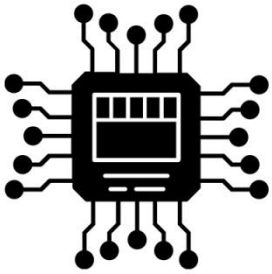


قـســـــــــم علوم الذكاء الاصطناعي

DEPARTMENT OF ARTIFICIAL INTELLIGENCE

SUBJECT:

Microprocessor

CLASS:

SECOND

LECTURER:

Dr. Abdulkadhem A. Abdulkadhem

LECTURE: (4)

Microprocessor Registers

1. Introduction

Registers are **high-speed storage locations located inside the CPU**. They are used to temporarily store **data**, **addresses**, and **intermediate results during instruction execution**. Compared to main memory, registers provide the fastest access time and therefore play a critical role in the performance and correct execution of assembly language programs.

Understanding the register organization of the 8086 microprocessor is a fundamental requirement for learning assembly language programming, addressing modes, instruction execution, and memory management.

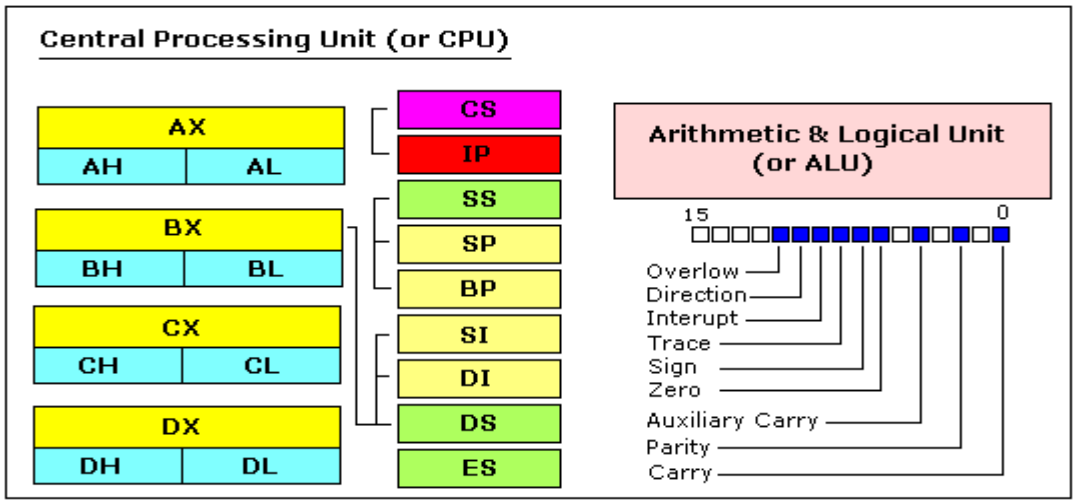
2. Register Set Overview

The register set of a microprocessor varies from one architecture to another in terms of:

- Number of registers
- Register size
- Functional specialization

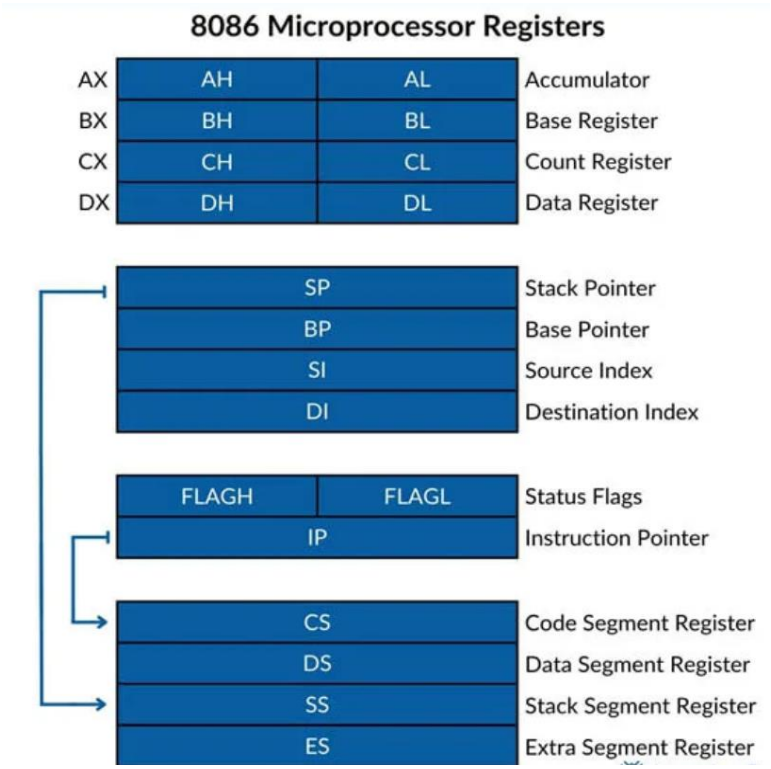
In the 8086 microprocessor, all registers are **16-bit wide**, and some of them can be accessed as two independent **8-bit registers**. The 8086 registers are classified into four main categories:

