微机实验平台简介

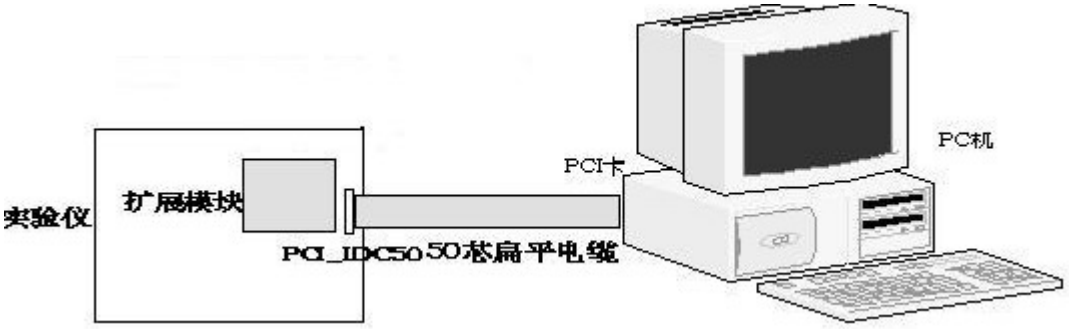
一、实验平台背景介绍

ISA 总线：ISA 总线是在 PC 总线的基础上，由 Intel 公司、IEEE 和 EISA 集团共同推出的一种总线标准。它是一种低速的 16 位系统总线，也可以用作 8 位总线，工作频率为 8MHz。有关 ISA 总线的相关知识可以参考《微机原理与接口技术》一书中的相关内容。

PCI 总线：PCI 是作为一种先进的局部总线提出来的，它已经成为了连接外部设备的主要总线。PCI 总线宽度为 32 位，并可升级为 64 位，而 ISA 总线只是 8 位/16 位总线。它的工作频率通常是 33MHz，支持即插即用。是目前最主流的一种接口类型。

二、实验相关器件介绍

1、实验连接简图：



2、PCI 扩展卡电路：

PCI 扩展卡的功能是将 PCI 总线信号变换成 ISA 的信号，并且经过驱动隔离以后经扁平电缆送到实验台，以完成 PC 程序对实验台单元电路的控制。PCI 总线扩展卡的电路框图如图：

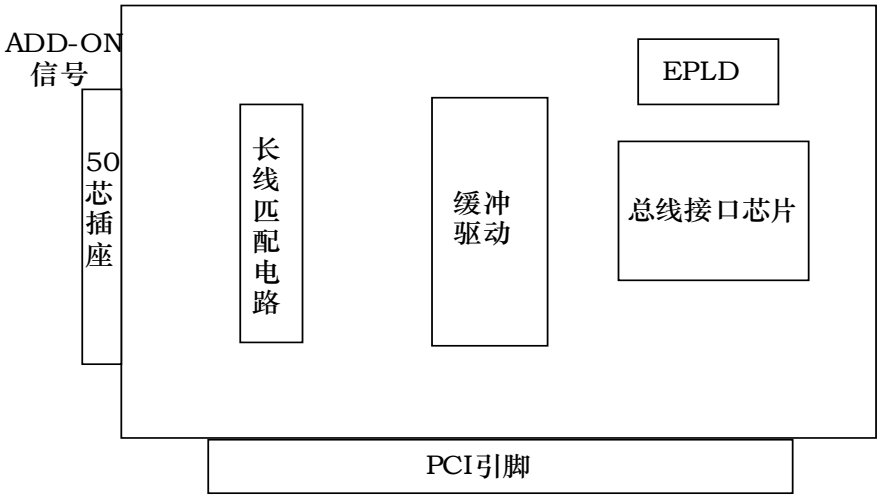


图 1－7 PCI扩展卡框图

PCI 扩展卡已经插在了机箱里面，通过扁平电缆与实验台相连。

3、PCI 总线接口芯片 CH365

CH365 是一个连接 PCI 总线的通用接口芯片，支持 I/O 端口映射、存储器映射、扩展

ROM 以及中断。CH365 将 32 位高速 PCI 总线转换为简便易用的类似于 ISA 总线的 8 位主动并行接口，用于制作低成本的基于 PCI 总线的计算机板卡，以及将原先基于 ISA 总线的板卡升级到 PCI 总线上。PCI 总线与其它主流总线相比，速度更快，实时性更好，可控性更佳，所以 CH365 适用于高速实时的 I/O 控制卡、通讯接口卡、数据采集卡、电子盘、扩展 ROM 卡等。

空间映射：

PC 机中包括三种空间：存储器空间、I/O 空间、配置空间。存储器空间主要包括内存、显存、扩展 ROM、设备缓冲区等，一般用于存放大量数据和进行数据块交换。I/O 空间主要包括设备的控制寄存器和状态寄存器，一般用于控制和查询设备的工作状态以及少量数据的交换。配置空间主要用于向系统提供设备自身的基本信息，并接受系统对设备全局状态的控制和查询。 为了避免地址冲突，PCI 总线要求各个设备所占用的地址能够重定位。重定位是由设备的配置空间的基址寄存器实现的，通常情况下，各个设备的基址寄存器总是被 BIOS 或者操作系统分配为不同的基址，从而将各个设备分别映射到不同的地址范围。在需要时，应用程序也可以自行修改基址。 CH365 的存储器空间占用 32K 字节，偏移地址是 0000H～7FFFH，可以全部提供给外部设备使用，实际地址是存储器基址加上偏移地址。CH365 的 I/O 空间占用 256 字节，去掉 CH365 自用寄存器，还可以提供 240 字节给外部设备使用，偏移地址是 00H～EFH，实际地址是 I/O 基址加上偏移地址。

