四 川 大 学 计 算 机 学 院、软 件 学 院

实 验 报 告

|  |  |  |  |
| --- | --- | --- | --- |
| 课程名称 | 微机系统与接口技术 | 实验课时 | 4 |
| 实验项目 | AD和DA转换实验 | 实验时间 | 2024.5.13 |
| 实验目的 | 1. 学习理解“模/数信号转换” 和“数/模信号转换” 的基本原理。  2. 掌握模/数转换芯片 ADC0809 和数/模信号转换芯片DAC0832 的的使用方法。 | | |
| 实验环境 | Wdm86 | | |
| 实验内容（算法、程序、步骤和方法） | 【基本实验要求】  （1）A/D转换实验：编写实验程序，将ADC单元中提供的0V-5V信号源作为ADC0809的模拟输入量，进行A\D转换，转换结果通过变量进行显示。（至少记录三组典型值：+5V——FFH、0V——00H、+2.5V——？？）  （2）D/A转换实验：设计实验电路图实验线路并编写程序，实现D/A转换，要求产生锯齿波、脉冲波（方波），并用示波器观察电压波形。  【扩展实验要求】  针对D/A转换实验编写程序，将锯齿波修改为三角波和梯形波，并通过虚拟示波仪观察并记录。  【基础实验步骤】   1. A/D转换实验 2. 按照下图连接实验线路。   C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps8.jpg   1. 编写实验程序，经编译、链接无误后装入系统。   代码如下：  *;=========================================================*  *; 文件名: AD1.ASM*  *; 功能说明: 进行AD采样，将结果显示。 片选为IOY1*  *;=========================================================*  SSTACK SEGMENT STACK             DW 64 DUP(?)  SSTACK ENDS  PUBLIC  VALUE           *;设置全局变量以便变量监视*  DATA SEGMENT      VALUE DB ?  DATA ENDS  CODE SEGMENT  *ASSUME* CS:CODE,DS:DATA      START:*MOV*    AX, DATA  *MOV*    DS, AX  *MOV*    DX, 640H           *;启动AD采样*  *OUT*    DX, AL  *CALL*   DALLY  *IN*     AL, DX             *;读AD采样结果*  *MOV*    VALUE, AL          *;将结果送变量*    *JMP*    START              *;在此处设置断点*      DALLY:*PUSH*   CX  *PUSH*   AX  *MOV*    CX, 100H      A5:   *MOV*    AX, 0800H      A6:   *DEC*    AX  *JNZ*    A6  *LOOP*   A5  *POP*    AX  *POP*    CX  *RET*    CODE ENDS          END START  3. 将变量 VALUE 添加到变量监视窗口中。  4. 在 JMP START 语句行设臵断点，使用万用表测量 ADJ 端的电压值，计算对应的采样  值，然后运行程序。  5. 程序运行到断点处停止运行，查看变量窗口中 VALUE 的值，与计算的理论值进行比较，  看是否一致（可能稍有误差，相差不大）。  6. 调节电位器，改变输入电压，比较 VALUE 与计算值，反复验证程序功能。  （2） DA转化实验  1. 实验接线图如下图所示，按图接线  C:\Users\刘桂君\AppData\Local\Temp\ksohtml17060\wps9.jpg   1. 编写实验程序，经编译、链接无误后装入系统。 代码如下：   *;=========================================================*  *; 文件名: DA1.ASM*  *; 功能描述: 利用DA0832产生锯齿波，通过延时变化波形*  *;=========================================================*  STACK SEGMENT STACK            DW 32 DUP(?)  STACK ENDS  CODE SEGMENT  *ASSUME* CS:CODE, SS:STACK      START:*MOV*    AX, 00H  *MOV*    DX, 600H  *MOV*    AL, 00H      AA1:  *OUT*    DX, AL  *CALL*   DELAY  *INC*    AL  *JMP*    AA1        DELAY:*PUSH*   CX  *MOV*    CX, 03FFH      AA2:  *PUSH*   AX  *POP*    AX  *LOOP*   AA2  *POP*    CX  *RET*    CODE ENDS         END START  3. 