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 | 8254定时计数实验 | 实验时间 | 2024.5.27 |
| 实验目的 | 1. 掌握 8254 的工作方式及应用编程。  2. 掌握 8254 典型应用电路的接法。 | | |
| 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.定时应用实验。编写程序，应用8254的定时功能，产生一个1s的方波。（可参考实验指导书4.4实验2 ）  2.电子发声设计实验。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发出相应的乐曲。（可参考实验指导书4.10）  【扩展实验要求】  1. 应用8254的定时功能，只用一个通道实现输出1Hz的方波。  2.在“扩展实验1”的基础上加入8255，实现当按动单次脉冲KK1+后改变示波器输出的方波的波形频率（大致为按动一次减少一倍，即第一次按动后为半秒一个方波）。  【实验原理】  8254 是 Intel 公司生产的可编程间隔定时器。是 8253 的改进型，比 8253 具有更优良的  性能。8254 具有以下基本功能：  （1）有 3 个独立的 16 位计数器。  （2）每个计数器可按二进制或十进制（BCD）计数。  （3）每个计数器可编程工作于 6 种不同工作方式。  （4）8254 每个计数器允许的最高计数频率为 10MHz（8253 为 2MHz）。  （5）8254 有读回命令（8253 没有），除了可以读出当前计数单元的内容外，还可以读  出状态寄存器的内容。  （6）计数脉冲可以是有规律的时钟信号，也可以是随机信号。计数初值公式为： n=fCLKi÷fOUTi、其中 fCLKi是输入时钟脉冲的频率，fOUTi 是输出波形的频率。 图 4.27 是 8254 的内部结构框图和引脚图，它是由与 CPU 的接口、内部控制电路和三个  计数器组成。8254 的工作方式如下述：   1. 方式 0：计数到 0 结束输出正跃变信号方式。 2. 方式 1：硬件可重触发单稳方式。 3. 方式 2：频率发生器方式。 4. 方式 3：方波发生器。 5. 方式 4：软件触发选通方式。 6. 方式 5：硬件触发选通方式   C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps33.jpg  【实验步骤】  编写程序，将 8254 的计数器 0 和计数器 1 都设臵为方式 3，用信号源 1MHz 作为 CLK0  时钟，OUT0 为波形输出 1ms 方波，再通过 CLK1 输入，OUT1 输出 1s 方波。  实验步骤：  **定时应用实验**  （1） 接线图如下图所示。  C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps34.jpg  （2）根据实验内容，编写实验程序，经编译、链接无误后装入系统，代码如下：  A8254    EQU  0600H  B8254    EQU  0602H  C8254    EQU  0604H  CON8254  EQU  0606H  SSTACK  SEGMENT STACK          DW 32 DUP(?)  SSTACK  ENDS  CODE    SEGMENT  *ASSUME* CS:CODE, SS:SSTACK  START:  *MOV* DX, CON8254         *;8254*  *MOV* AL, 36H             *;计数器0，方式3*  *OUT* DX, AL    *MOV* DX, A8254  *MOV* AL, 0E8H            *;03E8H=1000*  *OUT* DX, AL  *MOV* AL, 03H  *OUT* DX, AL  *MOV* DX, CON8254         *;8254*  *MOV* AL, 76H             *;计数器1，方式3*  *OUT* DX, AL    *MOV* DX, B8254  *MOV* AL, 0E8H  *OUT* DX, AL  *MOV* AL, 03H  *OUT* DX, AL  AA1:    *NOP*  *JMP* AA1    CODE    ENDS          END  START    （3）单击运行按钮，运行实验程序，8254 的 OUT1 会输出 1s 的方波。  （4）用示波器观察波形的方法：单击虚拟仪器菜单中的示波器按钮或直接单击工具栏的按钮，在新弹出的示波器界面上单击形如闪电的按钮运行示波器，就可以观测出 OUT1 输出的波形。  **电子发声设计实验**  1. 如下图所示连接实验线路。  C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps35.jpg  2. 