

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 3/06 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200680034856.4

[43] 公开日 2008年10月15日

[11] 公开号 CN 101288045A

[22] 申请日 2006.8.1

[21] 申请号 200680034856.4

[30] 优先权

[32] 2005.8.3 [33] US [31] 60/705,388

[32] 2005.12.21 [33] US [31] 11/316,577

[32] 2005.12.21 [33] US [31] 11/316,578

[86] 国际申请 PCT/US2006/030342 2006.8.1

[87] 国际公布 WO2007/019258 英 2007.2.15

[85] 进入国家阶段日期 2008.3.21

[71] 申请人 桑迪士克股份有限公司

地址 美国加利福尼亚州

[72] 发明人 艾伦·韦尔什·辛克莱

[74] 专利代理机构 北京律盟知识产权代理有限公司
代理人 刘国伟

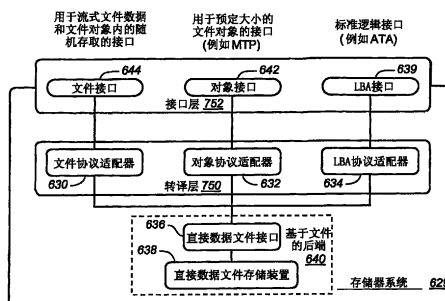
权利要求书5页 说明书26页 附图11页

[54] 发明名称

增强型主机接口

[57] 摘要

一种与使用不同协议的主机兼容的存储器系统包含用于所述不同协议的协议适配器。协议适配器允许将常用的后端系统用于以不同格式提供的数据。协议适配器产生对主机的响应并在适当时产生针对后端的命令。



1. 一种用于在非易失性存储器阵列中存储数据的存储器系统，所述存储器系统接收来自一个或一个以上应用程序的不同逻辑格式的数据，并以常用逻辑格式将数据存储在该存储器阵列中，所述存储器系统包括：

第一协议适配器，其接收来自第一应用程序的第一数据作为第一主机文件，在所述文件之前是所述第一主机文件长度的指示，并将所述第一数据发送到所述非易失性存储器阵列，其中所述第一数据存储在使用第一文件识别符记录的位置处；

第二协议适配器，其接收来自第二应用程序的第二数据作为识别为第二主机文件的数据的数据流而没有所述第二主机文件长度的指示，并将所述第二数据发送到所述非易失性存储器阵列，其中所述第二数据存储在使用第二文件识别符记录的位置处；以及

第三协议适配器，其接收来自第三应用程序的第三数据作为具有来自为所述存储器系统界定的逻辑地址范围的个别逻辑地址的多个扇区，并将所述第三数据发送到所述非易失性存储器阵列，其中所述第三数据存储在使用第三文件识别符记录的位置处。
2. 根据权利要求 1 所述的存储器系统，其中所述存储器系统包含在可通过标准化连接以可移除方式连接到主机系统的存储器卡中。
3. 根据权利要求 2 所述的存储器系统，其中所述第一应用程序在第一主机系统上运行，所述存储器卡在第一时间连接到所述第一主机系统；

所述第二应用程序在第二主机系统上运行，所述存储器卡在第二时间连接到所述第二主机系统；且

所述第三应用程序在第三主机系统上运行，所述存储器卡在第三时间连接到所述第三主机系统。
4. 根据权利要求 1 所述的存储器系统，其中所述第一、第二和第三数据的位置是用指示所述存储器阵列中对应于所述第一、第二和第三文件识别符中每一者的一个或一个以上区块的条目来记录的。
5. 根据权利要求 1 所述的存储器系统，其中所述第一协议适配器向所述主机产生所述第一主机文件已被接收的指示。
6. 根据权利要求 1 所述的存储器系统，其中所述第一协议适配器产生所述第一主机文件结束的指示符，所述指示符致使调度所述第一数据以进行垃圾收集。

7. 一种实施在可移除存储器卡上的存储器系统，所述可移除存储器卡连接到主机接口并存储通过所述主机接口接收的数据，所述存储器系统包括：
 - 非易失性存储器阵列；
 - 后端存储器管理系统，其将所述非易失性存储器阵列内的数据作为文件来管理；
 - 接口层，其与主机通信；
 - 转译层，其位于所述接口层与所述后端存储器管理系统之间，所述转译层接收来自所述接口层的主机命令，所述主机命令遵守对象协议，且响应于接收到所述主机命令，所述转译层向所述后端存储器管理系统产生经转译的命令，所述经转译的命令不遵守所述对象协议。
8. 根据权利要求7所述的存储器系统，其中所述对象协议是媒体传输协议（MTP）。
9. 根据权利要求7所述的存储器系统，其中主机在发送对象之前发送包含所述对象的大小的指示的元数据，且所述转译层根据所述指示确定何时已从所述主机接收到整个对象。
10. 根据权利要求9所述的存储器系统，其中响应于确定已接收到所述整个对象，所述转译层产生发送到所述主机的响应，并产生发送到所述后端存储器管理系统的文件结束指示符。
11. 根据权利要求7所述的存储器系统，其中所述转译层进一步包含：
 - 文件协议适配器，其将来自使用主机文件协议的第二主机的通信转译到所述后端文件协议；以及
 - LBA 协议适配器，其将来自使用逻辑地址协议的第三主机的通信转译到所述后端文件协议。
12. 根据权利要求11所述的存储器系统，其中所述LBA协议适配器从为所述存储器系统界定的逻辑地址空间接收具有由所述第三主机分配的逻辑地址的数据扇区，并将所述扇区映射到在大上等于所述存储器阵列的元区块的容量的虚拟文件。
13. 一种用于在非易失性存储器阵列中存储数据的存储器系统，所述存储器系统接收来自一个或一个以上应用程序的不同逻辑格式的数据，并以常用逻辑格式将数据存储于在所述存储器阵列中，所述存储器系统包括：
 - 第一协议适配器，其接收来自第一应用程序的第一数据作为第一主机文件，在所述文件之前是所述第一主机文件长度的指示，并将所述第一数据发送到所述非易失性存储器阵列，其中所述第一数据存储在使用第一文件识别符记录的位置处；以及

第二协议适配器，其接收来自第二应用程序的第二数据作为识别为第二主机文件的数据的数据流而没有所述第二主机文件长度的指示，并将所述第二数据发送到所述非易失性存储器阵列，其中所述第二数据存储在使用第二文件识别符记录的位置处。

14. 根据权利要求 13 所述的存储器系统，其中所述存储器系统包含在可通过标准化连接以可移除方式连接到主机系统的存储器卡中。

15. 根据权利要求 14 所述的存储器系统，其中所述第一应用程序在第一主机系统上运行，所述存储器卡在第一时间连接到所述第一主机系统；且

所述第二应用程序在第二主机系统上运行，所述存储器卡在第二时间连接到所述第二主机系统。

16. 根据权利要求 13 所述的存储器系统，其中所述第一和第二数据的位置是用指示所述存储器阵列中对应于所述第一和第二文件识别符中每一者的一个或一个以上区块的条目来记录的。

17. 根据权利要求 13 所述的存储器系统，其中所述第一协议适配器向所述主机产生所述第一主机文件已被接收的指示。

18. 根据权利要求 13 所述的存储器系统，其中所述第一协议适配器产生所述第一主机文件结束的指示符，所述指示符致使调度所述第一数据以进行垃圾收集。

19. 一种操作存储器系统以与多个主机协议兼容的方法，所述存储器系统包含区块可擦除存储器阵列和控制器，所述方法包括：

检测当连接到主机时的主机协议，并响应于检测到的所述主机协议从多个协议适配器中选择一协议适配器，所述多个协议适配器包含至少第一、第二和第三协议适配器；

响应于检测到第一主机协议选择所述第一协议适配器，所述第一主机协议发送预定长度的文件，在所述文件之前是文件长度指示符，个别文件具有唯一的文件指示符；

响应于检测到第二主机协议选择所述第二协议适配器，所述第二主机协议发送不具有文件长度指示符的文件，个别文件具有唯一的文件指示符；以及

响应于检测到第三主机协议选择所述第三协议适配器，所述第三主机协议发送数据扇区，每个扇区具有来自为所述存储器系统界定的逻辑地址范围的逻辑地址。

20. 根据权利要求 19 所述的方法，其进一步包括将通过所述第一协议适配器接收的第

一文件存储在所述存储器阵列的第一多个区块中，并记录所述第一文件识别符与所述第一多个区块之间的第一关联性；

将通过所述第二协议适配器接收的第二文件存储在所述存储器阵列的第二多个区块中，并记录所述第二文件识别符与所述第二多个区块之间的第二关联性；以及

将通过所述第三协议适配器接收的多个扇区存储在第三多个区块中，并记录所述第三文件识别符与所述第三多个区块之间的第三关联性。

21. 根据权利要求 19 所述的方法，其中所述第一协议适配器在已接收到所述预定长度的数据之后产生对所述第一主机的响应。
22. 根据权利要求 19 所述的方法，其中所述第一协议适配器响应于接收到所述预定长度的数据而产生文件结束指示符，所述指示符致使所述文件被关闭并经调度以进行垃圾收集。
23. 根据权利要求 19 所述的方法，其进一步包括在任何一个时间仅选择所述第一、第二或第三协议适配器中的一者。
24. 根据权利要求 19 所述的方法，其进一步包括在不同时间交替地选择所述第一、第二和第三协议适配器中的不同一者以允许一个以上主机应用程序的交错存取。
25. 根据权利要求 19 所述的方法，其中所述存储器系统实施在可移除存储器卡中。
26. 根据权利要求 25 所述的方法，其中所述存储器卡符合选自以下各项的标准：紧致快闪存储器、多媒体卡、安全数字、小型 SD、记忆棒、智能媒体或 TransFlash。
27. 一种操作可移除存储器卡上的非易失性存储器系统的方法，其包括：

从主机接收元数据，所述元数据包含对象中数据量的指示；

随后从所述主机接收所述对象，并通过将从所述主机接收的对象数据量与所述元数据所指示的数据量进行比较来确定是否已接收到整个对象；以及

响应于确定已接收到所述整个对象，向所述主机发送响应并关闭所述存储器系统中对应于所述对象的文件，所述文件由于被关闭而被标记以进行垃圾收集。
28. 根据权利要求 27 所述的方法，其进一步包括将所述元数据存储在该所述非易失性存储器系统中。
29. 根据权利要求 27 所述的方法，其中所述主机维持包含所述对象的分级，且所述对象在所述分级内的位置由所述元数据提供。
30. 根据权利要求 29 所述的方法，其中将所述文件存储在所述存储器系统中而不用考虑其在所述分级中的位置。

31. 根据权利要求 30 所述的方法，其中随后当主机请求所述文件时，所述元数据用于重新产生所述分级。
32. 一种操作可移除存储器系统以与多个主机协议兼容的方法，所述存储器系统包含区块可擦除存储器阵列和控制器，所述方法包括：
检测当连接到主机时的主机协议，并响应于检测到的所述主机协议从多个协议适配器中选择一协议适配器，所述多个协议适配器包含至少第一协议适配器和第二协议适配器；
响应于检测到第一主机协议选择所述第一协议适配器，所述第一主机协议发送预定长度的文件，在所述文件之前是文件长度指示符，个别文件具有唯一的文件识别符；以及
响应于检测到第二主机协议选择所述第二协议适配器，所述第二主机协议发送不具有文件长度指示符的文件，个别文件具有唯一的文件识别符。
33. 根据权利要求 32 所述的方法，其进一步包括将通过所述第一协议适配器接收的第一文件存储在所述存储器阵列的第一多个区块中，并记录所述第一文件识别符与所述第一多个区块之间的第一关联性；以及
将通过所述第二协议适配器接收的第二文件存储在所述存储器阵列的第二多个区块中，并记录第二文件识别符与所述第二多个区块之间的第二关联性。
34. 根据权利要求 32 所述的方法，其中所述第一协议适配器在已接收到所述预定长度的数据之后产生对所述第一主机的响应。
35. 根据权利要求 32 所述的方法，其中所述第一协议适配器响应于接收到所述预定长度的数据而产生文件结束指示符，所述指示符致使所述文件被关闭并被调度以进行垃圾收集。
36. 根据权利要求 32 所述的方法，其进一步包括在任何一个时间仅选择所述第一或第二协议适配器中的一者。
37. 根据权利要求 32 所述的方法，其进一步包括选择第一或第二协议适配器中的交替一者，其中所述可移除存储器系统与一个以上主机通信。

增强型主机接口

技术领域

无

背景技术

本申请案涉及例如半导体快闪存储器的可重新编程的非易失性存储器系统的操作，且更具体地说，涉及主机装置与存储器之间的接口的管理。本文引用的所有专利、专利申请案、文章和其它出版物、文档和事物（包含上文在“相关申请案的交叉参考”下引用的所有申请案）出于所有目的在此以全文引用的方式并入本文中。

在早一代商业快闪存储器系统中，将矩形存储器单元阵列划分为大量单元群组，每一者存储标准磁盘驱动器扇区的数据量，即 512 个字节。每个群组中还通常包含例如 16 个字节的额外数据量，以存储错误校正码（ECC）和可能的与用户数据和/或其中存储用户数据的存储器单元群组相关的其它开销数据。每个此群组中的存储器单元是可共同擦除的最小数目的存储器单元。也就是说，擦除单位有效地是存储一个数据扇区和所包含的任何开销数据的大量存储器单元。此类型的存储器系统的实例描述于第 5,602,987 号和第 6,426,893 号美国专利中。需要在用数据对其再编程之前擦除存储器单元是快闪存储器的特性。

最常见以存储器卡或快闪驱动器的形式提供快闪存储器系统，所述存储器卡或快闪驱动器与例如个人计算机、相机等多种主机可移除地连接，但也可嵌入在这些主机系统内。当向存储器写入数据时，主机通常向存储器系统的连续虚拟地址空间内的扇区、群集或其它数据单位指派唯一的逻辑地址。类似于磁盘操作系统（DOS），主机向存储器系统的逻辑地址空间内的地址写入数据以及从其读取数据。存储器系统内的控制器将从主机接收的逻辑地址转译为实际存储数据的存储器阵列内的物理地址，且接着跟踪这些地址转译。存储器系统的数据存储容量至少与在为存储器系统界定的整个逻辑地址空间上可寻址的数据量一样大。

在较晚的几代快闪存储器系统中，擦除单位的大小增加到足够的存储器单元的区块以存储多个数据扇区。尽管与存储器系统连接的主机系统可以例如扇区的较小的最小单位来编程和读取数据，但大量扇区存储在快闪存储器的单个擦除单位中。常见的是区块内的一些数据扇区在主机更新或替换逻辑数据扇区时变过时。由于必须在可对区块中存

储的任何数据进行覆写之前擦除整个区块，因此新的或更新的数据通常存储在已经擦除且具有用于数据的剩余容量的另一区块中。此过程清空了带有占据存储器内有价值的空间的过时数据的原始区块。但如果其中保留有任何有效数据，那么无法擦除所述区块。

