

THỰC HÀNH VI XỬ LÝ – VI ĐIỀU KHIỂN

GVHD: Phạm Minh Quân

Họ và tên sinh viên thực hiện: Nguyễn Đăng Thanh Tuệ

Mã số sinh viên: 22521616

BÁO CÁO THỰC HÀNH SỐ 1.

CỘNG 2 SỐ 32 BIT TRÊN VI XỬ LÝ 8086

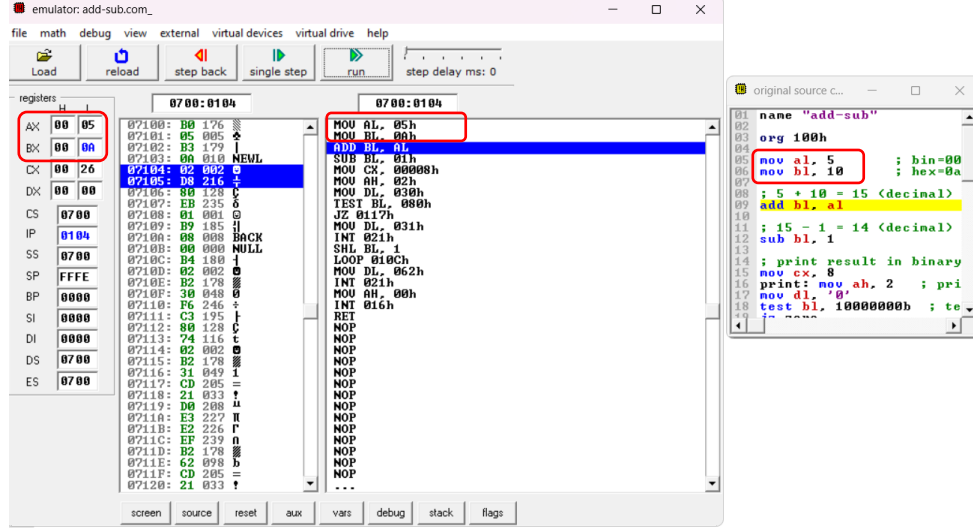
I. CỘNG 2 SỐ 32 BIT TRÊN VI XỬ LÝ 8086

1. Cách thức hoạt động của code mẫu:

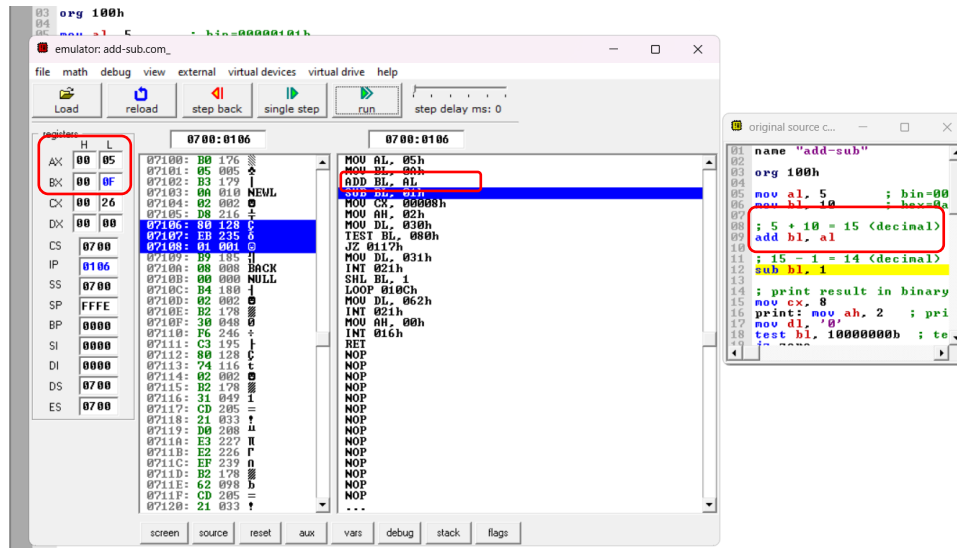
- Dưới đây là đoạn code mẫu cộng 2 số 32 bit trên vi xử lý 8086:

```
edit: C:\emu8086\examples\2_sample.asm
file  edit  bookmarks  assembler  emulator  math  ascii codes  help
new  open  examples  save  compile  emulate  calculator  convertor  options  help  about
01  name "add-sub"
02
03  org 100h
04
05  mov al, 5      ; bin=00000101b
06  mov bl, 10     ; hex=0ah or bin=00001010b
07
08  ; 5 + 10 = 15 (decimal) or hex=0fh or bin=00001111b
09  add bl, al
10
11  ; 15 - 1 = 14 (decimal) or hex=0eh or bin=00001110b
12  sub bl, 1
13
14  ; print result in binary:
15  mov cx, 8
16  print: mov ah, 2  ; print function.
17         mov dl, '0'
18         test bl, 10000000b ; test first bit.
19         jz zero
20         mov dl, '1'
21  zero:  int 21h
22         shl bl, 1
23         loop print
24
25  ; print binary suffix:
26  mov dl, 'b'
27  int 21h
28
29  ; wait for any key press:
30  mov ah, 0
31  int 16h
32
33  ret
34
35
36
```

