

THỰC HÀNH VI XỬ LÝ – VI ĐIỀU KHIỂN

GVHD: Trần Hoàng Lộc

BÀI THỰC HÀNH 06:

XỬ LÝ IO, TÍNH TOÁN VÀ BỘ NHỚ TRÊN 8086

I. Sinh viên chuẩn bị

- Sinh viên nắm rõ cách truy xuất và sử dụng bộ nhớ trên vi xử lý 8086
- Sinh viên nắm rõ tập lệnh hợp ngữ trên 8086

II. Nội dung thực hành

1. Nhập một số N có 2 chữ số từ bàn phím thông qua màn hình console.
2. In ra màn hình N số Fibonacci đầu tiên.

III. Bài tập

Cũng với yêu cầu trên, sử dụng một cách khác để thực hiện. **IV.**

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. Lưu đồ thuật toán xử lý yêu cầu trên.
2. Giải thích code, các bước xử lý trong code và trình bày kết quả thực hiện.
3. Trình bày lưu đồ xử lý và giải thích code trong phần bài tập.

PHỤ LỤC

ASCII Character Code Chart

MJ Karas

Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char	Dec	Hex	Oct	Char
0	00	000	NUL	32	20	040	SP	64	40	100	@	96	60	140	`
1	01	001	SOH	33	21	041	!	65	41	101	A	97	61	141	a
2	02	002	STX	34	22	042	"	66	42	102	B	98	62	142	b
3	03	003	ETX	35	23	043	#	67	43	103	C	99	63	143	c
4	04	004	EOT	36	24	044	\$	68	44	104	D	100	64	144	d
5	05	005	ENQ	37	25	045	%	69	45	105	E	101	65	145	e
6	06	006	ACK	38	26	046	&	70	46	106	F	102	66	146	f
7	07	007	BEL	39	27	047	'	71	47	107	G	103	67	147	g
8	08	010	BS	40	28	050	(72	48	110	H	104	68	150	h
9	09	011	TAB	41	29	051)	73	49	111	I	105	69	151	i
10	0A	012	LF	42	2A	052	*	74	4A	112	J	106	6A	152	j
11	0B	013	VT	43	2B	053	+	75	4B	113	K	107	6B	153	k
12	0C	014	FF	44	2C	054	,	76	4C	114	L	108	6C	154	l
13	0D	015	CR	45	2D	055	-	77	4D	115	M	109	6D	155	m
14	0E	016	SO	46	2E	056	.	78	4E	116	N	110	6E	156	n
15	0F	017	SI	47	2F	057	/	79	4F	117	O	111	6F	157	o
16	10	020	DLE	48	30	060	0	80	50	120	P	112	70	160	p
17	11	021	DC1	49	31	061	1	81	51	121	Q	113	71	161	q
18	12	022	DC2	50	32	062	2	82	52	122	R	114	72	162	r
19	13	023	DC3	51	33	063	3	83	53	123	S	115	73	163	s
20	14	024	DC4	52	34	064	4	84	54	124	T	116	74	164	t
21	15	025	NAK	53	35	065	5	85	55	125	U	117	75	165	u
22	16	026	SYN	54	36	066	6	86	56	126	V	118	76	166	v
23	17	027	ETB	55	37	067	7	87	57	127	W	119	77	167	w
24	18	030	CAN	56	38	070	8	88	58	130	X	120	78	170	x
25	19	031	EM	57	39	071	9	89	59	131	Y	121	79	171	y
26	1A	032	SUB	58	3A	072	:	90	5A	132	Z	122	7A	172	z
27	1B	033	ESC	59	3B	073	;	91	5B	133	[123	7B	173	{
28	1C	034	FS	60	3C	074	<	92	5C	134	\	124	7C	174	
29	1D	035	GS	61	3D	075	=	93	5D	135]	125	7D	175	}
30	1E	036	RS	62	3E	076	>	94	5E	136	^	126	7E	176	~
31	1F	037	US	63	3F	077	?	95	5F	137	_	127	7F	177	DEL

THỰC HÀNH VI XỬ LÝ – VI ĐIỀU KHIỂN

GVHD: Trần Hoàng Lộc Họ

và tên sinh viên thực hiện:

Mã số sinh viên:

BÁO CÁO THỰC HÀNH SỐ . . .

