IDE接口硬盘读写技术

徐小玲*

(浙江教育学院计算机系 杭州 310012)

【摘要】分析了IDE接口硬盘控制寄存器模型；论述了IDE接口硬盘的读写几项技术；给出了设计硬盘克隆软件的思想和方法，方法针对硬盘物理扇区进行读写，与硬盘上安装的具体操作系统的类型无关，并与硬盘驱动器的物理结构无关。结合C语言与汇编语言，经实际应用验证，方法简便，具有较强的实用性。

关 键 词 IDE接口； 硬盘控制寄存器； LBA寻址； ATA标准

中图分类号 TP302

Technology of Read-Write IDE Interface Hard Disk

Xu Xiaoling

(Department of Computers, ZheJiang Education College HangZhou 310012)

Abstract This paper analyse the model of controller registers for IDE interface hard disk drive, summarized some key technic about read-write hard disk drive and a design about clone hard disk. We can directly read-write the sector ,and it has no concern with operating system and the structure of hard disk drive. Combine with C and Assemble language, We give an realy application . the method is easy and useful.

Key words IDE interface; hard disk control register; LBA seek; ATA standard

硬盘读写是一个复杂的过程，它涉及到硬盘的接口方式、寻址方式、控制寄存器模型等。硬盘的存储介质经历了从磁性材料、光磁介质到Flash半导体存储材料，对它们的读写方法和寻址方式都一样，因为这些存储介质与计算机的接口共同遵循着ATA标准。主机与硬盘之间的数据传输按程序I/O或DMA方式进行，硬盘的寻址方式可按CHS或LBA。在计算机应用中，掌握硬盘读写技术很有必要， 像UNIX系统的dd命令和目前流行的Ghost、DiskEdit等软件，都可以把数十个GB容量硬盘上庞大的软件系统，在短时间内复制完成。这些工具软件的构造正是基于该技术而设计的。本文从IDE控制器的寄存器模型入手，分析硬盘的读写方法和寻址方式，结合实例剖析了这类复杂硬盘工具软件的设计思路及制作方法。

1 IDE控制器的寄存器模型

计算机主机对IDE接口硬盘的控制是通过硬盘控制器上的二组寄存器实现[1]。一组为命令寄存器组(Task File Registers)，I/O的端口地址为1F0H~1F7H，其作用是传送命令与命令参数，如表1所示。另一组为控制/诊断寄存器(Control/Diagnostic Registers)，I/O的端口地址为3F6H~3F7H，其作用是控制硬盘驱动器，如表2所示。

2002年4月23日收稿

* 女 39岁 大学 讲师

表1　Task File Registers命令寄存器组

I/O地址	读(主机从硬盘读数据)	写(主机数据写入硬盘)
1F0H	数据寄存器	数据寄存器
1F1H	错误寄存器(只读寄存器)	特征寄存器
1F2H	扇区计数寄存器	扇区计数寄存器
1F3H	扇区号寄存器或LBA块地址0~7	扇区号或LBA块地址0~7
1F4H	磁道数低8位或LBA块地址8~15	磁道数低8位或LBA块地址8~15
1F5H	磁道数高8位或LBA块地址16~23	磁道数高8位或LBA块地址16~23
1F6H	驱动器/磁头或LBA块地址24~27	驱动器/磁头或LBA块地址24~27
1F7H	状态寄存器	命令寄存器

表2　Control/Diagnostic Registers控制/诊断寄存器

I/O地址	读	写
3F6H	交换状态寄存器(只读寄存器)	设备控制寄存器(复位)
3F7H	驱动器地址寄存器	

在硬盘执行读写过程中，为了节省I/O地址空间，用相同的地址来标识不同的寄存器。例如，如表1中端口地址1F7H，在向硬盘写入数据时作为命令寄存器，而向硬盘读取数据时作为状态寄存器。表1中各寄存器功能如下：

数据寄存器：是主机和硬盘控制器的缓冲区之间进行8位或16位数据交换用的寄存器，使用该寄存器进行数据传输的方式称程序输入输出方式，即PIO方式，数据交换的另一种方式是通过DMA通道，这种方式不使用数据寄存器进行数据交换；

错误寄存器：该寄存器包含了上次命令执行后硬盘的诊断信息。每位意义见表3，在启动系统、硬盘复位或执行硬盘的诊断程序后，也在该寄存器中保存着一个诊断码。

表3　IDE错误寄存器

位	意　义
0	AMNF，没找到所要访问的扇区的数据区。
1	TK0NF，在执行恢复RECALIBRATE命令时，0磁道没有发现。
2	ABRT，对硬盘发非法指令或因硬盘驱动器故障而造成命令执行的中断。
3	MAC，该信号用来向主机发出通知，表示介质的改变。
4	IDNF，没有找到访问的扇区，或CRC发生错误。
5	MC， 这是发送给主机一个信号以通知主机使用新的传输介质。
6	UNC，在读扇区命令时出现不能校正的ECC错误，因此此次数据传输无效。
7	BBK，在访问扇区的ID数据场发现坏的数据块时会置1。

