

PRACTICE EXERCISES OF THE MICROPROCESSORS &  
MICROCONTROLLERS

Instructor: The Tung Than

Student's name: Trần Lê Minh Đăng

Student code: 21520684

---

PRACTICE REPORT NO 6

PROCESSING, CALCULATION AND MEMORY ON THE 8086  
MICROPROCESSOR

**I. Flowchart of the above request processing algorithm.**

**- Giải thuật tìm số Fibonacci**

B1: Tạo biến N là số Fibo cần tìm

B2: Sử dụng 3 biến S1, S2, S3

B3: Khởi tạo giá trị  $S1 = 0$ ,  $S2 = 1$ , S3

B4:  $N \leq 0$  thì tới B9

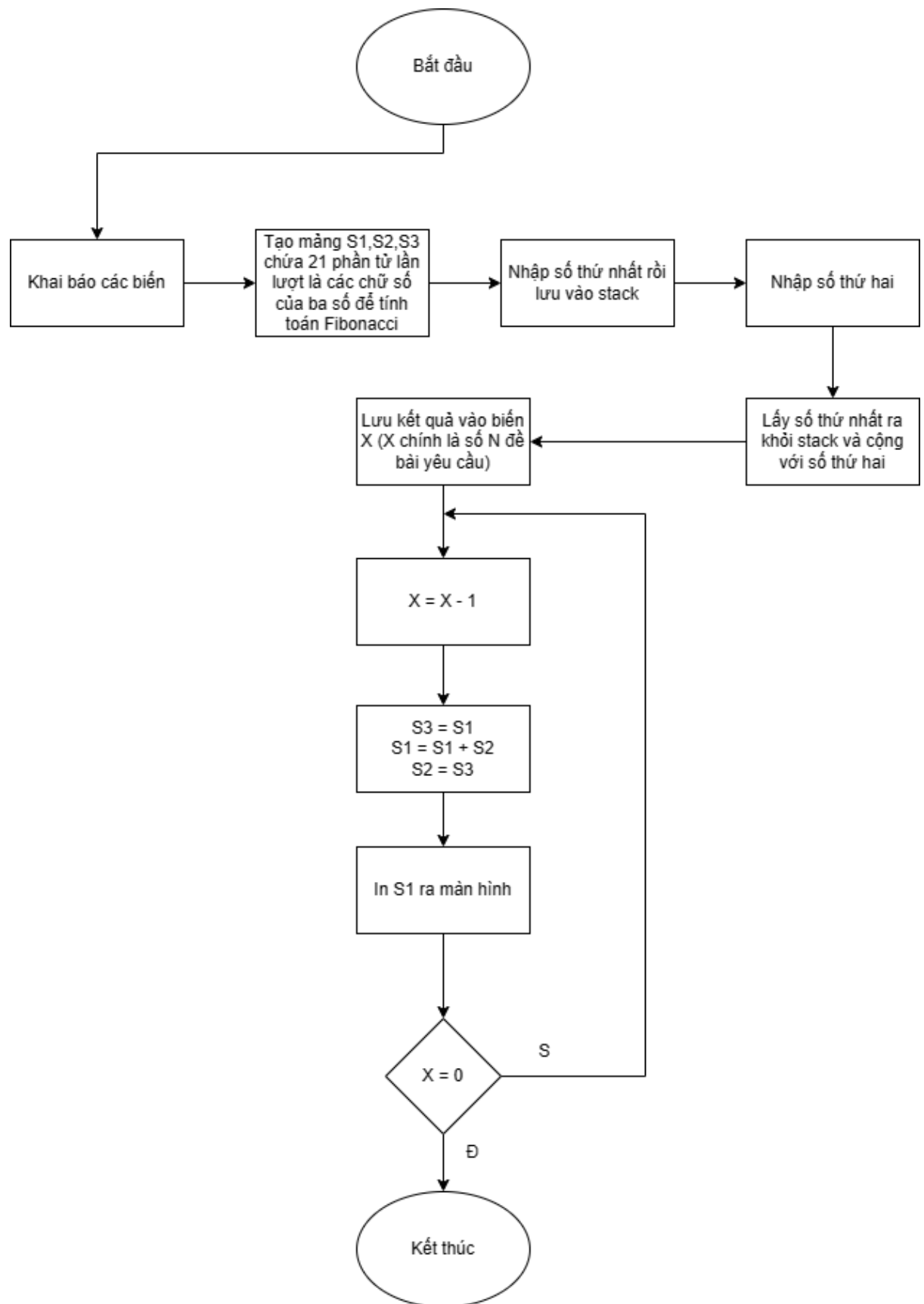
B5:  $S3 = S1$

B6:  $S1 = S1 + S2$

B7:  $S2 = S3$

B8:  $N = N - 1$ , quay lại B4

B9: Kết thúc



**II. Explain how the algorithm works, accompanied by a video (send a Google Drive link) to demonstrate the circuit operation in case the instructor cannot run the design file.**

Link:

<https://drive.google.com/drive/folders/1KsxMfc27DKhFKsino96lrRaNtbjdYfkG?usp=sharing>

SOURCE CODE:

Code	Comment
.MODEL SMALL	::
.STACK 100H	::
.DATA	:: Khai báo biến
STR1 DB "A= \$"	::
STR2 DB 10,13, "B= \$"	::
STR3 DB 10,13, "FIBO: \$"	::
X DW ?	:: X là N đề bài yêu cầu
S1 DB	::
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	::
S2 DB	::
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1	::
S3 DB	::
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	::
.CODE	::
;;;;;MAIN;,,,,,,,,,,,,,,,,;	::
MAIN PROC	::
MOV AX, @DATA	:: Lưu địa chỉ DATA vào AX
MOV DS, AX	:: DS = AX
MOV AH, 9	:: AH = 9
LEA DX, STR1	:: DX = OFFSET STR1
INT 21H	:: In ra màn hình STR1
CALL INPUT	:: Gọi Input
PUSH BX	:: Lưu BX vào Stack

MOV AH, 9	::
LEA DX, STR2	::
INT 21H	::
CALL INPUT	::
POP CX	:: Lấy đỉnh Stack lưu vào CX
ADD BX,CX	:: BX = BX + CX
MOV X,BX	:: X = BX
MOV AH, 9	::
LEA DX, STR3	::
INT 21H	:: In ra màn hình STR3
CALL PRINT_ENTER	:: Gọi PRINT_ENTER
FIBO:	:: Hàm xử lí Fibonacci
MOV AX,X	:: AX = X
SUB AX,1	:: AX = AX – 1
MOV X,AX	:: X = AX
CALL S1_TO_S3   ;;S3 = S1	:: Gọi S1_TO_S3
CALL ADD_FIBO   ;;S1 = S1 + S2	::
CALL S3_TO_S2   ;;S2 = S3	::
CALL PRINT_FIBO   ;;PRINT S1	::
CALL PRINT_ENTER	::
MOV CX,X	:: CX = X
ADD CX,1	:: CX = CX + 1
LOOP FIBO	:: CX = CX – 1. Lặp FIBO đến khi CX=0
JMP END_PROGRAM	:: Nhảy tới END_PROGRAM
MAIN ENDP	::
;;;;;;;;;NHAP SO;;;;;;;;;	::
INPUT PROC	::
MOV BX,0	:: BX = 0
LOOP1:	::
MOV AH,1	:: AH = 1
INT 21H	:: Nhập số từ bàn phím

CMP AL,13	::AL = 13 thì cờ ZF = 1. Hai dòng này có
JE EXIT	nghĩa khi nhấn Enter thì sẽ kết thúc nhập
SUB AL,30H	::AL = AL – 30H
MOV AH,0	::AH = 0
PUSH AX	:: Lưu AX vào Stack
MOV AX, 10	:: AX = 10
MUL BX	:: AX = AX x BX
MOV BX, AX	:: BX = AX
POP AX	:: Lấy đỉnh Stack lưu vào AX
ADD BX,AX	:: BX = BX + AX
JMP LOOP1	::
EXIT:	::
RET	::
INPUT ENDP	::
;;;;;;;;;S3 = S1;;;;;;;;;;	::
S1_TO_S3 PROC	::
MOV CX,21	Procedure thực hiện lưu các phần tử của
S1_1:	mảng S1 vào mảng S3
MOV SI,CX	
MOV BL,S1[SI]	
MOV S3[SI],BL	
LOOP S1_1	
RET	
S1_TO_S3 ENDP	
;;;;;;;;;S2 = S3;;;;;;;;;;	
S3_TO_S2 PROC	::
MOV CX,21	Procedure thực hiện lưu các phần tử của
S2_1:	mảng S3 vào mảng S2
MOV SI,CX	
MOV BL,S3[SI]	
MOV S2[SI],BL	

<pre> LOOP S2_1 RET S3_TO_S2 ENDP ;;;;;;;;;S1+S2;;;;;;;;;; ADD_FIBO PROC MOV CX,21 MOV AX,0 ADD_1: MOV BL,0 MOV SI,CX ADD BL,AL ADD BL,S1[SI] ADD BL,S2[SI] MOV AH,0 MOV AL,BL MOV BL,10 DIV BL MOV S1[SI],AH LOOP ADD_1 RET ADD_FIBO ENDP ;;;;;;;;;PRINT;;;;;;;;;; PRINT_FIBO PROC MOV CX,21 MOV SI,0 PRINT: MOV DL,S1[SI] ADD DL,48 MOV AH,2 INT 21H INC SI </pre>	<pre> ;; </pre> <p>Procedure thực hiện tính tổng các phần tử của mảng S1 với S2 theo chỉ số tương ứng. Kết quả được lưu vào mảng S1</p> <pre> ;; </pre> <p>Procedure thực hiện in các phần tử của mảng S1</p>
--	---

LOOP PRINT RET PRINT_FIBO ENDP ;;;;;;;;;ENTER;;;;;;;;;; PRINT_ENTER PROC MOV DL,13 MOV AH,2 INT 21H MOV DL,10 MOV AH,2 INT 21H RET PRINT_ENTER ENDP ;;;;;;;;;END;;;;;;;;;; END_PROGRAM: END	;; Xuống dòng         ;; ;; Kết thúc chương trình
--	--