

第 3 章 中断系统与可编程 8259A

学习目的

- (1) 了解 8086 中断的概念。
- (2) 了解 8086 的中断类型。
- (3) 了解 8086 的中断矢量表。
- (4) 了解 8259A 芯片引脚和内部结构。
- (5) 掌握 8259A 控制字及编程应用。
- (6) 熟悉 8259A 的编程应用。
- (7) 掌握相关项目编程方法和设计思想。

学习重点和难点

- (1) 了解 8086 的中断矢量表。
- (2) 8259A 控制字编程应用。
- (3) 相关项目编程方法。

引入项目：用外部中断控制继电器

1. 要求与目的

(1) 要求：用 8086 CPU 控制 8259 可编程中断控制器，实现对外部中断的响应和处理。要求程序对 IR_0 每产生中断一次，就去控制继电器动作，使 LED 闪烁。

(2) 目的：

- 了解 8086 中断的概念。
- 了解 8086 的中断类型。
- 了解 8086 的中断矢量表。
- 了解 8086 的中断过程。

2. 电路连接与说明

(1) 电路连接：如图 3-1 所示的粗线为要接的线。接线描述如下：8259A 的片选 $\overline{CS}$ 连至地址译码处的 210H~217H 插孔；8255A 的片选 $\overline{CS}$ 连至地址译码处的 200H~207H 插孔； PB_0 接到继电器的控制端；将 UP 脉冲按钮连接至 8259A 的 IR_0 插孔。

(2) 说明：8086 需要外接中断控制器才能对外部中断进行处理。8259 可外接 8 个中断源，本项目只响应 IR_0 中断。将单脉冲信号 UP 接到 8259 的 IR_0 脚，每产生一次中断时，就可以看到继电器控制的 LED 灯闪烁。

