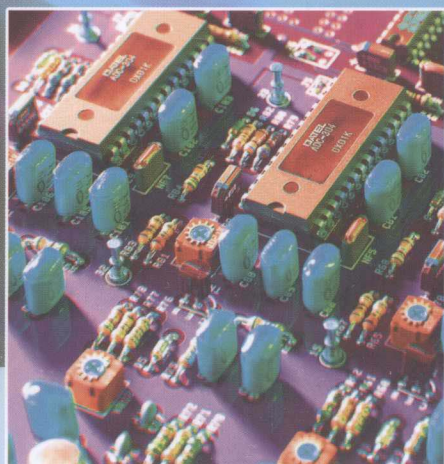


● 高等学校教材

单片机原理实验指导书

盛文利 编



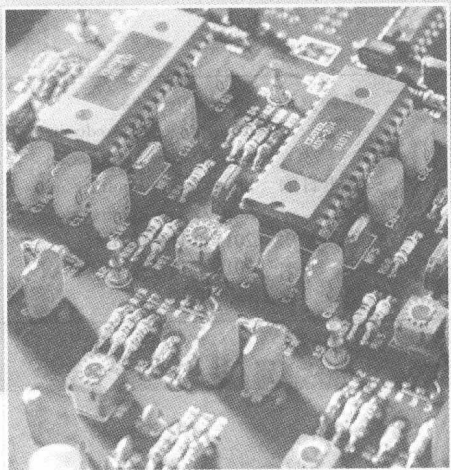
高等教育出版社

● 高等学校教材

单片机原理实验指导书

Danpianji Yuanli Shiyān Zhidāoshu

盛文利 编



高等教育出版社·北京

内容提要

本书是“单片机原理及应用”课程的实践环节指导书,包括软件基础、硬件连接和综合实验各个部分。全书侧重于实验的原理、内容以及报告的书写,尽量避免与教科书的内容重复。

全书共分十章和一个附录,第一章是实验环境的软硬件介绍,第二章是软件基础实验。第三章至第九章是单片机以及外围扩展等的实验内容。第十章是综合实验。本书重视课程的基础内容,同时结合多年的实验教学经验写成。书中各章实验内容以独立小节列出,以方便学生预习和书写实验报告。

图书在版编目(CIP)数据

单片机原理实验指导书/盛文利编.--北京:高等教育出版社,2015.11

ISBN 978-7-04-043973-1

I. ①单… II. ①盛… III. ①单片微型计算机-高等学校-教学参考资料 IV. ①TP368.1

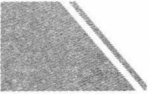
中国版本图书馆 CIP 数据核字(2015)第 237655 号

策划编辑 金春英	责任编辑 许海平	封面设计 于文燕	版式设计 马云
插图绘制 杜晓丹	责任校对 吕红颖	责任印制 毛斯璐	

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
印 刷 三河市骏杰印刷有限公司
开 本 787mm×1092mm 1/16
印 张 7.5
字 数 170 千字
购书热线 010-58581118

咨询电话 400-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
版 次 2015 年 11 月第 1 版
印 次 2015 年 11 月第 1 次印刷
定 价 12.60 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换
版权所有 侵权必究
物 料 号 43973-00



前言

“单片机原理及应用”课程是电气信息类专业的专业基础课,从几十年前的单板机时代,到现代的单片机技术,微处理器随着半导体技术的发展得到了迅速的发展。各种性能强大的单片机更是层出不穷。尽管如此,MCS-51 单片机仍然以入门简单,价格低廉等特点得到广泛应用。现在被重新封装,功能增强的 MCS-51 单片机也不断涌现。因此无论是课堂教学,还是爱好者的入门自学,仍然广泛地采用这一单片机。在这种情况下,各种实验系统和实验软件不断更新,实验教材也在随着硬件的变化不断更新。作为课程的重要环节,实验在单片机课程中尤其不可或缺,这次在新一轮设备更新和课程规范化的要求下,课程组委托我编写了这本实验指导书。

书中实验操作都是以南京伟福公司出品的实验系统为对象,内容基本涵盖了单片机应用的主要内容。现在市面上的实验系统和实验板层出不穷,就基本内容和原理而言都是相通的,本书中的操作虽然是针对南京伟福公司的实验系统,但基本理论和实验方法也适合其他 MCS-51 单片机应用实验系统。