1. Data Registers (General Purpose Registers)
2. Pointer and Index Registers
3. Flag Register (Status Register)
4. Segment Registers



3. Data Registers (General Purpose Registers)

Data registers are primarily used to store operands, intermediate results, and sometimes memory addresses. All data registers can participate in arithmetic, logical, and data transfer operations. Each register is 16-bit long and can be divided into two 8-bit registers (high and low bytes).



3.1 AX Register (Accumulator Register)

- AX is known as the **Accumulator Register**.
- It is heavily used by the Arithmetic Logic Unit (ALU).
- Many arithmetic, logical, and input/output instructions implicitly use AX.
- AX is divided into:
  - AH (Accumulator High – 8 bits)
  - AL (Accumulator Low – 8 bits)

AX is frequently used in multiplication and division instructions and in data transfer operations with I/O ports.

3.2 BX Register (Base Register)

- BX is known as the **Base Register**.
- It is commonly used to store the base address of data in memory.
- BX plays a key role in **based addressing modes**.
- BX is divided into:
  - BH (Base High)
  - BL (Base Low)

The BX register is often used when accessing variables stored in the data segment.

3.3 CX Register (Count Register)

- CX is known as the **Count Register**.
- It is primarily used as a counter in loop instructions.
- It is also used in shift and rotate instructions.
- CX is divided into:
  - CH (Count High)
  - CL (Count Low)

The LOOP instruction automatically decrements CX and checks its value to control iteration.

3.4 DX Register (Data Register)

- DX is known as the **Data Register**.
- It is used in input/output operations to hold port addresses.
- It is also used together with AX in multiplication and division instructions.
- DX is divided into:
  - DH (Data High)
  - DL (Data Low)

In some DOS interrupt services, DX is used to hold the offset address of data, while the segment register defines the memory segment.

4. Pointer and Index Registers

Pointer and index registers are mainly used to **store offsets that reference memory locations**. These offsets are combined with segment registers to form physical memory addresses.

4.1 Stack Pointer (SP)

- SP holds the offset of the top of the stack.
- It always works with the Stack Segment (SS).
- The stack operates according to the **First-In Last-Out (FILO)** principle.

4.2 Base Pointer (BP)

- BP is used to access data stored on the stack.
- Like SP, it works with the Stack Segment (SS) by default.
- BP is commonly used in procedure calls and parameter passing.

4.3 Source Index (SI)

- SI holds the offset address of the source operand.
- It is mainly used in string manipulation instructions.
- By default, SI works with the Data Segment (DS).

4.4 Destination Index (DI)

- DI holds the offset address of the destination operand.
- It is also used in string manipulation instructions.
- By default, DI works with the Extra Segment (ES).

5. Flag Register (Status Register)

The Flag Register is a **16-bit register used to reflect the status of the processor after executing arithmetic or logical instructions**. Only **9 bits** of this register are used as flags, while the remaining bits are reserved.

The flags are divided into two main groups:

- **Conditional (Status) Flags**
- **Control Flags**

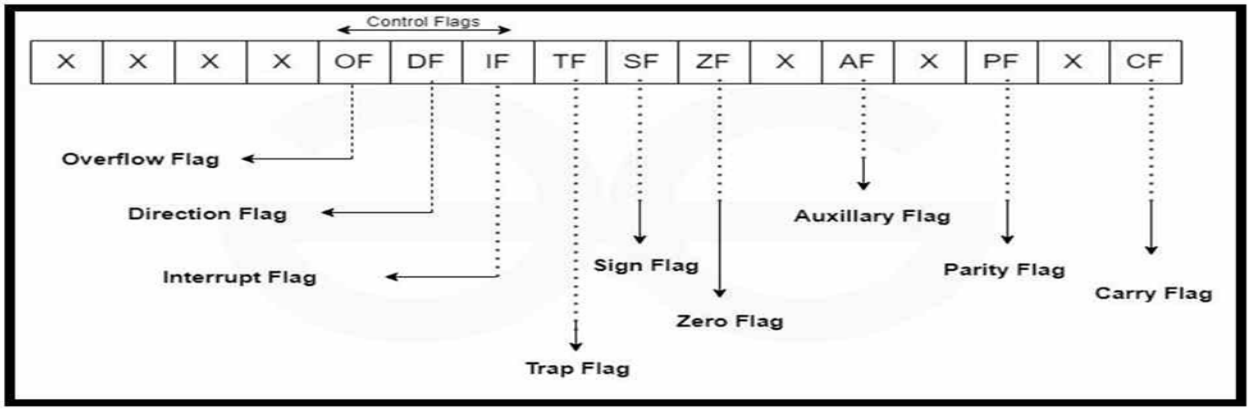


Figure : Flag Register

5.1 Conditional (Status) Flags

These flags indicate the result of the last arithmetic or logical operation.

