

没有足够的 I/O 引脚是单片机系统设计的瓶颈,本章利用 8255A 与移位寄存器实现 I/O 端口扩展。

5.1 可编程并行接口 8255A

5.1.1 基本特性

8255A 为可编程并行 I/O 接口芯片,具有 3 个 8 位并行 I/O 端口(A 口、B 口、C 口),其中 C 口可分为 2 个 4 位并行 I/O 口使用,并具有按位复位/置位功能,兼容 TTL/CMOS 电平。

8255A 地址线 A1 和 A0 与系统地址线 A1A0 连接,实现对 8255A 内部 PA、PB、PC 和控制端口的寻址,端口编址见表 5-1。

表 5-1 8255A 端口编址

$\overline{\text{CS}}$	$\overline{\text{RD}}$	$\overline{\text{WR}}$	A1	A0	选 择 端 口	传 送 方 向
0	0	1	0	0	读 A 端口	PA→数据总线
0	0	1	0	1	读 B 端口	PB→数据总线
0	0	1	1	0	读 C 端口	PC→数据总线
0	1	0	0	0	写 A 端口	PA←数据总线
0	1	0	0	1	写 B 端口	PB←数据总线
0	1	0	1	0	写 C 端口	PC←数据总线
0	1	0	1	1	写控制端口	控制端口←数据总线

8255A 具有 3 种工作方式。

方式 0: 基本 I/O 方式,适用于端口 A、端口 B 和端口 C。

方式 1: 选通 I/O 方式,适用于端口 A 和端口 B。

方式 2: 双向 I/O 方式,适用于端口 A。

8255A 定义工作方式控制字以设定 8255A 工作方式,定义见图 5-1。

8255A 定义端口 C 置位/复位控制字,对 C 口中的任意一位进行置位或者复位操作,定义见图 5-2。

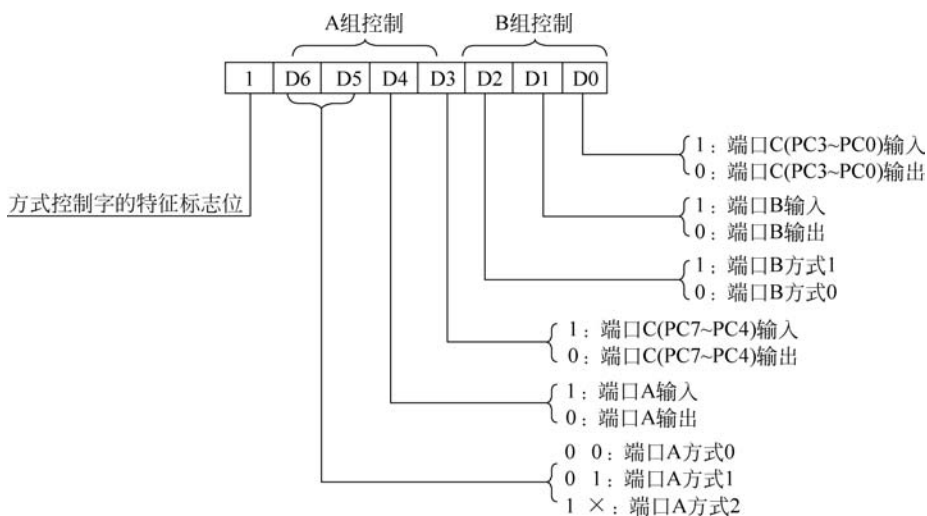


图 5-1 8255A 工作方式控制字定义

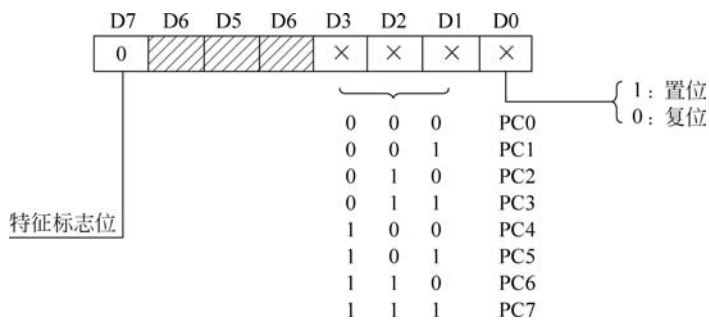


图 5-2 C口置位/复位控制字

5.1.2 拨码开关与显示接口

1. 原理图

利用 8255 扩展 8 位输入和 2 个 8 位输出接口(见图 5-3),定义 PA 口和 PC 口为工作方式 0 输出,定义 PB 口为工作方式 0 输入。拨码开关连接 8255A 的 PB 端口,8 个 LED 指示条连接 PA 端口,指示拨码开关状态。2 位 7 段 LED 显示器(BCD 码输入)连接 PC 端口,以十六进制形式显示当前拨码值。

以 3.1 节的设计模块为主控单元,8255A 片选信号连接系统地址译码信号 $\overline{Y1}$,8255A 的 PA 口、PB 口、PC 口以及控制口地址见表 5-2。

表 5-2 8255A 内部端口编址

$\overline{CS}(\overline{Y1})$	A15~A12	A11~A2	A1	A0	端口地址
0	0001	未用	0	0	PA 口: 1000H
			0	1	PB 口: 1001H
			1	0	PC 口: 1002H
			1	1	控制口: 1003H

