

本章设计一个具有较完备功能的主控单元,可满足中等规模以上系统的开发需要。本主控单元将作为后续各章节的主控单元,存储器及各 I/O 接口设计和地址分配在本主控单元地址译码电路基础上进行。

## 3.1 主控单元

### 1. 原理图

主控单元包括时钟电路、复位电路、外部三总线扩展电路和系统片选地址译码电路,电路原理见图 3-1。设置锁存器 U3,便于在程序设计与调试过程中观察数据总线上数据的变化,了解程序运行状况。

### 2. 时钟电路

时钟是时序的基础,由片内高增益反相放大器和外部晶振构成振荡器,产生时钟脉冲。

在 XTAL1 和 XTAL2 引脚外接晶振,内部反向放大器自激振荡,产生振荡脉冲,见图 3-2。时钟发生器对振荡脉冲二分频,产生双相脉冲信号(P1 相+P2 相),构成时钟脉冲信号。 $f_{osc}$  为 1.2~12MHz,电容 C1、C2 取 30pF。

### 3. 复位电路

系统定义复位引脚 RST 持续 24 个振荡周期的高电平为系统复位信号,复位完成后使 RST 保持低电平。如图 3-3 所示为上电复位与按键复位电路。

电源 VCC、电阻 R7 与按键构成按键复位电路。电容 C3、电阻 R5 构成上电复位电路。完成复位后,通过电阻 R5 接地,RST 保持低电平,使系统进入正常的程序运行状态。

### 4. 后备电源

后备电源电路见图 3-4。利用 D1 和 D2 实现常规电源和后备电源切换。主电源正常工作时,D1 将后备电源 B1 隔离。当主电源低电压或故障时,B1 通过 D1 对系统供电。大容量电容 C4 具有储能作用。

