

伟福单片机实验系统

使 用 指 南

## 目 录

|                       |    |
|-----------------------|----|
| 一、仿真实验系统的特点与基本组成..... | 2  |
| 二、伟福实验系统的硬件组成.....    | 7  |
| 三、板上仿真器使用方法.....      | 17 |
| 四、逻辑分析工具.....         | 27 |
| 五、系统自动检测功能.....       | 31 |
| 六、仿真软件系统的使用.....      | 34 |

## 一、仿真实验系统的特点与基本组成

伟福单片机仿真实验系统由板上仿真器、实验仪、伟福仿真软件、开关电源构成。本实验系统提供强大的逻辑分析、波形输出和程序跟踪功能，可以让学生直观地观察到单片机内部及外部电路工作的波形。实验系统包含丰富的实验电路并提供灵活的组合方法，系统电路模块既可以和 8031 系列、也可以和 80C196 系列 CPU 及 8088/8086CPU 组合完成各种实验。本实验仪将高档仿真器所具有的逻辑分析仪、波形发生器和程序跟踪器等强大的分析功能移植过来，让学生在做实验时不仅能知道软件的执行过程，也能直观地看到程序运行时，电路上的信号状态、工作时序，非常详细地了解电路的工作情况。

仿真实验系统具有三种使用方法：

- (1) 用 PC 机上的集成调试软件驱动板上仿真器进行仿真和实验。
- (2) 无 PC 机，仅用实验仪的板上仿真器进行仿真和实验。
- (3) 无实验仪、无仿真器，仅在 PC 机上采用软件模拟方式进行仿真。

配 MCS51/196 仿真板，可进行 8051 或 80C196 的实验，配 8088/8086 仿真板，可以进行 8088/8086 实验。实验系统自带键盘和显示器，自带系统监控程序。如果没有 PC 机也同样进行各种学习和实验。配备 PC 机集成调试软件，在有 PC 机的情况下，通过板上仿真器实现 64K 全空间的硬件断点和仿真。

伟福实验系统提供了完成以下实验的硬件电路：

- (1) 数模变换电路，提供 0--5V, -5V+5V, -8V+8V 三路输出
- (2) 模数变换电路，可接入两路模拟量。
- (3) 逻辑电平输入开关
- (4) 逻辑电平显示电路
- (5) 单脉冲电路
- (6) 逻辑笔电路

- (7) 4MHz 脉冲信号源
- (8) 多级分频电路，与 4MHz 脉冲源结合，可得多种脉冲信号。
- (9) PWM 转换电压电路
- (10) 模拟量电压（电位器）电路
- (11) 串口通信实验电路
- (12) 扬声器驱动电路
- (13) 继电器控制电路
- (14) 逻辑门电路
- (15) 六位 8 段码 LED 数字显示器
- (16) 4x6 键盘
- (17) 存储器
- (18) 8255 端口扩展电路
- (19) 液晶屏显示电路
- (20) 逻辑分析仪采样和可编程数字脉冲信号输出，提供直观的分析手段
- (21) 通用集成电路插座(DIP40/28/24/20/18/16/14)
- (22) 地址，数据及控制电路总线接出插座
- (23) 步进电机实验
- (24) 扫印机驱动实验
- (25) 温度传感器实验
- (26) 直流电机实验

PC 机软件具有全集成化仿真环境，中、英文两种界面，软件仿真与硬件仿真两种模式，软件仿真可以在无仿真仪的情况下进行。实验仪可以方便灵活地构成各种实验方案，在有无 PC 机和实验仪的情况下，都能进行相应的编程实验，从而具有极为广泛的应用范围，板上提供了基本的实验电路，减少

繁琐的连接线过程,板上也提供了 DIP40/28/24/20/16/14 插孔和 CPU 的地址数据总线引出插孔,供学生自己扩展其它实验,培养实际动手能力,加强对实验电路的理解。

伟福实验系统在硬件上预留了自主开发实验的空间。对基本实验仅需少量连线就可进行,同时也提供了一些需较多连线的扩展性实验,以进一步锻炼学生的动手能力。此外,它还为学生们提供了强大的软硬件调试手段。

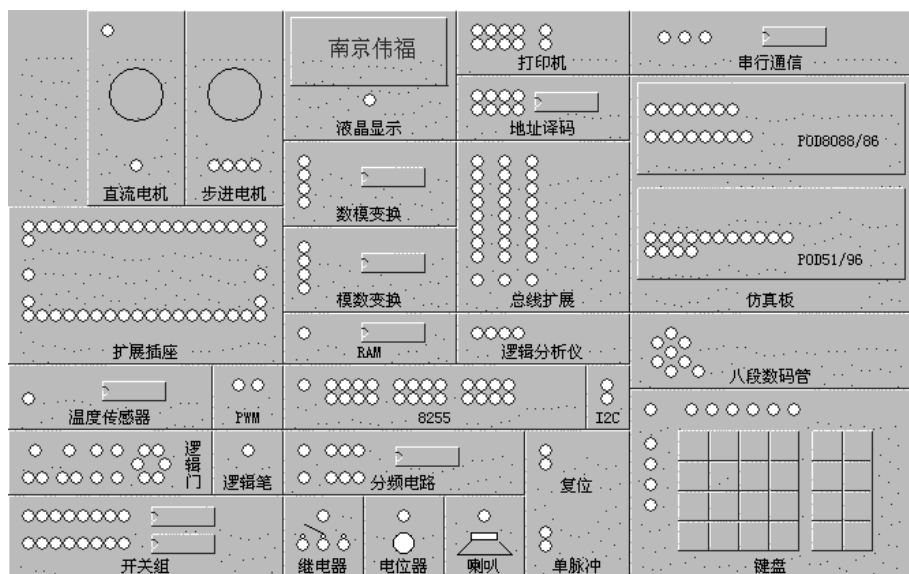


图 1.1 伟福仿真试验系统的功能模块排列

伟福仿真实验系统上各功能模块的排列如图 1.1 所示。其实物照片如图 1.2 所示。各个模块的引脚都被引出到相应的插孔,扩展插座为 40 芯通用插座,可兼容 28 芯、24 芯、16 芯和 14 芯插座。利用这个插座,可对双列直插式的各种微机芯片进行实验。总线扩展为三排插孔,用于引出各种总线信号,包括 8 根数据总线,16 根地址总线,以及 ALE, RD, WR 等控制总线,另外与 CPU 相关的一些控制信号和 I/O 信号例如 P1 口、RXD、TXD 等信号在仿真板上引出。

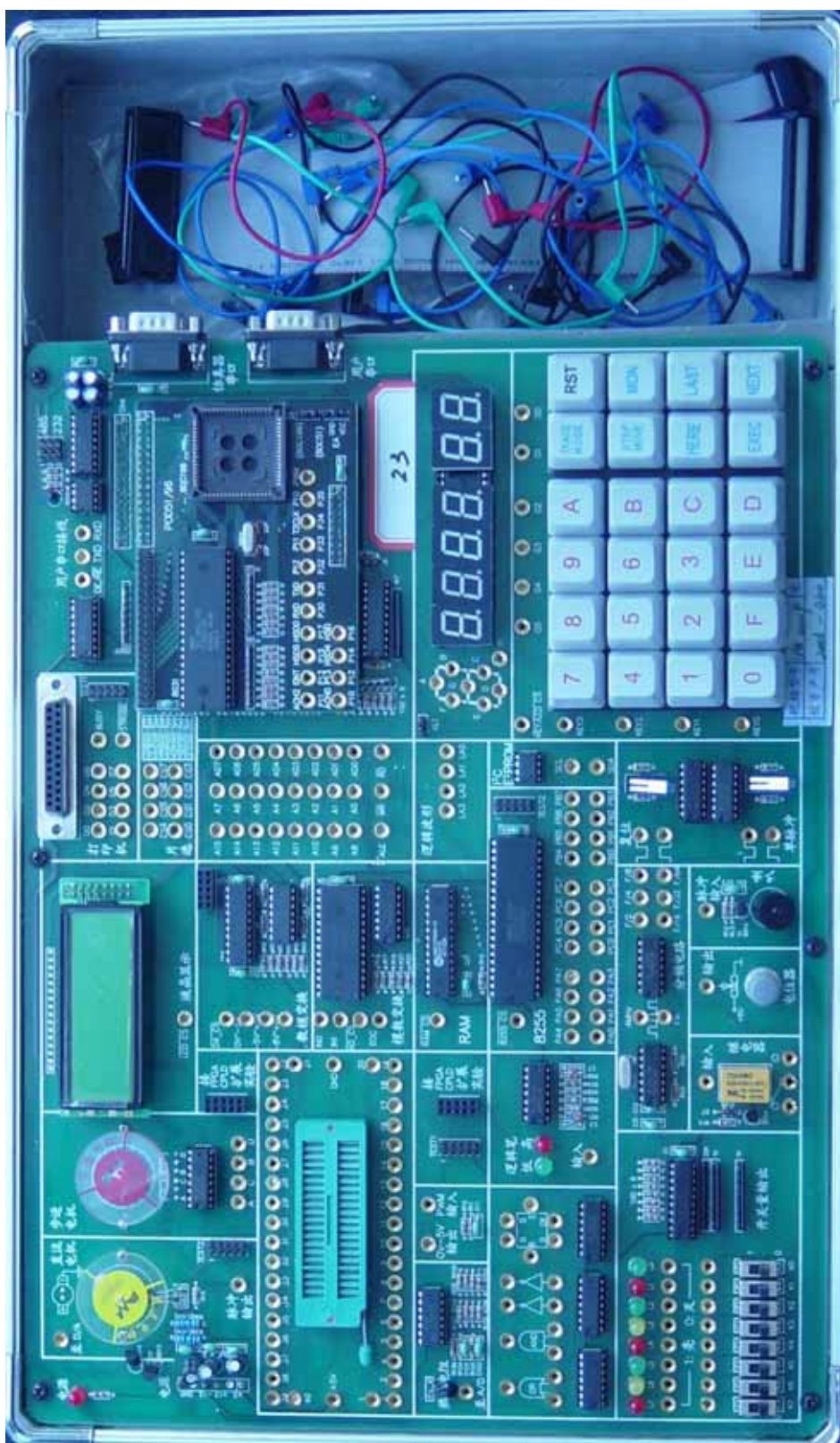


图 1.2 伟福仿真试验系统的实物图片

图中的圆点表示仿真实验系统上的连线插孔，每个插孔旁边都有相应的文字标识，进行不同实验的时候，需要根据要求用专门的连线将它们连接起来。

注意：无论是集成电路的插拔、通讯电缆的连接、跳线器的设置还是实验线路的连接，都应确保在断电情况下进行，否则可能造成对设备的损坏。实验线路连接完成后，应仔细检查无误后再接通电源。

## 二、伟福实验系统的硬件组成

本实验系统以模块化的结构设计为基础，配以丰富的实验电路，构成灵活多变的实验。这些电路既可以和 8031 系列、也可以和 80C196 系列 CPU 及 8088/8086CPU 组合完成各种实验。通过逻辑分析仪、波形发生器和程序跟踪器等强大的分析功能，让学生在实验时不仅能知道软件的执行过程，也能直观地看到程序运行时，电路上的信号状态，工作时序，非常详细地了解电路的工作情况。

