

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.



[12] 发明专利申请公布说明书

G11C 7/10 (2006.01)
G11C 29/24 (2006.01)
G11C 29/44 (2006.01)
G11C 16/06 (2006.01)

[21] 申请号 200810002657.6

[43] 公开日 2009年4月1日

[11] 公开号 CN 101399075A

[22] 申请日 2008.1.14

[21] 申请号 200810002657.6

[30] 优先权

[32] 2007.9.28 [33] US [31] 11/864,684

[71] 申请人 智多星电子科技有限公司

地址 美国加州圣荷西市

[72] 发明人 马治刚 周圭璋 李中和 俞一康

李威若 沈明祥

[74] 专利代理机构 中科专利商标代理有限责任公
司

代理人 汤保平

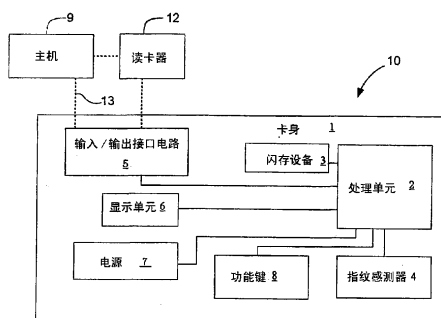
权利要求书 3 页 说明书 48 页 附图 15 页

[54] 发明名称

具多样闪存单元坏块管理的电子数据闪存卡

[57] 摘要

一种具多样闪存单元坏块管理的电子数据闪存卡，包括可通过主机存取电子数据闪存卡，电子数据闪存卡由连接至闪存设备的闪存控制器和输入-输出接口电路构成。在一个实施方案中，闪存卡采用 USB 接口电路和主机通讯。闪存控制器包括仲裁器，用于建立逻辑地址和实体地址之间的映像和执行块管理操作，块管理操作包括：在可用块中存储重新分配的数据、把过期块中的有效数据重新放置到上述可用块中并把逻辑块地址重新分配给上述可用块实体块地址、搜索闪存设备的坏块并用保留块替换、有效数据被重新放置到可用块中后擦除过期块进行回收以及块的擦除计数损耗平衡等。此外，各闪存设备都设有内部缓冲区用于加速块管理操作。



1. 一种电子数据闪存卡，可通过建立通讯连接被主机存取，其特征在于，该电子数据闪存卡包括：

一卡身；

一个或多个安装于该卡身的闪存设备，一个或多个闪存设备包含数个多层单元存储器单元用于存储数据文件；

一安装于卡身的输入/输出接口电路，用于建立和主机之间的通讯；

一安装于卡身的闪存控制器，与所述的闪存设备和所述的输入/输出接口电路之间的电气连接，其中该闪存控制器可在一数据检索模式下操作，从多层单元存储器单元的一块读取数据以决定是否在该数据读取中有任何错误以及如果在该读取数据有错误并决定在该读取数据中是否错位的数量超过预定极限和错误位的数量超过预定极限，即拷贝多层单元存储器单元的块的数据至该多层单元存储器单元的一闲置目标块和在逻辑到实体块位置表中更新相对应的逻辑到实体块位置映像。

2. 如权利要求1所述的电子数据闪存卡，其特

征在于，其中前述闪存控制器在拷贝多层单元存储器单元的块的数据至该闲置目标块后，还可在数据检索模式操作以擦除多层单元存储器单元的块。

3. 如权利要求1所述的电子数据闪存卡，其特征在于，其中前述闪存控制器在拷贝多层单元存储器单元的块的数据至该闲置目标块后，还可在数据检索模式操作以记录多层单元存储器单元的块为一坏块。

4. 如权利要求1所述的电子数据闪存卡，其特征在于，其中前述闪存控制器可在一编程模式操作将数据写入该多层单元存储器单元的一第二个块，以决定是否该第二个块可写入以及在数据写入该第二个块时，是否有任何错误，和如果该第二个块不可写入或数据写入该第二个块时有错误，上述闪存控制器还可在随机操作选取一第二个闲置好块为一第二个目标块，以将数据写入该第二个目标块和在逻辑到实体块位置表中更新相对应的逻辑到实体块映像。

5. 如权利要求4所述的电子数据闪存卡，其特征在于，其中，如果在数据写入该第二个块有错误时，上述闪存控制器还可在该编程模式下操作，拷贝在该第二个块有效扇区至该第二个块目标块和标示该第二个块为坏的。

6. 如权利要求1所述的电子数据闪存卡，其特

征在于，其中输入/输出接口电路为通用串行总线接口电路，该通用串行总线接口电路包含采用 BOT 协议传输数据的手段。

7. 如权利要求 1 所述的电子数据闪存卡，其特征在于，其中输入/输出接口电路可采用 Secure Digital SD 接口电路、Multi-Media Card MMC 接口电路、Compact Flash CF 接口电路、存储棒 Memory Stick MS 接口电路、PCI-Express 接口电路、Integrated Drive Electronics IDE 接口电路或 Serial Advanced Technology Attachment SATA 接口电路。

具多样闪存单元坏块管理的电子数据闪存卡

技术领域

本发明涉及电子数据闪存卡，更具体地说，是关于电子数据闪存卡中闪存设备的闪存块控制的系统和方法。

背景技术

机密数据文件通常存储在软盘中，或者通过需要口令或者安全加密的网络传送。机密文档则通过安全封印发送。然而，口令、加密、安全封印都有可能遭到破坏（被解密），从而给机密数据文件和机密文档带来危险，导致机密信息被未授权存取。

随着闪存技术变得越来越先进，闪存正逐步取代在移动系统中作为存储介质的传统磁盘的地位。和软盘或磁性介质硬盘相比，闪存具有以下一些显著优点：高阻抗和低功率耗散。由于闪存的实体尺寸小，更加有利于移动系统的发展。相应地，闪存的发展也得益于其与移动系统的兼容性和低功耗。

然而，闪存也有其固有限制。首先，已编程的闪存单元必须经过擦除才能进行再次编程。同时，闪存单元只有有限的使用寿命；即，闪存单元在失效前只能进行有限次数的擦除操作。例如，NAND 闪存单元典型的最大擦除次数为一百万次。相应地，由于“写入前擦除”的特点，闪存存取速度较慢，而且反复的擦除操作将会损坏闪存单元。

闪存设备的存储单元阵列由典型的基本结构构成，即分为“扇区”或“页面”，并由“扇区”或“页面”构成“块”。一个扇区包括有 X 个字节的数据段加上 Y 个字节的备用段，通常，一个扇区由 512 个字节（小块格式）或 2048 个字节（大块格式）构成一个数据段，由 16 个字节或 64 个字节构成一个备用段。部份多层单元（Multi-Level-Cell, MLC）闪存包括有 2048 字节和多于 64 字节的备用段，一扇区也可能有 4096 字节（或更多）数据字节。一个块由一组扇区构成，例如：16、32、64 或更多扇区，扇区数量根据具体情况确定。如果其中某个扇区包含一个或多个无效存储单元（即，编程或擦除操作过程中，一个或多个存储单元无法实现既定的最小操作状态），则认为这是一个“坏”块。在部份高可靠度的应用上，一个“坏”块定义能够被延伸到部份不

规则表征的任何的块，例如，检测到大量数据错误的位。如果一个块的全部存储单元都功能完好和有足够的可靠度，则认为这是一个“好”块。一般，储存区与一组实体块相关联的储存区共享一程序/读取页面缓冲内部快取。在大部份的例子中，拷贝回存（Copy back）指令可应用在移动相同储存区中两个块之间的数据。拷贝回存指令无法有效移动两个不同储存区之间的数据。

闪存设备可能一开始就有大量坏块（例如，10%）。此外，闪存设备原来的好块也可能在制造商规定的使用寿命之内变成坏块。这些坏块将在闪存设备的写入或擦除操作中表现出来。不幸的是，不断增加的坏块发生率极大地降低了闪存系统的性能。

大部分闪存系统（例如，电子数据闪存卡）同时使用多个闪存设备，并通过在多个闪存设备阵列中搜索可用好块的方法来解决坏块问题。存储在坏块中的有效数据（或分配给坏块的数据）需要重新分配或重新放置在一个或多个可用好块中。传统方法在系统全部闪存设备中搜索具有可用好扇区的可用好块，在搜索过程中，典型的重新分配/重新放置过程包括一个把数据传输外部缓冲区的过程（即，存储到闪存外）。如果找到了足够数量的可用好扇区，再把该数据写回这

些块中。

上述传统重新分配/重新放置过程可能会出现的一个问题，即如果一个或多个闪存设备达到容量极限（即，没有剩余可用好块），则闪存系统必须继续搜索其它闪存设备，直到搜索到具有足够数量的可用好扇区的可用好块。这可能导致外部缓冲区拥塞，从而导致闪存系统的总体性能的降低。

闪存设备中可用好块的数量随着闪存设备存储量越来越接近容量和过期（无效）块数量的增加而减少。过期块是指包含过期扇区的好块，过期扇区是指存储过有数据而该数据又被更新过的扇区。数据被更新后，过期数据仍保留在过期扇区中，被更新数据则被写入新扇区，该新扇区即成为包含有效数据的有效扇区。有效数据包括被更新数据和未被更新数据。相应地，过期块数量将随着文件修改或删除而增加。

过期块通常通过“垃圾回收”操作循环使用。在“垃圾回收”操作过程中，过期块将被擦除以便再次执行写入操作。过期块可能同时包含过期数据和有效数据。有效数据必须在过期块被擦除前复制到可用好块中。在垃圾回收操作中，过期块中的有效数据将复制到外部缓冲区，然后执行可用好块搜索程序，在全部闪存设备中搜索和定位可用好块。一旦可用好块搜

索成功，外部缓冲区中的有效数据将被复制回可用好块中。和重新分配/重新放置过程相同，传统的垃圾回收操作也可能发生外部缓冲区拥塞问题，导致闪存系统总体性能的降低。

另一个坏块解决方案为包括块替换过程的“损耗平衡”操作。在该操作中，有效数据将从一个块向另一个块转移，从而使数据分布更加均匀。然而，该操作同样包括外部缓冲区操作和多个设备中可用好块的搜索操作过程。如上所述，该操作也可能导致外部缓冲区拥塞，导致闪存系统总体性能的降低。

通常，传统方法没有解决在多个闪存设备中搜索可用好块所需的附加处理时间的问题，并且这些问题目前还没有一个很好的解决方案。已知的解决方案同样没有解决搜索过程中外部缓冲区潜在的拥塞问题。然而，这些限制都将影响坏块管理、垃圾回收和损耗平衡的效率。

因此，我们需要一套改进的闪存控制系统和方法。该系统和方法必须能解决处理坏块、垃圾回收和损耗平衡过程中可用好块搜索所需处理时间的问题。同时，该系统和方法必须简单可行、成本合理并且可以通过当前技术方便地实现。本发明的目的就是要解决该问题。

发明内容

本发明主要针对包含闪存设备的电子数据闪存卡、指纹传感器、输入/输出接口电路和处理单元。电子数据闪存卡可通过主机（外部计算机）存取，例如个人计算机，笔记本电脑或其它电子主机设备。由于电子数据闪存卡的便携性和耐用性，个人数据可以用加密形式存储在闪存设备中，这样只有通过诸如卡身上的指纹传感器的类的安全措施才能存取，确保不被未经授权人员误用。

本发明中，闪存控制器作为处理单元的一部分控制闪存设备的操作。处理单元与闪存设备和输入/输出接口电路相连接。闪存控制器逻辑包括闪存类型算法，用于检测该闪存设备是否为闪存逻辑所支持的闪存类型。通过将闪存检测算法代码动态部分连同机密数据存储在至少一个闪存设备中，不但电子数据闪存卡只读存储器（ROM）的大小可以降低，而且新的闪存类型无需硬件调整而仅需调整存储于闪存中闪存类型检测算法代码的动态部分即可被支持。这样既降低了总体成本又和消除了不必要的开发时间。

根据本发明另一个实施方案，电子数据闪存卡的处理单元可以通过选择工作在以下三个工作状态下：

可编程模式、数据读取模式、复位状态。当处理单元处于可编程模式时，处理单元启动输入/输出接口电路接收来自主机的机密数据文件，并把数据文件储存在闪存设备中。当处理单元处于数据读取模式时，处理单元启动输入/输出接口电路把数据文件传输到主机中。当处理单元处于数据复位状态时，数据文件（以及参考用指纹数据）将从该闪存设备中被擦除。

在本发明实施方案中，处理单元为一个微处理器，该微处理器可以是 8051、8052、80286、RISC、ARM、MIPS 或数字信号处理器（DSP）。

在本发明实施方案中，输入/输出（I/O）接口电路采用 USB 接口电路。

另一个在本发明实施方案中，USB 闪存设备和主机之间采用批量传输（BOT）协议高速传输数据。由于 BOT 传输的命令、数据、状态不但依赖于默认控制终端点还依赖于批量终端点，所以 BOT 协议是一种比 CBI 协议更加高效而且快速的传输协议。

在本发明另一个实施方案中，电子数据闪存卡（或其它闪存系统）包含带有处理器的闪存控制器，该处理器负责和系统连接的多个闪存设备的块管理操作。这些块管理操作包括坏块识别、过期块回收、损耗平衡操作。本发明中，处理器利用来自仲裁逻辑的数据，

在块管理操作中，把可用好块搜索限制在某个特定的闪存设备中，而不像传统方法那样在全部闪存设备中进行搜索。此外，在某个闪存设备中搜索可用好块时，处理器将利用该闪存设备的内部缓冲区储存有效数据。通过把可用好块搜索限制在一个特定闪存设备中，并使用指定闪存设备的内部缓冲区，本发明可减少可用好块搜索时间并消除对外部缓冲区使用的需要，从而可以避免外部缓冲区拥塞，提高传统闪存系统的性能。相应地，块管理操作的速度将有显著提高。

附图说明

为进一步说明本发明的具体技术内容，以下结合实施例及附图详细说明如后，其中：

图 1 为本发明一个实施方案的带有指纹识别能力的电子数据闪存卡方框图。

图 2 为本发明另一个实施方案的电子数据闪存卡电路原理方框图。

图 3 为本发明另一个实施方案的电子数据闪存卡采用的处理单元方框图。

图 4 为本发明另一个实施方案的电子数据闪存卡电路原理方框图。

图 5 为本发明另一个实施方案的闪存系统方框

图。

图 6 为本发明图 5 所示的闪存系统仲裁逻辑、暂存器文件、映像表之间接口的详细方框图。

图 7 为传统块管理操作方框图。

图 8 为本发明的块管理操作方框图。

图 9 为本发明闪存坏块方法管理高级流程图。

图 10 为本发明闪存设备的详细方框图，该详细方框图可用于实现图 6 和图 8 所示的闪存设备。

图 10 A 为本发明闪存设备的另一个实施例。

图 10 B 为本发明闪存设备的一实施例。

图 11 为本发明的数据存取方法流程图。

图 11 A 为本发明的数据存取方法一可选择的实施例流程图。

图 11 B 为本发明的数据存取方法另一可选择的实施例流程图。

图 12 为本发明的坏块替换方法流程图。

图 13 为本发明的垃圾回收操作方法流程图。

图 14 为本发明的损耗平衡操作方法流程图。

图 15 为本发明的损耗平衡操作方法一可选择的实施例的流程图。

具体实施方式

参看图1，根据本发明实施方案，电子数据闪存卡10可通过接口总线13、读卡器12或其它接口机制（图中未示）被外部计算机（主机）9存取，电子数据闪存卡10包括卡身1、处理单元2、一个或多个闪存设备3、指纹传感器4、输入/输出接口电路5、可选显示单元6、可选电源（例如，电池）7、可选功能键8。

