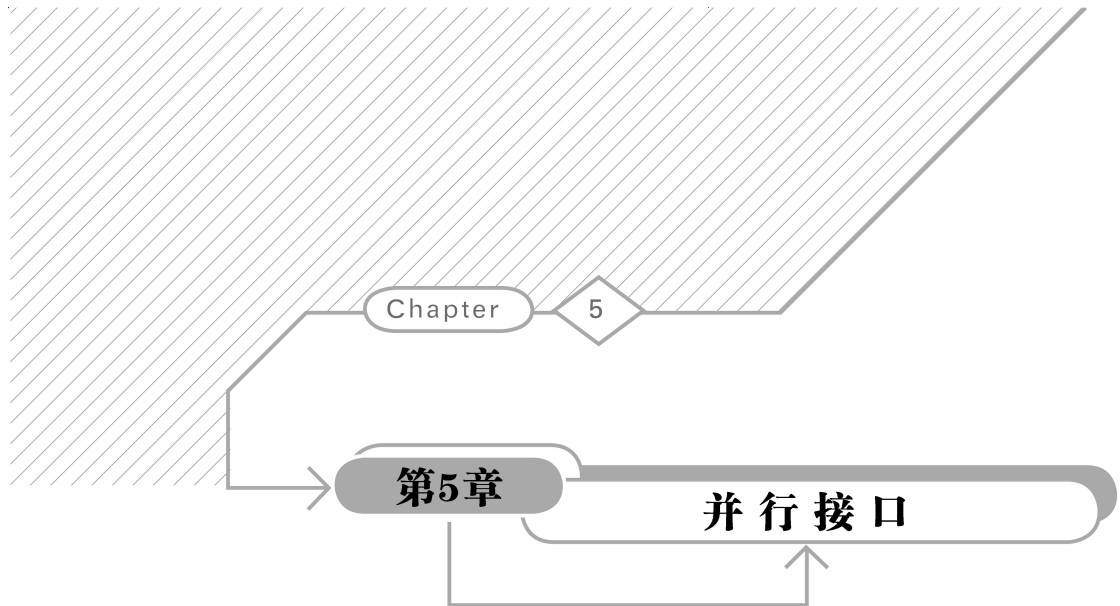




本章学习目标

- 了解并行接口的基本功能；
- 熟练掌握可编程并行接口芯片 8255A 的内部结构及引脚；
- 熟练掌握 8255A 的 3 种工作方式和功能；
- 熟练掌握 8255A 的应用设计方法。

本章首先介绍了可编程并行接口芯片 8255A 的内部结构及引脚,然后介绍了 8255A 的工作方式和编程,最后介绍了 8255A 的应用实例。

5.1 可编程并行接口芯片 8255A

微机系统中数据的存储和传输是以计算机的字长(如 8 位、16 位、32 位、64 位)为单位进行的。这种一次传送一个字长数据的方式即是并行传送方式。微机系统中 CPU 与存储器、CPU 与 I/O 接口、CPU 与磁盘之间的数据交换都是采用并行数据传送方式。并行传送方式的传送速率快,适用于微机之间近距离、大量和快速信息交换的场合。但并行传送时,信号之间容易产生干扰,所以不适合远距离传送。

并行接口电路在 CPU 与 I/O 设备之间传送数据,主要起着缓冲和锁存的功能。并行接口电路有多种,目前,微机系统设计中广泛使用的是可编程并行接口芯片。常用的可编程并行接口芯片有 Intel 公司的 8255A、Motorola 公司的 MC6820 等。

8255A 是 Intel 公司生产的 8 位并行 I/O 接口芯片。8255A 有 3 个 8 位并行 I/O 端口,可通过程序选择多种操作方式,通用性强,广泛应用于几乎所有系列的微机系统中。

