ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO THỰC HÀNH MÔN VI XỬ LÝ – VI ĐIỀU KHIỂN

BÀI THỰC HÀNH SỐ 1: CỘNG – TRỪ HAI SỐ 32 BIT TRÊN VI XỬ LÝ EMU8086

Sinh viên thực hiện:

Trần Ngọc Ánh 22520077

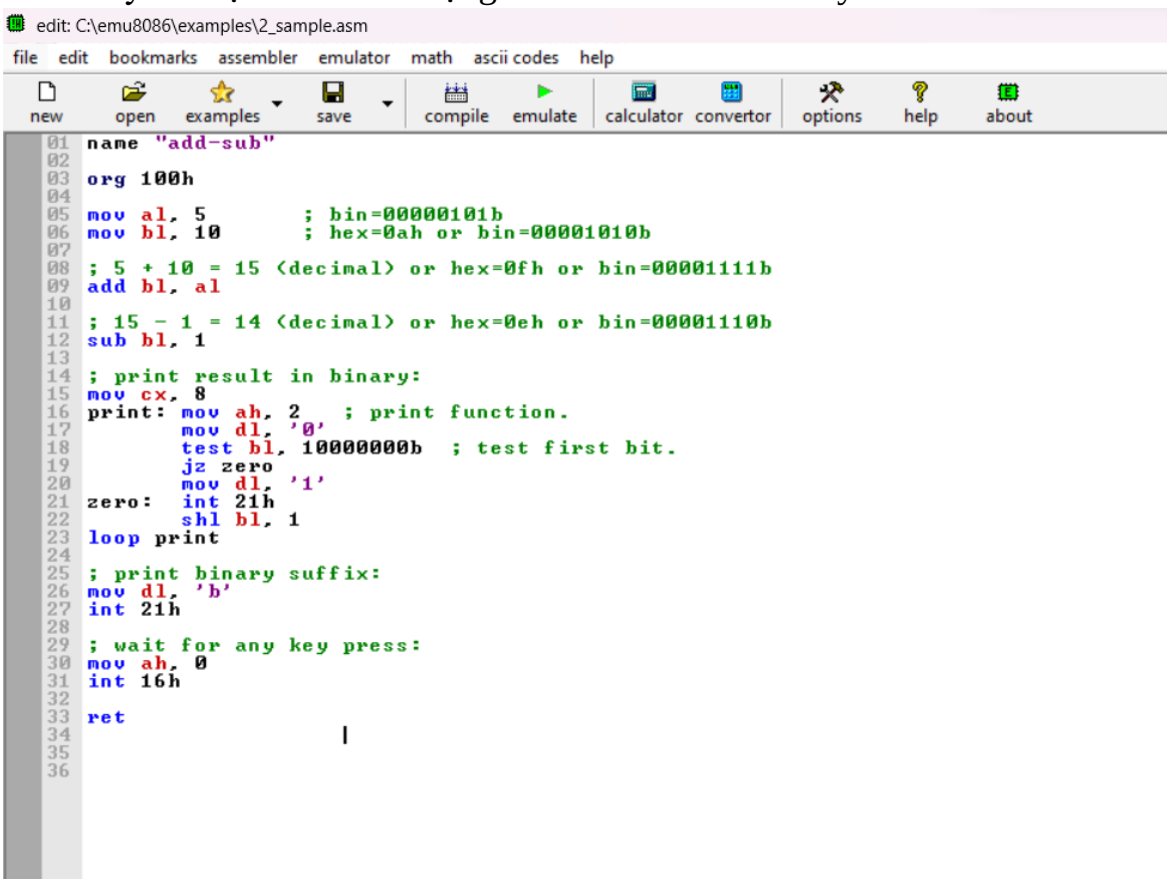
Giảng viên hướng dẫn: Phạm Minh Quân

Mã lớp: CE103.O22

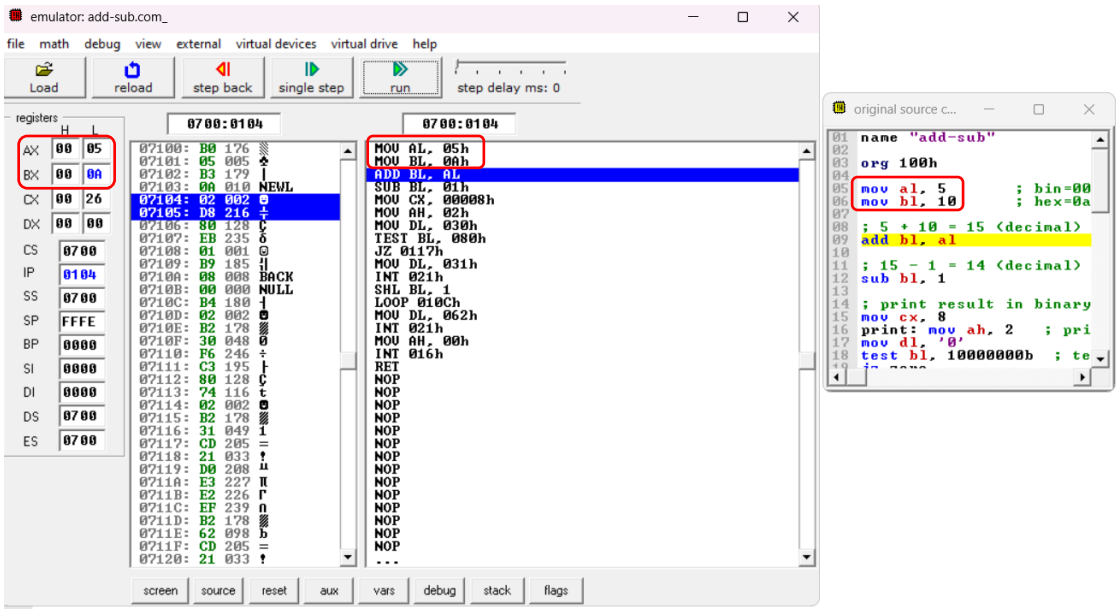
TP. HỒ CHÍ MINH, 17 THÁNG 3 NĂM 2024

1. **Mô tả cách thức hoạt động của code mẫu:**

