

# ADDRESSING MODES IN 80286 MICROPROCESSOR

## **PRESENTED BY:**

Pratik Dangal

Sudip Dahal

Pratik Bassi

# Contents

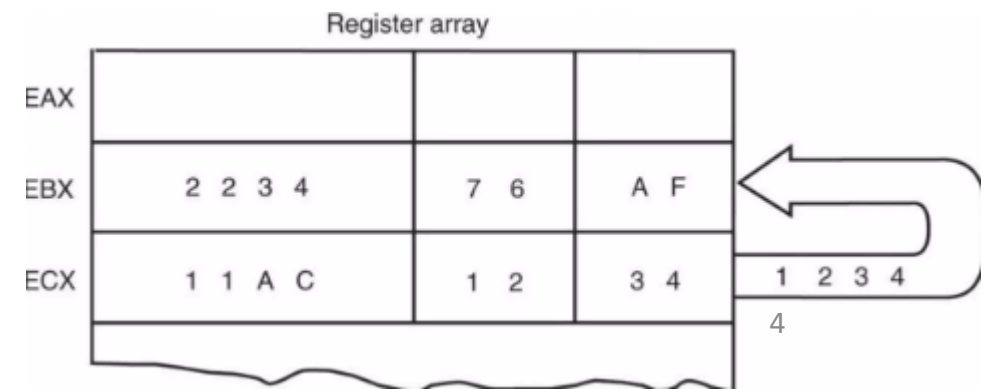
- Introduction
- Different Addressing Modes
  - Register Addressing Mode
  - Immediate Addressing Mode
  - Direct Addressing Mode
  - Register Indirect Addressing Mode
  - Based Addressing Mode
  - Based-Indexed Addressing Mode
  - Based-Indexed Mode with displacement
- References

# Introduction:

- In the context of the 80286 microprocessor, "addressing modes" refer to the various methods by which the CPU can access operands (data) in memory or registers.
- The 80286, being an x86 microprocessor, supports several addressing modes to facilitate flexible and efficient data manipulation. These addressing modes determine how the address of an operand is calculated.
- The 80286 has different addressing modes for instructions to access operands from memory. The different Addressing Modes of 80286 Microprocessor are as follows:

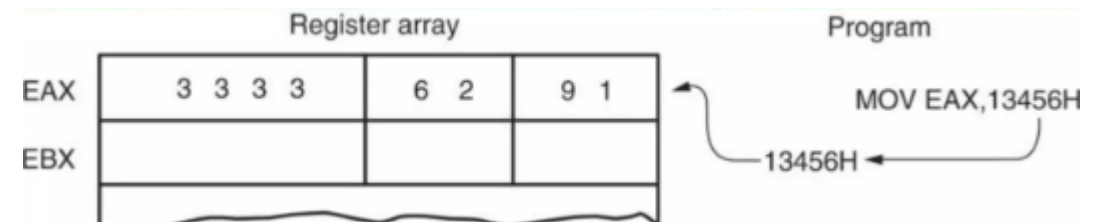
# Register Addressing Mode:

- It is the most common form of data addressing.
- The operand is located in a register, and the instruction specifies which register to use.
- The microprocessor contains these 8 bit register names used with register addressing : AH, AL, BH, BL, CH, CL, DH and DL.
- 16 bit register names : SP, BP, SI and DI.
- EG: MOV BX, CX .where, the effect of executing the MOV BX, CX instruction at the point just before the BX register changes.



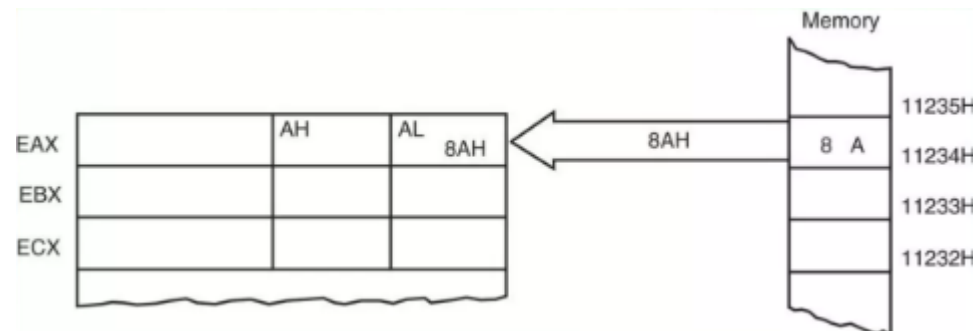
# Immediate Addressing Mode:

- The operand is specified directly in the instruction.
  - immediate data are constant data,
  - data transferred from a register or memory location are variable data.
- Immediate addressing operates upon a byte or word of data
- EG: The operation of the MOV AX,1345H instruction ,This instruction copies the immediate data (1345H) into AX
- As with the MOV instruction the source data overwrites the destination data.