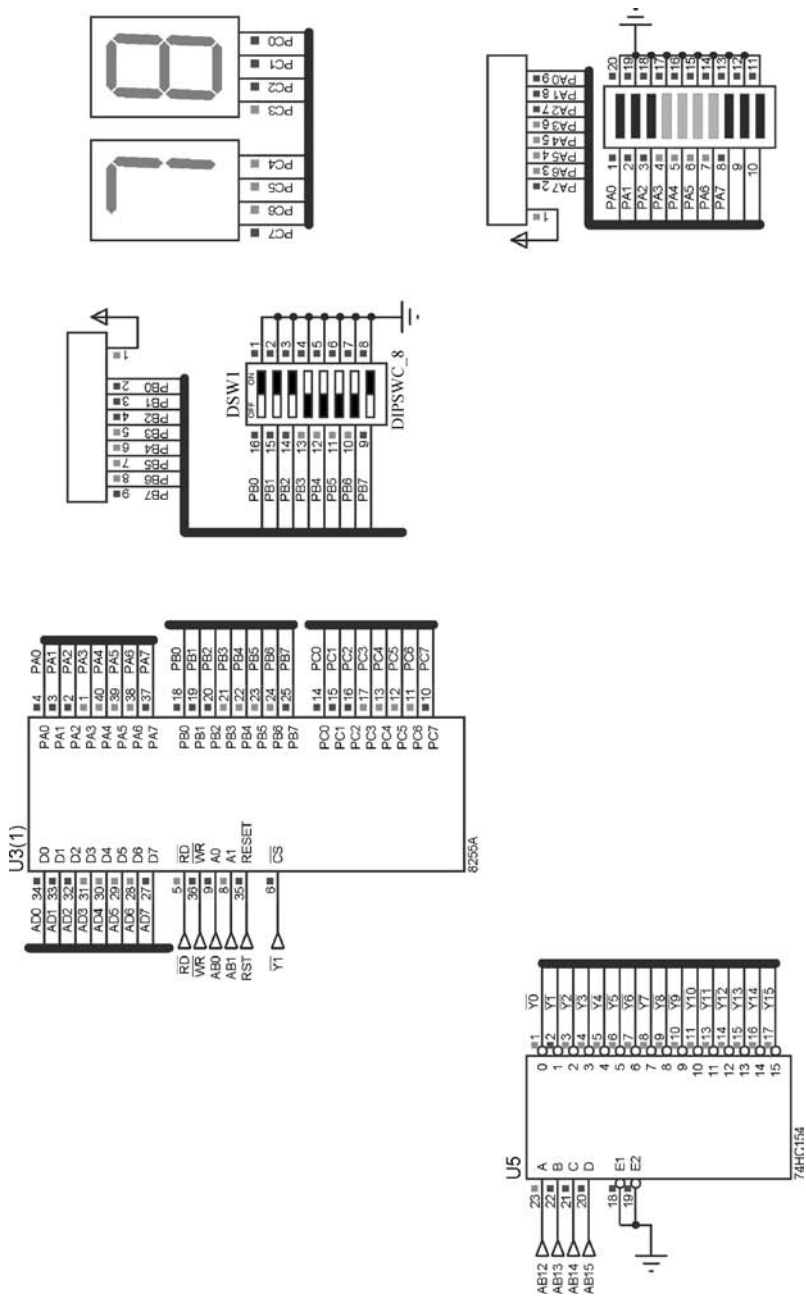


图 5-3 拨码开关与显示接口

2. 参考程序

```
#include "reg51.h"
#include <absacc.h>
#define P1A8255 XBYTE[0x1000] //PA 地址定义
#define P1B8255 XBYTE[0x1001] //PB 地址定义
#define P1C8255 XBYTE[0x1002] //PC 地址定义
#define P1COM8255 XBYTE[0x1003] //控制口地址定义

void vDelay(unsigned int uiT )
{
    while(uiT-- ) ;
}

void main()
{
    unsigned char ucD;
    P1COM8255 = 0x82; //定义 PA、PC 工作方式 0 输出,PB 工作方式 0 输入
    while(1)
    {
        ucD = P1B8255; //读拨码开关
        P1A8255 = ucD; //LED 条显示
        P1C8255 = ucD; //数码显示
    }
}
```

5.1.3 打印机接口

并行打印机接口标准 Centronics 规定了打印机接口的信号定义和数据传输时序。

1. 接口标准

完整的 Centronics 标准定义了 36 芯打印机接口,信号线定义见表 5-3。

表 5-3 Centronics 并行打印机接口标准

引脚号	信号	I/O	功能	引脚号	信号	I/O	功能
1	STB	输入	数据选通	13	SLCT	输出	正常工作
2	D1	输入	数据线	14	AUTO	输入	自动走纸
3	D2	输入	数据线	16	逻辑地		
4	D3	输入	数据线	17	机架地		
5	D4	输入	数据线	19~30	地		
6	D5	输入	数据线	31	INT	输入	复位
7	D6	输入	数据线	32	ERROR	输出	脱机出错
8	D7	输入	数据线	33	地		
9	D8	输入	数据线	35	+5V		电源
10	ACK	输出	准备就绪	36	SLCTIN	出	工作允许
11	BUSY	输出	打印机忙	15/18/34	NC		未用
12	PE	输出	缺纸				

表 5-3 中的输入和输出是相对打印机而言的,输入为控制信号线,输出为打印机状态线。Centronics 标准打印机接口数据传输时序见图 5-4。

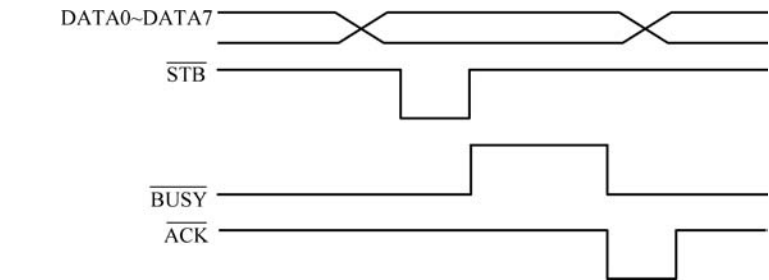


图 5-4 Centronics 标准打印机接口数据传输时序

当 CPU 打印数据时,首先查询信号 BUSY,当 BUSY=0 时,表明打印机处于不忙状态,将数据通过数据总线输出给接口,然后输出数据选通脉冲信号 $\overline{\text{STB}}$,将数据送入打印机内部数据锁存器。打印机在 $\overline{\text{STB}}$ 信号的上升沿将 BUSY 置 1,表明打印机正在处理数据,不能接收新的数据。待输入数据处理完成,打印机输出 $\overline{\text{ACK}}$ 信号,表明打印机可以接收下一个数据。同时,在 $\overline{\text{ACK}}$ 下降沿,使 BUSY 复位,撤销忙状态标志,一个数据传输过程结束。

2. 接口电路

以 3.1 节的设计模块为主控单元,用 1 片 8255A 设计查询方式打印机接口电路见图 5-5,8255A 片选信号 $\overline{\text{CS}}$ 连接地址译码器输出信号 $\overline{\text{Y1}}$,8255A 端口编址见表 5-4。

