

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.  
G06F 9/38 (2006.01)



# [12] 发明专利申请公开说明书

[21] 申请号 200510121761.3

[43] 公开日 2006年9月13日

[11] 公开号 CN 1831757A

[22] 申请日 2005.12.28  
[21] 申请号 200510121761.3  
[30] 优先权  
    [32] 2004.12.28 [33] US [31] 11/024,164  
[71] 申请人 英特尔公司  
    地址 美国加利福尼亚州  
[72] 发明人 A·海萨姆 D·奥伦斯坦恩  
    R·拉吉瓦 S·斯利尼瓦萨恩

[74] 专利代理机构 上海专利商标事务所有限公司  
    代理人 钱慰民

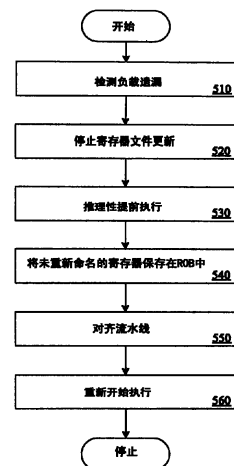
权利要求书 3 页 说明书 5 页 附图 5 页

## [54] 发明名称

中央处理单元中的提前执行

## [57] 摘要

根据一个实施例，公开了一种方法。该方法包括检测中央处理单元(CPU)中的加载遗漏，停止只读缓冲器(ROB)，推理性地撤回使ROB停止的指令以及后面的指令，当撤回时保持ROB中还没有被重新命名的寄存器，以及当从加载遗漏接收数据时清除CPU的通路。



- 1、一种方法，包括：  
检测中央处理单元（CPU）中的加载遗漏；
- 5 停止重新排序缓冲器（ROB）；  
推理性地撤回使 ROB 停止的指令以及后面的指令；  
当撤回时保留 ROB 中还没有被重命名的寄存器；以及  
当从加载遗漏接收数据时，清除 CPU 的流水线。
- 2、根据权利要求1所述的方法，其中，在停止 ROB 的步骤中包括：当加  
10 载遗漏到达 ROB 的头部时在一寄存器文件中停止寄存器文件更新。
- 3、根据权利要求1所述的方法，其中，导致 ROB 停止的指令的推理性提前和撤回以及随后的指令被执行，而不更新寄存器文件。
- 4、根据权利要求3所述的方法，其中，导致 ROB 停止的指令以及随后的  
指令的推理性提前和撤回被进一步执行，而不发布存储到存储设备。
- 15 5、根据权利要求3所述的方法，进一步包括：使用在寄存器文件中导致  
ROB 停止的指令的被停止状态重新开始执行。
- 6、根据权利要求1所述的方法，其中，在撤回时保留 ROB 中的寄存器的  
步骤包括：通过头部和尾部的指针调整将没有被重命名的寄存器从 ROB 的头部  
复制到尾部。
- 20 7、根据权利要求1所述的方法，其中，推理性地运行导致 ROB 停止的指  
令和随后指令的撤回的步骤中，进一步包括将寄存器数据从生产者微操作  
（uops）转发到消费者 uops。
- 8、根据权利要求7所述的方法，进一步包括：每当 uop 具有一被重命名的  
逻辑寄存器目的地时，就撤回该 uop。。
- 25 9、根据权利要求7所述的方法，进一步包括：每当 uop 具有一未被重命名  
的逻辑寄存器时，就回收该 uop 的 ROB 条目。
- 10、根据权利要求9所述的方法，进一步包括停止 uop 撤回直到 ROB 被装  
满。
- 11、根据权利要求10所述的方法，进一步包括通过提前 ROB 的头部指针，  
30 如果 ROB 被装满，则不停止 uop 的撤回。

