接口：CPU 与外界的连接电路，可以
（1）执行 CPU 命令
（2）返回外设状态
（3）数据缓冲
（4）信号转换
（5）设备选择
（6）数据宽度与数据格式转换

设备接口与 CPU 之间的数据交换：
（1） 查询
（2） 中断
（3） DMA

端口：接口电路中能被 CPU 访问的寄存器的地址。

独立编址：
优点：
（1） 不占用存储器空间
（2） 使用专门指令，短，执行速度快
（3） 简化地址译码电路硬件
（4） 编写出程序可读性好
（5） 地址可重叠，且不会相互混淆

缺点：需要增加/IOR 和/IOW 控制引脚。

统一编址：
优点：
（1） 增强 I/O 处理能力
（2） 给端口带来较大寻址空间
缺点：
（1） 占用存储器地址空间，使存储器容量减小
（2） 指令长，执行时间长
（3） 必须全地址线译码，增加了地址线

I/O 端口地址译码：将地址总线上的地址信号翻译成所要访问的端口
（1） 全译码
（2） 部分译码
（3） 开关式译码

GAL： 地址范围 300H~31FH

表 3.7 GAL 器件的 300H～31FH 范围的译码器地址线取值

0 0	A_9 A_8 A_7 A_6 A_5	A_4 A_3 A_2	A_1 A_0
	1 1 0 0 0	I_X I_X I_X	? ?

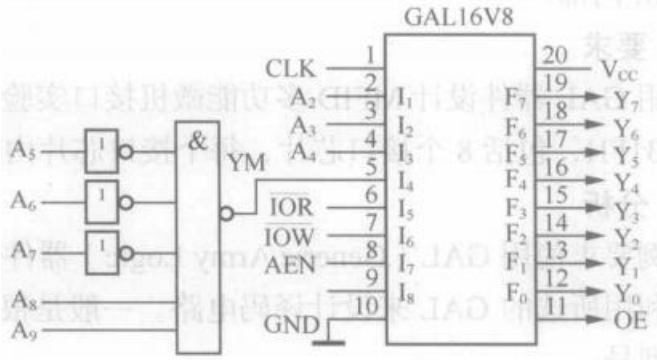


图 3.5　采用 GAL16V8 的地址译码电路

下面讨论 GAL 编程输入源文件中，产生 8 个输出信号（$\overline{Y}_0 \sim \overline{Y}_7$）的逻辑表达式[1]。

$\overline{Y}_0 = A_9*A8*/A7*/A6*/A5*/A4*/A3*/A_2*/AEN*/IOR + A_9*A8*/A7*/A6*/A5*/A4*/A3*/A_2*/AEN*/IOW$

$\overline{Y}_1 = A_9*A8*/A7*/A6*/A5*/A4*/A3*A_2*/AEN*/IOR + A_9*A8*/A7*/A6*/A5*/A4*/A3*A_2*/AEN*/IOW$

$\overline{Y}_2 = A_9*A8*/A7*/A6*/A5*/A4*A3*/A_2*/AEN*/IOR + A_9*A8*/A7*/A6*/A5*/A4*A3*/A_2*/AEN*/IOW$

$\overline{Y}_3 = A_9*A8*/A7*/A6*/A5*/A4*A3*A_2*/AEN*/IOR + A_9*A8*/A7*/A6*/A5*/A4*A3*A_2*/AEN*/IOW$

[1] 按照 GAL 器件编程输入源文件的格式要求，表达式中的逻辑符号“非”采用斜杠“/”，而不使用上划线

$\overline{Y}_4 = A_9*A8*/A7*/A6*/A5*A4*/A3*/A_2*/AEN*/IOR + A_9*A8*/A7*/A6*/A5*A4*/A3*/A_2*/AEN*/IOW$

$\overline{Y}_5 = A_9*A8*/A7*/A6*/A5*A4*/A3*A_2*/AEN*/IOR + A_9*A8*/A7*/A6*/A5*A4*/A3*A_2*/AEN*/IOW$

$\overline{Y}_6 = A_9*A8*/A7*/A6*/A5*A4*A3*/A_2*/AEN*/IOR + A_9*A8*/A7*/A6*/A5*A4*A3*/A_2*/AEN*/IOW$

