

MCS-51 单片机的系统扩展技术（四）

4.2 利用 8255A 可编程并行接口芯片扩展 I/O 口

8255A 是 INTEL 公司生产的一种通用可编程并行 I/O 接口芯片。它有 3 个并行端口，分别称为 PA、PB、PC 口，其中 PC 口又分为高 4 位口和低 4 位口两部份。它们都可以通过软件编程来设置各 I/O 口的工作方式。

一、8255A 的内部结构和引脚功能

图 17 是 8255A 的内部结构框图，图 18 是 8255A 的引脚排图，该芯片主要由以下几部份组成：

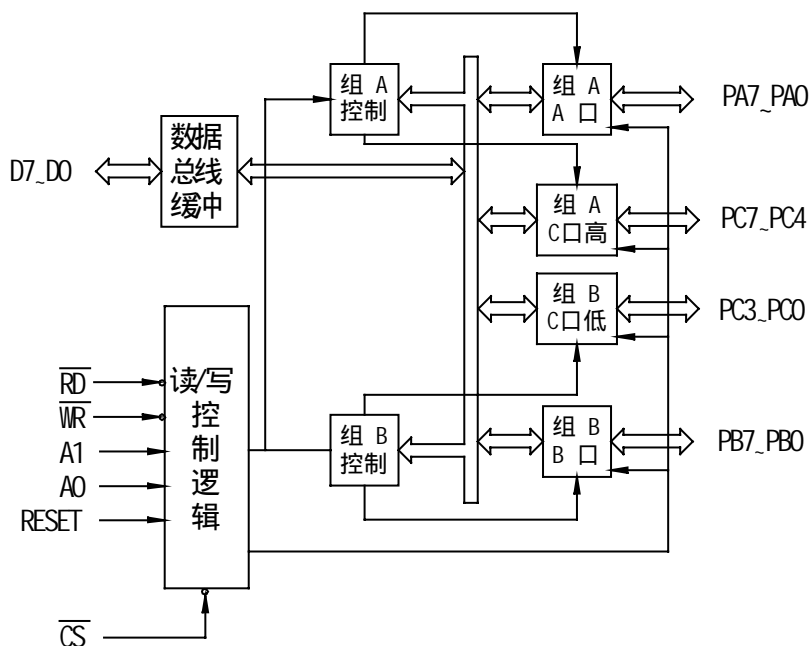


图 17 8255 的内部结构框图

(1) 并行端口 PA、PB、PC 这三个端口都为 8 位，都可被编程为输入或输出两种方式，但它们在结构和功能上有差异。PA 口有一个 8 位数据输出锁存器/缓冲器和一个 8 位数据输入锁存器，可编程为输入/输出或双向寄存器；PB 口有一个 8 位输入/输出锁存/缓冲器和一个 8 位数据输入缓冲器（不锁存），可编程为输入或输出，但不能双向输入/输出；PC 口有一个 8 位数据输出锁存/缓冲器和一个 8 位数据输入缓冲器，可分为两个 4 位口使用。它除了作为输入输出外，还可作为 PA、PB 口工作于选通方式时的状态控制信号。

(2) 总线接口电路 该电路主要用于实现 8255A 与单片机芯片的信号连接。它由两部分组成：

数据总线缓冲器 数据总线缓冲器为 8 位双向三态缓冲器，可以直接与 MCS-51 系统总线相连。MCS-51 进行 I/O 操作的有关数据、控制字和状态信息都是过该缓冲器进行传送。

读/写控制逻辑 这部份主要是与读写有关的控制信号，主要包括：

\overline{CS} ：片选输入信号，低电平有效（8255A 被选中）

\overline{RD} ：读信号，输入，低电平有效（允许 CPU 从 8255A 读取数据或状态信息）

\overline{WR} ：写信号，输入，低电平有效（允许 CPU 将控制字或数据写入 8255A）

RESET：复位信号，输入，高电平有效（8255A 被复位，所有控制寄存器被清 0，所有端口被置输入方式）；

A1、A0：端口选择信号，输入。8255A 的 PA、PB、PC 和一个控制寄存器，共有四个端口，根据 A1A0 输入的地址信号来进行寻址。在利用 8255A 进行 MCS-51 的 I/O 扩展时，应将 A1A0 和 MCS-51 的 P0.1 和 P0.0 相连。

