



(12) 发明专利

(10) 授权公告号 CN 115809207 B

(45) 授权公告日 2023.04.18

(21) 申请号 202310056066.1

G06F 12/0882 (2016.01)

(22) 申请日 2023.01.19

(56) 对比文件

(65) 同一申请的已公布的文献号

CN 102306092 A, 2012.01.04

申请公布号 CN 115809207 A

CN 112433961 A, 2021.03.02

(43) 申请公布日 2023.03.17

US 2006265551 A1, 2006.11.23

US 6718439 B1, 2004.04.06

(73) 专利权人 北京象帝先计算技术有限公司

杨祎巍 等. 嵌入式flash读取加速技术研究.《微电子学与计算机》.2022,第39卷(第8期),第107-118页.

地址 100029 北京市朝阳区安定路5号院1

号楼9层(09)901号901室

(72) 发明人 王海洋 姜莹

审查员 初星妍

(74) 专利代理机构 北京新知远方知识产权代理

事务所(普通合伙) 11397

专利代理师 王俊博 易杨

(51) Int. Cl.

G06F 12/0895 (2016.01)

G06F 12/0875 (2016.01)

权利要求书2页 说明书10页 附图6页

(54) 发明名称

Cache中地址标记方法、装置、图形处理系统及电子设备

(57) 摘要

本公开提供一种Cache中地址标记方法、装置、图形处理系统及电子设备,所述Cache的映射方式被配置为组相联,所述Cache中的每一路中的每个缓存行被配置有对应的连续地址标记,任一缓存行对应的连续地址标记用于标记:与该缓存行存在地址连续关系的缓存行所在的路;其中,存在地址连续关系包括:任意两个缓存行对应的主存地址连续;所述方法包括:在对目标缓存行进行更新后,确定更新后的目标缓存行的Tag标记,并获取与目标缓存行所在的目标组相邻的组之中的各缓存行的Tag标记;根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的缓存行的连续地址标记进行更新。

Way编号 way0

1	Tag=0x1000
0	Tag=0x2000
1	Tag=0x2000
0	Tag=0x1000
0	Tag=0x1000
0	Tag=0x1000
1	Tag=0x3000
4	Tag=0x1100

Way编号 way1

0	Tag=0x2000
4	Tag=0x1000
4	Tag=0x1200
4	Tag=0x2000
4	Tag=0x1200
1	Tag=0x1100
0	Tag=0x1100
4	Tag=0x1000

1. 一种Cache中地址标记方法,其特征在于,所述Cache的映射方式被配置为组相联,所述Cache中的每一路中的每个缓存行被配置有对应的连续地址标记,任一缓存行对应的连续地址标记用于标记:与该缓存行存在地址连续关系的缓存行所在的路;其中,存在地址连续关系包括:任意两个缓存行对应的主存地址连续;所述方法包括:

在对目标缓存行进行更新后,确定更新后的目标缓存行的Tag标记,并获取与目标缓存行所在的目标组相邻的组之中的各缓存行的Tag标记;

根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的缓存行的连续地址标记进行更新。

2. 根据权利要求1所述的方法,其特征在于,主存地址中的存储库地址以及行地址被映射为Tag标记;

所述根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的各缓存行的连续地址标记进行更新,包括:

若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记所述目标缓存行所在的路,并将其他第一缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路;其中,所述第一缓存行为目标组上一组中的缓存行;

若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记所述第二缓存行所在的路;其中,所述第二缓存行为目标组下一组中的缓存行。

3. 根据权利要求2所述的方法,其特征在于,所述将其他第一缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路,包括:

若任一第一缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

4. 根据权利要求1所述的方法,其特征在于,主存地址中的存储库地址以及行地址被映射为Tag标记;

所述根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的各缓存行的连续地址标记进行更新,包括:

若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记所述目标缓存行所在的路,并将其他第二缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路;其中,所述第二缓存行为目标组下一组中的缓存行;

若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记所述第一缓存行所在的路;其中,所述第一缓存行为目标组上一组中的缓存行。

5. 根据权利要求4所述的方法,其特征在于,所述将其他第二缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路,包括:

若任一第二缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

6. 一种Cache中地址标记装置,其特征在于,所述Cache的映射方式被配置为组相联,所

述Cache中的每一路中的每个缓存行被配置有对应的连续地址标记,任一缓存行对应的连续地址标记用于标记:与该缓存行存在地址连续关系的缓存行所在的路;其中,存在地址连续关系包括:任意两个缓存行对应的主存地址连续;所述装置包括:

获取模块,用于在对目标缓存行进行更新后,确定更新后的目标缓存行的Tag标记,并获取与目标缓存行所在的目标组相邻的组之中的各缓存行的Tag标记;

更新模块,用于根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的缓存行的连续地址标记进行更新。

7.根据权利要求6所述的装置,其特征在于,主存地址中的存储库地址以及行地址被映射为Tag标记;

所述更新模块,具体用于若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记所述目标缓存行所在的路,并将其他第一缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路;其中,所述第一缓存行为目标组上一组中的缓存行;

若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记所述第二缓存行所在的路;其中,所述第二缓存行为目标组下一组中的缓存行。

8.根据权利要求7所述的装置,其特征在于,所述更新模块,具体用于若任一第一缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

9.根据权利要求6所述的装置,其特征在于,主存地址中的存储库地址以及行地址被映射为Tag标记;

所述更新模块,具体用于若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记所述目标缓存行所在的路,并将其他第二缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路;其中,所述第二缓存行为目标组下一组中的缓存行;

若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记所述第一缓存行所在的路;其中,所述第一缓存行为目标组上一组中的缓存行。

10.根据权利要求9所述的装置,其特征在于,

所述更新模块,具体用于若任一第二缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

11.一种图形处理系统,其特征在于,包括上述权利要求6-10任一项所述的Cache中地址标记装置。

12.一种电子组件,其特征在于,包括权利要求11所述的图形处理系统。

13.一种电子设备,其特征在于,包括权利要求12所述的电子组件。

Cache中地址标记方法、装置、图形处理系统及电子设备

技术领域

[0001] 本公开涉及计算机技术领域,尤其涉及一种Cache中地址标记方法、装置、图形处理系统和电子设备。

背景技术

[0002] Cache即高速缓冲,是除寄存器以外最靠近处理器的存储单元,通常由SRAM组成。将Cache部署在处理器和主存之间,同时将主存中的部分数据也存储在Cache中,处理器可以通过访问Cache读写数据以避免直接访问主存,由于对Cache的读写速度比对主存读写的速度要快,因此提升了读写效率,同时也减少了处理器对主存的访问次数。

