

# 基于 AT89S52+CPLD 综合课程设计实验板的开发<sup>①</sup>

杨聪锬<sup>1</sup>, 杨雪芹<sup>2</sup>, 杨雪辉<sup>3</sup>

<sup>1</sup>(西安工业大学 电子信息工程学院, 西安 710021)

<sup>2</sup>(西安邮电大学 通信与信息工程学院, 西安 710121)

<sup>3</sup>(山东航天电子技术研究所, 烟台 264670)

**摘要:** 单片机技术和 CPLD 技术已经成为高校电子类学生所必备的专业知识. 该实验板结合在校大学生的课程内容选用了当前学校教学中较为常用的 AT89S52 单片机和 ALTERA 公司生产的 EPM7128SLC84-6, 并设计了常用的数码管显示、1602A 液晶显示、键盘、日历时钟等接口电路. 通过大量实例学习可使得学生能够在短时间之内快速掌握单片机 C 语言及 Verilog HDL 的编程技巧.

**关键词:** 单片机; CPLD; 实验板; C 语言

## Exploitation of Experimental Board Based on AT89S52 and CPLD for Integrated Course Design

YANG Cong-Kun<sup>1</sup>, YANG Xue-Qin<sup>2</sup>, YANG Xue-Hui<sup>3</sup>

<sup>1</sup>(Xi'an University of Technology, Electronic Information Engineering College, Xi'an 710021, China)

<sup>2</sup>(Xi'an University of Posts & Telecommunications, School of Telecommunication and Information Engineering, Xi'an 710121, China)

<sup>3</sup>(Shandong Institute of Aerospace Electronics Technology, Yantai 264670, China)

**Abstract:** Single chip microcomputer technology and CPLD technology has become the necessary professional knowledge of electronic class for students in Colleges and universities. The experimental board combined with curriculum content of college students selected AT89S52 microcontroller and Altera company production EPM7128SLC84-6. which are commonly used in the current school teaching. And it designs the common used digital tube display, 1602A LCD display, keyboard, and calendar clock interface circuit. Through a large number of examples of learning, students can quickly grasp the microcontroller C language and HDL Verilog programming skills in a short period of time.

**Key words:** MCU; CPLD; experimental board; C language

## 1 引言

近年来, 为提高全民综合素质, 全国高校实施不同程度的扩招, 但因学校硬件设施没有及时跟上, 使在校大学生缺少实验机会, 减少了学生实践, 必然造成实践动手能力差<sup>[1]</sup>. 然而社会科技一直快速发展, 如何让在校大学生通过实验来提高自己的学习实践能力, 以致可以短时间胜任专业工作, 这是电子类专业急需解决的问题.

单片机+CPLD 学习开发板解决了这一问题. 单片机+CPLD 学习开发板综合了单片机的强大控制能力、丰富指令和 CPLD 丰富的 I/O 口, 功能实现简洁等优点,

使系统开发成本低, 功能强和更简易<sup>[2-4]</sup>. 广大电子类学生通过实验既学习单片机 Keil C 编程, 又应用 Verilog HDL 语言掌握可编程逻辑器件硬件设计, 同时理解应用单片机和 CPLD 组合实现系统应用开发的思想及方法.

## 2 系统总体方案设计

系统由 AT89S52 单片机<sup>[5]</sup>和 MAX7000S 系列 CPLD(EPM7128SLC84-6)组成开发板控制部分. 系统中, 单片机采用总线控制 CPLD 的读、写数据命令, 即单片机把 CPLD 当作外部 RAM 进行接口的连接设计,

<sup>①</sup> 基金项目:西安邮电大学青年教师科研基金(205010321)

收稿时间:2015-12-10;收到修改稿时间:2016-01-27 [doi:10.15888/j.cnki.csa.005297]

单片机对 CPLD 数据的发送和接收可以通过发送读写外部 RAM 的地址进行控制, 而 CPLD 通过检测单片机的读、写外部 RAM 时序来翻译单片机控制命令, 并作出相应处理. 此时, 单片机外围接口已全部通过 CPLD 进行扩展. CPLD 的加入使系统去掉译码器、锁存器和缓冲器等分立元件, 大大简化硬件连线, 提高可靠性. 这样, 不仅有效克服了单纯以单片机为控制核心或单纯以 CPLD 为控制核心的系统缺点, 而且二者的优点被最大限度地利用. 系统框图如图 1 所示, 系统外围接口电路主要由键盘接口、LED 发光管接口、数码管显示接口、蜂鸣器接口、I2C 存储器件、日历时钟芯片、LCD 液晶、RS232 电平转换和 ISP/JTAG 接口组成.