### 2.1 LED 6 位数字显示器

### 2.2 4x6 键盘电路

本实验仪的 LED 显示电路和键盘电路如图 2.1。显示控制的位码由 74HC374 输出，经 MC1413 反向驱动后，做 LED 的位选信号。位选信号也可做为键盘列扫描码，键盘扫描的行数据从 74HC245 读回，374 输出的列扫描码经 245 读入后，用来判断是否有键被按下，以及按下的是什么键。如果没有键按下，由于上拉电阻的作用，经 245 读回的值是高，如果有键按下，374 输出的低电平经过按键被接到 245 的端口上，这样从 245 读回的数据就会有低位，根据 374 输出的列信号和 245 读回的行信号，就可以判断哪个键被按下。LED 显示的段码由另一个 74HC374 输出。

键盘和 LED 显示的地址译码见下图，做键盘和 LED 实验时，需将 KEY/LED CS 接到相应的地址译码上。位码输出的地址为 0X002H,段码输出的地址为 0X004H,键盘行码读回的地址为 0X001H,此处 X 是由 KEY/LED CS 决定，参见地址译码。例如将 KEY/LED CS 接到地址译码的 CS0 上，那么位码输出的地址就为 08002H,段码输出的地址就是 08004H,键盘行码读回的地址为 08001H。



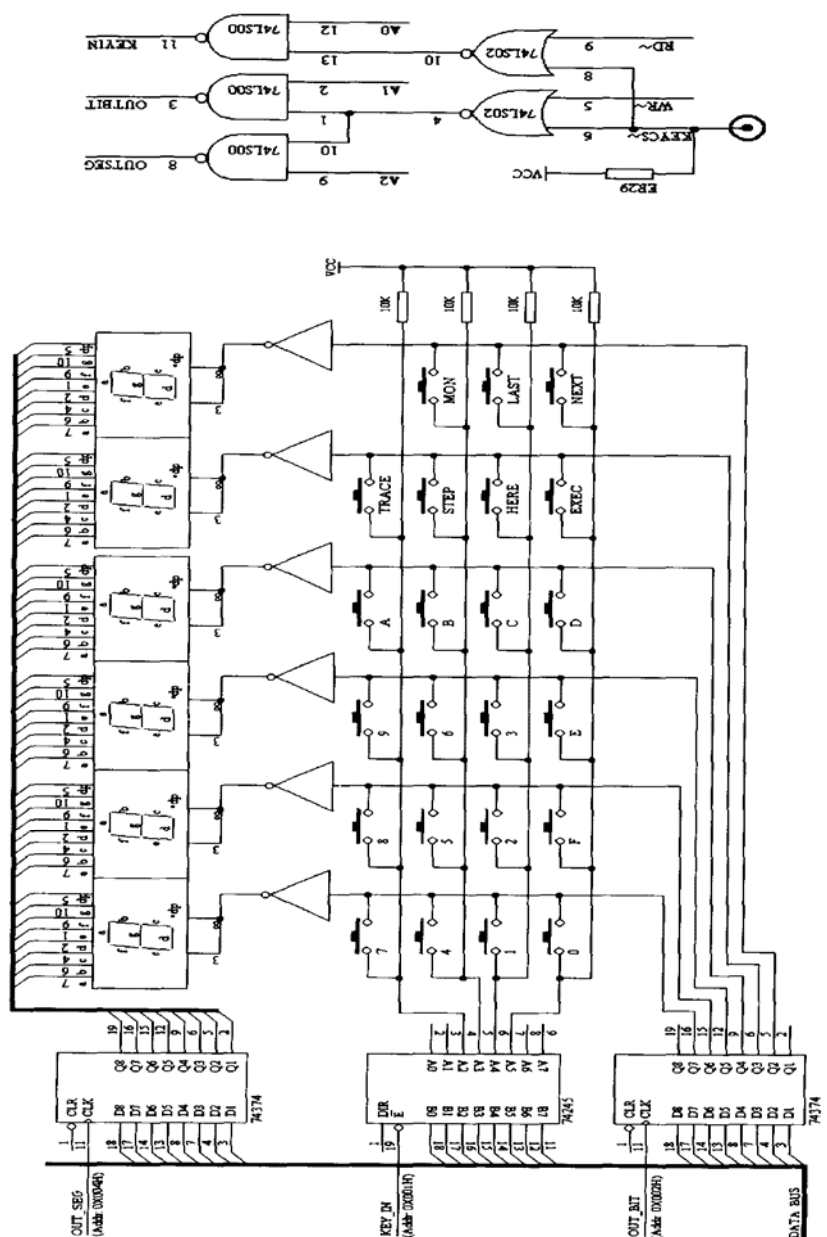


图 2.1 LED 显示电路和键盘电路如

## 2.3 LED 电平显示电路

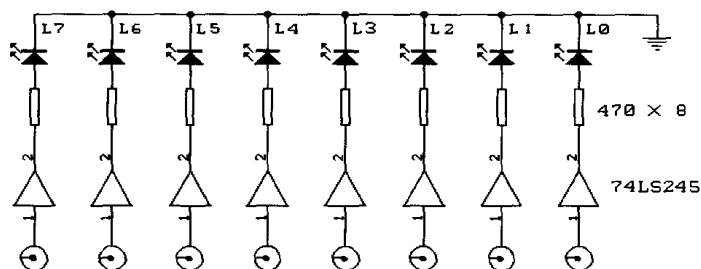


图 2.2 LED 电平显示电路

实验仪上装有 8 只发光二极管及相应驱动电路。见图 2.2, L0- L7 为相应发光二极管驱动信号输入端, 该输入端为高电压电平“1”时发光二极管点亮。我们可以通过 P1 口对其直接进行控制, 点亮或者熄灭发光二极管。

## 2.4 逻辑电平开关电路

实验仪上有 8 只开关 K0-K7, 并有与之相对应的 K0-K7 引线孔为逻辑电平输出端。开关向上拨相应插孔输出高电平“1”, 向下拨相应插孔输出低电平“0”。见图 2.3

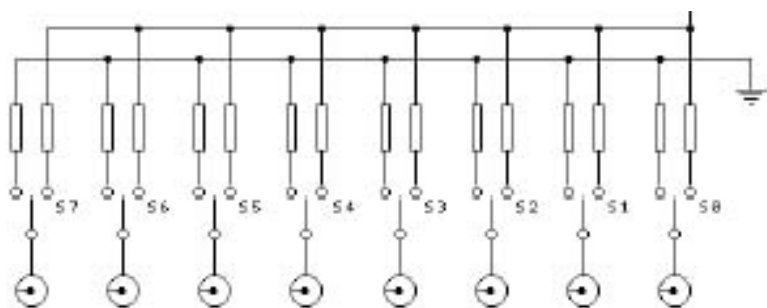
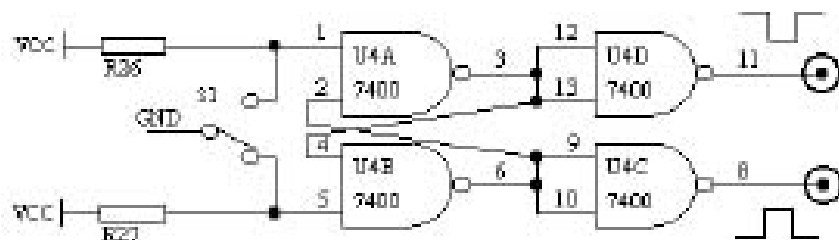


图 2.3 逻辑电平开关电路

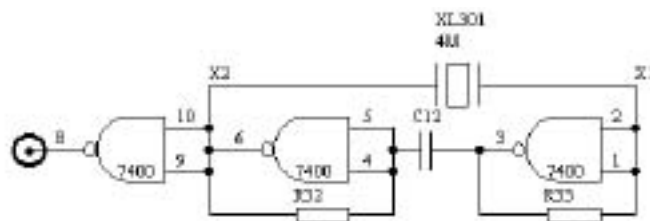
## 2.5 单脉冲电路

单脉冲电路由按键(PULSE)和去抖动电路组成, 每按一次(PULSE)键产生一个单脉冲。板上有单脉冲的输出信号插孔, 图为""和"", 分别为正脉冲和负脉冲。

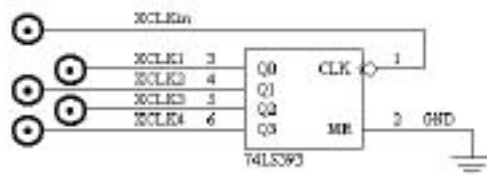


## 2.6 脉冲发生电路

下图是 4MHz 脉冲信号输出电路。如想得到其它频率的脉冲信号，可将 4MHz 脉冲信号接到分频电路 L, 经过分频后, 能得到 2MHz, 1MHz, 500KHz, 250KHz, 125KHz, 62.5KHz 多种频率的脉冲信号。脉冲产生电路和分频电路如下：



## 2.7 分频电路



## 2.8 继电器输出电路

当控制端电平置高，公共触点与常开端吸合。我们可以将常开端接入 1 发光二极管公共端接+5V 电平，通过对控制端进行控制，观察发光二极管的状态。见图 2.7。

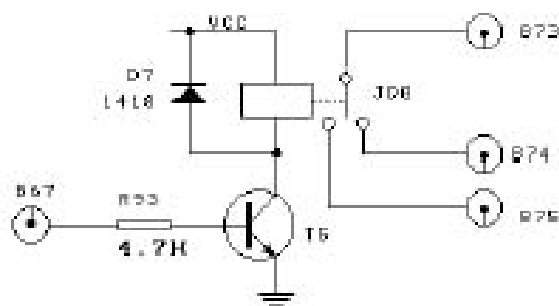


图 2.7 继电器控制电路

## 2.9 A/D 转换电路

实验仪上有一个 0-5V 的可调电位器，将可变电压输出端接入 A/D 转换电路的输入端通过 CPU 软件处理，读进 A/D 转换值，再将转换值送数码管显示。我们可以调节电位器，使之输出不同电压值，通过数码管的显示，检验 A/D 转换正确与否。

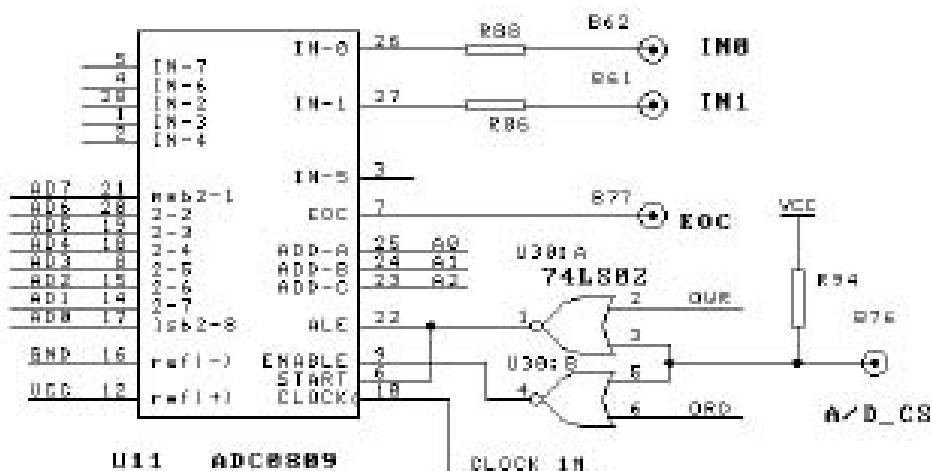


图 2.8 A/D 转换电路

## 2.10 D/A 转换电路

实验仪上提供了 D/A 转换电路如下图所示。我们可以通过软件编程控制 D/A 转换芯片 DAC0832, 输出相应电流值，经过采样电路取出模拟量电压值，用电压表测量电压输出端子，读出电压值。