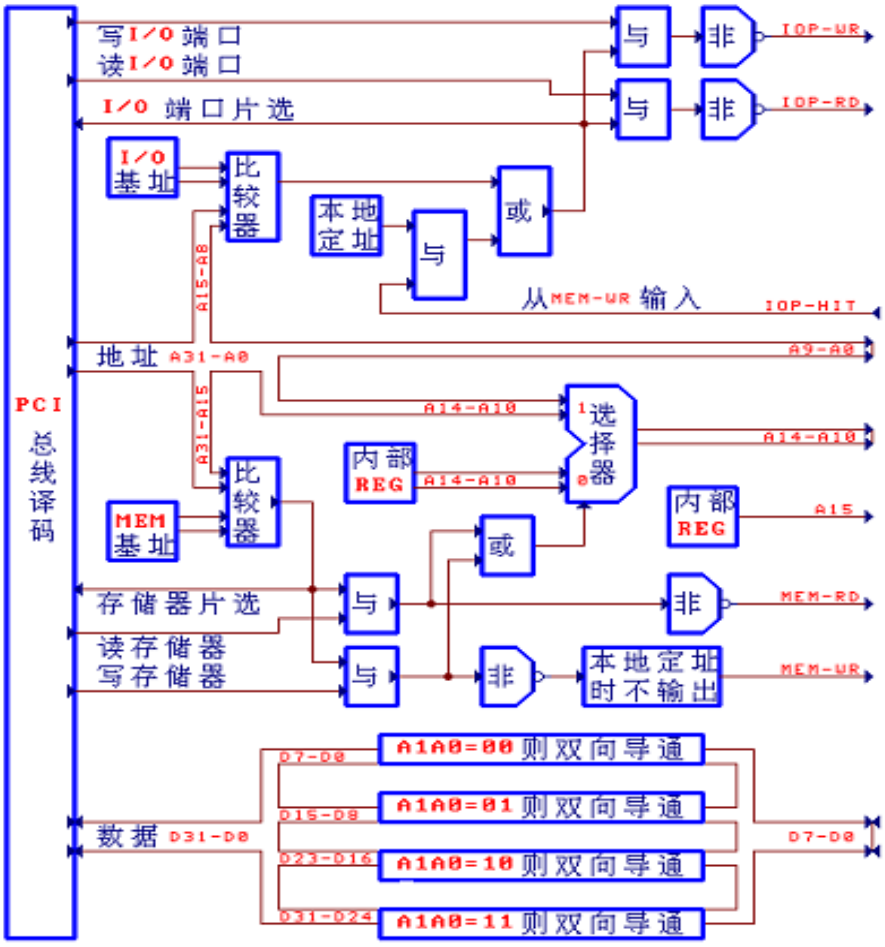
内部结构和信号线：

下图是 CH365 内部的主要结构。CH365 对 PCI 总线的各种信号进行译码后，产生内部数据总线 D31～D0、内部地址总线 A31～A0、读 I/O 端口信号、写 I/O 端口信号、读存储器信号、写存储器信号等。图中已经标明了各个信号的传输方向。

结构图右侧的信号是指 CH365 提供给本地端的各个外部引脚。 地址线 A15～A0 用于提供相对于基址的偏移地址，数据总线 D7～D0 在读操作时用于输入数据，在写操作时用于输出数据。IOP_RD 用于提供 I/O 读选通脉冲信号，IOP_WR 用于提供 I/O 写选通脉冲信号，MEM_RD 用于提供存储器读选通脉冲信号，MEM_WR 用于提供存储器写选通脉冲信号，上述引脚的读写选通脉冲信号都是低电平有效。CH365 提供的地址线、数据总线、读写选通信号线类似于 ISA 总线的信号线，所以非常适合将 ISA 板卡升级到 PCI 总线上。并且从图中可以看出，CH365 提供的读写选通信号已经在芯片内部被片选控制，CH365 输出的读写选通信号只是在其基址映射范围内有效，所以外部设备不再需要片选译码。

在 I/O 读写操作期间，CH365 的 A7～A0 输出 I/O 端口的偏移地址，提供给外部设备的有效偏移地址范围是 00H～EFH，外部设备可以进一步对 A7～A0 进行译码产生二级片选信号。在 I/O 读写操作期间，CH365 的 A15～A10 保持不变，但可以由内部寄存器事先设定为高电平或者低电平，A9～A8 输出 PCI 总线的地址，如果使用本地硬件定址的功能，则应该对 CH365 的 A9～A0 进行地址译码，通过 IOP_HIT 引脚向 CH365 请求本地定址，以实现与 ISA 总线相兼容的 000H～3FFH 地址范围内的 I/O 端口定址。

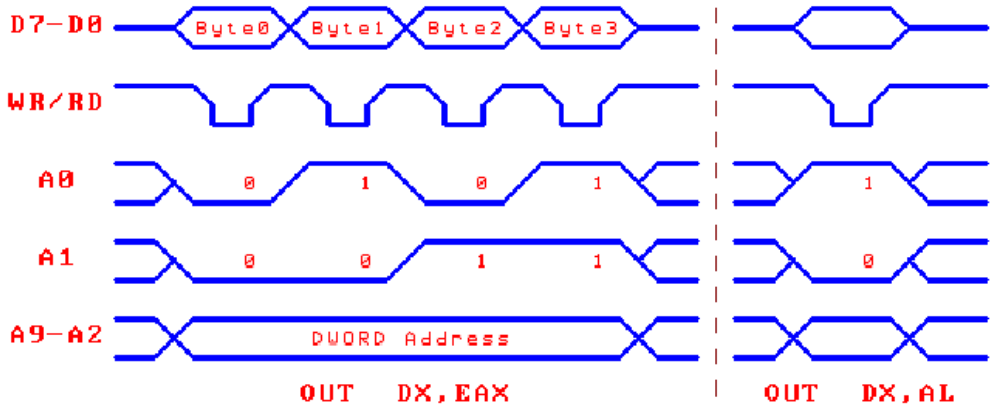
在存储器读写操作期间，CH365 的 A14～A0 输出存储器的偏移地址，提供给外部设备的有效偏移地址范围是 0000H～7FFFH。在存储器读写操作期间，CH365 的 A15 保持不变，但可以由内部寄存器事先设定为高电平或者低电平，用于存储器地址线扩展或者页面选择。由于扩展 ROM 属于存储器的一种，所以其操作方式以及操作时序与存储器相同。



数据宽度：

CH365 支持 PC 机程序以单字节、双字节（字）、四字节（双字）为单位对 I/O 端口或者存储器进行读写。在多字节连续读写操作期间，CH365 每读写完一个字节数据后，就会自动将偏移地址加 1，以指向下一字节的偏移地址。由于 PC 机只要执行一条读写四字节数据的指令，CH365 就会自动将其分解为 4 个 8 位数据的连续读写操作，所以在 PC 机看来，CH365 能够提供 8 位、16 位和 32 位的数据宽度，而且实际情况也是 32 位数据宽度时的工作效率更高，总体数据交换速度更快。

下图左边是汇编指令“OUT DX，EAX”的波形，其中 DX 是 CH365 的 I/O 基址范围内的任意双字边界的地址，例如 DX=IoBaseAddr+4；右边是“OUT DX，AL”的波形，其中 DX 是 CH365 的 I/O 基址范围内的任意地址， 例如 DX=IoBaseAddr+1， 对应的 C 语言指令是 “outportb （IoBaseAddr+1， Value） ” 。I/O 读操作和存储器的读写操作与 I/O 写操作的情况相类似。当 PC 机程序执行 32 位数据宽度的指令时，由于 CH365 自动产生连续的 4 次 8 位数据操作，所以数据传输效率比 PC 机程序执行 8 位数据宽度的指令时更高。



示例说明：

基于 CH365 设计一块类似于打印口的 PCI 板卡。设计约定板卡的 I/O 偏移地址 00H 是数据端口，偏移地址 01H 是状态端口，偏移地址 02H 是控制端口。当插入 PC 机后，该板卡可能被分配一个 I/O 基址 9500H，则数据端口的实际 I/O 地址是 9500H，状态端口的 I/O 地址是 9501H，控制端口的 I/O 地址是 9502H。区分各个端口是对 CH365 的 A7～A0 进行地址译码实现的，如果不需要其它端口，也可以只对 A1～A0 进行简化译码。

如果将两块完全相同的上述板卡插入 PC 机，则第二块板卡也会被系统自动分配一个 I/O 基址，但一定不会与第一块板卡的 I/O 基址相同。如果第二块板卡的 I/O 基址是 C700H，则第二块板卡的控制端口的实际 I/O 地址是 C702H，从而使得两块完全相同的 PCI 板卡分别具有不同的 I/O 端口地址，避免了 I/O 地址冲突。

板卡设计者和相关的应用程序事先知道各个端口的偏移地址，但是无法事先知道板卡的 I/O 基址，所以应用程序在对 PCI 板卡进行 I/O 操作前，需要通过板卡的配置空间的 I/O 基址寄存器了解当前板卡的 I/O 基址，再由 I/O 基址加上各个端口的偏移地址计算出各个端口的实际 I/O 地址，最后根据实际 I/O 地址对各个端口进行 I/O 操作。