因此，为了更好地利用存储器的存储容量，常常合并或收集有效部分区块数据量，这通过以下方式完成：将数据复制到经擦除的区块中使得这些复制的数据所来自的区块接着可经擦除且其整个存储容量可再使用。还期望对数据进行复制以便按区块内数据扇区的逻辑地址的次序对其进行分组，因为这增加读取数据并将读取的数据传送到主机的速度。如果这种数据复制的发生过于频繁，那么存储器系统的操作性能可能降级。在存储器的存储容量几乎与主机通过系统的逻辑地址空间可寻址的数据量没有差别的情况（典型情况）下，这尤其影响存储器系统的操作。在此情况下，在可执行主机编程命令之前可能需要进行数据合并或收集。因此编程时间增加。

在连续几代存储器系统中，区块的大小逐渐增加，以便增加给定半导体面积中可存储的数据位的数目。存储 256 个数据扇区和更多数据扇区的区块变得常见。另外，常常将不同阵列或子阵列的两个、四个或更多区块逻辑上链接在一起成为元区块，以便增加数据编程和读取时的并行性程度。与这种大容量操作单位一起出现的是有效地对其进行操作的挑战。

随着新的革新允许较大的存储器容量和速度，通常需要提供使用此类革新但仍与不使用此类革新的产品兼容的产品。这意味着新产品可利用技术革新但仍向后兼容，使得其可与使用较老技术的产品一起使用。这种向后兼容性对于可在具有多种不同技术的多种配置中使用的便携式产品来说尤其重要。此便携式产品的一个实例是可移除快闪存储器卡。

发明内容

一种存储器系统包含与主机通信的接口层以及将数据存储存储在存储器阵列中的后端。在所述接口层与所述后端之间，转译层根据主机使用的不同协议将由所述接口层接收的数据和命令转换为后端可理解的格式。因此，转译层允许常用的后端与使用多种协议的多种主机一起使用。这对于可移除存储器卡中的存储器系统来说是尤其有用的。转译层含有一个或一个以上协议适配器。协议适配器根据主机所使用的协议接收来自主机的通信（命令和数据），且作为响应为后端转换所述数据和命令。协议适配器还可对主机产生信号，其中这些信号是主机协议的一部分。

一种存储器系统包含对象协议适配器，其将根据对象协议发送的数据和命令转换成

与非易失性存储器中的基于文件的存储装置兼容的格式。特定来说，对象协议适配器在接收对象之前接收关于对象的元数据。对象的大小包含在元数据中。对象协议适配器通过将所接收数据量与元数据信息所指示的大小进行比较来确定何时已接收到整个对象。当对象协议适配器确定已接收到整个对象时，其产生对主机的响应并对存储器系统的后端产生文件结束指示符，使得文件被后端关闭。这允许后端调度文件以进行垃圾收集，并因而允许更有效地存储和管理文件数据。

一种存储器系统包含 LBA 协议适配器，其将根据 LBA 协议的数据和命令转换成与非易失性存储器中的基于文件的存储装置兼容的数据和命令。在一个实例中，将所接收的具有由主机从为存储器系统界定的逻辑地址空间指派的逻辑地址的数据映射到逻辑文件。逻辑文件接着由后端以与其它文件相同的方式处理。逻辑文件通常占据整个元区块，使得其不会与其它数据共享元区块。然而，相同的区块可在一个时间用于逻辑文件且在其它时间用于其它文件，使得存储器阵列在不同类型的文件之间没有硬划分。

一种存储器系统包含文件协议适配器，其将根据文件协议的数据和命令转换成与非易失性存储器中的基于文件的存储装置兼容的数据和命令。在后端使用与主机相同的协议的情况下（例如，均使用直接数据文件协议），可能不必需要转译。然而在主机使用不同的文件协议的情况下，文件协议适配器做出适当的转译。

在一些情况下，存储器系统可能与使用一个以上协议适配器的一个以上主机通信。在所述情况下，转译层可每次选择一个协议适配器来与后端通信。在一些情况下，转译层可通过交替选择不同的协议适配器在不同的主机之间进行仲裁，以提供对后端的交错存取。

接口层包含与多个主机兼容的逻辑接口。在一些情况下，可能存在单独的物理接口用于与主机装置上的相应接口进行连接。然而这不是必要的，且在一些情况下，提供例如 USB 连接器的单个物理接口并由所有逻辑接口使用。接口层和转译层功能可由专用电路执行或可由控制器上的固件执行。这可以是管理存储器阵列的存储器控制器。存储器阵列可以是 NAND 存储器阵列且可形成于一个或一个以上半导体芯片上。存储器系统可包含在在不同时间连接到不同主机的可移除卡中。

后端系统可将数据作为文件来管理，所述文件在一些情况下对应于主机文件（但在一些情况下，没有与主机文件的一一对应）。基于文件的后端系统的一个实例是第 11/060,174、11/060,248 和 11/060,249 号美国专利申请案以及第 60/705,388 号临时专利申请案中描述的直接数据文件后端。

附图说明

图 1 示意性说明目前实施的主机和连接的非易失性存储器系统；

图 2 是用作图 1 的非易失性存储器的示范性快闪存储器系统的框图；

图 3 是可在图 2 的系统中使用的存储器单元阵列的代表电路图；

图 4 说明图 2 的系统的示范性物理存储器组织；

图 5 展示图 4 的物理存储器的一部分的扩展图；

图 6 展示图 4 和 5 的物理存储器的一部分的进一步扩展图；

图 7 说明主机与可重新编程的存储器系统之间的常见现有技术逻辑地址接口；

图 8 以与图 7 不同的方式说明主机与可重新编程的存储器系统之间的常见现有技术逻辑地址接口；

图 9 说明根据本发明的主机与可重新编程的存储器系统之间的直接文件存储接口；

图 10 以与图 9 不同的方式说明根据本发明的主机与可重新编程的存储器系统之间的直接文件存储接口；

图 11 展示用于存储作为具有来自为存储器界定的共用逻辑地址空间的逻辑地址的扇区而从主机接收的主机文件的方案，所述扇区映射到逻辑文件且逻辑文件接着存储在存储器阵列中，其中每个元区块一个逻辑文件；

图 12 以不同于图 11 的方式说明用于将逻辑寻址数据存储在存储器阵列中的逻辑文件中的方案。

图 13 展示具有与基于文件的后端通信的文件接口与 LBA 接口的存储器系统，其中 LBA 协议适配器插入在 LBA 接口与基于文件的后端之间。

图 14A 展示 MTP “发送对象信息” 事务。

图 14B 展示 MTP “发送对象” 事务。

图 15 展示具有对象接口和基于文件的后端的存储器系统，其中对象协议适配器插入在对象接口与基于文件的后端之间。

图 16 展示具有文件接口、对象接口和 LBA 接口的存储器系统，其具有文件协议适配器、对象协议适配器和 LBA 协议适配器以便于接口与基于文件的后端之间的通信。

图 17 展示图 16 的存储器系统的替代视图，其中将文件接口、对象接口和 LBA 接口视为接口层的一部分，且将文件协议适配器、对象协议适配器和 LBA 协议适配器视为转译层的一部分。

具体实施方式

快闪存储器的大体描述

相对于图 1 到图 8 描述当前快闪存储器系统和与主机装置的典型操作。在此系统中可实施本发明的各个方面。图 1 的主机系统 1 将数据存储到快闪存储器 2 中并从其中检索数据。尽管快闪存储器可嵌入在主机内，但将存储器 2 说明为较普遍形式的卡，其通过机械和电气连接器的配合部分 3 和 4 可移除地连接到主机。目前存在许多不同的市售的快闪存储器卡，实例是紧致快闪存储器（CF）、多媒体卡（MMC）、安全数字（SD）、小型 SD、记忆棒、智能媒体和 TransFlash 卡。尽管这些卡中的每一者均具有根据其标准化规范的独特的机械和/或电气接口，但每一者中包含的快闪存储器是非常相似的。这些卡可从本申请案的受让人 SanDisk 公司购得。SanDisk 还提供以 Cruzer 为商标的快闪驱动器线，其为小封装的手持存储器系统，所述系统具有用于通过插入主机的通用串行总线（USB）插座而与主机连接的 USB 插头。这些存储器卡和快闪驱动器中的每一者均包含与主机介接并控制其内快闪存储器的操作的控制器。

使用这些存储器卡和快闪驱动器的主机系统有许多且各式各样。它们包含个人计算机（PC）、膝上型和其它便携式计算机、蜂窝式电话、个人数字助理（PDA）、数字静态相机、数字电影相机和便携式音频播放器。主机通常包含用于一种或一种以上类型的存储器卡或快闪驱动器的内置插座，但其中一些需要能插入存储器卡的适配器。

就存储器 2 涉及到的范围而言，图 1 的主机系统 1 可被视为具有两个主要部分，由电路和软件的组合构成。它们是应用程序部分 5 以及与存储器 2 介接的驱动程序部分 6。在例如个人计算机中，应用程序部分 5 可包含运行字处理、图形、控制或其它流行的应用程序软件的处理器。在相机、蜂窝式电话或其它主要专用于执行单一功能集合的主机系统中，应用程序部分 5 包含操作相机以拍摄和存储照片、操作蜂窝式电话以打电话和接电话等等的软件。

图 1 的存储器系统 2 包含快闪存储器 7 和电路 8，电路 8 和与卡连接以用于来回传递数据的主机介接，且还控制存储器 7。控制器 8 通常在数据编程和读取期间在主机 1 使用的数据的逻辑地址与存储器 7 的物理地址之间进行转换。

参看图 2，其描述可用作图 1 的非易失性存储器 2 的典型快闪存储器系统的电路。系统控制器通常实施在单个集成电路芯片 11 上，其通过系统总线 13 与一个或一个以上集成电路存储器芯片并联连接，图 2 中展示单个这种存储器芯片 15。所说明的特定总线 13 包含用于携载数据的单独一组导体 17、用于存储器地址的一组导体 19 以及用于控制和状态信号的一组导体 21。或者，单个一组导体可在这三个功能之间时间共享。此外，

可采用系统总线的其它配置，例如 2004 年 8 月 9 日申请的标题为“Ring Bus Structure and It's Use in Flash Memory Systems”的第 10/915,039 号美国专利申请案中描述的环形总线。

典型的控制器芯片 11 具有其自身的内部总线 23，内部总线 23 通过接口电路 25 与系统总线 13 介接。通常连接到总线的主要功能件是处理器 27（例如微处理器或微控制器）、含有用于初始化（“引导”）系统的代码的只读存储器（ROM）29、主要用于对存储器与主机之间传送的数据进行缓冲的随机存取存储器（RAM）31，以及对用于传递通过存储器与主机之间的控制器的数据的错误校正码（ECC）进行计算和检查的电路 33。控制器总线 23 通过电路 35 与主机系统介接，在图 2 的系统包含在存储器卡内的情况下，所述介接是通过卡的作为连接器 4 的一部分的外部接点 37 来完成的。时钟 39 与控制器 11 的每一其它组件连接并由其使用。

存储器芯片 15 以及任何其它与系统总线 13 连接的芯片通常含有组织成多个子阵列或平面的存储器单元阵列，为了简明起见而说明两个这种平面 41 和 43，但可替代使用更多平面，例如四个或八个这种平面。或者，可不将芯片 15 的存储器单元阵列划分为平面。然而当如此划分时，每个平面具有其自身的可彼此独立地操作的列控制电路 45 和 47。电路 45 和 47 从系统总线 13 的地址部分 19 接收其各自存储器单元阵列的地址，并对其进行解码以对各自位线 49 和 51 中的特定一者或一者以上进行寻址。响应于在地址总线 19 上接收的地址通过行控制电路 55 来寻址字线 53。源电压控制电路 57 和 59 也与各自的平面连接，如同 p 阱电压控制电路 61 和 63。如果存储器芯片 15 具有单个存储器单元阵列，且如果系统中存在两个或两个以上这种芯片，那么可类似于上述多平面芯片内的平面或子阵列来操作每个芯片的阵列。

数据通过与系统总线 13 的数据部分 17 连接的各自数据输入/输出电路 65 和 67 而传送进和传送出平面 41 和 43。电路 65 和 67 用于通过经由各自的列控制电路 45 和 47 连接到平面的线 69 和 71，来将数据编程到存储器单元中以及从其各自平面的存储器单元读取数据。

尽管控制器 11 控制存储器芯片 15 的操作来编程数据、读取数据、擦除和照管各种内务管理事件，但每个存储器芯片还含有某种控制电路，其执行来自控制器 11 的命令以执行这些功能。接口电路 73 连接到系统总线 13 的控制和状态部分 21。将来自控制器的命令提供到状态机 75，状态机 75 接着提供对其它电路的特定控制以便执行这些命令。控制线 77-81 连接状态机 75 与所述其它电路，如图 2 所示。来自状态机 75 的状态信息通过线 83 传送到接口 73，以用于通过总线部分 21 传输到控制器 11。

存储器单元阵列 41 和 43 的 NAND 结构是目前优选的，但也可替代使用例如 NOR 的其它结构。作为存储器系统的一部分的 NAND 快闪存储器及其操作的实例可参考第 5,570,315、5,774,397、6,046,935、6,373,746、6,456,528、6,522,580、6,771,536 和 6,781,877 号美国专利以及第 2003/0147278 号美国专利申请公开案。

由图 3 的电路图说明示范性 NAND 阵列，其为图 2 的存储器系统的存储器单元阵列 41 的一部分。提供大量全局位线，但为了简明解释起见，在图 2 中仅展示四个此类线 91-94。许多串联连接的存储器单元串 97-104 连接在这些位线之一与参考电位之间。使用存储器单元串 99 作为代表，多个电荷存储存储器单元 107-110 在串的任一端处与选择晶体管 111 和 112 串联连接。当使一个串的选择晶体管导电时，所述串连接在其位线与参考电位之间。接着对所述串内的一个存储器单元进行一次编程或读取。

图 3 的字线 115-118 在许多存储器单元串中每一串中的一个存储器单元的电荷存储元件上个别地延伸，且栅极 119 和 120 控制串的每一端处的选择晶体管的状态。使共享共用字线和控制栅极线 115-120 的存储器单元串形成可共同擦除的存储器单元区块 123。此单元区块含有可一次性物理擦除的最小数目的单元。一次对沿着字线 115-118 之一的一行存储器单元进行编程。通常，以预定次序对 NAND 阵列的行进行编程，在此情况下是以最靠近连接到接地或另一共用电位的串的末端的字线 118 的行开始。接着对沿着字线 117 的存储器单元行进行编程，如此在整个区块 123 中进行。最后对沿着字线 115 的行进行编程。

第二区块 125 是类似的，其存储器单元串连接到与第一区块 123 中的串所连接的相同全局位线，但具有不同的一组字线和控制栅极线。由行控制电路 55 将字线和控制栅极线驱动到其适当的操作电压。如果系统中存在一个以上平面或子阵列，例如图 2 的平面 1 和 2，那么一种存储器结构使用在其之间延伸的共用字线。替代地可存在共享共用字线的两个以上平面或子阵列。在其它存储器结构中，单独驱动个别平面或子阵列的字线。

