



(10) 申请公布号 CN 104424140 A

(21) 申请号 201410415065.2

(30) 优先权数据

61/871,618 2013.08.29 US

14/149,342 2014.01.07 US

(71) 申请人 三星电子株式会社

地址 韩国京畿道

(72)发明人 P. 比什特

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 邵亚丽

(51) Int. Cl.

G06F 13/20(2006.01)

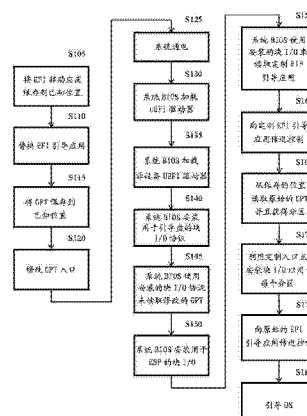
权利要求书2页 说明书5页 附图2页

(54) 发明名称

统一可扩展固件接口驱动器

(57) 摘要

加载和运行不依赖设备并且系统无关的统一可扩展固件接口 (UEFI) 驱动器, 该 UEFI 驱动器被配置为过滤用于存储设备的输入 / 输出 (I/O) 而不需要取决于外围部件互连 (PCI) 类型设备和 / 或修改系统 UEFI 基本输入 / 输出系统 (BIOS), 从而使能支持操作系统 (OS) 的引导的纯软件产品。



1. 一种通过计算机处理器加载定制的统一可扩展固件接口 (UEFI) 驱动器的方法, 所述方法包括:

将存储在第一位位置处的原始的可扩展固件接口 (EFI) 引导应用复制到第二位置;
利用定制的 EFI 引导应用替换第一位位置处的原始的 EFI 引导应用;
将存储在第三位位置处的原始的全局唯一标识符 (GUID) 分区表 (GPT) 复制到第四位置;
以及

利用定制的 GPT 替换第三位位置处的原始的 GPT。

2. 如权利要求 1 所述的方法, 其中所述定制的 GPT 包括指向 EFI 系统分区 (ESP) 的 ESP 分区入口, 该 ESP 包括定制的 EFI 引导应用, 以及

其中所述定制的 EFI 引导应用被配置为致使安装用于具有入口点的 ESP 的块输入 / 输出 (I/O) 协议, 通过该入口点过滤用于原始的 GPT 的操作系统 (OS) 分区的 I/O。

3. 如权利要求 2 所述的方法, 其中所述定制的 GPT 省略 OS 分区入口。

4. 如权利要求 2 所述的方法, 其中所述定制的 EFI 引导应用进一步被配置为一旦安装了用于 ESP 的块 I/O 协议, 就致使运行原始的 EFI 引导应用。

5. 如权利要求 4 所述的方法, 其中所述原始的 EFI 引导应用加载 OS。

6. 如权利要求 4 所述的方法, 其中所述原始的 EFI 引导应用被配置为致使安装用于原始的 GPT 中的 OS 分区入口的块 I/O 协议。

7. 一种加载定制的统一可扩展固件接口 (UEFI) 驱动器的计算机处理器方法, 所述方法包括:

将存储在第一位位置处的原始的可扩展固件接口 (EFI) 引导应用复制到第二位置;
利用定制的 EFI 引导应用替换第一位位置处的原始的 EFI 引导应用;
将存储在第三位位置处的原始的全局唯一标识符 (GUID) 分区表 (GPT) 复制到第四位置;
以及

利用定制的 GPT 替换第三位位置处的原始的 GPT。

8. 如权利要求 7 所述的方法, 其中所述定制的 GPT 包括指向 EFI 系统分区 (ESP) 的 ESP 分区入口, 该 ESP 包括定制的 EFI 引导应用, 以及

其中所述定制的 EFI 引导应用被配置为致使安装用于具有入口点的 ESP 的块输入 / 输出 (I/O) 协议, 通过该入口点过滤用于原始的 GPT 的操作系统 (OS) 分区的 I/O。

9. 如权利要求 8 所述的方法, 其中所述定制的 GPT 省略 OS 分区入口。

10. 如权利要求 8 所述的方法, 其中所述定制的 EFI 引导应用进一步被配置为一旦安装了用于 ESP 的块 I/O 协议, 就致使运行原始的 EFI 引导应用。

