

计算机硬件技术 基础实验指导

● 吴珍娣 主编



同济大学出版社
TONGJI UNIVERSITY PRESS

计算机硬件技术基础 实验指导

主 编 吴珍娣



同济大学出版社
TONGJI UNIVERSITY PRESS

内容提要

本书是配合《计算机硬件技术基础》教材的一本实验指导书。

本书的实验具有较强的系统性和操作性,要求实验环境相对简单和统一,学生通过完成设计的实验内容,能够深入掌握和理解计算机硬件技术的工作原理,增强处理实际问题的能力。本书还精选了一些学生作品,供读者参考。

本书可供非电类各专业学生阅读参考。

图书在版编目(CIP)数据

计算机硬件技术基础实验指导/吴珍娣主编. —上海:同济大学出版社,2009.1

ISBN 978-7-5608-3890-8

I. 计… II. 吴… III. 硬件—高等学校—教学参考资料 IV. TP303

中国版本图书馆 CIP 数据核字(2008)第 187017 号

计算机硬件技术基础实验指导

主编 吴珍娣

责任编辑 辜翔 责任校对 徐春莲 封面设计 陈益平

出版发行 同济大学出版社 www.tongjipress.com.cn

(地址:上海市四平路 1239 号 邮编:200092 电话:021-65985622)

经销 全国各地新华书店

印刷 同济大学印刷厂

开本 787mm×1092mm 1/16

印张 6.75

印数 1—3 600

字数 168 000

版次 2009 年 1 月第 1 版 2009 年 1 月第 1 次印刷

书号 ISBN 978-7-5608-3890-8

定 价 12.00 元

本书若有印装质量问题,请向本社发行部调换 版权所有 侵权必究

编者的话

21 世纪是知识经济的时代,是人才竞争的时代,随着 21 世纪的到来,人类已步入信息社会,信息产业正成为全球经济的主导产业,计算机科学与技术与信息产业中占据了最重要的地位,这就对培养 21 世纪高素质创新型计算机专业人才提出了迫切的要求。

随着计算机技术的日益普及和应用,计算机已成为各行各业最基本的应用工具之一,掌握计算机的硬件技术已成为大学生必备的技能。计算机硬件技术基础课程是高校非电类的公共基础课,它不仅理论性很强,而且实践性也很强,学生必须通过严格的实践训练才能真正掌握和深入理解计算机硬件的基本理论。目前,关于计算机硬件技术基础的书多以理论知识的介绍和讲解为主,很少涉及计算机硬件技术实验。本书是配合《计算机硬件技术基础》教材而编写的一本实验指导书,是为普通高校本科教育计算机硬件实验服务的。作者在总结多年计算机硬件技术实验教学经验的基础上,并结合同济大学所使用的教学仪器设备,根据学生的学习情况,结合实验难易程度,编制了理论知识与现实生活相结合的小实验,提供给学生学习。

本书的实验具有较强的系统性和可操作性,所要求的实验环境相对简单和统一,学生通过完成设计的实验内容,能够深入掌握和理解计算机硬件技术的工作原理,增强处理实际问题的能力。书中还选了一些学生作品,供同学们参考。

本书中的实验二是同济大学实验室与设备管理处评为精品实验项目,在此特别要感谢同济大学实验室与设备管理处的大力支持!

吴珍妮

2008 年 11 月

目 录

编者的话

实验一	实验箱系统认识实验	1
实验二	静态存储器扩展实验	5
实验三	电子发声设计实验	9
实验四	点阵 LED 显示设计实验	14
实验五	步进电机实验	26
实验六	直流电机闭环调速实验	30
实验七	8254 定时/计数器应用实验	51
实验八	A/D 转换实验	59
优秀实验作品选		65
附录 1	Wmd86 联机软件使用说明	92
附录 2	系统编程信息	100
附录 3	TD-PITC 实验箱布局图	102

