

第 5 章 实用接口技术



计算机系统通过一系列的外围芯片辅助 CPU 处理器完成工作。这些芯片分别承担着不同的任务。主要芯片有可编程中断控制器 8259A、可编程定时/计数器控制接口芯片 8253/8254、可编程输入/输出并行接口芯片 8255A、支持异步串行通信标准的可编程接口芯片 8250 以及数模和模数 (A/D 和 D/A) 转换器件。在计算机系统中以上芯片被集成到超大规模集成电路芯片中。对上述芯片的使用也就构成了相应的接口技术。

本章着重从实用的方面，详细地讲述各相关接口技术的具体内容。分别从各接口芯片的内部结构、外部引脚，工作方式，与 CPU 处理器的连接，接口芯片的使用等方面进行详细的讲述。本章是全书的重点。



- 8259A 的结构、工作方式、写初始化字和控制字
- 8253/8254 的结构、工作方式、写初始化字和控制字
- 8255A 的结构、工作方式、写初始化字和控制字
- 16550 的结构、工作方式、写初始化字和控制字
- D/A 和 A/D 转换器的工作原理、工作方式
- 综合应用各接口技术进行硬件电路设计和软件程序开发

5.1 中断接口技术

在上一章中，已经就中断的基本概念及相关知识作了简单介绍，本节主要讲解中断控制的芯片。

可编程中断控制器 8259A 是 Intel 公司专为 8086/80x86 CPU 控制优先级中断而设计开发的芯片。8259A 中断控制芯片集中了中断源优先级排队、中断源识别、中断向量提供和中断屏蔽等功能，因此中断系统无需附加任何电路，通过对 8259A 进行编程，可以实现管理 8 级优先级中断，设定中断请求方式和优先级模式。可以将多片 8259A 级联，组成两级主从式中断控制系统，最多构成 64 级优先级中断管理系统。

8259A 是一种广泛使用的可编程中断控制器，下面通过 8259A 讲述中断控制器的结构与寄存器的编程。

5.1.1 8259A 的内部结构和外部引脚

1. 8259A 的内部结构

由图 5-1 所示 8259A 内部结构框图可知, 它由数据总线缓冲器、读/写控制逻辑、级联缓冲/比较器、IMR 中断屏蔽寄存器、ISR 中断服务寄存器、PR 中断优先级判别器、IRR 中断请求寄存器和控制逻辑电路等组成。

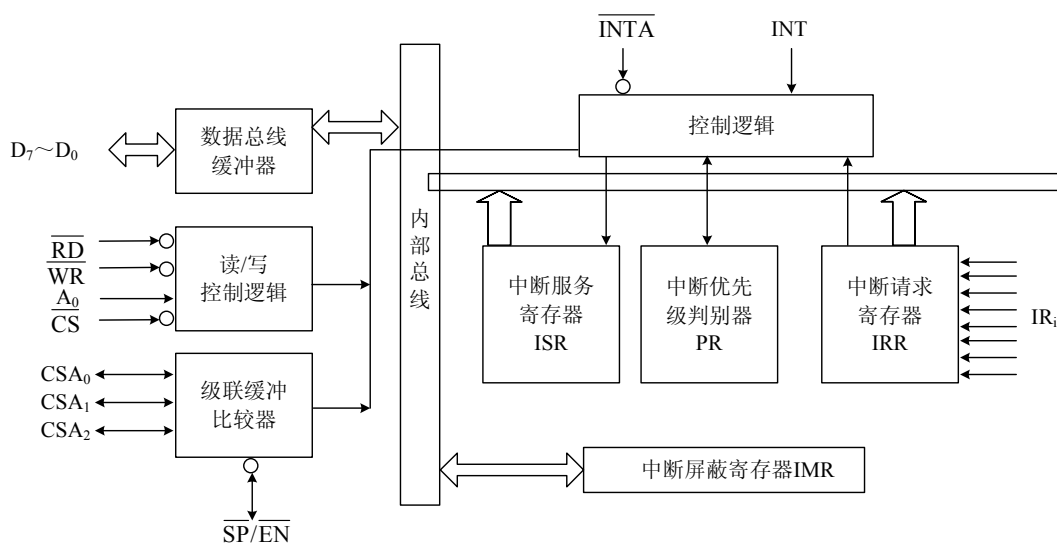


图 5-1 8259A 内部结构框图

(1) IRR 中断请求寄存器

中断请求寄存器, 8 位, 可存放 $IR_0 \sim IR_7$ 中断请求信号。外部中断源提出的请求, 由 $IRQ_0 \sim IRQ_7$ 送入中断请求寄存器, 使 IRR 中的相应位置 1。IRR 可以编程设定为电平触发或边沿触发。在 PC 中的 8 个中断请求输入端, 其中 IRQ_0 优先级最高, IRQ_7 优先级最低。

(2) IMR 中断屏蔽寄存器

中断屏蔽寄存器, 8 位, 与外部 $IR_0 \sim IR_7$ 相对应。IMR 接收来自处理器 CPU 的 8 位屏蔽字 (OCW1), 屏蔽字的各位分别对应 8 级中断请求, 某位为 1 时, 表示屏蔽对应的中断请求; 否则允许对应的中断请求进入优先级判别器。通过编程设定屏蔽字, 可以实现改变优先级。

(3) ISR 中断服务寄存器

中断服务寄存器, 8 位, 存放正在被服务的中断请求。无中断请求时, 各位都为 0, 某中断请求时, 相应位置 1, 表示中断服务程序未结束, 以便确定当有新的请求时, 能否进行中断嵌套, 处理完相应中断请求后清 0。中断服务寄存器对应中断请求寄存器的 8 位请求。

(4) PR 中断优先级判别器

中断优先级判别器, 用于确定中断请求寄存器 IRR 中各位的优先级等级。当 IRR 中有中断请求信号而被置位时, PR 就选出其中优先级最高的, 在 CPU 响应中断请求并发来第一个 $\overline{INTA}$ 脉冲时, 将其存放到 ISR 的相应位置。通过对 8259A 编程可以设定 $IR_0 \sim IR_7$ 的优先级等级。

（5）控制逻辑电路

控制逻辑的作用是根据 CPU 对 8259A 编程设定的工作方式产生 8259A 内部控制信号,在适当时候对 CPU 发出中断请求信号 INT,同时接收来自 CPU 的中断响应信号 $\overline{\text{INTA}}$,并将 $\overline{\text{INTA}}$ 转换为内部所需的各种控制信号。

（6）数据总线缓冲器

8 位双向三态缓冲器,用于连接系统数据总线和 8259A 内部总线,以便编程时由 CPU 对 8259A 写入控制字或读取状态字,或者中断响应时由 CPU 读取中断向量号。

（7）读/写控制逻辑

每片 8259A 占有两个端口地址,用 A_0 的取值来区分偶地址端口和奇地址端口。像 PC/XT 中只使用了一片 8259A,其端口地址为 20H 和 21H。而 PC/AT 系统中使用两片 8259A,主片的端口地址为 20H 和 21H,从片的端口地址为 A0H 和 A1H。CPU 可访问的 8259A 的内部寄存器不止两个,CPU 具体访问哪个寄存器,由 A_0 地址输入信号与读命令 $\overline{\text{RD}}$ 和写命令 $\overline{\text{WR}}$ 一起确定。

（8）级联缓冲/比较器

级联缓冲/比较器,用于控制多片 8259A 的级联,以实现将优先中断等级最多扩展到 64 级。多片连接时,一个为主片,其余的为从片。

8259A 在级联时,主片 8259A 的三条级联线 $\text{CSA}_0 \sim \text{CSA}_2$ 作为输出线,连接到每个从片 8259A 的 $\text{CSA}_0 \sim \text{CSA}_2$ 。每个从片 8259A 的中断请求信号 INT,连接到主片 8259A 的一个中断请求输入端 IR。主片 8259A 的 INT 线连接到 CPU 的中断请求输入端 INTR。

2. 8259A 的外部引脚

8259A 外部引脚图如图 5-2 所示,8259A 是具有 28 个引脚的集成芯片,各引脚功能如下所述:

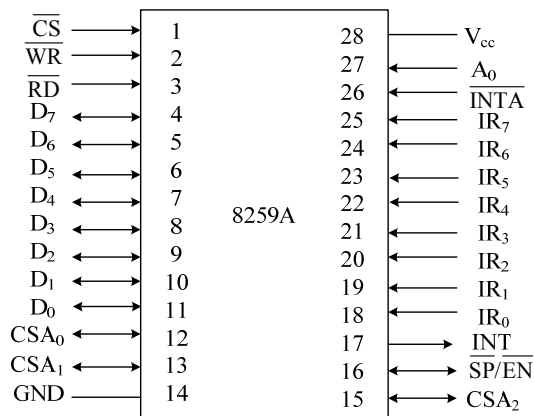


图 5-2 8259A 外部引脚图

$D_0 \sim D_7$: 双向数据线,与系统数据总线相连,接收 CPU 发来的命令字,给 CPU 提供中断向量号与内部寄存器状态;

$\text{IR}_0 \sim \text{IR}_7$: 中断请求输入端,与来自外设的中断请求输出端相连;

INT: 中断请求输出信号,与 CPU 的 INTR 引脚相连;

$\overline{\text{INTA}}$ ：中断响应输入端，接收 CPU 向 8259A 发送来的中断响应信号；

$\text{CSA}_0 \sim \text{CSA}_2$ ：级联地址，在 8259A 级联时使用。主片 8259A 从 $\text{CSA}_0 \sim \text{CSA}_2$ 输出级联地址，从片 8259A 从 $\text{CSA}_0 \sim \text{CSA}_2$ 接收级联地址，构成如图 5-3 所示。

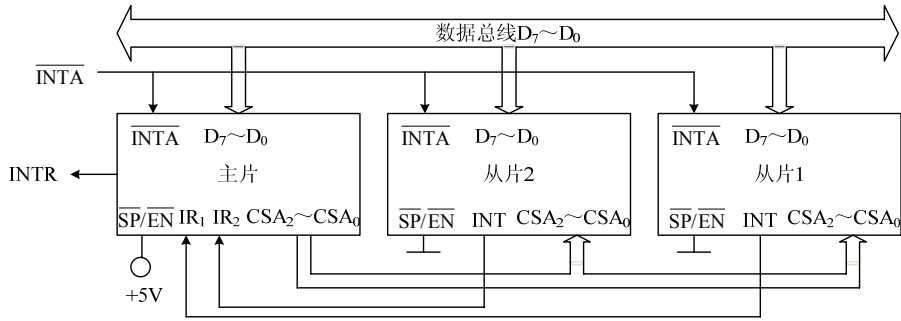


图 5-3 8259A 级联

设从片 1 与从片 2 同时对主片提出中断请求，而主片将从片 1 的中断请求发往 CPU 的 INTR 引脚，当 CPU 响应该中断请求后，发来第一个 $\overline{\text{INTA}}$ 脉冲，从片 1 和从片 2 均能收到这一信号，但不能确定是谁的中断请求被响应，由于从片 1 接在主片的 IR_0 引脚上，故主片在 $\text{CAS}_2 \sim \text{CAS}_0$ 放上“000”，表示接在 IR_0 引脚上的从片的请求被响应，从片 1 知道自己的 INT 引脚接在主片的 IR_0 输入端，故收到“000”后知道自己的中断请求被响应，它将在第二个 $\overline{\text{INTA}}$ 脉冲到来后将发出中断请求的外设的中断向量号放到系统数据线上供 CPU 读取。

$\overline{\text{CS}}$ ：片选信号输入端，与地址译码电路相连。低电平有效，若 $\overline{\text{CS}}=0$ ，该 8259A 芯片被选中，处于工作状态，允许它和 CPU 通信；若 $\overline{\text{CS}}=1$ ，则该 8259A 不接入系统。

A_0 ：地址输入端，与系统地址总线中的某位相连，用来选择 8259A 内部寄存器。与 $\overline{\text{CS}}$ 一起实现对 8259A 内部端口奇/偶地址的选择， $\text{A}_0=0$ ，选中偶地址，否则，选中奇地址。

$\overline{\text{RD}}$ 和 $\overline{\text{WR}}$ ：读/写控制输入信号端：分别与系统控制总线的 $\overline{\text{IOR}}$ 和 $\overline{\text{IOW}}$ 端相连，8259A 读写操作功能如表 5-1 所示。

表 5-1 8259A 读/写操作功能

$\overline{\text{CS}}$	$\overline{\text{WR}}$	$\overline{\text{RD}}$	A_0	D_4	D_3	功能
0	0	1	0	1	×	写 ICW1
0	0	1	1	×	×	写 ICW2
0	0	1	1	×	×	写 ICW3
0	0	1	1	×	×	写 ICW4
0	0	1	1	×	×	写 OCW1
0	0	1	0	0	0	写 OCW2
0	0	1	0	0	1	写 OCW3
0	1	0	0	×	×	读 IRR
0	1	0	0	×	×	读 ISR
0	1	0	1	×	×	读 IMR
0	1	0	1	×	×	读状态寄存器

注：表中 D_4 、 D_3 为对应状态寄存器中的标志位

$\overline{SP/EN}$ ：主从设定/缓冲读写控制，双功能线。当 8259A 工作在缓冲方式时，该引脚输出低电平控制信号，用来控制系统总线与 8259A 数据引脚之间的数据缓冲器的使能。当 8259A 工作在级联方式时，该引脚为输入线，此时，若 $\overline{SP}=1$ ，则设定 8259A 为主片；若 $\overline{SP}=0$ ，则设定 8259A 为从片。

V_{cc} ：+5V 电源输入线。

GND：接地线。

3. 8259A 的中断响应

具体中断请求和响应过程如图 5-4 所示，下面以 8259A 单片连接方式为例，说明中断请求及响应的基本过程。

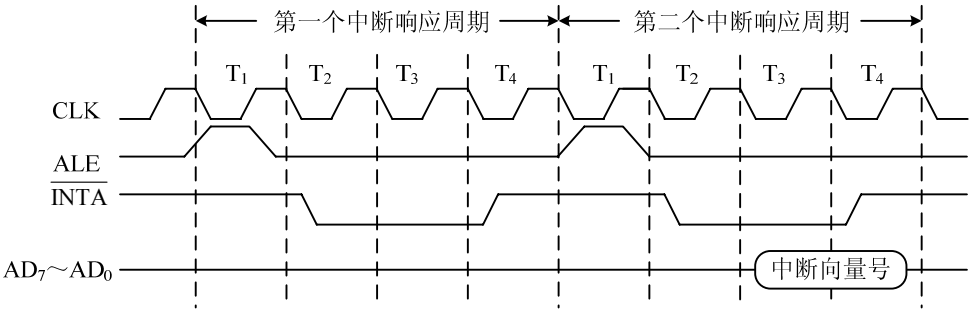


图 5-4 中断响应周期时序

1) 当 $IR_0 \sim IR_7$ 中有一个或几个中断源为高电平时，使相应的 IRR 位置位，表示中断请求。

2) 8259A 对 IRR 和 IMR 提供的情况进行分析，当请求的中断源未被 IMR 屏蔽时，如果这个中断请求是唯一的，或请求的中断比正在处理的中断优先级高，则向 CPU 发出 INT 中断请求信号。

3) CPU 在每个指令的最后一个时钟周期检查 INT 输入端的状态。当 CPU 允许可屏蔽中断状态，且无其他高优先级的中断时，就响应这个中断，产生两个中断响应周期。

4) 在 CPU 第一个中断响应周期中，8259A 接收第一个 $\overline{INTA}$ 信号后，将最高优先权的 ISR 位置位，而对应的 IRR 位则复位。

5) 在 CPU 第二个中断响应周期中，8259A 收到第二个 $\overline{INTA}$ 信号时，将中断向量号送到数据总线上。

5.1.2 8259A 的工作方式

1. 8259A 的工作方式

8259A 的中断管理功能很强，不仅能够实现常规的中断控制，还可以在此基础上进行扩展，来实现复杂的中断控制。单片 8259A 可以管理 8 级外部中断，在多片级联方式下最多可以管理 64 级外部中断，并且具有如图 5-5 所示的多种工作方式。

(1) 设置中断优先权方式

8259A 设置中断优先权方式有固定优先权方式和自动循环优先权方式两类工作方式。

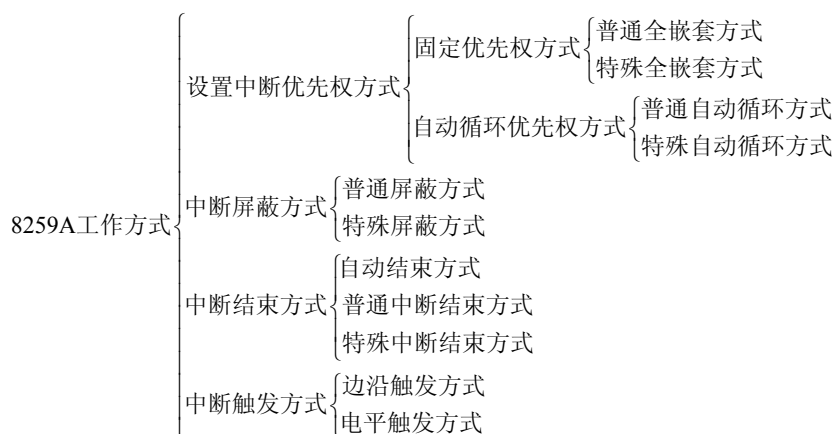


图 5-5 8259A 的工作方式

1) 固定优先权方式。固定优先权方式又可根据使用的系统不同分为普通全嵌套和特殊全嵌套两种工作方式。

- 普通全嵌套方式

8259A 的 $IR_0 \sim IR_7$ 的中断优先权的级别由系统确定。中断优先权的顺序固定不变，由高到低的优先级顺序是： $IR_0, IR_1, IR_2, \dots, IR_7$ ，其中， IR_0 的优先级最高， IR_7 的优先级最低。当有多个 IR_i 请求时，中断优先级判别器（PR）将它们与当前正在处理的中断源的优先权进行比较，选出当前优先权最高的 IR_i ，将其对应中断向量号送上数据总线，向 CPU 发出中断请求 INT，请求为其服务，对应 ISR 的 D_i 位置位，直到中断服务结束将该位复位。

普通全嵌套方式是 8259A 的默认工作方式，是最常用的工作方式，多用于 8259A 单片模式系统。

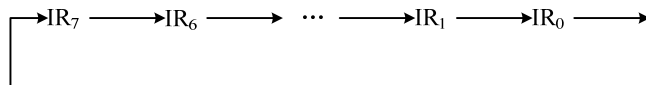
- 特殊全嵌套方式

该工作方式在处理中断请求时，允许同级中断的嵌套，除此之外，与普通全嵌套相同。该工作方式多用在 8259A 级联模式系统，在这样的系统中，主片 8259A 采用特殊全嵌套方式，从片采用其他优先权方式。此时，若来自某一从片的中断请求正在处理时，允许发生主片较高优先权的中断请求，同时也允许来自同一从片较高优先权的中断请求。

2) 自动循环优先权方式。自动循环优先权方式，8259A 的 $IR_0 \sim IR_7$ 的中断优先权的级别是可以改变的，按确定循环时的最低优先权的方式不同，可分为普通自动循环和特殊自动循环两种工作方式。

- 普通自动循环方式

在普通自动循环优先权方式中， $IR_0 \sim IR_7$ 中初始最高优先级由系统指定，即指定 IR_0 的优先级最高，以后变化规律按：当某一个中断请求 IR_i 的中断服务结束后，该中断的优先权自动降为最低，而紧随其后的中断请求 IR_{i+1} 的优先权自动变为最高。 $IR_0 \sim IR_7$ 优先权级别按如下所示的右循环方式改变。



设在初始状态 IR_0 有请求, 且为其服务完毕, IR_0 优先权自动降为最低, 排在 IR_7 之后, 而 IR_1 的优先权升为最高, 其余依次类推。这种优先权管理方式, 可以使 8 个中断请求都拥有享受同等优先服务的权利。

系统中是否采用自动循环方式, 由 8259A 的操作方式命令字 OCW2 决定。

- 特殊自动循环方式

同优先级普通自动循环方式, 但一开始时的优先级可以通过编程设定。例如, 一开始设定 IR_2 最低, 则 IR_3 的优先级最高, 则优先级顺序从高到低依次为: $IR_3, IR_4, \dots, IR_2$ 。

(2) 中断屏蔽方式

中断屏蔽方式是对 8259A 的外部中断源 $IR_0 \sim IR_7$ 实现屏蔽的一种中断管理方式, 分为普通屏蔽方式和特殊屏蔽方式两种。

1) 普通屏蔽方式。普通屏蔽方式是通过 8259A 的中断屏蔽寄存器 (IMR) 来实现对中断请求 IR_i 的屏蔽。由编程写入操作命令字 OCW1, 将 IMR 中的 D_i 位置 1, 则对应的 IR_i ($i=0 \sim 7$) 中断请求被屏蔽。如果 D_i 位置 0, 则允许 IR_i 中断请求。

2) 特殊屏蔽方式。特殊屏蔽方式是只对同级中断请求进行屏蔽, 对于高或低优先级中断请求均不屏蔽。在这种屏蔽方式中, 将 IMR 的 D_i 位置 1, 同时使 ISR 的 D_i 位置 0。这种屏蔽方式通常用于级联方式或在同一请求端 IR_i 上连接有多个中断源的场合。特殊屏蔽方式可以通过编程写入操作命令字 OCW3 来设置或取消。

(3) 中断结束方式

按照对中断处理的结束方法来分, 8259A 有两类结束方式, 即自动结束方式和非自动结束方式。而非自动结束方式又分为两种, 一种叫普通中断结束方式, 另一种叫特殊中断结束方式。

当一个中断请求得到响应时 8259A 就会在中断服务寄存器 ISR 中设置相应位 D_i , 这样, 为以后中断优先级判别器的工作提供了依据。当中断处理程序结束时, 必须使 D_i 位置零, 这个使 D_i 位清零的动作就是中断结束处理。

1) 自动结束方式 (AEIO 方式)。当一个中断请求被响应后, 在收到第一个 $\overline{INTA}$ 信号后, 8259A 将 ISR 中的对应位置 “1”, 在收到第二个 $\overline{INTA}$ 信号后, 8259A 将 ISR 中的对应位置 0。此刻, 中断服务程序并没有结束, 处理器正在执行该中断服务程序, 而在 8259A 中, 中断服务寄存器中却没有对应作指示, 故认为其服务已结束。此时若有更低级的中断请求信号, 8259A 仍可向 CPU 发送中断请求, 从而会造成低级中断打断高级中断的情况。这种方式一般用于单片 8259A 而且不会产生嵌套的情况。

2) 普通中断结束方式 (普通 EOI 方式)。普通中断结束方式配合全嵌套优先权方式使用。当 CPU 用 OUT 指令向 8259A 传送一个普通中断结束方式命令时, 8259A 就会把所有正在服务的优先权最高的中断 ISR 复位。因为在全嵌套方式中, 当前 ISR 最高优先权中断对应了最后一次被响应的和被处理的中断, 也就是当前正在处理的中断。所以, 当前最高优先权的 ISR 位复位相当于结束了当前正在处理的中断。

普通中断结束方式的发送: 在程序中向 8259A 的偶地址端口输出一个操作命令字 OCW2,

并使得 OCW2 中的 $EOI=1$, $SL=0$, $R=0$ 。

3) 特殊中断结束方式 (特殊 EOI 方式)。特殊中断结束方式配合循环优先权方式使用。在优先权循环的情况下, 无法根据 ISR 的内容来确定哪一级中断是最后响应和处理的, 即不能从 ISR 中 “1” 的位置确定当前的最高优先级。这样, 当 CPU 用 OUT 指令向 8259A 传送一个特殊中断结束方式命令时, 该命令中指明将 ISR 中的哪一位置 0。

(4) 中断触发方式

中断触发方式是指中断请求信号 IR 的有效形式, 8259A 中断请求输入端 $IR_0 \sim IR_7$ 的触发方式有边沿触发和电平触发两种, 由初始化命令字 ICW1 中的 LTIM 位来设定。

1) 边沿触发方式。在边沿触发方式下, 8259A 将中断请求输入端出现的上升沿作为中断请求信号。 IR_i 出现上升沿触发信号以后, 可以一直保持高电平。

2) 电平触发方式。在电平触发方式下, 以 IR_i 端输入高电平信号作为中断请求信号。在这种触发方式中, 要求触发电平必须保持到中断响应信号 $\overline{INTA}$ 有效为止, 并且在 CPU 响应中断后, 及时撤销该信号, 以防止 CPU 再次响应, 出现重复中断现象。

2. 8259A 与处理器的连接

8259A 与处理器的连接方式有缓冲和非缓冲两种。

(1) 非缓冲方式

非缓冲方式, 当系统中 8259A 的数量较少时, 可将 8259A 直接与系统数据总线相连, 如图 5-6 所示。该方式是通过 8259A 的初始化命令字 ICW4 设置。

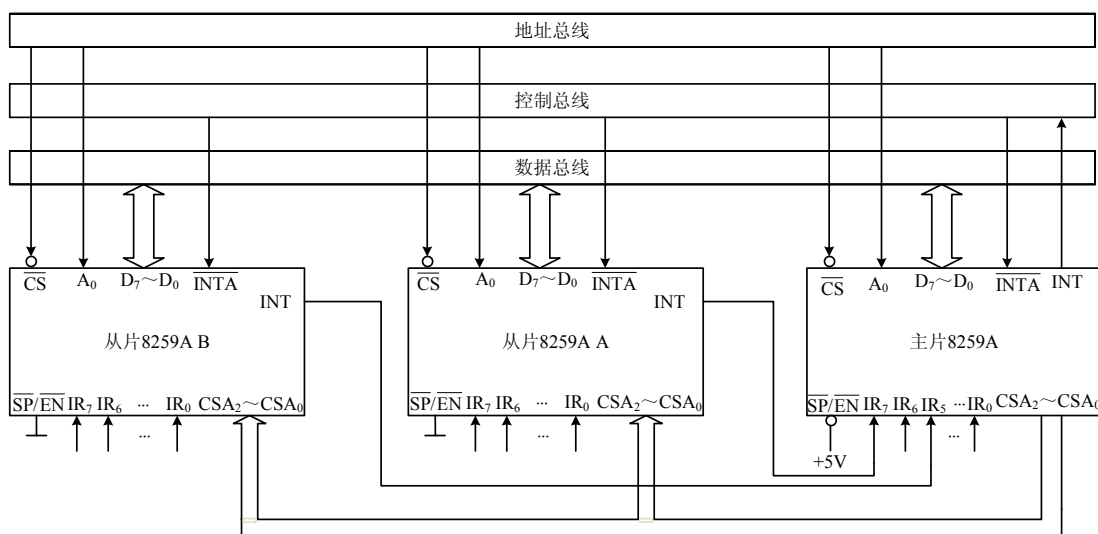


图 5-6 8259A 级联非缓冲连接方式

在此工作方式下, $\overline{SP/EN}$ 用于控制 8259A 在级联工作方式下的主/从关系。主片的 $\overline{SP/EN}$ 引脚接高电平, 从片的 $\overline{SP/EN}$ 引脚接低电平。

在级联系统中, 主/从 8259A 分别都要进行各自的初始化, 设置各自的工作方式。由于在中断响应周期主/从 8259A 的 ISR 所对应的 ISR_i 均被置 1, 故在中断结束时要分别对主/从 8259A 写入 EOI 命令, 结束中断操作。

(2) 缓冲方式

缓冲方式，当系统中 8259A 的数量较多时，考虑到系统总线带负载能力有限，在 8259A 的数据总线与系统数据总线间通过双向总线驱动器（如 8286）相连，如图 5-7 所示。缓冲方式也可以通过对 ICW4 的重新设置来实现。

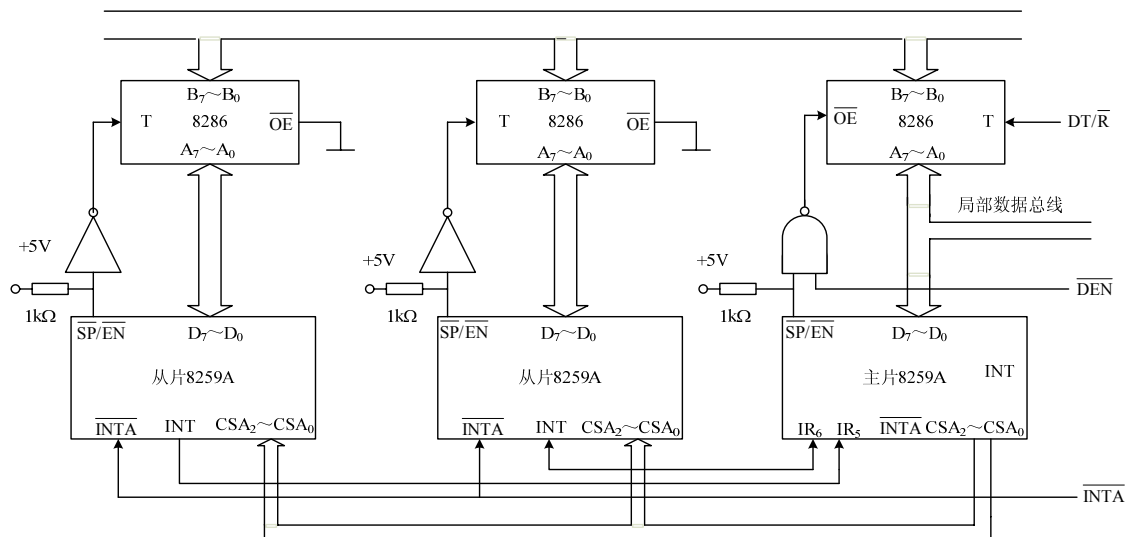


图 5-7 8259A 级联缓冲连接方式

8259A 的 $\overline{\text{SP/EN}}$ 输出为 $\overline{\text{EN}}$ ，用于控制数据缓冲器的数据传输方向，当 $\overline{\text{EN}}$ 为“0”，CPU 通过数据缓冲器读取 8259A 输出数据，当 $\overline{\text{EN}}$ 为“1”，CPU 通过数据缓冲器对 8259A 写入命令。图中 $\overline{\text{SP/EN}}$ 所接上拉电阻使 8259A 在初始化期间 $\overline{\text{SP/EN}}$ 为高电平，确保通过数据缓冲器对 8259A 执行正确的写操作。

5.1.3 8259A 的应用实例

1. 8259A 应用编程

8259A 应用编程分为初始化命令字编程和操作命令字编程两种方式。在 8259A 内部有两组寄存器，一组为命令寄存器，用于存放 CPU 写入的初始化命令字 ICW (Initialization Command Words)；另一组为操作命令寄存器，用于存放 CPU 写入的操作命令字 OCW (Operation Command Words)。8259A 在开始操作之前，必须对它写入初始化命令字，使它处于预定的初始状态；在 8259A 工作期间，可以在任何时刻通过写入操作命令字来控制 8259A 执行不同的操作方式，如优先权循环、中断屏蔽、中断结束和 8259A 内部寄存器状态的读出和查询等。OCW 设置灵活，视需要写入，可以用其来动态控制 8259A 的中断管理方式。

(1) 初始化命令字 ICW

如图 5-8 所示是 8259A 初始化编程流程图, 由于 8259A 只有一位内部寄存器选择信号 A_0 用来选择 8259A 内部寄存器, 所以初始化命令字必须按初始化编程流程 ICW1~ICW4 的顺序依次写入, 但若其中某个 (ICW3 或 ICW4) 不需要, 则不用写入, 而直接写入下一个命令字。

ICW1 占用一个端口地址，约定 A_0 选择线为“0”；ICW2~ICW4 公用一个端口地址，约定 A_0 选择线为“1”，即只要初始化命令字 D_4 位为“1”和 A_0 位为“0”，8259A 就认为是对 ICW1 寄存器写入命令字；若 A_0 为“1”，则分别按顺序对 ICW2~ICW4 寄存器写入命令字。

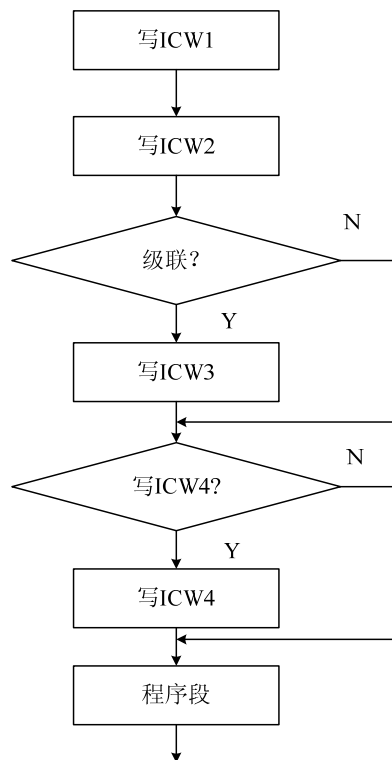


图 5-8 8259A 初始化编程流程图

1) 初始化命令字 ICW1。ICW1 是 8259A 第一个写入的命令字。占用偶地址端口，即 $A_0=0$ 。ICW1 的格式如图 5-9 所示，各位含义如下。

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
/	/	/	1	LTIM	ADI	SNGL	IC ₄

图 5-9 ICW1 格式

D_0 (ICW4 Needed /No ICW4 Needed, IC₄): 用于规定是否要 ICW4 初始化命令字。 D_0 位为“1”，表示要对 ICW4 初始化命令字进行初始化编程， D_0 位为“0”，则表示不要对 ICW4 初始化命令字进行初始化编程。在 80x86 CPU 系统中需要定义 ICW4，设 IC₄=1。

D_1 (Single/Cascade Mode, SNGL): 用于规定 8259A 是否为级联工作方式。 D_1 位为“1”，表示 8259A 为单片工作方式，不需要对 ICW3 初始化编程； D_1 位为“0”，表示 8259A 工作在级联工作方式，要对 ICW3 初始化编程。

D_2 (CALL Address Interval, ADI): 在 8086/8088 系统中这一位无效。

D_3 (Level/Edge Triggered Mode, LTIM): 用于规定 8259A IR_i 的中断请求触发方式。 D_3 位为“1”，表示中断请求为电平触发方式； D_3 位为“0”，表示中断请求为边沿触发方式。

D₄: ICW1 的标志位, D₄ 恒为“1”, 表示当前写入的是 ICW1 初始化命令字。

D₅~D₇: 在 8086/8088 系统中, 这 3 位未用, 通常设置为“0”。

当 CPU 向 8259A 送入一条 A₀=0、D₄=1 的命令时, 该命令被译码为 ICW1, 它启动 8259A 的初始化过程, 相当于 RESET 信号的作用, 自动完成下列操作: 清除中断屏蔽寄存器 IMR, 设置以 IR₀ 为最高优先级、依次递减, 以 IR₇ 为最低优先级的全嵌套方式, 固定中断优先权排序。

2) 初始化命令字 ICW2。ICW2 用于设置中断向量号, 格式如图 5-10 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
					ID ₂	ID ₁	ID ₀

图 5-10 ICW2 格式

ICW2 中的低 3 位 ID₀~ID₂ 由 8259A 自动按中断请求输入端 IR_i 编码引入: IR₀ 为 000, IR₁ 为 001, ..., IR₇ 为 111; 高 5 位 D₇~D₃ 由用户编程写入, 作为中断向量号的高 5 位, 若 ICW2 为 40H 时, 则 IR₀~IR₇ 对应的中断向量号为 40H~47H。

中断向量号 n 是在中断响应周期内的第二个 $\overline{\text{INTA}}$ 信号有效期间, 由 8259A 通过数据线总线 D₇~D₀ 传送给 CPU, 由 CPU 根据 4*n 地址指针从中断向量表中取出当前响应的中断服务程序的入口地址, 转去执行相应的中断服务程序。

D₃~D₇ 由用户定义。

3) 初始化命令字 ICW3。ICW3 用于设置级联方式, 占用奇地址端口, 当 8259A 工作在图 5-7 所示的级联方式时, 即当初始命令字 ICW1 中, D₁ 位 (SNGL) = 0 时, 需要对 ICW3 初始化命令字进行编程。主片和从片所对应的 ICW3 的格式不同, 主片 ICW3 的格式如图 5-11 所示, 从片 ICW3 的格式如图 5-12 所示。

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	S ₇	S ₆	S ₅	S ₄	S ₃	S ₂	S ₁	S ₀

图 5-11 主片 ICW3 格式

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	0	0	0	0	0	ID ₂	ID ₁	ID ₀

图 5-12 从片 ICW3 格式

- 主片 8259A

ICW3 的每一位 S₀~S₇ 与 IR₀~IR₇ 对应, 若主片 IR_i (i=0~7) 引脚上连接从片, 则 S_i=1, 否则 S_i=0。

- 从片 8259A

ICW3 的低 3 位 ID₀~ID₂ 有效, ID₀~ID₂ 可有 8 种组合编码, 表示从片的中断请求输出 INT 被连到主片的哪一个中断请求输入端 IR_i 上, 若该从片的中断请求信号 INT 连接到主片的 IR₃ 上, 则 ID₂~ID₀ 设置为 011。

4) 初始化命令字 ICW4。ICW4 用于设置 8259A 的基本工作方式, 即确定 8259A 是否工作在 80x86CPU 系统中, 其格式如图 5-13 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0	SFNM	BUF	M/ $\bar{S}$	AEOI	μ PM

图 5-13 ICW4 格式

D₀ (Microprocessor, μ PM): 用于确定选用的微处理器是 16 位 80x86 CPU 系统, 还是 8 位 8080/8085 CPU 系统, μ PM=1, 选择 80x86 CPU 系统, 否则, 选择 8 位系统。