读/写控制逻辑主要用于实现对 8255A 的硬件管理，其主要功能包括芯片选择、端口寻址及规定各端口和单片机之间的数据传送方向等，具体见表 3。

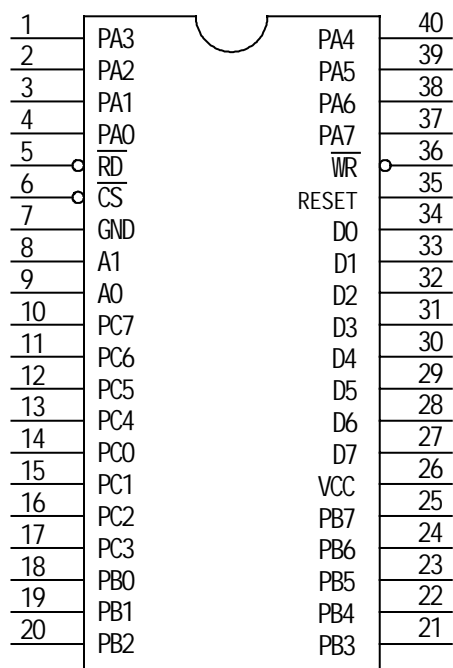


图 18 8255 引脚图

表 3 8255 各端口组合功能表

\overline{CS}	\overline{RD}	\overline{WR}	A ₁	A ₀	通道选择与操作功能
0	0	1	0	0	PA 口→数据总线
0	0	1	0	1	PB 口→数据总线
0	0	1	1	0	PC 口→数据总线
0	1	0	0	0	数据总线→PA 口
0	1	0	0	1	数据总线→PB 口
0	1	0	1	0	数据总线→PC 口
0	1	0	1	1	数据总线→控制寄存器
1	×	×	×	×	数据总线呈高阻态
0	0	1	1	1	非法条件
0	1	1	×	×	数据总线呈高阻态

(3) A、B 组控制电路 A、B 组控制电路包括 A 组控制和 B 组控制，合在一起构成 8 位控制寄存器，用于存放各端口的工作方式控制字。

8255A 的 40 个引脚中，除上述已提及的引脚外，还包括以下一些引脚：

GND：接地；

VCC：+5V 电源；

D0-D7：双向 8 三态 8 位数据线；

PA0-PA7：A 口 8 位双向数据线；

PB0-PB7：B 口 8 位双向数据线；

PC0-PC7：C 口 8 位双向数据线。当 8255A 工作于方式 0 时，PC0-PC7 分为两组（每组

4 位) 并行 I/O 数据线。当 8255A 工作于方式 1 或 2 时, PC0-PC7 为 PA、PB 口提供联络和中断信号, 这时 PC 口每根口线的的功能定义见表 4。

表 4 工作方式 1 和工作方式 2 时 PC 口每根线的功能

C 口位线	方式 1		方式 2	
	输入	输出	输入	输出
PC7		\overline{OBFA}		\overline{OBFA}
PC6		\overline{ACKA}		\overline{ACKA}
PC5	IBFA		IBFA	
PC4	\overline{STBA}		\overline{STBA}	
PC3	INTRA	INTRA	INTRA	INTRA
PC2	\overline{STBB}	\overline{ACKB}		
PC1	IBFB	\overline{OBFB}		
PC0	INTRB	INTRB		

二、8255A 的工作方式及数据 I/O 操作

(1) 8255A 的工作方式 8255A 共有三种工作方式, 即方式 0、方式 1 和方式 2。

方式 0 (基本输入/输出方式) 在这种工作方式下, PA、PB 口及 PC 口的两个 4 位口中的任何一个端口都可以被编程设定为输入或输出方式, 但不能既作输入又作输出。在作为输入时, 输入数据不被锁存, 而作为输出时, 数据被锁存。在方式 0 时, 不需要任何选通信号或联络信号。方式 0 适用于数据的无条件传送, 也可以人为指定某些位作为状态信息线, 进行查询式传送。

方式 1 (选通输入/输出) 工作于方式 1 时, PA 和 PB 口分别用于数据的输入/输出, PC 口可以确定附加某些位作为 PA 和 PB 口的联络信号, 以便 8255A 和外设之间, 或者 8255A 与 CPU 之间传送状态信息以及中断请求信号。在方式 1 中, PA、PB 口的数据输入输出都具有锁存功能。

若 PA 口和 PB 口都工作于方式 1, 则 PC 口中有 6 位固定作为 PA 口和 PB 口的状态和控制信号, PC 口剩下的 2 位可以编程为输入或输出。若 PA 口、PB 口中有 1 个工作于方式 1, 另一个工作于方式 0, 则 PC 口中有 3 位固定作为 PA 口或 PB 口的状态和控制信号, PC 口剩下的 5 位可以编程为输入或输出。方式 1 适用于查询中断方式的数据输入/输出。