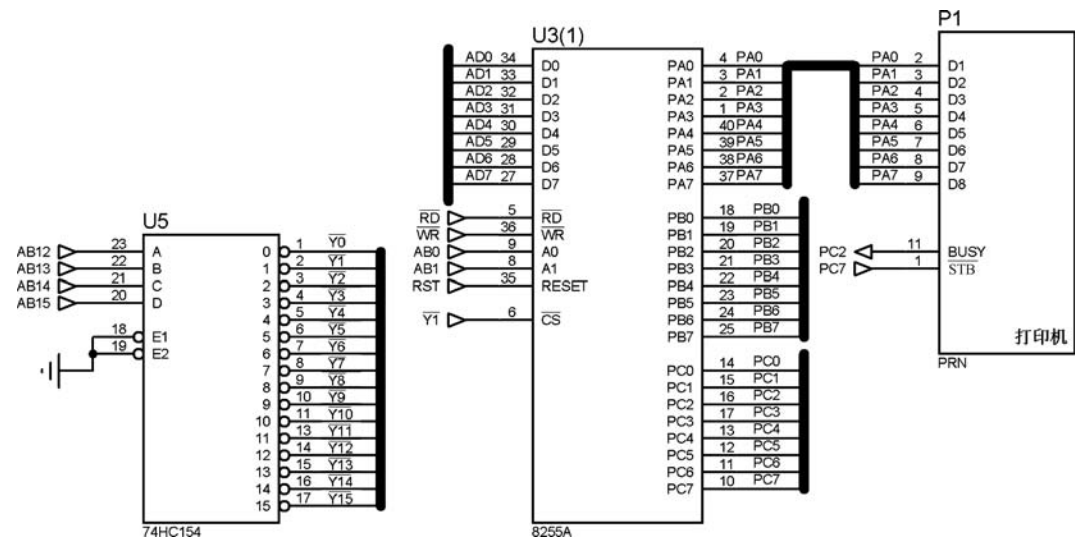


图 5-5 8255A 打印机接口电路

端口 A 设置为方式 0 输出,输出打印数据,端口 C 高 4 位设置为输出方式,由 PC7 提供选通信号 $\overline{\text{STB}}$,端口 C 低 4 位设置为输入方式,由 PC2 接收打印机状态信号 BUSY。

表 5-4 8255A 内部端口编址

$\overline{CS}(\overline{Y1})$	A15~A12	A11~A2	A1	A0	端 口 地 址
0	0001	未用	0	0	PA 口：1000H
			0	1	PB 口：1001H
			1	0	PC 口：1002H
			1	1	控制口：1003H

3. 参考程序

```
# include <reg51.h>
# define COM8255 XBYTE[0x1003]
# define PA8255 XBYTE[0x1000]
# define PB8255 XBYTE[0x1001]
# define PC8255 XBYTE[0x1002]
void vPrinter(unsigned char * ucD)
{
    while( * ucD!= '\0')
    {
        while(0x04 & PC8255);           //查询 BUSY 标志,等待打印机空闲状态 PC2
        PA8255 = * ucD;                 //输出字符
        COM8255 = 0X0E;COM8255 = 0X0E; //产生 STB 脉冲信号
        COM8255 = 0X0F;COM8255 = 0X0F;
        ucD++;
    }
}
void main()
{
    unsigned char Data[8] = "WELCOME!";
    COM8255 = 0x81;                    //设置 A 口方式 0,输出;C 口高 4 位输出,低 4 位输入
    vPrinter(Data);                    //打印字符串
}
```

5.1.4 键盘编码芯片 74C922

1. 74C922

74C922 为 16 键解码芯片,封装与引脚见图 5-6。

74C922 内部振荡器完成 4×4 键盘矩阵扫描、消抖和编码,矩阵键盘的 4 行分别连接 74C922 的 Y1~Y4,4 列分别连接 X1~X4。有按键按下时,DA 引脚输出高电平,同时封锁其他按键,片内锁存器保持当前按键的 4 位编码。74C922 的键盘接口如图 5-7 所示。

2. 接口电路

键盘编码输出连接 8255A 的 PB 端口的 PB0~PB3, DA 连接 INT0,在 INT0 中断处理程序中读按键编码,并在 8255A 的 PA 端口和 PC 端口显示。按键采用 Proteus 的 16 键小键盘,其原理见图 5-8。