$\overline{Y}_7 = A_9*A8*/A7*/A6*/A5*A4*A3*A_2*/AEN*/IOR + A_9*A8*/A7*/A6*/A5*A4*A3*A_2*/AEN*/IOW$

每个表达式的右边都是两个与或式，而前后两个与式中的不同在于读项和写项的差别，前者是读有效（$\overline{IOR}$ =0），后者是写有效（$\overline{IOW}$ =0），这表示该端口既可读又可写。各表达式的左侧 Y 项是与或式的输出值，即 GAL 器件译码输出的片选信号。

由 $\overline{Y}_0 \sim \overline{Y}_7$ 产生 8 个接口芯片的片选信号，再加上不参加译码的最低 2 位 00～11 变化可得：$\overline{Y}_0$ =300H～303H，$\overline{Y}_1$ =304H～307H，$\overline{Y}_2$ =308H～30BH，$\overline{Y}_3$ =30CH～30FH……$\overline{Y}_7$ =31CH～31FH。

定时/计数器 8254A

方式 0：计数到 0 结束输出正跃变信号方式

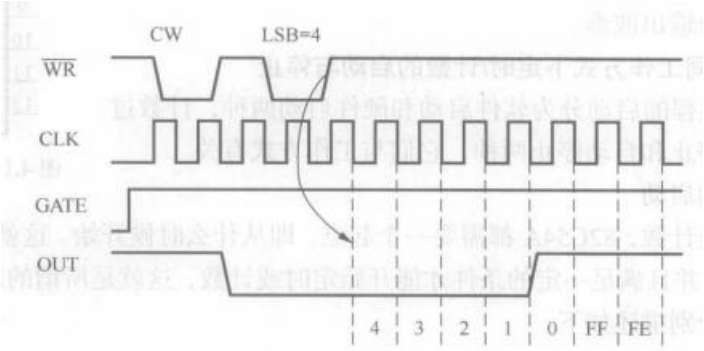


图 4.2　0 方式输出波形

方式 1：硬件可重触发单稳方式

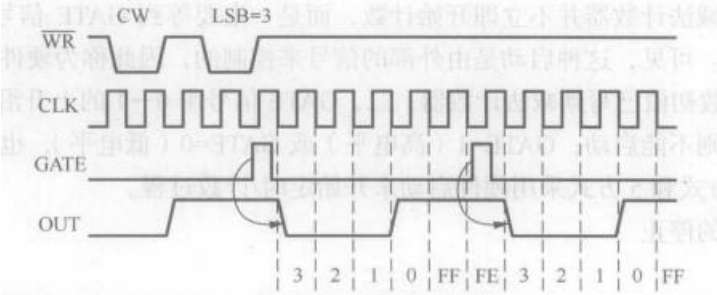


图 4.3　1 方式输出波形

方式 2：频率发生器方式

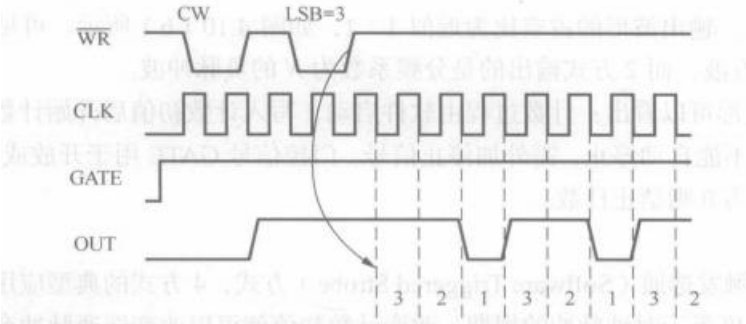


图 4.4　2 方式输出波形

方式 3：方波发生器方式

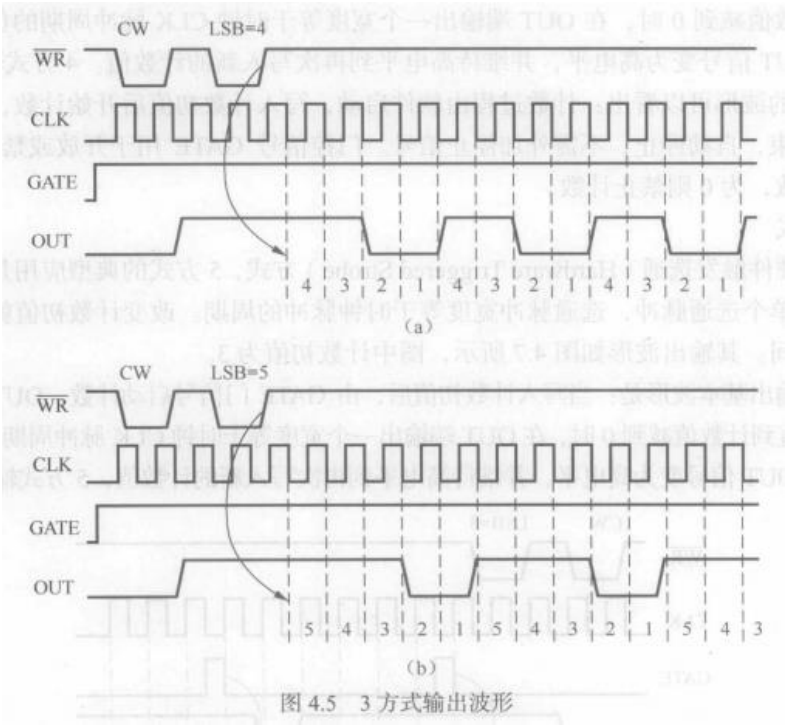