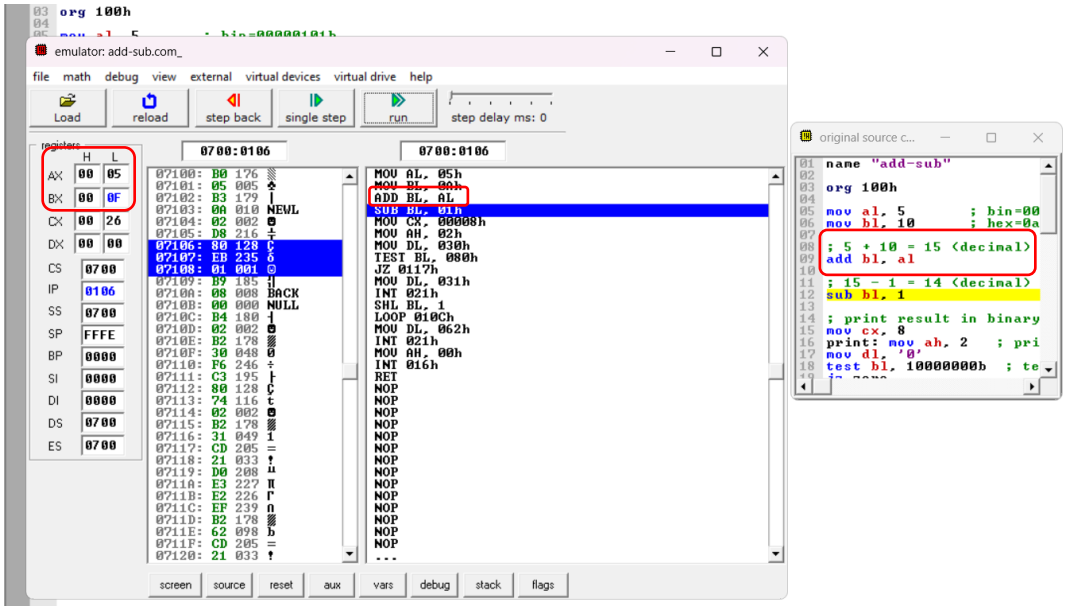
- Dưới đây là đoạn code mẫu cộng 2 số 32 bit trên vi xử lý 8086:



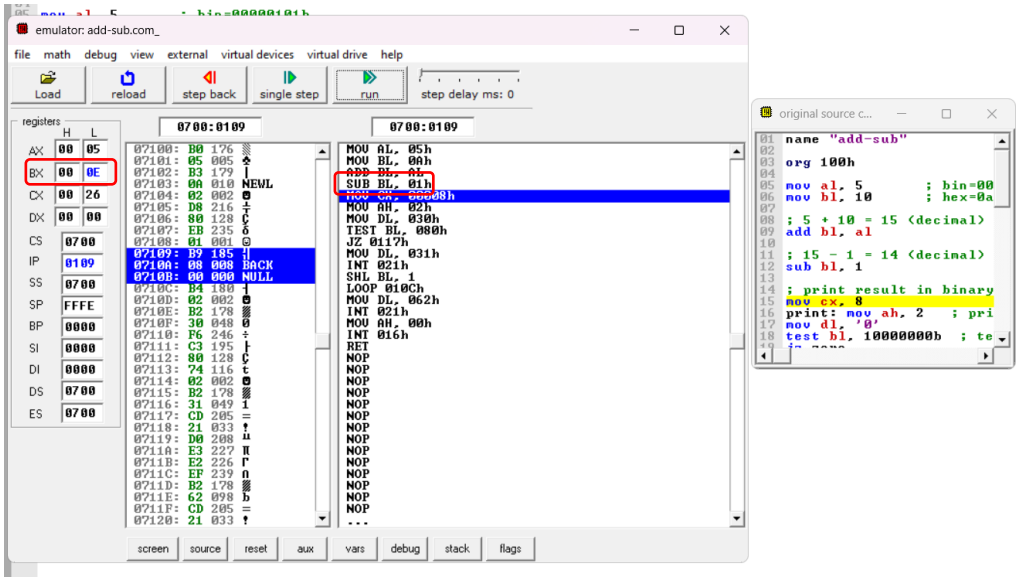
- Bước 1: Khởi tạo giá trị với thanh ghi AL = 0x05, BL = 0x0A với câu lệnh **mov al, 5** và **mov bl, 10**



- Bước 2 : Thực hiện phép cộng 2 thanh ghi BL , AL với câu lệnh : **add bl, al** . Câu lệnh trên được hiểu là BL = BL + AL, ta nhận được giá trị mới của thanh ghi BL = 0x0F.