11. 如权利要求 10 所述的方法, 其中所述原始的 EFI 引导应用加载 OS。

12. 如权利要求 10 所述的方法, 其中所述原始的 EFI 引导应用被配置为致使安装用于原始的 GPT 中的 OS 分区入口的块 I/O 协议。

13. 一种定制的统一可扩展固件接口 (UEFI) 驱动器, 所述定制的 UEFI 驱动器包括:

定制的全局唯一标识符 (GUID) 分区表 (GPT); 以及

定制的可扩展固件接口 (EFI) 引导应用,

其中所述定制的 GPT 指向存储定制的 EFI 引导应用的 EFI 系统分区 (ESP), 以及

其中所述定制的 EFI 引导应用被配置为致使安装用于具有入口点的 ESP 的块输入 / 输

出(I/O)协议,通过该入口点过滤用于原始的 GPT 的操作系统(OS)分区的 I/O。

14. 如权利要求 13 所述的定制的 UEFI 驱动器,其中所述定制的 GPT 包括指向 EFI 系统分区(ESP)的 ESP 分区入口。

15. 如权利要求 14 所述的定制的 UEFI 驱动器,其中所述定制的 GPT 省略 OS 分区入口。

16. 如权利要求 14 所述的定制的 UEFI 驱动器,其中所述定制的 EFI 引导应用进一步被配置为一旦安装了用于 ESP 的块 I/O 协议,就致使运行原始的 EFI 引导应用。

17. 如权利要求 16 所述的定制的 UEFI 驱动器,其中所述原始的 EFI 引导应用加载 OS。

18. 如权利要求 16 所述的定制的 UEFI 驱动器,其中所述原始的 EFI 引导应用被配置为致使安装用于原始的 GPT 中的 OS 分区入口的块 I/O 协议。

统一可扩展固件接口驱动器

[0001] 相关申请的交叉引用

[0002] 本申请要求于 2013 年 8 月 29 日在美国专利与商标局提交的美国临时申请第 61/871,618 号的权益,其公开内容通过引用全面合并于此。

技术领域

[0003] 与示例性实施例一致的方法和程序涉及一种统一可扩展固件接口 (Unified Extensible Firmware Interface, UEFI),并且更具体地,涉及加载和运行不依赖设备并且系统无关的 UEFI 驱动器,该 UEFI 驱动器被配置为过滤用于存储设备的输入 / 输出 (I/O) 而不需要取决于外围部件互连 (PCI) 类型设备和 / 或修改系统 UEFI 基本输入 / 输出系统 (BIOS),从而使能支持操作系统 (OS) 的引导的纯软件产品。

背景技术

[0004] 通常,存在用于在 UEFI 环境中加载 UEFI 驱动器的两种方法。

[0005] 在第一方法中,PCI 设备采用存在于 PCI 设备上的 ROM/ 闪存存储器中的选项 - 只读存储器 (选项 -ROM)UEFI 驱动器。选项 -ROM UEFI 驱动器为 PCI 设备制造商专有,可以闪存化为 PCI 设备 ROM,并且不能由第三方软件修改。

[0006] 此选项 -ROM UEFI 驱动器的目的是在 OS 运行前 (pre-OS) 环境期间访问 PCI 设备和它的子设备。例如,基于 PCI 的存储控制器和连接到基于 PCI 的存储控制器的盘由使用 UEFI 驱动器 BIOS 的系统 UEFI BIOS 访问。因此,PCI 设备需要选项 -ROMBIOS 来引导 OS。PCI 设备加载选项 -ROMBIOS 的这个需要使得诸如“回写高速缓存”软件或 RAID 软件之类的纯软件产品不可能支持引导 OS,因为这些软件只能在 OS 加载期间作为 OS 驱动器被加载。

[0007] 在第二方法中,将被加载到系统中的 UEFI 驱动器可以保存在 UEFI BIOS 非易失性随机存取存储器 (NVRAM) 中,并且系统 UEFI BIOS 可以在开始引导过程期间加载 UEFI 驱动器。这个 NVRAM 区域不能由第三方软件开发者使用,因为每个系统制造商具有他自己的专有格式。因此,试图写入此 NVRAM 的第三方软件可能造成系统无法修复的损害,并且修改 UEFI 驱动器使仅可以由系统制造商执行的系统 BIOS 修改成为必需。

