[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 13/14
G06F 11/00

[12] 发明专利说明书

[21] ZL 专利号 00814956.9

[45] 授权公告日 2004 年 12 月 8 日

[11] 授权公告号 CN 1179281C

[22] 申请日 2000.10.25 [21] 申请号 00814956.9
[30] 优先权
[32] 1999.10.26 [33] FI [31] 19992308
[86] 国际申请 PCT/FI2000/000921 2000.10.25
[87] 国际公布 WO2001/031458 英 2001.5.3
[85] 进入国家阶段日期 2002.4.26
[71] 专利权人 特拉博斯股份有限公司
地址 芬兰埃斯波
[72] 发明人 马泰洛·雅利
审查员 马晓亚

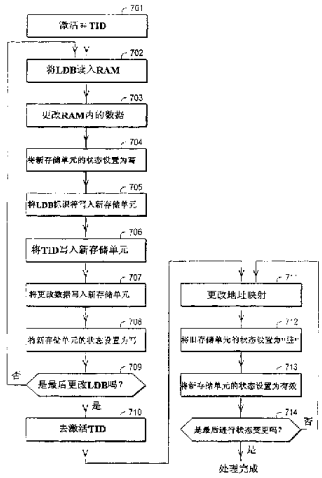
[74] 专利代理机构 中原信达知识产权代理有限责任公司
代理人 谷惠敏 袁炳泽

权利要求书 9 页 说明书 20 页 附图 10 页

[54] 发明名称 利用逻辑闪速存储器件进行最小单元更新的方法和设备

[57] 摘要

本发明披露了一种通过执行必须或者全部一起成功或者全部一起失败的许多更新操作来更新第一数字数据块(601)的方法和设备。更新操作的数量大于或等于 1。启动(701)对涉及第一数据块并且必须或者全部一起成功或者全部一起失败的所有更新操作通用的第一标识符(TID)。执行(703)所有那些涉及第一数据块并且必须或者全部一起成功或者全部一起失败的更新操作。将更新第一数据块与使更新第一数据块和所述第一标识符相关的数值一起存储(707)到逻辑闪速存储器件(1101)。关闭(710)所述第一标识符。



ISSN1008-4274

知识产权出版社出版

1. 一种通过执行多个更新操作更新第一数字数据块（601）的方法，上述所执行的多个更新操作必须全部一起成功或者全部一起失败，上述所执行的多个更新操作的数目大于或等于 1，其特征在于，该方法包括下列顺序的步骤：

- 激活（701）对涉及第一数据块并且必须全部一起成功或者全部一起失败的所有这些更新操作通用的第一标识符（TID）；
- 执行（703）涉及第一数据块并且必须全部一起成功或者全部一起失败的所有更新操作；
- 将更新后的第一数字数据块和使更新后的第一数据块与所述第一标识符相关的数值存储（707）到逻辑闪速存储器件（1101）；以及
- 去激活（710）所述第一标识符。

2. 根据权利要求 1 所述的方法，其特征在于，激活（701）第一标识符的步骤包括将所述第一标识符存储到被分配用于存储有效标识符的非易失性存储单元（850）的分步骤。

3. 根据权利要求 2 所述的方法，其特征在于，激活第一标识符的步骤包括分步骤：

a) 将被分配用于存储有效标识符的第一非易失性存储单元（501）的内容拷贝读入（551）工作空间存储器（503）；

b) 利用待激活的第一标识符（604）补充（552）被读入工作空间存储器（503）的拷贝，从而产生更新拷贝；

c) 将第二非易失性存储单元（502）的状态指示符（522）设置（553）为表示该存储单元进行写操作的第一数值；

d) 将所述第二非易失性存储单元（502）内的数据块标识符（521）设置（554）为与所述第一非易失性存储单元（501）内的相应数据块标识符（511）相同；

e) 将所述更新拷贝写入（555）所述第二非易失性存储单元（502）；

f) 将所述第二非易失性存储单元（502）的所述状态指示符设置（556）为表示该存储单元含有有效数据的第二数值；

g) 更新（557）地址映射实体以使所述第二非易失性存储单元（502）内的所述数据块标识符（521）与所述第二非易失性存储单元（502）的物理地址相关；以及

h) 将所述第一非易失性存储单元（501）的状态指示符（512）设置为表示该存储单元含有无效数据的第三数值。

4. 根据权利要求 3 所述的方法，其特征在于，分步骤 e)包括将循环指示符数值与所述更新拷贝一起写入所述第二非易失性存储单元（502）的子分步骤，在所述第一非易失性存储单元（501）内，将循环指示符数值从相应循环指示符数值增加 1。

5. 根据权利要求 2 所述的方法，其特征在于，去激活（710）所述第一标识符的步骤包括从被分配用于存储有效标识符的所述非易失性存储单元（850）内删除所述第一标识符的分步骤。

6. 根据权利要求 5 所述的方法，其特征在于，去激活所述第一标识符的方法包括分步骤：

a) 将被分配用于存储有效标识符的第二非易失性存储单元（502）的内容拷贝读入（551）工作空间存储器（503）；

b) 从读入工作空间存储器（503）的拷贝中删除（552）待去激活的第一标识符，从而产生更新拷贝；

c) 将第三非易失性存储单元（502）的状态指示符设置（553）为表示该存储单元进行写操作的第一数值；

d) 将所述第三非易失性存储单元内的数据块标识符设置（554）为与所述第二非易失性存储单元（502）内的相应数据块标识符相同；

e) 将所述更新拷贝写入（555）所述第三非易失性存储单元；

f) 将所述第三非易失性存储单元的所述状态指示符设置（556）为表示该存储单元含有有效数据的第二数值；

g) 更新（557）地址映射实体以使所述第三非易失性存储单元内的所述数据块标识符与所述第三非易失性存储单元的物理地址相关；以及

h) 将所述第二非易失性存储单元（502）的状态指示符（522）设置为表示该存储单元含有无效数据的第三数值。

7. 根据权利要求 6 所述的方法，其特征在于，分步骤 e)包括将循环指示符数值与所述更新拷贝一起写入其循环指示符数值从所述第二非易失存储单元（502）内的相应循环指示符数值增加 1 的所述第三非易失性存储单元的子分步骤。

8. 根据权利要求 2 所述的方法，其特征在于，去激活（710）所述第一标识符的步骤包括将所述第一标识符存储到被分配用于存储去激活标识符的非易失性存储器内的分步骤。

9. 根据权利要求 1 所述的方法，其特征在于，进行更新操作和存储更新第一数字数据块的步骤包括分步骤：

a) 将被分配用于存储待更新数据的第一非易失性存储单元的内容拷贝读入（701）工作空间存储器；

b) 更新（703）读入工作空间存储器的拷贝，从而产生更新的拷贝；

c) 将第二非易失性存储单元的状态指示符设置（704）为表示该存储单元进行写操作的第一数值；

d) 将所述第二非易失性存储单元内的数据块标识符设置（705）为与所述第一非易失性存储单元内的相应数据块标识符相同；

e) 将所述激活第一标识符写入（706）所述第二非易失性存储单元；

f) 将所述更新的拷贝写入（707）所述第二非易失性存储单元；

g) 将所述第二非易失性存储单元的所述状态指示符设置（708）为第二数值，所述第二数值表示该存储单元含有已经被所述多个更新操作的一部分操作而更新的数据，上述多个更新操作必须全部一起成功或者全部一起失败；以及

h) 检验（709）所述多个更新操作是否进行了，上述多个更新操作必须全部一起成功或者全部一起失败，如果没有进行的话，重新开始步骤 a)，从而在步骤 c)、d)、e)、f)和 g)涉及到不同的第二非易失性存储单元。

10. 根据权利要求 9 所述的方法，其特征在于，该方法还包括分步骤：

i) 更新（711）地址映射实体从而使含有存储更新拷贝的非易失性存储单元内的各数据块标识符与非易失性存储单元的物理地址相关；

j) 将经过分步骤 a)的各非易失性存储单元的状态指示符设置（712）为表示该存储单元含有有效数据的第三数值；

k) 将各第二非易失性存储单元的所述状态指示符设置（713）为表示该存储单元有效数据的第四数值。

11. 根据权利要求 1 所述的方法，其特征在于，在激活（701）对必须全部一起成功或者全部一起失败的所有更新操作通用的第一标识符的步骤与去激活（710）所述第一标识符的步骤之间，还包括步骤：

- 执行（703）更新操作以使得它们与其它数据块而不是第一数据块有关系，并且与第一数据块的所述更新操作必须全部一起成功或者全部一起失败；以及
- 将更新后的数字数据的各个块与使更新后的各数据块和所述第一标识符相关的数值存储（707）到逻辑闪速存储器件内。

12. 根据权利要求 1 所述的方法，其特征在于，执行所有涉及第

一数据块的并且必须全部一起成功或者全部一起失败的所有更新操作步骤包括选择第一数据块的第一部分（A1）作为涉及所述第一标识符（TID＃A）的更新操作的目标的步骤。

13. 根据权利要求 12 所述的方法，其特征在于，该方法还包括步骤：

- 激活第二标识符（TID＃B），上述第二标识符对涉及第一数据块（800）的并且必须全部一起成功或者全部一起失败的，但是不要求与涉及所述第一标识符（TID＃A）的更新操作一起成功或者一起失败的多个更新操作是通用；
- 选择第一数据块的第二部分（B1）作为涉及所述第二标识符（TID＃B）的更新操作目标，第二部分就不会与所述第一部分（A1）重叠；
- 执行涉及第一数据块、与第二标识符（TID＃B）相关并且必须全部一起成功或者全部一起失败的所有更新操作；
- 将与所述标识符（TID＃A）之一相关并且在与其它所述标识符（TID＃B）之一相关的更新操作之前完成的更新操作的更新结果（A2）变换为利用与其它所述标识符之一相关的更新操作操作的数据块的拷贝；
- 将更新后的第一数字数据块与使更新后的第一数据块和所述第二标识符（TID＃B）相关的数值存储到逻辑闪速存储器件；以及
- 去激活所述第二标识符（TID＃B）。