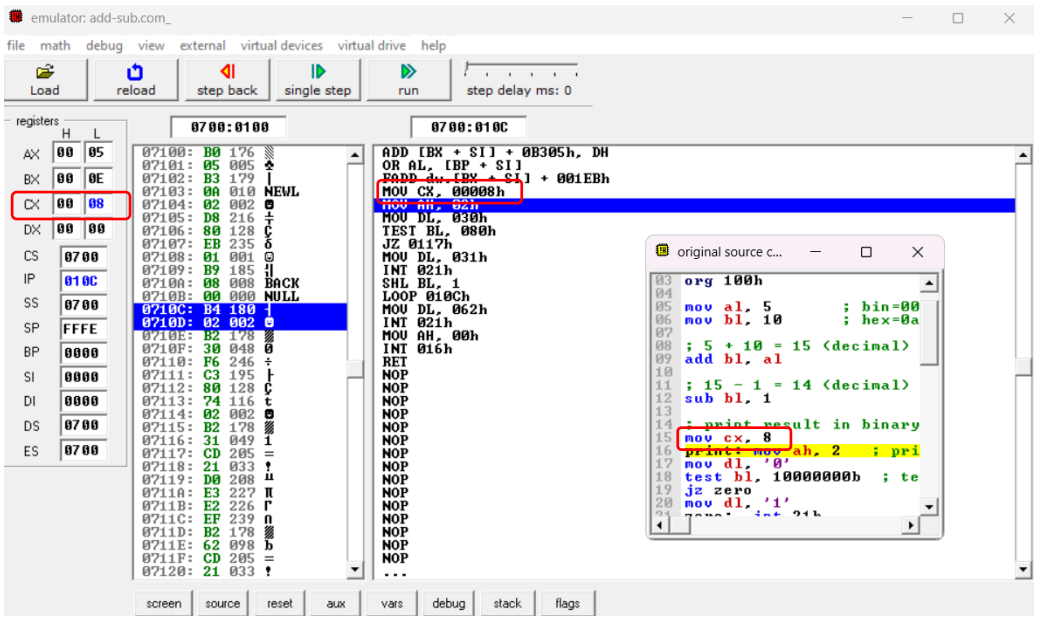


- Bước 3: Lấy giá trị của thanh ghi BL trừ đi 1 , ta có được giá trị mới của thanh ghi BL = 0x0E với câu lệnh là **sub bl, 1.**

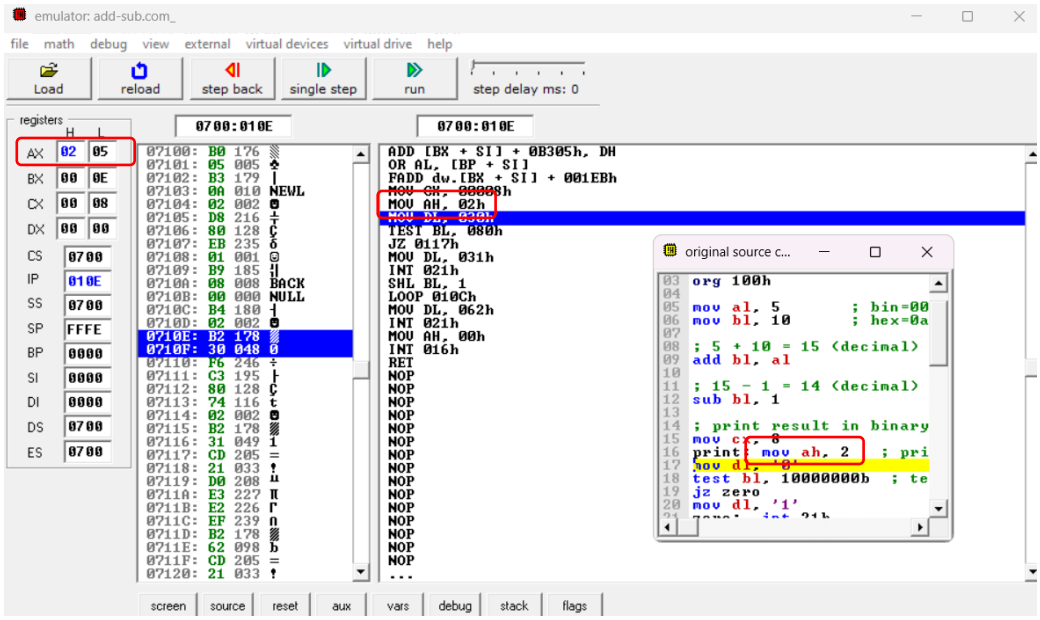


- Bước 4: In ra kết quả dưới dạng nhị phân. Sử dụng vòng lặp để in ra từng số của thanh ghi BL:

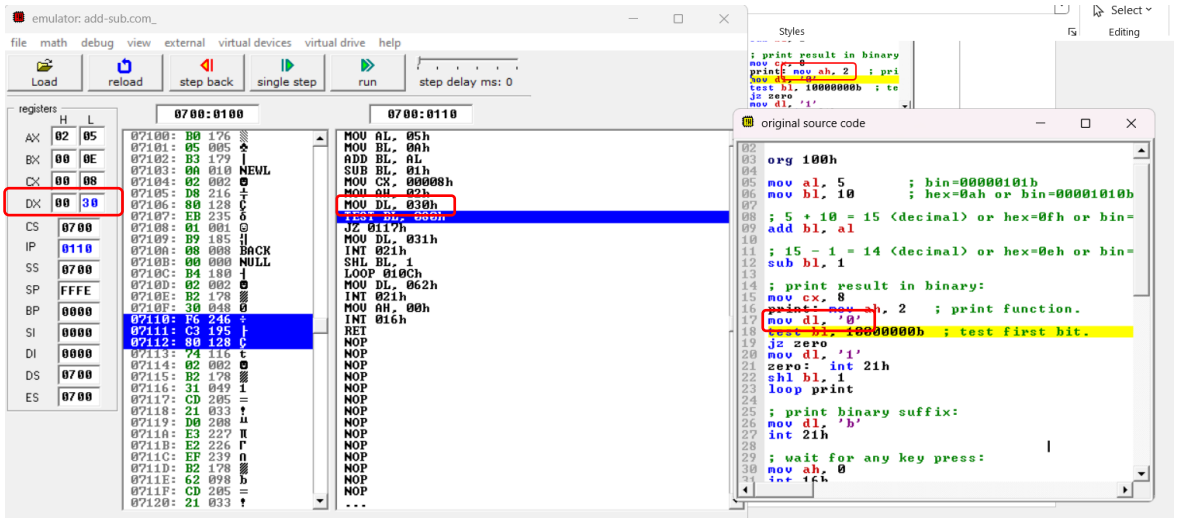
  + Gán thanh ghi CX với giá trị là 8 (do có 8bit) với câu lệnh **mov cx, 8**



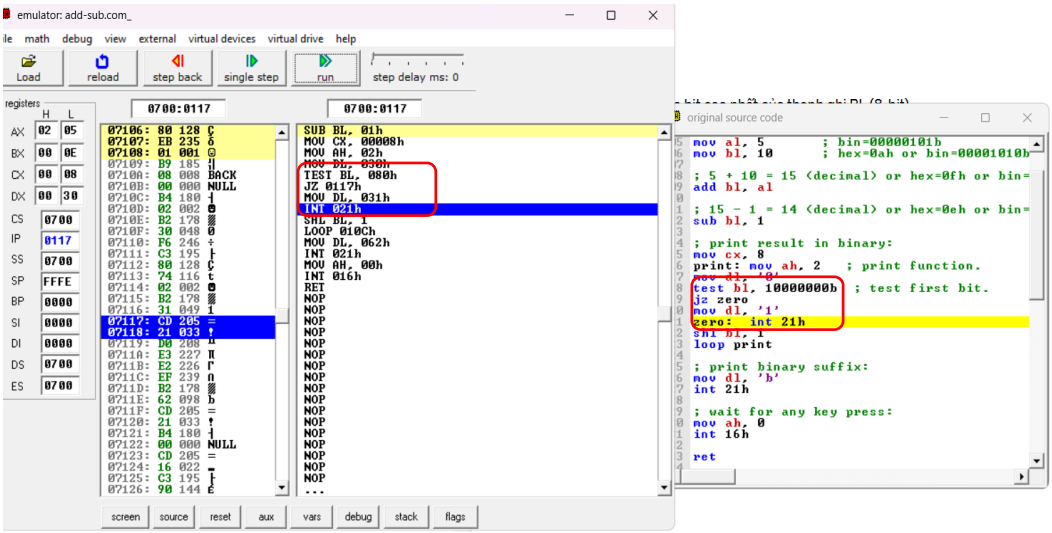
+ Vòng lặp print : gán thanh ghi AH với giá trị là 2, với câu lệnh là **mov ah, 2**



