[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 21/00 (2006.01)

[12] 发明专利申请公布说明书

[21] 申请号 200580048239.5

[43] 公开日 2008 年 2 月 6 日

[11] 公开号 CN 101120349A

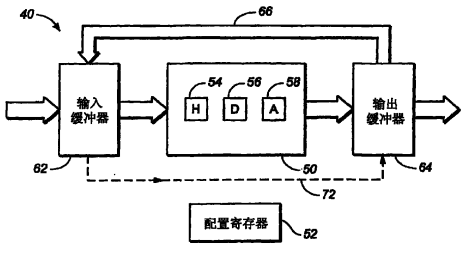
[22] 申请日 2005.12.21
[21] 申请号 200580048239.5
[30] 优先权
[32] 2004.12.21 [33] US [31] 60/639,442
[32] 2005.12.20 [33] US [31] 11/314,030
[32] 2005.12.20 [33] US [31] 11/314,032
[86] 国际申请 PCT/US2005/046586 2005.12.21
[87] 国际公布 WO2006/071725 英 2006.7.6
[85] 进入国家阶段日期 2007.8.20
[71] 申请人 桑迪士克股份有限公司
地址 美国加利福尼亚州
共同申请人 迪斯克雷蒂克斯科技公司
[72] 发明人 迈克尔·霍尔茨曼
巴鲁赫·鲍里斯·科亨
戴维·戴切尔 哈加伊·巴-埃尔
阿维朗姆·耶鲁哈米
[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 7 页 说明书 10 页 附图 4 页

[54] 发明名称

带有流中数据加密/解密的存储器系统

[57] 摘要

本发明提高了存储器系统的吞吐量，其中，在不密切涉及任何控制器的情况下，由电路对数据流中的数据进行密码化处理。优选地控制数据流，以使其具有多个源中所选择的数据源和多个目的地中所选择的目的地，所有这些情况都没有涉及控制器。优选地可配置密码化电路以启用对多个页的处理、多种算法中一种或多种密码化算法的选择从而使得在不涉及控制器的情况下进行加密和/或解密，并且在多个连续阶段中对数据进行密码化处理而无需涉及控制器。对于以交错存取的方式密码化处理来自多个数据流中的数据的存储器系统来说，当会话被中断时，就可能会丢失安全配置信息，从而使得在该会话重新进行时也不可能再继续处理过程。为了保留安全配置信息，控制器优选地使得用于该会话的安全配置信息在中断之前被存储起来，以使其在结束中断之后就能够被提取。



1. 一种用于存储经加密的数据的存储器系统，包括：

非易失性快闪存储单元；

电路，对来自或流向所述单元的数据流中的数据执行密码化处理；以及

控制器，配置所述电路并控制所述单元和所述电路，以通过使用密码化算法来执行密码化处理，从而使得在配置所述电路之后，所述电路在不涉及所述控制器的情况下对所述数据流中的数据进行密码化处理。

2. 根据权利要求 1 所述的系统，其中，数据被成页地写入所述单元中或从所述单元中被成页地读取，所述电路对每个小于页的数据单位执行密码化处理，以及所述控制器配置所述电路，以使得在配置所述电路之后，所述电路在不涉及所述控制器的情况下对多个页的数据执行密码化处理。

3. 根据权利要求 1 所述的系统，其中，所述控制器配置所述电路，以使得所述数据流具有多个源中所选择的数据源和多个目的地中所选择的目的地。

4. 根据权利要求 3 所述的系统，其中，所述控制器配置所述电路，以使得所述数据流中的数据发自所述单元并流向所述控制器或主机装置。

5. 根据权利要求 3 所述的系统，其中，所述控制器配置所述电路，以使得所述数据流中的数据发自所述控制器或主机装置，并流向所述单元。

6. 根据权利要求 3 所述的系统，其中，所述控制器配置所述电路，以使得所述数据流从所述单元流向主机装置，或者从所述主机装置流向所述单元，并旁路所述电路。

7. 根据权利要求 1 所述的系统，其中，所述控制器配置所述电路，以使得所选择的密码化算法被用于所述密码化处理过程中。

8. 根据权利要求 1 所述的系统，其中，所述控制器配置所述电路，以使得在配置所述电路之后，所述电路在不涉及所述控制器的情况下在多个连续阶段中密码化处理所

述数据流中的数据。

9. 根据权利要求 8 所述的系统，其中，在配置所述电路之后，所述电路在不涉及所述控制器的情况下通过在多个连续阶段中使用多于一个的密钥来密码化处理所述数据流中的数据。

10. 根据权利要求 8 所述的系统，其中，在配置所述电路之后，所述电路在不涉及所述控制器的情况下通过在多个连续阶段中使用多于一个的密码化处理过程来密码化处理所述数据流中的数据。

11. 一种用于存储经加密的数据的存储卡，包括：

非易失性快闪存储单元；

电路，对来自或流向所述单元的数据流中的数据执行密码化处理；以及

控制器，配置所述电路并控制所述单元和所述电路，以使用密码化算法来执行密码化处理，从而使得在配置所述电路之后，所述电路在不涉及所述控制器的情况下对所述数据流中的数据进行密码化处理，其中，所述存储单元、电路和控制器被封装到一个卡中。

12. 根据权利要求 11 所述的卡，其中，数据被成页地写入所述单元中或从所述单元中被成页地读取，所述电路对每个小于页的数据单位执行密码化处理，以及所述控制器配置所述电路，以使得在配置所述电路之后，所述电路在不涉及所述控制器的情况下对多个页的数据执行密码化处理。

13. 根据权利要求 11 所述的卡，其中，所述控制器配置所述电路，以使得所述数据流具有多个源中所选择的数据源和多个目的地中所选择的目的地。

14. 根据权利要求 13 所述的卡，其中，所述控制器配置所述电路，以使得所述数据流中的数据发自所述单元，并流向所述控制器或主机装置。

15. 根据权利要求 13 所述的卡，其中，所述控制器配置所述电路，以使得所述数据流中的数据发自所述控制器或主机装置，并流向所述单元。

16. 根据权利要求 13 所述的卡，其中，所述控制器配置所述电路，以使得所述数据流从所述单元流向主机装置，或者从所述主机装置流向所述单元，并旁路所述电路。

17. 根据权利要求 11 所述的卡，其中，所述控制器配置所述电路，以使得所选择的密码化算法被用于所述加密和/或解密中。

18. 根据权利要求 11 所述的卡，其中，所述控制器配置所述电路，以使得在配置所述电路之后，所述电路在不涉及所述控制器的情况下在多个连续阶段中密码化处理所述数据流中的数据。

19. 根据权利要求 18 所述的卡，其中，在配置所述电路之后，所述电路在不涉及所述控制器的情况下通过在多个连续阶段中使用多于一个的密钥来密码化处理所述数据流中的数据。

20. 根据权利要求 18 所述的卡，其中，在配置所述电路之后，所述电路在不涉及所述控制器的情况下通过在多个连续阶段中使用多于一个的密码化处理过程来密码化处理所述数据流中的数据。

21. 一种用于存储经加密的数据的存储卡，包括：

非易失性存储单元；

电路，对来自或流向所述单元的数据流中的数据执行密码化处理；以及

控制器，使数据被成页地写入所述单元中或从所述单元中被成页地读取，其中，所述电路对每个都小于页的数据单位都执行密码化处理，其中，在不涉及所述控制器的情况下，密码化处理并写入或读取具有多个源中的所选择的数据源和多个目的地中所选择的目的地的所述数据流的一页或多页。