存储器方面与 I/O 端口类似，以 CH365 连接一个 32KB 容量的双口 SRAM 进行高速数据交换为例。如果 CH365 的存储器基址被分配为 E3050000H，则计算机程序读写物理地址范围 E3050000H～E3057FFFH 就是读写该双口 SRAM。注意，实际的 PC 机程序通常使用转换后的虚拟地址而不是物理地址；另外，如果要在 DOS 下对存储器进行读写，通常要将存储器基址设置在 DOS 可寻址的 1MB 以下，例如 000D0000H 或者 000D8000H。

以下是相应的读写过程示例。

① 向控制端口写出数据 5AH，对应 C 语言程序“outportb(0x9502,0x5A)” 。执行后 CH365 的地址线 A7～A0 输出控制端口的偏移地址 02H （地址 9502 被分解为基址 9500H 和偏移地址 02H，CH365 只输出偏移地址，不输出基址），CH365 的数据线 D7～D0 输出 5AH，同时 IOP_WR 输出一个低电平脉冲，脉冲宽度由 CH365 的读写速度控制寄存器事先设定，默认是 240nS。

② 从数据端口和状态端口读入数据，对应 C 语言程序“inport(0x9500)” ，返回结果的低字节是从数据端口读取的数据，高字节是从状态端口读取的数据。执行后 CH365 的地址线 A7～A0 首先输出数据端口的偏移地址 00H，同时 IOP_RD 输出第一个低电平脉冲，外部设备应该将数据输出到数据总线 D7～D0 上；然后 CH365 的地址线 A7～A0 输出状态端口的偏移地址 01H， 同时 IOP_RD 输出第二个低电平脉冲， 外部设备应该将状态输出到数据总线 D7～D0 上。

③ 存储器读写与 I/O 读写类似，但有两点区别：第一是 CH365 的地址线 A14～A0 输出 15 位偏移地址，而 I/O 只有 A7～A0 输出 8 位偏移地址；第二是用 MEM_RD 引脚

输出读控制信号代替 IOP_RD 引脚输出读控制信号，用 MEM_WR 引脚输出写控制信号代替 IOP_WR 引脚输出写控制信号，从而能够让外部设备区分出是存储器读写操作，而不是 I/O 端口的读写操作。

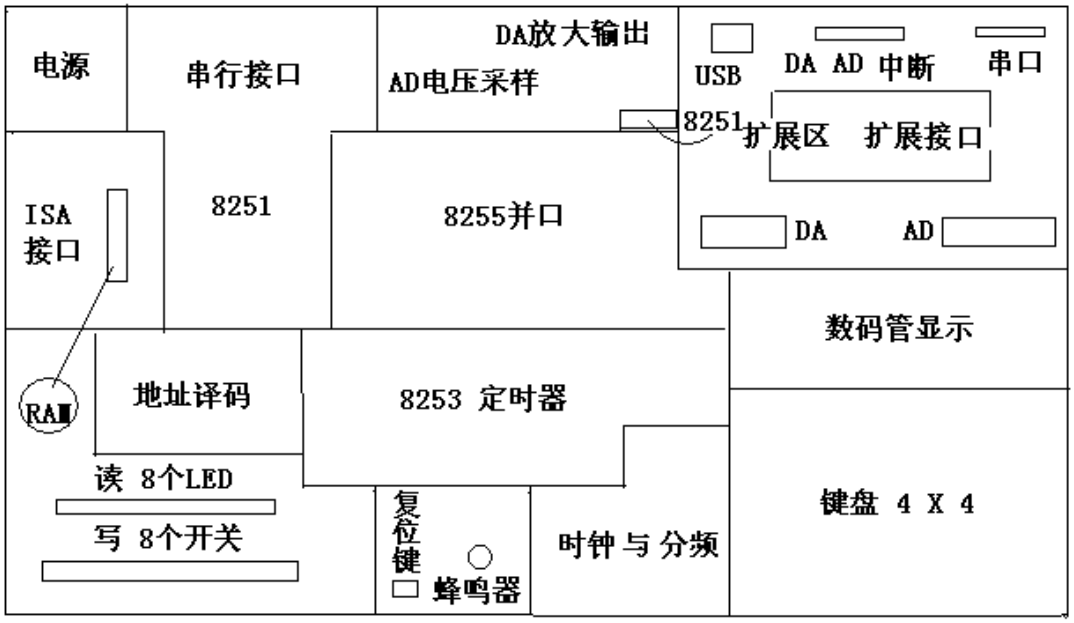
三、实验原理与要求

实验原理：PCI 设备具有自动配置特性，当一个 PCI 设备连接到系统时，系统会根据 PCI 板卡上的配置信息和当前 PC 系统状态进行硬件资源分配，如 I/O 端口基址和中断号等。I/O 端口基址是一个 16 位的 I/O 地址，同一个 PCI 设备在不同计算机中得到的资源信息会有所不同，也即 I/O 基址会有所变化。而要访问实验台上的实验单元，需要将 I/O 端口基址与 74LS138 译码器上的地址结合才能进行。要完成对实验单元的访问，需要完整的 16 位 I/O 端口地址，该地址由 PCI I/O 基址（由系统确定）＋本地译码电路产生的地址构成。

在 DOS 环境下，I/O 基址通过 PCI BIOS 获得，当访问 I/O 单元时，完整地址应是 I/O 基址加上偏移地址。

在 Windows 环境下，访问 I/O 单元地址只需使用偏移地址，而 I/O 基址由系统自动确定，并自动相加形成完整的地址。

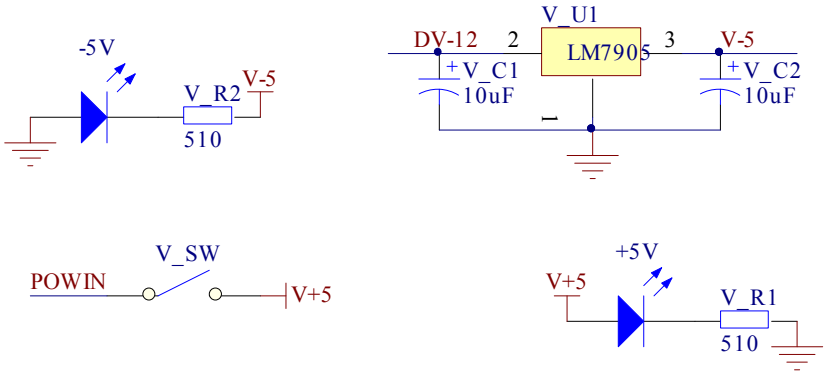
实验平台框图：



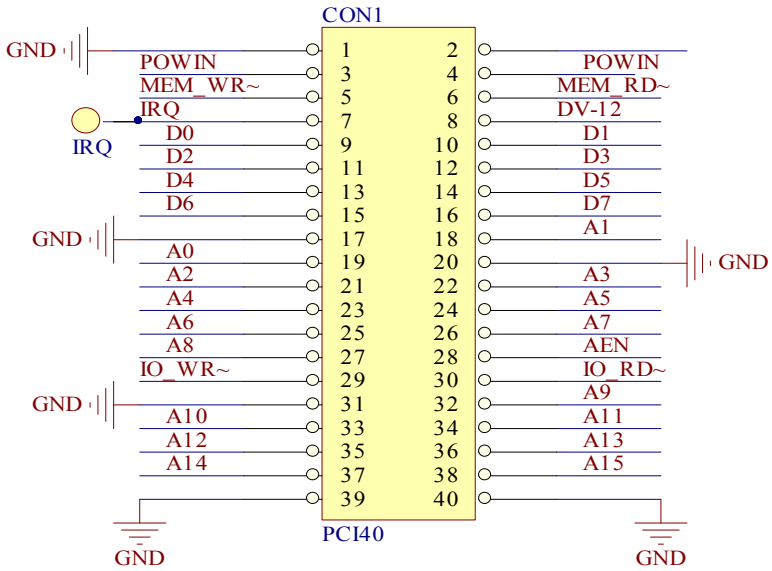
各个模块介绍：

1. **电源模块**

该模块主要对电源进行处理，加上控制开关与标志灯，且通过 LM7905 输出-5V.