a) Carry Flag (CF)

- Indicates a carry or borrow out of the most significant bit.
- Used in multi-byte arithmetic operations.

b) Auxiliary Carry Flag (AF)

- Used mainly in Binary-Coded Decimal (BCD) operations.
- Set when there is a carry from the lower nibble (4 bits).

c) Parity Flag (PF)

- Set to 1 if the number of 1-bits in the result is even.
- Checked only for the least significant byte.

d) Zero Flag (ZF)

- Set to 1 if the result of an operation is zero.

e) Sign Flag (SF)

- Reflects the sign of the result.
- Set to 1 if the result is negative.

f) Overflow Flag (OF)

- Set when a signed arithmetic overflow occurs.

5.2 Control Flags

These flags control the execution behavior of the processor.

a) Trap Flag (TF)

- Enables single-step execution.
- Used for debugging purposes.

b) Interrupt Flag (IF)

- Enables or disables maskable interrupts.

c) Direction Flag (DF)

- Controls the direction of string operations.
- DF = 0: Forward processing
- DF = 1: Backward processing

6. Segment Registers

The 8086 microprocessor uses a **segmented memory architecture** to access up to 1MB of physical memory using 16-bit registers. Memory is divided into logical segments, each with a base address stored in a segment register.

The physical address is calculated using the following equation:

$$Physical\ Address = (Segment\ Register \times 16) + Offset$$

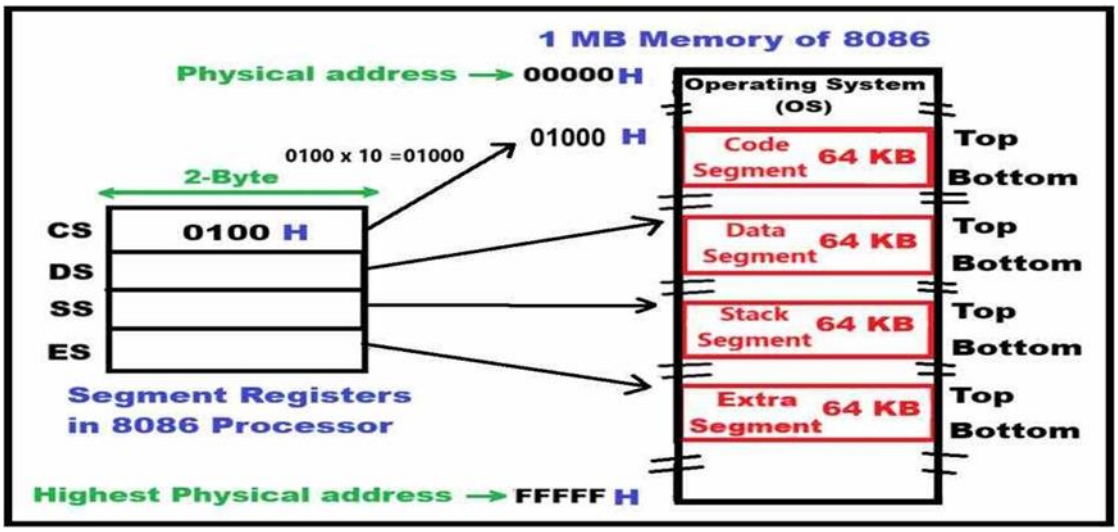


Figure :Segment Registers and 1MB Memory

6.1 Code Segment Register (CS)

- Holds the base address of the code segment.
- Contains executable program instructions.

6.2 Data Segment Register (DS)

- Holds the base address of the data segment.
- Stores variables, constants, and arrays.

6.3 Extra Segment Register (ES)

- Holds the base address of an additional data segment.
- Mainly used in string operations.

6.4 Stack Segment Register (SS)

- Holds the base address of the stack segment.
- Used for procedure calls, returns, and local storage.

7. Summary

The register organization of the 8086 microprocessor provides a powerful and flexible mechanism for data processing, memory addressing, and control flow. Mastering the function and proper usage of each register is essential for understanding assembly language programming and low-level computer architecture.

