

Registers of 8086: Pointer and Index Group

Overview

Pointer and Index Registers in 8086

The 8086 microprocessor includes a set of **Pointer and Index Registers** used for **memory addressing, stack operations, and string manipulation**. All are **16-bit registers** that hold the **offset addresses** within their corresponding segments.

- They allow efficient addressing for stack, data, and code segments.
- Some are automatically updated during operations (like PUSH, POP, or string instructions).
- Together, they enhance the 8086's ability to handle procedure calls, parameter passing, and repeated operations.

Stack Pointer (SP)

Function: Holds the **offset address of the top of the stack**. Physical address:

$$\text{Stack Address} = (SS \times 10H) + SP$$

- Automatically updated when using PUSH/POP.
- Stack grows toward lower memory addresses.
- Used with the **Stack Segment (SS)**.

Example:

```
MOV AX, 1234H
PUSH AX    ; SP decreases by 2
POP BX     ; SP increases by 2
```

Base Pointer (BP)

Function: Points to data within the **Stack Segment**. Physical address:

$$\text{Address} = (SS \times 10H) + BP$$

- Commonly used to access parameters in procedures or local variables.
- Useful in stack-based addressing.

Example (Access Function Parameter):

```
MOV BP, SP
MOV AX, [BP+4]
; Access parameter from stack
```

Source Index (SI)

Function: Acts as a pointer to the **source string** in the **Data Segment (DS)**.

$$\text{Source Address} = (DS \times 10H) + SI$$

- Used in string operations such as MOVSB, CMPSB.
- Automatically incremented or decremented based on the Direction Flag (DF).

Example:

```
CLD          ; DF = 0 → increment
MOV SI, 1000H
MOV AL, [SI] ; Load from DS:1000H
```

Destination Index (DI)

Function: Points to the **destination string** in the **Extra Segment (ES)**.

$$\text{Destination Address} = (ES \times 10H) + DI$$

- Works with SI during string operations.
- Auto-updated depending on DF.

Example (String Copy):

```
CLD          ; Increment mode
REP MOVSB
; Copy bytes from DS:[SI] to ES:[DI]
```

Instruction Pointer (IP)

Function: Holds the **offset address** of the **next instruction** in the **Code Segment (CS)**.

$$\text{Code Address} = (CS \times 10H) + IP$$

- Automatically updated as instructions execute.
- Modified by branch/jump/call instructions.
- Cannot be directly modified by MOV instruction.

Example:

```
JMP LABEL
; IP updated with offset of LABEL
CALL PROC
; IP pushed, jumps to procedure
```

Memory Segments and Pointer Relationships

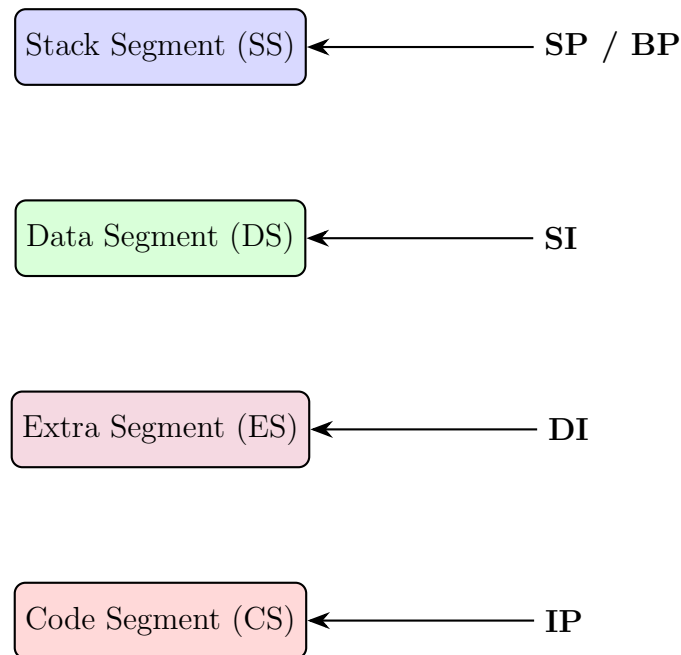


Figure: Pointer and Index Registers with their respective segments in 8086

Summary Table

Register	Associated Segment	Use	Key Operations / Example
SP	SS	Stack Top Pointer	Auto-updated on PUSH/POP
BP	SS	Access data in stack frame	Access local variables in functions
SI	DS	Source string pointer	Used in MOVSB, CMPSB
DI	ES	Destination string pointer	Used in MOVSB, SCASB, STOSB
IP	CS	Instruction sequencing	Automatically updated or via JMP/CALL

Additional Notes

- **SP, BP, SI, DI** are general-purpose registers optimized for addressing tasks.
- **IP** is dedicated to instruction control and cannot be modified directly.
- Together, these registers make stack handling, procedure calls, and string processing efficient.