**THỰC HÀNH VI XỬ LÝ – VI ĐIỀU KHIỂN**

**GVHD: Phạm Minh Quân**

**Họ và tên sinh viên thực hiện: Đoàn Vũ Phú Minh**

**Mã số sinh viên: 22520859**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

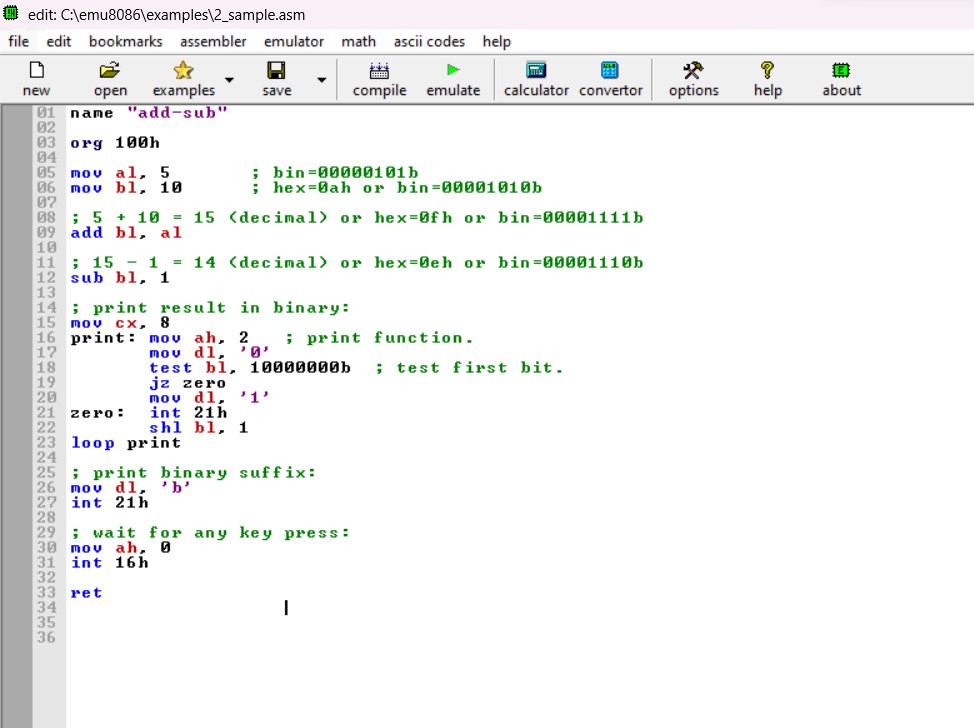
**BÁO CÁO THỰC HÀNH SỐ 1.**

**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:**

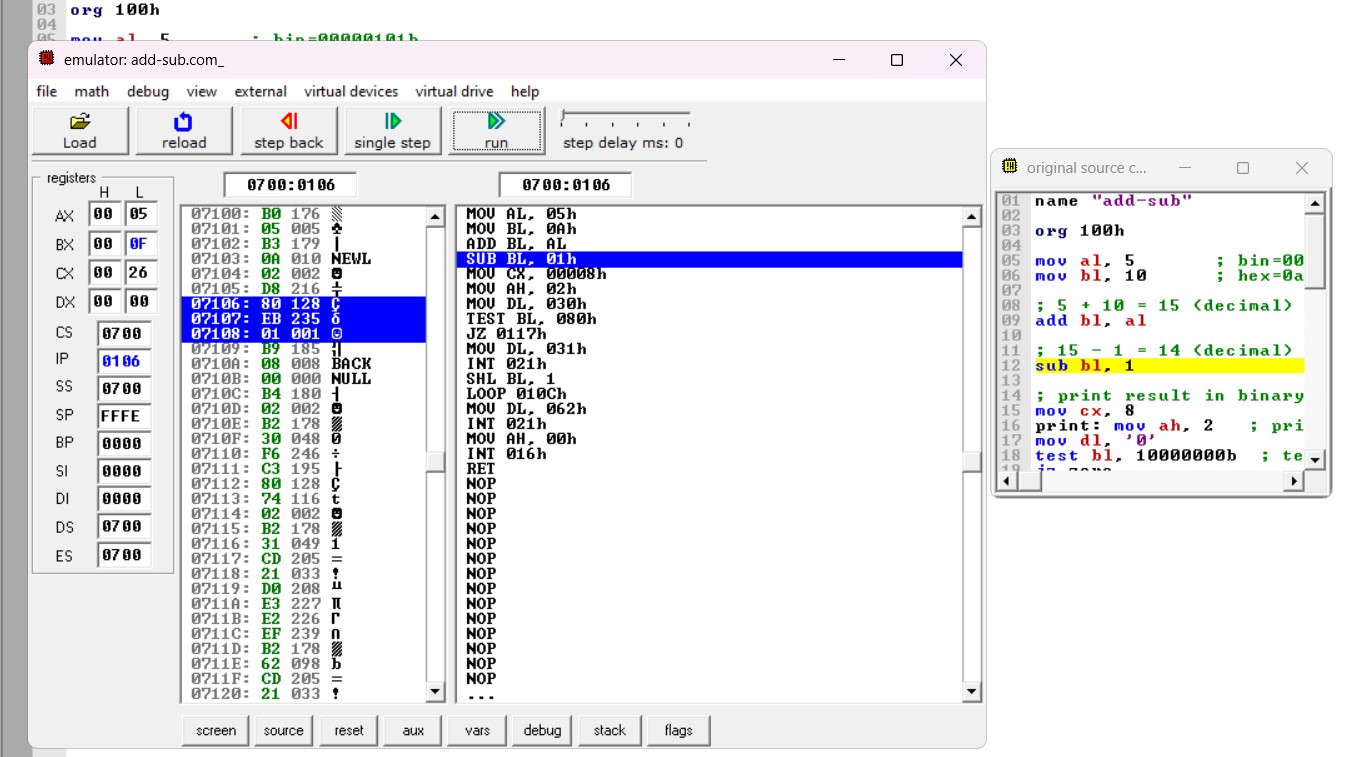
* Dưới đây là đoạn code mẫu cộng 2 số 32 bit trên vi xử lý 8086:



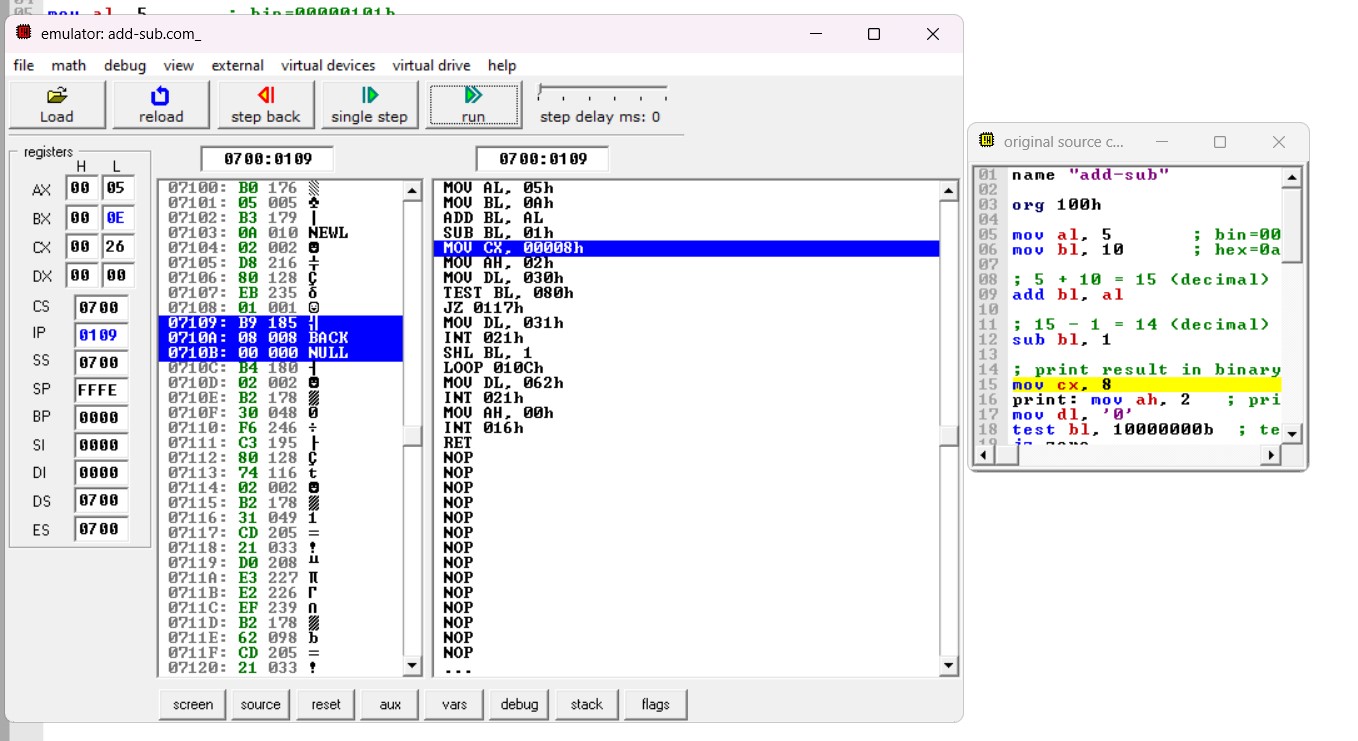
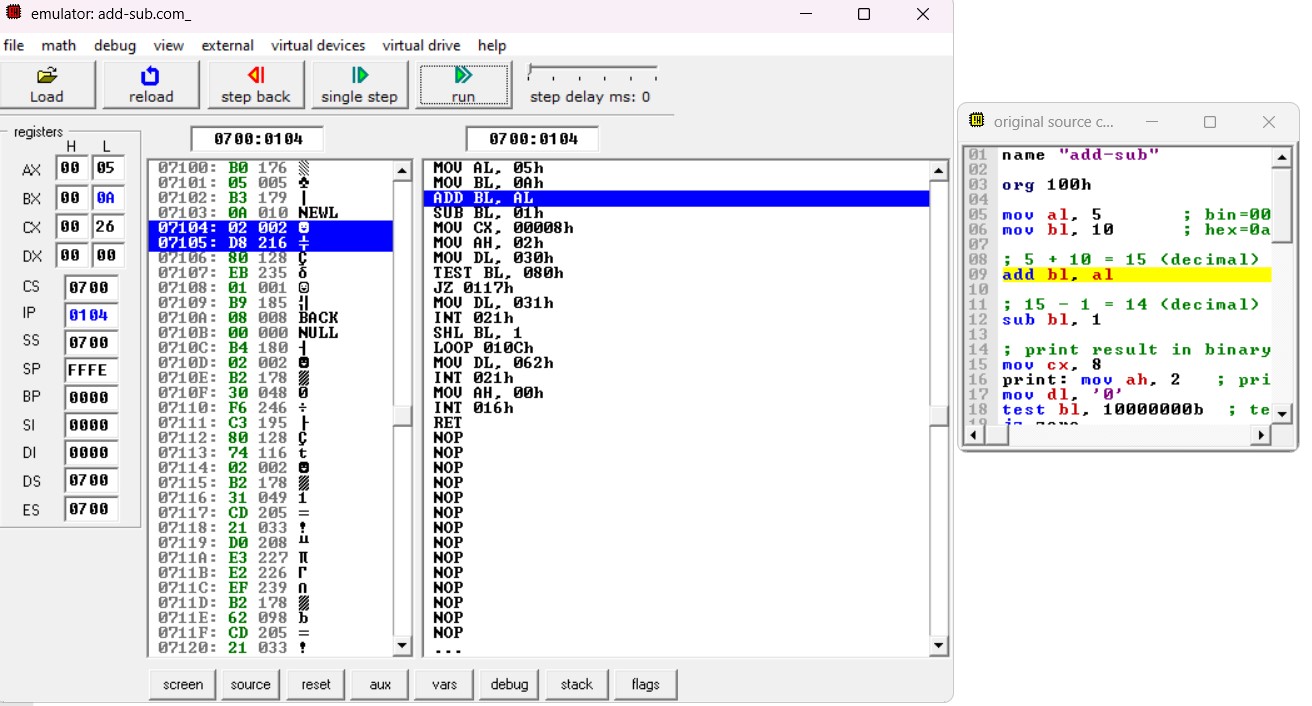
* Mô tả hoạt động của đoạn của mẫu cộng 2 số 32 bit trên vi xử lý 8086:

+ 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.

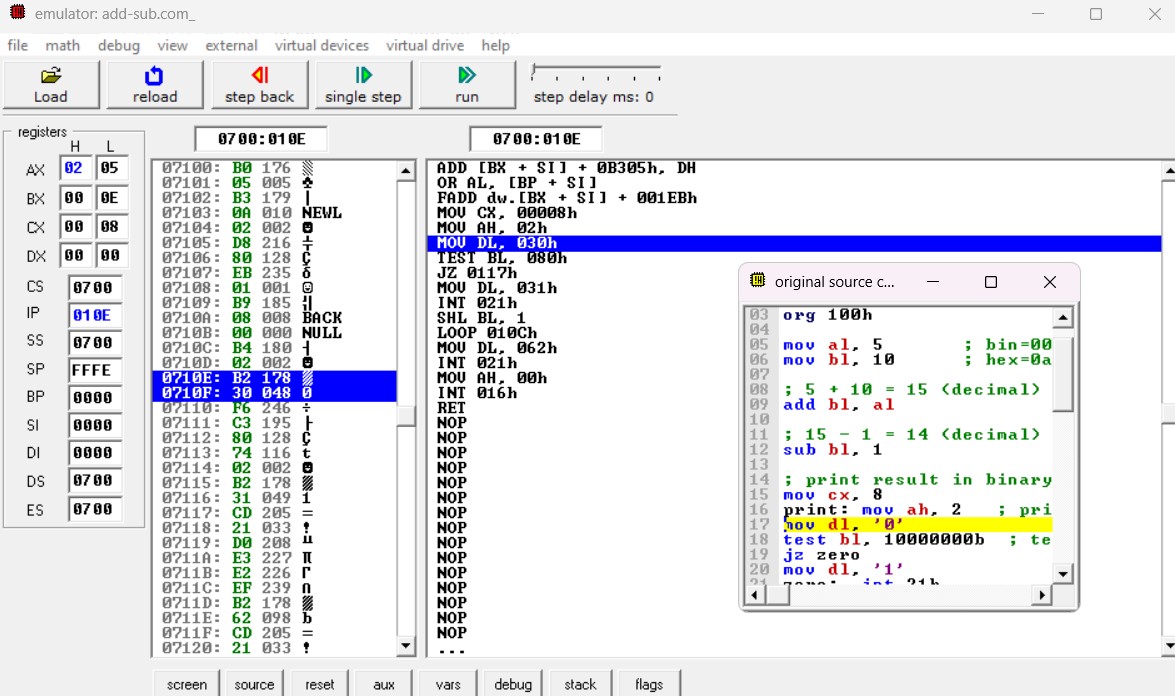
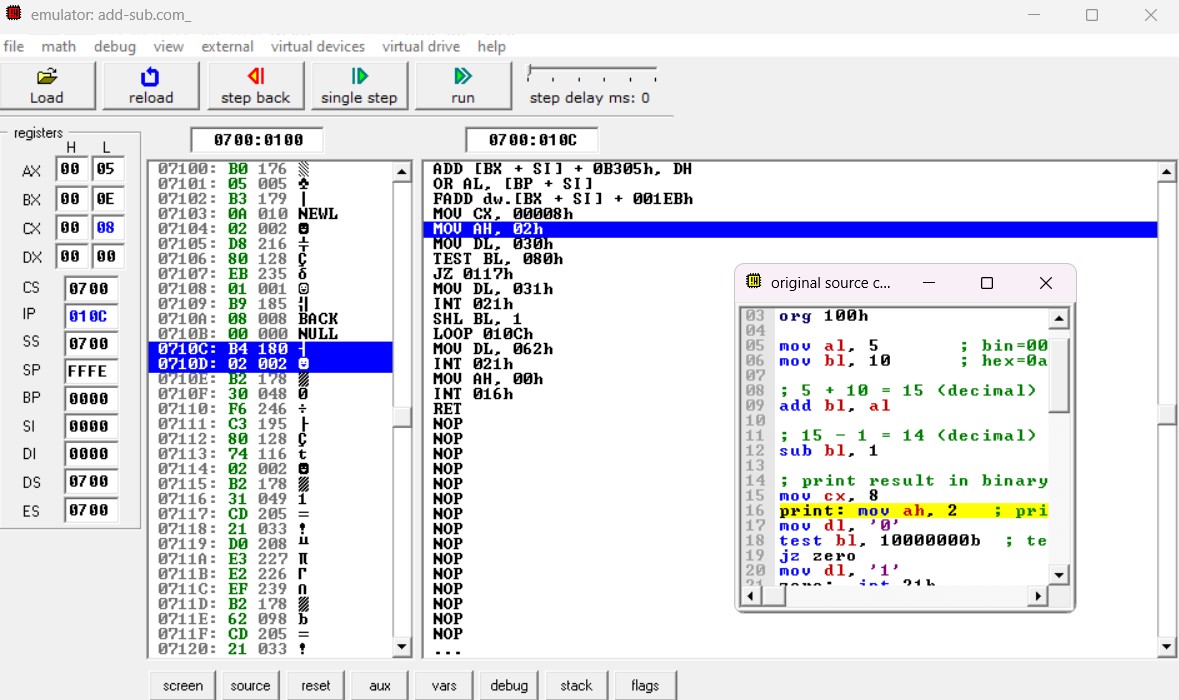


