



计算机接口与技术 课程设计报告

题 目： 计算机接口与技术课程设计报告
院 系： 计算机科学与技术
班 级： 191181
姓 名： 常文瀚
学 号： 20181001095
签 字： _____

目录

一、设计目标.....1

二、设计所用仪器和器件.....1

三、设计内容.....3

四、设计过程.....3

 4.1 设计说明书.....3

 4.2 使用说明书..... 14

五、分析与讨论.....14

六、总结.....15

七、参考文献.....15

一、 设计目标

1. 设计一个洗衣机控制系统，用步进电机的转速和持续时间模拟不同的洗涤方式（标准，轻柔，强力等），可以用蜂鸣器或发光二极管来表示洗衣机的状态（洗涤、脱水、完成等）。
2. 使用接口芯片：8086、74HC373、74HC138、8253A、8255A、ULN2003A
3. 实现功能：
 - （1） 用步进电机的转速和持续时间模拟不同的洗涤方式（标准，轻柔，强力等）
 - （2） 可以用蜂鸣器或发光二极管来表示洗衣机的状态（洗涤、脱水、完成等）

二、 设计所用仪器和器件

1. 8086

Intel 第一款 16 位微处理器

2. 74HC373（锁存器）

74HC373 是一款高速 CMOS 器件，74HC373 引脚兼容低功耗肖特基 TTL（LSTTL）系列。

74HC373 是八路 D 型锁存器，每个锁存器具有独立的 D 型输入，以及适用于面向总线的应用的三态输出。所有锁存器共用一个锁存使能（LE）端和一个输出使能（OE）端。

3. 74HC138（译码器）

74HC138 是一款高速 CMOS 器件，74HC138 引脚兼容低功耗肖特基 TTL（LSTTL）系列。

74HC138 译码器可接受 3 位二进制加权地址输入（A0，A1 和 A2），并当使能时，提供 8 个互斥的低有效输出（Y0 至 Y7）。

4. 8253A（计数器）

8253 内部有三个计数器，分别称为计数器 0、计数器 1 和计数器 2，他们的机构完全相同。每个计数器的输入和输出都决定于设置在控制寄存器中的控制字，互相之间工作完全独立。每个计数器通过三个引脚和外部联系，一个为时钟输入端 CLK，一个为门控信号输入端 GATE，另一个为输出端 OUT。

执行部件实际上是一个 16 位的减法计数器，它的起始值就是初值寄存器的值，而初始值寄存器的值是通过程序设置的。

5. 8255A（可编程外设接口电路-PPI）

8255A 芯片 Intel 8086/8088 系列的可编程外设接口电路（Programmable Peripheral Interface)简称 PPI，型号为 8255（改进型为 8255A 及 8255A-5），具有 24

条输入/输出引脚、可编程的通用并行输入/输出接口电路。它是一片使用单一+5V 电源的 40 脚双列直插式大规模集成电路。

Mode 0: 方式 0 为基本的输入 / 输出方式, 传送数据时不需要联络信号。A 口、B 口和 C 口 (或 C 高 4 位口及 C 低 4 位口) 均可独立设置成方式 0 输入口或方式 0 输出口。

工作方式 0 是 8255A 中各端口的基本输入/输出方式。它只完成简单的并行输入/输出操作, CPU 可从指定端口输入信息, 也可向指定端口输出信息, 如果三个端口均处于工作方式 0, 则可由工作方式控制字定义 16 种工作方式的组合。

Mode 1: 方式 1 为选通输入 / 输出方式, 即需要进行联络的输入 / 输出。A 组、B 组的 8 位口 (A 口或 B 口) 可被设置为方式 1 输入口或方式 1 输出口, 而这时要用相应 C 口的 3 根线作联络线。

工作方式 1 被称作选通输入/输出方式。在这种工作方式下, 数据输入/输出操作要在选通信号控制下完成。

Mode 2: 方式 2 为双向传送。该方式要使用 C 口的 5 根线作联络线。由于 C 口只有 8 根线, 因此只能有 A 组确定为使用方式 2。当 A 组被设置成方式 2 时, A 口被设置成双向端口, 即既可以输入数据, 也可以输出数据, C 口的 5 根线被指定为联络线。

