



第8章 中断技术

8.1 中断基础

8.2 8086 CPU的中断系统

8.3 可编程中断控制器8259A

8.4 8259A在微机中的编程应用



8.1 中断基础

一、中断的基本概念

中断：是指**CPU**正常运行时，由于内、外部事件或由程序预先安排引起的，**CPU**暂停正在运行的程序而转去执行内、外部事件或预先安排的事件服务程序，待处理完毕后又回到原来被中止的程序处继续执行的过程

中断源：产生中断请求的外设或引发内部中断的原因和事件。

中断请求：中断源为获得**CPU**处理而向**CPU**发出的请求信号。

中断响应：**CPU**接到中断源产生的中断请求信号后，若决定响应此中断请求，则向外设发出中断响应信号的过程。

中断屏蔽：禁止中断响应称为中断屏蔽。

主程序：**CPU** 在接收到中断请求时当前执行的程序。



8.1 中断基础

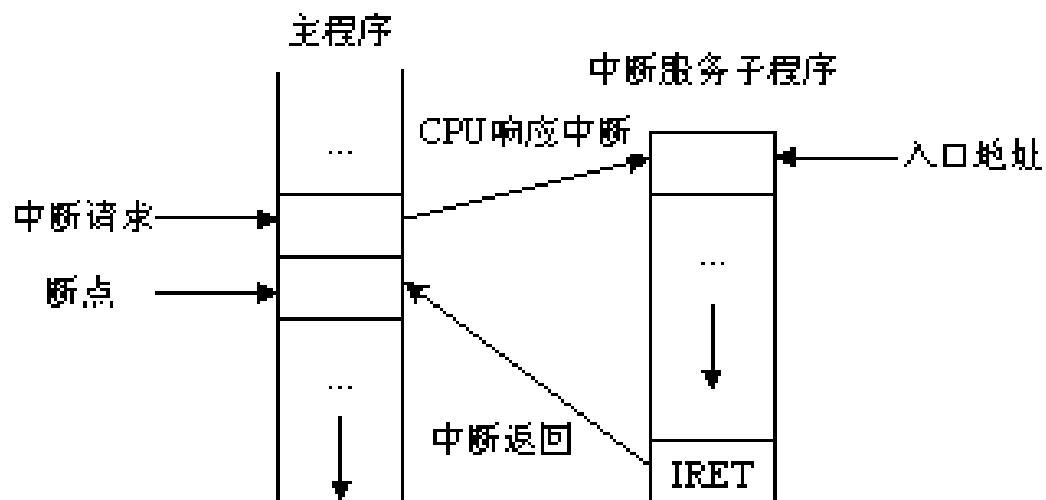
断点： CPU处理中断时，原程序的暂时中止处（即中断返回后继续执行的指令地址）。

中断服务程序： 中断源发出中断请求后请求CPU执行的处理程序。

入口地址： 中断服务程序首条指令的地址。

中断处理： CPU执行中断服务程序的过程。

中断返回： 当 CPU 执行完中断服务程序后，返回原程序断点处继续执行后续程序的过程。



微机响应中断的过程与执行转移指令**CALL**的过程非常类似。但中断是**随机发生**的。

图8.1 中断引起程序转移示意图



8.1 中断基础

二、中断优先级与中断嵌套

微机系统中存在多个中断源，往往会出现以下情况：

- (1) 多个中断源在同一时间向**CPU**发出中断请求信号；
- (2) 当**CPU**正在响应某一中断源的请求，执行相应中断服务程序时，又有别的中断源产生新的中断请求。

而 **CPU** 某一时刻只能响应一个中断请求。

优先级（优先权）：

CPU依据各中断源所请求任务的轻重缓急，将中断源分成若干个级别，这个级别就是中断优先级。

第（1）情况：**CPU**选择多个中断源中优先级最高的中断请求予以响应。

第（2）情况：当**CPU**正在处理中断时，也能响应优先级更高的中断请求，但屏蔽同级或低级的中断请求，这就是所谓**多重中断或中断嵌套**。

8.1 中断基础

解决中断的优先级的方法：

1. 软件查询方式

- ❑ 一方面各中断源的中断请求信号都被锁存于锁存器（中断状态端口）中，相应位置1；
- ❑ 另一方面，所有的中断请求信号相“或”后作为INTR信号，向CPU提出中断请；
- ❑ CPU响应中断后，通过一段公共查询程序来确定相应的中断源，并转入相应的中断服务程序。
- ❑ 查询程序中查询状态端口的顺序就是中断源的优先级顺序。

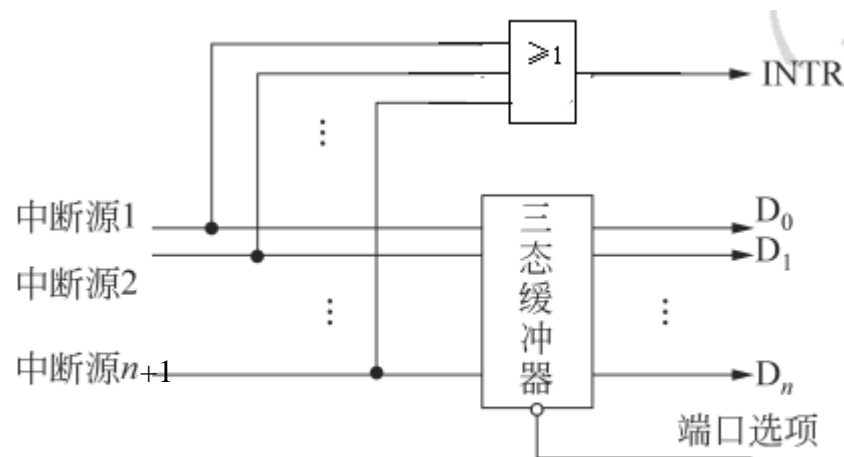


图 8.2 软件查询方式接口电路

优点：硬件简单，通过修改查询顺序即可改变中断源优先级

缺点：中断源较多时耗时较长，影响中断响应的实时性。

8.1 中断基础

2. 硬件排队方式

硬件排队方式：利用专门的硬件电路实现中断源优先级排队

(1) 链式优先级排队

在每个中断源接口电路中设置一个称为菊花链的逻辑电路，利用中断源在系统中的物理位置确定其中断优先级。

特点：调整优先级就涉及硬件改动，使用不方便。

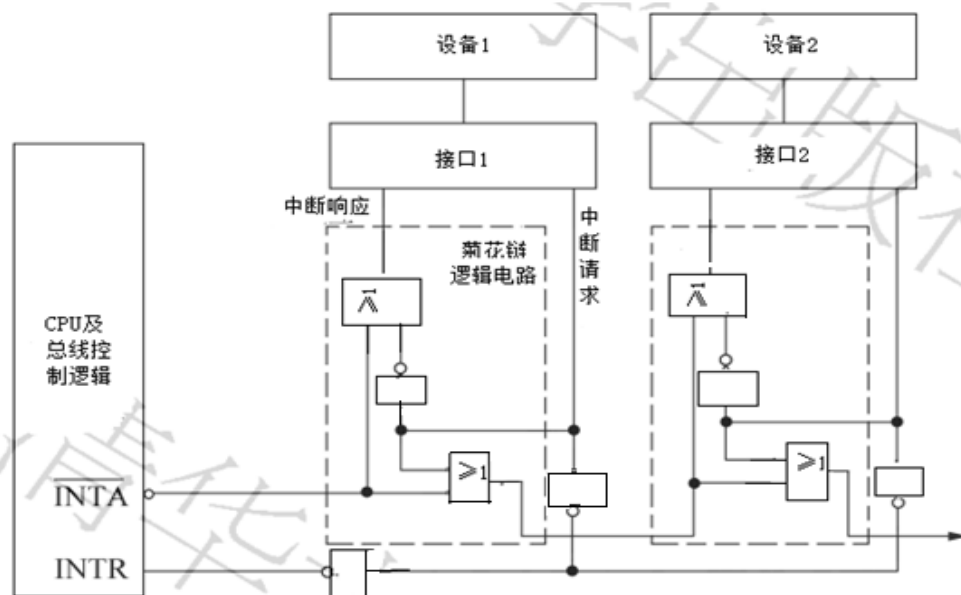


图 8.4 链式优先权排队电路

8.1 中断基础

2. 硬件排队方式

硬件排队方式：利用专门的硬件电路实现中断源优先级排队

(2) 向量优先级排队

目前微机系统中多用专门的优先级中断控制器构成向量优先级中断系统来管理中中断优先级。用户可通过编程实现，调整灵活，无需改动硬件接口电路。

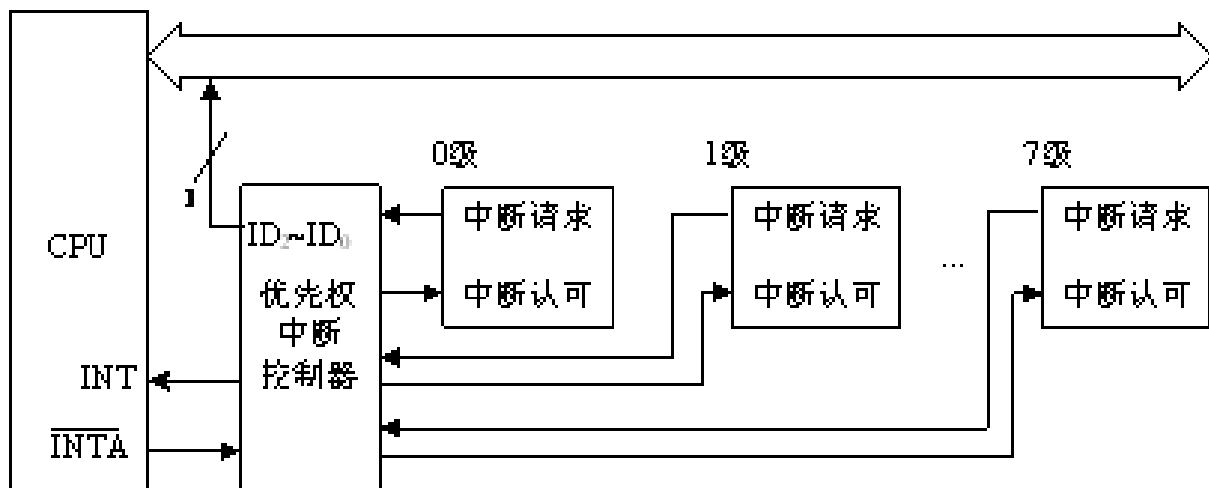


图8.5 向量优先级中断系统示意图



8.1 中断基础

三、中断过程

中断请求、中断响应、中断处理和中断返回四个阶段。

1. 中断请求

中断源向**CPU**提出中断请求的两个条件：

- (1) 中断源（如外设）需**CPU**为其服务，且本身已经准备就绪
- (2) 系统允许该中断源提出申请。多中断源情况下，常常在外设接口电路中设置一个**中断屏蔽寄存器**，只有在该中断源的中断请求未被屏蔽时，其中断请求才能送到**CPU**。

两个条件需同时满足



8.1 中断基础

2. 中断响应

当**CPU**检测到有中断请求发生时，其有权决定是否对该中断请求予以响应。若**CPU**允许中断，则予以响应，否则不予响应。当**CPU**决定响应中断请求时，进入中断响应周期。

CPU响应中断时要自动完成以下三项任务：

（1）**关中断**。因为**CPU**响应中断后，要进行必要的处理，在此期间不允许其它中断源来打扰。

（2）**断点保护**。通过内部硬件**保存断点及标志寄存器内容**，以便中断处理完毕后能正确返回。

（3）**获得中断服务程序的入口地址**。**CPU**响应中断后，将以某种方式查找中断源，获得中断服务程序的入口地址，转向对应的中断服务程序。

通常前两步由硬件完成，最后一步由硬件或软件实现



8.1 中断基础

3. 中断处理

中断处理（中断服务），由中断服务程序完成，不同的中断服务程序完成不同的功能，一般在中断服务程序中要做以下几项工作。

（1）保护现场

为了中断处理程序不破坏主程序中寄存器的内容，将断点处各寄存器的内容压入堆栈保护起来后再进入中断处理。

（2）开中断

中断响应阶段，硬件控制自动执行关中断，保护CPU在中断响应时不会被再次中断。有时有比该中断更紧急的情况要处理，需停止对该中断的服务而转到优先级更高的中断服务程序，以实现中断嵌套。

