



(12)发明专利

(10)授权公告号 CN 104252425 B

(45)授权公告日 2017.07.28

(21)申请号 201310269557.0

(22)申请日 2013.06.28

(65)同一申请的已公布的文献号  
申请公布号 CN 104252425 A

(43)申请公布日 2014.12.31

(73)专利权人 华为技术有限公司  
地址 518129 广东省深圳市龙岗区坂田华为总部办公楼  
专利权人 中国科学院计算技术研究所

(72)发明人 郭旭斌 侯锐 冯煜晶 苏东锋

(74)专利代理机构 北京中博世达专利商标代理有限公司 11274

代理人 申健

(51)Int.Cl.

G06F 9/54(2006.01)

(56)对比文件

CN 101510191 A,2009.08.19,权利要求4,说明书第3页第8段-第4页第2段.

US 2011/0320720 A1,2011.12.29,全文.

CN 103020003 A,2013.04.03,全文.

US 2002/0174285 A1,2002.11.21,说明书第[0008],[0011],[0020]-[0027],[0031]-[0034]段,附图2-5.

审查员 曾鹏飞

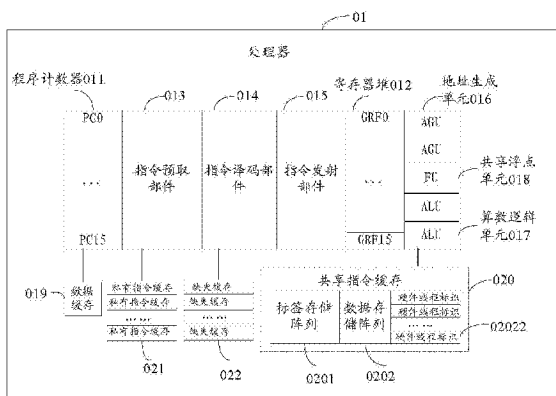
权利要求书2页 说明书9页 附图4页

(54)发明名称

一种指令缓存的管理方法和处理器

(57)摘要

本发明实施例提供一种指令缓存的管理方法和处理器,实际计算机领域,能够扩大硬件线程的指令缓存容量,降低指令缓存的缺失率,提高系统性能.该处理器的共享指令缓存中的硬件线程标识用于识别共享指令缓存中的缓存行对应的硬件线程,私有指令缓存用于存储从共享指令缓存中替换出的指令缓存行,还包括缺失缓存,当处理器的硬件线程在从指令缓存中获取指令时,同时访问指令缓存中的共享指令缓存和硬件线程对应的私有指令缓存,确定共享指令缓存和硬件线程对应的私有指令缓存是否存在指令,并根据判断结果从共享指令缓存或硬件线程对应的私有指令缓存中获取指令.本发明实施例用于管理处理器的指令缓存。



1. 一种处理器,其特征在于,包括程序计数器、寄存器堆、指令预取部件、指令译码部件、指令发射部件、地址生成单元、算术逻辑单元、共享浮点单元、数据缓存以及内部总线,其特征在于,还包括:

共享指令缓存,用于存储所有硬件线程的共享指令,包括标签存储阵列和数据存储阵列,所述标签存储阵列用于存储标签,所述数据存储阵列包括存储的指令和硬件线程标识,所述硬件线程标识用于识别所述共享指令缓存中的缓存行对应的硬件线程;

私有指令缓存,用于存储从所述共享指令缓存中替换出的指令缓存行,所述私有指令缓存与所述硬件线程一一对应;

缺失缓存,用于当所述共享指令缓存中不存在所取指令时,将从所述共享指令缓存的下一级缓存中取回的缓存行保存在所述硬件线程的缺失缓存中,在所述所取指令对应的硬件线程取指时,将所述缺失缓存中的缓存行回填至所述共享指令缓存中,所述缺失缓存与所述硬件线程一一对应。

2. 根据权利要求1所述的处理器,其特征在于,还包括:

标签比较逻辑,用于当所述硬件线程取指时,将所述硬件线程对应的私有指令缓存中的标签与翻译后援缓冲器转换的物理地址进行比较,所述私有指令缓存与所述标签比较逻辑相连,以使得所述硬件线程在访问所述共享指令缓存的同时访问所述私有指令缓存。

3. 根据权利要求2所述的处理器,其特征在于,所述处理器为多线程处理器,所述私有指令缓存的结构为全相联结构,所述全相联结构为主存储器中的任意一块指令缓存映射所述私有指令缓存中的任意一块指令缓存。

4. 根据权利要求3所述的处理器,其特征在于,所述共享指令缓存、私有指令缓存和所述缺失缓存为静态存储芯片或动态存储芯片。

5. 一种指令缓存的管理方法,其特征在于,包括:

当处理器的硬件线程在从指令缓存中获取指令时,同时访问所述指令缓存中的共享指令缓存和所述硬件线程对应的私有指令缓存;

确定所述共享指令缓存和所述硬件线程对应的私有指令缓存是否存在所述指令,并根据判断结果从所述共享指令缓存或所述硬件线程对应的私有指令缓存中获取所述指令;

其中,所述共享指令缓存包括标签存储阵列和数据存储阵列,所述标签存储阵列用于存储标签,所述数据存储阵列包括存储的指令和硬件线程标识,所述硬件线程标识用于识别所述共享指令缓存中的缓存行对应的硬件线程;

所述私有指令缓存的结构为全相联结构,所述全相联结构为主存储器中的任意一块指令缓存映射所述私有指令缓存中的任意一块指令缓存,所述私有指令缓存与所述硬件线程一一对应。

6. 根据权利要求5所述的方法,其特征在于,所述确定所述共享指令缓存和所述硬件线程对应的私有指令缓存是否存在所述指令,并根据判断结果从所述共享指令缓存或所述硬件线程对应的私有指令缓存中获取所述指令包括:

若所述共享指令缓存和所述硬件线程对应的私有指令缓存同时存在所述指令,则从所述共享指令缓存中获取所述指令;

若所述共享指令缓存中存在所述指令且所述硬件线程对应的私有指令缓存不存在所述指令,则从所述共享指令缓存中获取所述指令;

若所述硬件线程对应的私有指令缓存存在所述指令且所述共享指令缓存中不存在所述指令,则从所述硬件线程对应的私有指令缓存获取所述指令。

7. 根据权利要求6所述的方法,其特征在于,所述方法还包括:

若所述共享指令缓存和所述私有指令缓存都不存在所述指令,则通过所述硬件线程向所述共享指令缓存的下一级缓存发送缓存缺失请求;

若所述下一级缓存中存在所述指令,则通过所述硬件线程从所述下一级缓存中获取所述指令,并将所述指令所在的缓存行存储在所述硬件线程对应的缺失缓存中,在所述硬件线程取指时,将所述缓存行回填至所述共享指令缓存中;

若所述下一级缓存中不存在所述指令,则通过所述硬件线程向主存储器发送所述缺失请求,从所述主存储器中获取所述指令,并将所述指令所在的缓存行存储在所述硬件线程对应的缺失缓存中,在所述硬件线程取指时,将所述缓存行回填至所述共享指令缓存中;

其中,所述缺失缓存与所述硬件线程一一对应。

8. 根据权利要求7所述的方法,其特征在于,在将所述缓存行回填至所述共享指令缓存中时,若所述共享指令缓存不存在空闲资源,则将所述缓存行替换所述共享指令缓存中的第一缓存行,将所述缓存行回填至所述共享指令缓存中,同时根据获取所述第一缓存行的第一硬件线程的硬件线程标识,将所述第一缓存行存储在所述第一硬件线程对应的私有指令缓存中;

其中,所述第一缓存行是通过最近最少使用到算法确定的。

9. 根据权利要求8所述的方法,其特征在于,在将替换出来的第一缓存行存储在获取所述第一硬件线程对应的私有指令缓存中时,若所述第一硬件线程对应的私有指令缓存不存在空闲资源,则将所述第一缓存行替换所述第一硬件线程对应的私有指令缓存中的第二缓存行,将所述第一缓存行回填至所述第一硬件线程对应的私有指令缓存中;

其中,所述第二缓存行是通过所述最近最少使用到算法确定的。

## 一种指令缓存的管理方法和处理器

### 技术领域

[0001] 本发明涉及计算机领域,尤其涉及一种指令缓存的管理方法和处理器。

### 背景技术

[0002] CPU (Central Processing Unit,中央处理机) 缓存 (Cache Memory) 是位于CPU与内存之间的临时存储器,容量比内存小得多,解决了CPU运算速度与内存读写速度不匹配的矛盾,加快了CPU的读取速度。

[0003] 在多线程处理器中,多个硬件线程从同一块I-Cache (指令缓存) 中获取指令,当I-Cache中不存在所要获取的指令时,在向下一级Cache发送缺失请求的同时,切换到其他硬件线程访问I-Cache继续取指,减少了流水线由于I-Cache缺失所导致的停顿,提高了流水线效率。但是由于每个硬件线程分到的共享I-Cache资源不足时,I-Cache缺失率增大,I-Cache发往下一级Cache的缺失请求会频繁发生,且从下一级Cache取回指令回填时,在线程数据增多时,会导致填入的指令所在的Cache行立即填入到缺失的I-Cache中不会立即用到,而替换出的Cache行反而有可能被再次使用。

[0004] 另外,在根据Cache命中情况来调整Thread (线程) 的调度策略时,会尽量在一段时间内保证优先调度访存指令在Cache中命中率高的线程,但是对于每个硬件线程分到的共享I-Cache资源不足的问题并没有得到改善。

### 发明内容

[0005] 本发明的实施例提供一种指令缓存的管理方法和处理器,能够扩大硬件线程的指令缓存容量,降低指令缓存的缺失率,提高系统性能。

[0006] 为达到上述目的,本发明的实施例采用如下技术方案

[0007] 第一方面,提供一种处理器,其特征在于,包括程序计数器、寄存器堆、指令预取部件、指令译码部件、指令发射部件、地址生成单元、算术逻辑单元、共享浮点单元、数据缓存以及内部总线,还包括:

[0008] 共享指令缓存,用于存储所有硬件线程的共享指令,包括标签存储阵列和数据存储阵列,所述标签存储阵列用于存储标签,所述数据存储阵列包括存储的指令和硬件线程标识,所述硬件线程标识用于识别所述共享指令缓存中的缓存行对应的硬件线程;

[0009] 私有指令缓存,用于存储从所述共享指令缓存中替换出的指令缓存行,所述私有指令缓存与所述硬件线程一一对应;

[0010] 缺失缓存,用于当所述共享指令缓存中不存在所取指令时,将从所述共享指令缓存的下一级缓存中取回的缓存行保存在所述硬件线程的缺失缓存中,在所述所取指令对应的硬件线程取指时,将所述缺失缓存器中的缓存行回填至所述共享指令缓存中,所述缺失缓存与所述硬件线程一一对应。

[0011] 结合第一方面,在第一方面的第一种可能实现的方式中,还包括:

[0012] 标签比较逻辑,用于当所述硬件线程取指时,将所述硬件线程对应的私有指令缓

存中的标签与翻译后援缓冲器转换的物理地址进行比较,将所述私有指令缓存与所述标签比较逻辑相连,以使得所述硬件线程在访问所述共享指令缓存的同时访问所述私有指令缓存。

[0013] 结合第一方面的第一种可能实现的方式,在第二种可能实现的方式中,所述处理器为多线程处理器,所述私有指令缓存的结构为全相联结构,所述全相联结构为主存储器中的任意一块指令缓存映射所述私有指令缓存中的任意一块指令缓存。

[0014] 结合第一方面的第二种可能实现的方式,在第三种可能实现的方式中,所述共享指令缓存、私有指令缓存和所述缺失缓存为静态存储芯片或动态存储芯片。

[0015] 第二方面,提供一种指令缓存的管理方法,包括:

[0016] 当处理器的硬件线程在从指令缓存中获取指令时,同时访问所述指令缓存中的共享指令缓存和所述硬件线程对应的私有指令缓存;

[0017] 确定所述共享指令缓存和所述硬件线程对应的私有指令缓存是否存在所述指令,并根据判断结果从所述共享指令缓存或所述硬件线程对应的私有指令缓存中获取所述指令。

[0018] 结合第二方面,在第二方面的第一种可能实现的方式中,所述共享指令缓存包括标签存储阵列和数据存储阵列,所述标签存储阵列用于存储标签,所述数据存储阵列包括存储的指令和硬件线程标识,所述硬件线程标识用于识别所述共享指令缓存中的缓存行对应的硬件线程;