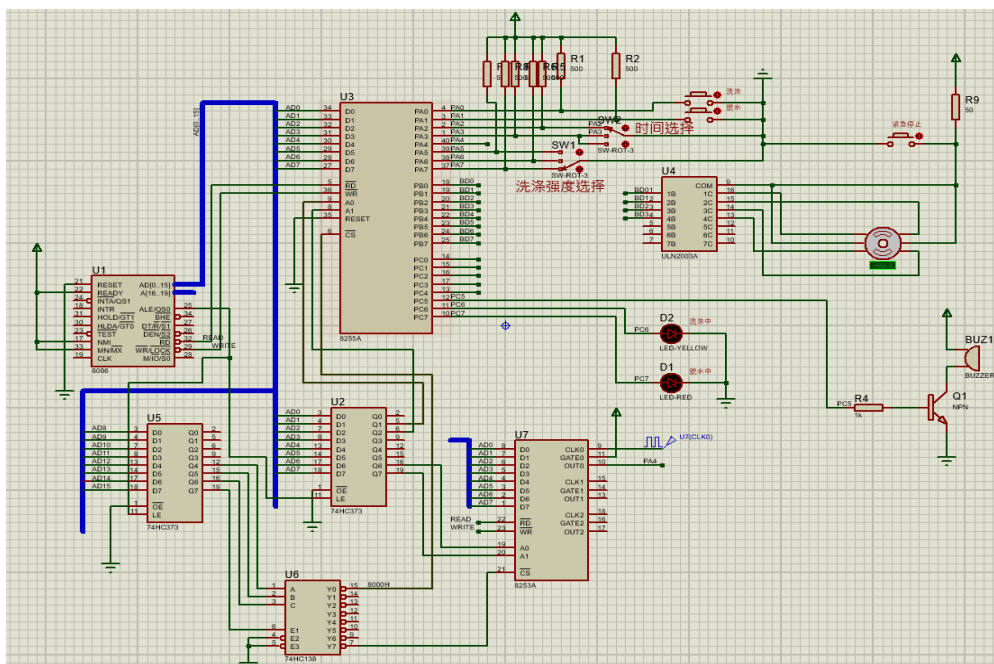
工作方式 2 被称作带选通的双向传送方式。8255A 中只允许端口 A 处于工作方式 2, 用来在两台处理机之间实现双向并行通信。其有关的控制信号由端口 C 提供, 并可向 CPU 发出中断请求信号。

6. ULN2003A (驱动器, 用于驱动步进马达)

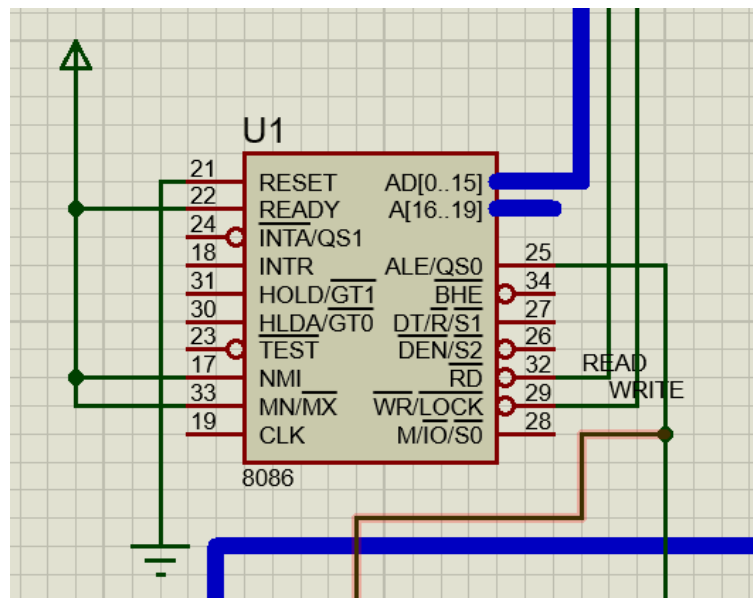
一种高压大电流的共发射极达林顿晶体管数组集成电路, 内含 7 组达林顿对管, 每组对管的电流容量是 500mA, 输出的电压最高 50V。该集成电路还集成了 7 只共阴极形式连接的续流二极管, 用于电感性负载的开关动作的电流续流。

