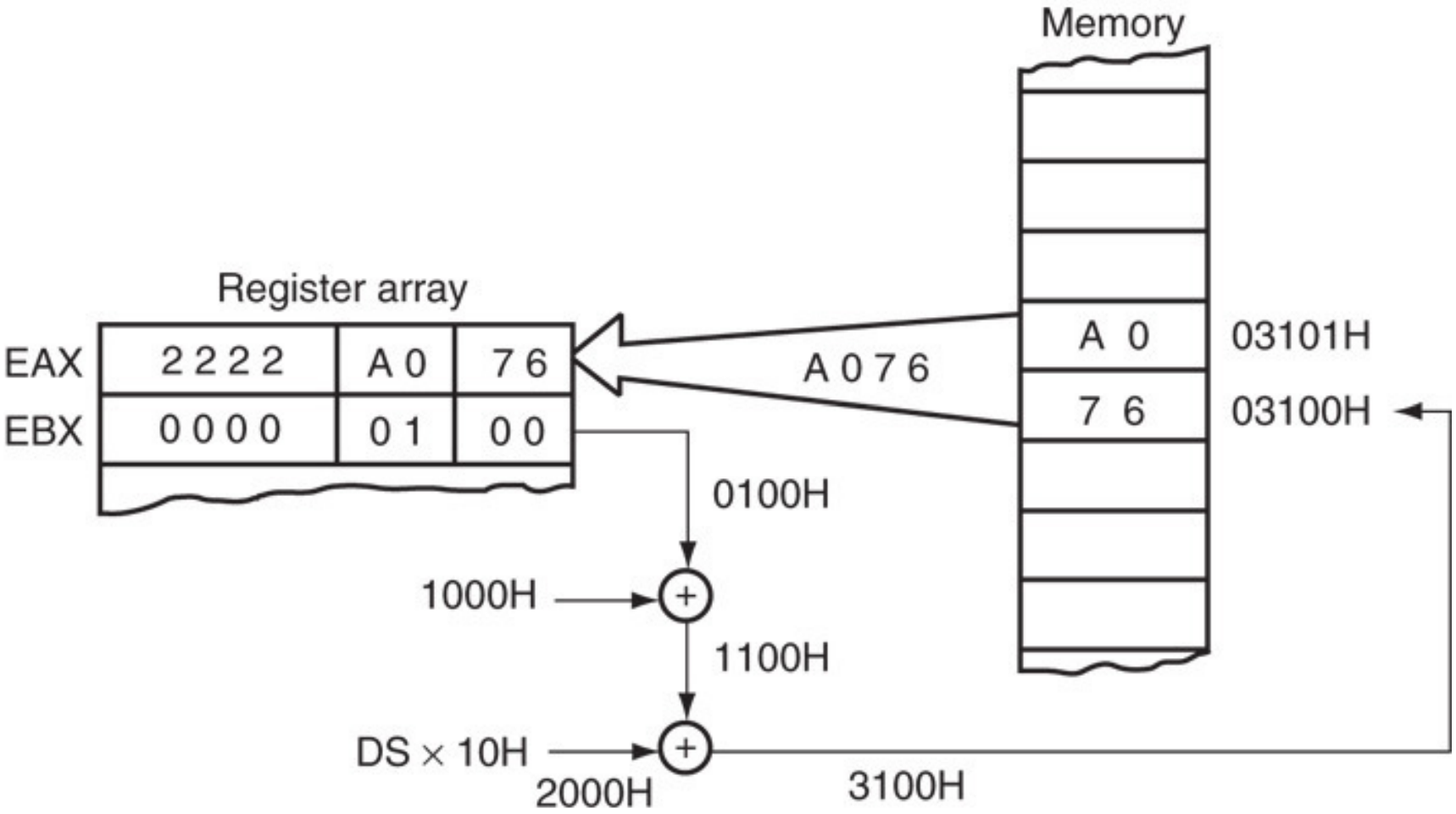
Register Indirect Addressing?

MOV DX,[BX + DI] instruction. Since DS=0100H, BX=100H and DI=0010H.