14. 根据权利要求 13 所述的方法，其特征在于，将更新后的第一数字数据块与使更新后的第一数据块和所述第一标识符（TID＃A）相关的数值存储到逻辑闪速存储器件的步骤以及将更新后的第一数字数据与使更新后的第一数据块和所述第二标识符（TID＃B）相关的数值存储到逻辑闪速存储器件的步骤包括分步骤：

a) 将第一非易失性存储单元（810）的状态指示符设置为表示该存储单元进行写操作的第一数值（W）；

b) 将与所述标识符之一（TID＃A）相关并且在完成与其它所述标识符之一（TID＃B）相关的更新操作之前完成的更新操作产生的这些更新拷贝写入所述第一非易失性存储单元（810）；

c) 还将与产生在分步骤 b)写入的更新拷贝的更新操作相关的标识符（TID＃A）写入所述第一非易失性存储单元（810）；

d) 将所述第一非易失性存储单元（810）的所述状态指示符设置为表示该存储单元含有作为必须全部一起成功或者全部一起失败的多个更新操作的一部分被更新的数据的第二数值（T）；

e) 去激活与产生在分步骤 b)写入的更新拷贝的更新操作相关的标识符（TID＃A）；

f) 将其内的数据块被作为更新目标的非易失性存储单元（800）的状态指示符设置为表示该存储单元含有无效数据的第三数值（D）；

g) 将所述第一非易失性存储单元（810）的所述状态指示符设置为表示该存储单元含有用户数据的第四数值（V）；

h) 将第二非易失性存储单元（820）的状态指示符设置为表示该存储单元进行写操作的第一数值（W）；

i) 将与剩余有效所述标识符之一（TID＃B）相关的更新操作产生的更新拷贝写入所述第二非易失性存储单元（820）；

j) 还将与产生在分步骤 i)写入的更新拷贝的更新操作相关的标识符（TID＃B）写入所述第二非易失性存储单元（820）；

k) 将所述第二非易失性存储单元（820）的所述状态指示符设置为第二数值，上述第二数值表示该存储单元含有被多个更新操作的一部分操作更新的数据，上述多个更新操作必须全部一起成功或者全部一起失败；

l) 去激活与产生在分步骤 i)写入的更新拷贝的更新操作相关的标识符（TID＃B）；

m) 将所述第一非易失性存储单元（810）的所述状态指示符设置为表示该存储单元含有无效数据的第三数值（D）；以及

n) 将所述第二非易失性存储单元（820）的所述状态指示符设置为表示该存储单元含有有效数据的第四数值（V）。

15. 一种在基于逻辑闪速存储器件的文件系统内排除系统故障的方法，其特征在于，该方法包括步骤：

• 从有效标识符表（850）内检验哪些标识符在发生系统故障的瞬间是有效的；以及

• 检验物理存储单元以定位含有使它们与发生系统故障瞬间有效的标识符相关的数值的数据块，并将含有这种数据块的各非易失性存储单元的状态指示符设置为表示该存储单元含有无效数据的数值。

16. 根据权利要求 15 所述的方法，其特征在于，该方法还包括步骤：

• 检验物理存储单元以定位数据块对；其中

• 同一个数据块标识符（602）出现在该数据块对的各数据块内；

• 数据块对内没有数据块与发生系统故障瞬间的有效标识符相关；

• 将数据块对内的第一数据块存储到具有表示该存储单元含有有效数据的状态指示符（603）的存储单元内；以及

• 将数据块对内的第二数据块存储到具有表示该存储单元含有作为必须全部一起成功或者全部一起失败的多个更新操作的一部分被更新的数据的状态指示符（603）的存储单元内；

• 将含有数据块对内的第一数据块的每个这种非易失性存储单元的状态指示符（603）设置为表示该存储单元含有无效数据的数值（D）；以及

• 将含有数据块对内的第二数据块的每个这种非易失性存储单元的状态指示符（603）设置为表示该存储单元含有有效数据的数值（V）。

17. 根据权利要求 15 所述的方法，其特征在于，该方法还包括步骤：

- 检验物理存储单元以定位数据块，上述数据块：
- 与发生系统故障瞬时有效的标识符无关并且
- 存储在具有表示该存储单元含有作为必须全部一起成功或者全部一起失败的多个更新操作的一部分被更新的数据的状态指示符（603）的存储单元内；以及
- 将含有这种数据块的每个这种非易失性存储单元的状态指示符（603）设置为表示该存储单元含有有效数据的数值（V）。

18. 根据权利要求 15 所述的方法，其特征在于，在有效标识符表（850）内检验哪些标识符在发生系统故障瞬间是有效的步骤包括分步骤：

a) 在逻辑闪速存储器件内定位含有有效标识符表和表示该存储单元含有有效数据的状态指示符的所有非易失性存储单元；

b) 如果在分步骤 a)仅定位一个所述类型的非易失性存储单元，则选择其内的有效标识符表作为有效标识符的有效表；以及

c) 如果在分步骤 a)定位不止一个所述类型的非易失性存储单元，则把定位的非易失性存储单元内的有效标识符表选为有效标识符的有效表。

19. 根据权利要求 18 所述的方法，其特征在于，分步骤 c)含有子分步骤：

c1)从每个定位的非易失性存储单元内读取循环指示符数值；以及

c2)在该定位的非易失性存储单元内选择有效标识符表作为与最高级循环指示符相关的有效标识符的有效表。

20. 一种通过执行必须全部一起成功或者全部一起失败的多个更新操作来更新第一数字数据块的设备（1100），所述许多为大于或等于 1，其特征在于，该设备包括：

- 逻辑闪速存储器件（1101）；

- 装置（1102、1103、1104、1105），用于激活对涉及第一数据块并且必须全部一起成功或者全部一起失败的所有更新操作通用的第一标识符；
- 装置（1102、1103、1104、1105），用于执行涉及第一数据块并且必须全部一起成功或者全部一起失败的所有更新操作；
- 装置（1102、1103、1104、1105），用于将更新后的第一数字数据块与使更新后的第一数据块和所述第一标识符相关的数值存储到所述逻辑闪速存储器件（1101）内；以及
- 装置（1102、1103、1104、1105），用于去激活所述第一标识符。

21. 根据权利要求 20 所述的设备，其特征在于，所述逻辑闪速存储器件（1101）至少包括两个在物理上的分离存储器件，这两个分离存储器件一起构成逻辑上的整体存储器件。

22. 根据权利要求 21 所述的设备，其特征在于，所述至少两个物理上的分离存储器件包括固态闪速存储器和磁盘驱动器。

利用逻辑闪速存储器件进行最小单元更新的方法和设备

技术领域

本发明一般地涉及对包括几个必须以一致方式必须全部一起成功或者全部一起失败的低级操作的更新进行处理的技术。本发明尤其涉及在执行事务时使用内存地址翻译层的技术。

背景技术

在保持文件系统领域广泛使用执行事务的概念，在文件系统中，数据的完整性及遵守特定的商务规则是主要的事情。作为例子，我们来研究金融应用程序，在金融应用程序中，帐户文件列出当前具有其有效余额的所有帐户，并且分开的帐户分录文件列出所有实际帐户分录。假定对帐户分录文件进行了新的记录，说明对特定帐户支付或提取特定金额。然而，不对帐户文件进行相应更新操作。结果破坏了文件系统的数据完整性，即从帐户文件读取的特定帐户的余额与在帐户分录文件内检索记录获得的余额不同。通常，将这种冲突称为违反事务一致性。危险情况的另一个例子是，违反所谓商务规则，这意味着破坏了特定应用专用规则，例如，对本当结帐待核的帐目进行登帐。

事务概念意味着将导致文件系统内数据变更的所有这种低级操作组织在一起形成一个所有操作必须或者一起成功或者一起失败的实体。还将低级操作称为更新操作。通常，将事务划分为 3 个相继的阶段，即开始事务、更新数据以及执行事务。如果出现严重系统故障（例如：系统复位），则根据是否已经到达执行阶段，或者对数据进行所有更新操作，或者不改变数据。在执行阶段出现严重系统故障会导致系统挂起并在接着激活系统时结束挂起。如果在事务到达执行阶段之前发生严重系统故障，则在下次开机时，自动删除该事务的所有记录。

事务机制通常支持被称为重新运行事务的反向操作，这意味着删除在更新阶段所做的所有更新操作，而不是让它们在整个执行过程中保持不变。

图 1 示出可以用于执行事务的简单、传统文件系统。控制单元 101 管理日志文件 102 和数据库文件 103。控制单元 101 首先将构成事务的操作写入日志文件 102，如箭头 104 所示。在成功将所有操作写入日志文件 102 后，控制单元 101 将它们重写（即：进行相应变更）到数据库文件 103。可以将此操作看作把变更从日志文件 102 传送到数据库文件 103，如箭头 105 所示。

如果将图 1 所示的日志文件设备与数据库文件 103 内的面向非事务的直通式更新数据进行比较，可注意到前者把文件写操作量加了一倍，并且引入了在后者不出现的文件读操作。显然，如此加载文件系统肯定具有某种有害影响，特别在大规模系统，诸如在日交易量很容易达到几十万件甚或上百万件的金融应用程序方面。

发明内容

本发明目的是提供一种比图 1 所示的现有技术设备对文件系统加载较少的情况下，执行事务的方法和设备。本发明另一个目的是提供一种可靠并且经济地执行事务的方法和设备。本发明的再一个目的是提供这样一种方法和设备，以致可以容易地按照单位时间的事务量确定其规模。

通过将闪速存储器或类似功能器件用作数据存储器，并按特定顺序对负荷量数据内容和所述数据存储器的存储地址信息进行操作，可以实现本发明的各目的。

根据本发明更新第一数字数据块的方法的特征在于，按如下顺序该方法包括步骤：

- 激活对涉及第一数据块并且必须全部一起成功或者全部一起失败的所有这些更新操作通用的第一标识符；
- 执行涉及第一数据块并且必须全部一起成功或者全部一起失败的所有更新操作；
- 将更新后的第一数字数据块和使更新后的第一数据块与所述第一标识符相关的数值存储到逻辑闪速存储器件；以及
- 去激活所述第一标识符。

本发明适于一种排除基于逻辑闪速存储器件的文件系统的系统故障的方法。其特征在于，该方法包括步骤：

- 在激活了的标识符表内检验哪个标识符在系统出现故障时是激活了的；以及
- 检验物理存储位置以定位含有使各数据块与发生系统故障时处于激活了的状态的标识符相关的数值的数据块，以及将含有此数据块的各非易失性存储器地址的状态指示符设置为指示该存储单元含有无效数据的数值。

本发明还可以应用于其特征在于包括下列内容的设备：

- 逻辑闪速存储器件
- 激活装置，用于激活第一标识符，该第一标识符对所有那些涉及第一数据块并且必须全部一起成功或者全部一起失败的更新操作是公共的；
- 装置，用于执行所有那些涉及第一数据块并且必须全部一起成功或者全部一起失败的更新操作；
- 装置，用于将更新后的第一数字数据块和使更新后的第一数据块与所述第一标识符关联的数值存储到所述逻辑闪速存储器件；以及
- 装置，用于去激活所述第一标识符。

闪速存储器的第一典型特征是物理存储单元的某种随意使用性。

作为所述第一特征的一种结果的另一个典型特征是将这种存储器排列成可利用特定访问规则组单独访问的模块。根据本发明，许多模块用于存储某些负荷量数据的未更新的形式和更新了的形式，所述这些数据的内容在事务中可能会受到修改。所述模块的地址信息和标题用于宣告哪个模块在任意给定瞬时含有最有效的负荷量数据形式。

闪速存储器的概念可以扩展为包括逻辑闪速存储器件或 LFMD，即：安排以根据闪速存储器的所述典型特征运行的任意存储装置：将存储数据组织为块、向块写入的一次性、状态指示符以指出模块内的数据状态。LFMD 的一般定义还包括至少使用两个物理上分离的存储单元存储逻辑上的完整的数据块的混合解决方案。

例如，设置传统磁盘驱动器或静态 RAM（随机存取存储器）可以安排得象 LFMD 那样运行。通常，不要求磁盘驱动器全面模仿闪速存储器特征，即在其内通常不物理擦除数据块。磁盘驱动器与实际固态闪速存储器组合在一起可以提供优势框架混合解决方案，例如，将数据块标题（或部分数据块标题，诸如状态指示符）存储到闪速存储器，而将数据块剩余部分，特别是数据负荷量存储到磁盘驱动器。尽管有两个物理上分离的存储单元，但是在逻辑上，该数据块是完整的。同样，本发明还可以独立应用于所有 LFMD，无论它们在物理上是否基于闪速存储技术。

