

实验一 LED灯实验

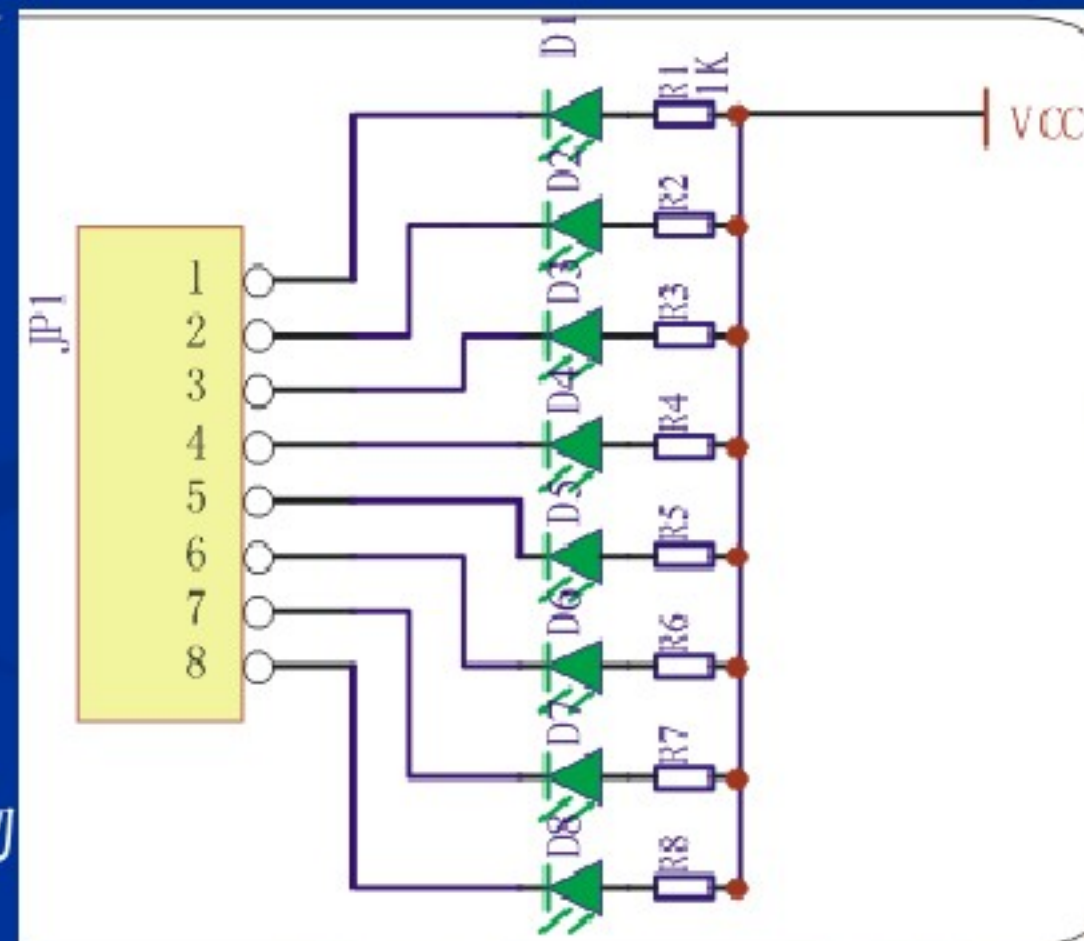
一、LED灯相关知识

LED Light Emitting Diode,即发光二极管,是一种半导体固体发光器件,它是利用固体半导体芯片作为发光材料,当两端加上正向电压,半导体中的载流子发生复合引起光子发射而产生光。LED可以直接发出红、黄、蓝、绿、青、橙、紫、白色的光。

LED灯应用有以下几个特点:

- 1.多变幻: LED光源可利用LED红、绿、蓝三基色原理,在计算机技术控制下实现色彩和图案的多变化,是一种可随意控制的“动态光源”。
- 2.寿命长: LED光源无灯丝、工作电压低,使用寿命可达5万到10万小时,也就是5年到10年时间。
- 3.利环保: 生产中无有害元素、使用中不发出有害物质、无辐射。
- 4.高节能: LED耗电相当低,一般来说LED的工作电压是2-3.6V。工作电流是20-30mA。这就是说: 它消耗的电能不超过0.1W。

二、开发试验的LED原理图



三、排线连接方法

看视频图像

四、相关的知识点

1、启动文件STARTUP.A51：它包含目标板启动代码，在每个工程中加入这个文件。只要复位，该文件则立即执行，其功能包括：1.定义内部RAM大小、外部RAM小、可重入堆栈位置。2.清除内部、外部或者以此页为单元的外部存储器。3.按存储模式初使化重入堆栈及堆栈指针。4.初始化8051硬件堆栈指针。5.向main()函数交控制权

2、头文件 #include<reg51.h>定义了51的所有寄存器,<>尖括号表示在安装目录下找reg51.h文件，""双引号表示在当前工程目录下找reg51.h文件。

3、子函数:C语言的执行从main函数开始，每个函数体是相互独立，被 main函数调用的函数都是子函数，子函数与子函数也可以相互调用

4、for循环格式： for(表达式 1;表达式 2;表达式 3) { 语句; }

5、16进制数，以0x开头的表示16进制，如： 0xff 0x12 0x00.

8进制数，用数字0开头的数是八进制 如： 023表示八进制的 23

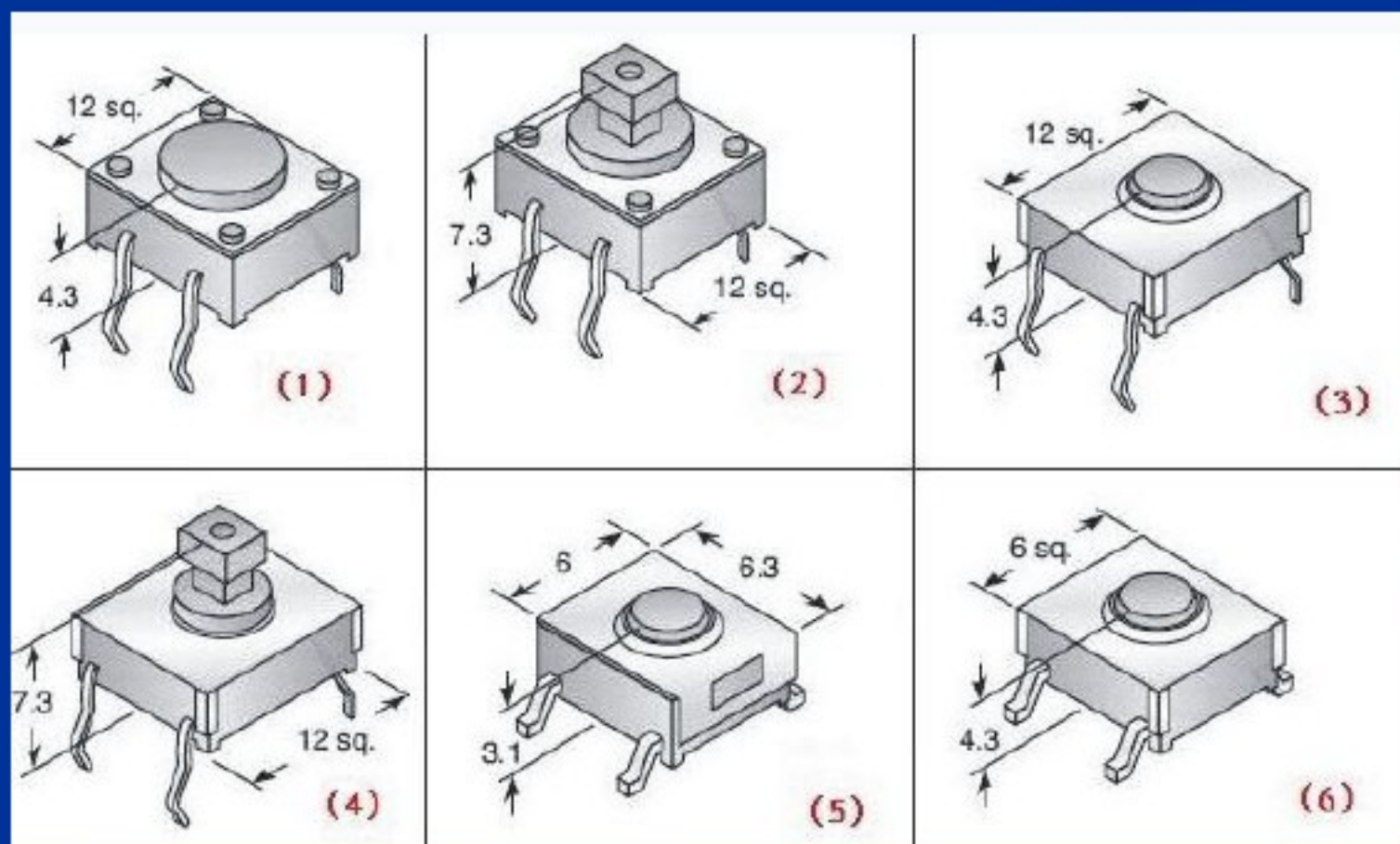
10进制数，日常生活中用到的数字，如： 100， 200， 30 等

五、程序示例

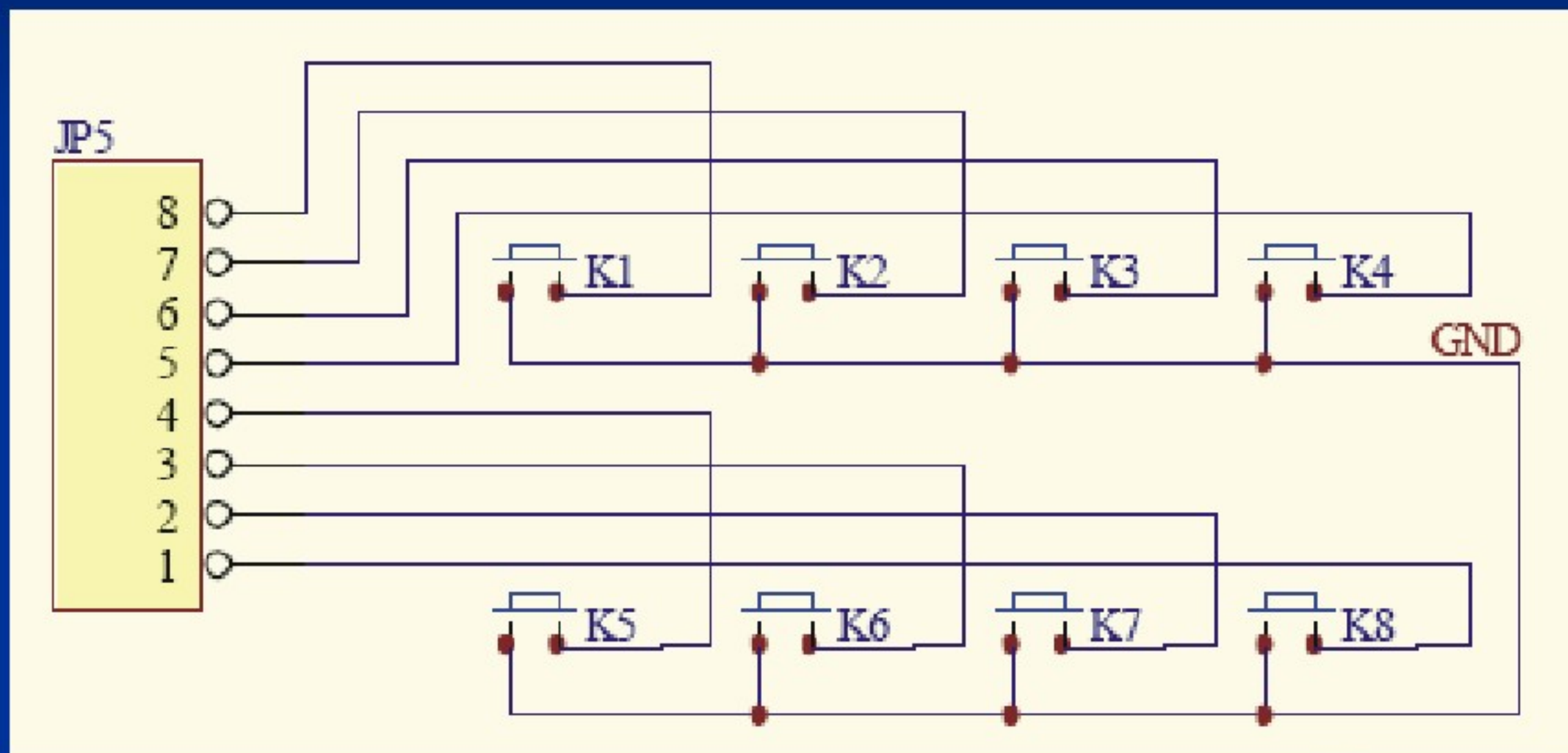
实验二 按键实验

轻触开关是一种电子开关，使用时，轻轻按开关按钮就可使开关接通，当松开手时，开关断开，其内部结构是靠金属弹片受力弹动来实现通断。轻触开关由于体积小重量轻在家用电器方面得到广泛的应用，如：影音产品、数码产品、遥控器、通讯产品、家用电器、安防产品、玩具、电脑产品、健身器材等等。但轻触开关也有它不足的地方，频繁的按动会使金属弹片疲劳失去弹性而失效。因此现在很多电器的按钮都使用导电橡胶或锅仔开关，比如电脑键盘，遥控器，手机等。

关于五脚轻触开关的脚位接法：两个引脚为一组，四角是为了焊接的更加稳固，第五个引脚是为接地线用的。



二、轻触按钮开关在开发仪上应用原理图



三、排线连接方法

看视频图像

四、知识点

- **1.intrins.h: _nop_函数**在此头文件中，此函数是空指令函数，相当汇编NOP指令。
- **2.Sbit 关键字:** 是Keil C增加的关键字，用来定义位变量，它有三种用法：
 - 1. sbit 位变量名 = 地址值 例如: sbit AC = 0xD6
 - 2.sbit 位变量名 = 寄存器名称^寄存器某位的序号 例如:sbit K1 = P0^0
 - 3.Sbit 位变量名 = 寄存器地址^寄存器某位的序号 例如:sbit K2 = 0x80^1
- **3.While循环语句:**
 - while语句用来“当型”循环结构，它的格式：`while(表达式) { 语句; }`当表达式为“真”或“1”时，循环执行while后面{ }内的语句，常称循环体，当为“假”或“0”时，不执行循环体或者退出循环体语句。
- **4.If条件判断选择语句:**
 - if语句是一个条件判断选择语句。这里介绍2种用法。它的格式：
 - ①if(表达式) { 语句; } //表达式为“真”或“1”则执行语句，为“假”或“0”则
//执行 语句后面的语句
 - if(表达式) 语句1; //表达式为真或1时，则执行语句1.
 - else 语句2; //表达式为假或0时，则执行语句2.

5.逻辑运算符:C语言提供了3种逻辑运算符,

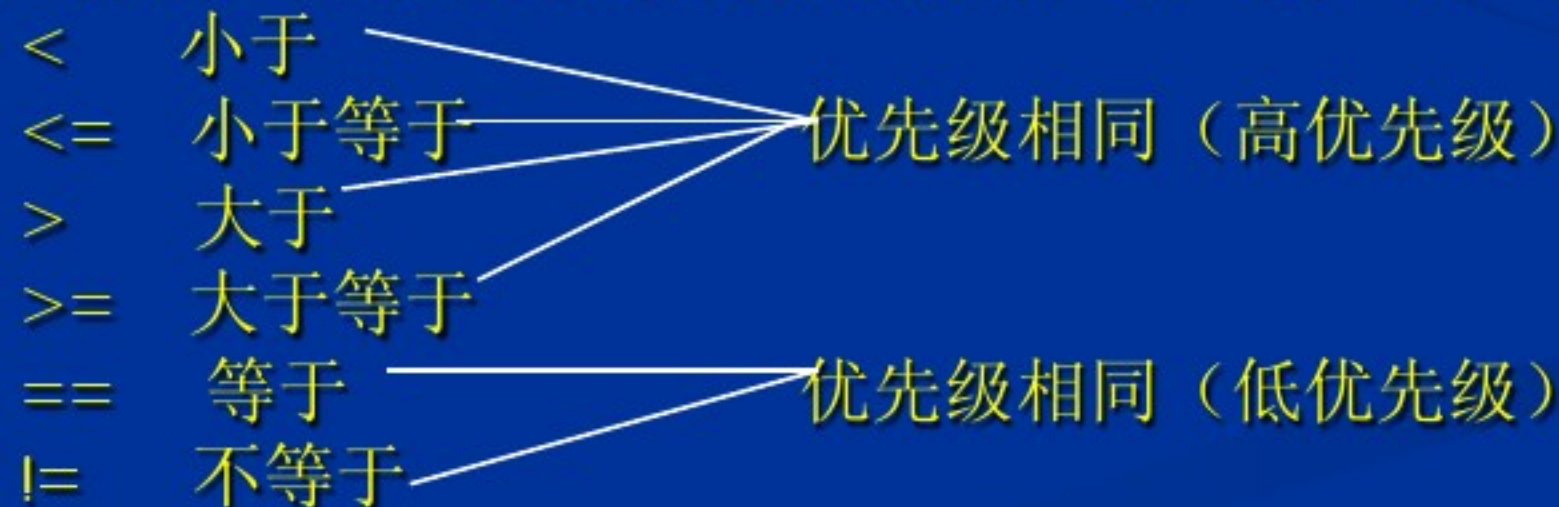
1. && 逻辑与 (双目运算)
2. || 逻辑或 (双目运算)
3. ! 逻辑非 (单目运算)

6.位逻辑运算符:大家不要把逻辑运算符和位逻辑运算符搞混淆,他们是有区别的,位逻辑运算符是以字节中的每个位分别进行逻辑处理。逻辑运算以位为单位和以字节为单位进行逻辑处理

1. & 位逻辑与 (双目运算) 相当汇编ANL
2. | 位逻辑或 (双目运算) 相当汇编ORL
3. ~ 位逻辑非 (单目运算) 相当汇编CPL

关于!与~的区别: $!2 = 0$, $\sim 2 = 0xfd$ (用8位存储) $\sim 0000\ 0010 = 1111\ 1101$

7.关系运算符:关系表达式的值只有2种,即“真”和“假”也就是“1”和“0”



实验三 继电器和蜂鸣器实验

继电器(英文名: Relay): 可以理解为继承控制, 它是一种电子控制器件。通常应用于自动控制电路中, 它实际上是用较小的电流去控制较大电流的一种自动开关。故在电路中起着自动调节、安全保护、转换电路等作用。

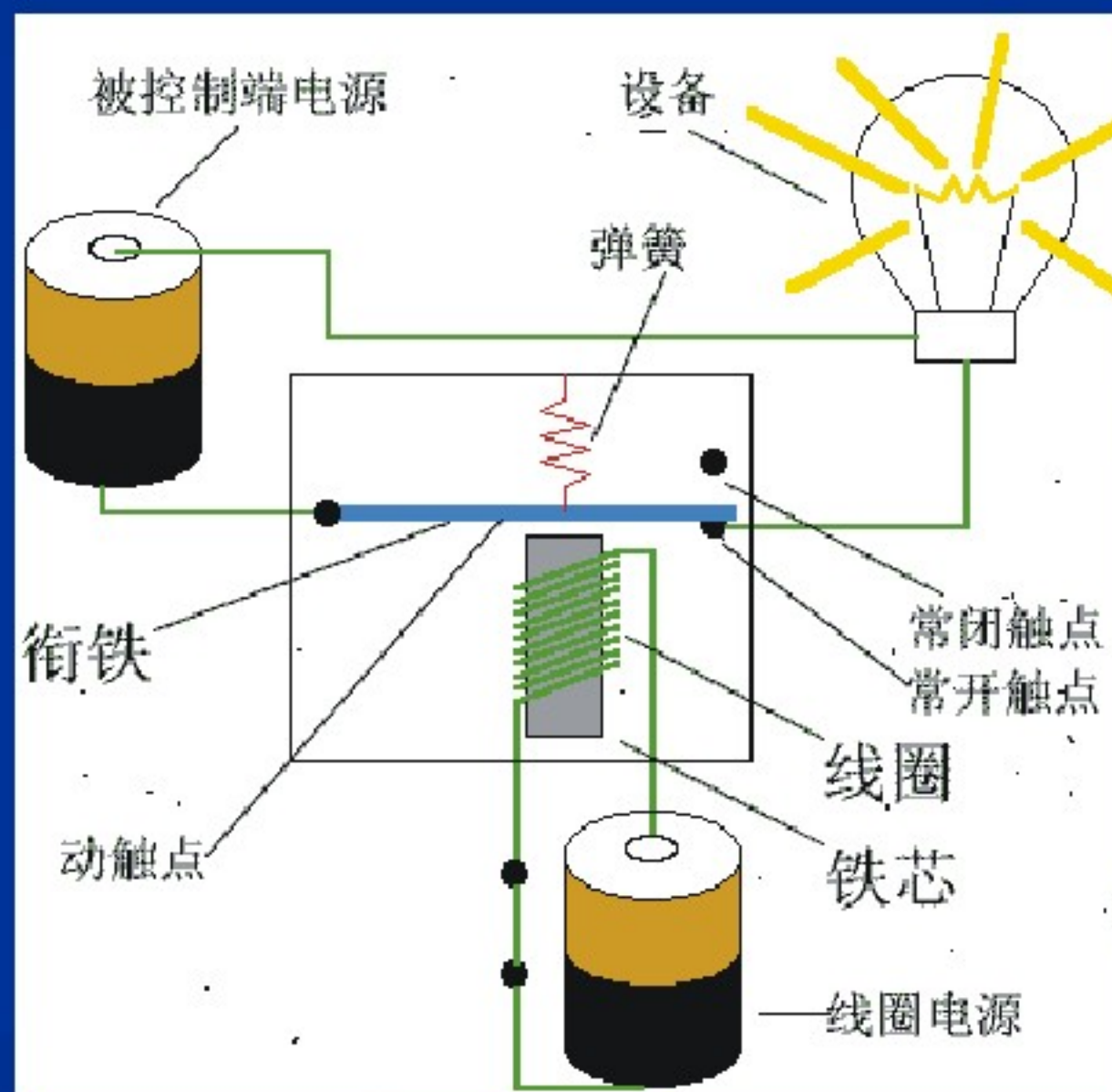
蜂鸣器有2种:1.压电式蜂鸣器 2.电磁式蜂鸣器

1. 压电式蜂鸣器主要由多谐振荡器、压电蜂鸣片、阻抗匹配器及共鸣箱、外壳等组成。

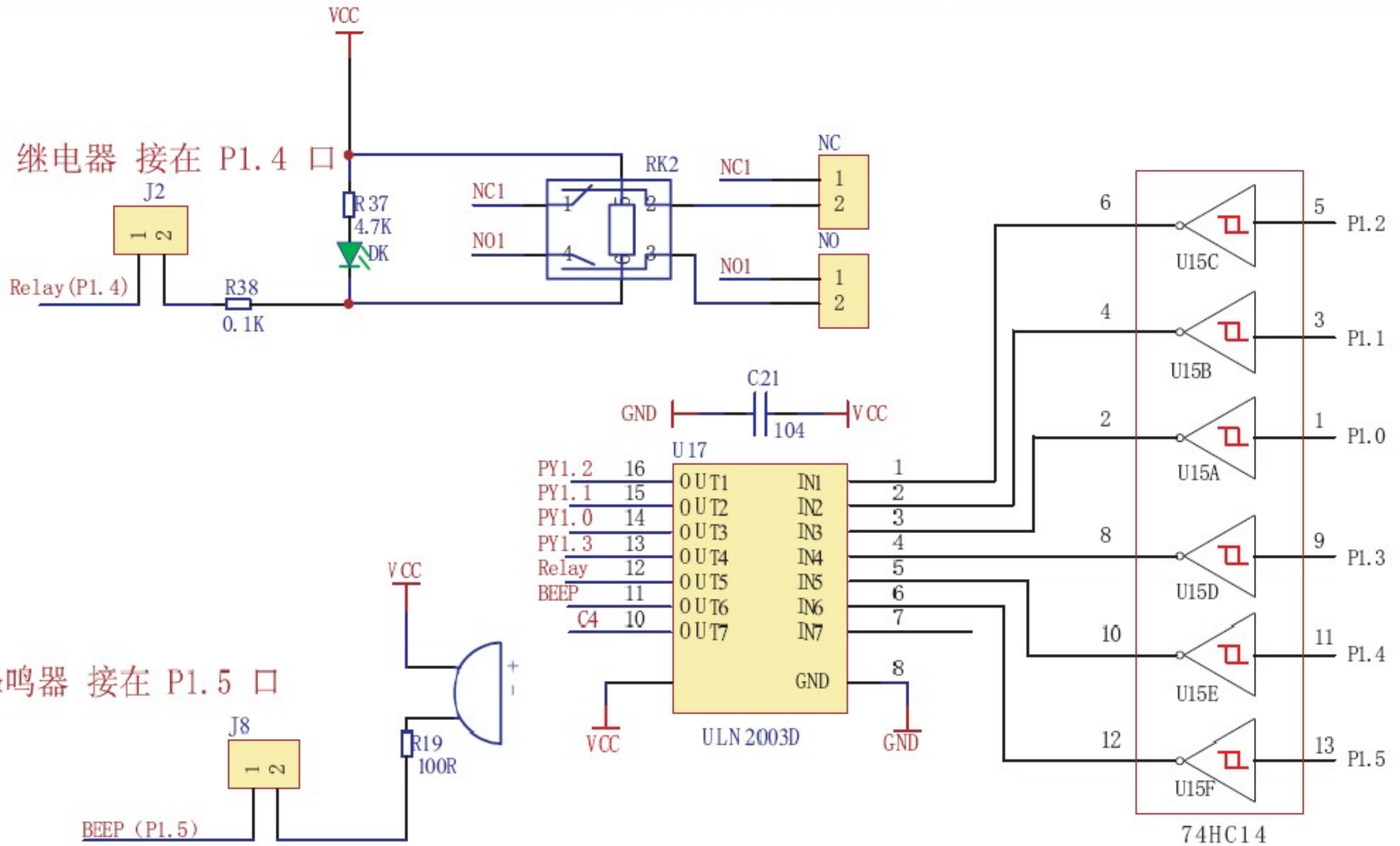
2. 电磁式蜂鸣器由振荡器、电磁线圈、磁铁、振动膜片及外壳等组成。接通电源后, 振荡器产生的音频信号电流通过电磁线圈, 使电磁线圈产生磁场。振动膜片在电磁线圈和磁铁的相互作用下, 周期性地振动发声。

关于有源蜂鸣器和无源蜂鸣器的区别: 这里的“源”不是指电源。而是指震荡源。也就是说, 有源蜂鸣器内部带震荡源, 所以只要一通电就会叫。无源蜂鸣器没有震荡源需要驱动的音频电流才能发声。

继电器模拟示意图



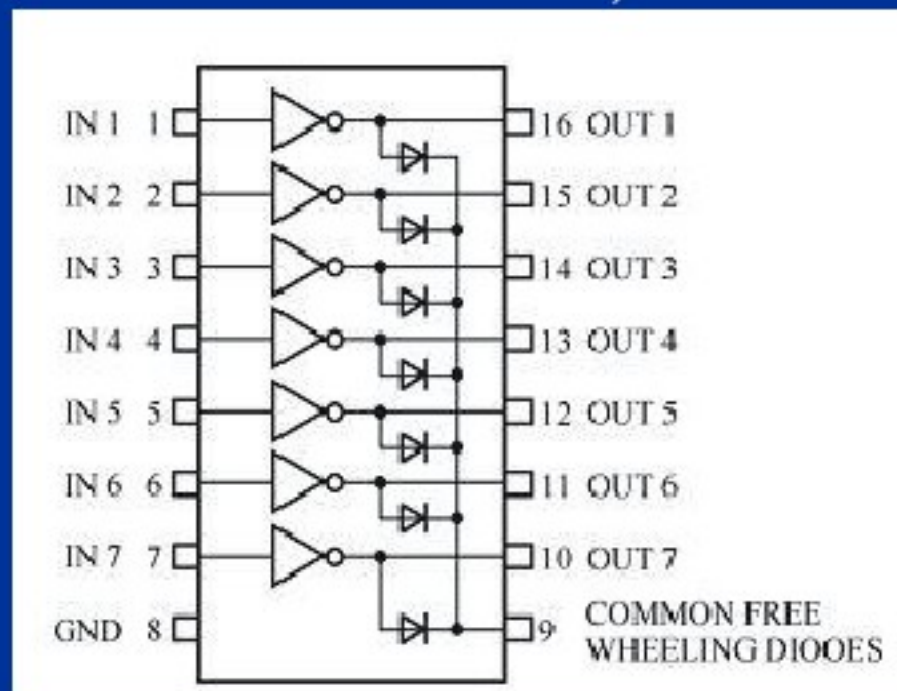
HC6800型号的原理图



ULN2003 是高耐压、大电流达林顿阵列，由七个硅NPN 达林顿管组成。ULN2003 的每一对达林顿都串联一个2.7K的基极电阻,在5V 的工作电压下它能与TTL 和CMOS 电路直接相连，可以直接处理原先需要标准逻辑缓冲器来处理的数据。

