四 川 大 学 计 算 机 学 院、软 件 学 院

实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | 微机原理与接口技术 | 实验课时 | 4 |
| 实验项目 | 8259中断控制实验 | 实验时间 | 2024.6.3 |
| 实验目的 | 1. 掌握8259 中断控制器的工作原理。  2. 学习8259 的应用编程方法。  3. 掌握8259 级联方式的使用方法。 | | |
| 实验环境 | Wdm86 | | |
| 实验内容（算法、程序、步骤和方法） | 【基础实验要求】  1.单中断实验要求：单次脉冲输出与主片8259的IR7相连，每按动一次单次脉冲，产生一次外部中断，在显示屏上输出一个字符“7”。（可参考实验指导书4.2实验1）  2.级联实验要求：KK1+连接主片8259的IR7，KK2+连接从片8259的IR1，当按一次KK1+时，显示屏上显示字符“M7”，按一次KK2+时，显示字符“S1”，编写程序。（可参考实验指导书4.2实验2）  3.计数产生中断实验要求：编写程序，应用8254的计数功能，使用单次脉冲模拟计数，使每当按动“KK1+”5次后，通过8259产生一次计数中断，并在屏幕上显示一个字符“M”。（可参考实验指导书4.4实验1）  【扩展实验要求】  1.在显示屏上以1s的固定间隔持续输出特定的字符。 （可参考实验指导书4.4实验1和实验2）  2. 利用8255和8259a实现开关控制流水灯。（自行设计实验接线图，在8255基础实验2的流水灯循环显示的基础上，实现按一次KK1＋时，流水灯暂停；按一次KK1＋时，16位流水灯从暂停位置开始继续循环亮起。编写程序。）  3.在扬声器发声实验基础上，按动KK1+控制扬声器发声的开和关。  【实验步骤】  基础实验1：单中断实验  （1）按图连接实验线路。  C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps43.png  （2）编写实验程序，经编译、链接无误后装入系统，代码如下：  SSTACK SEGMENT STACK             DW 32 DUP(?)  SSTACK ENDS  CODE SEGMENT  *ASSUME* CS:CODE      START:*PUSH*   DS  *MOV*    AX, 0000H  *MOV*    DS, AX  *MOV*    AX, OFFSET MIR7    *;取中断入口地址*  *MOV*    SI, 003CH          *;中断矢量地址*  *MOV*    [SI], AX           *;填 IRQ7 的偏移矢量*  *MOV*    AX, CS             *;段地址*  *MOV*    SI, 003EH  *MOV*    [SI], AX           *;填 IRQ7 的段地址矢量*  *CLI*  *POP*    DS  *;初始化主片 8259*  *MOV*    AL, 11H  *OUT*    20H, AL            *;ICW1*  *MOV*    AL, 08H  *OUT*    21H, AL            *;ICW2*  *MOV*    AL, 04H  *OUT*    21H, AL            *;ICW3*  *MOV*    AL, 01H  *OUT*    21H, AL            *;ICW4*  *MOV*    AL, 6FH            *;OCW1*  *OUT*    21H, AL  *STI*      AA1:  *NOP*  *JMP*    AA1      MIR7: *STI*  *CALL*   DELAY  *MOV*    AX, 0137H  *INT*    10H                *;显示字符 7*  *MOV*    AX, 0120H  *INT*    10H  *MOV*    AL, 20H  *OUT*    20H, AL            *;中断结束命令*  *IRET*      DELAY:*PUSH*   CX  *MOV*    CX, 0F00H      AA0:  *PUSH*   AX  *POP*    AX  *LOOP*   AA0  *POP*    CX  *RET*  CODE ENDS  END START  （3）运行程序，重复按单次脉冲开关 KK1＋，显示屏会显示字符7，说明响应了中断  基础实验2：8259级联实验  （1）按图连接实验线路。  C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps44.png  （2）输入程序，编译、链接无误后装入系统，代码如下：  SSTACK SEGMENT STACK             DW 32 DUP(?)  