

第5章

输入输出系统组织

输入输出系统是计算机重要的组成部分之一。简单地说,输入输出系统的主要功能是完成程序和数据的输入及机器运行结果的输出。随着计算机应用的发展,计算机能处理的数据种类越来越多,如简单的数值数据、文本数据、图形图像数据及音视频数据等各种多媒体数据,这对现代计算机的输入输出系统提出了更高的要求。

本章首先介绍计算机输入输出系统的组成,然后对计算机输入输出的控制方式进行详细讨论,最后介绍计算机存储设备——磁盘系统以及由磁盘阵列组成的 RAID 技术。

5.1 输入输出系统概述

输入输出系统主要由输入输出设备和与 CPU 相连接的输入输出接口逻辑组成,CPU 通过接口逻辑实现与输入输出设备之间的数据传输。

5.1.1 输入输出设备

输入输出设备又称外围设备,它涵盖的面非常广。事实上,除了 CPU 和主存外,计算机系统的其他部件都可看成外围设备。

外围设备种类繁多,随着计算机应用的发展,不断有新的外围设备推向市场。一般来讲,计算机的外围设备可以分为以下 5 类。

1. 输入设备

输入设备是指完成人机数据输入的外围设备。人们设计、建造和使用计算机的目的是让计算机帮助人们更高效地完成一定的功能。早期的计算机最主要的应用是科学计算,现代计算机的应用则深入到人们生活的方方面面。不管人们要求计算机完成什么样的功能,首先需要将所编制的程序和处理的数据输入到计算机中,而程序和数据等的输入就需要输入设备来完成。早期的计算机是使用纸带读入机等来完成程序和数据的输入的,键盘和鼠标是目前计算机标准配置的输入设备。随着现代计算机在多媒体信息领域的应用,不断涌现出各种用于多媒体信息输入的设备,如扫描仪、游戏控制杆、语音输入设备、图像输入设备以及各种音视频输入设备等。

2. 输出设备

输出设备是指完成人机数据输出的外围设备。早期用于科学计算的机器需要通过输出设备将程序运行的结果以某种方式输出给程序员,如通过显示器将结果显示在显示屏上,或通过打印机将结果打印在打印纸上等。而现代计算机的输出设备种类越来越丰富,如用于绘制工程图纸的绘图仪、MPC 的音箱设备以及各种音视频输出设备等。

3. 存储设备

第4章中讲到现代计算机存储器采取层次结构实现,即由“cache-主存-辅存”层次组成。按照所采用的存储介质分类,现代计算机辅存主要包括磁介质存储器(磁盘存储器和磁带存储器)和光盘存储器。可执行程序在运行前首先通过输入设备输入,并以文件的形式存放在辅存中。另外,计算机处理的各种信息(数据、文本、图形、图像、声音、视频、动画等)也均以文件的形式存放在辅存中。

由磁盘存储器、磁带存储器或光盘存储器构成的辅存就是计算机的存储设备。在机器中,它们是作为设备由操作系统进行管理的。有关磁盘存储器、磁带存储器或光盘存储器的内容将在5.3节详细讲述。

4. 数据通信设备

计算机与计算机之间的数据通信有多种方式。例如,早期的计算机通过模拟电话线实现计算机之间的点到点通信,通信双方要使用数据通信设备——调制解调器——完成数字信号与模拟信号的转换。现代计算机主要通过网络进行数据通信,每台计算机都是通过使用数据通信设备——网络适配器(又称网卡)——联入网络的。正是借助各种数据通信设备,才实现了计算机之间的互联。

5. 过程控制设备

计算机的一个重要应用领域就是工业控制。工业控制中通常需要使用一些过程控制设备完成对某一工业过程的控制。与上述列举的输入输出设备不同的是,这些过程控制设备往往是针对一特定应用专门设计的,属于计算机系统的非标准外围设备。

5.1.2 输入输出接口

由5.1.1节的内容可知计算机外围设备种类繁多,而且互相之间差异很大,这些差异主要体现在以下5方面。

(1) 物理特性方面。物理特性方面的差异主要是指设备连接的方式和读写驱动方式等,如接口的类型、机械尺寸、信号线的条数以及排列等。例如,打印机有针式打印机、激光打印机、喷墨打印机等,不同打印机的打印方式各自不同,对打印头和纸张移动的驱动方式也有所不同。

(2) 电气特性方面。数据在信号线上传递时,是以一定的电平值来表示二进制数字0和1的。电气特性定义每一条信号线的传递方向和有效电平范围。有的设备的电信号使用的是TTL电平标准,有的设备使用的是CMOS电平标准。这两者是不兼容的,其中TTL

电路电源电压使用的是 5V, 而 CMOS 电路电源电压使用的是 12V。另外, 两者在表示逻辑 0 和逻辑 1 的电平值也是不相同的。

(3) 功能特性方面。功能特性定义了设备连接的每一条信号线的功能, 如用于传递数据的信号线、用于传递地址的信号线、用于传递控制的信号线等。尤其对于种类繁多的外围设备来说, 不同设备所需的控制信号也各不相同。有的设备与主机之间采用中断传送方式, 需要中断控制信号; 有的设备与主机之间采用 DMA 传送方式, 需要相应的 DMA 控制信号等。

(4) 数据格式方面。外围设备与主机之间的数据传送主要分为两种形式: 串行传送和并行传送。串行传送是一位一位地进行, 而并行传送则是多位同时进行。不同的设备并行传送的位数也会有所不同。

(5) 传输速度方面。不同外围设备在速度上的差异是非常明显的。有的设备的数据传输速率高达每秒几百兆字节(如磁盘), 而有的设备则只有几十字节甚至更低。例如, 键盘与 CPU 之间传输数据的速度取决于人的手指敲键的速度, 一个键对应一个字节的数据。如果每秒敲 10 个键, 数据传输速率也才 10B/s。

作为计算机硬件的核心——CPU, 从功能上讲, 它能够完成对所有其他硬件部件的控制。这种控制功能的实现是通过 CPU 芯片的引出脚与其他硬部件连接, 并由 CPU 执行的指令产生的各种控制信号来完成的。但是, 从上面讨论的有关设备之间的差异可以看出, 一个 CPU 在设计时, 无法做到与现在已有的和将来新开发出来的各种不同的外围设备直接相连。因此, 计算机在设计时, 针对与一些标准输入输出设备(这些设备往往是计算机的标配设备, 如输入设备——鼠标键盘、输出设备——显示器、存储设备——硬盘或光驱等)的连接, 专门在主机板上设计了相应的接口电路, 使 CPU 可以通过这些接口电路实现对这些设备的控制。而一些计算机非标准配置的设备(如各种多媒体信息输入输出设备、过程控制设备等), 或者通过计算机的一些标准接口(如串口、并口、USB 接口等)连接, 或者使用专门的设备控制适配器与 CPU 连接, 如图 5-1 所示。

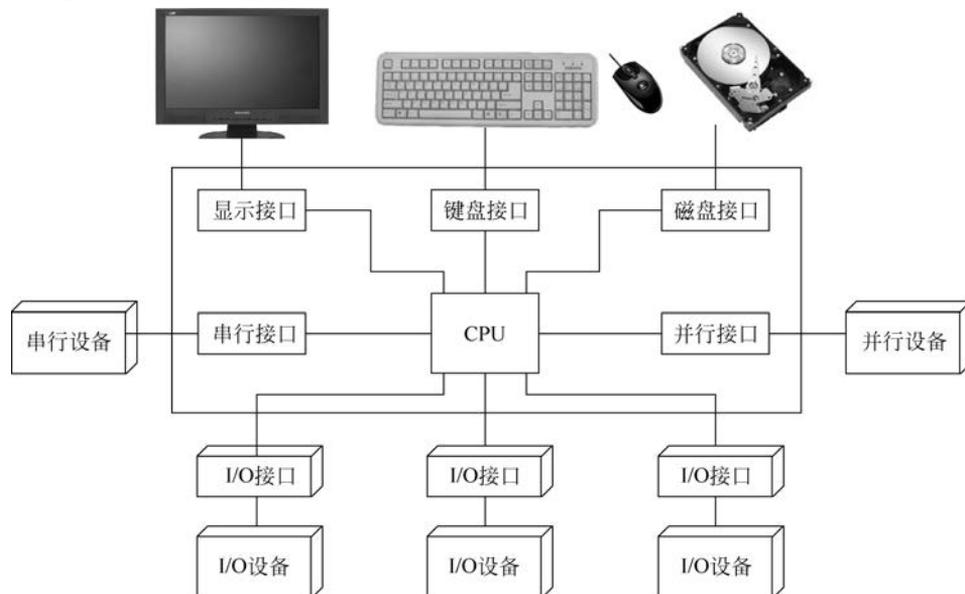


图 5-1 输入输出设备与接口

无论是与计算机标准输入输出设备连接的标准接口还是与计算机非标准配置设备连接的专用接口,统称为计算机的输入输出接口,简称 I/O 接口。也就是说,CPU 是通过 I/O 接口与各种外围设备连接的,并通过 I/O 接口实现对外围设备的控制功能。

1. I/O 接口的组成结构

不同设备的 I/O 接口不仅在组成结构上有所不同,复杂程度往往也差异很大。但一般来说,从 I/O 接口的内部结构看,大多数 I/O 接口中都包含有一些数据寄存器、地址寄存器、状态寄存器、控制寄存器和相应的控制电路。而且,为了控制的灵活性和适应性,很多 I/O 接口中的寄存器往往是可编程的,即可以对 I/O 接口的功能、工作方式、操作方式、数据格式等进行预设置,以满足不同应用场合的要求。I/O 接口与 CPU 及设备连接的结构图如图 5-2 所示。

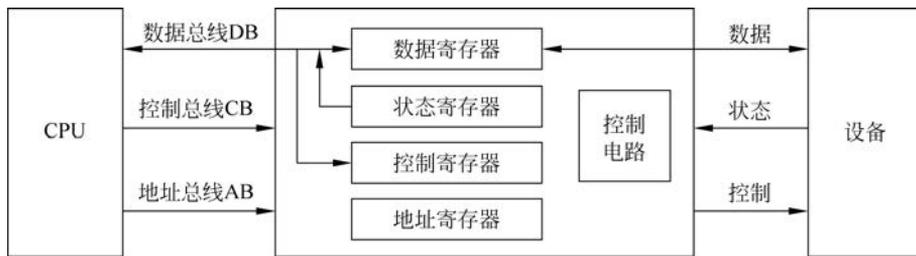


图 5-2 I/O 接口与 CPU 及设备连接的结构图

(1) 数据寄存器。数据寄存器用于保存 CPU 与设备之间交换的数据,又可以分成数据输入寄存器和数据输出寄存器。当接口电路连接输入设备时,需要从输入设备获取数据;数据从输入设备出来就暂时保存在数据输入寄存器,由 CPU 选择合适的方式进行读取。同样,当接口电路连接输出设备时,CPU 发往输出设备的数据被临时保存在数据输出寄存器中,然后到达输出设备。很多设备既可以输入、又可以输出,可使用同一个寄存器或不同寄存器与 CPU 交换数据。

在 I/O 接口中设置数据寄存器一方面可以暂存 CPU 与设备之间交换的数据,另一方面可以匹配 CPU 与设备之间的传输速度差异。例如,CPU 向一个慢速设备输出数据时,每次先将数据送至 I/O 接口中的数据寄存器,等到 I/O 接口中的数据被传送给设备后,再向 I/O 接口传送下一个数据。另外,I/O 接口中的数据寄存器还起到 CPU 与设备间数据格式的转换工作。例如,如果与 CPU 连接的是一个串行输出设备,则 CPU 每次将一个并行数据(字节或字)先送至 I/O 接口数据寄存器,由 I/O 接口控制将数据寄存器中的数据一位一位送往设备。

(2) 状态寄存器。状态寄存器用于保存设备当前的工作状态信息。CPU 与设备交换数据时,常常需要了解设备当前的工作状态,以便决定是否继续与设备进行数据的交换。例如,CPU 与慢速设备进行数据交换时,若 CPU 只顾按自己的速度发送,而不管设备是否有能力接收,则会导致传输数据的丢失。通过在 I/O 接口中设置状态寄存器,及时地将设备的状态反映给 CPU,CPU 能够按照设备的接收能力或速度传输数据。

(3) 控制寄存器。控制寄存器用于保存 CPU 对 I/O 接口电路和设备进行控制或操作的相关信息。对于可编程的 I/O 接口来说,常常有多种工作方式或操作模式等可以选

择。CPU 可通过向控制寄存器写入相应的控制字实现对 I/O 接口及设备工作方式等的预设置。

(4) 地址寄存器。有些 I/O 接口中还设置了地址寄存器。例如,在 CPU 与磁盘采用 DMA 数据传送方式的 DMA 接口电路中就设置了地址寄存器,其初始值由 CPU 预置,指出与磁盘进行块数据交换的主存空间首地址,然后在 DMA 接口(又称 DMA 控制器)的控制下完成块数据交换。

2. I/O 接口的功能

I/O 接口根据其复杂程度,实现的功能各不相同。一般来讲,I/O 接口的功能包括以下几个方面。

(1) 数据的寄存和缓冲。I/O 接口中设置的数据寄存器,一方面起到数据暂存的作用,另一方面起到数据缓冲的作用。I/O 接口的数据缓冲用于匹配快速的 CPU 与相对慢速的设备之间的数据交换。例如,很多打印机通过计算机的并行接口与主机相连,在主机板的并行接口电路里设置了数据缓冲寄存器,同时打印机的内部控制电路也设计有一个数据缓冲区;CPU 传送来的打印数据首先被送到并行接口电路的数据寄存器中缓存,然后再传送到打印机的数据缓冲区;打印机的控制电路控制打印机按照一定的打印速度从数据缓冲区中取出数据打印。

(2) 对设备的控制和监测。一方面,I/O 接口能够接收来自 CPU 的控制命令,进行解释分析,生成对设备的控制信号;另一方面,I/O 接口能够对设备的工作状态进行实时监控,并随时将设备的状态提供给 CPU 查询等。

(3) 对设备的寻址。一个计算机系统中的外围设备很多,CPU 对设备的访问也是通过地址的方式进行的。在 I/O 接口电路里,除了一些寄存器和控制电路外,还往往有地址译码电路。CPU 通过 I/O 接口中的地址译码电路完成对设备的寻址和访问。

(4) 信号变换。外围设备大都是复杂的机电设备,其所需的控制信号和所能提供的状态信号往往同 CPU 的引脚信号不兼容,这包括信号使用的电平标准、信号的时序逻辑关系及数据格式等。因此,I/O 接口必须完成 CPU 与设备之间的这种信号转换功能,使 CPU 与设备之间能进行正常的连接和通信。

在微型计算机系统中,有些 I/O 接口是以标准芯片的方式提供给用户使用的,它们往往具有可编程功能,以满足不同用户在不同应用场合的需要。

5.1.3 输入输出设备的编址与管理

第 4 章讲到 CPU 对存储器的组织管理采用的是编址的方式,即为每个存储单元分配一个唯一的地址,CPU 通过给出单元地址访问相应的存储单元。在一个计算机系统中,所拥有的各种输入输出设备和存储设备很多,这些设备以及设备的 I/O 接口中有很多可供 CPU 访问的寄存器,这些不同种类的寄存器称为 I/O 端口。CPU 对 I/O 端口的访问采用的是与访存类似的按地址访问方式,即为每一个 I/O 端口分配一个地址,又称为 I/O 地址或 I/O 端口号。CPU 通过给出 I/O 端口地址访问相应的 I/O 端口,也即访问相应的设备。

CPU 对 I/O 端口的编址方式主要有两种:一是独立编址方式,二是统一编址方式。

1. I/O 端口的独立编址方式

独立编址方式是指系统使用一个不同于主存地址空间之外的单独的一个地址空间,为外围设备及接口中的所有 I/O 端口分配 I/O 地址,如图 5-3(a)所示。

IBM PC 等系列计算机设置有专门的 I/O 指令,设备的编址可达 512 个,部分设备的地址码如表 5-1 所示。每种设备占用了若干个地址码,分别表示相应设备控制器中的各个寄存器地址。各个寄存器的地址码不能重复,具有唯一性。

表 5-1 输入输出设备地址分配表

输入输出设备	占用地址数	地址码(十六进制)
硬盘控制器	16	320~32FH
软盘控制器	8	3F0~3F7H
单色显示器/并行打印机	16	3B0~3BFH
彩色图形显示器	16	3D0~3DFH
异步通信控制器	8	3F8~3FFH

在这种方式下,CPU 指令系统中有用于与设备进行数据传输的、专门的输入输出指令,对设备的访问必须使用这些专用指令进行。例如,80x86 系列的 CPU 有两条专用输入输出指令,如下所示。

(1) IN AL,PORT; 输入指令,用于将 I/O 端口(Port)中的内容传送到 AL 寄存器中。

(2) OUT PORT,AL; 输出指令,用于将 AL 寄存器中的内容传送到 I/O 端口(Port)中。

执行 IN 指令时,CPU 的 I/O 读取 $\overline{\text{IOR}}$ 信号有效,产生 I/O 读总线周期;执行 OUT 指令时,CPU 的 I/O 写 $\overline{\text{IOW}}$ 信号有效,产生 I/O 写总线周期。

独立编址方式的优点:一是 I/O 端口的地址没有占用主存的地址空间;二是 I/O 端口的地址码较短,地址译码器设计及实现简单,译码时间也较短。其缺点是:只能使用专用输入输出指令访问 I/O 设备,对 I/O 设备操作的程序设计灵活性较差。

2. I/O 端口的统一编址方式

统一编址方式是指 I/O 端口与主存单元使用同一个地址空间进行统一编址,如图 5-3(b)所示。



图 5-3 I/O 端口的编址方式

在这种方式下,CPU 指令系统中无须设置与设备进行数据传输的、专门的输入输出指令,I/O 端口被当成主存单元同样对待,对主存单元进行访问和操作的指令可以同样用于对 I/O 端口的访问和操作。

统一编址方式的优点:可以使用访存指令访问 I/O,对 I/O 设备操作的程序设计灵活性较好。其缺点是:I/O 端口的地址占用了主存的部分地址空间,对 I/O 端口访问的地址译码更加复杂。

5.2 输入输出控制方式

从前面的讨论大家已经了解到,不同的设备在很多方面都存在差异。因此,作为机器的输入输出系统来说,必须针对不同的设备采取不同的输入输出控制方式,以最大程度发挥主机 CPU 和设备的效率。

5.2.1 程序控制方式

程序控制方式是指主机与设备间的数据传输通过 CPU 执行一段软件程序,在程序的控制下完成输入输出操作。根据设备的不同,程序控制方式又分为无条件传送控制方式和查询式传送控制方式。

1. 无条件传送控制方式

有些简单设备,如发光二极管(light-emitting diode,LED)和数码管、按键及开关等,它们的工作方式十分简单,相对 CPU 而言,其状态很少发生变化。例如,对于数码管,只要 CPU 将数据传给它,就可立即获得显示;又如电子开关,其状态或者为闭合,或者为打开,CPU 可以随时对其状态进行读取。因此,当这些设备与 CPU 交换数据时,可以认为它们总是处于就绪(ready)状态,随时可以进行数据传送。这就是无条件传送,有时也称为立即传送或同步传送,其程序控制流程如图 5-4 所示。

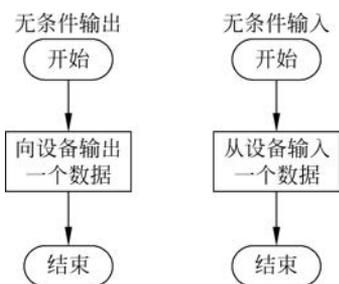


图 5-4 无条件传送程序控制流程

用于无条件传送的 I/O 接口电路十分简单,接口中只考虑数据缓冲,不考虑信号联络。实现数据缓冲的器件可以是三态缓冲器或锁存器。

图 5-5 给出了一个无条件输入输出的接口例子,其中无条件输入接口电路连接一个开关电路,无条件输出接口电路连接 LED。

三态缓冲器 74LS244 构成数据输入端口,它连接 8 个开关 $K_0 \sim K_7$,开关的输入端通过电阻连到高电平上,另一端接地。当开关打开时,缓冲器输入端为高电平(逻辑 1);开关闭合时,缓冲器输入端为低电平(逻辑 0)。

当 CPU 执行输入指令时,产生读控制信号 \overline{IOR} 低有效。译码输出和读控制同时低有效,使三态缓冲器控制端低有效;开关的当前状态被三态缓冲器传输到数据总线 $D_7 \sim D_0$,进而传送给 CPU,其中某位 $D_i = 0$,说明开关 K_i 闭合; $D_i = 1$,说明开关 K_i 断开。

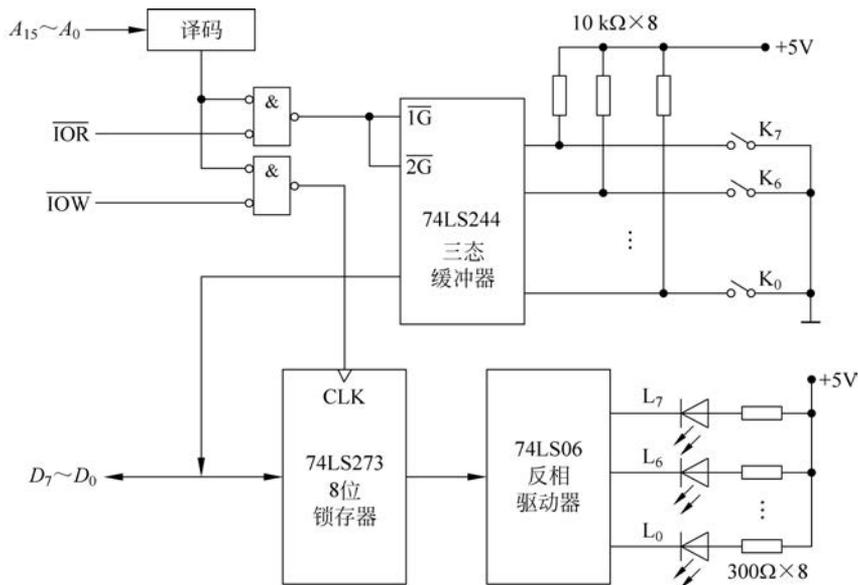


图 5-5 无条件输入输出接口

8 位锁存器 74LS273(无三态控制)构成数据输出端口。当其时钟控制端 CLK 出现上升沿时锁存数据,被锁存的数据输出后经反相驱动器 74LS06 驱动 8 个发光二极管($L_7 \sim L_0$)发光。74LS06 是集电极开路(open collector, OC)输出,它的每条输出线通过电阻挂到高电平上。当 CPU 的某个数据总线 D_i 输出高电平(逻辑 1)时,经反相为低电平接到发光二极管 L_i 的负极,发光二极管正极接高电平,使二极管形成导通电流,发光二极管 L_i 将点亮。当 CPU 输出 D_i 为低电平时,对应的发光二极管 L_i 不会导通,将不发光。

