

东南大学

《微机实验及课程设计》

实验报告

实验五 8253（8254）计数器/定时器

姓 名： 邹滨阳 学 号： 08022305

专 业： 自动化 实 验 室： 金智楼 416

实验时间： 2024 年 4 月 16 日 报告时间： 2024 年 4 月 16 日

评定成绩： _____ 审阅教师： _____

实验五 8253 (8254) 计数器/定时器

一. 实验目的与内容

实验目的:

- 1) 掌握计数器/定时器 8253/8254 的基本工作原理和编程应用方法;
- 2) 了解掌握 8253 (8254) 的计数器/定时器典型应用方法。

实验内容:

- 1) 按要求连接电路, 将计数器 0 设置为方式 0, 计数器初值为 N ($N \leq 0FH$), 用手动逐个输入单脉冲, 编程使计数值在屏幕上显示, 并同时用逻辑笔观察 OUT0 电平变化 (当输入 $N+1$ 个脉冲后 OUT0 变高电平)。
- 2) 按要求连接电路, 将计数器 0、计数器 1 分别设置为方式 2 和方式 3, 计数初值设为 1000, 用逻辑笔观察 OUT1 输出电平的变化 (频率 1Hz)。

二. 基本实验原理

1, 8253/8254 计数器/定时器

8253/8254 是一种常用的计数器和定时器, 它有三个独立的 16 位计数器, 可用于各种计时和计数应用。

2, 直接 I/O 操作

通过直接访问 8253/8254 的寄存器, 程序员可以控制计数器的工作模式和计数值, 以及观察计数器的输出电平变化。

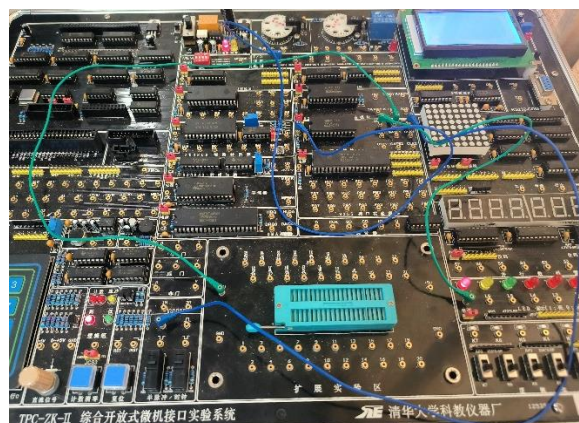
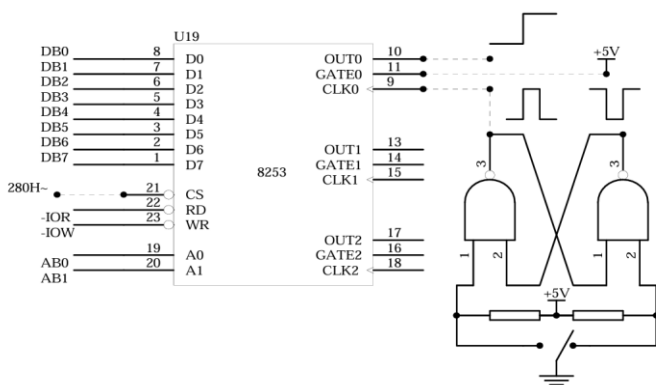
3, 编程应用