中断过程中，可以多次开放和关闭中断，一般只在程序的关键部分才关闭中断，其它部分则要开放中断以允许中断嵌套。



8.1 中断基础

(3) 中断服务

中断服务是主体部分，不同的中断请求中断服务内容不同。需据中断源要完成的功能，事先编写相应中断服务程序存入内存，等待中断请求响应后调用执行。

(4) 关中断

若第二步中执行了开中断，则需关中断以为恢复现场做准备

(5) 恢复现场

中断服务处理完毕后，在返回主程序前需要将前面通过 **PUSH** 指令保护的寄存器内容从堆栈中弹出，以便返回到主程序后能继续正确运行。

注意： **POP** 指令顺序应按先进后出的原则与进栈指令一一对应

8.1 中断基础

4. 中断返回

中断服务程序的最后一条是中断返回指令**IRET**。

执行该指令，使**CPU**转到被中断的程序中继续执行。

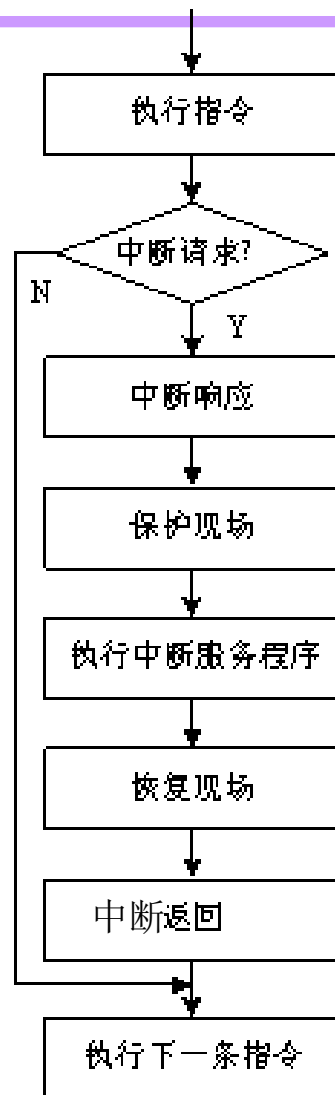


图8.6 中断过程示意图

8.2 8086 CPU的中断系统

一、8086 CPU 中断类型

8086 CPU中断系统可处理256种不同类型中断，并给每种中断都赋予了一个中断类型号，编号为0~255。CPU根据中断类型号来识别不同的中断源。

内部中断：中断源来自CPU内部

外部中断：中断源来自CPU外部

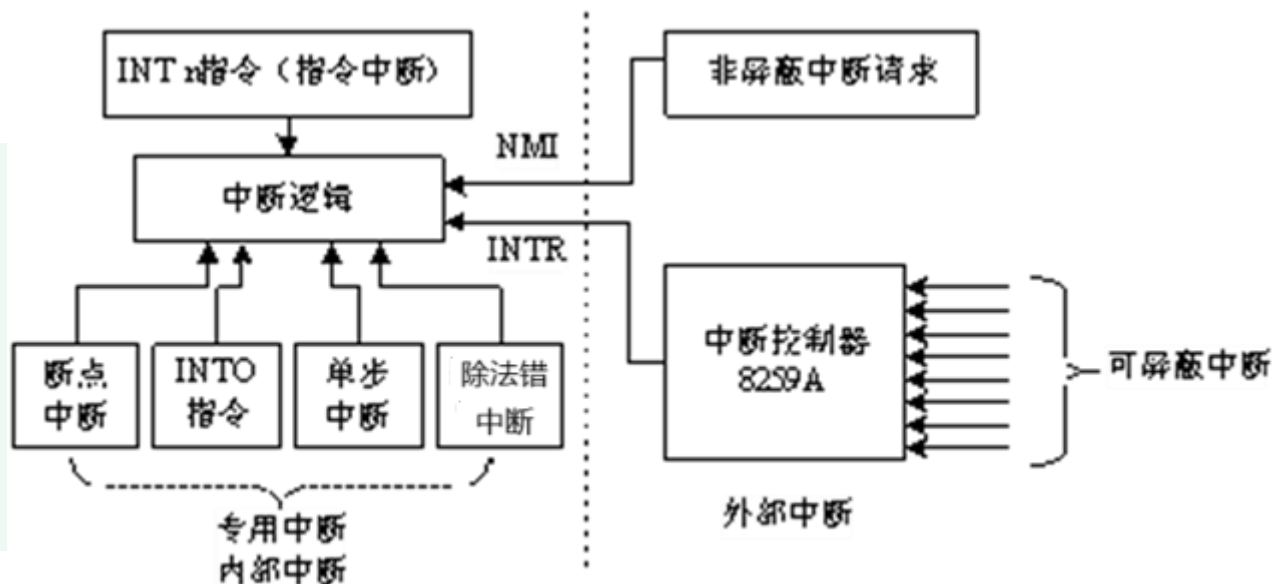


图8.7 8086 CPU中断源分类



8.2 8086 CPU的中断系统

1. 外部中断

外部中断（硬件中断），由外部设备通过硬件请求的方式所产生的中断。

（1）不可屏蔽中断NMI

当外设通过非屏蔽中断请求信号**NMI**向**CPU**提出中断请求时，**CPU**当前指令执行结束后，立即**无条件的**予以响应，执行**中断类型为2**的不可屏蔽中断。主要用于**紧急情况**的故障处理，如电源掉电、存储器读/写错误、扩展槽中输入/输出通道错误等。

（2）可屏蔽中断INTR

8086 CPU的**INTR**引脚收到一个高电平信号。受**CPU**的中断允许位**IF**控制。

IF=1，**CPU**可以响应（**STI**）

IF=0，禁止**CPU**响应（**CLI**）

绝大部分外部设备提出的中断请求都是可屏蔽中断



8.2 8086 CPU的中断系统

2. 内部中断

内部中断（软件中断），是由**CPU**运行程序错误或执行内部程序调用所引起的一种中断。内部中断（除单步中断）也是不可屏蔽，中断类型号固定。

（1）除法错中断

除法运算时，除数为零或者商超出了寄存器所能表示的范围，则会产生一个中断类型号为0的内部中断。

（2）单步中断

如果**CPU**的标志寄存器**Flags**中的单步标志**TF**位为1，则在每条指令执行后就引起一次中断，使程序单步执行。单步中断为用户调试程序提供了强有力的手段，其中断类型号为1。

PUSHF		；标志寄存器FLAGS入栈
POP	AX	；AX←FLAGS内容
OR	AX, 0100H	；使AX（即标志寄存器）的D ₈ =1，其余位不变
PUSH	AX	；AX入栈
POPF		；FLAGS寄存器←AX



8.2 8086 CPU的中断系统

(3) 溢出中断

Flags中的溢出标志位**OF=1**，且执行**INTO**指令时产生，中断类型为**4**。

该中断的产生需满足两个条件：**OF**位为**1**，且执行**INTO**指令，两者缺一不可。溢出中断通常在用户需要对某些运算操作进行溢出监控时使用。

(4) 断点中断

一条专用于设置断点的指令**INT 3H**，专门用于在程序中设置断点来调试程序，即断点中断。

(5) 中断指令**INT n**

INT n是用户自定义的软中断指令。其中**n**为中断类型号（范围为**0~255**）。



8.2 8086 CPU的中断系统

二、8086 CPU响应中断的过程

1、8086中断响应的流程

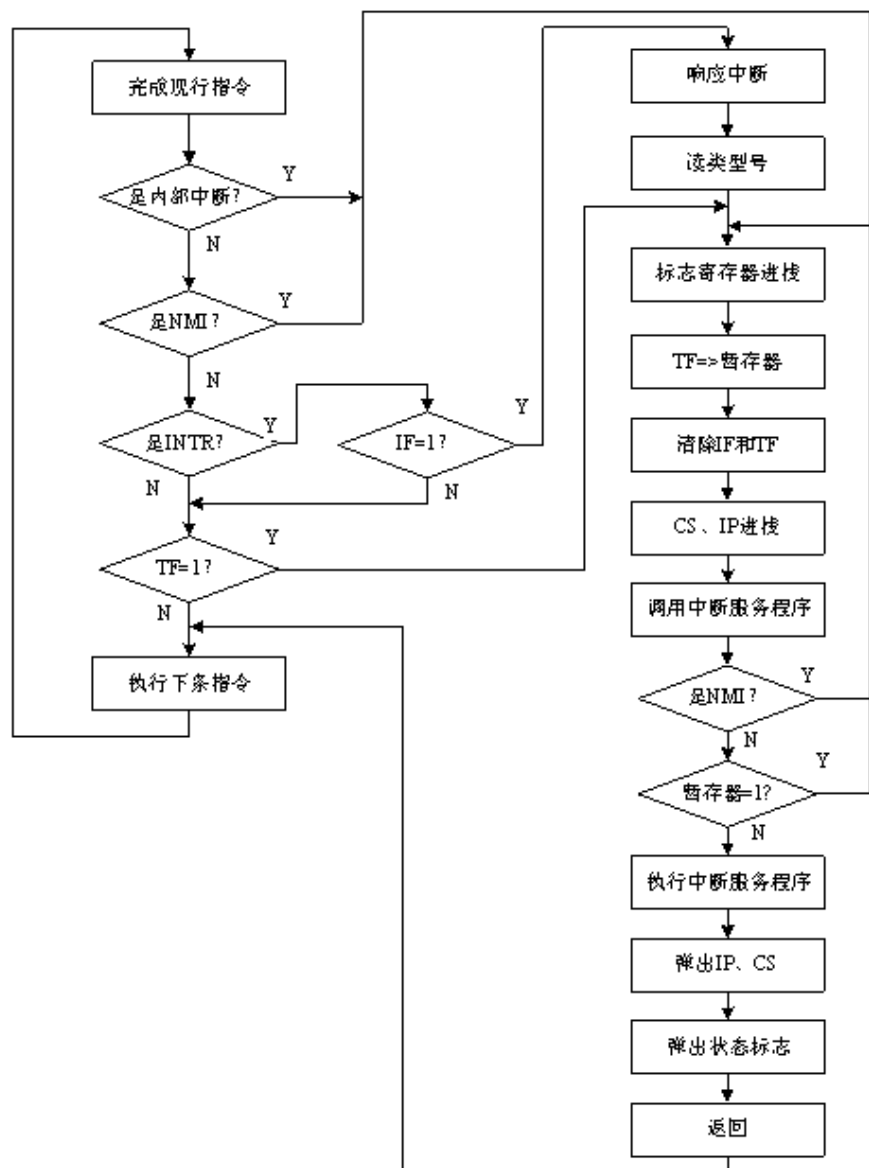


图8.8 8086 中断响应流程图



8.2 8086 CPU的中断系统

2、8086 CPU中断源的优先级顺序

内部中断（单步中断除外）的优先级高于外部中断，外部中断中不可屏蔽中断的优先级高于可屏蔽中断，单步中断的优先级最低。在**CPU**中断响应过程中，其仍然能对不可屏蔽中断**NMI**和单步中断予以响应。

表8-1 8086 CPU的中断优先级顺序

中 断	优 先 级
除法出错、INTO、INT n	最高
NMI	↓
INTR	
单步	最低



8.2 8086 CPU的中断系统

三、中断向量及中断向量表

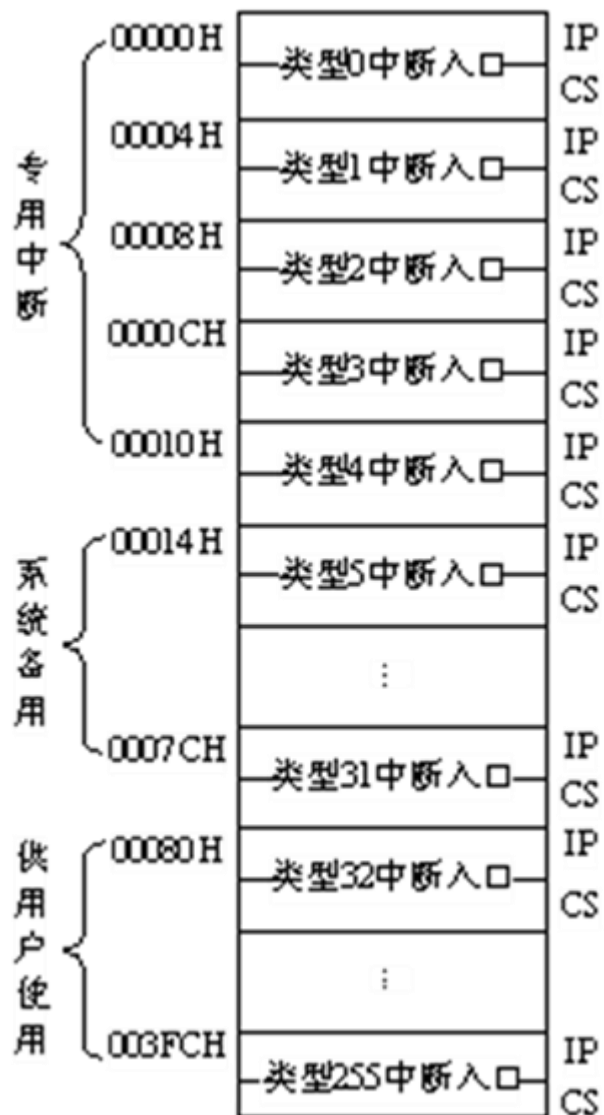
中断类型号：每个中断源的编号n，为0-255。

中断向量：每个中断服务程序的程序入口地址。每个中断向量都由段地址CS和偏移地址IP共4个字节组成。

中断向量表：存放中断向量的存储区。256个中断向量集中起来，按对应的中断类型号从小到大的顺序依次存放到了内存的最低端（00000H-003FFH），占用1K字节。

中断向量的地址：中断类型号对应的中断向量在中断向量表中的地址。

8.2 8086 CPU的中断系统



- 类型号为0~4的专用中断的中断向量分别存放在00H、04H、08H、0CH和10H开始的4个连续单元中，规定IP在前，CS在后。
- 将中断类型号n乘以4就能找到规定类型的中断向量。

图8.9 中断向量表

8.2 8086 CPU的中断系统

【例 8.1】已知某中断源的中断类型为 $n=40H$ ，其中断向量为 $1234:5678H$ ，分析该中断向量是如何存放在中断向量表中的？

分析：中断类型号 $n=40H$ ，故其中断向量存放在中断向量表中 $4n$ 开始的连续 4 个字节， $4n$ 和 $4n+1$ 存放中断向量的偏移地址， $4n+2$ 和 $4n+3$ 存放中断向量的段地址。
 $40H \times 4 = 100H$ ，即在 $00100H$ 开始的连续 4 个字节单元中依次存放 $78H$ 、 $56H$ 、 $34H$ 、 $12H$ ，具体如图 8.10

向量表地址	中断向量
00100H	78H
00101H	56H
00102H	34H
00103H	12H

图 8.10 40H 号中断在中断向量表中的存储示意图

8.3 可编程中断控制器8259A

一、8259A的内部结构和引脚

1. 8259A的内部结构

8259A芯片采用NMOS工艺制造，使用单一+5V电源供电。由数据总线缓冲器、读/写控制逻辑、级联缓冲/比较器、中断请求寄存器IRR、中断屏蔽寄存器IMR、中断服务寄存器ISR、优先权分析器PR及控制逻辑8大部分组成。

