جامعة المستقبل
الكلية التقنية الهندسية
قسم تقنيات الهندسة الكهربائية

Third Stage

Microprocessor

Addressing Modes

Presented By:

Dr. Mohammed Fadhil

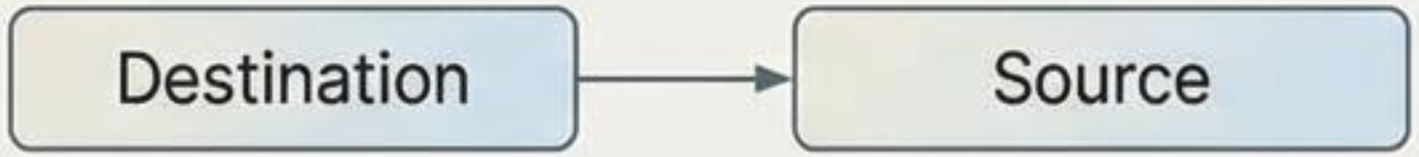
Email: mohammed.fadhil1@uomus.edu.iq

8086 Addressing Modes: Operands and Data Types

Operands hold the data to be processed, defined by register type and memory addressing.

Operand Structure & Roles

- First operand: **Destination** (Data is stored here).
- Second operand: **Source** (Data is read from here).



Register and Memory Types

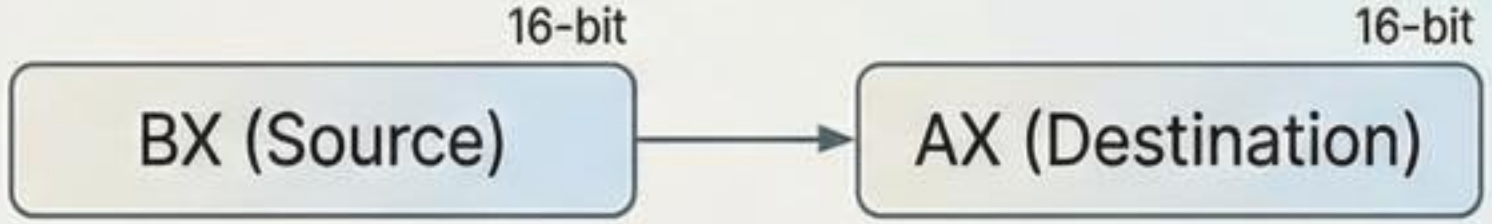
16-bit Registers (e.g., AX, BX, CX, DX)

8-bit Registers (e.g., AL, BL, AH, BH)

Memory (Uses offset addresses)

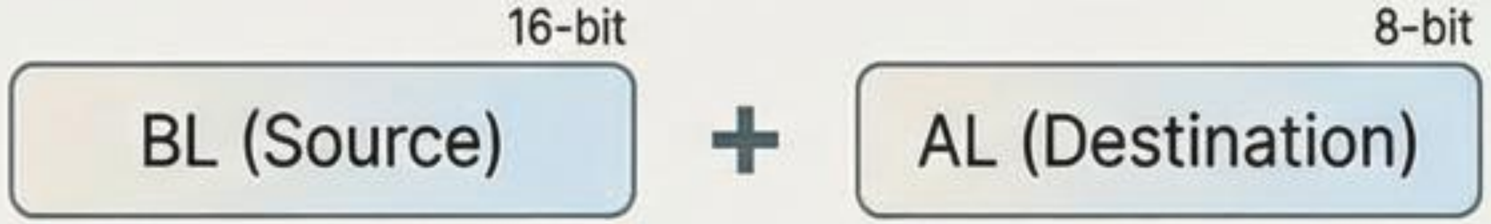
Detailed Examples

Example 1: MOV AX, BX



Copies contents of BX to AX

Example 2: ADD AL, BL

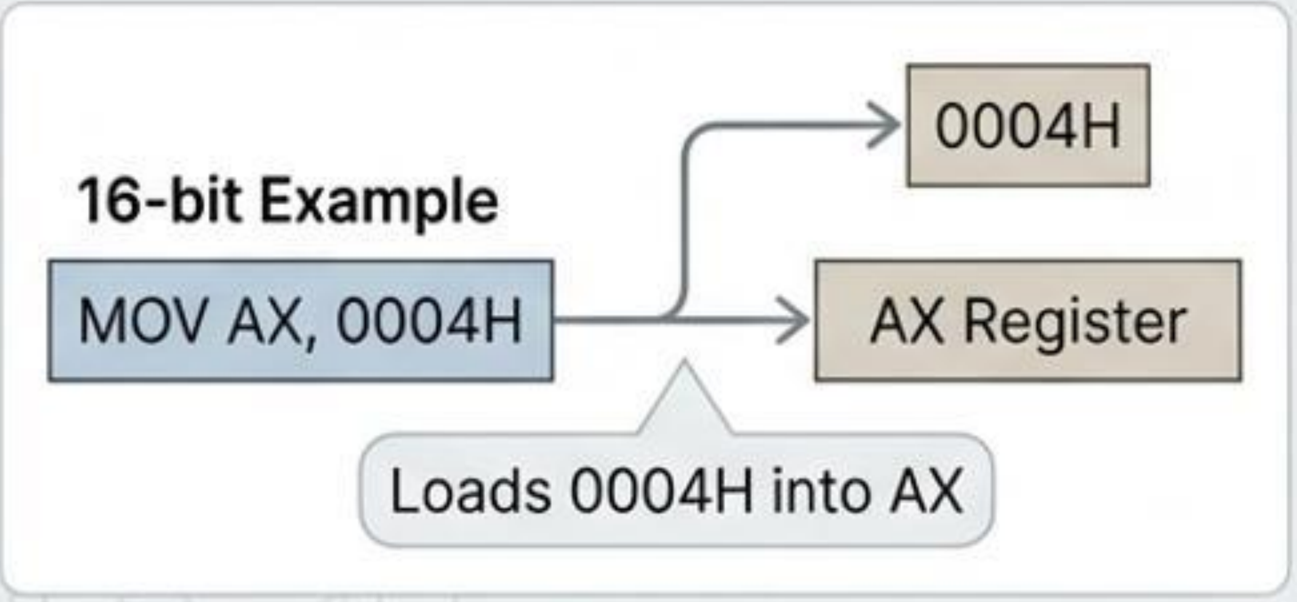


Adds contents of BL to AL

Immediate Addressing Mode

The operand value is encoded **directly** within the instruction.