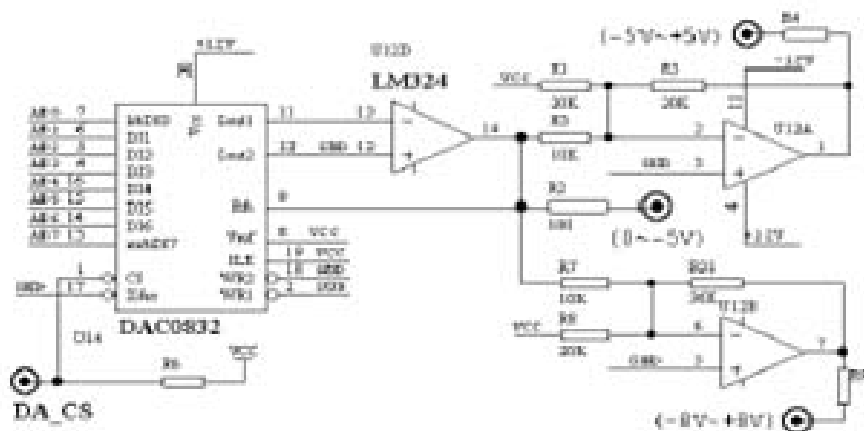


图 2.9 D/A 转换电路

## 2. 11 PWM 转换电路



**图 2.10 PWM 转换电路**

## 2.12 音频放大电路

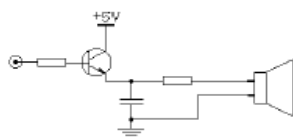
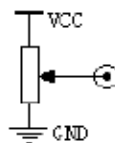


图 2.11 音频放大滤波电路



**图 2.12 电位器**

## 2.13 可调模拟量输入电路

电位器电路用于产生可变的模拟量。

## 2.14 逻辑测量（逻辑笔）电路

本实验仪上有逻辑测量电路，如图 2.13 所示，可用于测量各种电平，其中红灯亮表示高电平，绿灯亮表示低电平。如果两灯同时闪动，表示有脉

冲信号；两灯都不亮时，表示浮空（高阻态）。

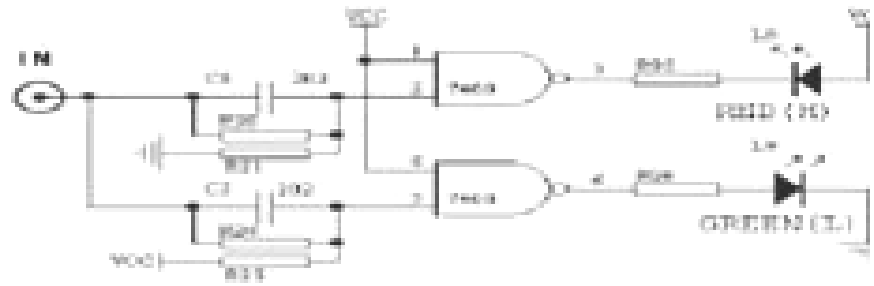


图 2.13 逻辑笔电路

2.15 存储器电路

本实验仪上有一片 32K 存储器 61256。提供给学生做存储器实验，由于地址译码为 4K 一段，所以只能提供 4K 容量使用，地址从 0000H-0FFFH。用 RAM CS 来选择不同的地址段，以适应不同的应用电路。

2.16 逻辑门电路

本实验仪提供系列门电路：非门，或门，与门，D 触发器。逻辑门电路由 7400 和 7404 组合实现。

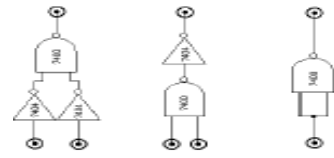


图 2.14 或门 与门 非门

2.17 液晶屏显示控制电路

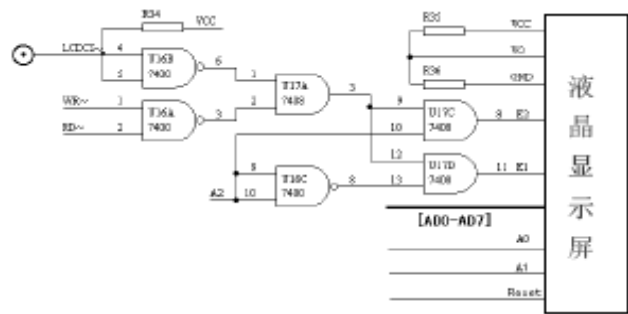


图 2.15 液晶屏显示控制电路

## 2.18 实验电路插座

本实验仪具有 1 个 40 芯通用电路插座,每个插座的全部引脚都被引出到相应的插孔,40 芯通用插座可兼容 28 芯、24 芯、16 芯和 14 芯插座。利用这个插座,可对双列直插式的各种微机芯片进行实验。

## 2.19 总线插孔

本实验仪上有三排总线插座,用于引出各种总线信号,其中 AD0-AD7 为 8 根数据总线,A0-A15 为 16 根地址总线。另外 ALE, RD, WR 为控制总线。与 CPU 相关的一些控制信号和 I/O 信号例如 PI 口、RXD, TXD 等信号在相应的仿真板上。

## 2.20 地址译码插孔

在进行不同实验的时候可能需要采用不同译码地址,地址插孔的地址分配如表 2-1 所示。

表 2-1 地址译码插孔的地址分配表

|     |                 |
|-----|-----------------|
| CS0 | 08000H~08FFFFH  |
| CS1 | 09000H~09FFFFH  |
| CS2 | 0A000H~0AFFFFH  |
| CS3 | 0B000H~0BFFFFH  |
| CS4 | 0C000H~0CFFFFH  |
| CS5 | 0D000H~0DFFFFH  |
| CS6 | 0E000H~0EFFFFH  |
| CS7 | 0F000H~0FFFFFFH |

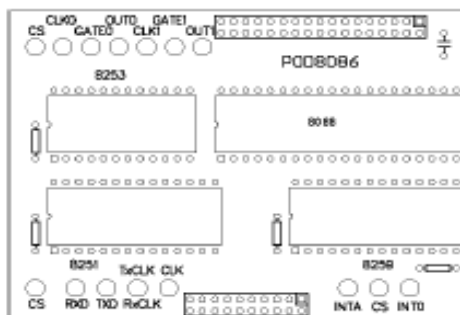
## 2.21 串口通信程序实验插孔

用户在做串行通信时,如果不需要将 TTL 电平转到 RS232 电平,可直接将 POD51/96 仿真板或 POD8086 仿真板上的 TXD, RXD、地一与通信对方交叉对接即可。

## 2.22 仿真板选择

在 POD51/96 仿真板上有两个跳线器，其中跳线器 SI 用于 CPU 种类的选择  
选择 8051 对 8051 单片机进行仿真实验。

另一个跳线器接 80C51 的 EA 脚,用于选择 80C51 的程序在内部还是外部,



本仿真板上有两个 CPU 插座，一个为 8051，一个为 80C196KC。仿真哪一种 CPU 就插哪一种 CPU 芯片，两种 CPU 芯片不能同时插。所插 CPU 的种



类应和跳线器的选择相同。

2.若要做 8088/86 的实验，将 POD8086 仿真板插在实验仪上，POD8086 仿真板如图：POD8086 仿真板上已包括 8088/86 常用的实验器件，有 8251 串行通信芯片，8253 定时器芯片，8259 外部中断芯片。并有这此实验所需的信号接线座。

### 三、板上仿真器使用方法

在无系统机的情况下，可以用实验仪上的键盘和 LED 显示器作为监控来进行仿真。若想做 8051 或 80C196 的实验，插上 POD51/96 仿真板，用跳线器 S1 选择 8051 或者 80196,并将 CPU 插入芯片插座，(两种 CPU 芯片不能同时插)。开启电源，在 LED 上出现 CPU 的型号，如 8051 或 8096。若要做 8088/86 实验，就要插上了 8088 仿真板，LED 显示的是“8088”。表示系统已经可以开始工作了。

采用这种工作方式时，仿真仪上的键盘和 LED 显示器既可作为监控系统用，也可以作为用户资源。当仿真仪未运行时，键盘和 LED 属于系统监控，用户用以输入程序，单步等，一旦运行时，显示器都作为用户资源。无系统机时，应先用汇编语言写好程序，用人工方法或者利用 PC 机把汇编语言翻成机器码，再用实验仪上的键盘将机器码输入，然后用 Trace 键，Here 键，Step 键进行调试，用 Exec 键使程序运行。

#### 3.1 键盘使用说明

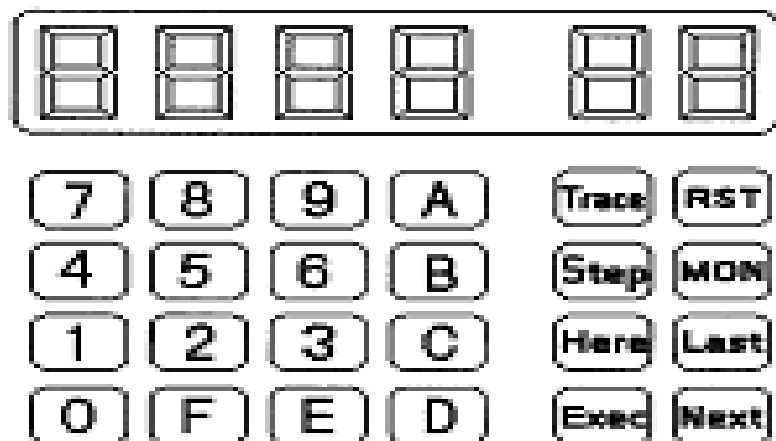


图 3.1 键盘和 LED 显示器

本实验仪共有 6 位 LED 发光数码管，左边 4 位为地址位，右边 2 位为数据位。当地址位 4 位数均有数值时，表示地址是程序存储器的地址或数据存

储器的地址。当地址数码管只有 2 位时（右边 2 位发光，左边 2 位暗），表示内部寄存器地址。当地址数码管只有 3 位时（最左面 1 位暗，第 2 位显示“0”，第 3 和 4 位是数字），表示是内部特殊寄存器(SFR)的地址。最右边的两位是数据位，它表示的是左边显示的地址单元中的数据。

面板上共有 24 个小键盘，为了介绍和使用时查找方便，约定用方括号表示按键，例如，[RST] 表示面板上的“RST”键。

### **3.1.1 [RST] 键 [整机复位键]**

整机复位键，复位后数码管的地址位显示“8051”字样（对于 MCS196 系列，显示“8096”，如果是 8088 仿真板则显示“8088”，表示复位操作完成。复位后程序存储器和外部数据存储器中的内容不变，程序指针回到 0000 处（8096 系列为 2080H。对于 8088 仿真板，初始地址为 1000H）。CPU 内部寄存器复位后为单片机复位操作规定的值，即有的寄存器的数据不变，有的寄存器中的数据被复位。