作 者

2015 年 6 月

目录

第一章 实验环境组成 1

1.1 硬件基础 1

1.1.1 仿真系统 1

1.1.2 实验箱硬件 1

1.2 汇编语言编程简介 3

1.2.1 汇编语言的结构 3

1.2.2 汇编语言书写格式 3

1.2.3 伪指令和汇编控制 3

1.3 汇编语言伪指令 4

1.3.1 常量和符号定义 4

1.3.2 源程序地址设置和程序结束标识 4

1.3.3 程序段的使用 5

1.3.4 链接指令 5

1.4 简单汇编语言编写 6

1.5 C51 入门 8

第二章 数据传送和运算实验 10

2.1 数据传送 10

2.2 实验一 数据传送实验 10

2.3 实验二 算术运算实验 13

第三章 控制转移和子程序实验 16

3.1 实验一 I/O 口输出实验 16

3.2 实验二 变换闪烁模式的循环彩灯 18

3.3 实验二的 C 语言实现 20

第四章 定时器与中断实验 21

4.1 与定时器和中断相关的特

殊功能寄存器 21

4.1.1 定时/计数单元计数寄存器

THx 和 TLx(x=0 或 1) 21

4.1.2 定时/计数单元方式寄存器

TMOD 22

4.1.3 定时/计数单元控制寄存器

TCON 22

4.1.4 中断允许寄存器 IE 23

4.1.5 中断优先级寄存器 IP 23

4.2 实验一 方波信号发生器 24

4.3 实验二 中断和定时器实验 27

第五章 单片机串行口应用 34

5.1 串行口有关的特殊功能

寄存器 34

5.1.1 数据缓冲器 SBUF 34

5.1.2 串行口控制寄存器 SCON 34

5.1.3 特殊功能寄存器PCON 35

5.2 串行口工作方式 35

5.2.1 方式 0 35

5.2.2 方式 1 35

5.2.3 方式 2 36

5.2.4 方式 3 36

5.3 波特率的设计 37

5.4 实验 双机通信实验 38

第六章 系统扩展实验 44

6.1 接口扩展实验 44

6.1.1 8255A 芯片的结构 44

6.1.2 8255A 芯片工作方式选择 45

6.2 实验一 8255A 基本输入

输出实验 46

6.3	模数与数模转换	49
6.4	实验二 数模转换实验	49
6.5	实验三 模数转换实验	52
第七章 键盘和显示实验		55
7.1	数码管显示原理	55
7.2	实验一 数码管显示实验	58
7.3	实验二 键盘和数码管显示实验	61
7.4	汉字 LED 点阵显示原理	67
7.5	实验三 汉字滚动显示	68
7.6	液晶显示器原理和控制	72
7.7	实验四 LCD 显示实验	74
第八章 模拟串行端口应用实验 ...		80
8.1	I ² C 总线器件 24C02	80
8.2	实验一 模拟 I ² C 总线实验 ...	83

8.3	实验二 模拟 SPI 总线实验 ...	89
第九章 电机驱动		95
9.1	直流电机驱动原理	95
9.2	实验一 直流电机驱动	95
9.3	步进电机驱动	97
9.4	实验二 步进电机驱动	98
第十章 综合实验		101
附录 集成开发软件使用简介		108
附录 1	建立工程	108
附录 2	编辑程序	110
附录 3	编译项目	110
附录 4	仿真设置	111
附录 5	仿真运行	112

第一章 实验环境组成

单片机实验是单片机与接口应用课程的实践环节,其宗旨在于通过在实验室的实验环境下验证所学内容并加以综合运用,以加深对课程的理解。

实验室采用南京伟福公司出品的 LAB8000 微控制器仿真实验系统。该仿真实验系统的硬件由 PC 机和实验箱构成,实验箱内包括仿真器、实验板和电源等。实验板上面有外围器件电路和模块,仿真器的作用是,一方面负责模仿单片机的行为,另一方面与 PC 机保持通信,把运行状态及时反映给 PC 机,使用户可以直观地看到单片机内部的运行状态。

在伟福实验系统的 PC 机上有自己的集成开发软件,也叫做集成开发环境(IDE),同时也支持 Keil 公司的集成开发软件。考虑到目前 Keil 公司的集成开发软件更加通用,实验室中我们以 Keil 公司的集成开发软件为基础。关于 Keil 公司的集成开发软件的基本使用方法见附录。