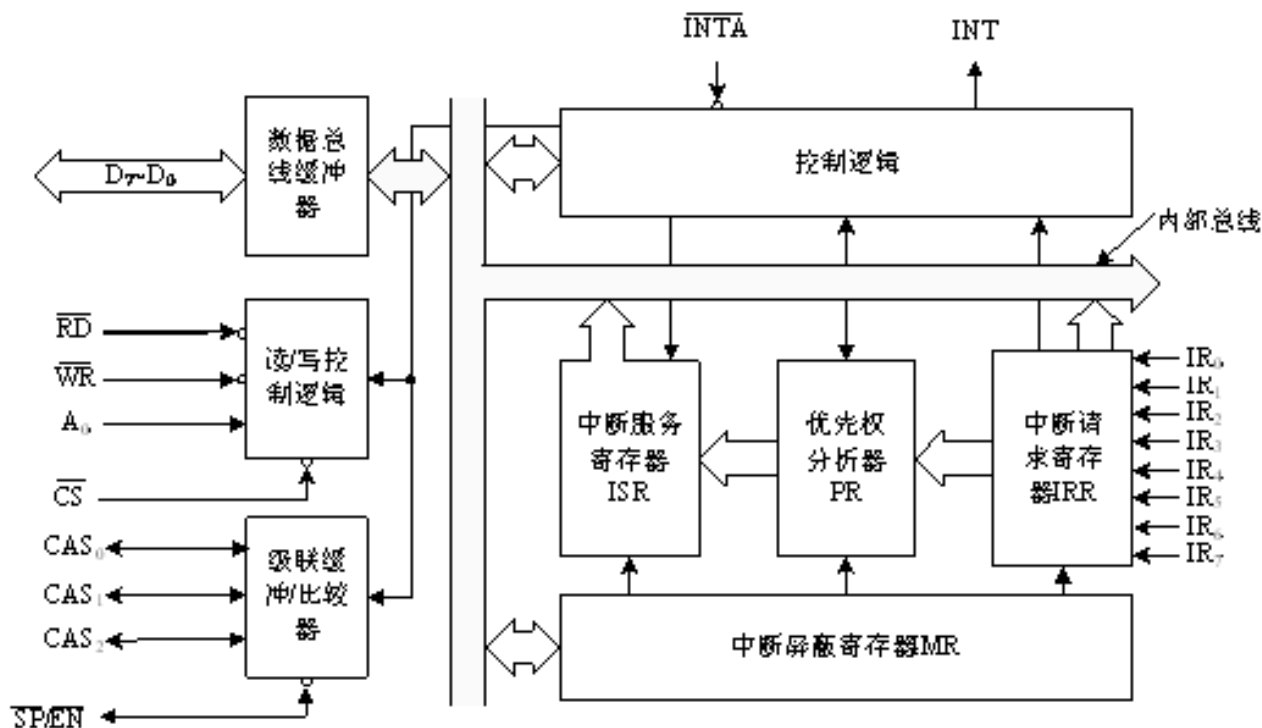


图8.11 8259A内部结构框图

8.3 可编程中断控制器8259A

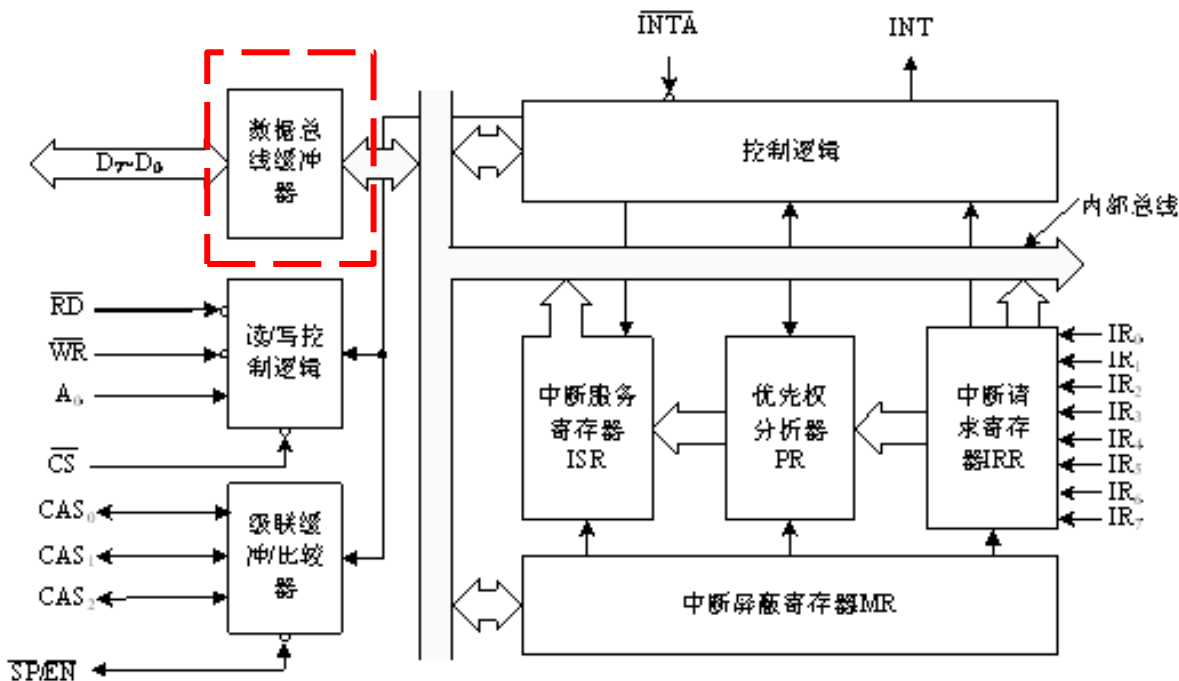


图8.11 8259A内部结构框图

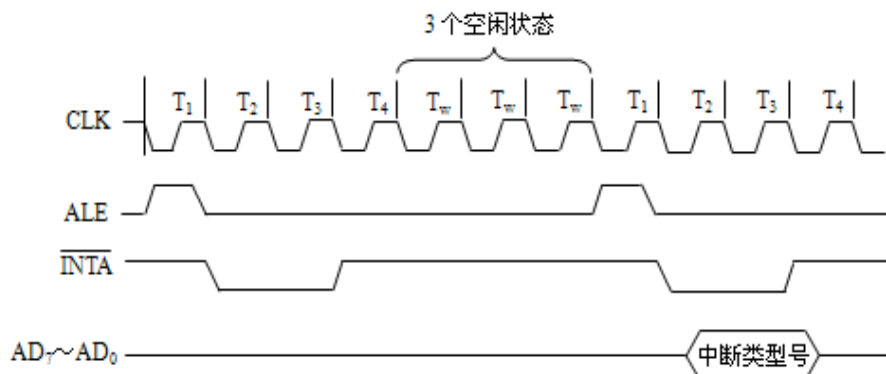


图 2.18 8086 中断响应周期时序

(1) 数据总线缓冲器

8位双向三态缓冲器，是8259A与系统数据总线的接口。通常连接低8位数据总线 $D_7 \sim D_0$ 。

CPU对8259A编程要写入的控制字、读出状态信息都由其完成。

8259A送至数据总线的中断类型号也是通过它传送的。

8.3 可编程中断控制器8259A

(2) 读/写控制逻辑

接收来自**CPU**的读/写命令，配合片选端的信号和**A₀**端的地址输入信号完成规定的操作。

把**CPU**送来的命令字传送到**8259A**中相应的命令寄存器中，把**8259A**中寄存器的内容输出到数据总线上。

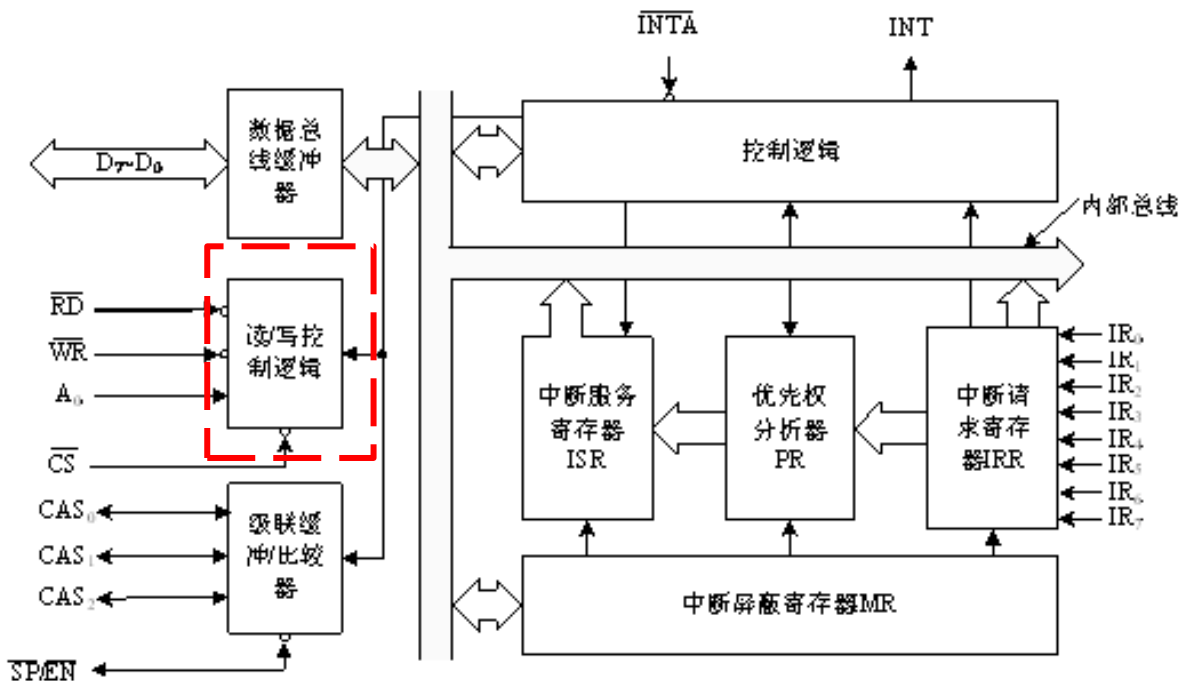


图8.11 8259A内部结构框图



8.3 可编程中断控制器8259A

(3) 级联缓冲/比较器

用于多片8259A的级联和数据缓冲方式。

当外部中断源超过8个时，就要通过多片

8259A的级联实现。此时，级联缓冲/比较器主要用来存放和比较系统中各相互级联的从片8259A的3位识别码。

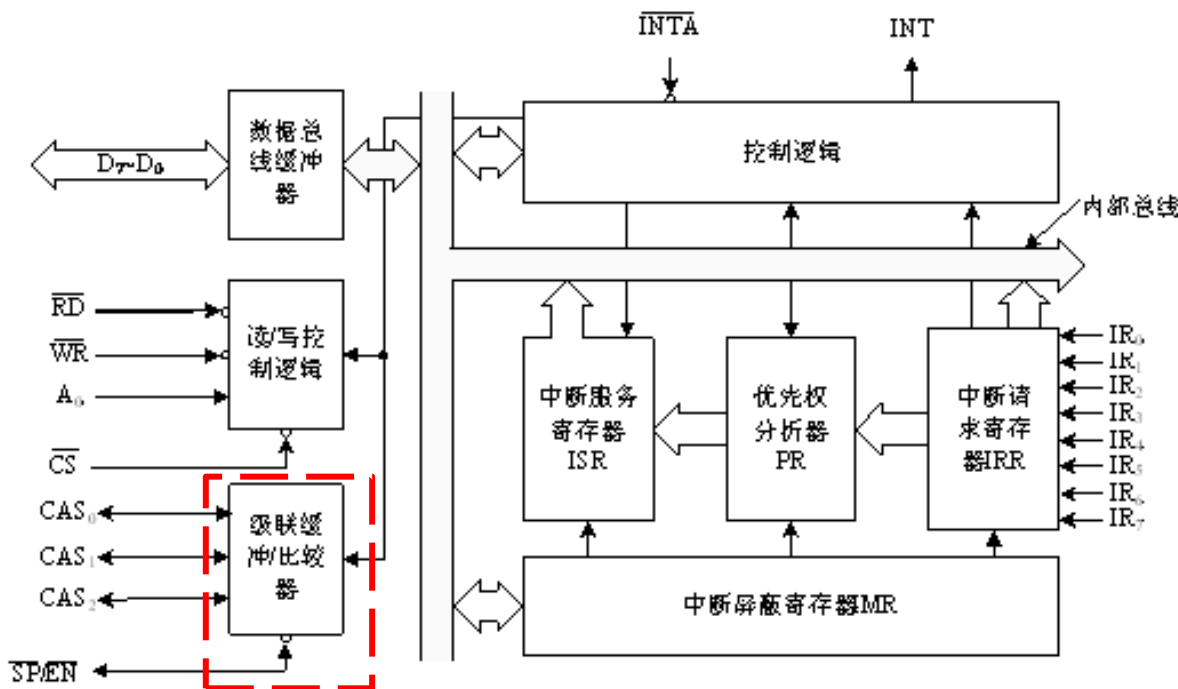


图8.11 8259A内部结构框图



8.3 可编程中断控制器8259A

(4) 中断请求寄存器 IRR

具有锁存功能的8位寄存器，存放外部输入的中断请求信号 $IR_7 \sim IR_0$ 。

当某个IR端有中断请求时，IRR寄存器中的相应位置1。

8259A允许8个中断请求信号同时进入，此时IRR被置成全1，当中断请求被响应时，IRR的相应位复位。

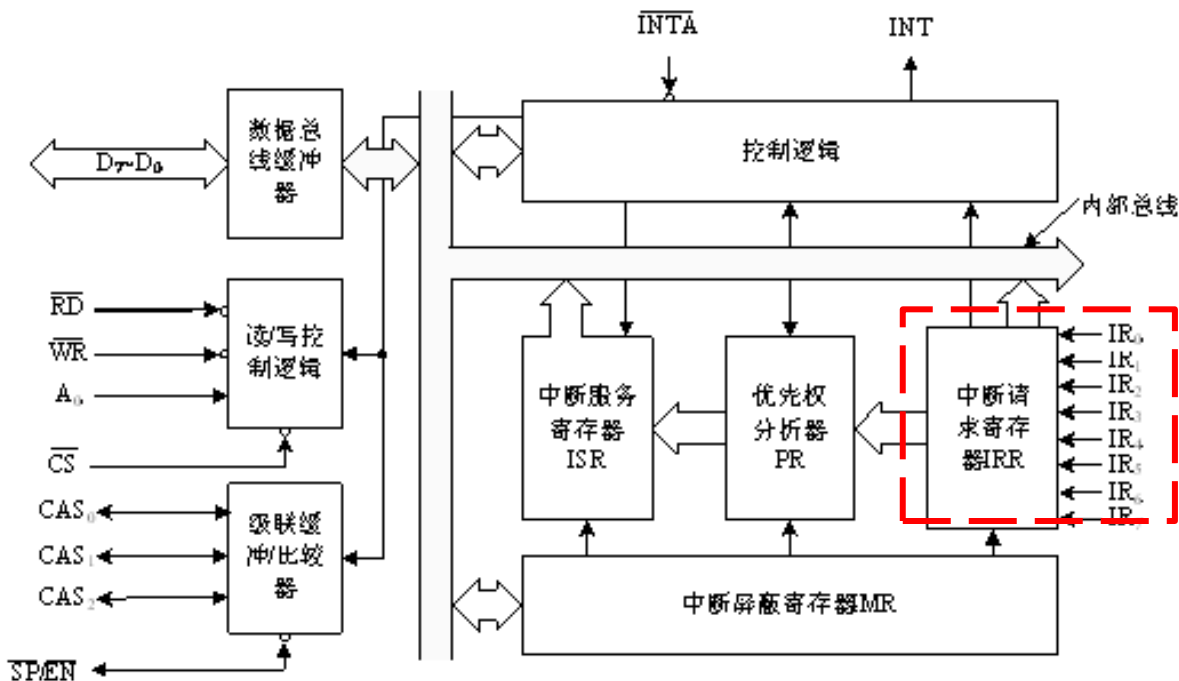


图8.11 8259A内部结构框图

外设产生中断请求的两种方式:

- ✓ 电平触发方式
- ✓ 边沿触发方式

何种触发方式通过编程决定

8.3 可编程中断控制器8259A

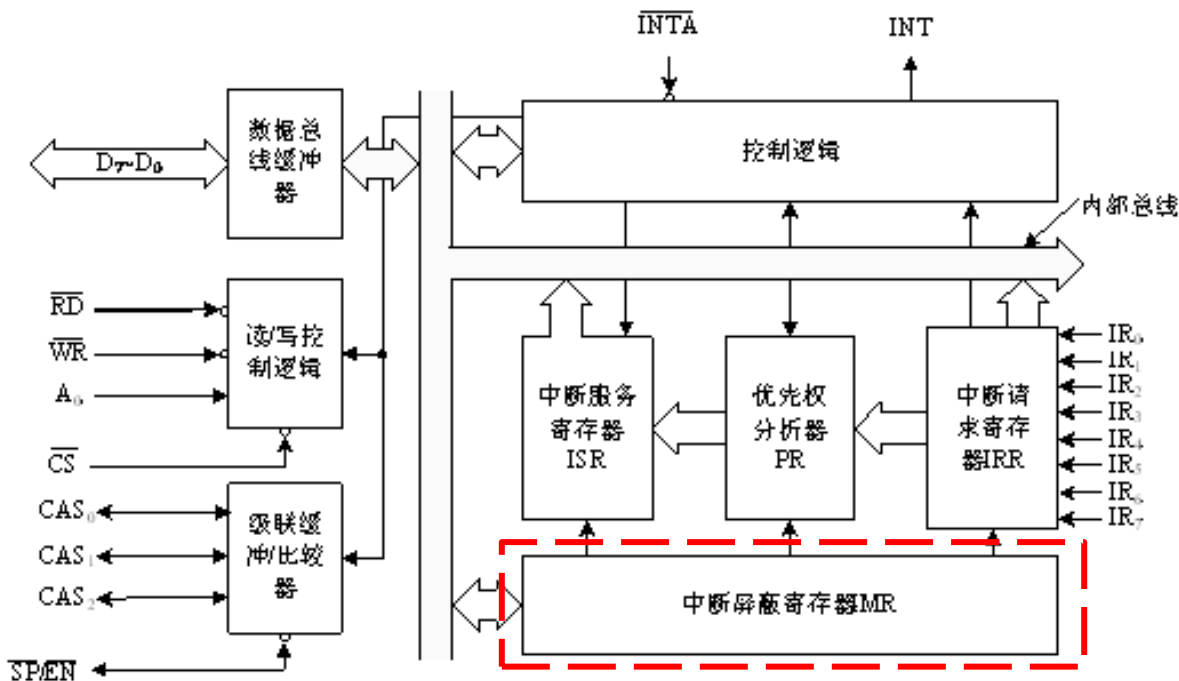


图8.11 8259A内部结构框图

(5) 中断屏蔽寄存器IMR

8位寄存器，与8259A的 $IR_7 \sim IR_0$ 相对应，用来存放对各级中断请求的屏蔽信息。

IMR寄存器某一位为0，允许IRR寄存器中相应位的中断请求进入中断优先权分析器，即开放该级中断；若为1，则此位对应的中断请求被屏蔽。

8.3 可编程中断控制器8259A

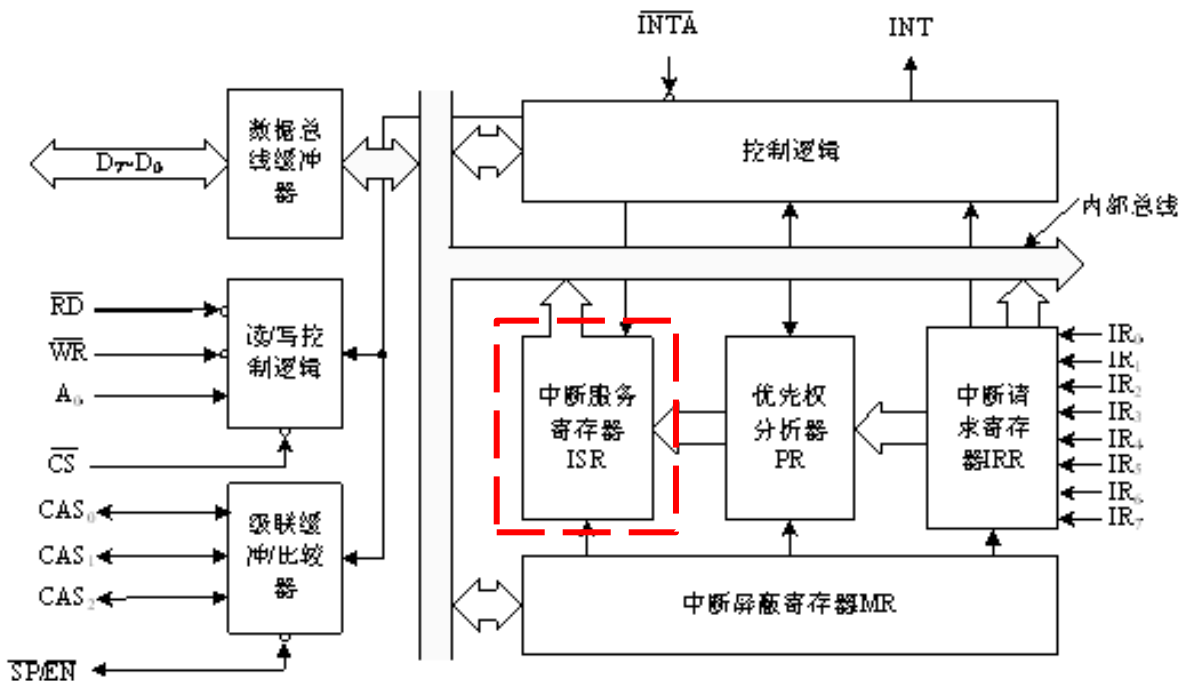


图8.11 8259A内部结构框图

(6) 中断服务寄存器ISR

8位寄存器，保存正在处理中的中断请求信号，某个IR端的中断请求被CPU响应后，当CPU发出第一个INTA信号时，ISR寄存器中的相应位置1，一直保存到该级中断处理结束为止。

只有在允许中断嵌套时，ISR中才有可能多位同时被置成1。优先级最高的位是正在服务的中断源的对应位。

8.3 可编程中断控制器8259A

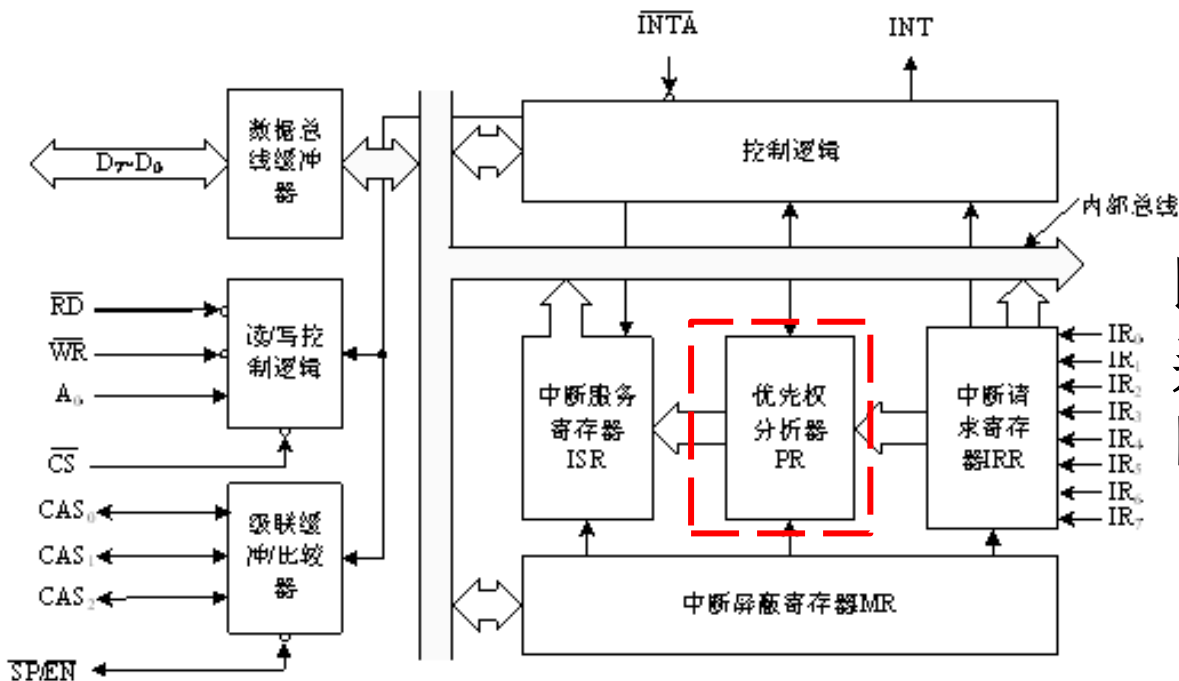


图8.11 8259A内部结构框图

(7) 优先权分析器PR

对IRR寄存器中的各中断请求信号进行优先级判别，将其中级别最高的中断请求送往CPU。

中断嵌套时，PR会将后来的中断请求与ISR中正在被服务的中断请求的优先级相比较，如果IRR中记录的中断请求的优先级高于ISR中记录的中断请求的优先级，则PR会向CPU发出中断请求信号INT，中止当前的中断服务，进行中断嵌套。

8.3 可编程中断控制器8259A

(8) 控制逻辑

8259A的内部控制器，其根据中断请求寄存器**IRR**的置位情况和中断屏蔽寄存器**IMR**的设置情况，通过优先级分析器**PR**判定优先级，向**8259A**内部及其它部件发出控制信号，并向**CPU**发出**INT**信号和接

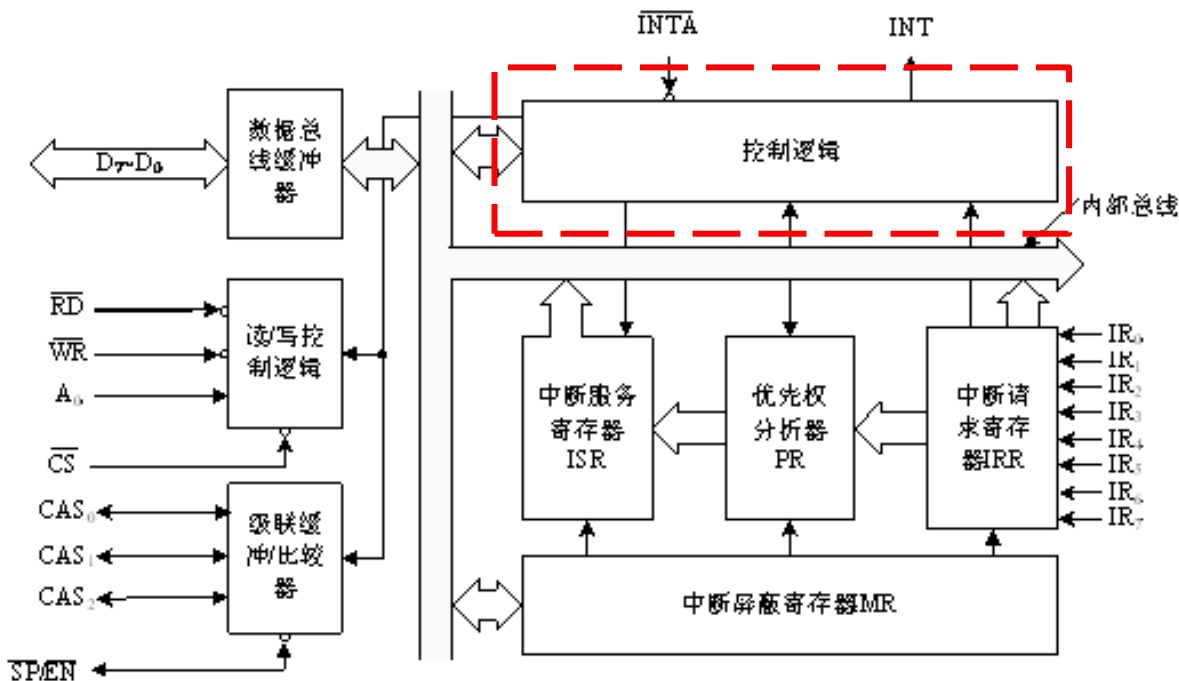


图8.11 8259A内部结构框图

收**CPU**的响应信号，使**ISR**寄存器相应位置1，同时清除**IRR**寄存器中的相应位。当**CPU**第二个**INTA**到来时，控制**8259A**送出中断类型号，使**CPU**转入中断服务程序。



8.3 可编程中断控制器8259A

2. 8259A的引脚信号

$D_7 \sim D_0$: 双向、三态数据信号。在较小系统中可直接与系统数据总线相连，在较大系统中须经总线驱动器与系统总线相连，实现和**CPU**的数据交换。

$IR_0 \sim IR_7$: 8条外设中断请求输入信号，由外设传给**8259A**。通常 **IR_0** 优先级最高， **IR_7** 优先级最低。

主从式的级联中断系统中，主片的 **$IR_0 \sim IR_7$** 分别和各从片的**INT**端相连，接收来自各从片的中断请求**INT**，由从片的中断请求输入端 **$IR_0 \sim IR_7$** 和主片未连接从片的中断请求输入端接受中断源的中断请求。