-+ Bước 3: Lấy giá trị của thanh ghi BL trừ đi 1 , ta có được giá trị mới của thanh ghi BL = 0x0E với câu lệnh là **sub bl, 1.**

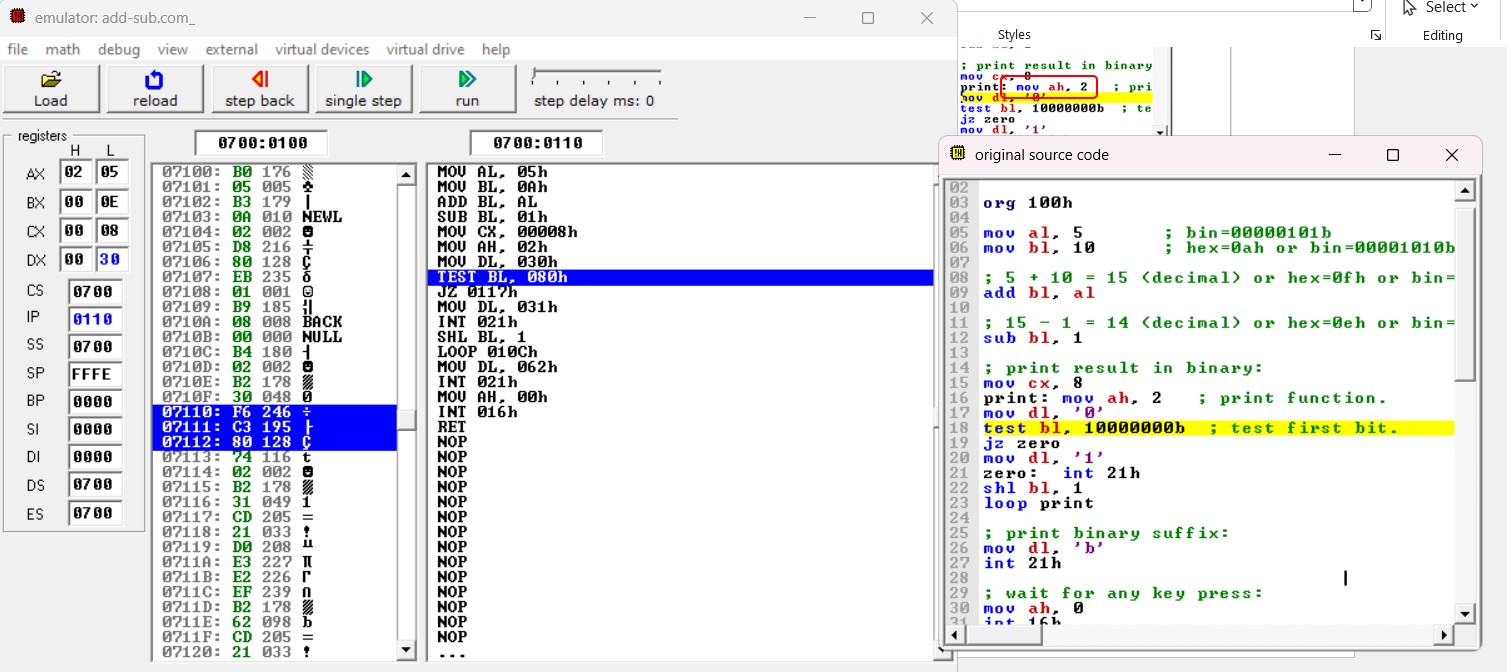


+ Bước 4: In ra kết quả dưới dạng nhị phân. Sử dụng vòng lặp để in ra từng số của thanh ghi BL ( theo code mẫu thì có 8bit)