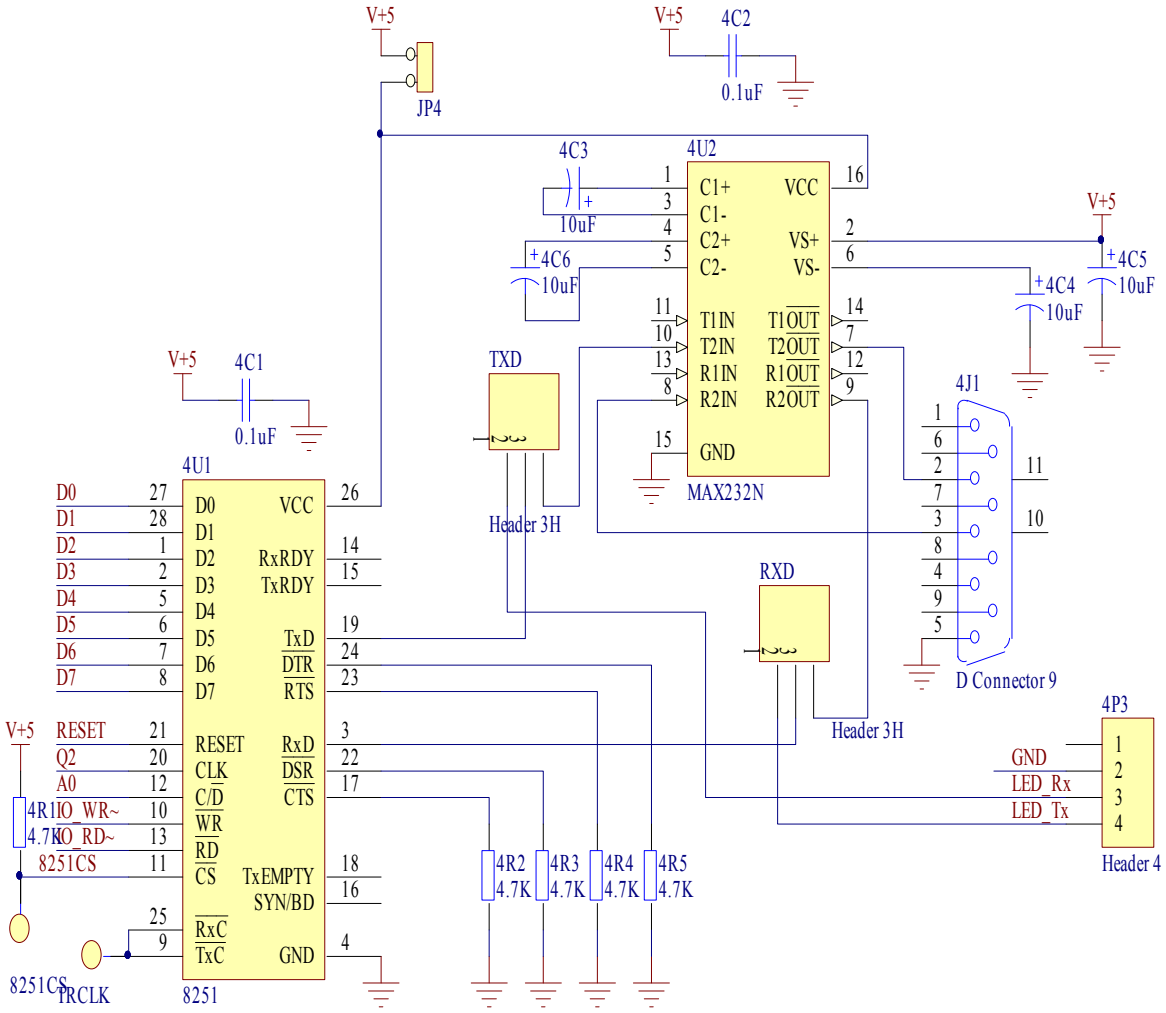


2. ISA 接口

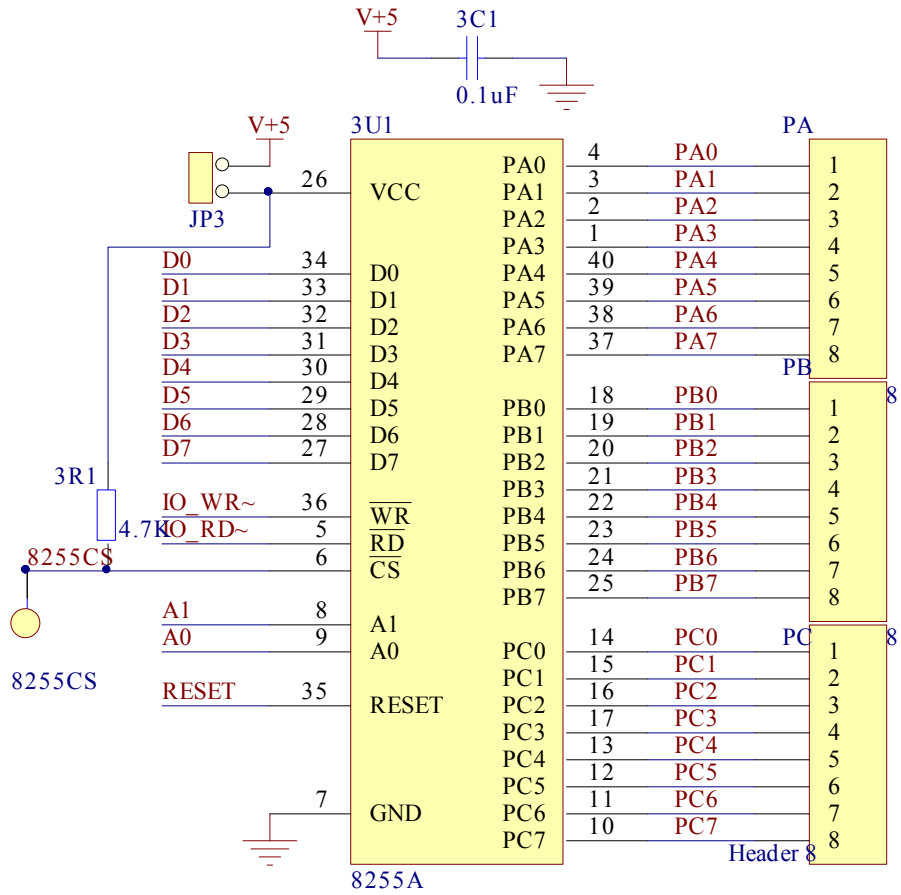


3. 8251 串行接口

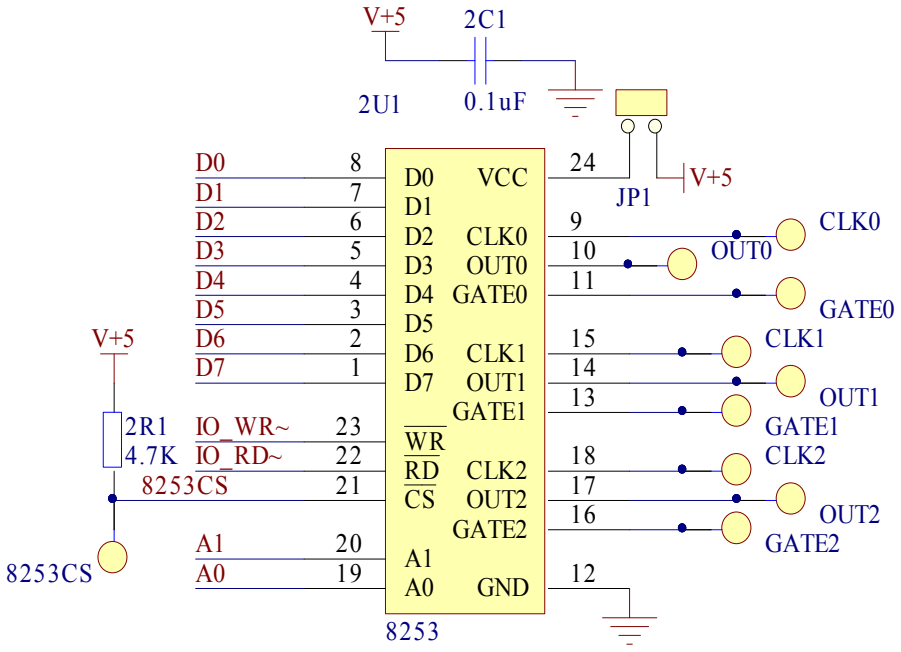
实验台有两个串行接口电路，两个串行口都使用了三线制，即仅使用 TXD、RXD、GND 信号线。一个串口，转 232 串口总线，使用 9 脚 D 型连接器与 PC 机的 9 脚 D 型插座定义相同，这样，使用一根电缆，不仅可以做 PC 和实验台之间的串行通信实验，同时留有原有串行接口可以接相关外设设备，如 LCD 屏等。


V+5
JP4
V+5
4C2
0.1uF
4C3
4U2
10uF
C1+
VCC
C1-
C2+
VS+
C2-
VS-
4C6
10uF
4C4
10uF
V+5
4C5
10uF
T1IN
T1OUT
T2IN
T2OUT
R1IN
R1OUT
R2IN
R2OUT
GND
MAX232N
V+5
4C1
0.1uF
TXD
4J1
4U1
Header 3H
RXD
D Connector 9
Header 3H
4P3
GND
LED_Rx
LED_Tx
Header 4
D0
D1
D2
D3
D4
D5
D6
D7
RESET
Q2
A0
IO_WR~
IO_RD~
8251CS
RxC
TxC
VCC
RxRDY
TxRDY
TxD
DTR
RTS
RxD
DSR
CTS
TxEMPTY
SYN/BD
GND
8251
4R1
4.7K
4R2
4.7K
4R3
4.7K
4R4
4.7K
4R5
4.7K
8251CS
TRCLK