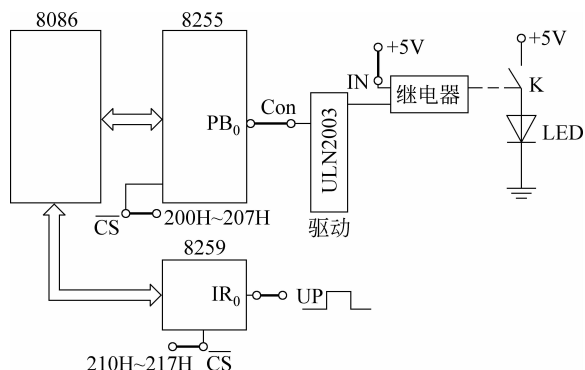


图 3-1 用外部中断控制继电器电路原理框图

3. 电路原理框图

电路原理框图如图 3-1 所示。电路由 8086 CPU、8255A 芯片、8259A 芯片、继电器及驱动和脉冲按钮 UP 组成。

4. 程序设计

1) 程序流程图

用外部中断控制继电器程序流程图如图 3-2 所示。

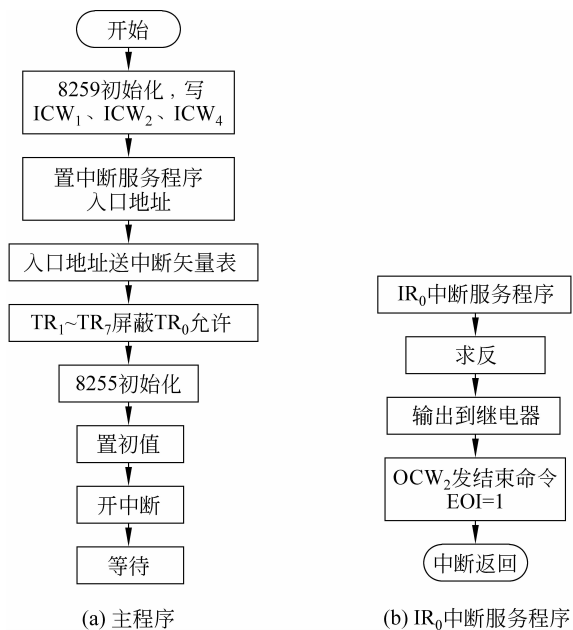


图 3-2 用外部中断控制继电器程序流程图

2) 程序清单

外部中断控制继电器程序清单如下。

```
CODE SEGMENT
```

```
    ASSUME CS:CODE
```

```
START: MOV     AL,13H
```

```
          ;00010011B,ICW1: 边沿触发,单片,要 ICW4
```

```

MOV    DX,210H                ;8259 地址
OUT    DX,AL
MOV    AL,8                    ;ICW2 中断类型为 8
MOV    DX,211H
OUT    DX,AL
MOV    AL,01H                  ;ICW4 不用缓冲方式,正常中断结束,非特殊的全嵌套方式
OUT    DX,AL
MOV    AX,0                    ;清零
MOV    DS,AX                  ;数据段清零
LEA    AX,INT0                 ;写 8259 中断程序的入口地址
MOV    DS:[4 * 8],AX           ;把中断服务程序的入口地址偏移量送中断矢量表
MOV    AX,CS
MOV    DS:[4 * 8 + 2],AX       ;把中断服务程序的入口地址段地址送中断矢量表
IN     AL,DX                   ;读中断屏蔽寄存器 IMR,
AND    AL,0FEH                 ;屏蔽 IR1 ~ IR7,允许 IR0 中的中断请求
OUT    DX,AL
MOV    DX,203H                 ;8255 初始化
MOV    AL,80H                  ;B 口输出,方式 0
OUT    DX,AL
MOV    BL,01H                  ;置继电器动作初值
STI                                ;开中断
AGAIN: HLT
      JMP    AGAIN             ;等待
INT0   PROC    NEAR             ;IR1 中断服务程序
      MOV    DX,201H            ;8255 的 PB 口地址
      NOT    BL                  ;求反
      MOV    AL,BL
      OUT    DX,AL              ;PB0 输出
      MOV    DX,210H
      MOV    AL,20H              ;OCW2 发结束命令 EOI=1
      OUT    DX,AL
      IRET
      INT1   ENDP
CODE   ENDS
      END    START

```

3.1 中断概述

3.1.1 中断基本概念

1. 中断

中断通过硬件来改变 CPU 的运行方向。当 CPU 在执行程序时,由内部或外部的原因引起的随机事件要求 CPU 暂时停止正在执行的程序,而转向执行一个用于处理该随机事

件的程序,处理完后又返回被中止的程序断点处继续执行,这一过程称为中断。中断之后所执行的相应处理程序通常称为中断服务或中断处理子程序,原来正常运行的程序称为主程序。主程序被断开的位(或地址)称为“断点”。引起中断的原因,或能发出中断申请的来源,称为“中断源”。中断源要求服务的请求称为“中断请求”(或“中断申请”)。中断示意图如图 3-3 所示。

2. 中断的特点

1) 提高 CPU 的效率

中断可以解决高速的 CPU 与低速的外设之间的矛盾,使 CPU 和外设并行工作。CPU 在启动外设工作后继续执行主程序,同时外设也在工作。每当外设做完一件事就发出中断申请,请求 CPU 中断它正在执行的程序,转去执行中断服务程序,中断处理完之后,CPU 恢复执行主程序,外设也继续工作。这样,CPU 可启动多个外设同时工作,大大地提高了 CPU 的效率。

2) 实时处理

在实时控制系统中,工业现场的各种参数和信息都会随时间而变化。这些外界变量可根据要求随时向 CPU 发出中断申请,请求 CPU 及时处理中断请求。如果满足中断条件,CPU 就会立刻响应,转入中断处理,从而实现实时处理。

3) 故障处理

控制系统的故障和紧急情况难以预料,如掉电、设备运行出错等,可通过中断系统由故障源向 CPU 发出中断请求,再由 CPU 转到相应的故障处理程序进行处理。

3. 中断的功能

1) 实现中断响应和中断返回

当 CPU 收到中断请求后,CPU 要根据相关条件(如中断优先级、是否允许中断)进行判断,决定是否响应这个中断请求。若响应,则在执行完当前指令后立刻响应这一中断请求。CPU 中断过程为:第一步,将断点处的 PC 值(即下一条应执行指令的地址)压入堆栈保留下来(这称为保护断点,由硬件自动执行)。第二步,将有关的寄存器内容和标志 PSW 状态压入堆栈保留下来(这称为保护现场,由用户自己编程完成)。第三步,执行中断服务程序。第四步,中断返回,CPU 将继续执行原主程序。中断返回过程为:首先恢复原保留寄存器的内容和标志位的状态,这称为恢复现场,由用户编程完成。然后,再加返回指令 IRET。IRET 指令的功能是恢复 PC 值,使 CPU 返回断点,这称为恢复断点。