5.1.1 8255A 的内部结构

8255A 的内部结构包括 3 个并行输入或输出数据端口、1 个控制端口、数据总线缓冲器、

读/写控制电路。8255A 的内部结构如图 5.1 所示。

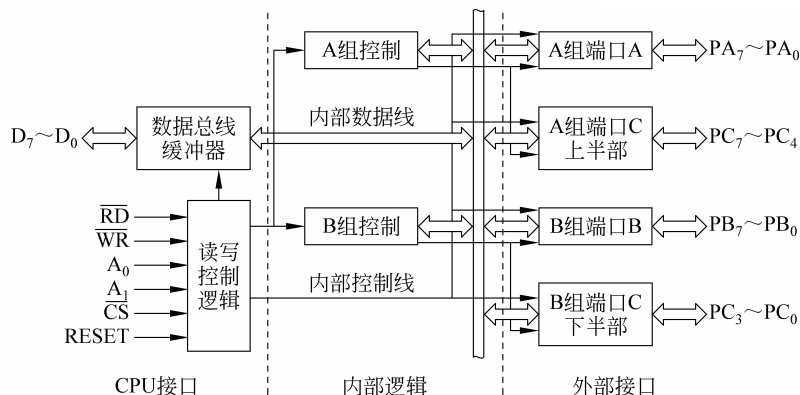


图 5.1 8255A 的内部结构

1. 并行输入或输出数据端口

8255A 内部有 3 个并行输入或输出数据端口,分别是 PA 口、PB 口、PC 口。3 个端口都是 8 位,可分别设置为输入或输出端口。这 3 个端口都可用作 CPU 与 I/O 设备之间的缓冲器或锁存器。

PA 端口: 包含一个 8 位数据输出锁存/缓冲器和一个 8 位数据输入锁存器。PA 端口中输入或输出的数据均能得到锁存。

PB 端口: 包含一个 8 位数据输出锁存/缓冲器和一个 8 位数据输入缓冲器。PB 端口中输入的数据不能锁存。

PC 端口: 包含一个 8 位数据输出锁存/缓冲器和一个 8 位数据输入缓冲器。PC 端口输入的数据不能锁存。PC 端口可以作为一个独立的 8 位输入或输出端口使用,也可分为两个独立的 4 位输入或输出端口使用。另外,PC 端口除了可以作为数据输入或输出端口外,还可以作为控制/状态端口,配合 PA 端口、PB 端口工作。PC 端口还可以由控制电路进行按位置位或复位的操作。

2. A 组和 B 组控制逻辑

A、B 两组控制电路内部有一个控制端口,用来接收数据总线送来的控制字和读/写控制信号,并根据控制字确定各端口的工作方式。A 组控制逻辑电路对 PA 端口和 PC 端口的高 4 位(PC₇~PC₄)进行控制。B 组控制逻辑电路对 PB 端口和 PC 端口的低 4 位(PC₃~PC₀)进行控制。

3. 数据总线缓冲器

双向、三态的 8 位数据总线缓冲器是 8255A 与 CPU 之间传输数据的通路。CPU 执行输出指令时,将控制字或数据通过数据总线缓冲器送入 8255A 内部的控制端口或数据端口。CPU 执行输入指令时,可将 8255A 内部数据端口内的数据通过数据总线缓冲器送入 CPU。

4. 读/写控制逻辑

读/写控制逻辑电路与 CPU 的控制信号和地址信号线相连,将这些信号转变为 8255A 内部的控制信号,完成片内端口译码、端口读/写操作等。

5.1.2 8255A 的外部引脚

8255A 是 40 引脚的双列直插集成电路芯片,芯片引脚排列如图 5.2 所示。电源引脚采用单一的+5V 电源供电。除了电源和地线引脚,另外的信号线分为与 CPU 相连的引脚和与 I/O 设备相连的引脚两类。

