



(12) 发明专利

(10) 授权公告号 CN 1902583 B

(45) 授权公告日 2010.05.12

(21) 申请号 200480039310.9

(51) Int. Cl.

(22) 申请日 2004.12.16

G06F 9/445(2006.01)

(30) 优先权数据

(56) 对比文件

10/751,033 2003.12.31 US

US 6665813 B1, 2003.12.16, 说明书第1栏第50-53行, 第3栏第33-37行, 第4栏第2-67行, 第5栏第1-60行, 第6栏第1-30行、附图1-5.

(85) PCT申请进入国家阶段日

2006.06.28

(86) PCT申请的申请数据

PCT/US2004/042965 2004.12.16

EP 1241571 A2, 2002.09.18, 全文.

EP 0803812 A1, 1997.10.29, 全文.

(87) PCT申请的公布数据

W02005/066773 EN 2005.07.21

US 20020039322 A1, 2002.04.04, 全文.

EP 0489204 A1, 1992.06.10, 全文.

(73) 专利权人 桑迪士克股份有限公司

地址 美国加利福尼亚州

审查员 孟宪超

(72) 发明人 卡洛斯·J·冈萨雷斯

安德鲁·汤姆林

(74) 专利代理机构 北京律盟知识产权代理有限

责任公司 11287

代理人 刘国伟

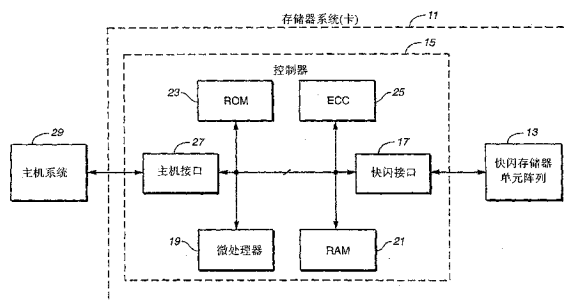
权利要求书 3 页 说明书 12 页 附图 7 页

(54) 发明名称

快闪存储器系统起动操作

(57) 摘要

本发明揭示将用于控制非易失性快闪存储器系统操作的固件码的多个副本存储于存储器系统的快闪存储器的不同合适位置处。还将这些位置的一地址映像图存储于所述快闪存储器内。一旦所述存储器系统初始化,其微处理器执行存储于所述存储器控制器内的启动码,以参考所述地址映像图并将所述固件的一副本从所述快闪存储器加载到一控制器存储器内,然后所述微处理器可从所述控制器存储器中执行所述固件副本以操作所述存储器系统来存储及检索用户数据。一错误校正码(ECC)用于检查所述数据,但所述快闪存储器内存储的所述二个或更多个固件副本的最佳部分用于减少对使用ECC的需要。当用户数据是以二种以上状态存储于所述快闪存储器内时,则可将所述固件码以二种状态存储于所述同一快闪存储器内。



1. 一种初始化一存储器存储系统的方法,所述存储器存储系统具有:快闪存储器,其包含存储于所述快闪存储器中不同位置中的固件码的至少第一及第二副本;一微处理器;一只读存储器,其包含微处理器可存取的启动码;及一随机存取存储器,其用于存储微处理器可存取的固件码,所述方法包含:

将所述固件码的第一及第二副本的每一者存储为数据区段或页面,所述数据区段或页面各自具有从其计算出来且与其一起存储的错误校正码,

执行所述启动码以将所述固件码的第一副本从所述快闪存储器转移至所述随机存取存储器,

识别所述固件码的所述转移的第一副本中的任何位错误,因而其发生所述数据区段或页面,

如果任何被识别的位错误是可用所述错误校正码来校正的,则校正所述可校正的错误位,

如果任何被识别的位错误是不可用所述错误校正码来校正的,则将所述固件码的所述第二副本的一部分,其包括包含所述不可校正位的数据区段或页面,读入所述随机存取存储器,以替代所述第一副本中相应的数据区段或页面,及

执行来自所述随机存取存储器的所述固件码的一无错误副本。

2. 如权利要求 1 所述的方法,其中识别所述转移的第一副本中的任何位错误包括:通过将所述固件部分在其从所述快闪存储器转移至所述随机存取存储器时连续传递经过错误校正码电路而依据所述固件码的所述第一副本的个别部分来计算错误校正码,并将所述计算的错误校正码与先前依据所述固件码的所述第一副本的所述部分计算出的错误校正码进行比较。

3. 如权利要求 2 所述的方法,其中校正所述错误位包括所述微处理器执行所述启动码的一错误校正算法以校正错误位。

4. 如权利要求 2 所述的方法,其中所述固件码的所述第一副本的所述个别部分包括:一个或一个以上数据区段,及先前依据所述个别部分计算出的并与所述个别部分一起存储于所述快闪存储器内的一错误校正码。

5. 如权利要求 1 所述的方法,其另外包含:在执行所述启动码以将所述固件码的所述第一副本从所述快闪存储器转移至所述随机存取存储器之前进行以下操作:

首先一次一个位置地存取所述快闪存储器内的复数个固定位置,直至发现一初始化存储器映像图是存储于所述复数个固定位置的至少一位置处且其包含其中存储有所述固件码的至少第一及第二副本的所述快闪存储器的所述不同位置的地址,

读取所述初始化存储器映像图的数据以获得所述地址,且

然后存取所述固件码的所述第一副本。

6. 如权利要求 5 所述的方法,其进一步包含:

识别从所述初始化存储器映像图读取的所述数据内的任何位错误,

如果所述读取数据内的位错误被识别为是可校正的,则校正所述错误位,及

如果所述读取数据内的位错误被识别为是不可校正的,则在不同条件下重新读取所述初始化存储器映像图的所述数据。

7. 如权利要求 1 所述的方法,其中将所述固件码的至少第一及第二副本存储于所述快

闪存储器的所述不同位置中,而所述存储器的每一快闪存储器存储组件仅存储一给定数量的一个或一个以上固件码位,而所述存储器存储系统的进一步特征在于将用户数据每一存储组件多于所述给定数量的用户数据位地存储于所述快闪存储器的其它位置中。

8. 如权利要求 7 所述的方法,其中所述给定数量的一个或一个以上位确切地是每一快闪存储器存储组件一个位。

9. 如权利要求 1 所述的方法,其进一步包含:

识别所述固件码的所述第二副本的所述转移的至少一部分内的任何位错误,及

如果识别所述固件码的所述第二副本的所述转移的至少一部分内的位错误是不可校正的,则在趋向于减少所述第二副本中所述转移的至少一部分内的位错误数量的条件下,重复读取所述固件码的所述第二副本的所述至少一部分。

10. 如权利要求 1 所述的方法,其进一步包含:在执行所述启动码以将所述固件码的所述第一副本从所述快闪存储器转移至所述随机存取存储器之前,检查在将固件存储于所述快闪存储器内时设定的一固件存在旗标的状态;及仅当设定所述固件存在旗标时,才继续执行所述启动码以将所述固件码的所述第一副本从所述快闪存储器转移至所述随机存取存储器。

11. 如权利要求 1 所述的方法,其另外包含:响应于识别一预定数量的一个或一个以上位错误的若干位错误,设定与所述固件码的所述第一副本的所述错误数据所存储的所述快闪存储器的位置相关的一内务处理旗标。

12. 如权利要求 11 所述的方法,其另外包含:响应于设定所述内务处理旗标,在已将所述固件码的一无错误副本转移进所述随机存取存储器后,校正所述固件码的所述第一副本的所述错误数据。

13. 如权利要求 12 所述的方法,其中校正所述固件码的所述第一副本的所述错误数据包括将所述固件码的所述校正的第一副本重新写入所述快闪存储器中。

14. 如权利要求 13 所述的方法,其中重新写入所述固件码的所述校正的第一副本包括将所述校正的第一副本重新写入与其原先存储的位置不同的一位置。

15. 如权利要求 12 所述的方法,其中校正所述固件码的所述第一副本的所述错误数据包括使用一错误校正码。

16. 如权利要求 12 所述的方法,其中校正所述固件码的所述第一副本的所述错误数据包括转移来自固件码的所述第二副本的良好数据。

17. 一种操作一存储器存储系统的方法,所述存储器存储系统具有:快闪存储器;一微处理器;一只读存储器,其包含可由所述微处理器存取的启动码;一随机存取存储器;及依据通过其的数据计算一错误校正码的电路,所述方法包含:

通过一次一个地将固件副本传递经过所述错误校正码电路而将固件码的至少第一及第二副本存储于所述快闪存储器的不同可寻址位置中及将由此计算出的所述错误校正码存储于所述快闪存储器内,

然后,通过促使所述微处理器执行所述启动码以经由所述错误校正码电路将所述固件码的所述第一副本从所述快闪存储器转移至所述随机存取存储器来起始所述存储器系统的操作,其中所述错误校正码电路依据转移的所述固件的所述第一副本计算一错误校正码,

