(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 103377063 A
(43)申请公布日 2013.10.30

(21)申请号 201210132893.6

(22)申请日 2012.04.28

(71)申请人 国际商业机器公司
地址 美国纽约

(72)发明人 唐文蔚 A·L·索德朗 吴松青

(74)专利代理机构 北京市中咨律师事务所 11247
代理人 于静 张亚非

(51)Int.Cl.
G06F 9/445(2006.01)

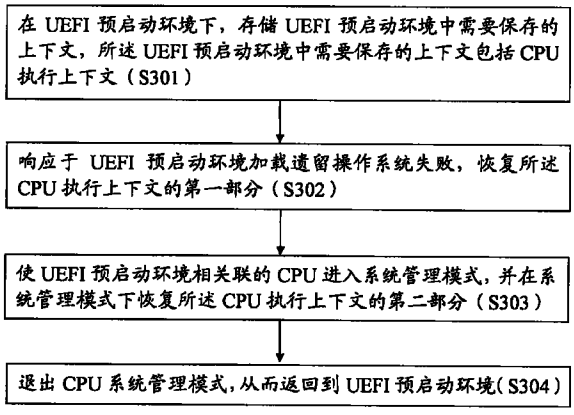
权利要求书2页 说明书10页 附图3页

(54)发明名称

从遗留操作系统环境恢复到UEFI预启动环境的方法和系统

(57)摘要

本发明公开了一种从遗留操作系统环境恢复到统一的可扩展固件接口UEFI预启动环境的方法和装置，方法包括：在UEFI预启动环境下，存储UEFI预启动环境中需要保存的上下文，所述UEFI预启动环境中需要保存的上下文包括CPU执行上下文；响应于UEFI预启动环境加载遗留操作系统失败，恢复所述CPU执行上下文的第一部分；使UEFI预启动环境相关联的CPU进入系统管理模式，并在系统管理模式下恢复所述CPU执行上下文的第二部分；以及退出CPU系统管理模式，从而返回到UEFI预启动环境。该方法和系统能够在遗留操作系统的启动尝试失败后恢复到UEFI预启动环境，从而来支持遗留操作系统启动失败时的诊断以及随后的UEFI操作系统的启动尝试。



CN 103377063 A

1. 一种从遗留操作系统环境恢复到统一的可扩展固件接口 UEFI 预启动环境的方法，包括：

在 UEFI 预启动环境下，存储 UEFI 预启动环境中需要保存的上下文，所述 UEFI 预启动环境中需要保存的上下文包括 CPU 执行上下文；

响应于 UEFI 预启动环境加载遗留操作系统失败，恢复所述 CPU 执行上下文的第一部分；

使 UEFI 预启动环境相关联的 CPU 进入系统管理模式，并在系统管理模式下恢复所述 CPU 执行上下文的第二部分；以及

退出 CPU 系统管理模式，从而返回到 UEFI 预启动环境。

2. 根据权利要求 1 所述的方法，其中所述 UEFI 预启动环境中需要保存的上下文还包括 UEFI 代码和数据所在的内存区域的地址以及该内存区域的内容。

3. 根据权利要求 2 所述的方法，其中所述存储 UEFI 预启动环境中需要保存的上下文步骤中采用 UEFI 保留内存来存储 UEFI 预启动环境中需要保存的上下文。

4. 根据权利要求 2-3 之一所述的方法，其中使用如下方法之一确定 UEFI 预启动环境加载遗留操作系统失败：

I) 响应于从遗留操作系统的加载器接收到软件中断，确定 UEFI 预启动环境加载遗留操作系统失败；

II) 在 UEFI 固件中加入看门狗时钟，UEFI 预启动环境在看门狗时钟预设的规定时间中没有接收到遗留操作系统启动成功的通知，看门狗时钟触发一个硬件中断，根据该硬件中断的触发确定 UEFI 预启动环境加载遗留操作系统失败；

III) 利用机器检查异常机制确定 UEFI 预启动环境加载遗留操作系统失败。

5. 根据权利要求 4 所述的方法，其中还包括：响应于 UEFI 预启动环境加载遗留操作系统失败，将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址。

6. 根据权利要求 5 所述的方法，其中还包括：在将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址后，关闭 PCI 设备的内存映射输入输出。

7. 根据权利要求 4 所述的方法，其中还包括：响应于 CPU 进入系统管理模式，将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址。

8. 根据权利要求 7 所述的方法，其中还包括：在将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址后，关闭 PCI 设备的内存映射输入输出。

9. 根据权利要求 5 或 7 所述的方法，其中还包括响应于 CPU 进入系统管理模式，对保存的指令指针寄存器进行修改，使所述指令指针寄存器指向进行遗留操作系统启动的调用指令的下一条指令。

10. 根据权利要求 5 或 7 所述的方法，其中还包括响应于返回到 UEFI 预启动环境，使 UEFI 控制的控制器的驱动程序重新运行。

11. 一种从遗留操作系统环境恢复到统一的可扩展固件接口 UEFI 预启动环境的系统，

包括：

存储装置，被配置为在 UEFI 预启动环境下，存储 UEFI 预启动环境中需要保存的上下文，所述 UEFI 预启动环境中需要保存的上下文包括 CPU 执行上下文；

第一恢复装置，被配置为响应于 UEFI 预启动环境加载遗留操作系统失败，恢复所述 CPU 执行上下文的第一部分；

第二恢复装置，被配置为使 UEFI 预启动环境相关联的 CPU 进入系统管理模式，并在系统管理模式下恢复所述 CPU 执行上下文的第二部分；以及

退出装置，被配置为退出 CPU 系统管理模式，从而返回到 UEFI 预启动环境。

12. 根据权利要求 11 所述的系统，其中所述 UEFI 预启动环境中需要保存的上下文还包括 UEFI 代码和数据所在的内存区域的地址以及该内存区域的内容。

13. 根据权利要求 12 所述的系统，其中所述存储装置中采用 UEFI 保留内存来存储 UEFI 预启动环境中需要保存的上下文。

14. 根据权利要求 12-13 之一所述的系统，其中所述第一恢复装置使用如下确定装置之一确定 UEFI 预启动环境加载遗留操作系统失败：

第一确定装置，被配置为响应于从遗留操作系统的加载器接收到软件中断，确定 UEFI 预启动环境加载遗留操作系统失败；

第二确定装置，被配置为在 UEFI 固件中加入看门狗时钟，UEFI 预启动环境在看门狗时钟预设的规定时间中没有接收到遗留操作系统启动成功的通知，看门狗时钟触发一个硬件中断，根据该硬件中断的触发确定 UEFI 预启动环境加载遗留操作系统失败；