单击运行按钮，运行实验程序，用示波器测量 DA 的输出，观察实验现象。  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示波器按钮，在新弹出的示波器界面上单击按钮运行示波器，观测实验波形。  5. 改写程序，产生方波，并观察，代码如下：  *;=========================================================*  *; 文件名: DA2.ASM*  *; 功能描述: 利用DA0832产生方波，通过延时变化波形*  *;=========================================================*  SSTACK SEGMENT STACK             DW 32 DUP(?)  SSTACK ENDS  CODE SEGMENT  *ASSUME* CS:CODE      START:*MOV*    AX, 00H  *MOV*    DX, 600H      AA1:  *MOV*    AL, 00H  *OUT*    DX, AL  *CALL*   DELAY  *MOV*    AL, 07FH  *OUT*    DX, AL  *CALL*   DELAY  *JMP*    AA1        DELAY:*PUSH*   CX  *MOV*    CX, 0FF00H      AA2:  *PUSH*   AX  *POP*    AX  *LOOP*   AA2  *POP*    CX  *RET*    CODE ENDS          END START    【扩展实验】  产生三角波代码：  *; 产生三角波*  STACK SEGMENT STACK            DW 32 DUP(?)  STACK ENDS  CODE SEGMENT  *ASSUME* CS:CODE, SS:STACK      START:*MOV*    AX, 00H  *MOV*    DX, 600H      L1:   *MOV*    AL, 00H  *MOV*    CX, 0FFH      AA1:  *OUT*    DX, AL  *CALL*   DELAY  *INC*    AL  *LOOP*   AA1  *MOV*    CX, 0FFH      AA2:  *OUT*    DX, AL  *CALL*   DELAY  *DEC*    AL  *LOOP*   AA2  *JMP*    L1      DELAY:*PUSH*   CX  *MOV*    CX, 0FFH      AA3:  *PUSH*   AX  *POP*    AX  *LOOP*   AA3  *POP*    CX  *RET*  CODE ENDS        END START  代码思路：   1. **定义数据段和栈段**：    * STACK SEGMENT 定义了一个32字节大小的栈空间，用于存储临时数据。 2. **初始化寄存器**：    * ASSUME 指令用来初始化代码段和栈段的寄存器。    * START 标签是程序的入口点。    * MOV AX, 00H 和 MOV DX, 600H 分别初始化AX寄存器和DX寄存器，其中DX寄存器用来设置输出端口的地址。 3. **主循环**：    * L1 标签是主循环的开始。    * MOV AL, 00H 将AL寄存器清零，AL寄存器用于存储要输出的值。    * MOV CX, 0FFH 设置CX寄存器为0xFF，用于控制循环的次数。 4. **上升沿**：    * AA1 标签是上升沿的开始。    * OUT DX, AL 将AL寄存器的值输出到DX指定的端口，从而生成信号。    * CALL DELAY 调用延时函数，以控制信号的频率。    * INC AL 将AL寄存器的值加1，实现信号的上升。    * LOOP AA1 循环直到CX寄存器减到0，实现信号的上升。 5. **下降沿**：    * AA2 标签是下降沿的开始，与上升沿类似，但是使用DEC AL 来实现信号的下降。 6. **延时函数**：    * DELAY 标签定义了延时函数。    * 通过嵌套循环来实现延时，PUSH CX 和 POP AX 用于保护和恢复CX寄存器的值。 7. **循环跳转**：    * JMP L1 将程序流跳转回 L1 标签，实现三角波的周期性输出。 8. **程序结束**：    * CODE ENDS 和 END START 标记代码段的结束和程序的入口点。   这段代码通过控制输出端口的值在0到0xFF之间循环变化，生成一个周期性的三角波信号。通过调整延时函数中的循环次数，可以改变三角波的频率。  产生梯形波代码：  *; 产生梯形波*  SSTACK SEGMENT STACK             DW 32 DUP(?)  