**上海大学微机实践报告**

**实验四：D/A转换器实验**

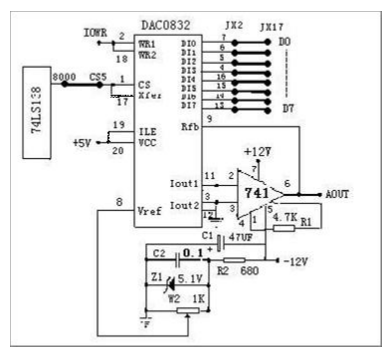
**【实验目的】**

了解D/A转换的基本原理，掌握DAC0832芯片的性能、使用方法及对应的硬件电路。

**【实验内容】**

编写程序，要求输出锯齿波、三角波及方波，分别用示波器观察波形；如有能力，把三段程序编在一起，循环输出三种波形。

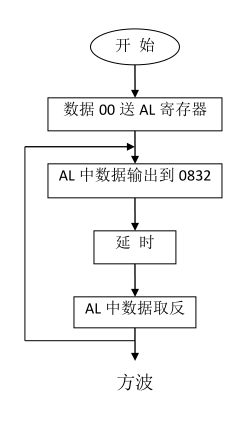
**【实验区域电路连接图】**

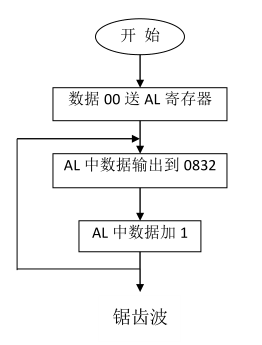


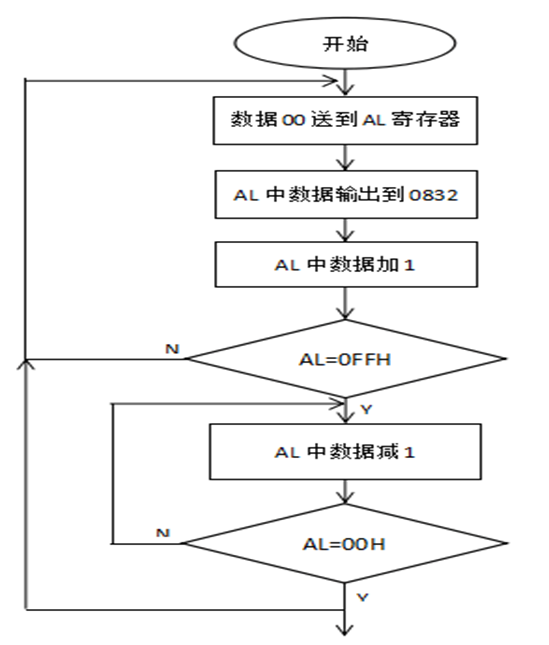
连线：

CS5→8000H；IOWR→IOWR；JX2→JX17；AOUT→示波器。

**【程序框图】**







三角波

**【编程】**

1）方波

CODE SEGMENT ;定义代码段

ASSUME CS:CODE

ORG 1200H

START:

MOV DX,8000H

MOV AL,00H

LOOP1:

OUT DX,AL

CALL DELAY

XOR AL,0FFH ;高低电平转换

JMP LOOP1

DELAY: ;若干秒延时

MOV BX,0FFH

LOOP2:

DEC BX

JNZ LOOP2

RET

CODE ENDS

END START

2）锯齿波

CODE SEGMENT ;定义代码段

ASSUME CS:CODE

ORG 1200H

START:

MOV DX,8000H

MOV AL,00H

LOOP1: ;逐步累加到5V,到最大值后又从零开始累加

OUT DX,AL

NOP

INC AL

JMP LOOP1

CODE ENDS

END START

LOOP2:

OUT DX,AL

MOV AL,00H ;达到最大值后，再从00H不断累加，使波形上升

JMP LOOP1 ;循环输出锯齿波

CODE ENDS

END START

3）三角波

CODE SEGMENT ;定义代码段

ASSUME CS:CODE

ORG 1200H

START:

MOV DX,8000H

MOV AL,00H

LOOP1:OUT DX,AL

INC AL

CMP AL,0FFH

JZ LOOP2 ;进行三角波下降沿

JMP LOOP1

LOOP2:OUT DX,AL

DEC AL

NOP

JZ LOOP1 ;重新进行三角波上升沿

JMP LOOP2

CODE ENDS

END START

**【问答题】**

**1、DAC产生波形的频率如何调节？频率上限的限制取决于那些因素？**

解：

如果是方波只要调节延时程序就可以实现改变频率。三角波和锯齿波的频率除了取决于程序执行的长度，及延时时间，还取决于CPU频率和转换频率。频率的上限取决于CPU频率和转换频率。

