

考点一：了解外设为什么不能直接与总线连接。接口起什么作用？了解接口完成的功能。

### 1.1 输入/输出接口的作用

### 1.2. CPU 与外设交换的信息

### 1.3 接口完成的功能

考点二：掌握接口内部基本组成，端口的概念，明确端口与接口的关系；掌握端口的两种编制方式及各自特点。

### 2.1 接口内部基本组成

### 2.2 I/O 端口的概念

### 2.3 端口的两种编制方式

### 2.4 三态门接口与锁存器接口的构成

考点三：掌握常用的输入输出方式，理解其各自特点及应用场合，要求深入理解条件传送方式和中断传送方式的区别。

### 3.1 基本输入输出方式

考点四：深入理解中断的概念，掌握中断全过程包含的步骤。掌握 8086CPU 响应中断的条件及响应过程，理解中断服务程序的一般构成。

### 4.1 中断的概念

### 4.2 8086 中断处理过程

考点五：了解 8086 系统中断源的分类，掌握 8086CPU 对他们的管理方式。

### 5.1 8086 中断系统

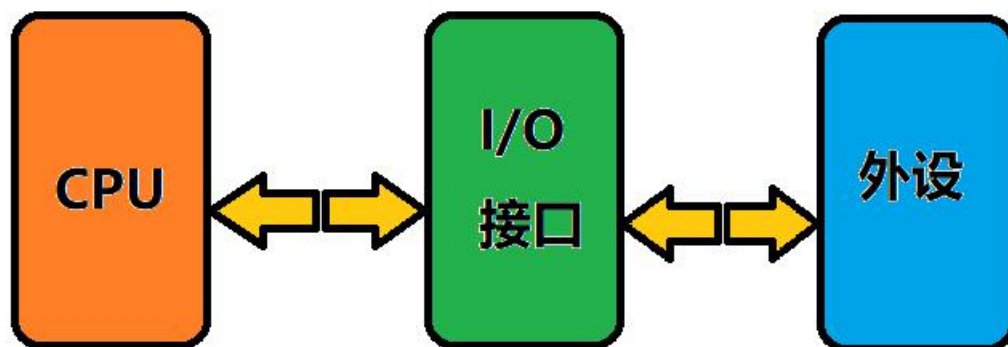
考点六：了解中断控制器 8259A 的基本构成与工作原理。

### 6.1 中断控制器 8259A 的内部结构及引脚

## 6.2 8259A 的中断工作方式

### 1.1 输入/输出接口的作用

接口是主机和外设之间交换信息的桥梁, 介于主机与外设之间的一种缓冲电路称为 I/O 接口。



那么, 主机与外设交换信息时为什么要通过接口电路呢? 这主要有以下三个原因。

#### ①总线隔离作用

数据总线是外设传送信息的公共线路. 任何外设都不允许长期占用数据总线, 而仅允许被选中的外设在读/写总线周期中享用数据总线。

因此, 大多数外设不能直接和 CPU 的数据总线相连, 要借助与接口电路使外设与总线隔离, 起缓冲、暂存数据的作用, 使主机和外设协调一致的工作, 当输入设备与总线连接时, 由接口中的缓冲电路实现隔离; 当输出设备与总线连接时, 由接口中的锁存电路实现隔离。

#### ②速度匹配作用

大多数外设输入输出信息的速度远远低于微处理器, 为同步外设与主机的工作. 在输入输出控制中, 常需要接口电路提供外设的工作状态给微处理器, 同时记忆

主机下达给外设的命令,从而使主机与外设之间协调一致地工作。

也就是说,接口电路可以提供握手信息给微处理器和外设,协调主机和外设间数据传送速度不匹配的矛盾。

对于接口提供的外设状态信号有输出设备的“忙”信号(BUSY)和输入设备的“准备好”信号(READY)。

### ③信号变换作用

外设的信息格式与微处理器不一致时,需要接口电路进行信息的变换。

从本质上说,微处理器的信息格式是并行的数字信号,而设由于其功能的多样性,信息格式也是多种多样的。这就需要进行:

- 电平变换
- 并串变换
- 数模变换

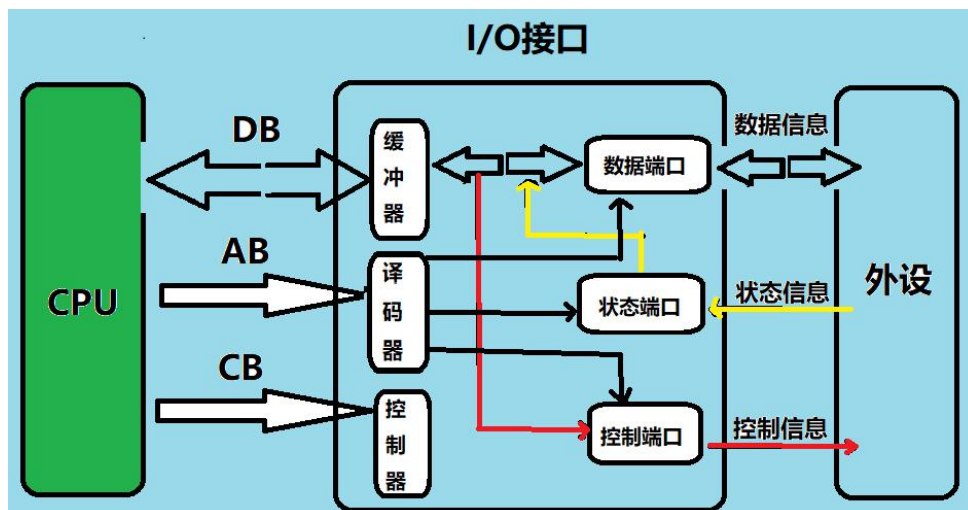
### 1.2. CPU 与外设交换的信息

主机与 IO 设备之间交换的信息可分为数据信息,状态信息和控制信息三种。

(1) 数据信息:它是 CPU 与外设之间传送的主要信息,可分为数字量、模拟量和开关量三种形式。

(2) 状态信息:是外设通过接口送往 CPU 的信息,作为外设与 CPU 之间交换数据的联络信号,反映了当前外设所处的工作状态。

(3) 控制信息:是 CPU 通过接口传送给外设的信息,用来设置外设(包括接口)的工作方式,控制外设的工作等。



### 1.3 接口完成的功能

数据的寄存和缓冲、信号电平转换功能、信号格式转换功能。、设备选择功能、对外设的控制与检测功能、产生中断请求及 DMA（直接内存操作）请求功能、可编程功能。

### 2.1 接口内部基本组成

I/O 接口电路内 部可以归结为 3 可编程的寄存器

(1) 数据寄存器(数据端口): 保存处理器与外设之间交换的数据，主要起数据缓冲作用。

(2) 状态寄存器(状态端口): 保存外设或其接口电路当前的工作状态信息。

(3) 控制寄存器(控制端口): 保存处理器控制接口电路和外设操作的有关控制信息。

(4) 数据缓冲器, 接口电路输入输出的数据、控制、状态信息都通过此缓冲电路传送，与系统的数据总线相连；

(5) 地址译码电路, 根据 CPU 的地址信息， 由片选信号是否有效判断 CPU 是否选中接口芯片，再寻址出片内的某个端口；

(6) 控制逻辑电路,产生接口的内部控制信号和对外控制信号,以实现处理器和外设间相互协调工作。

## 2.2 I/O 端口的概念

1、端口 (Port):是接口中可以进行寻址读/写的寄存器,简称为口。每个接口电路中都包含一组寄存器,CPU 与外设进行信息交换时,各类信息在接口中存入不同的寄存器,一般称这些寄存器为 I/O 端口。

2、一个接口往往含有几个端口,CPU 通过输入输出指令向这些端口取存信息.端口主要有三类:一类为状态口,一类为命令口(或控制口),一类是数据口。

3、CPU 通过输入指令从状态口获取外设的状态信息,通过输出指令从命令口发出控制命令,控制外设的工作。通过输入输出指令可以从数据口与外设交换数据,因此说,计算机主机与外设之间交换信息都是通过接口中的端口来实现的。