闪存设备3安装于卡身1上，以已知方式储存了一个或多个数据文件、参考口令以及通过扫描电子数据闪存卡10的一个或多个授权用户指纹获得的参考指纹数据。只有授权用户能存取所储存的数据文件。该数据文件可以为图片文件或文本文件。

指纹传感器4安装于卡身1上，通过扫描电子数据闪存卡10的用户指纹产生指纹扫描数据。指纹传感器4可参考本发明人共同拥有的美国专利“带有指纹识别能力的集成电路卡”，专利号为6,547,130，本发明采用该专利的技术方案。上述指纹传感器专利包括一个扫描单元阵列，该阵列定义了指纹扫描区域。指纹扫描数据包括通过扫描相应的扫描单元阵列线而获得的大量扫描线数据。扫描单元阵列线按照行和列的方式进行扫描。各扫描单元若检测到持卡

人指纹脊则产生第一逻辑信号，若检测到指纹持卡人指纹谷则产生第二逻辑信号。

输入/输出接口电路 5 安装于卡身 1 上，启动后，接口总线 1 3 或读卡器 1 2 通过适当的插口建立和主机 9 之间的通讯。在本发明的一个实施方案中，输入/输出接口电路 5 包含一个 USB、PCMCIA 或 RS 2 3 2 接口电路和控制逻辑结构，可与连接至或安装于主机 9 的插口相连。在另一个实施方案中，输入/输出接口电路 5 包含 SD 接口电路、MMC 接口电路、CF 接口电路、MS 接口电路、PCI-Express 接口电路、IDE 接口电路、SATA 接口电路，这些电路通过接口总线 1 3 或读卡器 1 2 和主机 9 连接。

处理单元 2 安装于卡身 1 上，通过卡身 1 上的导电线路与与闪存设备 3、指纹传感器 4 和输入/输出接口电路 5 连接。在本发明的一个实施方案中，处理单元 2 可采用英特尔公司的 8 0 5 1、8 0 5 2 或 8 0 2 8 6 微处理器。在其它实施方案中，处理单元 2 采用 RISC、ARM、MIPS 或其它数字信号处理器（DSP）。本发明中，处理单元 2 至少受部分储存在闪存设备 3 中的程控，这样处理单元 2 可以通过选择在以下三种模式下工作：（1）可编程模式：在该模式下，处理单元 2 启动输入/输出接口电路 5 接收来自主机 9 的数

据文件和参考指纹数据，并把数据文件和参考指纹数据储存在闪存设备3中；（2）数据读取模式：在该模式下，处理单元2启动输入/输出接口电路5把存储于闪存设备3中的数据文件传输到主机9中；（3）数据复位状态：在该模式下，数据文件和参考指纹数据将从闪存设备3中擦除。操作中，主机9通过接口总线13或读卡器12和输入/输出接口电路5发送写入和读取请求到电子数据闪存卡10上的处理单元2，处理单元2通过闪存控制器（图中未示）从一个或多个闪存设备3读取或向其写入数据。在本发明的一个实施方案中，为了进一步确保安全，一旦检测到存储于闪存设备3的数据文件自上一次授权存取后超过了预设的时间极限，处理单元2将自动启动复位操作。

可选电源7安装于卡身1上，并与卡身1上的处理单元2和其它相关单元相连，为其提供电源。

可选功能键8安装于卡身1上，并与处理单元2连接，通过选择可使处理单元2工作于编程、数据读取或数据复位状态。功能键8可用于输入处理单元2的口令。处理单元2把输入的口令和存储在闪存设备3中的参考口令进行比较，若输入口令和参考口令一致，则启动电子数据闪存卡10的授权操作。

可选显示单元6安装于卡身1上，与处理单元2

相连并受其控制，用于显示与主机 9 之间交换的数据文件以及电子数据闪存卡 10 的操作状态。

以下为本发明的部分优点：首先，电子数据闪存卡体积小、容量大，所以数据传输方便；其次，由于指纹的唯一性，电子数据闪存卡只允许授权人员存取卡上储存的数据文件，从而提高了安全性。

本发明其它特点和优点如下。

图 2 为本发明优选实施方案的电子数据闪存卡 10 A 方框图，其中省略了指纹传感器和相关用户识别程序部分。为了降低集成成本，电子数据闪存卡 10 A 包括一个高度集成的处理单元 2 A、输入/输出接口电路 5 A 和闪存控制器 2 1。输入/输出接口电路 5 A 包括一个收发器模块，串行接口引擎块、数据缓冲器、暂存器和中断逻辑。输入/输出接口电路 5 A 和内部总线相连，使输入/输出接口电路 5 A 各个单元都能和闪存控制器 2 1 的各单元通讯。闪存控制器 2 1 包括一个微处理器单元、只读存储器（ROM）、RAM、闪存控制器逻辑、错误校正代码逻辑、通用输入/输出（GPIO）逻辑。在本发明的一个实施方案中，GPIO 逻辑和数个 LED 相连，用于状态指示，例如：电源良好，闪存读取/写入中等，或其它 I/O 设备相连。闪存控制器 2 1 和一个或多个闪存设备 3 相连。

在本实施方案中，主机 9 A 设有功能键 8 A，在电子数据闪存卡 1 0 A 操作过程中，通过接口总线或读卡器和处理单元 2 A 连接。功能键 8 A 用于选择电子数据闪存卡 1 0 A 的工作模式：编程、数据读取或数据复位状态。功能键 8 A 同时可用于向主机 9 A 输入口令。处理单元 2 A 把输入的口令和存储在闪存设备 3 中的参考口令进行比较，若输入口令和参考口令一致，则启动电子数据闪存卡 1 0 A 的授权操作。

同时在本实施方案中，主机 9 A 设有显示单元 6 A，在操作电子数据闪存卡 1 0 A 的过程中，通过接口总线或读卡器与处理单元 2 A 连接。单元 6 A 用于显示与主机 9 A 交换的数据文件以及电子数据闪存卡 1 0 A 的操作状态。

图 3 为处理单元 2 A 的详细框图。电子数据闪存卡 1 0 A 设有功率调节器 2 2，用于向一个或多个处理单元 2 A 供电。电源根据电子数据闪存卡 1 0 A 相关单元不同的要求提供不同电压的电源。电容器（图中未示）可用于提高电源稳定性。电子数据闪存卡 1 0 A 设有复位电路 2 3，用于向处理单元 2 A 提供复位信号。通电后，复位电路 2 3 向全部单元发送复位信号。当内部电压达到稳定水平后，撤销复位信号，由电阻器和电容器（图中未示）保证足够长的复位调整时间。电子

数据闪存卡 1 0 A 同时包括一个石英晶体振荡器（图中未示），向处理单元 2 A 内的 PLL 提供基频。在一些例子中，电子数据闪存卡 1 0 A 没有限制内部时脉需求，所以石英晶体振荡器和 PLL 可以被一较便宜的时脉产生器所取代，例如，RC 振荡器或环式振荡器，这些是适合整合在硅基板上的。

在本发明的实施方案中，输入/输出接口电路 5 A、复位电路 2 3、功率调节器 2 2 被集成或部分集成在处理单元 2 A 内。这种高集成度大大减少了所需的空间、复杂度和制造成本。

紧密性和成本对移动设备至关重要，例如这里所涉及的电子数据闪存卡。现代 IC 封装可以在一个 IC 封装中集成采用不同技术和材料的离散 IC 组件。例如，输入/输出接口电路为模拟和数字混合电路，可与处理单元集成到 MCP（多芯片封装）中。复位电路和功率调节器为模拟电路，也可与处理单元同时集成到 MCP（多芯片封装）中。

混合信号 IC 技术本质上允许同时集成模拟和数字电路。因此，更高度的集成技术可纳入处理单元 2 A 模具中，包括输入/输出接口电路 5 A、闪存控制器 2 1、复位电路 2 3 和功率调节器 2 2。

在优选的实现方案中，通过利用多芯片封装技术

或混合信号 IC 技术，实现处理单元 2、输入/输出接口电路 5、功率调节器 2 2 和复位电路 2 3 集成或部分集成。

闪存进步促使了各种类型闪存设备的出现，以满足对不同性能、成本和容量的需求。例如，对于相同的形状是数，多层单元（MLC）闪存设备比单层单元（SLC）闪存设备具有更高的容量。AND 或 Super-AND 闪存则被发明用于避开 NAND 闪存的知识产权问题。同时，大页面（2 K 字节）闪存比小页面（512 字节）闪存具有更好的写入性能。此外，闪存的快速发展使得设备的容量不断提高。为了支持这些不同类型的闪存，闪存控制器必须能够相应地进行检测和存取。

在本发明的思想和范围之内，可同时或独立实现上述各新颖特点。例如，图 4 为本发明另一个实施方案的电子数据闪存卡 1 0 C（或电子数据储存介质、集成电路卡）。根据上述一个或多个实施方案，电子数据闪存卡 1 0 C 可通过接口总线或读卡器（即通讯连接装置）被主机 9 A 存取，由卡身 1 C、处理单元 2 C 和一个或多个闪存设备 3 C 构成，其中处理单元 2 C 由闪存控制器 2 1 C 和输入/输出接口电路 5 C 构成。电子数据闪存卡 1 0 C 可以是上述电子数据闪存卡 1 0 A 的功能子系统，也可以是其它应用系统的功能子系统。

闪存设备 3 C 受闪存控制器 2 1 C 产生的命令控制，并在闪存设备中储存数据文件。

处理单元 2 C 通过上述输入/输出接口电路与闪存设备连接。处理单元 2 C 中的闪存控制器 2 1 C 通过一个或多个上述方法控制闪存设备 3 C。在本发明的一个实施方案中，闪存控制器 2 1 C 通过执行闪存类型算法（利用存储于只读存储器（ROM）（图中未示）的闪存控制器逻辑静态部分）确定闪存设备 3 C 是否被支持；如果是“新”的闪存类型，则读取存储于闪存设备 3 C 中的闪存控制器逻辑动态部分。

另一方面，启动输入/输出接口电路 5 C 后，通过接口连接装置和主机 9 A 建立 BOT 通讯。主机和闪存设备（以下也称为“USB 设备”）USB 接口电路之间有四种类型的 USB 软件通讯数据流：控制、中断、批量和同步。控制传输是主机通过控制管道向 USB 设备发送的数据流，用于向 USB 设备提供配置和控制信息。中断传输为小数据量、非周期性、固定等待时间、设备发起的通讯，通常用于通知主机 USB 设备所请求的服务。如果没有实时性要求，通过 USB 接口电路移动的大块数据可采用批量传输。同步传输在同步数据工作时使用。同步传输在主机和 USB 设备之间提供周期性和连续性的通讯。USB 接口电路通常支持两种数据传输

协议：CBI 协议和 BOT 协议。海量储存类 CBI 传输规范适用于全速软盘驱动，而不适用于高速设备或软盘驱动以外的其它设备（由 USB 规范确定）。在本发明实施方案中，aUSB 闪存设备和主机之间传输高速数据只采用 BOT 协议。由于 BOT 传输的命令、数据、状态同时传输到批量终端点和默认控制终端点，所以 BOT 是一种比 CBI 协议更加高效和快速的传输协议。

根据前述实现方案，当处理单元 2 C 可通过选择工作于可编程模式下时，处理单元 2 C 启动输入/输出接口电路 5 C 以接收来自主机 9 A 的数据文件，并根据主机 9 A 发送至闪存控制器 2 1 C 的写入命令，把数据文件储存到闪存设备 3 C 中；当处理单元 2 C 工作于数据读取模式下时，处理单元 2 C 根据主机 9 A 发送至闪存控制器 2 1 C 的读取命令和存取存储于闪存设备 3 C 的数据文件的命令接收闪存设备 3 C 中的数据，并启动输入/输出接口电路 5 C 把数据文件传输至主机 9 A；当处理单元 2 C 工作于数据复位状态下时，数据文件将从闪存设备 3 C 中被擦除。

本发明中智能处理单元 2 C 的优点包括：

（1）高集成度，大大减少了所需空间、复杂度和制造成本；

（2）通过把软件程序和数据保存在闪存中，降

低了控制器成本；

(3) 采用先进的闪存控制逻辑，提高了闪存的存取速度。

以下为本发明中所使用的术语定义。块：基本存储器擦除单元。各块包含一定数量的扇区，例如 1 6、3 2、6 4 等。如果某个扇区发生写入错误，则整个块即被确认为坏块，并且该块内其它全部有效扇区将被重新放置到另一个块中。扇区：块的子单元。典型的扇区由两个段构成—数据段和备用段。过期扇区：在编程过程中存储了数据，而该数据又随后被更新的扇区。数据被更新后，过期数据被保留在过期扇区中，而更新后的数据则被写入新扇区，新扇区即成为有效扇区。无效块：包含过期扇区的块。有效扇区：在编程过程中存储了数据并且该数据为当前（未过期）数据的扇区。损耗平衡：为了延长闪存的使用寿命而均匀分配各闪存块擦除次数的方法。闪存块只能承受有限次的擦除操作。例如，典型的 NAND 闪存的最大擦除次数为一百万次。备用块：闪存中的保留空间。备用块使闪存系统可以提前准备处理坏块。簇：为了改善存储性能，操作系统用作文件存取指针的多个数据扇区。在小容量记忆体操作中，一个簇通常由两个数据扇区构成，簇为最小的文件大小单元。典型的存储器

小块的簇大小为 1 k 字节（即各扇区大小为 5 1 2 字节），存储器大块的簇大小为 4 k 字节（即各扇区大小为 2, 1 1 2 字节）。FAT：保存指向文件地址指针的文件分配表。簇是 FAT 指向的最小单元。例如，FAT 1 6 是指簇的地址为 1 6 位。目录和子目录：操作系统所定义的文件指针。主引导记录（MBR）：用于储存根目录指针和相关引导文件（如果可引导）的固定位置。该固定位置为第一个块的最后一个扇区，如果第一个是坏块，则为第二个块的最后一个扇区。包：USB 基本事务单元的可变长度格式。一个常规的符合 USB 规范的事务通常由三个包—权标（Token）包、数据包和信号交换/交握（Handshake）包。权标包有 IN, OUT 和 SETUP 三种格式。数据包的大小可变，例如，USB 1.1 版为 6 4 字节，USB 2.0 版为 5 1 2 字节。信号交换/交握包有 ACK 或 NAK 格式，用于通知主机事务是否完成。画面（Frame）：批量事务处理，如果 USB 流量为低，则对占有画面具有高优先级。如果 USB 流量为高，批量事务也能等待后续画面。终端点：三个终端点包括控制、批量输入和批量输出。控制终端点用于系统初始枚举。批量输入终端点用作主机系统读取数据管道。批量输出终端点用作主机系统写入数据管道。命令块包（CBW）：一个命令块包包含一个命令块和相

