

第 5 章

CHAPTER

总线、中断与 I/O 系统

本章主要内容包括输入输出系统的组成、功能,总线的分类及输入输出总线的控制方式、通信技术及总线标准化,通道处理机的工作原理及工作过程,输入输出系统中的中断技术。最后针对本章重点内容安排了习题及详细解答。

5.1 总线系统

5.1.1 总线的基本概念

1. 定义

总线是构成计算机系统的互联机构,是多个功能部件之间进行数据传送的公共通路。借助于总线连接,计算机在各系统功能部件之间实现地址数据和控制信息的传送与交换,并在争用资源的基础上进行工作。

2. 分类

1) 按总线所起的作用分类

(1) 内部总线: CPU 内部各寄存器及运算部件之间的总线,称为内部总线。

(2) 系统总线: 除了内部总线外, CPU 与存储器、通道等其他高速设备也需要连接,这类总线称为系统总线。系统总线包括数据总线 DB、地址总线 AB、控制总线 CB。

(3) I/O 总线: 中、低速 I/O 设备互连的总线称为 I/O 总线,又称为通信总线。

2) 按信息传送方向分类

(1) 单向传输。

(2) 双向传输:

- 半双向;
- 全双向。

3) 按使用方法分类

(1) 专用总线(热线): 只连接一对物理部件的总线,如图 5.1 所示。

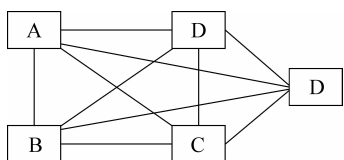


图 5.1 专用总线示例图

优点：系统的流量高，设备很少争用总线。控制简单，不用指明源和目的，失效不太敏感。

缺点：线数多、成本高，不利于模块化，总线的利用率低。

例如 N 台外设，需 $\frac{N(N-1)}{2}$ 根总线。

(2) 非专用总线：总线可被多种功能部件分时共享，但同一时刻只有一对部件使用总线。

优点：总线线数少、造价低，总线接口标准化、模块性强，可扩充能力强。

缺点：系统流量小，争用总线现象时有发生，总线失效敏感。

非专用总线可简单分成单总线结构和多总线结构。单总线结构如图 5.2 所示。

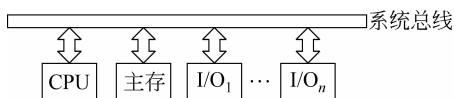


图 5.2 非专用总线的单总线结构

多总线结构的非专用总线在图 5.3 中给出了常用的 5 种结构。

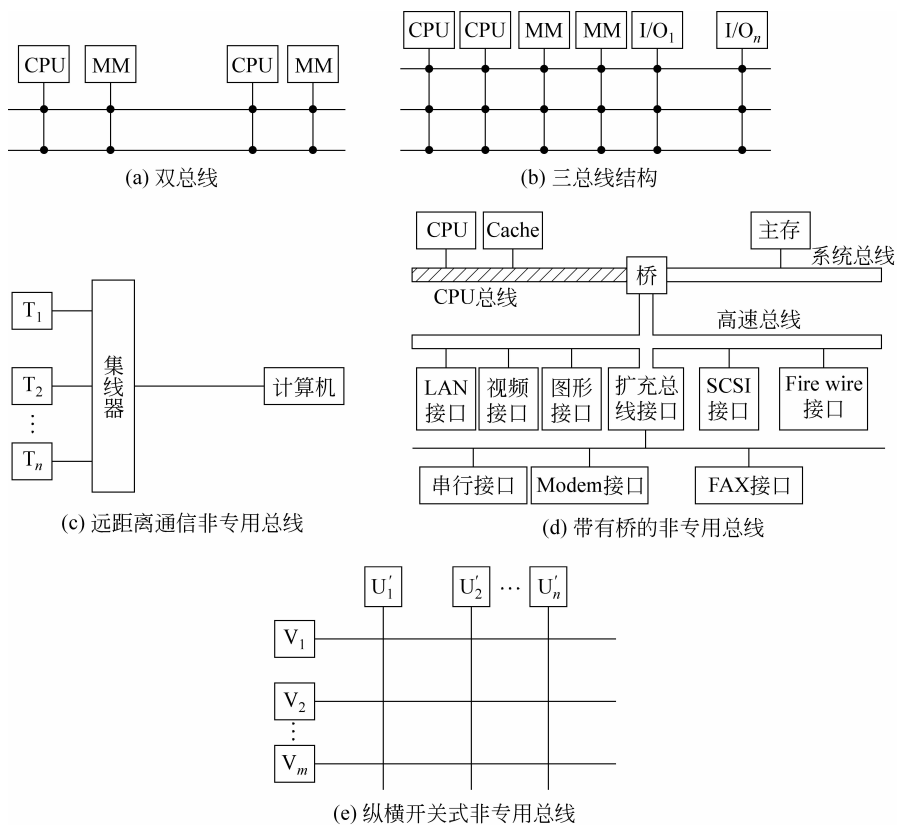


