

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G11C 29/00 (2006.01)

G06F 12/02 (2006.01)

G06F 3/06 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200480041681.0

[43] 公开日 2007年3月14日

[11] 公开号 CN 1930635A

[22] 申请日 2004.12.15

[21] 申请号 200480041681.0

[30] 优先权

[32] 2003.12.30 [33] US [31] 10/750,157

[86] 国际申请 PCT/US2004/042400 2004.12.15

[87] 国际公布 WO2005/066974 英 2005.7.21

[85] 进入国家阶段日期 2006.8.14

[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 卡洛斯·J·冈萨雷斯

艾伦·D·布莱斯

瑟吉·A·戈罗别兹

艾伦·D·贝内特

[74] 专利代理机构 北京律盟知识产权代理有限责任
公司

代理人 刘国伟

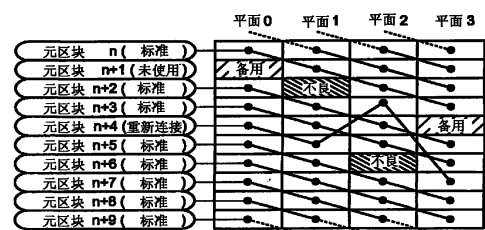
权利要求书 6 页 说明书 17 页 附图 11 页

[54] 发明名称

对多个区块进行适应性确定群组以成为多个多区块单元

[57] 摘要

本发明揭示一种用于将一非易失性存储器的物理区块链接到复合逻辑结构或“元区块”中的技术。在确定良好物理区块初始链接到元区块中之后，将所述链接的一记录保存在所述非易失性存储器中，当需要时可在所述非易失性存储器中容易地存取所述记录。在一组实施例中，根据一算法确定性地形成所述初始链接，并且可根据所述存储器中任何不良区块的图案来将其优化。当出现额外的不良区块时，通过以良好区块取代链接中的所述不良区块而更新所述链接，所述良好区块优选是在与其所取代的区块相同的存储器子阵列中。



1. 一种操作一包括一控制器和一非易失性存储器的存储器系统的方法，其中所述非易失性存储器由复数个擦除单元组成，所述方法包含：
 建立一组元区块链接，每个链接由复数个擦除单元组成，所述控制器通过所述链接来存取所述非易失性存储器；和
 在所述非易失性存储器中存储所述元区块链接的一记录。
2. 根据权利要求1所述的方法，其中所述记录是所述组链接根据擦除单元的一完整规格。
3. 根据权利要求1所述的方法，其中根据一规则来形成所述组链接，且所述记录由作为所述规则的例外的那些链接组成。
4. 根据权利要求1所述的方法，其进一步包含：
 确定所述元区块链接的一第一元区块链接中的一擦除单元是有缺陷的；
 更新所述第一元区块链接，使其不再包含所述有缺陷的擦除单元；和
 将所述已更新链接的一记录存储在所述非易失性存储器中。
5. 根据权利要求4所述的方法，其中所述更新步骤包含使用所述擦除单元中的另一擦除单元来取代所述有缺陷的擦除单元。
6. 根据权利要求5所述的方法，其中所述擦除单元中的所述另一擦除单元选自未链接擦除单元的一列表。
7. 根据权利要求6所述的方法，其中未链接擦除单元的所述列表保存在所述非易失性存储器中。
8. 根据权利要求7所述的方法，其进一步包含：
 在使用所述擦除单元中的另一擦除单元取代所述有缺陷的擦除单元之后，更新未链接擦除单元的所述列表。

9. 根据权利要求5所述的方法，其中所述擦除单元中的所述另一擦除单元选自先前属于另一链接的一擦除单元。
10. 根据权利要求1所述的方法，其进一步包含：
 - 保存未链接擦除单元的一列表；
 - 确定所述元区块链接的一第一元区块链接中的一个或一个以上擦除单元是有缺陷的；和
 - 将所述第一元区块中的所述无缺陷擦除单元添加到未链接擦除单元的所述列表中。
11. 根据权利要求1所述的方法，其进一步包含：
 - 确定所述元区块链接的一第一元区块链接中的一擦除单元是有缺陷的；
 - 确定一替代擦除单元是否可用于所述有缺陷的擦除单元；和
 - 响应确定一替代擦除单元为不可用，从所述组元区块链接移除所述第一元区块。
12. 根据权利要求1所述的方法，其中所述非易失性存储器包含复数个准独立阵列，且所述元区块链接的一给定元区块链接中的所述复数个擦除单元中的每个擦除单元都来自所述准独立阵列中的一不同的准独立阵列。
13. 根据权利要求1所述的方法，其中所述非易失性存储器包含复数个准独立阵列，且所述元区块链接的一给定元区块链接中的所述复数个擦除单元由来自相同准独立阵列的多对擦除单元组成，其中所述对中的每一对都来自所述准独立阵列中的一不同的准独立阵列。
14. 根据权利要求10所述的方法，其中所述准独立阵列位于分开的芯片上。
15. 根据权利要求1所述的方法，其中将所述元区块链接的所述记录存储在所述非易失性存储器中除分派给用户数据之外的一部分中。
16. 根据权利要求1所述的方法，其中所述擦除单元中的每个擦除单元都由复数个扇区

组成，且所述扇区中的每个扇区都包括一数据区域和一额外开销区域，且其中包含数据的那些擦除单元的记录信息保存在其额外开销区域中。

17. 根据权利要求16所述的方法，其中将那些不具有数据的擦除单元的记录信息保存在所述非易失性存储器中除分派给用户数据之外的一部分中。
18. 一种操作一包括一控制器和一非易失性存储器的存储器系统的方法，所述存储器包含至少两个可独立存取的子阵列，其中将所述个别子阵列分成复数个非重叠物理结构，且其中链接来自所述子阵列中的个别子阵列的至少一个物理结构，以形成一复合逻辑结构，藉此，所述控制器可将所述复合逻辑结构组成部分作为一个单元一起存取，所述方法包含：
 - 确定所述复合逻辑结构中的一物理结构是有缺陷的；和
 - 响应确定所述复合逻辑结构中的一物理结构是有缺陷的，使用一替代物理结构来取代所述复合逻辑结构中的所述有缺陷的物理结构，同时使所述物理结构中的无缺陷物理结构保持链接以形成所述复合逻辑结构。
19. 根据权利要求18所述的方法，其中所述物理结构是物理区块。
20. 根据权利要求18所述的方法，其中所述子阵列形成于相同的芯片上。
21. 根据权利要求18所述的方法，其中所述子阵列中少于全部子阵列形成于相同的芯片上。
22. 根据权利要求18所述的方法，其中形成一复合逻辑结构的物理结构的所述链接的一记录保存在所述存储器中，且其中更新所述记录以反映所述取代。
23. 根据权利要求22所述的方法，其中根据一算法来形成所述链接，且所述记录描述与所述算法的偏差。
24. 根据权利要求22所述的方法，其中所述记录描述所述链接中所有所述物理结构的物理地址。

25. 根据权利要求24所述的方法，其进一步包含：

在所述确定所述复合逻辑结构中的一物理结构是有缺陷之前，形成物理结构的一初始链接以形成所述复合逻辑结构，且其中所述取代步骤包括更新所述链接。

26. 根据权利要求25所述的方法，所述形成一初始链接的步骤包含：

从一可用物理结构池中选择形成所述复合逻辑结构的所述物理结构。

27. 根据权利要求25所述的方法，所述形成一初始链接的步骤包含：

根据一算法来形成所述复合逻辑结构。

28. 一种包括一控制器与一非易失性存储器的存储器系统，其中所述非易失性存储器由复数个擦除单元组成，其中所述控制器根据一组元区块链接来存取所述非易失性存储器，每个元区块链接由复数个擦除单元组成，其中所述控制器以一确定性方式建立所述组元区块链接。

29. 根据权利要求28所述的存储器系统，其中根据一算法来建立所述组元区块链接。

30. 根据权利要求29所述的存储器系统，其中非根据所述规则形成的元区块链接由一旗标指示。

31. 根据权利要求30所述的存储器系统，其中所述旗标保存在所述控制器中。

32. 根据权利要求30所述的存储器系统，其中所述旗标保存在所述非易失性存储器中。

33. 根据权利要求29所述的存储器系统，其中所述算法根据所述非易失性存储器中的有缺陷区块的图案来优化所述组链接。

34. 根据权利要求29所述的存储器系统，其中所述控制器基于所述非易失性存储器的一扫描来确定所述非易失性存储器中的有缺陷区块的图案。

35. 根据权利要求28所述的存储器系统，其中基于一随机分配来建立所述组元区块链接。

36. 根据权利要求28所述的存储器系统，其中通过使用来自一个或一个以上无缺陷区块的一列表的无缺陷区块取代一链接中的一缺陷区块来响应缺陷而更新所述组元区块链接。
37. 根据权利要求36所述的存储器系统，其中一个或一个以上无缺陷区块的所述列表的一记录保存在所述非易失性存储器中。
38. 根据权利要求37所述的存储器系统，其中一个或一个以上无缺陷区块的所述列表的所述记录由所述控制器高速缓存在所述控制器的易失性存储器中。
39. 根据权利要求28所述的存储器系统，其中所述组元区块链接的一记录保存在所述非易失性存储器中。
40. 根据权利要求39所述的存储器系统，其中所述组元区块链接的所述记录由所述控制器高速缓存在所述控制器的易失性存储器中。
41. 根据权利要求39所述的存储器系统，其中根据一算法来建立一组初始的元区块链接，且其中所述组元区块链接的所述记录仅列出那些不符合所述算法的链接。
42. 根据权利要求41所述的存储器系统中，其中所述组元区块链接的所述记录仅列出那些不符合所述算法的擦除单元。
43. 一种操作一具有一组织成至少两个平面的存储器存储元件阵列的非易失性存储器系统的方法，其中将所述个别平面分成复数个非重叠的存储元件区块，其中一区块包含可一起擦除的最小存储器存储元件群组，且将所述个别区块分成复数个存储元件页面，其中一个页面为可一起编程的最小存储器存储元件群组，所述方法包含：
将来自所述至少两个平面的个别平面的至少一个区块链接起来以形成一个元区块，其中所述元区块组成部分作为一个单元一起擦除；
如果一备用物理区块在相同平面中可用，那么将与一有缺陷的物理区块相关联的一元区块组成部分区块重新定向到所述相同平面中的一备用物理区块；

如果在相同平面中没有备用物理区块可用，那么将与所述有缺陷的物理区块不相关联的所述元区块组成部分分派给一备用区块区域；和

更新一缺陷图结构，其中所述缺陷图结构包括对应于所述元区块的一项目。

对多个区块进行适应性确定群组 以成为多个多区块单元

技术领域

本发明一般涉及半导体非易失性数据存储系统，更明确地说，涉及一种用于将物理区块形成为容纳非易失性数据存储系统中的缺陷的较大逻辑结构的系统和方法。

背景技术

例如闪存之类的非易失性存储器装置通常用作大量数据存储子系统。此类非易失性存储器装置通常以封闭卡(其以可移除的方式与主机系统连接)的形式加以封装，且还可作为主机系统内的非可移除的嵌入式存储器而加以封装。在一典型实施方案中，所述子系统包括一个或一个以上快闪装置，且通常包括一子系统控制器。

目前的商业存储卡格式包括个人计算机存储卡国际协会(PCMCIA)、CompactFlash(CF)、MultiMediaCard(MMC)、安全数字(SD)、存储棒和MemoryStick-Pro的格式。这些卡的供应商之一为SanDisk公司，其是本申请案的受让人。此类卡适用的主机系统包括数码相机、蜂窝式电话、个人计算机、笔记型计算机、手持计算装置、音频再现装置等。

所述非易失性存储装置本身由一个或一个以上非易失性存储元件阵列组成。每个存储元件能够存储一位或一位以上数据。非易失性存储器阵列的一个重要特征在于，即使在存储器阵列断电时，其也将经编程的数据保留在其中。相反，易失性存储装置要求周期性地更新阵列的功率，以便保留易失性存储器阵列中所含的数据。非易失性存储器的另一特征在于，一旦一小区被包含在编程非易失性存储器阵列内，那么在使用新的数据值重新编程所述小区之前，就必须首先擦除所述小区。

