

上海交通大学电子信息学院

# 《嵌入式系统原理与实验》实验指导书

-基于 Proteus 的 8086 接口仿真实验

《嵌入式系统原理与实验》课程组

2014-2-1

# 实验一 IO 译码实验

## 一. 实验目的

熟悉 Proteus 软件的使用方法, 掌握 IO 译码方法和 8086 的存储器扩展方法。编写程序, 以 74LS244 作为开关输入口, 以 74LS273 为输出口控制 LED 灯显示。进一步掌握 IO 译码方法及 8086 汇编语言编程方法和 8086 的存储器扩展方法。

## 二. 实验电路及连线

见目录中 Proteus 文件: 8086 实验一.DSN

## 三. 实验要求

1. 掌握 Proteus 软件的使用方法, 在 Proteus 软件环境中能够绘制系统的连线图, 并能在 Proteus 软件仿真环境装载程序, 能够利用 MASM32 或者 Emu8086 软件编写 8086 汇编程序, 并编译生成可执行文件。

2. 按照所提供的电路图, 编写程序, 读入 74LS244 的开关状态, 将该状态取反后写入 74LS273 的端口。

3. 编写程序, 控制三色 LED 灯(可发红,绿,黄光), 模拟十字路口交通灯管理。

4. 如果要求 74LS244 的选片地址为 90H~97H, 74LS273 的选片地址为 A0H~A7H, 请修改电路图, 再完成实验要求 2。

5\*. 编写程序, 将 00H~0FFH 依次写入存储器地址 00000H~000FFH、00100H~001FFH、...、0FF00H~0FFFFH, 将程序加载的内存地址范围设定为 10000H~1FFFFH。

## 四. 实验说明

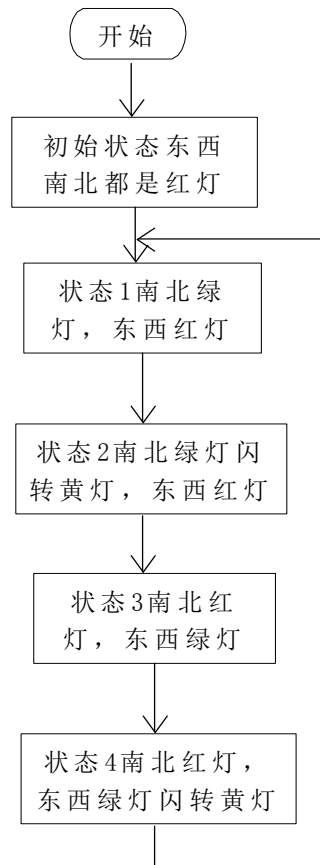
1. 对实验要求 4, 因为本实验是模拟交通灯控制实验, 所以要先了解实际交通灯的变化规律。假设一个十字路口为东西南北走向。初始状态 0 为东西红灯, 南北红灯。然后转状态 1 南北绿灯通车, 东西红灯。过一段时间转状态 2, 南北绿灯闪几次转亮黄灯, 延时几秒, 东西仍然红灯。再转状态 3, 东西绿灯通车, 南北红灯。过一段时间转状态 4, 东西绿灯闪几次转亮黄灯, 延时几秒, 南北仍然红灯。最后循环至状态 1。

2. 对应上述的各个步骤由多个演示环节, 建议在实验时完成每个步骤时, 保存原理图和对应的软件备份以备检查时使用。

3. 上述要求中带“\*”的为选做内容。

4. 提交实验报告是应附上带注释的程序源代码, 其风格参照实验一中提供的示例程序。

## 五. 实验程序框图 (实验要求 3)



273 的口状态

状态	含义	273 端口状态 D7---D0
初始状态	东西南北均为红灯	××110110 36H
状态 1	南北绿灯；东西红灯	××110011 33H
状态 2	南北绿灯闪转黄灯；东西红灯	南北绿，东西红：××110011 33H 南北绿黄，东西红：××110001 31H
状态 3	南北红灯；东西绿灯	××011110 1EH
状态 4	南北红灯闪转黄灯；东西绿灯	南北红，东西绿：××011110 1EH 南北红，东西绿黄：××001110 0EH