<TÊN BÀI THỰC HÀNH>

I. Nội dung 1 II.

Nội dung 2

III. Bài tập (nếu có)

IV. Tài liệu tham khảo

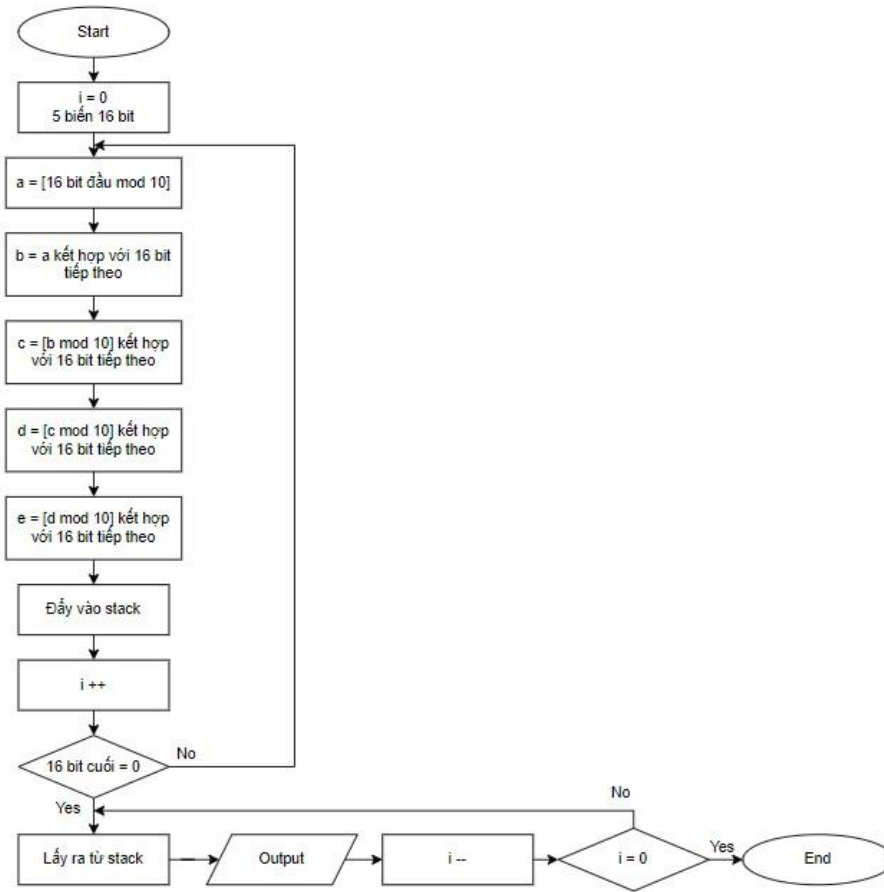
Họ tên: Đoàn Vũ Phú Minh

MSSV: 22520859

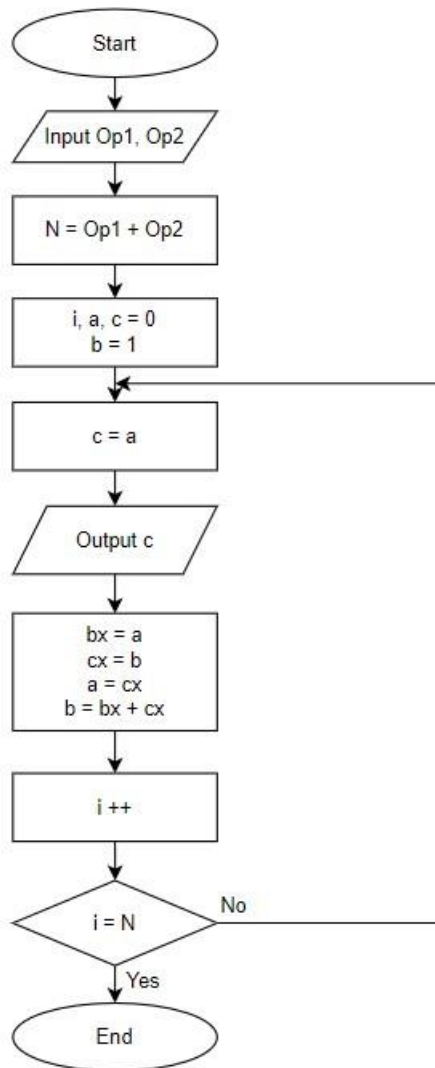
I. Nhập vào 2 chữ số có 2 chữ số, tính tổng N và in ra màn hình N số đầu tiên của chuỗi Fibonacci