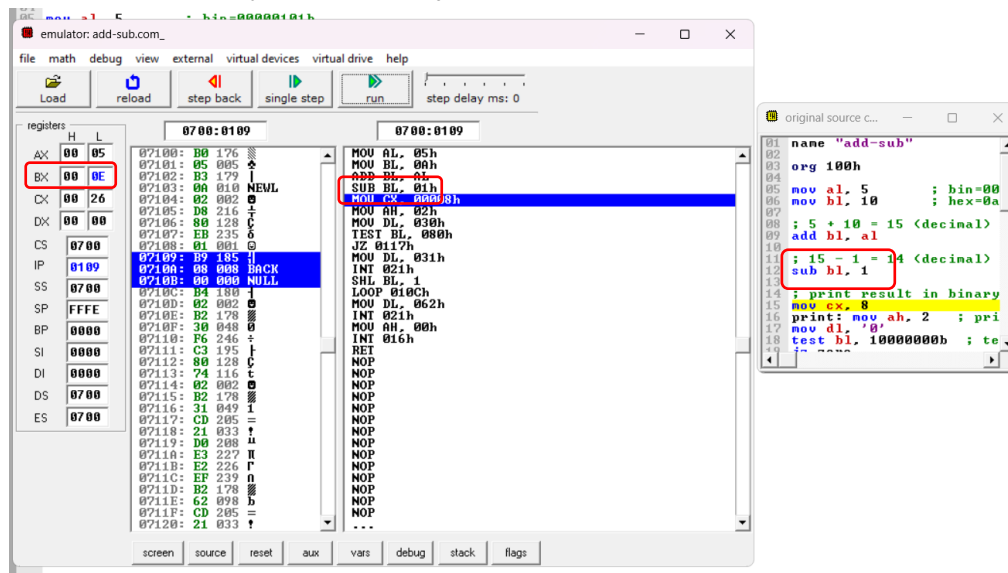
- Mô tả hoạt động của đoạn của mẫu cộng 2 số 32 bit trên vi xử lý 8086:
 - + Bước 1: Khởi tạo giá trị với thanh ghi AL = 0x05, BL = 0x0A với câu lệnh **mov al, 5** và **mov bl, 10**



+ 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 $BL = 0x0F$.

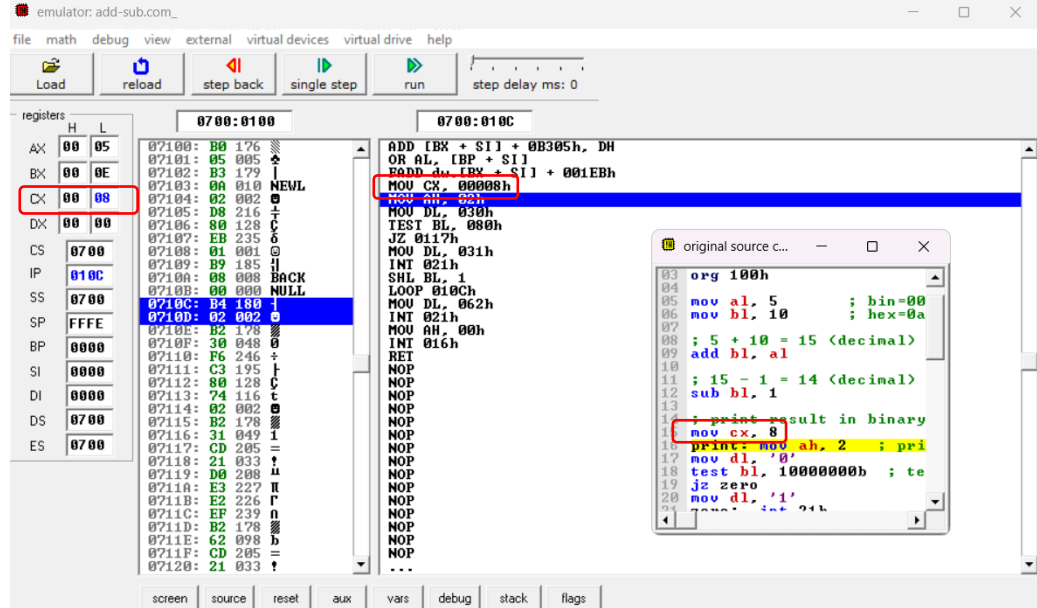


+ Bước 3: Lấy giá trị của thanh ghi BL trừ đi 1, ta có được giá trị mới của thanh ghi $BL = 0x0E$ với câu lệnh là **sub bl, 1**.