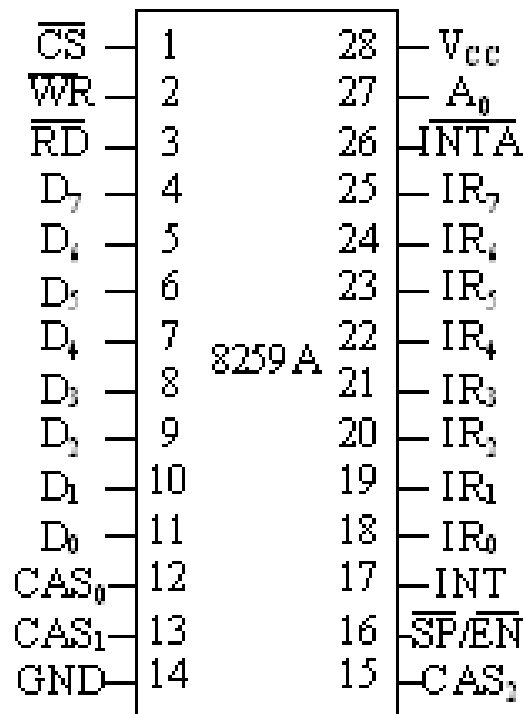


图8.12 8259A的引脚图



8.3 可编程中断控制器8259A

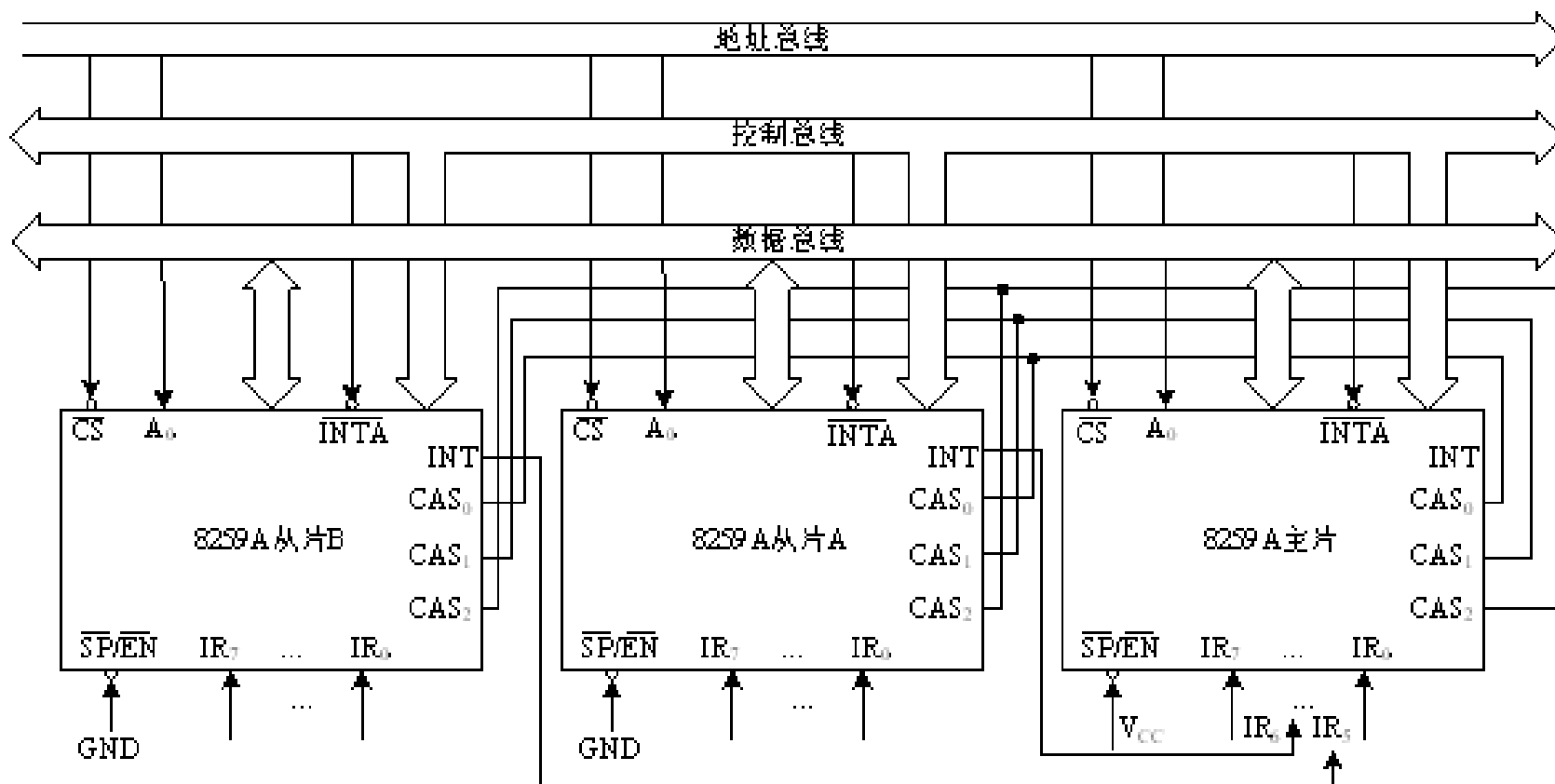


图8.13 8259A的特殊完全嵌套方式



8.3 可编程中断控制器8259A

INT: 中断请求信号，输出。与**CPU**的中断输入端**INTR**端相连（主片），向**CPU**发中断请求信号，若是从片则连接到主片的相应**IR_i**端，由从片**8259A**传给主片**8259A**。

INTA: 中断响应信号，输入。接收**CPU**送来的中断响应信号。

RD : 读信号，输入。有效时，通知**8259A**将中断类型号或某个内部寄存器的内容送给**CPU**。

WR : 写信号，输入。有效时，通知**8259A**从数据总线上接收来自**CPU**发来的命令。

CS : 片选信号，输入。有效时，**8259A**被选中。

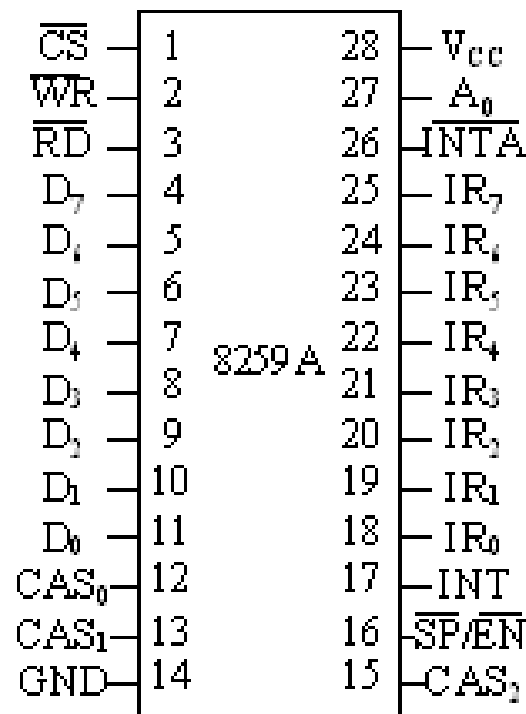


图8.12 8259A的引脚图



8.3 可编程中断控制器8259A

A_0 : 端口选择信号，输入。用于选择8259A内部的不同寄存器，通常直接接到地址总线的 A_0 。8259A内部的寄存器被安排在两个端口。

$CAS_2 \sim CAS_0$: 级联信号，为主片与从片的连线。作为主片，这3个信号是输出信号，根据它们的不同组合000~111，分别确定连在哪个 IR_i 上的从片工作。对于从片，这3个信号是输入信号，以此判别本从片是否被选中。

SP/EN : 主从片 / 缓冲允许。在不同的方式下有不同的作用，若8259A采用缓冲方式则为输出端，输出控制信号，以此来控制三态总线驱动器的开关；若采用非缓冲方式时则为输入端，为1表示芯片是主片，否则为从片。

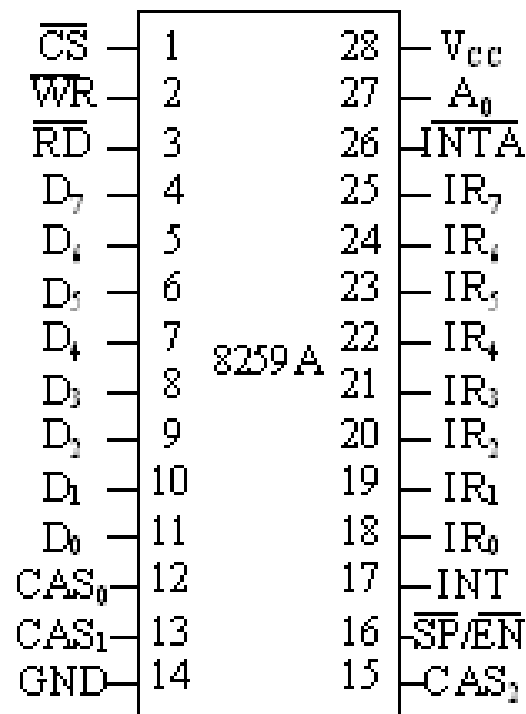


图8.12 8259A的引脚图



8.3 可编程中断控制器8259A

二、8259A的工作方式

1. 中断触发方式

决定外设以何种信号通知8259A有中断请求。

(1) 边沿触发方式

8259A将 IR_i 输入端出现的信号上升沿（正跳变）作为中断请求信号触发中断申请。

优点： IR_i 端只在上升沿申请一次中断，故该端可以一直保持高电平而不会误判为多次中断申请。

(2) 电平触发方式

8259A将 IR_i 输入端出现高电平作为中断请求信号触发中断申请

注意：当该中断请求得到响应后， IR_i 输入端必须及时撤除高电平，否则会引起不应有的第二次中断申请。

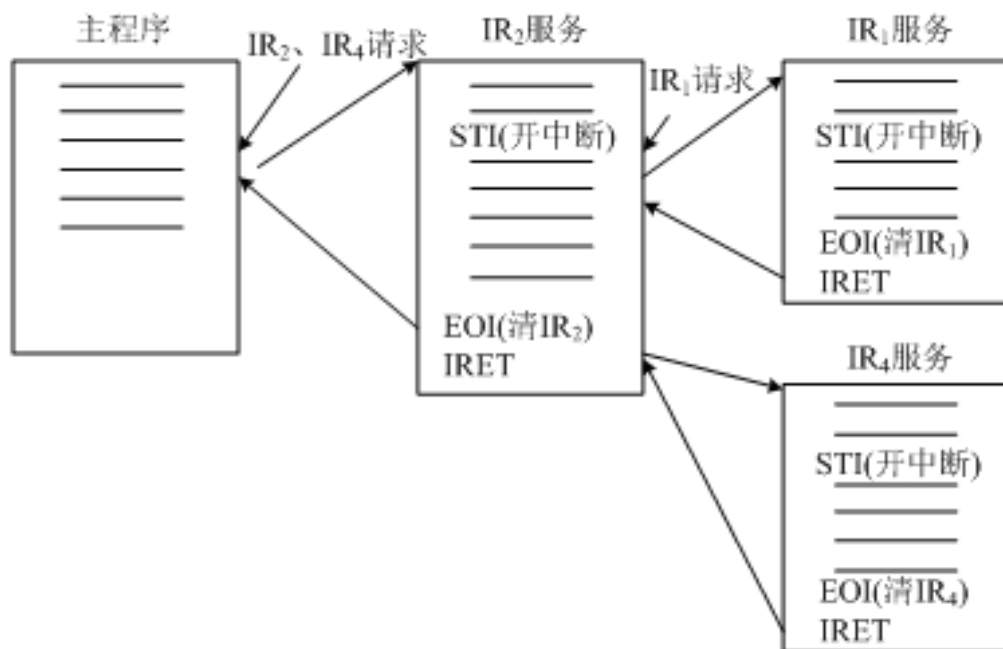


8.3 可编程中断控制器8259A

2. 中断优先级设置方式

(1) 完全嵌套方式

8259A在初始化时自动进入的一种最基本的优先级管理方式。只要不重新设置优先级别， $IR_0 \sim IR_7$ 就具有**固定不变**的优先级，默认 **IR_0 优先级最高**， IR_1 次之， IR_7 优先级最低。高优先级的中断能够中断低优先级的中断服务，实现中断嵌套。





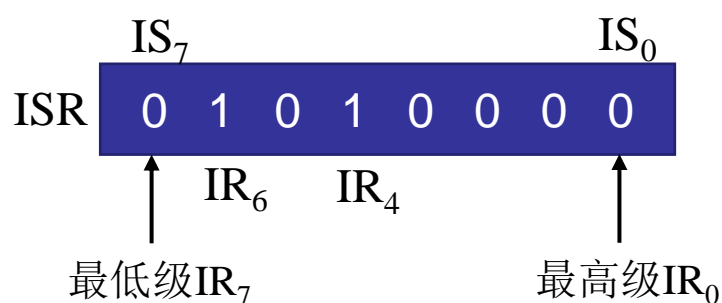
8.3 可编程中断控制器8259A

(2) 自动循环方式

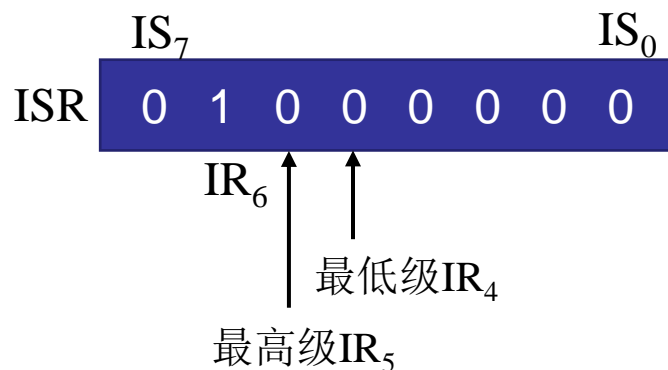
可以改变 $IR_7 \sim IR_0$ 优先级。

变化规律：初始时 IR_0 最高， IR_7 最低，但当某个中断请求被响应之后，它的优先级就**变为最低**，它的下一级中断**变为最高**优先级，优先级轮流。

例如：当 IR_4 的中断服务完后， IR_4 的优先级变为最低，而相邻的 IR_5 变为最高优先级， IR_6 次之，依次类推。



初始状态优先级

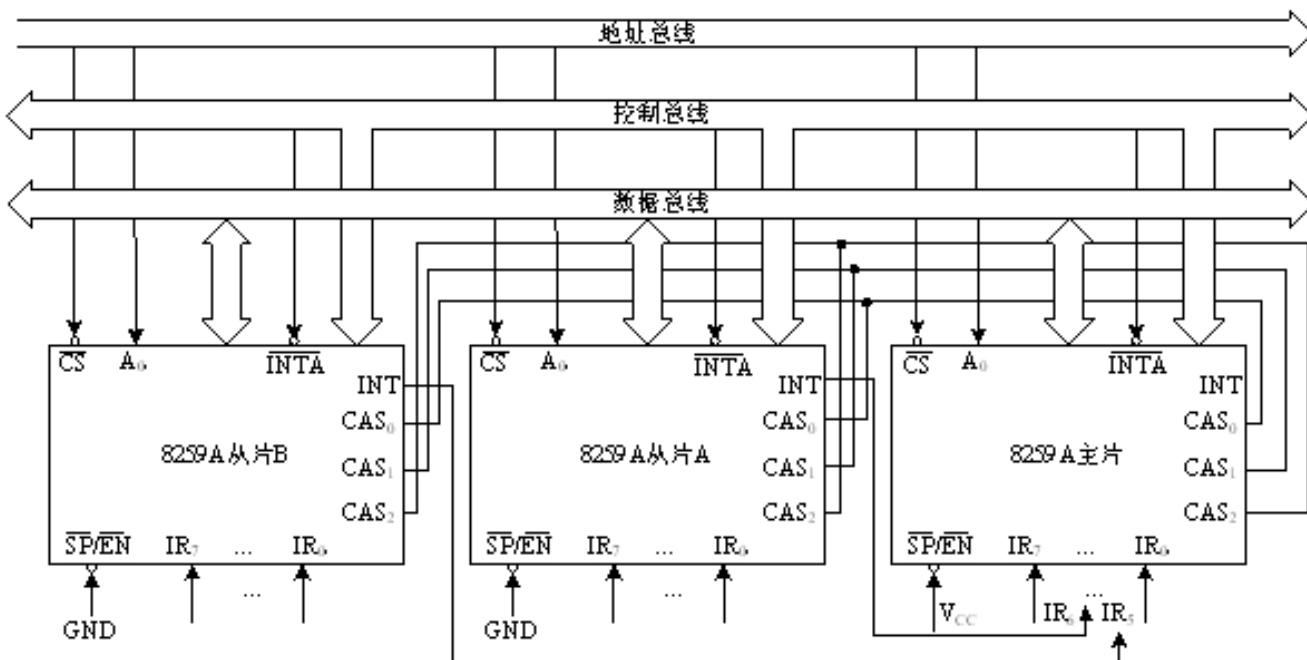


IR_4 服务完后优先级

8.3 可编程中断控制器8259A

(3) 特殊完全嵌套方式

在中断嵌套时，除了高优先级中断可以中断低优先级的中断服务，同级别的中断请求也能够予以响应，实现**同级中断的特殊嵌套**。主片工作在特殊完全嵌套方式，从片工作在完全嵌套方式。



