

实验三 可编程定时器 / 计数器（8253）

一、实验目的

掌握8253的基本工作原理和编程方法，观察OUT引脚输出电平的变化情况。

二、实验内容

- 1、按图3-1虚线连接电路，将计数器0设置为方式0，计数器初值为N(N=4)，用手动逐个输入单脉冲，编程使计数值在屏幕上显示，并通过LED灯观察OUT0电平变化。

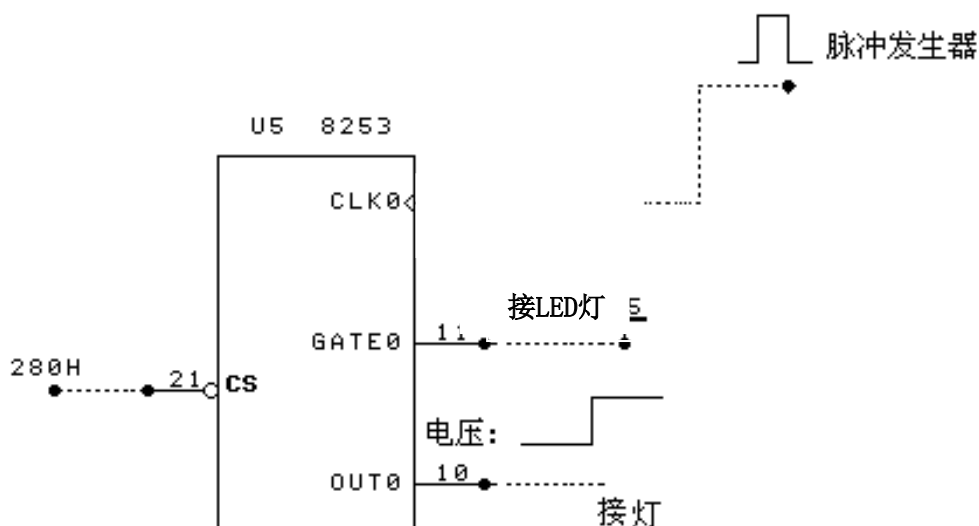


图3-1 定时计数器实验电路连接图

- 2、利用图3-1连接的电路，实现方式1~方式5的验证。（注意：对于需要硬件触发的方式1和方式5，GATE0需接到脉冲发生器产生负向脉冲的插孔上，以便产生上升沿的启动信号。）
- 3、按图3-2连接电路，将计数器0、计数器1分别设置为方式3，计数初值设为1000，用LED灯观察OUT1输出电平的变化(频率1HZ)。

