

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G07F 7/08 (2006.01)

G06K 19/067 (2006.01)



[12] 发明专利申请公开说明书

[21] 申请号 200510127054.5

[43] 公开日 2006年5月17日

[11] 公开号 CN 1773559A

[22] 申请日 1998.7.9

[21] 申请号 200510127054.5

分案原申请号 98115463.8

[30] 优先权

[32] 1997.7.9 [33] JP [31] 183534/1997

[71] 申请人 索尼株式会社

地址 日本东京

[72] 发明人 日下部进 高田昌幸

[74] 专利代理机构 中国国际贸易促进委员会专利商
标事务所
代理人 董 莘

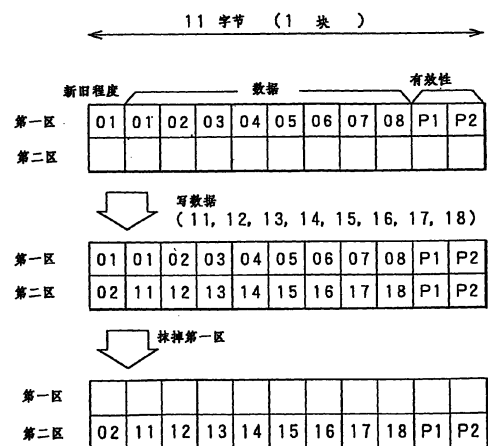
权利要求书5页 说明书37页 附图18页

[54] 发明名称

数据处理方法和设备

[57] 摘要

本发明公开了一种利用将信息存储在块单元中的存储器的数据处理方法，存储器包括用于将数据存储在多个块单元中的数据区以及用于存储多个块号的第一和第二区，多个块号被指配给数据区中的各块，第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息，包括步骤：将更新后的数据写入数据区的一个或多个块中，块对应于第一和第二区的其中一个区中的第二块号，基于新旧程度信息选择第一和第二区的其中一个区，在第一和第二区中的另一个区的第一块号中，存储在写入步骤中已经写入更新数据的块的块号，在第一和第二区中的另一个区的第二块号中，存储数据被原始写入的块的块号；在第一和第二区的其中一个区中擦除第一和第二块号以及新旧程度信息。



1、一种利用将信息存储在块单元中的存储器的数据处理方法，其中所述存储器包括用于将数据存储在多个块单元中的数据区以及用于存储多个块号的第一和第二区，所述多个块号被指配给所述数据区中的各块，

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息，其中所述第一块号用于指示被原始写入数据的块，所述第二块号用于指示数据被更新的块，

所述方法包括以下步骤：

将更新后的数据写入所述数据区的一个或多个块中，所述块对应于所述第一和第二区的其中一个区中的第二块号，并且基于所述新旧程度信息选择所述第一和第二区的所述其中一个区，

在所述第一和第二区中的另一个区的第一块号中，存储在所述写入步骤中已经写入更新数据的块的块号，并且在所述第一和第二区中的所述另一个区的第二块号中，存储数据被原始写入的块的块号；以及

在所述第一和第二区的所述其中一个区中擦除所述第一和第二块号以及所述新旧程度信息。

2. 根据权利要求1所述的数据处理方法，其中当所述新旧程度信息表明所述第一和第二区二者都有效时，更新根据所述新旧程度信息被认为最新的一个区中的数据，擦除被认为不是最新的另一个区中的数据，并且将数据写入所述数据区的一块，所述块对应于存储在其中一个所述区中的所述块号。

3. 根据权利要求1所述的数据处理方法，其中被写入数据的块的块号被存储在所述另一个区中，更新所述另一个区中的新旧信息和有效性信息，并且擦除在所述其中一个区中的数据。

4. 根据权利要求1所述的数据处理方法, 其中当基于所述有效性信息选择所述第一和第二区的任何一个, 并且在所述另一个区中的数据还未被擦除时, 擦除在所述另一个区中的数据, 并将数据写入与所述其中一个区中存储的所述块号相对应的块中。

5. 一种利用将信息存储在块单元中的存储器的数据处理设备, 其中所述存储器包括用于将数据存储在多个块单元中的数据区以及用于存储多个块号的第一和第二区, 所述多个块号被指配给所述数据区中的各块,

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息, 其中所述第一块号用于指示被原始写入数据的块, 所述第二块号用于指示数据被更新的块,

所述数据设备被设置成:

将更新后的数据写入所述数据区的一个或多个块中, 所述块对应于所述第一和第二区的其中一个区中的第二块号, 并且基于所述新旧程度信息选择所述第一和第二区的所述其中一个区,

在所述第一和第二区中的另一个区的第一块号中, 存储在所述写入步骤中已经写入更新数据的块的块号, 并且在所述第一和第二区中的所述另一个区的第二块号中, 存储数据被原始写入的块的块号; 以及

在所述第一和第二区的所述其中一个区中擦除所述第一和第二块号以及所述新旧程度信息。

6. 根据权利要求5所述的数据处理设备, 其中当所述新旧程度信息表明所述第一和第二区二者都有效时, 更新根据所述新旧程度信息被认为最新的一个区中的数据, 擦除被认为不是最新的另一个区中的数据, 并且将数据写入所述数据区的一块, 所述块对应于存储在其中一个所述区中的所述块号。

7. 根据权利要求5所述的数据处理设备, 其中被写入数据的块的块号被存储在所述另一个区中, 更新所述另一个区中的新旧信息和有效性信息, 并且擦除在所述其中一个区中的数据。

8. 根据权利要求5所述的数据处理设备, 其中当基于所述有效性信息选择所述第一和第二区的任何一个, 并且在所述另一个区中的数据还未被擦除时, 擦除在所述另一个区中的数据, 并将数据写入与所述其中一个区中存储的所述块号相对应的块中。

9. 一种利用将信息存入块单元的存储器的数据处理方法, 其中所述存储器包括用于在多个块单元中存储数据的数据区, 用于存储被指配给所述数据区中的各块以及这些块单元的更新块的块号的指针区, 以及用于存储被指配给所述指针区中的块的块号的第一和第二区,

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息, 其中所述第一块号用于指示与被原始写入数据的块相对应的所述指针区的块, 所述第二块号用于指示与数据被更新的块相对应的所述指针区的块,

所述方法包括以下步骤:

将更新后的数据写入到与存储在所述指针区中的块号相对应的数据区的一个或多个块中, 其中所述存储在指针区中的块号对应于所述第一和第二区的其中一个中的第二块号, 并且基于所述新旧程度信息选择所述第一和第二区中的所述其中一个区;

在对应于所述指针区的第一块号的块中, 存储在所述写入步骤中已经写入更新数据的所述数据区的块的块号,

在所述第一和第二区中的另一个区中, 存储在上述存储步骤中已经存储所述数据区的块号的所述指针区的第一块号; 以及

在所述第一和第二区的所述其中一个区中, 擦除所述第一和第二块号以及所述新旧程度信息。

10. 根据权利要求 9 中所述的数据处理方法, 其中当所述新旧程度信息表明所述第一和第二区二者都有效时, 更新根据所述新旧程度信息被认为最新的一个区中的数据, 擦除被认为不是最新的在另一个区中的数据, 并且将数据写入与指针区中的指针相关的所述数据区的块中, 所述指针区通过存储在所述其中一个区中的块号指针来识别。

11. 根据权利要求 9 中所述的数据处理方法, 其中被写入数据的块的块号与存储在由所述另一个区所识别的所述指针区中的指针相关, 更新在所述另一个区中的新旧程度信息和有效性信息, 并且擦除所述其中一个区中的数据。

12. 根据权利要求 9 中所述的数据处理方法, 其中当基于所述有效性信息选择所述第一和第二区的任何一个区, 并且在所述另一个区中的数据还未被擦除时, 擦除在所述另一个区中的数据, 并且将数据写入对应于存储在所述其中一个区中的所述块号的块中。

13. 一种利用将信息存入块单元的存储器的数据处理设备, 其中所述存储器包括用于在多个块单元中存储数据的数据区, 用于存储被指配给所述数据区中的各块以及这些块单元的更新块的块号的指针区, 以及用于存储被指配给所述指针区中的块的块号的第一和第二区,

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息, 其中所述第一块号用于指示与被原始写入数据的块相对应的所述指针区的块, 所述第二块号用于指示与数据被更新的块相对应的所述指针区的块,

所述数据处理设备被设置成:

将更新后的数据写入到与存储在所述指针区中的块号相对应的数据区的一个或多个块中, 其中所述存储在所述指针区中的块号对应于所述第一和第二区的其中一个中的第二块号, 并且基于所述新旧程度

信息选择所述第一和第二区中的所述其中一个区；

在对应于所述指针区的第一块号的块中，存储在所述写入步骤中已经写入更新数据的所述数据区的块的块号，

在所述第一和第二区中的另一个区中，存储在上述存储步骤中已经存储所述数据区的块号的所述指针区的第一块号；以及

在所述第一和第二区的所述其中一个区中，擦除所述第一和第二块号以及所述新旧程度信息。

14. 根据权利要求 13 所述的数据处理设备，其中当所述新旧程度信息表明所述第一和第二区二者都有效时，更新根据所述新旧程度信息被认为最新的一个区中的数据，擦除被认为不是最新的在另一个区中的数据，并且将数据写入与指针区中的指针相关的所述数据区的块中，所述指针区通过存储在所述其中一个区中的块号指针来识别。

15. 根据权利要求 13 所述的数据处理设备，其中被写入数据的块的块号与存储在由所述另一个区所识别的所述指针区中的指针相关，更新在所述另一个区中的新旧程度信息和有效性信息，并且擦除所述其中一个区中的数据。

16. 根据权利要求 13 所述的数据处理设备，其中当基于所述有效性信息选择所述第一和第二区的任何一个区，并且在所述另一个区中的数据还未被擦除时，擦除在所述另一个区中的数据，并且将数据写入对应于存储在所述其中一个区中的所述块号的块中。

数据处理方法和设备

本申请是由索尼株式会社于1998年7月9日申请的、申请号为98115463.8、发明名称为“数据处理方法和设备”一案的分案申请。

技术领域

本发明涉及一种数据处理方法和一种数据处理设备。具体地说，它涉及这样一种数据处理方法和数据处理设备，它们最好是用于例如完成数据发送和接收的IC（集成电路）卡或类似装置中。

背景技术

已经研制出一种IC卡（智能卡），期望在电子货币系统或保安系统中找到应用。

该IC卡把用于实现各种处理的CPU（中央处理单元）与处理所必须的存储数据用存储器或类似装置集成到一起，其中的数据发送和接收是在IC卡进入与预定读出器/写入器（R/W）接触状态时实现的。

另一方面，有一种自身不带电池的无电池型IC卡，其电源是由R/W单元向IC卡提供的。

再有，有一种IC卡完成该IC卡与R/W单元之间的数据发送与接收，但通过使用电磁波使IC卡与R/W单元没有物理接触，该IC卡还从电磁波中获得必要的电源。

然而，当在没有物理接触的IC卡和R/W之间发送和接收数据时，如果在访问IC卡的存储器的同时，接收电磁波失败，因而不能得到足够的电源，这时便会发生称作存储器讹误（memory corruption）的问题，而且在存储器中的数据当中发生兼容性缺陷。

再有，当使IC卡与R/W接触借以发送和接收数据时，在用户能自由地向R/W插入或从R/W拔出IC卡的情况下，如果在访问存储器

的同时从 R/W 中拔出 IC 卡，则也会发生存储器讹误。

当数据被存放在存储数据单元（在 MS-DOS 情况下为扇区）中时，例如在 MS-DOS（微软磁盘操作系统）（注册商标）的文件分配表（FAT）中，如果在 FAT 中发生存储器讹误，数据（文件）的所有位置信息会丢失，于是数据不能被访问。

因此，当发生存储器讹误时，在最坏的情况下，IC 卡不能被使用，所以需要某种对策来克服这种存储器讹误。

对付这一问题的一种方法会是这样：保留来自紧挨着的前一场合和两个场合以前的数据，并把来自两个场合以前的数据更新为新数据，这样，即使数据不能被满意地更新（当来自两个场合之前的数据是被破坏数据时），来自紧挨着的前一个场合的数据仍会保持下来。

然而，当这样做时，例如如果在来自两个场合以前的数据正在被更新时供电停止，就会出现一种不稳定情况，此时不可能读出被更新的数据，因为被读的数据可能是正确的，或者不正确的，或者是立即前一个场合的数据。

发明内容

考虑这种情况，本发明提供了一种方法，它总能保证稳定地读出一个数据。

根据本发明的一个方面，提供了一种利用将信息存储在块单元中的存储器的数据处理方法，其中所述存储器包括用于将数据存储多个块单元中的数据区以及用于存储多个块号的第一和第二区，所述多个块号被指配给所述数据区中的各块，

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息，其中所述第一块号用于指示被原始写入数据的块，所述第二块号用于指示数据被更新的块，

所述方法包括以下步骤：

将更新后的数据写入所述数据区的一个或多个块中，所述块对应于所述第一和第二区的其中一个区中的第二块号，并且基于所述新旧

程度信息选择所述第一和第二区的所述其中一个区，

在所述第一和第二区中的另一个区的第一块号中，存储与所述写入步骤所写入的更新数据相对应的、已经被原始写入数据的所述数据区的块的块号，并且在所述第一和第二区中的所述另一个区的第二区号中，存储由所述写入步骤所写入的、数据已经被更新的所述数据区的块的块号，以及

在所述第一和第二区的所述其中一个区中擦除所述第一和第二块号以及所述新旧程度信息。

根据本发明的另一个方面，提供了一种利用将信息存储在块单元中的存储器的数据处理设备，其中所述存储器包括用于将数据存储在多个块单元中的数据区以及用于存储多个块号的第一和第二区，所述多个块号被指配给所述数据区中的各块，

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息，其中所述第一块号用于指示被原始写入数据的块，所述第二块号用于指示数据被更新的块，

所述数据设备被设置成：

将更新后的数据写入所述数据区的一个或多个块中，所述块对应于所述第一和第二区的其中一个区中的第二块号，并且基于所述新旧程度信息选择所述第一和第二区的所述其中一个区，

在所述第一和第二区中的另一个区的第一块号中，存储与所述写入步骤所写入的更新数据相对应的、已经被原始写入数据的所述数据区的块的块号，并且在所述第一和第二区中的所述另一个区的第二区号中，存储由所述写入步骤所写入的、数据已经被更新的所述数据区的块的块号，以及

在所述第一和第二区的所述其中一个区中擦除所述第一和第二块号以及所述新旧程度信息。

根据本发明的再一个方面，提供了一种利用将信息存入块单元的存储器的数据处理方法，其中所述存储器包括用于在多个块单元中存储数据的数据区，用于存储被指配给所述数据区中的各块以及这些块

单元的更新块的块号的指针区，以及用于存储被指配给所述指针区中的块的块号的第一和第二区，

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息，其中所述第一块号用于指示与被原始写入数据的块相对应的所述指针区的块，所述第二块号用于指示与数据被更新的块相对应的所述指针区的块，

所述方法包括以下步骤：

将更新后的数据写入到与存储在所述指针区中的块号相对应的数据区的一个或多个块中，其中所述存储在所述指针区中的块号对应于所述第一和第二区的其中一个中的第二块号，并且基于所述新旧程度信息选择所述第一和第二区中的所述其中一个区；

在与所述指针区的第二块号相对应的块中，存储已经写入数据的所述数据区的块的块号；

在所述第一和第二区中的另一个区中，存储已经存储所述数据区的块号的所述指针区的第二块号；以及

在所述第一和第二区的所述其中一个区中，擦除所述第一和第二块号以及所述新旧程度信息。

根据本发明的再一个方面，提供了一种利用将信息存入块单元的存储器的数据处理设备，其中所述存储器包括用于在多个块单元中存储数据的数据区，用于存储被指配给所述数据区中的各块以及这些块单元的更新块的块号的指针区，以及用于存储被指配给所述指针区中的块的块号的第一和第二区，

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息，其中所述第一块号用于指示与被原始写入数据的块相对应的所述指针区的块，所述第二块号用于指示与数据被更新的块相对应的所述指针区的块，

所述数据处理设备被设置成：

将更新后的数据写入到与存储在所述指针区中的块号相对应的数据区的一个或多个块中，其中所述存储在所述指针区中的块号对应于

所述第一和第二区的其中一个中的第二块号，并且基于所述新旧程度信息选择所述第一和第二区中的所述其中一个区；

在与所述指针区的第二块号相对应的块中，存储已经写入数据的所述数据区的块的块号；

在所述第一和第二区中的另一个区中，存储已经存储所述数据区的块号的所述指针区的第二块号；以及

在所述第一和第二区的所述其中一个区中，擦除所述第一和第二块号以及所述新旧程度信息。

根据本发明的再一个方面，提供了一种利用将信息存储在块单元中的存储器的数据处理方法，其中所述存储器包括用于将数据存储在多个块单元中的数据区以及用于存储多个块号的第一和第二区，所述多个块号被指配给所述数据区中的各块，

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息，其中所述第一块号用于指示被原始写入数据的块，所述第二块号用于指示数据被更新的块，

所述方法包括以下步骤：

将更新后的数据写入所述数据区的一个或多个块中，所述块对应于所述第一和第二区的其中一个区中的第二块号，并且基于所述新旧程度信息选择所述第一和第二区的所述其中一个区，

在所述第一和第二区中的另一个区的第一块号中，存储在所述写入步骤中已经写入更新数据的块的块号，并且在所述第一和第二区中的所述另一个区的第二块号中，存储数据被原始写入的块的块号；以及

在所述第一和第二区的所述其中一个区中擦除所述第一和第二块号以及所述新旧程度信息。

根据本发明的再一个方面，提供了一种利用将信息存储在块单元中的存储器的数据处理设备，其中所述存储器包括用于将数据存储在多个块单元中的数据区以及用于存储多个块号的第一和第二区，所述多个块号被指配给所述数据区中的各块，

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息，其中所述第一块号用于指示被原始写入数据的块，所述第二块号用于指示数据被更新的块，

所述数据设备被设置成：

将更新后的数据写入所述数据区的一个或多个块中，所述块对应于所述第一和第二区的其中一个区中的第二块号，并且基于所述新旧程度信息选择所述第一和第二区的所述其中一个区，

