**Computer Organization and Architecture Lab**

**LAB ASSIGNMENT – 8**

**E. Likhith**

**AP22110010386**

**CSE-F**

**1. Write a program in assembly language to display a two-digit number on the screen. The two-digits number is required to be taken in the program itself.**

**Code:**

ORG 100h

; Two-digit number to be displayed (let's say 57)

MOV AL, 64 ; Load the two-digit number into AL

; Split the number into tens and units

MOV BL, 10 ; Set divisor to 10 to separate tens and units

DIV BL ; Divide AL by 10, AL = quotient (tens), AH = remainder (units)

; Store the quotient (tens) and remainder (units)

MOV BH, AL ; Store the tens digit in BH

MOV BL, AH ; Store the units digit in BL

MOV DX, OFFSET msg\_1

MOV AH, 09h

INT 21h

; Convert tens digit to ASCII

ADD BH, '0' ; Convert the tens digit to ASCII

MOV DL, BH ; Move the ASCII tens digit to DL for printing

MOV AH, 02h ; DOS interrupt to print a character

INT 21h ; Print the tens digit

; Convert units digit to ASCII

ADD BL, '0' ; Convert the units digit to ASCII

MOV DL, BL ; Move the ASCII units digit to DL for printing

MOV AH, 02h ; DOS interrupt to print a character

INT 21h ; Print the units digit

; Terminate the program

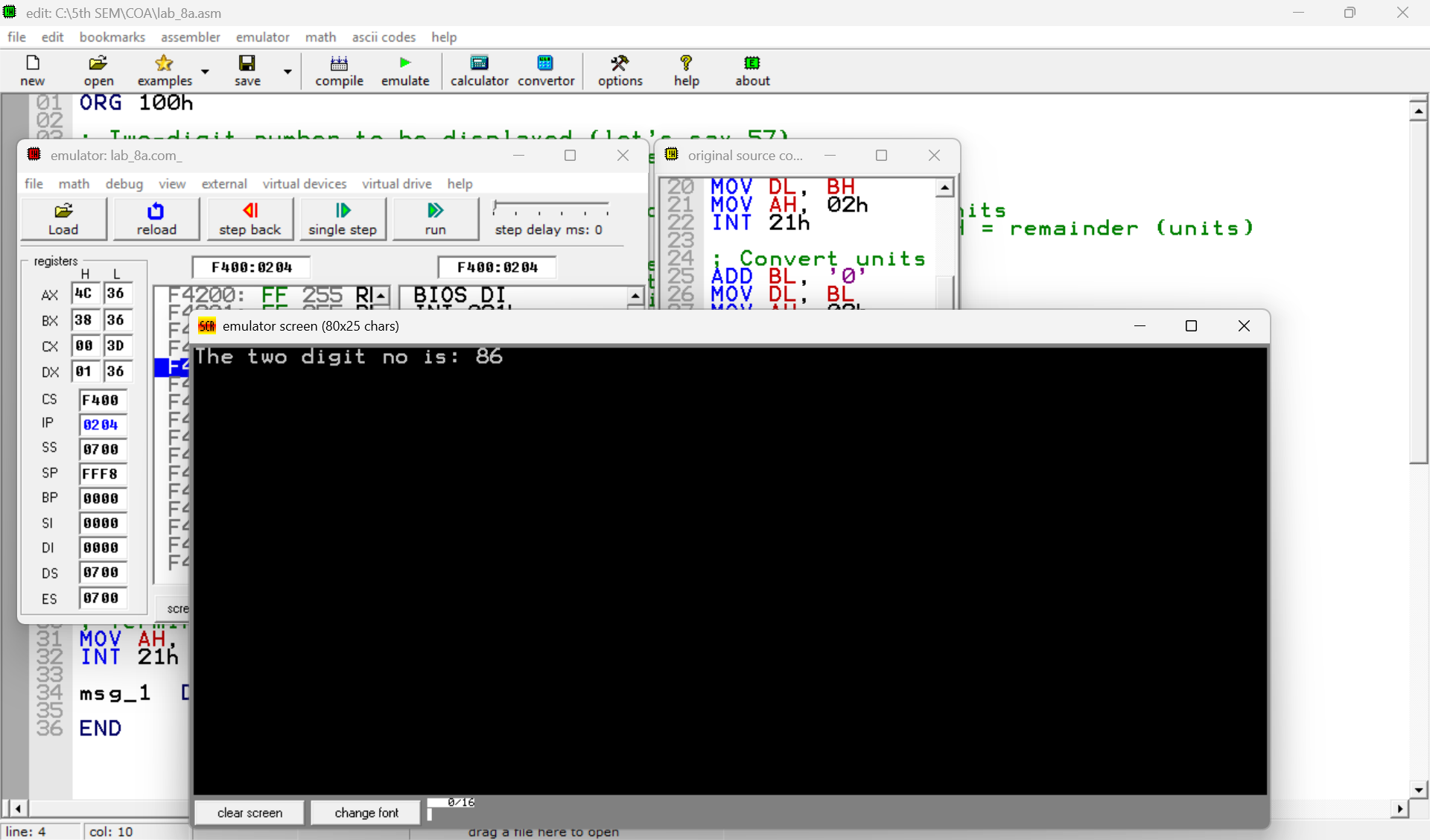
MOV AH, 4Ch ; DOS interrupt to exit the program

