



[12] 发明专利说明书

专利号 ZL 200410011909.3

[45] 授权公告日 2007年3月21日

[11] 授权公告号 CN 1306420C

[22] 申请日 2004.9.24

[21] 申请号 200410011909.3

[30] 优先权

[32] 2003.9.30 [33] US [31] 10/675,732

[73] 专利权人 国际商业机器公司

地址 美国纽约州

[72] 发明人 戴维·A·卢克

[56] 参考文献

CN1300006A 2001.6.20

CN1287311A 2001.3.14

CN1195812A 1998.10.14

CN1308744A 2001.8.15

审查员 赵小宁

[74] 专利代理机构 北京市柳沈律师事务所

代理人 马莹 邵亚丽

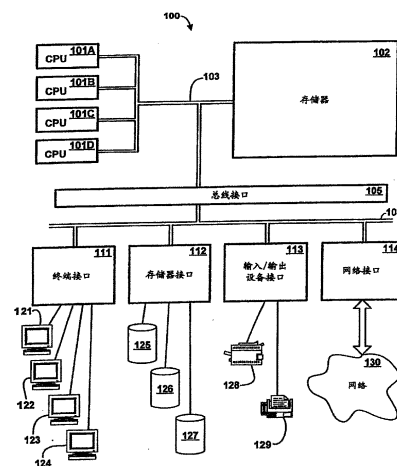
权利要求书 6 页 说明书 31 页 附图 18 页

[54] 发明名称

利用永久历史页表数据预取数据到高速缓存的装置和方法

[57] 摘要

本发明涉及利用永久历史页表数据预取数据到高速缓冲存储器的装置和方法。其中，计算机系统包括主存储器、至少一个处理器和至少一级高速缓冲存储器。该系统在存储器中、最好在页表中维护相对于每个可寻址页面的引用历史数据。引用历史数据最好用于确定应当将页面的哪些可高速缓存子单元预取到高速缓冲存储器中。引用历史数据最好是升或降计数器，它在可高速缓存子单元被装载到高速缓冲存储器中并且被处理器引用的情况下增量，而在该子单元被装载到高速缓冲存储器中但在被去除前未被引用的情况下减量。引用计数器因此表示根据近期历史该子单元将在近期被引用的大致可能。



1. 一种数字数据处理设备，包括：

存储器，所述存储器包含页表，所述页表具有对应于可寻址页面的多个页表项，其中，该多个页表项中的每一个对对应的可寻址页面的多个可高速缓存子单元中的每个分别包含各自的可高速缓存子单元引用历史数据；

至少一个处理器；

至少一个高速缓冲存储器，用于暂时存储来自所述存储器的数据；和

预取引擎，所述预取引擎将可寻址页面的选择的可高速缓存子单元预取到所述至少一个高速缓冲存储器中，所述预取引擎使用所述可高速缓存子单元引用历史数据来选择用于预取的可寻址页面的可高速缓存子单元。

2. 按照权利要求1所述的数字数据处理设备，其中，所述预取引擎在发生对用于选择性预取的页面的至少一个预定事件时识别所述页面，并且响应于对用于选择性预取的页面的识别、使用所述可高速缓存子单元引用历史数据来选择被识别的用于预取的页面的可高速缓存子单元。

3. 按照权利要求2所述的数字数据处理设备，其中，对于所述被识别的页面的所述预定事件包括将所述页面的地址翻译数据装载到地址翻译高速缓冲存储器结构。

4. 按照权利要求3所述的数字数据处理设备，其中，所述地址翻译高速缓冲存储器结构是有效地址到实际地址翻译表（ERAT）。

5. 按照权利要求1所述的数字数据处理设备，其中，所述引用历史数据包括升或降计数器，它在出现至少一个第一组预定事件时增量，而当出现至少一个第二组预定事件时减量。

6. 按照权利要求5所述的数字数据处理设备，其中，所述升或降计数器当相应的可高速缓存子单元被装载到所述高速缓冲存储器中并且当其在所述高速缓冲存储器中被引用时增量，而当所述相应的可高速缓存子单元被装载到所述高速缓冲存储器中并且当其在所述高速缓冲存储器中未被引用时减量。

7. 按照权利要求1所述的数字数据处理设备，其中，所述数字数据处理设备包括多个在多个高速缓冲存储器级上的高速缓冲存储器，并且其中所述预取引擎向不同的高速缓冲存储器级预取数据。

8. 按照权利要求 1 所述的数字数据处理设备, 其中, 所述数字数据处理设备包括:

多个处理器;

在多个高速缓冲存储器级上的多个高速缓冲存储器, 其中, 所述高速缓冲存储器级中的第一级包括多个高速缓冲存储器, 每个高速缓冲存储器分别与单个相应的处理器相关联, 并且其中所述高速缓冲存储器级中的第二级包括至少一个由多个处理器共享的高速缓冲存储器。

9. 按照权利要求 1 所述的数字数据处理设备, 其中, 所述数字数据处理设备还包括:

地址翻译机制, 它将在所述至少一个处理器上执行的任务的地址空间中的有效地址翻译为在所述数字数据处理系统的全局地址空间中的虚拟地址, 并且将所述虚拟地址翻译为与在所述存储器中的物理存储位置对应的实际地址。

10. 按照权利要求 9 所述的数字数据处理设备, 其中, 所述预取引擎还将地址翻译数据预取到所述地址翻译机制的至少一个地址翻译高速缓冲存储器结构中。

11. 一种向数字数据处理设备的至少一个高速缓冲存储器预取数据的方法, 包括步骤:

对于所述数字数据处理设备存储器中的多个存储器页面中的每个维护至少一个各自的升或降计数器, 每个所述计数器在出现相对于在与所述计数器对应的存储器页面中的数据的第一类事件时增量, 而在出现相对于在与所述计数器对应的存储器页面中的数据的第二类事件时减量; 并且

根据对应于存储器页面的所述至少一个升或降计数器的值来确定是否预取存储器页面中的数据。

12. 按照权利要求 11 所述的预取数据的方法,

其中, 所述维护至少一个升或降计数器的步骤包括对每个所述存储器页面维护多个在对应于该存储器页面的页表中的页面项中的相应的升或降计数器, 每个升或降计数器分别对应于由在所述页表中的对应页面项表示的存储器页面的相应的可高速缓存子单元, 每个所述计数器在出现相对于在对应于所述计数器的可高速缓存子单元中的数据的所述第一类事件时增量, 而在出现相对于在对应于所述计数器的可高速缓存子单元中的数据的所述第二类事件时减量; 并且

其中，所述确定是否预取存储器页面中数据的步骤包括：根据对应于可高速缓存子单元的升或降计数器的值来确定是否预取存储器中页面的可高速缓存子单元。

13. 按照权利要求 12 所述的预取数据的方法，其中，所述可高速缓存子单元对应于所述数字数据处理设备的至少一个高速缓冲存储器的一个高速缓冲存储器行的大小。

14. 按照权利要求 12 所述的预取数据的方法，其中，所述确定是否预取页面中的数据的步骤包括：

在出现相对于所述页面的至少一个第三类事件时识别用于选择预取的页面；

响应于对用于选择预取的页面的识别，使用所述可高速缓存子单元引用历史数据来选择用于预取的被识别页面的可高速缓存子单元。

15. 按照权利要求 14 所述的预取数据的方法，其中，相对于所述被识别页面的所述预定事件包括向地址翻译高速缓冲存储器结构中装载用于所述页面的地址翻译数据。

16. 按照权利要求 11 所述的预取数据的方法，

其中，所述第一类事件包括向所述数字数据处理设备的高速缓冲存储器中装载所述数据，并且当所述数据在高速缓冲存储器中时引用该数据；并且

其中，所述第二类事件包括向所述数字数据处理设备的高速缓冲存储器中装载所述数据，并且在所述数据被引用之前将所述数据去除出所述高速缓冲存储器。

17. 一种处理器，用于数字数据处理设备，包括：

指令单元，用于确定指令序列；

执行单元，用于以所述指令序列来执行指令；

至少一个高速缓冲存储器，用于暂时存储来自所述数字数据处理设备存储器的数据，以供所述指令单元和所述执行单元中的至少一个使用；

预取引擎，所述预取引擎向所述至少一个高速缓冲存储器中预取所述存储器的可寻址页面的选择的可高速缓存子单元，所述预取引擎使用从所述存储器获得的可高速缓存子单元引用历史数据来选择用于预取的可寻址页面的可高速缓存子单元；以及

引用历史维护逻辑单元，所述引用历史维护逻辑单元响应于对于各可高速

缓存子单元在所述处理器中发生的引用历史事件来更新与每个相应的可高速缓存子单元对应的所述引用历史数据。

18. 按照权利要求 17 所述的用于数字数据处理设备的处理器，其中，所述预取引擎在发生对于一个可寻址页面的至少一个预定事件时识别用于选择预取的所述可寻址页面，并且响应于对用于选择预取的可寻址页面的识别而使用所述可高速缓存子单元引用历史数据来选择用于预取的、被识别的可寻址页面的可高速缓存子单元。

19. 按照权利要求 18 所述的用于数字数据处理设备的处理器，其中，相对于所述被识别的可寻址页面的所述预定事件包括向地址翻译高速缓冲存储器结构中装载用于所述页面的地址翻译数据。

20. 按照权利要求 19 所述的用于数字数据处理设备的处理器，其中，所述地址翻译高速缓冲存储器结构是有效地址到实际地址翻译表 (ERAT)。

21. 按照权利要求 17 所述的用于数字数据处理设备的处理器，其中，所述引用历史数据包括升或降计数器，它在出现至少一个第一组预定事件时增量，而在出现至少一个第二组预定事件时减量。

22. 按照权利要求 21 所述的用于数字数据处理设备的处理器，其中，所述升或降计数器当一个相应的可高速缓存子单元被装载到所述高速缓冲存储器并当其在所述高速缓冲存储器中被引用时则增量，而当所述相应的可高速缓存子单元被装载到所述高速缓冲存储器并且当其在所述高速缓冲存储器中未被引用时减量。

23. 按照权利要求 17 所述的用于数字数据处理设备的处理器，其中，所述预取引擎还预取地址翻译数据到地址翻译机制的至少一个地址翻译高速缓冲存储器结构。

24. 一种数字数据处理设备，包括：

至少一个处理器；

存储器，所述存储器包含页表，所述页表具有对应于可寻址页面的多个页表项，其中，多个所述页表项中的每个包含对于对应的可寻址页面的永久引用历史数据，所述永久引用历史数据在对应的可寻址页面的整个生命周期内被保存在存储器中；

至少一个高速缓冲存储器，用于暂时存储来自所述存储器的数据；和

预取引擎，所述预取引擎从所述可寻址页面向所述至少一个高速缓冲存储

器预取数据，所述预取引擎使用所述永久引用历史数据来选择用于预取的数据。

25. 按照权利要求 24 所述的数字数据处理设备，其中，所述数字数据处理设备包括多个处理器，所述永久引用历史数据是基于在所述多个处理器中发生的引用事件的。

26. 按照权利要求 24 所述的数字数据处理设备，

其中，每个所述页表项对于对应可寻址页面的多个可高速缓存子单元中的每个包含各自的可高速缓存子单元引用历史数据；

其中，所述预取引擎根据相应的可高速缓存子单元引用历史数据来确定是否预取在存储器中的页面的可高速缓存子单元。

27. 按照权利要求 24 所述的数字数据处理设备，其中，所述永久引用历史数据包括升或降计数器，它在出现至少一个第一组预定事件时增量，而在出现至少一个第二组预定事件时减量。

