

东南大学

## 《微机实验及课程设计》

### 实验报告

#### 实验四 简单输入输出

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 2 日 报告时间： 2024 年 4 月 5 日

评定成绩： \_\_\_\_\_ 审阅教师： \_\_\_\_\_

## 实验四 简单输入输出

### 一. 实验目的与内容

实验目的:

- 1) 进一步掌握 TPC 实验装置的基本原理和组成结构;
- 2) 掌握利用 I/O 指令单步调试检查硬件接口功能, 学会利用示波器检测 I/O 指令执行时总线情况;
- 3) 掌握简单并行输入输出接口的工作原理及使用方法, 进一步熟练掌握输入输出单元的功能和使用。
- 4) 掌握 PC 机中断处理系统的基本原理;
- 5) 学会编写中断服务程序

实验内容:

- 1) 配置简单输入输出接口电路连接线路。
- 2) 编写程序, 从键盘输入字符或数字, 通过输出接口输出其 ASCII 码, 并验证输出结果。
- 3) 使用逻辑电平开关预置 ASCII 码, 编写程序将其显示在屏幕上, 并验证。
- 4) 结合输入输出程序, 编写程序将逻辑电平开关状态输出到发光二极管上, 并以任意键结束。
- 5) 运行中断程序, 并理解中断程序的原理

### 二. 基本实验原理

在微型计算机系统中, 使用 DEBUG/TD 进行直接 I/O 操作是一种常见的调试和控制方法。这种方法允许程序员直接访问计算机的输入/输出端口, 以便进行与外部设备的通信和控制。

#### 1. DEBUG/TD 工具

DEBUG/TD 是一种动态调试程序, 可用于微处理器系统的调试和单步执行程序。它允许程序员通过汇编语言指令直接控制计算机的硬件, 并观察程序执行过程中内存和寄存器的变化。

#### 2. 直接 I/O 操作

并行输出操作: 在进行并行输出时, 程序员首先需要将数据字节加载到适当的寄存器中, 通常使用 MOV 指令完成这一步骤。然后, 使用 OUT 指令将数据送往指定的输出端口, 以与外部设备进行通信。

并行输入操作: 与并行输出类似, 进行并行输入时, 程序员首先使用 IN 指令从指定的输入端口读取数据, 并将其加载到适当的寄存器中, 以供程序进一步处理或显示。

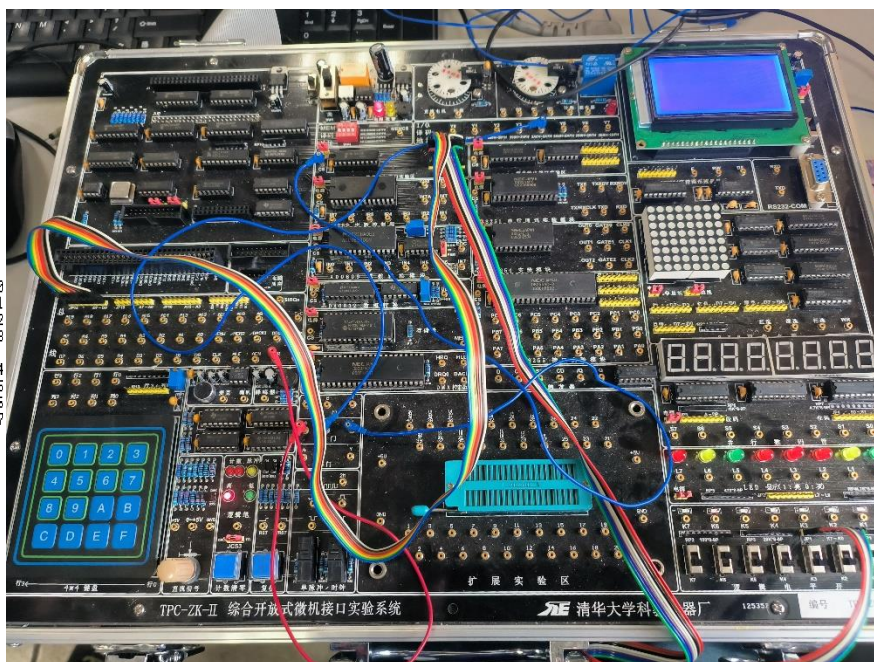
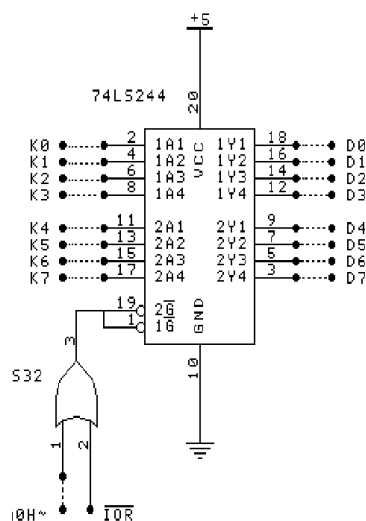
#### 3. 调试与控制