1. 与 I/O 设备相连的引脚

(1) $PA_7 \sim PA_0$: PA 端口与 I/O 设备相连的引脚,双向、三态数据线。

(2) $PB_7 \sim PB_0$: PB 端口与 I/O 设备相连的引脚,双向、三态数据线。

(3) $PC_7 \sim PC_0$: PC 端口与 I/O 设备相连的引脚,双向、三态数据线。

2. 与 CPU 相连的引脚

(1) $D_7 \sim D_0$: 与 CPU 系统数据总线相连引脚,双向、三态数据线。

(2) $A_1 A_0$: 端口地址选择线,输入。当 $A_1 A_0 = 00$ 时,选择 PA 端口;当 $A_1 A_0 = 01$ 时,选择 PB 端口;当 $A_1 A_0 = 10$ 时,选择 PC 端口;当 $A_1 A_0 = 11$ 时,选择控制端口。

(3) $\overline{CS}$: 片选信号,输入,低电平有效。 $\overline{CS}=0$ 时,表明 8255A 被选中,可以访问芯片进行操作。

(4) $\overline{RD}$: 读信号,输入,低电平有效。 $\overline{RD}=0$ 时,CPU 可以读取端口的数据。

(5) $\overline{WR}$: 写信号,输入,低电平有效。 $\overline{WR}=0$ 时,CPU 可以向端口写入信息。

(6) RESET: 复位信号,输入,高电平有效。当 $RESET=1$ 时,清除所有内部寄存器的内容,并将 PA、PB、PC 端口自动设置为方式 0,输入方式。RESET 信号可以与 CPU 的复位信号线相连,也可以分开单独设置。

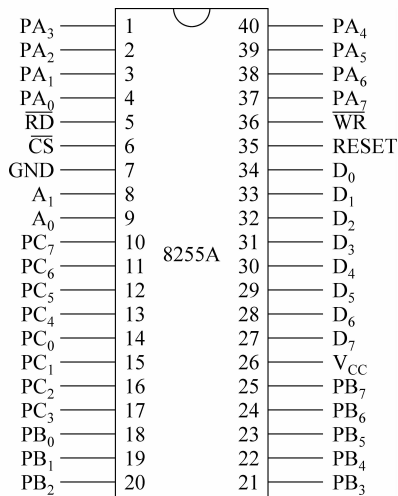


图 5.2 8255A 的引脚排列

5.1.3 8255A 的编程

CPU 通过输出指令向 8255A 的控制端口送入控制字。8255A 会根据控制字确定各端口的工作方式。8255A 有两个控制字,即方式选择控制字和 PC 端口置位/复位控制字。如果控制端口内的控制字 $D_7 = 1$,表示方式选择控制字; $D_7 = 0$,表示 PC 端口置位/复位控制字。

1. 方式选择控制字

8255A 有 3 种工作方式,即方式 0、方式 1 和方式 2。PA 端口可以工作于方式 2、方式 1 和方式 0;PB 端口可以工作于方式 1 和方式 0;PC 端口只能工作于方式 0。这些方式的操作详情将在 5.1.4 小节讲解。

方式选择控制字用于设置各端口的工作方式。将 8255A 的方式控制字传送到控制端口,完成 8255A 的初始化工作,才可以对 8255A 的数据端口进行访问。方式选择控制字格式如图 5.3 所示。

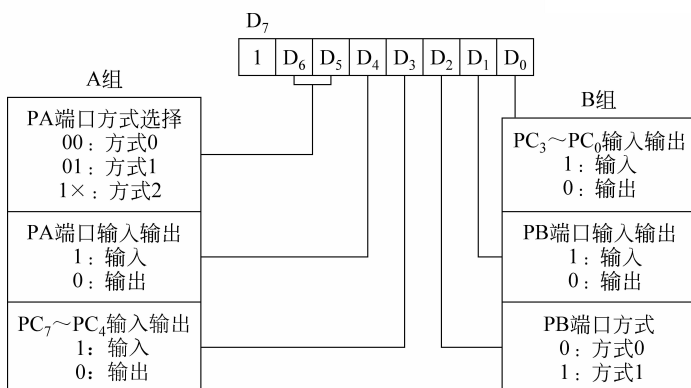


图 5.3 方式选择控制字格式

D₇: 1,特征位。D₇=1,表示写入控制端口的是方式选择控制字。

D₆D₅: PA 端口的方式选择。D₆D₅=00,PA 端口工作在方式 0;D₆D₅=01,PA 端口工作在方式 1;D₆D₅=10 或 11,PA 端口工作在方式 2。