28. 按照权利要求 27 所述的数字数据处理设备，其中，所述升或降计数器当一个相应的可寻址页面被装载到所述高速缓冲存储器并当其在所述高速缓冲存储器中被引用时则增量，而当所述相应的可寻址页面被装载到所述高速缓冲存储器并且当其在所述高速缓冲存储器中未被引用时则减量。

29. 按照权利要求 24 所述的数字数据处理设备，其中，所述数字数据处理设备包括多个在多个高速缓冲存储器级上的高速缓冲存储器，并且其中，所述预取引擎向不同的高速缓冲存储器级预取数据。

30. 按照权利要求 24 所述的数字数据处理设备，其中，所述数字数据处理设备还包括：

地址翻译机制，它将在所述至少一个处理器上执行的一个任务的地址空间中的有效地址翻译为在所述数字数据处理系统的全局地址空间中的虚拟地址，并且将所述虚拟地址翻译为与在所述存储器中的物理存储位置相对应的实际地址。

31. 按照权利要求 30 所述的数字数据处理设备，其中，所述预取引擎还向所述地址翻译机制的至少一个地址翻译高速缓冲存储器结构中预取地址翻译数据。

32. 一种数字数据处理设备，包括：

多个处理器；

存储器，所述存储器包含页表，所述页表具有对应于可寻址页面的多个页

表项，其中，多个所述页表项中的每个包含对于对应的可寻址页面的永久引用历史数据，所述永久引用历史数据包含所述多个处理器的引用历史数据；

至少一个高速缓冲存储器，用于暂时存储来自所述存储器的数据；和

预取引擎，所述预取引擎从所述可寻址页面向所述至少一个高速缓冲存储器预取数据，所述预取引擎使用所述永久引用历史数据来选择用于预取的数据。

33. 按照权利要求 32 所述的数字数据处理设备，

其中，每个所述页表项对于对应的可寻址页面的多个可高速缓存子单元中的每个包含各自的可高速缓存子单元引用历史数据；

其中，所述预取引擎根据各可高速缓存子单元引用历史数据来确定是否预取在存储器中页面的可高速缓存子单元。

34. 按照权利要求 32 所述的数字数据处理设备，其中，所述永久引用历史数据包括升或降计数器，它在出现至少一个第一组预定事件时增量，而在出现至少一个第二组预定事件时减量。

利用永久历史页表数据预取数据到高速缓存的装置和方法

技术领域

本发明涉及数字数据处理硬件，具体涉及用于数字数据处理设备的处理单元的、高速缓冲存储器和支持硬件的设计和制造。

背景技术

在二十世纪的后半部分，开始了被称为信息革命的现象。虽然信息革命是范围比任何一个事件或机器更大范围的历史发展，但是没有任何设备可以比数字电子计算机更能代表信息革命。计算机系统的发展已经肯定是一场革命。每年，计算机系统更快地发展，存储更多的数据，并且向它们的用户提供更多的应用。

现代计算机系统典型地包括中央处理单元（CPU）和用于存储、检索和传送信息所需要的支持硬件，诸如通信总线和存储器。它还包括：与外部世界通信所需要的硬件，诸如输入/输出控制器或存储控制器；与其附接的设备，诸如键盘、监视器、磁带驱动器、硬盘驱动器、耦接到网络的通信线路等。CPU 是系统的核心。它执行包括计算机程序的指令，并且指导其它系统部件的操作。

从计算机硬件的角度来看，多数系统以基本相同的方式来操作。处理器能够执行有限的一组很简单的操作，诸如算术、逻辑比较和从一个位置向另一个移动数据。但是每个操作被执行得很快。用于指导计算机执行大量这样的简单操作的程序给出计算机正在进行一些复杂事情的假象。通过执行实际上相同的一组很简单操作但是更快地执行它，可以使用户有可能感知到计算机系统的新的或改进的能力。因此，继续对计算机系统的改进需要使这些系统更快。

计算机系统的整体速度（也称为“流量”）可以被粗略地测量为每单元时间执行的操作的数量。在概念上，对于系统速度的所有可能改进中的最简单者是提高各种部件的时钟速度，尤其是处理器的时钟速度。例如，如果所有动作以两倍的速度运行但是以相同的方式工作，则系统将以一半的时间来执行给定的任务。由许多离散部件构成的早期的计算机处理器通过缩小和组合部件、最终

将整个处理器封装为在单个芯片上的集成电路而容易得到较大的时钟速度改进，并且通过进一步缩小尺寸和其它改进而提高的时钟速度继续成为目标。除了提高时钟速度，有可能通过提高每个时钟周期执行的操作的平均数来提高单独的 CPU 的流量。

典型的计算机系统可以存储大量的数据，并且处理器可以被调用来使用这个数据的任何部分。通常用于存储海量数据的设备（诸如旋转的磁性硬盘驱动器存储单元）要求较长的等待时间来存取存储在其上的数据。如果处理器在每次执行一个操作时都要直接从这样的海量存储设备存取数据，则它将花费几乎它的所有时间来等待存储设备返回数据，而它的流量将实际上很低。结果，计算机系统在分层的存储器或存储设备中存储数据，每个在后的层具有更快的存取，但是存储较少的数据。在最低层是一个或多个海量存储单元，它们在较慢的设备上存储所有的数据。向上一层是主存储器，它一般是半导体存储器。主存储器具有比存储单元小得多的数据容量，但是具有快得多的存取。再向上是高速缓冲存储器，它们在层次上可以是单层或多层（第一级是最高层）的。高速缓冲存储器也是半导体存储器，但是比主存储器更快，并且也具有更小的数据容量。甚至可以考虑外部存储的数据——诸如可以通过网络连接访问的数据——是在计算机系统本身的海量存储单元以下的另一层，因为从网络连接（例如因特网）获得的数据数量可能更大，但是访问时间更慢。