12、根据权利要求 11 所述的方法，进一步包括提前 ROB 的头部指针而不放弃 uop 目的地寄存器的值。

13、一种计算机系统，包括：

主存储器设备，以及

5 与主存储器设备连接的中央处理单元（CPU），包括：

一只读缓冲器（ROB）；

一寄存器文件；以及

一通过停止 ROB 而执行推理性提前执行的与执行单元。

14、根据权利要求 13 所述的计算机系统，其中，所述 CPU 进一步包括撤回单元，在推理性提前执行期间推理性撤回使 ROB 停止的指令以及后面的指令。

15 15、根据权利要求 14 所述的计算机系统，其中，使 ROB 停止的指令和后面指令的推理性提前执行以及撤回被执行而不更新寄存器文件或者存储到主存储器设备。

16、根据权利要求 15 所述的计算机系统，其中，ROB 通过头部和尾部的指针调整，将还没有被重命名的寄存器从 ROB 的头部复制到尾部，通过这种方法，ROB 保持在撤回时还没有被重命名的寄存器。

17、根据权利要求 13 所述的计算机系统，其中，所述执行通过使用导致 ROB 在寄存器文件中停止的指令的停止状态来重新开始执行。

20 18、根据权利要求 13 所述的计算机系统，其中，所述执行单元通过将寄存器数据从生产者微操作（uops）转发到消费者 uops 以执行推理性提前执行。

19、一种中央处理单元（CPU），包括：

一只读缓冲器（ROB）；和

一寄存器文件；和

25 一通过停止 ROB 以执行推理性提前执行的与执行单元。

20、根据权利要求 19 所述的 CPU，其中，当加载遗漏到达 ROB 的头部时，所述执行单元通过在寄存器文件中停止寄存器文件更新从而停止 ROB。

21、根据权利要求 19 所述的 CPU，进一步包括一撤回单元，用于在推理性提前执行期间撤回使 ROB 停止的指令以及后面的指令。

30 22、根据权利要求 21 所述的 CPU，其中，使 ROB 停止的指令和后续指令

的所述推理性提前执行和撤回被执行，而不更新寄存器文件或存储到主存储器设备。

- 23、根据权利要求 19 所述的 CPU，其中的 ROB 通过头部和尾部的指针调整，将还没有被重命名的寄存器从 ROB 的头部复制到尾部，通过这种方法，保
- 5 持还没有被重命名的寄存器。

## 中央处理单元中的提前执行

## 5 发明领域

本发明涉及计算机系统；更特别地，本发明涉及中央处理单元（CPUs）。

## 背景技术

计算机系统 CPUs 中提前执行（runahead execution）的实现，是为了容许必须由主存储器提供服务的 CPU 高速缓存中的长等待时间加载遗漏。特别地，当  
10 从存储器读取数据时，由于长等待时间加载遗漏阻塞依次撤回（in-order retirement）长达数百个周期导致缓存器全停止，提前执行使用这些由于对上述缓存器重新排序而产生的空闲时钟周期。。

建议的提前执行模型包括检查寄存器状态，推理地执行位于加载遗漏的阴影区中（例如，遗漏的负载之后）的指令，直到遗漏的数据被读取，确保推理  
15 性的提前执行不会导致存储器状态的更新，使用坏位（poison bits）以确保调度程序不受阻，当遗漏数据返回时放弃推理性的提前执行状态，恢复被检查的寄存器状态，并且重新开始执行。

建议的提前方案存在的问题是，检查寄存器状态以及使用坏位以确保推理性提前执行不停止调度程序的步骤需要附加的硬件，这增加了 CPU 设计的复杂性  
20 性和成本。

## 附图简述

本发明通过实例方式加以说明，并且不限于附图中的各幅图，其中相同的参考指示类似的部件，其中：

图 1 是计算机系统的一个实施例的方块图；

25 图 2 描绘了 CPU 的一个实施例的方块图；

图 3 描绘了读取/解码单元的一个实施例的方块图；

图 4 描绘了撤回单元的一个实施例；

图 5 描绘了提前执行的实施例的流程图；

图 6 描绘了重新排序缓冲器的一个实施例；以及

30 图 7 描绘了重新排序缓冲器的另一个实施例。

## 详细说明

下文描述了 CPU 中的提前执行。该提前执行过程包括当加载遗漏到达重新排序缓冲器的头部时停止寄存器文件更新。然后，继续加载遗漏的推理性提前和撤回，以及遗漏之后的指令而不更新寄存器文件或者发布存储到存储器。当  
5 未重命名的寄存器被撤回时，它们被保留在重新排序缓冲器中。这是通过重新排序缓冲器的头部和尾部的指针调整将未重命名的寄存器从重新排序缓冲器的头部复制到尾部而实现的。接着，当数据遗漏返回时，通路被清除（flushed）。最后，使用寄存器文件中加载遗漏的冻结状态重新开始执行。

在以下本发明的详细说明中，为了提供对本发明的全面理解而阐明了多个  
10 具体的细节。然而，对于本领域中的技术人员来说，显而易见的是，没有这些具体的细节本发明也可以实现。在其他的例子中，为了避免使本发明难以理解，公知的结构和设备以方框图的形式示出，而不示出细节。

说明书中引用的“一个（one）实施例”或“一（an）实施例”意味着，实施例中描述的特殊特征，结构，或特性包括在本发明的至少一个实施例中。  
15 说明书中不同位置中出现的短语“在一个实施例中”并不需要全部都指向同一个实施例。

图 1 是计算机系统 100 的一个实施例的方块图。计算机系统 100 包括连接到总线 105 的中央处理单元（CPU）102。芯片组 107 也连接到总线 105。芯片组 107 包括存储器控制集线器（MCH）110。MCH110 可以包括连接到一主系统存储器 115 的一存储器控制器 112。主系统存储器 115 存储由 CPU102 或者系  
20 统 100 中包含的任意其他设备执行的数据和指令序列。

在一个实施例中，主系统存储器 115 包括动态随机存取存储器（DRAM）；然而，主系统存储器 115 也可以使用其他类型的存储器实现。附加的设备也可以连接到总线 105，诸如多个 CPUs 和/或多个系统存储器。MCH110 通过集线  
25 器接口连接到输入/输出控制集线器（ICH）140。ICH140 为计算机系统 100 中的输入/输出（I/O）设备提供接口。

图 2 描绘了 CPU102 的一个实施例的方块图。CPU102 包括读取/解码单元 210，发送/执行单元 220，撤回单元 230 以及重新排序缓冲器（ROB）240。读取/解码单元 210 是一个有序的单元，它从指令高速缓存（未示出）取出一用户  
30 程序指令流作为输入并且将该流解码为一系列体现该流的数据流的微操作

(uops)。

图 3 描绘了读取/解码单元 210 的一个实施例的方块图。读取/解码单元 210 包括指令高速缓存 (Icache) 310, 指令解码器 320, 分支目标缓冲器 330, 指令序列发生器 340 以及注册别名表 (RAT) 350。Icache310 是一个本地指令高速缓存, 它根据分支目标缓冲器 330 提供的索引来读取超高速缓冲存储器线的指令。

指令被送到解码器 320, 它将该指令转换成 uops。使用由序列发生器 340 提供的微代码将一些指令解码成一到四个 uops。该 uops 被排列并被转发到 RAT350, 在这里, 寄存器引用被转换成物理寄存器引用。然后这些 uops 被传送到 ROB240。

返回参考图 2, 发送/执行单元 220 是一个无序单元, 它接收数据流, 调度执行依赖于数据和资源有效性的 uops 并且暂时地存储推理性执行的结果。撤回单元 230 是一个有序单元, 它将暂时的、推理性的结果指定 (撤回) 为永久状态。

