|  |
| --- |
| 实验名称8253定时/电子发声应用设计实验（基于Proteus工具）  6-2 定时应用程序   1. 实验内容、目的与要求   说明：这里将实验六分成三个小实验完成，分别是6-1计数应用程序（原4-6 8254定时/计数器应用实验——实验1），6-2 定时应用程序（原4-6 8254定时/计数器应用实验——实验2）， 6-3 电子发声设计实验（原4-12 电子发生设计实验），其中对8253的介绍放在实验6-1中  实验内容：  定时应用实验，编写程序，应用8253（原实验教材中应用的是8254）定时功能，产生一个1Hz的方波   1. （实验要求：参照实验教材基本内容要求，完成基于汇编语言的程序功能调试，掌握基本硬件端口功能设计及应用方法   提高要求：1. 通过开关或按键，实现两首以上乐曲的选择性演奏。适当利用之前的实验模块进行功能拓展  2. 通过开关或按键，输出不同的频率的方波信号   1. 实验硬件与软件环境（标注实验设备名称、设备号）   硬件环境：微型计算机  软件环境：Proteus设计工具   1. 实验原理与步骤   定时应用实验的要求是，使用8253（参考书使用8254）的计数器2（CNT2）使其工作在方式3下，用信号源1.8432MHz作为CLK2时钟，计数初值设为100，即对CLK2进行100分频。理论上讲，OUT2输出频率应为18.432KHz。再将OUT2接到计数器0（CNT0）的CLK0，设置计数器0同样工作在方式3，计数初值为18432，将在OUT0得到1Hz的输出。  在实践中，我们常常利用8253的几个计数器的独立性，使其一同工作完成更大的分频，因为8253计数寄存器只有16位，也就是说最高只能完成65536分频。对于更高的分频，就像本实验一样，要将1.8432MHz分频到1Hz，就要采用这样的办法。    Figure 1 实验6-2原理示意图  在实验过程中，通过示波器工具，我们可以查看输出端OUT0和OUT2的输出时钟频率，以验证理论结果。  下面看一下汇编程序的编写，整体来说这个程序较为简单，基本流程如下    Figure 2 实验6-2程序流程图  CNT0和CNT2均工作在方式3，CNT0的计数初值为18432（4800H），需要两个字节；CNT2的计数初值为100（64H），需要一个字节，所以CNT0和CNT2的方式控制字分别为0011 0110B（36H）和1001 0110B（93H）  由此给出实验代码为  CODE SEGMENT PUBLIC 'CODE'  ASSUME CS:CODE, DS:DATAS, SS:STACKS  MCU8253CNT0 EQU IOY2+00H ; CNT0  MCU8253CNT1 EQU IOY2+02H ; CNT1  MCU8253CNT2 EQU IOY2+04H ; CNT2  MCU8253MODE EQU IOY2+06H ; Mode Control  START:  ; And for Release version, keep START added.  ; Do not forget 'END START'  MOV AX, DATAS  MOV DS, AX  ; Config CNT0 -- COUNT = 18432  MOV DX, MCU8253MODE  MOV AL, 00110110B  OUT DX, AL  MOV DX, MCU8253CNT0  MOV AX, 18432  OUT DX, AL  MOV AL, AH  OUT DX, AL  ; Config CNT2 -- COUNT = 100  MOV DX, MCU8253MODE  MOV AL, 10010110B  OUT DX, AL  MOV DX, MCU8253CNT2  MOV AL, 100 ; 100 = 64H  OUT DX, AL ; For oberservation, switch:    JMP TEND  TEND:  MOV DX, MCU8255NA  MOV AL, 0FFH  OUT DX, AL  HLT ; Program ends  DELAY PROC NEAR  PUSH CX  MOV CX, 0FFFFH  D1:  DEC CX  JNZ D1    POP CX  RET  DELAY ENDP  CODE ENDS  END START  四、实验数据记录及实验结果  对实验编译完成后，根据报错信息进行修改，然后完成编辑（本次实验的proteus工程文件包含*firmware*（*固件信息*），所以直接在source code（源代码）界面编译（快捷键Ctrl+F7）即可）。在（自建）的proteus工具实验平台上，完成硬连电路搭接（如下图所示，放大查看）    Figure 实验用电路图  然后，设置时钟信号为1.8432MHz，并接在CLK2端；同时将CNT2的输出端OUT2与CNT0的时钟输入端CLK0相连。将CNT0的GATE端和CNT2的GATE端连接到LOGICSTATE的高电平段。均使用开关器件连接。    Figure 加入时钟信号    Figure 将CLK0与OUT2连在一起    Figure 将GATE0与GATE2接高电平（软件启动）  为了方便观察实验效果，还需要将OUT0和OUT2端接到示波器的输入端。以上准备工作完成后，我们点击Proteus的仿真按钮，开始进行仿真。    Figure 7某时刻实验过程  页面自动弹出示波器窗口，初始时channelC的方波形状较密。调整时间格子的分度值，我们发现channelA(也即是8253的CNT0的输出端)周期较大，而channelC（也即是8253的CNT2的输出端）周期较小。下图展示了仿真至0.2s左右时的OUT2端的波形。    Figure 8 OUT2端的波形  根据游标信息（上图仅展示了一个游标），以及时间刻度的内容。可以读出方波信号单周期约为53。计算周期约为18.867kHz。与理论上的频率（对1.8432MHz进行100分频）18.432kHz是相当接近的。  然后我们来考察示波器的channelA即计数器CNT0的输出端的周期，从理论上讲，CNT0与CNT2相独立，OUT0应当是CLK0也即OUT2的18432分频，对应于输出一个周期为1s的方波信号，或者更具体来讲，在0.5s左右应当有一个由高至低的跳变。在实际实验中，由于这里用的的时钟信号过高，所以总是会出现simulation not runing in real time的提示    Figure 9 仿真过程中的提示信息  这就会导致实践中很难仿真至0.5s之后并观察到其后的内容（该线程往往会自动终结），所以不易看到0.5s左右时段的OUT0端的由高至低的跳变。。经过多次实验，可以观察到此次跳变。  此外，还可以我们修改CNT0计数寄存器中的初值为更小的数字如1000等。      Figure 取CNT0的计数初值为1000的某时刻实验结果  上图是将CNT0的计数初值为1000后的结果，在（2ms的来由是通过对暂停处倒退所得）处观察到一处跳变。根据理论计算，此时的单周期应为。由于。这就验证了我们的观点，同时还说明当8253工作在方式3下，方波周期是“先高后低”的  五、实验分析与讨论、收获体会 |
|  |