图 4.5　3 方式输出波形

方式 4：软件触发选通方式

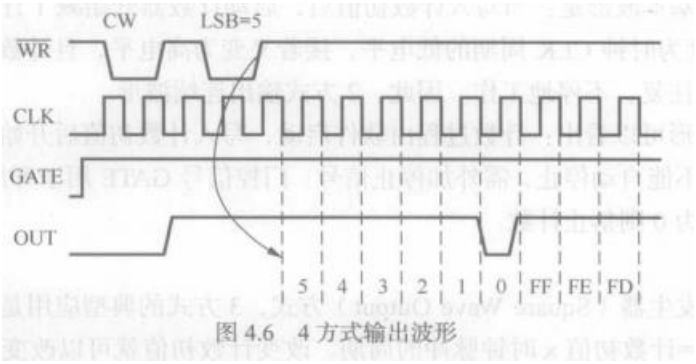


图 4.6　4 方式输出波形

方式 5：硬件触发选通方式

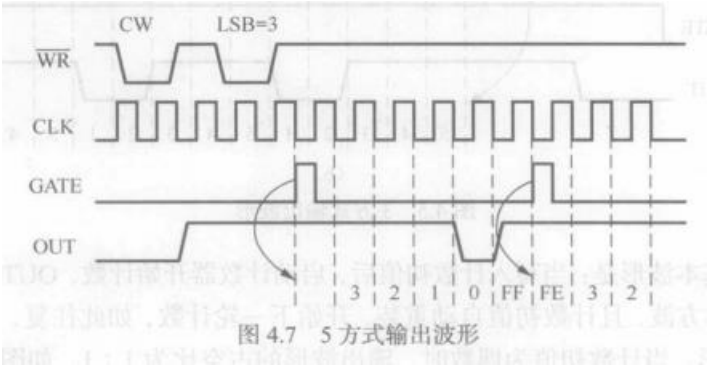


图 4.7　5 方式输出波形

控制字：

图 4.10　82C54A 工作方式命令格式

锁存：

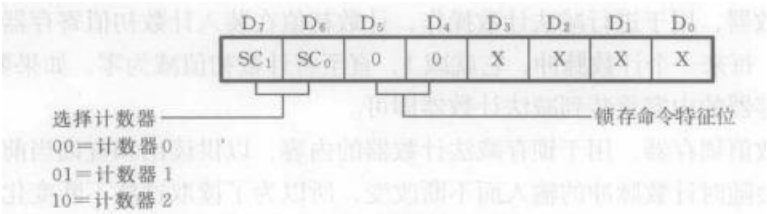


图 4.11　锁存命令格式

读回：（CNT ST 1 不读，0 读）

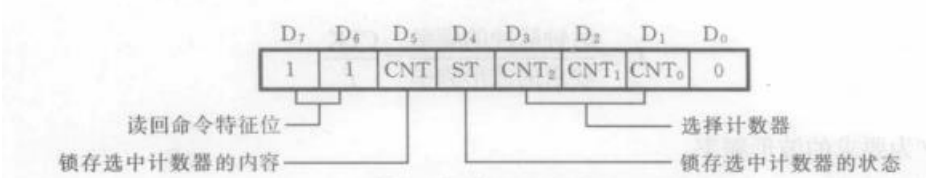


图 4.12　读回命令格式

计数初值、定时常数：

（1）要求产生定时时间间隔的定时常数 T_C

$$T_C=\frac{\text{要求定时的时间}}{\text{时钟脉冲周期}}=\frac{\tau}{1/\text{CLK}}=\tau\times\text{CLK}$$

式中，τ为要求定时的时间；CLK 为时钟脉冲频率。

例如，已知 CLK=1.19318MHz，τ=5ms，求 T_C，则

$$T_C=5\times10^{-3}\text{s}\times1193180/\text{s}=5965$$

（2）要求产生频率为 f 的信号波形的定时常数 T_C

$$T_C=\frac{\text{时钟脉冲的频率}}{\text{要求的波形频率}}=\frac{\text{CLK}}{f}$$

式中，f 为要求的波形频率。

例如，已知 CLK=1.19318MHz，f=800Hz，求 T_C，则