如若干 NAND 专利以及上文引用的公开申请案中所述，可操作存储器系统以在每个电荷存储元件或区域中存储两个以上可检测的电荷电平，进而在每一者中存储一个以上数据位。存储器单元的电荷存储元件最通常是导电浮动栅极，但替代地可以是非导电介电电荷捕集材料，如第 2003/0109093 号美国专利申请公开案中所述。

图 4 概念性说明用作下文进一步描述中的实例的快闪存储器单元阵列 7 (图 1) 的组织。存储器单元的四个平面或子阵列 131-134 可在单个集成存储器单元芯片上、在两个芯片上 (每个芯片上两个平面) 或在四个单独芯片上。具体布置对下文的论述来说不重

要。当然，在系统中可存在其它数目的平面，例如 1、2、8、16 或更多。将平面个别地划分为图 4 中用矩形展示的存储器单元区块，例如位于各自平面 131-134 中的区块 137、138、139 和 140。每个平面中可存在数十个或数百个区块。如上所述，存储器单元区块是擦除的单位，是物理上可共同擦除的最小数目的存储器单元。然而为了增加的并行性，以较大的元区块为单位来操作区块。将来自每个平面的一个区块逻辑上链接在一起以形成元区块。展示四个区块 137-140 形成一个元区块 141。元区块内的所有单元通常共同擦除。用于形成元区块的区块无需限于其各自平面内的相同的相对位置，如由区块 145-148 组成的第二元区块 143 中所示。尽管优选在所有平面上延伸元区块以获得高系统性能，但可以在不同平面中动态形成具有一个、两个或三个区块中的任一种或全部的元区块的能力来操作存储器系统。这允许元区块的大小较接近地与一次编程操作中可用于存储的数据量匹配。

为了操作目的，又将个别区块划分为存储器单元页，如图 5 说明。例如区块 131-134 中每一者的存储器单元各划分为八个页 P0-P7。或者，每个区块内可存在 16 个、32 个或更多存储器单元页。页是在区块内数据编程和读取的单位，含有一次编程的最小数据量。在图 3 的 NAND 结构中，页由沿着区块内字线的存储器单元形成。然而为了增加存储器系统操作并行性，可将两个或两个以上区块内的所述页逻辑上链接成元页。图 5 中说明元页 151，其由来自四个区块 131-134 中每一者的一个物理页形成。元页 151 例如包含所述四个区块中每一者中的页 P2，但元页的页无需在区块的每一者内一定具有相同的相对位置。尽管优选在所有四个平面上并行编程和读取最大数据量以获得高系统性能，但也可操作存储器系统以在不同平面中的单独区块中形成具有一个、两个或三个页中的任一种或全部的元页。这允许编程和读取操作适应性地与可方便地并行处理的数据量匹配，且减少了元页的一部分保持未以数据编程的机会。

用于每次使用单独页和单独区块管理数据的多数存储器管理技术可同样应用于元页和元区块。类似地，使用元页和元区块的技术通常也可应用于单独区块和单独页。大体上，使用页和区块给出的实例将理解为可应用于元页和元区块。类似地，相对于元页和元区块给出的实例将理解为通常可应用于页和区块。

如图 5 所说明，由多个平面的物理页形成的元页含有沿着那些多个平面的字线行的存储器单元。并非同时对一个字线行中的所有单元编程，而是较常见以两个或两个以上交错群组对其交替编程，每个群组存储一个数据页（在单个区块中）或一个数据元页（在多个区块上）。通过每次对交替的存储器单元进行编程，无需对每个位线均提供包含数据

寄存器和读出放大器的外围电路单位，而是在邻近的位线之间时间共享。这节省了外围电路所需的衬底空间量，且允许以沿着行而增加的密度来封装存储器单元。另外，优选对沿着行的每个单元同时编程，以便最大化可从给定存储器系统得到的并行性。

参看图 3，通过沿着 NAND 串的至少一端提供两行选择晶体管（未图示）而不是所示的单个行，来最方便地完成将数据同时编程到沿着一行的每隔一个存储器单元中。一行的选择晶体管接着响应于一个控制信号将区块内的每隔一个串连接到其各自的位线，且另一行的选择晶体管响应于另一控制信号将介入的每隔一个串连接到其各自的位线。因此将两个数据页写入到每行存储器单元中。

每个逻辑页中的数据量通常是整数个的一个或一个以上数据扇区，每个扇区惯例上含有 512 个字节的数据。图 6 展示页或元页数据的两个扇区 153 和 155 的逻辑数据页。每个扇区通常含有所存储的用户或系统数据的 512 个字节的部分 157 以及用于涉及部分 157 中的数据或涉及其中存储有数据的物理页或区块的开销数据的另一数目的字节 159。开销数据的字节数通常为 16 个字节，从而对于扇区 153 和 155 中每一者为总共 528 个字节。开销部分 159 可含有在编程期间从数据部分 157 计算出的 ECC、其逻辑地址、区块已擦除和再编程次数的经历计数、一个或一个以上控制旗标、操作电压电平和/或类似物，加上从此开销数据 159 计算出的 ECC。或者，开销数据 159 或其一部分可存储在其它区块中的不同页中。随着存储器的并行性增加，元区块的数据存储容量增加且数据页和元页的大小也因此增加。数据页接着可含有两个以上数据扇区。由于数据页中的两个扇区且每个元页两个数据页，所以一个元页中存在四个扇区。每个元页因此存储 2048 个字节的数据。这是高度的并行性，且可甚至在行中存储器单元数目增加时进一步增加。出于此原因，延伸快闪存储器的宽度以便增加页和元页中的数据量。

上文指出的物理上较小的可再编程非易失性存储器卡和快闪驱动器是市售的，其具有 512 兆字节 (MB)、千兆字节 (GB)、2 GB 和 4 GB 的数据存储容量，且可能变得更高。图 7 说明主机与此大容量存储器系统之间的最常见的接口。主机处理由主机执行的应用程序软件或固件程序产生或使用的数据文件。字处理数据文件是一个实例，且计算机辅助设计 (CAD) 软件的绘图文件是另一实例，主要存在于例如 PC、膝上型计算机等一般计算机主机中。pdf 格式的文档也是这种文件。图片数字摄像机为存储在存储器卡上的每幅图产生一个数据文件。蜂窝式电话利用来自内部存储器卡上的文件的数据，例如电话簿。PDA 存储并使用若干不同的文件，例如地址文件、日历文件等。在任何这种应用中，存储器卡还可含有操作主机的软件。

存储器系统，尤其是实施在可移除卡中的存储器系统，可经由标准接口与不同主机通信。不同的主机可使用不同的接口与存储器系统通信。两类接口是使用具有共用逻辑地址空间的逻辑寻址系统的接口和使用基于文件的寻址系统的接口。

LBA 接口

图 7 说明主机与存储器系统之间的常用逻辑接口。连续的逻辑地址空间 161 足够大以为可存储在存储器系统中的所有数据提供地址。通常将主机地址空间划分为数据群集的增量。每个群集在给定主机系统中可经设计为含有许多数据扇区，大约在 4 与 64 个扇区之间是典型的。标准扇区含有 512 个字节的数据。

图 7 的实例中展示已经产生了三个文件 1、2 和 3。在主机系统上运行的应用程序产生每个文件作为一组有序数据，并通过唯一名称或其它参考对其进行识别。尚未分配给其它文件的足够可用的逻辑地址空间由主机指派给文件 1。文件 1 被展示为已经被指派有连续的可用逻辑地址范围。通常还出于特定目的而分配地址范围，例如针对主机操作软件的特定范围，接着为存储数据而避开这些特定范围，即使在主机向数据指派逻辑地址时仍未利用这些地址。

当主机稍后产生文件 2 时，主机类似地在逻辑地址空间 161 内指派两个不同范围的连续地址，如图 7 中所示。文件无需被指派有连续的逻辑地址，而是可为已经分配给其它文件的地址范围之间的地址片段。此实例接着展示主机所产生的又一文件 3 被分配有主机地址空间的先前未分配给文件 1 和 2 和其它数据的其它部分。在此实例中，文件 1、文件 2 和文件 3 全部指派给共用逻辑地址空间（逻辑地址空间 161）的部分。

主机通过维持文件分配表（FAT）来跟踪存储器逻辑地址空间，其中由主机指派给各个主机文件的逻辑地址得以维持。FAT 表通常存储在非易失性存储器中以及主机存储器中，且随着新文件被存储、其它文件被删除、文件被修改等等，FAT 表由主机频繁地更新。例如当主机文件被删除时，主机接着通过更新 FAT 表来对先前分配给被删除文件的逻辑地址解除分配，以展示它们现在可用于与其它数据文件一起使用。FAT 中使用的逻辑地址可称为逻辑区块地址（LBA），因此使用在对来自不同文件的数据为共用的逻辑地址空间上的此类逻辑寻址的接口可称为 LBA 接口。类似地，针对使用用于正在传送的数据的逻辑地址的通信的协议可视为 LBA 协议。

主机不关心其中存储器系统控制器选择用来存储文件的物理位置。典型的主机仅知道其逻辑地址空间和其已经分配给其各个文件的逻辑地址。另一方面，存储器系统通过典型的主机/卡接口，仅知道逻辑地址空间的数据已经被写入到的部分，但不知道分配给

特定主机文件的逻辑地址，或甚至不知道主机文件的数目。存储器系统控制器将主机所提供的用于存储或检索数据的逻辑地址转换成其中存储有主机数据的快闪存储器单元阵列内的唯一物理地址。区块 163 表示这些逻辑到物理地址转换的工作表，其由存储器系统控制器维持。

存储器系统控制器经编程以便以使系统的性能维持在高等级的方式将数据文件存储在存储器阵列 165 的区块和元区块内。在此说明中使用四个平面或子阵列。在由来自每个平面的区块形成的整个元区块上优选以系统允许的最大程度的并行性编程和读取数据。至少一个元区块 167 通常经分配作为用于存储由存储器控制器使用的操作固件和数据的保留区块。可分配另一元区块 169 或多个元区块以用于存储主机操作软件、主机 FAT 表等。保留物理存储空间的大部分以用于存储数据文件。然而存储器控制器不知道主机如何在其各个文件对象之间分配所接收的数据。存储器控制器从与主机交互作用中通常知道的全部内容就是由主机写入到特定逻辑地址的数据存储在由控制器的逻辑到物理地址表 163 维持的相应物理地址中。

在典型的存储器系统中，提供比必需的多几个的区块的存储容量，以存储地址空间 161 内的数据的量。可提供这些额外区块中的一者或一者以上作为用于代替在存储器使用寿命期间可能变为有缺陷的其它区块的冗余区块。个别元区块内含有的区块的逻辑分组通常可因为各种原因而改变，包含冗余区块对原始指派给元区块的有缺陷区块的代替。例如元区块 171 的一个或一个以上额外区块通常维持在经擦除的区块池 (block pool) 中。当主机将数据写入到存储器系统时，控制器将由主机指派的逻辑地址转换成擦除区块池中的元区块内的物理地址。未用于将数据存储在逻辑地址空间 161 内的其它元区块接着经擦除并指定作为经擦除的池区块以供在后续数据写入操作期间使用。

在原始存储的数据变过时，由新数据频繁覆写存储在特定主机逻辑地址的数据。作为响应，存储器系统控制器将新数据写入经擦除的区块，并接着针对那些逻辑地址改变逻辑到物理地址表以识别存储这那些逻辑地址处数据的新的物理区块。接着擦除含有那些逻辑地址处原始数据的区块并使其可用于新数据的存储。如果在写入开始时在来自擦除区块池的预擦除区块中没有足够的存储容量，那么这种擦除常常必须在当前数据写入操作可完成之前发生。这可能不利地影响系统数据编程速度。存储器控制器通常了解到仅在主机向其相同逻辑地址写入新数据时由主机使给定逻辑地址处的数据变过时。因此许多存储器区块可能存储此类无效数据历时一定时间。

区块和元区块的大小逐渐增加，以便有效地使用集成电路存储器芯片的面积。这导

致较大比例的个别数据写入存储的数据量小于元区块的存储容量，且在许多情况下甚至小于区块的存储容量。由于存储器系统控制器通常将新数据引导到经擦除的池元区块，因此这可导致元区块的若干部分未被填充。如果新数据是对存储在另一元区块中的某个数据的更新，那么还期望以逻辑地址次序将其余有效的数据元页从具有与新数据元页的逻辑地址连续的逻辑地址的所述另一元区块复制到新的元区块中。老的元区块可保持其它有效的数据元页。这随着时间过去而导致个别元区块的某些元页的数据变为过时且无效，并由写入到不同元区块的具有相同逻辑地址的新数据所替换。

为了维持足够的物理存储器空间来在整个逻辑地址空间 161 上存储数据，对此数据进行周期性压缩或合并（垃圾收集）。还希望以与其实际一样多的逻辑地址的相同次序维持元区块内的数据扇区，因为这使得读取连续逻辑地址中的数据更为有效。因此通常以此额外目的来执行数据压缩和垃圾收集。在接收部分区块数据更新时对存储器进行管理的一些方面以及元区块的使用描述于第 6,763,424 号美国专利中。

数据压缩通常涉及从元区块读取所有有效数据元页并将其写入到新的区块，其中忽略了在过程中具有无效数据的元页。还优选以与具有有效数据的元页中存储的数据的逻辑地址次序匹配的物理地址次序来布置所述元页。新的元区块中占据的元页的数目将小于老的元区块中占据的元页的数目，因为并不将含有无效数据的元页复制到新的元区块。接着擦除老的区块并使其可用于存储新数据。通过合并获得的具有容量的额外元页接着可用于存储其它数据。

在垃圾收集期间，从两个或两个以上元区块收集具有连续或近似连续的逻辑地址的有效数据的元页，并将其重写到另一元区块中，通常是经擦除区块池中的一个元区块。当从原始的两个或两个以上元区块复制了所有有效数据元页时，可将所述元区块擦除以供未来使用。

数据合并和垃圾收集花费时间且可能影响存储器系统的性能，尤其是在数据合并或垃圾收集需要在来自主机的命令可执行之前进行的情况下。此类操作通常由存储器系统控制器调度以尽可能在后台进行，但执行这些操作的需要可能导致控制器必须对主机给出忙状态信号，直到此操作完成为止。主机命令的执行可被延迟的实例是在经擦除区块池中没有足够的预擦除元区块来存储主机希望写入到存储器中的所有数据，且首先需要数据合并或垃圾收集来清除一个或一个以上有效数据元区块，所述元区块接着可被擦除。因此注意力集中于管理存储器的控制以便最小化此类中断。在以下美国专利申请案中描述许多此类技术：2003 年 12 月 30 日申请的标题为“Management of Non-Volatile Memory

Systems Having Large Erase Blocks”的第 10/749,831 号美国专利申请案；2003 年 12 月 30 日申请的标题为“Non-Volatile Memory and Method with Block Management System”的第 10/750,155 号美国专利申请案；2004 年 8 月 13 日申请的标题为“Non-Volatile Memory and Method with Memory Planes Alignment”的第 10/917,888 号美国专利申请案；2004 年 8 月 13 日申请的标题为“Non-Volatile Memory and Method with Phased Program Failure Handling”的第 10/917,889 号美国专利申请案；2004 年 8 月 13 日申请的标题为“Non-Volatile Memory and Method with Control Data Management”的第 10/917,725 号美国专利申请案（下文统称为“LBA 专利申请案”）。