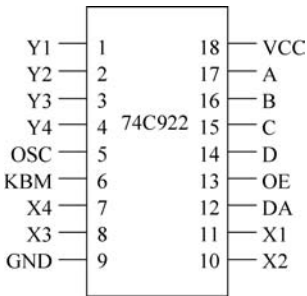


图 5-6 74C922 封装与引脚

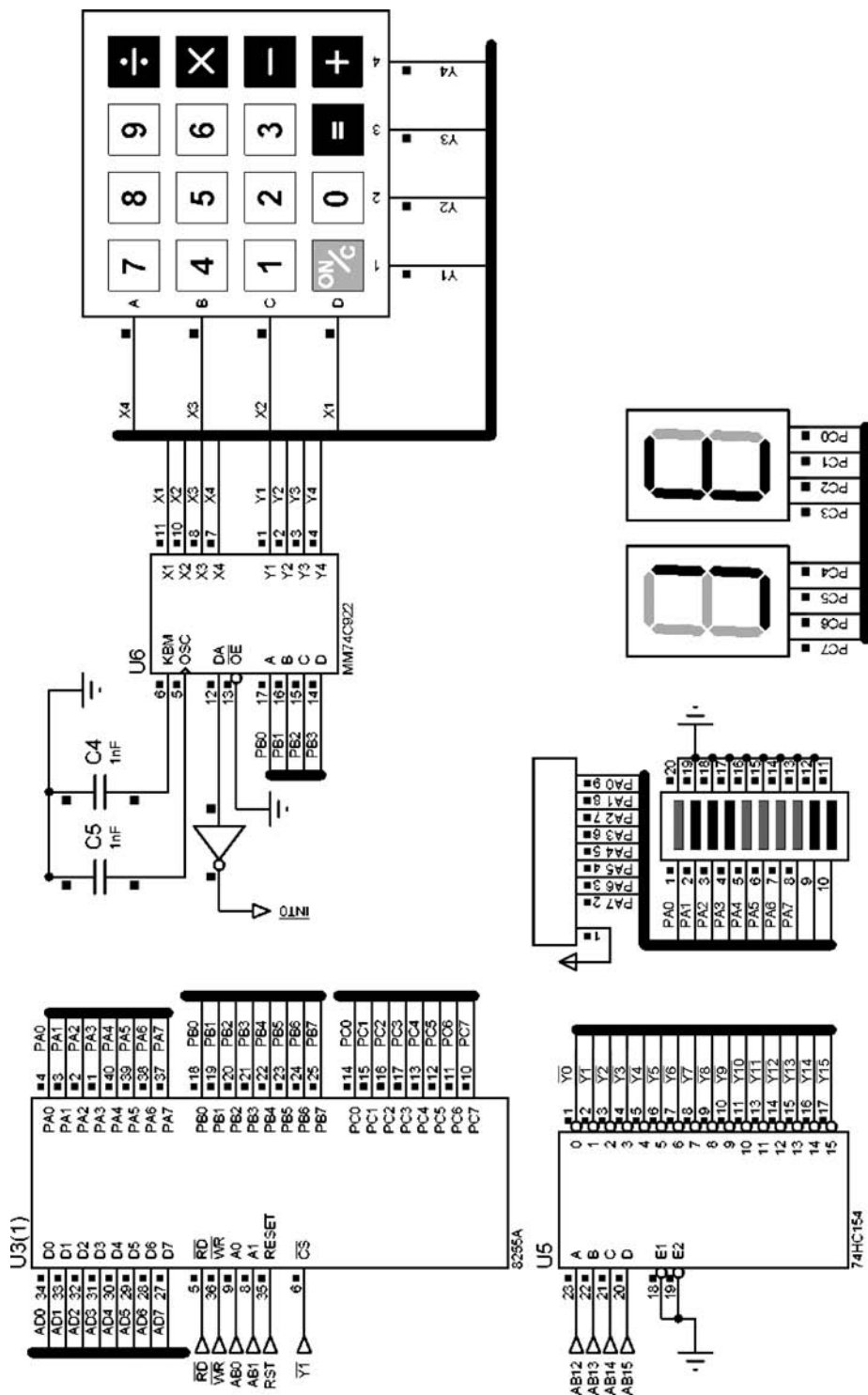


图 5-7 74C922 按键接口

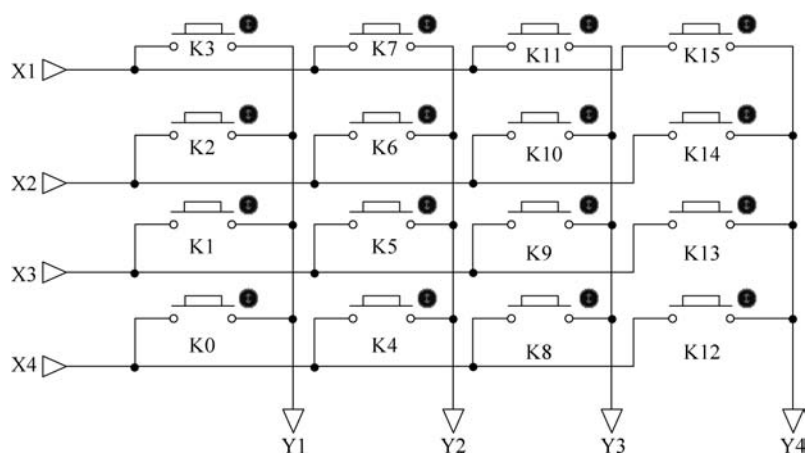


图 5-8 4×4 矩阵键盘原理图

3. 参考程序

```
#include "reg51.h"
#include <absacc.h>
#define uchar unsigned char
#define uint unsigned int

#define P1A8255 XBYTE[0x1000]    //PA 地址定义
#define P1B8255 XBYTE[0x1001]    //PB 地址定义
#define P1C8255 XBYTE[0x1002]    //PC 地址定义
#define P1COM8255 XBYTE[0x1003]  //控制口地址定义

void vDelay(unsigned int uiT )
{
    while(uiT-- ) ;
}

void EINT0() interrupt 0
{
    uchar ucD;
    ucD = P1B8255;           //读 PB 口
    P1A8255 = ucD;           //PA、PC 口输出
    P1C8255 = ucD;           //
}

void delays(uint j)
{
    while (j-- );
}

void main()
{
    unsigned char ucD;
    IE = 0x81;               //INT0 中断允许
    IT0 = 1;
```



```

P1COM8255 = 0x82;           //设定 PA、PC 为方式 0 输出, PB 方式 0 输入
while(1);
}

```

5.2 移位寄存器扩展并行输入接口

5.2.1 CD4014 扩展并行输入接口

1. CD4014

CD4014 为 8 位并入-串出移位寄存器,可编程实现并行 8 位二进制数的锁存和移位,可级联,引脚定义及功能见图 5-9。

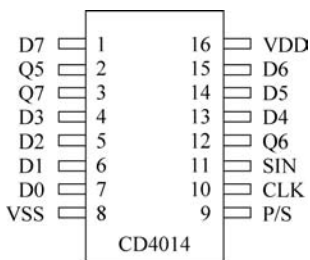


图 5-9 CD4014 引脚

引脚定义及功能：

- D[7..0]——8 位数据并行输入端。
- SIN——串行数据输入端,用于多片 CD4014 级联。
- CLK——时钟输入,用于串行移位和并行数据置位,上升沿有效。
- P/ $\bar{S}$ ——并/串选择, P/ $\bar{S}$ =1,并行置位工作方式,在 CLK 上升沿,将并行数据置入锁存。P/ $\bar{S}$ =0,串行移位工作方式。
- Q7、Q6、Q5——移位寄存器高 3 位输出端。

2. 原理图

用 2 片 CD4014 级联,实现并行输入接口电路见图 5-10。

信号定义及功能：

- SCLK——串行移位/并行锁存时钟。
- P/ $\bar{S}$ ——工作模式选择, P/ $\bar{S}$ =0,串行移位工作方式; P/ $\bar{S}$ =1,并行数据置入方式,在 CLK 上升沿将并行数据 D15~D0 置入。
- SERD——串行数据输出端。
- DI[15..0]——并行数据输入端。
- SIN——串行输入,CD4014 级联引脚。

3. 工作过程

串行端口设置为工作方式 0,即同步移位寄存器工作方式, TXD(P3.1)连接 SCLK,输出移位脉冲, RXD(P3.0)连接 SERD,接收来自 CD4014 的串行数据。

