BATU University | Fall 2025

MICROPROCESSORS

Lecture 4: Addressing Modes

Presentation By:

Prof. Mohamed E. NASSAR -Prof. Osama EL NAHASS

Prof. Hayman EL SAYED

Lecture Overview

1. Register Addressing
2. Immediate Addressing
3. Direct Data Addressing
4. Register Indirect Addressing
5. Base-Plus-Index Addressing
6. Register Relative Addressing
7. Base Relative-Plus-Index Addressing

Introduction

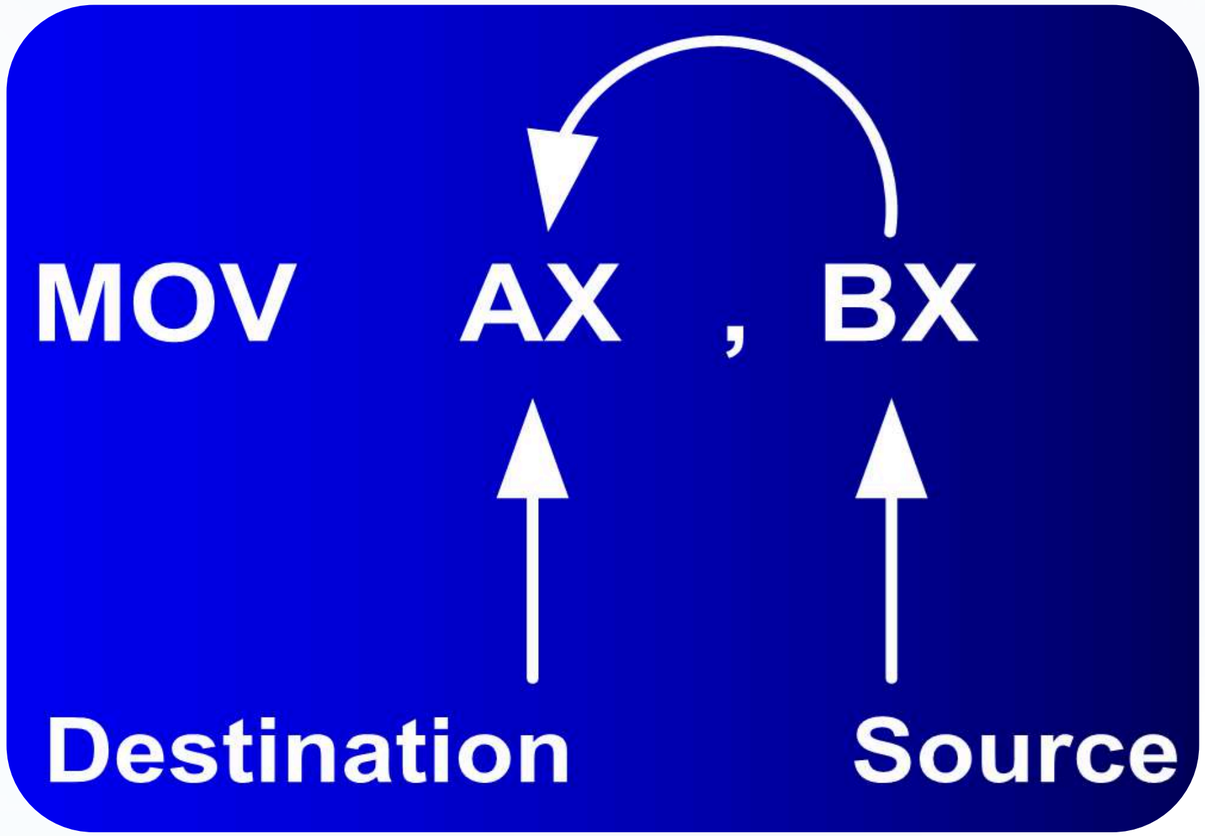
- Efficient software development for the microprocessor requires a complete familiarity with the addressing modes employed by each instruction.
- The MOV (move data) instruction is used to describe the data-addressing modes.
- The MOV instruction transfers bytes or word of data between registers and memory in 8086 ….80286, words or doublewords in 80386 and above.
- In describing the program memory-addressing modes, the CALL, and JMP instruction show how to modify the flow of the program.

Introduction

- The data addressing modes include (register addressing mode, immediate addressing mode, direct addressing mode, register indirect addressing mode, base-plus-index addressing mode, register relative addressing mode, and , base relative -plus-index addressing mode) in 8086 ….80286 microprocessor. The 80386 and above include a scaled-index mode of addressing memory data.

- The program memory-addressing modes includes (program relative memory-addressing, program direct memory-addressing, program indirect memory-addressing).