INT 21h

msg\_1 DB 'The two digit no is : $'

END

**OUTPUT:**

****

**A screenshot of a computer

Description automatically generated**

**Practice Set:**

**2. Write an assembly language program to take two single-digit integers from the user and print the result of addition on the screen.**

**Code:**

ORG 100h

; Prompt for the first single-digit number

mov dx, offset msg\_input1

mov ah, 09h

int 21h

; Get first digit

mov ah, 01h

int 21h

mov bl, al ; Store first digit in BL

cmp al, '0' ; Check if it's a valid digit

jl NotDigit

cmp al, '9'

jg NotDigit

; Display the first digit

mov dx, offset msg\_output1

mov ah, 09h

int 21h

mov dl, bl

mov ah, 02h

int 21h

; Prompt for the second single-digit number

mov dx, offset msg\_input2

mov ah, 09h

int 21h

; Get second digit

mov ah, 01h

int 21h

mov cl, al ; Store second digit in CL

cmp al, '0' ; Check if it's a valid digit

jl NotDigit

cmp al, '9'

jg NotDigit

; Display the second digit

mov dx, offset msg\_output2

mov ah, 09h

int 21h

mov dl, cl

mov ah, 02h

int 21h

; Perform addition of the two digits

mov dx, offset msg\_add

mov ah, 09h

int 21h

sub bl, '0' ; Convert first digit from ASCII to numeric value

sub cl, '0' ; Convert second digit from ASCII to numeric value

add bl, cl ; Add the two digits

; Check if the result is a two-digit number (>= 10)

cmp bl, 10

jl SingleDigit ; If less than 10, it's a single-digit result

; Handle two-digit result

mov dl, 1 ; Tens place is 1 for numbers between 10-18

add dl, '0' ; Convert tens place to ASCII

mov ah, 02h

int 21h

sub bl, 10 ; Adjust result for ones place (subtract 10)

add bl, '0' ; Convert ones place to ASCII

mov dl, bl

mov ah, 02h

int 21h

jmp endprogram

SingleDigit:

; Handle single-digit result

add bl, '0' ; Convert the result to ASCII

mov dl, bl

mov ah, 02h

int 21h

jmp endprogram

; Handle invalid input

NotDigit:

mov dx, offset msg\_error

mov ah, 09h

int 21h

; End the program

endprogram:

mov ah, 4Ch

int 21h

; Data section

msg\_input1 DB "Enter first digit: $"

msg\_output1 DB 0Dh, 0Ah, "The first digit is: $"

