



(12) 发明专利

(10) 授权公告号 CN 110287129 B

(45) 授权公告日 2021.07.13

(21) 申请号 201910568298.9

(22) 申请日 2019.06.27

(65) 同一申请的已公布的文献号  
申请公布号 CN 110287129 A

(43) 申请公布日 2019.09.27

(73) 专利权人 深圳忆联信息系统有限公司  
地址 518067 广东省深圳市南山区蛇口街  
道蛇口前海大道东角头厂房D24/F-02

(72) 发明人 左建 冯元元 冷志源

(74) 专利代理机构 深圳市精英专利事务所  
44242  
代理人 王海滨

(51) Int.Cl.  
G06F 12/02 (2006.01)

(56) 对比文件

CN 109918316 A, 2019.06.21

CN 102043727 A, 2011.05.04

CN 102662849 A, 2012.09.12

CN 103425600 A, 2013.12.04

CN 101382918 A, 2009.03.11

CN 103999038 A, 2014.08.20

CN 107943719 A, 2018.04.20

US 8645664 B1, 2014.02.04

US 2008189499 A1, 2008.08.07

US 2011022805 A1, 2011.01.27

高源. Flash存储器混合型地址映射算法研究.《中国优秀硕士学位论文全文数据库·信息科技辑》.2015,

审查员 宋巧君

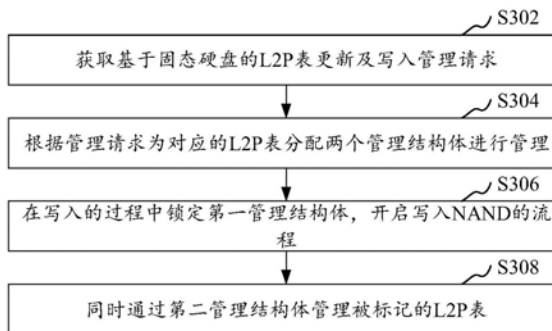
权利要求书1页 说明书6页 附图7页

(54) 发明名称

基于固态硬盘的L2P表更新及写入管理方法和装置

(57) 摘要

本申请涉及一种基于固态硬盘的L2P表更新及写入管理方法、装置、计算机设备和存储介质,其中该方法包括:获取基于固态硬盘的L2P表更新及写入管理请求;根据所述管理请求为对应的L2P表分配两个管理结构体进行管理;在写入的过程中锁定第一管理结构体,开启写入NAND的流程;同时通过第二管理结构体管理被标记的L2P表,从而在L2P表写入NAND的时间不必阻塞FTL处理其他控制节点以实现提升SSD的效率。本发明提出一种固态存储设备l2p表更新及写入的管理方法,在L2P表写入的时间不必阻塞FTL处理其他控制节点,提升SSD的效率。



1. 一种基于固态硬盘的L2P表更新及写入管理方法,所述方法包括:
  - 获取基于固态硬盘的L2P表更新及写入管理请求;
  - 根据所述管理请求为对应的L2P表分配两个管理结构体进行管理;
  - 在写入的过程中锁定第一管理结构体,开启写入NAND的流程;
  - 同时通过第二管理结构体管理被标记的L2P表,从而在L2P表写入NAND的时间不必阻塞FTL处理其他控制节点以实现提升SSD的效率;
  - 所述方法还包括:获取基于固态硬盘的L2P表更新请求;
  - 根据所述更新请求在DDR中修改对应的L2P表,其中,所述L2P表按照一定的颗粒度切分成多个mp,所述mp对应的映射表会在NAND中保存;
  - 根据mp的颗粒度计算修改位置所在mp,并检查所述mp对应的bmp中的位是否已经置上;若所述mp对应的bmp中的位没有置上,则将对应的位置上并更新dirty\_cnt; bmp为数组,每一个bit位被置上代表对应mp的内容被修改了,dirty\_cnt用于记录被修改的mp的个数;
  - 当所述dirty\_cnt的大小大于预设的刷新阈值时,触发L2P写入任务;
  - 获取基于固态硬盘的L2P表写入请求;
  - 根据所述写入请求将flush\_mng指向active\_mng的对象,将active\_mng指向的对象交换成另一个管理结构体;其中,两个指针active\_mng和flush\_mng分别指向两个管理结构体; active\_mng指向的管理结构体负责管理固件运行中L2P表修改的标记;在L2P写入过程中将处理flush\_mng管理的L2P表的写入,在写入完成后该管理结构体清零,同时在写入过程中,对L2P表的修改将在active\_mng中被标记;
  - 将任务中被标记的L2P表写入NAND,并更新M2P表,所述M2P表以mp编号索引,记录mp对应的映射表在NAND中保存的地址。
2. 根据权利要求1所述的基于固态硬盘的L2P表更新及写入管理方法,其特征在于,在所述检查所述mp对应的bmp中的位是否已经置上的步骤之后还包括:
  - 若所述mp对应的bmp中的位已经置上,则直接跳出。
3. 一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现权利要求1或2所述方法的步骤。
4. 一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现权利要求1或2所述的方法的步骤。