**2、如果要求产生正弦波一般用什么方法实现？具体如何实施的（给个方案）。**

解：

在三角波的程序中由于上升时比较操作用 CMPAL,0FFH。先根据你的采样率（确切说是设计的分辨率，例如16），把一个周期的平分n份，算出正弦值；每隔1/n周期，送算出的了n个数值；n越大正弦波形越明显，但不能超过系统频率。

**【心得体会】**

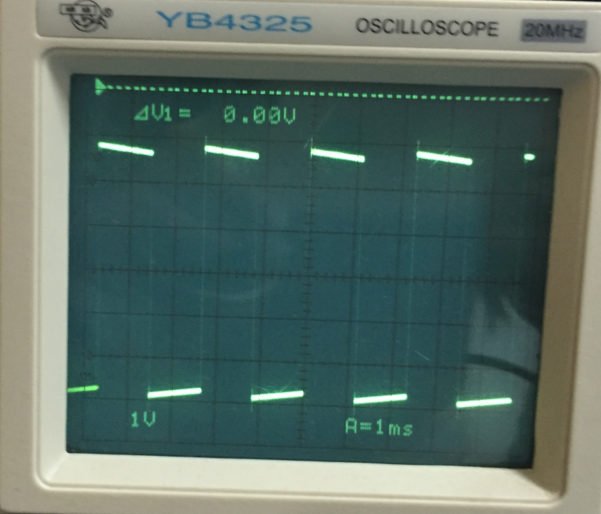
本次试验的程序相对比较简单。主要是输出三个波形：方波、锯齿波、三角波。方波的编译思想主要是现在AL中寄存一个低电平，输出之后取反为高电平，之后跳转回去输出，在取反为低电平，如此循环，中间再加入延时程序锯齿波的思路就是让AL寄存器从0开始累加，每加1就输出。知道最后AL寄存器里面的值就变成了0FFH。这个时候在执行加1就会变成00H。之后在循环这段程序就完成了锯齿波的输出。如果需要改变频率可以加入NOP指令。