作为本发明的赋能特征，提供一种最小可更新数据单元的故障安全最小单元更新过程。这种数据单元包括具有特定状态指示符的标题字段，该状态指示符具有有限数量的容许值。在更新数据被写入其内时，改变更新数据单元的状态指示符以指出相应负荷量数据的有效性。然后，改变相应旧数据单元的状态指示符以指出旧负荷量数据失效。

临时存储特定事务标识符以指出在进行相应事务。利用同一个事

务标识符，或者利用与其无歧义相关的标识符，标记作为所述事务一部分的所有更新操作。完成所有更新操作后，通过从其临时存储器内删除该事务标识符，或者通过另外对它进行标记以指出完成该事务，执行该项事务。

通过对由每个并行的事务要更新的数据分配一个单独的临时存储单元，可以增加对单个可更新块内存储的数据的不重叠部分上操作的多项并行事务的支持。在临时存储单元内执行并行更新操作，并且只有在此后才更新实际的存储存储单元。

本发明允许在低分级水平（进入文件系统）进行事务处理，这样可以提供许多优势。代表高分级水平并剩余文件系统存储数据的应用程序不需要知道实际上如何执行事务处理过程。文件系统级的事务处理执行过程独立于高级应用程序的执行过程，这样就可以使事务处理过程非常健全并且容易适应不同应用程序。较高的分级水平可能含有其它的更复杂的操作，所述更复杂的操作依赖于要么完成其子部分的全部，要么根本不完成的原则。

在所附权利要求中对被认为是本发明特征的新颖特征进行了具体说明。然而，通过结合附图阅读特定实施例的以下说明，可以最好地理解本发明本身的结构和操作方法及其其它目的和优势。

附图说明

图 1 示出传统文件系统的更新过程；

图 2 示出闪速存储器内的公知分级概念；

图 3 示出公知的地址翻译层；

图 4 示出更新数据的公知方式；

图 5a 示出故障安全最小单元更新操作设备；

图 5b 示出故障安全最小单元更新操作方法；

图 6 示出根据本发明使用的数据块结构；

图 7 示出执行闪速存储事务的方法；

图 8 示出执行并行闪速存储事务的设备；

图 9a 至图 9h 示出执行并行闪速存储事务的方法；

图 10a 至图 10f 示出利用链接写操作执行闪速存储事务的方法；以及

图 11 示出根据本发明实施例的设备。

具体实施方式

以上在说明现有技术时对图 1 进行了说明，因此在利用本发明的优势实施例对本发明所做的以下说明中，主要集中在图 2 至图 11。

为了全面理解本发明，首先简要说明闪速存储器以及在逻辑上具有类似结构和操作过程的其它设备的一些公知特征。通常，不首先擦除存储器的一个完整的逻辑块，就不能将闪速存储数据更新到同一个物理存储位置。擦除块级之后，被擦除块内的所有数位都具有通常为 1 的共同的第一常数。然后，通过将其值变更为通常为 0 的第二常数，可以单独地写块内的每个二进数位一次。此后，变更此位的数值的唯一方法是重新擦除整个块。

图 2 示意表示被划分为各闪速存储擦除块（FMEB）的闪速存储器件 201。总共有大值正整数 M 个 FMEB，图 2 分别示出其中的 4 个 FMEB（202、203、204 和 205）。FMEB 是可以从存储器件的其它部分单独擦除的最小逻辑存储单元。FMEB 的大小依赖于存储器件的类型，并且通常在 4kB 到 64kB 之间，其中 kB 为千字节。

每个 FMEB 分别含有一个或多个逻辑数据块（LDB）。图 2 示出将第一 FMEB 202 划分为 N 个 LDB 的情况，其中分别示出 5 个 LDB（206、207、208、209 和 210）。LDB 是逻辑闪速数据块设备内的最小可更新数据单元。从理论上说，它等于硬盘驱动器内的数据扇区。将每个 LDB 进一步划分为 LDB 标题（LDBH）（如图 2 内的 211 所

示）和负荷量数据或称 LDB 数据部分 212（缩写为 LDBD）。LDBH 特别包括：LDB 标识符字段 213，其内容用于无歧义地识别各 LDB；以及状态字段 214，指出相应物理存储区内的各数据的有效性。

由于 LDB 内的位数值只能写入一次，所以只有通过将整个 LDB 写入在最后块级擦除之前未接触过的“未用”或“空白”存储单元内，才能够对 LDB 内的负荷量数据进行随意更新。通过为其 LDBH 内的状态指示符设置相应数值，可以宣告旧存储单元“脏”。

前面已经间接说明了将 LDB 存储到至少两个物理分离单元的可能性。从逻辑的观点出发，物理存储格式不重要，只要对物理变换设置适当逻辑从而使数据块表现为逻辑上的完整实体即可。这种变换及其实际实现过程对于本技术领域内的一般技术人员是众所周知的。

图 3 示出作为使物理存储单元地址与各 LDB 标识符相关的一种变换表的物理闪速存储地址翻译层逻辑（LTPTL）的原理图。在必须读取某个 LDB 时，系统参考 LTPTL 获得当前含有特定 LDB 的物理存储地址。将该过程称为物理地址映射逻辑（LPAM）。更新 LDB 意味着改变其在物理闪速存储器内的存储单元。在 LTPTL 内进行相应改变，这样其内的新物理存储地址就指向含有更新 LDB 的新存储单元。

如上所述，存储器件可以根据上述参考图 2 和图 3 所述的原则运行，而与其物理实现过程是否依赖于实际闪速存储技术，或者数据块的物理存储是否占据一个或多个单独存储单元无关。

图 4 示出在闪速存储器内的“适当位置”更新数据的已知方式。第一步骤，将整个 FMEB 401 的内容读入易失性存储器 402，易失性存储器 402 通常为随机存取存储器或 RAM。这种存储器通常不限制对位数值重复进行反向变换和正向变换，因此第二步骤更新 RAM 402 内的数据。在第三步骤，擦除原始 FMEB 单元 401，并在第四步骤将

更新数据写回其原始闪速存储单元。如果在第三步骤或第四步骤期间出现严重错误，则有可能丢失有效数据，因为它们仅出现在易失性存储器内。图 4 所示方法的附加优势是，擦一写类型更新的效率通常非常低，并且容易导致闪速存储器老化。

图 5a 和图 5b 示出 LDB 的故障安全最小单元更新过程。图中示出了一个 LDB 大小的两个物理存储单元 501 和 502。假定对 LDBH 内的状态字段定义一组容许值，以致至少可以指出状态："未使用"、"写"、"有效"、"脏"以及"断路"。其中状态"断路"意味着相应物理存储单元不能正常运行，因此根本不能使用它。"断路"状态涉及到存储器件的一般使用过程，对于本发明并不重要。此外，在图 5 中示出可以用作工作空间的某种易失性存储器 503。

更新存储在当前常驻在第一存储单元 501 内的 LDB 的负荷量部分的数据的第一步骤 551 是将负荷量部分读入易失性存储器 503。在下一步骤 552 更新易失性存储器 503 内的数据。在此方面所述方法类似于图 4 所示的传统、非故障安全方法。然而，不将更新数据写入同一个物理存储单元，而是写入闪速存储单元 502。为了保证即使在出现不希望的系统故障之后，仍能保证始终可以得到有效数据拷贝，要执行仔细设计的系列后续步骤。

在步骤 553，第二存储单元 502 的状态指示符字段 522 由"未使用"变更为"写"。在步骤 554，将 LDB 标识符从第一存储单元 501 的 LDB 标识符字段 511 拷贝到第二存储单元 502 的 LDB 标识符字段 521。在步骤 551，从第一存储单元读取 LDB 标识符和负荷量数据，或者事先从其它位置（例如：LTPTL）获得 LDB 标识符，其最大优势在于，无需在步骤 554 对第一存储单元进行实际读取操作。然而，从理论上说，最容易认为准备新存储单元 502 的部分过程是直接从旧存储单元 501 内拷贝 LDB 标识符。在步骤 555，将更新负荷量数据写入新存储单元 502 的负荷量数据字段 523。可以改变步骤 554 和步骤

555 的相互顺序，而对本发明的应用没有显著影响。

只有在成功将 LDB 标识符和更新负荷量数据均写入新存储单元 502 后，才能在步骤 556 将新存储单元 502 的相应字段 522 内的状态指示符从“写”状态变更为“有效”状态。此后，在步骤 557，这样更新物理闪速存储器地址映射逻辑，以致它使 LDB 标识符（同样分别出现在旧存储单元和新存储单元的标识符字段 511 和 521 内）与新物理存储单元 502 相关。在步骤 558，将旧存储单元 501 的相应字段 512 内的状态指示符从“有效”变更为“脏”。在之后某个瞬时，执行步骤 559，在步骤 559，擦除旧存储单元所属的 FMEB（未示出），同时将旧存储单元 501 内的相应字段 512 内的状态指示符由“脏”变更为“未使用”。

更新物理闪速存储器地址映射逻辑的步骤关系不重要。最大优势在于，在将更新负荷量数据写入新存储单元步骤之后，并在擦除旧存储单元之前，或者至少在下一次访问更新负荷量数据之前，执行此步骤。以下将对涉及物理闪速存储器地址映射逻辑的更新步骤的存储单元进行研究。

总之，由于在各瞬时，在非易失性闪速存储器内存在特性有效数据拷贝，所以图 5a 和图 5b 所示方法可以应用于事务处理过程。如下的状态表示出在图 5b 所示各步骤之后，状态指示符字段内的值。

表 1

	551	552	553	554	555	556	557	558	559
512	有效	有效	有效	有效	有效	有效	有效	脏	未用
522	未用	未用	写	写	写	有效	有效	有效	有效

由不希望的严重故障恢复的一般原则是，从其内的 LDB 状态指示符字段指出数据有效的存储单元选择数据。然而，从表 1 中可以看出，在步骤 556 结束之后，并且在步骤 558 开始之前，存在两个存储

单元均要求包括有效数据的短暂周期，但是更新新存储单元内的数据，而不更新旧存储单元内的数据。从这种情况恢复的简单原则是，随机选择其内的 LDB 指示符相同的存储单元之一并而抛开其它存储单元，即将其状态指示符由“有效”变更为“脏”。因此，如果选择旧存储单元为有效，则未更新数据保持有效并且可以认为根本没有进行企图进行的更新。如果选择新存储单元，则可以认为企图的更新已经成功。

还有一种处理“有效＋有效”情况的方法，这种情况仅略比所属随机选择复杂，然而用全确定性取代了与随机选择相关的不确定性。可以规定 LDB 包括许多位，这些位用作基于时效顺序的循环指示符。每次在将 LDB 写入闪速存储器时，根据循环方式，直截了当地说，根据方式：00→01→10→11→00 等，指示符位的值从其先前值开始变更（或者如果每次优先改变一位的数值，则：00→01→10→11→00 等）。如果在存储器内发现同一个 LDB 的两个拷贝，则通过比较其循环指示符位，可以容易地发现其中哪个是新拷贝，哪个是旧拷贝。具有特定位数的循环指示符的有效性依赖于假定将同一个 LDB 的旧拷贝标记为“脏”或者在指示符数值的循环完成之前将同一个 LDB 的旧拷贝擦除。循环指示符内的位数越多，该数值的循环越长，并且通过检验循环指示符值可以无歧义跟踪的相应“LDB 历史”越长。