## 基于固态硬盘的L2P表更新及写入管理方法和装置

### 技术领域

[0001] 本发明涉及固态硬盘技术领域,特别是涉及一种基于固态硬盘的L2P表更新及写入管理方法、装置、计算机设备和存储介质。

### 背景技术

[0002] 目前,SSD(Solid State Disk固态硬盘)已经被广泛应用于各种场合,由于其在性能、功耗、环境适应性等方面的优秀指标,正逐步替换传统的硬盘。FTL(Flash Translation Layer)算法是SSD(Solid State Disk)固件的核心部分,由于NAND不允许page上重复写入,FTL需要维护逻辑到物理的地址映射。

[0003] 在传统技术中,目前SSD固件大都采用页映射的方式,地址映射表即L2P表在DDR中加载或者修改,然后FTL会管理这些被修改的L2P表,在某个时候触发一个L2P表写入的任务将这些“脏”的表格写入NAND。通常这样的处理机制,在映射表写入NAND的时间中,FTL需要停止处理可能更改地址映射表的操作节点,而这样会影响SSD的效率。

### 发明内容

[0004] 基于此,有必要针对上述技术问题,提供一种可以提升SSD效率的基于固态硬盘的L2P表更新及写入管理方法、装置、计算机设备和存储介质。

[0005] 一种基于固态硬盘的L2P表更新及写入管理方法,所述方法包括:

[0006] 获取基于固态硬盘的L2P表更新及写入管理请求;

[0007] 根据所述管理请求为对应的L2P表分配两个管理结构体进行管理;

[0008] 在写入的过程中锁定第一管理结构体,开启写入NAND的流程;

[0009] 同时通过第二管理结构体管理被标记的L2P表,从而在L2P表写入NAND的时间不必阻塞FTL处理其他控制节点以实现提升SSD的效率。

[0010] 在其中一个实施例中,所述方法还包括:

[0011] 获取基于固态硬盘的L2P表更新请求;

[0012] 根据所述更新请求在DDR中修改对应的L2P表,其中,所述L2P表按照一定的颗粒度切分成多个mp,所述mp对应的映射表会在NAND中保存;

[0013] 根据mp的颗粒度计算修改位置所在mp,并检查所述mp对应的bmp中的位是否已经置上;

[0014] 若所述mp对应的bmp中的位没有置上,则将对应的位置上并更新dirty\_cnt;

[0015] 当所述dirty\_cnt的大小大于预设的刷新阈值时,触发L2P写入任务。

[0016] 在其中一个实施例中,在所述检查所述mp对应的bmp中的位是否已经置上的步骤之后还包括:

[0017] 若所述mp对应的bmp中的位已经置上,则直接跳出。

[0018] 在其中一个实施例中,所述方法还包括:

[0019] 获取基于固态硬盘的L2P表写入请求;

- [0020] 根据所述写入请求将flush\_mng指向active\_mng的对象,将active\_mng指向的对象交换成另一个mng;
- [0021] 将任务中被标记的L2P表写入NAND,并更新M2P表。
- [0022] 一种基于固态硬盘的L2P表更新及写入管理装置,所述装置包括:
- [0023] 获取模块,所述获取模块用于获取基于固态硬盘的L2P表更新及写入管理请求;
- [0024] 分配管理模块,所述分配管理模块用于根据所述管理请求为对应的L2P表分配两个管理结构体进行管理;
- [0025] 第一管理模块,所述第一管理模块用于在写入的过程中锁定第一管理结构体,开启写入NAND的流程;
- [0026] 第二管理模块,所述第二管理模块用于同时通过第二管理结构体管理被标记的L2P表,从而在L2P表写入NAND的时间不必阻塞FTL处理其他控制节点以实现提升SSD的效率。
- [0027] 在其中一个实施例中,所述装置还包括L2P表更新模块,所述L2P表更新模块用于:
- [0028] 获取基于固态硬盘的L2P表更新请求;
- [0029] 根据所述更新请求在DDR中修改对应的L2P表,其中,所述L2P表按照一定的颗粒度切分成多个mp,所述mp对应的映射表会在NAND中保存;
- [0030] 根据mp的颗粒度计算修改位置所在mp,并检查所述mp对应的bmp中的位是否已经置上;
- [0031] 若所述mp对应的bmp中的位没有置上,则将对应该位置上并更新dirty\_cnt;
- [0032] 当所述dirty\_cnt的大小大于预设的刷新阈值时,触发L2P写入任务。
- [0033] 在其中一个实施例中,所述L2P表更新模块还用于:
- [0034] 若所述mp对应的bmp中的位已经置上,则直接跳出。
- [0035] 在其中一个实施例中,所述装置还包括L2P表写入模块,所述L2P表写入模块用于:
- [0036] 获取基于固态硬盘的L2P表写入请求;
- [0037] 根据所述写入请求将flush\_mng指向active\_mng的对象,将active\_mng指向的对象交换成另一个mng;
- [0038] 将任务中被标记的L2P表写入NAND,并更新M2P表。
- [0039] 一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,所述处理器执行所述计算机程序时实现上述任意一项方法的步骤。
- [0040] 一种计算机可读存储介质,其上存储有计算机程序,所述计算机程序被处理器执行时实现上述任意一项方法的步骤。
- [0041] 上述基于固态硬盘的L2P表更新及写入管理方法、装置、计算机设备和存储介质,通过获取基于固态硬盘的L2P表更新及写入管理请求;根据管理请求为对应的L2P表分配双Cache进行管理。通过双Cache的管理方法,在写入的过程中锁定一个Cache,开启写入NAND的流程,同时用另一个Cache管理新的标记为“脏”的映射表,从而在地址映射表写入的时间不必阻塞FTL处理其他控制节点,提升SSD的效率。

