# BÀI THỰC HÀNH 01:

**CỘNG 2 SỐ 32 BIT TRÊN VI XỬ LÝ 8086**

## Sinh viên chuẩn bị

- Sinh viên thực hiện việc cài đặt phần mềm emu8086 được chia sẻ ở trên course

## Nội dung thực hành

* 1. Tham khảo example:
     + Sinh viên mở File >> example >> add/subtract
     + Chạy thử, dựa vào instruction set mô tả hoạt động của nó
     + Biết các sử dụng lại đoạn mã print ra màn hình của chúng
  2. Viết một chương trình cộng 2 số 32 bit.

## Bài tập

Thực hiện chương trình trừ 2 số 32 bit

## Báo cáo

**Nén các file thiết kế và file báo cáo vào 1 file đặt tên như sau: [<LAB…>]-[<MSSV>]-Họ và tên**

File báo cáo yêu cầu gồm những nội dung sau:

* 1. Mô tả cách thức hoạt động của code mẫu
  2. Lưu đồ giải thuật xử lý chương trình cộng 2 số 32 bit.
  3. Giải thích cách thức hoạt động của thuật toán, kèm theo 1 video (gửi link Google Drive, set quyền truy cập cho “bất kì người nào có link”) để chứng minh thuật toán hoạt động.

Họ và tên sinh viên thực hiện: Nguyễn Văn Nhật Minh

Mã số sinh viên:

22520879



BÁO CÁO THỰC HÀNH SỐ 01

<BÀI THỰC HÀNH 01 >

1. Cách thức hoạt động của code mẫu:

Khởi tạo giá trị 5 cho thanh ghi al và giá trị 10 cho thanh ghi bl (al = 5, bl = 10)

- Cộng hai thanh ghi bl và al, kết quả lưu vào thanh ghi bl (bl = 15)

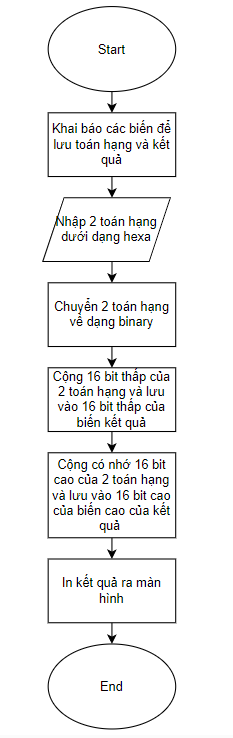
- Trừ thanh ghi bl cho 1 và lưu kết quả vào thanh ghi vl (bl = 14)

- Chọn interrupt 21h/02h để in từng kí tự ra màn hình

- And bit MSB với 80h để kiểm tra bit ’0’ hay ‘1’và in ra màn hình

- Dịch trái 1 bit để kiểm tra bit tiếp theo.

- In kí tự ‘b’

II. Lưu đồ thuật toán của cộng 2 số 32 bit

1. Bài Tập

3.Cộng 2 số 32 bit

org 100h

.data

msg1 db 13, 10,"Nhap vao so thu 1: "

msg2 db 13, 10,"Nhap vao so thu 2: "

msg3 db 13, 10," Tong: $"

buffer db 10,?,8 dup<0>,0,0

buffer1 db 10,?,8 dup<0>,0,0

op1 dd

op2 dd

r dd

.Code

Start:

;Nhap so thu nhat

mov dx, offset msg1 ;In thong bao nhap

mov ah, 9

int 21h

mov dx, offset buffer

mov ah, 10

int 21h

mov bx, 0

mov bl, buffer[1]

mov buffer [bx+2], 0

lea si, buffer+2

mov cl, bl

mov ch, 0

lap:

inc si

loop lap

dec si

mov cx, 0

call half

mov word ptr op1+2, dx

call half

mov word ptr op1, dx

;Nhap so thu hai

mov dx,offset msg2 ;In thong bao nhap

mov ah,9

int 21h

mov dx,offset,buffer

mov ah,10

int 21h

mov bx,0

mov bl,buffer1[1]

mov buffer1 [bx+2],0

lea si,buffer1 +2

mov cl,bl

mov ch,0

lap1:

inc si

loop lap1

dec si

mov cx,0

call half

mov word ptr op2+2, dx

call half

mov word ptr op2, dx

;Tinh toan

mov bx, word ptr op1 +2

mov cx, word ptr op2+2

add bx, cx

mov word ptr r+2, bx

mov bx, word ptr op1

mov cx, word ptr op2

adc bx, cx

jo error

mov word ptr , bx

;In ket qua

mov dx, offset msg3

mov ah, 9

int 21h

mov bx, word ptr r

call print\_bin

mov bx, word ptr r+2

call print\_bin

mov dl, 'b'

int 21h

jz exit

;Ham print

print\_bin proc near

mov cx, 16

print:

mov ah, 2

mov dl, '0'

test bx, 8000h

jz zero

mov dl, '1'

zero:

int 21h

shl bx, 1

loop print

ret

print\_bin endp

;Ham luu gia tri vao mang

half proc

ret

call lilproc

call digit

mov dl, al

; dec bh

call lilproc

call digit

mov c1, 4

shl al, cl

add dl, al

; dec bh

call lilproc

call digit

mov dh, al ; dec bh

call lilproc

call digit

mov cl, 4

shl al, cl

add dh, al

; dec bh

half endp

;Ham mo rong bit

lilproc proc

cmp bl, ch

ja lay

mov al, 48

jmp lay1

lay:

mov al, [si]

dec si

dec bl

layl:

ret

lilproc endp

error:

exit:

RET

**3.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.

**4. Trừ 2 số 32 bit**

**4.1. Source code**

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

**4.2. 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