三角波的编译思想和锯齿波的相差不多。当AL寄存器里面的值变成0FFH后执行减1操作，然后输出；当减到00时再执行加一操作，如此循环。

JZ LOOP2两步操作，而下降时只用了JZ LOOP1一步操作，因此加入一个NOP以保持上下时间相等。

**【实验波形】**

**1，方波**

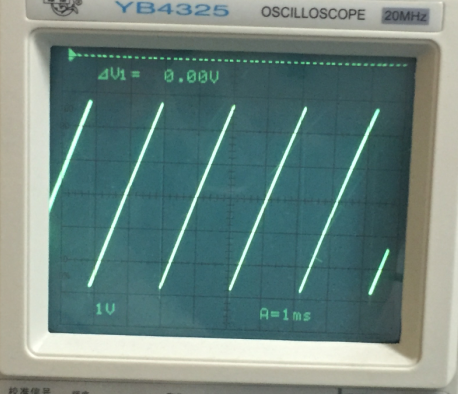


示波器横向每格时间标度为：1ms/格

示波器纵向每格电压幅度为：1V/格

由图：方波纵向有2.6格，则方波的幅度为2.6x1V=2.6V；方波一个周期有2.4格，因此方波周期为1msx2.4=2.4ms。

**2，锯齿波**

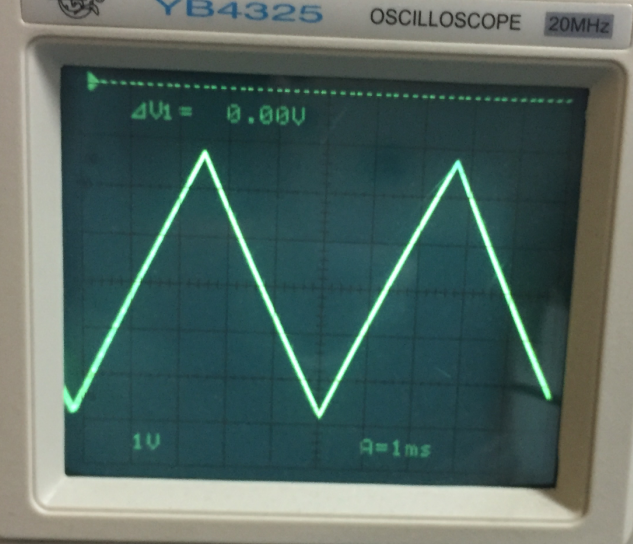


示波器横向每格时间标度为：1ms/格

示波器纵向每格电压幅度为：1V/格

由图:锯齿波纵向有5.6格，锯齿波的幅度为5.6x1V=5.6V；锯齿波一个周期有2格，因此方波周期为1msx2=2ms。

**3,三角波**



示波器横向每格时间标度为：1ms/格

示波器纵向每格电压幅度为：1V/格

由图可以看出三角波纵向有5.6格，因此三角波的幅度为5.6x1V=5.6V；三角波一个周期有5.2格，因此方波周期为1msx5.2=5.2ms。

**实验五：A/D转换器实验**

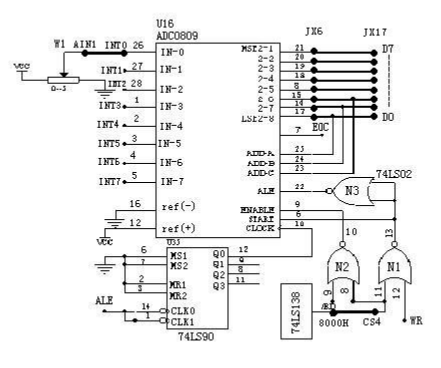
**【实验目的】**

了解模/数转换基本原理，掌握ADC0809的使用方法。掌握A/D转换与计算机的接口方法，了解ADC0809芯片的转换性能及编程，了解计算机如何进行数据采集。

**【实验内容】**

编写程序，用查询方式采样输入模拟电压（模拟量电压从实验装置的电位器接入），并将其转换为二进制数字量，用发光二极管显示。

**【实验区域电路连接图】**

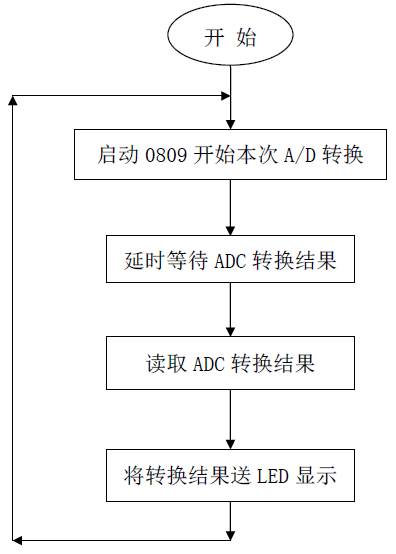


连线：

IN0→AOUT1(可调电压,VIN→+5V)；IOWR→IOWR；IORD→IORD；CLK→500K(单脉冲与时钟单元)；ADDA、ADDB、ADDC→GND；CS4→8000HJX6→JX17（数据总线）

PA0~7→L1~L8

**【程序框图】**



**【编程】**

CODE SEGMENT

ASSUME CS:CODE

ORG 1000H

START:

MOV DX,0FF2BH

MOV AL,80H

OUT DX,AL ;设置8255方式，A口出

L1: MOV DX, 8000H ;ADC0809端口地址

MOV AL,0 ;选择通道0

OUT DX,AL ;发CS和WR信号并送通道地址

MOV AH,0FFH

DELAY: DEC AH

JNZ DELAY ;延时

MOV DX,8000H

IN AL,DX ;读0809转换结果

NOT AL

MOV DX,0FF28H

OUT DX,AL ;通过8253 A口控制LED

JMP L1 ;循环采样A/D转换的结果

CODE ENDS

END START

**【数据记录】**

输入模拟电压 二进制数字量

**【问答题】**

**1、0809获取A/D转换数据的方法有哪几种？比较这些方法的优劣。**

解：

（1）0809获取A/D转换数据的方法有3种：中断方式、查询方式、延时方式。

（2）查询方式主要由你编写的程序进行查询状态来获取数据的；

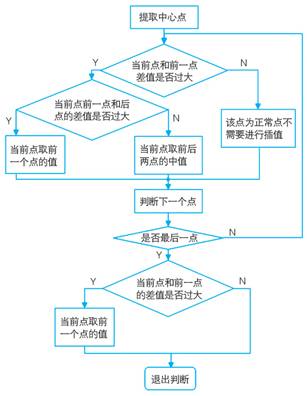
中断方式主要是通过系统本身的硬件机制再通过你自己编写的中断服务函数做出相应操作的；