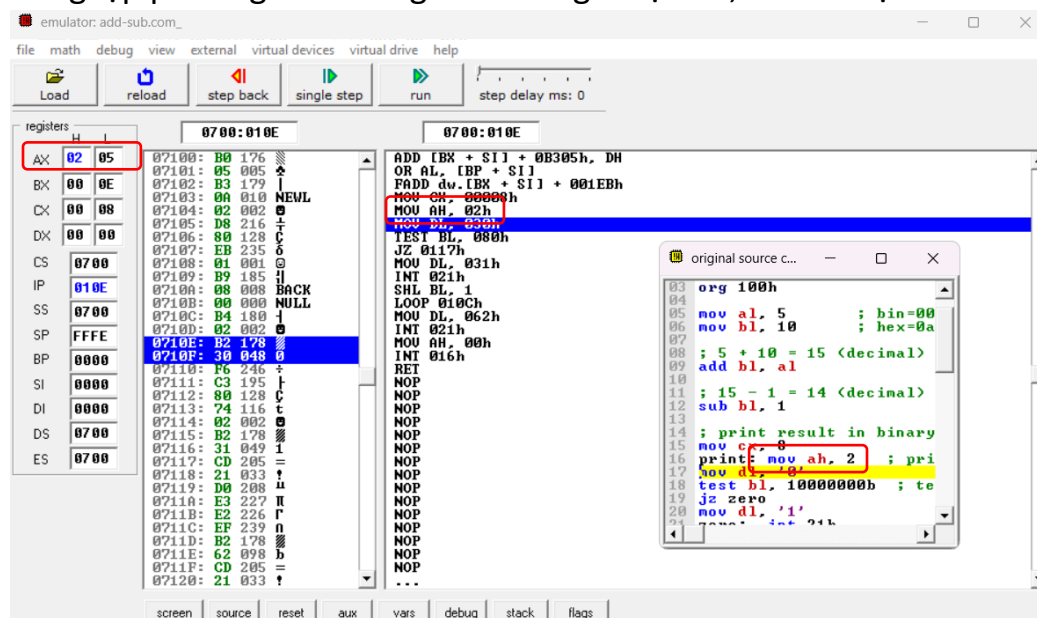


+ Bước 4: In ra kết quả dưới dạng nhị phân. Sử dụng vòng lặp để in ra từng số của thanh ghi BL (theo code mẫu thì có 8bit)

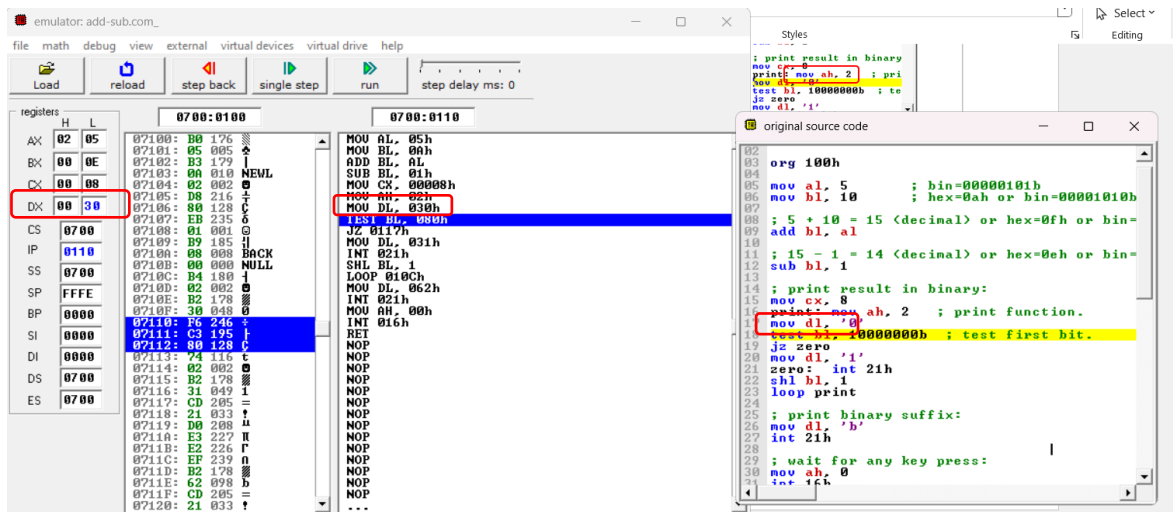
- Gán thanh ghi CX với giá trị là 8 (do có 8bit) với câu lệnh **mov cx, 8**



