Tanuja Chakka

AP22110010446

CSE – ‘G’

Lab Task-2

Write an assembly language program to perform addition of 8-bit data.

org 100h

num1 db 32h

num2 db 19h

start:

mov al,num1;moving num1 to al register

add al,num2;moving num2 to al register(i.e, num1 and num2 are in al)

mov bl,al ;coping al value to bl,storing for the result dispaly

mov ah,al;coping al values to ah,(converting upper nibble 4bit to get character)

and ah,0F0h; mask the lower nibble(i.e,converting lower nibble to 0)

shr ah,4;shifting right by 4 to get upper nibble

add ah,30h;(converting to ascii digit)

cmp ah,39h;compare ah value,if it is less than 39h

jle print\_first\_digit

add ah,7;convert to ascii

print\_first\_digit:

mov dl,ah;coping ah value to dl for printing first digit

mov ah,02h;BIOS interrupt to display character

int 21h;converting lowwer nibble 4bit to character

mov ah,bl;coping bl values to ah

and ah,0Fh; mask the upper nibble(i.e,converting upper nibble to 0)

add ah,30h;(converting to ascii digit)

cmp ah,39h;compare ah value,if it is less than 39h

jle print\_second\_digit

add ah,7;convert to ascii

print\_second\_digit:

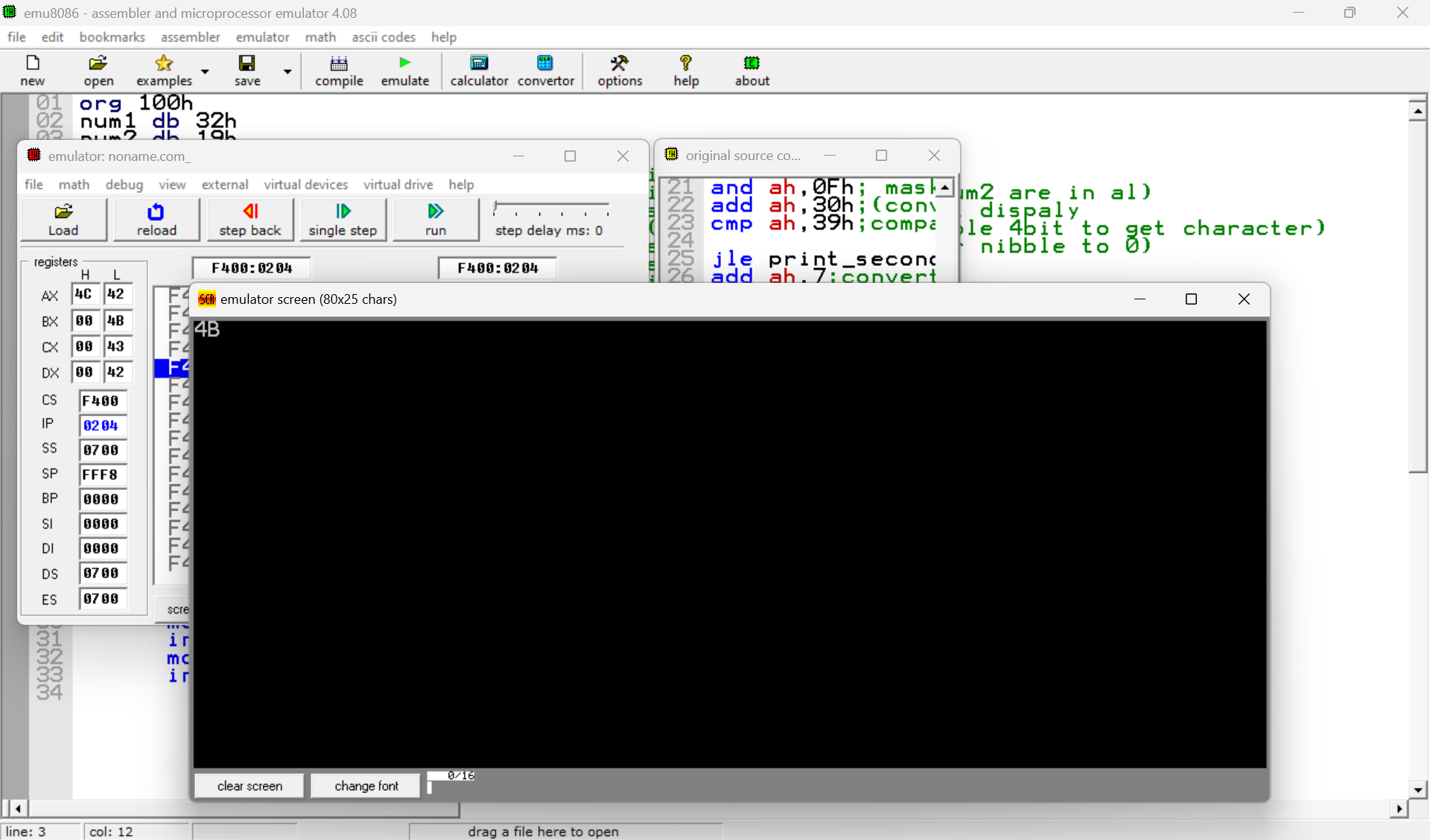
mov dl,ah;coping ah value to dl for printing first digit