Examples & Operation



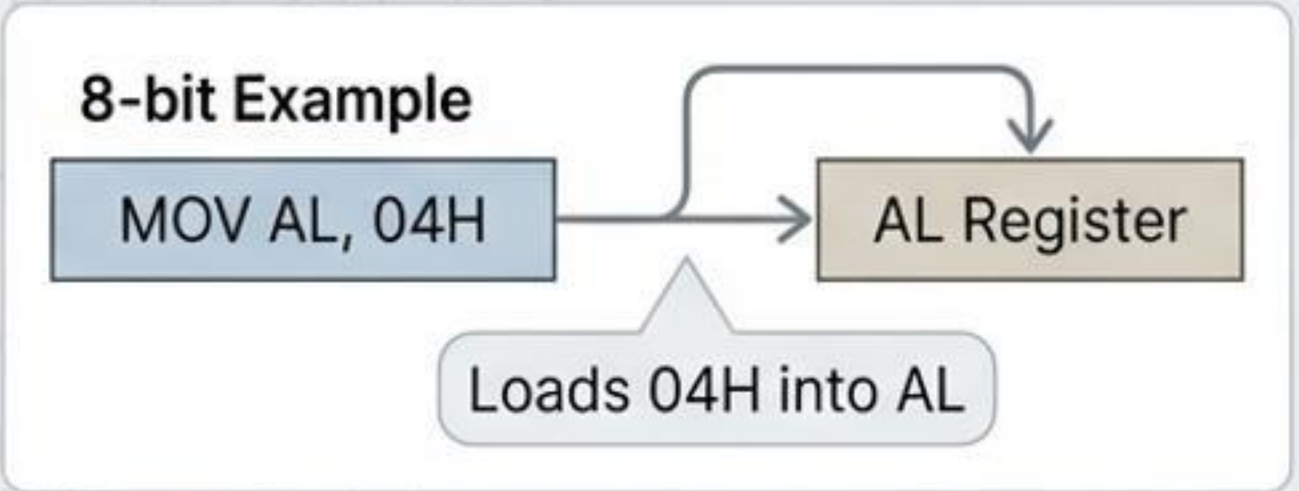
16-bit Example

MOV AX, 0004H

0004H

AX Register

Loads 0004H into AX



8-bit Example

MOV AL, 04H

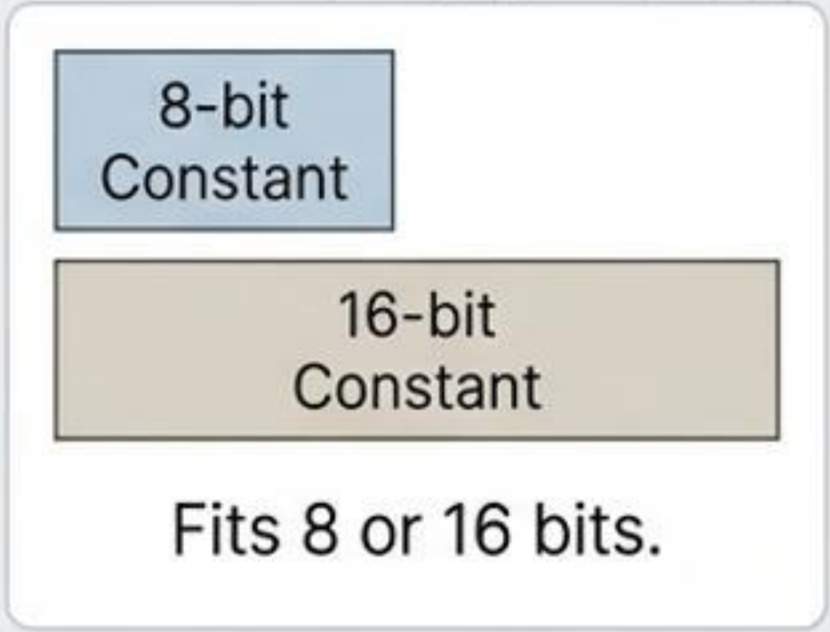
AL Register

Loads 04H into AL

Key Characteristics

- No memory fetch required.
- Execution is fast due to direct access.
- Constant data must be known at assembly time.

Data Constraints



8-bit Constant

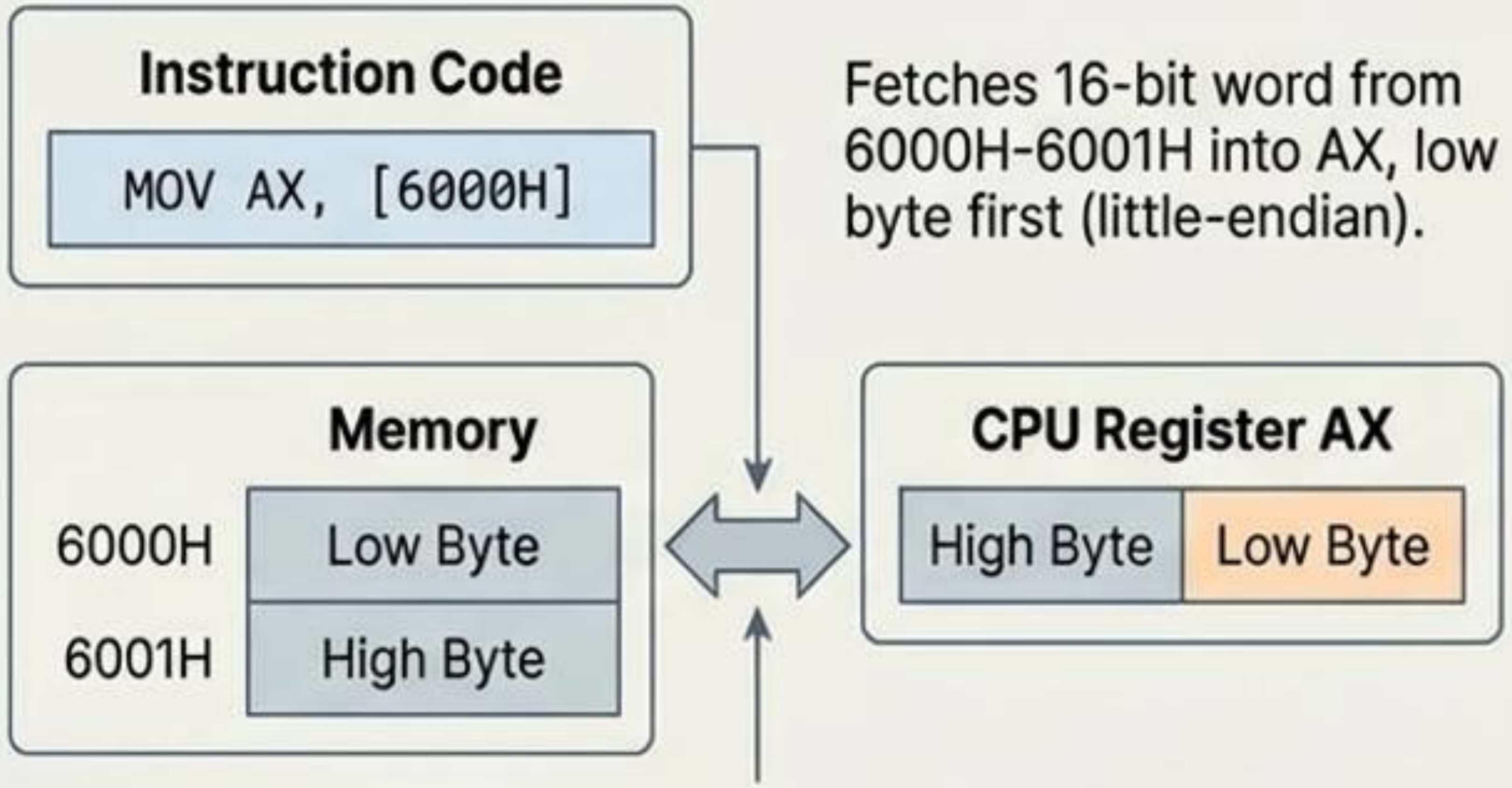
16-bit Constant

Fits 8 or 16 bits.

Direct Addressing Mode

Instruction specifies exact 16-bit memory or I/O offset. Effective for fixed locations.