4. 8255 并行接口

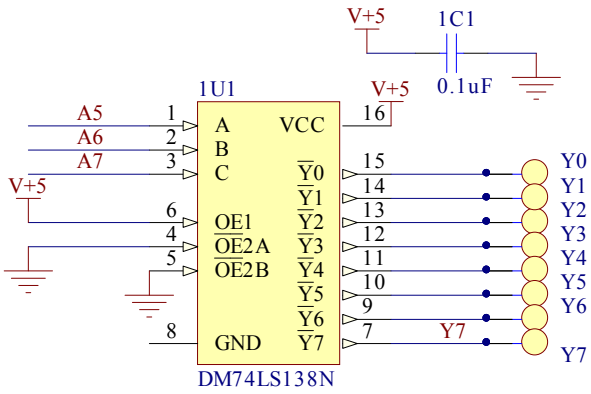


5. 8253 定时器

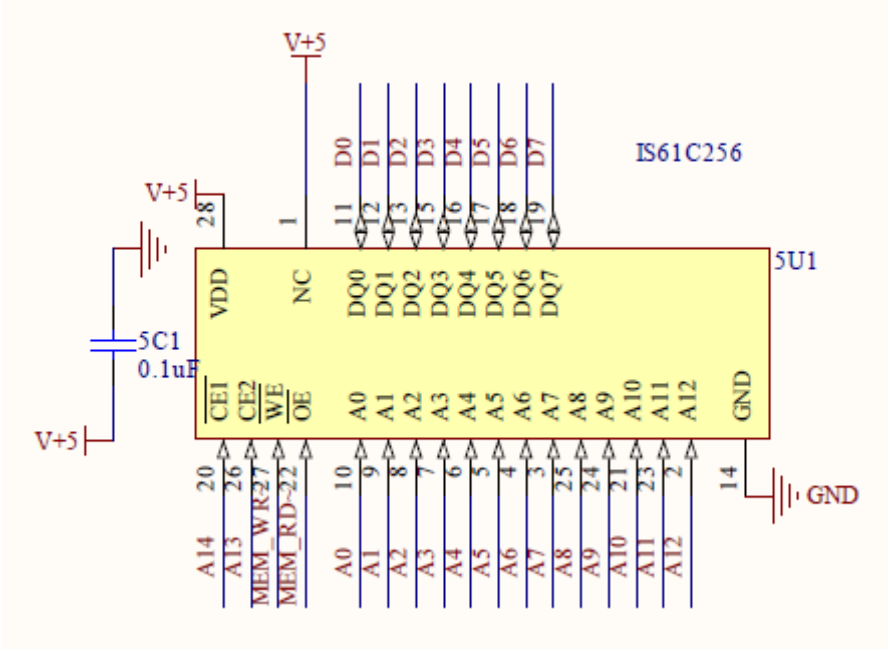


6. 地址译码电路

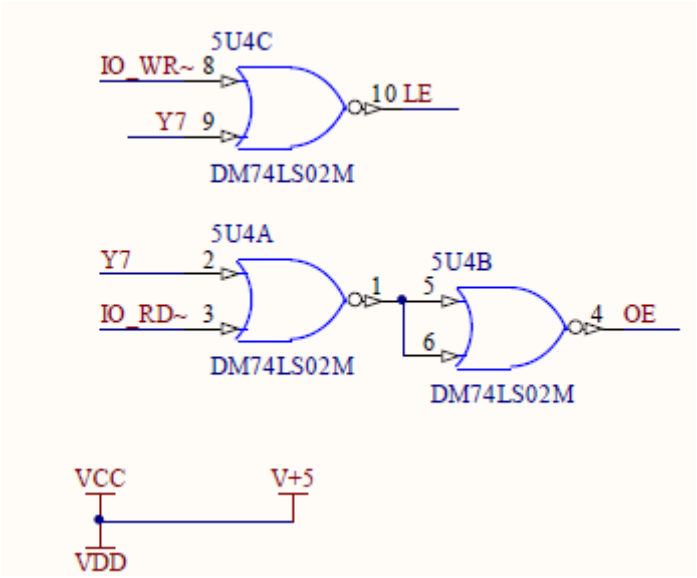
译码电路用来确定有效的 I/O 端口地址范围。

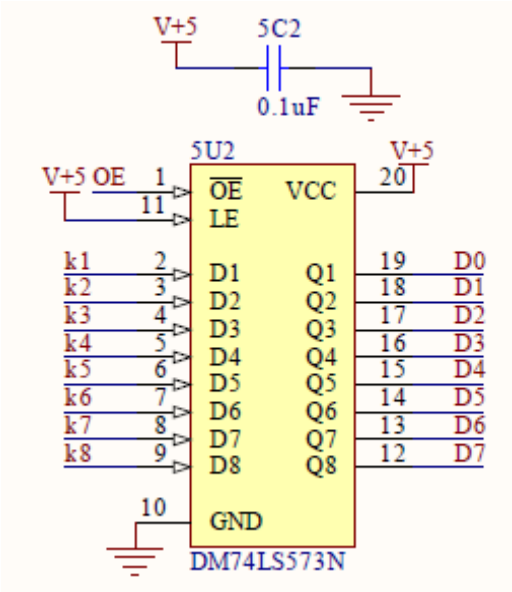


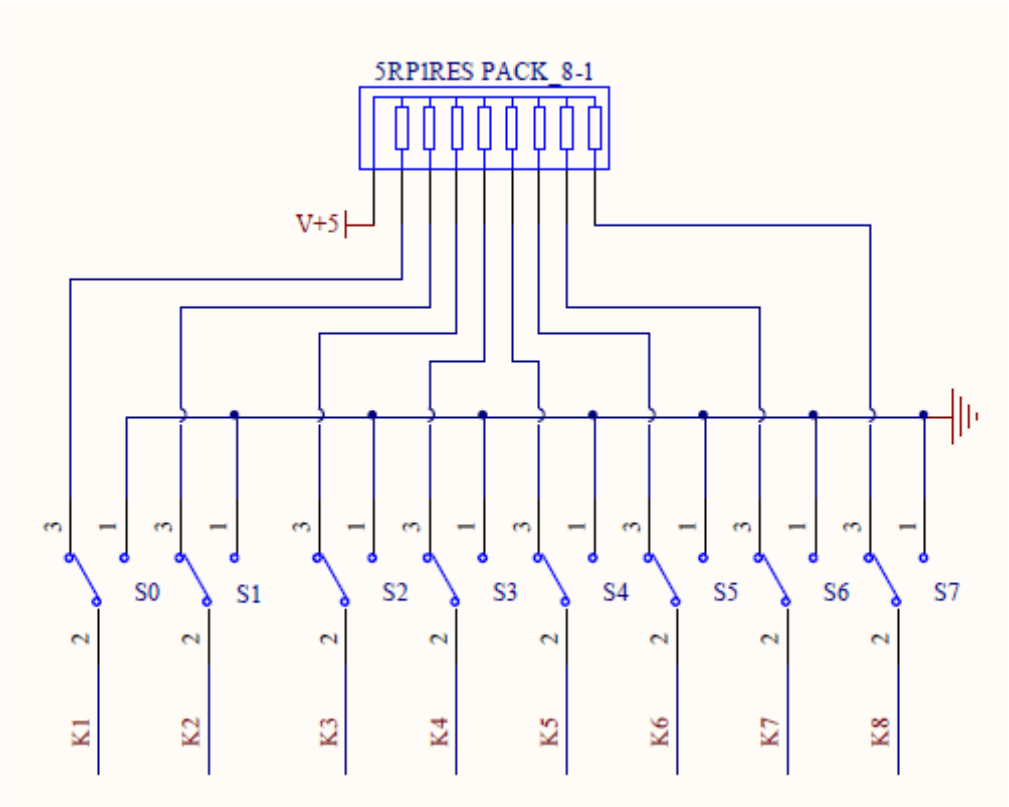
7. RAM 读写电路