D₄: PA 端口的 I/O 方式选择。D₄=0,PA 端口为输出端口;D₄=1,PA 端口为输入端口。

D₃: PC 端口高 4 位(PC₇~PC₄)的 I/O 方式选择。D₃=0,PC 端口高 4 位为输出端口;D₃=1,PC 端口高 4 位为输入端口。

D₂: PB 端口方式选择。D₂=0,PB 端口工作在方式 0;D₂=1,PB 端口工作在方式 1。

D₁: PB 端口的 I/O 方式选择。D₁=0,PB 端口为输出端口;D₁=1,PB 端口为输入端口。

D₀: PC 端口低 4 位(PC₃~PC₀)的 I/O 方式选择。D₀=0,PC 端口低 4 位为输出端口;D₀=1,PC 端口低 4 位为输入端口。

例 5-1 编程实现对 8255A 的工作方式进行设定。已知 8255A 的端口地址为 2A0H~2A6H。采用偶地址连接(即 CPU 的 A₀地址线为 0,从 A₁地址线开始分配端口选择线)。要求 PA 端口工作于方式 1,为输入端口;PB 端口工作于方式 0,为输出端口;PC 端口高 4 位(PC₇~PC₄)为 4 位输出端口,PC 端口低 4 位(PC₃~PC₀)为输入端口。

解 首先分析控制端口地址。因为是偶地址连接,所以,CPU 地址线 A₀悬空未接

8255A, 而用 A_2A_1 连接到 8255A 的端口选择引脚 A_1A_0 。8255A 的端口地址为 $2A0H \sim 2A6H$, 则 CPU 地址线 $A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0 = 1010100000 \sim 1010100110$ 中的偶地址时, 选择 8255A 的某个端口操作。当 CPU 的 A_2A_1 为 11 时, 8255A 的端口选择引脚 A_1A_0 为 11, 即选中 8255A 的控制端口。所以, 8255A 的控制端口地址为 $2A6H$ 。

程序段如下。

```
MOV DX, 2A6H           ;送控制端口地址至 DX
MOV AL, 10110001B      ;AL 中准备方式选择控制字的值
OUT DX, AL             ;AL 数据送到控制端口, 因为最高位为 0,
                        ;所以作为方式选择控制字
```

2. PC 端口置位/复位控制字

PC 端口置位/复位控制字用于对 PC 端口中的任一位单独设置为 1 或 0, 而不影响其他位的值。PC 端口置位/复位控制字格式如图 5.4 所示。

D_7 : 0, 特征位。 $D_7 = 0$, 表明写入控制端口的是 PC 端口置位/复位控制字。

$D_6D_5D_4$: 未使用, 可为任意值。

$D_3D_2D_1$: 选择对 PC 端口的哪一位进行操作。 $D_3D_2D_1$ 的组合表示要操作位的编码。

D_0 : 选择对 PC 端口指定的位是做置位操作还是复位操作。 $D_0 = 1$, PC 端口指定的位设置为 1; $D_0 = 0$, PC 端口指定的位设置为 0。

需要特别注意的是, 尽管该控制字是对 PC 端口进行操作, 但必须写入控制端口, 而不是写入 PC 端口。

例 5-2 已知系统中 8255A 的端口地址范围为 $4A0H \sim 4A3H$ 。利用 PC 端口置位/复位控制字, 编程实现在 8255A 的 PC_3 引脚上输出方波信号。

解 在一般连接方式下, CPU 的低位地址线 A_1A_0 接 8255A 的端口选择线 A_1A_0 。8255A 的端口地址为 $4A0H \sim 4A3H$, 则 CPU 地址线 $A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0 = 10010100000 \sim 10010100011$ 时, 选择 8255A 的某个端口操作。当 CPU 的 A_1A_0 为 11 时, 8255A 的端口选择引脚 A_1A_0 为 11, 即选中 8255A 的控制端口。所以, 8255A 的控制端口地址为 $4A3H$ 。