SSTACK ENDS  CODE SEGMENT  *ASSUME* CS:CODE      START:*PUSH*   DS  *MOV*    AX, 0000H  *MOV*    DS, AX  *MOV*    AX, OFFSET MIR7    *;取中断入口地址*  *MOV*    SI, 003CH          *;中断矢量地址*  *MOV*    [SI], AX           *;填 IRQ7 的偏移矢量*  *MOV*    AX, CS             *;段地址*  *MOV*    SI, 003EH  *MOV*    [SI], AX           *;填 IRQ7 的段地址矢量*  *MOV*    AX, OFFSET SIR1  *MOV*    SI, 00C4H  *MOV*    [SI], AX  *MOV*    AX, CS  *MOV*    SI, 00C6H  *MOV*    [SI], AX  *CLI*  *POP*    DS  *;初始化主片 8259*  *MOV*    AL, 11H  *OUT*    20H, AL            *;ICW1*  *MOV*    AL, 08H  *OUT*    21H, AL            *;ICW2*  *MOV*    AL, 04H  *OUT*    21H, AL            *;ICW3*  *MOV*    AL, 01H  *OUT*    21H, AL            *;ICW4*  *;初始化从片 8259*  *MOV*    AL, 11H  *OUT*    0A0H, AL           *;ICW1*  *MOV*    AL, 30H  *OUT*    0A1H, AL           *;ICW2*  *MOV*    AL, 02H  *OUT*    0A1H, AL           *;ICW3*  *MOV*    AL, 01H  *OUT*    0A1H, AL           *;ICW4*  *MOV*    AL, 0FDH  *OUT*    0A1H,AL            *;OCW1 = 1111 1101*  *MOV*    AL, 6BH  *OUT*    21H, AL            *;主 8259 OCW1*  *STI*      AA1:  *NOP*  *JMP*    AA1      MIR7: *CALL*   DELAY  *MOV*    AX, 014DH  *INT*    10H                *;M*  *MOV*    AX, 0137H  *INT*    10H                *;显示字符 7*  *MOV*    AX, 0120H  *INT*    10H  *MOV*    AL, 20H  *OUT*    20H, AL            *;中断结束命令*  *IRET*      SIR1: *CALL*   DELAY  *MOV*    AX, 0153H  *INT*    10H                *;S*  *MOV*    AX, 0131H  *INT*    10H                *;显示字符 1*  *MOV*    AX, 0120H  *INT*    10H  *MOV*    AL, 20H  *OUT*    0A0H, AL  *OUT*    20H, AL  *IRET*      DELAY:*PUSH*   CX  *MOV*    CX, 0F00H      AA0:  *PUSH*   AX  *POP*    AX  *LOOP*   AA0  *POP*    CX  *RET*  CODE ENDS  END START  （3）运行程序，按动 KK1＋或 KK2＋，观察实验结果，验证实验程序的正确性。  基础实验3：计数产生中断实验  （1）按图接线。  C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps45.png  （2）编写实验程序，经编译、链接无误后装入系统，代码如下：  A8254 EQU 06C0H  B8254 EQU 06C2H  C8254 EQU 06C4H  CON8254 EQU 06C6H  SSTACK SEGMENT STACK             DW 32 DUP(?)  SSTACK ENDS  CODE SEGMENT  *ASSUME* CS:CODE, SS:SSTACK      START:*PUSH*   DS  *MOV*    AX, 0000H  *MOV*    DS, AX  *MOV*    AX, OFFSET IRQ7       *;取中断入口地址*  *MOV*    SI, 003CH             *;中断矢量地址*  *MOV*    [SI], AX              *;填 IRQ7 的偏移矢量*  *MOV*    AX, CS                *;段地址*  *MOV*    SI, 003EH  *MOV*    [SI], AX              *;填 IRQ7 的段地址矢量*  *CLI*  *POP*    DS  *;初始化主片 8259*  *MOV*    AL, 11H  *OUT*    20H, AL               *;ICW1*  *MOV*    AL, 08H  *OUT*    21H, AL               *;ICW2*  *MOV*    AL, 04H  *OUT*    21H, AL               *;ICW3*  *MOV*    AL, 01H  *OUT*    21H, AL               *;ICW4*  *MOV*    AL, 6FH               *;OCW1*  *OUT*    21H, AL  *;8254*  *MOV*    DX, CON8254  *MOV*    AL, 10H               *;计数器 0，方式 0*  *OUT*    DX, AL  *MOV*    DX, A8254  *MOV*    AL, 04H  *OUT*    DX, AL  *STI*      AA1:  *JMP*    AA1      IRQ7: *MOV*    DX, A8254  *MOV*    AL, 04H  *OUT*    DX, AL  *MOV*    AX, 014DH  *INT*    10H                   *;显示字符 M*  *MOV*    AX, 0120H  *INT*    10H  *MOV*    AL, 20H  *OUT*    20H, AL               *;中断结束命令*  *IRET*  CODE ENDS  END START  （3）运行程序，按动 KK1＋产生单次脉冲，观察实验现象。  【扩展实验】  **扩展实验1：在显示屏上以1s的固定间隔持续输出特定的字符**  连线如下：  C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps46.jpg  代码如下：  A8254   EQU 06C0H  C8254   EQU 06C4H  CON8254 EQU 06C6H  SSTACK SEGMENT STACK             DW 32 DUP(?)  SSTACK ENDS  CODE SEGMENT  *ASSUME* CS:CODE      START:  *PUSH*   DS  *MOV*    AX, 0000H  *MOV*    DS, AX  *MOV*    AX, 0FFSET MIR7  *MOV*    SI, 003CH  *MOV*    [SI], AX  *MOV*    AX, CS  *MOV*    SI, 003EH  *MOV*    [SI], AX  *CLI*  *POP*    DS  *MOV*    AL, 11H  *OUT*    20H, AL  *MOV*    AL, 08H  *OUT*    21H, AL  *MOV*    AL, 04H  *OUT*    21H, AL  *MOV*    AL, 01H  *OUT*    21H, AL  *MOV*    AL, 6FH  *OUT*    21H, AL  *MOV*    DX, CON8254  *MOV*    AL, 36H  *OUT*    DX, AL  *MOV*    DX, A8254  *MOV*    AL, 00H  *OUT*    DX, AL  *MOV*    AL, 48H  *OUT*    DX, AL  *STI*      AA1:  *JMP*    AA1      MIR7: *STI*  *MOV*    AX, 014DH  *INT*    10H  *MOV*    AX, 0120H  *INT*    10H  *MOV*    AL, 20H  *OUT*    20H, AL  *IRET*  CODE ENDS           END START  代码思路：   1. **定义寄存器地址**：A8254、C8254 和 CON8254 分别定义了计时器的通道 A 和通道 C 的数据端口以及控制寄存器的端口地址。 2. **定义堆栈段**：SSTACK SEGMENT STACK 定义了一个堆栈段，用于存储程序运行时的局部变量和返回地址等。 3. **定义代码段**：CODE SEGMENT 开始定义代码段，并使用 ASSUME CS:CODE 确保代码段被正确识别。 4. **程序入口**：START: 是程序的入口点。 5. **设置数据段寄存器**：PUSH DS 将数据段寄存器压栈，MOV AX, 0000H 将 AX 寄存器清零，MOV DS, AX 将数据段寄存器设置为0，这通常用于访问 BIOS 数据区。 6. **设置中断向量**：MOV AX, 0FFSET MIR7 准备设置中断向量，MOV SI, 003CH 和 MOV [SI], AX 将中断向量表的第 0x3C 位置的中断向量设置为 MIR7 标签的地址。MOV AX, CS 将代码段地址加载到 AX，MOV SI, 003EH 和 MOV [SI], AX 设置中断向量的段地址。 7. **禁用中断**：CLI 指令禁用中断。 8. **设置中断控制器**：通过一系列的 OUT 指令设置 8253/8254 计时器和中断控制器的模式，以准备接受定时中断。 9. **设置计时器控制字**：MOV DX, CON8254 将控制寄存器地址加载到 DX，MOV AL, 36H 设置控制字，OUT DX, AL 发送控制字到控制寄存器。 10. **设置计时器初始值**：MOV DX, A8254 将通道 A 的数据端口地址加载到 DX，MOV AL, 00H 和 OUT DX, AL 设置计数器的低字节，MOV AL, 48H 和 OUT DX, AL 设置计数器的高字节。这里 48H 是定时器的初始值，用于设置定时周期。 11. **启用中断**：STI 指令启用中断。 