ULN2003 工作电压高，工作电流大，灌电流可达500mA，并且能够在关态时承受50V 的电压，输出还可以在高负载电流并行运行。

高压大电流达林顿晶体管阵列系列产品,具有电流增益高、工作电压高、温度范围宽、带负载能力强等特点,适应于各类要求高速大功率驱动的系统(更详细的技术参数参考ULN2003数据手册)



- 1: CPU脉冲输入端，端口对应一个信号输出端16。
- 2: CPU脉冲输入端。
- 3: CPU脉冲输入端。
- 4: CPU脉冲输入端。
- 5: CPU脉冲输入端。
- 6: CPU脉冲输入端。
- 7: CPU脉冲输入端。
- 8: 接地

9: 该脚是内部7个续流二极管负极的公共端，各二极管的正极分别接各达林顿管的集电极。用于感性负载时，该脚接负载电源正极，实现续流作用。如果该脚接地，实际上就是达林顿管的集电极对地接通。

实验四 译码器实验

译码器的相关知识:

译码器: 是一种具有“翻译”功能的逻辑电路,它是组合逻辑电路中的一个重要的器件,这种电路能将输入二进制代码的各种状态,按照其原意翻译成对应的输出信号。有一些译码器设有一个和多个使能控制输入端,又成为片选端,用来控制允许译码或禁止译码。

译码器可以分为: 变量译码和显示译码两类。

变量译码: 一般是一种较少输入变为较多输出的器件,一般分为 2^n 译码和8421BCD码译码两类。HC6800开发仪的译码器为变量译码器。

显示译码: 主要解决二进制数显示成对应的十、或十六进制数的转换功能,一般其可分为驱动LED和驱动LCD两类。

74138译码器: 是一种3线—8线译码器,三个输入端ABC共有8种状态组合(000—111),可译出8个输出信号Y0—Y7。这种译码器设有三个使能输入端,当G2A与G2B均为0,且G1为1时,译码器处于工作状态时,对应的输出为低电平。当译码器被禁止时,所有输出为高电平。

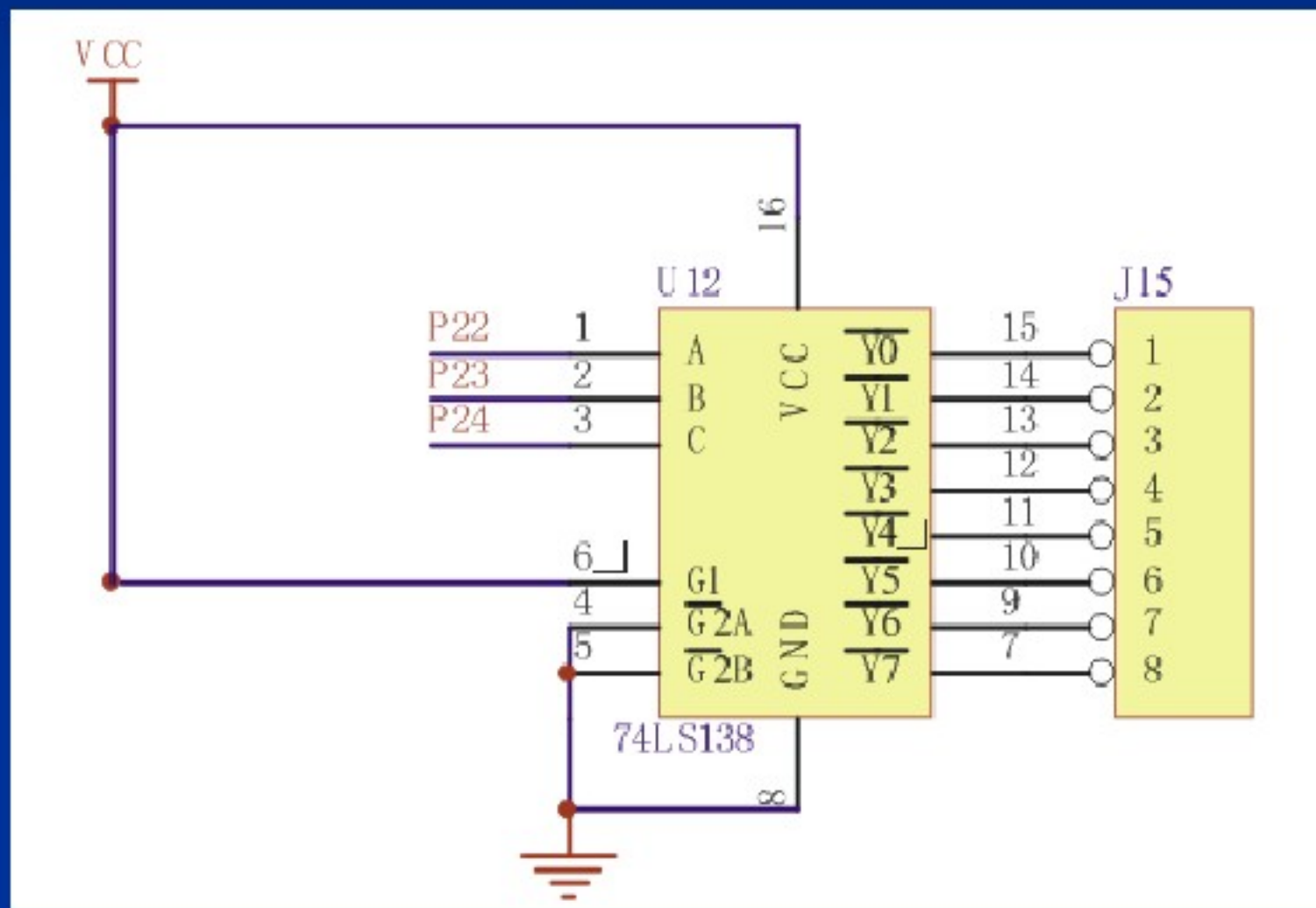
真值表

输入			输出											
G1	$\overline{G2A}$	$\overline{G2B}$	C	B	A	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0	
0	X	X	X	X	X	1	1	1	1	1	1	1	1	
X	1	X	X	X	X	1	1	1	1	1	1	1	1	
X	X	1	X	X	X	1	1	1	1	1	1	1	1	
1	0	0	0	0	0	1	1	1	1	1	1	1	0	
			0	0	1	1	1	1	1	1	1	0	1	
			0	1	0	1	1	1	1	1	0	1	1	
			0	1	1	1	1	1	1	0	1	1	1	
			1	0	0	1	0	1	1	0	1	1	1	1
			1	0	1	1	0	1	1	0	1	1	1	1
			1	1	0	1	1	1	0	1	1	1	1	1
			1	1	1	1	1	0	1	1	1	1	1	1

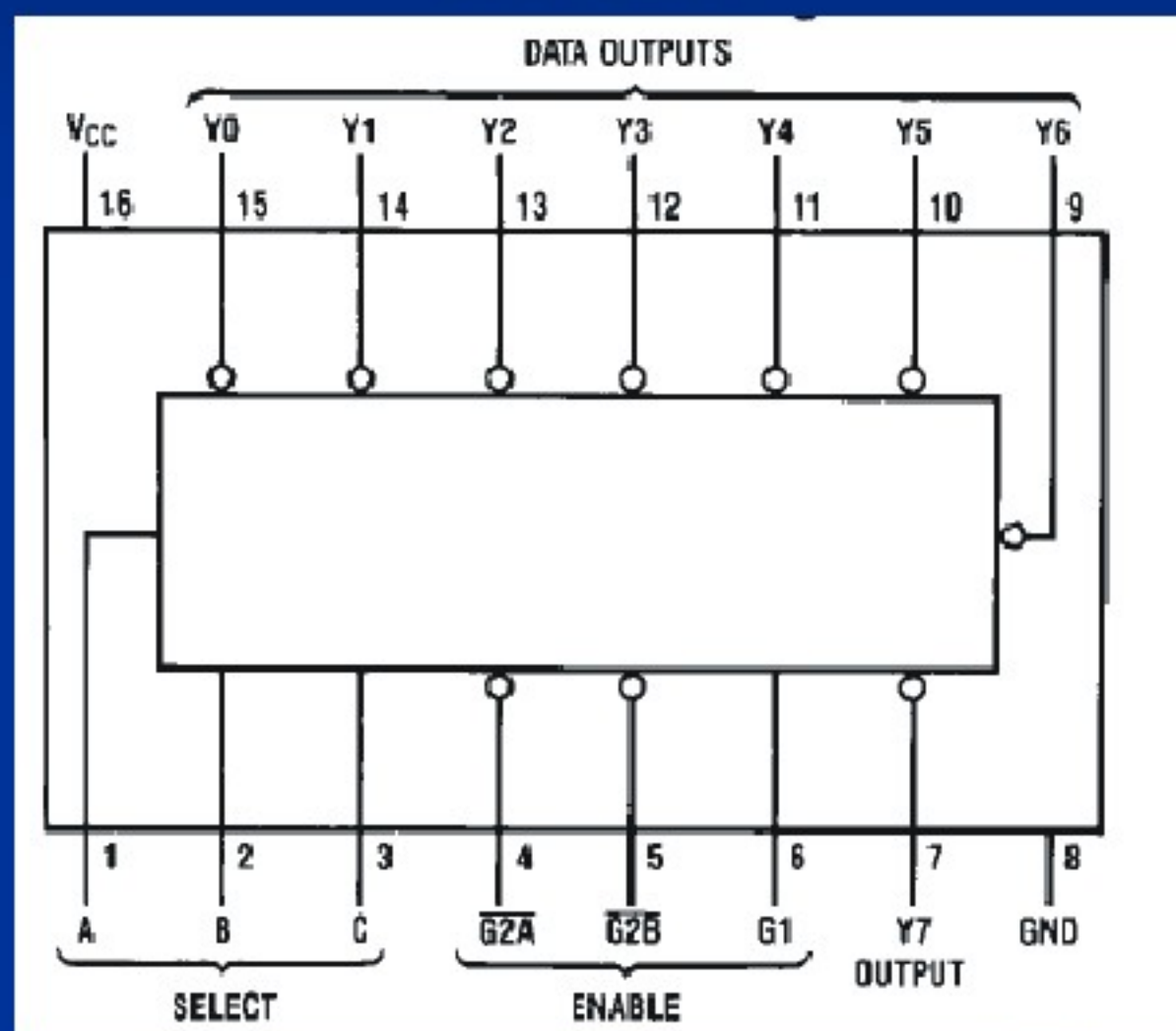
通过真值表可以看到74HC138的八个输出引脚，任何时刻要么全为高电平1，此时芯片处于不工作状态，要么只有一个为低电平0，其余7个输出引脚全为高电平1。如果出现两个输出引脚同时为0的情况，说明该芯片已经损坏。

原理图和连接逻辑图

原理图



连接逻辑图



当一个选通端（G1）为高电平，另两个选通端（ $\overline{G2A}$ 和 $\overline{G2B}$ ）为低电平时，可将地址端（A、B、C）的二进制编码在一个对应的输出端以低电平译出，利用G1、 $\overline{G2A}$ 和 $\overline{G2B}$ 可级联扩展成4线-16线译码器或5线-32线译码器

接线演示

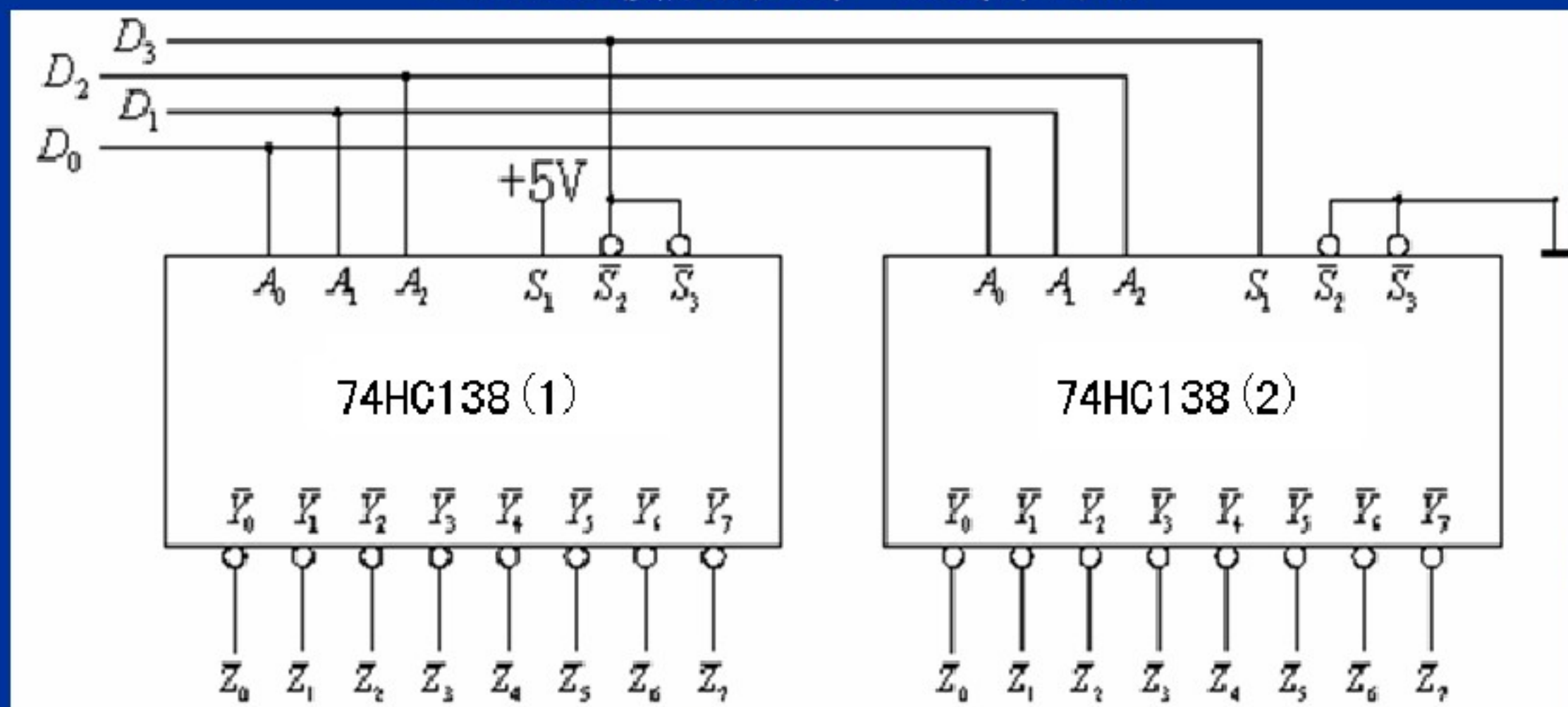
看视频演示

例程演示

译码器例程讲解

课后思考

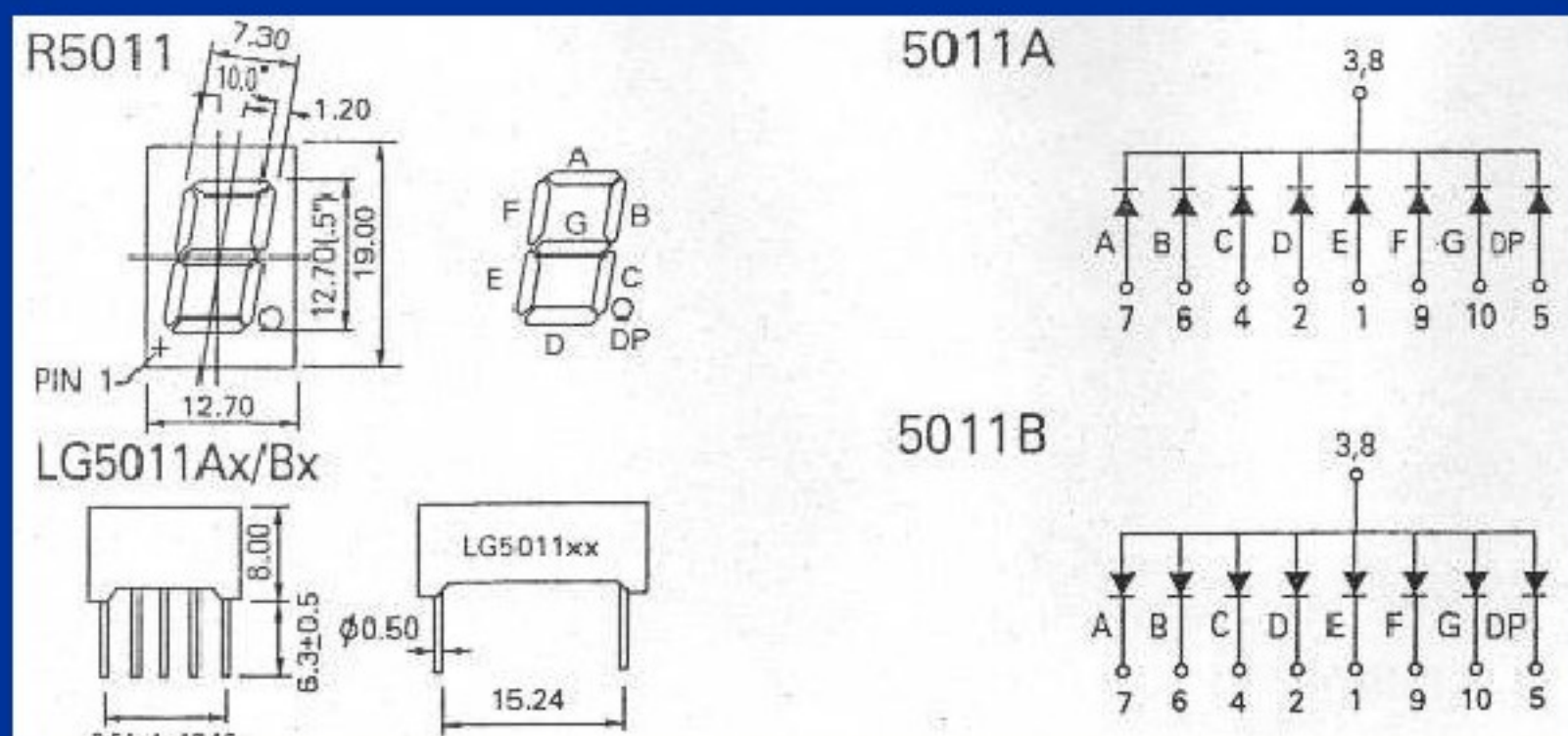
74HC138接成的4线—16线译码器



实验五 静态数码管实验

一、数码管相关知识：

LED数码管（LED Segment Displays）是由多个发光二极管封装在一起组成“8”字型的器件，引线已在内部连接完成，只需引出它们的各个笔划，公共电极。LED数码管常用的段数一般为7段，有的另加一个小数点。LED数码管根据LED的接法不同，分为共阴和共阳两类。

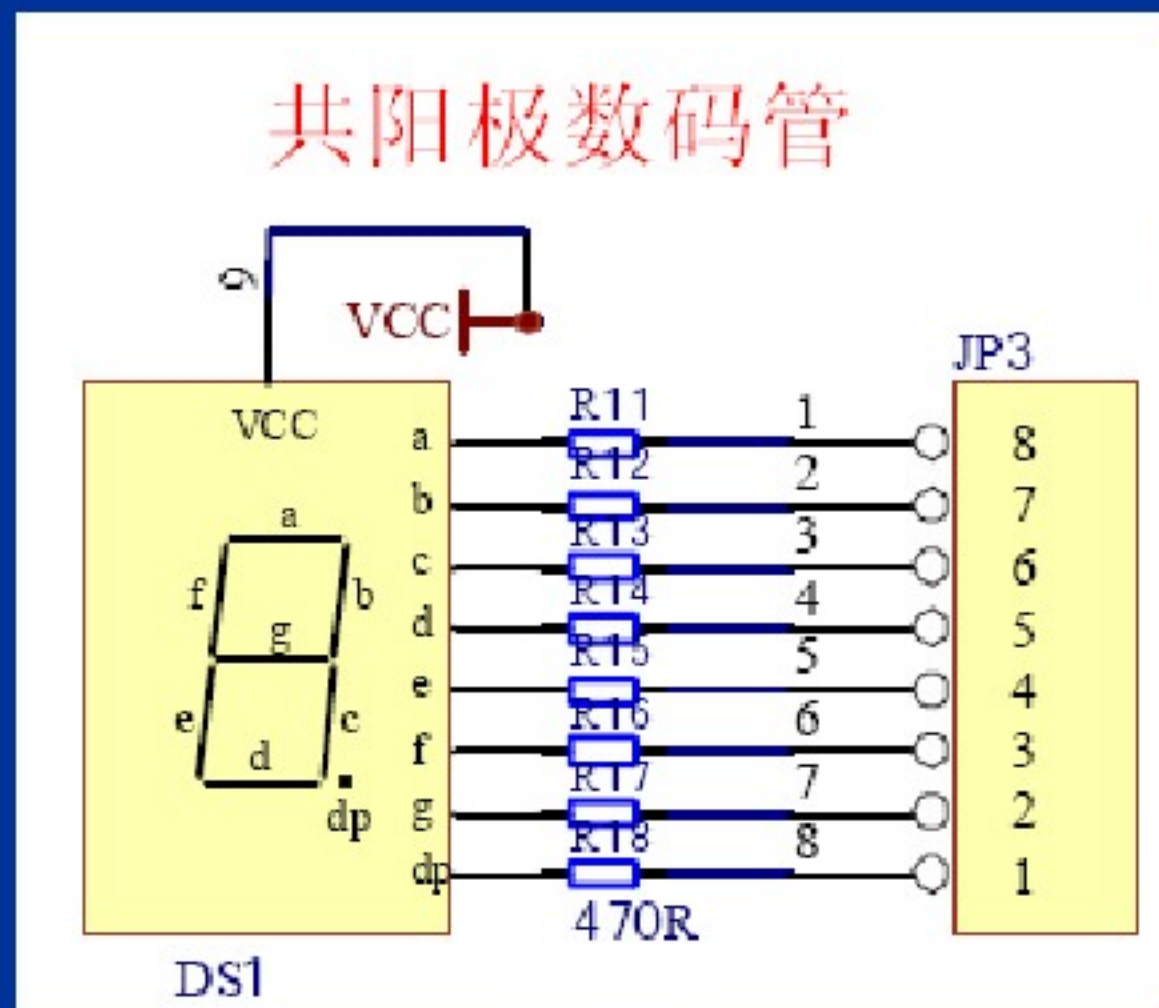


数码管使用的电流：静态时，推荐使用10-15mA；动态时，动态扫描时，平均电流为4-5mA，峰值电流可达到50-60mA

静态显示：静态显示就是静态驱动也称直流驱动。静态驱动是指每个数码管的每一个段码都由一个单片机的I/O端口进行驱动，或者使用如BCD码二-十进制译码器译码进行驱动。静态驱动的优点是编程简单，显示亮度高，缺点是占用I/O端口多，如驱动5个数码管静态显示则需要 $5 \times 8 = 40$ 根I/O端口来驱动，实际应用时必须增加译码驱动器进行驱动，增加了硬件电路的复杂性。

LED数码管的应用：LED数码管广泛用于仪表，时钟，车站，机器设备、家电等场合

二、原理图：



三、接线演示：

看视频演示

四、例程演示：

以《数码管（静态显示）》的C语言作为例程演示

关于数码管显示字模：

共阳极字模：

```
0 1 2 3 4 5 6 7 8 9 a b c d e f  
{~0x3F,~0x06,~0x5B,~0x4F,~0x66,~0x6D,~0x7D,~0x07,~0x7F,~0x6F,~0x77,~0x7C,~0x39,~0x5E,~0x79,~0x71};
```

```
{ 0xC0, 0xF9, 0xA4, 0xB0, 0x99, 0x92, 0x82 ,0xF8, 0x80, 0x90, 0x88, 0x83, 0xC6, 0xA1 0x86 ,0x8E};
```

共阴极字模：

```
0 1 2 3 4 5 6 7 8 9 a b c d e f  
{0x3F,0x06,0x5B,0x4F,0x66,0x6D,0x7D,0x07,0x7F,0x6F,0x77,0x7C,0x39,0x5E,0x79,0x71};
```


自增自减运算符

`++i` 使用*i*值之前, 使*i*的值加1, 再使用*i*的值

`i++` 使用完*i*值后, 再使*i*值加1

`i--` 使用完*i*值后, 再使*i*的值减1,

`--i` 使用*i*值之前使*i*的值减1, 再使用*i*的值

1. 对于普通独立的语句

`i++;` 等效于 `i=i+1;`

`++i;` 等效于 `i=i+1;`

2. 用于赋值语句

例如: `i = 5; j = ++i;` 执行一次后, `j = 6 i = 6`

`i = 5; j = i++;` 执行一次后, `j = 5 i = 6`

3. 用在判断条件表达式

```
i = 0;
```

```
while(i++)
```

```
{ printf("i=%d\n",i); if(i==3) break; } printf("i=%d\n",i);
```

则执行过程为 先把*i*的值拿出来判断条件, 此时`while(0)` 条件为假, 循环体内部的`printf` 没有做, 但是判断完条件后, 马上做了`i=i+1`, 则最后输出 `i=1`;

单片机开发仪视频教程

```
i = 0;
while(++i)
{ printf("i=%d\n",i);
  if(i==3) break;
}
printf("i=%d\n",i);
```

则执行过程为 先把i的值执行i=i+1，此时i=1; while(1) 条件为真, 然后再做循环体，此时输出 i=1
if不成立，继续循环

当i=1时，再判断循环条件++i，还是先执行i=i+1，此时i=2; while(2) 条件为真,然后再做循环体，
此时输出 i=2; if不成立，继续循环

当i=2时，再判断循环条件++i，还是先执行i=i+1，此时i=3; while(3) 条件为真,然后再做循环体，
此时输出 i=3; if成立，强制推出循环

符合运算符及表达式

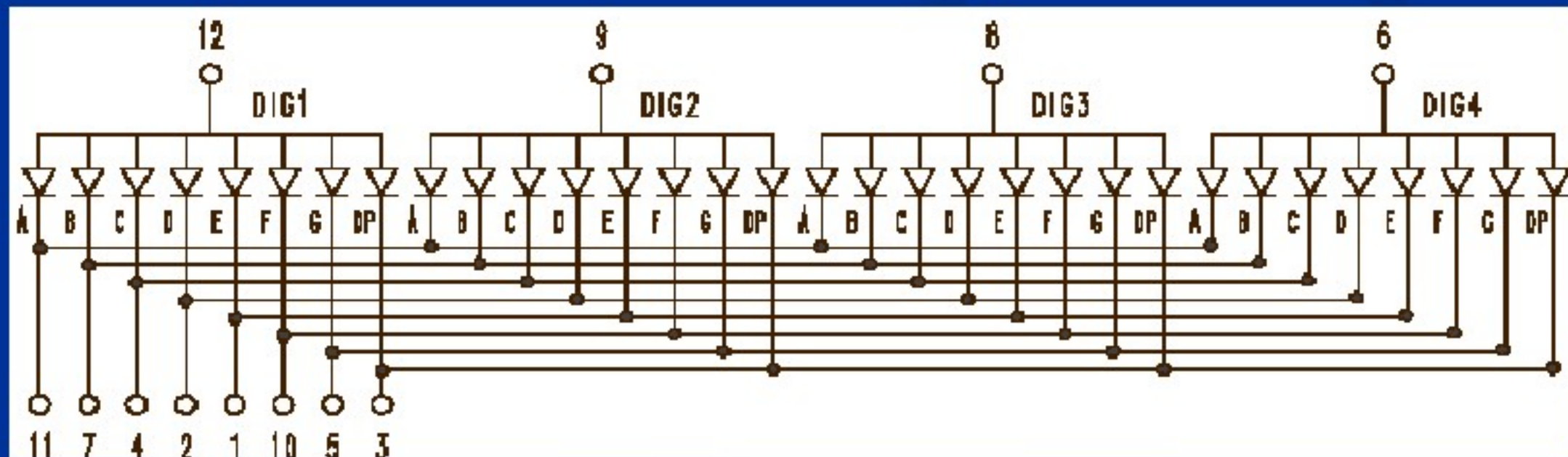
/ 取整运算符

% 取余运算符

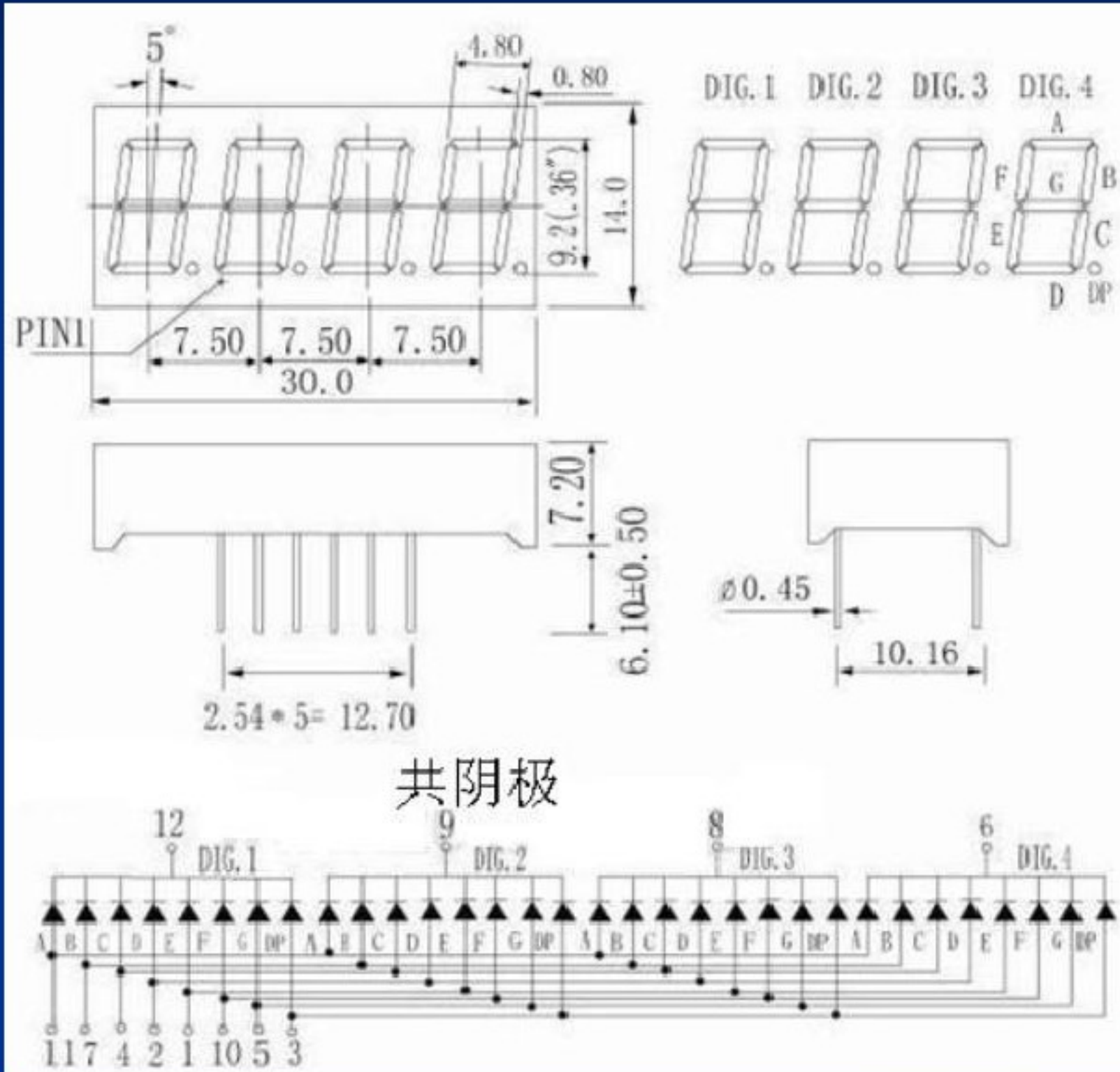
实验六 动态数码管实验

一、数码管动态显示和573锁存器相关知识：

数码管动态显示方式是单片机中应用最为广泛的一种显示方式之一，动态驱动是将所有数码管的8个显示笔划“a, b, c, d, e, f, g, dp”的同名端连在一起，另外为每个数码管的公共极COM增加位选通控制电路，位选通由各自独立的I/O线控制，当单片机输出字形码时，所有数码管都接收到相同的字形码，但究竟是哪个数码管会显示出字形，取决于单片机对位选通COM端电路的控制，所以我们只要将需要显示的数码管的选通控制打开，该位就显示出字形，没有选通的数码管就不会亮。通过分时轮流控制各个数码管的的COM端，就使各个数码管轮流受控显示，这就是动态驱动。在轮流显示过程中，每位数码管的点亮时间为1~2ms，由于人的视觉暂留现象及发光二极管的余辉效应，尽管各位数码管并非同时点亮，但只要扫描的速度足够快，给人的印象就是一组稳定的显示数据，不会有闪烁感，动态显示的效果和静态显示是一样的，能够节省大量的I/O端口，而且功耗更低。



