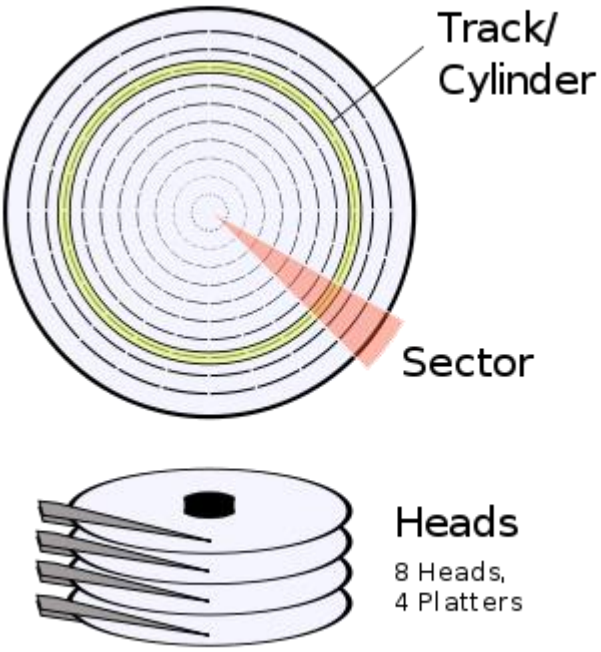
1、磁盘寻址

1.1、磁盘驱动器

磁盘物理结构图如下：

磁盘逻辑组成图如下：



1.2、什么是CHS（cylinder head sector）

通过上面材料，我们了解到磁盘通常由多个盘片、多个磁头组成。

每个盘片对应一个磁头（head），每个盘片被化成多个同心圆(track/cylinder)，每个同心圆被切断成多个段（sector）。磁盘存储最小单位是 sector，那么如何对 sector 进行定位？

CHS 是早期在 IBM PC 架构上面用来进行磁盘寻址的办法。

CHS 是一个三元组，组成如下：

1. 一共 24 个 bit 位。
2. 其中前 10 位表示 cylinder，中间 8 位表示 head，后面 6 位表示 sector。
3. 最大寻址空间

随着科技大发展，磁盘容量大幅提升。远远超过了 8GB 寻址范围，如何对 8GB 之外空间进行寻址？历史上曾经 CHS 从 24 位扩展到多 28 位，实现寻址 128GB，但是面对现在磁盘 2TB 容量还是无能为力，下面我们请出最终解决方案 LBA。

1.3、什么是 LBA（logical block addressing）

正如上文所说，LBA 是用来取代 CHS。那么 LBA 是怎么实现磁盘寻址？

1. LBA 是一个整数，通过转换成 CHS 格式完成磁盘具体寻址。
2. LBA 采用 48 个 bit 位寻址，最大寻址空间 128PB。

LBA 与 CHS 转换规则是怎么样的？

CHS->LBA

$$LBA = ((C \times HPC) + H) \times SPT + S - 1$$

LBA->CHS

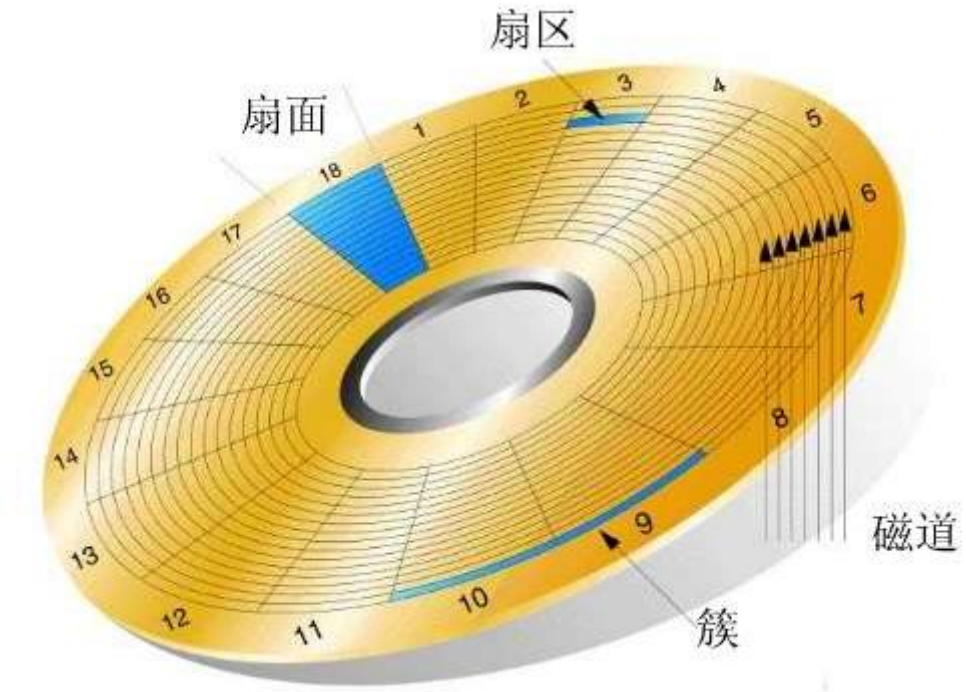
$$C = LBA \div (SPT \times HPC)$$
$$H = (LBA \div SPT) \bmod HPC$$
$$S = (LBA \bmod SPT) + 1$$

