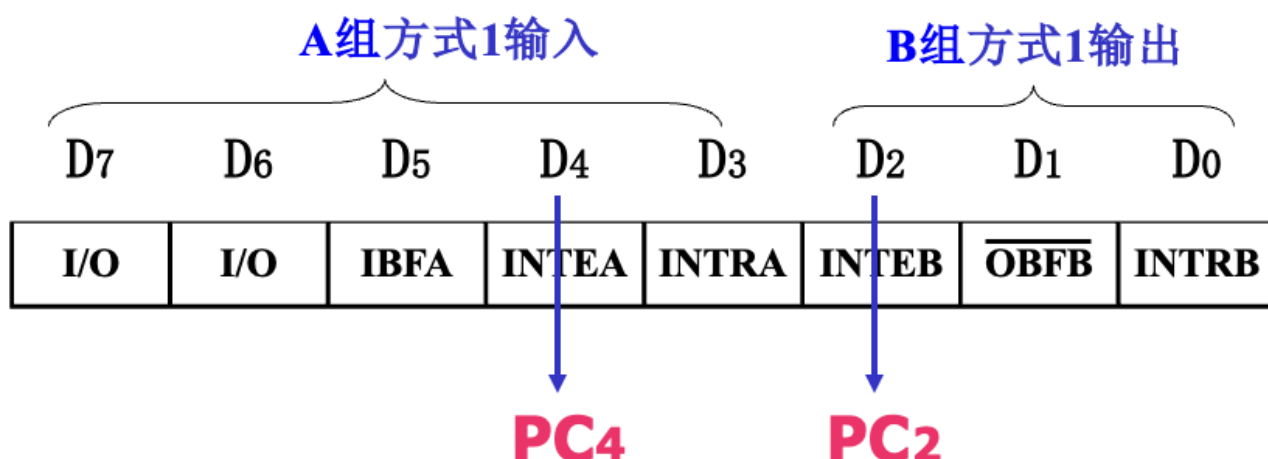


第4次作业

10.4

10.4 设定8255A的端口A为方式1输入，端口B为方式1输出，则读取端口C的数据的各位是什么含义？



- PC0: INTEB (中断请求信号)
- PC1: ACKB (响应信号)
- PC2: INTRA (中断请求信号)
- PC3: STBA (发送信号)
- PC4: ACRA (响应信号)
- PC5: IOBFRA (输入准备就绪线控制信号)
- PC6, PC7: I/O信号

10.5

10.5 对8255A的控制寄存器写入B0H，则其端口C的PC5引脚是什么作用的信号线？

- 判断线带4位计数输入是否编码端点输入控制信号
- 方式控制字为B0H = 10110000B，说明A组为方式1输入，它将征用PC5引脚作为输入缓冲器满信号IBFA