数据被从海量存储器向主存储器、向高速缓冲存储器移动以便由处理器使用。理想地，数据在被处理器需要之前移动到最接近处理器的高速缓冲存储器层，以便当需要它时，处理器不必等待较长等待时间的数据存取的完成。但是，因为任何高速缓冲存储器层的容量都仅是主存储器的容量的一小部分，而主存储器本身又仅是海量存储单元的容量的一小部分，因此不可能简单地向所述高速缓冲存储器装载所有的数据。必须有某种技术可用于选择要存储在高速缓冲存储器中的数据，以便当处理器需要特定的数据项时，它将可能在那里。

高速缓冲存储器通常被划分为称为行的数据单位，行是可以独立地装载到高速缓冲存储器中或从高速缓冲存储器中去除的数据最小单位。在简单的高速缓冲存储器设计中，数据在需要时、即在出现高速缓存未命中时被装载到高速缓冲存储器的行中。即，当处理器需要不在高速缓冲存储器中的一些数据（高速缓存未命中）时，从较高速缓冲存储器低的层或从存储器获得所需要的数据，并将其装载到一个高速缓冲存储器行中。这意味着必须选择去除高速缓冲存储器

数据的一现有行。存在用于选择要去除的高速缓冲存储器现有行的各种技术和设备。

根据要求装载在概念上是简单的，但是导致大量的高速缓存未中和结果所致的、处理器等待所需数据时的空闲时间。因此，许多复杂的处理器设计都采用某种形式的高速缓冲存储器数据预取。预取仅仅意味着存在预测技术，借此，在处理器实际需要数据之前，向一个或多个高速缓冲存储器层装载被认为可能不久就需要的数据。如果预测技术是准确和及时的，则数据将在第一级（L1）高速缓冲存储器未中发生之前位于第一级高速缓冲存储器中，或在高速缓冲存储器的其它层中，从那里可以比从主存储器更快地存取数据。

存在几种公知的预取技术，它们可以单独或组合使用。一种技术是顺序预取，即预取地址空间的下一顺序的行，其所基于的理论是这是接着需要的最可能的数据。可将一确认位用于顺序预取，其中，如果在高速缓冲存储器行中的数据当其被最后预取到高速缓冲存储器时被实际访问，则置位该位，否则将其复位。如果确认位对下一个顺序数据行置位，则该行将被预取到高速缓冲存储器中。

另一种技术涉及使用分支目标缓冲器。分支目标缓冲器是一组分支目标地址，每个地址与相应的高速缓冲存储器行相关联。分支目标缓冲器在处理器引用相关联的高速缓冲存储器行之后立即记录由处理器引用的地址。当一个高速缓冲存储器行被引用时，它的相关联的分支目标地址是预取的良好候选者。分支目标缓冲器比顺序预取更为通用，因为分支目标可以是任何地址，但是它要求实质更多的实现开销。

被称为“技术流预取”的另一种预取技术要求使用特殊的技术流地址缓冲器。每个缓冲器项是由处理器先前依序访问的地址的列表。如果该列表中的第一地址被处理器引用，则剩余的地址是预取的良好候选者。像分支目标缓冲器一样，这种技术涉及一些实现开销。

虽然诸如这些的传统预取技术具有一些预测值，但是它们仍然非常有限。已经观察到在许多环境中，处理器花费大量时间空闲在高速缓存未中上。利用更精确和全面的预取技术，实质的性能改进将是可能的，它们将大大降低高速缓存未中的频率。

发明内容

本发明提供一种数字数据处理设备，包括：存储器，所述存储器包含页表，所述页表具有对应于可寻址页面的多个页表项，其中，该多个页表项中的每一个对对应的可寻址页面的多个可高速缓存子单元中的每个分别包含各自的可高速缓存子单元引用历史数据；至少一个处理器；至少一个高速缓冲存储器，用于暂时存储来自所述存储器的数据；和预取引擎，所述预取引擎将可寻址页面的选择的可高速缓存子单元预取到所述至少一个高速缓冲存储器中，所述预取引擎使用所述可高速缓存子单元引用历史数据来选择用于预取的可寻址页面的可高速缓存子单元。

本发明还提供一种向数字数据处理设备的至少一个高速缓冲存储器预取数据的方法，包括步骤：对于所述数字数据处理设备存储器中的多个存储器页面中的每个维护至少一个各自的升或降计数器，每个所述计数器在出现相对于在与所述计数器对应的存储器页面中的数据的第一类事件时增量，而在出现相对于在与所述计数器对应的存储器页面中的数据的第二类事件时减量；并且根据对应于存储器页面的所述至少一个升或降计数器的值来确定是否预取存储器页面中的数据。

本发明还提供一种处理器，用于数字数据处理设备，包括：指令单元，用于确定指令序列；执行单元，用于以所述指令序列来执行指令；至少一个高速缓冲存储器，用于暂时存储来自所述数字数据处理设备存储器的数据，以供所述指令单元和所述执行单元中的至少一个使用；预取引擎，所述预取引擎向所述至少一个高速缓冲存储器中预取所述存储器的可寻址页面的选择的可高速缓存子单元，所述预取引擎使用从所述存储器获得的可高速缓存子单元引用历史数据来选择用于预取的可寻址页面的可高速缓存子单元；以及引用历史维护逻辑单元，所述引用历史维护逻辑单元响应于对于各可高速缓存子单元在所述处理器中发生的引用历史事件来更新与每个相应的可高速缓存子单元对应的所述引用历史数据。