(1) P/ $\bar{S}$ =1,并行数据锁存。

(2) P/ $\bar{S}$ =0,移位模式,在 CLK 移位脉冲下,并行数据转化为串行数据从 SERD 输出。

4. 参考程序

```

#include <reg51.h>
#include <absacc.h>
sbit PL = P3^7;
sbit PCLK = P3^1;
void vDelay(unsigned int uiT)
{

```

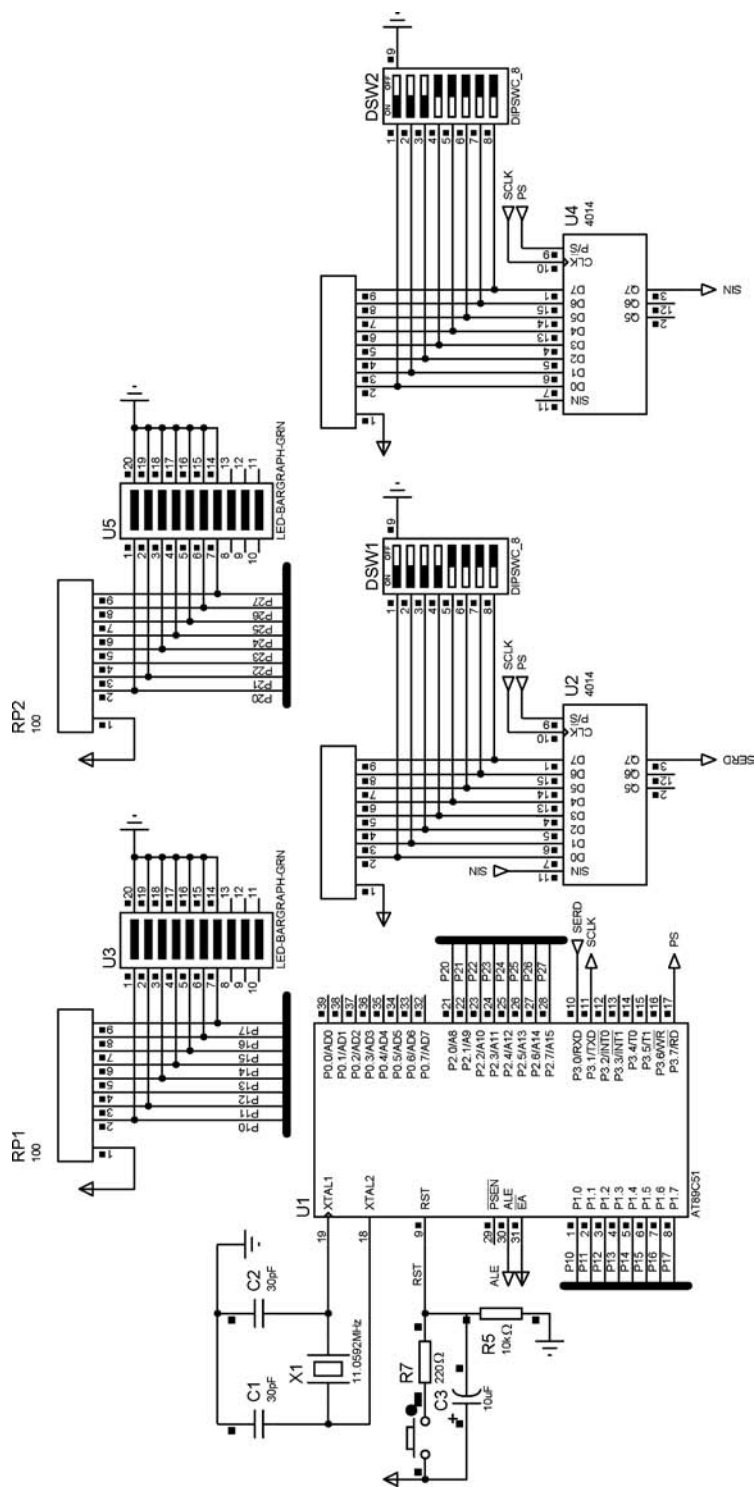


图 5-10 并行输入接口电路

```

    while(uiT--);
}
void main()
{
    unsigned char ucD = 0, i;
    while(1)
    {
        PL = 1;
        PCLK = 0; PCLK = 1;           //并行数据锁存
        PL = 0;                       //串行移位工作方式
        SCON = 0x00;
        REN = 1;
        while(!RI);
        ucD = SBUF;
        RI = 0;
        P1 = ~ucD;
        vDelay(1000);
        while(!RI);
        ucD = SBUF;
        RI = 0;
        P2 = ~ucD;
    }
}

```

5.2.2 74HC165 扩展并行输入接口

1. 74HC165

74HC165 为 8 位并入-串出移位寄存器,可编程实现并行 8 位二进制数的锁存和移位,可级联,其引脚见图 5-11。

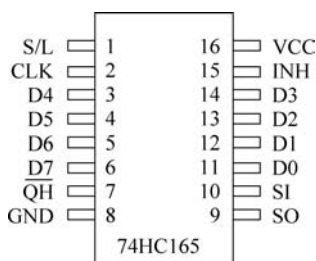


图 5-11 74HC165 引脚

引脚定义及功能:

- $D[7..0]$ ——8 位并行输入端。
- SI——串行数据输入端,用于多片级联。
- CLK——时钟脉冲输入,用于串行移位,上升沿有效。
- $SH/\overline{LD}$ ——移位/置数端, $SH/\overline{LD} = 1$, 移位工作方式,在 CLK 上升沿,串行移位; $SH/\overline{LD} = 0$, 将并行数据置入,并行数据置入与时钟无关。

- QH——串行数据输出。
- $\overline{QH}$ ——串行数据反相输出。
- INH——时钟禁止端。

2. 原理图

用 2 片 74HC165 级联,实现 16 位并行输入接口,原理电路见图 5-12。

信号定义及功能:

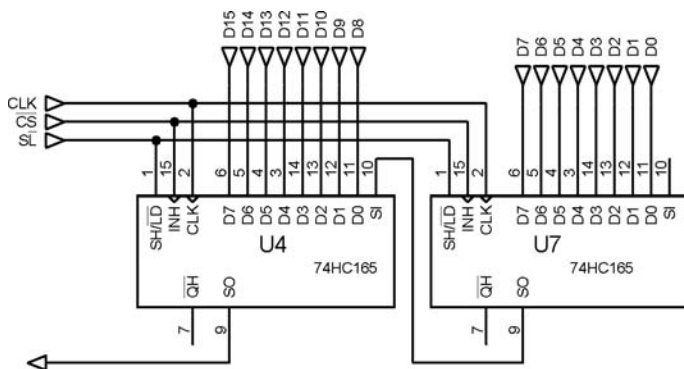


图 5-12 16 位并行输入接口电路(74HC165)