二、动态数码管的内部结构图



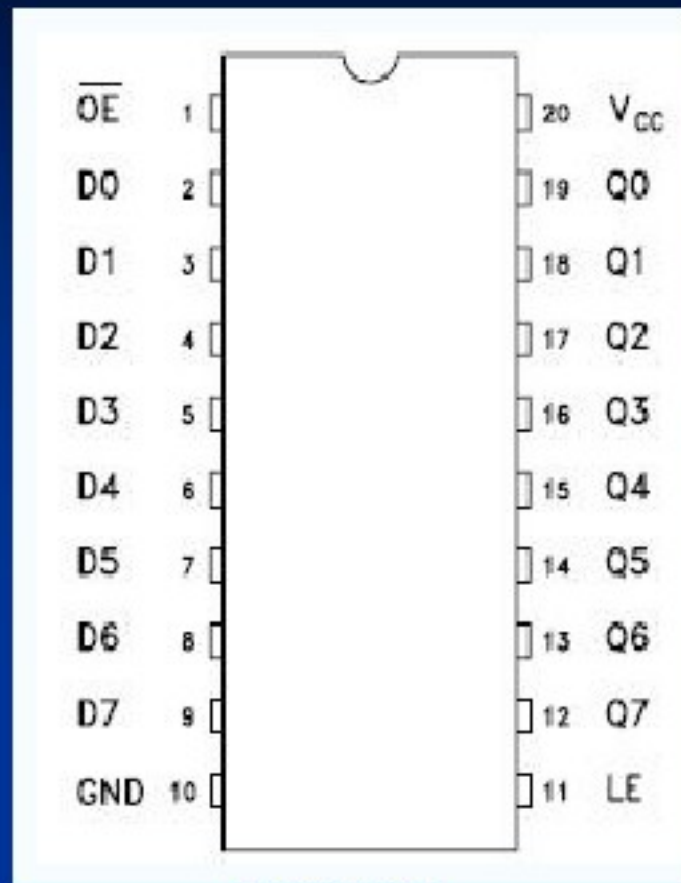
【数码管参数】

1、8字高度：8字上沿与下沿的距离。比外型高度小。通常用英寸来表示。范围一般为0.25-20英寸。（1英寸=25.4mm）

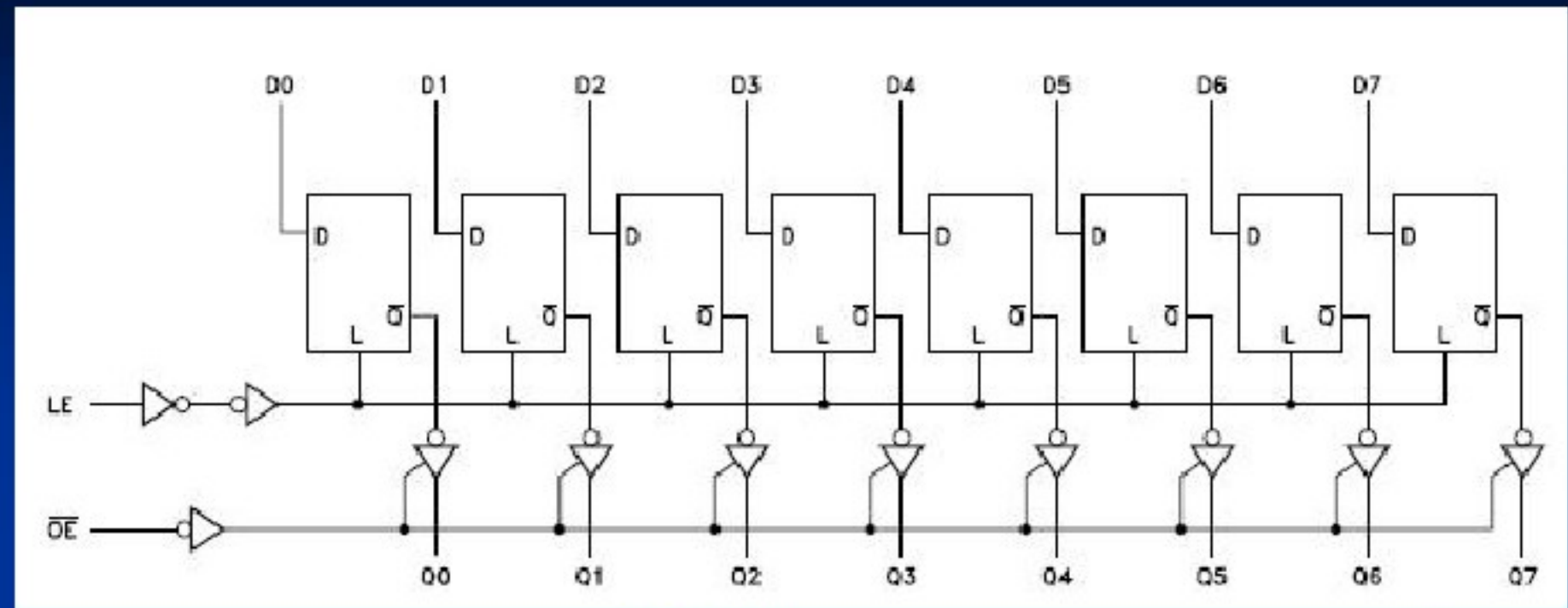
2、长*宽*高：长——数码管正放时，水平方向的长度；宽——数码管正放时，垂直方向上的长度；

3、高——数码管的厚度。

4、时钟点：四位数数码管中，第二位8与第三位8字中间的二点。一般用于显示时钟中的秒



573引脚图



573逻辑图

OE: 3 State output Enable Input (Active LOW) 3态输出使能输入 (低电平)

LE: Latch Enable Input 锁存使能输入

D0 to D7: Data Inputs 数据输入

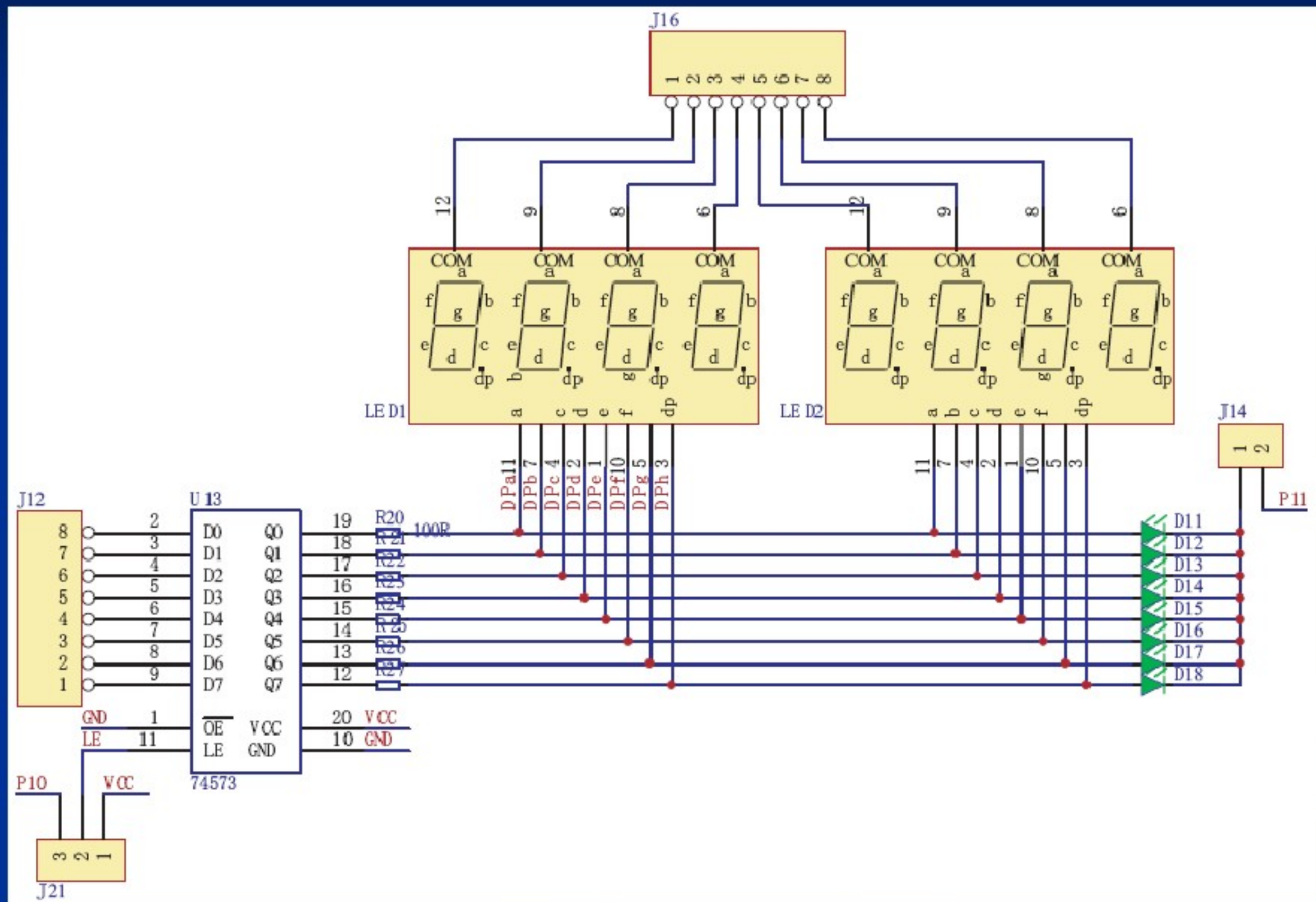
Q0 to Q7: Data Outputs 数据输出

原理说明:

74HC573的八个锁存器都是透明的D型锁存器，当使能(LE)为高时，输出(Q)将随输入数据(D)的变化而变化。当使能为低时，输出将锁存在已建立的数据电平上。输出控制不影响锁存器的内部工作，即老数据可以保持，甚至当输出被关闭时，新的数据也可以置入。这种电路可以驱动大电容或低阻抗负载，可以直接与系统总线接口相连并驱动总线，特别适用于缓冲寄存器，I/O通道，双向总线驱动器和工作寄存器。

74HC573的特点: 1、三态总线驱动输出 2、数据全并行存取 3、缓冲控制输入
4、使能输入有改善抗扰度的滞后作用

三、动态显示数码管原理图：



相关知识:

Switch/case语句: 是一个多分支选择语句

Switch(表达式) //这个表达是可以为任何类型

```
{  
    case 1: 语句1;  
    case 2: 语句2;  
    case 3: 语句3;  
    case 4: 语句4;  
    .  
    .  
    .  
    .  
    case n: 语句n;  
    default: 语句 n+1;  
}
```

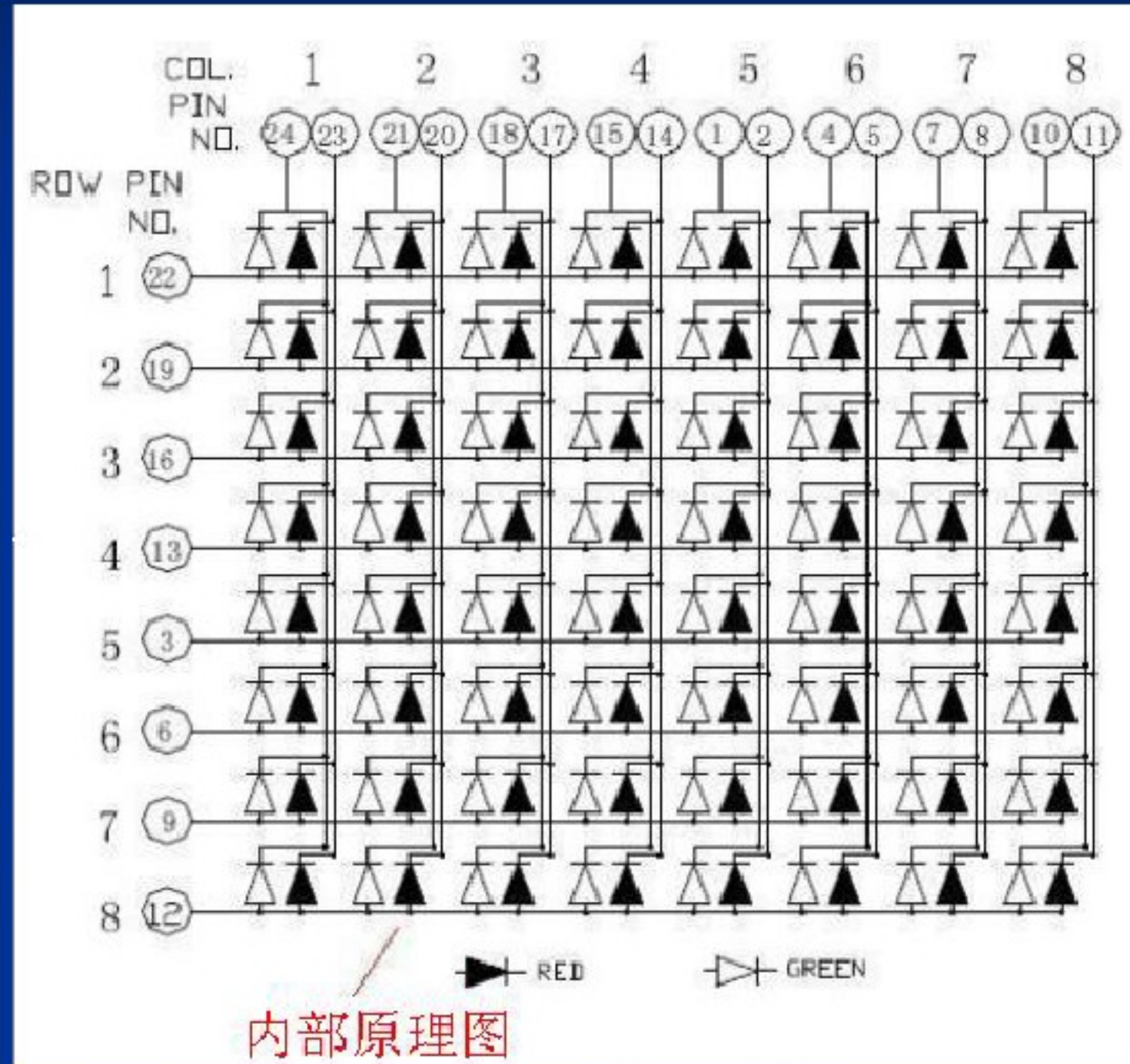
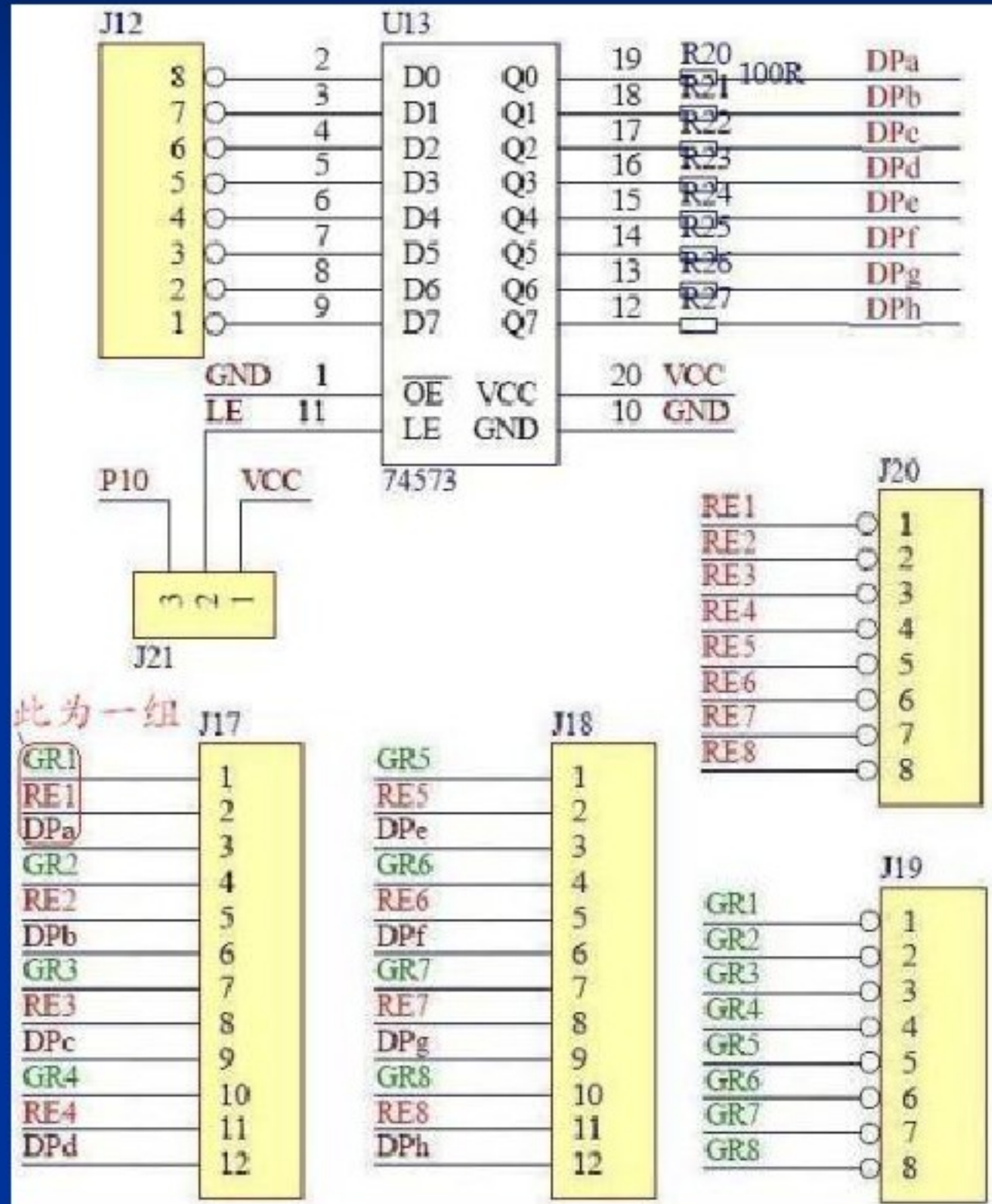

实验七 LED点阵实验

一、LED点阵的知识:

LED点阵: 它以发光二极管为像素，它用高亮度发光二极管芯阵列组合后，用环氧树脂和塑模封装而成。具有高亮度、功耗低、引脚少、视角大、寿命长、耐湿、耐冷热、耐腐蚀等特点。点阵显示器有单色、双色和彩色三类，可显示红，黄，绿，橙等。LED点阵有 4×4 、 4×8 、 5×7 、 5×8 、 8×8 、 16×16 、 24×24 、 40×40 等多种；

LED点阵扫描驱动方式: LED点阵采用动态扫描驱动方式工作，由于LED管芯大多为高亮度型，因此某行或某列的单体LED驱动电流可选用窄脉冲，但其平均电流应限制在20mA内。多数点阵显示器的单体LED的正向压降约在2V左右。但大亮点 $\phi 10$ 的点阵显示器单体LED的正向压降约为6V。大屏幕显示系统一般是将由多个LED点阵组成的小模块以搭积木的方式组合而成的，每一个小模块都有自己的独立的控制系统，组合在一起后，只要引入一个总控制器控制各模块的命令和数据即可，这种方法既简单而且具有易扩展、易维修的特点。LED点阵显示系统中各模块的显示方式有静态和动态显示两种。静态显示原理简单、控制方便，但硬件接线复杂，在实际应用中一般采用动态显示方式，动态显示采用扫描的方式工作，由峰值较大的窄脉冲驱动，从上到下逐次不断地对显示屏的各行进行选通，同时又向各列送出表示图形或文字信息的脉冲信号，反复循环以上操作，就可显示各种图形或文字信息。

二、LED点阵原理图：



三、LED点阵显示原理

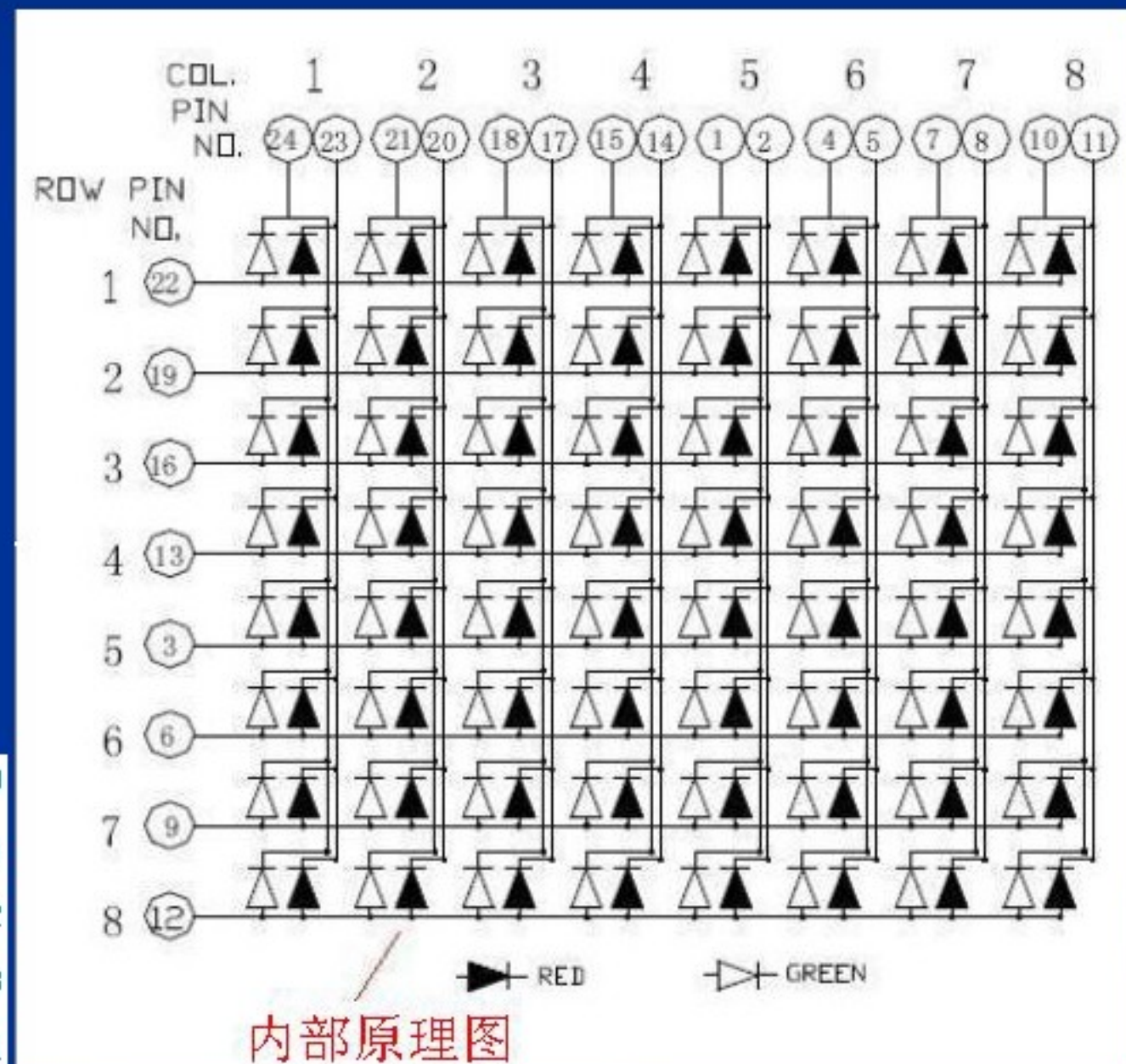


DPa
DPb
DPc
DPd
DPe
DPf
DPg
DPH

```
unsigned char code tab[]={0x7f,0xbf,0xdf,0xef,0xf7,0xfb,0xfd,0xfe,}:
```



```
{0x00, 0x00, 0x3e, 0x41, 0x41, 0x41, 0x3e, 0x00}, //0
{0x00, 0x00, 0x00, 0x00, 0x21, 0x7f, 0x01, 0x00}, //1
{0x00, 0x00, 0x27, 0x45, 0x45, 0x45, 0x39, 0x00}, //2
{0x00, 0x00, 0x22, 0x49, 0x49, 0x49, 0x36, 0x00}, //3
{0x00, 0x00, 0x0c, 0x14, 0x24, 0x7f, 0x04, 0x00}, //4
```