方式 2 (双向数据传送方式) 只有 PA 口才能选择这种方式。此时, PA 口既能输入数据又能输出数据, PC 口的 PC3-PC7 用作 PA 口的输入/输出同步控制信号。在方式 2 下, PC 口剩下的 3 位可编程为 I/O 线, 而 PB 口可编程为方式 0 或方式 1 工作。方式 2 适用于查询或中断方式的双向数据数据传送。

(2) 数据的输入操作 用于数据输入操作的联络信号 从表中可能看出 PC 口中用于数据输入操作的联络信号有:

\overline{STB} : 选通脉冲输入、低电平有效。外设送来的 \overline{STB} 下降沿将端口数据线上的输入数据状态入端口锁存器;

IBF: 输出缓冲器满信号输出, 高电平有效。在此信号有效时, 表示外设已将数据装入端口锁存器, 但 CPU 尚未读取。在 CPU 读取端口数据后, IBF 将变为低电平, 表示端口锁存器空。

INTR: 中断请求信号, 高电平有效。在 IBF 为高电平, \overline{STB} 信号由低变高时, 中断请求信号有效, 向 MCS-51 发出中断请求。

INTE: 8255A 端口内部的中断允许触发器。只有在 INTE 为高电平时才允许端口中断请求。INTEA 和 INTEB 分别由 PC4、PC2 的置位/复位控制。

数据输入操作过程 当外设准备好输入数据后，发出 $STB=0$ 信号，输入的数据进入 8255A 的端口缓冲器，并使 $IBF=1$ 。如 MCS-51 采用查询的方式，此时 CPU 可以查询 IBF 这一状态，以决定是否可以向输入数据。如采用中断方式，当 \overline{STB} 信号由低变高并且 $INTE$ 为高时， $INTR$ 有效，向 CPU 发出中断请求信号（必须加反相将此信号变为低有效），MCS-51 响应中断后，在执行中断服务程序时读入数据，并使 $INTR$ 信号变为低电平（无效），同时也使 IBF 信号变低，用以通知外设再次输入数据。

（3）数据的输出操作

用于数据输出操作的联络信号 从表中可以看出 8255A 的 PC 口中用于数据输出操作的联络信号有：

\overline{ACK} ：外设响应输入信号，低电平有效。它是外设取走并且处理完 8255A 的数据后，向单片机发出的响应信号。

\overline{OBF} ：输出，低电平有效，输出缓冲器满信号。当 CPU 把输出数据写入 8255A 锁存器后，该信号有效，用来启动外设开始接收数据。外设从端口取走数据后发回来的回答信号使 \overline{OBF} 变为高电平。

$INTR$ ：输出，高电平有效，中断请求信号。

$INTE$ ：端口内部的中断允许触发器。 $INTE$ 为高电平时才允许端口提出中断请求。 $INTEA$ 和 $INTEB$ 分别由 $PC6$ 和 $PC2$ 的置位/复位进行控制。

数据输出操作过程 当外设接收并处理完一组数据（如打印完毕）后，发出 \overline{ACK} 信号使 \overline{OBF} 变高，表示输出缓冲器已空（实际上表示缓冲器中的数据不必再保留）。如采用查询方式，则 \overline{OBF} 可作为状态信号供 CPU 查询使用；如使用中断方式，则在 \overline{ACK} 信号变高且 $INTE$ 信号有效时，使 $INTR$ 有效，向单片机发出中断请求信号。在中断服务过程中，CPU 将下一组数据写到 8255A 的输出缓冲器，并使 \overline{OBF} 有效，表示数据已送到。利用这一信号启动外设开始工作，取走并处理 8255A 中的输入数据，再给出 \overline{ACK} 信号。

三、8255A 的控制字

8255A 有两种控制字，即控制 PA、PB 和 PC 口工作方式的方式控制字；专门用于控制 PC 口各位置位/复位的控制字。这两种控制字都是写到控制寄存器中去，利用控制字的最高位来进行区别。

（1）方式控制字 图 5-19 是 8255A 的工作方式控制字的格式，方式控制字用于确定各口的工作方式及数据传的方向，其特征为最高位为 1。

在确定方式控制字时，应注意以下两点：

A 口有三种工作方式（方式 0、方式 1、方式 2），而 B 口只有两种工作方式（方式 0 和方式 1）。

在方式 1 和方式 2 下，对 PC 口的定义（输入或输出）不影响作为联络线使用的 PC 口各位的功能。例如，将 PA 口设置为方式 0 输入，PB 口设置为方式 1 输出，PC4-PC7 为输出，PC0-PC3 为输入。根据此工作要求，对照方式控制字的格式及各位的功能，可写出方式控制字为 10010101B，即 95H，然后，将 95H 写入控制寄存器即可。

（2）PC 口位控制字 在某些实际应用情况下，PC 口被用来定义控制信号和状态信号（如对 PC 口内的中断允许触发器的开中断或关中断的设置）。因此，可利用 PC 口所具有的位操作功能和 PC 口位置位/复位控制字，对 PC 口的每一位进行置位或复位（即将某一位置 1 或清 0）。图 20 给出了 PC 口位控制字的格式及其功能，PC 口位控制字的特征位为 0。

四、8255A 与 MCS-51 单片机的连接

图 21 是 8255A 与 MCS-51 单片机连接的电路图。从图中可以看出，8255A 的总线与单片机的总线直接相连， \overline{WR} 和 \overline{RD} 分别与单片机的 \overline{WR} 和 \overline{RD} 相连。

(1) 8255A 各端口地址的确定 8255A 的 PA、PB、PC 口和控制字寄存器的地址不仅和 A1、A0 有关，而且和 8255A 的片选有关。只有在 MCS-51 选通 8255A 时，才能对寄存器和端口进行操作。根据上述原则，可以确定出如图 21 所示 8255A 各端口的地址如表 5 所示

(2) 8255A 的初始化编程 在利用 8255A 对 MCS-51 进行 I/O 扩展时，要先对 8255A 进行初始化。所谓初始化，就是根据需要，将相应的控制字写入 8255A 的寄

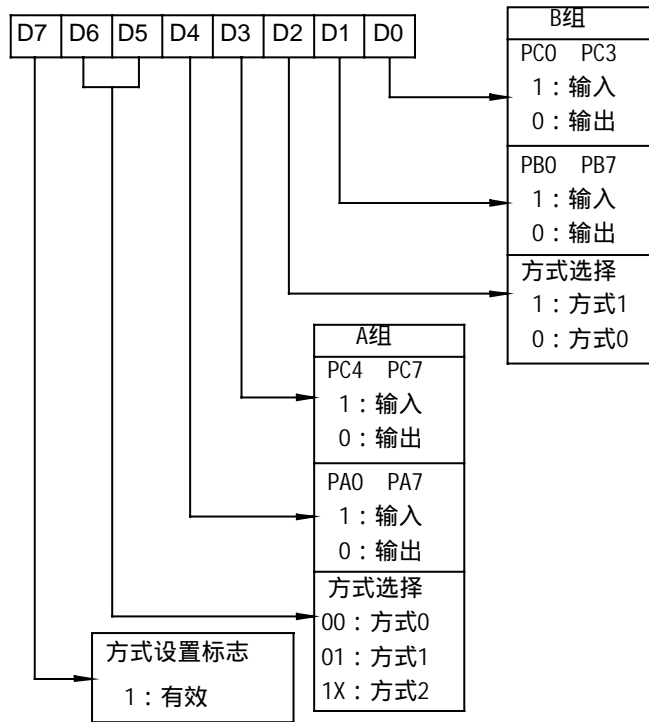


图 19 8255A 工作方式控制字

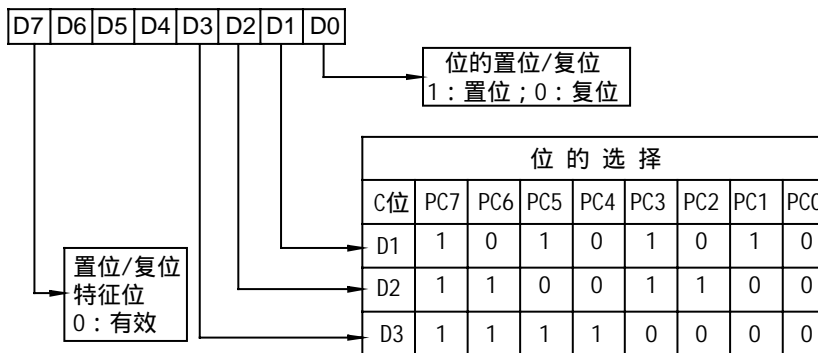


图 20 8255A 复位控制字

寄存器中。

例如，若对 8255A 各口作如下设置：PA 口方式 0 输入，PB 口方式 1 输出，PC 口高 4 位输出，低 4 位输入，设控制寄存器地址为 DFFFH。按各口的工作要求，工作方式控制字为 10010101B 即 95H，则初始化编程为：

```

MOV     DPTR, #0DFFFH
MOV     A, #95H
MOVX   @DPTR,A
    
```