有效控制具有非常大擦除区块的存储器阵列的操作的一个挑战是使给定写入操作期间存储的数据扇区数目与存储器区块的容量和边界匹配并对准。一种方法是以小于存储少于填充整个元区块的量的一定量数据所必需的区块的最大数目来配置用于存储来自主机的新数据的元区块。2003 年 12 月 30 日申请的标题为“Adaptive Metablocks”的第 10/749,189 号美国专利申请案中描述了自适应元区块的使用。2004 年 5 月 7 日申请的标题为“Data Run Programming”的第 11/016,271 号专利申请案中描述了数据区块之间边界的配合以及元区块之间的物理边界。

存储器控制还可使用来自主机存储在非易失性存储器中的 FAT 表的数据，以更有效地操作存储器系统。一种此类使用是了解主机何时通过对数据的逻辑地址解除分配来将数据识别为过时的。知道此情况允许存储器控制器在其通过主机向那些逻辑地址写入新数据而通常将知道情况之前调度含有此无效数据的区块的擦除。这描述于 2004 年 7 月 21 日申请的标题为“Method and Apparatus for Maintaining Data on Non-Volatile Memory Systems”的第 10/897,049 号美国专利申请案。其它技术包含监视主机向存储器写入新数据的型式，以便推断给定写入操作是否为单个文件，或在多个文件时推断文件之间的边界位于何处。2004 年 12 月 23 日申请的标题为“FAT Analysis for Optimized Sequential Cluster Management”的第 11/022,369 号美国专利申请案描述了此类技术的使用。

为了有效操作存储器系统，控制器需要尽其可能地了解由主机指派给其个别文件的数据的逻辑地址。接着控制器可将数据文件存储在单个元区块或元区块群组内，而不是当文件边界未知时散布在较大数目的元区块之间。结果是数据合并和垃圾收集操作的次数和复杂性降低。因此存储器系统的性能改进。但当主机/存储器接口包含如上所述的逻辑地址空间 161（图 7）时，存储控制器难以较多地了解主机数据文件结构。

参看图 8，以不同方式说明图 7 已经展示的典型逻辑地址主机/存储器接口。由主机

向主机产生的数据文件分配逻辑地址。存储器系统接着看到这些逻辑地址并将其映射到实际存储数据的存储器单元区块的物理地址中。

文件接口

主机与用于存储大量数据的存储器系统之间的基于文件的接口不使用逻辑地址空间。主机改为通过唯一文件 ID（或其它唯一参考）和所述文件内的数据单位（例如字节）的偏移地址在逻辑上寻址每个文件。此文件地址是直接给存储器系统控制器的，存储器系统控制器接着保存其自己的关于每个主机文件的数据物理上存储在何处的表。可用如上文相对于图 2 到图 6 所述的同一存储器系统来实施这种新的接口。与图 2 到图 6 中所述内容的主要不同之处是存储器系统与主机系统通信的方式。

图 9 中说明文件接口，其应与图 7 的 LBA 接口相比较。文件 1、2 和 3 中的每一者的识别以及图 9 的文件内的数据的偏移量被直接传递到存储器控制器。此逻辑地址信息接着由存储器控制器功能 173 转换成存储器 165 的元区块和元页的物理地址。

图 10 也说明文件接口，其应与图 8 的逻辑地址接口相比较。图 8 的逻辑地址空间和主机维持的 FAT 表在图 10 中不存在。而是通过文件编号和所述文件内的数据的偏移量向存储器系统识别由主机产生的数据文件。存储器系统接着直接将所述文件映射到存储器单元阵列的物理区块。

由于存储器系统知道构成每个文件的数据的位置，因此可在主机删除文件之后随即擦除这些数据。这对于典型的逻辑地址接口来说是不可能的。此外，通过文件而不是使用逻辑地址来识别主机数据，存储器系统控制器可以减少进行频繁数据合并和垃圾收集的需要的存储数据。数据复制操作的频率和所复制的数据量因此显著减少，进而增加存储器系统的数据编程和读取性能。

基于文件的接口的实例包含使用直接数据文件存储的接口。以 Alan W. Sinclair 个人名义或与 Peter J. Smith 一起在 2005 年 2 月 16 日申请的第 11/060,174、11/060,248 和 11/060,249 号待决美国专利申请案以及 Alan W. Sinclair 和 Barry Wright 申请的标题为“Direct Data File Storage in Flash Memories”的第 60/705,388 号临时申请案（下文中统称为“直接数据文件存储申请案”）中描述了直接数据文件存储存储器系统。

由于图 9 和图 10 说明的这些直接数据文件存储申请案的直接数据文件接口比图 7 和图 8 说明的上述逻辑地址空间接口简单，且允许存储器系统更好地执行，因此直接数据文件存储对许多应用来说是优选的。但主机系统目前主要配置成以 LBA 接口操作，因此具有直接数据文件接口的存储器系统与多数主机不兼容。因此需要向存储器系统提供以

任一种接口操作的能力。

双接口

对于一些存储器系统，尤其是包含在可与不同主机介接的可移除存储器卡中的那些存储器系统，向后兼容性是个重要问题。许多主机系统使用与图 7 和图 8 所示类似的基于扇区的存储的形式，且这些主机系统中的一些可能无法容易地适合于以例如图 9 和图 10 所示的基于文件的存储进行操作。因此，需要具有可与使用逻辑寻址接口或基于文件的接口的主机介接的存储器系统。插入在 LBA 接口与基于文件的后端之间的 LBA 协议适配器可允许使用逻辑寻址的主机将数据存储在与将数据作为文件来管理的存储器阵列中。

2005 年 8 月 3 日申请的标题为“Interfacing systems operating through a logical address space and on a direct data file basis”的第 11/196,869 号美国专利申请案描述了使存储器系统能够与使用逻辑寻址接口或基于文件的接口的主机介接的系统。图 11 说明此系统。此实例在存储器系统内组合了图 7 的主机操作与图 9 的基于文件的存储器操作，加上一个添加的地址转换 172。地址转换 172 将存储器空间 161 上的逻辑地址群组映射到经修改地址空间 161'上展示的个别逻辑文件 a-j 内。优选将整个逻辑地址空间 161 划分为这些逻辑文件，因此逻辑文件的数目取决于逻辑地址空间的大小以及个别逻辑文件的大小。每个逻辑文件均含有空间 161 上连续逻辑地址的群组的数据。优选使每个逻辑文件内的数据量相同，且所述量等于存储器 165 中一个元区块的数据存储容量。逻辑文件的不相等的大小和/或与存储器的区块或元区块的存储容量不同的大小当然是可能的，但不是优选的。

个别文件 a-j 中每一者内的数据由文件内的逻辑偏移地址表示。逻辑文件的文件识别符和数据偏移量在 173 处转换为存储器 165 内的物理地址。逻辑文件 a-j 通过直接数据文件存储申请案中描述的过程和协议直接存储在存储器 165 中。所述过程与用于在存储器 165 中存储图 9 的数据文件 1-3 的过程相同，不同的是每个逻辑文件中的已知数据量使得此过程较容易，尤其是在所述量等于存储器的区块或元区块的容量的情况下。图 11 中展示将逻辑文件 a-j 中的每一者映射到存储器 165 的元区块中的不同一者。还期望基于文件的数据存储以与主机经设计以介接的当前逻辑地址存储器系统相同或等效的方式与主机交互作用。通过将个别逻辑文件映射到相应的个别存储器元区块内，本质上以直接数据文件接口存储器系统实现了与使用逻辑地址空间接口时相同的性能和时序特性。

图 12 以不同方式说明图 11 的方法。图 12 与图 8 的逻辑地址存储器系统操作相同，但具有将逻辑地址空间划分为逻辑文件（刚描述的图 11 的步骤 172）的增加的功能。另外，图 12 的“将文件数据映射到物理存储区块的表”替代了图 8 的“将逻辑地址映射到物理存储区块的表”。逻辑地址到逻辑文件转换 172 可视为位于使用 LBA 系统的接口与基于文件的后端之间的 LBA 协议适配器的一部分。

经设计以通过传统逻辑地址空间接口与主机一起工作的图 11-12 的基于数据文件的后端存储系统也可添加有直接数据文件接口。来自文件接口的主机数据文件和来自逻辑接口的逻辑文件均转译为存储器元区块地址。接着通过执行直接数据文件协议将数据存储在存储器的那些地址中。此协议包含先前列出的直接数据文件存储申请案的直接数据文件存储技术。

通过提供具有两种类型主机接口的便携式存储器卡或快闪驱动器或其它形式的可移除存储器系统，存储器可与以逻辑地址空间接口操作的最新近主机一起使用，与将其文件直接介接到存储器的主机一起使用，或可在两种类型主机之间交换。这允许具有较新的基于文件的接口的主机的用户以其最有效的方式使用存储器，同时具有与传统逻辑地址空间接口的向后兼容性。而且，由于相同的一对一逻辑文件到元区块的映射而实现本质上相同的性能和时序特性。具有双主机接口的存储器允许用户获得所述存储器的较新的直接数据文件接口，同时其仍可用于具有传统逻辑地址空间接口的主机的扩展安装的基底。其提供了从当前接口迁移到直接数据文件接口的方法。

图 13 说明具有双主机接口的存储器系统 300。存储器系统存储由主机通过文件接口 307 直接供应的主机数据文件（HF1、HF2...HF_n）以及来自 LBA 接口 305 的由 LBA 协议适配器 301 转换的逻辑文件（LFa、LFb...LF_m）。基于文件的后端 303 不需要区别逻辑文件与主机文件，而是优选经优化以处理两种类型的文件。如此，逻辑文件等同于 LBA 专利申请案中描述的系统的逻辑群组，且从主机接口的角度来看，存储器系统 300 的性能因此与 LBA 专利申请案中所述的具有逻辑地址空间接口的系统的性能相匹配。

除了将主机数据从逻辑寻址格式转换为逻辑文件格式以外，LBA 协议适配器 301 可响应于特定条件或从主机接收的特定 LBA 命令而产生与基于文件的后端 303 兼容的命令。响应于此些条件而产生用于直接数据文件后端的命令的实例在下表中展示。

条件	所产生的直接数据文件命令
LBA 命令的开始	写入_指针 <文件ID> <偏移量> 此命令设置当前文件内的偏移地址，下一写入命令将在该地址处操作。
当前文件未打开 且 打开_文件 = 最大	关闭 <文件ID> + 闲置 + 暂停 <直到_不_忙> 此命令群组关闭最不新近存取的文件，并促使装置执行所有待决的对文件的垃圾收集操作。
当前文件未打开	打开 <文件ID> + 写入_指针 <文件ID> <偏移量> 此命令群组打开当前文件，并设置偏移地址，下一写入命令将在该处起作用。
当前文件已改变	写入 <文件ID> + 流 + 写入_指针 <文件ID> <偏移量> 此命令群组写入累积的前一文件的数据，并设置当前文件内的偏移地址，下一写入命令将在该处操作。
程序区块已满	写入 <文件ID> + 流 此命令群组写入累积的当前文件的数据。
程序区块已满 且 有效_页 > 最小值	关闭 <文件ID> + 闲置 + 暂停 <直到_不_忙> 此命令群组关闭当前文件，并促使装置执行所有待决的对文件的垃圾收集操作。
LBA 命令的结束	写入 <文件ID> + 流 + 保存_缓冲 <文件ID> 此命令群组写入累积的当前文件的数据，并促使保留在缓冲器中的未编程数据编程到快闪存储器中的“交换区块”。

对象接口

各种基于文件的接口可用于在电子装置之间传送数据。一些协议提供具有预定大小以及所述大小的指示的文件。文件的大小通常在此系统中保持不变，使得此系统可能不适于需要对文件进行编辑的应用。然而，对于从一个装置向另一装置传送文件，所述协议可能有利，且可允许高等级的安全性。大小的指示通常在发送文件数据之前发送。传送具有文件大小指示符的预定大小的文件的协议可视为对象协议。一种对象协议是微软公司的图片传送协议（PTP）。另一种此协议也是微软公司的媒体传送协议，也称为媒体传输协议（MTP）。对象协议尤其适于发送含有预定数据量的文件，例如数字照片或 MP3 音乐文件。举例来说，所述协议可用于在数字相机与 PC 之间传送数字照片，或将 MP3 音乐文件从 PC 传送到 MP3 播放器。

媒体传送协议（MTP）提供支持预定大小的文件对象传送的对象接口。其主要目的是允许可临时连接在一起的装置之间的通信，其中每一装置具有很大的存储容量。接口允许装置之间二进制数据对象的交换，以及一个装置接另一装置的内容的列举。下文列出 MTP 接口的某些特性。然而，其它对象接口可具有不同的特性。在来自微软公司的标题为“Media Transfer Protocol Enhanced”的文档中提供了对 MTP 的较完全的描述。

1. 通信协议

1.1 起始器和响应器：每次仅在两个装置之间发生交换，其中一个装置充当起始器且另一装置充当响应器。起始器是通过发送操作起始与响应器的动作的装置。响应器不起始任何动作，并对起始器发送的操作发送响应。充当起始器的装置应当能够列举并理解响应装置的内容，且控制协议中的操作流程。起始器通常是比响应器更强大的装置。响应器的实例是简单的内容产生装置（例如数码相机）以及简单的内容输出装置（例如便携式音频播放器）。

1.2 会话：会话是其中在起始器与响应器之间维持连接的通信状态。会话提供了用以参考对象的上下文，并保证在不向起始器发出报警的情况下响应器装置状态不会改变。会话由起始器打开，且由起始器或响应器关闭。装置可同时维持多个打开的会话。起始器在会话第一次打开时向会话指派唯一的识别符，并在发送操作时使用识别符来识别会话。

1.3 事务：从起始器中出现的任何动作在事务中执行，事务是向动作调用机制提供输入参数、二进制数据交换以及具有参数的响应的标准阶段序列。每个阶段内的数据流是单向的。在操作的起始期间，数据仅从起始器流动到响应器。在对所请求操作的响应期间，数据仅从响应器流动到起始器。在二进制数据交换阶段期间，数据可在任一方向上流动，但从不在两个方向上流动。必须通过多个操作来执行双向二进制数据交换。起始器向每个事务指派一个识别符。将经定义的识别符指派给在会话中起始的第一操作，且针对每个连续的事务将识别符递增 1。

事务由多达三个阶段组成：操作请求阶段、数据阶段以及响应阶段。操作请求阶段和响应阶段共享相同的识别符，且可选的数据阶段在需要时存在于另外两个阶段之间。

操作请求阶段由操作请求数据集（其识别由起始器调用的操作）的传输、其中其应被执行的上下文（会话和事务）以及有限的一组参数组成。

可选的数据阶段跟随操作请求阶段。其存在由操作请求阶段中定义的操作决定。数据可为在协议内定义的透明数据集，或可为经交换以存储在接收装置上的二进制数据。数据阶段中实际的数据传输可涉及以容器格式发送数据，或将数据分为若干个包，如正在使用中的特定传输所需。

