# **Logo, company name Description automatically generated**

# **EEE 302**

Microprocessors and Interfacing Experiment 3

Rotate, Shift and LOOPs in assembly language

Department of Electrical and Electronic Engineering, EWU

## Objective:

To get familiar with Rotate and Shift commands in assembly langiage. To use loops in complex problems.

## Introduction:

**Shift and Rotate command:**

Shift and Rotate commands are used to convert a number to another form where some bits are shifted or rotated. Basic difference between shift and rotate is shift command makes “fall of” bits at the end of register. Where rotate command makes “Wrap around” at the end of the register. There are both arithmatic (SAL and SAR) and logical (SHL and SHR) shift instructions. Graphical operation for these commands are shown below.

### MSB LSB

0

Data

CF

SHL (Shift Logical Left)

### 0

CF

Data

SHR (Shift Logical Right)

### MSB LSB

0

Data

CF

SAL (Shift Arithmatic Left)

SAR (Shift Arithmatic Right)

Data

CF

Data

CF

ROL (Rotate Left)

Data

CF

ROR (Rotate Right)

CF

Data

RCL (Rotate Through Carry Left)

CF

Data

RCR (Rotate Through Carry Right) Some simple codes can be given to clarify the idea.

MOV CL,03H ;

MOV AX,02F3H ; In binary 0000 0010 1111 0011

SHR AX,CL ; In binary 0000 0000 0101 1110

In this procedure, SHR commands inserts 0’s from right side. Each time a 0 is inserted left most bit is vanished from register content.

MOV CL,03H ;

MOV AX,82F3H ; In binary 1000 0010 1111 0011

SAR AX,CL ; In binary 1111 0000 0101 1110

In this procedure, SHR commands inserts MSB content from right side. Each time it is inserted left most bit is vanished from register content.

MOV CL,03H ;

MOV AX,82F3H ; In binary 1000 0010 1111 0011

ROR AX,CL ; In binary 0111 0000 0101 1110

The whole procedure can be visualized as follows.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 1 | 1 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 0 |

Here rotate by 3 operation is shown. It is clearly seen that every bit is assigned to a new position that is 3 places away from previous one. Unlike the shift command no right bit is destroyed. It is placed in the leftmost position.



## Exercise part 1:

1. Program 1:

CODE SEGMENT

ASSUME CS:CODE

MOV CL,02H MOV AX,105AH SHL AX,CL HLT

CODE ENDS

END

Obtain AX register value in write the previous value and present value in binary form. What type of operation is this?

1. Program 2:

CODE SEGMENT

ASSUME CS:CODE

MOV CL,04H MOV AX,564AH SAL AX,CL

HLT

CODE ENDS

END

Obtain AX register value in write the previous value and present value in binary form. What type of operation is this?

1. Perform for similar values of AX and CL with ROL, ROR. RCL, RCR command.

## LOOP in assembly language:

Loop commands are used to perform same operation again and again. This is like for, while type instructions in ‘C’ or ‘MATLAB’. A common example can be shown as,

MOV CX,0100D MOV AX,564AH

Lev: DEC AX

Loop LEV HLT

Here CX acts as a count register. Loop Lev instruction leads instruction to go back to Lev level until CX is zero. Each time Lev level is executed CX is decreased by 1. Loop command can be used for waiting purposes. Such as,

MOV CX,0100D Wt: NOP

Loop Wt HLT

Here the loop is executed until CX is zero. If 1 loop takes 1ms, the program will wait for 100ms.

## Exercise part 2:

1. Program 1:

CODE SEGMENT

ASSUME CS:CODE

MOV AX,1025H MOV BX,475AH MOV CX,50H

Lev: INC AX

DEC BX LOOP Lev HLT

CODE ENDS

END

Observe the operation of this code. What happens when the loop is executed again and again.

1. Program 2: This code is to find GCD of two numbers.

CODE SEGMENT

ASSUME CS:CODE

MOV AX,5H MOV BX,3H

Lev: XOR DX,DX DIV BX MOV AX,BX MOV BX,DX TEST DX,0H

JNZ Lev HLT

CODE ENDS

END

Here GCD of 5 and 3 are found. You can change the values of AX and BX and obtain the result for any other values. Find GCD of 08D4H and 235H.

Result:

1. Find Least Common Multiplier of 12H and 25H.

## Report:

* 1. Suppose x = 20 and y = 28. Add y with x for 30 times.
  2. Multiply 12 by 6 until result is below 3000H. If result is greater than this, divide the result by 2 for 3 times.
  3. You can get input into microprocessor via following code.

MOV AH, 1H ; keyboard input subprogram INT 21H

HLT

Take input from the keyboard until b is pressed.