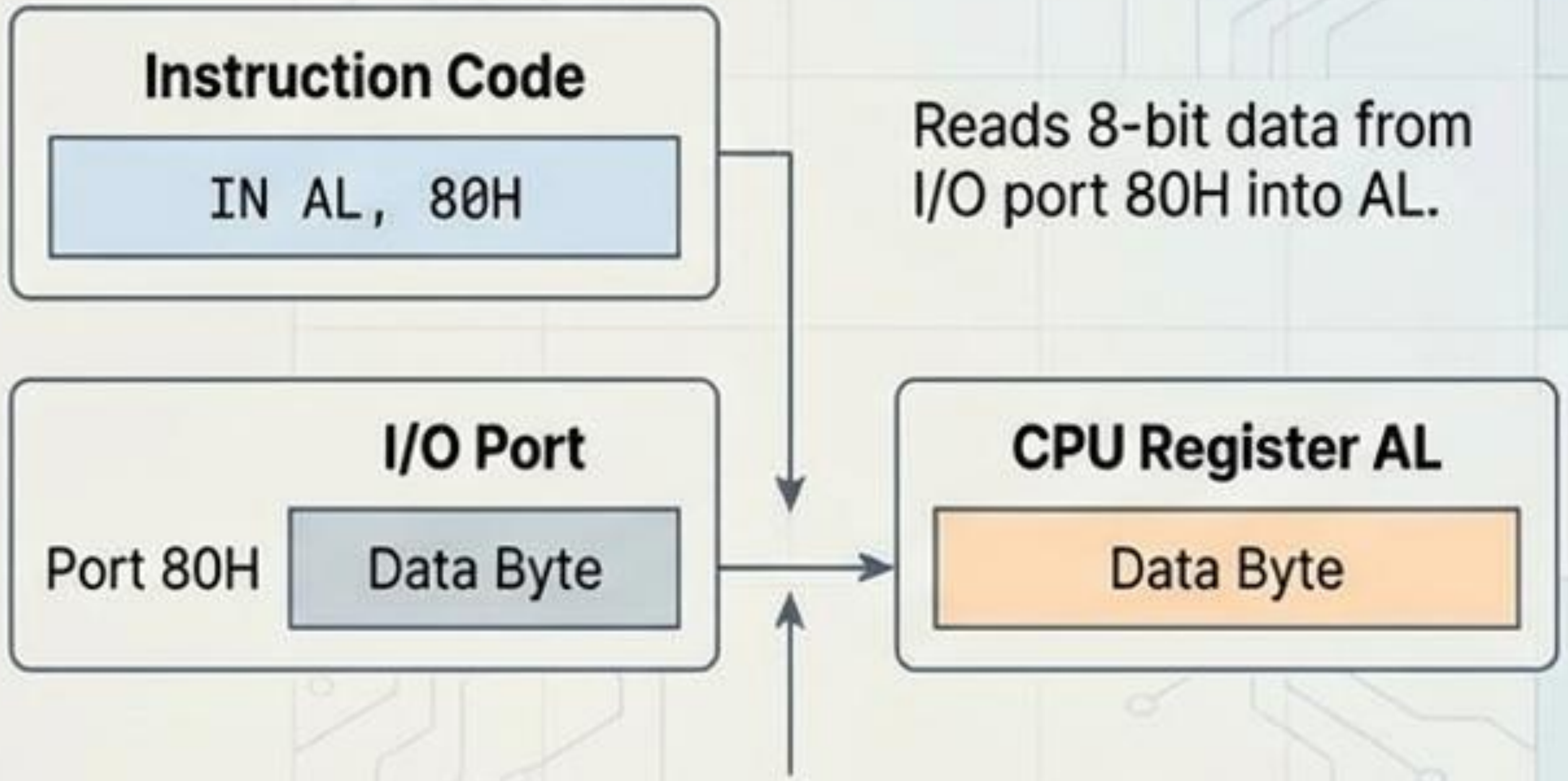
Memory Access Example: MOV AX, [6000H]



Fetches 16-bit word from 6000H-6001H into AX, low byte first (little-endian).

Fetches 16-bit word from 6000H-6001H into AX, low byte first (little-endian).

I/O Port Access Example: IN AL, 80H



Reads 8-bit data from I/O port 80H into AL.

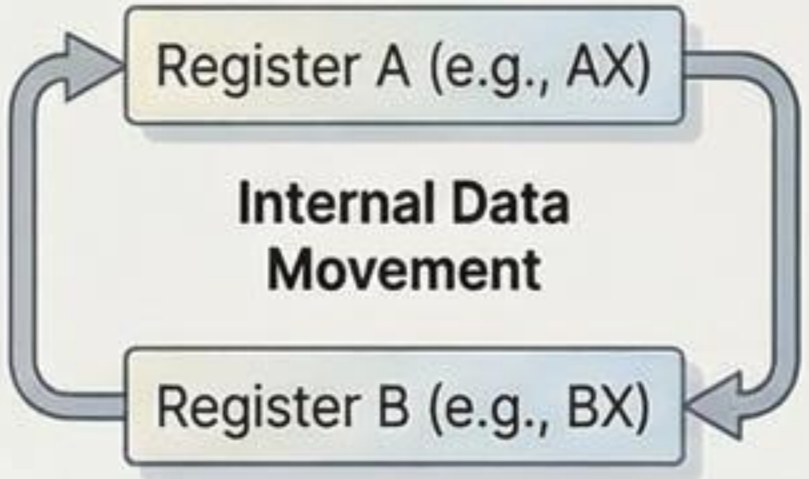
Reads 8-bit data from I/O port 80H into AL.

Analysis: Best for fixed memory/I/O; less flexible for variable data blocks.

Register Addressing Mode

Direct Data Transfer Between CPU Registers.

Core Concept



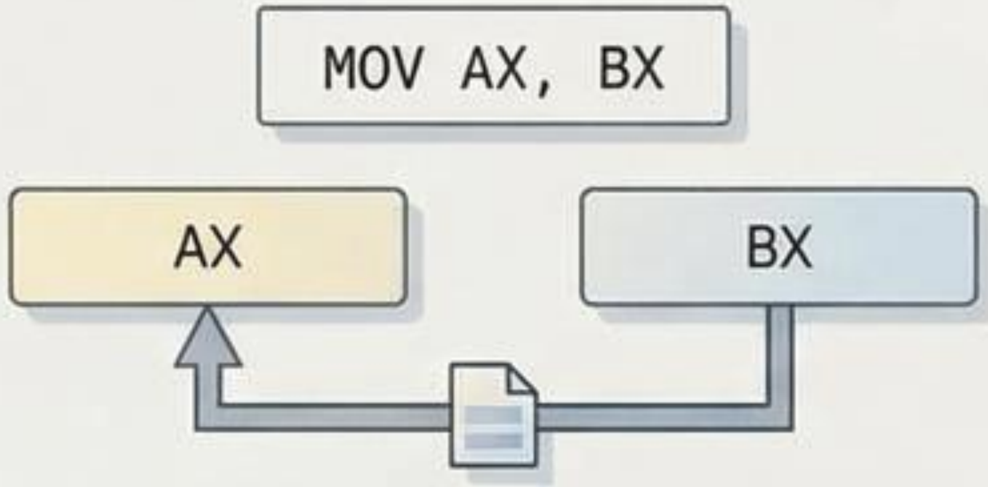
Register A (e.g., AX)

Internal Data Movement

Register B (e.g., BX)

- Both operands are registers
- No bus cycle occurs
- Data stays within CPU
- Maximal speed & compact code

16-Bit Example: MOV AX, BX



```
MOV AX, BX
```

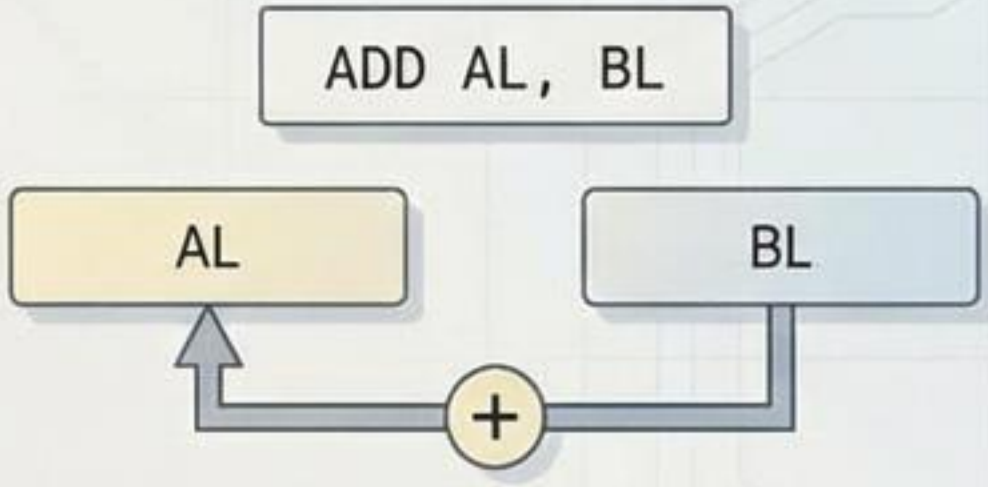
AX

BX

Copies content of BX register to AX. Both are 16-bit.