程序如下。

```
CODE SEGMENT
    ASSUME CS:CODE           ;没有数据段, 不需要设置数据段地址
START:
    MOV DX, 4A3H            ;送控制端口地址至 DX
```

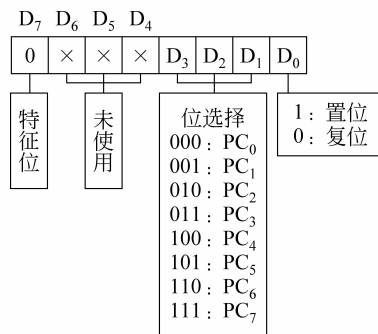


图 5.4 8255A 的 PC 端口置位/复位控制字

```

LL:  MOV    AL,00000111B    ;AL 中数据最高位为 0
      OUT    DX,AL          ;AL 中数据送控制端口,所以为控制字
                                ;因为控制字最高位为 0,是 PC 端口置位/复位控制字
                                ;完成对 PC3 置 1 操作,其他位不变

      MOV    CX,0FFFFH

L1:   LOOP   L1              ;CX 自减到 0,实现延时,维持方波的高电平
      MOV    AL,00000110B    ;AL 中数据最高位为 0
      OUT    DX,AL          ;AL 中数据送控制端口,所以为控制字
                                ;因为控制字最高位为 0,是 PC 端口置位/复位控制字
                                ;完成对 PC3 置 0 操作,其他位不变

      MOV    CX,0FFFFH

L2:   LOOP   L2              ;延时,方波的低电平维持一段时间
      JMP    LL              ;循环,产生周期性的方波信号

CODE  ENDS                  ;系统中没有 DOS 系统,不用 DOS 调用结束程序
      END    START

```

3. 8255A 数据端口访问

向 8255A 的控制端口写入方式选择控制字以后,8255A 的初始化便完成,CPU 就可以访问 PA、PB、PC 数据端口,与这些端口进行数据交换了。

例 5-3 已知系统中 8255A 的端口地址范围为 4A0H~4A3H。采用 CPU 向 PC 端口中输出数据的方式,改变 PC₃ 引脚的输出值,从而产生方波信号。

解 在一般连接方式下,CPU 的低位地址线 $A_1 A_0$ 接 8255A 的端口选择线 $A_1 A_0$ 。8255A 的端口地址为 4A0H~4A3H,则 CPU 地址线 $A_9 A_8 A_7 A_6 A_5 A_4 A_3 A_2 A_1 A_0 = 10010100000 \sim 10010100011$ 时,选择 8255A 的某个端口操作。当 CPU 的 $A_1 A_0$ 为 11 时,8255A 的端口选择引脚 $A_1 A_0$ 为 11,则选中 8255A 的控制端口。所以,8255A 的控制端口地址为 4A3H。当 CPU 的 $A_1 A_0$ 为 10 时,8255A 的端口选择引脚 $A_1 A_0$ 为 10,则选中 8255A 的 PC 端口。所以,8255A 的 PC 端口地址为 4A2H。

要往 PC 端口送数,必须先将方式控制字写入控制端口,完成初始化。本例中没有用 PA、PB 端口,所以方式控制字中与 PA、PB 端口有关的设置都无效,一般写 0。PC 端口用作输出端口,以便输出方波信号。所以方式控制字中 PC 端口设置要选择为输出。

对 PC 端口传送数据时,必须一次传送 8 位数据。

程序如下。

```

CODE  SEGMENT
      ASSUME  CS:CODE        ;没有数据段,不需要设置数据段地址

START:
      MOV     DX,4A3H        ;控制端口地址送 DX
      MOV     AL,10000000B    ;方式选择控制字:最高位为 1,PC 端口设为输出
      OUT     DX,AL          ;方式选择控制字送控制端口,初始化完成

LL:    MOV     AL,00001000B    ;送 PC 端口的 8 位数据,其中  $D_3=1$ 