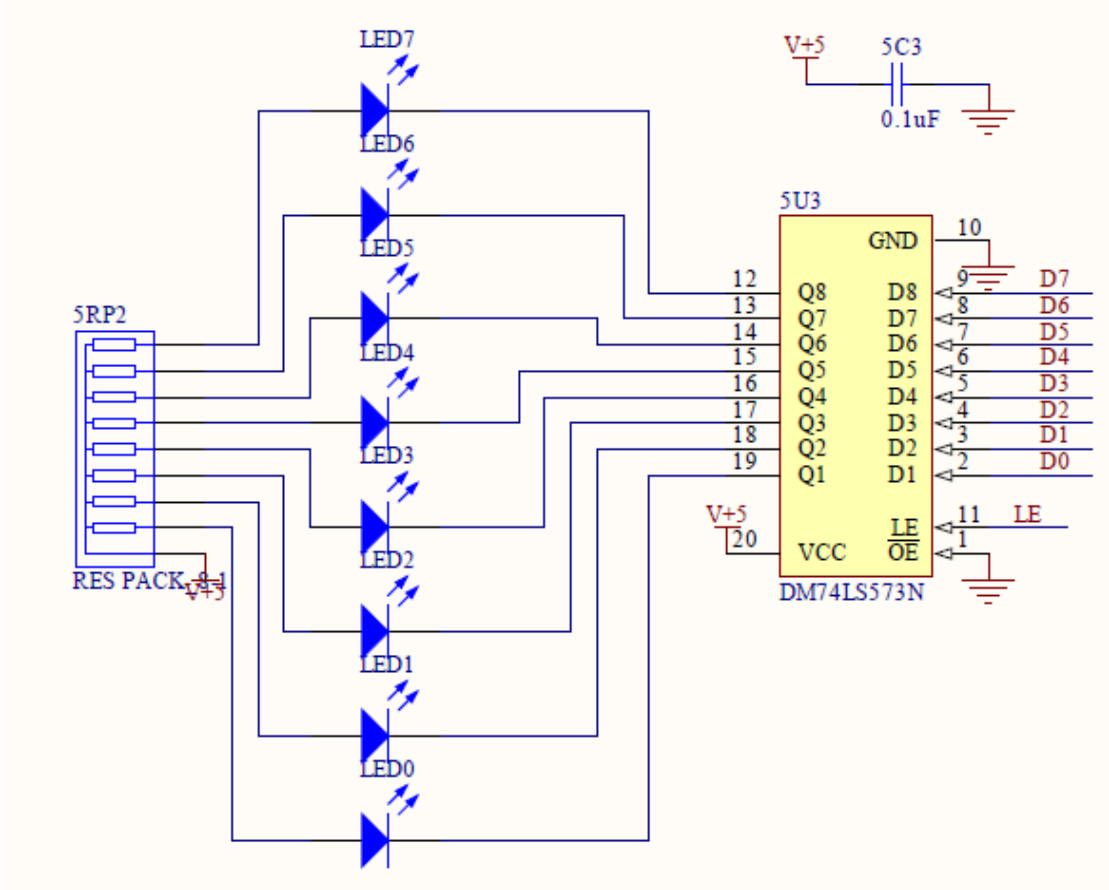


8. 8 个开关与 8 个 LED 电路。




V+5
5C2
0.1uF
5U2
V+5
V+5 OE
1
11
OE
LE
VCC
20
k1 2
k2 3
k3 4
k4 5
k5 6
k6 7
k7 8
k8 9
D1
D2
D3
D4
D5
D6
D7
D8
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
19 D0
18 D1
17 D2
16 D3
15 D4
14 D5
13 D6
12 D7
10
GND
DM74LS573N