本发明还提供一种数字数据处理设备，包括：至少一个处理器；存储器，所述存储器包含页表，所述页表具有对应于可寻址页面的多个页表项，其中，多个所述页表项中的每个包含对于对应的可寻址页面的永久引用历史数据，所述永久引用历史数据在对应的可寻址页面的整个生命周期内被保存在存储器中；至少一个高速缓冲存储器，用于暂时存储来自所述存储器的数据；和预取引擎，所述预取引擎从所述可寻址页面向所述至少一个高速缓冲存储器预取数

据, 所述预取引擎使用所述永久引用历史数据来选择用于预取的数据。

本发明还提供一种数字数据处理设备, 包括: 多个处理器; 存储器, 所述存储器包含页表, 所述页表具有对应于可寻址页面的多个页表项, 其中, 多个所述页表项中的每个包含对于对应的可寻址页面的永久引用历史数据, 所述永久引用历史数据包含所述多个处理器的引用历史数据; 至少一个高速缓冲存储器, 用于暂时存储来自所述存储器的数据; 和预取引擎, 所述预取引擎从所述可寻址页面向所述至少一个高速缓冲存储器预取数据, 所述预取引擎使用所述永久引用历史数据来选择用于预取的数据。

计算机系统包括主存储器、至少一个处理器和至少一层高速缓冲存储器。系统在存储器中、最好是在页表中维护相对于每个可寻址页面的引用历史数据。引用历史数据用于对页面中的数据进行预取决策。

在优选实施例的一个方面, 对于页内的多个可高速缓存的子单元的每个维护引用历史数据。按照本发明的可高速缓存的子单元是可以与其它可高速缓存的子单元无关地被装载到第一级高速缓冲存储器中的页面的任何部分。最小可能的可高速缓存子单元是第一级高速缓冲存储器行, 但是一个可高速缓存子单元可以是任何整数数量的高速缓冲存储器行。根据每个可高速缓存子单元的各自的引用历史来选择性地选择可高速缓存子单元用于预取。

在优选实施例的另一方面, 引用历史数据是升计数或降计数, 用于指示当页面中的数据被装载到高速缓冲存储器中时对其的引用频率。如果数据被装载到高速缓冲存储器中并且被处理器引用, 则递增引用计数; 如果同一数据被装载到高速缓冲存储器中并且在被从高速缓冲存储器中删除之前未被处理器引用, 则递减引用计数。引用计数因此表示根据近期历史数据在近期将被引用的近似可能性。

在优选实施例的另一方面, 计算机系统包括有效到实际地址翻译表 (ERAT), 它将由处理器产生的有效地址翻译为在主存储器中的实际地址。每当新的页面项被装载到 ERAT 中时将查看引用历史数据, 并且根据引用历史数据来选择要预取到高速缓冲存储器中的数据。

在优选实施例中, 将如在此所述的根据页表中的历史数据来向高速缓冲存储器预取页面子单元与根据段表中的与独立页面相关联的数据来向页面翻译机制预取页数据相结合。所述段表的使用是相关的共同待决的 David A. Luick 的申请第_____号的主体, 题目为“使用段表数据预取页面数据的装置和方法”(受

让方的记录号为 ROC920030255US1), 它在与本申请相同的日期被提交。但是, 本发明可以与任何这样的、根据段表中的数据来预取页面数据的技术无关地来使用, 并且事实上甚至可以在没有段表的系统中使用本发明。

在本发明的优点中有与被高速缓存的数据相比较, 存储在页表中的历史数据相对持久。同一数据可以被装载和从高速缓冲存储器中去除, 甚至可以在包含历史数据的对应页表项的生命周期内多次从不同处理器中的不同高速缓冲存储器被装载和去除。这样持久地改善了数据的预测预取的范围和精度。按照本发明优选实施例的数据预取技术可以因此而大大降低高速缓存未中的频率和由