实验一 实验箱系统认识实验

实验目的

掌握 TD-PITC 80X86 微机原理及接口技术教学实验系统的操作,熟悉 Wmd86 联机集成开发调试软件的操作环境。

实验设备

PC 机一台,TD-PITC 实验装置一套。

实验内容

编写实验程序,将 00H~0FH 共 16 个数写入内存 3000H 开始的连续 16 个存储单元中。

实验步骤

1. 运行 Wmd86 软件,进入 Wmd86 集成开发环境。
2. 根据程序设计使用语言的不同,通过在“设置”下拉列表来选择需要使用的语言,如图 1-1 所示。语言选择后,下次再启动软件,语言环境保持这次的修改不变。在这里,我们选择汇编语言。

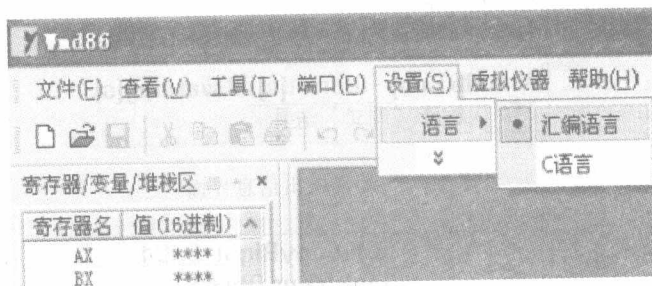


图 1-1 语言环境选择界面

3. 语言选择后,点击新建或按 Ctrl+N 组合键来新建一个文档,如图 1-2 所示。默认文件名为 Wmd861。

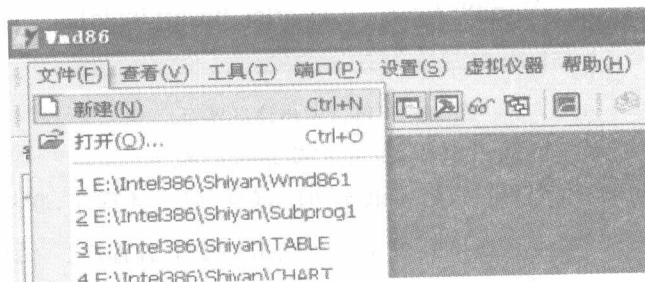


图 1-2 新建文件界面

4. 编写实验程序,如图 1-3 所示,并保存,此时系统会提示输入新的文件名,输完后点击保存。

```

SSTACK  SEGMENT STACK                      :定义堆栈段
        DW 32 DUP(?)
SSTACK  ENDS

CODE    SEGMENT
        ASSUME CS:CODE, SS:SSTACK
START:  PUSH DS
        XOR AX, AX
        MOV DS, AX
        MOV SI, 3000H                      :建立数据起始地址
        MOV CX, 16                          :循环次数
AA1:    MOV [SI], AL
        INC SI                               :地址自加1
        INC AL                               :数据自加1
        LOOP AA1
AA2:    JMP AA2
CODE    ENDS
        END START

```

图 1-3 程序编辑界面

5. 点击  ,编译文件,若程序编译无误,则输出如图 1-4 所示的输出信息,然后再点击  进行链接,链接无误后,输出如图 1-5 所示的输出信息。

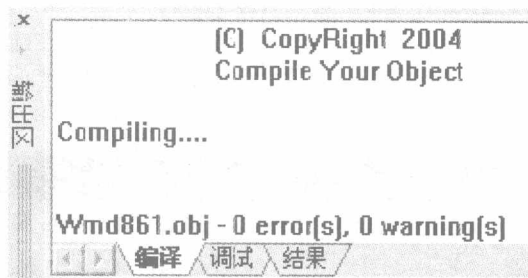


图 1-4 编译输出信息界面

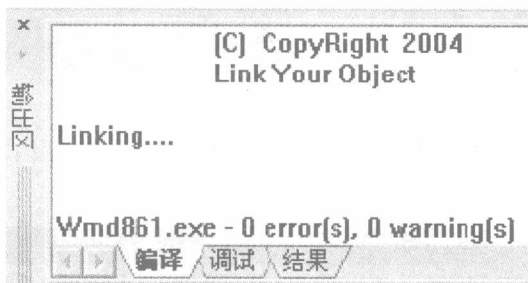


图 1-5 链接输出信息界面

6. 连接 PC 与实验系统的通讯电缆,打开实验系统电源。

7. 编译、链接都正确并且上下位机通讯成功后,就可以下载程序,联机调试了。可以通过端口列表中的“端口测试”来检查通讯是否正常。点击  下载程序。  为编译、链接、下载组合按钮,通过该按钮可以将编译、链接、下载一次完成。下载成功后,在输出区的结果窗中会显示“加载成功!”字样,表示程序已正确下载。起始运行语句下会有一条绿色的背景。如图