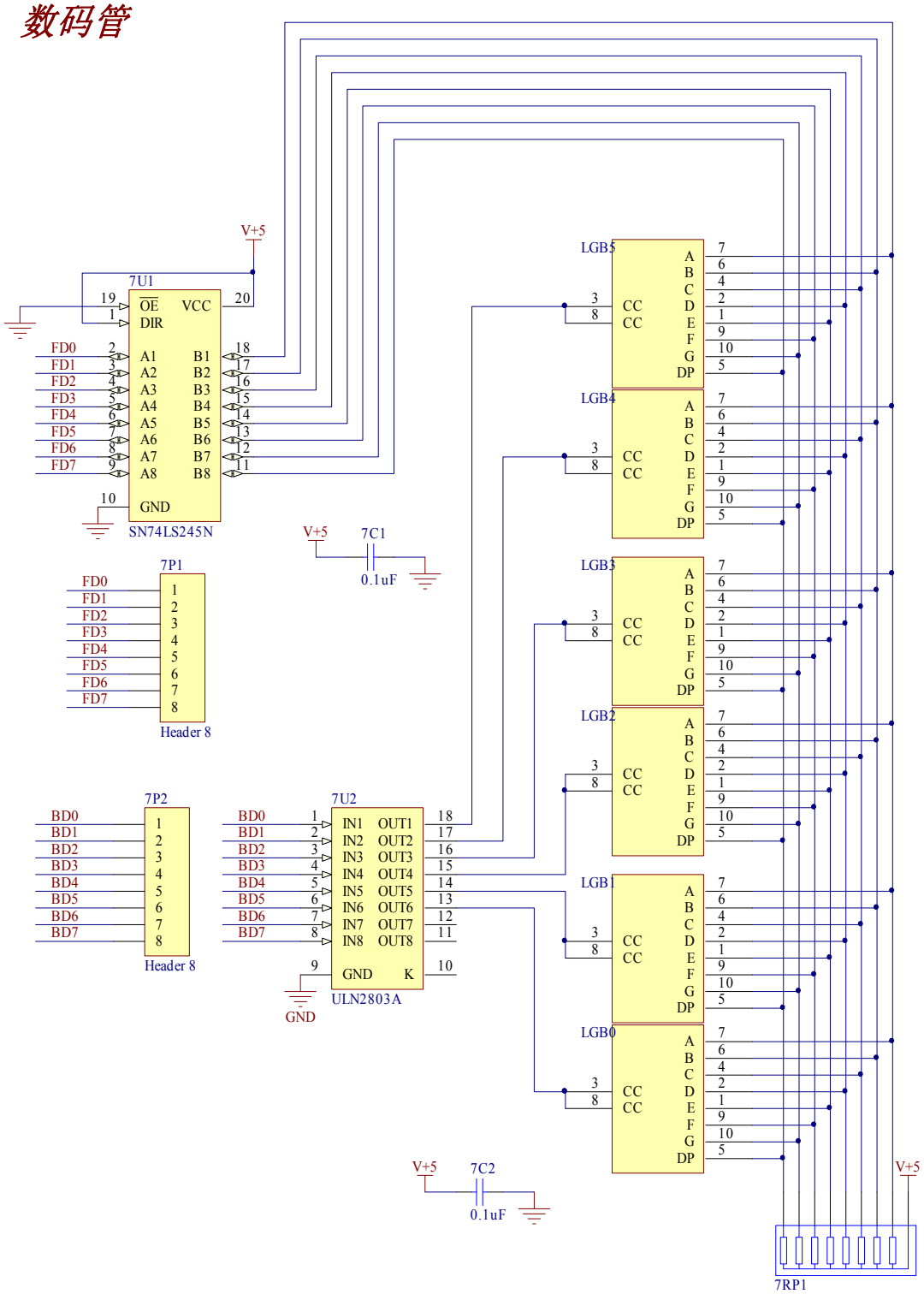

5RP1RES PACK_8-1
V+5
S0
S1
S2
S3
S4
S5
S6
S7
K1
K2
K3
K4
K5
K6
K7
K8



9. 数码管显示

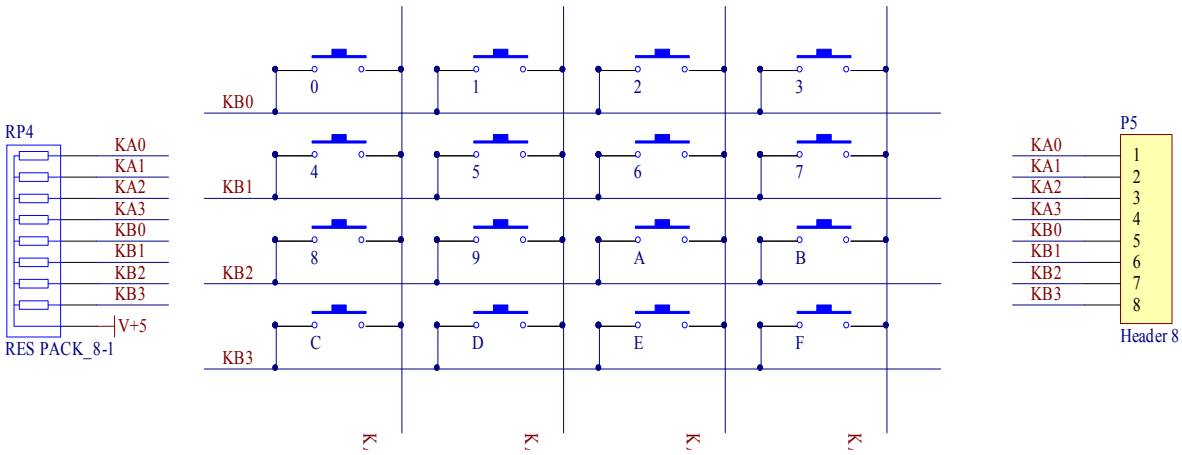
数码管采用共阴极结构，74LS273 触发器用于段选，TPIC6B273 用于位选，TPIC6B273 内部包括 D 触发器和驱动器，两个芯片都是上升沿触发。当段选码“1”置入 74LS273 之后，输出为高电平，将位选码“1”置入 TPIC6B273 后，输出为低电平，结果点亮某个数码管的某些段。要使六位数码管显示不同的数字，需要使用动态扫描方法