在表 1 内的用下划线和黑体字强调说明可以从其内有效读取数据的存储单元。

根据上述对故障安全更新过程的说明，现在，说明完全闪速存储事务的优势实施例。我们假定在系统内将其它字段附加到各 LDB 的 LDB 标题。图 6 示出其标题包括 LDB 标识符字段 602、LDB 状态指示符字段 603 以及事务标识符字段 604 的 LDB 601 的原理图。我们进一步假定 LDB 状态指示符字段 603 的容许值对应于状态：“未使用”、“写”、“事务写”、“有效”、“脏”以及“断路”，其中“事务

写”状态上述描述状态的附加状态。此外，假定系统在理论上具有选择事务标识符供其使用，因此在给定瞬时，各事务标识符或者处于使用状态（“有效”状态）或者处于空闲（“无效”状态）状态，并且假定至少指定一个 LDB 作为有效 LDB 的存储区。我们将此特定 LDB 称为 LDB＃T。

现在，参考图 7，在步骤 701，通过激活事务标识符，即选择到目前为止无效事务标识符并将它写入 LDB＃T，开始进行事务。我们将此特定事务标识符称为＃TID。通过利用参考图 5a 和图 5b 说明的故障安全最小单元更新过程，可以实现此写入过程。在成功激活＃TID 后，就可以认为已经开始执行事务。

激活事务标识符的概念对于本发明具有一般重要性，并且，显然，将它写入 LDB＃T 不是唯一可行实际实现过程。在本发明上下文中，将激活事务标识符理解为可以将事务标识符设置为特定第一状态的所有此过程，在特定第一状态，可以无歧义恢复它，即使系统发生故障。相应地，激活特定事务标识符意味着将它设置为特定第二状态，即使在系统发生故障之后，仍可以无歧义识别特定第二状态与所述特定第一状态。可以将已经激活并且还未去激活的所有这种事务标识符指定为有效事务标识符。

闪速存储事务情况下的实际数据更新过程包括多个分别涉及单个 LDB 的更新操作。在步骤 702，将特定 LDB 标识符 LDB＃X 标识的 LDB 的内容从第一闪速存储单元读入易失性工作空间存储器，并且在步骤 703，对其负荷量数据部分内的数据进行更新。在步骤 704，新的第二闪速存储单元的状态指示符字段由“未使用”变更为“写”，并且在步骤 705，将 LDB＃X 写入第二存储单元的 LDB 标识符字段。在步骤 706，将＃TID 写入第二存储单元的事务标识符字段，在步骤 707，将更新数据写入第二存储单元的负荷量数据部分。还可以改变步骤 705、706 和 707 之间的相互顺序。在将正确 LDB 标识符（LDB

#X）、事务标识符（#TID）以及更新负荷量数据成功发送到第二存储单元后，在步骤 708，将其状态指示符字段的数值从“写”变更为“事务写”。

在步骤 709，检验完成 LDB 更新操作是否是最后一个闪速存储事务。如果步骤 709 的检验结果是否定的，则意味着在当前事务情况下，仍然存在待执行的其它 LDB 更新操作，因此，除了开始点与上述不同并且更新结果写入的第二存储单元与上述第二存储单元不同之外，返回步骤 702。如果步骤 709 的检验结果是肯定的，则意味着与当前事务相关的数据更新过程已经结束，因此该事务进入其执行阶段。

图 7 所示的闪速存储事务的执行阶段包括步骤 701，在步骤 701，从 LDB#T 内删除#TID。故障安全最小单元更新过程还用于进行此删除过程。此时，应该注意，被更新的与事务相关的所有旧存储单元被其状态指示符字段宣告有效。因此，以包括步骤 711、712、713 和 714 的循环方式继续进行事务。在步骤 711，更新物理闪速存储器地址映射逻辑，从而使 LDB 标识符 LDB#X 与新物理存储单元相关。与其它方法步骤有关的地址映射更新步骤的实际存储单元不象上述根据故障安全最小单元更新过程说明的那样重要。在步骤 712，容纳被称为 LDB#X 的旧、未更新特定数据块的特定第一存储单元的状态指示符由“有效”变更为“脏”。

在步骤 713，容纳被称为 LDB#X 的新、更新数据块的相应第二存储单元的状态指示符由“事务写”变更为“有效”。如果未使用同一个事务标识符，则不必明确执行步骤 713，以下将详细说明延迟步骤 713。在步骤 714，检验与执行事务相关的所有 LDB 是否在步骤 711、712 和 713 进行改变，如果检验结果是否定的，则返回步骤 711 以处理另一个第一—第二存储单元对，如果步骤 714 的检验结果是肯定的，则意味着检验处理过程已经完成。

如下的状态表示出在图 7 所示各步骤之后位于特定旧存储单元和新存储单元内的状态指示符的数值。为了简洁起见使用简略符号，即简略符号 U 表示“未使用”、W 表示“写”、T 表示“建议写”、V 表示“有效”、D 表示“脏”。

表 2

	701	702	703	704	705	706	707	708	709	710	711	712	713	714
旧	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	**V**	V	V	**D**	D	D
新	U	U	U	W	W	W	W	T	T	**T**	**T**	**T**	**V**	**V**

表 2 内的黑体字和下划线强调说明在给定瞬时其内的有效数据可用的存储单元。我们现在分析处理事务过程的一般规则方面的数据有效性。

在上述说明中，我们注意到事务的所有更新操作必须全部一起成功或者全部一起失败。这两者之间取舍分界线是事务的执行阶段。在图 7 中，步骤 710 为执行步骤。

如果在事务到达步骤 710 之前，即在从有效事务标识符的指定存储器删除事务标识符＃TID 之前，出现严重系统故障，则基本上不能确定是否正确完成所有 LDB 更新。因此，唯一合理方法是使事务无效，即使所有更新操作无效。换句话说，在出现严重错误之后，接着启动系统时，所有物理存储单元均被标记为“脏”，这样就可用满足下列两个判据之一：

- 其状态指示符字段的数值表示“写”状态
- 其状态指示符字段的数值表示“事务写”状态，并且其事务标识符字段内的事务标识符还出现在有效事务标识符的指定存储器内。

在某些情况下，待宣告为“脏”的物理存储单元还满足其 LDB 标识符字段的数值还出现在另一个（旧）物理存储单元的相应字段内

的判据。然而，这种特征不是因为几个原因待宣告为“脏”的物理存储单元的可靠霍尔标记。其中最重要的是在将其擦除之前，旧物理存储单元保持该状态多长时间的典型不确定性。背景清除例程连续监视闪速存储器的状态并试图通过移动数据块、擦除 FMEB 产生空闲存储空间。此外，在某些情况下，会出现在没有 LDB 的“旧”拷贝的操作的最初阶段发生系统故障，因为在待执行的第一事务导致“新”拷贝被写入闪速存储器之前，闪速存储器被清除。

如果在完成步骤 710 之后，即在从有效事务标识符的指定存储器删除事务标识符＃TID 之后，出现严重系统故障，则可以确定正确进行了 LDB 更新。因此，可以安全宣告事务被成功执行，并且可以在接着启动系统时，从“新”存储单元读取更新 LDB。在这种情况下，所有物理存储单元被标记为“脏”，这样就满足了以下 3 个判据：

- 其 LDB 标识符字段的数值还出现在另一个（新）物理存储单元的相应字段内；
- 其状态指示符字段的数值表示状态“有效”；
- 在具有相同 LDB 标识符的其它（新）物理存储单元的事务标识符字段建立的事务标识符不出现在有效事务标识符的指定存储器内。

此外，将所有物理存储单元标记为有效，这样就满足如下两个判据：

- 其状态指示符字段的数值表示状态“事务写”
- 在其事务标识符字段建立的事务标识符不出现在有效事务标识符的指定存储器内。

可以延迟执行上述说明的将“事务写”块变更为“有效”的步骤，直到存储在其事务标识符字段内的同一个 TID 下一次被激活，或者直到重新更新这些特定块。

在分析如果在系统执行步骤 710 时发生严重系统故障会出现什么问题时，我们参考结合图 5a 和图 5b 说明的故障安全最小单元更新操作过程。我们给出了从不意外故障恢复的原则，因此在大多数情况下，不可能出现不确定性。只有在旧物理存储单元和新物理存储单元均有效时，如果允许随机选择旧物理存储单元或新物理存储单元，会出现发生严重系统故障时闪速存储器所处的状态不能明确确定在下一次启动系统时是选择重新运行中断事务，还是选择执行中断事务。如果更新 LDB＃T，我们自然要规定始终选择“有效＋有效”情况，例如在其内存储少量（大量）事务标识符的存储单元情况，因此或多（或少）可以将步骤 710 解释为已经完成。在此应用程序之前，最具优势的方法是可以将循环指示符用作更新 LDB＃T 过程的一部分，因此可以取消上述随机性。

接着，将对本发明应用于并行事务的情况进行说明。事务处理的一个基本原则是，从来不允许进行两个并行事务以准确并行更新同一个逻辑数据区。然而，可以允许两个或多个并行事务来并行更新特定 LDB 的非重叠部分，只要是顺序执行事务执行阶段即可，也就是只有在不同时执行另一个并行事务的执行阶段时，才可以开始特定并行事务之一的执行阶段。

为了简洁起见，我们仅说明并行事务对单个 LDB 的影响。然而，什么都不防碍把并行事务应用于更新非常大量 LDB。在以下说明中易于推广到涵盖这种情况。

图 8 以及图 9a 至图 9j 示出利用闪速存储器实现并行事务的设备和方法。在图 8 中，示出旧物理存储单元 800、多个新物理存储单元以及多个易失性存储器（RAM）缓冲区。为了处理并行事务，优势在于，有多少对单个 LDB 产生影响的并行事务就有多少新物理存储单元和多少易失性存储器缓冲区。在此，假定并行事务数为 3，则有 3 个新物理存储单元 810、820 和 830，并且有 3 个易失性存储器缓冲区

841、842 和 843。3 个并行事务与特定事务标识符 TID＃A、TID＃B 和 TID＃C 相关。通常，利用正整数 N 表示并行事务数。LDB 结构与上述相同，即在 LDB 标题，存在 LDB 标识符字段、LDB 状态指示符字段和事务标识符字段。假定将作为并行更新目标的 LDB 表示为 LDB ＃X。在开始执行并行事务时，其 TID 字段的内容不重要，因此可以用 TID＃？来表示它。

如果在非常不同的时间激活并行事务，则可以取消对有多少并行事务就有多少新物理存储单元和有多少易失性存储器缓冲区的要求，因此其中某些可以使它较早完成并且可以释放对其分配的资源。同样，可以对易失性存储器缓冲区的大小进行优化，以致各易失性存储器缓冲区的大小足以包含使用此易失性存储器缓冲区的特定事务允许更新的这些数据位。易失性存储器缓冲区尤其不需要包括待更新的 LDB 的标题字段的存储空间；通过用虚线示出相应部分，在图中说明此情况。实际上，完全可以认为，通过在更新过程中在用新数据替换旧数据的各步骤使用新闪速存储单元，就可以实现此事务，而无需易失性存储器缓冲区。然而，与以相对较大块形式将数据从逻辑闪速存储器件读入易失性工作空间存储器，在易失性工作空间存储器内执行所有更新过程并将更新块重新写入闪速存储器的解决方案相比，这种“从闪速到闪速”更新过程的效率通常非常低。