全部 22 个中断源的优先级别排队顺序从高到低为：

- 主片IR₀、主片IR₁、主片IR₂、主片IR₃、主片IR₄
- 从片B的IR₀至IR₇
- 从片A的IR₀至IR₇
- 主片IR₇。

图8.13 8259A的特殊完全嵌套方式



8.3 可编程中断控制器8259A

(4) 特殊循环方式

通过编程指定初始最低优先级中断源，初始优先级顺序按循环方式重新排列。

注意：

以上 **8259A** 的四种优先级设置方式可以分成两类。

固定优先级：包括完全嵌套方式和特殊完全嵌套方式，在对 **8259A** 初始化编程时通过初始化命令字 **ICW₄** 进行设置；

循环优先级：包括自动循环方式和特殊循环方式，可根据实际需要通过对 **8259A** 的操作命令字 **OCW₂** 进行切换。



8.3 可编程中断控制器8259A

3. 中断屏蔽方式

对8259A外部中断源 $IR_7 \sim IR_0$ 实现屏蔽的一种中断管理方式。

(1) 普通屏蔽方式

8259A内部的中断屏蔽寄存器IMR，其每一位对应一个中断请求输入端 IR_i 。

- ✓ IMR的某位置1，对应的IR被屏蔽，不允许该IR中断传送给CPU
- ✓ 置0，允许该IR中断传送给CPU。

通过编程可设置IMR中的某位为1或为0。



8.3 可编程中断控制器8259A

(2) 特殊屏蔽方式

能开放比本身优先级别低的中断请求。

方法：对**IMR**的某位置**1**，同时也使**ISR**中的对应位清**0**。这样，虽然系统当前仍然在处理一个较高级别的中断，但由于**8259A**的屏蔽寄存器**IMR**对应于此中断的位已经被置**1**，且**ISR**中对应位被清**0**，因此外界看来好像**CPU**现在没有处理任何中断，从而实现了对低优先级中断请求的响应。

一种非正常的中断优先级排队关系，正常的应用系统中很少使用。



8.3 可编程中断控制器8259A

4. 中断结束方式

对ISR中对应位的处理。当中断结束时，必须使**ISR寄存器中对应位清0**，否则，致使比它优先级低的中断请求无法得到响应。

(1) 自动结束方式 (AEOI)

自动结束方式：利用中断响应信号的第二个负脉冲的后沿，将ISR中的中断服务标志位清除。由硬件自动完成。是最简单的中断结束方式，通常只**适用于只有一片8259A**且没有中断嵌套的情况。

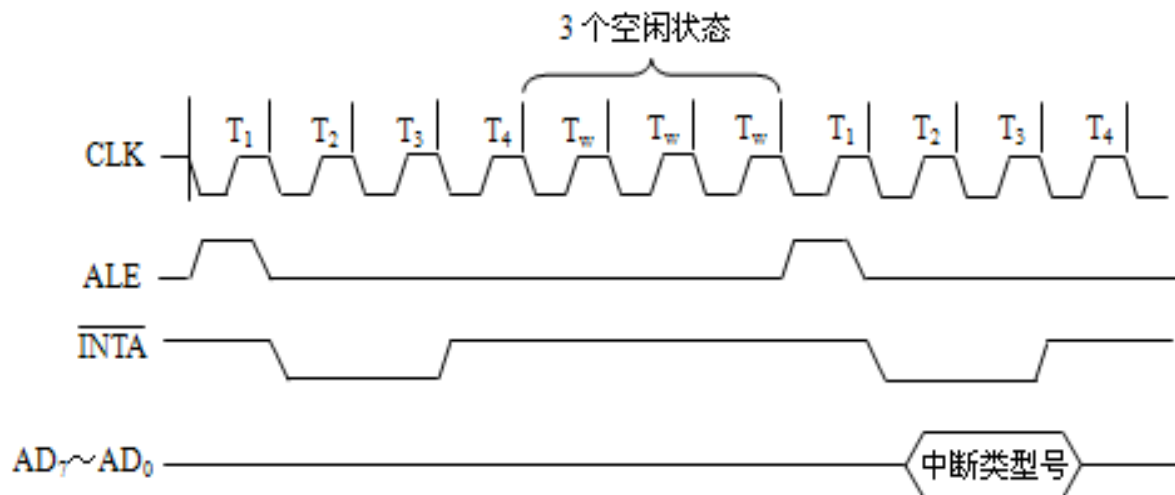


图 2.18 8086 中断响应周期时序



8.3 可编程中断控制器8259A

(2) 命令结束方式 (EOI)

分成两类:

1) 一般 EOI 方式

需要用户在中断服务程序结束之前, 向 **8259A** 发送一条一般 **EOI** 命令, 清除**ISR**中当前优先级别最高的标志位。通常, 一般 **EOI** 结束命令用于优先级全嵌套方式中。

2) 特殊 EOI 方式

需要用户在中断服务程序结束之前, 向 **8259A** 发送一条特殊 **EOI** 命令。由于在特殊 **EOI** 命令中明确指出了清除**ISR** 中的哪一位, 所以不会因嵌套结构而出现错误。这种方式仍然可以用于完全嵌套方式和特殊完全嵌套方式下的中断结束, 但更适用于嵌套结构有可能遭到破坏的中断结束。



8.3 可编程中断控制器8259A

注意：

中断的结束方式实际上包括**硬件**和**软件**两种方式。

硬件方式：由硬件完成，不需要发中断结束命令；

软件方式：需要发中断结束命令，该命令又有一般 **EOI** 和特殊 **EOI** 两种格式，是通过设置 **8259A** 的操作命令字 **OCW₂**完成的。



8.3 可编程中断控制器8259A

5. 连接系统总线的方式

(1) 非缓冲方式

适用于中小型系统中**只有一片或不多的几片8259A**情况。各片**8259A**直接和数据总线相连，无需通过总线驱动器；**8259A**的**SP/EN**端作为输入端。当系统中只有一片**8259A**时，**SP/EN**端必须**接+5V**；当系统中有多片**8259A**时，主片的**SP/EN**端接**+5V**，从片接地。

(2) 缓冲方式

多用在有多片**8259A**的大系统中。**8259A**通过总线驱动器和数据总线相连。此时，将**8259A**的**SP/EN**端和总线驱动器的允许端相连，因为**8259A**工作在缓冲方式时，会在输出状态字或中断类型码的同时，从**SP/EN**端输出一个低电平，此低电平正好可以作为总线驱动器的启动信号。



8.3 可编程中断控制器8259A

三、8259A的级联

当系统中的外部中断源数量大于8个时，就无法用一片8259A来进行管理，可采用多片8259A的级联来管理这些中断源，最多可以使用9个8259A芯片管理64个外部中断源。

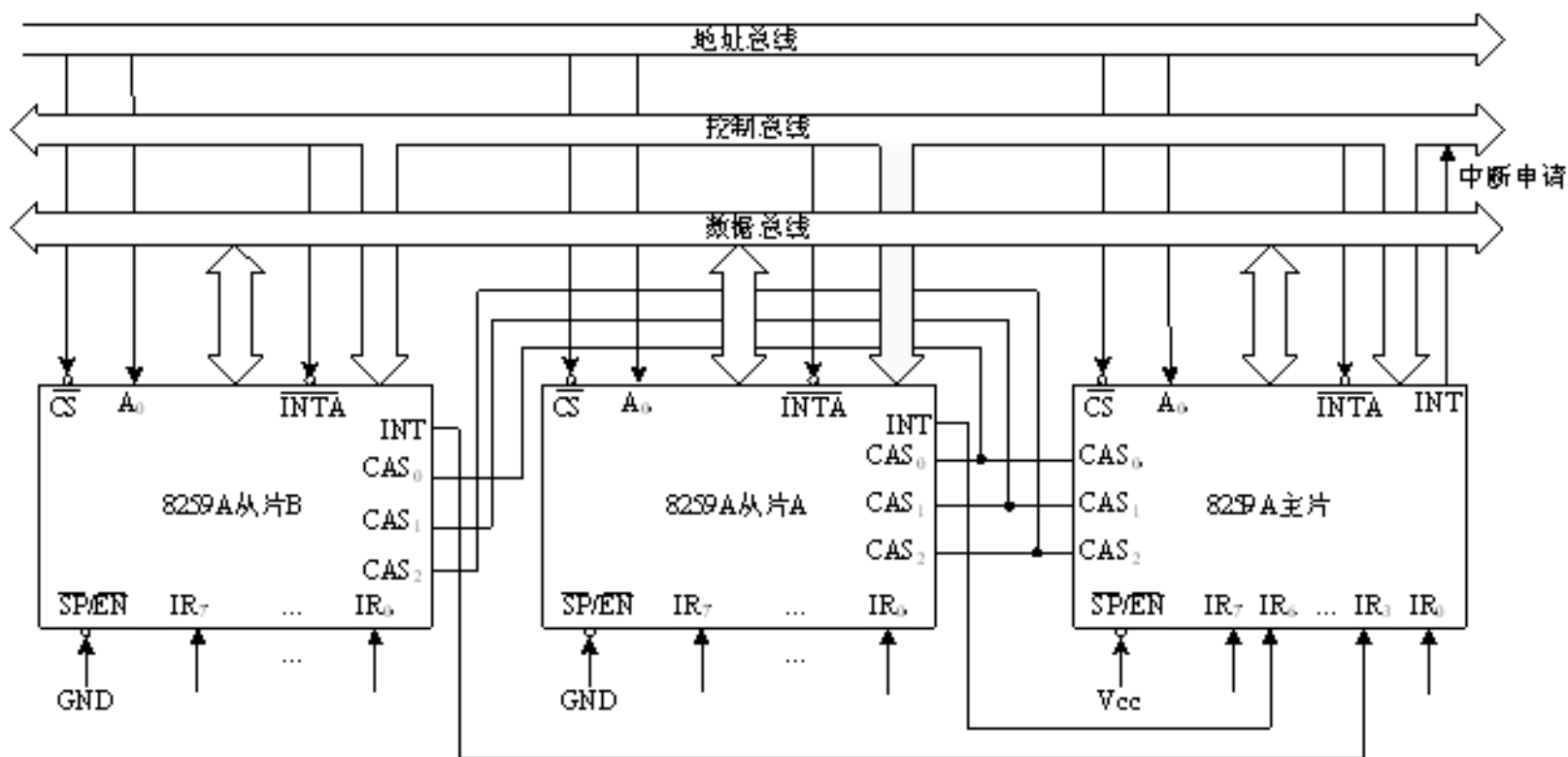


图8.14 多片8259A的级联



8.3 可编程中断控制器8259A

四、 8259A的命令字

1. 初始化命令字及其编程

初始化编程：系统加电或复位后，用初始化命令字对8259A进行初始化，以规定8259A的基本工作方式。

具体功能：

- 设定中断请求信号的触发方式，是电平触发还是边沿触发
- 设定8259A是单片还是多片级联的工作方式
- 设定8259A中断类型号基址，即 IR_0 对应的中断类型号
- 设定优先级设置方式；
- 设定中断结束方式。



8.3 可编程中断控制器8259A

8259A有4个初始化命令字ICW1~ICW4。

(1) 初始化命令字通常在计算机系统启动时由初始化程序设置，设定后一般不再改变。

(2) 初始化过程中，须严格按照规定顺序依次写入4个初始化命令字。

- ✓ ICW1写入偶地址
- ✓ ICW2~ICW4写入奇地址
- ✓ ICW1和ICW2必须设置
- ✓ ICW3和ICW4是否需要设置由ICW1的相应位决定
- ✓ 如果需要设置ICW3，则要分别对主片和从片的ICW3进行设置

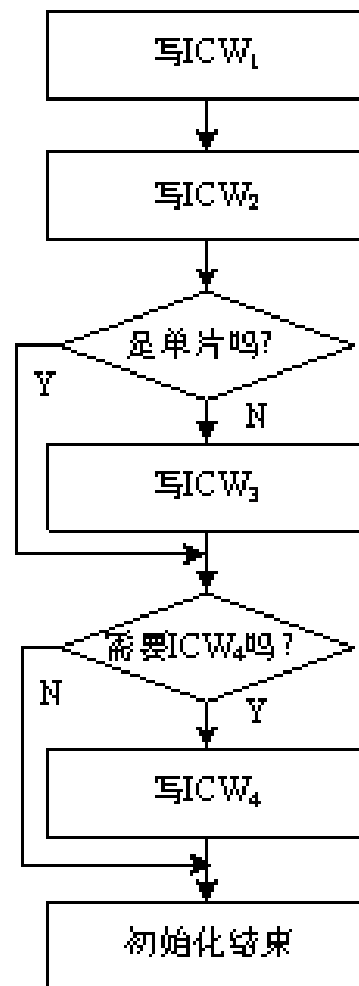


图8.15 8259A的初始化流程



8.3 可编程中断控制器8259A

(1) ICW1

ICW1命令字称为芯片控制初始化命令字，其格式如下：

	A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
ICW ₁	0	×	×	×	1	LTIM	ADI	SNGL	IC ₄

D7~D5：一般设定为“0”。

D4：特征位，必须为“1”，表明该命令字是**ICW1**。

D3：设置**8259A**的中断触发方式。**D3=1**电平触发，**D3=0**边沿触发。

D2：**8086/8088**系统中无效。

D1：设置**8259A**有无级联。**D1=1**单片工作方式，**D1=0**多片级联方式。

D0：设定是否需要初始化命令字**ICW4**。**D0=1**需要，**D0=0**不需要。对于**8086**，**D0**位须设为1。

CPU向**8259A**写入**ICW1**时，**D4**位必须为1，且必须写入偶地址，即**A0=0**。



8.3 可编程中断控制器8259A

(2) ICW2

ICW2: 设置中断类型号的基值，即8259A的 IR_0 所对应的中断类型号。

	A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
ICW ₂	1	T ₇	T ₆	T ₅	T ₄	T ₃	0	0	0
中断类型号的高5位							16位机无效		

1) D₇~D₃表示中断类型号的高5位，D₂~D₀固定为0。

2) 8259A IR_0 ~ IR_7 的中断类型号的高5位即为ICW2中的D₇~D₃，低3位由 IR_0 ~ IR_7 中的序号决定。

3) ICW2必须写入奇地址，即A₀=1。

PC/XT机8259A的8级中断类型号

D ₇ ~D ₃	D ₂ ~D ₀	中断类型号	中断输入引脚	中断源
00001	0 0 0	08H	IR_0	系统时钟
00001	0 0 1	09H	IR_1	键盘
00001	0 1 0	0AH	IR_2	保留
00001	0 1 1	0BH	IR_3	串口 2 (COM2)
00001	0 1 0	0CH	IR_4	串口 1 (COM1)
00001	1 0 1	0DH	IR_5	硬盘
00001	1 1 0	0EH	IR_6	软盘
00001	1 1 1	0FH	IR_7	打印机