关信息，例如数据传输长度（例如 5 1 2 字节，从第 8 字节至第 1 1 字节）。CBW 开始于包边界，结束于第 3 1 字节（字节 0 to 3 0），以小包形式传送。全部 CBW 传输都应按照从最低有效位（字节 0）向最高有效位的顺序排列。命令状态包（CSW）：CSW 开始于包边界。精简块命令（RBC）SCSI 协议：1 0 字节命令描述符。

图 5 为本发明另一个实施方案的电子数据闪存卡（存储器系统）1 0 0 方框图。闪存系统 1 0 0 通过接口转换逻辑 1 0 2 和主机系统 5 2 相连，接口转换逻辑 1 0 2 负责为微处理器 1 0 4 处理数据和时序校准。根据不同的应用环境，接口转换逻辑 1 0 2 兼容多种格式，例如 USB、PCI-Express、CF、SD、MMC、MS、IDE 等。主机系统 5 2 可以是个人计算机（PC）、数码相机、MP 3 播放器等。

微处理器 1 0 4 在闪存系统 1 0 0 内部执行读取、写入、擦除操作、块管理操作和其它内务操作。块管理操作包括复制和擦除操作，通过后台运行，即，对主机系统 5 2 隐蔽。只读存储器（ROM）1 0 6 用于储存微处理器 1 0 4 操作执行代码。

微处理器 1 0 4 通过仲裁逻辑 1 0 6 对闪存系统 1 0 0 的各闪存设备 1 1 0 a、1 1 0 b 和 1 1 0 c 分别执行块管理操作。仲裁逻辑可通过硬件逻辑或可编

程逻辑设备实现（例如，现场可编程门阵列（FPGAs））。

暂存器文件 1 1 2 为闪存设备 1 1 0 a-c 分配逻辑块地址。映像表 1 1 4 提供了闪存设备 1 1 0 a-c 相关的信息索引。这些信息包括逻辑块地址、设备编号、实体地址（Physical Base Address, PBA）、有效位和过期位。闪存接口控制器 1 1 6 与闪存 1 1 0 a-c 连接，执行来自处理器 1 0 4 的命令。这类命令包括读取、写入和擦除操作。

图 6 为本发明图 5 所示闪存系统 1 0 0 的仲裁逻辑 1 0 8、暂存器文件 1 1 2 和映像表 1 1 4 之间的详细方框图。主机系统 5 2 向闪存系统 1 0 0 发送逻辑块地址 3 0 2。逻辑块地址 3 0 2 包含一个扇区偏移地址。

暂存器文件 1 1 2 负责在各逻辑块地址 3 0 2 和特定的闪存设备编号之间建立联系。例如，若逻辑块地址在第一地址范围之内，则指向闪存设备编号 1 1 0 a，若逻辑块地址在第二地址范围之内，则指向闪存设备编号 1 1 0 b，以此类推。仲裁逻辑 1 0 8 发送逻辑块地址和相关设备编号至映像表 1 1 4。各闪存设备地址容量预先编程保存在相应的暂存器 1 1 4 a-1 1 4 c 中供逻辑块地址指派。一旦各特定的闪存设备的实体地址（PBA）被仲裁逻辑 1 0 8 确定，全部读取 /

写入操作都将在设备内部被执行。

映像表 1 1 4 把各逻辑块地址翻译成相应的实体地址，并把设备编号和实体地址输出至闪存接口控制器 1 1 6。索引映像表 1 1 4 由一个或多个查询表（LUT）构成，可通过易失性随机存取存储器（RAM）实现，例如静态 RAM（SRAM）。在本发明的一个实现方案中，闪存设备 1 1 0 a、1 1 0 b 和 1 1 0 c 分别对应映像表 1 1 4 a、1 1 4 b 和 1 1 4 c。例如，表 1 1 4 a 储存闪存设备 1 1 0 a 的实体块地址信息，表 1 1 4 b 储存闪存设备 1 1 0 b 的实体块地址信息，而表 1 1 4 c 则储存闪存设备 1 1 0 c 的实体块地址信息。映像表 1 1 4 把逻辑块地址翻译成由仲裁逻辑 1 0 8 提供的特定设备编号实体地址。映像表 1 1 4 同时还提供有效位的值。有效位的值在通电过程中被置零。通电后，闪存 1 1 0 首先被扫描以重建映像表 1 1 4。

在初始化过程中，特定扇区最大量的擦除操作被编程写入暂存器文件 1 1 2 用于地址仲裁。闪存接口控制器 1 1 6 向特定的闪存设备 1 1 0 发送一系列时序信号，执行闪存设备 1 1 0 块管理相关的写入和擦除操作。

由于闪存设备 1 1 0 a-c 的各暂存器可以独立编程，闪存系统 1 0 0 的各个闪存设备 1 1 0 a-c 可以具

有不同的数据储存容量。甚至可以采用不同品牌的闪存设备。这种灵活性降低了总体制造成本。在各闪存设备 1 1 0 a-c 内部的页面大小，即各实体扇区的字节数应相同（例如，5 1 2 字节或 2 1 1 2 字节）。

图 7 为传统块管理操作方框图。从图中可以看出，有效数据存储于闪存设备 A 中块 4 0 4 的扇区 4 0 2 a、4 0 2 b 和 4 0 2 c 中。在一个块管理操作（例如，坏块替换、过期块回收或损耗平衡）过程中，如果扇区 4 0 2 a，4 0 2 b 和 4 0 2 c 变成坏块、过期或需要进行损耗平衡，存储在扇区 4 0 2 a、4 0 2 b 和 4 0 2 c 中的有效数据将被重新放置。块管理操作通常被称为内务操作。这些操作通常以后台方式完成，以方便后续写入操作。本例中的块管理操作为坏块替换操作。

从图中可以看出，有效数据首先被重新放置（即复制）到外部缓冲区 4 0 6 中。然后在其它闪存设备中执行可用好块（即功能块）搜索程序。传统的闪存系统把多个闪存设备当作统一的全局单元处理。相应地，全部闪存设备的块都按照全局地址方案分配地址，这样，全部闪存设备都将进行搜索。如果在另一个闪存设备 B 的块 4 0 8 中搜索到可用好扇区，则把有效数据复制到闪存设备 B 的好扇区中。在其它块管理操作中，外部缓冲区 4 0 6 起相同的作用。

图 8 是为本发明的块管理操作方框图。从图中可以看出，有效数据存储 在闪存设备 5 0 4 中的块 5 0 3 的扇区 5 0 2 a、5 0 2 b 和 5 0 2 c 中。在块管理操作过程中，存储在扇区 5 0 2 a、5 0 2 b 和 5 0 2 c 的有效数据将被重新放置。

在本发明中，各闪存设备执行独立的块管理操作，并在各闪存设备边界内部执行。换句话说，在块管理操作过程中，存储在闪存设备 1 1 0 a 中的有效数据被重新放置在同一个闪存设备内部的最佳位置（即，数据不是转移到其它闪存设备 1 1 0 b 或 1 1 0 c 中）。传统块管理操作把有效数据重新放置在不同的闪存设备中，与传统块管理操作相比，本发明通过把有效数据重新放置操作限制在同一闪存设备之内，提高了总体系统的性能。传统块管理操作存在的问题是可用好块的搜索范围太大，需要搜索多个闪存设备。此外，传统块管理操作要求使用外部缓冲区（例如，图 7 所示的页面缓冲区 4 0 6）。把有效数据从一个闪存设备传输到外部缓冲区，然后再传输到另一个闪存设备将增加总体操作时间。本发明通过对各闪存设备执行独立的块管理操作来解决该问题，这样块管理操作可以在各闪存设备的内部执行。

本发明的另一个优点是块管理操作可以在多个闪

存设备内部同步发生，从而进一步提高了闪存系统的性能。例如，第一个数据重新放置操作可以在第一个闪存设备内执行，同时第二个数据重新放置操作在第二个闪存设备内执行。同时也使不同闪存设备可同步擦除和编程。系统并行性的提高显著提高了闪存系统的性能。

本例中的块管理操作为坏块替换操作。如果扇区 5 0 2 a、5 0 2 b 和 5 0 2 c 变成坏扇区，则存储在扇区 5 0 2 a，5 0 2 b 和 5 0 2 c 的有效数据将被重新放置。在本实施方案中，有效数据将被复制到内部缓冲区 5 0 6 中。内部缓冲区 5 0 6 是在闪存设备 5 0 4 内部功能正常的易失性存储器扇区。在本发明的一个实现方案中，易失性存储器扇区的块被预留用于向各闪存设备提供内部缓冲区 5 0 6，这样可以减少对外部搜索的需求（即在闪存设备的边界的外的搜索）。可用好块搜索在闪存设备 5 0 4 内部其它部分中执行。各闪存设备的全部块地址都根据分布式地址方案分配，如图 8 所示，同时全部搜索都被限制在闪存设备的边界之内。

搜索到好块之后，有效数据将被复制到一个或多个好块中。无论哪种情况，相比于图 7 所示的传统多芯片搜索方法，单芯片搜索大大降低了搜索时间。同

时，由于坏块操作发生在闪存设备 5 0 4 内部，有效数据重新放置时间大大减少。本发明消除了对外部缓冲区的需求。相应地，由于无需把有效数据转移到外部闪存设备 5 0 4，重新放置时间也大大减少。

本发明的其它块管理操作过程中，内部缓冲区 5 0 6 也起相同的作用。另一个方案时，在块管理操作过程中，有效数据无需先被存储到内部缓冲区 5 0 6 中而被直接重新放置到好扇区中。

本发明中，各闪存设备作为独立寻址单元工作，并且块管理操作在各闪存设备边界之内发生。相应地，本发明的另一个优点是在同一个闪存系统之内可以使用不同容量的闪存设备。

图 9 是本发明的闪存块管理方法高级流程图。首先，在具有数个闪存设备的闪存系统中启动至少一项操作，参考步骤 6 0 2。在本实施方案中为块管理操作。下一步，在闪存设备之内搜索目标块，参考步骤 6 0 4。下一步，有效数据在闪存设备内部从源块向目的块被重新放置，参考步骤 6 0 6。相应地，多个闪存设备可同时执行块管理操作。在多个闪存设备执行块管理操作过程中，各闪存设备都在各自内部执行独立的块管理操作。各闪存设备都能进行独立、同步的块管理操。

图 1 0 为本发明的闪存设备 7 0 0 的详细方框图，该方框图可用于实现图 5 所示的闪存设备。闪存设备 7 0 0 被分配了一个逻辑块地址范围，逻辑块地址的范围由图 5 所示的仲裁逻辑 1 0 8 决定。闪存设备 7 0 0 的实体块地址从零开始一直增加到闪存设备 7 0 0 的最大容量。图中画出了示例块 7 0 6 a、7 0 6 b、7 0 6 c 和 7 0 6 d。7 0 6 a 和 7 0 6 b 分别为第一和第二个实体地址块而 7 0 6 d 为最后一个实体地址块。

在本实施方案中，扇区（通常称为页面）7 0 1 包含 5 2 8 个字节，其中包括 5 1 2 个字节的的数据段和 1 6 个字节的备用段。闪存设备 7 0 0 的数据结构由实体地址 7 0 6 a、7 0 6 b、7 0 6 c 和 7 0 6 d 的数据段 7 0 2 和备用段 7 0 4 构成。各段都有一定的字节数，该字节数由具体应用决定。例如，数据段可以包含 5 1 2 字节、2 1 1 2 字节或更多字节，备用段可以包含 1 6、6 4 或更多字节。

数据段 7 0 2 储存原始数据，而备用段 7 0 4 储存存储器管理相关信息。备用段 7 0 4 包括有效扇区段 7 1 0、过期扇区段 7 1 2、坏块指示段 7 1 4、擦除计数段 7 1 6、错误校正代码（ECC）段 7 1 8 和逻辑块地址扇区地址段 7 2 0。有效扇区段 7 1 0 用

于指示是否扇区中的数据可被有效读取。过期扇区段 7 1 2 为标志段，用于指示扇区中的数据是否过期标志。若发生了后续的写入或擦除操作，则过期标志可以被修改。坏块指示段 7 1 4 用于指示坏块。在本发明的一个实施方案中，把 0 值存储于坏块指示段 7 1 4 用于指示该块已被损坏。若尝试写入特定扇区或擦除特定块失败，则认为该块为坏块。在具体的实现方案中，坏块指示由工厂设置确定。闪存系统的固件通过扫描各块的第一扇区来确定数据是否能被存取。完整的扫描信息随后被保存到各闪存设备最后的块中。

在本实施方案中，用两个字节（16 位）来记录各块（16 扇区）的坏块信息。用两个扇区（1024 字节）记录闪存设备（最多 512 个块）的坏块信息。为了达到更高的可靠性，储存了八份完全相同的坏块信息备份，以避免在标志记录过程中发生扇区变坏事故。为了实现快速存取，这些八份备份都存储在各闪存设备的最后一个块位置 7 0 6 d 中。坏扇区指示段 7 1 4 被保存在最后块 7 0 6 d 中，以便更方便地被闪存系统的固件读取。

然而，坏块记录格式应该被良好地组织，每一个块有两个字节（例如：16 位）是适合小的闪存芯片。在先前的方案中，512 个块总共有 1024 个字节，

以及如果每一个块有 1 2 8 K 字节，那么闪存芯片大小就只有 6 4 M 字节。在部份方案中，闪存芯片的大小可能到达 1 G 字节或者更多。在一个拥有 1 G 字节的 MLC 闪存方案中有 4 K 个块，其有 1 2 8 页面/块。这是意指坏块记录每一个块需要 1 2 8 位，和在保留闪存块总共有 6 4 K 字节。这个大小对于搜寻是不够的，且实际上控制器是无法提供高达 6 4 K 字节这样大小的暂存器和 SRAM 给表格。因此，在一些情况下，是以一个块有 1 6 位作为例子，对于像大部份的 MLC 闪存芯片般的大闪存尺寸，仅以一位或一字节记录坏块而不使用扇区/页面指示是为较好而且简单的。例如，如果一个块被发现是坏的（不论是那一个扇区是坏的），这一个块使用一个字节记录为坏块（例如：“0 xff”代表好块，“not 0 xff”代表坏块）。所以，对于 1 GB 闪存，记录的大小仅仅是 4 K 字节。而且，如果 1 位被用于一个块，一个较小的记录是有可能的，亦即 4 K 字节。本发明以下的描述假设每一个块为 1 6 位当作例子以简单化描述。

擦除计数段 7 1 6 用于记录一个块将在该闪存设备整个使用寿命内被擦除的次数。擦除计数段 7 1 6 包含三个字节，最多可记录 1 6 0 0 万次块擦除操作。ECC 段 7 1 8 包含六个 ECC 字节，以确保数据一致性。

众所周知，EEC 是一种非常先进的方法，用于错误检测和校正。逻辑块地址扇区段 7 2 0 用于电源备份和系统重入。由于闪存系统的映像表存储在易失性存储器中，因此在掉电过程中，无法保存有效扇区的信息。在系统初始化和停电时，逻辑块地址扇区地址段 7 2 0 用于重构映像表。逻辑块地址扇区地址段 7 2 0 记录了先前的写入操作、有效扇区和过期扇区信息以便重构映像表。设定了新的数据结构后，闪存系统的固件可以修复悬空簇。这可以通过校核存储在闪存设备中闪存阵列的 FAT 表实现。