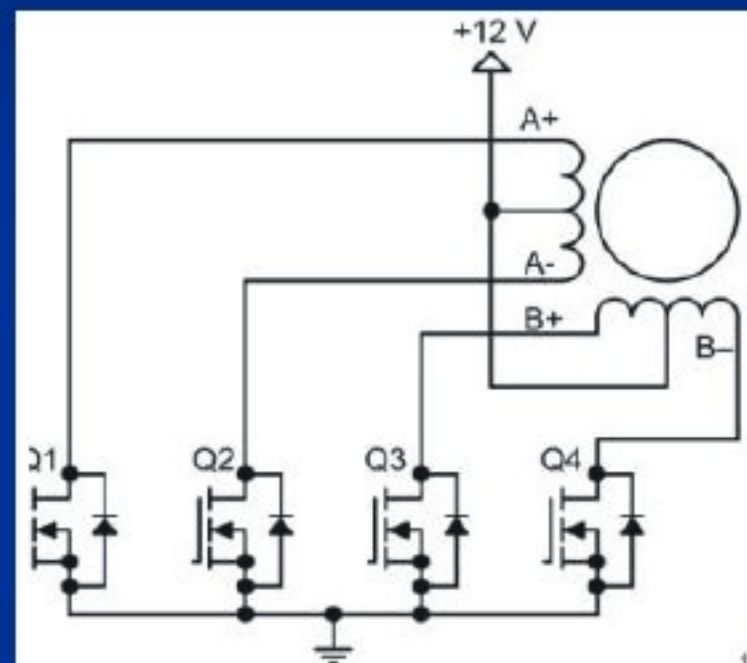
实验八 步进电机实验

一、步进电机的相关知识：

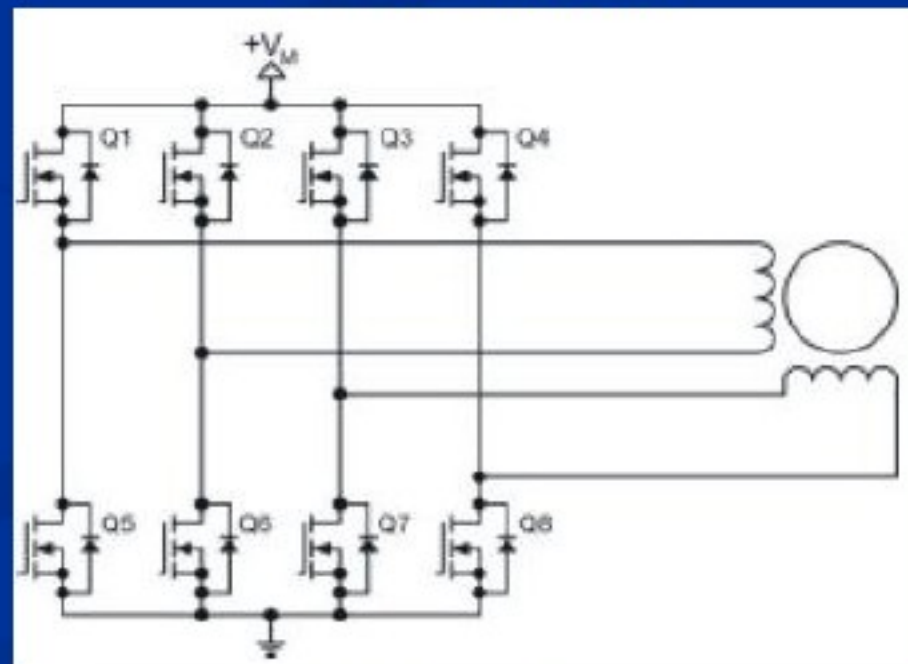
步进电机分单极性和双极性两种：

单极性步进电机有两个线圈，但有五条或六条线，也就是在一个线圈的中间增加了一个抽头，五条线的也可以看成是六条线，它是把两个线圈的两根中间线并在一起。由于在一个线圈的中间有了抽头，电流就可以在一个线圈的一半走不同的流向。但这时，只是用到电机线圈的一半而已。

双极性步进电机之所以如此命名，是因为每个绕组都可以两个方向通电。因此每个绕组既可以是N极又可以是S极。它又被称为单绕组步进电机，因为每极只有单一的绕组，它还被称为两相步进电机，因为具有两个分离的线圈。双极性步进电机有四根引线，每个绕组两条。与同样尺寸和重量的单极性步进电机相比，双极性步进电机具有更大的驱动能力，原因在于其磁极（不是中间抽头的单一线圈）中的场强是单极性步进电机的两倍。双极性步进电机的每个绕组需要一个可逆电源，通常由H桥式驱动电路提供。由于双极性步进电机比单极性步进电机的输出力矩大，因此总是应用于空间有限的设计中。这也是软盘驱动器的磁头步进机械系统的驱动之所以采用双极性步进电机的原因。



单极性驱动电路



H桥式驱动电路

步进电机运行有三种步进方式：单拍、双拍、半拍方式。

单拍方式：它是指每次仅给一个绕组通电，使转子旋转，并运动到转子永磁体与具有相反极性的绕组对齐的位置。

双拍方式：它同时给两个绕组通电，这样就使转子旋转，并在永磁体到达两个通电绕组的中间位置点时平衡。双拍方式的优点是比单拍方式多获得41.4%的输出力矩，不过代价是需要花费后者两倍的能量，因为它有两相绕组同时通电。

半拍方式：它工作时则让两个绕组通电与单个绕组通电方式交替地进行。半拍方式的输出力矩比双拍方式小，随设计不同，在15%——30%之间变化，不过它可以获得双拍方式两倍的步进分辨率（每圈两倍的步数）。

单极性步进电机的节拍有(5线4为例)

单四拍： A-B-C-D

双四拍： AB-BC-CD-DA

单双8拍： A-AB-B-BC-C-CD-D-DA（半步工作方式）

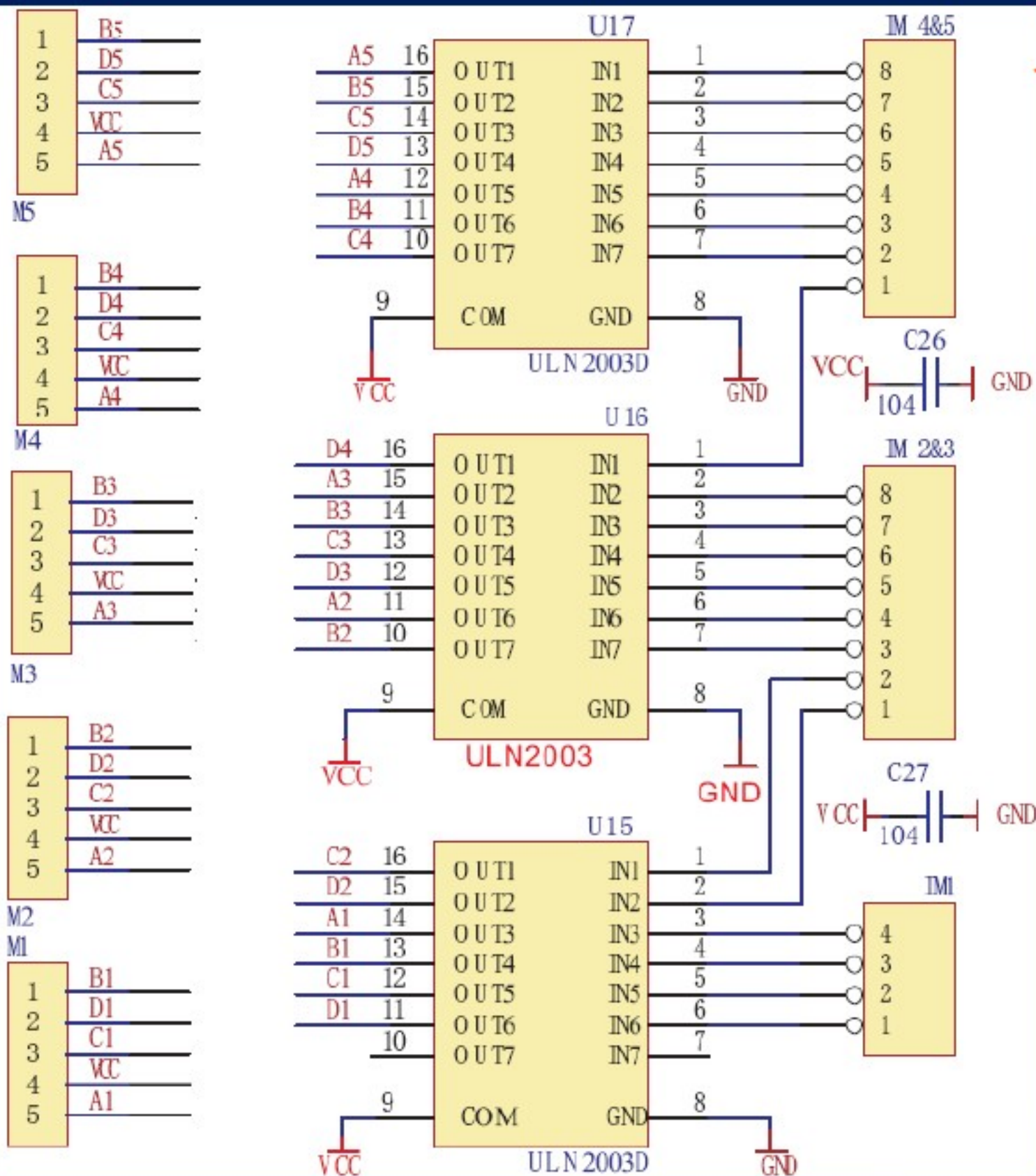
双极性步进电机的节拍（4线2相为例）

单四拍： A/ - B - A - B/

双四拍： A/B - AB - AB/ - A/B/

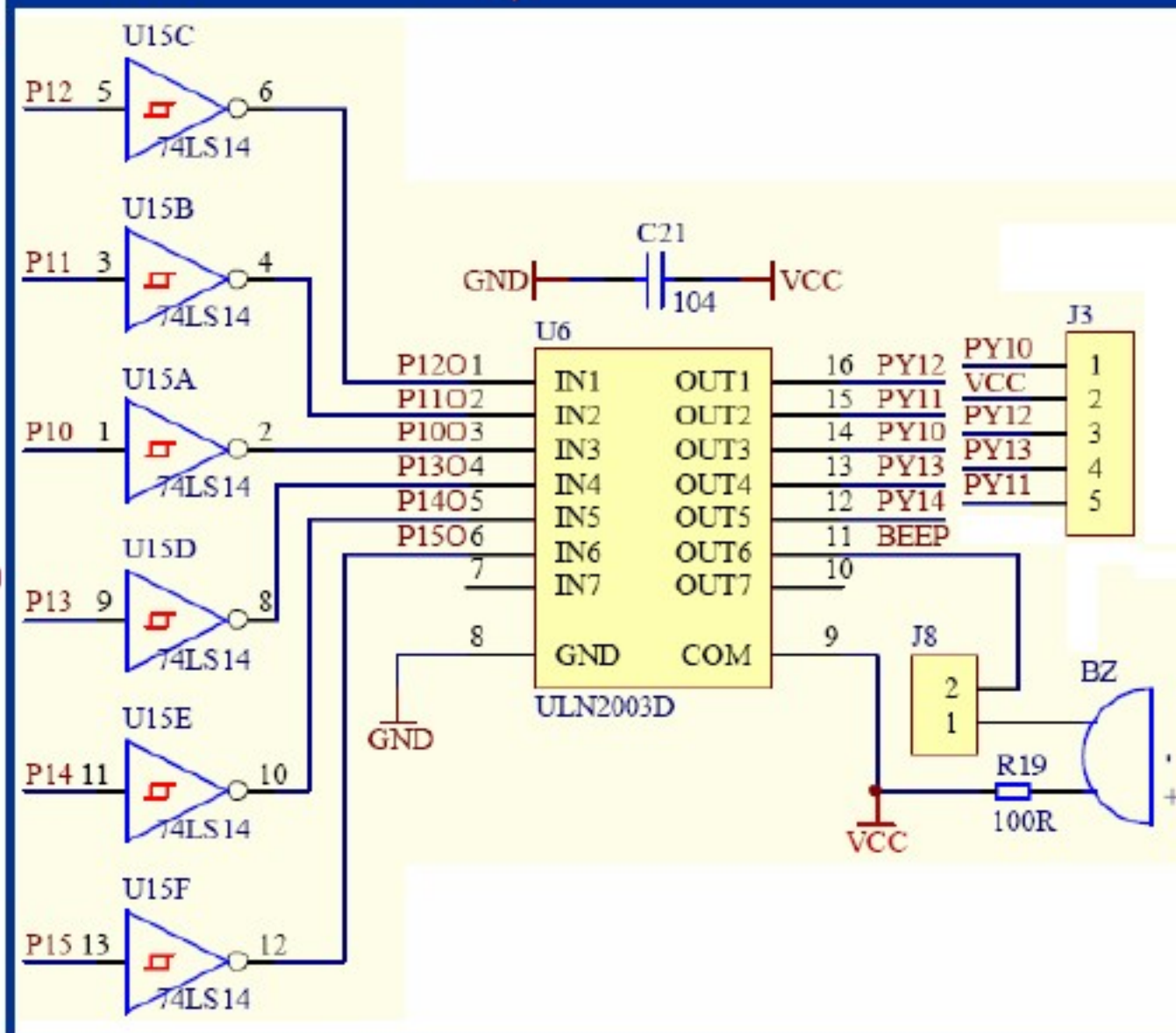
单双8拍： A/ A/B B AB A AB/ B/ A/B/

三、步进电机的原理图

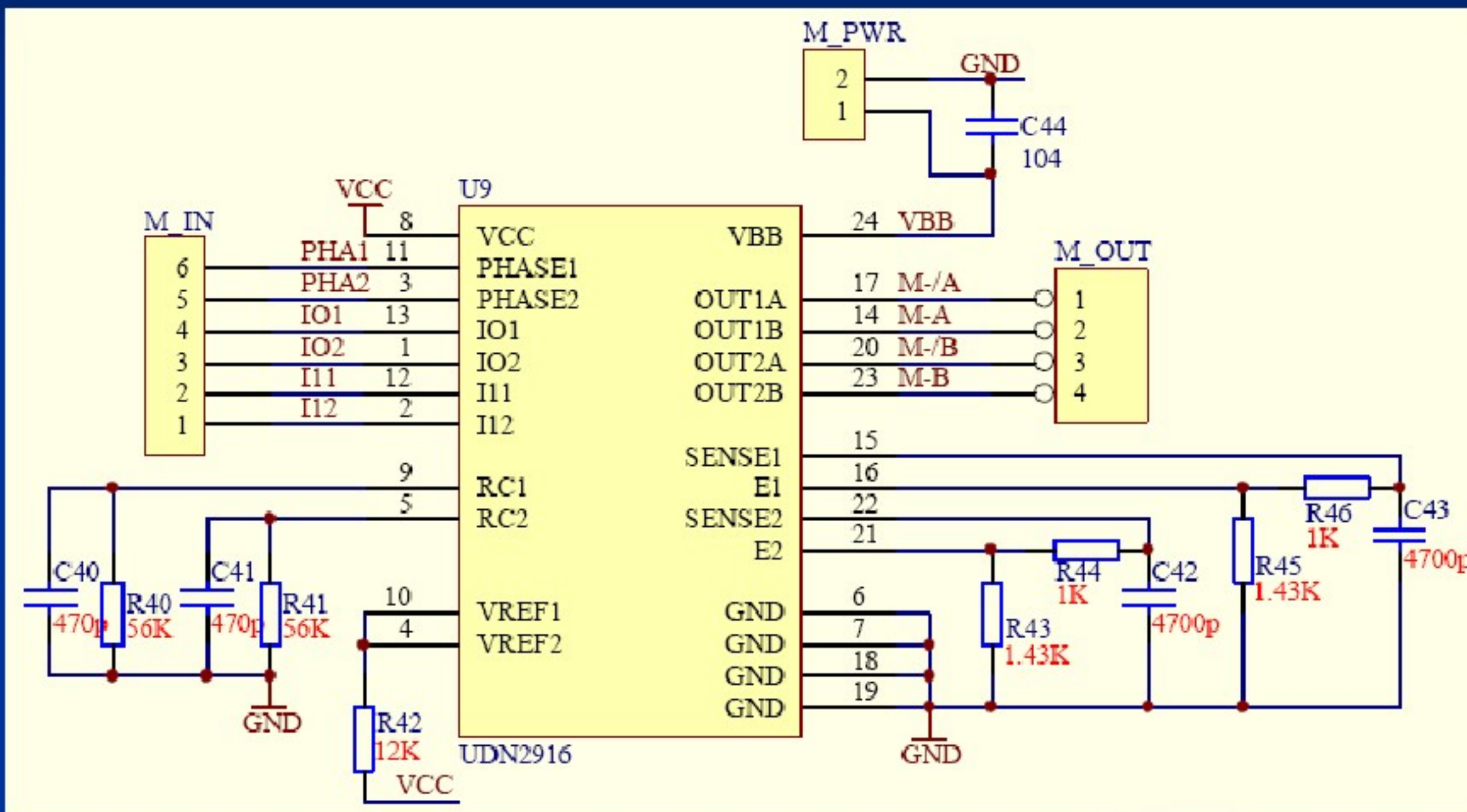


HC6800EM3原理图

HC6800原理图



专业步进驱动芯片原理图



四、实验的接线演示：

看视频演示

五、例程演示：

1. 《单双八拍1》的C语言作为例程演示

HC6800

拍	D	C	B	A	HEX
A	1	1	1	0	0xfe
AB	1	1	0	0	0xfc
B	1	1	0	1	0xfd
BC	1	0	0	1	0xf9
C	1	0	1	1	0xfb
CD	0	0	1	1	0xf3
D	0	1	1	1	0xf7
DA	0	1	1	0	0xf6

HC6800EM3

拍	D	C	B	A	HEX
A	0	0	0	1	0xf1
AB	0	0	1	1	0xf3
B	0	0	1	0	0xf2
BC	0	1	1	0	0xf6
C	0	1	0	0	0xf4
CD	1	1	0	0	0xfc
D	1	0	0	0	0xf8
DA	1	0	0	1	0xf9

```
RUN[8]={0xfe,0xfc,0xfd,0xf9,0xfb,0xf3,0xf7,0xf6}; //步进电机相序表
```

```
RUN[8]={0xf1,0xf3,0xf2,0xf6,0xf4,0xfc,0xf8,0xf9}; //步进电机相序表
```

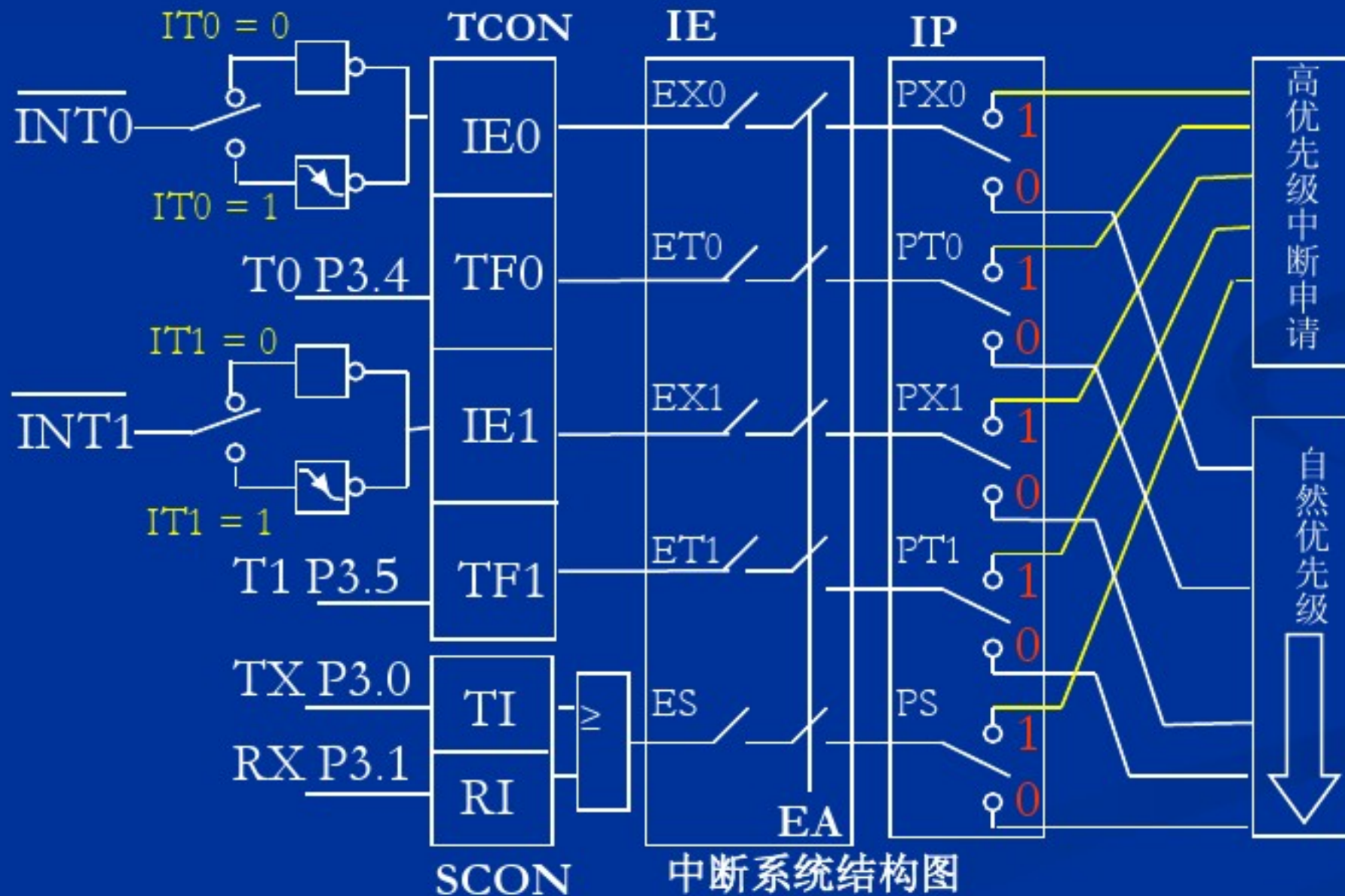

实验九 中断与外部中断实验

80C51 提供了5个中断源：

2个外部中断INT0(P3.2)和INT1(P3.3)，

2个片内定时器/计数器T0(P3.4)和T1(P3.5)的溢出中断TF0和TF1，

1个片内串行口的发送或接收中断请求源TI或RI。



定时器/计数器控制寄存器TCON

位	D7	D6	D5	D4	D3	D2	D1	D0
功能	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

IT0: INT0的触发方式控制位，由软件设置。

IT0=0时，INT0为低电平触发方式

IT0=1时，INT0为负跳变触发方式。

IE0: INT0中断请求标志位。当INT0引脚

上出现中断请求信号时，由硬件置位IE0

在CPU响应中断后，再由硬件将IE0清0

中断允许控制寄存器IE

位	EA	—	—	ES	ET1	EX1	ET0	EX0
地址	AFH	—	—	ACH	ABH	AAH	A9H	A8H

EA: CPU中断允许标志位，1: 开发所有，0: 禁止

ES: 串行口允许中断位，1: 允许，0: 禁止

ET1: 定时器1允许中断位，1: 允许，0: 禁止

EX1: 外部中断1允许中断位，1: 允许，0: 禁止

优先级控制寄存器IP

IP	—	—	—	PS	PT1	PX1	PT0	PX0
地址	—	—	—	BCH	BBH	BAH	B9H	B8H

PS: 串行口中断优先级控制位

PT1: 定时器T1中断优先级控制位

PX1: 外部中断1中断优先级控制位

PT0: 定时器T0中断优先级控制位

PX0: 外部中断0中断优先级控制位

1: 设置中断为高优先级
0: 设置中断为低优先级

5个中断源分别设置成不同的优先级。如果都被设置成同一优先级，这5个中断源因硬件的组成不同而形成不同的内部序号，从而构成了不同的自然优先级。每个中断源，有对应的中断服务程序，这些程序有固定的存放位置，当产生相应的中断后，就可以跳到相应的位置执行中断程序。这个位置就在中断向量的入口地址。每个位置只有8个字节的程序存储空间，但是中断程序很多都会超过8个字节。在汇编语言中，用LJMP跳转指令，把实际处理的中断程序放到ROM的任何位置。而在C语言编程中，C语言编译器自行处理，不需要我们考虑。

中断源	同级内部自然优先级	中断向量入口地址
外部中断0	最高级 ↓ 最低级	0003H
定时器0		000BH
外部中断1		0013H
定时器T1		001BH
串行口		0023H

5个中断源的自然优先级和中断入口地址

中断服务程序函数的格式：**返回类型 函数名(参数) interrupt n**

严格完整格式：**返回类型 函数名(参数) interrupt n using n**

其中**interrupt n**中n对应中断源的编号，其值从0开始。以80C51为例 n的编号0-4,分别是：
0、外部中断0 1、定时器0中断 2、外部中断1 3、定时器1中断 4、串口中断
using n中 n 的值是0-3，对应使用4组工作寄存器。

关于51单片机工作寄存器组

为了进行中断的现场保护，80C51单片机除采用堆栈技术外，还采用了寄存器的方式。51单片机有4组名称均为R0-R7的工作寄存器，中断产生时，可以通过简单地设置RS0和RS1来切换工作寄存器组。提高了51单片机现场保护和现场恢复的速度，对于提高CPU的工作效率和响应中断的速度是很有利。若在一个实际的应用系统中，不需要四组工作寄存器，那么这个区域中多余单元可以作为一般的数据缓冲器使用。

高优先级中断可以中断正在处理的低优先级程序，因而必须注意寄存器组，分配的方法是使用**using n**来制定。

RS1	RS0	类型	字节地
0	0	0组寄存器	00H~07H
0	1	1组寄存器	08H~0FH
1	0	2组寄存器	10H~17H
1	1	3组寄存器	18H~1FH

PSW程序状态寄存器

位	D7	D6	D5	D4	D3	D2	D1	D0
含义	CY	AC	F0	RS1	RS0	OV		P

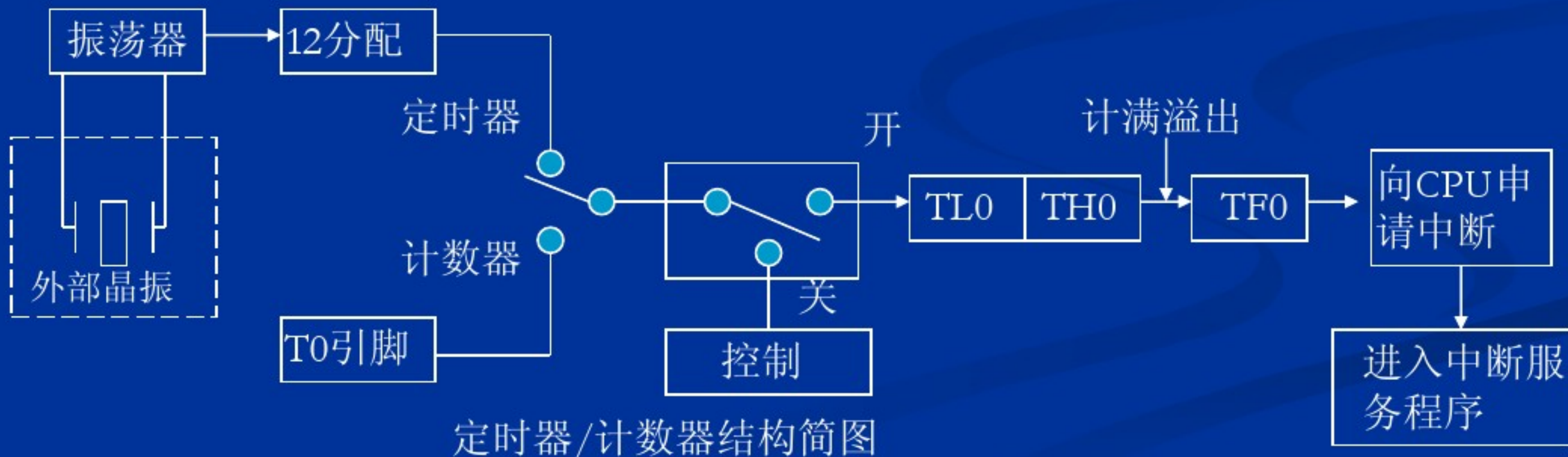
CY: 进位标志 AC: 辅助进、借位
F0: 用户标志位 OV: 溢出标志位
P: 奇偶校验位

实验十 定时器中断实验

80C51单片机内部集成了16位的可编程定时器/计数器，他们分别是定时器/计数器0(T0)和定时器/计数器1(T1)。他们既可以工作在定时器方式，实现对控制系统的定时或精准延时控制，又可以工作在计数方式，用于对外部的脉冲事件计数。

与T0和T1相关的寄存器：

- 1.有2个8位控制寄存器TMOD和TCON，
- 2.有2个16位的计数器TH0、TL0、TH1、TL1，他们都是16位的加1计数器也就是增量式计数器。



TMOD 工作模式控制寄存器

功能：用来设置各个定时器/计数器的工作方式，选择定时器或计数器功能。

控制T1

控制T0

位	D7	D6	D5	D4	D3	D2	D1	D0
含义	GATE	C/ \bar{T}	M1	M0	GATE	C/ \bar{T}	M1	M0

工作方式0：13位定时器 / 计数器 ,最多可计数2的13次方次。

工作方式1：16位定时器/计数器,最多可计数2的16次方次,。

工作方式2：8位定时器/计数器，计算次数最多为2^8。8位自动重装方式

工作方式3：8位定时器/计数器，计算次数最多为2^8。

由于TMOD只能直接字节寻址，所以对T0和T1的工作方式只能以字节写入。

```
TMOD|= 0x11;
TH1=0x1C; //12.000M
TL1=0x18;
IE = 0x8A;
TR1 = 1;
```

M1M0	方式
00	模式0
01	模式1
10	模式2
11	模式3

C/ \bar{T} = 0

定时器模式

C/ \bar{T} = 1

计数器模式

门控位

GATE = 0

与INT0无关

GATE = 1

与INT0有关

TCON 工作状态控制寄存器(地址 88H)

功能：用来控制定时器/计数器的启停和作为运行状态的标志等。

功能	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
位地址	8F	8E	8D	8C	8B	8A	89	88