可通过使用一浮栅晶体管，例如电可擦除可编程只读存储器(EEPROM)来实施用于将电荷存储在存储器小区中的物理构件。例如EEPROM之类的浮栅装置的一个已知问题在于，在非常大数目的写入、编程与擦除循环之后，所述浮栅最终损坏并发生故障。当发生这种情况时，所述小区不可再使用，必须将其从所述阵列中的可用存储器小区的列表中剔除。这类缺陷称为“成长”缺陷。在一个市售实施方案中，通过将所述有缺陷的小区映

射出去且以良好存储器小区的物理地址替代新检测到的有缺陷的存储器小区来处理这些缺陷。将有缺陷的小区或扇区映射出去并加以取代的实施方案的实例在1997年8月19日颁予Mehrotra等人的美国专利第5,659,550号、1997年9月23日颁予Harari等人的美国专利第5,671,229号；和1999年1月19日颁予Harari等人的美国专利第5,862,080号中得以描述，所述申请案全文以引用的方式特意并入本文中。

通常将非易失性存储器小区阵列分成多个群组，以有效地实施读取、编程和擦除功能。举例来说，在许多非易失性存储器结构中，将所述存储器小区配置成称为擦除单元的较大群组。此擦除单元为可一次擦除的最小数目的存储器小区。

擦除单元的大小取决于所实施的存储器结构。在较早的非易失性存储器中，擦除单元为大小与标准512字节磁盘驱动器扇区相同的区块。在一个商业形式中，每个区块都包含足够的小区，以存储一个扇区的用户数据加上与用户数据和/或其中存储有所述用户数据的区块有关的一些额外开销数据。为了确保个别地擦除小区的区块，必须使所述区块彼此充分地隔离。

因为此隔离占用了集成电路芯片上的有用空间，所以开发出另一种存储器结构，其中使擦除单元明显更大，因此此类隔离所需的空间较少。在2003年6月17日颁予Conley等人的美国专利第6,580,638号中描述此大区块系统结构的实例，所述申请案为2002年7月30日颁发的美国专利第6,426,893号的接续申请案。这两份专利全文以引用的方式特意并入本文中。在大区块系统中，通常进一步将所述擦除单元分成可个别寻址的页面，其为读取和编程用户数据的基本单元(编程和/或读取单元)。在一个商业实施方案中，所述擦除单元为元区块。元区块为由多个物理擦除单元组成的虚拟擦除单元。这些多个物理擦除单元可用在并行编程和擦除操作中，但作为单个逻辑区块来寻址。

在美国专利第6,034,897号中描述一种形成元区块或“超”区块的方法，所述申请案全文以引用的方式特意并入本文中。如本文所述，对于具有若干装置的存储器来说，可将所述装置中的每一者中的相同物理区块群聚成一个元区块。尽管这允许形成元区块且允许给定元区块中的所有区块由相同地址(即所述装置中的第一者中的区块的地址)来寻址，但其具有若干限制。举例来说，因为区块到元区块中的链接在此固定配置中是预定的，所以当一区块变坏时，所述区块所属的元区块变坏，尽管元区块内的其它区块仍在起作用。因此，理想的目标是提供一种用于可适应元区块配置，同时免去每次需要新链接时都建立一个新链接的操作额外开销的系统与方法。

发明内容

本发明的各个方面克服现有技术中将物理存储器结构链接到复合逻辑结构中时的这些和其它限制。在本发明的主要实施例中，揭示了用于在闪存中形成元区块结构的技术。根据本发明的第一方面，更新区块到元区块中的链接，以容纳存储器的有缺陷部分。一种用于保存缺陷图的系统包括一缺陷图结构，其用于在组织成元区块的非易失性存储器系统中追踪工厂缺陷和成长缺陷（Grown defect）的存在。一种用于保存缺陷图的方法利用缺陷图结构和元区块的特征。在一个实施例中，缺陷图结构的大小等于非易失性存储器系统的并行度。在另一实施例中，无法分派给元区块的剩余擦除单元将保存在备用擦除单元的一列表中，以供以后使用。

根据本发明的另一方面，将区块到元区块中的链接的一记录保存在非易失性存储器中。接着，当需要时，控制器可读出到其易失性存储器中的链接。响应由缺陷所引起的重新链接而更新记录。可将完整的链接记录保存在非易失性存储器中，例如，在指定的位置中，或仅可保存一部分链接记录。在本发明的另一方面中，通过一使用（例如）基于固件的实施方案的算法来确定性地形成区块到元区块中的初始链接。在这种情况下，存储在非易失性存储器中的链接记录仅需包含与标准算法的偏差。

本发明的附加方面、特征和优势包括在对示范性实施例的以下描述内容中，应结合附图来阅读所述描述内容。

附图说明

图1是其中非易失性存储器阵列与存储器控制器封装在一以可移除方式与主机系统连接的封闭卡中的现存非易失性存储器系统的方框图。

图2是其中非易失性存储器阵列封装在一以可移除方式与主机系统连接的封闭卡中的现存非易失性存储器系统的方框图。

图3是其中非易失性存储器阵列作为主机系统内的非可移除嵌入式存储器而封装的现存非易失性存储器系统的方框图。

图4a说明一显示平面、区域与备用区的实例芯片格式。

图4b到d说明物理区块到元区块中的各种链接。

图5到10说明所述系统可如何响应发生在存储器阵列的一部分上的一系列成长缺陷。

图11说明响应在擦除单元上检测到成长缺陷而映射元区块的虚拟结构元素的实例。

图12说明用于保存缺陷图的系统的实例，其中备用的擦除单元在与具有成长缺陷的

擦除单元相同的平面上不可用。

图13说明用于保存缺陷图的方法的实例。

图14说明使用图11到13的数据结构来保存缺陷图的方法的实例。

图15到18说明物理区块的基于算法的链接。

图19、20和22说明物理区块的重新链接。

图21显示二维情况下的重新链接的实例。

具体实施方式

一般来说，本发明的存储器系统在其主非易失性存储器中保存区块到多区块元区块结构中的链接的记录。此记录在可根据需要读取到非易失性存储器中的一映像表中包含工厂缺陷和成长缺陷的缺陷图。在碰到成长缺陷且将区块分派给新的位置时更新此图。元区块优选由数个子阵列或平面中的每一者中的一个区块形成，且在这种情况下，优选对每个平面中编号相同的区块进行群聚，除了被标记为不良且置于所述图中的区块。在一个实施例中，将所述区块直接映射到替代区块。接着使用对应平面中的替代区块来重新形成将包含缺陷区块的元区块。并非将区块到元区块中的链接的完整记录保存在非易失性存储器内的元区块中，“标准”链接可基于植入（例如）系统的固件中的算法，其中仅需要存储由缺陷引起的与此算法的偏差。那些无缺陷存在的标准元区块将称为可使用的标准元区块。剩余的元区块称为不可使用的标准元区块，且组成部分物理区块称为备用区块。

图1到图3是典型现存非易失性存储器系统的方框图，所述典型现存非易失性存储器系统包括一主机102、202、302；一控制器104、204、304和一非易失性存储器阵列或复数个阵列103、203、303。在图1中，将非易失性存储器阵列103和存储器控制器104显示为封装在封闭卡中，所述封闭卡以可移除的方式与主机系统102连接。在图2中，将控制器204显示为主机系统202的一部分，且将非易失性存储器阵列203显示为封装在封闭卡中，所述封闭卡以可移除的方式与主机系统连接。在图3中，将非易失性存储器阵列303显示为作为主机系统302内的非可移除的嵌入式存储器而封装。

通常，非易失性存储器阵列包括多个存储器小区，其包括一个或一个以上导电浮栅作为存储元件。存储器阵列可代替地包括其它长期电子电荷存储元件，例如电荷捕获电介质。可使用多种非易失性存储元件，包括非易失性多状态存储元件、为Flash/EEPROM/浮栅的存储元件、例如NROM、MNOS、MRAM/FRAM、NAND、NOR之类的非易失性

存储元件和组织成擦除单元的非易失性存储元件。可使用存储数字信息、模拟信息或两者的组合的非易失性存储元件。

可使用为每个存储元件定义的两个电荷电平来操作存储器小区阵列，因此使用每个元件来存储一位数据。或者，可为每个存储元件定义两个以上的存储状态，在这种情况下，可在每个存储元件中存储一位以上数据。

图4a说明可如何将存储器芯片格式化为擦除单元(或区块)的群组的实例，其中根据物理平面和逻辑区域来配置所述区块。以此方式分割芯片的一些原因包括均衡每个群组内的良好擦除单元的数目，且留出存储器芯片的专用区段。举例来说，可留出专门的“保留”区段，以存储例如系统与装置参数之类的信息和有关区域对准的信息。还可留出区段来存储备用的擦除单元，也称为备用区或备用区块。可将与备用区有关的信息（例如，其物理地址）保存在一备用区列表中，以供需要时进行检索。举例来说，当在N个区域中的一个区域中所含的多个擦除单元中的一个擦除单元中检测到缺陷时，可通过将有缺陷的擦除单元重新映射到可用的多个备用擦除单元中的一个备用擦除单元而找到备用区。

来自主机的数据通常只被提供到识别为逻辑扇区的存储器系统。举例来说，所述主机将发送根据起始逻辑扇区和扇区总数来描述的数据。在所述主机内，可将这些逻辑扇区建构成较大的单元，但通常不将此信息转到所述存储器的控制器。(尽管可使用这些主机结构的知识，以获得存储器中的物理结构或控制器构造，其会反映这些主机结构，当它们将反映主机如何传递数据时：例如，通常将存储器建构为基于大小与逻辑扇区相同的物理扇区来存储用户数据。)所述控制器将主机结构组织成模仿物理结构的逻辑构造，以便更有效地将它们存储在物理存储器中。举例来说，在常见配置中，所述存储器系统控制器将逻辑扇区(如主机所定义)群聚成逻辑区块，其大小对应于一区块的物理结构，所述区块为用于快闪型存储器的物理擦除单元。接着，所述控制器可（例如）以扇区地址表(Sector Address Table; SAT)的形式保存物理结构与逻辑构造的关系，且当逻辑与物理关系发生变化时更新此对应关系。

为了获得更大的效率，存储器系统通常通过引入更大的结构来增加并行度。举例来说，在具有若干可同时被写入、读取或读写的半自律阵列的存储器系统中，将来自不同阵列的区块群聚成“元区块”结构，且所述控制器将数据的逻辑区块形成为对应的逻辑元区块，以允许同时操作多个区块。可由具有单个存储器芯片的平面或越过几个存储器芯片而分布的一个或一个以上平面来形成元区块。以此方式，通过允许一组(逻辑上)循序扇区横跨多个可被并行读取/编程/擦除的物理区块，接着可以最少量的非同时读取、编程

或擦除操作来同时读取/编程/擦除此群组。此配置有利于循序读取和写入相对较大数量数据的逻辑上相邻的扇区。

首先将论述图5到10的结构，接着将个别地解释每个图式所示的操作。图5到10显示四个平面中六个区块的配置，用于总共24个擦除单元，作为图1到3的存储器阵列的实例。为方便起见，所述擦除单元将称为区块，因为区块是一种擦除单元。为便于解释，选择此6x4区块结构的大小，此点将在下文中显而易见。典型的存储器阵列将包含比此处所示的多得多的区块(具有多得多的行)，且如果非易失性存储器系统的并行度大于四，那么可能具有比所示更多的平面。

每个平面通常具有其自己的数据寄存器和编程电路。这允许同时编程到所述平面中的每一者的存储器小区的区块中。每个平面可半独立于其它平面而操作。为便于说明，图5到10所描绘的平面显示为具有6个物理区块0到5。在任何给定配置中区块的实际数目可能较低，但通常要高得多。

在具有大量区块的系统中，可将所述平面逻辑上分成如图4a所示的区域。所述区域越过复数个平面而延伸。将所述平面分成区域的做法可将一平面分成较小的更易于管理的程序段。一区域内每个平面中的区块可占用同一组相邻物理区块地址。区域的数目并未受特别限制，尽管实际的设计决策可影响特定存储器系统中所实施的区域数目。本发明的各个方面可在具有或不具有区域结构的存储器中实施，且对于具有区域结构的存储器来说，本发明的各个方面可在作为整体或具有个别区域的平面内实施。

应用于图5到10中的情形的存储器系统所使用的操作单元优选为元区块。元区块表示越过平面的复数个物理区块的虚拟链接。元区块的选择性使用包括对所述元区块内所有区块的管线式擦除和对来自所述元区块的每个区块的一个或一个以上页面的管线式编程和读取。