利用所述计算并存储的错误校正码来识别所述固件码的所述转移的第一副本内的任何位错误,及

如果位错误被识别为是可校正的,则促使所述微处理器执行所述启动码内的一错误校正算法以校正所述错误位,以将所述固件码无任何错误地加载到所述随机存取存储器内,或者

如果位错误被识别为是不可校正的,则将所述固件码的所述第二副本的至少一部分转移到所述随机存取存储器内,以替代所述第一副本中包含所述不可校正的位错误的至少一部分,以将所述固件码无任何错误地加载到所述随机存取存储器内。

18. 如权利要求 17 所述的方法,其中存储所述固件码包括存储依据所述固件码的一个或一个以上区段个别计算出的错误校正码。

19. 如权利要求 17 所述的方法,其另外包含将一映像图存储于所述快闪存储器的预定复数个位置中的一个位置内,所述映像图包含固件码的所述至少第一及第二副本的所述可寻址位置;且其中所述微处理器执行所述启动码包括:通过首先通过以一次一个地存取所述预定的复数个位置直至找到所述映像图来定位所述映像图,及在存储有所述映像图的所述位置处读取所述映像图的内容。

20. 如权利要求 17 所述的方法,其中存储所述固件码另外包括设定一旗标以指示至少一固件码副本存在于所述快闪存储器内,且其中执行所述启动码以转移所述固件码的所述第一或第二副本中的任一副本包括:首先读取与所述副本相关的所述旗标并仅在设定所述相关旗标的情况下继续读取所述固件码的所述副本。

21. 一种起始一存储器存储系统的方法,所述存储器存储系统具有:快闪存储器,其包含存储于所述快闪存储器的不同位置中的固件码的至少第一及第二副本;一微处理器;一只读存储器,其包含微处理器可存取的启动码;及一随机存取存储器,其用于存储微处理器可存取的固件码,所述方法包含:

将所述固件码的所述至少第一及第二副本的一个位存储于所述快闪存储器的所述不同位置中的存储器单元的个别存储组件中,而将所述用户数据的多个位存储于所述快闪存储器的所述其它位置中的存储器单元的个别存储组件中,

执行所述启动码以将所述固件码的所述第一副本从所述快闪存储器转移至所述随机存取存储器,

在转移所述固件码的所述第一副本时,识别所述固件码的所述第一副本中的任何位错误,

校正所述转移的第一固件码副本内所述识别的位错误中的特定错误,或读取所述固件码的所述第二副本的至少一部分,以替换所述第一固件码副本中包含所述识别的位错误的至少一部分,及

执行来自所述随机存取存储器的所述固件码的一无错误副本。

## 快闪存储器系统起动操作

### 技术领域

[0001] 本发明通常来说涉及非易失性快闪存储器系统的控制器的初始化,而更具体来说涉及将操作固件存储于快闪存储器内以及一旦初始化或重启所述存储器系统便将所存储的固件读入一控制器存储器内。

### 背景技术

[0002] 如今人们正使用许多商业上已取得成功的非易失性存储器产品,尤其是小形状因数卡形式的产品,其采用一快闪 EEPROM(电可抹除及可编程只读存储器)单元阵列。所述卡内还包括一存储器控制器以与所述卡所连接的一主机介接并控制所述卡内所述存储器阵列的操作。此一控制器通常包括一微处理器、某种非易失性只读存储器(ROM)以及一易失性随机存取存储器(RAM)。除所述记忆卡实施方案外,另一选择是,还可将一个或一个以上集成电路形式的此类存储器系统嵌入各类主机系统中。

[0003] 二种通用的存储器单元阵列架构已在商业上获得应用:NOR与NAND。在典型的NOR阵列中,存储器单元连接在相邻的位线源极及漏极扩散之间,所述源极及漏极扩散是在一列方向上延伸并有控制栅极连接至字线(沿单元行延伸)。一存储器单元包括位于所述源极与漏极之间的至少一部分单元通道区域上的至少一存储组件。因此,所述存储组件上的一编程的电荷电平控制所述单元的一操作特征,然后可通过向所寻址的存储器单元施加适当的电压来读取所述单元。美国专利案第5,070,032、5,095,344、5,313,421、5,315,541、5,343,063、5,661,053及6,222,762号中给出此类单元,及其存储器系统中的使用及制造方法的实例。

[0004] 所述NAND阵列利用与个别位线之间一个或一个以上选择晶体管连接在一起的多由二个以上(例如16或32个)存储器单元组成的串联与一参考电位来形成多个单元列。字线延伸跨过大量所述列内的单元。在编程期间通过以下方式读取及验证一列内的一个单元:致使硬导通所述串中的其余单元以使流经一串的电流取决于所述寻址单元内存储的电荷电平。美国专利案第5,570,315、5,774,397、6,046,935及6,522,580号中给出NAND架构阵列及其作为存储器系统的一部分操作的实例。

[0005] 前面所参考的专利案及文章中所论述的当前快闪EEPROM阵列的电荷存储组件中最常见的是导电浮动栅极,起通常是由掺杂的多晶硅材料形成。可用于快闪EEPROM系统中的另一类存储器单元使用一非导电介电材料替代一导电浮动栅极来以一非易失性方式存储电荷。Chan等人的文章“真正的单晶体管氧化物-氮化物-氧化物EEPROM装置”(IEEE电子装置学刊第EDL-8卷第3号,1987年3月,第93至95页)中说明了此一单元。一由氧化硅、氮化硅及氧化硅(“ONO”)形成的三层电介质被夹在一导电控制栅极与所述存储器单元通道上一导电衬底的一表面之间。通过将电子从所述单元通道注入所述氮化物来编程所述单元,其中所述电子被捕获并存储于一受限制区域内,并通过将热电洞注入所述氮化物来抹除所述单元。2002年10月25日申请的序列号为10/280,352的美国专利申请案(公告案号为2003-0109093)中说明了采用介电存储组件的数种特定单元结构。

[0006] 如同在绝大多数集成电路应用中,对于快闪 EEPROM 存储器单元阵列,同样存在缩小构件建某一集成电路功能所需硅衬底区域的压力。为了增加一给定尺寸的记忆卡及其它类型封装的存储容量,或为了增加容量同时减小尺寸,一直希望增加硅衬底的一给定区域内所能存储的数字数据的数量。增加数据存储密度的一种方法是每一存储器单元及/或每一存储组件存储多于一个数据位。此方法是通过将一存储组件电荷电平电压范围窗口分成多于二个状态而实现。使用四个此类状态允许每一单元存储二个数据位,八个状态允许每一存储组件存储三个数据位,以此类推。美国专利案第 5,043,940 及 5,172,338 号中说明了使用浮动栅极的多状态快闪 EEPROM 结构及其操作,而对于使用介电浮动栅极的结构,则在前面提到的序号为 10/280,352 的美国申请案中有相关说明。一多状态存储器单元阵列的选定部分也可基于各种原因以美国专利案第 5,930,167 及 6,456,528 号中所说明的方式在两种状态(二进制)中操作。

[0007] 一典型的快闪 EEPROM 阵列的存储器单元被划分成一起抹除的分立的单元区块。即,区块是抹除单位。每一区块通常存储一个或一个以上数据页面,所述页面是编程及读取的最小单位,但在单一操作中可编程或读取多个页面。每一页面通常存储一个或一个以上数据区段,所述区段的尺寸由主机系统来定义。根据一所建立的关于磁盘驱动器的标准,一实例区段包括 512 个用户数据字节,再加上某些数量的关于用户数据及/或所述用户数据存储于其中的区块的开销信息的字节。存储器系统通常在每一区块内配置有 16、32 或更多页面,且每一页面均存储一个或仅数个主机数据区段。

[0008] 一快闪存储器系统中的控制器通常包括一微处理器,所述微处理器执行来自一固件操作系统的指令以控制所述存储器阵列的操作以及数据在所述阵列与所述主机系统之间的流动。在某些商业产品中,此固件存储于作为所述控制器一部分的一小型快闪 EEPROM 内,其通常是一与一个或一个以上存储器单元阵列集成电路芯片分离的集成电路芯片。使用一快闪存储器允许通过重新编程而容易地更新所述固件。通常,一旦所述系统通电或重置,便从所述快闪存储器读取所述固件并将所述固件读入控制器 RAM。最初,所述控制器微处理器执行所述控制器 ROM 内存储的少量启动码以将所述固件加载到 RAM 内。然后,由于可使用能够比所述快闪存储器更快读取的此一类型的存储器,所述微处理器执行来自所述 RAM 的固件指令。当然,所述 RAM 是易失性的,但是,如果断电,则在恢复电力后所述固件被再次从所述快闪存储器加载到所述 RAM 内并恢复所述存储器系统的操作。