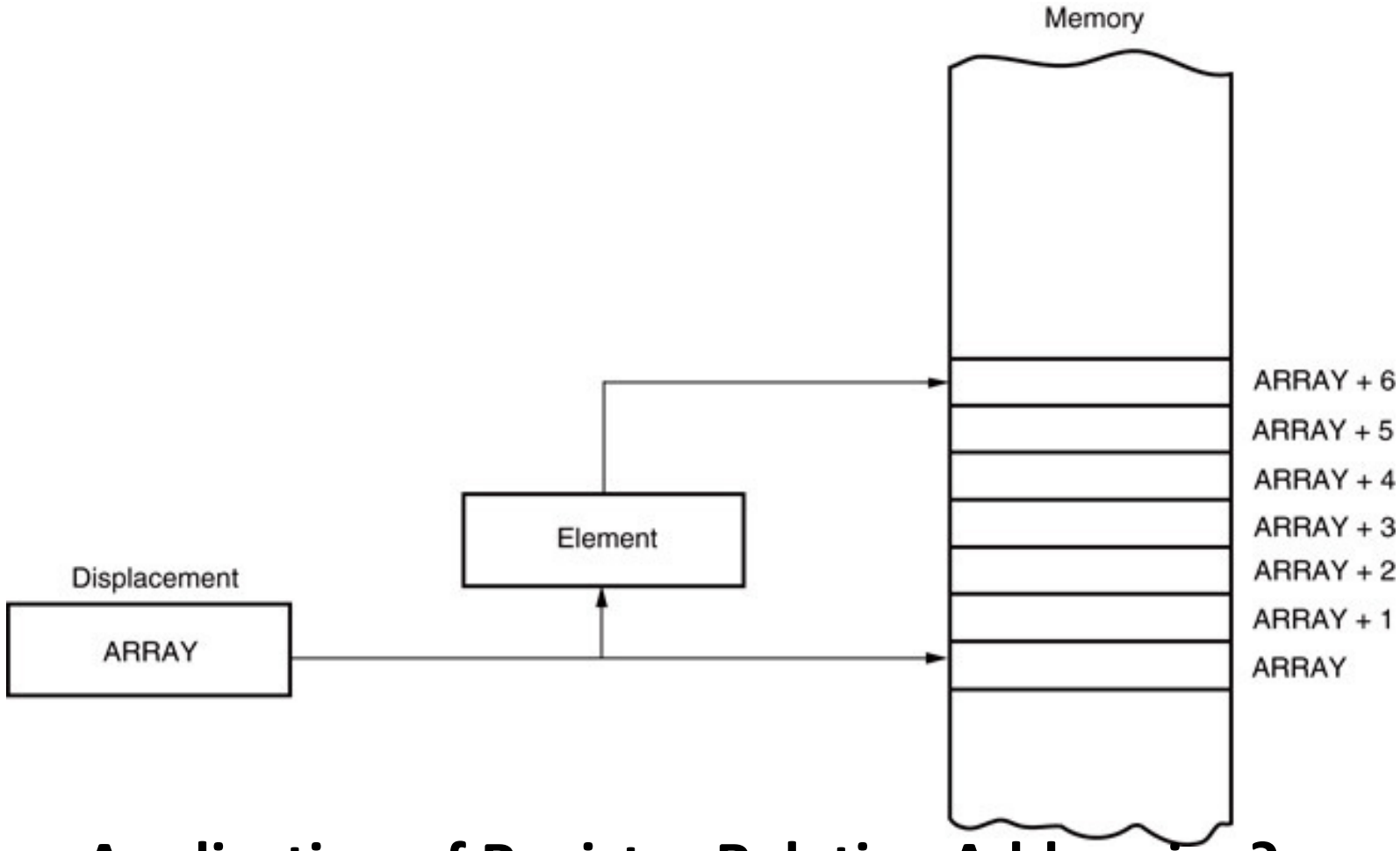


Base-plus-index addressing?

The operation of the MOV AX, [BX+1000H] instructon, when BX=1000H and DS=0200H .