## 2.3. I/O 端口的编址方式

CPU 通过对端口分配地址识别它们,称为编址。

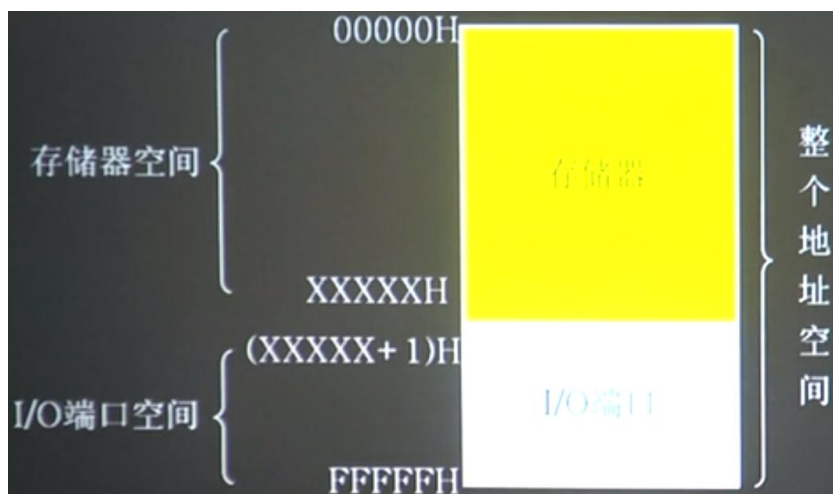
CPU 给每个端口分配一个地址,称为端口地址或端口号。

微机系统中,I/O 端口的编址方式分为统一编址和独立编址两大类。在 Intel 80x86 系列微机中,采用独立编址方式。

### ①I/O 端口的统一编址方式

统一编址又称存储器映象编址,就是将 I/O 端口看成是存储器空间的一个组成部分按照存储器单元的编址方法统一编排地址号,每个 I/O 端口占用一个地址。这样,CPU 对 I/O 端口的输入/输出操作如同对存储器单元的读/写操作一样,对存储器的各种寻址方式也同样适用于 I/O 端口。因此,它没有专门的输入输出指令。

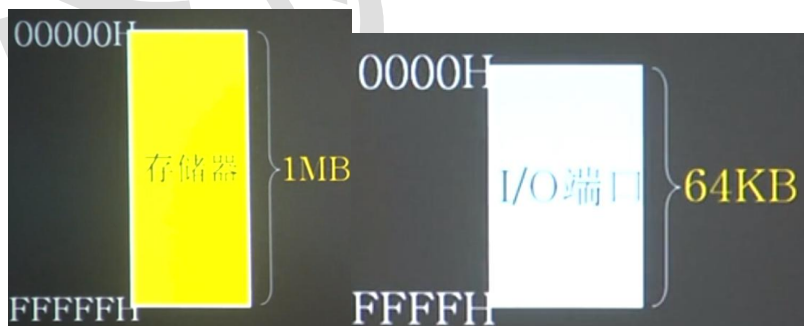
缺点:



I/O 端口空间会占用存储器空间，I/O 端口空间越大，存储器空间越小。

## ②I/O 端口的独立编址方式

在独立编址方式下，I/O 地址空间完全独立于存储器空间。在 I/O 地址空间中，每个端口有一个唯一的端口地址，CPU 有专用的 I/O 指令，用于 CPU 与 I/O 端口之间的数据传输。Intel 80x86 系列 CPU 中设有 IN、OUT 指令作为专用的 I/O 指令和寻址方式。在直接寻址方式下，CPU 只利用地址总线的低 8 位输出地址信息，所以只能寻址 I/O 地址空间的 00H~FFH 地址，而在间接寻址方式下，CPU 利用地址总线的低 16 位输出地址信息(用 DX 做间址寄存器)，可以寻址的 I/O 空间多达 64K 个端口。



## 1. 输入指令

### (1) 直接寻址的输入指令

指令格式及操作:

IN AL/AX, PORT

该指令把 8 位或 16 位的数据直接由输入端口 PORT (地址为 0~255) 输入到 AL 或 AX 寄存器中。

例如:

IN AL, 0FFH ; 从字节端口 0FFH 读一个字节到 AL 中

### (2) 间接寻址的输入指令

指令格式及操作:

IN AL/AX, DX

该指令把 8 位或 16 位的数据由 DX 寄存器指定的端口地址输入到 AL 或 AX 寄存器中。

例如:

MOV DX, 300H

IN AX, DX ; 从字节端口 300H 读一个字节到 AX 中

## 2. 输出指令

### (1) 直接寻址的输出指令

指令格式及操作: OUT PORT, AL/AX

该指令把 AL (8 位) 或 AX (16 位) 的数据直接输出到 PORT 指定的输出端口地址 (0~255) 。

例如:

OUT 80H, AL; 把 AL 的内容写到端口 80H 中

## (2) 间接寻址的输出指令

指令格式及操作:

OUT DX,AL/AX

该指令把 AL (8 位) 或 AX (16 位) 的数据送到由 DX 寄存器指定的输出端口。

例如:

MOV DX,310H

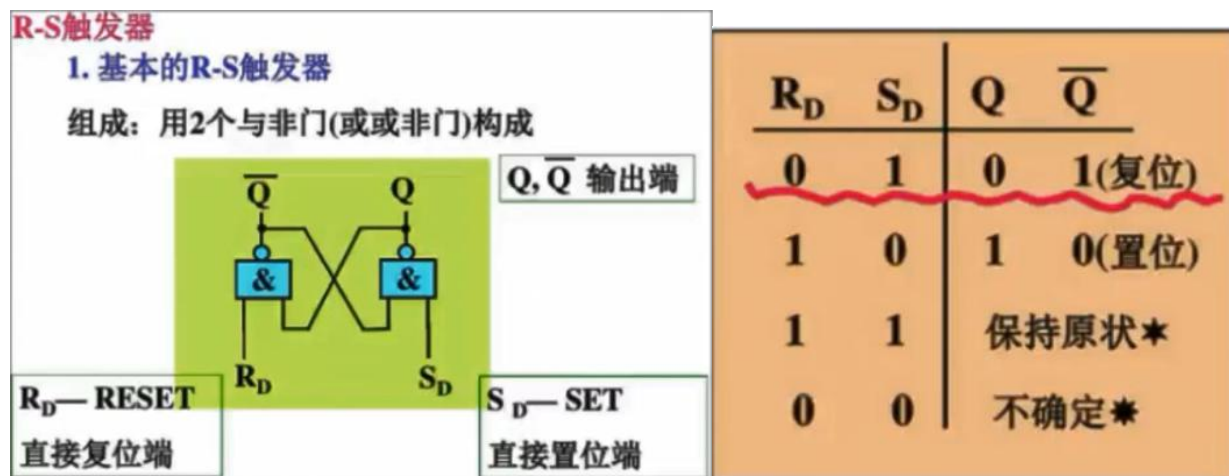
OUT DX,AL; 把 AL 的内容写到端口 310H 中

## 2.4 三态门接口与锁存器接口的构成

- 1、三态门亦称“三态输出门”、“三态门输出电路”。是一种重要的总线接口电路。具有高电平、低电平和高阻抗三种输出状态的门电路。
- 2、三态门作用，主要是用于总线的连接，因为总线只允许同时只有一个使用者。通常在数据总线上接有多个器件，如器件没有选通的话它就处于高阻态，相当于没有接在总线上，不影响其它器件的工作。
- 3、类型:主要有晶体管-晶体管逻辑(TTL)三态门电路和互补型金属-氧化物半导体(CMOS)三态门电路，两种电路都是在普通门电路的基础上附加控制电路而构成的。多用于各种集成电路中。
- 4、高阻态指的是电路的一种输出状态，既不是高电平也不是低电平高阻态相当于隔断状态(电阻很大，相当于开路)。
- 5、锁存器:是一种对脉冲电平敏感的存储单元电路。
- 6、可以在特定输入脉冲电平作用下改变状态。
- 7、输出端的状态不会随输入端的状态变化而变化，仅在有关锁存信号时输入的状态被保存到输出，直到下一个锁存信号到来时才改变。常用的锁存器逻辑电路如