[0019] 所述私有指令缓存的结构为全相联结构,所述全相联结构为主存储器中的任意一块指令缓存映射所述私有指令缓存中的任意一块指令缓存,所述私有指令缓存与所述硬件线程一一对应。

[0020] 结合第二方面的第一种可能实现的方式,在第二种可能实现的方式中,所述确定所述共享指令缓存和所述硬件线程对应的私有指令缓存是否存在所述指令,并根据判断结果从所述共享指令缓存或所述硬件线程对应的私有指令缓存中获取所述指令包括:

[0021] 若所述共享指令缓存和所述硬件线程对应的私有指令缓存同时存在所述指令,则从所述共享指令缓存中获取所述指令;

[0022] 若所述共享指令缓存中存在所述指令且所述硬件线程对应的私有指令缓存不存在所述指令,则从所述共享指令缓存中获取所述指令;

[0023] 若所述硬件线程对应的私有指令缓存存在所述指令且所述共享指令缓存中不存在所述指令,则从所述硬件线程对应的私有指令缓存获取所述指令。

[0024] 结合第二方面的第二种可能实现的方式,在第三种可能实现的方式中,所述方法还包括:

[0025] 若所述共享指令缓存和所述私有指令缓存都不存在所述指令,则通过所述硬件线程向所述共享指令缓存的下一级缓存发送缓存缺失请求,

[0026] 若所述下一级缓存中存在所述指令,则通过所述硬件线程从所述下一级缓存中获取所述指令,并将所述指令所在的缓存行存储在所述硬件线程对应的缺失缓存中,在所述硬件线程取指时,将所述缓存行回填至所述共享指令缓存中;

[0027] 若所述下一级缓存中不存在所述指令,则通过所述硬件线程向主存储器发送所述缺失请求,从所述主存储器中获取所述指令,并将所述指令所在的缓存行存储在所述硬件

线程对应的缺失缓存中,在所述硬件线程取指时,将所述缓存行回填至所述共享指令缓存中;

[0028] 其中,所述缺失缓存与所述硬件线程一一对应。

[0029] 结合第二方面的第三种可能实现的方式,在第四种可能实现的方式中,在将所述缓存行回填至所述共享指令缓存中时,若所述共享指令缓存不存在空闲资源,则将所述缓存行替换所述共享指令缓存中的第一缓存行,将所述缓存行回填至所述共享指令缓存中,同时根据获取所述第一缓存行的第一硬件线程的硬件线程标识,将所述第一缓存行存储在所述第一硬件线程对应的私有指令缓存中;

[0030] 其中,所述第一缓存行是通过最近最少使用到算法确定的。

[0031] 结合第二方面的第四种可能实现的方式,在第五种可能实现的方式中,在将替换出来的第一缓存行存储在获取所述第一硬件线程对应的私有指令缓存中时,若所述第一硬件线程对应的私有指令缓存不存在空闲资源,则将所述第一缓存行替换所述第一硬件线程对应的私有指令缓存中的第二缓存行,将所述第一缓存行回填至所述第一硬件线程对应的私有指令缓存中;

[0032] 其中,所述第二缓存行是通过所述最近最少使用到算法确定的。

[0033] 本发明实施例提供一种指令缓存的管理方法和处理器,处理器包括程序计数器、寄存器堆、指令预取部件、指令译码部件、指令发射部件、地址生成单元、算术逻辑单元、共享浮点单元、数据缓存以及内部总线,还包括共享指令缓存,私有指令缓存,缺失缓存和标签比较逻辑。其中,共享指令缓存,用于存储所有硬件线程的共享指令,包括标签存储阵列和数据存储阵列,数据存储阵列包括存储的指令和硬件线程标识,硬件线程标识用于识别共享指令缓存中的缓存行对应的硬件线程,私有指令缓存,用于存储从共享指令缓存中替换出的指令缓存行,私有指令缓存与硬件线程一一对应,标签比较逻辑,用于当硬件线程取指时,将该硬件线程对应的私有指令缓存中的标签与翻译后援缓冲器转换的物理地址进行比较,私有指令缓存与标签比较逻辑相连,以使得硬件线程在访问共享指令缓存的同时访问私有指令缓存,当处理器的硬件线程在从指令缓存中获取指令时,同时访问指令缓存中的共享指令缓存和硬件线程对应的私有指令缓存,确定共享指令缓存和硬件线程对应的私有指令缓存是否存在指令,并根据判断结果从共享指令缓存或硬件线程对应的私有指令缓存中获取指令,能够扩大硬件线程的指令缓存容量,降低指令缓存的缺失率,提高系统性能。

## 附图说明

[0034] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0035] 图1为本发明实施例提供的一种处理器结构示意图;

[0036] 图2为本发明实施例提供的一种指令缓存的管理方法流程示意图;

[0037] 图3为本发明实施例提供的一种同时访问共享指令缓存和私有指令缓存的逻辑示意图;

[0038] 图4为本发明实施例提供的一种根据缓存缺失请求取回缓存行时的逻辑示意图。

### 具体实施方式

[0039] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例,都属于本发明保护的范围。

[0040] 在现代多线程处理器设计中,随着硬件线程数目的增多,每个硬件线程所能分到的共享资源会存在不足,例如,对于Cache(缓存)中的L1(Level1)Cache这一重要共享资源更是如此。每个硬件线程分到的L1Cache的指令缓存容量过小,L1中会存在不命中的情况,L1缺失率增高,导致L1Cache与L2Cache通信增多而从L2Cache中取指,或者从主存储器中取指,处理器功耗增大。

[0041] 本发明实施例提供一种处理器01,如图1所示,包括程序计数器011、寄存器堆012、指令预取部件013、指令译码部件014、指令发射部件015、地址生成单元016、算术逻辑单元017、共享浮点单元018、数据缓存019、以及内部总线,还包括:

[0042] 共享指令缓存(I-Cache)020,私有指令缓存021,缺失缓存(Miss Buffer)022和标签(Tag)比较逻辑023。

[0043] 其中,共享指令缓存020,用于存储所有硬件线程的共享指令,包括标签存储阵列(Tag Array)0201和数据存储阵列(Data Array)0202,标签存储阵列0201用于存储标签,数据存储阵列0202包括存储的指令02021和硬件线程标识(Thread ID)02022,硬件线程标识02022用于识别共享指令缓存020中的缓存行对应的硬件线程。