22. 一种用于加密和/或解密存储器系统中非易失性快闪存储单元里的数据的方法，所述存储器系统具有控制所述单元和密码化电路的控制器，所述方法包括：

使用所述控制器来配置用于通过利用密码化算法以对来自或流向所述单元的数据流中的数据执行密码化处理的所述电路；以及

使得在配置所述电路之后，在不涉及所述控制器的情况下，所述电路密码化处理

所述数据流中的数据。

23. 根据权利要求 22 所述的方法，其中，数据被成页地写入所述单元中或从所述单元中被成页地读取，所述电路对每个小于页的数据单位都执行密码化处理，并且所述的使用过程使用所述控制器来配置所述电路，以使得在配置所述电路之后，在不涉及所述控制器的情况下，所述电路对多个页的数据执行密码化处理。

24. 根据权利要求 22 所述的方法，其中，所述的使用过程使用所述控制器来配置所述电路，以使得所述数据流具有多个源中所选择的数据源和多个目的地中所选择的目的地。

25. 根据权利要求 24 所述的方法，其中，所述的使用过程使用所述控制器来配置所述电路，以使得所述数据流中的数据发自所述单元并流向所述控制器或主机装置。

26. 根据权利要求 24 所述的方法，其中，所述的使用过程使用所述控制器来配置所述电路，以使得所述数据流中的数据发自所述控制器或主机装置并流向所述单元。

27. 根据权利要求 24 所述的方法，其中，所述的使用过程使用所述控制器来配置所述电路，以使得所述数据流从所述单元流向主机装置或者从所述主机装置流向所述单元，并旁路所述电路。

28. 根据权利要求 22 所述的方法，其中，所述的使用过程使用所述控制器来配置所述电路，以使得所选择的密码化算法被用于所述密码化处理过程中。

29. 根据权利要求 22 所述的方法，其中，所述的使用过程使用所述控制器来配置所述电路，以使得在配置所述电路之后，在不涉及所述控制器的情况下，所述电路在多个连续阶段中密码化处理所述数据流中的数据。

30. 根据权利要求 29 所述的方法，其中，所述的使用过程使用所述控制器来配置所述电路，以使得在配置所述电路之后，在不涉及所述控制器的情况下，所述电路通过在多个连续阶段中使用多于一个的密钥来密码化处理所述数据流中的数据。

31. 根据权利要求 29 所述的方法，其中，所述的使用过程使用所述控制器来配置所述电路，以使得在配置所述电路之后，在不涉及所述控制器的情况下，所述电路通过在多个连续阶段中使用多于一个的密码化处理过程来密码化处理所述数据流中的数据。

32. 一种用于加密和/或解密存储器系统中非易失性存储单元里的数据的方法，所述存储器系统具有控制所述单元和密码化电路的控制器，所述方法包括：

在不涉及所述控制器的情况下，所述电路对一页或多页数据执行密码化处理，其中，数据被成页地写入或被成页地读取，以及所述电路对每个小于页的数据单位执行密码化处理；以及

在不涉及所述控制器的情况下，控制所述数据流，以使所述数据流具有多个源中所选择的数据源和多个目的地中所选择的目的地。

33. 根据权利要求 32 所述的方法，进一步包括：配置所述密码化电路，其中，由所述电路的配置来启动所述密码化处理和所述控制过程。

34. 根据权利要求 33 所述的方法，其中，所述电路的配置也启动多个密码化算法中一个或多个密码化算法的选择，以使得在不涉及所述控制器的情况下，所述电路通过使用所选择的算法来密码化处理所述数据流中的数据。

35. 根据权利要求 33 所述的方法，其中，配置所述电路以使得所述数据流中的数据发自所述单元并流向所述控制器或主机装置。

36. 根据权利要求 33 所述的方法，其中，配置所述电路以使得所述数据流中的数据发自所述控制器或主机装置并流向所述单元。

37. 根据权利要求 33 所述的方法，其中，配置所述电路以使得所述数据流从所述单元流向主机装置或从所述主机装置流向所述单元，并旁路所述电路。

38. 根据权利要求 33 所述的方法，其中，配置所述电路以使得在配置所述电路之后，

不涉及所述控制器的情况下，所述电路在多个连续阶段中密码化处理所述数据流中的数据。

39. 根据权利要求 38 所述的方法，其中，配置所述电路以使得在配置所述电路之后，不涉及所述控制器的情况下，所述电路通过在多个连续阶段中使用多于一个的密钥来密码化处理所述数据流中的数据。

40. 根据权利要求 38 所述的方法，其中，配置所述电路以使得在配置所述电路之后，不涉及所述控制器的情况下，所述电路通过在多个连续阶段中使用多于一个的密码化处理过程来密码化处理所述数据流中的数据。

41. 一种用于存储经加密的数据的存储器系统，包括：

非易失性存储单元；

电路，对来自或流向所述单元的多于一个数据流中的数据执行密码化处理；以及

控制器，控制所述单元和所述电路，从而以交错方式来密码化处理不同数据流中的数据，以及其中，至少一个用于从所述单元存取数据的会话被其它的会话所中断，其中，所述控制器使得用于所述会话的安全配置信息在所述中断之前被存储起来，以便在结束所述中断之后所述安全配置信息能够被提取。

42. 根据权利要求 41 所述的系统，其中，所述安全配置信息包括：与数据的源或目的地、密码化钥匙、密码化算法、和/或消息认证代码相关的信息。

43. 根据权利要求 41 所述的系统，其中，当所述会话恢复时，所述控制器就使得被存储用于所述会话的安全配置信息被提取。

44. 根据权利要求 41 所述的系统，其中，所述控制器使得用于所述多于一个数据流之中的每一个数据流的安全配置信息被存储起来，从而该信息在结束所述中断之后能够被提取。

45. 根据权利要求 44 所述的系统，其中，所述控制器使得所述被存储用于所述多于一个数据流之中的每一个数据流的安全配置信息，当来自此数据流的数据的处理过程

被恢复时，就被提取。

46. 根据权利要求 41 所述的系统，其中，所述控制器在所述中断之前提取被存储的所述安全配置信息，所述信息包括消息认证代码，以及当所中断的会话被恢复时就从所被提取的所述消息认证代码导出被更新的消息认证代码。

47. 一种用于处理存储经加密的数据的存储器系统中的数据的方法，所述存储器系统包括非易失性存储单元和密码化电路，所述方法包括：

使用所述电路对来自或流向所述单元的数据流中的数据执行密码化处理；以及

促使所述电路以交错方式对不同数据流中的数据执行密码化处理，其中，至少一个用于处理来自或流向所述单元的数据的会话被其它会话所中断，以及促使所述至少一个会话的安全配置信息在所述中断之前被存储，从而在结束所述中断之后所述安全配置信息能够被提取。

