PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS

Instructor: The Tung Than

Student's name: Lê Hữu Đạt

Student code: 21520697

PRACTICE REPORT NO #5:

ADDITION OF TWO 32-BIT NUMBERS ON THE 8086 PROCESSOR

I. Content 1: Describe how sample code works

- Add two 32-bits numbers:

CODE	**Giải thích**
.MODEL SMALL	
.STACK 100H	Khai báo ngăn xếp kích thước 100H bytes
.DATA	Phần khai báo các biến dữ liệu
TB1 DB "S1= $"	Khai báo biến TB1 kiểu byte
TB2 DB 10,13, "S2= $"	Khai báo biến TB2 kiểu byte
TB3 DB 10,13, "S1+S2= $"	Khai báo biến TB3 kiểu byte
.CODE	Phần code
MAIN PROC	Hàm MAIN
MOV AX, @DATA	Gán AX = địa chỉ DATA
MOV DS, AX	Gán DS = AX
MOV AH, 9	Gán AH = 9
LEA DX, TB1	Gán DX = địa chỉ biến TB1
INT 21H	Xuất ra màn hình nội dung trong biến TB1
MOV CX, 16	Gán CX = 16
MOV BX, 0	Gán BX = 0
NHAP:	NHAP:
MOV AH, 1	Gán AH = 1
INT 21H	Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL
SUB AL, 30H	AL = AL – 30H

MOV AH, 0	Gán AH = 0
PUSH AX	Bỏ dữ liệu của thanh ghi AX vào ngăn xếp
MOV AX, 2	Gán AX = 2
MUL BX	AX = AX*BX
MOV BX, AX	BX = AX
POP AX	Lấy dữ liệu từ đầu ngăn xếp gán vào AX
ADD BX, AX	BX = BX + AX
LOOP NHAP	Lập NHAP 16 lần vì thanh ghi CX = 16
PUSH BX	Bỏ dữ liệu thanh ghi BX vào ngăn xếp
MOV BX, 0	Gán BX = 0
MOV CX, 16	Gán CX = 0
JMP NHAP1	Nhảy đến NHAP1
NHAP1:	NHAP1:
MOV AH, 1	Gán AH = 1
INT 21H	Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL
SUB AL, 30H	AL = AL – 30H
MOV AH, 0	Gán AH = 0
PUSH AX	Bỏ dữ liệu của thanh ghi AX vào ngăn xếp
MOV AX, 2	Gán AX = 2
MUL BX	AX = AX*BX
MOV BX, AX	BX = AX
POP AX	Lấy dữ liệu từ đầu ngăn xếp gán vào AX
ADD BX, AX	BX = BX + AX
LOOP NHAP1	Lập NHAP1 16 lần vì thanh ghi CX = 16
PUSH BX	Bỏ dữ liệu thanh ghi BX vào ngăn xếp
JMP NHAP2	Nhảy đến NHAP1
NHAP2:	NHAP2:
MOV AX, @DATA	Gán AX = địa chỉ DATA
MOV DS, AX	Gán DS = AX
MOV AH, 9	Gán AH = 9
LEA DX, TB2	DX = địa chỉ biến TB2

INT 21H	Xuất ra màn hình nội dung trong biến TB2
MOV CX, 16	Gán CX = 16
MOV BX, 0	Gán BX = 0
NHAP2_1:	NHAP2_1:
MOV AH, 1	Gán AH = 1
INT 21H	Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL
SUB AL, 30H	AL = AL – 30H
MOV AH, 0	Gán AH = 0
PUSH AX	Bỏ dữ liệu của thanh ghi AX vào ngăn xếp
MOV AX, 2	Gán AX = 2
MUL BX	AX = AX*BX
MOV BX, AX	BX = AX
POP AX	Lấy dữ liệu từ đầu ngăn xếp gán vào AX
ADD BX, AX	BX = BX + AX
LOOP NHAP2_1	Lập NHAP2_1 16 lần vì thanh ghi CX = 16
PUSH BX	Bỏ dữ liệu thanh ghi BX vào ngăn xếp
MOV BX, 0	Gán BX = 0
MOV CX, 16	Gán CX = 0
JMP NHAP3	Nhảy đến NHAP3
NHAP3:	NHAP3:
MOV AH, 1	Gán AH = 1
INT 21H	Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL
SUB AL, 30H	AL = AL – 30H
MOV AH, 0	Gán AH = 0
PUSH AX	Bỏ dữ liệu của thanh ghi AX vào ngăn xếp
MOV AX, 2	Gán AX = 2
MUL BX	AX = AX*BX
MOV BX, AX	BX = AX
POP AX	Lấy dữ liệu từ đầu ngăn xếp gán vào AX
ADD BX, AX	BX = BX + AX
LOOP NHAP3	Lập NHAP3 16 lần vì thanh ghi CX = 16