[0044] 私有指令缓存021,用于存储从共享指令缓存020中替换出的指令缓存行,私有指令缓存021与硬件线程一一对应。

[0045] 缺失缓存022,用于当共享指令缓存020中不存在所取指令时,将从共享指令缓存020的下一级缓存中取回的缓存行进行缓存在硬件线程的缺失缓存中,在所取指令对应的硬件线程取指时,将缺失缓存022中的缓存行回填至共享指令缓存中,缺失缓存022与硬件线程一一对应。

[0046] 标签比较逻辑,用于当硬件线程取指时,将该硬件线程对应的私有指令缓存中的标签与TLB(Translation Look-aside Buffers,翻译后援缓冲器)转换的PA(Physis Adress)进行比较,私有指令缓存021与标签比较逻辑相连,以使得硬件线程在访问共享指令缓存020的同时访问私有指令缓存021。

[0047] 其中,TLB或称为页表缓冲,里面存放的是一些页表文件(虚拟地址到物理地址的转换表,可以将所取指令的虚拟地址通过TLB转换为物理地址,在将物理地址与私有指令缓存中的标签进行比较后,若物理地址与私有指令缓存中的标签相同,使得硬件线程在访问共享指令缓存的同时也访问私有指令缓存。

[0048] 举例来说,PC(Program Counter,程序计数器)有16个,为PC0~PC15,一个处理器核内逻辑处理器核(硬件线程)的个数与PC的个数是一致的。

[0049] GRF(General Register File,寄存器堆),一个处理器核内的逻辑处理器核对应一个GRF,数量上与PC的数量一致。

[0050] Fetch (指令预取部件) 用于获取指令, Decoder (指令译码部件) 用于对指令进行解码, Issue 为指令发射部件, 用于发射指令, AGU (Address Generator Unit, 地址生成单元) 为进行所有地址计算的模块, 生成一个用于对访问存储器进行控制的地址。ALU (Arithmetic Logic Unit, 算术逻辑单元) 是 CPU (Central Processing Unit, 中央处理器) 的执行单元, 可以由“And Gate” (与门) 和“Or Gate” (或门) 构成的算术逻辑单元。共享浮点单元 (Shared Float Point Unit) 为处理器中专门进行浮点算术运算的电路单元, 数据缓存 (D-Cache) 用于存储数据, 内部总线用于连接处理器内各部件。

[0051] 举例来说, 处理器 01 为多线程处理器, 私有指令缓存 021 的结构为全相联结构, 全相联结构为主存储器中的任意一块指令缓存映射私有指令缓存中的任意一块指令缓存。

[0052] 举例来说, 共享指令缓存 020、私有指令缓存 021 和缺失缓存 022 为静态存储芯片或动态存储芯片。

[0053] 举例来说, 可以在 I-Cache Data Array (指令高速缓存数据存储阵列) 中新增 Thread ID (硬件线程标识), 该 Thread ID 用于表示 Cache Line 是由哪一个硬件线程发出的 Cache Miss (缓存不命中) 请求所取回的。

[0054] 例如, 当硬件线程访问 L1 的 I-Cache 不命中, 即 I-Cache 中不存在硬件线程所要获得的指令时, L1 向 L1 的下一级缓存 L2Cache 发送 Cache Miss 请求, 若 L2Cache 命中, 即 L2Cache 中存在该硬件线程所要获得的指令时, 该硬件线程将 L2Cache 中该指令所在的缓存行 (Cache Line) 回填至 L1Cache 中, 也可以是该硬件线程在收到返回的 Cache Line 时, 不直接将该 Cache Line 填入 L1Cache 中, 而是将该 Cache Line 保存在该硬件线程对应的 Miss Buffer 中, 直到轮到该硬件线程取指时, 将该 Cache Line 填入 L1Cache 中。

[0055] 这样, 当该硬件线程将 L2Cache 中该指令所在的缓存行回填至 L1Cache 中发生替换时, 不直接丢弃掉替换出的 Cache Line, 可以根据替换出的 Cache Line 对应的硬件线程的 Thread ID, 将替换出的 Cache Line 填入该替换出的 Cache Line 对应的硬件线程对应的私有指令缓存中。

[0056] 举例来说, 发生替换可以是由于 L1Cache 中不存在空闲资源导致的, 替换出的 Cache Line 可以根据 LRU (Least Recently Used, 最近最少使用到的) 算法得到。

[0057] 其中, LRU 算法就是一旦指令缓存出现缺失, 则将未被使用到的时间最长的一条指令替换出缓存, 换句话说, 缓存首先保留的是最近常被使用到的指令。

[0058] 举例来说, 一个硬件线程在取指时, 可以同时访问 I-Cache 和该硬件线程对应的私有 Cache。

[0059] 若 I-Cache 中存在所取指令而该硬件线程对应的私有 Cache 不存在所取指令, 则从 I-Cache 中获得所取指令;

[0060] 若 I-Cache 中不存在所取指令而该硬件线程对应的私有 Cache 存在所取指令, 则从该硬件线程对应的私有 Cache 存在所取指令获得所取指令;

[0061] 若 I-Cache 和该硬件线程对应的私有 Cache 中同时存在所取指令, 则从 I-Cache 中获得所取指令;

[0062] 若 I-Cache 和该硬件线程对应的私有 Cache 中都不存在所取指令, 则该硬件线程向 I-Cache 的下一级缓存发送 Cache Miss 请求, 以获得所取指令。

[0063] 举例来说, 根据硬件线程的调取策略, 当下一个周期切换到其它线程取指时, 在访

问共享指令缓存的同时,将新的硬件线程对应的私有Cache与Tag(标签)比较逻辑相连,将该私有Cache读出的Tag比较逻辑与TLB(Translation Look-aside Buffers,转译后备缓冲区)输出的PA(Physics Address,物理地址)进行比较,生成私有Cache Miss信号和私有Cache数据输出。当该新的硬件线程对应的私有Cache中存在所取指令,私有Cache Miss信号表示存在指令,并有指令输出。

[0064] 因此,本发明实施例提供一种处理器,该处理器包括程序计数器、寄存器堆、指令预取部件、指令译码部件、指令发射部件、地址生成单元、算术逻辑单元、共享浮点单元、数据缓存以及内部总线,还包括共享指令缓存,私有指令缓存,缺失缓存和标签比较逻辑,在共享指令缓存的数据存储阵列中增加硬件线程标识,用来在缓存缺失时,取回的缓存行是由哪一个硬件线程发出的缓存缺失请求取回的,当共享指令缓存发生替换时,将替换出的缓存行根据硬件线程标识存入到对应的硬件线程对应的私有指令缓存中,且缺失缓存用于硬件线程在收到缓存缺失请求返回的缓存行时,不直接将缓存行回填至共享指令缓存中,而是将缓存行保存在缺失缓存中,直到轮到该硬件线程取指时,将缓存行回填至共享指令缓存中,降低了即将被访问到底缓存行被替换出的指令缓存的几率,另外,增加的私有指令缓存,增大了每个硬件线程的缓存容量,提高了系统性能。