48. 根据权利要求 47 所述的方法，其中，所述安全配置信息包括：与数据的源或目的地、密码化钥匙、密码化算法、和/或消息认证代码相关的信息。

49. 根据权利要求 47 所述的方法，其中，所述的促使过程使得用于每个所述不同的数据流的所述安全配置信息被存储，以便此信息在结束所述中断之后能够被提取。

50. 根据权利要求 47 所述的方法，其中，所述的促使过程使得所述存储用于每个所述不同的数据流的安全配置信息，当来自此数据流的数据的处理被恢复时，就被提取。

51. 根据权利要求 47 所述的方法，进一步包括：在所述中断之前提取被存储的所述安全配置信息，所述信息包括消息认证代码，以及当所中断的会话被恢复时，就从被提取的所述消息认证代码导出被更新的消息认证代码。

带有流中数据加密/解密的存储器系统

技术领域

本发明总体上涉及存储器系统，尤其涉及一种带有流中数据加密/解密的存储器系统。

背景技术

移动装置市场正在朝着包括内容存储的方向发展，以使得通过生成更多的数据交换来增加平均收益。这就意味着内容被存储到移动装置上时必须受到保护。

便携式存储装置在商业中已经使用许多年了。它们从一个计算装置到另一个计算装置载入数据或用于存储备份数据。更加复杂的便携式存储装置，诸如便携式硬盘驱动器、便携式快闪存储盘以及快闪存储卡，还包括用于控制该存储管理的微处理器。

为了保护存储在便携式存储装置中的内容，所存储的数据通常被加密，只有被授权的用户才允许将该数据解密。

在具有已被提出的密码化能力的便携式存储装置中，用于存储管理的微处理器也紧密地被用在加密和解密处理过程中。例如，在美国专利6，457，126中描述了此种系统。在这种情况下，存储装置的吞吐量和性能就会受到严重影响。因此，需要有一种能缓解此问题的、改进了的本地存储装置。

发明内容

本发明的一方面是基于认可存储器系统的吞吐量可以被提高，其中，当数据流中的数据被传送到非易失性存储单元或从非易失性存储单元提取数据流中的数据时，在不密切地涉及任何控制器或微处理器的情况下，数据流中的数据由电路进行密码化处理。在一个实施例中，控制器仅仅涉及设置用于密码化处理过程中的参数而并不涉及该处理过程。在该实施例的一个执行中，通过配置寄存器来设置这些参数。

存储单元优选地包括快闪存储单元。同样优选地，存储单元、用于加密和/或解密数据的电路以及控制所述单元和该电路的控制器都被置于并封装到诸如存储卡或棒的实体内。

数据可被成页地写入存储单元中或从存储单元中被成页地读取。在许多用于加密和

解密的传统的密码化算法中对通常小于页的数据单位进行操作。。因此，本发明的其它方面是基于以下认可：密码化电路密码化处理正在被读取或写入的数据流中的一页或多页数据，以及数据流可以被控制，从而使得在全不涉及控制器的情况下，数据流具有多个源中所选择的源和多个目的地中所选择的目的地。

根据本发明的其它方面，可配置密码化电路，以启用多个算法中的一个或多个密码化算法的选择，从而在不涉及控制器或微处理器的情况下进行加密和/或解密。也可以配置电路，以使得在该配置之后，在不涉及控制器的情况下，该电路在多个连续阶段中密码化处理数据流中的数据。在该配置之后，在不涉及控制器的情况下，多个连续阶段中的密码化处理过程可以采用多于一个的密钥并且可以使用多于一种类型的密码化处理过程。

为了某些应用，可能希望存储器系统处理多于一个的数据流。在此情况下，控制器控制存储单元和电路，从而可以交错方式来密码化处理不同数据流中的数据。优选地，当数据流的处理在所述交错期间被中断时，用于密码化处理每个数据流的各种参数都被存储起来，以使得当此数据流的处理重新进行时，可以还原这些参数，从而继续进行密码化处理。在该特征的一个执行中，在启动写入操作时创建安全配置记录，以设置用于密码化处理的各种参数，并且这些参数在会话结束时被存储。然后，当读取操作启动时，从存储器中提取该记录，并在该操作的最后将之丢弃。当数据流被暂时中断以处理其它的数据流时，也将该记录存储起来，并且当原来的数据流的处理重新进行时，将该记录提取出。

上面描述的本发明的各方面可以单独使用或以它们的任意组合方式使用。

附图说明

图 1 是用于说明本发明的与主机装置进行通信的存储器系统的结构框图。

图 2 是图 1 的密码化引擎的某些方面的结构框图。

图 3 是用于说明本发明一方面的优选实施例的图 1 中的系统的操作的流程图。

图 4 是用于说明图 1 中的系统在处理多个数据流的操作以及安全配置记录的使用的流程图。

为了便于描述，在本申请中用相同的标号来标示相同的组件。

具体实施方式

图 1 的结构框图描述了本发明的各方面可在其中被执行的实例存储器系统。如图 1 所示，该存储器系统 **10** 包括中央处理单元（CPU）**12**、缓冲器管理单元（BMU）**14**、主机接口模块（HIM）**16** 和快闪接口模块（FIM）**18**、快闪存储器 **20** 以及外围存取模块（PAM）**22**。存储器系统 **10** 通过主机接口总线 **26** 和端口 **26a** 与主机装置 **24** 进行通信。可以是 NAND 类型的快闪存储器 **20** 为主机装置 **24** 提供数据存储。用于 CPU **12** 的软件代码也可以存储在快闪存储器 **20** 中。FIM **18** 通过快闪接口总线 **28** 和端口 **28a** 连接至快闪存储器 **20**。HIM **16** 适用于连接到类似数码相机、个人计算机、个人数字助理（PDA）、数字媒体播放器、MP3 播放器，以及蜂窝移动电话或其它数字装置的主机系统。外围存取模块 **22** 选择诸如用于与 CPU **12** 进行通信的 FIM、HIM 以及 BMU 的适当的控制器模块。在一个实施例中，虚线框内的系统 **10** 的所有组件可以包含在诸如存储卡或棒 **10'** 的单个单元中，并且优选地被封装在该卡或棒中。

缓冲器管理单元 **14** 包括主机直接存储器存取（HDMA）**32**、快闪直接存储器存取（FDMA）控制器 **34**、仲裁器 **36**、缓冲器随机存取存储器（BRAM）**38** 以及密码引擎 **40**。仲裁器 **36** 是共享总线仲裁器，以使仅仅一个主导装置（master）或启动器（initiator）（其可以是 HDMA **32**、FDMA **34** 或 CPU **12**）可在任何时间起作用，以及从属装置（slave）或目标装置（target）是 BRAM **38**。仲裁器负责将适当的启动器请求导入 BRAM **38** 中。HDMA **32** 和 FDMA **34** 都负责数据在 HIM **16**、FIM **18** 和 BRAM **38** 或 CPU 随机存取存储器（CPU RAM）**12a** 之间的传送。HDMA **32** 和 FDMA **34** 的操作是常规的，并且没有必要在这里详细描述。BRAM **38** 用于缓冲在主机装置 **24**、快闪存储器 **20** 和 CPU RAM **12a** 之间传递的数据。HDMA **32** 和 FDMA **34** 都负责在 HIM **16**/FIM **18** 和 BRAM **38** 或 CPU RAM **12a** 之间传送数据以及用于指出扇区传送的完成。