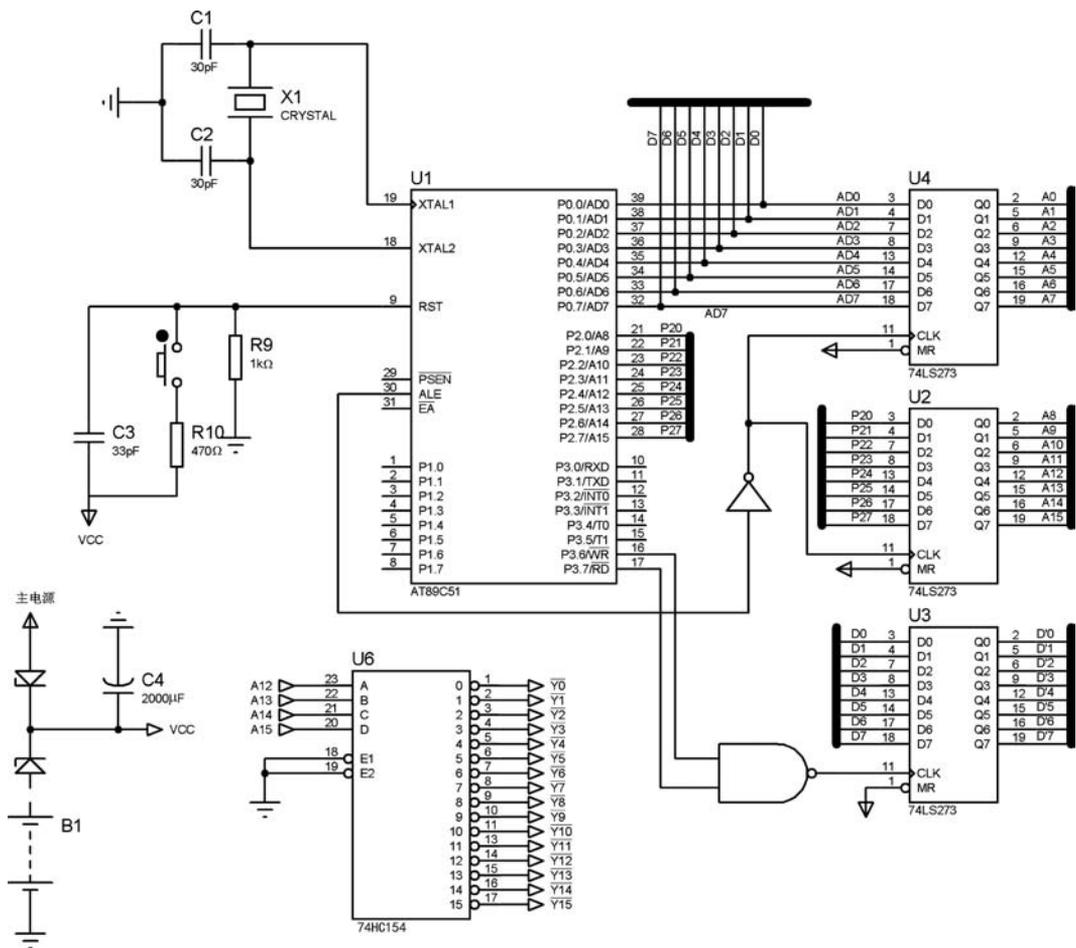


图 3-1 主控单元原理图

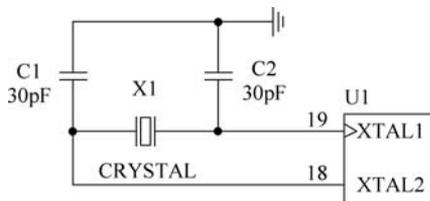


图 3-2 时钟电路

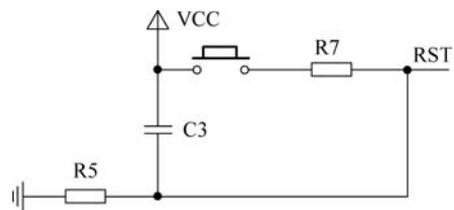


图 3-3 上电复位与按键复位电路

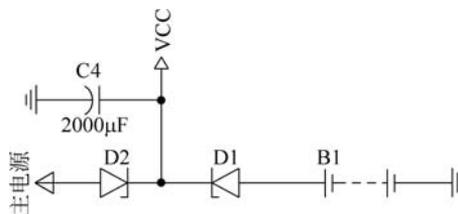


图 3-4 后备电源电路

### 3.2 地址译码

地址译码模块是系统设计的重要环节,需要对应用系统数据存储器的地址空间和 I/O 接口地址空间进行总体规划和分配,以便于存储器模块和 I/O 接口模块的设计与扩展。

常用地址译码器包括 74HC139、74HC138 和 74HC154。

#### 3.2.1 74HC139

##### 1. 74HC139 译码器

74HC139 为 2-4 译码器,引脚见图 3-5。

引脚说明:

- $\bar{E}$ ——使能端,低有效。
- B、A——2 位二进制码输入端。
- $\bar{Y}_0 \sim \bar{Y}_3$ ——编码信号输出端,低有效。

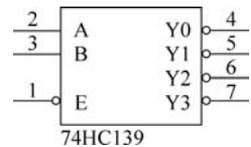


图 3-5 74HC139 引脚

当使能端有效时,对 2 位二进制输入码 B、A 进行译码,相应译码器输出信号  $\bar{Y}_i$  有效(=0),其他  $\bar{Y}_i$  无效(=1)。

##### 2. 地址译码电路

采用 1 片 74HC139 地址译码器,两位二进制代码输入连接系统地址线 A15 和 A14,提供 4 路片选信号  $\bar{Y}_0 \sim \bar{Y}_3$ ,将 64KB 地址空间划分为 4 个 16KB 区域,满足小规模系统设计需要,连接电路见图 3-6。

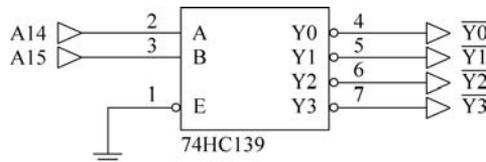


图 3-6 地址译码电路(74HC139)

$\bar{E}=0$ ,芯片常态使能,系统地址总线 A15、A14 连接译码器的 B、A 输入端,为系统提供片选信号,地址分配见表 3-1。

表 3-1 地址分配表

A15	A14	A13..A0	有效片选 $\bar{Y}_i$	端口地址范围
0	0	00..00	$\bar{Y}_0$	0000H
		11..11		3FFFH
0	1	00..00	$\bar{Y}_1$	4000H
		11..11		7FFFH
1	0	00..00	$\bar{Y}_2$	8000H
		11..11		BFFFH
1	1	00..00	$\bar{Y}_3$	C000H
		11..11		FFFFH