### **3.1.2 [0..F] 键 [数字输入键]**

数字键，用于输入 16 进制数。仿真器中的数据 and 地址均是用 16 进制表示。

### **3.1.3 [MON] 键 [监控键]**

表示某些操作已结束，某些操作的所有步骤完成后，按 MON 键表示操作结束。

### **3.1.4 [Trace]键 [跟踪执行键]**

在调试程序时先按 [MON] 键，最左侧一只 LED 数码管出现“P”字符，这时[Trace]档起作用，每按一次[Trace]便执行一条用户编写的单片机指令，如果遇到“CALL”这类调用命令时，跟踪到调用内部。执行条指令，可逐条检查用户程序的执行情况。

### **3.1.5 [Step] 键 [单步执行键]**

用户调试程序时每按一次 [Step] 键，执行一条命令。但是，当执行到调用

语句时，按 1 次 [Step] 键将执行调用所含的所有语句，[Step]与[Trace]不同的是，[Trace]遇到调用语句将进入调用语句内部仍然一步一步地执行。

### 3.6 [Last] 键 [地址减 1 键]

它有 2 个作用：

- 1.触发显示寄存器和存储器中的数据
- 2.地址减 1

触发显示寄存器和存储器中的数据

刚在地址位上输入了寄存器和存储器中的地址时，数据显示数码管是暗的未发光，此时，第一次按[Last]键，数据显示数码管开始显示数据。

地址减 1 操作

当在仿真器上输入地址后，在第二次以及之后的各次按键时，每按一次 [Last] 键，地址值自动减 1,地址值所对应的数据同时更换。

### 3.1.7 [Next] 键 [地址加 1 键]

它有 2 个作用：

- 1.触发显示寄存器和存储器中的数据
- 2.地址加 1

触发显示寄存器和存储器中的数据

刚在地址位上输入了寄存器或和存储器中的地址时，数据显示数码管是暗的未发光，此时，第次按[Next]键后，数据显示数码管开始显示数据。

地址加 1 操作

当在仿真器上输入地址后，在第二次以及之后的各次按键时，每按一次 [Next] 键，地址值自动加 1,地址值所对应的数据同时更换。

### 3.1.8 [Here] 键 [断点运行键]

设置中断程序运行的地址，使程序执行到中断地址处停止执行，在中断地址处等待新的操作命令（如 RUN. STEP, TRACE 等）。设置中断点时有两种情

况：

1.程序从 0000 地址（80C196 起始地址为 2080H，8088/86 的起始为 1000H）

执行到中断点

①按[RST]复位，再[MON]键，数码管显示“P”。

②输入 4 位地址。

③按[Here]。

按 Here 后，程序自动执行到中断点，此时显示中断点的地址和 A 寄存器中的内容。按 Here 键是确定中断地址，同时启动了程序，并使程序执行到中断点停下来等待用户的命令。

2.先使程序执行到某处，再设置中断点

①，先用单步(Step)、跟踪 (Trace)、中断等运行方式使程序执行的某处停止。

②. 按“MON”键，使仿真器地址数码管显示“P”。

③. 输入中断的地址值

④. 按"Here"键

⑤. 程序自动从设置前的停止处执行到所设的中断地址处。

### 3.1.9 [Exec] 键 [全速执行键]

用仿真器调试程序时，设置好执行的初始地址后，按[Exec]可自动执行程序。

## 3.2 脱机仿真

下面根据仿真时的各类需要，逐一介绍有关的操作方法。

### 3.2.1 将汇编源程序转换为机器码

在仿真工作之前，应把汇编源程序转换为机器码。可用人工查手册的方法逐条翻译成机器码，在翻译成机器码的同时还要为各条机器码安排地址。也可用计算机自动汇编并生成列表文件，列表文件是指同时含有源程序、机器码和机器码地址的文件。用计算机生成机器码可避免人工翻译造成的人为错误，

汇编的效率也远高于人工翻译。

### **3.2.2 输入程序的机器码**

#### **3.2.2.1 由计算机输入程序机器码**

为了节省输入机器码的时间，可先由计算机向仿真器输入机器码，然后再脱离计算机独自仿真。用计算机输入程序时先启动仿真软件，调入用户的汇编源文件，选择仿真模式，执行汇编命令，汇编结束后机器码自动装入仿真器中。然后，关断计算机和仿真器的电源，拔出 RS232 接口插头。

#### **3.2.2.2 人工输入程序机器码**

如果没有计算机时，则用人工方法输入。人工输入的方法是：

- (1) 按 [RST] 键，整机复位。
- (2) 按 [MON] 键，使地址数码管出现 “p” 字符。
- (3) 输入 4 位地址码，此时地址位上显示的是输入的地址数，数据位上的数码管不亮。
- (4) 先按[Next]或[Last] 键，数据位上的数码管闪烁，此时，再输入 2 位数据。
- (5)程序输入完毕，可依次按[RST]、[MON]、4 位地址数、[Next] 键，检查输入的机器码，不断地按[Next]或[Last]键，可依次逐个检查各个地址中的数据，数据形式是 16 进制。

### **3.2.3 执行程序**

#### **3.2.3.1 跟踪执行程序**

当需要由用户通过键盘控制，逐条执行程序，以便检查单片机内部和外部电路时，可采用跟踪执行的方式。跟踪执行时如果遇到调用指令，仿真器将使程序的指针进入被调用的程序段内部。跟踪执行键[Trace]，以跟踪方式执行程序的操作方法是：

- (1) 按 [RST] 键整机复位。

(2) 按 [MON] 键使地址数码管出现字符"P".

(3) 输入程序执行的起始地址，此时，4 位地址数码管显示地址值，数据数码管暗。

(4) 按[Trace]键，数据数码管亮，此后每按一次[Trace]键，执行一条指令。

### 3.2.3.2 单步执行程序

单步执行程序的作用与跟踪执行相仿，区别是单步执行遇到调用时，将自动连续地执行调用内部的所有指令，然后停在调用执行后的第一条指令上，可继续单步执行。单步执行键[Step] 的详细说明见第 5.1.5 条，以单步方式执行程序的操作方法是：

(1) 按 [RST] 键整机复位。

(2) 按[MON]键使地址数码管出现字符“ P ”。

(3)输入程序执行的起始地址，此时，4 位地址数码管显示地址值，数据数码管暗。

(4)按[Step]键，数据数码管亮，此后每按一次[Step]键，执行一条指令。

### 3.2.3.3 全速执行程序

当需从用户指定的程序地址处开始全速运行整个程序，或全速运行到断点处时，可采川全速执行方式。

全速执行的操作方法是：

(1) 按 [RST] 键，整机复位。

(2) 按[MON]键使地址数码管出现字符"P".

(3) 输入程序执行的起始地址，此时，4 位地址数码管显示地址值，数据数码管暗。

(4) 如果需要设置断点，可按上面介绍的[Here]命令。

(5)按[Exec]键，全速执行。程序运行时地址数码管数据数码管均暗。

(6) 按 [RST] 中断运行。

### 3.2.3.4 检查执行结果

当运用上述的各种执行方式运行程序时，需要及时地了解程序执行的结果，而运行结果很大程度上是由一单片机内部各个部分的当前值来反映的。仿真器提供了查看单片机（由仿真器模仿）内部各个部分情况的功能，检查执行结果的主要任务就是查看单片机内部的情况。

检查单片机内部各个寄存器、累加器、接口电路的方法在下而讲述。

检查和修改单片机内部寄存器数据

在调试过程中，如果需要查看当前状态下内部寄存器的情况，可按下列步骤进行：

- (1) 查出单片机内部寄存器地址码。如 R1 地址为 01H。
- (2) 根据内部寄存器地址码输入 2 位地址码，此时地址数码管的右 2 位显示地址数，左 2 位暗。
- (3) 按[Next]或[Last]键，使数码位出现内部寄存器中的数值。
- (4) 如果要查看相邻地址的寄存器的内容，可继续按[Next]或[Last]键。
- (5) 如果要修改内部寄存器中的内容，可按数字键[0]…[F]。

### A 检查和修改 8051 单片机内部专用寄存器(SFR)数据

在调试过程中，如果需要查看当前状态下单片机内部专用寄存器(SFR)的情况，可按下列步骤进行：

- (1) 查出单片机内部专用寄存器(SFR)地址码。如 ACC 地址为 E0H。
- (2) 仿真器规定专用寄存器地址要用 3 位数表示，专用寄存器的地址是 2 位，需先输入一个先导“0”，再输入 2 位专用寄存器地址码。地址输入后，此时地址数码管的右 3 位显示地址数，最左 1 位暗。
- (3) 按[Next]或[Last]键，使数码位出现一专用寄存器中的数值。
- (4) 如果要查看相邻地址的专用寄存器的内容，可继续按[Next]或[Last]



键。

(5) 如果要修改专用寄存器中的内容，可按数字键[0]…[F]。

注意：在查看专用寄存器时，不可按[RST]键，因为按此键后各个一专用寄存器中的内容将被复位。

**检查和修改单片机程序存储器和外部数据存储器**

(1) 按[MON]键，地址数码管出现“P”字样。

(2) 输入4位地址，此时4位地址数码管亮，2位数据数码管暗。

(3) 按[Next]或[Last]键，2位数据数码管亮，显示的即为4位地址单元中的机器码值，如要修改则可按[0]…[F]键。

(4) 再按[Next]或[Last]键可查看相邻的地址中的机器码值。

## **B 检查和修改 80C196 单片机内部专用寄存器(SFR)数据**

因为 80C196 地址是统一编码的，所以检查和修改单片机内部寄存器数据与修改程序区数据是一样的：

在调试过程中，如果需要查看当前状态下一些内部寄存器或 RAM 的情况，可按下列步骤进行：

(1) 查出单片机该寄存器地址码。如 int mask 地址为 0008H。

(2) 根据内部寄存器地址码输入4位地址码，此时地址数码管的右2位显示地址数，左2位暗。

(3) 按[Next]或 [Last] 键，使数码位出现内部寄存器中的数值。

(4) 如果要查看相邻地址的寄存器的内容，可继续按[Next]或[Last]键。

(5) 如果要修改内部寄存器中的内容，可按数字键[0]…[F]。

## **C 检查和修改 8088/86CPU 程序存储器或外部数据存储器（输入4位地址）**

(1) 按[MON]键，地址数码管出现"P"字样。

- (2) 输入 4 位地址，此时 4 位地址数码管亮，2 位数据数码管暗。
- (3) 按[Next]或[Last]键，2 位数据数码管亮，显示的即为 4 位地址单元中的机器码值，如要修改此内容，可按[0]…[F]数字键输入十六进制值。
- (4)再按[Next]或[Last]键可查看或修改相邻的地址中的机器码值。

#### 检查 8088/86 CPU 内部各个寄存器、累加器的方法如下（输入 2 位地址）

- (1) 按 [MON] 键，地址数码管出现“P”。
- (2) 因为 8088/8086 寄存器没有相应地址。我们为寄存器定义了以 F1 号，输入 2 位寄存器序号。
- (3)按[Next]或[Last]键，寄存器的名称和寄存器的内容就会显示在数码管的地址区和数据区。按[0]..[F]数字键可修改寄存器的内容。
- (4)按 [Next] 或[Last] 键，可观察/修改相邻寄存器的内容。