`B8 XX YY` (illustrative)

8-Bit Example: ADD AL, BL



```
ADD AL, BL
```

AL

BL

Adds content of BL to AL, stores result in AL. Both are 8-bit.

`02 C3` (illustrative)

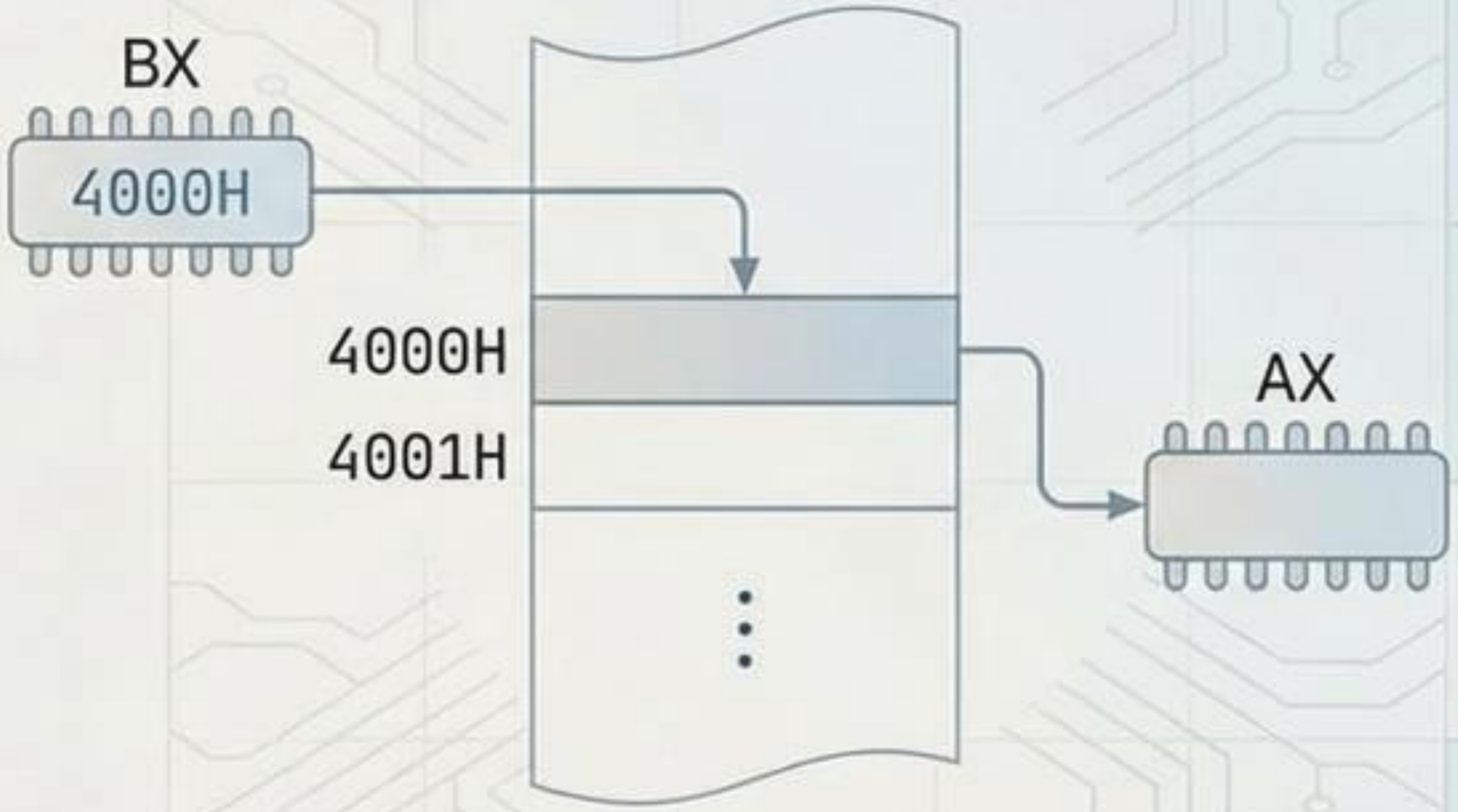
Key Considerations

- Register choice follows data size (AX for 16-bit, AL for 8-bit).
- Fastest mode due to no external memory access.

Register Indirect Addressing

Concept & Syntax

- A register inside brackets supplies the memory offset.
- Square brackets distinguish memory from register addressing.
- Only BX, BP, SI, DI are valid for 16-bit offset.



Example:

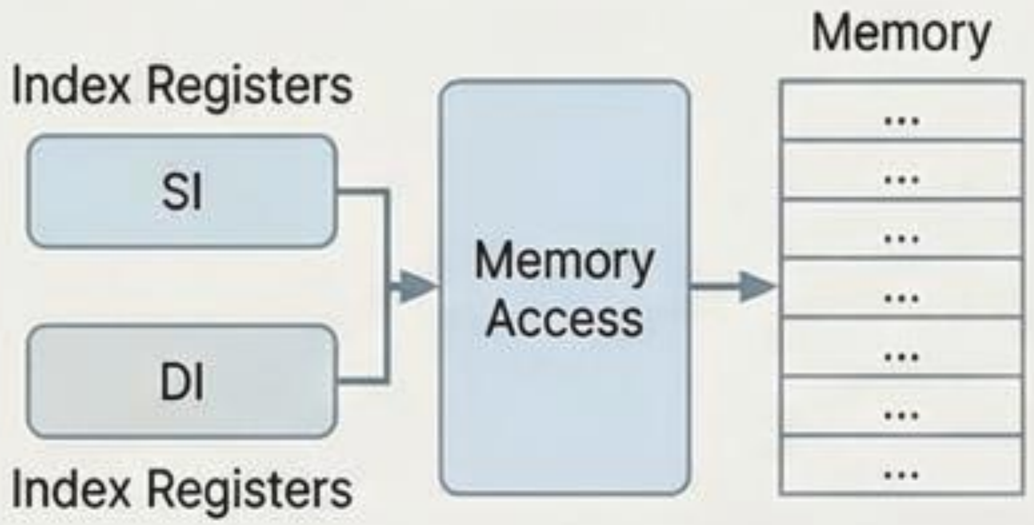
```
MOV AX, [BX]
```

With BX=4000H, fetches the word at 4000H-4001H into AX.

Indexed Addressing Mode

Using Index Registers for Memory Access and Traversal.