在响应阶段中，将固定数据集从响应器传输到起始器以报告关于先前事务的信息，例如成功/失败。

1.4 事件：事件主要由响应器发送作为以前摄方式传输信息或警报的方式。不同于

操作，不需要对事件进行应答或动作。事件需要与数据传输或操作事务异步通信。传输应定义事件可在事务期间与数据流交错的过程。

1.5 同步和异步事务：通信协议中的所有事务都是同步的，即在前一操作完全完成之前无法起始新的操作。可通过在操作在装置上的后台中执行的同时由响应器发送的事件，通过将操作分为起始（其使操作开始）以及进程监视来模拟异步操作。如果在异步操作在处理中时尝试新的操作，那么响应器以忙失败状态响应，且起始器应稍后再次尝试。

2. 信息数据集

当交换数据收集时，其以称为数据集的预定义结构被收集。存在三种信息数据集，其可通过使用适当的操作来存取。

2.1 装置信息数据集：装置信息数据集提供对装置的描述，且大部分是静态的。

2.2 对象信息数据集：对象信息数据集提供对对象的核心属性的综述。这些核心属性包含对象的数据分量的大小。属性还包含关联性，其可用于使数据对象相关联，且描述装置上的分级文件系统。可不依赖于专用于特定文件系统的任何路径或命名惯例来表示装置上的文件分级。对象的属性还可在对象属性数据集中检索。

2.3 存储装置信息数据集：存储装置信息数据集描述装置中包含的存储装置。描述包含最大容量和仍有待写入的自由空间两者，且可包含使用中的文件命名惯例或目录结构惯例。

3. 属性

3.1 装置属性：装置属性识别装置的设置或状态条件，且不链接到装置上的任何数据对象。装置属性可为只读或读写的。属性包含在装置属性描述数据集中，且可通过使用适当的操作来设置或检索。

3.2 对象属性：对象属性提供用于交换与对象本身分离的对象描述元数据的机制。对象属性的主要益处是允许对大存储装置的快速列举，而不论文件系统如何。对象属性提供关于装置上的对象的信息，且指定其可包含的值。属性包含在对象属性数据集中，且可通过使用适当的操作来设置或检索。

4. 对象句柄

对象句柄是提供对装置上的逻辑对象的一致参考（在会话中是唯一的）的识别符。对象句柄的值没有特定的意义。响应器响应于来自起始器的打开会话操作而为装置内的对象产生对象句柄阵列。起始器借助于获得对象句柄操作来获取对象句柄，这促使响应

器向起始器发送对象句柄阵列。当起始器使用发送对象信息操作来定义待发送的对象时，响应器装置分配对象句柄，并在操作的响应阶段中将其返回到起始器。当会话关闭时，所有的对象句柄均失效，且必须由起始器重新获取。装置可保持相同的对象句柄或可改变对象句柄以用于下一会话。

5. 对象参考

因为对象接口是独立于文件系统的协议，所以无法通过嵌入文件名在对象之间形成复杂的链接。已定义一个抽象的参考机制以允许任意的对象参考。参考是单向的，且在不检查对装置中所有对象的所有参考的情况下无法确定哪些对象参考给定的对象。可通过使用适当的操作来设置或检索参考。文件句柄参考的对象在会话之间必须一致。对删除的对象的参考不能错误地参考另一对象。任何对象句柄都不应再次使用，或者装置应对一个对象的所有参考连同对象一起删除。

6. 操作和响应

操作定义在事务内在起始器与响应器之间发生的通信。响应器对起始操作起作用所需的信息可作为参数在操作请求上传递。可发送 5 个参数。额外的信息也可在事务的数据阶段中以预定义数据集传递。在每次操作之后，响应器返回具有多达 5 个参数的响应数据集以及指示操作结果的响应码。指定大量的操作。操作序列的实例是跟随有“发送对象”操作的“发送对象信息”操作，起始器通过此操作向响应器发送对象，以及跟随有“获得对象”操作的“获得对象信息”操作，起始器通过此操作接收来自响应器的对象。

图 14A 展示起始器 410 与响应器 412 之间的事务的实例。起始器 410 可为 PC，且响应器 412 可为 MP3 音乐播放器或数字相机。起始器 410 和响应器 412 经连接以允许例如通过 USB 电缆进行通信。首先，在操作请求阶段中，起始器 410 识别作为“发送对象信息”操作而调用的操作。第二，在数据阶段中，起始器 410 向响应器 412 发送对象信息 414。对象信息 414 是关于将由起始器 410 发送到响应器 412 的特定对象的信息。对象信息 414 在其涉及的对象被发送之前由起始器 410 发送。对象信息可包含关于对象的多种信息，包含对象的大小。在将发送的对象是 MP3 音乐文件的情况下，MP3 音乐文件的大小作为对象信息的一部分发送。第三，在响应阶段中（在对象信息 414 由响应器 412 接收之后），响应器 412 指示已接收到对象信息 414。事务可在此时结束。。

图 14B 展示跟随图 14A 的事务且作为起始器 410 与响应器 412 之间的同一会话的一部分的第二事务。第二事务是“发送对象”事务。首先，在操作请求阶段中，起始器 410

识别作为“发送对象”操作调用的操作。第二，在数据阶段中，起始器向响应器 412 发送对象 416。对象 416 可为 MP3 音乐文件或具有例如 JPEG、GIF 或位图格式的数字照片。对象 416 的对象信息 414（包含文件大小）已经由起始器 410 发送到响应器 412，如相对于图 14A 所述。第三，在响应阶段中（在接收对象 416 之后），响应器 412 向起始器 410 指示已接收到对象 416。事务可在此时结束。

对象协议适配器

在本发明的实施例中，在存储器系统中提供对象协议适配器，其允许使用对象协议的主机与使用如上所述基于文件的后端的存储器系统交接。对象协议适配器根据对象协议通过对象接口接收数据和命令，并在向后端发送数据和命令之前执行适当的转译。在一个实例中，主机使用 MTP 来与存储器系统交接，且存储器系统使用直接数据文件后端来存储数据。对象协议适配器对在主机接口与后端之间的命令与数据两者执行适当的转换。

图 15 展示插入在对象接口 520 与基于文件的后端 522 之间的对象协议适配器的实例。对象接口 520 是用于预定大小的文件对象（例如 MTP 或 PTP 对象）的接口。对象的大小在对象由主机发送之前发送。对象协议适配器 524 管理协议，在所述协议内存储器系统 526 与主机通信。对象协议适配器 524 还管理用于与主机的信息交换的事务并执行事务。事务是根据基于文件的协议在对象协议适配器与基于文件的后端之间执行的。

对象协议适配器 524 在新的文件对象由主机发送（根据对象协议）时通过向基于文件的后端 522 发送适当的命令来管理文件的打开和关闭。特定来说，因为文件在对象协议中具有预定大小，所以对象协议适配器 524 负责在已接收到预定量的文件数据时关闭文件。使用对象协议的主机通常将不发送单独的文件结束指示符，因此对象协议适配器 524 在已接收到整个文件对象时产生文件结束指示符并将其发送到基于文件的后端 522。以此方式，当存储器系统 526 接收到来自使用对象协议的主机的多个文件时，基于文件的后端 522 不会保持文件打开，直到其达到某个最大数目的打开文件为止（如直接数据文件后端中通常发生）。替代地，每个文件作为完整的对象而接收，且基于文件的后端在所述完整的文件被接收之后接收来自对象协议适配器 524 的命令，使得基于文件的后端关闭文件。这减少了维持许多打开文件的负担。一旦文件关闭就可调度文件以进行垃圾收集。

对象协议适配器可管理存储器系统的状态。特定来说，对于具有直接数据文件后端的存储器系统，定义三种状态，其允许存储器系统响应于主机命令而改变其操作。这三

种状态是：“闲置”、“备用”和“关闭”。响应于来自主机的相应状态命令来起始这三种状态，其中主机使用直接数据文件命令集或等效物。在主机使用对象协议的情况下，对象协议适配器可产生状态命令。如果主机使用包含所述命令的对象协议，那么对象协议适配器可响应于从主机接收的等效命令来向基于文件的后端发送状态命令。或者，对象协议适配器可基于从其它因素推断主机的状态来产生状态命令。举例来说，主机可给出某个指示：其在一段时间内将不会移除功率，且作为响应，对象协议适配器可向基于文件的后端发送“闲置”状态命令。类似地，主机可指示其将移除功率，或对象协议适配器可基于主机行为而推断出将移除功率，且作为响应，对象协议适配器可向基于文件的后端发送“关闭”命令。

对象协议适配器的主要功能之一是将来自从主机接收的预定义文件的主机数据转换（根据对象协议）为用于基于文件的后端的流式数据。当使用对象协议的主机（充当 MTP 中的起始器）发送具有预定义大小的文件且要求来自响应器的响应时，基于文件的后端通常不会配置成提供此响应。特定来说，在使用直接数据文件后端的情况下，数据通常被流化，且没有 MTP 的响应阶段的等效物。对象协议适配器将来自主机的对象转换为流式数据，并在已接收到对象的所有数据时产生对主机的适当响应。

在例如 MTP 的对象协议中，读取或写入对象的操作之前是用于传送关于所述对象的信息的单独操作。此信息传送操作是在正写入的对象的情况下针对对象属性（对象信息）的“设置”操作。此信息传送操作在正读取的对象的情况下是针对对象属性的“获得”操作。对象属性具有包含对象长度的数据集的形式。

当对象正写入图 15 的存储器系统 526 时，对象的数据集首先由对象协议适配器 524 从主机接收。在 MTP 项中主机被视为起始器。数据集中的信息用于控制实施写入操作的后续事务。数据集还作为元数据存储于存储器系统 526 中。在写入操作期间，对象协议适配器 524 计数在数据写入操作阶段期间从起始器传送的数据量。对象协议适配器 524 从此计数中识别对象数据的结束。对象协议适配器接着产生发送到主机的响应。对象协议适配器 524 还向基于文件的后端 522 发送命令以关闭文件。

当正从图 15 的存储器系统 526 读取对象时，对象的元数据首先由对象协议适配器 524 从基于文件的后端 522 获得。对象的元数据接着用于控制实施读取操作的后续事务。对象协议适配器 524 还将存储在存储器系统 526 中的元数据发送到主机作为用于对象的数据集。对象协议适配器 524 计数在数据读取操作阶段期间从基于文件的后端 522 传送的数据量。对象协议适配器 524 从此计数中识别数据的结束。对象协议适配器 524 在对

象已发送之后接着产生发送到主机的响应。

对象协议适配器在用于例如 MTP 的对象协议的对象信息数据集与存储在例如使用直接数据文件后端的那些存储器系统的基于文件的存储器系统中的元数据之间进行转译。“元数据”是用于指代关于对象的数据的术语，所述数据与对象分离地存储且单独管理。因此，MTP 中的对象信息数据集是元数据的一个实例。直接数据文件后端可以与对象协议所使用的不同格式或相同格式来存储元数据。术语“文件_信息”也用于直接数据文件系统中的元数据。当对象协议适配器 524 接收到来自主机的元数据相关命令时，对象协议适配器 524 可将命令转译为与基于文件的后端 522 兼容的格式。然而在一些情况下，不需要任何转译，因为用于对象协议中的元数据的命令与基于文件的后端 522 兼容。

元数据是由主机产生的与文件相关联的信息。元数据的特征和内容由主机确定，且其通常不会由存储文件和元数据的装置解译。元数据命令用于起始针对由直接数据文件后端存储的指定文件的元数据输入和输出操作，且定义元数据内的偏移地址值。元数据命令可在从主机接收到关于 MTP 数据集的相应命令时由对象协议适配器产生。由直接数据文件后端系统使用的元数据命令的实例在下表中展示。

命令	参数	描述
元数据_写入	<文件 ID>	将指定文件的元数据写入由元数据_写入_指针的当前值界定的偏移地址
元数据_读取	<文件 ID>	在由元数据_写入_指针的当前值界定的偏移地址处读取指定文件的元数据
元数据_写入_指针	<文件 ID> <偏移量>	为指定文件的元数据_写入_指针界定新的当前值
元数据_读取_指针	<文件 ID> <偏移量>	为指定文件的元数据_读取_指针界定新的当前值

元数据_写入：在元数据_写入命令的接收之后流化到装置的元数据覆写在由元数据_写入_指针的当前值界定的偏移地址处指定文件的元数据。指定文件的元数据的内容和长度由主机确定。元数据_写入命令通过接收任何其它命令来终止。

元数据_读取：在由元数据_读取_指针的当前值界定的偏移地址处的指定文件的元数据可在接收元数据_读取命令之后从装置流化。当达到元数据的末尾时，元数据流化终止，且此条件可由主机借助于状态命令来识别。元数据_读取命令通过接收任何其它命令来终止。

元数据_写入_指针：元数据_写入_指针命令将指定文件的元数据_写入_指针设置到指定的偏移地址。元数据_写入_指针在元数据跟随元数据_写入命令流化到装置时由装置递增。

元数据_读取_指针：元数据_读取_指针命令将指定文件的元数据_读取_指针设置到指定的偏移地址。元数据_读取_指针在元数据跟随元数据_读取命令从装置流化时由装置递增。

在一些情况下，主机可具有对象的分级布置。举例来说，文件可存储在目录和子目录中。直接数据文件后端通常存储没有任何分级结构（即，处于逻辑上平坦的布置）的文件。为了调和两个系统，对象协议适配器可通过使用与对象相关联的元数据重新产生所存储对象的分级结构。在主机维持分级的情况下，关于分级中文件的状态的信息在文件存储时作为元数据存储。当主机存取存储器系统时，可首先读取元数据，使得对象协议适配器可确定分级结构并可在此信息返回给主机。以此方式，即使文件在不考虑此分级信息的情况下存储在存储器系统中，对象协议适配器也可重新产生分级信息。举例来说，文件位于主机的分级中的目录和子目录可在文件存储时作为元数据存储。稍后当主机尝试存取存储器系统的内容时，在返回到主机的信息中反映目录和子目录信息。

多协议接口

在本发明的实施例中，提供存储器系统，其可接收和存储具有预定大小（例如根据MTP协议）的对象、没有预定大小的作为经流化文件接收的文件，以及具有在为存储器系统界定的逻辑地址空间上界定的逻辑地址的数据扇区。对应于这三种协议来配置协议适配器，且根据主机使用的协议来选择协议适配器。

图16展示具有连接到共用直接数据文件接口636的三个协议适配器632、632、634的存储器系统629。直接数据文件接口636和直接数据文件存储装置638在基于文件的后端640中，所述后端640可视为直接数据文件后端。因此，存储器系统629使用用于从所述文件接口的任一者接收的数据的共用后端。在此系统中存储器的分区可能不是必要的，从而有效地使用存储器阵列中的可用空间。因此，阵列中的区块可用于存储在不同时间从文件接口、对象接口和LBA接口接收的数据。然而，从LBA接口接收的数据可存储在在该时间仅存储从LBA接口接收的数据的区块中。以通常反映其文件结构的方式管理从文件接口和对象接口接收的数据，从而减少垃圾收集。

