

Proteus研究型实验报告

8255流水灯

|  |  |
| --- | --- |
| 专 业： | 计算机科学与技术 |
| 学 院： | 詹天佑学院 |
| 参与者： |  |
| 分工情况： |  |
| 提交日期： |  |

**一、功能描述**

在Proteus仿真软件中实现8255流水灯接口实验的相关内容。

具体内容为：8255-A并口芯片PA口接8个拨动开关K0-K7，PB口接8个LED。初始由开关K0-K7设定8位不同的值，当执行程序后LED按K0-K7初始设定的值点亮，并向右流动（8255A工作在0方式）。选择完成在数码管上显示“8255-A”。

由于本实验所需的基本代码已在接口实验中实现，因此本实验的**重点和难点**在于结合Proteus仿真软件部分。主要有三点：8086模式下地址和数据的分配、CPU与数码管间I/O接口芯片的设计、实际代码与硬件电路的匹配。

**二、硬件电路设计**

（1）基于8086 CPU地址、数据总线的搭建

结合参考例程，可以搭建如图1所示的电路。由于8086的AD[0..15]存在复用，因此利用3片74273八D触发器和总线将CPU的数据和地址总线分离标注，其中ADR[0..19]作为地址总线，AD[0..15]作为数据总线。

图示, 示意图

描述已自动生成

图1 基于8086 CPU地址、数据总线的搭建

（2）I/O设备片选信号产生电路

使用1片74145译码器对4位地址信号进行译码后选中对应的I/O设备（IO0-IO15），如图2所示。其中，ADR[8..11]作为四位地址译码，信号和ADR[12..15]的或作为使能信号。

图示, 示意图

描述已自动生成

图2 I/O设备片选信号产生电路

（3）8255并口芯片

根据实验要求，8255-A并口芯片的PA口接8个拨动开关K0-K7，PB口接8个LED D0-D7。八根数据线D0-D7与CPU数据总线AD相连；片选端接，作为2号IO设备。

**8255**-A**各端口地址的确定**：

结合（2）中I/O设备片选信号产生电路，选中IO2时，ADR15-ADR12为0，ADR11-ADR8为0010。8255-A地址线A0与ADR1相连，地址线A1与ADR2相连，因此可以列出8255-A各端口的地址如表1。

|  |  |  |  |
| --- | --- | --- | --- |
| 端口 | ADR11-ADR8 | ADR7-ADR0 | 十六进制 |
| X0（PA） | 0010 | 00000000 | 200H |
| X1（PB） | 0010 | 00000010 | 202H |
| X2（PC） | 0010 | 00000100 | 204H |
| X3（命令口） | 0010 | 00000110 | 206H |

表1 8255-A端口地址

图示, 示意图

描述已自动生成 图示, 示意图

描述已自动生成

图3 8255并口芯片

（4）数码管I/O接口芯片

利用Proteus软件搭建子电路的功能搭建数码管的I/O接口，外部管脚如图4。

整体包括：

* 八根数据线D[0..7]，与数据总线AD[0..7]相连。
* 片选信号，低电平有效，接，作为4号IO设备。
* 地址线A0接地址总线ADR1，用于芯片内部端口寻址，可形成两个端口地址。
* 控制线写信号，低电平有效，配合数据线，用于写入数据。
* 面向数码管的输出信号线：Q[0..7]向数码管发送字形代码，SE[0..7]向数码管发送位选代码。

示意图

描述已自动生成

图4 数码管I/O接口外部管脚

数码管I/O接口内部封装如图5。使用2片74LS373八D锁存器分别对位选代码和字形代码进行输出（图5中上片输出位选代码，下片输出字形代码），并分别设计逻辑电路进行内部端口寻址。

**数码管I/O接口各端口地址确定**：

结合（2）中I/O设备片选信号产生电路，同样可以得到数码管I/O接口端口地址如表2。

|  |  |  |  |
| --- | --- | --- | --- |
| 端口 | ADR11-ADR8 | ADR7-ADR0 | 十六进制 |
| Q（字形） | 0100 | 00000000 | 400H |
| SE（位选） | 0100 | 00000010 | 402H |

表2 数码管I/O接口端口地址

图示, 示意图

描述已自动生成

图5 数码管I/O接口内部封装

（5）七段数码管

使用7SEG-MPX6-CC七段共阴极数码管，如图6。字形代码高电平有效，位选代码低电平有效，分别与I/O接口芯片Q[0..7]和SE[0..5]相连。

钟表的特写

描述已自动生成

图6 七段数码管

（6）整体电路图

图示, 示意图

描述已自动生成

图7 整体电路图

**三、软件设计**

首先了解一下该实验的接线框图，如图8。

图示, 示意图

描述已自动生成

图8 接线框图

该实验的任务有两个，一是在数码管上显示8255-A，二是用8个LED灯实现流水灯效果。我们不妨分开来实现。

对于数码管上实现显示8255-A，我们的大致思路是只需要找到数码管对应的端口，将要输出的数据发送到对应的端口输出，然后适当设置选位信号，使得可以在正确的位置上显示正确的数字。因为这样只能输出一次，无法实现显示的目的，所以再将整体循环（达到人眼的视觉暂留，人视觉上表现为一直显示）便可达到目的。