在所述第一和第二区中的另一个区的第一块号中，存储在所述写入步骤中已经写入更新数据的块的块号，并且在所述第一和第二区中的所述另一个区的第二块号中，存储数据被原始写入的块的块号；以及

在所述第一和第二区的所述其中一个区中擦除所述第一和第二块号以及所述新旧程度信息。

根据本发明的再一个方面，提供了一种利用将信息存入块单元的存储器的数据处理方法，其中所述存储器包括用于在多个块单元中存储数据的数据区，用于存储被指配给所述数据区中的各块以及这些块单元的更新块的块号的指针区，以及用于存储被指配给所述指针区中的块的块号的第一和第二区，

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息，其中所述第一块号用于指示与被原始写入数据的块相对应的所述指针区的块，所述第二块号用于指示与数据被更新的块相对应的所述指针区的块，

所述方法包括以下步骤：

将更新后的数据写入到与存储在所述指针区中的块号相对应的数据区的一个或多个块中，其中所述存储在指针区中的块号对应于所述第一和第二区的其中一个中的第二块号，并且基于所述新旧程度信息选择所述第一和第二区中的所述其中一个区；

在对应于所述指针区的第一块号的块中，存储在所述写入步骤中已经写入更新数据的所述数据区的块的块号，

在所述第一和第二区中的另一个区中，存储在上述存储步骤中已经存储所述数据区的块号的所述指针区的第一块号；以及

在所述第一和第二区的所述其中一个区中，擦除所述第一和第二块号以及所述新旧程度信息。

根据本发明的再一个方面，提供了一种利用将信息存入块单元的存储器的数据处理设备，其中所述存储器包括用于在多个块单元中存储数据的数据区，用于存储被指配给所述数据区中的各块以及这些块单元的更新块的块号的指针区，以及用于存储被指配给所述指针区中的块的块号的第一和第二区，

所述第一和第二区中的每一个都存储第一块号、第二块号以及存储内容的新旧程度信息，其中所述第一块号用于指示与被原始写入数据的块相对应的所述指针区的块，所述第二块号用于指示与数据被更新的块相对应的所述指针区的块，

所述数据处理设备被设置成：

将更新后的数据写入到与存储在所述指针区中的块号相对应的数据区的一个或多个块中，其中所述存储在所述指针区中的块号对应于所述第一和第二区的其中一个中的第二块号，并且基于所述新旧程度信息选择所述第一和第二区中的所述其中一个区；

在对应于所述指针区的第一块号的块中，存储在所述写入步骤中已经写入更新数据的所述数据区的块的块号，

在所述第一和第二区中的另一个区中，存储在上述存储步骤中已经存储所述数据区的块号的所述指针区的第一块号；以及

在所述第一和第二区的所述其中一个区中，擦除所述第一和第二块号以及所述新旧程度信息。

根据本发明的信息处理方法和设备的特点在于一个存储器包括数据区用于以块为单元存储数据以及第一和第二区用于存储多个块号，这些块号是指配给数据区中各块的，数据被写入数据区中的一块，该块对应于存储在所述第一和第二区之一中的一个块号，已被写入数据的块的块号被存储在所述第一和第二区中的另一个区中，而在所述第一和第二区之

一中的数据被擦除。

本发明的特点还在于该存储器还可以包括一个区按所述块单元存储数据，一个指针区用于存储多个指配给数据区中各块的块号，以及第一和第二区用于存储多个指配给指针区中各块的块号。数据被写到指针区中一块的块号，该块号对应于存储在第一和第二区之一中的块号，并写入数据区中的相应块。已被写入数据的块的块号被存储在指针区的一块中，在指针区中被存储的该块的块号被存储在第一和第二区中的另一个区，而在第一和第二区之一中存储的数据被擦除。

附图说明

图 1 是应用本发明的非接触卡系统的一种形式的典型结构。

图 2 是图 1 中的读出器/写入器 1 的典型结构的方框图。

图 3 是图 1 中的 IC 卡 2 的典型结构的方框图。

图 4A - 4C 是描述本发明基本原理的图。

图 5A - 5E 是描述本发明基本原理的图。

图 6 是描述读处理的流程图，这个读处理与具有图 4A - 4C 所示结构的存储器有关。

图 7 是描述写处理的流程图，这个写处理与具有图 4A - 4C 所示结构的存储器有关。

图 8 显示出应用单指针系统的存储器的典型结构。

图 9 显示出应用该单指针系统的存储器的典型结构。

图 10 是描述由单指针系统进行读处理的流程图。

图 11 是描述由单指针系统进行写处理的流程图。

图 12 显示应用双指针系统的存储器的典型结构。

图 13 显示应用该双指针系统的存储器的典型结构。

图 14 是描述由双指针系统进行读处理的流程图。

图 15 是描述由双指针系统进行写处理的流程图。

图 16 显示当由双指针系统完成数据读时图 3 所示 EEPROM 66 的典型结构。

图 17A 和 17B 显示一个物理块的典型结构, 该物理块是有效性确定块。

图 18A 和 18B 显示组成指针区的各物理块的举例。

图 1 显示应用本发明的非接触卡系统的一种形式的典型结构。

具体实施方式

非接触卡系统由 R/W1, IC 卡 2 及控制器 3 组成, 利用电磁波在非接触状态在 R/W1 和 IC 卡 2 之间发送和接收数据。就是说, R/W1 向 IC 卡 2 发送一个预定命令, IC 卡 2 接收这个命令并根据该命令进行处理。再有, IC 卡 2 根据处理结果向 R/W1 发送响应数据。

R/W1 通过一个预先确定的接口(例如遵守 RS-485A 规范或类似规范的接口)与控制器 3 相连, 控制器 3 通过向 R/W1 提供预定控制信号来完成预定处理。

图 2 显示图 1 中的 R/W1 的典型结构。

在 IC21 中, 用于处理数据的 DPU(数据处理单元) 31、用于处理发送到 IC 卡 2 的数据及从 IC 卡 2 收到的数据的 SPU(信号处理单元) 32、用于与控制器 3 通信的 SCC(串行通信控制器) 33 以及存储器单元 34 通过总线相连, 存储器单元 34 由预先存储处理数据所需信息的 ROM41 及在处理过程中暂时存储数据的 RAM42 组成。

再有, 存储预定数据的闪速(flash)存储器 22 也连到总线上。

DPU31 向 SPU32 输出一个向 IC 卡 2 发送的命令, 并从 SPU32 接收来自 IC 卡 2 的响应数据。

SPU32 根据向 IC 卡 2 发送的命令完成预定的处理(例如 BPSK(双相移键控法)调制)并把它输出给调制电路 23, 从解调电路 25 接收由 IC 卡 2 发送的响应数据, 并完成对数据的预定处理(如 BPSK)。

调制电路 23 调制一个由振荡器 26 提供的预定频率(例如 13.56MHz)的载波, 该载波被来自 SPU32 的数据进行 ASK(幅移键控法)调制, 调制电路 23 通过天线 27 以电磁波形式向 IC 卡 2 输出被调制的波。在这种情况下, 调制电路 23 以小于 1 的调制度完成 ASK

调制。就是说，即使当数据为低电平，被调制波的最大振幅也不会为零。

解调电路 25 解调通过天线 27 收到的被调制波（ASK 调制波）并向 SPU32 输出解调后的数据。

图 3 显示图 1 所示 IC 卡 2 的典型结构。在 IC 卡 2 中，IC51 接收由 R/W1 发送来的被调制波。电容器 52 包括一个 LC 电路和天线 53，并与具有预定频率（载波频率）的电磁波同步（谐振）。

在 IC51 中，RF 接口单元 61 在其 ASK 解调单元 81 中检测和解调经由天线 53 收到的被调制波（ASK 调制波），并向 BPSK 解调电路 62 及 PLL（锁相环）单元 63 输出被解调的数据。在电压调整器 82 中，被 ASK 解调单元 81 检测到的信号被稳压并作为直流电源提供给各种电路。

在 RF 接口单元 61 中，振荡电路 83 产生一个频率与数据时钟频率相同的信号并输出到 PLL 单元 63。

RF 接口单元 61 的 ASK 解调单元 84 通过改变天线 53 的负载来调制经天线 53 收到的被调制波，这是 IC 卡 2 的电源（例如，预定的开关元件根据数据切换到 ON/OFF（通/断），而预定的负载只当开关元件为 ON（通）时才平行连接于天线 53），对应于从计算单元 64 提供的数据，而这个被调制分量通过天线 53 发送给 R/W1（R/W1 的天线 27 的终端电压被改变）。

PLL 单元 63 从 ASK 解调单元 81 所提供的数据中产生一个与数据同步的时钟信号，并把该时钟信号输出到 BPSK 解调电路 62 和 BPSK 调制电路 68。

当被 ASK 解调单元 81 解调出的数据是被 BPSK 调制的的数据时，BPSK 解调电路 62 根据 PLL 单元 63 提供的时钟信号完成数据的 BPSK 解调，并把被解调的数据输出到计算单元 64。

当由 BPSK 解调电路 62 提供的数据是被编码数据时，计算单元 64 在编码器/解码器单元 92 中对数据解码并在序列发生器 91（数据写单元）（块号写单元）（第一和第二块号写单元）中处理数据。再有，

当数据不是被编码数据时，由 BPSK 解调电路 62 提供的数据被直接提供给序列发生器 91 而不受编码器/解码器单元 92 处理。

序列发生器 91 完成与作为命令向它提供的数据相对应的处理。就是说，例如序列发生器 91 向 EEPROM66（存储器单元）写入数据或从中读出数据。

计算单元 64 的奇偶检验计算单元 93 从要存储在 EEPROM66 中的数据或从已存储在 EEPROM66 中的数据计算 Reed-Solomon 代码作为奇偶检验位。

再有，在序列发生器 91 中完成预定处理后，计算单元 64 向 BPSK 调制电路 68 输出对应于该处理的响应数据（用于向 R/W1 发送的数据）。

BPSK 调制电路 68 对来自计算单元 64 的数据进行 BPSK 调制，并把调制过的数据输出到 RF 接口电路 61 的 ASK 调制单元 84。

ROM65 存储要被序列发生器 91 执行的处理程序或其他必要的数
据。RAM67 暂时存储在处理过程中的数据之类。

EEPROM66 是一个非易失存储器，即使在 IC 卡 2 已结束与 R/W1 的通信并停止电源之后，它仍继续存储数据。

下面将解释 R/W1 和 IC 卡 2 之间的数据发送与接收。

R/W1（图 2）从天线 27 发送预定的电磁波，监测天线 27 的负载状态，并等待直至检测到由于 IC 卡 2 的临近造成的负载状态的变化。

在这种情况下，R/W1 可以通过发送由具有预定短模式（pattern）的数据进行 ASK 调制的电磁波和呼叫 IC 卡 2 来完成重复处理（定时询问），直至固定的时间间隔内得到 IC 卡 2 的响应为止。

在 R/W1 中，当检测到 IC 卡 2 的临近时，R/W1 的 SPU32 完成由要发送给 IC 卡 2 的数据进行的 BPSK 调制（这些数据包括与要由 IC 卡 2 完成的处理相对应的命令和被写入 IC 卡 2 的数据等），并以具有预定频率（例如数据时钟频率的 2 倍）的矩形波作为载波，并向调制电路 23 输出被调制波（BPSK 调制信号）。

在 BPSK 调制中，能通过差分变换使数据对应于被调制波的相位

变化,在这种情况下,由于即使 BPSK 调制信号被反向,其 BPSK 调制的信号也能被解调为原始数据,所以不需要考虑被调制波的极性。

调制电路 23 完成对预定载波的 ASK 调制,其调制度 (= 数据信号最大幅度/载波最大幅度) 小于 1 (例如 0.1), 并通过天线 27 向 IC 卡 2 发送被调制波 (ASK 调制波)。

当不进行传输时,调制电路 23 产生一个例如具有数字信号二电平 (高电平和低电平) 的高电平的被调制波。

在 IC 卡 2 (图 3) 中,由 R/W1 的天线 27 发射的电磁波的一部分在天线 53 和电容 52 组成的 LC 电路中被转换成电信号,该电信号 (被调制波) 被输出到 IC51 的 RF 接口 61。然后,RF 接口 61 的 ASK 解调单元 81 通过对被调制波进行整流和平滑来完成包络检测,并把由此形成的信号提供给电压调整器 82,通过抑制信号中的直流分量来提取数据信号并把数据信号输出到 BPSK 解调电路 62 和 PLL 单元 63。

电压调整器 82 对 ASK 解调单元 81 提供的信号进行稳压,产生直流电源提供给各种电路。

在这种情况下,天线 53 的终端电压 V_0 由例如下列方程表示:

$$V_0 = V_{10}(1 + kV_s(t))\cos(\omega t)$$

这里 $V_{10} \cos(t)$ 代表载波, k 指定调制度,而 $V_s(\omega t)$ 代表从 SPU32 输出的数据。

再有,由 ASK 解调单元 81 整流的电压 V_1 的低电平值 V_{LR} 由例如下式表示:

$$V_{LR} = V_{10} (1 + k(-1)) - V_f$$

在此式中, V_f 代表在 ASK 解调单元 81 中构成完成整流和平滑的整流电路的二极管处的电压降,其值一般为 0.7 伏左右。

当电压调整器 82 收到被 ASK 解调单元 81 整流和平滑的信号时,该信号被稳压并作为直流电源提供给各种电路,包括计算单元 64。由于被解调波的解调度“ k ”小于 1,故在解调后的电压涨落 (高电平和低电平之差) 是小的。所以能在电压调整器 82 中容易地产生直流电源。

例如,当接收到调制度“ k ”为 5% 的被调制波时,如果 V_{10} 为 3

伏或更高,则在整流后的低电平电压 V_{LR} 是 $2.15 (= 3(1-0.05) - 0.7)$ 伏或更高,所以电压调整器 82 能提供足够高的电压作为各种电路的电源。再有,在这种情况下,在整流后的电压 V_1 的交流分量(数据分量)幅值 $2kV_{10}$ (峰-峰值)是 $0.3 (= 2 \times 0.05 \times 3)$ 伏或更高,因此 ASK 解调单元 81 能以足够高的信噪比(S/N)解调数据。

这样,利用调制度“k”小于 1 的 ASK 调制波能实现低差错率(具有高信噪比)的通信,并向 IC 卡 2 提供足以做为电源的直流电压。

一旦收到来自 ASK 解调电路 81 的数据信号(BPSK 调制信号),BPSK 解调电路 62 便根据 PLL 单元 63 提供的时钟信号解调数据信号,并把被调制过的数据输出到计算单元 64。

如果由 BPSK 解调电路 62 提供的数据是加密的,则计算单元 64 在编码器/解码器单元 92 中对数据解码,并把数据(命令)提供给序列发生器 91。在这期间,即在把数据传送给 IC 卡 2 之后,R/W1 在发出值为 1 的数据后等待一段时间,直至收到对它的响应。所以在这段时间里 IC 卡 2 收到有恒定最大幅度的被调制波。

在已结束处理之后,序列发生器 91 向 BPSK 调制电路 68 输出关于处理结果之类的数据(用于向 R/W1 发送的数据)。BPSK 调制电路 68 以与 R/W1 的 SPU32 类似的方式对数据进行 BPSK 调制,然后该电路向 RF 接口单元 61 的 ASK 调制单元 84 输出数据。

ASK 调制单元 84 根据要传送的数据,通过利用开关元件使连到天线 53 两端的负载发生变化,来对所收到的被调制波进行 ASK 调制(如前所述,当 IC 卡 2 传送数据时,由 R/W1 输出的被调制波最大幅度是常数)。于是 R/W1 的天线 27 的终端电压被改变,于是数据被传送到 R/W1。

另一方面,当从 IC 卡 2 收到数据时,R/W1 的调制电路 23 继续发送值为 1(高电平)的数据。由 IC 卡 2 发送的数据被解调电路 25 从天线 27 的终端电压的很小变化(例如几十微伏)中检测出来,该天线 27 是与 IC 卡 2 的天线 53 电磁耦合的。

在解调电路 25 中,被检测到的信号(ASK 调制波)被高增益放

大器放大并被解调,作为解调结果所得到的数字信号被输出到 SPU32。SPU32 解调数据 (BPSK 调制信号) 并把它输出到 DPU31。DPU31 处理来自 SPU32 的数据并根据处理结果确定是否要终止通信。再有,当确定要再次进行通信时,则如上述情况那样,在 R/W1 和 IC 卡 2 之间完成通信。

另一方面,当确定要终止通信时,R/W1 终止与 IC 卡 2 的通信处理。

如前所述,R/W1 利用调制度“k”小于 1 的 ASK 调制来向 IC 卡 2 发送数据。IC 卡 2 接收该数据,完成对应于该数据的处理,并向 R/W1 回送对应于处理结果的数据。

接下来解释 IC 卡 2 (图 3) 的序列发生器 91 向 EEPROM66 写数据和从 EEPROM66 读数据的处理过程。然而,作为准备措施,首先要解释根据本发明对付存储器讹误的一种方法。

首先,将解释对付存储器讹误的方法的基本原理。

例如,假定对存储器读、写信息是由预定的块单元完成的,并考虑存于某块 B1 的数据被更新的情况(存于块 B1 中的数据被其他数据重写)。

在这种情况下,当新数据被写入(覆盖写到已存储的数据上)到块 B1 本身时,如果在写操作过程中向 IC 卡供电失效,则新数据不能被完全写入,于是引起存储器讹误。

当要被写入块 B1 的数据被实际写入块 B1 时发生存储器讹误,则系统不能对付它。就是说,例如,如果当存储在块 B1 中的数据是电子货币的余额或类似物,进行了新的购物,在减去购物钱之后的剩余之和作为新数据按上述方法写入,如果此时发生了存储器讹误,则 IC 卡不能被使用。

因此,要被写入块 B1 的新数据被写入不同于块 B1 的块 B2。这样,如果当把新数据写入块 B2 的过程中发生了存储器讹误,则新数据不能被完全写入,于是它的有效性不能保证,但至少是已存储在块 B1 中的数据能免于被破坏。最后,在向块 B2 写入数据完成之后,存

储在块 B1 中的数据被擦除。

当提供甚至是更加新的数据时，这数据被写入不同于块 B2 的块，例如块 B1。这样，当提供新数据时，新数据被写入到与紧挨着的前一个场合被写入数据的那个块不同的块中。所以，至少是在紧挨着的前一个场合所写入的数据免于受写入新数据的破坏，而且即使在最坏的情况，IC 卡仍能被使用。