## 六.实验结果

应能在 Proteus 环境中演示实验结果。

## 实验二 内存扩展和 I/O 空间操作实验

### 一. 实验目的和内容

1. 理解内存地址空间和 IO 空间操作方法的异同, 学习利用静态内存芯片扩展系统存储器的电路设计方法, 并且掌握内存映射到地址空间中不同区域的连线方案, 理解奇偶地址、数据总线高字节、低字节和奇偶内存芯片的对应关系。

2. 学习 8255 芯片接口的方法, 进一步学习微处理器的汇编语言编程技术。编写程序, 以 8255 作为输入输出, 完成接收开关量输入, 控制 LED 灯和数码管的正常显示。

### 三. 实验电路及连线

实验素材包括在实验 2 目录下的实验二 8255Demo.DSN 原理图文件、实验二 8255Demo.asm 和实验二内存扩展.asm 二个汇编源代码文件。

### 三. 实验要求

1) 8255 芯片接口操作要求:

1. 打开上述原理图文件, 分析原理图, 计算 8255 芯片在 IO 空间中的地址, 列出 8255 中四个寄存器的端口地址, 查看 8255 芯片数据端口与 8086 芯片数据总线的连接方法。

2. 导入实验二 8255Demo.asm 源文件并编译运行, 应能能在数码管上显示 0,1,2,3, 并点亮所有发光二极管。

3. 修改步骤 2 的源文件, 使其接收开关量输入, 将 8 位开关量的高 4 位以二进制形式显示在发光二极管上, 低 4 位的状态显示在(第一位)数码管上。

4. 修改程序, 接收开关量输入, 将 8 位开关量的高 4 位和低 4 位分别以十六进制形式显示在第 1、2 个数码管的上, 第 3、4 个数码管重复第 1、2 个数码管的显示; 开关量输入高 4 位的状态显示在 4 个发光二极管上。

5\*. 调整 8255 芯片数据端口连线到 8086 芯片数据总线的 D8-D15, 重复上述的步骤 2-步骤 4。

2) 内存扩展方案设计

1. 8255Demo.DSN 原理图文件, 通过对所提供的原理图中存储芯片解码方案的分析, 计算出该芯片映射到内存地址空间的地址和内存容量的大小;

2.a 打开“实验二内存扩展.asm”源文件运行并观察 U10 和 U11 内存芯片的内容。其方法是先运行程序一段时间后暂停, 勾选菜单项“Debug”->“Memory Contents -U10”选项和勾选菜单项“Debug”->“Memory Contents -U11”选项, 弹出 U10 和 U11 二个 62256 内存芯片的内容观察窗口, 其内容应该均为 FFh。该演示程序是采用单字节写入的方法完成内容的填写。要求修改上述程序, 采用单字节写操作方法, 将所有奇数地址的内存单元从低地址到高地址依次填入无符号数 1、3、5、7、。。。。。。; 通过 Proteus 的观察存储芯片内容的查看功能, 观察原理图中二个内存芯片 U10 和 U11 的写入内容与预想结果做比较;

2.b 继续修改程序, 采用单字节写操作方法, 将所有偶数地址的内存单元从低地址到高地址依次填入无符号数 0、2、4、6、。。。。。。; 通过 Proteus 的观察存储芯片内容

的查看功能，观察原理图中二个内存芯片 U10 和 U11 的写入内容。

3. 修改原理图，将内存芯片映射到内存地址从 80000h 开始的地方，重复上述的步骤 2-步骤 3。

4\*. 编写程序，采用双字节写操作方法，把 55AAh 填满上述的二个内存芯片通过 Proteus 的观察存储芯片内容的查看功能，观察原理图中二个内存芯片的写入内容。