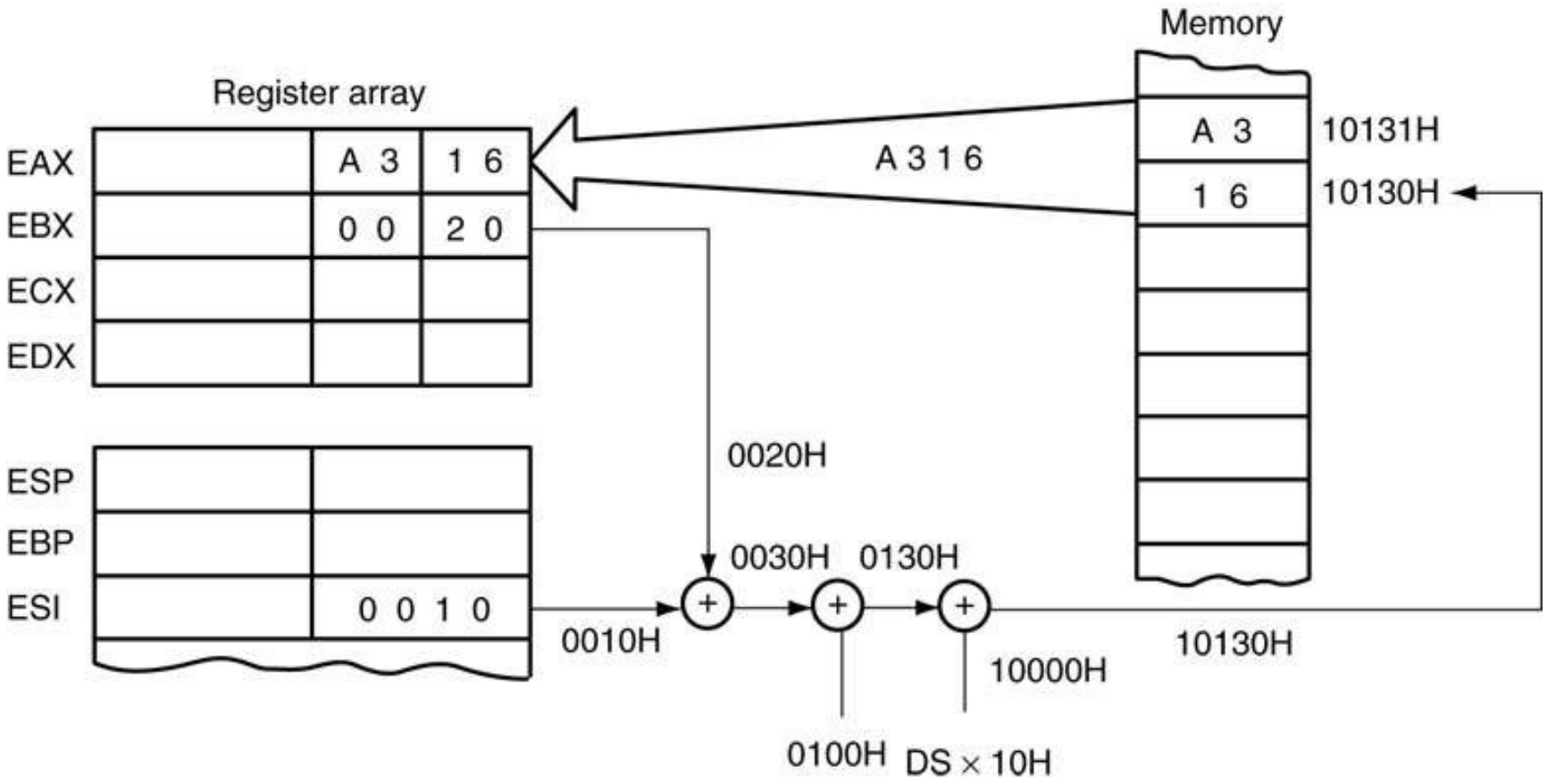
Register Relative Addressing?