元区块越过多个平面而延伸，且通常包括来自这些平面中的每一平面的一个区块，如由图5到10中所示的每一组连接箭头中的箭头所示。图5显示元区块的最简单形式，其中每一个元区块的每个物理区块都具有相同的区块偏移，且越过所有四个平面0到3而延伸。图10说明另外两个复杂的元区块。将在图5到10的描述内容中论述这些元区块的创建，其中当碰到缺陷时，更新元区块中所含有的组成部分区块。

图4b中显示区块到元区块中的链接的更一般的情况。系统的存储器部分(分别为图1到3的103、203、303)由芯片0到Z所组成，其中每个芯片具有若干平面，例如芯片0具有平面00到0W。一平面中由X所指示的每个经链接的物理单元可由单个物理区块所组成，

如图4c所示,或由多个物理区块所组成,如图4d所示。当物理区块较小时,图4d所示的不仅越过平面与芯片而且也在深度方向链接区块的所述种二维元区块可能特别方便。然而,方便可能不是图4d的配置的唯一或甚至主要原因。在通常或一维链接中,主要的动机是要提供一用于处理资料的机制,所述资料可被同时写入到不同芯片/平面中的多个区块。同样,如果存储器设计允许并行地存取(读取和编程)一平面中多于一个的位置,那么链接相同平面内的多个区块也可能是有意义的。

显示数个元区块链接,其中根据所述链接中第一区块的第一行来对每个元区块MB进行编号;例如,将以区块0开始的元区块标记为 MB_0 等。尽管图4b显示每个平面在一给定链接中具有一区块(且仅具有一个区块)且所述链接越过数个平面而延伸,但这两种情况不一定总是如此。举例来说,(例如)图5到10中的四个区块的链接可由单个芯片上的四个平面或来自所述平面的一个平面所组成。

如图4b所示,链接 MB_0 的所有区块都来自相同的行, MB_1 的大多数区块来自相同的行,并且 MB_2 的每个区块都来自前一区块下面的行。为便于论述,图5到10的论述将采用情况 MB_0 ,其中所有区块都来自相同的行,下文论述更一般的情况。这些各种链接方法可为固定,或根据下文详述的本发明的一方面,基于一算法来确定。(或者,初始确定可基于随机的分配。)在更一般的情况下,应了解,本发明的各方面一般来说可应用在物理结构形成于较大的逻辑集块中时,且应用于如何管理这些集块。

图10所示的第一元区块在四个平面0到3中的每个平面中都包括区块0。第二元区块在平面0中包括区块1,在平面1中包括区块4,在平面3中包括区块5,且在平面3包括区块4。第三元区块在四个平面0到3中的每个平面中包括区块2。第四元区块在平面0中包括区块3,在平面1中包括区块3,在平面2中包括区块3,且在平面3中包括区块1。

或者,元区块仅可包括单个区域内所含有的若干平面的一子集。未显示此实例,但如果并行度的可变程度为理想的,那么所述实例可能是有用的。具有可变程度的并行度的系统可允许区块在具有不同并行度的区域之间移动。举例来说,包含两个而非四个区块的元区块可从并行度为二(两个平面)的区域移动到并行度为四(如同在四个平面实例中)的区域。如果在这种系统中存在各包含两个部分的两个元区块,那么它们可组合成一个四部分元区块。使用这种混合同行度系统的一个警告在于,必须调整缺陷图以说明变化的并行度。

一般来说,使用元区块可通过允许一次并行地处理更多区块而增加存储器系统的并行度。这增加的并行度显著增加了存储器系统的性能。所述存储器系统保存各个元区块

内的个别区块的识别项。举例来说，各个元区块内个别区块的识别项可作为链接的列表而保存在所述系统内任何方便位置中。

通常，与每个页面的用户数据一起存储的额外开销数据将包括一逻辑和/或物理地址，其足以识别页面所在的平面、区域和区块，以及所述区块内所述页面的偏移。接着，通过控制器读取页面额外开销数据的这些地址字段而在所述控制器的存储器内创建一地址图。这通常一次对存储器的一部分而进行，在针对存储器的所述部分的编程、读取或擦除操作之前。

图5到10显示系统如何响应在存储器阵列的物理区块中检测到缺陷。这些缺陷可能为工厂缺陷或成长缺陷。此系列的图式意欲显示每次碰到新的缺陷时会发生什么情况。出于说明的目的，编号为1到5的物理区块包含缺陷。

第一图式，图5显示其中不存在缺陷的存储器阵列的一部分。通过越过所有四个平面而延伸的六个水平箭头来描绘复数个元区块。每个元区块包含已定义数目的物理区块。在图5到10中，对于所有的元区块来说，物理区块的已定义数目为四，其对应于平面的数目。

图5显示存储器阵列中未找到工厂缺陷或成长缺陷的一部分。越过图式水平延伸的黑色箭头描绘六个线性标准元区块。如图5所示，线性标准元区块为包含具有匹配偏移的物理区块的元区块。初始链接可根据一算法来建立，如下文进一步描述，或从可用良好区块的擦除池（erase pool）形成。

在本发明的一个实施例中，元区块的特征在于，第一虚拟区块地址(其与第一物理区块地址相关)为用于确定与特定元区块相关的剩余物理区块的位置所需的唯一信息，经链接的元区块的其余部分由一链接算法来确定。这缩小了存储元区块所需的数据的大小。如果在第一物理区块中碰到缺陷，那么整个元区块变为无效，且从地址空间中映射出去，因为不再存在任何机制用于指向越过所述可用平面中的每一者或多个平面而定位的元区块的剩余部分。下文将在图9的论述中显示此情况的实例。

图6显示在碰到第一个新缺陷之后阵列的状态。新缺陷位于 MB_1 中标为“A”的物理区块处，位于平面2上的区块1处。当元区块内的物理区块失效时，元区块无法照常使用。一个简单的解决方案是将所述元区块从地址空间映射出去，且永不再使用所述元区块。如果系统仅具有单个不良物理区块，且没有备用的区块可用，那么这是唯一的可用解决方案。然而，如果出现多个不良物理区块，那么因为与失效的元区块相关的剩余物理区块可用来创建其它良好元区块，所以此解决方案很快就变成多余的。由于图6中所找到的

缺陷(标为“A”), 剩余的三个良好物理区块(由影线标记显示)变成备用区。

图7显示在碰到第二个新缺陷之后阵列的状态。新缺陷位于标为“B”的物理区块处, 位于 MB_3 中平面3上的区块3处。在用于保存例如图11所示的缺陷图的方法中, 将在发生缺陷的平面(平面3)中搜索任何可用的备用区块。此处, 物理区块1在平面3中可用(且由图6中的影线标记显示)。因为区块1在平面3中可用, 所以会更新元区块以指向区块1, 以便取代元区块的缺陷部分; 即 MB_3 中平面3处的缺陷区块(“B”)在链接中由 MB_1 中平面3处的美好区块取代, 所述良好区块由于 MB_1 的平面2处的区块“A”失效而变为冗余。当碰到物理区块失效时, 将备用物理区块链接到元区块中, 以取代失效的物理区块。

(根据所用的命名常规, 在图6和7的点处不存在元区块 MB_1 , 因为整个元区块是基于平面0中的区块编号来命名的。除非平面0处的区块1返回到新的链接中, 名称 MB_1 才会后退, 如图8中所发生的情况。)

图8显示在碰到第三个新缺陷之后阵列的状态。新缺陷位于标为“C”的物理区块处, 位于平面0上的区块4处。因为缺陷区块位于平面0上, 所以无法重新链接此元区块。位于平面0上的区块为包含元区块的其它部分的参考信息的区块, 且如果没有平面0上的区块, 那么无法根据所采用的常规来参考所述元区块。因此, 即使备用区块在平面0上的区块1处可用, 其仍无法用于完成此元区块。替代地, 平面0上的区块1重新映射到虚拟的地址空间中, 且用作新的元区块中的第一区块, 以利用先前元区块 MB_4 的平面1、2与3中当前冗余的区块。因此, 缺陷区块“A”可由平面2的区块4来取代。因为平面3的区块1已在图7中重新映射, 所以其也需要被取代。因此, 可通过使用来自平面2和3的区块4处的备用区来形成重新链接的 MB_1 。对于平面1来说, 一区块在区块1和区块4中都可用。尽管可选择两者中的任一者来完成重新链接的 MB_1 , 但在所述实例中使用先前在 MB_4 中的区块。因此, 重新链接的 MB_1 包含平面0上的区块1、平面1上的区块4、平面2的区块4和平面3上的区块4。此时, 仅有一个备用区块保留在平面1上的区块1处。

图9显示在碰到第四个新缺陷之后阵列的状态。新缺陷位于标为“D”的物理区块处, 位于平面0上的区块5处。由于此时没有备用区块位于平面0中, 所以无法重新链接此元区块。因此, 可将剩余的物理区块添加到备用物理区块池。这包括平面1上的区块5、平面2上的区块5和平面3上的区块5。此时, 四个备用区块在所示的存储器阵列中可用。

图10显示在碰到第五个新缺陷之后阵列的状态。新缺陷位于标为“E”的物理区块处, 位于平面2上的区块4处。由于备用区块在同一平面上可用(平面2上的区块5), 所以通过使用所述备用区块来重新链接元区块 MB_1 。

应注意，在图10中，尽管六个原始元区块中的四个原始元区块(对应于行1、3、4和5)具有不良区块，但重新链接已得到四个良好的元区块。在例如美国专利第6,034,897号中所述的固定链接(阐述于背景技术部分中)下，这些缺陷会导致损失六个所示元区块中的四个元区块。以此方式，当存储器老化并出现缺陷时，其仍保持较大的数据容量。

图11说明可如何通过使用逻辑到物理映像表1101来映射元区块的组成部分。由影线标记所标记的区块显示与LBN[0]相关的存储器1103(或使用所述结构的实施例中的存储器区域)中的元区块。所述存储器的四个平面(在此实例中)可再次分布在多于一个芯片上。LBN[0]内所包含的PBN项目是指由影线标记所标记的区块中的每个区块的物理地址，且所述区块为元区块的部分。平面0中所包含的元区块的第一部分用于存取其它元区块部分。如关于图5到10所述，如果发现平面0中的元区块部分是有缺陷的，那么必须将整个元区块标记为有缺陷，因为它们经由平面0上的部分链接到元区块的其余部分。

映像表1101为形成元区块的区块的链接的记录的一个实例。允许元区块成分的适应性的一种技术为，只要写入过程需要元区块，就从可用良好区块池动态地形成元区块。尽管此技术允许元区块结构的适应性，但其要求只要需要区块，就重新建立链接；另外，其要求存取每个平面以便确定此池。根据本发明的主要方面之一，将此记录存储在非易失性存储器中一位置处。接着，可根据需要将此记录从闪存读取到控制器上的易失性存储器(或替代的非易失性存储器)中，以用于地址转换。

在一组实施例中，可将链接信息的记录保存在一特殊的非易失性存储器位置，例如在存储用户数据的正常物理区块外部的某处。举例来说，可将此记录保存在图4a的平面0中的保留区域的一者中，或在例如美国专利申请公开案US2003/0065899中所述的配置中的系统区域的任一者中，所述公开案以引用的方式并入本文中。在另一组实施例中，可将链接信息保存在已写入或已部分写入的区块的用户数据扇区/控制数据扇区的标头区域中。对于擦除状态中的区块来说，可将链接信息存储在非易失性存储器中的特殊控制数据区域中。

通过使用这种表格，可更新区块到元区块中的链接以容纳缺陷，且通过将一记录保存在非易失性存储器中，需要时可容易地对其进行存取，从而克服背景技术中所论述的现有技术的限制。

图12说明本发明的缺陷图方面，其在平面1上显示一不良区块。所述不良区块以“X”标记。在发现区块有缺陷之前，元区块LBN[0]的第二部分指向所述区块，如由虚线箭头1201所示。备用区块集合1203在由影线标记所示的相同平面上具有可用的备用区。(或者，

如在图5到10的过程中，如果可在平面1中从已出错的另一链接获得一备用区，那么可将其用于替代来自备用区集合1203的一区块。)备用区PBN'的物理区块编号(PBN)将取代失效区块的PBN。因此，此备用区变成元区块的一部分，如由箭头1202所示。

图13说明与图11的映像结构类似的映像结构的实例。图13的映像结构显示一不良区块，可能为成长缺陷，如由平面2上的“X”所标记的区块上所示。在与不良区块相同的平面上无备用区可用，如由平面2上的擦除池1302中的备用区块集合不具有备用区的事实所示。因此，并非将平面2上的不良区块重新定向为指向备用区，而是将与元区块的剩余部分(平面0、1和3上的区块)相关的物理区块添加到备用区池，如由三个指向下方的箭头所示。