$$T_C=\frac{1.19318\times10^6\text{Hz}}{800\text{Hz}}=1491$$

例题：

1. 要求

某应用系统中，要求方波发生器产生 f=1000Hz 的方波，系统提供的输入时钟 CLK=1.19318MHz，采用二进制计数。试写出分频器的初始化程序。

2. 分析

（1）选择工作方式

为了产生方波，选择 82C54A 的 3 方式是合适的。为此，利用 82C54A 的计数器 0，将它的

OUT_0 作为方波输出。

（2）计算计数初值

将系统提供的 CLK 作为计数器 0 的输入时钟 CLK_0，按照要求输出 OUT_0=1000Hz 的方波，根据式（4-2），可得定时常数为

$$T_C=CLK_0/OUT_0=1.19318MHz/1000Hz=1193=4A9H$$

3. 初始化程序

分频器汇编语言初始化程序段如下。

```
MOV DX,307H        ;82C54A 的命令口
MOV AL,36H         ;方式命令
OUT DX,AL
MOV DX,304H        ;计数器 0 的数据口
MOV AX,4A9H
OUT DX,AL          ;装入定时常数低字节
MOV AL,AH
OUT DX,AL          ;装入定时常数高字节
```

中断控制 8259A

外部中断：可屏蔽中断 INTR，不可屏蔽中断 NMI

中断响应周期：CPU 收到外部设备通过中断控制器发出的中断请求 INT 后，如果当前一条指令已经执行完，且中断标志位 IF=1 时（即允许中断），又没有 DMA 请求，那么 CPU 进入中断响应周期，发出两个连续中断应答信号 INTA，分别确定中断优先级和送出中断向量，CPU 读入。