#### 四. 实验说明

1. 对应上述的各个步骤由多个演示环节，建议在实验时完成每个步骤时，保存原理图和对应的软件备份以备检查时使用。

2. 上述要求中带“\*”的为选做内容。

3. Proteus 的 8086 芯片仿真在特定情况下会发生错误，使其产生的信号与实际情况不符合。针对此仿真问题，在汇编源代码中加入了补丁（Work Around）代码解决。该部分代码在源代码中有注释，在使用过程中请不要修改、删除或者移动其位置，简单忽略其存在即可。

4. 提交实验报告是应附上带注释的程序源代码，其风格参照实验一中提供的示例程序。

5. 原理图中提供了逻辑分析仪和数字波形图的演示，供参考使用。

#### 五. 实验框图

略

#### 六. 实验结果

应能在 Proteus 环境中演示实验结果。

## 实验三 定时器、计数器和中断应用实验

### 一. 实验目的和内容

实验目的要达到以下二点要求：

1. 学习可编程定时/计数器接口芯片 8253 的使用方法，掌握 8253 的基本组成和功能、控制字格式、各个寄存器的地址分配、6 种工作方式的特点（尤其是方式 0、2 和 3）和使用方法。

2. 理解中断的基本概念、中断的处理过程和中断的优先级管理，掌握 8086 系统的中断指令、中断分类、中断向量表、中断入口地址、中断处理过程和中断服务程序等。熟练掌握 8086 中断服务程序的编写方法，了解 ISR 的编写与一般子程序编写方式的区别，认识到 ISR 对系统性能可能会产生的影响。

3. 了解采用 16 位数据总线扩展 8 位外围设备的方法。

实验内容包括：

1. 采用子程序调用的方式完成 8255 芯片的初始化，将 8255 设置成 PA 和 PB 口驱动 7 段数码管显示，PC 口配置为可读取外部拨位开关信号，并可输出开关量。具体参数可参照原理图完成。
2. 采用子程序调用的方式完成 8253 芯片的初始化，将 OUT0 端、OUT1 端和 OUT2 端分别配置为分频、计数和超时检测应用模式。
3. 采用定时器中断完成程序定时运行。
4. 编写采用中断方式完成看门狗应用的喂狗程序（注 1）。

注 1：看门狗应用及其喂狗程序的原理请自行寻找资料学习。在实际的应用中，看门狗电路作用时将触发电路系统的复位程序以重新启动系统。本实验未将复位信号连接到系统复位，为演示其作用，用蜂鸣器和 LED 信号灯代替。看门狗电路发生作用时，将在蜂鸣器上发出报警声音和灯光示警。

### 二. 实验电路及连线

实验素材包括在实验 3 目录下的实验三 8253Demo.DSN 原理图文件、实验三 8253Demo.asm。

原理图简要说明如下：

- 1) 8255 芯片使用 PA 和 PB 端口驱动七段数码管显示，PCU 和 PCL 端口用于数据输出和开关量输入。
- 2) 8253 芯片的 Timer0 口用于对输入的时钟信号做分频；Timer1 口级联前级 Timer0 的输出作为输入；Timer2 口被用于看门狗功能验证。
- 3) U15 为中断请求信号生成逻辑；DSW1 用于设定中断矢量号。
- 4) LS1 为看门狗蜂鸣器。
- 5) 调试仪器部分请参考仪器下方红色文字的用途说明。

### 三. 实验要求

- 1) 8255 芯片初始化

1. 分析原理图 8255 芯片周边的连线，与实验二中的相应电路作比较，计算 8255 内部寄存器的端口地址和范围，修改“实验三 8253Demo.asm”源程序中的相应定义完成 IO 空间地址的符号定义。分析原理图中 PA、PB 和 PC 口的信号传输方向确定各端口的输入及输出状态，填充源代码框架中 8255 芯片初始化子程序，完成 8255 芯片初始化代码。

2. 参考实验二的 8255 实验部分相应代码，同时点亮四个 7 段数码管，输出学号的最后四位数字。

3. 分析原理图中 PC 口的电路功能，完成对 PC 口各信号线的控制。编写程序，实现对 PC 口 PC0 的开关信号量查询，将其值输出到 PC 口的 PC6 引脚，C 口其他引脚的值不可改变（注 2）。