[0003] 在使用Cache进行访问的过程中,存在从Cache中识别出主存地址连续的多个缓存行(Cache line)的需求。

发明内容

[0004] 本公开的目的是提供一种Cache中地址标记方法、装置、图形处理系统和电子设备。

[0005] 根据本公开的一个方面,提供一种Cache中地址标记方法,所述Cache的映射方式被配置为组相联,所述Cache中的每一路中的每个缓存行被配置有对应的连续地址标记,任一缓存行对应的连续地址标记用于标记:与该缓存行存在地址连续关系的缓存行所在的路;其中,存在地址连续关系包括:任意两个缓存行对应的主存地址连续;所述方法包括:

[0006] 在对目标缓存行进行更新后,确定更新后的目标缓存行的Tag标记,并获取与目标缓存行所在的目标组相邻的组之中的各缓存行的Tag标记;

[0007] 根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的缓存行的连续地址标记进行更新。

[0008] 在一个实施例中,其中,主存地址中的存储库地址以及行地址被映射为Tag标记;

[0009] 所述根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的各缓存行的连续地址标记进行更新,包括:

[0010] 若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记所述目标缓存行所在的路,并将其他第一缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路;其中,所述第一缓存行为目标组上一组中的缓存行;

[0011] 若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记所述第二缓存行所在的路;其中,所述第二缓存行为目标组下一组中的缓存行。

[0012] 在一个实施例中,所述将其他第一缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路,包括:

[0013] 若任一第一缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第一

缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

[0014] 在一个实施例中,主存地址中的存储库地址以及行地址被映射为Tag标记;

[0015] 所述根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的各缓存行的连续地址标记进行更新,包括:

[0016] 若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记所述目标缓存行所在的路,并将其他第二缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路;其中,所述第二缓存行为目标组下一组中的缓存行;

[0017] 若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记所述第一缓存行所在的路;其中,所述第一缓存行为目标组上一组中的缓存行。

[0018] 在一个实施例中,所述将其他第一缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路,包括:

[0019] 若任一第一缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

[0020] 在一个实施例中,其中,主存地址中的存储库地址以及行地址被映射为Tag标记;

[0021] 所述根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的各缓存行的连续地址标记进行更新,包括:

[0022] 若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记所述目标缓存行所在的路,并将其他第二缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路;其中,所述第二缓存行为目标组下一组中的缓存行;

[0023] 若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记所述第一缓存行所在的路;其中,所述第一缓存行为目标组上一组中的缓存行。

[0024] 在一个实施例中,所述将其他第二缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路,包括:

[0025] 若任一第二缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

[0026] 根据本公开的第二个方面,提供一种Cache中地址标记装置,所述Cache的映射方式被配置为组相联,所述Cache中的每一路中的每个缓存行被配置有对应的连续地址标记,任一缓存行对应的连续地址标记用于标记:与该缓存行存在地址连续关系的缓存行所在的路;其中,存在地址连续关系包括:任意两个缓存行对应的主存地址连续;

[0027] 所述装置包括:

[0028] 获取模块,用于在对目标缓存行进行更新后,确定更新后的目标缓存行的Tag标记,并获取与目标缓存行所在的目标组相邻的组之中的各缓存行的Tag标记;

[0029] 更新模块,用于根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的缓存行的连续地址标记进行更新。

[0030] 在一个实施例中,其中,主存地址中的存储库地址以及行地址被映射为Tag标记;

[0031] 所述更新模块,具体用于若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记所述目标缓存行所在的路,并将其他第一缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路;其中,所述第一缓存行为目标组上一组中的缓存行;若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记所述第二缓存行所在的路;其中,所述第二缓存行为目标组下一组中的缓存行。

[0032] 在一个实施例中,所述更新模块,具体用于若任一第一缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

[0033] 在一个实施例中,其中,主存地址中的存储库地址以及行地址被映射为Tag标记;

[0034] 所述更新模块,具体用于若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记所述目标缓存行所在的路,并将其他第二缓存行的连续地址标记更新为不用于标记所述目标缓存行所在的路;其中,所述第二缓存行为目标组下一组中的缓存行;

[0035] 若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记所述第一缓存行所在的路;其中,所述第一缓存行为目标组上一组中的缓存行。

[0036] 在一个实施例中,所述更新模块,具体用于若任一第二缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

[0037] 根据本公开的第三个方面,提供一种图形处理系统,包括上述第二方面所述的装置。

[0038] 根据本公开的第四个方面,提供一种电子组件,包括上述第三方面所述的图形处理系统。

[0039] 根据本公开的第五个方面,提供一种电子设备,包括第四方面的电子组件。

[0040] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本公开。

附图说明

[0041] 图1为本公开一个实施例提供的主存和缓存直接映射的结构示意图;

[0042] 图2为本公开一个实施例提供的主存和缓存组相联映射的结构示意图;

[0043] 图3为本公开一个实施例提供的主存和缓存全相联映射的结构示意图;

[0044] 图4为本公开一个实施例提供的一种基于Cache地址标记方法的流程示意图;

[0045] 图5为本公开一个实施例提供的一种后向址连续数据标记的示意图;

[0046] 图6为本公开一个实施例提供的另一种后向址连续数据标记的示意图;

[0047] 图7为本公开一个实施例提供的又一种后向址连续数据标记的示意图;

[0048] 图8为本公开一个实施例提供的再一种后向址连续数据标记的示意图;

[0049] 图9为本公开一个实施例提供的一种Cache中地址标记装置的结构示意图;

[0050] 图10为本公开一个实施例提供的图形处理系统的结构示意图。

具体实施方式

[0051] 在介绍本公开实施例之前,应当说明的是:

[0052] 本公开部分实施例被描述为处理流程,虽然流程的各个操作步骤可能被冠以顺序的步骤编号,但是其中的操作步骤可以被并行地、并发地或者同时实施。

[0053] 本公开实施例中可能使用了术语“第一”、“第二”等等来描述各个特征,但是这些特征不应当受这些术语限制。使用这些术语仅仅是为了将一个特征与另一个特征进行区分。

[0054] 本公开实施例中可能使用了术语“和/或”,“和/或”包括其中一个或更多所列出的相关联特征的任意和所有组合。

[0055] 应当理解的是,当描述两个部件的连接关系或通信关系时,除非明确指明两个部件之间直接连接或直接通信,否则,两个部件的连接或通信可以理解为直接连接或通信,也可以理解为通过中间部件间接连接或通信。

[0056] 为了使本公开实施例中的技术方案及优点更加清楚明白,以下结合附图对本公开的示例性实施例进行进一步详细的说明,显然,所描述的实施例仅是本公开的一部分实施例,而不是所有实施例的穷举。需要说明的是,在不冲突的情况下,本公开中的实施例及实施例中的特征可以相互组合。