1-6所示。

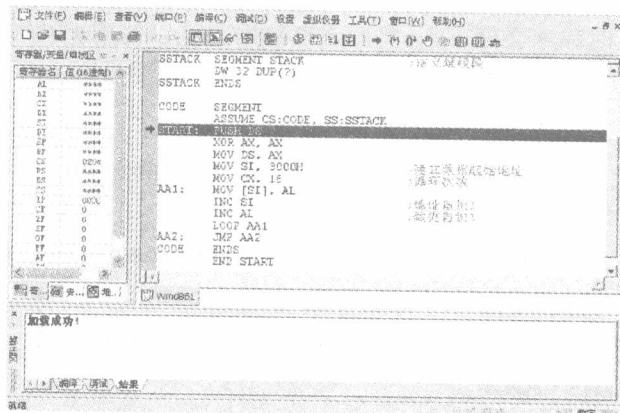


图 1-6 加载成功输出显示界面

8. 将输出区切换到调试窗口,使用 D0000:3000 命令查看内存 3000H 起始地址的数据,如图 1-7 所示。存储器在初始状态时,默认数据为 CC。

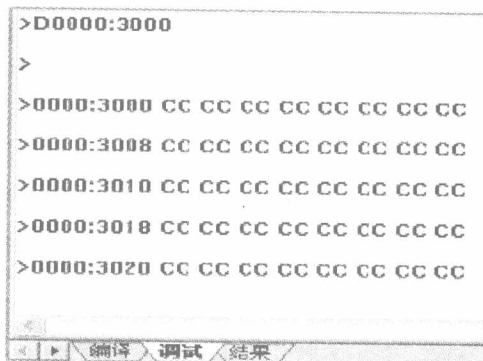


图 1-7 内存地址单元数据显示

9. 点击按钮  运行程序,然后再点击  来停止程序运行,接下来观察程序运行结果,仍使用命令 D0000:3000 来观察数据变化。如图 1-8 所示。

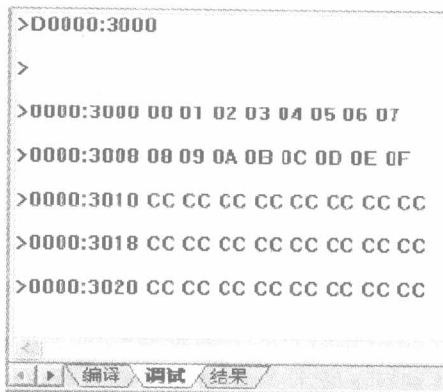


图 1-8 运行程序后数据变化显示

10. 也可以通过  在语句 AA2: JMP AA2 处设置断点,断点显示如图 1-9 所示,然后运行程序,当遇到断点时程序会停下来,然后观察数据。可以使用 E0000:3000 来改变该地址单元的数据,如图 1-10 所示,输入 11 后,按“空格”键,可以接着输入第二个数,如 22,结束输入按“回车”键。