图14说明通过使用图11到13的数据结构来保存缺陷图的方法的实例。如果检测到或碰到缺陷，那么通过查找物理位置或物理区块编号(PBN)来确定所述缺陷的位置。如果备用区块在与缺陷区块的物理位置相同的平面中可用，那么与所述缺陷相关的元区块指向备用区，以便取代所述缺陷区块，然后更新缺陷图。图12中显示此情况。否则，如果在与缺陷区块相同的平面中不存在剩余的备用区块，那么将来自元区块的剩余的良好区块引导到备用区池。图13中显示此情况。

如上文所述，本发明的原理方面之一在于，控制器将链接表的记录保存在非易失性存储器中。此记录可能为形成对应于给定物理地址的每个元区块的区块的完整列表，例如图11到13的逻辑到物理映像表1101。或者，物理区块到元区块中的标准链接可能基于一确定性算法，且保存在非易失性存储器中的记录仅需包含与链接规则的任何偏差。由于链接因缺陷而被更新且因此与标准链接发生偏差，所以记录对应地被更新。

将所述用于将物理区块链接到元区块中的机制设计成最大化用于不良物理区块的任何分布的元区块数目，且当物理区块在系统的寿命期间失效时，允许创建新的元区块。如上文所述，根据并入元区块中的第一平面中物理区块的地址来定义元区块数目。

“标准”元区块包含物理区块，其物理区块地址为元区块数目的确定性函数。举例来说，此可为图4b中的元区块 MB_0 的配置，其还用于图5到10的实例，其中标准元区块的所有区块都位于相同的行中。可将此配置表示为 $MB_i=(i,i,i,i)$ ，其中圆括号中的第n个项目表示第n个平面中的区块所属的行。

在另一实例中，如在图4b中的元区块 MB_2 的配置中，通过元区块编号加上一偏移来给定物理区块地址，所述偏移等于物理区块所属的平面的循序编号；即元区块i由区块 $i+(\text{平面编号})$ 组成。对于由四个区块所组成的元区块来说，这可表示为 $MB_i=(i, i+1, i+2,$

i+3)。图15显示此结构，且此结构将用于论述基于算法的链接表。更一般地说，可将基于算法的确定性链接描述为

$$MB_i = (i, f_1(i, \text{平面编号}), f_2(i, \text{平面编号}), f_3(i, \text{平面编号}))$$

其针对四区块的元区块的情况，可按明显的方式扩展到其它数目。在所述表达式中， i 为行编号，且 f_1 、 f_2 和 f_3 为行与平面编号的一些函数。依照元区块编号与平面0的行编号相同的常规， MB_i 中的第一项目为 i 。对于图5到10的实例来说， $f_1=f_2=f_3=i$ ，而对于图15来说， $f_1=f_2=f_3=i+(\text{平面编号})$ 。对基于算法的链接的重新链接过程的论述将在很大程度上经历图5到10的过程的简化形式，但基于图15的标准链接。所述控制器可使用基于固件的实施方案来确定标准链接。

根据最大化标准元区块的可能数目的规则来将物理区块链接到元区块中。当链接元区块时，如果所需的物理区块可用，那么创建标准区块。如果形成标准元区块所需的任何物理区块都为不良，那么将元区块编号归类为未使用。如果形成标准元区块所需的任何物理区块都为不良，那么将元区块编号归类为未使用。在这种情况下，将其它因此而保留为未链接的可用物理区块归类为备用，如针对图15中的元区块 $n+1$ 所示。

本发明的另一方面为基于卡上的缺陷图案来最佳化标准链接图案或另外最大化可用标准元区块的数目。当格式化(或重新格式化)一卡时，可扫描不良区块的图案，且作出关于“标准”链接图案的决策，使其尽可能相互匹配。在许多情况下，用于标准链接的算法与不良区块图案的匹配将减少所需的重新链接元区块与元地址到物理地址转换的额外开销，从而(至少最初)最优化所述链接。图16和17中显示此情况的一实例。

图16和17考虑四个不良物理区块以对角形式配置的实例，每一个区块都来自平面0到3中的每一者。如图16所示，在不良区块的对角图案的情况下，标准区块的最佳图案也将为对角的。因此，在此实例中，将完全不存在重新链接的区块，因此，所述元区块都不需要通过链接表转换，且可用元区块的数目增加。图17显示当标准链接为笔直越过而非匹配不良区块图案时的等效情况。在标准方法为笔直越过的实例中，需要重新链接三个元区块($n+2$ 、 $n+3$ 和 $n+4$)。

对于每一卡，对于每一次重新格式化，或对于两者来说，标准区块图案都是独特的。这可提供额外等级的数据安全，其中每一卡都具有越过不同区块而扰乱数据的独特方法。因此，尽管标准元区块将通过使用相同的规则而形成，但其没有必要具有相同的配置。在另一实施例中，主机可能想要提供元区块链接规则的规则和代码，使得算法并非系统专用。

如图20所示，当在初始的格式化过程期间链接元区块时，无法将区块地址空间的起点附近或芯片或晶粒边界附近的一些物理区块链接到标准元区块。将这些物理区块归类为备用。如图19所述，备用的物理区块记录在备用物理区块列表中，且随后可用于形成重新链接的元区块。

可将不能作为标准区块而存在的元区块建构为重新链接的元区块。可在存储器系统的初始格式化过程期间，或响应现存元区块内的物理区块的失效而形成重新链接的元区块。不能将图19中的元区块 $n+4$ 作为标准元区块来链接，因为平面2中的所需的物理区块为不良。在此情况下，通过取代平面2中的备用物理区块来创建重新链接的元区块。代替不良物理区块而重新链接的备用物理区块优选位于与不良区块相同的平面中，以保持元区块内的编程并行度。

在优选实施例中，优选仅在相同的晶粒或芯片内重新链接区块，因为越过到另一晶粒或芯片的重新链接可能触发额外的晶粒/芯片读取或编程。举例来说，在4平面元区块的示范性情况下，其中元区块横跨2个芯片，每个芯片具有2个平面，可将数据编程到所有4个平面(或从其读取)，这需要并行的2芯片读取/编程。(此可能不是100%并行的，而是管线式的，使得页面读取、区块编程和擦除可并行地进行，但数据总线被共享，使得数据传递为不并行)。在这种情况下，如果重新链接元区块，使得物理区块从另一、第三芯片获取，那么不可能与另外两个芯片同时存取所述元区块，且操作时间可能加倍。

如果通过来自平面0的备用区块来取代平面0中的不良物理区块，那么可通过备用区块的编号来定义元区块编号，如针对图20中的元区块 $n+1$ 所示。如果随后的物理区块失效，那么可进一步重新链接已重新链接的元区块。靠近地址空间顶部的元区块(其无法作为标准元区块而链接)可采取重新链接的元区块的形式。

如上文关于图4d所述，还存在使用二维元区块的配置，所述二维元区块不仅跨越平面和芯片链接区块，而且还在给定平面内沿深度方向链接区块。图21显示在二维情况下重新链接的实例。在图21中，所述“2-D元区块”为 $2*4$ 元区块的实例。当能够并行地存取所有8个区块时，可达成最大性能益处。1-D链接方法的主链接和重新链接原理也适用，但需作一些扩展。如在图式中可看出，一平面内任何物理区块中的不良物理区块都会导致元区块未使用，如针对元区块 n 所示的情况。当重新链接2-D元区块时，如对于平面2中的元区块 $n+2$ 的上部区块，仅一给定平面中的不良物理区块需要被取代；例如，平面2中的元区块 $n+2$ 的上部区块被重新链接，而平面2中的元区块 $n+2$ 的下部区块却未重新链接。

在其中仅将与标准链接的偏差保存在非易失性存储器中一记录中的实施例中，用于元区块链接的管理数据结构可包含在闪存中一个或一个以上专用区块链接管理区块中的两种类型的扇区内。此配置的优势在于，相对于保存完整链接，仅需存储较少的信息。存储器系统中每一重新链接的元区块的元区块编号包含在一组链接表扇区中的一链接表中。不需要将标准元区块的项目存储在链接表中，因为它们的物理区块地址从元区块编号中确定性地产生。可将RF(重新链接)旗标用作元区块的属性，以指示所述区块是否为标准的，且是否所有区块以标准方式来链接，或所述区块是否重新链接且需要通过链接表(LT)来进行额外的区块地址转换。可通过分析LT来确定RF旗标，且接着将其存储在SRAM中，或将其存储在主非易失性存储器中的控制数据结构中作为元数据区块的属性。或者，可将重新链接的扇区的列表保存在可遍历搜索以确定哪些元区块为重新链接的列表中。可通过元区块编号来排序链接表中的项目，且每个项目都具有用于元区块编号和每个经链接物理区块的区块地址的字段。

在链接表结构的扩展中，与建立标准链接的基本算法的偏差本身可基于一组规则；例如，非标准链接可限于单个平面或限制一区块可从其原始行重新链接到什么程度。仅在重新链接为所有算法的例外时，才需要将其保存在非易失性存储器中。

关于图22和表1来描述链接表和可如何压缩其所含信息的特定实例。链接表扇区保存关于卡内已重新链接的区块的信息。根据卡的大小和每个晶粒内不良区块的数目，可能会存在多于一个的链接表扇区，可将其存储在引导(Boot Block)区块中。如上文所述，元区块具有用于链接的标准方法，且如果所述区块中的一个区块为不良，那么元区块能够使用相同平面中的备用物理区块来重新链接。链接表扇区存储关于哪些区块用于重新链接的信息。图22为元区块的重新链接的另一实例。

关于图22所示的元区块，仅元区块0具有标准链接，此处理解为笔直地越过八个平面。图式显示数种区块链接情况：

- A. 元区块0为不具有重新链接的标准元区块，且将因此不出现在链接表扇区中
- B. 元区块1具有在平面0*和1中的2个不良区块
- C. 元区块4具有在平面0*和5中的2个不良区块
- D. 元区块5具有在平面1、3和4中的3个不良区块
- E. 元区块7具有在平面0*中的1个不良区块
- F. 已从现存的备用物理区块完整地重新链接元区块9，尽管平面0和7中的区块来自原始的元区块

在标为*的情况下，重新链接由于平面0中的不良区块而发生。在这种情况下，物理地址指代原始元区块存在的位置。由于图22的实例中的大量不良区块，所以不存在元区块2、3、6或7。

用于图22所示的区块重新链接的压缩算法可如表1中那样定义。

图中的元区块	元区块数目	项目中重新链接的项的数目	匹配项数的重复值		项目中的总字节
			平面	物理地址	
	2字节	1字节	1字节	2字节	
1	0x0001	0x02	0x00	0x0002	
			0x01	0x0005	9
4	0x0004	0x02	0x00	0x0003	
			0x05	0x0001	9
5	0x0005	0x03	0x01	0x0004	
			0x03	0x0004	
			0x04	0x0007	12
7	0x0007	0x01	0x00	0x0006	6
9	0x0009	0x06	0x01	0x0008	
			0x02	0x0007	
			0x03	0x0007	
			0x04	0x0008	
			0x05	0x0008	
			0x06	0x0007	21

表1

任何链接表项目的大小可定义为：

字节的数目=3+(3*重新链接的区块的数目)

举例来说，对于元区块1来说，前三个字节识别元区块1，且所述元区块1具有两个重新链接，且分配三个字节以指定这些重新链接中的每个重新链接。通过所述重新链接的区块对应于八个平面中的那个平面(1字节)和其所重新链接到的行编号来指定所述重新链接。

链接表中的全组项目可越过多个链接表扇区而展开。创建一链接表扇区，或通过所述连接表扇区写入区块链接管理区块中的下一可用扇区位置而对其加以修改。因此可能存在链接表扇区的多个副本，其中仅最新版本才有效。区块链接管理区块中的有效链接表扇区的位置的索引可包含在区块链接管理区块中的备用物理区块列表扇区中的一字段中，在下文中对其进行描述。与每个有效链接表扇区中第一项目有关的元区块编号可保存在控制器RAM中的一列表中，以便允许容易地识别任何元区块编号的相关链接表扇区。如果必须将额外的项目添加到已包含最大数目的项目的链接表扇区中，那么将所述扇区分成两个新的扇区，每个扇区具有所述项目的一半，且将新的项目插入适当的新扇