注 2：经测试 Proteus 对 8255 仿真操作时，其 BSR 模式存在问题，应该是 Proteus 的 Bug，请在对 C 口操作时请勿使用 BSR 模式。

### 2) 8253 芯片初始化和计数器级联应用

1. 分析原理图，计算 8253 芯片在 IO 空间中各内部寄存器的端口地址和范围，修改“实验三 8253Demo.asm”源程序中的相应定义完成 IO 空间地址的符号定义。

连接 8255 芯片的 PC6 信号到 8253 的 GATE0 管脚。分析原理图中 8253 芯片 Timer0 的外部引线连接。计算 Timer0 的设置参数，使其能够在 OUT0 段输出 10ms 周期的方波信号。填充源代码框架中 8253 芯片初始化子程序，完成 8253 芯片及 Timer0 的初始化代码。

运行程序并且：

A) 设置 PC0 连线上的拨位开关到“OFF”，用示波器观察 OUT0 的输出信号波形并分析其原因。

A'\*) 使用逻辑分析仪观察 CLK0，GATE0 和 OUT0 波形。

B) 设置 PC0 连线上的拨位开关到“ON”，用示波器观察 OUT0 的输出信号并分析其原因。

B'\*) 使用逻辑分析仪观察 CLK0，GATE0 和 OUT0 波形。

2. 修改原理图，将 Timer0 的输出信号 OUT0 级联到 Timer1 的输入时钟端。计算 Timer1 的设置参数，使 Timer1 的 OUT1 输出端口产生周期为 1s 的方波信号。填充源代码框架中 8253 芯片初始化子程序，完成对 Timer1 的初始化代码。

运行程序并且：

A) 设置 PC0 连线上的拨位开关到“OFF”，用示波器观察 OUT0 的输出信号。

B) 设置 PC0 连线上的拨位开关到“ON”，用示波器观察 OUT0 的输出信号。

3\*. 计算配置 8253 的 Timer2 定时器为模式 0 的参数，准备用于看门狗模式。

### 3) 中断应用方案设计

本实验采用 8086 的外部中断信号 INTR 作为中断信号源，在每次中断产生后进入中断服务程序，通过对连接在 PC7 上的 D13 LED 灯的开关的操控，观察中断程序的进入情况。接着将编写定时中断服务程序，完成对步骤 3.1.1 中的显示管软件延时定时程序功能的替代。

1. 分析原理图中的 U15 芯片，理解手动产生 INTR 信号的逻辑和在 INTA 的作用下清除 INTR 信号的原理。按照以下方式确定要设计的中断服务程序矢量号码，并编写程序完成中断向量表相应位置的中断矢量初始化：