1.1 硬件基础

1.1.1 仿真系统

单片机的开发过程中,硬件系统完成后,需要对软件进行调试。调试过程需要观察单片机内部运行状态信息,初学者在学习过程中也需要了解单片机内部的状态信息,可通过仿真器实现这一目的。仿真器仿真单片机的运行过程,可以模仿单片机一条一条的执行的指令,也可以随意中断程序的执行过程。同时仿真器与 PC 机的集成开发软件相互通信,可控制仿真单片机的指令单步或分段运行、观察并修改单片机的内部和外部存储器。另一面,仿真器仿真单片机的各个端口连接在实验板上。通过这些端口和实验板上面的电路和模块等外部器件,完成实验,达到学习和调试目的。

1.1.2 实验箱硬件

实验箱中提供以下实验电路和模块:

- (1) 逻辑电平输入开关。
- (2) 逻辑电平显示电路。
- (3) 单脉冲电路。
- (4) 扬声器驱动电路。
- (5) 继电器控制电路。
- (6) 逻辑笔电路。
- (7) 1MHz 和 10MHz 时钟脉冲信号。
- (8) PWM 转换电压电路。
- (9) 模拟量电压(电位器)电路。
- (10) 串口通信实验电路。
- (11) 6 位 8 段码 LED 数字显示器。
- (12) 4×6 键盘。
- (13) 存储器。
- (14) 8255 端口扩展电路。
- (15) 模数变换电路,可接入两路模拟量。
- (16) 数模变换电路,提供 0~-5V,-5~+5V,-8~+8V 三路输出。
- (17) 液晶屏显示电路。
- (18) 直流电机实验模块。
- (19) 步进电机实验控制模块。
- (20) 温度传感器实验模块。
- (21) 压力传感器实验模块。
- (22) 红外通信实验模块。
- (23) 16×16 点阵显示实验模块。
- (24) I²C 总线实验模块。
- (25) SPI 总线实验模块。
- (26) 1-Wire 实验模块。
- (27) 地址译码输出模块。

这些模块在实验箱上面的布局如图 1-1 所示。

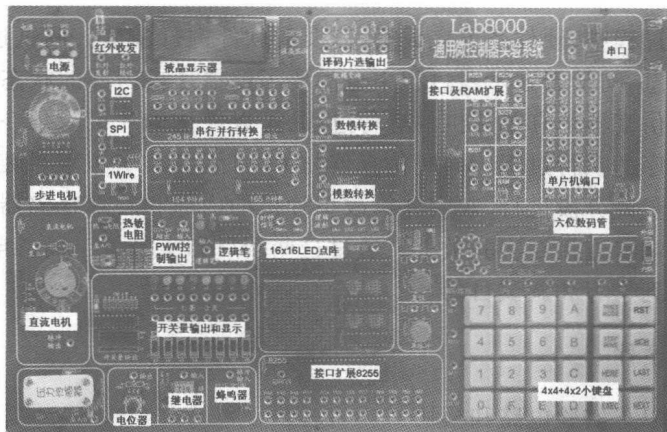


图 1-1 实验箱模块布局

每一个模块的具体电路原理和使用方法,在后面各章节的实验说明中给出。实验系统的仿真器没有独立外壳,以仿真板的形式集成到实验仪的仿真电路板背面,仿真电路板的仿真端口已经接到实验箱的电路板上,等同于 MCS-51 单片机的四个基本输入/输出端口 P0~P3,实验时只要接到相应的接线柱上即可。

1.2 汇编语言编程简介

单片机的状态改变由机器码控制,汇编语言中除了与机器码对应的指令外,还有进行编译控制的伪指令,汇编语言即是这些指令的组合。在这一节中,简单介绍 MCS-51 单片机汇编语言的用法。

1.2.1 汇编语言的结构

汇编语言是由我们书写的,用标号和助记符等编写成的文本程序,并不能由单片机直接运行。它要通过汇编器把它变成二进制格式的机器码,才能写入单片机程序存储器中运行。如果程序复杂,也可将程序分作所谓段。

段是一块程序或数据存储单元,定义于汇编语言源文件中,它的地址可以是绝对的,也可以是可重新定位的。可重定位段有名称、类型等属性。绝对地址段因为有确定地址,所以不需要名称,也不可以重新定位。由一个或多个段可以组成模块,可以给模块定义一个名称。一个程序文件可以有一个或多个这样的模块。

链接程序以段为基础,将不同模块中可重定位段和不可重定位段组成一个完整的程序,其中的可重定位段被赋予绝对地址,与不可重定位的绝对段合并,一起构成只含有可执行机器码的目标代码。

