Computer Organization and Architecture Lab

LAB ASSIGNMENT-2

Likhith Edupuganti

AP22110010386

CSE-F

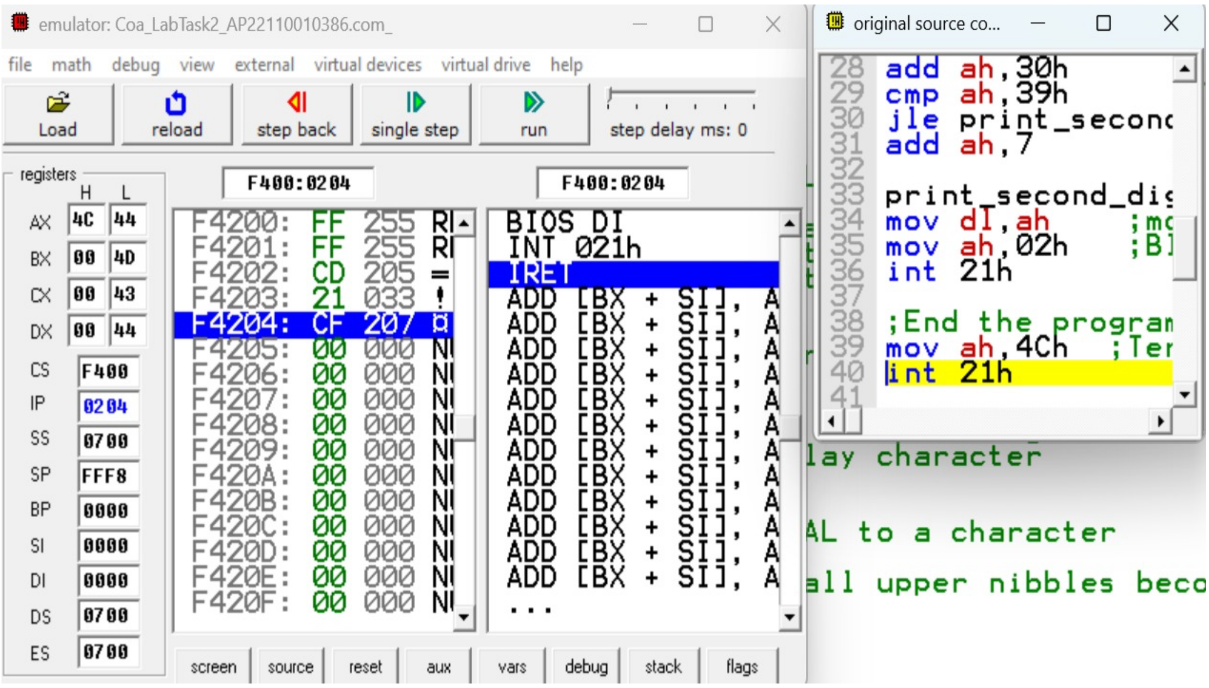
1. Write an assembly language program to perform addition of 8-bit data.

Code:

```
org 100h
num1 db 24h
num2 db 29h
start:
  mov al, num1   ;moving number1 value to AL Register
  add al, num2   ;adding number2 value with existing value in AL register

  mov bl,al

  ;to covert upper nibble(4 bits) of AL to a character
  mov ah,al
  and ah,0F0h  ;mask the lower nibble(all lower bits become 0's)
  shr ah,4     ;shift right by 4 to get upper nibble
  add ah,30h   ;convert to ASCHII digits(0-9)
  cmp ah,39h   ;compare ah value if ah is less than 39h
  jle print_first_digit
```

```
    add ah,7     ;covert to ASCHII letter(A-F) if ah>39h

print_first_digit:
    mov dl,ah    ;move first digit to DL for printing
    mov ah,02h   ;BIOS interrupt to display character
    int 21h


    ;to convert lower nibble(4 bits) of AL to a character
    mov ah,bl
    and ah,0Fh   ;mask the upper nibble(all upper nibbles become 0's)
    add ah,30h
    cmp ah,39h
    jle print_second_digit
    add ah,7

print_second_digit:
    mov dl,ah    ;move second digit 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
```

Output:



emulator: Coa_LabTask2_AP22110010386.com_

registers	H	L
AX	4C	44
BX	00	4D
CX	00	43
DX	00	44
CS	F400	
IP	0204	
SS	0700	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