[0009] 尽管此固件启动系统操作得相当好,但所述控制器集成电路芯片较昂贵,因为必须使用与用于形成所述芯片上的其它电路不同的一程序来形成所述快闪存储器。因此人们已建议:为了降低产品成本,将所述固件存储于不允许所述主机存取以存储用户数据的快闪存储器单元阵列的指定区块内。一旦系统初始化,所述控制器微处理器便执行所述 ROM 启动码以将所述固件从快闪存储器的指定区块加载到所述 RAM 内。此仍然允许通过重新编程所述快闪区块来改变及更新所有固件,因为所述 ROM 内仅存储少量码来载入所述固件。

## 发明内容

[0010] 显然,极为重要的是保持加载到所述控制器 RAM 内的固件的完整性。所加载的固件中的任何错误(即使仅在一个位内)也能致使所述存储器系统不规则地操作,或者甚至根本不能操作。除此类不规则操作或失效的其它不利影响外,用户数据在其编程或读取时

可能受到破坏,或者甚至使其不能从所述存储器中检索。即使后来通过重新加载所述固件或所述固件的另一副本来校正所述错误,用户数据或从所述快闪存储器读取用户数据的能力可能已经遭到不能修复的损害。因此,在将所述固件编程到所述快闪存储器的保留区块内的过程中以及在所述固件在系统初始化后向 RAM 转移期间,均采取步骤以改善所述固件的完整性,所述固件控制所述微处理器的操作以编程用户数据并将用户数据存储于所述快闪存储器的其它区块内。

[0011] 将所述操作固件的二个或二个以上副本存储于所述快闪存储器内的不同位置处。所述位置无需固定。而是,可最佳地选择每一卡或嵌入式存储器系统的存储器单元的保留区块以用于在将所述固件编程到所述存储器内(作为每一系统的最后制造步骤之一)时存储所述固件。在制造期间,还将具有所述固件副本所存储区块的地址的映像图编程到所述存储器的数个指定区块的一区块中。所述 ROM 启动码致使所述控制器微处理器依次寻址所述数个指定区块直至其找到所述地址映像图。然后将所述地址映像复制到所述控制器 RAM 内,并通过所述微处理器从所述地址映像图中读取所述固件的第一副本的起点存在于其中的快闪存储器区块的地址,然后使用读取的所述地址来寻址所述区块以开始将所述第一固件副本复制到 RAM 内。所述可编程固件地址映像图的使用允许将所述固件存储于不同卡上的不同快闪存储器位置而不会增加所述 ROM 启动码或启动过程的复杂性。

[0012] 尽管寻常的快闪存储器是在用于用户数据的多个状态中操作,但存储所述固件的保留区块可在二进制状态或者数量比用于所述用户数据的状态少的状态中操作。即,尽管所述快闪存储器大部分中的存储器单元的存储组件存储二个或二个以上数据位,但可操作所述保留区块的存储组件以各自存储较少数量的一个或一个以上数据位。此改进了不同状态之间的容限,并因此使得所述固件数据对干扰及其它可能引起错误的影响不太敏感。除此改进的可靠性之外,可更快地读取以较少数量的状态存储的数据。尽管此减小了所述保留区块内存储的数据的密度,但仅涉及相对较少的区块。因此,对作为一整体的存储器系统的影响通常较低。

[0013] 通常将所述固件数据与依据一已知冗余码算法从所述数据中计算出的一错误校正码(ECC)一起存储于所述快闪存储器内。所述 ECC 通常是与固件数据的每一区段或页面一起存储。一旦将所述固件数据从所述快闪存储器保留区块上载到所述控制器,便使用所述 ECC 检查所述数据的错误。如果一区段的一个或一个以上位是错误的,则可使用所述区段的 ECC 来校正所述位,其前提是错误位的数量在所使用的特定 ECC 算法的能力范围内。另一选择是,可从存储于快闪存储器内的一个或一个以上替代副本中读取所述区段。如果所述第一副本内的错误数量超出所述 ECC 算法对其进行校正的能力,则当然从一替代固件副本中读取所述区段。如果一区段的所有替代副本均包含错误,则可校正并使用错误最少的一副本。优选由作为所述控制器一部分提供的一硬件电路来执行所述 ECC 计算,而非由所述控制器微处理器在所述 ROM 启动码的控制下执行所述 ECC 计算。但是,一旦检测到一可校正的数据错误,所述微处理器便着手通过执行 ROM 启动码指令来校正所述错误。

[0014] 当在读取期间检测到固件数据位的错误时,还可使用一容限技术。即,如果通过使用一 ECC 确定存在错误,则第二次可以不同的参考电平读取相同的数据;即,为所述固件的一个或一个以上位的存储等级中的某一错误提供一容限。通常,将响应所述 ECC 检测到数量多于其可校正的错误而使用所述容限读取。另一选择是,可在所述 ECC 检测到错误后但

在使用所述 ECC 来校正所述错误之前采用一容限读取。在任一情况下,均使用所述 ECC 来验证借助容限读取的数据的有效性。如果不能以此方式读取有效数据,则读取第二固件副本。

[0015] 可在所述快闪存储器的一保留区段内设定一“固件存在”旗标以指示存储所述固件码的一副本。然后,所述启动码促使在寻址包含所述启动码的区域之前读取所述旗标。如果所述旗标不指示存在一相关的固件副本,则不做读取所述旗标的尝试。对于有效存储的固件也可使用一“不加载”旗标以防止在正执行诊断时加载所述固件。所述二个旗标的组合控制所述微处理器是否尝试从所述快闪存储器读取及加载固件。

[0016] 本发明的其他方面、优点及特征包含在以下关于其实例的说明中,应结合附图阅读此说明。本文所引用的所有专利案、专利申请案、文章及其它公告案的全部内容均出于各种目的以引用的方式并入本文中。

### 附图说明

[0017] 图 1 是一其中可实施本发明各个方面的非易失性存储器系统的方块图;

[0018] 图 2 示意性地说明图 1 的系统的非易失性存储器的地址空间,其显示包括其一映像图及固件文件在内的保留文件的存储;

[0019] 图 3 说明存储在图 1 及 2 的非易失性存储器内的映像文件的数据内容;

[0020] 图 4 是一显示图 1 的存储器系统在系统初始化后即刻将固件从非易失性存储器上下载到控制器的操作的一第一实施例的流程图;

[0021] 图 5 说明图 4 的流程图所说明的操作的一个方面;

[0022] 图 6 是一显示将固件从一主机下加载到图 1 的非易失性存储器系统内的流程图;

[0023] 图 7 是一显示在卡制造期间发生的一装载或升级固件的操作的流程图;及

[0024] 图 8 是一显示图 1 的存储器系统在系统初始化后即刻将固件从非易失性存储器上下载到控制器的操作的一第二实施例的流程图。

### 具体实施方式

[0025] 如上文背景技术中所述的快闪 EEPROM 系统在功能上可如图 1 所示。一非易失性存储器系统 11 通常包括一非易失性存储器单元阵列 13 和一控制器 15。存储器单元阵列 13 可以是上文背景技术中所述的类型,或某些其它类型的可再编程非易失性存储器。所述控制器包括:与存储器单元阵列 13 介接的电路 17、一微处理器 19、一随机存取存储器 (RAM) 21、一只读存储器 (ROM) 23、错误校正码 (ECC) 计算电路 25 及主机接口电路 27。存储器系统 11 通过主机接口电路 27 连接至一主机 29。主机 29 可以是一个人计算机、数字照相机、个人数字助理、蜂窝式电话、数字音频播放器或其它需要此一非易失性存储器系统的电子系统。存储器系统 11 可以嵌入所述主机内,并因此构成所述主机系统的一部分,或者呈一以电及机械可拆除方式与所述主机连接的卡的形式。

[0026] 对于用于具有一内部控制器的快闪存储器的此类可拆除卡,存在若干现有的标准。一此类标准(即个人计算机(PC)卡标准)提供三种类型的个人计算机卡的规格。所述个人计算机卡标准是个人计算机记忆卡国际协会(PCMCIA)的产物。CompactFlash™卡(CF™卡)在功能上可与所述个人计算机卡兼容但要小得多。一甚至更小的非易失性记忆卡

是多媒体卡 (MMC<sup>TM</sup>)。MMC<sup>TM</sup> 卡的一具有约相同尺寸的修改版本是后来的安全数字 (SD) 卡。对于适合用作记忆卡的这些及其它电子卡,存在若干标准。所述标准中的一些标准对任何公众成员开放,而其它标准则对使用所述卡而形成的协会成员开放。

[0027] 在操作中,控制器 11 从主机 29 接收数据区段连同将要编程所述数据的逻辑地址。然后,所述控制器将所述地址映像到存储器单元阵列 13 内物理页面的地址内。图 2 中给出阵列 13 的一实例性物理存储器地址映像图。所述存储器被划分成若干区块,例如一区块 31,其中每一区块均是作为单一抹除操作一部分而一起抹除的最小存储器单元单位。进而,每一区块被划分成若干存储器单元页面,例如页面 31 的页面 33、35、37 和 39。尽管为解说简单起见显示每一区块内仅存在四个页面,但通常使用更多的页面。发展趋势是具有更多页面的更大区块,其中每一页面均存储多个数据区段。