D₁ (Auto End of Interrupt, AEOI): 用于确定 8259A 采用自动中断结束方式, 还是采用非自动中断结束方式, AEOI=1, 采用自动中断结束方式, 否则, 采用非自动中断结束方式。

D₂ (Master/Slave, M/ $\bar{S}$): 用于确定 8259A 是主片, 还是从片, M/ $\bar{S}$ =1, 8259A 是主片, 否则, 8259A 是从片。

D₃ (Buffer, BUF): 用于确定 8259A 数据采用缓冲方式, 还是非缓冲方式, BUF=1, 采用缓冲方式, 否则, 采用非缓冲方式。

当在多片 8259A 级联系统中, 若在 8259A 的数据线与系统总线之间加入双向总线驱动器, $\overline{SP}/\overline{EN}$ 引脚作为总线驱动器的控制信号, D₃ 位 BUF 应设置为 1, 此时主片和从片的区分不能依靠 $\overline{SP}/\overline{EN}$ 引脚, 而是由 M/ $\bar{S}$ 来选择, 当 M/ $\bar{S}$ =0 时为从片; 当 M/ $\bar{S}$ =1 时为主片。如果 BUF=0, 则 M/ $\bar{S}$ 定义无意义。

D₄ (Special Fully Nested Mode, SFNM): 用于确定 8259A 工作在特殊全嵌套方式, 还是工作在普通全嵌套方式, SFNM=1, 8259A 工作在特殊全嵌套方式, 否则, 工作在普通全嵌套方式。

D₇~D₅: 未使用, 建议设置为“0”。

(2) 操作命令字 OCW

当按照一定的初始化顺序对 8259A 设置完毕后, 8259A 便进入了工作状态, 这时可以写入操作命令字 OCW 来设定和控制 8259A 的工作状态和工作方式。此时, 8259A 开始接收通过 IR_i 输入的中断请求信号, 来响应和管理中断请求。8259A 的操作命令字有 OCW1、OCW2 和 OCW3 三个。和初始化命令字 ICW 不同, OCW 不是按照既定的流程写入, 而是由 CPU 按照用户程序的需要写入的。操作命令字可以在主程序中使用, 也可以在中断服务程序中使用。

1) 操作命令字 OCW1。OCW1 是中断屏蔽操作命令字, 用于屏蔽 8259A 的外部中断请求信号 IR_i, 对中断屏蔽寄存器 IMR 的各位进行设置操作。其格式如图 5-14 所示。OCW1 使用端口选择线 A₀ 为“1”的端口地址, 即占用奇地址端口。OCW1 与中断屏蔽寄存器 IMR 中的各位一一对应, M₇~M₀ 分别对应 IR₇~IR₀ 上的中断请求, 如某位置“1”, 相应的 IR_i 输入被屏蔽, 但不影响其他中断请求输入引脚; 若某位置“0”, 则相应中断请求允许。在工作期间可根据需要随时写入或读出。在程序中设置屏蔽位时, 一般采用先读出中断屏蔽寄存器的现有内容 (初始状态为 0), 然后再使对应的屏蔽位置“1”或置“0”, 其他屏蔽位原有的内容保持不变, 将新的屏蔽字写入中断屏蔽寄存器 IMR。

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
1	M ₇	M ₆	M ₅	M ₄	M ₃	M ₂	M ₁	M ₀

图 5-14 OCW1 格式

2) 操作命令字 OCW2。OCW2 是用于控制中断结束方式和优先级循环方式的操作命令字。其格式如图 5-15 所示。

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	R	SL	EOI	0	0	L ₂	L ₁	L ₀

图 5-15 OCW2 格式

D₂~D₀ (IR Level to Be Acted Upon, L₂~L₀): 用于说明是 8 个中断请求输入端 IR₇~IR₀ 哪一级中断, 对应 IR_i 的编码。当 SL=1 时, D₂~D₀ 有效, 否则无效。

D₄、D₃: 是 OCW2 的特征位, 恒为 0, 表示 CPU 执行写入 OCW2 操作。

D₅ (EOI): 中断结束命令位。在非自动中断结束方式下, EOI=1, 设置中断结束命令, 8259A 收到此命令, 使中断服务寄存器 ISR 中具有最高优先权的对应位复位; EOI=0, 则没有设置中断结束命令字。

D₆ (Specific Level, SL): 用于指明 OCW2 中 L₂~L₀ 位是否有效。SL=1 时, L₂~L₀ 有效; 否则无效。

D₇ (Rotation, R): 优先级循环方式设置位。当 R=1 时, 使用自动循环优先级方式; 否则, 为固定优先级方式。

D₇~D₅ 三个控制位组合用于选择中断结束和循环的操作方式, 命令如表 5-2 所示。

表 5-2 D₇~D₅ 三个控制位组合

R	SL	EOI	选择
0	0	1	普通 EOI 方式
0	1	1	特殊 EOI 方式
1	0	1	自动循环普通 EOI 方式
1	1	1	自动循环特殊 EOI 方式
1	0	0	设置优先级自动循环命令
0	0	0	清除优先级自动循环命令
1	1	0	置位优先级命令
0	1	0	无操作

3) 操作命令字 OCW3。OCW3 用于控制 8259A 的运行方式为特殊中断屏蔽方式 (设置), 还是一般屏蔽方式 (撤销); 还用于查询和读取 8259A 的有关寄存器的状态。其格式如图 5-16 所示。

A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	/	ESMM	SMM	0	1	P	RR	RIS

图 5-16 OCW3 格式

D₁ (Read Register Command, RR): 读寄存器命令位。用于是否允许读取中断请求寄存

器 IRR 和中断服务寄存器 ISR。当 RR=1 时, 允许读取 IRR 或 ISR; 否则禁止读取这两个寄存器。

D₀ (Read Interrupt Register Select, RIS): 读寄存器选择位。只有当 RR 位为“1”时, 该位有意义。当 RR=1, RIS=0, 读取 IRR 寄存器的状态; 当 RR=1, RIS=1, 读取 ISR 寄存器的状态。

D₂ (Poll Command, P): 查询方式位。当 P=1 时, 8259A 被设置为中断查询方式工作; 当 P=0 时, 表示 8259A 未被设置为中断查询方式工作。当 CPU 向 8259A 发出查询命令后, 通过读入状态寄存器的内容, 如果某 IR 中断请求信号有效, 即查询到有中断请求正在被处理, 则把相应的 ISR 置位, 并读出该中断的优先级。在查询读期间, 8259A 输出到数据总线上的中断状态字格式如图 5-17 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
IR	/	/	/	/	L ₂	L ₁	L ₀

图 5-17 中断状态寄存器

D₄、D₃: 是 OCW3 的特征位, 固定为 01, 表示 CPU 执行写入 OCW3 操作。

D₅ (Special Mask Mode, SMM) 和 D₆ (Enable Special Mask Mode, ESMM): 组合用于允许和禁止特殊屏蔽方式。当 ESMM=1 时, SMM=1, 则将 8259A 设置为特殊屏蔽方式, 该方式下只要 CPU 的 IF=1, 系统就只屏蔽本级中断请求, 开放高级或低级的中断请求; SMM=0, 则撤销 8259A 特殊屏蔽方式, 恢复为原来的优先级控制, 普通屏蔽方式; 当 ESMM=0 时, 禁止 SMM 起作用。

2. 8259A 应用举例

下面就将以上内容以实例的形式讲述关于 8259A 的应用。

例 5-1 在某个 8086 最小方式系统中接有一片 8259A, 有一外设中断请求从 IR₇ 输入, 8259A 端口地址如图 5-18 所示。试按要求写出 8259A 的初始化程序。

要求: 单片工作方式; 8259A 的工作方式为: 中断请求信号上升沿触发; 采用全嵌套方式; 采用普通 EOI 方式; 中断向量号分布为 0CH~0C7H; 屏蔽 IR₀~IR₆ 的中断请求, 允许 IR₇ 的中断请求。

如图 5-18 所示, 8259A 具有两个端口地址, 由 CPU 的地址线 A₁ 控制, 其端口号地址的高位由译码器的输出端 $\bar{Y}_1$ 提供, 组合逻辑电路可以使其 8259A 为偶地址, 因而 8259A 的命令等可以由 8086CPU 低 8 位数据线传输, 端口地址为 84H 和 86H。从 IR₇ 输入的中断向量号为 0C7H, 8086CPU 获得中断向量号并乘以 4, 从中断服务入口地址表中找到相应的中断服务程序的地址, 然后转去执行中断服务程序。初始化程序包括对 8259A 的设置和将中断服务程序首地址填入中断向量表中。

```
INTRRUP SEGMENT AT 0
    ORG 0C7H*4
    DW OFFSET INTR_IR7    ;填写中断向量表
    DW SEG INTR_IR7
    ...
```

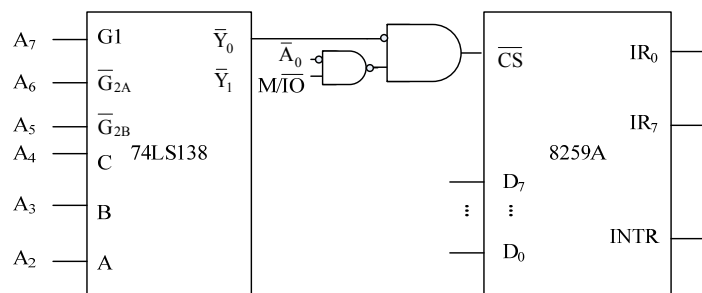


图 5-18 8259A 端口地址形成电路

MAIN SEGMENT

...

```

MOV AL, 00010011B      ;写 ICW1，表示上升沿触发及单片工作方式
                        ;并要设 ICW4

OUT 84H, AL

MOV AL, 0C0H           ;写 ICW2，表示中断向量基值为 0C0H

OUT 86H, AL

MOV AL, 00000001B      ;写 ICW4，表示全嵌套方式非缓冲，普通 EOI 方式

OUT 86H, AL

MOV AL, 01111111B      ;写 OCW1，中断屏蔽字，仅允许 IR7 中断

OUT 86H, AL

STI

```

...

由于 ICW4 所设定的优先级方式为全嵌套方式，中断结束管理方式是普通 EOI 方式，所以在中断服务子程序 INTR_IR7 结束时，要给 8259A 传输一个 EOI 命令，使 8259A 将 ISR 寄存器中级别最高的位“1”复位为“0”。执行：

```

MOV AL, 20             ;写 OCW2

OUT 84H, AL

```

例 5-2 若 8086 系统采用级联方式，主 8259A 的中断类型码从 30H 开始，端口地址为 20H，21H。从 8259A 的 INT 接主片的 IR₇，从片的中断类型码从 40H 开始，端口地址为 0A0H, 0A1H。均不要 ICW4。试对其进行初始化编程。

源代码如下：

```

M8259 EQU 20H          ;主 8259A 地址
S8259 EQU 0A0H         ;从 8259A 地址

```

```

STACK SEGMENT STACK
STA    DW 50 DUP(?)
TOP    EQU LENGTH STA
STACK ENDS

```

```

CODE SEGMENT
    ASSUME CS:CODE,DS:CODE,SS:STACK
START:    ;主 8259A 初始化
        MOV AL,00010000B    ;设 ICW1, 边缘触发, 多片级连, 不需要 ICW4
        MOV DX,M8259
        OUT DX,AL
        MOV AL,30H          ;设 ICW2, 设置中断类型码 30H
        INC DX
        OUT DX,AL
        MOV AL,80H          ;设 ICW3, 接 IR7
        OUT DX,AL
        ;从 8259A 初始化
        MOV AL,00010000B    ;设 ICW1, 边缘触发, 多片级连, 不需要 ICW4
        MOV DX,$8259
        OUT DX,AL
        MOV AL,40H          ;设 ICW2, 设置中断类型码 40H
        INC DX
        OUT DX,AL
        MOV AL,07H          ;设 ICW3, IR7 引入
        OUT DX,AL
CODE ENDS
    END START

```

5.2 定时/计数器控制接口技术

在微机应用系统中, 定时控制具有重要的作用。经常会有这样的应用要求: 一种是要有一些外部实时时钟, 以实现延时控制或定时; 另一种是要求能对外部事件计数的计数器。在微机系统中, 常采用以下三种方法实现: 软件定时, 不可编程硬件定时和可编程硬件定时。

软件定时就是通过执行一段固定的循环程序来实现定时。由于 CPU 执行每条指令需要一定的时间, 因此执行一个循环程序就可以实现定时作用。这种方法通用性和灵活性都很好, 但要占用 CPU 的时间。

不可编程硬件定时就是设计数字逻辑电路来实现计数或定时, 采用中小规模集成电路器件来构成电路。比较常见的定时器有单稳触发器和 555 定时器等。这种电路一经连接好后, 定时值就不便控制和修改, 若要改变计数/定时的要求, 必须改变电路参数。通用性、灵活性差。

可编程硬件定时就是在不可编程硬件定时的基础上加以改进, 使定时值可以通过软件来确定和改变。采用可编程定时器/计数器来实现, 其定时与计数功能可以通过程序方便灵活的设定, 设定后可以同 CPU 并行工作, 不占用 CPU 的时间。

目前,在微机应用系统中普遍采用 Intel 公司生产的可编程定时/计数接口芯片 8253/8254。8253 是可编程间隔定时器,同时可用于事件计数器。每个 8253 芯片内部有 3 个独立的 16 位计数器,每个计数器有 6 种工作方式。8254 是 8253 的改进芯片,内部工作方式和外部引脚与 8253 完全相同,只是增加了读回命令和状态字。

本章主要讲述可编程计数器/定时器的基本原理,对定时/计数器芯片 Intel 8253/8254 进行详细的讨论,最后以实用的角度举例说明其使用方法。

5.2.1 8253/8254 的内部结构和外部引脚

1. 8253/8254 的内部结构

由图 5-19 所示 8253 内部结构框图可知,它由数据总线缓冲器、读/写控制逻辑、控制字寄存器以及 3 个计数器(计数器 0、计数器 1、计数器 2)等组成。

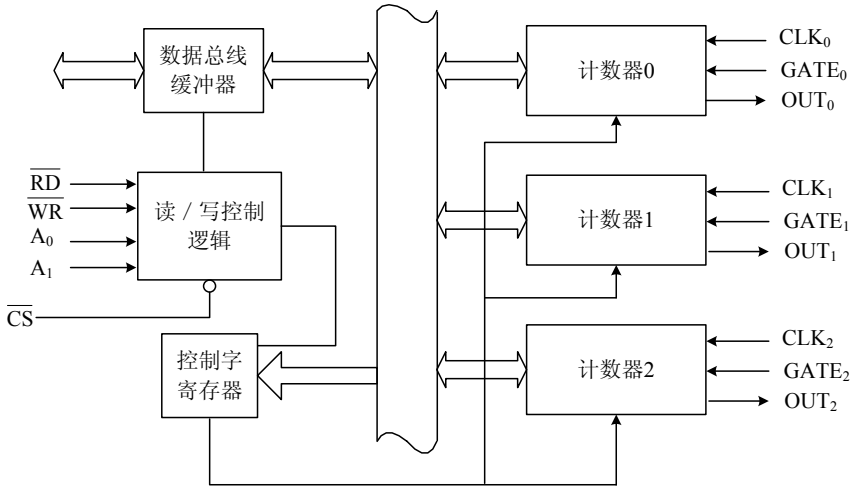


图 5-19 8253 内部结构框图

(1) 数据总线缓冲器

8 位双向三态缓冲器,用于连接系统数据总线和 8253 内部总线。以便编程时由 CPU 对 8253 写入控制字、写入计数初值或从计数器读取计数值。

(2) 读/写控制逻辑

8253 内部的控制电路,读写逻辑接收来自 CPU 的控制信号,包括读信号 $\overline{RD}$ 、写信号 $\overline{WR}$ 、片选信号 $\overline{CS}$ 和芯片内部寄存器寻址信号 A_1A_0 。片选信号 $\overline{CS}=0$ 时,由 A_1A_0 信号选择芯片内部寄存器的寻址,由读信号 $\overline{RD}$ 和写信号 $\overline{WR}$ 完成对寄存器的读、写操作。

(3) 控制字寄存器

控制字寄存器接收 CPU 送来的控制字,决定每个计数器的工作方式、读写格式和计数的数制。当 $A_1A_0=11$ 时,用来接收 CPU 输出的控制字。该寄存器是个只能写入寄存器。

(4) 计数器(计数器 0、计数器计 1、数器 2)

如图 5-20 所示,8253 内部有 3 个相互独立且结构功能完全相同的定时/计数器:0、1 和 2。每个计数器包含一个控制字寄存器、16 位的初值寄存器 CR (Count Register)、减“1”计数器

CE (Counting Element) 和 16 位输出锁存寄存器 OL (Output Latch)。减“1”计数器的初始值由程序设定，是初值寄存器的内容。计数器的当前值存放在输出锁存寄存器，CPU 可以在需要的时候读出这个值。

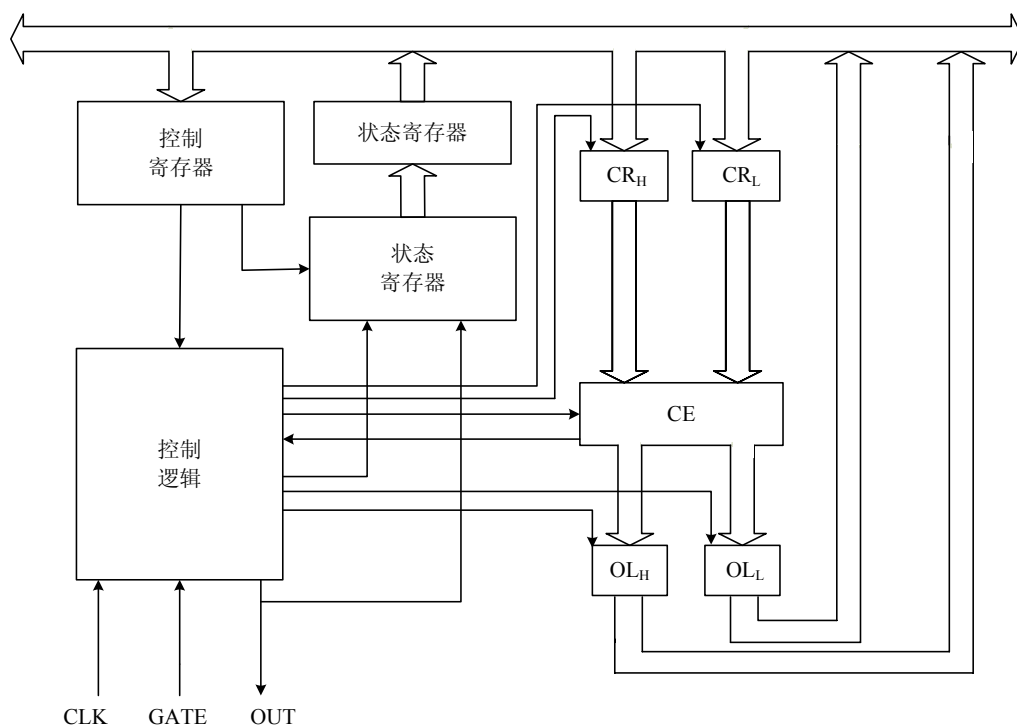


图 5-20 8253 计数器的逻辑结构

每个计数器有 3 根信号线：时钟输入信号 CLK、门控信号输入 GATE 和输出信号 OUT。计数器工作时，每出现一个 CLK 脉冲，减“1”计数器中的计数值减 1，当计数值减为 0 时，通过 OUT 输出结束信号，表明计数执行单元已为 0，输出信号的波形由工作方式确定。当 CLK 是一个周期性的时钟信号时，计数器为定时器功能；当 CLK 是一个非周期性事件计数信号时，此时为计数器功能。8253 的定时时间取决于时钟脉冲的频率和计数器的初值，

定时时间 $T = \text{时钟脉冲周期 } t_c \times \text{计数初值 } n$

此外，可以通过 GATE 引脚上的门控信号允许或停止计数器的过程。

2. 8253 的外部引脚

如图 5-21 所示为 8253 外部引脚图，8253 是具有 24 个引脚的集成芯片，各引脚功能如下所述：

$D_7 \sim D_0$ ：双向数据线，与系统数据总线相连，供 CPU 对 8253 进行读/写操作，进行信息交换。

$\overline{CS}$ ：片选线，输入低电平有效。当 $\overline{CS} = 0$ 时，表示 CPU 选中 8253 芯片，可对 8253 进行读/写操作；当 $\overline{CS} = 1$ 时，表示 CPU 未选中 8253 芯片。 $\overline{CS}$ 信号是由 CPU 输出的高位地址（通常 $A_9 \sim A_2$ ）和 AEN 信号经过电路译码后产生。

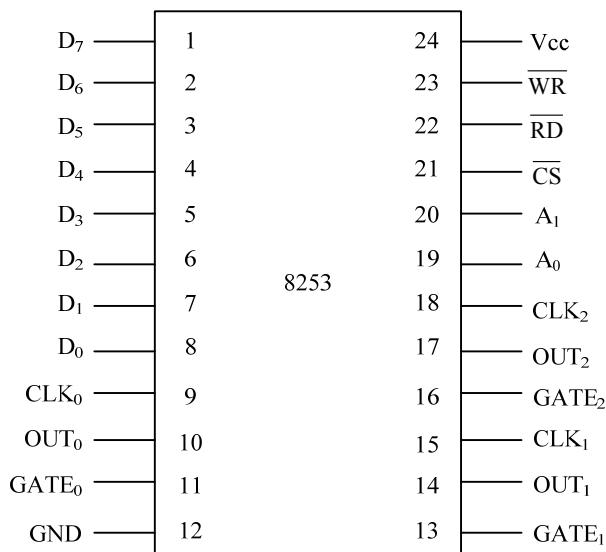


图 5-21 8253 外部引脚图

$A_1 \sim A_0$ ：地址输入信号，用于选择 8253 内部寄存器。当 $\overline{CS}$ 为低电平有效时，选中 8253 芯片，然后通过 $A_1 A_0$ 组合编码决定选中计数器，如表 5-3 所示。

表 5-3 $A_1 A_0$ 组合编码

A_1	A_0	寄存器选择
0	0	计数器 0
0	1	计数器 1
1	0	计数器 2
1	1	控制寄存器

$\overline{RD}$ 和 $\overline{WR}$ ：读/写控制输入信号端，低电平有效。当 $\overline{CS}=0$ ， $\overline{RD}=0$ 时，表示 CPU 可对 8253 的一个计数器进行读操作；当 $\overline{CS}=0$ ， $\overline{WR}=0$ 时，表示 CPU 可对 8253 的一个计数器或控制字寄存器进行写操作。

GATE：门控输入信号。是一根外部控制计数器工作的输入信号线，用作控制启动或中止定时/计数。对于 8253 这 6 种不同的工作方式，GATE 信号的有效方式也不同，可以是上升沿信号有效或是电平信号有效。

CLK：外部脉冲信号输入端 CLK_0 、 CLK_1 、 CLK_2 ，分别为计数器 0、计数器 1、计数器 2 的脉冲信号输入端。计数器对该引脚输入的脉冲信号进行计数，CLK 脉冲信号可以由系统时钟、系统时钟分频或者其他脉冲源提供。CLK 输入脉冲可以是均匀的、连续的或周期精确的信号，也可以是不均匀的、不连续的或非周期的信号。如果输入脉冲是精确的时钟信号，则 8253 起定时作用；如果输入脉冲是非周期的信号，则 8253 起计数作用。

OUT：定时时间到或计数减为零的向外输出信号端 OUT_0 、 OUT_1 、 OUT_2 ，分别为计数器 0、计数器 1、计数器 2 的脉冲信号输出端。无论 8253 的计数器工作在什么方式，当计数器减到 0 时，信号输出端 OUT 必定有一个电平或脉冲信号输出，用以指示定时时间到或者计数结

束。该信号可供 CPU 检测，也可以作为中断请求信号。

5.2.2 8253/8254 的工作方式

8253 共有 6 种工作方式，由方式控制字确定，同一芯片中的三个计数器可以通过初始化分别设定为不同的工作方式。各方式下的工作状态是不同的，输出的波形也不同，但大致相同。

(1) 方式 0：计数结束产生中断 (Interrupt on Terminal Count)

方式 0 为计数结束产生中断，工作时序如图 5-22 所示。

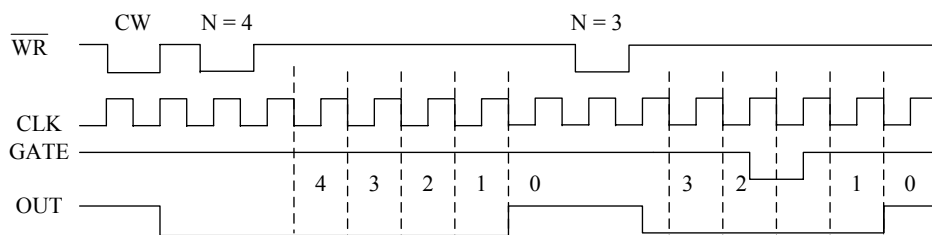


图 5-22 方式 0 时序

当计数器工作方式设置为方式 0 后，该计数器的 OUT 输出信号立即变为低电平，且计数过程中一直保持低电平。在经初值寄存器赋初值后，如图中初值为 $N=4$ ，开始计数，在每个 CLK 时钟下降沿，计数器进行减 1 计数，当计数减 1 减到 0 时，OUT 输出信号变为高电平，且一直保持到该计数器重新赋初值，或重新设定工作方式。该信号可以作为中断请求信号。

GATE 输入信号用于控制计数的过程。GATE 作用于计数器，GATE 为高电平时，允许计数；GATE 为低电平时，暂停计数。当 GATE 重新为高电平时恢复先前的计数过程。

方式 0 无自动装入计数初始值的功能，若要继续计数，则需要重新写入计数初始值。在计数期间，装入新的初始值，计数器会在初始值写入后重新开始计数。

(2) 方式 1：可编程单稳态触发器 (Hardware Retriggerable one-shot)

方式 1 为一种硬件触发启动，无自动重复计数方式，可在 OUT 输出单稳脉冲。工作时序如图 5-23 所示。

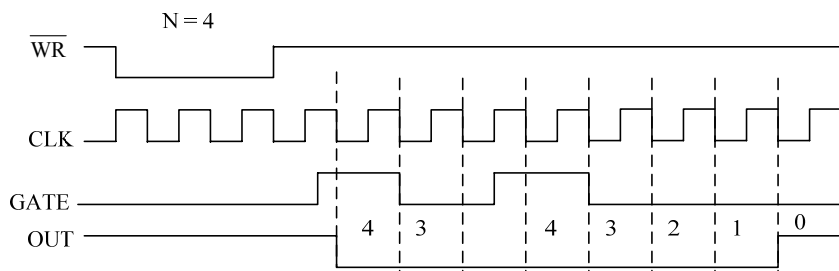


图 5-23 方式 1 时序

写入方式 1 的控制字后，OUT 输出高电平，一直保持。当 CPU 写入计数初值，等到 GATE 上升沿到达后，计数器被启动。此时，在 CLK 脉冲下降沿开始计数，OUT 输出低电平。在每个 CLK 时钟下降沿，计数器进行减 1 计数，当计数减 1 减到 0 时，OUT 输出信号变为高电平。故 OUT 输出一个宽度为计数值 N 个时钟周期的单稳脉冲。

计数结束后，若 GATE 上升沿到达，再次触发启动，则可以又以上次写入的初值开始，在下一个 CLK 下降沿开始计数，不需要重新写入初值，OUT 输出一个单稳脉冲。

在整个计数过程中，可以写入新的初值，这对正在进行的计数过程不影响。但当有新的 GATE 上升沿到达，则计数器将先前的新初值装入，从头开始计数。

（3）方式 2：分频器（Rate Generator）

方式 2 是具有自动装入计数常数、GATE 为电平触发方式的分频器，工作时序如图 5-24 所示。

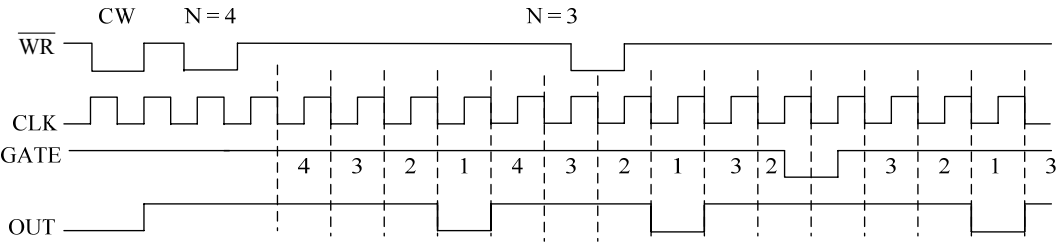


图 5-24 方式 2 时序

写入方式 2 控制字后，OUT 输出高电平。写入计数初值后，如果 GATE 为高电平，计数器开始对输入时钟 CLK 脉冲计数。OUT 输出信号在计数过程中一直是高电平，直到减 1 计数器减到 1 时，OUT 输出变为低电平，维持一个 CLK 时钟周期，OUT 又恢复输出高电平，同时计数器开始重新计数。

方式 2 可以自动连续工作，输出固定频率的脉冲信号。如果计数值为 N，则输出的脉冲信号频率为 $1/N$ 的时钟频率，故该方式被称为分频器。

方式 2 中，在计数过程中写入新的值，不影响正在进行的计数过程，从下个计数周期开始按新值计数。GATE 为低电平时，禁止计数器，并使 OUT 输出为高电平；GATE 为高电平时，计数器恢复写入的初值，开始计数。

（4）方式 3：方波发生器（Square Wave Mode）

方式 3 与方式 2 类似，也是一种具有自动装入计数常数 N 分频方波发生器，工作时序如图 5-25 所示。

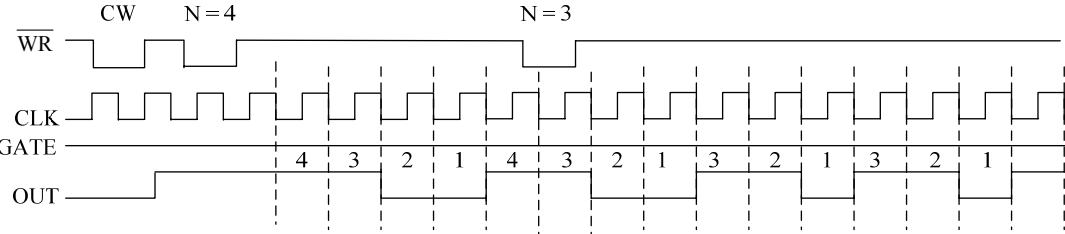


图 5-25 方式 3 时序

写入方式 3 控制字后，OUT 输出高电平。写入计数初值后，如果 GATE 为高电平，计数器开始对输入时钟 CLK 脉冲计数，当计数到计数值一半时，OUT 输出变为低电平，直到计数

为 0 时, OUT 输出又变为高电平。当计数值为偶数时, OUT 输出高、低电平都为 $N/2$ 个 CLK 周期的周期信号;当计数值为奇数时, OUT 输出高电平为 $(N+1)/2$ 个 CLK 周期、低电平为 $(N-1)/2$ 个 CLK 周期的周期信号,即 OUT 输出前半多一个 CLK 周期的高电平。

(5) 方式 4: 软件触发选通脉冲 (Software Triggered Strobe)

方式 4 是一种软件触发启动,无自动重复计数功能的计数方式,工作时序如图 5-26 所示。

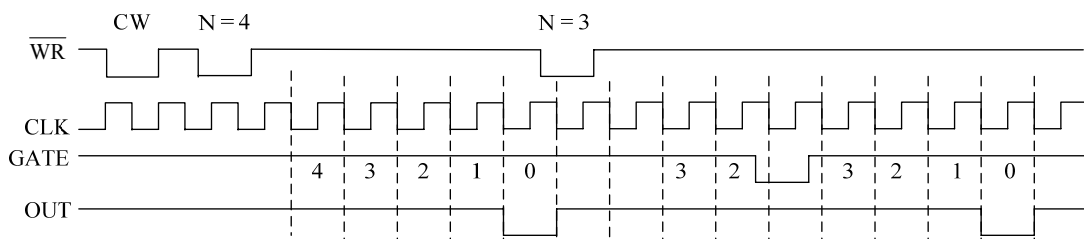


图 5-26 方式 4 时序

写入方式 4 控制字后, OUT 输出立即变为高电平,如果 GATE 为高电平,当写入计数初值后,在下一个 CLK 脉冲的下降沿开始减 1 计数,当计数器减到 0 时, OUT 输出为低电平,计数器停止计数。计数过程只执行一次,只有写入新的计数值,才开始新的计数。计数期间,如果写入新的计数值,则在下一个时钟的下降沿立即按新写入的计数值重新计数。当 GATE 为低电平时,禁止计数;当 GATE 为高电平时,从初值开始重新计数。

(6) 方式 5: 硬件触发选通脉冲 (Hardware Triggered Strobe)

方式 5 为硬件触发启动、无自动重复计数功能的计数方式,工作时序如图 5-27 所示。

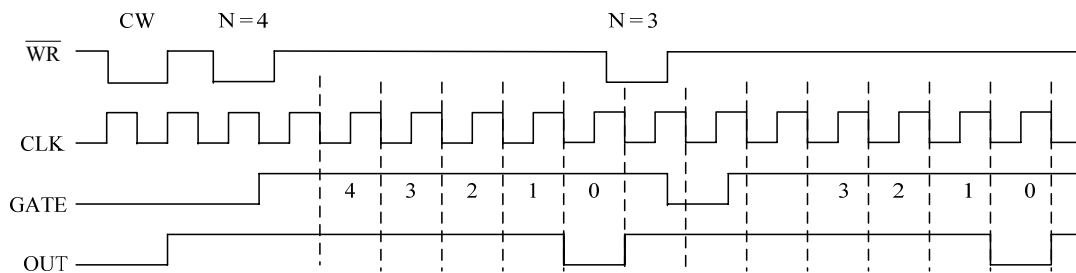


图 5-27 方式 5 时序

写入方式 5 控制字后, OUT 输出保持高电平,写入计数初值后,但不开始计数,当 GATE 出现上升沿后,启动计数器,在下一个 CLK 下降沿开始减 1 计数,当计数到 0 时, OUT 输出为低电平,经过一个 CLK 脉冲, OUT 输出恢复高电平,计数器停止。在计数过程中, GATE 重新出现上升沿时,将使计数器从初值开始重新计数。

OUT 输出脉冲宽度在正常计数情况下,如果写入的计数值为 N, OUT 输出维持 N 个 CLK 时钟周期的高电平, 1 个 CLK 时钟周期的低电平。

8253 的 6 种工作方式特点总结见表 5-4。

表 5-4 8253 的 6 种工作方式的特点及功能

工作方式	OUT 触发计数	OUT 终止计数	初始值自动装载	功能
0	高电平	低电平	无	计数（定时）中断
1	上升沿	无影响	无	单脉冲发生器
2	高电平或上升沿	低电平	有	频率发生器
3	高电平或上升沿	低电平	有	方波发生器
4	高电平	低电平	无	单脉冲发生器
5	上升沿	无影响	无	单脉冲发生器

5.2.3 8253/8254 的应用实例

1. 8253 应用编程

8253 的初始化编程就是对其工作方式的确定。在 8253 启动后，由 CPU 向 8253 的控制寄存器写入一个控制字，规定 8253 的工作方式、计数值以及计数所用的数制等。根据要求将计数初值写入 8253 相应的计数器。

（1）8253 初始化编程

8253 加电后的工作方式不确定，为了正常的工作，需要对芯片进行初始化。8253 的初始化编程包括向控制寄存器中写入方式控制字和向选定的计数器中写入计数初值。

1) 写入方式控制字。8253 的每个计数器内有一个 8 位控制寄存器，用来存放 CPU 写入的工作方式控制字，工作方式控制字格式如图 5-28 所示，该寄存器只能执行写入操作，不能执行读出操作。8253 内部的 3 个计数器在结构上相互独立，在使用时须对指定的计数器写入方式控制字，写入控制字的 I/O 地址相同，要求 $A_1A_0 = 11$ 。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
SC1	SC0	RW1	RW0	M ₂	M ₁	M ₀	BCD

图 5-28 8253 工作方式控制字

D₀：计数码选择。8253 提供丰富的计数制，该位用来决定计数器在减 1 计数过程中采用的是二进制还是十进制，0 表示采用二进制计数制，1 表示采用 BCD 计数制。

在 BCD 计数制下，写入计数值的范围为 0000~9999，最大值为 10^4 ，即十进制数的 10000，表示为 0000；在二进制计数制下，写入计数值的范围为 0000~FFFFH，最大值为 2^{16} ，即十进制数的 65536，表示为 0000H。

D₃~D₁：M₂、M₁、M₀ 为 8253 的工作方式的选择。8253 的每个计数器的 6 种工作方式如表 5-5 所示。

D₅、D₄：RW1 和 RW0 读写格式控制。计数值的读出或写入可按字节或字两种方式进行操作。D₅D₄=01，只读写低 8 位，高 8 位自动置 0；D₅D₄=10，只读写高 8 位，低 8 位自动置 0；D₅D₄=11，必须先读写低 8 位，后读写高 8 位；D₅D₄=00，锁存命令，用于把当前计数值存入输出锁存器，供以后读取。

表 5-5 8253 工作方式选择

M ₂	M ₁	M ₀	工作方式选择
0	0	0	方式 0
0	0	1	方式 1
×	1	0	方式 2
×	1	1	方式 3
1	0	0	方式 4
1	0	1	方式 5

D₇、D₆：计数器选择。D₇D₆ = 00，选择计数器 0；D₇D₆ = 01，选择计数器 1；D₇D₆ = 10，选择计数器 2；D₇D₆ = 11，在 8253 中为非法编码，在 8254 中用于读回命令。