接下来参考图 4A 和 4B 来解释应用上述原理的对存储器读、写数据的方法。

在图 4A 和 4B 中，一块包括 11 个字节，其中 1 字节（8 个二进制位）为新旧程度信息，8 字节为数据，2 字节为有效性信息，它们从前端起顺序排列。然而，1 块中的位数以及分配给新旧程度信息、数据和有效性信息的位数都不特别限定于上述数量。

新旧程度信息指示该块存储内容的新旧程度。例如，可以使用绝对日期和时刻，或一个顺序数或类似量。当使用绝对日期和时刻时，在这种情况下存储数据的日期和时间便作为新旧程度信息被存储，于是能从新旧程度信息中检测出存储数据的最新块。再有，当使用顺序数时，在这种情况下，例如当每个数据被写入的场合便把一个增量数作为新旧程度信息存储起来，具有最大值的块便是写入数据所用的最新的一块。

这里，如果能识别至少两个块当中哪一个是新的，这就足够了。因此新旧程度信息至少应能代表三个状态。就是说，例如当假定使用 0、1、2 三个值作为能代表三个状态的新旧程度信息时，每次数据被写入两块中的这一个或那一个，0、1、2、0...可以被写入作为数据写入块的新旧程度信息。在这种情况下，当两块的新旧程度信息为 0 和 1 时，有 1 的块为较新的块，当它们为 1 和 2 时，有 2 的块为较新的块，而当它们为 2 和 0 时，则有 0 的块为较新的块。当使用 0、1、2 这三个值以这种方式作为新旧程度信息时，只需要二个二进制位。另一种作法是，例如可以使用三个标志作为新旧程度信息，每个标志有一个二进制位，每当数据被写入块时，和标志联系在一起的二进制位可

以相继地改变。

这里，在下文中将使用 0、1、2 三个值作为新旧程度信息。

其次，有效性信息指明眼下的新旧程度信息和数据的有效性（有效或无效），涉及该块的新旧程度信息和数据的写入是否是正常完成的，对此，可使用 RS（Reed-Solomon）代码或类似代码。

有效性信息不限于 RS 代码之类错误校正码。就是说，如果有效性信息能表明有效性信息和数据是有效或无效（如前面描述的那样），便也就足够了，该有效性信息可以是例如在写新旧程度信息和数据时计算出的奇偶检验码，或者可以是例如 1 位的标志，它是在正确写入新旧程度信息和数据之后添加的。

在图 4A 和 4B 中，向存储器提供二个块，它们是第一区和第二区。图 4A 所显示的情况中数据 01H 至 08H（H 代表一个 16 进制数）已经被写入第一区块，而在第二区块中的数据已被抹去。

在这种情况下，假定例如提供了 11H 至 18H 作为新数据，首先，通过参考有效性信息来确定第一和第二区这两个块的有效性。这里，假定只有第一区块是有效的。

在写新数据的情况中，如前所述，为了防止在紧挨着的前一个场合写入的数据被破坏，新数据被写到这样的一块，其中数据尚未被记录。因此，在这种情况下，如图 4（B）所示，数据 11H 至 18H 被写到第二区块。在完成写数据 11H 至 18H 之后，首先，向其写数据的块的新旧程度信息被更新。就是说，由于曾是最新块的第一区块的新旧程度信息被指定为 01H，现在刚刚已经写入数据的第二区块的新旧程度信息被指定为 02H，它是对 01H 增量 1 得到的，然后完成写有效性信息，它表示新旧程度信息 02H 的有效性。

最后，在对第二区块的全部写完成之后，记录在第一区块的全部数据被擦除，例如如图 4C 所示。

接下来，参考图 6 和图 7 的流程图更详细地描述在具有图 4 所示结构的存储器中的数据读/写处理过程。

首先，将参考图 6 的流程图来描述读处理过程。

首先，在步骤 S1，通过访问第一和第二区的有效性信息，确定有效性。当在步骤 S1 确定只有第一区沟块有效时，该例程进入步骤 S2 并从第一区（它是有效块）读数据，于是过程终止。再有，当在步骤 S1 确定只有第二区的块有效时，该例程进入步骤 S3，并从第二区（它是有效块）读数据，于是过程终止。当确定第一和第二两区的块都无效时，IC 卡 2 的硬件寿命到了尽头，读处理不能完成（进行错误处理）。

接下来，将参考图 7 解释写处理过程。

在这种情况下，与图 6 中步骤 S1 的情况类似，在步骤 S11 中确定有效性。当第一和第二两区都无效时，IC 卡 2 发生了致命错误，例如由于寿命结束，于是进行错误处理。在步骤 S11，当确定第一区块有效时，例程进入步骤 S12，并确定是无效块的第二区中的数据是否已被完全擦除。当确定这数据尚未被完全擦除时，例程进入步骤 S20，当确定这数据已被完全擦除时，例程进入步骤 S13。在步骤 S20，在第二区（它是无效块）中的数据被完全擦除，于是例程进入步骤 S13。

在步骤 S13，新数据连同新旧程度和有效性信息被写入（更新）到其中数据被擦除的第二区块，于是例程进入步骤 S14。在步骤 S14，第一区块中的数据被擦除。

在步骤 S11，当确定第二区块有效时，例程进入步骤 S15，并确定是无效块的第一区中的数据是否已被完全擦除。当确定这数据尚未被完全擦除时，例程进入步骤 S22，当确定这数据已被完全擦除时，例程进入步骤 S16。在步骤 22，在第一区（它是无效块）中的数据被完全擦除，于是例程进入步骤 S16。

在步骤 S16，新数据连同新旧程度和有效性信息被写入（更新）到其中数据被擦除的第一区块，于是例程进入步骤 17。在步骤 17，第二区块中的数据被擦除。

如果例如在擦除旧数据之前或当在步骤 S14 或 S17 中擦除旧数据期间供电失败，则可能发生旧数据仍未被擦除或只有一部分数据被更新，从而如图 5A 所示奇偶检验恰好是正确的，而发生了存储器讹误。在这种情况下，在步骤 S11 两块中的数据被确定是有效的，于是例程

进入步骤 S18。

在步骤 S18，当确定第二区比第一区新时，例程进入步骤 S21，数据连同其新旧程度和有效性信息被再写到新块（此时新块是第二区），例如如图 5B 所示，然后例程进入步骤 S22。接下来在步骤 S22，以前述同样的方式，在第一区（它不是最新块）中的数据被擦除（例如如图 5C 所示），例程进入步骤 S16。

在步骤 S16，新数据连同其新旧程度和有效性信息被写到（更新）其中数据已被擦除的第一区块，然后例程进入步骤 S17。在步骤 S17，第二区块中的数据被擦除，例如如图 5E 所示。

在步骤 S18，当确定第一区比第二区新时，例程进入步骤 S19，数据连同其新旧程度和有效性信息被再写到新块（此时新块是第二区）。然后例程进入步骤 S20。接下来在步骤 S20，以前述同样方式，在第二区（它不是最新块）中的数据被擦除，例程进入步骤 S13。在步骤 S13，新数据连同其新旧程度和有效性信息被写入（更新）到其中数据被擦除的第二区块，然后例程进入步骤 S14。在步骤 S14，第一区块中的数据被擦除。

每次读数据时，通过允许有足够的时间用于记录，从而避免了数据被认为是正确或不正确这种不稳定状态。由于上述处理，擦除第一区或第二区意味着可得到足够的时间用于在另一剩余区间中记录。所以可以避免读这一被记录数据时出现不稳定状况。

从外部看，第一和第二区这两个块被识别为同样的逻辑块。具体地说，第一和第二区块是物理上存在的两个物理块，而这些物理块被分别指配给物理块号，这些块号是标识物理块的唯一号码。第一和第二区的物理块的物理块号被做成对应于用来识别逻辑块的同一逻辑块号，利用这一逻辑块号由外部识别第一和第二区的物理块。

在下述内容中，物理块号和逻辑块号都假定由一字节表示。

在上述情况中，在完成对一物理块的数据写入之后，该物理块的新旧程度信息和有效性信息都被更新，所以当多个相关的数据被写入多个物理块时，可能会产生新旧数据的混合。从外部看来，如果在把

多个数据写入多个逻辑块时发生了存储器讹误，虽然在存储器讹误发生之前写入新数据的逻辑块中存在新数据，但从发生存储器讹误的逻辑块到如果未曾发生存储器讹误则新数据会已经写入的逻辑块这一范围内存在旧数据。

在多个相关的数据作为整体是有意义的因而数据整体是有用的这样一种情况下，如前所述，当新数据和旧数据混合存在于多个相关数据中时，便不能建立起兼容性。就是说，例如有这样一种情况：多个相关联的数据是上火车的车站和时间，下火车的车站和时间，以及上下火车的两车站之间的车费，那么，如前所述，当被写入的新数据是关于上车的车站，而关于上车的时间仍为旧数据，这样便不能建立起兼容性，因数据是无意义的。

因此，在把多个数据写入多个逻辑块的情况中如果在写这多个数据完成之前发生存储器讹误，则从外部看来，必须在这个逻辑块中存在多个相关联数据中的所有旧数据。

因此，一个存储器具有例如图 8 所示的结构。

就是说，存储器包括存储数据用的数据区物理块，以及第一和第二区物理块，用于存储构成数据区所用物理块的物理块号。

再有，在与图 4 和图 5 的情况相似的图 8 中，一个物理块由 11 个字节组成。

由数据区组成的各物理块把数据存储在这些 11 字节单元中，而在图 8 中，数据区由 8 个物理块组成，分别被指配给#00H 到#07H 作为物理块号（#代表物理块号）。

在这种情况下，当从外部看时，4 个逻辑块是可见的，并有 %00H 至 %03H 被分配给这 4 个逻辑块的每一个作为逻辑块号。在图 8 所示状态中，逻辑块号 %00H 至 %03H 被做成分别对应于物理块号 #00H 至 #03H。

在第一和第二区的物理块中，头一个字节被分配给新旧程度信息，而最后两个字节被分配给有效性信息，这是图 4 所示共同特征。然而，在新旧程度信息和有效性信息之间的 8 个字节中安排是数据区中物理

块的物理块号，即指向构成数据区的各物理块的指针。

在第一和第二区的新旧程度信息和有效性信息之间的 8 字节中，数据区的目标块的物理块号被安排在头 4 个字节中，而数据区的更新块的物理块号被安排在后 4 个字节。

在本文的上下文中，目标块是指信息应被写入的物理块。就是说，例如，根据前述基本原理，当存储在块 B1 中的数据被新数据重写时，新数据被写入块 B2，但在这种情况下，块 B1 是新数据应被写入的块，因此块 B1 对应于目标块。

更新块是指当信息被写到一个物理块而其存储内容被更新时被用来进行更新的物理块。就是说，例如，根据基本原理，当存储在块 B1 中的数据被新数据重写时，块 B2 被实际写入新数据，因此块 B2 对应于更新块。

在第一和第二区的新旧程度信息和有效性信息之间的 8 个字节的头 4 个字节中安排的 4 个物理块号被做成对应于逻辑块号。就是说，例如，从其第一到第四字节的物理块号分别对应于逻辑块号 %00H 至 %03H。

再有，在图 8 中，第一和第二区的物理块号分别被指配给 #FE 或 #FF。

在具有上述结构的存储器中，在图 8 中，4 个相关联的数据被存储在物理块 #00H（有物理块号 #00H 的物理块）至 #03H 中，而其物理块号 #00H 至 #03H 被按此顺序安排在第一区中新旧程度信息和有效性信息之间的 8 个字节的前 4 个字节中（下文中将恰当地称作指向数据区中目标块的指针列）。

因此，例如观察第一区，物理块号 #00H 至 #03H 被做成分别对应于逻辑块号 %00H 至 %03H。

在这种情况下，观察第一区，被做成对应于逻辑块 %00H（逻辑块号为 %00H 的逻辑块）至 %03H 的物理块 #00H 至 #03H 中的任何一个都应是可访问块，即目标块。

更新块可以是在那时不构成目标块的物理块，或者可以说是空闲

区。

相应地，在图 8 中，观察第一区，物理块 # 04H 至 # 07H 可以是更新块。更新块的物理块号 # 04H 至 # 07H 被安排在第一区的新旧程度信息和有效性信息之间的 8 个字节中的后 4 个字节（下文中将恰当地称作指向数据区的更新块的指针列）。

现在，例如假定第一和第二区中只有第一区是有效的，而从外部向对应于逻辑块号 % 00H 和 % 02H 的逻辑块做出了写两个相关联数据的请求。

在这种情况下，通过访问指向有效第一区中的数据区的目标块的指针列，识别出了对应于逻辑块号 % 00H 和 % 02H 的物理块号 # 00H 和 # 02H，而物理块 # 00H 和 # 02H 被识别出作为要写入两个相关联数据的目标块。

再有，通过访问指向这第一区的数据区的更新块的指针列，识别出更新块的物理块号为 # 04H 和 # 05H，它们的块号等于目标块的块号。例如，从指向数据区的更新块的指针列的左侧依次识别出更新块的物理块号。

本应写入作为目标块的物理块 # 00H 或 # 02H 的两个数据结果被分别写入作为更新块的物理块 # 04H 或 # 05H，如同图 9 所示。

在这种情况下，图 9 所示状态是：要写入对应于逻辑块号 % 00H 至 % 02H 的物理块号 # 00H 和 # 02H 的两个数据分别假定为 40AH 至 4AH 和 50AH 至 5AH，这些数据被分别写入作为更新块的物理块 # 04H 和 # 05H。

然后，被写入新数据的作为更新块的物理块的物理块号 # 04H 和 # 05H 被写入到指向无效的第二区中数据区的目标块的指针列。就是说，指向第一区中数据区的目标块的指针列被复制到第二区，并且如图 9 所示，目标块的物理块号 # 00H 或 # 02H 被分别重写成数据被实际写入的更新块的物理块号 # 04H 或 # 05H。结果，观察第二区，物理块号 # 04H 和 # 05H 被分别做成对应于逻辑块号 % 00H 或 % 02H，而物理块号 # 00H 或 # 02H 曾被做成对应于这两个逻辑块号。

再有，曾要被写入数据的物理块 # 00H 和 # 02H 现在变成了更新块，而物理块号 # 00H 和 # 02H 被写到指向无效的第二区中的数据区的更新块的指针列中。就是说，指向第一区中的数据区的更新块的指针列被复制到第二区，而且如图 9 所示，曾写入数据的物理块的物理块号 # 04H 或 # 05H 分别被目标块的物理块号 # 00H 或 # 02H 所重写。结果，观察第二区，物理块 # 00H 和 # 02H 变成更新块。

在如前所述更新第二区之后，与图 4 所示情况类似，第二区中的新旧程度信息和有效性信息依次被写入（重写），然后第一区中的数据被完全擦除。

所以在这种情况中，除非在第二区中完成了写入指向数据区中目标块的指针列，否则不会被写入指明其有效性的有效性信息。就是说，在上述情况中，除非完成了对两个逻辑块 % 00H 和 % 02H 二者写入数据，否则不会向第二区中写入指明其有效性的有效性信息。

结果，如果在数据正在写入两个逻辑块 % 00H 和 % 02H 的时间区间内引起了存储器讹误，只有第一区仍然有效，那么通过访问有效的第一区，存储在物理块 # 00H 和 # 02H（它们曾分别被做成对应于逻辑块 % 00H 和 % 02H）中的数据被从逻辑块 % 00H 和 % 02H 中读出。

所以，当作为多个数据存储于逻辑块 % 00H 和 % 02H 的两个数据是彼此相关联的数据时，即使当发生存储器讹误时，也不仅能保持单个数据的兼容性，而且能保持两个数据之间的兼容性。

接下来，将参考图 10 和图 11 的流程图，进一步解释在具有图 8 和图 9 所示结构的存储器中数据的读、写处理过程。

首先，将参考图 10 的流程图来描述读处理。

在这种情况下，首先在步骤 S31 中完成与图 6 中的步骤 S1 中类似的确定处理。

当第一区为有效时，例程进入步骤 S32，而当第二区为有效时，例程进入步骤 S34。

在步骤 S32，通过访问指向第一区中数据区的目标块的指针列，识别出了一个对应于要读数据的逻辑块号（这个逻辑块号例如是由外

部提供的)的物理块号,即作为指向一个目标块的指针的物理块号,然后例程进入步骤 S33。在步骤 S33,从对应于在步骤 S33 中识别出的物理块号的物理块中读出数据,于是处理过程终止。

另一方面,在步骤 S34,通过访问指向第二区中数据区的目标块的指针列,识别出一个对应于要读数据的逻辑块号的物理块号,即作为指向一个目标块的指针的物理块号,然后例程进入步骤 S35。在步骤 S35,从对应于在步骤 S34 中识别出的物理块号的物理块中读出数据,于是处理过程终止。

接下来,将参考图 11 的流程图来描述写处理过程。

在这种情况下,在步骤 S41,也与在图 7 中的步骤 S11 类似,完成确定处理。

当只有第一区为有效时,例程进入步骤 S42,并在那里确定在无效的第三区中的数据是否已被完全擦除。当确定数据未被完全擦除时,例程进入步骤 S54,而当确定数据已被完全擦除时,例程进入步骤 S43。在步骤 S54,作为无效块的第三区中的数据被擦除,例程进入步骤 S43。在步骤 S43,通过访问指向第一区中数据区的更新块的指针列,识别出能作为更新块的物理块的物理块号,然后例程进入步骤 S44。在步骤 S44,例如由外部提供的要写数据被写入对应于所识别出的物理块号的物理块,即更新块。

再有,当由外部提供要写入多个逻辑块的数据时,在步骤 S43,识别出更新块的物理块号,这些更新块的个数等于多个逻辑块的个数。在这种情况下,在步骤 S44,来自外部的数据依次写入分别对应于所识别出的多个物理块号的更新块。

然后,例程进入步骤 S45,指向第一区中数据区的目标块的指针列和指向第一区中数据区的更新块的指针列都被更新,而且更新后的新旧程度信息和有效性信息被复制到第二区。

就是说,关于指向第一区中的数据区的目标块的指针列,被做成与那些要从外部写入数据的逻辑块的逻辑块号相对应的物理块号(在图 8 和图 9 的例中为 #00H 和 #02H)被更新为数据被实际写入的更