对于实现8个开关控制LED灯实现流水灯效果，我们想分为三个步骤：第一个步骤是将初始的开关状态保存。因为根据指导书要求，一旦流水开始，LED灯次序不再理睬开关重新变化。所以第一步是保存初始开关状态，读取初始开关状态到AL；第二步是实现本次的LED灯显示，因为如果要让流水明显，需要给彩灯加延时。我们的想法是在彩灯显示后加入循环指令，其设定的循环次数也就是延时时长可根据需要改变。所以这一步只需要输出AL中保存的顺序，并且增加延时；第三步是实现流水。这一步首先是要储存现在的状态，然后右移一位（即右移一个灯），并且把新状态存入，然后进行下一个循环。由于本实验不需要暂停或者结束循环，所以设置的循环可以不设跳出条件，达到一直进行的目的。

程序流程图如图9。

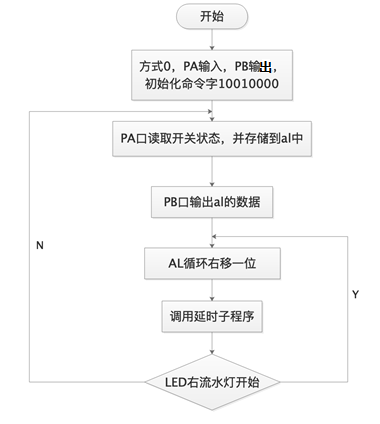


图9 程序流程图

经过上面的分析之后，我们就可以编写实现上述功能的代码了，具体代码如下：

CODE SEGMENT PUBLIC 'CODE'

ASSUME CS:CODE

START:

MOV DX,206H ;8255命令口206H

MOV AL,10010000B ;初始化，PA口输入，PB口输出

OUT DX,AL

MOV DX,200H ;PA口

IN AL,DX ;读PA口至寄存器AL

MOV BL,AL

ROL BL,1 ;BL初始化，不带进位循环左移一位

NEXT: ;右流水

ROR BL,1 ;不带进位循环右移一位

MOV AL,BL

MOV DX,202H ;PB口

OUT DX,AL ;输出到PB口（LED灯）

MOV CX,2FFFH ;LP循环显示8255-A，LED流水延时用CX控制

LP:

;使数码管显示8

MOV DX,402H

MOV AL,11111111B

OUT DX,AL

MOV DX,400H

MOV AL,7FH

OUT DX,AL

MOV DX,402H

MOV AL,11011111B

OUT DX,AL

;使数码管显示2

MOV DX,402H

MOV AL,11111111B

OUT DX,AL

MOV DX,400H

MOV AL,5BH

OUT DX,AL

MOV DX,402H

MOV AL,11101111B

OUT DX,AL

;使数码管显示5

MOV DX,402H

MOV AL,11111111B

OUT DX,AL

MOV DX,400H

MOV AL,6DH

OUT DX,AL

MOV DX,402H

MOV AL,11110111B

OUT DX,AL

;使数码管显示5

MOV DX,402H

MOV AL,11111111B

OUT DX,AL

MOV DX,400H

MOV AL,6DH

OUT DX,AL

MOV DX,402H

MOV AL,11111011B

OUT DX,AL

;使数码管显示-

MOV DX,402H

MOV AL,11111111B

OUT DX,AL

MOV DX,400H

MOV AL,40H

OUT DX,AL

MOV DX,402H

MOV AL,11111101B

OUT DX,AL

;使数码管显示A

MOV DX,402H

MOV AL,11111111B

OUT DX,AL

MOV DX,400H

MOV AL,77H

OUT DX,AL

MOV DX,402H

MOV AL,11111110B

OUT DX,AL

LOOP LP ;循环显示8255-A

JMP NEXT ;跳转NEXT

CODE ENDS

END START

**四、Proteus仿真验证**

在编写完代码并且正确连接电路图后，使用Proteus仿真。

打开开关K1、K3、K5，其余开关闭合，运行仿真后结果如图10。可以看到对应的发光二极管D1、D3、D5亮，其余灭；七段数码管显示“8255-A”。

图示, 示意图

描述已自动生成

图10 仿真1

经过一段时间，发光二极管D0、D2、D4亮，其余灭，如图11。

图示, 示意图

描述已自动生成

图11 仿真2

从上图的结果我们可以看到LED灯部分已经出现了流水灯的效果，并且数码管部分可以正常的显示“8255-A”字样，符合设计要求。

**五、心得体会**

本次实验是用了proteus软件进行了虚拟仿真，在已经完成过8255实验后，对于编程和电路连接有了一定的基础。对于汇编程序部分问题不大。这次实验主要遇到的问题在于proteus软件的使用以及对于数码管数据端和选位端的连接。在上网查阅了许多proteus软件的资料，并且通过调试指导老师给的例程后，对于软件使用有了初步的了解，也获得了数码管部分相关问题的提示，最终解决了问题。

在这次实验中，我加深了对课堂上学到的8255相关芯片的知识的理解，以及在仿真软件中的应用。也通过编写代码，从而思考了理论和实践当中的区别和联系，增强了自己的动手能力，也对相关的代码功能有了更加深刻的理解。

在汇编接口实验中，对8255芯片已经有了一定了解，并编写了能够实现流水灯功能的汇编程序，但是对于实际硬件的连接与内部原理的学习仍然不够深入。

在本次结合Proteus的研究型学习中，我更加深入地了解了8086指令系统下汇编程序的编写，熟悉了各个接口芯片与总线之间的连接关系以及数据传输关系。其中，从本次学习中的两大方面——8255流水灯和LED数码管中，我通过上网查阅资料、亲身动手实践，学习Proteus软件的基本操作，分别掌握了8255芯片的初始化与使用方法、外设与总线间I/O接口的设计方法，也对于汇编程序运行过程中数据在系统中流通的过程也有了更深刻的体会。