表 5 图 21 中端口地址的确定

端口	P2	P0	端口地址
PA	××0×××××	××××××00	DFFCH
PB	××0×××××	××××××01	DFFDH
PC	××0×××××	××××××10	DFFE H
控制字寄存器	××0×××××	××××××11	DFFFH

注：按一般习惯将所有的×取 1 即得到了端口地址

五、8255A 的简单应用

对于图 21 的电路，将 PA 口设置为输出，PB 口设置为输入，均工作于方式 0，由此可以写出控制字为：1000×01×，如果将×全取 1，则控制字就是 10001011 即 8BH。

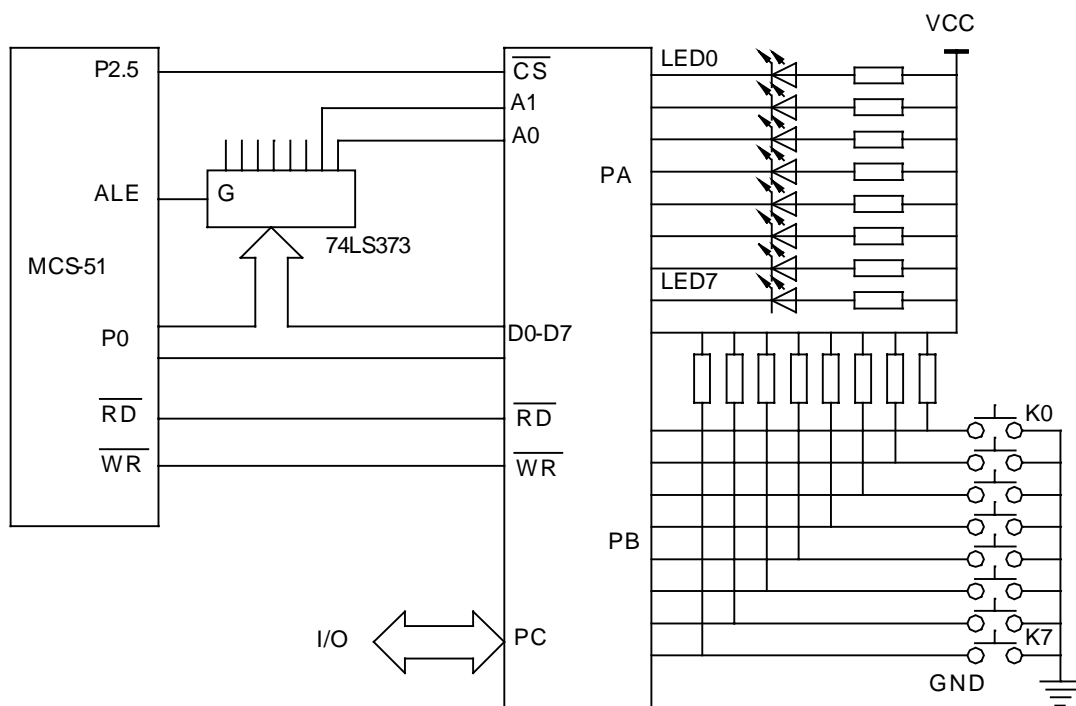


图 21 8255A 应用例

如果要实现流水灯，可以这样来做：

```

ORG    0000H
JMP    START
START: MOV    SP, #5FH        ; 堆栈初始化
        MOV    A, #8BH        ; 方式控制字
        MOV    DPTR, #0DFFFH ; 控制字端口的地址
        MOVX   @DPTR, A       ; 设置工作方式
        MOV    A, #80H
        MOV    DPTR, #0DFFCH ; 端口 A 的地址
LOOP:  MOVX   @DPTR, A
        CALL   DELAY          ; 调用延时子程序
        RL    A                ; 左移
        JMP    LOOP           ; 循环
DELAY: .....                 ; 与第二程序同，不再重复
    
```

其中 CALL 不是 MCS-51 的指令,最终会由汇编程序根据实际情况决定使用 ACALL 或 LCALL 指令。

如果上面的程序要实现按某键而灯亮,可以这样做:

```
ORG    0000H
JMP    START
START : MOV    SP, #5FH
        MOV    A, #8BH
        MOV    DPTR, #0DFFFH
        MOVX   @DPTR, A
LOOP :  MOV    DPTR, #0DFFDH ;端口 B 的地址
        MOVX   A, @DPTR     ;读 B 口的数据
        MOV    DPTR, #0DFFCH ;端口 A 的地址
        MOVX   @DPTR, A
        JMP    LOOP
```