新块的物理块号（在图 8 和图 9 的例中为 # 04H 和 # 05H）。再有，关于指向第一区中的数据区的更新块的指针列，被写入数据的物理块（这些物理块是更新块）的物理块号（在图 8 和图 9 的例中为 # 04H 和 # 05H）被更新为被做成与被写入数据的逻辑块的逻辑块号相对应的物理块号（在图 8 和图 9 的例中为 # 00H 和 # 02H）。再有，新旧程度信息和有效性信息被更新，上述更新结果被写入第二区，然后例程进入步骤 S46。在步骤 S46，作为第一区的块中的数据被擦除。

另一方面，当在步骤 S41 中确定第二区有效时，在步骤 S47 至 S51 及步骤 S56 中完成与步骤 S42 至 S46 及步骤 S54 中完成的处理相同的处理，只是对第一和第二区完成的处理分别反过来变成对第二和第一区的处理。

当发生存储器讹误时，如图 7 的步骤 S18 那样，在步骤 S52 确定已被定为第一区或第二区的块的新旧程度。在步骤 S52，当确定第二区比第一区新时，例程进入步骤 S55，指向第二区中数据区的目标块的指针列和指向第二区中数据区的更新块的指针列都被更新，并在步骤 S56 完成擦除不是最新的第一区。其后，从步骤 S48 到 S51，其处理过程与前面所述相同。

另一方面，如果在步骤 S52 确定第一区比第二区新，则在步骤 S53、步骤 S54 以及步骤 S43 至 S46 完成与步骤 S55、S56 及步骤 S48 至 S51 相同的处理，只是对第一和第二区的处理分别反过来变成对第二和第一区的处理。

参考图 8 至图 11 所描述的存储器读、写系统使用指向数据区的一级指针，因此，在下文中恰当地称作单指针系统。

在单指针系统的情况中，如图 8 和图 9 所示，在指向数据区的目标块的指针列中能存储 4 个物理块号，所以最多能对 4 块保持其数据兼容性。

在这种情况下，如图 8 和图 9 所示，与指向数据区的目标块的指针列相似，在指向数据区的更新块的指针列中能存储 4 个物理块号，因为也需要 4 个更新块来写 4 块数据，这是能保持兼容性的最大块数。

当把单指针系统应用于图 8 和图 9 所示存储器结构时,如前所述,最多能对 4 块保持数据兼容性,即难于对多于 4 块保持兼容性。就是说,当由第一和第二区的物理块来存储和管理构成数据区的物理块的物理块号时,能保持数据兼容性的块的个数受限于第一和第二区的物理块中能存储的块号的个数。

因此,存储器可以有例如如图 12 所示的结构。

就是说,在这种情况下,存储器的组成是:作为存储数据的数据区的物理块,作为存储块号(这些块号是指向构成数据区的物理块的指针)的指针区的物理块,以及作为存储块号(这些块号是指向构成指针区的物理块的指针)的第一和第二区的物理块。

在图 12 中,为了方便,一个物理块占 8 个字节。

构成数据区的各物理块把数据存于 8 个字节单元中,在图 12 中的数据区包括 24 个物理块,分别被指定物理块号 #00H 至 #17H。

在这种情况下,16 个逻辑块是从外部可见的,这 16 个逻辑块分别被指配给逻辑块号 %00H 至 %0FH。然而,在这种情况下,为了解释方便,假定存在的逻辑块个数与构成数据区的物理块个数相同,除了从外部可见作为逻辑块的物理块以外,另 8 个物理块分别被分配虚逻辑块号 %10 至 %17。在图 12 所示状态中,逻辑块号 %00H 至 %17H 被做成分别对应于物理块号 #00H 至 #17H。

构成指针区(块号区)的物理块把物理块号(这些物理块号是指向构成数据区的物理块的指针)存储在 8 个字节单元,在图 12 中指针区包括 6 个物理块,分别被指定物理块号为 #18 至 #1D。在这种情况下,构成指针区的物理块把物理块号存储在 8 个字节单元中,而一个物理块号由一个字节表示,故一个物理块可存储 8 个物理块号。

再有,安排在指针区的每个物理块中的 8 个物理块号被做成与逻辑块号相对应。就是说,从第 1 字节到第 8 字节安排的物理块号被做成所对应的逻辑块号是以某一逻辑块号(如下文中提到的 %00H、%08H 或 %10H)为基值从 0 至 7 偏移(移动)得到的。

在第一和第二区,分别将其第一字节分配给新旧程度信息,而将

其最后一个字节分配给有效性信息。再有，作为指向指针区物理块的指针的物理块号被安排在新旧程度信息和有效性信息之间的6个字节中。

就是说，在第一和第二区的新旧程度信息和有效性信息之间的这6个字节当中，前3个字节中安排指针区目标块的物理块号（此后把这3个字节恰当地称作指向指针区目标块的指针列），而在后3个字节中安排指针区更新块的物理块号（此后把这3个字节恰当地称作指向指针区更新块的指针列）。

指向第一和第二区的指针区目标块的指针列中安排了3个物理块号，与这些物理块号相对应的指针区物理块所存储的物理块号被做成与逻辑块号相对应。就是说，与第1至第3字节中的物理块号相对应的指针区物理块所存储的物理块号分别被做成对应于逻辑块号%00H至%07H、%08H至%0FH、或%10H至%17H。

具体地说，例如在图12中，观察第一区，指向指针区目标块的指针列的第1字节中为物理块号#18H，由它指出的指针区物理块第1至第8字节中安排了物理块号\$00H至#07H，这些物理块号被做成分别对应于逻辑块号%00H至%07H，这些逻辑块号被做成对应于指向第一区的指针区目标块的指针列的第1字节。就是说，在指向指针区目标块的指针列的第1字节中的物理块号所指出的指针区物理块第1至第8字节中安排的物理块号被做成分别对应于物理块号%00H至%07H，它们是以逻辑块号%00H为基值加上0至7偏移值。

类似地，在指向指针区目标块的指针列的第2字节中的物理块号所指出的指针区物理块第1至第8字节中安排的物理块号被做成分别对应于逻辑块号%08H至%0FH，它们是以逻辑块号%08H为基值加上0至7偏移值。

再有，在图12中，第一和第二区的物理块号分别被指定为#1E或#1F。

在具有前述结构的存储器中，例如观察第一区，指向指针区目标块的指针列中给出的物理块号所对应的物理块包括能被访问的指针区

目标块。因此，在图 12 中，物理块 # 18H 至 # 1AH 构成指针区的目标块。再有，在图 12 中，在指针区中的其余物理块 # 1BH 至 # 1DH 是用于更新指针区的更新块，所以物理块号 # 1BH 至 # 1DH 被描述为指向第一区中指针区的更新块的指针列。

再有，在图 12 中，与指针区目标块所描述的物理块相对应的逻辑块 % 00H 至 % 0FH 所对应的数据区物理块是数据区的目标块。因此，在图 12 中，物理块 # 00H 至 # 0FH 构成数据区的目标块，而数据区的其余物理块 # 10H 至 # 17H（对应于逻辑块 % 10H 至 % 17H 的物理块）构成更新数据区用的更新块。

在这种情况下，如前所述，指针区更新块的物理块号是在指向指针区更新块的指针列中给出的，所以通过访问该指针列能识别出更新块。

另一方面，由于数据区的更新块是对应于逻辑块 % 10H 至 % 17H 的物理块，通过访问由指向指针区目标块的指针列中第 3 字节描述的物理块号所对应的指针区物理块，便能从那里所描述的物理块号识别出更新块。

现在假定第一或第二区中只有一个区是有效的，例如第一区有效，而从外部发出了向对应于逻辑块号 % 00H 和 % 02H 的逻辑块写入两个相关联数据的请求。

在这种情况下，通过访问指向第一区（它是有效的）中指针区目标块的指针列和访问对应于那里所描述的物理块号的物理块，便能识别出要写入这两个相关联数据的目标块，它们是数据区的物理块。就是说，首先如前所述，通过访问指向第一区中指针区目标块的指针列的第一字节，能识别出指针区的物理块 # 18H，其中给出了对应于逻辑块 % 00H 和 % 02H 的数据区中物理块的物理块号。再有，通过访问物理块 # 18H 的第 1 字节和第 3 字节，能分别识别出对应于逻辑块 % 00H 和 % 02H 的数据区物理块（目标块）的物理块号 # 00H 和 # 02H。

然后，识别出数据区的更新块。就是说，如前所述，通过访问指向第一区中指针区目标块的指针列的第 3 字节，能识别出指针区的物

理块 # 1AH, 其中给出了对应于逻辑块 % 10H 至 % 17H 的构成数据区更新块的物理块的物理块号。再有, 通过访问物理块 # 1AH, 能识别出构成数据区更新块的物理块的物理块号 # 10H 至 # 17H。这里, 通过访问物理块 # 1AH, 例如从左侧开始访问, 便能识别出那些物理块, 它们的块数等于作为数据区更新块被写入数据的数据区的目标块的块数。所以, 对应于物理块号 # 10H 和 # 11H 的数据区物理块被作为更新块对待。

因此, 如图 13 所示, 本应被写入物理块 # 00H 或 # 02H (它们是数据区的目标块) 数据被分别写入物理块 # 10H 或 # 11H (它们是数据区的更新块)。

在这种情况下, 图 13 所示状态是: 要写入对应于逻辑块号 % 00H 和 % 02H 的物理块号 # 00H 和 # 02H (分别由 8 字节 00H 和 8 字节 02H 指定) 之中的两个数据被分别写入物理块 # 10H 和 # 11H, 它们是数据区的更新块。

如前所述, 在把要写入对应于物理块号 # 00H 和 # 02H (由对应于第一区中指针区物理块号 # 18H 的物理块存储这些物理块号) 的两个数据分别写入数据区的更新块 # 10H 和 # 11H 之后, 刚才作为数据区更新块并向其中写入数据的物理块的物理块号 # 10H 和 # 11H 被写入指针区的更新块, 而本应写入数据的数据区物理块 (目标块) 的物理块号 # 00H 和 # 02H 被写入指针区的其他更新块。

就是说, 首先, 通过访问指向第一区中指针区更新块的指针列, 识别出指针区的更新块。在这种情况下, [指针区更新块的情况], 通过从左侧开始相继访问指向指针区更新块的指针列, 能识别出那些物理块, 其个数等于指针区中那些目标块的个数, 如前所述当把数据写入数据区时这些块必定被改变 (被更新)。

因此, 在上述情况中, 通过从左侧开始相继访问指向第一区中指针区更新块的指针列, 能识别出指针区的两个更新块 # 1BH 和 # 1CH (它们的个数等于物理块 # 1AH 的两个物理块和个数, 物理块 # 1AH 中有要写入数据的物理块 (数据区的目标块) 的物理块号 # 00H 和 #

02H)、指针区的物理块 # 18H 以及实际写入数据的物理块的物理块号 # 10H 和 # 11H。

例如,在识别出的指针区更新块 # 1BH 或 # 1CH 当中的更新块 # 1BH 中复制了指针区的物理块 # 18H (其中存储数据区目标块的物理块号 # 00H 和 # 02H) 中的存储内容,而这存储内容中的数据区目标块的物理块号 # 00H 和 # 02H 又分别被重写为实际写入数据的数据区更新块的物理块号 # 10H 和 # 11H (如图 13 所示)。

再有,在识别出的指针区更新块 # 1BH 或 # 1CH 当中的另一个更新块 # 1CH 中复制了指针区物理块 # 1AH (其中存储实际写入数据的物理块的物理块号 # 10H 和 # 11H) 的存储内容,而物理块号 # 10H 和 # 11H 被分别重写为本应被写入数据的物理块 (数据区目标块) 的块号 # 00H 和 # 02H (如图 13 所示)。

然后,刚才是指针区更新块 (那里的物理块号被更新) 的物理块的物理块号 # 1BH 和 # 1CH 被写入指向第二区 (它曾为无效区) 的指针区目标块的指针列。就是说,指向第一区中的指针区目标块的指针列被复制到第二区,而第一区中对应于逻辑地址 % 00H 至 % 07H 的第 1 字节被重写为该指针区物理块的物理块号 # 1BH,其对应于逻辑地址 % 10H 至 % 17H 的第 3 字节被重写为该指针区的物理块号 # 1CH (见图 13)。

结果,观察第二区,物理块 # 1BH、# 19H 和 # 1CH 变为指针区的目标块,而曾为数据区更新块的物理块 # 10H 或 # 11H 被做成分别对应于逻辑块号 % 00H 或 % 02H,曾是数据区目标块的物理块 # 00H 或 # 02H 曾被做成与这两个逻辑块号 % 00H 或 % 02H 对应。再有,曾是目标块的物理块 # 00H 或 # 02H 被分别做成对应于逻辑块 % 10H 或 % 11H,并成为更新块。

就是说,访问指向第二区的指针区目标块的指针列的第 1 字节得知,其中给出对应于逻辑块 % 00H 和 % 02H 的数据区物理块的物理块号的那个指针区物理块是物理块 # 1BH,而访问物理块 # 1BH 的第 1 字节和第 3 字节得知,对应于逻辑块 % 00H 和 % 02H 的数据区物理块

分别为物理块 # 10H 和 # 11H。访问指向第二区的指针区目标块的指针列的第 3 字节得知，其中给出对应于逻辑块 % 10H 和 % 11H 的数据区物理块的物理块号的那个指针区物理块是物理块 # 1CH，而访问物理块 # 1CH 的第 1 字节和第 2 字节得知，对应于逻辑块 % 10H 和 % 11H 的数据区物理块（即数据区的更新块）是物理块 # 00H 和 # 02H。

再有，指针区中物理块的物理块号 # 18H（该块中曾存储本应写入数据的物理块的物理块号 # 00H 和 # 02H）及物理块的物理块号 # 1AH（该块中曾存储实际写入数据的物理块的物理块号 # 10H 和 # 11H）被写入指向第二区（它曾是无效的）的指针更新块的指针列中。就是说，指向第一区中指针区更新块的指针列被复制到第二区，而块中所存内容已被更新的指针区物理块的物理块号 # 1BH 或 # 1CH 被分别重写为物理块号 # 18H（其中曾存放的物理块号对应于逻辑地址 % 00H 至 % 07H）或物理块号 # 1AH（该块中曾存放的物理块号对应于逻辑地址 % 10H 至 % 17H）（如图 13 所示）。

结果，观察第二区，物理块 # 18H、# 1AH 和 # 1DH 是指针区的更新块。

在如上述更新第二区之后，与单指针系统的情况类似，在第二区中相继写入新旧程度信息和有效性信息。

所以在这种情况中，除非在数据区和指针区的写操作都已完成，否则有效性信息不能被写入。就是说，在上述情况中，除非已完成了向两个逻辑块 % 00H 和 % 02H 的写数据操作，否则指明有效性的有效性信息不能被写入第二区。

结果，如果在向两个逻辑块 % 00H 和 % 02H 写数据期间发生了存储器讹误，只有第一区处于有效状态，通过访问有效的第一区，则从逻辑块 % 00H 和 % 02H 中读出存储在物理块 # 00H 和 # 02H（它们分别被做成对应于逻辑块 % 00H 和 % 02H）中的旧数据。

另一方面，当向两个逻辑块 % 00H 和 % 02H 的写数据操作已经完成而且第二区中的有效性信息已被更新时，通过访问有效的而且是最新的第二区，则从逻辑块 % 00H 和 % 02H 中读出存储在物理块 # 10H

和 # 11H(它们分别被做成对应于逻辑块 % 00H 和 % 02H)中的新数据。

因此,当多个数据是存储在逻辑块 % 00H 和 % 02H 中的两个相关联数据时,即使在引起存储器讹误的情况下,也能象在单指针系统的情况中那样保持这两个数据之间的兼容性。

再有,在指向第一和第二区的指针区目标块的指针列中能存储 3 个物理块的物理块号,而在指针区的一个物理块中能存储 8 个物理块的物理块号,所以最多能控制数据区的 24 (= 3×8) 个物理块。因此,最多有 12 块(这是 24 块的一半)能作为数据区的更新块,而且能对这么多块保持数据兼容性,这个数目大于单指针系统中的数目。

再有,在图 12 和图 13 中,24 块中的 8 块是更新块,因而最多能保持 8 块的数据兼容性。

这里,上文中参考图 12 和图 13 所描述的存储器读、写系统可以说是关于数据区的两级指针,所以在下文中恰当地把该系统称作双指针系统。

下面将参考图 14 和图 15 的流程图更详细地描述根据双指针系统的读、写处理过程。

首先,将参考图 14 的流程图描述读处理过程。

在这种情况下,首先在步骤 S61,完成与图 6 的步骤 S1 类似的确定处理。

当只有第一区有效时,例程进入步骤 S62,而当只有第二区有效时,例程进入步骤 S65。

在步骤 S62,通过访问指向第一区中指针区目标块的指针列,识别出指针区一个物理块(其中存有对应于要读数据逻辑块号的物理块号)的物理块号,然后例程进入步骤 S63。就是说,例如当存储器有参考图 11 所描述的结构时,在逻辑块号落入 % 00H 至 % 07H 或 % 08H 至 % 0FH 区间的情况中,存储在指向指针区目标块的指针列的第一或第二字节中的物理块号被分别识别出来。

在步骤 S63,通过访问对应于步骤 S62 中识别出的物理块号的那个指针区物理块,对应于要读数据的逻辑块号的物理块号被识别出来,

于是例程进入步骤 S64。就是说，当要读数据的逻辑块号是以某一参考值（在图 12 的情况中是 %00H 或 %08H）为基值加上 0 至 7 的偏移值所得到的值，从与步骤 S63 中识别出的物理块号相对应的物理块的左侧按这个偏移值移动得到一个位置，在这个位置中存储的物理块号被识别出来作为对应于要读数据逻辑块号的物理块号。

在步骤 S64，从对应于步骤 S63 中识别出的物理块号的那个物理块中读出数据，然后过程终止。

另一方面，在步骤 S65，通过访问指向第二区中指针区目标块的指针列，识别出指针区一个物理块（其中存有对应于要读数据逻辑块号的物理块号）的物理块号，然后例程依次进入步骤 S66 和 S67，那里完成的处理分别对应于步骤 S63 和 S64，然后处理过程终止。