D 触发器等。



$R_p=0$  同时  $S_p=1$  时,  $Q=0$ 。故  $R_p$  称为复位端,或称为清 0 端

$S_p=0$  同时  $R_p=1$  时,  $Q=1$ 。故  $S_p$  称为置位端,或称为置 1 端

### 3.1 基本输入输出方式

这里所说的数据传送方法是指 CPU 通过接口与外设之间数据传送的方式,一般有三种:程序控制方式(无条件传送方式、有条件传送方式)、中断控制方式和直接存储器存取控制方式(DMA 控制方式)。

#### (1) 程序控制方式

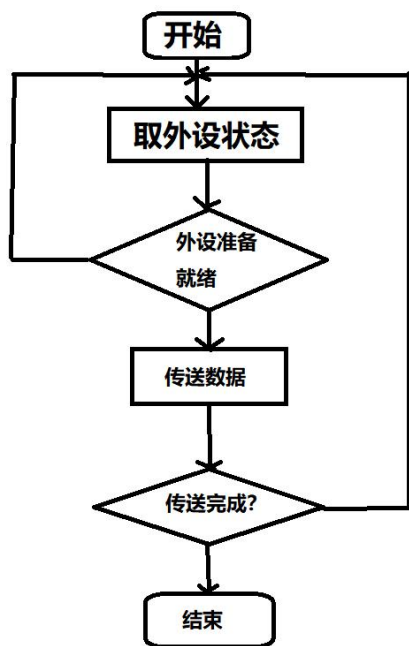
该方式完全通过执行程序来控制 CPU 与外设之间的数据交换, I/O 指令序列事先就排在程序中所需要的位置。这种传送方式又分为无条件传送和有条件传送。

##### ① 无条件传送

程序中 I/O 指令的执行不需要事先测试外设的状态,而是直接执行,认为外设已经准备就绪。这种传送方式只适用于对简单外设的操作,这些外设始终处于就绪状态,典型的输入是开关,典型的输出是 LED 数码管。

## ②有条件传送

也称查询方式传送，程序中 I/O 指令的执行需要事先测试外设的状态，待外设准备就绪后，执行 I/O 指令进行数据传送，否则循环测试等待。例如，CPU 向打印机送打印字符，就需要测试打印接口电路的 BUSY（0 空闲）信号。



## (2) 中断控制方式

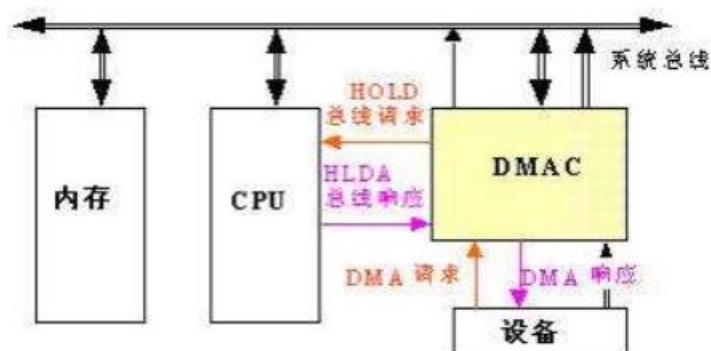
无论是条件传送方式还是查询传送方式，都不能发现和处理预先无法估计的错误和异常情况。为了提高 CPU 的效率、增强系统的实时性，并且能对随机出现的各种异常情况做出及时反应，通常采用中断传送式。

该方式借助于 CPU 响应外部中断请求的能力，实现输入输出的控制。简单地说，就是外设将准备就绪的信号转换成有效的中断请求信号通知给 CPU，CPU 响应中断后，在中断服务子程序中执行 I/O 指令，进行数据传送。

查询传送方式是由 CPU 来查询外设的状态，CPU 处于主动地位，而外设处于被动地位。中断传送方式则是由外设主动向 CPU 发出请求，等候 CPU 处理。在没有发

出请求时，CPU 和外设都可以独立进行各自的工作。

### (3) 直接存储器存取控制方式 DMA



DMA 传送方式是在存储器和外设之间、存储器和存储器之间进行数据传送，传送过程无需 CPU 介入。

在 DMA 方式下，高速外部设备利用专用的接口电路直接和存储器进行高速数据交换，CPU 暂停指令的执行并让出总线控制权。

与前两种方式不同，在 DMA 方式下，数据的传送不依赖 CPU 执行 I/O 指令，而是直接由专用的接口电路 DMA 控制器 (DMAC) 来控制外设和内存之间的数据传送。

- 当接口准备好就向 DMAC 发 DMA 请求，接着 CPU 通过 HOLD 引脚接收 DMAC 发出的总线请求；
- CPU 在完成当前总线操作后，就发出 HLDA 的总线响应信号给 DMAC，DMAC 收到此信号后便接管对总线的控制权，开始 DMA 操作。
- 当 DMA 传送结束，DMAC 将 HOLD 信号变成低电平，并放弃总线的控制权。
- CPU 检测到 HOLD 为低后，也将 HLDA 变成低，并恢复对总线的控制权。

## 4.1 中断的基本概念

### (1) 中断

CPU 在执行程序中, 由于突发(随机)事件的发生, 引起 CPU 暂停现行政程序的执行, 转去执行另外一段相应的处理程序(称为中断服务程序), 以处理该事件, 处理程序执行完后, CPU 再返回到原先暂停的程序, 执行中断开始前将要执行的那条指令(称为断点)。这一过程称为中断。

### (2) 中断源

能够引起中断的事件或外设称为中断源。即有软件中断源和硬件中断源。

硬件中断源主要包括外设(如键盘、打印机等)、数据通道(磁盘机、磁带机等)、时钟电路(如定时计数器 8253)和故障源(如电源掉电)等;

软件中断源主要包括为调试程序设置的中断(如断点、单步执行等)、中断指令(如 INT 21H 等)以及指令执行过程出错(如除法运算时除数为零)等。

### (3) 中断处理过程

对于一个中断源的中断处理过程应包括中断请求、中断响应、保护断点、中断处理和中断返回等五个过程。

①中断请求: 是中断源向 CPU 发出的请求中断的要求。软件中断源是在 CPU 内部由中断指令或程序出错直接发中断请求; 而硬件中断源必须通过专门的电路(如中断控制器 8259A)将中断请求信号送给 CPU, CPU 也有专门的引脚接收中断请求信号。

例如, 8086/8088 CPU 用 INTR 引脚(可屏蔽中断请求)和 NMI 引脚(非屏蔽中断请求)接收硬件中断请求信号。一般外设发出的都是可屏蔽中断请求。

②中断响应: 是指当计算机系统接收到中断请求后应作出的反应。中断响应应具

备两个条件:一是中断允许触发器的状态为 1 (即  $IF=1$ , 开中断), 二是 CPU 在执行完现行指令之后。

③保护断点:所谓断点是指处理完中断后返回主程序时执行的第一条指令的逻辑地址(即当时的 CS 和 IP 的值)。因此, CPU 一旦响应中断, 需要对其正在执行程序的中断点信息进行保护(即将当前 CS 和 IP 的值压入堆栈), 以便在中断处理结束后仍能回到该断点处继续执行。

④中断处理:是指执行完成中断服务功能的处理程序。它主要包括保护现场(将中断服务程序中涉及到的寄存器内容压入堆栈)、中断服务(可完成中断服务的功能)和恢复现场(将原先压入堆栈的寄存器内容再弹回到 CPU 相应的寄存器中)等。