[0028] 存储器单元阵列 13 被划分成众多区块,如图 2 所示。所述区块中的数个区块被指定为“保留区块”,以用于存储控制器 15 用来管理阵列 13 的操作并与主机 29 通讯的固件、参数及其它数据。此实例中显示区块 41、43、45、47 及 49 是保留区块。所述区块无需是所述相同存储器系统的不同副本中的相同区块,而且甚至可能在一单一存储器系统操作期间动态地重新指定所述区块(如果变化的条件使得重新指定合意的话)。控制器 15 不允许主机 29 存取所述保留区块,相反,所述控制器使用存储于所述保留区块内的数据来操作所述系统。控制器 15 可使用所述系统的其余区块来存储由主机 29 提供的用户数据,不过,在某些情况下,不允许若干区块用于正常使用并将其指定为可在所述系统操作期间替代其它区块的冗余区块(如果此变得需要的话)。所述控制器将逻辑地址从所述主机映像到变得可用于存储用户数据的阵列的物理地址空间内。

[0029] 为解说所述控制器对操作固件的存储及使用,显示所述固件的第一副本是存储于相邻的保留区块 47 及 49 内,然而,不一定需要将超出一个区块容量的固件存储于物理上相邻的区块内。所述固件的一第二副本存储于保留区块 41 及 43 内。由于所述控制器所使用的固件及其它数据(例如,操作参数的数据)对于所述存储器系统的操作是如此重要,因此将其存储在二个副本内以确保防止在使用所述存储器系统时所述第一副本随时间而可能遭到破坏。所述固件文件所占据的区块数量当然取决于所述文件的尺寸及所述区块的数据存储容量。

[0030] 在所述系统的初始化(例如在通电时发生或响应一硬重置)期间,将所述固件码的一副本从快闪存储器 13 加载到所述控制器的 RAM 21 内。然后,微处理器 19 执行来自 RAM 21 的固件码。为了在加载所述固件码时向微处理器 19 提供指令,在 ROM 23 内提供少量的启动码。一旦系统初始化,微处理器 19 便读取并执行所述启动码的指令以将一固件副本及其它必需的操作数据从快闪存储器 13 的保留区段加载到 RAM 21 内。然后,微处理器 19 执行所加载的固件码的指令以控制所述存储器系统的操作。对于通常类型的 ROM 23,由于不能容易地更新或以其它方式改变所述启动码,因此将尽可能多的系统码放置于快闪存储器 13 内存储的固件码内,因为可容易地重新写入此固件码。

[0031] 所述启动码促使所述微处理器起初存取所述快闪存储器内一包含一映射图(其包括此信息)的较小文件,而非将所述固件及操作数据的地址永久地包含于所述启动码快闪存储器内。在此实例中,将所述映像图存储于所述快闪存储器的一页面中。为了能够将此映像图存储于若干页面中的任一页面中,所述启动码促使所述微处理器依次存取若干指

定的页面并从所述页面中读取所述数据直至其找到所述映像图。图 2 中显示此操作的一实例,其中指定页面 51、53、55 及 57。在此实例中,所述指定页面物理地分布于整个阵列的不同位置处的若干区块的第一页面(页面 0)。因此,如果已知所述阵列中的某些位置比其它位置更佳,则将所述映像数据编程到所述较佳位置内,因为这对于控制器能够精确地读取所述映像图很重要。例如,在具有一或数千吉位用户数据存储容量的一存储器系统中,可指定十或十二个此类潜在的映像页面。

[0032] 图 3 中显示存储于一实例地址映像页面中的数据。字段 61 包含一识别所述页面包含所述地址映像数据的唯一签名。所述微处理器需要存取的若干保留文件的地址数据包含在所述地址映像页面中,图 3 中显示两个此类保留文件 0 及 1 的数据。每一保留文件均包括四个数据字段。对于保留文件 0,字段 63 包含所述文件的一识别符(ID)。可将此文件指定为固件文件。字段 65 规定所述文件的长度,例如页面的数量。字段 67 中规定所述快闪存储器内所述保留文件的一第一副本的一地址,字段 69 中规定其第二副本的一地址。所述地址通常将是所述文件的第一页,其存储于所述文件长度字段 65 所表示的若干连续页面中。所述地址映像图内还包括所述保留文件 1 的相同数据、至少在所述初始化过程期间需要由所述控制器处理的任何其它数据。

#### [0033] 将固件上载到控制器的第一实施例

[0034] 图 4 的一流程图显示初始化存储器系统 11 以将固件码从快闪存储器 13 加载到控制器 RAM 21 内的一实例。如步骤 71 所示,一旦给所述系统通电或在一通电的系统中接收一硬重置,所述过程便开始。下一步骤 73 初始化所述控制器专用集成电路(ASIC),向快闪存储器 13 供电并向所述主机传送一指示所述存储器忙碌的信号。然后在步骤 75 中,读取并通过微处理器 19 来执行 ROM 23 内的启动码。

[0035] 步骤 75 包括在快闪存储器内定位所述地址映像页面(参见图 2),读取所述地址映像页面且然后读取一保留区块内的一文件,所述文件在所述地址映像中被识别为包括二个一位的旗标。所述旗标中的一个是一“FW\_Present”旗标,其指示所述快闪存储器内是否存储有固件码。此旗标允许所述控制器在所述快闪存储器内不存在任何固件码的情况下避免尝试读取固件码。当希望不将所述固件加载到所述控制器内(即便所述固件存在于所述快闪存储器内)时,设定一第二“不加载”旗标。例如,当在无固件码的情况下执行诊断例程时,设定此第二旗标。但是,如果未预期此一操作,则不需要使用所述“不加载”旗标。如果使用所述“不载入”旗标,则可将其存储于所述快闪存储器外,例如一寄存器或一硬件旗标内。假定使用二个旗标,则下一步骤 77 决定所述旗标是否指示存在固件而不存在所设定的“不加载”旗标。如果指示,则以一步骤 79 开始的一序列继续加载所述固件。如果不指示,则所述控制器进入一闲置回路,如步骤 101 所指示,且可选择让所述主机介入来控制所述过程。

[0036] 作为将所述二个旗标作为二个数据位存储于一保留区段内的一替代方案,可指定所述封装外的二个管脚用于设定所述二个位。例如,如果所述管脚其中之一接地,则设定那个管脚所代表的旗标。从而,作为步骤 75 及 77 的一部分,询问所述管脚的电状态。

[0037] 如步骤 79 所指示,所述控制器设置自身以读取所述固件,其中包括汇编步骤 75 中从所述地址映像页面读取的二个副本的地址并准备读取所述第一副本。可在读取所述 FW\_Present 旗标(步骤 75)的同时从所述保留区块中读取标称读取条件,例如在针对特定存储

器阵列的读取期间拟使用的各种电压电平,并在所述步骤 79 中使用所述标称读取条件设置所述读取。

[0038] 在步骤 81 中,读取所述第一固件副本的第一页面。在此实例中,由于每一页面均存储一数据区段,因此术语“区段”在所述流程图中用作读取单位。假如所述个别页面将要存储二个、四个或更多个数据区段,则将会同时读取一页面内的所有数据区段。然而,如图 4 的流程图所指示,一次仍只可处理一个数据区段。

[0039] 基本上采取与从所述快闪存储器中读取任何数据相同的方式将读取的固件数据传递经过所述控制器的 ECC 电路 25。从每一数据区段中计算出一 ECC 并将所述 ECC 与作为所述区段的一部分存储为开销数据的 ECC 相比较。所存储的 ECC 是在其编程期间根据所述数据计算得出。只要在所述区段层执行错误校正,便可进行此操作,无论每一页面中是否存储有一个或一个以上数据区段。另一选择是,如果将一单一 ECC 用于一单一页面内的多个主机区段数据,则对一页面仅进行一次所述 ECC 检查。然而,按惯例每一数据区段包括二个 ECC,一个用于所述固件数据部分,而另一个用于包括所述数据 ECC 及 / 或其它操作参数的开销部分。作为另一替代方案,可将与一个或一个以上固件数据区段相关的开销数据存储于不同于所述固件数据的一页面或区块内,但连续读取所述开销数据并使其相关在一起。

[0040] 在一步骤 83 中,在所说明的特定实例中,ECC 电路 25 促使从读取的固件数据中计算一 ECC,而且作为与连同所述数据一起读取的 ECC 的比较结果,提供若干指示所述 ECC 所覆盖的所读取数据区段状态的输出位。然后,执行所述启动码的处理器 19 促使读取所述状态位。如果所述状态位指示所读取的数据内无错误,则如步骤 85 所示,为读取下一页面而更新所述页面地址及类似者。如果未读取包含所述固件文件的第一副本的所有页面,则所述处理返回到步骤 79 以读取下一页面。然而,如果刚读取的页面是所述第一固件文件的最后页面,则所述处理继续进行下文说明的步骤 95、97 及 99。