第三确定装置，被配置为利用机器检查异常机制确定 UEFI 预启动环境加载遗留操作系统失败。

15. 根据权利要求 14 所述的系统，其中所述第一恢复装置还被配置为响应于 UEFI 预启动环境加载遗留操作系统失败，将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址。

16. 根据权利要求 15 所述的系统，其中所述第一恢复装置还被配置为在将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址后，关闭 PCI 设备的内存映射输入输出。

17. 根据权利要求 14 所述的系统，其中所述第二恢复装置还被配置为响应于 CPU 进入系统管理模式，将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址。

18. 根据权利要求 17 所述的系统，其中所述第二恢复装置还被配置为在将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址后，关闭 PCI 设备的内存映射输入输出。

19. 根据权利要求 15 或 17 所述的系统，其中所述第二恢复装置还被配置为响应于 CPU 进入系统管理模式，对保存的指令指针寄存器进行修改，使所述指令指针寄存器指向进行遗留操作系统启动的调用指令的下一条指令。

20. 根据权利要求 15 或 17 所述的系统，其中所述退出装置还被配置为响应于返回到 UEFI 预启动环境，使 UEFI 控制的控制器的驱动程序重新运行。

从遗留操作系统环境恢复到UEFI预启动环境的方法和系统

技术领域

[0001] 本发明涉及计算机系统中的固件，更具体地，涉及一种从遗留操作系统环境恢复到UEFI预启动环境的方法和系统。

背景技术

[0002] 传统的(Legacy)BIOS(Basic Input/Output System)是一种固件，作为基本输入/输出系统，负责在开机时做硬件启动和检测等工作，并且担任操作系统控制硬件时的中介角色。从Windows NT、Linux开始，这些操作系统已将过去需要通过BIOS完成的硬件控制程序放在操作系统中完成，不再需要调用BIOS功能。因为硬件发展迅速，传统的BIOS已经成为进步的包袱。

[0003] 现在已发展出最新的可扩展固件接口EFI(Extensible Firmware Interface)。统一的可扩展固件接口(Unified Extensible Firmware Interface)，即UEFI，是由EFI1.10为基础发展起来的，UEFI是一种详细描述全新类型接口的标准。这种接口用于操作系统自动从预启动的操作环境，加载到一种操作系统上，从而使开机程序化繁为简，节省时间。

[0004] UEFI使用C语言风格的参数堆栈传递方式以及动态链接的形式来构建系统，它比BIOS更易于实现，容错和纠错特性也更强，可以缩短系统研发的时间。并且，UEFI运行于32位或64位模式，达到处理器的最大寻址，克服了BIOS代码运行缓慢的弊端。而且UEFI体系的驱动是用EFI字节代码(EFI Byte Code)编写而成的，EFI字节代码是一组用于UEFI驱动的虚拟机器指令，在UEFI驱动运行环境下被解释运行，可以保证UEFI充分的向下兼容性。另外UEFI内置图形驱动功能，可以提供一个高分辨率的彩色图形环境，用户进入后能用鼠标点击调整配置，就像操作Windows系统下的应用软件一样简单。此外，UEFI使用模块化设计，在逻辑上分为硬件控制与操作系统软件管理两部分，硬件控制为所有UEFI版本所共有，而操作系统软件管理其实是一个可编程的开放接口，借助这个接口，主板厂商可以实现各种丰富的功能。比如本领域技术人员熟悉的各种备份及诊断功能可通过UEFI加以实现。因此，目前许多电脑厂商已经开始使用UEFI固件，并预计UEFI固件支持的机型的销售以后将占主导地位。

[0005] 从UEFI固件的角度看，操作系统可分为两种：一种是能够支持并利用UEFI固件的操作系统，例如Windows Server 2008R2；第二种是不能支持UEFI固件的操作系统，即遗留操作系统(Legacy OS)，UEFI可以提供兼容性支持模块，该兼容性支持模块使得UEFI固件可以加载并启动遗留操作系统，比如Windows XP 32-bit edition，Windows Server 2003for x86/等操作系统。

[0006] UEFI固件的运行环境为UEFI预启动环境，该环境执行UEFI固件代码，为操作系统准备启动环境的系统启动阶段。当UEFI固件的系统加载模块加载支持并利用UEFI的操作系统时，如果碰到问题不能成功加载操作系统，该系统加载模块可以直接回到UEFI预启动环境。但是当UEFI的系统加载模块加载并启动遗留操作系统时，系统加载模块中需要一

个兼容性支持模块，或者不存在系统加载模块，而直接利用兼容性支持模块，来使得加载并启动遗留操作系统成为可能。但是在现有技术中，一旦 UEFI 进入系统兼容性模块进行遗留操作系统的启动尝试，即使遗留操作系统的启动尝试失败，也没有办法回到 UEFI 预启动环境，这样，系统启动人员无法进行问题的诊断。

发明内容

[0007] 根据本发明一个方面，提供了一种从遗留操作系统环境恢复到 UEFI 预启动环境的方法，包括：

[0008] 在 UEFI 预启动环境下，存储 UEFI 预启动环境中需要保存的上下文，所述 UEFI 预启动环境中需要保存的上下文包括 CPU 执行上下文；

[0009] 响应于 UEFI 预启动环境加载遗留操作系统失败，恢复所述 CPU 执行上下文的第一部分；

[0010] 使 UEFI 预启动环境相关联的 CPU 进入系统管理模式，并在系统管理模式下恢复所述 CPU 执行上下文的第二部分；以及

[0011] 退出 CPU 系统管理模式，从而返回到 UEFI 预启动环境。

[0012] 根据本发明的另一个方面，提供了一种从遗留操作系统环境恢复到 UEFI 预启动环境的系统，包括：

[0013] 存储装置，被配置为在 UEFI 预启动环境下，存储 UEFI 预启动环境中需要保存的上下文，所述 UEFI 预启动环境中需要保存的上下文包括 CPU 执行上下文；