```

```

MOV    DX, 4A2H          ;PC 端口地址送 DX
OUT    DX, AL            ;PC 端口中 PC3=1,同时其他位被置为 0
MOV    CX, 0FFFFH
L1:    LOOP L1            ;延时,方波的高电平维持一段时间
MOV    AL, 00000000B      ;送 PC 端口的 8 位数据,其中 D3=0
MOV    DX, 4A2H          ;PC 端口地址送 DX
OUT    DX, AL            ;PC 端口中 PC3=0,同时其他位被置为 0
MOV    CX, 0FFFFH
L2:    LOOP L2            ;延时,方波的低电平维持一段时间
JMP    LL                ;循环,产生周期性的方波信号
CODE   ENDS              ;系统中没有 DOS 系统,不能用 DOS 调用结束
END     START

```

对比例 5-2 和例 5-3,虽然都是在 PC₃引脚信号线上产生方波,但是用 PC 端口置位/复位命令字实现时,只会改变指定的 PC₃位输出值,其他位不受影响;但是用写数据到 PC 端口的的方法,则会将 PC 端口中的 8 位都置入新数据,PC₃以外的其他位也被改变了。为了保护其他位不受影响,可以将 PC 端口中原有数据读出,和 00001000B 做或运算,再送到 PC 端口中。

5.1.4 8255A 的工作方式

8255A 有 3 种工作方式:方式 0 是基本 I/O 方式;方式 1 是选通 I/O 方式;方式 2 是双向 I/O 方式。

1. 方式 0: 基本 I/O 方式

在基本 I/O 方式中,PA 端口、PB 端口、PC 端口的高 4 位和 PC 端口的低 4 位都是独立的数据输入或输出端口。方式 0 一般可用于采用无条件或查询方式传送数据的场合。无条件传送时,CPU 直接与 8255A 的 3 个端口进行数据交换。采用查询方式传送时,可以用一个端口作数据端口,用另外的端口存放 I/O 设备的状态信息。CPU 先读取状态信息,进行条件判断,在条件状态允许时,再对数据端口操作。

例 5-4 在某系统中,需要将内存中的 100 个字节数据传送到 8 个 LED 发光二极管构成的显示电路上显示。设计系统的硬件电路和程序。

解 先进行硬件设计。要将字节数据传送到 8 个 LED 发光二极管显示电路,数据是多位传输的,可以选用 8255A 并行接口作为 CPU 与显示电路的接口。选择 8255A 的 PA 端口作为输出数据的锁存,所以将显示电路连接到 PA 端口的引脚上。CPU 与 8255A 采用一般的连接方式即可。端口地址只要选择系统中不冲突的地址范围即可。设计硬件电路逻辑示意如图 5.5 所示。

再进行软件设计。首先分析端口地址译码电路,确定 8255A 各端口的地址。端口译码电路采用固定式门电路译码方式。CPU 地址线 A₉ A₈ A₇ A₆ A₅ A₄ A₃ A₂ 必须为 10110000B,才可以使 8255A 的 $\overline{\text{CS}}$ 信号有效。所以可知,系统中 8255A 的端口地址范围为 2C0H~