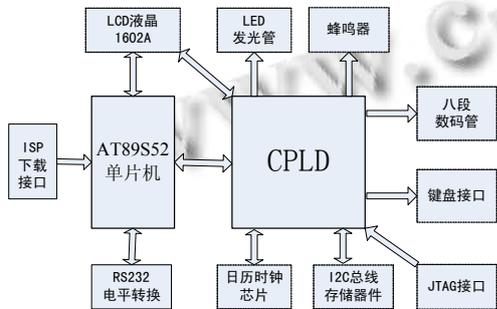


图 1 实验板系统整体框图

### 3 系统单元电路设计

#### 3.1 系统电源电路设计

系统电源是为各元器件提供能量的模块, 是实验板的重要组成部分. 系统电源质量的好坏直接影响各主芯片及元器件的工作稳定性. 如图 2 所示, 电源模块中 DC\_CON 为外接的 9V 直流电子稳压源, 但电子稳压电源是由工频 50Hz, 电压 220V 的市电经降压变换输出, 要求输入 9V 电源输入电压平滑, 所以在输入级接入 47uF/16V 的电解电容和 0.1uF 的瓷片电容, 防止电源电压波动及高频干扰, 同时考虑外围电路负载电压波动较大, 在输出端并联 22uF/25V 和 0.1uF 瓷片电容.

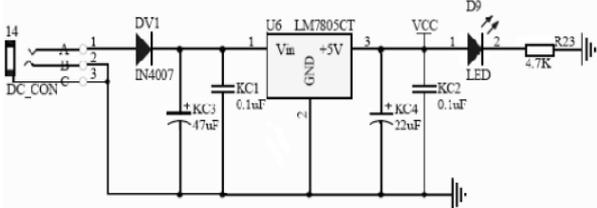


图 2 系统供电电源电路设计

#### 3.2 CPLD 电路设计

考虑当前高校课堂教学中可编程逻辑器件选用的多为 Altera 公司生产的主流芯片, 所以选用 Altera 公司 MAX7000S 系列 EPM7128SLC84-6 作为实验板中的 CPLD<sup>[6]</sup>.

EPM7128SLC84-6 的 84 管脚除去电源 VCC, 引脚地, 下载引脚, 时钟引脚外的 60 引脚全部被用作普通 I/O, 分别分配到键盘、LED 发光管、数码管、单片机等外围接口电路中.

#### 3.3 单片机晶振及复位电路设计

文中主控制器选用美国 ATMEL 公司生产的低电压, 高性能、8 位 AT89S52 单片机, 内含 8KB 可反复擦写 Flash 只读程序存储器及 256B 随机存取数据存储器(RAM), 兼容标准 MCS-51 指令系统, 内置通用 8 位中央处理器和 Flash 存储单元.

如图 3 所示, 复位电路采用 RC 电平按键方式复位,  $R=10K, C=10uF$ , 复位时间  $T=RC=100ms$ ; 晶振电路采用无源晶振, 需在晶振两侧加入两个 30pF 的电容作为起振电容.

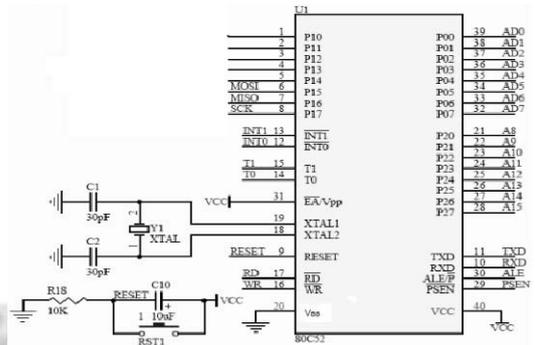


图 3 单片机晶振及复位电路

#### 3.4 数码管显示电路设计

如图 4 所示为数码管显示电路设计, 电路选用两个 4 位 8 段共阳显示动态数码管, 其中 A~DP 为段码引脚, 接 CPLD 的 IO 口; G1~G4 为位选引脚, 接 PNP 三极管的集电极引脚, 数据显示采用动态扫描方法.