## 附图说明

- [0042] 图1为传统技术中逻辑页物理映射表的示意图;

- [0043] 图2为传统技术中映射表写入过程中的管理结构体的示意图；
- [0044] 图3为一个实施例中基于固态硬盘的L2P表更新及写入管理方法的流程示意图；
- [0045] 图4为另一个实施例中基于固态硬盘的L2P表更新及写入管理方法的流程示意图；
- [0046] 图5为再一个实施例中基于固态硬盘的L2P表更新及写入管理方法的流程示意图；
- [0047] 图6为一个实施例中映射表写入过程中的双管理结构体的示意图；
- [0048] 图7为一个实施例中更新L2P表步骤的流程示意图；
- [0049] 图8为一个实施例中L2P表写入步骤的流程示意图；
- [0050] 图9为一个实施例中基于固态硬盘的L2P表更新及写入管理装置的结构框图；
- [0051] 图10为另一个实施例中基于固态硬盘的L2P表更新及写入管理装置的结构框图；
- [0052] 图11为再一个实施例中基于固态硬盘的L2P表更新及写入管理装置的结构框图；
- [0053] 图12为一个实施例中计算机设备的内部结构图。

### 具体实施方式

[0054] 为了使本申请的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本申请进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本申请，并不用于限定本申请。

[0055] 现有的SSD系统中通常会在DDR中更新L2P表然后触发写入任务后台写入NAND；可能会采用下面介绍的方式。

[0056] 如图1所示，左边的L2P table为逻辑页物理地址映射表，表格以逻辑页地址索引，表格中为逻辑页对应的物理地址，这份表格在DDR中保存，将这份表格按照一定的颗粒度切分，本例中这个切割为每64个表项组成一个mp，mp对应的映射表会在NAND中保存，图1右的M2P table以mp编号索引，记录mp对应的映射表在NAND中保存的地址。

[0057] 如图2所示，为映射表写入的管理结构体，dirty\_bmp[]为数组，每一个bit位被置上代表对应的mp的内容被修改了(标记为“脏”，未被写入NAND)，dirty\_cnt记录“脏”的mp的个数，当大于某个阈值，则触发一次写入任务，将DDR中“脏”的映射表写入NAND中，再将这个结构体清零；写入某段mp的L2P表需要分配一个新的页，并将这段数据复制到一段缓冲区与其他“脏”的L2P表凑成一个页，如果在写入过程中，这段表格已经被复制，其他的任务对这段表格修改，那么写入NAND中数据将会是旧的数据，而该“脏”的mp将在写入完成后对应的bit位被清除，这样后果就是表格被修改而这次更新不是最新的数据；而如果写入任务中不断有新的“脏”mp被标记则当前写入任务状态机会无法跳出；所以通常做法是在进入映射表写入任务时候锁定L2P表，禁止修改该表格的任务；这样会在一定程度上影响SSD的效率。

[0058] 本发明基于上述技术问题提出了一种优化的SSD坏块管理方法，提出一种固态存储设备L2P表更新及写入的管理方法，在L2P表写入的时间不必阻塞FTL处理其他控制节点，提升SSD的效率。

[0059] 在一个实施例中，如图3所示，提供了一种基于固态硬盘的L2P表更新及写入管理方法，该方法包括：

- [0060] 步骤302，获取基于固态硬盘的L2P表更新及写入管理请求；
- [0061] 步骤304，根据管理请求为对应的L2P表分配两个管理结构体进行管理；
- [0062] 步骤306，在写入的过程中锁定第一管理结构体，开启写入NAND的流程；

[0063] 步骤308,同时通过第二管理结构体管理被标记的L2P表,从而在L2P表写入NAND的时间不必阻塞FTL处理其他控制节点以实现提升SSD的效率。

[0064] 具体地,参考图6所示,有两个负责L2P表更新及写入的管理结构体mng[0]和mng[1],以及两个指针active\_mng和flush\_mng指向上述两个管理结构体之一。active\_mng指向的结构体负责管理固件运行中L2P表修改的标记,当其中的dirty\_cnt大于某个阈值,则触发L2P表写入任务。在L2P表写入任务的开始,将flush\_mng指向active\_mng的对象,将active\_mng的对象交换成另一个mng。在本次L2P写入任务中将处理flush\_mng管理的L2P表的写入,在写入完成后该管理结构体清零,同时在写入过程中,对L2P表的修改将在active\_mng中被标记。这样,实现了在L2P表写入的时间不必阻塞FTL处理其他控制节点,提升SSD的效率。

[0065] 在本实施例中,通过获取基于固态硬盘的L2P表更新及写入管理请求;管理请求为对应的L2P表分配双Cache进行管理。通过双Cache的管理方法,在写入的过程中锁定一个Cache,开启写入NAND的流程,同时用另一个Cache管理新的标记为“脏”的映射表,从而在地址映射表写入的时间不必阻塞FTL处理其他控制节点,实现了提升SSD的效率。