[0057] Cache的容量相对于主存的容量较小,它保存的数据是主存的一个子集,且Cache和主存进行地址交换是以缓存行为单位的。即把主存和Cache分割成一定大小的块,这个块在主存中称为数据块data block,在Cache中称为缓存行即Cacheline,将主存中的data block映射到Cache中的缓存行中就需要采用相应的映射方式,常见的映射方式包括直接映射、组相联映射以及全相联映射。

[0058] 直接映射采用取模的方式进行一对一映射,容易出现Cache不命中的情况。组相联中每一组(set)中可以存储更多的信息,因此相对于直接映射的方式,增加了Cache命中的几率。全相联是极端的组相联,即Cache只有一个set,实现较为复杂,目前业界最常用的映射方法为组相联。

[0059] 如图1所示,为直接映射的示意图,在直接映射的方式中,如果Cache中共有8个缓存行,那么主存中的0、8、16、24...号数据块会被映射到缓存行0中,同理1、9、17...号数据块会被映射到缓存行1中。当主机的读取顺序是数据块0-数据块8-数据块0-数据块8的情况下,由于缓存行0只能同时缓存一个数据块,因此,在读取数据块8以及第二次读取数据块0时,会发生Cache不命中的情况,即在Cache中无法找到数据块需要去主存中查找数据块。因此,直接映射的方式中Cache命中的几率并不高,进而整体的读写效率也不高。

[0060] 在组相联的方式中包括N个路(way),每个way中包括M组(set)。每一组即每一set中包括N个缓存行。如图2所示,存在两个way,即way0和way1,每一way有8行,对应8组,每组中有2个缓存行,即way0的Cache line0和way1的Cache line0为一组。

[0061] 这样主存中数据块0、8、16、24...中任意两个数据块可以同时存储在缓存中的两个Cache line0中,进而组相联映射的方式相对于直接映射增加了Cache命中的概率提升了整体的读写效率。

[0062] 如图3所示,为全相联的示意图,全相联是极端的组相联,即cache只有一个组。每一个数据块都可以存进任何一个缓存行。全相联方式中Cache命中的概率也较高,但是实现

较为复杂,工作延时也较高。

[0063] 在直接映射和组相联的映射方式中,处理器发送到Cache中的地址被划分为3段,包括Tag、Index、Line offset。其中line offset用来表示该地址在缓存行中的偏移,Index表示该地址位于哪一个组(set)中(组相联映射的方式中)或哪一行中(直接映射的方式中),Tag用于确定是否命中数据块。其中,每个缓存行中包括有效位V、标记位Tag和数据位Data block。Cache根据接收地址的Tag和标记位Tag进行对比就能判断出该地址在Cache中是否命中。

[0064] 而在全相联的场景中不存在组的概念,因此处理器发来的地址只包括块号Tag和Line offset。

[0065] 本公开中的主存可以是DDR SDRAM,即双倍速率同步动态随机存储器,主存地址通常由column(列)地址、bank(存储库)地址、row(行)组成。

[0066]	主存地址	row[13:0]	bank[3:1]	bank[0]	column[11:6]	column[5:0]
--------	------	-----------	-----------	---------	--------------	-------------

[0067] 表1

[0068] 例如,如表1所示,DDR颗粒由12bit的column(列)地址,4bit的bank(存储库)地址和14bit的row(行)地址组成。另外,在一些场景中,主存地址中可能并不包含bank,仅由row和column组成。其中,一个row也为主存的一个主存页,如果需要从主存中读写数据,则需要先定位和开启主存页,进一步在主存页中的相应位置读写数据。

[0069] 另外,如前文介绍,在直接映射和组相联映射的方式中,在缓存中Cache line地址被分为了Tag、Index以及Line offset,因此主存中的row、bank、column在映射到缓存中时,会被映射为Tag、Index以及Line offset。

[0070] 在实际应用中,在直接映射和组相联的方式中,可以将主存地址中的row地址和bank地址被映射为Tag。

[0071] 例如,如表2所示:

[0072]	主存地址	row[13:0]	bank[3:1]	bank[0]	column[11:6]	column[5:0]
	缓存地址	Tag			Index	Line offset

[0073] 表2

[0074] 此外,row地址也可以被映射为Tag;row地址和bank地址还可能被映射为包含于Tag,例如Tag中包括row地址、bank地址以及column中的部分地址,具体的映射方式往往和Cache中cache line即缓存行的数量以及缓存行中包含的数据量的大小相关,具体方式可以参照相关技术,本公开对此不进行详述。

[0075] 由于组相联相对于直接相联Cache命中率较高,且相对于全相联效率较高,因此目前业界通常使用组相联的方式进行映射。

[0076] 在使用Cache进行访问的过程中,存在从Cache中识别出主存地址连续的多个Cache line的需求,例如将多个主存地址连续的缓存行一起写回至主存,以提升写回效率,其中主存地址连续是指多个缓存行中的数据在主存中的地址是连续的,将连续的数据写回

到主存中,相对于将不连续的多个缓存行分别写回至主存的不同位置,写回效率明显较高。

[0077] 在组相联的映射方式中,如果想要找到主存地址连续的多个Cache缓存行,则需要通过对各个way中各个set的缓存行的tag值进行查找和对比,来找的主存地址连续的多个Cache line,这种方式效率较低,需要较高的延时。

[0078] 以图2示出的组相联为例,如果想要查找出目前Cache中主存地址连续的缓存行,需要查看每个Cache line的tag,例如,如果主存地址中的row地址和bank地址被映射为Tag,在获取了way0的Cache line0的tag后,如果,想要知道是否有Cache line与way0的Cache line0存在主存地址连续的关系,则需要获取way0的Cache line1以及way1的Cache line1的Tag,如果way1的Cache line1的Tag或者way0的Cache line1的Tag与way0的Cache line0的Tag相同,则确定way0的Cache line1或者way1的Cache line1与way0的Cache line0存在主存地址连续的关系。

[0079] 可见,如果想要确定Cache中全部存在主存地址连续的缓存行时,需要遍历全部Cache line的TAG,并进行匹配了,效率较低,如果需要利用主存地址连续的缓存行进行后续事务处理,则需要较高延时。

[0080] 为了解决上述问题,本公开提出在Cache中为每个缓存行配置对应的连续地址标记,该连续地址标记用于标记:与该缓存行存在地址连续关系的缓存行所在的Way,并且在Cache中的缓存行发生更新时,对连续地址标记进行更新,以对Cache中的各个缓存行的主存地址连续性进行标记,进而在需要获取或确定主存地址连续的缓存行时,可以基于连续地址标记快速获取。