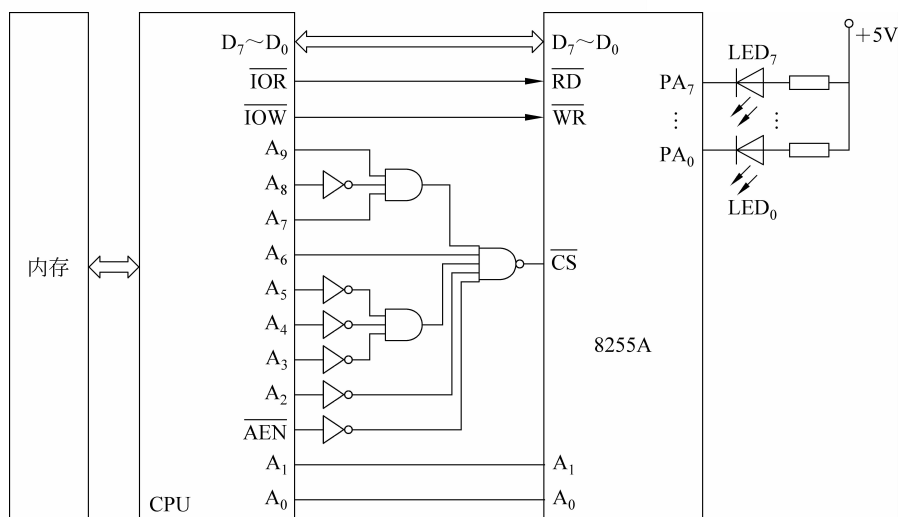


图 5.5 例 5-4 系统硬件电路逻辑示意

2C3H。CPU 的 $A_1 A_0$ 直接连到 8255A 的端口选择线 $A_1 A_0$ ，则端口地址中 $A_1 A_0$ 为 00 时，是 PA 端口的地址， $A_1 A_0$ 为 11 时，是控制端口的地址。程序采用循环结构，无条件传送方式。程序的流程是：先对 8255A 初始化，再循环从内存读取数据传送到 PA 端口。

程序如下。

```
DATA SEGMENT
    BUFF DB 100 DUP(?) ;数据区定义
DATA ENDS
CODE SEGMENT
    ASSUME CS:CODE,DS:DATA
START:
    MOV AX,DATA ;设数据段地址
    MOV DS,AX
    LEA BX,BUFF ;数据区首地址送 BX
    MOV CX,100 ;循环次数
    MOV AL,10000000B ;方式控制字,PA 端口输出
    MOV DX,2C3H ;控制端口地址
    OUT DX,AL ;方式控制字送控制端口
L1: MOV AL,[BX] ;数据区取数到 AL
    MOV DX,2C0H ;PA 端口地址送 DX
    OUT DX,AL ;数据输出至 PA 端口
    INC BX ;修改数据区地址指针
    PUSH CX ;CX 入栈保护
    MOV CX,0FFFFH
```



```

L2:   LOOP   L2           ;延时,使显示电路数据显示稳定
      POP    CX           ;CX 出栈
      LOOP   L1           ;循环次数判断
CODE  ENDS               ;系统中没有 DOS 系统,不能用 DOS 调用结束
      END    START

```

例 5-5 设计一个系统,使开关 K 合上时,8 个 LED 灯依次循环点亮;开关 K 断开时,8 个 LED 灯全灭。

解 先进行硬件设计。选用 8255A 并行接口,作为 CPU 与开关 K、显示电路的接口。选择 8255A 的 PB 端口作为输出数据的锁存,所以将显示电路连接到 PB 端口的引脚上。选择 PA₇ 引脚作为开关 K 的输入信号线,开关 K 输入的值会在 PA 端口的 D₇ 位保存。CPU 与 8255A 的连接,采用一般的连接方式即可。端口地址只要选择系统中不冲突的地址范围即可。设计硬件电路逻辑示意如图 5.6 所示。