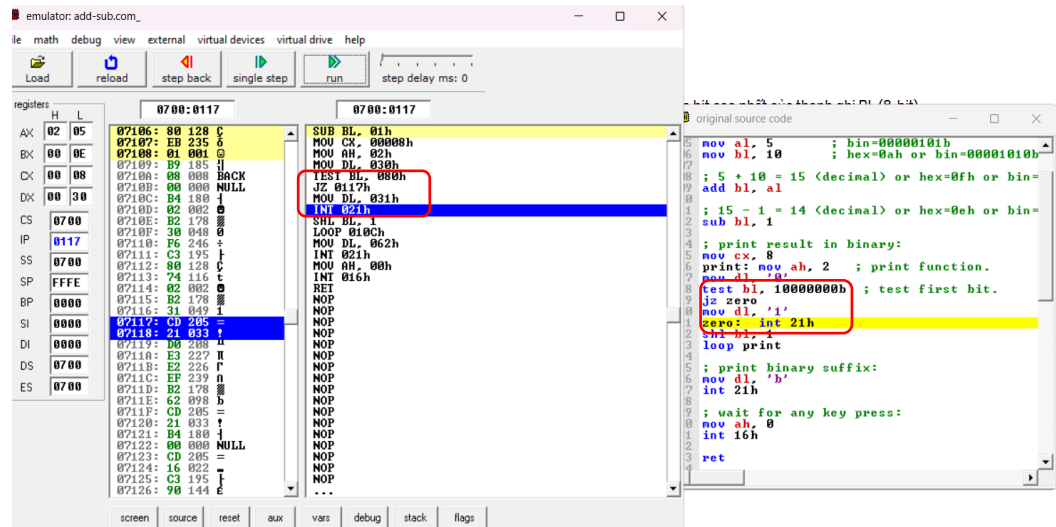
- Vòng lặp print : gán thanh ghi AH với giá trị là 2 , với câu lệnh là **mov ah , 2**



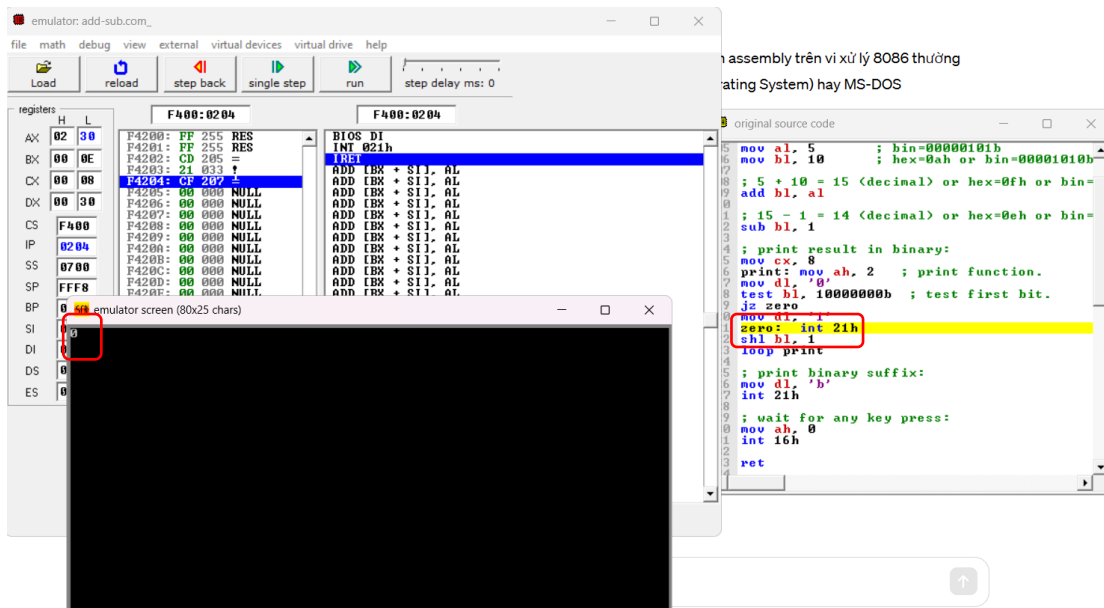
- Gán ký tự '0' vào thanh ghi DL với câu lệnh **mov dl , '0'** . Trong trường hợp này, thanh ghi DL được dùng để lưu trữ ký tự để in ra màn hình. Ký tự '0' là một ký tự ASCII, trong bảng ASCII thì '0' có giá trị là 48 (tương đương với 0x30 trong hệ cơ số 16) nên thanh ghi DL sẽ có giá trị là 0x30.