图 5.3 常用的几种非专用总线

对于纵横开关式总线，每行、每列至多只有一对部件可以通信，最大值为 K , $K = \min\{m, n\}$, m 是横向总线线数, n 是纵向总线线数。

在图 5.3 的(d)图中的桥是一种具有缓冲、转换、控制功能的逻辑电路,它目前被广泛使用在计算机系统设计中。

3. 总线标准化

以系统总线为例,系统总线比 CPU 内部总线的种类多得多,连接在系统总线上的部件可能由不同厂商提供。为了总线的通用性,必须制定标准。

总线除了具有规范的标准,还具有物理特性、功能特性和电气特性等基本特征。总线的标准化,实质上是 I/O 接口的标准化。

I/O 功能模块通常简称 I/O 接口,也叫适配器。广义地讲,I/O 接口是指 CPU、主存和外设之间通过系统总线进行连接的标准化逻辑部件,I/O 接口在它动态连接的两个部件之间起到“转换器”的作用,以便实现信息传递。图 5.4 和图 5.5 给出了外围设备通过 I/O 接口与主机的简单连接方式及 I/O 接口的模块框图。

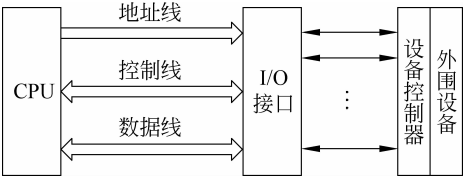


图 5.4 外围设备的连接方式

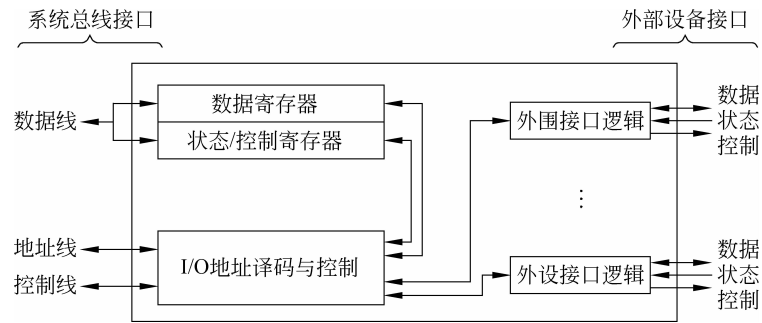


图 5.5 I/O 接口模块框图

一个标准的 I/O 接口可能连接一个设备,也可能连接多个设备,在它所连接的设备之间起转换器的作用,以解决计算机主机与外围设备的各种信号之间的差别。

总线接口通常具有 6 个功能:控制、缓冲、状态设置、数据转换、整理、程序中断。

以传送一个字节的的数据 00000101 为例,即可以串行传送也可以并行传送,如图 5.6 所示。

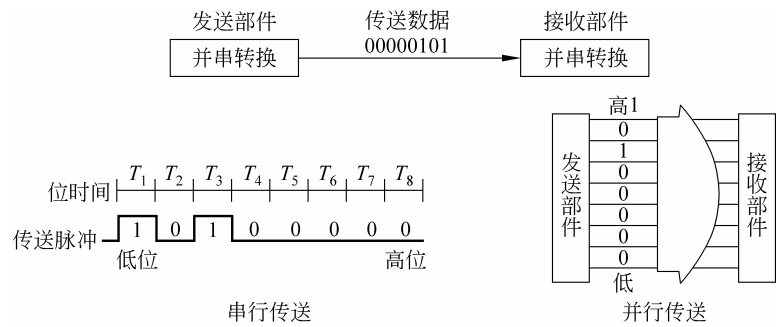


图 5.6 数据的串行与并行传送

总线带宽是指总线本身所能达到的最高传输速率(MB/s)。受总线长度、总线驱动器、接收器性能、连接在总线上的模块数等众多因素影响,可能会造成线上信号的延时和畸变,故最高速率受限。

例 5-1: (1) 某总线在一个总线周期中并行传送 4 字节的数据,假设一个总线周期等于一个总线时钟周期,总线时钟频率为 33MHz,总线带宽是多少?

