

PRACTICE EXERCISES OF THE MICROPROCESSORS & MICROCONTROLLERS

Instructor: The Tung Than

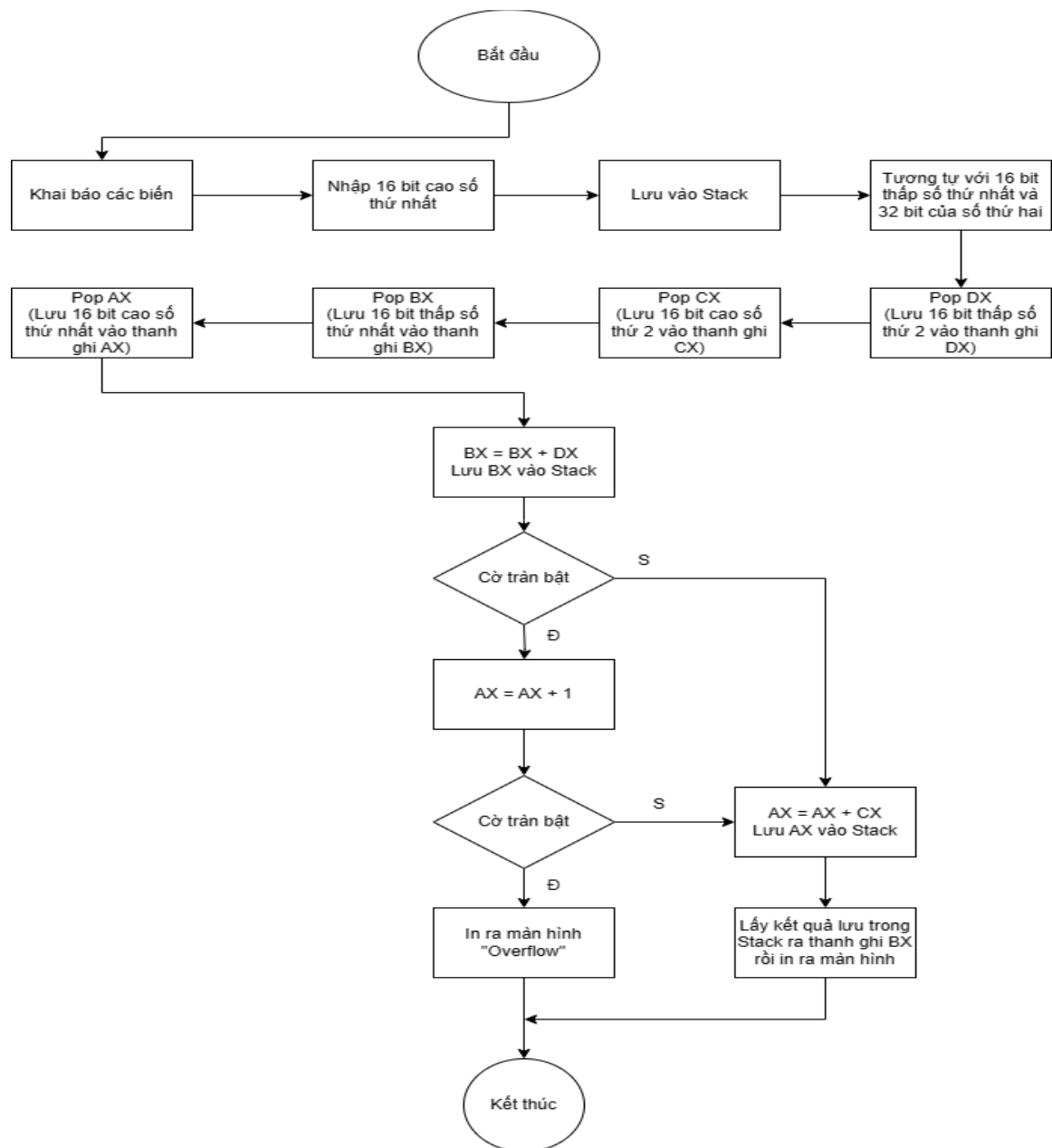
Student's name: Trần Lê Minh Đăng

Student code: 21520684

PRACTICE REPORT NO 5

ADDITION OF TWO 32-BIT NUMBERS ON THE 8086 PROCESSOR

I. Flowchart of the program algorithm to add two 32-bit numbers.



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/1FC9OfZ9IOX0agB_rgpHXGZzKj3MOXIV8?usp=sharing

