



(12)发明专利

(10)授权公告号 CN 106484519 B

(45)授权公告日 2019.11.08

(21)申请号 201610889217.1

(22)申请日 2016.10.11

(65)同一申请的已公布的文献号  
申请公布号 CN 106484519 A

(43)申请公布日 2017.03.08

(73)专利权人 东南大学苏州研究院  
地址 215123 江苏省苏州市苏州工业园区  
独墅湖高教区林泉街399号

(72)发明人 齐志 孟炜 时龙兴

(74)专利代理机构 南京苏高专利商标事务所  
(普通合伙) 32204

代理人 柏尚春

(51)Int.Cl.

G06F 9/48(2006.01)

G06F 9/54(2006.01)

(56)对比文件

CN 105210040 A,2015.12.30,

CN 104346223 A,2015.02.11,

CN 102012803 A,2011.04.13,

CN 101923491 A,2010.12.22,

US 2015324207 A1,2015.11.12,

US 2015324204 A1,2015.11.12,

底涛.一种CPU+GPU的多核异构平台设计方案.《计算机与网络》.2016,第90页-93页.

审查员 唐文俊

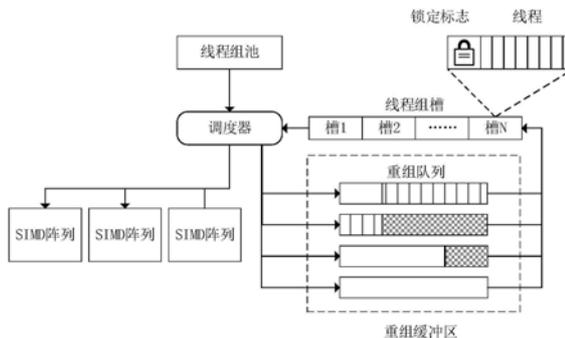
权利要求书2页 说明书5页 附图3页

(54)发明名称

异步线程重组方法及基于该方法的SIMT处理器

(57)摘要

本发明公开了一种异步线程重组方法及基于该方法的SIMT处理器,通过在不同线程组之间异步交换线程,消除线程组内的任务差异性,从而避免SIMD阵列中处理单元的闲置,提高GPU的性能。该方法在SIMT处理器中增加了两个功能单元:重组缓冲区、线程组槽。该方法借助重组缓冲区异步生成重组方案,使用线程组槽存储重组方案。与现有的线程重组方法相比,该异步方法不会带来线程组同步操作产生的SIMD流水线停顿,可使SIMT处理器取得更高性能。



1. 一种异步线程重组方法,用于SIMT处理器,其特征在于:所述SIMT处理器包含:  
线程组池:存储SIMT处理器当前执行的线程组、  
调度器:负责在每个时钟周期从线程组池中选取一个就绪的线程组并将其分配到一个空闲的SIMD阵列上执行一条指令、  
一个或多个SIMD阵列:负责执行线程组指令、  
线程组槽:用于存储线程组,以及与所述线程组一一相对应的线程组信息、  
以及重组缓冲区:用于生成线程重组方案;  
其中,所述线程重组方案为线程重组后每个线程组包含的线程;  
其中,所述重组缓冲区由若干重组队列组成;所述重组队列存储的信息包括重组队列中的线程对应的指令PC值、重组队列中的线程的任务属性、以及任务属性相同的线程;  
其中,所述线程组槽存储当前时刻线程组中包含的线程且线程组槽的锁定标志体现线程组的锁定状态;  
所述方法包括以下工作流程:
  - a) 通过所述调度器选取一个线程组槽未被锁定的线程组;
  - b) 判断所述线程组是否存在任务差异性,若所述线程组不存在任务差异性,则将直接其分配到SIMD阵列执行;反之,设置所述线程组对应的线程组槽的锁定标志为锁定状态,将所述线程组槽中的线程按任务属性分组,任务属性相同的线程加入重组缓冲区中形成重组队列,所述新加入的线程的指令PC值和任务属性与先前位于所述重组队列的线程的指令PC值和任务属性相一致;
    - c1) 依次检查在步骤b) 中被添加线程的重组队列,若当重组队列中的线程数满足一个或多个线程组所能容纳的线程数时,依次从重组队列中移除数量为线程组线程数的倍数的线程,打包成一个待写入线程组;
    - c2) 处理步骤c1) 中的待写入线程组,将其写入至已被锁定的线程组槽内,所述被移入线程的线程组槽对应的线程组的指令PC值与相应的被移除线程的重组队列的指令PC值相同,重复步骤b),并解锁所述线程组槽;
    - d) 重复步骤a) ~步骤c2) 直至SIMT处理器上的程序执行完成。
2. 根据权利要求1所述的一种异步线程重组方法,其特征在于:在步骤c2) 和步骤d) 之间还包括步骤c3):查找在重组缓冲区中被滞留的线程,若不存在,则执行步骤a);若存在,由滞留时间长至短依次从重组缓冲区中移出数量为线程组线程数的倍数的线程,打包成一个待写入线程组,执行步骤c2)。
3. 根据权利要求1所述的一种异步线程重组方法,其特征在于:处于解锁状态的线程组槽对应的线程组才会被分配到SIMD阵列执行。
4. 根据权利要求1所述的一种异步线程重组方法,其特征在于:在重组缓冲区中等待的过程中,线程组之间为异步状态,线程组被异步加入重组缓冲区或被异步移出重组缓冲区写入线程组槽。
5. 基于权利要求1所述的一种异步线程重组方法的SIMT处理器,所述SIMT处理器至少包含:  
线程组池:存储SIMT处理器当前执行的线程组、  
调度器:负责在每个时钟周期从线程组池中选取一个就绪的线程组并将其分配到一个