[0014] 第一恢复装置，被配置为响应于 UEFI 预启动环境加载遗留操作系统失败，恢复所述 CPU 执行上下文的第一部分；

[0015] 第二恢复装置，被配置为使 UEFI 预启动环境相关联的 CPU 进入系统管理模式，并在系统管理模式下恢复所述 CPU 执行上下文的第二部分；以及

[0016] 退出装置，被配置为退出 CPU 系统管理模式，从而返回到 UEFI 预启动环境。

附图说明

[0017] 通过结合附图对本公开示例性实施方式进行更详细的描述，本公开的上述以及其它目的、特征和优势将变得更加明显，其中，在本公开示例性实施方式中，相同的参考标号通常代表相同部件。

[0018] 图 1 示出了适于用来实现本发明实施方式的示例性计算系统 100 的框图；

[0019] 图 2 示出本发明所针对的 UEFI 预启动环境的框图；

[0020] 图 3 示出了根据本发明的一种实施方式的从遗留操作系统环境恢复到 UEFI 预启动环境的方法的步骤；

[0021] 图 4 示出了 UEFI 固件定义的内存区域的类型；以及

[0022] 图 5 示出了从遗留操作系统环境恢复到 UEFI 预启动环境的系统结构框图。

具体实施方式

[0023] 下面将参照附图更详细地描述本公开的优选实施方式。虽然附图中显示了本公开的优选实施方式，然而应该理解，可以以各种形式实现本公开而不应被这里阐述的实施方

式所限制。相反，提供这些实施方式是为了使本公开更加透彻和完整，并且能够将本公开的范围完整的传达给本领域的技术人员。

[0024] 图 1 示出了适于用来实现本发明实施方式的示例性计算系统 100 的框图。如图 1 所示，计算机系统 100 可以包括 :CPU(中央处理单元)101、RAM(随机存取存储器)102、ROM(只读存储器)103、系统总线 104、硬盘控制器 105、键盘控制器 106、串行接口控制器 107、并行接口控制器 108、显示控制器 109、硬盘 110、键盘 111、串行外部设备 112、并行外部设备 113 和显示器 114。在这些设备中，与系统总线 104 耦合的有 CPU 101、RAM 102、ROM 103、硬盘控制器 105、键盘控制器 106、串行控制器 107、并行控制器 108 和显示控制器 109。硬盘 110 与硬盘控制器 105 耦合，键盘 111 与键盘控制器 106 耦合，串行外部设备 112 与串行接口控制器 107 耦合，并行外部设备 113 与并行接口控制器 108 耦合，以及显示器 114 与显示控制器 109 耦合。应当理解，图 1 所述的结构框图仅仅是为了示例的目的，而不是对本发明范围的限制。在某些情况下，可以根据具体情况增加或减少某些设备。

[0025] 所属技术领域的技术人员知道，本发明可以实现为系统、方法或计算机程序产品。因此，本公开可以具体实现为以下形式，即 :可以是完全的硬件、也可以是完全的软件 (包括固件、驻留软件、微代码等)，还可以是硬件和软件结合的形式，本文一般称为“电路”、“模块”或“系统”。此外，在一些实施例中，本发明还可以实现为在一个或多个计算机可读介质中的计算机程序产品的形式，该计算机可读介质中包含计算机可读的程序代码。

[0026] 可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储介质。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。计算机可读存储介质的更具体的例子 (非穷举的列表) 包括 :具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器 (RAM)、只读存储器 (ROM)、可擦式可编程只读存储器 (EPROM 或闪存)、光纤、便携式紧凑磁盘只读存储器 (CD-ROM)、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

[0027] 计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括——但不限于——电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

[0028] 计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括——但不限于——无线、电线、光缆、RF 等等，或者上述的任意合适的组合。

[0029] 可以以一种或多种程序设计语言或其组合来编写用于执行本发明操作的计算机程序代码，所述程序设计语言包括面向对象的程序设计语言 - 诸如 Java、Smalltalk、C++，还包括常规的过程式程序设计语言 - 诸如”C”语言或类似的程序设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络——包括局域网 (LAN) 或广域网 (WAN)- 连接到用户计算机，或者，可以连接到外部计算机 (例如利用因特网服务提

供商来通过因特网连接）。

[0030] 下面将参照本发明实施例的方法、装置（系统）和计算机程序产品的流程图和/或框图描述本发明。应当理解，流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合，都可以由计算机程序指令实现。这些计算机程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器，从而生产出一种机器，这些计算机程序指令通过计算机或其它可编程数据处理装置执行，产生了实现流程图和/或框图中的方框中规定的功能/操作的装置。

[0031] 也可以把这些计算机程序指令存储在能使得计算机或其它可编程数据处理装置以特定方式工作的计算机可读介质中，这样，存储在计算机可读介质中的指令就产生出一个包括实现流程图和/或框图中的方框中规定的功能/操作的指令装置（instruction means）的制造品（manufacture）。

[0032] 也可以把计算机程序指令加载到计算机、其它可编程数据处理装置、或其它设备上，使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤，以产生计算机实现的过程，从而使得在计算机或其它可编程装置上执行的指令能够提供实现流程图和/或框图中的方框中规定的功能/操作的过程。

[0033] 现在参看图2，该图示出本发明所针对的UEFI预启动环境的框图，也就是UEFI固件的框图，本说明书中，UEFI预启动环境与UEFI固件是同义词，不加区分。UEFI预启动环境主要包括UEFI执行环境模块，该模块进一步包括安全模块，初始化模块，驱动器执行环境模块、启动驱动器选择模块以及系统加载模块或者兼容性支持模块。当UEFI的系统加载模块加载支持并利用UEFI的操作系统时，由于此类操作系统中加入了对UEFI规范定义的某些标准接口的支持，如果碰到问题不能成功加载操作系统，该系统加载模块可以直接回到UEFI预启动环境。但是当UEFI的兼容性支持模块加载并启动遗留操作系统时，一旦UEFI进入系统兼容性模块进行遗留操作系统的启动尝试，即使遗留操作系统的启动尝试失败，也没有办法回到UEFI预启动环境，这样，系统启动人员无法进行问题的诊断。