当 CPU 执行输出指令时,产生写控制信号 \overline{IOW} 低有效。译码输出和写控制同时低有效,使 8 位锁存器控制输入 CLK 为低;经过一个时钟周期,译码输出或写控制无效将使 CLK 恢复为高。在 CLK 的上升沿,8 位锁存器将锁存此时出现在其输入端、由 CPU 传送到数据总线 $D_7 \sim D_0$ 上的输出数据,控制发光二极管的发光和不发光。

2. 查询式传送控制方式

查询式传送控制方式也称为异步传送,它是指当 CPU 需要与外围设备交换数据时,首先查询设备的状态,只有在设备准备就绪时才进行数据传输。与无条件传送方式相比,查询式传送增加了一个传送前查询设备状态的环节。查询式输入和输出的程序控制流程如图 5-6 所示。

(1) 查询式输入接口。图 5-7 所示是一个采用查询式传送控制方式的输入数据的 I/O 接口电路示意图。图中 8 位锁存器与 8 位三态缓冲器构成数据输入寄存器,即数据端口。它一侧连接输入设备,另一侧连接系统的数据总线。1 位锁存器和 1 位三态缓冲器构成状态寄存器,即状态端口,其输出端连接到数据总线的最低位 D_0 。

在输入设备通过选通信号 \overline{STB} 将数据送入数据寄存器的同时,将 D 触发器置 1(因为其输入端 D 总是为高电平),说明数据寄存器中已经有设备送来的数据,可以提供给 CPU 读取,也就是表示设备就绪。

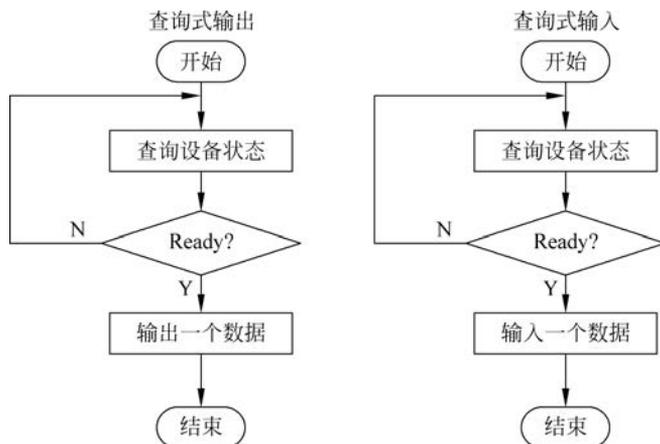


图 5-6 查询式传送程序控制流程

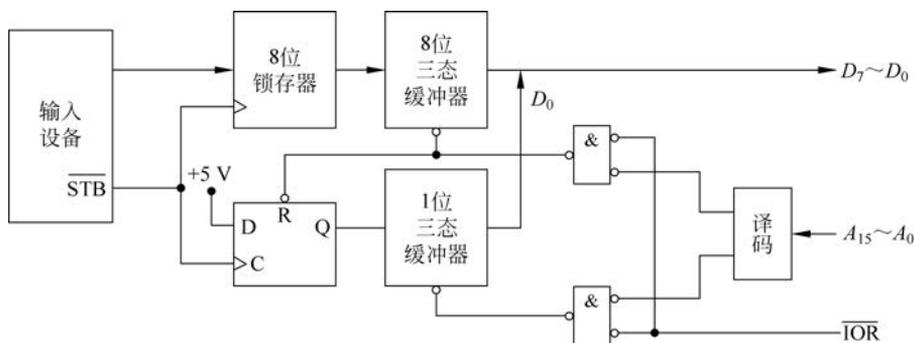


图 5-7 查询式输入接口

CPU 可以随时通过读取状态端口来查询设备的状态。如果 $D_0 = 1$, 说明输入数据已就绪, 此时, CPU 读取数据端口便可得到设备提供的数据。读取数据产生的控制信号还被连接到 D 触发器的 R 复位端 (低电平有效), 该复位信号将触发器输出 Q 置为 0, 表示数据已被取走。如果检测到 $D_0 = 0$, 说明输入数据尚未就绪, 程序应继续查询。

(2) 查询式输出接口。图 5-8 所示是一个采用查询式传送控制方式的输出数据的接口电路示意图。8 位锁存器构成数据输出寄存器, 即数据端口, 它一侧连接系统的数据总线, 另一侧连接输出设备。1 位锁存器和 1 位三态缓冲器构成状态寄存器, 即状态端口, 其输出端连接到数据总线的最高位 D_7 。

当 CPU 要输出数据时, 首先查询状态端口。若 $D_7 = 0$, 表示设备就绪, 可以接收数据, 此时 CPU 可将数据写入数据端口。写入数据产生的控制信号也作为 D 触发器的控制信号, 它将 D 触发器置 1, 以便通知外围设备接收数据。若 $D_7 = 1$, 说明接口电路中的上一个数据尚没有被设备取走, CPU 不能送下一个数据, 只能继续查询设备状态, 直到设备准备就绪。

输出设备可根据状态锁存器的输出信号 Q 接收数据。若为 1, 表示 CPU 已送来一个数据, 设备可以从数据寄存器中将数据取出, 并待数据处理结束时, 给出应答信号 $\overline{\text{ACK}}$ 。该信号将状态寄存器重新复位为 0, 表示设备准备就绪, 可以接收下一个数据了。

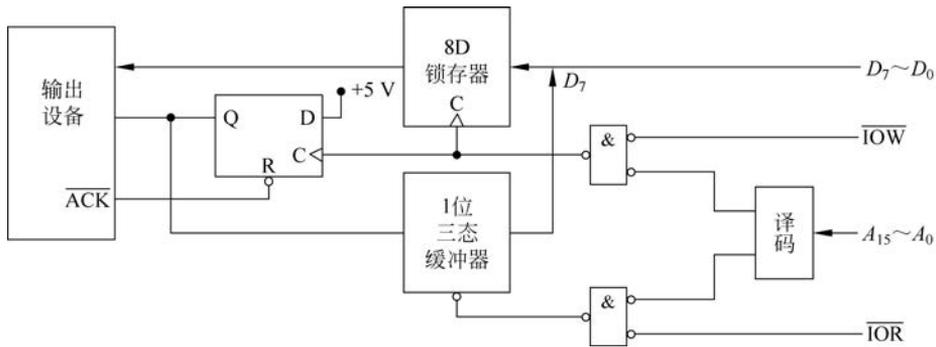


图 5-8 查询式输出接口

5.2.2 中断控制方式

上述的程序控制方式对于单用户单道程序系统环境下对简单设备的输入输出控制是有效的。在这种环境下,用户程序可以占用全部系统资源,包括 CPU、存储器和输入输出设备。当用户程序需要进行输入输出操作时,CPU 执行用户的输入输出程序,控制与设备的数据交换。但是,在多用户多道程序系统中,若仍采用这种控制方式,则某一道程序需要占用 CPU 进行输入输出操作,而其他程序只能等待,这对 CPU 这一宝贵资源将是极大的浪费。尤其是当 CPU 控制与慢速设备的数据交换时,一方面 CPU 大部分时间处于等待设备准备就绪的空置状态,另一方面其他程序因得不到 CPU 的资源而不能运行,使 CPU 的有效利用率很低。为解决这种矛盾,计算机设计者提出了中断控制方式。

1. 中断的基本概念

老师正在教室上课,一位迟到的学生推门进来,老师停下正在讲的课,目送着学生在座位上就坐,然后继续上课,这一过程就是中断。

对于计算机系统来说,中断是指 CPU 正在运行一个程序时发生了某种非预期的事件,不得不暂停正在运行的程序,转而执行对这一事件进行处理的程序(称为中断服务程序),完成后再返回原程序继续运行的过程,如图 5-9 所示。

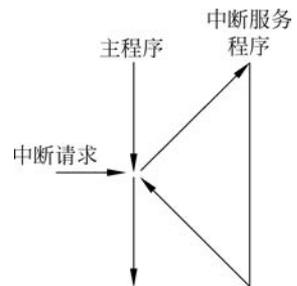


图 5-9 中断的概念

引起中断的事件称为中断源。对于计算机系统来说,中断源分为外部中断和内部中断。外部中断主要是指由计算机外围设备、系统定时时钟及人工干预等外部事件引起的中断设备产生的外部中断能使 CPU 与设备间进行中断方式的数据传输,这也是本小节主要讲述的内容。内部中断主要包括指令中断和故障中断。指令中断是由软件指令引起的中断。设置指令中断的目的通常是为用户程序提供对系统资源的访问,例如,80x86 CPU 指令系统提供了一条软中断指令 INT,执行该指令后,系统会转入执行一段驻留在主存中的系统程序。该程序主要完成对系统某一资源的访问服务。故障中断主要是指由系统软硬件故障引起的中断,如内存校验故障、电源掉电、除零错误、算术溢出、内存越界、指令非法、虚拟存储器页面失效等。计算机中断源的类型归纳总结如图 5-10 所示。

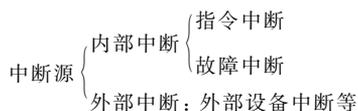


图 5-10 计算机中断源的类型

2. 中断控制的基本原理

前面已经提到,程序控制方式在多道程序系统中会因为某一程序长时间占用 CPU 进行输入输出操作而浪费 CPU 资源,而中断控制方式则可以有效提高 CPU 利用率。下面通过对比程序控制方式和中断控制方式下 CPU 控制打印机打印输出的过程,阐述中断控制的基本原理。

图 5-11(a)和图 5-11(b)分别给出了程序控制方式和中断控制方式这两种方式下打印机的打印输出过程。

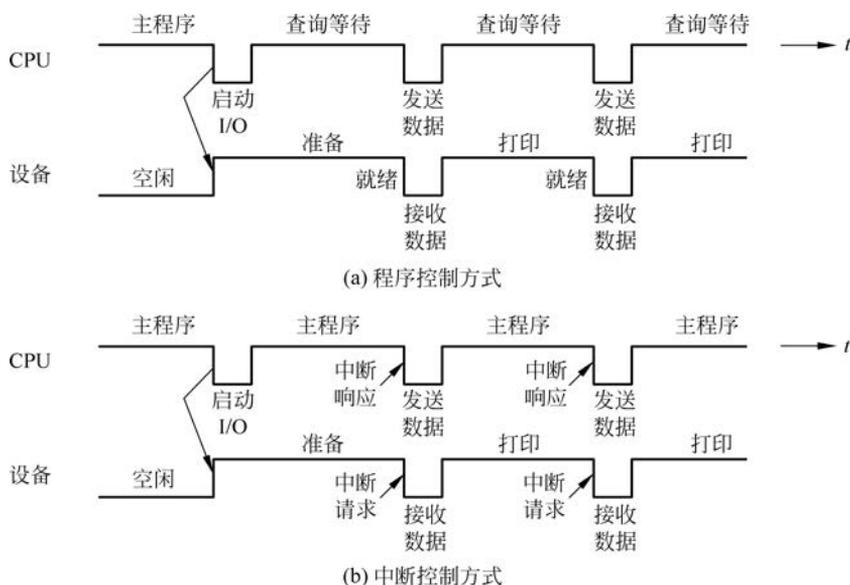


图 5-11 程序控制方式和中断控制方式下打印输出的过程对比

从图 5-11(a)可以看出,在程序控制方式下,当 CPU 执行的一个主程序要打印输出数据时,首先启动打印机,使打印机进入打印准备工作状态;在打印机准备的过程中,CPU 一直处于查询等待的状态,即查询打印机是否准备就绪;当 CPU 查询到打印机已做好打印准备时,便向打印机传送第一个数据,打印机接收到该数据后进行打印;在打印机打印的过程中,CPU 又开始进入查询等待状态;等到打印机打印完一个数据可以接收下一个数据时,CPU 再向打印机发送下一个数据;如此重复,直到将所有数据打印完。一般来讲,CPU 向打印机传送一个数据的时间远比打印机打印一个数据所花的时间少得多。而在上述过程中,当打印机进行打印操作时,CPU 一直在执行一段查询打印机状态的循环控制程序,无法执行其他程序,这对 CPU 资源是极大的浪费。

而在中断控制方式下,情况就不一样了。如图 5-11(b)所示,当 CPU 执行的一个主程序要打印输出数据时,首先启动打印机,使打印机进入打印准备工作状态;在打印机准备的

过程中,CPU可以由系统调度去执行其他的主程序;当打印机做好打印准备时,向CPU发出一个中断请求信号;CPU接收到该请求后,暂停正在执行的主程序,向打印机传送一个数据,然后返回被中断的主程序继续执行;打印机接收到数据后进行打印,打印完成后,又向CPU发出中断请求;CPU响应中断,向打印机传送下一个数据进行打印;如此重复,直到将所有数据打印完。从这一过程可以看到,在打印机打印的同时,CPU可以被调度去执行其他主程序,而无须查询等待,使CPU的利用率得到了提高。

通过对以上两种不同控制方式控制打印机打印输出过程的比较,可以看出它们的不同之处。

(1) 在程序控制方式下,CPU是通过查询方式了解打印机的状态的;而在中断控制方式下,CPU是通过中断方式了解打印机的状态的。

(2) 在程序控制方式下,CPU和打印机之间是串行工作的;而在中断控制方式下,CPU和打印机可以并行工作。

(3) 程序控制方式对于单用户单道程序系统来说是有效的;而中断控制方式对于多用户多道程序系统来说可以大大提高CPU的利用率。

3. 中断处理过程

一个中断的处理过程是由中断源的中断请求引起的。在一个可实现中断系统功能的机器中,CPU在执行一个主程序时,每执行完一条指令都会检查是否有中断请求发生。若没有,则继续执行原程序;若有,则在条件满足的情况下,暂停正在执行的程序,对中断请求进行响应。在中断响应过程中,中断系统要识别是哪一个中断源发出的中断请求,在有多个中断源同时发出中断请求的时候还要决定首先响应哪一个中断源的请求。在确定了要响应的中断源后,进入对该中断源的中断请求进行处理的中断服务程序去执行,执行完成后返回原来被中断的程序继续执行。中断系统的中断处理过程如图5-12所示。

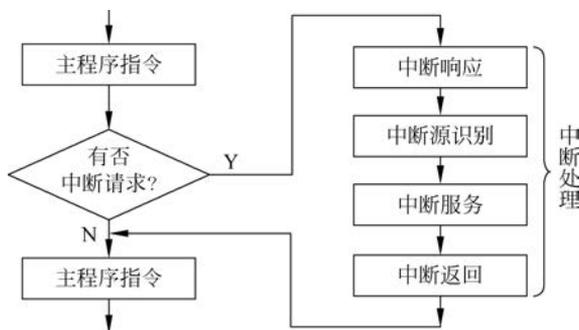


图 5-12 中断处理过程

(1) 中断请求的建立。CPU在设计时,其对外引脚通常会有一条或多条来自中断源的中断请求信号线,CPU通过这些引脚信号的状态来判断是否有外部中断源中断请求的发生。例如,8086/8088 CPU有两个引脚INTR和NMI,其中INTR(interrupt request)主要用于定时时钟和外围设备中断源的中断请求,NMI(non-masked interrupt)用于一些系统硬件故障中断源的中断请求等。

在中断系统中,外部中断源及硬件故障中断源的中断请求是由硬件实现的。当某中断

源要向 CPU 发出中断请求时,首先通过硬件方式为其建立和保持一个中断请求信号。通常是在其中断接口电路里设置一个“中断请求触发器”,当中断源有中断请求时(对设备来说就是设备准备就绪,对故障中断源来说就是发生了硬件故障),将接口中的中断请求触发器置位,中断请求触发器的输出将作为发往 CPU 的中断请求信号,如图 5-13(a)所示。

为了能灵活控制,通常在中断接口电路中还会设置一个中断屏蔽触发器,其作用是对中断源的中断请求进行屏蔽和开放。中断系统允许在程序中对中断屏蔽触发器进行设置,以决定在程序的执行过程中哪些中断源允许请求中断,哪些中断源不允许请求中断,如图 5-13(b)所示。

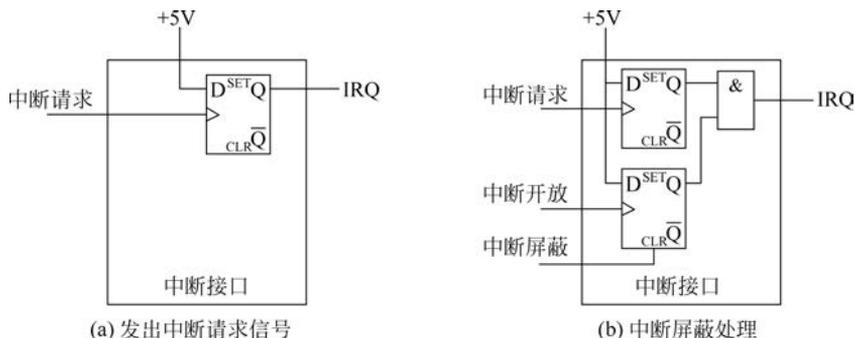


图 5-13 中断请求与中断屏蔽

一个机器系统中往往有多个中断源,将这些中断源的中断请求触发器合在一起会构成一个中断请求寄存器,而将这些中断源的中断屏蔽触发器合在一起会构成一个中断屏蔽寄存器。其中,中断屏蔽寄存器作为 I/O 端口可供 CPU 访问,在程序中通过将—个中断屏蔽字写入该端口,便可实现对中断源的屏蔽和开放功能。

(2) 中断响应。CPU 在其中断请求线上检测到外部中断源的中断请求时,并不是都会给予响应的。通常,大多数 CPU 内部都设置了一个中断允许触发器,该触发器的作用是对中断请求线上来的中断请求进行禁止和允许。中断允许触发器可以通过 CPU 提供的指令进行设置。例如,8086/8088 CPU 提供了两条指令 STI 和 CLI,前一条指令称为开中断指令,用于将 CPU 内部的中断允许触发器置 1,即中断允许或开中断;后一条指令称为关中断指令,用于将中断允许触发器清零,即中断禁止或关中断。在 CPU 处于中断允许状态时,可以对来自中断请求线上的中断请求进行响应;而 CPU 处于中断禁止状态时,则对来自中断请求线上的中断请求不予响应。

当然,并不是所有中断源的中断请求都能被 CPU 禁止,尤其是一些比较紧迫的事件的中断,如因电源掉电、内存校验出错等导致的硬件故障中断,CPU 必须立即响应。因此,多数 CPU 在引脚设计上会设置多条中断请求信号线,一些用于可屏蔽的中断请求,一些用于不可屏蔽的中断请求(如 8086/8088 CPU 的 INTR 和 NMI)。

一旦 CPU 响应了中断,便进入中断响应周期。在中断响应周期里,中断系统主要完成以下 3 项功能。

① 关中断和保护断点。CPU 响应中断时,会自动执行一条中断隐指令,一方面将中断允许触发器清零,即关中断;另一方面将 CPU 内部的指令指针和程序状态字 PSW 等压入堆栈。当前指令指针指向的是下一条要执行的指令,又称为断点。将指令指针压入堆栈的

目的是使在中断服务程序执行完成后能正确返回到当前被中断的程序的下一条指令继续执行。而程序状态字 PSW 记录的是当前指令执行完成后程序和机器的状态,将 PSW 压入堆栈的目的是保证中断返回后 PSW 能恢复成被中断前的状态。

② 进行中断源的识别。一个机器的中断源和 CPU 的中断请求线往往不是一一对应的。换句话说,CPU 的一条中断请求输入线会对应多个中断源的中断请求输出。当 CPU 检测到一条中断请求信号有效时,它可以判定外部中断源有了中断请求,但却无法确定具体是哪一个中断源发出的请求。因此,在中断响应周期中,CPU 要对发出中断请求的中断源进行识别。另外,在某一时刻,有可能同时有两个或两个以上的中断源向 CPU 发出中断请求信号,而 CPU 一次只能响应一个中断源的请求。在这种情况下,CPU 除了要进行中断源的识别外,还要根据一定的规则选择其中一个进行响应。

③ 形成中断源中断服务程序的入口地址。每一个中断源都对应有一段驻留在内存中的软件程序,称为中断服务程序。该程序的功能是完成中断源需实现的功能。例如,打印机是一个中断源,它所对应的中断服务程序的功能是实现 CPU 向打印机传输数据。在中断响应周期中,识别中断源后,最后还需形成该中断源对应的中断服务程序在内存中的入口地址,以便 CPU 从当前被中断的主程序转入中断服务程序执行。

(3) 中断源识别。中断源识别的任务是确定某次中断响应具体该响应的是哪个中断源。中断源识别的方法很多,常用的方法主要有软件查询法、硬件查询法和中断向量法等。

软件查询法是通过执行一段软件查询程序,对中断请求寄存器的状态进行逐位判断,从而确定某次该响应哪个中断源。前面讲到,将各中断源接口电路中的中断请求触发器合在一起便构成一个中断请求寄存器。也就是说,中断请求寄存器的每一位就对应了一个中断源的中断请求状态。将中断请求寄存器的内容读出,按某一种顺序一位一位进行判别,遇到第一个“1”,这一位所对应的中断源就是本次 CPU 识别响应的中断源。

软件查询法的优点:一是实现容易,无须额外的硬件;二是控制灵活,查询顺序很容易调整变化,这一顺序也决定了优先响应的中断源的顺序。其缺点是:速度慢,只适合一些对中断处理速度要求不是很高的场合。

硬件查询法是通过专门的硬件电路实现中断源识别的。一种实现中断源识别的串行排队链路如图 5-14 所示。

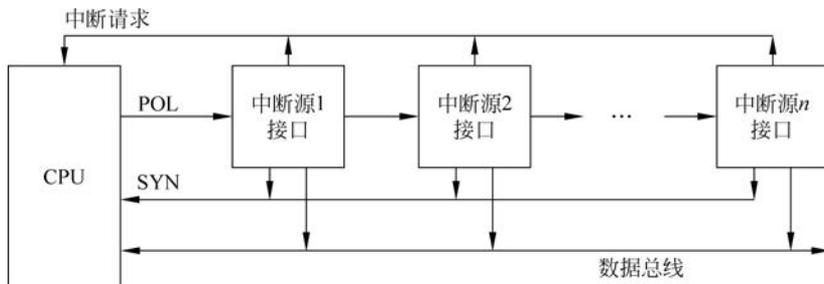


图 5-14 串行排队链中断源识别

在中断响应周期中,CPU 发出查询信号 POL,沿着串行排队链依次经过各中断源接口;当 POL 到达某一中断源接口时,如果该中断源没有中断请求,则将 POL 信号继续往下传;如果该中断源有中断请求,则 POL 信号不再往下传;接口向 CPU 发出回答信号 SYN,

同时形成中断源的中断服务程序入口地址,经数据总线传送给 CPU。

中断向量法是一种通过硬件控制电路形成一个所识别的中断源的中断向量号,并由此中断向量号实现中断响应的方法。在这种方法中,每个中断源对应一个中断向量号,中断向量号对应一个中断向量,即中断服务程序入口地址,并将所有中断向量集中存放在内存中的一片固定区域中。在中断响应周期中,首先由一个专门的中断控制电路进行中断识别,并形成一应该中断源的中断向量号;然后将此中断向量号传送给 CPU;最后由 CPU 依据中断向量号生成该中断源的中断向量在内存中的首地址,从这一地址单元中即可取出中断服务程序的入口地址,如图 5-15 所示。