编写实验程序，经编译、连接无误后装入系统，代码如下：  *;=================================================================*  *; SOUND.asm*  *; 电子发声设计实验*  *;=================================================================*  *; 端口定义*  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*    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  *MOV* DX,MY8254\_MODE          *;初始化8254工作方式*  *MOV* AL,36H                  *;定时器0、方式3*  *OUT* DX,AL  BEGIN:  *MOV* SI,OFFSET FREQ\_LIST     *;装入频率表起始地址*  *MOV* DI,OFFSET TIME\_LIST     *;装入时间表起始地址*  PLAY:   *MOV* DX,0FH                  *;输入时钟为1MHz，1M = 0F4240H*  *MOV* AX,4240H  *DIV* WORD PTR [SI]           *;取出频率值计算计数初值，0F4240H / 输出频率*  *MOV* DX,MY8254\_COUNT0  *OUT* DX,AL                   *;装入计数初值*  *MOV* AL,AH  *OUT* DX,AL  *MOV* DL,[DI]                 *;取出演奏相对时间，调用延时子程序*  *CALL* DALLY  *ADD* SI,2  *INC* DI  *CMP* WORD PTR [SI],0         *;判断是否到曲末？*  *JE*  BEGIN  *JMP*  PLAY  DALLY   PROC                        *;延时子程序*  D0:     *MOV* CX,0010H  D1:     *MOV* AX,0FF0H  D2:     *DEC* AX  *JNZ* D2  *LOOP* D1  *DEC* DL  *JNZ* D0  *RET*  DALLY   ENDP  CODE    ENDS          END START  3. 运行程序，听扬声器发出的音乐是否正确。  【扩展实验】  **只用1个通道实现1Hz方波**  实验连线如图    代码如下：  A8254    EQU  0600H  B8254    EQU  0602H  C8254    EQU  0604H  CON8254  EQU  0606H  SSTACK  SEGMENT STACK          DW 32 DUP(?)  SSTACK  ENDS  CODE    SEGMENT  *ASSUME* CS:CODE, SS:SSTACK  START:  *MOV* DX, CON8254         *;8254*  *MOV* AL, 36H             *;计数器0，方式3*  *OUT* DX, AL    *MOV* DX, A8254  *MOV* AL, 020H            *;03E8H=1000*  *OUT* DX, AL  *MOV* AL, 04EH  *OUT* DX, AL  AA1:    *NOP*  *JMP* AA1    CODE    ENDS          END  START  代码思路：   1. **定义寄存器地址**：首先定义了 8254 计时器的三个通道和控制寄存器的端口地址。A8254、B8254、C8254 分别对应通道 A、B、C 的数据端口，CON8254 是控制寄存器的端口。 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，ASSUME CS:CODE, SS:SSTACK 指令确保代码段和堆栈段被正确识别。 4. **程序入口**：START: 是程序的入口点。 5. **设置控制字**：MOV DX, CON8254 将控制寄存器的地址加载到 DX 寄存器。MOV AL, 36H 准备控制字，其中 36H 表示计数器 0 工作在方式 3（方波模式）和二进制计数模式。然后 OUT DX, AL 将控制字输出到控制寄存器。 6. **设置计数器 0**：MOV DX, A8254 将通道 A 的数据端口地址加载到 DX 寄存器。MOV AL, 020H 和 MOV AL, 04EH 准备计数器 0 的初始值，这里使用的是两个字节的加载方式，即先加载低字节 020H，然后加载高字节 04EH。这两个值组合起来是 03E8H，这是计数器的初始值，用于设置计时器的频率。 7. **生成方波**：计数器 0 被设置为从 03E8H 开始计数，每次计数到 0 后会重载 03E8H 并翻转输出状态，从而生成 1Hz 的方波信号。 