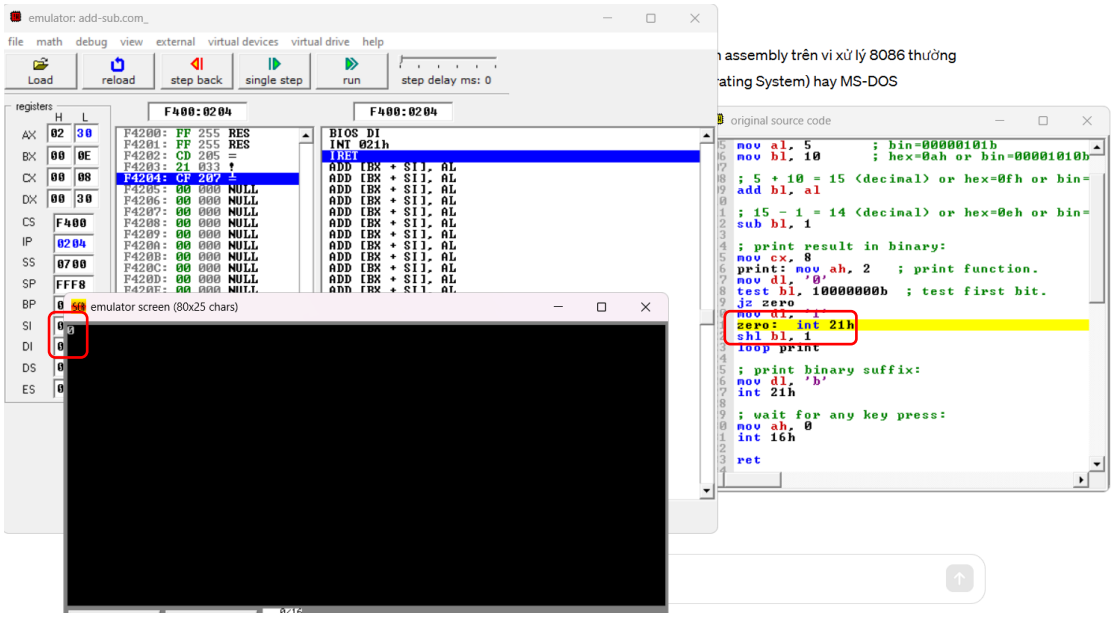
+ Gán ký tự ‘0’ vào thanh ghi DL với câu lệnh **mov dl, ‘0’** . Trong trường hợp này, thanh ghi DL được dùng để lưu trữ ký tự để in ra màn hình. Ký tự ‘0’ là một ký tự ASCII, trong bảng ASCII thì ‘0’ có giá trị là 48 (tương đương với 0x30 trong hệ cơ số 16) nên thanh ghi DL sẽ có giá trị là 0x30.



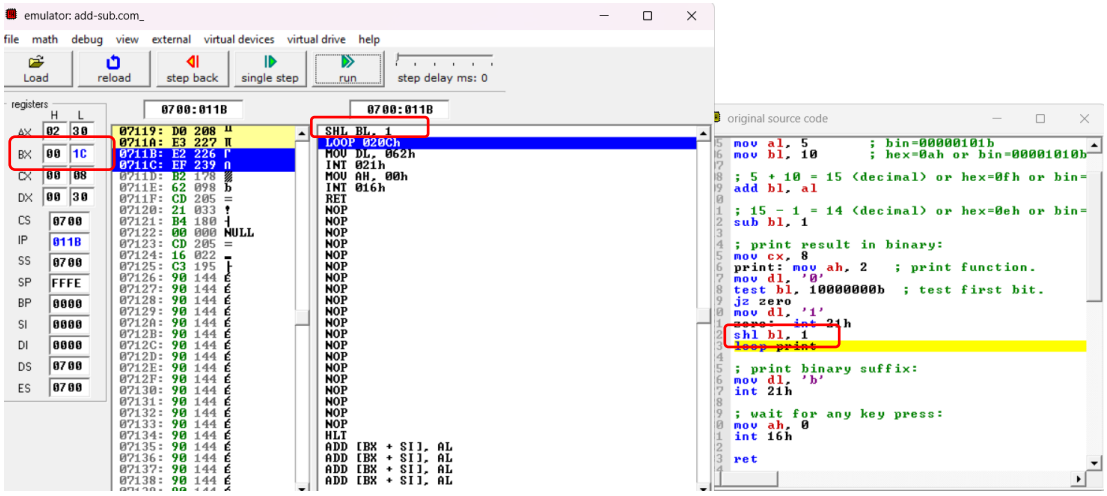
+ Kiểm tra xem bit cao nhất của thanh ghi BL có bằng 1 hay không với câu lệnh : **mov bl, 1000000b**. Số 1000000b tương ứng với 0x80 trong hệ cơ số 16, bit cao nhất của thanh ghi BL sẽ được kiểm tra bằng cách thực hiện phép AND logic giữa giá trị của thanh ghi BL với số 1000000b. Nếu bit cao nhất của BL là 1, kết quả của phép AND sẽ là 1 và cờ zero flag sẽ không được thiết lập (được đặt ở trạng thái "không zero"), và do đó lệnh "jz" sẽ không nhảy tới nhãn được chỉ định. Nếu bit cao nhất của BL là 0, kết quả của phép AND sẽ là 0 và cờ zero flag sẽ được thiết lập (được đặt ở trạng thái "zero"), và lệnh "jz" sẽ nhảy tới nhãn được chỉ định. Trong code mẫu thì bit cao nhất của BL là 0 nên chương trình sẽ nhảy tới nhãn được chỉ định **là zero: int 21h**.