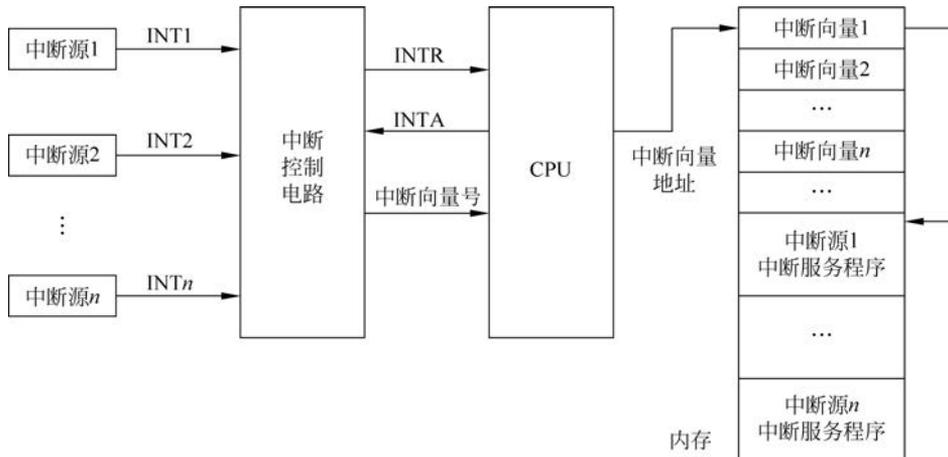


图 5-15 向量中断示意图

80x86 CPU 采用的就是中断向量法,其中断响应过程如下:

- ① 当某一中断源 i 需要申请中断时,向中断控制电路发出一个请求中断信号 INT_i ;
- ② 在该中断源的中断请求未被屏蔽的情况下,中断控制电路向 CPU 发出中断请求信号 INTR;

③ 在 CPU 处于开中断的情况下,CPU 响应中断,向中断控制电路发回一个中断响应信号 INTA;

④ 中断控制电路完成中断源的识别,并将中断源的中断向量号通过数据总线传送给 CPU;

⑤ CPU 依据此中断向量号计算得到中断向量地址,并从此地址单元中取出该中断源对应的中断服务程序入口地址。

(4) 中断服务。CPU 在中断响应周期获取到中断服务程序的入口地址后,便可转入中断服务程序执行。如图 5-16 所示,一般来讲,中断服务程序包括以下 7 个步骤。

① 保护现场。所谓现场,是指主程序执行完当前指令时的一些寄存器内容等。由于中断服务程序在执行过程中有可能用到主程序使用到的一些寄存器,因此必须在执行中断服务程序前将这些寄存器的内容保护起来。保护现场的具体方法就是将 CPU 所有程序可用的寄存器内容压入堆栈,以便从中断服务程序返回到主程序时再将这些寄存



图 5-16 中断服务程序

器的内容恢复到中断前的状态。

② 开中断。前面讲到,CPU 响应中断时执行了一次硬件自动关中断的操作,这次关中断的目的是阻止在保护现场的过程中再次被中断。如果本次执行的中断服务程序在后续的中断服务过程中允许其他的中断,在这里就必须开中断。开中断的方法是执行一条开中断指令,将 CPU 的中断允许触发器置位。

③ 中断服务。其实,中断处理的核心就是执行中断服务,在此之前和之后的所有操作都是为这里的中断服务提供支持的。中断服务的内容就是完成中断源的功能。例如,键盘作为计算机的标准输入设备,它与 CPU 之间采用中断控制方式进行数据传输。每当用户在键盘上敲一个键,就会由键盘接口电路产生一个中断请求信号发往 CPU,CPU 响应键盘中断请求后进入中断服务程序。在键盘中断服务程序里的中断服务主要实现对用户所敲键的识别,并根据所敲键实现相应的功能。

④ 关中断。这次的关中断是在程序中使用关中断指令将 CPU 中的中断允许触发器清零,禁止一切可屏蔽中断,使后续的恢复现场工作不再被新的中断源中断。

⑤ 恢复现场。恢复现场指将中断服务程序开始时保护起来的寄存器内容恢复到中断响应前的状态。若保护现场使用入栈的方法,则恢复现场应使用相应的出栈操作。

⑥ 开中断。再次使用开中断指令将 CPU 的中断允许触发器置位,以便该中断服务程序执行完成后系统恢复到正常中断工作状态。

⑦ 中断返回。CPU 的指令系统通常会提供一条中断返回指令,中断服务程序结束前会执行中断返回指令。执行该指令的结果是将系统中中断响应保护断点时压入栈的内容出栈,一是恢复 PSW 值,二是恢复主程序被中断时的指令指针值,从而将程序控制返回到主程序被中断的指令处继续执行。

4. 单级中断和多级中断

一个机器系统中有多个中断源,但 CPU 一次只能响应和处理一个中断源的中断请求。当某一时间有两个或两个以上的中断源同时发出中断请求时,中断系统就必须从中选择一个进行响应,选择的依据就是各个中断源的中断优先权。中断系统可以对各中断源依据其发生的中断事件的重要性和紧迫性预先设定中断优先权,也可以由用户根据程序运行的需要灵活设置。

如果机器系统的中断源很多,还可以在中断优先权的基础上进一步分级,高一级的任何一个中断源的优先权都比低一级的任何一个中断源的优先权高,如图 5-17 所示。

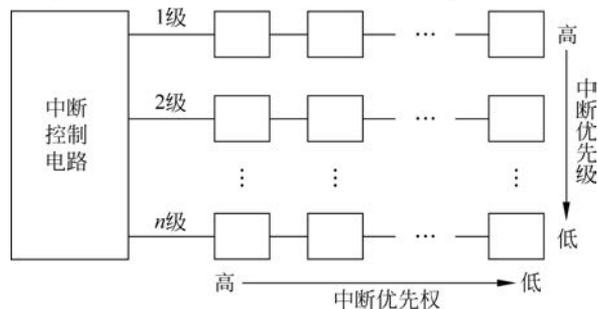


图 5-17 中断优先权和中断优先级

如果一个机器系统只有一个中断级,则称该机器的中断系统为单级中断系统;如果一个机器系统有多个中断级,则称该机器的中断系统为多级中断系统。

若 CPU 在执行一个主程序时发生了多个中断源的中断请求,中断系统会选择—个优先权高的中断源予以响应。那么,如果 CPU 在执行一个中断服务程序时发生了新的中断,而且新来的中断请求的优先权更高,中断系统该如何处理呢?有两种解决方案:一是响应,二是不响应,而响应与否取决于机器采用的是单级中断系统还是多级中断系统。

单级中断系统采用一种简单的处理方式:当几个不同优先权的中断源同时请求中断时,系统按照它们优先权的高低先后顺序—一响应。而当 CPU 正在处理一个中断时,不再响应其他新的中断源的中断请求,即使新的中断源的优先权更高也不予响应,只有当前的中断请求处理完毕后才响应新的中断请求。多级中断系统则允许先处理高优先级的中断源,再中断低优先级的中断服务,这称为多重中断或中断嵌套。理论上多重中断可以无限制地嵌套。

【例 5.1】 设 CPU 在执行一个主程序(非中断服务程序)时,先是 B 设备有了中断请求,经过一段时间后又有了 A 设备和 C 设备的中断请求。

(1) 假设机器是单级中断系统,且 A、B、C 设备的优先权顺序由高到低为 A→B→C。

(2) 假设机器是多级中断系统,且 A、B、C 设备分属不同的优先级,它们的优先级顺序由高到低为 A→B→C。

设 CPU 始终处于开中断状态,试通过图示的方法说明在单级中断系统和多级中断系统中的中断响应和处理过程。

解 (1) 在单级中断系统中,CPU 首先响应先来的 B 设备的中断请求,处理完成后返回主程序;然后从 A、C 中选择优先权更高的 A 设备的中断请求予以响应,处理完后再次返回主程序;最后响应 C 设备的中断请求,处理完后返回主程序,如图 5-18(a)所示。

(2) 在多级中断系统中,CPU 仍然首先响应 B 设备的中断请求;在 B 设备的中断服务过程中又发生了 A、C 中断,由于 A 设备的中断优先级更高,于是中断 B 设备的中断服务,响应 A 的中断请求,待 A 设备的中断处理完后返回 B 设备的中断服务程序;由于 C 设备的中断优先级比 B 设备的低,因此 B 设备的中断服务不再响应 C 设备的中断,直到 B 设备的中断处理完成后返回到主程序再响应 C 设备的中断请求,如图 5-18(b)所示。

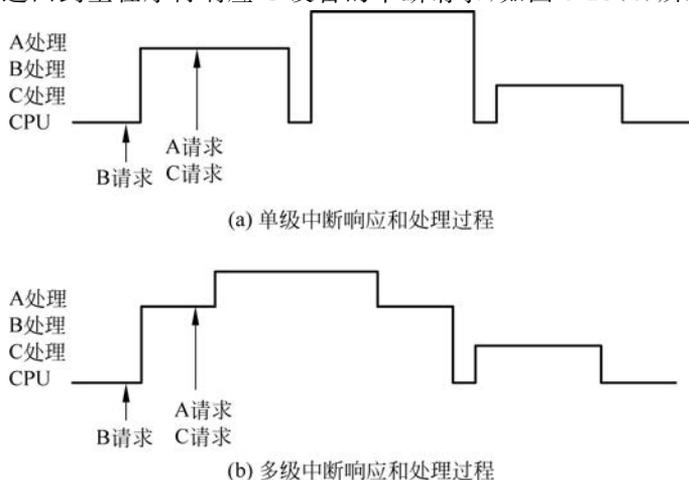


图 5-18 单级中断和多级中断系统的中断响应和处理过程

从例 5.1 可以看出,中断优先权决定了各中断源的中断响应顺序;而中断优先级决定了中断处理的顺序,先响应的中断不一定先处理完。

在多级中断系统中,利用中断屏蔽码可以改变中断源的中断处理顺序,使机器的中断系统控制更灵活。对于这一点,仍通过一个例子加以说明。

【例 5.2】 设 CPU 在执行一个主程序时,同时发生了 A、B、C 三个设备的中断请求,A、B、C 的中断优先级顺序由高到低为 A→B→C。

(1) 若 A、B、C 三个设备在执行中断服务程序时的中断屏蔽码如表 5-2 所示,其中“1”表示屏蔽,“0”表示开放,试通过图示的方法说明对 A、B、C 三个设备的中断响应和处理过程。

表 5-2 中断屏蔽码设置情况 1

设备名称	屏蔽码		
	A	B	C
A	1	1	1
B	0	1	1
C	0	0	1
CPU	0	0	0

(2) 若 A、B、C 三个设备在执行中断服务程序时的中断屏蔽码如表 5-3 所示,试通过图示的方法说明对 A、B、C 三个设备的中断响应和处理过程。

表 5-3 中断屏蔽码设置情况 2

设备名称	屏蔽码		
	A	B	C
A	1	0	0
B	1	1	0
C	1	1	1
CPU	0	0	0

解 以上(1)、(2)两种情况下的中断响应和中断处理顺序分别如图 5-19(a)和图 5-19(b)所示。

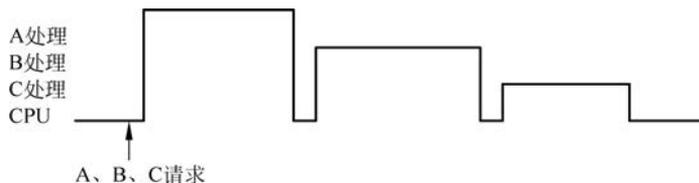
从例 5.2 可以看出,通过改变中断屏蔽码可以改变中断源的中断处理顺序,中断优先级低的中断源可以比中断优先级高的中断源先处理完成。

5. 中断接口电路

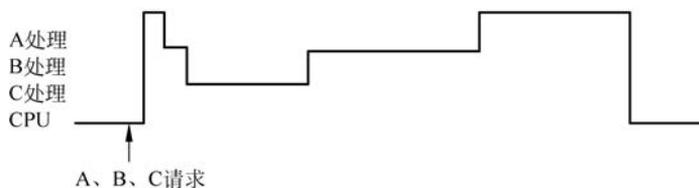
在 CPU 与设备之间采用中断控制方式进行传输的中断接口电路中,主要包含数据缓冲寄存器(DBR)、地址译码电路、中断请求触发器(IR)、中断屏蔽触发器(IM)、中断判优排队电路、中断向量逻辑和控制逻辑等,如图 5-20 所示。

根据图 5-20,CPU 控制设备进行一次输出的过程如下。

(1) CPU 启动设备,通过中断接口的控制逻辑向设备发出一个启动控制信号。



(a) 表5-2的中断响应和中断处理顺序



(b) 表5-3的中断响应和中断处理顺序

图 5-19 中断屏蔽码改变中断处理顺序

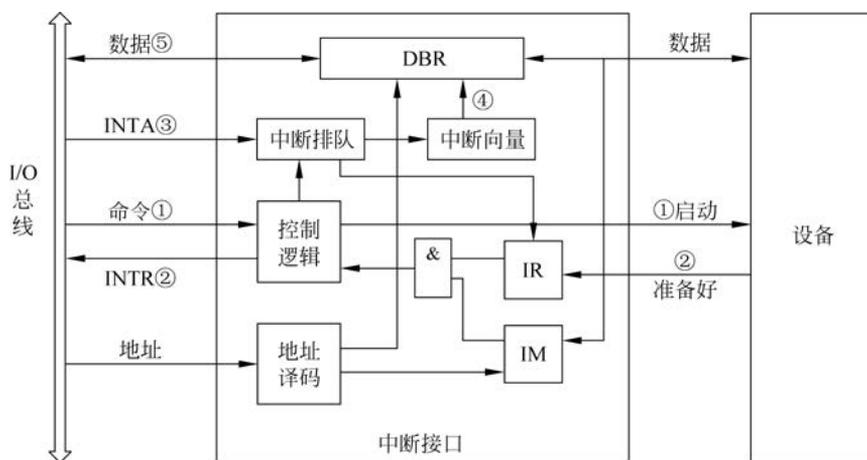


图 5-20 中断接口电路

(2) 设备准备好后向接口发回一个 READY 信号,将接口中的中断请求触发器置 1。若此时设备处于中断开放状态,即接口中的中断屏蔽触发器为 1,则可以通过控制逻辑向 CPU 发出一个中断请求信号 INTR。

(3) CPU 接收到中断请求后,若内部中断允许触发器处于开中断状态,则进入中断响应周期,向接口发出一个中断响应信号 INTA,并在该信号控制设备接口进行中断判优和中断源识别。

(4) 若本设备被选中,则将其对应的中断向量号送数据缓冲器,并经数据总线传送给 CPU; CPU 依据此中断向量找到中断服务程序的入口地址,从而转入中断服务程序执行。

(5) 在中断服务程序中,CPU 向设备传送一个数据。

至此,CPU 通过中断方式向设备传输一个数据的过程就完成了。之后,设备进行输出数据的处理工作,与此同时,CPU 返回原主程序继续执行。若设备输出数据处理完毕,又通过接口向 CPU 发出一个中断请求信号,则进入下一次中断响应和中断处理过程。

在计算机中,中断接口电路可以由专门的集成电路芯片实现。例如,Intel 8259A 就是

一个可编程的中断控制器,其内部结构如图 5-21 所示。它由 8 个部分组成:8 位中断请求寄存器 IRR、8 位中断屏蔽寄存器 IMR、8 位中断服务寄存器 ISR、优先级判别器、8 位数据总线缓冲器、中断控制逻辑、级联缓冲器/比较器和读写逻辑。

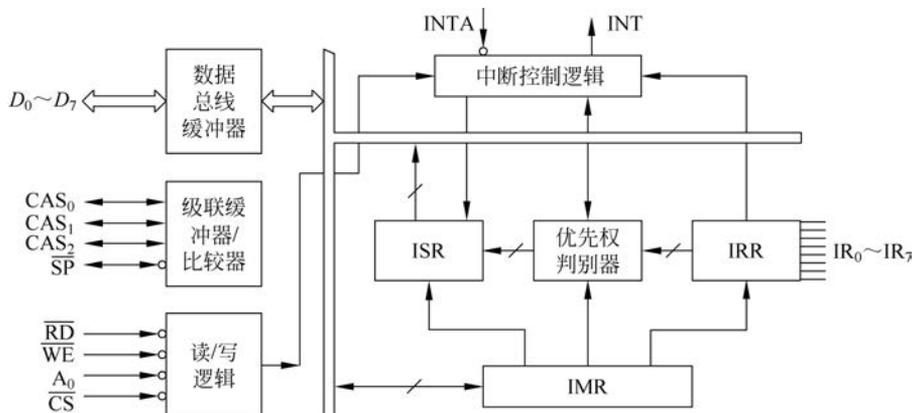


图 5-21 8259A 中断控制器

数据总线缓冲器通过数据总线与 CPU 连接,用于与 CPU 之间的双向数据交换,如 CPU 向 8259A 传送控制字、置中断屏蔽字和 8259A 向 CPU 传送中断向量等。读/写逻辑用于 CPU 对芯片进行读写操作和内部 I/O 端口的地址译码。一个 8259A 通过其内部的中断请求寄存器可以外接 8 个中断源的中断请求。通过级联控制逻辑,可以将 9 片 8259A 级联起来,提供最多 64 个外部中断源的中断请求和处理。

8259A 的中断处理过程如下:

- (1) 当外部中断源有了中断请求后,将中断请求寄存器的对应位置位;
- (2) 8259A 接收这些中断请求,并根据中断屏蔽寄存器的状态将未被屏蔽的所有中断请求送优先级判别器进行判优;
- (3) 选择一个优先级最高的中断源,经中断控制逻辑向 CPU 发出中断请求信号 INT;
- (4) 若 CPU 处于开中断状态,则响应中断,向 8259A 发回中断应答信号 INTA,进入由两个总线周期构成的中断响应周期;
- (5) 在第一个总线周期,8259A 把当前响应的最高优先权中断源所对应的中断服务寄存器的相应位置位,并将中断请求寄存器相应位清零;
- (6) 在第二个总线周期,8259A 将当前响应的中断源的中断类型号(中断向量号)通过数据总线送至 CPU;
- (7) CPU 依据此中断类型号便可获得该中断源的中断服务程序入口地址,从而转入中断服务程序执行。

8259A 提供了四种优先级选择方式。

(1) 完全嵌套方式。这是一种固定的优先级方式,它规定任何情况下总是 IR_0 的中断源优先级最高, IR_7 的中断源优先级最低。

(2) 轮转优先级方式,对每个中断源同等对待。当某个中断源处理完后,把它放到最低级别的位置上,这样保证每个中断源都有机会被响应。

(3) 轮转优先级 B 方式。CPU 可以在任何时候规定一个中断源为最高优先级,其他中

断源的优先顺序依据信号线的顺序相应确定。

(4) 查询方式。通过软件读取中断请求寄存器的状态,依据软件查询的顺序确定中断源的优先级顺序。

8259A 提供了两种屏蔽方式。

(1) 简单屏蔽方式。8 位屏蔽字与 8 位中断请求位一一对应,为“1”则屏蔽,为“0”则开放。

(2) 特殊屏蔽方式。通过屏蔽字的设置,允许低优先权中断源的中断请求中断高优先权中断源的中断服务。

例如,若 8259A 被设置为完全嵌套优先级方式和特殊屏蔽方式,则当屏蔽字设置为 11001111 时,IR₄ 和 IR₅ 上的中断请求除了可以中断低级别 IR₆~IR₇ 的中断服务外,还可以中断高级别 IR₀~IR₃ 的中断服务。这也就是前面讲到的通过对中断屏蔽字的设置可以改变中断处理的顺序。

8259A 的不同控制和工作方式是通过编程来实现的。

6. Pentium CPU 的中断机制

Pentium CPU 定义了两类中断源,即中断和异常。

中断主要是指外部中断,它是由 CPU 的外部硬件信号引起的。中断包括两种情况:一是可屏蔽中断,是由 CPU 的 INTR 引脚上接收到的中断请求。这类中断的允许或禁止受 CPU 标志寄存器中的 IF 位(即中断允许触发器)的控制。二是非屏蔽中断,是由 CPU 的 NMI 引脚接收到的中断请求。这类中断不能被禁止。

异常主要是指异常中断,它是由指令执行引发的。异常包括两种情况:一是执行异常,如 CPU 执行一条指令过程中出现错误、故障等不正常情况引发的中断;二是执行软件中断指令(如 INT 0、INT 3、INT n 等软中断指令)产生的异常中断。

Pentium 共能提供 256 种中断和异常。每种中断源分配一个编号,称为中断向量号或中断类型号。所有的中断和异常按其中断优先级分为 5 级,其中异常中断的优先级高于外部中断的优先级。

Pentium 是采用中断向量法进行中断响应的,即首先得到中断向量号,再通过中断向量号找到中断向量,从而获取中断服务程序的入口地址。Pentium CPU 取得中断向量号的方式根据中断源类型的不同也有所不同。对于软件中断指令 INT n 来说,其中的 n 即中断向量号;对于错误、故障等异常来说,系统会根据异常产生的条件自动指定向量号,每种异常的向量号是固定的,如中断向量号 0 对应除 0 错误异常、中断向量号 4 对应算术溢出错误异常等;而外部中断的中断向量号是系统设计时在中断控制器中预先设定的,当响应中断时由中断控制器提供;非屏蔽中断的中断向量号固定为 2。

所有中断服务程序的入口地址信息均存储在一个中断向量号检索表中。在 Pentium 实模式下,该表为中断向量表 IVT,保护模式下为中断描述符表 IDT。

(1) 实模式。中断向量表 IVT 位于内存地址从 0 开始的 1KB 空间。实模式下是 16 位寻址,中断服务子程序入口地址(段,偏移)的段寄存器和段内偏移量各为 16 位。它们直接登记在 IVT 中,每个中断向量号对应一个中断服务程序的入口地址。每个入口地址占 4 字节。256 个中断向量号共占 1KB。CPU 取得向量号后自动乘以 4,作为访问 IVT 的偏移,

读取 IVT 相应表项,将段地址和偏移量设置到 CS 和 IP 寄存器,从而进入相应的中断服务程序。实模式下的中断响应过程如图 5-22(a)所示。

(2) 保护模式。保护模式下为 32 位寻址。中断描述符表 IDT 的每一表项对应一个中断向量号,表项称为中断门描述符和陷阱门描述符。这些门描述符长度为 8 字节,共对应 256 个中断向量号,IDT 的表长为 2KB。用中断描述符表寄存器 IDTR 来指示 IDT 的内存地址。以中断向量号乘以 8 作为访问 IDT 的偏移,读取相应的中断门/陷阱门描述符表项。门描述符用于给出中断服务程序的入口地址(段,偏移),其中 32 位偏移量装入 EIP 寄存器,16 位的段值装入 CS 寄存器。由于此段值是选择符,因此还必须访问全局描述符表 GDT 或局部描述符表 LDT,才能得到段的基地址。保护模式下的中断响应过程图 5-22(b)所示。