初始化命令格式：

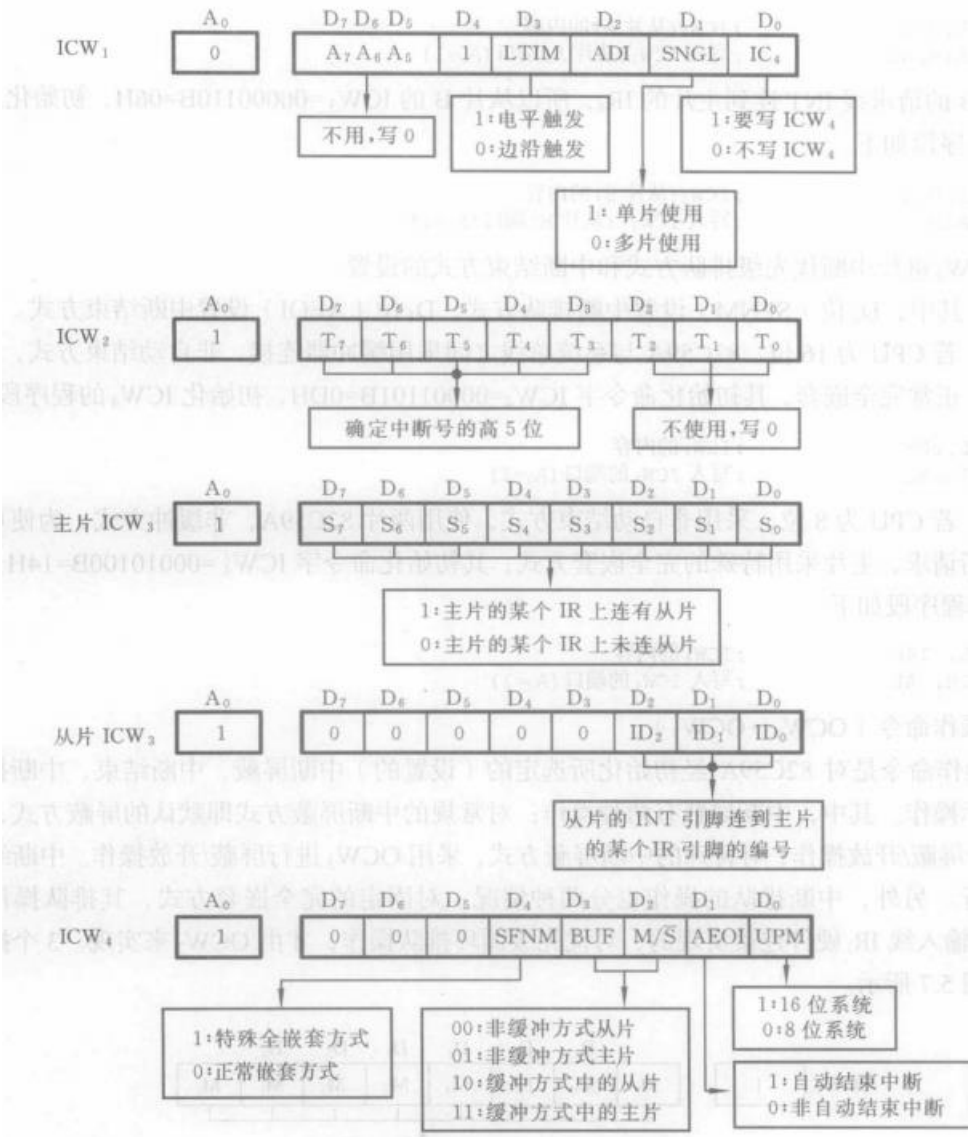


图 5.6　82C59A 初始化命令格式

③ ICW_3进行级联方式设置。

8 位，ICW_3命令只有系统存在 2 片以上 82C59A 时才启用，否则不用 ICW_3命令。分主片和从片，分开设置。主片级联方式命令 ICW_3的 8 位，表示 IR_i的哪一个输入引脚上有从片连接，若有，该位写 1；若无，该位写 0。如果主片的 IR_4上有从片连接，则主片的 ICW_3=10H。从片的 8 位，表示它的中断请求线 INT 连到了主片哪一个 IR_i上，若连到主片的 IR_4，则从片的 ICW_3=04H。

例如，假设主片的 IR_3和 IR_6两个输入端分别连接了从片 A 与 B 的 INT，故主片的 ICW_3=01001000B=48H，初始化主片的 ICW_3程序段如下。

```
MOV  AL,48H          ;ICW3(主)的内容
OUT  21H,AL          ;写入 ICW3(主)的端口(A0 =1)
```

从片 A 的请求线 INT 连到主片的 IR_3，所以从片 A 的 ICW_3=00000011B=03H，初始化从片 A 的 ICW_3程序段如下。


```
MOV AL,03H           ;ICW3(从片 A)的内容
OUT 0A1H,AL          ;写入 ICW3(从片 A)端口(A0=1)
```