⑤中断返回:是指执行完中断服务程序后返回到原先被中断的程序。它是保护断点的逆过程, 即将原先压入堆栈的断点值弹回到 CS 和 IP 中, 继续执行原先被中断的程序。

## 4.2 8086 中断处理过程

中断响应的操作过程, 对于可屏蔽中断、非可屏蔽中断和内部中断, 是不尽相同的。

### 1、可屏蔽中断的响应和处理过程

(1)CPU 要响应可屏蔽中断请求, 必须满足一定的条件, 即中断允许标志置 1 ( $IF=1$ ), 无内部中断, 没有非可屏蔽中断 ( $NMI=0$ ), 没有总线请求。

(2)当某一外部设备通过其接口电路中断控制器 8259A 发出中断请求信号时, 经 8259A 处理后, 得到相应的中断向量号, 并同时向 CPU 申请中断  $INTR=1$ 。

(3)CPU 执行完当前指令后便向 8259A 发出中断响应信号  $INTA\#$ , 表明 CPU 响应该可屏蔽中断请求。

(4) 8259A 连续两次(2 个总线周期)接收到 INTA#的中断响应信号后,便通过总线将中断向量号送 CPU。

(5) 保护断点。将标志寄存器内容、当前 CS 内容及当前 IP 内容压入堆栈。

(6) 清除 IF 及 TF,以便禁止其他可屏蔽中断或单步中断发生。

(7) 根据 8259A 向 CPU 送的中断向量号 n 求得中断向量,从中断向量表中得到相应中断处理程序首地址(段内偏移地址和段地址),并将其分别置入 IP 及 CS 中。

(8) 中断处理程序包括保护现场、中断服务、恢复现场等部分

(9) 中断处理程序执行完毕,最后执打一条中断返回指令 IRET,将原压入堆栈的标志寄存器内容及程序断点地址弹出至原处。

## 2. 非可屏蔽中断的响应和处理过程

(1) 非可屏蔽的中断请求在 NMI 端加入。CPU 对它的响应不受 IF 位的控制(但可以在外部逻辑中对加入 NMI 端的信号进行控制)。

(2) 与可屏蔽中断一样,非可屏蔽中断也要等待当前指令执行结束。如果同时出现了非可屏蔽中断请求和可屏蔽中断请求,CPU 将优先响应非可屏蔽中断请求。因为非可屏蔽的中断类型号为 2,是微处理器硬件决定的,所以不需要从外部取回一个字节的 interrupt type 操作,非可屏蔽中断响应的其他操作和可屏蔽中断相同。

## 3. 内部中断的响应和处理过程

所有的内部中断,其中断响应操作有以下共同点:

(1) 中断类型号要么是指令码给定的,要么是处理硬件决定的,都不需要从外部逻辑输入;

(2) 没有包括 INTA#信号的响应周期;

- (3) 不受 IF 位的控制, 但单步中断受 TF 位控制;
- (4) 内部中断响应也要执行可屏蔽中断响应的 (5) ~ (9) 项操作。
- (5) 保护断点。将标志寄存器内容、当前 CS 内容及当前 IP 内容压入堆栈。
- (6) 清除 IF 及 TF, 以便禁止其他可屏蔽中断或单步中断发生。
- (7) 根据 8259A 向 CPU 送的中断向量号 n 求得中断向量, 从中断向量表中得到相应中断处理程序首地址(段内偏移地址和段地址), 并将其分别置入 IP 及 CS 中。
- (8) 中断处理程序包括保护现场、中断服务、恢复现场等部分
- (9) 中断处理程序执行完毕, 最后执行一条中断返回指令 IRET, 将原压入堆栈的标志寄存器内容及程序断点地址重又弹出至原处。

## 5.1 8086/8088 中断系统

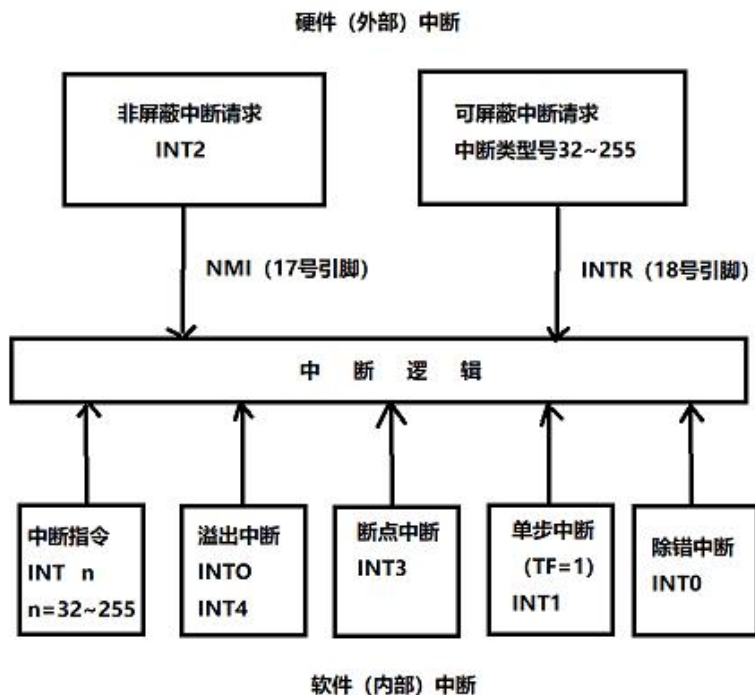
### 1、中断源类型

8086/ 8088 的中断源共有 256 种, 可分为两大类: 一类来自 CPU 的外部, 由外设的请求引起, 称为硬件中断(又称外部中断); 另一类来自 CPU 的内部, 由执行指令时引起, 称为软件中断 (又称内部中断)。

### 2、中断类型号

8086/8088CPU 可以处理 256 种不同类型的中断, 每一种中断都给定一个编号 (0~255), 称为中断类型号, CPU 根据中断类型号来识别不同的中断源。





### 硬件中断(外部中断)

8086/8088CPU 有两条外部中断请求线 NMI（非屏蔽中断）和 INTR（可屏蔽中断）

#### (1) 非屏蔽中断 NMI（中断类型号为 2 是固定的）

整个系统只有一个非屏蔽中断，它不受 IF 标志位的屏蔽。一旦出现，CPU 将予以响应。非屏蔽中断被响应时，其中断矢量号不由外部中断源提供，而是由系统固定分配。

#### (2) 可屏蔽中断 INTR

可屏蔽中断请求信号从 INTR 引脚送往 CPU，高电平有效。它受中断允许标志位 IF 的影响和控制。当 IF 值 1（STI 指令）时，表明可屏蔽中断被允许，CPU 响应可屏蔽中断。当 IF 置 0（CLI 指令）时，表明可屏蔽中断被禁止，CPU 不响应可屏蔽中断，并将该中断信号挂起，直到 IF 被置位或外部事件撤销中断请求为止。

CLI 禁止中断发生，STI 允许中断发生。程序员可以改变段地址和偏移地址，但是在这个过程中如果需要改变段寄存器 SS 和 SP 必须禁止中断，当改变完成后再恢复中断（也就是说在 cli 指令后需要有与其配对的 sti 指令，否则计算机最常见的反应就是崩溃）



## 软件中断(内部中断)

8086/8088 的软件中断主要有五类，分为三类

### (1) 处理运算过程中某些错误的中断

执行程序时，为及时处理运算中的某些错误，CPU 以中断方式终止正在运行的程序，提醒程序员出错

①除错中断(中断类型号为 0)。在 8086/8088CPU 执行除法指令(DIV/IDIV) 时，若发现除数为 0，或所得的商超过了 CPU 中有关寄存器所能表示的最大值，则立即产生一个类型号为 0 的内部中断，CPU 转去执行除法错中断处理程序。