(2) 如果一个总线周期中并行传送 64 位数据,总线时钟频率升为 66MHz,总线宽度是多少?

解: (1) 设总线带宽为 D_r , 通常的总线周期为 T , T 与总线周期传送的数据量 D 的关系如图 5.7 所示。则

$$T : D = 1 : x$$

$$D_r = x = \frac{D}{T} = D \times f$$

$$= 4\text{B} \times 33 \times 10^6 / \text{s} = 132\text{MB/s}$$

(2) $64 \text{ 位} = 8\text{B} = D$

则:

$$x = D_r = \frac{D}{T} = D \times f = 8\text{B} \times 66 \times 10^6 / \text{s} = 528\text{MB/s}$$

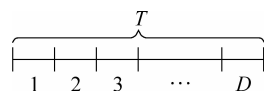


图 5.7 总线周期与总线的时钟周期

总线标准是指总线的机械物理尺寸、引线数目、信号定义、功能和时序、数据传输率、工作频率、总线协议等所进行的统一的严格定义,使它具有高度的科学性和权威性,以便被计算机界广泛接受。表 5.1 所示是几类常用的总线标准。

表 5.1 常用总线标准

总线类型 特征	ISA	EISA	PCI
数据传输位数	8/16	8/16/32	32,64
最高速率	8MB/s	33MB/s	528MB/s
驱动程序	与硬件有关	与硬件有关	与硬件有关

5.1.2 总线的控制方式(仲裁策略)

I/O 系统一般都采用非专用总线。当采用非专用总线时,由于可能发生多个设备或部件同时申请使用总线,就得有总线控制机构按照某种优先次序进行裁决,保证在同一时间内只能有一个高优先级的申请者取得对总线的使用权。

- 分散式总线控制: 总线控制逻辑分散连到总线的各个部件当中。
- 集中式总线控制: 总线控制逻辑集中放在连接到总线上的某个部件或放在单独设置的硬件设备中。

集中式总线控制方式包括串行链接、定时查询和独立请求。

1. 串行链接

所有部件都经公共的“总线请求”线向总结控制器发出要求使用总线的申请。只有当

“总线忙”信号未建立时,“总线请求”才被总线控制器响应,送出“总线可用”信号,它串行通过每个部件。如果某个部件接收到“总线可用”信号,但未发过“总线请求”进,就将信号继续送往下一个部件。如果该部件接到“总线可用”信号并发出过“总线请求”时,则停止传送“总线可用”信号。该部件建立“总线忙”,并去除其“总线请求”,意即该部件获得了使用总线的权利,之后即可准备数据的传送。在数据传送期间,“总线忙”信号维持“总线可用”的建立。完成传送后,部件去除“总线忙”信号,“总线可用”随之去除。其后,当“总线请求”再次建立时,就开始新的总线分配过程,其原理如图 5.8 所示。

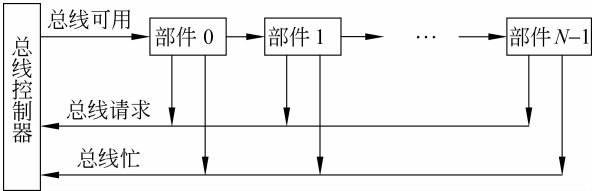


图 5.8 集中式串行链接

优点：选择设备优先级的算法简单,控制线数只有 3 根,可扩充性好。
缺点：“总线可用”失效敏感,优先级灵活性差,“总线可用”信号的传递会限制分配的速度,受总线长度的影响,增加设备受限。

2. 集中式定时查询

总线上的每个部件通过“总线请求”线发出请求,若总线处于空闲时,“总线忙”信号未建立,则总线控制器收到请求后,让计数器开始计数,定时地查询各个部件以确定是谁发出的请求。当查询线上的计数值与发出请求的部件号一致时,该部件就建立“总线忙”,并使计数器停止计数,即停止查询,同时去除自身的请求信号,该部件获得总线使用权,开始进行数据的传送,其过程如图 5.9 所示。

