**Basic I/O, ASCII Conversion, Arithmetic, and Flags**

### **1. 🔡 Basic Input/Output Using INT 21h**

The DOS interrupt INT 21h allows communication between the processor and the user. We control the operation by setting a value in the AH register.

| **AH Value** | **Operation** |
| --- | --- |
| 01h | Single Character Input |
| 02h | Single Character Output |
| 09h | String Output (ends with $) |

#### **💡 Example: Read and Echo a Character**

MOV AH, 01h ; Function to read character

INT 21h ; User types 'A' → AL = 41h (ASCII of A)

MOV DL, AL ; Copy to DL

MOV AH, 02h ; Function to display character

INT 21h ; Outputs 'A'

### **2. 🔢 ASCII to Decimal and Decimal to ASCII**

* When a digit like '3' is typed, its ASCII value is 33h.
* To convert it to numeric (3), subtract 30h.
* To display a number as a digit, add 30h to get ASCII.

#### **💡 Example: Add 2 and 5 (user inputs)**

; Input '2' and '5' using INT 21h AH=01h

SUB AL, 30h ; Convert ASCII '2' → 2

MOV BL, AL ; Store in BL

; Input '5'

SUB AL, 30h ; Convert ASCII '5' → 5

MOV BH, AL ; Store in BH

; Add

MOV AL, BL

ADD AL, BH ; AL = 7

ADD AL, 30h ; Convert 7 → ASCII '7' for output

### **3. 🔠 Uppercase to Lowercase Character Conversion**

In ASCII:

* 'A' = 65 (41h), 'a' = 97 (61h)
* So: 'a' = 'A' + 32

#### **💡 Example:**

MOV AL, 'D' ; AL = 44h

ADD AL, 32 ; AL = 44h + 20h = 64h → 'd'

This lets us display lowercase letters from uppercase input.

### **4. 🧮 Hex Digit (A–F) to Decimal Conversion**

To convert 'C' to decimal:

* ASCII 'C' = 43h
* 'C' - 'A' = 2
* 2 + 10 = 12

#### **💡 Example:**

MOV AL, 'C'

SUB AL, 'A' ; AL = 2

ADD AL, 10 ; AL = 12

ADD AL, 30h ; Convert 12 → ASCII 'C' for output

### **5. 🖥️ String Output Using INT 21h AH=09h**

To print a message, the string must end with a dollar sign ($).

#### **💡 Example:**

msg DB 'Hello, World!$'

LEA DX, msg ; Load address of message

MOV AH, 09h

INT 21h ; Outputs: Hello, World!

### **6. 🔁 New Line (Line Feed + Carriage Return)**

You need to print **both ASCII characters** to move to the next line:

* Carriage Return = 13 (0Dh)
* Line Feed = 10 (0Ah)

#### **💡 Example:**

newline DB 13,10,'$'

LEA DX, newline

MOV AH, 09h

INT 21h ; Moves cursor to new line

### **7. ➕➖✖️➗ Arithmetic Operations Using Input**

After converting characters to decimal, we can perform arithmetic:

#### **💡 Example: Multiply**

MOV AL, 2

MOV BL, 4

MUL BL ; AX = AL \* BL = 8

#### **💡 Example: Divide**

MOV AX, 8

MOV BL, 2

DIV BL ; AL = 4 (quotient), AH = 0 (remainder)

### **8. ⚑ Flag Register Overview**

After arithmetic, the 8086 sets **status flags** based on the result:

| **Flag** | **Meaning** |
| --- | --- |
| **ZF (Zero Flag)** | Set if result = 0 |
| **SF (Sign Flag)** | Set if result is negative |
| **CF (Carry Flag)** | Set if carry/borrow occurs |
| **OF (Overflow Flag)** | Set if signed overflow occurs |

#### **💡 Example: Overflow Demo**

MOV AL, 80h ; -128 in signed

MOV BL, 80h ; -128

ADD AL, BL ; AL = 0 (wrong), overflow occurs

; OF = 1, CF = 0

This demonstrates why **flag checking** is important after arithmetic operations.