CMPE 310
Systems Design and Programming

L19: Chapter 3 – *Addressing Modes*

UMBC
AN HONORS UNIVERSITY IN MARYLAND

L19 Objectives

* Addressing modes
  * Recognize the addressing modes of the x86

Review: Addressing modes

- Addressing modes: how data is specified
  - Register: MOV AX, BX
  - Immediate: MOV AX, 1234H
  - Memory
    - Direct: MOV AX, [1010H]
    - Will cover remaining modes today
    - Can have at most one memory operand
    - Segment base assumed to be DS; can override
- How do we specify how many bytes we are moving?

Register Indirect Addressing Mode

PA = Segment base: Indirect address

PA = { CS, DS, SS, ES, FS, GS } : { BX, BP, SI, DI }

- Register indirect addressing mode
  - Similar to direct addressing in that the effective address is combined with the contents of DS to obtain the physical address
  - Effective address resides in either a base or index register
- Physical address computation

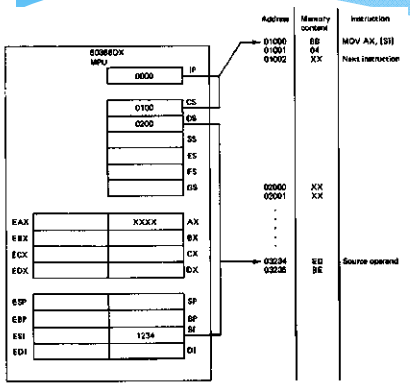
  PA = SBA:EA → 20-bit address

  PA = SBA:[Rx] → 16-bit offset
  - Segment base address is DS by default

    PA = DS:[Rx]
  - Segment override prefix (SEG) is required to enable use of another segment register

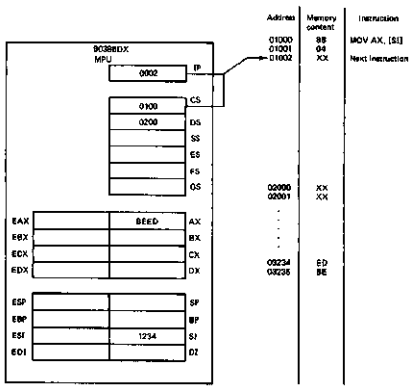
    PA = ES:[Rx]

Register Indirect Addressing Mode (example: MOV AX, [SI])



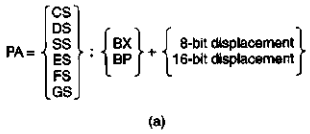
* State before fetch and execution

Register Indirect Addressing Mode (example: MOV AX, [SI])

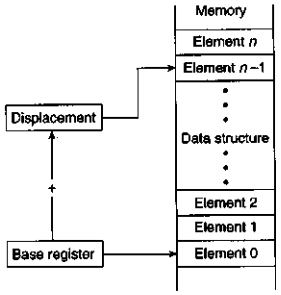


* State after execution

Base Addressing Mode



(a)

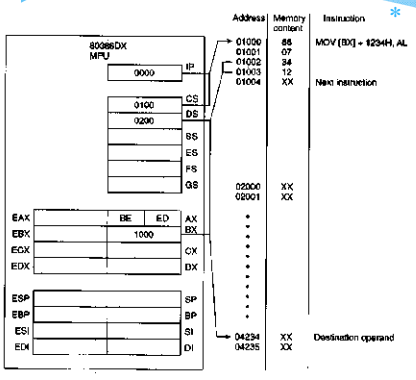


(b)

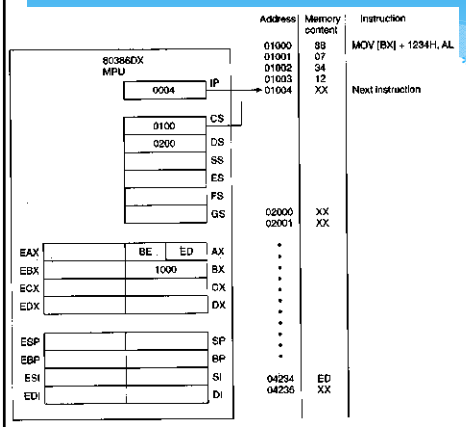
- **Based addressing mode**
  - Effective address formed from contents of a base register and a displacement
    - Base register is either BX or BP (stack)
    - Direct/indirect displacement is 8-bit or 16bit
- **Physical address computation**
  - PA = SBA:EA → 20-bit address
  - PA = SBA:[BX or BP] + DA
- **Accessing a data structure**
  - Based addressing makes it easy to access elements of data in an array
  - Address in base register points to start of the array
  - Displacement selects the element within the array
  - Value of the displacement is simply changed to access another element in the array
  - Program changes value in base register to select another array