PUSH BX	Bỏ dữ liệu thanh ghi BX vào ngăn xếp
CONG:	CONG:
MOV AX, @DATA	Gán AX = địa chỉ DATA
MOV DS, AX	Gán DS = AX
MOV AH, 9	Gán AH = 9
LEA DX, TB3	DX = địa chỉ biến TB3
INT 21H	Xuất ra màn hình nội dung trong biến TB2
POP DX	DX = dữ liệu đầu ngăn xếp
POP CX	CX = dữ liệu tiếp theo trong ngăn xếp
POP BX	BX = dữ liệu tiếp theo trong ngăn xếp
POP AX	AX = dữ liệu tiếp theo trong ngăn xếp
ADD BX, DX	BX = BX + DX
JC CARRY	Nhảy đến CARRY nếu Carry flag = 1
JMP NOT_CARRY	Nếu không thì nhảy đến không NOT_CARRY
CARRY:	CARRY:
ADD AX, 1	AX = AX + 1
JC CARRY2	Nhảy đến CARRY2 nếu Carry flag = 1
MOV DL, 0	Gán DL = 0
JMP NOT_CARRY	Nhảy đến NOT_CARRY
CARRY2:	CARRY2:
MOV DL, 1	Gán DL = 1
NOT_CARRY:	NOT_CARRY:
ADD AX, CX	AX = AX + CX
JC CARRY3	Nhảy CARRY3 nếu Carry flag = 1
MOV DL, 0	Gán DL = 0
JMP EXIT	Nhảy đến EXIT
CARRY3:	CARRY3:
MOV DL, 1	Gán DL = 1
EXIT:	EXIT:
PUSH BX	Bỏ dữ liệu thanh ghi BX vào ngăn xếp
PUSH AX	Bỏ dữ liệu thanh ghi AX vào ngăn xếp

MOV BL, DL	BL = DL
MOV AH, 2	AH = 2
MOV DL, '0'	DL = ‘0’
TEST BL, 1	BL and 1 và ZF = 1 nếu kết quả and = 0 và = 0 nếu kết quả and #0
JZ PRINT	Nhảy đến PRINT nếu ZF = 1
MOV DL, '1'	DL = ‘1’
PRINT:	PRINT:
INT 21H	Ngắt 21H
POP BX	BX = dữ liệu ở đầu ngăn xếp
MOV CX, 16	CX = 16
PRINT1:	PRINT1:
MOV AH, 2	AH = 2
MOV DL, '0'	DL = ‘0’
TEST BX, 8000H	BX and 8000H nếu kết quả and = 0 và = 0 nếu kết quả and #0
JZ PRINT2	Nhảy đến PRINT2 nếu ZF = 1
MOV DL, '1'	DL = ‘1’
PRINT2:	PRINT2:
INT 21H	Ngắt 21H
SHL BX, 1	BX = BX << 1
LOOP PRINT1	Lặp lại PRINT1 16 lần
POP BX	Lấy dữ liệu ở đầu ngăn xếp gán vào BX
MOV CX, 16	CX = 16
PRINT3:	PRINT3
MOV AH, 2	AH = 2
MOV DL, '0'	DL = ‘0’
TEST BX, 8000H	BX and 8000H nếu kết quả and = 0 và = 0 nếu kết quả and #0
JZ PRINT4	Nhảy đến PRINT4 nếu ZF = 1
MOV DL, '1'	DL = ‘1’
PRINT4:	PRINT4
INT 21H	Ngắt 21H
SHL BX, 1	BX = BX << 1