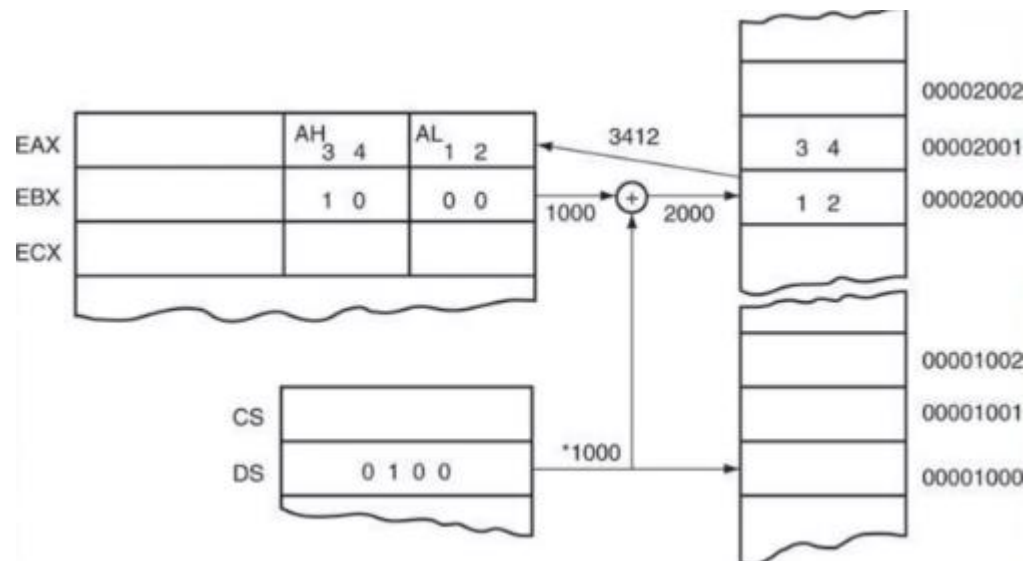
# Direct Addressing mode:

- The instruction specifies the memory address directly.
- Direct addressing with a MOV instruction transfers data between a memory location , located within the data segment, and the AL(8-bit), AX(16-bit) or EAX(32-bit) registers.
- Usually a 3-byte long instruction.
- EG: The operation of the MOV AL,[1234H] (Here ,the content at memory address '1234H' is moved into the AX register). The operation of the MOV AL,[1234H] instruction when DS=1000H



# Register Indirect Addressing mode:

- Allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI and SI.
- In the 64-bit mode, the segment registers serve no purpose in addressing a location
- EG: The operation of the MOV AX,[BX] instruction when BX=10000H and DS=0100H. Note that this instruction is shown after the contents of memory are transferred to AX.



# Based Addressing Mode:

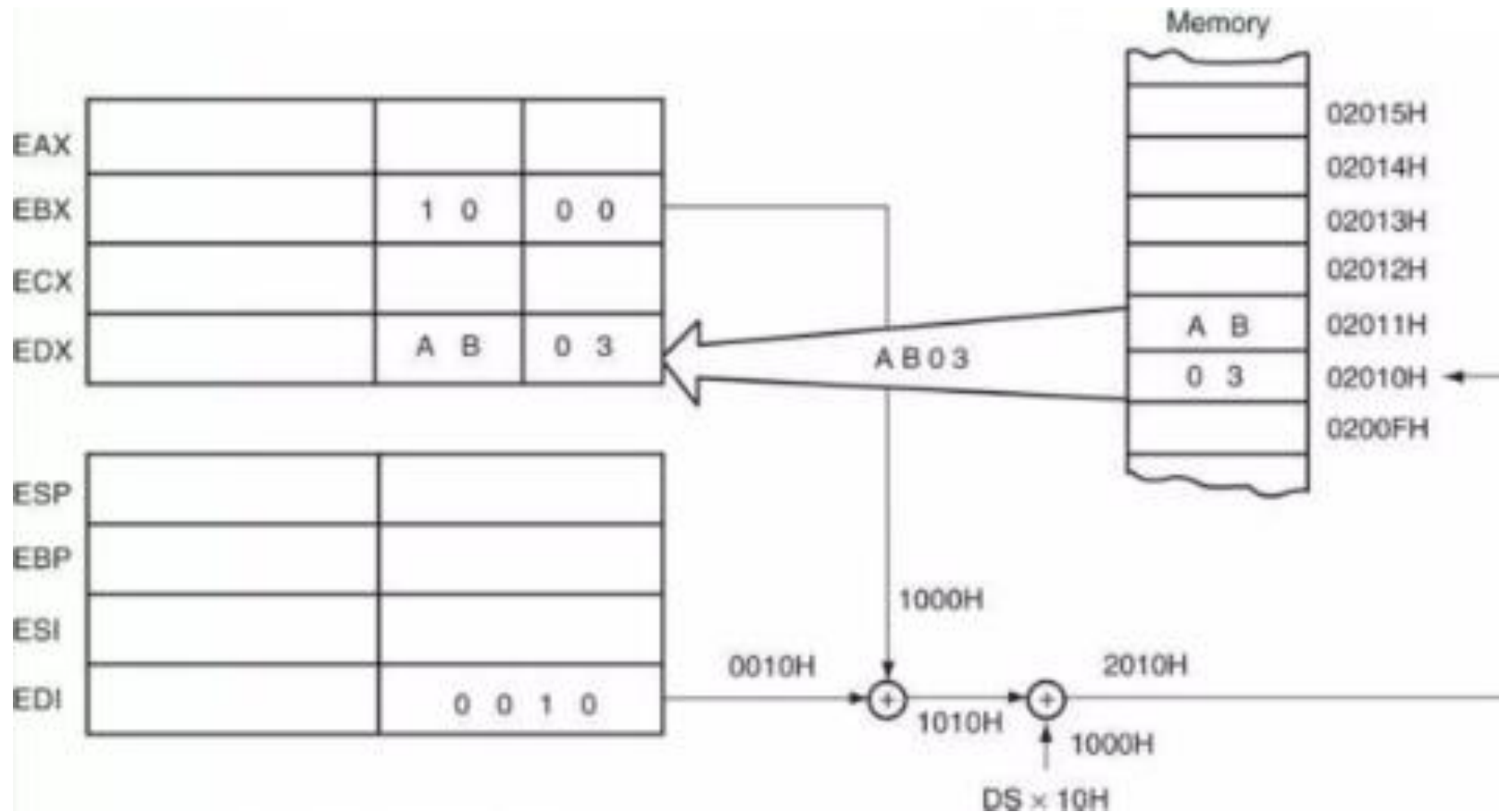
- Uses a base register to hold the address of the operand.
- Example: `MOV AX, [BX+1234h]` (Here, the address is calculated by adding 1234h to the value in the BX register and The instruction fetches the data from the computed memory address and moves it into the AX register.)