发明内容

[0008] 示例性实施例的方面提供一种用于实现存储用于不依赖设备并且系统无关的 UEFI 驱动器的程序的计算机可读介质的方法。

[0009] 根据示例性实施例的一方面,提供一种加载定制的统一可扩展固件接口 (UEFI) 驱动器的方法,所述方法包括:计算机处理器将存储在第一位位置处的原始的可扩展固件接口 (EFI) 引导应用复制到第二位位置,利用定制的 EFI 引导应用替换第一位位置处的原始 EFI 引导应用,将第三位位置处的原始全局唯一标识符 (GUID) 分区表 (GPT) 复制到第四位置,以及利用定制的 GPT 替换第三位位置处的原始 GPT。

[0010] 定制的 GPT 可以包括指向 EFI 系统分区 (ESP) 的 ESP 分区入口,该 ESP 包括定制

的EFI引导应用,并且该定制的EFI引导应用可以被配置为致使安装用于具有入口点的ESP的块输入/输出(I/O)协议,用于原始的GPT的操作系统(OS)分区的I/O通过该入口点被过滤。

[0011] 所述定制的GPT可以省略OS分区入口。

[0012] 所述定制的EFI引导应用还可以被配置为一旦安装了用于ESP的块I/O协议就致使运行原始的EFI引导应用。

[0013] 该原始的EFI引导应用可以加载OS。

[0014] 该原始的EFI引导应用可以被配置为致使安装用于原始的GPT中的OS分区入口的块I/O协议。

[0015] 根据示例性实施例的一方面,提供一种在其上具体实现了定制的统一可扩展固件接口(UEFI)驱动器的计算机可读记录介质,该定制的UEFI驱动器包括计算机可读代码,该计算机可读代码被配置为致使计算机处理器将存储在第一位位置处的原始的可扩展固件接口(EFI)引导应用复制到第二位位置,利用定制的EFI引导应用替换第一位位置处的原始EFI引导应用,将第三位位置处的原始全局唯一标识符(GUID)分区表(GPT)复制到第四位置,以及利用定制的GPT替换第三位位置处的原始GPT。

[0016] 根据示例性实施例的一方面,提供一种在其上具体实现了定制的统一可扩展固件接口(UEFI)驱动器的计算机可读记录介质,所述定制的UEFI驱动器包括定制的全局唯一标识符(GUID)分区表(GPT)和定制的EFI引导应用,该定制的GPT指向存储定制的EFI引导应用的EFI系统分区(ESP),并且该定制的EFI引导应用被配置为安装用于具有入口点的ESP的块输入/输出(I/O)协议,用于操作系统(OS)分区的I/O通过该入口点被过滤。

附图说明

[0017] 从以下结合附图对示例性实施例的描述,示例性实施例的以上及其他方面将变得清楚和更加容易理解。附图中:

[0018] 图1示出根据示例性实施例的用于加载和运行UEFI驱动器的方法的流程图。

[0019] 图2示出根据示例性实施例的执行“回写高速缓存”的方法的流程图。

具体实施方式

[0020] 下文中,参考附图详细描述本发明构思的示例性实施例。

[0021] 在以下描述中,即使在不同的附图中,相同的附图标记也用于相同的元素。提供描述中定义的主题,如具体的构造和元件,是为了帮助综合理解示例性实施例。然而,能够在没有那些具体定义的细节的情况下实施本发明构思。此外,不详细描述公知的功能或构造,因为他们将以不必要的细节模糊本发明。

[0022] 除非另有陈述,否则如下所述的方法的操作可以以任一次序执行。可替换地,操作可以同时执行。

[0023] 图1示出用于加载和运行UEFI驱动器以过滤用于存储设备的I/O而不需要取决于PCI设备和/或修改系统UEFI BIOS的方法的流程图。通过加载定制的UEFI驱动器而无需PCI设备或系统UEFI BIOS修改使能用于支持引导OS的纯软件产品。