首先，当主机装置 **24** 读取来自快闪存储器 **20** 的数据时，通过快闪接口总线 **28**、FIM **18**、FDMA **34**、经加密的数据在其中被解密并存储在 BRAM **38** 中的密码引擎 **40** 来提取存储器 **20** 中所经加密的数据。然后，通过 HDMA **32**、HIM **16**、主机接口总线 **26** 将经解密的数据从 BRAM **38** 传送到主机装置 **24**。从 BRAM **38** 提取的数据可在其被传递到 HDMA **32** 之前，再次由密码引擎 **40** 进行加密，从而使传送到主机装置 **24** 的数据被再次加密，但是，与存储在存储器 **20** 中的数据被解密相比，使用了不同的密钥和/或算法。优选地，以及在另一实施例中，上述过程中不是将经解密的数据存储在 BRAM **38** 中，这样数据可能变得易被未经授权存取，而是，来自存储器 **20** 的数据可以被解密并且在其被传送到 BRAM **38** 之前再次被密码引擎 **40** 加密。然后，BRAM **38** 中被加密的数据如前所

述被传送到主机装置 **24**。这就说明了在读取过程中的数据流。

当数据由主机装置 **24** 写入存储器 **20** 时，数据流的方向被反转。举例来说，如果未被加密的数据由主机装置通过主机接口总线 **26**、HIM **16**、HDMA **32** 传送到密码引擎 **40**，则这样的数据可以在被存储到 BRAM **38** 之前由密码引擎 **40** 加密。另外，未被加密的数据可以存储在 BRAM 38 中。然后，数据在其通向存储器 **20** 的通道上被传送到 FDMA **34** 之前被加密。有鉴于被写入的数据经历多级密码化处理，优选地，在经处理的数据被存储到 BRAM **38** 中之前，密码引擎 **40** 完成此处理过程。

本发明的一个方面基于以下认可：如果在主机装置 **24** 和存储器 **20** 之间传递的数据流中的数据的上述密码化处理可以在最小程度涉及 CPU **12** 的情况下被执行，则装置 **10** 的吞吐量和由此导致的性能可以被大大改进。在下面对图 1 的描述中对此进行说明。

在上述的过程中，已经描述了具有两个不同数据源和目的地的数据流。在读取过程中，数据源是存储器 **20** 以及目的地是主机装置 **24**。在写入过程中，数据源是主机装置 **24** 以及目的地是存储器 **20**。另外，数据源（或目的地）也可以是 CPU **12**，而相应的目的地（或数据源）为存储器 **20**。然而在另一个操作中，为了批量加密和哈希（hash）操作，数据流可以从 BMU **14** 流向 CPU **12**。在下面的表格中给出了数据输入源和数据输出目的地以及可被应用的相应密码化处理过程的各种组合。

操作	引擎	数据输入源	数据输出目的地	描述
FDMA 写入 CPU	AES/DES/HASH	FDMA CPU 总线	CPU	该数据流动启动了对从安全存储器加载到 CPU 的数据的密码化操作（解密）
FDMA 从 CPU 读取	AES/DES/HASH	CPU	FDMA	该数据流动启动了由 CPU 存储到安全存储器的数据的密码化操作（加密）
FDMA 写入 BRAM	AES/DES/HASH	FDMA BRAM 总线	BRAM	该数据流动启动了对从 FIM 向 BRAM 传送的数据流的密码化操作
FDMA 从 BRAM 读取	AES/DES/HASH	BRAM	FDMA	该数据流动启动了对从 BRAM 向 FIM 传送的数据流的密码化操作
PAM 存取	AES/DES/HASH/PKI	PAM	PAM	该数据流动启动 CPU 存取用于批量加密和哈希操作的核心硬件
旁路	n/a	无写操作	无读操作	该数据流动启动 FDMA 在对数据流没有任何密码化操作的情况下存取 CPU 或 BRAM

如上述表格所示，一个附加的操作模式是旁路模式，其使 FDMA **34** 在没有对数据流

进行任何密码化操作的情况下能够沿着旁路通道(未在图1中示出)存取CPU **12**或BRAM **38**，好像并不存在密码引擎 **40** 而且 HDMA 和 FDMA 都沿着此旁路通道通过仲裁器 **36** 直接连接至 BRAM **38**。根据本发明的一个实施例，通过设置图 2（其是图 1 的密码引擎 **40** 的某些功能模块的结构框图）中的配置寄存器 **52**，CPU **12** 可以从多个数据源、多个目的地以及多个算法中预选择诸如数据源、数据目的地的处理参数以及诸如将被应用的密码化算法（或旁路模式）的密码化参数。

图 2 是更详细地示出其某些部件的密码引擎 **40** 的结构框图。如图 2 所示，密码引擎 **40** 包括：密码化模块 **50**、配置寄存器 **52**，根据上面的表格和将被使用的密钥（除了旁路模式之外），以及数据是否被加密、被解密或被哈希化（其被包含在短语“被加密处理”中）或不被密码化处理，配置寄存器 **52** 存储关于所选择的数据源、所选择的数据目的地、以及将被采用的密码化算法或旁路模式的安全配置信息或安全配置记录。安全配置信息或记录可以由 CPU **12** 写入配置寄存器 **52** 中。在这些信息被存储到配置寄存器 **52** 之后，密码引擎 **40** 然后就在不涉及 CPU **12** 的情况下执行相应的密码化处理过程。许多公共的加密算法将 128 位数据作为一个单位进行处理。这就可能小于一次写入或读取诸如快闪存储器的存储装置的多页数据一页的尺寸。每页通常存储一个或多个扇区的数据，扇区的大小由主机系统定义。在遵循磁盘驱动器所建立的标准之下，一个例子是由 **512** 字节用户数据的扇区加上关于用户数据的和/或这些数据存储在其中的块的开销信息（overhead information）的一些字节数。

在密码引擎 **40** 中可以采用计算机逻辑电路（未示出），以使得在由密码引擎 **40** 进行的密码化过程中不必涉及 CPU **12**，从而使整页数据每次均以小于一页的单位被密码引擎 **40** 进行密码化处理。在一个实施例中，密码引擎 **40** 是硬件电路。

如图 2 所示，方块框 **54**、**56** 和 **58** 表示可以由 CPU 选择以被密码化模块 **50** 执行的三种不同的密码化算法（分别为 Hash、DES、AES）。不同于这些算法的密码化算法也可以被使用并且也处在本发明的范围之内。将由密码化模块 **50** 处理的和发源自主机装置 **24** 或存储器 **20** 或 CPU **12** 的数据首先被存储在输入缓冲器 **62** 中，然后，由密码化模块 **50** 根据配置寄存器 **52** 中指定的密码化算法进行密码化处理。再后，经密码化处理的数据在根据配置寄存器 **52** 中的目的地信息而被传送到目的地中之前，先被存储到输出缓冲器 **64** 中。图 2 也包括从输入缓冲器 **62** 到输出缓冲器 **64** 的旁路通道 **72**，在该旁路通道 **72** 上写入存储器 **20** 或从存储器 **20** 读取的数据没有被密码化处理，这就是表格中的模式之一和上面描述的一种情况。