1. Flow chart

1.1. Xuất ra số 80 bits phù hợp với giá trị của số fibonacci thứ 99 (67 bits)



1.2. Xuất chuỗi fibonacci với N phần tử, với N là tổng 2 số 2 chữ số nhập từ bàn phím



2.Source code

```

org 100h
.Data
msg1 db 10, 13, "Nhap vao so thu nhat: $"
msg2 db 10, 13, "Nhap vao so thu hai: $"
msg3 db 10, 13, "Fibonacci: $"
op1 db ?
op2 db ?
count dw ?
;So 80 bit
a1 dw 0 ;a
a2 dw 0
a3 dw 0
a4 dw 0
a5 dw 0

b1 dw 0 ;b
b2 dw 0
b3 dw 0
b4 dw 0
b5 dw 1

c1 dw 0 ;c
c2 dw 0
c3 dw 0
c4 dw 0
c5 dw 0
.Code
Start:
;Nhap so thu nhat
mov dx, offset msg1 ;In thong bao nhap
mov ah, 9
int 21h
call input
mov op1, bl
;Nhap so thu hai
mov dx, offset msg2 ;In thong bao nhap
mov ah, 9
int 21h
call input
mov op2, bl

;Tinh N
mov al, op1
mov bl, op2
add al, bl
mov ah, 100
cmp al, ah
jl Hehe
mov al, 99
Hehe:
xor ah, ah
mov count, ax
mov dx, offset msg3 ;In thong bao xuat chuoai
mov ah, 9
int 21h
call fibonacci
Done:
ret
;=====
;Ham xuat so 80 bit ra man hinh
Output_Num proc
push ax
push bx
push cx
push dx

mov bx, 10
mov cx, 0

```

```

Num_Separation:
    xor dx, dx
    mov ax, c1
    div bx
    mov c1, ax

    mov ax, c2
    div bx
    mov c2, ax

    mov ax, c3
    div bx
    mov c3, ax

    mov ax, c4
    div bx
    mov c4, ax

    mov ax, c5
    div bx
    mov c5, ax

    push dx
    inc cx
    cmp ax, 0
    je Output
    jmp Num_Separation
Output:
    pop dx

    add dl, 30h
    mov ah, 2
    int 21h

    dec cx

    cmp cx, 0
    jne Output

    pop dx
    pop cx
    pop bx
    pop ax

ret
Output_Num endp

```

;Den so ki tu

;Cong 2 so 80 bit

Add_80bit proc

push bx

push cx

;Cong 16 bit cuoi

mov bx, a5

mov cx, b5

mov a5, cx

add bx, cx

mov b5, bx

;Cong 16 bit tiep theo

mov bx, a4

mov cx, b4

mov a4, cx

adc bx, cx

mov b4, bx

;Cong 16 bit tiep theo

mov bx, a3

mov cx, b3

mov a3, cx

adc bx, cx

mov b3, bx

;Cong 16 bit tiep theo

mov bx, a2

mov cx, b2

mov a2, cx

adc bx, cx

mov b2, bx

;Cong 16 bit dau

mov bx, a1

mov cx, b1

mov a1, cx

adc bx, cx

mov b1, bx

pop cx

pop bx

ret

Add_80bit endp

;=====

Fibonacci proc

mov bx, 0

fibo:

mov ax, a1

;c=a

mov c1, ax

mov ax, a2

mov c2, ax

mov ax, a3

mov c3, ax

mov ax, a4

mov c4, ax

mov ax, a5

mov c5, ax

mov ah, 2

;in khoang cach

mov dl, 9

int 21h

call Output_Num

call Add_80bit

inc bx

cmp bx, count

je exitfibo

jmp fibo

exitfibo:

ret

Fibonacci endp

```

;=====
;Ham nhap
input proc
    mov ah, 1
    int 21h
    mov bl, al
    sub bl, 30h
    int 21h
    mov bh, al
    cmp bh, 13
    je input1
    sub bh, 30h
    mov al, 10
    mul bl
    add al, bh
    mov bl, al
input1:
    ret
input endp

```

3. Nguyên lý hoạt động

Khai báo a, b, c là 3 biến với 80 bit.

