

**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KỸ THUẬT MÁY TÍNH**



BÁO CÁO THỰC HÀNH LAB 05
MÔN THỰC HÀNH VI XỬ LÝ-VI ĐIỀU KHIỂN

Họ và tên sinh viên : Nguyễn Hữu Tứ
Mã số sinh viên: 19522453

GIẢNG VIÊN HƯỚNG DẪN
Bùi Phùng Hữu Đức

TP. HỒ CHÍ MINH, 6/2021

I. Cộng trừ 2 số hexal có tràn số:

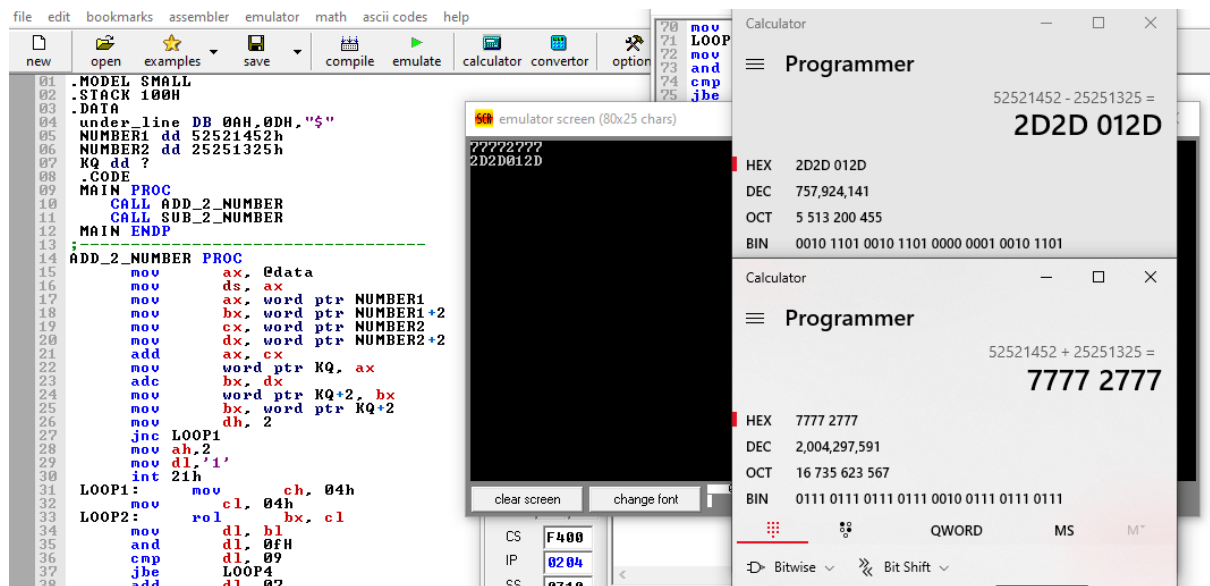
*Trình tự 2 số lưu vào NUMBER1,NUMBER2

-Thực hiện phép tính cộng, sau đó thực hiện phép tính trừ

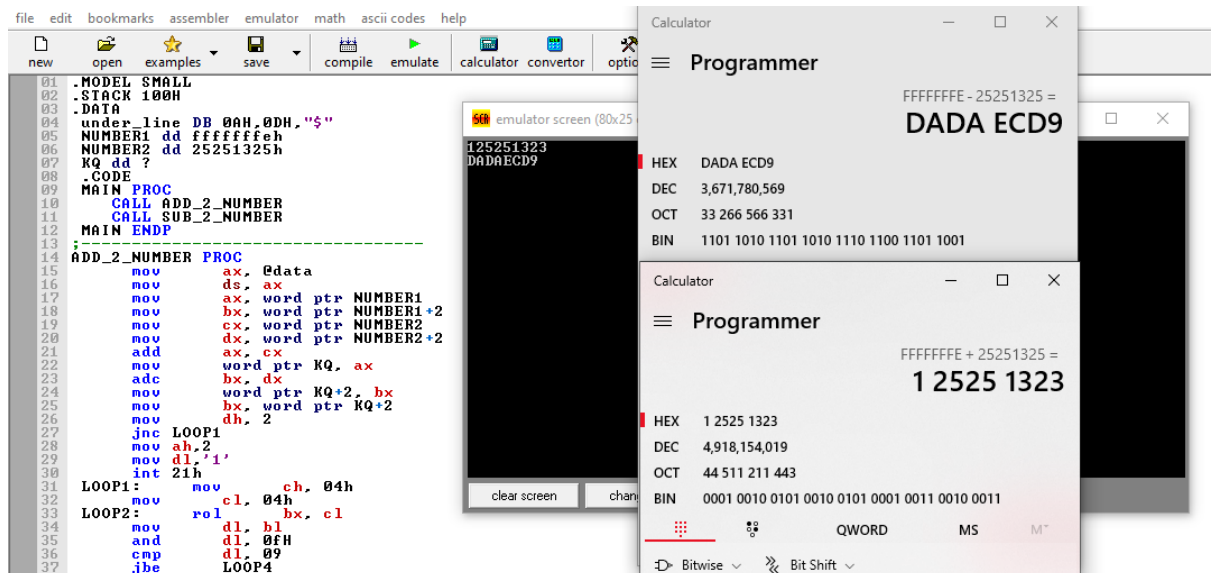
Chương trình	Ý nghĩa
<pre> .MODEL SMALL .STACK 100H .DATA under_line DB 0AH,0DH,"\$" NUMBER1 dd 52521452h NUMBER2 dd 25251325h KQ dd ? .CODE MAIN PROC CALL ADD_2_NUMBER CALL SUB_2_NUMBER MAIN ENDP ;----- ADD_2_NUMBER PROC mov ax, @data mov ds, ax mov ax, word ptr NUMBER1 mov bx, word ptr NUMBER1+2 mov cx, word ptr NUMBER2 mov dx, word ptr NUMBER2+2 add ax, cx mov word ptr KQ, ax adc bx, dx mov word ptr KQ+2, bx mov bx, word ptr KQ+2 mov dh, 2 jnc LOOP1 mov ah,2 </pre>	<p>Dữ liệu dùng để xuống dòng</p> <p>Số thứ nhất</p> <p>Số thứ hai</p> <p>Biến lưu kết quả</p> <p>Vào hàm chính</p> <p>Gọi hàm cộng 2 số</p> <p>Gọi hàm trừ 2 số</p> <p>Kết thúc hàm chính</p> <p>Hàm cộng 2 số</p> <p>Lưu lsb của NUMBER1 vào ax</p> <p>Lưu msb của NUMBER1 vào bx</p> <p>Lưu lsb của NUMBER2 vào cx</p> <p>Lưu msb của NUMBER2 vào dx</p> <p>Cộng