通过 DEBUG/TD 进行直接 I/O 操作, 程序员可以实时观察和调试程序的执行过程, 查看输入/输出端口的状态, 以及与外部设备进行交互。这种方法对于调试硬件接口功能或测试设备驱动程序非常有用。

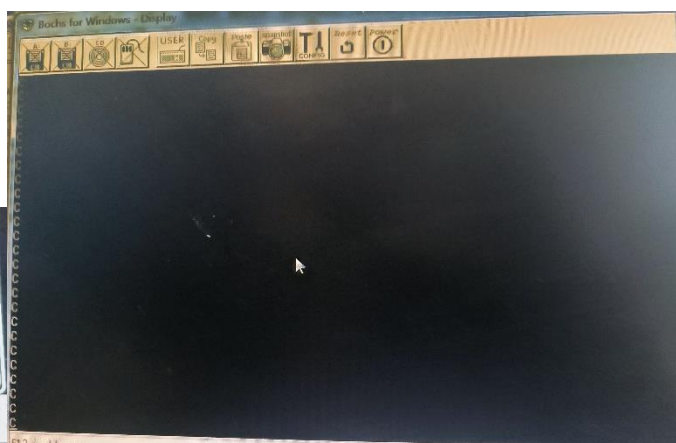
### 三. 方案实现与测试

#### 1, 实现简单并行输入接口的程序

首先是参考原理电路图完成对简单并行输入接口的硬件连接

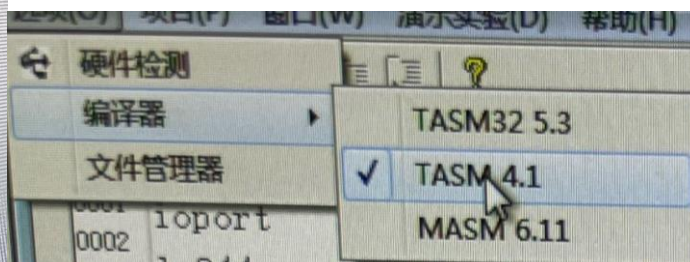
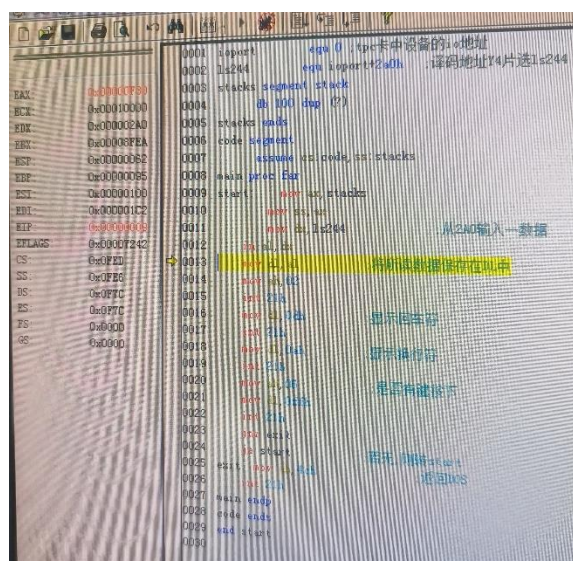