图 4 描绘了撤回单元 230 的一个实施例的方块图。撤回单元 230 包括一寄存器文件 (RF) 410。撤回单元 230 从 ROB240 中读取潜在的候选者用于撤回并且确定这些候选者中的哪些是原程序序列中的下一个。撤回的结果被写入到 RF410。

ROB240 是一个重新排序机构, 它通过有效地保持暂时的指令结果直到得知前面的指令结果, 从而维持一结构状态。根据一个实施例, ROB240 的实施用于简化 CPU102 中的提前执行, 正如下面将要更加详细讨论的。

如上所述, 提前执行使用由于重新排序缓冲器全停止而遇到的空闲时钟周期。这些停止是由于从主存储器读取数据时阻塞有序撤回达数百个周期的长等待时间加载遗漏的结果。图 5 描绘了提前执行的实施例的流程图。在处理模块 510 中, 检测到了一加载遗漏。在处理模块 520 中, 当一加载遗漏到达 ROB240 的头部时, RF410 的更新被停止。

在处理模块 530 中, 继续执行加载遗漏的推理性提前和撤回以及遗漏之后的指令。根据一个实施例, 执行推理性提前和撤回而不更新 RF410 或者发布存储到存储器 115。在处理模块 540 中, 当那些还没有被重命名的 RF410 中的该寄存器被撤回时, 它们被保存在 ROB240 中。在一个实施例中, 这是通过头部和尾部的指针调整, 将未重新命名的寄存器从 ROB410 的头部复制到尾部实现

的。

在处理模块 550 中，当从存储器 115 返回加载遗漏的数据时，CPU102 的流水线被清除。在处理模块 560 中，使用 RF410 中加载遗漏的冻结状态重新开始执行。在一个实施例中，将寄存器数据从发生器转发到消费者 uops 以便实施  
5 提前执行。由于提前模式中的 RF410 更新被冻结以避免执行检测寄存器状态，ROB240 和一转回数据旁路，被用于转发寄存器的值。其结果是，撤回过程被修改。

在一个实施例中，每当 uop 具有已经被重命名的逻辑寄存器目的地，则该 uop 被安全地撤回，而它的值被放弃。此外，由于寄存器已经被重命名，新读取  
10 的 uops 不需要该寄存器，而发送/执行引擎 220 中等待预定位置的读取器已经从 ROB240 或转回数据旁路中获取了该值。图 6 描绘了当 ROB240 满时，撤回 ROB240 中重命名的寄存器的动作的一个实施例。如图 6 中所示，条目被释放而值被放弃。

在进一步的实施例中，当一 uop 具有一个没有被重命名的逻辑寄存器时，  
15 则终止撤回直到它被重命名，或者直到 ROB240 装满为止。如果当 ROB240 装满时寄存器没有被重命名，通过提前 ROB240 的头部指针而不停止撤回，不需要放弃 uop 的目的地寄存器的值。在一个实施例中，这是通过提前 ROB240 的头部指针和尾部指针来实现的。

提前两个指针有效地将 uop 和它的值从 ROB240 头部移动到尾部，而不需  
20 要实际地读写 ROB240 条目。由于不改变 ROB240 中的位置将 uop 从 ROB240 的头部移动到尾部，一 RAT350 重命名表为该逻辑寄存器保持正确的位置。图 7 描绘了当 ROB240 装满时，撤回 ROB240 中没有重新命名的寄存器的动作的一个实施例。如图 7 中所示，与头部指针一起提前尾部指针，留下 uop 和它在 ROB240 和 RAT350 中的输出用于以后的读取器。

还执行了其他的修改以使在 CPU102 中的提前执行能够实现。在一个实施  
25 例中，ROB240 寄存器转发机构中具有重命名目的地址的 uops 被识别。在该实施例中，为了避免必须增加 RAT350 端口的数量，以半重命名带宽执行提前并且变得可用的读取端口被用于为所有资源以及重命名的 uops 的目的地读取 RAT350。由逻辑目的地索引的 RAT350 中的 ROB240 条目是重命名的 uop  
30 ROB240 条目。该 ROB240 条目中一重命名的位可以被设置为将条目标记为重



命名的。应当注意到，其他实施例中，可以简单地增加 RAT 端口的数量。

在一个进一步的实施例中，从推理性存储器推理性加载的数据在提前执行中被转发。在该实施例中，推理性存储被存储在存储缓冲器中，甚至在它们在 ROB240 中的“伪撤回”之后，以允许转发到可能需要存储数据的任意加载。

5 然而，当存储器缓冲器装满时，最旧的提前存储被放弃而不将这些存储发布到存储器 113，从而为新的提前存储提供空间。作为该机构的结果，用于从废弃的存储中接收数据的提前加载将改为从高速缓冲存储器中读取旧数据。此外，由于 RF240 的状态在加载遗漏点被冻结，当在提前模式中时，跳转执行清除（JEClear）被禁止。

10 上述的机构可以实现提前执行，同时为了提前执行而避免检查和恢复寄存器文件。此外，提供了一种快速，费用低的机构，用于通过 ROB 将寄存器的值从生产者传送到消费者 uops 而不必在撤回时更新寄存器文件。

15 然而，在阅读了以上的说明书之后，本发明的各种改变和修饰对于本领域的普通技术人员来说将无疑是显而易见的，应当理解的是，通过图解显示和描述的所有特定的实施例并不用于限定。因此，不同实施例的参考细节不用于限定权利要求的范围，在权利要求中只引用被认为对发明必要的那些特征。

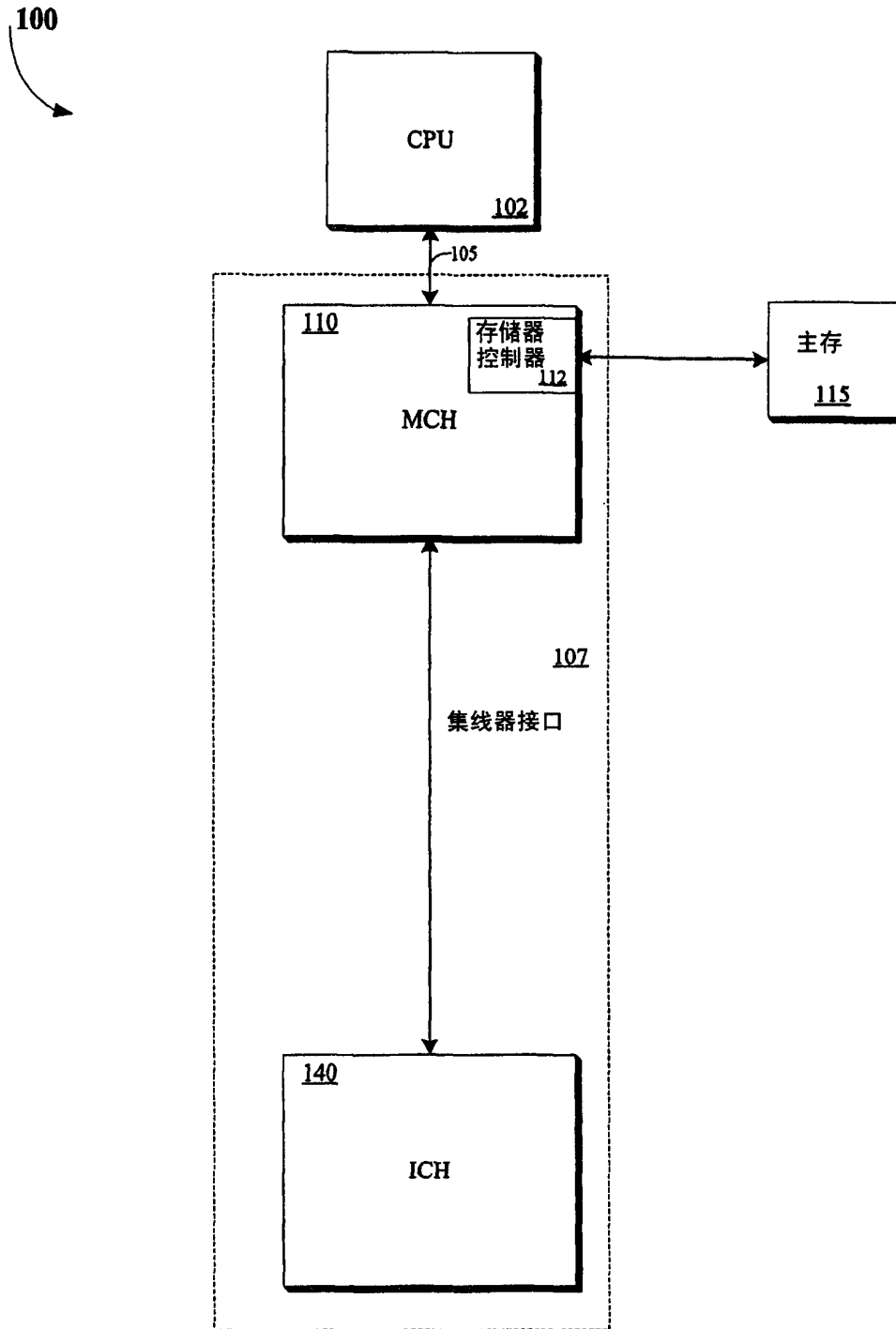


图 1

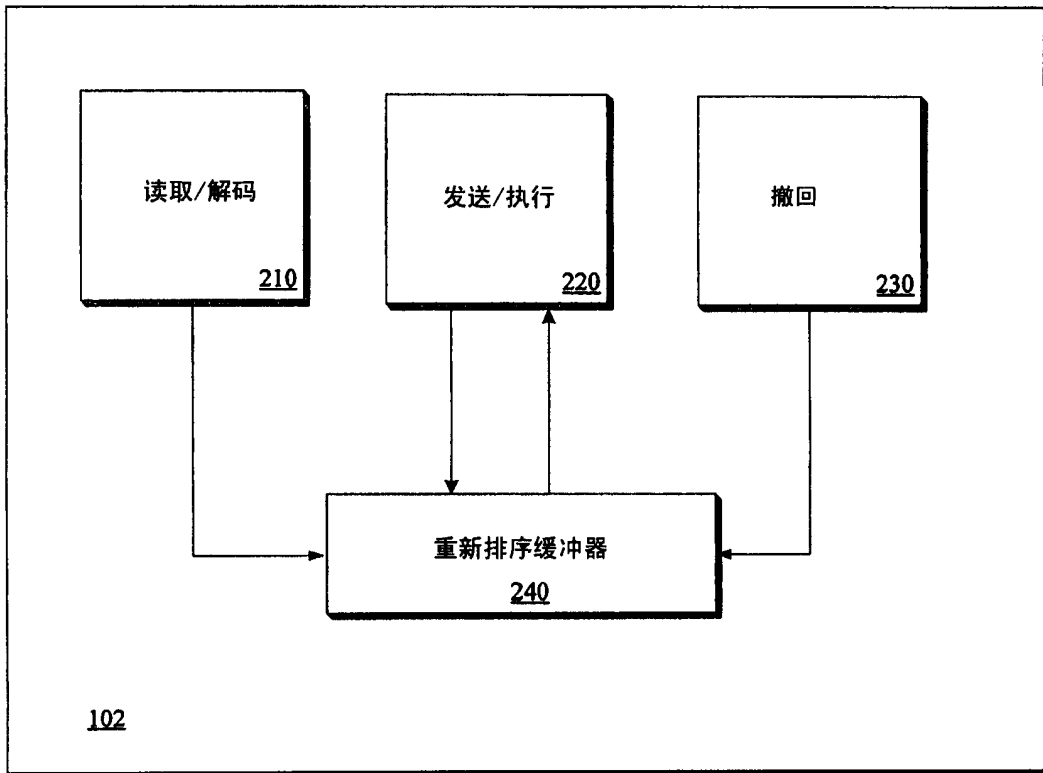


图 2