lsb NUMBER1 cho lsb NUMBER2</p> <p>Lưu kết quả vào lsb của KQ</p> <p>Cộng msb NUMBER1 với msb NUMBER2 cộng nhớ</p> <p>Lưu kết quả vào msb của KQ</p> <p>Lưu msb của KQ vào bx</p> <p>Nếu cờ nhớ CF = 0 thì nhảy(TH không tràn)</p> <p>Nếu có thì in ra số 1 ngoài màn hình(Tràn)</p>

<pre> mov dl,'1' int 21h LOOP1: mov ch, 04h mov cl, 04h LOOP2: rol bx, cl mov dl, bl and dl, 0fH cmp dl, 09 jbe LOOP4 add dl, 07 LOOP4: add dl, 30H mov ah, 02 int 21H dec ch jnz LOOP2 dec dh cmp dh, 0 mov bx, word ptr KQ jnz LOOP1 mov ah, 4ch RET ADD_2_NUMBER ENDP ;----- SUB_2_NUMBER PROC lea dx,under_line mov ah,9 int 21h mov ax, @data mov ds, ax mov ax, word ptr NUMBER1 mov bx, word ptr NUMBER1+2 mov cx, word ptr NUMBER2 mov dx, word ptr NUMBER2+2 sub ax, cx </pre>	<p>Đặt count = 4 (4 kí tự in ra)</p> <p>Lưu bl vào dl</p> <p>Chỉ lấy giá trị lsb của dl</p> <p>Kiểm tra kí tự là 0-9 hay là A-F</p> <p>Nếu là A-F thì cộng 37h</p> <p>Nếu là 0-9 thì cộng 30h</p> <p>Ngắt để in kí tự</p> <p>Giảm count</p> <p>Lưu lsb của KQ vào bx</p> <p>Kết thúc hàm cộng 2 số</p> <p>Hàm trừ 2 số</p> <p>Hàm để xuống dòng ở display</p> <p>Lưu lsb của NUMBER1 vào ax</p> <p>Lưu msb của NUMBER1 vào bx</p> <p>Lưu lsb của NUMBER2 vào cx</p> <p>Lưu msb của NUMBER2 vào dx</p> <p>Trừ lsb NUMBER1 cho lsb NUMBER2</p> <p>Lưu kết quả vào lsb của KQ</p>
---	---