图16展示通过LBA协议适配器634连接到基于文件的后端640的LBA接口639。LBA协议适配器634可将LBA数据转换为逻辑文件，或使用某种其它方式将LBA数据转换为用于由基于文件的后端接收的合适的格式。

对象接口642通过对象协议适配器632连接到基于文件的后端640。大体上，对象协议适配器632将来自一个协议的数据和命令转换到另一协议，以允许使用对象协议的

主机存取基于文件的后端。尽管这可仅涉及从对象协议命令到基于文件的命令的一对一转译，但在一些情况下，命令在另一协议中不具有等效物。在所述情况下，对象协议适配器可能做出比简单转译更多的事。举例来说，在 MTP 主机发送包含对象大小的元数据并接着发送对象的情况下，对象协议适配器识别对象的结束并产生对 MTP 主机的响应。对象协议适配器在此时还对基于文件的后端产生关闭文件命令。

文件接口 644 通过文件协议适配器 630 连接到基于文件的后端 640。在一些情况下，使用适当的基于文件的协议的主机可与基于文件的后端直接通信，而不必需任何转译。然而在其它情况下，主机可使用基于文件的但与基于文件的后端的协议不同的协议来发送文件。在此情况下，文件协议适配器 630 执行任何必要的转译。

图 16 的存储器系统与使用至少三种不同协议的主机兼容。如果提供额外的协议适配器，那么存储器系统也可与其它主机兼容。

图 17 展示图 16 的存储器系统 629，其中文件协议适配器 630、对象协议适配器 632 和 LBA 协议适配器 634 共同视为转译层 750。每个协议适配器提供在特定主机协议与基于文件的后端 640 之间所需的转译。在一些情况下，存储器系统 629 可每次与一个以上主机通信。举例来说，存储器系统 629 可连接到附接有多个主机的网络。在其它实例中，单个主机可具有表现为它们好像是不同主机的运行的不同应用程序，且可使用不同的协议与存储器系统通信。在这些情况下，转译层 750 必须解决不同主机之间的任何冲突。转译层 750 提供不同主机之间的仲裁，使得基于文件的后端 640 不接收冲突命令。这可能意味着拒绝一个主机存取，直到另一主机完成特定任务为止。举例来说，如果对象协议适配器 632 正在传送对象或元数据，那么转译层 750 可防止 LBA 协议适配器 634 和文件协议适配器 630 与基于文件的后端 640 通信，直到操作完成为止。这可能意味着向尝试经由 LBA 接口 639 或文件接口 634 存取存储器系统的主机发送忙信号。在一些实例中，不同的主机可能能够通过交错的事务来存取基于文件的后端 640，使得其具有相同时间周期中的存取。在这些情况下，转译层 750 在主机之间进行仲裁。

图 17 展示含有文件接口 644、对象接口 642 和 LBA 接口 639 的接口层 752。文件接口 644、对象接口 642 和 LBA 接口 639 展示为相异元件，然而这是展示存储器系统 629 的逻辑接口的逻辑表示，且可能不一定存在三种单独的物理接口。在一些实例中，单个物理接口（例如 USB 连接器或 SD 连接器）对于文件接口 644、对象接口 642 和 LBA 接口 639 以及使用的任何其它接口来说是共用的。使用哪种接口且因此使用哪种协议适配器取决于主机使用的协议。在一些情况下，当存储器系统第一次连接到主机时，主机可

将主机协议作为握手例行程序的一部分来指示。在其它情况下，存储器系统 629 从主机发送的命令中推断主机协议。接口层 752 可从主机发送的指示或以某种其它方式检测所使用的主机协议，并根据主机协议选择适当的通信接口。因此，在存储器系统检测到主机正使用 MTP 协议时，选择对象接口 642 和对象协议适配器 632。大体上，在任何一个时间仅选择一个协议适配器。然而在一些情况下，可交替选择协议适配器以允许两个或两个以上主机的交错存取。

应了解，图 17 的元件对应于存储器系统 629 的逻辑组件，且不一定对应于单独的物理元件。因此可通过专用电路来实现或可使用在控制器上操作的适当固件来实现接口层 752 和转译层 750 的功能。在一个实例中，单个物理接口由存储器控制器管理，存储器控制器选择在存储器控制器上运行的协议适配器以将数据和命令转换为与共用后端兼容的格式。尽管图 17 仅展示三个协议适配器，但真正的存储器系统可能具有多于或少于三个协议适配器。在一些情况下可能存在一个以上对象协议适配器。举例来说，可能存在用于 MTP 的对象协议适配器以及用于另一对象协议（例如，PTP）的单独对象协议。类似地，可能存在用于不同文件协议的一个以上文件协议适配器，以及用于不同 LBA 协议的不同 LBA 协议适配器。举例来说，Alan W. Sinclair 的标题为“Logically-Addressed File Storage Methods”的第 11,302,764 号专利申请案提供了替代性 LBA 协议的实例。

结论

尽管已相对于本发明的示范性实施例描述了本发明的各个方面，但应了解本发明受到所附权利要求书的完整范围内的保护。

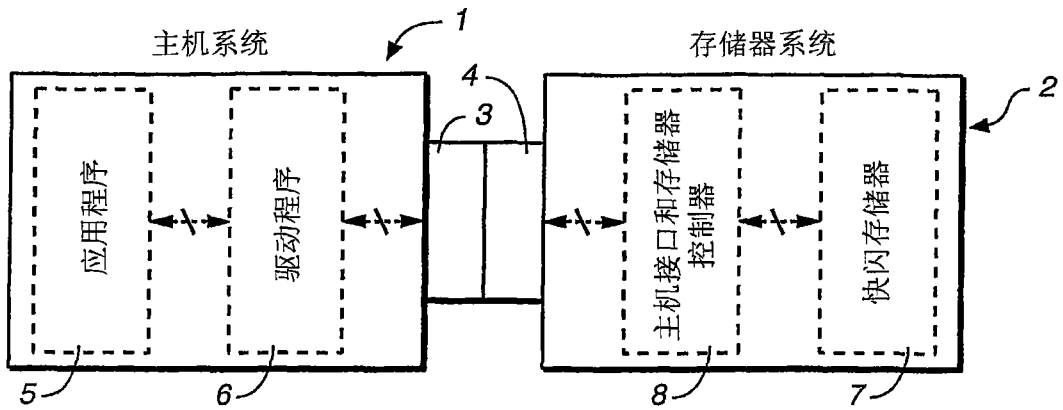
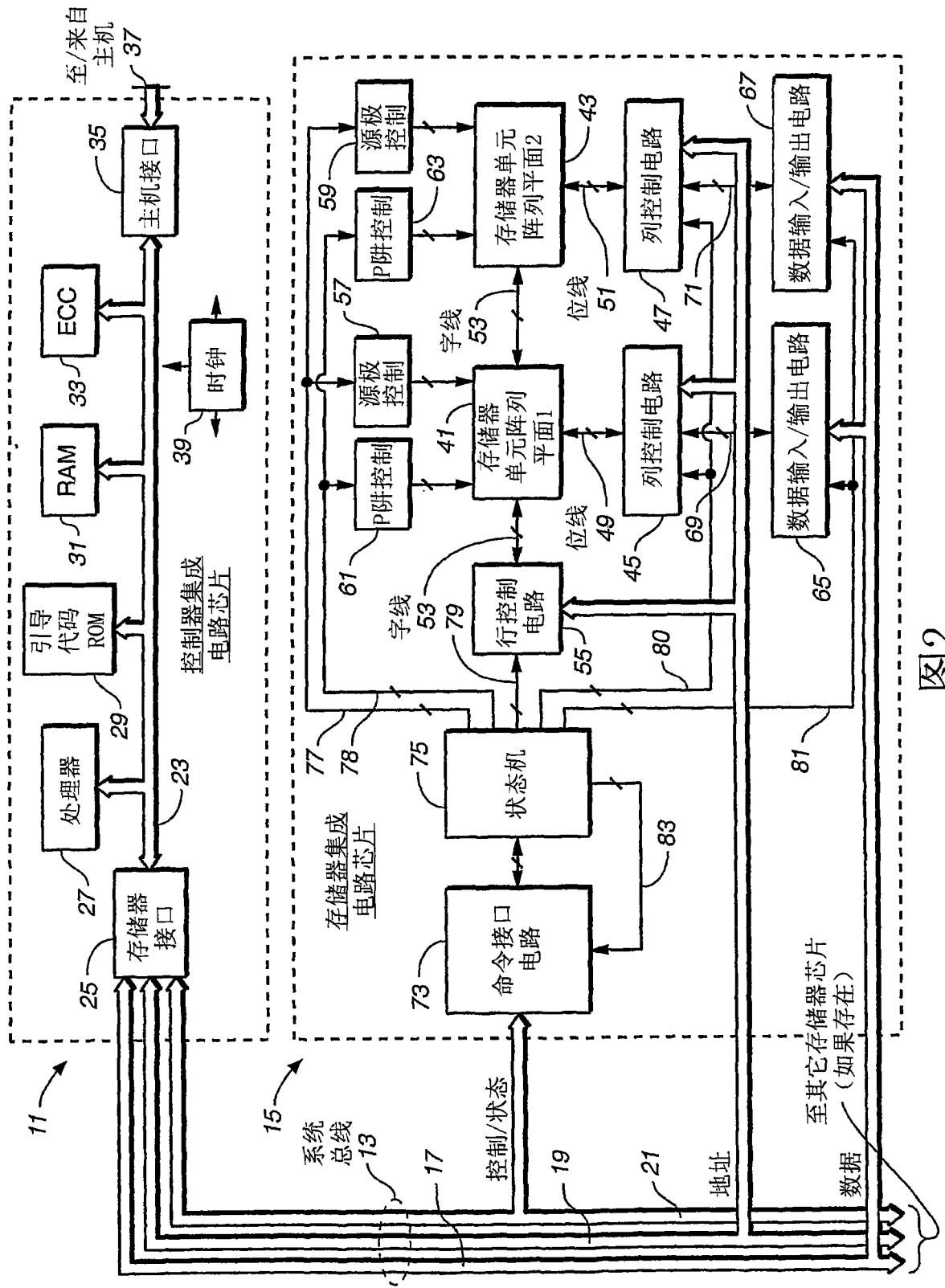