[0065] 本发明又一实施例提供一种指令缓存的管理方法,如图2所示,包括:

[0066] 101、当处理器的硬件线程在从指令缓存中获取指令时,处理器同时访问指令缓存中的共享指令缓存和硬件线程对应的私有指令缓存。

[0067] 示例性的,该处理器(Central Processing Unit,CPU)可以为多线程处理器。一个物理内核可以有多个硬件线程,也称为逻辑内核或逻辑处理器,但是一个硬件线程并不代表一个物理内核,Windows将每一个硬件线程是被为一个可调度的逻辑处理器,每一个逻辑处理器可以运行软件线程的代码。指令缓存可以为处理器中的L1Cache中的共享指令缓存(I-Cache)和硬件线程的私有指令缓存。其中L1Cache包括数据缓存(D-Cache)和指令缓存(I-Cache)。

[0068] 具体的,可以在每个硬件线程中设置一块全相联的私有Cache,即私有Cache与硬件线程一一对应。其中,全相联结构为主存储器中的任意一块指令缓存映射私有指令缓存中的任意一块指令缓存。

[0069] 另外,还可以增加一个Tag(标签)比较逻辑,当硬件线程取指时,主动将硬件线程的私有Cache与Tag比较逻辑相连,这样,当一个硬件线程取指时,同时访问I-Cache和该硬件线程对应的私有Cache。

[0070] 102、处理器确定共享指令缓存和硬件线程对应的私有指令缓存是否存在指令,而后进入步骤103或104或105或106。

[0071] 示例性的,硬件线程同时访问I-Cache和该硬件线程对应的私有Cache时,同时判断I-Cache和该硬件线程对应的私有Cache是否存在所取指令。

[0072] 设若该多线程处理器有32个硬件线程,该32个硬件线程共用一块64KB的I-Cache,即共享指令缓存容量为64KB。每个硬件线程包含有一块32路全相联的私有Cache,可以存储32条替换出来的Cache Line(缓存行),每一条Cache Line包含64Bytes,这样,每一块私有Cache容量为2KB。

[0073] 当新增一个32路Tag比较逻辑时,硬件线程在访问I-Cache共享指令缓存的同时,

将该硬件线程读出的32路Tag与TLB输出的PA (Physics Address,物理地址) 进行比较,并生成私有Cache Miss信号和私有Cache数据输出。若32路Tag与PA相同,则私有Cache Miss信号表示该硬件线程的私有Cache存在所取指令,私有Cache数据为有效指令。如图3所示。

[0074] 其中,TLB或称为页表缓冲,里面存放的是一些页表文件(虚拟地址到物理地址的转换表,可以将所取指令的虚拟地址通过TLB转换为物理地址,在将物理地址与私有指令缓存中的标签进行比较后,若物理地址与私有指令缓存中的标签相同,使得硬件线程在访问共享指令缓存的同时也访问私有指令缓存。

[0075] 103、若共享指令缓存和硬件线程对应的私有指令缓存同时存在指令,则处理器从共享指令缓存中获取指令。

[0076] 示例性的,在同时对I-Cache和私有Cache进行访问时,若I-Cache和私有Cache同时存在所取指令,则从I-Cache中取得指令。

[0077] 104、若共享指令缓存中存在指令且硬件线程对应的私有指令缓存不存在指令,则处理器从共享指令缓存中获取指令。

[0078] 示例性的,若I-Cache中存在所取指令,硬件线程的私有Cache不存在所取指令,则从I-Cache中存在所取指令。

[0079] 105、若硬件线程对应的私有指令缓存存在指令且共享指令缓存中不存在指令,则处理器从硬件线程对应的私有指令缓存获取指令。

[0080] 示例性的,若I-Cache中不命中,即不存在所取指令,硬件线程对应的私有Cache中存在所取指令,则从硬件线程对应的私有Cache中取得指令。这样,通过主动选取硬件线程对应的私有Cache参与Tag比较,可以扩大每个硬件线程分到的Cache容量,增大了硬件线程的指令缓存的命中率。

[0081] 106、若共享指令缓存和私有指令缓存都不存在指令,则处理器通过硬件线程向共享指令缓存的下一级缓存发送缓存缺失请求。

[0082] 示例性的,若I-Cache和硬件线程对应的私有Cache都不存在所取指令,则该硬件线程向I-Cache的下一级缓存发出Cache Miss(缓存缺失)。

[0083] 例如,L1Cache和硬件线程对应的私有Cache都不存在所取指令,则硬件线程向L1Cache的下一级缓存L2Cache发出Cache Miss,以求从L2Cache中获得所取指令。

[0084] 107、若下一级缓存中存在指令,则处理器通过硬件线程从下一级缓存中获取指令,并将指令所在的缓存行存储在硬件线程对应的缺失缓存中,在硬件线程取指时,将缓存行回填至共享指令缓存中。

[0085] 示例性的,当L2Cache中存在所取指令,则从L2Cache中获得指令,并不直接将该指令所在的Cache Line回填至L1Cache中,而是将所取指令所在的Cache Line保存在该硬件线程对应的Miss Buffer(缺失缓存)中,直到轮到该硬件线程取指时,将该Cache Line填入L1Cache中。

[0086] 其中,Miss Buffer与硬件线程一一对应,即每个硬件线程都有一个Miss Buffer,每个硬件线程都利用一个Miss Buffer来缓存Cache Miss请求返回的Cache Line,这是由于假设将该Cache Line回填至L1Cache时发生替换,替换出的Cache Line有可能是即将被访问到的Cache Line,Miss Buffer的存在优化了Cache Line的回填时机,降低了将被访问到的Cache Line被替换出的缓存的几率。

[0087] 108、若下一级缓存中不存在指令,则处理器通过硬件线程向主存储器发送缺失请求,从主存储器中获取指令,并将指令所在的缓存行存储在硬件线程对应的缺失缓存中,在硬件线程取指时,将缓存行回填至共享指令缓存中。