空闲的SIMD阵列上执行一条指令、

一个或多个SIMD阵列：负责执行线程组指令；

其特征在于：还包括

用于存储线程组以及与所述线程组一一相对应信息的线程组槽、

以及用于生成线程重组方案的重组缓冲区；所述线程重组方案为线程重组后每个线程组包含的线程。

6. 根据权利要求5所述的SIMT处理器，其特征在于：所述重组缓冲区由若干重组队列组成；所述重组队列存储的信息有位于重组队列中的线程对应的指令PC值、位于重组队列中的线程的任务属性和任务属性相同的线程。

7. 根据权利要求6所述的SIMT处理器，其特征在于：位于重组缓冲区的线程组之间为异步状态。

8. 根据权利要求5所述的SIMT处理器，其特征在于：所述线程组槽存储当前时刻线程组中包含的线程和存储相应线程组的锁定状态。

## 异步线程重组方法及基于该方法的SIMT处理器

### 技术领域

[0001] 本发明涉及到计算机架构领域,提供了一种异步线程重组方法及基于该方法的SIMT处理器。

### 背景技术

[0002] SIMT (Single Instruction Multiple Thread) 是并行计算中的一种常见执行模型。SIMT处理器中包含一个或多个SIMD (Single Instruction Multiple Data) 阵列。每个SIMD阵列由大量结构相同的处理单元组成。在同一时钟周期内, SIMD阵列内的所有处理单元处理不同的数据,但必须执行相同的指令。这是SIMD执行方式的限制。在SIMT处理器上执行的并行程序包含大量线程,这些线程被划分为若干线程组交替执行。在每个时钟周期, SIMT处理器中的调度器选择一个线程组,将其安排到空闲的SIMD阵列上执行。SIMD阵列中的每个处理单元负责执行该线程组中的一个线程。

[0003] 当同一线程组内的线程所处理的任务存在差异时, SIMD阵列内的处理单元无法全部同时工作,导致SIMT处理器的处理能力不能得到最大利用。任务差异性包括但不限于: 执行条件指令时判断结果的差异、执行访存指令时访存地址的差异、任务总执行时间的差异。以条件指令判断结果的差异为例,线程组内的线程根据判断结果的不同而执行不同的指令。由于SIMD执行方式的限制,判断结果为真的线程对应的处理单元和判断结果为假的线程对应的处理单元因执行不同的指令而无法同时工作。SIMD阵列需要先执行判断结果为真的线程,再执行判断结果为假的线程。在此过程中, SIMD阵列中总存在部分处理单元闲置,其处理能力不能得到最大利用。当线程组内的线程存在访存地址差异和任务总执行时间之间的差异时,也会出现类似的问题。

