〔19〕中华人民共和国专利局

[11]公开号 CN 1049923A

〔12〕发明专利申请公开说明书

[21] 申请号 90106879.9

[51] Int.Cl[5]
G06F 12/14

〔43〕公开日 1991年3月13日

[22]申请日 90.8.11

[30]优先权

[32]89.8.25 [33]US [31]398,820

[71]申请人 国际商业机器公司

地址 美国纽约

[72]发明人 理查德·比尔柯斯基
小约翰·威利·布莱克莱奇
道伊尔·斯坦弗尔·克朗克
理查德·阿兰·大岩
斯科特·杰拉尔德·基尼尔
乔治·D·科瓦奇
小马修·斯蒂芬·帕尔卡
罗伯特·萨奇塞恩梅尔
凯文·马歇尔·兹沃罗斯基
杰利·杜亚恩·迪克逊
安德鲁·鲍伊斯·麦克内尔
爱德华·欧文·沃奇事尔

[74]专利代理机构 中国国际贸易促进委员会专利代理部

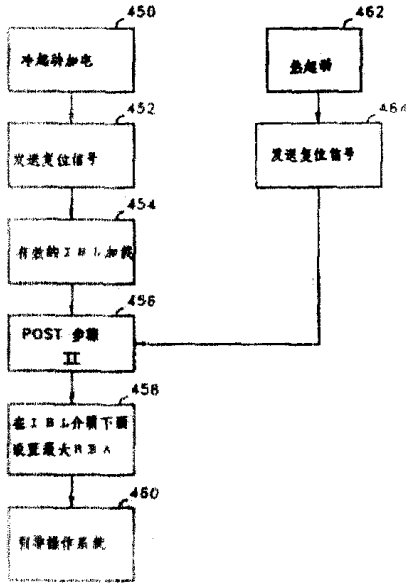
代理人 李勇

说明书页数: 26 附图页数: 13

[54]发明名称 防止未授权存取基本输入输出系统的设备和方法

[57]摘要

本发明提供了保护存储在个人计算机系统中直接存取存储器设备上的BIOS的设备和方法。个人计算机系统包括系统处理器，系统平板，随机存取主存储器，只读存储器，保护机构和至少一个直接存取存储器设备。只读存储器包括BIOS的第一部分和表示系统处理器类型与系统平板I/O配置的数据。BIOS的第一部分初始化系统和直接存取存储器设备，并且复位保护机构，以便从直接存取存储器设备上的可保护分区中将主引导记录读到随机存取存储器中。



权　利　要　求　书

1．一种在个人计算机系统中保护ＢＩＯＳ的设备，个人计算机系统具有执行操作系统的系统处理器，只读存储器，随机存取存储器及至少一个直接存取存储器设备，其特征在于：

直接存取存储器设备控制器，它具有一个保护机构，用于保护直接存取存储器设备的一个区域，所说的保护机构允许响应复位信号存取所保护的区域；

包括在直接存取存储器设备的保护区域中的主引导记录，所说的主引导记录包括了一个可执行代码段，该代码段具有从直接存取存储器设备中加载信息的机构；

包括在只读存储器中的ＢＩＯＳ的第一部份，所说的ＢＩＯＳ的第一部份初始化系统处理器，並且起动用于对直接存取存储器设备控制器复位的复位信号的产生，使系统处理器可以存取所说的主引导记录，以便将所说的主引导记录加载到随机存取存储器；

包括在直接存取存储器设备的保护区域中的ＢＩＯＳ的剩余部份，通过可执行代码段将所说的ＢＩＯＳ的剩余部份加载到随机存取存储器中，以便所说的ＢＩＯＳ的第一部份将控制传送给可执行代码段，可执行代码段再将控制传送给所说的ＢＩＯＳ的剩余部份以便引导操作系统，所说的ＢＩＯＳ的剩余部份激活所说的保护机构，以便在操作系统的正常操作期间防止存取直接存取存储器设备的保护区域。

2．权利要求1的设备的特点在于，直接存取存储器设备包括了

3．权利要求1的设备的特点在于，为了响应系统的加电，所说的ＢＩＯＳ的第一部份起动复位信号的生成。

4。权利要求1的设备的特点在于，为了响应用于系统中的复位状态，所说的ＢＩＯＳ的第一部份起动复位信号的生成。

5．权利要求1的设备的特点在于，主引导记录还包括一个数据段，数据段表示与所说的主引导记录兼容的个人计算机系统的硬件配置，並且只读存储器包括了表示系统处理器的硬件配置的数据，其中，在所说的ＢＩＯＳ剩余部份加载到随机存取存储器以前，所说的ＢＩＯＳ的第一部份要比较主引导记录中的硬件配置数据与只读存储器中的硬件配置数据，以便校验主引导记录与系统处理器的兼容性。

6．权利要求5的设备的特点在于，主引导记录的数据段包括了一个表示与主引导记录兼容的系统平板值，並且系统平板还包括用于唯一区分系统平板的机构，以便校验主引导记录与系统平板的兼容性。

7．权利要求5的设备的特点在于，主引导记录上的硬件配置数据包括型号值和子型号值。其中，型号值用于区分与所说的主引导记录兼容的系统处理器，而子型号值表示与主引导记录兼容的系统平板的Ｉ／Ｏ配置，並且所说的只读存储器包括相应的用于区分系统处理器的型号值和表示系统平板Ｉ／Ｏ配置的子型号值，其中，主引导记录的所说的型号值和子型号值要分别与只读存储器中的相应的型号值和子型号值加以比较，以便校验主引导记录与系统处理器和系统平板的Ｉ／Ｏ配置的兼容器。

8．权利要求1的设备的特点在于，个人计算机系统还包括一个用电连到系统处理器的非易失性随机存取存储器，所说的非易失性随

机存取存储器包括表示系统配置的数据，当系统配置修改时，要更新所说的数据，其中，所说的ＢＩＯＳ的第一部份要将在非易失性随机存取存储器中的所说的数据与只读存储器中的相应数据加以比较，以便决定系统的配置是否已经修改。

９．权利要求２的设备的特点在于，所说的系统处理器将数据记录以块的格式传送给所说的磁盘控制器，该格式是按顺序给块编号的格式，並且所说的主引导记录和所说的ＢＩＯＳ的剩余部份是有效地存储在编号为较大位数的块中。

１０．权利要求９的设备的特点在于，所说的保护机构包括设置最大的可寻址块，所说的最大的可寻址块是主引导记录和ＢＩＯＳ的剩余部份的编号为最小位数的块，所说的保护机构禁止存取大于或等于最大的可寻址块的那些块，而允许存取小于最大的可寻址块的那些块。

１１．保护个人计算机系统的系统驻留程序的设备，个人计算机系统具有系统处理器，只读存储器，主存储器和至少一个能存储许多数据记录的直接存取存储器设备，其特点在于：

包括在只读存储器中的第一个程序，所说的第一个程序初始化系统处理器，所说的第一个程序还起动送到直接存取存储器设备的复位信号的生成，以便允许存取数据记录；

从直接存取存储器设备将数据记录加载到主存储器的加载机构，所说的加载机构是存储在直接存取存储器的可保护的分区中，通过所说的第一个程序将所说的加载机构从直接存取存储器设备中读到主储器中，其中，所说的第一个程序激活了所说的加载机构；

存储在直接存取存储器设备的可保护分区中的主存储器驻留程序

映象，通过所说的加载机构从直接存取存储器设备中将所说的主存储器驻留程序映象读到主存储器中以便产生主存储器驻留程序；

用于保护直接存取存储器设备的可保护分区的机构，所说的保护机构是由所说的主存储器驻留程序激活的，以防止未授权地存取所说的加载机构和所说的主存储器驻留程序映象。

12．权利要求11的设备的特点在于：所说的加载机构还包括一个校验机构，用于确认个人计算机系统与主存储器驻留程序的兼容性。

13．权利要求12的设备的特点在于，所说的加载机构包括一个个具有可执行代码段的主引导记录，用于实现主存储器驻留程序的加载，其中，所说的第一个程序将控制传送给所说的可执行代码段，以实现将所说的主存储器驻留程序映象加载到主存储器中。

14．权利要求11的设备的特点在于，所说的第一个程序包括了一个加电自检（POST）例行程序，所说的加电自检例行程序只是初始化和测试那些对加载主存储器驻留程序所必须的系统功能。

15．权利要求14的设备的特点在于，所说的加电自检初始化系统处理器功能，存储器子系统和直接存取存储器设备子系统。

16．权利要求12的设备的特点在于，所说的校验机构包括了表示系统处理器类型和连到系统处理器的系统平板配置的数据。

17．权利要求11的设备的特点在于，至少有一个直接存取存储器设备包括一个硬盘驱动器，其中，所说的加载机构从所说的硬盘驱动器将数据记录加载到主存储器中。

18．权利要求17的设备的特点在于，所说的硬盘驱动器包括一个磁盘控制器并且所说的系统处理器将数据记录以将块顺序编号的

块的格式传送到所说的磁盘控制器中，而且所说的主引导记录和所说的ＢＩＯＳ的剩余部份是有效地存储在编号为较大位数的块中。

１９．权利要求１８的设备的特点在于，所说的保护机构包括设置一个最大的可寻址块，所说的最大的可寻址块是主引导记录和ＢＩＯＳ的剩余部份中的编号为最小位数的块，所说的保护机构防止存取大于或等于最大的可寻址块的那些块，而允许存取小于最大可寻址块的那些块。

２０．权利要求１１的设备的特点在于，为了响应用于系统的加电顺序，所说的第一个程序起动复位信号的生成。

２１．权利要求１１的设备的特点在于，为了响应用于系统的复位状态，所说的第一个程序起动复位信号的生成。

２２．用于防止未授权的存取ＢＩＯＳ的设备，ＢＩＯＳ存储在具有系统处理器的个人计算机系统中的海量存储器设备中，海量存储器设备能存储许多定义在第一和第二数据块末端之间的数据块，系统处理器可以以单个可定义的连续数据块的格式存取ＢＩＯＳ，ＢＩＯＳ可从第三个数据块的末端扩展到第四个数据块的末端，第一和第二数据块末端与第三和第四数据块的末端相连接，所说的设备包括：

（ａ）连接在所说的系统处理器和所说的海量存储器设备之间的控制器设备，用于将输入或输出请求从系统处理器变换为海量存储器设备的实际特性，输入／输出请求是单个可定义的连读数据块的格式；

（ｂ）第一个逻辑机构，用于起动复位信号的生成；

（ｃ）第二个逻辑机构，用于生成防止存取ＢＩＯＳ代码的第二信号；

（ｄ）保护机构，响应所说的复位信号允许存取所说的ＢＩＯＳ代码，所说的保护机构要响应用于设置第三个数据块边界的所说的第二信号，以便在由系统处理器授权的程序正常执行期间防止存取ＢＩＯＳ代码。

２３．权利要求２２的设备，其中海量存储器设备包括一个以柱面、磁头和扇区格式来实现输入／输出请求的硬盘，而且其中，所说的控制器将数据块的格式转换为柱面、磁头和扇区的格式。

２４．权利要求２２的设备，其中所说的控制器设备包括响应所说的系统处理器的ＳＣＳＩ适配器卡。

２５．权利要求２２的设备，其中第一个逻辑机构起动复位信号的生成，以响应系统处理器的加电状态。

２６．权利要求２２的设备，其中第一个逻辑机构起动复位信号的生成，以响应连到系统的键盘的输入。

２７．保护个人计算机系统中的ＢＩＯＳ的方法，该系统包括系统处理器，只读存储器，随机存取存储器和直接存取存储器设备，所说的方法包括下述步骤：

（ａ）在只读存储器中存储ＢＩＯＳ的第一部份，ＢＩＯＳ的第一部份包括用于初始化系统的机构，

（ｂ）在直接存取存储器设备上的可保护分区存储主引导记录和ＢＩＯＳ的剩余部份，在个人计算机系统的正常操作期间ＢＩＯＳ的剩余部份驻留在随机存取存储器中；

（ｃ）初始化系统和起动送到直接存取存储器设备的复位信号的生成；

（ｄ）取消对可保护分区的保护，以允许系统处理器存取主引导

记录和ＢＩＯＳ的剩余部份，响应复位响号取消保护，

（ｅ）将主引导记录加载到随机存取存储器中，主引导记录包括一个可执行代码段；

（ｆ）将控制传送给可执行代码段，以便将ＢＩＯＳ的剩余部份加载到随机存取存储器中；并且

（ｇ）将控制传送给随机存取存储器中的ＢＩＯＳ剩余部份，ＢＩＯＳ的剩余部份在可保护分区上设置保护，以防止未授权地存取存储在直接存取存储器设备上的可保护分区中的主引导记录和BIOS的剩余部份。

２８．权利要求２７的方法，还包括通过将存储在ＢＩＯＳ的第一部份中的数据与存储在主引导记录中的相应数据相比较的方法，来校验主引导记录与系统兼容性的步骤（ｈ）。

２９．权利要求２７的方法，还包括通过将存储在只读存储器中的数据与包括在主引导记录中的相应数据相比较的方法，来校验主引导记录与系统处理器的兼容性的步骤（ｉ）。

防止未授权存取基本输入输出系统的设备和方法

本发明与个人计算机系统有关，它特别阐述了保护存储在个人计算机系统中的海量存储器设备中的ＢＩＯＳ的方法和设备。

通常而言的个人计算机系统，特别是ＩＢＭ个人计算机在提供计算能力方面已经在今天现代社会的许多部门中获得了广泛的应用。通常可以将个人计算机系统定义为台式的、落地式的或便携式的微计算机。它由具有单个系统处理器的系统部件、显示监视器、键盘、一个或几个软盘驱动器、硬盘存储器及一个可任选的打印机所组成。这些系统的明显的特点之一是使用将这些部件用电连接在一起的母板式系统平板。这些系统的主要的设计目的是给单用户提供独立的计算能力，而且对于个人或者对于小企业而言，购买这些系统的价格是不贵的。这样的个人计算机系统实例有ＩＢＭ的个人计算机ＡＴ（ＰＣＡＴ）和ＩＢＭ的个人系统／２（ＰＳ／２）的２５、３０、５０、６０、７０和８０型。