寄存器序号表：（十六位寄存器分成两个八位显示）

**表 3-1 寄存器序号表**

| 寄存器序号   | 寄存器序号   | 寄存器序号   | 寄存器序号   |
|---------|---------|---------|---------|
| AL 00H  | AH 01H  | BL 02H  | BH 03H  |
| CL 04H  | CH 05H  | DL 06H  | DH 07H  |
| SPL 08H | SPH 09H | BPL 0AH | BPH 0BH |
| SIL 0CH | SIH 0DH | DIL 0EH | DIH 0FH |
| IPL 10H | IPH 11H | FL 12H  | FH 13H  |
| CSL 14H | CSH 15H | DSL 16H | DSH 17H |
| SSL 18H | SSH 19H | ESL 1AH | ESH 1BH |

例如想观察堆栈指针 SP 的值，可以按以下步骤做：

- (1) 按 [MON] 键，LED 显示“P”字样。
- (2) 按[0]..[F]键，输入两位 SP 低字节序号“08”。

- (3) 按[Next]键，LED 地址区显示“ SPL ”，数据区显示的就是 SP 低八位值。
- (4) 按[Next]键，LED 地址区显示"SPH"数据区显示的为 SP 高八位值。

## 四、逻辑分析工具

### 一、逻辑分析仪

本实验仪带有 32 路逻辑分析仪功能，可以采样 4 路外部波形，28 路内部波形，让学生直观地看到电路的工作时序，增加感性认识。外部 4 路逻辑波形由实验仪上的 LA0..LA3 接入，信号来源由学生选择，任意接到实验仪上所要观察的数字信号，内部 28 路逻辑波形分别为 8 路数据信号，16 路地址信号，还有 RD, WR, PSEN, ALE 4 路控制信号。当用户程序运行时，逻辑分析就采样信号，当程序停下来时，仿真板将采样到的数据传到系统机上并显示出来。

图 4.1 是单片机在读写静态存储器(SRAM)时，采样到的波形。由于要写数据地址为 08234H,所以要将 RAM CS 接到 CS0 上，将 LA2 接到 RAM CS 上,以便观察,采样速率为 10M/S.当程序将数据 56H 写到 SRAM 地址 08234H 时，地址高 8 位从 P2 口（80C196 为 P4 口）输出，地址最高 3 位译码使 CS0 为低（见 LA2 信号），选中 SRAM 芯片，地址低 8 位由 P0 口（80C196 为 P3 口）输出，经过 ALE 锁存，做为地址低 8 位，然后从 P0 口(80C196 为 P3 口)输出被写数据 56H，在数据有效时，WR 信号变低，将数据写入 SRAM。完成一次写操作。



图 4.1 静态存储器采样到的波形

当程序从 SRAM 读数据时，时序与写数据相似，地址高 8 位从 P2 口（80C196 为 P4 口）输出，地址最高 3 位译码使 CS0 为低，选中 SRAM 芯片，地址低 8 位由 P0 口（80C196 为 P3 口）输出，经过 ALE 锁存，做为地址低 8 位，然后 P0 口（80C196 为 P3 口）浮空，等待数据，RD 信号变低，将数据从 SRAM 中读到 P0 口（80C196 为 P3 口）。CPU 将此数据送到寄存器，完成一次读操作。

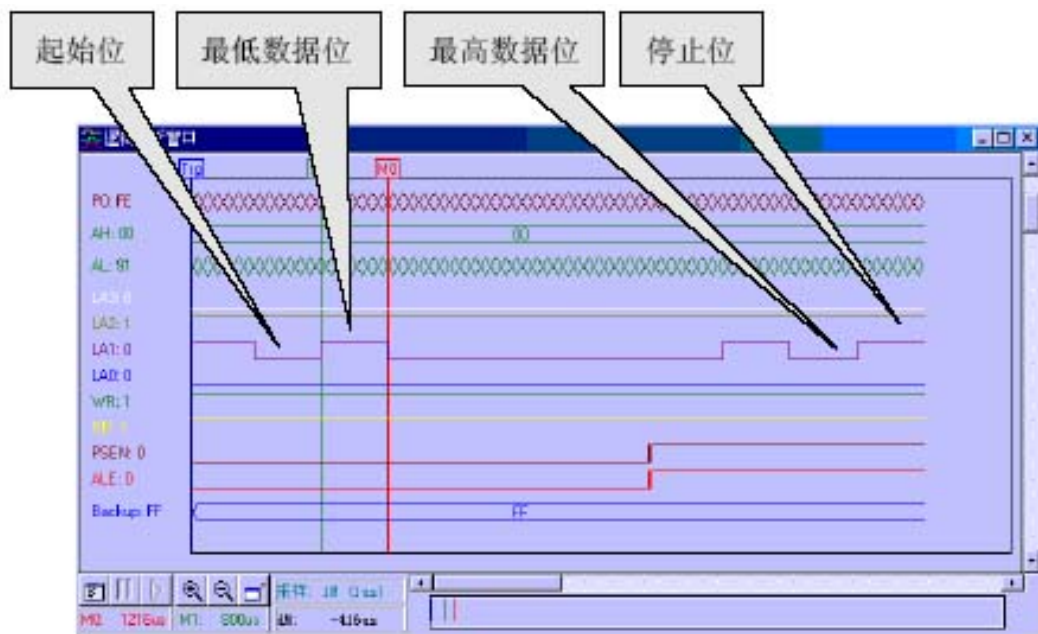


图 4.2 通作时的采样波形

图 4.2 是单片机在通信时采样的波形，采样速率为 1M/S。以 2400 特率发送字符‘A’，将 LA1 接到仿真板的 TXD 端，在程序中检测到 TI 后，延时一段时间再停止,以便采样到信号的停止位。

从采样到的波形可以看出，在 2400 波特率时，单个脉冲的宽度为 416us。字符‘A’的二进制为‘0100 0001’，从上图可以看到串口通信时，产生信号的全过程，先发一个低脉冲做为串行信一号的起始位，然后是数据的最低位，从低位到高位逐个发送数据位,最后再发一个高脉冲做为串行信号的停止位。完成一个串行通信的发送过程。

## 二、波形发生器

Lab2000P 实验仪带有一个 4 路的波形发生器，可用来产生可编程的数字信号。数字信号通过实验仪上的 LA0. . LA3 输出。

图 4.3 是用波形产生脉冲信号序列，LA3 接到步进电机的 A 相，LA2 接

步进电机的 B 相，LA1 接步进电机的 C 相，LA0 接步进电机的 D 相。经驱

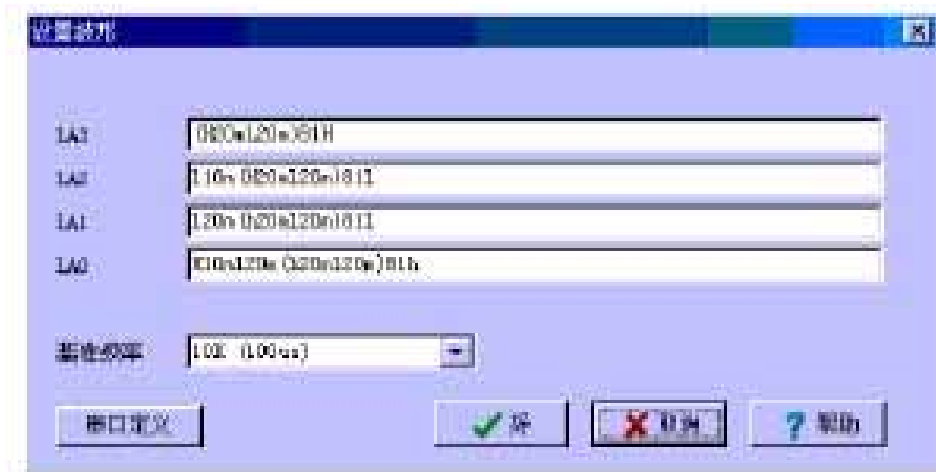


图 4.3 脉冲信号序列

动后控制步进电机的运转，采用双四拍控制方式，循环控制。波形定义说明如下：

LA3 的定义 (h20m 120m) 81 h，表示产生（高 20ms 低 20ms）的脉冲 81 个，然后保持高。直到循环。LA2 的定义 110m(h20m 120m)81 L，表示产生低电平 10ms 再产生（高 20ms 低 20ms）脉冲 81 个，然后保持低。直到循环。LA1、LA0 的定义依此类推。由此定义产生波形如下：



图 4.4 脉冲信号波形

## 五、系统自动检测功能

本实验系统带有自动检测功能，可以检测实验仪上各种电路的工作是否正常，这些电路包括 A/D 变换电路、D/A 变换电路、直流电机控制、步进电机控制、8255 控制、RAM 读写、液晶显示电路、温度传感器、键盘/LED 显示电路。

启动自检功能的方法是：

1)关闭电源。

2)随机附带有两根 20 芯自检电缆，将一根电缆的两端插到两个 TEST1 插座中，另一根电缆的两端插到两个 TEST2 插座中。TEST1 和 TEST2 两端连接的信号如一下图显示。

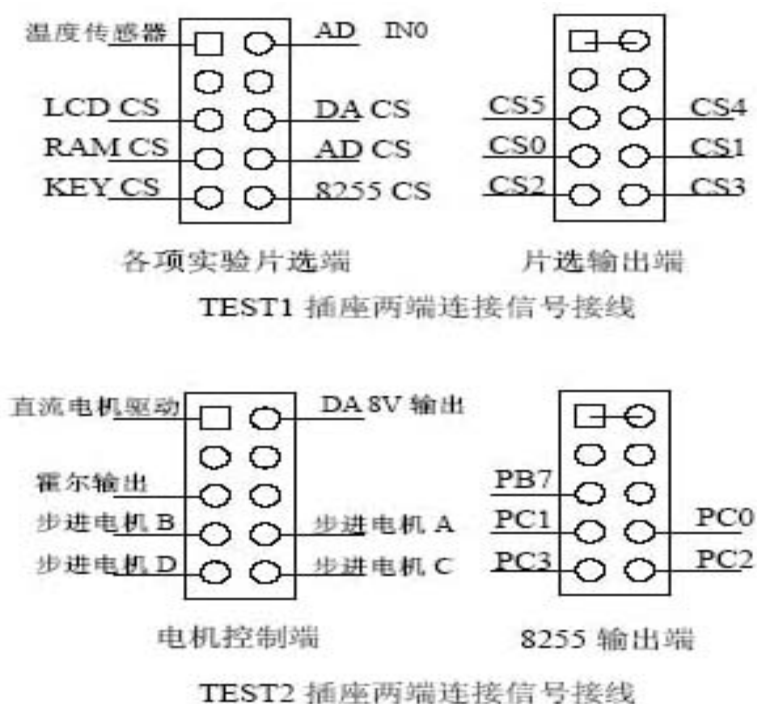


图 4.5 信号连接图



3)按住键盘上的 RST 键。

4)开启电源。

5)等到 LED 最右边无显示后，松开 RST 键。系统就进入了自动检测状态。  
在自检状态，程序流程如下：

#### 系统自检步骤：

1、LED 一位一位地显示“8”字，如果有一位不显示，检测相应位的输出，及位驱动是否正确。

2、.然后再显示八段码的各段，如果所有位的一段不亮，可能段输出时有错，如果是只有一位的一段不亮可能是 LED 管坏。

3、检测 RAM 时，对某个地址单元写数据，然后再读出数据进行比较，如果有错，会各项实验片选端片选输出端在左起第三位 LED 上显示‘-’，检查 RAM 芯片是否损坏。