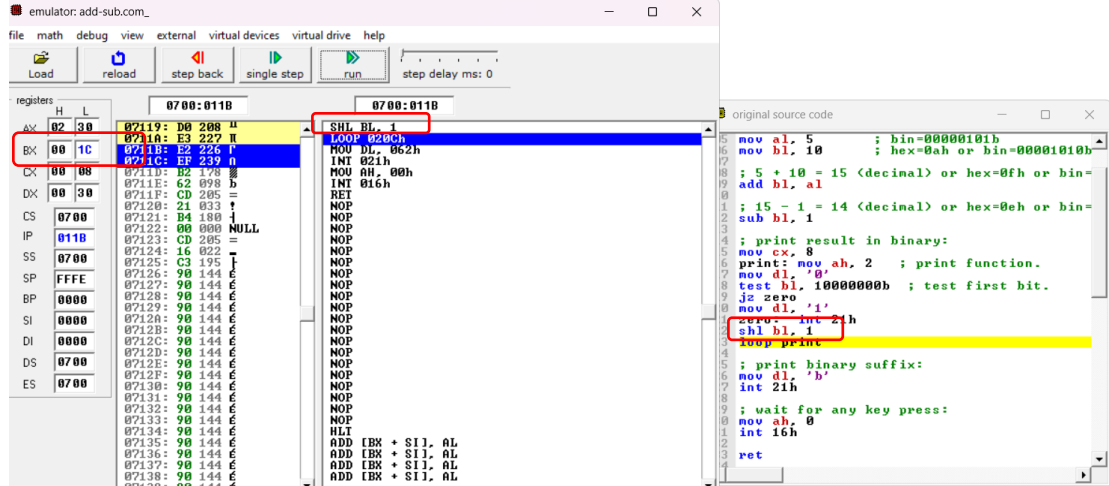
- Kiểm tra xem bit cao nhất của thanh ghi BL có bằng 1 hay không với câu lệnh : **mov bl, 1000000b**. Số 1000000b tương ứng với 0x80 trong hệ cơ số 16, bit cao nhất của thanh ghi BL sẽ được kiểm tra bằng cách thực hiện phép AND logic giữa giá trị của thanh ghi BL với số 1000000b. Nếu bit cao nhất của 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**.



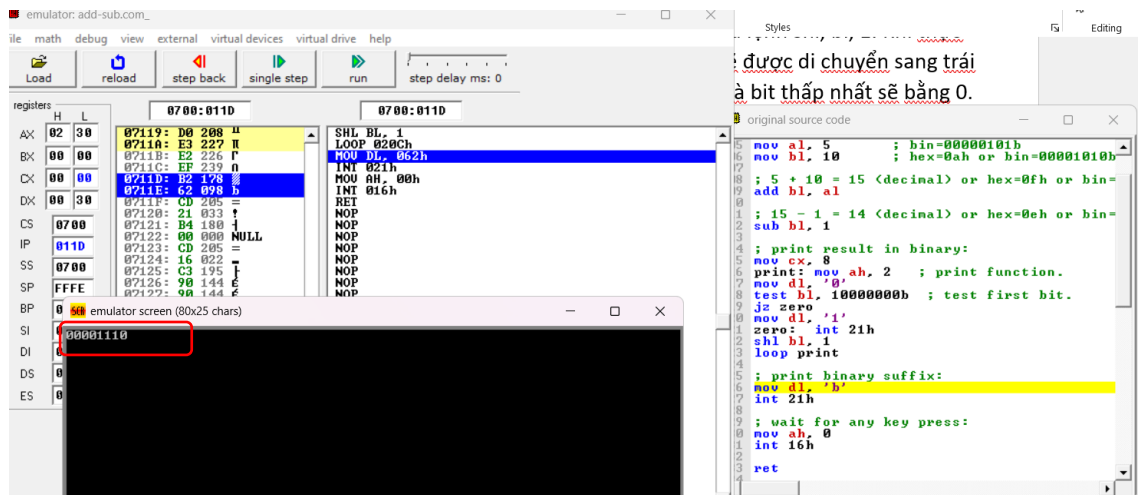
- Gọi hàm để in ký tự ra màn hình với dòng lệnh là **int 21h**. Sau đó màn hình sẽ in ra ký tự '0'.



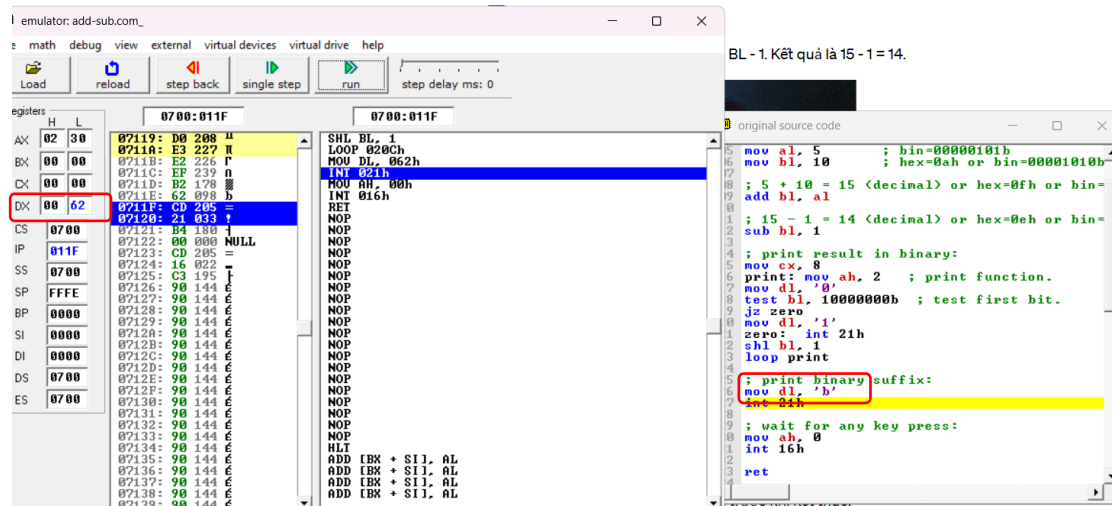
- Dịch trái thanh ghi BL sang trái một bit với câu lệnh **shl, bl, 1**. Khi thực hiện dịch trái, tất cả các bit trong thanh ghi sẽ được di chuyển sang trái một vị trí, bit cao nhất sẽ được đẩy ra ngoài và bit thấp nhất sẽ bằng 0.