区中。

将所有备用物理区块的区块地址作为项目保存在备用物理区块列表扇区中未排序的列表中。在一些实施例中，限制每芯片每平面的备用区块的最大数目可能较方便。在示范性实施例中，将此限制于16，如在此应为足够大的大多数情况下。如果在一平面中存在多于16个的备用区块，那么它们将会“丢失”。

由于最初在将闪存配置为元区块的期间，无法创建标准元区块，或由于链接到标准或重新链接的元区块中的物理区块已失效而变成备用的区块作为新的项目添加到备用物理区块列表的末端。当创建重新链接的元区块需要备用区块时，可使用相关平面中最靠近备用物理区块列表的起点的第一区块。备用物理区块列表扇区还包含具有有效链接表扇区位置的索引的字段。

当发生元区块重新链接操作且创建或修改链接表扇区时，备用物理区块列表和链接表扇区索引字段两者都可在备用物理区块列表扇区中修改。因此有效备用物理区块列表扇区为区块链接管理区块中最后写入的扇区。当区块链接管理区块已满时，可在控制写入操作期间通过将所有有效扇区重新写入到新的区块位置来对其进行压缩，随后将整个区块擦除。

为了简化备用物理区块扇区的保存并限制寻找取代区块时所需的搜索量，备用物理区块列表可由芯片、晶粒和平面存储。所需扇区的实际数目取决于存储器装置的大小、装置的数目和所使用的交错方法。如上文所述，每平面仅存储某一数目的物理区块可能较方便。所保存的数目将取决于预期的区块失效率和每平面的区块数目。这意味着，每个备用物理区块列表扇区将保存数个平面的备用物理区块。固件将计算要加载的正确扇区，且接着将正确的值编入要加载的扇区的索引中。

当备用物理区块列表的数目由于重新链接或找到另一不良区块而发生变化时，新的备用物理区块列表扇区会写入。这可意味着覆盖相同物理区域的两个备用物理区块列表扇区。引导扇区保存包含有效扇区的索引的备用物理区块列表索引。如果由备用物理区块列表所覆盖的区域中不存在备用物理区块，那么引导区块中将不存在扇区。

回头参看图22，此涉及8路交错卡。因此，将必须存在至少8个阵列，用于保存所有的备用物理区块列表信息。对于卡上的每个后续晶粒来说，将需要另外8个阵列。表2中显示对应于图22的备用物理区块列表。

平面			
0	0x08		
1			
2	0x01	0x04	0x08
3	0x01	0x08	0x09
4	0x01	0x04	0x05
5	0x04	0x07	0x09
6	0x01	0x04	0x08
7	0x01	0x04	0x07

表2

与链接表一样，可将备用物理区块列表作为图表存储在非易失性存储器中，所述图表可根据需要高速缓存到易失性RAM中。

尽管已经根据基于固件的实施方案描述了本发明的各个方面，但它们还可以基于硬件的实施方案来实施。位于控制器上，或优选位于存储器芯片本身上的硬件引擎可执行重新链接。重新链接信息可存储在芯片本身上一特殊的存储器位置处。所述控制器接着可将所有区块作为标准区块来处理，且所述引擎会将区块地址转换成真实的内部地址。所述存储器在主机(或控制器)看来为仅具有标准区块的存储器，或为具有较大物理区块的存储器，而不必了解较小的区块结构或任何重新链接的细节。