4、在热敏及 AD 变换检测时，会在 LED 的左边显示温度值，因为是负温度系数传感器，所以当温度升高，显示数值会减小。可以用手抓紧温度传感器，使其升温，观察显示是否有变化。如果温度显示不变，检测热敏是否有输出及 A/D 变换电路是否工作正常。

5、检测液晶显示时，会在液晶上显示一条竖线，并自右往左扫描，在动态扫描时，显示的竖线会较淡，如果显示不对，可以按复位键，对液晶屏复位。也换液晶屏，检查其是否损坏。如果读不到液晶返回的信号，会在 LED 的第四位显示‘-’。

6、检测 8255 及步进电机时，由 8255 输出步进电机的驱动脉冲。如果电机不转，检查 8255 的 PC0~PC3 是否有输出及驱动块是否损坏。

7、检测 DA 及直流电机时，由 DA 输出直流电压，经驱动后，加在直流电机上，电机旋转后，霍尔元件会输出脉冲，经 8255 读入。如果 8255 读

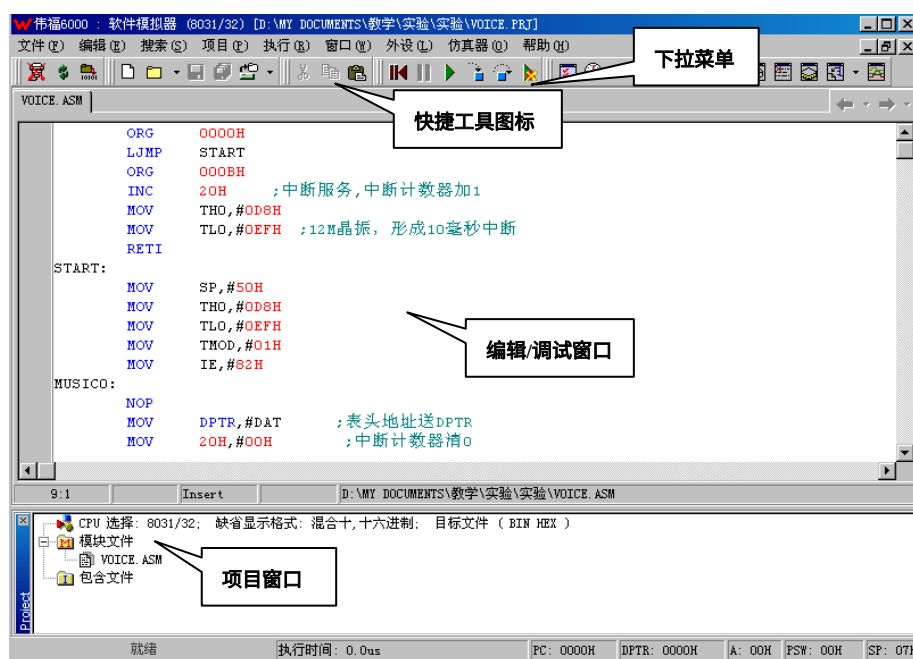
不到霍尔输出的脉冲，会在 LED 的第五位显示 ‘ - ’；此时应检查电机是否运转，霍尔元件是否损坏，8255 工作是否正常。

8、检测键盘及 LED 显示，在键盘上按下某键，会在 LED 的最右边显示该键值。如果显示不对，观察键盘是否被锁住。

## 六、仿真软件系统的使用

### 6.1、软件系统的安装和启动

1. 在 PC 机上安装伟福仿真开发系统集成调试软件。
2. 进行 8051 单片机实验时, 应插上 P0D51 仿真板, 并插上 8051 或 8052 CPU。
3. 将配套的串行通讯电缆的一端与实验仪上的“仿真器串口”9 芯 D 形插座相连, 另一端与 PC 机的串行口相连。
4. 将实验台的电源线与 220V 电源相连。(实验结束后应拔下)
5. 打开实验台电源开关, 红色电源指示灯亮。仿真开发器初始化成功后, LED 会显示 8051, 表示仿真系统正常。
6. 打开 PC 计算机电源, 执行伟福(WAVE)集成调试软件。集成调试环境界面如图 6.1 所示。



7. 通过下拉菜单“仿真器”对仿真实验系统进行设置, 如图 6.2 所示。

选择仿真器：伟福 Lab2000P 实验系统。

选择仿真头：MCS51 实验。

选择 CPU：(8031/32)。

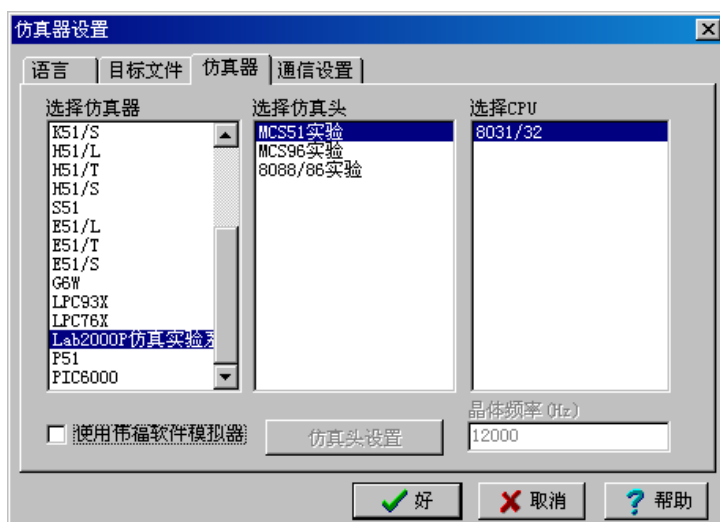


图 6.2 设置仿真实验系统

通过仿真器设置中的“语言”栏可以设置用户所采用的编程语言，如图 6.3 所示。采用汇编语言编程时，应选中“伟福汇编器”复选框，采用 C51 高级语言编程时，应选中“Keil C”复选框。