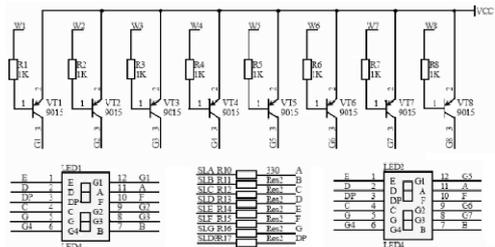


图 4 数码管显示接口电路设计

### 3.5 LED 灯电路设计

如图 5 所示设计了 8 个 LED 发光管, 按红、绿、黄顺序依此排列, 如此安排可以让使用者既可以做交通灯实验, 也可以做流水灯实验. 由于 LED 发光管采用反向接法, 所以只需 CPLD 向 LED 发光管任一引脚发送低电平信号, 即可点亮相应发光管.

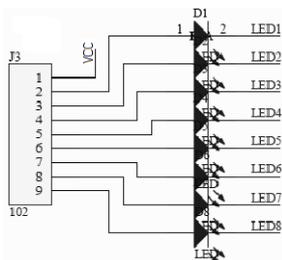


图 5 LED 灯电路设计

### 3.6 1602A 液晶显示电路设计

如图 6 所示为 1602 液晶接口设计电路, 是一个间距为 2.54mm 的 16 脚单排插孔座, 上插 1602A 液晶即可. 1602 液晶接口 1 脚接电位器一端, 供用户手动调整液晶偏压. BLA: 背光源正极, 接 +5V; BLK: 背光源负极, 接地; RS 接 A13, 决定传输类型是数据/命令(1 表示数据, 0 表示命令); RW 接 A14, 决定控制信号读/写命令(1 表示读, 0 表示写); E 接 A15, 使能端, 高电平有效; DB0~DB7 数据端口, 接 AD0~AD0.

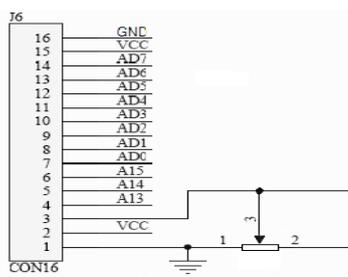


图 6 1602A 液晶显示电路设计

### 3.7 键盘接口电路设计

如图 7 所示, 考虑到做实验时的方便实用性, 实验板中的键盘单元设计为由 4\*4 矩阵键盘和 4 个独立按键组成, 且都只与 CPLD 相连. 在所需按键数量较少时采用简单的独立按键即可, 所需按键较多时, 则采用 4\*4 矩阵键盘.

独立按键中, 按键一端接地, 一端和 CPLD 相连, 只要有按键按下, 则 CPLD 的 IO 口即和地连通, 电平被拉低, 这样便可以检测到独立键盘的键值.

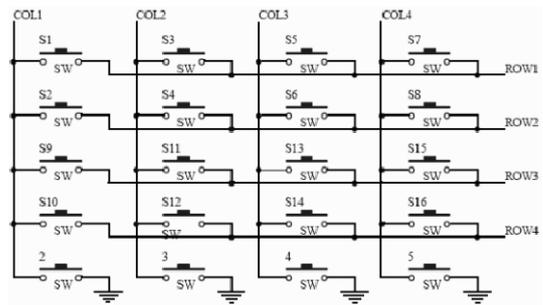


图 7 键盘接口电路设计

### 3.8 蜂鸣器接口电路设计

如图 8 所示为蜂鸣器接口电路设计, 蜂鸣器选用交流蜂鸣器, 考虑 I/O 驱动能力, 系统采用三极管 VT9 驱动蜂鸣器. 当 BUZZ 输出高电平时, 蜂鸣器不响; 当 BUZZ 输出低电平时, 蜂鸣器发出声响. 只要控制 BUZZ 输出高低电平时间, 就可以让蜂鸣器发出悦耳的音乐.