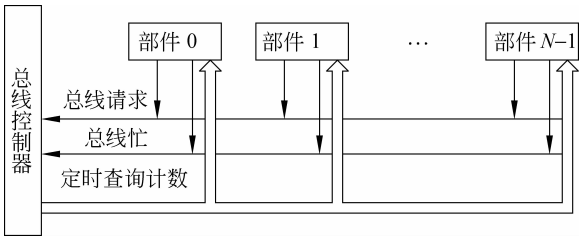


图 5.9 集中式定时查询

优点：

- (1) 可靠性高。
- (2) 优先级的确定灵活：
 - ① 每次总线分配前,计数器既可清零,也可从中止点继续查询,让每个部件有同等使用总线的机会。计数器的初值是可变的,就可以让某个设备具有最高优先级。
 - ② 每次总线分配前,可将设备号重新设定,则可指定设备为任何优先级。

缺点：

- (1) 扩展性差,控制较复杂,需要 $2 + \lceil \log_2 N \rceil$ 根总线。
- (2) 总线的分配速度取决于计数信号的频率和部件数。

3. 集中式独立请求

每个部件的“总线请求”信号可以同时并行送到总线控制器。控制器可以根据多种算法中的某一种对同时到来的多个请求进行仲裁,随后将“总线准许”信号送回被选中的部件。同时去掉该部件的请求信号,建立“总线已被分配”的信号,开始数据的传送。传送结束后,部件去除“总线已被分配”信号,并经控制器去除本部件的“总线允许”信号,如图 5.10 所示。

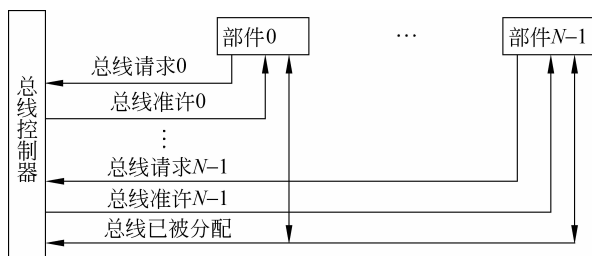


图 5.10 集中式独立请求

优点:

- (1) 分配速度快,既不串行传递,也不定时查询。
- (2) 优先级确定灵活。可用程序方式,也可用硬件屏蔽位方式。

缺点:

- (1) 线数过多,共需 $2N+1$ 根总线。
- (2) 控制复杂。

现实应用中巨、大、中型机上宜用后两种,而小、微型机上则用串行连接。

5.1.3 总线的通信技术

当获得总线的使用权后,还必须给出通信的“源”和“目的”部件、传送信息的类型和方向等,之后才开始数据的传送。传送方法又分为同步通信和异步通信两种方式。

1. 同步通信(不应答方式)

两个部件之间的信息传送是通过定宽、定距的系统时标进行同步的。这种方式的信息传送速率高,受总线的长度影响小,但会因时钟在总线上的时滞而造成同步误差,且时钟线上的干扰信号易引起误同步。

提高可靠性的方法是对收到数据正确与否给予回答。但此时同步时间片的宽度应是最低速设备的,所以这种方法会降低同步通信的速度。另一种方法是正确时不回答,错误时发一个出错信号,此时不会降低正常时的传送速率。但为了能做到一旦出错便要重发,就必须设置较大容量的缓冲器,以备重发之用。

2. 异步通信(应答方式)

异步通信分为单向控制和双向控制两类。

- 单向控制是指通信过程中只由源或目的部件中的一个控制。又可进一步分为单

向源控式和单向目控式两种。

- 双向控制是指由源和目的双方共同控制(请求/回答)。又可进一步分为源控式为主和目控式为主两种。其中源控式为主又可区分为互锁式源控与非互锁式源控。

(1) 单向源控式通信

单向源控的通信原理示意图如图 5.11 所示。

优点：简单、高速。

缺点：没有来自目的部件的回答信号，无法知道收到数据的对与错。而且对具有不同速度的部件之间的通信，需设置一定容量的缓冲器来暂存来不及处理的信息。对“数据准备”线的要求极高，干扰要特小才可以稳定工作。

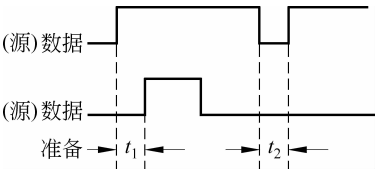


图 5.11 单向源控式

(2) 单向目控式通信

其通信原理示意图如图 5.12 所示。

目的部件建立“数据请求”后，使源部件把数据放在“数据”线上，经二次传送延迟，即 $2t_d$ 后，数据才到达目的部件，并由其校验有效性。如果有错，目的部件发“数据有错”信号代替下一个“数据请求信号”。显然，目的部件接收数据后的这种校验会降低传送速率。