尽管已经描述了本发明各个方面的特定实例，但应了解，本发明有权在所附权利要求书的范围内受到保护。

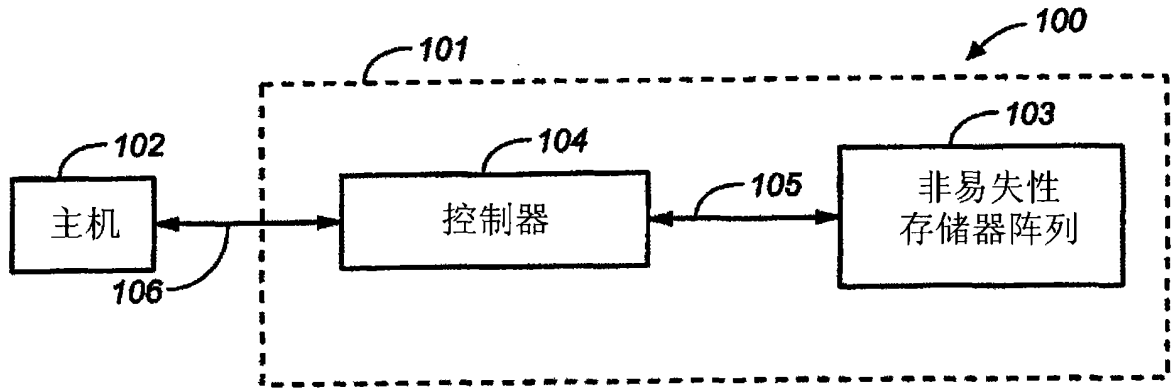


图1

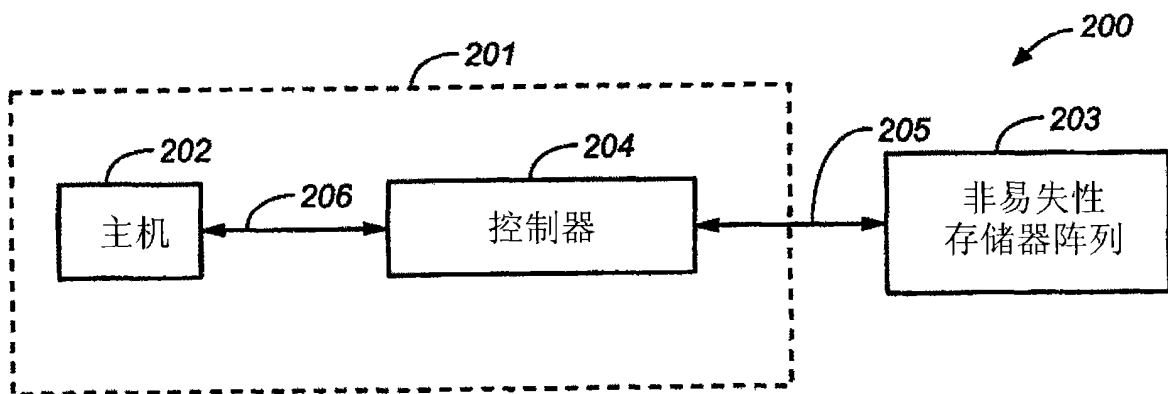


图2

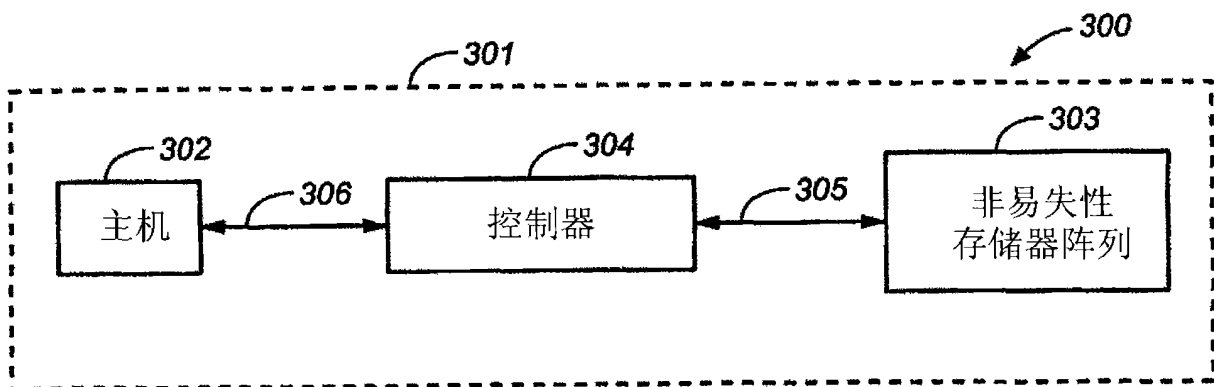


图3

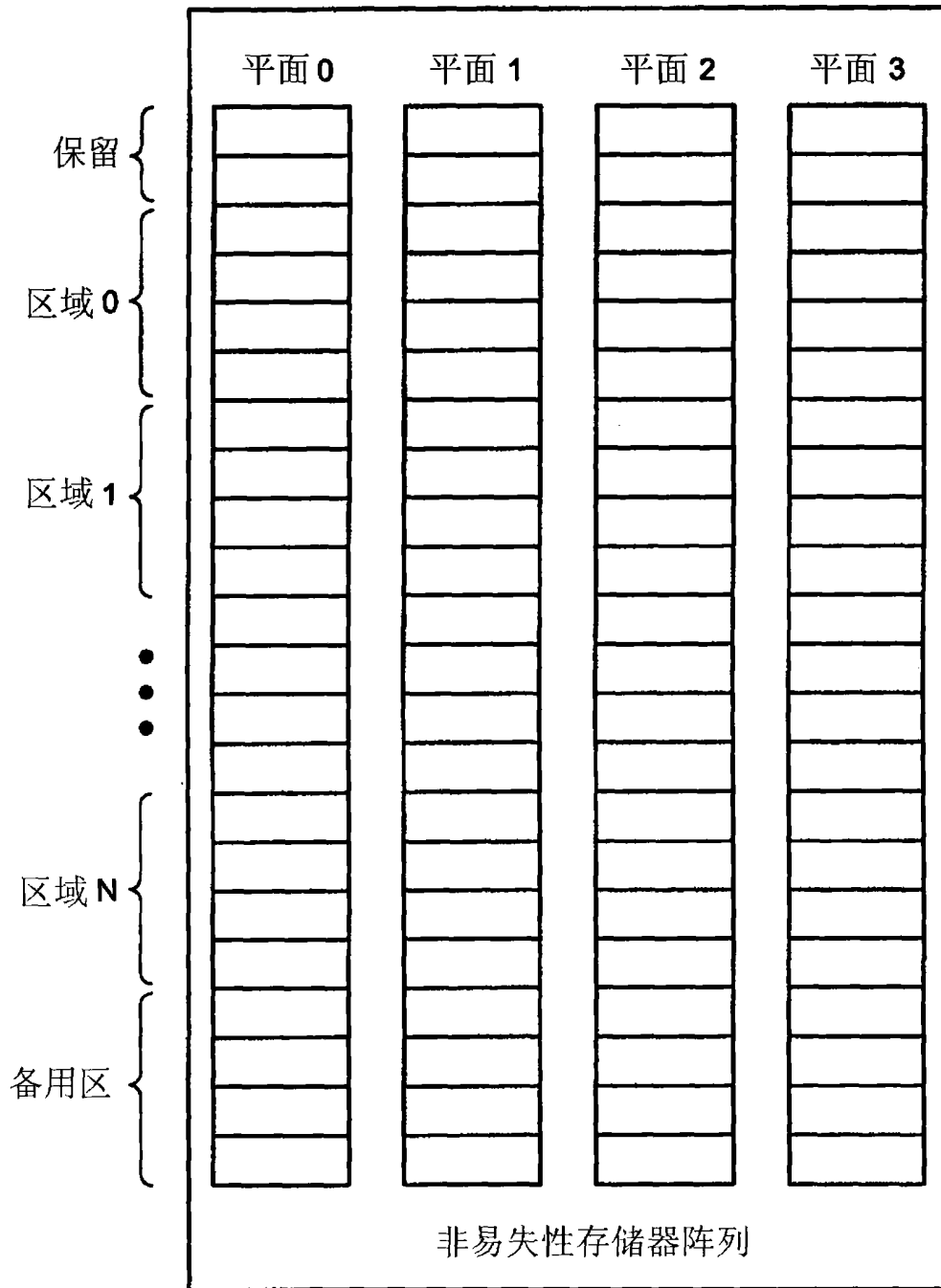


图4a

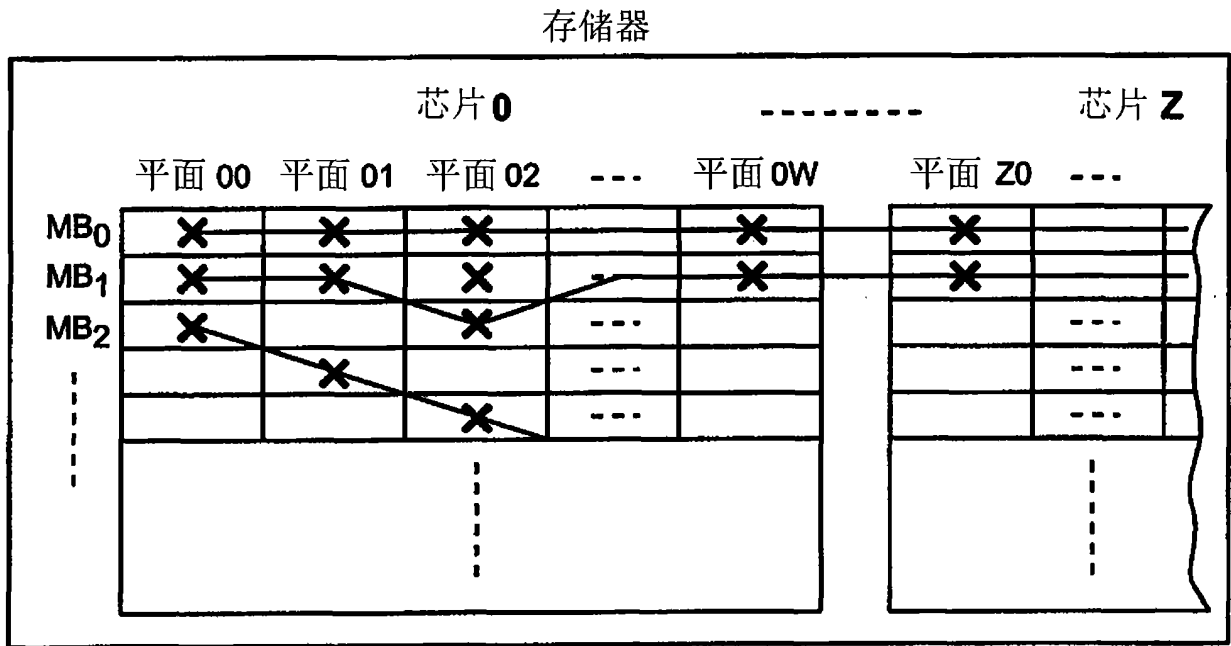


图4b

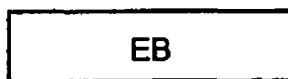


图4c

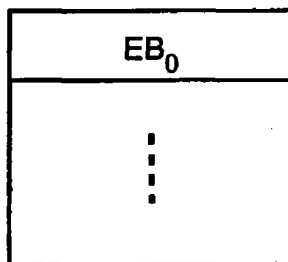


图4d

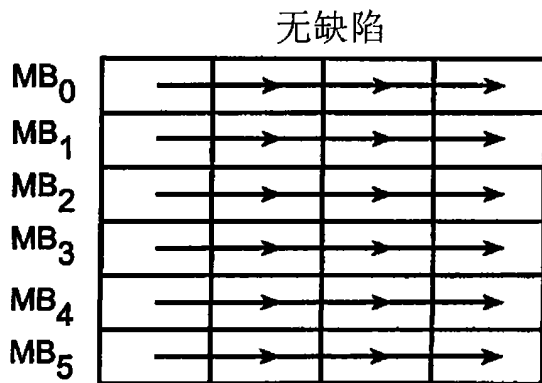