# Based-Indexed Addressing Mode:

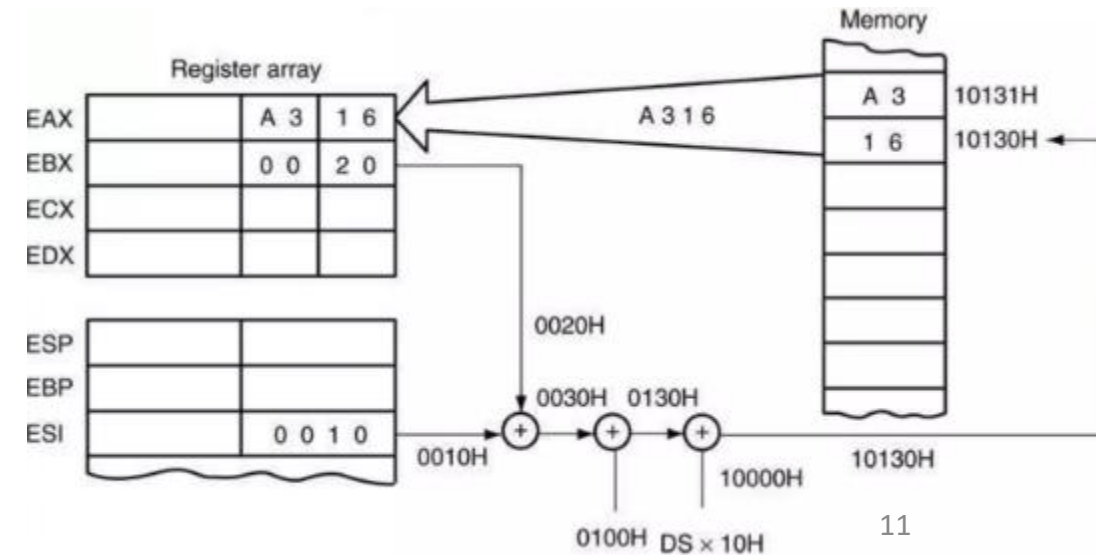
- The effective address is calculated using a base address from a register plus an offset.
- Similar to indirect addressing because it indirectly address memory data.
- The base register holds the beginning location of a memory array.
- The index register holds the relative position of an element in the array.
- Whenever BP addresses memory data, both the stack segment register and BP generate the effective address.
- Example: `MOV AX, [BX+SI]` (Here, the effective address is calculated as the sum of the contents of BX and SI registers.)

Here, An example showing how the base index addressing mode functions for the MOV DX,[BX+DI] instruction. Notice that memory address 02010H is accessed because DS=0100H, BX=1000H and DI=0010H.



# Based-Indexed Mode with displacement:

- Least used addressing mode.
- The operand's offset is the sum of a base register's contents, an index register's contents, and an 8 or 16-bit displacement.
- This addressing mode is too complex for frequent use in programming.
- EG: `MOV AX, [BX+SI+100H]` (Here, the effective address is calculated as the sum of BX, SI, and an additional displacement value 100h.)



# References:

- <https://www.eeeguide.com>
- <https://datasheets.chipdb.org>

**THANK YOU**