mov ah,02h;BIOS interrupt to display character

int 21h

mov ah,4Ch

int 21h



Write a program in assembly language to perform addition of 16-bit data.

org 100h

num1 dw 1436h ; First 16-bit number

num2 dw 5789h ; Second 16-bit number

start:

mov ax, num1 ; Load num1 into AX register (16-bit register)

add ax, num2 ; Add num2 to AX

; Store the result in BX so we can convert it to hexadecimal

mov bx, ax

; Convert and print the upper byte (higher 8 bits)

mov ah, bh ; Move upper byte of BX to AH

call convert\_to\_hex ; Convert upper nibble to hex

mov dl, ah ; Move first character to DL for printing

mov ah, 02h ; BIOS interrupt to display character

int 21h

mov ah, bh ; Move upper byte of BX to AH again

call convert\_lower\_nibble ; Convert lower nibble to hex

mov dl, ah ; Move second character to DL for printing

mov ah, 02h ; BIOS interrupt to display character

int 21h

; Convert and print the lower byte (lower 8 bits)

mov ah, bl ; Move lower byte of BX to AH

call convert\_to\_hex ; Convert upper nibble to hex

mov dl, ah ; Move third character to DL for printing

mov ah, 02h ; BIOS interrupt to display character

int 21h

mov ah, bl ; Move lower byte of BX to AH again

call convert\_lower\_nibble ; Convert lower nibble to hex

mov dl, ah ; Move fourth character to DL for printing

mov ah, 02h ; BIOS interrupt to display character

int 21h

; End the program

mov ah, 4Ch ; Terminate the program

int 21h

convert\_to\_hex:

; Mask the upper nibble and convert it to a character

and ah, 0F0h

shr ah, 4 ; Shift the upper nibble to the lower nibble

add ah, 30h ; Convert to ASCII digit

cmp ah, 39h ; Compare if the value is less than or equal to '9'

jle skip\_conversion

add ah, 7 ; Convert to ASCII letter (A-F)

skip\_conversion:

ret ; Return from the procedure

convert\_lower\_nibble:

; Mask the lower nibble and convert it to a character

and ah, 0Fh ; Mask upper nibble, keep lower nibble

add ah, 30h ; Convert to ASCII digit

cmp ah, 39h ; Compare if the value is less than or equal to '9'

jle skip\_lower\_conversion

add ah, 7 ; Convert to ASCII letter (A-F)

skip\_lower\_conversion:

ret ; Return from the procedure