### 3.2.2 74HC138

#### 1. 74HC138 译码器

74HC138 为 3-8 译码器,引脚见图 3-7。

引脚说明:

- $S_1$ 、 $\overline{S_2}$ 、 $\overline{S_3}$ ——使能端,  $S_1$  高有效,  $\overline{S_2}$ 、 $\overline{S_3}$  低有效。
- $A_2$ 、 $A_1$ 、 $A_0$ ——3 位二进制码输入端。
- $\overline{Y_0} \sim \overline{Y_7}$ ——编码信号输出端,低有效。

当使能端有效时,对 3 位二进制输入码  $A_2$ 、 $A_1$ 、 $A_0$  进行译码,相应译码信号  $\overline{Y_i}$  有效(=0),其他  $\overline{Y_i}$  无效(=1)。

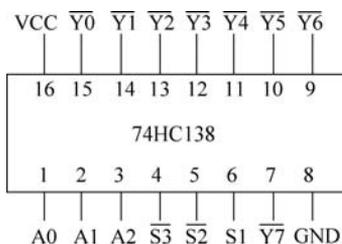


图 3-7 74HC138 逻辑结构

#### 2. 地址译码电路

采用 1 片 74HC138 地址译码器,3 位二进制代码输入连接系统地址线  $A_{15}$ 、 $A_{14}$  和  $A_{13}$ ,提供 8 路片选信号  $\overline{Y_0} \sim \overline{Y_7}$ ,将 64KB 地址空间划分为 8 个 8KB 地址空间,满足中规模系统设计需要,连接电路见图 3-8。

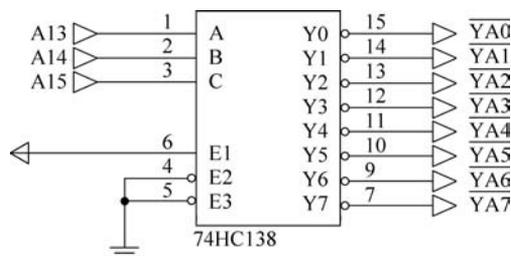


图 3-8 地址译码电路(74HC138)

$E_1/E_2/E_3=100$ ,芯片常态使能,系统地址总线  $A_{15}$ 、 $A_{14}$ 、 $A_{13}$  连接译码器的  $C$ 、 $B$ 、 $A$  输入端,为系统提供片选信号,地址分配见表 3-2。

表 3-2 地址分配表

$A_{15}$	$A_{14}$	$A_{13}$	$A_{12}..A_0$	有效片选 $\overline{Y_i}$	端口地址范围
0	0	0	00..00	$\overline{Y_0}$	0000H
			11..11		1FFFH
0	0	1	00..00	$\overline{Y_1}$	2000H
			11..11		3FFFH
0	1	0	00..00	$\overline{Y_2}$	4000H
			11..11		5FFFH
0	1	1	00..00	$\overline{Y_3}$	6000H
			11..11		7FFFH
1	0	0	00..00	$\overline{Y_4}$	8000H
			11..11		9FFFH
1	0	1	00..00	$\overline{Y_5}$	A000H
			11..11		BFFFH

续表

A15	A14	A13	A12..A0	有效片选 $\overline{Y_i}$	端口地址范围
1	1	0	00..00	$\overline{Y_6}$	C000H
			11..11		DFFFH
1	1	1	00..00	$\overline{Y_7}$	E000H
			11..11		FFFFH

### 3.2.3 74HC154

#### 1. 74HC154

74HC154 为 4-16 译码器,引脚见图 3-9。

引脚说明:

- $\overline{G1}/\overline{G2}$ ——使能端,低有效。
- DCBA——4 位二进制码输入。
- $\overline{Y_0} \sim \overline{Y_{15}}$ ——译码信号输出。

#### 2. 地址译码电路

地址译码电路(见图 3-10)采用 1 片 74HC154,系统地址总线 A15、A14、A13、A12 连接译码器的 D、C、B、A 输入端,产生 16 路片选信号  $\overline{Y_0} \sim \overline{Y_{15}}$ ,将 64KB 地址空间划分为 16 个 4KB 地址空间,为系统提供片选信号,地址分配如表 3-3。