将允许进行特定事务 TID＃i 以进行更新的 LDB＃X 的未更新拷贝内的数据位标记为 i1，其中 i＝A、B、C。在此，假定几乎同步进行并行事务，但是并不限定要求：在开始一个事务之前，在相当长时间内进行另一个事务。在我们的例子中，通过将事务标识符 TID＃A、TID＃B 和 TID＃C 写入有效事务标识符 850 的存储器内，开始进行并行事务处理过程。通过将 LDB＃X 从旧物理存储单元 800 读入所有分配易失性存储器缓冲区 841、842 和 843，开始进行数据更新过程。图 9a 示出读操作之后的情况，在此步骤，不对数据进行更新，因为各事务同步开始。

仅允许各并行事务分别更新在易失性存储器缓冲区之一内与其相关的数据选择，因此我们可以假定 TID＃A 更新缓冲区 841 内的 A 位、TID＃B 更新缓冲区 842 内的 B 位，TID＃C 更新缓冲区 843 内的 C 位。此时，不改变新物理存储单元 810、820 和 830 内的内容。

图 9b 示出最快并行事务之一（即事务 TID＃A）完成必须在相应易失性存储器缓冲区 841 内执行的所有更新时的情况，即利用更新 A2 位代替所有 A1 位情况。然后，根据上述结合单个事务说明的方法步骤，将该易失性存储器缓冲区的内容写入相应新闪速存储单元 810。图 9b 示出如何通过将状态标识符从“未使用”变更为“写”并且通过将 LDB 标识符 LDB＃X 和事务标识符 TID＃A 写入正确标题字段开始进行写。

图 9c 示出如何将第一易失性存储器缓冲区 841 的内容写入相应新闪速存储单元 810。此外，图 9c 还示出并行事务处理过程中的重要步骤：还将第一事务执行的更新 A1→A2 拷贝到其它易失性存储器缓冲区 842 和 843。把完成的新闪速存储单元 810 的状态标识符从“写”变更为“事务写”。

图 9d 示出并行事务处理中的另外两个改进步骤。通过从有效事务标识符 850 的存储器内删除其标识符、将旧物理存储单元 800 的状态指示符从“有效”变更为“脏”并将完成新物理存储单元的状态指示符从“事务写”变更为“有效”，完成事务 TID＃A。同时，发生了第二最快速事务 TID＃B 到达在易失性存储器缓冲区 842 内进行所有更新的阶段，因此通过将状态标识符从“未使用”变更为“写”并且通过将 LDB 标识符 LDB＃X 和事务标识符 TID＃B 写入正确标题字段，开始写入相应闪速存储单元 820。

显然，除了以下不同之外，图 9e 和图 9f 所示的步骤分别与图 9c

和图 9d 所示的步骤相同：

- 在图 9e 中，完成写入第二闪速存储单元 820，并且就是 B1→B2 的更新被拷贝到剩余易失性存储器缓冲区（最大优势在于，在此步骤不分配缓冲区 841，通过将它放在括号内表示的）
- 在图 9f 中，执行事务 TID＃B（使先前有效物理存储单元 810 “脏”并使最新物理存储单元 820 有效）并且就是最后事务 TID＃C 结束易失性存储器缓冲区 843 内的更新，从而开始写入第三闪速存储单元 830。

显然，除了完成写入第三闪速存储单元 830，并且图 9g 所示的步骤最大优势在于，在此步骤不分配所有其它易失性存储器缓冲区，所以不将 C1→C2 更新拷贝到任何其它易失性存储器缓冲区，除此之外，图 9g 所示的步骤与图 9e 所示步骤相同。 在图 9h 中，通过从有效事务标识符 850 的存储器内删除标识符 TID＃C、将相应状态标识符从“事务写”变更为“有效”以及使先前有效物理存储单元 820 变成“脏”，执行完成事务 TID＃C。

由于利用上述说明的最小单元更新过程，所以更新 LTPTL 表以指向特定 LDB 的最新有效拷贝的各步骤的存储单元并不重要，但是最大优势在于，它是在将所述最新有效拷贝写入闪速存储器之后并且在擦除旧、无效拷贝之后发生的。

通过分别分析并行事务的各执行阶段，可以容易地发现上述过程与一般事务处理原则之间的一致性。在所有这些事务中，留待在下一次启动系统时，执行在最后出现系统故障之前从有效事务标识符 850 的存储器内删除的标识符，因为具有被这些事务更新的数据的闪速存储单元处于“有效”状态。重新运行仍然具有被作为有效标识符存储的所有这些事务。

在对本发明实施例的说明中，到目前为止假定：在完成从逻辑闪

速存储器件读取旧 LDB 之后，并且在将新、完成更新 LDB 写入闪速存储器之前，完成数据的所有更新过程。为了实现完整性（接着将参考图 10a 至图 10f 说明），采用对闪速存储器进行所谓链接写操作的本发明实施例。为了使本说明简洁，我们仅研究单个、非并行事务，但是同样的原理完全适用于并行事务。

图 10a 示出在将数据从旧物理存储单元 1000 读入易失性工作空间存储器（未示出）进行第一次更新的开始阶段时的旧物理存储单元 1000 和两个新物理存储单元 1010 和 1020。在图 10b 所示的步骤，准备利用被称为 TID＃Y 的事务进行第一次更新，因此开始将部分更新 LDB 写入第一新物理存储单元 1010。在此，还假定首先写入标题字段，并使状态指示符获得数值“写”。在图 10c 所示的步骤，结束写入部分更新拷贝，因此其状态指示符被变更为“事务写”。与此同时，第二次更新接近结束，因此开始写入第二新物理存储单元 1020。

图 10d 示出也已经结束写入完成更新拷贝，因此也将其状态指示符变更为“事务写”的步骤。图 10e 示出旧、部分更新状态指示符变如何更为“脏”，同时仍保持执行事务。下一个步骤是通过从有效事务标识符（未示出）的存储器内删除 TID＃Y，执行事务的步骤。在只有一个更新数据保持“事务写”状态阶段进行此执行步骤，因为如果在执行之后，立即发生系统故障，则在下一次启动时，处于此状态的多个更新数据会产生歧义。图 10f 所示的最后步骤包括这样地更新状态指示符：使第二新物理存储单元 1020 现在具有有效数据，而其它物理存储单元为“脏”。

在链接的写操作情况下，变更物理存储单元的状态指示符的顺序有几种选择。例如，即使是在开始写入第二新物理存储单元 1020 之前，仍可以将第一新物理存储单元 1010 内的部分更新拷贝的状态变更为“脏”。换句话说，图 10c 所示的状态指示符从上到下是 V、D 和 W，并且在进一步步骤中，第一新物理存储单元的状态将保持“脏”。

另一种可能性是，在开始写入第二新物理存储单元 1020 之后，而在将第二新物理存储单元 1020 的状态从“写”变更为“事务写”之前，将第一新物理存储单元 1010 内的部分更新拷贝的状态变更为“脏”。换句话说，第一新物理存储单元是图 10d、图 10e 和图 10f 所示的“脏”。

以上结合并行事务说明的一般方法可以应用于实现基于 RAM 的闪速存储写高速缓存，该基于 RAM 的闪速存储写高速缓存有助于精简链接写操作方法。这意味着，在事务 TID＃X 第一次更新 LDB＃X 并且仅将部分更新 LDB＃X 写入高速缓存时，始终将特定 LDBD＃的闪速拷贝写入 RAM。即使当前不存在更新同一个 LDB 的其它事务，仍将数据从闪速存储器读入 RAM 缓冲器。如果同时存在更新该 LDB 的其它事务，则直接进行处理并行事务的处理过程。从数据的完整性观点出发，在其执行阶段之前在执行事务期间进行的写高速缓存是安全的。可以显著改善文件系统的性能，特别是在事务中仅更新 LDB 的一位或一个字节情况下。

图 11 示出根据本发明优选实施例的设备 1100 的原理图。闪速存储器 1101 通过读写单元 1103 连接到工作空间 RAM 1102，读写单元 1103 在闪速存储器 1101 与工作空间 RAM 1102 之间实现实际数据传送，并利用地址映射块 1104 将逻辑数据块标识符映射到闪速存储器内的物理存储单元。通过执行从程序存储器 1106 读取和/或从输入/输出接口 1107 获得的程序，程序执行块 1105 控制操作过程。背景清除单元 1108 以公知方式保持闪速存储器 1101 的状态。

上述说明的本发明典型实施例不应限制以其它方式应用本发明，显然，根据上述说明，在所附权利要求所述的本发明范围内，本技术领域内的熟练技术人员可以实现各种变换实施例。例如，即使我们指出这样将事务标识符与更新 LDB 一起存储到闪速存储器，仍可以通过使更新 LDB 与特定事务标识符无歧义相关的某些其它数值与更新 LDB 一起存储，实现同样效果。