[0081] 如图4所示,本公开提出一种Cache中地址标记方法,其中,Cache的映射方式被配置为组相联,Cache中的每一Way中的每个缓存行被配置有对应的连续地址标记,任一缓存行对应的连续地址标记用于标记:与该缓存行存在地址连续关系的缓存行所在的Way;其中,存在地址连续关系包括:任意两个缓存行对应的主存地址连续;方法包括:

[0082] S401,在对目标缓存行进行更新后,确定更新后的目标缓存行的Tag标记,并获取与目标缓存行所在的目标组相邻的组之中的各缓存行的Tag标记;

[0083] S402,根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的各缓存行的连续地址标记进行更新。

[0084] 采用上述方式,可以在缓存行发生更新时,就将Cache中各缓存行的主存地址连续关系进行标记,在需要获取主存地址连续的缓存行时,可以基于连续地址标记快速获取。

[0085] 在Cache中,通常存在对Cache中的缓存行进行更新的场景,本公开中对缓存行进行更新的场景包括:将Cache中的缓存行进行替换,以及对Cache中的空缓存行进行写入。对Cache中的缓存行进行替换是指,在需要将主存中的某个数据块写入Cache中,但是Cache中的各个缓存行均占用后,需要将某个缓存行的数据写回至主存,并将数据块写入该缓存行中。在确定需要替换的缓存行时,可以是将较长时间不使用的缓存行中的数据写回至主存,或者将最早写入Cache的缓存行的数据写回至主存,对Cache中的数据进行替换的场景可以参照相关技术,本公开不进行详述。而对Cache中的空缓存行进行写入是指,在Cache中的某个缓存行还未存储数据时,可以将需要使用Cache的数据块存储到该缓存行中。

[0086] 上述S401,可以是在发生替换或者写入空的缓存行时执行。

[0087] 在下述实施例中,默认主存地址中的bank地址以及row地址被映射为Tag标记。下

面以替换的场景为例,对本公开的地址连续数据标记方法进行详述。

[0088] 在组相联中,存在主存地址连续关系的缓存行必定位于相邻的组中,因此,想要标记与某个缓存行存在主存地址连续关系的缓存行,实际上就是标记缓存行所在的way。

[0089] 如图5所示,为采用本公开提出的其中一种Cache中地址标记方法进行处理后的Cache,图5中的Way编号,即为每个缓存行配置的对应的连续地址标记,如way0的Cache line0的Tag=0x1000,与way1中的Cache line1的Tag=0x1000相同,因此,将way0的Cache line0的连续地址标记记为1,以标记与way0的Cache line0存在主存地址连续关系的缓存行在way1,即为way1的Cache line1。图5中,way编号中的0用于表示与该缓存行存在的主存地址连续的缓存行在way0,1用于表示与该缓存行存在的主存地址连续的缓存行在way1。4用于表示不存在与该缓存行具有主存地址连续关系的缓存行,其中4也可以是任意不为0和1的值。

[0090] 图5中用连续地址标记,标记出了目标缓存行下一组中,哪个way中存在与目标缓存行主存地址连续的缓存行,本公开称这种标记方式为后向标记。在另一实施例中,还可以是标记出了目标缓存行上一组中,哪个way中存在与目标缓存行主存地址连续的缓存行,本公开称这种标记方式为前向标记。

[0091] 在后向标记的方式中,上述S402具体为:

[0092] 若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记目标缓存行所在的way,并将其他第一缓存行的连续地址标记更新为不用于标记目标缓存行所在的way;其中,第一缓存行为目标组上一组中的缓存行;

[0093] 若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记第二缓存行所在的way;其中,第二缓存行为目标组下一组中的缓存行。

[0094] 其中,若任一第一缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

[0095] 例如,如图6所示,如果发生Cache替换1,即将Tag=0x1200的数据块替换掉way1 cache line1中的数据块,由于更新后的Tag=0x1200与way1 cache line2相同,因此需要将way1 cache line1的连续地址标记由4更新为1,以指示与该way1 cache line1存在主存地址连续的缓存行在way1cache line2;同时,由于way0 cache line0的Tag=0x1000,与更新前的way1 cache line1的Tag相同,而与更新后的way1cache line1的Tag不同,因此需要将way0 cache line0的连续地址标记由1更新为4。

[0096] 又如7所示,如果发生Cache替换2,即将Tag=0x1200的数据块替换掉way1 cache line5中的数据块,由于更新后的Tag=0x1200与way1 cache line4相同,因此要将way1 cache line4的连续地址标记由4更新为1,以指示与该way1 cache line4存在主存地址连续的缓存行在way1cache line5,同时,由于更新后的way1 cache line5地Tag不再与way1 cache line6的Tag相同,因此,需要将way1 cache line5的连续地址标记由1更新为4。

[0097] 再如8所示,如果发生Cache替换3,即将Tag=0x4000的数据块替换掉way0 cache line5中的数据块,由于更新后的Tag=0x4000与way0 cache line6和way1 cache line6的Tag都不同,因此要将way0cache line5的连续地址标记由0更新为4,同时,由于更新后的

way0 cache line5的Tag不再与way0 cache line4的Tag相同,因此,需要将way0 cache line4的连续地址标记由0更新为4。

[0098] 上述实施例中,均是在目标缓存行的行更新后,及需要对目标缓存行对应的连续地址标记进行更新,也需要对目标缓存行所在的目标组相邻的组之中的缓存行的连续地址标记进行更新。

[0099] 在其他实施例中,可能仅需要对目标缓存行对应的连续地址标记进行更新,无需对目标缓存行所在的目标组相邻的组之中的缓存行的连续地址标记进行更新,例如,将图8中的way1 cacheline1的Tag=0x1000数据块更新为Tag=0x1200的数据块,则仅需要将way1 cache line1的连续地址标记更新为1,无需更新其他缓存行的标记。

[0100] 在一些实施例中,也可以无需对目标缓存行对应的连续地址标记进行更新,仅需对目标缓存行所在的目标组相邻的组之中的缓存行的连续地址标记进行更新。

[0101] 例如,将图6中的way1 cacheline4的Tag=0x1200数据块更新为Tag=2000的数据块,则仅需要将way1 cache line3的连续地址标记更新为1,无需更新其他缓存行的标记。

[0102] 在前向标记的方式中,上述S402具体为:

[0103] 根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的各缓存行的连续地址标记进行更新,包括:

[0104] 若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记目标缓存行所在的Way,并将其他第二缓存行的连续地址标记更新为不用于标记目标缓存行所在的Way;其中,第二缓存行为目标组下一组中的缓存行;