②溢出中断 INTO (中断类型号为 4)。CPU 进行带符号数的算术运算时，若发生了溢出，则标志位 OF=1，若此时执行 INTO 指令，会产生溢出中断，打印出一个错误信息，结束时不返回，而把控制权交给操作系统。若 OF=0，则 INTO 不产生中断，CPU 继续执行下一条指令。INTO 指令通常安排在算术指令之后，以便在溢出时能及时处理。

### (2) 为调试程序设置的中断

①单步中断(中断类型号为 1)。当 TF=1 时， 每执行一条指令，CPU 会自动产生一个单步中断。单步中断可一条一条指令地跟踪程序流程，观察各个寄存器及存储单元内容的变化，帮助分析错误原因。单步中断又称为陷阱中断，主要用于程序调试。

②断点中断(中断类型号为 3)。调试程序时可以在一些关键性的地方设置断点，它相当于一组 INT 3 指令插入到程序中，CPU 每执行到断点处，INT 3 指令便产生一个中断，使 CPU 转向相应的中断服务程序。

### 3、中断向量

所谓中断向量，就是中断服务子程序的入口逻辑地址（32 位），即由段地址（16 位）和偏移地址（16 位）构成。

### 4、中断向量表

中断向量表是存放中断向量的一个特定的内存区域。它位于整个内存区域的最低端，物理地址范围从 00000H~ 003FFH。

		IP内容
专用中断 (5个)	00000H	类型0中断入口地址
	00003H	(除法出错)
	00004H	类型1中断入口地址
	00007H	(单步中断)
	00008H	类型2中断入口地址
系统保留中断(27个)	0000BH	(NMI)
	0000CH	类型3中断入口地址
	0000FH	(断点中断)
	00010H	类型4中断入口地址
	00013H	(溢出中断)
	00014H	类型5中断入口地址
	⋮	⋮
	0007CH	类型31中断入口地址
	0007FH	⋮
	00080H	类型32中断入口地址
用户自定义中断(224个)	⋮	⋮
	003FCH	类型255中断入口地址
	003FFH	⋮

### 5、中断向量地址

存放中断向量的存储单元地址。即存放中断服务程序的入口地址的地址。也可以说就是中断向量在中断向量表中的地址。

## 6、中断类型号和中断向量地址的关系

8086/8088CPU 共有 256 个类型(编号 0~255) 的中断, 每一个类型都有相应的中断服务程序, 每一个中断服务程序的入口地址(中断向量)需要 4 个字节存放。共需要 1024 个字节, 形成中断向量表。

中断向量地址与中断类型号之间的固定关系为:

中断向量地址=中断类型号 X4

=用二进制表示的中断类型号逻辑左移 2 位

## 7、向量中断

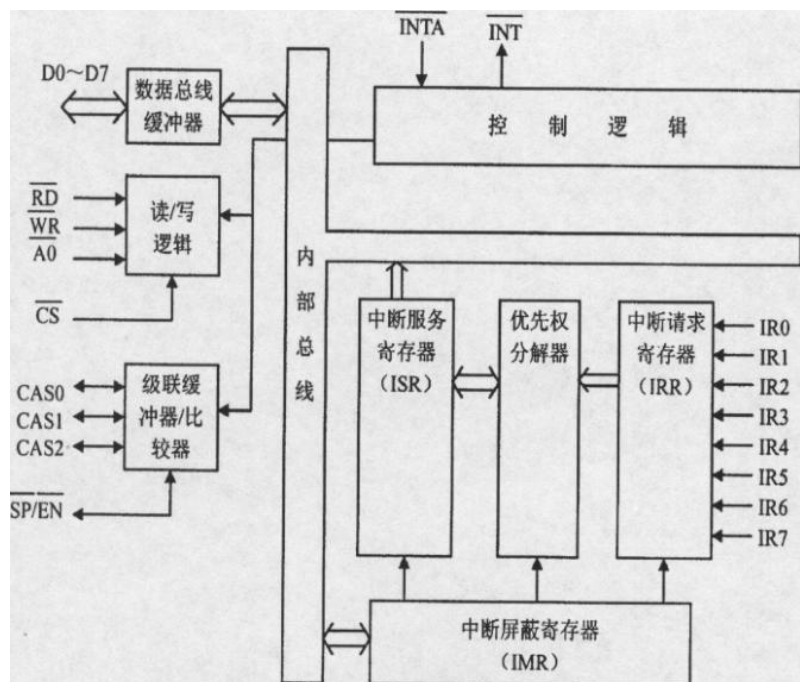
向量中断也称矢量中断, 是指通过中断向量地址获取中断向量(中断服务程序的入口地址)的方式。例如, 8086/8088 就是具有向量中断的中断系统。

8086/8088 外部的每个中断源都预先指定一个中断类型号, 对应于每个中断源的中断服务程序的入口地址(中断向量)都按类型号顺序地存放在内存中, 形成一张地址表, 称为中断向量表。当 CPU 识别到中断请求并予以响应时, 控制逻辑就会产生相应的中断类型号送给 CPU, CPU 的硬件按类型号直接产生相应的向量地址。按这个地址访问中断向量表, 取得相应的中断服务程序的入口地址(中断向量), 转入相应的中断服务程序。

### 6.1 中断控制器 8259A 的内部结构及引脚

8259A 是用于管理和控制外部中断请求, 实现中断优先级判定, 提供中断类型号. 屏蔽中断输入等功能。另外, 它还负责为 CPU 处理中断提供中断类型号。

8259A 的内部主要包括实现功能的部件: 中断请求寄存器 IRR、中断优先级判别器 PR、中断服务寄存器 ISR、中断屏蔽寄存器 IMR; 实现级联的部件: 级联缓冲/比较器; 实现工作方式控制的部件; 读写控制逻辑等。



## 1. 8259A 的内部结构

### (1) 中断请求寄存器 IRR

8 位, 接受并锁存来自  $IR0 \sim IR7$  的中断请求信号, 当  $IR0 \sim IR7$  上出现某一中断请求信号时, IRR 对应位置被置 1.

### (2) 中断屏蔽寄存器 (IMR)

8 位, 若 IRR 中记录的各级中断中有任何一级需要屏蔽, 只要将 IMR 的相应位置 1 即可, 未被屏蔽的中断请求进入优先级判断器。”

### (3) 中断服务寄存器 ISR

8 位, 保存当前正在处理的中断请求。

### (4) 优先级判别器 PR

能够将各中断请求中优先级最高者选中, 并将 ISR 中相应位置 1。

#### (5) 数据总线缓冲器

数据总线缓冲器是 8259A 与系统之间传送信息的数据通道。

#### (6) 读/写控制逻辑

读/写控制逻辑包含了初始化命令字寄存器和操作命令字寄存器。其功能 CPU 发读信号时将 8259A 的状态信息放到数据总线上; 当 CPU 发写信号时, 将 CPU 发来的命令字信息送入指定的命令字寄存器中。

#### (7) 级联缓冲/比较器

用来存放和比较在系统中用到的所有 8259A 的级联地址。

### 2. 8259A 的外部引脚

8259A 采用 28 脚双列直插封装形式。

CS# : 片选信号, 输入, 低电平有效, 来自地址译码器的输出。只有该信号有效时, CPU 才能对 8259A 进行读/写操作。

WR# : 写信号, 输入, 低电平有效, 通知 8259A 接收 CPU 从数据总线上送来的命令字。

RD# : 读信号, 输入, 低电平有效, 用于读取 8259A 中某些寄存器的内容。

D7~D0 : 双向, 三态数据线, 连接系统数据总线的 D7~D0, 用来传送控制字、状态字和中断类型号等。

IR7~IR0 : 中断请求信号, 输入, 从 I/O 接口或其他 8259A 上接收中断请求信号。

INT : 8259A 向 CPU 发出的中断请求信号, 高电平有效, 该引脚接 CPU 的 INTR 引脚。

INTA# : 中断响应信号, 输入, 接收 CPU 发来的中断响应脉冲以通知 8259A 中断

请求已被响应,使其将中断类型号送到数据总线上。

CAS0~CAS2 : 级联总线,输入或输出,用于区分特定的从控制器件。8259A 作为主控器时,该总线为输出,作为从控制器时,为输入。

SP#/EN# : 从片/允许缓冲信号,输入或输出,该引脚为双功能引脚。在缓冲方式中,该引脚被用作输出线,控制收发器的接收或发送;在非缓冲方式中,该引脚作为输入线,确定该 8259A 是主控制器(置 1)还是从控制器(置 0)。