这些系统可以分为两大系列。第一个系列，通常叫做系列Ｉ型，使用的是总线结构，ＩＢＭ　ＰＣ　ＡＴ和其它的“ＩＢＭ兼容”机就是这种例证。而第二个系列，叫做系列Ⅱ型，使用的是ＩＢＭ的微通道总线结构，ＩＢＭ的ＰＳ／２的５０型至８０型则是这种例证。

从系列Ｉ型的最早的个人计算机系统，如ＩＢＭ　ＰＣ开始，人

Ⅱ型的ＢＩＯＳ中。于是系列Ⅰ型的ＢＩＯＳ称做为兼容的ＢＩＯＳ或ＣＢＩＯＳ。然而，正如前面关于ＩＢＭ　ＰＣ　ＡＴ解释那样，只有３２Ｋ字节的ＲＯＭ驻留在系统平板上。幸运的是系统可以使ＲＯＭ扩展到９６Ｋ字节。可惜由于系统的局限性，这９６Ｋ字节的ＲＯＭ便是ＢＩＯＳ可以得到的最大容量。幸运的是即使外加了ＡＢＩＯＳ，ＡＢＩＯＳ和ＣＢＩＯＳ仍可以挤在９６Ｋ字节的ROM中。然而，在９６Ｋ　ＲＯＭ区域中只有很小的百分比可以保留为扩充之用。如果将来Ｉ／Ｏ设备不断增加的话，ＣＢＩＯＳ和ABIOS最后将用完ＲＯＭ的空间。于是，新的Ｉ／Ｏ技术将不能容易地集成在ＣＢＩＯＳ和ＡＢＩＯＳ之中。

由于这些问题，再加上希望在开发周期中对尽可能新的系列Ⅱ的ＢＩＯＳ作修改，将ＢＩＯＳ中的部份代码从ＲＯＭ中卸载下来是很必要的。这是通过将ＢＩＯＳ的部份代码存储在海量存储器设备如硬盘上来实现的。由于磁盘除了可读以外，还提供了写的能力，因此在磁盘上修改实际的ＢＩＯＳ代码是可行的。当磁盘提供快速、有效的方法来存储ＢＩＯＳ代码时，磁盘仍然会大大地增加使ＢＩＯＳ代码变得不纯洁的可能性。由于ＢＩＯＳ是操作系统不可缺少的部份，一个不纯洁的ＢＩＯＳ可能导致毁坏的结果並且在许多情况下，造成系统的故障和不能操作。因此，很明显，提供一种防止未授权的修改硬盘上的ＢＩＯＳ代码的机构是非常需要的。

本发明是为了解决上述问题的。因此，本发明的目的之一是提供一种机构来防止未授权就修改存储在个人计算机系统中的直接存取存储器设备上的ＢＩＯＳ代码。

本发明的另一目的是对加载ＢＩＯＳ的磁盘提供保护。由于

ＢＩＯＳ在实现上是化费不多的而且对最终用户又是相当透明的，因此，它並不減損计算机系统的商业上的可接受性。

大体上可认为，按照本发明的个人计算机系统是由系统处理器，随机存取存储器，只读存储器以及至少一个直接存取存储器设备组成的。在系统处理器和直接存取存储器设备之间连接的直接存取存储器设备控制器包括一个保护存储器设备区域的机构。该存储器设备的保护区域包括主引导记录和ＢＩＯＳ映象。为了响应复位信号，该保护机构允许存取保护区域，以使主引导记录加载到随机存取存储器。在操作中，该主引导记录还将ＢＩＯＳ映象加载到随机存取存储器。当执行现在在随机存取存储器中的ＢＩＯＳ时，生成一个第二信号，它激活了保护机构以便禁止存取含有主引导记录和ＢＩＯＳ映象的磁盘上的区域。然后ＢＩＯＳ引导操作系统，以便开始系统的操作。

具体讲，只读存储器包括ＢＩＯＳ的第一部份。ＢＩＯＳ的第一部份对系统处理器和直接存取存储器设备进行初始化，並复位保护机构，以便从直接存取存储器设备上的保护区域或分区中将主引导记录读到随机存取存储器中。该主引导记录包括一个数据段和一个可执行代码段。数据段包括表示系统硬件的数据和表示由主引导记录支持的系统配置数据。ＢＩＯＳ的第一部份通过校验主引导记录的数据段数据与表示系统处理器，系统平板和平板Ｉ／Ｏ配置的ＢＩＯＳ第一部份中的数据的一致性，来确认主引导记录和系统硬件的兼容性。

如果主引导记录与系统硬件兼容的话，ＢＩＯＳ的第一部份就引导系统处理器来执行主引导记录的可执行代码段。可执行代码段确认系统配置还没有修改的话，就从直接存取存储器设备中将ＢＩＯＳ的剩余部份加载到随机存取存储器中。然后可执行代码段校验ＢＩＯＳ

们已经认识到软件兼容性的极端重要性。为了达到该目标，在硬件和软件之间建立了一个系统驻留代码（也叫做“微码”）的隔离层。该代码在用户的应用程序／操作系统与设备之间提供了一个操作接口，这样减轻了用户对硬件设备特性的关注。为了使新的设备能加到系统中，同时又将应用程序和硬件的特性隔离开来，在基本的输入／输出系统（ＢＩＯＳ）中开发了该代码。ＢＩＯＳ的重要性是十分显然的，因为它将设备驱动程序从依赖于专门设备硬件特性中解放出来，同时又把对设备的中间接口提供给设备驱动程序。因为ＢＩＯＳ是系统的不可缺少的部份，它控制了系统处理器数据输入与输出的传送，所以它是驻留在系统平板上的，並且是以只读存储器（ＲＯＭ）方式送给用户的。例如，在最初的ＩＢＭ　ＰＣ中，占据８Ｋ　ＲＯＭ的ＢＩＯＳ是驻留在平板上的。

当个人计算机系列新的型号提出时，ＢＩＯＳ必须更新及扩充，使其能包括对新的硬件和Ｉ／Ｏ设备的支持。正如可以预料的那样，ＢＩＯＳ开始增加其存储器的大小。例如，当引入ＩＢＭ　ＰＣ　ＡＴ时，ＢＩＯＳ增长到需要３２Ｋ字节的ＲＯＭ。

今天，当用新技术开发时，系列Ⅱ型个人计算机系统渐渐变得更复杂了，而且对消费者而言，则可以更经常获得这些个人计算机系统。由于技术在急剧地变化，且新的输入／输出（Ｉ／Ｏ）设备在不断增加到个人计算机系统中，因此在个人计算机系统的开发周期中，对ＢＩＯＳ的修改已经成为一个十分重要的问题。

例如，当引入具有微通道结构的ＩＢＭ　ＰＳ／２时，已经开发了一个重要的、新的ＢＩＯＳ，通称为先进的ＢＩＯＳ，或ABIOS。然而，为了维护软件的兼容性，系列Ⅰ型的ＢＩＯＳ必须包括在系列

剩余部份的可靠性，再引导系统处理器开始执行现在在随机存取存储器中的ＢＩＯＳ。在随机存取存储器中执行的ＢＩＯＳ，生成一个用于保护具有ＢＩＯＳ剩余部份的磁盘分区的第二信号，並再引导操作系统以便开始个人计算机系统的操作。为了保护ＢＩＯＳ代码的完整性，则要保护拥有ＢＩＯＳ剩余部份的磁盘分区，以便防止存取磁盘上的ＢＩＯＳ代码。

在下述与附图有关的描述中解释了本发明的前景方面和其它的特点。其中：

图１为个人计算机系统的剖面视图，说明了连到许多直接存取存储器设备的系统平板；

图２为图１所示的个人计算机系统的系统框图；

图３为装在系统平板上的ＲＯＭ　ＢＩＯＳ的存储器映象变换图；

图４为描述从直接存取存储器设备中加载ＢＩＯＳ映象的总的过程的流程图；

图５为主引导记录的记录格式；

图６Ａ为描述ＩＢＬ程序操作的流程图；

图６Ｂ为说明从硬盘中加载ＢＩＯＳ映象的步骤的流程图；

图６Ｃ为说明从软盘中加载ＢＩＯＳ映象的步骤的流程图；

图６Ｄ为较详细说明检查主引导记录与平板／处理器之间的兼容性的流程图；

图７为说明主引导记录的可执行代码段操作的详细流程图；

图８为直接存取存储器设备的控制器的框图；

图９为说明为了保护存储在磁盘驱动器上ＩＢＬ介质的磁盘控制

器操作的流程图；並且

图１０为说明保护ＢＩＯＳ映象的方法的流程图。

为了实现本发明，下述详细描述是目前预期的最好方式。本描述並不打算在限定的意义上加以描述，但本描述只是为了说明本发明的总的原则，因为由附录的专利范围已最好地定义了本发明的范围。

现在参考附图，特别是图１。图１是个人计算机系统１０的剖面图。它有许多ＤＡＳＤ（直接存取存储器设备）１２－１６，这些ＤＡＳＤ通过许多Ｉ／Ｏ槽１８连到系统板或平板２４上。电源２２以众所周知方式向系统１０提供电能。平板２４包括一个系统处理器，系统处理器在计算机指令的控制下操作输入、过程和输出信息。

在使用时，设计个人计算机系统的主要目的是为一小组用户或单个用户提供独立的计算能力，而且对于个人或小企业而言，其购买的价格也是不贵的。在操作时，系统处理器在操作系统，如ＩＢＭ ＯＳ／２操作系统或ＰＣ－ＤＯＳ的控制下运行。这种类型的操作系统包括了ＤＡＳＤ１２－１６和该操作系统之间的ＢＩＯＳ的接口。可以按功能将ＢＩＯＳ分成模块，而ＢＩＯＳ的一部份存储在平板２４的ＲＯＭ上，下文将其称作ＲＯＭ－ＢＩＯＳ。ＢＩＯＳ在硬件和操作系统软件之间提供了一个接口，使程序员或用户可以对它们的机器编程，而不需具有对特别的设备的深入的操作知识。例如，ＢＩＯＳ的软盘模块可允许程序员对软盘驱动器编程而不需深入了解软盘驱动器的硬件。因此，不同的公司设计和制造的若干种软盘驱动器均可以用在系统中。这不仅降低了系统１０的成本，而且使用户可从若干种软盘驱动器中加以选择。

在将上述结构和本发明联系起来之前，对个人计算机系统１０一

般的操作小结是值得回顾的。考虑图2，图2为个人计算机系统10的框图，它说明了平板24的部件、平板24和I／O槽18的连接及个人计算机系统的其它硬件。由微处理器组成的系统处理器26位于平板24上，局部总线28将微处理器连到存储器控制器30，而该控制器再连到随机存取存储器（RAM）32。虽然任意合适的微处理器均可以使用，但一种恰当的微处理器便是由英特尔公司销售的80386。

尽管在下文描述的本发明是特别参照图2的系统框图加以描述的，但在下面描述开始时就可以被理解为，凡按照本发明所用的设备和方法，均可用在平板（母板）的其它硬件配置上。如系统处理器可以为Intel 80286或80486微处理器。

处理器可存取平板标识号（平板ID）平板ID对于某特定的平板是唯一的，它可区分所用平板的类型。例如平板ID可以通过硬线连接，由系统处理器通过I／O端口将其读出或通过使用开关将其读出。另外，系统处理器26的另一I／O端口可用来生成使用平板逻辑线路连到磁盘控制器的复位信号。例如，寻址I／O端口的软件和激活平板逻辑来生成复位信号的软件可以用来初始该复位信号。

局部总线28通过总线控制器34再连到平板24上的只读存储器（ROM）。

附加的非易失性存储器（NVRAM）58通过再连到总线控制器34的串行／并行端口的接口40连到微处理器26。该非易失性存储器可以为有电池备份的CMOS电路，使无论何时系统中电源掉电时，仍能保持其上的信息。由于ROM通常是驻留在平板上的，而存储在ROM中的型号和子型号值相应地是用来区分系统处理器和系

统平板I／O配置情况。因而，这些值实际上将用来区分处理器和平板I／O的配置。NVRAM用来存储系统配置的数据。即NVRAM将保持描述系统现在配置的值。例如，NVRAM中具有描述硬盘或软盘的容量，显示的类型，存储器的容量，时间和日期等的信息。此外，每当执行专门的配置程序如SET Configuration（设置配置）时，存储在ROM中的型号和子型号的值总会复制到NVRAM中。设置配置程序的目的就是将系统配置特点的值存储在NVRAM中。因此，对于配置正确的系统而言，其NVRAM中的型号和子型号的值分别等于存储在ROM中的型号与子型号的值。如果这些值不相等的话，它表明系统的配置已被修改过。请参阅图6D，在那儿结合加载BIOS情况，将较详细地予以解释该特点。

参见图2，I／O平板总线43将总线控制器34再连到I／O插槽18，串行／并行接口40和外围控制器42。外围控制器42再连到键盘44，鼠标器46，诊断板47和软盘控制器64。除了NVRAM58以外，串行／并行接口再连到串行端口48和并行端口50，以便将信息输入／输出到打印机，硬拷贝设备等。正如在该技术上众所周知的那样，局部总线也可连到高速缓冲存储器控制器52和高速缓冲存储器的存储器68，协处理器54和DMA控制器56。

系统处理器26除了与个人计算机系统10的其它部件接口外，还控制它的内部操作。例如，可以看到系统处理器26连到一个小计算机系统接口（SCSI）I／O卡60上，该卡又再连到一个DASD上，如硬盘驱动器62。当然，按照本发明，SCSI磁盘驱动器不能用作硬盘。除硬盘62以外，系统处理器26可以和控制

软盘驱动器66的软盘控制器64相接口。关于术语方面，术语“硬文件”描述的是硬盘驱动器62，而术语“软盘”描述的则是软盘驱动器66。

在本发明以前，ROM36可能包括操作系统和硬件外围设备接口的全部BIOS的代码。然而，按照本发明的一个方面而言，ROM36只适合存储BIOS的一部份。当系统处理器26执行这一部份时，就会从硬盘62或软盘66中输入BIOS的第二部份或BIOS的余下部份。以下将这部份BIOS称作BIOS的映象。该BIOS映象替代了BIOS的第一部份，并且作为系统中不可缺少的部份，它是驻留在主储器中如RAM32。作为存储在ROM36中的BIOS的第一部份（ROM－BIOS）在图3－4上将作一般的解释，而在图6A－D上将详细地解释。图5将解释BIOS的第二部份（BIOS映象），而图7将解释BIOS映象的加载。从DASD加载BIOS映象的另一好处是能够将BIOS直接加载到系统处理器的RAM32中。由于存取RAM比存取ROM要快得多，所以可以使计算机系统处理速度得到重大的改进。