LOOP PRINT3	Lặp lại PRINT3 16 lần
MAIN ENDP	Kết thúc chương trình
ENDP MAIN	

- Sub two 32-bits numbers:

CODE	**Giải thích**
.MODEL SMALL	
.STACK 100H	Khai báo ngăn xếp kích thước 100H bytes
.DATA	Phần khai báo các biến dữ liệu
TB1 DB "S1= $"	Khai báo biến TB1 kiểu byte
TB2 DB 10,13, "S2= $"	Khai báo biến TB2 kiểu byte
TB3 DB 10,13, "S1-S2= $"	Khai báo biến TB3 kiểu byte
.CODE	Phần code
MAIN PROC	Hàm MAIN
MOV AX, @DATA	Gán AX = địa chỉ DATA
MOV DS, AX	Gán DS = AX
MOV AH, 9	Gán AH = 9
LEA DX, TB1	Gán DX = địa chỉ biến TB1
INT 21H	Xuất ra màn hình nội dung trong biến TB1
MOV CX, 16	Gán CX = 16
MOV BX, 0	Gán BX = 0
NHAP:	NHAP:
MOV AH, 1	Gán AH = 1
INT 21H	Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL
SUB AL, 30H	AL = AL – 30H
MOV AH, 0	Gán AH = 0
PUSH AX	Bỏ dữ liệu của thanh ghi AX vào ngăn xếp
MOV AX, 2	Gán AX = 2
MUL BX	AX = AX*BX
MOV BX, AX	BX = AX
POP AX	Lấy dữ liệu từ đầu ngăn xếp gán vào AX

ADD BX, AX	BX = BX + AX
LOOP NHAP	Lập NHAP 16 lần vì thanh ghi CX = 16
PUSH BX	Bỏ dữ liệu thanh ghi BX vào ngăn xếp
MOV BX, 0	Gán BX = 0
MOV CX, 16	Gán CX = 0
JMP NHAP1	Nhảy đến NHAP1
NHAP1:	NHAP1:
MOV AH, 1	Gán AH = 1
INT 21H	Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL
SUB AL, 30H	AL = AL – 30H
MOV AH, 0	Gán AH = 0
PUSH AX	Bỏ dữ liệu của thanh ghi AX vào ngăn xếp
MOV AX, 2	Gán AX = 2
MUL BX	AX = AX*BX
MOV BX, AX	BX = AX
POP AX	Lấy dữ liệu từ đầu ngăn xếp gán vào AX
ADD BX, AX	BX = BX + AX
LOOP NHAP1	Lập NHAP1 16 lần vì thanh ghi CX = 16
PUSH BX	Bỏ dữ liệu thanh ghi BX vào ngăn xếp
JMP NHAP2	Nhảy đến NHAP1
NHAP2:	NHAP2:
MOV AX, @DATA	Gán AX = địa chỉ DATA
MOV DS, AX	Gán DS = AX
MOV AH, 9	Gán AH = 9
LEA DX, TB2	DX = địa chỉ biến TB2
INT 21H	Xuất ra màn hình nội dung trong biến TB2
MOV CX, 16	Gán CX = 16
MOV BX, 0	Gán BX = 0
NHAP2_1:	NHAP2_1:
MOV AH, 1	Gán AH = 1
INT 21H	Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL

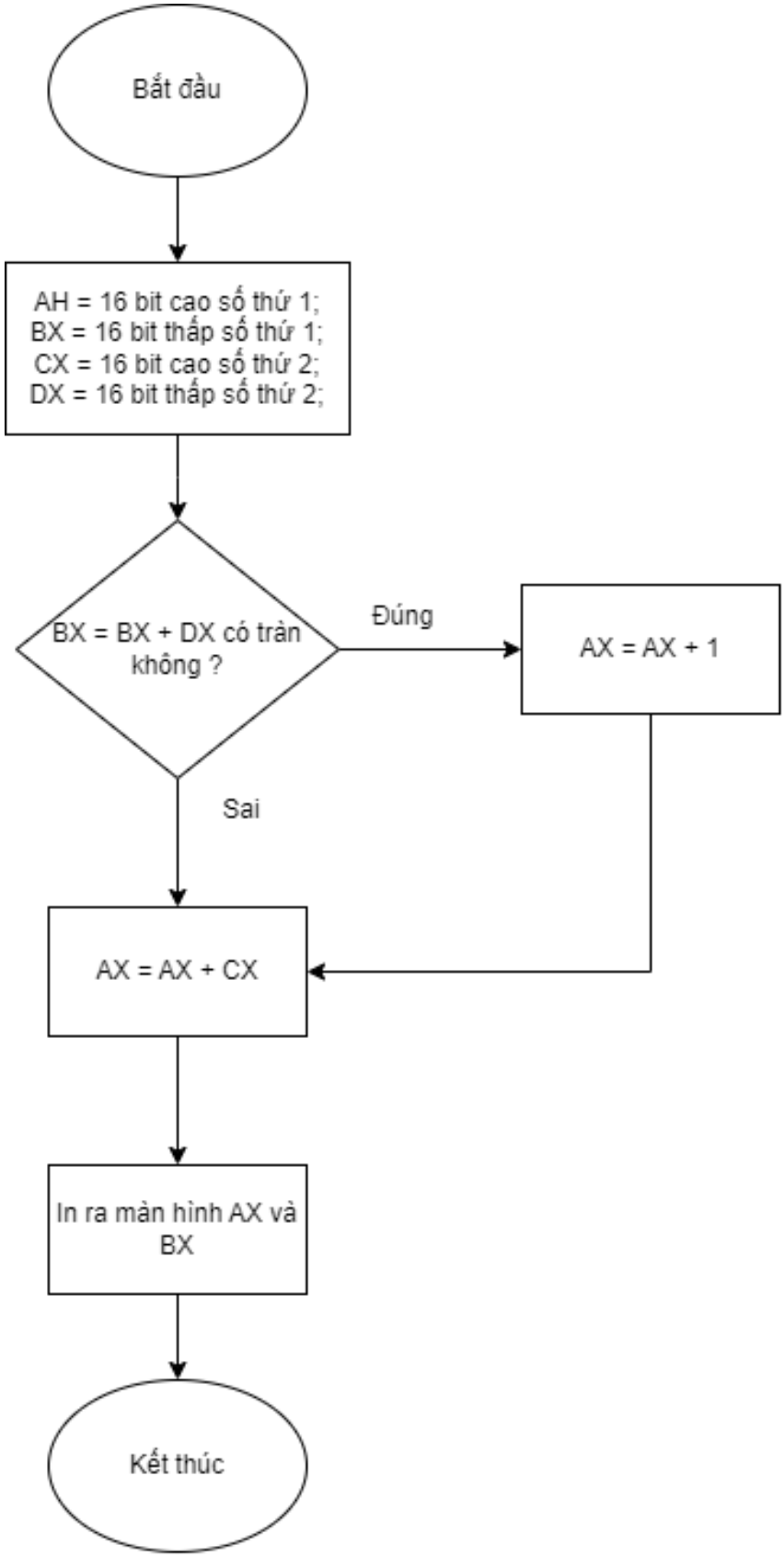
SUB AL, 30H	AL = AL – 30H
MOV AH, 0	Gán AH = 0
PUSH AX	Bỏ dữ liệu của thanh ghi AX vào ngăn xếp
MOV AX, 2	Gán AX = 2
MUL BX	AX = AX*BX
MOV BX, AX	BX = AX
POP AX	Lấy dữ liệu từ đầu ngăn xếp gán vào AX
ADD BX, AX	BX = BX + AX
LOOP NHAP2_1	Lập NHAP2_1 16 lần vì thanh ghi CX = 16
PUSH BX	Bỏ dữ liệu thanh ghi BX vào ngăn xếp
MOV BX, 0	Gán BX = 0
MOV CX, 16	Gán CX = 0
JMP NHAP3	Nhảy đến NHAP3
NHAP3:	NHAP3:
MOV AH, 1	Gán AH = 1
INT 21H	Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL
SUB AL, 30H	AL = AL – 30H
MOV AH, 0	Gán AH = 0
PUSH AX	Bỏ dữ liệu của thanh ghi AX vào ngăn xếp
MOV AX, 2	Gán AX = 2
MUL BX	AX = AX*BX
MOV BX, AX	BX = AX
POP AX	Lấy dữ liệu từ đầu ngăn xếp gán vào AX
ADD BX, AX	BX = BX + AX
LOOP NHAP3	Lập NHAP3 16 lần vì thanh ghi CX = 16
PUSH BX	Bỏ dữ liệu thanh ghi BX vào ngăn xếp
TRU:	TRU:
MOV AX, @DATA	Gán AX = địa chỉ DATA
MOV DS, AX	Gán DS = AX
MOV AH, 9	Gán AH = 9
LEA DX, TB3	DX = địa chỉ biến TB3