[0066] 在一个实施例中,如图4所示,提供了一种基于固态硬盘的L2P表更新及写入管理方法,该方法还包括:

[0067] 步骤402,获取基于固态硬盘的L2P表更新请求;

[0068] 步骤404,根据更新请求在DDR中修改对应的L2P表,其中,L2P表按照一定的颗粒度切分成多个mp,mp对应的映射表会在NAND中保存;

[0069] 步骤406,根据mp的颗粒度计算修改位置所在mp,并检查mp对应的bmp中的位是否已经置上;

[0070] 步骤408,若mp对应的bmp中的位没有置上,则将对应的位置上并更新dirty\_cnt;

[0071] 步骤410,当dirty\_cnt的大小大于预设的刷新阈值时,触发L2P写入任务。

[0072] 在一个实施例中,在检查mp对应的bmp中的位是否已经置上的步骤之后还包括:

[0073] 若mp对应的bmp中的位已经置上,则直接跳出。

[0074] 具体地,参考图7所示,在本实施例中更新L2P表的流程包括:

[0075] 更新L2P表首先在DDR中修改,然后根据mp的颗粒度计算修改的位置所在mp,检查active\_mng结构体中mp对应的bmp中的bit位是否已经置上(代表标记为“脏”),如果已经置上则跳出,否则置上对应的位,更新dirty\_cnt,如果大于L2P表写入任务的阈值则触发该任务,否则跳出。

[0076] 在本实施例中,在L2P表写入的时间不必阻塞FTL处理其他控制节点,提升了SSD的效率。

[0077] 在一个实施例中,如图5所示,提供了一种基于固态硬盘的L2P表更新及写入管理方法,该方法还包括:

[0078] 步骤502,获取基于固态硬盘的L2P表写入请求;

[0079] 步骤504,根据写入请求将flush\_mng指向active\_mng的对象,将active\_mng指向的对象交换成另一个mng;

[0080] 步骤506,将任务中被标记的L2P表写入NAND,并更新M2P表。

[0081] 具体地,参考图8所示,在本实施例中L2P表写入任务的流程包括:

[0082] 在L2P写入任务的开始,首先将flush\_mng指向active\_mng的对象,将active\_mng的对象交换成另一个mng,任务中将flush\_mng管理的标记为“脏”的L2P表写入NAND,更新M2P表。

[0083] 应该理解的是,虽然图3-8的流程图中的各个步骤按照箭头的指示依次显示,但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明,这些步骤的执行并没有严格的顺序限制,这些步骤可以以其它的顺序执行。而且,图3-8中的至少一部分步骤可以包括多个子步骤或者多个阶段,这些子步骤或者阶段并不必然是在同一时刻执行完成,而是可以在不同的时刻执行,这些子步骤或者阶段的执行顺序也不必然是依次进行,而是可以与其它步骤或者其它步骤的子步骤或者阶段的至少一部分轮流或者交替地执行。

[0084] 在一个实施例中,如图9所示,提供了一种基于固态硬盘的L2P表更新及写入管理装置900,该装置包括:

[0085] 获取模块901,用于获取基于固态硬盘的L2P表更新及写入管理请求;

[0086] 分配管理模块902,用于根据所述管理请求为对应的L2P表分配两个管理结构体进行管理;

[0087] 第一管理模块903,用于在写入的过程中锁定第一管理结构体,开启写入NAND的流程;

[0088] 第二管理模块904,用于同时通过第二管理结构体管理被标记的L2P表,从而在L2P表写入NAND的时间不必阻塞FTL处理其他控制节点以实现提升SSD的效率。

[0089] 在一个实施例中,如图10所示,提供了一种基于固态硬盘的L2P表更新及写入管理装置900,该装置还包括L2P表更新模块905,用于:

[0090] 获取基于固态硬盘的L2P表更新请求;

[0091] 根据更新请求在DDR中修改对应的L2P表,其中,L2P表按照一定的颗粒度切分成多个mp,mp对应的映射表会在NAND中保存;

[0092] 根据mp的颗粒度计算修改位置所在mp,并检查mp对应的bmp中的位是否已经置上;

[0093] 若mp对应的bmp中的位没有置上,则将对应在位置上并更新dirty\_cnt;

[0094] 当dirty\_cnt的大小大于预设的刷新阈值时,触发L2P写入任务。

[0095] 在一个实施例中,L2P表更新模块905还用于:

[0096] 若mp对应的bmp中的位已经置上,则直接跳出。

[0097] 在一个实施例中,如图11所示,提供了一种基于固态硬盘的L2P表更新及写入管理装置900,该装置还包括L2P表写入模块906,用于:

[0098] 获取基于固态硬盘的L2P表写入请求;

[0099] 根据写入请求将flush\_mng指向active\_mng的对象,将active\_mng指向的对象交换成另一个mng;

[0100] 将任务中被标记的L2P表写入NAND,并更新M2P表。

[0101] 关于基于固态硬盘的L2P表更新及写入管理装置的具体限定可以参见上文中对于基于固态硬盘的L2P表更新及写入管理方法的限定,在此不再赘述。

[0102] 在一个实施例中,提供了一种计算机设备,其内部结构图可以如图12所示。该计算机设备包括通过装置总线连接的处理器、存储器以及网络接口。其中,该计算机设备的处理