根据不同的工作方式和初值设置, 8253/8254 可以实现各种计数和定时功能, 如频率计数、脉冲计数和定时器应用。

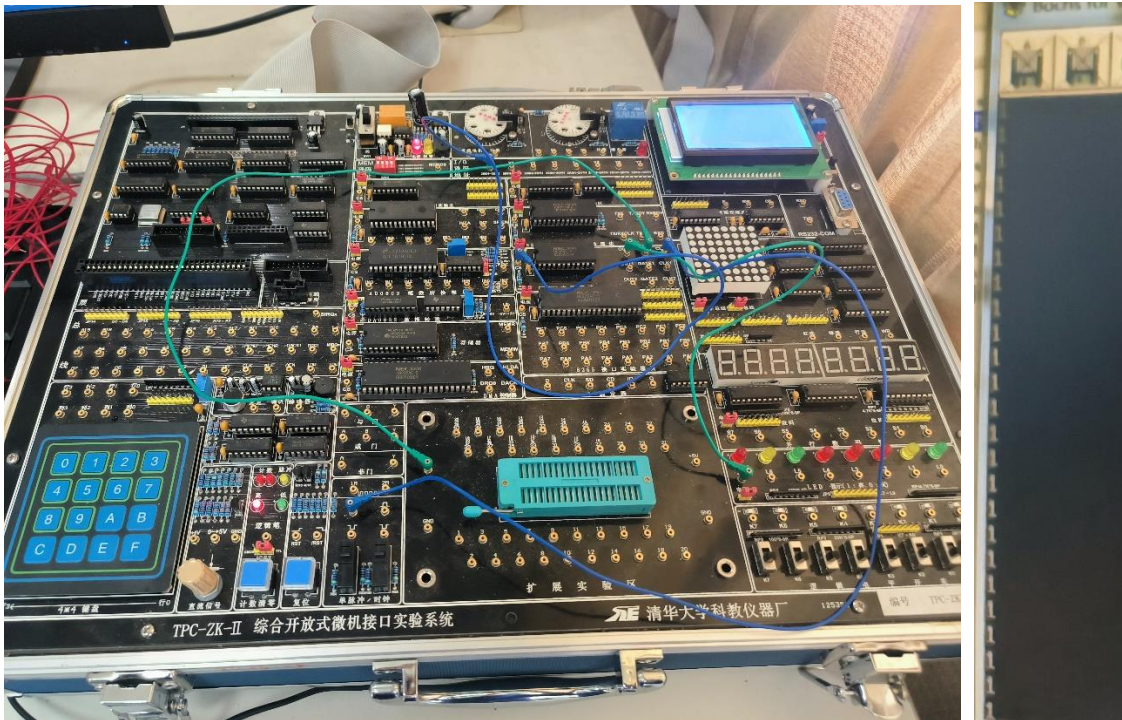
三. 方案实现与测试

1, 8253 (8254) 计数器

参考图 3.6-1 虚线连接电路, 将计数器 0 设置为方式 0, 计数器初值为 N ($N \leq 0FH$), 用手动逐个输入单脉冲, 编程使计数值在屏幕上显示, 并同时用逻辑笔观察 OUT0 电平变化 (当输入 $N+1$ 个脉冲后 OUT0 变高电平)



运行程序, 检查程序效果:

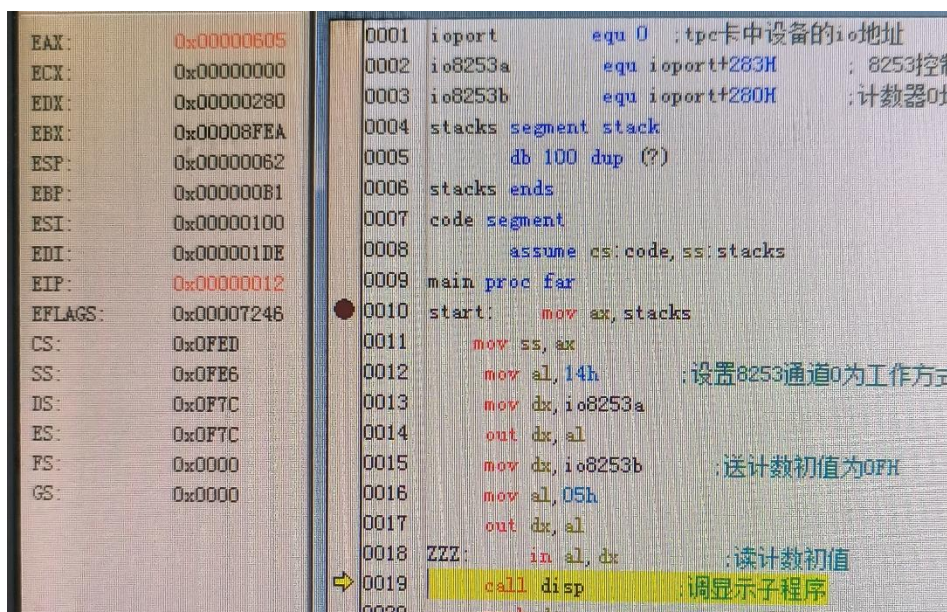


发现程序运行成功，最开始显示的原本锁存的值，按下一次后转变为设定的值，接着按下后便开始逐次递减，期间灯泡一直保持点亮状态，直到等到结果变为1后，灯泡熄灭，统计次数一共按下了N+1次后小灯熄灭，与源程序存在出入，实际情况是输入N+1次脉冲后，output变为低电平