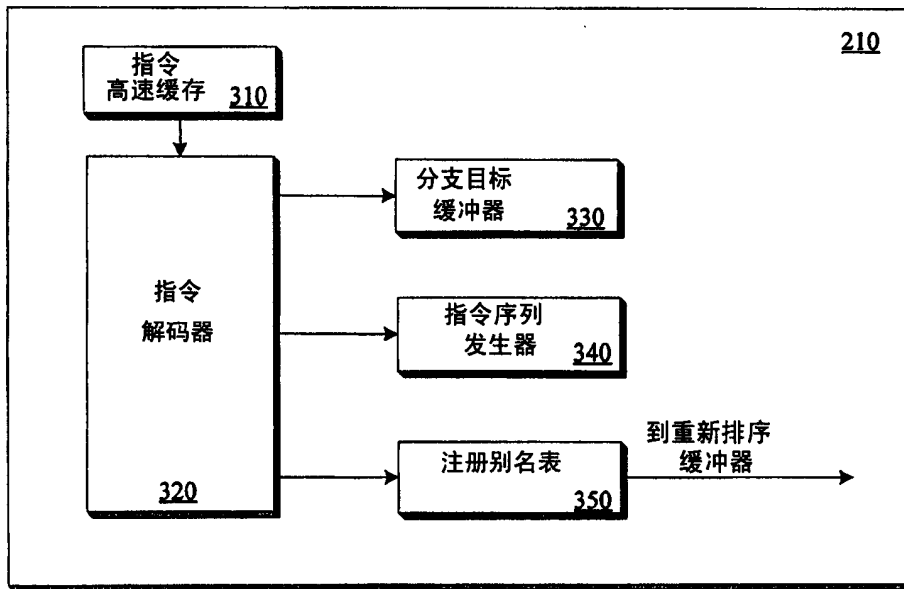


图 3

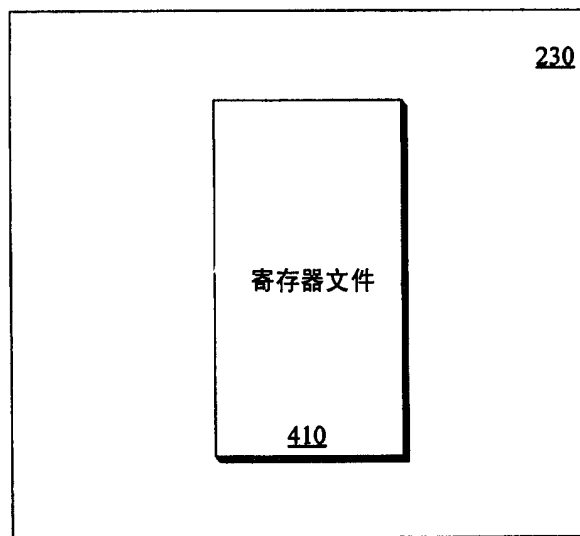


图 4

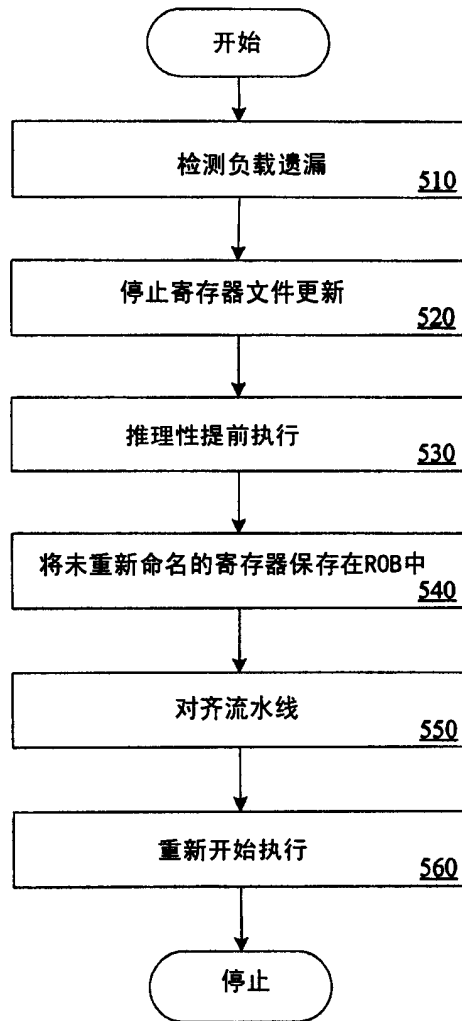


图 5

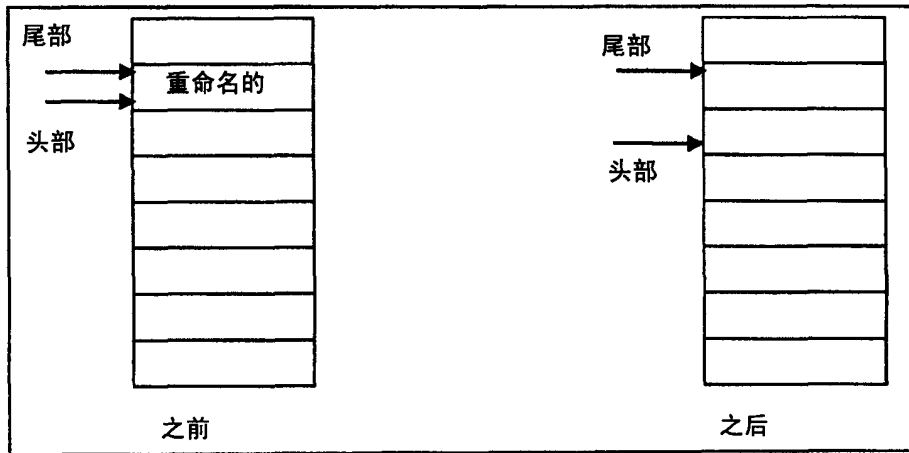


图 6

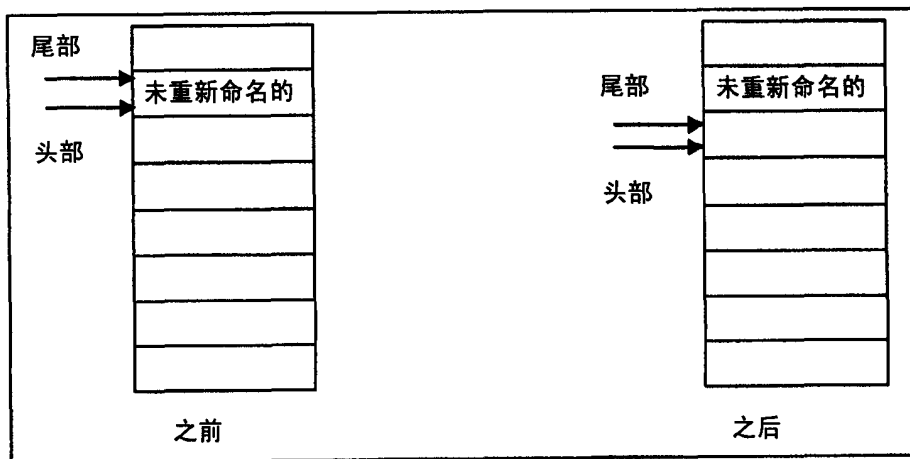


图 7