12. **无限循环**：AA1: JMP AA1 创建了一个无限循环，程序会一直停留在这里。 13. **中断服务例程**：MIR7 是中断服务例程的开始，STI 指令确保中断可以被处理，MOV AX, 014DH 和 INT 10H 调用 BIOS 视频服务来输出字符 'M'，MOV AX, 0120H 和 INT 10H 调用 BIOS 视频服务来设置光标位置，MOV AL, 20H 和 OUT 20H, AL 发送 EOI (End of Interrupt) 信号给中断控制器，IRET 返回中断。 14. **程序结束**：CODE ENDS 结束代码段的定义，END START 指定程序的入口点。   **扩展实验2：利用8255和8259a实现开关控制流水灯**  连线如下：    代码如下：  SSTACK  SEGMENT STACK  DW 32 DUP(32)  SSTACK ENDS  CODE SEGMENT  *ASSUME* CS:CODE  START: *PUSH* DS  *MOV* DX,0646H  *MOV* AL,80H  *OUT* DX,AL  *MOV* BX,0001H  *MOV* DX,1  *MOV* AX,0000H  *MOV* DS,AX  *MOV* AX,OFFSET MIR7  *MOV* SI,003CH  *MOV* [SI],AX  *MOV* AX,CS  *MOV* SI,003EH  *MOV* [SI],AX  *MOV* AX,OFFSET SIR1  *MOV* SI,00C4H  *MOV* [SI],AX  *MOV* AX,CS  *MOV* SI,00C6H  *MOV* [SI],AX  *CLI*  *POP* DS  *MOV* AL,11H  *OUT* 20H,AL  *MOV* AL,08H  *OUT* 21H,AL  *MOV* AL,04H  *OUT* 21H,AL  *MOV* AL,01H  *OUT* 21H,AL  *MOV* AL,11H  *OUT* 0A0H,AL  *MOV* AL,30H  *OUT* 0A1H,AL  *MOV* AL,02H  *OUT* 0A1H,AL  *MOV* AL,01H  *OUT* 0A1H,AL  *MOV* AL,0FDH  *OUT* 0A1H,AL  *MOV* AL,6BH  *OUT* 21H,AL  *STI*  AA2: *CMP* DX,0  *NOP*  *JE* AA2  *CLI*  *PUSH* DX  *MOV* DX,0640H  *MOV* AL,BL  *OUT* DX,AL  *MOV* DX,0642H  *MOV* AL,BH  *OUT* DX,AL  *CALL* DELAY  *CALL* DELAY  *ROL* BX,1  *POP* DX  *STI*  *LOOP* AA2  MIR7: *CALL* DELAY  *MOV* DX,0  *MOV* AL,20H  *OUT* 20H,AL  *IRET*  SIR1: *CALL* DELAY  *MOV* DX,1  *MOV* AL,20H  *OUT* 0A0H,AL  *OUT* 20H,AL  *IRET*  代码思路：   1. **定义堆栈段**：SSTACK SEGMENT STACK 定义了一个堆栈段，用于存储程序运行时的局部变量和返回地址等，堆栈大小为32个字（DW 32 DUP(32)）。 2. **定义代码段**：CODE SEGMENT 开始定义代码段，并使用 ASSUME CS:CODE 确保代码段被正确识别。 3. **程序入口**：START: 是程序的入口点。 4. **初始化**：    * PUSH DS 将数据段寄存器压栈。    * MOV DX,0646H 设置端口地址，用于控制8255的控制寄存器。    * MOV AL,80H 准备8255的控制字，设置为模式0，即基本输入/输出模式。    * OUT DX,AL 将控制字输出到8255的控制寄存器。 5. **设置流水灯初始状态**：    * MOV BX,0001H 设置BX寄存器为1，这将作为流水灯的初始状态。    * MOV DX,1 将DX寄存器设置为1，可能用于后续的输出操作。 6. **设置中断向量**：    * MOV AX, OFFSET MIR7 和 MOV SI, 003CH 设置中断向量表的第 0x3C 位置的中断向量，用于8259A的IRQ0中断。    * MOV AX, CS 和 MOV SI, 003EH 设置中断向量的段地址。 7. **设置8255的中断向量**：    * MOV AX, OFFSET SIR1 和 MOV SI, 00C4H 设置8255的中断向量。 8. **禁用中断**：CLI 指令禁用中断。 9. **设置中断控制器**：通过一系列的 OUT 指令设置8259A的模式和屏蔽位。 10. **启用中断**：STI 指令启用中断。 11. **流水灯主循环** AA2:：     * CMP DX, 0 检查DX寄存器是否为0，如果是，则跳转到标签 AA2。     * NOP 空操作，用于延迟。     * JE AA2 如果DX为0，则跳转到AA2，形成循环。     * CLI 禁用中断，准备进入中断服务例程。     * PUSH DX 将DX寄存器的值压栈，以备后续使用。     * MOV DX,0640H 设置端口地址，用于8255的端口A。     * MOV AL,BL 将BX寄存器的低8位（当前流水灯状态）输出到8255的端口A。     * MOV DX,0642H 设置端口地址，用于8255的端口B。     * MOV AL,BH 将BX寄存器的高8位输出到8255的端口B。     * CALL DELAY 调用延时子程序。     * ROL BX, 1 将BX寄存器的值循环左移一位，实现流水灯效果。     * POP DX 恢复DX寄存器的值。     * STI 启用中断。     * LOOP AA2 循环直到DX寄存器减到0。 12. **中断服务例程** MIR7 和 SIR1：     * MIR7 是8259A的IRQ0中断服务例程，调用延时子程序，发送EOI（End of Interrupt）信号给8259A，然后通过 IRET 返回。     * SIR1 是8255的中断服务例程，同样调用延时子程序，发送EOI信号给8259A，然后通过 OUT 0A0H, AL 向8259A发送中断结束信号，最后通过 OUT 20H, AL 向CPU发送中断结束信号，并通过 IRET 返回。 13. **延时子程序** DELAY：这个子程序通过循环来实现延时，以便控制流水灯的速度。 14. **程序结束**：CODE ENDS 结束代码段的定义，END START 指定程序的入口点。   **扩展实验3：按动KK1+控制扬声器发声的开和关**  连线如图：    代码如下：  IOY3          EQU  06C0H  MY8254\_COUNT0 EQU  IOY3+00H   *; 8254 计数器 0 端口地址*  MY8254\_COUNT1 EQU  IOY3+02H   *; 8254 计数器 1 端口地址*  MY8254\_COUNT2 EQU  IOY3+04H   *; 8254 计数器 2 端口地址*  MY8254\_MODE   EQU  IOY3+06H   *; 8254 控制寄存器端口地址*  SSTACK SEGMENT STACK             DW 32 DUP(?)  SSTACK 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      START:  *; 保存 DS 寄存器*  *PUSH*   DS  *MOV*    AX, 0000H  *MOV*    DS, AX  *; 设置 IRQ7 中断向量*  *MOV*    AX, OFFSET IRQ7\_HANDLER    *; 取中断入口地址*  *MOV*    SI, 003CH                  *; 中断向量地址*  *MOV*    [SI], AX                   *; 填 IRQ7 的偏移地址*  *MOV*    AX, CS  *MOV*    SI, 003EH  *MOV*    [SI], AX                   *; 填 IRQ7 的段地址*  *CLI*                               *; 禁用中断*  *POP*    DS                         *; 恢复 DS*  *; 初始化主片 8259*  *MOV*    SI, 0  *MOV*    AL, 11H  *OUT*    20H, AL                    *; ICW1*  *MOV*    AL, 08H  *OUT*    21H, AL                    *; ICW2*  *MOV*    AL, 04H  *OUT*    21H, AL                    *; ICW3*  *MOV*    AL, 01H  *OUT*    21H, AL                    *; ICW4*  *MOV*    AL, 6FH  *OUT*    21H, AL                    *; OCW1*  *MOV*    BP, 0  *STI*                               *; 启用中断*  *; 设置数据段*  *MOV*    AX, DATA  *MOV*    DS, AX  *; 初始化 8254 工作方式*  *MOV*    DX, MY8254\_MODE  *MOV*    AL, 36H                    *; 定时器 0，方式 3*  *OUT*    DX, AL      BEGIN:  *; 装入频率表和时间表起始地址*  *MOV*    SI, OFFSET FREQ\_LIST  *MOV*    DI, OFFSET TIME\_LIST      PLAY:  *; 设置初始时钟频率 1MHz*  *MOV*    DX, 0FH  *MOV*    AX, 4240H  *DIV*    WORD PTR [SI]              *; 计算计数初值，0F4240H / 输出频率*  *MOV*    DX, MY8254\_COUNT0  *OUT*    DX, AL                     *; 装入计数初值低字节*  *MOV*    AL, AH  *OUT*    DX, AL                     *; 装入计数初值高字节*  *; 调用延时子程序，演奏相对时间*  *MOV*    DL, [DI]  *CALL*   DELAY  *; 更新指针*  *ADD*    SI, 2  *INC*    DI  *; 判断是否到达曲末*  *CMP*    WORD PTR [SI], 0  *JE*     BEGIN  *; 循环等待*      WAIT\_LOOP:  *CMP*    BP, 0FFFFH  *JZ*     WAIT\_LOOP  *JMP*    PLAY  *; 延时子程序*      DELAY:  *PUSH*   CX  *MOV*    CX, 0F000H      DELAY\_LOOP:  *PUSH*   AX  *POP*    AX  *LOOP*   DELAY\_LOOP  *POP*    CX  *RET*  *; 额外延时子程序*  DALLY PROC  *MOV*    CX, 0010H      D1:          *MOV*    AX, 0FF0H      D2:          *DEC*    AX  *JNZ*    D2  *LOOP*   D1  *DEC*    DL  *JNZ*    D0  *RET*  DALLY ENDP  *; IRQ7 中断处理程序*      IRQ7\_HANDLER:  *STI*  *NOT*    BP  *MOV*    AL, 20H  *OUT*    20H, AL                    *; 中断结束命令*  *IRET*  CODE ENDS  END  START  代码思路：   1. **定义常量**：IOY3 是8254定时器的基地址，MY8254\_COUNT0、MY8254\_COUNT1、MY8254\_COUNT2 和 MY8254\_MODE 分别是8254定时器各个计数器和控制寄存器的端口地址。 2. **定义堆栈段**：SSTACK SEGMENT STACK 定义了一个堆栈段，用于存储程序运行时的局部变量和返回地址等。 3. **定义数据段**：DATA SEGMENT 中定义了两个数组，FREQ\_LIST 是频率表，包含了音乐中各个音符对应的频率值；TIME\_LIST 是时间表，包含了各个音符的持续时间。 4. **定义代码段**：CODE SEGMENT 开始定义代码段，并使用 ASSUME CS:CODE 确保代码段被正确识别。 5. **程序入口**：START: 是程序的入口点。 6. **保存和设置 DS 寄存器**：PUSH DS 保存 DS 寄存器，MOV AX, 0000H 和 MOV DS, AX 将 DS 设置为0，用于访问 BIOS 数据区。 7. **设置 IRQ7 中断向量**：设置8254定时器的IRQ7中断处理程序的中断向量。 8. **初始化8259A**：通过发送初始化命令字（ICW）初始化8259A中断控制器。 9. **设置8254工作方式**：设置8254定时器0为方式3（方波发生器模式）。 10. **初始化数据段**：设置数据段寄存器 DS 指向 DATA 段。 11. **播放音乐**：     * PLAY: 标签开始播放音乐的循环。     * 加载频率表和时间表的起始地址到 SI 和 DI 寄存器。     * 计算定时器的初始计数值，设置定时器0的频率，以产生相应的音符频率。     * 调用 DELAY 子程序实现音符的持续时间。     * 更新频率表和时间表的指针。     * 判断是否到达音乐的末尾，如果没有到达则继续播放。 12. **延时子程序** DELAY：通过循环实现延时，控制音符的持续时间。 13. **额外延时子程序** DALLY：提供了另一个延时子程序，可能用于更精细的时间控制。 14. **IRQ7 中断处理程序** IRQ7\_HANDLER：处理定时器0的中断，中断服务例程中通过 NOT BP 可能用于切换扬声器的输出状态，发送EOI（End of Interrupt）信号，并返回。 15. **程序结束**：CODE ENDS 结束代码段的定义，END START 指定程序的入口点。 | | |
| 数据记录  和计算 | 【基础实验】  **基础实验1：**    **基础实验2：**    **基础实验3：**    **【扩展实验】**  **扩展实验1：**    **扩展实验2：**    **扩展实验3：**    (音频无法展示) | | |
| 结 论  （结 果） | 成功实现三个基础实验，即单中断实验、级联实验和计数产生中断实验，同时实现了三个扩展实验（在显示屏上以1s的固定时间间隔持续输出字符“M”、 利用8255和8259a实现开关控制流水灯以及在扬声器发声实验基础上，按动KK1+控制扬声器发声的开和关 | | |
| 小 结 | 通过这次实验，我对8259中断控制器的工作机制、编程技巧以及级联技术有了深刻的理解和实践应用。通过亲自动手操作，我将理论知识转化为实际操作，这对于加深对知识点的理解和掌握至关重要。  在进行扩展实验3时，我最初认为控制电子扬声器发声与之前做过的电子发声实验原理相似，因此初步尝试时直接采用了之前的代码进行修改。然而，在实际控制过程中，我遇到了问题：按下KK1+并没有实现预期的声音中断，而是产生了不同的音调。这个问题让我意识到可能需要重新审视和调整代码。  在同学的支持和讨论帮助下，我决定对代码进行彻底的修改。通过团队合作和集思广益，我们最终克服了难题，成功实现了KK1+控制声音的开和关的功能。这次经历不仅锻炼了我的编程能力，也让我体会到了团队协作的力量。 | | |
| 指导老师评 议 | 成绩评定： 指导教师签名： | | |