接下来将参考图 15 的流程图来描述写处理过程。

在这种情况下，在步骤 S71，也是完成与图 7 的步骤 S11 类似的确处理。

当第一区是有效的，例程进入步骤 S72，并确定第二区中的数据是否被擦除。当确定它未被擦除时，例程进入步骤 S88，而当确定它已被擦除时，例程进入步骤 S73。在步骤 S88，第二区（它是无效块）中的数据被擦除，例程进入步骤 S73。在步骤 S73，通过访问指向第一区中指针区目标块的指针列，识别出指针区一个物理块（其中曾存入物理块号）的物理块号，例程进入步骤 S74。就是说，例如，当存储器有图 12 所示结构时，指向第一区中指针区目标块的指针列的第 3 字节中存储的物理块号 #1AH 被识别出来。

在步骤 S74，通过访问数据区更新块的物理块号（它存储在由步骤 S73 识别出的物理块号所对应的物理块中），数据区更新块被识别出来，然后例程进入步骤 S75，于是由外部或其他类似方式提供的要写入数据被写到数据区的更新块。

再有，当从外部提供的数据要写入多个逻辑块时，在步骤 S74，通过相继查寻（例如从左到右查询）对应于步骤 S73 中识别出的物理块号的那些物理块（在指针区中的物理块），识别出了数据区中更新块

的物理块号，这些更新块的个数等于多个逻辑块的个数。再有，在这种情况下，在步骤 S75，从外部提供的数据被相继写入数据区的更新块，这些更新块分别对应于这样识别出的多个物理块号。

然后，例程进入步骤 S76，在其中存有物理块号并曾被更动（这些物理块号被做成对应于要被写入数据的逻辑块的逻辑块号）的物理块被复制到指针区的更新块。就是说，例如参考图 12 和图 13 来描述这一点，那么存储物理块号 # 00H 和 # 02H（它们被做成与写入数据用的逻辑块的逻辑块号 % 00H 和 % 02H 相对应）的物理块 # 18H 被更新，物理块号 # 00H 和 # 02H（它们已被变成 # 10H 和 # 11H）被复制到指针区的更新块 # 1BH。

再有，在步骤 S76，存储数据区更新块指针（数据区更新块的物理块号）的物理块（它们已被更新）被复制到指针区的更新块。例如参考图 12 和图 13 来描述这一点，那么存储数据区更新块的物理块号 # 10H 至 # 17H 的物理块 # 1AH 被更新，而且物理块号 # 10H 和 # 11H 分别被改变成 # 00H 和 # 02H，这被复制到指针区的更新块 # 1C。

接下来，例程进入步骤 S77，指向第一区中指针区目标块的指针列和指向第一区中指针区更新块的指针列被更新，而且被更新的新旧程度信息和有效性信息被复制到第二区。

这是以图 12 和图 13 的情况为例，关于指向第一区中指针区目标块的指针列，在第 1 字节中存储的含有指向数据区物理块 # 00H 和 # 02H（它们被做成对应于写外部提供数据的逻辑块）的指针的物理块号 # 18H 被更新为含有指向数据区物理块 # 10H 和 # 11H（这是数据曾实际写入区）的指针的物理块号 # 1BH，而含有指向数据区物理块 # 10H 和 # 11H（这是数据曾实际写入区）的指针的物理块号 # 1AH 被更新为含有指向数据区的物理块 # 00H 和 # 02H（它们被做成对应于数据写入的逻辑块）的指针的物理块号 # 1CH。

关于指向第一区的指针区更新块的指针列，作为更新块的指针区物理块（其中的物理块号曾被更新）的物理块号 # 1BH 和 # 1CH 分别被更新为已成为指针区更新块的物理块的物理块号 # 18H 和 # 1AH。

再有，在第一区中的新旧程度信息和有效性信息被更新，前述更新结果被写入第二区，然后例程进入步骤 S78。在步骤 S78，在第一区中的所有数据被擦除。

另一方面，在步骤 S71，当确定第二区是有效区时，在步骤 S79 至 S85 以及 S90 完成与步骤 S72 至 S78 以及步骤 S88 相同的处理，只是对第一或第二区完成的处理反过来分别变成对第二或第一区的处理。

当发生存储器讹误时，在步骤 S71，确定第一和第二区的有效性。在这种情况下，例程进入步骤 S86，并确定第一和第二区的块的新旧程度。在步骤 S86，当确定第二比第一区新时，例程进入步骤 S89，指向第一区中指针区的指针被更新到第二区的一个新块，被更新的新旧程度和有效性信息被重写，然后例程进入步骤 S90。接下来，在步骤 S90 和 S80 至 S85，完成如前一情况中相同的处理。

当在步骤 S86 中确定第一区比第二区新时，在步骤 S87 以及步骤 S88、S73 和 S78 中完成与步骤 S89、S90、S80 及 S85 相同的处理，只是对第一或第二区完成的处理反过来分别变成对第二或第一区的处理。

在双指针系统的情况中，如同在单指针系统的情况中一样，能保持兼容性的物理块的个数是受限制的，然而，如果与单指针系统扩展成双指针系统的情况相类似地扩展双指针系统，而对数据区的指针含有 3 级、4 级、...，则能够保持兼容性的物理块的个数能进一步增加。因此，通过考虑一块的长度和恰当地设置数据区指针的级数，可以把能保持兼容性的物理块个数的上限安排成一个所希望的值。

实际上，如图 3 所示由 IC 卡 2 的序列发生器 91 对 EEPROM 66 MUCK 进行数据读、写是根据上述双指针系统来实现的，在这种情况下对 EEPROM63 的存储器进行分配的举例示于图 16。

在图 16 中，对 EEPROM 66 提供了 256 个物理块。每个物理块含有总数 40 个字节，其中 32 个数据单元字节 (D00 至 D1f) 和 8 个奇偶检验单元字节 (P0 至 P5)。

再有，在图 16 中，物理块号 # 00H 至 # FFH 被分别分配给这 256 个物理块，其中物理块 # 00H 至 # EFH 构成数据区，物理块 # F0H 至 # FDH 构成指针区，而物理块 # FEH 和 # FFH 构成分别构成第一和第二区。

再有，关于数据区，在 240 个物理块 # 00H 至 # FEH 中，224 个物理块是目标块（下文中也称作数据块），而其余的 16 个物理块是更新块。因此，最多能保持 16 块的数据兼容性。

再有，在图 16 所示状态中，物理块 # 00H 至 # DFH 是数据块，而物理块 # E0H 至 # FEH 是更新块。

关于指针区，14 个物理块 # F0H 至 # FDH 中 7 个是目标块（下文中称作指针块），这 14 个物理块中另 7 个是更新块。再有，在图 16 所示状态中，物理块 # F0H 至 # F6H 是指针块，物理块 # F7H 至 # FDH 是更新块。

再有，在图 16 中，第一和第二区被图示为有效性确定块。

图 17 显示出有效性确定块（第一和第二区）的结构举例。如图 17A 所示，新旧程度信息（确定新旧程度信息）被安排在有效性块的数据单元的第 1 字节（D00）中，指针区目标块（指针块）的物理块号被安排在第 2 至第 8 字节（D01 至 D07）。

在数据区中的更新块的物理块号被安排在有效性确定块的数据区的第 9 至第 24 字节（D08 至 D17）。就是说在上述情况中，在第一和第二区中，存储的是存储数据区更新块的物理块号的指针区物理块的物理块号（指向数据区更新块的指针的指针），然而在这一情况中，数据区更新块的物理块号（指向数据区更新块的指针）被直接安排在有效性确定块。

因此，在这一情况中，在指针区中没有物理块用于存储数据区更新块的物理块号。

指针区更新块的物理块号被安排在有效性确定块的数据单元的第 25 至第 31 字节（D18 至 D1e），而其第 32 字节（D1f）被放在一边作为保留字节。

再有，对应于上述有效性确定信息的 RS 代码或类似代码被安排在有效性确定块的奇偶检验单元中。

这里，当 EEPROM 66 有图 16 所示结构时，图 17 显示出此时有效性确定块中数据部分的典型初始值。

接下来，图 18 显示组成指针区的物理块（指针块和更新块）的结构举例。

如图 18A 所示，数据区物理块的物理块号被安排在数据部分中的 32 个单元（D00 - D1f）中。构成指针区的物理块的奇偶检验部分没有被用作前述有效性信息，而是作为通常的奇偶检验值，因此，最好向奇偶检验部分中安排的信息提供纠错能力。这是因为在双指针系统中，“在组成指针区的物理块的数据部分中安排的信息中是否有错”这一判定的有效性最终不是由安排在奇偶检验部分中的信息来确定，而是根据作为有效性确定块中的有效性信息的奇偶检验部分来判定。

这里，当 EEPROM 66 有图 16 所示结构时，图 18B 显示出此时构成指针区的物理块的数据部分的典型初始值。

再有，构成数据区的物理块（数据区的数据块和更新块）与构成指针区的物理块类似，只是其中安排的不是物理块指针，而是供读、写的的数据，所以略去对它们的描述。

如前所述，在向数据区写入之后和向指针区（那里存储构成数据区的物理块的指针）写入之后，有效性确定性（第一或第二区，用于存储指向构成指针区的物理块的指针）的存储内容被更新，所以能解决在一个或多个物理块中的物理存储器讹误问题。就是说，不仅能解决组成数据区的一个物理块的存储器讹误问题，而且还能解决多个物理块的存储器讹误问题。

再有，如前所述，有效性确定块和存储在指针区的信息作为一个转换表用于把数据区中的逻辑块号转换成物理块号，因此，当关于有效性确定块或指针区的写例程进行过程中发生故障时，虽然引起了逻辑存储器讹误，但这种逻辑存储器讹误能被解决。

结果，即使在写多个相关联数据时发生故障而且存储器讹误，该

存储器讹误能被有效地解决，保持多个数据的兼容性。

如前所述，虽然所给出的描述是针对把本发明应用于非接触卡系统的情况，但本发明可应用于存储器之类或任何数据写入装置的存储单元。然而，本发明对这样的系统特别有用，即在这种系统中进行的数据发送与接收是在所谓不稳定状态中进行的，例如在上述非接触卡系统或者甚至是接触型系统，其中用户能自由地插入或拔出一块 IC 卡或类似装置。

再有，根据本发明的这一形式，虽然指针区的更新块的物理块号或数据区的更新块的物理块号被存储在作为所谓指针区或数据区的空闲区的有效性确定块或指针区，但并不一定需要存储更新块的物理块号。然而，当更新区的物理块号未被存储时，在写数据时，必须通过查寻 EEPROM66 的存储内容来检测出构成更新块的物理块，所以最好是存储更新块的物理块号，以达到高速处理。

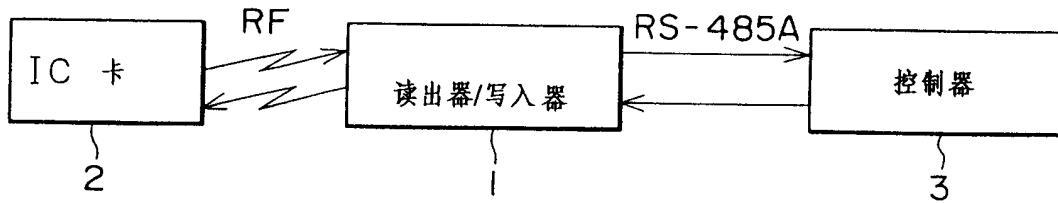
再有，根据本发明的这一形式，虽然存储有效性确定块的区、指针区以及数据区被分配在 EEPROM66 上的预定位置，但分配这些区所用的位置并不特别限定。

再有，这些区不需要是 EEPROM66 上的连续范围内的区。就是说，分别存储有效性确定块、指针区或数据区的各区能被分配在 EEPROM66 上的非连续位置。

再有，根据本发明的这一形式，在 EEPROM66 中只保留立即前一个场合的存储内容，否则，可保留立即前一个场合和在它之前的场合的存储内容。然而，在这种情况下需要更大的存储容量。

将会理解，在本发明的范围和精神内能预期各种修改和应用，所以，本发明不限于这里所描述的发明形式。

图 1



← 11 字节 (1 块) →

图 4 A

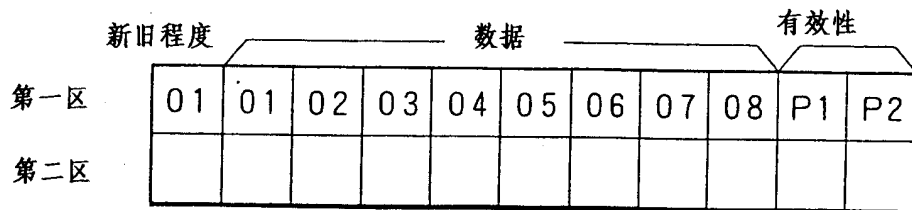


图 4 B

↓ 写数据 (11, 12, 13, 14, 15, 16, 17, 18)

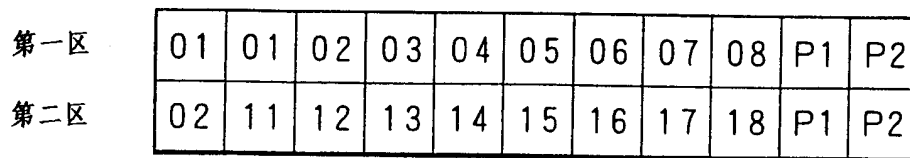
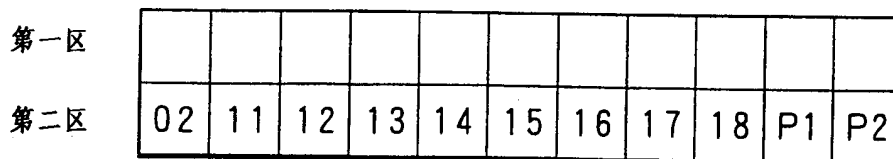


图 4 C

↓ 抹掉第一区



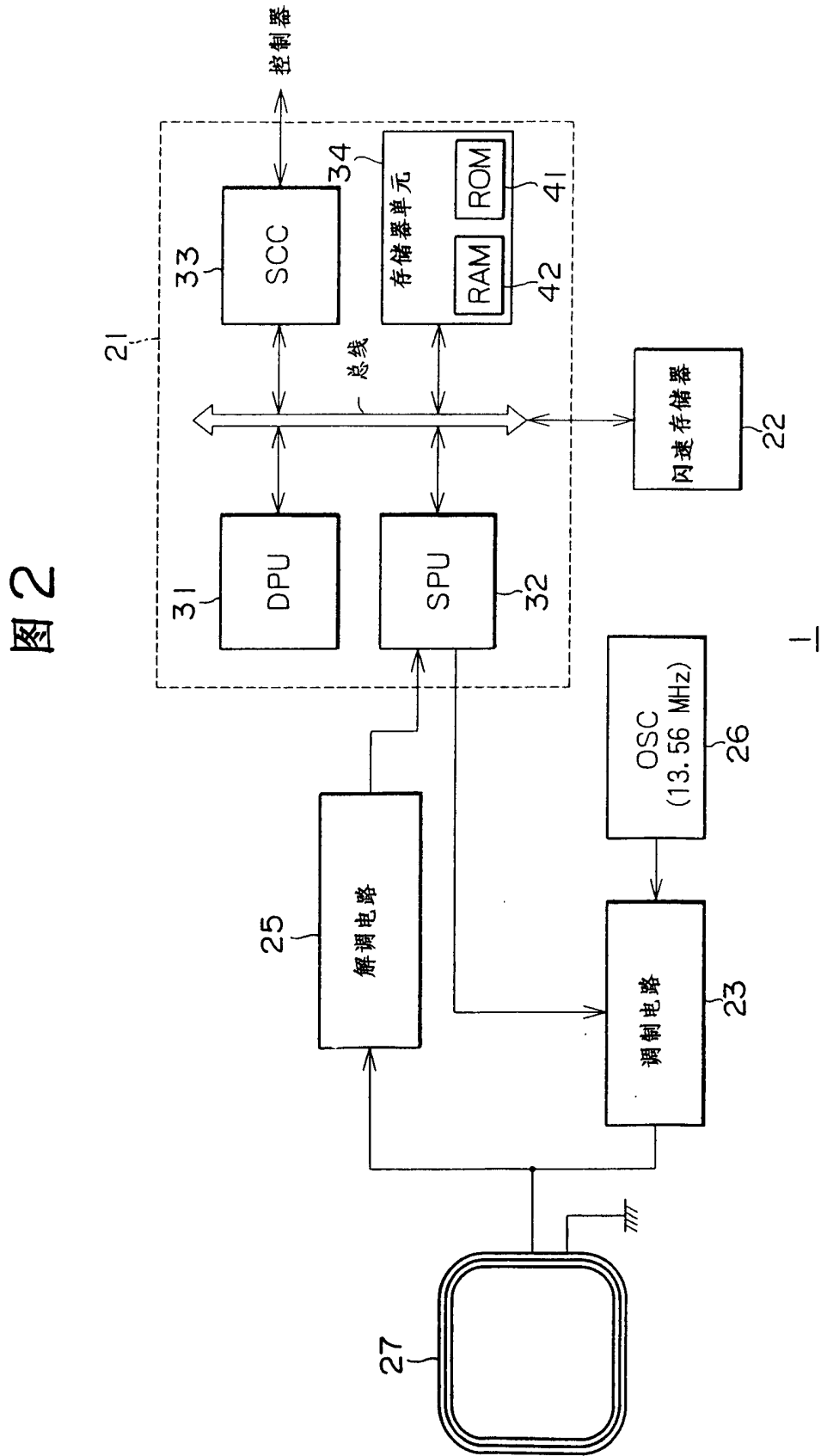
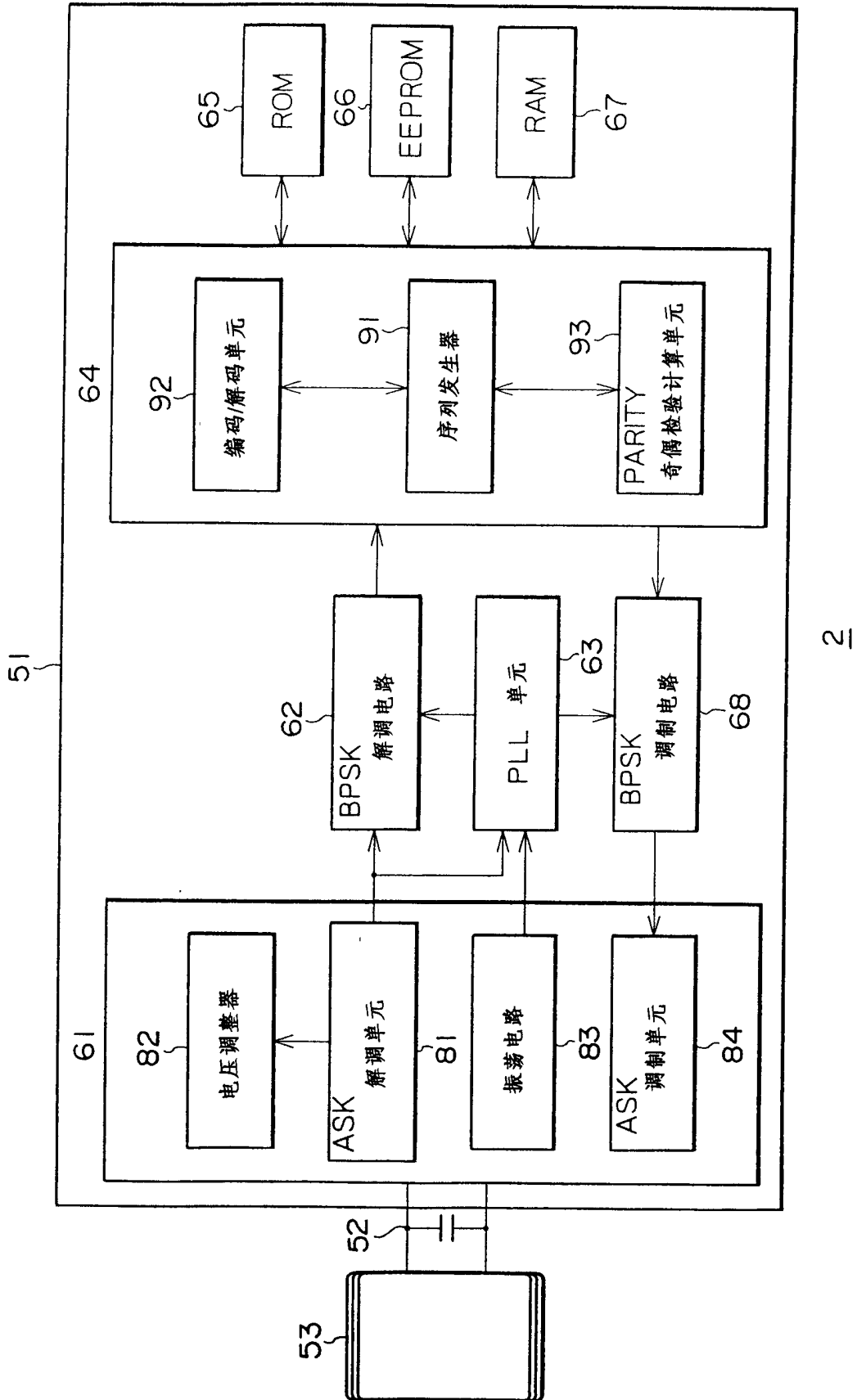
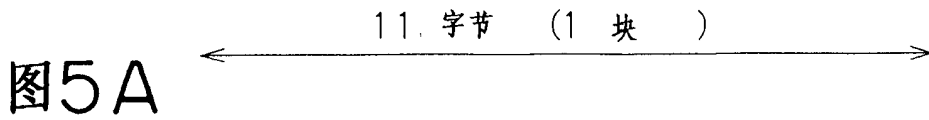
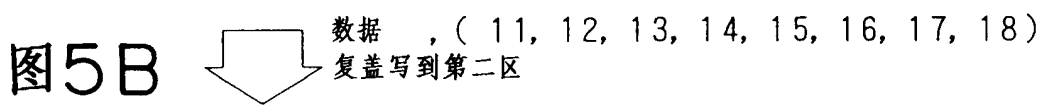