Base Addressing Mode (Example: MOV [BX] + 1234H, AL)

- State before fetch and execution

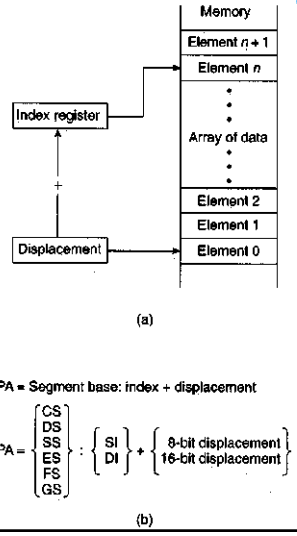


Base Addressing Mode (Example: MOV [BX] + 1234H, AL)



* State after execution

Indexed Addressing Mode



(a)

PA = Segment base: index + displacement

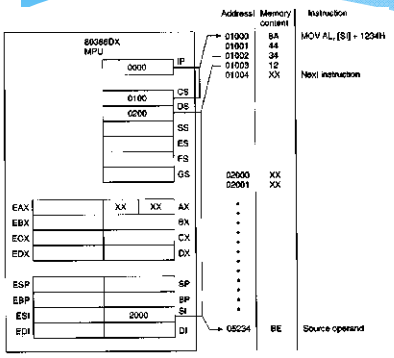
$$PA = \begin{Bmatrix} CS \\ DS \\ SS \\ ES \\ FS \\ GS \end{Bmatrix} : \begin{Bmatrix} SI \\ DI \end{Bmatrix} + \begin{Bmatrix} 8\text{-bit displacement} \\ 16\text{-bit displacement} \end{Bmatrix}$$

(b)

* **Indexed addressing mode**
  * Similar to based addressing, it makes accessing elements of data in an array easy
  * Displacement points to the beginning of array in memory
  * Index register selects element in the array
  * Program simply changes the value of the displacement to access another array
  * Program changes (recomputes) value in index register to select another element in the array
* **Effective address formed from direct displacement and contents of an index register**
  * Direct displacement is 8-bit or 16-bit
  * Index register is either SI → source operand or DI → destination operand
* **Physical address computation**
  PA = SBA:EA → 20-bit address
  PA = SBA: DA + [SI or DI]

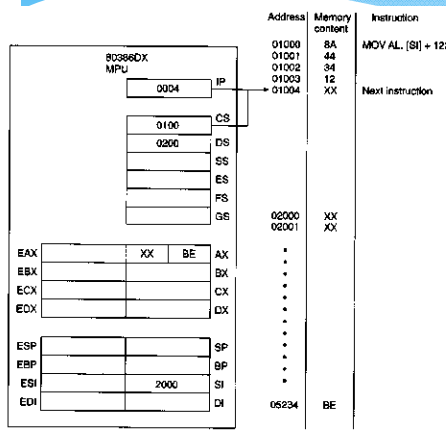
Indexed Addressing Mode (Example: MOV AL,[SI] +1234H)

* State before fetch and execution

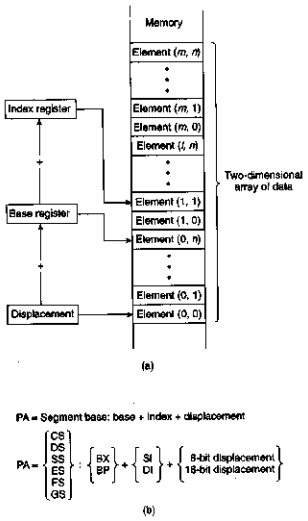


Indexed Addressing Mode (Example: MOV AL,[SI] +1234H)

* State after execution

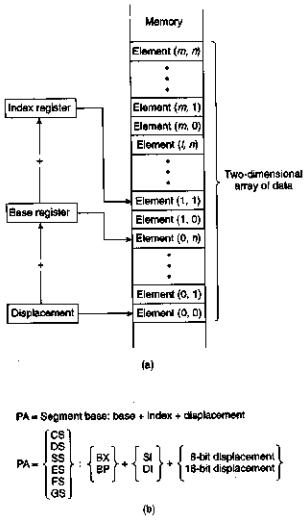


Based-Indexed Addressing Mode



- Combines the functions of based and indexed addressing modes
- Enables easy access to two-dimensional arrays of data
- Displacement points to the beginning of array in memory
- Base register selects a row (*m*) of elements
- Index register selects an element in column (*n*)
- Program simply changes the value of the displacement to access another array
- Program changes (recomputes) value in base register to select another row of elements
- Program changes (recomputes) the value of the index register to select the element in another column

