

单片机课程的教学与实验改革

方怡冰

(集美大学 信息工程学院, 福建 厦门 361021)

摘 要:单片机课程是电子信息类专业的重要课程,本文介绍了单片机课程的改革实践与体会,着重从教学方法,实验手段两方面阐述改革的方法,并以 SPI 模块为例介绍 PROTEUS 软件开发过程和设计分析过程,实践证明能加深学生对 PIC16F877 单片机工作原理的理解,也锻炼了学生进行软硬件综合开发的能力。

关键词:单片机;教学改革;实验手段;PICC;PROTEUS

中图分类号:G642.423;TP368.1

文献标识码:A

文章编号:1008-0686(2006)03-0076-04

Reform on Teaching and Experiment of “Single Chip Microcomputer” Course

FANG Yi-bin

(School of Engineering, Jimei University, Xiamen 361021, China)

Abstract: “Single Chip Microcomputer” course is an important course in the Electronics Information Specialty. In this paper, the practice, experience and achievements of teaching reform in the course are introduced, and also the teaching method and experiment measures in detail. And by example on the SPI module, it introduces the software PROTEUS development process and design analysis process, it has proved that the new method not only let students understand the PIC16F877 operating principle, but also practice their capability to perform multiple projects.

Keywords: single chip microcomputer; teaching reform; experiment measures; PICC; PROTEUS

1 改革过程

“单片机原理与应用”是大多数高校电子信息类专业必开的课程,多数专业选择 MCS-51 作为授课对象,与之配套的教材、实验设备丰富多彩,各种程序范例、功能子程序应有尽有,极大地方便了教师的授课和学生的学习,所以我院通信工程专业 2000 级以前都以 MCS-51 作为授课对象。

我们从通信工程专业 2000 级(2002 年授课)开始,选择了 PIC16F877 作为该课程的授课工具。主要原因是 MCS-51 的 I/O 口驱动能力及抗干扰能力有限,可选型号不多,片内资源少,而美国 Microchip 公司的 PIC 8 位单片机,产量居世界第二位(仅

次于 Motorola 公司),品种超过 120 种,所具有的睡眠功能、看门狗、EEPROM、CCP、ADC、SPI、IIC、CAN 等,是当时 MCS-51 及类似产品没有的。

考虑到“微机原理与接口技术”课程和“单片机原理与应用”课程的相似性,我们将两门课合并为一门,统称“微机原理与接口技术”,主要讲授 8086 微处理器和 PIC16F877 单片机,授课课时 80 学时,实验 32 学时。针对 8086 微处理器特点,主要讲解微处理器的结构,工作原理,汇编语言程序设计, DOS 中断调用,存储器扩展方法,8255A 接口设计等。而其中的中断接口、定时器、串行接口、A/D、D/A 转换等不讲解,相关概念放在 PIC16F877 单片机原理中。在讲解 PIC16F877 单片机原理时,CPU 内部结

构、工作原理、指令系统等只作简单介绍,主要讲解中断系统、定时器、CCP、ADC、USART、SPI。由于许多概念的相似性,可以前后做比较性学习,使得合并后的课时比单独开设两门课少。经过2001、2002级教学实践,证明了上述教学改革是可行的。

由于PIC16单片机的指令系统简单,可用的程序范例、功能子程序不多及文件寄存器的体选,FLASH存储器的页选问题,使其汇编语言程序设计较MCS-51复杂,所以从2003级开始,尝试直接用PICC语言来教学。C语言设计的灵活性加上PIC16单片机丰富的硬件资源使得学生能轻松地设计以往用汇编语言很难设计的应用程序,本学期的期末实验考试证明绝大多数学生都能正确完成。

综上所述,单片机课程已经合并到“微机原理与接口技术”课程中,PIC16F877单片机使用PICC语言讲授。

2 直接用PICC语言进行PIC16F877单片机教学

通常,讲授单片机或微处理器都是用汇编语言,直接用C语言进行PIC16F877单片机教学将面临两大问题:一是一些汇编语言指令没有相类似的C语言表示方法,二是如何对片内的RAM进行绝对地址存储。