图 3





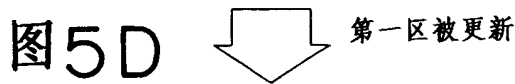
	新旧程度		数据						有效性		
第一区	01	01	02	03	04	05	06	07	08	P1	P2
第二区	02									P1	P2



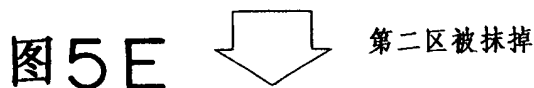
第一区	01	01	02	03	04	05	06	07	08	P1	P2
第二区	00	11	12	13	14	15	16	17	18	P1	P2



第一区											
第二区	00	11	12	13	14	15	16	17	18	P1	P2



第一区	01	11	12	13	14	15	16	17	18	P1	P2
第二区	00	11	12	13	14	15	16	17	18	P1	P2



第一区	01	11	12	13	14	15	16	17	18	P1	P2
第二区											

图 6

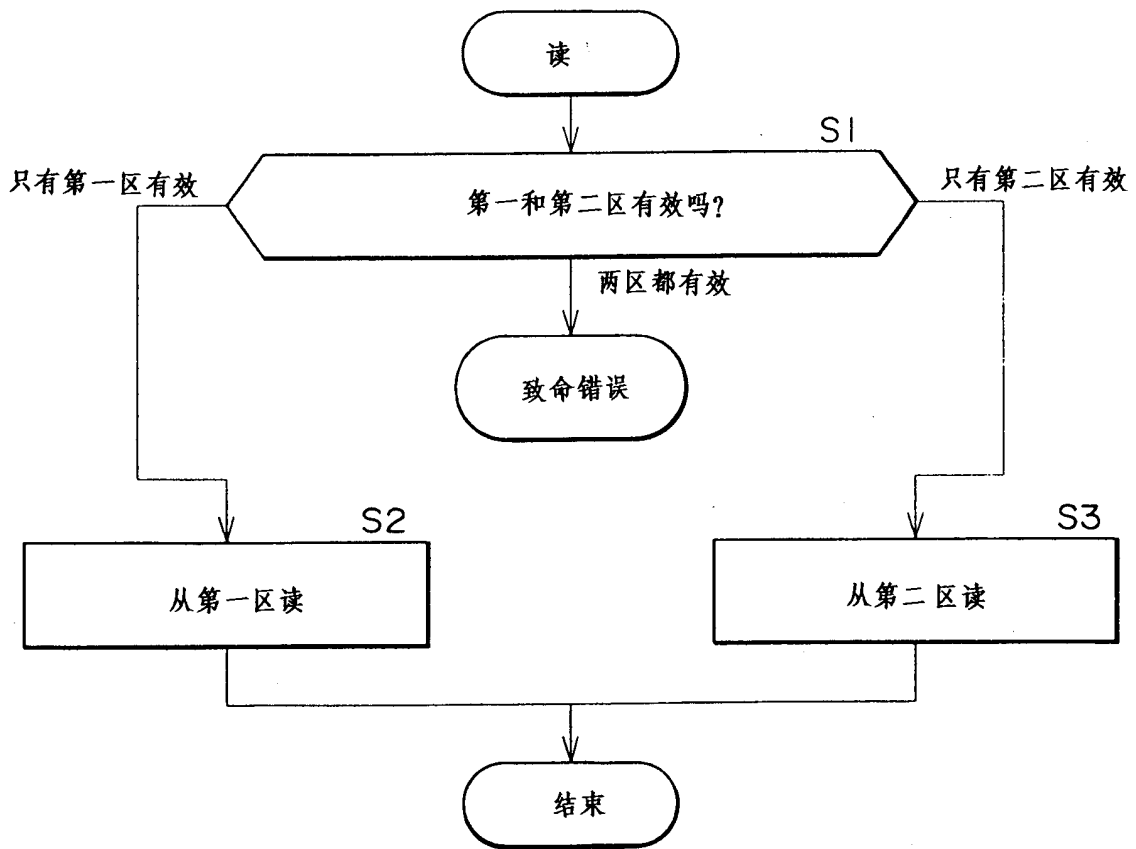


图7

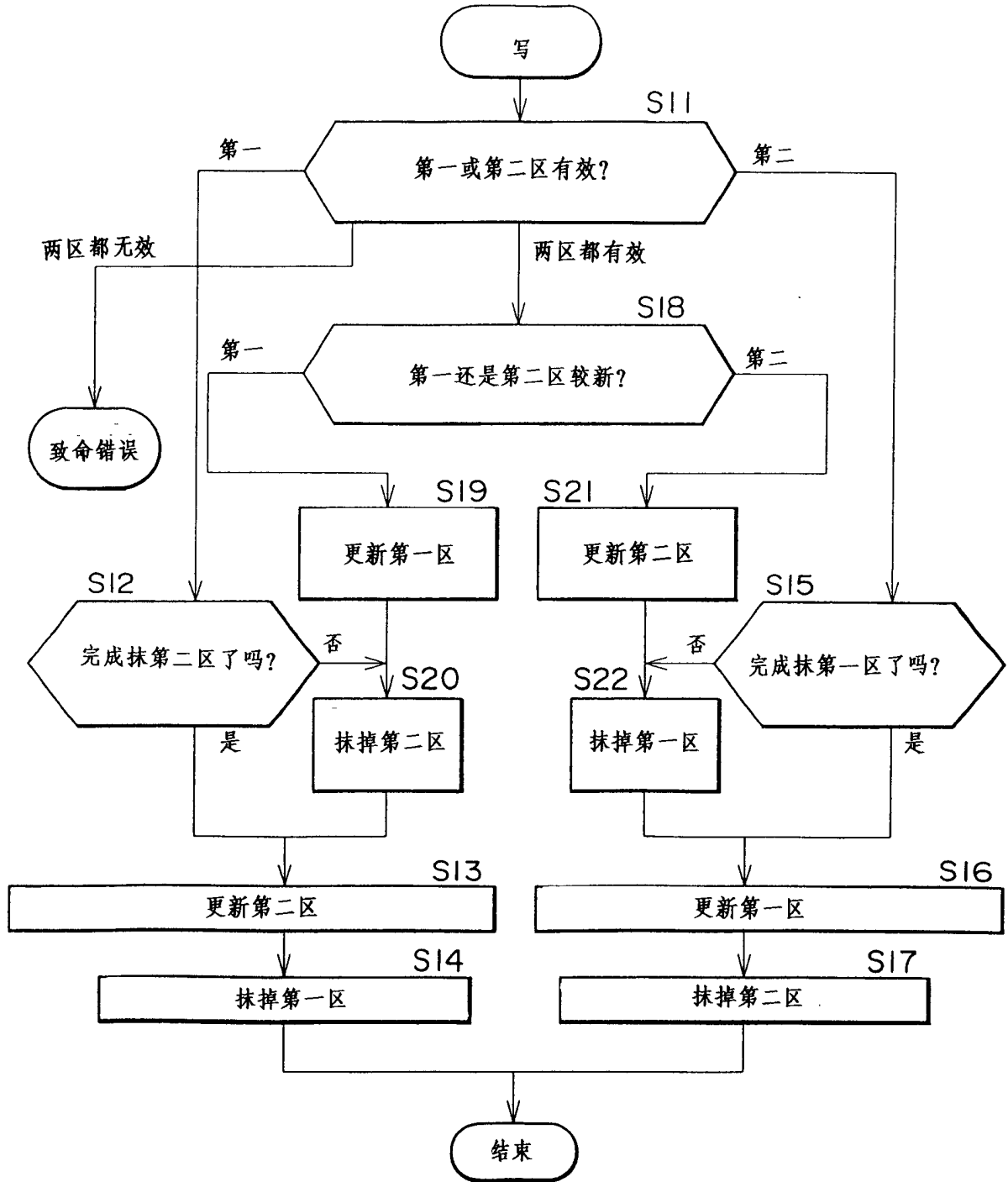


图 8

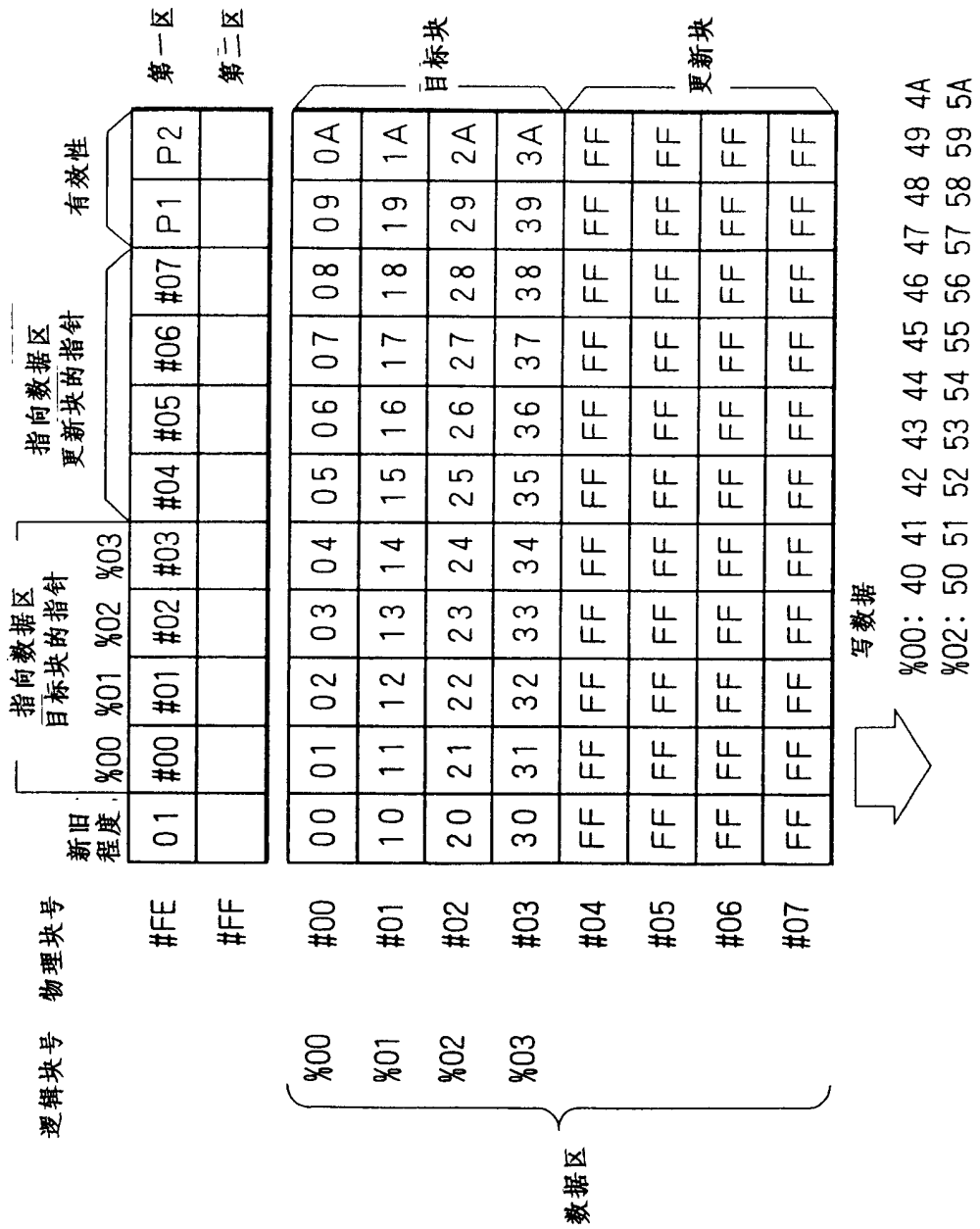


图 9