SOURCE CODE:

| Code | Giải thích |
|--------------------------|------------------------------------|
| .MODEL SMALL | :: |
| .STACK 100H | Khai báo các biến |
| .DATA | :: |
| STR1 DB "A= \$" | :: |
| STR2 DB 10,13, "B= \$" | :: |
| STR3 DB 10,13, "A+B= \$" | :: |
| STR4 DB "Overflow \$" | :: |
| .CODE | :: |
| MAIN: | :: Chương trình chính |
| MOV AX, @DATA | :: Lấy địa chỉ của DATA lưu vào AX |
| MOV DS, AX | :: DS = AX |
| MOV DX, OFFSET STR1 | :: DX = OFFSET STR1 |
| MOV AH, 9 | :: AH = 9 |
| INT 21H | :: In ra màn hình STR1 |
| CALL INPUT | :: Gọi INPUT |
| PUSH BX | :: Lưu BX vào Stack |
| CALL INPUT | |
| PUSH BX | |
| MOV DX, OFFSET STR2 | |
| MOV AH, 9 | |
| INT 21H | |
| CALL INPUT | |
| PUSH BX | |

| | |
|---------------------|--------------------------------|
| CALL INPUT | |
| PUSH BX | |
| MOV DX, OFFSET STR3 | |
| MOV AH, 9 | |
| INT 21H | |
| POP DX | ::Lấy đỉnh Stack lưu vào DX |
| POP CX | |
| POP BX | |
| POP AX | |
| ADD BX,DX | ::BX = BX + DX |
| PUSH BX | ::Lưu BX vào Stack |
| JC CARRY | ::Cờ Carry = 1 nhảy tới CARRY |
| JMP NOT_CARRY | ::Nhảy tới NOT_CARRY |
| CARRY: | ::Hàm CARRY |
| ADD AX,1 | ::AX = AX + 1 |
| JC CARRY2 | ::Cờ Carry = 1 nhảy tới CARRY2 |
| JMP NOT_CARRY | |
| NOT_CARRY: | ::Hàm NOT_CARRY |
| ADD AX,CX | ::AX = AX + CX |
| JC CARRY2 | |
| PUSH AX | ::Lưu AX vào Stack |
| POP BX | ::Lấy đỉnh Stack lưu vào BX |
| CALL PRINT | ::Gọi PRINT |
| POP BX | |
| CALL PRINT | |
| JMP EXIT | ::Nhảy tới EXIT |
| CARRY2: | ::Hàm CARRY2 |
| MOV AX, @DATA | |
| MOV DS, AX | |
| MOV DX, OFFSET STR4 | |
| MOV AH, 9 | |

| | |
|---------------|--------------------------------------|
| INT 21H | |
| JMP EXIT | |
| | |
| INPUT PROC | ::Procedure INPUT |
| MOV CX, 16 | ::CX = 16 |
| MOV BX, 0 | ::BX = 0 |
| INPUT1: | |
| MOV AH, 1 | ::AH = 1 |
| INT 21H | ::Đọc kí tự được nhập từ bàn phím |
| SUB AL, 30H | ::AL = AL – 30H |
| MOV AH, 0 | ::AH = 0 |
| PUSH AX | |
| MOV AX, 2 | ::AX = 2 |
| MUL BX | ::AX = BX x AX |
| MOV BX, AX | ::BX = AX |
| POP AX | |
| ADD BX, AX | ::BX = BX + AX |
| LOOP INPUT1 | ::Lặp INPUT1 đến khi CX = 0 |
| RET | ::Trở lại địa chỉ gọi |
| INPUT ENDP | ::Kết thúc |
| | |
| PRINT PROC | ::Procedure PRINT |
| MOV CX,16 | ::CX = 16 |
| PRINT1: | |
| MOV AH, 2 | ::AH = 2 |
| MOV DL, '0' | ::DL = '0' |
| TEST BX,8000H | ::BX = BX and 8000H, ZF = 1 nếu BX=0 |
| JZ ZERO | ::ZF = 1 nhảy tới ZERO |
| MOV DL, '1' | ::DL = '1' |
| ZERO: | |
| INT 21H | ::In kí tự ra màn hình |

| | |
|-------------|---|
| SHL BX, 1 | ::Dịch trái 1 bit |
| LOOP PRINT1 | ::Lặp PRINT1 đến khi CX = 0 (tức là 16 lần) |
| RET | |
| PRINT ENDP | |
| | |
| EXIT: | |
| END | ::Kết thúc chương trình |