图 1 0 A 为较详细的闪存设备 7 0 0 的方块图，其被用于实现图 5 的依据本发明的闪存设备。

在这个特定实施例中，一个扇区（一般而言为一页面）7 0 1 有 2 1 1 2 字节，包括有 2 0 4 8 字节的数据和 6 4 字节的备用段。闪存设备 7 0 0 有一数据结构，其包括有四个数据段 7 0 3 a/7 0 3 b/7 0 3 c/7 0 3 d 和其相对应的备用段 7 0 3 as/7 0 3 bs/7 0 3 cs/7 0 3 ds。数据段 7 0 3 a/7 0 3 b/7 0 3 c/7 0 3 d 各个都有 5 1 2 数据字节，备用段 7 0 3 as/7 0 3 bs/7 0 3 cs/7 0 3 ds 各个都有 Y 备用字节。在这个例子中 Y 是 1 6，如果闪存能够提供更多备用字节 Y 也可以是其它的值。

备用段 7 0 3 as/ 7 0 3 bs/ 7 0 3 cs/ 7 0 3 ds 各个都有 ECC 段 7 3 1，LBA 扇区地址 7 3 2，和同位 7 3 3。对于 MLC 闪存需要强大的 ECC 保护。ECC 段 7 3 1 可能占据 1 2 字节或更多（例如，在此假设为 1 3 . 5 字节）以支持在每一块 5 1 2 字节数据上 1 2 位错误检测/修正能力。LBA 扇区地址 7 3 2 如同图 1 0 中 7 2 0 的定义（假设为 3 字节）。同位 7 3 3 有一些位被用于检测在 LBA 扇区地址 7 3 2 段是否有任何位错误，同位 7 3 3 可以是任何长度，在此例中为 4 位。

在图 1 0 A 中，备用段 7 0 3 as/ 7 0 3 bs/ 7 0 3 cs/ 7 0 3 ds 在 LBA 扇区地址区域 7 3 2 中全部都具有相同的值。所以，在一个扇区中总共有四个拷贝的 LBA 扇区地址。根据检测同位 7 3 3，一旦在 7 0 3 as 检测到位错误，闪存控制器可存取其它代替的拷贝。因此，LBA 扇区地址信息提供非常高的可靠度。

相较于图 1 0，在图 1 0 中没有明确的坏块标志段。此结构的好处为给予 ECC 区域尽可能的多个位，因此，提供较高的数据可靠度，特别是在 MCL 闪存芯片的高位错误事件，此强大的 ECC 保护非常的重要。在图 1 0 A 中，坏块信息可能归因于实际上 LBA 扇区地址的四个拷贝已被更改。例如，如果一个控制器试着标记一个块为坏块，仅仅只写入随机值或能被识别为

坏块的特殊形态在四个 LBA 扇区地址区域 7 3 2 和同位位 7 3 3，如此，即使不需使用到图 1 0 中的标志描述图 1 0 A 中的坏块也能够被认定。

图 1 0 中标示记录坏块信息的保留块，在图 1 0 A 并没有标示，但是图 1 0 B 和图 1 0 A 相结合后，即能显示保留块的记录方法。在图 1 0 B 中，第一个实体块 7 4 0 和最后一个实体块 7 4 0 a 都是记录坏块信息的保留块。第一块和最后一块的格式是相同的（在图 1 0 B 中，仅第一块详细的标示）。在保留块中，一些扇区（或页面）7 4 1 为控制器的韧体码，应用信息（例如，SD 卡的 CSD/CID 信息）等的保留区。扇区（页面）群组 7 4 2 用于储存坏块信息（相同于图 1 0 中的 7 0 6 d）。扇区群组 7 4 3 从 7 4 2 拷贝过来，所以，总共有 4 个坏块信息拷贝在第一块 7 4 0 和最后一块 7 4 0 a，在一些实施例中有可能保留更多的拷贝。

在图 1 0 B 中，第一块 7 4 0 用于储存坏块信息和韧体码以及应用信息。考量最后一块本身可能为坏块和第一块一般为闪存供应者保证为好块的可能性，在图 1 0 B 中的实施例大致上有比图 1 0 较高的可靠度。

图 1 1 为本发明数据存取方法流程图。闪存系统初始化后，通过查询闪存阵列识别号（ID）确定闪存

设备中闪存阵列的容量，参考步骤 8 0 2。同时，扫描各闪存设备的实体地址以确定现有坏扇区（步骤 8 0 2）。该判断可以通过读取坏块指示段实现。

逻辑块地址的范围通过编程写入闪存控制器的暂存器文件，参考步骤 8 0 4。给定一个闪存设备，由于该闪存设备中部分空间被预留用于坏块替换，所以实体块地址的范围比逻辑块地址的范围大。例如，把闪存阵列的 1 0 % 用作保留空间是比较合理的。

下一步，从主机系统接收逻辑块地址扇区地址、数据和命令，参考步骤 8 0 6。采用簇数据缓冲和写入后高速缓冲策略以提高闪存系统的性能。下一步，由映像表确定闪存设备编号和实体地址，参考步骤 8 0 8。下一步，分析来自主机系统的命令，参考步骤 8 1 0。如果该命令是读取命令，则执行读取操作，参考步骤 8 1 2。然后，校核读取操作获得的数据，参考步骤 8 1 4。利用 ECC 段字节中保存的信息校核数据。如果数据校核正确，则把数据返还主机系统，参考步骤 8 1 6。如果读取操作得到的数据不正确，则执行 EEC 操作来校正数据，参考步骤 8 1 8。

如果该命令为写入命令（参考步骤 8 1 0 为“是”），则执行写入操作。写入操作所需要的时间明显比执行读取操作所需要的时间长。例如，写入操作

所需的时间可能比读取操作所需时间的长 2 0 倍。首先，检查自由扇区（即，可用扇区），参考步骤 8 2 0。如果自由扇区的数量低于自由扇区的阈值，则通过垃圾回收操作回收块，参考步骤 8 2 2。

如果自由扇区的数量不低于自由扇区的阈值，则把数据写入该闪存设备，参考步骤 8 2 4。写入操作完成后，即可确定写入操作是成功还是失败，参考步骤 8 2 6。如果写入操作成功，则结束写入操作，参考步骤 8 2 8。如果写入操作失败，意味着该块是坏块，则执行坏块操作，参考步骤 8 3 0。

通常，若某个块是坏块，则该扇区中的数据不可靠。即使一个块中只有一个扇区为坏扇区，这个块将被确定为坏块。为了确保数据的可靠性，数据将不再赋值到坏块中，而是重新分配至好块中。相应地，储存在坏块中有效扇区的数据将被转移到好块中供进一步参考。该操作称为坏块替换。闪存设备将在内部发出回拷命令用于减少事务处理时间。对于没有支持拷贝回存指令的闪存，控制器必需读取信息至本身的缓冲存储器以便将数据编程到欲达成的好块目标区。

一些应用可能需要高的可靠度，例如，固态硬盘（SSD）支持窗口操作系统（Windows Operation System）。因此，即使在 SSD 中有一个位错误也可能造

成致命的系统错误。然而，MLC 闪存比 SLC 闪存高的位错误率，典型为高于 1 0 0 0 次以上，所以，位错误率模块显示即使当编程成功，在数以千计的读取操作后，位错误也可能增加。这是意指新的坏块不仅发生在编程和 / 或擦除期间，也发生在仅读取操作期间。在 SLC 闪存中，遇到高位率的可能性非常非常的低，但是在 MLC 闪存中不可能被忽视。此技术于图 1 1 A 和图 1 1 B 中提出说明。

在图 1 1 中，步骤 8 1 2 为读取操作，和步骤 8 1 4 利用 ECC 算法检查是否有任何位错误。如果发现任何位错误，控制器以 ECC 算法的结果来修正其错误和结束读取操作 8 1 6。然而，在多次的编程、擦除、和读取后如果有块或扇区变得不稳定，其位错误可能性可能逐渐地增加和最后增加到控制器 ECC 算法无法提供的上限。例如，在 5 1 2 字节数据单元中，具有 4 位 ECC 的控制器修正的能力不可能到 5 位错误。在 MLC 闪存中，四位错误率是足以被检测，所以大部份的 MLC 闪存需要 8 位 ECC 保护或更多。在此，坏块观念被扩大到包括位错误被发现已达到某种严重程度的块，例如，超出控制器的修正能力。

在图 1 1 A 中，步骤 8 1 4 检查是否有任何位错误，如果有错误在步骤 8 1 4 a 中被查测到，控制器检

查是否位错误数低于临界初始定义（可能基于设计者的经验）。如果位错误数低于此临界 VT ，于是控制器修正其错误的位（步骤 8 1 8），读取操作成功的完成。如果位错误率达到临界 VT ，控制器将此块视为不稳定并在未来读取操作可能变成较差的状态以及未来造成数据遗失。所以，控制器视此块为潜在的坏块并移动所有有效数据扇区到另一个好块（步骤 8 1 9 a 和 8 1 9 b）。最后，控制器擦除来源块成空的块。一般而言，当块被发现达到危险位错误率，在原来的块的数据变成不可修正的前，控制器会移动数据到安全且好的块。在此， VT 可被定义为低于 ECC 能力上限一个或两个位。例如，如果 ECC 修正能力为 8 位， VT 可被设定为 6 位。

理论上，图 1 1 B 相同于图 1 1 A，但是其视原来的块为曾经有高位错误率被发现（步骤 1 1 5 e）的块而非经擦除为可再使用的空的块（步骤 8 1 5 c）。在擦除之后，如果此块的位错误率能够被降低，对于数据而言，此方法为较安全的，但是可能会浪费一些块。

图 1 1 A 和图 1 1 B 为 MLC 闪存在高可靠度应用的两个增强的模式，基于系统不同的应用/需求使用 MLC 闪存，控制器可选择其中的一的方法。

图 1 2 为本发明的坏块替换方法流程图。坏块中坏扇区的位置被记录在闪存设备的最后两个块的保留

区中，参考步骤 9 0 2。在本发明的一个实施方案中，各块都由 1 6 个扇区组成。另有十六个位分别与着十六个扇区相关联。这些位用于指示坏扇区。相应地，如果某个位为 0，则表明相关的扇区为坏扇区，而整个块即被确认为坏块。这些位的编程方式是通过读取整个扇区的值，然后把原值和位值相加，最后写入来实现。为了确保正确性，固件将生成四个备份。全部八个块（4 x 2 块）都位于各闪存设备的最后空间中。在扇区的使用寿命期限内，每个位将被设置一次，用于指示坏扇区的位置。

下一步，确定是否有写入命令，参考步骤 9 0 4。如果读取操作失败且没有写入命令，坏块扇区中的有效数据即被确认，参考步骤 9 0 6。如果有写入命令（参考步骤 9 0 4 为“是”），则在同一个设备之内执行可用扇区搜索操作，参考步骤 9 0 8。如果没有足够数量的可用扇区，则执行垃圾回收操作，参考步骤 9 1 0，直到有足够数量的可用扇区。如果有足够数量的可用扇区，映像表的逻辑块地址将被更新，参考步骤 9 1 2。下一步，写入操作完成，参考步骤 9 1 4。写入操作完成后，坏块中扇区的有效数据被确定，参考步骤 9 0 6。若写入或擦除操作失败，则执行该操作。

下一步，确定好块的目的扇区，用于重新分配坏块的有效数据，参考步骤 9 1 6。下一步，有效数据被重新放置（即，复制）到目标扇区中，参考步骤 9 1 8。在重新放置过程中，在闪存设备内部执行回拷操作，以避免外部通信并提高闪存系统的性能。下一步，闪存控制器中的映像表被更新以反映变化供以后数据存取用，参考步骤 9 2 0。下一步，确定是否全部来自坏块的有效数据都已经被转移到好块中，参考步骤 9 2 2。如果没有，操作循环回到块 9 0 6。如果全部有效数据都已经被转移，则坏块替换操作结束。

图 1 3 为本发明的垃圾回收操作方法流程图。垃圾回收操作是在各闪存设备边界之内执行的单芯片操作。相应地，多个垃圾回收操作可以同步在各闪存设备内部发生。首先，在闪存设备内部进行搜索，以定位含有最多过期扇区的过期块，参考步骤 1 0 0 2。特别地，固件通过扫描整个过期扇区段以确定各块过期扇区的数量。搜索结果存储在暂存器中。暂存器用于指示含有最多过期扇区的过期块。例如，用四个暂存器指示含有最多过期扇区的四个过期块。扫描结果和逻辑块地址的值被同时保存用于更新地址映像表。同时，不同的暂存器组被同时用于记录该设备中含有最多过期扇区的四个有效块。目的是在把有效扇区重

新放置到目标（有效）块中之后擦除一个过期（源）块。这四个暂存器组用于提供最匹配的源块和目标选择。

下一步，确定源块中有效扇区的数量，参考步骤 1 0 0 4。下一步，确定有效扇区的地址，参考步骤 1 0 0 6。这些地址指向的有效扇区被称为目的扇区。下一步，执行回拷操作把有效数据从过期（源）块向目标扇区进行复制，参考步骤 1 0 0 8。在回拷操作中，有效数据可以临时存储在内部缓冲区中。

下一步，确定在垃圾回收操作中是否出现了坏扇区，参考步骤 1 0 1 0。如果出现了坏扇区，则执行坏块重新放置操作，参考步骤 1 0 1 2。如果没有出现坏扇区，源块（含有最多过期扇区的块）将被擦除并且这个块各个位的值将被设为 1，参考步骤 1 0 1 4。下一步，确定擦除操作是否失败，参考步骤 1 0 1 6。如果擦除操作失败，则执行坏块重新放置操作，参考步骤 1 0 1 8。如果擦除操作没有失败，则过期块的垃圾回收操作完成，映像表被更新以反映修改供后续写入操作用，参考步骤 1 0 2 0。下一步，擦除计数段中被擦除块的擦除计数增加，参考步骤 1 0 2 2。

图 1 4 为本发明的损耗平衡操作方法流程图。在

没有来自主机系统的数据传输请求时，损耗平衡操作将后台运行。损耗平衡操作过程中有可能会收到来自主机系统的数据传输请求。损耗平衡操作通常把有效数据从低擦除计数的块移向高擦除计数的块。低擦除计数的块将被擦除，其擦除计数将增加。这一操作过程可以通过降低块的最高擦除计数使其接近平均设备擦除计数的方法来平衡块的擦除计数。该过程同时推迟了给定块达到其最大擦除计数的时间。

首先，固件读取各块擦除计数段中的擦除计数，并确定各闪存设备的平均设备擦除计数，参考步骤 1 1 0 2。然后确定全部闪存设备的平均全局擦除计数，同样参考步骤 1 1 0 2。平均设备擦除计数值存储在各闪存设备的暂存器中，参考步骤 1 1 0 4。对于各闪存设备，如果平均设备擦除计数大于设备阈值计数暂存器的值，则设备阈值计数将被更新为该值。同理，如果平均全局擦除计数大于全局阈值计数暂存器的值，全局阈值计数将被更新为该值。这些值今后都将被用到，参考步骤 1 1 0 4。在本发明的一个实现方案中，指定了三个暂存器用于保存各闪存设备的擦除计数值。第一个暂存器储存特定的闪存设备的平均擦除计数，称为设备擦除计数。第二个暂存器储存特定的闪存设备的平均擦除计数，称为设备阈值计数。第