2) 实现优先权排队

中断优先权也称为中断优先级。中断系统中存在着多个中断源,同一时刻可能会有不止一个中断源提出中断请求,因此需要给所有中断源安排不同的优先级别。CPU 可通过中断优先级排队电路首先响应中断优先级高的中断请求,等到处理完优先级高的中断请求后,再来响应优先级低的中断请求。

3) 实现中断嵌套

当 CPU 响应某一中断时,若有中断优先级更高的中断源发出中断请求,CPU 会暂停正在执行的中断服务程序,并保留这个程序的断点,转向执行中断优先级更高的中断源的中断

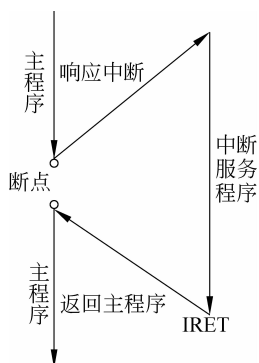


图 3-3 中断示意图

服务程序,等处理完这个高优先级的中断请求,再返回来继续执行被暂停的中断服务程序,这个过程称为中断嵌套。中断嵌套流程图如图 3-4 所示。

4. 中断源

引发中断的事件称为中断源。8086 CPU 的中断源有两类:一是内部中断源,由 CPU 内部的标志 TF 为 1 或执行一条软件中断指令而引起;二是外部中断,由外部请求信号引起。外部中断又分为可屏蔽中断(INTR)和非屏蔽中断(NMI)。中断源形式如图 3-5 所示。

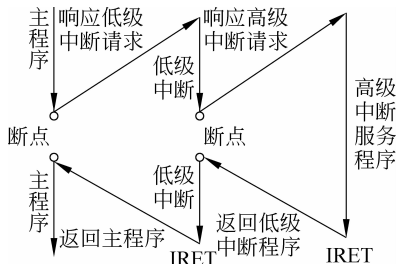


图 3-4 中断嵌套流程图

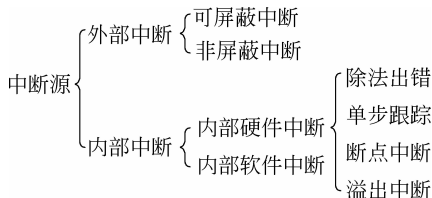


图 3-5 8086 CPU 的中断源

5. 禁止中断与中断屏蔽

在有些情况下,CPU 不允许响应可屏蔽中断 INTR,当中断源向 CPU 申请中断后,CPU 不能终止当前正在运行的程序并转到中断服务程序,这种情况就是禁止中断。

在另一些情况下,例如有多个中断源发出申请时,CPU 只响应其中某些中断,而屏蔽其他类型的中断,这就叫中断屏蔽。中断屏蔽可以通过中断屏蔽触发器来实现。将中断源对应的中断屏蔽触发器置 1,则中断源的中断请求被屏蔽,否则该中断源的中断请求被允许。

3.1.2 中断类型与中断向量表

8086 CPU 的中断系统能够处理 256 个不同的中断源,每个中断源都有一个为它服务的中断处理程序。CPU 响应中断后,就要转入相应的中断处理程序中。下面就介绍 CPU 如何转入中断处理程序。

- 中断类型号: 就是为每个中断源指定一个编号,称为中断类型号。
- 中断矢量: 每一个中断服务程序有一个确定的入口地址,该地址称为中断矢量。
- 中断矢量表: 把系统中所有的中断矢量集中起来,按中断类型号从小到大的顺序安排到存储器的某一个区域内,这个存放中断矢量的存储区叫中断矢量表。