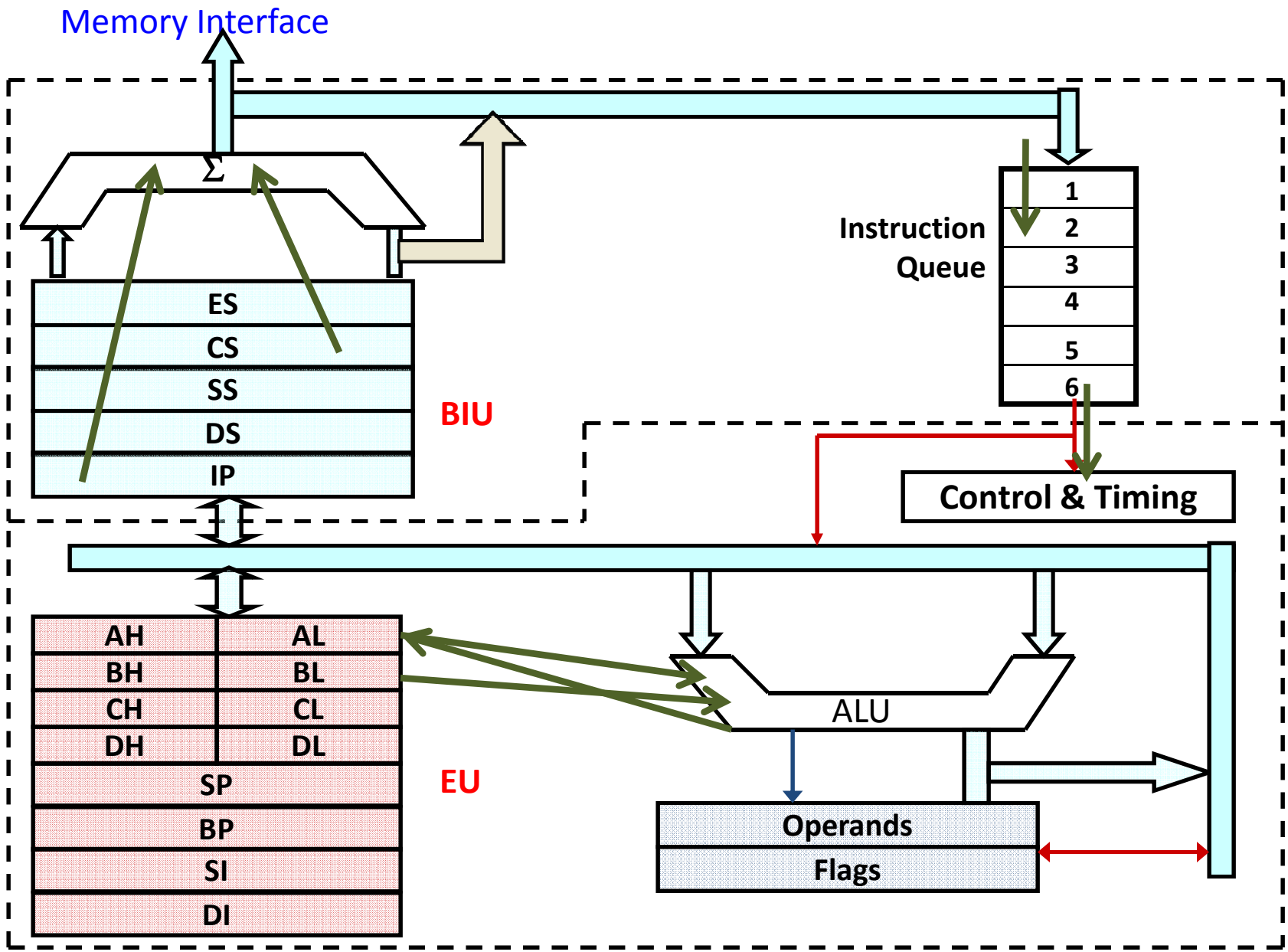
Application of Register Relative Addressing?

To address an element of ARRAY

MOV AX,[BX+SI+1000H] instruction. DS=1000H



Base relative-plus-index addressing?