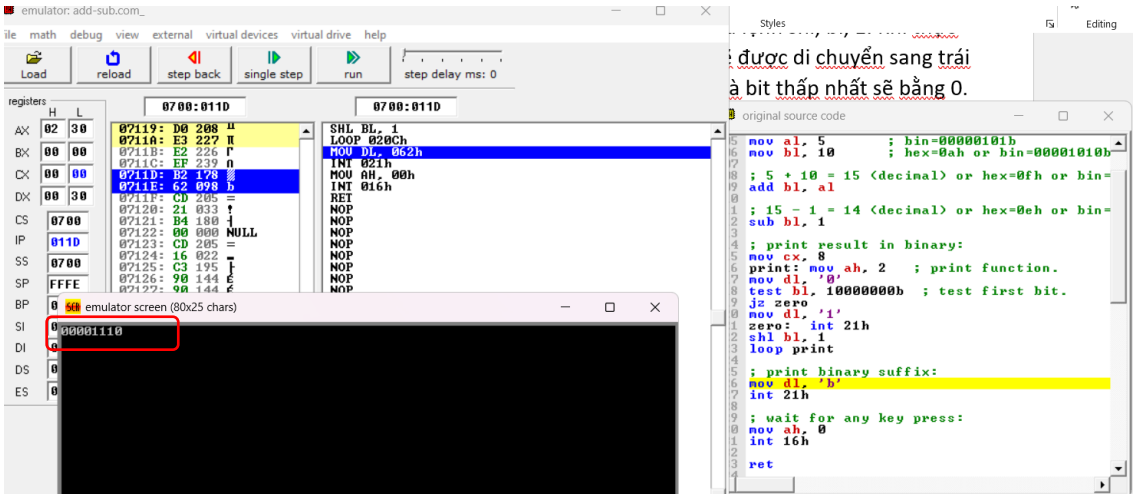
+ Gọi hàm để in ký tự ra màn hình với dòng lệnh **là int 21h**. Sau đó màn hình sẽ in ra kí tự ‘0’.



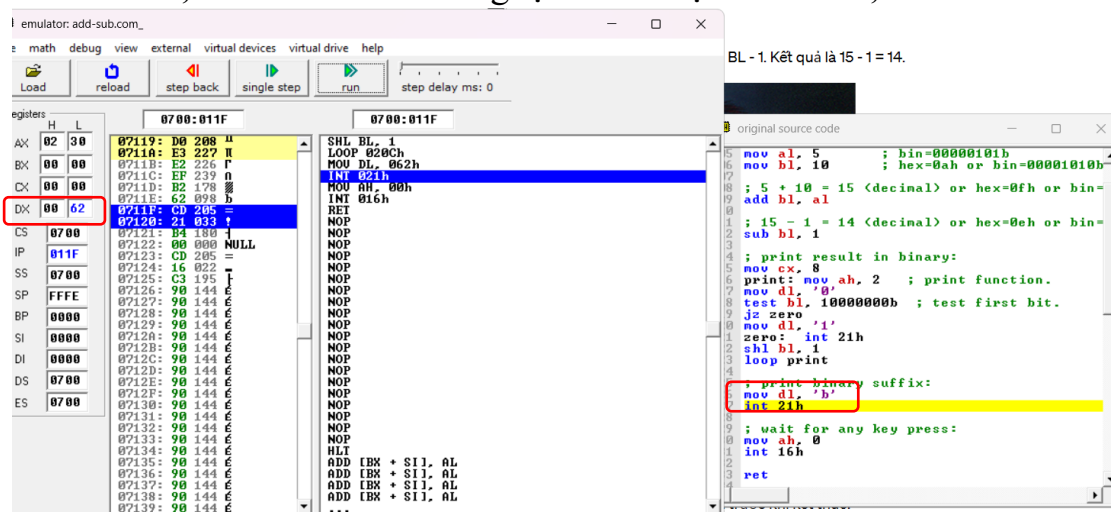
+ Dịch trái thanh ghi BL sang trái một bit với câu lệnh **shl, bl, 1**. Khi thực hiện dịch trái, tất cả các bit trong thanh ghi sẽ được di chuyển sang trái một vị trí, bit cao nhất sẽ được đẩy ra ngoài và bit thấp nhất sẽ bằng 0.



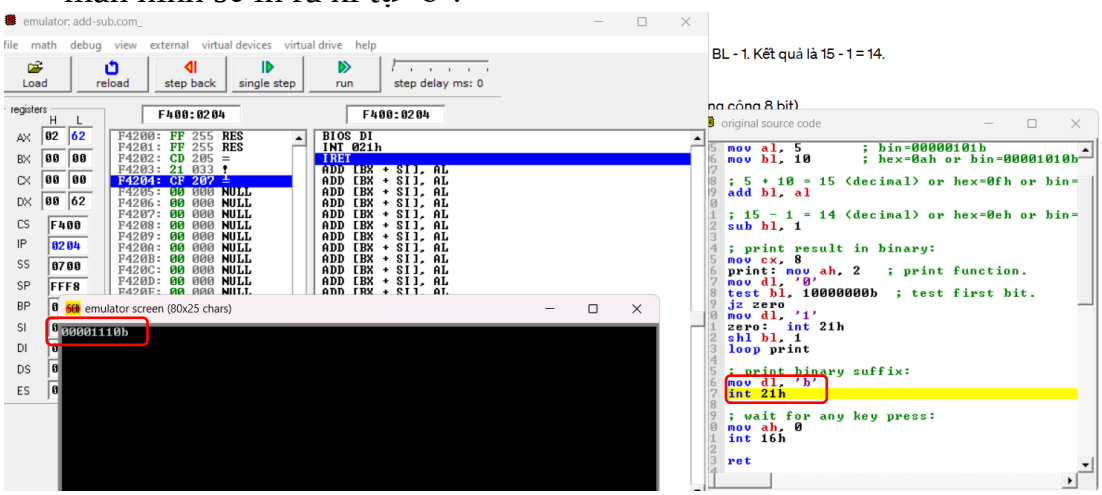
+ Tiếp tục lặp lại vòng lặp thêm 7 lần nữa, ta sẽ in ra màn hình một dãy số.