下面的扇区数寄存器、磁道数寄存器、驱动器/磁头寄存器三者合称为介质地址寄存器，介质地址可以用CHS方式或LBA方式，在驱动器/磁头寄存器中指定用何种方式。

扇区计数寄存器：指明所要读/写的扇区总数，其中0表示传输256个扇区，如果在数据读写过程发生错误，寄存器将保存尚未读写的扇区数目。

磁道数寄存器：指明所要读/写的磁道数。

驱动器/磁头寄存器：指定硬盘驱动器号与磁头号和寻址方式，如表4所示。

表4 IDE驱动器/磁头寄存器

7	6	5	4	3	2	1	0
1	L	1	DRV	HS3	HS2	HS1	HS0

状态寄存器：保存硬盘控制器命令执行后的状态和结果，如表5所示。

表5 IDE状态寄存器

位	意 义
0	ERR，错误(ERROR)，该位为1表示在结束前次的命令执行时发生了无法恢复的错误。在错误寄存器中保存了更多的错误信息。
1	IDX，反映从驱动器读入的索引信号。
2	CORR，该位为1时，表示已按ECC算法校正硬盘的读数据。
3	DRQ，为1表示请求主机进行数据传输(读或写)。
4	DSC，为1表示磁头完成寻道操作，已停留在该道上。
5	DF，为1时，表示驱动器发生写故障。
6	DRDY，为1时表示驱动器准备好，可以接受命令。
7	BSY，为1时表示驱动器忙(BSY)，正在执行命令。在发送命令前先判断该位。

命令寄存器：包含执行的命令代码。当向命令寄存器写命令时，相关该命令的参数必须先写入。在写命令时，状态寄存器的BSY位置1。如果命令是非法，则中止执行。

在ATA标准中，IDE命令一共有30多个，其中有10个是通用型(也称强制型)命令。主要的参数如表6所示，表中的Word指2个字节。

表6 IDE硬盘的参数

Word 1	Word 3	Word 6	Word 10-19	Word 60-61
磁道数	磁头数	每磁道的扇区数	20个ASCII码系列号	LBA可以寻找的最大扇区数

20H 读扇区命令(带重试)：从硬盘读取指定磁道、磁头上的1~256个扇区到主机。送到主机的数据可以添加4个字节的ECC校验码，读的起始扇区号和扇区个数在命令块指定。这条命令也隐藏着寻找指定的磁道号，所以不需要另外的寻道命令。

30H 写扇区命令(带重试)：本命令是将主机内的数据写入硬盘，可以写指定磁道、磁头上的1~256个扇区，与读扇区命令相似，这条命令也隐藏着寻找指定的磁道号，写的起始扇区号和扇区个数由命令块指定。

90H 硬盘诊断命令：以判断硬盘是否已经连接到主机上，可以读取错误寄存器以检查需要的结果，如果是01H或81H表示设备是好的，否则表示设备没有连接或设备是坏的。

设备控制寄存器：将该寄存器的SRST位设置为1，可以使硬盘驱动器处于复位状态。IEN表示是否允许中断，其中0为允许。由此可见，对该寄存器发送0X0CH命令即令硬盘复位，其格式如表7所示。

表7 IDE设备控制寄存器

7	6	5	4	3	2	1	0
-	-	-	-	1	SRST	IEN	0

2 硬盘的寻址方式

硬盘驱动器的介质是通过磁头、磁道(柱面)、扇区组织起来的。在ATA标准中，磁道数可以达到65 636，一个扇区在正常情况下有512字节，可以用两种方法来寻址，即物理寻址方式和逻辑寻找方式。

IDE驱动器为逻辑寻址方式使用了线性映射的方式，即扇区从0柱面0头1扇区开始，在0磁头后是同柱面的1磁头，在整个柱面后是下一个柱面的0磁头，在ATA标准中，从物理结构CSH到逻辑块编号的影射如下：

LBA=(柱面号*磁头数+磁头号)*扇区数+扇区编号–1

逻辑扇区在访问时间上也是按顺序排列的。在UNIX、WINDOWS NT等操作系统中，硬盘的寻址方式是在内存中建了一个介质地址包，地址包里保存的是64位LBA地址，如果硬盘支持LBA寻址，就把低28位直接传递给ATA界面，如果不支持，操作系统就先把LBA地址转换为CHS地址，再传递给ATA界面。