现在我们着手解释ROM36中的BIOS的操作和从硬盘或软盘中加载BIOS映象的操作。一般说，第一部份程序如ROM--BIOS先检查系统，并且将BIOS的主引导记录加载到RAM中。主引导记录包括具有校验信息的数据段，而且作为加载机构，它还具有可执行代码的代码段。可执行代码使用数据信息来校验硬件的兼容性和系统的配置。当对硬件兼容性和正确的系统配置测试之后，可执行代码就将BIOS映象加载到RAM中，产生一个主存储器驻留程序。BIOS映象继ROM BIOS之后，加载操作系统并开

始机器的操作。为了使叙述更明了，我们将主引导记录的可执行代码段叫做ＭＢＲ代码，而主引导记录的数据段叫做ＭＢＲ数据。

图３为包括ＲＯＭ－ＢＩＯＳ的不同代码模块的存储器映象图。ＲＯＭ－ＢＩＯＳ包括加电自检（ＰＯＳＴ）步骤Ｉ模块７０，初始ＢＩＯＳ加载（ＩＢＬ）程序模块７２，软盘模块７４，硬文件模块７６，视频模块７８，诊断板模块８０和硬件兼容性数据８２。简而言之，ＰＯＳＴ步骤Ｉ７０执行系统的预初始化和测试。ＩＢＬ程序７２决定ＢＩＯＳ映象究竟从硬盘加载还是从软盘加载，检查兼容性及加载主引导记录。软盘模块７４提供了软盘驱动器的输入／输出功能。硬文件模块７６控制对硬盘或类似东西的输入／输出。视频模块７８控制再连到视频显示的视频输入／输出控制器的输出功能。诊断板模块８０提供了系统诊断显示设备的控制功能。硬件兼容性数据８２包括了系统型号的值和子型号的值。这些值将在以后关于图５的描述中加以说明。

现在再参考图４，它给出了从硬盘或软盘将ＢＩＯＳ映象加载到系统中的处理概况。当系统加电时，将系统处理器引导到ＰＯＳＴ步骤Ｉ的输入点，见图４的步骤１００。ＰＯＳＴ步骤Ｉ初始化系统并且测试从所选的ＤＡＳＤ中加载ＢＩＯＳ映象的所需要的那些系统功能，见步骤１０２。特别是，如果需要的话，ＰＯＳＴ步骤Ｉ会初始化处理器／平板的功能，诊断板，存储器子系统，中断控制器，定时器，ＤＭＡ子系统，硬盘ＢＩＯＳ程序（硬文件模块７６）和软盘ＢＩＯＳ程序（软盘模块７４）。

在ＰＯＳＴ步骤Ｉ预初始化系统之后，ＰＯＳＴ步骤Ｉ引导系统处理器执行包括在初始ＢＩＯＳ加载模块７２中的初始ＢＩＯＳ加载

（ＩＢＬ）程序。ＩＢＬ程序首先决定ＢＩＯＳ映象是否存储在硬盘上或者是否可从软盘加载；其次从所选的介质（硬盘或软盘）将主引导记录加载到ＲＡＭ中，见步骤１０４。ＭＢＲ数据是用于校验目的而执行ＭＢＲ代码是用来加载ＢＩＯＳ映象的。ＩＢＬ程序的操作的详述将在关于图６Ａ－Ｄ的描述中给出。

继续参考图４，在ＩＢＬ程序将主引导记录加载到ＲＡＭ之后，系统处理器引导到ＭＢＲ代码的起始地址，以便开始执行，见步骤１０６。ＭＢＲ执行一系列的校验测试以决定ＢＩＯＳ映象的可靠性並校验系统的配置。为了更好地了解ＭＢＲ代码的操作，请注意图７，它将更详尽地描述ＭＢＲ代码。

在这些校验测试的基础上，ＭＢＲ代码将ＢＩＯＳ映象加载到ＲＡＭ中，並将控制传送给新加载到主存储器的ＢＩＯＳ映象中，见步骤１０８。ＢＩＯＳ映象被加载到由ＲＯＭ－ＢＩＯＳ以前占据的地址空间。即，如ＲＯＭ－ＢＩＯＳ的地址为ＥＯＯＯＯＨ至ＦＦＦＦＦＨ，则ＢＩＯＳ映象也加载到该ＲＡＭ地址空间，因此，ＢＩＯＳ映象替代了ＲＯＭ－ＢＩＯＳ。然后控制传送给ＰＯＳＴ步骤Ⅱ，它包括在新加载的ＢＩＯＳ映象中，于是废弃了ＲＯＭ－ＢＩＯＳ的控制。为了加载操作系统的引导程序，现在在ＲＡＭ中的ＰＯＳＴ步骤Ⅱ则初始化並测试余下的系统，见步骤１１０－１１４。在ＰＯＳＴ步骤Ⅱ将控制传送给操作系统之前，为了防止存取由ＢＩＯＳ映象占据的磁盘分区，ＰＯＳＴ步骤Ⅱ设置了一个保护机构。为了详细讨论该保护过程，请参阅图８－１０。值得注意的是在热起动期间，处理器引导到第１０８步，並旁路了第１００—１０６步。

为了明确起见，现在来描述表示主引导记录的格式。参考图5，它给出了主引导记录。该引导记录包括可执行代码段120和数据段122－128。MBR代码120包括与DASD有关的代码，负责校验ROM－BIOS的标记，检查IBL引导记录与系统的兼容情况，校验系统配置并从所选的DASD（硬盘或软盘）中加载BIOS映象。数据段122－138包括了用于定义介质、区分并校验主引导记录、定位BIOS映象和加载BIOS映象的信息。

主引导记录由引导记录的特征122加以区分。引导记录特征可以为唯一的位的模式，如字符串“ABC”，它在该记录的开始三个字节中。主引导记录的完整性是由校验和的值132来测试的，该值要和该引导记录加载时的计算的校验和的值相比较。数据段还包括至少一个兼容的平板ID值134，兼容的型号和子型号值136。主引导记录的平板ID的值定义了主引导记录对于哪个平板是有效的。同样，主引导记录的型号和子型号值定义了主引导记录对应于哪个处理器和平板I／O配置是有效的。值得注意的是该引导记录的特征和校验和是区分有效的主引导记录的，而引导记录的平板ID，引导记录的型号和引导记录的子型号的比较是用来区分引导记录和系统的兼容情况并决定系统配置是否有效。另外一个值——引导记录模式124是用来决定ROM－BIOS的有效性。引导记录模式124与存储在ROM中的相应的模式值相比较，如果这两个值相匹配的话，它表明了有效的ROM－BIOS已经从所选的介质中起动了对BIOS映象的加载。

下面将更详细地描述主引导记录的每个值和它们的功能：

MBR标识符（122）：IBL引导记录的开始三个字节，它可以

由字符组成，如“ＡＢＣ”。该特征是用来区分引导记录的。

ＭＢＲ代码段（１２０）：该代码通过比较相应的平板ＩＤ和型号／子型号值来校验该引导记录与该平板及处理器的兼容性。如果这些值匹配的话，它将从所选择的介质中将ＢＩＯＳ映象加载到系统ＲＡＭ中。如果系统映象（ＢＩＯＳ映象加载到存储器）的校验和为有效并且没有介质加载错误发生的话，ＭＢＲ代码将控制传送给系统映象的ＰＯＳＴ步骤Ⅱ程序。

ＭＢＲ模式（１２４）：ＩＢＬ引导记录的数据段的第一个字段包含一个模式，如字符串“ＲＯＭ－ＢＩＯＳ１９８９”。通过将引导模式值和相应的存储在ＲＯＭ中的值（ＲＯＭ－模式）加以比较的方法，该字符串可用来校验ＲＯＭ－ＢＩＯＳ。

ＭＢＲ版本日期（１２６）：主引导记录包括由更新实用程序所使用的版本日期。

系统分区指针（１２８）：数据段含有由ＰＯＳＴ步骤Ⅱ使用的，并指向介质系统分区区域开始处的介质指针。该指针在ＩＢＬ软盘上是以磁道－磁头－扇区格式记录的，而在硬盘上是以相对块地址（ＲＢＡ）格式记录的。

系统分区类型（１３０）：系统分区类型表明介质系统分区的结构。系统分区可以有三种类型的系统分区结构—完全的，最小的和没有提出的三种类型。完全的系统分区除了具有ＢＩＯＳ映象和主引导记录外，还包括建立实用程序和诊断程序。最小的系统分区只有ＢＩＯＳ映象和主引导记录。它可能发生系统还没有存取具有ＩＢＬ映象的硬文件的情况，在这种情况下系统分区类型为“没有提出”。此时，将从软盘中产生ＩＢＬ。这三种类型的系统分区使系统分区在介质上占

据多大的空间上具有相当的灵活性。

校验和的值（１３２）：初始化数据段的校验和的值以使产生主引导记录代码的记录长度值（１．５Ｋ字节）的有效校验和。

ＭＢＲ平板ＩＤ的值（１３４）：数据段包括一个值，如定义兼容平板ＩＤ的字串。每个字由１６位的平板ＩＤ组成并且该字串以值为０的字结束。如果系统的平板ＩＤ和主引导记录的平板ＩＤ的值，如该字串中的一个字，相匹配的话，则ＩＢＬ介质映象和系统平板是兼容的。如果系统的平板ＩＤ没有和字串中的任意一个字相匹配的话，则ＩＢＬ介质映象和系统平板不兼容。

ＭＢＲ型号和子型号的值（１３６）：数据段包括这样的值，如定义与处理器兼容的字串。每个字由型号和子型号的值组成，并且该字串以值为零的字结束。如果系统的型号和子型号的值（存储在ＲＯＭ中）和字串中的一个字相匹配的话，则ＩＢＬ介质映象和系统处理器相兼容。如果ＲＯＭ型号和ＲＯＭ子型号值与字串中的任意一个字不相匹配的话，则ＩＢＬ介质映象和系统处理器不兼容。

ＭＢＲ映象图的长度（１３８）：初始化ＩＢＬ映象图的长度为介质映象块的数量。换言之，如ＢＩＯＳ映象分成４个块的话，则该映象图的长度为４，它表明为４个块指针／长度字段。通常，该长度置为１，因为该介质映象是一个连续的１２８Ｋ的块。

ＭＢＲ介质扇区大小（１３８）：初始化该字值为以每扇区的字节数为单位的介质扇区大小。

介质映象块指针（１３８）：介质映象块指针在介质上定位系统映象块。通常只有一个指针，因为介质映象是作为一个连读块来存储的。指针在ＩＢＬ软盘上是以磁道－磁头－扇区格式记录的，而在硬盘上

是以相对块地址格式记录的。

介质映象块长度（１３８）：介质映象块的长度表示位于对应的映象块指针上的块的大小（以扇区为单位）。在包括ＢＡＳＩＣ空间的１２８Ｋ连续介质映象情况下，该字段置为２５６，表明在介质映象块指针地址上开始的ＢＩＯＳ映象块占据２５６扇区（５１２字节／扇区）。

图６Ａ－Ｄ给出了ＩＢＬ程序操作的详细流程图。通常情况下，ＩＢＬ程序从系统硬盘上将主引导记录加载到ＲＡＭ的指定的地址上，然后，引导系统处理器开始执行主引导记录的代码段。ＩＢＬ程序也包含了对软盘隐含（缺席）方式的措施，在这种方式下是可以从软盘加载主引导记录。然而，如果系统在系统硬盘上具有ＩＢＬ介质且在ＮＶＲＡＭ中有有效的口令的话，ＩＢＬ程序不允许执行软盘隐含方式。用户具有在ＮＶＲＡＭ中设置口令的任选项。阻止软盘隐含方式有效的目的是防止从软盘中加载未授权的ＢＩＯＳ映象。换言之，软盘隐含方式只是用在当系统硬盘不能操作且用户具有希望能从软盘加载的指示（通过不设置口令）。如果ＩＢＬ程序不能从两种介质中加载主引导记录的话，则生成错误信息且系统停机。

现在参考图６Ａ，在通常情况下系统具有由ＩＢＬ程序初始化的系统硬盘，见步骤１５０。为了描述起见，假设硬盘配置为个人计算机系统的驱动器Ｃ。同样，假设驱动器Ａ为软盘驱动器。然后，ＩＢＬ程序检查驱动器Ｃ来决定它是否具有ＩＢＬ介质，见步骤１５２。请注意图６Ｂ，它详细地描述了该过程。ＩＢＬ程序从硬盘的最后三个扇区开始读，再将介质指针的值减去一常数，再连续读，直到读了９９个扇区或者有效的主引导记录找到为止。如果找到主引

导记录，它就检查与系统平板和处理器的兼容性，见步骤１５６。如果它与平板或处理器不兼容的话，则报告错误信息，见步骤１５８。再返回来讨论第１５２步，如果在硬盘（基本的硬文件）的最后９９个扇区没有找到主引导记录，则报告错误信息，见步骤１５４。

再返回来讨论步骤１５６，如果主引导记录找到的话，则执行一系列的正确性检查以便决定主引导记录是否与计算机系统兼容。此外，还检查了系统的配置。图６Ｄ更详细地揭露了该过程。如果该引导记录与平板ＩＤ、型号和子型号兼容的话，并且如果进一步检查系统配置没有修改的话，则加载主引导记录并且执行主引导记录的代码段，见步骤１６０。

回过来讨论步骤１５４和１５８，如果错误发生在从硬盘加载主引导记录时或者系统没有硬盘的话，则ＩＢＬ程序决定有效的口令是否包括在ＮＶＲＡＭ中，见步骤１６２。该口令决定了ＢＩＯＳ映象是否能从软盘加载。注意，只有当由用户运行设置特点的实用程序来安装系统时，该口令才会存在。如果口令安装在ＮＶＲＡＭ中，则阻止从软盘加载ＢＩＯＳ映象，见步骤１６４。这使用户通过只用硬盘上的ＢＩＯＳ映象来加载系统的方法来保证系统操作的完整性。该口令可以为存储在ＮＶＲＡＭ中的字符串的格式。