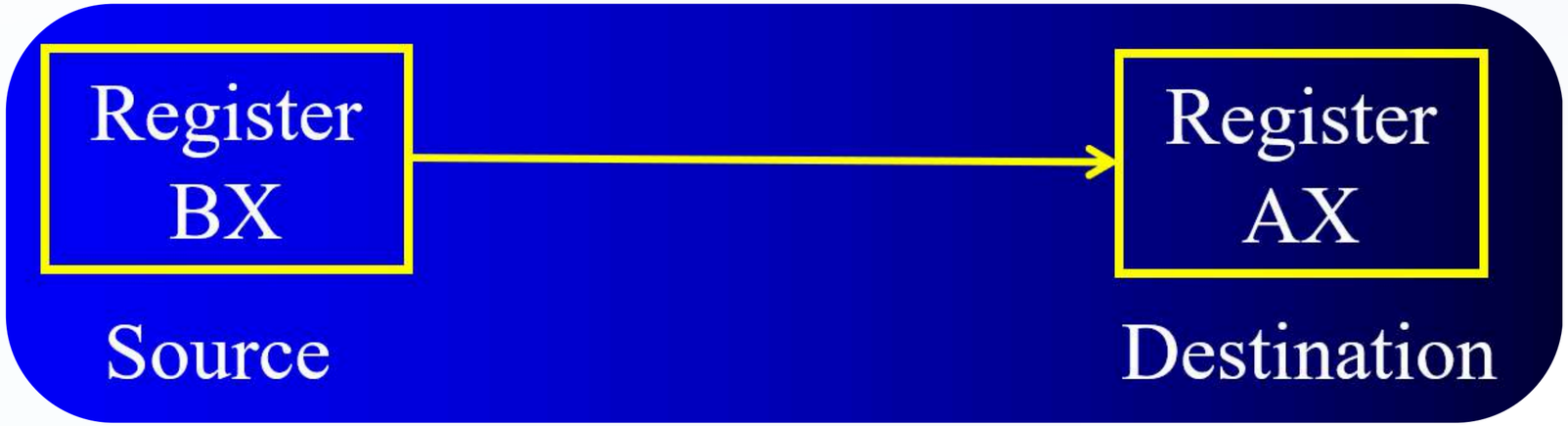
The MOV instruction



Data-Addressing Modes

Register addressing mode	Transfers a copy of a byte or word from the source register or memory location to the destination register or memory location.

MOV **AX, BX**



Register Addressing

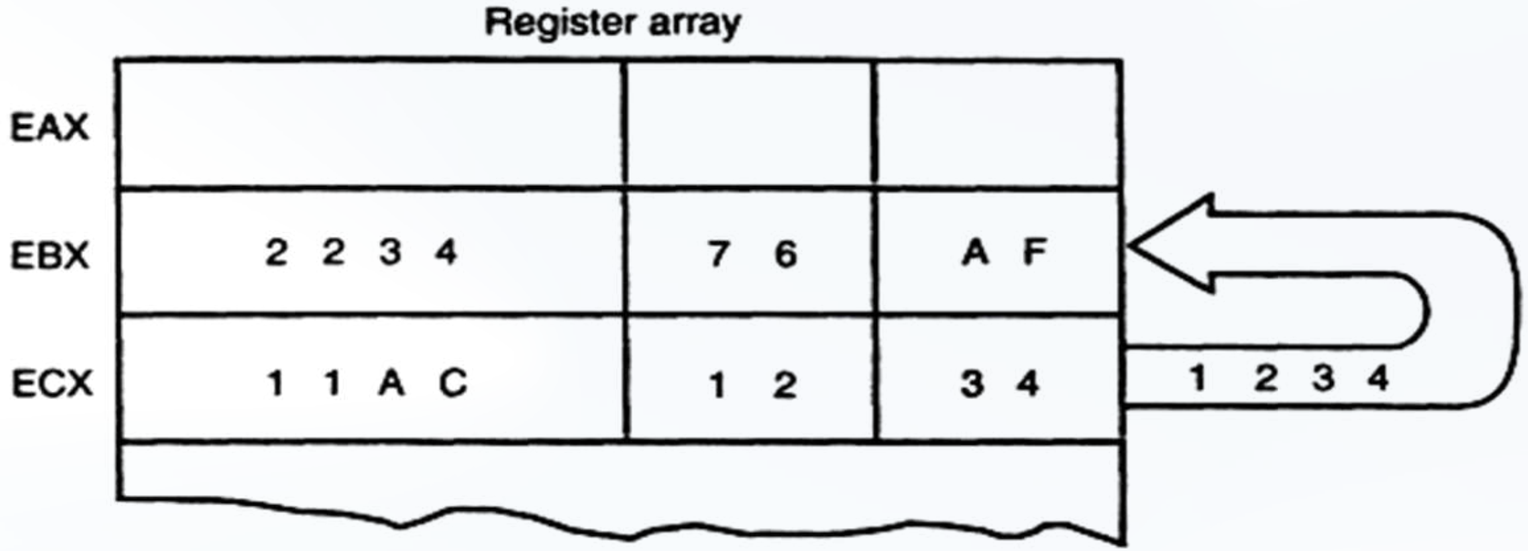
- **Register addressing is the most common form of data addressing and, once the register names are learned, it is the easiest to apply.**

- **Examples of the register-addressed instructions.**

Assembly Language	*Size*	*Operation*
MOV AL, BL	8-bits	Copies BL into AL
MOV CH, CL	8-bits	Copies CL into CH
MOV AX, CX	16-bits	Copies CX into AX
MOV SP, BP	16-bits	Copies BP into SP
MOV BX, ES	16-bits	Copies ES into BX
MOV ECX, EBX	32-bits	Copies EBX into ECX
MOV ESP, EDX	32-bits	Copies EDX into ESP
MOV ES, DS	—	Not allowed (segment-to-segment)
MOV BL, DX	—	Not allowed (mixed sizes)