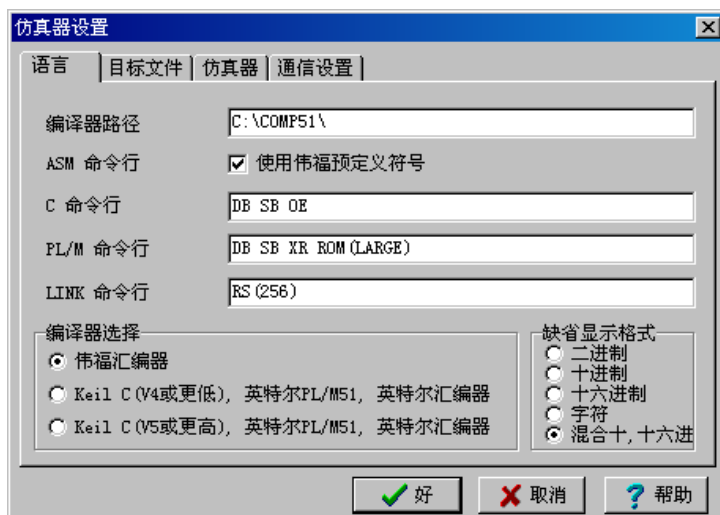


图 6.3 设置用户编程语言

通过仿真器设置中的“目标文件”栏可以设置最终生成的用户目标文件，一般可以按图 6.4 进行设置。

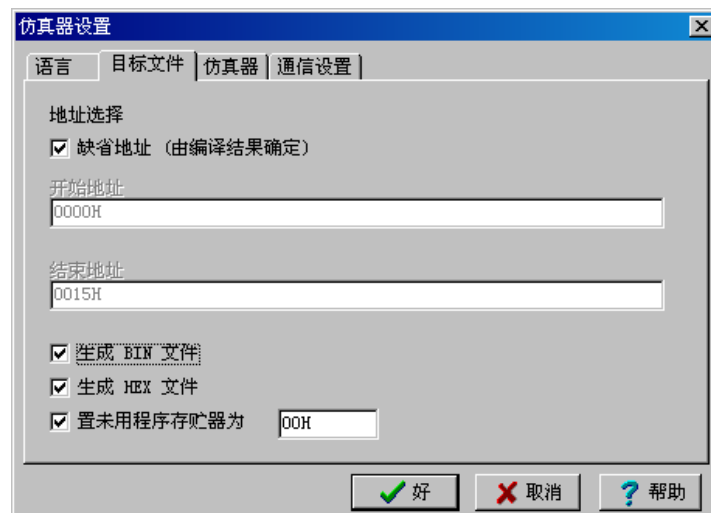


图 6.4 设置目标文件

通过仿真器设置中的“通信设置”栏可以设置集成调试环境与 PC 机的通信端口和波特率，如图 6.5 所示。

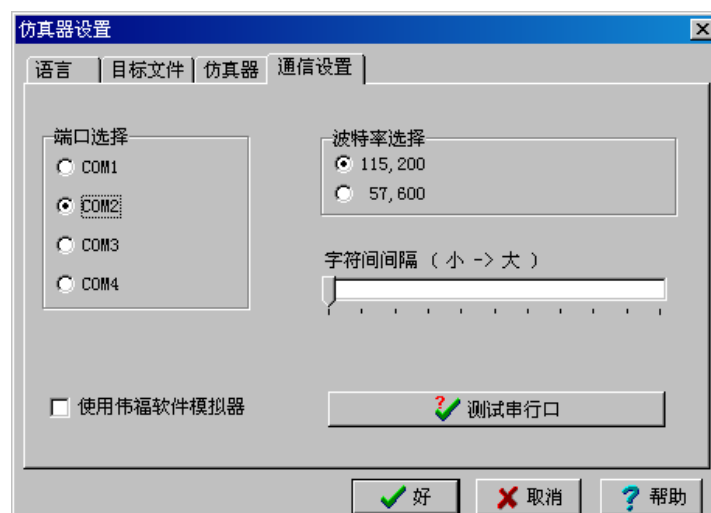


图 6.5 设置通信方式

## 2. 上机操作及调试步骤

(1) 启动伟福集成调试软件后，单击快捷工具栏中的“新建文件”图标

 (或按 Alt+F+N 键)，在弹出的编辑窗口中键入上述源程序文件，如图 6.6 和图 6.7 所示。

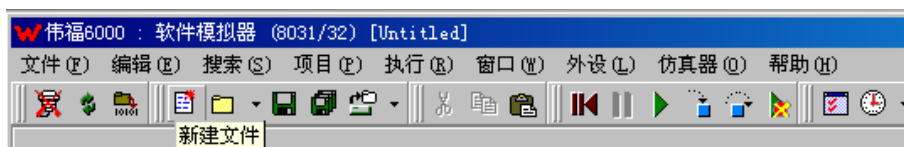


图 6.6 创建新文件

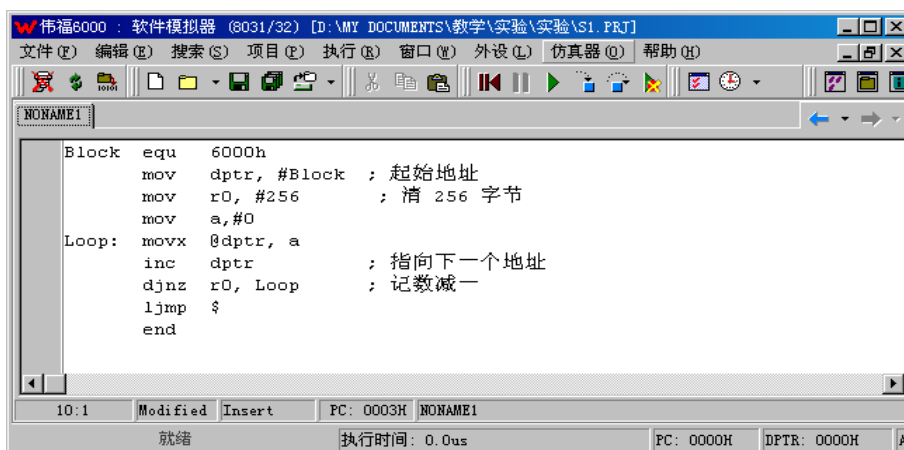


图 6.7 在编辑窗口中键入源程序文件

(2) 单击快捷工具栏中的“保存文件”图标  (或 F2 键)，在弹出的保存文件窗口中选定保存源程序的文件夹，并给定源程序文件名 (汇编语言源程序文件的扩展名为 “.ASM”)，单击“保存”按钮，如图 6.8 和 6.9 所示。

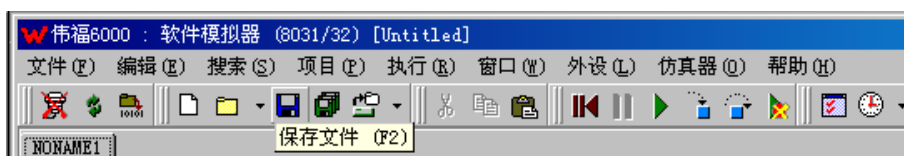


图 6.8 保存源程序文件

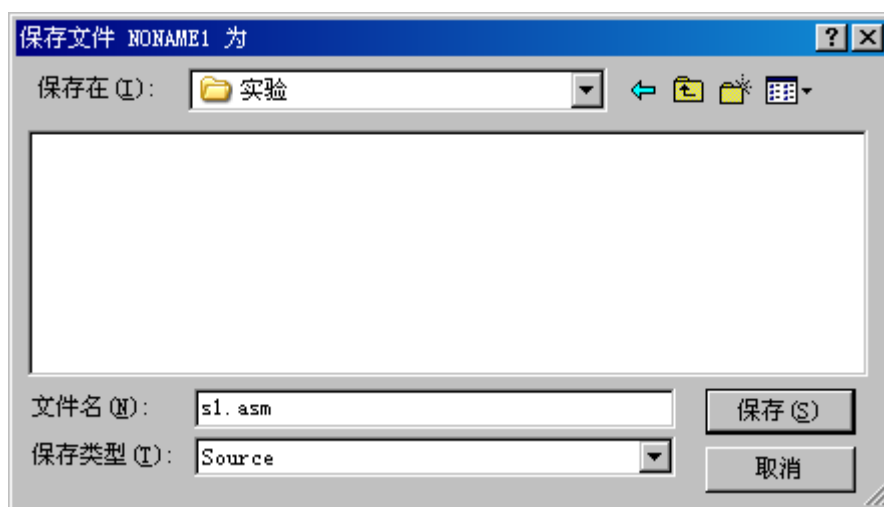


图 6.9 选定保存文件夹并给定文件名

(3) 单击下拉菜单“文件”中的“新建项目”，如图 6.10 所示，项目窗口如图 6.11 所示。

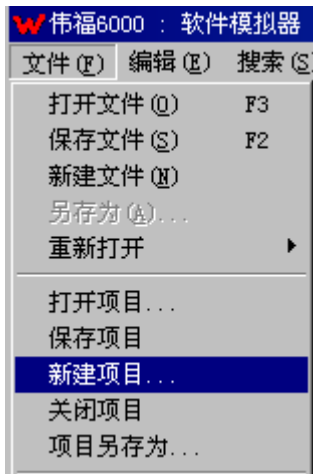


图6.10 新建项目

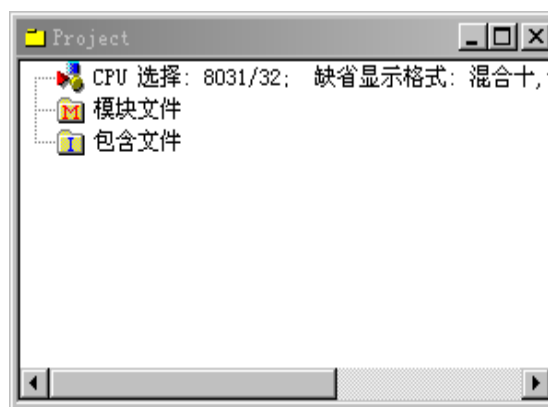


图6.11 项目窗口

新建项目会自动分如下三步：

①加入模块文件。在加入模块文件的对话框中选择刚才保存的文件 S1.ASM，按所示。打开键。如果是多模块项目，可以同时选择多个文件再打开，如图6.12所示。

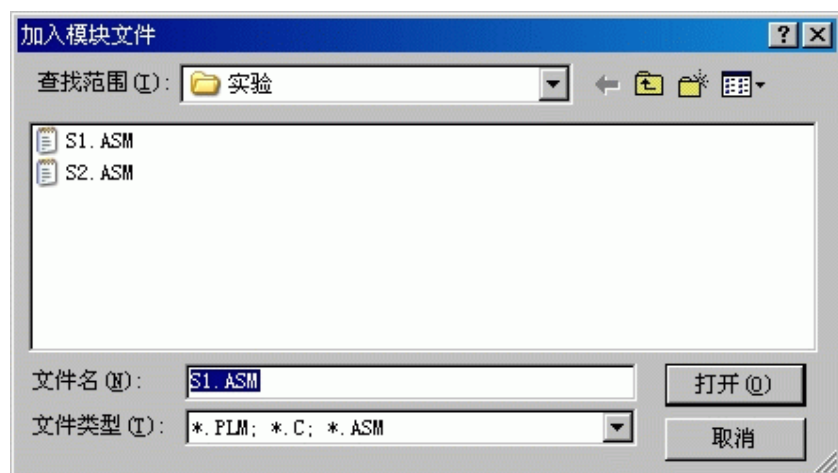


图 6.12 加入模块文件窗口

②加入包含文件。在加入包含文件对话框中，选择所要加入的包含文件（可多选）。包含文件通常用于多模块或高级语言编程中，如果没有包含文件，按取消键，如图6.13所示。

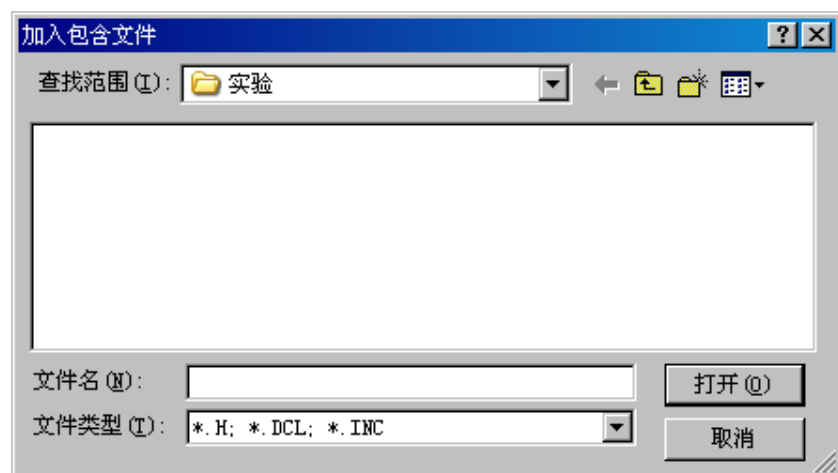


图 6.13 加入包含文件窗口

③保存项目。在保存项目对话框中输入项目名称“S1”，无须加后缀。



软件会自动将后缀设成 “.PRJ”。按保存键将项目保存到与源程序相同的文件夹下，如图6.14所示。

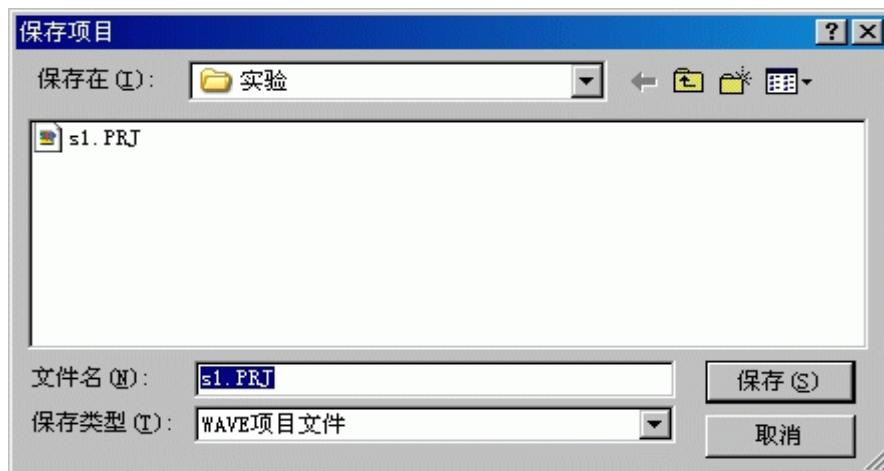


图 6.14 保存项目窗口

项目保存好后，如果项目是打开的，可以从项目窗口看到该项目中的“模块文件”。对于上例，可以看到项目中已经有了一个模块文件“S1.ASM”，如图6.15所示。

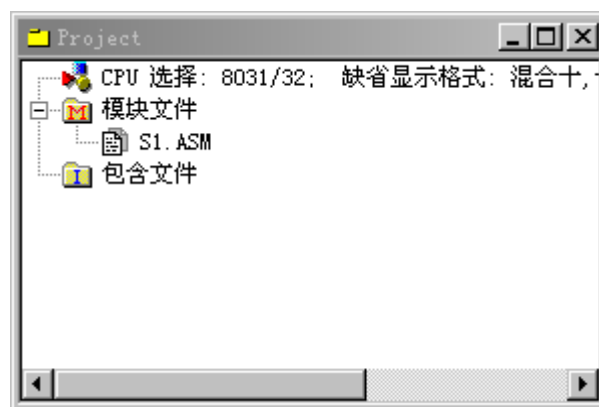


图 6.15 加入模块文件后的项目窗口

(4) 单击快捷工具栏中的“编译”图标 (或 F9 键)，对刚才保存的项目进行编译和连接，如图 6.16 所示。在编译之前，软件会自动将项目中的源程序文件存盘。如果发生编译连接错误，则在“信息窗口”中报告如图 6.17 所示编译连接出错信息。此时可将鼠标指向“信息窗口”中的错误行并双击鼠标左键，光标将自动跳转到编辑窗口中源程序的相应出错位置，便于用户修改。修改源程序文件并保存之后，重新进行第(3)步。如果编译连接通过，将在“信息窗口”中报告如图 6.18 所示编译连接正确信息。此时就可以对源程序进行调试了。