<pre> mov word ptr KQ, ax sub bx, dx mov word ptr KQ+2, bx mov bx, word ptr KQ+2 mov dh, 2 LOOP11: mov ch, 04h mov cl, 04h LOOP21: rol bx, cl mov dl, bl and dl, 0fH cmp dl, 09 jbe LOOP31 add dl, 07 LOOP31: add dl, 30H mov ah, 02 int 21H dec ch jnz LOOP21 dec dh cmp dh, 0 mov bx, word ptr KQ jnz LOOP11 mov ah, 4ch int 21h RET SUB_2_NUMBER ENDP ;----- END MAIN </pre>	<p>Trừ msb NUMBER1 với msb NUMBER2</p> <p>Lưu kết quả vào msb của KQ</p> <p>Lưu msb của KQ vào bx</p> <p>Nếu cờ nhớ CF = 0 thì nhảy(TH không tràn)</p> <p>Nếu có thì in ra số 1 ngoài màn hình(Tràn)</p> <p>Đặt count = 4 (4 kí tự in ra)</p> <p>Lưu bl vào dl</p> <p>Chỉ lấy giá trị lsb của dl</p> <p>Kiểm tra kí tự là 0-9 hay là A-F</p> <p>Nếu là A-F thì cộng 37h</p> <p>Nếu là 0-9 thì cộng 30h</p> <p>Ngắt để in kí tự</p> <p>Giảm count</p> <p>Lưu lsb của KQ vào bx</p> <p>Kết thúc hàm cộng 2 số</p>
---	--



Hình 1: Trường hợp cộng trừ không tràn



Hình 2: Trường hợp cộng trừ tràn

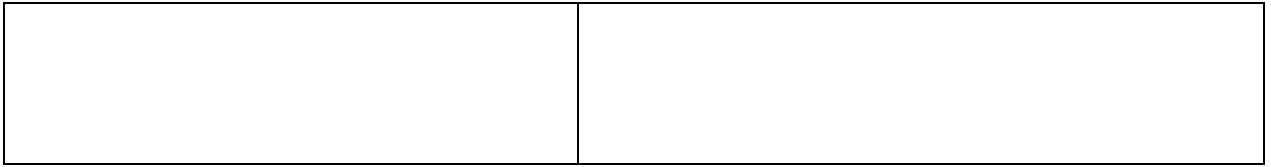
II. Chuyển một số từ Decimal sang Binary:

Chương trình	Ý nghĩa
.MODEL SMALL .STACK 100H	

<pre> .DATA ten DW 10 two DB 2 tb db "Nhap so can chuyen: \$ " tb1 db "So sau khi chuyen la: \$" KQ db 8 dup('0') temp DW ? minus DB ? ;----- MAIN PROC mov ax,@data mov ds,ax mov ah,9 lea dx,tb int 21h CALL INPUT CALL PRINT_ARRAY db 13, 10, 0 CALL CONVERT_BIN mov ah,9 lea dx,tb1 int 21h LEA SI, KQ ADD SI, 7 MOV AH, 0Eh MOV CX, 8 print_bin: MOV AL, [SI] INT 10h DEC SI LOOP print_bin MAIN ENDP ;----- CONVERT_BIN PROC PUSH AX PUSH DI LEA DI, KQ MOV AX, CX MOV AH, 0 </pre>	<p>Khai báo biến ten có giá trị là 10 Khai báo biến two có giá trị là 2 Thông báo chuỗi nhập decimal Thông báo chuỗi xuất binary Mảng KQ có 8 phần tử với giá trị là 0 Biến temp không có giá trị ban đầu Biến minus không có giá trị ban đầu</p> <p>Hàm chính Gán địa chỉ của .DATA vào ax</p> <p>Thực hiện hàm ngắt 9 để in ra 1 chuỗi kí tự thông báo nhập số cần chuyển sang binary</p> <p>Sau đó, gọi hàm nhập Xuất giá trị đó ra Chuỗi cần in</p> <p>Hàm chuyển sang binary Thực hiện in chuỗi thông báo số sau khi chuyển là</p> <p>Gán SI = địa chỉ KQ SI + 7 (SI = địa chỉ KQ + 7) Gán AH = 0Eh (phục vụ ngắt) Gán CX = 8, số vòng lặp Hàm này để in ra từng kí tự trong SI từ địa chỉ lớn nhất đến nhỏ nhất. Mục đích in ngược các giá trị được lưu vào stack Gán AL = [SI] (= Result + 7) Ngắt để in kí tự Giảm địa chỉ SI Lặp lại Kết thúc hàm main</p> <p>//Phần hàm con Hàm chuyển sang nhị phân Lưu lại AX vào stack Lưu lại DI Gán DI = địa chỉ bộ nhớ KQ Gán AX = CX Gán AH = 0</p>
--	---