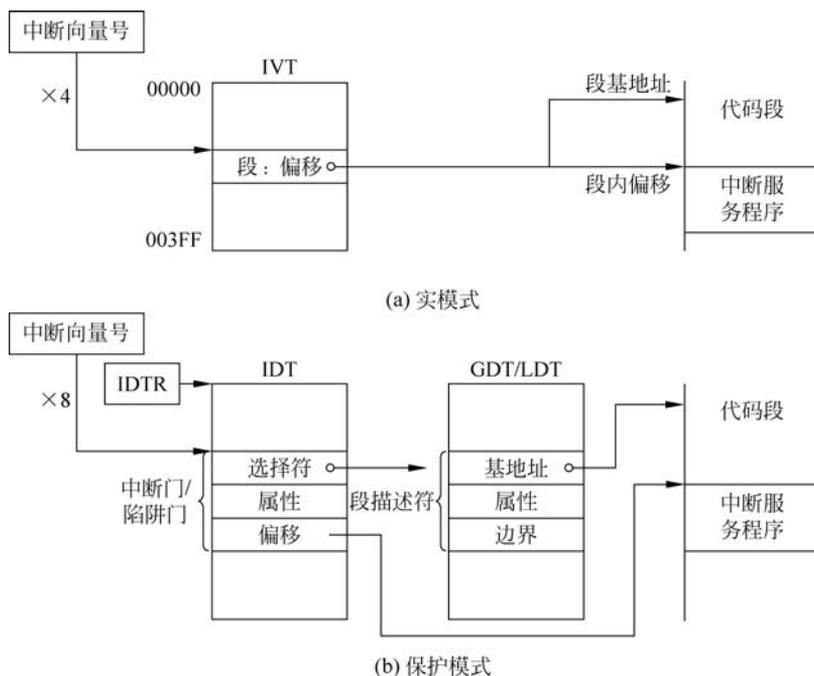


图 5-22 Pentium CPU 的中断响应过程

进入中断服务程序后,其处理过程如下:

(1) 当中断处理的 CPU 控制权转移涉及特权级改变时,必须把当前的 SS 和 ESP 寄存器的内容压入系统堆栈予以保存;

(2) 标志寄存器 EFLAGS 的内容也压入堆栈;

(3) 清除标志触发器 TF 和 IF;

(4) 当前的 CS 和 EIP 也压入此堆栈;

(5) 如果中断发生并伴随有错误码,则错误码也压入此堆栈;

(6) 完成上述中断现场保护后,将从中断向量号获取的中断服务程序入口地址(段,偏移)分别装入 CS 和 EIP,开始执行中断服务子程序;

(7) 中断服务子程序最后的 IRET 指令使中断返回。保存在堆栈中的中断现场信息被恢复,并从中断点继续执行原程序。

5.2.3 DMA 控制方式

虽然中断控制方式很好地解决了 CPU 与设备间并行工作的问题,尤其是对于慢速设备来说,采用中断控制方式进行数据传输,可以大大提高 CPU 的利用率,但是,在中断控制方式下,CPU 每经历一次中断,都要进行从中断请求信号的建立、中断源识别、中断响应到中断服务等操作,在中断服务程序里还要执行一系列诸如保护现场/恢复现场、开中断/关中断等的指令,这些操作和指令的执行花费了不少时间。对于 CPU 与一些高速设备间采用成组数据交换的应用来说,中断控制方式就有些显得力不从心了。为此,人们提出了一种 DMA 传送控制方式。

1. DMA 的基本概念

直接存储器访问(direct memory access,DMA)是一种完全由硬件(称为 DMA 控制器)控制主机与设备间进行数据交换的输入输出传送控制方式。它通过在主存与设备间建立一条直通的方法,来进一步提高 I/O 数据的传输效率。

在机器中,依据各部件所处的地位将它们划分为两大类:一类是主设备,一类是从设备。主设备是指占用系统总线并通过总线对其他从设备进行控制的设备。一般来讲,主设备能够在总线上给出地址和控制等信号,完成对存储器和外围设备等的访问,如 CPU 就是机器系统中的主设备。而从设备是指被主设备控制和访问的设备,如存储器及各种外围设备等。前面讲到的程序控制方式和中断控制方式,都是在 CPU 这一主设备的控制下完成存储器与外围设备间的数据交换。

为实现 DMA 传送,机器系统专门设置了一个主设备——DMA 控制器,代替 CPU 完成存储器与外围设备间的数据交换。DMA 控制器可以像 CPU 一样,通过总线向存储器和外围设备给出地址和控制信号,实现对这些设备的访问和控制。

表 5-4 给出了 DMA 控制方式、程序控制方式及中断控制方式的比较。

表 5-4 几种输入输出控制方式的比较

输入输出控制方式	输入输出过程控制	数据传输通道	CPU 效率	适用场合
程序控制方式	软件实现	经 CPU	低	单用户单道程序环境
中断控制方式	软、硬件结合实现	经 CPU	一般	多用户多道程序环境下慢速或速度不确定的设备
DMA 控制方式	硬件实现	不经 CPU	高	成组、高速设备

2. DMA 的工作模式

任何时候,机器的系统总线只能被一个主设备所占用。当 CPU 执行程序时,需要通过总线进行访存操作,如取指令、读/写数据等。DMA 控制器在进行 DMA 传输控制时也要使用总线对存储器进行读写操作等,这时就会产生总线的冲突。为此需要一种机制来协调 CPU 与 DMA 控制器对总线的使用,通常有以下 3 种方式。

(1) 突发方式。突发方式(burst mode)又称块传送方式。在这种方式下,当 CPU 所执行的程序要对一个 DMA 设备进行一块数据的读/写操作时,CPU 首先对 DMA 控制器进行

初始化,并发送如下信息。

- ① 是读还是写。
- ② 所读/写的存储器缓冲区的首地址。
- ③ 所读/写的 DMA 设备的地址。
- ④ 所读/写的字数。

然后 CPU 暂停对部分总线的控制,将总线控制权交给 DMA 控制器,由 DMA 控制器完成本次数据块的传送。当 DMA 传送结束时,DMA 控制器会向 CPU 发出一个中断请求信号,并将总线控制权重新交回给 CPU。

这种方式的优点是控制简单,适用于数据传输率很高的 I/O 设备实现成组数据的传送;缺点是 DMA 控制器在占用总线进行 DMA 传送时,CPU 基本上处于不工作状态或保持原状态,在一定程度上影响 CPU 的利用率。

(2) 周期挪用方式。周期挪用方式(cycle stealing)是一种常用的 DMA 工作模式。这种方式是 CPU 对 DMA 控制器完成一次数据块传送的初始化后继续工作;当 DMA 设备准备好一个数据的传送时,向 DMA 控制器发出 DMA 请求;DMA 控制器收到请求后向 CPU 申请占用总线,并在 CPU 响应后挪用若干个总线周期进行一个数据的传送,然后将总线控制权交还给 CPU;如此重复,直到完成一个数据块的传送。

与前一种方式相比,CPU 不需要暂停,只是在有了 DMA 请求时暂时让出若干个总线的控制权。而且,如果在 DMA 控制器使用总线的这些周期里,CPU 正好不需要使用总线(如使用总线访存等),则不影响 CPU 的执行,从而提高了 CPU 的利用率。这种方式的缺点是 DMA 控制器每控制传送一个数据都要向 CPU 申请占用总线 and 交回总线,在一定程度上影响 DMA 传送的效率。

(3) 透明方式。透明方式(transparent mode)又称为直接存储器访问方式。CPU 典型的指令周期包含若干个总线周期(又称 CPU 周期),如取指令周期、指令译码周期、取操作数周期和指令执行周期等。在整个指令周期里,CPU 并不是始终都需要进行总线操作的,如乘法指令的执行周期只涉及 CPU 内部的乘法运算,而且时间也较长。因此,可以利用 CPU 不进行总线操作的一些总线周期进行 DMA 传送,这样 CPU 和 DMA 控制器可以交替使用总线,而且总线使用的交替可以通过不同的总线周期直接分配,无须 CPU 和 DMA 控制器之间的请求、建立和归还过程。例如,将每个指令的指令译码周期固定分配给 DMA 传输之用,当某个指令译码周期有 DMA 请求时,直接进入 DMA 操作。这就是透明方式的基本思想。这种方式对 CPU 和 DMA 来说效率都是最高的,但决定 CPU 何时不使用总线的硬件设计十分复杂,而且随着 CPU 指令预取、指令流水等并行技术的应用,现代大多数 CPU 每个周期都使用总线,因此这种方式通常不被采用。

为提高 DMA 的传输效率,还可以对总线的配置进行设计。图 5-23(a)是一种传统的单总线系统。在这种系统中,DMA 控制器和 CPU、存储器、I/O 接口及设备一起连接在一条单总线上。DMA 控制器每控制一个数据的传输都至少要占用两个 CPU 周期,一个用于对设备的读或写,另一个用于对存储器的读或写,这两个 CPU 周期都需要占用总线。

图 5-23(b)是一种在传统的单总线基础上增加一条专用 I/O 总线的结构。在这种系统里,DMA 控制器对存储器的访问需要与 CPU 共享系统总线,而对 DMA 设备的访问则通过

I/O 总线进行,这样可以减少在 DMA 传输过程中因 DMA 控制器过多占用系统总线而对 CPU 产生的影响。

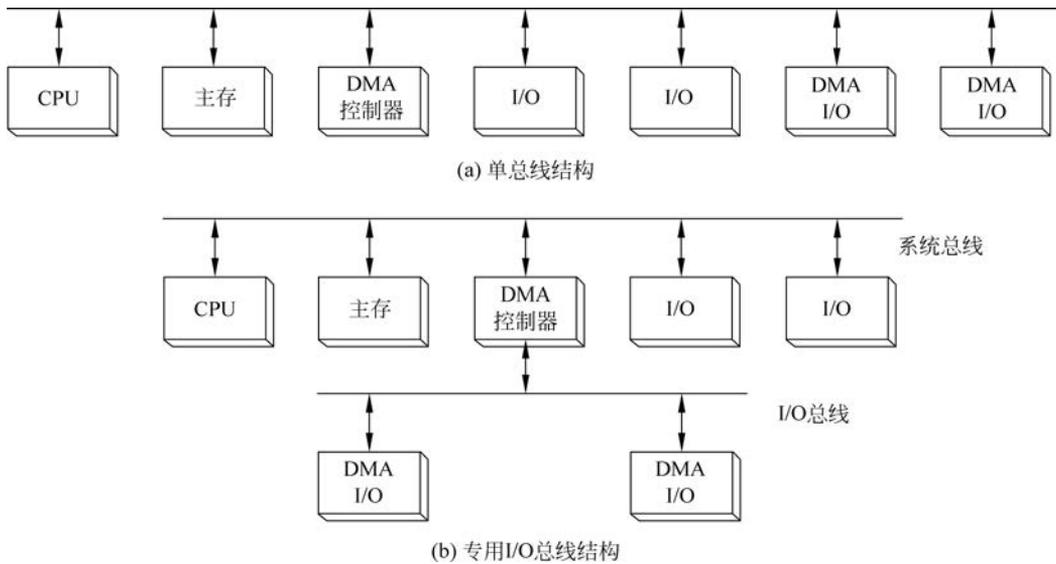


图 5-23 DMA 控制的总线配置

3. DMA 控制器的组成及工作原理

图 5-24 给出了一个 DMA 控制器的内部结构及与 CPU 等部件相联的组成框架图。其中,从 DMA 的内部结构看,它主要包含以下部件。

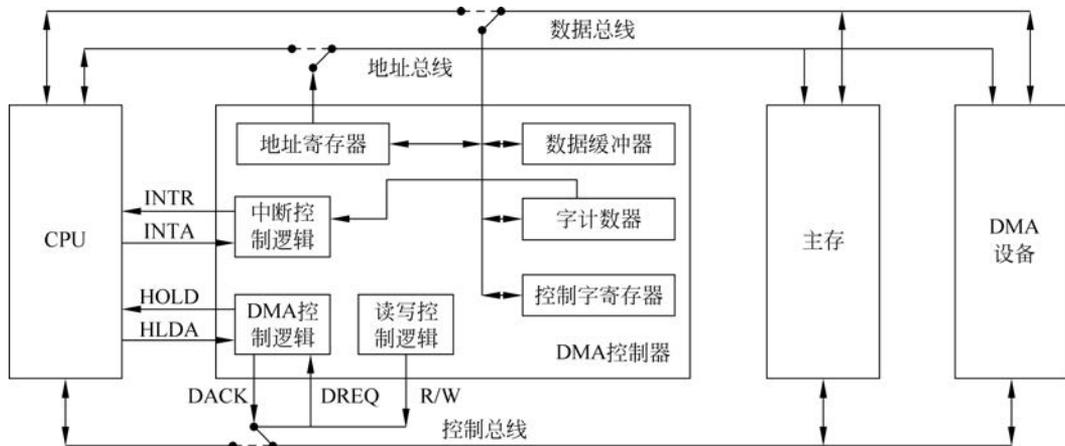


图 5-24 DMA 控制器的组成

(1) 地址寄存器。用于存放所交换的数据块在主存的首地址。在 DMA 传送前,由 DMA 初始化程序将数据块在主存中的首地址送到该地址寄存器。在 DMA 传送过程中,每交换一次数据,地址寄存器自动增 1,以指向下一内存单元,直到一批数据传送完毕为止。

(2) 字计数器。用于记录所传送数据的总字数。在 DMA 传送前,同样由 DMA 初始化程序将所传送数据的总字数送到该计数器。在 DMA 传送过程中,每传送一个字,字计数器

自动减 1,直到减为 0,表示该批数据传送完毕,然后通过中断控制逻辑向 CPU 发出 DMA 中断请求信号。

(3) 数据缓冲器。用于暂存每次传送的数据。通常 DMA 控制器与主存之间采用字传送,而与设备之间可能是字节或位传送。因此 DMA 控制器中还可能包括有装配或拆卸字信息的硬件逻辑,如数据移位缓冲寄存器、字节计数器等。

(4) DMA 控制逻辑。它用于负责管理 DMA 的传送过程,包括接收设备的 DMA 请求和向 CPU 申请总线的占用及归还等。

(5) 中断控制逻辑。在字计数器计为 0 时,表示一批数据交换完毕,由中断控制逻辑向 CPU 发出中断请求,请求 CPU 进行 DMA 操作的后处理。

(6) 读写控制逻辑。用于 DMA 控制器对存储器和设备进行读或写操作的控制,其功能与 CPU 的读写控制相同。

(7) 控制字寄存器。一些可编程的 DMA 控制器以集成芯片的形式提供,通常具有多种工作模式或操作方式,在使用前可通过软件编程的方法对其进行初始化设置。

结合图 5-24,可知 DMA 传输控制的过程如下:

(1) 在 DMA 数据传输前,首先由 CPU 通过初始化程序对 DMA 控制器进行预设置,包括将所传输数据块在内存的首地址送 DMA 内部的地址寄存器,将所传输数据的字数送 DMA 内部的字计数寄存器等;

(2) DMA 控制器选择一个 DMA 设备开始工作。当被选中的设备准备就绪时(对输入设备来说就是准备好了一个数据,对输出设备来说就是准备好接收),向 DMA 控制器发出一个 DMA 请求信号 DREQ;

(3) DMA 控制器接收到设备请求后,向 CPU 发出 HOLD 信号,申请占用总线;

(4) CPU 通过 HLDA 信号进行总线响应,同时将其引出脚的地址、数据和部分控制线置为浮空状态,即将总线的控制权让出;

(5) DMA 控制器获得总线控制权后,向设备回答一个 DMA 响应信号 DACK,并开始启动数据的传输;

(6) DMA 控制器将其地址寄存器的内容输出到地址总线上,并给出读/写控制信号,控制设备与存储器之间的一次数据交换,然后地址寄存器增 1,字计数寄存器减 1;

(7) 重复以上过程,直到字计数寄存器减为零,DMA 控制器向 CPU 发出中断请求,同时结束 DMA 传输,将总线控制权归还 CPU;

(8) CPU 响应 DMA 的中断请求,并进行 DMA 传输的后处理操作。

从上述 DMA 的传输控制过程可以看出,存储器与设备的数据交换没有经过 CPU,完全是在 DMA 控制器的控制下完成的,地址寄存器的增 1 操作、字计数器的减 1 操作等都是由硬件自动完成的,每传输一个数据也无须像中断方式那样经历很多的过程和执行一大堆的指令,因此传输效率大大提高了。

5.2.4 通道控制方式

1. I/O 通道的概念

在前面介绍的几种输入输出传输控制方法中,程序控制方式全部依赖 CPU 进行,中断

控制方式要求 CPU 不断介入,这两种方式主要适合于对速度要求不高的设备;在 DMA 控制方式中,CPU 无须干预数据传输的过程,而只需进行传输前的初始化和传输后的后处理,由于传输控制是硬件支持,所以适用于高速的数据输入输出。以上几种控制方式广泛应用于微型和小型计算机系统中。但在大型计算机系统中,由于外围设备配置多,输入输出操作频繁,因此对输入输出的传输方式有了更高的要求,主要体现在以下两个方面:

- (1) 尽量减少对 CPU 资源的占用,使用独立的控制部件完成输入输出的操作;
- (2) 控制更加灵活,输入输出控制部件能适应不同性能设备的要求。

为此,在 DMA 方式的基础上进一步发展了 I/O 通道方式,以满足大型计算机系统对输入输出控制的要求。

I/O 通道(I/O channel)又称通道处理器,是一种能执行有限指令集的专用处理器,它通过执行存储在内存中的固定或由 CPU 设置的通道程序来控制设备的输入输出操作。与 DMA 控制器一样,通道也是一个独立的控制部件,但它比 DMA 控制器更进了一步。一方面它是一个处理器,具有有限的指令集,能够执行程序;另一方面它控制灵活,可以适应不同工作方式、不同速度要求和不同数据格式的不同种类的设备的设备的要求。当然,通道处理器并不是一个通用处理器,而是专用于输入输出控制的 I/O 处理器。

通道处理器可以分担 CPU 大部分的 I/O 处理工作,如管理所有低速外围设备的输入输出操作,对 DMA 控制器的初始化工作,控制 DMA 的数据传输、数据格式转换、设备状态检测等,使 CPU 能从烦琐的 I/O 处理中解脱出来,真正发挥其“计算”的能力。

2. I/O 通道的功能

一个具有通道功能的典型机器结构如图 5-25 所示。

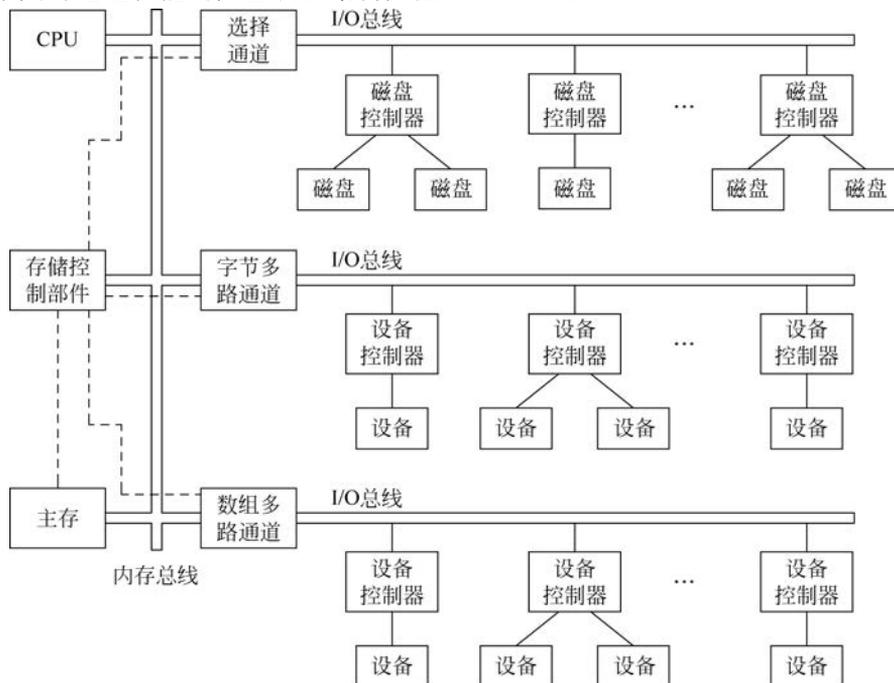


图 5-25 通道控制结构

一般来讲,通道主要包括寄存器部分和控制部分。寄存器部分包括数据缓冲寄存器、主存地址寄存器、字计数寄存器、通道命令字寄存器、通道状态寄存器等;控制部分包括分时控制、地址分配、数据传输、数据装配和拆卸等控制逻辑。

使用通道方式组织的输入输出系统一般采用“主机-通道-设备控制器-I/O 设备”四级连接方式。通道对 I/O 设备的控制通过设备控制器或 I/O 接口进行。对于不同的 I/O 设备,设备控制器的结构和功能各有不同,但通道与设备控制器之间一般采用标准 I/O 接口相连接。通道执行指令产生的控制命令经设备控制器的解释转换成对设备操作的控制,设备控制器还能将设备的状态反映给通道和 CPU。

具体来说,通道一般具有以下几方面的功能:

- (1) 接收来自 CPU 的 I/O 指令,根据指令要求选择设备;
- (2) 执行 CPU 为通道组织的通道程序,包括从主存中取出通道指令、对通道指令进行译码以及根据指令的要求向设备控制器发出各种命令;
- (3) 控制设备与主存之间的数据传输,提供主存地址和传送的数据字数控制,根据需要完成传输过程中的数据格式转换等;
- (4) 检查设备的工作状态,并将完整的设备状态信息送往主存或指定单元保存;
- (5) 向 CPU 发出输入输出操作中断请求,将外围设备的中断请求和通道本身的中断请求按次序报告 CPU。

设备控制器的具体任务包括:

- (1) 从通道接收通道命令,控制设备完成指定的操作;
- (2) 向通道提供设备的状态;
- (3) 将各种设备的不同信号转换成通道能够识别的标准信号。

CPU 通过执行 I/O 指令以及处理来自通道的中断,实现对通道的管理。来自通道的中断有两种:一种是数据传输结束中断;另一种是故障中断。通道的管理一般是由操作系统实现的。

3. I/O 通道的种类

按通道的数据传输及工作方式划分,通道可分成字节多路通道、选择通道和数组多路通道三种类型。一个机器系统可以兼有三种通道,也可以只包含其中一种或两种,以适应不同种类设备的需要。

(1) 字节多路通道。字节多路通道用于连接多个慢速或中速的设备,这些设备的数据传送以字节为单位。一般来讲,这些设备每传送一个字节需要较长的等待时间。因此,通道可以以字节为单位轮流为多个设备服务,以提高通道的利用率。字节多路通道的操作模式有两种:字节交叉模式和猝发模式。在字节交叉模式中,通道将时间分为一个个时间段,有数据传输要求的每一个设备都可以通过轮流分配得到一个时间段,完成一次与通道间的数据交换。如果某一设备需要传输的数据量比较大,则通道可以采用猝发的工作模式为其服务。在猝发模式下,通道与设备之间的传输一直维持到设备请求的传输完成为止。通道使用一种超时机制判断设备的操作时间(即逻辑连接时间),并决定采用哪一种模式。如果设备请求的逻辑连接时间大于某个额定的值,通道就转换成猝发模式,否则就以字节交叉模式工作。