[0105] 若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记第一缓存行所在的Way;其中,第一缓存行为目标组上一组中的缓存行。

[0106] 其中,若任一第二缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

[0107] 可以理解,前先标记的过程与后向标记的过程类似,区别仅在于,后向标记,标记出了目标缓存行下一组中,哪个way中存在与目标缓存行主存地址连续的缓存行。前向标记,标记出了目标缓存行上一组中,哪个way中存在与目标缓存行主存地址连续的缓存行,本公开不再对前向标记进行赘述。

[0108] 在对连续地址进行标记后,在需要获取主存地址连续的缓存行时,即可以根据连续地址标记快速查找,如图5所示,通过查看way0 cacheline0的连续地址标记确定way1 cacheline1与该way0 cacheline0存在主存地址连续的关系,连续的地址被串成一个指针链,way编号=4打断指针链。因此可依次读取连续地址标记来查找连续的地址块,如基于图7得到的主存地址连续的Cacheline,包括:

[0109] Way0 cache line0-way1 cache line1;way1 cache line0- way0 cache line1-way0 cacheline2- way1cache line3;way0 cache line3-way0 cache line4- way0 cacheline5- way0 cachelin6-way0cachelin7;way1 cache line4-way1 cache line5。

[0110] 可见,采用本公开的方式,在Cache中为每个缓存行配置对应的连续地址标记,该连续地址标记用于标记:与该缓存行存在地址连续关系的缓存行所在的Way,并且在Cache

中的缓存行发生更新时,对连续地址标记进行更新,以对Cache中的各个缓存行的主存地址连续性进行标记,进而在需要获取或确定主存地址连续的缓存行时,可以基于连续地址标记快速获取,有效提升了获取效率。

[0111] 可以理解,在主存地址中的bank地址以及row地址被映射为包含于Tag标记时,可以是在获取Tag标记后,从中获取bank地址以及row地址,基于bank地址以及row地址进行标记,即将上述实施例中的基于Tag标记相同进行标记,改为基于bank地址以及row地址相同进行标记,本公开对此不在进行详述。另外,可以理解的是,上述例子中虽然均以两个way为例进行说明,但是本公开可以适用于多个way的场景中,在连续地址标记中用不同的标记表示不同的way,用预定义的、不同于各个way的标记,来表示不存在连续地址。

[0112] 与前述一种Cache中地址标记方法相对应,如图9所示,本公开还提出一种Cache中地址标记装置,其中,Cache的映射方式被配置为组相联,Cache中的每一路中的每个缓存行被配置有对应的连续地址标记,任一缓存行对应的连续地址标记用于标记:与该缓存行存在地址连续关系的缓存行所在的路;其中,存在地址连续关系包括:任意两个缓存行对应的主存地址连续;该装置包括:

[0113] 获取模块910,用于在对目标缓存行进行更新后,确定更新后的目标缓存行的Tag标记,并获取与目标缓存行所在的目标组相邻的组之中的各缓存行的Tag标记;

[0114] 更新模块920,用于根据获取的Tag标记,对目标缓存行对应的连续地址标记,和/或,与目标缓存行所在的目标组相邻的组之中的缓存行的连续地址标记进行更新。

[0115] 在一个实施例中,主存地址中的存储库地址以及行地址被映射为Tag标记;更新模块920,具体用于若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记目标缓存行所在的路,并将其他第一缓存行的连续地址标记更新为不用于标记目标缓存行所在的路;其中,第一缓存行为目标组上一组中的缓存行;

[0116] 若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记第二缓存行所在的路;其中,第二缓存行为目标组下一组中的缓存行。

[0117] 在一个实施例中,更新模块920,具体用于若任一第一缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第一缓存行的连续地址标记更新为用于标记不存在连续地址的预定义标记。

[0118] 在一个实施例中,主存地址中的存储库地址以及行地址被映射为Tag标记;

[0119] 更新模块920,具体用于若更新后的目标缓存行的Tag标记和任一第二缓存行的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记该目标缓存行所在的路,并将其他第二缓存行的连续地址标记更新为不用于标记该目标缓存行所在的路;其中,该第二缓存行为目标组下一组中的缓存行;

[0120] 若更新后的目标缓存行的Tag标记和任一第一缓存行的Tag标记相同,则将该目标缓存行的连续地址标记更新为用于标记该第一缓存行所在的路;其中,该第一缓存行为目标组上一组中的缓存行。

[0121] 在一个实施例中,更新模块920,具体用于若任一第二缓存行的Tag标记与该目标缓存行更新前的Tag标记相同,则将该第二缓存行的连续地址标记更新为用于标记不存在

连续地址的预定义标记。

[0122] 本公开实施例还提供一种图形处理系统,包括上述地址标记装置。

[0123] 在一个具体的实施例中,本公开中所描述图形处理系统,如图10所示,其至少包括:

[0124] GPU core,用来处理命令,例如处理画图的命令,根据画图命令,执行图像渲染的Pipeline。其中,GPU core里面主要包含了计算单元,用于执行shader编译后的指令,属于可编程模块,由大量的ALU组成;Cache(存储器),用于缓存GPU core的数据,以减少对内存的访问;控制器(图中未示出)另外,GPU core 还具有多种功能模块,例如光栅化(3D渲染管线的一个固定的阶段)、Tiling(TBR和TBDR GPU架构中对一帧进行划片处理)、裁剪(3D渲染管线的一个固定的阶段,裁剪掉观察范围外,或者背面不显示的图元)、后处理(对画完的图进行缩放,裁剪,旋转等操作)等。

[0125] 通用DMA,用于执行主机内存到GPU显卡内存之间的数据搬移,例如,用于3D画图的vertex数据,通用DMA将vertex数据从主机内存搬到GPU显卡内存;

[0126] 片上网络,用于SOC上各个master和salve之间的数据交换;

[0127] 应用处理器,用于SOC上各个模块任务的调度,例如GPU渲染完一帧图后通知应用处理器,应用处理器再启动显示控制器将GPU画完的图显示到屏幕上;

[0128] PCIe控制器,用于和主机通信的接口,实现PCIe协议,使GPU显卡通过PCIe接口连接到主机上。主机上运行了图形API,以及显卡的驱动等程序;

[0129] 内存控制器,用于连接内存设备,用于保存SOC上的数据;

[0130] 地址标记装置,用于执行上文中所描述的地址标记方法;

[0131] 显示控制器,用于控制将内存里的frame buffer以显示接口(HDMI, DP等)输出到显示器上;