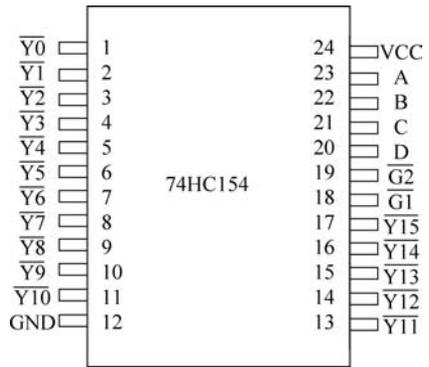


图 3-9 74HC154 引脚

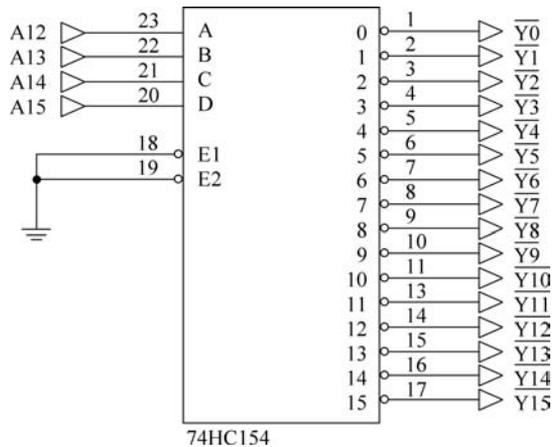


图 3-10 地址译码电路(74HC154)

本书后续章节选择该地址译码方案,将 RAM 地址和后续各章所设计的 I/O 接口地址统一安排,见表 3-3。

表 3-3 地址分配表

A15	A14	A13	A12	A11..A0	片选 $\overline{Y_i}$	端口地址范围	备注
0	0	0	0	00..00	$\overline{Y_0}$	0000H	4.2 节: RAM,2KB
				11..11		0FFFH	

续表

A15	A14	A13	A12	A11..A0	片选 $\overline{Y_i}$	端口地址范围	备 注
0	0	0	1	00..00	$\overline{Y_1}$	1000H	5.1.2 节: 8255A 扩展 I/O
				11..11		1FFFH	
0	0	1	0	00..00	$\overline{Y_2}$	2000H	6.2 节: 74HC244 中断扩展
				11..11		2FFFH	
0	0	1	1	00..00	$\overline{Y_3}$	3000H	14.1 节: LCD1602
				11..11		3FFFH	
0	1	0	0	00..00	$\overline{Y_4}$	4000H	
				11..11		4FFFH	
0	1	0	1	00..00	$\overline{Y_5}$	5000H	
				11..11		5FFFH	
0	1	1	0	00..00	$\overline{Y_6}$	6000H	
				11..11		6FFFH	
0	1	1	1	00..00	$\overline{Y_7}$	7000H	7.2.2 节: CD4052 通信端口
				11..11		7FFFH	
1	0	0	0	00..00	$\overline{Y_8}$	8000H	7.2.3 节: CD4051 通信端口
				11..11		8FFFH	
1	0	0	1	00..00	$\overline{Y_9}$	9000H	10.3 节: 8253 看门狗
				11..11		9FFFH	
1	0	1	0	00..00	$\overline{Y_{10}}$	A000H	11.2 节: ADC0809
				11..11		AFFFH	
1	0	1	1	00..00	$\overline{Y_{11}}$	B000H	
				11..11		BFFFH	
1	1	0	0	00..00	$\overline{Y_{12}}$	C000H	
				11..11		CFFFH	
1	1	0	1	00..00	$\overline{Y_{13}}$	D000H	
				11..11		DFFFH	
1	1	1	0	00..00	$\overline{Y_{14}}$	E000H	
				11..11		EFFFH	
1	1	1	1	00..00	$\overline{Y_{15}}$	F000H	
				11..11		FFFFH	

### 3.3 Proteus 仿真

#### 1. 仿真原理图

主控单元仿真连接如图 3-11 所示。

系统对外部程序存储器、外部数据存储器 and 片外 I/O 接口访问时,选择不同的地址范围,可从连接在译码器输出端的 LED 指示灯读出当前有效  $\overline{Y_i}$ ,从连接在数据总线上的 LED 指示,读出当前数据总线上 8 位数据值,便于系统调试。