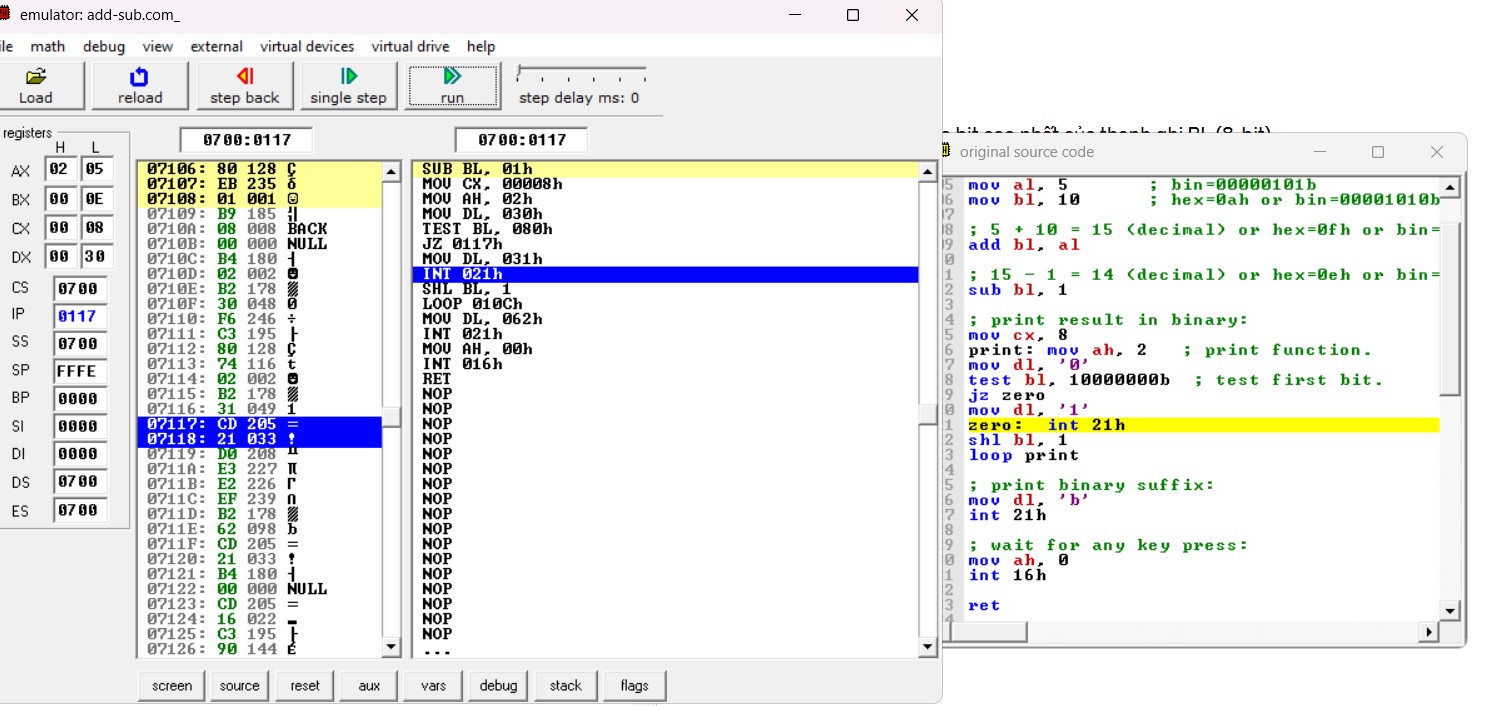
* Gán thanh ghi CX với giá trị là 8 ( do có 8bit) với câu lệnh **mov cx, 8**
* Vòng lặp print : gán thanh ghi AH với giá trị là 2 , với câu lệnh là **mov ah , 2**



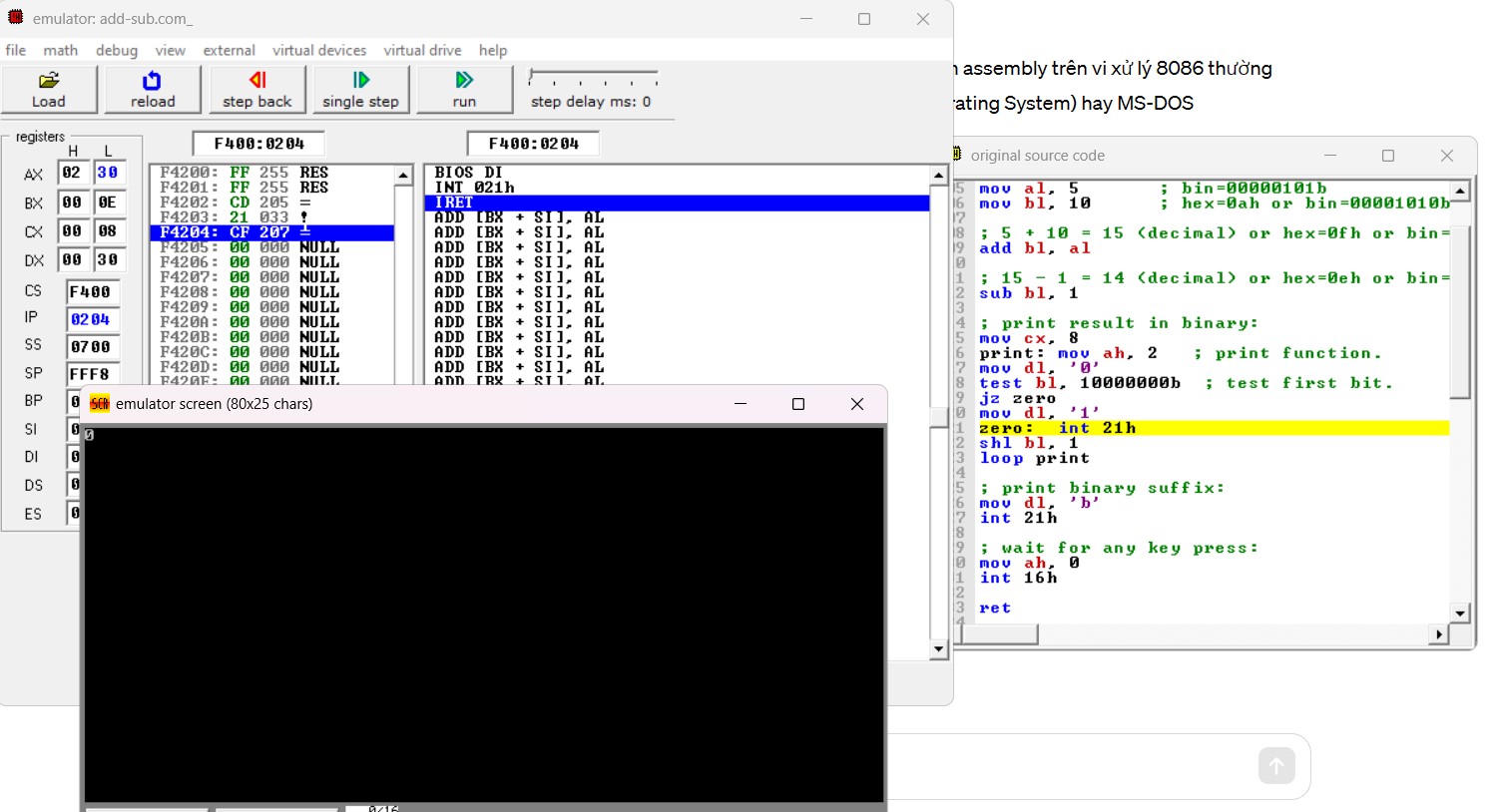
* Gán ký tự ‘0’ vào thanh ghi DL với câu lệnh **mov dl , ‘0’** . Trong trường hợp này, thanh ghi DL được dùng để lưu trữ ký tự để in ra màn hình. Ký tự ‘0’ là một ký tự ASCII, trong bảng ASCII thì ‘0’ có giá trị là 48 ( tương đương với 0x30 trong hệ cơ số 16) nên thanh ghi DL sẽ có giá trị là 0x30.



