|  |
| --- |
| 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  .DATA  TB1 DB "S1= $"  TB2 DB 10,13, "S2= $"  TB3 DB 10,13, "S1+S2= $"  .CODE  MAIN PROC  MOV AX, @DATA  MOV DS, AX  MOV AH, 9  LEA DX, TB1  INT 21H  MOV CX, 16  MOV BX, 0  NHAP:  MOV AH, 1  INT 21H  SUB AL, 30H  MOV AH, 0  PUSH AX  MOV AX, 2  MUL BX  MOV BX, AX  POP AX  ADD BX, AX  LOOP NHAP  PUSH BX  MOV BX, 0  MOV CX, 16  JMP NHAP1  NHAP1:  MOV AH, 1  INT 21H  SUB AL, 30H  MOV AH, 0  PUSH AX  MOV AX, 2  MUL BX  MOV BX, AX  POP AX  ADD BX, AX  LOOP NHAP1  PUSH BX  JMP NHAP2  NHAP2:  MOV AX, @DATA  MOV DS, AX  MOV AH, 9  LEA DX, TB2  INT 21H  MOV CX, 16  MOV BX, 0  NHAP2\_1:  MOV AH, 1  INT 21H  SUB AL, 30H  MOV AH, 0  PUSH AX  MOV AX, 2  MUL BX  MOV BX, AX  POP AX  ADD BX, AX  LOOP NHAP2\_1  PUSH BX  MOV BX, 0  MOV CX, 16  JMP NHAP3  NHAP3:  MOV AH, 1  INT 21H  SUB AL, 30H  MOV AH, 0  PUSH AX  MOV AX, 2  MUL BX  MOV BX, AX  POP AX  ADD BX, AX  LOOP NHAP3  PUSH BX  CONG:  MOV AX, @DATA  MOV DS, AX  MOV AH, 9  LEA DX, TB3  INT 21H  POP DX  POP CX  POP BX  POP AX  ADD BX, DX  JC CARRY  JMP NOT\_CARRY  CARRY:  ADD AX, 1  JC CARRY2  MOV DL, 0  JMP NOT\_CARRY  CARRY2:  MOV DL, 1  NOT\_CARRY:  ADD AX, CX  JC CARRY3  MOV DL, 0  JMP EXIT  CARRY3:  MOV DL, 1  EXIT:  PUSH BX  PUSH AX  MOV BL, DL  MOV AH, 2  MOV DL, '0'  TEST BL, 1  JZ PRINT  MOV DL, '1'  PRINT:  INT 21H  POP BX  MOV CX, 16  PRINT1:  MOV AH, 2  MOV DL, '0'  TEST BX, 8000H  JZ PRINT2  MOV DL, '1'  PRINT2:  INT 21H  SHL BX, 1  LOOP PRINT1  POP BX  MOV CX, 16  PRINT3:  MOV AH, 2  MOV DL, '0'  TEST BX, 8000H  JZ PRINT4  MOV DL, '1'  PRINT4:  INT 21H  SHL BX, 1  LOOP PRINT3  MAIN ENDP  ENDP MAIN | Khai báo ngăn xếp kích thước 100H bytes  Phần khai báo các biến dữ liệu  Khai báo biến TB1 kiểu byte  Khai báo biến TB2 kiểu byte  Khai báo biến TB3 kiểu byte  Phần code  Hàm MAIN  Gán AX = địa chỉ DATA  Gán DS = AX  Gán AH = 9  Gán DX = địa chỉ biến TB1  Xuất ra màn hình nội dung trong biến TB1  Gán CX = 16  Gán BX = 0  NHAP:  Gán AH = 1  Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL  AL = AL – 30H  Gán AH = 0  Bỏ dữ liệu của thanh ghi AX vào ngăn xếp  Gán AX = 2  AX = AX\*BX  BX = AX  Lấy dữ liệu từ đầu ngăn xếp gán vào AX  BX = BX + AX  Lập NHAP 16 lần vì thanh ghi CX = 16  Bỏ dữ liệu thanh ghi BX vào ngăn xếp  Gán BX = 0  Gán CX = 0  Nhảy đến NHAP1  NHAP1:  Gán AH = 1  Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL  AL = AL – 30H  Gán AH = 0  Bỏ dữ liệu của thanh ghi AX vào ngăn xếp  Gán AX = 2  AX = AX\*BX  BX = AX  Lấy dữ liệu từ đầu ngăn xếp gán vào AX  BX = BX + AX  Lập NHAP1 16 lần vì thanh ghi CX = 16  Bỏ dữ liệu thanh ghi BX vào ngăn xếp  Nhảy đến NHAP1  NHAP2:  Gán AX = địa chỉ DATA  Gán DS = AX  Gán AH = 9  DX = địa chỉ biến TB2  Xuất ra màn hình nội dung trong biến TB2  Gán CX = 16  Gán BX = 0  NHAP2\_1:  Gán AH = 1  Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL  AL = AL – 30H  Gán AH = 0  Bỏ dữ liệu của thanh ghi AX vào ngăn xếp  Gán AX = 2  AX = AX\*BX  BX = AX  Lấy dữ liệu từ đầu ngăn xếp gán vào AX  BX = BX + AX  Lập NHAP2\_1 16 lần vì thanh ghi CX = 16  Bỏ dữ liệu thanh ghi BX vào ngăn xếp  Gán BX = 0  Gán CX = 0  Nhảy đến NHAP3  NHAP3:  Gán AH = 1  Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL  AL = AL – 30H  Gán AH = 0  Bỏ dữ liệu của thanh ghi AX vào ngăn xếp  Gán AX = 2  AX = AX\*BX  BX = AX  Lấy dữ liệu từ đầu ngăn xếp gán vào AX  BX = BX + AX  Lập NHAP3 16 lần vì thanh ghi CX = 16  Bỏ dữ liệu thanh ghi BX vào ngăn xếp  CONG:  Gán AX = địa chỉ DATA  Gán DS = AX  Gán AH = 9  DX = địa chỉ biến TB3  Xuất ra màn hình nội dung trong biến TB2  DX = dữ liệu đầu ngăn xếp  CX = dữ liệu tiếp theo trong ngăn xếp  BX = dữ liệu tiếp theo trong ngăn xếp  AX = dữ liệu tiếp theo trong ngăn xếp  BX = BX + DX  Nhảy đến CARRY nếu Carry flag = 1  Nếu không thì nhảy đến không NOT\_CARRY  CARRY:  AX = AX + 1  Nhảy đến CARRY2 nếu Carry flag = 1  Gán DL = 0  Nhảy đến NOT\_CARRY  CARRY2:  Gán DL = 1  NOT\_CARRY:  AX = AX + CX  Nhảy CARRY3 nếu Carry flag = 1  Gán DL = 0  Nhảy đến EXIT  CARRY3:  Gán DL = 1  EXIT:  Bỏ dữ liệu thanh ghi BX vào ngăn xếp  Bỏ dữ liệu thanh ghi AX vào ngăn xếp  BL = DL  AH = 2  DL = ‘0’  BL and 1 và ZF = 1 nếu kết quả and = 0 và = 0 nếu kết quả and #0  Nhảy đến PRINT nếu ZF = 1  DL = ‘1’  PRINT:  Ngắt 21H  BX = dữ liệu ở đầu ngăn xếp  CX = 16  PRINT1:  AH = 2  DL = ‘0’  BX and 8000H nếu kết quả and = 0 và = 0 nếu kết quả and #0  Nhảy đến PRINT2 nếu ZF = 1  DL = ‘1’  PRINT2:  Ngắt 21H  BX = BX << 1  Lặp lại PRINT1 16 lần  Lấy dữ liệu ở đầu ngăn xếp gán vào BX  CX = 16  PRINT3  AH = 2  DL = ‘0’  BX and 8000H nếu kết quả and = 0 và = 0 nếu kết quả and #0  Nhảy đến PRINT4 nếu ZF = 1  DL = ‘1’  PRINT4  Ngắt 21H  BX = BX << 1  Lặp lại PRINT3 16 lần  Kết thúc chương trình |