- a) 将学号的最后四位与十进制值 224 做模运算取余数;
  - b) 将余数加上 32 作为后续所设计的中断服务程序矢量号, 以备在后续实验步骤中使用;
2. 完成中断服务程序的编写, 采用软件中断指令进入该中断服务程序, 完成其调试; 然后用 BT1 按钮产生手动硬件中断的请求信号, 可以观察到由硬件中断产生的该中断服务程序被调用, 用单步执行的方式完成中断服务程序运行:
- A) 先采用软件中断调试方式完成中断服务程序调试:
- a) 编写中断服务程序, 实现在每次进入中断时, 将 PC7 的开关量取反后再输出到 PC7, 实现程序的第奇数次进入时 D13 点亮、偶数次进入时 D13 熄灭, 或者相反。
  - b) 编写中断向量表初始化子程序, 完成由学号计算所得的中断号在中断向量表入口点地址的初始化。
  - c) 在主程序中采用软件中断指令调用上述编写完成的中断服务程序, 完成中断服务程序调试, 观察 D13 的 LED 灯明灭变化。
- B) 分析中断请求信号置位/清除、中断矢量设置电路的原理, 修改原理图, 把手动硬件中断信号连接到 8086 芯片。
- a) 把编号为 U15 的 74LS74 的 Q 输出端连接到 8086 芯片的 INTR 信号端; 把低电平有效的 R 端 (即 Reset) 连接到 8086 芯片的 INTA 信号端。
  - b) 在硬件逻辑上调整拨位开关 DSW1, 连接 U11A 和 U11B 的 OE 端到 8086 芯片的/INTA 信号端。
  - c) 确认程序执行的第一条指令为 CLI 关中断指令; 修改程序, 将 3.2.A.c 中编写的软件中断指令删除; 添加读取 DSW1 拨位开关数值的指令 (在 IO 空间中用/Device4 信号激活), 在此指令位置用设置断点的方法停止程序运行, 运行该指令后通过检查 CPU 中的寄存器确定开关设置了正确的中断矢量值。
  - d) 在主程序中开中断运行, 应能够观察到在每次按下按钮开关 BT1 (手动产生中断信号) 时 D13 LED 灯的明灭变化。
- 3\*. 采用中断方式实现程序代码定时运行:
- a) 完成手动定时方式。按照以下要求修改 3.1.1 部分的 8255 芯片 7 段显示管显示控制逻辑, 并用 BT1 手动产生中断信号实现 4 位显示管的逐位显示, 观察数码管的显示状况。(提示: 可以利用 PC7 的变化 (D13 LED) 观察到中断的进入情况)
    - 主程序的无限循环按照指定变量值(假定其范围为 0~3, 分别代表从左到右的第 1 到第 4 个显示管)的要求刷新 4 个显示管中的指定一位;
    - 中断服务程序在每次进入时调整上述的该指定变量值指向下一个显示管位置。也就是每次进入程序时将上述的变量值加 1, 并做模 4 运算取余数后存回。
4. 完成定时器中断程序的软件和硬件配置并运行:
- a) 利用 OUT1 输出做中断源定时。

修改原理图, 把 8253 产生的 OUT1 信号连接到 U15 的 CLK 端替代手动生成中断信号逻辑, 观察 D13 LED 灯的明灭变化和看门狗警告信号。
  - b) 利用 OUT0 输出做中断源定时。

修改原理图, 把 8253 产生的 OUT0 信号连接到 U15 的 CLK 端替代 OUT1 产生中断信号逻辑, 观察 D13 LED 灯的明灭变化和看门狗警告信号。



c)\* 利用 OUT1 输出做中断源定时。

修改原理图，把 8253 产生的 OUT1 信号连接到 U15 的 CLK 端，观察数码管显示情况和看门狗警告信号。

d)\* 利用 OUT0 输出做中断源定时。

修改原理图，把 8253 产生的 OUT0 信号连接到 U15 的 CLK 端，观察数码管显示情况和看门狗警告信号。

5.\*\* 试编写程序，把进入中断的次数在以十进制的方式显示在 7 段数码显示管上。

## 四. 实验说明

1. 实验源代码需要修改后才可以编译通过执行。在实验的各个步骤中，所有对源代码的修改都将采用逐步添加的方式完成，不需要把前期的修改删除再添加后续的新代码。示例源代码已经构建了实验所需要的完整程序框架，建议在此框架内参照源代码注解完成实验代码的修改。