[0004] 解决上述问题的一种可行思路是线程重组。当SIMT处理器执行到存在任务差异性的指令时,线程重组在不同线程组之间交换线程,将不存在任务差异性的线程安排在同一线程组。这样,在执行这些线程组时SIMD阵列中的所有处理单元可以同时工作。然而,现有的线程重组技术在交换线程前要求线程组之间进行同步。仅当所有线程组均执行到待线程重组的指令后,线程重组才能开始交换线程。在SIMT处理器上,同步操作会带来SIMD阵列的额外停顿,存在不可忽视的性能开销。

### 发明内容

[0005] 发明目的: 针对上述现有技术存在的问题和不足,本发明的目的是提供一种异步线程重组方法及基于该方法的SIMT处理器,在不进行线程组同步的情况下,通过线程重组消除线程组内任务差异性,从而避免SIMD阵列中处理单元的闲置,使SIMT处理器的处理能力得到更充分利用。

[0006] 技术方案: 为实现上述发明目的,本发明采用的技术方案为一种异步线程重组方法,用于SIMT处理器,该SIMT处理器包含: 存储有当前SIMT处理器上执行的线程组的线程组池、负责在每个时钟周期从线程组池中选取一个就绪的线程组将其分配到一个空闲的SIMD

阵列上执行一条指令的调度器、一个或多个负责执行线程组指令的SIMD阵列、用于存储线程组以及与该线程组相对应信息的线程组槽和用于生成线程重组方案的重组缓冲区；

[0007] 其中,该线程重组方案为线程重组后每个线程组中包含的线程;

[0008] 其中,该重组缓冲区由若干重组队列组成;该重组队列存储的信息有位于重组队列中的线程对应的指令PC值、位于重组队列中的线程的任务属性和任务属性相同的线程;

[0009] 其中,该线程组槽包含存储当前时刻线程组中包含的线程和存储相应线程组的锁定状态;

[0010] 具体的,该方法包括以下工作流程:

[0011] a) 通过该调度器选取一个线程组槽未被锁定的线程组;

[0012] b) 判断该线程组是否存在任务差异性,若该线程组不存在任务差异性,则将直接其分配到SIMD阵列执行;反之,设置该线程组对应的线程组槽的锁定标志以锁定该线程组槽,将该线程组槽中的线程按任务属性分组,任务属性相同的线程加入重组缓冲区中形成重组队列,该新加入的线程的指令PC值和任务属性与先前位于该重组队列的线程的指令PC值和任务属性相一致;

[0013] c1) 依次检查在步骤b) 中被添加线程的重组队列,若当重组队列中的线程数满足一个或多个线程组所能容纳的线程数时,依次从重组队列中移除数量为线程组线程数的倍数的线程打包成一个待写入线程组;

[0014] c2) 处理步骤c1) 中的待写入线程组,将其写入至已被锁定的线程组槽内,该被移入线程的线程组槽对应的线程组的指令PC值与相应的被移除线程的重组队列的指令PC值相同,重复步骤b),并解锁该线程组槽;

[0015] d) 重复步骤a) ~步骤c) 直至SIMT处理器上的程序执行完成。

[0016] 进一步的,在步骤c2) 和步骤d) 之间还包括步骤c3): 查找在重组缓冲区中被滞留的线程,若不存在,则执行步骤a);若存在,由滞留时间长至短依次从重组缓冲区中移出数量为线程组线程数的倍数的线程打包成一个待写入线程组,执行步骤c2)。

[0017] 进一步的,处于解锁状态的线程组槽对应的线程组才会被分配到SIMD阵列执行。

[0018] 进一步的,在重组缓冲区中等待的过程中,线程组之间为异步状态,线程组被异步加入重组缓冲区或被异步移出重组缓冲区写入线程组槽。

[0019] 本发明还公开了一种基于异步线程重组方法的SIMT处理器,该SIMT处理器至少包含:存储有当前SIMT处理器上执行的线程组的线程组池、负责在每个时钟周期从线程组池中选取一个就绪的线程组将其分配到一个空闲的SIMD阵列上执行一条指令的调度器、一个或多个负责执行线程组的指令的SIMD阵列、用于存储线程组以及与该线程组相对应信息的线程组槽和用于生成线程重组方案的重组缓冲区;该线程重组方案为线程重组后每个线程组中包含的线程。