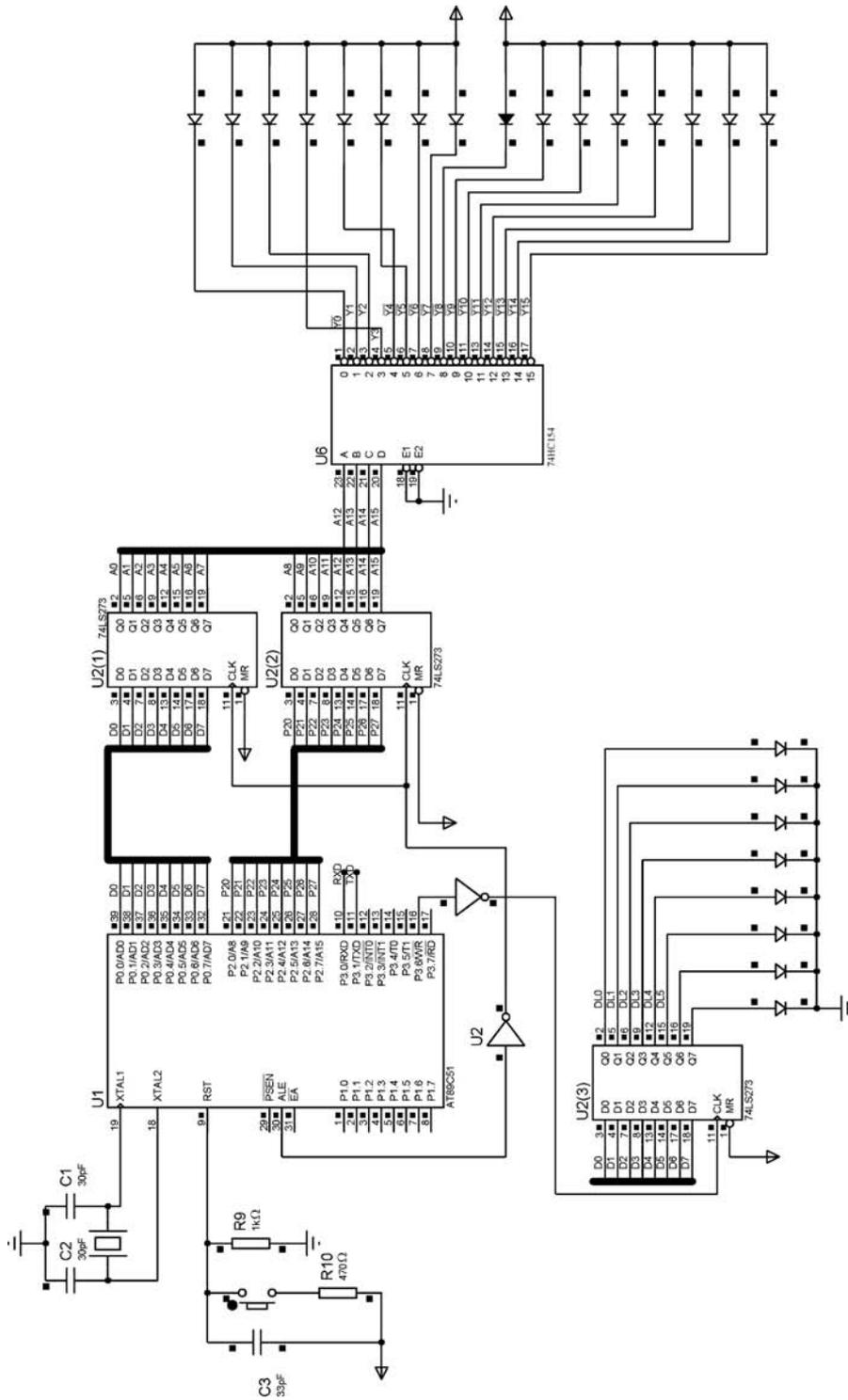


图 3-11 主控单元仿真电路

## 2. 参考程序

```
# include "reg51.h"
# include < absacc.h >
# define DE154Y0 XBYTE[0x0000]

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

void main()
{
    unsigned char i;
    unsigned char * ucAdd;
    ucAdd = &DE154Y0;
    while(1)
    {
        for(i = 0; i < 16; i++)
        {
            * ucAdd = 0x55; vDelay(10000);
            * ucAdd = 0xaa; vDelay(10000);
            ucAdd = ucAdd + 0x1000;
        }
    }
}
```