- Sub two 32-bits numbers:

|  |  |
| --- | --- |
| **CODE** | **Giải thích** |
| .MODEL SMALL  .STACK 100H  .DATA  TB1 DB "S1= $"  TB2 DB 10,13, "S2= $"  TB3 DB 10,13, "S1-S2= $"  .CODE  MAIN PROC  MOV AX, @DATA  MOV DS, AX  MOV AH, 9  LEA DX, TB1  INT 21H  MOV CX, 16  MOV BX, 0  NHAP:  MOV AH, 1  INT 21H  SUB AL, 30H  MOV AH, 0  PUSH AX  MOV AX, 2  MUL BX  MOV BX, AX  POP AX  ADD BX, AX  LOOP NHAP  PUSH BX  MOV BX, 0  MOV CX, 16  JMP NHAP1  NHAP1:  MOV AH, 1  INT 21H  SUB AL, 30H  MOV AH, 0  PUSH AX  MOV AX, 2  MUL BX  MOV BX, AX  POP AX  ADD BX, AX  LOOP NHAP1  PUSH BX  JMP NHAP2  NHAP2:  MOV AX, @DATA  MOV DS, AX  MOV AH, 9  LEA DX, TB2  INT 21H  MOV CX, 16  MOV BX, 0  NHAP2\_1:  MOV AH, 1  INT 21H  SUB AL, 30H  MOV AH, 0  PUSH AX  MOV AX, 2  MUL BX  MOV BX, AX  POP AX  ADD BX, AX  LOOP NHAP2\_1  PUSH BX  MOV BX, 0  MOV CX, 16  JMP NHAP3  NHAP3:  MOV AH, 1  INT 21H  SUB AL, 30H  MOV AH, 0  PUSH AX  MOV AX, 2  MUL BX  MOV BX, AX  POP AX  ADD BX, AX  LOOP NHAP3  PUSH BX  TRU:  MOV AX, @DATA  MOV DS, AX  MOV AH, 9  LEA DX, TB3  INT 21H  POP DX  POP CX  POP BX  POP AX  SUB BX, DX  JC CARRY  JMP NOT\_CARRY  CARRY:  SUB AX, 1  JC CARRY2  MOV DL, 0  JMP NOT\_CARRY  CARRY2:  MOV DL, 1  NOT\_CARRY:  SUB AX, CX  JC CARRY3  MOV DL, 0  JMP END  CARRY3:  MOV DL, 1  EXIT:  PUSH BX  PUSH AX  MOV BL, DL  MOV AH, 2  MOV DL, '0'  TEST BL, 1  JZ ZERO  MOV DL, '1'  PRINT:  INT 21H  POP BX  MOV CX, 16  PRINT1:  MOV AH, 2  MOV DL, '0'  TEST BX, 8000H  JZ PRINT2  MOV DL, '1'  PRINT2:  INT 21H  SHL BX, 1  LOOP PRINT1  POP BX  MOV CX, 16  PRINT3:  MOV AH, 2  MOV DL, '0'  TEST BX, 8000H  JZ PRINT4  MOV DL, '1'  PRINT4:  INT 21H  SHL BX, 1  LOOP PRINT3  MAIN ENDP  ENDP MAIN | Khai báo ngăn xếp kích thước 100H bytes  Phần khai báo các biến dữ liệu  Khai báo biến TB1 kiểu byte  Khai báo biến TB2 kiểu byte  Khai báo biến TB3 kiểu byte  Phần code  Hàm MAIN  Gán AX = địa chỉ DATA  Gán DS = AX  Gán AH = 9  Gán DX = địa chỉ biến TB1  Xuất ra màn hình nội dung trong biến TB1  Gán CX = 16  Gán BX = 0  NHAP:  Gán AH = 1  Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL  AL = AL – 30H  Gán AH = 0  Bỏ dữ liệu của thanh ghi AX vào ngăn xếp  Gán AX = 2  AX = AX\*BX  BX = AX  Lấy dữ liệu từ đầu ngăn xếp gán vào AX  BX = BX + AX  Lập NHAP 16 lần vì thanh ghi CX = 16  Bỏ dữ liệu thanh ghi BX vào ngăn xếp  Gán BX = 0  Gán CX = 0  Nhảy đến NHAP1  NHAP1:  Gán AH = 1  Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL  AL = AL – 30H  Gán AH = 0  Bỏ dữ liệu của thanh ghi AX vào ngăn xếp  Gán AX = 2  AX = AX\*BX  BX = AX  Lấy dữ liệu từ đầu ngăn xếp gán