[0020] 进一步的,重组缓冲区由若干重组队列组成;该重组队列存储的信息有位于重组队列中的线程对应的指令PC值、位于重组队列中的线程的任务属性和任务属性相同的线程。

[0021] 进一步的,位于重组缓冲区的线程组之间为异步状态。

[0022] 进一步的,线程组槽包含存储当前时刻线程组中包含的线程和存储相应线程组的锁定状态。

[0023] 有益效果:本发明提出的SIMT处理器上的异步线程重组方法可以在不进行线程组同步的情况下,通过在不同线程组之间异步交换线程来消除线程组内任务差异性,从而避免SIMD阵列中处理单元的闲置,使SIMT处理器的处理能力得到更充分利用。与现有的线程重组方法相比,该异步方法不会带来线程组同步操作产生的SIMD流水线停顿,不存在额外的性能开销,可使SIMT处理器取得更高性能。

### 附图说明

[0024] 附图用来提供对本发明的进一步理解,并且构成说明书的一部分,与本发明的实施例一起用于解释本发明,并不构成对本发明的限制。在附图中:

[0025] 图1为本发明的实施例中所假设的基准SIMT处理器的结构框图;

[0026] 图2为基于本发明的异步线程重组方法的SIMT处理器的结构框图;

[0027] 图3为本发明的异步线程重组方法的工作流程图;

[0028] 图4以访存地址差异性为例,为本发明的异步线程重组方法在差异性时的工作原理演示图。

### 具体实施方式

[0029] 下面结合附图和具体实施例,进一步阐明本发明,应理解这些实施例仅用于说明本发明而不适用于限制本发明的范围,在阅读了本发明之后,本领域技术人员对本发明的各种等价形式的修改均落于本申请所附权利要求所限定的范围。

[0030] 图1为本发明的实施例中所假设的基准SIMT处理器的结构框图。SIMT处理器至少包含:线程组池、调度器、一个或多个SIMD阵列。线程组池中存储有当前SIMT处理器上执行的线程组。调度器负责在每个时钟周期从线程组池中选取一个就绪的线程组,将其分配到一个空闲的SIMD阵列上执行一条指令。SIMD阵列负责执行线程组的指令。

[0031] 图2为基于本发明的异步线程重组方法的SIMT处理器的结构框图。本发明的异步线程重组方法在SIMT处理器中新增了线程组槽和重组缓冲区。

[0032] 本实施例中假设SIMT处理器上共有N个线程组并行执行,相应地,SIMT处理器中新增N个线程组槽,该线程组槽包括头部和后续部分,该线程组槽头部为相应线程组的锁定标志,记录该线程组是否被锁定;该线程组槽的后续部分存储线程重组方案,即线程重组实施后每个新线程组中包含的线程,每个线程用一个线程编号标识。

[0033] 重组缓冲区由一系列重组队列组成,每个重组队列存储如下信息:该重组队列中的线程对应的指令PC值、该重组队列中的线程的任务属性、任务属性不存在差异性的若干线程。

[0034] 图3为本发明的异步线程重组方法的详细工作流程图。此实施例中假设每个线程组中的线程数为W。

[0035] 第1步,从线程组池中选取一个线程组,这一工作由SIMT处理器中的调度器完成。

[0036] 第2步,读取该线程组对应的线程组槽的锁定标志,以检查该线程组槽是否被锁定。若被锁定,则转至第1步。

[0037] 第3步,检查该线程组将执行的指令是否可能带来任务差异性;若否,转第11步。

[0038] 第4步,设置该线程组对应的线程组槽中的锁定标志,以锁定该线程组槽。

[0039] 第5步,将该线程组对应的线程组槽中的线程按任务属性分组,将任务属性相同的线程加入到重组缓冲区中形成指令PC值、任务属性均相同的重组队列。该步骤中,可能有一个或多个重组队列被添加线程。

[0040] 第6步,依次检查在第5步中被添加线程的重组队列,找出累积线程数目不小于W的重组队列;如果不存在这样的重组队列,转第9步。

[0041] 第7步,依次处理第6步中找到的重组队列,将重组队列中的前W个线程移出打包成一个待写入线程组。

[0042] 第8步,处理所有的待写入线程组,对于每个待写入线程组,将其写入一个满足以下两项条件的线程组槽内:(1)该线程组槽已被锁定,(2)该线程组槽对应的线程组即将执行的指令PC值与待写入线程组中的线程对应的PC值相同;将上述线程组槽内的线程分配到SIMD阵列执行,并清除该线程组槽中的锁定标志,以解锁该线程组槽,接着,转至第1步。