运行简单并行输入接口程序：



可以看到我们开关拨的结果是 01000011，也就是 43H 对应的 ASCII 码 “C” 说明程序输出正确  
调试简单并行输入接口程序：

首先是把代码输入到程序当中，同时注意把当前的环境进行调整，调整提供代码对应的编译器



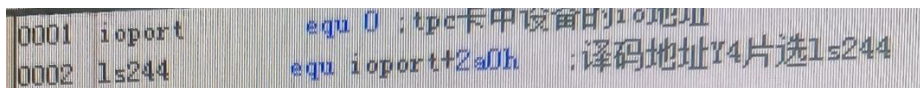


同时修改代码的开头部分，因为我们是直接在程序中运行的，所以可以不用管 ioport，直接设置 ioport 为 0 即可，这样就与我们的 I/O 译码地址符合了

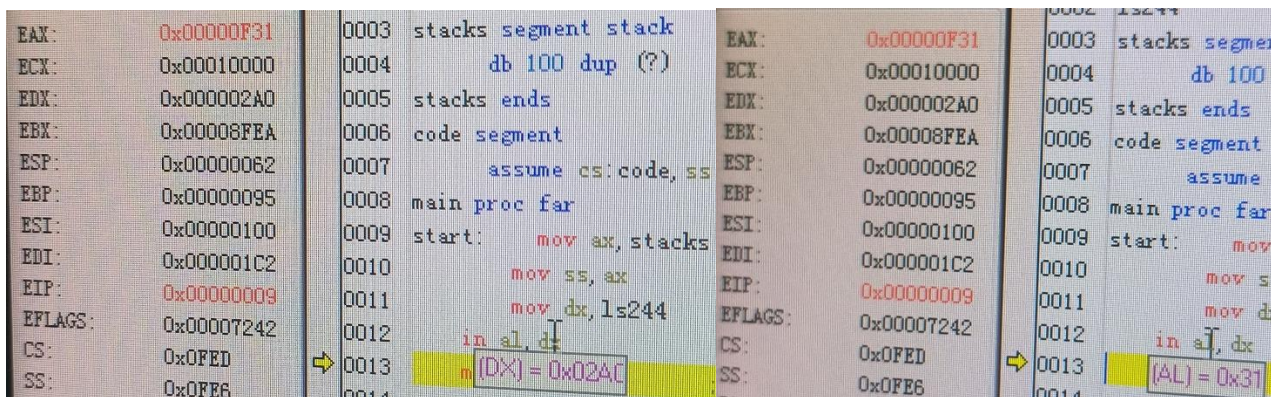


图是输出电路

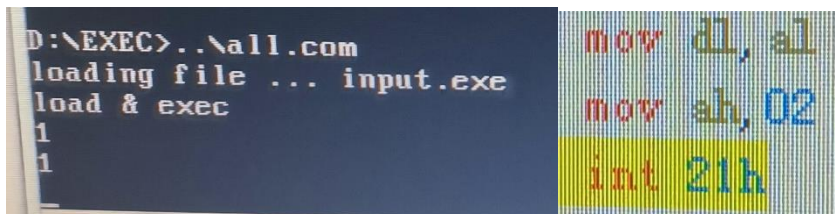
```
ioport equ 3100h-0280h ;tpc 卡中设备的 io 地址
ls244 equ ioport+2a0h ; 地址译码输出端片选 LS244
```