两者主要区别在实时性上中断处理很及时，只要发生这个时间久会执行对应操作，查询的话要等到你查询状态之后才会做出相应操作。中断的主要优势在程序较复杂的情况下和对实时性要求高的情况下更明显。

**2、为获取比较平稳的数据显示，采取数据滤波措施，你能想到采用什么滤波措施比较合理。（列出算法，程序实现的流程图）**

解：

中值滤波法：这种方法的原理是将采集到的若干个周期的变量值进行排序，然后取排好顺序的值得中间的值，这种方法可以有效的防止受到突发性脉冲干扰的数据进入。在实际使用时，排序的周期的数量要选择适当，如果选择过小，可能起不到去除干扰的作用，选择的数量过大，会造成采样数据的时延过大，造成系统性能变差。



**【心得体会】**

这次的实验与上次相反，是A/D转换器实验。为了得到我们想要的效果，程序采用延时方式等待A/D转换结果。通过本实验程序的编写，我熟悉了ADC0809的使用方法，掌握A/D转换与计算机的接口方法，了解了模/数转换基本原理。本实验程序相对来说比较简单，主要涉及到对ADC0809和8255两芯片的端口写控制字，这在前面几个实验中已经多次运用。重点在于如何确认ADC0809转换数据完毕。

**实验六、七：8259中断控制**

**【实验目的】**

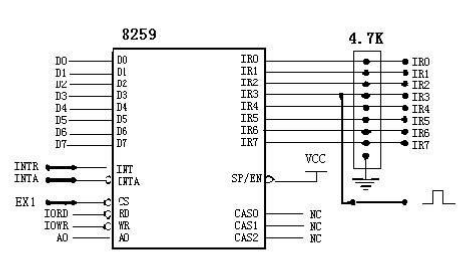
1. 学习8086/8088CPU中断系统的知识。
2. 学习8259中断控制器的使用。

**【实验内容】**

1，编写程序，使8255的A口控制LED灯。CPU执行主程序时四个绿灯亮。用“”作为8259的IR3的输入信号，向CPU请求中断。CPU在中断服务程序中熄灭绿灯，并使4个红灯亮。中断服务程序结束，又返回主程序，再使绿灯亮。

2，编写程序，使8255的A口控制LED灯。CPU执行主程序时四个绿灯亮。用“”作为8259的IR3的输入信号，向CPU请求中断。CPU在中断服务程序中每中断一次，使4个红色LED灯中每次只亮一个灯，并随每一次中断逐次移动一个灯的位置，中断服务程序结束，又返回主程序，再使绿灯亮。

**【实验区域电路连接图】**



**连线：**

8259的INT连8088的INTR（Xl5）；

8259的INTA连8088的INTA（Xl2）；

“”插孔和8259的3号中断IR3插孔相连；（单脉冲与时钟单元)；

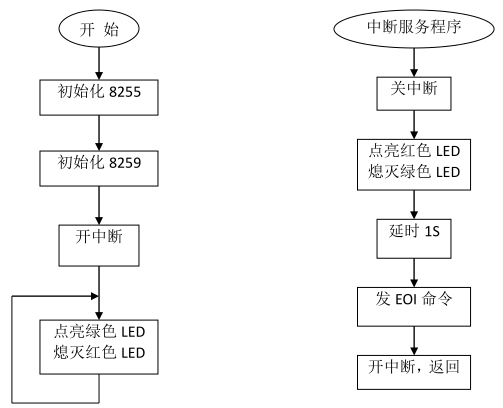
8259的CS端接EX1（60H）；

连JX4→JX17；IOWR→IOWR；IORD→IORD；A0→A0；

PA0…PA3→L2,L6，L10，L14；PA4…PA7→L3,L7,L11,L15。

**【程序框图】**

**1：**



**2：**

****

**【编程】**

**1:**

CODE SEGMENT

ASSUME CS:CODE

INTPORT1 EQU 0060H

INTPORT2 EQU 0061H

INTQ3 EQU INTREEUP3

INTCNT DB ?