Register Addressing Example

MOV BX, CX

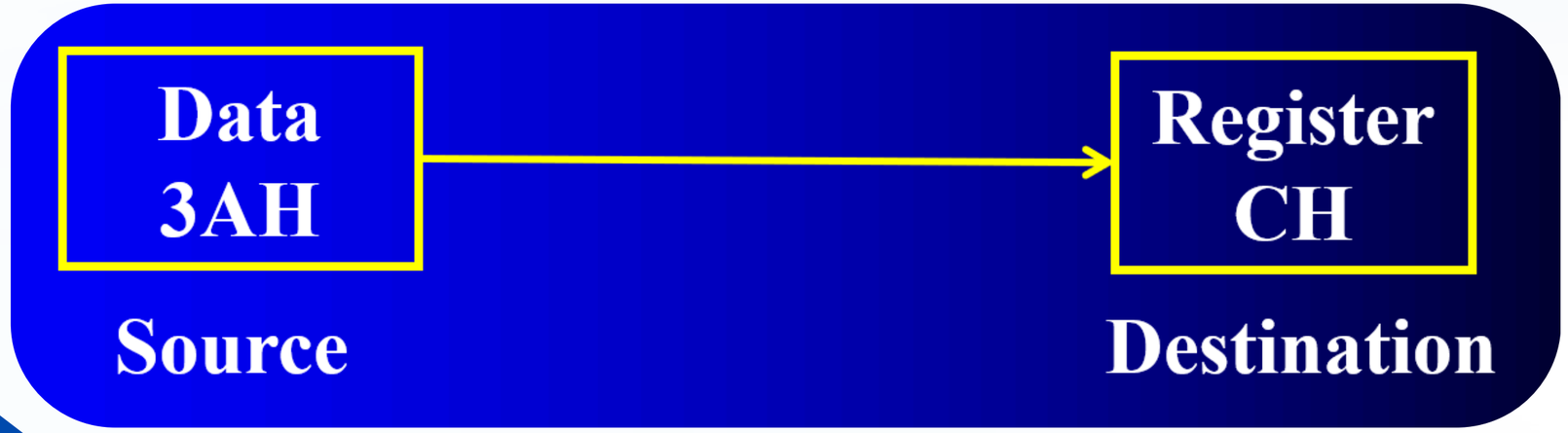


The effect of executing the MOV BX, CX instruction at the point just before the BX register changes. Note that only the rightmost 16 bits of register EBX change.

Data-Addressing Modes

Immediate addressing	Transfers the source-immediate byte or word of data into the destination register or memory location.

MOV CH, 3AH



Immediate Addressing

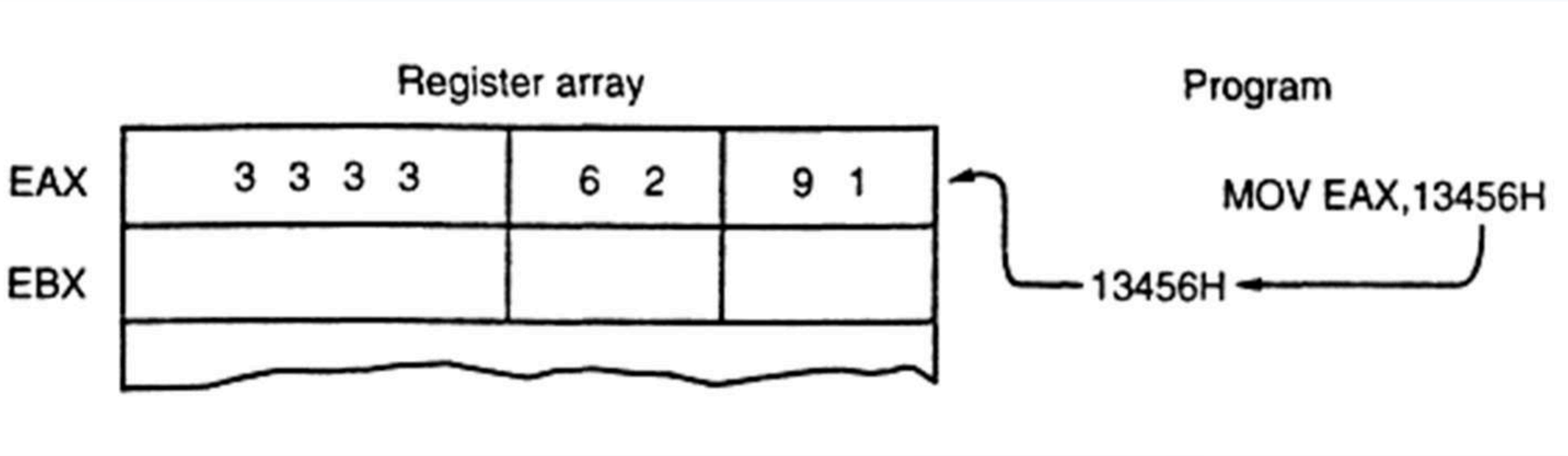
- The term immediate implies that the data immediately follow the hexadecimal code in the memory. Also note that immediate data are constant data, while the data transferred from a register are variable data.

- **Examples of the immediate-addressed instructions.**

Assembly Language	*Size*	*Operation*
MOV BL, 44	8-bits	Copies a 44 decimal (2CH) into BL
MOV AX, 44H	16-bits	Copies a 0044H into AX
MOV CH, 100	8-bits	Copies a 100 decimal (64H) into CH
MOV AL, 'A'	8-bits	Copies an ASCII A into AL
MOV AX, 'AB'	16-bits	Copies an ASCII BA into AX
MOV CL, 11001110B	8-bits	Copies a 11001110 binary into CL
MOV EBX, 12340000H	32-bits	Copies a 12340000H into EBX
MOV ESI, 12	32-bits	Copies a 12 decimal into ESI
MOV EAX, 100B	32-bits	Copies a 100 binary into EAX

Immediate Addressing Example:

MOV EAX, 13456H



The operation of the MOV EAX,13456H instruction. This instruction copies the immediate data (13456H) into EAX.

Data-Addressing Modes

Direct addressing	Moves a byte or word between a memory location and a register. The instruction set does not support a memory-to-memory transfer, except for the MOVS instruction.