Concept



Index Registers

SI

DI

Index Registers

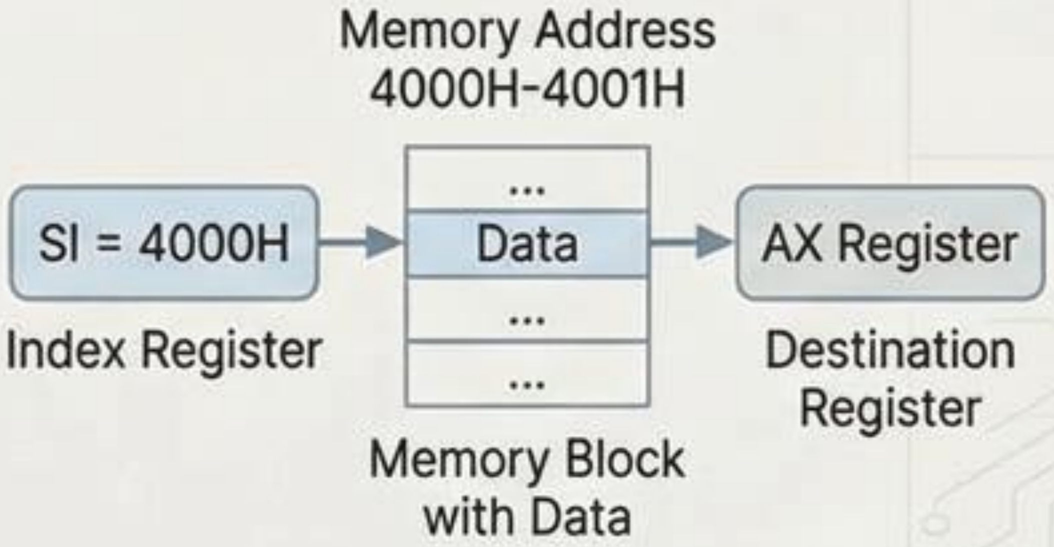
Memory Access

Memory

...

SI or DI holds the offset.

Example: Loading



Memory Address 4000H-4001H

SI = 4000H

Index Register

...

Data

...

...

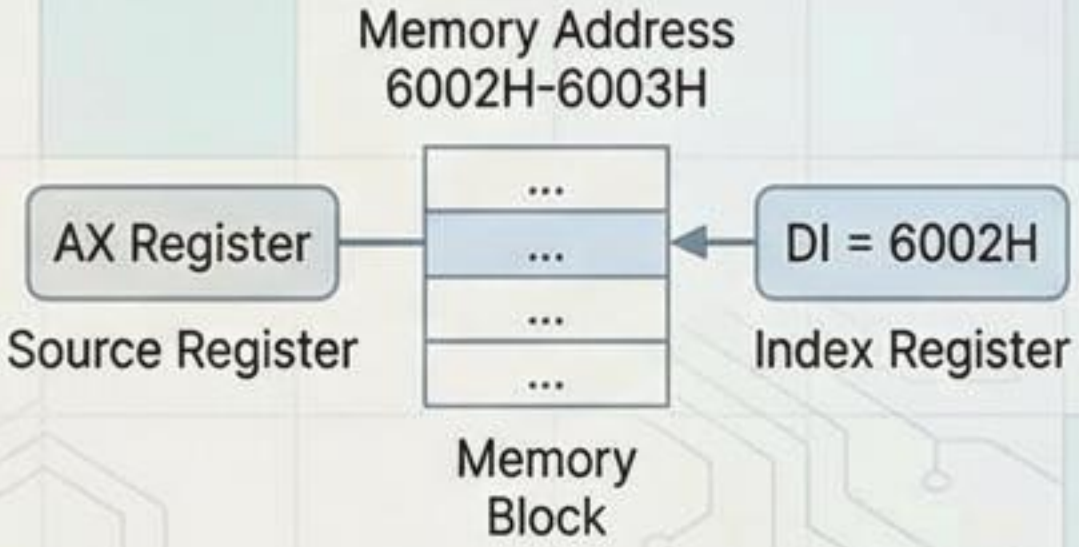
Memory Block with Data

AX Register

Destination Register

```
MOV AX, [SI]
```

Example: Storing



Memory Address 6002H-6003H

AX Register

Source Register

...

...

...

...

Memory Block

DI = 6002H

Index Register

```
MOV [DI], AX
```

Ideal for string or array traversal where the index changes inside loops.

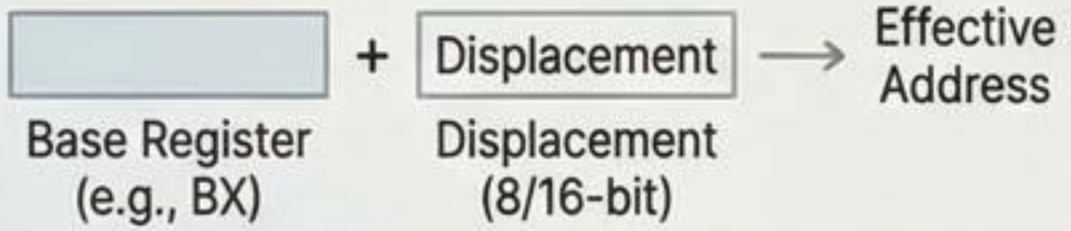
Register Relative Addressing Mode

Mechanism and Application Example

Concept & Syntax

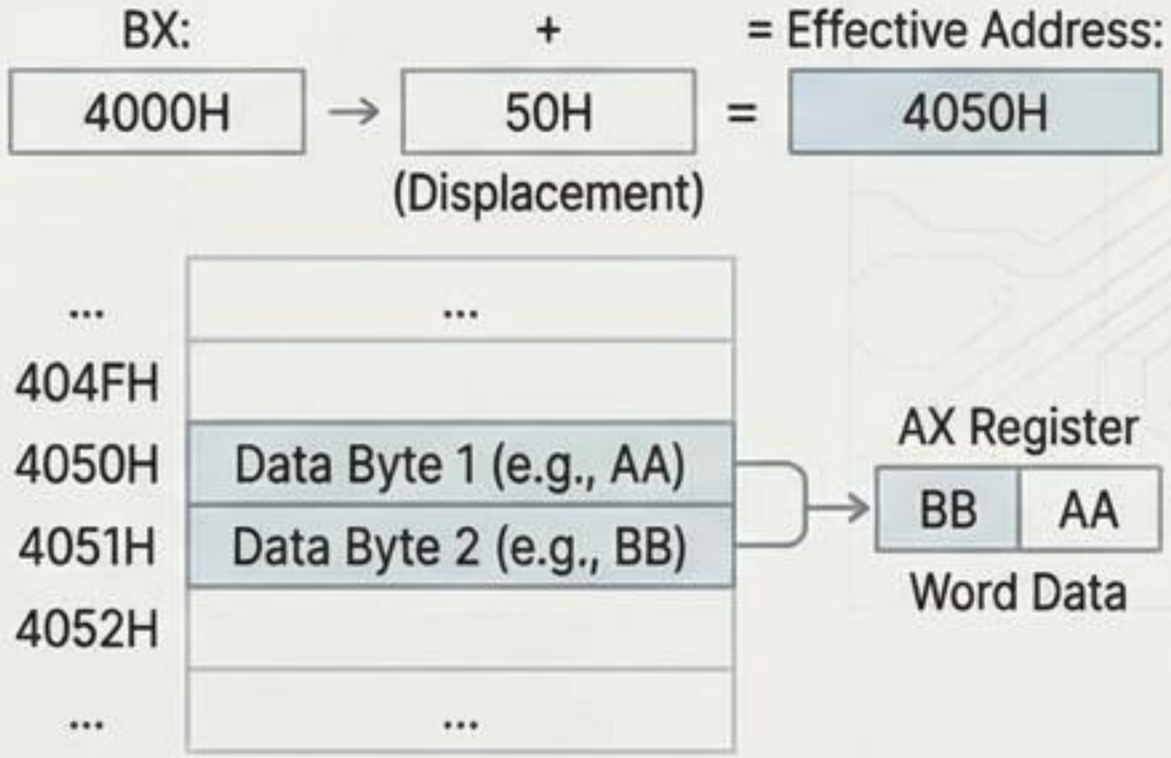
An 8- or 16-bit displacement is added to a base register (BX or BP).

```
MOV AX, 50H[BX]       (Primary Syntax)
MOV AX, [BX + 50H]    (Alternative Syntax)
```



Detailed Example Calculation

Assume BX = 4000H



Fetches the word at memory locations 4050H-4051H into AX.