达林顿晶体管 (或称达林顿对管), 借由两只双极型晶体管以直流耦合形式连接 (两只晶体管级联), 达到电流的二次放大, 可以很容易地达成大电流放大能力。最终的电流放大倍率, 约为两个双极型晶体管的电流放大倍率的乘积

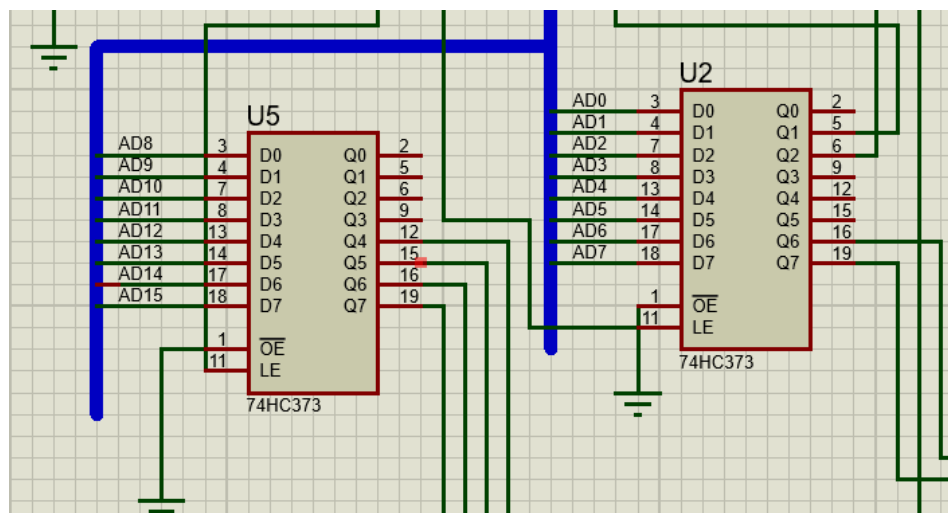
ULN2003 在电子电路里的主要特点是大电流容量和高电压输出。常作为驱动器使用, 每组达林顿对管可并联使用以达成更大的电流容量, 甚至可以几颗集成电路芯片堆叠并联使用。步进马达一般需要驱动器件有大电流输出的能力, 而微控制器等小电流的集成电路是不能直接驱动马达的, 因此可用作步进马达的驱动器。