器用于提供计算和控制能力。该计算机设备的存储器包括非易失性存储介质、内存储器。该非易失性存储介质存储有操作装置、计算机程序和数据库。该内存储器为非易失性存储介质中的操作装置和计算机程序的运行提供环境。该计算机设备的网络接口用于与外部的终端通过网络连接通信。该计算机程序被处理器执行时以实现一种基于固态硬盘的L2P表更新及写入管理方法。

[0103] 本领域技术人员可以理解,图12中示出的结构,仅仅是与本申请方案相关的部分结构的框图,并不构成对本申请方案所应用于其上的计算机设备的限定,具体的计算机设备可以包括比图中所示更多或更少的部件,或者组合某些部件,或者具有不同的部件布置。

[0104] 在一个实施例中,提供了一种计算机设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,处理器执行计算机程序时实现以上各个方法实施例中的步骤。

[0105] 在一个实施例中,提供了一种计算机可读存储介质,其上存储有计算机程序,计算机程序被处理器执行时实现以上各个方法实施例中的步骤。

[0106] 本领域普通技术人员可以理解实现上述实施例方法中的全部或部分流程,是可以通过计算机程序来指令相关的硬件来完成,所述的计算机程序可存储于一非易失性计算机可读存储介质中,该计算机程序在执行时,可包括如上述各方法的实施例的流程。其中,本申请所提供的各实施例中所使用的对存储器、存储、数据库或其它介质的任何引用,均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器(ROM)、可编程ROM(PROM)、电可编程ROM(EPROM)、电可擦除可编程ROM(EEPROM)或闪存。易失性存储器可包括随机存取存储器(RAM)或者外部高速缓冲存储器。作为说明而非局限,RAM以多种形式可得,诸如静态RAM(SRAM)、动态RAM(DRAM)、同步DRAM(SDRAM)、双数据率SDRAM(DDRSDRAM)、增强型SDRAM(ESDRAM)、同步链路(Synchlink)DRAM(SLDRAM)、存储器总线(Rambus)直接RAM(RDRAM)、直接存储器总线动态RAM(DRDRAM)、以及存储器总线动态RAM(RDRAM)等。

[0107] 以上实施例的各技术特征可以进行任意的组合,为使描述简洁,未对上述实施例中的各个技术特征所有可能的组合都进行描述,然而,只要这些技术特征的组合不存在矛盾,都应当认为是本说明书记载的范围。

[0108] 以上所述实施例仅表达了本申请的几种实施方式,其描述较为具体和详细,但并不能因此而理解为对发明专利范围的限制。应当指出的是,对于本领域的普通技术人员来说,在不脱离本申请构思的前提下,还可以做出若干变形和改进,这些都属于本申请的保护范围。因此,本申请专利的保护范围应以所附权利要求为准。