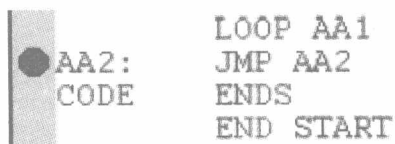


图 1-9 断点设置显示

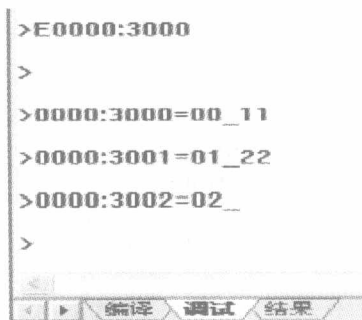


图 1-10 修改内存单元数据显示界面

实验参考例程(文件名 Wmd861.asm)

```

SSTACK    SEGMENT STACK                ;定义堆栈段
            DW 32 DUP(?)
SSTACK    ENDS

CODE SEGMENT
            ASSUME CS:CODE, SS:SSTACK
START:    PUSH DS
            XOR AX, AX
            MOV DS, AX
            MOV SI, 3000H                ;建立数据起始地址
            MOV CX, 16                   ;循环次数
AA1:      MOV [SI], AL
            INC SI                        ;地址自加 1
            INC AL                        ;数据自加 1
            LOOP AA1
AA2:      JMP AA2
CODE ENDS
            END START

```

思考

编写程序,将内存 4000H 单元开始的 8 个数据复制到 4100H 单元开始的数据区中。通过调试验证程序功能,使用 E 命令修改 4000H 单元开始的数据,运行程序后使用 D 命令查看 4100H 单元开始的数据。

实验二 静态存储器扩展实验

实验目的

1. 了解存储器扩展的方法和存储器的读/写;
2. 掌握 CPU 对 16 位存储器的访问方法。

实验设备

PC 机一台, TD-PITC 实验装置一套, 示波器一台。

实验内容

编写实验程序, 将 0000H~000FH 共 16 个数写入 SRAM 的从 0000H 起始的一段空间中, 然后通过系统命令查看该存储空间, 检测写入数据是否正确。

实验原理

存储器是用来存储信息的部件, 是计算机的重要组成部分, 静态 RAM 是由 MOS 管组成的触发器电路, 每个触发器可以存放 1 位信息。只要不掉电, 所储存的信息就不会丢失。因此, 静态 RAM 工作稳定, 不要外加刷新电路, 使用方便。但一般 SRAM 的每一个触发器是由 6 个晶体管组成, SRAM 芯片的集成度不会太高, 目前较常用的有 6116(2K×8 位), 6264(8K×8 位) 和 62256(32K×8 位)。本实验平台上选用的是 62256, 两片组成 32K×16 位的形式, 共 64K 字节。图 2-1 为 62256 的外部引脚图。

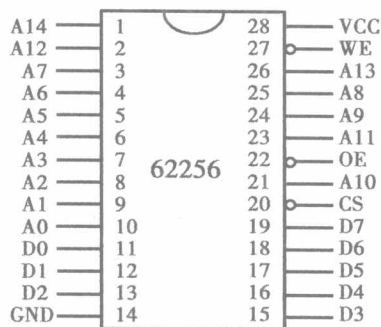


图 2-1 62256 引脚图

本系统采用准 32 位 CPU, 具有 16 位外部数据总线, 即 D0、D1、...、D15, 地址总线为 BHE# (#表示该信号低电平有效)、BLE#、A1、A2、...、A20。存储器分为奇体和偶体, 分别由字节允许线 BHE# 和 BLE# 选通。

存储器中, 从偶地址开始存放的字称为规则字, 从奇地址开始存放的字称为非规则字。处理器访问规则字只需要一个时钟周期, BHE# 和 BLE# 同时有效, 从而同时选通存储器奇体和偶体。处理器访问非规则字需要两个时钟周期, 第一个时钟周期 BHE# 有效, 访问奇字节;

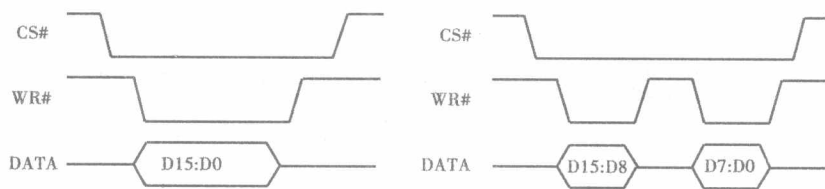


图 2-2 写规则字(左)和非规则字(右)简单时序图

第二个时钟周期 BLE# 有效,访问偶字节。处理器访问字节只需要一个时钟周期,视其存放单元为奇或偶,而 BHE# 或 BLE# 有效,从而选择奇体或偶体。写规则字和非规则字的简单时序图如图 2-2 所示。两片 62256 连接的电路如图 2-3 所示。