TCON寄存器即可字节寻址也可以位寻址
字节地址为 88H，
位地址为88H—8FH

```

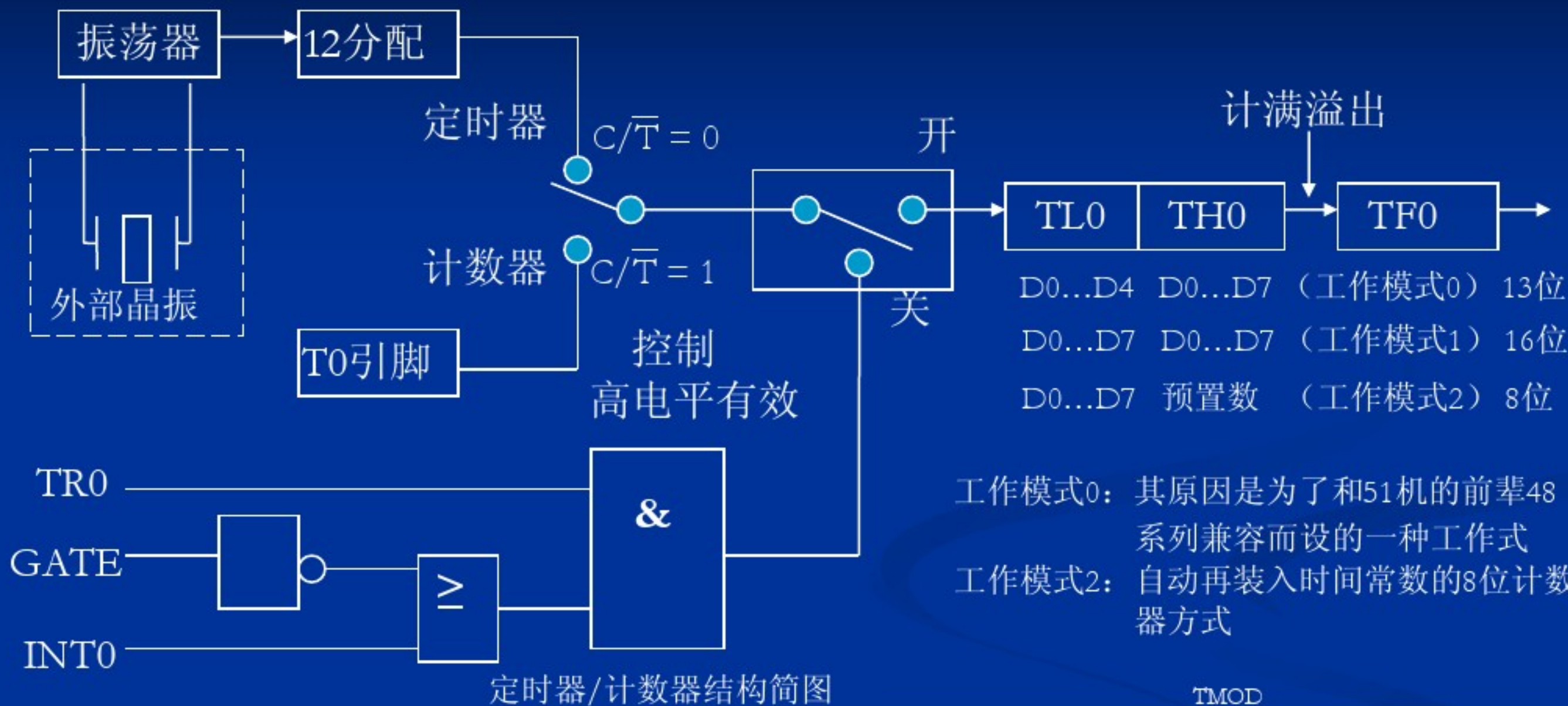
/*****
purpose: 系统初始化
*****/
void system_Ini ()
{
    TMOD|= 0x11;
    TH1=0x1C; //12.000M
    TL1=0x18;
    IE = 0x8A;
    TR1 = 1;
}
    
```



TCON的低四位和外部中断有关

什么是中断？
由于某个事件的发生，CPU暂停当前正在执行的程序，转而执行处理该事件的一个程序。该程序执行完成后，CPU接着执行被暂停的程序。这个过程称为中断。

定时器/计数器4种工作方式

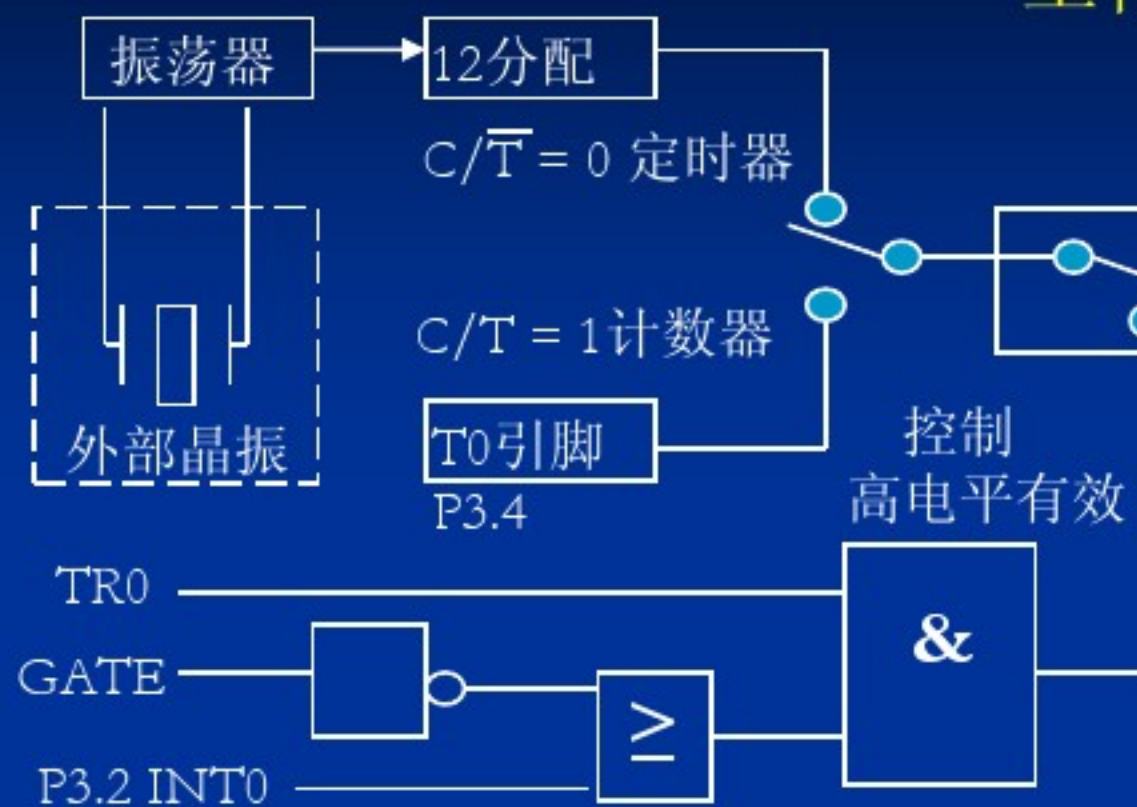


定时器/计数器的打开或关闭，由GATE、软件控制位TR和INT共同决定。

GATE = 0，只要用指令置TR为1即可启动定时器/计数器，而不管INT的状态如何。

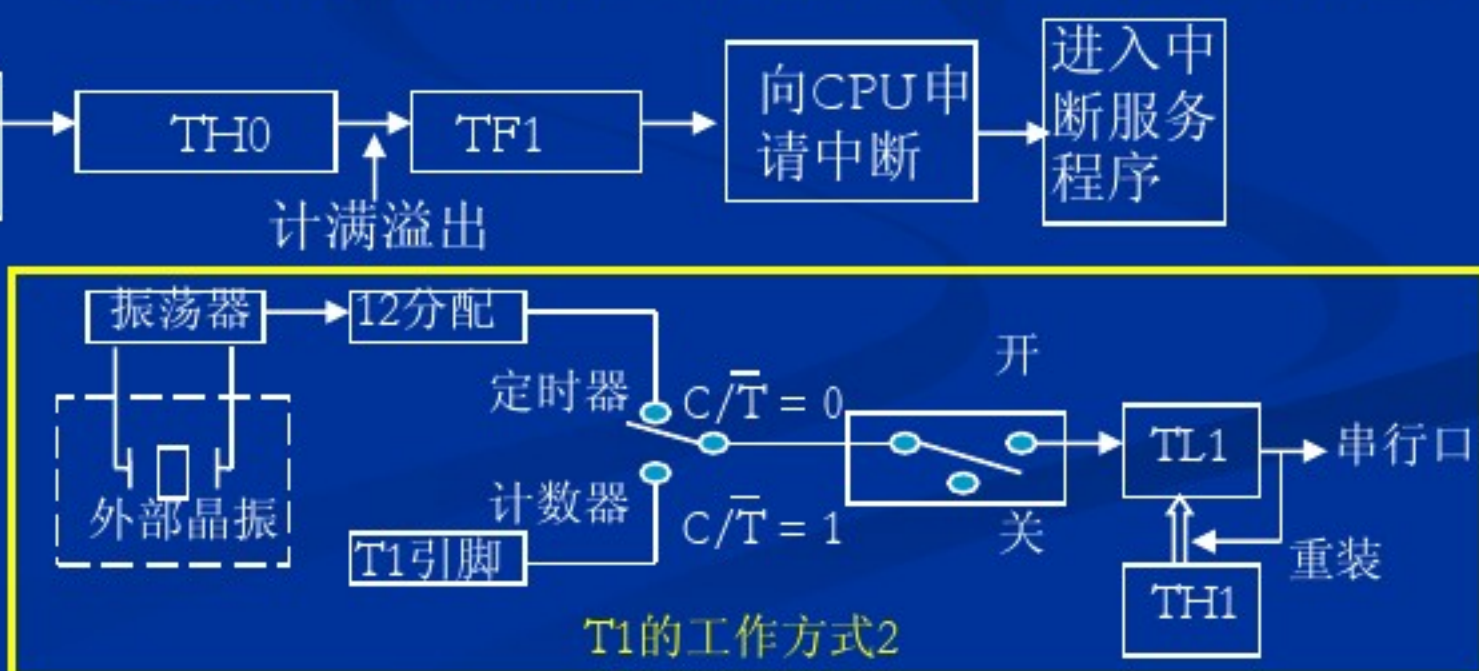
GATE = 1，只要INT为高电平且用指令置TR为1时，才能启动定时器/计数器

工作模式3



T0工作在方式3，TL0构成8位计数器可用于定时/计数，使用T0的控制位（TR0）和TF0的中断源。但是，TH0则只能工作在定时器状态，它借用T1的TR1和TF1的中断源。如硬把T1工作在方式3时，T1将处于关闭状态。

一般情况下，只有T1以工作方式2运行（当波特率发生器用）时，才让T0工作于方式3。因为T1的TR1和TF1的中断源被定时/计数器0借用了，只能把计数溢出直接送给串行口。



定时器/计数器初值的计算

80C51的T1和T0中的计数器都是增量式的，因此，不能直接将实际要计数的值作为初值放入计数寄存器中，而是将计数的最大值减去实际要计数的值放入计数寄存器中。

定时模式计数脉冲是由单片机的晶体振荡器产生的频率信号经12分频得到。因此在设置定时时间之前，要确定机器的晶振频率。

开发试验仪采用的12M的晶振
频率的计算公式：

$f_{osc} = 1 / T_{osc}$ 。(fosc表示频率 Tosc表示周期)

计数的周期：

$T = (1/12\ 000\ 000) * 12 = 1\mu S$ (12分频)

计数的次数：

计数的次数 = 定时时间 / $1\mu S$

假设我们定时的时间是 $100\mu s$

计数的次数 = $100\ \mu S / 1\mu S = 100$ 次

如果选用定时器0

工作方式0：计算出来的初值 = $8192 - 100$

工作方式1：计算出来的初值 = $65535 - 100$

工作方式2：计算出来的初值 = $256 - 100$ (自动装载)

工作方式3：计算出来的初值 = $256 - 100$

```

/*****
purpose: 系统初始化
*****/
void system_Ini()
{
    TMOD |= 0x11;
    TH1=0x1C: //12.000M
    TL1=0x18;
    IE = 0x8A;
    TR1 = 1;
}

```

```

/*****
[ t1 (1ms)中断] 中断
*****/
void T1zd(void) interrupt 3
{
    TH1 = 0x1c; //12.000
    TL1 = 0x18;

    LedNumVal++;
}

```


四、实验的接线演示：

看视频演示

五、例程演示：

1. 《timer0》和《计数器功能》的C语言作为例程演示

知识点：

中断程序的编写

中断服务程序函数的格式：**返回值 函数名 interrupt n** 其中n对应中断源的编号，其值从0开始。以80C51为例 n的编号0-4,分别是：

0 外部中断0

1 定时器0中断

2 外部中断1

3 定时器1中断

4 串口中断

实验十一 矩阵键盘实验

矩阵式键盘的结构与工作原理:

在键盘中按键数量较多时，为了减少I/O口的占用，通常将按键排列成矩阵形式，在矩阵式键盘中，每条水平线和垂直线在交叉处不直接连通，而是通过一个按键加以连接。这样，一个端口（如P1口）就可以构成 $4 \times 4 = 16$ 个按键，比之直接将端口线用于键盘多出了一倍，而且线数越多，区别越明显，比如再多加一条线就可以构成20键的键盘，而直接用端口线则只能多出一键。因此，在需要的键数比较多时，采用矩阵方式来做键盘是合理的。

矩阵式键盘结构显然比直接法要复杂一些，识别也要复杂一些。下图中，我们用程序把行线置低为电平，列线为高电平。当按键没有按下时，行线和列线状态不变，一旦有键按下，则高电平的某条列线将被低电平的行线拉低，这样，通过读入被拉低列线的状态就可得知是否有键按下了。确定矩阵式键盘上哪个键被按下是一种“行扫描法”。行扫描法又称为逐行（或列）扫描查询法，是一种最常用的按键识别方法，

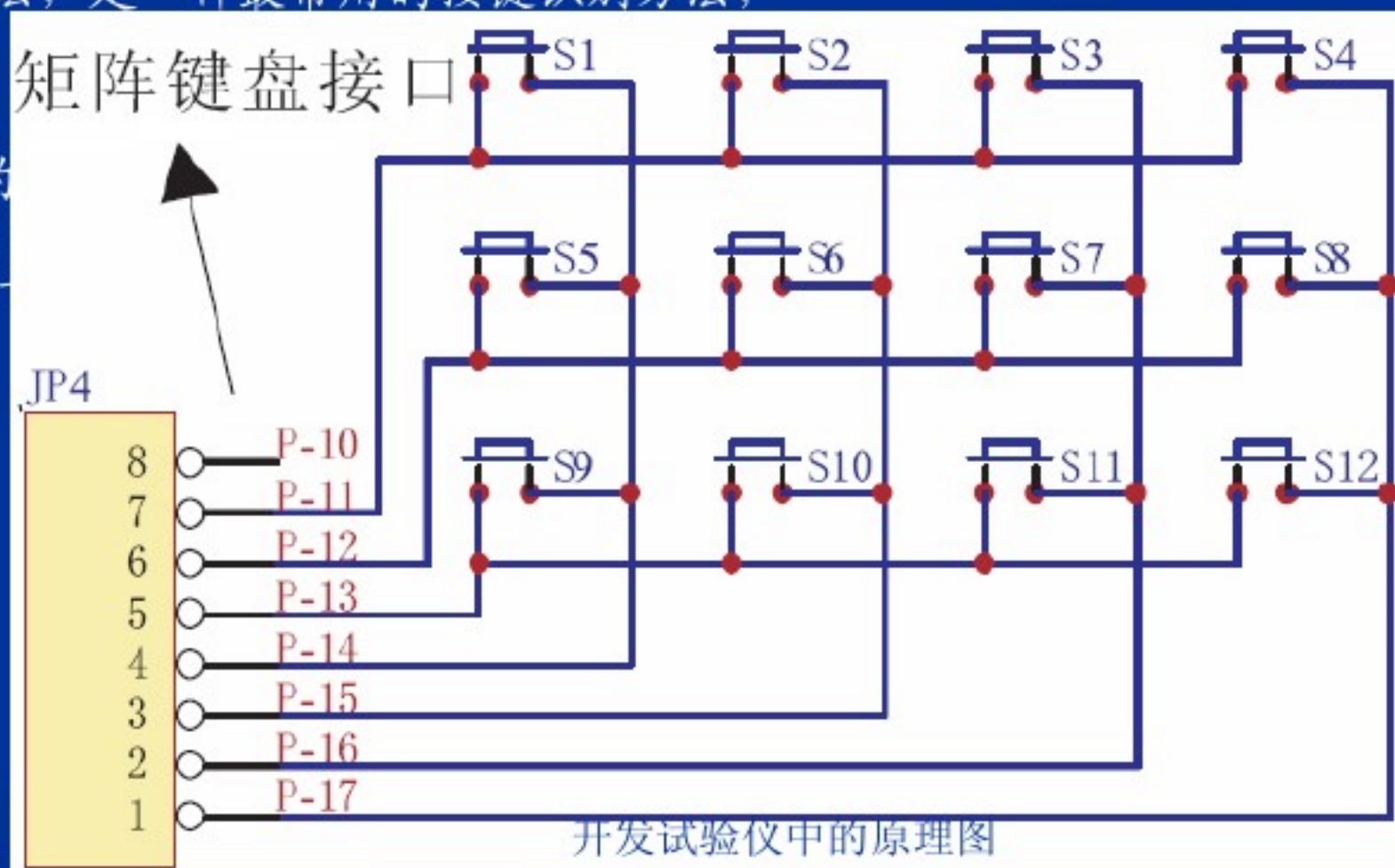
介绍开发仪键盘识别过程如下：

1. 判断键盘中是否有键按下 将全部行线P10-P13置低电平，列线P14-P17置高电平，然后检测列线的状态。只要有一列的电平为低，则表示键盘中有键被按下。若所有列线均为高电平，则键盘中无键按下。

2. 判断闭合键所在的位置 在确认有键按下后，即可进入按键检测的过程。其方法是：

1. 把行线置为低电平，列线置位高电平，用程序检测4条列线的状态。

2. 停止列线的检测，把列线置位低电平，行线置位高电平，用程序检测3条行线的状态。



实验十二 单片机IO扩展一(74HC595)实验

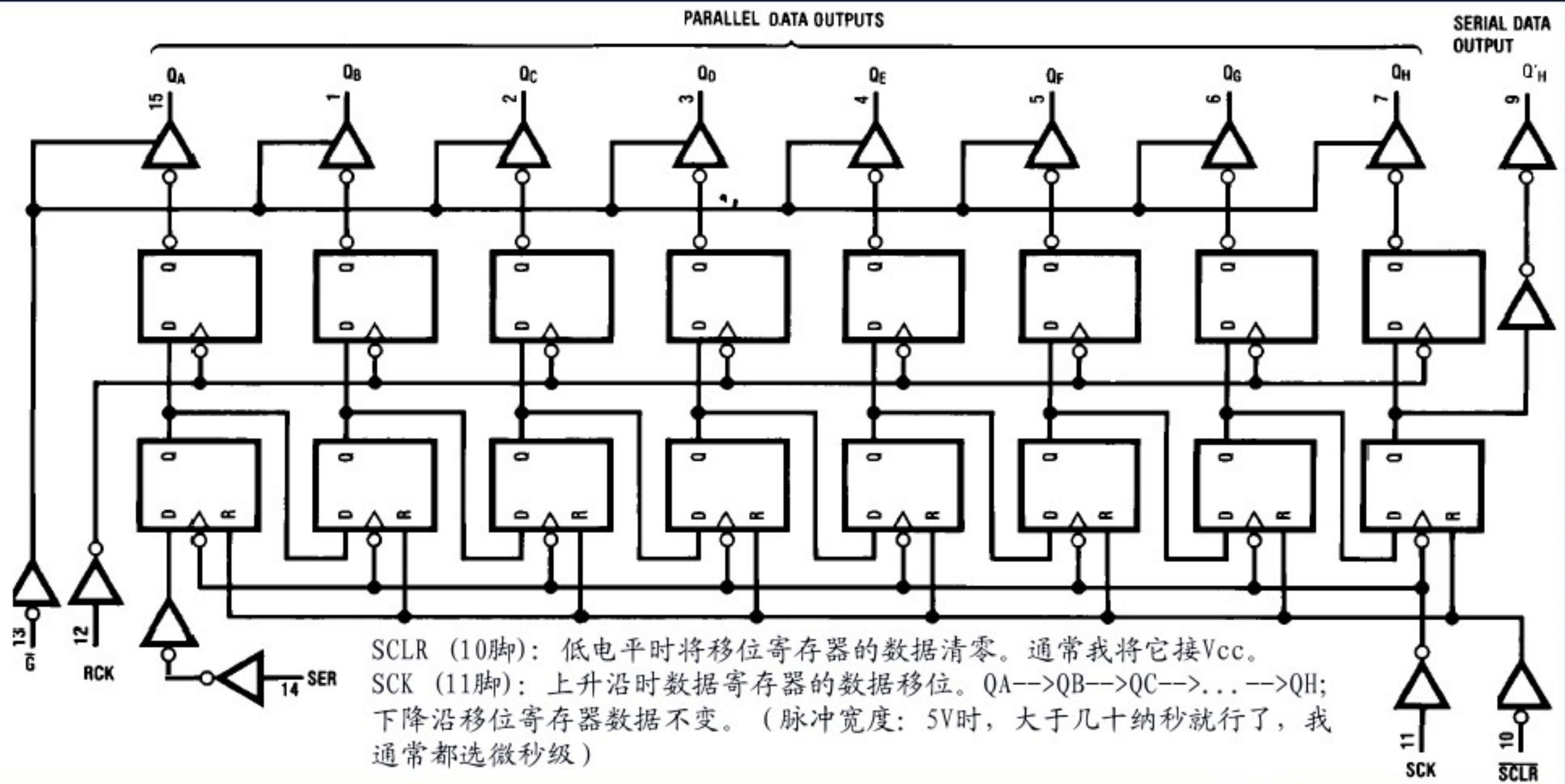
SPI总线简介

SPI (Serial Peripheral Interface) 顾名思义, 就是串行外围设备接口, 由摩托罗拉公司推出的一种高速、全双工、同步的通信总线, 并且在芯片的管脚上只占用四根线, 节约了芯片的管脚。它由一个主设备和一个或多个从设备组成, 主设备启动一个与从设备的同步通讯, 从而完成数据的交换。该总线大量用在EEPROM、ADC、显示驱动器之类的慢速外设器件通信。

SPI接口由四种信号构成

1. SDI: 串行数据输入
2. SDO: 串行数据输出
3. SCK: 串行移位时钟
4. CS: 从设使能信号

我们开发试验仪采用的74HC595和74HC165这两个芯片是遵循SIP协议的。它们只用到SPI的3条线, 所以只能单向传输数据。由于51单片机没有硬件的SPI总线控制器, 所以只能用软件模拟的方式来实现SPI协议通信。



QA--QH:
八位并行
输出端可
以直接控
制数码管
的8个段

SQH (9脚):
级联输出
端。可以
接下一个
595的串行
数据输入
端

SCLR (10脚): 低电平时将移位寄存器的数据清零。通常我将它接Vcc。
SCK (11脚): 上升沿时数据寄存器的数据移位。QA-->QB-->QC-->...-->QH;
下降沿移位寄存器数据不变。(脉冲宽度: 5V时, 大于几十纳秒就行了, 我
通常都选微秒级)

74HC595包括一个8位移位寄存器和一个8位D型锁存器和三态并行输出。移位寄存器接收串行数据并提供串行输出。移位寄存器和锁存器都有独立的时钟输入。这个IC还具有异步复位的功能。

RCK (12脚): 上升沿时移位寄存器的数据进入数据存储寄存器, 下降沿时存储寄存器数据不变。通常我将RCK置为低电平, 当移位结束后, 在RCK端产生一个正脉冲(5V时, 大于几十纳秒就行了, 通常都选微秒级), 更新显示数据。

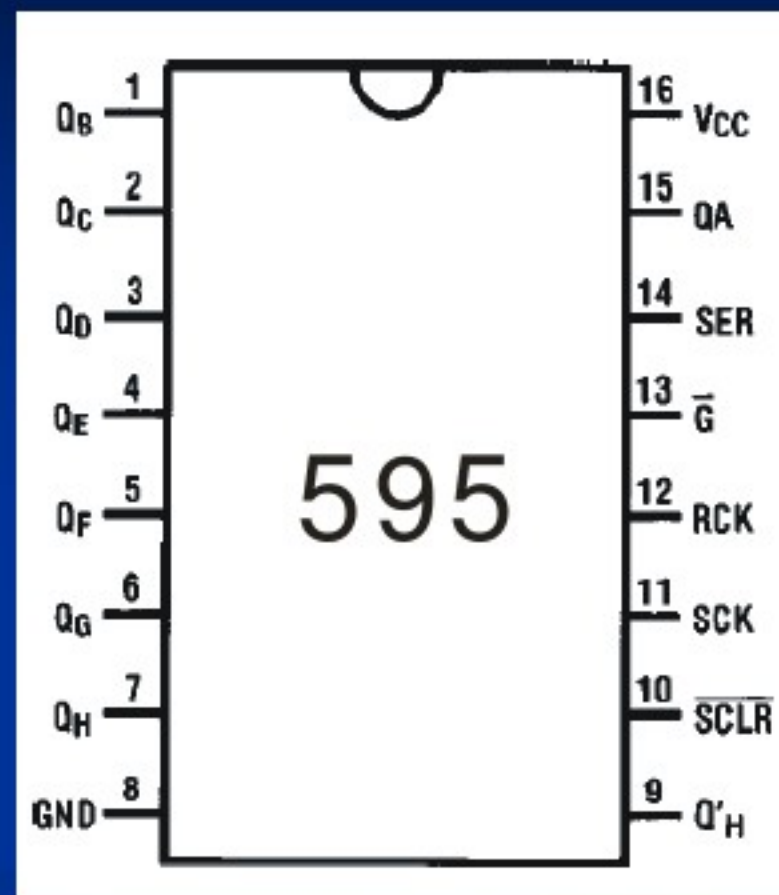
G (13脚): OUTPUT ENABLE 高电平时禁止输出(高阻态)。

74HC595优点:

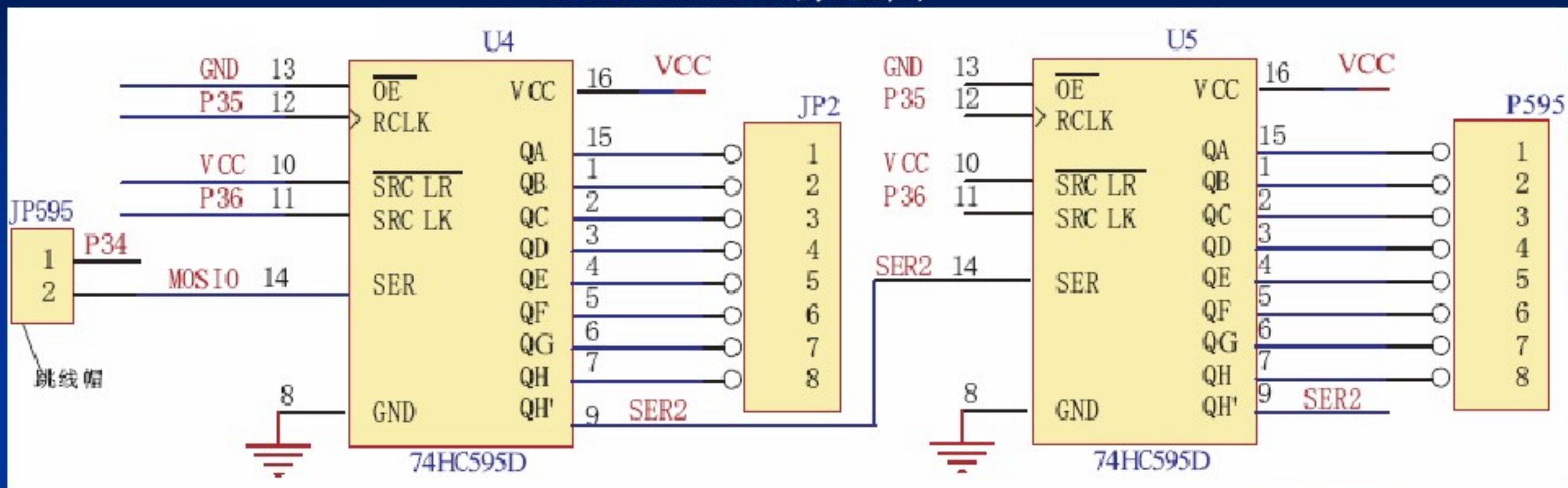
具有数据存储寄存器，在移位的过程中，输出端的数据可以保持不变，抗干扰强。这在串行速度慢的场合很有用处，数码管没有闪烁感。

注意:

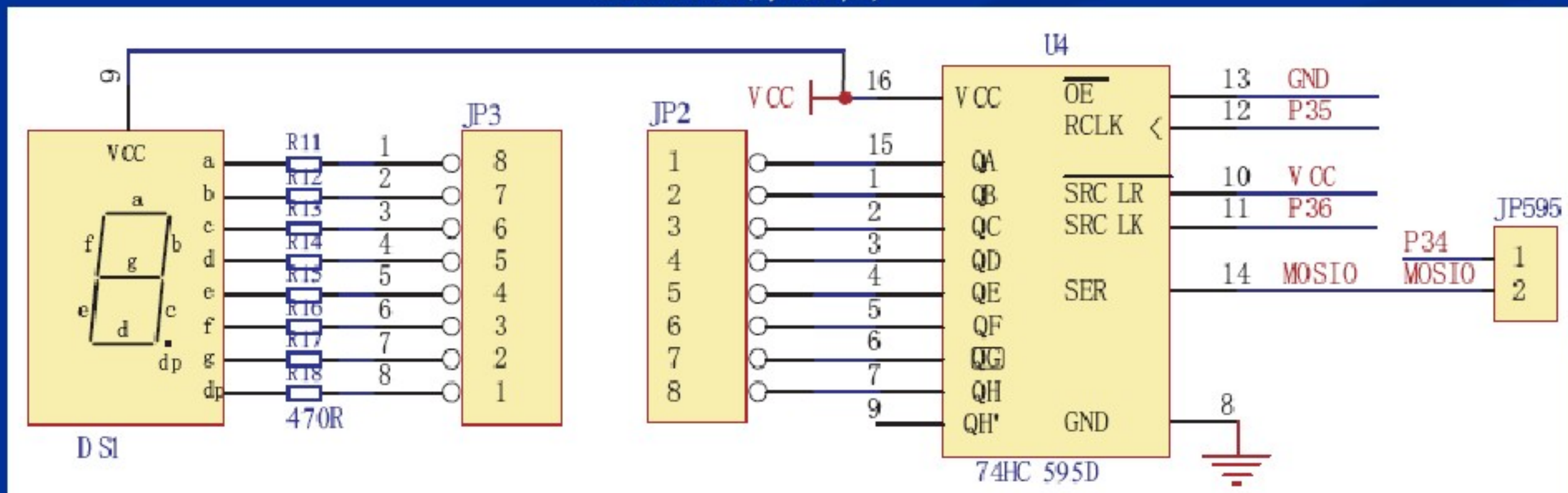
74164和74595功能相仿，都是8位串行输入转并行输出移位寄存器。74164的驱动电流(25mA)比74595(35mA)的要小，14脚封装，体积也小一些。



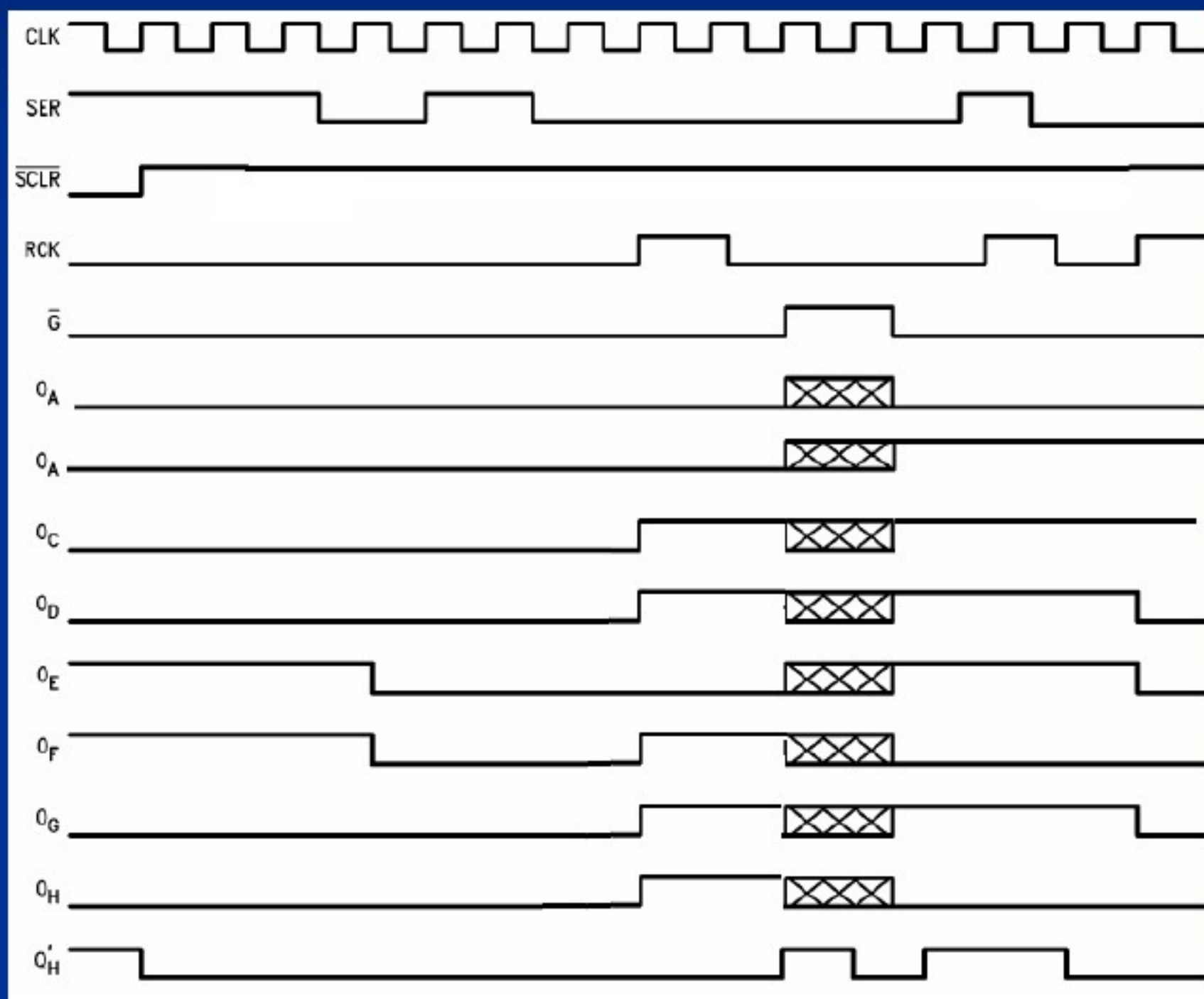
HC6800EM3 原理图



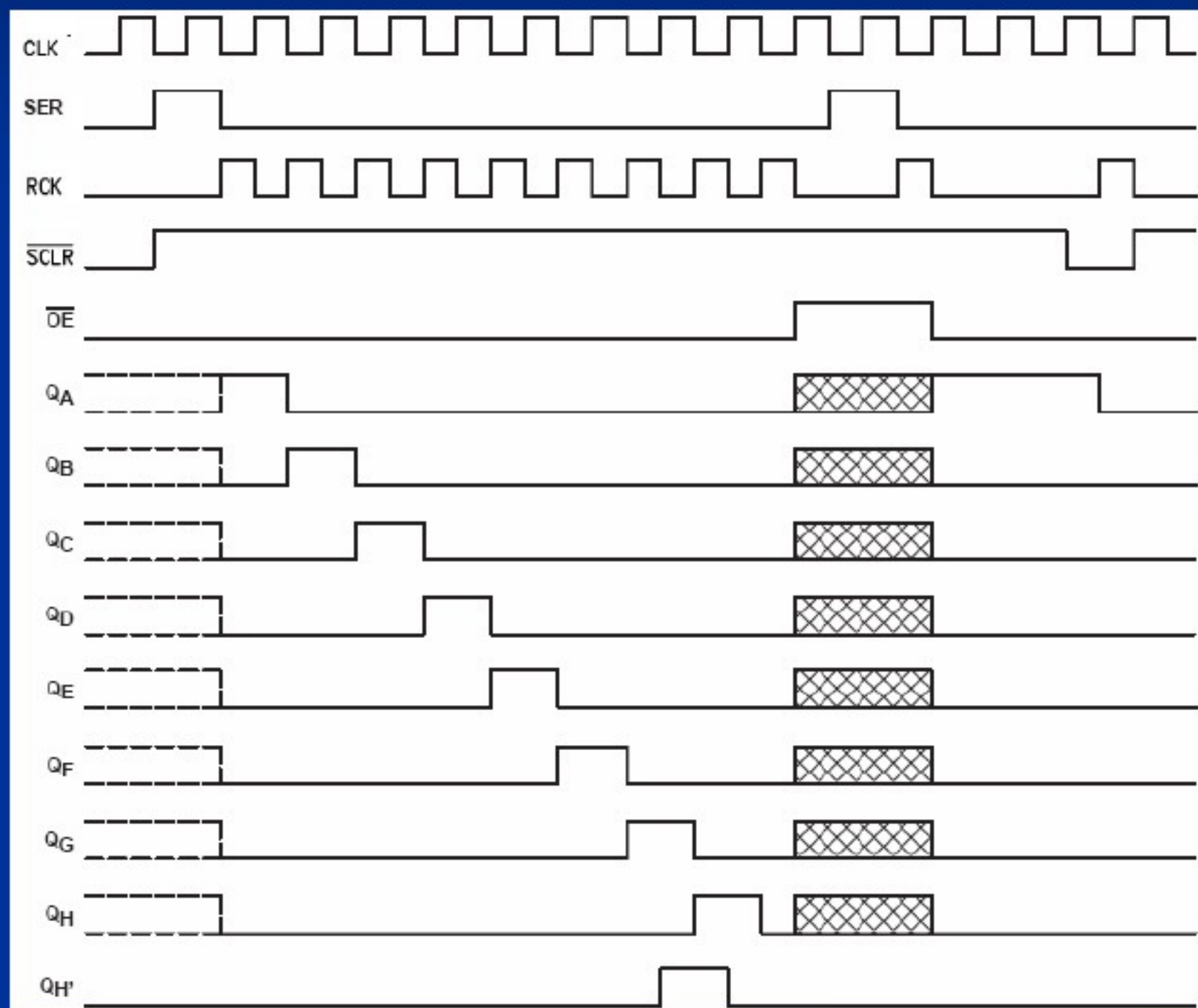
HC6800原理图



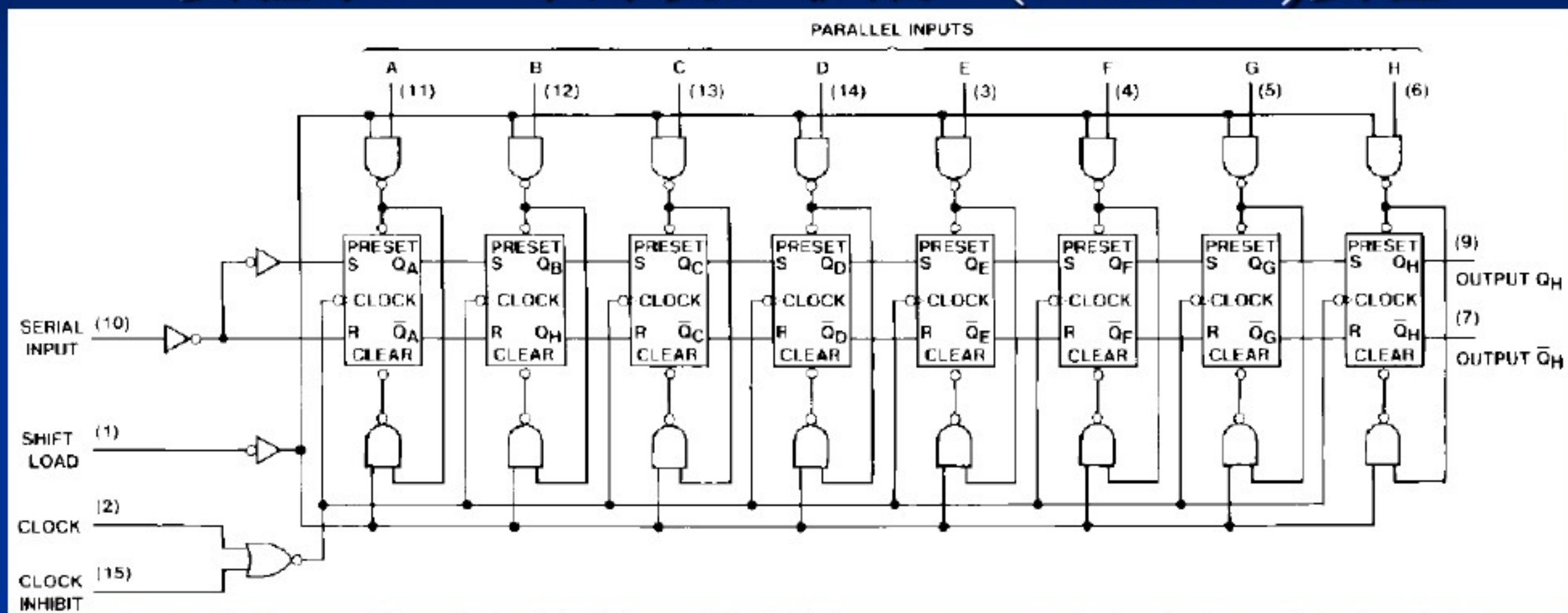
时序图1



时序图2



实验十三 单片机IO扩展二(74HC165)实验



74HC165是一款高速CMOS器件，它的引脚兼容低功耗肖特基TTL（LSTTL）系列。它是8位并行读取或串行输入移位寄存器，可在末级得到互斥的串行输出（QH和 \bar{Q}_H ）。

SERIAL INPUT为串行输入端，将数据串行进入到寄存器，在每个时钟脉冲的上升沿向右移动一位（QA → QB → QC，等等）。利用这种特性，只要把QH输出接到下一级的SERIAL INPUT输入，即可实现并转串扩展。

SHIFT LOAD为低电平时，将A到H口并行数据同时传到寄存器中。

SHIFT LOAD为高电平时，并行数据不能传入寄存器中。

74HC165的时钟输入是一个“或非门”结构，CLOCK和CLOCK INHIBIT功能是等效的可以互换使用，都是上升沿有效。

当CLOCK和CLOCK INHIBIT有一个为低电平，并且SHIFT LOAD为高电平时，另一个时钟可以输入。当CLOCK和CLOCK INHIBIT有一个为高电平时，另一个时钟被禁止。

74HC165 参数

74HC165 基本参数

电压 2.0~6.0V

驱动电流 +/-5.2 mA

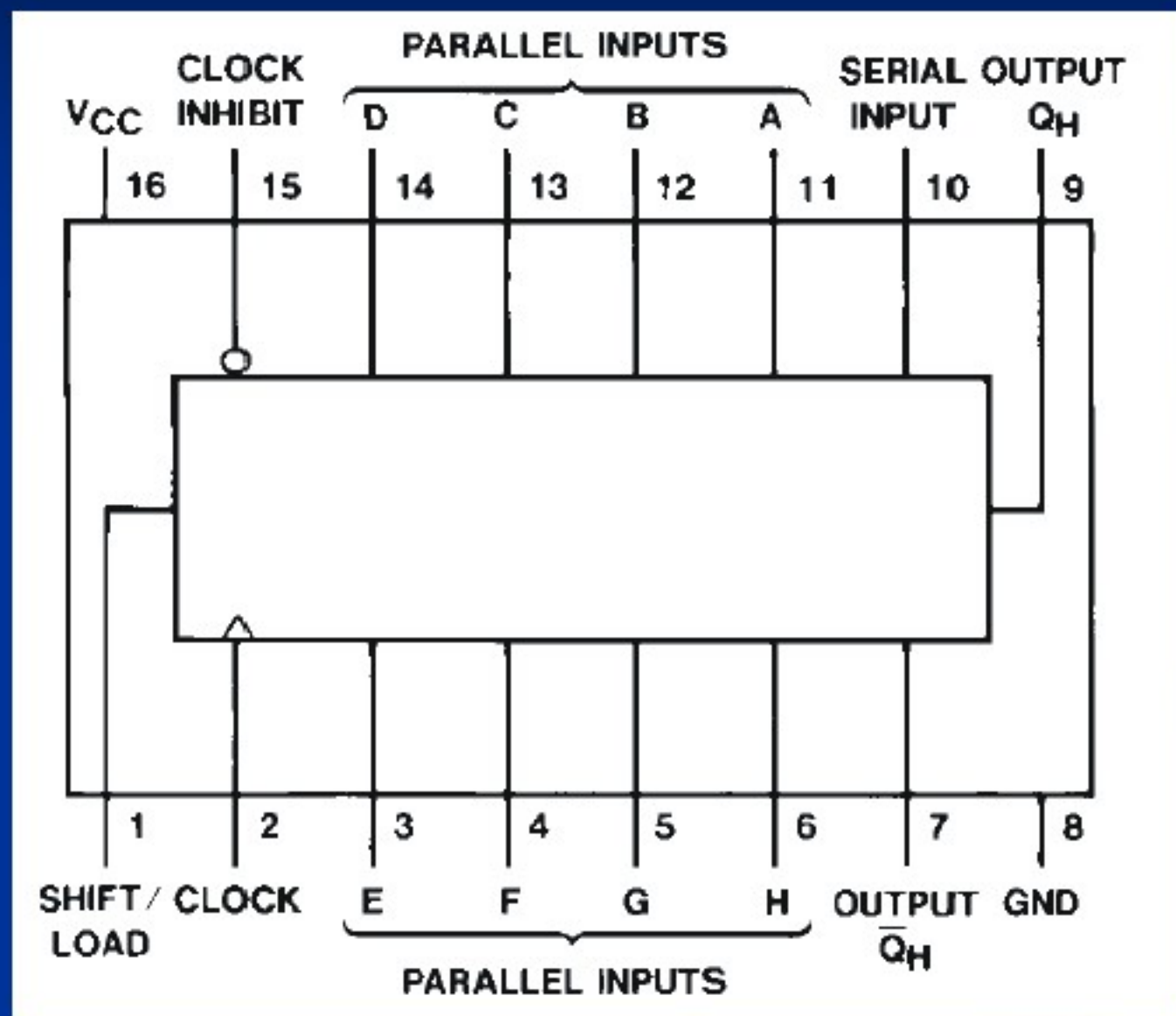
传输延迟 16 ns@5V

74HC165 其他特性

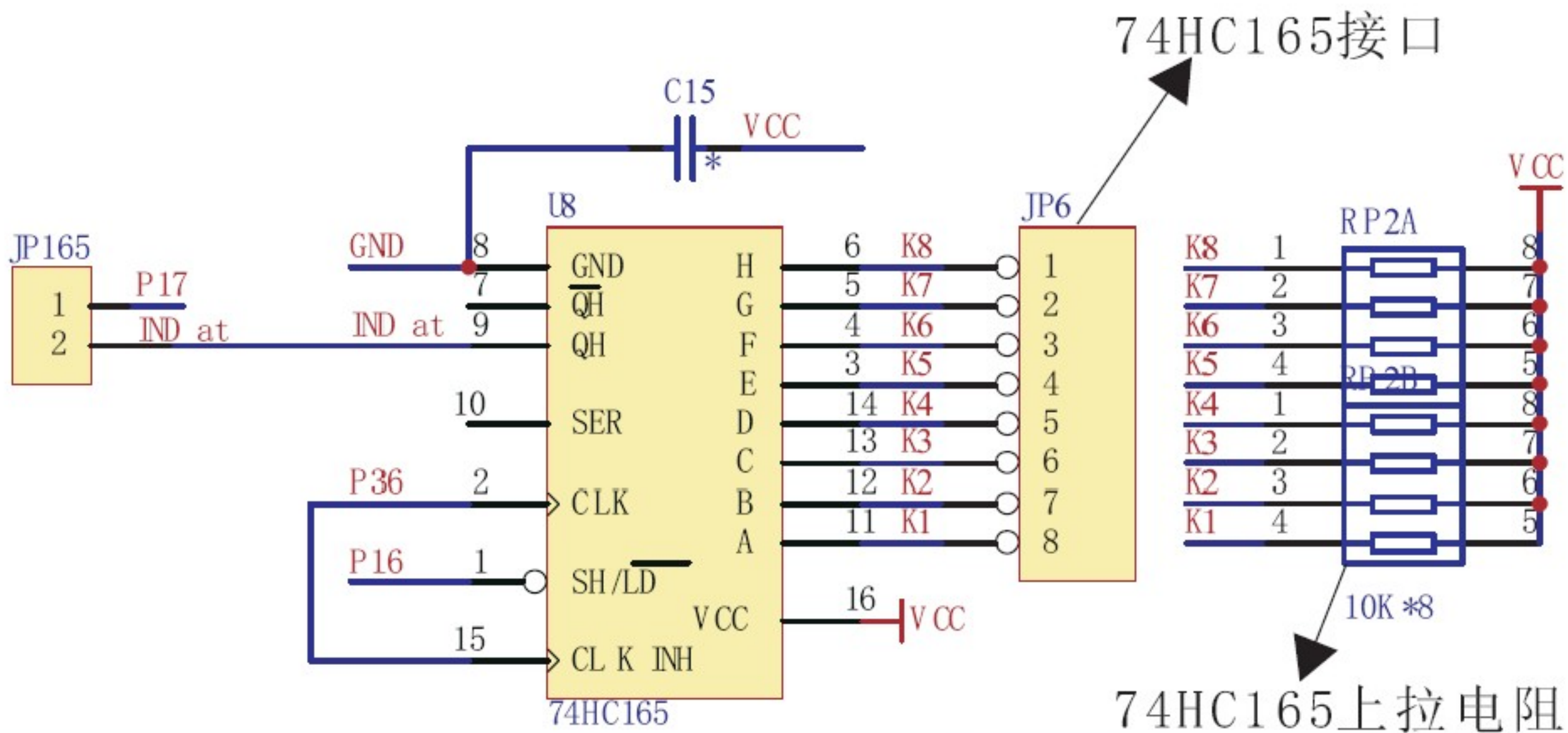
最高频率 56 MHz

逻辑电平 CMOS

功耗考量 低功耗或电池供电应用



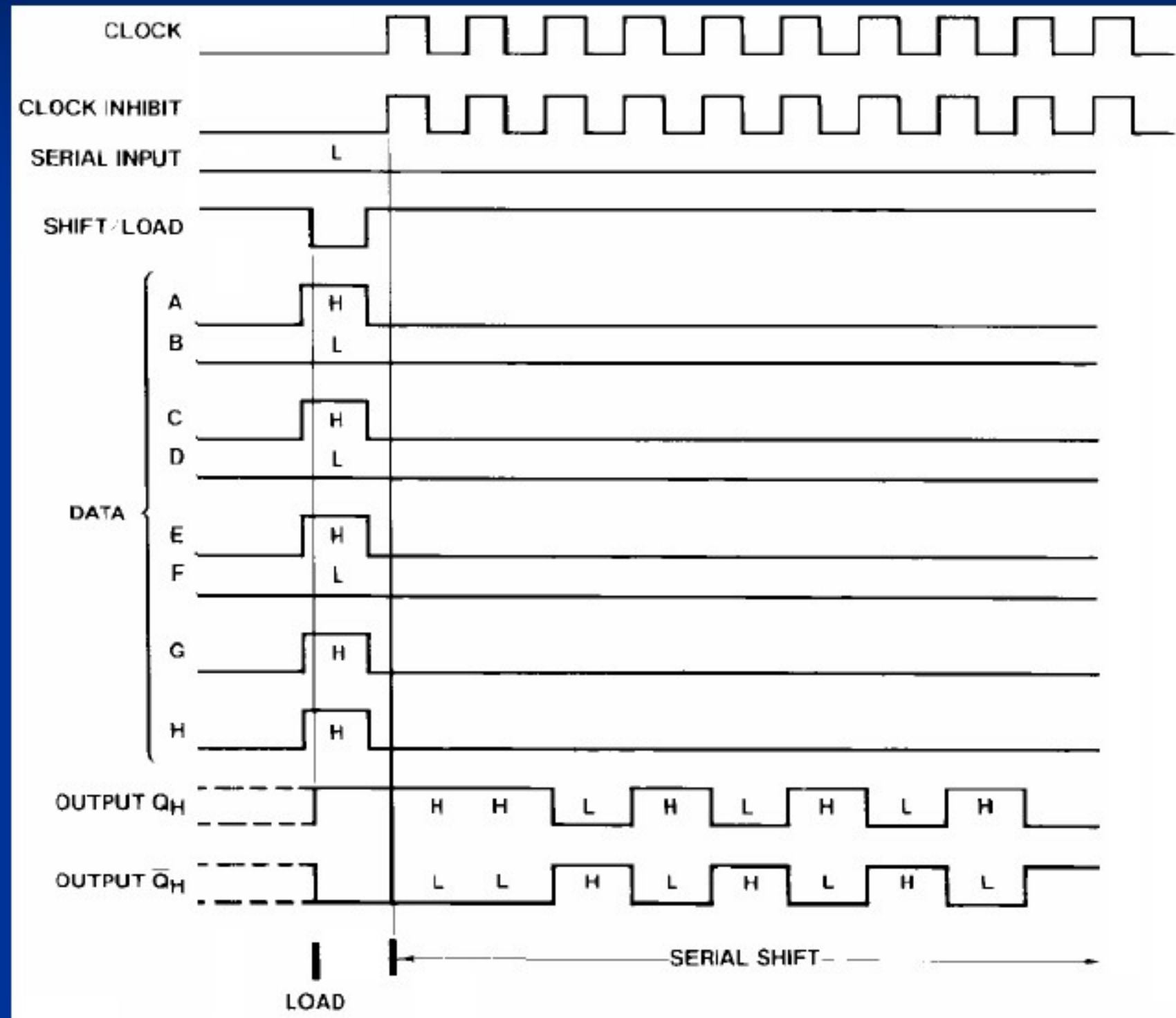
开发试验仪的原理图



相关知识:

- data:** 可寻址片内ram, 固定指前面0x00-0x7f的128个字节RAM, 可以用acc直接读写的, 速度最快, 生成的代码也最小。
- bdata:** 可位寻址的片内ram
- idata :** 可寻址片内ram, 允许访问全部内部ram。固定指前面0x00-0xff的256个字节的RAM,其中前128和data的128完全相同
- pdata:** 分页寻址片外ram (MOVX @R0) (256 BYTE/页)
- xdata:** 可寻址片外ram (64k 地址范围FFFFH), 一般指外部0x0000-0xffff空间, 用DPTR访问
- code:** 程序存储区 (64k 地址范围),对应MOVC @DPTR

时序图

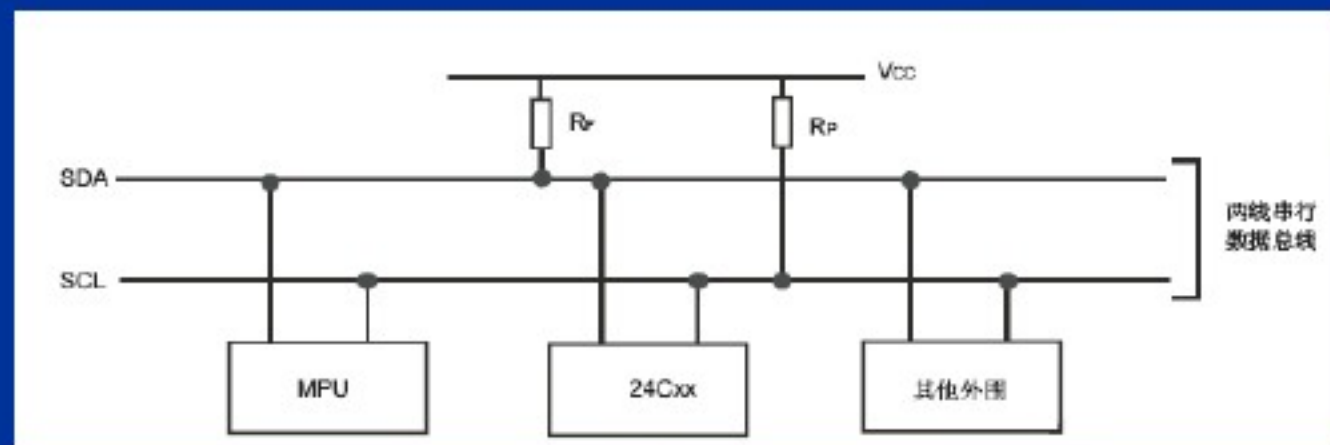


实验十四 I²C总线和EEPROM实验

I²C (Inter - Integrated Circuit) 总线是一种由PHILIPS公司开发的两线式串行总线，用于连接微控制器及其外围设备。I²C总线产生于80年代，最初为音频和视频设备开发。

I²C总线是由**数据线SDA**和**时钟线SCL**构成的串行总线，可发送和接收数据。在CPU与被控IC之间、IC与IC之间进行双向传送，最高传送速率100Kbps。被控制电路均并联在这条总线上，每个电路和模块都有唯一的地址，就像电话机一样只有拨通各自的号码才能工作。I²C总线上每一模块电路既是主控器（或被控器），又是发送器（或接收器）。

CPU发出的控制信号分为地址码和控制码两部分，地址码用来要控制的器件。控制码决定控制的内容。这样，各控制电路虽然挂在同一条总线上，却彼此独立，互不相关。



I²C总线最主要的优点：

1. I²C总线占用的空间非常小，减少了电路板的空间和芯片管脚的数量，降低了互联成本，总线的长度可高达25英尺。
2. 支持多主控 (Multimastering)，其中任何能够进行发送和接收的设备都可以成为主总线。一个主控能够控制信号的传输和时钟频率。当然，在任何时间点上只能有一个主控。