[0034] UEFI固件与操作系统都是运行于CPU硬件上的软件，UEFI固件是先于操作系统运行的一段代码，负责进行操作系统启动之前的硬件初始化和可供操作系统使用的服务接口的准备。CPU系统存在多种模式，包括实模式、保护模式、虚拟8086模式以及系统管理模式（System Management Mode），各种模式间可以互相切换。CPU上电后一般运行在实模式下，CPU使用寄存器通知CPU指令在内存中的位置；使用DS、ES、FS、GS、SS等寄存器用于指示不同用途的数据段在内存中的位置，来指示下一个程序运行所需的主要内容。保护模式和实模式一样，保护模式下程序运行的实质仍是“CPU执行指令操作相关数据”，因此实模式下的各种代码段、数据段、堆栈段、中断服务程序仍然存在，且功能、作用不变。UEFI固件在CPU上电之后运行在实模式下，UEFI中还包括运行在CPU的保护模式下的代码以及运行在CPU的系统管理模式下的代码。当UEFI进入系统兼容性模块进行遗留操作系统的启动尝试时，UEFI的代码正运行在CPU的保护模式，如果启动尝试失败需要返回UEFI预启动环境，也先要返回CPU的保护模式。系统管理模式是一种特殊的处理器模式，这种模式可以提供某些系统级的功能，比如电源管理、系统硬件控制等。系统管理模式的主要好处是它能提供和普通CPU操作环境截然不同并且独立的处理器环境。这种环境不容易被操作系统或应用软件察觉。在CPU进入系统管理模式的时候，CPU会把CPU的寄存器值等一部分CPU执行

上下文信息保存在一块系统管理模式内存 (SMM RAM) 中，即所谓的 CPU 保存状态区域 (CPU Save State Region)。当退出系统管理模式时，处理器会利用处理器保存状态区域中的一部分 CPU 执行上下文来恢复进入系统管理模式之前的处理器状态。UEFI 固件可以利用上述退出系统管理模式时恢复进入系统管理模式之前的处理器状态这一技术特点，来回到 UEFI 预执行环境。

[0035] 对于支持 UEFI 的操作系统，因为系统加载模块是 UEFI 直接调用的，所以系统加载模块如果失败的话就会直接返回到 UEFI 预执行环境。操作在没有加载成功之前是不会破坏 UEFI 预启动环境上下文的。操作系统启动成功之后就没法直接回到 UEFI 预启动环境了。

[0036] 兼容性支持模块可以启动遗留操作系统，是由于兼容性支持模块在 UEFI 固件中提供了原本存在于传统 BIOS 中的遗留操作系统或遗留 Option ROM 所必需的一些接口的实现，其中 Option ROM 是控制各种板卡的一种固件，它可以存放在板卡上也可以包含在 UEFI 固件或 BIOS 固件中。但是一旦 UEFI 固件进入了兼容性支持模块做遗留操作系统启动，就会破坏 UEFI 预启动环境上下文，因此，即使遗留操作系统的启动尝试失败，由于 UEFI 预启动环境上下文被破坏，就没有办法回到 UEFI 预启动环境。

[0037] 因此，要想回到 UEFI 预启动环境，很重要的一点是需要保证 UEFI 预启动环境的完整性，即保存 UEFI 预启动环境上下文中的必要信息。另外一点就是利用上述 CPU 退出系统管理模式时恢复进入系统管理模式之前的处理器状态的特点。

[0038] 因此，本发明提供了一种从遗留操作系统环境恢复到 UEFI 预启动环境的方法和系统。在支持多操作系统的环境里面，可以通过这种方法和系统来支持遗留操作系统启动失败时的诊断以及随后的 UEFI 操作系统的启动尝试。

[0039] 图 3 示出了根据本发明的一种实施方式的从遗留操作系统环境恢复到 UEFI 预启动环境的方法的步骤，根据图 3 ：

[0040] 在步骤 S301，在 UEFI 预启动环境下，存储 UEFI 预启动环境中需要保存的上下文，所述 UEFI 预启动环境中需要保存的上下文包括 CPU 执行上下文 ；

[0041] 在步骤 S302，响应于 UEFI 预启动环境加载遗留操作系统失败，恢复所述 CPU 执行上下文的第一部分 ；

[0042] 在步骤 S303，使 UEFI 预启动环境相关联的 CPU 进入系统管理模式，并在系统管理模式下恢复所述 CPU 执行上下文的第二部分 ；以及 ；

[0043] 在步骤 S304，退出 CPU 系统管理模式，从而返回到 UEFI 预启动环境。

[0044] UEFI 预启动环境上下文中需要保存的上下文包括 CPU 执行上下文。具体 CPU 执行上下文可以分为两个部分，第一部分包括全局描述符表 (GDT)、中断描述符表 (IDT)、控制寄存器、以及段寄存器 ；第二部分包括标志寄存器，指令指针寄存器，通用寄存器，系统管理基址寄存器。因为 CPU 执行上下文的第一部分恢复后，CPU 才可以进入系统管理模式 ；而 CPU 执行上下文的第二部分必须在系统管理模式下恢复，这样才能够使得 CPU 退出系统管理模式时，CPU 利用该 CPU 执行上下文来恢复进入系统管理模式之前的处理器状态，从而恢复到 UEFI 预启动环境。

[0045] CPU 的寄存器是软件跟 CPU 硬件进行沟通的软硬件接口，需要保存和恢复的 CPU（以 64 位 CPU 为例）寄存器值有 ：标志寄存器（如 RFLAGS）、指令指针寄存器（如 RIP）、通用寄存器（如 RDI，RSI，RBP，RSP，RBX，RDX，RCX，RAX，R8 ～ R15）、系统管理基址寄存器

(SMBASE)、控制寄存器（如 CR0、CR3)、段寄存器（如 GS，FS，DS，SS，CS，ES)。可以通过修改 CPU 保存状态区域恢复的寄存器包括：标志寄存器，指令指针寄存器，通用寄存器，系统管理基址寄存器。段寄存器可以通过 MOV 或者 POP 指令进行行恢复，控制寄存器可以通过 MOV 指令进行恢复。全局描述符表位于内存区域，里面存放着全局数据结构（如段描述符），GDTR 寄存器指向存放全局描述符表的内存区域，可以通过 LGDT/SGDT 指令来加载或者保存全局描述符表的地址；中断描述符表位于内存区域，里面存放着中断描述符，IDTR 寄存器指向存放全局描述符表的内存区域，可以通过 LIDT/SIDT 指令来加载或者保存中断描述符表的地址。这里指令指针寄存器用于指向当前执行代码的内存地址。