[0024] 图1的方法可以通过软件应用或模块的运行来执行以加载UEFI驱动器,该软件应

用或模块在计算机系统中运行。

[0025] 当从储存器盘引导 UEFI 系统时,系统 UEFI BIOS 加载总是存在于引导盘的扇区 0 处的盘的主引导记录 (MBR)。如果 MBR 指示其是全局唯一标识符 (GUID) 分区表 (GPT) 盘,则系统 BIOS 读取 GPT、列数盘上的分区、以及安装用于每个分区的块输入 / 输出 (I/O) 协议。块 I/O 协议是访问盘上的分区的接口。没有块 I/O 协议,不能访问盘上的分区。

[0026] 一旦块 I/O 协议被安装用于每个分区,系统 UEFI BIOS 就寻找包含 OS 引导管理器的 EFI 系统分区 (ESP) (例如,对于微软视窗,bootx64.efi 是位于...efi\microsoft\boot\bootmgfw.efi 的微软视窗引导管理器的副本)。系统 UEFI BIOS 使用块 I/O 协议访问 ESP,将 OS 引导管理器加载到存储器中并且向 OS 引导管理器传送控制,该 OS 引导管理器使用块 I/O 协议来访问 OS 分区 (例如 C:) 并且加载 OS 加载器 (例如,对于微软视窗,winload.efi),该 OS 加载器继续加载和运行其余 OS (例如,微软视窗)。

[0027] 根据图 1 中示出的示例性实施例,定制的 OS 引导管理器可以被提供,并且原始的 OS 引导管理器可以保存到存储器中。OS 引导管理器将被称为 EFI 引导应用。

[0028] 根据图 1 中示出的示例性实施例,还可以提供定制 GPT。原始的 GPT 可以被修改为仅保留 ESP 分区入口,并且全部其他分区入口可以从 GPT 除去,以获得定制 GPT。选择性地,OS 引导不需要的、与分区有关的入口也可以保留在定制 GPT 中。例如,OS 引导不需要的分区可以是恢复或备份分区。原始的 GPT 可以朝向盘的末端保存。

[0029] 如图 1 中示出的,在步骤 S105 中,当前 EFI 引导应用被保存到存储器,从而创建 EFI 引导应用的副本。在微软视窗 OS 中,EFI 引导应用可以是 bootx64.efi 或 bootmgfw.efi。

[0030] 在步骤 S110 中,当前 EFI 引导应用被替换为定制的 EFI 引导应用。当前 EFI 引导应用可以通过利用定制的 EFI 引导应用重写当前的 EFI 引导应用来替换,或通过编辑当前的 EFI 引导应用来替换。当被运行时,定制的 EFI 引导应用可以取得存储在存储器中的原始的 GPT 并且使用可以过滤 I/O 的定制入口点安装除 ESP 分区入口之外的、用于 GPT 中的全部有效分区入口 (例如, C:) 的块 I/O 协议。定制的 EFI 引导应用可以包括需要加载用于定制块 I/O 协议的 UEFI 驱动器入口点。一旦加载了用于全部有效分区入口的块 I/O 协议,定制的 EFI 引导应用就可以从存储器取得原始的 EFI 引导应用,加载原始的 EFI 引导应用,并且向原始的 EFI 引导应用传送控制以用于加载 OS。

[0031] 在步骤 S115 中, GPT 保存到存储器,从而创建 GPT 的副本。原始的副本可以朝向盘的末端保存。

[0032] 在步骤 S120 中,从 GPT 中除去了 ESP 入口之外的全部 GPT 入口。选择性地,除了 ESP 入口以外,OS 引导不需要的、与分区有关的入口也可以保留在 GPT 中。因此,修改后的 GPT 可以构成定制 GPT。

[0033] 此时,UEFI 驱动器被加载,并且系统准备好运行新的 UEFI 驱动器。

[0034] 在起动或重新引导期间,在步骤 S125 中,UEFI 系统从 GPT 盘引导。

[0035] 在步骤 S130 中,当从储存器盘引导 UEFI 系统时,系统 UEFI BIOS 加载总是存在于引导盘的第一扇区 (即,扇区 0) 处的盘的 MBR。