小结：

不管 CHS（寻址方式）也好，还是 LBA（寻址方式）也好。磁盘存储寻址都需要通过 cylinder、head、sector 这三个变量来实现；CHS、LBA 都是一个数字，CHS 按照固定格式把 24 个 bit 位分成 cylinder、head、sector；LBA 则需要通过求模运算得出 cylinder、head、sector。即由 chs 值可以直接获得 cylinder、head、sector 这三个变量的值，而由 LBA 值则需要通过运算间接得出 cylinder、head、sector 这三个变量的值。

补充一下：

a、CHS 方式寻址是在传统的扇区分布下进行的，即过盘片中心画直线来分扇区，具体如下图所示：



而 LBA 转换为 CHS 寻址则使用的 ZBR（Zone Bit Recording）方式分配扇区，即圆周上的扇区长短都一样。

b、磁盘在经过磁盘低级格式化程序格式化后，才出现有扇区、磁道这些对象（一个柱面是由各个盘面上的同一位置的磁道组成的，所以柱面不是磁盘低级格式化程序直接产生出来的对象，而是由磁道这个对象引申出来的一个概念）的。

（不同的）磁盘低级格式化程序采用不同的寻址方式算法时，盘面划分后的图像也不一样（如上图）。

一个磁盘，比如，原先使用的 CHS 寻址方式，后来可以修改为 LBA 寻址方式，当然这样盘面划分后的图像也变了，总之，一个磁盘可以修改它的寻址方式。

c、还有，由于 CHS 寻址方式的寻址空间在大概 8GB 以内，所以在磁盘容量小于大概 8GB 时，可以使用 CHS 寻址方式或是 LBA 寻址方式；在磁盘容量大于大概 8GB 时，则只能使用 LBA 寻址方式。

137GB、2.1TB 硬盘容量限制和 CHS、LBA 寻址及 GPT 分区表、UEFI/EFI

引言：各种寻址方式的所能寻址的寻址空间收到限制跟各个层次上寻址方式所能使用的寄存器个数（CPU 的寄存器个数对应能支持的操作系统的位数，BIOS 的各个层次的寄存器个数对应能支持的寻址空间大小）有关。

目前市场上的硬盘价格越来越低，相比以前同样价格买的硬盘容量越来越大。但是更换硬盘时，也要注意你的 PC 主板是否支持目前的大容量硬盘。计算机的发展一直受着硬盘容量限制或 BIOS（软件系统，主板是其硬件载体）访问容量限制的制约。

要明白这个原因我们需要对 IDE 接口硬盘的工作方式做一个介绍，即使到现在我们今天的 IDE 硬盘驱动依旧使用早先的 DOS-BIOS(磁盘操作系统-基本输入/输出系统)的分层结构上。它的基本工作模式就是：程序调用->DOS 功能调用->文件管理设备->INT 13 中断管理(读/写)->BIOS 磁盘服务->IDE(ATA)界面->磁盘控制器。也就是说我们如果需要对硬盘进行操作必须通过以上的一系列步骤才能完成，那么我们以下一起来看看这一系列步骤的作用。

文件管理设备

其负责文件及其在磁盘上存储位置之间的映射关系，不过需要通过磁盘读写中断 INT13 执行读写命令来存储、调入文件。当新文件被保存时，文件管理器决定它在当前目录里的存储位置，在文件分配表中为这个新文件添加文件目录项，并把文件写入磁盘。当读文件时，文件管理器在 FAT 中找到文件在磁盘上的位置，接着就调入文件。