(3) 双向源控非互锁

其通信原理示意图如图 5.13 所示。

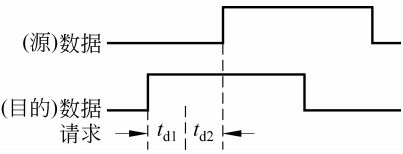


图 5.12 单向目控式

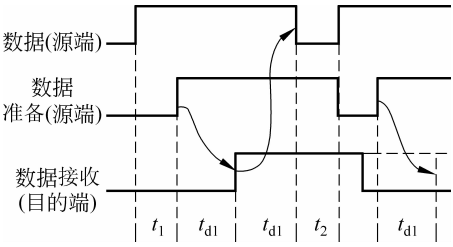


图 5.13 双向源控非互锁方式

优点：

- (1) 提供出错控制。
- (2) 便于实现不同速度部件之间的通信。

缺点：

- (1) 存在二次总线延迟，速度受到影响。
- (2) 控制逻辑的负担增加。
- (3) 如果总线传输延迟与通讯信号脉宽的比值不合适，就可能出现在下一个“数据准备”到达目的端时，上一个“数据接收”仍处于高电平，这样“数据准备”信号会使“数据接收”线一直维持在高电平上，从而造成下一个数据的接收错误。

(4) 双向源控互锁

消除上述错误的方法是：使下一个“数据准备”信号在上一个“数据接收”信号结束后再发出。如图 5.14 所示，只有当“数据接收”的后沿返回到源端后，下一个“数据准备”信号才可以发出。

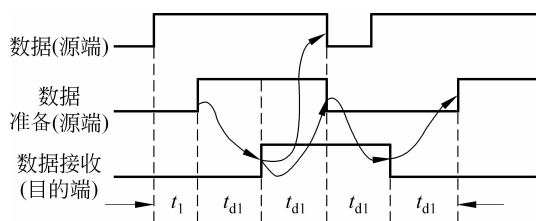


图 5.14 双向源控互锁方式

虽然双向互锁方式增加了信号沿总线来回传送的次数,也使控制硬件复杂些,但它既能适应各种 I/O 设备的不同速度,保证数据传送的正确性,又能有较高的数据传送速率。因为它总是以所接源和目的部件中相对较低的速率来通信,而不是所有部件中最低的速率,这比同步方式效率要高。所以,目前 I/O 总线中最广泛使用的还是异步双向互锁的通信方式。

5.2 中断系统

5.2.1 基本概念

(1) 中断源: 能引起各种中断发生的事件称为中断源。中断源可来自系统内部和系统外部。

- 系统内部: 如机器掉电、硬件故障、程序性出错。
- 系统外部: 如它机告急、外来的各种突时中断请求、用户需求进入管态等。

(2) 中断请求: 中断源向中断系统发出请求中断的申请,称为中断请求。同时可能存在多个中断请求,这时中断系统需要按事先确定的中断响应优先次序对优先级高的中断请求予以响应。

(3) 中断响应: 对某个中断请求来讲,允许其中断 CPU 现程序的运行,转去对该请求进行预处理,包括保护好断点现场,调出有关处理中断的中断处理程序,准备运行,这便是中断响应。这部分工作在大多数机器上都是采用交换新旧程序状态字 PSW 的办法实现的。

(4) 中断响应时间: 从发生中断请求到进入中断处理程序的时间。

(5) 中断系统: 中断装置和中断处理程序统称为中断系统。中断系统是计算机的重要组成部分,实时控制、故障自动处理、计算机与外围设备间的数据传送往往多用中断系统,中断系统的应用大大提高了计算机效率。

(6) 断点: 发生中断时被打断程序的暂停点称为断点。

(7) 中断处理程序: 中断服务程序、中断子程序、完成请求的处理程序称为中断处理程序。这些程序事先被存放在存储器中,需要时由 CPU 取出并执行。

在图 5.15 中, a、b、c 是三个程序断点, t_0 、 t'_0 、 t_1 、 t'_1 、 t_2 、 t'_2 和 t_3 是不同程序段被执行时所占用的 CPU 时间。