完成对程序的调试：

EAX:	0x0000F03	0001	ioport	equ 0	:tpc卡中设备的io地址
ECX:	0x00000000	0002	io8253a	equ ioport+283H	:8253通道0的I/O地址
EDX:	0x00000280	0003	io8253b	equ ioport+280H	:8253通道1的I/O地址
EBX:	0x00008FEA	0004	stacks segment stack		
ESP:	0x00000062	0005	db 100 dup (?)		
EBP:	0x000000B1	0006	stacks ends		
ESI:	0x00000100	0007	code segment		
EDI:	0x000001DE	0008	assume cs:code, ss:stacks		
EIP:	0x00000012	0009	main proc far		
EFLAGS:	0x00007202	0010	start: mov ax, stacks		
CS:	0x0FED	0011	mov ss, ax		
SS:	0x0FE6	0012	mov al, 14h	:设置8253通道0为工作模式0	
DS:	0x0F7C	0013	mov dx, io8253a		
ES:	0x0F7C	0014	out dx, al		
FS:	0x0000	0015	mov dx, io8253b	:送计数初值为0Fh	
GS:	0x0000	0016	mov al, 05h		
		0017	out dx, al		
		0018	ZZZ: in al, dx	:读计数初值	
		0019	call disp	:调显示子程序	

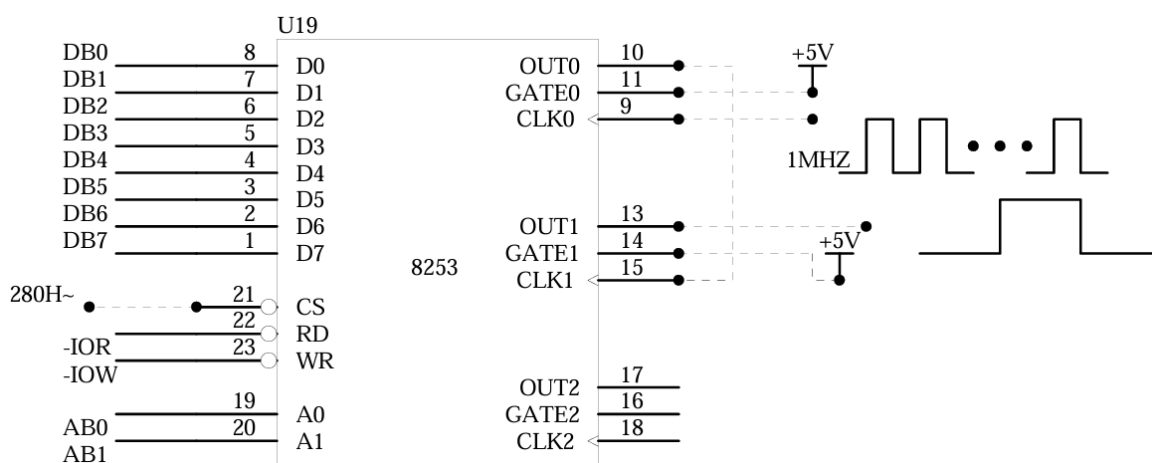
这里可以看到虽然我们往计数器中导入了5，但是从计数器第一次读取的结果是3，也就是上一次寄存器的终值。



但是当我们按下按钮产生脉冲后，读取的计数器的值就变成了 5，之后按下脉冲后，计数器的值也逐次递减，也说明了计数成功，符合流程图要求。

2, 8253 (8254) 定时器

按图 3.6-2 连接电路，将计数器 0、计数器 1 分别设置为方式 2 和方式 3，计数初值设为 1000，用逻辑笔观察 OUT1 输出电平的变化（频率 1Hz）。