图5

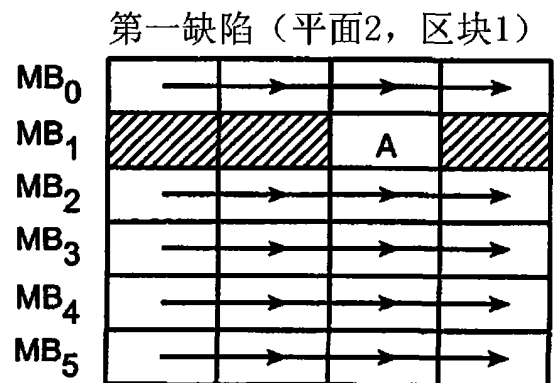


图6

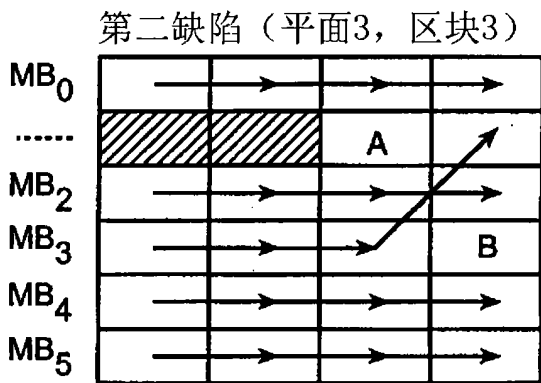


图7

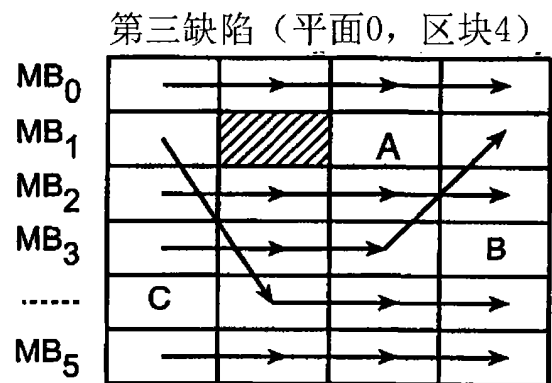


图8

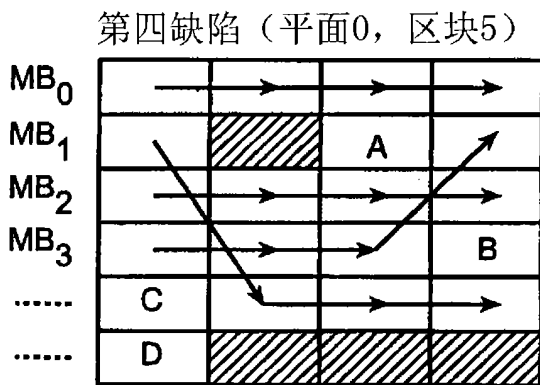


图9

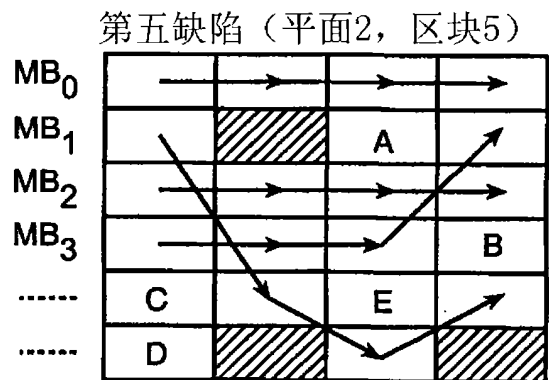


图10

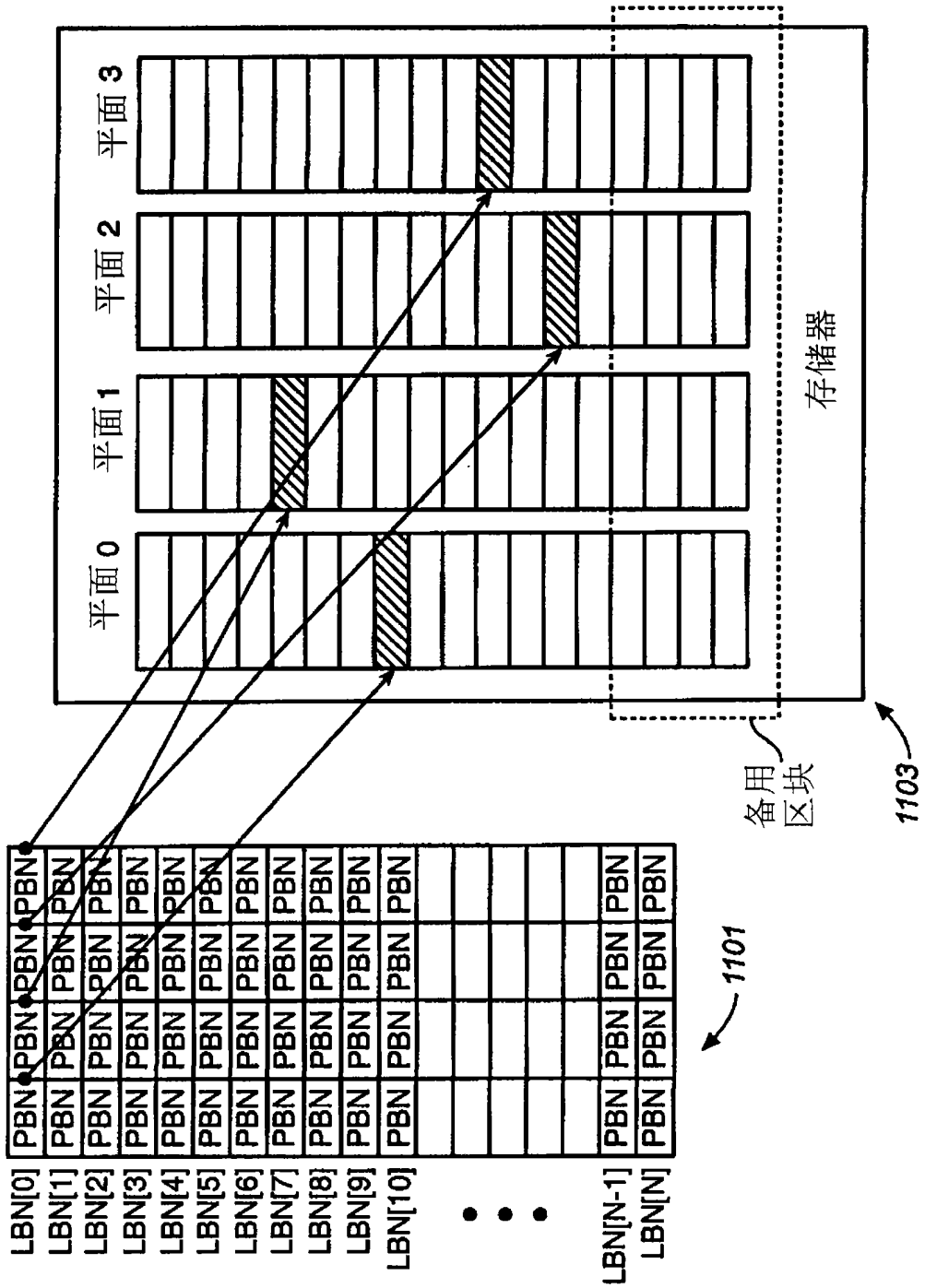


图11

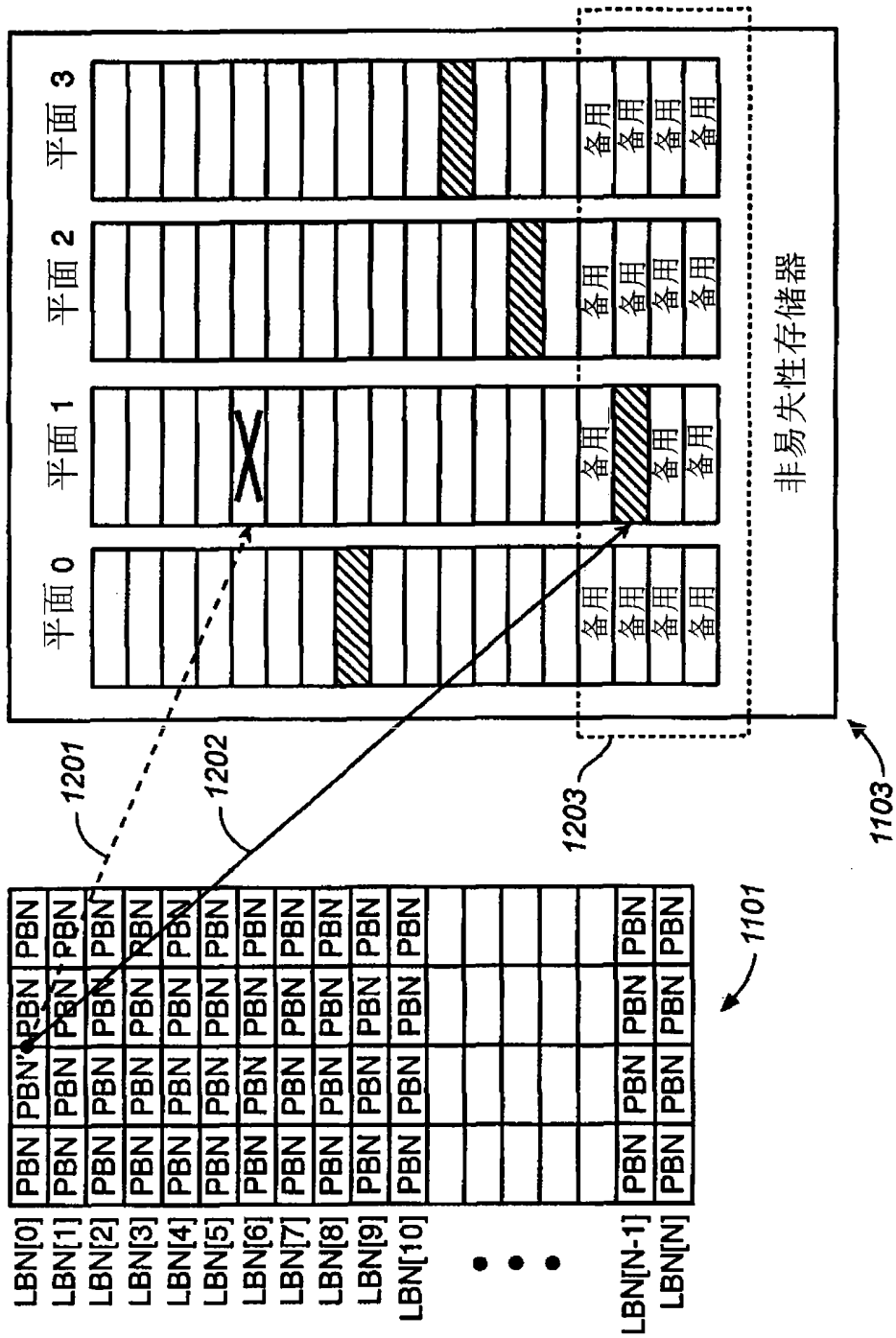


图12

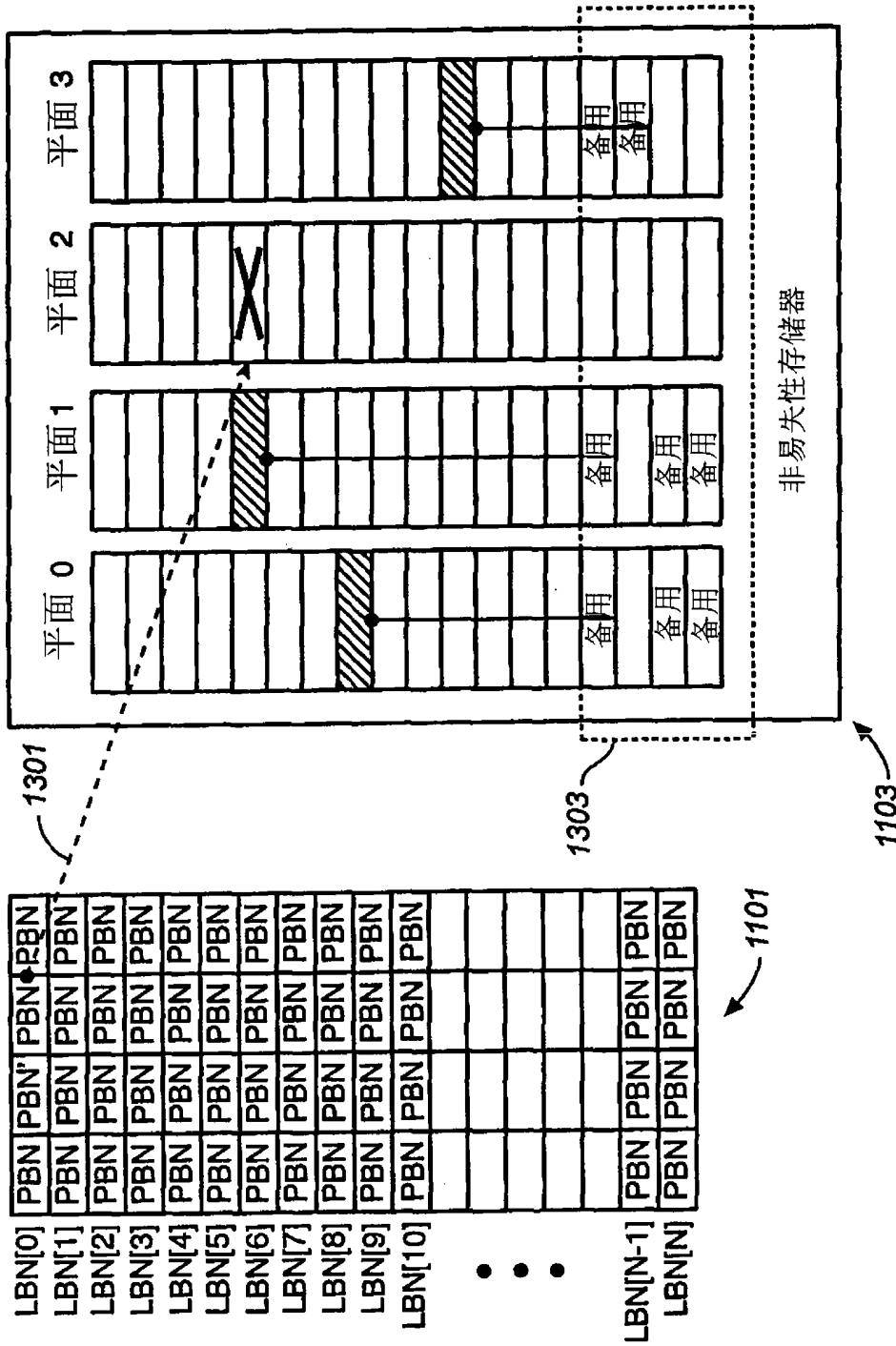


图 13