INT 21H	Xuất ra màn hình nội dung trong biến TB2
POP DX	DX = dữ liệu đầu ngăn xếp
POP CX	CX = dữ liệu tiếp theo trong ngăn xếp
POP BX	BX = dữ liệu tiếp theo trong ngăn xếp
POP AX	AX = dữ liệu tiếp theo trong ngăn xếp
SUB BX, DX	BX = BX - DX
JC CARRY	Nhảy đến CARRY nếu Carry flag = 1
JMP NOT_CARRY	Nếu không thì nhảy đến không NOT_CARRY
CARRY:	CARRY:
SUB AX, 1	AX = AX - 1
JC CARRY2	Nhảy đến CARRY2 nếu Carry flag = 1
MOV DL, 0	Gán DL = 0
JMP NOT_CARRY	Nhảy đến NOT_CARRY
CARRY2:	CARRY2:
MOV DL, 1	Gán DL = 1
NOT_CARRY:	NOT_CARRY:
SUB AX, CX	AX = AX - CX
JC CARRY3	Nhảy CARRY3 nếu Carry flag = 1
MOV DL, 0	Gán DL = 0
JMP END	Nhảy đến EXIT
CARRY3:	CARRY3:
MOV DL, 1	Gán DL = 1
EXIT:	EXIT:
PUSH BX	Bỏ dữ liệu thanh ghi BX vào ngăn xếp
PUSH AX	Bỏ dữ liệu thanh ghi AX vào ngăn xếp
MOV BL, DL	BL = DL
MOV AH, 2	AH = 2
MOV DL, '0'	DL = ‘0’
TEST BL, 1	BL and 1 và ZF = 1 nếu kết quả and = 0 và = 0 nếu kết quả and #0
JZ ZERO	Nhảy đến PRINT nếu ZF = 1
MOV DL, '1'	DL = ‘1’