三个暂存器储存全部闪存设备的平均擦除计数值，称为全局阈值计数。例如，设备阈值计数可设为 5，0 0 0，全局阈值计数可设为 2 0，0 0 0。这两个值同时被预编程成为闪存系统初始化的一部分。

下一步，确定闪存设备的设备阈值计数是否大于或等于全局阈值计数，参考步骤 1 1 0 6。如果不是，确定该设备的设备擦除计数是否大于或等于设备阈值计数，参考步骤 1 1 0 8。如果不是，损耗平衡操作结束。如果是，确定设备中具有最高擦除计数的块，参考步骤 1 1 1 0。下一步，确定设备中具有最低擦除计数块，参考步骤 1 1 1 2。下一步，具有最低擦除计数的块中的有效数据被重新放置到另一个块中，参考步骤 1 1 1 4。下一步，具有最低擦除计数的块被擦除，同时其擦除计数增加，参考步骤 1 1 1 6。下一步，具有最高擦除计数的块中的有效数据被重新放置到具有最低擦除计数的块中，参考步骤 1 1 1 8。下一步，更新映像表，参考步骤 1 1 2 0。下一步，设备阈值计数增加，参考步骤 1 1 2 2。损耗平衡操作结束。

一个闪存设备的块管理操作可能同时牵涉到多个闪存设备，此时有效数据通过外部从一个闪存设备向另一个设备被重新放置。这从本质上提高闪存设备系

统的总体性能。在另一个特定的实施方案中，如果某个特定的闪存设备进行块管理操作，其擦除计数高于其它闪存设备，有效数据可通过外部从一个闪存设备向另一个执行重新放置操作，以达到不同闪存设备之间的平衡。在本发明中，既包括内部重新放置又包括外部重新放置。

返回步骤 1 1 0 6，如果闪存设备的设备阈值计数大于或等于全局阈值计数，则确定设备中具有最高擦除计数的块，参考步骤 1 1 2 8。下一步，确定具有最低平均擦除计数的闪存设备，参考步骤 1 1 3 0。下一步，确定设备中具有最低擦除计数的块，参考步骤 1 1 3 2。下一步，具有最低擦除计数的块中的有效数据被重新放置到另一个块中，参考步骤 1 1 3 4。下一步，具有最低擦除计数的块被擦除，其擦除计数增加，参考步骤 1 1 3 6。下一步，具有最高擦除计数的块中的有效数据被重新放置到具有最低擦除计数的块中，参考步骤 1 1 3 8。在本发明的一个实施方案中，有效数据被移动到另一个闪存设备中。下一步，更新映像表，参考步骤 1 1 4 0。下一步，全局阈值计数增加，参考步骤 1 1 4 2。损耗平衡操作结束。

上述描述的损耗平衡算法的实施例是简单和有效

的。然而，在一些例子中，例如，MLC 闪存应用，因为其有一些缺点，此算法可能被改良。通常，MLC 闪存在备用区域需要尽可能多的位以提供强大的 ECC 保护，所以，在备用区域的擦除计数位能够被移除以储存较多的 ECC 位是比较好的。上述实施例的另一个问题是在被擦除之后，擦除计数位应该被写回块，这指的是在被擦除之后，闪存块的数据全是“空的”，但是备用区域却被编程。就大部份的 MLC 闪存的限制因素而言，在被擦除和多次编程之后，扇区仅只能够编程一次，可能在这扇区中造成不明确的内容。在这个例子中，可能在每一个块中持续追踪擦除计数是不切实际的。因此，图 1 5 说明损耗平衡方法的另一个实施例以解决上述问题。

在图 1 5 中，当接收到一写入指令，控制器将利用映像表将接收到的 LBA 转换成 PBA (步骤 1 2 0 1)。如果接收到的数据能够直接写入当前的 PBA 的扇区，控制器执行编程操作至 PBA。如果编程失败，当前的 PBA 被示为坏块且控制器移动有效扇区到闲置块 (目标块)，其闲置块为从所有的闲置块中随意选取的。如果接收的数据无法直接写入当前的块，控制器从所有的闲置块中随意寻找一个闲置块当作目标块将数据写入和执行相对应的编程。在数据被写入或拷贝到目标块

之后，如果原来的块被示为坏的则原来的块会被擦除或表示为坏块。最后，控制器修改映像表和结束操作。因为每一次 LBA 被安排至新的闲置块，替代总是连结至某些实体块，LBA 至 BPA 的关系是随机选取闲置块以不断变化地改写。所以，理论上，每一个 LBA 有相同的机会被储存在任何的实体块。反的，每一个实体块可被选取以接受 LBA 的数据，但是下一次，会被改变成接受其它 LBA 的数据。这个程序是随机的。如此，每一个实体块有相同的机率被编程或擦除，因为其对应的 LBA 是随机改变的。然而，使用这个方法没有明确的标示或计数器告知擦除的次数和可能无法知道最多擦除次数在那一个块，在控制擦除次数时并不准确。但是，这个方法比较简单。从闲置块随机选取，大致上此损耗平衡方法的实施例是有效以允许每一个实体块有相同的擦除机率。

本发明中的闪存控制器可以执行多块数据存取。传统闪存设备内置一个 512 字节的页面暂存器。写入闪存设备的数据首先要写入该页面暂存器，然后才能写入闪存阵列。传统的闪存控制器及其固件控制着闪存系统存取周期。传统闪存控制器每次只能传输一个块（512 字节）的数据到闪存设备的页面暂存器中。如果 512 个字节的页面暂存器已被写入，则不

能对该闪存设备进行其它存取。相应地，传统闪存控制器使用单块数据存取技术，这限制了闪存系统的性能。

在本发明中，闪存控制器采用大小为 2048 字节或更大的页面暂存器。本发明中的闪存控制器为多块存取控制器，通过同时向闪存设备发送多块数据来写入页面暂存器。与传统的单块数据传输控制器相比，该控制器显著提高了数据传输的性能。

本发明中的闪存控制器可同时进行双通道处理，从而进一步改善了闪存系统的性能。双通道计数可提供第二通道，或“自由通道”，用于执行闪存控制器和闪存设备之间的事务。传统闪存控制器采用单存储器总线结构，多个闪存设备同时与总线连接。然而，传统的单通道架构限制了传统闪存控制器的性能。

在本发明中，至少采用了两条存储器总线。各条存储器总线都和独立的闪存设备相连。存储器控制器可以同时或单独存取各闪存设备。结果，操作执行可达到采用双通道处理的两倍速度。此外，各存储器总线还可以进一步扩展为多条存储器总线结构。

本发明的闪存控制器还可进行交错式操作。传统闪存控制器采用单存储器总线结构，多个闪存设备同时与总线连接。然而，传统闪存控制器每次只能存取

一个闪存设备，而这限制了系统的性能。

本发明中，至少采用了一个或两个存储器控制信号（例如片选和忙）。此外，共享的存储器总线至少有两个闪存设备与的相连。当一个闪存设备为读忙或写忙时，本发明中的闪存控制器可存取另一个闪存设备。相应地，本发明中的闪存控制器充分利用了共享存储器总线，从而显著提高了性能。此外，通过共享存储器 I/O 和控制信号，减少了闪存控制器的管脚数量。这使闪存系统的成本实现了最小化。

在本发明中，在单个闪存设备的存储器取周期中同时集成多块存取技术，多存储器交错技术和多通道操作技术，使其达到了最佳性能。

本发明提出的系统和方法具有诸多优点。例如，大大提高了闪存控制器在块管理操作过程中的搜索可用好块的速度。同时，消除了闪存控制器对外部缓冲区的需求。此外，闪存控制器支持多块数据存取、双通道处理和多存储单元交错式存取。相应地，大大提高了块管理操作执行速度。

本发明主要介绍了闪存块管理系统和方法。该系统和方法提出了一种配有处理器的闪存控制器，用于执行闪存系统操。这里的操作是指块管理操作，具体包括坏块处理、过期块回收和损耗平衡操作。处理器

利用来自仲裁逻辑的数据，对闪存系统中特定的闪存设备执行这些操作。由于这些操作都发生在特定的闪存设备内部，处理器可以利用来自仲裁逻辑的数据把可用好块搜索限定在特定的闪存设备内部。同时，在有效数据被重新放置前的搜索过程中，处理器可以利用闪存设备的内部缓冲区来储存有效数据。结果，可用好块搜索时间大大减少，同时消除了对外部缓冲区的需求。相应地，块管理操作的执行速度将显著提高。

虽然本说明书介绍的是带有或不带有指纹识别功能的电子数据闪存卡，在本发明思想和范围之内，本发明同样适用于其它类型的存储器系统。此外，虽然本文所介绍的是 USB 标准，在本发明思想和范围之内，本发明同样适用于其它标准。此外，本发明的方案可通过硬件、软件、包含程序指令的计算机可读介质或其组合实现。相应地，通过相关技术对本发明进行的修改仍在以下权利要求的思想和范围之内。