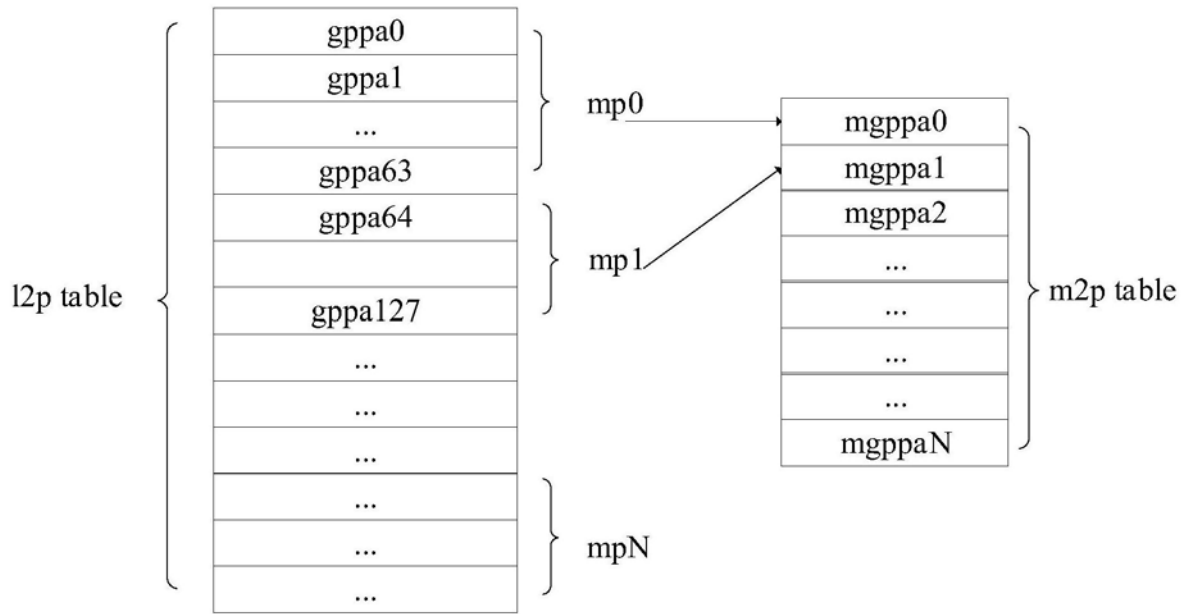


图1

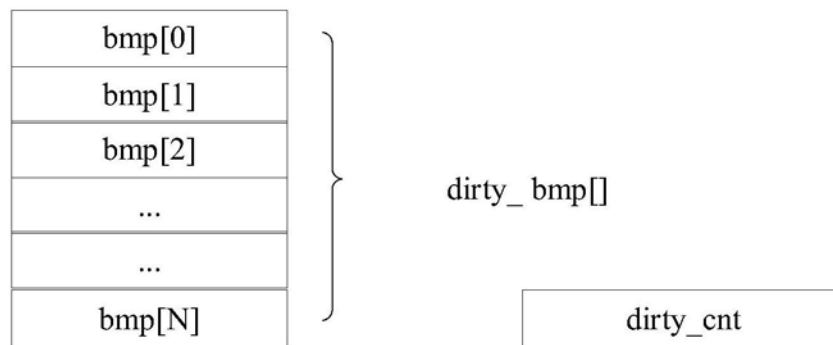


图2

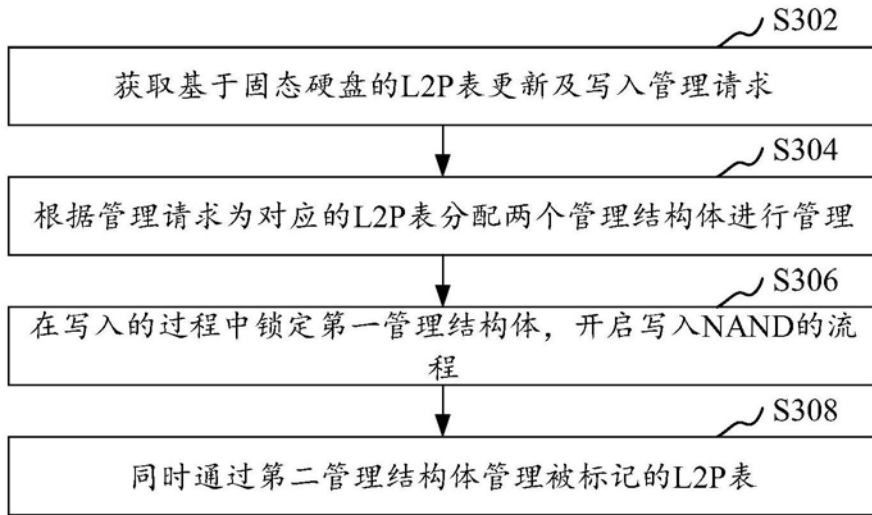


图3

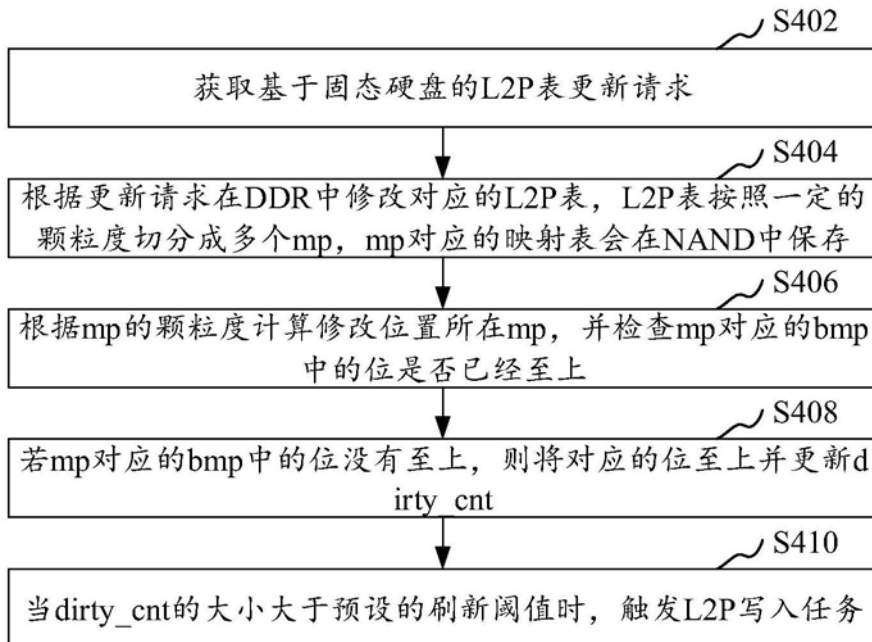