配置寄存器 52 也可以存储将被用在密码化过程中的密钥。在一个实施例中，该密钥被 CPU 12（诸如从存储器 20 中）提取并在由密码化模块 50 加密或解密之前先被存储到配置寄存器 52 中。在 CPU 12 将相关信息写入配置寄存器 52 之后，在没有涉及 CPU 12 的情况下，上述过程发生在密码引擎 40 中。为简化图 2，已经省略了某些计算机逻辑电路，它们使用配置寄存器 52 中的信息以选择密码引擎 40 中的算法、数据源和目的地，以及使用用于密码化处理过程的唯一的密钥和所选择的算法。在把所处理过的数据传送到输出缓冲器 64 之前，可以不止一次使用密码化模块 50 来处理输入缓冲器 62 中的数据。例如，希望首先对来自数据源中的数据进行解密，以及接着在将所解密过的数据传送至输出缓冲器 64 之前使用不同的密钥和/或算法加密所解密的数据。除了加密或解密数据之外，为确保数据的完整性，还将哈希算法应用于数据以获得数据的摘要（digest）或哈希值也是有益的。在所有这些情况中，或是通过使用密钥解密然后使用不同的密钥加密，或是为了获得摘要以及加密或解密数据，希望由密码化模块 50 对数据进行两次处理。很明显，也可以由密码化模块 50 对该数据进行多于两次的处理，例如，在顺序的阶段（多级操作）中连续发生的数据被解密、哈希化、然后被加密。换句话说，在多级（例如，具有两个或更多阶段）过程中，为了密码化模块 50 的多次处理，通过将已经被密码化模块 50 处理过的输出缓冲器 64 中的数据沿着反馈通道 66 传送至输入缓冲器 62，数据可以不止一次通过密码化模块 50。如果设想有多于两个阶段，则数据可以为了额外的处理过程被反馈额外次数。在过程的每一阶段，都可使用不同的算法和/或密钥。

如果希望进行多级处理过程，则 CPU 12 可用于将安全配置信息或记录输入到配置寄存器 52 中以确定数据被密码化处理的次数、以及多级处理过程中每个阶段使用的密钥和/或算法。在将这些信息写入配置寄存器 52 之后，多级处理过程中就不必要涉及 CPU 12 了。

当图 1 中的存储器系统 10 包括快闪存储器时，该系统可以另外包括其它类型可替代的非易失性存储器，诸如磁盘、光学 CD，以及所有其它类型的可再写非易失性存储器系统，并且上述各种优点可以等同地应用到这些可选的实施例中。在可选实施例中，存储器也可以优选地随同该存储器系统的余下的元件一起被封装到同一实体（诸如存储卡或棒）中。

图 3 的流程图说明了操作系统 10 的读取过程。CPU 12 在从主机装置 24 接收到读取指令之后启动读取操作（椭圆 150）。然后，CPU 12 通过将适当的安全配置信息或记录写入配置寄存器 52 来配置密码引擎 40，以及配置用于读取操作的 BMU 14，和用于操作的

诸如存储空间在 BRAM **38** 中的分配的其它的参数（方框 **152**、**154**）。CPU **12** 还配置 FIM **18**，例如通过确定存储器 **20** 中数据将被读取的位置（方框 **156**）的方式。然后，启动 HDMA 引擎 **32** 和 FDMA 引擎 **34**，从而，在没有涉及 CPU 的情况下（除纠错之外），就可执行包括密码化过程的上述过程。参看方框 **158**，当 CPU 接收到中断信号时，其检查以确认该中断信号是否是 FIM 中断信号（菱形 **160**）。当接收到 FIM 中断信号时，该 CPU 进行检查以确认该中断信号是否是指示在数据流中存在一个或多个错误的中断信号（**162**）。如果错误被指示，则 CPU 继续纠正 BRAM **38** 中的错误（方框 **164**）并且返回去配置 FIM **18** 以改变下次在存储器 **20** 中数据将被读取的位置（方框 **156**）。当 FIM 中断信号没有指示数据流中的错误时，其意味着 FIM 已经完成其操作并且 CPU 也返回方框 **156** 以重新配置 FIM。如果由 CPU 探测到的中断信号不是 FIM 中断信号，则 CPU 进行检查以确认其是否数据中断信号结束（菱形 **166**）。如果是，然后读取操作结束（椭圆 **168**）。如果不是，则该中断信号与数据的密码化处理过程不相关（即，时钟中断信号）并且 CPU 将其维护（未示出）并返回菱形 **160** 以进行中断信号检查。

对于写操作，仅需简单修改图 3。因为不存在对将被写入存储器 **20** 中的数据中 ECC 错误的处理，所以在写入操作中 CPU **12** 可以跳过菱形 **162** 和方框 **164** 中的过程。如果在写入操作中 CPU **12** 接收到 FIM 中断信号，这意味着 IFM 已经完成了其操作，并且 CPU 也返回方框 **156** 以重新配置 FIM。除此不同之处，写入操作基本上类似于读取操作。因此，一旦配置好密码引擎 **40**、BMU **14** 和 FIM **18**，则系统 **10** 就能够密码化处理所有数据（除旁路模式之外），并且在不涉及 CPU **12** 的情况下，完成用于会话的所有页的写入或读取，即使密码引擎 **40** 可以处理比页更小的单位的数据。

交错数据流

为了处理多个数据流，希望多个主机应用程序能够以并行的方式存取存储器 **20**。这就意味着，为了存储器系统 **10** 处理另外不同的数据流，当一个数据流的密码化处理过程被中断时，其也许还未完成。不同数据流的密码化处理过程通常采用不同的参数（例如，不同的密钥和算法，以及不同的数据源和目的地）。这些参数提供在数据流的相应的安全配置记录中。为了确保当已经中断的特定数据流的处理过程稍后被恢复时其相应的安全配置记录不被丢失，此记录被存储，并优选地存储在 CPU RAM **12a** 中。一旦恢复先前被中断的数据流的处理时，CPU **12** 就提取被存储的用于此数据流的安全配置记录，从而可以根据被存储的相应安全配置记录，使用正确参数继续进行该数据流的恢复性密码化处理过程。

图 4 是用于说明图 1 和图 2 中的系统在处理多个数据流的操作和安全配置记录的使用的流程图。CPU 检查是否已经接收到主机指令（方框 **202**、菱形 **204**）。当接收到主机指令（例如用于密码化处理第一数据流的指令）后，CPU 检查关于该指令是否是启动会话指令，诸如用于第一应用程序在装置 **24** 上运行的指令（菱形 **206**）。如果是，然后 CPU 检查是否已经请求写入会话（Write Session）（菱形 **208**）。如果已经请求写入会话，那么 CPU 根据来自主机装置的信息来创建安全配置记录（例如，根据上述表格和将被使用的密钥，以及数据是否将被加密、解密或被哈希化，所述安全配置记录为所选择的数据源、所选择的数据目的地，以及将被采用的密码化算法）（方框 **210**），并且启动用于第一数据流的第一会话。CPU **12** 将这些安全配置信息或记录存储在 CPU RAM **12a** 中。如果所请求的会话是读取会话，则 CPU 从存储器 **20** 中读取用于将被读取的数据的安全配置记录（方框 **240**）并将其存储到 CPU RAM **12a** 中。然后 CPU 返回并等待进一步的主机指令（**202**）。

当 CPU 接收到另一主机指令时，其再次检查以确认该指令是否是启动会话指令（菱形 **206**）。如果是，那么可以通过继续进行方框 **210** 或方框 **240** 来启动第二会话，例如，请求对第二数据流进行密码化处理的用于运行在主机装置 **24** 上的不同的第二应用程序的新的第二会话。再次将用于此第二数据流的安全配置信息或记录存储到 CPU RAM **12a** 中，这是写入和读取会话都存在的情形（方框 **210**，**240**）。可以使用同样的方式为另外的数据流创建另外的会话。CPU 返回方框 **202**，并检查下一个主机指令以确认该主机指令是否是启动会话指令（菱形 **206**）。因此，如所述来创建另外的会话，直到 CPU **12** 探测到不是菱形 **206** 中的启动会话指令的主机指令。