1.2.2 汇编语言书写格式

汇编语言的书写格式如下:

[标号:] 操作码 [目的操作数/寻址方式] [,源操作数/寻址方式] [;注释]

方括号 [] 表示可选项,一个具体的指令可以有这部分也可以没有这部分。标号为用户自定义部分,代表指令所在的程序存储器地址,可以是 1~8 个以字母开头的字符串,以“:”结尾。操作码就是指令助记符。目的操作数存储指令的操作结果,源操作数是操作的数据源。注释以“;”开头,不进行任何操作,起到帮助解释程序功能的作用。

1.2.3 伪指令和汇编控制

除机器码的助记符指令外,汇编语言还需要定义程序结构、数据、常数等,这些由伪指令完成。另外还有一些控制语句,控制汇编器的编译模式和流程。

1.3 汇编语言伪指令

这里简单介绍几个常用的汇编语言伪指令及其写法,涉及基本的段和控制语句。然后使用它们给出简单程序的写法。

1.3.1 常量和符号定义

可使用 DB 和 DW 定义表格等常量,DB 为字节,DW 为双字。同时通过标号赋予表格地址一个符号,如:

```
TAB:DB 33H,0A4H,12H,78H ;以字节初始化存储器
TAB1:DW 33A4H,1278H ;以字初始化存储器,但存储内容相同
```

通过引用符号(或标号)TAB 或 TAB1,引用表格的内容。

DS 伪指令用于直接说明在存储器中保留的字节数量,例如:

```
SNTC:DS 4
```

那么在 SNTC 开始的存储器处,保留 4 个字节。

可使用 EQU 定义常数或寄存器名:

```
PAR EQU 100 ;PAR 等同于常数 100
```

符号的首字符必须是字母或下划线,后面可以是字母、数字、问号“?”或下划线。

1.3.2 源程序地址设置和程序结束标识

ORG 伪指令用于设置指令的地址,如:

```
ORG 0 ;程序从 0 开始,即上电复位后的程序从此处开始
ORG 0BH ;定时器 T0 的中断向量地址
```

END 伪指令用于程序结束标识,汇编程序在遇到 END 指令时,就认为程序结束于此,停止继续汇编。

使用以上伪指令,就可以设计简单的汇编语言程序,例如:

```
CONST EQU 200
ORG 0
MOV R2, #CONST
LOOP:DJNZ R2,$ ;$代表本条指令地址
END
```

USING 伪指令用于标明当前代码使用的寄存器组,书写格式:USING exp ;exp 是 0, 1, 2 或 3,代表 4 个 51 寄存器组。

1.3.3 程序段的使用

程序分段是由链接程序决定的,使用定义语句 SEGMENT 来定义一个段。该语句由段名字、段定义语句、段存储类型和空间定位属性构成。

```
myseg SEGMENT class reloctype alloctype
```

myseg 是自定义段名称(标识符),class 说明段的存储类型,可选的 reloctype 和 alloctype 字段是段存储空间的定位属性,用于进一步通知汇编器该段在存储器中的位置。

表 1-1 为 class 字段的说明。

表 1-1 class 字段的说明

标识符	标识符说明的地址
BIT	可位寻址空间,内部 RAM 的 20H~2FH 地址字节
CODE	程序存储器,在程序存储器的 0000H~0FFFH 地址范围内
IDATA	内部 RAM,内部数据空间的 00H~7FH 和特殊功能寄存器内
XDATA	外部数据存储空间,地址在 0000H~0FFFH 地址范围内

可选字段用法可参考下面例子,进一步用法请查看软件附带说明。

myseg SEGMENT CODE AT 0023H ;定义一个名字为 myseg 的段,放到程序存储器空间,地址为 0023H

xdseg SEGMENT XDATA PAGE ;定义一个名字为 xdseg 的数据段,PAGE 说明数据要起始于 256 字节——页

可重定位段定义后,用 RSEG 语句选择并激活该段,例如:

```
myseg SEGMENT CODE ;声明 myseg 段
      RSEG myseg ;选择该段
      MOV A,#0
      MOV P0,A
```

1.3.4 链接指令

这些指令用来标明模块中需要使用的符号,或者模块中的哪些符号是可以被其他模块使用的符号,这些指令在各个不同模块之间建立关联。

PUBLIC 被这个指令声明的模块作为公共模块,可被任何其他模块使用,如:

```
      PUBLIC delay
myseg SEGMENT CODE
      REG myseg
Delay: .....
      .....
      RET
      END
```