[0132] 视频解码器,用于将主机硬盘上的编码的视频解码成能显示的画面。

[0133] 视频编码器,用于将主机硬盘上原始的视频码流编码成指定的格式返回给主机。

[0134] 基于相同的发明构思,本公开实施例还提供一种电子组件,该电子组件包括上述任一实施例中所描述的图形处理系统。在一些使用场景下,该电子组件的产品形式体现为显卡;在另一些使用场景下,该电子装置的产品形式体现为CPU主板。

[0135] 本公开实施例还提供一种电子设备,该电子设备包括上述的图形处理系统。在一些使用场景下,该电子设备的產品形式是便携式电子设备,例如智能手机、平板电脑、VR设备等;在一些使用场景下,该电子设备的產品形式是个人电脑、游戏主机等。

[0136] 以上尽管已描述了本公开的优选实施例,但本领域内的技术人员一旦得知了基本创造性概念,则可对这些实施例做出另外的变更和修改。所以,所附权利要求意欲解释为包括优选实施例以及落入本公开范围的所有变更和修改,本领域的技术人员在不脱离本公开的精神和范围的前提下,做出的变更和修改也应视为本公开实施例的保护范围。

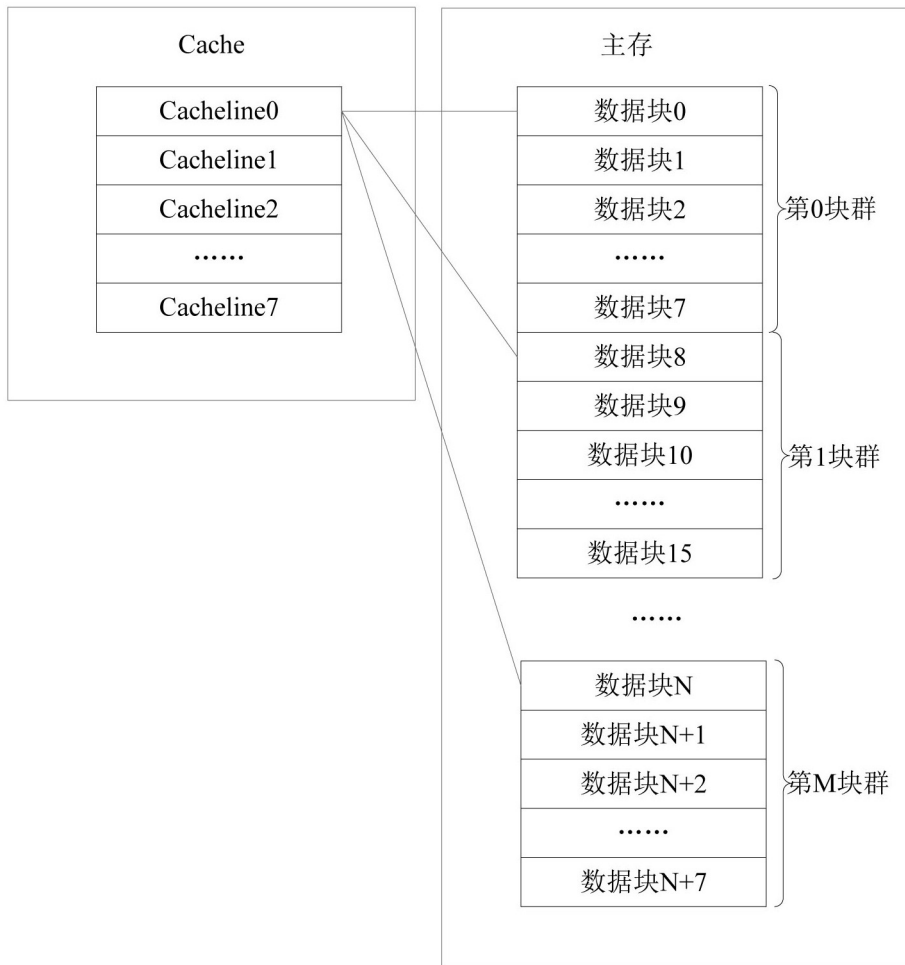


图1

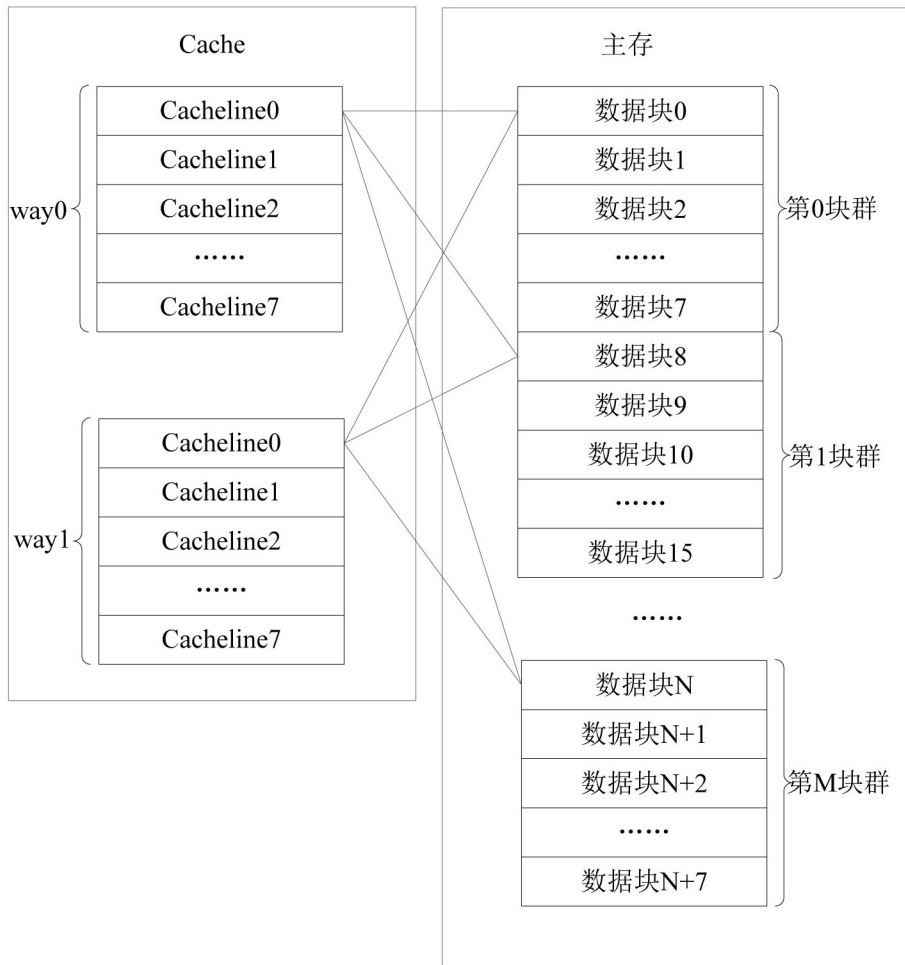


图2

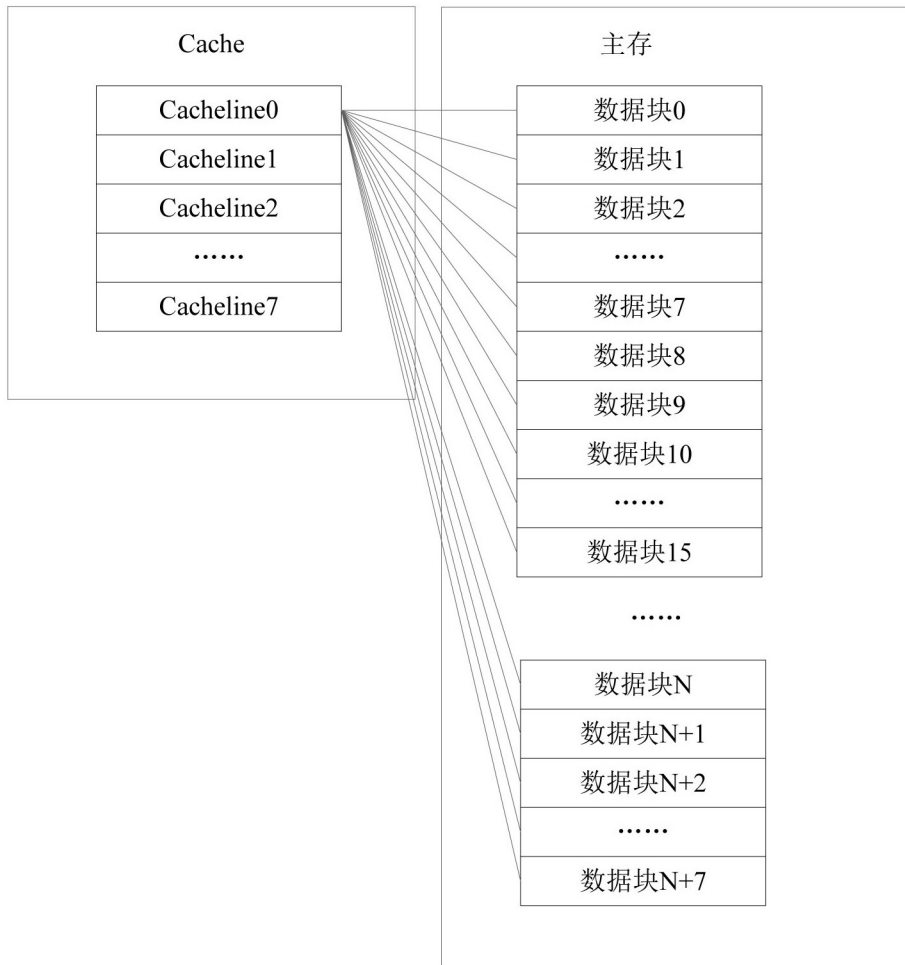