This knowledge forms the foundation for upcoming topics such as addressing modes, instruction execution, stack operations, and interrupt handling.

Check your understanding

? Easy Level (12 Questions)

Q1. Registers in the 8086 microprocessor are best described as:
A) Permanent storage devices
B) High-speed memory inside the CPU
C) Secondary memory units
D) Input/output devices

Q2. Compared to main memory, registers provide:
A) Larger storage capacity
B) Slower access time

C) Faster access time
D) External storage

Q3. All registers in the 8086 microprocessor are:
A) 8-bit wide
B) 12-bit wide
C) 16-bit wide
D) 32-bit wide

Q4. Which register is known as the Accumulator?
A) BX
B) CX
C) DX
D) AX

Q5. Which register is mainly used as a counter in loop instructions?
A) AX
B) BX
C) CX
D) DX

Q6. Which of the following is a general-purpose register?
A) CS
B) AX
C) IP
D) FLAGS

Q7. Which register is divided into AH and AL?
A) BX
B) CX
C) DX
D) AX

Q8. The stack in 8086 follows which principle?
A) FIFO
B) LIFO
C) FILO
D) Random access

Q9. Which register holds the offset of the top of the stack?
A) BP
B) SP

C) SI
D) DI

Q10. Which register stores the base address of the code segment?
A) DS
B) SS
C) CS
D) ES

Q11. Which flag is set when the result of an operation is zero?
A) CF
B) ZF
C) SF
D) OF

Q12. Which register controls the direction of string operations?
A) IF
B) TF
C) DF
D) CF

□ Medium Level (13 Questions)

Q13. Which register pair is used together during multiplication and division operations?
A) BX and CX
B) AX and DX
C) SI and DI
D) BP and SP

Q14. The BX register is mainly used for:
A) Stack control
B) Loop counting
C) Based addressing
D) Interrupt handling

Q15. Which register is commonly used to access procedure parameters on the stack?
A) SP
B) BP
C) SI
D) DI

Q16. Which register is the default source index in string operations?
A) DI
B) BP
C) SI
D) BX

Q17. By default, the Destination Index (DI) works with which segment register?
A) DS
B) CS
C) SS
D) ES

Q18. Which flag indicates a carry or borrow from the most significant bit?
A) ZF
B) OF
C) CF
D) AF

Q19. The Auxiliary Carry Flag (AF) is mainly used in:
A) Signed arithmetic
B) Logical operations
C) BCD operations
D) Loop control

Q20. Which flag reflects the sign (positive or negative) of the result?
A) ZF
B) PF
C) SF
D) OF

Q21. Which flag is checked only for the least significant byte of the result?
A) SF
B) PF
C) CF
D) OF

Q22. Which control flag enables single-step execution for debugging?
A) IF
B) DF
C) TF
D) CF

Q23. The Interrupt Flag (IF) controls:
A) Stack operations
B) String direction
C) Maskable interrupts
D) Arithmetic overflow

Q24. The physical memory address in 8086 is calculated using:
A) Segment + Offset
B) Segment × 2 + Offset
C) Segment × 16 + Offset
D) Offset × 16 + Segment

Q25. Which register holds the base address of the stack segment?
A) DS
B) CS
C) ES
D) SS

□ Hard Level (10 Questions)

Q26. Which register cannot be directly used for arithmetic operations?
A) AX
B) BX
C) CX
D) CS

Q27. Which flag is set when a signed arithmetic overflow occurs?
A) CF
B) ZF
C) OF
D) PF

Q28. If the Direction Flag (DF) is set to 1, string operations will process data:
A) Forward
B) Backward
C) Randomly
D) In parallel

Q29. Which register pair allows access to memory using based indexed addressing?
A) AX and DX

B) BX and SI
C) CX and DI
D) SP and BP

Q30. Which register combination defines the execution location of the next instruction?
A) CS and SP
B) DS and BX
C) CS and IP
D) SS and BP

Q31. Which flag is affected when the result of an operation has an even number of 1s?
A) SF
B) ZF
C) PF
D) OF

Q32. Which register is most appropriate for passing parameters to procedures?
A) AX
B) BX
C) BP
D) DI

Q33. Which segment register is primarily used during string destination operations?
A) CS
B) DS
C) ES
D) SS

Q34. Which of the following registers is NOT divided into high and low bytes?
A) AX
B) BX
C) CX
D) SP

Q35. Mastery of register organization is MOST essential for understanding:
A) High-level programming
B) File systems
C) Addressing modes and instruction execution
D) Computer networking