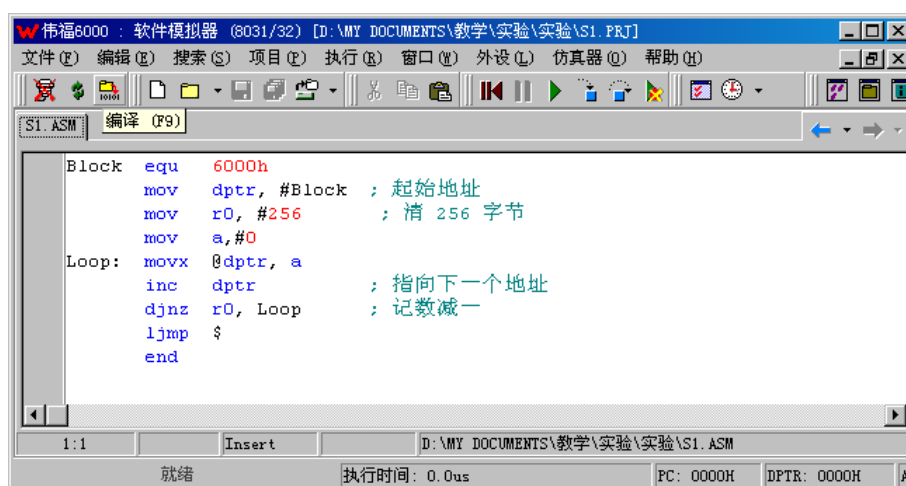


图 6.16 对已保存的项目进行编译连接

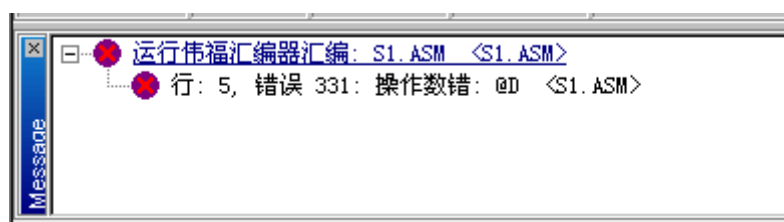


图 6.17 编译连接出错信息

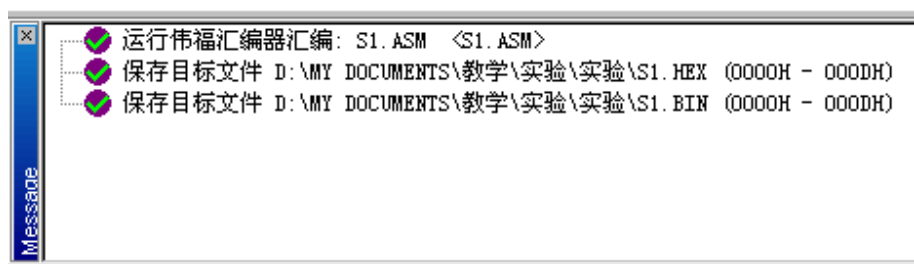


图 6.18 编译连接正确信息

(5) 单击快捷工具栏中的“跟踪”图标或按F7键启动单步跟踪调试程序，如图6.19所示。单步跟踪就是一条指令一条指令地执行，若有子程序调用，也会跟踪到子程序中去。这种方式可以观察程序每一条指令的执行的结果，光标“=>”所指的就是下次将要执行的程序指令。源程序窗口最左边的“o”代表此行为有效程序，此行在编译过程中产生了可以指行的机器指令。

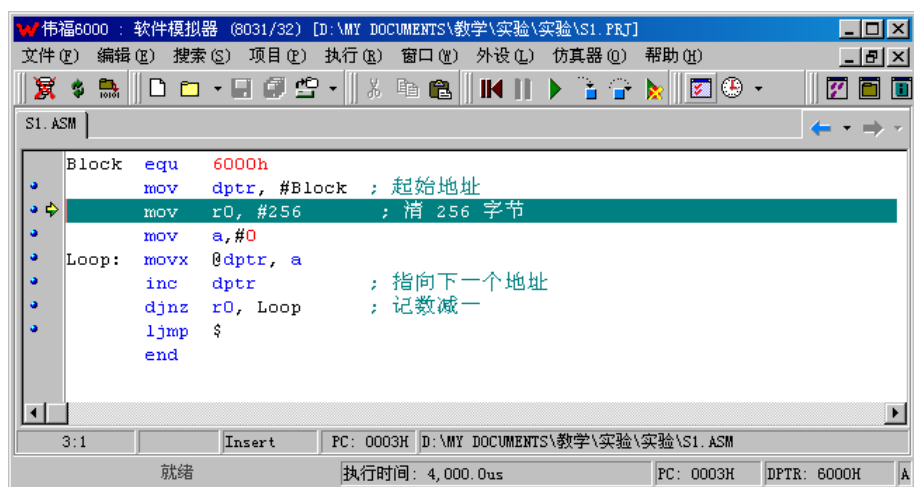


图 6.19 启动单步调试程序

跟踪调试执行速度较慢，也可以采用单步调试(快捷图标为)，此时若有子程序调用，不会跟踪到子程序中去，而是将子程序一次执行完成。还可以单击快捷图标将启动程序全速运行，单击快捷图标停止程序运行，

单击快捷图标将复位CPU。调试时除了上述快捷图标之外，还可以采用“执行到光标处”的功能，先将鼠标指向程序想要暂停的地方，例如程序中最后一条指令“LJMP \$”，单击鼠标右键，弹出一个右键菜单，选择右键菜单中的“执行到光标处”，程序将全速执行到指定行暂停，如图6.20所示。

右键菜单中还有其他一些功能，用户可以自己通过实践和阅读伟福仿真器的说明书来逐步掌握。下面介绍一下设置断点功能。

将鼠标移到源程序窗口的左边灰色区，光标变成“手指圈”，单击左键设置断点，也可以用右键菜单的“设置/取消断点”功能或用Ctrl+F8组合键设置断点。如果断点有效图标为“红圆绿勾”，无效断点的图标为“红圆黄叉”。

|                |         |
|----------------|---------|
| 执行到光标处(C)      | F4      |
| 设置PC           |         |
| 添加观察项          | Ctrl+F7 |
| 编辑观察项...       | Ctrl+F5 |
| 转到指定行(G)...    | Ctrl+G  |
| 转到指定地址/标号      | Ctrl+A  |
| 转到当前 PC 所在行(P) | Ctrl+P  |

图 6.20 部分右键菜单

断点设置好后，就可以用全速执行的功能，当程序执行到断点时，会暂停下来，这你可以观察程序中各变量的值，及各端口的状态，判断程序是否正确。

在程序调试过程中单击CPU快捷图标，可以观察CPU的运行状态，此时源程序窗口将出现反汇编(Disassembly)结果，以及工作寄存器和CPU寄存器窗口，如图6.21所示。可以通过这些窗口观察程序运行时CPU内部寄存器的变化情况。请读者在对上述程序进行调试的过程中，学会观察这些窗口中不同

寄存器内容的状态，从而判断程序运行是否正常。

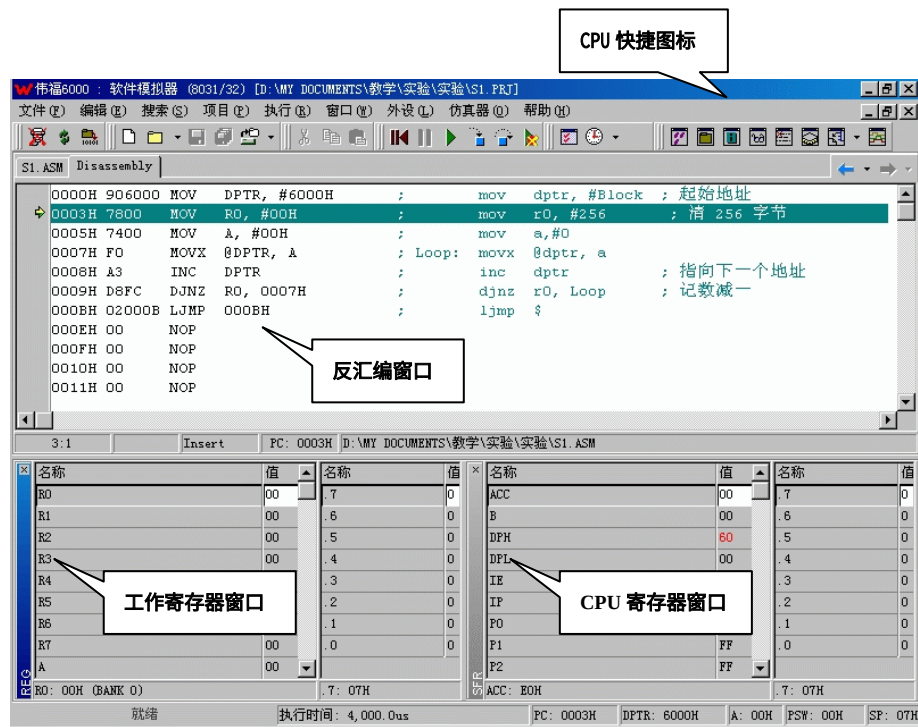


图 6.21 CPU 运行状态观察窗口

伟福集成调试软件中还提供了一个 DATA 快捷图标 ，单击该图标右边的小箭头，出现如图 6.22 所存储器空间选择菜单。通过该菜单可以分别观察 8051 单片机内部数据存储器空间(DATA)、程序存储器空间(CODE)、外部数据

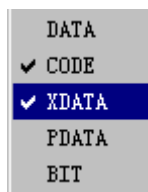


图 6.22 数据空间菜单

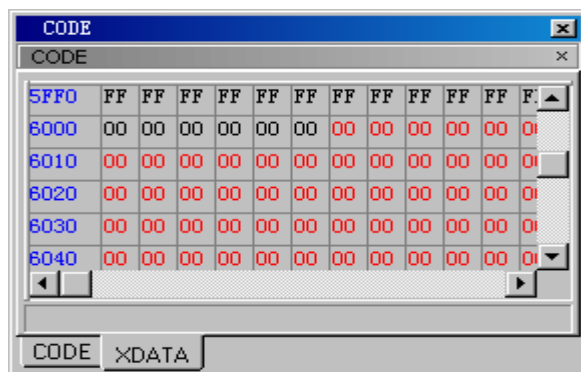


图 6.23 外部数据存储器 XDATA 空间窗口

存储器空间(XDATA)、外部数据存储器分页空间(PDATA)以及位寻址空间(BIT)。图 6.23 所示为本实验程序运行后的 XDATA 空间数据变化的情况。

至此,已经介绍了如何应用伟福集成软件开发环境调试 8051 单片机汇编语言程序的基本方法,读者可以通过实际应用来加深理解,例如若要将从 6000H 开始的 256 个字节外部数据存储器块的内容置成 11H 值,请修改以上程序完成此操作,注意在程序调试过程中数据的变化情况。