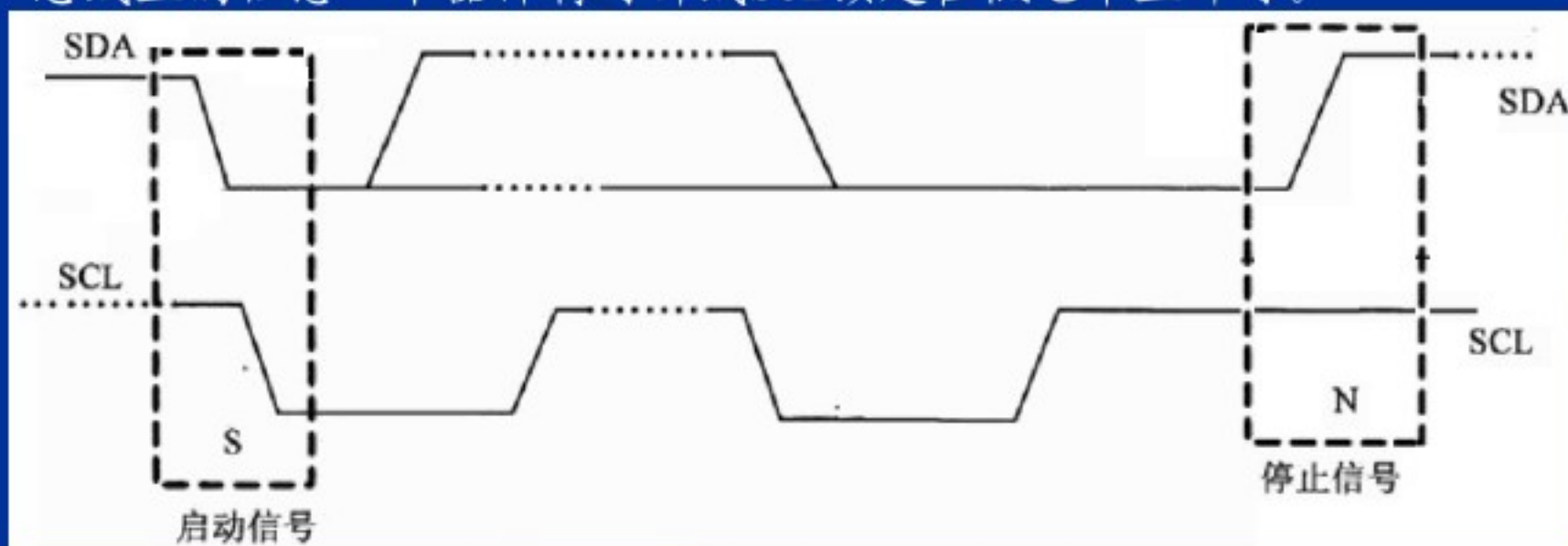
分析I²C总线的时序和几种信号状态

1. **总线空闲状态**: I²C总线的SDA和SCL两条信号线同时处于高电平时, 规定为总线的空闲状态。此时各个器件的输出级场效应管均处在截止状态, 即释放总线, 由两条信号线各自的上拉电阻把电平拉高。

2. **启动信号**: 在时钟线SCL保持高电平期间, 数据线SDA上的电平被拉低(即负跳变), 定义为I²C总线总线的启动信号, 它标志着一次数据传输的开始。

3. **停止信号**: 在时钟线SCL保持高电平期间, 数据线SDA被释放, 使得SDA返回高电平(即正跳变), 称为I²C总线的停止信号, 它标志着一次数据传输的终止。也是一种电平跳变时序信号, 而不是一个电平信号, 停止信号也是由主控器主动建立的, 建立该信号之后, I²C总线将返回空闲状态。

4. **总线封锁状态**: 在特殊情况下, 如果需要禁止所有发生在I²C总线上的通信活动, 封锁或关闭总线是一种可行途径, 只要挂接于该总线上的任意一个器件将时钟线SCL锁定在低电平上即可。

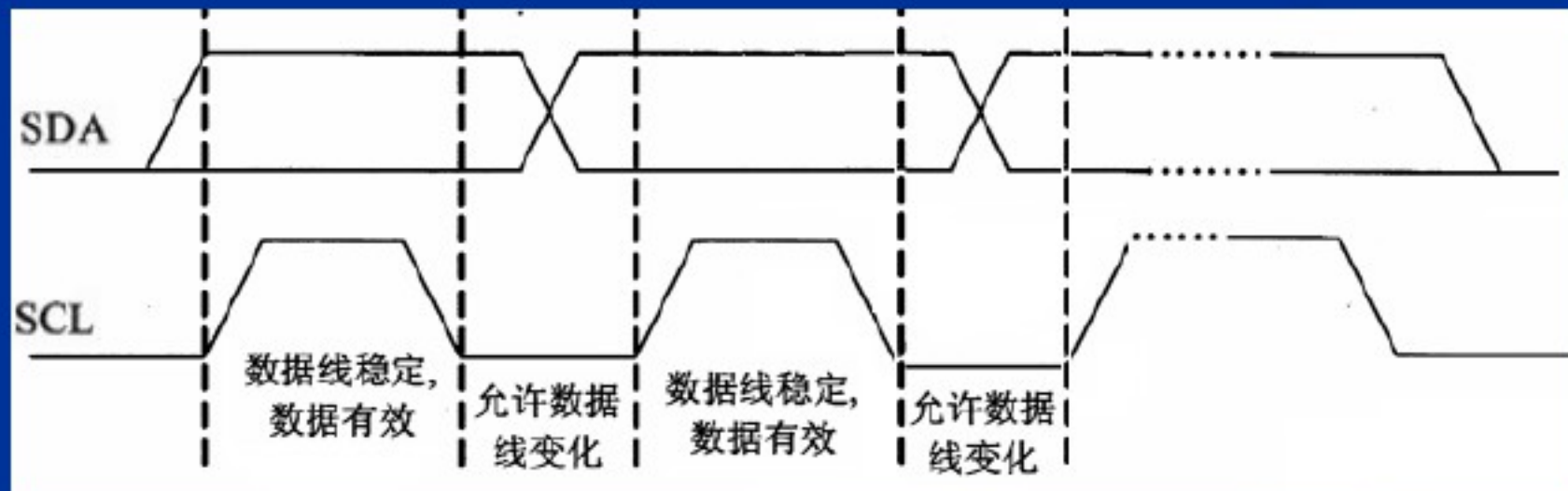


5. 数据位传送。

在I²C总线上传送的每一位数据都有一个时钟脉冲相对应（或同步控制），即在SCL串行时钟的配合下，在SDA数据线上逐位地串行传送每一位数据。

进行数据传送时，在SCL呈现高电平期间，SDA上的电平必须保持稳定，

只有在SCL为低电平期间，才允许SDA上的电平改变状态。



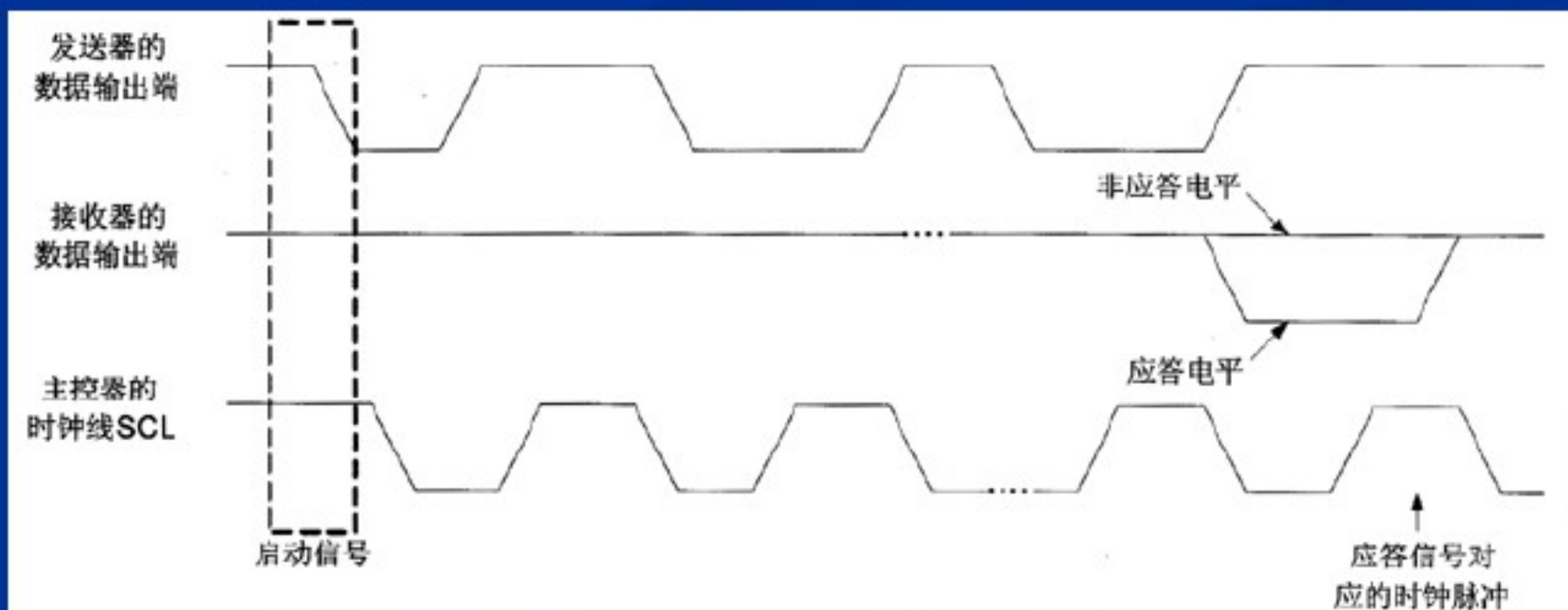
6. 应答信号。

I²C总线上的所有数据都是以8位字节传送的，发送器每发送一个字节后，就在时钟脉冲9期间释放数据线，由接收器反馈一个应答信号。

应答信号为低电平时，规定为有效应答位（ACK简称应答位），表示接收器已经成功地接收了该字节；应答信号为高电平时，规定为非应答位（NACK），一般表示接收器接收该字节没有成功。

对于反馈有效应答位ACK的要求是，接收器在第9个时钟脉冲之前的低电平期间将SDA线拉低，并且确保在该时钟的高电平期间为稳定的低电平。

如果接收器是主控器，则在它收到最后一个字节后，发送一个NACK信号，以通知被控发送器结束数据发送，并释放SDA线，以便主控接收器发送一个停止信号P，

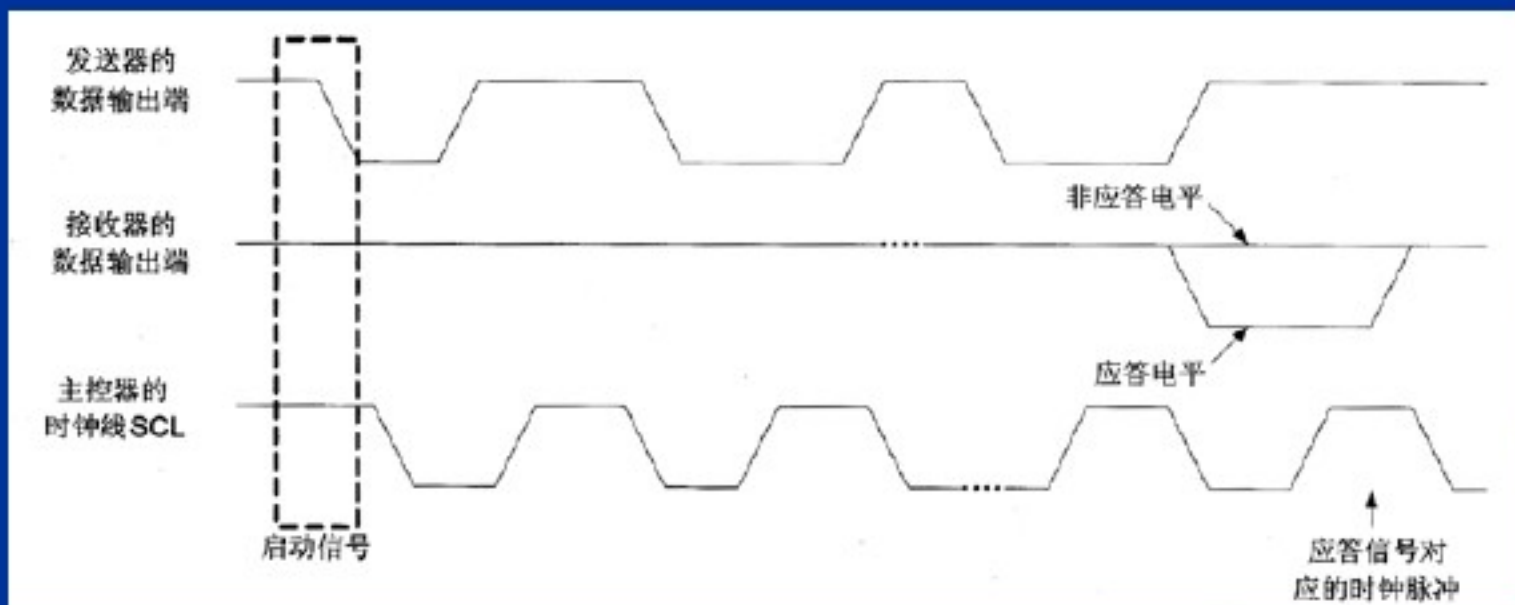


7. 插入等待时间。

如果被控器需要延迟下一个数据字节开始传送的时间，则可以通过把时钟线SCL电平拉低并且保持，使主控制器进入等待状态。

一旦被控器释放时钟线，数据传输就得以继续下去，这样就使得被控器得到足够时间转移已经收到的数据字节，或者准备好即将发送的数据字节。

带有CPU的被控器在对收到的地址字节做出应答之后，需要一定的时间去执行中断服务子程序，来分析或比较地址码，其间就把SCL线钳位在低电平上，直到处理妥当后才释放SCL线，进而使主控制器继续后续数据字节的发送。



8. 重新启动信号。

在主控制器控制总线期间完成了一次数据通信（发送或接收）之后，如果想继续占用总线再进行一次数据通信（发送或接收），而又不释放总线，就需要利用重新启动Sr信号时序。

重新启动信号Sr既作为前一次数据传输的结束，又作为后一次数据传输的开始。利用重新启动信号的优点是，在前后两次通信之间主控制器不需要释放总线，这样就不会丢失总线的控制权，即不让其他主器件节点抢占总线。

24C02的知识

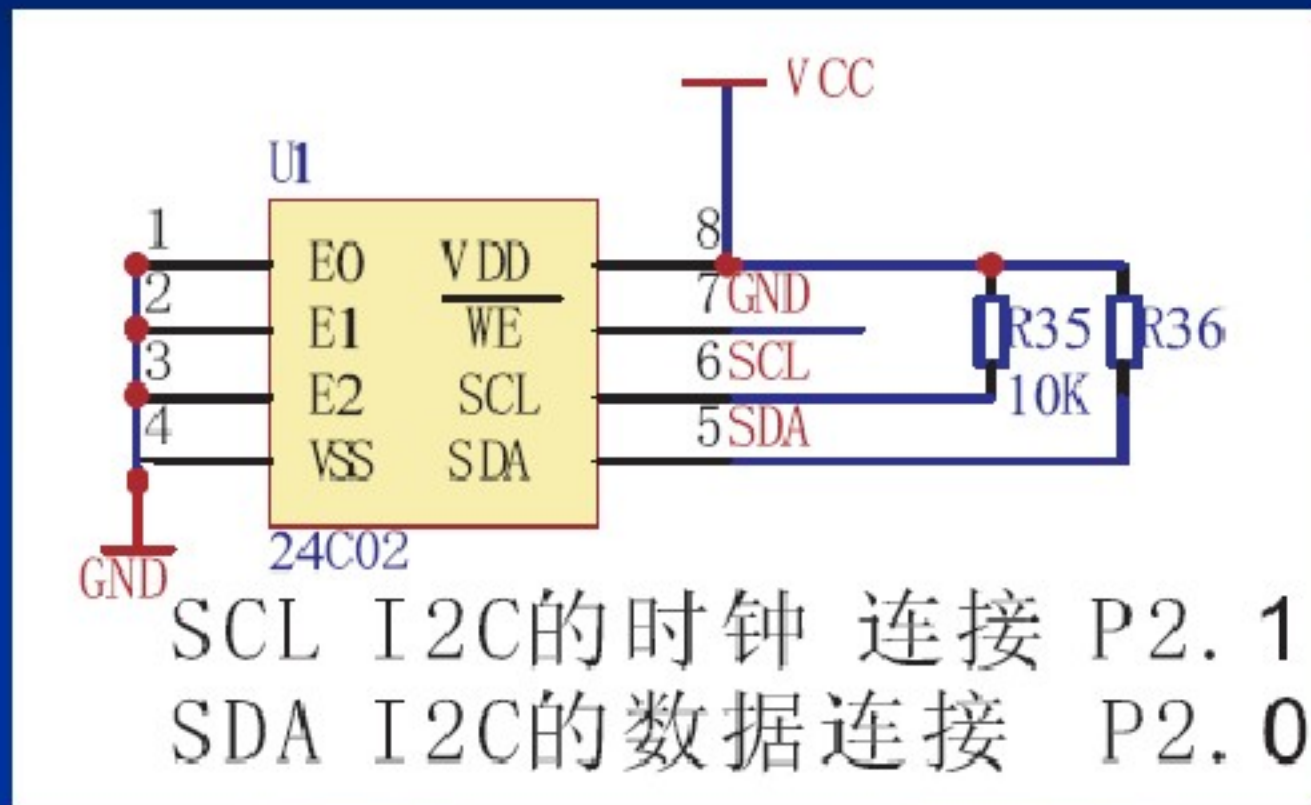
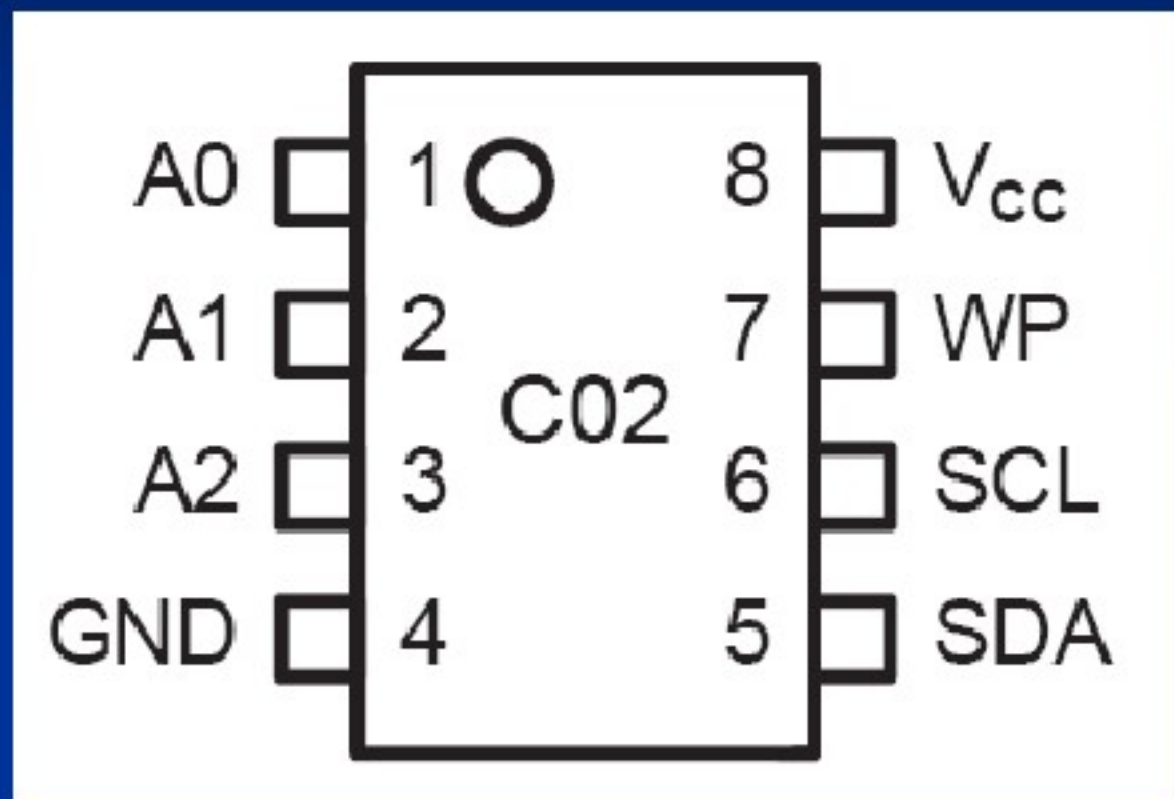
一、特性

1. 24C02是电可擦除PROM，所有通常也叫它EEPROM（**Electrically Erasable Programmable Read-Only Memory**），采用256*8的组织结构。
2. 具有两线串行I2C接口。
3. 电压可允许低至1.8V，待机电流和工作电流分别为1 μ A和1mA
4. 24C02以8个字节为一页，而24C04/08/16以16个字节为一页，24C32/64以32个字节为一页。
5. 自动递增地址
6. 读取操作可以按字节，随机和序列读取数据。
7. 擦写寿命可达100万次。
8. 数据保存100年
9. 内部写周期最大5MS

二、应用范围

智能仪表 工业控制 家用电器 计算机笔记本电脑 汽车电子 通信设备

三、管脚定义及原理图



A0、A1和A2是器件地址输入引脚 24C02 使用这几个硬件作为硬件地址，总线上可以同时并联8个器件。

SDA 串行地址和数据输入/输出，SDA是双向串行数据传输引脚，漏极开路，许外接上拉电阻到VCC

SCL 串行时钟输入，SCL同步数据传输，上升沿数据写入，下降沿数据读取。

WP 写保护。WP 引脚提供硬件数据保护，当WP接地时，允许数据正常读写操作。当WP接VCC时，写保护，只读。

四、24C02读写操作说明

当I2C的起始条件建立后，芯片就能够进行读写操作了。这时需要把芯片的器件地址传给对应的芯片。EEPROM要求是一个8位的器件地址。对于串行的EEPROM高四位的地址都是一样的。器件地址信息的LSB为读/写操作的选择位，1为读操作，0为写操作。

当单片机把地址信息发出去后，I2C总线所有器件将接收这个地址信息，这时24C02比较与器件的地址是否一致，一致24C02将输出应答0。如果不一致，则返回到待机状态。待机状态具有低功耗待机的特点，条件为：（1）电源上电，（2）接收停止条件及完成任何内部操作。

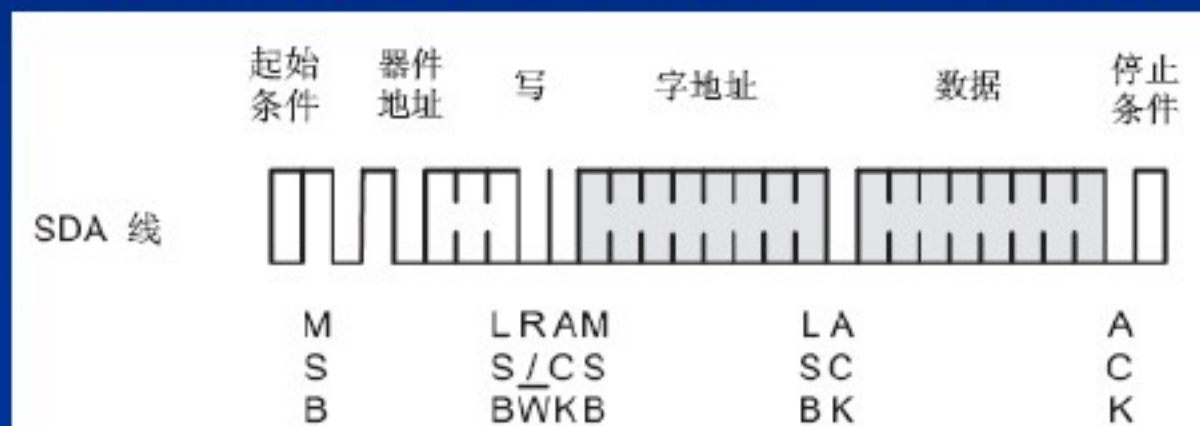
24C02/32/64	1	0	1	0	A2	A1	A0	R/W	MSB: Most Significant Bit最高有效位
	MSB				LSB				
24C04	1	0	1	0	A2	A1	P0	R/W	LSB: Least Significant Bit最低有效位
24C08	1	0	1	0	A2	P1	P0	R/W	
24C16	1	0	1	0	P2	P1	P0	R/W	

12. 存储结构

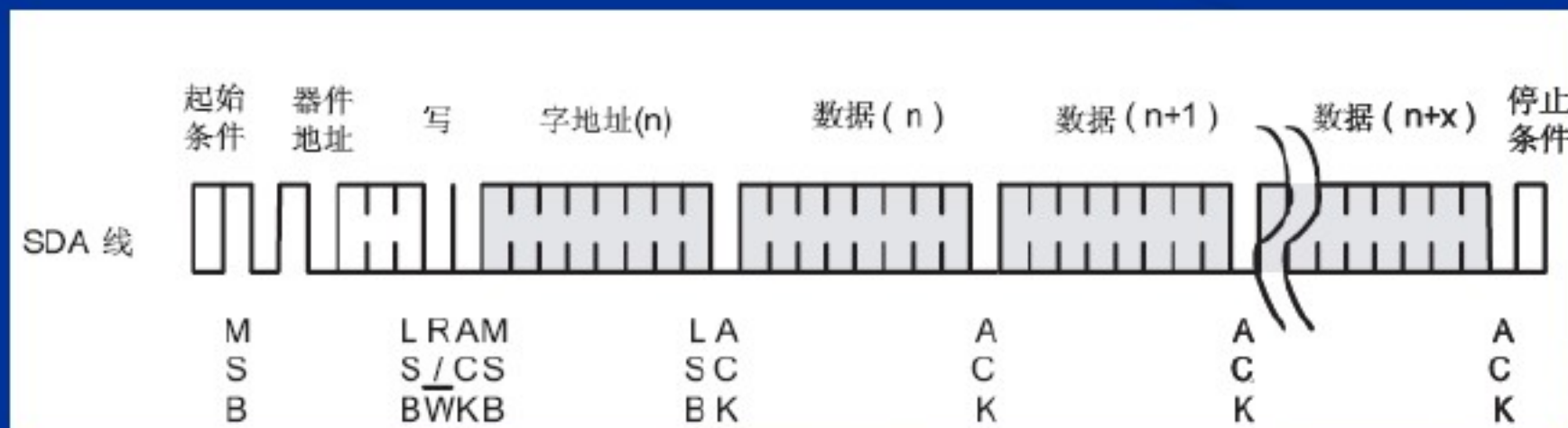
器件	总容量 (位)	总页数	字节/页	字地址长度
24C02	2K	32	8	8位
24C04	4K	32	16	9位
24C08	8K	64	16	10位
24C16	16K	128	16	11位
24C32	32K	128	32	12位
24C64	64K	256	32	13位

写操作

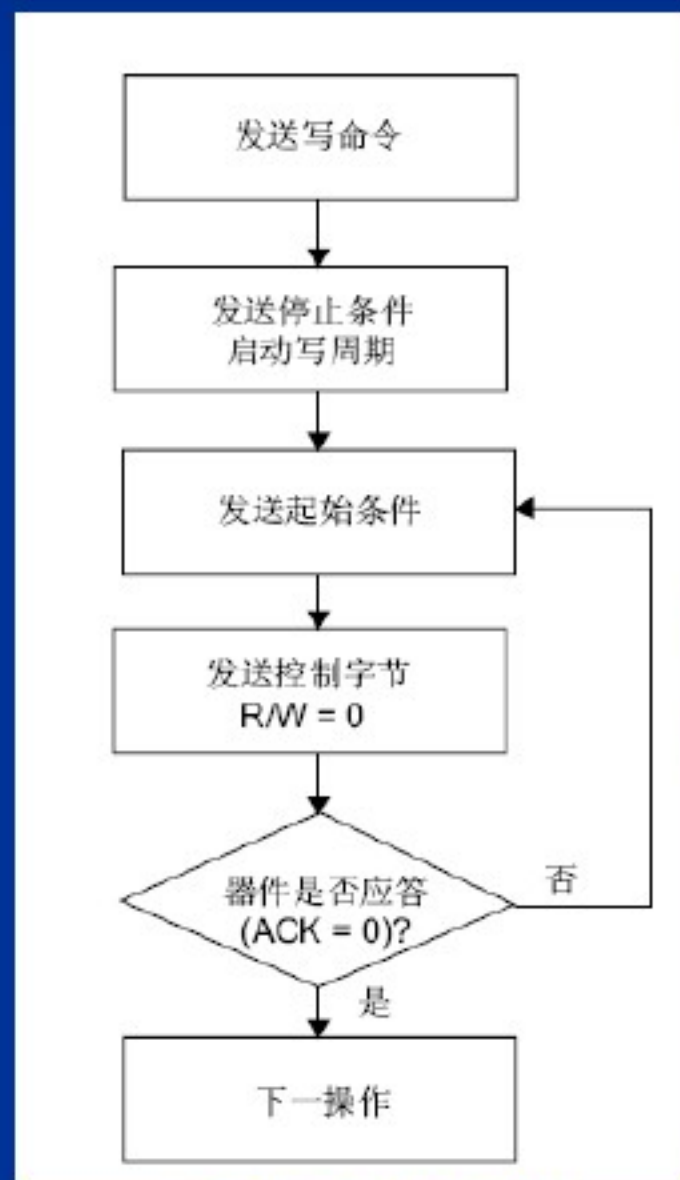
1. 字节写：当I2C的起始条件建立后，使芯片能够进行读写操作，给24C02发一个器件地址，接收到ACK应答为0后，再发8位存储器的地址，接收到这个地址后，EEPROM应答为0，再发8位的数据，EEPROM回答为0，接着由主控制器发终止条件来终止写操作。