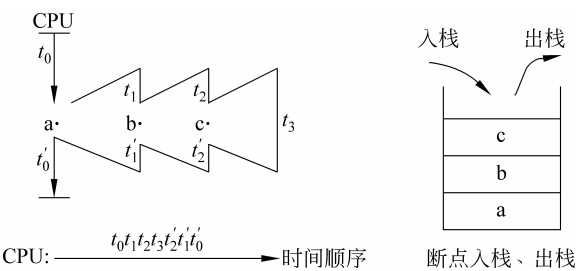


图 5.15 中断处理程序在时间上的执行顺序

5.2.2 中断分类

1. 硬件中断与软件中断

- (1) 硬件中断是指由某个硬件中断请求信号引发的中断,它通过中断请求信号形成向量地址。
- (2) 软中断(又称软件中断)是由执行软中断指令 INT 而引起的中断,它由指令提供中断号 n ,再由 n 转换为向量地址。软中断发生于 CPU 内部,它不属于外部中断。

2. 强迫中断与自愿中断

- (1) 强迫中断是由于故障、外部请求等所引起的中断,非程序本身安排,往往是随机的。
- (2) 自愿中断也称程序性自中断,即软中断,是程序有意安排的,且可以随机插入。

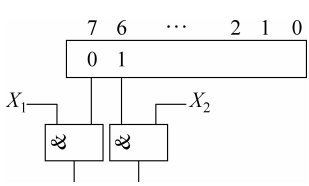
3. 内中断与外中断

- (1) 内中断是指来自主机内部的中断请求,如掉电中断、CPU 故障中断、软件中断。
- (2) 外中断是指来自主机外部的中断请求,如时钟中断、键盘中断、显示器中断、打印机、磁盘中断。

4. 可屏蔽中断与不可屏蔽中断

- (1) 可屏蔽中断 INTR(一般的外围设备)。
- (2) 非屏蔽中断 NMT(掉电、软中断)。

对于可屏蔽中断的实现方式可采用图 5.16 所示的借助于“与门”逻辑站实现,通过两个“与门”将字节中的第 6、7 位的中断信号进行有效的屏蔽。例如当 $X_1 X_2 = 01$ 时,屏蔽第 7 位,开放第 6 位。



5. 向量中断与非向量中断

如何根据中断源获得服务程序的入口地址,这个过程如果是直接依靠硬件实现的,则属于非向量中断。如果是通过软件查询方式实现的则属于向量中断。

6. 响应中断的条件

- (1) 有中断请求信号发生。
- (2) 该中断请求信号未被屏蔽。

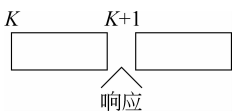


图 5.17 响应中断的时机

- (3) CPU 处于开中断状态,即允许中断触发器 $T_{EN}=1$ 。
- (4) 没有更重要的事要处理(即没有比它更高优先级的请求信号发生)。
- (5) CPU 刚执行完的指令不是停机指令。
- (6) 在一条指令结束时给予响应,如图 5.17 所示, K 与 $K+1$ 是两条相邻的指令。

5.2.3 中断服务程序入口地址的获取方式

(1) 中断向量表: 将所有中断服务程序的入口地址组织成一个一维表格,存放在一段连续区域中。向量表可以提供 256 个中断源。256 个中断类型形成的中断向量表如图 5.18 所示。

中断类型	向量地址	中断向量表		
0型	00000H	偏移地址	}	专用区
	00001H			
	00002H	段地址		
	00003H			
1型	00004H			
	00023H			
	00024H			
				系统保留区
	0007FH			
	00080H			
255型				用户扩展区
	003FFH			

图 5.18 中断向量表

(2) 向量地址(向量指针): 是访问中断向量表的地址码。

中断类型码 $\times 4$ 生成向量地址,根据向量地址从中向量表中读出中断服务性程序的段地址和偏移地址并生成物理地址,依据物理地址访主存读取指令,从而转向服务性程序。

5.2.4 中断处理程序的优先级别

1. 中断系统分类与分级的原因

分类原因: 简化硬件。

分级原因: 因同类的中断源也有很多,需进行优先次序的判定。

优先级别最高的为 1 级,以此类推,1 级 $\rightarrow$ 2 级 $\rightarrow$ 3 级 $\rightarrow$ 4 级 $\rightarrow$ 5 级。

通常在应用中响应中断的级别从高到低的优先次序是紧急的机器校验、管理程序调用和程序性、可抑制的机器校验中断、外部中断、I/O 中断,最后是重新启动中断。