vào AX  BX = BX + AX  Lập NHAP1 16 lần vì thanh ghi CX = 16  Bỏ dữ liệu thanh ghi BX vào ngăn xếp  Nhảy đến NHAP1  NHAP2:  Gán AX = địa chỉ DATA  Gán DS = AX  Gán AH = 9  DX = địa chỉ biến TB2  Xuất ra màn hình nội dung trong biến TB2  Gán CX = 16  Gán BX = 0  NHAP2\_1:  Gán AH = 1  Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL  AL = AL – 30H  Gán AH = 0  Bỏ dữ liệu của thanh ghi AX vào ngăn xếp  Gán AX = 2  AX = AX\*BX  BX = AX  Lấy dữ liệu từ đầu ngăn xếp gán vào AX  BX = BX + AX  Lập NHAP2\_1 16 lần vì thanh ghi CX = 16  Bỏ dữ liệu thanh ghi BX vào ngăn xếp  Gán BX = 0  Gán CX = 0  Nhảy đến NHAP3  NHAP3:  Gán AH = 1  Cho phép người dùng nhập từ bàn phím vào và lưu ở thanh ghi AL  AL = AL – 30H  Gán AH = 0  Bỏ dữ liệu của thanh ghi AX vào ngăn xếp  Gán AX = 2  AX = AX\*BX  BX = AX  Lấy dữ liệu từ đầu ngăn xếp gán vào AX  BX = BX + AX  Lập NHAP3 16 lần vì thanh ghi CX = 16  Bỏ dữ liệu thanh ghi BX vào ngăn xếp  TRU:  Gán AX = địa chỉ DATA  Gán DS = AX  Gán AH = 9  DX = địa chỉ biến TB3  Xuất ra màn hình nội dung trong biến TB2  DX = dữ liệu đầu ngăn xếp  CX = dữ liệu tiếp theo trong ngăn xếp  BX = dữ liệu tiếp theo trong ngăn xếp  AX = dữ liệu tiếp theo trong ngăn xếp  BX = BX - DX  Nhảy đến CARRY nếu Carry flag = 1  Nếu không thì nhảy đến không NOT\_CARRY  CARRY:  AX = AX - 1  Nhảy đến CARRY2 nếu Carry flag = 1  Gán DL = 0  Nhảy đến NOT\_CARRY  CARRY2:  Gán DL = 1  NOT\_CARRY:  AX = AX - CX  Nhảy CARRY3 nếu Carry flag = 1  Gán DL = 0  Nhảy đến EXIT  CARRY3:  Gán DL = 1  EXIT:  Bỏ dữ liệu thanh ghi BX vào ngăn xếp  Bỏ dữ liệu thanh ghi AX vào ngăn xếp  BL = DL  AH = 2  DL = ‘0’  BL and 1 và ZF = 1 nếu kết quả and = 0 và = 0 nếu kết quả and #0  Nhảy đến PRINT nếu ZF = 1  DL = ‘1’  PRINT:  Ngắt 21H  BX = dữ liệu ở đầu ngăn xếp  CX = 16  PRINT1:  AH = 2  DL = ‘0’  BX and 8000H nếu kết quả and = 0 và = 0 nếu kết quả and #0  Nhảy đến PRINT2 nếu ZF = 1  DL = ‘1’  PRINT2:  Ngắt 21H  BX = BX << 1  Lặp lại PRINT1 16 lần  Lấy dữ liệu ở đầu ngăn xếp gán vào BX  CX = 16  PRINT3  AH = 2  DL = ‘0’  BX and 8000H nếu kết quả and = 0 và = 0 nếu kết quả and #0  Nhảy đến PRINT4 nếu ZF = 1  DL = ‘1’  PRINT4  Ngắt 21H  BX = BX << 1  Lặp lại PRINT3 16 lần  Kết thúc chương trình |

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

- Add two 32-bits numbers:

A picture containing diagram, text, technical drawing, drawing

Description automatically generated

- Sub two 32-bits numbers:

A picture containing diagram, text, drawing, sketch

Description automatically generated

**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