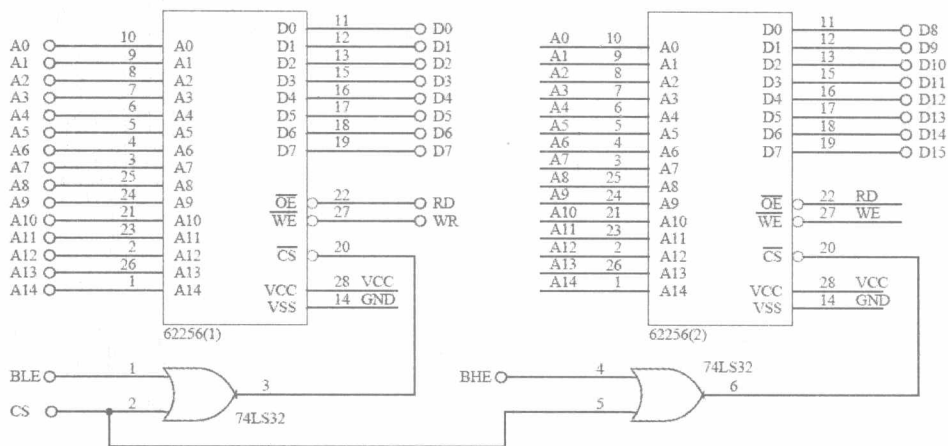


图 2-3 SRAM 单元电路图

实验步骤

1. 实验接线图如图 2-4 所示,按图接线;

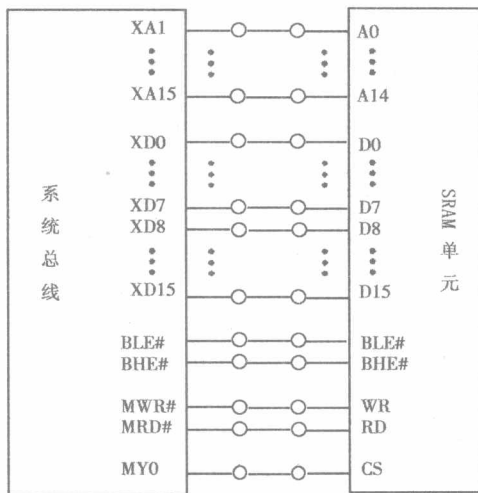


图 2-4 SRAM 实验接线图

2. 编写实验程序,经编译、链接无误后装入系统;
3. 先运行程序,然后再停止程序运行;
4. 通过 D 命令查看写入存储器中的数据,应为 0001、0002、…、000F 共 16 个字;
5. 改变实验程序,按非规则字写存储器,观察实验结果;
6. 改变实验程序,按字节方式写存储器,观察实验现象。

实验参考例程(文件名 MEM1. ASM)

```

SSTACK  SEGMENT STACK
          DW 32 DUP(?)
SSTACK  ENDS

CODE     SEGMENT
START    PROC FAR
          ASSUME CS:CODE

          MOV AX, 8000H                ;存储器扩展空间段地址
          MOV DS, AX
AA0:     MOV SI, 0000H                ;数据首地址
          MOV CX, 0010H
          MOV AX, 0000H
AA1:     MOV [SI], AX
          INC AL
          INC SI
          INC SI
          LOOP AA1
          NOP
HERE:    JMP HERE
START    ENDP
CODE     ENDS
END START

```

实验参考例程(CMEM. C)

```

void main(void)
{
    unsigned int * Addr;
    unsigned int i;
    Addr = (unsigned int *)0x80000000;
    for(i=0; i<16; i++)
    {

```

```
    * Addr = i;  
    Addr += 1;  
}  
while(1);  
}
```

思考

通过本实验的程序调试,你可以进一步了解程序写入 SRAM 中存放的情况。你还可以编写一段文字写入 SRAM 中的某一段存储空间里。然后用系统命令查看该存储空间的内容;并仔细观察它存放的位置。