<pre> next_step: DIV CS:two MOV [DI], AH ADD [DI], 30h INC DI MOV AH, 0 CMP AX, 0 JNE next_step do_reset: CMP DI, OFFSET KQ + 8 JAE done MOV [DI], '0' INC DI JMP do_reset done: POP DI POP AX RET CONVERT_BIN ENDP ;----- INPUT PROC PUSH DX PUSH AX PUSH SI MOV CX, 0 MOV CS:minus, 0 next: MOV AH, 00h INT 16h MOV AH, 0Eh INT 10h CMP AL, 13 JNE not_cr JMP stop_input not_cr: CMP AL, 8 JNE checked MOV AX, CX DIV CS:ten MOV CX, AX JMP next checked: CMP AL, '9' JBE digit </pre>	<pre> AX = AX/2 số dư lưu vào AH Gán [DI] = AH [DI] + 30h để chuyển về ASCII Tăng DI Gán AH = 0 So sánh AX với 0 So sánh DI với chỉ số của KQ Nhảy nếu ZF = 0 và CF = 0 Gán [DI] = '0' Tăng địa chỉ DI Nhảy tới do_reset Hàm nhập kí tự vào Mục đích nhập vào kí tự.Nếu người dùng nhấn phím enter thì thoát. Hàm nhận kí tự nhập vào Push DX(chứa giá trị hex của số nhập) Push AX(Chứa kí tự cuối của giá trị nhập) Gán CX = 0 Gán minus = 0 Gán AH = 00h(Phục vụ ngắt) Ngắt Gán AH = 0Eh(Phục vụ ngắt) Ngắt AL – 13 nếu bằng 0 thì ZF = 1 Nhảy nếu ZF = 0 Nhảy đến stop_input AL – 8 nếu = 0 thì ZF = 1 Nhảy nếu ZF = 0 Gán AX = CX AX = AX/10 Gán CX = AX Nhảy next Hàm để xét nếu chỉ có 1 kí tự So sánh với 9 Nhảy nếu ZF = 1 Nhảy đến next </pre>
---	--

<pre> JMP next digit: PUSH AX MOV AX, CX MUL CS:ten MOV CX, AX POP AX CMP DX, 0 SUB AL, 30h MOV AH, 0 MOV DX, CX ADD CX, AX JMP next stop_input: CMP CS:minus, 0 JE not_minus NEG CX not_minus: POP SI POP AX POP DX RET INPUT ENDP ;----- PRINT_ARRAY PROC MOV CS:temp, SI POP SI PUSH AX next_char: MOV AL, CS:[SI] INC SI CMP AL, 0 JZ printed MOV AH, 0Eh INT 10h JMP next_char printed: POP AX PUSH SI MOV SI, CS:temp RET PRINT_ARRAY ENDP END MAIN </pre>	<pre> Push AX vào stack Gán AX = CX AX = AX*10 Gán CX = AX Pop AX AL – 30 để chuyển sang dạng ASCII Gán AH = 0 Gán DX = CX CX = CX + AX Nhảy hàm next So sánh make với 0 Nhảy nếu điều kiện trên đúng Đảo dấu CX Lấy lại giá trị SI Lấy lại giá trị AX Lấy lại giá trị DX Quay lại từ lệnh gọi Kết thúc hàm ;----- Hàm in ra display Gán temp = SI Lấy giá trị vào SI Đưa giá trị AX vào stack Gán AL = temp[SI] Tăng SI (tăng địa chỉ) So sánh AL với 0 Nhảy nếu ZF = 1 Gán AH = 0Eh để ngắt Nhảy tới next_char Kết thúc hàm Kết thúc chương trình </pre>
--	--



*Chạy kết quả mô phỏng:

