



(12)发明专利申请

(10)申请公布号 CN 109614146 A

(43)申请公布日 2019.04.12

(21)申请号 201811355072.2

(22)申请日 2018.11.14

(71)申请人 西安翔腾微电子科技有限公司
地址 710065 陕西省西安市高新一路25号
创新大厦S303室

(72)发明人 齐晶晶 史嘉涛 范飞虎 王维
张淑 王菁

(74)专利代理机构 西安嘉思特知识产权代理事
务所(普通合伙) 61230
代理人 张捷

(51)Int.Cl.
G06F 9/30(2006.01)
G06F 9/38(2006.01)

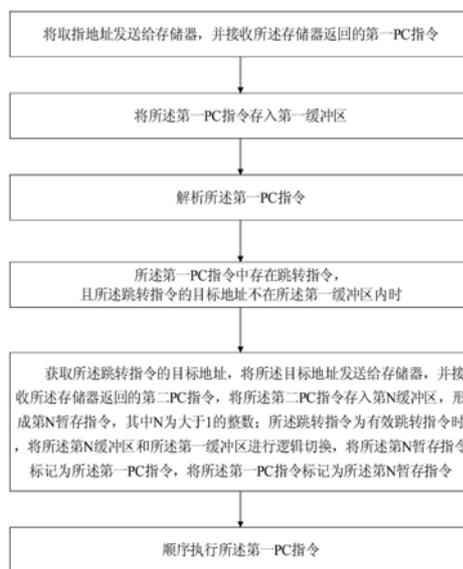
权利要求书2页 说明书7页 附图2页

(54)发明名称

一种局部跳转指令取指方法及装置

(57)摘要

本发明涉及一种局部跳转指令取指方法及装置,所述方法对从存储器取回的指令进行判断,若遇到跳转指令,则执行跳转动作,若没有发现跳转指令则正常执行指令内容。如果跳转不成功,则能够以零延迟继续执行指令,且跳转指令的解析过程能够与取回跳转的目标地址处指令的过程重叠执行,完全隐藏跳转指令解析过程中的延迟。本发明专利还提供一种局部跳转指令取指装置。



1. 一种局部跳转指令取指方法,其特征在于,应用于多个缓冲区,包括:

步骤1:将取指地址发送给存储器,并接收所述存储器返回的第一PC指令;

步骤2:将所述第一PC指令存入第一缓冲区;

步骤3:解析所述第一PC指令;

步骤4:判断所述第一PC指令中是否存在跳转指令;

若是,且判断所述跳转指令的目标地址不在所述第一缓冲区内时,执行步骤5;

若否,则执行步骤6;

步骤5:获取所述跳转指令的目标地址,将所述目标地址发送给存储器,并接收所述存储器返回的第二PC指令,将所述第二PC指令存入第N缓冲区,形成第N暂存指令,其中N为大于1的整数;

所述跳转指令为有效跳转指令时,将所述第N缓冲区和所述第一缓冲区进行逻辑切换,将所述第N暂存指令标记为所述第一PC指令,将所述第一PC指令标记为所述第N暂存指令,执行步骤4;

步骤6:顺序执行所述第一PC指令。

2. 根据权利要求1所述的取指方法,其特征在于,所述步骤4中,

若是,且判断所述跳转指令的目标地址在所述第一缓冲区内时,执行步骤6。

3. 根据权利要求1或2所述的取指方法,其特征在于,所述步骤5中,

所述跳转指令为无效跳转指令时,执行步骤6。

4. 根据权利要求1所述的取指方法,其特征在于,所述步骤6之后还包括:

步骤8:接收执行结束信号,根据所述执行结束信号生成所述取指地址,执行步骤1。

5. 根据权利要求4所述的取指方法,其特征在于,根据所述执行结束信号生成所述取指地址,具体为:

将所述取指地址顺序加1后标记为所述取指地址。

6. 根据权利要求1所述的取指方法,其特征在于,所述缓冲区包括缓冲0区和缓冲1区,所述缓冲0区和所述缓冲1区以乒乓方式存储所述第一PC指令或者所述第二PC指令。

7. 根据权利要求1所述的取指方法,其特征在于,所述步骤3之后,所述步骤4之前还包括:

步骤Y:判断所述第一PC指令种是否存在入栈指令或者出栈指令;

若是,则执行所述入栈指令或者出栈指令;

若否,则执行步骤4。

8. 根据权利要求1所述的取指方法,其特征在于,步骤1之前还包括:

获取初始地址,将初始地址标记为所述取指地址。

9. 根据权利要求1所述的取指方法,其特征在于,所述多个缓冲区的位宽为256位宽。

10. 一种局部跳转指令取指装置,其特征在于,包括存储器、处理器和寄存器,所述寄存器内存储有计算机程序,

所述处理器执行所述计算机程序实现如下步骤:将取指地址发送给存储器,并接收所述存储器返回的第一PC指令;将所述第一PC指令存入第一缓冲区;解析所述第一PC指令;当所述第一PC指令中存在跳转指令,且所述跳转指令的目标地址不在所述第一缓冲区内时,获取所述跳转指令的目标地址,将所述目标地址标记发送给存储器,并接收所述存储器返

回的第二PC指令,将所述第二PC指令存入第N缓冲区,形成第N暂存指令,其中N为大于1的整数;所述跳转指令为有效跳转指令时,将所述第N缓冲区和所述第一缓冲区进行逻辑切换,将所述第N暂存指令标记为所述第一PC指令,将所述第一PC指令标记为所述第N暂存指令,并循环上述指令;

所述处理器还用于执行,当所述第一PC指令中不存在跳转指令时,顺序执行所述第一PC指令。

一种局部跳转指令取指方法及装置

技术领域

[0001] 本发明属于处理器技术领域,具体涉及一种局部跳转指令取指方法及装置。

背景技术

[0002] 微处理器与传统的中央处理器相比,具有体积小、重量轻和容易模块化等优点。目前,微处理器已经无处不在,无论是录像机、智能洗衣机、移动电话等家电产品,还是汽车引擎控制,以及数控机床、导弹精确制导等都要嵌入各类不同的微处理器。微处理器不仅是微型计算机的核心部件,也是各种数字化智能设备的关键部件。国际上的超高速巨型计算机、大型计算机等高端计算系统也都采用大量的通用高性能微处理器建造。

[0003] 目前,现代微处理器通过在一个时钟周期里同时发射和执行多条指令,从而可以获得更高的性能。然后,由于跳转和分支指令的存在,且未集成分支预测机制,从处理跳转和分支指令开始到取回并继续执行后续有效指令之间,处理器一直处于等待状态,使得在流水线的流水线中插入了较多的无效操作,这种简单化和直接化的处理方式导致分支指令执行的效率不高,也降低了处理器流水线的资源利用率。

发明内容

[0004] 为了解决现有技术中存在的上述问题,本发明提供一种局部跳转指令取指方法及装置。本发明要解决的技术问题通过以下技术方案实现:

[0005] 本发明实施例提供一种局部跳转指令取指方法,应用于多个缓冲区,包括:

[0006] 步骤1:将取指地址发送给存储器,并接收所述存储器返回的第一PC指令;

[0007] 步骤2:将所述第一PC指令存入第一缓冲区;

[0008] 步骤3:解析所述第一PC指令;

[0009] 步骤4:判断所述第一PC指令中是否存在跳转指令,;

[0010] 若是,且判断所述跳转指令的目标地址不在所述第一缓冲区内时,执行步骤5;

[0011] 若否,则执行步骤6;

[0012] 步骤5:获取所述跳转指令的目标地址,将所述目标地址发送给存储器,并接收所述存储器返回的第二PC指令,将所述第二PC指令存入第N缓冲区,形成第N暂存指令,其中N为大于1的整数;