IDE(ATA) 界面

在介绍 IDE(ATA)界面前，简单说说硬盘的结构：硬盘分为一定数量的柱面(以硬盘中心为圆心的同心圆磁迹)，每个柱面都需要磁头来读写数据。另外，硬盘上的数据都是以每扇区 512 字节的格式存储的，所有的数据传输都是以扇区(柱面被等分的园弧磁迹)为单位的。IDE(ATA)界面是寄存器驱动式的并口总线。要传输数据，BIOS 首先往 IDE(ATA)里特定的寄存器写入数据的开始地址和数据传输的长度，再把有关的读/写命令往特定的寄存器里发送从而开始数据传输。

现在的硬盘一般都支持逻辑块寻址(LBA)和柱面磁头扇区寻址(CHS)，我们以 CHS 寻址方式来举例：数据传输的开始地址是写到 4 个 8 位寄存器里的，分别是：

柱面低位寄存器

柱面高位寄存器

扇区寄存器

设备/磁头寄存器

柱面地址是 16 位[柱面低位寄存器(8 位)，柱面高位寄存器(8 位)]，扇区地址是 8 位(注意：扇区寄存器里第一个扇区是 1 扇区，而不是 0 扇区)，而磁头地址是 4 位(没有完全占用 8 位)。因此，硬盘柱面的最大数是 65,536(2 的 16 次方)，磁头的最大数是 16(2 的 4 次方)，扇区的最大数是 255(2 的 8 次方-1，注意刚刚我们提到的扇区寄存器问题)。

所以，能寻址的最大扇区数是 267,386,880 (65,536x16x255)。一扇区又是 512 字节，也就是说如果以 CHS 寻址方式，IDE 硬盘的最大容量为 136.9GB。LBA 寻址方式，上述的总共 28 位可用的寄存器空间(16+8+4)被看作一个完整的 LBA 地址，因为包括位 0(CHS 里扇区不能从 0 开始计算)，其能寻址的扇区数是 268,435,456 (65,536x16x256)，这时 IDE 硬盘的最大容量为 137.4GB。

INT 13 管理

INT 13 管理其实也是按照寄存器的模式来设计的，它的高层即文件管理器层发布数据读写命令和有关的参数给 CPU，然后触发 INT 13 中断的进行，激活 BIOS 的磁盘服务来执行数据传输。数据的开始地址被写到 3 个 8 位寄存器里，分别是：

柱面低位寄存器

柱面高位/扇区寄存器

磁头寄存器

柱面地址是 10 位(柱面低位寄存器占用 8 位、柱面高位寄存器占用 2 位)，扇区地址为 6 位(8 位-已经被计算过的高位寄存器的 2 位)。磁头寄存器为 8 位。因此如果这样的话：柱面的最大数是 1024(2 的 10 次方) ，磁头的最大数是 256(2 的 8 次方)，扇区的最大数是 63(2 的 6 次方-1)。所以，通过 INT 13 管理能寻址的扇区数是 16,515,072 (1,024x256x63)。一扇区是 512 字节，也就是说如果以 CHS 寻址方式，IDE 硬盘的最大容量为 8.456GB。LBA 寻址方式能寻址的扇区数是 16,777,216(1024x256x64)，这时 IDE 硬盘的最大容量为 8.601GB。

看到这里，我们应该感到硬盘容量限制的成因有了一些“眉目”了吧，那么我们具体来到底是什么让硬盘出现了所谓的限制：

528MB 硬盘容量限制

由于早先的硬盘容量比较小，因此设计的 BIOS 的时候当把地址从 Int 13 的地址寄存器转换为 IDE(ATA)的地址寄存器时，仅仅把 INT 13 管理中 10 位的柱面地址用来对应 IDE(ATA)界面中的 16 位柱面寄存器，而把没有用到的 6 位(高位寄存器)地址都设定为 0。并且也仅把 6 位的扇区地址来对应 IDE(ATA)界面的 8 位扇区寄存器，其中没有用到的 2 位设置为 0。并且 INT 13 管理的磁头寄存器 4 位(又去掉了 4 位)来对应 IDE(ATA)。因此，此时的磁盘柱面最大数为 1024(2 的 10 次方)，磁头的最大数是 16(2 的 4 次方)，扇区的最大数是 63(2 的 6 次方-1)。因此能寻址的扇区数就成了 1,032,192 (1,024x16x63)。一个扇区的容量是 512 字节，也就是说如果以 CHS 寻址方式，IDE 硬盘的最大容量为 528.4MB。因此 528MB 的硬盘容量限制就出现了。