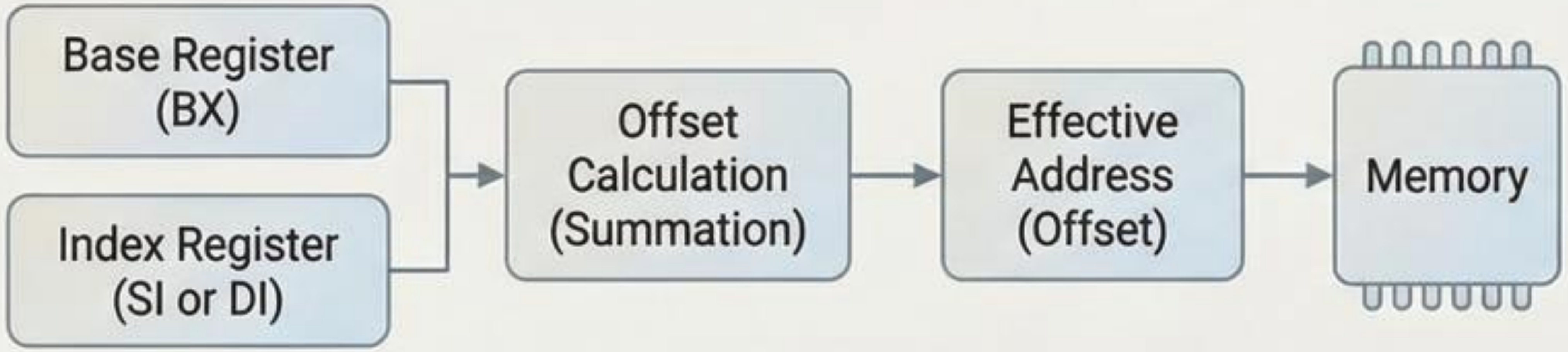
Use Cases & Benefits

- Accessing structure fields (e.g., `struct.field`).
- Accessing stack variables (local data, parameters).
- Efficient for indexed data structures with fixed offsets.

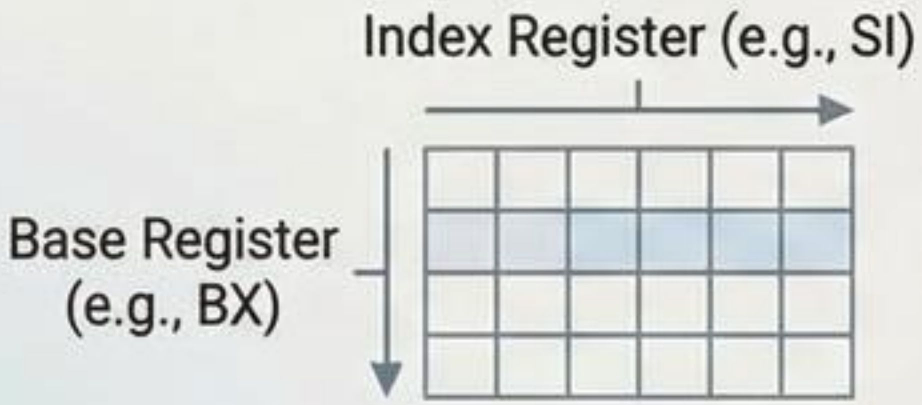
Base Indexed Addressing Mode: Detailed Breakdown

Base Indexed Addressing: [BX] + [SI] or [DI]

Combines a Base Register and an Index Register to calculate the effective address offset.



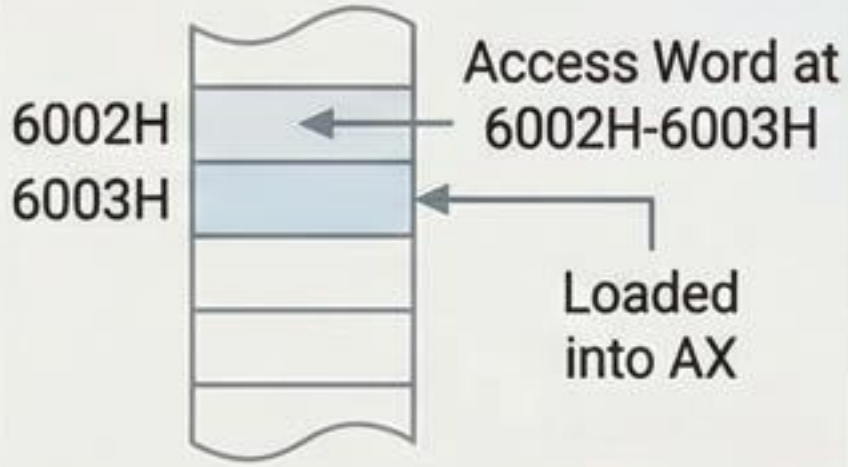
Application: 2-D Array Access



Efficiently facilitates 2-D array traversal. The **Base Register** typically points to the row, while the **Index Register** points to the column within that row.

Example Instruction:
`MOV AX,[SI][BX]`

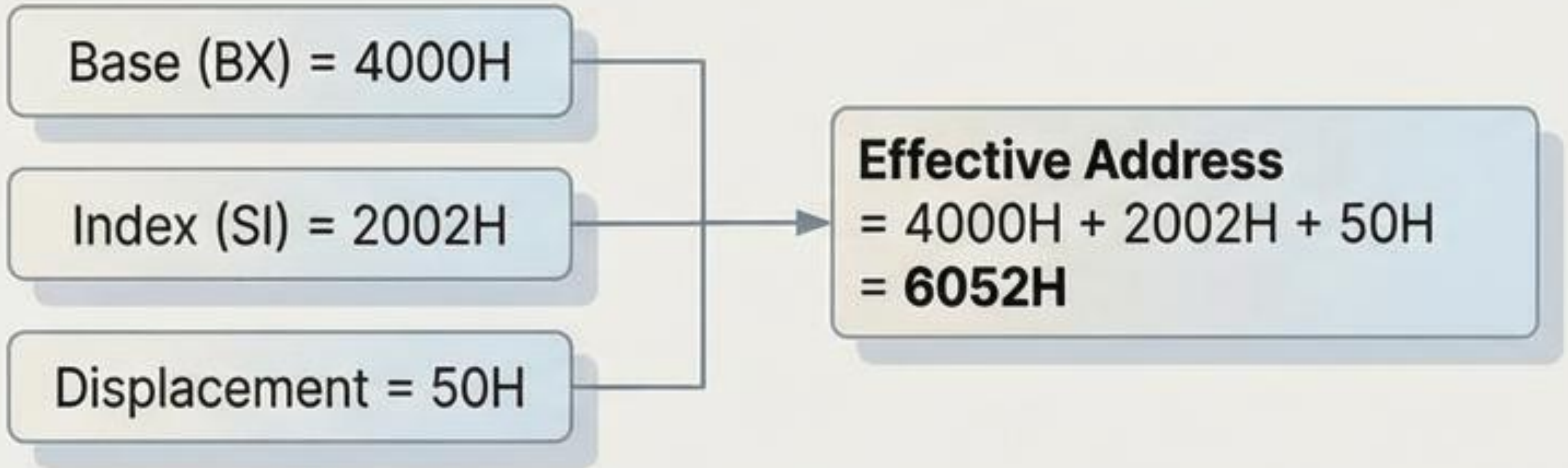
Register Values:	BX = 4000H (Base)
	SI = 2002H (Index)

Calculation:	4000H + 2002H = 6002H (Offset)
Memory Access:	 6002H 6003H Access Word at 6002H-6003H Loaded into AX

Base Indexed with Displacement Addressing.