(2) 选择通道。磁盘等高速设备要求较高的数据传输速度,而通道难以同时对多个这样的设备进行操作,只能一次对一个设备进行操作,这种通道称为选择通道。选择通道与设备之间的传输一直维持到设备请求的传输完成为止,然后为其他外围设备传输数据。选择通道的数据宽度是可变的,通道中包含一个保存输入输出数据传输所需的参数寄存器。参数寄存器包括存放下一个主存传输数据存放位置的地址和对传输数据进行计数的寄存器。选择通道的输入输出操作启动之后,该通道就专门用于该设备的数据传输,直到操作完成。

(3) 数组多路通道。数组多路通道以数组(数据块)为单位在若干高速传输操作之间进行交叉复用,这样可减少外围设备申请使用通道时的等待时间。数组多路通道适用于高速外围设备,这些设备的数据传输以块为单位。通道用块交叉的方法轮流为多个外围设备服务。当同时为多台外围设备传送数据时,每传送完一块数据后选择下一个外围设备进行数据传送,使多路传输可并行进行。数组多路通道既保留了选择通道高速传输的优点,又能同时为多个设备提供服务,使通道的功能得到有效发挥,因此数组多路通道在实际系统中得到了较多的应用。特别是对于磁盘和磁带等一些块设备,它们的数据传输本来就是按块进行的。而在传输操作之前又需要寻找记录的位置,在寻找的期间让通道等待是不合理的。数组多路通道可以先向一个设备发出一个寻找的命令,然后在这个设备寻找期间为其他设备服务。在设备寻找完成后才真正建立数据连接,并一直维持到数据传输完毕。因此采用数组多路通道可提高通道数据传输的吞吐率。

字节多路通道和数组多路通道都是多路通道,在一段时间内可以交替地执行多个设备的通道程序,使这些设备同时工作。但两者也有区别,首先数组多路通道允许多个设备同时工作,但只允许一个设备进行传输型操作,而其他设备进行控制型操作;而字节多路通道不仅允许多个设备同时操作,而且允许它们同时进行传输型操作。其次,数组多路通道与设备之间数据传送的基本单位是数据块,通道必须为一个设备传送完一个数据块以后才能为别的设备传送数据块;而字节多路通道与设备之间数据传送的基本单位是字节,通道为一个设备传送一个字节之后,又可以为另一个设备传送一个字节,因此各设备与通道之间的数据传送是以字节为单位交替进行的。

4. I/O 通道的工作过程

在使用了通道的机器里,CPU 有两种工作状态:一是目态,二是管态。目态是指 CPU 执行用户的目标程序,而管态是指 CPU 执行管理程序。通道传输过程是由用户程序执行一条广义访管指令引起的,其输入输出过程如图 5-26 所示。

通道完成一次数据传输的主要过程分为以下 4 个步骤:

- (1) 用户程序执行访管指令进入管理程序;
- (2) 管理程序根据用户的访管指令组织形成一个通道程序,并启动通道;
- (3) 通道处理器执行通道程序,控制完成指定数据的输入输出传输操作;
- (4) 通道在执行完通道程序后向 CPU 发出中断请求。

CPU 执行用户程序和管理程序以及通道执行通道程序的时间关系如图 5-27 所示。

由于大多数计算机的 I/O 指令都属于管态指令,因此,用户可通过在目标程序中设置一条要求进行输入输出的广义访管指令来使用设备。广义访管指令由访管指令和一组参数组成,它的操作码实际上就是对应此广义指令的管理程序入口。访管指令是目态指令,当目

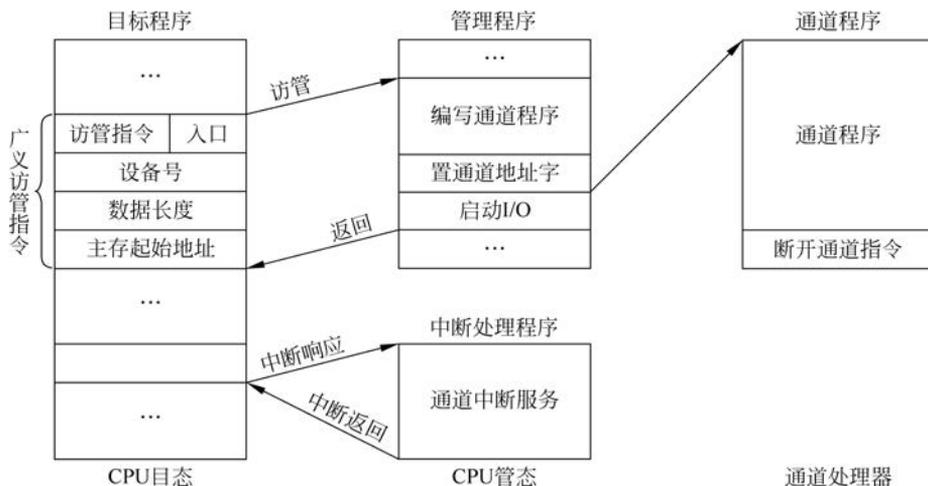


图 5-26 通道数据传输过程



图 5-27 CPU 和通道执行程序的时间关系

标程序执行到要求输入输出的访管指令时，会产生一个自愿访管中断。CPU 响应中断后，转向该管理程序入口，进入管态。

管理程序根据用户程序中的广义访管指令提供的参数，如设备号、交换数据长度和数据在主存中的起始地址等信息来编制形成一段通道程序。通道程序由若干条通道指令组成，能够完成 CPU 一条 I/O 指令所要求的操作。通道程序编制好后，放在主存中与这个通道相对应的通道程序缓冲区中。另外，在管理程序中还要把通道程序的入口地址放入相应的通道地址单元中。在管理程序的最后，用一条启动 I/O 设备的指令来启动通道开始工作。

接下来，通道处理器开始执行 CPU 为它组织的通道程序，完成指定的输入输出传输工作。此时，CPU 也返回用户程序继续执行，通道处理器和 CPU 并行工作。当通道处理器执行完通道程序后，会执行一条断开通道指令，向 CPU 发出中断请求，CPU 响应中断并进行相应的后处理。

至此，一次由通道控制的主存与设备间的数据交换结束。

5. 通道结构的发展

随着通道结构的进一步发展，出现了两种计算机 I/O 体系结构。

(1) 输入输出处理器 IOP。IOP 是通道结构的 I/O 处理器，它可以和 CPU 并行工作，提供高速的 DMA 处理能力，实现数据的高速传送。但是它不是独立于 CPU 工作的，而是主机的一个部件。有些 IOP，如 Intel 8089 IOP，还提供数据的变换、搜索以及字装配/拆卸能力。这类 IOP 广泛应用于中小型及微型计算机中。

(2) 外围处理机 PPU。PPU 基本上是独立于主机工作的,它有自己的指令系统,可完成算术/逻辑运算、读/写主存储器、与外围设备交换信息等。有的外围处理机直接就选用已有的通用计算机。外围处理机一般应用于大型计算机系统中,用于处理大量而繁杂的输入输出操作,以使 CPU 能从输入输出控制操作中最大限度地解脱出来,专用于“计算”的处理。

5.3 外部存储器

在第 4 章讲到,存储器按在计算机中所起的作用分为两大类,即内存和外存。内存又称为主存,主要由半导体存储器件构成,其特点是速度快;而外存又称为辅存,从现代存储介质的发展看,主要由磁介质存储器和光盘存储器构成,其特点是容量大,价格低廉。在计算机中,外存是作为设备来进行管理的。这一节主要讲述构成计算机外存的磁盘存储器、磁带存储器和光盘存储器。

5.3.1 磁盘存储器

磁盘是在一定的基质上涂上一层磁性材料而构成的圆盘,在磁盘表面利用磁存储原理来存储信息。磁盘分为硬盘和软盘两种,因它们的基质分别使用了“硬”的金属合金和“软”的聚脂薄膜而得名。硬盘和软盘在构成、容量和访问速度上都不同,但从信息的存储原理来讲,它们是完全相同的。

1. 磁记录原理和读写方式

磁盘是依靠由一个个同心圆组成的磁道上的、具有不同磁化方向的磁化元来存储 0、1 信息的。对这些磁化元的读写是通过一个磁头来进行的。磁头由磁性材料制作而成,形状如同一个矩形环,靠近磁道方向上开有一个小间隙,在磁头上还分别绕有一组写线圈和读线圈。磁头在对一个磁道进行读写操作时,磁头固定不动,而磁道运动,如图 5-28 所示。

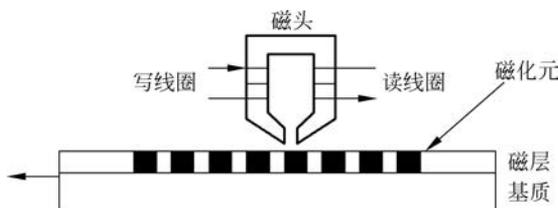


图 5-28 磁盘读写原理

进行写操作时,写线圈中会通过一定方向的脉冲电流,磁铁芯内随即产生一定方向的磁通,并在磁头间隙处产生很强的磁场。在这个磁场作用下,位于磁头下的磁道上的某个固定单元就被磁化成相应极性的磁化元。每个磁化元记录一位二进制位。当磁盘相对于磁头运动时,就可以连续写入一连串的二进制信息。

进行读操作时,经过磁头的磁化元会使磁铁芯内产生磁通的变化,从而使读线圈中产生一定的感应电势;经转换变成一定方向的脉冲电流,由此脉冲电流即可判定所读出的是 0 还是 1。

传统的磁盘是将读写操作共用一个磁头完成的,而新一代的硬盘系统则采用了不同的

读写机制。它将读写操作分别在两个不同的磁头上完成,写机制与传统方式相同,而读机制则有所不同。读磁头是由一个磁阻(magneto resistive,MR)式感应器组成,MR 的电阻值取决于在它下面运动的介质的磁化方向。电流通过 MR 感应器时,电阻的变化作为电压信号被检测出来,从而检测出读出的是 0 还是 1。这种新的读机制允许有更高的读操作频率,从而使磁盘可以达到更高的存储密度和读操作速度。

2. 磁盘存储器的物理构成

磁盘存储器主要由磁盘片、磁盘驱动器和磁盘控制器等组成。软盘是由单个盘片构成,而硬盘则由多个盘片构成,通常称为盘片组。

磁盘驱动器由机械和电器两大部分组成。在硬磁盘中,盘片组固定安装在一个主轴上,主轴通过传动带与一个主电机相连。硬盘组的每一个盘面上都有一个独立的磁头,磁头通过磁头臂连接在一个固定滑轨上运动的小车上,再通过小车与一个定位驱动装置相连,如图 5-29 所示。

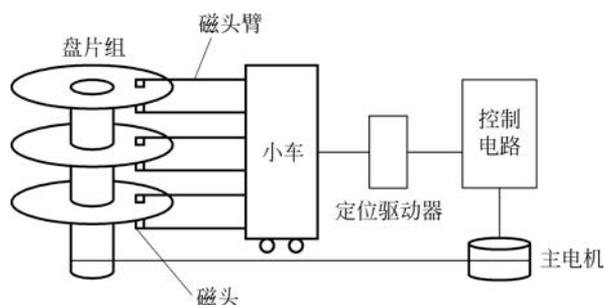


图 5-29 磁盘存储器的结构示意图

磁盘驱动器的机械部分一方面驱动盘片组按一定转速绕主轴转动,另一方面驱动小车通过磁头臂带动磁头沿径向运动,进行磁道的定位。目前磁头小车的驱动方式包括步进电机或音圈电机两种。步进电机靠脉冲信号驱动,控制简单,整个驱动定位系统是开环控制的,因此定位精度较低,一般用于道密度不高的硬磁盘驱动器。而音圈电机是线性电机,可以直接驱动磁头做直线运动;驱动定位系统是一个带有速度和位置信息反馈的闭环控制系统,驱动速度快,定位精度高,因此被更多的硬磁盘驱动器所采用。

磁盘驱动器的电器部分由一些控制电路组成,其功能主要包括主电机和定位电机的控制、磁头的选择和读写控制、磁盘索引的识别和扇区的定位等。

磁盘控制器是主机与磁盘驱动器之间的接口。磁盘存储器是高速外存设备,它与主机之间采用成组数据交换方式。作为主机与驱动器之间的控制器,磁盘控制器有两个接口:一个是与主机的接口,控制磁盘与主机之间通过外部总线交换数据;另一个是与磁盘驱动器的接口,根据主机命令控制磁盘驱动器的操作。另外,磁盘控制器还能完成磁盘与主机之间的数据格式转换以及数据的编码和解码工作。

3. 磁盘存储器的数据组织

磁盘是以“盘面 | 磁道 | 扇区”的方式来进行数据组织的。磁盘的盘面由一个个同心圆环组成,每一个圆环称为一个磁道。当对磁盘进行读写操作时,磁头定位在磁道上。为防止

或减少由于磁头定位不准确或磁域间的干扰所引起的错误,相邻磁道间有一定的间隙。一般来讲,一个磁盘面有上千个磁道。

磁道又进一步被分割成几十到上百个等长的圆弧,每一段圆弧称为一个扇区。一个扇区可以存储若干位信息,它也是磁盘与主机之间交换信息的基本单位。也就是说,主机对磁盘的读写操作一次至少是一个扇区。一个扇区所存储的二进制位数由操作系统对磁盘进行格式化时确定,大多数系统所定义的扇区为 512B~2KB。相邻扇区间同样留有一定的间隙。磁盘的数据组织形式如图 5-30 所示。

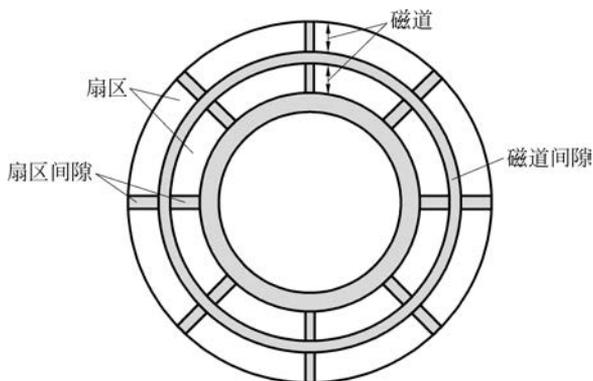


图 5-30 磁盘的数据组织形式

当磁盘转动时,内圈磁道经过磁头的速率要比外圈慢。因此,需要寻找一种方式来补偿速率的变化,使磁头能以同样的速率读写所有的磁道。一种有效的办法是通过增大信息位的间隙来实现。这样以恒定角速度(constant angular velocity, CAV)转动的盘,就能使磁头以相同的速率来扫描所有信息位。图 5-30 就是一种采用 CAV 技术的磁盘布局格式,它的好处是能方便地以磁道号和扇区号来直接寻址各个数据块。要将磁头从当前位置移到指定位置,只需将磁头径向移动到指定磁道,然后等待指定扇区转到磁头下即可。CAV 的缺点是外圈的长磁道上存储的数据位数与内圈的短磁道相同,浪费了长磁道的存储空间。

一个磁道的存储容量是与磁道的长度(周长)有关的。在上述 CAV 方式中,为使所有磁道的容量相同,从外圈磁道到内圈磁道的线性密度是逐渐增大的,磁盘的容量也就受到最内圈磁道所能达到的最大记录密度的限制。为提高存储容量,现代磁盘系统使用一种称为多区式记录(multi zone recording, MZR)的技术。它将磁盘盘面划分成若干个区(典型的是 16 区),每个区中所有磁道的扇区数是相同的。从而磁道的容量也是相同的。但不同区的磁道所拥有的扇区数是不同的,从而容量也就不同。越往内圈的区,磁道容量越小。当磁头由一个区移动到另一个区时,容量的改变会引起磁头上读写时序的变化,也就使磁盘控制电路的控制更为复杂,但这种电路的复杂所带来的的是磁盘容量的提升。

为使磁头能对扇区进行准确定位,磁道上会有一个起始点,称为磁盘索引,同时每个扇区还有起点和终点的标识。磁盘经格式化后,相关控制信息被记录在磁盘上,并由磁盘驱动器所识别和使用。

磁盘格式化的例子如图 5-31 所示。在此例中,每个磁道包含 30 个固定长度扇区,每个扇区的大小为 600B,其中 512B 为有效数据,剩余的为磁盘控制器使用的控制信息。其中各控制信息的含义如下。

ID 域：是唯一的一个标识或地址，用于定位具体扇区。

同步字节：是一个特殊的位序列，用来定义区域的起始点。

磁道号：用来标识磁盘上的一个磁道。

磁头号：用来标识磁盘组中的一个磁头。

扇区号：用来标识磁道上的一个扇区。

CRC：循环冗余校验码。

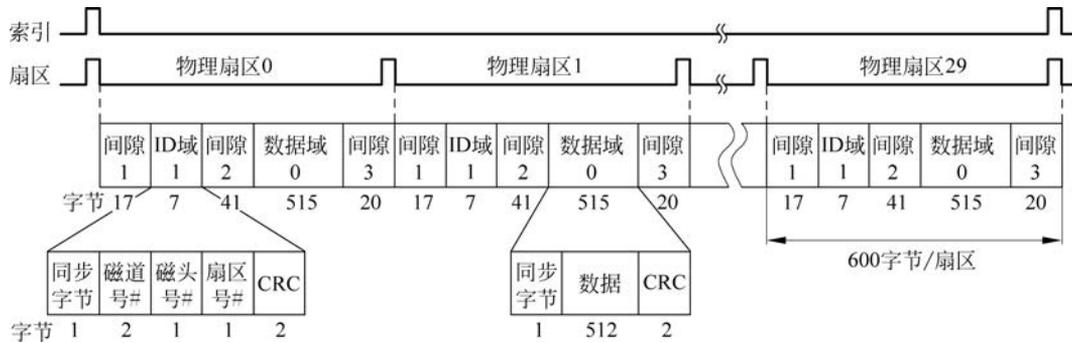


图 5-31 磁盘格式化形式

4. 磁盘存储器的性能参数

磁盘存储器的主要性能指标包括存储密度、存储容量和访问时间等。

(1) 存储密度。磁盘表面的存储密度主要分为道密度和位密度。道密度是指沿磁盘径向单位长度上的磁道数，单位为道/英寸(TPI)或道/毫米(TPM)。位密度是指磁道单位长度上能记录的二进制代码位数，单位为位/英寸(BPI)或位/毫米(BPM)。

(2) 存储容量。一个磁盘存储器所能存储的字节总数称为磁盘存储器的存储容量。存储容量有格式化容量和非格式化容量之分。格式化容量是指按照某种特定的记录格式所能存储信息的总量，也就是用户可以真正使用的容量。非格式化容量是磁记录表面可以利用的磁化单元总数。将磁盘存储器用于某计算机系统中，必须首先进行格式化操作，然后才能供用户记录信息。从图 5-30 可以看出，格式化后的磁盘的每个扇区内部和扇区之间都留有间隙及控制和校验信息，因此，格式化容量比非格式化容量小，一般为非格式化容量的 60%~70%。

在所有磁道容量相同的磁盘存储器中，格式化容量可以按下述方法计算：

$$\text{磁盘存储器总容量} = \text{盘面数} \times \text{每面容量}$$

$$\text{面容量} = \text{磁道数} \times \text{每道容量}$$

$$\text{道容量} = \text{扇区数} \times \text{每扇区容量}$$

而非格式化容量则可以按下述方法计算：

$$\text{磁盘存储器总容量} = \text{盘面数} \times \text{每面容量}$$

$$\text{面容量} = \text{磁道数} \times \text{每道容量}$$

$$\text{磁道数} = \text{道密度} \times \text{径向有效距离}$$

$$\text{道容量} = \text{位密度} \times \text{磁道周长}$$

(3) 访问时间。磁盘存储器的访问时间主要由寻道时间、旋转延时和传送时间三部分

组成。

当磁盘驱动器操作时,磁盘主轴电机带动盘片以恒定的速度转动。为了读或写,磁头必须精确定位在所含数据的磁道和扇区的起始处。磁道选择包括在可移动磁头系统中移动磁头或在固定磁头系统选择某个磁头。在可移动磁头系统中,磁头定位到该磁道所花的时间称为寻道时间。无论哪一种磁头系统,一旦磁道选定,磁盘控制器将处于等待状态,直到相关扇区旋转到磁头可读写的位置,这段时间称为旋转延时。寻道时间和旋转延时的总和称为存取时间,即定位到读写位置所需要的时间。从所访问的扇区头开始,整个扇区从磁头下经过,即完成了该扇区的数据传送,这部分时间称为传送时间。

由于寻道时间是不确定的,因此一般取平均寻道时间。平均寻道时间是最大寻道时间与最小寻道时间的平均值,目前硬盘的平均寻道时间通常在 5~10ms,而 SCSI 硬盘则应小于或等于 8ms。

磁盘存储器的旋转延时也是不同的,因此旋转延时也取平均值,现在一般在 3~6ms。平均旋转延时和磁盘转速有关,它用磁盘旋转一周所需时间的一半来表示。转速为 7200r/min 的磁盘,其平均旋转时间为 4.17ms。

磁盘的数据传送时间除了与所传送的数据量有关外,主要取决于磁盘的数据传输率。磁盘存储器在单位时间内向主机传送数据的字节数称为数据传输率。数据传输率与存储设备和主机接口逻辑有关。从主机接口逻辑考虑,应有足够快的传送速度向设备接收/发送信息;从存储设备考虑,磁盘转速越快,数据传输率也就越高。假设磁盘旋转速度为每秒 r 转,每条磁道容量为 N 个字节,则数据传输率为:

$$D = rN \text{ (B/s)}$$

若要传送的数据为 b 字节,则传送时间为:

$$T = \frac{b}{rN}$$

故磁盘存储器总的平均访问时间 T_a 可表示成:

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

其中, T_s 为平均寻道时间。

表 5-5 给出了一些典型磁盘存储器的性能参数。

表 5-5 典型磁盘存储器的性能参数

特 性	Seagate	Seagate	Toshiba	Toshiba	Hitachi
	Barracuda 180	CheetahX15-36LP	DT01ACA05050	HDD1242	Microdrive
应用	大容量服务器	高性能服务器	桌面系统	便携式	手持设备
容量/GB	181.6	36.7	500	5	4
最小寻道时间/ms	0.8	0.3	0.6	—	1
平均寻道时间/ms	7.4	3.6	15.5	15	12
转速/(r/min)	7200	15 000	7200	4200	3600
平均旋转延时/ms	4.17	2	4.17	7.14	8.33
最大传输率/(MB/s)	160	522~709	140	66	7.2
每扇区字节数	512	512	4096	512	512

续表

特 性	Seagate	Seagate	Toshiba	Toshiba	Hitachi
	Barracuda 180	CheetahX15-36LP	DT01ACA05050	HDD1242	Microdrive
每道扇区数	793	485	—	63	—
盘面数	24	8	2	2	2
柱面数(每盘面磁道数)	24 247	18 851	—	10 350	—

【例 5.3】 设某磁盘组有 6 片磁盘,每片有两个记录面,最上和最下两个面不用。存储区域内圈直径为 20mm,外圈直径为 80mm,道密度为 20 道/mm,最内圈磁道位密度为 1000B/mm。问:

(1) 盘组总存储容量是多少字节?