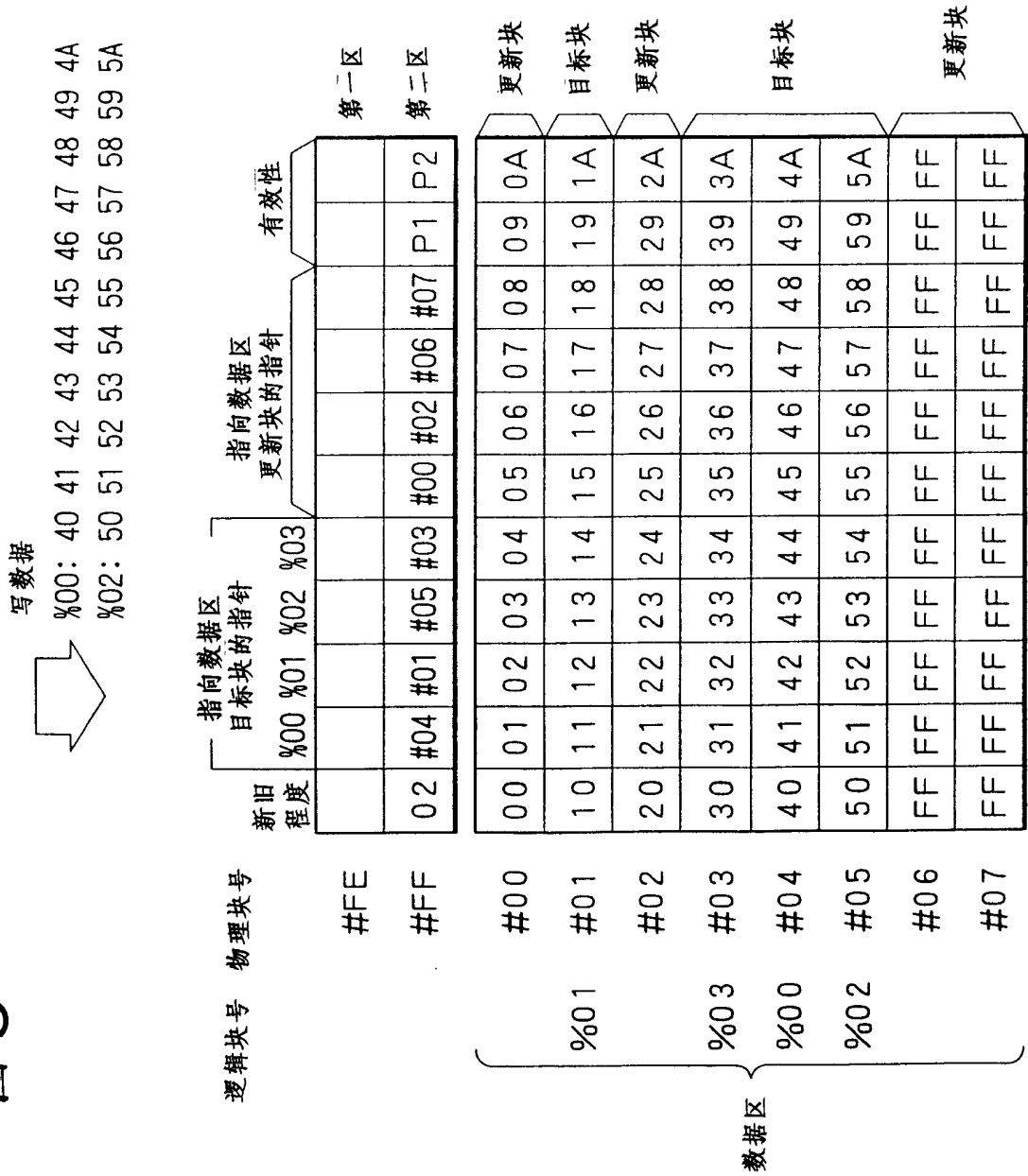


图 10

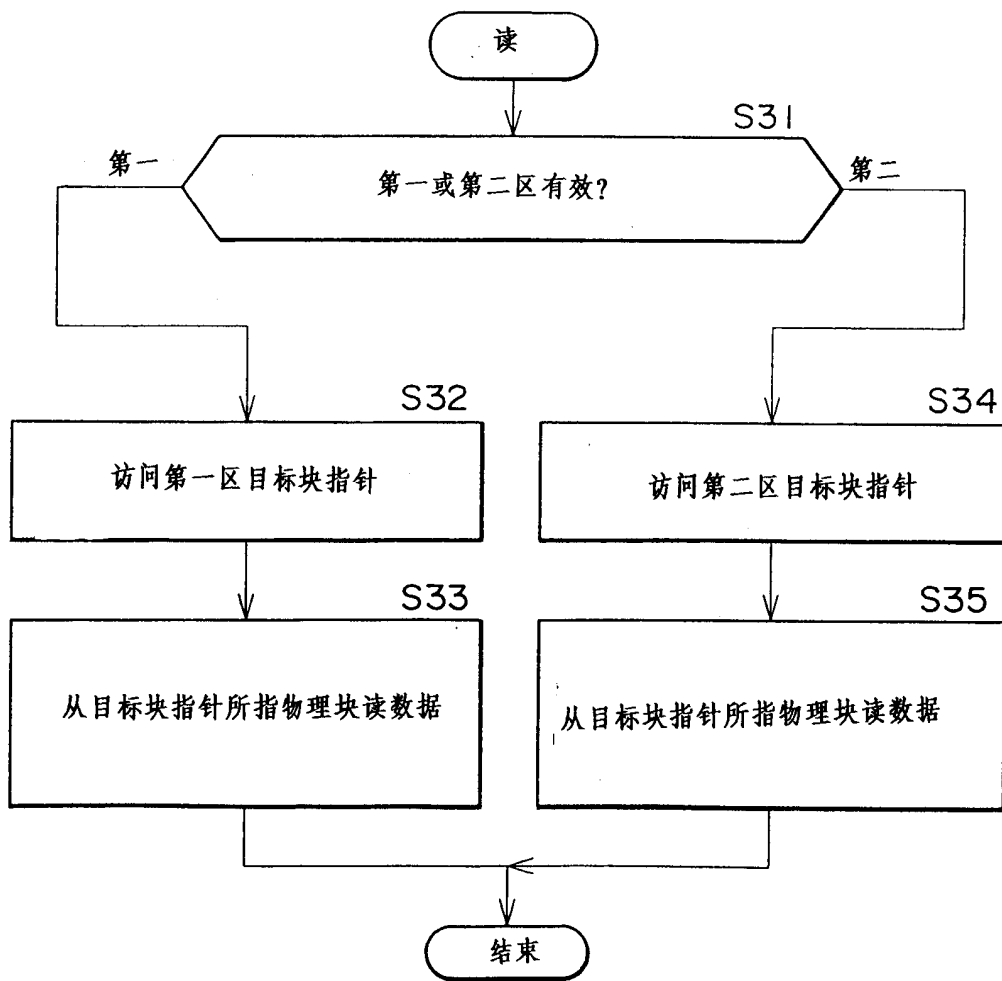


图 11

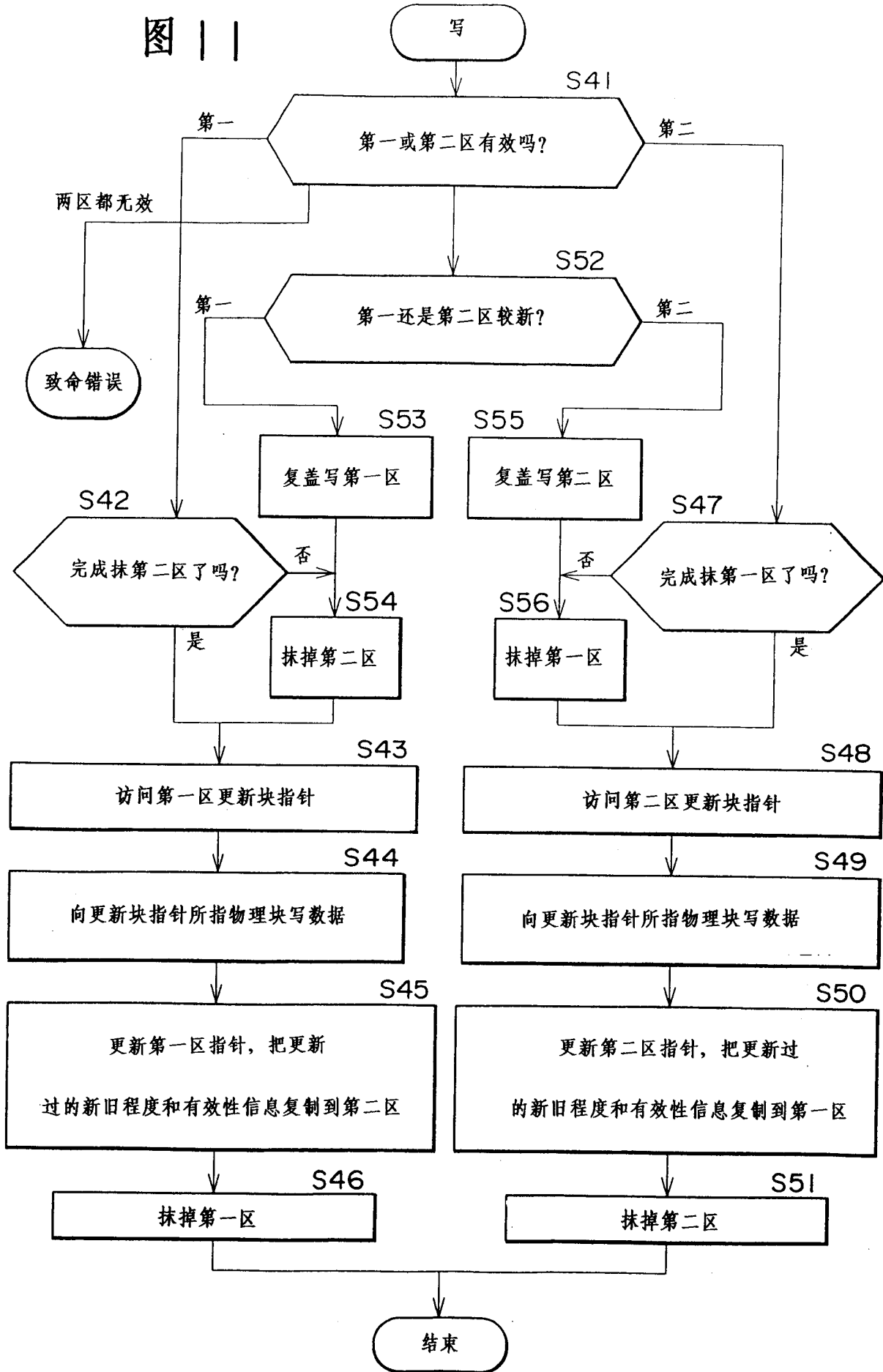


图 12

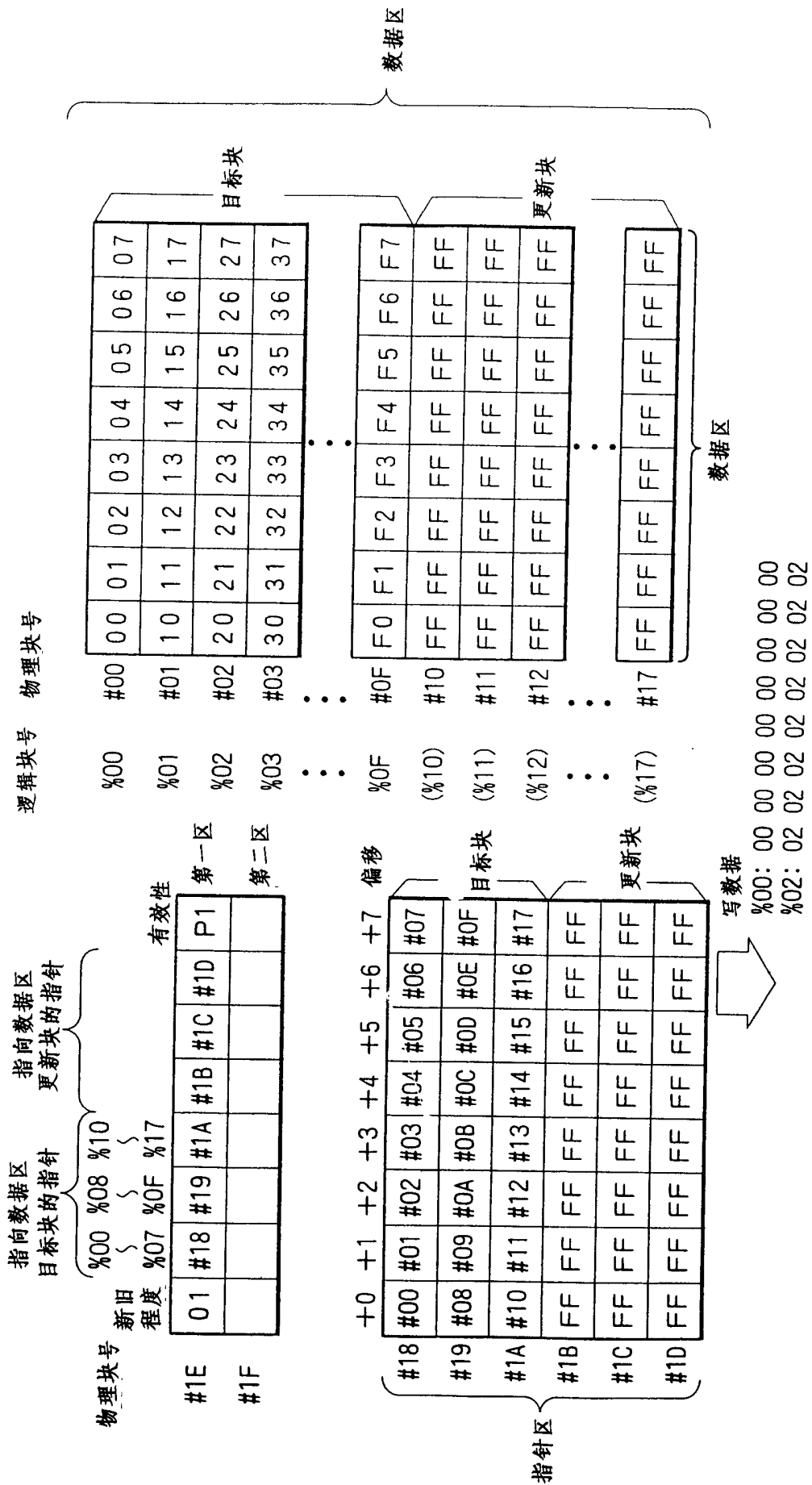


图 13

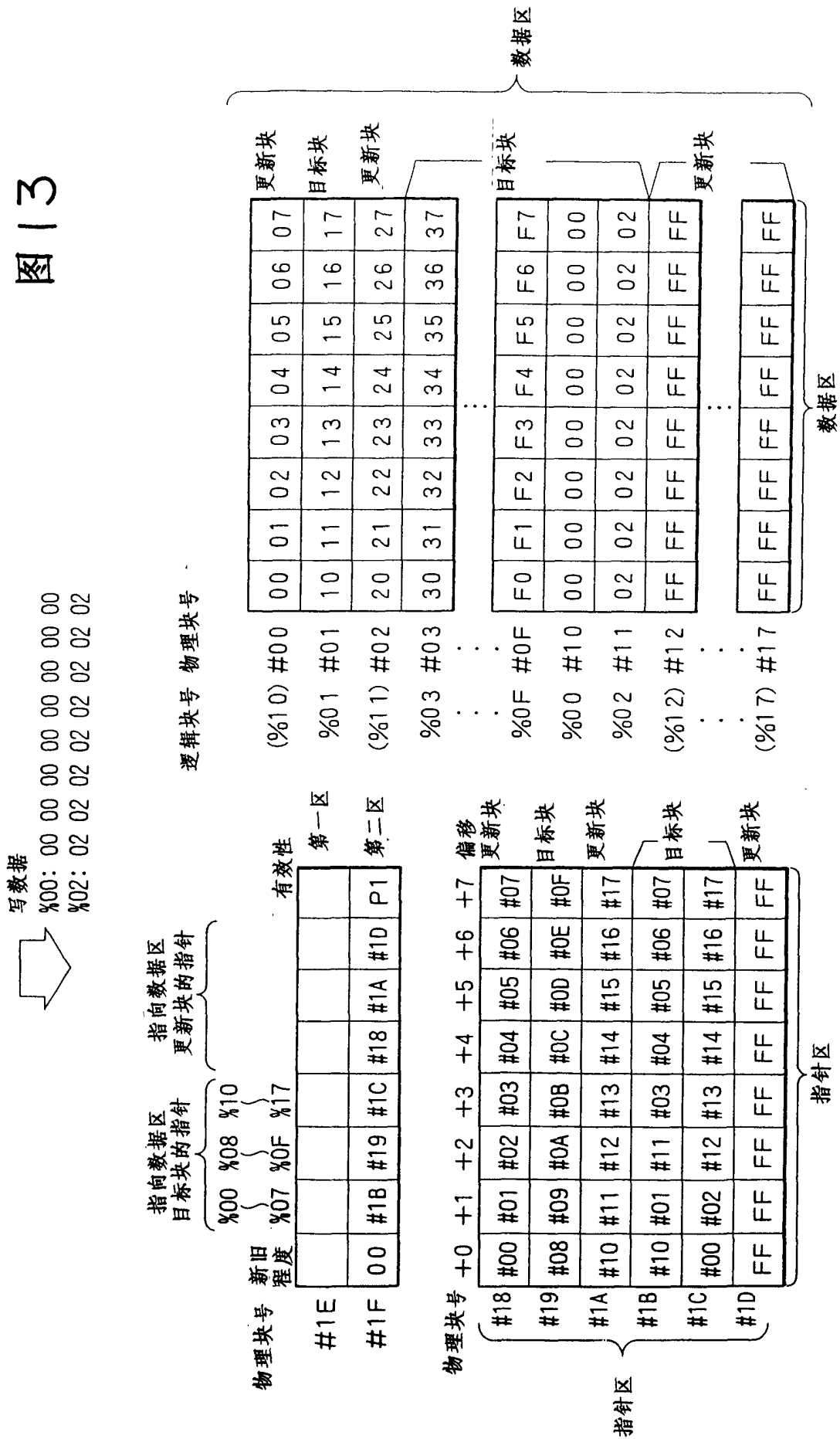


图14

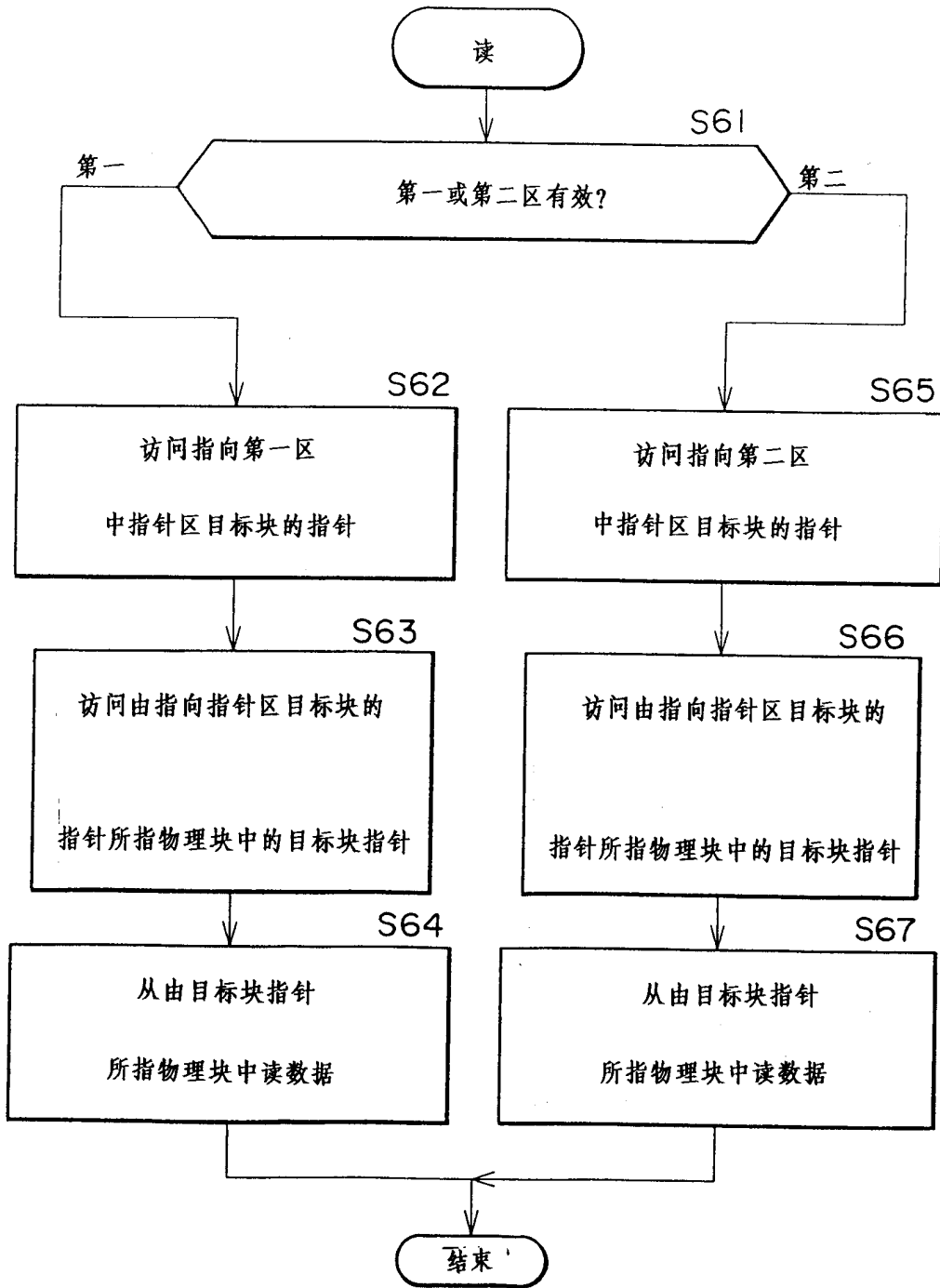


图15A

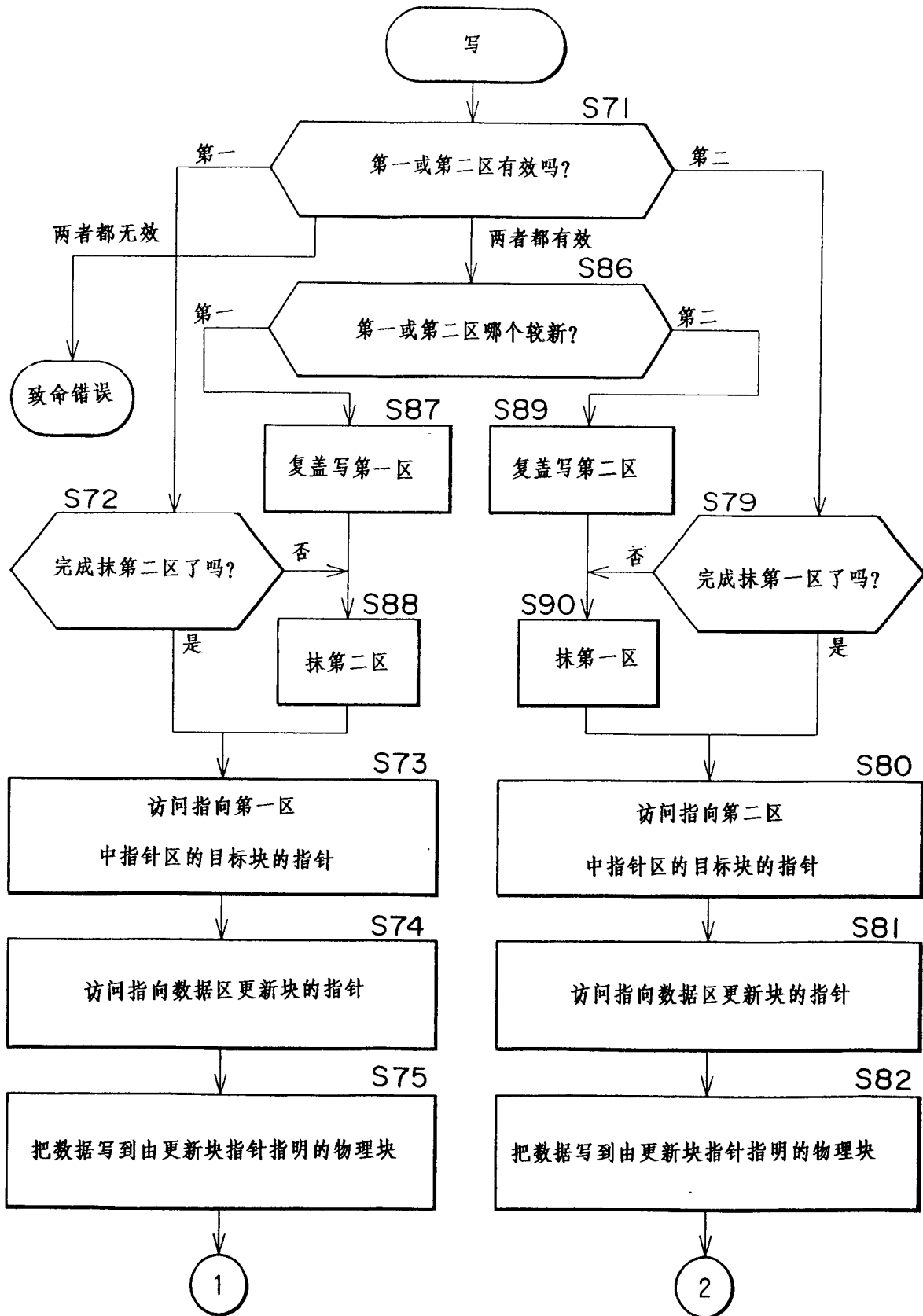


图15B

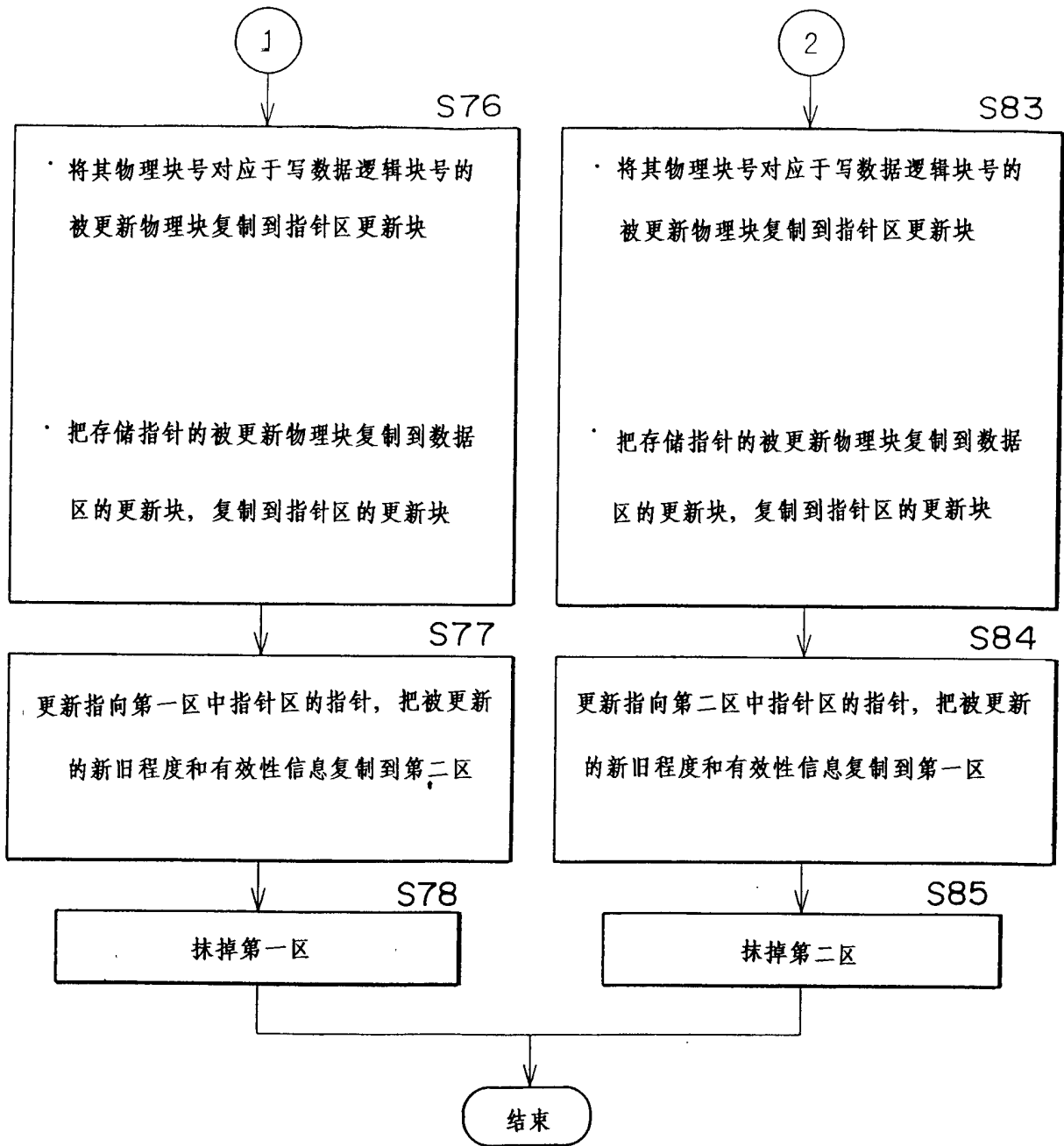


图15

图15A

图15B

图 16

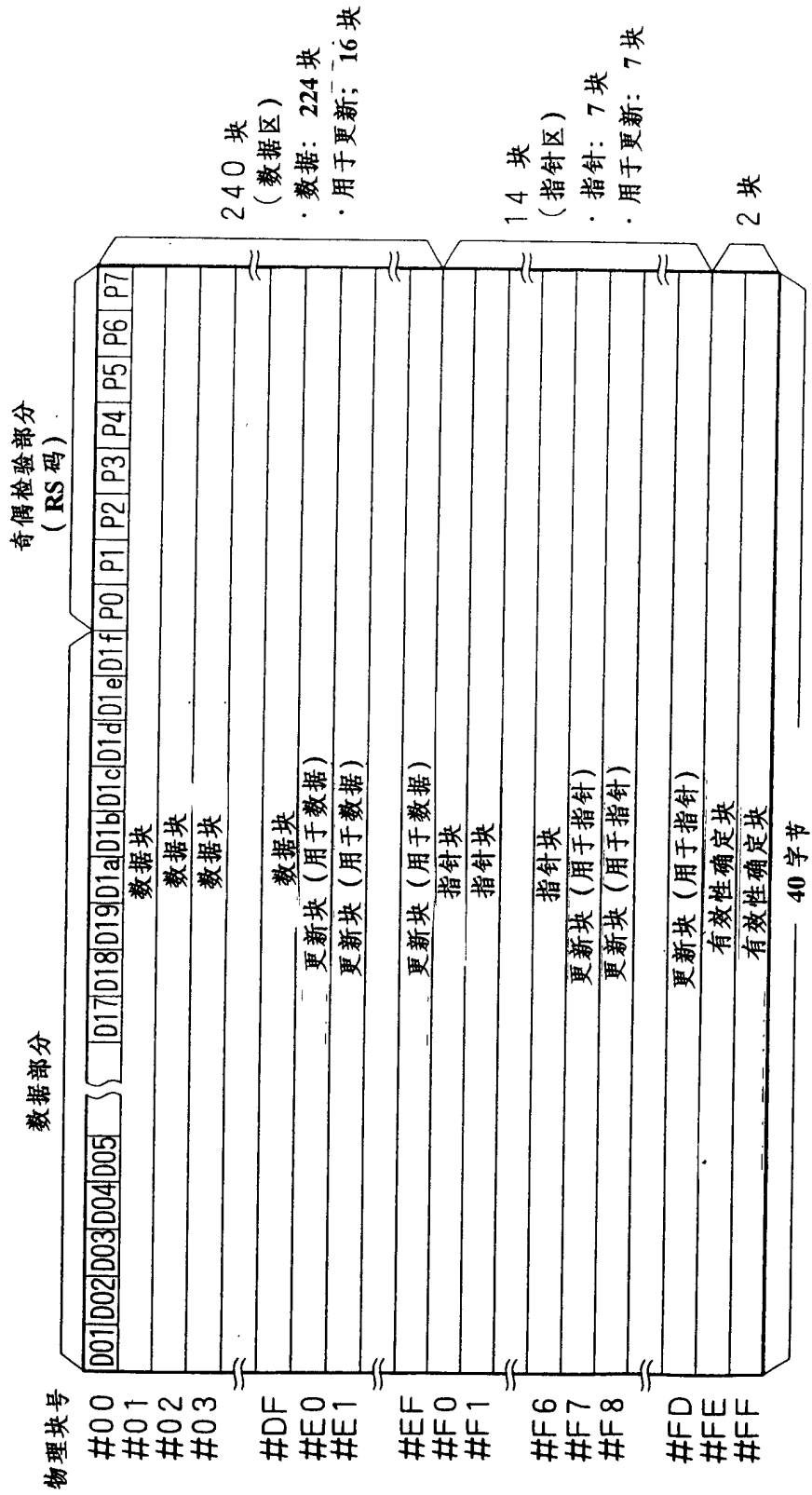


图 17A

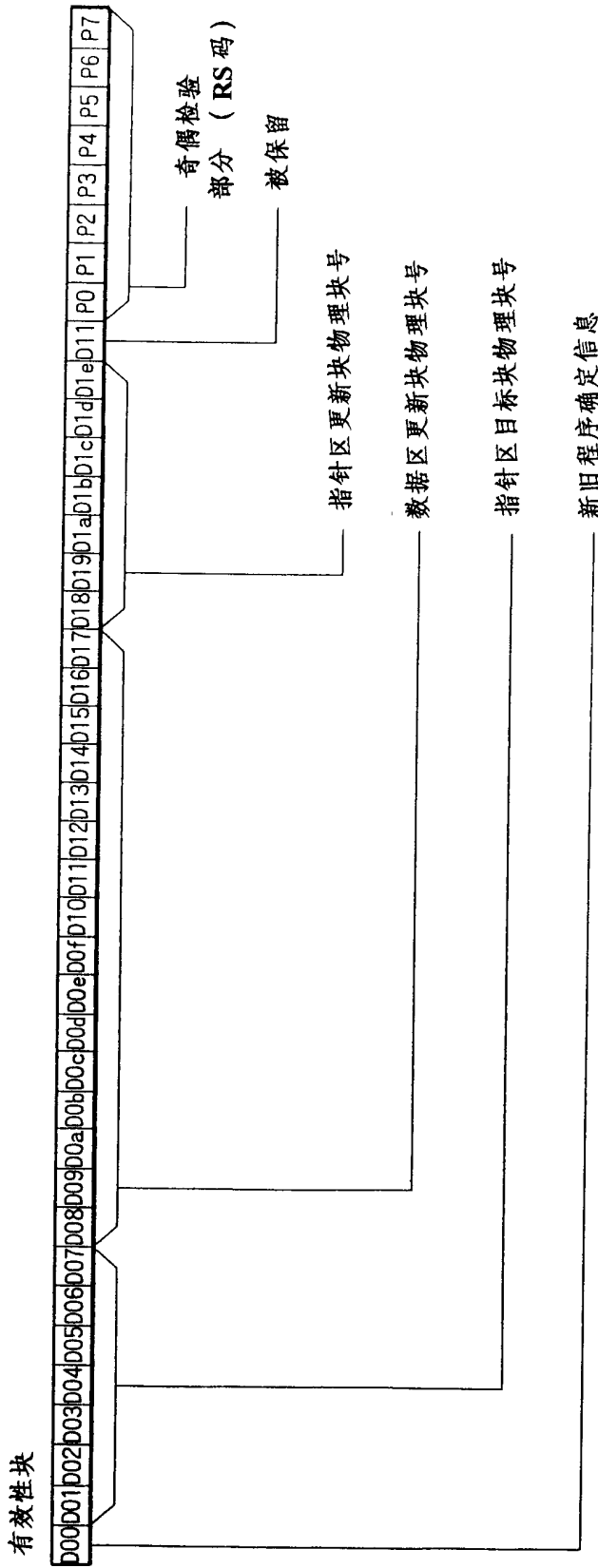


图 17B

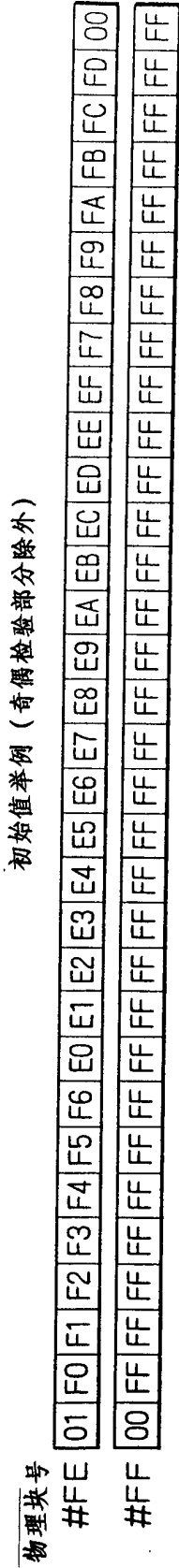


图18A

指针块更新块, 指针