实验三 电子发声设计实验

实验目的

学习用 8254 定时/计数器使蜂鸣器发声的编程方法。

实验设备

PC 微机一台、TD-PITC 实验箱一台。

实验内容

根据实验提供的音乐频率表和时间表,编写程序控制 8254,使其输出连接到扬声器上能发出相应的乐曲。

实验说明及步骤

一个音符对应一个频率,将对应一个音符频率的方波传输到扬声器上,就可以发出这个音符的声音。将一段乐曲的音符对应频率的方波依次送到扬声器,就可以演奏出这段乐曲。利用 8254 的方式 3——“方波发生器”,将相应一种频率的计数初值写入计数器,就可产生对应频率的方波。计数初值的计算如下:

$$\text{计数初值} = \text{输入时钟} \div \text{输出频率}$$

例如,输入时钟采用 1MHz,要得到 800Hz 的频率,计数初值即为 $1000000 \div 800$ 。音符与频率对照关系如表 3-1 所示。对于每一个音符的演奏时间,可以通过软件延时来处理。首先确定单位延时时间程序(根据 CPU 的频率不同而有所变化)。然后确定每个音符演奏需要几个单位时间,将这个值送入 DL 中,调用 DALLY 子程序即可。

;单位延时时间

DALLY PROC

D0: MOV CX, 0010H

D1: MOV AX, 0F00H

D2: DEC AX

JNZ D2

LOOP D1

RET

DALLY ENDP

; N 个单位延时时间 (N 送至 DL)

DALLY PROC

D0: MOV CX, 0010H

```

D1: MOV AX, 0F00H
D2: DEC AX
JNZ D2
LOOP D1
DEC DL
JNZ D0
RET
DALLY ENDP

```

表 3-1

音符与频率对照表

单位:Hz

音 符 音 调	1	2	3	4	5	6	7
A	221	248	278	294	330	371	416
B	248	278	312	330	371	416	467
C	131	147	165	175	196	221	248
D	147	165	185	196	221	248	278
E	165	185	208	221	248	278	312
F	175	196	221	234	262	294	330
G	196	221	248	262	294	330	371

音 符 音 调	1	2	3	4	5	6	7
A	441	495	556	589	661	742	833
B	495	556	624	661	742	833	935
C	262	294	330	350	393	441	495
D	294	330	371	393	441	495	556
E	330	371	416	441	495	556	624
F	350	393	441	467	525	589	661
G	393	441	495	525	589	661	742

音 符 音 调	1	2	3	4	5	6	7
A	882	990	1112	1178	1322	1484	1665
B	990	1112	1248	1322	1484	1665	1869
C	525	589	661	700	786	882	990
D	589	661	742	786	882	990	1112
E	661	742	833	882	990	1112	1248
F	700	786	882	935	1049	1178	1322
G	786	882	990	1049	1178	1322	1484

下面提供了乐曲《友谊地久天长》实验参考程序。程序中频率表是将曲谱中的音符对应的频率值依次记录下来(B 调、四二拍),时间表是将各个音符发音的相对时间记录下来(由曲谱中节拍得出)。

频率表和时间表是一一对应的,频率表的最后一项为 0,作为重复的标志。根据频率表中的频率算出对应的计数初值,然后依次写入 8254 的计数器。将时间表中相对时间值带入延时

程序来得到音符演奏时间。实验参考程序流程如图 3-1 所示。

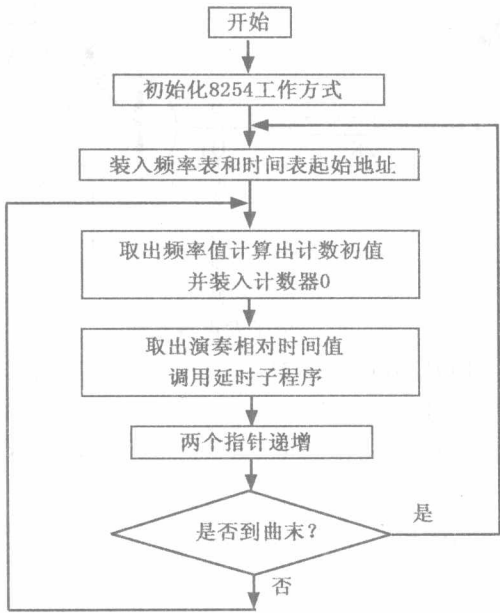


图 3-1 实验参考流程图

实验步骤如下：

1. 参考图 3-2 所示连接实验线路；
2. 编写实验程序,经编译、连接无误后装入系统；
3. 运行程序,听扬声器发出的音乐是否正确。电子发声器原理如图 3-3 所示。