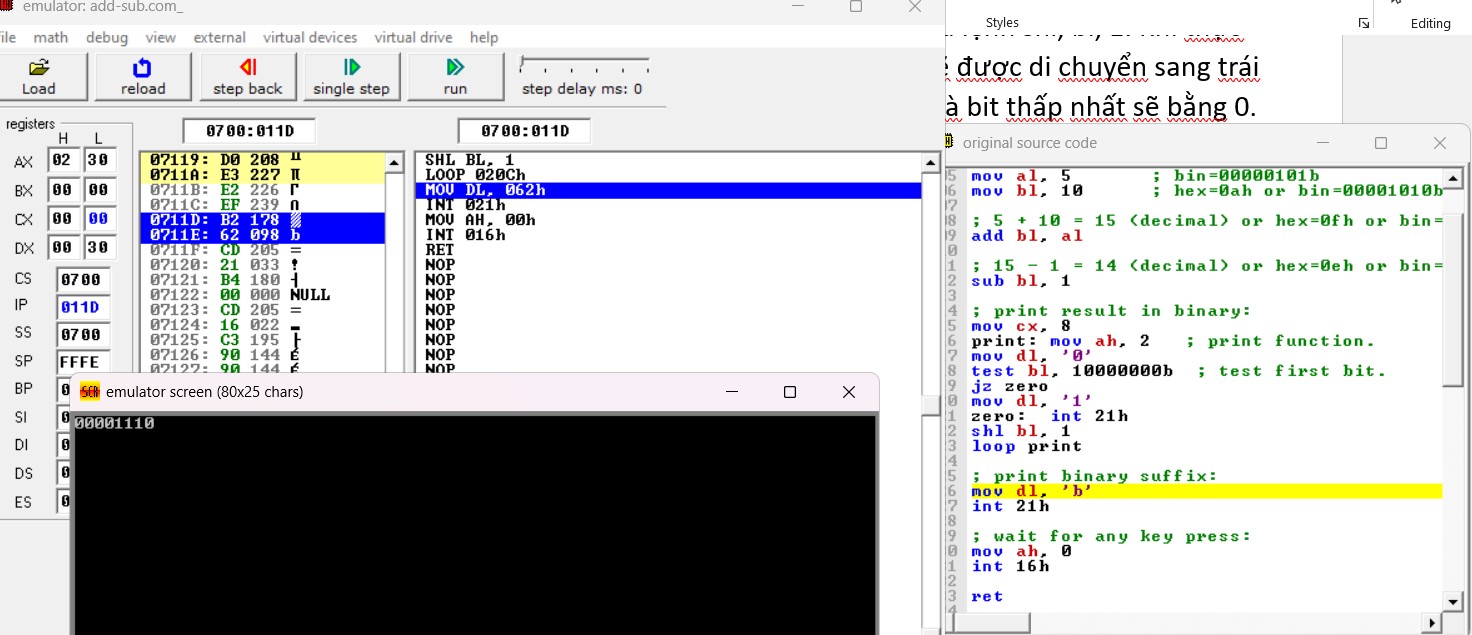
* Kiểm tra xem bit cao nhất của thanh ghi BL có bằng 1 hay không với câu lệnh : **mov bl, 1000000b**. Số 1000000b tương ứng với 0x80 trong hệ cơ số 16, bit cao nhất của thanh ghi BL sẽ được kiểm tra bằng cách thực hiện phép AND logic giữa giá trị của thanh ghi BL với số 1000000b. Nếu bit cao nhất của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



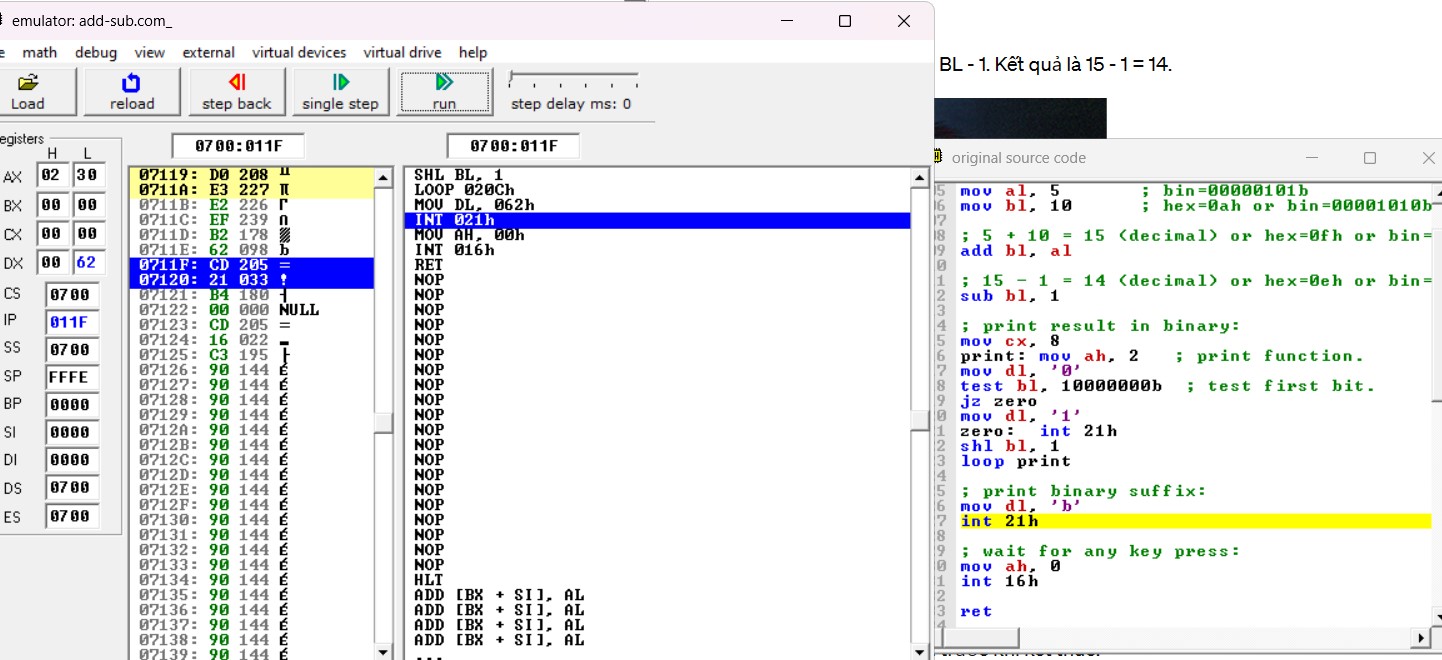
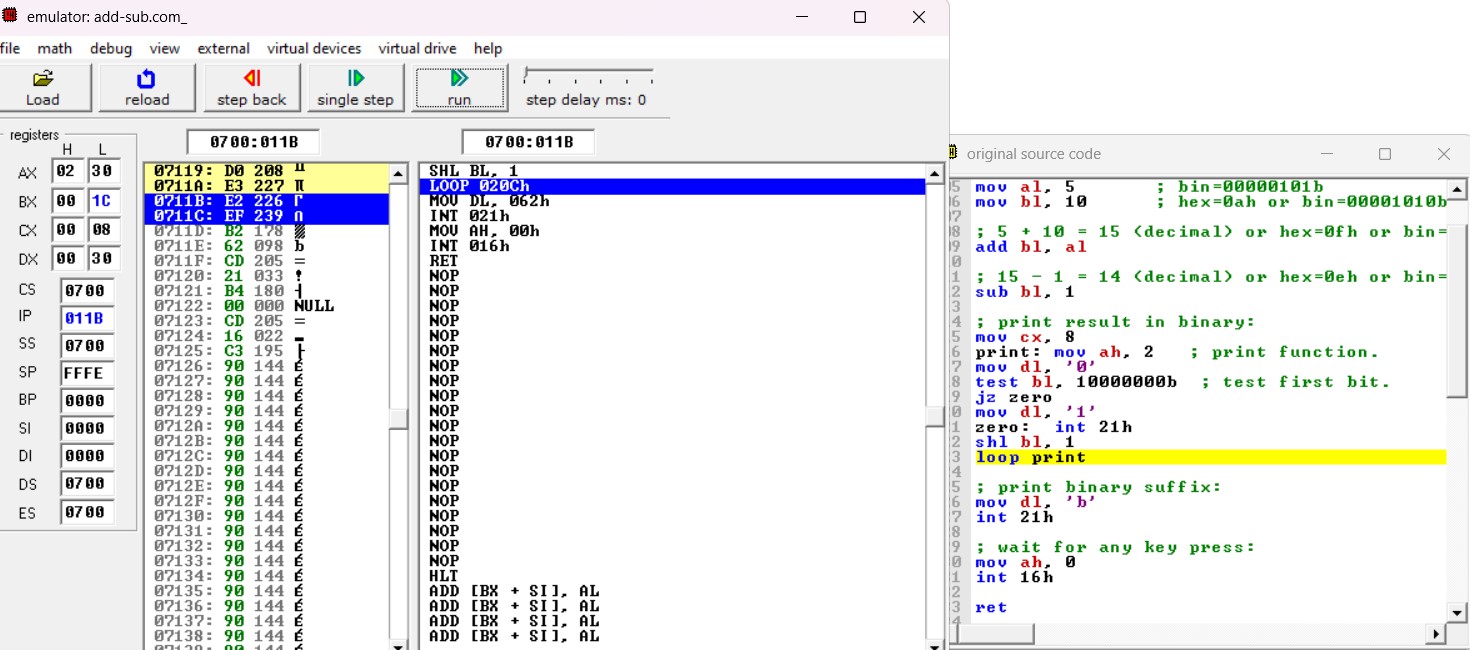
* Gọi hàm để in ký tự ra màn hình với dòng lệnh **là int 21h**. Sau đó màn hình sẽ in ra kí tự ‘0’.