(2) 若该磁盘经格式化后,每个磁道有 12 个扇区,每扇区的容量为 512B,则该磁盘格式化容量为多少字节?

解 (1) 每个盘面的磁道数 $= 20 \times (80 - 20) / 2 = 600$ (道)

每道的容量 $= 3.14 \times 20 \times 1000 = 62\ 800$ (B)

盘组总存储容量 $=$ 盘面数 \times 磁道数 \times 道容量 $= 10 \times 600 \times 62\ 800 / 8 = 47$ (MB)

(2) 磁盘格式化容量 $=$ 盘面数 \times 磁道数 \times 每道扇区数 \times 每扇区容量

$= 10 \times 600 \times 12 \times 512 = 36.8$ (MB)

【例 5.4】 设一个转速为 15 000r/min 的磁盘,其平均寻道时间为 4ms,每磁道有 500 扇区,每扇区大小为 512B。现欲从该磁盘上读取一个由 2500 个扇区组成的、总长为 1.28MB 的文件。

(1) 假设该文件随机分布在磁盘的不同扇区中,试计算读取该文件所需的总访问时间。

(2) 若该文件所占扇区分布在同一个柱面的 5 个磁道上,试计算读取该文件所需的总访问时间。

解 (1) 根据磁盘转速可以计算得到平均旋转延时为 2ms,对每个扇区而言:

访问时间 $=$ 平均寻道时间 $+ 平均旋转延时 + 传送时间 = 4 + 2 + 4 / 500 = 6.008$ (ms)

则读取该文件所需的总访问时间为

总访问时间 $= 2500 \times 6.008 = 15.02$ (s)

(2) 由于 2500 个扇区正好占用了 5 个磁道,且这 5 个磁道又位于同一柱面上,因此寻道只需一次,时间为 4ms,读取该文件所需的总访问时间为

总访问时间 $= 4 + 5 \times 读取一个磁道的时间 = 4 + 5 \times (2 + 4) = 0.034$ (s)

知识拓展

现代磁盘接口

主机与硬盘存储器的连接是通过磁盘接口进行的。计算机中的磁盘接口主要包括 IDE 接口、SCSI 接口、光纤通道和 USB 接口等。

传统 PC 机主板上主要集成的是 IDE 接口,如图 5-32 所示。IDE 是 Integrated Device Electronics 的简称,又被称为 ATA(advanced technology attachment)。这两个名词都有厂商在使用,指的是相同的东西。IDE 之后又推出了 EIDE,即扩展的 IDE(Enhanced IDE),而这个接口标准同时又被称为 Fast ATA。所不同的是 Fast ATA 专指硬盘接口,而 EIDE

还制定了连接光盘等产品的标准。之后又陆续推出了速度更快的接口,名称都留有 ATA 的字样,如 Ultra ATA 以及传输速度分别达到 66MB/s、100MB/s、133MB/s 的 ATA/66、ATA/100、ATA/133 等。

SCSI 的英文全称为 small computer system interface(小型计算机系统接口),是同 IDE (ATA)完全不同的接口。SCSI 并不是专门为硬盘设计的接口,而是一种广泛应用于小型机上连接各种具有 SCSI 接口设备的高速数据传输接口。SCSI 接口具有支持多任务、带宽大、CPU 占用率低以及支持热插拔等优点,但其硬盘价格较高,很难在 PC 中普及,因此 SCSI 硬盘主要应用于中、高端服务器和高档工作站中。SCSI 接口从诞生到现在已经历了 30 多年的发展,先后衍生出了 SCSI-1、Fast SCSI、FAST-WIDE-SCSI-2、Ultra SCSI、Ultra 2 SCSI、Ultra 160 SCSI、Ultra 320 SCSI 等,现在市场中占据主流的是传输速度分别达到 160MB/s、320MB/s 的 Ultra 160 SCSI、Ultra 320 SCSI 接口产品。

以上传统的 ATA 和 SCSI 接口均采用并行传输方式。现代 PC 机主要采用串行硬盘接口。由 ATA 和 SCSI 分别发展出的串行 ATA 和串行 SCSI 接口就是目前被广泛使用的两种硬盘接口。

串行 ATA(serial ATA,SATA)接口如图 5-33 所示。它以连续串行的方式传送数据,可以在较少的位宽下使用较高的工作频率来提高数据传输的带宽。serial ATA 1.0、serial ATA 2.0 和 serial ATA 3.0 所定义的数据传输率分别为 150MB/s、300MB/s 和 600MB/s。SATA 一次只传送 1 位数据,能减少 SATA 接口的针脚数目,因此连接电缆数目变少。而且其接口非常小巧,排线也很细,有利于机箱内部的空气流动,不仅有利于机箱散热,也使机箱内部显得不太凌乱。与并行 ATA 相比,SATA 还有一大优点就是支持硬盘的热插拔。

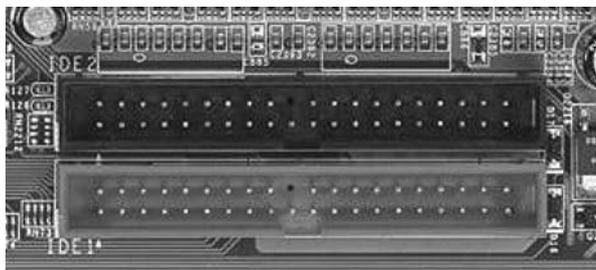


图 5-32 IDE 接口

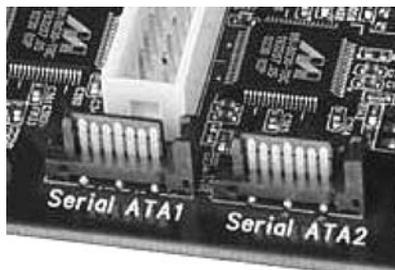


图 5-33 串行 ATA 接口

SAS(serial attached SCSI)即串行连接 SCSI,是新一代的 SCSI 技术,和 serial ATA (SATA)接口相同,都采用串行技术来获得更高的传输速度,并通过使用更少的连接电缆数和更小巧的接口来改善机箱内部空间等。SAS 的接口技术可以向下兼容 SATA。具体来说,二者的兼容性主要体现在物理层和协议层的兼容。在物理层,SAS 接口和 SATA 接口完全兼容,SATA 硬盘可以直接使用在 SAS 的环境中。从接口标准上而言,SATA 是 SAS 的一个子标准,因此 SAS 控制器可以直接操控 SATA 硬盘。但是 SAS 却不能直接使用在 SATA 的环境中,因为 SATA 控制器并不能对 SAS 硬盘进行控制。在协议层,SAS 由 3 种类型的协议组成,根据连接设备不同使用相应的协议进行数据传输。其中串行 SCSI 协议(SSP)用于传输 SCSI 命令,SCSI 管理协议(SMP)用于对连接设备进行维护和管理,SATA 通道协议(STP)用于在 SAS 和 SATA 之间进行数据传输。在这 3 种协议的配合下,

SAS 可以和 SATA 以及部分 SCSI 设备无缝结合。与传统的并行 SCSI 接口相比, SAS 在数据传输速度上也得到了进一步提升。

光纤通道(fibre channel)和 SCSI 接口一样最初也不是为硬盘设计开发的接口技术,而是专门为网络系统设计的,随着存储系统对速度的需求,才逐渐应用到硬盘系统中。光纤通道硬盘是为提高多硬盘存储系统的速度和灵活性而开发的,它的出现大大提高了多硬盘系统的通信速度。光纤通道的主要特性有热插拔性、高速带宽、远程连接、连接设备数量大等。

光纤通道是为像服务器这样的多硬盘系统环境而设计的,能满足高端工作站、服务器、海量存储子网络、外设间通过集线器/交换机和点对点连接进行双向串行数据通讯等系统对高数据传输率的要求。

USB 接口不是专为硬盘设计的接口标准。目前在 PC 上, USB 接口用途广泛,如可以用于连接移动硬盘。有关 USB 接口的内容将在第 6 章详细讲述。

5.3.2 磁带存储器

磁带存储器也属于一种磁介质存储器,与磁盘存储器一样,它也是通过磁记录原理来存储信息的。但与磁盘不同的是,它是通过在一条柔韧的聚酯薄膜带上涂上一层磁性材料来记录信息的。磁带卷在两个可以来回转动的转轴上,并固定装入一个盒中,组成盒式磁带。计算机所使用的磁带和磁带机与家用的磁带录音机相似,也是通过磁带在一个固定磁头下的平行移动来完成数据存取的。计算机中所用的磁带宽度为 0.15in(0.38cm)~0.5in(1.27cm),长度为 600in(182m)到 2400in(728m)。

磁带上的磁道是沿磁带运动方向平行排列的。早期的磁带系统一般使用 9 个磁道,每次存取一个字节;其中 8 个磁道构成一个有效字节信息,第 9 个磁道上附加的奇偶校验位。后来的磁带系统使用 18 或 36 个磁道,对应于数字的一个字或双字。这种记录格式称为并行记录。然而,现代大多数磁带系统使用串行记录方式,数据作为一系列二进制位串沿同一磁道顺序存储,就如同磁盘在同一磁道上顺序存储数据一样。随着技术的发展,磁带上的磁道数可以达到几百道,这大大提高了磁带存储器的容量。

串行记录磁带使用一种被称为蛇形记录(serpentine recording)的记录方式。按此方式,数据从一个磁带的头部开始,沿一个磁道从头到尾记录;到达磁带尾部时,再沿另一磁道从尾到头记录;再次回到磁带头部时,又沿第三个磁道从头到尾记录;如此往复,如图 5-34 所示。

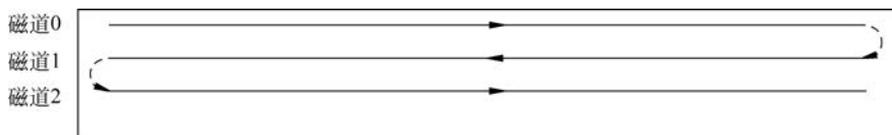


图 5-34 磁带的蛇形记录方式

磁带在数据组织上也是采用分块的方式。图 5-35(a)和 5-35(b)分别是采用并行记录的 0.5in 9 道启停式磁带和采用串行记录的 0.25in 数据流磁带的的数据记录格式。

0.5in 9 道启停式磁带是一种国际上通用的标准磁带。每盘带均设有卷头标 BOT 和卷

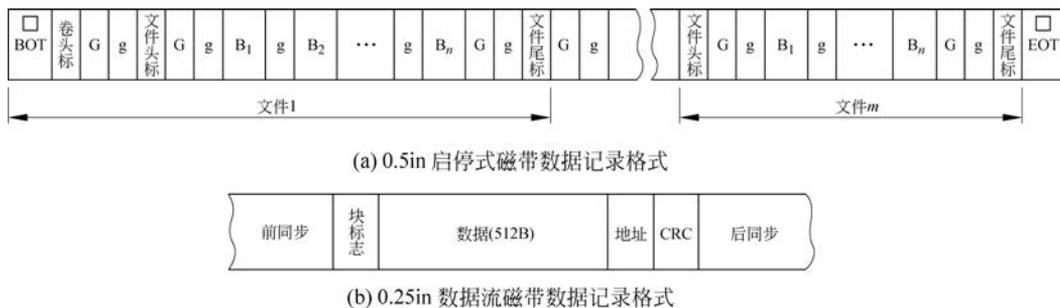


图 5-35 磁带记录格式

尾标 EOT, 标记用一块矩形金属反光薄膜制成。光电检测元件可检测到这两个标记, 表示记录的开始和结束。磁带上留有间隙 G(3.75in) 或 g(0.6in), 后者为数据块间的间隙, 它们的大小取决于磁带机的快启停性能。

信息可用两种形式存储: 一种是文件形式。一盘带可记录若干个文件, 一个文件又分若干数据块(B_i), 每个文件始末有文件头标和文件尾标, 卷头标、索引、文件头标、文件尾标的大小均为 80B, 其内容视操作系统而定。第二种是数据块形式, 磁带可在数据块之间启停, 进行数据传输。在 9 道带中, 8 位是数据磁道, 存储一个字节; 另一位是这一字节的奇偶校验位, 叫作横向奇偶校验码。在每一数据块 B_i 内部, 沿走带方向每条磁道还有 CRC 校验码。

0.25in 数据流磁带也是一种通用的标准磁带。其中 9 道磁带的的数据记录格式包括前同步、数据块标志(1B)、用户数据(512B)、地址号(4B)、CRC 校验码(2B)和后同步。

为提高磁带存储器的访问速度, 磁头能同时对几个相邻磁道(通常是 2~8 个磁道)进行读写操作。数据仍是沿各磁道蛇形串行记录, 但数据块不是在同一磁道上顺序存放, 而是在相邻磁道依序排列的。如图 5-36 所示是一个能同时进行 4 个磁道读写的磁带的块分布图。

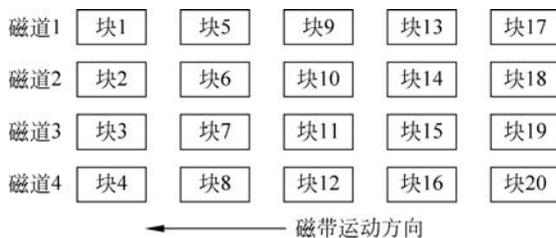


图 5-36 磁带的块分布图

表 5-6 给出了几个磁带的系统参数。

磁带驱动器是一种顺序存取设备。若磁头当前定位于块 1, 而需读块 N , 则它必须依次读物理块 $1 \sim N-1$, 每次一个块。若磁头当前定位已超越所需访问的块时, 它必须倒带, 从前面开始重新向前读。

磁带存储器是一种低价格、慢速、大容量的辅助存储器, 在计算机系统中常用于数据备份。

表 5-6 几个磁带的系统参数

系统参数	DLT4000	DLT8000	SDLT600
容量/GB	20	40	300
数据速率/(MB/s)	1.5	6	36
位密度/(Kb/cm)	32.3	38.6	92
磁道密度/(道/cm)	101	164	587
磁带长度/m	549	549	597
磁带宽度/cm	1.27	1.27	1.27
磁道数	128	208	448
同时读写的磁道数	2	4	8

5.3.3 光盘存储器

光盘存储器是一种采用光存储技术存储信息的存储器,它采用聚焦激光束,在盘式介质上非接触地记录高密度信息,以介质材料光学性质(如反射率、偏振方向)的变化来表示所存储信息的“1”或“0”。由于光盘存储器具有容量大、价格低、携带方便及交换性好等特点,已成为计算机中一种重要的辅助存储器,也是现代多媒体计算机 MPC 不可或缺的存储设备。

光存储技术源于 20 世纪 70 年代。1972 年,Philips 公司设计出世界上第一个能播放模拟电视信号的光盘系统。1978 年,世界上第一台商品化的激光视盘机(laser vision, LV)由 Philips 推出,其原理是仿效声音唱片的形式,把图像和伴音信号记录在圆盘上;用激光束检测盘上记录的信息,将其转换成电信号;经处理后还原成视频和音频信号,由电视机显示图像和发出声音。1981 年,Philips 公司和 Sony 公司携手推出了数字激光唱盘(compact disc-digital audio,即 CD-DA),并为此制定了光盘技术领域非常重要的基础性技术文件——《红皮书标准》。

1985 年,Philips 和 Sony 的研究人员在经过几年的努力后终于解决了光盘上只能记录数字音乐信息,而不能记录计算机文件信息的问题。具体来说就是解决如何在光盘上划分地址,以便计算机系统可以根据地址编号随时存取数据的问题和降低光盘数据存取误码率问题。为此他们公布了在光盘上记录计算数据的《黄皮书标准》。后来国际标准化组织 ISO 又对该标准进行了完善,发布了 ISO 9660 标准。由此,CD-ROM 便进入了计算机,并很快得到了广泛的应用,现已成为现代多媒体计算机中的标准配置之一。随后,研究人员们一方面努力提高 CD-ROM 的读取速度,由最初的 2 倍速、4 倍速(MPC3 标准)发展到今天的 52 倍速;另一方面又进一步推出了用于计算机中的、可读写的光盘和 DVD 等,巩固和确立了光盘存储器在计算机辅助存储器中的重要地位。

1. 光盘存储器的分类

按可擦写性分类,光盘主要包括只读型光盘和可擦写型光盘。

只读型光盘所存储的信息由光盘制造厂家先用模板一次性写入,以后只能读出数据而不能写入任何数据。按照盘片内容所采用的数据格式的不同,又可以将盘片分为 CD-DA、CD-I、Video-CD、CD-ROM、DVD 等。

可擦写型光盘是由制造厂家提供空盘片,用户可以使用刻录光驱将自己的数据刻写到

光盘上,包括 CD-R、CD-RW、相变光盘及磁光盘等。

常见的光盘种类、功能及相关标准如表 5-7 所示。

表 5-7 常见光盘种类、功能及相关标准

光盘种类	数据容量	执行标准	出现时间	功能说明
CD-DA	最大播放音乐时长 74 min	红皮书	1982 年	CD 系列光盘的始祖,由 Philips 和 Sony 于 1982 年正式发布,主要用于音乐存储
CD-ROM	可存储 650MB 数据	黄皮书	1985 年	由 Philips 和 Sony 联合制定,定义了存储计算机数据的规范,规定了地址数据结构、数据纠错、扇区大小等,使光存储进入计算机领域
CD-I	760MB	绿皮书	1986 年	Philips 和 Sony 针对消费电子市场推出的一种交互式多媒体数据存储格式,使之能同步播放声音、影像及其他信息如文字等
CD-R	700MB	橙皮书	1992 年	一次刻录型的光盘片,不管数据是否填满盘片,只能写入一次,即使还剩余空间,也不能再写
CD-RW	700MB	橙皮书的第三部分	1996 年	刻录方式与 CD-R 相同,区别是其可以擦除和重复写入,CD-RW 驱动器完全兼容 CD-R 盘片
Video CD	70 min MPEG1 格式数据	白皮书	1993 年	可存储按 MPEG1 格式压缩的视频和音频信息,主要应用播放电影等
DVD	可存储 17GB 数据	ISO/IEC 16448	1996 年	全称是数字视盘(digital video disk),将计算机和家庭娱乐融合起来,DVD 驱动器可以识别各种 CD 盘片,已取代 CD-ROM

2. CD-ROM

标准 CD-ROM 盘片的直径为 120mm,中心装卡孔径为 15mm,厚度为 1.2mm,质量约 14~18g,其基质由树脂(如聚碳酸酯)制成,数据信息以一系列微凹坑的样式刻录在光盘表面上。CD-ROM 光盘在制作时,首先用精密聚焦的高强度激光束制造一个母盘,然后以母盘作为模板压印出聚碳酸酯的复制品;再在凹坑表面上镀一层高反射材料(铝或金),最后在外层上涂一层丙烯酸树脂以防灰尘或划伤,如图 5-37 所示。

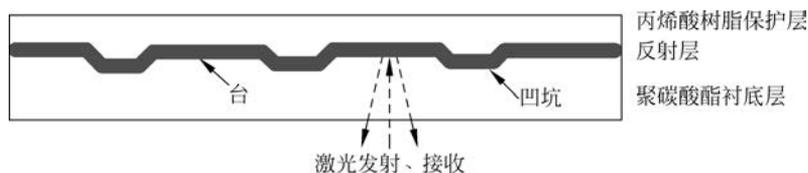


图 5-37 CD-ROM 的组成结构

CD-ROM 是通过安装在光盘驱动器内的激光头来读取盘片上的信息的。当盘片转动并经过激光头时,激光头能产生可以穿过透明聚碳酸酯层的低强度激光束。激光束照射到盘片的不同区域时,反射的激光强度会发生变化。具体来说,当激光束照射在凹坑上时,由于凹坑表面有些不平,光被散射,反射回的光强度变低;凹坑之间的区域称为岸台(land),岸台的表面光滑平坦,反射回的光强度高。光传感器将检测到的这种光强变化转换成数字信号。传感器以固定的间隔检测盘表面,一个凹坑的开始或结束表示存储了一位二进制“1”;间隔之间无标高变动出现时,记录的是“0”。

CD-ROM 与磁盘在数据记录方式上有所不同。磁盘由一个个同心圆的磁道组成；而 CD-ROM 却不同，它的整个盘面上只有一条螺旋式轨道，由靠近中心处开始，逐圈向外旋转直到盘的外沿。靠外的扇区与靠内的扇区具有相同的长度，因此按同样大小的段分组的信息可以均匀分布在整个盘上。

CD-ROM 的数据存储也是以块为单位进行组织的，典型的块格式如图 5-38 所示。



图 5-38 CD-ROM 的数据块格式

它由下列字段组成。

(1) Sync: 同步字段，标志一个块的开始。Sync 由 12 B 组成，第 1 B 为全 0，第 2~11 B 为全 1，第 12 B 为全 0。

(2) ID: 标识字段，包含块地址和模式字节。模式 0 表示一个空的数据域，模式 1 表示使用纠错码和 2048 B 的数据，模式 2 表示不带纠错码的 2336 B 的用户数据。

(3) Data: 用户数据域。

(4) Auxiliary: 此辅助域在模式 2 下是附加用户数据，在模式 1 下是 288 B 的纠错码。

与传统的硬盘相比，CD-ROM 有两个优点：

(1) 价格便宜，单片容量达 650MB，单位价格比硬盘低得多；

(2) 光盘是可交换的，可以用于批量制作软件产品，也方便用户存档和交换等。

CD-ROM 的缺点是只能读，不能修改；存取时间比磁盘驱动器长得多。

CD-ROM 是通过专门的 CD-ROM 驱动器（即通常所说的光驱）来进行读操作的。CD-ROM 驱动器一方面完成对光盘的读操作，另一方面与主机相接口。常见的 CD-ROM 驱动器接口标准主要有 3 种。

(1) 专用接口。专用接口是由各 CD-ROM 驱动器生产厂家提供的，用于将 CD-ROM 与主机连接起来。目前专用接口正逐步被取代。

(2) IDE(EIDE)接口。IDE 接口的 CD-ROM 驱动器直接插在计算机主板上的 IDE 或 EIDE 插口上，无须配置总线接口卡，这也是目前微型计算机中普遍采用的一种接口方式。

(3) SCSI 接口。SCSI (small computer system interface, 小型计算机系统接口) 是目前比较流行的输入输出接口标准。相对来说，SCSI 接口比 IDE 接口速度更快。