Block Diagram of CPU of 8086

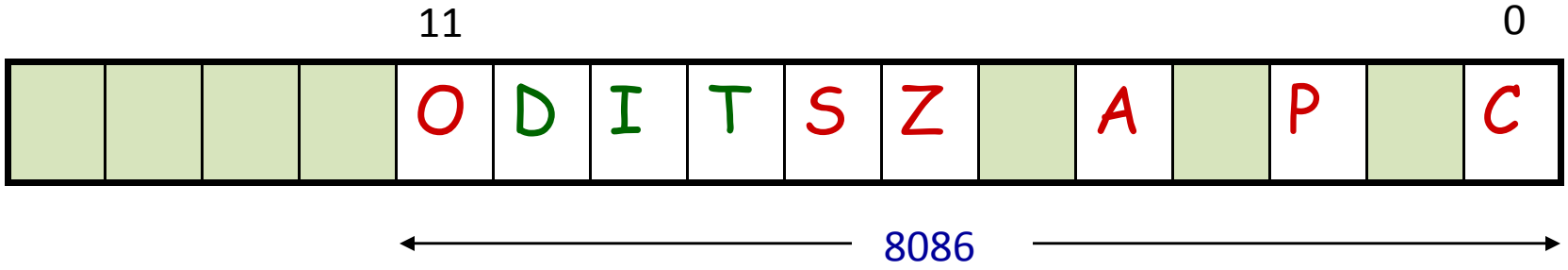
MULTIPURPOSE REGISTERS

AX, BX, CX, DX, BP, DI, SI

SPECIAL PURPOSE REGISTERS

IP, SP, FLAGS

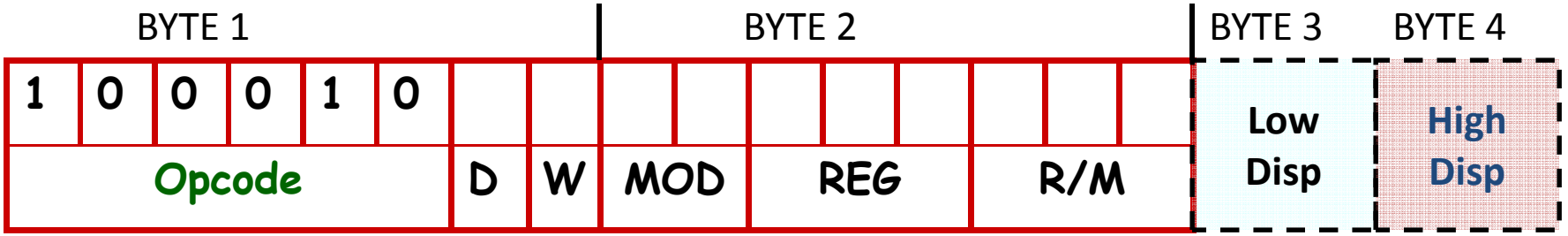
CS, DS, SS, ES
(Segment Registers)



The Processor executes an instruction – it performs the specified function on data

Data- called operands

May be a part of the instruction
May reside in one of the internal registers of the µp
May be stored at an address in memory



D = 0 (Direction from Reg)

= 1 (Direction to Reg)

W = 0 (Data – byte)

= 1 (Data – word)

AX/AL	000
BX/BL	011
CX/CL	001
DX/DL	010
SP/AH	100
BP/CH	101
SI/DH	110
DI/BH	111

OR

Dir Addr LB	Dir Addr HB

MOD + R/M - Addressing Modes

Effective address can be specified directly in the instruction OR contained in a register OR Sum of one or two registers and a displacement

Coding template for 8086 instructions which MOV data between Registers or between register & memory location.

