1. General-Purpose Registers in 8086

The 8086 microprocessor has eight general-purpose registers, each 16-bit wide. These are divided into two categories:

- **Data Registers:**
    - **AX (Accumulator Register):** Used for arithmetic, logic, and data movement operations.
    - **BX (Base Register):** Used for addressing memory locations.
    - **CX (Count Register):** Used for counting operations and loop instructions.
    - **DX (Data Register):** Used in I/O operations and multiplication/division instructions.
- **Pointer and Index Registers:**
    - **SP (Stack Pointer):** Points to the top of the stack.
    - **BP (Base Pointer):** Helps in accessing function parameters stored in the stack.
    - **SI (Source Index):** Used for string operations and memory addressing.
    - **DI (Destination Index):** Used for string operations and memory addressing.

---

2. Concept of Segmented Memory & Advantages

The 8086 microprocessor follows a **segmented memory model**, where the 1MB memory is divided into segments of 64KB each. The segments include:

- **Code Segment (CS):** Stores executable instructions.
- **Data Segment (DS):** Holds program data.
- **Stack Segment (SS):** Manages the stack.
- **Extra Segment (ES):** Used for additional data storage.

Advantages of Segmented Memory:

- Allows efficient memory management.
- Enables addressing of 1MB memory using 16-bit registers.
- Supports modular programming.
- Provides memory protection by separating code, data, and stack.

---

3. Physical Address Formation in 8086

The **physical address** in 8086 is computed using **segment and offset** addresses.

Formula:

Physical Address=Segment Address×10H+Offset\text{Physical Address} = \text{Segment Address} \times 10H + \text{Offset}

Example:
If **CS = 2000H** and **IP = 1234H**, then:

Physical Address=2000H×10H+1234H=20000H+1234H=21234H\text{Physical Address} = 2000H \times 10H + 1234H = 20000H + 1234H = 21234H

---

4. Assembly Program to Multiply 05H and 04H Without Using Arithmetic Instructions

```
MOV AL, 05H
MOV BL, 04H
MOV CL, 00H

SHIFT_LOOP:
SHR BL, 1
JNC SKIP
ADD CL, AL

SKIP:
SHL AL, 1
CMP BL, 00H
JNZ SHIFT_LOOP

MOV [5000H], CL
HLT
```

This code uses **shift and add** instead of the `MUL` instruction.

---

5. Logical Instructions in 8086

a) SHL (Shift Left) / SAL (Shift Arithmetic Left)

- Shifts bits **left** and fills the lowest bit with **0**.
- Multiplication by **2** for each shift.
- Example: `SHL AL, 1` shifts `00001101` → `00011010`.

b) SHR (Shift Right)

- Shifts bits **right** and fills the highest bit with **0**.
- Equivalent to **division by 2** per shift.
- Example: `SHR AL, 1` shifts `00001101` → `00000110`.

c) SAR (Shift Arithmetic Right)

- Similar to `SHR`, but retains the sign bit (MSB).
- Example: `SAR AL, 1` shifts `10001101` (signed -115) → `11000110` (signed -57).

d) ROR (Rotate Right)

- Rotates bits **right**, moving the lowest bit to the highest position.
- Example: `ROR AL, 1` shifts `10001101` → `11000110`.

e) ROL (Rotate Left)

- Rotates bits **left**, moving the highest bit to the lowest position.
- Example: `ROL AL, 1` shifts `10001101` → `00011011`.

Let me know if you need further clarifications!

1. General-Purpose Registers in 8086

- **AX** – Accumulator, used in arithmetic and I/O.
- **BX** – Base register, used for memory addressing.
- **CX** – Count register, used in loops and shifts.
- **DX** – Data register, used in multiplication and I/O.
- **SP** – Stack pointer, points to stack top.
- **BP** – Base pointer, used for stack addressing.
- **SI** – Source index, used in string operations.
- **DI** – Destination index, used in string operations.

---

2. Segmented Memory & Advantages

- **Segments:** CS (Code), DS (Data), SS (Stack), ES (Extra).
- **Advantages:**
  - Access **1MB** memory using 16-bit registers.
  - Efficient memory management.
  - Supports modular programming.

---

3. Physical Address Formation

Formula:

Physical Address=Segment×10H+Offset\text{Physical Address} = \text{Segment} \times 10H + \text{Offset}

Example: If **CS = 2000H** and **IP = 1234H**, then:

Physical Address=20000H+1234H=21234H\text{Physical Address} = 20000H + 1234H = 21234H

---

4. Assembly Program to Multiply 05H and 04H Without MUL

```
MOV AL, 05H
MOV BL, 04H
MOV CL, 00H

SHIFT_LOOP:
SHR BL, 1
JNC SKIP
ADD CL, AL

SKIP:
SHL AL, 1
CMP BL, 00H
JNZ SHIFT_LOOP

MOV [5000H], CL
HLT
```

---

5. Logical Instructions

- **SHL/SAL –** Left shift, multiplies by 2 per shift.
- **SHR –** Right shift, divides by 2 per shift, fills MSB with 0.
- **SAR –** Right shift, retains sign bit (for signed numbers).
- **ROR –** Rotates bits right, LSB moves to MSB.
- **ROL –** Rotates bits left, MSB moves to LSB.