[0041] 但是,如果在步骤 83 中,来自所述 ECC 电路 25 的状态位指示所述读取位中一个或一个以上位是错误的,则接下来在步骤 89 中通过执行所述启动码来决定是否启用控制器 15 以使用所述 ECC 来校正所述错误(如果能的话)。下文对此进行解说。如果启用,则在步骤 91 中由执行所述启动码的微处理器 19 依据所述状态位确定所述错误的数量及类型是否可由所使用的特定错误校正算法来校正。如果能,则在步骤 93 中通过执行所述启动码来校正所读取的数据,而然后所述处理继续进行到步骤 85,好似已无任何错误地读取了所述数据。但是如果依据 ECC 电路 25 所产生的状态位确定不能通过使用所述启动码中所包括的错误校正算法来校正所述错误,则在步骤 105 中识别依次轮到的下一固件副本。一典型的 ECC 算法能够依据位错误的复杂性而识别给定数量的位错误,但能够校正的位数量要少于所识别的给定数量。例如,如果不能校正第一固件文件副本 0 的区段数据,则代之以读取并使用第二固件文件副本 1 的相同区段。步骤 107 确定是否保留一未读取的固件数据副本,且如果保留,则促使所述处理继续回到步骤 81 以对所述第二固件副本的相同区段重复上面说明的过程。但是,如果未成功地从所述固件码的二个副本(或者二个以上副本,如果提供了的话,)读取所述区段,则所述处理继续进行到步骤 109,如下文所说明。

[0042] 但是首先返回步骤 89,如果未启用校正(此是寻常的初始条件),则下一步骤 103 促使设定一旗标以指示存在一 ECC 错误。然后,在刚才说明的回路中,通过步骤 105、107、81 及后面的步骤来读取下一副本的相同区段。通过步骤 103 设定的旗标与一固件副本中包含

一个或一个以上位错误的区段相关,且优选地存储为所述区段的开销数据的一部分或存储在其它地方以供随后参考,以便就所述数据区段采取某一补救动作。所述固件可包括一例程以用于搜寻在所述上载过程期间设定的所有此类旗标,从而一旦成功地将所述固件加载到所述控制器内,便可采取某一动作来直接校正所述快闪存储器内存储的错误固件数据区段或解决所述错误数据的一可能成因。举例而言,此时可通过校正所述数据且然后在相同的快闪存储器页面中重新写入经校正的数据来擦除或刷新旗标所指示的固件数据区段。例如,美国专利案第 5,532,962、6,151,246 及 6,222,762 号及 2003 年 10 月 3 日申请的序列号为 10/678,345 的美国专利申请案中即说明了若干特定的擦除过程。可通过使用一区段数据 ECC 或通过以不同于正常所用的参考电平读取数据(“容限读取”)来校正所述区段的数据,以便对已经移动(或似乎已由于与其它电荷存储组件的场耦合而移动)出原始严格范围(对于所存储的电荷电平已编程至的状态)外的所存储的电荷电平做出补偿。另一选择是,可从所述固件的另一副本读取具有未校正错误的的数据区段,并将所述数据区段插入所校正的副本。然后,将所校正的副本重新写入所述快闪存储器,优选重新写入一新的位置内,但也可重新写入至相同的位置(在抹除后)。

[0043] 再次返回步骤 89,在此实例中,最初停用对任何错误读取数据的校正。首先从依次轮到的下一固件副本中读取相同的区段直至确定不能无错误地从快闪存储器内存储的任一副本中读取一特定数据区段,而非校正来自一副本的错误数据。此是在无任何错误地从所有固件副本中读取给定区段的尝试已失败后通过步骤 107 来决定。然后,所述处理促使通过包括步骤 109、111、112 及 113(返回至步骤 81)的一回路依序一次一个地从所述固件副本中重新读取所述区段达一预定次数,直至无错误地读取所述区段数据,从而促使所述处理从步骤 83 继续到步骤 85。可在不同的容限条件下进行所述再试中的某些尝试,如步骤 112 所示。

[0044] 但是,如步骤 113 所确定,如果此读取在所允许数量的再试的最后尝试后仍未成功,则通过启用步骤 115 中的 ECC 校正来允许一旦下一次读取相同的区段数据便发生对所述错误的校正。当步骤 83 确定存在一可校正的 ECC 错误时,此举促使步骤 89 将所述处理引导经过步骤 91 及 93 的路径,如上文所说明。如果存在一不可校正的 ECC 错误,则通过步骤 105 及 107 再次对所述数据区段的下一副本做相同尝试。但是,如果借助所述固件数据区段的所有副本的 ECC 校正进行的所述读取失败,则在步骤 111 中,将所述处理置入步骤 101 所示的闲置回路状态。

[0045] 总之,对于此实例实施方案,当二个固件副本存储于所述快闪存储器内时,可对二个副本内均有错误的一给定数据区段进行多达一定次数的读取而不尝试借助所述 ECC 来校正所述数据,所述读取次数是由步骤 109 中建立的再试设定次数所决定。然后,借助以所述 ECC 校正所述数据的能力实施从每一副本中读取所述给定区段的最后一次再试。仅在二个数据区段均不可校正的情况下,所述过程才失败。

[0046] 另一选择是,但通常并非优选,可将步骤 89 设定为始终启用校正,但同时步骤 103 中设定所述旗标用于随后处理所述区段内的数据错误。在此一情况下,图 4 的处理将仅一次性从每一固件副本中读取所述给定数据区段。读取所述第一副本,并尝试通过所述 ECC 来校正其错误。如果成功,则到达步骤 85。如果不成功,则读取所述第二固件副本的给定区段,而且,如果存在错误,则尝试校正。如果所述作法均不成功,则所述处理继续到步骤

109。

[0047] 尽管图 4 中未具体说明,但优选采取与针对所述固件副本所述的方式类似的方式来执行读取所述地址映像页面及一保留区块的任何部分的步骤 75。即,一个或一个以上 ECC 是与此类数据存储在一起并用于在读取期间确定所读取的数据是否包含任何位错误。如果包含位错误且如果可校正,则可通过使用所述 ECC 来校正所述数据。如果不可校正,则可在不同的读取条件下重新读取所述数据。如果错误仍然存在,则对重新读取的数据使用所述 ECC。一旦得到校正,便重写有错误的映射页面及 / 或保留区块部分,通常写入所述快闪存储器内的一不同位置。如果不能从所述一个副本中获得良好的数据,则可保留所述映像页面的一复制副本。

[0048] 参考图 5,其在概念上显示依据结合图 4 所说明的所述处理的一部分从快闪存储器内所存储的二个副本中成功地读取固件的一实例。为解说简单起见假定所述固件为 8 个区段长,读入控制器 RAM 21 的一固件文件 121 的所述 8 个区段被显示为取自二个固件副本 123 与 125。区段 0、1、2、4、5 及 7 来自第一固件文件 123 及区段 3 和 6,由于不能从所述第一副本中成功地读取所述区段,因此,其是取自第二固件文件 125。

[0049] 返回图 4 的处理流程图,特别是步骤 87,一旦已读取一固件文件(例如图 5 的文件 121),所述处理便继续到步骤 95。作为步骤 95 的一部分,检查所读取的固件文件的开销字段 127 及 129(图 5)。这些字段各自包含所述固件文件的长度及 ROM 23 内存储的启动码版本的一指定,因为所述启动码在不同的存储器系统中可能随时间变化。将所读取文件 121 的实际长度与从字段 127 和 129 中读取的值相比较。还将从字段 127 及 129 读取的所述 ROM 启动码版本的指定与所述启动码本身中所包括的一版本指定相比较。如果在所述步骤 97 中确定所述长度及启动码指定的比较均为肯定的话,则将固件文件 121(图 5)上载到控制器 RAM 21(图 1),而所述过程结束。但是,如果所述固件文件长度或所述启动码版本指定的比较均为否定的话,则如步骤 101 所示,所述处理停止直至从所述主机接收某些进一步的指令。

#### [0050] 将固件下载至快闪存储器

[0051] 图 6 说明用于将固件编程到图 1 及 2 的存储器系统 11 的快闪存储器 13 的保留区段的一过程。此过程通常是在生产的最后阶段或在更新较早制造的卡的固件时由存储器系统的制造商通过主机系统 29(通常是计算机形式)来完成。所述存储器系统的采购者及用户通常将不会将固件码加载到所述快闪存储器内。

