**8-BIT ADDITION**

**EXP NO: 1**

**AIM:**

To write an assembly language program to implement 8-bit addition using 8085 processor.

**ALGORITHM:**

1. Start the program by loading the first data into the accumulator.
2. Move the data to a register.
3. Get the second data and load it into the accumulator.
4. Add the two register contents.
5. Check for carry.
6. Store the value of sum and carry in the memory location.
7. Halt.

**PROGRAM:**

LDA 8500

MOV B, A

LDA 8501

ADD B

STA 8502

RST 1

**INPUT:**

; 8-bit addition using AAA (works for decimal digits 0–9)

; Emulator: emu8086, MASM, TASM

data segment

num1 db '4' ; first digit (ASCII '4')

num2 db '7' ; second digit (ASCII '7')

sum db ? ; result (ASCII)

data ends

code segment

assume cs:code, ds:data

start:

mov ax, data

mov ds, ax

; Load numbers (ASCII digits)

mov al, num1

sub al, 30h ; convert to decimal

mov bl, num2

sub bl, 30h

; Add

add al, bl ; AL = num1 + num2

aaa ; adjust to unpacked BCD

add ax, 3030h ; convert result back to ASCII

mov sum, al ; store ones digit

mov dl, ah ; check if tens digit is nonzero

; Print tens digit (if not '0')

cmp dl, '0'

je print\_ones

mov ah, 02h

int 21h

print\_ones:

mov dl, sum ; print ones digit

mov ah, 02h

int 21h

; Exit

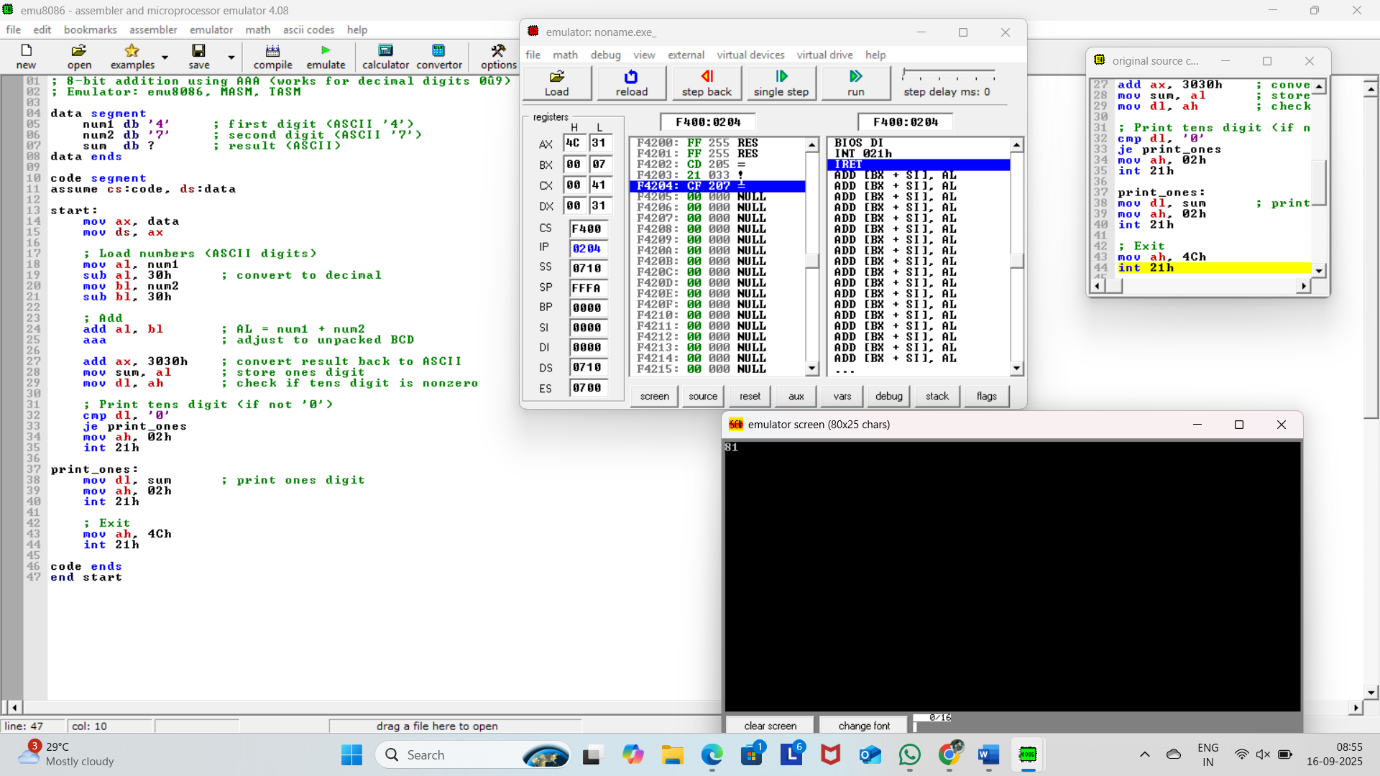
mov ah, 4Ch

int 21h

code ends

end start

**OUTPUT:**

****

**RESULT:** Thus the program was executed successfully using 8085 processor simulator.