实验三

一、实验目的

- 学习逻辑运算指令和移位指令的用法

二、实验内容

内容1

- 实现两个无符号整数的乘法，使用SHL和ADD指令（不使用乘法指令），结果存于product中
- 说明：操作数不能是2的整数次幂

内容2

- 对一个长度为***n***的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：

Source = ‘20210428’

key = {5，0，6，2，3，1，4，6}

- 方法一：采用XOR指令。单字符密钥范围(0～255)
- 程序运行结果依次显示：

  所产生的密文

  解密后的明文

内容3

- 对一个长度为***n***的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：

Source = ‘20210428’

key = {5，0，6，-2，3，-1，4，6}

- 方法二：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
- 程序运行结果依次显示：

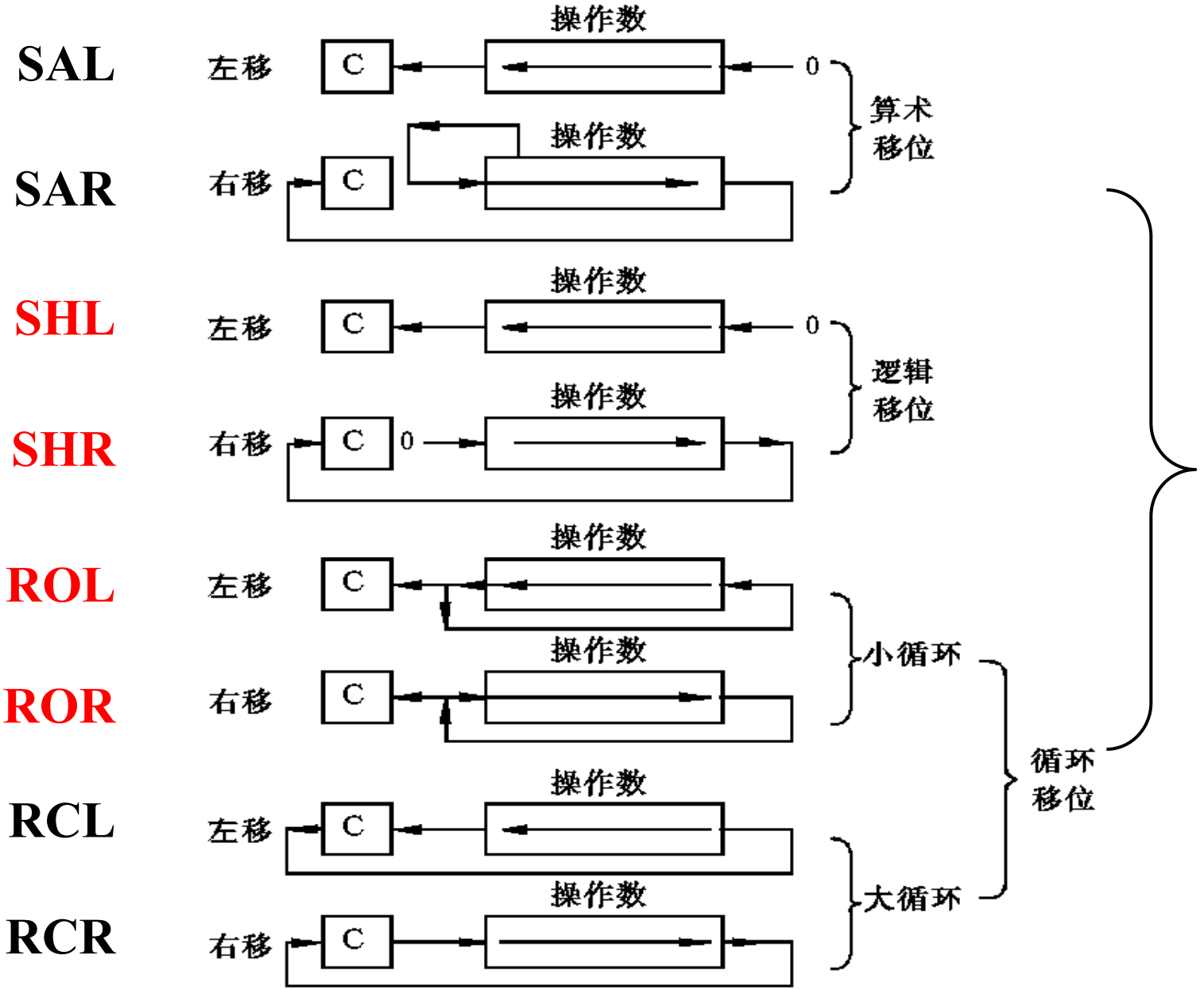
  所产生的密文

  解密后的明文

三、背景知识

1. 移位指令

移位操作


SAL
左移
C
操作数
0
SAR
右移
C
操作数
算术
移位
SHL
左移
C
操作数
0
SHR
右移
C
操作数
0
逻辑
移位
ROL
左移
C
操作数
ROR
右移
C
操作数
小循环
RCL
左移
C
操作数
RCR
右移
C
操作数
大循环
循环
移位

- 移位指令格式

SHL 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

- 说明：用LOOP指令实现循环，用堆栈操作解决CX与CL使用冲突的问题；
- PUSH REG/MEM16
- POP REG/MEM16

2、字符串赋值方法

```
plainText  DB  '20210428',13,10,'$'
cipherText  DB  8 DUP (?),13,10,'$'
decryptedText DB 8 DUP (?),13,10,'$'
```

3、字符串显示输出

```
LEA  DX, plainText
MOV  AH, 09H
INT  21H
```