图1



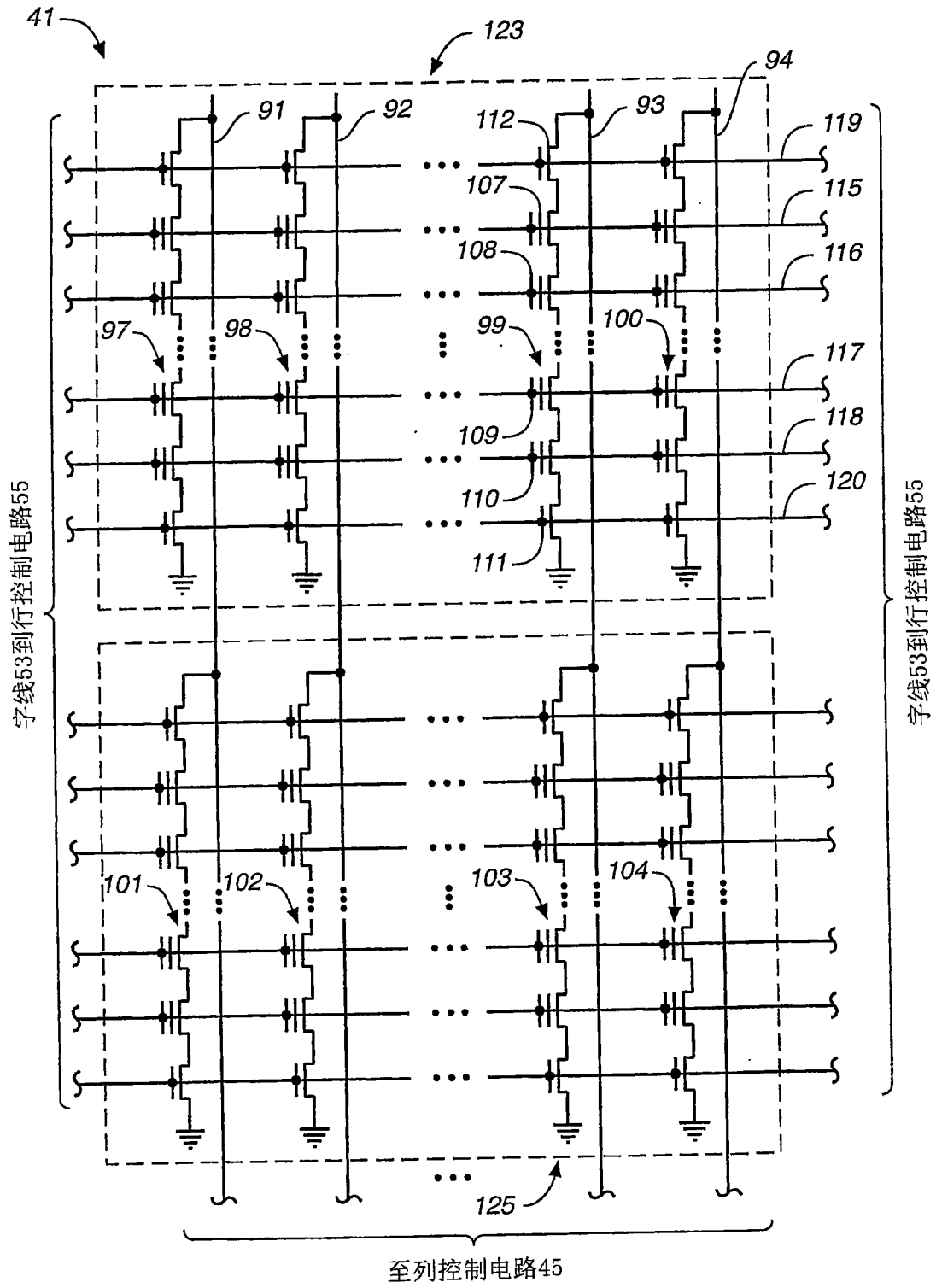


图3

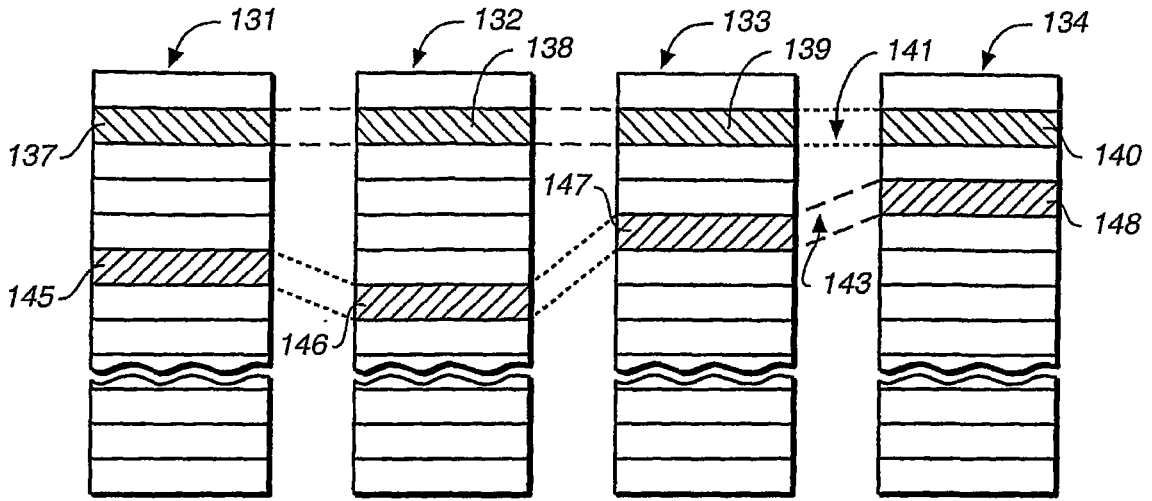


图4

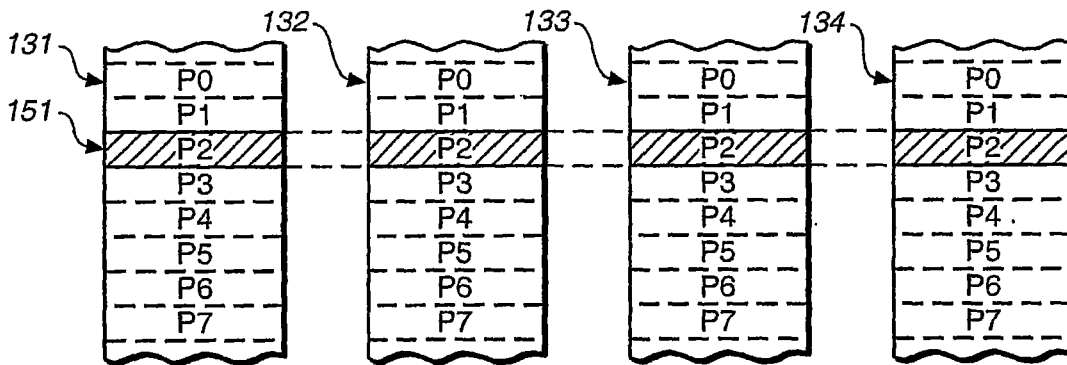


图5

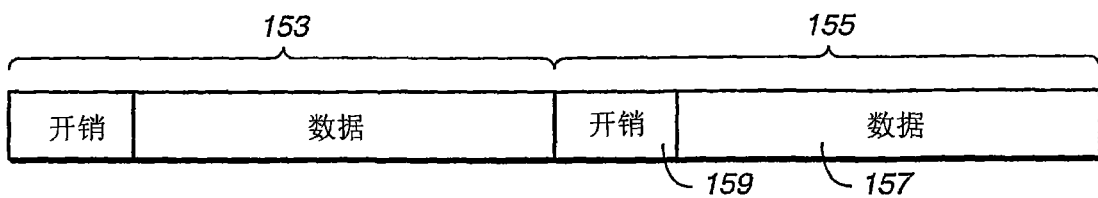


图6

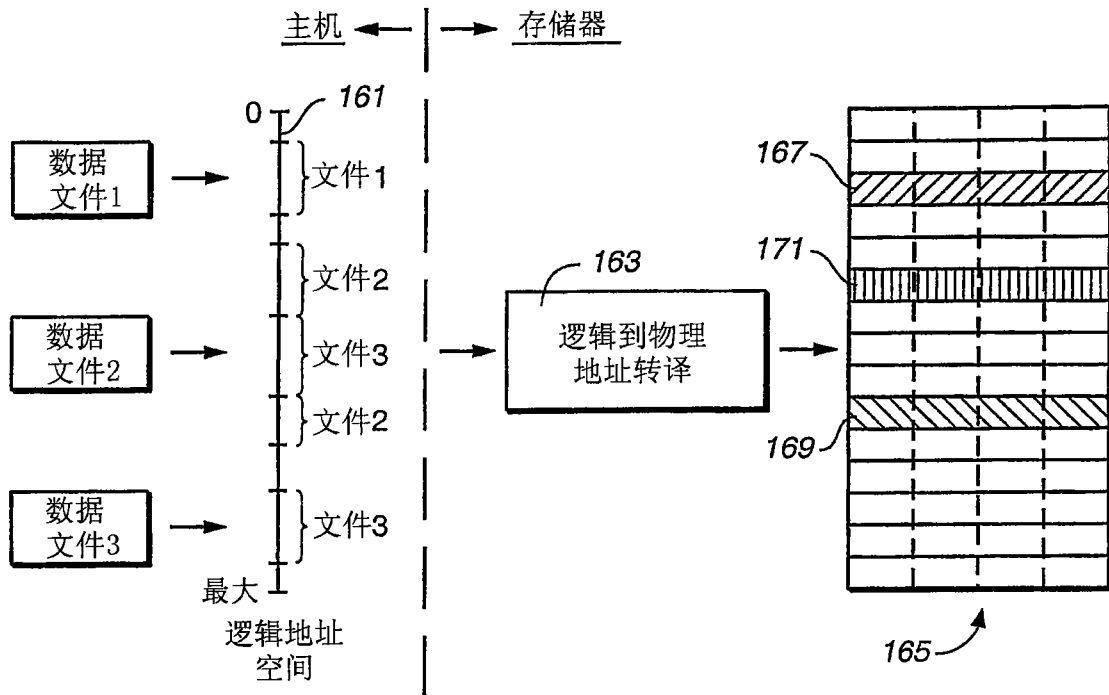


图7

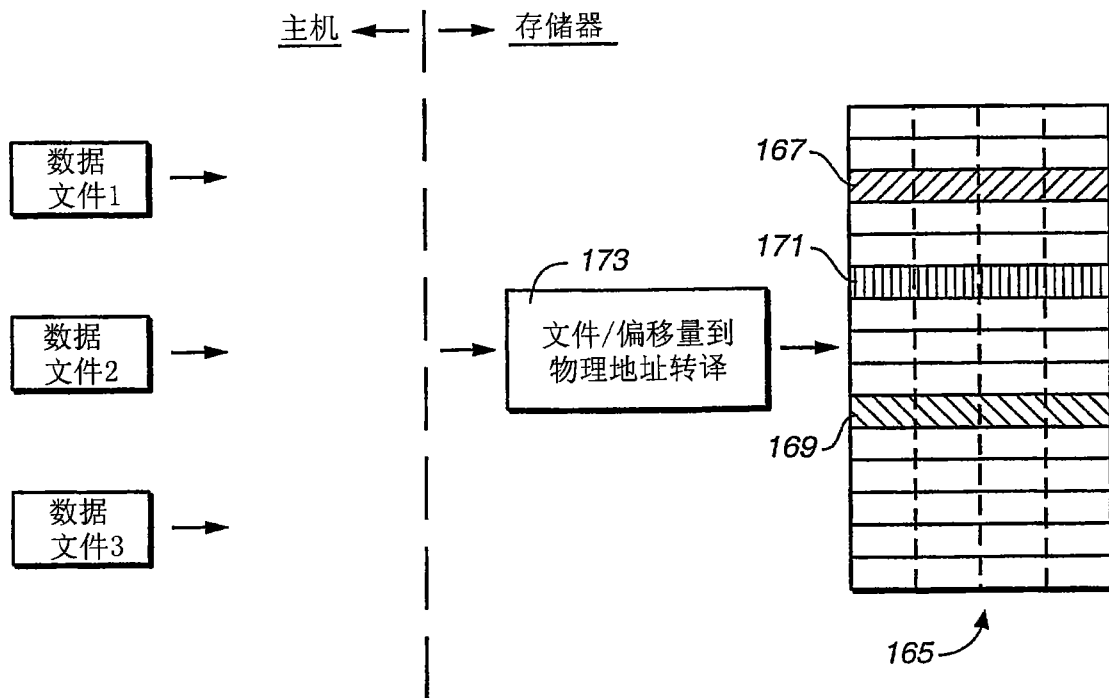
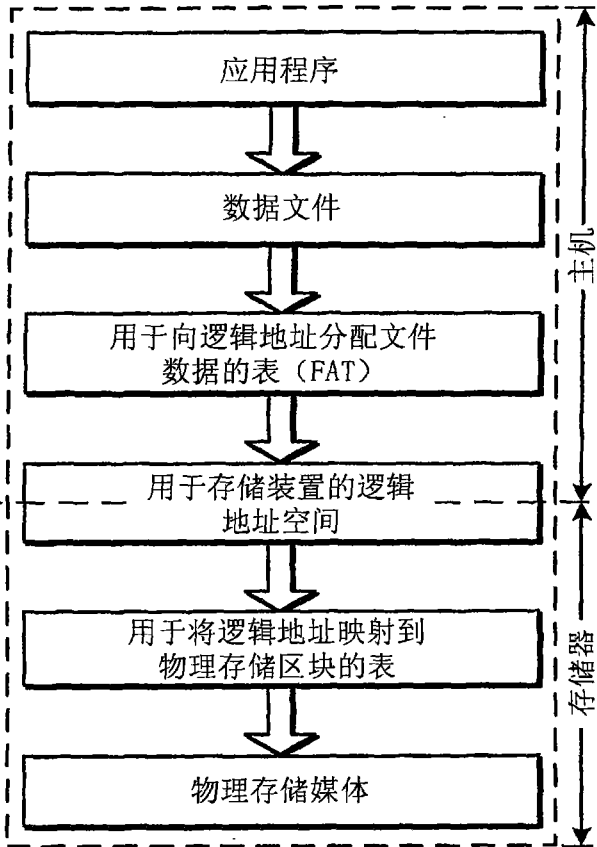
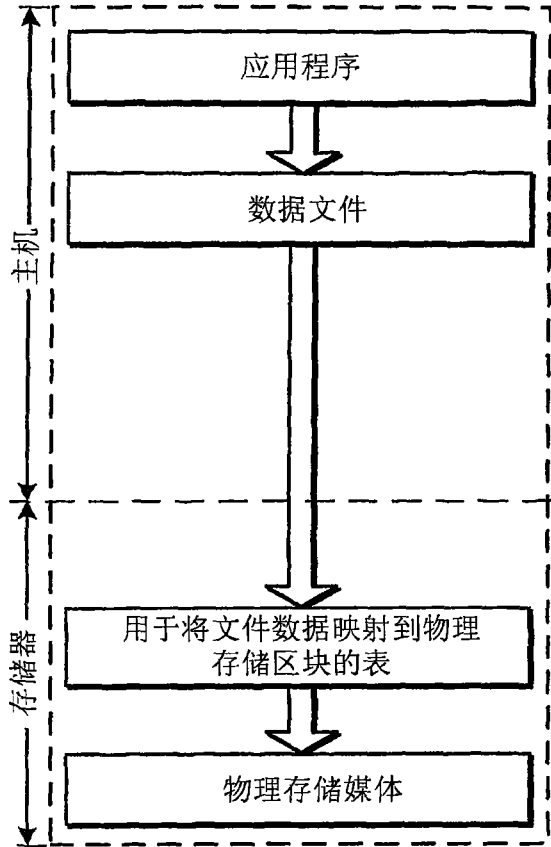