80x86 CPU 的中断矢量表位于主存储器最低地址 00000H~003FFH 中,共占用 1024 个存储器单元。中断矢量表中共有 256 项,能处理 256 级中断矢量,对应中断类型号是 00H~0FFH,中断矢量表的结构如图 3-6 所示。

中断类型号	内容	地址
0	类型0的(IP)	00000H
	类型0的(CS)	
1	类型1的(IP)	00004H
	类型1的(CS)	
2	类型2的(IP)	00008H
	类型2的(CS)	
255	类型255的(IP)	003FCH
	类型255的(CS)	

图 3-6 中断矢量表结构图

中断服务程序的入口地址在中断矢量表中,占4个字节,其中两个低地址字节中存放中断服务程序入口地址的段内地址偏移量(IP)部分,两个高地址字节中存放中断服务程序入口地址的段地址(CS)部分。

在中断矢量表中,各个中断矢量表按中断类型码从0到255顺序存放。这样,知道了中断类型码,很快就可计算出相应中断矢量的存放位置,从而取出中断矢量。而每个中断矢量的地址可由中断类型码乘以4计算出来。CPU响应中断时,只要把中断类型码N左移2位(乘以4),即可得到中断矢量在中断矢量表中的对应地址4N,然后在两个低地址字节中存放中断服务程序入口地址的段内地址偏移量(IP)部分,在两个高地址字节中存放中断服务程序入口地址的段地址(CS)部分。

【例 3-1】 中断类型码为 27H,中断矢量为:

00100111B(27H)

↓ 左移2位(乘以4)

10011100B(9CH)

中断矢量应存放在从 0000H : 009CH 开始的4个连续字节单元中,即:

(0000H : 009CH, 0000H : 009DH) → IP

(0000H : 009EH, 0000H : 009FH) → CS

如(0000H : 009CH) = 21H

(0000H : 009DH) = 43H

(0000H : 009EH) = 65H

(0000H : 009FH) = 87H

则 27H 号中断的中断服务程序的入口地址为 8765H : 4321H。

【例 3-2】 中断类型码为 17H,如中断服务程序的入口地址是 3240H : 87FFH,试指出中断矢量表中存放该中断矢量的4个字节单元的地址及内容。

解: 由于中断类型码为 17H,则中断矢量为:

00010111B(17H)

↓ 左移2位(乘以4)

010111B(5CH)

中断矢量应存放在从 0000H : 005CH 开始的4个连续字节单元中,即:

(0000H : 005CH) = FFH

(0000H : 005DH) = 87H

(0000H : 005EH) = 40H

(0000H : 005FH) = 32H

3.1.3 中断响应过程

1. 8086 中断响应过程

8086 的中断响应过程如图 3-7 所示。从图中可以看出,该流程已经考虑了不同中断源的响应优先级,当多个中断同时申请中断时,CPU 先响应内部中断,其次响应非屏蔽中断 NMI,然后响应可屏蔽中断 INTR,最后响应单步中断。这个中断响应过程,不同的中断略有不同。