此导致的处理器的空闲时间。这样的技术涉及在实现所需要的附加存储资源和控制结构上的较小的成本。

附图说明

可以参照附图来最好地被理解关于本发明的细节、其结构和运行，其中同样的附图标记指的是相同的部件，并且其中：

图 1 是按照本发明的优选实施例的、使用根据持久的历史页表数据采用高速缓冲存储器预取的计算机系统的主要硬件部件的高层方框图。

图 2 是所述优选实施例的计算机系统 CPU 的主要部件的高层示意图。

图 3 是在优选实施例的计算机系统中支持的地址和地址翻译的不同形式的概念图。

图 4 更详细地示出按照所述优选实施例的、用于存储和寻址数据的各高速缓冲存储器和相关结构的层次结构。

图 5 示出按照所述优选实施例的、包含用于页面的可高速缓存子单元的历史引用数据的页表的结构。

图 6 示出按照所述优选实施例的、包含用于页面的可高速缓存子单元的历史引用数据的翻译后备缓冲器 (TLB) 的结构。

图 7 示出按照所述优选实施例的、包含历史页面引用数据的段表的结构。

图 8 示出按照所述优选实施例的、包含历史页面应用数据的段后备缓冲器 (SLB) 的结构。

图 9 示出按照所述优选实施例的、包含预取指示符的有效到实际地址表 (ERAT) 的结构。

图 10 示出按照所述优选实施例的、第一级高速缓冲存储器的结构。

图 11 示出按照所述优选实施例的预取引擎的结构，其中示出了预取引擎如何与各种其它硬件交互。

图 12 是用于在高层图解按照所述优选实施例的、由系统部件响应于由处理器产生的数据引用地址而提供数据所采取的行为的流程图。

图 13 是图解按照所述优选实施例的、由系统部件在第一级高速缓冲存储器中提供所需要的数据所采取的行为的流程图。

图 14 是图解按照所述优选实施例的、由系统部件在 ERAT 中提供所需要的页面项所采取的行为的流程图。

图 15 是图解按照所述优选实施例的、由系统部件在 SLB 中提供所需要的段项所采取的行为的流程图。

图 16 是图解按照所述优选实施例的、由系统部件在 TLB 中提供所需要的页面项所采取的行为的流程图。

图 17 是图解按照所述优选实施例的、用于预取高速缓冲存储器行的预取引擎的操作的流程图。

图 18 是图解按照所述优选实施例的、用于预取 ERAT 项的预取引擎的操作的流程图。

具体实施方式

参见附图，其中在几个视图中，同样的标号表示同样的部件，图 1 是按照本发明的优选实施例的、使用根据持久历史页表数据采用高速缓冲存储器预取的计算机系统 100 的主要硬件部件的表示。计算机系统 100 的主要部件包括一个或多个中央处理单元 (CPU) 101A-101D，主存储器 102、终端接口 111、存储接口 112、I/O 设备接口 113 和通信/网络接口 114，它们都经由总线 103、104 和总线接口 105 耦接以用于部件之间的通信。

系统 100 包括一个或多个通用可编程中央处理单元 (CPU) 101A-101D，在此通称为部件 101。在所述优选实施例中，系统 100 包括多个典型的较大系统的处理器；但是，系统 100 或者还可以是单 CPU 系统。每个处理器 101 执行存储在存储器 102 中的指令。指令和数据被装载到一层或多层高速缓冲存储器中 (图 1 中未示出) 以进行处理。存储器 102 是随机存取半导体存储器，用于存储数据和程序。存储器 102 在概念上是单片实体，可以明白存储器实际上是更复杂的装置，包括高速缓冲存储器和其它存储器件。

存储器总线 103 提供数据通信路径，用于在 CPU 101、主存储器 102 和输入/输出总线接口单元 105 之间传送数据。输入/输出总线接口 105 还耦接到系统输入/输出总线 104，用于向/从各种输入/输出单元传送数据。输入/输出总线接口 105 通过系统输入/输出总线 104 与多个输入/输出接口单元 111-114 通信，输入/输出接口单元 111-114 也被称为输入/输出处理器 (IOP) 或输入/输出适配器 (IOA)。系统输入/输出总线可以是例如工业标准 PCI 总线或任何其它适当的总线技术。输入/输出接口单元支持与各种存储和输入/输出设备的通信。例如，终端接口单元 111 支持一个或多个用户终端 121-124 的附接。存储接口单元 112 支

持一个或多个直接存取存储设备 (DASD) 125-127 (它们通常是旋转磁盘驱动器存储设备, 虽然它们还可以是其它设备, 包括被配置来作为对于主机的单个大存储设备出现的盘驱动器阵列) 的附接。输入/输出和其它设备接口 113 提供各种其它输入/输出设备或其它类型的设备的任何一个的接口。在图 1 的示范实施例中示出了两种这样的设备: 打印机 128 和传真机 129, 可以明白还有许多其它的这样设备, 它们可以是不同类型的。网络接口 114 提供从系统 100 到其它数字设备和计算机系统的一个或多个通信路径; 这样的路径可以包括例如一个或多个网络 130, 诸如因特网、局域网或其它网络, 或者还可以包括远程设备通信线路、无线连接等等。

应当明白, 图 1 意欲在高层上描述系统 100 的代表性的主要部件, 各部件可以比图 1 所示的具有更大的复杂性, 可以提供除了或补充图 1 所示的部件的部件, 这样的部件的数量、类型和配置可以改变。在此公开了这样的附加复杂性或附加变化的几个具体示例, 应当明白这些仅仅是举例, 而不必仅是这些变化。

虽然主存储器 102 在图 1 被示出为单片实体, 但是存储器 102 事实上包括至少一层、最好是多层高速缓冲存储器, 如在此进一步详细描述。存储器可以进一步被分布到和相关联于不同的 CPU 或不同的 CPU 组, 如在各种所谓的非一致存储器存取 (NUMA) 计算机结构的任何一种中公知的。虽然存储器总线 103 在图 1 中被示为提供在 CPU 101、主存储器 102 和输入/输出总线接口 105 之间的直接通信路径的较为简单的、单总线结构, 但事实上存储器总线 103 可以包括多个不同的总线或通信路径, 它们可以以任何一种形式来配置, 诸如在分层、星型或万维网配置中的点到点链路、多级总线、并行和冗余路径等。此外, 虽然输入/输出总线接口 105 和输入/输出总线 104 被示为单个的相应单元, 但是系统 100 可以事实上包括多个输入/输出总线接口单元 105 和/或多个输入/输出总线 104。虽然示出了多个输入/输出接口单元——它们将各种延伸到各种输入/输出设备的通信路径与系统输入/输出总线 104 分开, 但是还可以将一些或所有的输入/输出设备直接连接到一个或多个系统输入/输出总线。

图 1 所示的计算机系统 100 具有多个附接的终端 121-124, 例如, 这可能在多用户“大型”计算机系统是典型的。通常, 在这样的情况下, 所附接的设备的实际数量大于图 1 中所示的那些, 尽管本发明不限于任何特定大小的系统。计算机系统 100 还可以是通常仅仅包括单个用户显示器和键盘输入的单用户系