[0043] 第9步,查找在重组缓冲区中停留时间过长的线程;如果不存在这样的线程,转至第1步。

[0044] 第10步,将第9步找出的线程中停留时间最长的W个线程移出重组队列并打包成一个待写入线程组,接着转至第8步。

[0045] 第11步,直接将调度器在第1步选取的线程组对应的线程组槽中的线程分配到SIMD阵列执行,接着转至第1步。

[0046] 图4为本发明的异步线程重组方法在消除访存地址差异性时的工作原理演示图。此实施例中假定所执行的程序包含一条可能导致任务差异性的Load指令,PC值为32;还包含两条不可能导致任务差异性的Add指令,PC值分别为64、96。此实施例中还假定SIMT处理器上共有3个线程组,每个线程组中所能容纳的线程数为4。

[0047] 图4中四幅子图分别对应t<sub>0</sub>至t<sub>3</sub>四个时刻,每幅子图的上半部分为线程组槽的状态,从左向右分别对应第一线程组1、第二线程组2和第三线程组3;下半部分为重组队列的状态,该实施例中线程重组用于消除访存地址的差异性,因此重组队列存储的任务属性即访存地址,线程组中的线程亦按照访存地址分组进入重组队列。

[0048] 时刻t<sub>0</sub>,第三线程组3被分配到SIMD阵列上执行PC值为64的第一条Add指令,而另两个线程组尚未开始执行。

[0049] 时刻t<sub>1</sub>,第一线程组1的PC值为32的Load指令被发射,由于Load指令可能带来任务差异性,线程组1所在的线程组槽被锁定。随后,第一线程组1的线程按照访存地址分组进入重组缓冲区中PC值匹配且对应访存地址0x00和0x80的重组队列中。

[0050] 时刻t<sub>2</sub>,第二线程组2中的PC值为32的Load指令被发射,对应的线程组槽同样被锁定,该线程组槽内的线程按照访存地址分组进入重组缓冲区中PC值匹配且对应访存地址0x80和0x100的重组队列,至此,重组队列0x80已凑满一个线程组。

[0051] 因此,在时刻t<sub>3</sub>,上述已凑满一个线程组的重组队列被清空,其中的线程5,6,3,4被写入第一个线程组槽,并分配到SIMD阵列执行,该线程组槽被解锁。

[0052] 上述过程中以下两点需要注意:第一,由于线程组槽的锁定机制,线程3,4从时刻t<sub>1</sub>进入重组缓冲区到时刻t<sub>3</sub>离开缓冲区,PC值一直保持不变,相比之下,执行PC值为96的第2条Add指令而未被锁定的第三线程组槽3中线程9~12的PC值在此期间由64变化为96。第二,线程5,6和3,4原先分别来自两个执行步调不同的线程组,但线程组槽的锁定机制保证

这四个任务属性相同的线程在被写入到第一线程组槽1时PC值相同。

[0053] 最后应说明的是:以上该仅为本发明的优选实施例而已,并不用于限制本发明,尽管参照前述实施例对本发明进行了详细的说明,对于本领域的技术人员来说,其依然可以对前述各实施例所记载的技术方案进行修改,或者对其中部分技术特征进行等同替换。凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