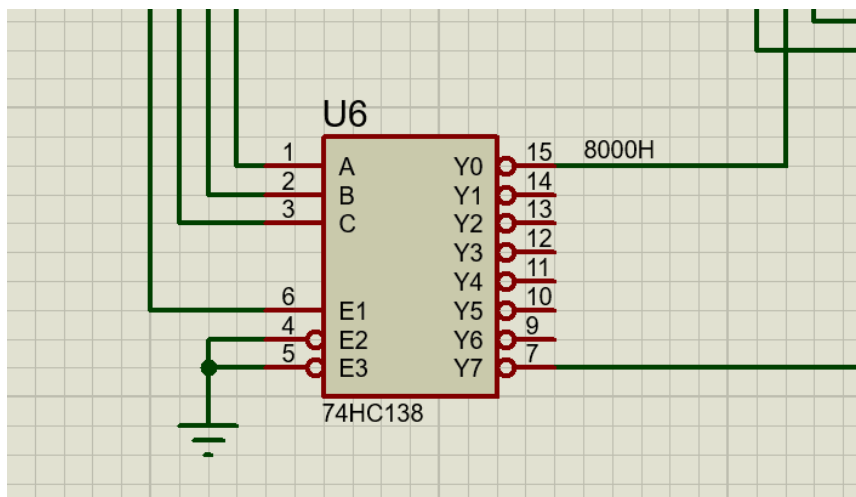
(2) 8086 部分



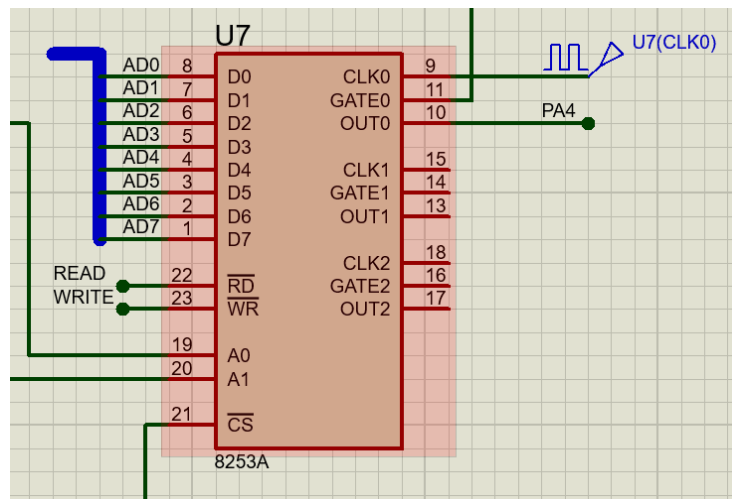
(3) 74HC373 (锁存器)



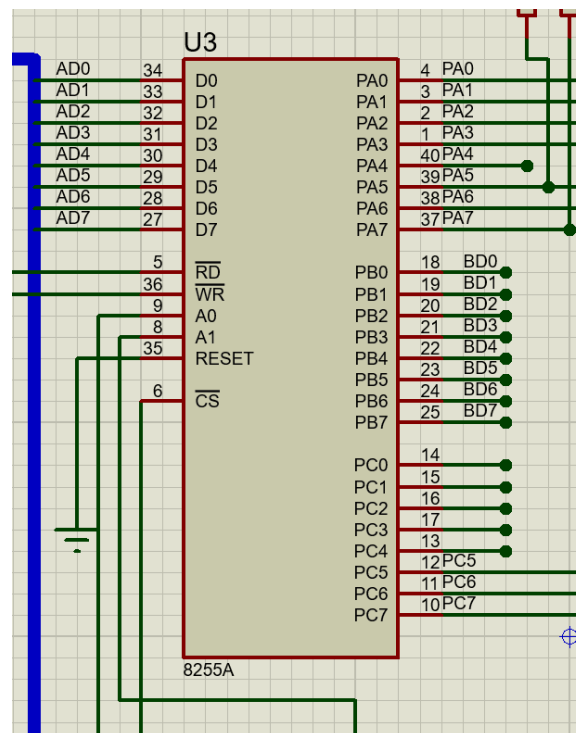
(4) 74HC138 (译码器)



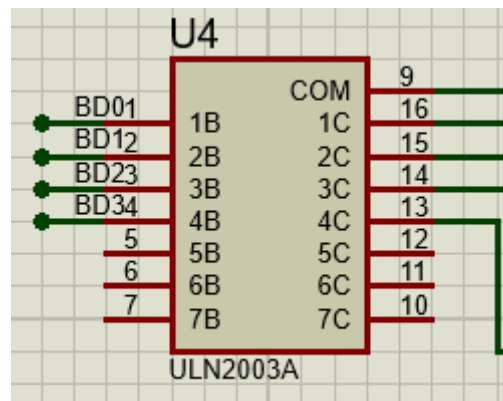
(5) 8253A



(6) 8255A



(7) ULN2003A



3. 软件代码

```
#define BASE8255 8000H      //设定值

#define PA BASE8255+00H*2  //PA 口

#define PB BASE8255+01H*2  //PB 口

#define PC BASE8255+02H*2  //PC 口

#define COM8255 BASE8255+03H*2  //工作方式

#define TIMEPORT 0F0C0H    //8253 写控制字端口

#define TIME0 0F000H       //8253 0 号计数器


char xiangxu[8]={02H, 06H, 04H, 0CH, 08H, 09H, 01H, 03H};


void outp(unsigned int addr, char data) {

    __asm {

        mov dx, addr //将 addr 移入 dx

        mov al, data

        out dx, al //将 al 的值写入 dx

    }

}


// Read a byte from the specified I/O port

char inp(unsigned int addr) {

    char result;

    __asm {

        mov dx, addr

        in al, dx

        mov result, al

    }

    return result;

}
```



```
void delay(int n)
{
    for(int i=0;i<n;i++)
    {
    }
}
```