Khai báo op1, op2 lần lượt là 2 chữ số nhập vào, count là tổng N của 2 chữ số trên.

Nhập vào 2 số thập phân 2 chữ số, tính tổng N của 2 số trên. Nếu $N > 99$ thì $N = 99$.

In ra dãy số fibonacci với quy tắc: $c = a$, output c, temp = b, $b = a + b$, $a = temp$.

3.1. Hàm input

Nhập kí tự đầu tiên và thứ hai. Kiểm tra kí tự thứ 2: nếu là enter thì số vừa nhập chỉ có 1 chữ số; nếu khác enter thì số vừa nhập là số có 2 chữ số, lấy kí tự đầu nhân cho 10 và cộng với kí tự thứ 2 để ra giá trị của số vừa nhập vào.

3.2. Hàm Fibonacci

Khởi tạo giá trị 0 cho thanh ghi bx. Cho biến $c = a$, sau đó in biến c ra màn hình bằng hàm Output_Num. Tính toán giá trị của số tiếp theo trong chuỗi fibonacci. Tiếp tục thực hiện cho đến khi in ra đủ N chữ số trong chuỗi fibonacci.

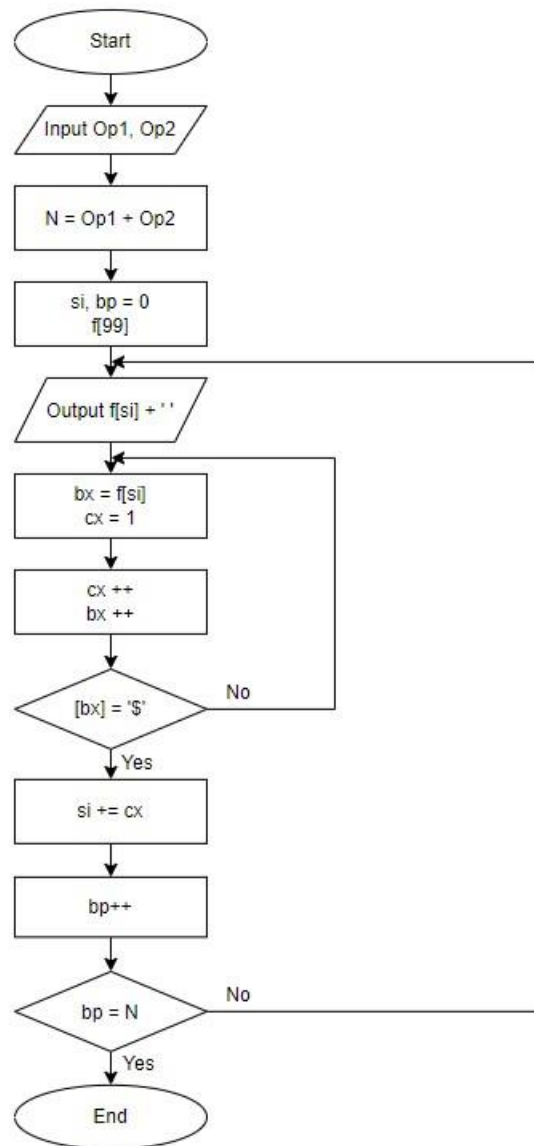
3.3. Hàm Add_80bit

Tính toán giá trị của chữ số tiếp theo với quy tắc $a = b$, $b =$ tổng giá trị của 2 chữ số trước đó.

3.4. Hàm Output_Num

Lấy 16 bit đầu chia lấy dư cho 10 rồi kết hợp kết quả đó với 16 bit tiếp theo, tiếp tục chia lấy dư cho 10. Lặp lại cho đến khi chia đến 16 bit cuối, sau đó lấy số dư để vào stack. Khi giá trị 16 bit cuối = 0 thì dừng rồi lấy từng giá trị trong stack để in ra.

