Assembly Language Programming(ALP) 8086

Microprocessor and Microcontroller

Session Objectives

- Discuss various programs with different set of 8086 instructions.
- Student can able to write a program for different simple problems.

Session Outcomes

- At the end of the session, students will be able to
  - Understand the instruction set of 8086 microprocessor.

Outline

- To discuss the
  - Different types of problems to understand the concept of 8086 programming.

Increment AL register

```
CODE  SEGMENT
        ASSUME CS:CODE
        MOV AL, 0AH
        INC AL
         HLT
CODE ENDS
        END
```

Increment an AX Register

```
CODE  SEGMENT
        ASSUME CS:CODE
        MOV AX, 0AH
        INC AX
         HLT
CODE ENDS
        END
```

Decrement AL register

```
CODE  SEGMENT
        ASSUME CS:CODE
        MOV AL, 0AH
        DEC AL
         HLT
CODE ENDS
        END
```

Decrement AX register

```
CODE  SEGMENT
        ASSUME CS:CODE
        MOV AX, 0AH
        DEC AX
         HLT
CODE ENDS
        END
```

1’s complement of an 8-bit number.

```
CODE  SEGMENT
        ASSUME CS:CODE
        MOV AL, 0AH
        NOT AL
         HLT
CODE ENDS
        END
```

1’s complement of an 16-bit number.

```
CODE  SEGMENT
        ASSUME CS:CODE
        MOV AX,100AH
        NOT AX
         HLT
CODE ENDS
        END
```

2’s complement of an 8-bit number.

```
CODE  SEGMENT
        ASSUME CS:CODE
        MOV AL, 0AH
        NOT AL
         INC AL
         HLT
CODE ENDS
        END
```

2’s complement of an 16-bit number.

```
CODE  SEGMENT
        ASSUME CS:CODE
        MOV AX,100AH
        NOT AX
         INC AX
         HLT
CODE ENDS
        END
```

Add two 8-bit numbers

```
CODE  SEGMENT
      ASSUME CS:CODE
      MOV AL , 040H
      MOV BL,  03H
      ADD AL, BL
      HLT
CODE ENDS
            END
```

Add two 16-bit numbers

```
CODE  SEGMENT
      ASSUME CS:CODE
      MOV AX , 00040H
      MOV BX,  0003H
      ADD AX, BX
      HLT
CODE ENDS
            END
```

Subtraction of two 8-bit numbers

```
CODE  SEGMENT
      ASSUME CS:CODE
      MOV AL , 040H
      MOV BL,  03H
      SUB AL, BL
      HLT
CODE ENDS
            END
```

Subtraction of two 16-bit numbers

```
CODE  SEGMENT
      ASSUME CS:CODE
      MOV AX , 00040H
      MOV BL,  0003H
      SUB  AX, BX
      HLT
CODE ENDS
            END
```

Factorial for two numbers

```
CODE  SEGMENT
      ASSUME CS:CODE
      MOV CX , 0005H
      MOV AX, 0001H
       MOV DX,  0000H
L1:   MUL CX
      DEC CX
      CMP CX, DX
      JNZ L1
      HLT
CODE ENDS
            END
```

ASCENDING ORDER

```
DATA SEGMENT
            STRING1 DB  88H,11H,22H,44H,33H
DATA ENDS
CODE SEGMENT
            ASSUME CS:CODE,DS:DATA
               START: MOV AX,DATA
            MOV DS,AX
            MOV CH,04H
UP2:        MOV CL,04H
            LEA SI,STRING1
UP1:         MOV AL,[SI]
            MOV BL,[SI+1]
            CMP AL,BL
            JC DOWN
            MOV DL,[SI+1]
            XCHG [SI],DL
            MOV [SI+1],DL
DOWN:       INC SI
            DEC CL
            JNZ UP1
            DEC CH
            JNZ UP2
            INT 3
CODE ENDS
END START
```

LARGEST, NUMBER IN ARRAY

```
Data segment
      STR DB 22h,33 h,00h,0ah,10h
       result db ?
data ends
 code segment
                assume cs:code, ds:data
                start: mov ax, data
                mov ds, ax
                mov cx, 04h
                mov bl, 00h
                LEA SI, STR
up:             mov al, [SI]
                cmp al, bl
                jl nxt
                mov bl, al
nxt:            inc si
                dec cx
                jnz up
                mov res,bl
                int 3
code ends
end
```

Summary

- The different types of programs for 8086 were discussed.

Test of your understanding

- Write an ALP for BCD number addition .
- Write an for ALP for find string in the array.

References

- Yu-Cheng Liu, Glenn A. Gibson, “Microcomputer Systems: The 8086 / 8088 Family -Architecture, Programming and Design”, Second Edition, Prentice Hall of India, 2007.
- Doughlas V. Hall, “Microprocessors and Interfacing, Programming and Hardware”, TMH, 2012.

Thank you