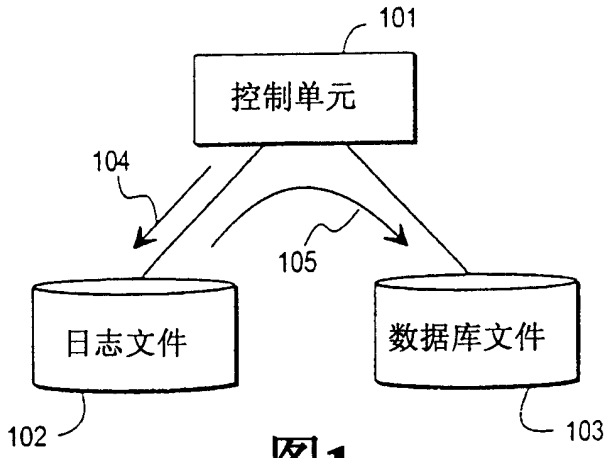


图1
现有技术

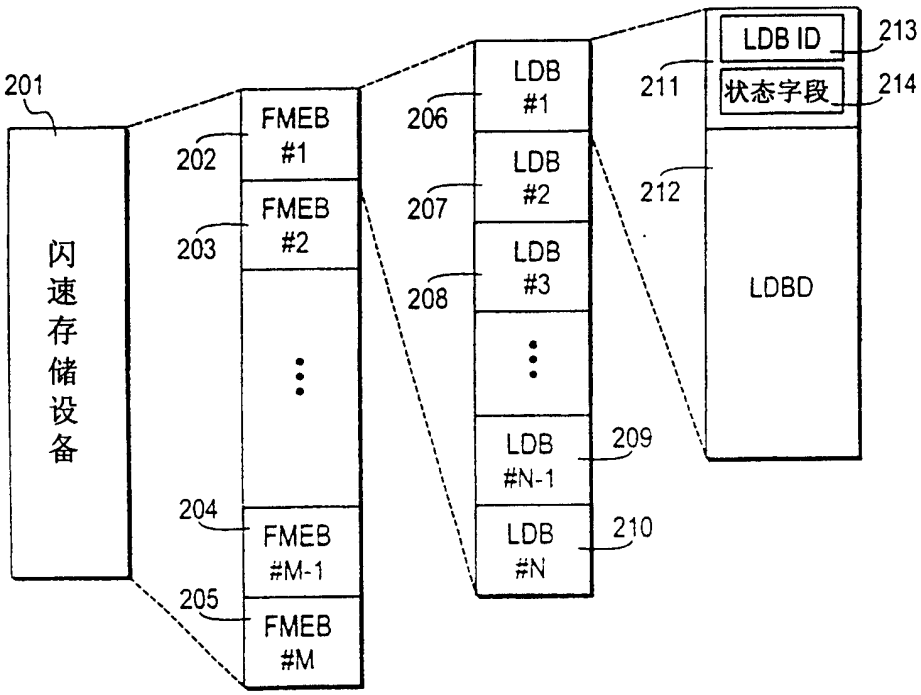


图2
现有技术

LDB ID	LDB ID	LDB ID	LDB ID	...
地址	地址	地址	地址	

图3
现有技术

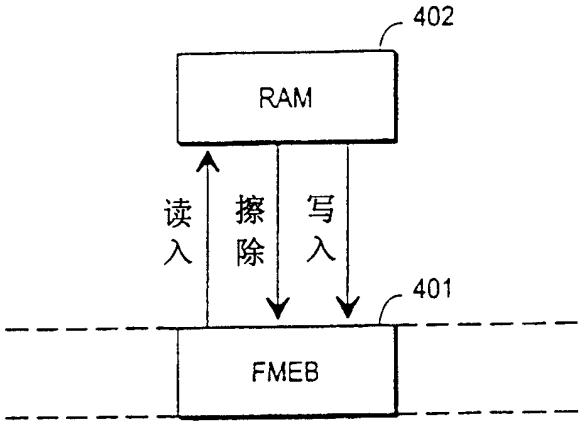


图4
现有技术

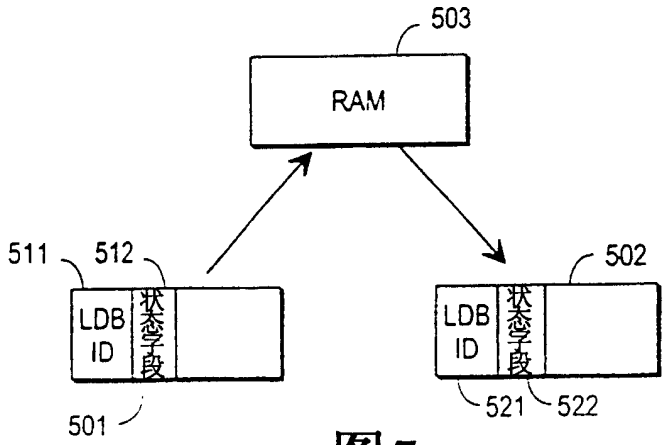


图5a

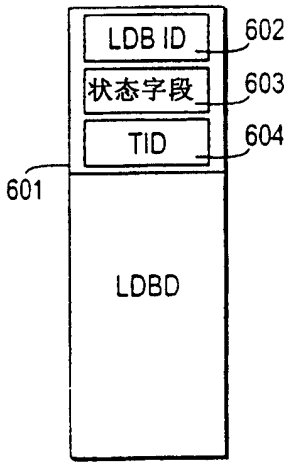


图6

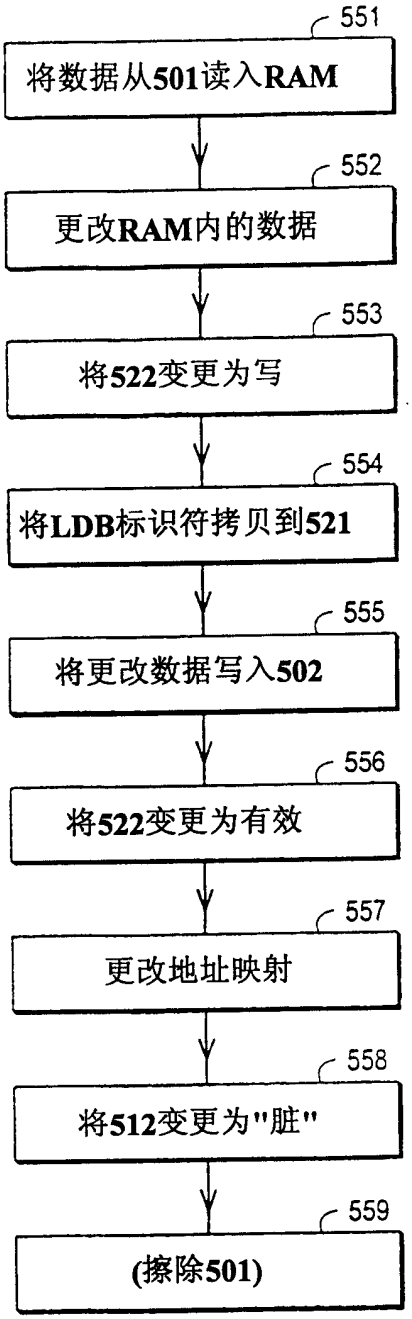

551
将数据从501读入RAM
552
更改RAM内的数据
553
将522变更为写
554
将LDB标识符拷贝到521
555
将更改数据写入502
556
将522变更为有效
557
更改地址映射
558
将512变更为"脏"
559
(擦除501)

图5b

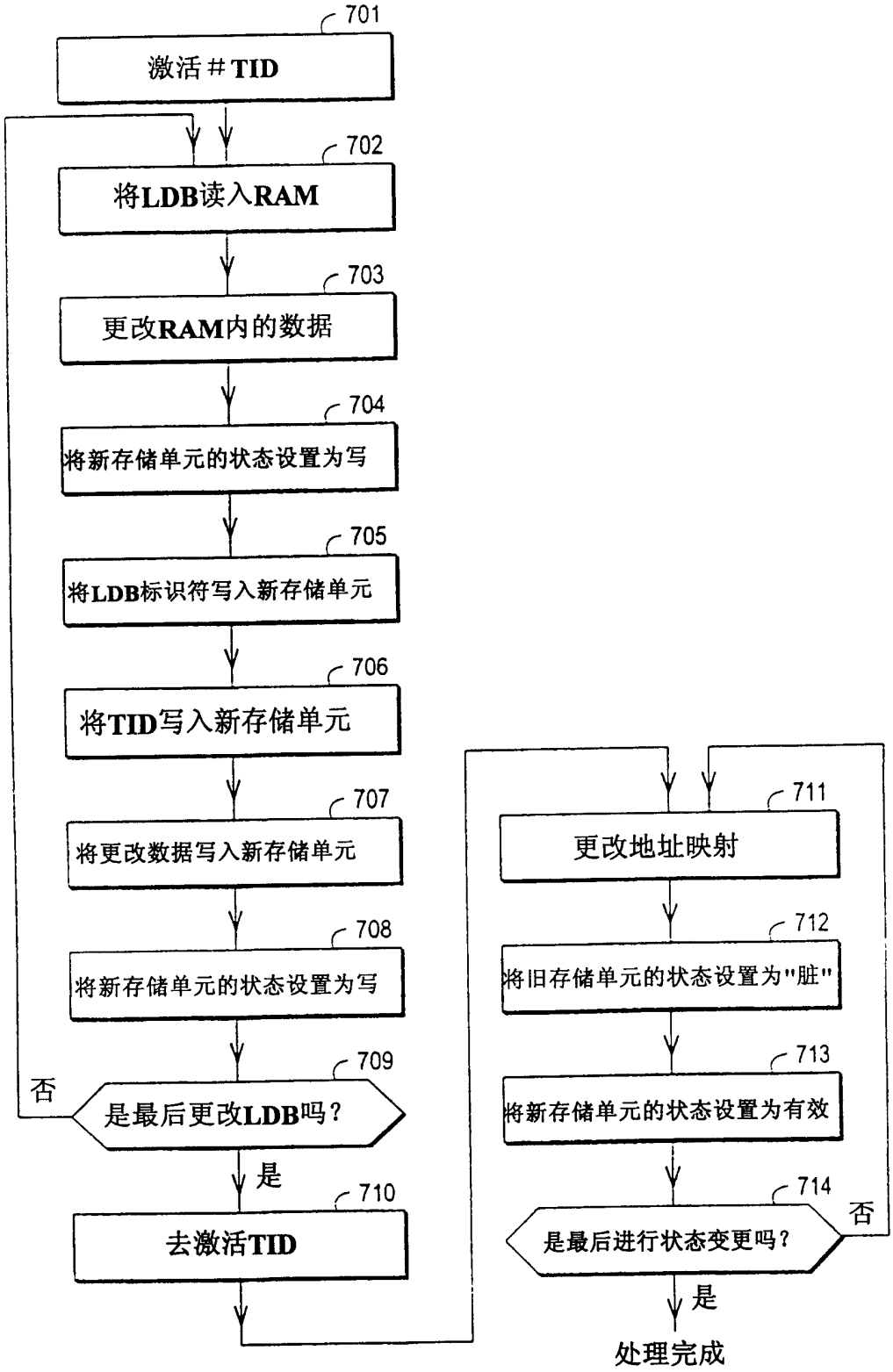

701
激活#TID
702
将LDB读入RAM
703
更改RAM内的数据
704
将新存储单元的状态设置为写
705
将LDB标识符写入新存储单元
706
将TID写入新存储单元
707
将更改数据写入新存储单元
708
将新存储单元的状态设置为写
709
否
是最后更改LDB吗？
是
710
去激活TID
711
更改地址映射
712
将旧存储单元的状态设置为"脏"
713
将新存储单元的状态设置为有效
714
否
是最后进行状态变更吗？
是
处理完成