A0 位地址输入信号,用于对 8259A 内部寄存器端口的寻址。

## 6.2 8259A 的中断工作方式

优先级设置方式->{完全嵌套方式、特殊嵌套方式(ICW4)}、{优先自动循环方式、优先特殊循环方式(OCW2)}

中断结束方式->{自动 EOI 方式、普通 EOI 方式(ICW4)}、特殊 EOI 方式(OCW2)

中断屏蔽方式->{普通屏蔽方式、特殊屏蔽方式(OCW3)}

中断触发方式->{边沿触发方式、电平触发方式(ICW1)}、中断查询方式(OCW3)

总线连接方式->{缓冲方式、非缓冲方式(ICW4)}

8259A 的工作方式是通过编程发送命令字(初始化命令字 ICW 和操作命令字 OCW)来设置或修改的。

## 4. 8259A 的编程

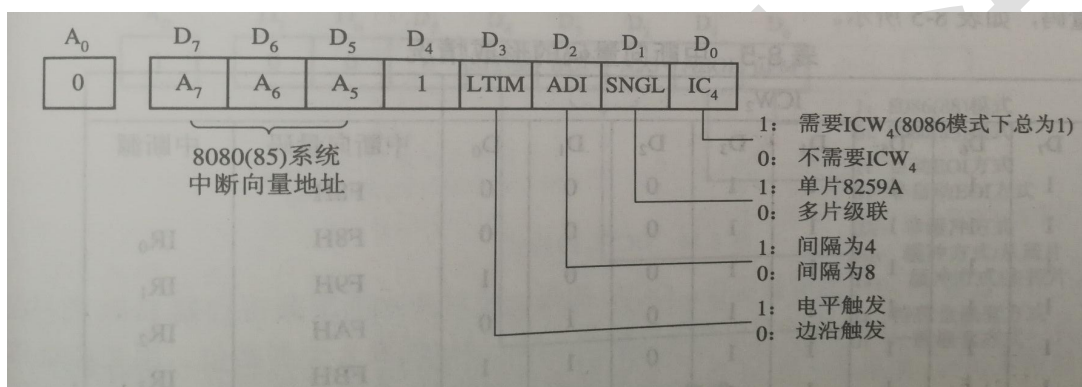
8259A 是可编程的中断控制器,它的工作状态和操作系统是由 CPU 通过命令字进行控制的。8259A 有两类命令字:初始化命令字 ICW 和操作命令字 OCW。相应地,在 8259A 的控制部分有 7 个 CPU 可访问的寄存器,这些寄存器分为两组:一组用来存 ICW,另一组存 OCW。

当计算机刚启动时, 用初始化程序设定 ICW, 用来建立起 8259A 操作的初始状态, 此后的整个工作过程中该状态保持不变。相反, 操作命令字 OCW 用于动态控制中断处理, 是在需要改变或控制操作时随时发送的。每片 8259A 有两个片内地址 A0=0 和 A0=1. 所有的命令都是通过这两个端口来发送的。

### (1) 初始化命令字

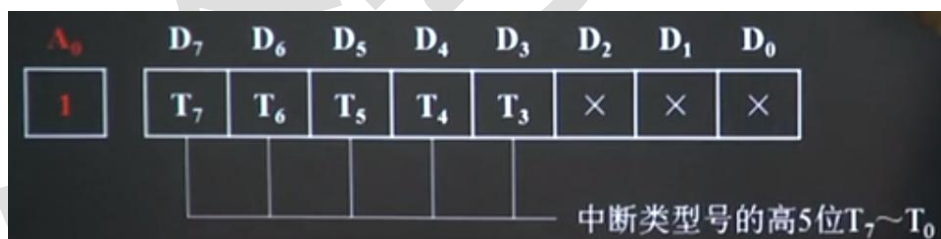
#### ① ICW1

ICW1 主要用于设置工作方式, 写入 A0=0 的端口, 其格式如下图所示。



#### ② ICW2

ICW2 主要用于设置中断类型号, 写入 A0=1 的端口, 其格式如下图所示。

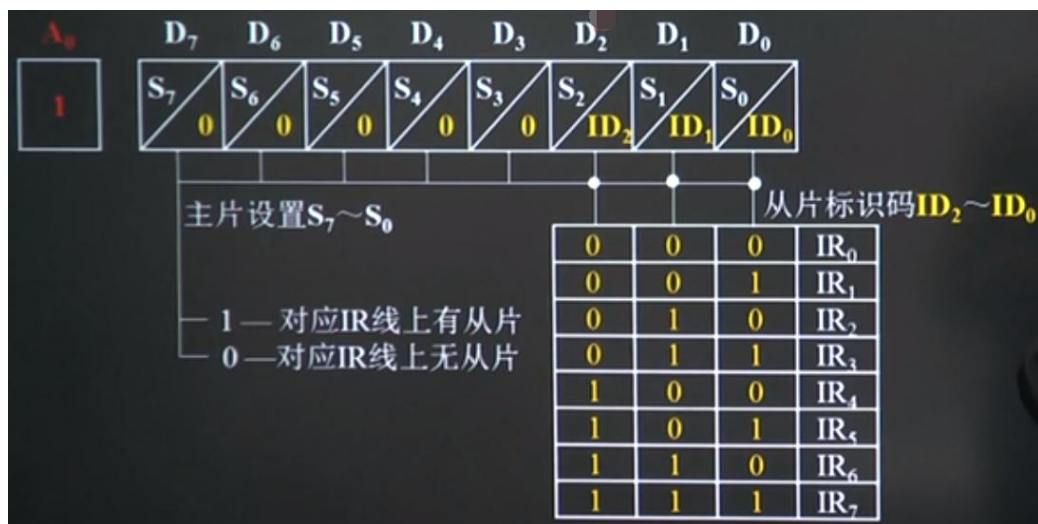


低 3 位 D<sub>2</sub>~D<sub>0</sub> 的实际内容由 8259A 根据中所请求来自 IR<sub>7</sub>~IR<sub>0</sub> 的哪一个输入端, 自动填充为 000~111 的编码, 与高 5 位一同构成 8 位的中断类型号。