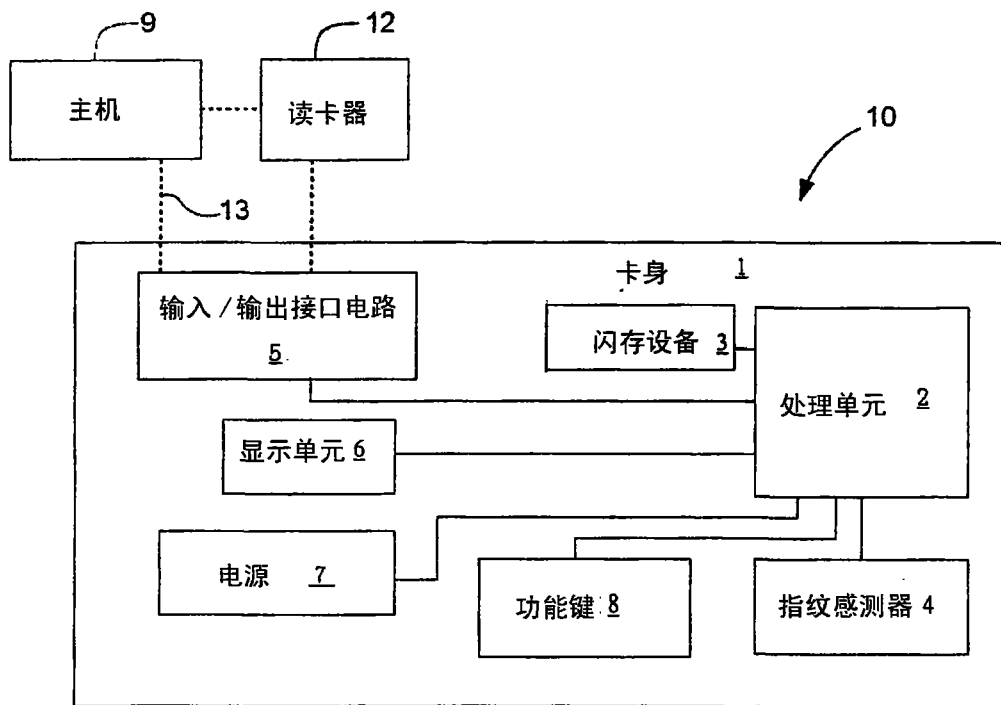


图 1

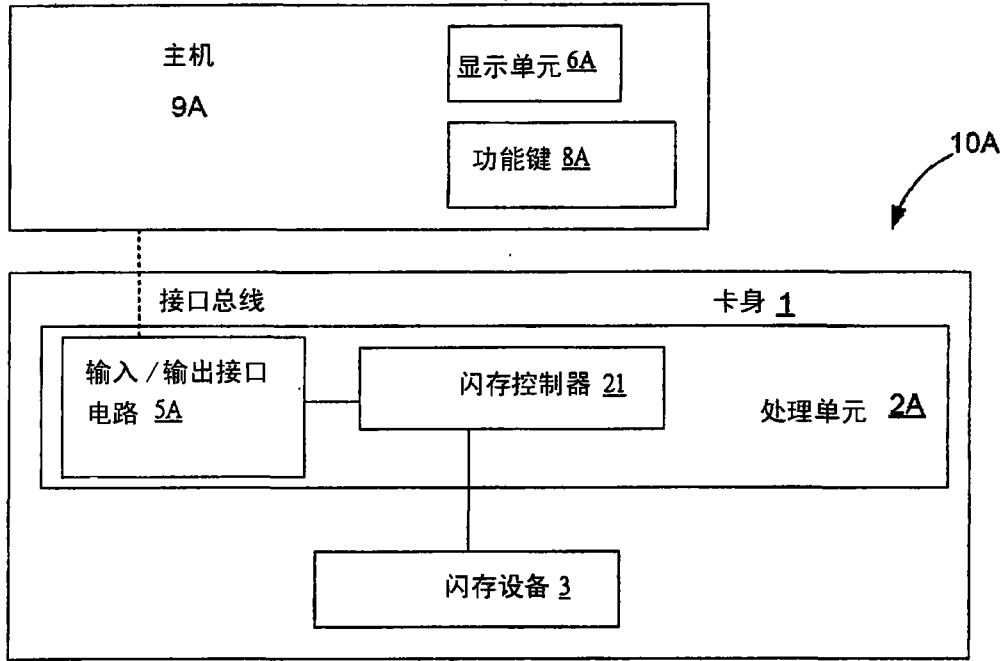


图 2

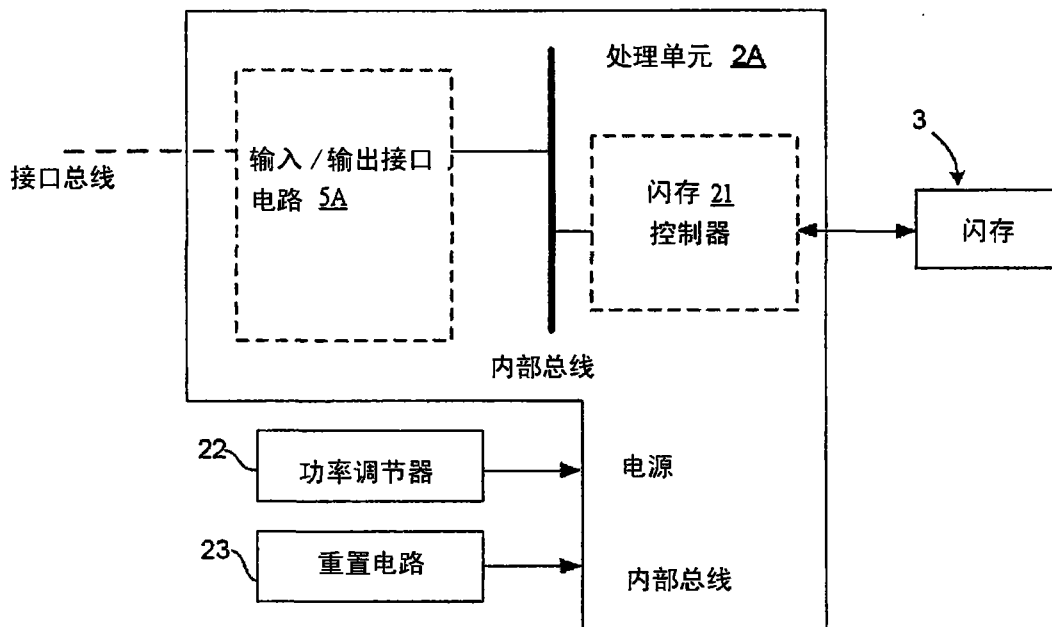


图 3

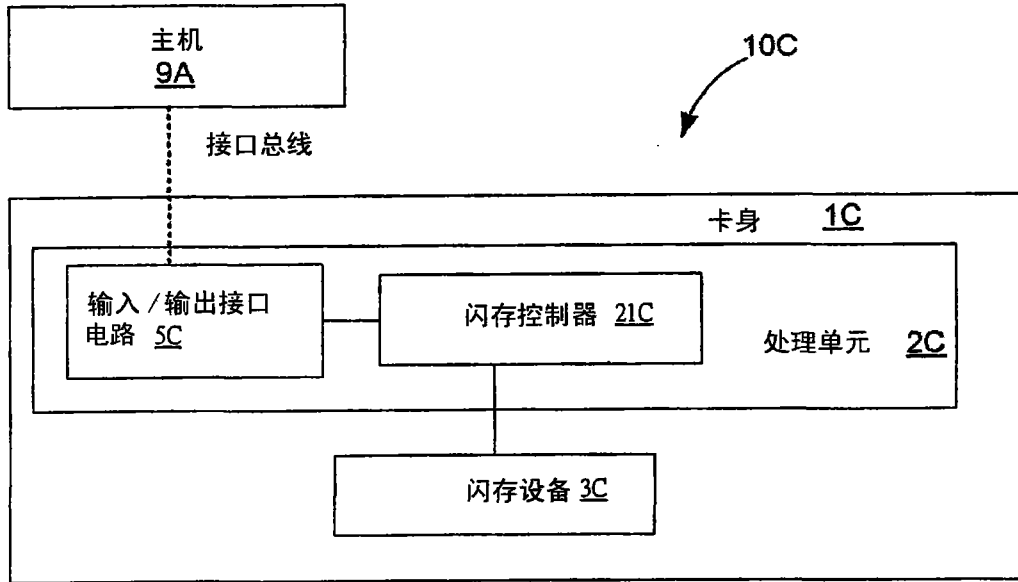


图 4

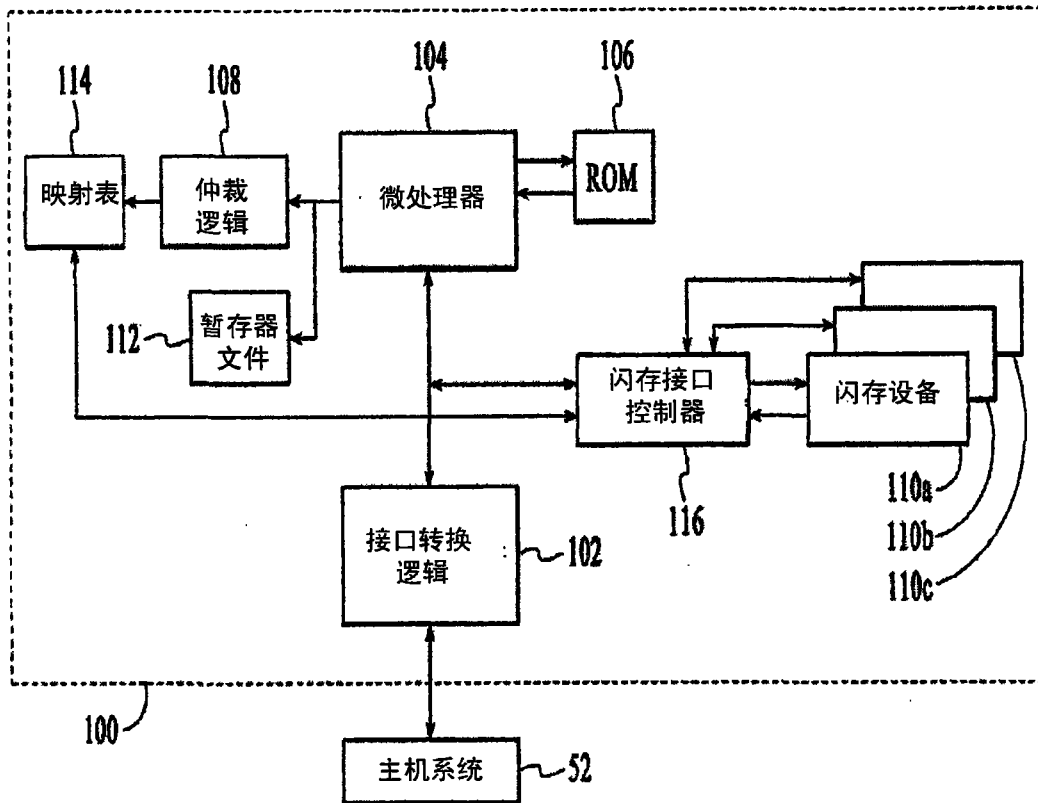


图 5

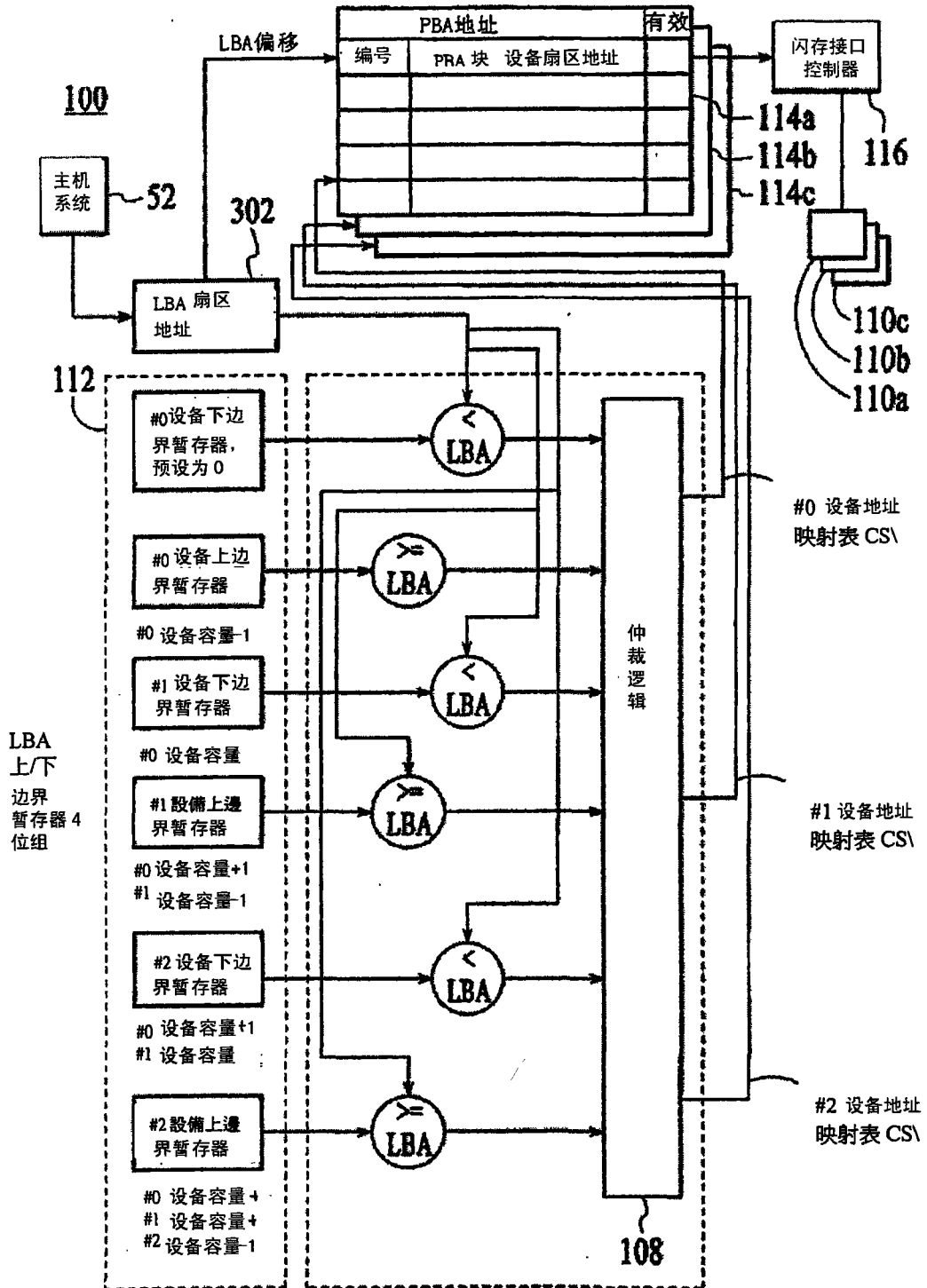


图 6