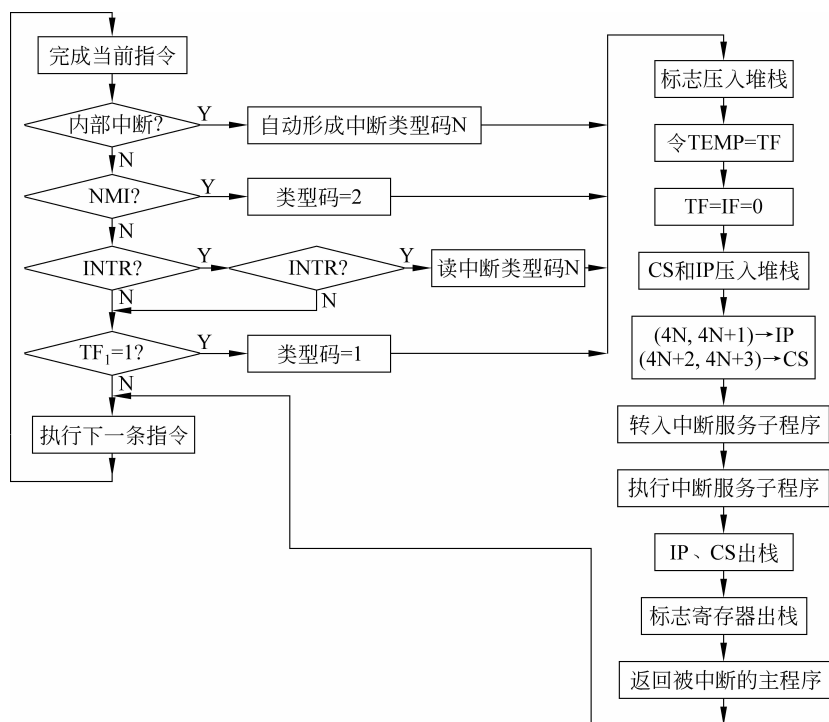


图 3-7 8086 的中断响应过程

2. INTR 中断响应过程

不同的微型计算机的中断系统略有所不同,但实现中断时的过程基本相同。中断过程包括 4 个阶段:中断请求、中断响应、中断服务、中断返回。下面以 INTR 中断为例来说明中断的响应过程。

CPU 在 INTR 引脚上接到一个中断请求信号,若此时的中断允许标志 $IF=1$,并且,当前的中断请求具有最高的优先级,CPU 就在执行完当前指令以后开始响应外部的中断请求。这时,CPU 通过 $\overline{INTA}$ 引脚连续发出两个负脉冲信号,外设接口在接到第二个负脉冲以后,就在数据线上发送中断类型号,CPU 接到此中断类型号后,将进行如下 5 步操作:

第 1 步,将中断类型号送入暂存器保存。

第 2 步,将标志寄存器 PSW 内容压入堆栈,以保护中断时的状态。

第 3 步,把 IF 和 TF 标志清零。IF 清零是为了防止在中断响应的同时又来别的中断,而 TF 清零是为了防止 CPU 以单步方式执行中断处理程序。

第 4 步,保护断点。就是将当前程序执行的下一条指令的 IP 和 CS 的内容压栈。保护断点的作用是为了执行中断服务程序后能返回主程序。

第 5 步,根据得到的中断类型号,在中断向量表中找出中断向量,将其装入 IP 和 CS,这时程序就转向中断服务程序去执行。

3.1.4 8086 中断结构

8086 中断结构如图 3-8 所示,从图中可以看出,8086 中断系统分为外部中断和内部中

断两大类。外部中断也称为硬件中断,它由非屏蔽中断 NMI 和可屏蔽中断 INTR 构成。内部中断也称为软件中断,它由除法错误中断、溢出中断、单步中断、断点中断和 INT n 指令中断构成。同时 8086 CPU 规定了各类中断的优先级,最高为除法错误中断 INT 0、溢出中断 INT 0 及 INT n 指令、非屏蔽中断 NMI、可屏蔽中断 INTR,最低的是单步中断。下面简单介绍。

1. 外部中断

8086 CPU 有两条外部中断请求线: NMI 和 INTR,即 NMI(17 引脚)非屏蔽中断和 INTR(18 引脚)可屏蔽中断。

(1) NMI 非屏蔽中断: 中断类型号为 2, NMI 非屏蔽中断的中断请求信号从 CPU 的外部送往 CPU 的引脚 NMI 上,采用边沿触发方式。

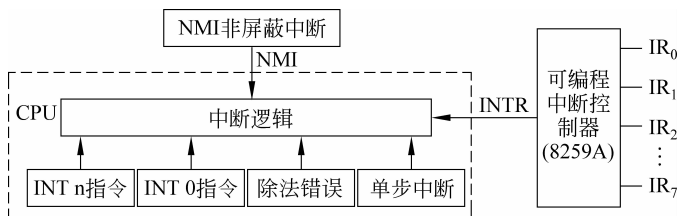


图 3-8 8086 中断结构

(2) INTR 可屏蔽中断: 中断类型号为 08H~0FH。INTR 可屏蔽中断的中断源一般是外部设备,可屏蔽中断请求信号通过 INTR 引脚输入,所有的可屏蔽中断请求公用一条 INTR 线,由可编程中断控制器 8259A 管理。

