

实验三

一、实验目的

- 学习循环移位指令
- 掌握简单的字符串加密和输入输出方法

二、实验内容

- 输入一个任意长度的字符串（姓名的拼音字母）
- 对字符串进行加密和解密操作。每个单字符采用不同的密钥。密钥存于数组Key中。
 - 例： $\text{key} = \{5, 0, 6, 2, 3, 1, 4, 6\}$
- 采用循环移位加密
 - 单字符密钥范围(-7~7)。其中负数表示循环左移，正数表示循环右移，0不变，数字表示移动的位数（须判断密钥正负，不能采用反向移位）
- 采用两种方法显示密文和解密后的明文。
 - 使用指令INT 21H
 - 用不同颜色显示

三、背景知识

1、字符输入方法

MOV AH, 01H

INT 21H

MOV CHAR, AL

- AL存放输入字符的ASCII码

2、字符串赋值方法

plainText DB '20230525'

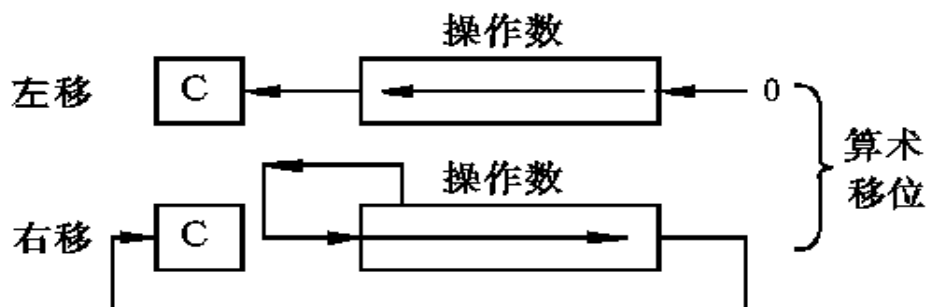
cipherText DB 8 DUP (?)

decryptedText DB 8 DUP (?)

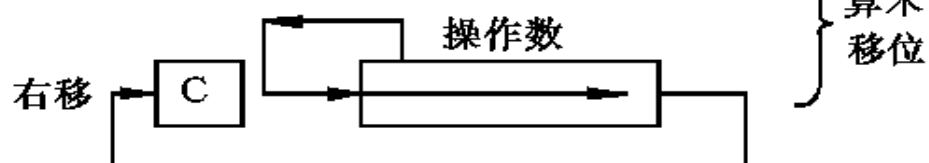
3、加密解密

- 循环移位指令ROR、ROL

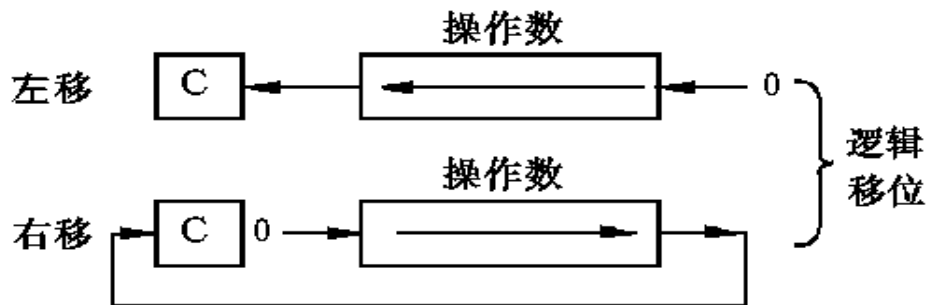
SAL



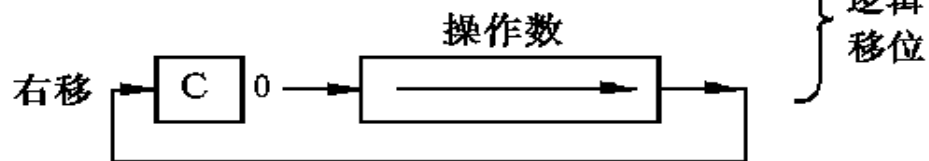
SAR



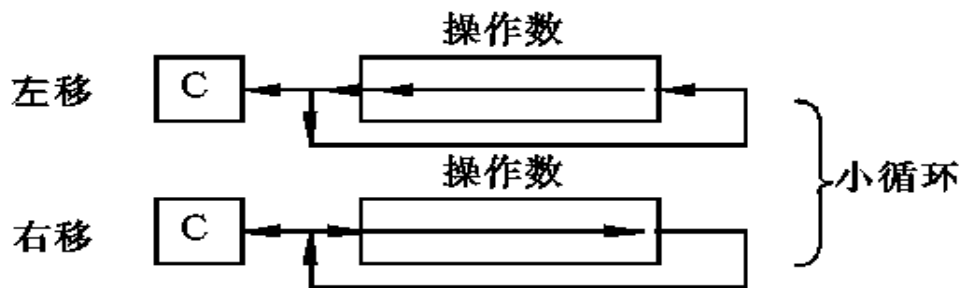
SHL



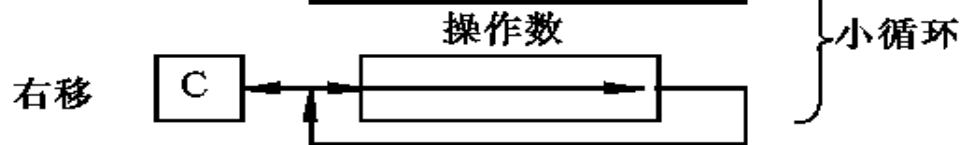
SHR



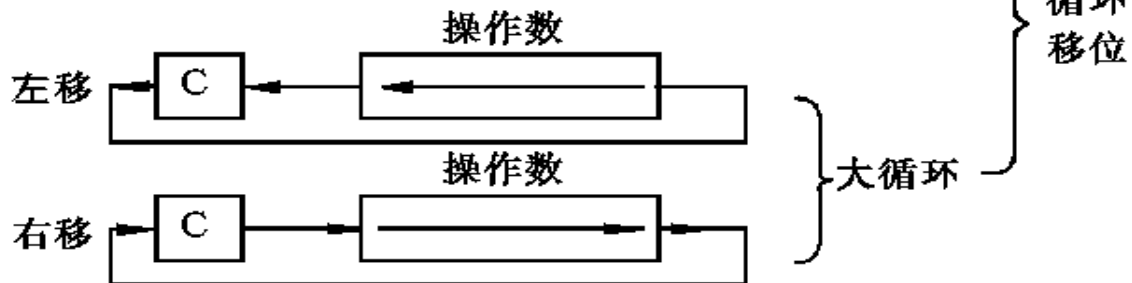
ROL



ROR



RCL



RCR



移位操作

循环移位

- 移位指令格式

ROL REG/MEM, CL/IMM8

- 左边是目的操作数，右边是移位次数
（立即数，或存于CL寄存器中）
- 如果移动位数大于1，必须使用CL

4、字符串显示输出

1. 单个字符显示输出

需要输出的字符保存在DL中

```
MOV AH, 02H
```

```
INT 21H
```

2. 用直接写显存的方法输出字符串

- 段定义

display segment

string1 db 10 dup(?)

display ends

- 段地址声明

assume es:display

- 显存地址范围: **B800H~BFFFH**

- 示例

MOV AX, 0B800H

MOV ES, AX

MOV string[DI], AL

MOV string[DI+1], 0AH ;存入颜色信息

MOV string[DI+160], AL;换行显示

MOV string[DI+160+1], 0AH