2.1GB 硬盘容量限制

这里分为两个部分，一部分是由磁盘服务的限制造成的，另外一个是由于磁盘格式造成的，通常我们把前者称为 2.1GB 的硬件容量限制，后一种称为 2.1GB 的软件容量限制。

硬件容量硬件：为了 528MB 容量限制的问题，人们提出一些不同的办法，其中一个办法就是 INT 13 服务的磁头寄存器没有用到的 4 位中的 2 位(确切的说是高 2 位)保留给柱面数的第 11、12 位使用。这样，最大的磁头数就是 64(2 的 6 次方)。但是，当时的操作系统不使用这种转换方法，其认为磁头寄存器的所有位数只可能记录磁头数。比如，为了正确地转换柱面数为 2,048、磁头数为 64 的硬盘，就需要操作系统把柱面数除以 4(512 个逻辑柱面数)，磁头数乘以 4(256 个逻辑磁头数)。不过由于 BIOS 中并没有开放所有的磁头数寄存器，当然无法记录这样的磁头数。因此遇到这种运行机制的 BIOS，在系统自检的时候就会造成系统当机。

软件容量限制：当时 DOS 分区的限制是由文件分配表(FAT)决定的。FAT 处理存储空间是以簇为单位的，它处理一簇的最大长度是 32,768 字节，最多能处理 65,536 个簇，如果将两个数字相乘，就会得到 DOS 的最大分区界限值是 2,147,483,648 字节或 2,048MB (2,147,483,648 / 1,0242)。因此超过这个容量的硬盘，如果使用 FAT 格式，就最大只能识别 2.048GB 的硬盘容量。

3.2GB 的容量限制

一些版本的 BIOS 不能识别超过 6322 柱面的硬盘，不过这种 BIOS 比较少见，由于柱面有限制，其最高支持扇区数为 6,372,576 (6,322x16x63)，如果乘以 512 扇区容量的话，其最高支持容量为 6,372,576x512=3,262,758,912/1024=3.18GB。

4.2GB 的容量限制

当时一些操作系统使用 8 位寄存器来存储磁头数，这样当 BIOS 报告硬盘的磁头数等于 256(最高容量)时，只有磁头数的最先一位(即 0)被系统保存，从而导致硬盘配置错误。一旦硬盘的磁头数是 16，柱面数大于 8,192(2 的 13 次方，由于后三位寄存器已经被磁头寄存器借用，其实这里牵涉到一个突破 528MB 容量限制的转换做法的问题，由于这一段比较复杂，在这里就不详细介绍了，我们只要明白有这个限制就够了)，系统就无法正常识别了，因此其最大的容量就被限制在了 4.2GB=8,192x16x63x512/1024。

8.4GB 硬盘容量限制

我们已经知道 INT 13 服务的寻址方式最高可以支持 8.4GB 以下的容量(柱面数、磁头数、扇区数的最大值分别是 16,383、16 和 63，而三者相乘就是 8.456GB)。因此，这个容量限制出现是迟早的问题了。所以，这个限制是我们目前最常遇到的容量限制。

33.8GB 的容量限制

在 CHS 寻址中，由于 IDE(ATA)界面的限制，柱面数最高支持 65,535(2 的 16 次方-1)，所以，当遇到柱面数大于 65,535 的时候，系统就无法识别这种硬盘了，不过 LBA 由于独特的寻址模式就不存在这个问题，这个容量限制具体为：65,535x16x63x512/1024=33.8GB。

137GB 硬盘容量限制

目前的磁盘工作方式就注定 IDE 硬盘存在这个问题，前面介绍 IDE(ATA)界面的时候，这个问题就已经出现了。为了解决 INT 138.4G 限制，一些厂商定义了新的扩展 INT 13 服务扩展标准。新的 INT 13 服务扩展标准不使用操作系统的寄存器传递硬盘的寻址参数，它使用存储在操作系统内存里的地址包。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。通过这种方式，能实现在 ATA 总线基础上 CHS 寻址最大容量是 136.9 GB，而 LBA 寻址最大容量是 137.4GB。