对设计硬盘克隆程序，如果一个硬盘支持LBA寻找方式，在设计硬盘读写程序时可以不考虑硬盘的物理几何结构。但如果不支持LBA寻址方式，则需要用CHS寻址方式。在CHS寻址方式下，如果目标与源硬盘的磁头数一样，而仅磁道数不一样，并且源硬盘的磁道数<＝目标盘的磁道数时，克隆程序按扇区、磁头、磁道寻址顺序，在源盘读一个扇区，然后写到目标盘对应的扇区中。这样，克隆的目标盘数据与源盘一样，但目标盘可以使用的磁道数可能比实际的少一些，可以在克隆程序结束之前修改最后一个分区表的参数。

3 驱动器读写过程

用PIO方式使主机读写指定的起始磁道、头、扇区号，共读取*N*个扇区，其过程颇为复杂。过程包括发送指令、判断盘的状态、处理错误信息等。硬盘有自己的缓冲区，所以每次可以读取1个磁道上的所有扇区(1个磁道一般有63个扇区)保存在缓冲区，通过盘的数据寄存器(1F0H)与主机传输数据。

PIO方式读命令的执行过程如下：

1) 根据要读的扇区位置，向控制寄存器1F2H~1F6H发命令参数，等驱动器的状态寄存器1F7H的DRDY置位后进入下一步；

2) 主机向驱动器命令控制器1F7H发送读命令20H；

3) 驱动器设置状态寄存器1F7H中的BSY位，并把数据发送到硬盘缓冲区；

4) 驱动器读取一个扇区后，自动设置状态寄存器1F7H的DRQ数据请求位，并清除BSY位忙信号。DRQ位通知主机现在可以从缓冲区中读取512字节或更多(如果用READ LONG COMMAND命令)的数据，同时向主机发INTRQ中断请求信号；

5) 主机响应中断请求，开始读取状态寄存器1F7H，以判断读命令执行的情况，同时驱动器清除INTRQ中断请求信号；

6) 根据状态寄存器，如果读取的数据命令执行正常，进入7)，如果有错误，进入错误处理，如果是ECC错误，再读取一次，否则退出程序运行；

7) 主机通过数据寄存器1F0H读取硬盘缓冲区中的数据到主机缓冲区中，当一个扇区数据被读完，扇区计数器–1，如果扇区计数器不为0，进入3)，否则进入8)；

8) 当所有的请求扇区的数据被读取后，命令执行结束。

PIO方式写命令的执行过程如下：

1) 根据要写的扇区位置，向驱动器控制寄存器1F2H~1F6H发出命令参数，等驱动器DRDY置

位后进入下一步；

2) 主机向驱动器命令控制器1F7H发送写命令30H；

3) 驱动器在状态寄存器1F7H中设置DRQ数据请求信号；

4) 主机通过数据寄存器1F0H把指定内存(BUF)中的数据传输到缓冲区；

5) 当缓冲区满，或主机送完512个字节的数据后，驱动器设置状态寄存器1F7H中的BSY位，并清除DRQ数据请求信号；

6) 缓冲区中的数据开始被写入驱动器的指定的扇区中，一旦处理完一个扇区，驱动器马上清除BSY信号，同时设置INTRQ；

7) 主机读取驱动器的状态1F7H和错误寄存器1F1H，以判断写命令执行的情况，如果有无法克服的错误(如坏盘)，退出本程序，否则，进入下一步；

8) 如果还有扇区进行写操作，进入3)否则，进入下一步；

9) 当所有的请求扇区的数据被写后，命令执行结束。

虽然硬盘缓冲区可以容纳很多个扇区，但每读/写一个扇区，就判断其命令执行的状态寄存器，这样就可以保证读写的数据的正确性。

4 硬盘读写技术的应用

硬盘克隆软件的制作分两部分：1) 用C语言编写，控制要读写的起始磁道、磁头及扇区及扇区数，即对命令及参数块设置；2) 用汇编语言编写，以实现硬盘与主机之间的数据传送[2~4]。2个硬盘作为主(Host)盘和副(Slave)盘。Host盘容量应小于或等于Slave盘的容量，程序流程如图1所示。