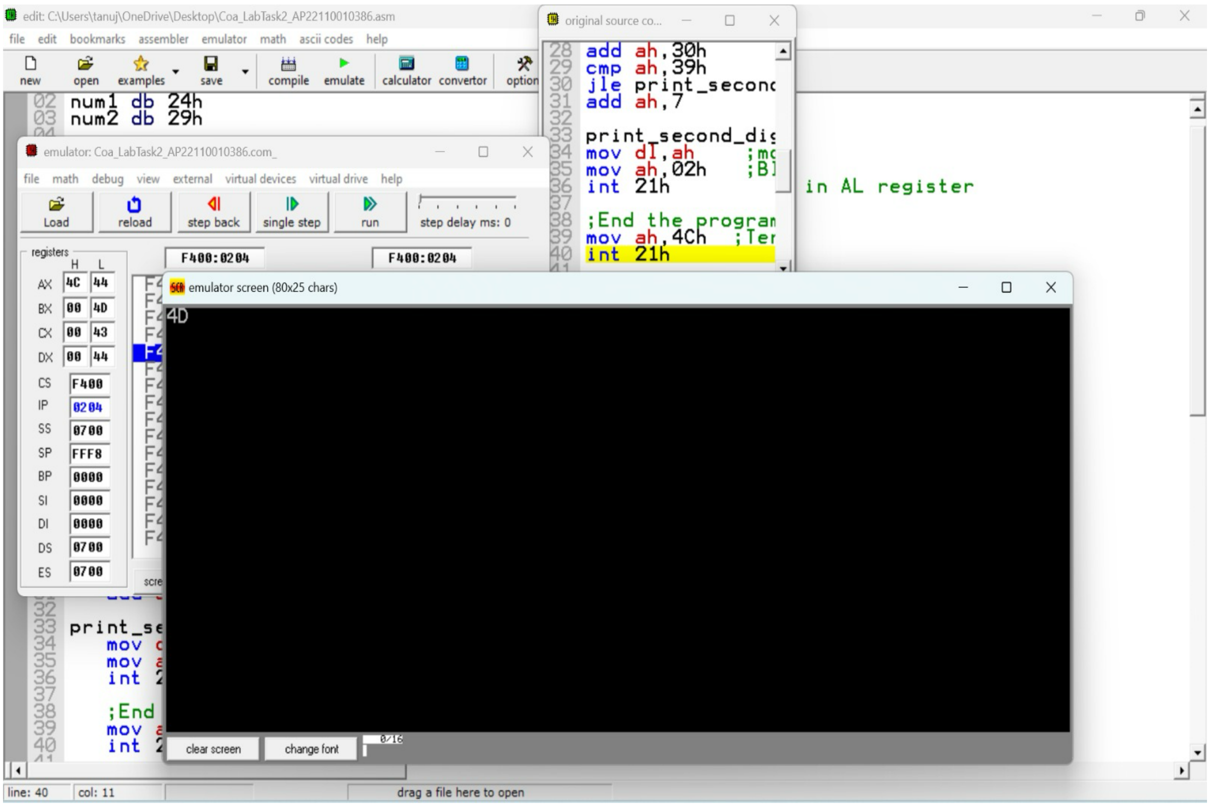
```
F400:0204

F4200: FF 255 R
F4201: FF 255 R
F4202: CD 205 =
F4203: 21 033 !
F4204: CF 207 ¤
F4205: 00 000 N
F4206: 00 000 N
F4207: 00 000 N
F4208: 00 000 N
F4209: 00 000 N
F420A: 00 000 N
F420B: 00 000 N
F420C: 00 000 N
F420D: 00 000 N
F420E: 00 000 N
F420F: 00 000 N
```

```
F400:0204

BIOS DI
INT 021h
IRET
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
ADD [BX + SI], A
...
```

original source co...

```
add ah,30h
cmp ah,39h
jle print_seconc
add ah,7

print_second_dis
mov dl,ah      ;mc
mov ah,02h     ;Bl
int 21h

;End the program
mov ah,4Ch    ;Ter
int 21h
```

```
lay character

AL to a character
all upper nibbles beco
```



edit: C:\Users\tanuj\OneDrive\Desktop\Coa_LabTask2_AP22110010386.asm

```
num1 db 24h
num2 db 29h
```

emulator: Coa_LabTask2_AP22110010386.com_

registers	H	L
AX	4C	44
BX	00	4D
CX	00	43
DX	00	44
CS	F400	
IP	0204	
SS	0700	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

F400:0204 F400:0204

original source co...

```
add ah,30h
cmp ah,39h
jle print_seconc
add ah,7

print_second_dis
mov dl,ah      ;mc
mov ah,02h     ;Bl
int 21h

;End the program
mov ah,4Ch    ;Ter
int 21h
```

```
in AL register
```