2.1 用嵌入式汇编解决无法用C语言表示的少数汇编指令

例如:

```
#include <pic.h>
Main()
{TRISC=0;PORTC=0X56H;CARRY=0;
#asm
    Rlf_PORTC,F;    //带进位位C左移,回送PORTC,与8086的
                    RCL指令功能相似,PORTC=0XAB
    Rrf_PORTC,F;    //带进位位C右移,回送PORTC,与8086的
                    RCR指令功能相似,PORTC=0X56
    Swapf_PORTC,F; //高低半字节交换指令,回送PORTC,
                    PORTC=0X65
    Sleep;          //睡眠指令
    Nop;             //空操作
    Clrwdt;          //清看门狗指令
#endasm
}
```

由于学生学习过C语言程序设计,并且刚学过

8086汇编语言程序设计,上述例子理解起来并不困难,还能通过本例讲解PICC编程时的注意事项,如SFR寄存器只能大写,嵌入汇编时应加下划线等。

2.2 文件寄存器的绝对定位解决片内的RAM绝对地址存储

在PICC编程时,有时需要将结果存储在文件寄存器的某个具体单元中,需要对文件寄存器进行绝对定位,方法如下:

```
static volatile unsigned char RAM40H @ 0x40;
//体0的40h单元
bank2 static volatile unsigned char RAM41H @ 0x41;
//体2的41h单元
```

文件寄存器的绝对定位方法:

```
static volatile bit 40H7 @(unsigned)&.40H*8+7;
//40H7表示40H单元的最高位bit7
```

3 直接用PICC语言进行PIC16F877单片机教学的优点

3.1 利用PICC编译器所带的头文件简化程序编写

2003级以前的单片机教学中,由于课时有限,基本上不讲EEPROM和FLASH的汇编编程,用PICC教学以后,基于PICC的EEPROM和FLASH的编程非常方便,利用PICC编译器所带的头文件读写函数读写EEPROM和FLASH即可,如:

```
#include <pic.h>
main(void)
{unsigned char data,data2;
 unsigned char address,address2;
 address=0x10;data2=0xaa
 data=EEPROM_READ(address);
 //将EEPROM的0x10单元的值送RAM的data单元
 address2=0x30;
 EEPROM_WRITE(address2,data2);
 //将0xaa写入EEPROM的0x30单元
}
```

3.2 利用PICC库函数简化程序编写

由于篇幅所限,在此不举例介绍,可参考参考文献^[1]。

总之,应用PICC进行教学后,编程的难度大大减少,为加快教学进度,加深教学难度打下基础,收到了很好的效果。

4 应用 PROTEUS 单片机仿真软件进行课堂教学

4.1 概述

虽然 PIC 单片机的 MPLAB7. 2 软件(可到 [www. microchip. com. cn](http://www.microchip.com.cn) 下载,是免费版的)的仿真功能很强,但对有些模块的仿真显得力不从心,特别是需要外部电路配合的模块,如 PSP 从动, LCD, LED 动态扫描, 矩阵键盘, USART, SPI, IIC 等, 学生理解起来难度较大,应用 PROTEUS 单片机仿真软件就能解决这个问题。(可到 www. labcenter. co. uk 下载 demo 版)。

Proteus 与其它单片机仿真软件不同的是,它不仅能仿真单片机的工作情况,也能仿真单片机外围电路或没有单片机参与的其它电路的工作情况。因此在仿真和程序调试时,关心的不再是某些语句执行时单片机寄存器和存储器内容的改变,而是从工程的角度直接看程序运行和电路工作的过程和结果。对于这样的仿真实验,从某种意义上讲,弥补了实验和工程应用间脱节的矛盾。