MOV [1234H], AX



Direct Data Addressing

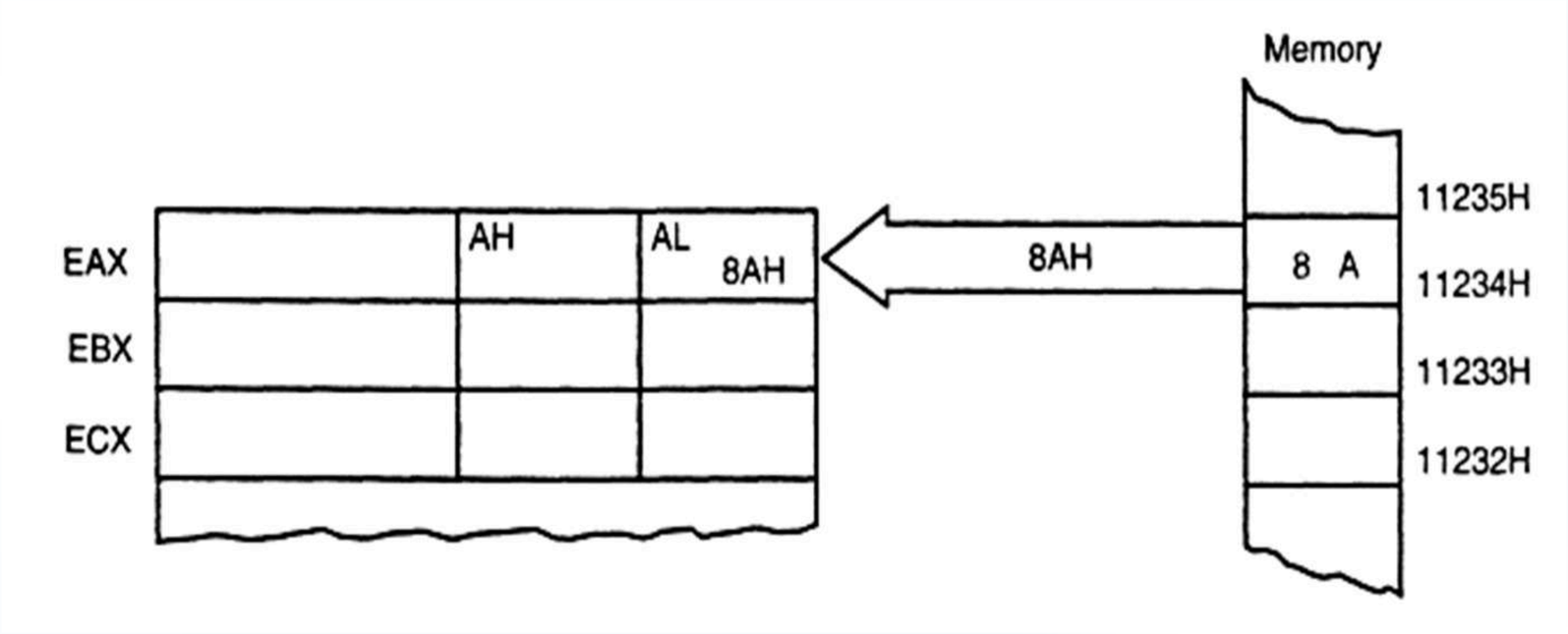
- Direct addressing with a MOV instruction transfers data between a memory location, located within the data segment, and the AL (8-bit), AX (16-bit), or EAX (32-bit) register.

- **Examples of the Direct addressed instructions.**

Assembly Language	*Size*	*Operation*
MOV CH, DOG	8-bits	Copies the byte contents of data segment memory location DOG into CH
MOV CH, [1000H]	8-bits	Copies the byte contents of data segment offset address 1000HintoCH
MOV ES, DATA6	16-bits	Copies the word contents of data segment memory location DATA6 into ES
MOV DATA7, BP	16-bits	Copies BP into data segment memory location DATA7
MOV NUMBER, SP	16-bits	Copies SP into data segment memory location NUMBER
MOVDATA1, EAX	32-bits	Copies EAX into data segment memory location DATA1
MOV EDI, SUM1	32-bits	Copies the doubleword contents of data segment memory location SUM1 into EDI

Direct Addressing

Example : MOV AL, [1234H]