需要注意的是非屏蔽中断与可屏蔽中断的不同点:

- 可屏蔽中断 INTR 由 CPU 的程序状态控制字中的 IF 位为 1 还是为 0 决定的。IF=1 时,CPU 响应由 INTR 端引入的中断申请;IF=0 时,CPU 不响应由 INTR 端引入的中断申请。由 INTR 端引入,依靠 IF 端置 0 还是置 1 决定 CPU 是否响应的中断称为可屏蔽中断。IF 位置 0 可通过指令 CLI 实现,称为关中断。IF 位置 1 通过指令 STI 实现,称为开中断。
- 由 NMI 端引入,不受 IF 位状态的控制,只要有中断申请就必须响应的中断称为非屏蔽中断。

2. 内部中断

内部中断的中断源在 CPU 的内部。内部中断包括除法错误中断、溢出中断、单步中断、断点中断和 INT n 指令中断等。内部中断是由 CPU 内部事件引发的中断。

(1) 除法错误中断: 除法错误中断的中断类型号为 0。当 CPU 执行除法操作,除数为零或商超出计算机可表示的最大数值范围时,产生 0 号,除法错误中断。

(2) 单步中断: 单步中断的中断类型号为 1。当 PSW 标志寄存器的单步标志位 TF 为 1 时,每执行完一条指令,CPU 立即暂停程序的执行,产生 1 号中断。单步中断可用于进行程序的调试。

(3) 断点中断: 断点中断的中断类型号为 3。为了调试汇编语言程序经常需要设置断点,可以在程序中加入 INT 3 指令。

(4) 溢出中断: 溢出中断的中断类型号为 4,以指令 INTO 的形式出现。当 PSW 标志

寄存器的溢出标志 OF 为 1 时,且 CPU 执行 INTO 指令,则产生溢出中断。

(5) INT n 中断: INT n 中断也称为软件中断,以指令 INT n 的形式出现,中断类型号就是 n。CPU 执行 INT n 指令,产生类型号为 n 的内部中断。

3.2 8259A 芯片引脚和内部结构

3.2.1 概述

Intel 8259A 可编程中断控制器用于管理 PC 中的 INTR 中断(外部可屏蔽中断)。外部设备可通过中断控制器 8259A 的中断请求线 INT 把中断请求信号送往 CPU 的 INTR 线,以便提出中断请求。8259A 的主要功能如下。

- (1) 可管理 8 个中断源电路的中断,并能对其进行优先级管理。
- (2) 用 9 片 8259A 可组成主从式中断系统,管理 64 个中断源电路的中断,并能对其进行优先级管理。
- (3) 对中断源有屏蔽或允许申请中断的操作。
- (4) 能自动送出中断类型号,使 CPU 迅速找到中断服务程序的入口地址。

3.2.2 8259A 芯片引脚

8259A 为 28 脚双列直插式芯片,芯片引脚如图 3-9 所示,各引脚信号功能如下。

(1) $\overline{CS}$ (1 脚): 片选信号。输入,低电平有效。该信号有效时,CPU 可对该 8259A 进行读写操作。

(2) $\overline{WR}$ (2 脚): 写信号。输入,低电平有效。该信号有效时,允许 CPU 把命令字(ICW 和 OCW)写入相应命令寄存器。

(3) $\overline{RD}$ (3 脚): 读信号。输入,低电平有效。该信号有效时,允许该 8259A 将状态信息放到数据总线上供 CPU 检测。

(4) $D_0 \sim D_7$ (11~4 脚): 双向数据总线。用来传送控制、状态和中断类型号。

(5) $IR_0 \sim IR_7$ (18~25 脚): 外部中断请求信号。

(6) INT(17 脚): 中断请求信号。用来向 CPU 发送中断请求信号。

(7) $\overline{INTA}$ (26 脚): 中断响应信号。CPU 同意中断申请后,发此信号作为响应中断的回答信号。

(8) A_0 (27 脚): 地址输入信号,用于寻址 8259A 内部寄存器,一般与地址总线的 A_0 连接。同 $\overline{CS}$ 、 $\overline{WR}$ 和 $\overline{RD}$ 在 PC AT 机上的组合操作功能如表 3-1 所示。