[0046] 上述的方法可以退出 CPU 系统管理模式，从而返回到原来的运行环境，也就是 UEFI 预启动环境，但是并不一定是原来的 UEFI 预启动环境，主要是因为回到原来的 UEFI 预启动环境的另一部分上下文信息不知道是否被破坏。因此，UEFI 预启动环境上下文中需要保存的上下文还包括 UEFI 代码和数据所在的内存区域的地址以及该内存区域的内容，因为这些地址和内容是 UEFI 原来运行的环境，这些信息可以使得系统直接回到原来 UEFI 运行的环境。UEFI 固件的实现中必须提供 UEFI 规范定义的 UEFI 启动服务例程，其中的 GetMemorymap() 方法可以用来得到内存区域表 (memory map)。该方法会返回一个包含内存描述符的数组，每个内存描述符会指出对应内存区域的类型，起始地址，大小，以及属性。通过内存描述符里面的起始地址和大小就可以确定需要保存的内存区域的地址，从而获得该内存区域的内容。

[0047] 图 4 示出了 UEFI 固件定义的内存区域的类型，其中，图 4 的上半部分为 UEFI 预启动环境中需要保存的内存区域的类型。其中，内存类型描述如下：

[0048] EfiReservedMemoryType- 保留内存类型，这种类型的内存区域受 UEFI 固件和操作系统保护；

[0049] EfiLoaderCode，EfiLoaderData- 用于存放 UEFI 加载器代码和数据的内存区域类型；

[0050] EfiBootServicesCode，EfiBootServicesData- 用于存放启动服务驱动程序的代码和数据的内存区域类型；

[0051] EfiRuntimeServicesCode，EfiRuntimeServicesData- 用于存放运行时服务驱动程序的代码和数据的内存区域类型；

[0052] EfiConventionalMemory- 可用常规内存区域类型；

[0053] EfiUnusableMemory- 发现存在错误而导致不可用的内存区域类型；

[0054] EfiACPIReclaimMemory- 用于存放高级配置与电源接口表 (ACPI Table) 的内存区域类型；

[0055] EfiACPIMemoryNVS- 保留给固件使用的内存区域类型；

[0056] EfiMemoryMappedIO，- 用于建立内存映射输入输出的内存区域类型；

[0057] EfiMemoryMappedIOPortSpace- 用于内存映射输入输出端口的内存区域类型；

[0058] EfiPalCode- 用于存放处理器抽象层代码的内存类型；

[0059] 在一种实施方式中，为了使得系统直接回到原来 UEFI 运行的环境，图 3 所示的方法还包括响应于 UEFI 预启动环境加载遗留操作系统失败，将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内

存区域的地址。但是，这种恢复方式存在内存区域同时被操作系统访问从而破坏 UEFI 预启动环境上下文完整性的风险。在进一步的实施方式中，该方法还包括响应于 UEFI 预启动环境加载遗留操作系统失败，在将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址后，关闭 PCI 设备的内存映射输入输出 (MMIO)。这样可以防止直接内存访问设备 (DMA) 对内存的访问破坏恢复区域，从而减少 UEFI 预启动环境上下文完整性被破坏的风险。关闭步骤可通过清除 PCI 设备配置空间中的基址寄存器来实现。

[0060] 在另外一种实施方式中，为了使得系统直接回到原来 UEFI 运行的环境，图 3 所示的方法还包括响应于 CPU 进入系统管理模式，将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址。在进一步的实施方式中，该方法还包括响应于 CPU 进入系统管理模式，在将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址后，关闭 PCI 设备的内存映射输入输出 (MMIO)。这样可以防止直接内存访问设备 (DMA) 对内存的访问破坏恢复区域。关闭步骤可通过清除 PCI 设备配置空间中的基址寄存器来实现。

[0061] UEFI 预启动环境中需要保存的上下文可以保存在任何合适的存储区域，只要该存储区域所存储的数据在 UEFI 预启动环境以及遗留操作系统环境都不被破坏即可。在一种简单的实施方式中，可以采用外部存储（一般是一个 USB 存储介质、光盘、移动硬盘或 floppy 软盘等存储介质）来存储 UEFI 预启动环境中需要保存的上下文，这些存储方式存储的内容都可以通过程序控制，不被 UEFI 预启动环境以及遗留操作系统环境破坏，但是，恢复时需要的时间较长，恢复也较为复杂，例如恢复时需要运行能驱动该设备的代码，而且保存在外部存储设备上的上下文内容如没有特殊的保护机制（例如存放于隐藏分区）的话，还存在被破坏的可能性。

[0062] 在另外一种实施方式中，可以采用 UEFI 保留内存来存储 UEFI 预启动环境中需要保存的上下文。这种 EFI 保留内存的内存区域的完整性无论在 UEFI 预启动环境还是遗留操作系统环境下都可以得到保证。可以在进入 UEFI 预启动环境后的任何时刻保存 UEFI 预启动环境中需要保存的上下文，只要在 UEFI 进入兼容支持模块进行遗留操作系统启动的调用之前即可。在优选的实施方式中，还要首先调用 UEFI 系统服务来分配足够的保留内存，从而把上述 UEFI 预启动环境中需要保存的上下文保存在所述分配的保留内存中。

[0063] 在一种实施方式中，该方法还响应于 CPU 进入系统管理模式，对保存的指令指针寄存器进行修改，使所述指令指针寄存器指向进行遗留操作系统启动的调用指令的下一条指令，这样才能避免再次调用遗留操作系统，又再次回到 UEFI 预启动环境，成为一个恶性循环。

[0064] 在步骤 S302，响应于 UEFI 预启动环境加载遗留操作系统失败，恢复所述 CPU 执行上下文的第一部分，这里就如何确定 UEFI 预启动环境加载遗留操作系统失败提供一些实施方式。

[0065] 在一种实施方式中，兼容性支持模块在启动遗留操作系统的加载器时，响应于启动失败，遗留操作系统的加载器一般会调用软件中断（如 INT18）来通知 UEFI 固件。从 UEFI 固件的角度，就是响应于从遗留操作系统的加载器接收到软件中断，就可以确定为 UEFI 预

启动环境加载遗留操作系统失败。确定为失败后如何使 UEFI 预启动环境相关联的 CPU 进入系统管理模式可以利用中断处理函数来恢复所述 CPU 执行上下文的第一部分和 / 或 UEFI 代码和数据所在的内存区域的地址、该内存区域的内容。