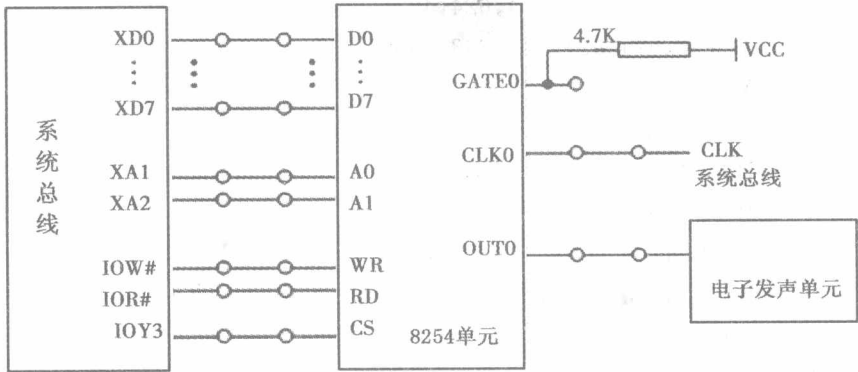


图 3-2 8254 电子发声实验接线图

实验参考例程(SOUND)

；端口定义

IOY0

EQU 06C0H

MY8254_COUNT0

EQU IOY0+00H

；8254 计数器 0 端口地址

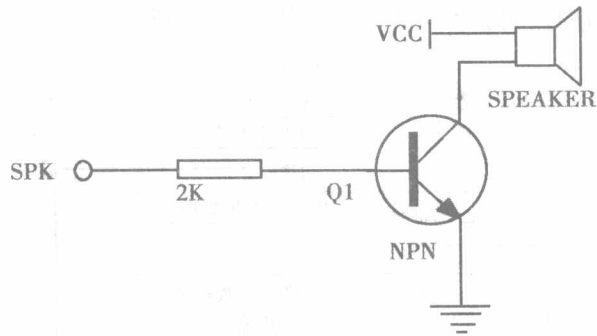


图 3-3 电子发声器原理图

```

MY8254_COUNT1    EQU IOY0+02H    ;8254 计数器 1 端口地址
MY8254_COUNT2    EQU IOY0+04H    ;8254 计数器 2 端口地址
MY8254_MODE      EQU IOY0+06H    ;8254 控制寄存器端口地址

```

```

STACK1    SEGMENT STACK
           DW 256 DUP(?)
STACK1    ENDS

```

```

DATA      SEGMENT
FREQ_LIST DW 371,495,495,495,624,556,495,556,624    ;频率表
           DW 495,495,624,742,833,833,833,742,624
           DW 624,495,556,495,556,624,495,416,416,371
           DW 495,833,742,624,624,495,556,495,556,833
           DW 742,624,624,742,833,990,742,624,624,495
           DW 556,495,556,624,495,416,416,371,495,0
TIME_LIST DB 4, 6, 2, 4, 4, 6, 2, 4, 4    ;时间表
           DB 6, 2, 4, 4, 12, 1, 3, 6, 2
           DB 4, 4, 6, 2, 4, 4, 6, 2, 4, 4
           DB 12, 4, 6, 2, 4, 4, 6, 2, 4, 4
           DB 6, 2, 4, 4, 12, 4, 6, 2, 4, 4
           DB 6, 2, 4, 4, 6, 2, 4, 4, 12

```

```
DATA    ENDS
```

```

CODE      SEGMENT
          ASSUME CS:CODE,DS:DATA

```

```

START:    MOV AX,DATA
          MOV DS,AX

```