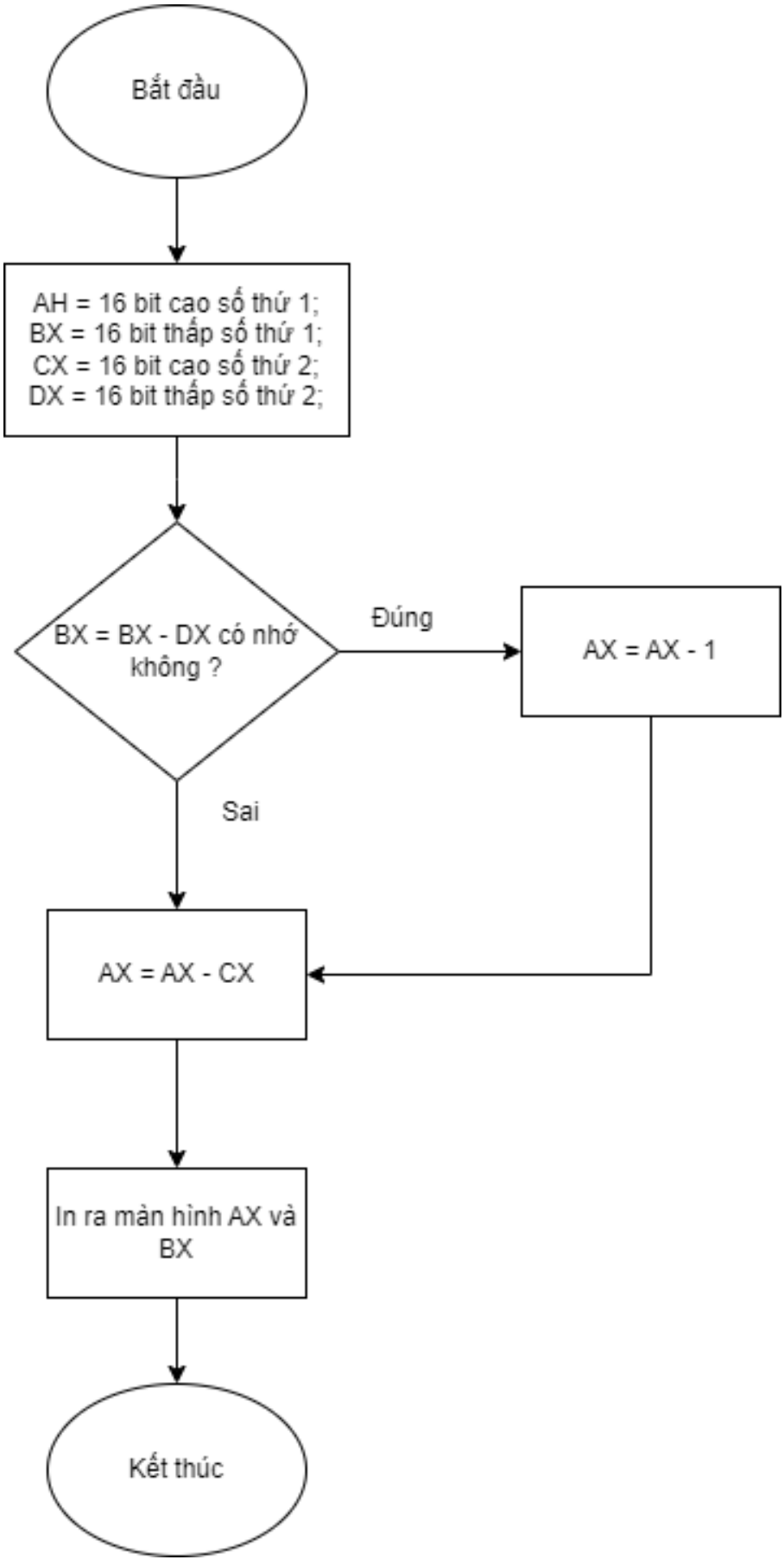
PRINT:	PRINT:
INT 21H	Ngắt 21H
POP BX	BX = dữ liệu ở đầu ngăn xếp
MOV CX, 16	CX = 16
PRINT1:	PRINT1:
MOV AH, 2	AH = 2
MOV DL, '0'	DL = ‘0’
TEST BX, 8000H	BX and 8000H nếu kết quả and = 0 và = 0 nếu kết quả and #0
JZ PRINT2	Nhảy đến PRINT2 nếu ZF = 1
MOV DL, '1'	DL = ‘1’
PRINT2:	PRINT2:
INT 21H	Ngắt 21H
SHL BX, 1	BX = BX << 1
LOOP PRINT1	Lặp lại PRINT1 16 lần
POP BX	Lấy dữ liệu ở đầu ngăn xếp gán vào BX
MOV CX, 16	CX = 16
PRINT3:	PRINT3
MOV AH, 2	AH = 2
MOV DL, '0'	DL = ‘0’
TEST BX, 8000H	BX and 8000H nếu kết quả and = 0 và = 0 nếu kết quả and #0
JZ PRINT4	Nhảy đến PRINT4 nếu ZF = 1
MOV DL, '1'	DL = ‘1’
PRINT4:	PRINT4
INT 21H	Ngắt 21H
SHL BX, 1	BX = BX << 1
LOOP PRINT3	Lặp lại PRINT3 16 lần
MAIN ENDP	Kết thúc chương trình
ENDP MAIN	

II. Content 2: Flowchart of the program algorithm to add and sub two 32-bit numbers.

- Add two 32-bits numbers:



- Sub two 32-bits numbers:



III. Content 3: Explain the operating principle of the effects, accompanied by a video (send a Google Drive link) to demonstrate the circuit operation in case the instructor cannot run the design file.

- Add two 32-bits numbers: https://drive.google.com/file/d/1bo0Gwmo_slhsD3nfbfQfClHYL_QfZ7AC/view?usp=sharing

- Sub two 32-bits numbers: https://drive.google.com/file/d/19Npv87YHZ3dkP-Hfm3FzsOXhaJJgJtIB/view?usp=sharing