//正向旋转的函数

```
void clockWise(int n)
{
    for(int i=0;i<8;i++)
    {
        outp(PB,xiangxu[i]);
        delay(n);
    }
}
```

//逆向旋转函数

```
void antiClockWise(int n)
{
    for(int i=7;i>=0;i--)
    {
        outp(PB,xiangxu[i]);
        delay(n);
    }
}
```

//用这个函数读取 PA4 的状态，确定计时是否结束

```
int timeFinished()
{
```

```

char tmp=inp(PA);

if((tmp&0b00010000)==0b00010000)    //PA4 为 1，则计时结束

    return 1;

return 0;

}

```

//设置 8253 0 的时间值，以秒为单位

```

void setTime0(int n)
{
    outp(TIMEPORT, 0b00010001);

    outp(TIME0, n);

    //outp(TIME0, 0b00000000);
}

```

//首洗涤的旋转

```

void washFun(int washLevel, int time)
{

```

//需要正转反转结合

```

    if(washLevel==1)
    {
        //for(int i=0;i<7;i++)

        setTime0(time);

        while(!timeFinished())
        {
            for(int j=0;j<2;j++)

                clockWise(1000);

            delay(1000);

            for(int j=0;j<2;j++)

                antiClockWise(1000);

```

```

        delay(1000);
    }
}
else if(washLevel==2)
{
    //for(int i=0;i<5;i++)
    setTime0(time);
    while(!timeFinished())
    {
        for(int j=0;j<2;j++)
            clockWise(7000);
        delay(7000);
        for(int j=0;j<2;j++)
            antiClockWise(7000);
        delay(7000);
    }
}
else if(washLevel==3)
{
    //for(int i=0;i<3;i++)
    setTime0(time);
    while(!timeFinished())
    {
        for(int j=0;j<2;j++)
            clockWise(20000);
        delay(20000);
        for(int j=0;j<2;j++)
            antiClockWise(20000);
        delay(20000);
    }
}

```

```
    }  
}
```

//脱水用的函数，朝着一个方向快速转动即可

```
void dryOut(int time)  
{  
    setTime0(time);  
    while(!timeFinished())  
        for(int i=0;i<8;i++)  
        {  
            outp(PB,xiangxu[i]);  
            delay(1500);  
        }  
}
```

```
void main(void)  
{  
    //程序一直运行  
    while(1)  
    {  
        char tmp, judge;  
        int i=0;  
        //设置工作方式：1 00 1 0 0 0 0（特征位=1，A组0方式=00，PA输入=1，PC上  
        半没用到=0，B组0方式=0，PB输出=0，PC下半没用到=0）  
        outp(COM8255,090H); //090H = 1001 0000  
        //读 PA 口  
        while(1)  
        {  
            tmp=inp(PA);  
            //if(tmp&0b11100000==0b11100000)
```

```

if((tmp&0b000000011)==0b000000011)
{
    //读取 PA 口，如果前三个都是一代表还没有按下一个键
}
else
    break;
}

//judge 用于判断工作状态，末两位 10 为洗涤，01 为脱水
judge=(tmp&0b000000011);

int washLevel=1; //档位
//根据输入确定档位
//if((tmp|0b01111111)==0b01111111)
if((tmp&0b10000000)==0b00000000) //PA7 为 0
{
    washLevel=1;
    printf("1")
}
//else if((tmp|0b10111111)==0b10111111)
else if((tmp&0b01000000)==0b00000000) //PA6 为 0
{
    washLevel=2;
    printf("2")
}
else if((tmp&0b00100000)==0b00000000) //PA5 为 0
{
    washLevel=3;
    printf("3")
}