在此情况下，CPU **12** 检查下一个主机指令以确认该主机指令是否会话指令结束（菱形 **222**）。如果不是，则 CPU 然后检查以确认其是否是数据指令（菱形 **224**）。假设其是数据指令，则 CPU 确定哪个数据流是将被处理的数据流，并且根据用于该数据流的安全配置记录（通过写入配置寄存器 **52**）来配置密码引擎 **40**，并且密码引擎 **40** 以诸如根据图 3 中的过程的上述方式（或以旁路模式旁路密码引擎 **40**）执行读取或写入操作（方框 **226**）。

如果在读取或写入的过程中没有中断信号，则该过程将会继续进行直到 CPU 接收到结束会话指令（方框 **222**），这意味着在会话中将要被处理的所有页都已经被处理了。然而，如果存在中断信号，CPU 将接收主机数据指令以处理来自与系统 **10** 当前正在处理的数据流不同的数据流中的数据。在此种情况下，需要重新配置密码引擎 **40** 以处理此不同

的数据流。然后，CPU 从 CPU RAM **12a** 中提取用于此不同数据流的安全配置记录，（通过将提取的记录写入配置寄存器 **52**）重新配置密码引擎 **40**，以使密码引擎 **40** 正确地处理不同的数据流。

当在写入会话中接收到结束会话指令（方框 **222**）时，CPU 将安全配置记录连同被写入的数据存入存储器 **20** 中，以使得该记录可以在随后的读取操作中被提取（方框 228，方框 **230**）。对于读取操作，丢弃了存储在 RAM **12a** 中的安全配置记录，但是为了将来可能的读取操作而保留了存储在存储器 **20** 中的记录（方框 **242**）。

对于某些应用程序，避免篡改而保持存储器 **20** 中数据的完整性可能是重要的。为了确保存储在存储器 **20** 中的数据不被改变或损坏，希望从数据中导出与数据存储在一起的该数据的哈希值（hashed value）或摘要。当读取数据时，摘要或哈希值也被读取，从而使读取的哈希值或摘要可以与从已经被读取的数据中计算得到的摘要或哈希值相比。如果二者之间不存在差别，那么存储器 **20** 中的数据可能已经被改变或者是损坏。

一个普通的哈希函数是链式模块密码（CBC），在其中以时间顺序从正在被写入或读取的数据块中导出消息认证代码（MAC）。下面给出一个普通的 CBC 函数：

加密：

输入：m 位的密钥 k；l 位的 IV；l 位的纯文本块 p_1,---p_r。

输出：c_0，---，c_r 使得对于 $1\le i\le r$，$c_0\leftarrow IV$ 以及 $c_i\leftarrow e_k$（$c_{i-1}\oplus p_i$）。

解密：

输入：m 位的密钥 k；l 位的 IV；l 位的加密文本块 c_1,---c_r。

输出：p_0，---，p_r 使得对于 $1\le i\le r$，$p_0\leftarrow IV$ 以及 $p_i\leftarrow c_{i-1}\oplus\ e_k^{-1}(c_i)$。

上述值 c_0，---，c_r 是数据流 p_1，---，p_r 的消息认证代码（MAC）。IV 是初始向量，以及 k 是密钥。因此，当希望将数据 p_1，---，p_r 块写入存储器 **20** 时，由系统 **10** 中的密码引擎 **40** 通过使用诸如上述 CBC 函数的哈希函数来从数据块中计算出 MAC 值（例如，c_0，---，c_r），并且，包括 MAC 值、IV 以及密钥 k 和上述其它参数的相关安全配置记录和数据本身一起写入存储器 **20** 中。在上述公式中，e_k(x)表示 x 通过密钥 k 进行加密的一个处理过程，以及 e_k^{-1}(x)表示使用密钥 k 来进行解密 x。

当随后从存储器 **20** 中读出数据块 p_1，---，p_r 时，相关的安全配置记录也被读出，并且密码引擎 **40** 从 IV、安全配置记录中的密钥 k、和被读取的数据中计算出一组 MAC 值，并将该组值与从存储器 **20** 读出的那组 MAC 值进行比较。如果两组 MAC 值之间存在不同之处，则读取的数据可能已被改变或损坏。对于诸如上述 CBC 函数的一些哈希函数，

除了序列中的第一值之外，每个 MAC 值都是从前面的 MAC 值中得出的。这就意味着，在此情形下，该组 MAC 值是以时间顺序顺次被导出的。

对于主机装置 **24** 中的多个应用程序来说，希望能够以并行的方式存取存储器 **20**，以使得用户在使用另外的应用程序存取存储器 20 之前不必等待正在使用存储器 **20** 的应用程序的完成。这可表示，例如，当读取过程被中断时，并不是所有的数据块 p_1，---，p_r 都已经从存储器 **20** 中读出，从而使存储器系统（例如，图 1 和图 2 中的系统 **10**）可用于在装置 **24** 上运行的另外不同的应用程序。然而，在此情形下，在整个数据流已被读取之前以及在所有 MAC 值被计算出来之前，可以中断上述计算 MAC 值的过程。因此，当存储器系统恢复对数据 p_1，---，p_r 中未读取块的读取时，可能丢失先前计算得到的不完整组的 MAC 值，从而变得不可能计算出剩余的 MAC 值，这是因为其计算依赖于在前计算得到的 MAC 值。因此，本发明的另一方面是基于以下特征：前面计算得到的不完整组的 MAC 值与安全配置记录中余下的值（例如，IV、密钥 k、数据源和目的地、算法）一起被存储到诸如图 1 的 CPU RAM **12a** 中。那么，当存储器系统恢复对数据 p_1，---，p_r 中未读取块的读取时，前面计算得到的不完整组的 MAC 值仍是可用的，从而可以计算出剩余的 MAC 值。

在从主机 **24** 探测到会话指令结束之后，在方框 **242** 中的读取会话的结尾，CPU 把从存储器 **20** 读取的数据计算而得到的 MAC 值与存储在存储器 **20** 中的 MAC 值进行比较以确认所读取的数据的有效性。如果被接收到的主机指令不是上面指出的任一种，则 CPU 12 仅仅执行该指令并返回方框 **202**（方框 **250**）。

虽然以上通过结合各种实施例描述了本发明，但是，应当理解在不脱离本发明的范围内可以做出改变和修改，这仅仅由所附的权利要求和其等同物限定。本文所提到的所有参考文献结合于此以供参考。

