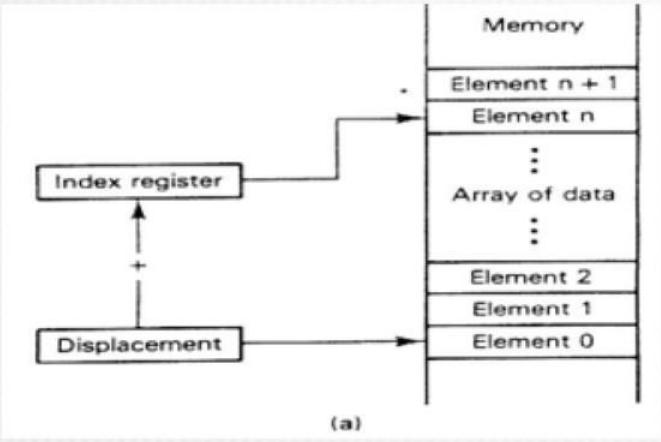
e. Indexed with displacement Addressing Mode

- works the same as the based relative addressing mode, except that registers DI and SI hold the offset address
- Examples:
  1. MOV DX, [SI]+5 ;PA = DS (shifted left) + SI + 5
  2. MOV CL, [DI]+20 ;PA = DS (shifted left) + DI + 20



(a)

PA = Segment base: Index + Displacement

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

(b)

Example:

- 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.

1. MOV [BX]+20, AX
2. MOV [SI]+10, AX
3. MOV [DI]+4, AX
4. MOV [BP]+12, AX

- Solution: In each case

PA = segment reg. (shifted left) + (offset reg.) + displacement

1. DS:BX+20 location 47120 = (12) and 47121 = (25)
2. DS:SI+10 location 46496 = (12) and 46497 = (25)
3. DS:DI+4 location 4D504 = (12) and 4D505 = (25)
4. DS:BP+12 location 27826 = (12) and 27827 = (25)

f. Base-Indexed with displacement Addressing Mode

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

- 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

5. Stack addressing mode

PUSH ,POP , PUSHF and POPF instructions.

Example:

```
PUSH    BX
POP     [SI]
```

6. String addressing mode

MOVS,LODS,STOS,CMPS and SCAS instructions.

7. input output addressing mode

IN , OUT instructions.

8. Implied addressing mode

XLAT , AAA , DAA , AAD , AAM and DAS instructions.

Important Notes

- The coding of the instructions above can vary; for example, the last example could have been written as:
  - MOV AH, [BP+SI+29]; or
  - MOV AH, [SI+BP+29] ;
  - *the register order does not matter.*
  - MOV AX, [SI][DI] + displacement is illegal.
- In many of the examples above, the *MOV instruction was used for the sake of clarity*, even though one can use any instruction as long as that instruction supports the addressing mode. For example, the instruction
  - ADD DL, [BX] would add the contents of the memory location pointed at by DS:BX to the contents of register DL.

Segment Overrides

- 80X86 allows the program to override the default segment registers
  - Specify the segment register in the code

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