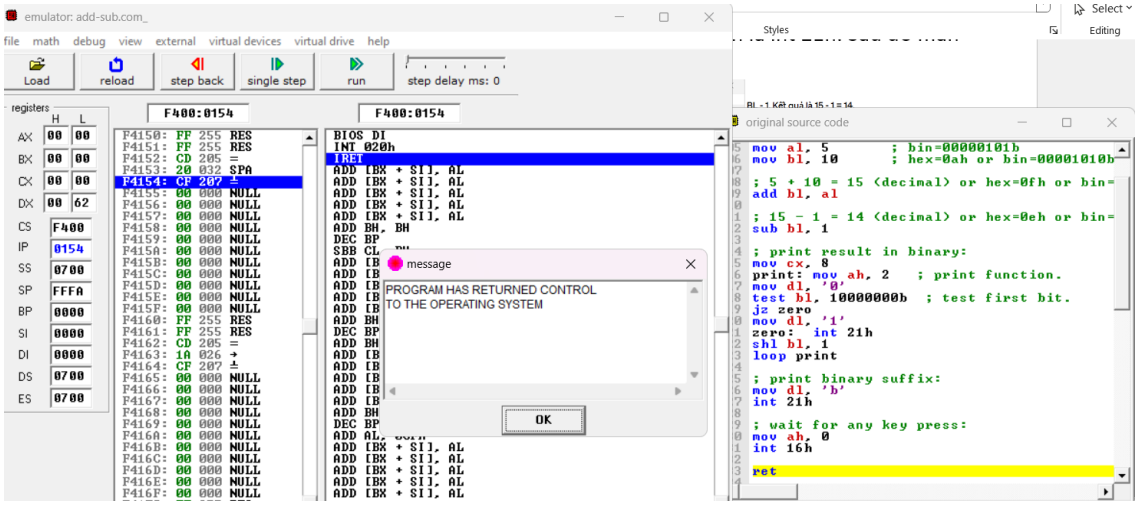
+ In ra ký tự ‘b’ để đánh dấu kết thúc chuỗi số nhị phân với câu lệnh **mov dl, ‘b’**. Giải thích tương tự với câu lệnh **mov dl, ‘0’** .



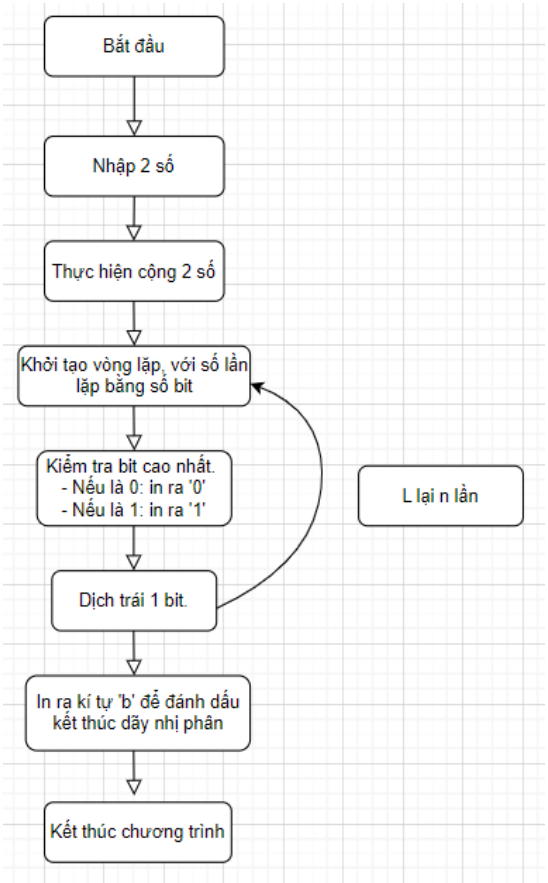
+ Gọi hàm để in ký tự ra màn hình với dòng lệnh là **int 21h**. Sau đó màn hình sẽ in ra kí tự ‘b’.



+ Gán giá trị 0 vào thanh ghi AH với câu lệnh **mov ah, 0**. Sau đó dùng câu lệnh **int 16h** để người dùng nhấn vào một nút bất kì trên màn hình để kết thúc chương trình. Sau đó kết thúc chương trình bằng câu lệnh ret.