Based-Indexed Addressing Mode



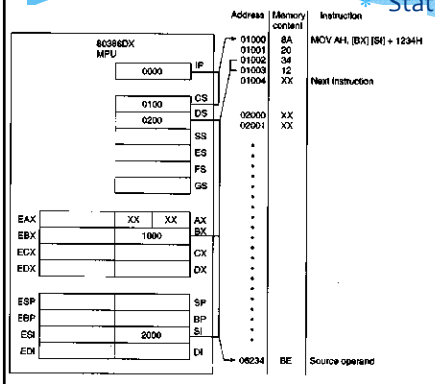
- Effective address formed from direct displacement and contents of a base register and an index register
  - Direct displacement is 8-bit or 16bit
  - Base register either BX or BP (stack)
  - Index register is either SI → source operand or DI →destination operand
- Physical address computation

  PA = SBA:EA → 20-bit address

  PA = SBA:DA + [BX or BP] + [SI or DI]

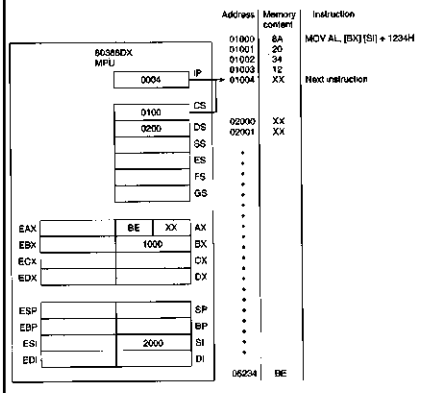
Based- Indexed Addressing Mode Example(MOV AH,[BX][SI] +1234H)

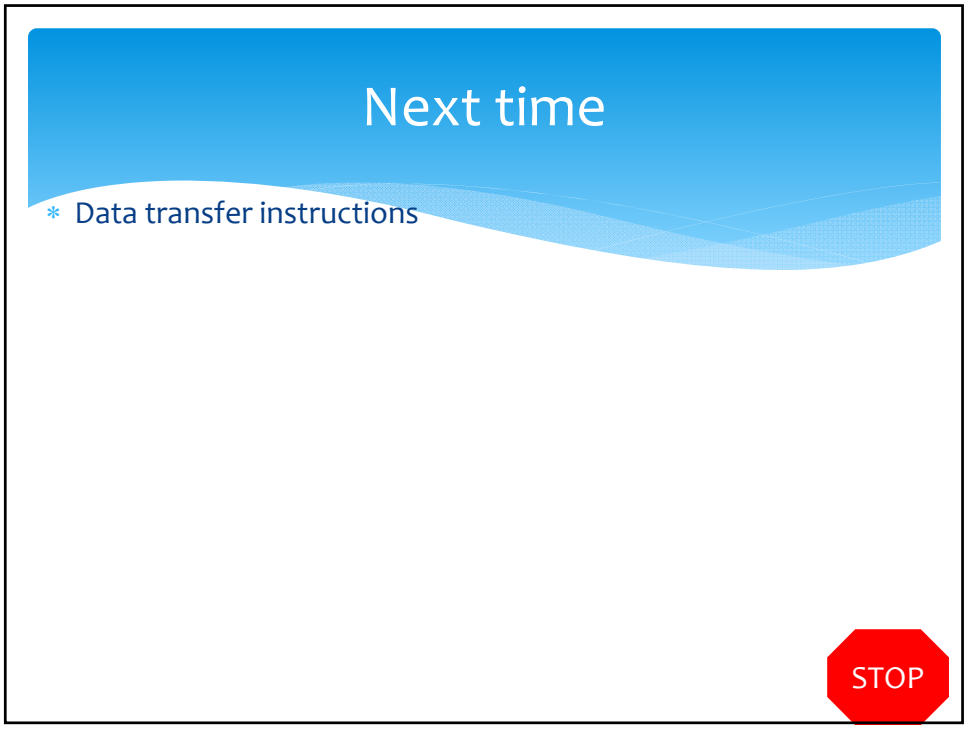
* State before fetch and execution



Based- Indexed Addressing Mode Example(MOV AH,[BX][SI] +1234H)

* State after execution




Next time
Data transfer instructions
STOP