4.2 Proteus 软件的特点

(1)具有模拟电路仿真、数字电路仿真、单片机及其外围电路组成的系统的仿真、RS232 动态仿真、I2C 调试器、SPI 调试器、键盘和 LCD 系统仿真的功能;有各种虚拟仪器,如示波器、逻辑分析仪、信号发生器等。

(2)目前支持的单片机类型有:68000 系列、8051 系列、AVR 系列、PIC12 系列、PIC16 系列、PIC18 系列、Z80 系列、HC11 系列、ARM7 以及各种外围芯片。

(3)支持大量的存储器和外围芯片。

(4)是一款集单片机和 SPICE 分析于一身的仿真软件,功能极其强大,是其它任何一款软件不能相比的。

4.3 以 SPI 模块的教学为例说明仿真实验的方法

(1)在 PROTEUS 的 ISIS 环境画出如图 1 所示电路。proteus 的绘图过程如下:

运行 proteus 的 ISIS 程序后,进入该软件的主界面,在工作前,要设置 view 菜单下的捕捉对齐和 system 下的颜色、图形界面大小等项目,通过工具栏中的 p(从库中选择元件命令)命令,在 pick de-

vices 窗口中选择电路所需的元件,放置元件并调整其相对位置,调整元件参数设置,进行元器件间连线。

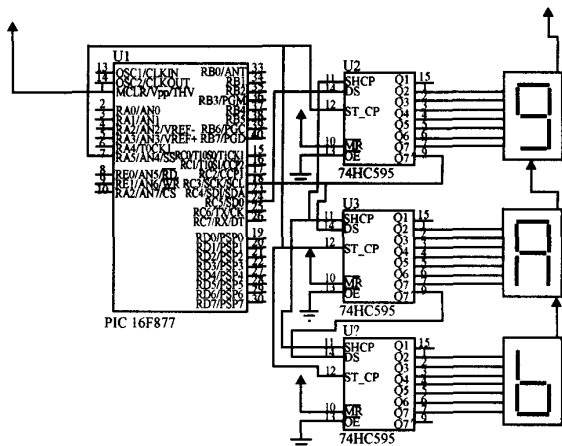


图 1 运行 SPI 模块的电路

(2)在 MPLAB7. 2 中设计并编译以下 SPI 的 PICC 程序(由于篇幅所限,略去 spi 初始化子程序、发送子程序),该程序用于在 3 个 LED 上依次显示 1~F 等 3 个字符。

```
// * 主程序 * /
main()
{
    unsigned int i;
    initial(); /* 系统初始化 */
    SPIINIT(); /* * SPI 初始化 * /
    for(i=0xB;i>8;i--) /* 连续发送 3 个数据 */
    {
        data=table[i];
        /* 通过数组的转换获得待显示的段码 */
        SPILED(data); /* 发送显示段码 */
    }
    PORTA_5=1;
    /* 最后给锁存信号,代表显示任务完成 */
}
```

(3)回到 ISIS 中,在 source 菜单的 Define code generation tools 菜单命令下,选择程序编译的工具、路径、扩展名等项目;在 source 菜单的 Add/remove source files 命令下,加入单片机硬件电路的对应程序;通过 debug 菜单的相应命令仿真程序和电路的运行情况。

(4)改变 main()中的

“for(i=0xB;i>8;i--)”//连续发送 3 个数据 * /”为“for(i=0x5;i>2;i--)”//连续发送 3 个数据 * /”,仿真结果如图 2 所示。