3. CD-R

CD-R 是一种一次写、多次读的可刻录光盘系统，它由 CD-R 盘片和刻录光驱组成。

CD-R 光盘与普通 CD-ROM 光盘在外观尺寸、记载数据的方式等方面是相同的，也同样是利用激光束的反射原理来读取信息的。但与 CD-ROM 不同的是，CD-R 光盘表面除了含有聚碳酸酯层、反射层和丙烯酸树脂保护层外，另外还在聚碳酸酯层和反射层之间加上了一个有机染料记录层。

当使用 CD-R 刻录光驱对空白盘片进行刻录时，将写激光束照射到有机染料记录层上，激光照射时产生的热量将有机染料烧熔，并使其产生光痕。光痕会使今后读激光束改变光

的反射率,从而达到一次刻录改写信息的目的。

4. CD-RW

CD-RW 光存储系统是在 CD-R 基础上进一步发展起来的,是一种多次写、多次读的可重复擦写的光存储系统。

CD-RW 光盘结构与 CD-ROM 基本相同,只是在盘片中增加了可改写的染色层。读写数据采用相变技术。相变技术利用物质的状态变化进行数据的读、写和擦除。CD-RW 盘片在内部镀上了一层一定厚度的薄膜,即相变记录层。相变记录层由一种银合金材料组成,随着加热温度的不同,它可以形成晶体,也可以形成非晶体。因此,适当调整加热温度就可以自由地控制记录层的结晶状态。在晶体状态中,原子排列整齐,光反射率高;相反,在非晶体状态中,原子排列不整齐,光反射率低。对 CD-RW 的读、写和擦除正是利用光反射率的这种变化来实现的。由于材料的因素,晶体状态改变的次数是有限的,因此 CD-RW 盘片的擦写次数也是有限的。

CD-RW 盘片中相变记录层的记录膜在出厂时处于晶体状态,写入时用强的激光束照射,使之变为非晶体状态。如果此时中止激光照射,记录膜温度急剧下降,写入数据的区域便稳定在非结晶状态,数据被写入。读出时用弱的激光束照射记录区,并根据反射光的反射率判别是 0 还是 1。仅用弱光照射时,记录膜记录的数据不会被破坏,这与普通光驱读取光盘的原理是一样的。在擦除数据时,用中等强度的激光束照射记录膜,使其温度上升少许,记录膜又返回晶体状态,数据被擦除。

对 CD-RW 盘片的读写操作是通过 CD-RW 刻录机完成的。目前的 CD-RW 刻录机兼容 CD-ROM 和 CD-R 盘片,它分为内置式和外置式两种。在与主机连接上,内置式刻录机主要通过 IDE、SCSI 等接口连接,而外置式刻录机通过计算机的外部并行接口连接。

5. DVD

DVD 不仅可用来存储视频数据,还可以用来存储其他类型的数据,因此 DVD 又称为 Digital Versatile Disk,即数字通用盘,是一种能够保存视频、音频和计算机数据的光盘,它容量更大,运行速度更快,采用 MPEG2 压缩标准。

DVD 采用了类似 CD-ROM 的技术,但是可以提供更高的存储容量。从表面上看,DVD 盘片与 CD-ROM 盘片很相似,其直径为 80mm 或 120mm,厚度为 1.2mm。但实质上,两者之间有本质的差别。相对于 CD-ROM 光盘 650MB 的存储容量,DVD 光盘的存储容量高达 17GB。另外在读盘速度方面,CD-ROM 的单倍速传输速度是 150KB/s,而 DVD 的单倍速传输速度是 1358KB/s。

如图 5-39 是 DVD 和 CD-ROM 盘片数据记录道和凹坑密度比较。从图 5-39 中可以看出,CD-ROM 盘的道间距为 $1.6\mu\text{m}$,而 DVD 盘的道间距为 $0.74\mu\text{m}$; CD-ROM 盘的最小凹坑为 $0.83\mu\text{m}$,而 DVD 盘的最小凹坑为 $0.4\mu\text{m}$ 。DVD 盘片的道密度和凹坑密度都远高于 CD-ROM 盘片。单从这两方面的改进,就使 DVD 的单片单层容量提高到 CD-ROM 的 7 倍多,可达 4.7GB。

DVD 盘片分为单面单层、单面双层、双面单层和双面双层 4 种物理结构。因此,可以将 DVD 盘片分为 4 种规格,分别是 DVD-5、DVD-9、DVD-10 和 DVD-18,它们的容量分别如

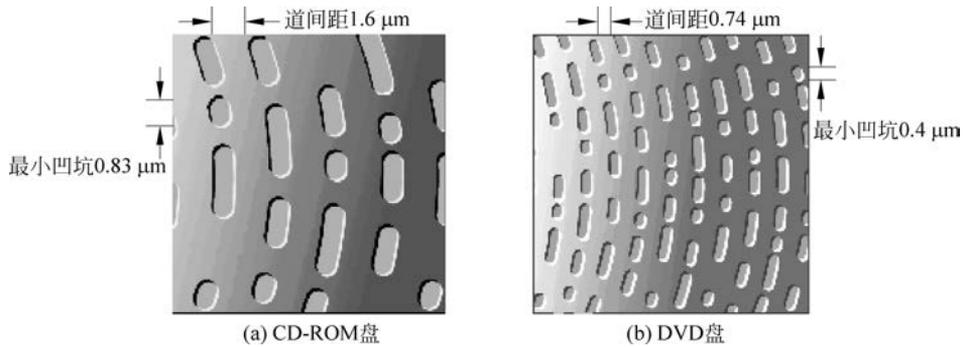


图 5-39 DVD 盘与 CD-ROM 盘片数据记录道和凹坑密度比较

表 5-8 所示。

表 5-8 4 种 DVD 盘片比较

盘片类型	盘片直径/cm	面数/层数	容量/GB
DVD-5	12	单面单层	4.7
DVD-9	12	单面双层	8.5
DVD-10	12	双面单层	9.4
DVD-18	12	双面双层	17

无论是单层盘还是双层盘,都是由两片基底组成,每片基底的厚度均为 0.6mm,因此 DVD 盘片的厚度为 1.2mm。对于单面盘来说,只有下层基底包含数据,上层基底没有数据;而双面盘的上下两层基底均包含数据。

从读写功能上,DVD 盘片也分为两类:只读型和可擦写型。只读型 DVD 包括用于计算机辅助存储器的 DVD-ROM 和用于存储音频、视频的 DVD-Audio 及 DVD-Video;可擦写型 DVD 主要有一次可刻录的 DVD-R 和多次可擦除、可改写的 DVD-RW 等。

5.4 RAID 技术

计算机技术发展到今天,人们已可以在包括微型计算机在内的各种机器中配置大容量磁盘。而在此之前,只有在一些大型计算机系统中根据其应用需要才配备大容量磁盘系统。那时,大容量磁盘的价格还相当高,对工作环境的要求也非常高。虽然那时磁盘的可靠性已很高,但对磁盘不恰当的使用仍然容易造成磁盘的损坏。一旦磁盘损坏,就会造成不可修复的磁盘错误,将使其在商业、科研及学术等领域造成不可估量的损失。因此使用单一的磁盘存储重要的数据总存在安全问题。解决数据存储安全问题的主要方法是使用磁带机等设备进行数据备份。这种方法虽然可以在一定程度上提高数据的安全性,但一方面数据的备份需要科学合理的安排和严格的制度来保障,另一方面今后对数据的检索工作相当烦琐。

1988 年,美国加州大学伯克利分校的 David Patterson、Garth Gibson 和 Randy Katz 三人发表了一篇题为 *A Case of Redundant Array of Inexpensive Disks* (《廉价磁盘冗余阵列方案》) 的论文,首次提出了 RAID 一词。在论文中他们提出将多个小容量、价格低廉的磁盘进行有机组合,来替代通常在大型计算机中使用的昂贵的大容量磁盘系统,并使其具有更好的性能和更高的可靠性。在这篇论文中,Patterson、Gibson 和 Katz 还定义了 5 种类型(称

为级,level)的 RAID,每一级 RAID 都具有不同的性能和可靠性。以前这些级的编号是从 1 到 5,后来人们又定义了 RAID 的第 0 级和第 6 级。所以 RAID 主要分为 7 个级,即从第 0 级到第 6 级。当然,一些研究机构和公司还定义了其他的一些 RAID 级,但 7 级 RAID 是业界普遍认同的标准。

总的来说,RAID 的设计思想是通过在多个硬盘上(又称为磁盘阵列)同时存取数据来大幅提高磁盘存储系统的数据吞吐率,而且一些 RAID 模式还通过较为完备的相互校验/恢复的措施,甚至是直接相互的镜像备份,来提高 RAID 系统的容错度,从而提高了磁盘存储系统的安全性和可靠性。RAID 的这一设计思想很快被接受,从此 RAID 技术得到了广泛应用,数据存储进入了更快速、更安全、更廉价的新时代。

表 5-9 对 RAID 分级进行了概括。

表 5-9 RAID 分级

RAID 级	描 述	磁盘数	容错性能	并行 I/O 响应
0	无容错的条带磁盘阵列	N	无容错	有
1	磁盘镜像	$2N$	容错性最好	有
2	专用汉明校验盘位分布磁盘系统	$N+m$	允许一个磁盘失效	无
3	专用奇偶校验盘位分布磁盘系统	$N+1$	允许一个磁盘失效	无
4	专用奇偶校验盘分块独立存取磁盘系统	$N+1$	允许一个磁盘失效	有
5	分散校验分块独立存取磁盘系统	N	允许一个磁盘失效	有
6	分散双校验分块独立存取磁盘系统	N	允许两个磁盘失效	有

1. RAID0

RAID0 的全称是 striped disk array without fault tolerance(无容错的条带磁盘阵列),其结构如图 5-40 所示。

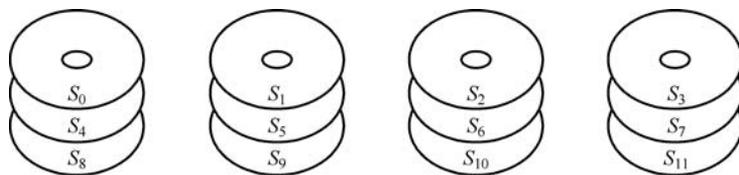


图 5-40 RAID0 的分条带数据组织

图 5-40 中,一个圆盘就是一个磁盘(以下同)。磁盘以条带(strip)的形式划分,每个条带是一些物理的块、扇区或其他单位。所有的磁盘组成一个逻辑磁盘,系统数据和用户数据存储在该逻辑磁盘上。逻辑磁盘上的一个个条带数据以轮转方式映射到连续的阵列磁盘中。例如,在一个由 N 个磁盘组成的阵列中,逻辑磁盘的第 $1 \sim N$ 个条带数据按顺序依次分布在第 $1 \sim N$ 个磁盘的第 1 个条带上,第 $N+1 \sim 2N$ 个条带数据按顺序依次分布在第 $1 \sim N$ 个磁盘的第 2 个条带上……这种布局的优点是,如果单个 I/O 请求由多个逻辑相邻的条带组成,则对多达 n 个条带的请求可以并行处理,这样可以大大提高 I/O 的数据传输率。

实际上,RAID0 有些类似在计算机主存中采用的交叉存储技术,通过使用多个磁盘的

并行存取提高主机对磁盘的访问速度。每个磁盘都有自己独立的访问控制电路,能够独立地进行数据的传输。而逻辑磁盘和物理磁盘间的数据映射则在磁盘阵列管理系统(由软件或硬件实现)的统一控制下完成。

不过,RAID0 不是 RAID 家族中的真正成员,因为它没有数据冗余能力。由于没有采用备份和校验恢复技术,RAID0 阵列中任何一个磁盘损坏都会导致整个磁盘阵列数据的损坏,因为数据都是分布存储的。而且,RAID0 磁盘阵列的整体可靠性会随着磁盘数量的增加而降低。例如,如果阵列由 5 个磁盘组成,每个磁盘的设计寿命为 50 000h,那么整个系统的期望设计寿命为 $50\,000/5=10\,000$ h。当磁盘数量进一步增加时,磁盘阵列失效的概率会随之增加,最后达到某个不确定的值。

2. RAID1

RAID1 又称为磁盘镜像(disk mirroring),是所有 RAID 级中具有最佳失效保护的一种方案。它使用两组互为镜像的磁盘进行简单的完全数据备份,从而实现数据冗余,其结构如图 5-41 所示。

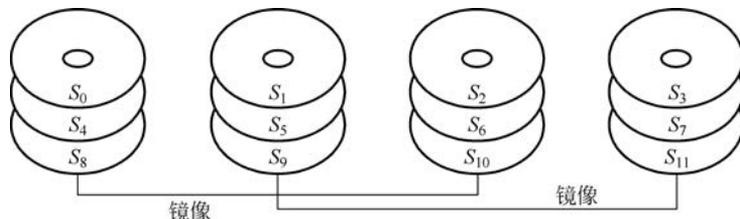


图 5-41 RAID1 的磁盘镜像

图 5-41 中包含了两组相同的磁盘阵列。RAID1 在每次写入时,都会同时将数据写入到两组磁盘中,使两组磁盘的数据保持完全相同,以实现磁盘阵列的高可靠性。

RAID1 也采用与 RAID0 相同的条带数据划分形式,即在每组内,所有磁盘以条带方式进行数据组织,以保持与 RAID0 同样的高性能。

与 RAID0 相比,RAID1 具有以下特点。

(1) 一个读请求可由包含请求数据的两组磁盘中的某一个提供服务,只要它的寻道时间加旋转延迟较小。因此,RAID1 具有比 RAID0 更优的读性能。尤其是在一些面向事务的应用场合,当需要对磁盘进行大量数据读取时,RAID1 的性能甚至能近似达到 RAID0 性能的两倍。

(2) 一个写请求需要更新两组磁盘中两个对应的条带,而这组磁盘的写操作可以并行进行。因此,写性能由两者中较慢的一个写操作来决定,即由寻道时间较长和旋转延迟较大的那一个写。

(3) 恢复一个损坏的磁盘很简单。当一个磁盘损坏时,数据仍能从与之镜像的磁盘中读取。

RAID1 的主要缺点是价格昂贵,它需要支持两倍于逻辑磁盘的磁盘空间。因此,RAID1 的配置常用在存储关键的系统数据和用户数据的场合中。在这种情况下,RAID1 对所有的数据提供实时备份,即使一个磁盘损坏,所有的关键数据仍能立即可用。

3. RAID2

RAID1 虽然同时具有高可靠性和高性能,但成本太高,需要整整两倍于实际所需的磁盘数量才能达到数据的冗余。更好的方式是只使用磁盘组中的一个或几个磁盘,用于数据冗余或数据校验。RAID2 就定义了这些方法中的一种。

RAID2 称为汉明码校验,也采用条带划分的方式。但它在每个条带中只写入 1 位二进制位,而不是采用像 RAID0 和 RAID1 中的数据块。因此,如果以字节为单位进行数据组织,则一个磁盘阵列中至少需要 8 个磁盘。RAID2 采用汉明纠错码进行数据校验。与数据磁盘相对应,磁盘阵列中还需要一组磁盘用于存储纠错码信息,如图 5-42 所示。

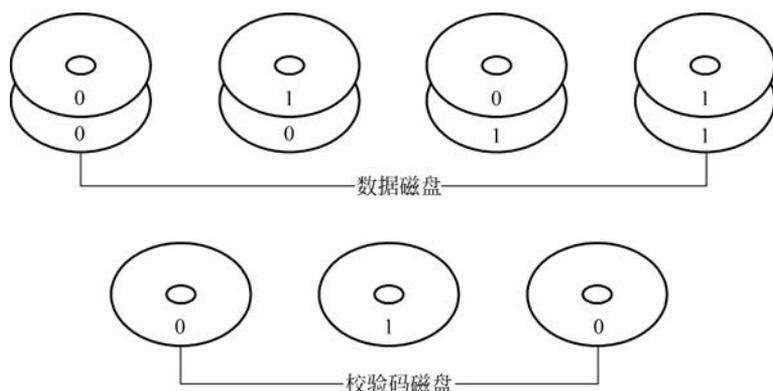


图 5-42 RAID2 的采用汉明码校验的分条带数据组织

纠错码所需的磁盘数量取决于汉明纠错码所需的校验位数。无论是数据盘还是校验盘,只要有一个磁盘损坏,其中的数据都可以通过汉明纠错码来重建恢复。

因为每个磁盘都只写入 1 位数据,因此整个 RAID2 的磁盘组就好像是一个大型的数据磁盘。所有的磁盘数据(包括数据磁盘和校验码磁盘)都必须严格地同步,否则数据变乱,使汉明码起不到校验的作用。由于系统对磁盘的读写是按位并行执行的,因此数据传输率高。但由于生成汉明码较为耗时,所以 RAID2 对大多数商业应用来说速度太慢。事实上,今天大多数磁盘驱动器都有内置的 CRC 纠错功能。在单个磁盘和驱动器具有高可靠性的情况下,RAID2 就没有太大的应用意义了。

4. RAID3

RAID3 称为带校验的并行传输。像 RAID2 一样,RAID3 按照每次 1 位的方式将数据交错分配到各个数据盘的条带上。但是,与 RAID2 不同的是,RAID3 只使用一个磁盘来存储一个简单的奇偶校验位,如图 5-43 所示。这种奇偶校验位可以使用简单的硬件快速计算出来,具体方法是对分布在每一个磁盘上的每一个数据位(D_i)进行异或运算(对偶校验),生成的偶校验位存储到校验盘上。例如,假设磁盘组共有 8 个数据磁盘,分布在这 8 个盘上的数据位分别为 $D_0 \sim D_7$,则偶校验位为:

$$P = D_0 \oplus D_1 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_5 \oplus D_6 \oplus D_7$$

其中, P 位存储在校验盘中。

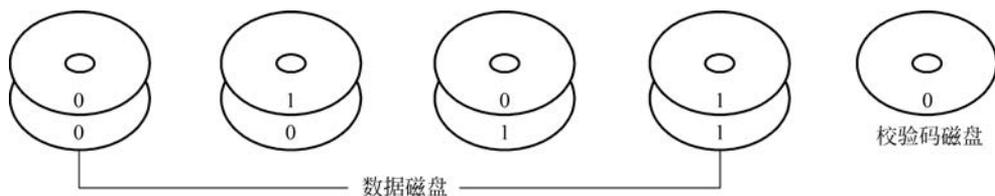


图 5-43 RAID3 的采用奇偶校验码的分条带数据组织

利用相同的方法可以在确认了某一磁盘损坏的情况下,很方便地对该磁盘数据进行重建。例如,假设图 5-43 中的 5 号磁盘损坏了,并且被替换掉,用户可以通过对其他剩余的数据磁盘和校验码磁盘的对应数据进行如下偶校验运算,即可重新生成损坏了的磁盘上的数据:

$$D_5 = D_0 \oplus D_1 \oplus D_2 \oplus D_3 \oplus D_4 \oplus D_6 \oplus D_7 \oplus P$$

当然,RAID3 也要求使用与 RAID2 同样的数据复制方法和同步操作,以免造成数据的混乱。由于只使用了一个校验盘用于数据的恢复,因此,RAID3 比 RAID1 和 RAID2 都更经济。

5. RAID4

RAID2 和 RAID3 都采用将数据以位为单位分布到各个数据磁盘上的方式,对数据的读写必须在所有盘上同时进行。因此,RAID2 和 RAID3 不适合多个并行 I/O 的响应。而从 RAID4 到 RAID6 都采用了一种独立的存取方式,磁盘阵列中每个磁盘的操作都是独立的,从而也就能同时响应多个 I/O 请求。

RAID4 又称为带有共享校验磁盘的独立数据磁盘系统。它与 RAID3 一样,同样采用“数据磁盘+校验码磁盘”的组织形式,如图 5-44 所示。但与 RAID3 不同的是,RAID4 不是以位为单位进行数据的读写。它将所有磁盘划分成大小相同的条带,每个条带都能存储一个块的数据。

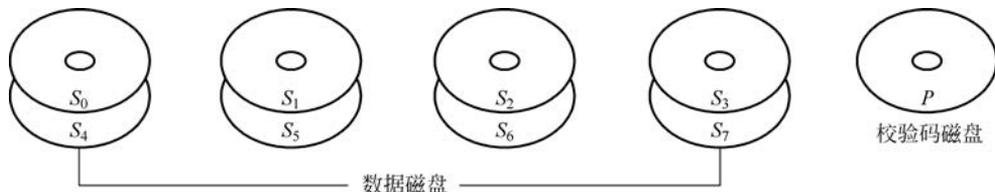


图 5-44 RAID4 的带奇偶校验磁盘的独立数据磁盘组织

校验方法是每次对每个数据盘上对应条带的同一位进行奇偶校验运算,生成该位的奇偶校验位。对应一个数据条带,就生成一个奇偶校验条带。例如,假设某个由 4 张磁盘组成的 RAID4 磁盘系统的条带单位是字节,4 个磁盘对应的条带分别为 S_0 、 S_1 、 S_2 、 S_3 ,与该条带对应的偶校验条带为 P ,则:

$$\begin{aligned} P(1) &= S_0(1) \oplus S_1(1) \oplus S_2(1) \oplus S_3(1) \\ P(2) &= S_0(2) \oplus S_1(2) \oplus S_2(2) \oplus S_3(2) \\ &\vdots \\ P(8) &= S_0(8) \oplus S_1(8) \oplus S_2(8) \oplus S_3(8) \end{aligned}$$

RAID4 的这种奇偶校验组织方式能有效地解决对并行 I/O 的响应问题,但它的写效率较低。对于每一次写操作,磁盘阵列管理系统不仅要修改用户数据,而且要修改相应的奇偶校验位。对于上例来说,假设某次写操作只在 S_1 条带上执行。

初始时,对每位 i 有下列关系式:

$$P(i) = S_0(i) \oplus S_1(i) \oplus S_2(i) \oplus S_3(i)$$

修改后,可能改变的位以撇号“'”表示:

$$\begin{aligned} P(i) &= S_0(i) \oplus S_1'(i) \oplus S_2(i) \oplus S_3(i) \\ &= S_0(i) \oplus S_1'(i) \oplus S_2(i) \oplus S_3(i) \oplus S_1(i) \oplus S_1(i) \\ &= S_0(i) \oplus S_1(i) \oplus S_2(i) \oplus S_3(i) \oplus S_1(i) \oplus S_1'(i) \\ &= P(i) \oplus S_1(i) \oplus S_1'(i) \end{aligned}$$

可以看出,为了计算新的奇偶校验位,系统必须读取旧的数据条带和奇偶校验条带,然后用新的数据和新计算出的奇偶校验位修改上述两个条带。因此,每个条带的写操作包括两次读和两次写,写操作效率比较低。

从 RAID4 的架构和写操作可以看出,RAID4 的校验码磁盘是整个系统的关键,它的失效会带来整个系统的失效。正是这个原因限制了 RAID4 的实际应用。

6. RAID5

RAID5 的全称是带分散校验码磁盘的条带数据磁盘系统,它是在 RAID4 基础上加以改进的优化方案。对 RAID4 而言,校验码磁盘是独立的,它的失效将带来整个磁盘系统的失效,因此校验码磁盘成为 RAID4 的瓶颈。RAID5 则是将奇偶检验分散到整个磁盘阵列中,如图 5-45 所示。