接下来进行调试，我们可以看到运行完输入 in al, dx 后 DX 是 02A0 没有变化，对应 I/O 译码输出，用于控制 74244 缓冲器，若 DX 不是 02A0，则 74244 关闭，CPU 无法读到数据。只有当 DX 是 2A0 并且 IOR 有效时，74244 才开启，CPU 才可以正常读取数据，于是我们可以看到在代码中，al 顺利读取到数据 0x31 也就是 31H，这时候开关的开闭为 00110001。

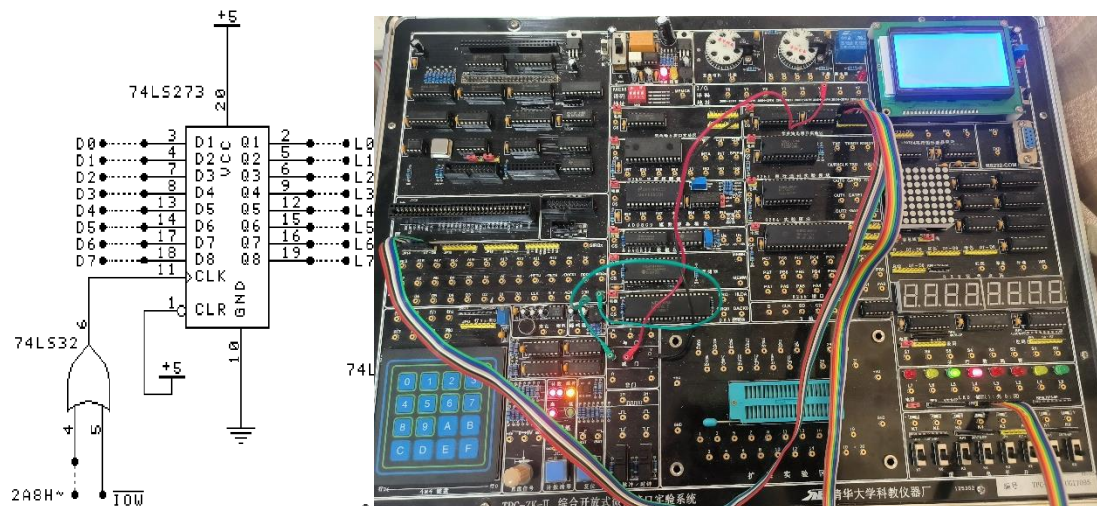


最后经过专门用于输出 ASCII 码的 INT 21 后，成功输出对应的结果，也就是 1



## 2. 实现简单并行输出接口的程序

首先是参考原理电路图完成对简单并行输出接口的硬件连接

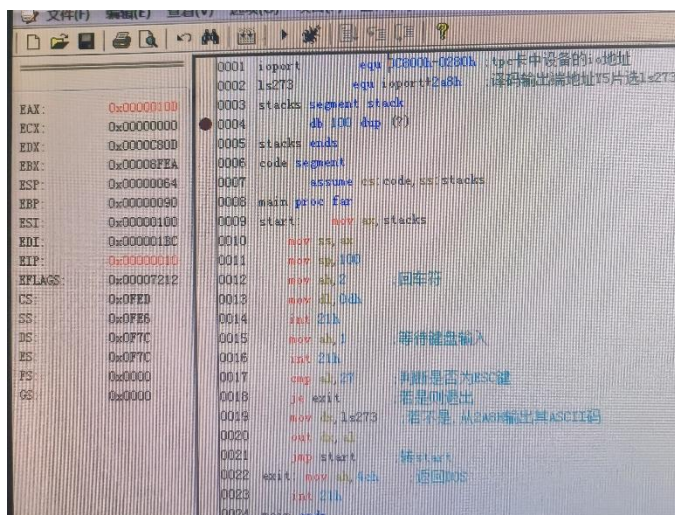




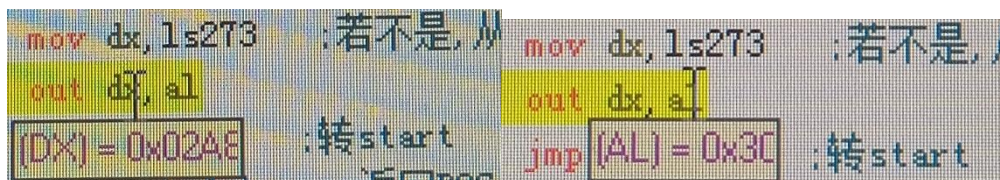
运行简单并行输入接口程序，我们输入‘1’，对应的 ASCII 码是 31H 也就是 00110001 可以看到灯的结果正确，输入‘2’，ASCII 码是 32H 也就是 00110010 也是正确的