II. Dùng cách khác để thực hiện yêu cầu trên 1. Flow chart



2.Source code

```
org 100h
.Data
msg1 db 10, 13, "Nhap vao so thu nhat: $"
msg2 db 10, 13, "Nhap vao so thu hai: $"
msg3 db 10, 13, "Fibonacci: $"
op1 db ?
op2 db ?
count dw ?
;Mang tu 0 den 7778742049
f1 db "0$", "1$", "1$", "2$", "3$", "5$", "8$", "13$", "21$", "34$",
;Mang tu 12586269025 den 135301852344706746049
f2 db "12586269025$", "20365011074$", "32951280099$", "53316291173$",
.Code
Start:
;Nhap so thu nhat
mov dx, offset msg1
mov ah, 9
int 21h
call input
mov op1, bl
;Nhap so thu hai
mov dx, offset msg2
mov ah, 9
int 21h
call input
mov op2, bl
;Tinh N
mov al, op1
mov bl, op2
add al, bl
mov ah, 100
cmp al, ah
jl Hehe
mov al, 99
Hehe:
xor ah, ah
mov count, ax
mov dx, offset msg3
mov ah, 9
int 21h
call fibonacci
Done:
ret
```

```

;Fibonacci
fibonacci proc
    mov si, 0
    mov bp, 0
    next:
        mov cx, 1
        lea bx, f1[si]
        mov dx, bx
        mov ah, 9
        int 21h
        mov ah, 2
        mov dl, ','
        int 21h
    next1:
        cmp [bx], '$'
        je next2
        inc bx
        inc cx
        jmp next1
    next2:
        add si, cx
        inc bp
        cmp bp, 50
        je next3
        cmp bp, count
        je Exit_fibo
        jmp next
    next3:
        mov si, 0
    next4:
        mov cx, 1
        lea bx, f2[si]
        mov dx, bx
        mov ah, 9
        int 21h
        mov ah, 2
        mov dl, ','
        int 21h
    next5:
        cmp [bx], '$'
        je next6
        inc bx
        inc cx
        jmp next5
    next6:
        add si, cx
        inc bp
        cmp bp, count
        je Exit_fibo
        jmp next4
Exit_fibo:
    ret
fibonacci endp
;=====

;Ham nhap
input proc
    mov ah, 1
    int 21h
    mov bl, al
    sub bl, 30h
    int 21h
    mov bh, al
    cmp bh, 13
    je input1
    sub bh, 30h
    mov al, 10
    mul bl
    add al, bh
    mov bl, al
input1:
    ret
input endp
;=====

```

;In day fibonacci tu 1 den 50

;In day fibonacci tu 51 den 99

3. Nguyên lý hoạt động

Đưa 99 số fibonacci vào 2 chuỗi f1 và f2: f1 chứa 50 số đầu tiên, từ 0 đến

7778742049; f2 chứa các số còn lại, từ 1258626025 đến 135301852344706746049.

Khai báo op1, op2 lần lượt là 2 chữ số nhập vào, count là tổng N của 2 chữ số trên.

Nhập vào 2 số thập phân 2 chữ số, tính tổng N của 2 số trên. Nếu $N > 99$ thì $N = 99$. Xuất lần lượt n phần tử trong mảng.

3.1. Hàm input

Nhập kí tự đầu tiên và thứ hai. Kiểm tra kí tự thứ 2: nếu là enter thì số vừa nhập chỉ có 1 chữ số; nếu khác enter thì số vừa nhập là số có 2 chữ số, lấy kí tự đầu nhân cho 10 và cộng với kí tự thứ 2 để ra giá trị của số vừa nhập vào.

3.2. Hàm fibonacci

B1: Khai báo thanh ghi si và bp bằng 0

B2: Khai báo thanh ghi cx bằng 1, đây là thanh ghi đếm số kí tự của số fibonacci

B3: In chữ số ở vị trí si ra màn hình

B4: Kiểm tra từng kí tự của số cho đến khi tìm ra kí tự '\$'

B5: Đưa con trỏ đến vị trí của số tiếp theo

B6: Kiểm tra đã in ra đủ N số chưa? Nếu chưa, lặp lại B2

B7: Kết thúc