2) 写入计数初值。每个计数器都有对应的计数器 I/O 地址用于读写计数初值。读写计数初值须按照方式控制字规定的读写格式进行读写。

计数初值 N 可以由输入时钟频率 f_{clk} 和输出波形 f_{out} 频率计算得到：

$$\text{计数初值 } N = \frac{\text{输入时钟频率 } f_{\text{clk}}}{\text{输出波形 } f_{\text{out}}} = \frac{T_{\text{out}}}{T_{\text{clk}}}$$

计数器是在减 1 后再判断是否为 0，则写入初值为 0 时，表示写入的是最大值。计数初值的取值范围是：0000H~FFFFH（二进制）或 0000~9999（BCD 码），其中，写入 0000H 表示计数初值为 65536，写入 0000 表示计数初值为 10000。

例 5-3 已知某个 8253 的计数器 0、1、2 和方式控制的端口地址为：40H~43H，要求设置计数器 2 对 CLK₂ 的外部输入脉冲进行计数，计数满 1024 个后向 CPU 发中断请求，工作方式 0，采用二进制计数，先低后高写入计数初值，试写出相应的初始化程序。

```
MOV AL, 0B0H      ;方式控制字：0B0H = 10110000B
OUT 43H, AL        ;写入方式控制端口：43H
MOV AX, 400H       ;计数初值：1024 = 400H
OUT 42H, AL        ;写入计数器 2，42H，写入低字节计数初值
MOV AL, AH
OUT 42H, AL        ;写入高字节计数初值
```

(2) 8254 读回命令

8254 较 8253 的改进主要有：8254 的计数频率更高；8254 多了一个读回命令。这个读回命令（写入控制寄存器）可以使 3 个计数器的计数初值和状态锁存，向 CPU 返回一个状态字。其格式如图 5-29 所示。

8254 中每个计数器都有一个状态字，可通过读回命令将其锁存，然后由 CPU 读取。状态字的格式如图 5-30 所示。

格式中 D₅~D₀ 位为写入此计数器的控制字的相应部分。D₇ 位反映该计数器的输出引脚的现行状态，输出为高电平，D₇ = 1；输出为低电平，D₇ = 0。D₆ 位反映计数初值寄存器中计数初值是否已装入减 1 计数器中，若最后写入计数初值寄存器的计数值已装入减 1 计数器，则 D₆ = 0，表示可读计数；若计数初值寄存器的计数值未装入减 1 计数器，则 D₆ = 1，表示无效

计数，读取的计数值将不反映刚才写入的那个新计数值。

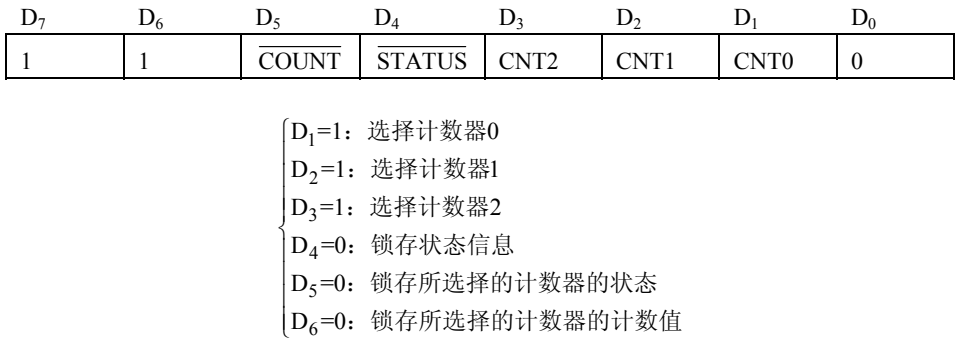


图 5-29 8254 的读回命令

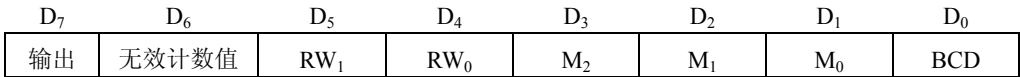


图 5-30 8254 状态字的格式

2. 8253 应用举例

在 PC 系列机中，使用了 1 片 8253/8254，其内部 3 个计数器分别用于：日历时钟，包括年月日、时分秒，精确到 0.01s；动态存储器刷新；声音，根据不同的音频产生不同的频率信号，驱动扬声器。其连接如图 5-31 所示。

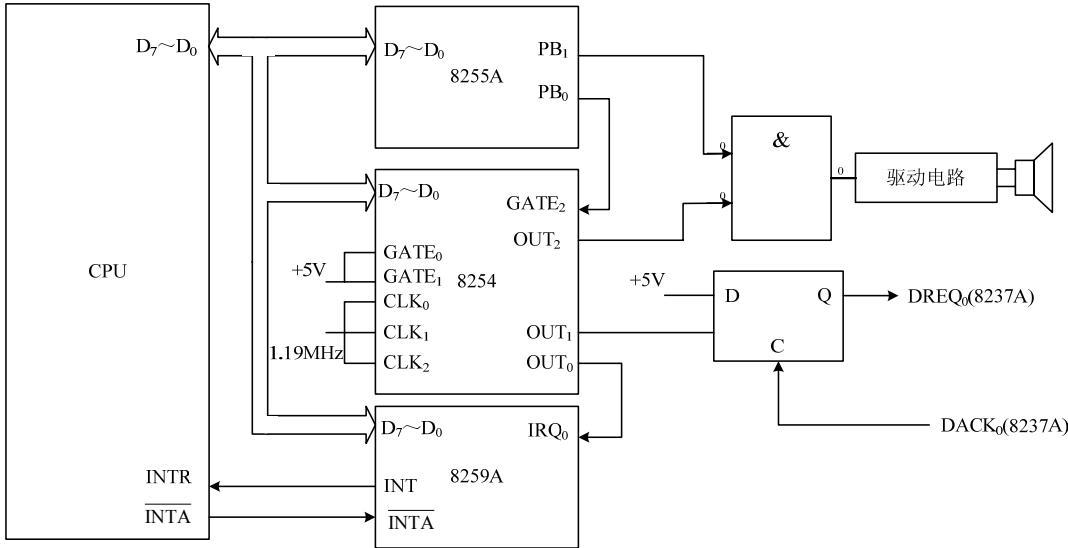


图 5-31 8254 的接口电路

8253/8254 的 I/O 地址范围是 040H~05FH，计数器 0、计数器 1、计数器 2 的计数器端口地址分别为 40H、41H 和 42H，控制字的端口地址为 43H。3 个计数器的时钟信号 CLK₀~CLK₂ 由系统时钟分频后得到 1.19318MHz 提供。

(1) 计数器 0

计数器 0 是一个产生实时时钟信号的系统计时器。利用它系统可以完成日历时钟计数、写操作后的电动机的自动延迟停机以及为用户提供定时中断调用。用户可以使用这个中断调用运行自己的中断处理程序。

以下为系统 ROM-BIOS 对计数器 0 的初始化编程:

```
MOV  AL, 36h          ;计数器 0 为方式 3, 采用二进制计数, 先低后高写入计数值
OUT  43H, AL          ;写入方式控制字
MOV  AL, 0            ;计数初值 0
OUT  40H, AL          ;写入低字节计数初值
MOV  AL, AH
OUT  40H, AL          ;写入高字节计数初值
```

由上图及初始化程序可知, $GATE_0$ 接+5V, CLK_0 输入为 1.19318MHz 方波, 采用工作方式 3, 减 1 计数器的计数初值为 0 (65536), 输出信号 OUT_0 接到系统上 8259A 的 IRQ_0 中断请求输入端。因而 OUT_0 输出频率为 $1.19318/65536 = 18.206\text{Hz}$ 的方波信号, 每隔 54.925439ms 产生一次中断请求。根据该中断请求, 系统直接调用固化在 BIOS 中的 08 号中断处理程序, 实现了日时钟计时功能, 即记录 18 次中断就是时间经过了 1 秒, 调用命令为 INT 8H。

(2) 计数器 1

计数器 1 专门用作 DRAM 定时刷新请求。PC/XT 机要求在 2ms 内进行 128 次刷新操作, PC/AT 机要求在 4ms 内进行 256 次刷新操作, 由此可知, 每隔 $2\text{ms}/128 = 15.6\mu\text{s}$ 必须进行一次刷新。这样计数器 1 的 $GATE_1$ 接+5V, CLK_1 输入为 1.19318MHz 方波, 采用工作方式 2, 减 1 计数器 CE 的预置计数初值为 18, 因此在 OUT_2 端输出一负脉冲, 周期为 $18/1.19318 = 15.084\mu\text{s}$, 该信号作为 D 触发器的触发时钟信号, 使每隔 $15.084\mu\text{s}$ 产生一个正脉冲, 周期性地对 DRAM 进行刷新, 满足刷新要求, 初始化程序如下:

```
MOV  AL, 54H          ;计数器 1 采用工作方式 2, 二进制计数, 写入低 8 位
OUT  43H, AL
MOV  AL, 18           ;计数初值为 18
OUT  41H, AL
```

(3) 计数器 2

PC 系列机利用计数器 2 的输出控制扬声器的发声, 作为机器的报警信号或伴音信号。计数器 2 的时钟输入 CLK_2 输入为 1.19318MHz 方波, 采用工作方式 3, 系统中初值寄存器的内容预置为 0533H (十进制 1331), 当 $GATE_2$ 为高电平时, OUT_2 输出频率为 $1.19318\text{MHz}/1331 = 900\text{Hz}$ 的方波, 该方波信号经放大和滤波后可以推动扬声器发声。由图 5-31 可知, 送到扬声器的信号受到从并行接口芯片 8255A 来的双重控制, 8255A 的 PB_0 位 (即 I/O 端口地址 61H 的 D_0 位) 接计数器 2 的 $GATE_2$ 引脚, 计数器 2 的 OUT_2 信号和 8255A 的 PB_1 同时作为与门的输入。 PB_0 和 PB_1 位可由程序决定为 0 或 1, 只有 PB_0 和 PB_1 都是 1 时, 才能使扬声器发声。改变计数初值, 可以改变 OUT_2 输出信号的频率, 从而可以改变扬声器发出的音调。实现扬声器发声的程序如下:

```
                ;扬声器发声初始化程序
Speaker        PROC
```

```

        PUSH    AX
        MOV     AL, 0B6H      ;计数器 2 采用工作方式 3, 先低后高写入 16 位计数初值
        OUT     43H, AL
        POP     AX
        OUT     42H, AL      ;写入计数初值
        MOV     AL, aH
        OUT     42, AL
        RET
Speaker    ENDP
        ;扬声器打开子程序
SpeakerOn  PROC
        PUSH    AX
        IN      AL, 61H      ;读取 61H 端口的控制信息
        OR      AL, 03H      ;D1D0 = PB1PB0 = 11B
        OUT     61H, AL      ;控制打开发声
        POP     AX
        RET
SpeakerOn  ENDP
        ;扬声器关闭子程序
SpeakerOff PROC
        PUSH    AX
        IN      AL, 61H
        AND     AL, 0FCH     ;D1D0 = PB1PB0 = 00B
        OUT     61H, AL      ;控制关闭发声
        POP     AX
        RET
SpeakerOff ENDP

```

(4) 软件延时

在编写程序中经常需要进行延时，有多种延时的方法：可以利用反复循环进行延时，还可以利用读取日时钟的时间进行延时等。但是这些方法都有一些缺点，下面介绍 BIOS 的 15H 号中断调用的 86 号子功能，该中断调用可以使用户很好地实现较短时间的延时。

中断调用命令：INT 15H，子功能号：AH = 86H；

入口参数：CX.DX = 延时时间的微秒值（CX 是高 16 位、DX 是低 16 位），

出口参数：标志位 CF = 0，表示成功，执行了延时，CF = 1，表示不成功，未执行延时。

虽然以微秒为单位，但因为实时时钟的最小单位是 976.5625μs (1/1024Hz)，所以，该功能调用的实际延时是 976μs 的整数倍。例如要延时 1s，可以用以下程序：

```

MOV     AH, 86H
MOV     CX, 15              ;1000000 微秒的高 16 位
MOV     DX, 16960           ;1000000 微秒的低 16 位

```

INT 15H

例 5-4 用 8254 作为某数据采集系统的定时器,每隔 10ms 用中断方式采集一次数据,已知输入时钟频率为 10kHz,8259 端口地址为 20H~21H,中断类型为 13H,8254 端口地址为 40H~43H,请为 8259 和 8254 编制初始化程序。

程序源代码:

```

INT0    EQU    20H
INT1    EQU    21H
D8254   EQU    43H
D8254A  EQU    40H
STACK SEGMENT STACK
    STA  DW 50 DUP(?)
    TOP  EQU  LENGTH STA
STACK ENDS
CODE SEGMENT
    ORG 1000H
    ASSUME CS:CODE,DS:CODE,SS:STACK
START:    ;初始化
    ;设置中断的入口地址
    XOR    BX,BX
    PUSH   BX
    POP     DS        ;将 DS 清零
    CLI     ;关中断
    ;如果设置 8259 的起始中断向量号为 10
    ;以下 4 个中断向量分别分配给 IRQ0、IRQ1、IRQ2、IRQ3
    LEA    AX,IRQ0_INTERRUPT    ;中断向量 10H
    MOV     DS:WORD PTR[64],AX
    MOV     DS:WORD PTR[66],CS
    NOP
    LEA    AX,IRQ1_INTERRUPT    ;中断向量 11H
    MOV     DS:WORD PTR[68],AX
    MOV     DS:WORD PTR[70],CS
    NOP
    LEA    AX,IRQ2_INTERRUPT    ;中断向量 12H
    MOV     DS:WORD PTR[72],AX
    MOV     DS:WORD PTR[74],CS
    NOP
    LEA    AX,IRQ3_INTERRUPT    ;中断向量 13H
    MOV     DS:WORD PTR[76],AX
    MOV     DS:WORD PTR[78],CS

```

```

NOP
PUSH CS                ;将 CS 中的内容传到 DS
POP DS
;初始化 8259
CLI                    ;关中断
;ICW1: 特征 A0 = 0, D4 = 1
;边沿触发要求上升沿后保持高电平
MOV AL, 00010011B      ;边沿触发, 单片使用, 需要 ICW4
OUT INT0, AL
JMP $+2                ;延时
;ICW2: 特征 A0 = 1 按顺序判断
MOV AL, 00010000B      ;中断矢量从 10H 号开始
OUT INT1, AL
JMP $+2
;ICW4: 特征 A0 = 1 按顺序判断, 需要设置 ICW4
;EOI, 全嵌套方式, 非缓冲
MOV AL, 00000001 ;采用全嵌套, 非缓冲, 非自动结束, 16 位微机处理
OUT INT1, AL
JMP $+2
;8259 操作命令
;*****
;
;OCW1: 特征 A0 = 1 按顺序判断
;
; 初始化结束后, 只有 OCW1 要求 A0=1
;
; 其他操作命令 OCW2, OCW3 都要求 A0=0
;*****
;
MOV DX,INT1
MOV AL,0F0H            ;假设只用 4 个可屏蔽中断, 4 个中断全部开放
OUT DX,AL
MOV CX,0FFFH
LOOP $
;*****
;
;OCW2: 特征 A0 = 0, D4=0, D3=0
;*****
;
MOV DX,INT0
MOV AL,20H             ;00100000B, 非特殊 EOI 结束中断
OUT DX,AL
MOV CX,0FFFH
LOOP $
;*****
;

```

;OCW3: 特征 A0=0, D4=0, D3=1

;初始化 8254

MOV AL, 00110001B ;8254 的 0 号计数器, 工作方式 0, 16 位数, 二进制

OUT D8254, AL

MOV AX, 100 ;为计数器 0 送计数初值, 先送低 8 位

OUT D8254A, AL

MOV AL, AH

OUT D8254A, AL ;为计数器 0 送计数初值高 8 位

;中断服务程序

IRQ0_INTERRUPT PROC NEAR

PUSH AX

PUSH DX

STI ;开中断

;.....不处理返回

;

CLI ;关中断

POP DX

POP AX

IRET

IRQ0_INTERRUPT ENDP

IRQ1_INTERRUPT PROC NEAR

PUSH AX

PUSH DX

STI ;开中断

;.....不处理返回

;

CLI ;关中断

POP DX

POP AX

IRET

IRQ1_INTERRUPT ENDP

IRQ2_INTERRUPT PROC NEAR

PUSH AX

PUSH DX

STI ;开中断

;.....不处理返回

;

```

        CLI                      ;关中断
        POP    DX
        POP    AX
        IRET
IRQ2_INTERRUPT ENDP
IRQ3_INTERRUPT PROC NEAR
        PUSH   AX
        PUSH   DX
        STI                      ;开中断
        MOV    DX,INT0
        MOV    AL,20H            ;结束中断
        OUT    DX,AL
        CLI                      ;关中断
        POP    DX
        POP    AX
        IRET
IRQ3_INTERRUPT ENDP
CODE    ENDS
END     START

```

5.3 并行接口技术

CPU 通过接口来实现与外设间的数据传输。并行传输是数据在多根传输线上一次以 8 位或 16 位为单位进行传输。与串行传输相比，并行传输需要较多的传输线，成本较高，但传输速度快，因此，并行通信总是用在数据传输率要求较高，而传输距离较短的场合。

实现并行传输的接口称为并行接口，并行接口有不可编程并行接口和可编程并行接口。不可编程接口的工作方式和功能由硬件电路来设定，控制比较简单，要改变其功能必须改变硬件电路，不能用软件编程的方法改变。可编程接口的工作方式和功能可以用软件编程的方法改变，具有极大的灵活性。

8255A 是 Intel 公司生产的一种通用的可编程并行 I/O 接口芯片，它的通用性强，在使用中可利用软件编程来实现其功能。8255A 内部有 3 个 8 位的数据口（A 口、B 口和 C 口），3 个数据口均可用来输入或输出。通过 8255A，CPU 可以和大多数并行传输的外设直接连接，是应用最广的典型可编程并行接口芯片。

5.3.1 8255A 的内部结构和外部引脚

1. 8255A 的内部结构

由图 5-32 所示 8255A 内部结构框图可知，它由数据总线缓冲器、读/写控制逻辑、A 组和 B 组控制电路以及 3 个数据端口（A 口、B 口和 C 口）等组成。

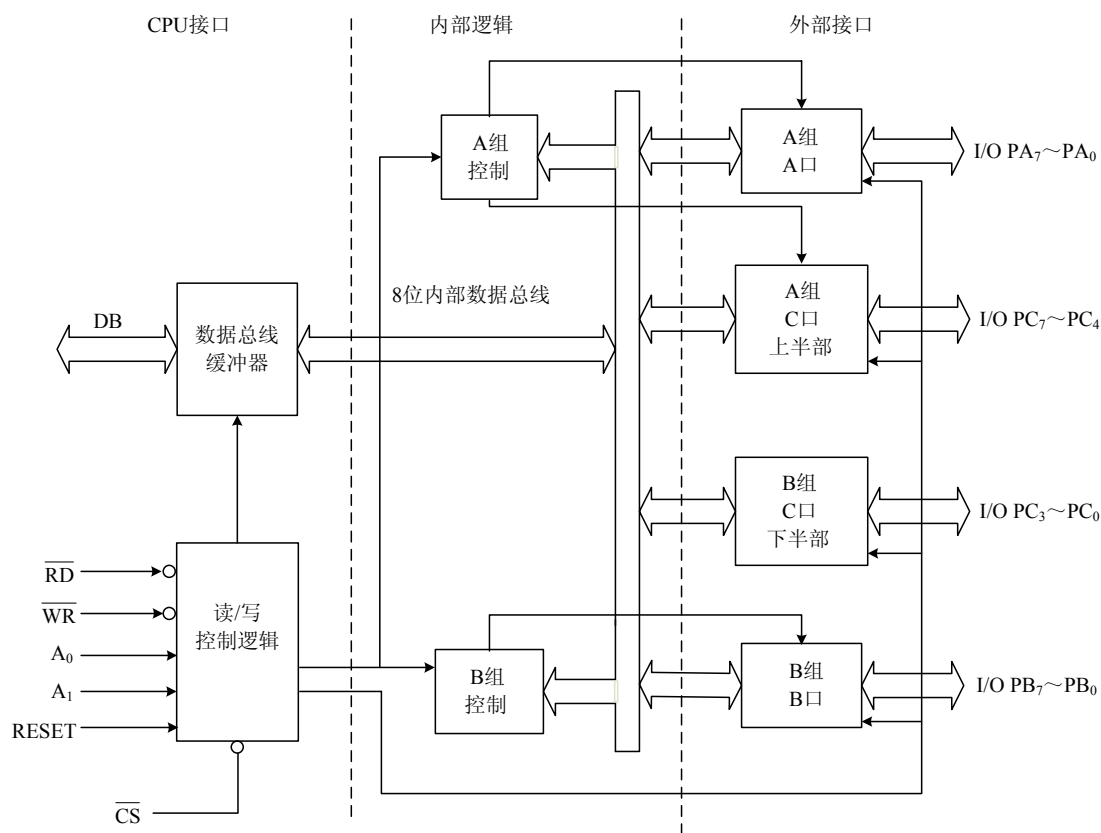


图 5-32 8255A 内部结构框图

(1) 数据总线缓冲器

8 位双向三态缓冲器，用于连接系统数据总线和 8255A 内部总线。CPU 通过执行输入/输出指令来实现对缓冲器发送或接收数据。8255A 控制字和状态字也是通过该缓冲器传输。

(2) 读/写控制逻辑

8255A 内部的控制电路，用来管理数据信息、控制字和状态字的传输。读写逻辑接收来自 CPU 的控制信号，包括读信号 $\overline{RD}$ 、写信号 $\overline{WR}$ 和芯片内部寄存器寻址信号 A_1A_0 等，向 A 组和 B 组控制电路发送命令。

(3) 数据端口（A 口、B 口和 C 口）

8255A 内部有 3 个 8 位输入/输出端口 A 口、B 口和 C 口。每个端口都有一个数据输入寄存器和一个数据输出寄存器，输入时端口有三态缓冲器的功能，输出时端口有数据锁存器功能，每个端口都可以由程序设定为各种不同的工作方式。

在使用中，A 口和 B 口经常作为独立的输入或输出端口，C 口可配合 A 口和 B 口的工作用作一个 8 位数据端口。C 口通过控制命令被分为 2 个 4 位端口，每个 4 位端口含 1 个 4 位的输入缓冲器和 1 个 4 位的输出缓冲器/锁存器，它们分别用来为 A 口和 B 口提供控制信号和状态信号。

(4) A 组和 B 组控制电路

8255A 的 3 个数据端口分为 A 组和 B 组两组。A 口和 C 口的高 4 位构成 A 组，由 A 组控

制电路进行控制；B 口和 C 口的低 4 位构成 B 组，由 B 组控制电路进行控制。这两组控制电路根据 CPU 发出的方式选择控制字来控制 8255A 的工作方式，每个控制组都接收来自读/写控制逻辑的命令，接收来自内部数据总线的控制字，并向与其相连的端口发出适当的控制信号。

2. 8255A 的外部引脚

如图 5-33 所示为 8255A 外部引脚图，8255A 是具有 40 个引脚的双列直插封装集成芯片，各引脚功能如下所述：

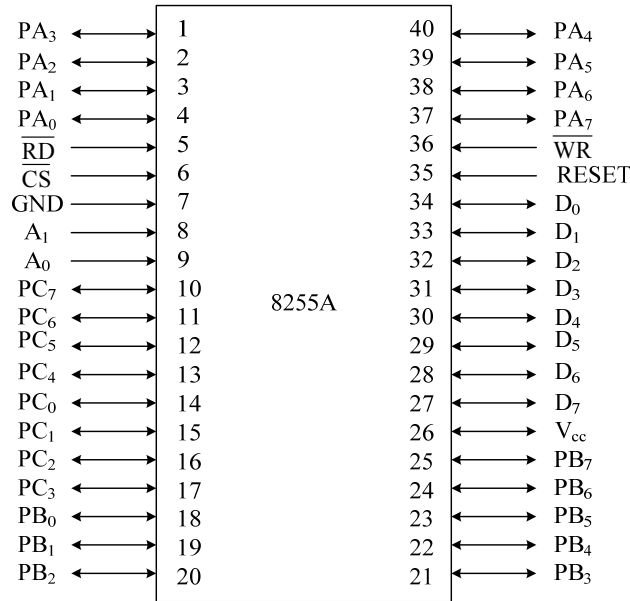


图 5-33 8255A 外部引脚图

D₇~D₀：双向数据线，与系统数据总线相连，供 CPU 对 8255A 进行读/写操作，进行信息交换。

$\overline{CS}$ ：片选线，输入低电平有效。当 $\overline{CS}=0$ 时，表示 CPU 选中 8255A 芯片，可对 8255A 进行读/写操作；当 $\overline{CS}=1$ 时，表示 CPU 未选中 8255A 芯片。

A₁~A₀：地址输入信号，用于选择 8255A 内部 3 个口和控制寄存器。当 $\overline{CS}$ 为低电平有效时，选中 8255A 芯片，然后通过 A₁ A₀ 组合编码决定选中的口或控制寄存器，如表 5-6 所示。

表 5-6 A₁A₀ 组合编码

A ₁	A ₀	寄存器选择
0	0	A 口
0	1	B 口
1	0	C 口
1	1	控制寄存器

$\overline{RD}$ 和 $\overline{WR}$ ：读/写控制输入信号端，低电平有效。当 $\overline{CS}=0$ ， $\overline{RD}=0$ 时，表示 CPU 可对 8255A 进行读操作，通常接系统总线的 $\overline{IOR}$ 信号；当 $\overline{CS}=0$ ， $\overline{WR}=0$ 时，表示 CPU 可对 8255A

进行写操作，通常接系统总线的 $\overline{\text{IOW}}$ 。

RESET: 复位输入信号。高电平有效，当此端输入为高电平时，可以使 8255A 复位。复位后，8255A 的 A 口、B 口和 C 口均被定义为输入状态。

PA₇~PA₀: A 口的 8 位输入/输出信号线。这 8 位信号线由编程方式决定是工作于输入、输出还是双向。

PB₇~PB₀: B 口的 8 位输入/输出信号线。这 8 条信号线由编程方式决定是工作于输入还是输出。

PC₇~PC₀: C 口的 8 位信号线。根据 8255A 的工作方式，可将 C 口的这 8 位信号线用作输入或输出信号线，也可以用作控制信号的输出线或状态信号的输出线。

8255A 的 3 个数据端口既可以写入数据又可以读出数据，控制端口只能写入命令而不能读出，读/写控制信号 ($\overline{\text{RD}}/\overline{\text{WR}}$) 和端口选择信号 ($\overline{\text{CS}}$ 、A₁ 和 A₀) 的状态组合可以实现 A、B、C 3 个端口和控制端口的读/写操作。8255A 的端口分配及读/写功能见表 5-7。

表 5-7 8255A 读、写功能表

$\overline{\text{CS}}$	$\overline{\text{WR}}$	$\overline{\text{RD}}$	A ₁	A ₀	功能
0	0	1	0	0	数据写入 A 口
0	0	1	0	1	数据写入 B 口
0	0	1	1	0	数据写入 C 口
0	0	1	1	1	命令写入控制寄存器
0	1	0	0	0	读取 A 口数据
0	1	0	0	1	读取 B 口数据
0	1	0	1	0	读取 C 口数据
0	1	0	1	1	无操作

5.3.2 8255A 的工作方式

8255A 共有 3 种工作方式：基本输入输出方式、选通输入输出方式和双向选通输入输出方式。

1. 方式 0: 基本输入输出方式 (Basic Input/Output)

方式 0 是一种基本输入输出方式。在这种工作方式下，不需要应答式的联络信号。A 口、B 口和 C 口的高 4 位、低 4 位两组都可以工作在基本输入输出方式。

当 8255A 工作在方式 0 时，8255A 具有两个 8 位端口和两个 4 位端口，任何一个端口都可以由工作方式控制字选择设置为输入端口或输出端口；输出具有锁存能力，输入只起缓冲作用，不对外设输入的数据进行锁存；每个端口都可由 CPU 用简单的输入或输出指令来与外设进行数据交换，因此方式 0 可用于无条件输入输出方式；方式 0 也可用于查询输入输出方式，此时 A 口和 B 口可用作数据端口，而 C 口的某些位作为控制位和状态位。

2. 方式 1: 选通输入输出方式 (Strobe Input/Output)

方式 1 是一种单向选通输入输出方式，借助一种应答联络信号进行输入输出。8255A 只有 A 口和 B 口可以在这种工作方式下用作数据输入输出端口，每个输入输出端口都要利用 C 口

的某些位作为控制信息位或状态信息位。如果一个端口工作于方式 1，则剩下的 13 位都可以工作于方式 0，A 口还可以工作于方式 2。

方式 1 中设置了专用的中断请求和联络信号线，因此，这种方式通常用于查询传送或中断传送数据，对输入和输出的数据都进行锁存。

(1) 方式 1 的输入

当 A 口和 B 口工作于方式 1 的输入时，其端口的内部结构及引脚如图 5-34 所示。

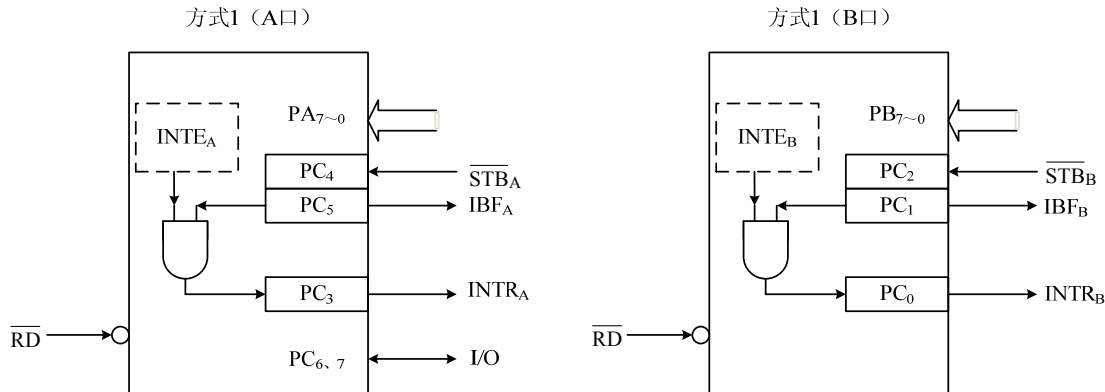


图 5-34 方式 1 输入时 C 口对 A、B 口的控制

方式 1 端口有 3 个控制信号线，用作联络信号。

STB (Strobe Input): 选通输入信号，低电平有效。此信号由外设产生，当有效时，将外设输入的数据写入 8255A 的端口的数据锁存器。当 A 口用作输入时，PC₄ 为 STB_A 信号线；当 B 口用作输入时，PC₂ 为 STB_B 信号线。

IBF (Input Buffer Full): 输入缓冲器满信号，高电平有效。这是 8255A 输出的联络信号，是对 STB 信号的应答。当有效时，表示外设发送给 CPU 的数据已经锁存在输入锁存器。当 A 口用作输入时，PC₅ 为 IBF_A 信号线；当 B 口用作输入时，PC₁ 为 IBF_B 信号线。STB 为低电平时，IBF 置为高，CPU 读取数据后 IBF 被清除为低无效。

INTR (Interrupt Request): 中断请求信号，高电平有效。这是 8255A 向 CPU 提出中断请求的输出信号，要求 CPU 读取外设数据。当 STB = 1，IBF = 1，允许中断请求时被置为有效。当 A 口用作输入时，PC₃ 为 INTR_A 信号线；当 B 口用作输入时，PC₀ 为 INTR_B 信号线。

INTE (Interrupt Enable): 中断允许信号。A 口用 PC₄ 的按位置位/复位控制，B 口用 PC₂ 的按位置位/复位控制。只有当 PC₄ 或 PC₂ 置 1 时，才允许对应的端口送出中断请求。

方式 1 输入时的工作时序如图 5-35 所示。

下面以 A 口为例，对方式 1 的工作过程描述如下：

当外设将数据送至 8255A 的端口数据线，并发出选通信号 STB，8255A 在 STB 的下降沿将数据线的的数据锁存在 A 口的数据锁存器中；其后将 IBF 变成高电平，输出给外设，阻止外设输入新的数据；在 INTE = 1，允许中断的条件下，当 IBF 有效时，在 STB 信号无效后，INTR 变为高有效，向 CPU 发出中断请求信号，通知 CPU 输入的数据已锁存在数据锁存器中；CPU 响应中断，执行 IN 命令，发出读信号 RD，CPU 读取数据，RD 的下降沿清除中断请求，读取结束，RD 的上升沿则使 IBF 复位为零。外设检测到 IBF 为零后，便可以发送下一数据。

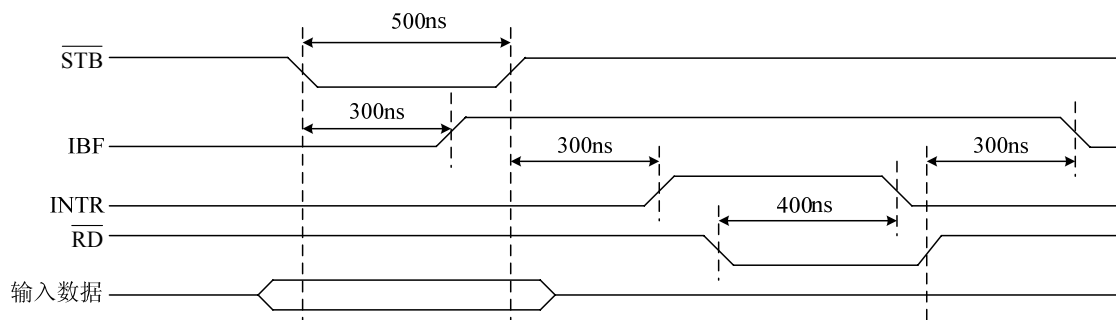


图 5-35 A 口和 B 口方式 1 数据输入的时序图

(2) 方式 1 的输出

当 A 口和 B 口工作于方式 1 的输出时，其端口的内部结构及引脚如图 5-36 所示。

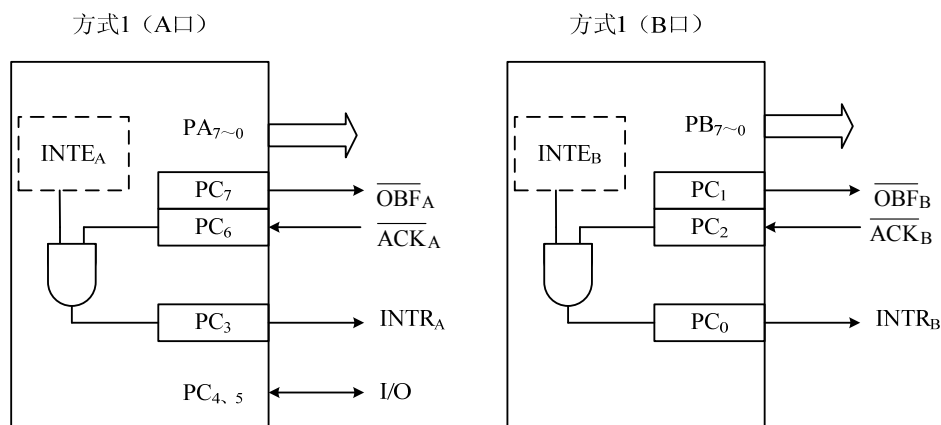


图 5-36 A 口和 B 口方式 1 的内部结构及引脚

方式 1 端口有 3 个控制信号线，用作联络信号。

$\overline{\text{OBF}}$ (Output Buffer Full): 输出缓冲器满信号，低电平有效。此信号是由 8255A 输出给外设的控制信号。当有效时，表示 CPU 已将数据输出到指定的输出端口，外设可以取走数据。当 A 口用作输出时，PC₇ 为 $\overline{\text{OBF}}$ 信号线；当 B 口用作输出时，PC₁ 为 $\overline{\text{OBF}}$ 信号线。CPU 的输出信号 $\overline{\text{WR}}$ 的上升沿使其有效， $\overline{\text{ACK}}$ 信号使其无效。

$\overline{\text{ACK}}$ (Acknowledge): 响应信号，低电平有效。这是外设对 $\overline{\text{OBF}}$ 信号的应答。当有效时，表示 8255A 的端口数据已由外设取走。当 A 口用作输出时，PC₆ 为 $\overline{\text{ACK}}$ 信号线；当 B 口用作输出时，PC₂ 为 $\overline{\text{ACK}}$ 信号线。

INTR (Interrupt Request): 中断请求信号，高电平有效。这是 8255A 向 CPU 提出中断请求的输出信号，要求 CPU 继续发送数据。当 $\overline{\text{OBF}} = 1$ ， $\overline{\text{ACK}} = 1$ ，允许中断请求时被置为有效。 $\overline{\text{WR}}$ 信号的下降沿使其无效。

INTE (Interrupt Enable): 中断允许信号。A 口用 PC₆ 的按位置位/复位控制，B 口用 PC₂ 的按位置位/复位控制。

方式 1 输出时的工作时序如图 5-37 所示。

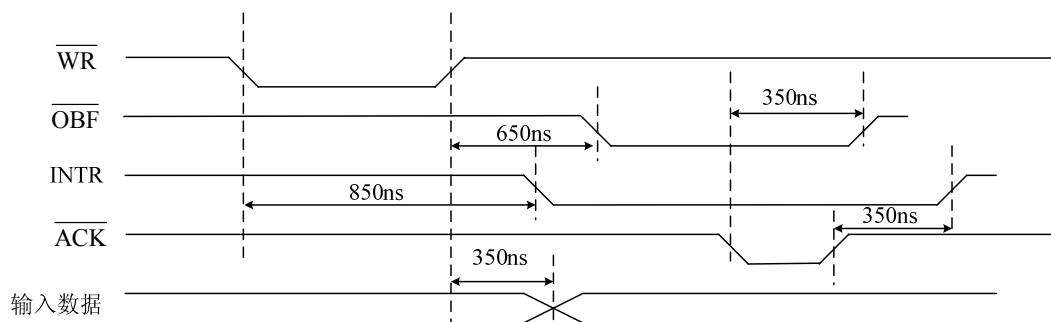


图 5-37 A 口和 B 口方式 1 数据输出的时序图

下面以 A 口为例，对方式 1 的工作过程描述如下：

在中断方式下，CPU 响应中断请求，执行 OUT 命令，发出写信号 $\overline{WR}$ ，将输出的数据送入指定的输出数据锁存器中锁存；当 CPU 输出结束时， $\overline{WR}$ 信号的上升沿将 $\overline{OBF}$ 变为有效，表示输出缓冲器满，通知外设接收数据；当外设开始接收数据后，发出 $\overline{ACK}$ 信号，作为应答信号； $\overline{ACK}$ 信号的下降沿使 $\overline{OBF}$ 变为无效， $\overline{ACK}$ 的上升沿使 INTR 变为有效，在允许中断请求条件下，向 CPU 发出中断请求，通知 CPU 数据已由外设接收，可以向 8255A 输出下一个数据。

3. 方式 2: 双向选通输入输出方式 (Bi-directional Bus Strobe Input/Output)

方式 2 是一种双向选通输入输出方式，这种工作方式外设既可以发送数据，又可以接收数据。

只有 A 口可以工作在方式 2，利用 C 口的 $PC_7 \sim PC_3$ 这 5 位作为 A 口的信号线；此时，B 口可以工作在方式 1 或方式 0。当 8255A 工作在方式 2 时，数据传送可用程序查询或中断方式实现。

当 8255A 的 A 口工作于方式 2 时，其端口的内部结构及引脚如图 5-38 所示。

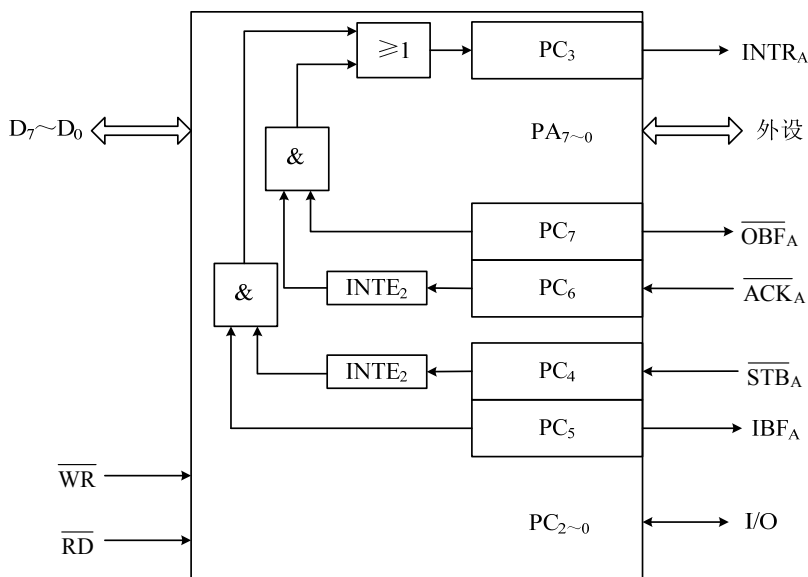


图 5-38 A 口方式 2 引脚信号

$\overline{\text{STB}}$ (Strobe Input): 选通输入信号, 低电平有效。此信号由外设产生, 当有效时, 将外设输入的数据写入 8255A 的端口的数据锁存器。

$\overline{\text{IBF}}$ (Input Buffer Full): 输入缓冲器满信号, 高电平有效。这是 8255A 输出的控制信号, 是对 $\overline{\text{STB}}$ 信号的应答。当有效时, 表示外设发送给 CPU 的数据已经锁存在输入锁存器。

$\overline{\text{OBF}}$ (Output Buffer Full): 输出缓冲器满信号, 低电平有效。此信号是由 8255A 输出给外设的控制信号。当有效时, 表示 CPU 已将数据输出到 A 口的输出锁存器, 但还没有传送到端口引脚 $\text{PA}_7 \sim \text{PA}_0$ 上。

$\overline{\text{ACK}}$ (Acknowledge): 响应输入信号, 低电平有效。 $\overline{\text{ACK}}$ 的下降沿使能 A 口三态输出锁存缓冲器, 送出数据, $\overline{\text{ACK}}$ 的上升沿作为数据已输出的响应信号。其他时间, A 口输出缓冲器处于高阻状态。

INTR (Interrupt Request): 中断请求信号, 高电平有效。用作输入或输出数据时的中断请求信号。

INTE (Interrupt Enable): 中断允许信号。A 口输出的中断允许由 PC_6 的按位置位/复位控制, A 口输入的中断允许由 PC_4 的按位置位/复位控制。

方式 2 的数据输入过程与方式 1 的输入方式工作过程一样, 可以参考方式 1 的输入方式工作过程; 方式 2 的数据输出过程与方式 1 的输出方式工作过程有些区别, 方式 2 的数据输出过程: 在中断方式下, CPU 响应中断请求, 执行输出指令, 向 8255A 的端口 A 写入一个新的数据, 发出写信号 $\overline{\text{WR}}$, 清除中断请求信号 INTR , 同时使端口 A 输出缓冲器满信号 $\overline{\text{OBF}}$ 变为有效低电平以通知外设读取数据, 外设取走数据后提供响应信号 $\overline{\text{ACK}}$, 以清除 $\overline{\text{OBF}}$ 有效信号并置位 INTR , 向 CPU 再次发出中断请求, 从而开始下一个数据传输过程。方式 2 的输出过程与方式 1 的输出过程的区别是: 数据输出时 8255A 不是在 $\overline{\text{OBF}}$ 有效时向外设输出数据, 而是在外设提供响应信号 $\overline{\text{ACK}}$ 时才送出数据。

5.3.3 8255A 的应用实例

1. 8255A 初始化编程

8255A 是通用可编程并行接口芯片, 8255A 的应用编程, 就是用户在使用 8255A 时, 要根据实际情况用软件来设定各端口的工作方式, 选择需要的功能。

8255A 的 A 口、B 口和 C 口的工作方式是在初始化编程时, 通过向 8255A 的控制端口写入控制字来设定的。写入控制字有工作方式控制字和按位置位/复位控制字。工作方式控制字是用于设定 A 口、B 口和 C 口的工作方式以及数据传送方向; 按位置位/复位控制字是用于设定 C 口的 $\text{PC}_7 \sim \text{PC}_0$ 上的电平。这两个命令都是写入到 8255A 的同一控制端口地址 ($\text{A}_1\text{A}_0 = 11$), 因此由控制字的最高位来区分这两个控制字。若写入的控制字 $\text{D}_7 = 1$, 则该控制字为工作方式控制字; 若写入的控制字 $\text{D}_7 = 0$, 则该控制字为 C 口的按位复位/置位控制字。

1) 工作方式控制字。工作方式控制字是用于设定 A 口、B 口和 C 口的工作方式以及数据传送方向, 其格式如图 5-39 所示。

D_0 : 设置 C 口低 4 位 $\text{PC}_3 \sim \text{PC}_0$ 的数据传送方向。 $\text{D}_0 = 0$, 设置 C 口的低 4 位数据线作为输出线; $\text{D}_0 = 1$, 设置 C 口的低 4 位数据线作为输入线。

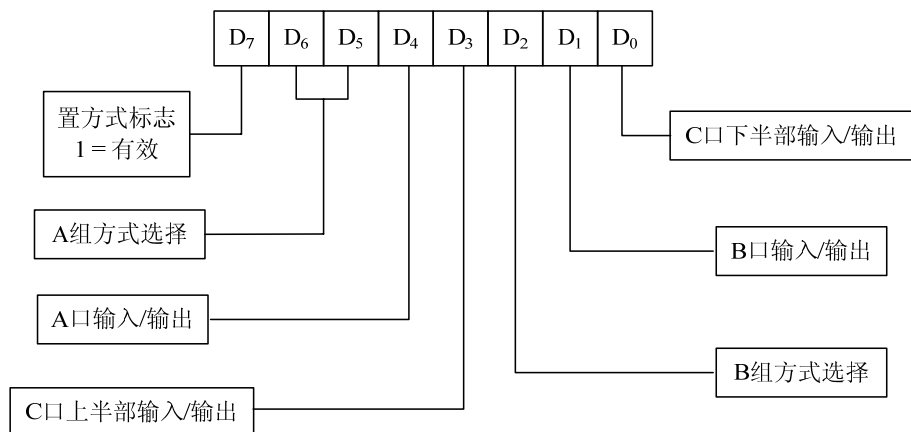


图 5-39 工作方式控制字

D₁: 设置 B 口 PB₇~PB₀ 的数据传送方向。D₁ = 0, 设置 B 口的 8 位数据线作为输出线; D₁ = 1, 设置 B 口的 8 位数据线作为输入线。

D₂: 设置 B 组的工作方式, 控制 B 口和 C 口低 4 位的工作方式。D₂ = 0, 设置 B 组工作于方式 0; D₂ = 1, 设置 B 组工作于方式 1。

D₃: 设置 C 口高 4 位 PC₇~PC₄ 的数据传送方向。D₃ = 0, 设置 C 口的高 4 位数据线作为输出线; D₃ = 1, 设置 C 口的高 4 位数据线作为输入线。

D₄: 设置 A 口 PA₇~PA₀ 的数据传送方向。D₄ = 0, 设置 A 口的 8 位数据线作为输出线; D₄ = 1, 设置 A 口的 8 位数据线作为输入线。

D₆D₅: 设置 A 组的工作方式, 控制 A 口和 C 口高 4 位的工作方式。D₆D₅ = 00, 设置 A 组工作于方式 0, D₆D₅ = 01, 设置 A 组工作于方式 1, D₆D₅ = 10 或 11, 设置 A 组工作于方式 2。

D₇: 写入控制字的特征位, 该位必须为 1, 此时该控制字为工作方式控制字。

当 RESET 引脚信号为高电平、有效时, 所有端口均被设置为输入方式; 当 RESET 引脚信号为低电平时, 如果端口将工作于输入方式, 则不再需要进行设置。A 组和 B 组中的端口可分别工作于输入方式或输出方式, 并不需要同为输入方式或同为输出方式。例如: 把 A 口设置为方式 1 输入, C 口高 4 位设置为输出, B 口设置为方式 0 输出, C 口低 4 位设置为输入。

例 5-5 对 8255A 初始化编程, 已知 8255A 的三个端口地址为 0218H~021AH, 控制寄存器的地址为 021BH, 现在要求 A 口工作于方式 0, 数据线 PA₇~PA₀ 作为输入, B 口工作于方式 1, 数据线 PB₇~PB₀ 作为输出, C 口数据线 PC₇~PC₄ 作为输出线, C 口数据线 PC₃~PC₀ 配合 B 口工作, 设为输入。

由题意可知, 8255A 的工作方式控制字为:

1	0	0	1	0	1	0	0
---	---	---	---	---	---	---	---

初始化程序:

```
MOV  DX, 021BH           ;8255A 控制寄存器端口地址
MOV  AL, 10010100B
OUT  DX, AL              ;8255A 的 A 口工作在方式 0, 输入, C 口高 4 位输出,
                           ;B 口工作在方式 1, 输出, C 口低 4 位输入
```

2) C 口按位置位/复位控制字。置位操作就是将某位置“1”，同理，复位操作就是将某位置“0”，为了让读者有置位/复位的概念，下面举例说明。

例 5-6 在某个外设接口电路中，8255A 的 A 口、B 口和 C 口的端口地址为：40H~42H，控制寄存器的端口地址为 43H，C 口工作为输出，控制 8 个继电器。设定 C 口的输入/输出线为“1”，表示继电器闭合；为“0”，表示继电器断开。现在要求某个继电器闭合，如与 PC₅ 对应的继电器闭合，其他继电器保持原来的状态。

实现程序：

```
IN    AL, 42H           ;读取 C 口的状态信息
OR    AL, 00100000B      ;将 AL 中的第 6 位置“1”
OUT   42H, AL           ;设置 C 口状态
```

上面采用或的方式将 C 口的 8 根输出线进行了设定，在或过程中只有 C 口的第 6 位 PC₅ 发生了变化，被置“1”，其他位并没有变化。下面如果将 PC₅ 控制的继电器断开之前的闭合，只要使用与指令“AND AL, 11011111B”便可实现，即复位 PC₅。这就是典型的置位/复位操作，特点是只改变指定的位，其他位保持原有状态。

C 口按位置位/复位控制字就是用来设置 C 口某一位线的状态为高电平或低电平，仅对 C 口有效，其格式如图 5-40 所示。

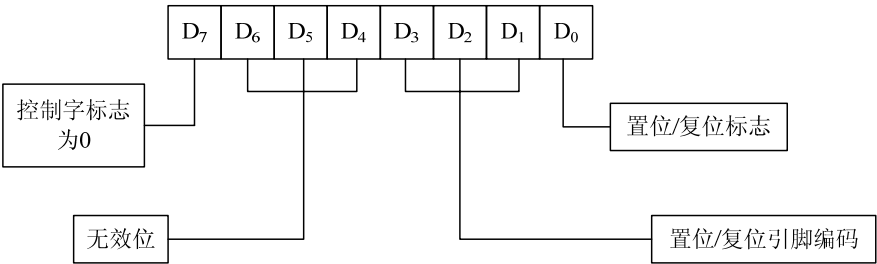


图 5-40 C 口按位置位/复位控制字

D₀：置位/复位标志位。D₀ = 0，将 D₃~D₁ 的按位选择编码所对应的位复位，清零；D₀ = 1，将 D₃~D₁ 的按位选择编码所对应的位置位。

D₃~D₁：C 口的按位选择编码，三位 8 种编码分别对应 C 口的 PC₀~PC₇ 位，具体对应如下所示。

D ₃ ~D ₁	C 口	D ₃ ~D ₁	C 口
000	PC ₀	100	PC ₄
001	PC ₁	101	PC ₅
010	PC ₂	110	PC ₆
011	PC ₃	111	PC ₇

D₆~D₄：无效位，没有定义，通常建议设为 0。

D₇：C 口按位置位/复位控制字的特征位，该位必须为 0，此时该控制字为 C 口按位置位/复位控制字。

采用写入 8255A 的 C 口按位置位/复位控制字的方式实现上例要求，程序段代码：

```
MOV AL, 00001011B      ;C 口按位置位/复位控制字置位 PC5
MOV DX, 43H
OUT DX, AL              ;将控制字写入控制寄存器
```

例 5-7 已知 8255A 的三个端口地址为 0060H~0062H，控制寄存器的地址为 0063H，要求置位 PC₂，复位 PC₆，程序代码如下：

```
MOV DX, 0063H
MOV AL, 05H             ;置位 PC2，D3~D0 = 0101
OUT DX, AL              ;写入控制字
MOV AL, 0CH             ;复位 PC6，D3~D0 = 1100
OUT DX, AL              ;写入控制字
```

3) 读取状态字。在前面讲述 8255A 工作方式时已经指出，当通过程序将 8255A 设定为工作于方式 1 或方式 2 时，C 口会根据不同的工作方式产生或接收不同的选通联络信号。如果对 C 口进行读取操作，读出的内容会包含两部分内容：一部分是作为输入/输出线上的内容；一部分是与联络信号相关的内容。8255A 的状态字为程序查询数据交换方式提供了状态标志位 IBF、 $\overline{\text{OBF}}$ 。

通过读取 C 口的内容，可以查询或检查外部设备的状态，从而影响工作流程。3 个状态字的格式如图 5-41 所示。

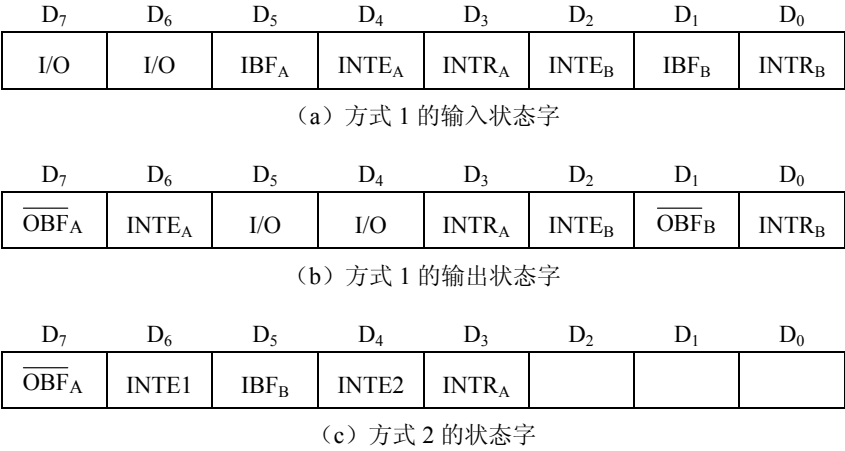


图 5-41 8255A 的 C 口状态字

如图所示，A 组的状态位占 C 口的高 5 位，B 组的状态位占 C 口的低 3 位。需要注意的是，C 口的状态位与 C 口的外部引脚状态不完全一致，在方式 1 的输入时，PC₄ 和 PC₂ 分别表示 INTE_A 和 INTE_B 的状态，而不是 $\overline{\text{STB}}$ 的状态；在方式 1 的输出时，PC₆ 和 PC₂ 分别表示 INTE_A 和 INTE_B 的状态，而不是 $\overline{\text{ACK}}$ 的状态。

在方式 1 的输入，8255A 相当于有两个中断请求输入端 $\overline{\text{STB}}_A$ 和 $\overline{\text{STB}}_B$ ，外设通过向这两个输入端发出低电平有效信号，再通过 INTE_A 或 INTE_B 向 CPU 发出中断请求。在方式 1 的输出，8255A 相当于有两个中断请求输入端 $\overline{\text{ACK}}_A$ 和 $\overline{\text{ACK}}_B$ ，外设通过向这两个输入端发出低电平有效信号，再通过 INTE_A 或 INTE_B 向 CPU 发出中断请求。

方式 2 状态字的含义是 A 口方式 1 的输入和输出状态组合。该状态字中有 INTE1 是输出中断允许位, INTE2 是输入中断允许位。

2. 8255A 应用举例

(1) 并行打印机

例 5-8 利用 8255A 的 A 口与微型打印机相连, 将内存缓冲区 Buffer 中的字符打印输出, 试完成相应的软硬件设计 (CPU 为 8088)。

打印机内有一个以 8 位专用微处理器为核心的打印机控制器, 负责打印功能的处理, 以及打印机本身的管理, 并通过机内一个并行接口 (Centronics 标准接口) 与主机进行通信, 接收主机送来的打印数据和控制命令。该标准接口引脚的功能如下:

$\overline{\text{STROBE}}$: 选通脉冲, 输入, 主机发出, 有效时用于将 $\text{DATA}_8 \sim \text{DATA}_1$ 上的数据置入打印机的缓冲器, 接收数据。

$\text{DATA}_8 \sim \text{DATA}_1$: 数据线 1~8, 输入, 主机送给打印机的 8 位数据。

$\overline{\text{ACK}}$: 响应信号, 输入, 作为打印机已接收到一个数据的应答信号, 有效时准备接收数据。

BUSY : 忙信号, 输出, 有效时表示打印机当前忙, 不能接收数据。在数据输入期间、打印操作期间、脱机状态、出错状态情况下有效。

PE : 打印机缺纸, 输出。

SLCT : 选择状态, 输出, 高电平表示脱机状态, 低电平表示联机状态。

$\overline{\text{AUTOLF}}$: 自动换行走纸, 输入, 有效时每当打印完后 (即遇到回车符), 打印机自动前进一步。

$\overline{\text{INIT}}$: 打印机初始化, 输入, 有效时复位打印控制器, 并清除打印缓冲器。

$\overline{\text{ERROR}}$: 出错信号, 输出, 有效时表示打印机出错、脱机或缺纸。

$\overline{\text{SLCTIN}}$: 选择输入, 输入, 有效时用于打印机接收 $\text{DATA}_8 \sim \text{DATA}_1$ 上的数据, 且只有该信号有效时才能接收。

打印机的工作过程如下:

主机查询 BUSY 信号, 如果 $\text{BUSY} = 1$, 表示打印机忙, 主机等待, 如果 $\text{BUSY} = 0$, 打印机不忙, 主机可以送出数据, 此时主机将要打印的数据送到数据线上; 然后发出选通信号 $\overline{\text{STROBE}}$, 打印机收到该信号后, 将数据读入内部缓冲器; 同时发出忙信号, 将 BUSY 置位, 通知主机停止传送数据, 其正在处理打印数据; 打印机打印完输入的数据或执行完一个功能操作后, 发出 $\overline{\text{ACK}}$ 应答信号给主机, 同时撤销忙信号, 即将 BUSY 复位, 通知主机可以发送下一个打印数据。

由以上工作原理, 设计该并行打印机电路, 总体设计如下:

8255A 的 A 口工作于方式 0, 输出, C 口高 4 位工作于方式 0 输入, 低 4 位工作于方式 0 输出, 由 C 口的 PC_0 产生打印机的选通信号, 由 C 口的 PC_7 接收打印机发出的忙信号 (BUSY), 查询是否可以进行数据的传送, 设 8255A 的 A 口、B 口和 C 口的端口地址为 $80\text{H} \sim 82\text{H}$, 控制寄存器的端口地址为 83H 。

主机、打印机和 8255A 的接口硬件电路设计如图 5-42 所示。

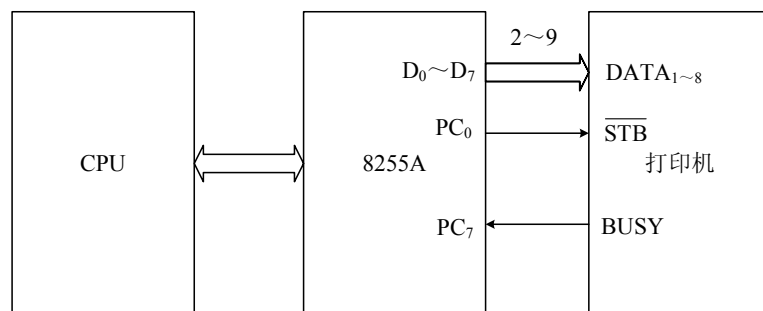


图 5-42 打印机接口电路原理图

详细程序源代码:

```

        MOV  AL, 01H
        OUT  83H,AL           ;写入 C 口按位置位/复位控制字, 置位 PC0
DATA SEGMENT
Buffer  DB  'HELLO WORLD!', '#'
DATA  ENDS
CODE SEGMENT
        ASSUME  CS:CODE, DS:DATA
START: MOV  AX, DATA
        MOV  DS, AX
        MOV  SI, OFFSET Buffer
        MOV  AL, 88H           ;8255A 初始化 A 口工作于方式 0, 输出
        OUT  83H, AL           ;C 口高 4 位工作于方式 0 输入
                                   ;低 4 位工作于方式 0 输出
LOOP:   IN   AL, 82H
        TEST AL, 80H           ;查询 BUSY 信号, 查询 PC7 是否为 1
        JNZ  LOOP             ;如果 PC7 为 1, 表示忙, 则等待
        MOV  AL, [SI]
        CMP  AL, '#'           ;判断是否是 Buffer 中的结束符
        JZ   DONE             ;如果是, 则跳转到 DONE 标号处执行输出回车
        OUT  80H, AL           ;如果判断不是结束符, 则从 8255A 的 A 口输出数据
        MOV  AL, 00H
        OUT  83H, AL           ;复位 PC0, 停止接收数据
        MOV  AL, 01H
        OUT  83H, AL           ;置位 PC0, 产生选通信号
        INC  SI                ;修改指针位置, 指向下一个字符, 继续读取数据
        JMP  LOOP
DONE:   MOV  AL, 0DH
        OUT  80H, AL           ;处理完打印机缓冲器中的数据, 输出回车符

```

```

MOV AL, 00H
OUT 83H, AL
MOV AL, 01H
OUT 83H, AL
LOOP1: IN AL, 82H
TEST AL, 80H           ;查询 BUSY 信号, 查询 PC7 是否为 1
JNZ LOOP1              ;如果 PC7 为 1, 表示忙, 则等待
MOV AL, 0AH
OUT 80H, AL            ;输出换行符
MOV AL, 00H
OUT 83H, AL
MOV AL, 01H
OUT 83H, AL            ;产生选通
MOV AH, 4CH             ;功能号 4CH, 程序终止
INT 21H                ;返回 DOS
CODE ENDS
END START

```

本例是采用查询方式传送数据的过程, 也可采用中断方式传送数据, 利用 $\overline{\text{ACK}}$ 信号来产生中断请求, 在中断服务程序中送出下一个打印数据。如此重复工作, 就可以将全部字符打印出来。

(2) 双机并行通信

例 5-9 8255A 应用于双机通信, 要求在甲乙两台 PC 机之间并行传送 1KB 数据。要求甲机作为发送主机, 乙机作为接收主机。甲机的 8255A 工作于方式 1 发送数据, 乙机的 8255A 工作于方式 0 接收数据, 甲乙两机均采用查询式输入/输出方式工作。

根据题意要求, 进行硬件电路设计, 甲机的 8255A 为方式 1 发送, 故把 A 口指定为输出, 发送数据, PC₇ 和 PC₆ 引脚分别作联络信号线 $\overline{\text{OBF}}$ 和 $\overline{\text{ACK}}$; 乙机的 8255A 为方式 0 接收, 故把 A 口指定为输入, 接收数据, 选择 PC₇ 和 PC₃ 作为联络信号线, PC₇ 输入、PC₃ 输出。接口电路连接如图 5-43 所示。

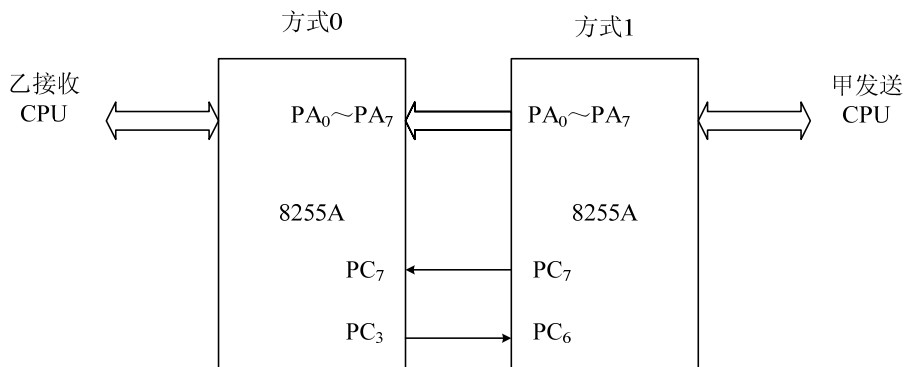


图 5-43 利用 8255A 进行查询方式的双机并行通信

双机并行通信源程序:

;甲机发送的程序段

```
BUFFER DB 1024 DUP(?)... ;定义 1024 个待发送数据
MOV DX, 303H ;8255A 命令端口
MOV AL, 1010000B ;初始化工作方式控制字
OUT DX, AL ;输出方式字
MOV AL, 0DH ;置发送允许  $INTE_A = 1$ ,  $PC_6$  置位
OUT DX, AL
MOV SI, OFFSET BUFFER ;设置发送数据区的指针
OUT CX, 3FFH ;发送字节数
MOV DX, 300H ;向 A 口写数据, 同时产生  $\overline{OBF}$  信号
MOV AL, [SI] ;送给乙机, 获取乙机的  $\overline{OBF}$  信号
OUT DX, AL
INC SI ;内存地址加 1
DEC CX ;传送字节数减 1
LOOP: MOV DX, 302H ;8255A 的 C 口地址
IN AL, DX ;查发送中断请求  $INTR_A=1$ 
AND AL, 08H ;是否  $PC_3=1$ 
JZ LOOP ;若无中断请求则等待; 若有向 A 口写数据
MOV DX, 300H ;8255A 的 A 口地址
MOV AL, [SI] ;从内存取数
OUT DX, AL ;通过端口 A 向乙机发送第 2 个数据
INC SI ;内存地址加 1
DEC CX ;字节数减 1
JNZ LOOP ;数据未完, 继续
MOV AH, 4CH ;已完, 功能号 4CH, 程序终止
INT 21H ;返回 DOS
;乙机接收的程序段
BUFFER DB 1024 DUP(?) ;接收数据缓冲区
MOV DX, 303H ;8255A 命令端口
MOV AL, 10011000B ;初始化工作方式控制字
OUT DX, AL
MOV AL, 00000111B ;置位  $PC_3$ 
OUT DX, AL
MOV DI, OFFSET BUFFER ;设置接收数据区的指针
MOV CX, 3FFH ;接收字节数
LOOP1: MOV DX, 302H
IN AL, DX ;查询甲机的  $\overline{ACK}$  是否为 0,  $PC_7$  是否为 0
```

AND AL, 80H	;查询甲机是否有数据送来
JNZ LOOP1	;若无数据送来, 则等待; 若有数据, 则从 A 口读数
MOV DX, 300H	;8255A 的 A 口地址
IN AL, DX	;从端口 A 读入数据
MOV [DI], AL	;存入内存
MOV DX, 303H	;产生 $\overline{\text{ACK}}$ 信号, 并发回给甲机
MOV DX, 00000110B	;PC ₃ 复位
OUT DX, AL	
INC DI	;内存地址加 1
DEC CX	;字节数减 1
JNZ LOOP1	;数据未完, 则继续
MOV AX, 4C00H	;已完, 功能号 4CH, 程序终止
INT 21H	;返回 DOS

(3) 数码管显示

例 5-10 利用并行接口 8255A 控制 8 个数码管显示。8255A 的 A 口接 4 个开关: K₃~K₀, A 口工作于方式 0, 作输入; B 口连接一个共阴极 8 段 LED 数码管, B 口工作于方式 0, 作输出。将 8255A 的 A 口连接的四个开关不同输入的 16 (2⁴) 种状态 0H~0FH 经 B 口输出显示。画出接口硬件连接电路图, 并编制程序实现上述功能。

实现设计: 根据要求, 本例 8255A 工作于方式 0, 8255A 的端口地址可以由主机 A₁₅~A₀ 提供, 8255A 的 D₇~D₀、 $\overline{\text{WR}}$ 、 $\overline{\text{RD}}$ 分别与主机 CPU 的 D₇~D₀、 $\overline{\text{WR}}$ 、 $\overline{\text{RD}}$ 连接, 8255A 的片选信号 $\overline{\text{CS}}$ 与地址译码器输出端相连, 8255A 的端口选择信号线 A₀ 和 A₁ 与 CPU 的地址线 A₀ 和 A₁ 连接, 并设 8255A 的 A 口、B 口和 C 口的端口地址为 0FFD0H~0FFD2H, 控制寄存器的端口地址为 0FFD3H。A 口的 PA₃~PA₀ 连接 4 个开关 K₃~K₀, 其输入有 16 种组合状态, 即 0000~1111 (0H~0FH), 经 B 口后与 LED 数码管连接, 可输出一位十六进制数 0~F, 在 LED 数码显示管上显示。

8 段 LED 数码显示管有共阳极和共阴极两种结构, 共阳极 LED 的二极管阳极均接+5V, 输入端为低电平时, 二极管导通发亮; 共阴极 LED 的二极管阴极均接地, 输入端为高电平时, 二极管导通发亮。本例要求采用一个共阴极 8 段 LED 数码管, 显示代码如表 5-8 所示。

表 5-8 LED 数码管显示代码

显示字符	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
共阴极七段 显示代码 (H)	3F	06	5B	4F	66	6D	7D	07	7F	6F	77	7C	39	5E	79	71

如表所示, 当 8255A 的 A 口所接开关 K₃~K₀ 的状态为 1010B, B 口对应输出七段显示代码为 77H, LED 数码管显示为十六进制数 A。

基于以上分析, 硬件连接电路如图 5-44 所示。

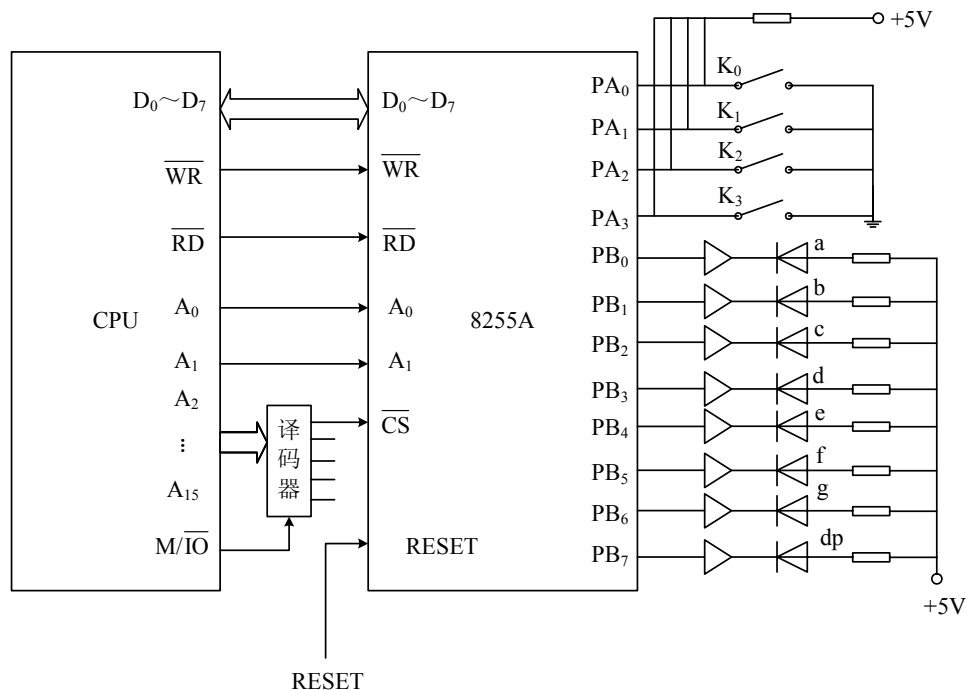


图 5-44 8255A 显示器接口图

编制实现程序如下：

```

DATA SEGMENT
LIST DB 3FH, 06H, 5BH, 4FH, ..., 71H ; 共阴极七段显示代码
PORTA EQU FFD0H
PORTB EQU FFD1H
CTRL EQU FFD3H
DATA ENDS
CODE SEGMENT
ASSUME CS: CODE, DS: DATA
START: MOV AX, DATA
MOV DS, AX
MOV AL, 90H ;工作方式控制字：A 口方式 0 输入，B 口方式 0 输出
MOV DX, CTRL ;工作方式控制口地址
OUT DX, AL ;将控制字写入控制端口
LOOP0: MOV DX, PORTA ;A 口地址
IN AL, DX ;读取 A 口开关状态
AND AL, 0FH ;屏蔽 A 口高 4 位
MOV BX, OFFSET LIST ;共阴极七段显示代码表的首地址送给 BX
AND AX, 00FFH ;屏蔽 AX 的高位字节，保留 A 口的开关状态
ADD BX, AX ;形成显示字符的代码地址
MOV AL, [BX] ;取出显示代码送给 AL

```

```

MOV DX, PORTBH      ;B 端口地址
OUT DX, AL           ;显示代码送 B 口显示
CALL DELAY           ;调用显示延时子程序
JMP LOOP0            ;循环显示
DELAY PROC           ;显示延时子程序
PUSH CX
PUSH AX
MOV CX, 0010H
T1: MOV AX, 0010H
T2: DEC AX
    JNZ T2
    LOOP T1
POP AX
POP CX
RET                  ;子程序返回
DELAY ENDP
CODE ENDS
END START

```

(4) 行列式键盘

例 5-11 扫描键盘按键，并保存相应的值，硬件接口连接电路如图 5-45 所示。设 8255A 的 A 口、B 口和 C 口的端口地址为 400H~402H，控制寄存器的端口地址为 403H，接收 4×4 的行列式键盘的按键后结束。

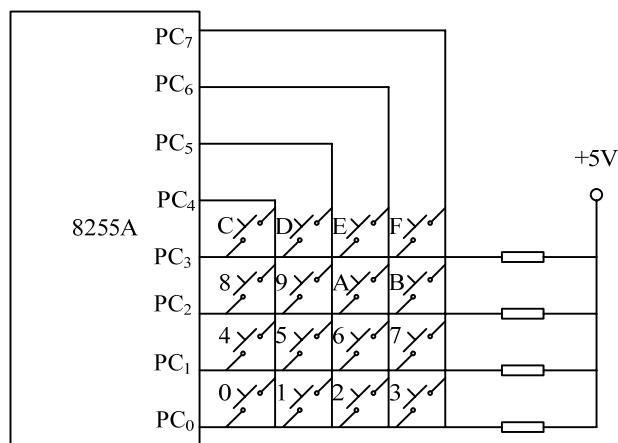


图 5-45 行列式键盘硬件连接电路。

行列式键盘的工作过程如下：PC₄~PC₇ 送全“0”，再读 PC₀~PC₃，若全为“1”，则表示没有按键被按下。否则，有按键按下，进行按键扫描。按键扫描方法：使 PC₄=0 低电平，PC₅~PC₇ 为高电平，读取 PC₀~PC₃，如果全是“1”，则表示该列没有按键被按下；否则，有按键在该列上被按下，进行进一步的按键确定判断，判断读取的 PC₀~PC₃ 数据中哪一位

为“0”，从而确定被按下的按键。若该列没有按键被按下，依次按以上过程进行 PC₅、PC₆、PC₇ 的按键扫描。

在键盘的设计中，还需要处理在按键过程中的抖动和重键问题。在机械按键过程中，往往会出现按键在按下和弹起之间抖动几下才稳定到被按下状态；在释放时，也会出现类似的情况。抖动的持续时间一般为 10ms 左右，利用硬件或软件延时的方法都可以消除抖动。重键就是两个或多个键同时被按下，通常只承认先被识别的按键，其他被按下的键不做处理，直到所有按键都弹起后再进行新的按键扫描。

程序执行流程如图 5-46 所示。

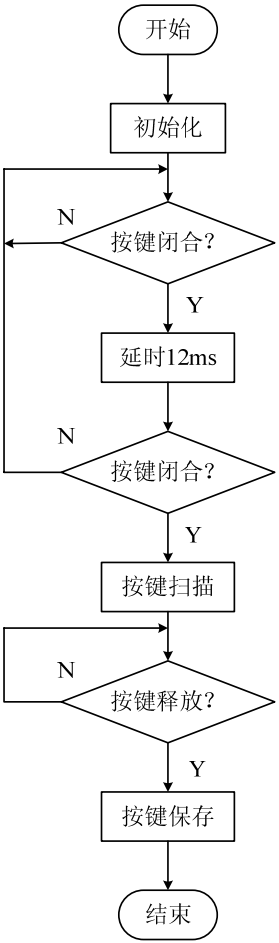


图 5-46 键盘扫描程序流程图

程序源代码：

```
DATA SEGMENT
BUFFER DB 16 (?)
DATA ENDS
CODE SEGMENT
```

```

        ASSUME CS:CODE,DS:DATA
START: MOV  AX, DATA
        MOV  DS, AX
        LEA  SI, BUFFER
        MOV  CL, 16                ;初始化按键次数
        MOV  AL, 81H               ;8255A 控制字
        MOV  DX, 403H
        OUT  DX, AL                ;8255A 初始化
LOOP:   CALL KS                    ;读取按键
        CMP  AL, 0FH               ;判断有无按键按下
        JZ   LOOP                  ;没有按键按下，循环等待
        CALL DELAY                 ;延时 12ms，消除抖动
        CALL KS
        CMP  AL, 0FH               ;再次判断有无按键按下
        JZ   LOOP
        MOV  BL, 0EFH              ;初始化列码
        MOV  BH, 0                 ;初始化列计数器
AGAIN:  MOV  DX, 402H
        MOV  AL, BL
        OUT  DX, AL                ;输出列码
        IN   AL, DX                ;读取行码
        AND  AL, 0FH
        CMP  AL, 0FH
        JZ   NEXT                  ;该列没有按键按下，继续扫描下一列
        CMP  AL, 0EH               ;判断该列是否第一个按键按下
        JNZ  TWO
        MOV  AL, 0
        JMP  FREE
TWO:    CMP  AL, 0DH               ;判断该列是否第二个按键按下
        JNZ  THREE
        MOV  AL, 4
        JMP  FREE
THREE:  CMP  AL, 0BH               ;判断该列是否第三个按键按下
        JNZ  FOUR
        MOV  AL, 8
        JMP  FREE
FOUR:   CMP  AL, 07H               ;判断该列是否第 4 个按键按下
        JNZ  NEXT

```

```

        MOV AL, 0CH
FREE:   PUSH AX
WAIT1:  CALL KS
        CMP AL, 0FH
        JNZ WAIT1           ;按键未释放，等待
        POP AX
        ADD AL, BH          ;按键键值 = 扫描键值 + 列计数数值
        MOV [SI], AL        ;保存相应按键键值
        INC SI
        DEC CL
        JZ EXIT
        JMP LOOP
NEXT:   INC BH              ;列计数值加 1
        ROL BL, 1           ;列码循环左移一位
        CMP BL, 0FEH        ;判断该轮键值扫描是否结束
        JNZ AGAIN
        JMP LOOP
EXIT:   MOV AH, 4CH
        INT 21H             ;返回 DOS
KS      PROC NEAR
        MOV DX, 402H
        MOV AL, 0FH
        OUT DX, AL          ;使所有列线为低电平
        IN AL, DX           ;读取行值
        AND AL, 0FH         ;屏蔽高 4 位
        RET
KS      ENDP
DELAY  PROC NEAR
        PUSH BX
        PUSH CX
        MOV BX, 2000
DEL1:   MOV CX, 0
DEL2:   LOOP DEL2
        DEC BX
        JNZ DEL1
        POP CX
        POP BX
        RET

```

```
DELAY ENDP
```

```
CODE ENDS
```

```
END START
```

(5) 键盘与 LED 数码管综合应用

例 5-12 如图 5-47 所示，由 8255A 的 A 口和 C 口组成的键盘接口电路，扫描键盘的工作原理如前所述。要求将当前按下的按键号在 4 位 LED 数码管上同时显示出。在此基础上让 4 位 LED 数码管按“先按先出”方法，采用左移或右移方法在 4 位 LED 上显示出最后 4 个按键号。

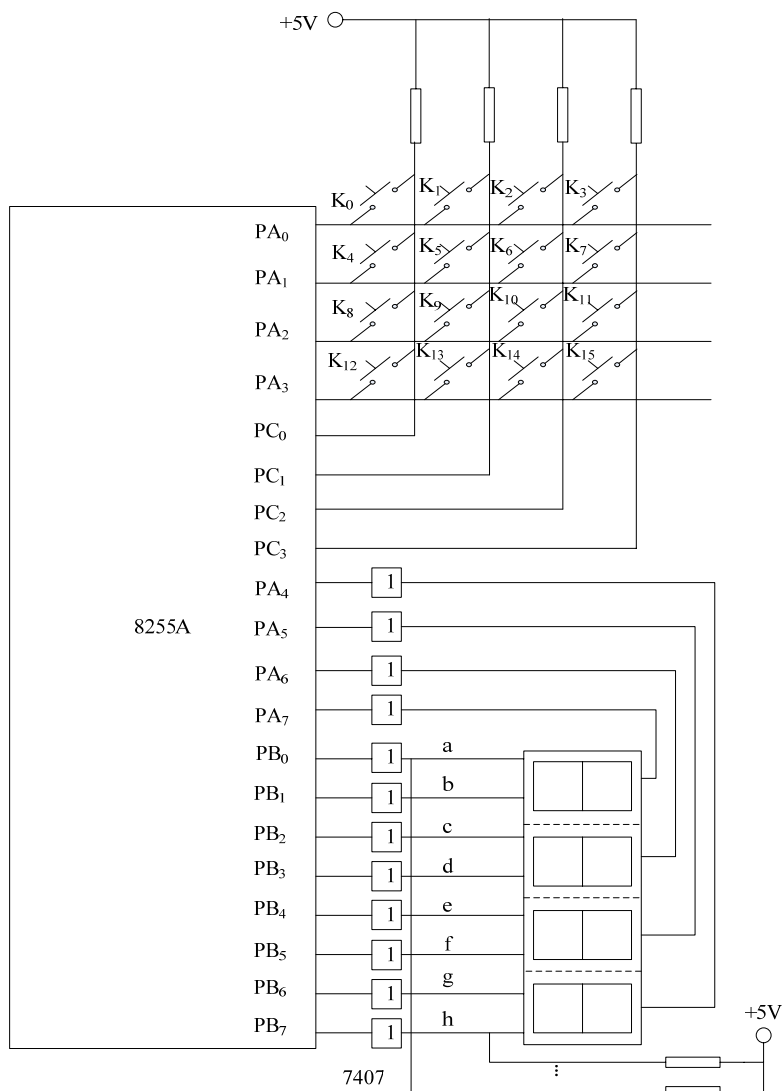


图 8-47 8255A、键盘和 LED 接口

程序源代码:

```
DATA SEGMENT
```

```

LED_CODE DB 3FH, 06H, 5BH, 4FH, 66H, 6DH, 7DH, 0EH
          DB 7FH, 6FH, 77H, 7CH, 39H, 5EH, 79H, 71H
DATA     ENDS
CODE     SEGMENT
          ASSUME CS:CODE,DS:DATA
START: MOV  AX, DATA
        MOV  DS, AX
        LEA  SI, LED_CODE
        MOV  BH, 0
        MOV  AL, 89H                ;8255A 的 A 口工作在方式 0 输出, B 口工作在
                                     ;方式 0 输出, PC7~PC4 输入, PC3~PC0 输入

        MOV  DX, 403H
        OUT  DX, AL                ;8255A 初始化
WAIT1:  MOV  AL, 0F0H
        MOV  DX, 400
        OUT  DX, AL                ;PA3~PA0 输出为 0, PA7~PA4 输出为 1
        MOV  DX, 402H
        IN   AL, DX                ;通过读 PC3~PC0 判断是否有按键按下
        AND  AL, 0FH
        CMP  AL, 0FH
        JZ   WAIT1                ;没有按键按下, 重新扫描
        CALL DELAY                ;有按键按下, 延时 12ms 消除抖动
        MOV  DX, 402H
        IN   AL, DX
        AND  AL, 0FH
        CMP  AL, 0FH
        JZ   WAIT1                ;没有按键按下, 重新扫描
        MOV  BL, 0                ;将 K0 键号送 BL
        MOV  CL, 0FEH            ;设置扫描值, 首先使 PA0 输出为 0
        MOV  CH, 1                ;赋行数初值
WAIT2:  MOV  AL, CL
        MOV  DX, 400H
        OUT  DX, AL                ;使对应的 PAi 行输出为 0
        MOV  DX, 402H
        IN   AL, DX                ;读列线
        AND  AL, 0FH
        CMP  AL, 0FH
        JNZ  WAIT3                ;当前行有键按下, 转到 WAIT3 查询列号

```

```

        ROL  CL, 1           ;左移, 分别使下一行 PA1、PA2、PA3 输出 0
        MOV  AL, CL
        ADD  BL, 04          ;键号加 4 分别指向 K4、K8 和 K12 行键
        INC  CH              ;当前行加 1
        CMP  CH, 5
        JNZ  WAIT2          ;4 行未扫描完, 转 WAIT2 继续扫描
        JMP  WAIT1          ;4 行扫描结束, 重新扫描
WAIT3:  RCR  AL, 1           ;右移, 判断按键是否在此列
        JNC  WAIT4          ;当前列有按键按下, 转 WAIT4
        JNC  BL              ;指向 K1、K2、K3 列
        JNC  SI
        JMP  WAIT3
WAIT4:  MOV  AL, [SI]        ;从 LED_CODE 中获取 7 段码, 按键号最小值为 0
        MOV  DX, 401H
        OUT  DX, AL          ;显示按键号
        MOV  AL, 0FH         ;PA7~PA4 均为 0, PA3~PA0 均为 1
        MOV  DX, 400H
        OUT  DX, AL          ;4 位 LED 显示当前按键号
        CALL DELAY
        JMP  WAIT1
DELAY  PROC NEAR
        PUSH BX
        PUSH CX
        MOV  BX, 2000
DEL1:  MOV  CX, 0
DEL2:  LOOP DEL2
        DEC  BX
        JNZ  DEL1
        POP  CX
        POP  BX
        RET
DELAY  ENDP
CODE  ENDS
END    START

```

(6) 8255A、8259A、8254 综合应用

例 5-13 图 5-48 为定时中断电路, 图中 8254 计数器 1 和计数器 2 的时钟分别由对应的外设提供, 当计数器 1 和计数器 2 在分别计满外设输出的脉冲信号后, 便通过各自的输出向 8259A 申请中断, 若 8259A 在同一时刻接收到计数器 1 和计数器 2 的中断请求时, 首先响应优先级

高的 IR_0 中断请求， IR_0 的中断服务程序通过 8255A 的 A 口向外设输出数据， IR_1 的中断服务程序通过 8255A 的 B 口接收外设输出的数据。

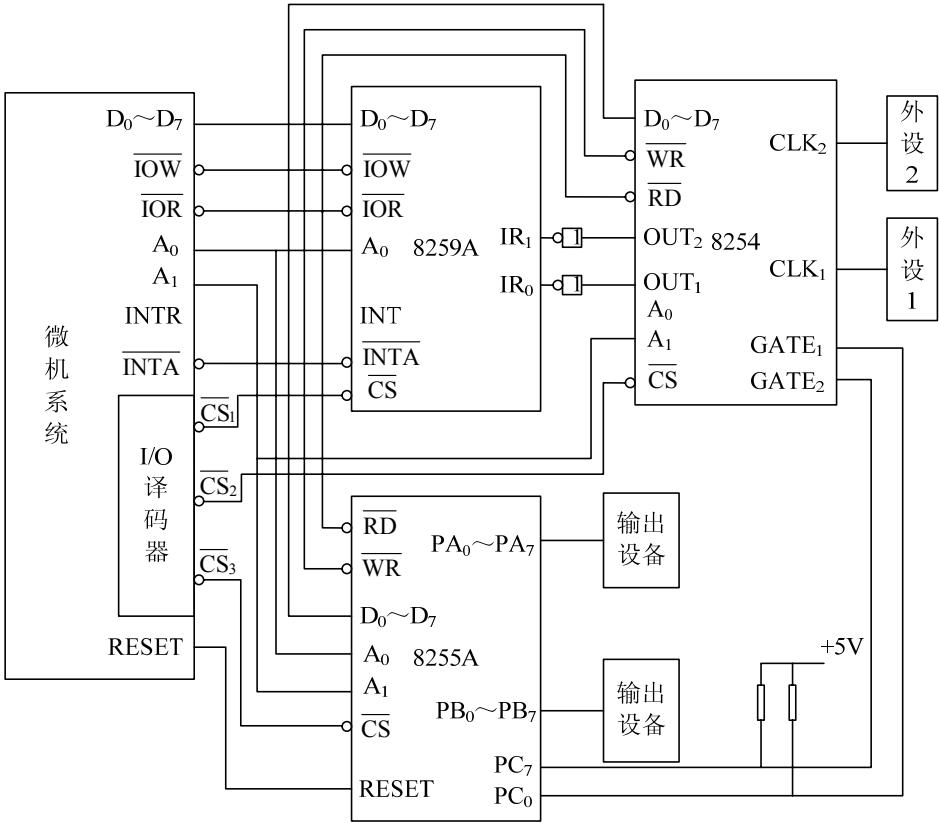


图 5-48 外设数据的输入输出

部分源程序：

```

...
MOV  DX, 8255_控制端口
MOV  AL, 0EH
OUT  DX, AL                ;PC7 = 1, GATE2 = 0
MOV  AL, 00H
OUT  DX, AL                ;PC0 = 1, GATE1 = 0
MOV  DX, 8254_控制端口
MOV  AL, 55H                ;8254 计数器 1 工作在方式 2
OUT  DX, AL
MOV  AL, 0B5H               ;8254 计数器 2 工作在方式 2
OUT  DX, AL
MOV  DX, 8254_计数器 1
MOV  AL, XXH                ;8254 计数器 1 的计数初值
OUT  DX, AL

```

```

MOV  DX, 8254_计数器 2
MOV  AL, XXH                      ;8254 计数器 2 的计数初值低 8 位
OUT  DX, AL
MOV  AL, XXH                      ;8254 计数器 2 的计数初值高 8 位
OUT  DX, AL
MOV  AL, 17H                      ;ICW1: 边沿触发, 单组件
OUT  20H, AL                      ;8259A 内部寄存器选择信号 A0 为 0
MOV  AL, 40H                      ;ICW2: 中断类型为 64
OUT  21H, AL                      ;8259A 内部寄存器选择信号 A0 为 1
MOV  AL, 03H                      ;ICW1: 自动 EOI
OUT  21H, AL
MOV  AL, 0FCH                     ;OCW1: 屏蔽 IR7~IR2
OUT  21H, AL
MOV  AL, 0FCH
OUT  21H, AL
MOV  DX, 8255A_控制端口
MOV  AL, 86H                      ;8255A 的 A 口工作在方式 0 输出
                                   ;B 口工作在方式 1 输入

OUT  DX, AL
MOV  AL, 0FH
OUT  DX, AL                      ;置位 PC7、GATE2, 8254 计数器 1 开始计数
MOV  AL, 01H
OUT  DX, AL                      ;置位 PC0、GATE1, 8254 计数器 2 开始计数
STI
...
中断向量号      中断向量指针      中断向量
INT64            100H            0200H      ;IR0 中断服务程序有效地址
                102H            1000H      ;IR0 中断服务程序有效地址
INT65            104H            0400H      ;IR1 中断服务程序有效地址
                106H            1000H      ;IR1 中断服务程序有效地址

ORG  1000H:200H                  ;IR0 中断服务程序
PUSH  AX                        ;数据输出
...
MOV  DX, 8255A_A 口
OUT  DX, AL
...
POP  AX
STI
IRET

```

```

        ORG 1000H:400                ;IR1 中断服务程序
        PUSH AX
        ...
        IN AL, 8255A_C 口
        AND AL, 00000010B
        JZ RET1
        IN AL, 8255A_B 口            ;IBFB 为 1，读 B 口输入缓冲器中数据
        ...
RET1:    POP AX
        STI
        IRET

```

例 5-14 用 8086, 8255, 8259 和 8254 构造系统实现对指示灯控制。8255 的 PA0、PA1、PA2 的三位 DIP 开关，通过 DIP 开关的闭合状态决定接在 PB 口上的八个指示灯之一闪烁。如 PA2、PA1、PA0 为 000 时，PB0 上所接的指示灯闪烁，其余灯熄灭。要求闪烁频率为每秒 10 次。设 8259 地址为 20H 和 21H，8255 地址为 60H~63H，8254 地址为 40H~43H，时钟频率为 50KHz，8259 中断向量号为 70H 和 71H。试设计硬件连接电路，填写中断向量表，编写全部初始化程序，完成控制程序编写。

程序源代码：

;全局定义为了便于在实验机上调试，现将地址改成可以在实验机上调试的地址

;中断矢量号从 20H 开始，8254 工作于方式 3

```

D8255A    EQU 210H
D8255B    EQU 211H
D8255C    EQU 212H
Z8255     EQU 213H
D8254_0   EQU 218H
D8254_1   EQU 219H
D8254_2   EQU 21AH
Z8254     EQU 21BH
INT0      EQU 200H
INT1      EQU 201H

STACK SEGMENT STACK
        STA DW 50 DUP(?)
        TOP EQU LENGTHSTA
STACK ENDS
CODE SEGMENT
        ASSUME CS:CODE,DS:CODE,SS:STACK
        ORG 1000H
START:                                     ;初始化

```

;8255

;PA 口方式 0 输入, PB 口方式 0 输出, C 口上半口输入, 下半口输出

MOV AL, 10010000B

MOV DX, Z8255

OUT DX, AL

;MOV AL, 0FFH

MOV AL, 00H

MOV DX, D8255B

OUT DX, AL

;8254

;用计数器 0 计数, 工作方式 3 产生中断, 计数初值为 2500, 十六进制计数

MOV AL, 00111110B

MOV DX, Z8254

OUT DX, AL

MOV AL, 0C4H

MOV DX, D8254_0

OUT DX, AL

MOV AL, 9H

OUT DX, AL

;8259

;设置中断的入口地址

XOR BX, BX

PUSH BX

POP DS

CLI ;关中断

;以下 3 个中断向量分别分配给 IRQ0

LEA AX, IRQ0_INTERRUPT ;中断向量 20H

MOV DS:WORD PTR[80H], AX

MOV DS:WORD PTR[82H], CS

NOP

PUSH CS

POP DS

;ICW1, 采用单片 8259, 边缘触发, 需要 ICW4

MOV AL, 00010011B

MOV DX, INT0

OUT DX, AL

;ICW2, 中断起始矢量号为 70H 号

MOV AL, 70H

```

        MOV  DX,INT1
        OUT  DX,AL
;ICW4, 指定非特殊中断结束
        MOV  AL,01H
        OUT  DX,AL
;8259 操作命令
;OCW1: 特征 A0 = 1 按顺序判断
;初始化结束后, 只有 OCW1 要求 A0=1
;其他操作命令 OCW2, OCW3 都要求 A0=0
        MOV  AL,0FEH  ;开放 0 中断
        OUT  DX,AL
;OCW2: 特征 A0 = 0, D4=0, D3=0
        MOV  AL, 20H  ;非特殊 EOI 结束中断
        MOV  DX,INT0
        OUT  DX,AL
        STI
;主程序
        MOV  PBOFF,0
MAIN:
        ;不做任何事情
        JMP  MAIN
        ;数据区
        ;PB0, PB1, PB2, PB3, PB4, PB5, PB6, PB7 显示码
PBFLASH DB 01H,02H,04H,8H,10H,20H,040H,080H
PBOFF DB 0
PA8255 DB 0

;外部硬件中断处理程序
;*****
IRQ0_INTERRUPT PROC NEAR
        PUSH  AX
        PUSH  BX
        PUSH  DX
;为保证实时钟计数不开中断
        STI  ;开中断
        MOV  AL,PBOFF
        CMP  AL,01H
        JE   OUTPB0
        MOV  DX,D8255A

```

```

        IN  AL, DX
        AND AL, 07
        LEA  BX, PBFLASH
        XLAT
        MOV  DX, D8255B
        OUT  DX, AL
        MOV  PBOFF, 1
        JMP EXIT0
OUTPB0:
        MOV DX, D8255B
        MOV AL, 0H
        OUT DX, AL
        MOV  PBOFF, 0
EXIT0:   CLI      ;关中断
        ;EOI 方式
        MOV  AL, 20H ;不指定 EOI 结束中断
        MOV  DX, INT0
        OUT  DX, AL
        POP DX
        POP BX
        POP AX
        IRET
IRQ0_INTERRUPT ENDP
        CODE ENDS
        END START

```

5.4 串行接口技术

由于计算机网络及多微机系统的发展, 通信技术应用日益广泛。通信包括计算机与计算机之间、计算机与外设之间的信息传送。

计算机与外部信息交换即通信有两种基本方式: 并行通信方式和串行通信方式。

在并行通信方式中, 是将一组 8 位、16 位, 甚至 32 位的数据同时从一个设备传送到另一个设备。这种情况下, 数据的位数决定了传输线的根数。在多微机系统中, 当各台微机相距比较远时, 由于并行通信方式存在连接数量大、故障几率大、工程造价高等原因, 一般不能使用并行通信方式进行数据的传送。串行通信方式是在传输过程中, 数据一位一位地沿着一条传输线从一个设备传到另一个设备, 每一位都占据规定的时间间隔。串行通信由于是按位传送, 传输的速率慢, 数据的各位分时使用一个传输通道, 连接线数量少。在远距离通信中, 串行通信工程造价低, 而且可以利用显存的通信信道, 只要增加调制解调器即可。以传送 8 位数据 01101010 为例, 并行传送与串行传送的过程如图 5-49 所示。

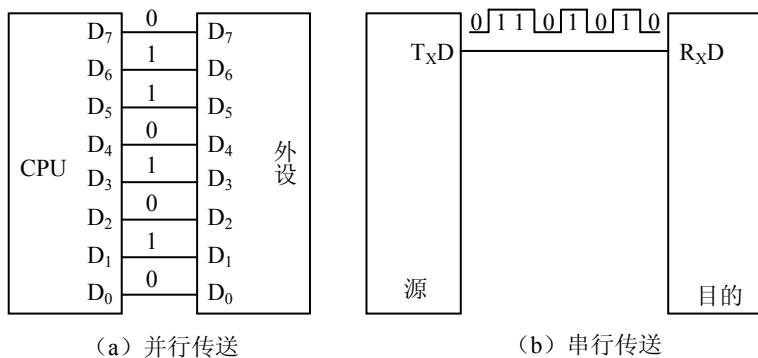


图 5-49 并行传送与串行传送

对于在计算机系统内部或近距离的设备之间传送数据，可采用并行通信方式，并行通信数据线的传送速度快、工作效率高。而远距离的设备之间的数据传送，则采用串行通信方式。目前远距离通信利用电话线，比并行电缆和放大器更经济、方便。

微型计算机系统向外传送数据或者接收外部传送来的数据是以并行方式进行的，这是计算机内部数据所固有的特征。若要在计算机和外部设备之间进行串行通信，就需要一种接口电路，它具有把 CPU 传送来的并行数据转换为串行数据发送出去的功能，还具有把外部设备传送来的串行数据转换成并行数据传送给 CPU 的功能。这种接口电路称为串行通信接口电路。

5.4.1 串行通信接口概述

串行通信按通信约定的格式分为两种：异步串行通信方式和同步串行通信方式。

1. 异步串行通信 ASYNC (Asynchronous Data Communication)

异步串行通信以字符为通信单位进行传输，在信息传输过程中，不必与数据一起发送同步脉冲，而用起始位表示字符的开始，用停止位表示字符的结束，发送及接收的数据不需要保持严格的同步，所采用的信息格式如图 5-50 所示。

在异步数据传送中，在 CPU 与外设之间必须遵循三项规定：

1) 字符格式：异步通信中一个字符正式发送之前，先发送一个起始位，低电平，宽度为 1 位；紧跟其后的就是要传送的数据，数据位占 5~8 位，总是低位先传送。可在数据位内设 1 位奇偶校验位，可以选择奇校验或偶校验，也可以不设置校验位；结束时发一个停止位，高电平，宽度为 1 位、1.5 位或 2 位。一个字符传送结束，可以接着传送下一个字符，字符之间也可有空闲位，都是高电平。

将一组字符编码称为一帧信息，一帧数据代表一个特定的字符，一帧数据每个位的时间间隔长度是固定的。异步通信的字符是一帧一帧发送的，而每帧字符的传送则依靠起始位来同步。发送端发送完一个字符后，可以连续发送或者随机发送。连续发送是指前一个字符的停止位发送后，立即发送下一帧的起始位，继续发送下一个字符。随机发送指的是上一个字符发送完停止位后，发送高电平，表示通信两端没有进行数据通信，空闲。发送下一个字符时，再用起始位进行同步。由于串行通信采用起始位作为同步信号，接收端总是在接收到每个字符的头部即起始位处进行一次重新定位，保证每次采样对应一个数位。所以异步传送的发送器和接收器不必用同一个时钟，而是各有自己的局部时钟，只要使用同一标称频率，略有偏差也不会导致数据传送错误。

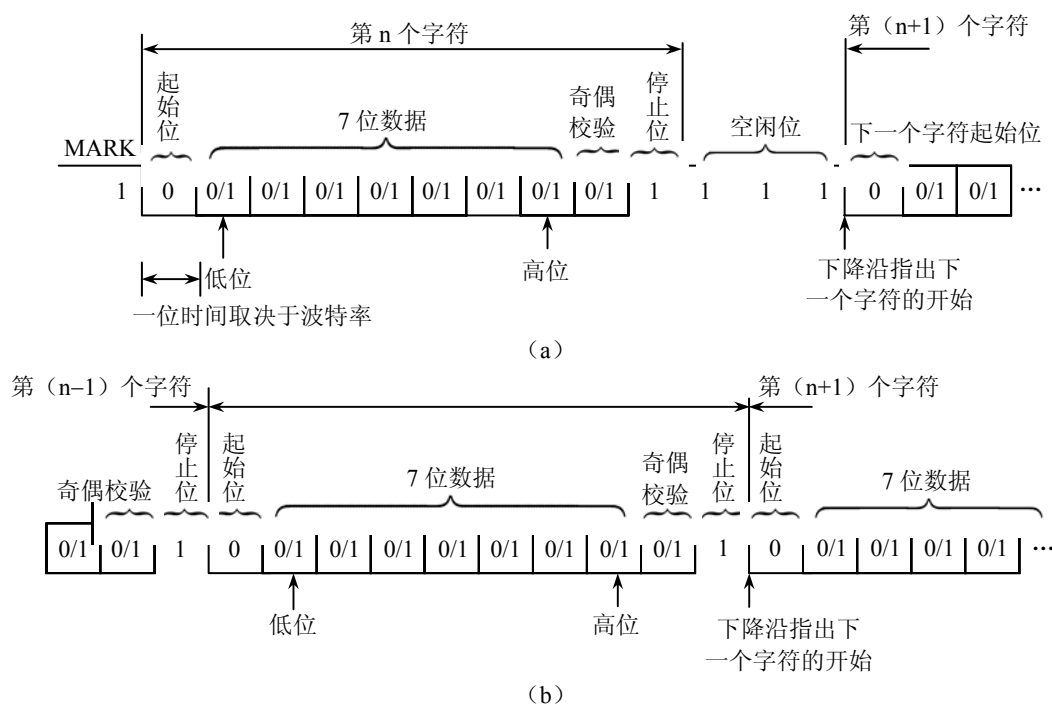


图 5-50 异步通信的格式

2) 传送速率: 传送速率是指每秒钟传送的二进制位数, 通常称为波特率 (Baud Rate), 其单位为波特 (Baud)。国际上规定了标准波特率系列, 最常用的标准波特率是: 110 波特、300 波特、600 波特、1200 波特、1800 波特、2400 波特、4800 波特、9600 波特和 19200 波特。波特率是衡量传输通道频带宽度的指标。波特率越高, 数据传输速率越快, 对传输通道的频带要求越宽。

例如, 在某个异步串行通信系统中, 数据传送速率为 960 字符/秒, 每个字符包括一个起始位、8 个数据位和一个停止位, 则波特率为 $10 \times 960 = 9600$ (位/秒) = 9600 (波特)。

每秒传输的二进制位数称为比特率, 当进行二进制数码传输时, 比特率与波特率在数值上相等。

3) 时钟频率: 在串行通信中, 发送方有输出移位寄存器, 接收方有接收 (输入) 移位寄存器。在波特率指定后, 输入移位寄存器/输出移位寄存器在接收时钟/发送时钟控制下, 按指定的波特率进行移位。数据传输过程中, 根据传送的波特率来确定发送时钟和接收时钟的频率。在异步传送中每发送一位数据的时间长度由发送时钟决定, 每接收一位数据的时间长度由接收时钟决定, 它们和波特率之间有如下的关系:

$$\text{时钟频率} = n \times \text{波特率}$$

式中的 n 叫做波特率系数或波特率因子, 它的取值可以为 1、16、32 或 64。

2. 同步串行通信 SYNC (Synchronous Data Communication)

由于异步串行通信要在每个数据前后附加起始位和停止位, 在传输过程中会有 20% 左右的冗余时间, 因此通信效率不高。同步串行通信不再使用起始位来标识字符的开始, 而是用一串特定的二进制序列, 称为同步字符, 去通知接收器串行数据第一位何时到达。同步串行数据

信息以连续的形式发送，每个时钟周期发送一位数据。数据信息间不留空隙，数据信息后是两个错误校验字符。接收方接收到数据后，要用校验字符对接收到的数据进行校验，以判断数据是否正确。对按以上结构构成的一组数据块称为一帧信息。一帧信息可包含成百上千个字符并且字符个数是可变的，由用户来设置。同步通信采用的同步字符的个数不同，存在着不同的格式结构，具有一个同步字符的数据格式称为单同步数据格式，有两个同步字符的数据格式称为双同步数据格式，如图 5-51 所示。

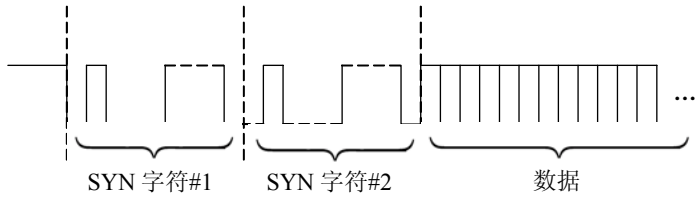


图 5-51 同步字符

在接收过程中，接收设备首先搜索同步字符，在接收到同步字符后，开始接收数据，接收完毕后，用校验字符进行校验。同步传送的速度高于异步，但它要求有时钟来实现发送端与接收端之间的同步，故硬件复杂。CRC（Cyclic Redundancy Check）为循环冗余校验码，用于校验在信息传输过程中是否有错误，是保证传输可靠性的一种重要的手段。

3. 串行通信数据传输方式

在串行通信中，按照同一时刻数据流的方向可分为三种基本传送模式：单工方式、半双工方式和全双工方式。

（1）单工（simplex）方式

单工传送方式仅允许在一个固定的方向上传输数据。采用该方式时，已经确定了通信两点中的一点为接收端，另一点为发送端，即由设备 A 传送到设备 B。在这种传送模式中，A 只作为发送器，B 只作为接收器。反之，是不行的，如图 5-52 所示。

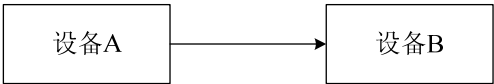


图 5-52 单工方式示意图

（2）半双工（Half-duplex）方式

半双工传送方式是允许在设备 A 和设备 B 之间使用同一条传输线来交替地传送数据。即设备 A 为发送器发送数据到设备 B，设备 B 为接收器。也可以将设备 B 作为发送器发送数据到设备 A，设备 A 为接收器。由于 A、B 之间仅有一根数据传送线，它们都具有独立的发送和接收能力，所以在同一个时刻只能进行一个方向的传送。半双工的连接方式如图 5-53 所示。

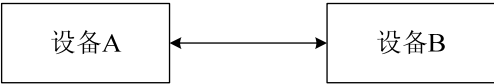


图 5-53 半双工示意图

（3）全双工（Full-duplex）方式

全双工传送方式是数据的发送和接收采用不同的传输线，数据可以在两个方向同时传送。

即设备 A 可发送数据到设备 B，设备 B 也可以发送数据到设备 A，它们都具有独立的发送和接收能力，并有两条传送线。全双工的连接方式如图 5-54 所示。

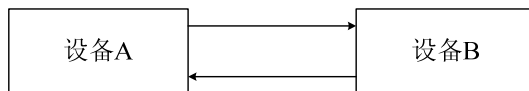


图 5-54 全双工示意图

4. 异步串行接口标准

RS-232C 是异步串行通信使用最广泛的总线接口标准。RS-232C 是美国电子工业协会 (Electronic Industry Association, EIA) 于 1962 年公布并于 1969 年修订的串行接口标准。已经成为国际通用的标准串行接口。1987 年 1 月，RS-232C 经修改后，正式改名为 EIA-232D。

在串行通信中，数据终端设备 (DTE) 中的数据要求通过数据通信设备 (DCE) 传送。数据终端可以是计算机或显示器；数据通信设备常指调制器和解调器，二者一起被称为 Modem。目前，RS-232C 成为数据终端设备 (DTE) 与数据通信设备 (DCE) 的标准接口。利用 RS-232C 接口不仅可以实现远距离通信，还可以近距离连接两台计算机或电子设备。在 PC 机上配置的 COM1 和 COM2 两个串行接口，都采用了 RS-232C 标准。

调制解调器 (Modem) 是一种计算机硬件，是计算机系统与通信线路之间的转换电路。分为调制 (Modulating) 和解调 (Demodulating) 两部分功能。为了通过通信线路发送计算机处理的数字信号，必须先把数字信号转换为适合在通信线路上传送的脉冲信号，这是调制；经过通信线路传输后，这些脉冲信号在接收端被接收，并转换为计算机可处理的数字信号，这是解调。多数情况下，通信是双向的，具有调制和解调功能的器件设计在一个装置中，形成调制解调器。

(1) RS-232C 的机械特性

RS-232C 标准设有 25 条信号线，一般常见的 RS-232C 接口大多以 D 型 25 针的连接座 (DB-25) 与外界相连，如图 5-55 所示。表 5-9 列出了所有引脚与信号之间的对应关系。而在实际使用中，常常只需要用到 25 条信号线中的 3~9 条信号线，因此 RS-232C 还有一种 9 针的 D 型连接器，其引脚信号规定如图 5-56 所示。

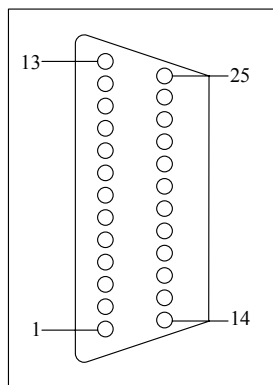


图 5-55 RS-232C D 型 25 针

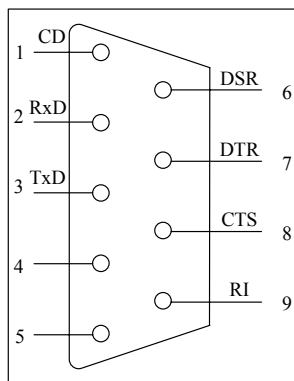


图 5-56 D 型 9 针

表 5-9 RS-232C 引脚与信号对应关系

引脚	含义	引脚	含义
1	保护地	14	第二通道发送数据
2	发送数据 TxD	15	传输信号单元定时 TxC
3	接收数据 RxD	16	第二通道接收数据
4	请求发送 RTS	17	接收信号单元定时 RxC
5	清除发送 CTS	18	未分配
6	数据准备好 DSR	19	第二通道请求数据
7	信号地 GND	20	数据终端准备就绪 (DTR)
8	接收线路建立检测 (CD)	21	信号质量检测
9	接收线路建立检测 (CD)	22	振铃指示 (RI)
10	接收线路建立检测 (CD)	23	数据信号速率选择 (DSRD)
11	未分配	24	发送信号单元定时
12	第二通道接收线信号检测	25	未分配
13	第二通道清除发送		

TxD (Transmitted Data): 发送数据, 串行通信中数据的发送端。

RxD (Received Data): 接收数据, 串行通信中数据的接收端。

RTS (Request To Send): 请求发送, 当数据终端设备准备好送出数据时, 就发出 RTS 有效, 用于通知数据通信设备准备接收数据。

CTS (Clear To Send): 清除发送, 当数据通信设备准备好接收数据终端设备的传送数据时, 发出该有效信号来响应 RTS 信号, 数据通信设备允许发送数据。

DSR (Data Set Ready): 数据准备好, 表示数据通信设备已接通电源连接到通信线路上, 并处于数据传输方式, 而不是处于测试方式或断开状态。

DTR (Data Terminal Ready): 数据终端准备就绪, 当数据终端设备加电时, 该信号就有效, 表明数据终端设备就绪。

CD/DCD (Carrier Detected): 接收线路建立检测, 即载波检测。载波是用于传输数据的模拟波形信号。当本地调制解调器接收到来自对方的载波信号时, 就从该引脚向数据终端设备提供有效信号。

TxC (Transmitter Clock): 传输信号单元定时, 发送器时钟, 控制数据终端发送串行数据的时钟信号。

RxC (Receiver Clock): 接收信号单元定时, 接收器时钟, 控制数据终端接收串行数据的时钟信号。

RI (Ringing Indicator): 振铃指示, 在调制解调器接收到对方的拨号信号期间, 该信号作为电话铃响的指示, 保持有效。

GND (Ground): 信号地, 为所有的信号提供一个公共的参考电平。

保护地: 是一个起屏蔽作用的接地端, 一般连接到设备的外壳和机架上, 必要时连接到大地。

(2) RS-232C 与 TTL 转换

RS-232C 接口采用的是负逻辑，其逻辑电平与 TTL 和 MOS 的电平规定不同，不能兼容，因此，为了能够使计算机接口与终端的 TTL 器件连接，必须在 RS-232C 与 TTL 电路之间进行电平和逻辑转换。实现这种转换可用分立元件，也可用集成芯片。实际中一般使用专用集成发送器电路 MC1488（或 SN75150/75188 等）和接收器电路 MC1489（或 SN75154/75189），如图 5-57 所示。

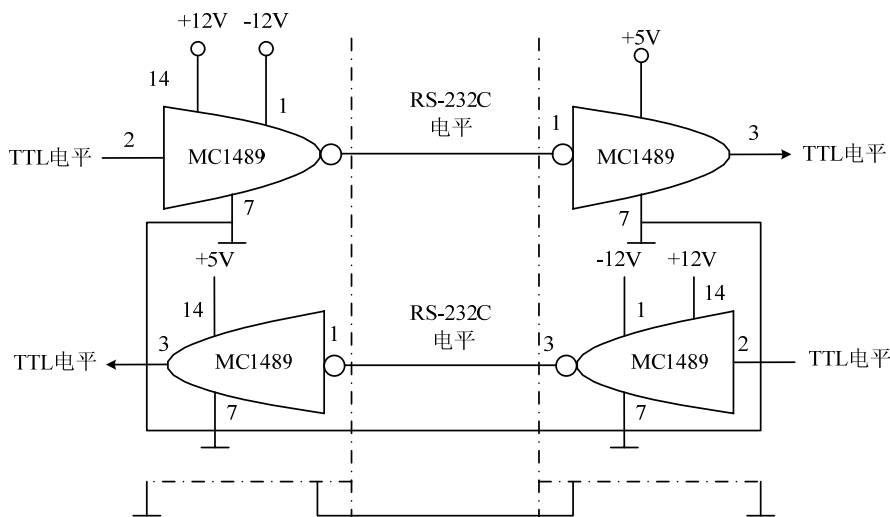


图 5-57 RS-232C 与 TTL/MOS 间的电平变换

(3) RS-232C 应用

利用 PC 的 RS-232C 接口，可以方便地与另一台 PC 机或单片机系统，如 MCS51 系统之间相互传输数据。

1) RS-232C 串口通信接线方法。虽然标准接口的信号线很多，但由于 RS-232C 是全双工通信，故 RS-232C 通信接口的信号线有近距离和远距离两种连接方法。

近距离（传输距离小于 15m）线路连接比较简单，采用三线制就可以了。三线就是发送数据线 TxD、接收数据线 RxD 及信号地线 GND。连接时，双方的地线直接相连，收发数据线交叉相连。如图 5-58 所示，一个 PC 与 MCS51 单片机相连，由于单片机是 TTL 电平，故需要电平转换电路。

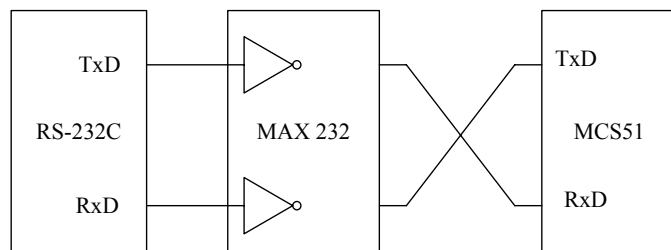


图 5-58 双机近距离通信

在进行远距离通信时，通信线路使用公用电话网，因为电话线上只能传输音频模拟信号，而计算机传送的是数字信号，故需要在通信双方加 Modem 进行数字信号与模拟信号之间的转换。双机远距离通信连接如图 5-59 所示。发送方将计算机发送的数字信号用调制器转换为模拟信号，送到电话线路上；接收方将接收到的模拟信号由解调器转换为数字信号，送计算机处理。

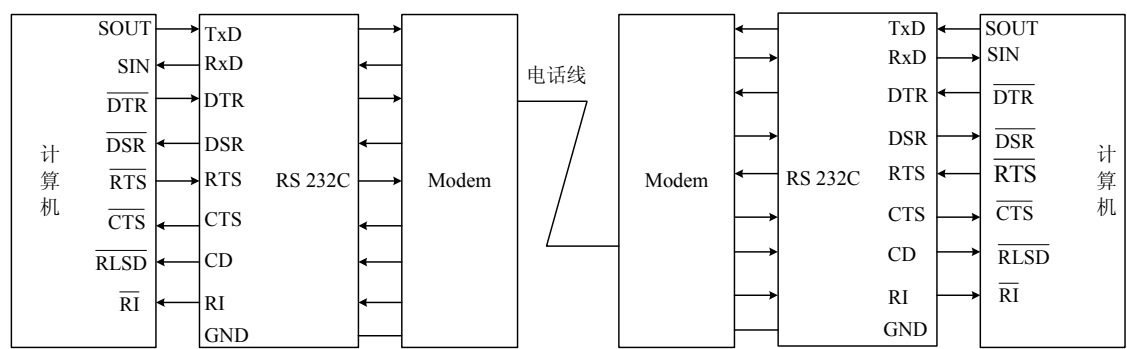


图 5-59 双机远距离通信

DTR 和 DSR 是一对握手信号，当甲方计算机准备就绪时，向 Modem 发送 DTR。乙方 Modem 接收到 DTR 后，若同意通信，则向甲方计算机回送 DSR，于是“握手”成功。

RTS 和 CTS 也是一对握手信号，当甲方计算机准备发送数据时，向 Modem 发送 RTS。乙方 Modem 接收到 RTS 后，若同意接收，则向甲方计算机回送 CTS，于是“握手”成功，甲方开始传送数据，乙方接收数据。

2) 软件编程。设通信双方有一台计算机为 PC，使用 COM1 端口。在串行通信中，必须首先设定通信双方所使用字符串的数据结构，才能进行软件编程。这里设数据在接收和发送的字符串中，序号为 0 的字节为数据长度，其后的字节为所接收的数据。设 COM1 端口的传输率为 2400 波特，字长为 8 位，1 位停止位，无奇偶校验。

对于接收过程，程序首先用“INT 14H, AH = 3”来获得 COM1 端口的状态，如果检测到“数据准备好”位有效，表明 COM1 端口接收到一个数据，则用“INT 14H, AH = 2”功能，将字符读到 AL 寄存器。

主程序：

BUFFER DB 100DUP(?)

```
...
MOV AH, 0
MOV AL, 0A3H
MOV DX, 0
INT 14H
```

```
...
CALL RECEIVE ;接收第 0 号数据
TEST AH, 80H ;测试读是否成功
```

```

        JNZ REC_ERROR           ;不成功, 转到出错处理
        MOV CH, 0
        MOV CL, AL              ;CL 为接收字符串长度
        MOV BX, OFFSET BUFFER   ;BX 字符串首地址
        MOV [BX], AL            ;存数据长度
REC_LOOP1: INC BX                ;指针加 1
        CALL RECEIVE            ;接收数据
        TEST AH, 80H
        JNZ REC_ERROR
        MOV [BX], AL            ;存数据
        LOOP REC_LOOP1          ;循环
REC_ERROR: ...                  ;接收出错处理

```

子程序:

```

RECEIVE PROC FAR                ;接收数据子程序, 出口 AL, AH
REC_CHECK: MOV AH, 3            ;读通信端口状态字
        MOV DX, 0
        INT 14H
        TEST AH, 01            ;测试数据准备好位
        JZ REC_CHECK           ;数据未准备好, 再读状态字
        MOV AH, 2              ;读通信端口数据
        MOV DX, 0
        INT 14H
        RET
RECEIVE ENDP

```

发送程序与接收程序类似。程序首先用“INT 14H,AH = 3”来获得 COM1 端口的状态, 如果检测到“发送保存寄存器空”位有效, 表明可以写入一个数据到 COM1, 就用“INT 14H,AH = 1”功能, 将字符写到 COM1 的发送保存寄存器中。

主程序:

```

        MOV BX, OFFSET BUFFER   ;BX 为字符串首地址
        MOV AL, [BX]            ;取数据长度
        MOV CL, AL
        MOV CH, 0                ;CX 为发送数据长度
        CALL SEND                ;发送数据长度
        TEST AH, 80H            ;测试发送是否成功
        JNZ SEND_ERROR          ;不成功, 转到出错处理
SEND_LOOP1: INC BX               ;指针加 1
        MOV AL, [BX]            ;取数据
        CALL SEND                ;发送数据

```

```

TEST  AH, 80H                ;测试发送是否成功
JNZ   SEND_ERROR
LOOP  SEND_LOOP1             ;循环
    .....
SEND_ERROR    .....
子程序:
;发送数据子程序, 输入参数: AL, 输出参数: AL, AH
SEND  PROC FAR
    PUSH  AX
SEND_CHECK:MOV  AH, 3          ;读通信端口状态字
    MOV  DX, 0
    INT  14H
    TEST AH, 20H              ;测试“发送保存寄存器空”位
    JZ   SEND_CHECK           ;发送保存寄存器满, 再读状态字
    POP  AX
    MOV  AH, 2                ;发送数据
    MOV  DX, 0
    INT  14H
    RET
SEND  ENDP

```

5.4.2 16550 的内部结构和外部引脚

16550 芯片是 National Semiconductor 公司推出的 PC 机上常用的一种可编程的通用异步接收/发送器 (Universal Asynchronous Receiver Transmitter, UART), 与早期的 Intel 8250 芯片完全兼容, 并具有更高的性能, 用在 PC 机的 COM 端口。与 8250 不同之处有: ①8250 的发送和接收数据缓冲器只有一个字节, 每发送或接收一个字节都要求 CPU 来干预, 而 16550 有 16 个字节的 FIFO 发送和接收数据缓冲器, 它可以连续发送或接收 16 个字节的数据; ②8250 的最大通信速率为 19200b/s, 而 16550 的最大通信速率可达 115200b/s, 因此它是高速系统通信接口的理想器件。

早期的 IBM PC/XT 中 UART 芯片是 8250, 后续 PC 则采用兼容的 16450 和 16550。现在 32 位 PC 芯片组中使用的是与 16550 兼容的逻辑电路。16550 经过电平转换接口电路, 能够提供 RS-232C 标准规定的信号线, 以便与 Modem 进行联络和传送, 使得通信可以方便地在电话系统或无线通信系统中实现。

16550 与 8250 芯片内部都包含发送控制电路和接收控制电路, 可实现全双工通信; 都支持异步通信; 都具有控制 Modem 功能和完整的状态报告功能。此外, 16550 增加了 FIFO 模式, 使得 16550 更容易与微处理器之间实现高速数据通信。

本节及以后几节将以 16550 为例, 讨论异步串行接口芯片的内部结构、引脚信号、初始化编程以及应用实例。

1. 16550 的内部结构

由图 5-60 所示 16550 内部结构框图可知, 它由数据总线缓冲器、读/写控制逻辑、波特率发生器、数据接收器、数据发送器、FIFO 控制寄存器、线路控制寄存器、线路状态寄存器、Modem 控制逻辑、中断控制逻辑以及除法等组成。

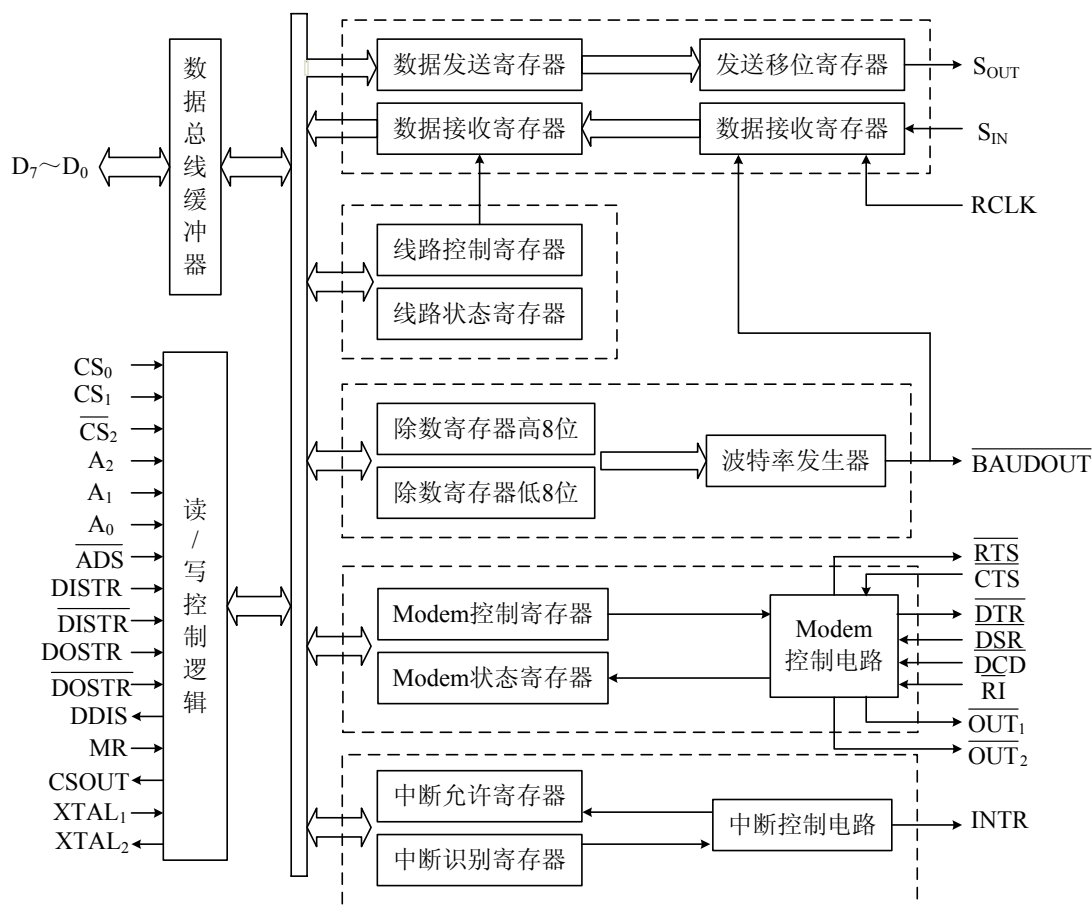


图 5-60 16550 内部结构

(1) 数据总线缓冲器

8 位双向三态缓冲器, 用于连接系统数据总线和 16550 内部总线。数据线 $D_7 \sim D_0$ 与 CPU 的数据线连接, 构成 CPU 与 16550 之间信息传送通道。

(2) 读/写控制逻辑

接收来自 CPU 的读/写控制信号和端口选择信号, 用于控制 16550 内部寄存器的读/写操作。

(3) 波特率发生器和除法寄存器

16550 使用 18.432MHz 的基准输入时钟, 分频后产生的内部基准时钟作为接收器和发送器的同步时钟, 以控制接收移位寄存器和发送移位寄存器的移位操作。发送或接收串行数据时使用的时钟频率是数据传输波特率的 16 倍。当设定串行数据传输速率时, 需要根据其使用的波特率确定分频器的系数值。这个系数值将被写入除法寄存器。

$$\text{系数值} = 18.432\text{MHz} / (\text{波特率} \times 16)$$

(4) 数据接收器

数据接收器由接收缓冲区 FIFO、接收缓冲寄存器、接收移位寄存器和接收定时控制逻辑等电路组成。

数据接收器的基本工作原理如下：在异步方式接收数据时，接收移位寄存器将外部送至 S_{IN} 信号引脚的串行数据进行移位输入。当 S_{IN} 输入信号变低后，连续采样 8 个 RCLK 时钟周期，如果都是低电平，则认为是起始位。从起始位开始，每隔 16 个 RCLK 时钟周期对 S_{IN} 输入的数据进行一次采样，直到完整的字符格式结束。接收定时控制电路按照预先规定的数据格式和波特率自动去除起始位、奇偶校验位和停止位，把串行数据输入，在接收移位寄存器接收完一个字符后，控制接收移位寄存器将串行数据转换为并行数据，然后送入接收缓冲寄存器和接收数据缓冲区 FIFO 中。当 FIFO 在接收到 16 字节数据后，芯片发出 $RxRDY$ 信号，请求 CPU 读取。

(5) 数据发送器

数据发送器由发送保持寄存器、数据发送缓冲区 FIFO、发送移位寄存器和发送定时控制逻辑等电路组成。

数据发送器的基本工作原理如下：CPU 将要发送的数据写入数据发送缓冲区 FIFO，当芯片发送数据时，数据发送器从 FIFO 取出一个字节数据，存入发送保持寄存器中，由发送定时控制电路控制发送移位寄存器将该数据进行并串处理，按照先前规定的格式加上起始位、奇偶校验位和停止位，编程串行数据帧后由 S_{OUT} 引脚输出串行数据。当数据进入发送移位寄存器后，发送保持寄存器自动变空，缓冲区 FIFO 可以写入新的数据到发送保持寄存器。发送移位寄存器输出一个字符后，可继续接收下一个字符。当缓冲区 FIFO 的 16 字节数据均被取走后，芯片输出 $TxRDY$ 信号，可请求 CPU 输出新的数据到 16550。

(6) FIFO 控制寄存器

FIFO 控制寄存器用来控制接收 FIFO 缓冲区和发送 FIFO 缓冲区，格式如图 5-61 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
RT ₁	RT ₀	0	0	DMA	XMIT	REVC	EN

图 5-61 FIFO 控制寄存器格式

D₇D₆：接收器触发器值，D₇D₆ = 00，FIFO 中有 1B；D₇D₆ = 01，FIFO 中有 4B；D₇D₆ = 10，FIFO 中有 8B；D₇D₆ = 11，FIFO 中有 14B。

D₃：DMA 方式控制，D₃ = 0，用作 16450；D₃ = 1，FIFO 方式。

D₂：发送器复位，D₂ = 0，无效；D₂ = 1，复位。

D₁：接收器复位，D₁ = 0，无效；D₁ = 1，复位。

D₀：FIFO 允许位，D₀ = 0，禁止；D₀ = 1，允许。

(7) 线路控制寄存器

线路控制寄存器是一个 8 位寄存器，主要用于决定在串行通信时的数据通信格式。格式如图 5-62 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
DL	SB	ST	P	PE	S	L1	L0

图 5-62 线路控制寄存器格式

D₇: 允许除数锁存, 当允许对波特率产生器进行除数锁存时, 该位置 1, 在读/写其他寄存器时, 该位复位为 0。

D₆: 发送中断, 确定是否从 S_{OUT} 引脚发送一个中断 (一个中断定义为至少连续两帧逻辑 0 数据), D₆ = 0, 不发送; D₆ = 1, 发送。

D₅: 保留位, D₅ = 0, 保留校验关闭; D₅ = 1, 保留校验打开。

D₄: 校验类型, D₄ = 0, 奇校验; D₄ = 1, 偶校验。

D₃: 检验允许, D₃ = 0, 不校验; D₃ = 1, 允许校验。

D₂: 停止位长, D₂ = 0, 1 位停止位; D₂ = 1, 1.5 或 2 位停止位。

D₁D₀: 数据长度, D₁D₀ = 00, 数据 5 位长; D₁D₀ = 01, 数据 6 位长; D₁D₀ = 10, 数据 7 位长; D₁D₀ = 11, 数据 8 位长。

(8) 线路状态寄存器

线路状态寄存器是一个 8 位寄存器, 提供了 UART 和远程 UART 之间传输数据的状态, 包含了接收器和发送器的错误条件和状态信息, 格式如图 5-63 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
ER	TE	TH	BI	FE	PE	OE	DR

图 5-63 线路状态寄存器格式

D₇: 接收 FIFO 中出现任何错误的时刻, D₇ = 0, 无错误; D₇ = 1, FIFO 中至少出现一个错误。

D₆: 发送 FIFO 和发送移位寄存器两者为空的时刻, D₆ = 0, 发送器不空; D₆ = 1, 发送器空。

D₅: 发送保持寄存器, D₅ = 0, 等待发送; D₅ = 1, 数据发送就绪。

D₄: 线路中断标志, D₄ = 0, 无中断; D₄ = 1, 正在接收中断。

D₃: 结构错误标志, 表示一帧数据的起始位与停止位不在它们正确的位置上, 当接收器正在以不正确的波特率接收数据时会出现结构错误。D₃ = 0, 无帧格式错误; D₃ = 1, 帧格式错误。

D₂: 奇偶校验错误标志, 表示接收的数据中包含错误的校验, 通常在信息传输中遇到噪声干扰时容易产生校验错误。D₂ = 0, 无校验错误; D₂ = 1, 校验错误。

D₁: 超限错误标志, 表示数据已超过内部的接收 FIFO 缓冲器, 超限错误仅出现在接收 FIFO 满之前软件从 UART 读数据失败时。D₁ = 0, 无超限错误; D₁ = 1, 超限错误。

D₀: 数据就绪标志, 一旦在接收 FIFO 中有数据, 该位就置位为 1。

(9) Modem 控制逻辑

Modem 控制逻辑由 Modem 控制寄存器和 Modem 状态寄存器组成。在串行通信中, 当通信双方距离较远时, 为增强系统的抗干扰能力, 防止传输数据发生畸变, 需要在通信双方使用 Modem。发送方将数字信号经 8250 送至 Modem 进行调制, 转换为模拟信号, 送到电话线上进行传输; 接收方 Modem 对接收到模拟信号进行解调, 转换为数字信号, 经 16550 送 CPU 处理。Modem 状态寄存器和控制寄存器格式分别如图 5-64 和图 5-65 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
DCD	RI	DSR	CTS	Δ DCD	Δ RI	Δ DSR	Δ CTS

图 5-64 Modem 状态寄存器格式

D₇、D₆、D₅、D₄: 如前 RS-232C 引脚信号所讲。

D₃: 数据载波检测改变。

D₂: 响铃指示改变。

D₁: 通信设备就绪改变。

D₀: 允许发送改变。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0	LOOP	$\overline{\text{OUT}}_2$	$\overline{\text{OUT}}_1$	RTS	DTR

图 5-65 Modem 控制寄存器格式

D₄: 字检控制位, D₄ = 0, 正常工作; D₄ = 1, 自收自发。

D₃、D₂: 输出信号, 为 0 时, 输出低电平; 为 1 时, 输出高电平。

D₁: 请求发送, D₁ = 0, 无效; D₁ = 1, 有效, 请求发送。

D₀: 数据终端就绪, D₀ = 0, DTR 未准备好; D₀ = 1, DTR 准备好。

(10) 中断控制逻辑

16550 的中断控制逻辑由中断允许寄存器、中断标识寄存器和中断逻辑组成。

中断允许寄存器是用来设置 16550 的某些部件允许或禁止中断, 格式如图 5-66 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0	0	EM	EL	ET	ER

图 5-66 中断允许寄存器格式

D₃: 允许调制解调器中断。

D₂: 允许线路中断。

D₁: 允许发送器中断。

D₀: 允许接收器中断。

以上 4 位均是在等于 1 时允许, 等于 0 时禁止。其中线路中断包括超限错误、奇偶校验错误、结构错误以及间断等由中断源引起的中断。

中断标识寄存器为 8 位, 高 4 位为 0, 低 4 位用来指示是否有正在悬挂的中断及中断类型, 格式如图 5-67 所示。

D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0	0	ID	ID	ID	PN

图 5-67 中断标识寄存器格式

D₀: 是否中断悬挂, D₀ = 0, 中断悬挂; D₀ = 1, 无中断。

D₃~D₁: 中断标识位。中断标识码如表 5-10 所示。

表 5-10 16550 中断标识码

ID			PN	优先级	类型
D ₃	D ₂	D ₁	D ₀		
0	0	0	1		无中断
0	1	1	0	1	接收器错误
0	1	0	0	2	接收器数据有效
1	1	0	0	2	字符暂停, 至少 4 个字符时间内没有从接收器 FIFO 取数据
0	0	1	0	3	发送器空
0	0	0	0	4	调制解调器状态

2. 16550 的外部引脚

16550 外部引脚图如图 5-68 所示, 16550 是具有 40 个引脚的双列直插封装集成芯片, 各引脚功能如下所述:

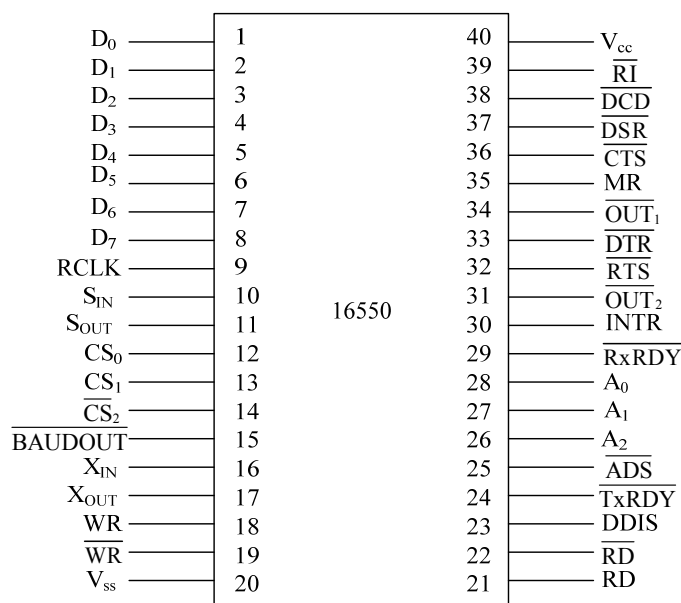


图 5-68 16550 外部引脚图

D₇~D₀: 双向数据线, 与系统数据总线相连, 供 CPU 对 16550 进行读/写操作、传送数据、控制信号和状态信息交换。

$\overline{CS_2}$ 、CS₁、CS₀: 片选输入信号。当 $\overline{CS_2} = 0$, CS₁ = 1, CS₀ = 1 时, 16550 被选中。

X_{IN}、X_{OUT}: 主时钟连接端。这两端与外部晶体振荡器电路连接。

A₂~A₀: 端口选择信号。用来确定片内 10 个寄存器的端口地址。具体确定关系见表 5-11。

表 5-11 16550 片内寄存器确定

A ₂ A ₁ A ₀	寄存器	A ₂ A ₁ A ₀	寄存器
0 0 0	数据接收缓冲器（读） 数据发送缓冲器（写）	1 0 0	Modem 控制寄存器
0 0 1	中断允许寄存器	1 0 1	线路状态寄存器
0 1 0	中断识别寄存器（读） FIFO 控制寄存器（写）	1 1 0	Modem 状态寄存器
0 1 1	线路控制寄存器	1 1 1	暂存寄存器

RCLK：接收时钟。该引脚是加载到 UART 接收器的时钟输入，该引脚的频率为接收信号波特率的 16 倍。

ADS：地址选通输入信号。当 $\overline{\text{ADS}} = 0$ 时，将片选信号 $\overline{\text{CS}}_2$ 、 CS_1 、 CS_0 和地址信号 $A_2 \sim A_0$ 输入锁存。在 Intel 系统中，该信号不需要直接接地。

WR、 $\overline{\text{WR}}$ ：写信号，用于 CPU 写入控制字或数据到 16550，两个都可用。

RD、 $\overline{\text{RD}}$ ：读信号，用于 CPU 读取 16550 内部的状态字或数据，两个都可用。

BAUDOUT：波特率输出，是由发送器电路中的波特率产生器产生的发送时钟信号。此信号通常与 RCLK 相连，使接收时钟与发送时钟相同。该引脚输出的时钟等于主时钟频率除以 16550 内部波特率产生器中的除数后所得到的频率信号，是发送波特率的 16 倍。

S_{OUT}：串行数据输出引脚。

S_{IN}：串行数据输入引脚。

$\overline{\text{TxRDY}}$ ：发送器准备好信号，输出，低电平有效。当有效时，表示发送器已准备好接收 CPU 送来的字符数据，通知 CPU 可以向 16550 输出数据。

$\overline{\text{RxRDY}}$ ：接收器准备好信号，输出，低电平有效。该信号可用来通知 CPU，16550 的数据接收缓冲器有数据可向 CPU 传送。

DDIS：禁止数据传送信号，输出，高电平有效。当 $\text{DDIS} = 1$ 时，禁止 CPU 对 16550 内部寄存器的读取；当 $\text{DDIS} = 0$ 时，表示 CPU 正在读 UART。

INTR：中断请求输出信号，高电平有效。当 16550 中断允许时，接收错误、接收数据寄存器满、发送数据寄存器空以及 Modem 的状态均可产生有效的 INTR 中断请求。

MR：主复位信号，它与系统总线的 RESET 信号连接。当 16550 的 MR 引脚输入复位信号后，其内部电路复位。

$\overline{\text{RTS}}$ ：请求发送信号，输出低电平有效。用于通知 Modem，16550 要求发送。它由命令字的 D₅ 位置 1 来使其有效。

CTS：清除发送即允许发送信号，输入低电平有效。是 Modem 对 16550 的 RTS 信号的响应，它有效时 16550 才能发送数据。

$\overline{\text{DTR}}$ ：数据终端就绪，低电平有效。该输出信号表示数据终端 16550 准备起作用，它是向外发送数据的请求信号。

$\overline{\text{DSR}}$ ：数据装置就绪，低电平有效。该输入信号表示 Modem 或数据装置准备进行接收数据的操作。

DCD：载波信号检测到输入信号，低电平有效。表示传输线上有信号传输。

$\overline{\text{RI}}$: 振铃指示, 低电平有效, 输入。表示 Modem 已经接收到一个电话振铃信号。

$\overline{\text{OUT}}_1$ 、 $\overline{\text{OUT}}_2$: 用户编程定义的输出信号引脚, 在系统需要时可以为 Modem 或任何设备提供信号。

5.4.3 16550 的初始化编程

1. 初始化编程

串口初始化程序必须在系统复位之后, 在 16550 工作以前进行。16550 的初始化编程主要用来设置 16550 的通信格式、传输波特率、是否使用中断、是否自检测试等操作。需要编程的寄存器有: 除法寄存器、线路控制寄存器、Modem 控制寄存器和中断允许寄存器。由于写入除法寄存器时要求线路控制寄存器的 $D_7 = 1$, 建立访问除法寄存器标志, 因此在写除法寄存器之前先将线路控制寄存器中的 D_7 位置位。在 PC 机系统中, 为串口寄存器分配有专门的 I/O 端口地址, 通过这些端口地址可对 16550 内部的寄存器进行操作。表 5-12 给出了串行接口的 I/O 端口地址分配。

表 5-12 PC 机串口 I/O 端口地址表

DLAB	串口 1 地址	串口 2 地址	寄存器	读/写操作
0	3F8H	2F8H	发送保持寄存器	写
0	3F8H	2F8H	数据接收寄存器	读
1	3F8H	2F8H	除法寄存器低字节	读/写
1	3F9H	2F9H	除法寄存器高字节	读/写
0	3F9H	2F9H	中断允许寄存器	读/写
×	3FAH	2FAH	中断识别寄存器	读
×	3FAH	2FAH	FIFO 控制寄存器	写
×	3FBH	2FBH	线路控制寄存器	读/写
×	3FCH	2FCH	Modem 控制寄存器	读/写
×	3FDH	2FDH	线路状态寄存器	读
×	3FEH	2FEH	Modem 状态寄存器	读
×	3FFH	2FFH	暂存寄存器	读/写

在 PC 机中, 串口 1 分配的 I/O 端口地址为 3F8H~3FFH, 串口 2 分配的 I/O 端口地址为 2F8H~2FFH。下面通过实例来说明串口 1 的初始化。

(1) 设置通信格式

设通信格式为每字符 7 位数据、2 位停止位, 采用偶校验, 初始化程序段:

```
MOV AL, 00011110B    ;线路控制寄存器的控制字
MOV DX, 3FBH          ;线路控制寄存器端口地址
OUT DX, AL             ;写入线路控制寄存器
```

(2) 设置通信波特率

串口通信波特率需要设置为 115200 波特, 初始化程序段:

```
MOV AL, 10000000B     ;线路控制寄存器控制字 DLAB = 1
MOV DX, 3FBH          ;线路控制寄存器端口地址
```

OUT DX, AL	;写入线路控制寄存器
MOV AL, 10	;产生 115200 波特率的除数系数低字节
MOV DX, 3F8H	;除数低字节寄存器端口地址
OUT DX, AL	;写入除数低字节
MOV AL, 00H	;除数高字节
INC DX	;指向除数高字节寄存器端口地址
OUT DX, AL	;写入除数高字节

(3) 设置操作方式

设串口需要自检测试，初始化程序段：

MOV AL, 00011011B	;Modem 控制寄存器控制字
MOV DX, 3FCH	;Modem 控制寄存器端口地址
OUT DX, AL	;写入 Modem 控制寄存器

(4) 设置中断允许

设允许接收数据中断，初始化程序段：

MOV AL, 00000001B	;中断允许控制寄存器控制字
MOV DX, 3F9H	;中断允许控制寄存器端口地址
OUT DX, AL	;写入中断允许控制寄存器

16550 的初始化流程如图 5-69 所示。

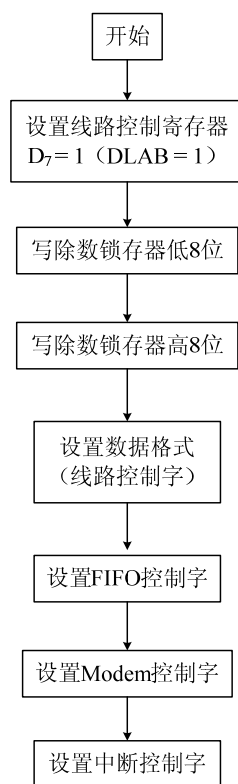


图 5-69 16550 初始化流程图

2. FIFO 工作方式

图 5-70 所示 16550 中接收/发送 FIFO 用来提高数据传输速率,同时又避免了 8250 等存在的在下一帧数据到达接收/发送缓冲寄存器前,CPU 对接收/发送缓冲器中前一帧数据的读取操作必须结束或发送必须结束,否则会产生接收/发送数据的重叠错误。BIOS 不支持 FIFO 工作模式。

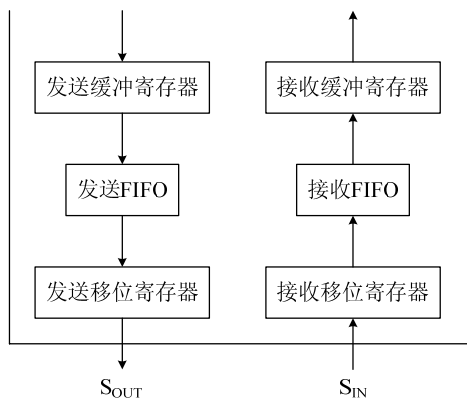


图 5-70 16550 FIFO 工作方式

(1) 接收 FIFO

当 FIFO 控制寄存器 D_0 位为 1 时,接收 FIFO 被启用。16550 将接收到的帧数据,按规定的帧数据格式,从移位寄存器移入接收 FIFO 缓冲器中,直到接收 FIFO 缓冲器中的字节数满足 FIFO 所设置的中断触发计数值时,16550 便输出中断请求信号。当 CPU 从 FIFO 中读取帧数据使字节数小于中断触发计数值时,中断请求信号便立即无效,接收 FIFO 缓冲器又可继续接收帧数据。当接收 FIFO 中已满,若移位寄存器中的内容仍向 FIFO 移入,此时立即产生接收重叠中断。

同所有的 UART 一样,可将 FIFO 工作模式设置为中断或查询方式。在中断方式下,可通过对 FIFO 控制寄存器编程改变中断触发值。当接收串行数据块结束,FIFO 中所存数据不能满足中断触发值,即在超时期限内没有接收到数据,这时 16550 按下式:

$$\text{超时 (ms)} = (\text{每帧中数据位数} / \text{波特率}) \times 4 \times 1000\text{ms}$$

产生一个超时中断。16550 每接收一帧数据,或 CPU 从 FIFO 中读取一个数据(包括超时中断),都会使用于超时的计数器自动复位。如果 16550 工作在查询方式,它和无 FIFO 模式的查询方式一样,通过查询线路状态寄存器 D_0 位,来判断 FIFO 中是否有数据。

(2) 发送 FIFO

当 FIFO 控制寄存器 D_0 位为 1 时,发送 FIFO 被启用,当发送 FIFO 为空时,产生发送保持寄存器中断,中断服务程序将需要发送的帧数据块写入 FIFO 中,通过发送移位寄存器开始发送。

5.4.4 16550 的应用实例

例 5-15 16550 的地址为 03F8H~03FFH。

(1) 16550 初始化

对 16550 进行初始化程序:

```
MOV  DX, 3FBH
MOV  AL, 80H
OUT  DX, AL                ;置位线路控制寄存器 D7, DLAB = 1
MOV  DX, 3FBH
MOV  AL, 60H
OUT  DX, AL                ;锁存除数寄存器低 8 位
INC  DX
MOV  AL, 0
OUT  DX, AL                ;锁存除数寄存器高 8 位
MOV  DX, 3FBH
MOV  AL, 0AH
OUT  DX, AL                ;初始化线路控制寄存器
MOV  DX, 3FAH
MOV  AL, 07H
OUT  DX, AL                ;初始化 FIFO 控制器
MOV  DX, 3FCH
MOV  AL, 03H
OUT  DX, AL                ;初始化 Modem 控制器
MOV  DX, 3F9H
MOV  AL, 0
OUT  DX, AL                ;写中断允许控制寄存器
```

(2) 以查询方式串行发送与接收数据

根据 (1) 中对 16550 进行的初始化, 发送数据的控制程序接在初始化程序之后。设采用查询方式发送数据, 且待发送数据的字节数放置于 BX 中, 待发送的数据顺序存放在以 SEND_DATA 为首地址的内存中。

发送数据的程序:

```
SEND: MOV  DX, 3FDH
      LEA  SI, SEND_DATA
WAIT1: IN   AL, DX
      TEST AL, 20H
      JZ   WAIT1
      PUSH DX
      MOV  DX, 3F8H
      MOV  AL, [SI]
      OUT  DX, AL
      POP  DX
      INC  SI
      DEC  BX
```

JNZ WAIT1

利用查询方式实现数据的接收，下面是 16550 接收一个数据的程序。

接收数据程序：

RECV: MOV DX, 3FDH

WAIT2: IN AL, DX

TEST AL, 1EH

JNZ ERROR

TEST AL, 01H

JZ WAIT2

MOV DX, 3F8H

IN AL, DX

AND AL, 7FH

该数据接收程序首先测试状态寄存器，看接收的数据是否有错。如果有错就转向错误处理程序 ERROR；否则，再看是否已收到一个完整的数据。如果收到一个完整的数据，则从 16550 的接收数据寄存器中读出，并取先前规定的 7 位数据，将其放入 AL 中。

例 5-16 在甲乙两台 PC 机之间使用串口 1 进行近距离通信。采用异步格式，字符长度为 8 位，2 位停止位，无校验，通信波特率为 9600 位/秒。双机同时运行该程序，一方键入的字符可以在另一方屏幕上显示。当按下 Esc 键时，程序运行结束。在 PC 机系统的串口电路中，16550 与系统总线以及 RS-232C 串口插件的硬件电路连接如图 5-71 所示。

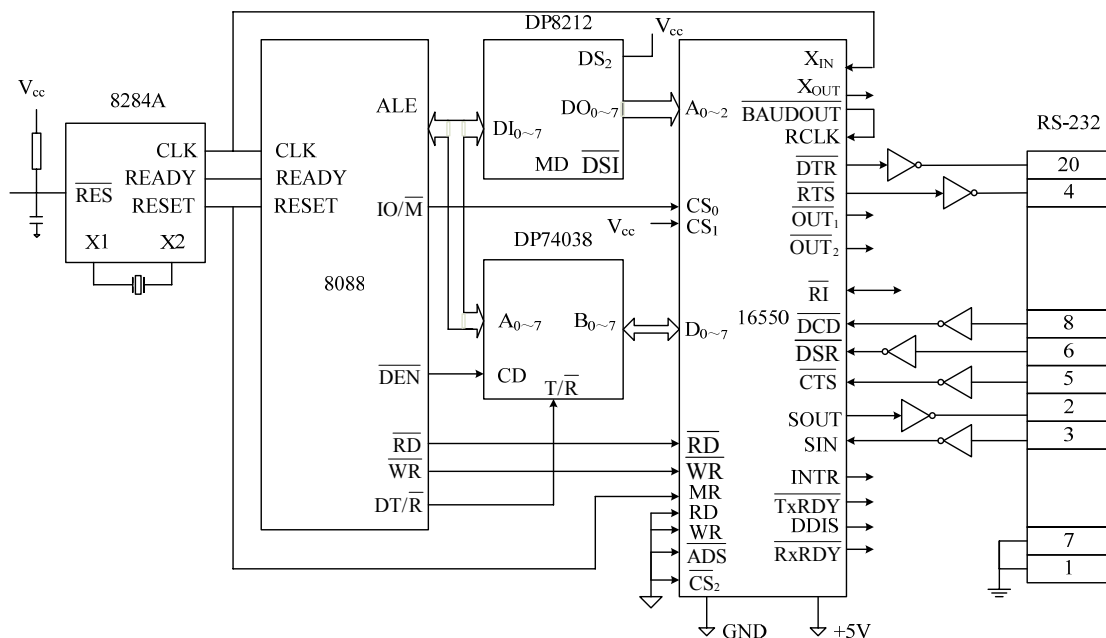


图 5-71 PC 机串口电路中硬件连接电路

在实际两台 PC 机近距离通信应用系统中，不需要对 16550 电路连接进行详细的了解，通常采用图 5-72 所示的简化信号连接。

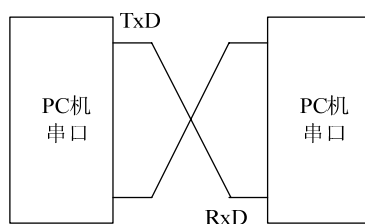


图 5-72 PC 机近距离串行通信方案

串行通信源程序:

```

STACK  SEGMENT  PARA  STACK
        DB  20H DUP(0)
STACK  ENDS
CODE    SEGMENT
        ASSUME CS:CODE, SS: SATCK
SATRT:  PUSH  DS
        MOV  AX, 0
        PUSH  AX                ;保存段前缀
        MOV  DX, 3FBH           ;线路控制寄存器端口地址
        MOV  AL, 10000000B       ;置位线路控制寄存器 D7, DLAB = 1
        OUT  DX, AL             ;写入线路控制寄存器
        MOV  DX, 3F8H           ;除数寄存器低字节端口地址
        MOV  AL, 120
        OUT  DX, AL             ;9600 波特率的除法系数值为 120
        INC  DX                 ;除数寄存器高字节端口地址
        MOV  AL, 0
        OUT  DX, AL             ;写入除数寄存器高字节
        MOV  DX, 3FBH
        MOV  AL, 00000111B       ;格式控制字, 无校验, 8 位数据, 2 位停止位
        OUT  DX, AL
        INC  DX                 ;Modem 控制寄存器端口地址
        MOV  AL, 00000011B       ;Modem 控制字, 置位 DTR 和 RTS
        OUT  DX, AL             ;写入 Modem 控制寄存器
        MOV  DX, 3F9H           ;中断允许寄存器端口地址
        MOV  AL, 0
        OUT  DX, AL             ;写入中断允许寄存器端口地址, 禁止所有串口中断
LOOP:   MOV  DX, 3FDH           ;线路状态寄存器端口地址
        IN  AL, DX              ;读线路状态寄存器
        TEST AL, 00011110B       ;测试错误状态位
        JNZ  ERROR             ;有错误, 转向错误处理程序 ERROR

```

```

TEST AL, 00000001B ;无错误, 测试数据接收寄存器是否满
JNZ RECEIVE ;如果满, 转向接收数据处理程序 RECEIVE
TEST AL, 00100000B ;如果不满, 测试发送保持寄存器是否空
JZ LOOP ;如果不空, 循环测试
MOV AH, 1 ;如果空, 读键盘缓冲区
INT 16H ;读取键盘缓冲区状态
JZ LOOP ;ZF = 1, 缓冲区没有字符, 循环等待
MOV AH, 0 ;ZF = 0, 读取键盘代码到 AL
INT 16H
CMP AL, 'ESC' ;判断读取的键值是否为 ESC
JZ RETURN ;如果是, 转 RETURN 退出程序
MOV DX, 3F8H ;发送保持寄存器端口地址
OUT DX, AL ;将键盘字符写入发送保持寄存器
JMP LOOP ;继续
RECEIVE: MOV DX, 3F8H ;数据接收寄存器端口
IN AL, DX ;读取数据接收寄存器
AND AL, 01111111B
PUSH AX
MOV BX, 0
MOV AH, 0EH ;BIOS 中断 INT 10H 的 14 号
INT 10H ;在当前光标处显示 AL 中的字符
POP AX
CMP AL, 0DH ;判断接收的字符是否为回车
JNZ LOOP ;不是, 跳转到 LOOP 继续
MOV AL, 0AH ;如果是回车, 再输出显示一个换行符
MOV BX, 0
MOV AH, 0EH
INT 10H
JMP LOOP
ERROR: MOV DX, 3F8H ;数据接收寄存器端口地址
IN AL, DX ;读取接收器内容
MOV AL, '?'
MOV BX, 0
MOV AH, 0EH
INT 10H
JMP LOOP
RETURN: RET ;结束程序, 返回
CODE ENDS
END START

```

5.5 模拟接口技术

目前，随着数字技术和自动化技术的飞速发展，计算机已广泛地应用于自动控制和测量等领域。在实际控制系统和工业生产过程中，需要进行加工和处理的信号可以分为模拟信号和数字信号两种类型。例如，传感器所检测的信号如温度、压力、流量、速度、湿度等物理量都是随着时间连续变化的模拟信号，而现在广泛使用的微型计算机内部都是采用二进制表示的数字量进行信号的输入、存储、传输、加工与输出，微型计算机只能处理数字量的信息。为了能用计算机对模拟信号进行采集、加工和处理，就需要把采集到的模拟信号转换成数字信号送入到计算机中，同样经过计算机处理后的数字信号，要对外部设备实现控制必须将数字信号转换成模拟信号。模拟接口的作用就是实现模拟信号和数字信号之间的转换。模/数（Analog to Digit, A/D）转换就是把输入的模拟量转换为数字量，供微型计算机处理；数/模（Digit to Analog, D/A）转化器就是将微型计算机处理后的数字量转换为模拟量形式的控制信号输出。

5.5.1 模拟输入输出系统概述

在工业生产过程中，一般被测试和控制的物理量多数为模拟量。如：温度、压强、流量等。在计算机处理过程中，需要将测试到的物理量经过传感器变换为电信号（模拟信号），再经过模/数转换器变换为数字信号，送计算机处理；计算机处理后的结果需要经过数/模转换器变换成模拟信号去驱动模拟调节机构工作。一个典型的计算机控制系统如图 5-73 所示。

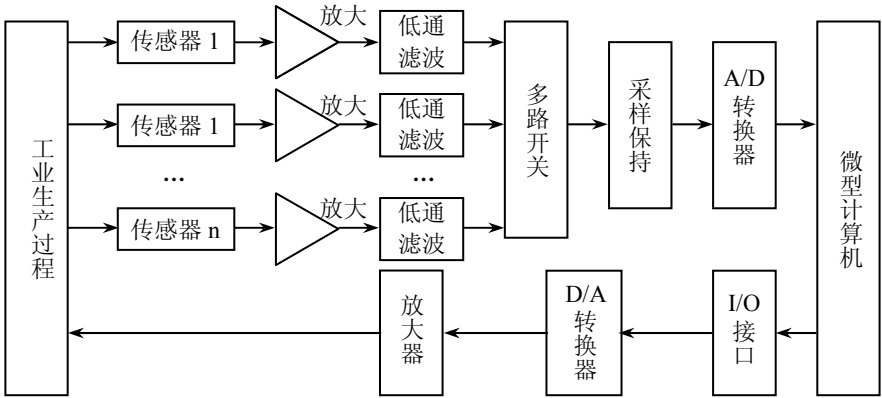


图 5-73 模拟输入输出控制系统

该系统由工业生产过程、模拟输入通道、模拟输出通道和计算机组成。其模拟输入通道由传感器、放大器、低通滤波器、多路开关、采样保持电路以及 A/D 转换器等部件组成，实现模拟信号到数字信号的数据采集功能；其模拟输出通道由 D/A 转换器、放大器等部件组成，实现由数字信号到模拟信号去驱动受控设备的功能。

1. 传感器

传感器是采集或传感最原始物理量的部件，它能把非电量的模拟信号转换成电量的模拟信号。因此，传感器也称为非电量转换器。常用的传感器有：温度传感器、压力传感器、流量传感器、振动传感器和重量传感器等。根据传感器结构特点或物理效应，传感器可分为：应变

式、电容式、压电式和压阻式等。

2. 低通滤波器

低通滤波器用来降低噪声，滤去不必要的干扰，以增加信噪比。

3. 多路开关

通常，在一个测控系统中，要监测的模拟量往往不止一个，尤其是在数据采集系统中，需要采集的模拟量一般比较多，且它们的变化缓慢。如果每一个模拟量都连接放大器、采样保持电路和 A/D 转换器，这样成本较高，为了节省投资，可以用多路开关，使多个模拟量共用一个放大器、采样保持电路和 A/D 转换器进行分时采样和转换。

4. 采样保持电路

采样保持电路广泛应用于数据采集系统和实时控制系统，具有采样跟踪状态和保持状态两种功能。在 A/D 进行转换期间，保持输入信号不变的电路称采样保持电路。由于输入模拟信号是连续不断变化的，而 A/D 转换器转换总需要一定的时间去完成。对于变化较快的模拟输入信号来说，如果不采取措施，可能造成所转换的数字量误差较大，影响转换精度，从而影响检测系统和控制系数的精度。如果将输入的电压在转换过程中保持一段时间不变，以便 A/D 转换器转换，就可以避免以上问题。采样保持电路的功能就是为了此目的而设计的。

5. 放大器

由传感器转换后得到的电信号比较微弱（通常在毫伏级或微伏级），因此通过放大器将其整形放大到合适的量程，以适合 A/D 转换器的采样。

6. A/D 转换器

A/D 转换器的功能是把模拟信号转换成数字信号。按转换原理可将 A/D 转换器分为：逐次逼近型和积分型；按分辨率可分为：4 位、6 位、8 位、12 位等；按转换速度可分为：超高速、次超高速、高速、中速和低速等。

7. D/A 转换器

D/A 转换器的功能是把数字信号转化成模拟信号。按制造工艺可将 D/A 转换器分为：双极型和 MOS 型等；按输出形式分为：电流型和电压型。

5.5.2 D/A 转换器

1. D/A 转换器工作原理

D/A 转换器通常由基准电源、电阻解码网络、运算放大器和缓冲寄存器等部件构成。基本原理如图 5-74 所示。典型的 D/A 转换器主要有电阻网络 D/A 转换器和 T 形电阻网络 D/A 转换器。

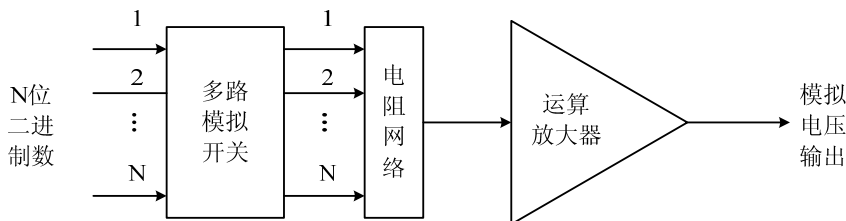


图 5-74 D/A 转换器框图

(1) 权电阻网络 D/A 转换器

权电阻网络 D/A 转换器的原理如图 5-75 所示。

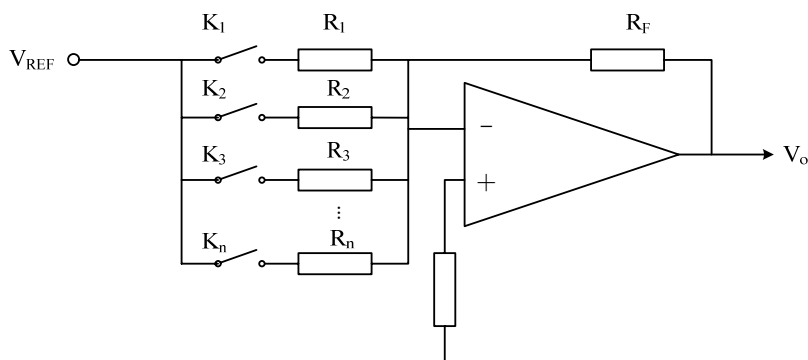


图 5-75 加权电阻网络 D/A 转换器原理图

对于理想运算放大器而言，其中对于其他支路断开时任意一支开关控制电路在输出端对应的电压为： $V_o = -V_{REF} \times R_F \times D_i / R_i$ 。式中， D_i 是数字量对应的控制位， $D_i = 0$ 时， S_i 打开， $D_i = 1$ 时， S_i 闭合。若输入端有 N 个支路，根据叠加原理，输出电压应该是各分支之和。

假定在制造 D/A 转换器时，使 $R_F = R$ ， $R_1 = 2R$ ， $R_2 = 4R$ ， $R_3 = 8R$ ， $\dots$ ， $R_n = 2^n R$ ，即每一个电阻的加权为 2^i ，则输出电压可以表示为： $V_o = -V_{REF} \times (2^{-1}D_1 + 2^{-2}D_2 + 2^{-3}D_3 + \dots + 2^{-n}D_n)$ 。以 8 位转换器为例，当 $D = 0$ 时，开关都断开， $V_o = 0$ ；当 $D = 11111111B$ 时，开关都闭合， $V_o = -(255/256) \times V_{REF}$ 。

从上面分析可以知道，D/A 转换器的转换精度与基准电压 V_{REF} 和权电阻的精度以及数字量的位数 n 有关。显然，位数越多，权电阻的精度越高，转换精度就越高，但同时所需要的权电阻种类就越多，由于在集成电路中制造大量的高精度、高阻值的电阻比较困难，所以，通常使用 T 形电阻网络来代替加权电阻网络。

(2) T 形电阻网络 D/A 转换器

T 形电阻网络 D/A 转换器又称 R-2R 梯形电阻网络，它只用两种阻值的电阻（ R 和 $2R$ ），此种转换器在工艺生产和制造精度方面都容易保证，因此应用广泛。其原理如图 5-76 所示。

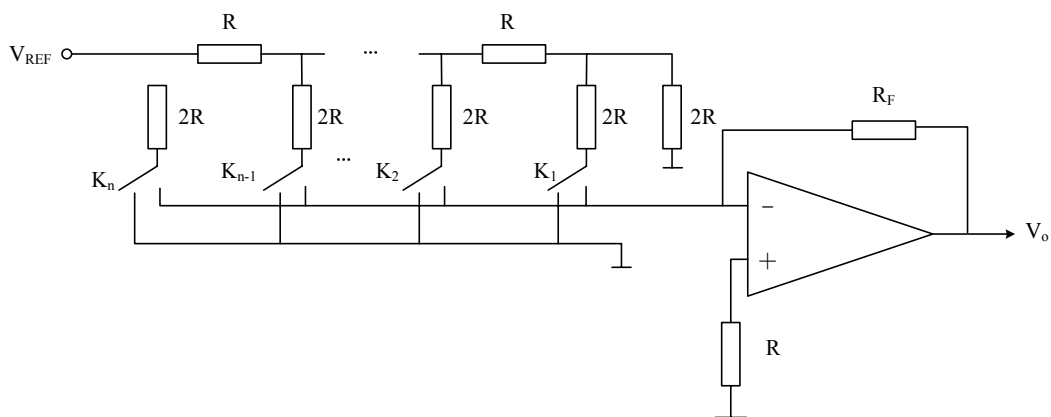


图 5-76 T 形电阻网络 D/A 转换器原理图

图中的 S_i 同样由输入的数字量对应的控制位 D_i 控制, 但是 $D_i = 0$ 时, S_i 接地; 只有 $D_i = 1$ 时, S_i 接通运放输入端。其工作原理与加权电阻网络一样, 输出电压为: $V_o = -V_{REF} \times B/2^n$ 。式中 B 表示待转换的十进制数字量。

2. D/A 转换器的性能指标

(1) 分辨率

分辨率是指 D/A 转换器所能分别出来的最小输出电压 (最小模拟量增量)。这个参数表明 D/A 转换器对模拟量的分辨能力。分辨率越高, 转换时, 对应最小数字量输入的模拟信号电压数值越小, 也就越灵敏。有以下两种表示法。

1) 用数相对值表示, 即用最小模拟量增量与满量程输出值之比。对于 N 位 D/A 转换器, 其分辨率为:

$$\frac{1}{2^N - 1}$$

例如 $N = 8$, 则其分辨率为 $\frac{1}{2^8 - 1} = 0.4\%$ 。

2) 用数字量最低有效位 LSB (Least Significant Bit) 所对应的模拟量表示。对于 N 位 D/A 转换器, 其分辨率为:

$$LSB = \frac{V_{FS}}{2^N}$$

其中, V_{FS} 为满量程模拟量。

例如 $N = 8$ 的 D/A 转换器, 满量程电压为 10V, 则其分辨率为: $LSB = \frac{10}{2^8} = 39.1\text{mV}$ 。

方法 2) 的表示比方法 1) 的表示直观, 但在工程上却常用方法 1) 来表示。

(2) 转换精度

转换精度表明 D/A 转换器转换的精确程度, 分为绝对精度和相对精度。

D/A 的绝对精度 (绝对误差) 指的是在数字输入端加有给定的数字量时, 在输出端实际测得的模拟输出值 (电压或电流) 与相应的理想输出值之差。它是由 D/A 的增益误差、零点误差、线性误差和噪声等综合因素引起的。因此, 在 D/A 的数据图表上往往是以单独给出各种误差的形式来说明绝对误差。

D/A 的相对精度指的是满量程值校准以后, 任何一个数字输入的模拟输出与它的理论输出值之差。对于线性 D/A 来说, 相对精度就是非线性度。

在 D/A 数据图表中, 精度特性一般是以满量程电压 V_{FS} 的百分数或以最低有效位 (LSB) 的分数形式给出, 有时用二进制数的形式给出。

精度 $\pm 0.1\%$ 指的是最大误差为 V_{FS} 的 $\pm 0.1\%$ 。例如, 满度值为 10V 时, 则最大误差为 $V_E = 10\text{V} \times (\pm 0.1\%) = \pm 10\text{mV}$

N 位 D/A 的精度为 $\pm 1/2\text{LSB}$ 指的是最大可能误差为:

$$V_E = \pm \frac{1}{2} \times \frac{1}{2^N} V_{FS} = \pm \frac{1}{2^{N+1}} V_{FS}$$

精度和分辨率是两个截然不同的参数。分辨率取决于转换器的位数, 而精度则取决于构成转换器各部件的精度和稳定性。

(3) 温度灵敏度

温度灵敏度定义为在满刻度输出的条件下, 温度每升高 1°C , 输出变化的百分数。这个参数表明 D/A 转换器受温度变化的影响特性。

(4) 建立时间

建立时间是指从数字输入端发生变化开始, 到输出模拟值稳定在额定值的 $1/2\text{LSB}$ 时所需要的时间。它是 D/A 转换速率快慢的一个重要参数。在实际应用中, 要正确选择 D/A 转换器, 使它的转换时间小于数字输入信号发生变化的周期。

3. DAC0832 的内部结构和外部引脚

集成的 DAC 芯片有多种形式, 根据结构来分, 可以分为两大类: ①DAC 芯片内设置有数据寄存器、片选信号和其他控制信号, 可直接与 CPU 或微机系统总线相连; ②没有数据寄存器, 因此需要通过接口芯片与 CPU 或微机系统总线相连, 有接口芯片进行数据锁存。目前生产的大多数 DAC 芯片内部都带有数据寄存器, 从而可以直接与 CPU 连接, 使用比较方便。D/A 转换器的输出形式有电压和电流两大类, 又可分为电压输出型和电流输出型两类。

DAC0832 是 National Semiconductor Corporation 公司生产的 8 位数/模转换器芯片, 其与微机接口方便, 转换控制容易, 而且价格便宜, 故在实际工程中应用广泛。

DAC0832 具有如下特性: 数字输入端具有双重缓冲功能, 可以双缓冲、单缓冲或直通数字输入; 与所有通用微处理器可直接接口; 分辨率为 8 位, 满量程误差 $\pm 1\text{LSB}$, 建立时间为 $1\mu\text{s}$, 增益温度系数为 $20 \times 10^{-6}/^{\circ}\text{C}$, 功耗 20mW ; 是电流输出型 D/A 转换器。

(1) DAC0832 的内部结构

DAC0832 采用 R-2R 的 T 型电阻解码网络, 有二级缓冲寄存器 (即输入数据可分别经过两个寄存器保存)、D/A 转换电路及 D/A 转换控制电路组成, DAC0832 的第一个寄存器称为 8 位输入寄存器, 数据输入端可直接连接到数据总线上, 第二个寄存器为 8 位 DAC 寄存器。其内部结构逻辑如图 5-77 所示。

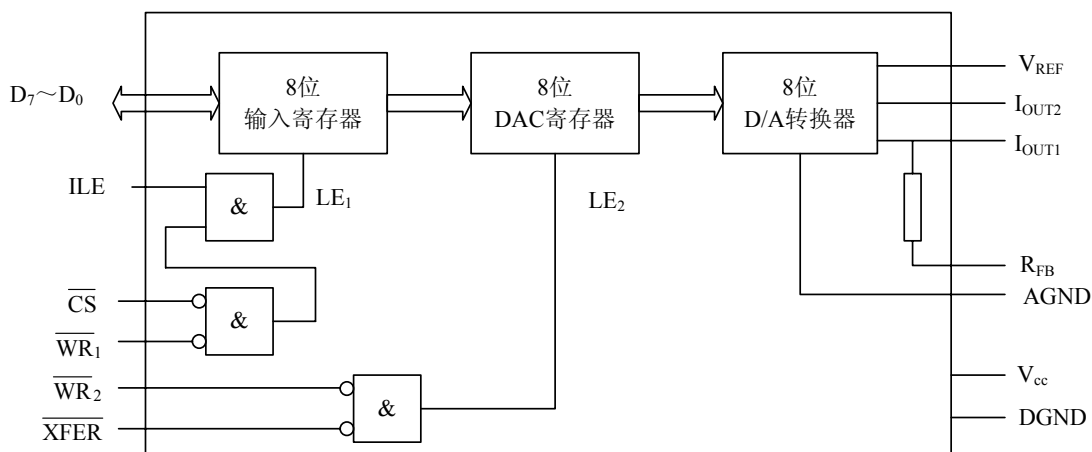


图 5-77 DAC0832 的内部结构

(2) DAC0832 的外部引脚

DAC0832 外部引脚图如图 5-78 所示, DAC0832 是具有 20 个引脚的双列直插封装集成芯片, 各引脚功能如下所述:

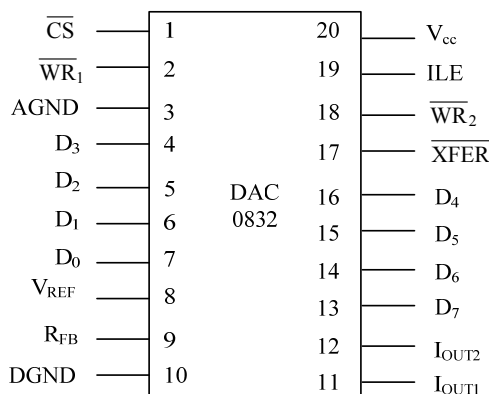


图 5-78 DAC0832 外部引脚

$D_7 \sim D_0$: 数字量输入, D_0 为最低位, D_7 为最高位。

$\overline{CS}$: 数据寄存器片选信号, 低电平有效。

ILE : 数据寄存器的选通信号, 高电平有效。

$\overline{WR}_1$: 数据写入寄存器的控制信号 1, 低电平有效。

输入锁存器的锁存信号 LE_1 由 ILE 、 $\overline{CS}$ 、 $\overline{WR}_1$ 的逻辑组合产生。当 ILE 为高电平, $\overline{CS}$ 和 $\overline{WR}_1$ 同时为低电平时, LE_1 为高电平, 输入寄存器的输出随输入变化; 当 $\overline{WR}_1$ 编程高电平时, LE_1 变为低电平, 输入数据被锁存在输入寄存器中。

$\overline{WR}_2$: DAC 寄存器的写选通信号, 低电平有效。

$XFER$: 传送控制信号, 低电平有效。此信号和 $\overline{WR}_2$ 控制信号是决定 8 位 DAC 寄存器是否工作的控制信号。

V_{cc} : 芯片电源。一般在 +5V ~ +15V 范围内。典型值 +15V。

$AGND$: 模拟地。芯片模拟电路接地点。

$DGND$: 数字地。芯片数字电路接地点。

I_{OUT1} : 模拟电流输出 1, 它是逻辑电平为 1 的各位输出电流之和。当输入数字全为 1 时, 电流最大; 全为 0 时, 电流为 0。

I_{OUT2} : 模拟电流输出 2, 它是逻辑电平为 0 的各位输出电流之和。 $I_{OUT1} + I_{OUT2} = \text{常数}$ 。

R_{FB} : 反馈电阻, 该电阻被制作在芯片内部, 用作 DAC 提供输出电压的运放的反馈电阻。

V_{REF} : 基准电源输入。一般在 -10V ~ +10V 范围内, 通过外电路提供。

DAC0832 的 8 位输入寄存器的输入端 $D_7 \sim D_0$ 可以直接与 CPU 的数据总线相连接。两个数据缓冲寄存器的工作状态分别受 $\overline{LE1}$ 和 $\overline{LE2}$ 控制。当 $\overline{LE1} = 0$ 时, 输入数据寄存器的输出同输入的变化而变化, 当 $\overline{LE1} = 1$ 时, 输入数据被锁存。DAC 寄存器的控制方法与前者相同。

4. DAC0832 的工作方式

DAC0832 内部有两个寄存器, 即输入寄存器和 DAC 寄存器, 能实现 3 种工作方式: 单缓冲方式、双缓冲方式和直通方式。

(1) 单缓冲方式

单缓冲方式是使具有锁存功能的两个寄存器, 其中一个处于直通状态, 而另一个处于受控的锁存状态。一般在使用时, 使 DAC 寄存器处于直通, 即将 $\overline{WR}_2$ 和 $XFER$ 引脚接地, 而使输

入寄存器为受 CPU 控制的锁存，将 $\overline{WR}_1$ 与 CPU 写信号 $\overline{WR}$ 连接。当 CPU 执行写操作时，使 $\overline{WR}_1$ 变为有效，若此时 $\overline{CS}=0$ ， $ILE=1$ ，输入寄存器的状态与输入线 $D_7\sim D_0$ 的状态一致，当 $\overline{WR}_1$ 由低变为高电平时，输入数据被锁存，被锁存的信号经 DAC 寄存器直通输入到 D/A 转换器中转换输出。此方式适用于多个 DAC 芯片转换的模拟信号，不需要同步输出的场合。

(2) 双缓冲方式

DAC0832 工作在双缓冲方式时，必须将 ILE 接高电平， $\overline{WR}_1$ 、 $\overline{WR}_2$ 均接到 CPU 的 $\overline{IOW}$ ，而 $\overline{CS}$ 和 $\overline{XFER}$ 分别接两个端口的地址译码信号。 $\overline{CS}$ 作为输入寄存器的选通信号， $\overline{XFER}$ 作为 DAC 转换寄存器的选通信号。

在双缓冲工作方式下，CPU 对 DAC 芯片进行两次写操作。首先，将 8 位数据写入输入寄存器，然后进行虚拟写操作，将输入寄存器的内容写入到 DAC 转换寄存器，并启动 D/A 转换器开始转换；第二次写操作，是一个虚拟写，目的是为了产生一个 $\overline{XFER}$ 信号。

双缓冲方式具有数据接收和启动转换分步进行的优点。在 D/A 转换的同时，可以进行下一次的数据接收，从而提高多通道模拟输出的转换速率。

(3) 直通方式

直通方式是两个寄存器都处于直通状态，即 ILE 、 $\overline{CS}$ 、 $\overline{WR}_1$ 、 $\overline{WR}_2$ 和 $\overline{XFER}$ 都处于有效电平状态，数据直接送入 D/A 转换器电路进行 D/A 转换，并输出结果。由于该方式下，DAC0832 不能直接和 CPU 的数据总线连接，故很少采用，仅用于一些不采用微机的控制系统中。

5. DAC0832 的应用实例

(1) D/A 转换器与 CPU 的接口

D/A 转换器与 CPU 间的信号连接主要包括数据线、控制线、地址线、地址译码电路和运算放大器。

DAC0832 内部具有数据输入寄存器，CPU 的数据总线能够与 D/A 转换器的数据输入端直接连接，而 CPU 的地址总线则通过译码器与 D/A 转换器的片选端 $\overline{CS}$ 连接。具体接口电路如图 5-79 所示。

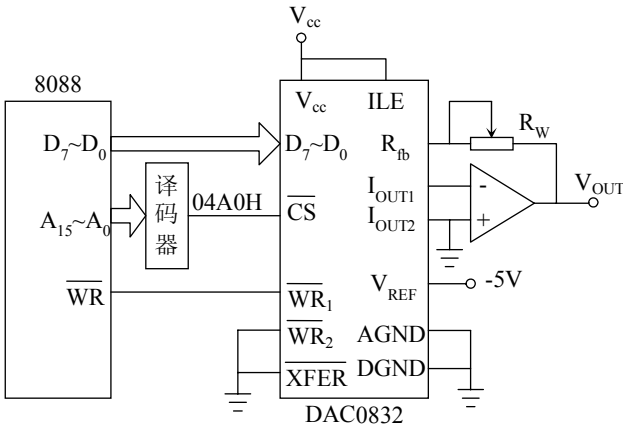


图 5-79 DAC0832 与 CPU 的典型连接

该接口连接工作在单缓冲方式下，转换结果以一组差分电流 I_{OUT1} 和 I_{OUT2} 输出。为了输出

电压量,在电流输出端外接了运算放大器,从而形成单极性电压输出。如果要输出负电压,参考电压必须为正电压;如果要输出正电压,参考电压必须为负电压。为保证输出的线性度,两个电流输出端 I_{OUT1} 和 I_{OUT2} 的电位一般应尽可能接近零电位,否则运算放大器输入端的微小电位差就将导致很大的输出线性误差。

$$V_o = -\frac{V_{REF}}{2^N} \times D$$

式中, N 为 D/A 转换器位数, D 为输入的数字量的数值。

例 5-17 采用单缓冲方式,通过 DAC0832 输出产生三角波,三角波最高电压 5V,最低电压 0V。

硬件设计:

依据题意要求,DAC0832 内部有二级锁存寄存器,从 CPU 送来的数据能被保存,不用外加锁存器,可直接与 CPU 数据总线相连。采用单缓冲方式,假设我们采用第一级锁存,第二级直通,那么第二级的控制端 $\overline{WR}_2$ 和 $\overline{XFER}$ 应处于有效电平状态,使第二级锁存寄存器一直处于打开状态。第一级寄存器具有锁存功能的条件是 $\overline{ILE}$ 、 $\overline{CS}$ 、 $\overline{WR}_1$ 都要满足有效电平。为减少控制线条数,可使 $\overline{ILE}$ 一直处于高电平状态,控制 $\overline{CS}$ 、 $\overline{WR}_1$ 端。电压输出范围为 0V~5V,故可以按照 DAC0832 与 CPU 典型连接设计电路,如图 5-79 所示。

软件设计:

如图 5-79 所示, $\overline{CS}$ 端与译码电路的输出端相连,其地址数既是选中该 DAC0832 芯片的片选信号,也是第一级寄存器打开的控制信号。由于 CPU 的控制信号 $\overline{WR}$ 与 DAC0832 的写信号 $\overline{WR}_1$ 相连,当执行 OUT 指令时,CPU 的 $\overline{WR}_1$ 写信号有效,与 $\overline{CS}$ 信号一起,打开第一级寄存器,输入数据被锁存。

假设 DAC0832 地址为 04A0H,输出 0V 电压程序如下:

```
MOV    AL, 00H           ;设置输出电压值
MOV    DX, 04A0H         ;DAC0832 片选地址
OUT    DX, AL            ;输出数据,使 DAC0832 输出端得到 0V 模拟电压输出
```

依据题意要求,产生三角波电压范围为 0V~5V,那么所对应输出数据 00H~FFH。所以三角波上升部分,从 00H 起加 1,直到 FFH。三角波下降部分从 FFH 起减 1,直到 00H,流程图如图 5-80 所示。

```
MOV    AL, 00H           ;设置输出电压值
MOV    DX, 04A0H         ;DAC0832 芯片地址送 DX
AA1:   OUT    DX, AL
        INC    AL           ;修改输出数据
        CMP    AL, 0FFH
        JNZ    AA1
AA2:   OUT    DX, AL
        DEC    AL           ;修改输出数据
        CMP    AL, 00H
        JNZ    AA2
        JMP    AA1
```

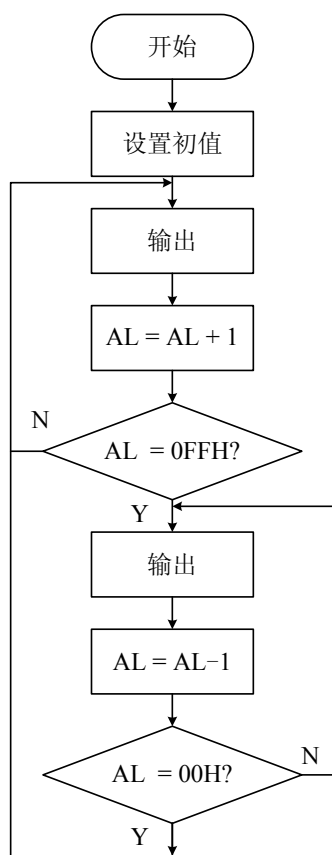


图 5-80 流程图

上述程序段产生的正向锯齿波如图 5-81 所示。从 0 增长到最大输出电压，中间要分成 256 个小台阶，分别对应 0, 1LSB, 2LSB, 3LSB, ..., 255LSB 时的模拟输出电压。