#### ③ ICW3

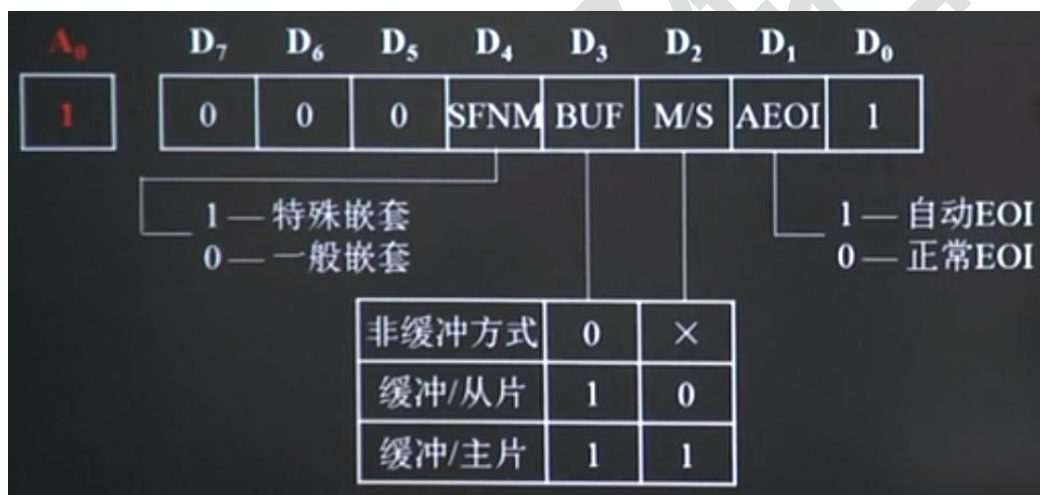
ICW3 用于设置级联, 写入 A0=1 的端口, 其格式如下图所示。





#### ④ICW4

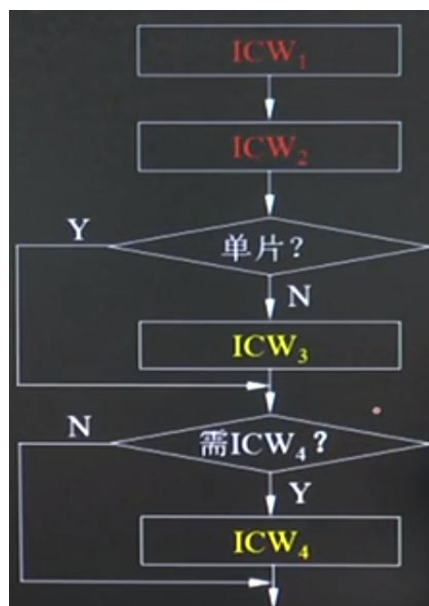
ICW4 用于设置工作方式，写入  $A0=1$  的端口，其格式如下图所示。



#### (2) 初始化编程

8259A 初始化流程图如图所示。任何一种 8259A 的初始化都必须发送 ICW1 和 ICW2。只有在 ICW1 中指明需要 ICW3 和 ICW4 以后，才发送 ICW3 和 ICW4。一旦初始化以后，若要改变某一个 ICW，则必须重新在进行初始化编程，不能只写入单独的一个 ICW。



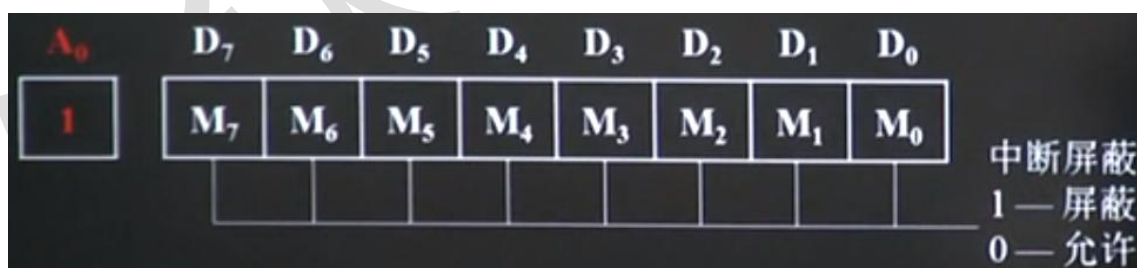


### (3) 操作命令字

系统初始化完成以后，可以在应用程序中随时向 8259A 送操作命令字，以改变 8259A 的工作方式，读出 8259A 内部寄存器的值。

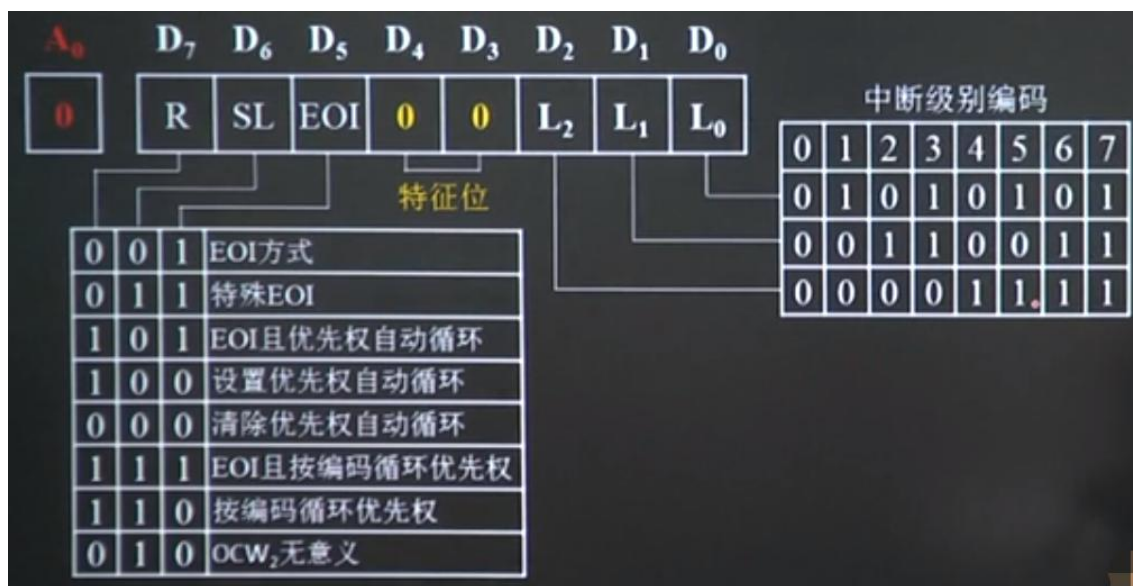
#### ①OCW1

OC1 的功能是设置和清除中断屏蔽寄存器的相应位，写入 A0=1 的端口，其格式如下图所示。



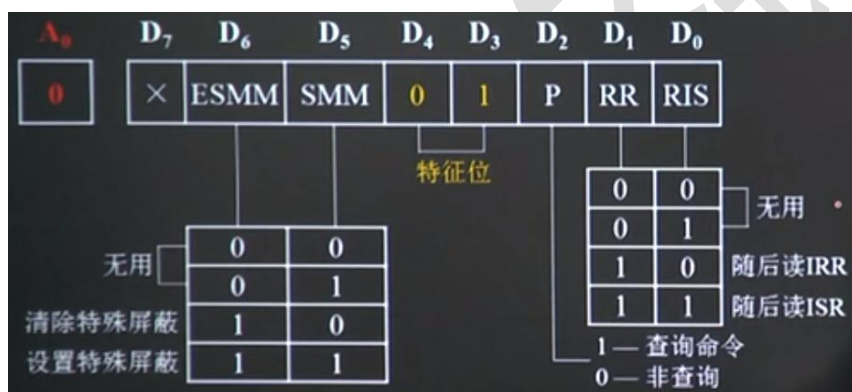
#### ②OCW2

OCW2 用于设置优先级循环方式和中断结束方式，写入 A0=0 的端口，其格式如下图所示。



### ③OCW3

OCW3 的功能有三方面: 设置和撤销中断屏蔽方式、设置中断查询方式以及设置对 8259A 的内部寄存器读出, 写入 A0=0 的端口, 其格式如下图所示。



### 填空题

- 设计输入输出接口电路时, 输入接口电路的关键器件是 三态缓冲器; 输出接口电路的关键器件是 锁存器。
- I/O 端口的编址方式般有 统一编址方式 和 独立编址方式。
- 微机系统中, 主机和外设之间交换信息通常采用 程序控制方式 (有条件和无条件)、中断控制方式 和 DMA 控制方式。

## 选择题

1. CPU 与 I/O 设备之间传送的信息有 (D)

A. 控制信息      B. 状态信息      C. 数据信息      D. 以上三种都有

2. 在 DMA 传送方式下数据传送 (A)

A. 不需要 CPU 干预也不需要软件介入

B. 需 CPU 干预又需要软件介入

C. 不需要 CPU 干预但需要软件介入

D. 需 CPU 干预不需要软件介入

3. CPU 传送数据最快的 I/O 方式是 (C)

A. 中断      B. 查询      C. DMA      D. 无条件

1. CPU 用无条件传送方式工作比中断传送方式用效率高。

2. 主机与设备传送数据时, 采用程序查询控制方式, 主机与外设是串行工作。中断查询是并行工作。

3. DMA 在传送过程中, 实现总线控制的部件是 DMAC。

## 简答题

1. 在 8086/8088 系统中, I/O 端口的寻址方式是什么?