[0013] 所述跳转指令为有效跳转指令时,将所述第N缓冲区和所述第一缓冲区进行逻辑切换,将所述第N暂存指令标记为所述第一PC指令,将所述第一PC指令标记为所述第N暂存指令,执行步骤4;

[0014] 步骤6:顺序执行所述第一PC指令。

[0015] 在一个具体的实施例中,所述步骤4中,

[0016] 若是,且判断所述跳转指令的目标地址在所述第一缓冲区内时,执行步骤6。

[0017] 在一个具体的实施例中,所述步骤5中,

[0018] 所述跳转指令为无效跳转指令时,执行步骤6。

- [0019] 在一个具体的实施例中,所述步骤6之后还包括:
- [0020] 步骤8:接收执行结束信号,根据所述执行结束信号生成所述取指地址,执行步骤1。
- [0021] 在一个具体的实施例中,根据所述执行结束信号生成所述取指地址,具体为:
- [0022] 将所述取指地址顺序加1后标记为所述取指地址。
- [0023] 在一个具体的实施例中,所述缓冲区包括缓冲0区和缓冲1区,所述缓冲0区和所述缓冲1区以乒乓方式存储所述第一PC指令或者所述第二PC指令。
- [0024] 在一个具体的实施例中,所述步骤3之后,所述步骤4之前还包括:
- [0025] 步骤Y:判断所述第一PC指令中是否存在入栈指令或者出栈指令;
- [0026] 若是,则执行所述入栈指令或者出栈指令;
- [0027] 若否,则执行步骤4。
- [0028] 在一个具体的实施例中,步骤1之前还包括:
- [0029] 获取初始地址,将初始地址标记为所述取指地址。
- [0030] 在一个具体的实施例中,所述多个缓冲区的位宽为256位宽。
- [0031] 本发明另一实施例还提供一种局部跳转指令取指装置,包括存储器、处理器和寄存器,所述寄存器内存储有计算机程序,
- [0032] 所述处理器执行所述计算机程序实现如下步骤:将取指地址发送给存储器,并接收所述存储器返回的第一PC指令;将所述第一PC指令存入第一缓冲区;解析所述第一PC指令;当所述第一PC指令中存在跳转指令,且所述跳转指令的目标地址不在所述第一缓冲区内时,获取所述跳转指令的目标地址,将所述目标地址标记发送给存储器,并接收所述存储器返回的第二PC指令,将所述第二PC指令存入第N缓冲区,形成第N暂存指令,其中N为大于1的整数;所述跳转指令为有效跳转指令时,将所述第N缓冲区和所述第一缓冲区进行逻辑切换,将所述第N暂存指令标记为所述第一PC指令,将所述第一PC指令标记为所述第N暂存指令,并循环上述指令;
- [0033] 所述处理器还用于执行,当所述第一PC指令中不存在跳转指令时,顺序执行所述第一PC指令。
- [0034] 与现有技术相比,本发明的有益效果:
- [0035] 1、本发明实施例通过设置多个缓冲区,缓冲区用于存储指令,这样指令取回、指令解析以及指令执行工作可以重叠执行。
- [0036] 2、当第一缓冲区中执行的指令发生跳转时,无论指令是否发生相关,直接将跳转的目标地址所在指令取入第N缓冲区中,第一缓冲区中仍然保留着顺序执行的指令流,如果跳转不成功,或者跳转的目标地址在第一缓冲区之后的范围内,则能够以零延迟继续执行指令,且跳转指令的解析过程能够与取回跳转的目标地址处指令的过程重叠执行,完全隐藏跳转指令解析过程中的延迟。
- [0037] 3、第一缓冲区能够存储当前正在执行的8条指令和向后8条指令的局部范围,在发现跳转指令,并计算得到跳转的目标地址后,比较该目标地址是否在第一缓冲区缓存的16条的指令中。如果在,则无需额外取指操作,直接从第一缓冲区跳转的目标地址处开始执行。
- [0038] 4、第N缓冲区和第一缓冲区具有相同的结构,能够按需要切换第N缓冲区和第一缓

冲区的逻辑关系,其中,第N缓冲区和第一缓冲区以乒乓方式接收取回的指令,且两个缓冲区内部分别又分为0区和1区,0区和1区以乒乓方式接收取回的指令数据,指令输出接口也是以乒乓方式对其中的指令进行译码,而不以流水方式在两个缓冲区间传递指令。

附图说明

[0039] 图1为本发明提供的基于双路径的高效局部跳转指令取指电路的逻辑图;

[0040] 图2为本发明提供的基于双路径的高效局部跳转指令取指电路的流程示意图。

具体实施方式

[0041] 在本发明实施例仅是为了便于对本发明的技术方案进行解释说明,本技术方案并不限于本发明实施例所提供的内容,因此不能理解为对本发明的限制。

[0042] 实施例一

[0043] 如图1和图2所示,本发明实施例提供一种基于双路径的高效局部跳转指令取指方法,其中,优选的多个缓冲区为两个缓冲区,分别为第一缓冲区和第二缓冲区,具体实施过程如下:

[0044] 步骤1:将取指地址发送给存储器,并接收所述存储器返回的第一PC指令;

[0045] 步骤2:将所述第一PC指令存入第一缓冲区;

[0046] 步骤3:解析所述第一PC指令;

[0047] 步骤4:判断所述第一PC指令中是否存在跳转指令,;

[0048] 若是,且判断所述跳转指令的目标地址不在所述第一缓冲区内时,执行步骤5;

[0049] 若否,则执行步骤6;

[0050] 步骤5:获取所述跳转指令的目标地址,将所述目标地址发送给存储器,并接收所述存储器返回的第二PC指令,将所述第二PC指令存入第N缓冲区,形成第N暂存指令,其中N为大于1的整数;

[0051] 所述跳转指令为有效跳转指令时,将所述第N缓冲区和所述第一缓冲区进行逻辑切换,将所述第N暂存指令标记为所述第一PC指令,将所述第一PC指令标记为所述第N暂存指令,执行步骤4;

[0052] 步骤6:顺序执行所述第一PC指令。

[0053] 具体的,处理器负责更新取指地址并从外部接口取回指令,取指地址有三种类型,第一种为初始地址,初始地址也为触发信号,第二种为取指地址顺序加1形成的新的取指地址;第三种是跳转指令中的目标地址标记为取指地址。首先获取初始地址,将初始地址标记为取指地址,根据取指地址生成取指请求,并将取指请求发送给存储器,取指请求可以认为就是取指地址。

[0054] 存储器接收到取指请求,并从取指请求中解析出取指地址,根据取指地址向处理器返回多条PC指令。

[0055] 需要说明的是,现有技术中,每一次取指只能取一条指令,而本申请中每一次取指能够取回多条PC指令,优选的,每一次取指能够取回8条PC指令。这样能够提高取指效率,节约消耗。

[0056] 处理器接收返回的第一PC指令后,将其存入第一缓冲区,第一缓冲区为256位宽,

能够一次性存入8条PC指令,即第一PC指令,包括8条PC指令。

[0057] 然后处理器对存入第一缓冲区的8条PC指令进行解析,具体的,每两条PC指令一起解析,获取每一条PC指令的类型,并判断两条PC指令中是否存在跳转指令,如果没有跳转指令,则将该两条PC指令发送给执行模块去执行指令内容即执行步骤6。如果存在跳转指令,且如果两条PC指令中的第一条是跳转指令,那么执行步骤5,两条PC指令中的第二条指令暂不执行;如果两条PC指令中的第二条为跳转指令,那么将两条PC指令中的第一条发送给执行模块执行步骤6,而对第二条PC指令执行步骤5。

[0058] 进一步的,当某一PC指令为跳转指令时,还需要进行再次判断。

[0059] 即判断所述跳转指令的目标地址是否在所述第一缓冲区内;

[0060] 如果在第一缓冲区中,则无需额外取指操作,直接从第一缓冲区的目标地址处开始执行。

[0061] 如果不在第一缓冲区中,则执行步骤5的内容。

[0062] 具体的,步骤5的内容为:获取所述跳转指令的目标地址,将所述目标地址标记发送给存储器,并接收所述存储器返回的第二PC指令,将所述第二PC指令存入第二缓冲区,形成第二暂存指令,其中N等于二;所述跳转指令为有效跳转指令时,将所述第二缓冲区和所述第一缓冲区进行逻辑切换,将所述第二暂存指令标记为所述第一PC指令,将所述第一PC指令标记为所述第二暂存指令,执行步骤4

[0063] 其中第二缓冲区是和第一缓冲区结构完全相同的另一缓冲区,将跳转指令取回的PC指令标记为第二PC指令,存入第二缓冲区,形成第二暂存指令,而第一缓冲区中剩余的其他PC指令仍然暂存在第一缓冲区中,不被覆盖。

[0064] 处理器在根据跳转指令的目标地址重新取指的过程中,还需要执行一个判断步骤,也即判断所述跳转指令是否为有效跳转指令;

[0065] 如果是有效的跳转指令,那么第二缓冲区和第一缓冲区的逻辑关系进行互换,将第二缓冲区标记为新的第一缓冲区,将原第一缓冲区标记为第二缓冲区,然后执行新的第一缓冲区中的第一PC指令。而第二缓冲区中剩余的PC指令为第二暂存指令,暂不执行。

[0066] 需要说明的是,执行新的第一缓冲区的第一PC指令,具体的包括判断第一PC指令是否为跳转指令,即步骤4的内容。

[0067] 如果不是有效跳转指令,那么对于处理器重新取指后取回并放入第二缓冲区内的第二暂存指令不做处理,第一缓冲区仍然是第一缓冲区,执行模块依然执行第一缓冲区内剩余的第一PC指令。

[0068] 上述内容中,所有的判断和步骤选择最终都会落脚到执行模块执行PC指令的内容。当执行模块将第一缓冲区内的指令读取并执行完后,会向处理器发送执行结束信号,表示第一缓冲区内已经没有可执行的PC指令,这样处理器将之前的取指地址顺序加1生成新的取指地址,然后根据新的取指地址进行取指操作,并将取回的多条PC指令存入第一缓冲区。

[0069] 需要说明的是,在上述步骤中,判断跳转指令是否为有效跳转指令具体为:

[0070] 每一跳转指令携带有跳转条件和跳转参数,例如跳转条件为大于等于0,那么判断跳转参数是否大于等于0,如果跳转参数大于或者等于0,那么跳转条件成立,即该跳转指令为有效跳转指令,如果跳转参数小于0,说明跳转条件不成立,那么该跳转指令为无效跳转

指令。

[0071] 综上所述,本发明专利中,取回的指令存入第一缓冲区中,遇到跳转指令,则取指并存入第二缓冲区中,并将第二缓冲区和第一缓冲区进行逻辑互换,执行模块总是执行第一缓冲区中的指令。且只有在遇到跳转指令时,将指令存入第二缓冲区中,若解析并执行第一PC指令时,没有发现跳转指令,那么再次取回的指令依然存入第一缓冲区中。

[0072] 当第一缓冲区中执行的指令发生跳转时,无论指令是否发生相关,直接将跳转的目标地址所在指令取入第二缓冲区中,第一缓冲区中仍然保留着顺序执行的指令流,如果跳转不成功,或者跳转的目标地址在第一缓冲区之后的范围内,则能够以零延迟继续执行指令,且跳转指令的解析过程能够与取回跳转的目标地址处指令的过程重叠执行,完全隐藏跳转指令解析过程中的延迟。

