

# 实验四 中断控制及计数/定时电路应用

## 实验报告

姓名：\_\_\_\_\_ 赵文亮 \_\_\_\_\_

学号：\_\_\_\_\_ 2016011452 \_\_\_\_\_

班级：\_\_\_\_\_ 自 64 \_\_\_\_\_

日期：\_\_\_\_\_ 2018 年 5 月 21 日 \_\_\_\_\_

# 目录

<b>1</b>	<b>实验目的</b>	<b>1</b>
<b>2</b>	<b>实验准备</b>	<b>1</b>
<b>3</b>	<b>实验内容</b>	<b>1</b>
3.1	8259 中断管理电路 . . . . .	1
3.2	8253 计数电路 . . . . .	1
3.3	选做任务 . . . . .	2
<b>4</b>	<b>实验程序及说明</b>	<b>2</b>
4.1	8259 中断管理电路 . . . . .	2
4.2	8253 计数电路 . . . . .	4
4.3	选做任务 . . . . .	5
<b>5</b>	<b>完成情况和心得体会</b>	<b>8</b>

## 1 实验目的

理解中断控制器 8259 及计数/定时器 8253 的工作原理，掌握其使用方法。

## 2 实验准备

### 1. 复习 8259 及 8253 的工作原理及应用。

8259 是可编程中断控制电路。实验中对 8259A 的操作主要包括以下几个步骤：

- 编写中断服务程序。中断服务程序结束时需要通过 IRET 返回。
- 保护中断矢量。使用 DOS 功能调用，将实验中所用的中断类型号对应的中断矢量提取出来并保存。
- 装入新中断矢量。通过伪操作获得中断服务程序的段基址和偏移地址，并通过 DOS 功能调用装入新中断矢量。
- 初始化完成后，需要消除屏蔽并打开中断。
- 程序退出前，需要恢复屏蔽，并恢复中断矢量。

8253 是可编程硬件定时/计数器。其内部有 3 个独立的计数器通道，每个通道有六种工作方式。本实验中可以使用方式 3（方波发生器）来实现编程。8253 编程的关键是写控制字，其格式如下：

- $D_7D_6$ ：计数器。00、01、10 分别表示计数器 0、1、2；11 表示非法选择。
- $D_5D_4$ ：读写格式。00、10、01、11 分别表示计数器锁存、只读写高八位、只读写低八位、先读写低八位再读写高八位。
- $D_3D_2D_1$ ：工作方式。000、001、X10、X11、100、101 分别表示方式 0~5。
- $D_0$ ：数制。0、1 分别表示二进制和十进制（压缩 BCD）。

### 2. 自学附录中 TPC\_USB 通用微机接口实验系统介绍及集成软件开发环境的相关内容。

### 3. 按实验内容要求编写汇编语言源程序。

## 3 实验内容

### 3.1 8259 中断管理电路

用一根导线将单脉冲信号引至 IRQ 插孔，按一下按键，通过程序在屏幕上显示一行提示信息，10 次之后退回 DOS（两次按键之间要有 0.5s 时间间隔）。

### 3.2 8253 计数电路

将 8253 的 CNT0 和 CNT1 串起来实现 106 分频，即将实验台上的 1 MHz 时钟信号接 CNT 0 的时钟输入端 CLK 0，其输出端 OUT0 再接 CNT1 的 CLK1，GATE0 和 GATE1 都接 +5V，CNT0 和 CNT1 全工作于方式 3，由 OUT1 输出 1Hz 的方波。

### 3.3 选做任务

将实验内容 1 和 2 合并，用 8253 的 OUT1 接 IRQ 发中断请求，其它要求同原来相同。无需任何输入，程序在屏幕上显示 10 次提示信息后自动退出。