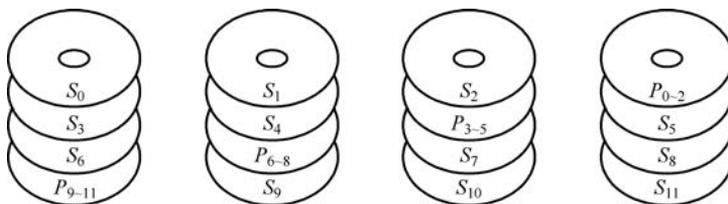


图 5-45 RAID5 的带分散校验盘的分条带数据磁盘组织

在图 5-45 中,条带 (S_0, S_1, S_2) 、 (S_3, S_4, S_5) 、 (S_6, S_7, S_8) 、 (S_9, S_{10}, S_{11}) 的对应校验条带 P_{0-2} 、 P_{3-5} 、 P_{6-8} 和 P_{9-11} 分布在不同的磁盘上。这样一方面保存了 RAID4 原有的独立磁盘所带来的对并行 I/O 的响应能力,另一方面又较好地解决了 RAID4 的校验码磁盘瓶颈问题。但相对来说,RAID5 的磁盘控制功能是最复杂的。

RAID5 是目前在商业上得到了很好应用的 RAID 方案之一。

7. RAID6

前面所讨论的大多数 RAID 系统只能允许最多有一个磁盘失效,但实际情况是大型计算机系统的磁盘常常会发生成群成簇的失效。发生这种情况一般有两原因:一是几乎同一时间生产的磁盘会在相同的时间里到达它们预期的使用寿命;二是磁盘的损坏通常是某些灾难性事件引起的,比如电源故障等。如果在第一个损坏的磁盘还未来得及更换之前第

二个磁盘又损坏了,那么整个磁盘系统将崩溃。RAID6 就是为解决这种问题而提出的一种方案。

RAID6 是由一些大型企业提出来的私有 RAID 级别标准,全称为带有两个独立分布式校验方案的独立数据磁盘。它是在 RAID5 的基础上发展而成的,因此它的工作模式与 RAID5 有相似之处。所不同的是 RAID5 只使用了一种奇偶校验码,并只写入到一个磁盘上;而 RAID6 除使用奇偶校验码外,还使用 Reed-Soloman 纠错编码来提供第二层保护,并将这两种校验码写入到两个不同的磁盘上面。这样就增强了磁盘的容错能力,允许磁盘阵列中出现故障的磁盘可以达到两个。图 5-46 是 RAID6 的组织架构图。

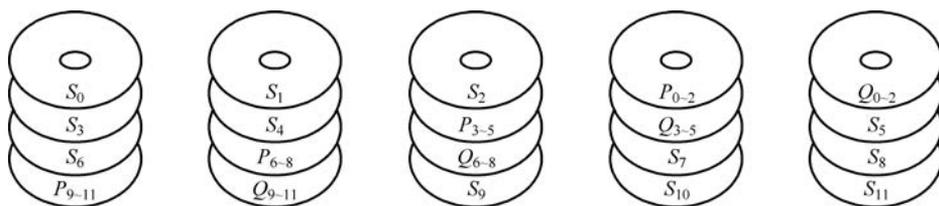


图 5-46 RAID6 的带有两个独立分布式校验方案的独立数据磁盘组织

虽然 RAID6 有更强的容错能力,但直到最近,还没有实际商业配置的 RAID6 系统。这主要有两个方面的原因:一是生成 Reed-Soloman 编码需要相当大的费用代价;二是在 RAID6 系统中,更新磁盘上的纠错编码要求双倍的读写操作。

8. 混合 RAID 系统

许多大型的计算机系统并不局限于只使用一种类型的 RAID。在某些情况下,平衡磁盘系统的高可用性和经济性是非常重要的。例如,人们可能希望使用 RAID1 来保护操作系统文件等重要数据,而用户数据文件使用 RAID5 就足够了;RAID0 非常适合于存放大程序运行过程中产生的临时文件,并且 RAID0 高效并行的 I/O 访问能力特别适合于频繁磁盘操作的多进程运行。

用户还可以将多个 RAID 方案组合起来构建一种“新型”的 RAID。其中 RAID10 就是这样一种磁盘系统,它组合了 RAID0 的条带方式和 RAID1 的镜像功能。虽然代价昂贵,但是 RAID10 可以提供优良的读取性能和最佳的可用性。

最后,通过表 5-10 总结 RAID0~RAID6 的优缺点及适应的应用场合。

表 5-10 RAID0~RAID6 的优缺点及应用

RAID 级	优 点	缺 点	应 用
RAID0	(1) 支持并行 I/O 的处理; (2) 不会因为校验、容错等占用磁盘及 CPU 资源; (3) 使用和配置简单	(1) 缺乏校验恢复机制; (2) 没有磁盘数据容错能力; (3) 随磁盘数的增加可靠性降低	视频生成、图像处理以及其他并行 I/O 处理频度较高的应用
RAID1	(1) 具有最佳失效保护功能; (2) 对磁盘的读取速度最快; (3) 使用和配置简单	占用的冗余磁盘最多	一些需要数据高可靠性的行业,如金融、政府部门等

续表

RAID 级	优 点	缺 点	应 用
RAID2	(1) 即时数据校验能力,具有较好的数据恢复功能; (2) 并行数据传输速度快,但一次只能执行一个 I/O 请求	(1) ECC 占用比例较大; (2) ECC 的生成成本高且费时,导致整个磁盘阵列读写速度慢	目前的实际应用较少
RAID3	(1) 简单高效的校验功能,具有较好的数据恢复功能; (2) 并行数据传输速度快,但一次只能执行一个 I/O 请求	磁盘控制器设计比较复杂	视频生成、图像处理以及其他需要较高数据传输率的应用
RAID4	(1) 高效率的数据校验功能; (2) 支持并行 I/O 的处理	(1) 磁盘控制器设计非常复杂; (2) 数据恢复及重建工作复杂; (3) 写效率低; (4) 校验码磁盘成为系统瓶颈	在商业上没有得到应用
RAID5	(1) 高效率的数据校验功能; (2) 支持并行 I/O 的处理	磁盘控制器设计非常复杂	实际应用较广,包括应用于各种数据库服务器、文件和应用服务器等
RAID6	(1) 最强的磁盘容错功能,允许两个磁盘同时失效; (2) 支持并行 I/O 的处理	(1) 磁盘控制器设计非常复杂; (2) 写效率很低; (3) 校验码占用磁盘空间较多	目前的实际应用较少

知识拓展

网络存储系统

随着 Internet 的发展,网络上大量的数据需要存储,人们对更加简便、快速、安全地存储数据的需求、对存储系统的容量和速度提出了空前的要求。传统的以服务器为中心的 DAS 方式(将 RAID 硬盘阵列直接安装到网络系统的服务器上)已不能满足用户的需要,越来越多的用户已经从原来的“以服务器为中心”模式转变为“以数据为中心”的网络存储模式,这其中使用较为广泛的是 NAS 和 SAN。

NAS(network attached storage,网络直接存储)的架构如图 5-47(a)所示。在 NAS 存储结构中,存储系统不再通过 I/O 总线附属于某个特定的服务器或客户机,它完全独立于网络中的主服务器,可以看作一个专用的数据或文件服务器。NAS 包括处理器、磁盘系统和数据管理模块,和服务器、客户机一起连接成网络,客户机对数据的访问不再需要服务器的干预,允许客户机与 NAS 之间进行直接的数据访问。

SAN(storage area network,存储区域网络)的架构如图 5-47(b)所示。在 SAN 结构中,各种存储设备通过光纤及光纤通道交换机与服务器一起连接成高速网络,使与服务器连接成 LAN 的客户机可以非常方便地访问存储设备。在 SAN 结构中,每一个应用服务器上都应安装文件管理系统,客户机对存储系统中文件的访问必须通过服务器进行。

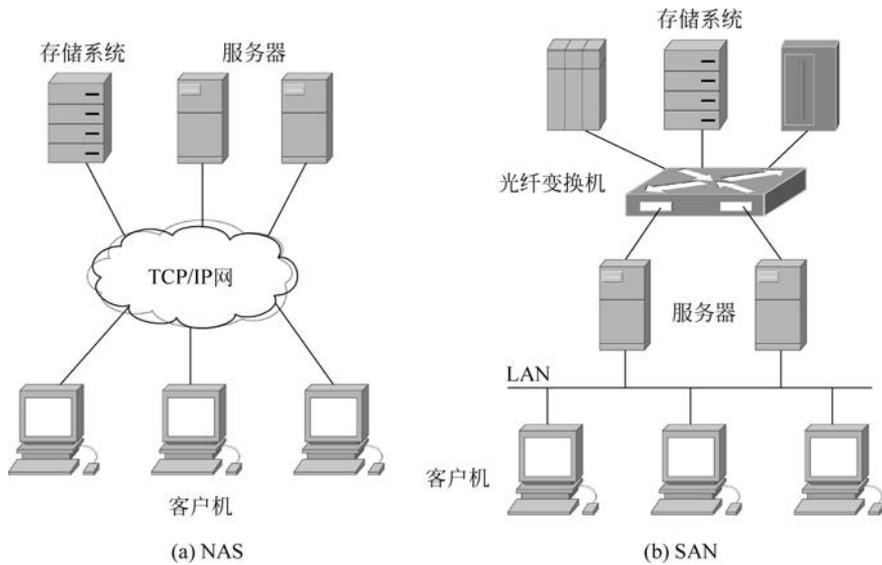


图 5-47 网络存储系统

本章小结

本章主要讲述了以下内容。

- (1) 输入输出接口的组成与功能, CPU 对输入输出设备的编址和管理。
- (2) 输入输出的传输控制方式, 包括程序控制方式、中断控制方式、DMA 控制方式和通道方式等。这些不同的控制方式在 CPU 利用率、数据传输效率等方面各有不同, 分别应用于对不同设备的输入输出控制。
- (3) 外部存储器的组织, 包括磁盘存储器、磁带存储器和光盘存储器的数据组织及工作原理。
- (4) RAID 技术, 包括 RAID1 到 RAID6 的磁盘组织、校验及冗余技术的应用等。

习题

一、名词解释

- | | | | |
|------------|------------|-------------|----------|
| 1. 外围设备 | 2. I/O 接口 | 3. I/O 端口 | 4. 中断 |
| 5. 中断向量 | 6. 单级中断 | 7. 多级中断 | 8. 中断嵌套 |
| 9. DMA | 10. 通道 | 11. 磁道 | 12. 扇区 |
| 13. 磁盘存储密度 | 14. 磁盘访问时间 | 15. 磁盘数据传输率 | 16. RAID |

二、选择题

1. 中断向量地址是()。

A. 子程序入口地址	B. 中断服务程序入口地址的地址
C. 主程序地址	D. 中断返回地址

2. 为了便于实现多级中断,保存现场信息最有效的方法是采用()。
- A. 通用寄存器 B. 堆栈 C. 辅存 D. 通道
3. 如果有多个中断同时发生,系统将根据中断优先级响应优先级最高的中断请求。若要调整中断事件的中断处理次序,可以利用()。
- A. 中断嵌套 B. 中断向量 C. 中断响应 D. 中断屏蔽
4. 外部中断包括不可屏蔽中断(NMI)和可屏蔽中断。下列关于外部中断的叙述中,错误的是()。(2020年全国硕士研究生入学统一考试计算机学科专业基础综合试题)
- A. CPU处于关中断状态时,也能响应NMI请求
B. 一旦可屏蔽中断请求信号有效,CPU将立即响应
C. 不可屏蔽中断的优先级比可屏蔽中断的优先级高
D. 可通过中断屏蔽字改变可屏蔽中断的处理优先级
5. 异常是指令执行过程中在处理器内部发生的特殊事件,中断是来自处理器外部的请求事件。下列关于中断或异常情况的叙述中,错误的是()。(2016年全国硕士研究生入学统一考试计算机学科专业基础综合试题)
- A. “访存时缺页”属于中断 B. “整数除以0”属于异常
C. “DMA传送结束”属于中断 D. “存储保护错”属于异常
6. 内部异常(内中断)可分为故障(fault)、陷阱(trap)和终止(abort)三类。下列有关内部异常的叙述中,错误的是()。(2015年全国硕士研究生入学统一考试计算机学科专业基础综合试题)
- A. 内部异常的产生与当前执行指令相关
B. 内部异常的检测由CPU内部逻辑实现
C. 内部异常的响应发生在指令执行过程中
D. 内部异常处理后返回到发生异常的指令处继续执行
7. 下列是关于多重中断系统中CPU响应中断的叙述,其中错误的是()。(2021年全国硕士研究生入学统一考试计算机学科专业基础综合试题)
- A. 仅在用户态(执行用户程序)下,CPU才能检测和响应中断
B. CPU只有在检测到中断请求信号后,才会进入中断响应周期
C. 进入中断响应周期时,CPU一定处于中断允许(开中断)状态
D. 若CPU检测到中断请求信号,则一定存在未被屏蔽的中断源请求信号
8. 在采用中断I/O方式控制打印输出的情况下,CPU和打印控制接口中的I/O端口之间交换的信息不可能是()。(2015年全国硕士研究生入学统一考试计算机学科专业基础综合试题)
- A. 打印字符 B. 主存地址 C. 设备状态 D. 控制命令
9. 在采用DMA方式高速传输数据时,数据传送是()。
- A. 在总线控制器发出的控制信号控制下完成的
B. 在DMA控制器本身发出的控制信号的控制下完成的
C. 由CPU执行的程序完成的
D. 由CPU响应硬中断处理完成的

10. 下列陈述中正确的是()。
- A. 在 DMA 周期内,CPU 不能执行程序
 - B. 中断发生时,CPU 首先执行入栈指令,将程序计数器的内容保护起来
 - C. 在 DMA 传送方式中,DMA 控制器每传送一个数据就窃取一个指令周期
 - D. 输入输出操作的最终目的是实现 CPU 与外设之间的数据传输
11. 若设备采用周期挪用 DMA 方式进行输入和输出,每次 DMA 传送的数据块为 512 字节,相应的 I/O 接口中有一个 32 位的数据缓冲寄存器。对于数据输入过程,下列叙述中错误的是()。(2020 年全国硕士研究生入学统一考试计算机学科专业基础综合试题)
- A. 每准备好 32 位数据,DMA 控制器就发出一次总线请求
 - B. 相对于 CPU,DMA 控制器总线使用权的优先级更高
 - C. 在整个数据块的传送过程中,CPU 不可以访问主存储器
 - D. 数据块传送结束时,会产生“DMA 传送结束”中断请求
12. 一般来讲,主机与硬盘之间的数据交换比软盘快,其主要原因是()。
- A. 硬盘有多个盘片
 - B. 硬盘采用密封式安装
 - C. 硬盘使用的是铝片等“硬”基质,转速快
 - D. 硬盘容量更大
13. 以下可用作数据备份之用的光盘是()。
- A. CD-R
 - B. CD-ROM
 - C. CD-RW
 - D. CD
14. 以下不具有校验恢复机制的 RAID 技术是()。
- A. RAID0
 - B. RAID1
 - C. RAID2
 - D. RAID3
15. 下列有关 I/O 接口的叙述中,错误的是()。(2014 年全国硕士研究生入学统一考试计算机学科专业基础综合试题)
- A. 状态端口和控制端口可以合用同一寄存器
 - B. I/O 接口中 CPU 可访问的寄存器称为 I/O 端口
 - C. 采用独立编址方式时,I/O 端口地址和主存地址可能相同
 - D. 采用统一编址方式时,CPU 不能用访存指令访问 I/O 端口
16. 某设备中断请求的响应和处理时间为 100ns,每 400ns 发出一次中断请求,中断响应所容许的最长延迟时间为 50ns,则在该设备持续工作过程中 CPU 用于该设备的 I/O 时间占整个 CPU 时间的百分比至少是()。(2014 年全国硕士研究生入学统一考试计算机学科专业基础综合试题)
- A. 12.5%
 - B. 25%
 - C. 37.5%
 - D. 50%
17. 下列选项中,用于提高 RAID 可靠性的措施有()。(2013 年全国硕士研究生入学统一考试计算机学科专业基础综合试题)
- I. 磁盘镜像
 - II. 条带化
 - III. 奇偶校验
 - IV. 增加 cache 机制
- A. 仅 I、II
 - B. 仅 I、III
 - C. 仅 I、III 和 IV
 - D. 仅 II、III 和 IV
18. 若磁盘转速为 7200 转/分,平均寻道时间为 8ms,每个磁道包含 1000 个扇区,则访问一个扇区的平均存取时间大约是()。(2015 年全国硕士研究生入学统一考试计算机

学科专业基础综合试题)

- A. 8.1ms B. 12.2ms C. 16.3ms D. 20.5ms

19. 某磁盘的转速为 10 000 转/分,平均寻道时间是 6ms,磁盘传输速率是 20MB/s,磁盘控制器的延迟为 0.2ms,读取一个 4KB 的扇区所需的平均时间约为()。(2013 年全国硕士研究生入学统一考试计算机学科专业基础综合试题)

- A. 9ms B. 9.4ms C. 12ms D. 12.4ms

20. 下列关于中断 I/O 方式和 DMA 方式比较的叙述中,错误的是()。(2013 年全国硕士研究生入学统一考试计算机学科专业基础综合试题)

- A. 中断 I/O 方式请求的是 CPU 处理时间,DMA 方式请求的是总线使用权
B. 中断响应发生在一条指令执行结束后,DMA 响应发生在一个总线事务完成后
C. 中断 I/O 方式下数据传送通过软件完成,DMA 方式下数据传送由硬件完成
D. 中断 I/O 方式适用于所有外部设备,DMA 方式仅适用于快速外部设备

21. 在响应外部中断的过程中,中断隐指令完成的操作除保护断点外,还包括()。(2012 年全国硕士研究生入学统一考试计算机学科专业基础综合试题)

- I. 关中断 II. 保存通用寄存器的内容 III. 形成中断服务程序入口地址并送 PC
A. 仅 I、II B. 仅 I、III C. 仅 II、III D. I、II、III

三、综合题

- I/O 接口一般由哪些部件组成? I/O 接口主要应实现哪些功能?
- 对 I/O 端口的编址主要有哪两种方式?
- 计算机都有哪些输入输出传输控制方式? 这些方式各有何特点? 分别应用于什么样的场合?
- 计算机系统中断源是怎样分类的? 都有哪些中断源?
- 简述中断系统的中断处理过程。
- 中断响应所实现的功能主要有哪些?
- 简述采用中断向量法的中断响应过程。
- 简述 DMA 的传输控制过程。
- 通道与 DMA 有什么不同?
- 通道主要有哪几种类别? 它们各有何特点? 分别适合与什么样的设备连接?
- 光盘主要有哪几种类别?
- 设某机器中断系统有 5 级中断 A、B、C、D、E,其中断响应优先级由高到低顺序为 A→B→C→D→E,现要求将中断处理次序改为 B→D→A→E→C。

试问:

(1) 表 5-11 中各级中断处理程序的各中断屏蔽码如何设置(“1”表示屏蔽中断,“0”表示开放中断)?

(2) 若在执行一个 CPU 程序时这 5 级中断同时都发出中断请求,按更改后的次序画出各级中断响应和中断处理的过程示意图。

表 5-11 中断屏蔽码

中断处理程序	中断屏蔽码				
	A	B	C	D	E
A					
B					
C					
D					
E					

13. 假定计算机的主频为 500MHz, CPI 为 4。现有设备 A 和 B, 其数据传输率分别为 2MB/s 和 40MB/s, 对应的 I/O 接口中各有一个 32 位数据缓冲寄存器。请回答下列问题, 要求给出计算过程。(2018 年全国硕士研究生入学统一考试计算机学科专业基础综合试题)

(1) 若设备 A 采用定时查询 I/O 方式, 每次输入输出都至少执行 10 条指令。设备 A 最多间隔多长时间查询一次才能不丢失数据? CPU 用于设备 A 输入输出的时间占 CPU 总时间的百分比至少是多少?

(2) 在中断 I/O 方式下, 若每次中断响应和中断处理的总时钟周期数至少为 400, 则设备 B 能否采用中断 I/O 方式? 为什么?

(3) 若设备 B 采用 DMA 方式, 每次 DMA 传送的数据块为 1000B, CPU 用于 DMA 预处理和后处理的总时钟周期数为 500, 则 CPU 用于设备 B 输入输出的时间占 CPU 总时间的百分比最多是多少?

14. 设某磁盘存储器的转速为 7200 转/分, 平均寻道时间为 10ms, 每道存储容量为 600KB, 求磁盘的平均存取时间和数据传输率。

15. 设某磁盘存储器的转速为 3000 转/分, 共有 18 个记录面, 每面有 20 000 个磁道, 每道存储容量为 300KB, 每道扇区数为 600。

试求:

- (1) 磁盘存储器的总存储容量。
- (2) 磁盘数据传输率。
- (3) 磁盘的平均等待时间。
- (4) 每扇区的容量。
- (5) 给出一个磁盘地址格式方案。

16. 设某磁带机有 9 个磁道, 带长为 600m, 带速为 2m/s, 每个数据块大小为 1KB, 块间隔为 14mm。若数据传输率为 128 000B/s。

试求:

- (1) 磁带的记录密度。
- (2) 若带的首尾各空 2m, 求磁带的最大有效存储容量。

17. 有一台磁盘机, 其平均寻道时间为 30ms, 平均等待时间为 10ms, 数据传输率为 500KB/ms, 磁盘机上存放着 1000 个文件, 每个文件大小为 3000KB。现需要将所有文件逐个取走, 更新后再放回原处(文件大小不变)。已知每个文件的更新时间为 4ms, 且更新操作与磁盘读写不相重叠。

试问：

(1) 更新磁盘上所有文件并重新写回磁盘共需多少时间？

(2) 若磁盘机转速提高一倍,则更新磁盘上所有文件并重新写回磁盘共需多少时间？

18. RAID 有哪些分级? 各有什么特点?

19. 设某单位要构建一个双机磁盘冗余阵列系统(使用 RAID1),试查阅资料,给出一个具体实现的方案。

20. 某计算机的 CPU 主频为 500MHz, CPI 为 5(即执行每条指令平均需 5 个时钟周期)。假定某外设的数据传输率为 0.5MB/s,采用中断方式与主机进行数据传送,以 32 位为传输单位,对应的中断服务程序包含 18 条指令,中断服务的其他开销相当于 2 条指令的执行时间。请回答下列问题,要求给出计算过程。(2009 年全国硕士研究生入学统一考试计算机学科专业基础综合试题)

(1) 在中断方式下,CPU 用于该外设 I/O 的时间占整个 CPU 时间的百分比是多少?

(2) 当该外设的数据传输率达到 5MB/s 时,改用 DMA 方式传送数据。假设每次 DMA 传送的数据为 5000B,且 DMA 预处理和后处理的总开销为 500 个时钟周期,则 CPU 用于该外设 I/O 的时间占整个 CPU 时间的百分比是多少?(假设 DMA 与 CPU 之间没有访存冲突)