此时,Delay 程序可被其他不同模块调用。

EXTRN 当前模块若使用其他模块声明的符号,必须使用这个指令说明,例如有上例 PUBLIC 声明后,若在另外模块中使用 delay,必须使用 EXTRN 声明,才可使用。

```
EXTRN CODE (delay)
```

这样声明后,程序即可引用 delay。

NAME 用于命名目标模块的名字,如:

```
NAME modulname
```

将当前目标模块命名为 modulname。

——这一节给出了常用的伪指令,更进一步说明请参考软件的在线帮助信息。

1.4 简单汇编语言编写

本节给出三个汇编语言最简单写法的例程,初学者可以把它作为框架或参考它的写法。

在单一文件里面,单一程序可以不使用模块化程序下的复杂写法,可以忽略段信息,直接用伪指令 ORG 定义其后面指令的地址。这样的程序简单易懂,但无法应付复杂系统的程序设计,所以只适合初学者。

例一:

```

ORG 0
LJMP MAIN
ORG 100H
MAIN:  MOV SP, #60H
      CLR A
      NOP
START:  MOV P1, A
      LCALL DELAY
      INC A
      SJMP START
DELAY:  MOV R6, #100
DLY1:  MOV R7, #100
      DJNZ R7, $
      DJNZ R6, DLY1
      RET
END
```

这是一个顺序驱动 P1 口为 1 的程序,程序从地址 0 处开始,这也是 MCS-51 单片机程序的入口。然后跳转到地址为 100H 的主程序,进行必要的初始化后,进入循环程序

START。每循环一次,都有调用延时程序 DELAY。

例二:

如果程序变得庞大复杂,这种写法不易维护也很难复用,所以,可以按例二这样编写。

主文件 file1.asm 中程序:

```

        EXTRN CODE (DELAY)
        ORG 0
        LJMP MAIN
        ORG 100H
MAIN:   MOV SP, #60H
        CLR A
        NOP
START:  MOV P1, A
        LCALL DELAY
        INC A
        SJMP START
        END

```

子程序文件 delay.asm 中包含的子程序:

```

        PUBLIC    DELAY
DLY     SEGMENT CODE
        RSEG      DLY
DELAY:   MOV R6, #100
DLY1:   MOV R7, #100
        DJNZ R7, $
        DJNZ R6, DLY1
        RET
        END

```

例三:

编写有参数传递的子程序,例如计算两个字节的乘法,并返回结果。这里给出可被任何主程序调用的子程序,忽略主程序。

```

        PUBLIC    MULT
MULT     SEGMENT CODE
        RSEG      MULT
MULT:   MOV A, R6
        MOV B, R7
        MUL AB
        MOV R6, B ;乘法结果的高位
        MOV R7, A ;乘法结果的低位
        RET
        END

```

主程序将 R6 和 R7 两个参数备好,乘数结果被放到 R6、R7,返回。

1.5 C51 入门

对于单片机入门学习来说,使用汇编语言可以加深理解单片机的内部原理和指令系统的由来,但对于工程开发应用来说,效率较低,维护工作量大。现在 C 语言做单片机开发已经流行,它的最大优势在于入门快、效率高。MCS-51 单片机的 C 语言编译器最流行的是 Keil 公司的 C51,它符合标准的 C 语言语法,并有扩展。这里不打算详细说明 Keil 公司的 C51 的用法,只通过两个例子做简单说明,在后面实验部分会进一步说明。

例一:

把上面汇编语言的例一用 C 程序编写。

```
#include <REG52.h>           //包含 51 的寄存器定义文件

typedef unsigned char  uchar;
typedef unsigned int   uint;

void delay(void);           //声明超前引用函数
main() {
    uchar i;
    i=0;
    for(;;) {
        P1=i;
        i++;
        delay();
    }
}

void delay() {               //延时函数
    uchar i,j;
    for(i=0;i<100;i++){
        for(j=0;j<100;j++)
            ;
    }
}
```

例二:

上面 delay() 函数作为外部函数,在另外一个文件里如 delay.c 文件,且 delay.c 已经加入项目中,那么声明引用时前面加 extern 说明。

```
extern void delay(void) ;    //说明外部引用
```

也可用汇编语言编写 delay 函数:

```

$NOMOD51
NAME DLY
? PR? delay? DLY      SEGMENT CODE
PUBLIC_dlylong

RSEG ? PR? delay? DLY
delay:

    LOOP:
    DJNZ R7, LOOP ;4μs 6 MHz
    RET ;1μs
END