[0073] 实施例二

[0074] 在实施例一的基础上,本实施例对多个缓冲区进行详细说明。

[0075] 本实施例中,如图2所示,第一缓冲区和第N缓冲区的结构是完全相同的,N为大于1的整数。

[0076] 具体的,缓冲区包括缓冲0区和缓冲1区,所述缓冲0区和所述缓冲1区以乒乓方式存储所述PC指令。

[0077] 步骤3中,将所述第一PC指令存入第一缓冲区中,举例而言,是将存储器接收到的第一PC指令存入缓冲0区,然后处理器进行再次取指,将取指地址顺序加1生成新的取指地址,将取指地址发送给存储器,并接收存储器返回的新的第一PC指令,将返回的新的第一PC指令存入第一缓冲区的缓冲1区。

[0078] 缓冲0区和缓冲1区并没有严格的先后顺序,若步骤3中,将多条PC指令存入缓冲1区,那么再次取指后的多条PC指令则存入缓冲0区。

[0079] 需要说明的是,处理器的再次取指动作是自发的。

[0080] 本实施例中,处理器每一次取指动作,取回8条指令放入缓冲0区,然后再次取指,取回8条指令放入缓冲1区,每一个缓冲区可以同时存入16条PC指令。

[0081] 多个缓冲区具有相同的结构,以乒乓方式接收取回的指令,且缓冲区内部各又分为0区和1区,0区和1区以乒乓方式接收取回的指令数据,指令输出接口也是以乒乓方式对其中的指令进行译码,而不以流水方式在两个缓冲区间传递指令。

[0082] 实施例三

[0083] 在上述实施例一和实施例二的基础上,本实施例提供一种多路径的高效局部跳转指令取指方法,其中,本实施例中,多个缓冲区至少为三个缓冲区,分别为第一缓冲区、第二缓冲区和第三缓冲区。

[0084] 本实施例中的实施过程与实施例一相同,不同之处在于:

[0085] 当发现跳转指令时,将根据跳转的目标地址取回的多条PC指令存入第二缓冲区中,并将第二缓冲区标记为第一缓冲区,原第一缓冲区标记为第二缓冲区,新的第二缓冲区内剩余的指令暂存。

[0086] 解析新的第一缓冲区内多条PC指令,判断每一条PC指令是否为跳转指令,如果是跳转指令,并且跳转指令的目标地址不在当前指令缓冲区中,那么执行跳转命令,即获取跳转指令的目标地址,将目标地址标记为取指地址,将取指地址发送给存储器,并接收存储

器返回的PC指令,将PC指令存入第三缓冲区中,

[0087] 然后判断跳转指令是否为有效跳转指令,如果是有效跳转指令,那么将第三缓冲区标记为第一缓冲区,第一缓冲区标记为第三缓冲区,执行新的第一缓冲区内的PC指令。原先的第一缓冲区内剩余指令暂存。

[0088] 需要说明的是,本实施例中的第一、第二并不具有先后顺序,仅是起到区分作用。

[0089] 进一步的,当需要更新取指地址进行取指时,判断新的取指地址是否在第二缓冲区或者第三缓冲区暂存的那些指令中,如果在,则无需取指操作,将目标地址所在的第N缓冲区和第一缓冲区进行逻辑互换,然后执行第一缓冲区中目标地址所在的指令。

[0090] 本实施例中,通过多个缓冲区,能够实现多条指令暂存,这样就能够减少取指操作次数,从而减少不必要的消耗,提高取指效率。

[0091] 实施例四

[0092] 在上述实施例一、实施例二和实施例三的基础上,在步骤3之后,步骤4之前还包括步骤Y,

[0093] 对PC指令每两条进行解析和执行时,首先判断PC指令是否为入栈指令或者出栈指令,

[0094] 如果是入栈指令,那么执行入栈指令;如果是出栈指令,那么执行出栈指令,如果不是入栈指令,也不是出栈指令,那么判断该条PC指令是否为跳转指令,执行步骤4的内容。