[0088] 示例性的,若L2Cache中也不存在所取指令,则硬件线程向主存储器发出Cache Miss请求,以求从主存储器中获得所取指令。若主存储器中存在所取指令,则获得所取指令,并将所取指令所在的Cache Line保存在该硬件线程对应的Miss Buffer中,直到轮到该硬件线程取指时,将该Cache Line填入L1Cache中。

[0089] 也可以是L2Cache中也不存在所取指令时,硬件线程向L3Cache发出Cache Miss请求,若L3Cache中存在所取指令,则获得所取指令,若L3Cache中不存在所取指令,则向主存储器发出Cache Miss请求以获得所取指令。

[0090] 其中,CPU与Cache之间的交换单位为字,当CPU要读取主存中一个字时,发出该字的内存地址同时到达Cache和主存,L1Cache或L2Cache或L3Cache可以在Cache控制逻辑依据地址的Tag标记部分判断是否存在字,若命中,CPU获得该字,若未命中,则要用主存读取周期从主存中读出并输出至CPU,即使当前CPU仅读一个字,Cache控制器也要把主存储器中包含该字的一个完整的Cache行复制到Cache中,这种向Cache传送一行数据的操作就称为Cache行填充。

[0091] 另外,在将缓存行回填至共享指令缓存中时,若共享指令缓存不存在空闲资源,则将缓存行替换共享指令缓存中的第一缓存行,将缓存行回填至共享指令缓存中,同时根据获取第一缓存行的第一硬件线程的硬件线程标识,将第一缓存行存储在在第一硬件线程对应的私有指令缓存中。其中,第一缓存行是通过LRU (Least Recently Used,最近最少使用到的)算法确定的。

[0092] 示例性的,可以在I-Cache Data Array (指令高速缓存数据存储阵列)中增加Thread ID (硬件线程标识),该硬件线程标识用于表示一条Cache Line是由哪一个硬件线程发出的Cache Miss请求所取回的。这样,当在每个硬件线程中设置一块全相联的私有Cache后,当I-Cache发生替换时,不直接丢弃掉替换出的Cache Line,而可以根据该Thread ID,将替换出的Cache Line填入Thread ID标识的硬件线程的私有Cache中,这是由于替换出的Cache Line有很快要被访问的可能性。如图4所示。

[0093] 在将替换出来的第一缓存行存储在获取第一硬件线程对应的私有指令缓存中时,若第一硬件线程对应的私有指令缓存不存在空闲资源,则将第一缓存行替换第一硬件线程对应的私有指令缓存中的第二缓存行,将第一缓存行回填至第一硬件线程对应的私有指令缓存中。其中,第二缓存行是通过LRU算法确定的。

[0094] 其中,LRU算法就是一旦指令缓存出现缺失,则将未被使用到的时间最长的一条指令替换出缓存,换句话说,缓存首先保留的是最近常被使用到的指令。

[0095] 这样一来,通过增加私有Cache有效的扩大了每个硬件线程分到的指令Cache容量,增大了硬件线程的指令Cache的命中率,减少了I-Cache与下一级Cache之间的通信,同时通过增加的Miss Buffer优化了Cache Line的回填时机,降低了即将被访问到的Cache Line被替换出的几率,增加的Tag比较逻辑,使得访问I-Cache时同时访问共享指令缓存和私有指令缓存,增加了指令缓存的命中率。

[0096] 本发明实施例提供一种指令缓存的管理方法,当处理器的硬件线程在从指令缓存

中获取指令时,同时访问指令缓存中的共享指令缓存和硬件线程对应的私有指令缓存,确定共享指令缓存和硬件线程对应的私有指令缓存是否存在指令,并根据判断结果从共享指令缓存或硬件线程对应的私有指令缓存中获取指令,若共享指令缓存和私有指令缓存都不存在指令,则通过硬件线程向共享指令缓存的下一级缓存发送缓存缺失请求,并将指令所在的缓存行存储在硬件线程对应的缺失缓存中,在硬件线程取指时,将缓存行回填至共享指令缓存中,在将缓存行回填至共享指令缓存中时,若共享指令缓存不存在空闲资源,则将缓存行替换共享指令缓存中的第一缓存行,将缓存行回填至共享指令缓存中,同时根据获取第一缓存行的第一硬件线程的硬件线程标识,将第一缓存行存储在第一硬件线程对应的私有指令缓存中,这样能够扩大硬件线程的指令缓存容量,降低指令缓存的缺失率,提高系统性能。

[0097] 在本申请所提供的几个实施例中,应该理解到,所揭露的处理器和方法,可以通过其它的方式实现。例如,以上所描述的设备实施例仅仅是示意性的,例如,所述单元的划分,仅仅为一种逻辑功能划分,实际实现时可以有另外的划分方式,例如多个单元或组件可以结合或者可以集成到另一个系统,或一些特征可以忽略,或不执行。另一点,所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口,装置或单元的间接耦合或通信连接,可以是电性,机械或其它的形式。

[0098] 另外,在本发明各个实施例中的设备和系统中,各功能单元可以集成在一个处理单元中,也可以是各个单元单独物理包括,也可以两个或两个以上单元集成在一个单元中。且上述的各单元既可以采用硬件的形式实现,也可以采用硬件加软件功能单元的形式实现。

[0099] 实现上述方法实施例的全部或部分步骤可以通过程序指令相关的硬件来完成,前述的程序可以存储于一计算机可读取存储介质中,该程序在执行时,执行包括上述方法实施例的步骤;而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read Only Memory,简称ROM)、随机存取存储器(Random Access Memory,简称RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0100] 以上所述,仅为本发明的具体实施方式,但本发明的保护范围并不局限于此,任何熟悉本技术领域的技术人员在本发明揭露的技术范围内,可轻易想到变化或替换,都应涵盖在本发明的保护范围之内。因此,本发明的保护范围应以所述权利要求的保护范围为准。