再回过来讨论１６２步骤，如果ＮＶＲＡＭ中的有效口令不存在的话，就允许从软盘中加载ＢＩＯＳ映象，且ＩＢＬ程序初始化软盘子系统，见步骤１６６。然后ＩＢＬ程序决定驱动器Ａ中的软盘上是否有ＩＢＬ介质，见步骤１６８。如果驱动器Ａ中没有ＩＢＬ介质，就产生错误信息，并通知用户驱动器中已经插入的是无效的软盘，见步骤１７０。然后系统就停机，见步骤１７２。请注意图６Ｃ，它对

第168步骤有更详尽的讨论。

再回来讨论第168步骤，在驱动器A检查了IBL介质之后，就将主引导记录加载到RAM中，並且执行主引导记录中的代码段，见第160步骤。重要的是，对于软盘而言，IBL程序没有包括硬盘系统所用的正确性检查。没有正确性检查的理由是为了从软盘中加载一个不兼容的IBL映象。例如，如将新的处理器加到系统中的话，则新的BIOS映象将包括在软盘上。由于当从硬盘加载时，新的处理器会引起正确性错误，所以IBL程序通过用从软盘加载BIOS映象的办法来提供旁路这些测试的能力。

下面扼要地重述一下，通过比较系统平板ID、处理器型号／子型号的值与主引导记录值的匹配性来检查主引导记录与系统的兼容性。对于硬盘而言，该检查首先在IBL程序72中进行的，然后再在IBL引导记录中进行。首先进行的检查（在IBL程序中）是肯定该引导记录是和系统兼容的。其次进行的检查（在引导记录中）是保证兼容的ROM将控制传送给了该引导记录。注意，在硬盘引导记录所进行的检查中，对于兼容的ROM而言是决不会失败的，因为IBL程序已经检查了兼容性。和上面相对比，对于软盘而言是不会进行兼容性检查的。只是在软盘引导记录执行期间，才会检查平板／处理器的兼容性。该方法可以使在从软盘中加载新的BIOS映象中作进一步的修改。

考虑到对图6A的IBL程序的描述，现在将对上面已讨论的正确性测试作一更广泛的和全面理解的解释。参考图6B，它给出了图6A的第152步的详细流程图，以决定有效的主引导记录是否在驱动器C上。通过获取该驱动器参数来使能IBL程序存取驱动器C的

IBL加载地址置为软盘的最后三个扇区（地址系以柱面，磁头和扇区的格式表示），见步骤232。读最后三个扇区，见步骤234。如检测到软盘驱动器错误的话，就指示出错，见步骤236－238。并置错误状态码再将控制返回给IBL程序，见步骤240－242。

再返回到第236步，如没有检测到驱动器错误的话，则检查软盘记录的引导记录特征且计算校验和，见步骤244。如引导记录特征漏掉的话或者校验和不正确的话，则指示出错，並将控制返回给IBL程序，见步骤244、246、240和242。如检测到正确的引导记录特征与正确的校验和的话，则设置该标志並将控制返回给IBL程序，见步骤248和242。值得注意的是在软盘加载时，IBL程序没有检索整个介质，正如在硬盘加载时那样。因此，在软盘加载时，IBL介质必须存储在软盘的指定的地址上。

最后，图6D给出了IBL程序是如何测试系统平板和处理器的兼容性以及正确的系统配置的。通过将引导记录平板ID的值和由系统处理器读出的系统平板ID值相比较，主引导记录检查与系统平板的兼容性，见步骤260。如果系统平板ID和该引导记录平板ID的值不匹配的话，它表明该主引导记录和该平板不兼容。则指示出错並将控制返回给IBL程序，见步骤262，264和266。

如主引导记录和该平板兼容的话，则主引导记录检查与处理器的兼容性，见步骤268。再将该引导记录的型号值和子型号值与存储在ROM中的相应的型号值和子型号值相比较。如不匹配的话，表明可能已经插入新的处理器且该引导记录和新处理器不兼容。则指示出错並将控制返回给IBL程序，见步骤270、264和266。如主引导记录与平板和处理器兼容的话，则处理器检查並决定NVRAM

是否可靠，见步骤２７２。如ＮＶＲＡＭ不可靠的话，则指示出错并将控制返回给ＩＢＬ程序，见步骤２７４和２６６。如ＮＶＲＡＭ可靠的话，则检查系统配置，见步骤２７６。如果存储在ＮＶＲＡＭ中的型号和子型号的值与存储在ＲＯＭ中的相应的值不匹配的话，则表明系统配置已有修改。注意，这最后的比较仅是表明配置的错误。如果指出配置错误的话，则给用户产生错误信息。该错误信息告诉用户，自从上一次运行ＳＥＴ配置程序以来，系统的配置已经有了修改。并通知用户修改的配置并将控制返回给ＩＢＬ程序，见步骤２７８、２６４和２６６。该错误对它本身而言并不是致命的，但是通知用户必须再执行ＳＥＴ配置（配置程序）。再返回讨论步骤２７６，如系统型号／子型号值匹配的话，则置兼容标志且该程序返回，见步骤２７６、２７４和２６６。因而，与决定系统配置是否已经被修改一起，测试了主引导记录和系统之间的兼容性。

在ＩＢＬ程序将主引导记录加载到ＲＡＭ之后，它将控制传送至ＭＢＲ代码的开始地址处。参见图７，主引导记录的可执行代码段首先检验该引导记录模式和ＲＯＭ模式，见步骤３００。如果在主引导记录的模式和ＲＯＭ中的模式不匹配的话，则产生错误信息且系统停机，见步骤３０２和３０５。检查ＲＯＭ和引导记录模式的一致性保证了从硬盘或软盘加载的主引导记录是和平板上的ＲＯＭ兼容的。再返回讨论第３００步，如果ＲＯＭ中的模式和引导记录中的模式相匹配的话，则ＭＢＲ代码将系统平板ＩＤ值、型号和子型号值与主引导记录中的相应的值加以比较，见步骤３０４。该过程已经在关于图６Ｄ的讨论中进行了较详细的讨论。如果这些值不匹配，则主引导记录和系统平板、处理器不兼容，或者系统配置已经修改，且产生错误

方法来开始该过程，见步骤２００。将ＩＢＬ加载地址置为硬盘的最后三个扇区。（通常在最后三个扇区上为主引导记录），见步骤２０２。将表示要从硬盘读主引导记录次数的加载记数器置为１，见步骤２０４。在ＩＢＬ加载地址上读硬盘的三个扇区，见步骤２０６。只要检测到任意的磁盘驱动器错误的话或者如果发生硬盘驱动器读错误的话，则报告出错信息，见步骤２０８－２１０。然后过程以错误标志而返回。见步骤２１２－２１４。

返回来讨论第２０８步，如果没有驱动器错误发生的话，则扫描硬盘记录的主引导记录的特征，见步骤２１６。将引导记录的特征，如字符“ＡＢＣ”，和该硬盘记录的开始三个字节相比较。如该硬盘记录确实具有正确的引导记录特征（字符“ＡＢＣ”）以及从加载到存储器的硬盘记录中计算的校验和等于该引导记录的校验和的话，表明该硬盘记录为无错误的正确的引导记录，然后该过程将返回，见步骤２１４。

返回来讨论第２１６步，如果该引导记录特征或校验和是不正确的话，则加载记数器加１，见步骤２２０。然后，该加载记数器与预置的常数如９９相比较，见步骤２２２。如９９次读引导记录得到的结果是失败的话，则表示出错且该过程返回，见步骤２２４、２１２和２１４。如果读引导记录的次数小于９９的话，则ＩＢＬ加载地址减一，且再从新的加载地址读三个新的扇区，见步骤２２６和２０６。因此，如果不能从最后的９９扇区（等效于３３次）加载有效的ＩＢＬ引导记录的话，则置错误状态码，且控制返回到ＩＢＬ程序。

现在讨论图６Ｃ，它为从驱动器Ａ的软盘加载主引导记录的详细流程图。首先，检索存取驱动器Ａ的软盘驱动器参数，见步骤２３０。

信息，见步骤３０６。当ＩＢＬ记录与平板、型号或子型号值不兼容的话，系统将停机，见步骤３０５。

再返回讨论第３０４步，如果系统平板ＩＤ值，型号和子型号的值与主引导记录中对应的值相匹配的话，则ＭＢＲ代码将从所选出的介质中ＢＩＯＳ映象加载到系统ＲＡＭ中，见步骤３０８。如果在读数据中介绍加载错误发生的话，见步骤３１０，则产生错误且系统停机，见步骤３１２和３０５。再返回讨论第３１０步，如果没有介质加载错误发生的话，则计算存储器中的ＢＩＯＳ映象的校验和，见步骤３１４。如校验和不正确，则产生错误且系统停机，见步骤３１８和３０５。再回过来讨论第３１６步，如校验和正确，则保存系统分区指针，见步骤３２０，且系统处理器引导到ＰＯＳＴ步骤Ⅱ并开始加载系统，见步骤３２２。

参见图８，它为控制磁盘驱动器３５１和系统处理器之间的数据传送的智能磁盘控制器框图。当然，磁盘控制器３５０可以合并到适配器卡６０中，而盘驱动器３５１可以包括在图２的驱动器６２中。一种适宜的磁盘控制器３５０是零年号为３３Ｆ８７４０的ＳＣＳＩ适配器，它是由ＩＢＭ公司生产的。当然磁盘控制器３５０包括一个在它自己内部时钟操作下的微处理器，除了用于与磁盘子系统的其它部件和系统处理器接口外，还用于控制它的内部操作。指令总线３５４将微处理器３５２连到存储指令的只读存储器（ＲＯＭ）３５６，而磁盘控制器３５０执行这些指令来处理和控制磁盘驱动器与系统处理器之间的数据传送。当然也可以说，磁盘控制器３５０可以包括连到微处理器３５２並用于存储或检索数据的随机存取存储器。磁盘控制器３５０与系统处理器之间的数据传送是通过数据总线

３５８与指令总线３６０来实现的。在线３６２上的复位信号在加电顺序或系统复位期间复位或初始化磁盘控制器逻辑。复位信号由平板逻辑生成，且它可采用通道复位信号的格式。正如在“ＩＢＭ　ＰＳ／２研讨会会刊”第5卷第３号中描述那样，ＩＢＭ公司输入系统部１９８７年5月出版，该通道复位信号是由ＩＢＭ的微通道结构所提供的。而且通过ＢＩＯＳ将一个特别的位配置输出到连接平板逻辑的系统处理器的Ｉ／Ｏ端口的办法，可以有效地初始该复位信号。

众所周知，为了实现磁盘驱动器和系统处理器之间的有效的数据传送，微处理器３５２提供了所有接口和定时信号。为了叙述明瞭起见，下面只给出了对于理解本发明有重要意义的那些信号。当然，其它的信号和信号线，如数据总线３６４，在系统中是使用的，但是这儿並没有给出，因为它们对于理解本发明並不重要。当然，图９解释的是那些对于理解本发明有重要意义的存储在ＲＯＭ３５６中的程序或例行程序。

现在参考图９，它是由存储在ＲＯＭ３５６中的例行程序的操作来实现的磁盘控制器的读、写和保护功能的流程图。在操作中，磁盘指令是由系统处理器初始化的，並传送到磁盘控制器３５０中。为了执行指定的操作，磁盘控制器接收並解释指令，见步骤４００。磁盘控制器首先决定该操作是否为写操作，在写操作中从系统处理器来的数据是存储在磁盘驱动器硬件上，见步骤４０２。如果该指令为写指令，则从系统处理器接收的数据格式为相对块地址（ＲＢＡ）格式。

在继续上述讨论之前，简单地解释应用在海量存储器设备如磁盘上的相对块地址格式是值得的。ＲＢＡ是一种方法，在这种方法中，存储在海量存储器中的数据是以预定大小的块並按照顺序号来寻址

的，即按照单个可定义的连读数据块来寻址。例如，假设块的大小为１０２４字节，对于１０兆字节硬盘而言，系统处理器约可寻址１０，０００块。即系统处理器可以寻址Ｎ块的磁盘介质，其中Ｎ为０至９９９９。已经发现，在本发明的个人计算机系统所用的操作系统中，ＲＢＡ的使用提供了非常快和有效的寻址海量存储器的方法。

为了方便起见，下面假设：第一，磁盘共能支持Ｎ块；第二，系统处理器传送Ｋ块，其中Ｋ为大于等于０且小于等于（Ｎ－１）；第三，磁盘控制器可以设置最大可寻址块为Ｍ；其中，如允许存取数据块，则Ｋ为小于Ｍ，如拒绝存取数据块，则Ｋ大于等于Ｍ。注意，通过设置Ｍ小于Ｎ的办法，则可从Ｍ至Ｎ－１块中生成一个在磁盘上的保护区域。该特点使ＩＢＬ介质得到保护，正如下文将要描述的那样。

我们继续讨论图９，从磁盘接收的数据格式为ＲＢＡ格式，见步骤４０４。然后磁盘控制器决定接收的块Ｋ是否小于最大的块值Ｍ，其中Ｍ小于Ｎ，见步骤４０６。如果Ｋ小于Ｍ，则磁盘控制器将ＲＢＡ格式转换为海量存储器设备的特殊格式，如对硬盘而言的柱面－磁头－扇区（ＣＨＳ）格式，见步骤４０８。例如，磁盘控制器通过使用查表的方法可以将ＲＢＡ地址转换为唯一的柱面－磁头－扇区的地址。另一方法是使用转换公式将ＲＢＡ转换为ＣＨＳ。对于具有一个磁头，６４个柱面，９６扇区的磁盘而言：磁头＝０，柱面＝ＲＢＡ／（９６）的商，而扇区＝ＲＢＡ／（９６）的余数。在将ＲＢＡ格式转换为ＣＨＳ格式之后，将数据写到磁盘上已转换的ＣＨＳ地址上，见步骤４１０。然后磁盘控制器场待系统处理器的另一指令，见步骤４１２。