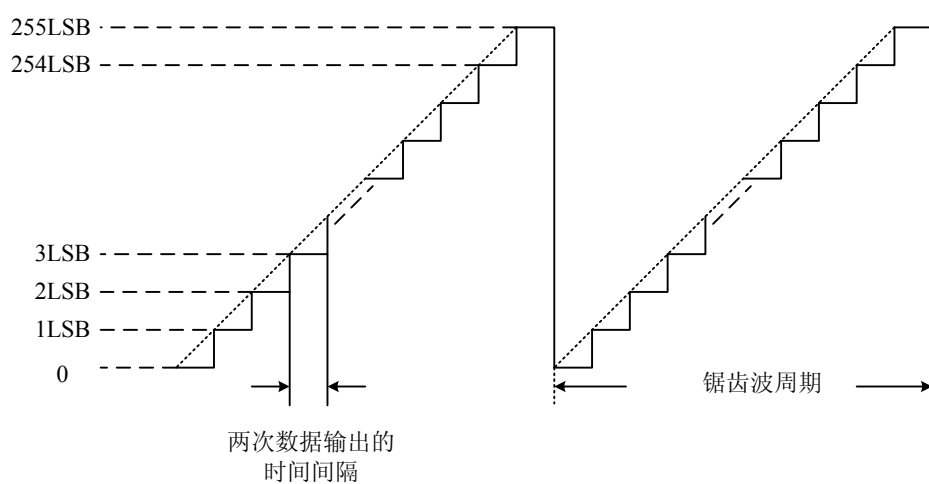


图 5-81 DAC0832 编程产生的正向锯齿波

(2) 利用 D/A 转换器产生各种波形信号

利用 D/A 转换器输出模拟量与输入数字量成正比关系的特点，将 D/A 转换器作为微机输出接口，CPU 通过程序向 D/A 转换器输出随时间呈现不同变化规律的数字量，则 D/A 转换器就可以输出各种各样的模拟量。利用 D/A 转换器可以产生各种波形信号，如方波、三角波、锯齿波等，以及这些波形组合产生的复合波形。

下面以图 5-82 为例，使用 DAC0832 产生各种波形信号。利用 8255A 作为 CPU 与 DAC0832 之间的接口。8255A 的 A 口为数据输出口，DAC0832 的 $\overline{CS}$ 、 $\overline{WR}_1$ 、 $\overline{WR}_2$ 、 $\overline{XFER}$ 接地，ILE 接高电平，工作于直通方式。设 8255A 的端口地址为 0FFF8H、0FFFAH、0FFFC H 和 0FFFEH。

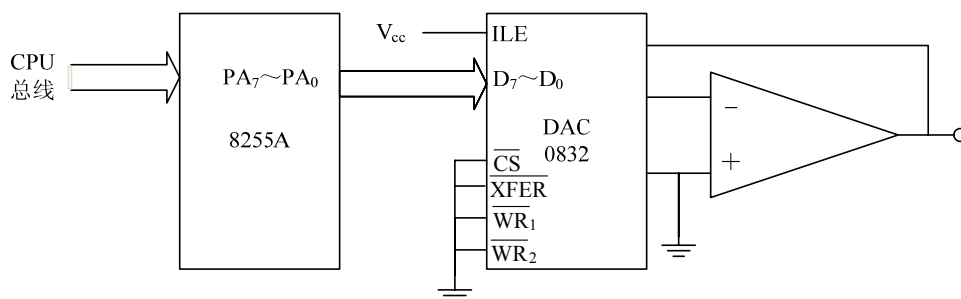


图 5-82 DAC0832 波形产生电路

1) 方波。

```

MOV  DX, 0FFFEH           ;8255A 控制端口地址
MOV  AL, 82H
OUT  DX, AL               ;设置 8255A 的 A 口工作于方式 0，输出
MOV  DX, 0FFF8H           ;8255A 的 A 口地址
LOOP: MOV AL, 00H
      OUT DX, AL           ;输出方波“0”
      CALL DELAY           ;方波宽度
      MOV AL, 0FFH
      OUT DX, AL           ;输出方波“1”
      CALL DELAY
      JMP LOOP

```

```

DELAY  PROC NEAR
        PUSH  BX
        PUSH  CX
        MOV  BX, 2000
DEL1:   MOV  CX, 0
DEL2:   LOOP DEL2
        DEC  BX

```

```

        JNZ DEL1
        POP CX
        POP BX
        RET
DELAY ENDP
2) 锯齿波。
        MOV DX, 0FFFEH
        MOV AL, 82H
        OUT DX, AL
        MOV DX, 0FFF8H
        MOV AL, 00          ;输出数据初值
LOOP:   OUT DX, AL          ;锯齿波输出
        INC AL
        JMP LOOP

```

3) 三角波。利用正、负向锯齿波组合，产生三角波。

```

        MOV DA, 0FFFEH
        MOV AL, 82H
        OUT DX, AL
LOOP1:  MOV DX, 0FFF8H
        MOV AL, 00H          ;正向初值
LOOP2:  OUT DX, AL
        INC AL
        JNZ LOOP2
        MOV AL, 0FFH         ;负向初值
LOOP3:  OUT DX, AL
        DEC AL
        JNZ LOOP3
        JMP LOOP1

```

5.5.3 A/D 转换器

1. A/D 转换器工作原理

实现 A/D 转换的方法很多，按照转换原理的不同可将 A/D 电路分成直接转换和间接转换两大类，每类中又包含许多结构不同、性能各异的电路。直接 A/D 就是把模拟量直接转换为数字量，有并行 A/D 转换器和逐次逼近 A/D 转换器。间接 A/D 就是把模拟量转换为其他形式的量，然后通过简单电路，如定时器/计数器，再转换为数字量，有电压—时间变换 A/D 转换器、双积分 A/D 转换器和电压—频率变换 A/D 转换器。

逐次逼近 A/D 转换器经常用在微机应用系统中。其原理如图 5-83 所示。

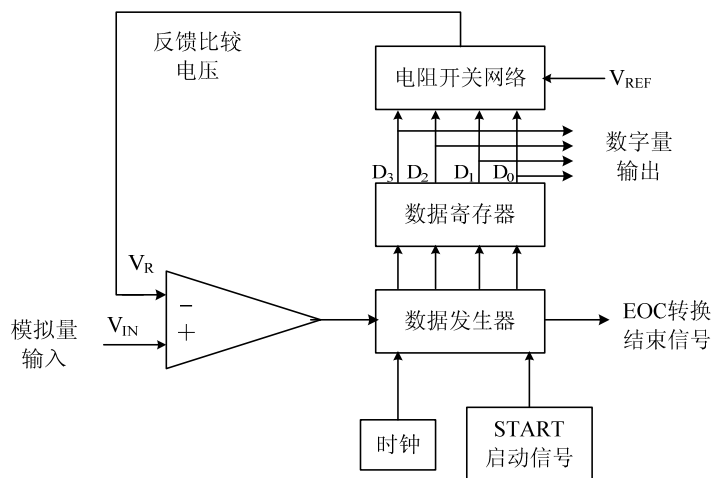


图 5-83 逐次逼近 A/D 转换器原理

逐次逼近 ADC 采用寄存器控制 DAC。转换前，寄存器各位清除为 0。转换时，寄存器先由最高位置 1，DAC 输出值与被测的模拟值进行比较：如果“低于”，该位的 1 被保留；如果“高于”，该位的 1 被清除。然后下一位再置 1，再比较，均定清除与保留，直到最低位完成同一过程。寄存器从最高位到最低位试探完的最终值就是 A/D 转换的结果。

逐次逼近 ADC 属于反馈比较型 A/D 转换器，采用从最高位开始逐位试探法。对 N 位 ADC，逐次逼近 ADC 只要 N 次比较就可完成转换。逐次逼近 ADC 是中速、8~16 位 ADC 的主流产品。

有关 A/D 转换原理的详细内容，已超过本书所讨论的范围，感兴趣的读者可以参阅其他相关书籍。

2. A/D 转换器的性能指标

(1) 分辨率

分辨率用来反映 A/D 转换器对输入量微小变化响应的分辨能力，一般用数字输出最低位 (LSB) 对应的模拟输入电压值表示。由于分辨率直接与转换器的位数有关，所以一般简单地用数字量的位数来表示分辨率，也就是 N 位二进制数最低位所具有的权值。例如，对 8 位 A/D 转换器，其数字量输出的变化范围为 0~255，当输入电压满量程为 5V 时，转换电路对输入模拟电压的分辨能力为 $5V/255 = 19.6mV$ 。目前常用的 A/D 转换集成芯片的转换位数有 8 位、10 位、12 位和 14 位等。

(2) 转换精度

转换精度是指与数字输出量所对应的模拟输入量的实际值与理论值之间的差值。A/D 转换电路中与每一个数字量对应的模拟输入量并非是单一的数值，而是一个范围 Δ 。例如，对满量程输入电压为 5V 的 12 位 A/D 转换器， $\Delta = 5V/FFFH = 1.22mV$ ，定义为数字量的最小有效位 LSB。若理论上输入的模拟量 A，产生数字量 D，而输入模拟量 $A \pm \Delta/2$ 产生还是数字量 D，则称此转换器的精度为 $\pm 0.5 \text{ LSB}$ 。当模拟电压 $A + \Delta/2 + \Delta/4$ 或 $A - \Delta/2 - \Delta/4$ 还是产生同一数字量 D，则称其精度为 $\pm 1.5 \text{ LSB}$ 。目前常用的 A/D 转换器的转换精度为 $1/4 \sim 2 \text{ LSB}$ 。

(3) 转换时间

完成一次 A/D 转换所需的时间。不同型号、不同分辨率的器件，其转换时间的长短相差很大，从几微秒到几百毫秒不等。在选择器件时要根据应用的需要和成本来具体地考虑。

(4) 误差

A/D 转换器的误差包括：量化误差、线性误差、偏置误差等。

量化误差：在 A/D 转换过程中因量化而产生的误差，通常为 $\pm 1/2$ LSB。

线性误差：由于量化的作用，A/D 转换的输出呈阶梯状。线性误差指的是输入的理论输出与实际输出的两个不同水平阶梯中点之间的电压差。

偏置误差：A/D 转换器的最低输出 LSB 为 1 时，模拟输入电压的实际值与理论值之差。

(5) 量程

指 A/D 转换器能够转换的模拟输入电压的变化范围，分单极性、双极性两种类型。例如，单极性常见量程为 0~5V，0~10V，0~20V；双极性量程通常为-5V~+5V，-10V~+10V 等。

3. ADC0809 的内部结构和外部引脚

ADC 芯片的性能主要用分辨率、转换时间和转换精度等反映，还有转换技术的区别。有些 ADC 芯片不仅具有 A/D 转换的基本功能，还包含内部放大器和三态输出锁存器；有的甚至还包括了开关及采样保持器等。

ADC0809 是逐次逼近型 8 位 A/D 转换芯片。片内有 8 路模拟开关，可以同时连接 8 路模拟量，单极性，量程为 0~5V。典型的转换速度为 100 μ s。片内有三态输出缓冲器，可直接与 CPU 总线连接。该芯片有较高的性价比，适用于对精度和采样速度要求不高的场合或一般的工业控制领域。

(1) ADC0809 的内部结构

由图 5-84 所示 ADC0809 内部结构原理框图可知，它由 8 路模拟开关、模拟开关的地址锁存与译码电路、比较器、T 型网络、树状电子开关、逐次逼近寄存器、三态输出锁存缓冲存储器、控制与时序电路等组成。内部结构分为四部分。

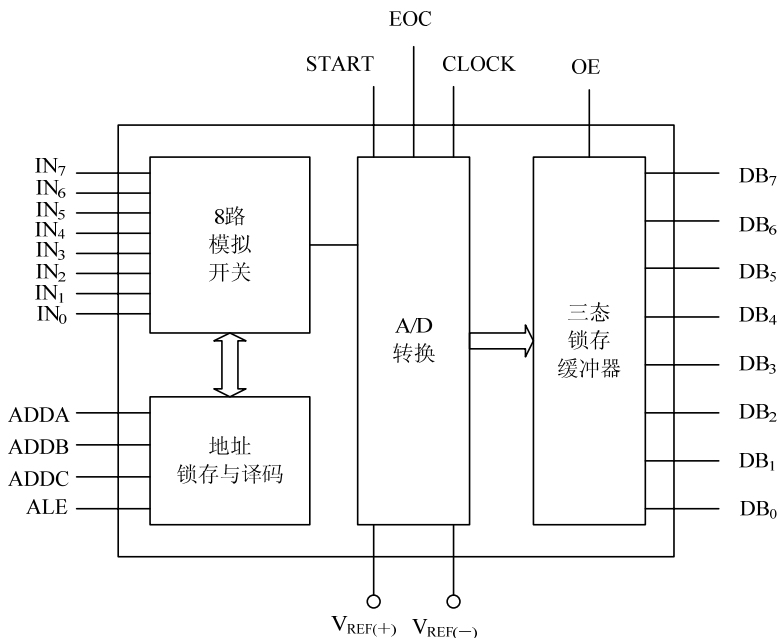


图 5-84 ADC0809 内部结构框图

1) 模拟输入部分。ADC0809 的模拟输入部分提供一个 8 通道的多路开关和寻址逻辑, 可以接入 8 个模拟输入电压。其中 $IN_0 \sim IN_7$ 是 8 个模拟电压输入端, ADDA、ADDB 和 ADDC 是 3 个地址输入线, 而 ALE 地址锁存允许信号的上升沿用于锁存 3 个地址输入的状态, 然后由译码器选中一个模拟输入端进行 A/D 转换, 具体模拟输入通道选择如表 5-13 所示。

表 5-13 ADC0809 模拟通道选择

模拟通道	ADDC	ADDB	ADDA	模拟通道	ADDC	ADDB	ADDA
IN_0	0	0	0	IN_4	1	0	0
IN_1	0	0	1	IN_5	1	0	1
IN_2	0	1	0	IN_6	1	1	0
IN_3	0	1	1	IN_7	1	1	1

通道的选择可以在转换前独立地进行。而实际通常把通道选择和启动转换结合起来完成, 这样可以将一条输出指令既用于选择模拟通道又用于启动转换。

2) A/D 转换器部分。主要由电阻网络、比较器逐次逼近寄存器 SAR 和控制逻辑组成。

3) ADC0809 时序。ADC0809 的转换过程由时钟脉冲 CLK 控制, 频率范围为 $10\text{kHz} \sim 1280\text{kHz}$, 典型值为 640kHz 。ADC0809 的工作时序如图 5-85 所示。

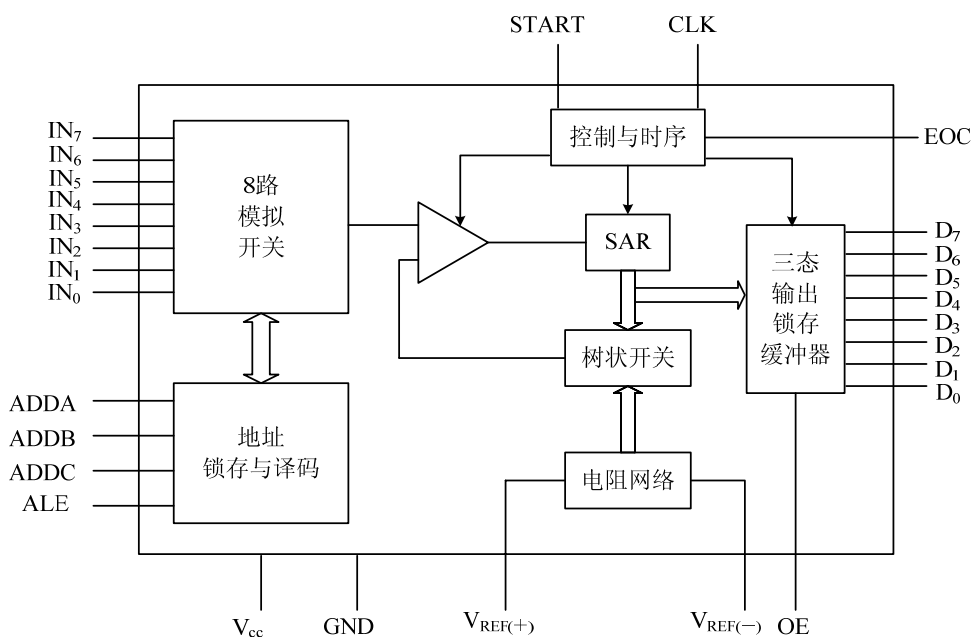


图 5-85 ADC0809 工作时序

ADC0809 工作过程（即完成一次模数转换）需要经过：①在 ALE 信号作用下，地址引脚 ADDA $\sim$ ADDC 上的信号被锁存，选择模拟信号输入通道；②转换过程由 START 信号启动，START 信号的上升沿将内部逐次逼近寄存器复位，下降沿启动 A/D 转换；③转换完成后，ADC0809 转换结束信号 EOC 由低电平变为高电平，可利用该信号进行查询或进行中断请求，

通知 CPU，完成一次模数转换；④执行 ADC0809 数据端口的指令，该指令是使 OE 有效的信号，打开输出三态缓冲器，转换结果通过系统数据总线传送 CPU。

4)基准电压输入。转换使用的基准电压，决定了输入模拟电压的最大值和最小值，由 $V_{REF(+)}$ 和 $V_{REF(-)}$ 提供，接基准电压的正、负极。 $V_{REF(-)}$ 接地时作为 ADC 的模拟地。 V_{cc} 是电源电压，+5V，GND 是数字地。 V_{REF} 基准电压根据 V_{cc} 确定，典型值为 $V_{REF(+)} = V_{cc}$ 、 $V_{REF(-)} = 0$ ，不允许比 V_{cc} 大，不允许比地电平小。

(2) ADC0809 的外部引脚

ADC0809 外部引脚图如图 5-86 所示，ADC0809 是具有 28 个引脚的双列直插封装集成芯片，各引脚功能如下所述：

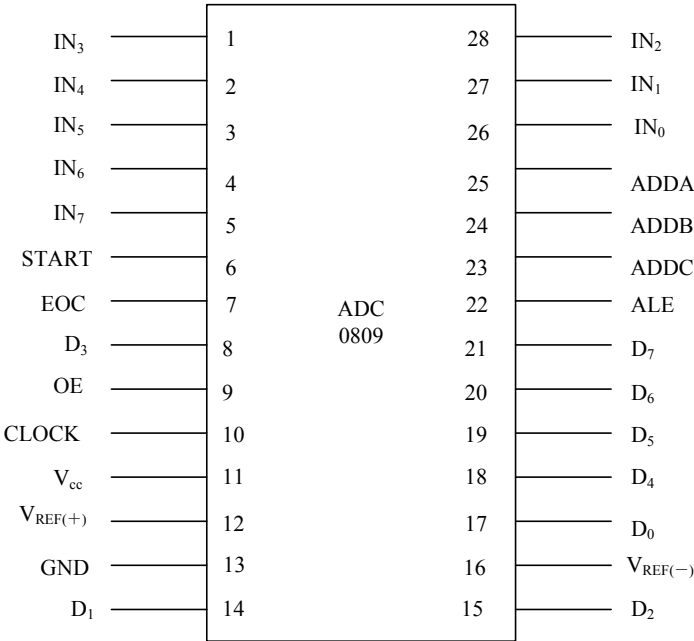


图 5-86 ADC0809 外部引脚图

- $IN_7 \sim IN_0$: 8 路模拟量输入通道。
- $D_7 \sim D_0$: $2^{-1} \sim 2^{-8}$, 8 位数字量输出端, 2^{-8} 为最低有效位, 2^{-1} 为最高有效位。
- ADDA、ADDB、ADDC: 3 位地址输入线, 用于选通 8 路模拟输入中的一路。
- ALE: 地址锁存允许信号, 输入, 由低电平到高电平正跳变时将地址选择线的状态锁存, 以选通相应的输入通道。
- START: A/D 转换启动信号输入端, 输入一个正脉冲 (宽度至少为 100ns) 使其启动 (脉冲上升沿使 ADC0809 复位, 下降沿启动 A/D 转换)。
- EOC: A/D 转换结束信号, 输出, 当 A/D 转换结束时, 此端输出一个高电平 (转换期间一直为低电平)。
- OE: 数据输出允许信号, 输入, 高电平有效。当 A/D 转换结束时, 此端输入一个高电平, 才能打开输出三态门, 输出数字量, 即有效时将输出寄存器中的数据放到数据总线上。
- CLK: 时钟脉冲输入端。要求时钟频率范围为 10kHz~1280kHz, 典型值为 640kHz。

REF(+)、REF(-): 基准电压的正极和负极。

V_{cc} : 芯片电源, 接+5V。

GND: 地线。

4. ADC0809 的应用实例

由于 ADC 芯片输出的数据都要连接到系统的数据总线上。为防止数据总线的数据冲突和保持稳定的数据, 多数向系统数据总线发送数据的设备都设置有三态锁存缓冲器。根据 ADC 芯片的数据输出端是否带有三态锁存缓冲器, 与主机的连接方式分为: ①直接连接: 主要用于输出带有三态锁存缓冲器的 ADC 芯片, 例如 ADC0809; ②通过三态锁存缓冲器或通用并行接口和主机连接, 例如通过 8255A 连接, 适用于不带三态锁存缓冲器的 ADC 芯片。在有些情况下, 带有三态锁存缓冲器 ADC 芯片也常通过间接的方式与主机相连。

例 5-18 ADC0809 与系统直接连接。

如图 5-87 所示, 占用 3 个 I/O 端口: 端口 1 用来向 ADC0809 输出模拟通道号并锁存; 端口 2 用于启动转换; 端口 3 读取转换后的数据结果。采用软件延时。

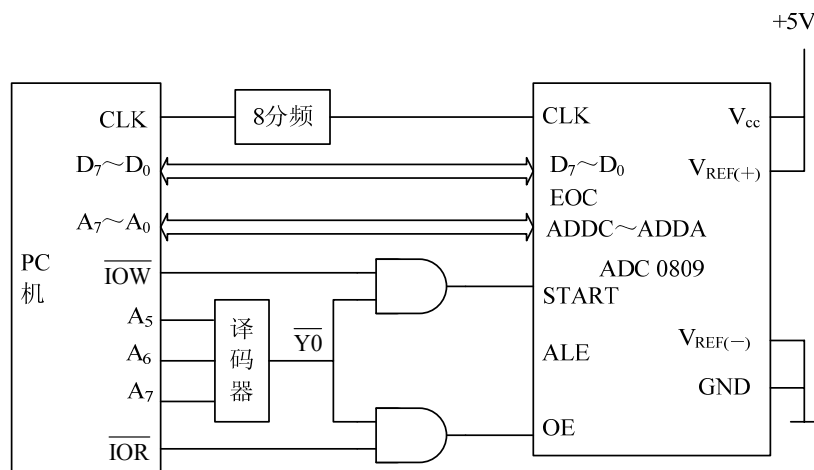


图 5-87 ADC0809 与系统直接连接电路

转换程序段:

```
MOV AL, 07H
OUT 1FH, AL
CALL DELAY100
IN AL, 1FH
HLT
```

例 5-19 ADC0809 通过并行接口 8255A 与系统连接。

如图 5-88 所示, 为 ADC0809 芯片通过 8255A 与 8088 系统的接口电路。

ADC0809 的输出数据通过 8255A 的 A 口传输给 CPU, 地址输入信号 ADDA、ADDB、ADDC 以及地址锁存信号 ALE 由 8255A 的 B 口的低 4 位提供。A/D 转换的状态信号 EOC 通过 8255A 的 C 口的 PC₄ 输入。CLK 外加时钟。

以查询方式读取 A/D 转换后的结果，8255A 的 A 口工作在方式 0 输入，B 口工作在方式 0 输出，PC₄ 为输入。

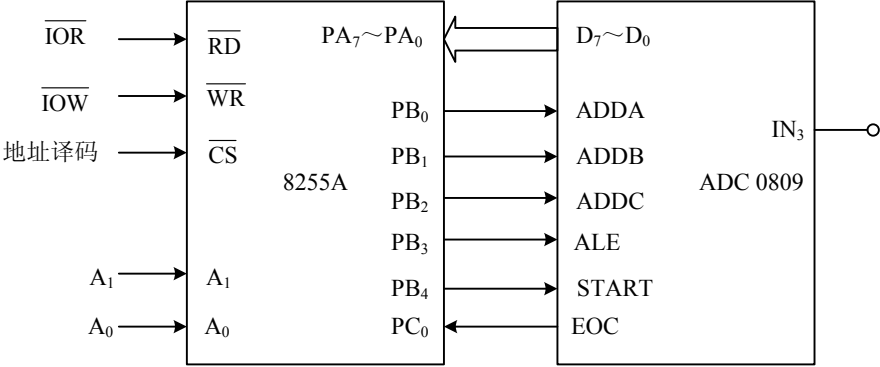


图 5-88 ADC0809 通过并行接口 8255A 与系统连接

源程序:

START: MOV AL, 98H	;8255A 的工作方式控制字, A 口工作在方式 0 输入
	;B 口工作在方式 0 输出
MOV DX, 03FFFH	;8255A 控制寄存器端口地址
OUT DX, AL	;写入 8255A 工作方式控制字
MOV AL, 0BH	;选 IN ₃ 输入端和地址输入信号
MOV DL, 0FDH	;8255A 的 B 口端口地址
OUT DX, AL	;送 IN ₃ 通道地址, 同时使 ALE = 1
MOV AL, 1BH	;PB ₄ = 1, 即 START = 1
OUT DX, AL	;启动 A/D 转换器
MOV AL, 03H	;PB ₄ = 0, 撤销 START 信号
OUT DX, AL	
MOV DL, 0FEH	;8255A 的 C 口端口地址
TST: IN AL, DX	;读取 C 口状态
AND AL, 10H	;查询 PC ₄ 是否为 1, 即查询 ALE 状态
JZ TST	;如果转换没有结束, 循环测试, 直到转换完
MOV DL, 0FCH	;8255A 的 A 口端口地址
IN AL, DX	;从 8255A 的 A 口输入转换后的数字信号

例 5-20 设计 ADC0809 与 PC 机总线的接口电路, 编写程序实现一段轮流从 IN₀~IN₇ 采集 8 路模拟信号, 并把采集到的数字量存入 0100H 开始的 8 个单元内, 设端口地址为 300H 开始。

本例采用三种方式实现, 即查询工作方式、无条件工作方式和中断工作方式。

(1) 查询工作方式

该方式下的硬件电路连接如图 5-89 所示。

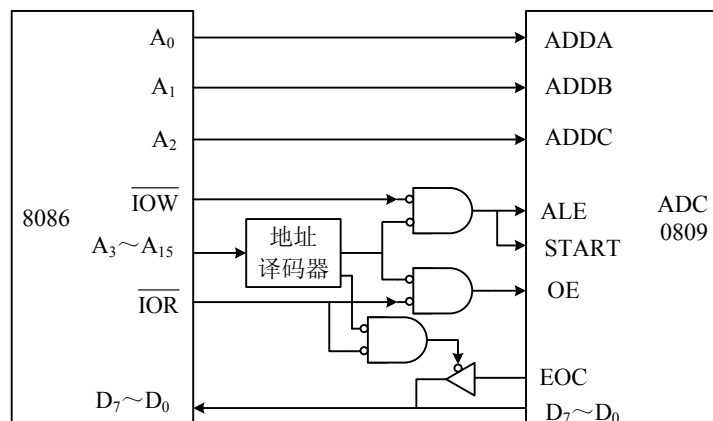


图 5-89 ADC0809 查询方式工作

程序段如下:

```

MOV  DI, 0100H           ;设置存放转换后数字信号的首地址
MOV  CX, 8               ;模拟输入通道计数器
MOV  DX, 300H           ;第一个模拟通道的端口地址
START1: OUT  DX, AL      ;启动 A/D 转换器
        PUSH  DX         ;将模拟通道端口地址暂存
        MOV  DX, 308H    ;状态端口地址
WAIT1:  IN   AL, DX       ;读 EOC 状态信号
        TEST AL, 80H     ;判断转换是否开始
        JNZ  WAIT1       ;非 0, 表示没有开始, 等待
WAIT2:  IN   AL, DX       ;再读 EOC
        TEST AL, 80H     ;判断转换是否结束
        JZ   WAIT2       ;是 0, 表示转换没有结束
        POP  DX          ;是 1, 转换结束, 恢复暂存通道端口地址
        IN   AL, DX      ;读取转换后的数据
        MOV  [DI], AL    ;转换后的数据转存
        INC  DX          ;指向下一个模拟通道
        INC  BX          ;数据缓冲单元地址加 1
        LOOP START1      ;循环下一个通道, 直到全部通道完

```

(2) 无条件工作方式

ADC0809 的 EOC 信号不接。A/D 转换过程中调用一段延时程序, 只要延时时间比 A/D 转换的时间略长, 就可以保证 A/D 转换的数据精度。

程序段如下:

```

MOV  DI, 0100H
MOV  CX, 8
MOV  DX, 300H
START1: OUT  DX, AL

```

```

        PUSH  DX
        MOV   DX, 308
WAIT1:  LOOP  WAIT           ;延时，等待 A/D 转换结束
WAIT2:  IN    AL, DX         ;读 EOC
        TEST  AL, 80H       ;查询是否转换结束
        JZ    WAIT2         ;是 0，表示转换没有结束
        POP   DX            ;是 1，转换结束，恢复暂存通道端口地址
        IN    AL, DX        ;读取转换后的数据
        MOV   [DI], AL
        INC   DX
        INC   BX
        LOOP  START1

```

(3) 中断工作方式

ADC0809 中断工作方式采用图 5-90 所示电路。

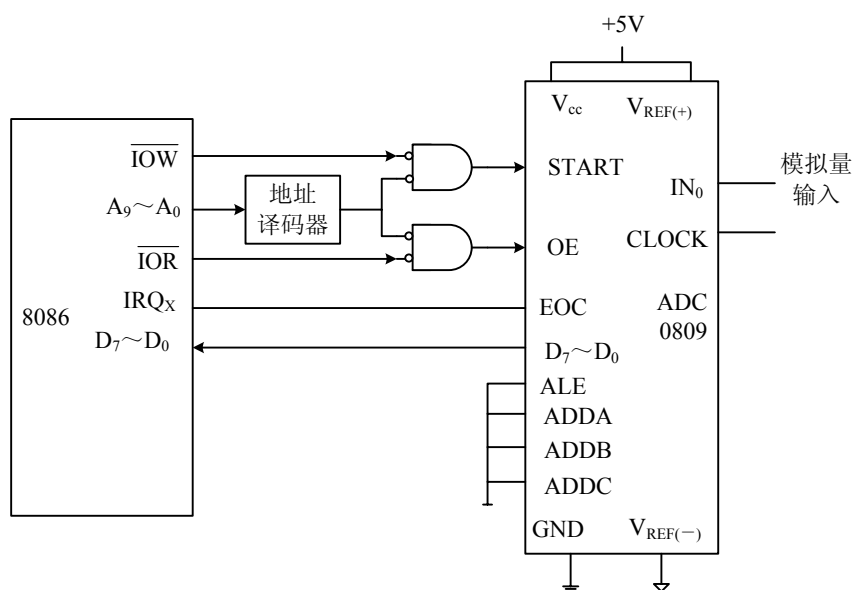


图 5-90 ADC0809 中断方式工作

主程序段：

```

...                               ;设置中断向量，8259A 初始化
MOV  DI, 0100H
MOV  CX, 8
STI                                     ;开中断
MOV  DX, 300H
OUT  DX, AL
...                               ;执行其他程序，同时等待中断

```

中断服务程序段:

```

    PUSH  AX                ;保护现场
    PUSH  DX
    STI                    ;开中断, 允许中断嵌套
    MOV   DX, 300H
    IN    AL, DX            ;读取转换的数据
    MOV   [DI], AL         ;转换后数据转存
    INC   DX
    INC   BX
    OUT   DX, AL            ;启动下一个模拟通道
    LOOP  RETURN           ;全部通道未完, 循环下一个通道
    CLI                    ;关中断
    MOV   AL, 20H          ;送 EOI 中断结束命令
    OUT   20H, AL          ;8259A 端口写
    POP   DX               ;恢复现场
    POP   AX
    RETURN:IRET            ;中断返回

```



习题五

1. 什么叫中断? 简述一个中断的全过程。
2. 什么是硬件中断和软件中断? 在 PC 机中两者的处理过程有什么不同?
3. 8259A 中断控制器的功能是什么?
4. 已知 8086 系统中采用单片 8259A 来控制中断, 中断类型码为 20H, 中断源请求线与 8259A 的 IR_4 相连, 计算中断向量表的入口地址。如果中断服务程序入口地址为 2A310H, 则对应该中断源的中断向量表的内容是什么?
5. 已知对应于中断类型码为 18H 的中断服务程序存放在 0020H:6314H 开始的内存区域中, 求对应于 18H 类型码的中断向量存放位置和内容。在编写程序时, 为什么通常总要用 STI 和 CLI 中断指令来设置中断允许标志? 8259A 的中断屏蔽寄存器 IMR 和中断允许标志 IF 有什么区别?
6. 8255A 的方式选择控制字和按位置位/复位控制字都是写入控制端口的, 那么, 它们是由什么来区分的?
7. 8259A 的初始化命令字和操作命令字有什么区别? 它们分别对应于编程结构中哪些内部寄存器?
8. 8259A 对中断优先权的管理和对中断结束的管理有几种处理的方式? 各自应用在什么场合?
9. 试按照如下要求对 8259A 设置初始化命令字: 系统中有 1 片 8259A, 中断请求信号用电平触发方式, 下面要用 ICW4, 中断类型码为 60H、61H……67H, 用特殊全嵌套方式, 不用缓冲方式, 采用中断自动结束方式。8259A 的端口地址为 90H、92H。

10. 已知 8086 系统采用单片 8259A, 中断请求信号使用电平触发方式, 完全嵌套中断优先级, 数据总线无缓冲, 采用自动中断结束方式, 中断类型码为 20H~27H, 8259A 的端口地址为 B0H 和 B1H, 试编程对 8259A 设定初始化命令字。

11. 设 8253 三个计数器的端口地址为 201H、202H、203H, 控制寄存器端口地址 200H。试编写程序片段, 读出计数器 2 的内容, 并把读出的数据装入寄存器 AX。

12. 设 8253 计数器的时钟输入频率为 1.91MHz, 为产生 25KHz 的方波输出信号, 应向计数器装入的计数初值为多少?

13. 将 8253 定时器 0 设置为方式 3 (方波发生器), 定时器 1 设置为方式 2 (分频器)。要求定时器 0 的输出脉冲作为定时器 1 的时钟输入, CLK₀ 连接总线时钟 4.77MHz, 定时器 1 输出 OUT1 约为 40Hz, 试编写实现上述功能要求的程序。

14. 编程将 8253 计数器 0 设置为模式 1, 计数初值为 3000H; 计数器 1 设置为模式 2, 计数初值为 2010H; 计数器 2 设置为模式 4, 计数初值为 4030H; 地址设为 0070H、0072H、0074H、0076H。

15. 8255A 接口芯片地址为 60H~63H, 请指出下列程序段功能

(1) MOV AL, 80H

OUT 63H, AL

(2) MOV AL, 08H

OUT 63H, AL

16. 8255A 的三个端口在使用时有什么差别?

17. 对 8255A 设置工作方式, 8255A 的控制口地址为 00C6H。要求端口 A 工作在方式 1, 输入; 端口 B 工作在方式 0, 输出; 端口 C 的高 4 位配合端口 A 工作; 低 4 位为输入。

18. 试按以下要求对 8255A 进行初始化编程:

(1) 设端口 A、端口 B 和端口 C 均为基本输入/输出方式, 且不允许中断。请分别考虑输入/输出。

(2) 设端口 A 为选通输出方式, 端口 B 为基本输入方式, 端口 C 剩余位为输出方式, 允许端口 A 中断。

(3) 设端口 A 为双向方式, 端口 B 为选通输出方式, 且不允许中断。

19. 用 8255A 的 A 口作为数据口, B 口作为状态口, 采用查询方式从某外设输入 10 个数据, 存入 0100H 开始的单元, 试进行软、硬件设计。

20. 采用 8255A 作为两台计算机并行通信的接口电路, 请画出查询式输入/输出方式工作的接口电路, 并写出查询式输入/输出方式的程序。

21. 什么叫同步通信方式? 什么叫异步通信方式? 它们各有什么优缺点?

22. 什么叫异步工作方式? 画出异步工作方式时 8251A 的 TxD 和 RxD 线上的数据格式。

23. 什么是波特率? 若要产生一个波特率为 2400 的串行信号, 且波特率因子编程为 16, 那么串口发送/接收时钟的频率是多少?

24. 设计一个采用异步通信方式输出字符的程序段, 规定波特率因子为 64, 7 个数据位, 1 个停止位, 用偶校验, 端口地址为 40H、42H, 缓冲区首址为 2000H:3000H。

25. A/D 与 D/A 转换器在微机应用系统中有什么作用?

26. 模拟量输入输出通道主要由哪几个部件组成? 各部件的主要功能有哪些?

27. D/A 转换的基本工作原理是什么？描述 D/A 转换器的性能指标有哪些主要参数？
28. D/A 转换器的分辨率是如何定义的？它与精度有什么关系？
29. DAC0832 转换器有哪些特点？其内部结构由哪几部分组成？
30. 试设计一个利用 DAC0832 产生锯齿波和正弦波的电路，并编写相应的程序。
31. 举例说明高于 8 位的 D/A 转换器如何与微机进行接口连接。
32. 试画出 ADC0809 直接与 CPU 扩展槽的连接图，并编写采样程序。
33. 试设计一个 8088CPU 与两片 DAC0832 的接口电路，并编写程序使其所输出的信号 (x,y) 在示波器上显示一个正六边形的六个顶点。