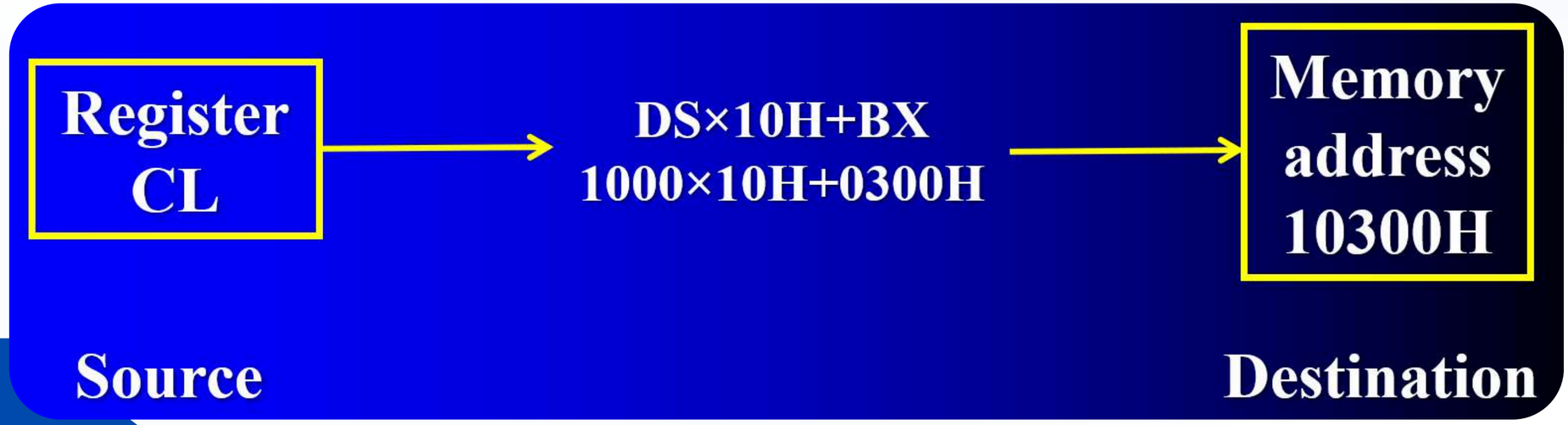


The operation of the MOV AL,[1234H] when DS = 1000H.

Data-Addressing Modes

Register indirect addressing	Transfers a byte or word between a register and a memory location addressed by an index or base register. The index and base registers are BP, BX, DI, and SI.

MOV [BX], CL



Register Indirect Addressing

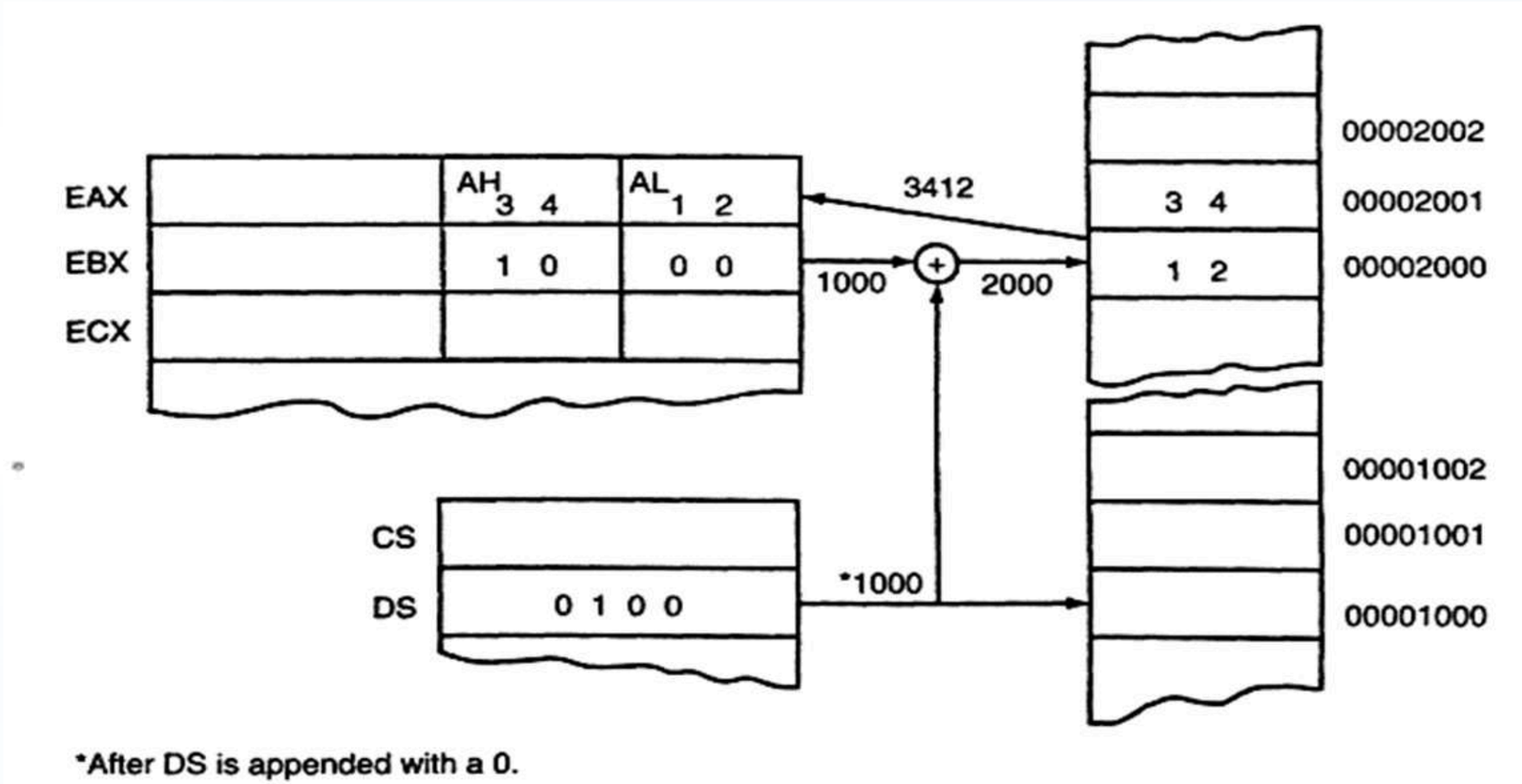
- Register indirect addressing allows data to be addressed at any memory location through an offset address held in any of the following registers: BP, BX, DI, and SI.

- **Examples of the Register indirect addressed instructions.**

Assembly Language	*Size*	*Operation*
MOV CX, [BX]	16-bits	Copies the word contents of the data segment memory location address by BX into CX
MOV [BP], DL	8-bits	Copies DL into the stack segment memory location addressed by BP
MOV [DI], BH	8-bits	Copies BH into the data segment memory location addressed by DI
MOV [DI], [BX]	—	Memory-to-memory moves are not allowed except with string instructions
MOV AL, [EDX]	8-bits	Copies the byte contents of the data segment memory location addressed by EDX into AL
MOV ECX, [EBX]	32-bits	Copies the doubleword contents of the data segment memory location addressed by EBX into ECX

Register indirect Addressing

Example : MOV AX, [BX]



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.