从片 B 的请求线 INT 连到主片的 IR_6，所以从片 B 的 ICW_3=00000110B=06H，初始化从片 B 的 ICW_3程序段如下。

```
MOV AL,06H           ;ICW3(从片 B)的内容
OUT 0A1H,AL          ;写入 ICW3 (从片 B)端口(A0=1)
```

```
INTA00 EQU 020H      ;8259A 主片端口(A0=0)
INTA01 EQU 021H      ;8259A 从片端口(A1=0)
... ...
MOV AL,11H          ;ICW1
OUT INTA00,AL
JMP SHORT $ +2

MOV AL,8            ;ICW2
OUT INTA01,A1
JMP SHORT $ +2

MOV A1,04H          ;ICW3,主片的 IR2 接上从片
OUT INTA01,A1
JMP SHORT $ +2

MOV AL,01H          ;ICW4
OUT INTA01,A1
... ...


INTB00 EQU 0A0H ;  8259A 从片端口(A0=0)
```

```
INTB01 EQU 0A1H ;  8259A 从片端口(A0=1)
... ...
MOV AL,11H
OUT INTB00,Al
JMP SHORT $ +2

MOV AL,70H
OUT INTB01,Al
JMP SHORT $ +2

MOV Al,02H          ;从片接主片的 IR2
OUT INTB01,Al
JMP SHORT $ +2

MOV AL,01H
OUT INTB01,Al
... ...
```

修改中断向量：

```
CLI                          ;关中断
MOV AH,35H                   ;取原中断向量
MOV AL,N
INT21H
MOV OLD_SEG,ES               ;保存原中断向量
MOV OLD_OFF,BX

MOV DX,SEG INTRnew           ;设置新中断向量
MOV DS,DX                    ;DS 指向新中断服务程序段基址
MOV DX,OFFSET INTRnew        ;DX 指向新中断服务程序偏移量
MOV AL,N                     ;中断号
MOV AH,25H
INT21H
STI                          ;开中断
  :
MOV DX,OLD_SEG               ;恢复原中断向量
MOV DS,DX
MOV DX,OLD_OFF
MOV AH,25H
MOV AL,N
INT21H
```

中断服务程序：

```
NEW_INT PROC  FAR
  (寄存器进栈)              ;现场保护
       :
  (服务程序主体)
       :
    MOV AL,20H              ;向从片 82C59A 发结束命令
    MOV DX,0A1H
    OUT DX,AL
    OUT 20H,AL              ;向主片 82C59A 发结束命令
   (寄存器出栈)             ;恢复现场
    IRET                    ;中断返回
NEW_INT ENDP
```

程序设计：

3. 程序设计

包括主程序和中断服务程序，可以对照 5.9.1 节所阐明的中断向量修改、中断的屏蔽与开放及中断服务程序的格式来分析下面的程序。

汇编语言主片 82C59A 的中断服务程序如下。

```
STACK SEGMENT
            DW 200 DUP(?)                    ;堆栈区地址空间
STACK ENDS
DATA  SEGMENT
      OLD_IV  DD ?                           ;保存原中断向量的双字单元
      MK-BUF  DB ?                           ;保存原屏蔽字的字节单元
         BUF  DB'OK !',0DH,0AH,$             ;提示符
       COUNT  DB  (0)                        ;计数单元，初值为 0
DATA  ENDS

CODE SEGMENT                                 ;主程序开始
   ASSUME CS:CODE,DS:DATA,SS:STACK
START: MOV AX,DATA
       MOV DS,AX
       IN  AL,21H                            ;保存 82C59A 原屏蔽字
       MOV MK-BUF,AL
       CLI                                   ;关中断
       AND AL,01111111B                      ;7FH 是开放主片 IR7 的屏蔽码
       OUT 21H,AL
       CALL GET_IV                           ;获取原中断向量的子程序
       CALL SET_IV                           ;设置新中断向量的子程序

L1:    STI                                   ;开中断
       CMP COUNT,08H                         ;计数是否到 8 次
       JNZ  L1                               ;未到，继续;已到，恢复原向量和原屏蔽字
                                             ;显示 "OK" 并结束程序
       CLI                                   ;关中断
       CALL RENEW_IV                         ;恢复原中断向量的子程序
       MOV AL,MK-BUF                         ;恢复 82C59A 原屏蔽字
       OUT 21h,AL
       STI                                   ;开中断
       MOV AX,SEG BUF                        ;显示提示符 "OK!"
       MOV DS,AX
       MOV DX,OFFSET BUF
```

```
        MOV AH,09H
        INT 21H
        MOV AX,4C00H                        ;返回 DOS
        INT 21H

SW_INT  PROC FAR                            ;用户中断服务程序开始
        STI                                 ;开中断
        PUSH AX                             ;寄存器进栈
        INC COUNT                           ;计数加 1
        CLI                                 ;关中断
        MOV AL,67H                          ;发中断结束命令(OCW2,指定结束方式)
        OUT 20H,AL
        POP AX                              ;寄存器出栈
        IRET                                ;中断返回
SW_INT  ENDP                                ;用户中断服务程序结束

GET_IV  PROC NEAR                           ;获取原中断向量的子程序
        PUSH AX
        PUSH BX
        MOV AX,350FH
        INT 21H
        MOV WORD PTR OLD_IV,BX
        MOV WORD PTR OLD_IV+2,ES
        POP BX
        POP AX
        RET
GET_IV  ENDP

SET_IV  PROC NEAR                           ;设置新中断向量的子程序
        PUSH AX
        PUSH DX
        PUSH DS
        MOV AX,CODE
        MOV DS,AX
        MOV DX,OFFSET SW_INT
        MOV AX,250FH
        INT 21H
        POP DS
        POP DX
        POP AX
        RET
SET_IV  ENDP

RENEW_IV PROC NEAR                          ;恢复原中断向量的子程序
         PUSH AX
         PUSH DX
         MOV DX,WORD PTR OLD_IV
         MOV DS,WORD PTR OLD_IV+2
         MOV AX,250FH
         INT 21H
         POP DX
         POP AX
         RET
RENEW_IV ENDP
CODE    ENDS                                ;主程序结束
         END START
```