Adds displacement to base plus index registers.

Example Analysis: MOV AX,50H[SI][BX]

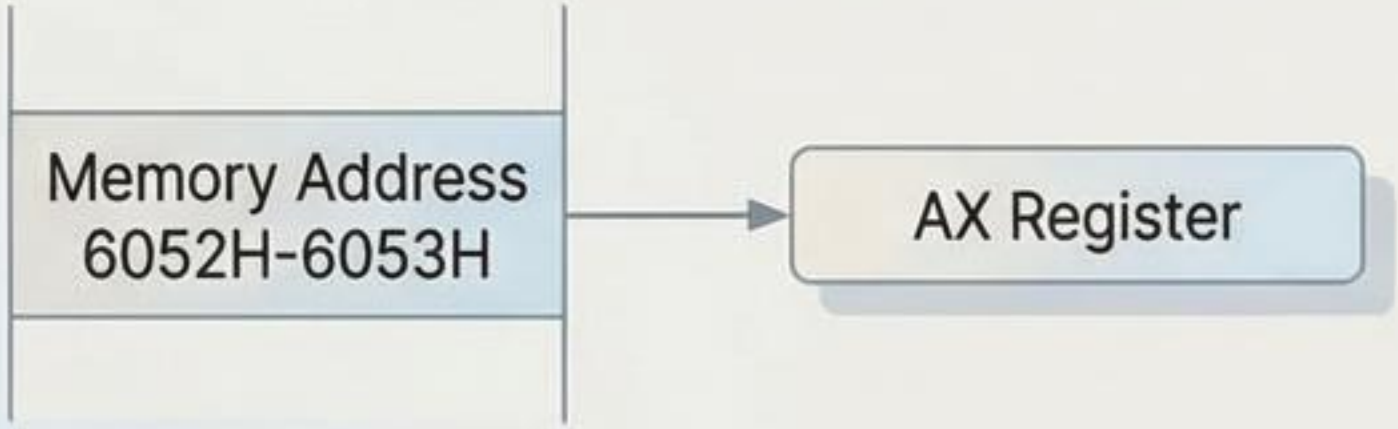


Base (BX) = 4000H

Index (SI) = 2002H

Displacement = 50H

Effective Address
= 4000H + 2002H + 50H
= **6052H**



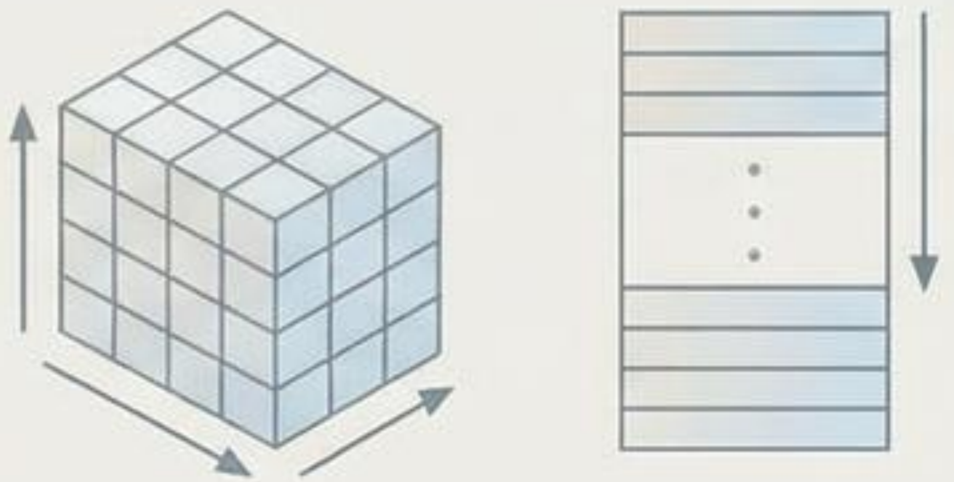
Memory Address 6052H-6053H

AX Register

Loads AX with 16-bit data from memory addresses 6052H and 6053H.

Applications & Use Cases

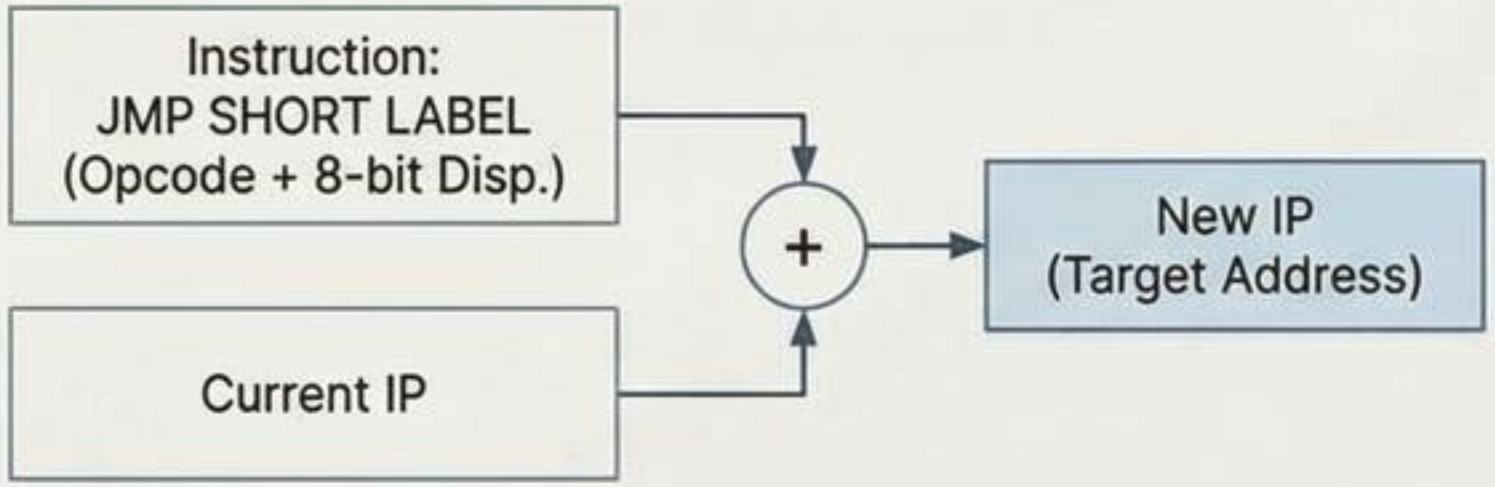
- Handles complex data structures.
- Accesses 3-D arrays.
- Manages stack frames with flexibility.