Data-Addressing Modes

Base-plus-index addressing	Transfers a byte or word between a register and the memory location addressed by a base register (BP or BX) plus an index register (DI or SI).

MOV [BX+SI], BP



Base-Plus-Index Addressing

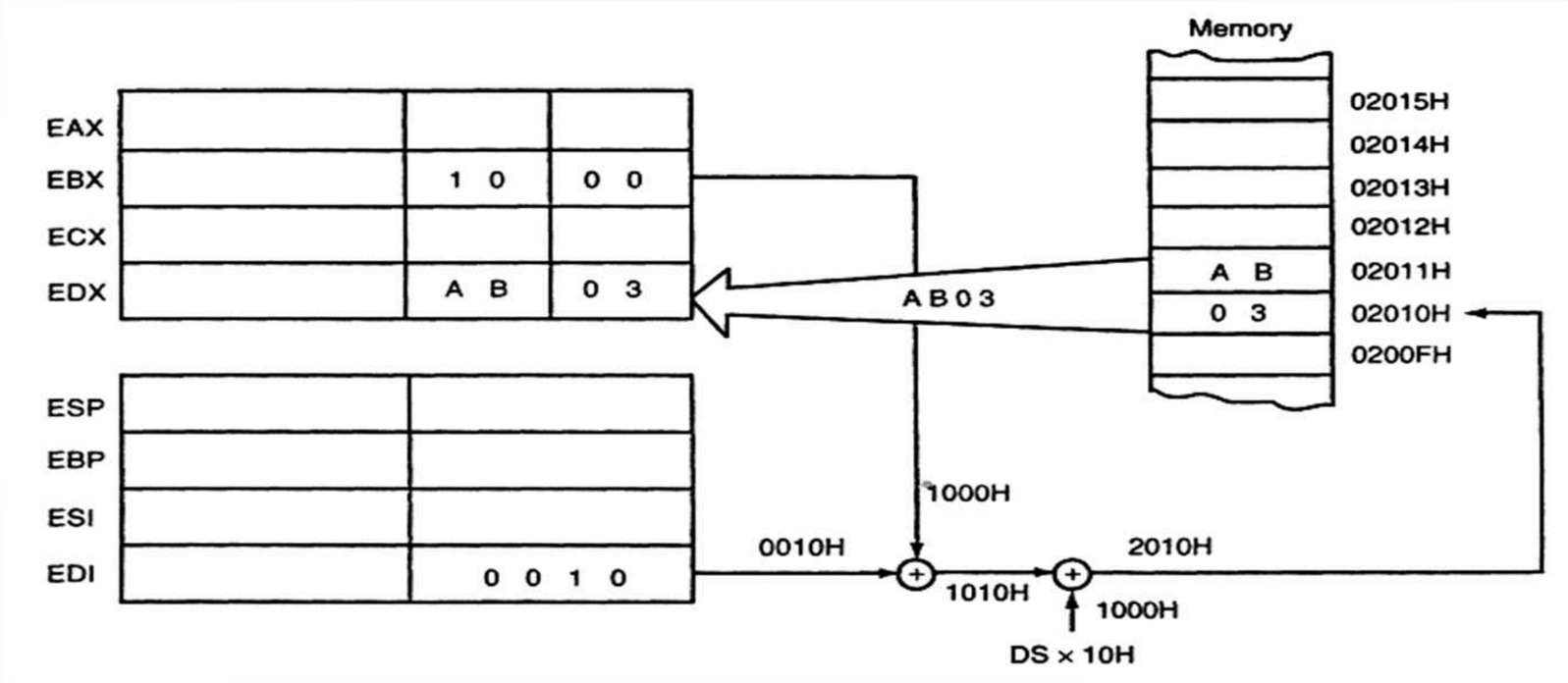
- Base-plus-index addressing is similar to indirect addressing because it indirectly addresses memory data.
- In the 8086 through the 80286, this type of addressing uses one base register (BP or BX), and one index register (DI or SI) to indirectly address memory.

- **Examples of the Base-Plus-Index addressed instructions.**

Assembly Language	*Size*	*Operation*
MOV CX,[BX+DI]	16-bits	Copies the word contents of the data segment memory location address by BX plus DI into CX
MOV CH,[BP+SI]	8-bits	Copies the byte contents of the stack segment memory location addressed by BP plus SI into CH
MOV [BX+SI],SP	16-bits	Copies SP into the data segment memory location addresses by BX plus SI
MOV [BP+ DI],AH	8-bits	Copies AH into the stack segment memory location addressed by BP plus DI

Base-Plus-Index Addressing

Example : MOV DX, [BX + DI]