2. 页写：24C02器件按 8字节/页执行页写，也写的初始化和字节写相同，只是主器件不会在第一个数据后发送停止条件，而是在EEPROM的ACK以后，接着发剩下的7个数据。EEPROM收到每个数据后都应答0最后仍需要主器件发送停止条件，终止写操作。接收到每个数据后，字地址的低3位内部会自动加1，高位地址位不变，维持在当页内。当内部产生的字地址达到该页的边界地址时，随后的数据将写入该页的页首，先前的字节将会被覆盖。



3. 应答查询: 一旦内部写周期启动, EEPROM输入是无效的, 此时可启动应答查询, 发送起始条件和器件地址(读写位为期望的操作)。只有内部写周期完成, EEPROM才应答0。之后可以继续读写操作。



应答查询流程

读操作

读操作和写操作初始化相同，只是器件地址中的读/写选择位应为1。读操作有三种方式：
当前地址读，随机读和顺序读。

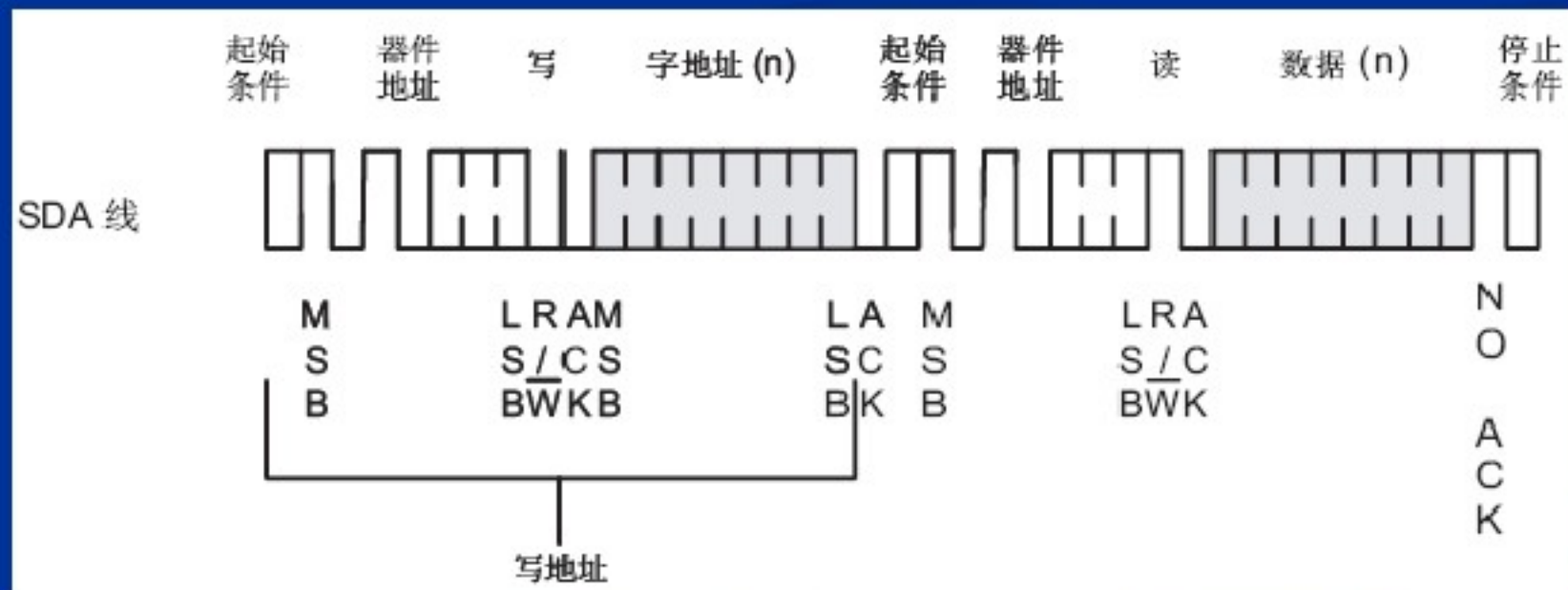
1. 当前地址读

内部地址计数器保存着上次访问时最后一个地址加1的值，只要芯片有电，该地址就一直保存。当读到最后一页的最后一个字节，地址会回转到0，当读到某页尾的最后一个字节，地址会回到该页的首字节。



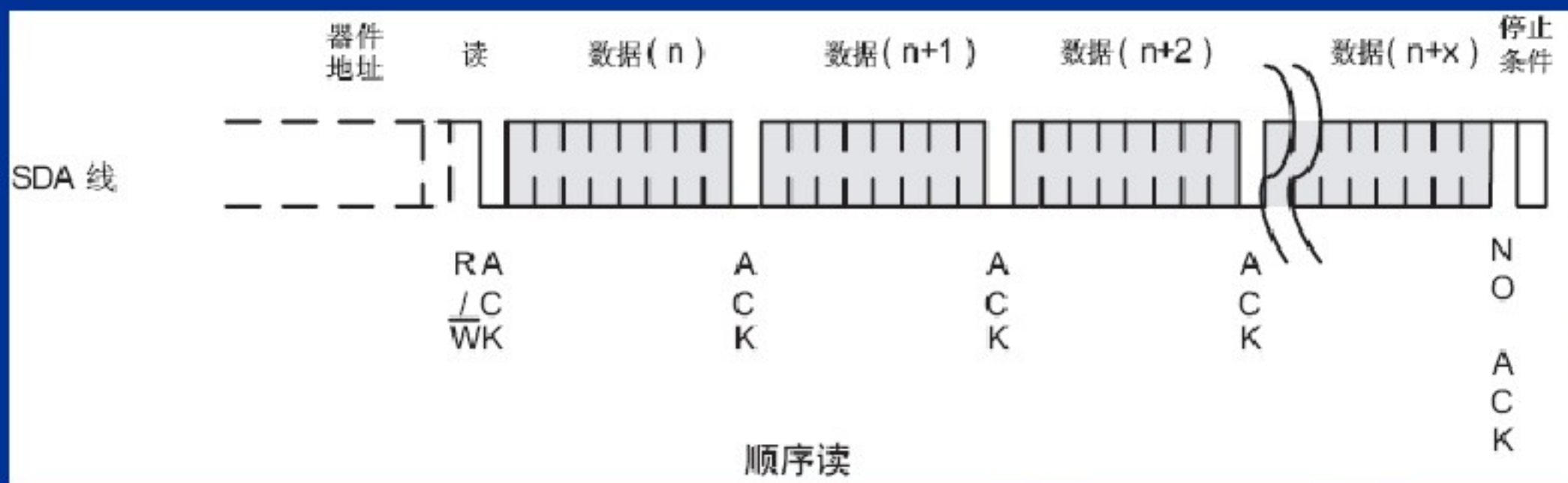
2. 随机读

需先写一个目标字地址，一旦EEPROM接收到器件地址和字地址并应答了ACK，主器件就产生一个重复的起始条件。然后，主器件发送器件地址（读写位选择1），EEPROM应答ACK，并随时钟送出数据。主器件无需应答0，但需发送停止条件。



3. 顺序读

顺序读可以通过“当前地址读”或“随机读”启动。主器件接收到一个数据后，应答ACK。只要EEPROM接收到ACK，将自动增加字地址并继续随时钟发送后面的数据。若达到存储器地址末尾，地址自动回转到0，仍可继续顺序读取数据。



实验十五 模数 (AD) /数模 (DA) 转换

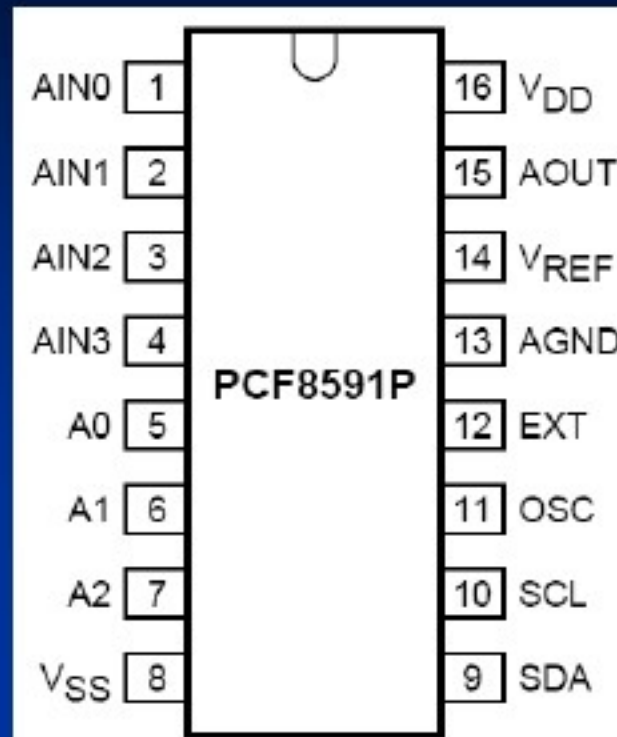
PCF8591介绍

PCF8591是一款单电源、低功耗8位COMS型A/D、D/A转换芯片，它具有4路模拟量输入通道、一路模拟量输出通道和1个I2C总线接口。由于是I2C总线接口，所以在不增加任何硬件的情况下同一条I2C总线最多可以挂接8个PCF8591。PCF8591由于其使用的简单方便和集成度高，在单片机应用系统中得到了广泛的应用。

PCF8591具有以下特点：

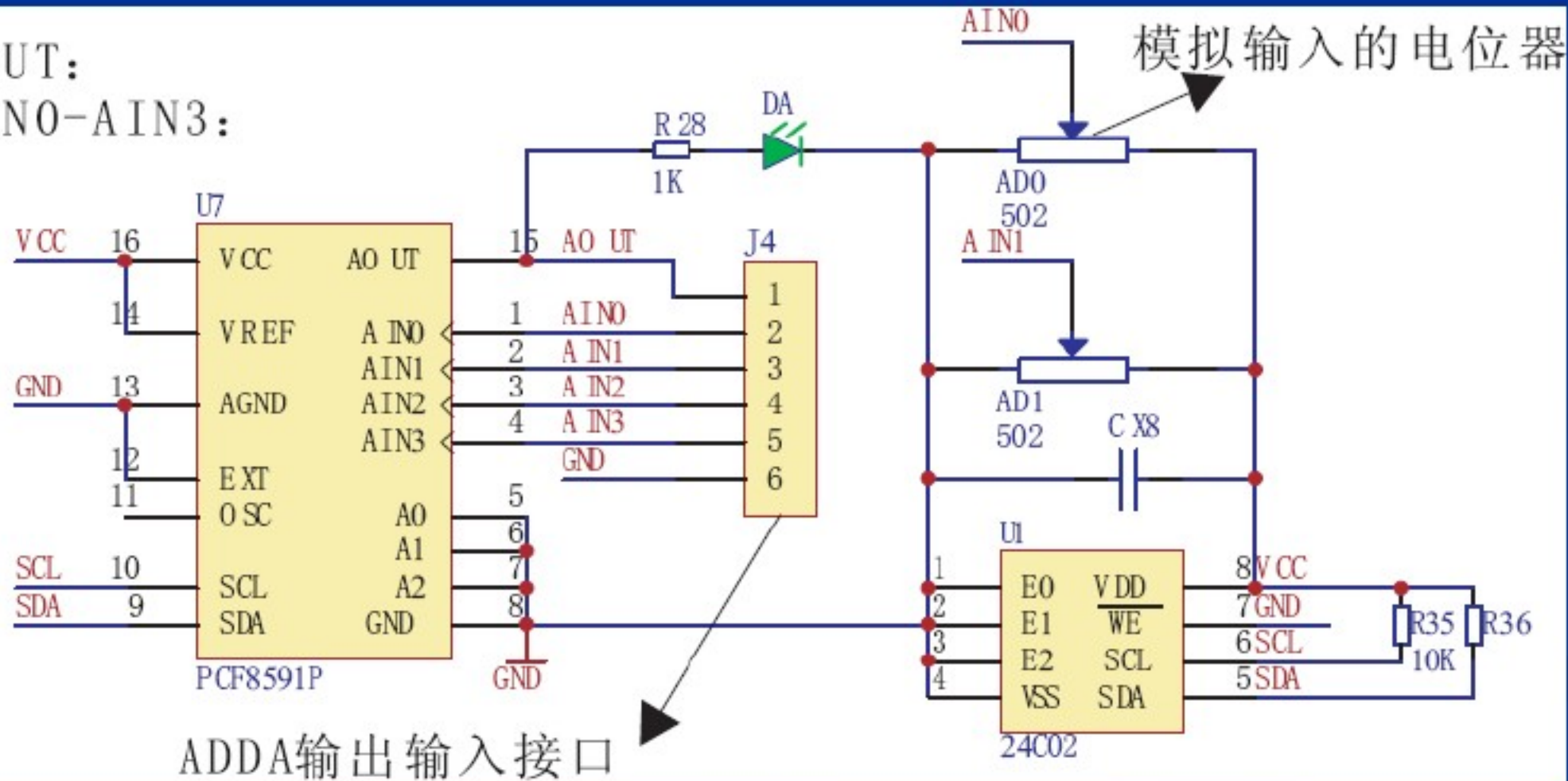
1. 单电源供电；
2. 正常工作电源电压范围为2.5V~6V；
3. 通过I2C总线完成数据的输入/输出；
4. 器件地址由3个地址引脚决定(对应芯片的A0、A1和A2三个地址)；
5. 采样频率由I2C总线传输速率决定；
6. 4路模拟量输入可编程为单端输入或差分输入；
7. 可配置转换通道号自动增加功能；
8. 片上跟踪保持功能；
9. 8位逐次逼近A/D转换器；
10. 带有一路8位模拟量输出的乘法D/A转换器。

管脚定义和原理图



AIN0~AIN3: 模拟输入 (A/D转换)。
 AOUT: 模拟输出 (D/A转换)。
 A0-A2: 硬件设备地址。
 GND: 电源负极地。
 VREF: 参考电压输入。
 EXT: 振荡器输入时, 内部/外部的切换开关。
 OSC: 振荡器输入/输出。
 SCL: I2C BUS 时钟输入。
 SDA: I2C BUS 数据输入/输出。
 AGND: 模拟地, 模拟信号和基准电源的参考地。

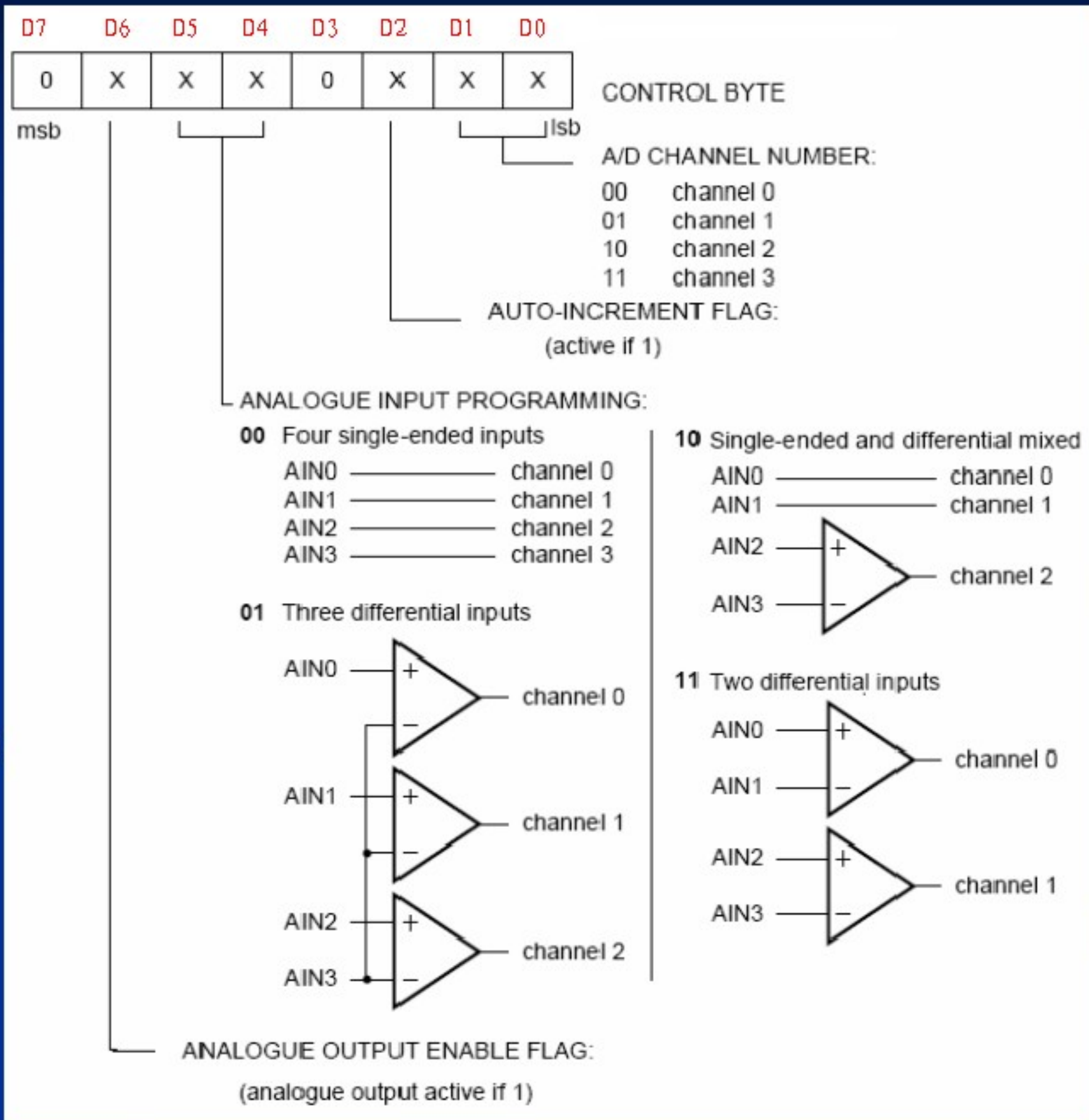
AOUT:
 AIN0-AIN3:



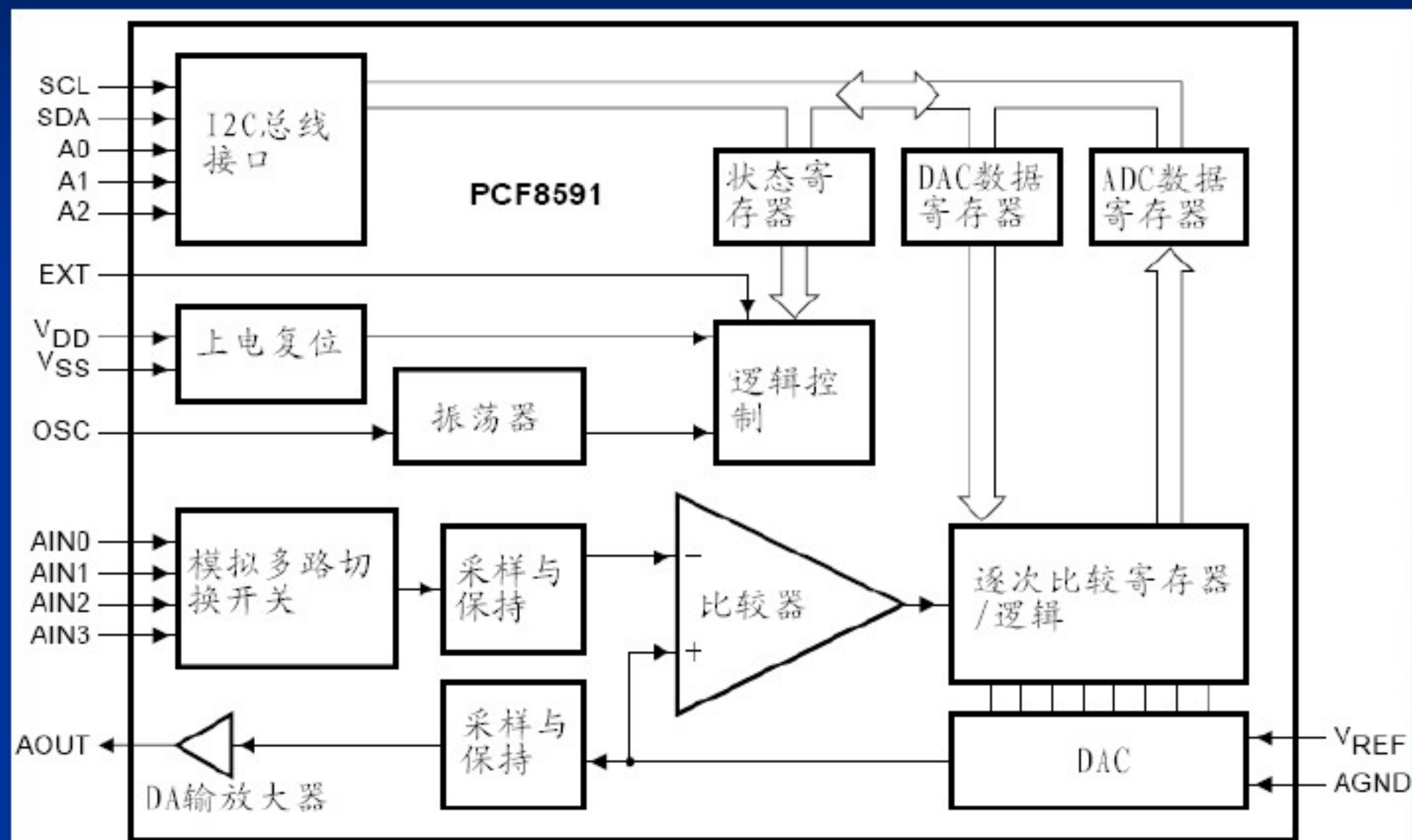


飞利浦规定PCF8591的器件地址高四位为1001，引脚地址为A2A1A0 这个由用户自己选择，因此I2C系统中最多可接 $2^3 = 8$ 个PCF8591。地址的最后一位为方向位R/W 1为读操作 0 为写操作。

- D0 D1 用于4个通道设置
- D2 自动增益选择 (有效位为1)
- D5 D4 模拟量输入选择
 - 00 4路单端输入
 - 01 3路差分输入
 - 10 单端与差分输入
 - 11 2路差分输入



内部结构图

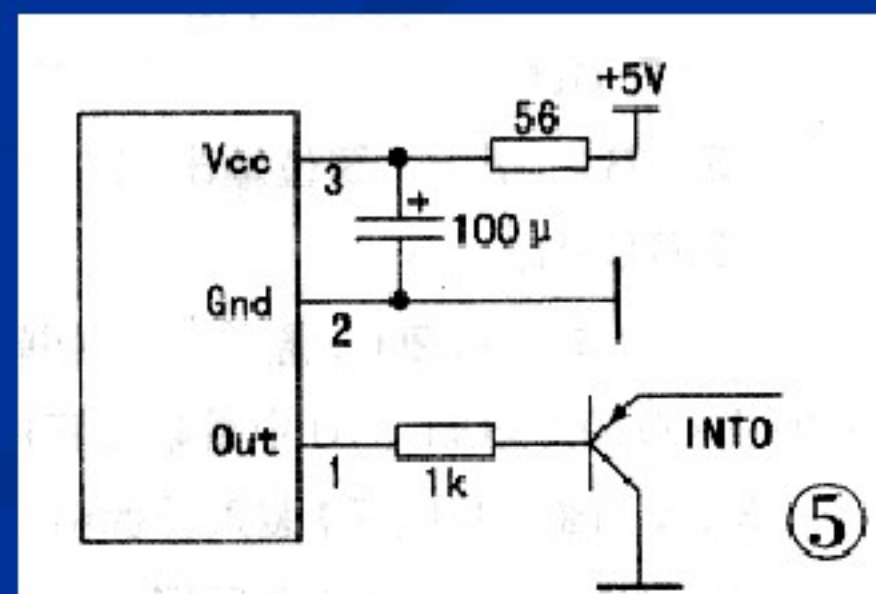


AD转换器利用逐次逼近转变技术,在AD转换周期期间, D/A转换器和一个高增益比较器被暂时使用。一旦一个转换周期被触发,被设置的那个通道,将采集到的电压存储在芯片中,并将其转换成8位的二进制代码,将转换结果被保存在模数数据寄存器中,等待传输。如果自动增益标志被设置,下一个通道即被选择。

实验十六 红外遥控实验

人的眼睛能看到的可见光按波长从长到短排列，依次为红、橙、黄、绿、青、蓝、紫。其中红光的波长范围为 $0.620.76 \mu\text{m}$ ；紫光的波长范围为 $0.38 \sim 0.46 \mu\text{m}$ 。比紫光波长还短的光叫紫外线，比红光波长还长的光叫红外线。红外线遥控就是利用波长为 $0.76 \sim 1.5 \mu\text{m}$ 之间的近红外线来传送控制信号的。

接收部分的红外接收管是一种光敏二极管。在实际应用中要给红外接收二极管加反向偏压，它才能正常工作，亦即红外接收二极管在电路中应用时是反向运用，这样才能获得较高的灵敏度。红外接收二极管一般有圆形和方形两种。



由于红外发光二极管的发射功率一般都较小（100mW左右），所以红外接收二极管接收到的信号比较微弱，因此就要增加高增益放大电路。前些年常用 μ PC1373H、CX20106A等红外接收专用放大电路。

最近几年不论是业余制作还是正式产品，大多都采用成品红外接收头。成品红外接收头的封装大致有两种：一种采用铁皮屏蔽；一种是塑料封装。均有三只引脚，即电源正（VDD）、电源负（GND）和数据输出（VO或OUT）。

红外接收头的引脚排列因型号不同而不尽相同，可参考厂家的使用说明。成品红外接收头的优点是不需要复杂的调试和外壳屏蔽，使用起来如同一只三极管，非常方便。

但在使用时注意成品红外接收头的载波频率。红外遥控常用的载波频率为38kHz，这是由发射端所使用的455kHz晶振来决定的。在发射端要对晶振进行整数分频，分频系数一般取12，所以 $455\text{kHz} \div 12 \approx 37.9\text{kHz} \approx 38\text{kHz}$ 。也有一些遥控系统采用36kHz、40kHz、56kHz等，一般由发射端晶振的振荡频率来决定。

不同型号的红外接头互换要注意的问题:

接收头中心频率应与遥控发射器频率相同: 大多数红外接收头解调中心频率为 38 kHz, 但也有一些接收头中心频率为 32.7 kHz、36.7 kHz、37.9 kHz、40 kHz、56 kHz, 如果发射频率与接收频率相差正负为 1 kHz, 大多可以正常遥控, 相差 2 kHz 以上则会出现遥控不灵或距离短现象, 此时可通过更换遥控发射器的晶体振荡器来解决。

常见为 455 kHz 晶振 (对应发射频率 38 kHz), 其他有 429 kHz、432 kHz、445 kHz、465 kHz、480 kHz 等型号的晶振, 相对应的发射频率分别为 36 kHz、36.7 kHz、37.9 kHz、38 kHz、40 kHz。

信号极性：大多数遥控接收头输出信号极性为负极性，即输出端在无信号时为高电位（一般为4.8~5.0v），接收到信号后信号输出端电压下降。但也有少数接收头输出信号为正极性，如松下TC-2180、M25等彩电的红外接收头，若用常见型号接收头直接代换，则无法遥控，对于此种情况可在信号输出端加接反相器解决。

2. 引脚顺序。

遥控接收头引脚顺序有如下几种：（接收面左侧起）

- ①地、信号输出、电源；
- ②信号输出、地、电源；
- ③地、电源、信号输出

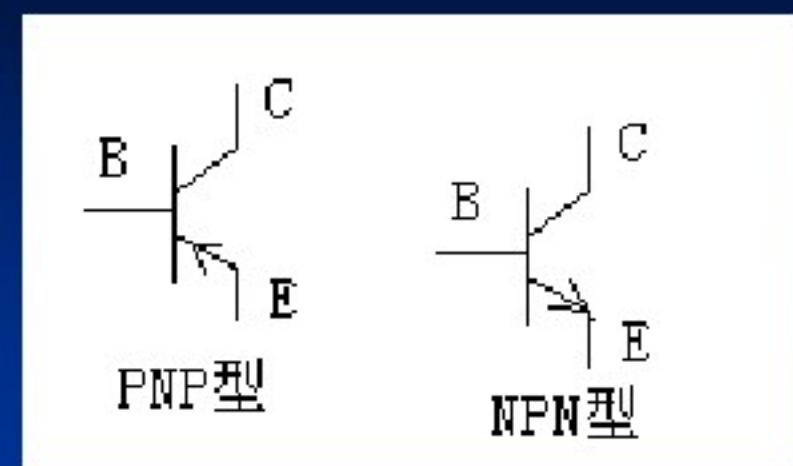
代换时应仔细区分。对于引脚顺序相同的可直接按顺序接入，如引脚顺序不对，则可用细导线引接。注意地线与电源线切不可接反，否则通电后接收立即损坏。

三、接线演示

看视频图像

四、知识点

三极管：半导体三极管也称为晶体三极管，可以说它是电子电路中最重要器件。它最主要的功能是电流放大和开关作用。三极管顾名思义具有三个电极



8550	PNP	40V	1500mA	1000mW	200MHz	放大倍数40-140
8050	NPN	25V	700mA	200mW	150MHz	放大倍数30-100
9011	NPN	30V	30mA	400mW	150MHz	放大倍数20-80
9012	PNP	50V	500mA	600mW	低频管	放大倍数30-90
9013	NPN	20V	625mA	500mW	低频管	放大倍数40-110
9014	NPN	45V	100mA	450mW	150MHz	放大倍数20-90