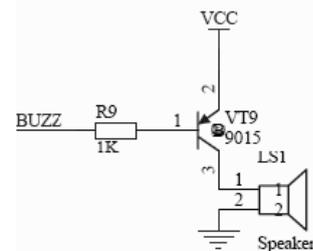


图 8 蜂鸣器接口电路设计

### 3.9 E<sup>2</sup>PROM 存储器电路设计

I<sup>2</sup>C (Inter-Integrated Circuit) 总线是两线式串行总线, 用于连接微控制器及其外围设备, 其特点是简单性和有效性. I<sup>2</sup>C 总线占用空间小, 减少电路板空间和芯片管脚数量, 降低互连成本.

I<sup>2</sup>C 总线传送数据过程共有三类信号: 开始信号、结束信号和应答信号.

开始信号: SCL 为高电平时, SDA 由高电平向低电平跳变, 开始传送数据.

结束信号: SCL 为高电平时, SDA 由低电平向高电平跳变, 结束传送数据.

应答信号: 接收数据的 IC 在接收到 8b 数据后, 向发送数据的 IC 发出特定的低电平脉冲, 表示已收到数据. 若未收到应答信号, 则判断为受控单元出现故障. 如图 9 所示为 I<sup>2</sup>C 总线的操作时序.

如图 10 所示为 AT24C256 与 CPLD 的接口电路, 二者之间通信采用 I<sup>2</sup>C 总线. 为了操作方便, AT24C256

的 A0、A1、A2 均接地，则其操作地址最后 3 位就被固定为 000，因此编程时最后三位地址就必须写成 000，同时由于 SDA 和 SCL 也是开漏输出，所以在其引脚上加了 10K 的上拉电阻。

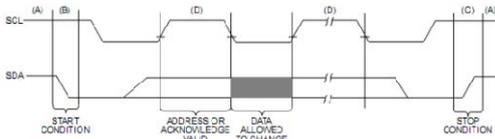


图 9 I<sup>2</sup>C 总线操作时序

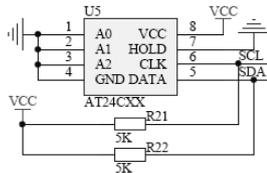


图 10 E<sup>2</sup>PROM 存储芯片 AT24C04 电路设计

### 3.10 日历/时钟接口电路设计

PCF8563 是 PHILIPS 推出的一款工业级内含 I<sup>2</sup>C 总线的低功耗多功能时钟/日历芯片，具有报警、定时器、时钟输出功能以及中断输出功能，能完成各种复杂的定时服务，甚至可以提供看门狗功能。如图 11 所示为日历/时钟电路设计，晶振为 32.768KHz，为增加驱动，SDA 和 SCL 端口接入 R20、R19 两个上拉电阻。

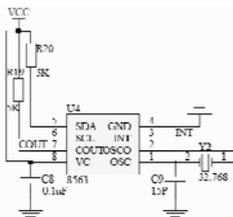


图 11 日历/时钟 PCF8563 电路设计

### 3.11 RS232 电平转换电路

RS-232 是用于串行通信的接口标准。RS-232 采用负逻辑(即：逻辑"1"为-3 到-15V；逻辑"0"，为+3 到+15V)，单端发送/接收的半双工非平衡传输方式，将±5V 到±15V 的电压转换成 TTL 电平。如图 12 所示为 RS-232 电平转换电路设计，仅需要 5 个 0.1uF 电容(用作倍升电压存储)作为外部元件即能工作。

### 3.12 单片机 ISP 下载接口电路设计

ISP 是 In System Programming(在系统编程)<sup>[7-9]</sup>。AT89S52 单片机编程采用 ISP 编程，可以通过下载线直接下载或擦除程序，不需要拔出来，也不需要专门

的编程器，提供极大方便，减少因频繁拔插而造成的管脚损坏几率。如图 13 所示，单片机在编程模式 P1.7、P1.6、P1.5 分别被定义为 SCK、MISO、MOSI 三个信号线，以供下载程序使用。只要准确接入 SCK、MISO、MOSI、RESET 信号线，单片机通过 USB 下载线和电脑相连就可以下载程序了。