[0095] 需要说明的是,入栈指令对应于入栈操作,具体是将入栈指令存入PC栈中,PC栈用来实现函数嵌套时PC的保存与恢复,保存跳转指令的下一条指令位置,通过读取栈,恢复之前跳转指令的下一条指令位置。根据入栈指令中的目标地址进行重新取指,即将入栈指令中的目标地址标记为新的取指地址,进行取指,将取回的多条PC指令存入第N缓冲区中,其余执行步骤与跳转指令的执行步骤相同,在此不再赘述。

[0096] 出栈指令对应于出栈操作,出栈操作具体为将之前存入PC栈中的指令取出并执行。

[0097] 入栈指令和出栈指令是成对出现的,具体可参考现有技术中的入栈操作和出栈操作。

[0098] 实施例五

[0099] 在上述实施例一至实施例四的基础上,本发明专利还提供一种局部跳转指令取指装置。

[0100] 包括存储器、处理器和寄存器,所述寄存器内存储有计算机程序,

[0101] 所述处理器执行所述计算机程序实现如下步骤:将取指地址发送给存储器,并接收所述存储器返回的第一PC指令;将所述第一PC指令存入第一缓冲区;解析所述第一PC指令;当所述第一PC指令中存在跳转指令,且所述跳转指令的目标地址不在所述第一缓冲区内时,获取所述跳转指令的目标地址,将所述目标地址标记发送给存储器,并接收所述存储器返回的第二PC指令,将所述第二PC指令存入第N缓冲区,形成第N暂存指令,其中N为大于1的整数;所述跳转指令为有效跳转指令时,将所述第N缓冲区和所述第一缓冲区进行逻辑切换,将所述第N暂存指令标记为所述第一PC指令,将所述第一PC指令标记为所述第N暂存指令,并循环上述指令;

[0102] 所述处理器还用于执行,当所述第一PC指令中不存在跳转指令时,顺序执行所述

第一PC指令。

[0103] 最后应说明的是：以上实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：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；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