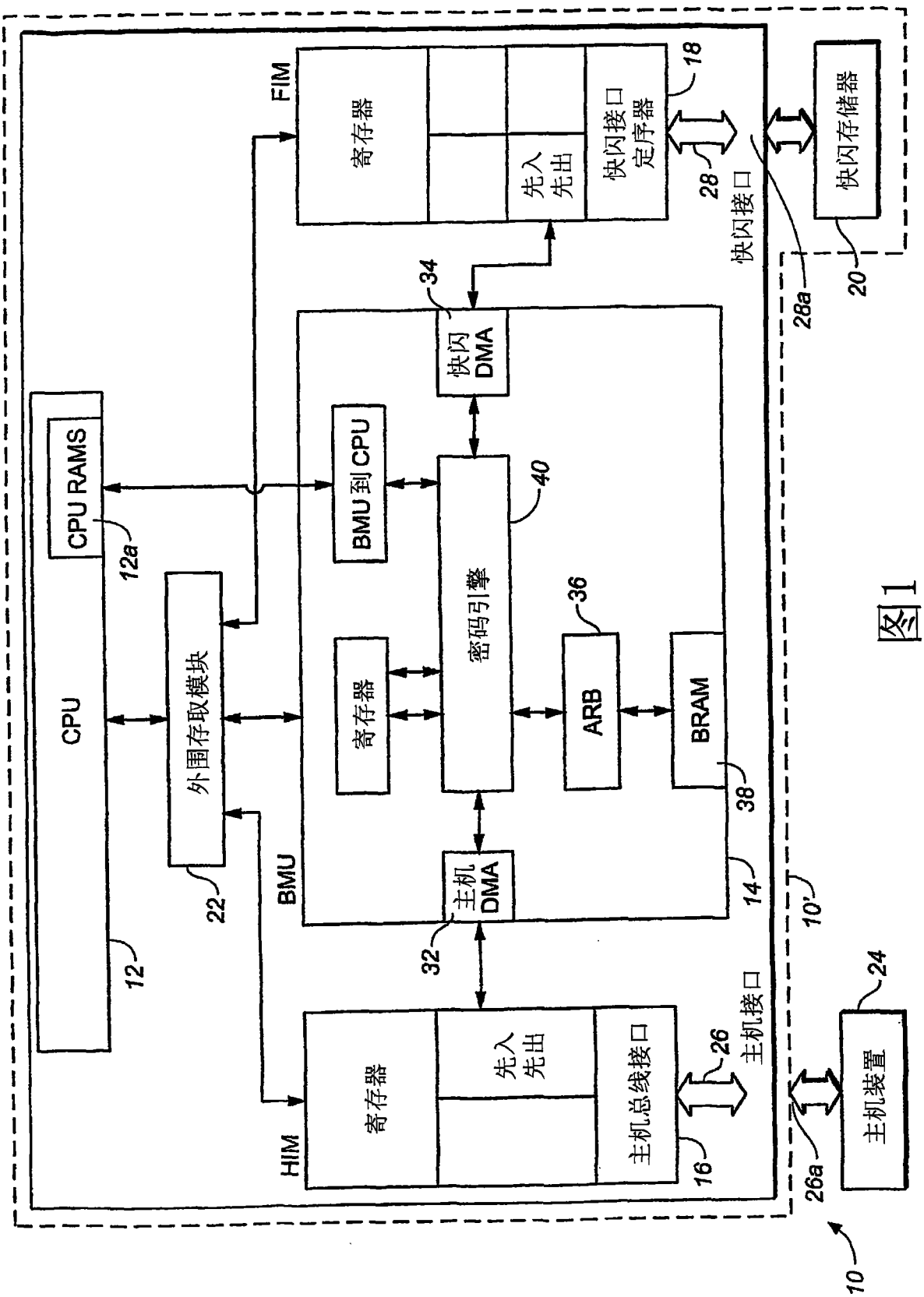

FIM
寄存器
先入
先出
快闪接口
定序器
18
28
快闪接口
快闪存储器
20
28a
34
快闪
DMA
CPU RAMS
12a
BMU到CPU
40
密码引擎
CPU
外围存取模块
寄存器
ARB
36
BRAM
38
22
BMU
主机
DMA
14
10'
32
12
主机接口
24
寄存器
先入
先出
主机总线接口
26
HIM
16
主机装置
26a
10