可以观察大灯每隔 1 秒闪烁，证明实验成功呢。

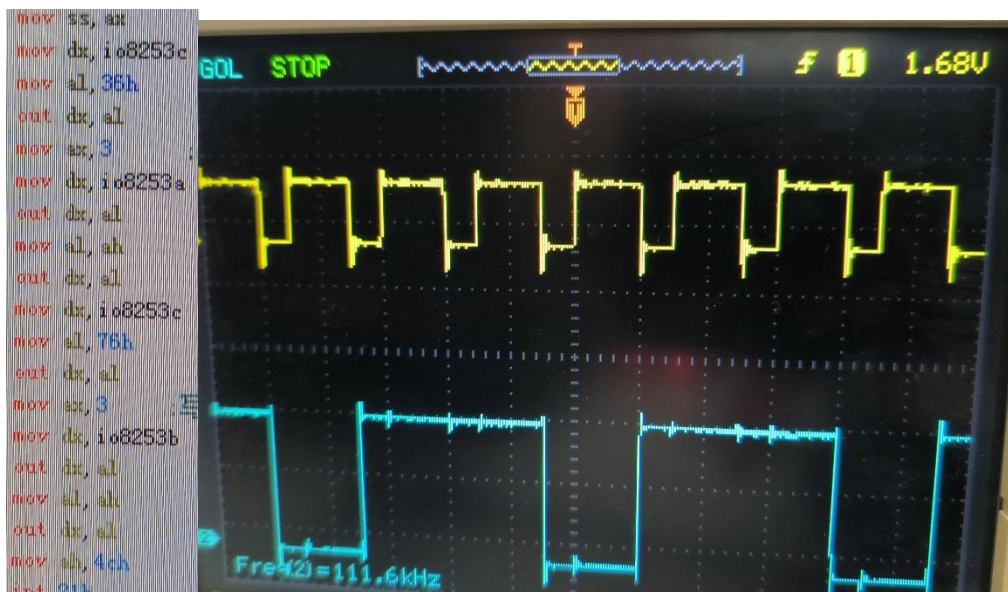
接下来用示波器验证并修改相关参数进行验证，重点观察两个计数的方式 3 定时器功能



修改参数为 2 和 2，发现结果仍然符合预期



修改参数为 3 和 2，发现结果符合方式 3 方波发生器的功能，也就是面对计数次数为 3，因为 $(3+1)/2=2$ ，所以 3，2 都是高电平，而 1 是低电平，符合要求



同样，面对当两个值为 3 和 3 时，结果也是正确的，符合要求



而当我们修改相关计数器的参数，比如修改计数的运行模式，从模式 3 改成模式 2 后，我们就发现波形从方波变成了负脉冲，证明了计数模式的改变，验证了结果的正确。

四. 提高与创新研究

使用 PC 微机内 8253 驱动声音接口。

IBM PC 机内扬声器（蜂鸣器）发声驱动系统如图 3.6-5 所示。由机内的 8255I/O 接口的 PB0 控制 8253 通道 2 的定时计数；PB1 来控制扬声器的接通和断开，以此来发声。PC 机内 8255PB 口地址为 61H，8253 通道 2 口地址为 42H，控制口为 43H。编程使扬声器发出 ‘1·2·3·4·5·6·7·1’（对应的频率分别为 262、294、330、347、392、440、494、523Hz）的数字乐声。

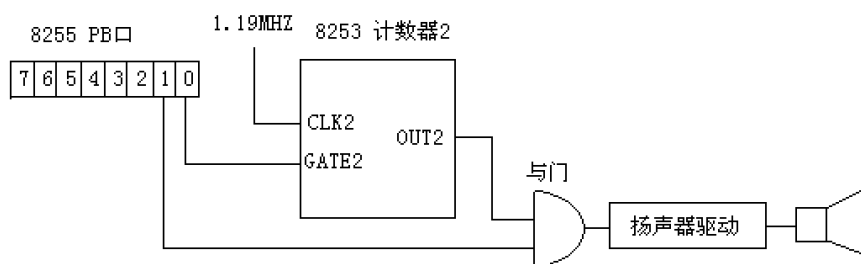


图3.6-5 IBM PC机内扬声器接线原理图

完成参考程序并运行

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

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

51592 + 464952 Bytes symbol space free

0 Warning Errors
0 Severe Errors
```

```
D:\TASM>link epiano.obj
Microsoft (R) Overlay Linker Version 3.60
Copyright (C) Microsoft Corp 1983-1987. All rights reserved.

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

```
D:\TASM>epiano.exe
23234322
```

运行成功

```
dseg segment
    freq dw 262, 294, 330, 349, 392, 440, 494, 523 ;扬声器发声各频率值
    tim dw 400
dseg ends

sseg segment stack
    dw 256 dup(?)
sseg ends

cseg segment
main proc far
    assume cs:cseg, ds:dseg
start:mov ax, dseg
    mov ds, ax
    mov si, 0
again:mov di, freq[si]
    mov bx, tim
    call gen
    add si, 2
    cmp si, 16
    jnz again
    mov ax, 4c00h

    int 21h
main endp
gen proc near
    mov al, 0b6h ;设置 8253 通道 2 方式 3 二进制计数
    out 43h, al
    mov dx, 18
    mov ax, 10352 ;由 dx*65536+ax=1.19M 可得 dx=18, ax=10352
    div di
    out 42h, al ;8253 通道 2 口设置初值
    mov al, ah
    out 42h, al
    in al, 61h ;读 8255PB 口
    mov ah, al
    or al, 3
    out 61h, al
    wait1: mov cx, 0h
    delay: mov ax, 10 ;延时子程序
    delay2: nop
    dec ax
    jnz delay2
    loop delay
    dec bx
    jnz wait1
    mov al, ah
    out 61h, al
    ret
gen endp
cseg ends
end start
```