[0036] 如果 MBR 指示该盘是 GPT 盘,则在步骤 S135 中系统 BIOS 选择性地加载非设备 (non-device) 的 UEFI 驱动器。

[0037] 在步骤 S140 中,系统 BIOS 开始安装用于盘的分区的块 I/O 协议。

[0038] 在步骤 S145 中,系统 BIOS 读取作为步骤 S120 的定制的 GPT 的 GPT,并且列数通过 GPT 指示的盘上的分区,该分区可以仅是在步骤 S120 的定制 GPT 中的 ESP 分区入口。如上面讨论的,对 OS 引导不必要的其他分区入口可以被保留,诸如备份或恢复分区,并且因此这种分区入口也可以在步骤 S145 中读取。

[0039] 在步骤 S150 中,系统 UEFI BIOS 安装用于读取分区的块 I/O 协议,即用于 ESP 分区的块 I/O 协议。再次,对 OS 引导不必要的其他分区入口可以被保留,诸如备份或恢复分区,并且因此用于这种分区入口的块 I/O 协议也可以在步骤 S150 中安装。

[0040] 在步骤 S155 中,系统 BIOS 使用安装的块 I/O 协议从 ESP 分区读取 EFI 引导管理器。EFI 引导管理器是步骤 S110 的定制 EFI 引导管理器。

[0041] 在步骤 S160 中,系统 BIOS 向 EFI 引导管理器传送控制。

[0042] 如上所述,块 I/O 协议是用于访问盘上的分区的接口,没有该块 I/O 协议不能访问盘上的分区。因为从定制的 GPT 省略除了 ESP 分区入口之外的分区入口,所以块 I/O 协议仅被安装用于 ESP 分区。因此,在这一点上,盘显得原始而没有其他有效分区。

[0043] 在步骤 S165 中,EFI 引导应用取得原始的 GPT,该 GPT 保存在仅为步骤 S115 中的定制 EFI 引导应用所知的位置处。

[0044] 在步骤 S170 中,EFI 引导应用读取原始的 GPT 的分区入口,并且利用定制入口点来安装块 I/O 协议以用于相应于除了已经预先加载的 ESP 之外的分区(例如,类似 C: 的 OS 分区)的每个有效分区入口。因为没有安装用于这些分区的块 I/O 协议,所以直到现在这些新的加载分区对系统 UEFI BIOS 或任一其他 UEFI 应用是不可见的。

[0045] 在此点之后,因为随后安装用于每个分区(OS 分区,例如 C:\)的块 I/O 协议,所以指向这些分区的全部 I/O 经过用于步骤 S150 中加载的 ESP 的块 I/O 协议,该 ESP 包括定制入口点。这使能过滤用于这些新加载的分区的 I/O。

[0046] 在步骤 S175 中,一旦块 I/O 协议对于原始的 GPT 中的全部剩余有效分区被安装,EFI 引导应用就从存储器加载原始的 EFI 引导管理器,并且向加载 OS 加载器的原始的 EFI 引导管理器传送控制。OS 加载器在步骤 S180 中继续加载和运行 OS。

[0047] 现在可以过滤用于块 I/O 协议安装用于的 OS 分区的全部 I/O,以及其他分区。

[0048] 图 2 示出根据示例性实施例的、用于使用 UEFI 驱动器执行“回写高速缓存”的方法。

[0049] 根据示例性实施例的一方面,UEFI 驱动器的使用可以是“回写高速缓存”纯软件产品。这种软件需要存在于 OS 运行前 UEFI 环境中以用于支持引导 OS,因为一些数据可以存在于高速缓存器件上而且可以不存在于目标储存器盘(正在高速缓存的盘)上。

[0050] 传统方法采用基于 PCI 的存储控制器,在其上已经安装了选项 ROM UEFI。因为使用 UEFI 驱动器执行全部 I/O,所以基于 PCI 的存储控制器可以每当需要时将 I/O 重新指向高速缓存器件。但是,这需要高速缓存逻辑在基于 PCI 的存储控制器的 UEFI 驱动器之内,这对于第三方软件是不可能的。

[0051] 另一方面,根据示例性实施例的方面,通过使用加载到存储器中的单独的 UEFI 驱动器/应用,该 I/O 可以被过滤而无需修改专有的 UEFI 驱动器或系统 UEFI BIOS。

[0052] 根据示例性实施例的方面,此方法也在过滤用于任一 PCI 设备及其子设备的 I/O 方面占优势,而驻留在 PCI 器件上的 UEFI 驱动器可以仅访问指向特定 PCI 设备及其子设备