[0066] 在另外一种实施方式中，可以在 UEFI 固件中加入看门狗时钟，用来侦测遗留操作系统启动期间的挂起 (system hang) 状况。看门狗时钟被预设成规定时间，UEFI 预启动环境在看门狗时钟预设的规定时间中没有接收到遗留操作系统启动成功的通知，看门狗时钟触发一个硬件中断，根据该硬件中断的触发确定 UEFI 预启动环境加载遗留操作系统失败。相反，如果 UEFI 预启动环境加载遗留操作系统成功，UEFI 预启动环境在看门狗时钟预设的规定时间中会接收到遗留操作系统启动成功的通知，这时 UEFI 固件可以将看门狗时钟停止或者预设成无限长时间。具体实施中可以在中断处理函数来恢复所述 CPU 执行上下文的第一部分和 / 或 UEFI 代码和数据所在的内存区域的地址、该内存区域的内容。

[0067] 在又一种实施方式中，可以利用机器检查异常机制。机器检查异常 (Machine-Check Exception) 是 CPU 提供的一种用来侦测和报告硬件错误的机制。是 PC 系统里面优先级最高的异常，其他中断 / 异常都不能打断它。机器检查异常中，例如对于 X86 的 CPU 可以通过 IA32_MCi_STATUS 寄存器来报告发生的错误的状态，其中的两个比特可以用来检查当前 CPU 执行上下文是否被破坏 :Valid(bit 63) ＝ 1 且 PCC(bit 57) ＝ 0，则代表上下文未被破坏，如果 CPU 执行上下文没有被破坏，就可以在机器检查异常的处理函数中来恢复所述 CPU 执行上下文的第一部分和 / 或 UEFI 代码和数据所在的内存区域的地址、该内存区域的内容。在进一步的实施方式中，在异常处理函数种需要首先判断所述 CPU 执行上下文是否被破坏，如果 CPU 执行上下文没有被破坏的话，就可以在错误被纠正之后恢复 CPU 执行上下文的第一部分和 / 或 UEFI 代码和数据所在的内存区域的地址、该内存区域的内容 ;如果 CPU 执行上下文已经被破坏了的话，就只能重启系统了。

[0068] 恢复所述 CPU 执行上下文的第一部分包括恢复全局描述符表 (GDT)、中断描述符表 (IDT)、控制寄存器以及段寄存器，可以使用如下伪码 :

[0069] MemoryCopy(ORIGINAL_GDT_ADDRESS, SAVED_GDT_ADDRESS, sizeof(GDT))// 恢复全局描述符表

[0070] LGDT ORIGINAL_GDT_ADDRESS

[0071] MemoryCopy(ORIGINAL_IDT_ADDRESS, SAVED_IDT_ADDRESS, sizeof(IDT))// 恢复中断描述符表

[0072] LIDT ORIGINAL_IDT_ADDRESS

[0073] MOV CONTROL_REGISTER, SAVED_CR_VALUE// 恢复控制寄存器

[0074] FARJMP ADDRESS_OF_NEXT_INSTRUCTION// 远跳转到下一条指令，改变执行流程并序列化处理器

[0075] MOV SEGMENT_REGISTER, // 恢复段寄存器

[0076] SAVED_SEGMENT_REGISTER_VALUE

[0077] 恢复 UEFI 代码和数据所在的内存区域的地址、该内存区域的内容可以利用存储 UEFI 预启动环境中需要保存的上下文将 UEFI 代码和数据所在的内存区域的内容拷贝到 UEFI 代码和数据所在的内存区域的地址，当采用保留内存存储 UEFI 预启动环境中需要保存的上下文时，是一种内存对内存的拷贝方式。

[0078] 步骤 S303，使 UEFI 预启动环境相关联的 CPU 进入系统管理模式，并在系统管理模式下恢复所述 CPU 执行上下文的第二部分。恢复 CPU 执行上下文的第二部分包括恢复标志寄存器，指令指针寄存器，通用寄存器，系统管理基址寄存器，可以使用与恢复 CPU 执行上下文的第一部分类似的伪码。使 UEFI 预启动环境相关联的 CPU 进入系统管理模式可以使用硬件触发的方式，也可以采用触发软件系统管理中断 (software SMI) 的方式。在使 UEFI 预启动环境相关联的 CPU 进入系统管理模式前，首先要使 CPU 切换到保护模式，然后触发软件系统管理中断 (software SMI)，这样才能进入系统管理模式

[0079] 在步骤 S304 退出 CPU 系统管理模式，从而返回到 UEFI 预启动环境，其中，退出 CPU 系统管理模式可以通过 CPU 执行 RSM 指令来实现，就会回到一开始就保存的 UEFI 预启动执行环境，从而使得尝试下一个 UEFI 启动选项或者进行 UEFI 系统诊断成为可能。另外，由于系统中的控制器在做遗留系统启动尝试的时候都被 UEFI 执行了断开 (disconnect) 的动作，所以在恢复 UEFI 预启动执行环境之后，响应于返回到 UEFI 预启动环境，使 UEFI 控制的控制器的驱动程序重新运行，这样能够恢复 UEFI 对控制器的控制。

[0080] 在同一个发明构思下，本发明还公开了一种从遗留操作系统环境恢复到 UEFI 预启动环境的系统，图 5 示出了该从遗留操作系统环境恢复到 UEFI 预启动环境的系统结构框图，根据图 5，该系统包括 ：存储装置 501，被配置为在 UEFI 预启动环境下，存储 UEFI 预启动环境中需要保存的上下文，所述 UEFI 预启动环境中需要保存的上下文包括 CPU 执行上下文 ；第一恢复装置 502，被配置为响应于 UEFI 预启动环境加载遗留操作系统失败，恢复所述 CPU 执行上下文的第一部分 ；第二恢复装置 503，被配置为使 UEFI 预启动环境相关联的 CPU 进入系统管理模式，并在系统管理模式下恢复所述 CPU 执行上下文的第二部分 ；以及退出装置 504，被配置为退出 CPU 系统管理模式，从而返回到 UEFI 预启动环境。

[0081] 在一种实施方式中，所述 UEFI 预启动环境上下文中需要保存的上下文还包括 UEFI 代码和数据所在的内存区域的地址以及该内存区域的内容。优选地，所述存储装置是采用 UEFI 保留内存来存储 UEFI 预启动环境中需要保存的上下文。

[0082] 在一种实施方式中，第一恢复装置使用如下确定装置之一确定 UEFI 预启动环境加载遗留操作系统失败 ：第一确定装置，被配置为响应于从遗留操作系统的加载器接收到软件中断，确定 UEFI 预启动环境加载遗留操作系统失败。第二确定装置，被配置为在 UEFI 固件中加入看门狗时钟，UEFI 预启动环境在看门狗时钟预设的规定时间中没有接收到遗留操作系统启动成功的通知，看门狗时钟触发一个硬件中断，根据该硬件中断的触发确定 UEFI 预启动环境加载遗留操作系统失败。第三确定装置，被配置为利用机器检查异常机制确定 UEFI 预启动环境加载遗留操作系统失败。