并行接口 8255A

1. 0 方式的特点与功能

0 方式是一种无条件的数据传输方式，也是 82C55A 的基本输入/输出方式。

0 方式的特点为：82C55A 做单向数据传送，即一次初始化只能把某个并行端口置成输入或输出，不能置成既输入又输出；不要求固定的联络（应答）信号，无固定的工作时序和固定的工作状态字；适用于采用无条件或查询方式与 CPU 交换数据，不能采用中断方式交换数据。因此，0 方式使用起来不受什么限制。

0 方式的功能为：A 端口做数据端口（8 位并行）；B 端口做数据端口（8 位并行）；C 端口做数据端口（分高 4 位和低 4 位，4 位并行），或做位控，按位输出逻辑 1 或逻辑 0。

图 7.2 的最高位 D_7 是特征位，因为 82C55A 有两个命令，用特征位加以区别：D_7=1，表示是方式命令；D_7=0，表示是按位置位/复位命令。

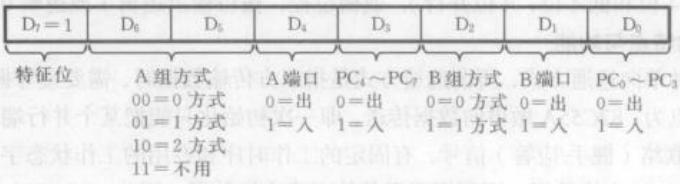


图 7.2　82C55A 方式命令的格式

从方式命令的格式可知，A 组有 3 种方式（0 方式、1 方式、2 方式），而 B 组只有两种工作方式（0 方式、1 方式）。C 端口分成两部分，上半部属 A 组，下半部属 B 组。对 3 个并行端口的输入输出设置是，置 1 指定为输入，置 0 指定为输出。

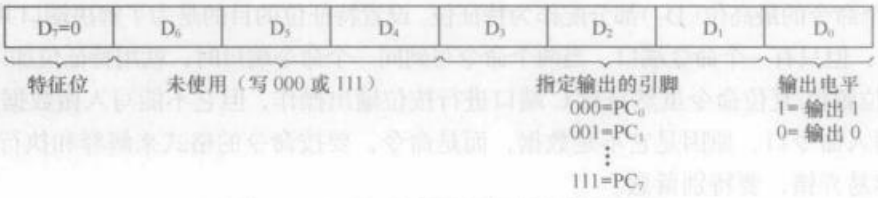


图 7.3　82C55A 按位置位/复位命令的格式

程序：

2. 当将LED的阳极接高电平，阴极接低电平时，LED就会发光；否则就会熄灭。如图所示，某系统应用并行接口82 C55 A连接LED电路和按键，用LED作为显示设备，用按键作为输人设备。82C55A的PC口外接8个发光二极管L0～L7。用PA1外接一个按键K。试编写程序，实现下述功能：每按一次按键，使PB口上的发光二级管按L0、L1、L2...L7次序循环点亮显示。已知：82C55A的PA口地址是218H。