再返回来讨论４０６步骤，如果接收的ＲＢＡ大于最大设置的ＲＢＡ值，就拒绝存取，见步骤４１４。即，如果Ｋ大于等于Ｍ，则不会将Ｋ块写到盘中。请注意，如ＩＢＬ介质存储在Ｍ至Ｍ－１块中，则会保护ＩＢＬ介质免受写的破坏。

再返回来讨论４０２步骤，如果从系统处理器来的指令不是写指令，则测试是否为读指令，见步骤４１６。如指令为读指令，系统处理器对于所请求的数据发送ＲＢＡ格式的数据，见步骤４１８。然后磁盘控制器决定所请求的ＲＢＡ（Ｋ）是否小于最大设置ＲＢＡ(Ｍ)值。如果所请求的ＲＢＡ（Ｋ）小于最大设置ＲＢＡ（Ｍ）值，则磁盘控制器将该ＲＢＡ转换为相应的ＣＨＳ格式，并从磁盘中读数据，见步骤４２２和４２４。然后再将数据传送到系统处理器，见步骤４１２。

再返回讨论第４２０步骤，如所接收的ＲＢＡ（Ｋ）大于或等于最大设置ＲＢＡ（Ｍ）值，则禁止存取，见步骤４２６。如果ＩＢＬ介质存储在Ｍ块和（Ｎ－１）块之间，则禁止存取这个区域。请注意，在这种情况下，也保护了ＩＢＬ介质以禁止复制其上的数据。

返回讨论第４１６步骤，如指令不是写指令或读指令的话，则测试它是否为设置最大ＲＢＡ指令，见步骤４２８。该指令使磁盘控制器在磁盘驱动器硬件上建立可保护的区域或分区。该指令使磁盘控制器将Ｍ设置在０与Ｎ块之见，见步骤４３０。重要的是要注意，当磁盘控制器复位时（通过复位信号），则将Ｍ置位，因此可得到最大的块数。即，当磁盘控制器复位时，Ｍ＝Ｎ。基本上，在复位磁盘控制器时会取消对可保护区域的保护，即允许存取该区域。然而，一当执行了设置最大ＲＢＡ指令以后，只有复位或者另一个设置最大ＲＢＡ

指令将允许存取可保护的区域。从概念上讲，设置最大的ＲＢＡ可以认为是设置一个围墙，在围墙之上，将保护存取该区域，而在围墙以下则允许存取该区域。然后，磁盘控制器返回去等待另一指令，见步骤４１２。

返回来讨论第４２８步骤，如果指令不是读、写或设置最大RBA指令，则将测试它是否为另一个磁盘控制器指令並执行它，见步骤４３２。这些指令会使用设置最大ＲＢＡ值，但是对理解本发明是不重要的，並且为了简洁起见，这儿就不再陈述了。然后，磁盘控制器返回去等待另一指令，见步骤４１２。

考虑到所进行的讨论，现在要进行对加载和保护ＩＢＬ介质的操作的解释。一般地说，无论是冷起动（加电）还是热起动（同时按ＡＬＴ－ＣＴＲＬ－ＬＥＴ键），均令复位具有ＩＢＬ介质的磁盘控制器。这使最大ＲＢＡ（Ｍ）设置为Ｎ，即把围墙除去了，就可存取ＩＢＬ介质。需要上述过程来允许系统加载ＩＢＬ介质並开始操作。一当加载並执行ＩＢＬ介质之后，就建立了围墙（设置最大ＲＢＡ小于ＩＢＬ介质的相应值），以便阻止对存储在磁盘上的ＩＢＬ介质的存取。

现在讨论图１０，它为实现对ＩＢＬ介质保护的流程框图。从加电状态起，初始化系统並且开始平板逻辑上的活动，将复位状态送到磁盘控制器，见步骤４５０和４５２。复位信号删除了“围墙”，並且使系统处理器存取以前存储在磁盘Ｍ块至Ｎ块的区域中的ＩＢＬ介质。系统则加载ＩＢＬ介质，就像以前关于图４－７描述的那样。在ＩＢＬ加载顺序期间，执行ＰＯＳＴ步骤Ⅱ，见步骤４５６。ＰＯＳＴ步骤Ⅱ的任务之一是用最大ＲＢＡ设置来执行设置最大ＲＢＡ指令，

而指定为M的最大的RBA设置送到IBL介质的第一块，见步骤458。正如以前所解释的那样，M取决于分区的类型（没有，部分或全部）。实际上，这就设置了围墙、不准存取IBL介质而允许存取磁盘的其它的区域。然后操作系统以正常方式引导起来了，见步骤460。

如果系统以热起动方式起动的话，如同时按ALT－CTRL－DEL键，POST步骤II命令平板逻辑来复位磁盘控制器，见步骤462和464。这就使围墙被删除掉了。在这种情况下，由于IBL介质经存在于RAM中，所以不再加载IBL介质了。然而，由于对IBL介质的保护被删除了，所以必须执行POST步骤II来复位围墙，见步骤456与458。因此围墙建立了，从而保护了IBL介质，然后系统以正常方式重新引导，见步骤460。

因此，已经说明了保护存取存储在海量存储器，如磁盘驱动器上的IBL介质的方法和设备。通过以块的方式寻址海量存储器和设置正常操作期间系统可以存取的最大的块的方法来保护IBL介质。IBL介质是连续存储在最大可存取的块与磁盘驱动器支持的总的块数之间的那些块中。复位信号送到磁盘控制器以删除最大的可存取块，并允许系统寻址IBL介质。在加电状态或热起动状态期间，产生复位信号来允许存取IBL介质，再引导操作系统。

在优选的具体化的描述中已经说明了本发明后，那些技术上一般熟练的人可对本发明作许多变化，因此本发明的范围只是通过本文所附的本专利范围及高效的范围才予以定义。

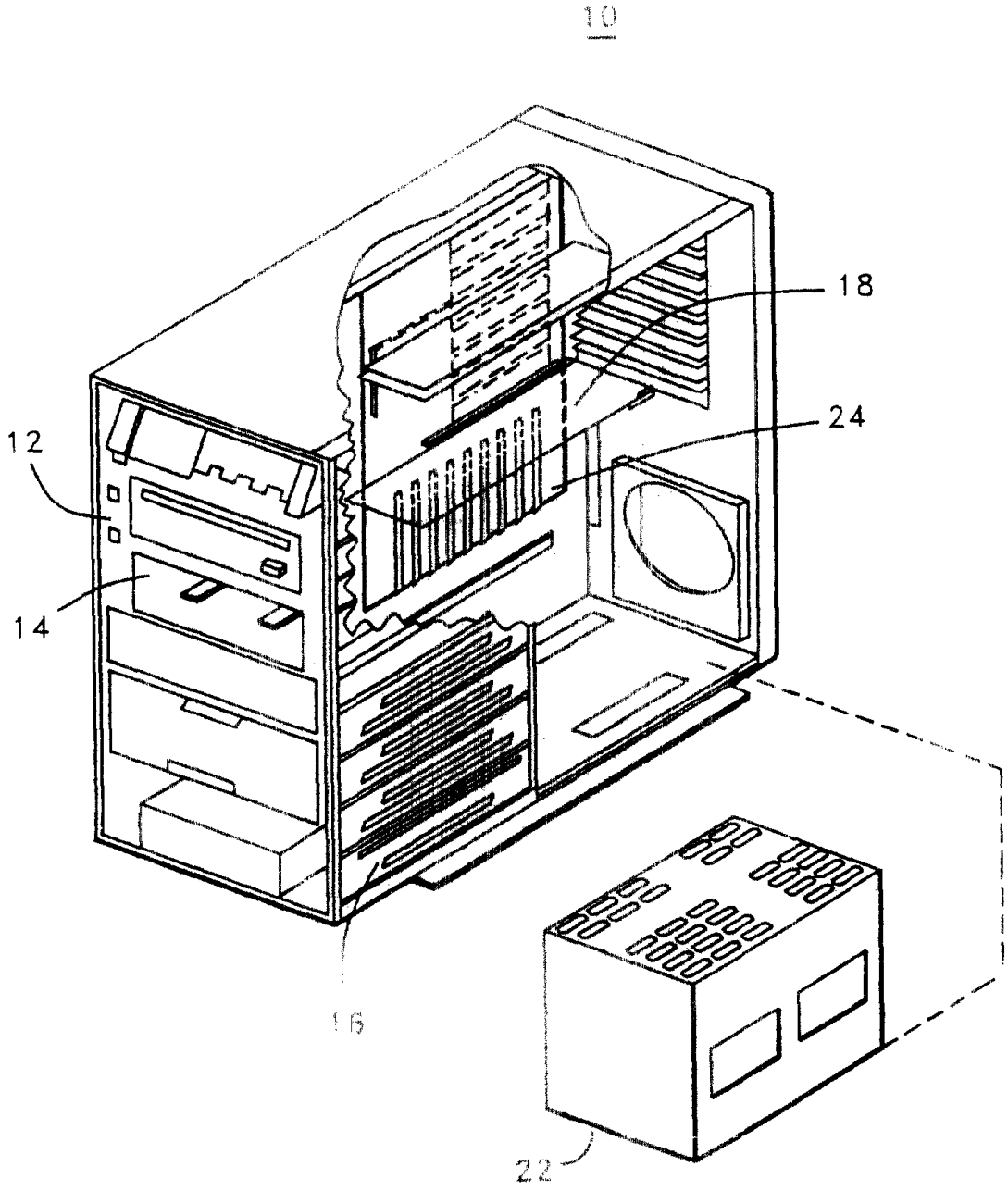


图1

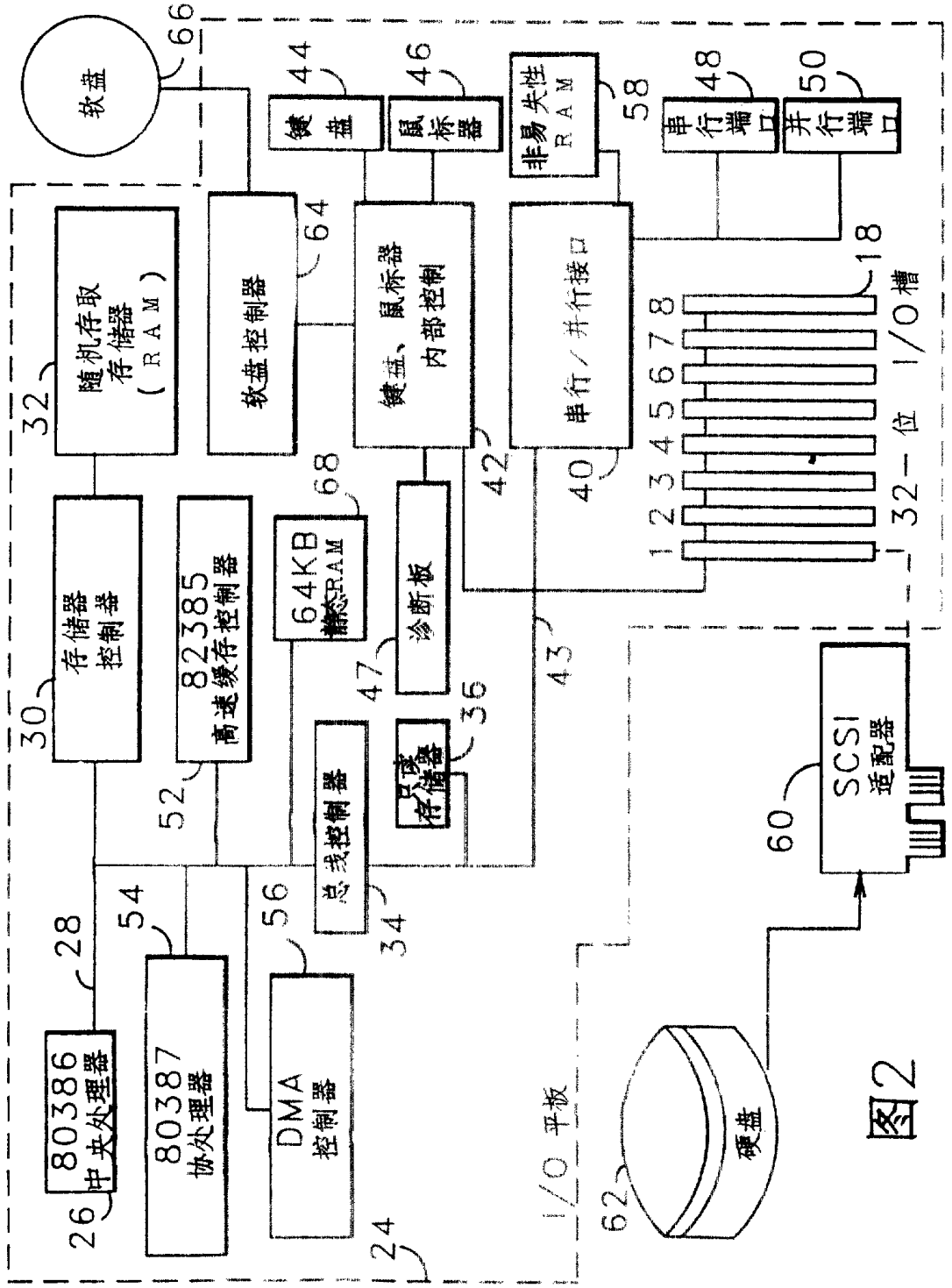

软盘
66
键盘
44
鼠标器
46
非易失性 RAM
58
串行端口
48
并行端口
50
随机存取存储器 (RAM)
32
软盘控制器
64
键盘、鼠标器内部控制
42
串行/并行接口
40
18
1 2 3 4 5 6 7 8
32-位 I/O 槽
存储器控制器
30
82385 高速缓存控制器
52
64KB 静态RAM
68
诊断板
47
43
总线控制器
34
存储器
36
80386 中央处理器
28
26
80387 协处理器
54
DMA 控制器
56
24
I/O 平板
SCSI 适配器
60
硬盘
62