emulator screen (80x25 chars)

```
4D
```

line: 40 col: 11 drag a file here to open

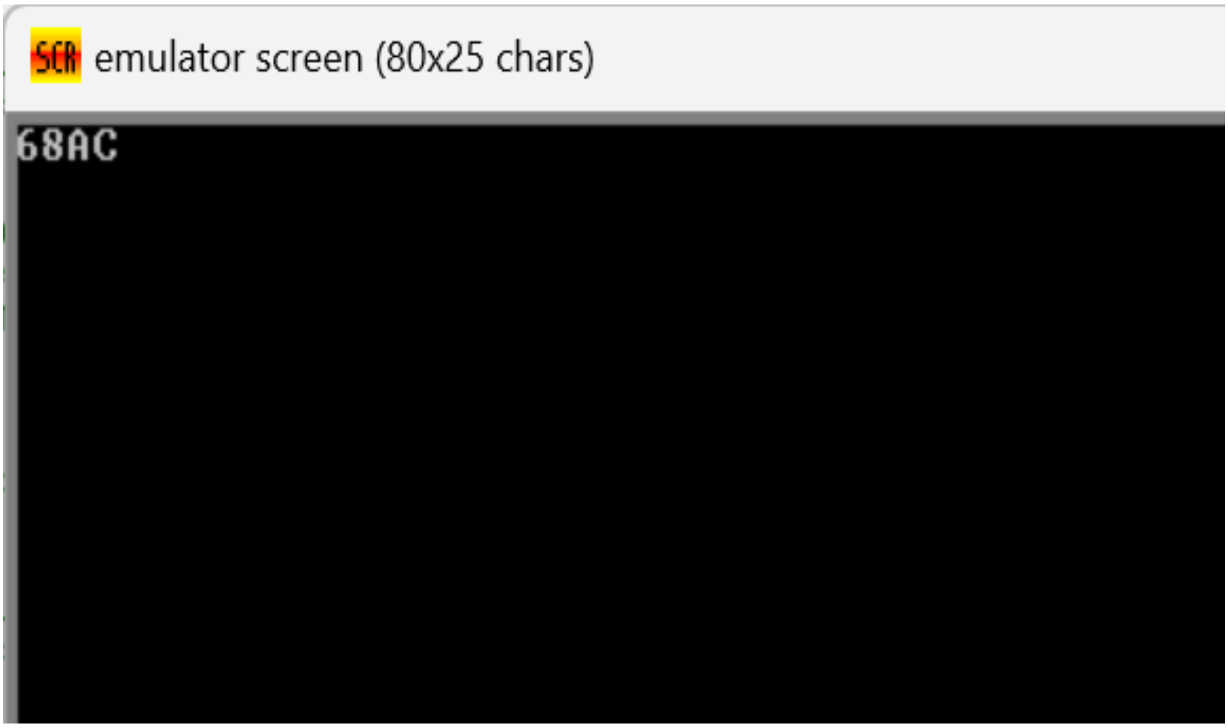
Practice set:

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

Code:

```
org 100h
num1 dw 1234h  ; define first 16-bit number
num2 dw 5678h  ; define second 16-bit number
start:
    mov ax, num1   ; move num1 to ax register
    add ax, num2  ; add num2 to ax register (ax = num1 + num2)
    ; convert result to ascii and display
    mov bx, ax    ; copy result to bx for further processing
    ; process higher byte of the result
    mov ah, bh    ; move higher byte of result to ah
    shr ah, 4     ; shift right by 4 to get upper nibble
    add ah, 30h   ; convert to ascii
    cmp ah, 39h   ; compare if less than '9'
    jle print_high_nibble
    add ah, 7     ; convert to ascii letter if necessary
print_high_nibble:
    mov dl, ah    ; move ah to dl for printing
    mov ah, 02h   ; bios interrupt to display character
    int 21h
    ; process lower nibble of the higher byte
    mov ah, bh    ; move higher byte of result to ah
    and ah, 0fh   ; mask upper nibble
    add ah, 30h   ; convert to ascii
    cmp ah, 39h   ; compare if less than '9'
    jle print_low_nibble
    add ah, 7     ; convert to ascii letter if necessary
print_low_nibble:
    mov dl, ah    ; move ah to dl for printing
    mov ah, 02h   ; bios interrupt to display character
    int 21h
    ; process upper nibble of the lower byte
    mov ah, bl    ; move lower byte of result to ah
    shr ah, 4     ; shift right by 4 to get upper nibble
    add ah, 30h   ; convert to ascii
    cmp ah, 39h   ; compare if less than '9'
    jle print_high_nibble2
    add ah, 7     ; convert to ascii letter if necessary
print_high_nibble2:
    mov dl, ah    ; move ah to dl for printing
    mov ah, 02h   ; bios interrupt to display character
    int 21h
    ; process lower nibble of the lower byte
    mov ah, bl    ; move lower byte of result to ah
    and ah, 0fh   ; mask upper nibble
    add ah, 30h    ; convert to ascii
    cmp ah, 39h    ; compare if less than '9'
    jle print_low_nibble2
    add ah, 7      ; convert to ascii letter if necessary
print_low_nibble2:
    mov dl, ah     ; move ah to dl for printing
    mov ah, 02h    ; bios interrupt to display character
    int 21h
    mov ah, 4ch    ; terminate program
    int 21h
```

Output:



```
68AC
```