接着是把代码输入到程序当中，同时注意把当前的环境进行调整，调整提供代码对应的编译器，同理也修改代码的开头部分。



接下来进行调试，在程序中输入 0，然后进行调试，可以看到 DX 的结果为 0x02A8，对应 I/O 译码输出，用于控制 74273 寄存器，若 DX 不是 02A8，则 74273 无法更新数据，外设也因此无法输出新数据。只有当 DL 是 2A8 并且 IOW 有效时，74273 才会读取 CPU 新输出的数据，外设才可以输出最新数据，于是我们可以看到在代码中，a1 顺利输出我们输入的数据‘0’对应的 ASCII 码为 0x30 也就是 30H，这时候灯泡的开闭为 00110000。



注意在运行这个 out 命令前，指示灯输出的 74273 寄存器中的数据，在我们这里就是‘6’ 36H



在 out 后, 74273 更新为 0, 也就是 30H



## 四. 提高与创新研究

### 中断系统研究

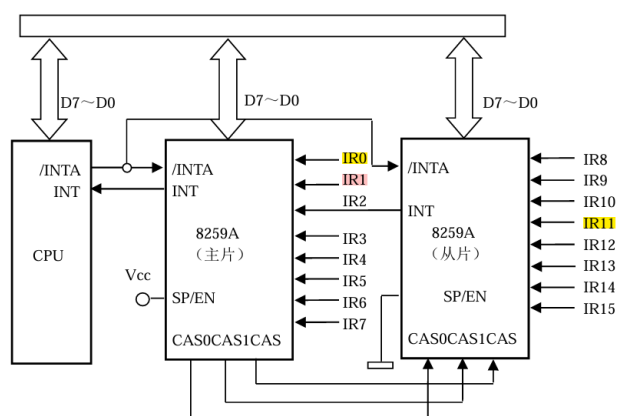
首先把所需要的文件比如编辑文件 edit.com, 查看基本信息的文件 portr.exe, 以及我们运行的文件 int.asm 等文件, 拷入 DOS 盘中, 注意是在自己拷入了哪个盘中。接着启动 DOS 操作系统, 用 DIR 指令查看硬盘中是否有我们拷入的文件, 最后用 cd 指令定位到对应文件夹。

接着运行文件 portr.exe, 可以得到以下信息

```
TPC pci card 9054 Chip I/O Base Address : 3000
TPC pci card I/O Base Address : 3100
TPC pci card Memory Base Address : F0000000
TPC pci card Interrupt Line : 000B
```

参考实验书上的数据, 以及反馈的信息 PCI card Interrupt Line: 000B, 我们可以看出这个使用的从片 IRQ11 中断源, 同时对应的主片接口是 IR2, 而这两个接口对应的位也应该置零, 从而能够实现中断功能, 也就是始能。而 I/O Base Address 为 3100, 但是在本程序中没有用到。最后 9054 的 Io 地址是 3000。以上部分均需要对实验代码中的数据进行更新

| 中断源          | 中断类型号       | 中断功能                       |
|--------------|-------------|----------------------------|
| IRQ8         | 070H        | 实时时钟                       |
| IRQ9         | 071H        | 用户中断                       |
| IRQ10        | 072H        | 保留                         |
| <b>IRQ11</b> | <b>073H</b> | 保留, 本实验使用 IRQ 11 (0BH) 中断源 |
| IRQ12        | 074H        | 保留                         |
| IRQ13        | 075H        | 协处理器                       |
| IRQ14        | 076H        | 硬盘                         |
| IRQ15        | 077H        | 保留                         |