图7

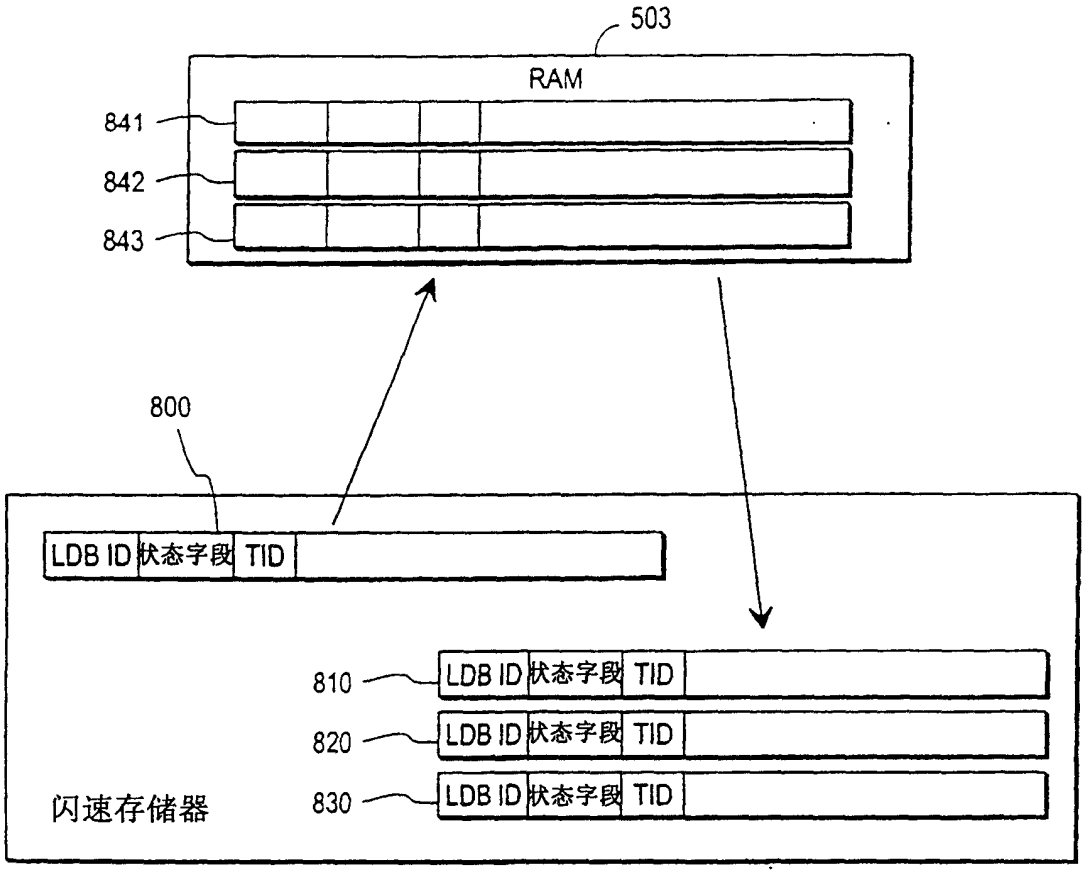

503
RAM
841
842
843
800
LDB ID
状态字段
TID
810
820
830
闪速存储器

图8

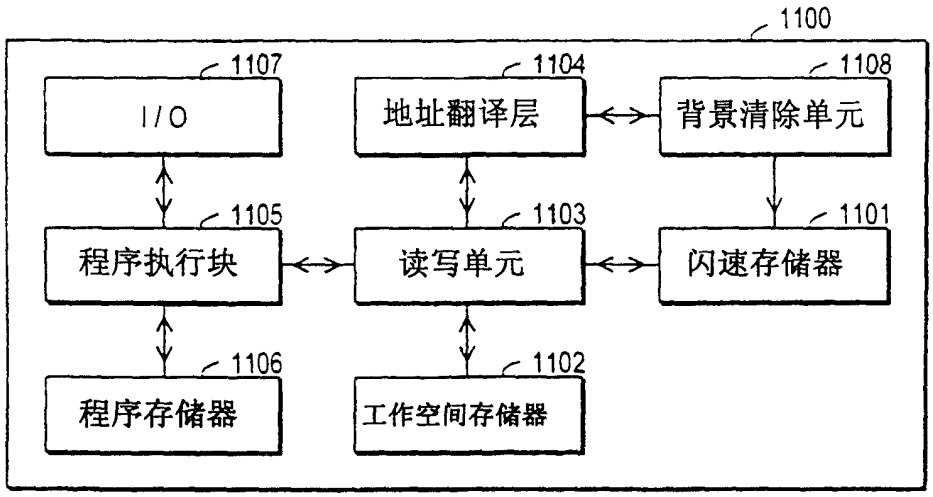

1100
1107
I/O
1104
地址翻译层
1108
背景清除单元
1105
程序执行块
1103
读写单元
1101
闪速存储器
1106
程序存储器
1102
工作空间存储器

图11

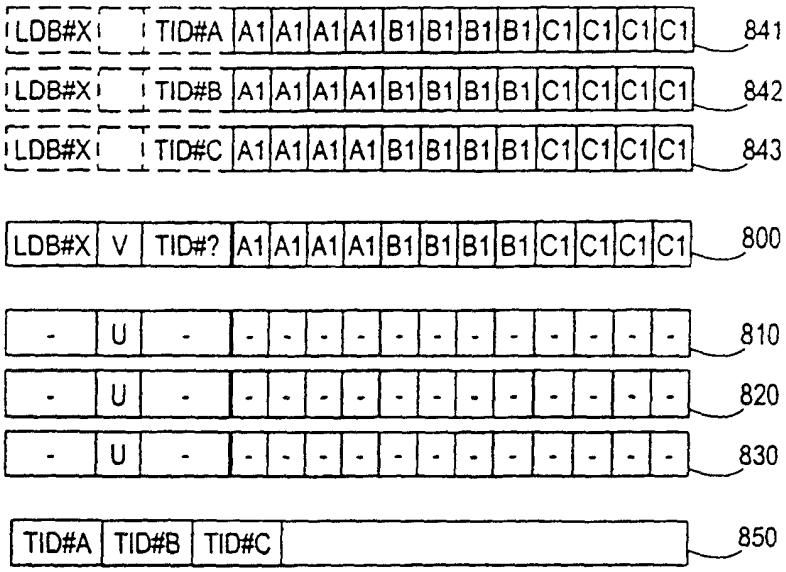

LDB#X TID#A A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 841
LDB#X TID#B A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 842
LDB#X TID#C A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 843
LDB#X V TID#? A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 800
- U - - - - - - - - - - - - 810
- U - - - - - - - - - - - - 820
- U - - - - - - - - - - - - 830
TID#A TID#B TID#C 850

图9a

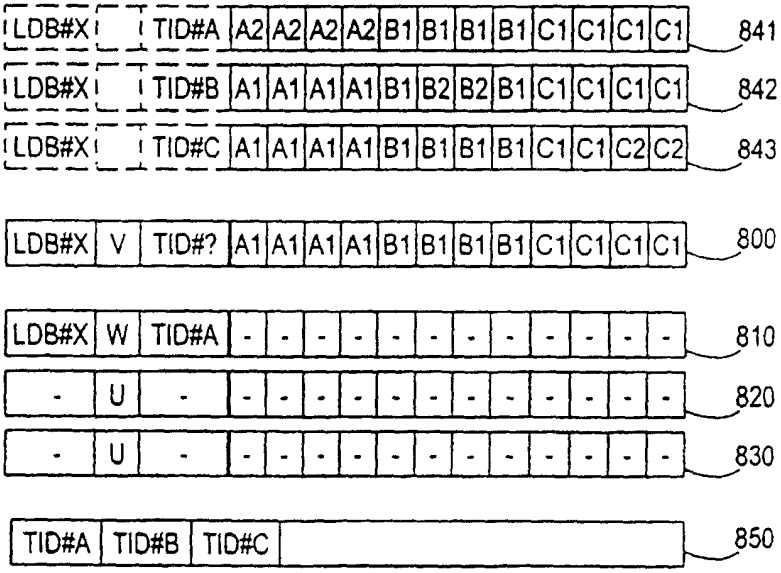

LDB#X TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 841
LDB#X TID#B A1 A1 A1 A1 B1 B2 B2 B1 C1 C1 C1 C1 842
LDB#X TID#C A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C2 C2 843
LDB#X V TID#? A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 800
LDB#X W TID#A - - - - - - - - - - - - 810
- U - - - - - - - - - - - - 820
- U - - - - - - - - - - - - 830
TID#A TID#B TID#C 850

图9b

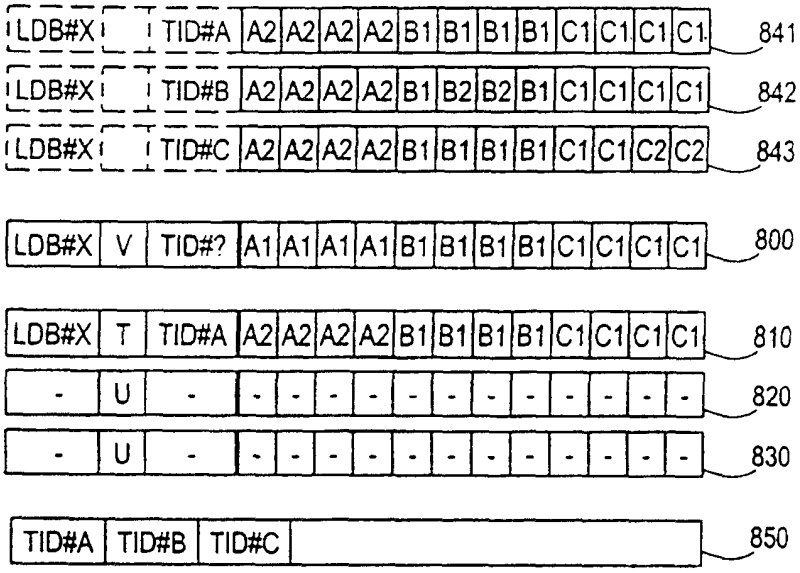

LDB#X TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 841
LDB#X TID#B A2 A2 A2 A2 B1 B2 B2 B1 C1 C1 C1 C1 842
LDB#X TID#C A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C2 C2 843
LDB#X V TID#? A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 800
LDB#X T TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 810
- U - - - - - - - - - - - - 820
- U - - - - - - - - - - - - 830
TID#A TID#B TID#C 850

图9c

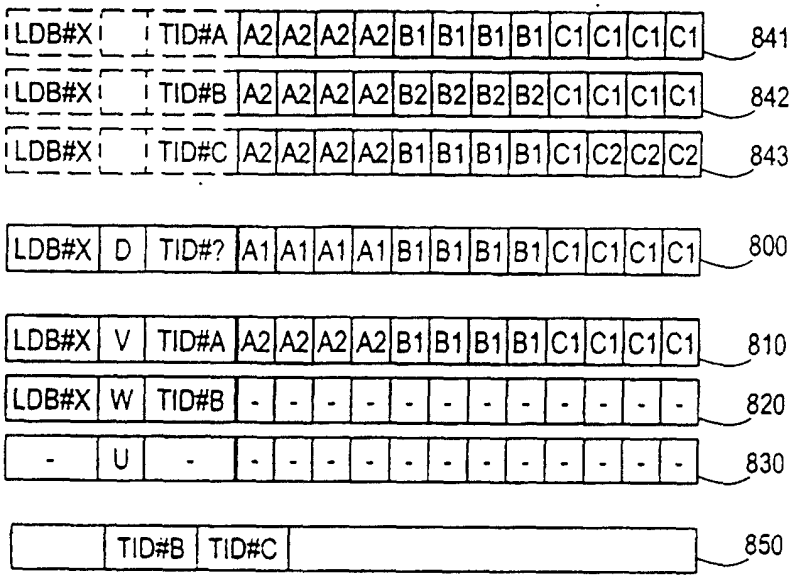

LDB#X TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 841
LDB#X TID#B A2 A2 A2 A2 B2 B2 B2 B2 C1 C1 C1 C1 842
LDB#X TID#C A2 A2 A2 A2 B1 B1 B1 B1 C1 C2 C2 C2 843
LDB#X D TID#? A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 800
LDB#X V TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 810
LDB#X W TID#B - - - - - - - - - - - - 820
- U - - - - - - - - - - - - 830
TID#B TID#C 850