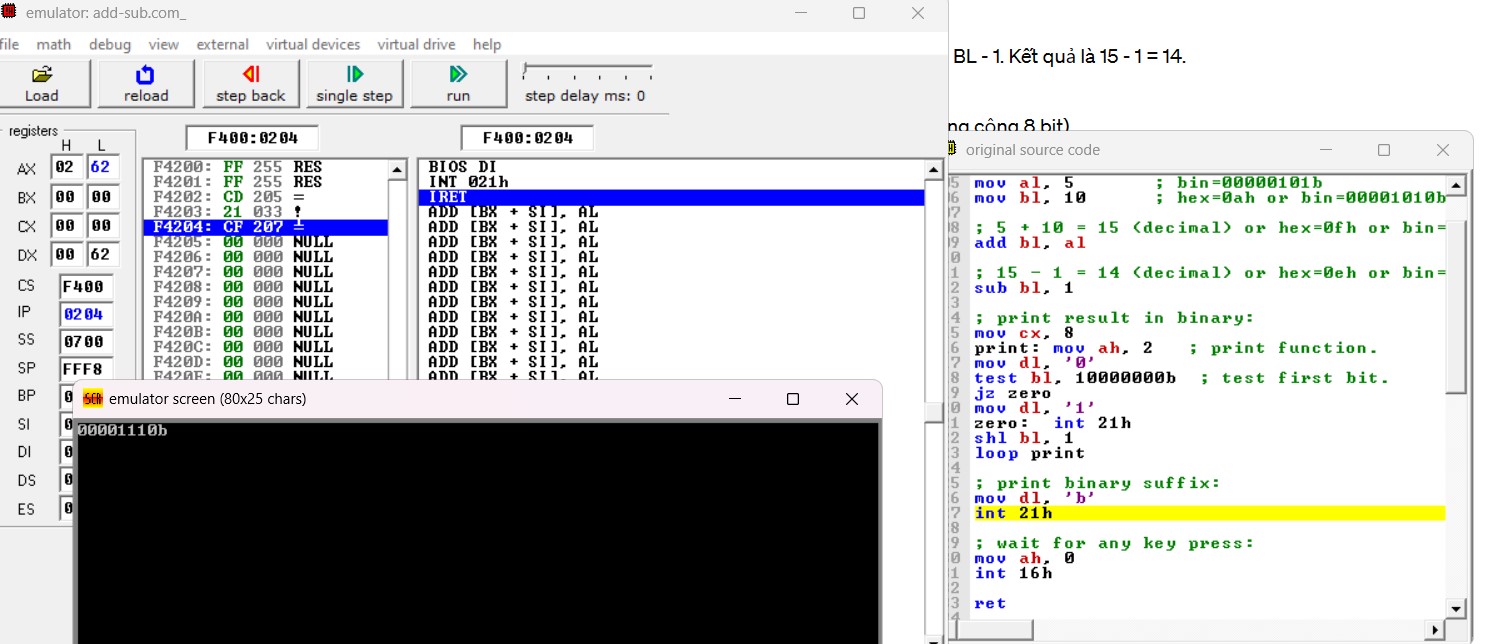
* Dịch trái thanh ghi BL sang trái một bit với câu lệnh **shl, bl, 1**. Khi thực hiện dịch trái, tất cả các bit trong thanh ghi sẽ được di chuyển sang trái một vị trí, bit cao nhất sẽ được đẩy ra ngoài và bit thấp nhất sẽ bằng 0.
* Tiếp tục lặp lại vòng lặp thêm 7 lần nữa, ta sẽ in ra màn hình một dãy số.

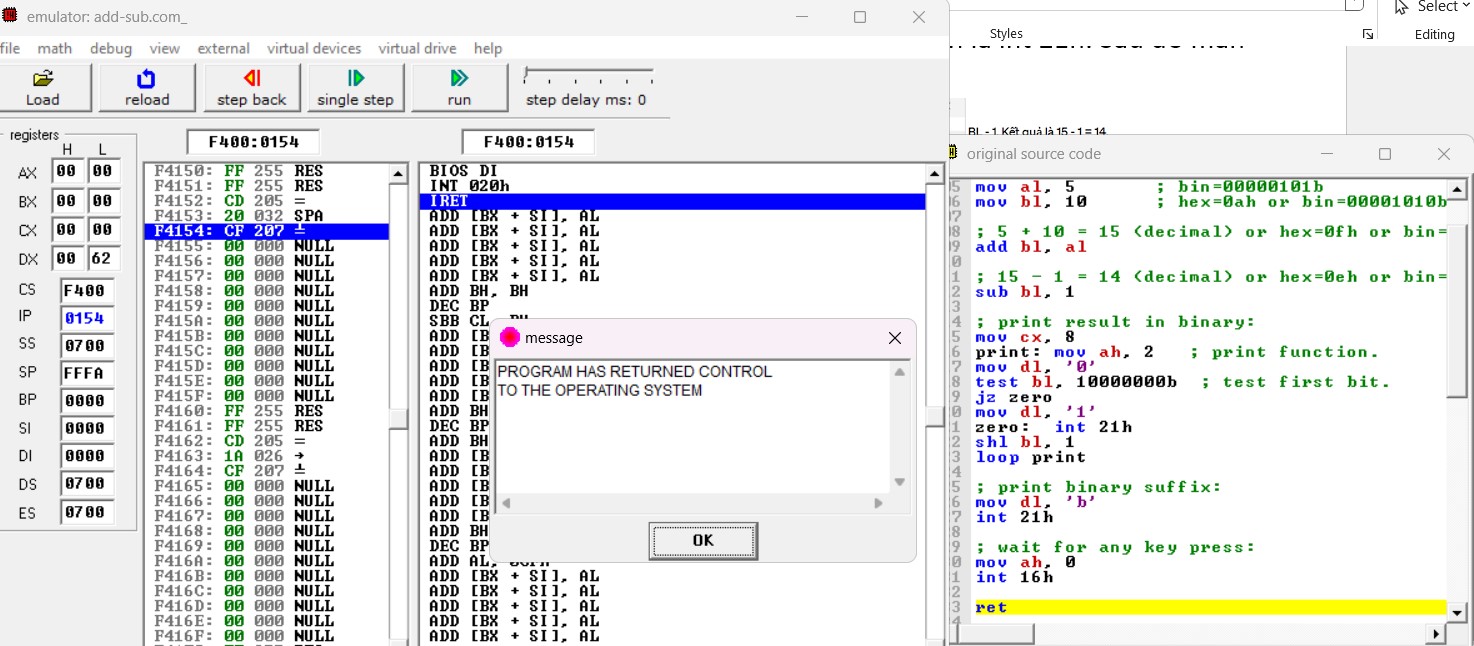


* In ra ký tự ‘b’ để đánh dấu kết thúc chuỗi số nhị phân với câu lệnh **mov dl, ‘b’** . Giải thích tương tự với câu lệnh **mov dl, ‘0’** .