8086 Intrasegment Jumps: Direct & Indirect Addressing

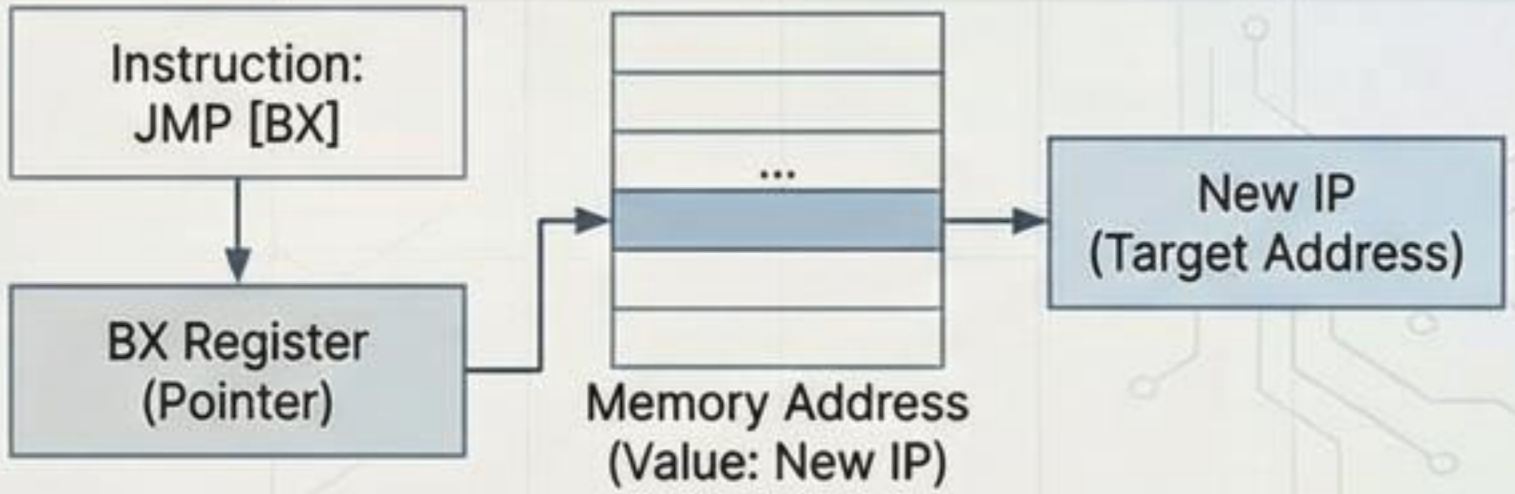
Concept: Jumps remain within the current code segment. CS register is unchanged; only the Instruction Pointer (IP) is modified.

1. Intrasegment Direct: JMP SHORT LABEL



- Embeds an 8-bit signed displacement, which is added to the current IP to calculate the new target address.

2. Intrasegment Indirect: JMP [BX]

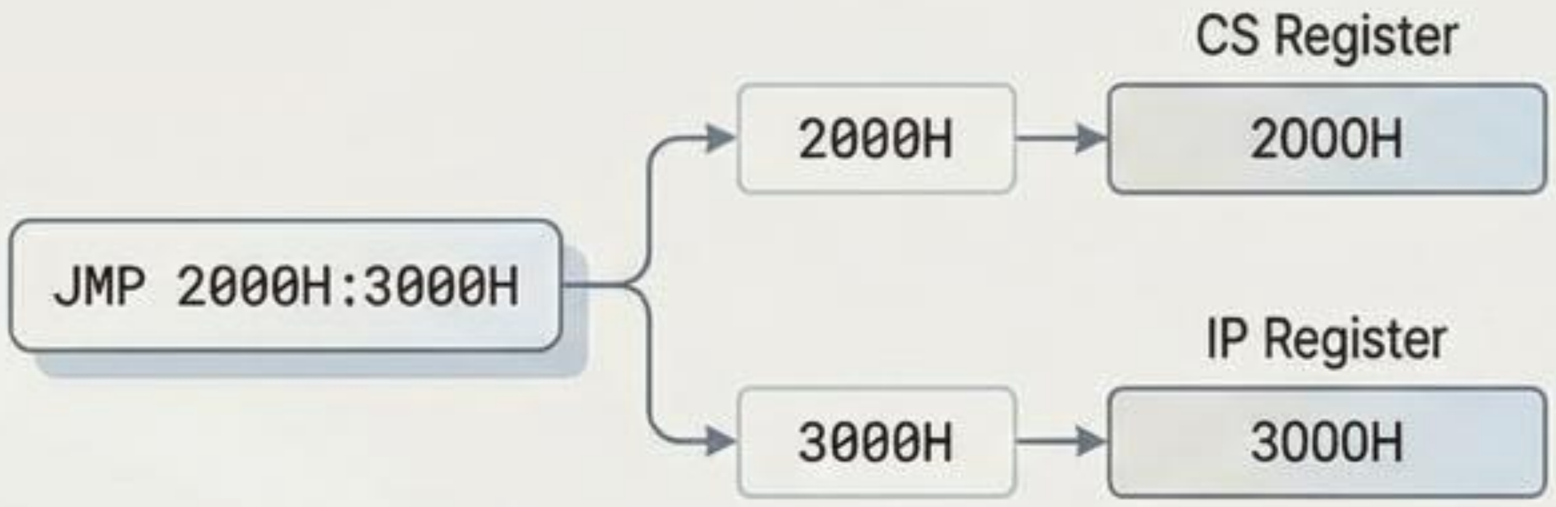


- Loads the new IP value directly from the memory location pointed to by the BX register.

Intersegment Control Transfer

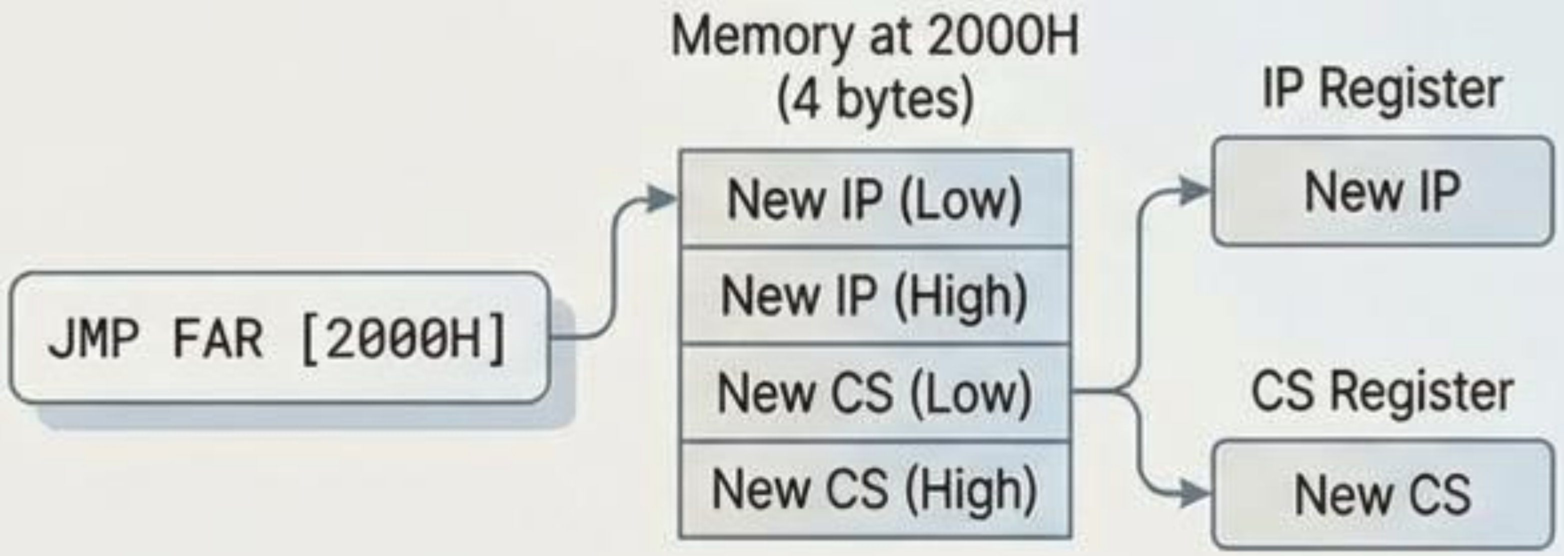
Enabling calls to routines in separate 64 kB segments.

Intersegment Direct



Directly loads new CS and IP values from instruction.

Intersegment Indirect



Fetches 4 bytes from memory. New IP first, then New CS.

جامعة المستقبل

Question And Answer

Thank You For Your Attention