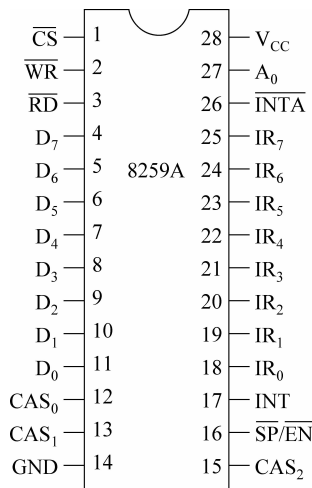


图 3-9 8259A 芯片引脚图

表 3-1 8259A 读/写操作与 I/O 端口地址

$\overline{\text{CS}}$	$\overline{\text{WR}}$	$\overline{\text{RD}}$	A_0	读/写操作	主片	从片
0	0	1	0	写 ICW_1 、 OCW_2 、 OCW_3	20H	0A0H
0	0	1	1	写 ICW_2 、 ICW_3 、 ICW_4 、 OCW_1	21H	0A1H
0	1	0	0	读 IRR 、 ISR ，查询命令	20H	0A0H
0	1	0	1	读 IMR	21H	0A1H

(9) $\text{CAS}_0 \sim \text{CAS}_2$ (12、13、15 脚): 级联信号。双向引脚, 用来控制多片 8259A 的级联使用。对主片来说, $\text{CAS}_0 \sim \text{CAS}_2$ 为输出; 对从片来说, $\text{CAS}_0 \sim \text{CAS}_2$ 为输入。

(10) $\overline{\text{SP}}/\overline{\text{EN}}$ (16 脚): 从片/允许缓冲器信号。此信号线为双向, 作用有二: 一是当为输入时, 用来决定本片是主片还是从片, 当 $\overline{\text{SP}}=1$, 该片为主片, 当 $\overline{\text{SP}}=0$, 该片为从片; 二是当为输出时, $\overline{\text{SP}}/\overline{\text{EN}}$ 可作为启动数据总线驱动器的启动信号。

3.2.3 8259A 芯片内部结构

8259A 的内部结构如图 3-10 所示, 各个部分的功能如下。

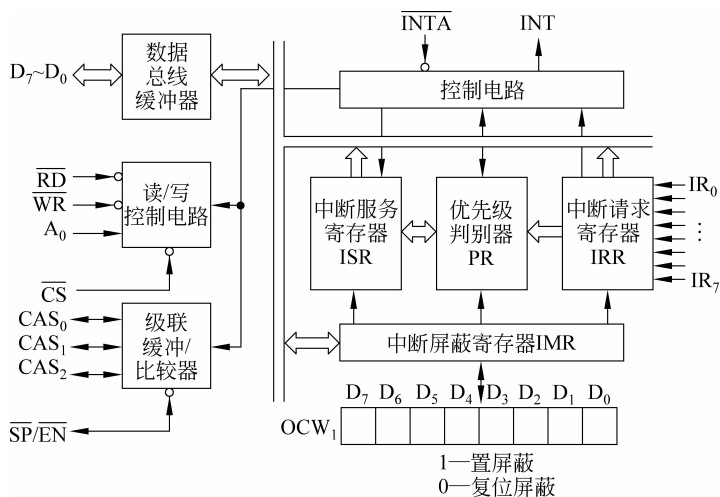


图 3-10 8259A 芯片的内部结构

1) 数据总线缓冲器

这是一个 8 位的双向三态缓冲器, 是 8259A 与 CPU 之间的数据接口, 在读/写逻辑的控制之下实现 CPU 与 8259A 之间的信息交换。

2) 读/写控制电路

读/写逻辑电路控制 CPU 与 8259A 之间的信息交换。为片选信号线, 低电平时选中 8259A 工作。由片选信号 $\overline{\text{CS}}$ 和 A_0 指定内部寄存器, CPU 可以通过执行 OUT 指令, 将初始化命令字和工作命令字写入相应的命令寄存器 ICW 和 OCW 中; 也可以通过执行 IN 指令, 将 8259A 中的 IRR、ISR、IMR 等寄存器的内容读入 CPU 中。