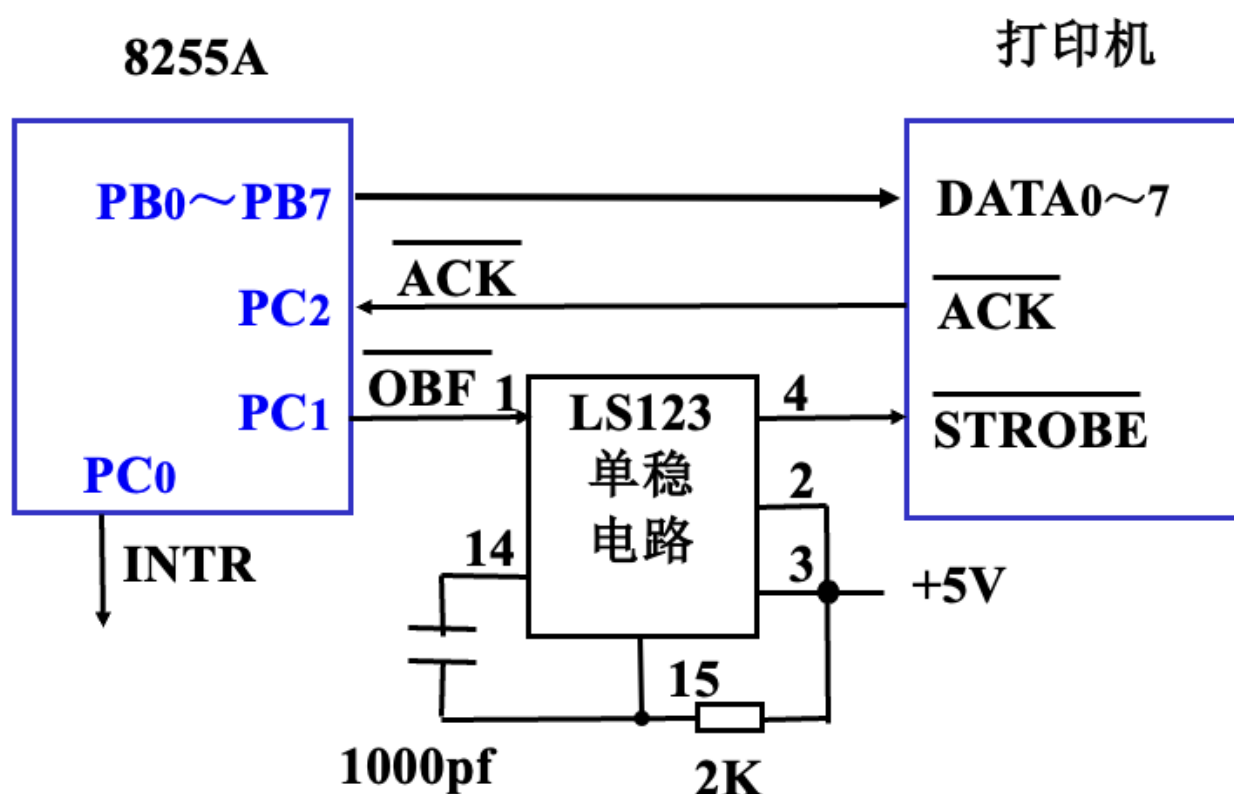
10.6

10.6 10.2.2节用8255A端口A方式0与打印机接口，如果采用端口B，其他不变，请说明应该如何修改接口电路和程序。

- 接口电路：用端口引脚PB0~PB7与打印机DATA0~7连接，其他不变
- 程序：用端口B的I/O地址FFFAH替换端口A的FFF8H
- 另外，应设置端口B为方式0输出。但由于原控制字已经做到，所以无需修改

10.7

10.7 10.2.3节用8255A端口A方式1与打印机接口，如果采用端口B，其他不变，请说明如何修改接口电路和程序。



```

mov     dx, 0FFFEh

mov     al, 84h
out     dx, al

mov     al, 04h
; Set INTEB (PC2) to 0 to disable interrupt
out     dx, al

; ...

mov     cx, counter      ; Load the number of
bytes to print into CX
mov     bx, offset buffer ; Load the address of
the string into BX
call    prints           ; Call the print
subroutine

prints  proc
    push    ax           ; Save AX register
    push    dx           ; Save DX register

print1:
    mov     al, [bx]     ; Load a character
from the string
    mov     dx, 0FFFAh
    out     dx, al       ; Output the character
to port B

    mov     dx, 0FFCh
print2:
    in      al, dx
    test    al, 02h      ; Check if (PC1) is

```

```

set to 1
    jz      print2          ; If not, loop until
it is set to 1

    inc     bx              ; Move to the next
character
    loop    print1

    pop     dx              ; Restore DX register
    pop     ax              ; Restore AX register
    ret

prints endp

```

10.8

```

;写入方式字
mov al,81h
mov dx,控制口地址
out dx,al
;使L0~L3全亮
mov al,0fh
mov dx,端口C地址
out dx,al
;控制程序段
mov dx,端口C地址
in al,dx
mov cl,4
shl al,cl
out dx,al

```

10.16

```
table    db 0c0h
          db 0f9h,0a4h,0b0h,99h
          db 92h,82h,0f8h,80h
          mov dx,0fffbh
          mov al,89h
          out dx,al
again0:   mov dx,0fffaH
          in al,dx
          mov cx,8
          mov ah,01h
again1:   shr al,1
          jnc disp0
          inc ah
          loop again1
          jmp disp1
disp0:    mov bx,offset table
          mov al,ah
          xlat
          mov dx,0fff8h
          out dx,al
disp1:    mov cx,8
          mov al,01h
          mov dx,0fff9h
disp2:    out dx,al
          call delay
          shl al,1
          loop disp2
          jmp again0
```