的 I/O。示例性实施例的方面对于像通过允许相关的储存器盘（例如，目标盘和高速缓存盘）在不同的存储控制器上的“回写高速缓存”、“软件 RAID”和“整盘加密”的纯软件产品更富有吸引力。

[0053] 在“回写高速缓存”的情况下，此方法可以进一步用于几乎完全地从高速缓存盘引导，从而除非绝对地需要，减轻目标储存器盘的旋转（spin）。在图 2 中示出“回写高速缓存”的方法。

[0054] 在步骤 S210 中，目标盘的 GPT 的副本保存在与目标盘相同的高速缓存盘上的位置处。高速缓存盘可以是高速盘，诸如固态硬盘（SSD）。

[0055] 在步骤 S220 中，除去目标储存器盘上的 GPT 的全部分区。

[0056] 在步骤 S230 中，将 ESP 分区从目标盘复制到高速缓存盘。

[0057] 在步骤 S240 中，将高速缓存盘设置为可引导盘。

[0058] 在步骤 S250 中，如上相对于图 1 所述，原始的 EFI 引导应用被保存在高速缓存盘上，并且高速缓存盘上的原始的 EFI 引导应用被替换为定制的 EFI 应用。

[0059] 根据此示例性实施例的方面，系统可以从高速缓存 SSD（而不是目标盘）引导，从高速缓存盘上的 ESP 加载定制 EFI 引导应用，在建立块 I/O 协议之后该高速缓存盘上的 ESP 加载原始的 EFI 引导应用，但是这次是从高速缓存盘自己。因此，避免目标盘的旋转直到在高速缓存盘上存在读取错误。

[0060] 类似方法可以被用于从类似休眠的睡眠状态重新开始（resume）。

[0061] 根据示例性实施例的方面，此方法提供与全部 OS 安装在高速缓存盘上的情况几乎相同的引导和从休眠重新开始的速度。

[0062] 根据示例性实施例的方面，描述了加载定制 UEFI 驱动器而无需 PCI 器件或系统 UEFI BIOS 修改的新方法。从而，使用支持引导和从休眠重新开始的 UEFI 的纯软件产品是可能的。

[0063] 根据示例性实施例的方面，UEFI 驱动器可以被加载而无需 PCI 设备依赖，并且 UEFI 驱动器可以被加载而无需系统 BIOS 修改。

[0064] 上述讨论到的方法的操作可以通过运行存储在计算机可读记录介质（例如，盘、存储器、光盘等等）上的计算机可读代码的计算机处理器（例如，中央处理单元、CPU）实现。计算机处理器可以是计算机的通用处理器、计算机的专用处理器、或存储控制器的处理器。计算机可读代码可以包括用于将 UEFI 驱动器加载到存储器的安装程序，并且可以包括加载到存储器的 UEFI 驱动器，当引导计算机时由计算机处理器运行该 UEFI 驱动器。

[0065] 尽管已经示出和描述了若干示例性实施例，但本领域技术人员将会理解，可以改变这些示例性实施例而不偏离本发明构思的原理和精神，本发明构思的范围在权利要求及其等效物中限定。