图3

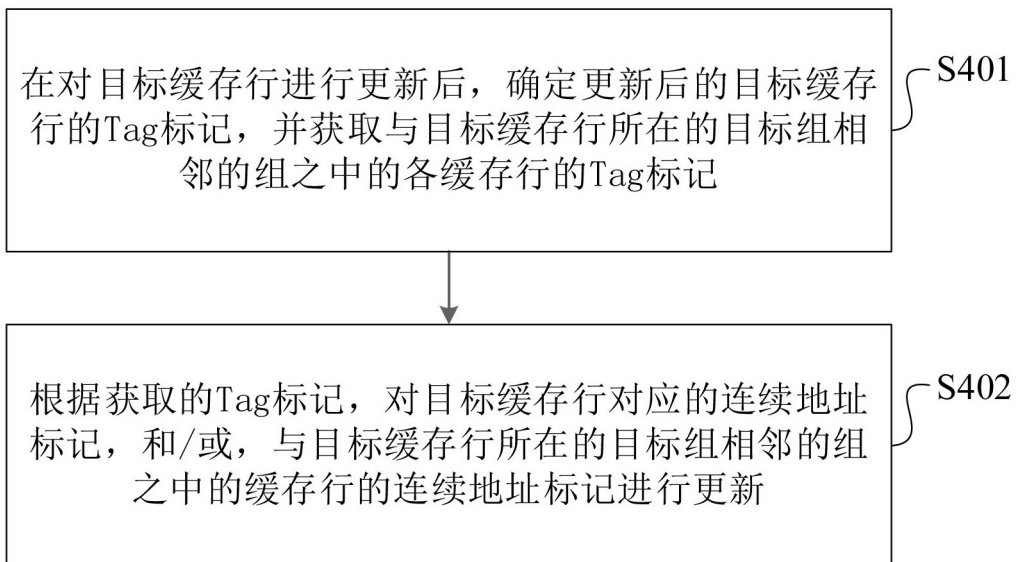


图4



图5

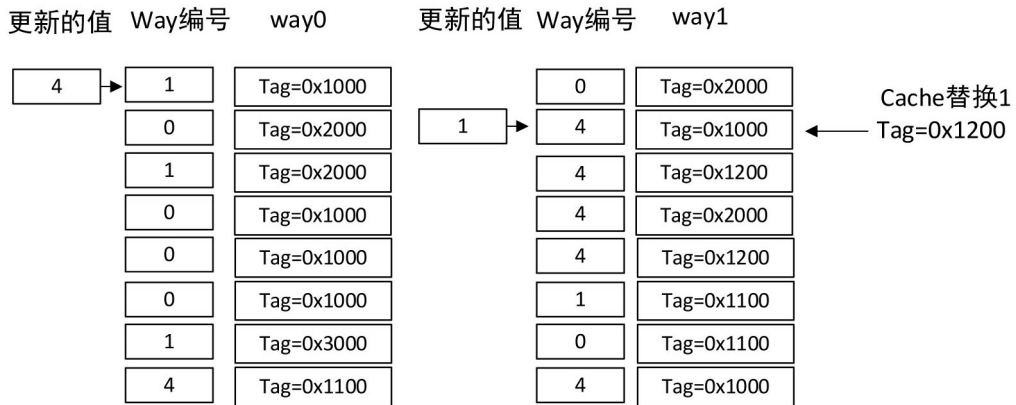


图6

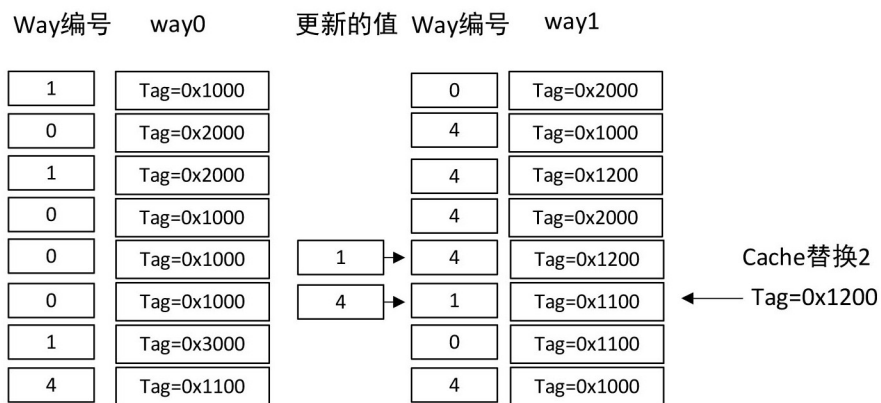


图7

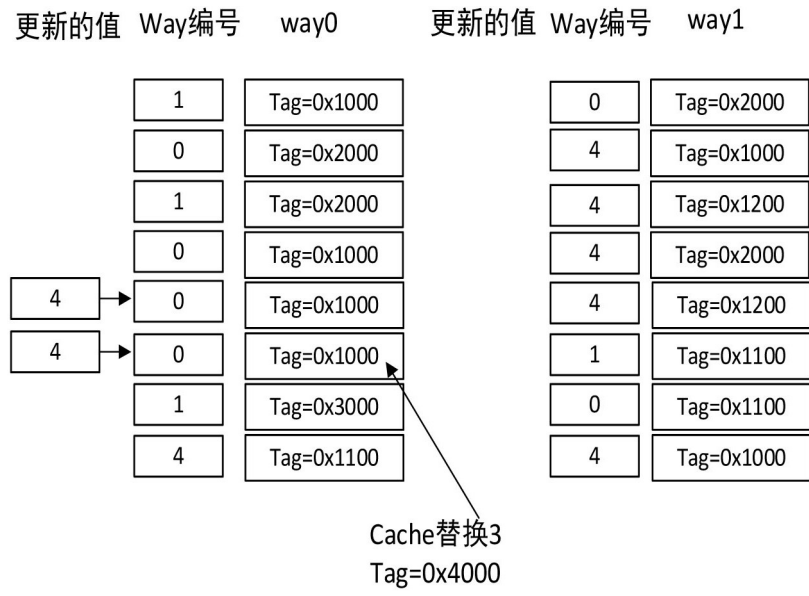


图8

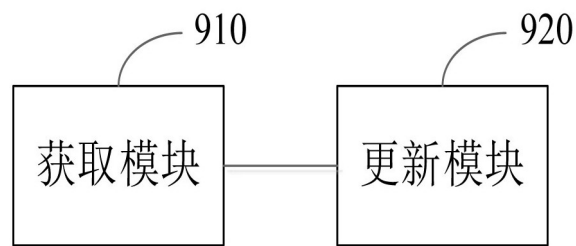


图9

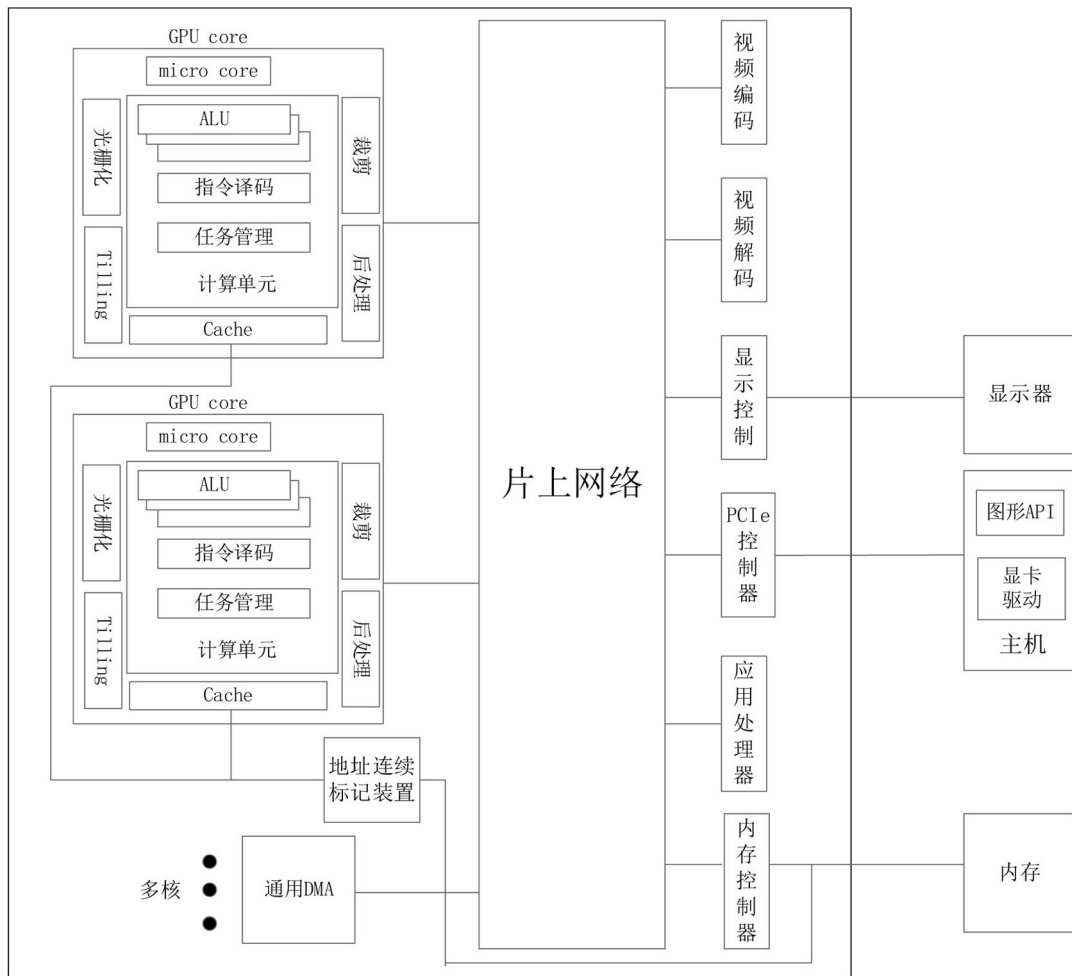


图10