[0052] 参考图 6,步骤 131 指示所述下载过程的起点。在步骤 133 期间,交替地将不同的数据模式写入各种快闪存储器区块,并从不同的快闪存储器区块读取不同的数据模式,所述快闪存储器区块是存储所述固件数据的保留区块的候选者。作为此测试的结果,选择一个或一个以上区块的各二个固件码存储区域(例如,图 2 中的区块对 41、43 及 47、49)。仅选择允许无任何错误地读取已编程测试数据的区块。进一步,可额外地对所存储的测试数据执行一测试以决定所存储的电荷电平是否在最佳范围内。同样,可在编程所述测试数据后使所述存储器系统经历各种环境例程以决定所存储的数据是否因此而改变。

[0053] 在下一步骤 135 中,所述主机计算机将下载固件加载到所述控制器 RAM 21 内以提供指令将所述存储器控制固件写入快闪存储器 13 的选定保留区块内(图 1 及 2)。接下来,在步骤 137 中,通过所述主机来读取作为拟编程到所述快闪存储器的固件的一部分存储的

ROM 启动码兼容性指示以及存储于所述 ROM 23 内的启动码版本指示并对其进行比较。所述存储器系统控制固件的下载仅在与永久且不可重写地存储于 ROM23 内的启动码相兼容的情况下才会继续。如果指明不兼容性,则补救办法是将拟加载到所述快闪存储器内的系统固件码的版本改变成一与已存在的 ROM 启动码兼容的版本。

[0054] 然后,在步骤 139 中,所述主机向控制器 15 发送一固件写入命令,而在步骤 141 中,控制器 15 最初通过清除所述 FW\_Present 旗标来作出回应。上文已说明此旗标。接下来,在步骤 143 中,所述主机将所述固件数据的一组区段转移至所述存储器系统。如步骤 145 所示,指定由步骤 133 选择用于存储所述固件码的二个保留区块区域其中之一来接收所述第一固件码副本。然后控制器 15 将先前从所述主机接收的固件数据区段编程到所述第一选定保留区块内,如步骤 147 所示。如果此编程步骤成功,如步骤 149 所确定,则所述控制器促使将所述相同的固件数据区段编程到所述第二选定保留区块内。包括步骤 151、153 及 145 在内的一局部回路促使寻址所述第二位置,且其中通过步骤 147 将相同的数据区块编程到所述第二位置内。

[0055] 一旦已将该组数据区块编程到所述快闪存储器的二个位置内,则所述处理继续到一步骤 155 以确定是否有更多的固件数据区段需要编程。如果有,则在步骤 143 中通过所述主机来发送另一组数据区段,并通过包括步骤 145 至 153 在内的回路以相同的方式将所述数据区段编程到二个快闪存储器位置内。如步骤 155 所确定,在以此方式将所有固件码区段编程后,设定 FW\_Present 旗标(如步骤 157 所示),并在步骤 159 中完成所述下载操作。

[0056] 如果在步骤 149 中检测到一组区段编程失败,则将步骤 147 中对所述区段的编程重复预定次数,如步骤 161 及 163 所示。如果在所述次数的尝试中未成功地完成对任何一组区段的编程,则在步骤 165 中指明所述过程操作的失败,并终止所述过程。

[0057] 尽管如上文背景技术中所述在多状态中操作所述快闪存储器以每一电荷存储组件存储多于一个数据位,但可操作一多状态系统的保留区块以每一电荷存储组件存储较少数量的一个或一个以上位。举例而言,如果是在用于用户数据的八个状态中操作所述电荷存储组件,则可以每一电荷存储组件四个状态将数据存储于所述保留区块内。同样,如果所述用户数据将被存储在四个状态中,则可以二进制来存储保留区块数据。此举对于存储所述存储器映像图及固件的保留区块尤其有价值。由于当采用较少数量的状态时定义每一状态的范围较大,因此,以较少数量的状态存储的数据不太会受到错误的读取。当每一电荷存储组件中存储较少的状态时,在存储器晶体管阈电压范围与电荷存储电平范围之间有较大的容限,且所述范围较大。当然,以较少数量的状态操作的区块中存储较少的数据,但存储上文所述的存储器映像、操作参数、旗标及固件的保留区块的此类操作不会明显地影响所述存储器的存储容量,因为此仅需在较少数量的状态中操作小部分存储器。

[0058] 图 7 显示记忆卡形式的此类系统的制造商用于将固件数据编程到快闪存储器系统内的一整个过程,此过程既用于新的空白卡也用于其中正更新固件的卡。此过程包括使用图 4 步骤 165 中的固件上载以及图 6 步骤 167 中的固件下载。首先,将所述卡与一主机连接,所述主机在步骤 169 与 171 中确定所述记忆卡是否是一空白卡。如果是空白卡,则在步骤 173 及 175 中将所述存储器阵列格式化。接下来,如先前结合图 6 所述,将所述固件副本编程到所述卡存储器阵列内。一旦在步骤 177 中确定已成功地将所述固件编程到所述卡

内,所述过程即结束。然后所述卡准备好供使用。

[0059] 如果在步骤 171 中确定所述卡并非空白,则将假定驻留于所述快闪存储器内的固件加载到所述控制器 RAM 内,如上面结合图 4 所述。如果在一步骤 179 中确定所述固件上载失败,则执行一失败分析功能 181 以了解所述失败的原因。在图 4 中,当所述过程处于闲置回路 101 中时,认定存在失败。然后,作为响应而舍弃所述卡。

[0060] 但是,如果在步骤 179 中确定已正确地上载所述固件,且如果确定需要重新格式化所述卡(步骤 183)或应将所述固件升级至一更新的版本(步骤 185),则可通过步骤 167 的过程重新写入所述固件。然而,如果既不需要重新格式化也不需要固件升级,则对所述固件正确地所述快闪存储器加载到所述控制器 RAM 内的确认致使所述过程结束。然后,已确认所述卡准备好供使用。

#### [0061] 将固件上载到控制器的第二实施例

[0062] 图 8 显示在所述 ROM 启动码控制下将所述固件从快闪存储器 13 上载进入控制器 15 的 RAM 21(图 1)的一过程的一第二实施例。此程序类似于上文结合图 4 所述的第一实施例。图 8 中与图 4 的步骤相同或基本上相同的步骤具有相同的参考编号,在此不做进一步解说。这二个实施例之间的差异在于对所读取数据中存在错误的确定做出的响应。

[0063] 参考图 8,对应于步骤 83 中确定已通过使用所述 ECC 检测到一数据错误,此实施例的处理直接进行到上述步骤 103 中旗标的设定以供随后使用。在下一步骤 191 中,确定是否能通过所使用的 ECC 来校正所述错误以及所述错误是否包括比规定数量少的位(少于 N 个位)。如果是,则在步骤 193 中通过所述 ECC 来校正所述数据,而所述处理继续读取下一区段的固件数据。在此实施例中,仅在拟校正的位数量较少且少于所述 ECC 能够校正的读取区段的位数量的情况下,才会在所述处理的此较早阶段校正所述错误数据。这是因为校正所述数据要花费与所校正的位数量相关的处理时间量。

[0064] 因此,如果步骤 191 中的确定是不可校正或可校正所述数据但所述数据有 N 或多个位要校正,则下一步骤 195 决定哪一情况才是实际情况。如果所述 N 或多个错误位是可校正的,则步骤 197 确定是否已从所述快闪存储器内存储的一个或一个以上其它固件副本中读取了所述给定区段。如果是,则在步骤 193 中使用所述 ECC 来校正所述数据,而所述处理通过步骤 85 而继续到拟读取的下一数据区段。如果仍有尚未读取其给定区段的一固件副本,则步骤 199 切换至促使步骤 81 读取另一副本。

[0065] 但是,如果在步骤 197 中确定不可校正所述 N 或多个错误位,则可使用步骤 201 来确定是否已以一规则方式从所有所述固件副本中读取所述给定区段达规定的次数。如果否,则步骤 199 将寻址另一固件副本内的给定区段,并将通过步骤 81 来读取所述区段。如果所有固件副本均已被读取规定的次数,则采用异常努力来读取数据。规则读取的规定次数可以是 2、3 或更多次,但重新读取数据花费时间,而且在初始读取已产生不可校正的错误数据后,在相同的条件下重新读取相同的数据区段不太可能提供正确的数据。因此,可通过对每一固件副本进行规定次数的读取来实施所述过程,在所述规定次数完成后,如果无一次读取成功,则采取异常读取步骤。

[0066] 所述异常读取过程中的一第一步骤 203 是确定是否已经执行从所述给定固件副本中读取所述给定区段数据的一或多次规定尝试。如果未执行,则通过步骤 211 来设定异常读取参数,并通过所述不同的参数而在步骤 81 中再次读取所述区段。但是,如果通过所