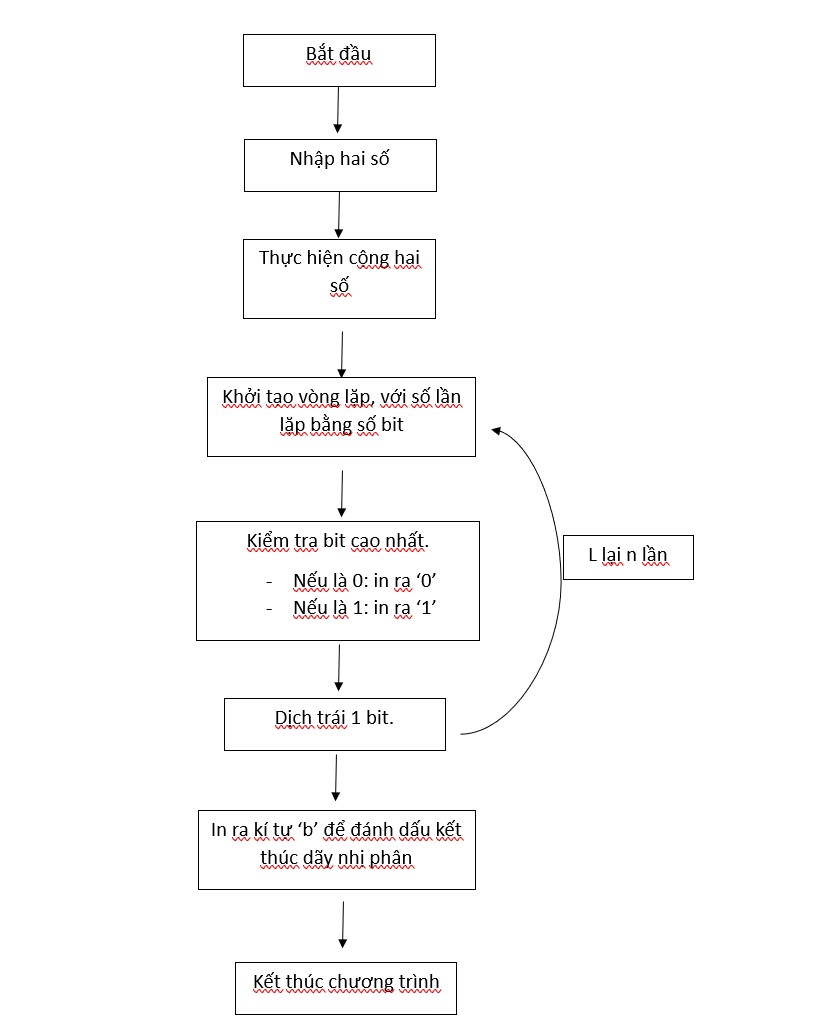


* Gọi hàm để in ký tự ra màn hình với dòng lệnh là **int 21h**. Sau đó màn hình sẽ in ra kí tự ‘b’.
* Gán giá trị 0 vào thanh ghi AH với câu lệnh **mov ah, 0**. Sau đó dùng câu lệnh **int 16h** để người dùng nhấn vào một nút bất kì trên màn hình để kết thúc chương trình. Sau đó kết thúc chương trình bằng câu lệnh ret.





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

* 1. **Chương trình cộng hai số 32 bit trên vi xử lý 8086**
* **Chú thích : Source code sẽ được đính kèm trong bài nộp.**

**Nguyên lý hoạt động**

- Khai báo toán hạng thứ nhất, toán hạng thứ hai, kết quả lần lượt là op1, op2, r. Đồng thời khai báo 2 biến buffer và buffer1 để lưu giá trị nhập từ bàn phím.

- Nhập vào toán hạng thứ nhất ở dạng số thập lục phân. Giá trị này sẽ được lưu vào biến buffer dưới dạng mã ascii. Lưu số kí tự được nhập vào thanh ghi bl. Nếu số vừa nhập vào không đủ 32 bit thì thêm số 0 vào các bit cao. - Lần lượt lấy các kí tự ra bắt đầu từ cuối mảng, chuyển đổi về dạng nhị phân và đưa vào biến op1.

- Tương tự với toán hạng thứ hai.

- Thực hiện cộng hai toán hạng lại như sau:

+ Đưa 16 bit thấp của 2 toán hạng vào thanh ghi bx và cx.

+ Lấy 2 thanh ghi trên cộng cho nhau và lưu kết quả vào 16 bit thấp của biến r. Cờ CF sẽ được bật lên nếu phép toán này có nhớ.

+ Đưa 16 bit cao của 2 toán hạng vào thanh ghi bx và cx.

+ Nếu CF = 1, thực hiện cộng có nhớ 2 thanh ghi trên và lưu kết quả vào 16 bit cao của biến r.

- Sử dụng hàm print\_pin để in từng bit của kết quả ra màn hình.

**3.2.1. Hàm print\_bin**

Khai báo thành ghi cx với giá trị 16 để lặp 16 lần.

Kiểm tra bit cao nhất là ‘0’ hay ‘1’, in ra màn hình bit đó.

Dịch trái 1 bit để kiểm tra bit tiếp theo.

**3.2.2. Hàm half và hàm lilproc**

Hai hàm này được kết hợp để lấy các bit trong buffer ra và chuyển thành số nhị phân 16 bit. Hàm lilproc dùng để chọn giá trị để ghi vào các biến toán hạng dựa trên số lượng kí tự được nhập vào. Nếu số nhập vào không đủ 32 bit, ta sẽ thêm số 0 vào các bit cao.

**3.2.3. Hàm digit**

Kiểm tra và đưa kí tự về dạng binary. Nếu kí tự nhập vào >= 97 thì số vừa nhập là chữ cái in thường, ta sẽ lấy kí tự trên trừ 87. Nếu kí tự nhập vào >= 65 thì số vừa nhập là chữ cái in hoa, ta sẽ lấy kí tự trên trừ 55. Các trường hợp còn lại là số nên ta sẽ trừ cho 48.