- $\overline{CS}$ ——接口模块片选信号,输入,低有效。
- CLK——串行移位/并行锁存时钟,输入。
- $\overline{SL}$ ——工作模式选择, $\overline{SL}=1$, 串行移位工作方式; $\overline{SL}=0$, 并行数据置入方式, 在 CLK 上升沿将并行数据置入。
- SO——串行数据输出端。
- $DI[15..0]$ ——16 位并行数据输入端。

3. 工作过程

在 $\overline{CS}$ 有效情况下,

$\overline{SL}=0$, 2×8 位并行数据锁存。

$\overline{SL}=1$, 移位模式, 在 CLK 移位脉冲下, 16 位并行数据转化为串行数据从 SO 输出。

4. Proteus 仿真

功能: 用 74HC165 实现 16 位开关输入。

1) 同步移位寄存器工作方式

该模块可工作于 MCS-51 串口工作方式 0 (同步移位寄存器工作方式), 只需初始化串口为工作方式 0, 即可按串口接收模式读取数据, 程序设计简单, 但占用系统串口, 工作原理见图 5-13。

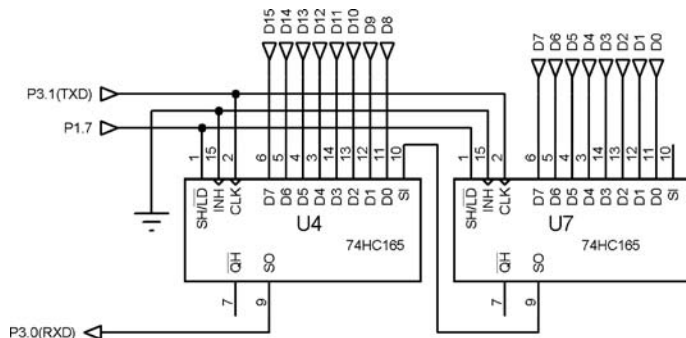


图 5-13 同步移位寄存器工作方式

(1) 信号定义。

- $\overline{\text{CS}}$: 直接接地, 即 $\text{INH}=0$, 时钟允许。
- CLK : 连接 TXD , 方式 0 时, 产生移位脉冲。
- SO : 连接 RXD , 同步方式接收串行数据。
- $\text{S}/\overline{\text{L}}$: 连接 AT89C51 的 P1.7 , 作为工作方式控制引脚, $\text{S}/\overline{\text{L}}=0$, 并行数据锁存, $\text{S}/\overline{\text{L}}=1$, 串行移位发送。

(2) 参考程序。

```
#include "reg51.h"
#include <absacc.h>
////////控制位定义////////
sbit SL = P1^7;
sbit SLCLK = P3^1;          //TXD
void vDelay(unsigned int uiT )
{
    while(uiT-- ) ;
}
////////读 16 位按键状态///
void main()
{
    unsigned char ucData[2];    //存放按键状态
    SL = 1;
    SCON = 0x00;                //初始化串口为工作方式 0, 即移位寄存器方式
    ucData[0] = SBUF;           //读低 8 位键值
    ucData[1] = SBUF;           //读高 8 位键值
}
```

2) 模拟同步移位寄存器工作方式

为节省串口资源, 可采用模拟串行工作方式 0 的方式, 即采用 3 个 I/O 引脚, 连接见图 5-14。

引脚定义及功能:

P2.2 ——控制 74LS165 工作模式, $\text{P2.2}=0$, 锁存模式, $\text{P2.2}=1$, 移位寄存器模式。

P2.1 ——模拟产生移位脉冲信号。

P2.0 ——串行数据输入端。

$\overline{\text{CS}}$ ——直接接地, 即 $\text{INH}=0$, 时钟允许。

5. 参考程序

在程序中实现串-并转换, 在 CPU 中得到 16 位并行数据, 在 P1 和 P3 显示。

```
#include <reg52.h>
#include <intrins.h>
#define NOP() _nop_()
sbit CLK = P2^1;
sbit IN_PL = P2^2;
sbit IN_Dat = P2^0;
unsigned char ReHC74165(void)
{
    unsigned char i, ucD;
    ucD = 0;
```