ORG 1200H

START:CLD

MOV DX,0FF2BH

MOV AL,80H;设置8255方式字，A口出

OUT DX,AL

CALL WRINTVER ;WRITEINTRRUPT

MOV AL,13H ;ICW1=00010011B,边沿触发、单8259、需ICW4

MOV DX,INTPORT1

OUT DX,AL

MOV AL,08H ;ICW2=00001000B,IR3 进入则中断号 =0BH

MOV DX,INTPORT2

OUT DX,AL

MOV AL,09H;ICW4=00001001B,非特殊全嵌套方式、缓冲/从、正常EOI

OUT DX,AL

MOV AL,0F7H;OCW1=11110111B

OUT DX,AL

MOV INTCNT,01H;延时

STI

WATING:

MOV DX,0FF28H ;主程序绿灯亮（低四位为0则绿灯亮，高四位为1故红灯灭）

MOV AL,0F0H

OUT DX,AL

JMP WATING

WRINTVER:

MOV AX,0H

MOV ES,AX

MOV DI,002CH ;中断向量地址2CH=0BH\*4

LEA AX,INTQ3

STOSW ;送偏移地址

MOV AX,0000h

STOSW ;送段地址

RET

INTREEUP3:CLI;中断服务子程序开始

PUSH AX

PUSH DX

MOV DX,0FF28H;中断服务子程序执行红灯亮

MOV AL,0FH;低四位为1则绿灯灭，高四位为0故红灯亮

OUT DX,AL

CALL DELAY1S

MOV AL,20H ;OCW2=001 00 000B;非特殊EOI命令，结束命令，用于完全嵌套方式的中断结束

MOV DX,INTPORT1

OUT DX,AL

POP DX

POP AX

STI ;开系统中断

IRET

DELAY1S:

MOV CX,0FFFFH

MOV BX,5

L:DEC CX

JNZ L

DEC BX

JNZ L

RET

CODE ENDS

END START

**2:**

CODE SEGMENT

ASSUME CS:CODE

ORG 1200H

INTCNT DB 0EFH ;全局变量INTCNT=11101111B

START:

MOV DX,0FF2BH

MOV AL,80H ;设置8255方式字：A口出

OUT DX,AL

MOV AL,0FFH

MOV DX,0FF28H

OUT DX,AL

CALL Y2

MOV AL,13H;ICW1=00010011B,边沿触发、单8259、需ICW4

MOV DX,60H

OUT DX,AL

MOV AL,08H ;ICW2=00001000B,IR3（ICW2低两位自动跳变为11，即00001011B）进入则中断号=0BH

MOV DX,61H

OUT DX,AL

MOV AL,09H ;ICW4=00001001B,非特殊全嵌套方式、缓冲/从、正常EOI

OUT DX,AL

MOV AL,0F7H;OCW1=11110111B

OUT DX,AL

MOV INTCNT,0EFH

STI

Y1:

MOV DX,0FF28H ;主程序绿灯亮（低四位为0则绿灯亮，高四位为1故红灯灭）

MOV AL,0F0H ;D7…D4=1111，D3…D0=0000

OUT DX,AL

JMP Y1 ;循环

Y2:

MOV AX,0H

MOV ES,AX

MOV DI,002CH ;中断向量地址2CH=0BH\*4

LEA AX,INTREEUP3

STOSW ;送偏移地址

MOV AX,0000H

STOSW ;送段地址

RET

INTREEUP3:

CLI

PUSH AX ;保护现场

PUSH DX

MOV DX,0FF28H;中断服务子程序执行红灯亮

MOV AL,INTCNT

OUT DX,AL

ROL INTCNT,1 ;循环左移一位

JC Y4

MOV INTCNT,0EFH

Y4:

MOV AX,10H ;延时服务程序

INTRE0:

MOV CX,10000

LOOP $

DEC AX ;AX=AX-1

JNZ INTRE0

INTER2: ;设置OCW2

MOV AL,20H ;D5=EOI=1，发送EOI指令结束中断

MOV DX,60H

OUT DX,AL

POP DX

POP AX

STI;开系统中断

IRET

CODE ENDS

END START

**【心得体会】**

本次的实验分两次进行，循序渐进，有助于我们由浅入深地了解8259中断控制。在实验过程中,我渐渐熟悉了8086/8088CPU中断系统的知识以及8259中断控制器的使用。还有就是在关于保护现场时是先保护偏移地址然后是段地址，而在现场恢复时是先恢复段地址，然后是偏移地址，在刚开始是弄反了，发生了一些错误，这个需要注意到。还有就是学到了巧妙地利用循环平移的指令实现循环的操作。中断的思想在程序的设计中无疑占有重要的位置.采取中断的方法,大大提高程序的有效性和CPU的高效率工作.因此,8259中断控制实验在微机的几个实验中,就显得尤其重点.掌握8259中断控制器的工作原理和应用编程方法,需要对8259系统有一个深入的了解,这些是我们学习的重点,应该熟练掌握。