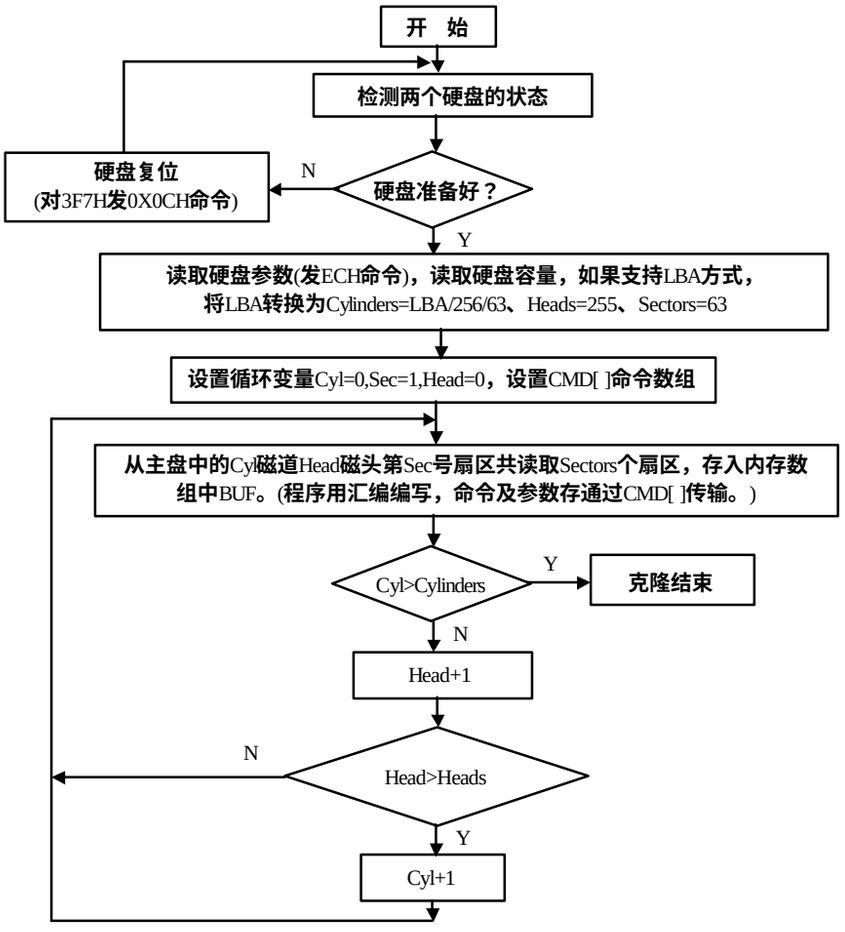

图1 C部分程序处理流程图

建立一个命令块数组如CMD[ ]，数组元素分别对应上述的0~7的任务寄存器及状态寄存器、错误寄存器、硬盘延时时间等。当CPU向硬盘控制器发布命令时，先将命令块等写在数组中，再将数组传到硬盘控制器所对应的寄存器中，控制器就会自动对命令进行分析和处理，在命令执行完成后，将状态控制器返回供主机判断命令执行的结果。主机向硬盘控制器发送的命令的程序处理流程(汇编程序段)如图2所示。

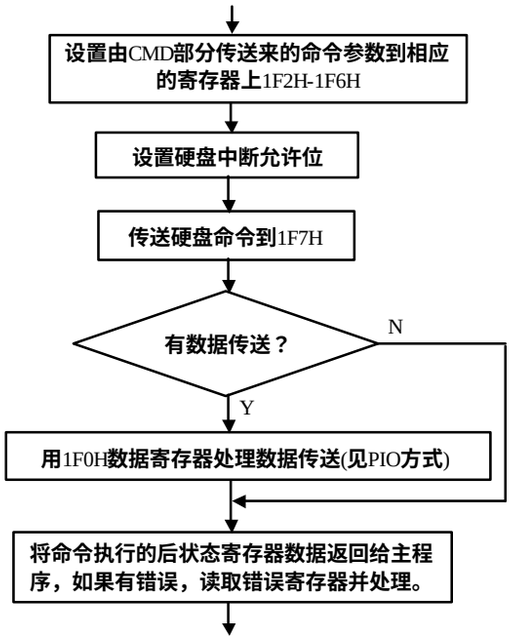

图2　汇编部分程序处理流程图

5　结 束 语

在了解硬盘的读写原理后，可以设计出任何针对硬盘物理扇区读写的程序。因为程序是直接读取驱动器的扇区数据，与硬盘上安装的具体操作系统的类型无关。如果用LBA寻址方式，还可以写出与硬盘驱动器的物理几何结构无关的读写程序。进一步，结合硬盘上的操作系统，可以编写检索硬盘上存储的任何特征信息的程序，还可以处理如多媒体信息这种特殊类的存储数据，故硬盘的读写技术有着很高的实用价值。

参 考 文 献

1　FRIEDHELM SCHMIDT. IDE接口: 协议、应用和编程. 北京: 中国电力出版社, 2001

2　刘　力, 陈建革. 最新实用IBM PC软、硬件技术参考大全. 北京: 北京市新闻出版局, 1990

3　艾德才. Pentium/80486实用汇编语言程序设计. 北京: 清华大学出版社, 2001

4　李向荣. 实用C语言软件开发工具. 北京: 清华大学出版社, 1996