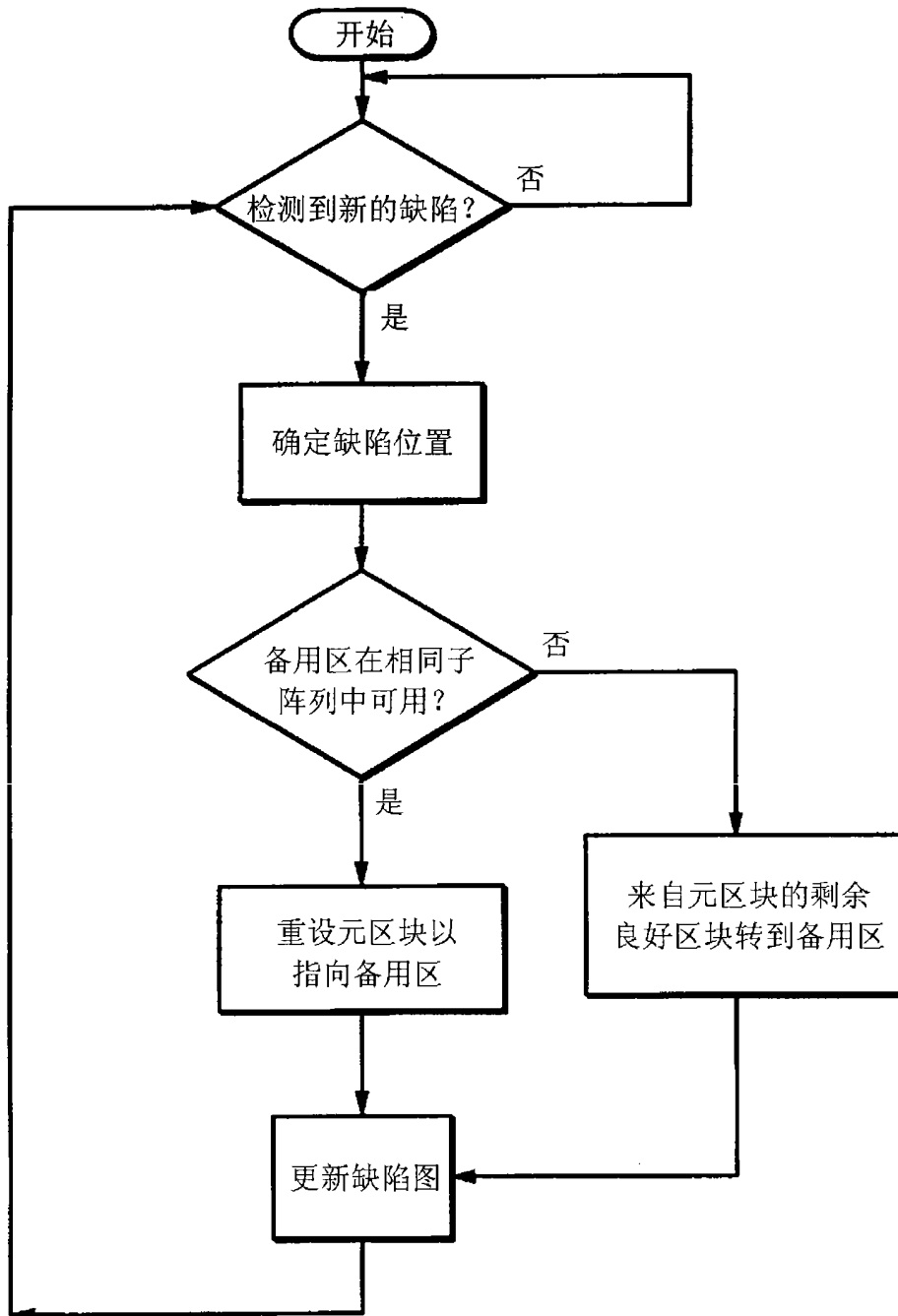


图14

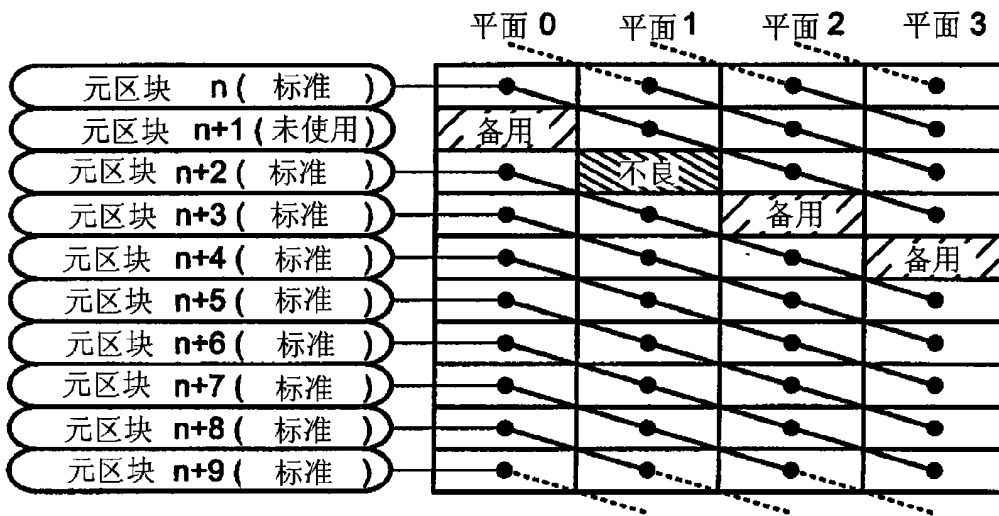


图15

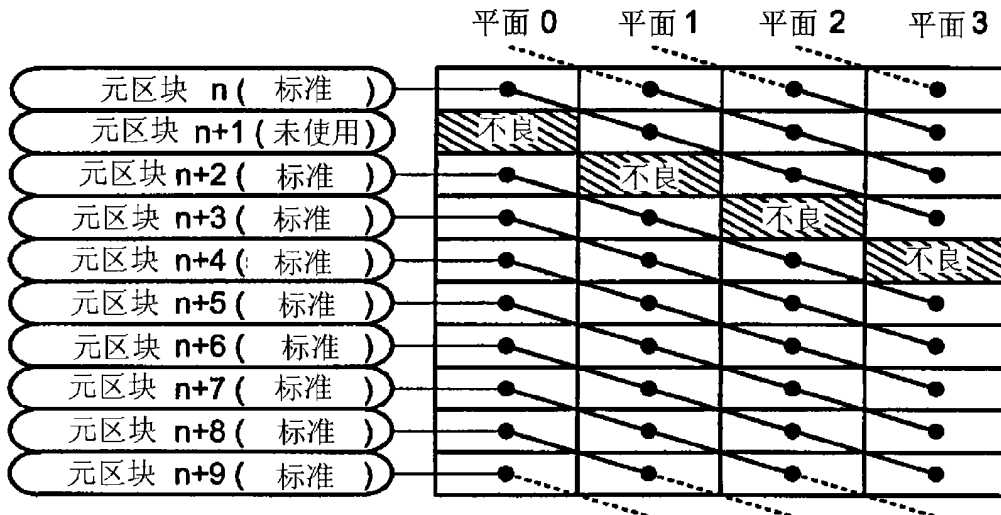


图16

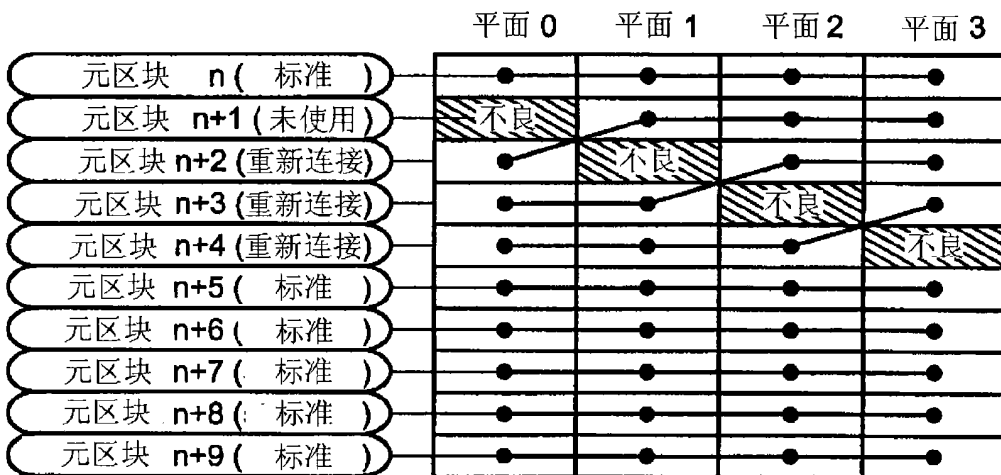


图17

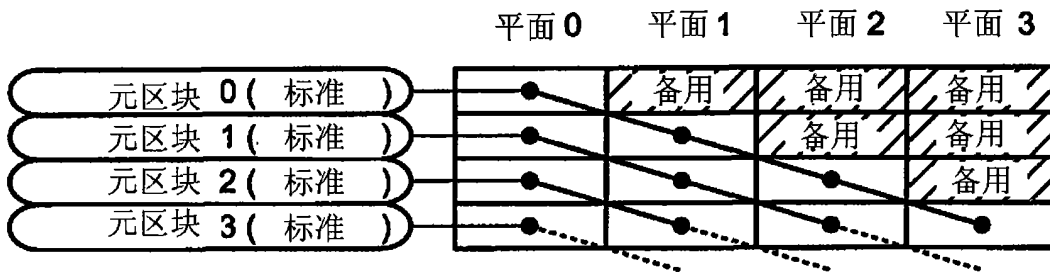


图18

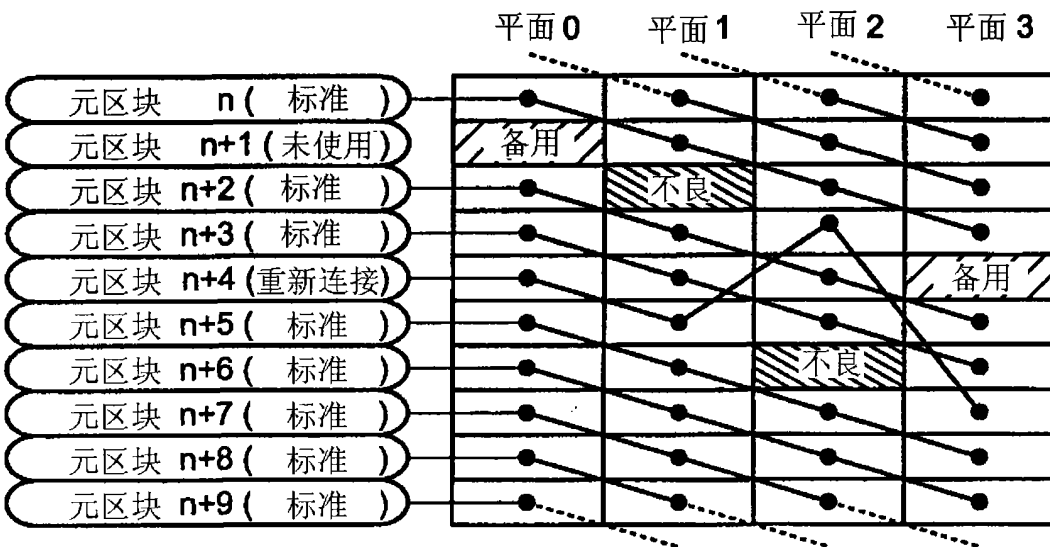


图19

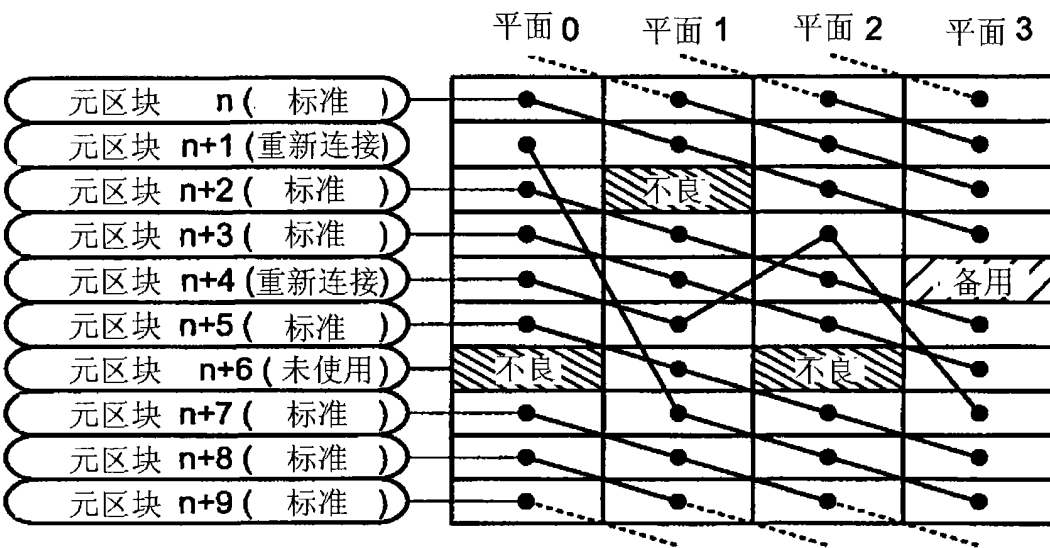


图20

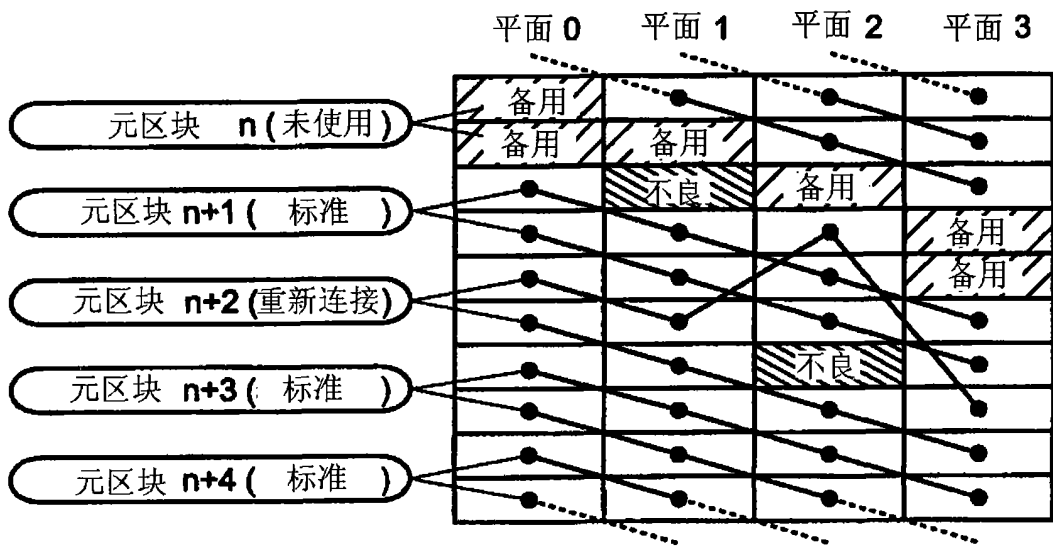


图21

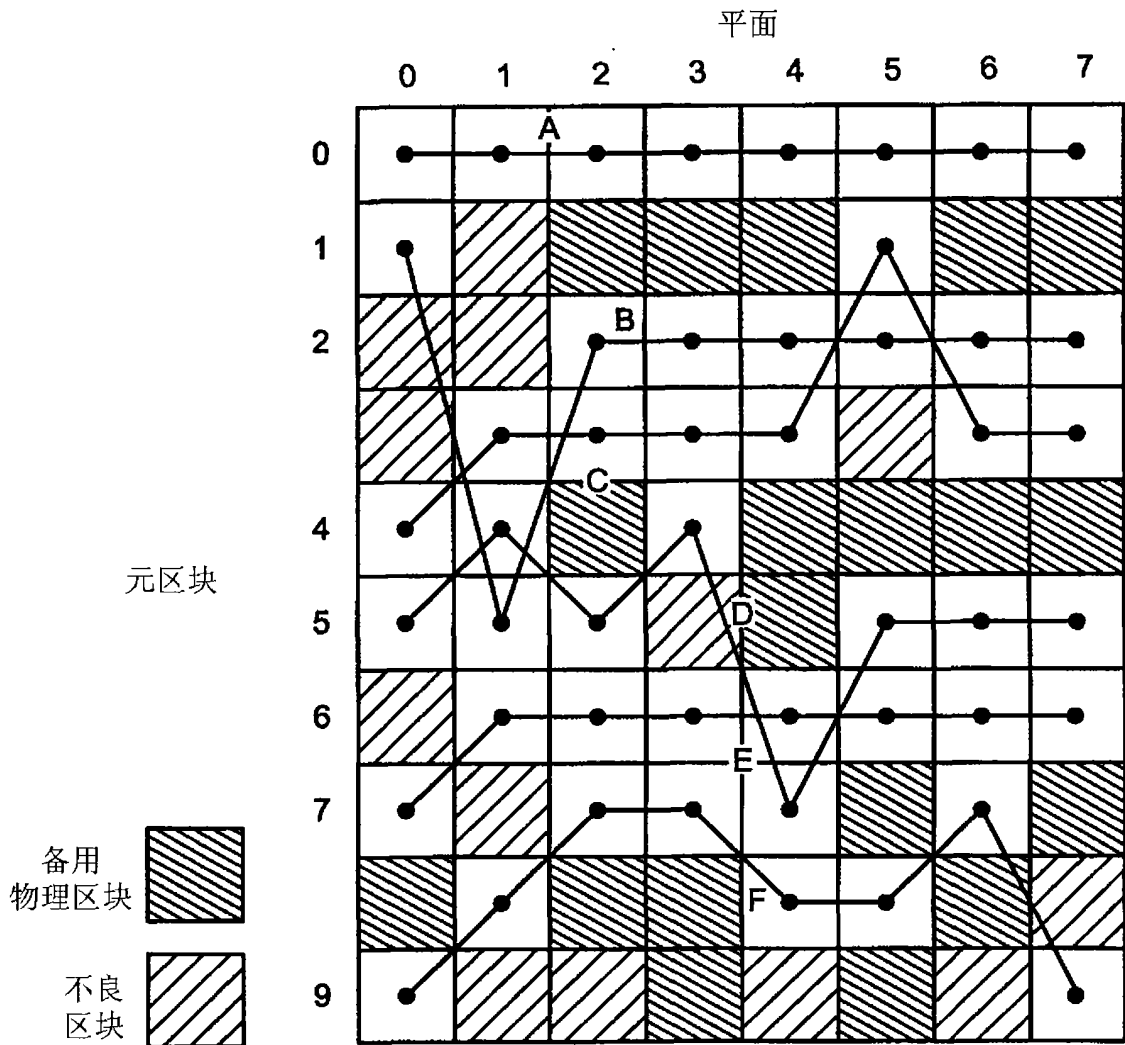


图22