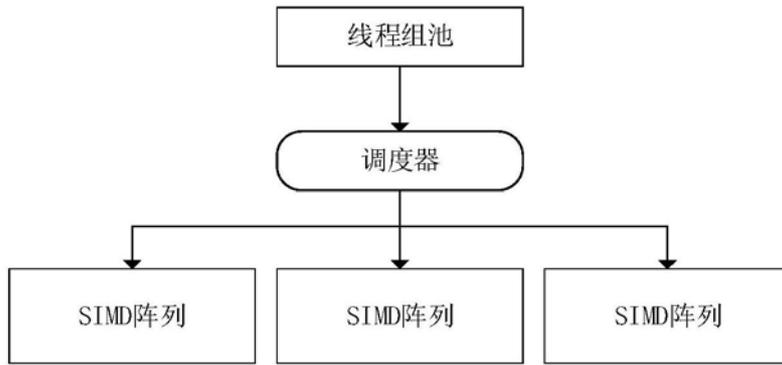


图1

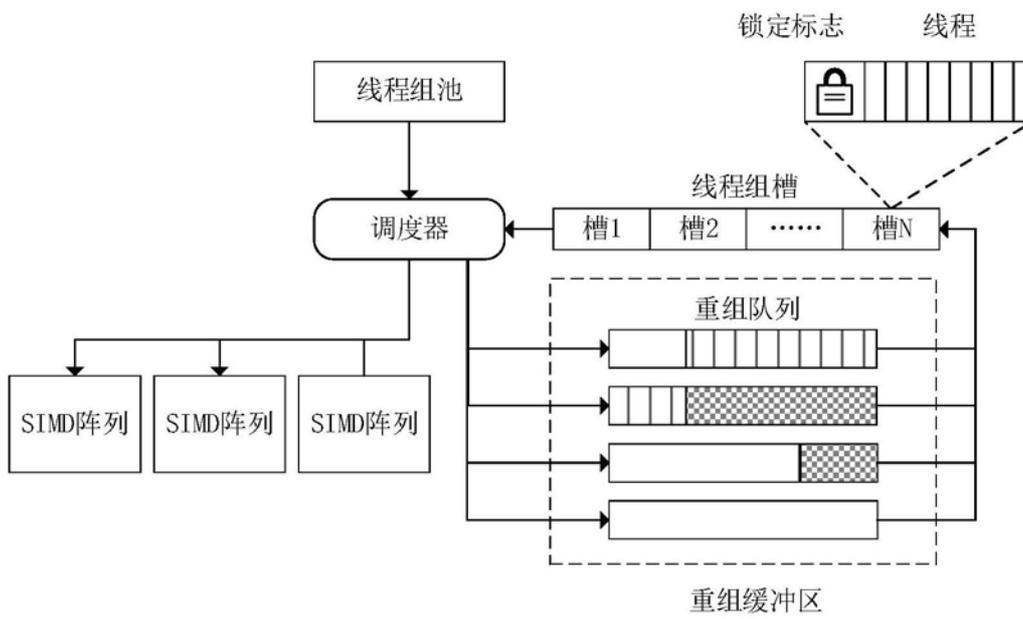


图2