8.3 可编程中断控制器8259A

【例 8.2】 已知 8259A 的 $ICW2=10H$ ，分析 8259A 的 IR_2 的中断向量在中断向量表中是如何存放的？

解析：

1) IR_2 的中断类型号的高 5 位为 $ICW2$ 的高 5 位，低 3 位为 IR_2 的序号 2。 $ICW2=10H=00010000B$ ，故 IR_2 的中断类型号 $n=00010010B=12H$ 。

2) 中断向量应存放在中断向量表中 $4n$ 开始的连续 4 个单元中，其中 $4n$ 和 $4n+1$ 存放断向量的偏移地址， $4n+2$ 和 $4n+3$ 存放中断向量的段地址。 因此， $4 \times 12H=48H$ ， $00048H$ 和 $00049H$ 存放中断向量的偏移地址， $0004AH$ 和 $0004BH$ 存放中断向量的段地址。



8.3 可编程中断控制器8259A

(3) ICW3

仅在多片8259A级联时写入，主片和从片分别写入，格式不同。

主片 ICW ₃	A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
	1	IR ₇	IR ₆	IR ₅	IR ₄	IR ₃	IR ₂	IR ₁	IR ₀
从片 ICW ₃	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
							从片标志码		

- 1) 主片ICW3: D₇~D₀分别对应其8条中断请求输入线IR₇~IR₀，若某条线上接有从片，则其对应位为“1”，否则为“0”。
- 2) 从片ICW3: D₇~D₃固定为0，D₂~D₀为从片标志码，表示该从片与主片的哪个中断请求输入线连接。
- 3) ICW3只有在ICW1的D₁=0时才需写入，且必须写入奇地址，即A₀=1。



8.3 可编程中断控制器8259A

【例 8.3】如图 8.13 所示，在这个多片 8259A 的级联系统中，假设 8259A 主片的端口地址为 20H-21H，8259A 从片 A 的端口地址为 0A0H-0A1H，8259A 从片 B 的端口地址为 0B0H-0B1H。请分别写出每片 8259A 设置 ICW3 的指令。

解析：

8259A 主片：

MOV AL, 01100000B ; 主片的 IR5 和 IR6 连有从片
OUT 21H, AL

8259A 从片 A：

MOV AL, 00000110B ; 从片 A 连至主片的 IR₆
OUT 0A1H, AL

8259A 从片 B：

MOV AL, 00000101B ; 从片 B 连至主片的 IR₅
OUT 0B1H, A



8.3 可编程中断控制器8259A

(4) ICW4

ICW4: 设定8259A的工作方式:

	A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
ICW4	1	0	0	0	SFNM	BUF	M/S	AEOI	\overline{I}/PM

D7~D5: 未定义，通常设置为0。

D4: SFNM位，**D4=1**表示8259A工作于特殊完全嵌套方式，**D4=0**表示工作于完全嵌套方式。

D3: BUF位，**D3=1**表示8259A采用缓冲方式；**D3=0**为非缓冲方式。

D2: M/S位，非缓冲方式下无意义。缓冲方式下，**D2=1**表示该8259A为主片，**D2=0**表示为从片。

D1: AEOI位，中断结束方式。**D1=1**表示采用自动结束方式；**D1=0**表示采用非自动结束方式，中断服务程序结束时需向8259A发送EOI命令以清除ISR。

D0: 为1选择8086系统

ICW4只有在**ICW1**的**D0=1**时才需要写入，且必须写入奇地址，即**A0=1**。



8.3 可编程中断控制器8259A

【例 8.4】设一片8259A工作于8086系统，8259A的I/O端口地址为20H和21H。对8259A的初始化规定为：边沿触发方式，缓冲器方式，中断结束为EOI命令方式，中断优先级管理采用全嵌套方式，IR₇中断源的类型码为0FH，写出初始化程序段，

答：

```
MOV    AL, 00010011B    ; ICW1边沿触发，单片8259A，需要ICW4
OUT    20H, AL
MOV    AL, 00001000B    ; ICW2中断类型码基数08H
OUT    21H, AL          ; 可响应的8个中断类型码为08H~0FH
MOV    AL, 00001101B    ; ICW4为全嵌套、缓冲，普通EOI结束
OUT    21H, AL
```



8.3 可编程中断控制器8259A

2. 操作命令字及其编程

对**8259A**初始化编程后，**8259A**就可接受外部中断请求信号进行工作了。工作期间，可以通过操作命令字**OCW**使它按不同的方式工作。

8259A有3个操作命令字**OCW1**、**OCW2**、**OCW3**，它们都有各自的特征位；写入时没有顺序要求；而且可以重复多次写入。

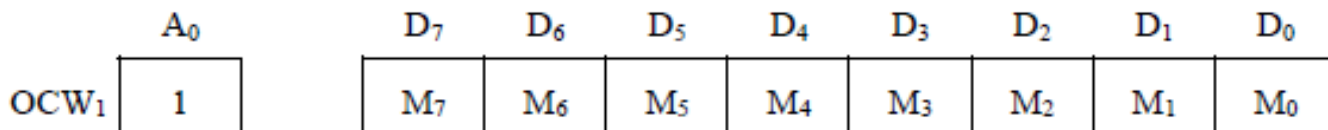
注意： **OCW1**必须写入奇地址，**OCW2**、**OCW3**必须写入偶地址



8.3 可编程中断控制器8259A

(1) OCW1

OCW1: 是中断屏蔽命令字，对中断屏蔽寄存器**IMR**的各位进行置位和复位。



M_i=1: 该位对应的**IR_i**的中断请求被屏蔽

M_i=0: 相应中断请求被允许。

初始化开始时，默认屏蔽字全“0”，所有中断源都未被屏蔽。

OCW1必须写入奇地址，即**A0=1**。**IMR**的内容也允许**CPU**随时读。

【例 8.5】假设 **8259A** 的端口地址为 **20H** 和 **21H**，请写出屏蔽 **IR₃** 和 **IR₅** 的指令序列。

答: **MOV AL, 00101000B**

OUT 21H, AL

; 写入**OCW1**，即**IMR**



8.3 可编程中断控制器8259A

【例】对8259A中断屏蔽寄存器IMR写入全0，然后读出看是否全0，若是则正确，否则转出错处理。再写入全1，作类似检查。

```
CLI
MOV     AL, 0           ; OCW1, IMR各位清0
OUT     21H, AL
IN      AL, 21H         ; AL←IMR, 不影响FLAGS
OR      AL, AL          ; IMR=0?
JNZ     D6              ; 非0, 转出错处理程序
MOV     AL, 0FFH        ; 是0, 再将全1写入IMR寄存器
OUT     21H, AL         ; 写入OCW1
IN      AL, 21H         ; 读IMR
ADD     AL, 1           ; IMR各位置1? 全1+1应为00H
JNZ     D6              ; 非全1, 转出错处理程序
        |
D6:                                ; 出错处理程序
```



8.3 可编程中断控制器8259A

(2) OCW2

OCW2: 用于设置中断优先级循环方式和中断结束方式:

A ₀		D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
OCW ₂	0	R	SL	EOI	0	0	L ₂	L ₁	L ₀
		优先级 循环	指定中 断等级	中断结 束命令	特征位		中断等级编码		

D7: R位, 优先级方式控制位。D7=0, 中断优先级固定(即IR₀最高, IR₇最低); D7=1时, 中断优先级自动循环。

D6: SL位, 决定D2~D0是否有效。D6=1, 有效; D6=0, 无效。

D5: EOI位, 中断结束命令位。D5=1, 中断服务程序结束时向8259A回送中断结束命令EOI, 以便使中断服务寄存器ISR中当前最高优先级位复位, 或由L2~L0表示的优先级位复位。

D4~D3: OCW1的特征位, D4D3=00。

D2~D0: 指定L2~L0。



8.3 可编程中断控制器8259A

表8-2 OCW₂规定的工作方式

D ₇ (R)	D ₆ (SL)	D ₅ (EOI)	工作方式	L ₂ L ₁ L ₀ 值 有无意义	说明
1	0	0	中断优先级自动循环方式	无	只规定了中断优先级方式
0	0	0	设定固定优先级		
1	1	0	特殊优先级循环方式	有	
0	1	0	无意义		
1	0	1	中断优先级自动循环方式及中断一般结束方式	无	规定了中断优先级循环方式，并执行中断返回前的中断一般结束命令，使对应ISR位清零
1	1	1	中断优先级特殊循环方式和特殊中断结束方式	有	
0	1	1	中断特殊结束方式	有	中断返回前执行中断特殊结束命令，使相应的ISR位清零
0	0	1	中断一般结束方式	无意义	中断返回前执行中断一般结束命令，使相应的ISR位清零



8.3 可编程中断控制器8259A

【例 8.6】 假设 8259A 的端口地址是 20H 和 21H。

(1) 写出一般 EOI 命令。

(2) 写出清 ISR 中 IR_5 中断服务标志位的特殊 EOI 命令。

解答： (1) **MOV AL,00100000B**
OUT 20H,AL

(2) **MOV AL,01100101B** ; L2~L0=101
OUT 20H,A



8.3 可编程中断控制器8259A

【例】发OCW2命令，结束2级中断，并置优先级为自动循环方式

MOV AL, 11100010B ; OCW2 的SEOI循环命令, L2~L0=010
OUT 20H, AL ; IS2=0结束2级中断
;优先级IR₂ 最低, IR₃为最高

【例】要求发OCW2命令，使IR₃设为最低优先级，优先权按循环方式给出。

MOV AL, 11000011B ; 将IR3置为最低优先级,
; IR4置为最高优先级
OUT 20H, AL



8.3 可编程中断控制器8259A

(3) OCW3

OCW3: 管理特殊屏蔽方式和查询方式, 控制8259A的中断请求寄存器IRR和中断服务寄存器ISR的读取。

	A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
OCW ₃	0	0	ESMM	SMM	0	1	P	RR	RIS
		无关	特殊屏蔽允许	特殊屏蔽方式	特征位		查询位	读寄存器允许	读 ISR

D6: 特殊屏蔽方式允许位。**D6=1**时**SMM**位有效。

D5: **SMM**位, 与**ESMM**组合用来设置或取消特殊屏蔽方式。当**ESMM=1**, **SMM=1**时, 设置特殊屏蔽; 当**ESMM=1**, **SMM=0**时, 取消特殊屏蔽。

D4~D3: **OCW3**的特征位, **D4D3=01**。

D2: **P**位, 中断状态查询位。若**P=1**向8259A发中断查询命令, 紧接着用**IN**指令读取中断查询字。

D1: **RR**位, 控制对寄存器的读取。**D1=1**, 允许读取**D0**所指定的寄存器。

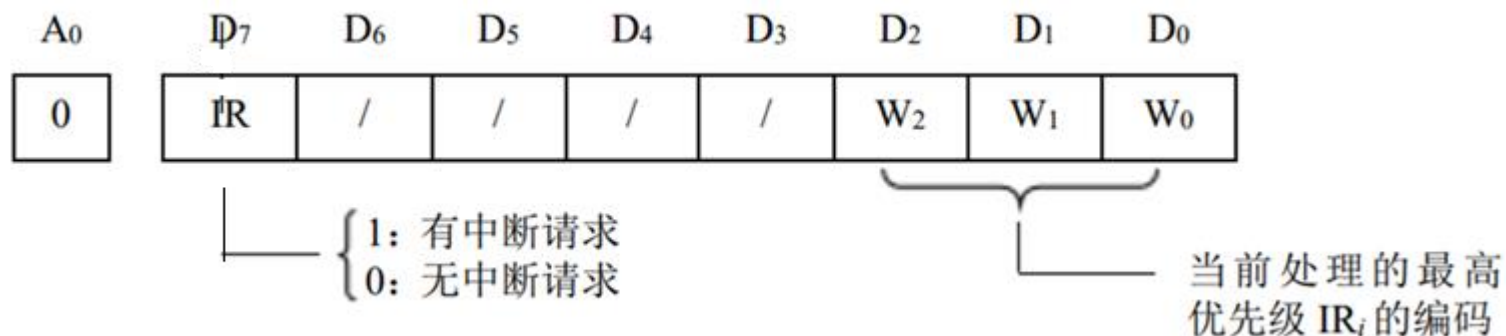
D0: **RIS**位, 确定读取**ISR**还是**IRR**寄存器。在**D1=1**时, 若**D0=1**, 读取**ISR**寄存器; 若**D0=0**, 则读取**IRR**寄存器。

OCW3必须写入偶地址, 即**A0=0**, 且特征位**D4D3=01**, 以区别于**OCW2**。



8.3 可编程中断控制器8259A

8259A 的状态寄存器（中断查询字）





8.3 可编程中断控制器8259A

【例】编写读取中断屏蔽字、中断查询字及中断服务寄存器的程序段。

(1) 读取中断屏蔽字的程序段：

IN AL, 21H ; 从奇地址口读IMR, 获得中断屏蔽字

(2) 读取中断查询字的程序段：

MOV AL, 00001100B ; OCW₃, P=1

OUT 20H, AL ; OCW₃写入偶地址口

IN AL, 20H ; AL←中断查询字

; 若AL=10000110B, 表示有中断

; 请求, 且IR₆上的优先级最高

(3) 读取中断服务寄存器ISR的程序段：

MOV AL, 00001011B ; OCW₃, RR-RIS=11,

; 下次读ISR

OUT 20H, AL ; OCW₃写入偶地址口

IN AL, 20H ; AL←ISR内容



8.3 可编程中断控制器8259A

3. 8259A 内部寄存器的访问方式

8259A 内部共有 11 个寄存器，它们分别是 IRR、IMR、ISR、ICW1~ICW4、OCW1~OCW3 和状态寄存器。这些寄存器均通过 8259A 的一条地址线 A_0 进行访问，即这 11 个寄存器共用 2 个端口地址。

表8-3 8259A的读/写操作

\overline{CS}	\overline{RD}	\overline{WR}	A_0	D_4	D_3	读写操作	指令
0	1	0	0	1	×	CPU写入ICW ₁	OUT
0	1	0	1	×	×	CPU写入ICW ₂ 、ICW ₃ 、ICW ₄ 、OCW ₁	
0	1	0	0	0	0	CPU写入OCW ₂	
0	1	0	0	0	1	CPU写入OCW ₃	
0	0	1	0			CPU读取IRR/ISR、查询字	IN
0	0	1	1			CPU读取IMR	
1	×	×	×			高阻	
×	1	1	×			高阻	