NOTE: At least TWO CODE BYTES are required for the instruction

MOV INSTRUCTION TEMPLATE

MOD / R/M	00	01	10	11	
				W = 0	W = 1
000	[BX] + [SI]	[BX]+[SI] +d8	[BX]+[SI]+d16	AL	AX
001	[BX] + [DI]	[BX]+[DI]+d8	[BX]+[DI]+d16	CL	CX
010	[BP] + [SI]	[BP]+[SI]+d8	[BP]+[SI]+d16	DL	DX
011	[BP] + [DI]	[BP]+[DI]+d8	[BP]+[DI]+d16	BL	BX
100	[SI]	[SI]+d8	[SI]+d16	AH	SP
101	[DI]	[DI]+d8	[DI]+d16	CH	BP
110	d16 (direct address)	[BP] + d8	[BP] + d16	DH	SI
111	[BX]	[BX]+d8	[BX]+d16	BH	DI

- What is code for following instructions?
- MOV SP,BX
  - Opcode = 100010
  - Opcode D= 100010 1
  - Opcode D W=100010 1 1
  - Opcode D W mod =100010 1 1 11
  - Opcode D W mod REG=100010 1 1 11 100
  - Opcode D W mod REG R/M= 100010 1 1 11 100 011
  - **1000 10 1 1 11 100 011**
  - 8 B E 3
- MOV 43H[SI],DH
  - ?
- What is the length of each instruction?

HA: Find instruction code for following instructions

MOV CL, DH

MOV CL, CH

MOV AL, 0

MOV BH, 100

MOV AL, ‘A’

MOV [DI], BH

MOV [BP], DL

- Data Transfer Instructions
- Arithmetic Instructions
- Logical Instructions
- Branch and Program control Instructions

MOV Instruction

MOV destination , source

ASSEMBLY LANGUAGE PROGRAMME INSTRUCTION FORMAT

LABEL	OP CODE	OPERAND	COMMENT
NEXT:	ADD	AL, 07H	; ADD a number

- All labels must begin with
  - A letter
  - or one of the following special characters: @, $, -, or ?.
  - a label may any length from 1 to 35 characters
- comments always begin with a semicolon (;)

Data Structures

- Used to specify how information is stored in a memory array.
  - a template for data
- The start of a structure is identified with the STRUC assembly language directive and the end with the ENDS statement.

PROGRAM MEMORY-ADDRESSING MODES

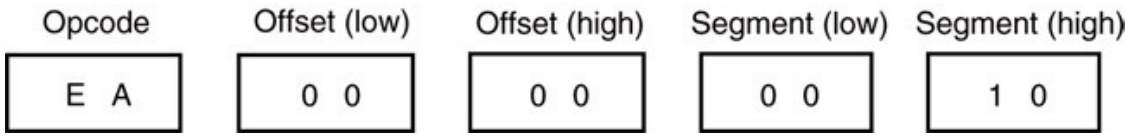
- Used with the JMP (jump) and CALL instructions.
- Consist of three distinct forms:
  - direct, relative, and indirect

JMP Instruction

- **JMP - Go to specified address to get next instruction**
  - **Conditional**
  - **Unconditional (jmp)**
    - **JMP BX , JMP WORD PTR [BX] ,**
    - **JMP NEXT**
      - **Location of label NEXT**
        - » **NEXT is in same segment (short)**
          - **IP related**
            - **Direct**
            - **Opcode =E9 if IP <- IP+d16**
            - **Opcode =E8 if IP <- IP+d8**
            - **Indirect**
              - **Opcode =FF if IP <- Reg 16**
              - **Opcode =FF if IP <- Mem 16**
        - » **NEXT is in some another segment(inter-seg /long)**
          - **IP <- offset**
          - **CS ,-segment base**
          - **Opcode =EA for Direct & FF for indirect addressing**

The 5-byte machine language version of a JMP [10000H] instruction.

Unconditional inter-segment direct JMP



CALL instruction

- CALL LABEL - Call a procedure (subprogram),save return address on stack
  - **Direct**
  - **Opcode =E8 if IP <- IP+Disp(16)-SP <- return**
  - **Opcode =FF if IP <- Reg(16) –SP<-return**
  - **Indirect**
    - **Opcode =FF if IP <- Reg 16 –SP<-return**
    - **Opcode =FF if IP <- Mem 16 –SP<-return**

» **LABEL is in some another segment(inter-seg / far)**
- **IP <- offset**
- **CS ,-segment base**
- **Opcode =9A for Direct& FF for indirect addressing**