|  |
| --- |
| 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

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

A diagram of a flowchart

Description automatically generated with low confidence

1. **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_rgpHXGZzKj3MOXlV8?usp=sharing>

SOURCE CODE:

|  |  |
| --- | --- |
| Code | Giải thích |
| .MODEL SMALL  .STACK 100H  .DATA  STR1 DB "A= $"  STR2 DB 10,13, "B= $"  STR3 DB 10,13, "A+B= $"  STR4 DB "Overflow $"  .CODE  MAIN:  MOV AX, @DATA  MOV DS, AX  MOV DX, OFFSET STR1  MOV AH, 9  INT 21H  CALL INPUT  PUSH BX  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  POP CX  POP BX  POP AX  ADD BX,DX  PUSH BX  JC CARRY  JMP NOT\_CARRY  CARRY:  ADD AX,1  JC CARRY2  JMP NOT\_CARRY  NOT\_CARRY:  ADD AX,CX  JC CARRY2  PUSH AX  POP BX  CALL PRINT  POP BX  CALL PRINT  JMP EXIT  CARRY2:  MOV AX, @DATA  MOV DS, AX  MOV DX, OFFSET STR4  MOV AH, 9  INT 21H  JMP EXIT  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;  INPUT PROC  MOV CX, 16  MOV BX, 0  INPUT1:  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 INPUT1  RET  INPUT ENDP  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;  PRINT PROC  MOV CX,16  PRINT1:  MOV AH, 2  MOV DL, '0'  TEST BX,8000H  JZ ZERO  MOV DL, '1'  ZERO:  INT 21H  SHL BX, 1  LOOP PRINT1  RET  PRINT ENDP  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;  EXIT:  END | ;;  Khai báo các biến  ;;  ;;  ;;  ;;  ;;  ;;  ;; Chương trình chính  ;;Lấy địa chỉ của DATA lưu vào AX  ;;DS = AX  ;;DX = OFFSET STR1  ;;AH = 9  ;;In ra màn hình STR1  ;;Gọi INPUT  ;;Lưu BX vào Stack  ;;Lấy đỉnh Stack lưu vào DX  ;;BX = BX + DX  ;;Lưu BX vào Stack  ;;Cờ Carry = 1 nhảy tới CARRY  ;;Nhảy tới NOT\_CARRY  ;;Hàm CARRY  ;;AX = AX + 1  ;;Cờ Carry = 1 nhảy tới CARRY2  ;;Hàm NOT\_CARRY  ;;AX = AX + CX  ;;Lưu AX vào Stack  ;;Lấy đỉnh Stack lưu vào BX  ;;Gọi PRINT  ;;Nhảy tới EXIT  ;;Hàm CARRY2  ;;Procedure INPUT  ;;CX = 16  ;;BX = 0  ;;AH = 1  ;;Đọc kí tự được nhập từ bàn phím  ;;AL = AL – 30H  ;;AH = 0  ;;AX = 2  ;;AX = BX x AX  ;;BX = AX  ;;BX = BX + AX  ;;Lặp INPUT1 đến khi CX = 0  ;;Trở lại địa chỉ gọi  ;;Kết thúc  ;;Procedure PRINT  ;;CX = 16  ;;AH = 2  ;;DL = ‘0’  ;;BX = BX and 8000H, ZF = 1 nếu BX=0  ;;ZF = 1 nhảy tới ZERO  ;;DL = ‘1’  ;In kí tự ra màn hình  ;;Dịch trái 1 bit  ;;Lặp PRINT1 đến khi CX = 0 (tức là 16 lần)  ;;Kết thúc chương trình |

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

|  |  |
| --- | --- |
| Code | Giải thích |
| .MODEL SMALL  .STACK 100H  .DATA  STR1 DB "A= $"  STR2 DB 10,13, "B= $"  STR3 DB 10,13, "A-B= $"  STR4 DB "Overflow $"  .CODE  MAIN:  MOV AX, @DATA  MOV DS, AX  MOV DX, OFFSET STR1  MOV AH, 9  INT 21H  CALL INPUT  PUSH BX  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  POP CX  POP BX  POP AX  SUB BX,DX  PUSH BX  JC CARRY  JMP NOT\_CARRY  CARRY:  ADD CX,1  JC CARRY2  JMP NOT\_CARRY  NOT\_CARRY:  SUB AX,CX  JC CARRY2  PUSH AX  POP BX  CALL PRINT  POP BX  CALL PRINT  JMP EXIT  CARRY2:  MOV AX, @DATA  MOV DS, AX  MOV DX, OFFSET STR4  MOV AH, 9  INT 21H  JMP EXIT  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;  INPUT PROC  MOV CX, 16  MOV BX, 0  INPUT1:  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 INPUT1  RET  INPUT ENDP  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;  PRINT PROC  MOV CX,16  PRINT1:  MOV AH, 2  MOV DL, '0'  TEST BX,8000H  JZ ZERO  MOV DL, '1'  ZERO:  INT 21H  SHL BX, 1  LOOP PRINT1  RET  PRINT ENDP  ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;  EXIT:  END | ;;  Khai báo các biến  ;;  ;;  ;;  ;;  ;;  ;;  ;; Chương trình chính  ;;Lấy địa chỉ của DATA lưu vào AX  ;;DS = AX  ;;DX = OFFSET STR1  ;;AH = 9  ;;In ra màn hình STR1  ;;Gọi INPUT  ;;Lưu BX vào Stack  ;;Lấy đỉnh Stack lưu vào DX  ;;BX = BX – DX  ;;Cờ Carry = 1 nhảy tới CARRY  ;;Nhảy tới NOT\_CARRY  ;;Hàm CARRY  ;;CX = CX + 1  ;;Cờ Carry = 1 nhảy tới CARRY2  ;;Hàm NOT\_CARRY  ;;AX = AX - CX  ;;Lưu AX vào Stack  ;;Lấy đỉnh Stack lưu vào BX  ;;Gọi PRINT  ;;Nhảy tới EXIT  ;;Hàm CARRY2  ;;Procedure INPUT  ;;CX = 16  ;;BX = 0  ;;AH = 1  ;;Đọc kí tự được nhập từ bàn phím  ;;AL = AL – 30H  ;;AH = 0  ;;AX = 2  ;;AX = BX x AX  ;;BX = AX  ;;BX = BX + AX  ;;Lặp INPUT1 đến khi CX = 0  ;;Trở lại địa chỉ gọi  ;;Kết thúc  ;;Procedure PRINT  ;;CX = 16  ;;AH = 2  ;;DL = ‘0’  ;;BX = BX and 8000H, ZF = 1 nếu BX=0  ;;ZF = 1 nhảy tới ZERO  ;;DL = ‘1’  ;In kí tự ra màn hình  ;;Dịch trái 1 bit  ;;Lặp PRINT1 đến khi CX = 0 (tức là 16 lần)  ;;Kết thúc chương trình |