图9d

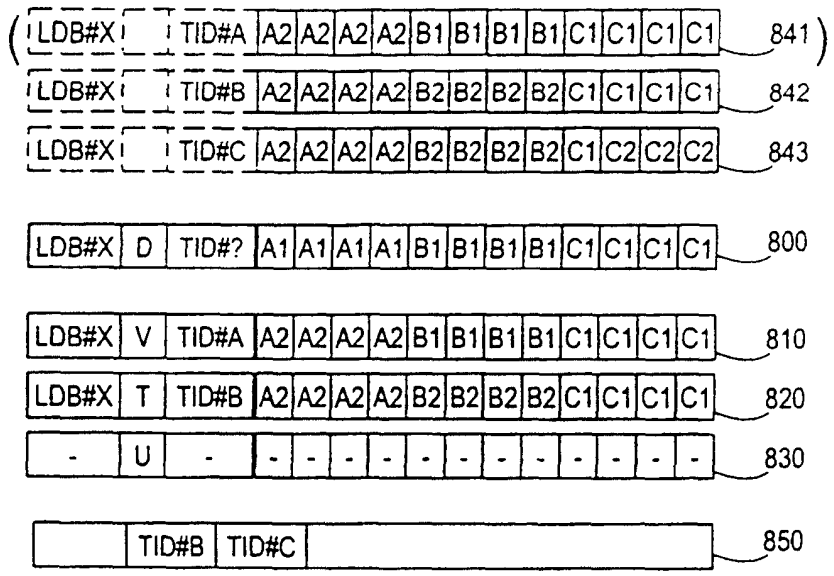

LDB#X TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 841
LDB#X TID#B A2 A2 A2 A2 B2 B2 B2 B2 C1 C1 C1 C1 842
LDB#X TID#C A2 A2 A2 A2 B2 B2 B2 B2 C1 C2 C2 C2 843
LDB#X D TID#? A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 800
LDB#X V TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 810
LDB#X T TID#B A2 A2 A2 A2 B2 B2 B2 B2 C1 C1 C1 C1 820
- U - - - - - - - - - - - - 830
TID#B TID#C 850

图9e

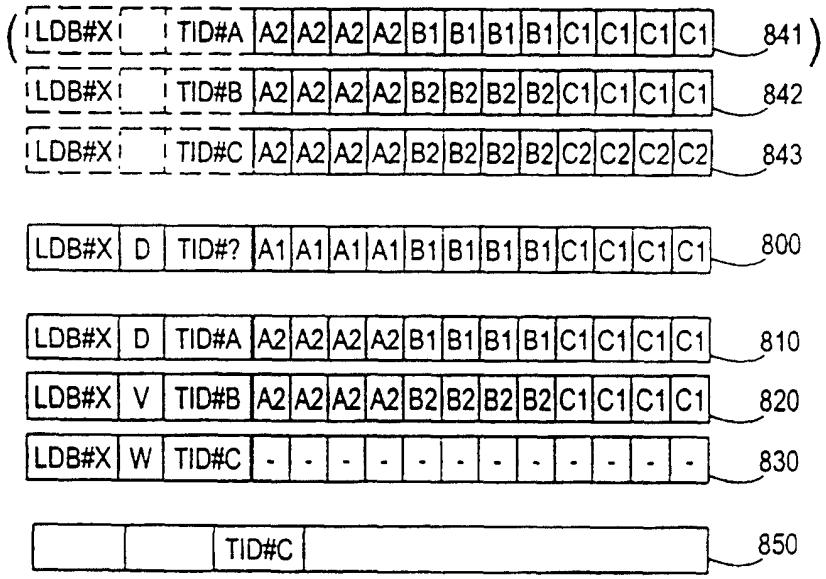

LDB#X TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 841
LDB#X TID#B A2 A2 A2 A2 B2 B2 B2 B2 C1 C1 C1 C1 842
LDB#X TID#C A2 A2 A2 A2 B2 B2 B2 B2 C2 C2 C2 C2 843
LDB#X D TID#? A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 800
LDB#X D TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 810
LDB#X V TID#B A2 A2 A2 A2 B2 B2 B2 B2 C1 C1 C1 C1 820
LDB#X W TID#C - - - - - - - - - - - - 830
TID#C 850

图9f

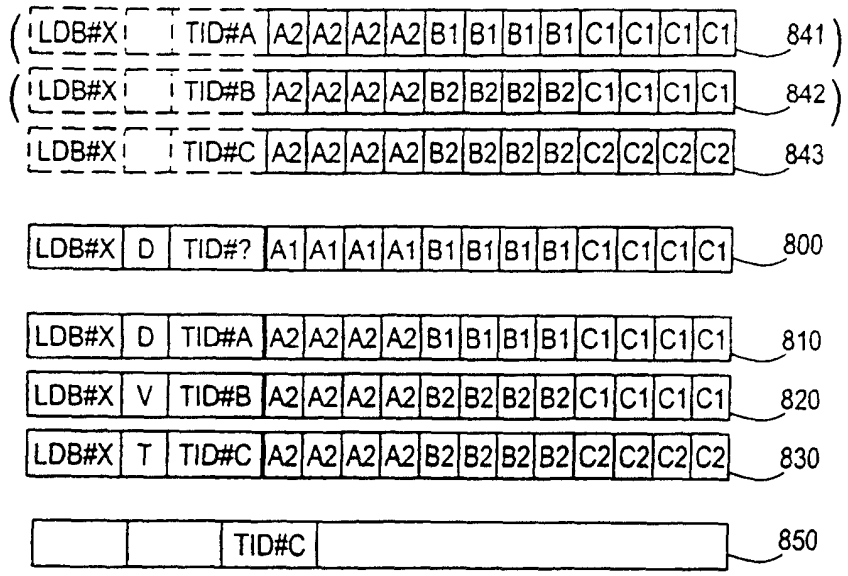

LDB#X TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 841
LDB#X TID#B A2 A2 A2 A2 B2 B2 B2 B2 C1 C1 C1 C1 842
LDB#X TID#C A2 A2 A2 A2 B2 B2 B2 B2 C2 C2 C2 C2 843
LDB#X D TID#? A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 800
LDB#X D TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 810
LDB#X V TID#B A2 A2 A2 A2 B2 B2 B2 B2 C1 C1 C1 C1 820
LDB#X T TID#C A2 A2 A2 A2 B2 B2 B2 B2 C2 C2 C2 C2 830
TID#C 850

图9g

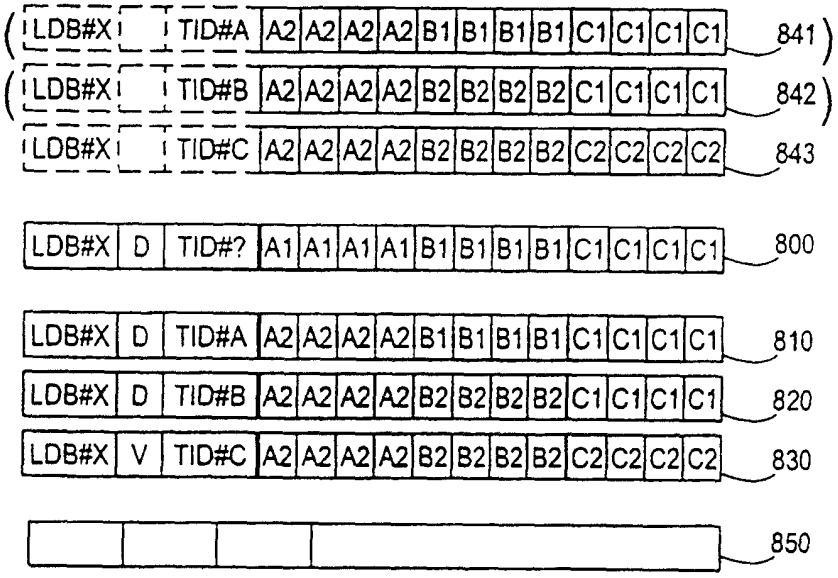

LDB#X TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 841
LDB#X TID#B A2 A2 A2 A2 B2 B2 B2 B2 C1 C1 C1 C1 842
LDB#X TID#C A2 A2 A2 A2 B2 B2 B2 B2 C2 C2 C2 C2 843
LDB#X D TID#? A1 A1 A1 A1 B1 B1 B1 B1 C1 C1 C1 C1 800
LDB#X D TID#A A2 A2 A2 A2 B1 B1 B1 B1 C1 C1 C1 C1 810
LDB#X D TID#B A2 A2 A2 A2 B2 B2 B2 B2 C1 C1 C1 C1 820
LDB#X V TID#C A2 A2 A2 A2 B2 B2 B2 B2 C2 C2 C2 C2 830
850

图9h

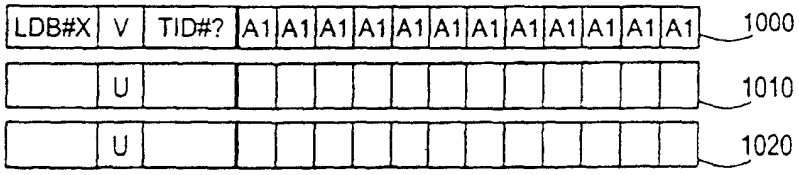


图10a

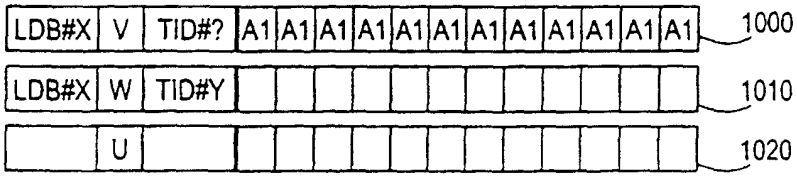


图10b

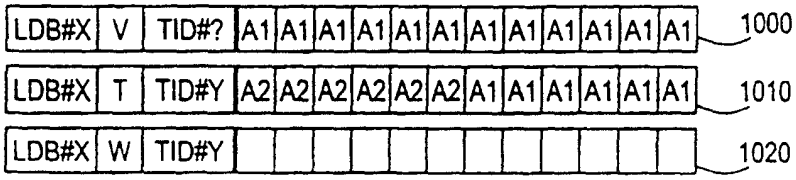


图10c

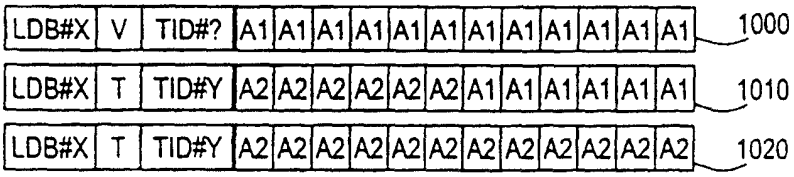


图10d

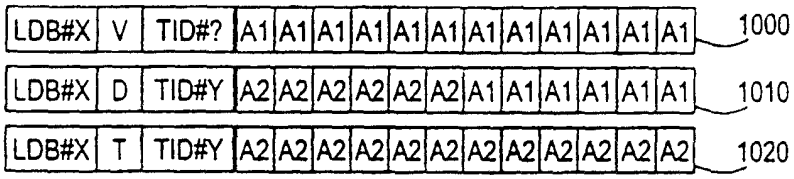


图10e

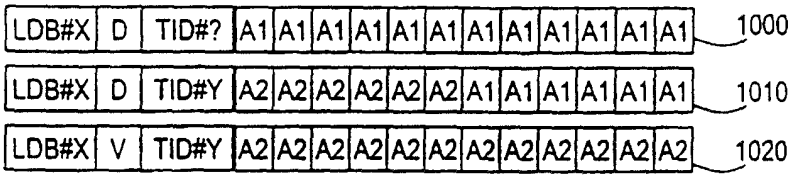


图10f