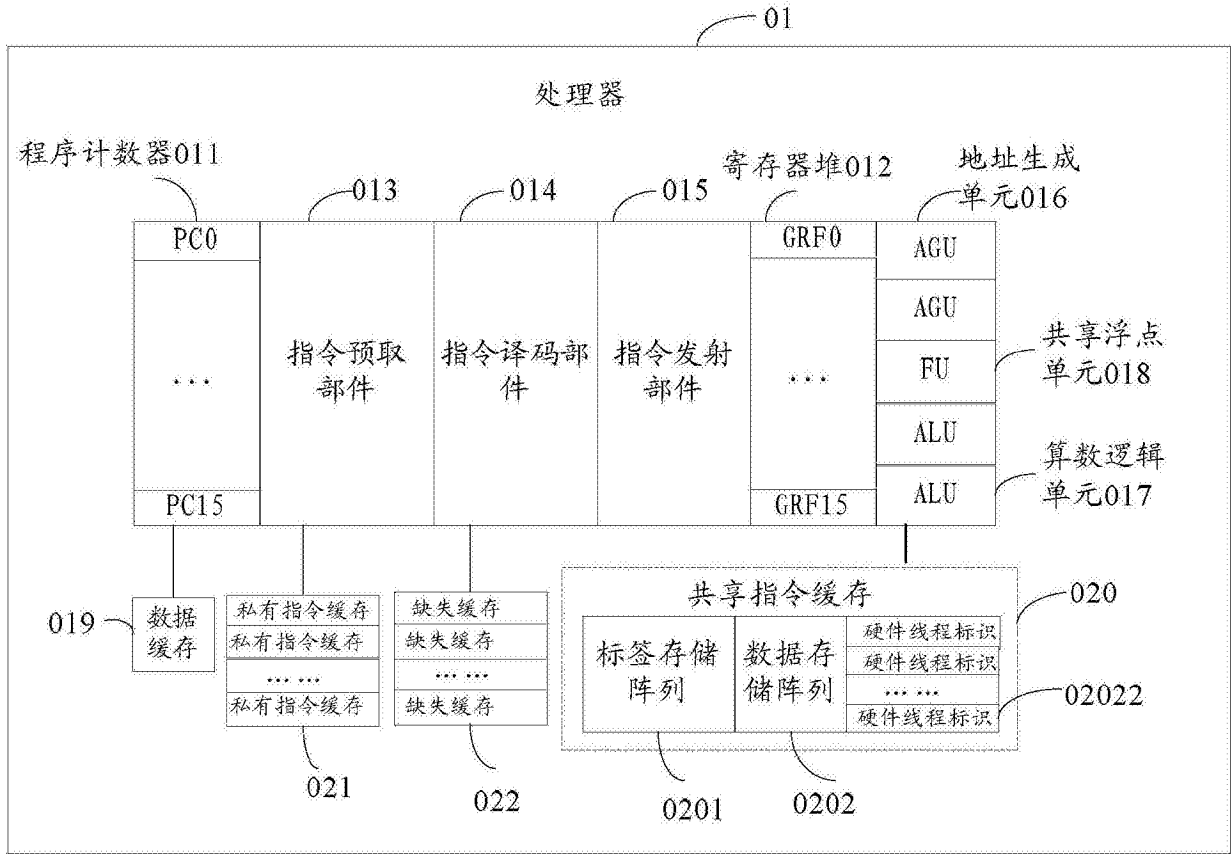


图1

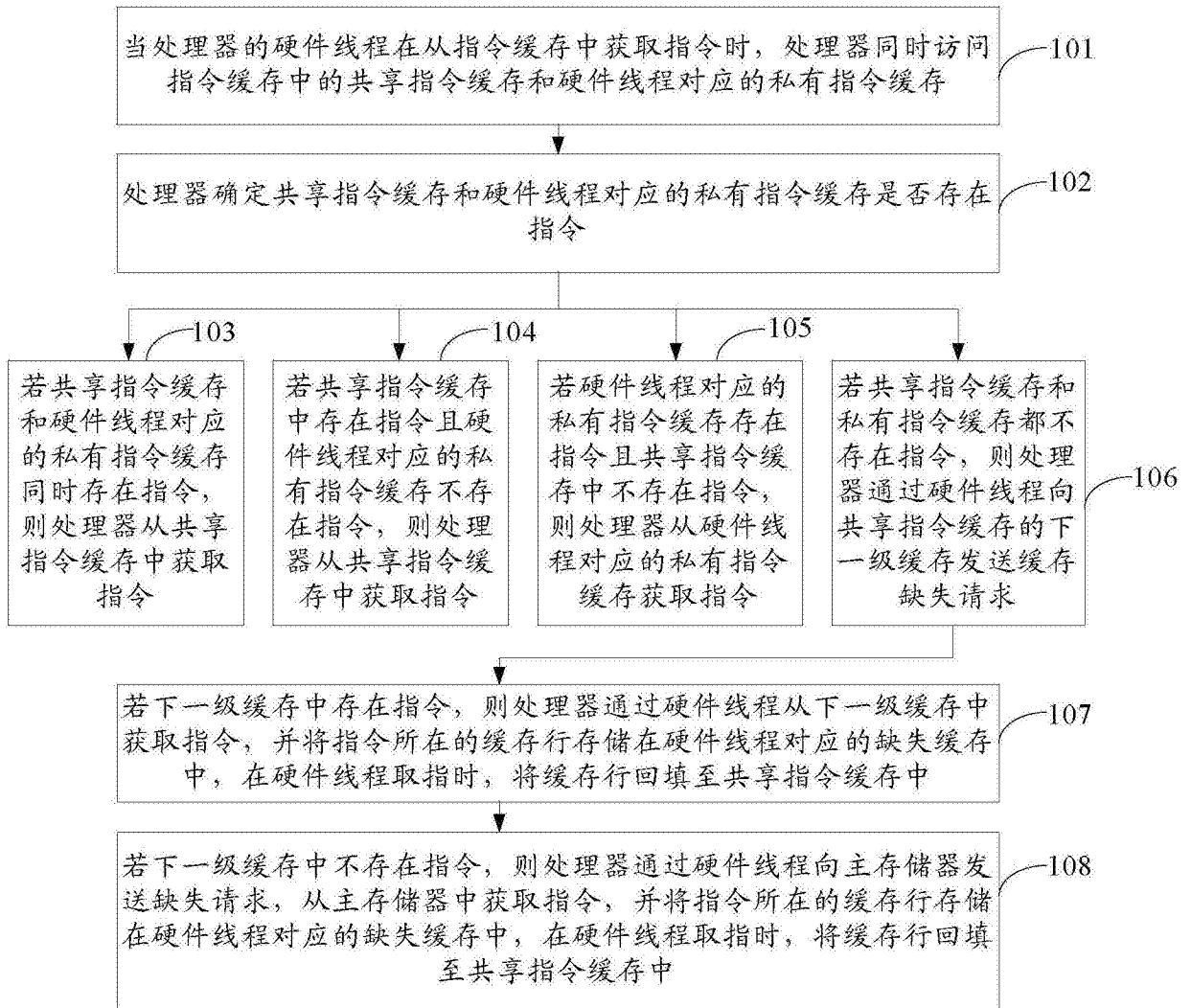


图2