图2

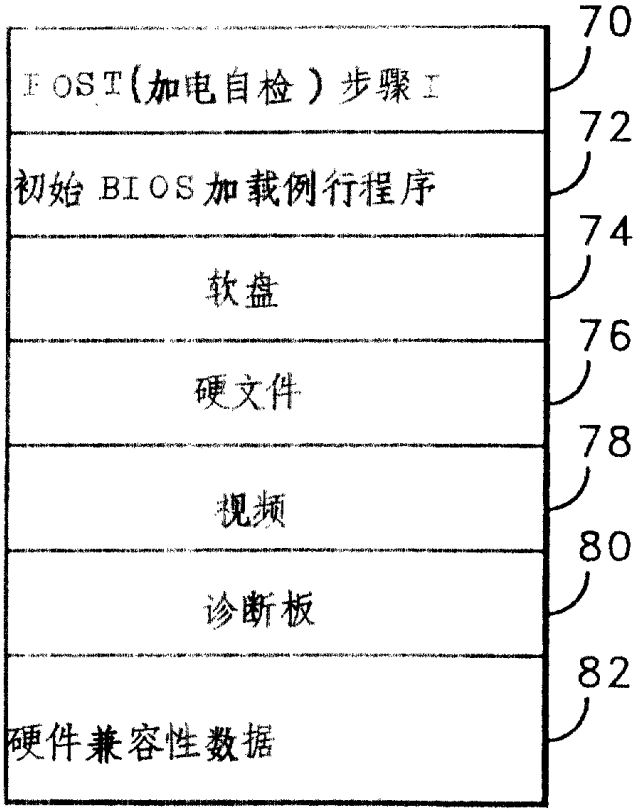

POST(加电自检)步骤I
初始BIOS加载例行程序
软盘
硬文件
视频
诊断板
硬件兼容性数据
70
72
74
76
78
80
82
ROM-BIOS

图3

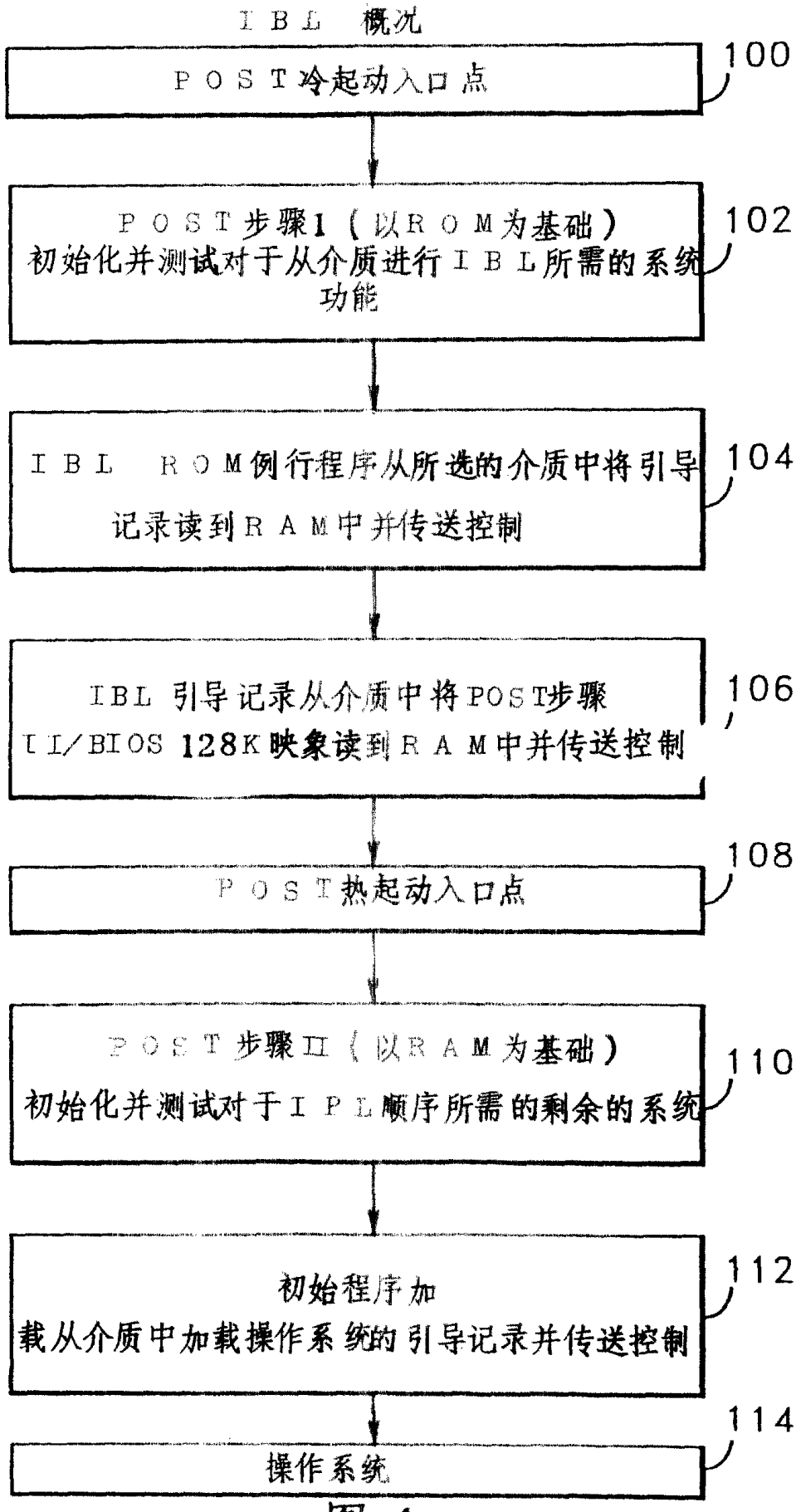

IBL 概况
POST冷起动入口点
100
POST步骤1（以ROM为基础）
初始化并测试对于从介质进行IBL所需的系统功能
102
IBL ROM例行程序从所选的介质中将引导记录读到RAM中并传送控制
104
IBL 引导记录从介质中将POST步骤II/BIOS 128K映象读到RAM中并传送控制
106
POST热起动入口点
108
POST步骤II（以RAM为基础）
初始化并测试对于IPL顺序所需的剩余的系统
110
初始程序加
载从介质中加载操作系统的引导记录并传送控制
112
操作系统
114

图 4

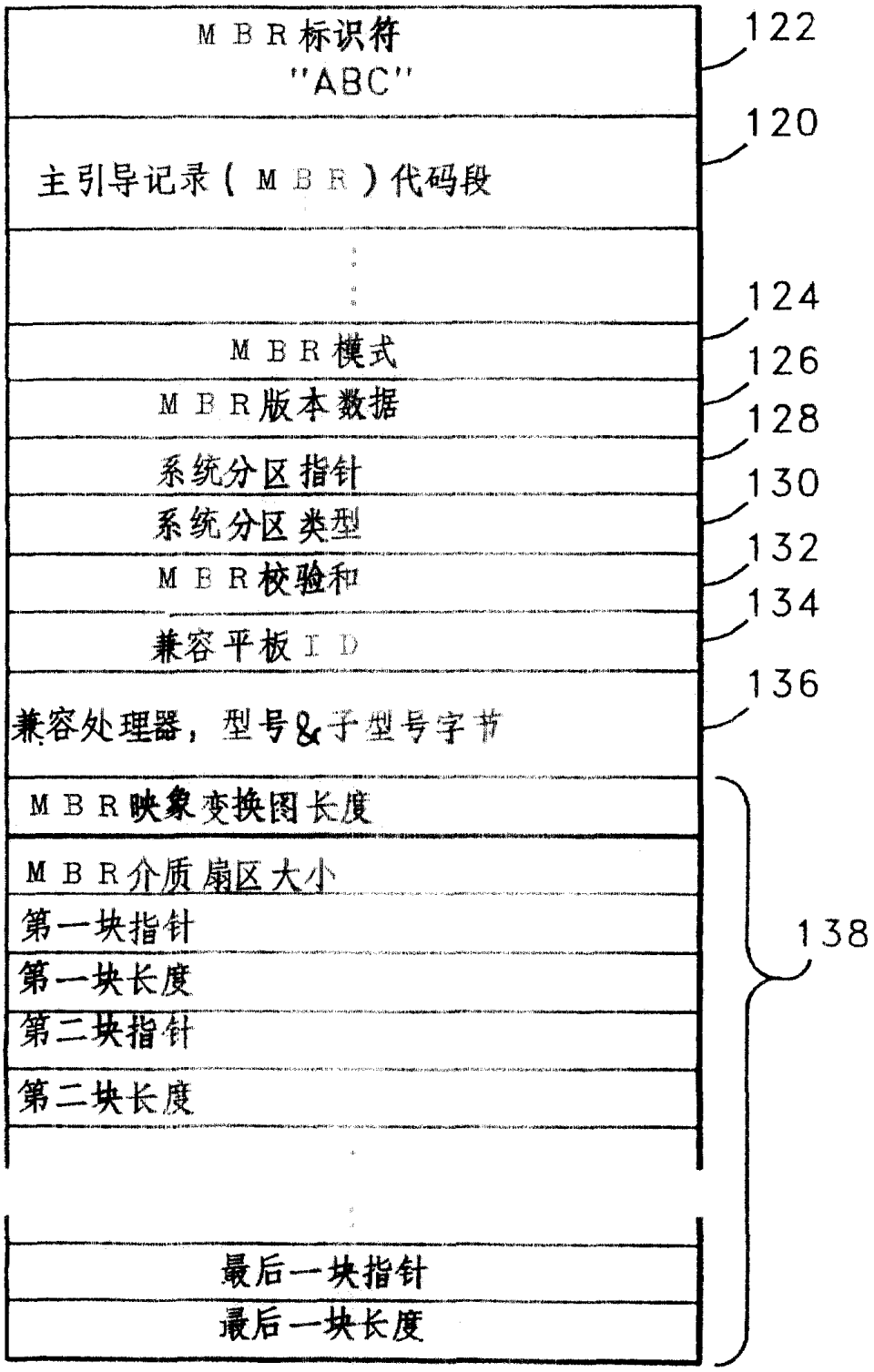

MBR标识符
"ABC"
122
主引导记录（MBR）代码段
120
124
MBR模式
126
MBR版本数据
128
系统分区指针
130
系统分区类型
132
MBR校验和
134
兼容平板ID
136
兼容处理器，型号&子型号字节
MBR映象变换图长度
MBR介质扇区大小
第一块指针
第一块长度
第二块指针
第二块长度
最后一块指针
最后一块长度
138

图5

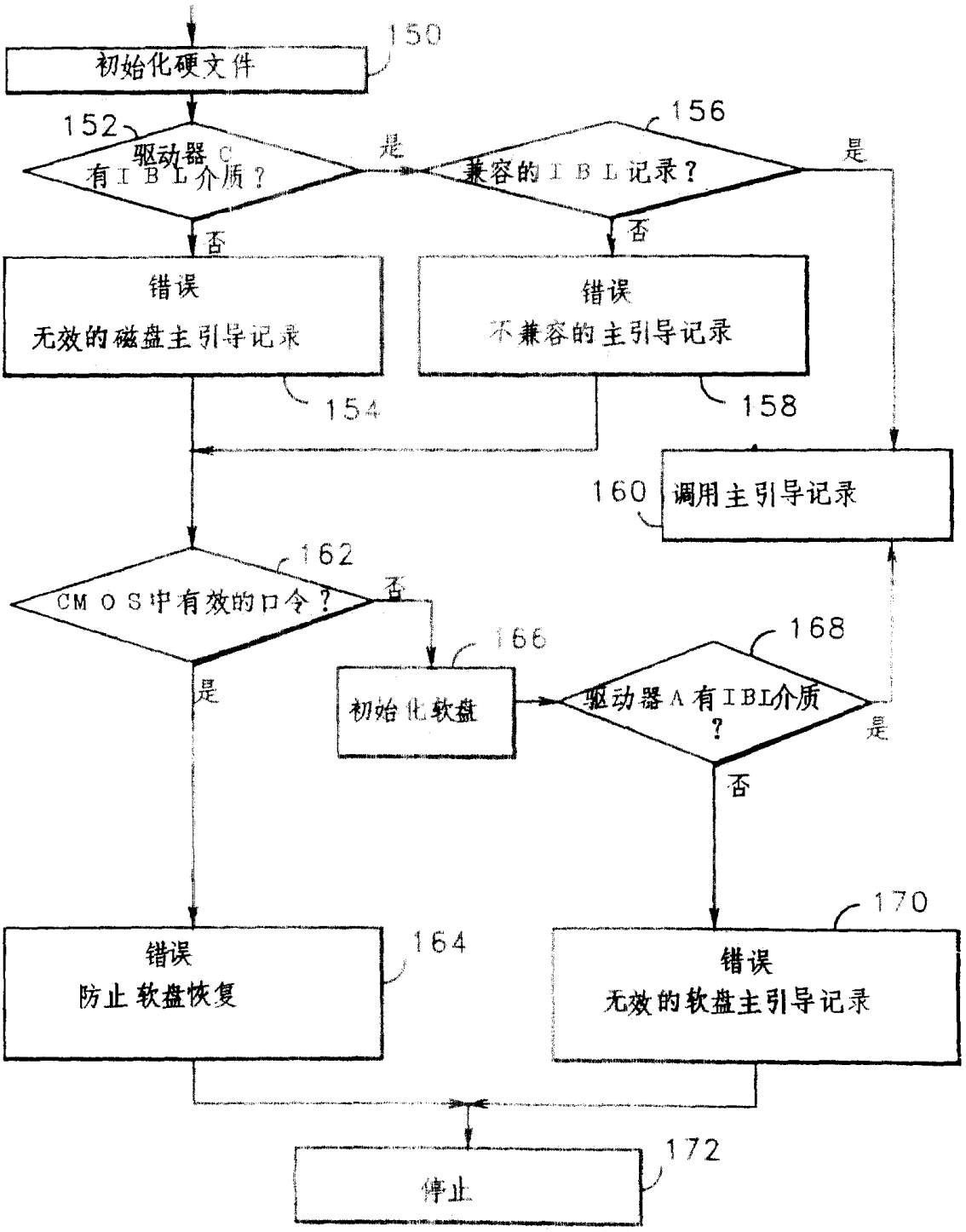

初始化硬文件
150
152
驱动器C有IBL介质？
是
156
兼容的IBL记录？
是
否
否
错误
无效的磁盘主引导记录
错误
不兼容的主引导记录
154
158
160
调用主引导记录
162
CMOS中有效的口令？
否
是
166
初始化软盘
168
驱动器A有IBL介质？
是
否
164
错误
防止软盘恢复
170
错误
无效的软盘主引导记录
172
停止