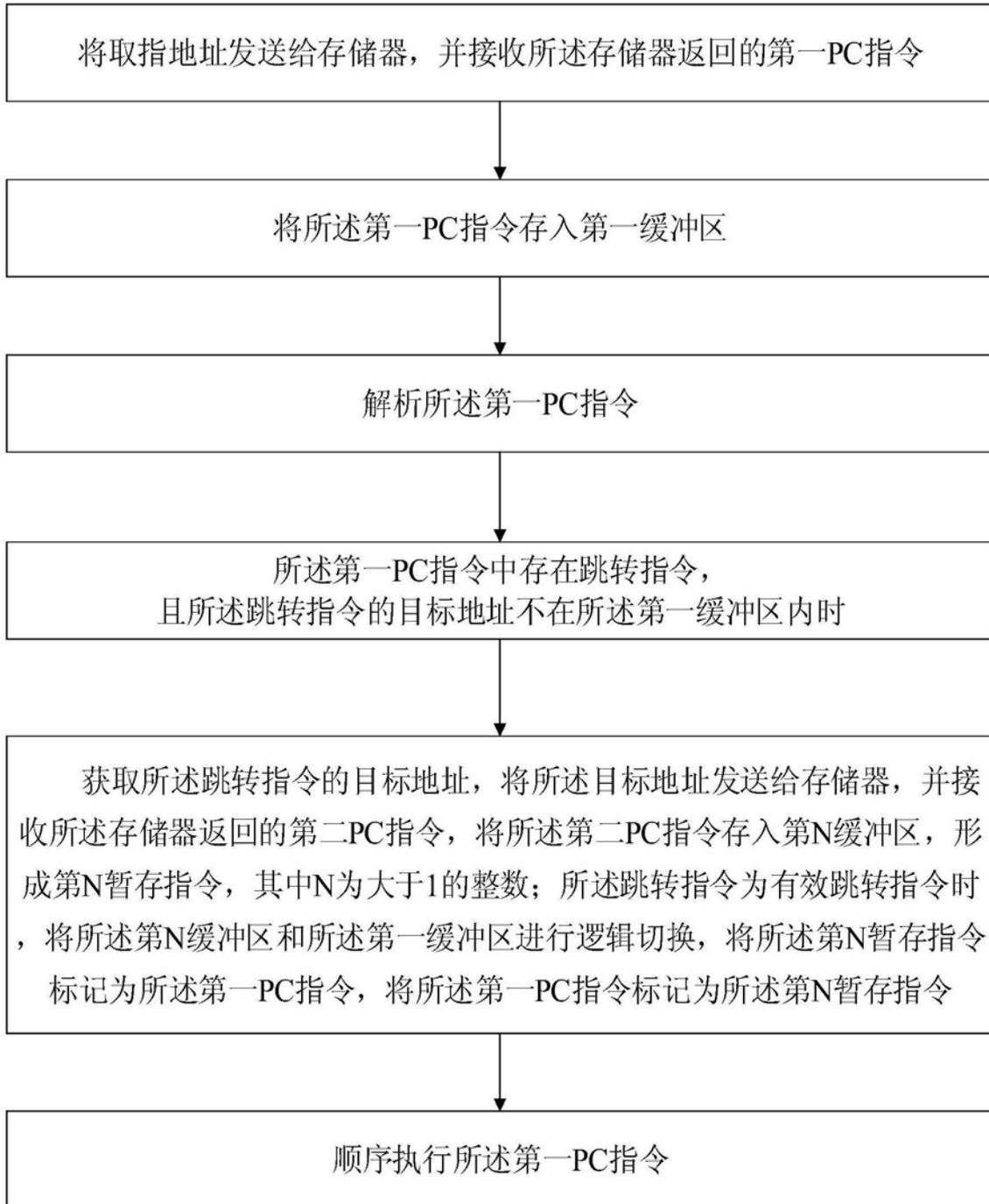


图1

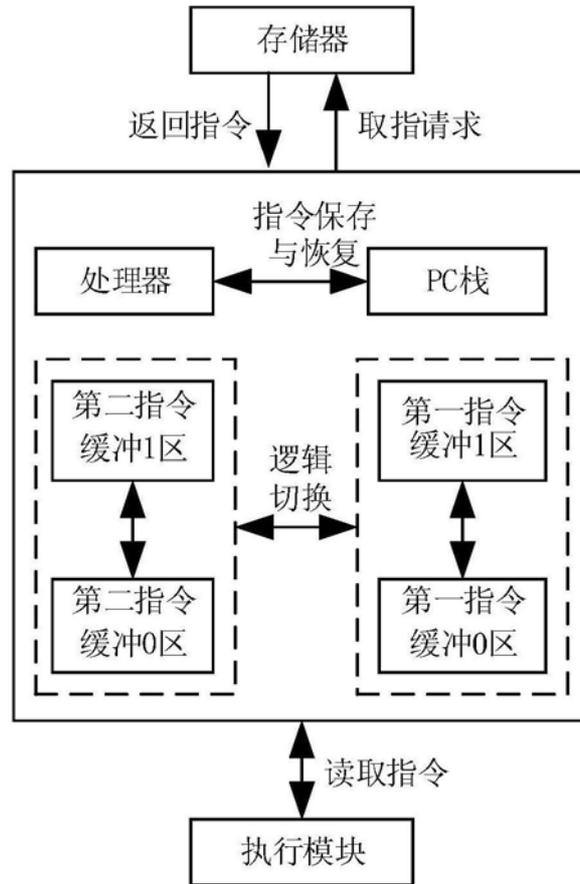


图2