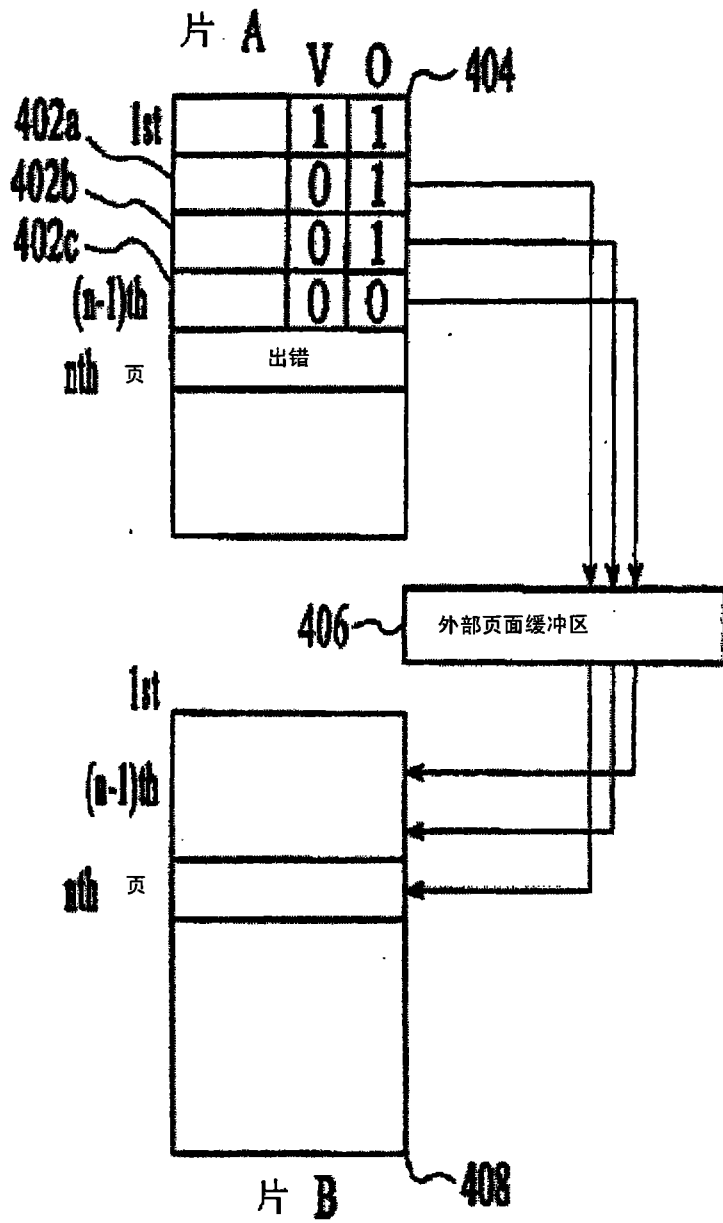


图 7

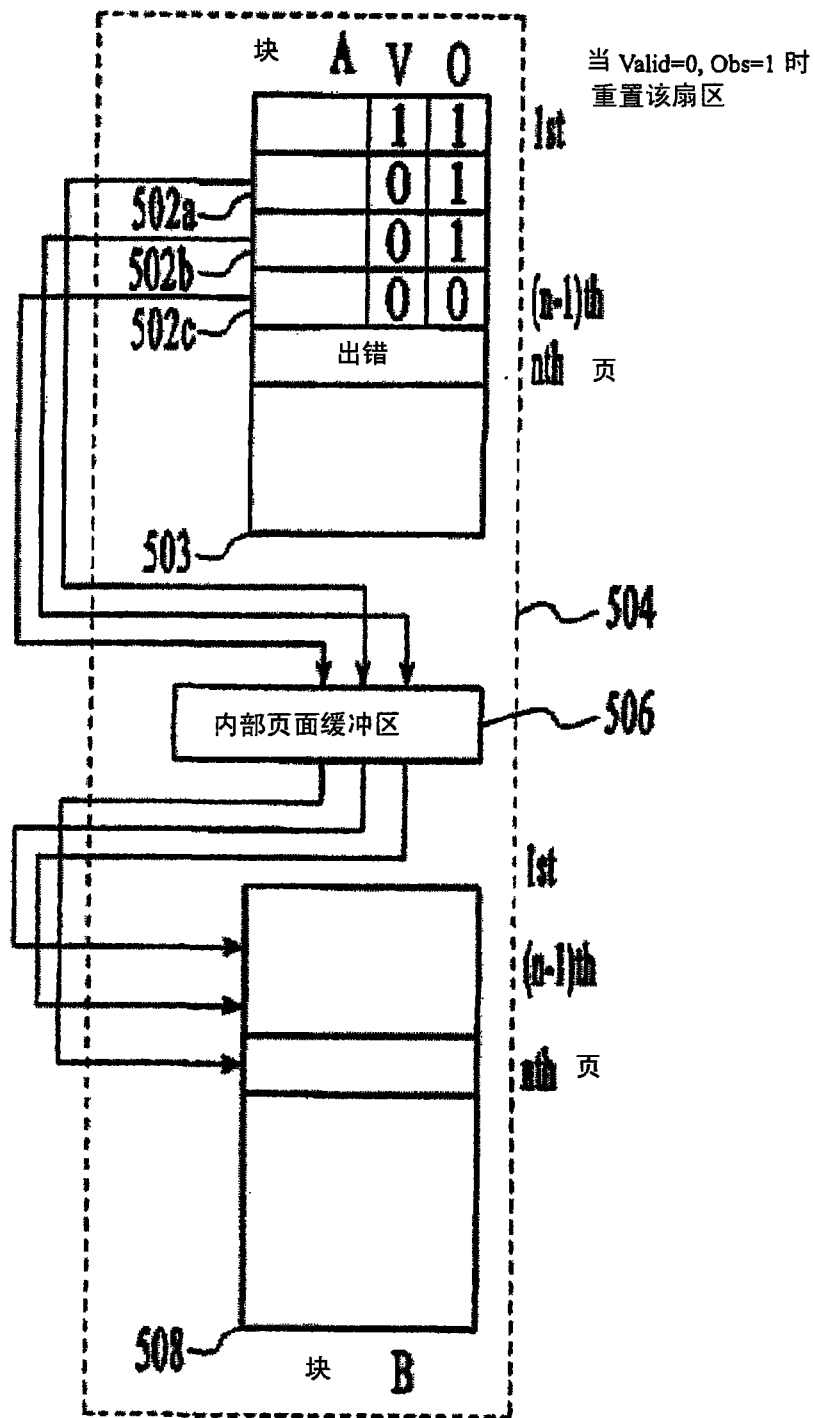


图 8

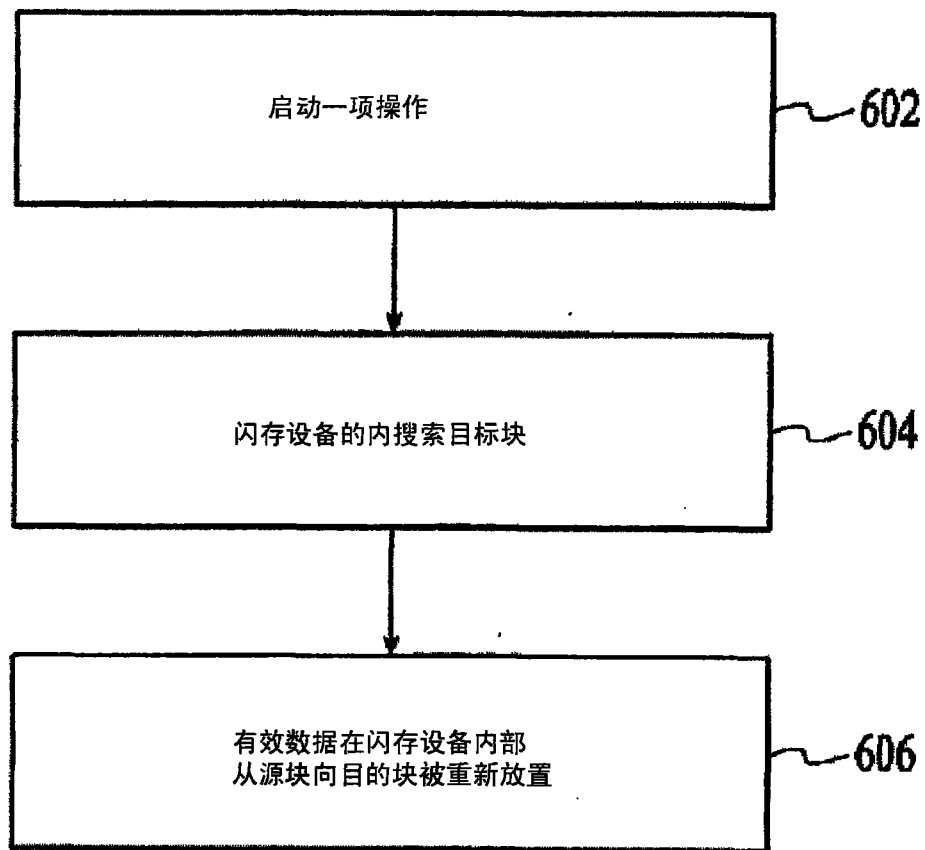


图 9

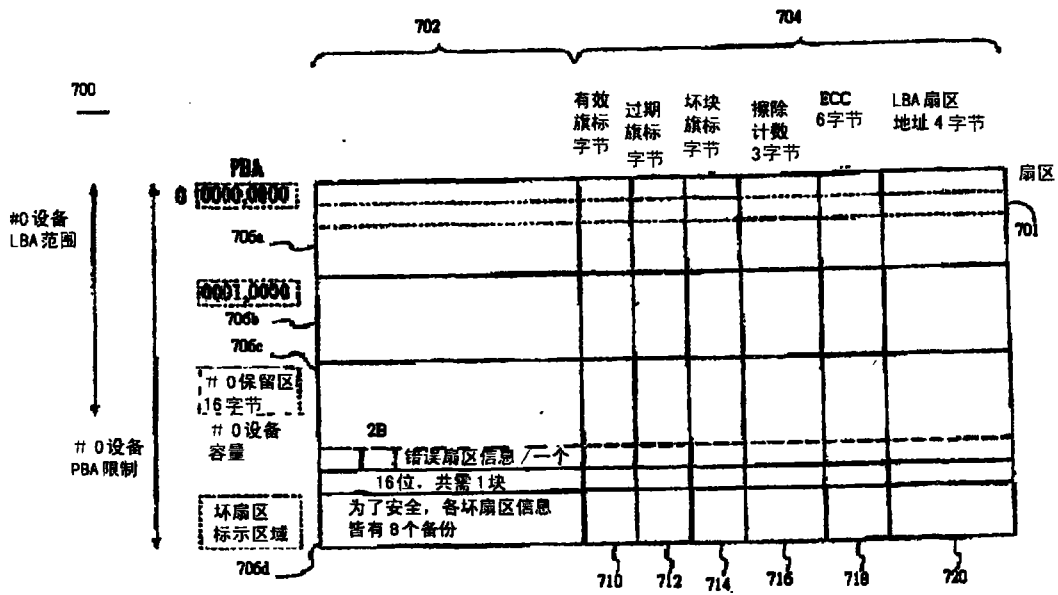


图 10

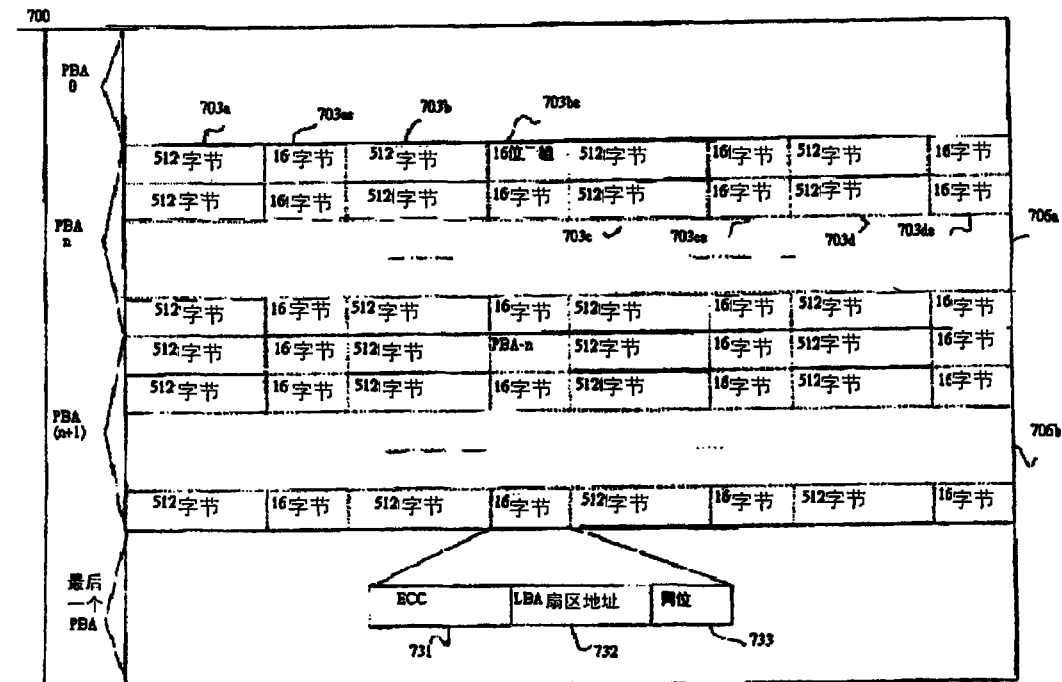


图 10 A

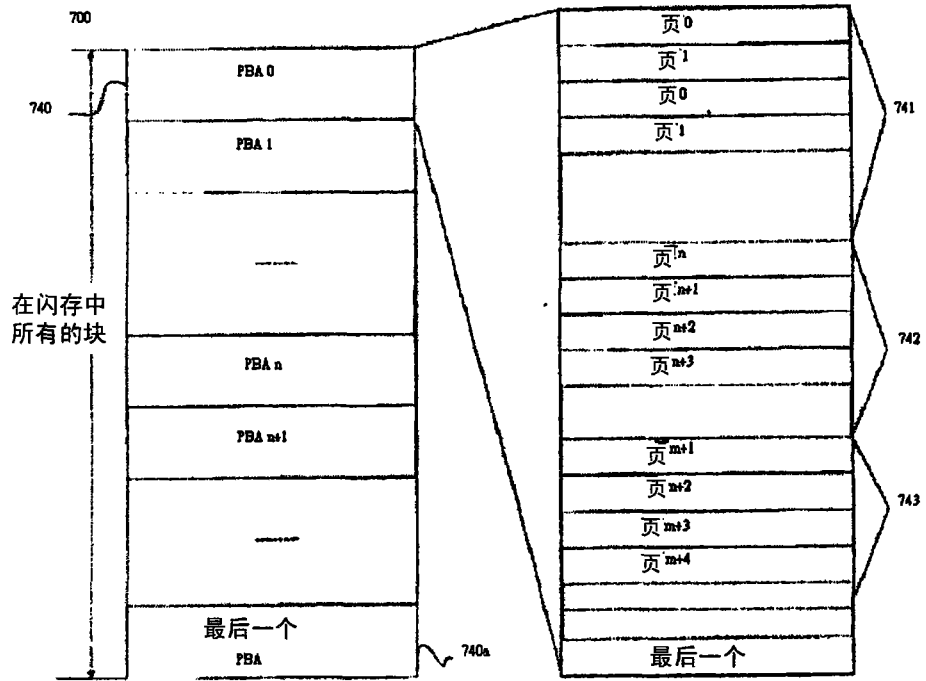


图 10B

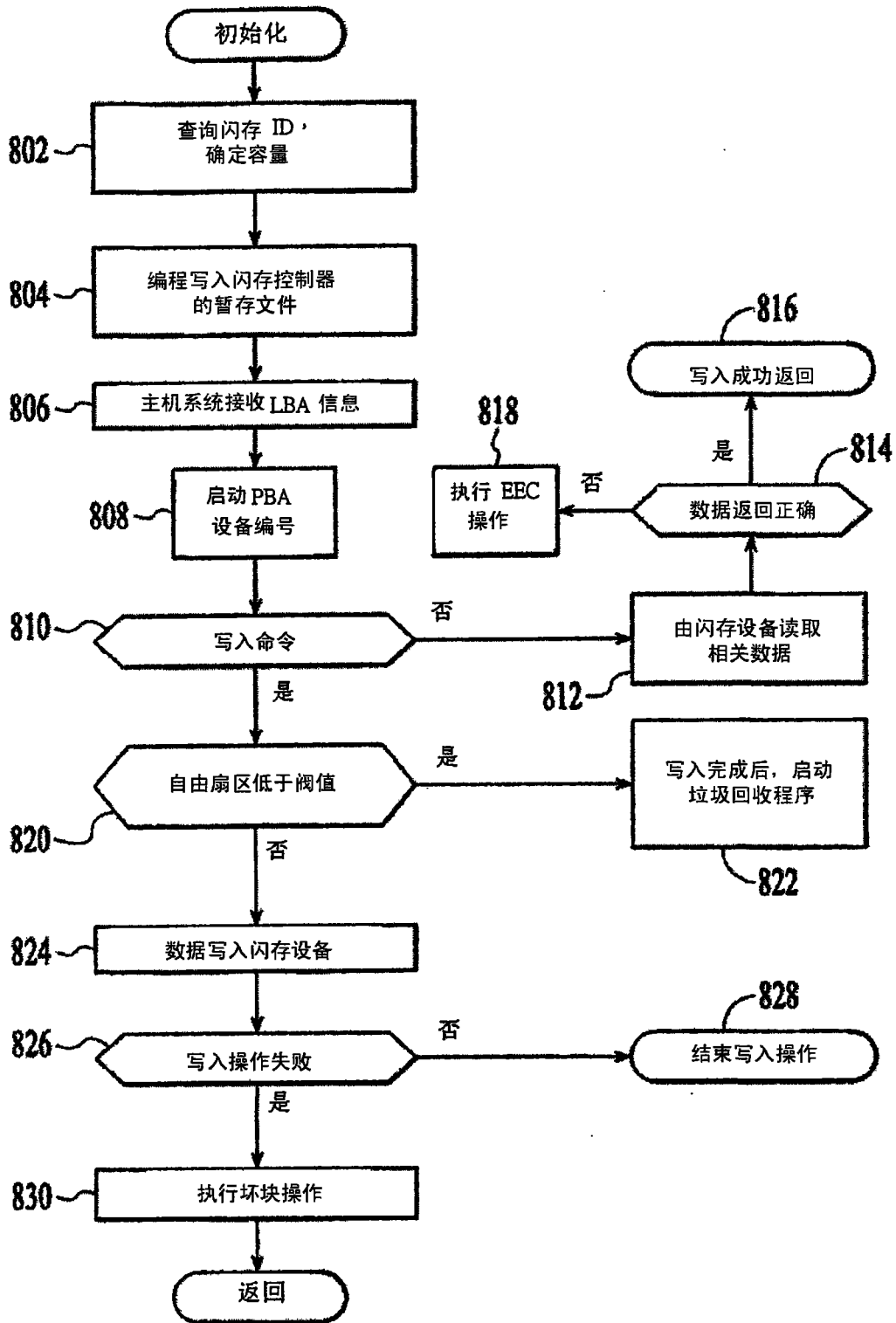