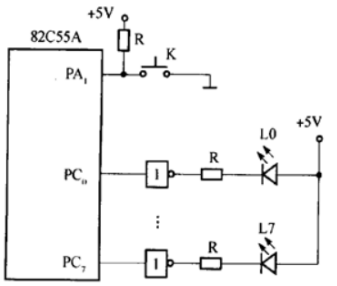
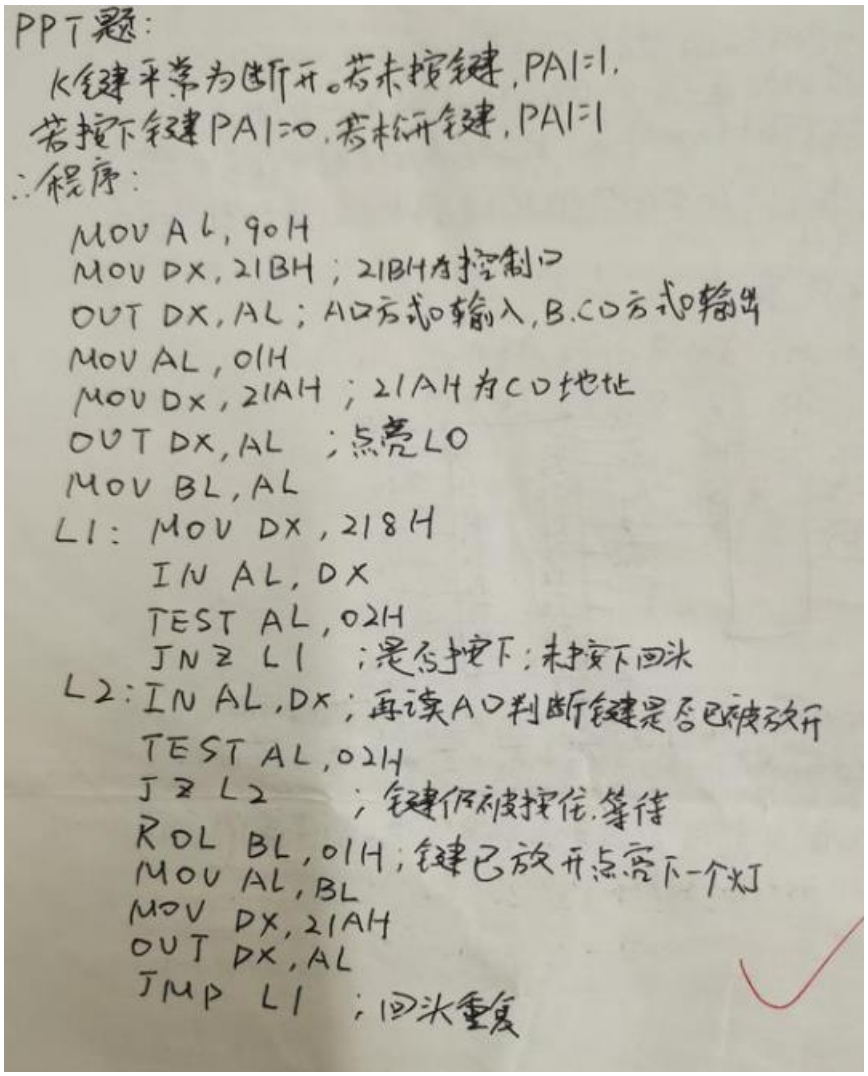


图 10-16　82C55A 应用于 LED 电路



串行通信：

波特率：每秒传输串行数据的位数。

发送/接受时钟作用：进行移位，执行数据的发送和接收。

波特率因子：每传输一个数据位需要多少个移位脉冲。

帧格式：起始位、数据位、奇偶校验位、停止位、空闲位

常见错误类型：奇偶校验错、溢出错、帧格式错、超时错