

网络接口、鼠标（用作指点设备）、老鼠（用作心理学实验室实验），以及大多数与磁盘不同的设备都可看作是字符设备。

这种分类方法并不完美，有些设备就没有包括进去。例如，时钟既不是块可寻址的，也不产生或接收字符流。它所做的工作就是按照预先规定好的时间间隔产生中断。内存映射的显示器也不适用于此模型。但是，块设备和字符设备的模型具有足够的一般性，可以用作使处理I/O设备的某些操作系统软件具有设备无关性的基础。例如，文件系统只处理抽象的块设备，而把与设备相关的部分留给较低层的软件。

I/O设备在速度上覆盖了巨大的范围，要使软件在跨越这么多数量级的数据率下保证性能优良，给软件造成了相当大的压力。图5-1列出了某些常见设备的数据率，这些设备中大多数随着时间的推移而变得越来越快。

### 5.1.2 设备控制器

I/O设备一般由机械部件和电子部件两部分组成。通常可以将这两部分分开处理，以提供更加模块化和更加通用的设计。电子部件称作设备控制器（device controller）或适配器（adapter）。在个人计算机上，它经常以主板上的芯片的形式出现，或者以插入（PCI）扩展槽中的印刷电路板的形式出现。机械部件则是设备本身。这一安排如图1-6所示。

控制器卡上通常有一个连接器，通向设备本身的电缆可以插入到这个连接器中。很多控制器可以操作2个、4个甚至8个相同的设备。如果控制器和设备之间采用的是标准接口，无论是官方的ANSI、IEEE或ISO标准还是事实上的标准，各个公司都可以制造各种适合这个接口的控制器或设备。例如，许多公司都生产符合IDE、SATA、SCSI、USB或火线（IEEE 1394）接口的磁盘驱动器。

控制器与设备之间的接口通常是一个很低层次的接口。例如，磁盘可以按每个磁道10 000个扇区，每个扇区512字节进行格式化。然而，实际从驱动器出来的却是一个串行的位（比特）流，它以一个前导符（preamble）开始，接着是一个扇区中的4096位，最后是一个校验和，也称为错误校正码（Error-Correcting Code, ECC）。前导符是在对磁盘进行格式化时写上去的，它包括柱面数和扇区号、扇区大小以及类似的数据，此外还包含同步信息。

控制器的任务是把串行的位流转换为字节块，并进行必要的错误校正工作。字节块通常首先在控制器内部的一个缓冲区中按位进行组装，然后在对校验和进行校验并证明字节块没有错误后，再将它复制到主存中。

在同样低的层次上，监视器的控制器也是一个位串行设备。它从内存中读入包含待显示字符的字节，并产生用来调制CRT电子束的信号，以便将结果写到屏幕上。该控制器还产生信号使CRT电子束在完成一行扫描后做水平回扫，并且产生信号使CRT电子束在整个屏幕扫描结束后做垂直回扫。如果没有CRT控制器，那么操作系统程序员只能对显像管的模拟扫描直接进行编程。有了控制器，操作系统就可以用几个参数（这些参数包括每行的字符数或像素数、每屏的行数等）对其初始化，并让控制器实际驱动电子束。平板TFT显示器的工作原理与此不同，但是也同样复杂。

### 5.1.3 内存映射I/O

每个控制器有几个寄存器用来与CPU进行通信。通过写入这些寄存器，操作系统可以命令设备发送数据、接收数据、开启或关闭，或者执行某些其他操作。通过读取这些寄存器，操作系统可以了解设备的状态，是否准备好接收一个新的命令等。

除了这些控制寄存器以外，许多设备还有一个操作系统可以读写的数据缓冲区。例如，在屏幕上显示像素的常规方法是使用一个视频RAM，这一RAM基本上只是一个数据缓冲区，可供程序或操作系统写入数据。

于是，问题就出现了：CPU如何与设备的控制寄存器和数据缓冲区进行通信？存在两个可选的方法。在第一个方法中，每个控制寄存器被分配一个I/O端口（I/O port）号，这是一个8位或16位的整数。所有I/O端口形成I/O端口空间（I/O port space），并且受到保护使得普通的用户程序不能对其进行访问（只有操作系统可以访问）。使用一条特殊的I/O指令，例如

```
IN REG, PORT
```

CPU可以读取控制寄存器PORT的内容并将结果存入到CPU寄存器REG中。类似地，使用