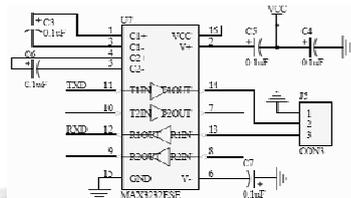


图 12 基于 RS-232 标准的电平转换电路

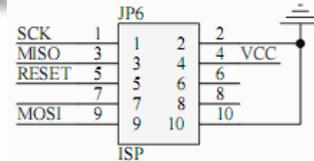


图 13 单片机 ISP 下载接口电路设计

### 3.13 JTAG 接口电路

JTAG 是一种国际标准测试协议(IEEE 1149.1 兼容)，可用于实现 ISP，对 FLASH 等器件进行编程。相关 JTAG 引脚定义为：TCK 为测试时钟输入；TDI 为测试数据输入，数据通过 TDI 从 JTAG 接口输入；TDO 为测试数据输出，数据通过 TDO 从 JTAG 接口输出；TMS 为测试模式选择，用来设置 JTAG 接口处于某种特定测试模式，接口电路如图 14 所示。

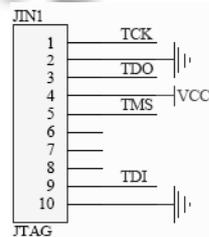


图 14 JTAG 调理电路设计

## 4 实验板调试

实验板调试是本文设计的最后阶段，为验证 PCB 实验板中各元器件之间连线的正确性及系统设计的合理性，设计好的 PCB 图必须先发给专业制作 PCB 板的生产厂家做成实际的电路板，然后根据自己的设计买来对应的元器件，并通过手工或机器工具把相应元器件焊到实验板上，最后用各种测量工具进行调试验证。

通过检查改错, 最终做成无误的实验板成品.

#### 4.1 电源和地的短路检查

对于厂家发回来的实验板, 在焊元器件之前必须进行电源和地两种信号之间的短路检查. 这是因为: 首先, 即使 PCB 图中所连接的线路是正确的, 但是由于生产工艺的原因, 使得制作出来的电路板电源和地之间造成了短路连接. 其次, 电源和地之间一旦连接就会造成短路, 短路的结果是使线路之中的电流过大, 导致实验板上电即会烧坏某些元器件, 造成不必要的损失; 最后, 一旦元器件焊好之后才来检查电源与地之间的连接, 此时由于元器件的存在, 使得检查起来更加困难. 所以当拿到电路板的第一步就是用“万用表”的测二极管档进行电源和地的短路检查.

#### 4.2 电源供电情况检查

电路板焊好元器件之后, 一块实验板成品就出来了, 接下来用外接的电子稳压电源给实验板供上 9V 的直流稳压电源. 首先观察实验板的电源指示灯有没有亮, 指示灯亮则说明实验板的供电电源已经供上电了, 接下来再用万用表对 LM7805 的输出电压进行测量, 保证了供电模块输出的电压值为正常的 5V, 这样以使得实验板上的各种要求 5V 电源供电的元器件工作正常. 方便程序的编写及实验的进行.

#### 4.3 实验板软件调试

由于实验板中液晶接口被设计为总线的方式进行通信, 所以 1602A 液晶既和单片机相连、又和 CPLD 相连. 当仅用单片机对 1602A 液晶进行控制显示时, 就需要编写 1602A 液晶的显示控制程序. 如下所示为单片机与 1602A 连接的液晶初始化程序.

```
sbit RS = P2^5; //1602A 液晶数据/命令控制端
sbit RW = P2^6; //1602A 液晶读、写控制端
sbit EN = P2^7; //1602A 液晶使能控制端
/***** 1602A 初始化程序 *****/
void lcd_initial()
{
    BL = 0;
    delay_ms(15);
    lcd_wcmd(0x38);
    delay_ms(5);
    lcd_wcmd(0x38);
    delay_ms(5);
```

```
    lcd_wcmd(0x38);
    delay_ms(5);
    lcd_wcmd(0x38); //显示模式设置, 16*2 显示, 5*7 点阵,
    // 8 位数据接口
    delay_ms(5);
    lcd_wcmd(0x08); //显示关
    delay_ms(5);
    lcd_wcmd(0x01); //清屏
    delay_ms(5);
    lcd_wcmd(0x0e); //显示开, 并显示光标
    delay_ms(5);
    lcd_wcmd(0x06);
    delay_ms(5);
    // lcd_wcmd(0x07); //写一个字符, 整屏显示左移
    // delay_ms(5);
    // lcd_wcmd(0x10);
    delay_ms(5);
```