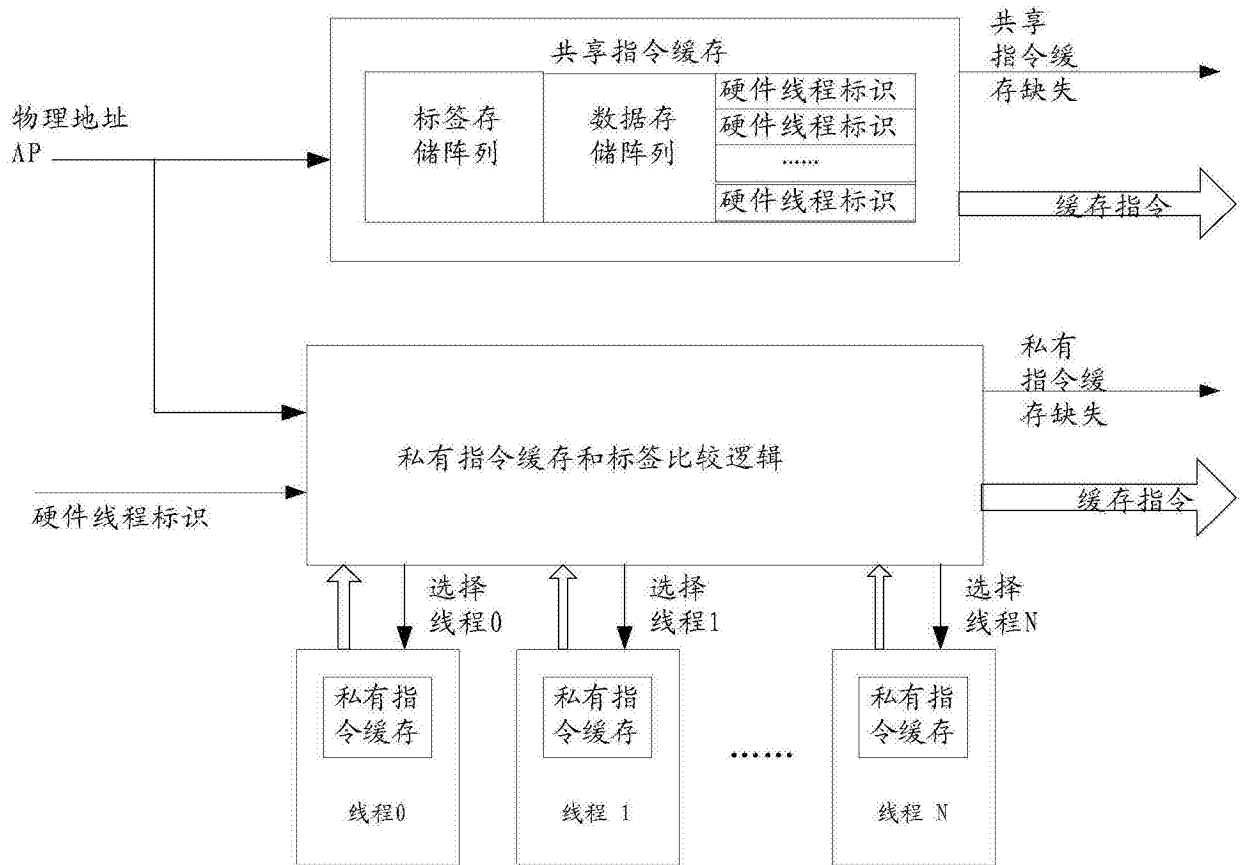


图3

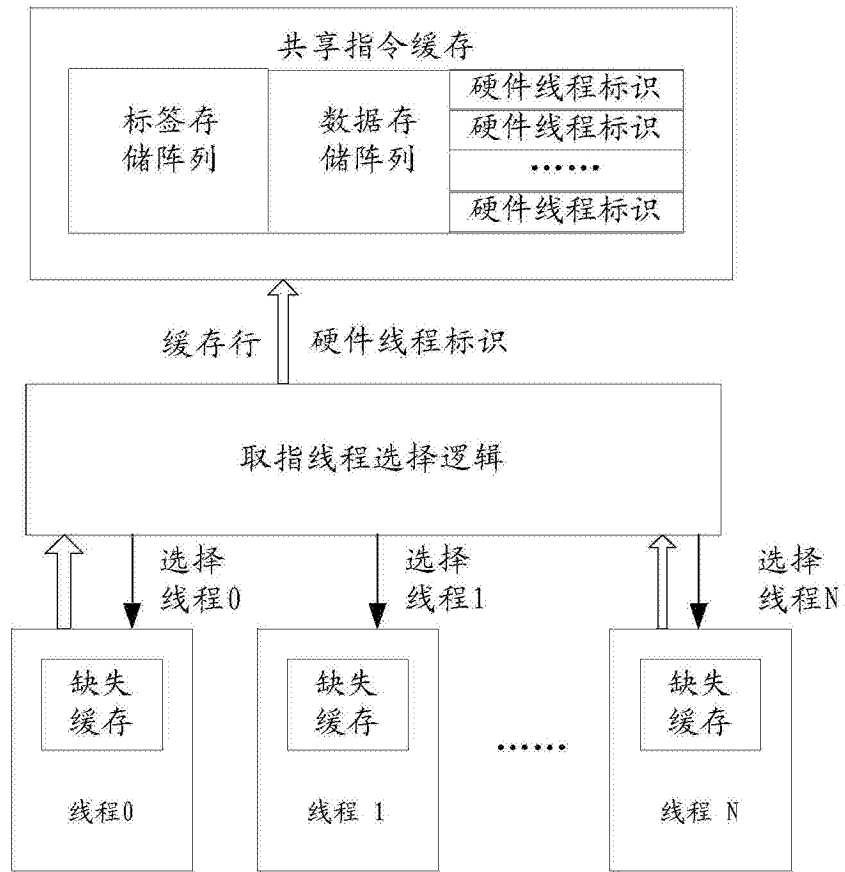


图4