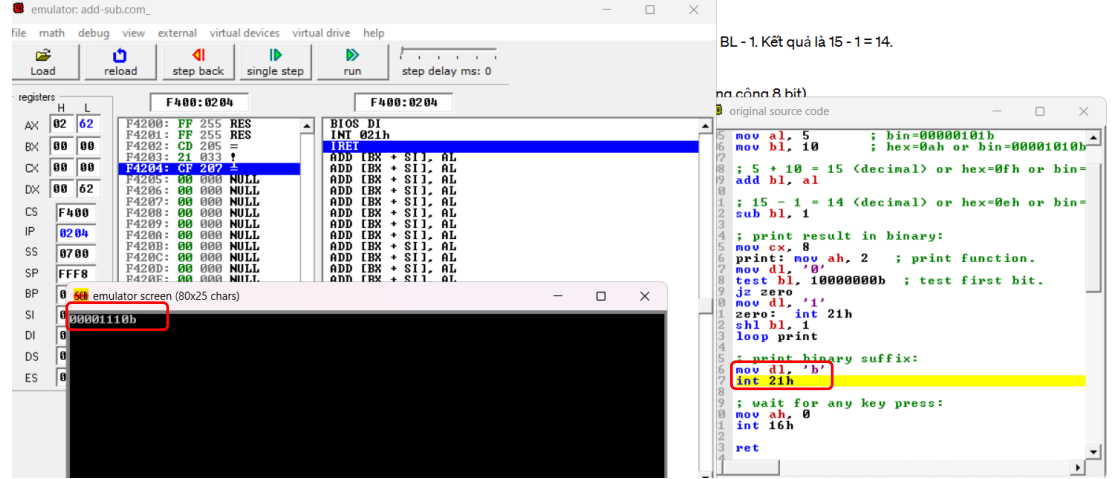
- Tiếp tục lặp lại vòng lặp thêm 7 lần nữa, ta sẽ in ra màn hình một dãy số.



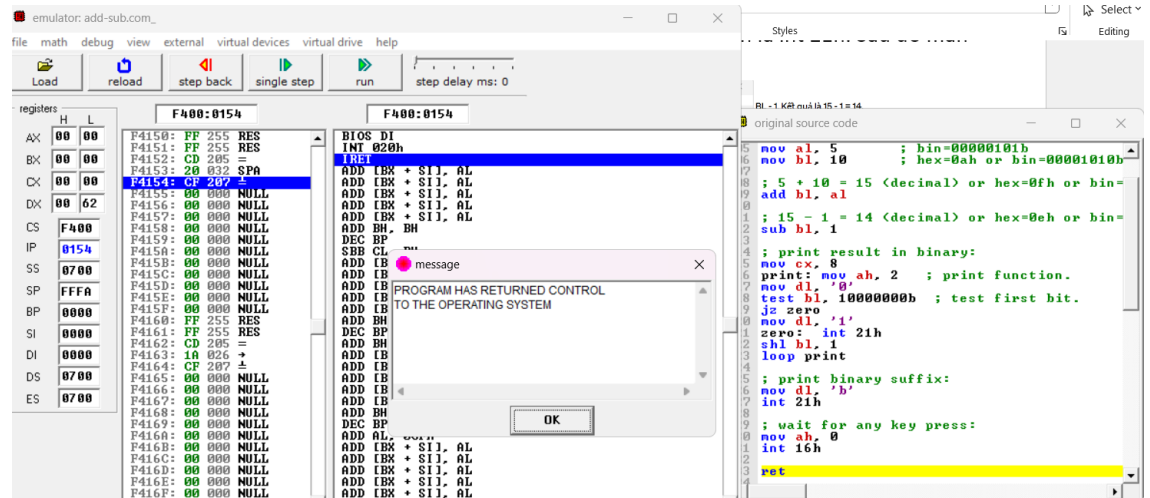
- In ra ký tự 'b' để đánh dấu kết thúc chuỗi số nhị phân với câu lệnh `mov dl, 'b'`. Giải thích tương tự với câu lệnh `mov dl, '0'`.