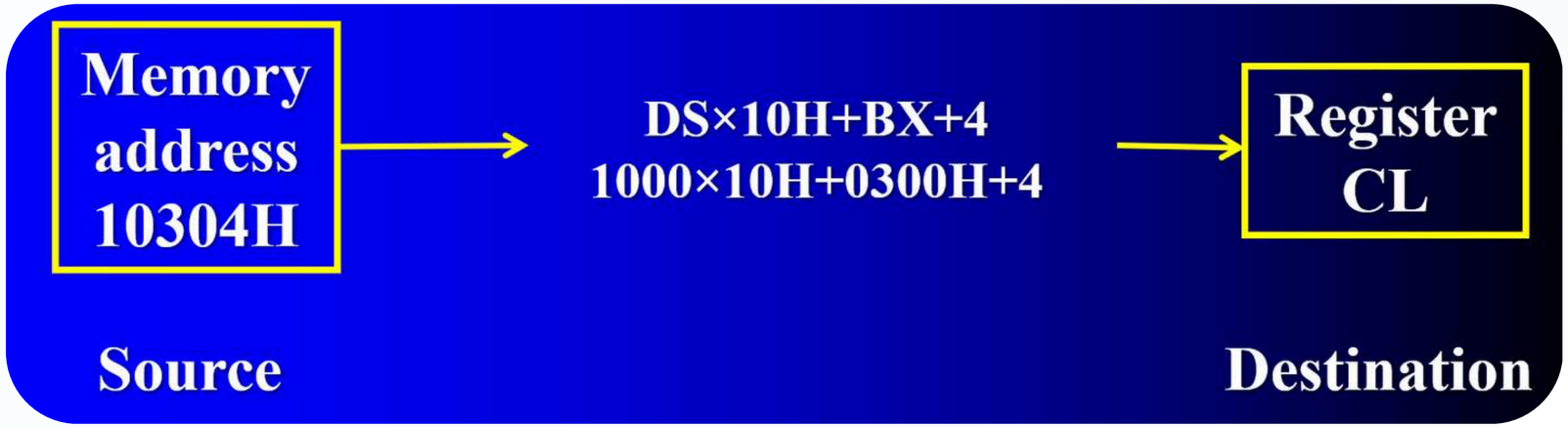


An example showing how the base-plus-index addressing mode functions for the MOV DX,[BX+DI] instruction. Notice that memory address 02010H is accessed because DS = 0100H, BX = 1000H, and DI = 0010H.

Data-Addressing Modes

Register relative addressing	Moves a byte or word between a register and the memory location addressed by an index or base register plus a displacement.

MOV CL,[BX+4]



Register Relative Addressing

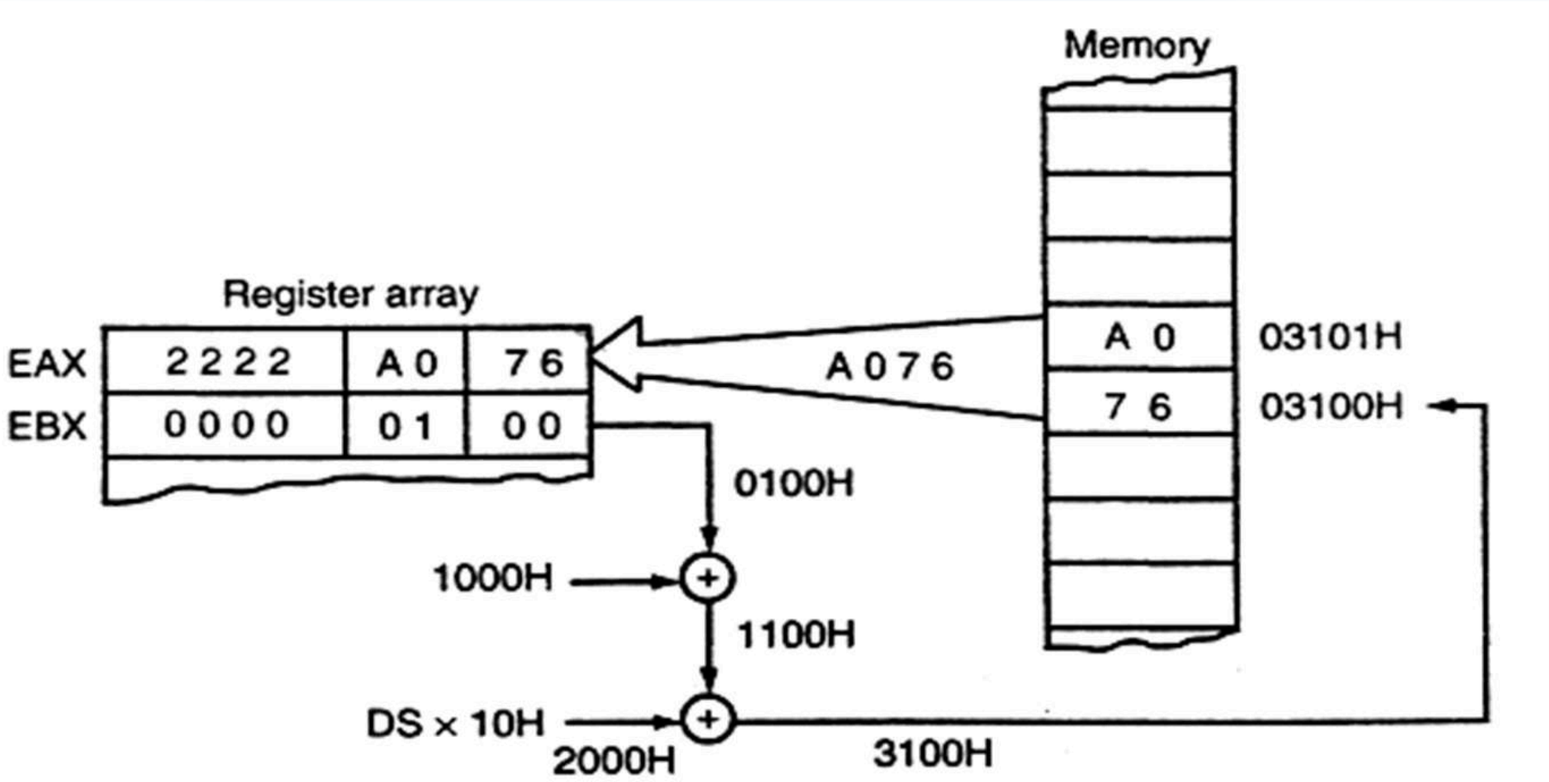
•Register relative addressing is similar to base-plus-index addressing and displacement addressing. In register relative addressing, the data in a segment of memory are addressed by adding the displacement to the contents of a base or an index register (BP, BX, Dl, or SI).