答: I/O 端口的寻址方式是直接端口寻址方式 (8 位端口地址) 和间接端口寻址方式 (16 位端口地址) 2 种。

2. 什么是端口? 8086 共可有多少个端口?

答: 端口是在接口电路中可编址的寄存器, 或者说是能被 CPU 访问的寄存器。8086

共可有 64K 个字节端口。

### 填空题

1. Intel 8086/8088 中断分为硬件中断和软件中断，其中断响应过程都要把 FR 和 断点地址 (CS:IP) 推入堆栈，且对相应标志位把 TF (单步执行) 和 IF (中断允许) 清零。
2. 8086 CPU 在每条指令的最后一个时钟周期采样 INTR 信号,若其为 高 电平，及标志 IF 为 1，并且当前无 高 优先级服务，则 8086 响应 INTR 中断请求。
3. 中断向量地址是指 存放中断向量的单元 的地址。

### (二) 选择题

1. 8086/8088 的中断是向量中断，其中断服务程序的首地址由 D 提供。  
A. 外设中断控制器  
B. CPU 的中断逻辑电路  
C. 从中断逻辑器读回中断类型号左移 2 位  
D. 由中断类型号指向的向量地址表中读出
2. 下 面的中断中，只有 D 需要硬件提供中断类型号。  
A. INTO      B. INT n      C. NMI      D. INTR

### (三) 判断题

1. 在一个中断服务中，都必须对主程序所产生的一些中间结果进行现场保护，并在返回时恢复现场。（错误）  
只有发生冲突在子程序对主程序对中间结果产生影响的需要保护。
2. 进入非屏蔽中断响应周期，被响应的外设(或接口芯片)必须送出一个字节的中断向量类型号。（错误）

只有可屏蔽中断需要中断向量类型号。

3. 在执行软件中断时, 如果有可屏蔽中断请求, 并且这时  $IF=1$ , CPU 也不会响应这个可屏蔽中断。(错误)

软中断单步中断在级别中最低

#### (四) 简答题

1. 8086 的可屏蔽中断 INTR 和非屏蔽中断 NMI 有何不同?(要求从触发方式、优先级、CPU 对其响应来比较)

答: ①INTR 是高电平触发; NMI 是边沿触发。

②NMII 的优先级高于 INTR

③CPU 响应可屏蔽中断要执行两个中断响应周期, 获取相应的中断类型号; 非屏蔽中断的中断类型号固定为 2, 因此, 不需要执行两个中断响周期来获取。

2. 简答硬件中断和调用子程序的异同?

答: 相同: 保护断点, 实现返回。都是暂停执行现行程序, 转去执行另一程序段。

不同: 调用子程序是事先安排好的, 调用点固定, 并有远距离调用和近距离调用之分: 不会影响标志寄存器的内容。硬件中断是随机的, 调用点不固定, 都是远距离调用: 会影响标志寄存器的内容, 故需要保护标志寄存器。

#### (一) 填空题

1. 3 片 8259A 级联, 最多可接 22 个可屏蔽中断源。

2. 有 2 片 8259A 级联, 从片接入主片的 IR2. 则主片 8259A 的初始化命令字 ICW<sub>1</sub> 应为 0000 0 100B 从片的初始化命令字 ICW<sub>3</sub> 应为 0000 0010B

3. 多片 8259A 级联. 非缓冲方式不运行, 主片 SP/EN 应接高电平. 从片 SP/EN 应接低电平.

### (二) 选择题

1. 在 8259A 中, 寄存器 IMR 的作用 (C)

A. 记录处理的中断请求

B. 判断中断优先级的级别

C. 有选择的屏蔽

D. 存放外部输入的中断请求信号

2. 当多片 8259A 级联使用时, 对于从片 8259A, 级联信号 CAS2~ CAS 是 (A)

A. 输入信号      B. 输出信号      C. 全部信号      D. 中断信号

3. 对于 8259A, 当其单片使用时可同时接收①外设的中断请求: 当级联时, 其主片的②应与从片的③连接。

①A. 8 个      B. 12 个      C. 4 个      D. 16 个

②A. SP/EN      B. CS      C. INTA      D. IR;

③A. INT      B. INTA      C. CS      D. CAS

### (三) 判断题

1. 判断: 如果 8259A 的中断结束方式设定为自动 EOI 结束方式, 则在执行某一级的中断服务程序中由较低优先级的中断源请求中断, 就立即响应较低优先级的中断。(正确)

分析: 对自动 EOI 结束方式, 8259A 自动会在中断响应周期的第二个 INTA 脉冲结束时, 使 ISR 中的相应位复位。此时, 尽管该中断处理尚未完成, 但是由于 ISR 中的相应位为 0, 8259A 认为该中断处理已结束。因此, 只要有中断请求, 且符合



CPU 响应条件, 就会立即响应较低级优先级的中断。

#### (四) 简答题

1. 某可编程中断控制器 8259A, 初始化命令字 ICW2 的内容为 23H. 问该片中断类型号的范围是多少?

分析: 单片 8259A 连续占 8 个中断类型号。每个中断类型号有两部分组成, 如:



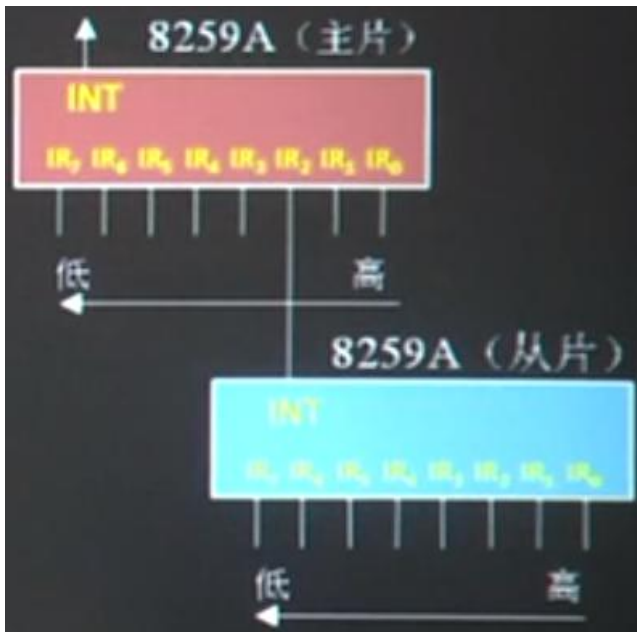
本题初始化命令字内容为 23H 所以该片的中断类型号范围为:



答案: 中断类型号范围: 20H~ 27H

2. 设有 2 片 8259A 接成主从级联方式, 主片设为特殊完全嵌套方式. 从片设为士般完全嵌套方式) 从片的 INT 引脚接至主片的 IR<sub>7</sub> 引脚。写出主、从片 8259A 引脚的中断优先级顺序。

分析: 本例主要考查级联方式下, 各 8259A 的 IR<sub>7</sub> 引脚的中断优先级别。单片 8259A 作在完全嵌套方式下, 中断优先级从高到低顺序为 IR<sub>0</sub>, IR<sub>1</sub>, ... IR<sub>7</sub>. 中断嵌套时不能被同级打断。特殊完全嵌套方式下. 中断优先级从高到低顺序为 IR<sub>0</sub>, IR<sub>1</sub>, ..., IR<sub>7</sub>,. 中断嵌套时能被同级打断。所以该中断系统优先级顺序如图所示。



答案: 中断优先级从高到低顺序为

IR0 IR1, (IR0, IR1, IR2, IR3, IR4, IR5, IR6, IR7), IR3, IR4, IR5, IR6, IR7

高----->低