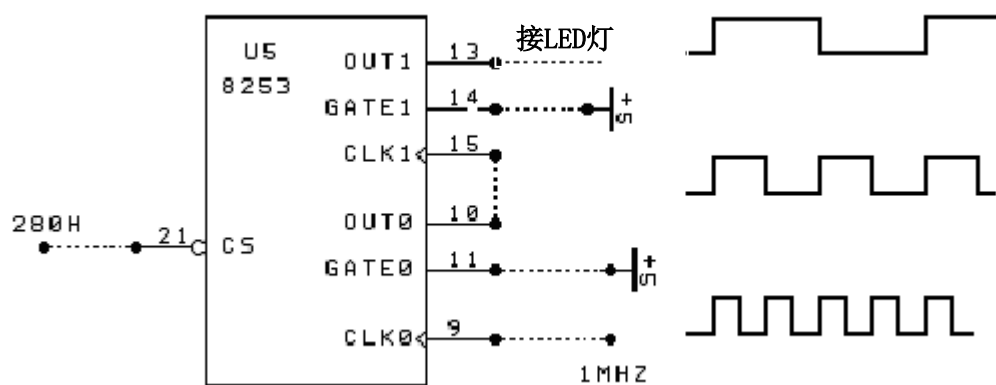


图3-2 方式3下定时计数器实验电路连接图

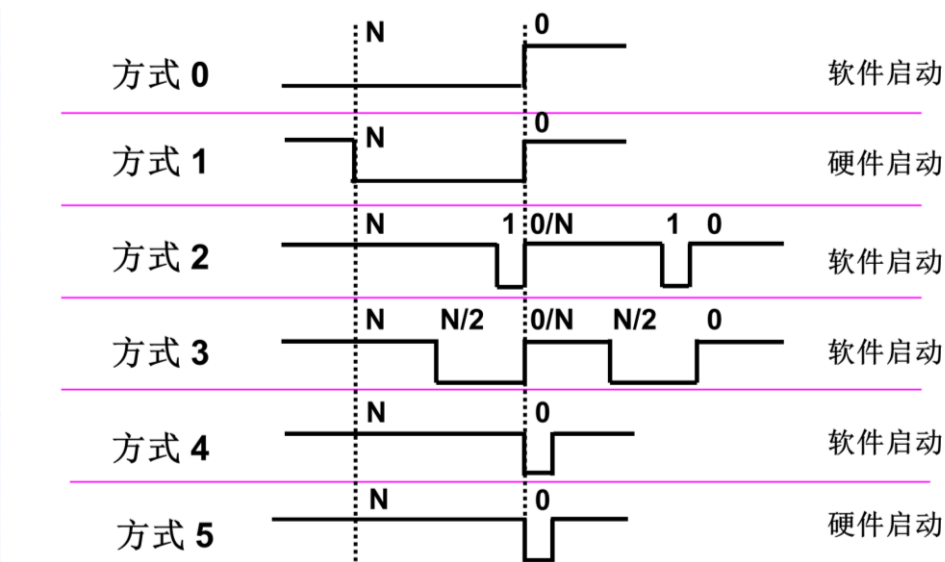
三、编程提示

- 1、8253控制寄存器地址 283H
 计数器0地址 280H
 计数器1地址 281H

2、8253方式控制字

计数器	读写格式	工作方式	数制
D7 D6	D5 D4	D3 D2 D1	D0
00: 计数器 0	00: 计数器锁存命令	000: 方式 0	0: 二进制
01: 计数器 1	01: 只读写低字节	001: 方式 1	1: 十进制
10: 计数器 2	10: 只读写高字节	
11: 非法	11: 先读写低字节, 后读写高字节	101: 方式 5	

3、6种工作方式输出波形对比



4、参考程序:

```
程序 ①
;*****;
;* 8253方式0计数器实验 *;
;*****;
data segment
    io8253_0    equ 280h    ;计数器0地址
    io8253_k    equ 283h    ;控制口地址
    counter     equ 04h
    msg db 'Current Count Value: $'
data ends
code segment
    assume  cs:code,ds:data
```

```

    main proc far
start:
    mov ax, data
    mov ds, ax
    mov dx, io8253_k
    mov al, 10h          ;设置8253计数器0为工作方式0, 二进制计数
    out dx, al
    mov dx, io8253_0
    mov al, counter      ;送计数初值为counter
    out dx, al
    call Init_Screen     ;清屏、定光标位置等初始化操作
L5:
    mov dx, io8253_0
    in al, dx            ;读计数初值
    call disp            ;调显示子程序
    cmp al, 0            ;计数结束, 返回DOS
    jz L1
    mov ah, 01h
    int 16h              ;检测键盘状态, 无键按下时, ZF=1
    jz L5
L1: mov ax, 4c00h        ;退出
    int 21h
    main endp

```

```

Init_Screen proc near
    mov ah, 6
    mov al, 0
    mov bh, 7
    mov ch, 0
    mov cl, 0
    mov dh, 24
    mov dl, 79
    int 10h              ;清屏
    mov bh, 0
    mov dx, 1010h
    mov ah, 2
    int 10h              ;定位光标位置于第10h行第10h列
    mov ah, 09h
    lea dx, msg
    int 21h              ;显示提示信息"Current Count Value: "
    mov dx, io8253_0
    in al, dx
    mov bl, al           ;用于检测第一个脉冲的到来
L4:

```

```

    mov dx, io8253_0
    in al, dx          ;输入第1个脉冲前显示'?'
    cmp al, bl
    jnz L6
    mov al, '?'
    mov bh, 0
    mov cx, 1
    mov ah, 0ah        ;在原位置处显示字符，光标不向下移
    int 10h
    jmp L4
L6:
    ret
Init_Screen endp

disp proc near        ;显示子程序
    push ax
    add al, 30h
    mov bh, 0
    mov cx, 1
    mov ah, 0ah        ;显示
    int 10h
    pop ax
    ret                ;子程序返回
disp endp
code ends
end start

```

程序 ②

```

;*****
;*      8253分频      *
;*****
io8253_0      equ 280h
io8253_1      equ 281h
io8253_k      equ 283h
code segment
    assume    cs:code
    main proc far
start:
    mov dx, io8253_k    ;向8253写控制字
    mov al, (    )      ;计数器0工作在方式3下
    out dx, al
    mov ax, 1000        ;写入循环计数初值1000
    mov dx, io8253_0
    out dx, al          ;先写入低字节

```

```

    mov al, ah
    out dx, al          ;后写入高字节
    mov dx, (          )
    mov al, (          ) ;计数器1工作在方式3
    out dx, al
    mov ax, 1000        ;写入循环计数初值1000
    mov dx, (          )
    out dx, al          ;先写低字节
    mov al, ah
    out dx, al          ;后写高字节
    mov ax, 4c00h       ;程序退出
    int 21h
    main endp
code ends
end start

```