8.4 8259A在微机中的编程应用

一、8259A在IBM PC/AT系统中的应用

【例 8.7】 8259A 在 IBMPC/AT 系统中的硬件连接及初始化编程。

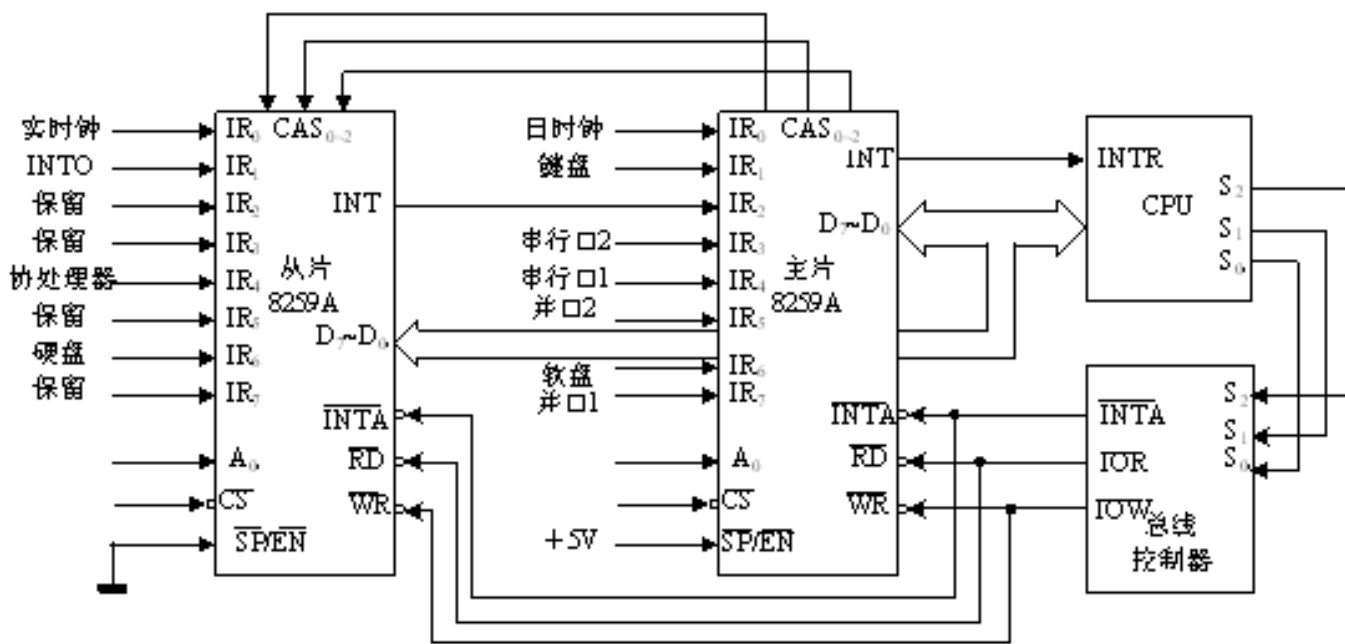


图8.16 IBM PC/AT系统中8259A的硬件连接图

- IBM PC/AT系统共有两片8259A芯片，从片的INT引脚直接连到主片的IR₂引脚，可接收最多15级中断。
- 端口地址主片仍然为20H和21H，从片为0A0H和0A1H。
- 主、从片均为边沿触发，均采用完全嵌套方式，优先级依次为0级，1级，8~15级，3~7级。系统采用非缓冲方式，主片的中断类型为08H~0FH，从片的中断类型为70H~77H。



8.4 8259A在微机中的编程应用

根据系统要求和硬件连接图，8259A的主片和从片初始化程序段如下：

；初始化主片

INTA00 EQU 20H

INTA01 EQU 21H

...

MOV AL, 00010001B ;设置ICW₁为边沿触发，多片 8259A，需要ICW₄

OUT INTA00, AL

MOV AL, 00001000B ;设置ICW₂中断类型码基数为08H

OUT INTA01, AL ;可响应的8个中断类型码为08H~0FH

MOV AL, 00000100B ;主片IR₂引脚上接从片

OUT INTA01, AL

MOV AL, 00000001B ;设置ICW₄为8086/8088模式，普通EOI，非缓冲方
式，完全嵌套方式

OUT INTA01, AL

...



8.4 8259A在微机中的编程应用

;初始化从片

INTB00 EQU 0A0H

INTB01 EQU 0A1H

...

MOV AL, 00010001B ;设置ICW1为边沿触发, 多片 8259A, 需要ICW4

OUT INTB00, AL

MOV AL, 01110000B ;设置ICW2中断类型码基数为70H

OUT INTB01, AL ;可响应的8个中断类型码为70H~77H。

MOV AL, 00000010B ;从片接主片的IR2引脚

OUT INTB01, AL

MOV AL, 00000001B ;设置ICW4为8086/8088模式, 普通EOI, 非缓冲方
;式, 完全嵌套方式

OUT INTB01, AL

...



8.4 8259A在微机中的编程应用

二、8259A的编程

8259A 的编程是由主程序和中断服务程序两个模块构成：

主程序：完成 8259A 的初始化， 为中断服务程序做一些必要的准备工作。其中**设置中断向量是必须完成的**

中断服务程序：完成中断源请求的中断服务操作。
所谓设置中断向量，就是将中断服务程序的入口地址，即中断向量写入中断向量表中。



8.4 8259A在微机中的编程应用

【例 8.8】 设 8259A 的 ICW2=08H，现使用 IR₇ 引入硬件中断请求，8259A 的端口地址为 20H 和 21H，8259A 的中断服务程序入口地址为 IRQ7，写出设置中断向量的程序代码。

分析：由 ICW2=08H，可知 IR7 的中断类型号 n=0FH，则中断向量应写入 4n=3CH 开始的连续 4 个字节内存单元。

解答：

```
MOV AX, 0
MOV ES, AX
MOV BX, 3CH
MOV AX, OFFSET IRQ7
MOV ES:WORD PTR[BX], AX ;置入中断服务程序入口地址的偏移地址
MOV AX, SEG IRQ7
MOV ES:WORD PTR[BX+2], AX ;置入中断服务程序入口地址的段地址
|
IRQ7: ;中断服务程序
|
IRET
```



8.4 8259A在微机中的编程应用

□ DOS功能调用专门提供了在中断向量表中设置和取得中断向量的手段，功能号为25H和35H。

设置中断向量

入口参数 DS: DX=中断向量（中断服务程序入口地址）

AL=中断类型号N

AH=25H（DOS功能号）

执行 INT 21H指令

结果 将AL中指定的中断类型号为N的中断向量（DS: DX）置入中断向量表中。

取得中断向量

入口参数 AL=中断类型号N

AH=35H（DOS功能号）

执行 INT 21H 指令

结果 N号中断的中断向量从中断向量表中取到ES: BX中



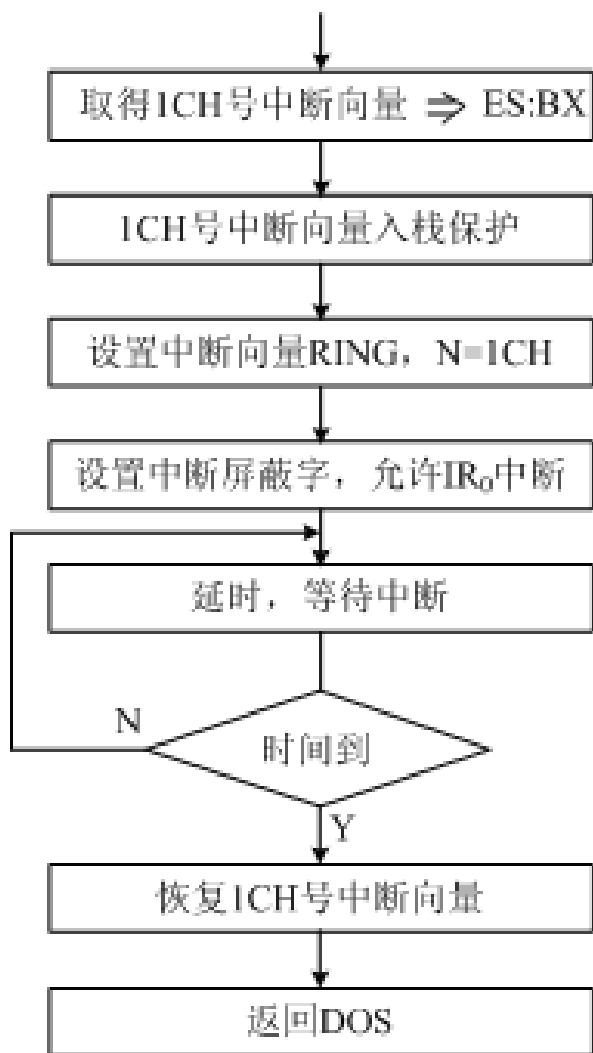
8.4 8259A在微机中的编程应用

【例】 编写中断处理程序，要求主程序运行时，每隔10秒钟响铃一次，并在CRT上显示一行信息“The bell is ring.”，运行一定时间后停止运行。

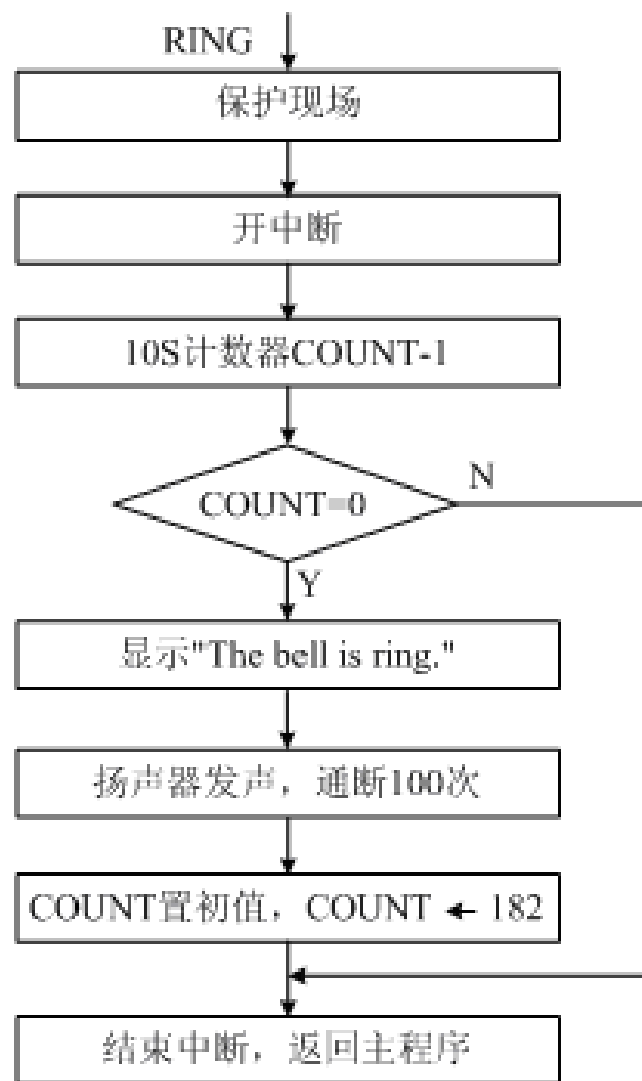
- ❑ 在PC机中，每隔55ms执行一次“INT 8H”中断服务程序。PC机每产生一次中断都要调用一次N=1CH的中断处理程序，该处理程序只有一条IRET指令，用户若有周期性的定时工作要做，就可以利用它，用自己设计的程序替代原有的1CH中断处理程序。
- ❑ 用户编写新的中断处理主程序时，要先保存当前中断向量（1CH号中断），再设置新的中断向量（完成响铃、显示工作），结束时恢复原中断向量。



8.4 8259A在微机中的编程应用



(a) 中断处理主程序



(b) 响铃显示中断服务程序



8.4 8259A在微机中的编程应用

```
DATA    SEGMENT                ; 数据段
COUNT  DW  1                  ; 10s计数器首次值置为1
MESS     DB  'The bell is ring.', 0AH, 0DH, '$'
DATA     ENDS
```

;主程序

```
CODE     SEGMENT
        ASSUME  CS:CODE, DS:DATA
```

START:

```
        MOV     AX, DATA        ; 设置数据段
        MOV     DS, AX
        MOV     AL, 1CH          ; 指向1CH号中断向量
        MOV     AH, 35H          ; 取1CH号中断向量
        INT     21H              ; ES: BX←中断向量
        PUSH    ES
        PUSH    BX               ; 1CH中断向量入栈保护
        PUSH    DS              ; 保护DS
```



8.4 8259A在微机中的编程应用

```
MOV    DX, OFFSET RING    ; DS: DX←RING的
MOV    AX, SEG RING        ; 段基址和偏移量
MOV    DS, AX
MOV    AL, 1CH             ; RING的中断类型号
MOV    AH, 25H
INT     21H                ; 设置RING的中断向量
POP     DS                 ; 恢复DS
IN      AL, 21H            ; 读IMR
AND     AL, 0FEH          ; 使IMR的D0位=0,
OUT     21H, AL           ; 以允许IR0 中断
STI     ; 开中断, 等待定时中断
MOV     DI, 20000         ; 延时一定时间(自定义)
DELY:
MOV     SI, 30000
```



8.4 8259A在微机中的编程应用

DELY1:

```
DEC     SI
JNZ     DELY1
DEC     DI
JNZ     DELY      ; 时间没到，继续循环
POP     DX        ; 时间到，原中断向量
           ; 弹出到DS: DX中

POP     DS
MOV     AL, 1CH    ; 中断类型号
MOV     AH, 25H    ; 恢复原中断向量
INT     21H
MOV     AX, 4C00H  ; 返回DOS
INT     21H        ; 主程序结束
```



8.4 8259A在微机中的编程应用

； 中断服务程序RING

RING PROC NEAR

PUSH DS ; 保护现场

PUSH AX

PUSH CX

PUSH DX

MOV AX, DATA ; 设置数据段

MOV DS, AX

STI ; 开中断，允许中断嵌套

DEC COUNT ; 10s计数器COUNT减1

JNZ EXIT ; 非0，10s时间未到，则退出

MOV DX, OFFSET MESS

； 是0，10s到，显示提示信息

MOV AH, 09H

INT 21H



8.4 8259A在微机中的编程应用

```
MOV    DX, 100          ; 扬声器通断100次
IN      AL, 61H          ; 扬声器发声程序
AND     AL, 0FCH         ; 使8255A的PB1PB0=00
```

SOUND:

```
XOR     AL, 02H          ; 61H口的D1位 (PB1)
OUT     61H, AL          ; 由1→0, 由0→1
MOV     CX, 140H         ; 延时使方波有一定宽度
```

WAIT1:

```
LOOP    WAIT1
DEC     DX                ; 通断100次了吗?
JNE     SOUND             ; 没有, 继续发声
MOV     COUNT, 182        ; 已满100次,
                          ; 重置COUNT←182
```

EXIT:

```
MOV     AL, 20H           ; 用EOI命令结束中断
OUT     20H, AL
```



8.4 8259A在微机中的编程应用

POP DX ; 恢复现场

POP CX

POP AX

POP DS

IRET ; 返回主程序, 等待下次中断

RING ENDP ; 中断服务程序结束

CODE ENDS ; 代码段结束

END START ; 程序将从START执行