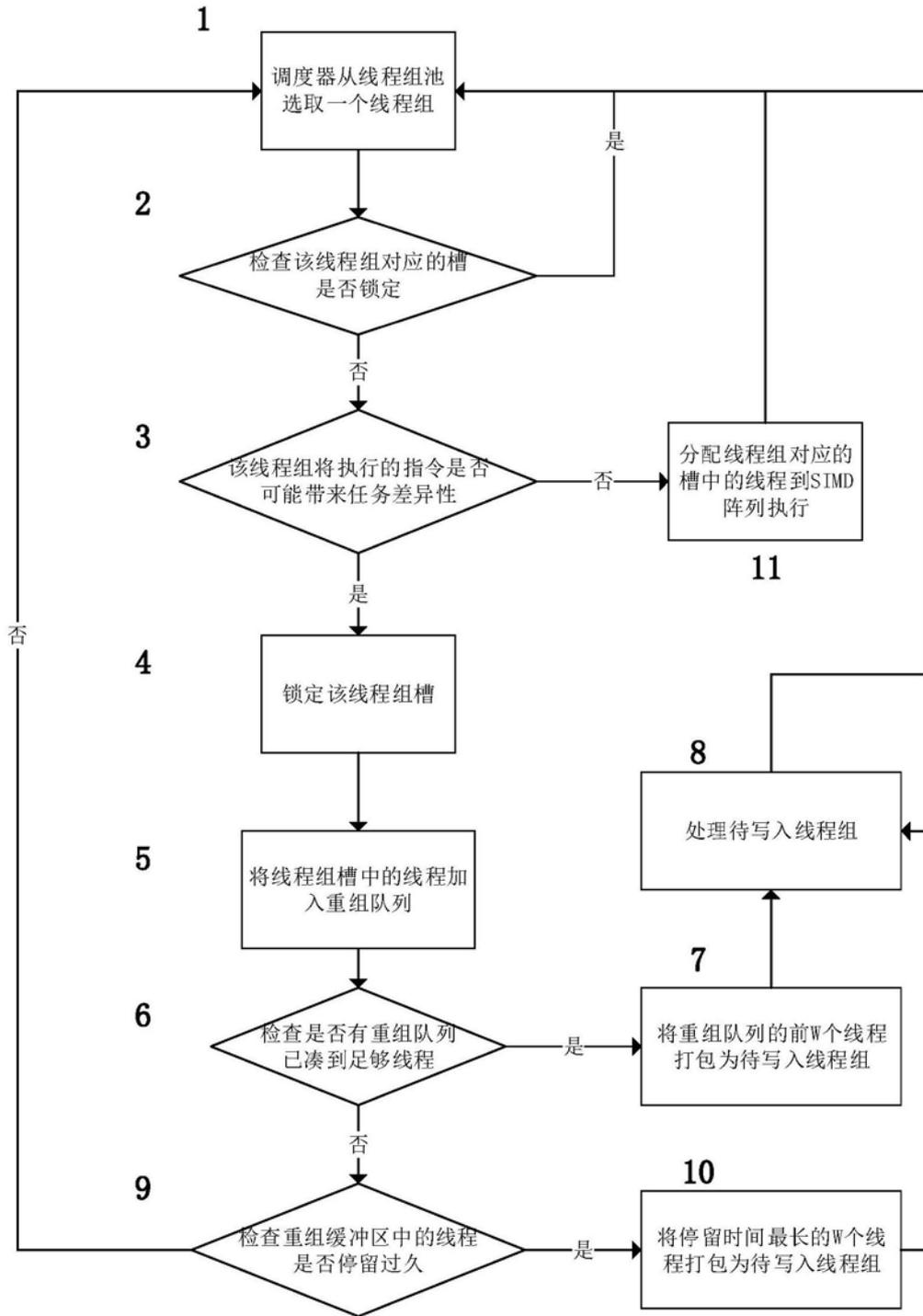


图3

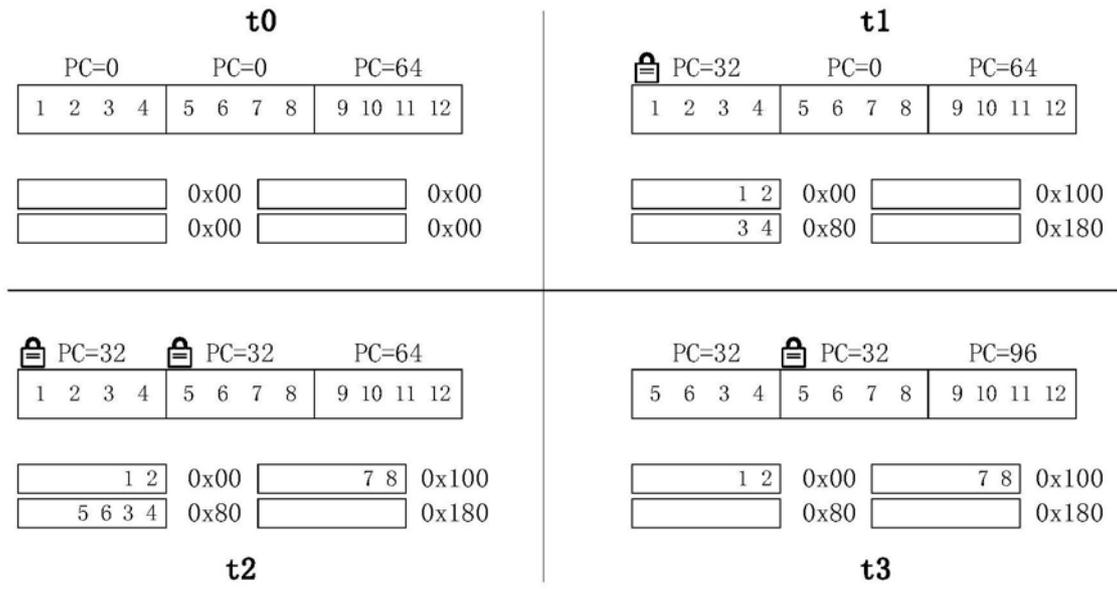


图4