图4

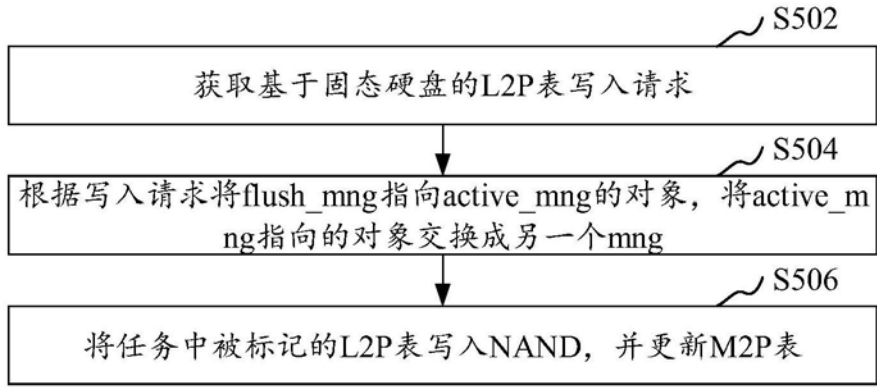


图5

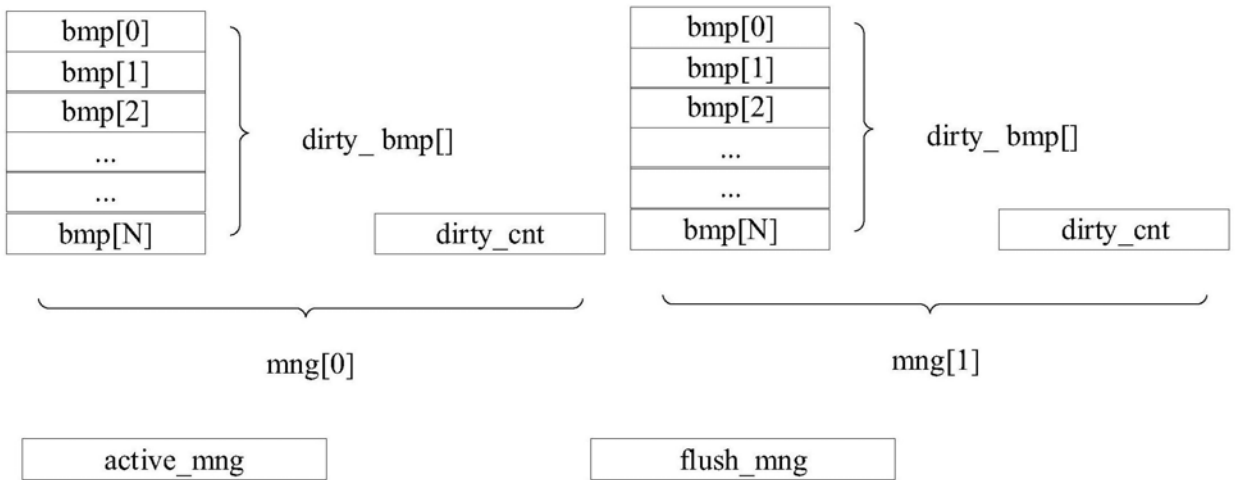


图6

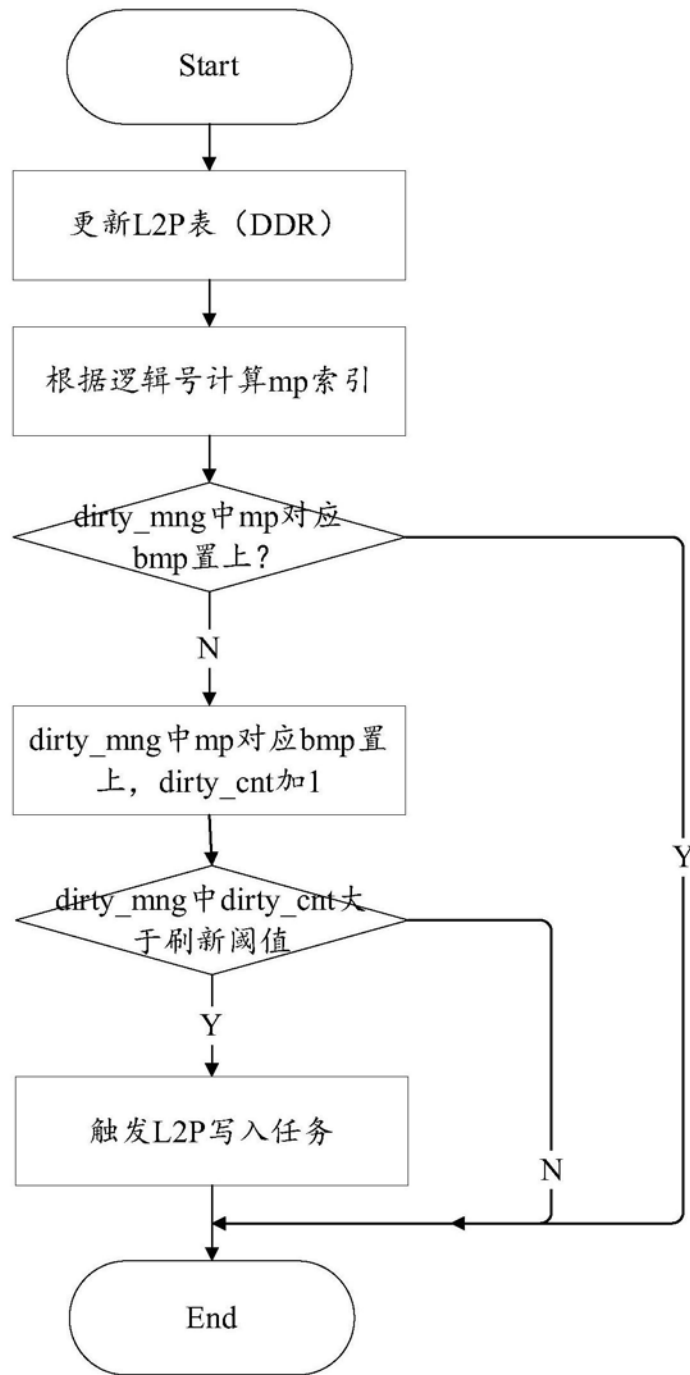


图7



图8