SSTACK ENDS  CODE SEGMENT  *ASSUME* CS:CODE      START: *MOV*    AX, 00H      *;*  *MOV*    CX, 0FH      AA1:  *MOV*    DX, 600H  *MOV*    AL, 00H  *OUT*    DX, AL  *CALL*   DELAY2        AA2:  *MOV*    DX, 600H  *OUT*    DX, AL  *CALL*   DELAY1  *CMP*    AL, 0FFH  *JE*     AA3  *INC*    AL  *JMP*    AA2        AA3:   *MOV*    DX, 600H  *MOV*    AL, 0FFH  *OUT*    DX, AL  *CALL*   DELAY2        AA4:  *MOV*    DX, 600H  *OUT*    DX, AL  *CALL*   DELAY1  *CMP*    AL, 00H  *JE*     AA1  *DEC*    AL  *JMP*    AA4      DELAY1:  *PUSH*   CX  *MOV*    CX,00FFH      CC2:  *PUSH*   AX  *POP*    AX  *LOOP*   CC2  *POP*    CX  *RET*        DELAY2:  *PUSH*   CX  *MOV*    CX,0FFFFH      CC1:   *PUSH*   AX  *POP*    AX  *LOOP*   CC1  *POP*    CX  *RET*  CODE ENDS          END START  代码思路：   1. **定义栈段**：    * SSTACK SEGMENT 定义了一个32字节大小的栈空间，用于存储临时数据。 2. **初始化寄存器**：    * ASSUME 指令用来设置代码段的段寄存器。    * START 标签是程序的入口点。    * MOV AX, 00H 将AX寄存器清零。    * MOV CX, 0FH 初始化CX寄存器，这里CX可能用于控制某些循环的次数。 3. **上升沿**：    * AA1 标签开始上升沿的生成。    * MOV DX, 600H 设置DX寄存器为输出端口的地址。    * MOV AL, 00H 将AL寄存器清零，用于存储要输出的值。    * OUT DX, AL 将AL寄存器的值输出到DX指定的端口。    * CALL DELAY2 调用延时函数。 4. **线性上升**：    * AA2 标签开始线性上升的部分。    * MOV DX, 600H 和 OUT DX, AL 重复输出当前AL寄存器的值。    * CALL DELAY1 调用较短的延时函数。    * CMP AL, 0FFH 比较AL寄存器的值是否达到0xFF。    * JE AA3 如果达到，则跳转到下降沿的开始。    * INC AL 如果没有达到，则AL寄存器的值加1。    * JMP AA2 继续线性上升。 5. **下降沿**：    * AA3 标签开始下降沿的生成，与上升沿类似，但是使用 DEC AL 来实现信号的下降。 6. **线性下降**：    * AA4 标签开始线性下降的部分。    * 与线性上升类似，但是AL寄存器的值递减。    * JE AA1 如果AL寄存器的值减到0，则跳转回上升沿的开始。 7. **延时函数**：    * DELAY1 和 DELAY2 定义了两个不同的延时函数，通过嵌套循环来实现延时。    * PUSH CX 和 POP AX 用于保护和恢复CX寄存器的值。    * MOV CX, 00FFH 和 MOV CX, 0FFFFH 分别设置了两种不同的延时长度。 8. **程序结束**：    * CODE ENDS 和 END START 标记代码段的结束和程序的入口点。   这段代码通过控制输出端口的值在0到0xFF之间线性变化，生成一个周期性的梯形波信号。通过调整延时函数中的循环次数，可以改变梯形波的频率以及上升和下降沿的持续时间。 | | |
| 数据记录  和计算 | 【AD转化实验】  **实验设备有问题，最大峰值为03！**  5V – 03：    0V – 00：    2.5V – 02：    【DA转化实验】  锯齿波：    方波：    【扩展实验】  三角波：    梯形波： | | |
| 结 论  （结 果） | 成功实现了A/D转换和D/A转换实验，产生了锯齿波和方波，同时用代码实现了扩展实验的三角波和梯形波。 | | |
| 小 结 | 因为实验器材有问题，当电压最大时，值也只有3，我们找了很多遍自己的问题，也没发现问题所在，在请教了老师后，成功解决了问题。  在本次实验中，我通过实践操作，对三角波、锯齿波和脉冲波的生成有了更深入的了解。我发现，三角波可以通过对锯齿波和脉冲波进行一些简单的调整来生成。同样，梯形波的生成也是基于脉冲波，并结合了三角波的特性进行调整。  通过这次动手实验，我对模拟信号与数字信号之间的转换有了更加深刻的认识。我不仅理解了A/D（模拟/数字）和D/A（数字/模拟）转换的基本原理，还学会了如何使用A/D转换芯片ADC0809和D/A转换芯片DAC0832。这让我对这些芯片的功能和操作方法有了实际的掌握和体验。 | | |
| 指导老师评 议 | 成绩评定： 指导教师签名： | | |