参考实验书中的批注信息对代码中相关数据进行更新

int\_vect EQU **073H** ;中断向量, IRQ0-7 对应 08h-0fH, 8-1570h-77H

irq\_mask\_2\_7 equ 011111**0**11b ;主 8259 中断掩码, IRQ0-7 时相应位为 0, IRQ8-15 时 BIT2=0 (级联)

irq\_mask\_9\_15 equ 011111**0**111b ;从 8259, IRQ8-15 时从低至高相应位为 0

ioport\_cent equ 3000h ;tpc 卡中 9054 芯片的 io 地址

在 edit.com 更新 int.asm 后的数据如下

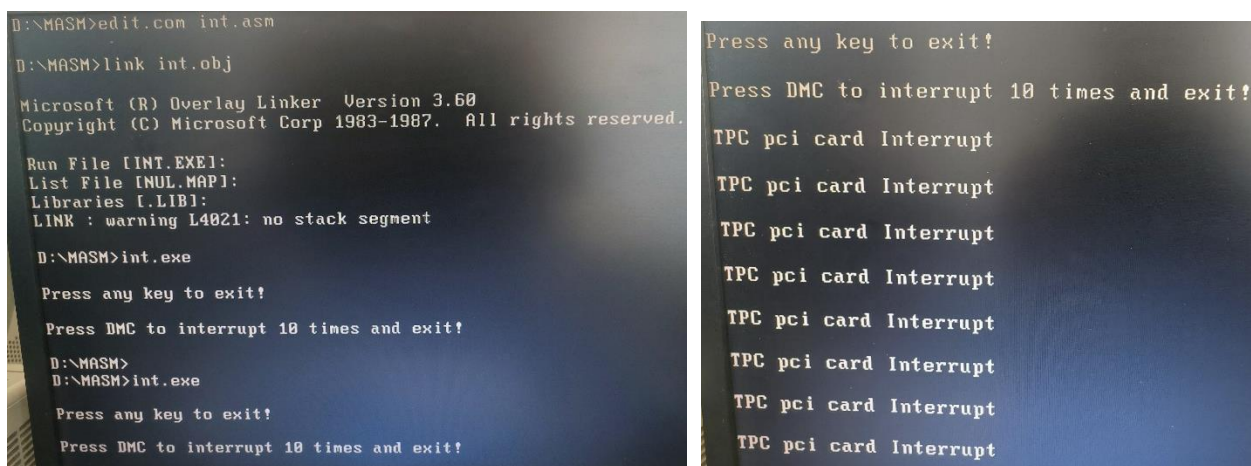


```

data segment
;ioport equ 3100H-280H ;tpc;io
int_vect EQU 073H ;;0-7 ;:08h-0fh, ;8-15 ;:70h-77h
irq_mask_2_7 equ 011111011b ;;0-7 ;;8-15 ;;70h-77h
irq_mask_9_15 equ 011110111b ;;0-7 ;;8-15 ;;70h-77h
ioport_cent equ 3000H;c ;;9054 ;;io
csreg dw ?
ipreg dw ? ;;
irq_times dw ? ;;
num dw ?
msg1 db 0dh,0ah,'TPC pci card Interrupt',0dh,0ah,'$'
msg2 db 0dh,0ah,'Press any key to exit!',0dh,0ah,'$'
msg3 db 0dh,0ah,'Press DMC to interrupt 10 times and exit!',0dh,0ah,'$'
data ends
stacks segment
db 100 dup (?)
stacks ends
code segment
assume cs:code,ds:data,ss:stacks,es:data
F1=Help

```

接着用 masm int.asm, link int.obj 完成对这个汇报程序的编译和运行, 最后连接电路, 运行程序, 查看效果, 发现效果良好



```

D:\MASM>edit.com int.asm
D:\MASM>link int.obj

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

Run File [INT.EXE]:
List File [NUL.MAP]:
Libraries [LIB]:
LINK : warning L4021: no stack segment

D:\MASM>int.exe

Press any key to exit!

Press DMC to interrupt 10 times and exit!

D:\MASM>
D:\MASM>int.exe

Press any key to exit!

Press DMC to interrupt 10 times and exit!