图6A

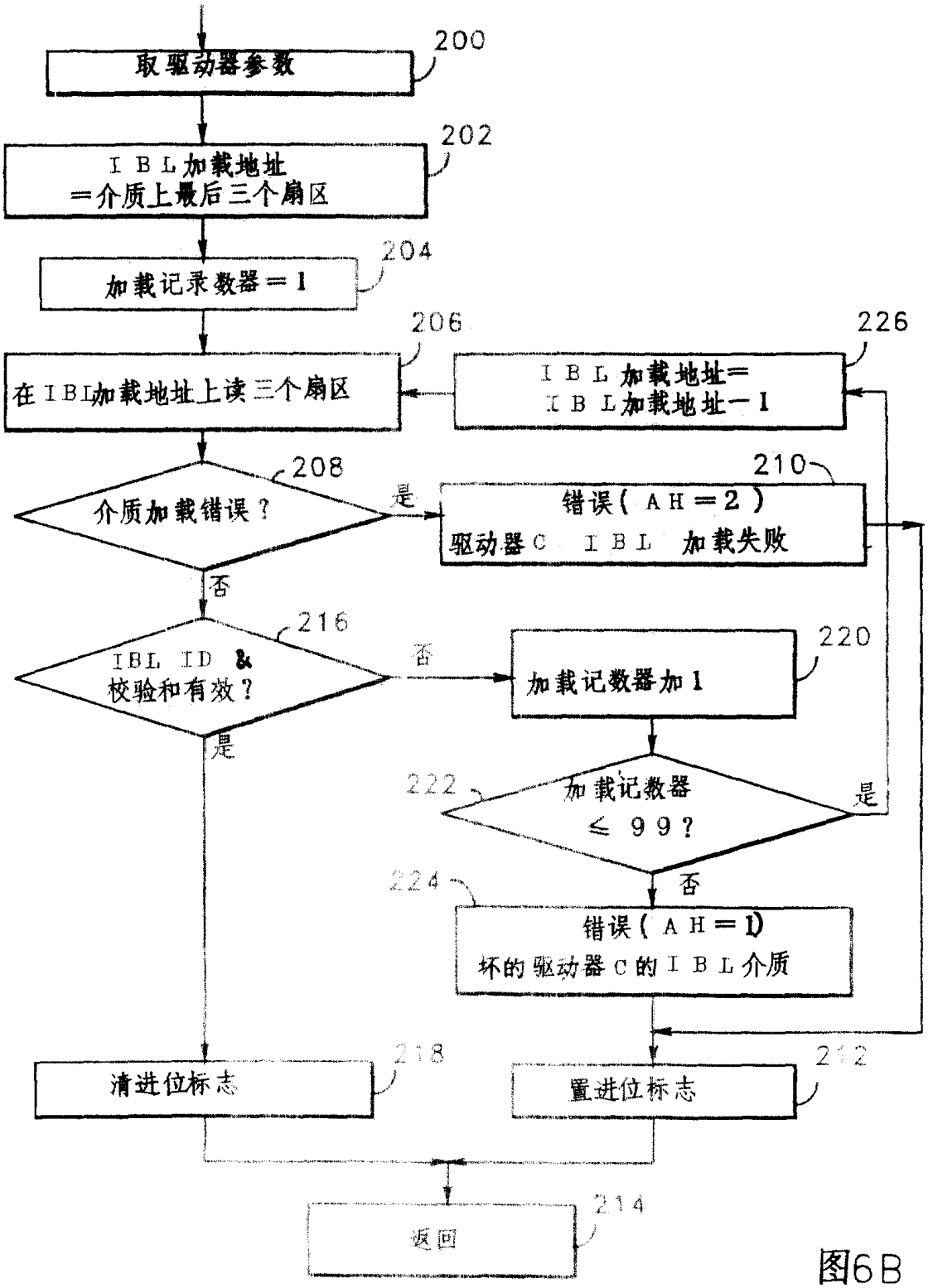

200
取驱动器参数
202
IBL加载地址
=介质上最后三个扇区
204
加载记录数器=1
206
在IBL加载地址上读三个扇区
226
IBL加载地址=
IBL加载地址-1
208
介质加载错误?
是
210
错误(AH=2)
驱动器C IBL 加载失败
否
216
IBL ID &
校验和有效?
否
220
加载记数器加1
是
222
加载记数器
≤99?
是
否
224
错误(AH=1)
坏的驱动器C的IBL介质
218
清进位标志
212
置进位标志
214
返回

图6B

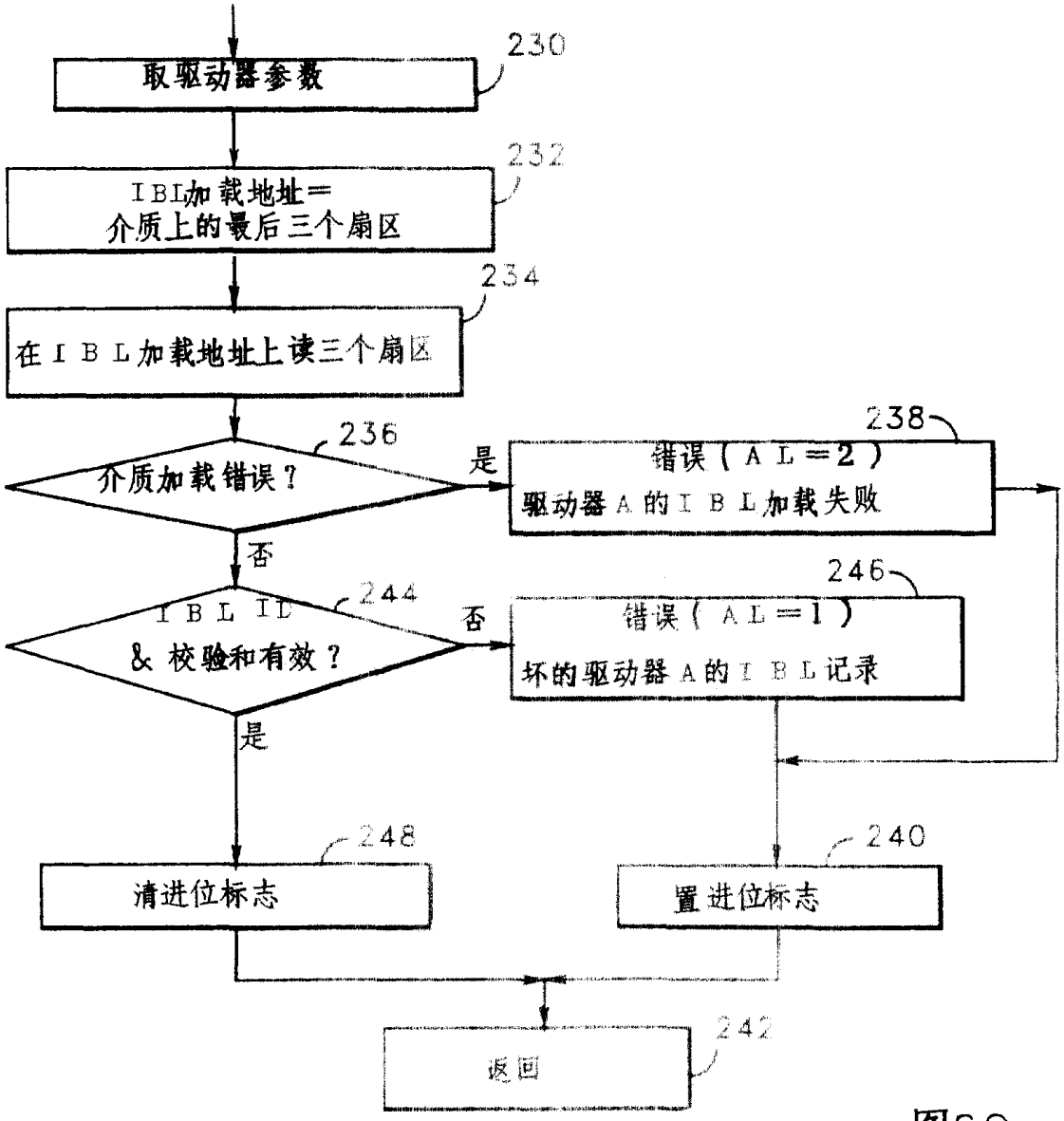

取驱动器参数
230
IBL加载地址=
介质上的最后三个扇区
232
在IBL加载地址上读三个扇区
234
介质加载错误？
236
是
错误（AL=2）
驱动器A的IBL加载失败
238
否
IBL ID
& 校验和有效？
244
否
错误（AL=1）
坏的驱动器A的IBL记录
246
是
清进位标志
248
置进位标志
240
返回
242

图6C

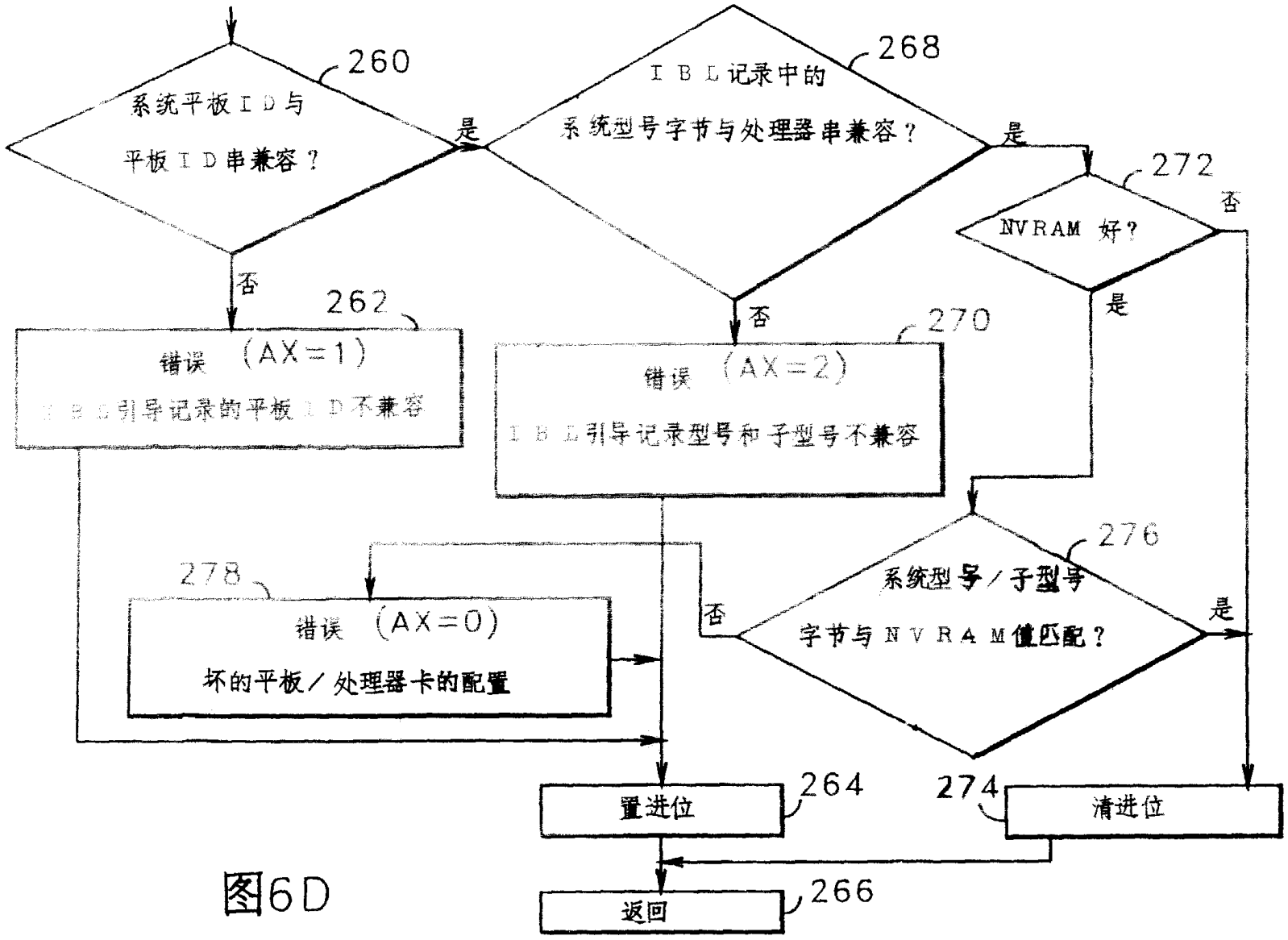

260
系统平板ID与
平板ID串兼容？
是
否
262
错误 (AX=1)
IBL引导记录的平板ID不兼容
268
IBL记录中的
系统型号字节与处理器串兼容？
是
否
270
错误 (AX=2)
IBL引导记录型号和子型号不兼容
272
NVRAM 好？
否
是
276
系统型号/子型号
字节与NVRAM值匹配？
否
是
278
错误 (AX=0)
坏的平板/处理器卡的配置
264
置进位
274
清进位
266
返回