## 4 实验程序及说明

### 4.1 8259 中断管理电路

汇编程序如下，实验中将单脉冲接入 IRQ3 插孔。流程图如图 1 所示。

---

```
1 NAME    INTERRUPT
2 DATA    SEGMENT
3 KEEPIRQ DB 0
4 KEEP_CS DW 0
5 KEEP_IP DW 0
6 MESS     DB 'INTERUPT ONCE! ', 0DH, 0AH, '$'
7 DONE     DB 'PROGRAM DONE', 0DH, 0AH, '$'
8 DATA    ENDS
9
10 STACK   SEGMENT PARA STACK
11         DB 100 DUP(?)
12 STACK   ENDS
13
14 CODE     SEGMENT
15         ASSUME CS: CODE, DS: DATA, ES: DATA, SS: STACK
16 INTR     PROC
17         LEA DX, MESS;    中断服务程序
18         MOV AH, 9H
19         INT 21H
20         MOV AL, 20H
21         OUT 20H, AL
22         INC CX
23         IRET
24 INTR     ENDP
25 START:   MOV AX, DATA
26         MOV DS, AX
27         MOV CX, 0
28
29         MOV AH, 35H;    保护中断矢量
30         MOV AL, 0BH
31         INT 21H
32         MOV KEEP_IP, BX
33         MOV KEEP_CS, ES
34
35         PUSH DS;        装入新中断矢量
36         MOV DX, OFFSET INTR
```

```

37      MOV AX, SEG INTR
38      MOV DS, AX
39      MOV AH, 25H
40      MOV AL, 0BH
41      INT 21H
42      POP DS
43
44      IN AL, 21H;      消除屏蔽
45      MOV KEEPIRQ, AL
46      AND AL, 11110111B
47      OUT 21H, AL
48
49      STI;            打开中断
50
51 WAITINT: CMP CX, 10
52          JNZ WAITINT
53
54      LEA DX, DONE;    输出提示信息
55      MOV AH, 9H
56      INT 21H
57
58      MOV AL, KEEPIRQ; 恢复屏蔽
59      OUT 21H, AL
60
61      PUSH DS;         恢复中断矢量
62      MOV DS, KEEP_CS
63      MOV DX, KEEP_IP
64      MOV AH, 25H
65      MOV AL, 0BH
66      INT 21H
67      POP DS
68
69      MOV AH, 4CH
70      INT 21H
71 CODE   ENDS
72 END     START

```

---

对代码分析如下：

- 16-24：中断服务程序。其中输出一行信息，并将 **CX** 递增。
- 29-33：保护中断矢量，其中 **0BH** 为 **IRQ3** 对应的中断类型号。将对应的原中断矢量保存在内存中。
- 35-42：装入新中断矢量，使用 **DOS** 功能调用，将中断服务程序的段基址和偏移地址装入中断矢量表。
- 44-47：消除屏蔽，先获得屏蔽字，再将 **IRQ3** 对应的位打开，重写屏蔽字。
- 51-52：主循环，当 **CX** 小于 10 时无限循环等待中断。