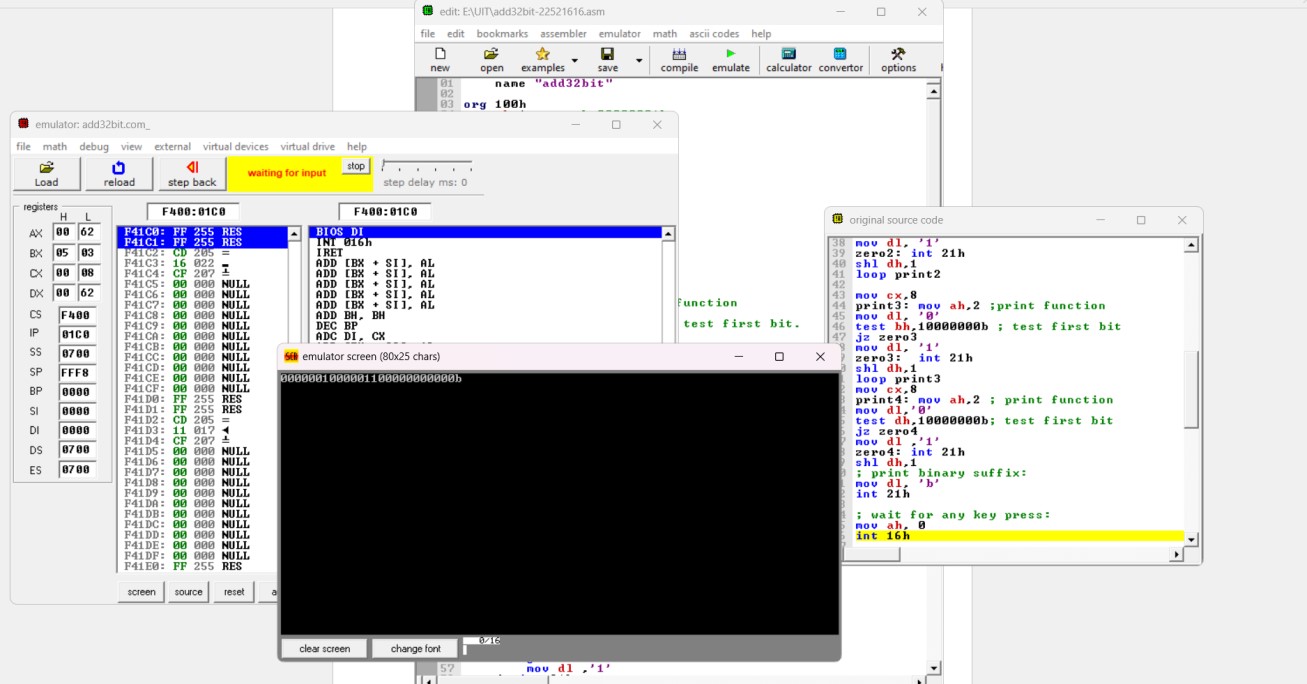
**II. Trừ hai số 32 bit trên vi xử lý 8086.**

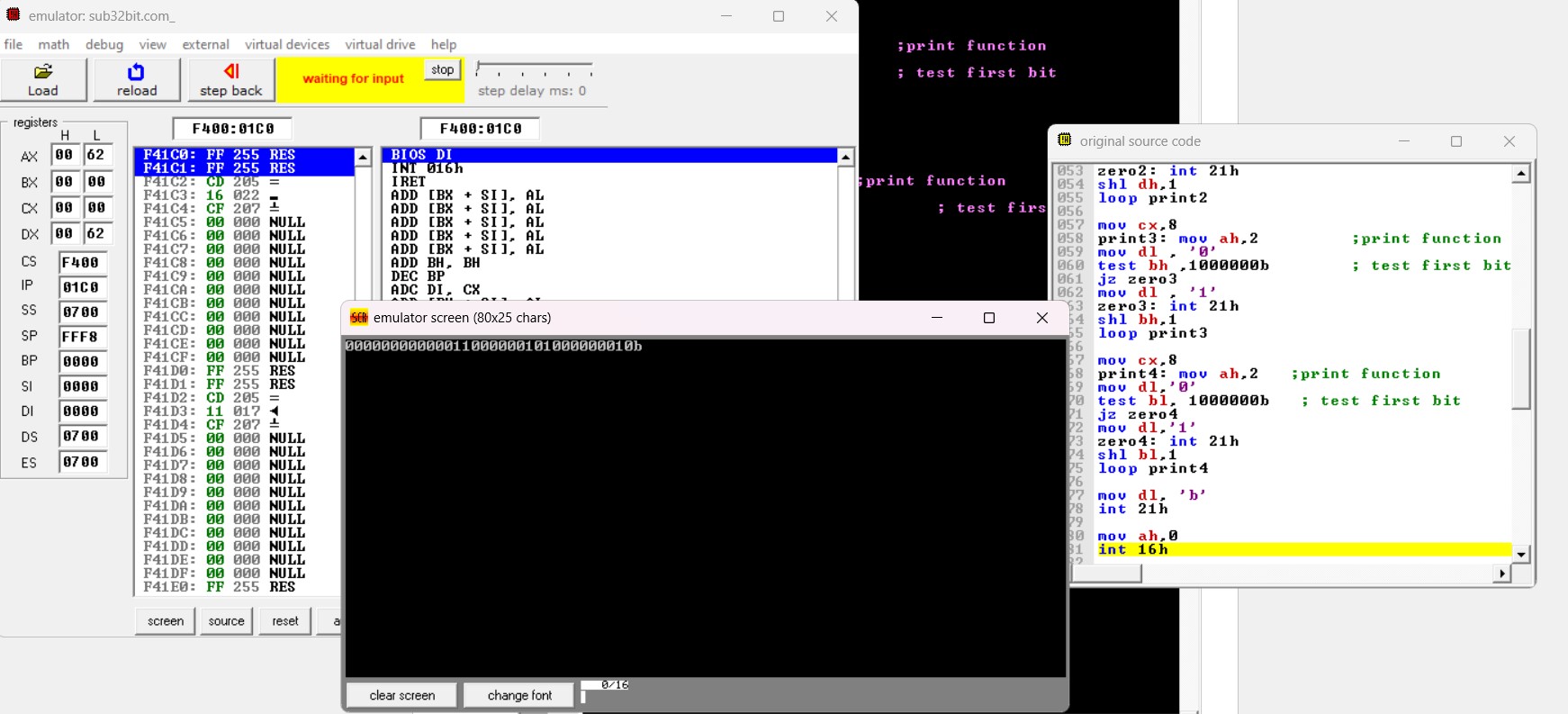
1. **Chương trình cộng hai số 32 bit trên vi xử lý 8086**

Chỉ khác chương trình Cộng 2 số 32 bit ở hàm tính toán

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ

Mô tả được tạo tự động

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



**Nguyên lý hoạt động**

Đưa 16 bit thấp của 2 toán hạng vào thanh ghi bx và cx.

Lấy 2 thanh ghi trên trừ cho nhau và lưu kết quả vào 16 bit thấp của biến r. Cờ CF sẽ được bật lên nếu phép toán này có nhớ.

Đưa 16 bit cao của 2 toán hạng vào thanh ghi bx và cx.

Nếu CF = 1, trừ thanh ghi bx cho 1. Lấy 2 thanh ghi trên trừ cho nhau và lưu kết quả vào 16 bit cao của biến r

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

Khởi tạo các biến và gán giá trị cho chúng

Th

ự

c hi

ệ

n phép đ

ả

o bit (NOT) cho các bi

ế

n

Th

ự

c hi

ệ

n phép

tr

ừ

gi

ữ

a các thanh ghi và c

ậ

p

nh

ậ

t các bit nh

ớ

In k

ế

t qu

ả

nh

ị

phân c

ủ

a thanh ghi ah.

Th

ự

c hi

ệ

n phép c

ộ

ng 1 cho bi

ế

n dl và c

ậ

p

nh

ậ

t các bit nh

ớ

adc

(

).

L

ấ

y giá tr

ị

t

ừ

ngăn x

ế

p và gán l

ạ

i vào

thanh ghi

ax.

Lưu giá tr

ị

c

ủ

a thanh ghi vào ngăn x

ế

p

In kết quả nhị phân của thanh ghi al.

In kết quả nhị phân của các thanh ghi bh và bl.

In kết quả nhị phân của thanh ghi dl.

In k

ế

t

thúc chu

ỗ

i nh

ị

phân ('b').

Ch

ờ

ngư

ờ

i dùng nh

ấ

n m

ộ

t phím b

ấ

t k

ỳ

.

K

ế

t thúc chương trình.