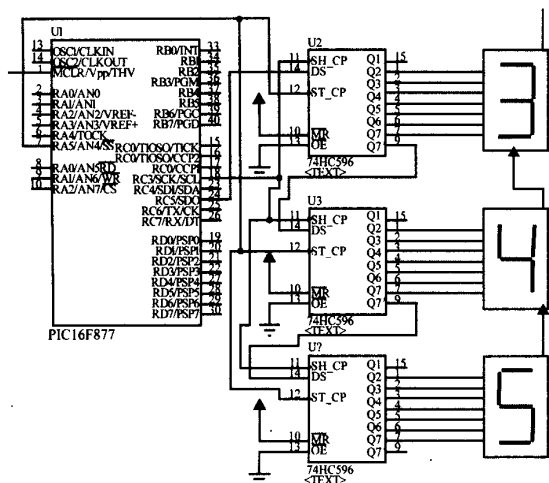


图2 仿真结果

可见,能随时修改程序,即刻显示仿真结果,帮助学生理解 SPI 模块工作原理,实践证明效果很好。

应用 PROTEUS 单片机仿真软件还可以进行学生实验,采用仿真软件后,不但能够减少实验室设备的投入,而且对实际工程问题的研究,可以先在软件环境中模拟,通过后再进行硬件的投入,这样不仅省时省力,还可以节省因方案不正确所造成的硬件投入的浪费。

学生可以利用在 Protues 做一个具体的工程项目,但需将最后的结果移植到一个具体的硬件电路中。

5 几点说明

5.1 PICC 编译器

文中所提的 PICC 是 Hitech 公司的 PICC 编译器,它稳定可靠,编译生成的代码效率高,在用 PIC 单片机进行系统设计和开发的工程师群体中得到广泛认可,其正式完全版软件需要购置,但在其网站上可下载到有限时的试用版软件。另外,Hitech 公司针对广大 PIC 的业余爱好者和初学者还提供了完全免费的学习版 PICC—Lite 编译器套件,它的使用方式和完全版相同,只是支持的 PIC16F87X 系列单片机型号限制在 PIC16F84。

软件下载网址:www.htsoft.com

5.2 教材问题

当前大部分的 PIC 单片机教材都是基于汇编语言的,直接用 PICC 的几乎没有,所以缺少合适的教材,我们暂且选用这类教材。另外根据需要编写了一本补充教材,给出了我们选择的 PIC 单片机教材中所有用汇编语言设计的例程的 PICC 程序。

5.3 C 语言编写的程序代码效率低的问题

一般情况下,程序用 C 语言编写,C 语言与汇编语言最大的区别就在于汇编程序执行效率较高些,因为 C 语言首先要用 C 编译器生成汇编代码,在不少情况下,C 编译器生成的汇编代码不如用汇编语言编写的代码效率高。

在 PICC 中,可以用两种办法在 C 程序中调用汇编程序,一是使用 `#asm`, `#endasm` 及 `asm()` 在 C 语言中直接嵌入汇编代码, `#asm` 和 `#endasm` 指令分别用于标示嵌入汇编程序块的开头和结尾; `asm()` 用于将单条汇编指令嵌入到编译器生成的代码中。

另一种方法是将汇编作为一个独立的模块,用汇编编译器(ASPIC)生成目标文件,然后用连接器和 C 语言生成的其他模块的目标文件连接在一起,如果变量要公用时,则在另一个模块中说明为外部类型,并允许使用形式参数和返回值。

我们通过将同一个例子的汇编语言程序和 PICC 语言程序的代码利用率进行比较,发现两者相差不是太大。如 CCP 模块的输出比较范例程序,用汇编语言编写时 ROM 使用 96 字,用 PICC 编写时使用 112 字。尽管 PICC 编译器产生的代码比较繁琐,但它的结构和逻辑性很强,开发效率高,调试与维护都很方便,不论是从程序的开发速度、软件质量还是从程序的可维护性和可移植性上讲,PICC 的优点绝非汇编语言所能比拟的。

参考文献:

- [1] 刘和平. PIC16F87X 单片机实用软件与接口技术—C 语言及其应用[M]. 北京:北京航空航天大学出版社,2004
- [2] 武锋,陈新建. PIC 单片机 C 语言开发入门[M]. 北京:北京航空航天大学出版社,2005
- [3] http://www.labcenter.co.uk/index_uk.htm