图1

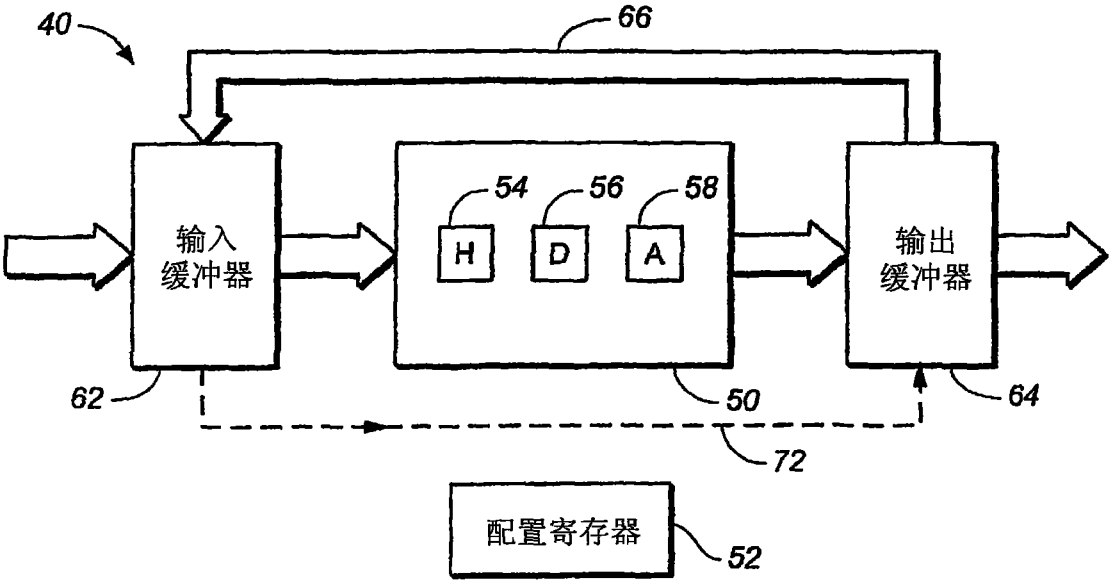


图2

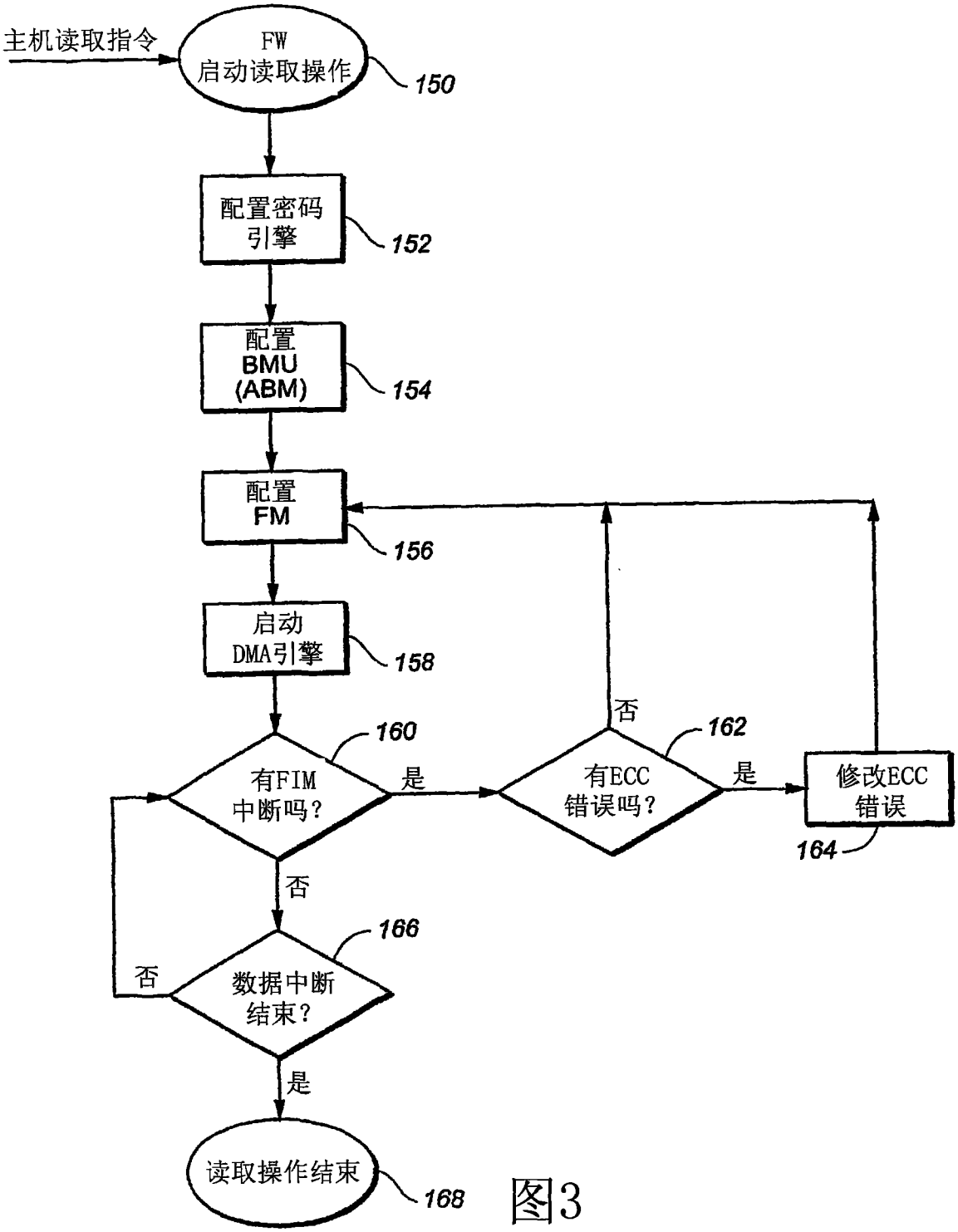

主机读取指令
FW
启动读取操作
150
配置密码
引擎
152
配置
BMU
(ABM)
154
配置
FM
156
启动
DMA引擎
158
160
有FIM
中断吗?
是
否
162
有ECC
错误吗?
是
否
修改ECC
错误
164
166
数据中断
结束?
否
是
读取操作结束
168

图3

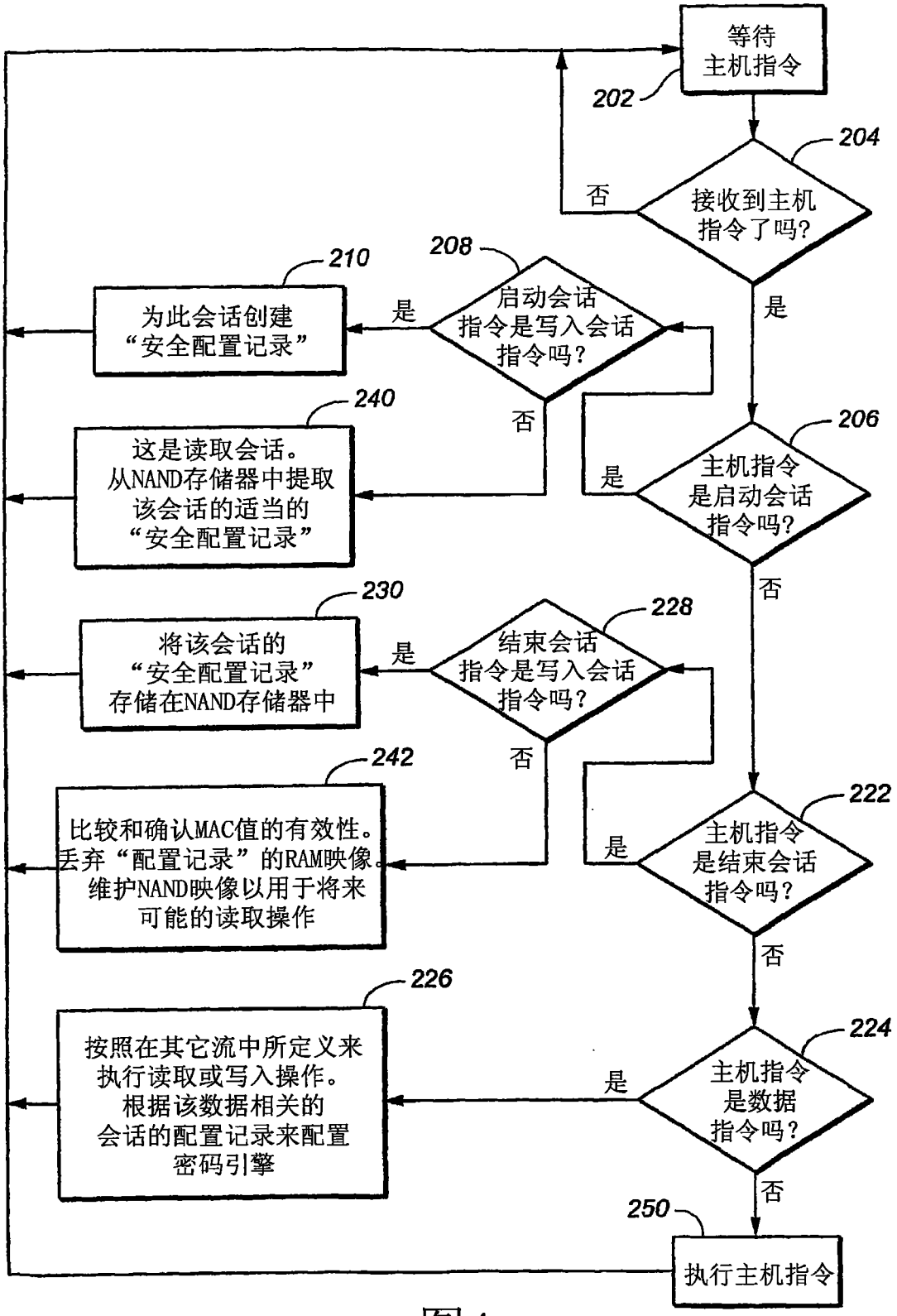

等待
主机指令
202
204
接收到主机
指令了吗?
否
是
208
启动会话
指令是写入会话
指令吗?
是
210
为此会话创建
“安全配置记录”
否
240
这是读取会话。
从NAND存储器中提取
该会话的适当的
“安全配置记录”
是
206
主机指令
是启动会话
指令吗?
否
228
结束会话
指令是写入会话
指令吗?
是
230
将该会话的
“安全配置记录”
存储在NAND存储器中
否
242
比较和确认MAC值的有效性。
丢弃“配置记录”的RAM映像。
维护NAND映像以用于将来
可能的读取操作
是
222
主机指令
是结束会话
指令吗?
否
226
按照在其它流中所定义来
执行读取或写入操作。
根据该数据相关的
会话的配置记录来配置
密码引擎
是
224
主机指令
是数据
指令吗?
否
250
执行主机指令

图4