图6D

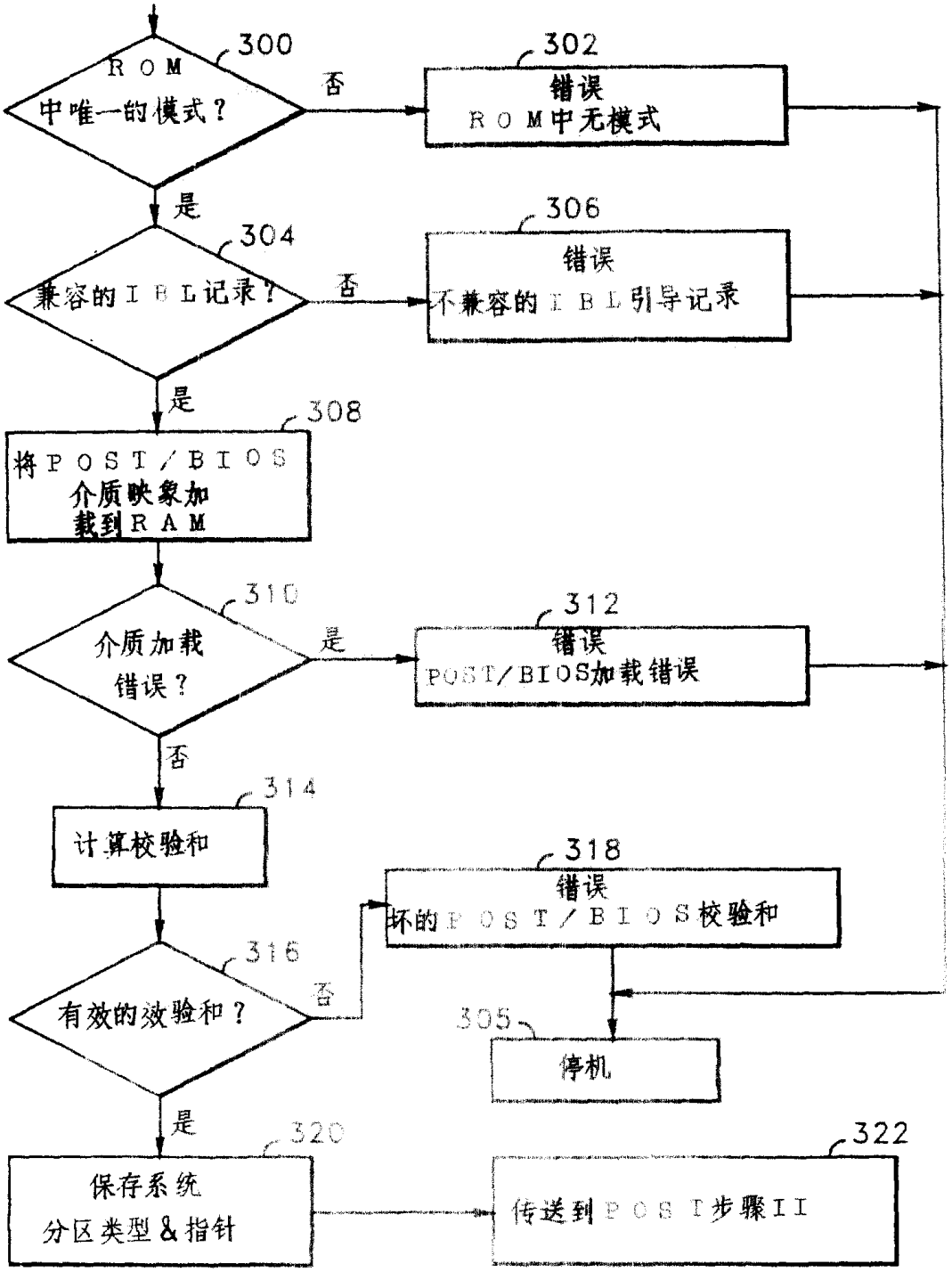

300
ROM
中唯一的模式？
否
302
错误
ROM中无模式
是
304
兼容的IBL记录？
否
306
错误
不兼容的IBL引导记录
是
308
将POST/BIOS
介质映象加
载到RAM
310
介质加载
错误？
是
312
错误
POST/BIOS加载错误
否
314
计算校验和
318
错误
坏的POST/BIOS校验和
316
有效的效验和？
否
305
停机
是
320
保存系统
分区类型&指针
322
传送到POST步骤II

图7

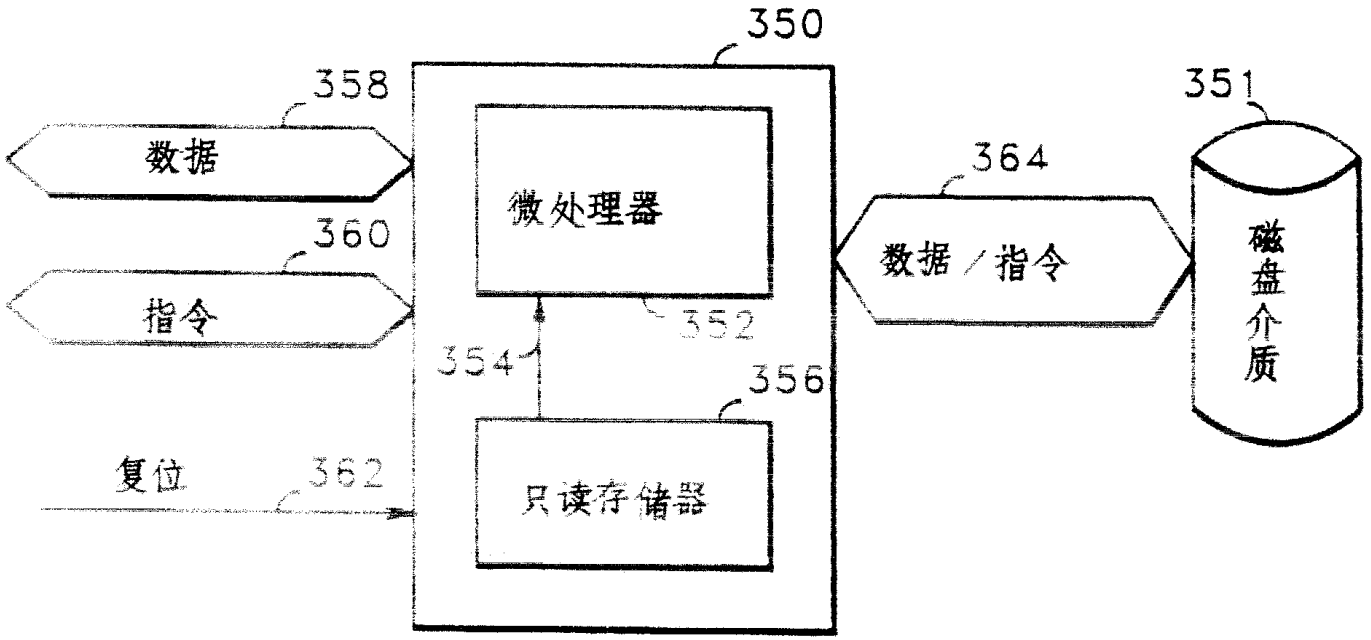

350
358
数据
360
指令
微处理器
352
354
356
只读存储器
复位
362
364
数据／指令
351
磁盘介质

图 8

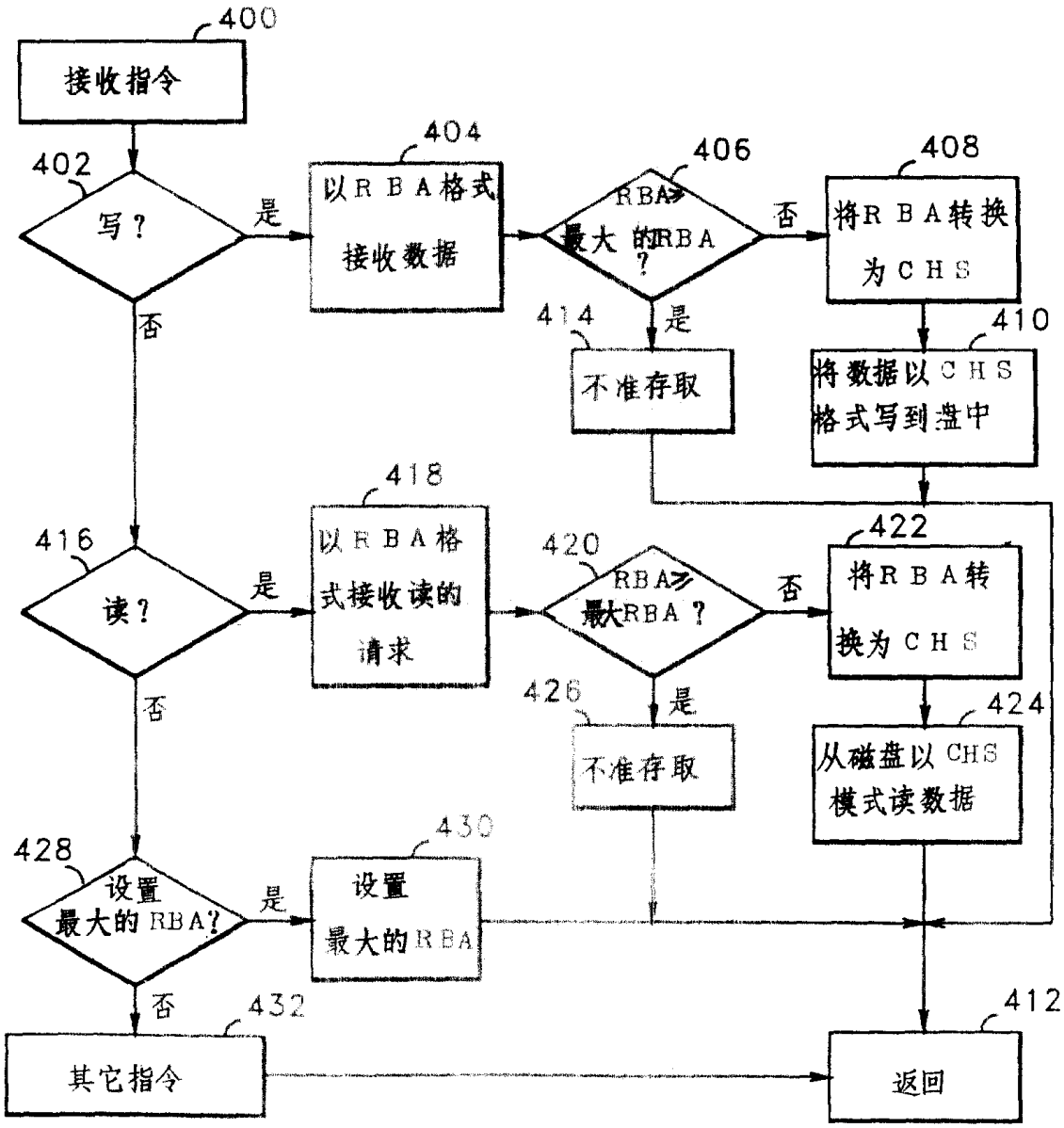

400
接收指令
402
写？
是
否
404
以RBA格式接收数据
406
RBA>最大的RBA？
否
是
408
将RBA转换为CHS
410
将数据以CHS格式写到盘中
414
不准存取
416
读？
是
否
418
以RBA格式接收读的请求
420
RBA>最大RBA？
否
是
422
将RBA转换为CHS
424
从磁盘以CHS模式读数据
426
不准存取
428
设置最大的RBA？
是
否
430
设置最大的RBA
432
其它指令
412
返回

图 9

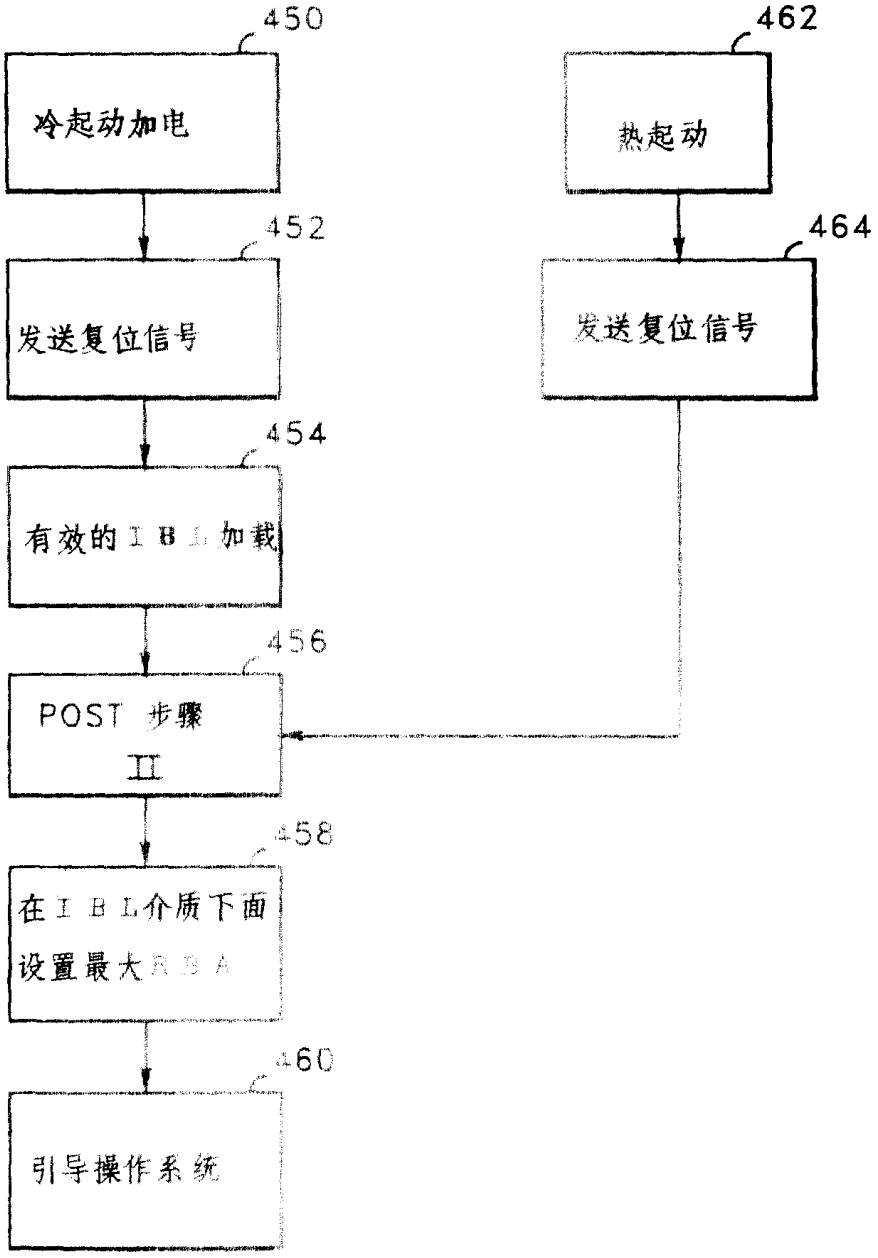

450
冷起动加电
452
发送复位信号
454
有效的IBL加载
456
POST 步骤
II
458
在IBL介质下面
设置最大RBA
460
引导操作系统
462
热起动
464
发送复位信号

图10