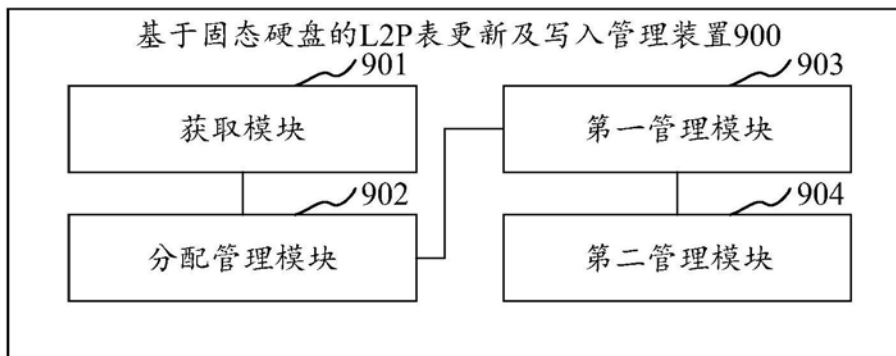


图9

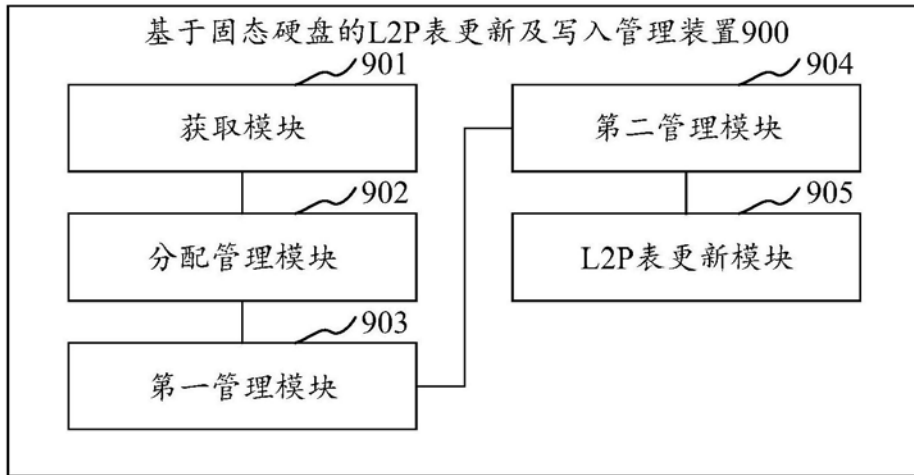


图10

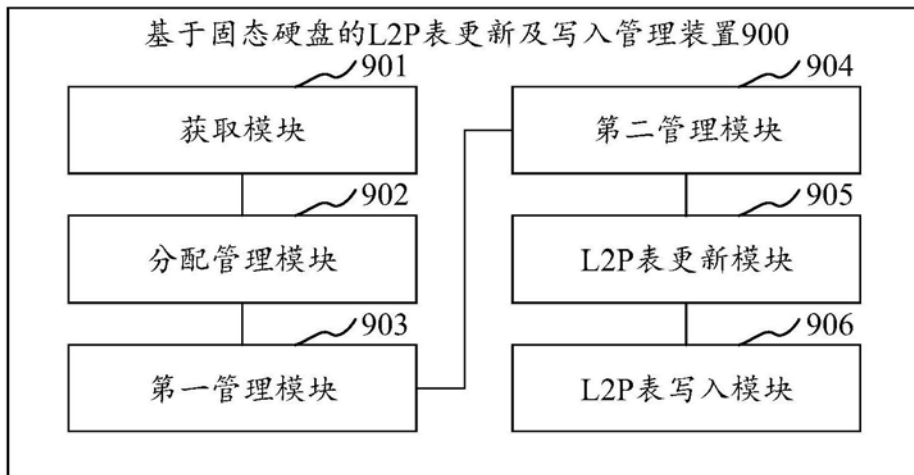


图11

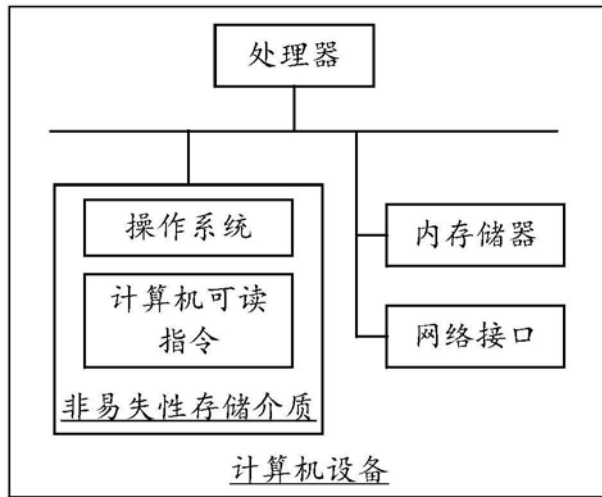


图12