因为 137GB 容量限制正是我们刚刚面临的问题，因此变得十分有名，我也是因为遇到了这个问题才学习了下硬盘容量限制这些知识。目前已经使用了新的方法解决了此问题，因此，我们的电脑上已经用上了 500GB、1TB 的硬盘了。但是，我们目前使用的电脑依然存在这种容量限制：

2.1TB 系统访问容量限制

目前的主流电脑大多无法运行 2.1TB 以上容量的硬盘。也就是说采用普通 32 位 Windows7、Windows Vista 等系统的电脑目前最大也只能识别到 2.1TB 的硬盘。那么到底是什么原因造成了这一问题呢？答案就是传统的 LBA(Ligical Block Address，逻辑块寻址)寻址方式以及现有系统的限制。

硬盘容量限制，最著名的当数 137GB 硬盘容量限制。当时不少人认为 100GB 的硬盘已经足够了，所以工程师们便推出了 28bit LBA 寻址模式，137GB 由此而来。为了消除 28bit LBA 寻址模式的限制、解除 137GB 容量的限制，Technical Committee T13 组织对 ATA/ATAPI-6 标准进行了一些修改，通过 48bit LBA 来支持更多的扇区。后来由 Compaq(康柏)、Microsoft(微软)、Maxtor(迈拓)联合推出的 Big Drives 规范就是以 T13 组织提出

的 48bit 方案为基础，这种规范的中心思想就是增加 CHS(C：柱面，H：磁头，S：扇区)的位数，而柱面寄存器不变，这样就将原来 LBA 寻址中可用的寄存器空间从 28bit 提高到了 48bit，可以寻址的扇区数就为 281,474,976,710,655，这样可支持的硬盘容量就达到了 281,474,976,710,655×512=144,115,188,075,855,872 字节，大致相当于 144PB(相当于 144000GB)的容量，即使从现在的眼光来看，如此大容量的硬盘也足够我们用上很久了。（当然，可能以后还会遇到 144PB 容量限制。）

既然 48bit LBA 支持 144PB 容量的硬盘，那现在的系统怎么识别不到 2.1TB 以上的硬盘呢？要真正实现 144PB，除了硬件存储系统及其接口芯片方面的改造外，还需要操作系统的支持。目前只有专业领域才支持 64bit 运算，而主流操作系统仍停留在 32bit 的编码模式，所以对硬盘的寻址自然最大也只有 32 位。现有 32 位系统的硬盘容量限制大小可以根据 32bit LBA 来计算，最后得出结论是最大只能支持 2.1 TB 的硬盘容量。这有点类似于当年“528MB、2.1GB”时所遇到的问题，因此，要让硬盘突破 2.1TB 的限制，用户必须先将系统升级到 64 位操作系统。目前的 Windows7 或者 Windows Vista 的 64 位系统和修改版的 Linux 系统都可以支持 48bit 这种长字节寻址模式。

* GPT 分区表

除了彻底解放 48bit LBA 寻址模式以外，我们还要考虑到另外一个关键因素，即分区表。目前的硬盘都采用 MBR 分区方式，而这种方式也是限制硬盘容量的关键因素。MBR 位于硬盘的第一个扇区，它所记录的便是硬盘的分区信息。如果使用 Fdisk、Partition Magic 或者直接在 Windows 安装过程中对硬盘进行分区，分区的信息都将被写入硬盘的 MBR 中，硬盘就根据 MBR 的内容来确定自己的逻辑分区情况。

MBR 最多可支持四个主分区或三个主分区、一个扩展分区的组合。这一方案采用 4 个字节来存储分区的总扇区数，最大能表示 4,294,967,296 (2 的 32 次方)个扇区，按每扇区 512 字节计算，分区最大容量不能超过 2.1TB(2,199,023,255,552)。而当磁盘容量超过 2.1TB 以后，分区的起始位置也就无法表示了，当然也就无法进行分区。在硬盘还未进入 TB 时代时，这种分区方式没有什么问题，但是当硬盘容量达到 2TB 同时还有向上的趋势时，MBR 分区方案显然就无法满足需要了。(编注：谈到 MBR 所支持的最大分区，也许有不少朋友说我目前的电脑分区就有七八个，这是咋回事？这里是说 MBR 只支持 4 个主分区表项。要获得更多分区，则需要次级结构——扩展分区。扩展分区可以再次让一个磁盘被分成一个或多个逻辑磁盘，打个比喻：MBR 就像是一本书的结构，目录、页码，而扩展分区则为这本书划分出了具体章节，这就是 MBR 分区方法。)