统，或是具有很少或没有直接用户界面但从其它计算机系统（客户端）接收请求的服务器或类似设备。

虽然以在高层描述和示出了各种系统部件，但是应当明白典型的计算机系统包括许多其它未示出的部件，它们对于理解本发明不是必要的。

图 2 是按照所述优选实施例的、包括某些相关联的高速缓冲存储器结构的 CPU 101 的主要部件的高层示图，其中以比图 1 所示的更详细地示出了 CPU 101。在这个实施例中，图 2 所示的部件被封装在单个半导体芯片中。CPU 101 包括指令单元部分 201、执行单元部分 211、第一级指令高速缓冲存储器（L1 I 高速缓冲存储器）205、第一级数据高速缓冲存储器（L1 D 高速缓冲存储器）206、有效到实际地址表（ERAT）207、第二级高速缓冲存储器（L2 高速缓冲存储器）208、存储器接口 209、地址翻译硬件 216、段后备缓冲器（SLB）217、翻译后备缓冲器（TLB）218 和预取引擎 220。一般，指令单元 201 从 L1 I 高速缓冲存储器 205 获得指令，解码这些指令以确定要执行的操作，并且解析分支条件以控制程序流。执行单元 211 对在寄存器中的数据执行算术和逻辑运算，并且从 L1 D 高速缓冲存储器装载或存储数据。L2 高速缓冲存储器 208 是第二级高速缓冲存储器，它一般大于 L1 I 高速缓冲存储器或 L1 D 高速缓冲存储器，用于向 L1 I 高速缓冲存储器 205 和 L1 D 高速缓冲存储器 206 提供数据。第二级高速缓冲存储器 208 耦接到外部存储器接口 209，它从外部存储位置装载数据或向其存储数据，该外部存储器诸如第三级高速缓冲存储器或主存储器 102。

ERAT 207 是寻址高速缓冲存储器，用于为在其它高速缓冲存储器结构中的某些数据提供地址翻译。在 SLB 217 和 TLB 218 中的地址翻译数据被翻译硬件 216 访问和翻译以构造在 ERAT 207 中的地址翻译项。在此更全面地解释这些单元的操作。

指令单元 201 包括分支单元 202、指令解码/分派单元 203 和指令寄存器和缓冲器 204。来自 L1 I 高速缓冲存储器 205 的指令在执行之前被装载到缓冲器 204 中。依赖于 CPU 的设计，可以存在多个缓冲器（例如，一个用于一个顺序的指令序列，其它用于转到的位置），其中每个可以包含多个指令。解码/分派单元 203 从缓冲器之一接收要执行的当前指令，并且解码该指令以确定要执行的操作或分支条件。分支单元 202 通过评估分支条件来控制程序流，并且从 L1 I 高速缓冲存储器 205 重新填充缓冲器。

L1 I 高速缓冲存储器 205 和 L1 D 高速缓冲存储器 206 是分立的指令和数据

高速缓冲存储器，用于向指令和执行单元提供数据。第二级高速缓冲存储器是未区分的高速缓冲存储器，其中包含指令和非指令数据。通常，通过指令或执行单元从第一级高速缓冲存储器取出或向第一级高速缓冲存储器存储数据，如果在第一级高速缓冲存储器中不可获得数据，则从第二级高速缓冲存储器 208 向第一级高速缓冲存储器中装载该数据，第二级高速缓冲存储器 208 继而从外部位置获取所述数据，然后所述数据被从第一级高速缓冲存储器向对应的单元传送。依赖于处理器的设计，有可能绕过第一级高速缓冲存储器而从第二级高速缓冲存储器 208 向执行或指令寄存器装载数据。存储器总线接口 209 处理通过存储器总线 103 的数据传送，数据可以经由总线接口 105 传输到第三级高速缓冲存储器、主存储器或输入/输出单元。

执行单元 211 包括：一组通用寄存器 212，用于存储数据；整数算术逻辑单元（ALU）213，用于响应于由指令单元 201 解码的指令而对在 GP 寄存器 212 中的数据执行算术和逻辑运算。执行单元还包括复杂操作子单元 214，用于执行数学密集的运算，诸如浮点运算。子单元 214 可以包括其本身的特殊寄存器（未示出）。整数 ALU 213 和复杂操作子单元 214 通常被实现为多级流水线。除了在图 2 中所示的部件之外，执行单元还可以包括附加的专用寄存器和计数器、用于从高速缓冲存储器或存储器、控制硬件等获取数据或向它们存储数据的装载和存储硬件。尤其是，除了整数 ALU 和复杂操作子单元 214 之外，执行单元 211 还可以包括流水线（未示出）。可以在处理器 101 内存在附加的流水线，诸如指令获取和解码流水线。

在指令单元或执行单元进行实际的数据引用之前，预取引擎 220 预先将数据取到 ERAT 207 和高速缓冲存储器中。预取引擎 220 特别使用历史引用数据来预测应当高速缓存哪些数据。在此更详细地描述预取引擎的操作。

在所述优选实施例中，CPU 101 支持多个（最好两个）线程的同时执行。对于每个线程有独立的一组 GP 寄存器 212（未示出）以及某些其它结构。但是，所支持的线程数可能不同，并且本发明也可以用于仅支持单线程的执行的处理器中。

虽然已经在高层上描述和示出了各种 CPU 部件，但是应当明白，所述优选实施例的 CPU 包括未示出的许多其它部件，它们对于理解本发明不是必要的。例如，在典型的设计中要求各种附加的专用寄存器。而且，可以明白，图 2 的 CPU 仅仅是 CPU 结构的一个示例，CPU 101 内部件的数量、类型和布置上可以