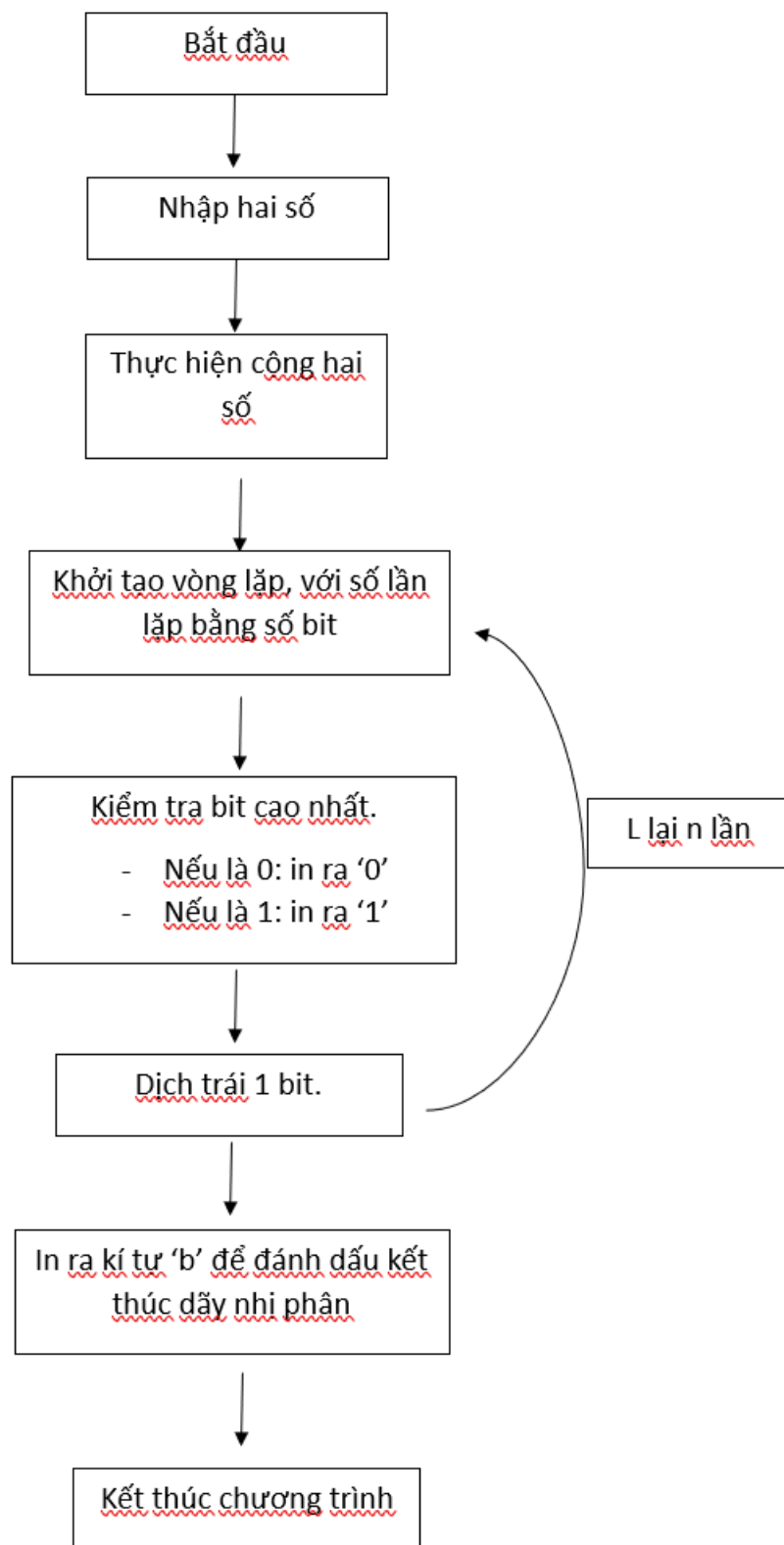
- Gọi hàm để in ký tự ra màn hình với dòng lệnh là `int 21h`. Sau đó màn hình sẽ in ra ký tự 'b'.



- 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**.