```

存入 delay.a51, 就可以被项目中其他文件调用, 调用前同样要用 extern 说明。

这一章介绍了实验系统的硬件和软件基础知识, 后续实验中涉及的更进一步知识, 将在各章节中说明。

第二章 数据传送和运算实验

数据处理是计算机主要活动之一,51 单片机的存储空间分为程序存储器和数据存储器,数据存储器又分成内部数据存储器 and 外部数据存储器,它们分别由不同指令寻址。

2.1 数据传送

51 单片机的数据传送指令有三类,如表 2-1 所示。

表 2-1 基本数据传送指令

MOV	内部存储器间传送
MOVX	外部数据存储器与累加器间传送
MOVC	程序存储器向累加器传送

实验一通过练习数据的传送操作,练习实验环境的使用方法。实验包含两个内容,内容 A 通过在数据存储器建立和传送数据,了解存储器属性和操作方法。内容 B 通过查表法将 0~9 转换为 ASCII 字符,熟悉 MOVC 的查表方法。

内容 A 首先在内部数据存储器 30H~3FH 中建立数据区,内容为 00H~0FH。然后把数据传送到外部 RAM 的 2000H~200FH 中,然后再把数据传送到内部 RAM 的 50H~5FH 中。

2.2 实验一 数据传送实验

一、实验目的

- (1) 学习 MCS-51 单片机片内、外部数据传送类指令的使用方法。
- (2) 学习查表指令的使用方法。

(3) 学习简单汇编语言书写方法和 Keil 公司的集成开发软件环境的使用方法。

二、实验设备及器件

PC 机和 LAB8000 仿真实验箱各一台。

三、实验内容

(1) 在 Keil 公司的集成开发软件环境中建立项目、建立汇编语言程序文件、编写程序。将程序加入项目、编译。这时在建立项目时,要选择具体器件;保存文件时,要选择文件的类型。

(2) 启动仿真运行程序,观察各种寄存器中相应地址的内容。

四、实验要求

(1) 在 Keil 公司的集成开发软件仿真环境下编写、运行实验程序。

(2) 掌握 MCS-51 单片机内部、外部数据传送指令,以及 A、Rn、Direct、@ Ri、@ DPTR 为目的操作数的使用方法。

(3) 掌握仿真环境下如何建立一个新文件、编写程序,编译、仿真及调试程序,学会在集成开发环境下,查看内部 RAM、外部 RAM 和特殊功能寄存器的数据变化情况。

五、实验步骤

(1) 运行软件,新建项目(project)文件存盘。存盘时选择适当的驱动器,建立自己的文件夹,然后将文件存在文件夹下面。这时软件提示选择目标系统的具体器件,此时选择与书本内容接近的芯片(如 Atmel 下的 AT89C51),在接下来出现的对话框中选择“否”。

(2) 建立新的文本文件,然后保存新建文件,在文件名对话框上键入文件名,此时注意文件是用汇编语言编写的,所以类型名必须是.ASM 或.A51。

(3) 新文件存盘后,在项目的子文件夹“Source Group 1”上面,单击鼠标右键,在弹出菜单中选择“Add Files to Group ‘Source Group 1’”,将新建汇编文件加入项目中,然后在文件中编写实验程序。

(4) 程序编写完成后存盘、编译。如果编译出现错误,根据提示重新改写,然后存盘、再编译。直至编译通过,编译信息为“0 warning(s) 0 error(s)”为止。

(5) 设置仿真环境启动仿真(参见附录)。

(6) 在开发软件的菜单栏“View”菜单中,选择“Memory Window”。在弹出的 Memory 窗口上,地址(Address)栏中,输入“C:地址”、“D:地址”或“X:地址”,分别表示程序存储器、内部 RAM 或外部 RAM 数据地址。例如“D:0”表示内部 RAM,地址 0 开始,还要注意地址不可以超范围。

(7) 按 F10 或 F11 键,单步运行程序。在左边栏内观察特殊功能寄存器的状态变化,在存储器窗口(memory window),观察相关存储器的状态变化。记录程序中涉及的寄存器和存储器的变化。最后将实验程序重新以全速运行,观察程序状态。

六、预习要求

(1) 预习教材与存储器内容相关章节,掌握用于片内和片外数据存储器传送指令的使用方法。

(2) 写出预习报告。

七、参考程序

1. 实验内容 A 参考程序