数码管



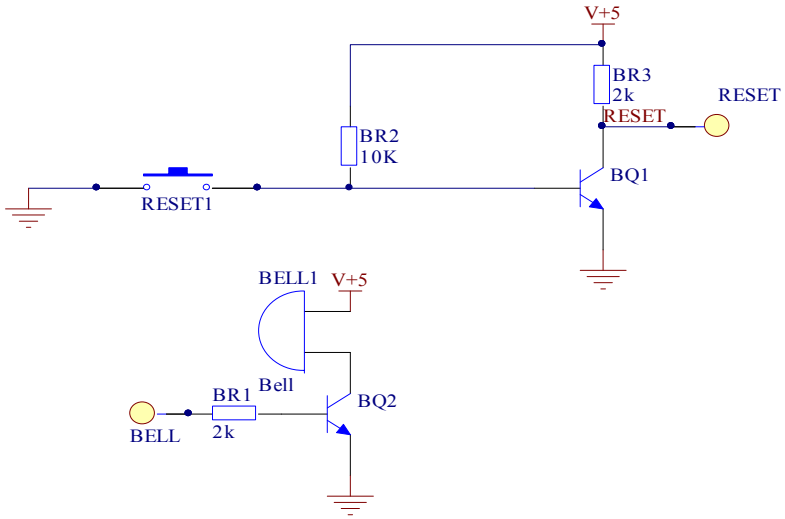
10. 键盘电路

实验系统提供一个 16 按键的小键盘，采用 4×4 键盘矩阵结构，如图 1—6 所示。当某键按下时，行和列上相应两条线短路，检测两条线的状态即可判断某键是否按下。通常，该键盘矩阵的行线和列线分别与 8255 并行接口的两个端口连接，通过程序的检测和判断来识别按键操作，具体见键盘实验内容。



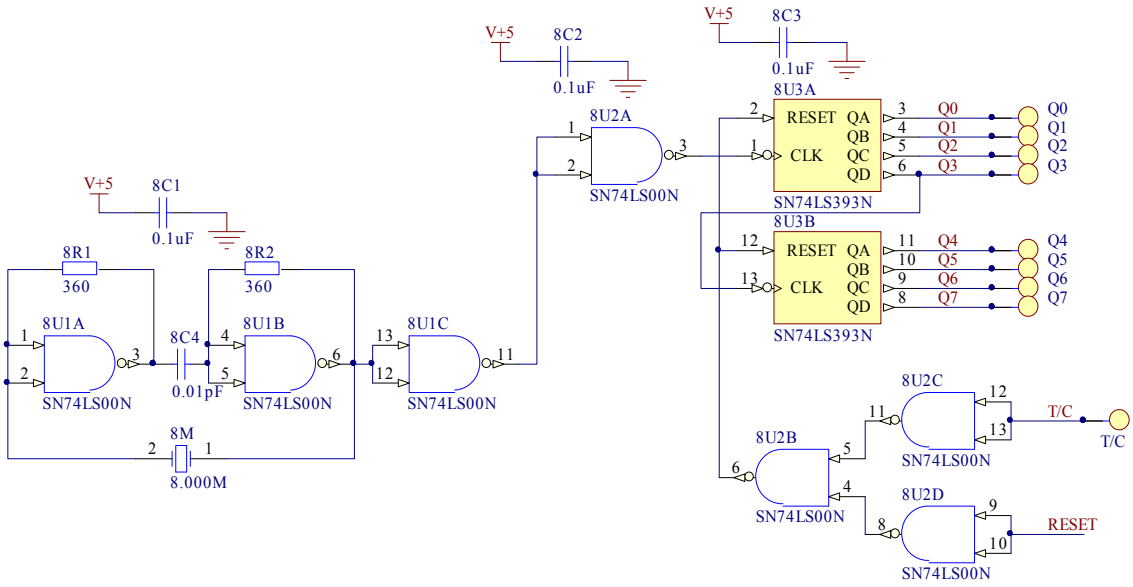
11. 复位与蜂鸣器

复位键用完成相关芯片的复位重启。蜂鸣器在介入点输入高电平时会发蜂鸣声，可以完成报警等功能。



12. 时钟与分频

实验台提供了一个时钟电路，由 8MHz 晶振电路经分频以后产生 8 个不同频率的脉冲信号，分别是 4MHz、2MHz、1MHz、500KHz、250KHz、125KHz、62.5KHz、31.25KHz。在实验台上标有不同脉冲频率的信号插孔即为这些脉冲信号的输出端。



实验一　熟悉实验环境及 IO 的使用

一、 实验目的

1. 通过实验了解和熟悉实验台的结构，功能及使用方法。
2. 通过实验掌握直接使用 Debug 的 I、O 命令来读写 IO 端口。
3. 学会 Debug 的使用及编写汇编程序

二、实验内容及要求

1. 学习使用 Debug 命令，并用 I、O 命令直接对端口进行读写操作，
2. 用汇编语言编写跑马灯程序。（使用 EDIT 编辑工具）实现功能
   A. 通过读入端口状态（ON 为低电平），选择工作模式（灯的闪烁方式、速度等）。
   B. 通过输出端口控制灯的工作状态（低电平灯亮）

注意：电源打开时不得插拔电缆及各种器件
连接电路时一定要在断电的情况下连接，否则可能会烧坏整个实验系统

三、操作步骤

1.实验板的 IO 端口地址为　EEE0H

在 Debug 下，

I 是读命令。（即读输入端口的状态---拨码开关的状态）

O 是写命令。（即向端口输出数据---通过发光管来查看）

进入 Debug 后，

读端口　拨动实验台上八位拨码开关

输入　I　端口地址 回车

屏幕显示　xx　表示从端口读出的内容，即八位开关的状态 ON 是 0，OFF 是 1

写端口

输入　O　端口地址　xx （xx 表示要向端口输出的内容）回车

查看实验台上的发光二极管状态，0 是灯亮，1 是灯灭。

2. 在 Debug 环境下，用 a 命令录入程序，用 g 命令运行

C>Debug

-a

mov　　dx, 端口地址

mov　　al,输出内容

out　　dx, al

mov　　ah, 0bh

int　　21h

or　　al, al

jz　　0100

int　　20h

-g

运行查看结果

修改输出内容

再运行查看结果

分析

mov　　ah, 0bh

int　　21h

or　　al, al

jz　　0100

int　　20h

该段程序的作用

3.利用 EDIT 工具编写汇编写跑马灯程序程序

实现功能

A．通过读入端口状态（ON 为低电平），选择工作模式（灯的闪烁方式、速度等）。

B．通过输出端口控制灯的工作状态（低电平灯亮）

C>EDIT 文件名.asm

录入程序

按 Alt 键　打开菜单 进行存盘或退出

编译文件

C>MASM　文件名.asm

```
E:\masm5>masm aa.asm
Microsoft (R) Macro Assembler Version 5.00
Copyright (C) Microsoft Corp 1981-1985, 1987.  All rights reserved.

Object filename [aa.OBJ]:
Source listing  [NUL.LST]:
Cross-reference [NUL.CRF]:

  50094 + 415298 Bytes symbol space free

       0 Warning Errors
       0 Severe  Errors
```

连接文件

C>LINK　文件名.obj

```
E:\masm5>link aa.obj

Microsoft (R) Overlay Linker  Version 3.60
Copyright (C) Microsoft Corp 1983-1987.  All rights reserved.

Run File [AA.EXE]:
List File [NUL.MAP]:
Libraries [.LIB]:
```

运行文件
或用 Debug 进行调试。

四、实验报告

1. 画出程序流程图，打印程序并加注释。
2. 通过实验说明用 debug 中的 a 命令录入实验中给出的小程序中，有些语句可以不写出“h”字符的原因。
3. 实验收获和体会。

附录　部分 DOS 命令

1. 查看文件
C>dir/p

Dir 命令的帮助
C>dir/?

2. 建立子目录
C>md 子目录名

3. 进入子目录
C>cd 子目录名

4. 退到上一级子目录
C>cd..

5. 退到根目录
C>cd\

6. 复制文件
C>copy　源文件盘符：源文件名　目标文件盘符：

7. DOS 命令帮助
C>help