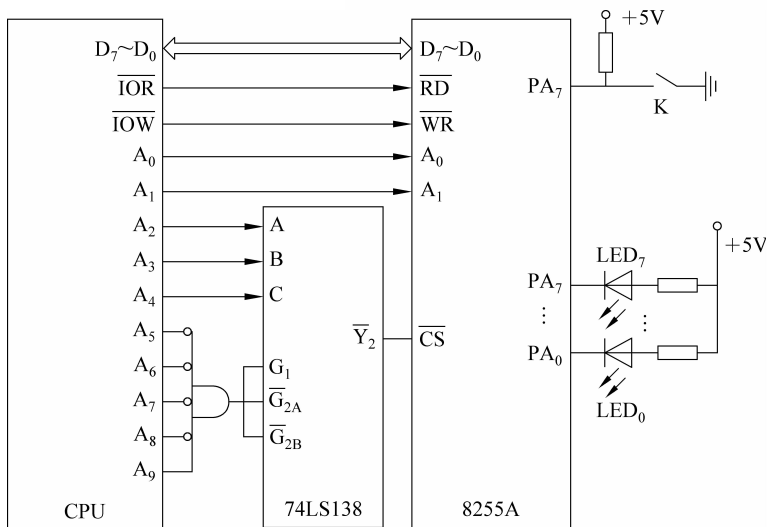


图 5.6 例 5-5 系统硬件电路逻辑示意

再进行软件设计。首先分析端口地址译码电路,确定 8255A 各端口的地址。端口译码电路采用译码器译码方式。CPU 地址线 A₉ A₈ A₇ A₆ A₅ A₄ A₃ A₂ 必须为 10000010B,才可以使 8255A 的 $\overline{CS}$ 信号有效。所以可知,系统中 8255A 端口地址范围是 208H ~ 20BH (10000010XXB)。CPU 的 A₁ A₀ 直接连到 8255A 的端口选择线 A₁ A₀,则端口地址中最低两位为 00 时,是 PA 端口的地址;最低两位为 01 时,是 PB 端口的地址;最低两位为 11 时,是控制端口的地址。

根据逻辑电路图可知,在 PB 端口的某引脚输出 0,则对应连接的 LED 灯亮;在 PB 端口的某引脚输出 1,则对应连接的 LED 灯不亮。所以要实现 LED 灯的亮灭控制,是由 CPU 向 PB 端口内写入不同数据实现的。

PA 端口是 8 位的锁存/缓冲器,只能进行 8 位数据的读出和写入。在系统逻辑电路图中,PA₆~PA₀未接信号,则这些引脚上的数据是高阻态。在判断 PA₇的值时,需要将其他位屏蔽。由输入开关的连接电路可知,K 合上时,PA₇输入 0 信号;K 断开时,PA₇输入 1 信号。

程序采用循环结构,查询传送方式。程序流程图如图 5.7 所示。

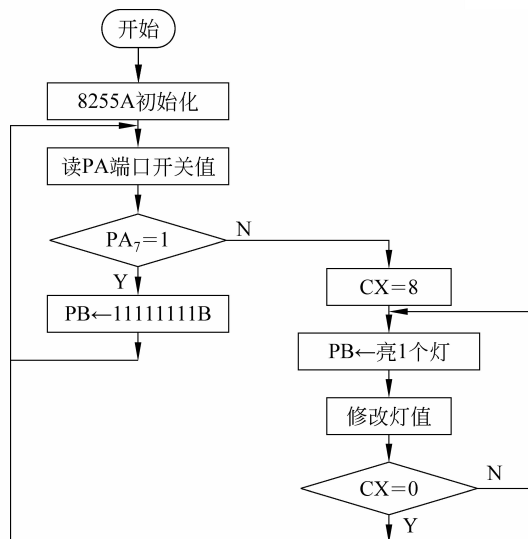


图 5.7 例 5-5 程序流程

程序如下。

```

CODE SEGMENT
    ASSUME CS:CODE

START:
    MOV     AL,10010000B    ;方式控制字,PA 方式 0 输入,PB 方式 0 输出
    MOV     DX,20BH        ;控制端口地址
    OUT     DX,AL          ;方式控制字送入控制端口
L:      MOV     DX,208H     ;PA 端口地址
    IN      AL,DX          ;读 PA 端口值到 AL
    TEST    AL,10000000B   ;判断 PA7 所接开关状态,屏蔽其他位
    JNZ     L1             ;开关断开转 L1
    MOV     CX,8           ;开关闭合,依次点亮 8 个 LED 灯
    MOV     AL,11111110B   ;点亮 PB0 所接 LED 灯,PB 端口中放 8 位值
L2:     MOV     DX,209H     ;PB 端口地址
    OUT     DX,AL
    PUSH    CX             ;循环次数入栈保护
    MOV     CX,0FFFFH
L1:     LOOP    L1         ;延时,保持 PB 口数据不变,LED 灯显示稳定
    
```