```

Press any key to exit!  
Press DMC to interrupt 10 times and exit!  
TPC pci card Interrupt  
TPC pci card Interrupt  
TPC pci card Interrupt  
TPC pci card Interrupt  
TPC pci card Interrupt  
TPC pci card Interrupt  
TPC pci card Interrupt  
TPC pci card Interrupt

总的来说这个代码的功能是首先保护了原中断向量将它们压入堆栈, 接着用在原位置设置了新中断向量, 并指向了我们新写的中断程序, 也就输出指定字符串并减一计数。同时我们需要加入设置 tpc 卡中 9054 芯片 io 口, 使能中断、设置中断掩码、恢复中断掩码关闭 9054 芯片 IO 口等必要的部分, 最后就能实现我们的目的。

注意 INT 与 CALL 的不同, call 本质上还是按照程序内部的顺序运行, 没法随时打断已经编译好程序, 只会在固定的地方运行, 同时无法很好的应对来自外部的信息, 而 int 则提供了另外一种方法, 也就是让外部给出中断请求, 而 cpu 一旦接收到这个中断请求就会执行相关的命令, 从而能够很好的应对外部设备的相关需求。因此 INT 可以随时在 CPU 执行任何指令的适合插入, 而 call 则做不到, 这就是两者的区别。

## 五. 分析与总结

本次实验旨在通过实践掌握 TPC 实验装置的基本原理和组成结构, 以及掌握简单输入输出接口的工作原理及使用方法, 同时学习 PC 机中断处理系统的基本原理和编写中断服务程序。以下是对实验过程的分析与总结:

实验内容完成情况:

在本次实验中, 我们完成了配置简单输入输出接口电路连接线路、编写并运行了简单的输入输出程序, 包括从键盘输入字符或数字, 通过输出接口输出其 ASCII 码, 使用逻辑电平开关预置 ASCII 码并显示在屏幕上, 以及将逻辑电平开关状态输出到发光二极管上等任务。同时, 我们还成功运行了中断程序并理解了中断程序的原理。

#### 实验结果验证：

通过运行程序并观察实验现象，我们验证了输入输出接口的功能正常，程序能够正确地从键盘输入字符或数字，并将其转换为相应的 ASCII 码进行输出，以及根据预置的逻辑电平开关状态进行灯光输出等功能。在运行中断程序时，我们也成功实现了对中断请求的响应，并执行了相应的中断服务程序。

#### 实验过程中的收获与问题：

通过本次实验，我们进一步加深了对微型计算机系统中直接 I/O 操作、调试与控制、中断处理系统等方面的理解。在实验过程中，可能会遇到硬件连接错误、程序编写错误等问题，但通过仔细检查和调试，这些问题都能得到解决。

#### 实验的局限性与改进方向：

实验中可能存在一些局限性，如硬件设备的性能限制、实验条件的限制等。为了进一步提高实验的可靠性和全面性，可以考虑增加更多的输入输出功能，扩展实验内容，同时提供更加完善的实验指导和调试支持。

#### 实验目的达成情况评价：

通过本次实验，我们达到了实验目的，掌握了 TPC 实验装置的基本原理和组成结构，熟悉了简单输入输出接口的工作原理和使用方法，理解了 PC 机中断处理系统的基本原理，并学会了编写中断服务程序。这些都对我们进一步深入学习计算机系统和嵌入式系统编程打下了良好的基础。

总的来说，本次实验不仅加强了理论知识的学习，也提高了实践操作能力，为我们今后的学习和工作打下了坚实的基础。