8. **无限循环**：AA1: NOP 和 JMP AA1 创建了一个无限循环，程序会一直停留在这里，计数器 0 会继续工作，生成方波信号。 9. **程序结束**：CODE ENDS 结束代码段的定义，END START 指定程序的入口点。   **随着开关按动减小频率的方波**  实验连线如图：  代码如下：  A8254    EQU  0600H  B8254    EQU  0602H  C8254    EQU  0604H  CON8254  EQU  0606H  SSTACK  SEGMENT STACK          DW 32 DUP(?)  SSTACK  ENDS  CODE    SEGMENT  *ASSUME* CS:CODE, SS:SSTACK  START:  *MOV* BX,4E20H  AA0:      *MOV* DX, CON8254         *;8254*  *MOV* AL, 36H             *;计数器0，方式3*  *OUT* DX, AL    *MOV* DX, A8254  *MOV* AL, BL  *OUT* DX, AL  *MOV* AL, BH  *OUT* DX, AL  AA1:    *NOP*  *MOV* DX, 0640H  *IN* AL, DX    *CALL* DELAY  *CALL* DELAY  *CALL* DELAY  *CALL* DELAY  *CALL* DELAY  *CALL* DELAY    *AND* AL,01H  *CMP* AL,01H  *JZ*  AA3  *JMP* AA1    AA3:  *ROR* BX,01H  *JMP* AA0    DELAY:  *PUSH* CX  *MOV* CX, 0F000H  AA2:    *PUSH* AX  *POP*  AX  *LOOP* AA2  *POP*  CX  *RET*    CODE    ENDS          END  START  代码思路：   1. **定义寄存器地址**：A8254、B8254、C8254 和 CON8254 分别定义了计时器的三个通道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. **定义代码段**：CODE SEGMENT 开始定义代码段，ASSUME CS:CODE, SS:SSTACK 指令确保代码段和堆栈段被正确识别。 4. **程序入口**：START: 是程序的入口点。 5. **初始化计数器值**：MOV BX,4E20H 将计数器 0 的初始值 4E20H 载入 BX 寄存器。这个值决定了计时器的频率和输出方波的周期。 6. **设置控制字**：MOV DX, CON8254 将控制寄存器的地址加载到 DX 寄存器。MOV AL, 36H 设置控制字，其中 36H 表示计数器 0 工作在方式 3（方波模式）和二进制计数模式。然后 OUT DX, AL 将控制字输出到控制寄存器。 7. **设置计数器 0**：MOV DX, A8254 将通道 A 的数据端口地址加载到 DX 寄存器。接着，MOV AL, BL 和 MOV AL, BH 将 BX 寄存器的低字节和高字节分别输出到计数器 0，设置其初始值。 8. **主循环**：AA1: 标签开始了一个主循环，在这个循环中，程序会不断检测某个端口（MOV DX, 0640H 和 IN AL, DX）的状态，并根据状态执行不同的操作。 9. **软件延迟**：CALL DELAY 调用了一个延迟子程序，该子程序通过循环消耗 CPU 时间来实现延迟。 10. **状态检测与翻转**：AND AL,01H 将 AL 寄存器的值与 01H 进行 AND 操作，这可能是为了检测某个位的状态。CMP AL,01H 和 JZ AA3 检查该位是否为 1，如果是，则跳转到 AA3。 11. **状态翻转**：ROR BX,01H 将 BX 寄存器的值右旋转一位，这可以模拟输出状态的变化。 12. **延迟子程序**：DELAY: 定义了一个延迟子程序，通过 PUSH CX 和 POP AX 以及 LOOP AA2 实现循环计数，从而实现延迟。 13. **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】  **定时应用实验**    **电子发声设计实验**  C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps40.png  【扩展实验】  **扩展实验1**    **扩展实验2** | | |
| 结 论  （结 果） | 成功实现基础实验要求（即定时应用实验和电子发声设计实验）和两个扩展实验 | | |
| 小 结 | 通过本次实验的动手，帮助我将课本上的理论变为实际，可以更好的理解理论知识，让知识不再那么抽象，通过本次实验，我掌握了8254的工作方式和应用编程，同时也掌握了8254典型应用电路的接法。 | | |
| 指导老师评 议 | 成绩评定： 指导教师签名： | | |