在实验板中, 数码管接口设计为与 CPLD 相连, 如下所示为数码管动态显示程序.

```
module dled(seg,sl,clock);
output [7:0] seg;
output [7:0] sl;
input clock;
reg[7:0] seg_reg;
reg[7:0] sl_reg;
reg[3:0] disp_dat;
reg[23:0] count;
always@(posedge clock)
begin
    count=count+1;
end
always@(count[10]) //定义显示数据触发事件
begin
    case(count[12:10])
        3'h0:disp_dat=4'b0001; //要显示的数据
        3'h1:disp_dat=4'b0010;
        3'h2:disp_dat=4'b0011;
        3'h3:disp_dat=4'b0100;
        3'h4:disp_dat=4'b0101;
        3'h5:disp_dat=4'b0110;
        3'h6:disp_dat=4'b0111;
```

```

        3'h7:disp_dat=4'b1000;
    endcase
    case(count[12:10])
        3'h0:sl_reg=8'b1111_1110; //显示相应数据的
段码
        3'h1:sl_reg=8'b1111_1101;
        3'h2:sl_reg=8'b1111_1011;
        3'h3:sl_reg=8'b1111_0111;
        3'h4:sl_reg=8'b1110_1111;
        3'h5:sl_reg=8'b1101_1111;
        3'h6:sl_reg=8'b1011_1111;
        3'h7:sl_reg=8'b0111_1111;
    endcase
end
always@(disp_dat) //显示解码输出
begin
    case(disp_dat)
        4'h0:seg_reg=8'hc0; //显示 0
        4'h1:seg_reg=8'hf9; //显示 1
        4'h2:seg_reg=8'ha4;
        4'h3:seg_reg=8'hb0;
        4'h4:seg_reg=8'h99;
        4'h5:seg_reg=8'h92;
        4'h6:seg_reg=8'h82;
        4'h7:seg_reg=8'hf8;
        4'h8:seg_reg=8'h80;
        4'h9:seg_reg=8'h90;
        4'ha:seg_reg=8'h88;
        4'hb:seg_reg=8'h83;
        4'hc:seg_reg=8'hc6;
        4'hd:seg_reg=8'ha1;
        4'he:seg_reg=8'h86; //显示 e
        4'hf:seg_reg=8'h8e; //显示 f
    endcase
end
assign seg=seg_reg; //输出数码管解码结果
assign sl=sl_reg; //输出数码管选择
endmodule

```

在实验板设计中, 单片机和 CPLD 的接口设计思想是把 CPLD 当作单片机的外部 RAM 来进行控制的, 如下所示为单片机与 CPLD 的综合通信调试程序。

单片机通信部分的程序:

```

#include <AT89X52.H>
#include <absacc.h>
#define KEY_RD XBYTE[0xff00] //定义键盘
操作地址
#define LED_WR XBYTE[0xff02] //定义 LED
操作地址
#define uchar unsigned char
#define uint unsigned int
//void wr_cpld(uchar dat); //写 CPLD 函数
void getkey(uchar keyin); //键值处理子程序
uchar rd_cpld(); //读 CPLD 函数
void delay()
{
    uchar i;
    for(i=1; i<120; i++);
}
uchar led_buf;
void main(void)
{
    uchar key_buf,buffer; //定义中间变量
    key_buf = 0x0f;
    led_buf = 0xff;
    while(1)
    {
        buffer = rd_cpld(); //读取键盘状态
        delay();
        if(!(key_buf == buffer) && (buffer != 0x0f))
        //若有新键按下
        {
            key_buf = buffer; //保存该时
            刻键盘状态
            getkey(key_buf); //按键处理
            更新显示数据
            LED_WR = 0x22; //数据送
            CPLD, 更新显示
            delay();
        }
    }
}
void getkey(uchar keyin)