```

```

//确定时间

int time=0;

if((tmp&0b00001000)==0b00000000) //PA3 为 1
{
    time=8;
}

else if((tmp&0b00000100)==0b00000000) //PA2 为 1
{
    time=4;
}

int n=2;

while (1){
n--;
if(n==0) break;
if(judge==0b00000010)
{ //洗涤
    //设置洗涤状态灯亮
    outp(PC, 0b01000000); //PC6 为 1, 洗涤灯亮
    washFun(washLevel, time);
    outp(PC, 0b10000000); //PC7 为 1, 脱水灯亮
    dryOut(4);
    outp(PC, 0b00000000); //灯全灭
}

else if(judge==0b00000001)
{ //脱水
    outp(PC, 0b10000000); //PC7 为 1, 脱水灯亮
    dryOut(time);
    outp(PC, 0b00000000); //灯全灭
}

delay(1000);

```


4.2 使用说明书

(1) 洗衣强度

强度开关	时间开关	强度等级	延迟时间	洗衣时间	脱水时间
PA5	PA2	弱	20000	12s	4s
PA6	PA2	中	7000	4s	4s
PA7	PA2	强	1000	4s	4s

表 1 不同强度洗衣与 PA2 时间开关搭配说明表

强度开关	时间开关	强度等级	延迟时间	洗衣时间	脱水时间
PA5	PA3	弱	20000	12s	4s
PA6	PA3	中	7000	8s	4s
PA7	PA3	强	1000	8s	4s

表 2 不同强度洗衣与 PA3 时间开关搭配说明表

(2) 脱水强度

时间开关	脱水时间
PA2	4s
PA3	8s

表 3 不同时间开关接通后对应的脱水时间

五、 分析与讨论

(1) 优点

该程序电路连接清晰，逻辑清楚，鲁棒性强，可以正常工作，并且具有类似现实生活中的洗衣机那样具有紧急停止程序，可以让操作者在遇到紧急情况时停止洗衣机的运行，整个程序较为真实的模拟了洗衣机系统。

(2) 缺点

该作品主要缺点是代码较为混乱，但是可以进行重新整理。

其次，对所有芯片引脚的主要功能并没有了解完全。

对两个单刀三置开关的功能效果没有描述清晰，可能会给用户在使用时造成麻烦，可以预设一个显示器显示出洗衣的具体时间与脱水的具体时间。

六、 总结

课程设计,使自己的动手实践与所学知识结合应用的能力有了很大的提高,让我认识到实践的重要性,但同时要实际应用过和当中,也发现理论知识的重要性,没有理论知识的支撑,是不能很好的完成本次课程设的。在设计实验的过程中,不断的让理论与实践结合,讲课堂上所学的知识转化为在实践中所需的过程,这个过程对我学习很宝贵。

本次设计步进电机转动控制,通过我掌握了的微机原理与接口技术知识,查资料、百度、请教同学老师等等。通过利用 8255A 芯片来控制洗衣机系统的课程设计,为以后的学习实践打下了基础,并且很好的消化了所学知识的难点、问题。

在设计过程当中也发现了自己经验的不足。首先是对试验箱的认识和应用,很多模块的使用存在误解。然后就是在编程序方面的,有好多程序只是一知半懂,不能熟练地操作,需要提示和看书,需要大量的练习,为以后的学习指明方向。

感谢老师提供给我们实践课本知识的机会,并且认真地陪同学生一起设计课程设计,奠基了以后自主研究设计项目的能力。

七、 参考文献

- [1] 谢春祥,陈龙. 基于 Proteus 的 8086 和 8255A 接口实验仿真[J]. 蚌埠学院学报, 2013, 2(4): 12-14.
- [2] 白莉媛,肖乐,钱进. Intel 8255A 微机芯片的应用[J]. 光盘技术, 2006 (6): 50-51.
- [3] 温阳东,鲍远慧,林勇. 微机原理及应用课程中 8255A 的教学方法探讨[J]. 计算机时代, 2013 (7): 68-70.
- [4] 赖扬川. 基于 8255A 温度与压力检测系统的设计[J]. 福建电脑, 2009 (3): 136-137.
- [5] 黄志文,邓彬伟. 串行 A/D 与 8255A 接口的设计与实现[J]. 黄石理工學院學報 (人文社科版), 2007, 23(3): 34-37.
- [6] Binwei H Z D. The Design and Implementation of The 8255A Access SPI A/D [J][J]. Journal of Huangshi Institute of Technology, 2007, 3.