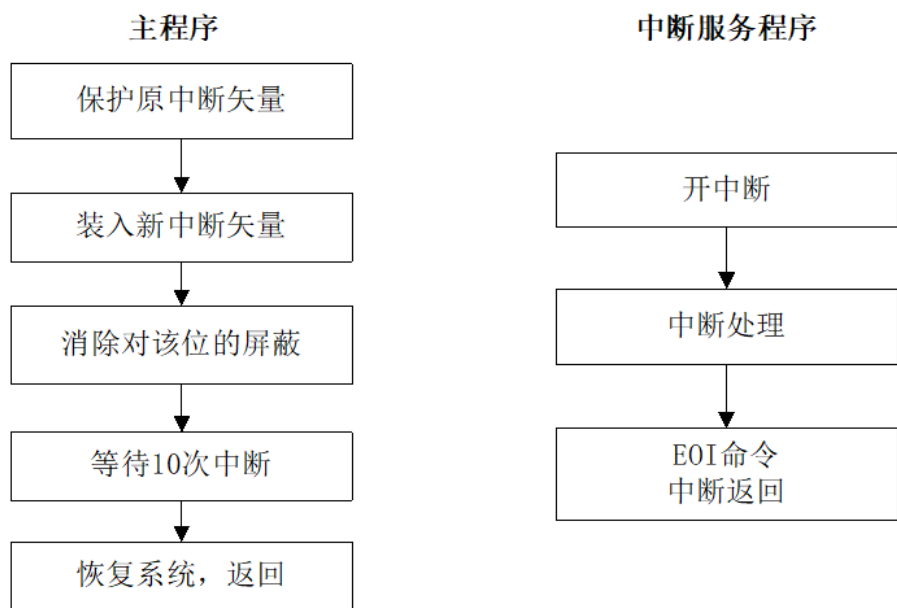


图 1: 8259 中断管理电路流程图

- 54-56: 输出提示信息。
- 58-59: 恢复中断屏蔽字。
- 61-67: 恢复中断矢量。

## 4.2 8253 计数电路

代码如下所示，其中对计数器 0 和计数器 1 的编程完全类似。

```

1 NAME    INTERRUPT
2 DATA   SEGMENT
3 DONE    DB 'PROGRAM DONE', 0DH, 0AH, '$'
4 DATA   ENDS
5
6 STACK   SEGMENT PARA STACK
7         DB 100 DUP(?)
8 STACK   ENDS
9
10 CODE    SEGMENT
11         ASSUME CS: CODE, DS: DATA, ES: DATA, SS: STACK
12
13 START:  MOV AX, DATA
14         MOV DS, AX
15
16         MOV DX, 283H
17         MOV AL, 00110111B; 计数器0、先低字节后高字节、方式3、十进制
18         OUT DX, AL;      CNT0

```

```

19
20     MOV DX, 280H
21     MOV AL, 00H
22     OUT DX, AL
23     MOV AL, 10H
24     OUT DX, AL
25
26     MOV DX, 283H
27     MOV AL, 01110111B;   计数器1、先低字节后高字节、方式3、十进制
28     OUT DX, AL;         CNT1
29
30     MOV DX, 281H
31     MOV AL, 00H
32     OUT DX, AL
33     MOV AL, 10H
34     OUT DX, AL
35
36     LEA DX, DONE
37     MOV AH, 9H
38     INT 21H
39     MOV AH, 4CH
40     INT 21H
41
42 CODE ENDS
43 END START

```

---

8253 编程的流程图如图 ?? 所示。分析代码如下：

- 16-18：写 CNT1 控制字。其中 283H 是控制字寄存器的地址。设定 CNT0 位方式 3 工作方式，初值通过先低字节后高字节以十进制写入。
- 20-24：按照先低字节后高字节的顺序写入初值。其中 280H 是 CNT0 的地址。
- 25-28：写 CNT1 控制字。设定 CNT1 位方式 3 工作方式，初值通过先低字节后高字节以十进制写入。
- 30-34：按照先低字节后高字节的顺序写入初值。其中 281H 是 CNT1 的地址。

实验接线时将实验台上的 1 MHz 时钟信号接 CNT 0 的时钟输入端 CLK 0，其输出端 OUT0 再接 CNT1 的 CLK1，GATE0 和 GATE1 都接 +5V，CNT0 和 CNT1 全工作于方式 3。只要对计数器 0 和计数器 1 编程分别实现 1000 分频，则由 OUT1 就可以输出 1Hz 的方波。8253 芯片的片选端需要接入到实验台上 280H~287H 的地址译码。

### 4.3 选做任务

选做任务是两个必做任务的结合，代码上只需要将两个代码合在一起，即先对 8253 编程，再对 8259 编程，最后进入到主循环中等待中断。接线只需将 8253 的 CNT1 改接到 IRQ3 上即可。