2. Lưu đồ giải thuật xử lý chương trình cộng 2 số 32 bit:



3. Thực hiện chương trình trừ 2 số 32 bit:

- Source code:

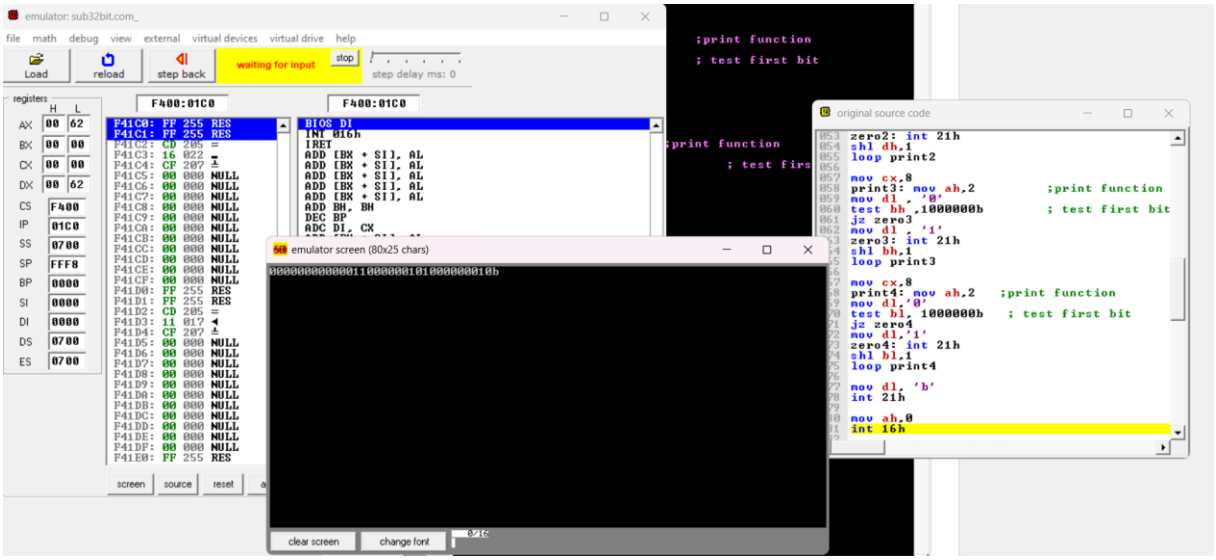
```
name "sub32bit"
org 100h

.code
main proc
    mov ah,1   ; ah=00000001b
    mov al,9   ; al=00001001b
    mov bh,5   ; bh=10000101b
    mov bl,2   ; bl=10000010b
    mov ch,1   ; ch=00000001b
    mov cl,3   ; cl=00000011b
    mov dh,0   ; dh=00000000b
    mov dl,1   ; dl=00000001b

    not ch
    not cl
    not dh
    not dl

    add dl,1
    adc dh,0
    adc cl,0
    adc ch,0

    add bl,dl
```

```
    adc bh,dh
    adc al,cl
    adc ah,ch

    PUSH ax

    POP ax
    mov dh,ah
    PUSH ax

; print result in binary
    mov cx,8

print1:
    mov ah,2      ; print function
    mov dl,'0'
    test dh,1000000b    ; test first bit
    jz zero1
    mov dl, '1'

zero1:
    int 21h
    shl dh,1
    loop print1
    POP ax
    mov dh,al
    mov cx,8
```

```
print2:
     mov ah,2                ; print function
     mov dl, '0'
     test dh, 1000000b       ; test first bit
     jz zero2
     mov dl, '1'

zero2:
     int 21h
     shl dh,1
     loop print2

     mov cx,8

print3:
     mov ah,2          ; print function
     mov dl, '0'
     test bh,1000000b         ; test first bit
     jz zero3
     mov dl, '1'

zero3:
     int 21h
     shl bh,1
     loop print3

     mov cx,8
```

```
print4:
     mov ah,2   ; print function
     mov dl,'0'
     test bl, 1000000b   ; test first bit
     jz zero4
     mov dl,'1'

zero4:
     int 21h
     shl bl,1
     loop print4

     mov dl, 'b'
     int 21h

     mov ah,0
     int 16h

; print binary suffix:
     mov dl, 'b'
     int 21h

; wait for any key press:
     mov ah, 0
     int 16h

main endp
```

- Kết quả:


emulator: sub32bit.com_
file math debug view external virtual devices virtual drive help
Load
reload
step back
waiting for input
stop
step delay ms: 0
registers
AX 00 62
BX 00 00
CX 00 00
DX 00 62
CS F400
IP 01C0
SS 0700
SP FFF8
BP 0000
SI 0000
DI 0000
DS 0700
ES 0700
F400:01C0
F400:01C0
BIOS DI
INT 016h
IRET
ADD [BX + SI], AL
DEC BP
ADC DI, CX
emulator screen (80x25 chars)
0000000000001100000010100000010b
clear screen
change font
;print function
; test first bit
original source code
zero2: int 21h
shl dh,1
loop print2
mov cx,8
print3: mov ah,2 ;print function
mov dl ,'0'
test bh,1000000b ; test first bit
jz zero3
mov dl ,'1'
zero3: int 21h
shl bh,1
loop print3
mov cx,8
print4: mov ah,2 ;print function
mov dl,'0'
test bl, 1000000b ; test first bit
jz zero4
mov dl,'1'
zero4: int 21h
shl bl,1
loop print4
mov dl, 'b'
int 21h
mov ah,0
int 16h