1. **数据段定义** 在数据段，**freq** 数组定义了音调的频率值，这些值是频率值的序列，代表了不同音调的频率。**tim** 变量可能是用来控制每个音调的持续时间的计数器。
2. **代码段开始** 在代码开始部分，首先初始化了数据段寄存器 DS，确保程序可以正确访问数据段。接着，索引寄存器 SI 被初始化为 0，用于遍历音调频率数组。
3. **音调生成循环** 这部分代码是程序的主要循环。它遍历 **freq** 数组中的每一个频率值，并对每一个值调用 **gen** 过程。这个过程的目的是为了根据给定的频率生成对应的音调。
4. **退出程序** 这段代码负责程序的正常退出。它使用 DOS 系统调用来结束程序执行。
5. **生成音调的过程** **gen** 过程是整个程序的核心部分。在这里，通过 8253 计数器来生成音调。这部分代码首先设置 8253 计数器的工作模式，然后计算并设置计数器的初值，最后通过 I/O 端口控制计数器开始计数。这样，当计数器工作时，它会以特定的频率发出脉冲，这些脉冲经过扬声器转换为声音。
6. **延时和音调持续时间** 这部分代码负责控制每个音调的持续时间。通过简单的延时循环，程序可以控制每个音调的持续时间，从而控制音乐的节奏和时长。

总结，这段代码主要利用 8253 计数器来生成一系列预定义频率的音调。通过遍历预定义的频率数组并为每个频率调用生成音调的过程，程序能够生成一个音乐序列。同时，通过延时循环控制每个音调的持续时间，从而形成完整的音乐片段。

五. 分析与总结

本次实验的目的是通过实践掌握计数器/定时器 8253/8254 的基本工作原理和编程应用方法, 同时了解掌握 8253 (8254) 的计数器/定时器典型应用方法。在实验过程中, 我们按照要求连接电路, 成功地进行了两个主要的实验部分。

1. 8253 (8254) 计数器部分

在这一部分, 我们设置了计数器 0 为方式 0, 并将计数初值设置为 5。经过实验, 我们发现在输入 $N+1$ 次脉冲后, OUT0 电平变为低电平, 与预期有出入。经过对程序的调试和分析, 我们发现了计数器初值与实际输出结果之间的不匹配问题。通过适当的调整和修正, 我们成功地使计数器按照预期工作。

2. 8253 (8254) 定时器部分

在这一部分, 我们将计数器 0 设置为方式 2 和计数器 1 设置为方式 3, 计数初值都设为 1000。通过观察, 我们验证了 OUT1 输出电平的变化频率为 1Hz。接着, 我们进行了参数修改实验, 验证了不同计数模式和参数对输出波形的影响, 结果符合预期。

提高与创新研究

我们还利用 8253 计数器实现了音乐生成功能, 通过设置不同的频率值和持续时间, 成功地生成了一个音乐序列。这部分实验不仅提高了我们对 8253 计数器应用的理解, 还拓展了实验的应用范围, 增加了实验的趣味性。

实验的局限性与改进方向

在实验过程中, 我们遇到了一些问题, 如计数器初值与实际输出结果不匹配。这些问题主要由于程序编写或硬件连接错误引起。在未来的实验中, 我们可以更加细致地检查硬件连接, 提高程序的稳定性和准确性。

实验目的达成情况评价

通过本次实验, 我们成功地掌握了计数器/定时器 8253/8254 的基本工作原理和编程应用方法, 了解并掌握了其典型应用方法。同时, 我们还拓展了实验内容, 增加了音乐生成功能, 提高了实验的趣味性和实用性。

总结

本次实验不仅加强了我们 8253/8254 计数器/定时器的理论知识和应用能力, 还培养了我们的实践操作技能。通过实验, 我们对微机计数器/定时器的工作原理和应用有了更深入的理解, 这对我们今后的学习和工作都具有重要的意义。