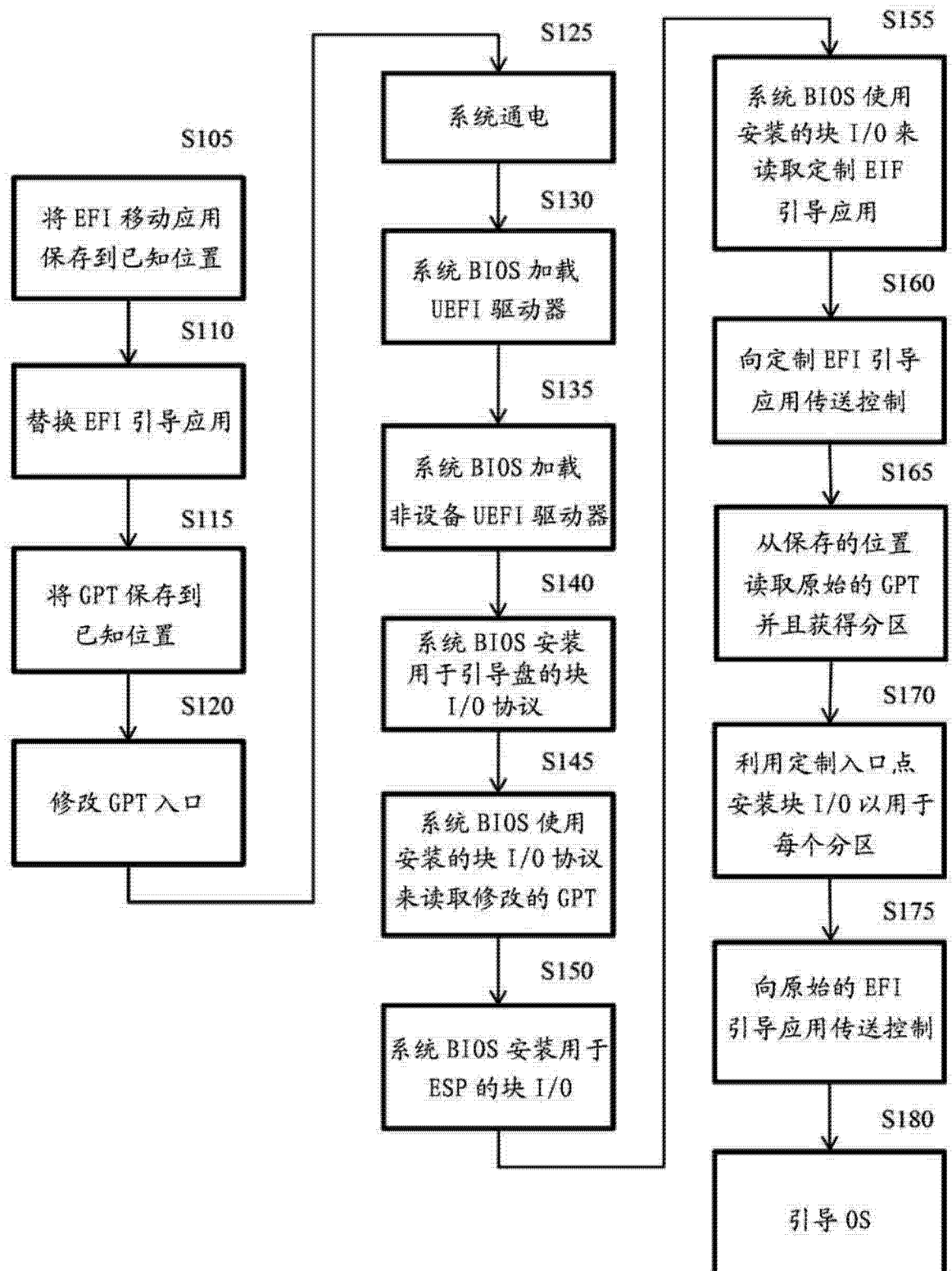


图 1

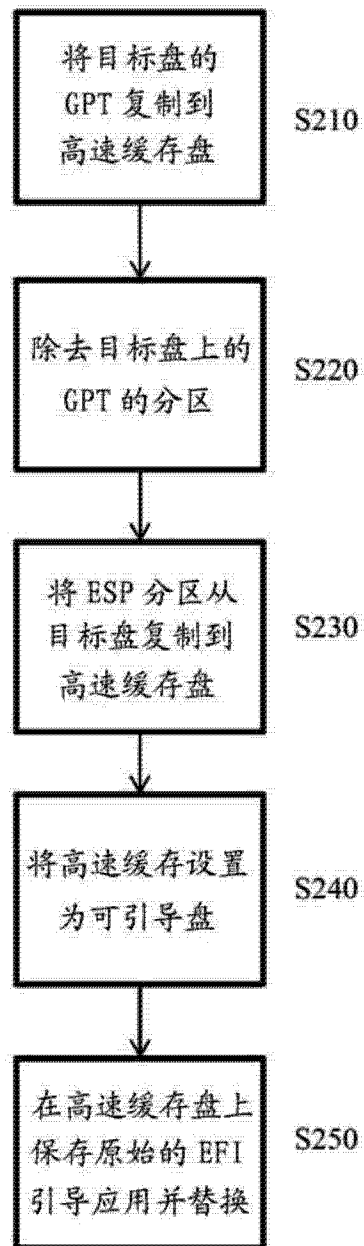


图 2