[0083] 在一种实施方式中，第一恢复装置还被配置为响应于 UEFI 预启动环境加载遗留操作系统失败，将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址。优选地，第一恢复装置还被配置为在将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址后，关闭 PCI 设备的内存映射输入输出。

[0084] 在另一种实施方式中，第二恢复装置还被配置为响应于 CPU 进入系统管理模式，将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢

复到 UEFI 代码和数据所在的内存区域的地址。优选地，第二恢复装置还被配置为在将所述 UEFI 预启动环境中需要保存的上下文中 UEFI 代码和数据所在的内存区域的内容恢复到 UEFI 代码和数据所在的内存区域的地址后，关闭 PCI 设备的内存映射输入输出。优选地，第二恢复装置还被配置为响应于 CPU 进入系统管理模式，对保存的指令指针寄存器进行修改，使所述指令指针寄存器指向进行遗留操作系统启动的调用指令的下一条指令。

[0085] 在一种实施方式中，退出装置还被配置为响应于返回到 UEFI 预启动环境，使 UEFI 控制的控制器的驱动程序重新运行。

[0086] 附图中的流程图和框图显示了根据本发明的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段或代码的一部分，所述模块、程序段或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个连续的方框实际上可以基本并行地执行，它们有时也可以按相反的顺序执行，这依所涉及的功能而定。也要注意的是，框图和 / 或流程图中的每个方框、以及框图和 / 或流程图中的方框的组合，可以用执行规定的功能或操作的专用的基于硬件的系统来实现，或者可以用专用硬件与计算机指令的组合来实现。

[0087] 以上已经描述了本发明的各实施例，上述说明是示例性的，并非穷尽性的，并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围和精神的情况下，对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择，旨在最好地解释各实施例的原理、实际应用或对市场中的技术改进，或者使本技术领域的其它普通技术人员能理解本文披露的各实施例。

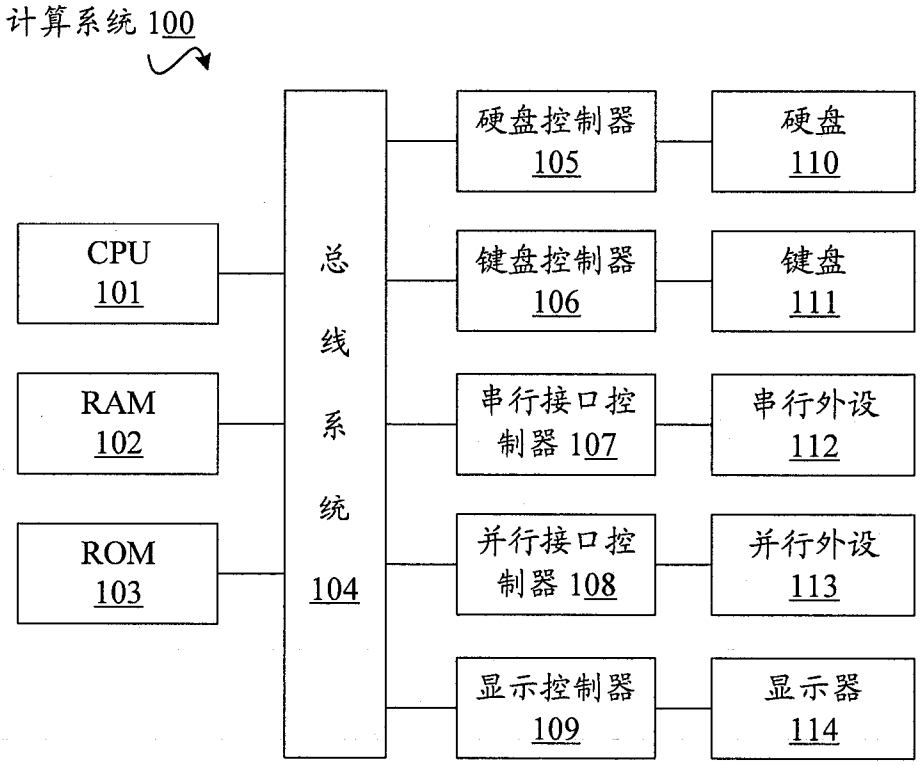

计算系统 100
CPU 101
RAM 102
ROM 103
总线系统 104
硬盘控制器 105
键盘控制器 106
串行接口控制器 107
并行接口控制器 108
显示控制器 109
硬盘 110
键盘 111
串行外设 112
并行外设 113
显示器 114