```

```

{
    switch(keyin)
    {
        case 0x0e: led_buf = 0xfe;
            break;
        case 0x0d: led_buf = 0xfc;
            break;
        case 0x0b: led_buf = 0xe0;
            break;
        case 0x07: led_buf = 0x00;
            break;
        default: led_buf = 0x01;
            break;
    }
}
/***** 写 CPLD 子程序, dat 为
存放数据变量 *****/
void wr_cpld(uchar dat)
{
    LED_WR = dat;
}
/***** 读 CPLD 子程序
*****/
uchar rd_cpld()
{
    uchar temp;
    temp = KEY_RD;
    return (temp);
}
CPLD 通信部分的程序:
#define key_addr 16'hff00
#define led_addr 16'hff01
#define no_rw 2'h0
#define 'key_rw 2'h1
#define 'led_rw 2'h2
module key_led(mcu_data, mcu_addr, keyin, led, wr, rd,
ale);
input [7:0] mcu_addr; //定义地址总线
inout [7:0] mcu_data; //数据总线
input [3:0] keyin; //键盘输入口
output [7:0] led; //output int; //定义键盘中断输出口
input wr; //定义写信号线
input rd; //读信号线
input ale; //地址锁存允许信号
reg [1:0] addr_sel; //总线状态寄存器
reg [3:0] led_reg; //LED 灯状态寄存器
reg [7:0] mcu_data_reg; //定义总线寄存器
reg [3:0] key_status; //键盘状态寄存器
//reg int_reg; //中断信号寄存器
always@(negedge ale)
begin
    if({mcu_addr,mcu_data} == 16'hff00)
        addr_sel = 2'h1; //是对键盘进行操作吗?
    else if({mcu_data,mcu_data} == 16'hff01)
        addr_sel = 2'h2; //是对 LED 灯进行操作吗?
    else
        addr_sel = 2'h0; //否则挂起
end
/*always@(keyin) //键盘中断处理
begin
    if(keyin == key_status)
        int_reg = 1;
    else
        int_reg = 1;
end*/
always@(negedge wr)
begin
    if(addr_sel == 2'h2)
        led_reg = mcu_data;
end
always@(negedge rd)
begin
    if(addr_sel == 2'h2)
        mcu_data_reg = led_reg;
    else if(addr_sel == 2'h1)
        begin
            mcu_data_reg = keyin;
            key_status = keyin;
        end
end

```

```
end
assign mcu_data = ((rd==1'b1) || (addr_sel== 2'h0)) ?
8'hz : mcu_data_reg;
assign led = led_reg;
//assign int = int_reg;
Endmodule
```



图 15 实验板调试结果

### 参考文献

- 1 代芬,王卫星,邓小玲,姜晟,孙道宗,陆健强.单片机综合实验开发板设计.实验室研究与探索,2010,(8):213-215.
- 2 田开坤,徐海霞.基于 CPLD 的单片机接口设计.现代电子技术,2008,(2):34-36.
- 3 樊京,郭俊杰.单片机+CPLD结构体系在电子设计中的应用.单片机与嵌入式系统应用,2002,(6):57-59.
- 4 尹常永.采用 CPLD 器件扩充单片机 I/O 口.沈阳电力高等专科学校学报,2000,(1):42-44.
- 5 周丽荣.基于 AT89S52 的单片机开发板设计.科学之友,2011,(22):64,73.
- 6 王彦芳,王小平,樊金生,刘玉红.用可编程逻辑器件 EPM7128SLC84 实现细分电路.半导体情报,2001,(1):44-47.
- 7 赵希才,陆虹雨.利用单片机实现 CPLD 的在系统编程.单片机与嵌入式系统应用,2001,(11):10-11,23.
- 8 吴桂华.在系统可编程逻辑器件的设计方法及其应用.中南民族大学学报(自然科学版),2002,(2):51-53,57.
- 9 刘红,韦穗.ISP 器件在高速通信系统中的应用.现代电子技术,2005,(16):1-2.