2. 对应上述的各个步骤由多个演示环节，建议在实验时完成每个步骤后，保存原理图和对应的软件备份以备检查时使用。

3. 建议充分利用各种硬件输出显示来协助程序的调试，如 D13 等。熟练掌握逻辑分析仪和示波器的使用将对于程序的调试起到非常重要的帮助作用。

4. 上述要求中带“\*”和“\*\*”的为选做内容，其中“\*\*”的内容有较大的难度和工作量。

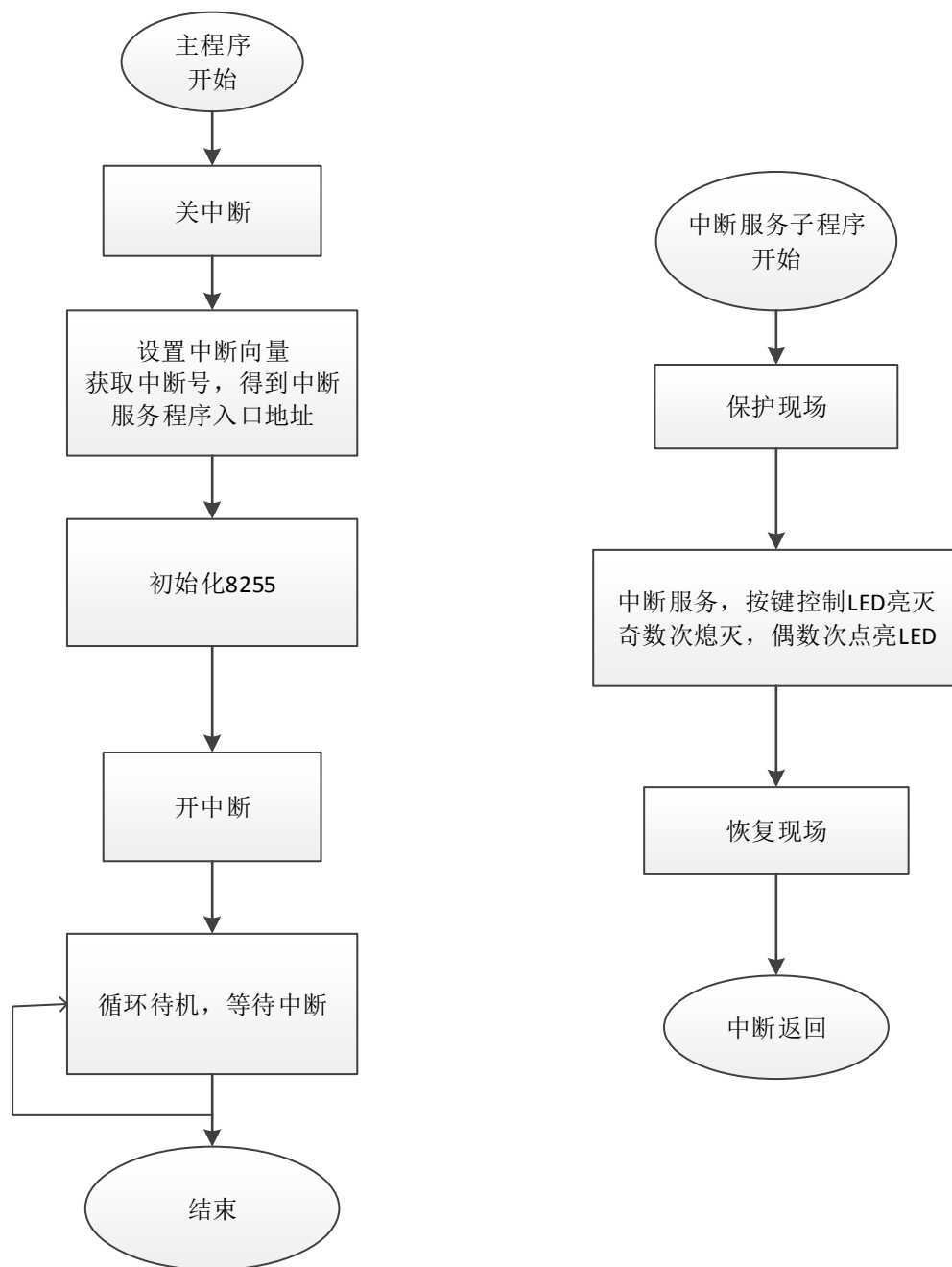
5. 提交实验报告时应附上带注释的程序源代码，并附上程序流程图，代码风格建议参照实验中提供的示例程序。

6. 推荐使用免费的 Notepad++ 作为汇编程序源代码编辑器，具有语法高亮显示和多文件编辑能力。可从网上搜索下载。

7. 因为 Proteus 程序在仿真 8086 硬件中断时存在严重错误，因此在原理图和程序中做了相应的补丁处理，请不要删除，直接忽略该部分代码和电路即可，详细信息看对应的说明。

## 五. 实验框图

部分实验步骤的参考程序框图：



其余省略

## 六.实验结果

应能演示实验结果。