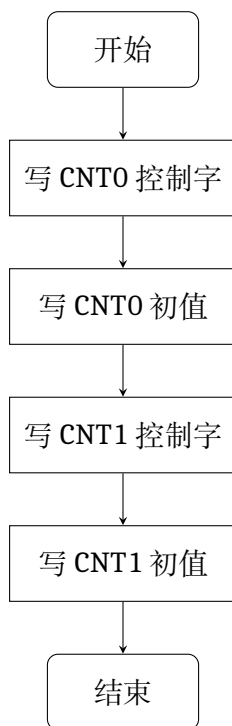


图 2: 8253 计数/定时电路流程图

---

```

1 NAME    INTERRUPT
2 DATA   SEGMENT
3 KEEPIRQ DB 0
4 KEEP_CS DW 0
5 KEEP_IP DW 0
6 MESS    DB 'INTERUPT ONCE! ', 0DH, 0AH, '$'
7 DONE    DB 'PROGRAM DONE', 0DH, 0AH, '$'
8 DATA   ENDS
9
10 STACK  SEGMENT PARA STACK
11        DB 100 DUP(?)
12 STACK  ENDS
13
14 CODE   SEGMENT
15        ASSUME CS: CODE, DS: DATA, ES: DATA, SS: STACK
16
17 INTR   PROC
18        LEA DX, MESS;          中断服务程序
19        MOV AH, 9H
20        INT 21H
21        MOV AL, 20H
22        OUT 20H, AL
23        INC CX
24        IRET
  
```



```

25 INTR ENDP
26
27 START: MOV AX, DATA
28         MOV DS, AX
29
30         MOV DX, 283H; 8253
31
32         MOV AL, 00110111B; 计数器0、先低字节后高字节、方式3、十进制
33
34         OUT DX, AL; CNT0
35         MOV DX, 280H
36         MOV AL, 00H
37         OUT DX, AL
38         MOV AL, 10H
39         OUT DX, AL
40
41         MOV DX, 283H
42         MOV AL, 01110111B; 计数器1、先低字节后高字节、方式3、十进制
43
44         OUT DX, AL; CNT1
45
46         MOV DX, 281H
47         MOV AL, 00H
48         OUT DX, AL
49         MOV AL, 10H
50         OUT DX, AL
51
52         MOV CX, 0; 8259
53
54         MOV AH, 35H; 保护中断矢量
55         MOV AL, 0BH
56         INT 21H
57         MOV KEEP_IP, BX
58         MOV KEEP_CS, ES
59
60         PUSH DS; 装入新中断矢量
61         MOV DX, OFFSET INTR
62
63         MOV AX, SEG INTR
64         MOV DS, AX
65         MOV AH, 25H
66         MOV AL, 0BH
67         INT 21H
68         POP DS
69
70         IN AL, 21H; 消除屏蔽
71         MOV KEEP_IRQ, AL

```

```

72      AND AL, 11110111B
73      OUT 21H, AL
74
75      STI;                      打开中断
76
77 WAITINT: CMP CX, 10
78          JNZ WAITINT
79      MOV AL, KEEP_IRQ;        恢复屏蔽
80      OUT 21H, AL
81
82      PUSH DS;                 恢复中断矢量
83      MOV DS, KEEP_CS
84      MOV DX, KEEP_IP
85      MOV AH, 25H
86      MOV AL, 0BH
87      INT 21H
88      POP DS
89
90      MOV AH, 4CH
91      INT 21H
92
93      LEA DX, DONE
94      MOV AH, 9H
95      INT 21H
96      MOV AH, 4CH
97      INT 21H
98
99 CODE ENDS
100 END START

```

---

## 5 完成情况和心得体会

本次实验整体来说比较顺利，我在较短的时间内就完成了实验。不过其实还是存在一些需要调试的小问题：

1. 最主要的问题是我写代码的时候使用了中文注释。之前的几次使用中文注释后，至多在电脑上显示乱码，而这次却发现，中文变成乱码后，会把后面的一部分代码都当做注释处理。我机房的 TPC-USB 的编辑器中发现代码的颜色和注释的颜色一样，发现了这个问题。为了解决这个问题，我不得不在每一行被注释掉的代码前添加空行，浪费了许多时间。
2. 一个小问题是一开始我的中断服务程序没有写成过程，而是只是在主程序里面写了一个标号，这就导致中断来临前中断服务程序也会执行一次。我马上发现了这个问题，于是我把中断服务程序改写为子程序的形式，顺利地将其解决。

在课上讲解 8259 和 8253 时，我掌握得并不是很好。但是通过实验的预习过程，我认真地查看教材中 8259

和 8253 的工作原理和编程方式，对相关概念有了更深的理解。由此可以看到计原实验的设置十分合理，能够让我们对课本上的知识点有更加清晰、形象的认识。

## 参考文献

- [1] 微型计算机系统及应用/杨素行等编著.—3 版.—北京：清华大学出版社,2009.4.ISBN 978-7-302-19352-4