述异常读取参数对所述给定数据区段进行规定的一次或多次读取而未成功,则步骤 205 确定此情形对于所述快闪存储器内存储的其它固件副本是否也如此。如果也如此,则认为所述固件上载已失败,如 207 中所示,不提供任何其它技术来读取或校正所述错误数据。但是,如果尚未以所述异常方式读取所有固件副本,则通过步骤 209 来设定所述异常读取参数,而下一步骤 199 促使通过步骤 81 来读取一不同固件副本的给定区段。

[0067] 所述异常读取过程可包括步骤 209 及 211 中对更可能读取错误数据的不同读取条件的设定。一称作“容限”的技术使得一存储器单元电荷存储组件的每一状态的读取范围比在对所述数据区段进行首次、普通读取期间使用的读取范围加宽。此具有减小所述范围之间的容限的作用,从而可能在所述读取过程中引入其它错误。但是,当不能正常读取所述数据时,容限还可致使正确读取已漂移出其各自范围的单元的状态。容限读取的正确与否是通过使用所述 ECC 并通过步骤 83 及后续步骤以与正常数据读取相同的方式来确定。

[0068] 结论

[0069] 虽然已经依据本发明的实例性实施例说明了本发明的各个方面,但是应了解,本发明享有在随附权利要求书的完整范畴内受到保护的权利要求。

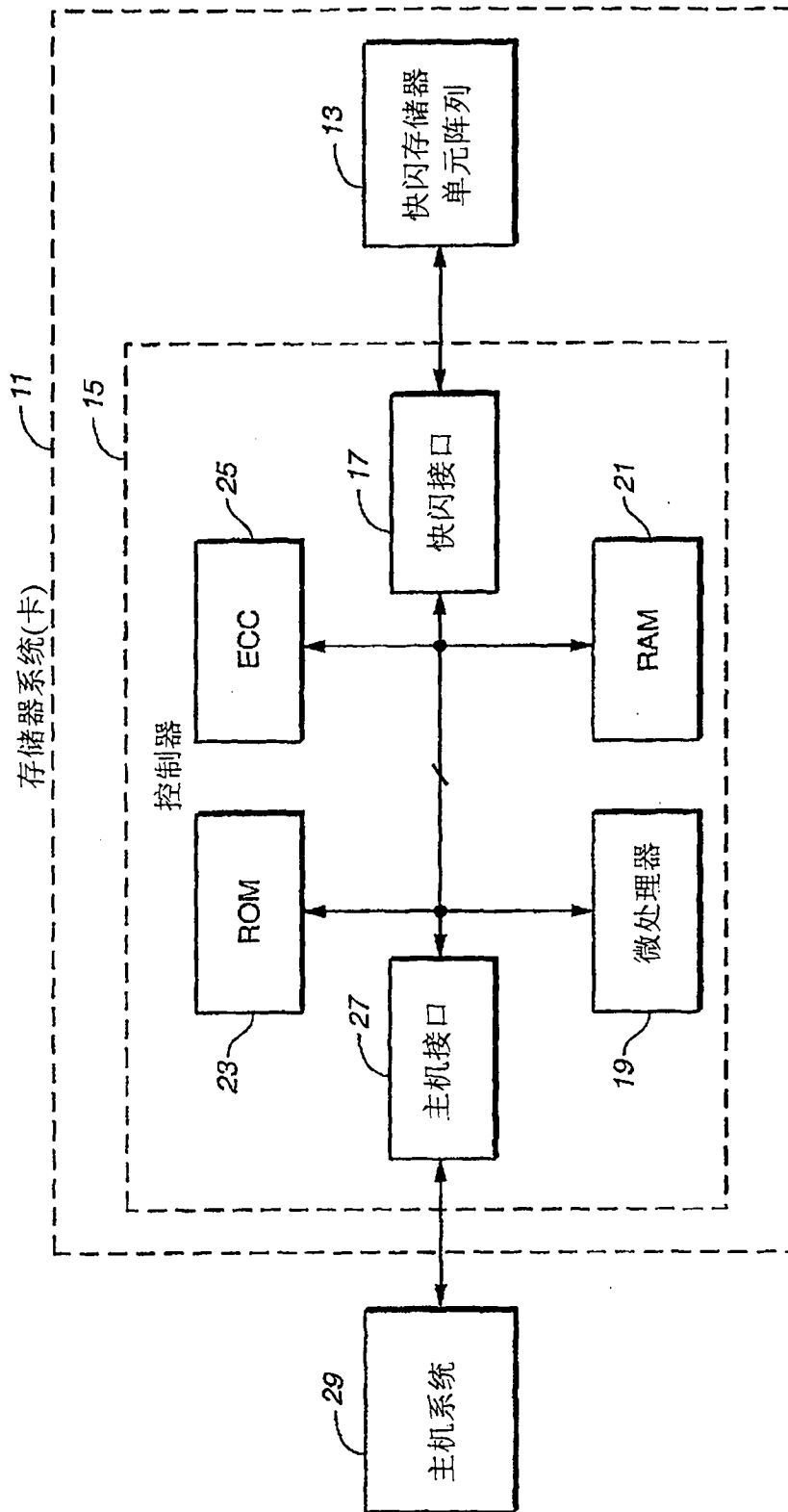


图 1



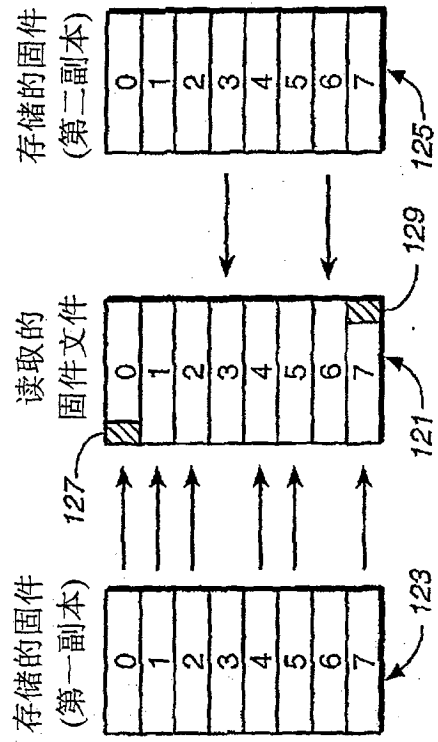


图 5



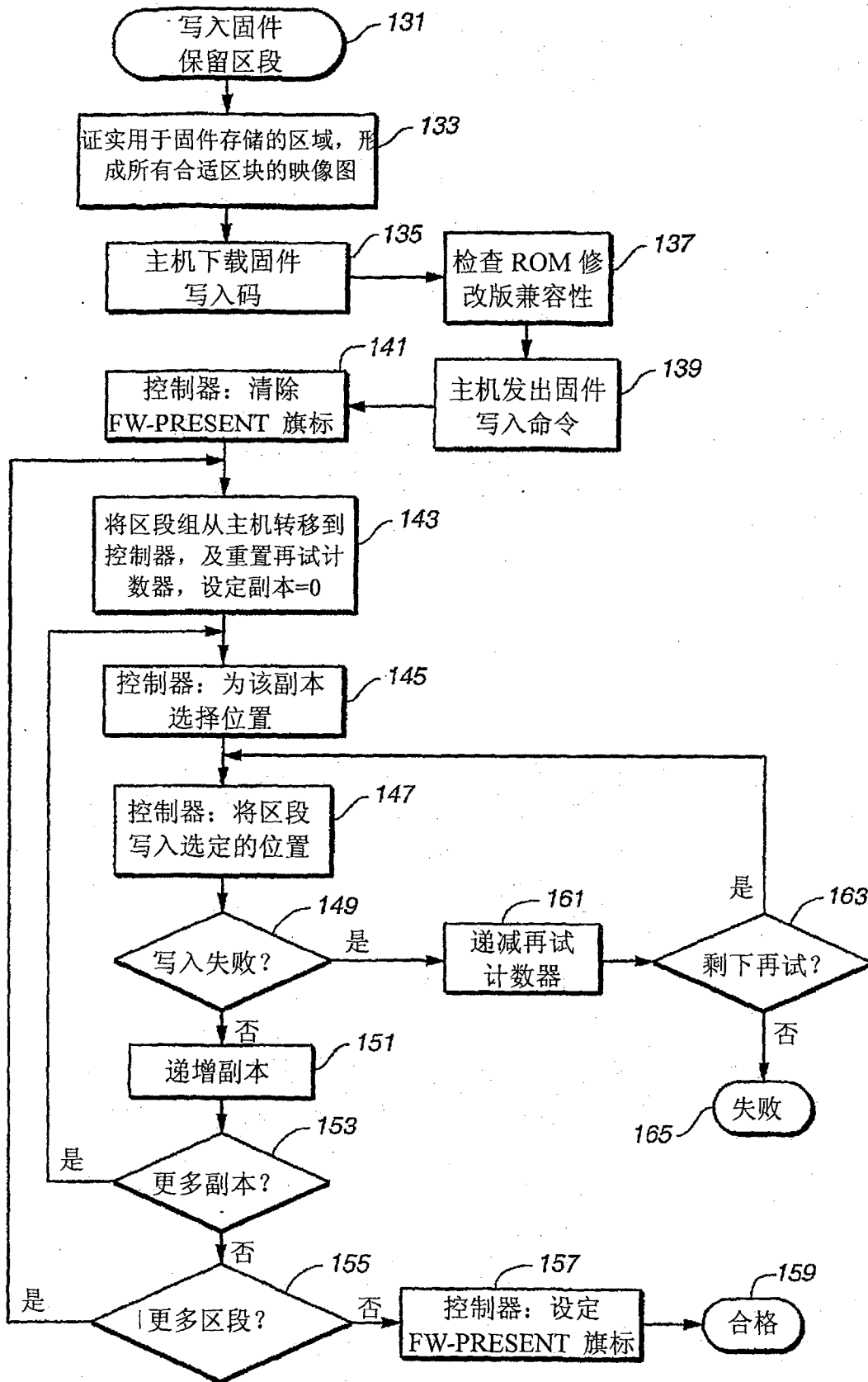


图 6

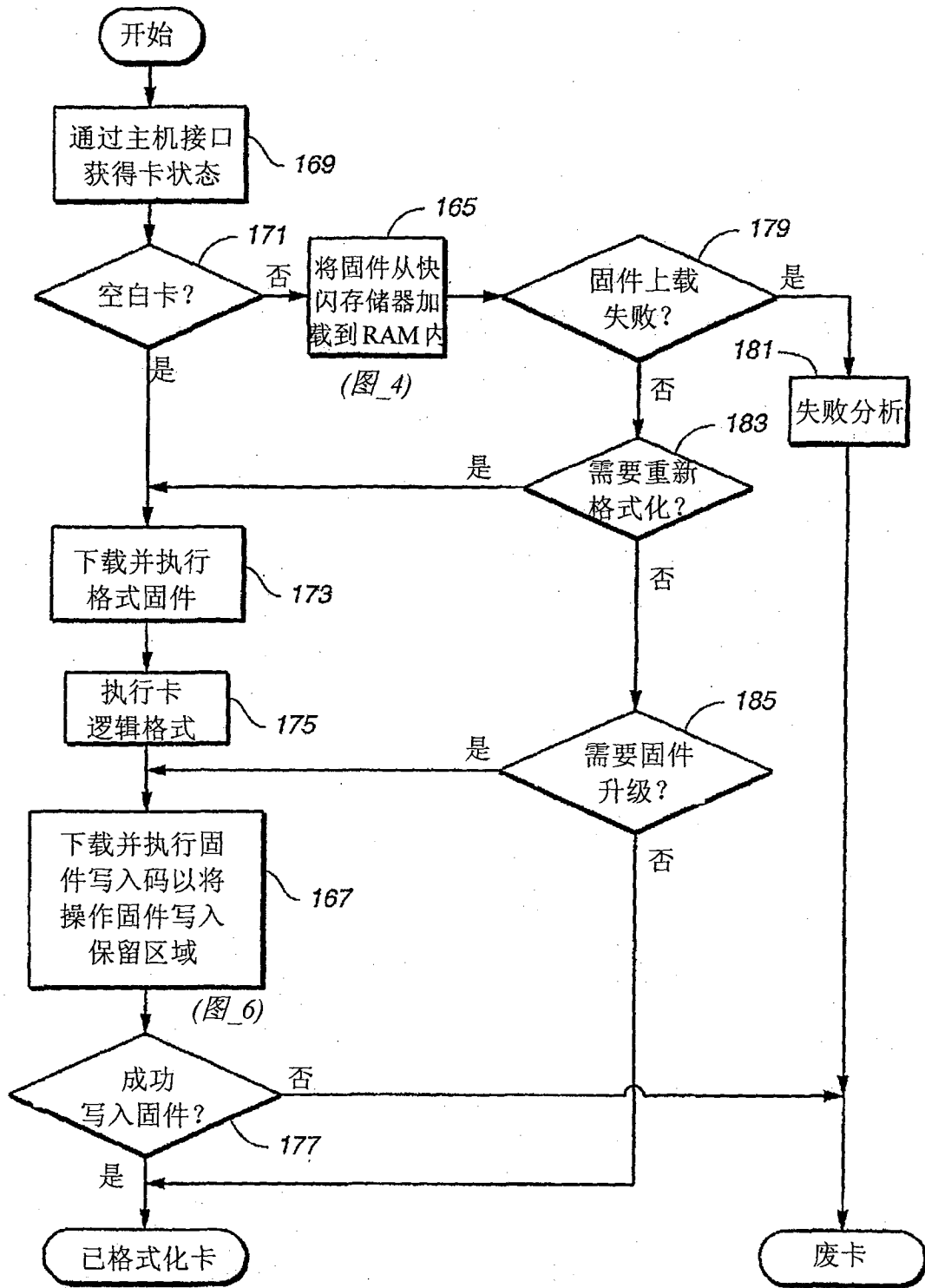


图 7

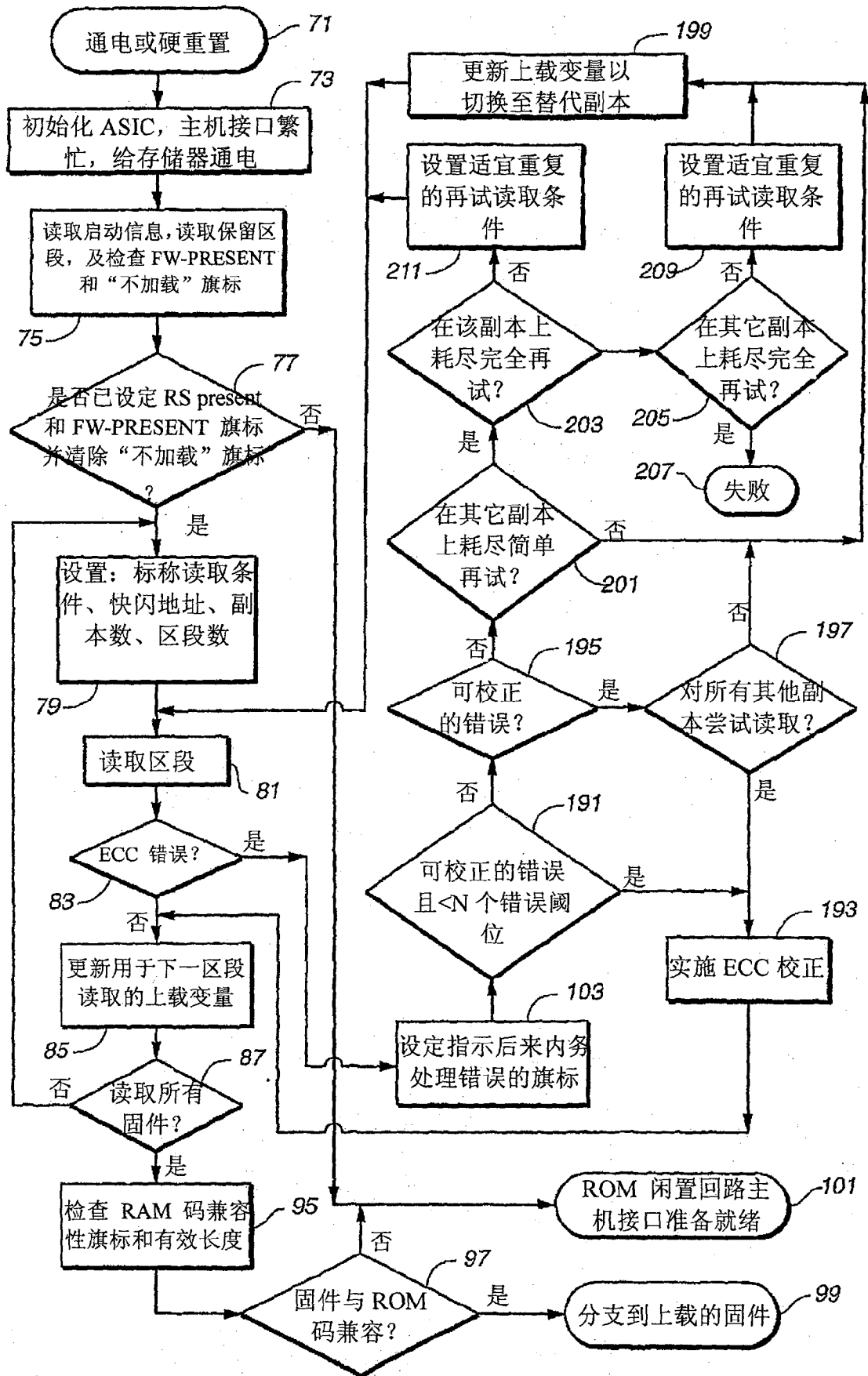


图 8