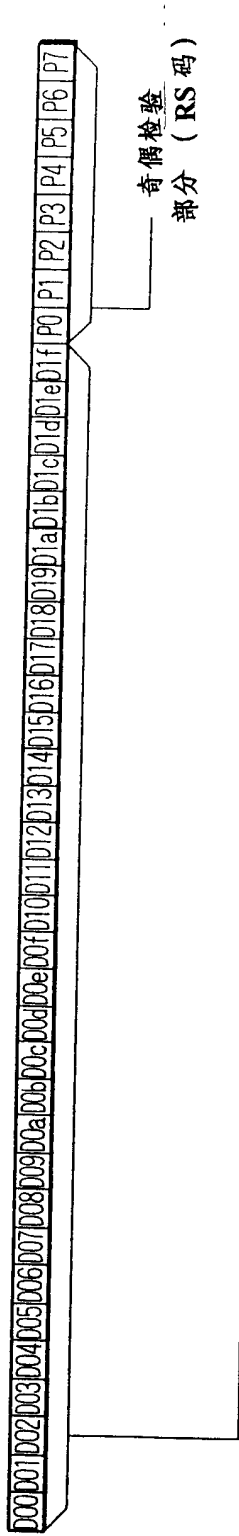


图18B

初始值举例 (奇偶检验部分除外)

物理块号	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
#F0	20	21	22	23	24	25	26	27	28	29	2A	2B	2C	2D	2E	2F	30	31	32	33	34	35	36	37	38	39	3A	3B	3C	3D	3E	3F	
...																																	
#F7	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	CA	CB	CC	CD	CE	CF	D0	D1	D2	D3	D4	D5	D6	D7	D8	D9	DA	DB	DC	DD	DE	DF	
#F8	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
#F9	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	
...																																	
#FD	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	