图 11

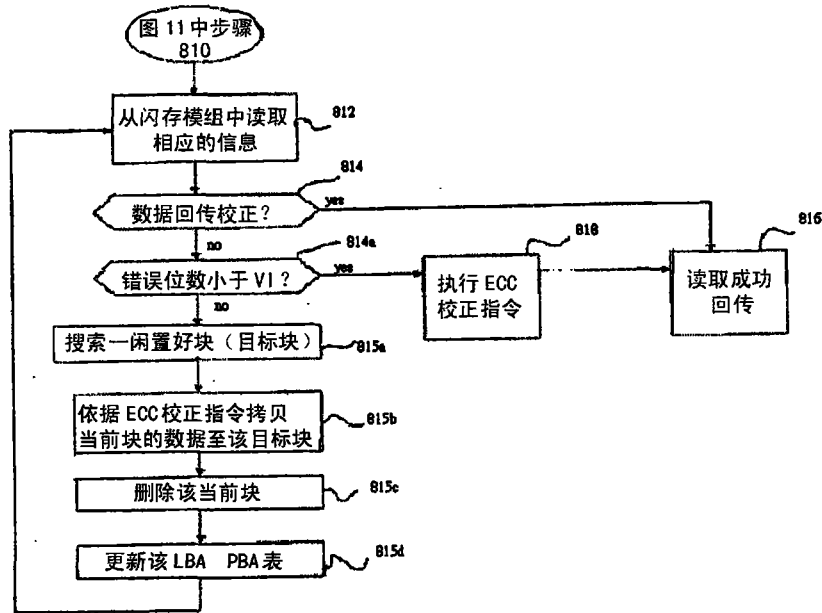


图 11 A

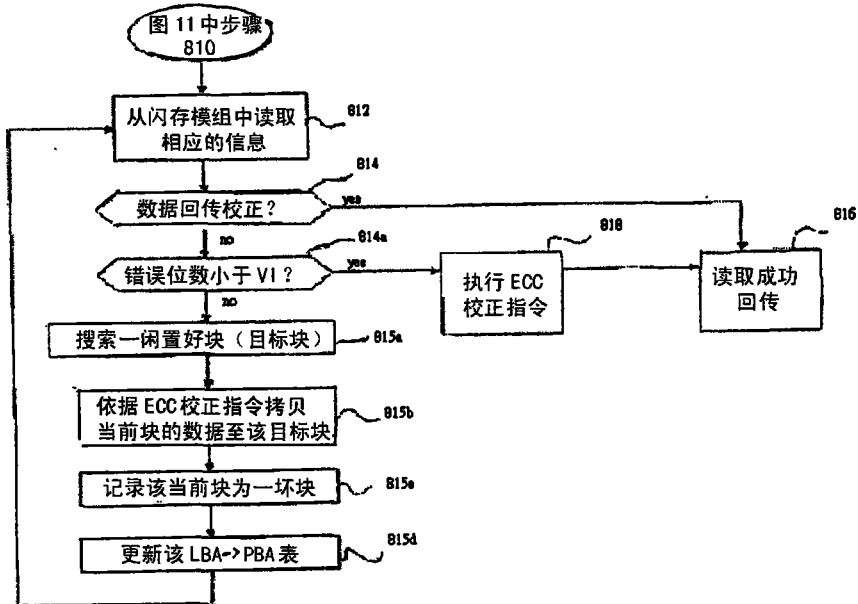


图 11 B

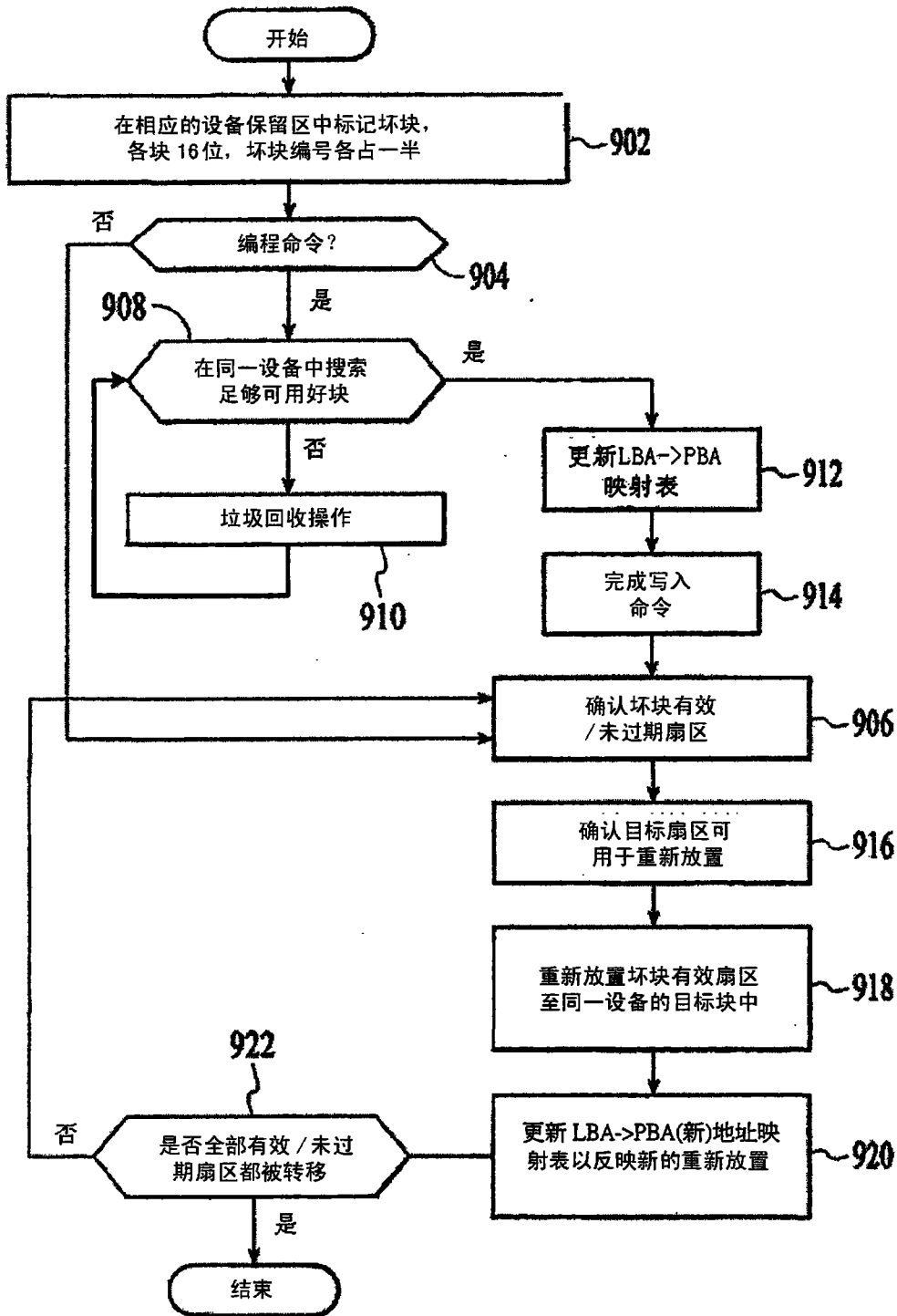


图 12

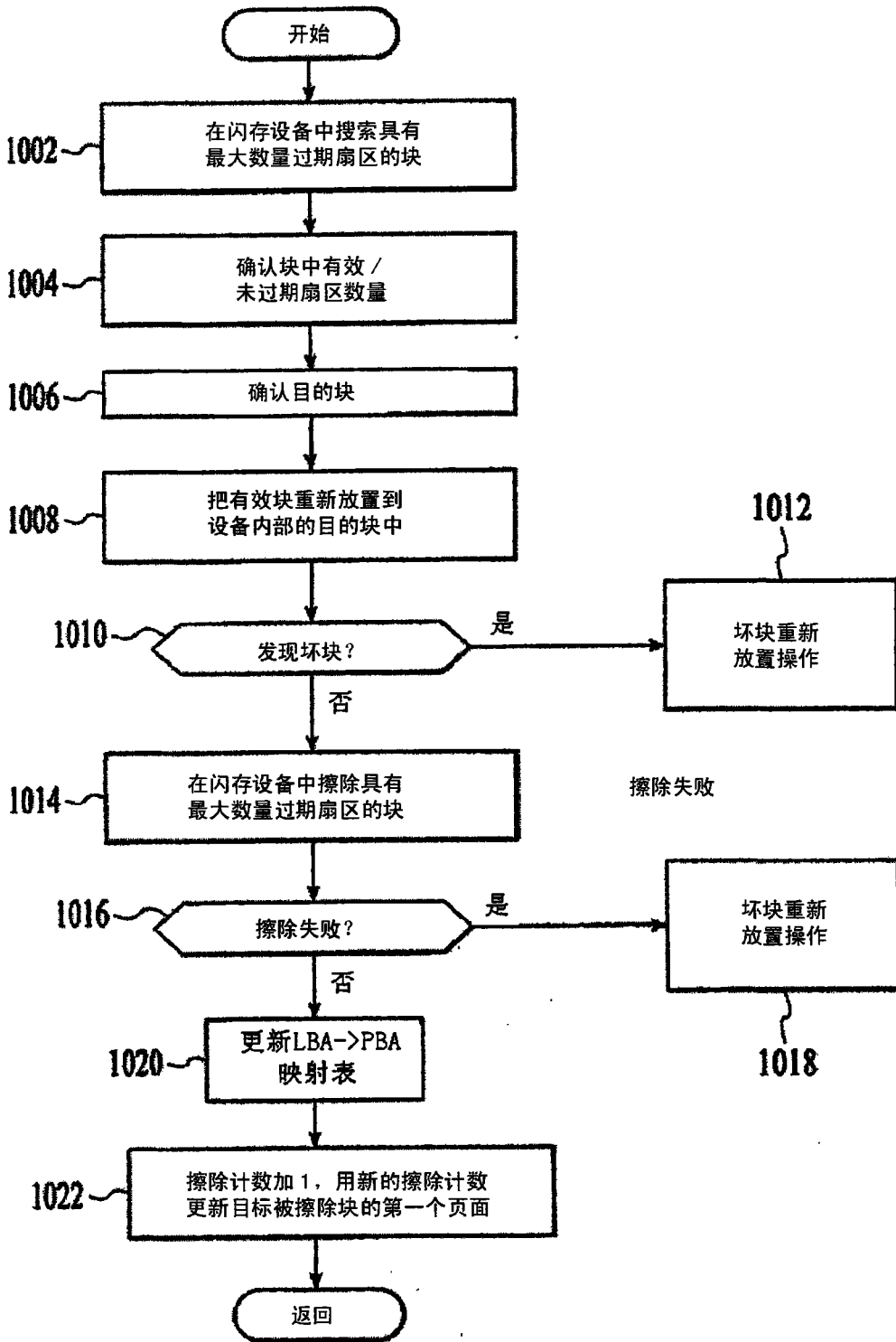


图 13

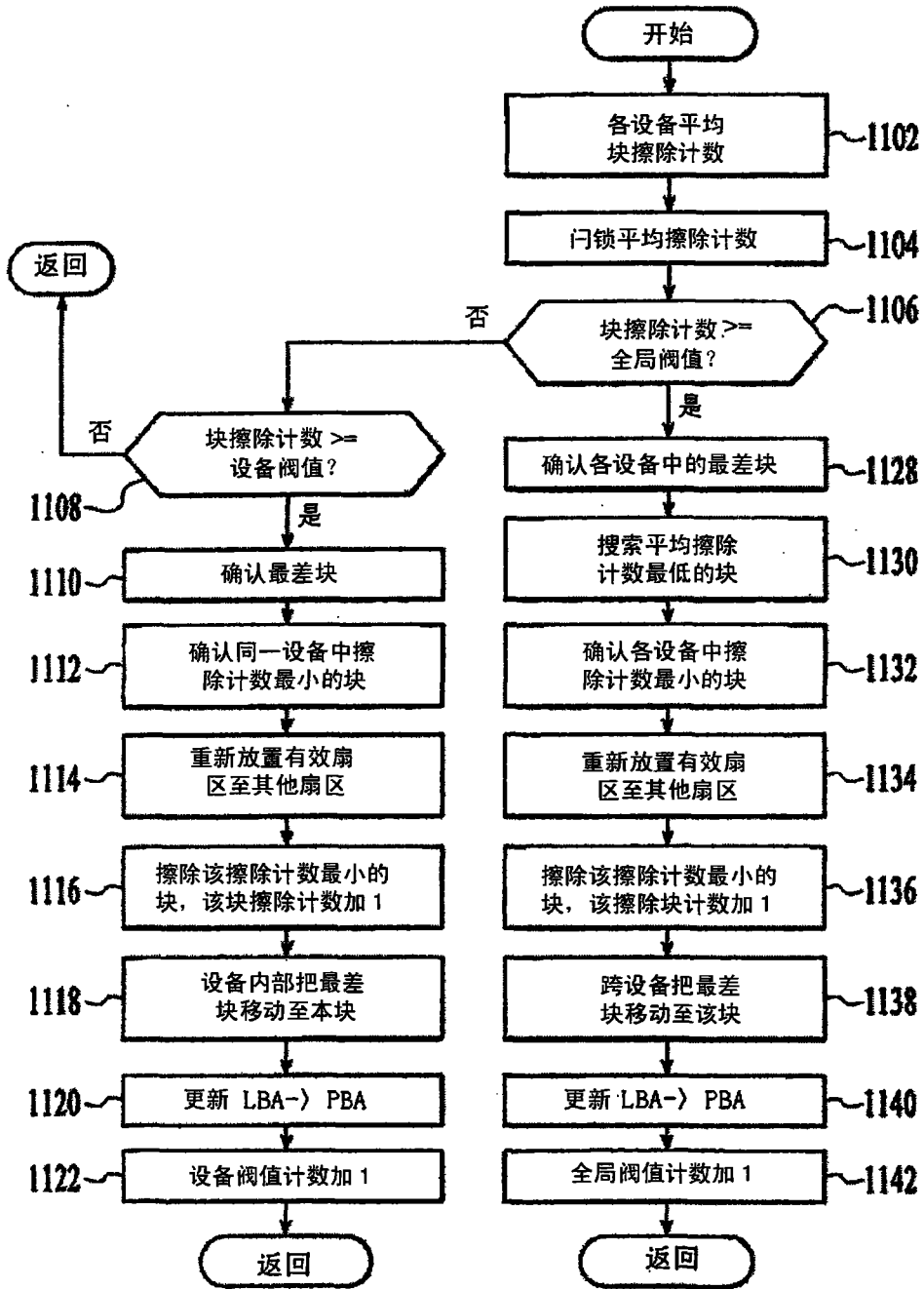


图 14

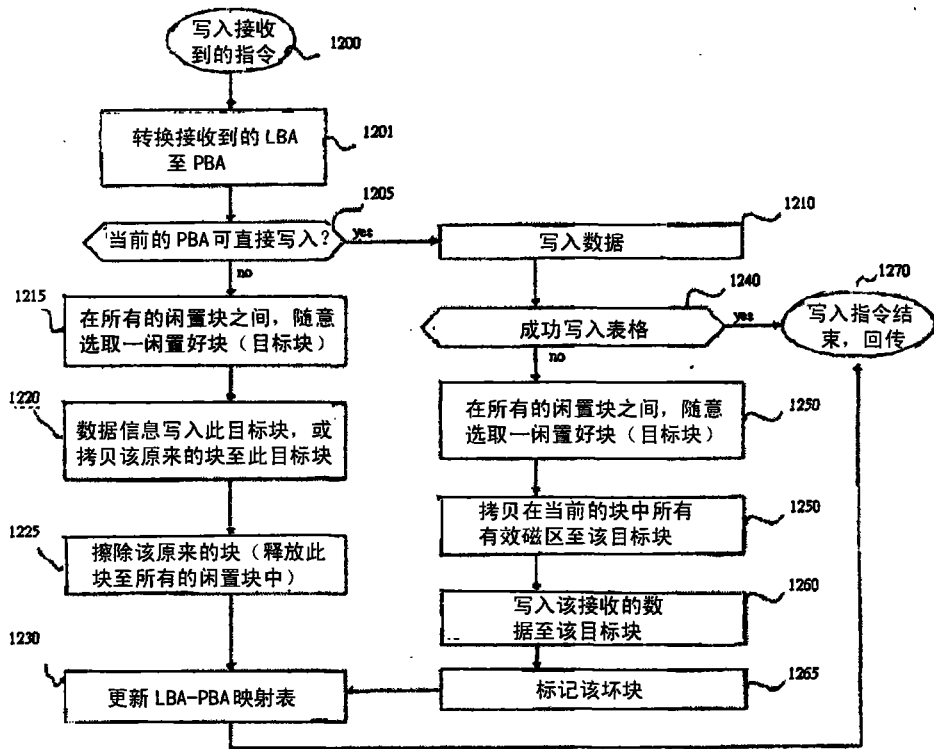


图 15