要解决 MBR 所存在的容量限制，还需要使用 GUID(Globally Unique Identifier，全局唯一标志符)分区模式，即采用 GPT(GUID 分区表)。与 MBR 最大 4 个主分区表项的限制相比，GPT 对分区数量没有限制(目前 Windows 系统最大仅支持 128 个 GPT 分区)，每个分区都拥有唯一的 ID 标志码，可管理的存储资源范围远远超过 MBR。而在分区容量方面，因为它使用 64 位的整数表示扇区号，所以理论上允许用户使用最高 18EB 容量(1EB=100 万 TB)进行分区，这绝对是一个高得令人匪夷所思的数字。GPT 的分区信息是在分区中，而不像 MBR 一样在主引导扇区，为保护 GPT 不受 MBR 类磁盘管理软件的危害，GPT 在主引导扇区建立了一个保护分区(Protective MBR)的 MBR 分区表，这种分区的类型标志为 0xEE，这个保护分区的大小在 Windows 下为 128MB，Mac OS X

下为 200MB，在 Windows 磁盘管理器里名为 GPT 保护分区，可让 MBR 类磁盘管理软件把 GPT 看成一个未知格式的分区，而不是错误地当成一个未分区的磁盘。

既然 GPT 分区方案具有如此多的优点，在分区时是不是可以全部采用这种方案呢?答案是否定的。并不是所有的Windows 系统都支持这种分区方案，至少 Windows XP/2000/NT/98 这些系统就无法支持 GPT 分区方案，所以在希捷的声明中，也说明了遇到 3TB 的硬盘，Windows XP 最多只能认出 990MB 的容量。

* UEFI/EFI 助硬盘容量腾飞

不过光是系统支持 GPT 还不行，那样用户只能在 GPT 分区上存储资料而不能在分区上启动电脑。如果要电脑支持 GPT 磁盘启动，还需要配合 UEFI/EFI 此类新一代 BIOS。正如我们之前所言，由于目前主流的主板都采用 BIOS 系统，也就是说只支持 MBR 的分区方式，那么即使我们拥有了 64 位的操作系统，拥有了 GPT 磁盘，也无法享受到 2.1TB 以上的大容量硬盘。要想使用 2.1TB 硬盘，我们还必须要保证自己的主板采用的是 EFI/UEFI，而非传统的 BIOS，这样才能支持 GPT 磁盘启动电脑。

Intel 在制定 EFI 方案时就将 GPT 列为其中的一部分内容，后来的 UEFI 也延续了这一点。由于现有的 BIOS 不可能将 MBR 分区方式立刻进化为 GPT 分区方式，所以要让主板识别到 2.1TB 以上容量的硬盘，那么只能使用 UEFI/EFI。因此，如果用户的主板支持 UEFI/EFI，那么可以从 GPT 分区直接启动 Windows Vista/7 系统，操作系统加载程序和启动分区均驻留在 GPT 磁盘上。与基于 x86 的计算机上的系统卷相同，EFI/UEFI 系统分区包含操作系统加载程序、驱动程序和启动 Windows Vista/7 需要的其他文件。在仅包含一个 GPT 磁盘的计算机上，UEFI/EFI 系统分区是磁盘上的第一个分区。如果用户的电脑没有 EFI/UEFI 支持，GPT 实际上并不能被真正的支持，仅仅能作为 MBR/GPT 混合磁盘来启动。也就是说，原本 GPT 的标准中，存在一个保护性质的 MBR(Protective MBR)，在不支持 EFI/UEFI 的芯片组主板下，BIOS 会继续使用该 MBR 来引导。

可以说，目前用户要突破硬盘所面临的 2.1TB 容量限制，首先要升级操作系统，其次还要有一定的硬件支持。目前的硬盘、BIOS 和操作系统制造商均将 2.1TB 的最大容量视为制造标准，主流配置的主板基本不会配备 UEFI 系统，尽管微星等少数主板制造商在一些主板中装载了 UEFI 系统，但它仍未成为真正的标准配置，个人电脑里也许只有苹果在使用 EFI/UEFI，因此大家要真正使用单个容量超过 2.1TB 的硬盘，还需要 UEFI 普及开来才行，这想必还需要一段时间。但我们相信，随着未来硬件规格的升级，容量在 2.1TB 以上的超级硬盘的巨大优越性(特别是速度)仍然会很快让不少玩家毫不犹豫地投怀送抱……