图9



现有技术系统

图8



直接文件存储系统

图10

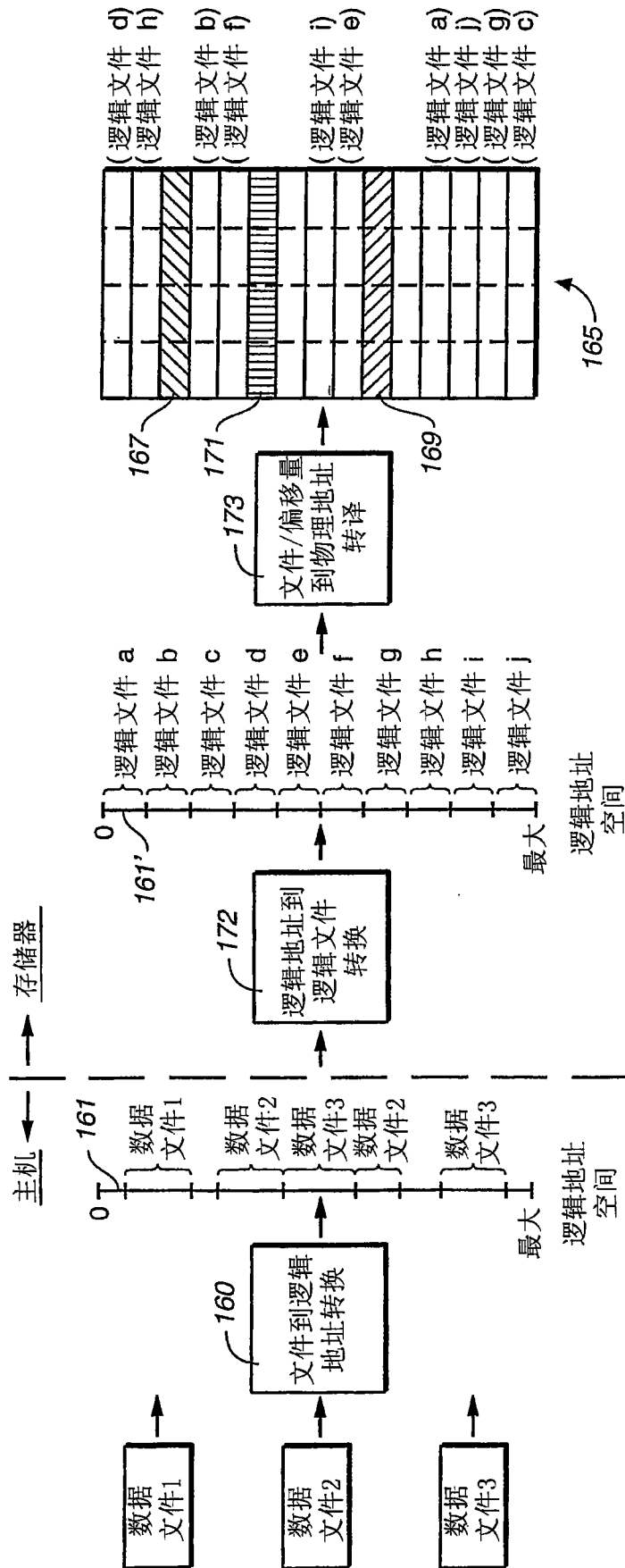


图11

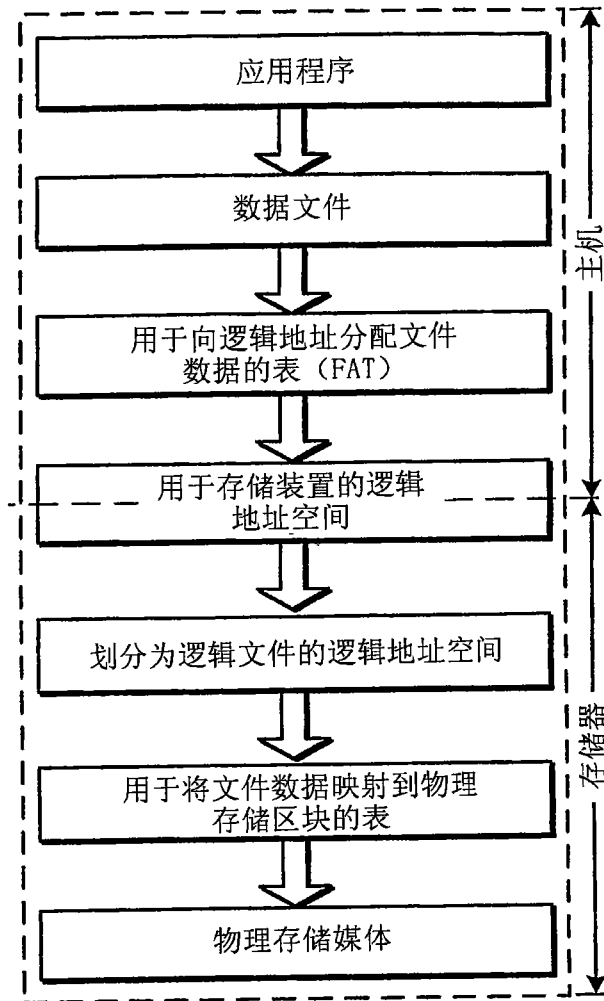


图12

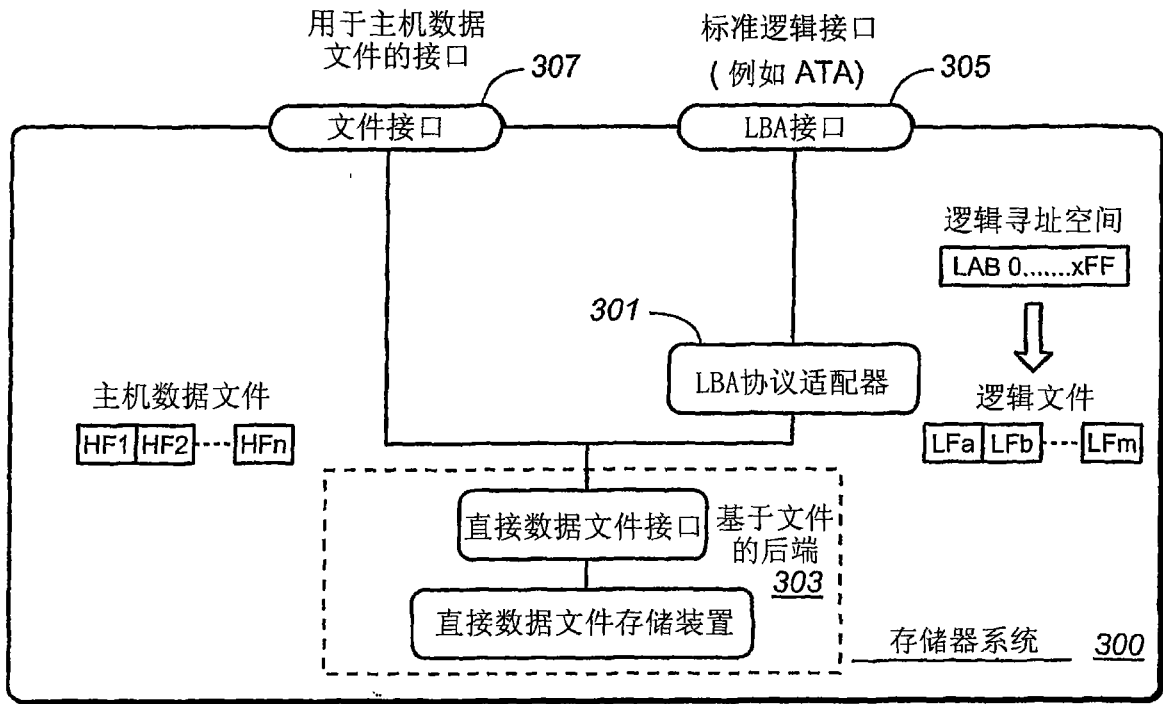


图13

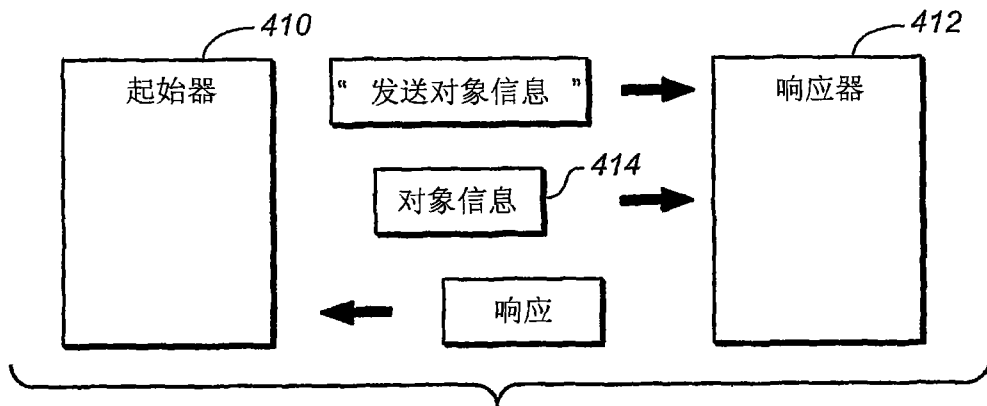


图14A

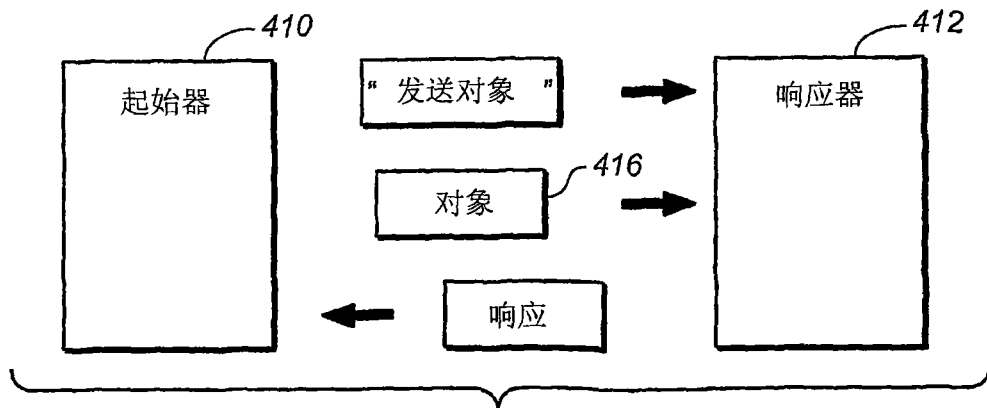


图14B

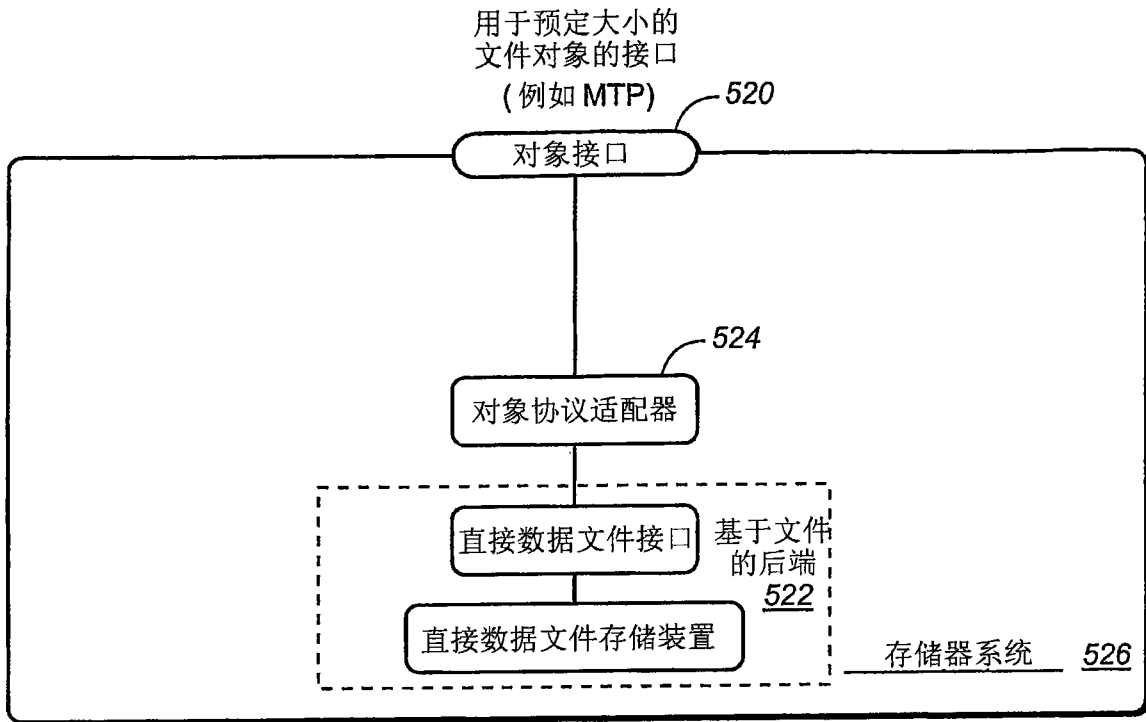


图15

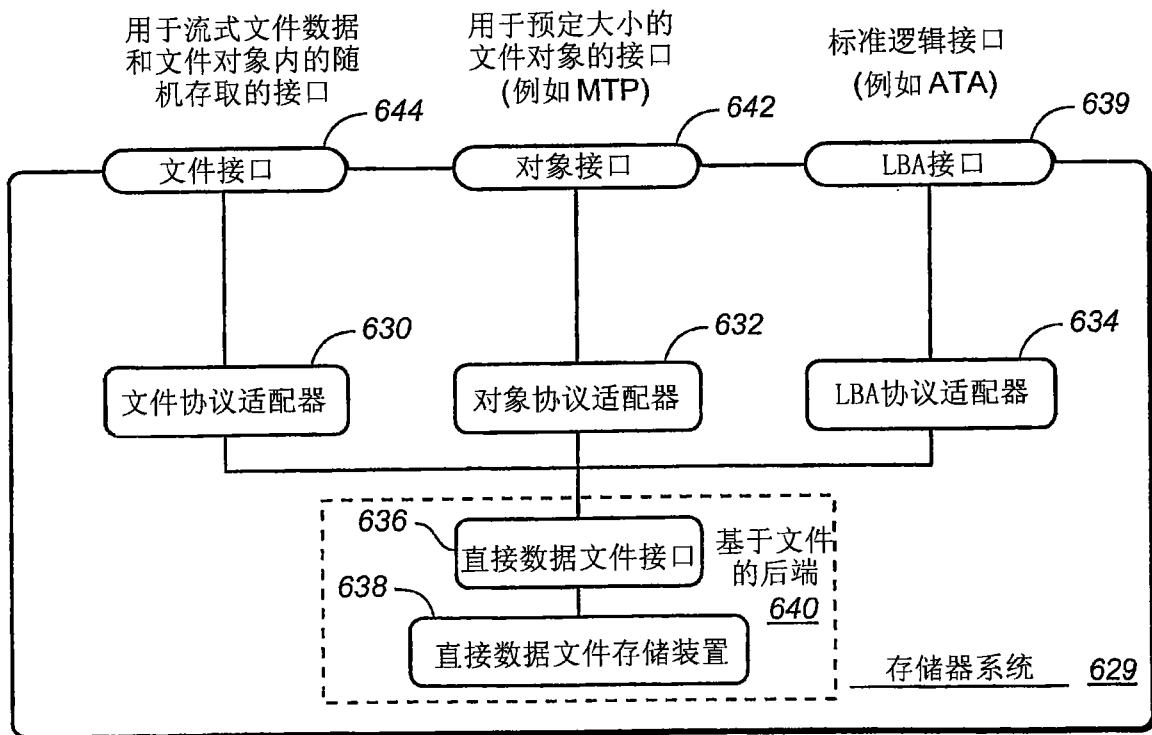


图16

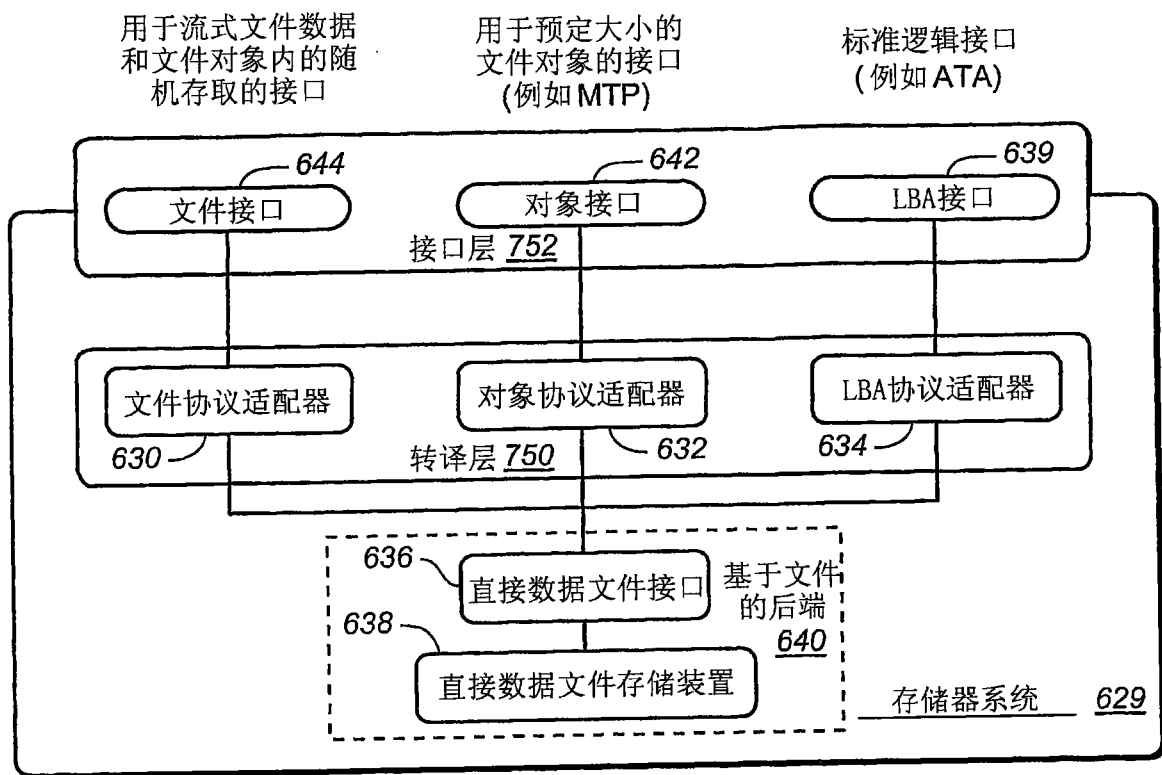


图17