图 1

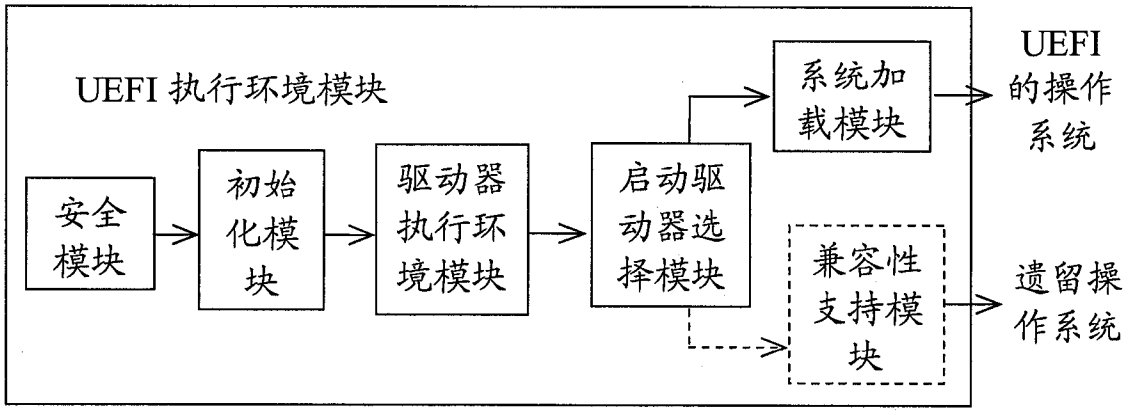

UEFI 执行环境模块
安全模块
初始化模块
驱动器执行环境模块
启动驱动器选择模块
系统加载模块
兼容性支持模块
UEFI 的操作系统
遗留操作系统

图 2

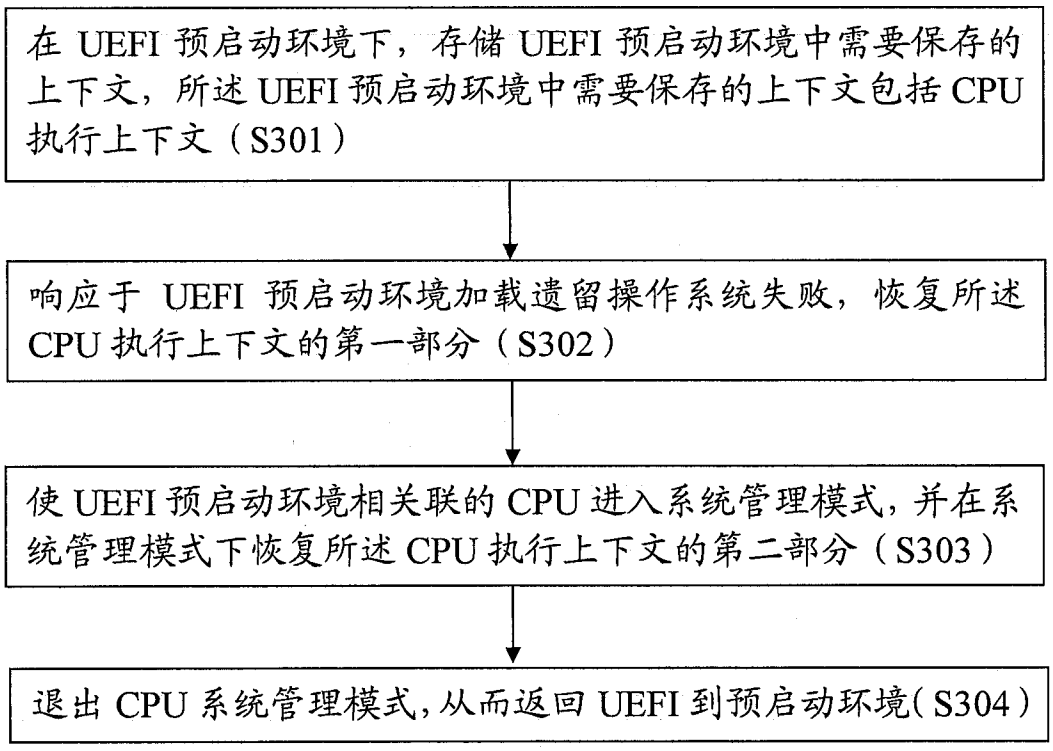

在 UEFI 预启动环境下，存储 UEFI 预启动环境中需要保存的上下文，所述 UEFI 预启动环境中需要保存的上下文包括 CPU 执行上下文（S301）
响应于 UEFI 预启动环境加载遗留操作系统失败，恢复所述 CPU 执行上下文的第一部分（S302）
使 UEFI 预启动环境相关联的 CPU 进入系统管理模式，并在系统管理模式下恢复所述 CPU 执行上下文的第二部分（S303）
退出 CPU 系统管理模式，从而返回 UEFI 到预启动环境（S304）

图 3

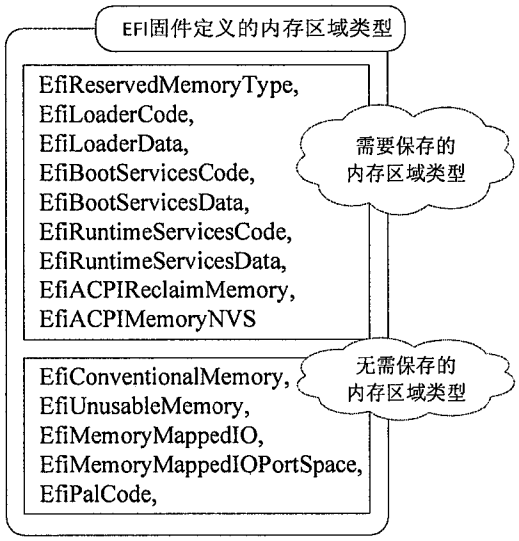

EFI固件定义的内存区域类型
EfiReservedMemoryType,
EfiLoaderCode,
EfiLoaderData,
EfiBootServicesCode,
EfiBootServicesData,
EfiRuntimeServicesCode,
EfiRuntimeServicesData,
EfiACPIReclaimMemory,
EfiACPIMemoryNVS
需要保存的内存区域类型
EfiConventionalMemory,
EfiUnusableMemory,
EfiMemoryMappedIO,
EfiMemoryMappedIOPortSpace,
EfiPalCode,
无需保存的内存区域类型

图 4

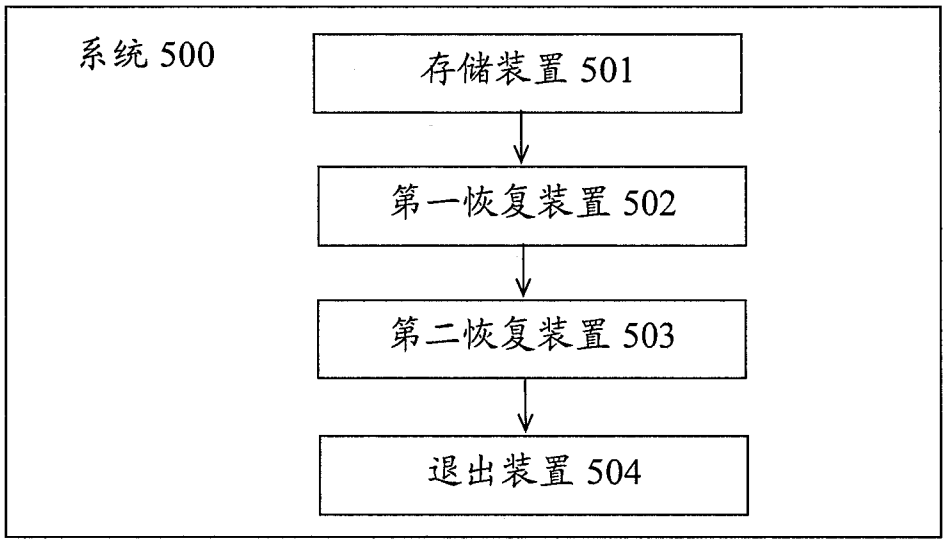


图 5