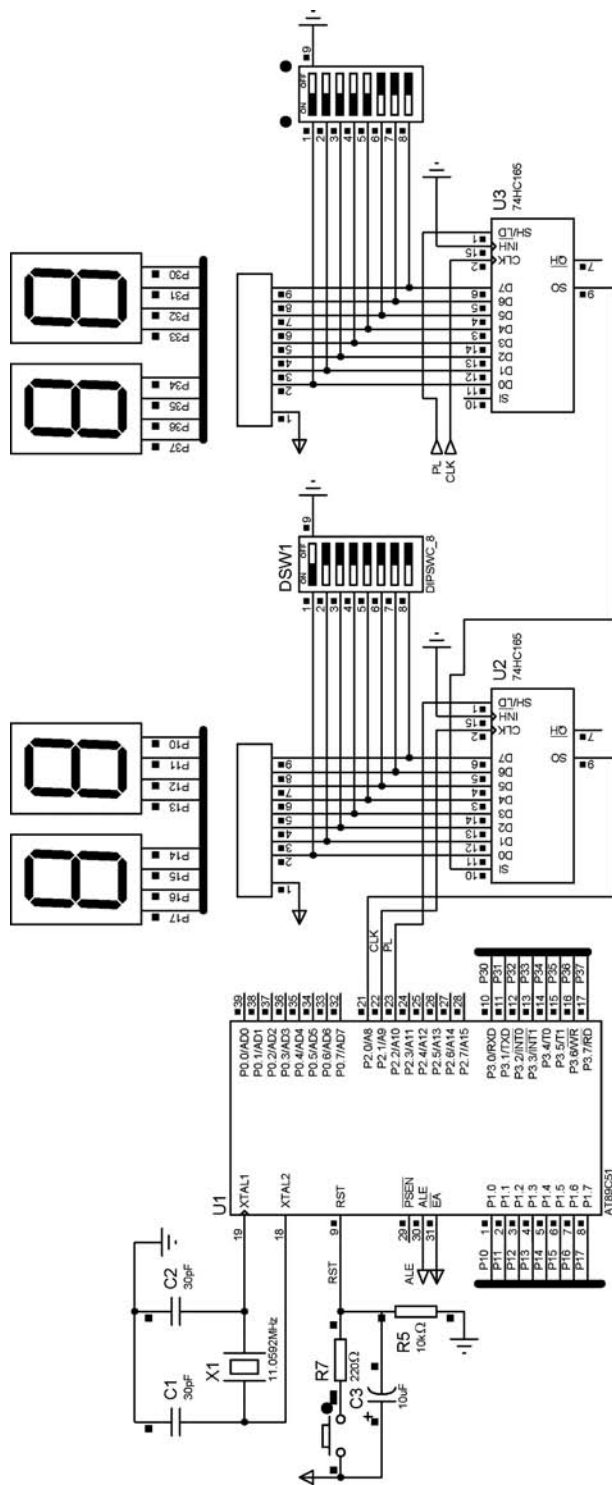


图 5-14 模拟同步移位寄存器工作方式

```

    for(i = 0; i < 8; i++)
    {
        ucD = ucD << 1;
        if(IN_Dat == 1)ucD = ucD + 1;
        CLK = 0;
        NOP();
        CLK = 1;
    }
    return ucD;
}
void main()
{
    while(1)
    {
        IN_PL = 0; NOP(); IN_PL = 1;NOP();
        P1 = ReHC74165();          //送两位七段 LED 显示器显示,BCD 码
        P3 = ReHC74165();          //送两位七段 LED 显示器显示,BCD 码
    }
}

```

5.3 移位寄存器扩展并行输出接口

5.3.1 74HC164 扩展并行输出接口

1. 74HC164

74HC164 为串入-并出移位寄存器,可级联,其引脚见图 5-15。

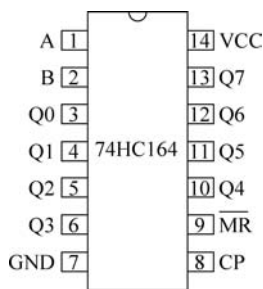


图 5-15 74HC164 引脚

引脚定义及功能:

- $Q[7..0]$ ——8 位并行输出。
- A、B——串行数据输入。
- $\overline{MR}$ ——数据清零。
- CP——移位时钟。

数据通过两个输入端(A 和 B)之一串行输入,任一端可用作高电平使能,控制另一端的输入。

2. 原理图

4 片 74HC164 扩展输出接口,原理图见图 5-16。

在图 5-16 中,信号定义如下:

- DAT: 接收来自单片机串行数据。
- SCLK: 移位时钟。

4 片 74HC164 通过 Q7 级联,输入引脚 A 与 B 并接,作为串行数据输入端。利用 P1.1 产生移位脉冲,串行数据从 P1.0 输出。

3. 参考程序

```

#include <reg51.h>
#include <absacc.h>

```

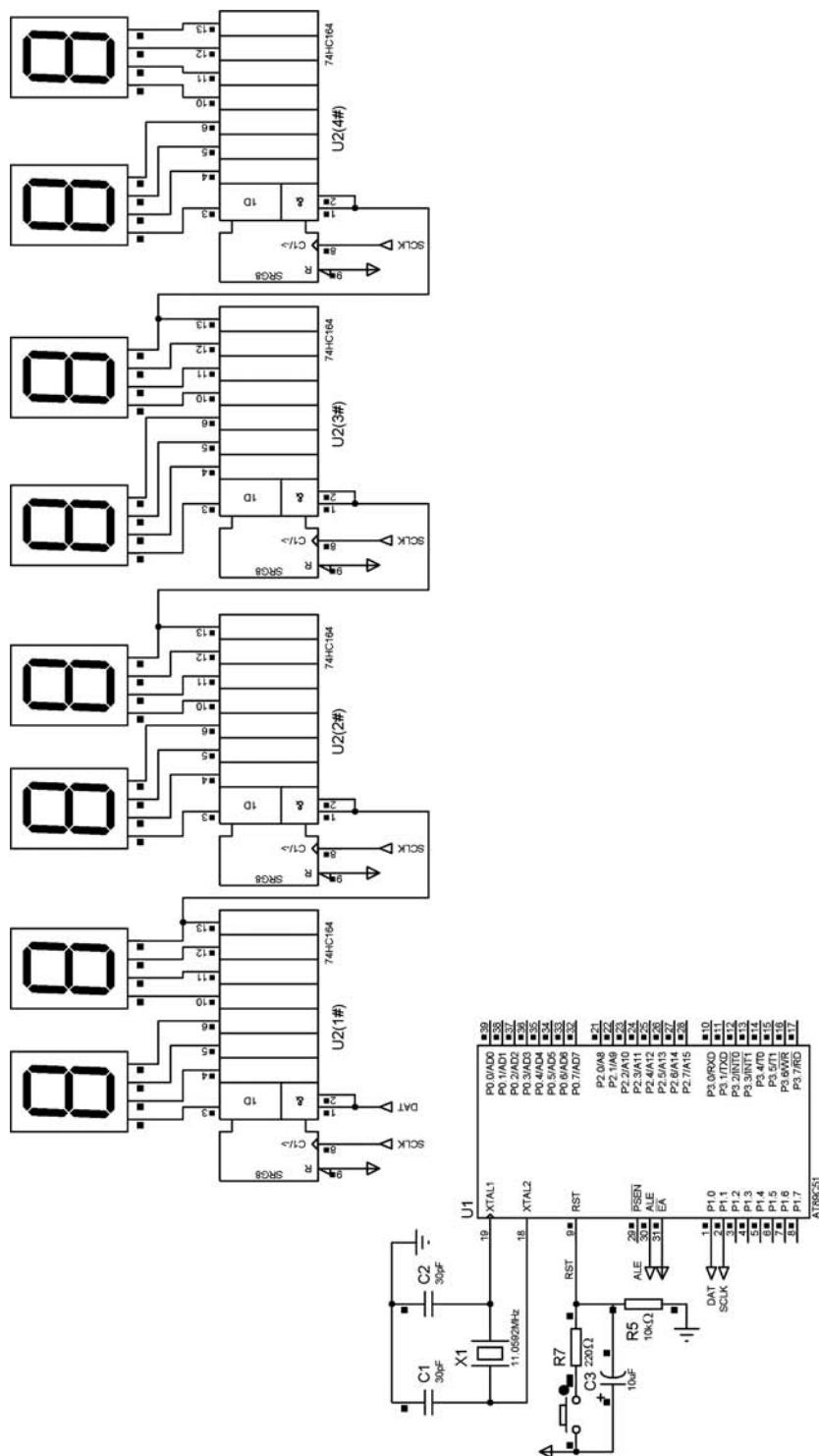


图 5-16 并行输出接口电路(74HC164)


```

sbit DAT = P1^0;
sbit CLK = P1^1;
void vDelay(unsigned int uiT)
{
    while(uiT--);
}
void vSendByte(unsigned char ucD)
{
    unsigned char i;
    for(i = 0; i < 8; i++)
    {
        CLK = 0; DAT = ucD & 0x01; CLK = 1;
        ucD = ucD >> 1;
    }
}
void main()
{
    unsigned char i;
    vSendByte(0x01);
    vSendByte(0x02);
    vSendByte(0x03);
    vSendByte(0x04);
    while(1);
}

```

5.3.2 74HC595 并行输出接口

1. 74HC595

1 位串入 8 位并出移位寄存器,可级联,其引脚见图 5-17。

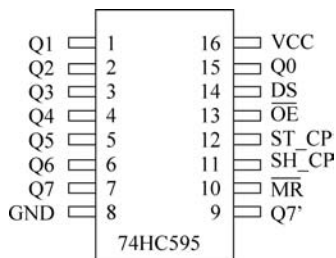


图 5-17 74HC595 引脚

引脚定义及功能:

- Q[7..0]——8 位并行输出。
- Q7'——级联输出。
- DS——串行数据输入。
- $\overline{\text{MR}}$ ——低电平时将 74LS595 数据清零。
- SH_CP——移位时钟,在上升沿将数据移位,下降沿寄存器数据保持。
- ST_CP——锁存时钟。
- $\overline{\text{OE}}$ ——输出使能。

2. 原理图

2 片 74HC595 扩展 16 位并行输出接口电路见图 5-18。

在图 5-18 中,

- $\overline{\text{OE}}$ ——输出使能信号,接地。
- $\overline{\text{MR}}$ ——复位端,接高电平。
- DS(U2)——单片机串行数据输入,接 P3.4。

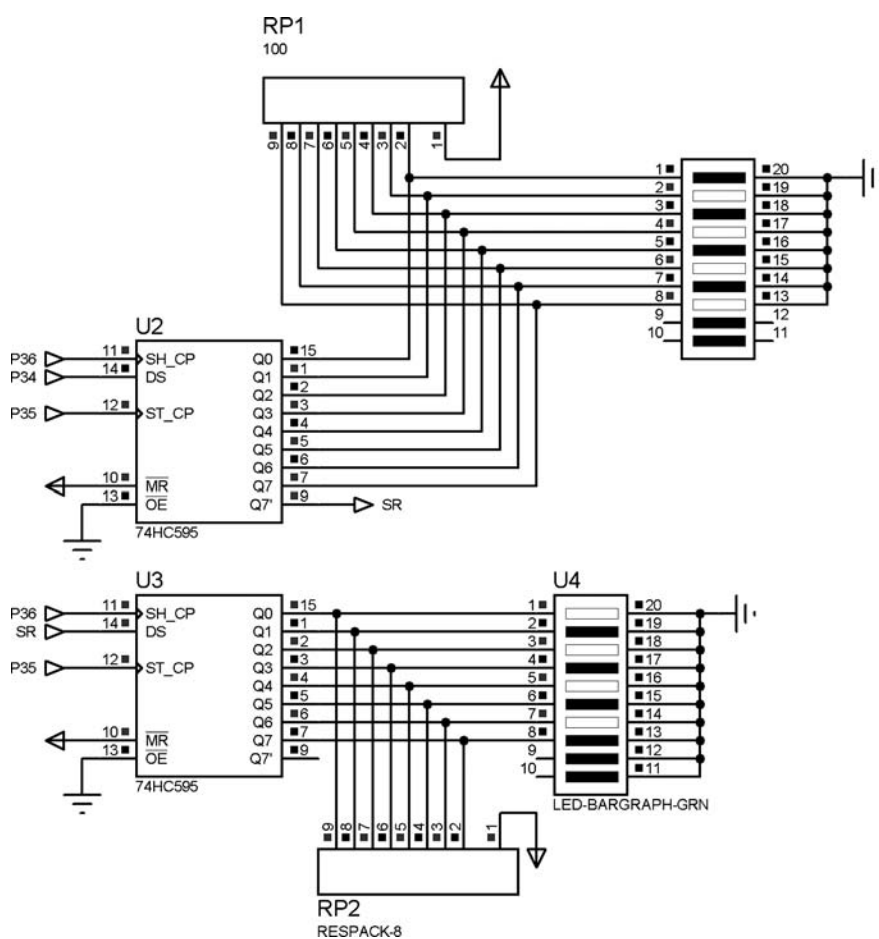


图 5-18 并行输出接口电路(74HC595)

- SH_CP——移位时钟,接 P3. 6。
- ST_CP——锁存时钟,接 P3. 5。
- Q0~Q7——并行输出。
- SR——74HC595 级联。

3. 参考程序

```
#include <reg51.h>
#include <intrins.h>
sbit SRCLK = P3^6;
sbit RCLK = P3^5;
sbit SER = P3^4;
void Hc595SendByte(unsigned char dat);
void Delay(unsigned int );
void main()
{
    Hc595SendByte(0x55);
    Hc595SendByte(0xaa);
    while(1);
}
```

```
}  
void Hc595SendByte(unsigned char dat)  
{  
    unsigned char i;  
    SRCLK = 1;  
    RCLK = 1;  
    for(i = 0; i < 8; i++)        //发送 8 位数  
    {  
        SER = dat >> 7;        //从最高位开始发送  
        dat <<= 1;  
        SRCLK = 0;              //发送时序  
        _nop_();  
        _nop_();  
        SRCLK = 1;  
    }  
    RCLK = 0; _nop_();_nop_();  
    RCLK = 1;  
}  
void Delay(unsigned int uiT)  
{  
    while(uiT--);  
}
```