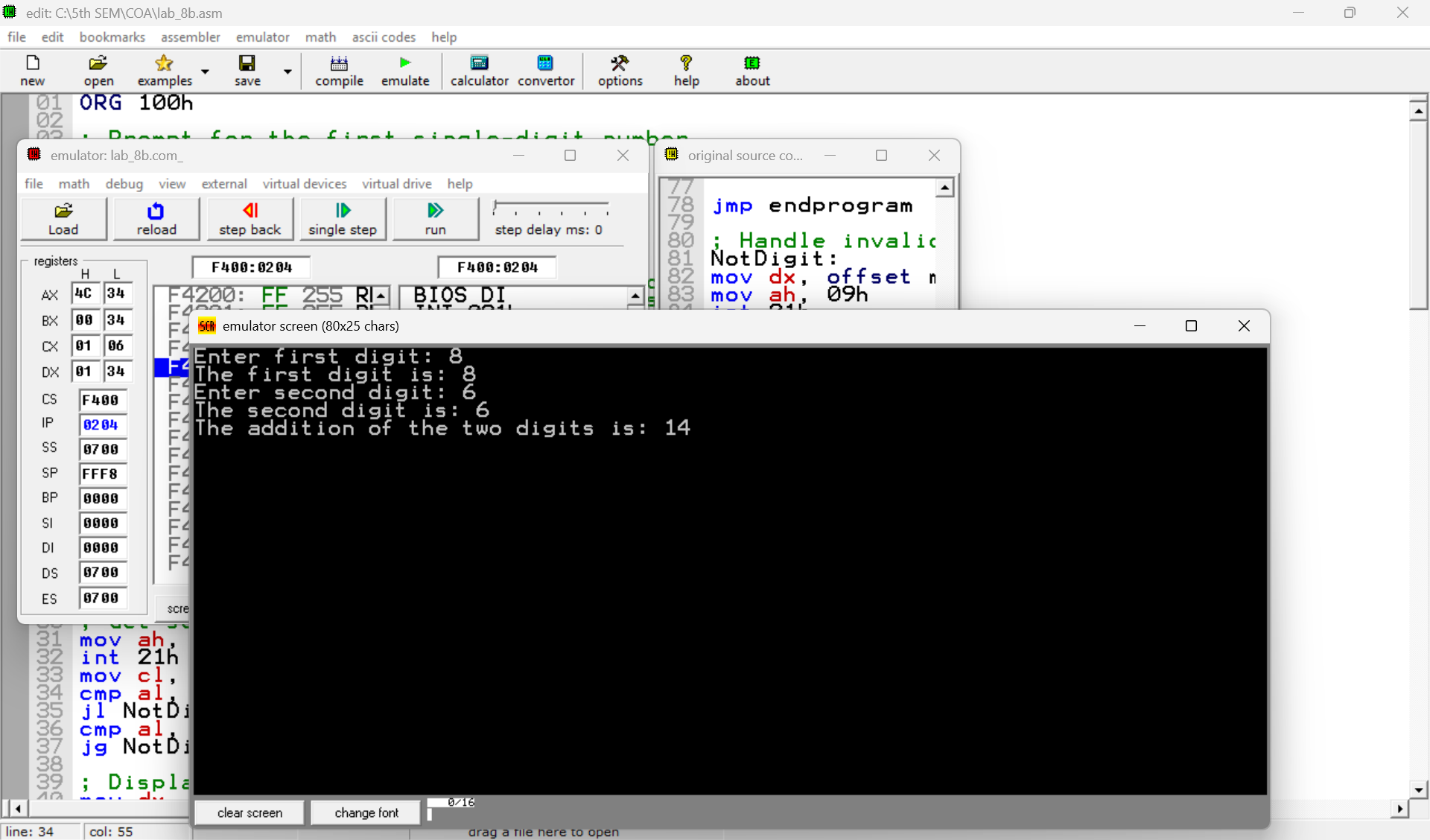
msg\_input2 DB 0Dh, 0Ah, "Enter second digit: $"

msg\_output2 DB 0Dh, 0Ah, "The second digit is: $"

msg\_add DB 0Dh, 0Ah, "The addition of the two digits is: $"

msg\_error DB 0Dh, 0Ah, "Error: Not a digit!$"

**Output:**

****

**A screenshot of a computer

Description automatically generated**