III. Exercise: Write a program to subtract 2 32-bit numbers

| Code | Giải thích |
|--------------------------|-----------------------------------|
| .MODEL SMALL | :: |
| .STACK 100H | Khai báo các biến |
| .DATA | :: |
| STR1 DB "A= \$" | :: |
| STR2 DB 10,13, "B= \$" | :: |
| STR3 DB 10,13, "A-B= \$" | :: |
| STR4 DB "Overflow \$" | :: |
| .CODE | :: |
| MAIN: | :: Chương trình chính |
| MOV AX, @DATA | ::Lấy địa chỉ của DATA lưu vào AX |
| MOV DS, AX | ::DS = AX |
| MOV DX, OFFSET STR1 | ::DX = OFFSET STR1 |
| MOV AH, 9 | ::AH = 9 |
| INT 21H | ::In ra màn hình STR1 |
| CALL INPUT | ::Gọi INPUT |
| PUSH BX | ::Lưu BX vào Stack |
| CALL INPUT | |
| PUSH BX | |

| | |
|---------------------|--------------------------------|
| MOV DX, OFFSET STR2 | |
| MOV AH, 9 | |
| INT 21H | |
| CALL INPUT | |
| PUSH BX | |
| CALL INPUT | |
| PUSH BX | |
| MOV DX, OFFSET STR3 | |
| MOV AH, 9 | |
| INT 21H | |
| POP DX | ::Lấy đỉnh Stack lưu vào DX |
| POP CX | |
| POP BX | |
| POP AX | |
| SUB BX,DX | ::BX = BX – DX |
| PUSH BX | |
| JC CARRY | ::Cờ Carry = 1 nhảy tới CARRY |
| JMP NOT_CARRY | ::Nhảy tới NOT_CARRY |
| CARRY: | ::Hàm CARRY |
| ADD CX,1 | ::CX = CX + 1 |
| JC CARRY2 | ::Cờ Carry = 1 nhảy tới CARRY2 |
| JMP NOT_CARRY | |
| NOT_CARRY: | ::Hàm NOT_CARRY |
| SUB AX,CX | ::AX = AX - CX |
| JC CARRY2 | |
| PUSH AX | ::Lưu AX vào Stack |
| POP BX | ::Lấy đỉnh Stack lưu vào BX |
| CALL PRINT | ::Gọi PRINT |
| POP BX | |
| CALL PRINT | |
| JMP EXIT | ::Nhảy tới EXIT |

| | |
|---------------------|-----------------------------------|
| CARRY2: | ::Hàm CARRY2 |
| MOV AX, @DATA | |
| MOV DS, AX | |
| MOV DX, OFFSET STR4 | |
| MOV AH, 9 | |
| INT 21H | |
| JMP EXIT | |
| | |
| INPUT PROC | ::Procedure INPUT |
| MOV CX, 16 | ::CX = 16 |
| MOV BX, 0 | ::BX = 0 |
| INPUT1: | |
| MOV AH, 1 | ::AH = 1 |
| INT 21H | ::Đọc kí tự được nhập từ bàn phím |
| SUB AL, 30H | ::AL = AL – 30H |
| MOV AH, 0 | ::AH = 0 |
| PUSH AX | |
| MOV AX, 2 | ::AX = 2 |
| MUL BX | ::AX = BX x AX |
| MOV BX, AX | ::BX = AX |
| POP AX | |
| ADD BX, AX | ::BX = BX + AX |
| LOOP INPUT1 | ::Lặp INPUT1 đến khi CX = 0 |
| RET | ::Trở lại địa chỉ gọi |
| INPUT ENDP | ::Kết thúc |
| | |
| PRINT PROC | ::Procedure PRINT |
| MOV CX,16 | ::CX = 16 |
| PRINT1: | |
| MOV AH, 2 | ::AH = 2 |
| MOV DL, '0' | ::DL = '0' |

| | |
|---------------|---|
| TEST BX,8000H | ::BX = BX and 8000H, ZF = 1 nếu BX=0 |
| JZ ZERO | ::ZF = 1 nhảy tới ZERO |
| MOV DL, '1' | ::DL = '1' |
| ZERO: | |
| INT 21H | ;In kí tự ra màn hình |
| SHL BX, 1 | ::Dịch trái 1 bit |
| LOOP PRINT1 | ::Lặp PRINT1 đến khi CX = 0 (tức là 16 lần) |
| RET | |
| PRINT ENDP | |
| | |
| EXIT: | |
| END | ::Kết thúc chương trình |