- **Examples of the register Relative addressed instructions.**

Assembly Language	Size	Operation
MOV AX, [DI+100H]	16-bits	Copies the word contents of the data segment memory location addressed by DI plus 100H into AX
MOV ARRAY[SI], BL	8-bits	Copies BL into the data segment memory location addressed by ARRAY plus SI
MOV LIST[SI+2], CL	8-bits	Copies CL into the data segment memory location addressed by sum of LIST, SI, and 2
MOV DI,SET_IT[BX]	16-bits	Copies the word contents of the data segment memory location addressed by the sum of SET_IT and BX into DI
MOV ARRAY[EBX],EAX	32-bits	Moves EAX into the data segment memory location addressed by the sum of ARRAY and EBX

Register Relative Addressing

MOV AX, [BX + 1000H]



The operation of the MOV AX,[BX+1000H] instruction, when BX = 0100H and DS = 0200H.

Data-Addressing Modes

Base relative-plus-index addressing	**Transfers a byte or word between a register and the memory location addressed by a base and an index register plus a displacement.**

MOV ARRAY[BX+SI],DX

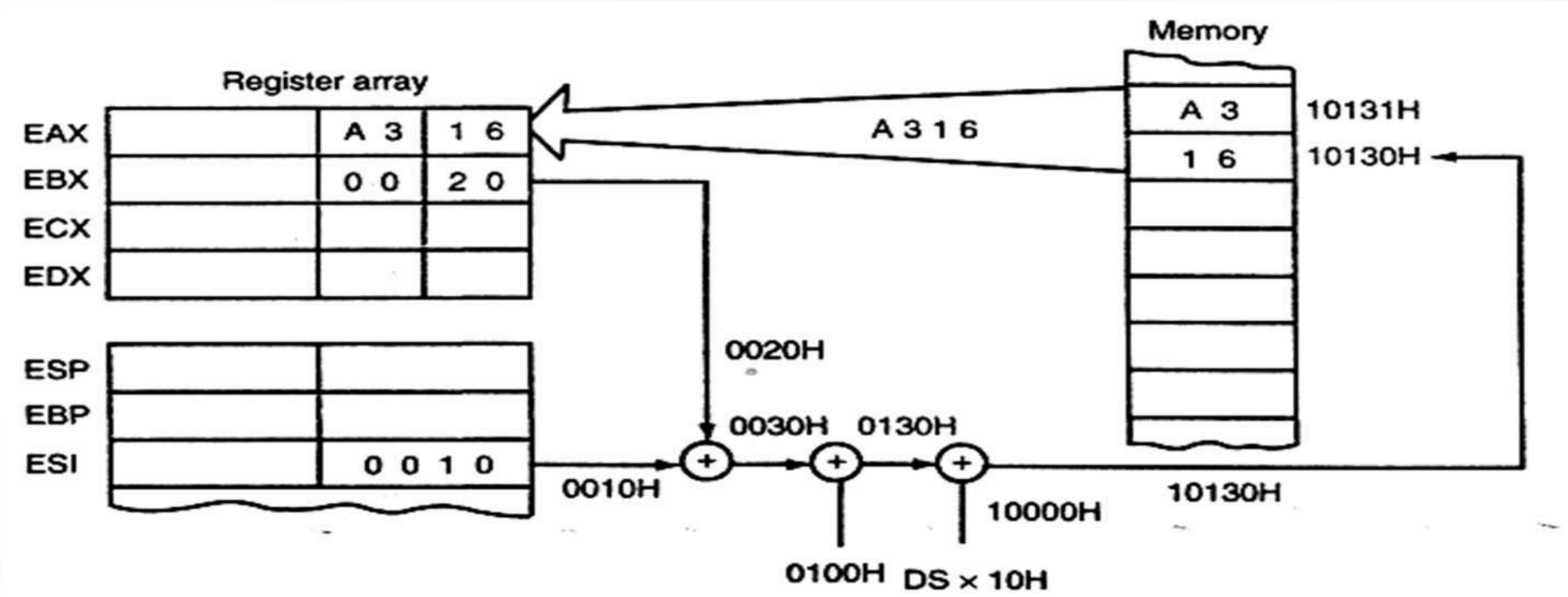


Register Relative Addressing

Assembly Language	*Size*	*Operation*
MOV DH, [BX+DI+20H]	8-bits	Copies the byte contents of the data segment memory location addressed by the sum of BX, Dl, and 20H into DH.
MOV AX, FILE[BX+DI]	16-bits	Copies the word contents of the data segment memory location addressed by the sum of FILE, BX, and Dl into AX.
MOV LIST[BP+DI], CL	8-bits	Copies CL into the stack segment memory location addressed by the sum of LIST, BP, and Dl

Base Relative-Plus-Index Addressing

MOV AX, [BX+SI+100H]



- An example of base relative-plus-index addressing using a MOV AX, [BX+SI+100H] instruction.
- Note: DS = 1000H.

Data-Addressing Modes

Scaled-index addressing	**Is available only in the 80386 through the Pentium 4 microprocessor. The second register of a pair of registers is modified by the scale factor of 2X, 4X, or 8X to generate the operand memory address.**

MOV [EBX+2×ESI],AX