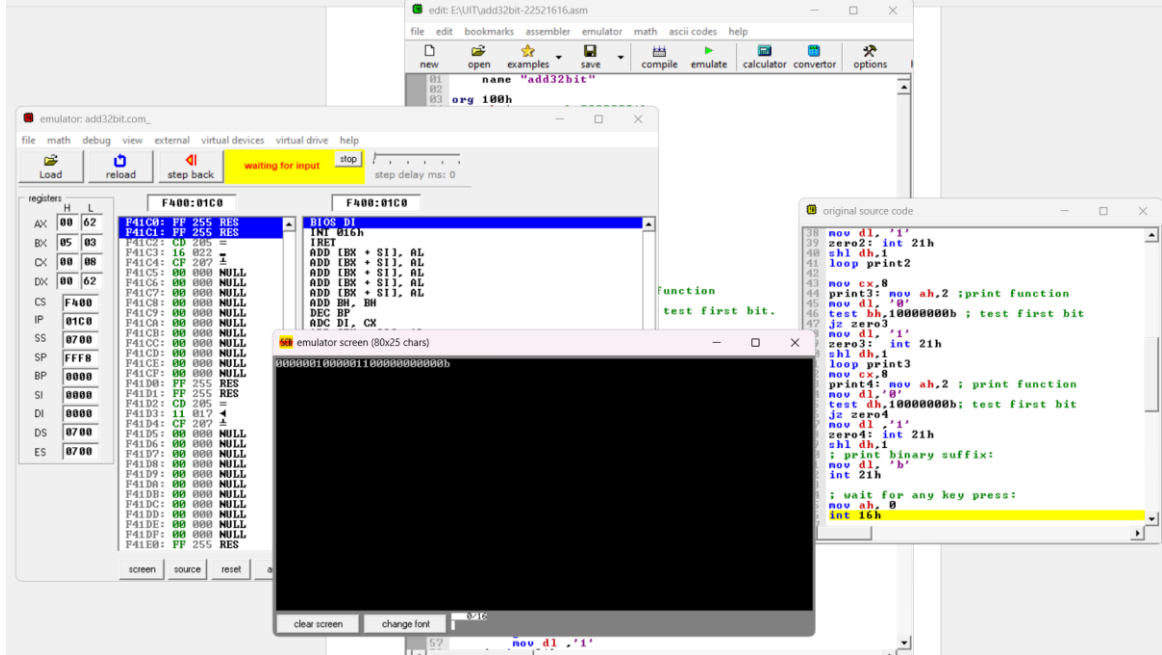


2. Lưu đồ giải thuật xử lý chương trình cộng 2 số 32 bit.



3. Chương trình cộng hai số 32 bit trên vi xử lý 8086

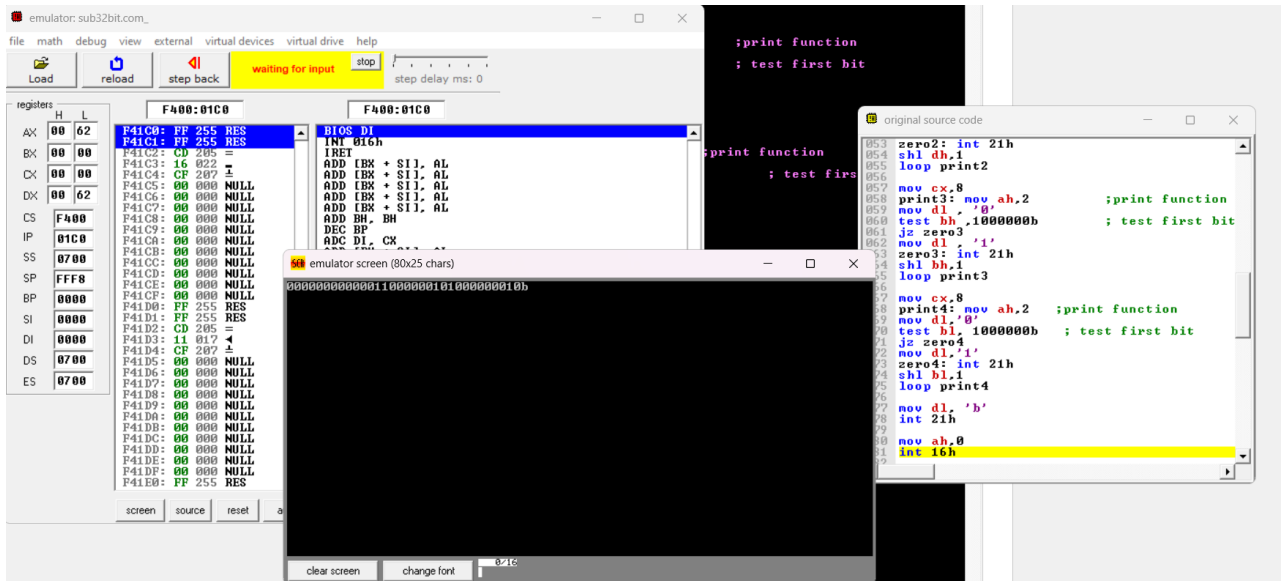
- Chú thích : Source code sẽ được đính kèm trong bài nộp.
- Hình ảnh minh chứng rằng đoạn code hoạt động :



II. Trừ hai số 32 bit trên vi xử lý 8086.

1. Chương trình cộng hai số 32 bit trên vi xử lý 8086

- Chú thích : Source code sẽ được đính kèm trong bài nộp.
- Hình ảnh minh chứng rằng đoạn code hoạt động :



2. Lưu đồ giải thuật của chương trình trừ hai số 32 bit trên vi xử lý 8086