存在许多变化，除了所述的那些之外还可以存在未示出的部件，不是所有所述部件都存在于 CPU 设计中。例如，缓冲器和高速缓冲存储器的数量和配置可以不同；执行单元流水线的数量和功能可以不同；可以以不同的阵列或组来配置寄存器；专用的浮点硬件可有可无，等等。而且，CPU 101 可以具有简单或复杂的指令集，它支持对于每个处理器周期的单个操作或多个操作的分派。

第一级高速缓冲存储器 205 和 206 和第二级高速缓冲存储器 208 被示出在图 2 中作为 CPU 101 的一部分，因为每个 CPU 具有其本身相应的第一级和第二级高速缓冲存储器。在结构上，高速缓冲存储器可以被考虑为存储器 102 的一部分，或被考虑为 CPU 的一部分，或可以被整体地考虑为独立的实体。图 2 的图示意欲是典型的，而不意欲将本发明限定到任何特定的物理高速缓冲存储器的实现，处理器芯片或芯片组可以包括比图 2 所示的更多或更少的高速缓冲存储器。

所述优选实施例的系统 100 支持多级地址翻译，如图 3 中逻辑地所示。三种基本的寻址构造是有效地址 301、虚拟地址 302 和实际地址 303。“有效地址”是从处理器的指令和执行单元来看的地址。即，由指令单元 201 或执行单元 211 产生的、用来引用指令或数据的地址是有效地址。可以以本领域内公知的各种方式产生有效地址，例如：作为直接在指令中编码的地址（它可以或不可以与专用寄存器中的一些高位地址位连接，这种专用寄存器例如当开始执行新任务时不常改变）；作为与通用寄存器中地址的计算的偏移；作为与当前执行的指令的偏移；等等。在所述优选实施例中，有效地址包括 64 位，被编号为 0-63（0 是最高位位）。“虚拟地址”是一种操作系统构造，用于分立不同执行任务的地址空间。即，如果每个任务可以引用有效地址的全范围，则不同任务的有效地址空间必须被映射到更大的虚拟地址空间中以避免冲突。在某些计算机系统结构中，虚拟地址也可以被映射到海量存储器中的位置以用于存取所存储的数据。虚拟地址不是像有效地址那样被存储在寄存器中的意义上的物理实体；它是逻辑结构，是通过连接 52 位的虚拟段 ID 314 和有效地址的低 28 位（总共 80 位）而产生的。实际地址表示在存储数据的存储器 102 中的物理位置。实际地址的位数对于具体的机器不同，但是一般小于有效地址的位数。在图 3 的图示中，实际地址包括标号为 12-63 的 52 位。在所述优选实施例中，使用实际地址来寻址第一级、第二级和第三级高速缓冲存储器。

如图 3 所示，有效地址 301 包括 36 位的有效段 ID 311、16 位的页号 312

和 12 位的字节索引 313, 有效段 ID 占用了最高位的位置。虚拟地址 302 是从通过下述方式由有效地址构造的: 将 36 位的有效段 ID 311 映射为 52 位的虚拟段 ID 314, 将结果产生的虚拟段 ID 314 与页号 312 和字节索引 313 连接。通过下述方式从虚拟地址得到实际地址 303: 将虚拟段 ID 314 和页号 312 映射为 40 位的实际页号 315, 并且将实际页号与字节索引 313 连接。因为主存储器的页面包含 4K (即 2^{12}) 字节, 因此字节索引 313 (最低 12 位地址位) 指定页内地址, 并且不论地址是有效的、虚拟的或实际的都相同。较高的位指定一个页面, 并且因此有时被称为“有效页号”或“实际页号”, 视情况而定。

可以明白, 这些地址范围参数仅仅是可以改变的寻址方案的一个示例。尤其是, 一个系统可以支持不同的页面大小和/或多种页面大小。当使用较大的页面大小时, 可因此而期望提高可高速缓存的子单元的大小。

计算机系统 100 包括用于将由 CPU 101 产生的有效地址翻译为在存储器 102 或高速缓冲存储器中的实际地址的地址翻译机制。这个地址翻译机制在图 3 中以高概念层得到描述, 并且在此更详细地描述。地址翻译机制包括: 段翻译机制 321 (包括段表和段后备缓冲器), 用于将有效段 ID 311 映射为虚拟段 ID 314; 页面翻译机制 322 (包括页表和翻译后备缓冲器), 用于将虚拟段 ID 314 和页号 312 映射为实际页号 315。

图 4 更详细地示出按照所述优选实施例的、用于存储和寻址数据的各种高速缓冲存储器和相关联的结构层次结构。在这个实施例中, 除了主存储器之外还有三层高速缓冲存储器。来自主存储器的数据被首先装载到第三级的高速缓冲存储器 (L3 高速缓冲存储器) 401A、401B (在此通称为部件 401), 然后从这里被装载到第二级高速缓冲存储器 208 中, 最后被装载到 L1 I 高速缓冲存储器 205 或 L1 D 高速缓冲存储器 206 中。根据设计, 也可能在装载数据时绕过高速缓冲存储器层, 例如从第三级高速缓冲存储器直接向第一级高速缓冲存储器中装载数据。在所述优选实施例中, 多个处理器共享第三级高速缓冲存储器 401, 并且在系统 100 中可能存在多个第三级高速缓冲存储器。优选的是, 第三级高速缓冲存储器被多达四个处理器共享, 虽然在图 4 中仅示出了两个。在物理上, L3 高速缓冲存储器被封装得比主存储器更靠近它所关联的处理器。例如, 第三级高速缓冲存储器和它服务的处理器可以被封装在共同的印刷电路板上, 或者甚至被封装在共同的集成电路芯片上。

来自存储器中的页表 402 或段表 403 的数据可以直接被装载到 TLB 218 或

SLB 217 中而绕过通用高速缓冲存储器。还可以向第三级高速缓冲存储器 401 中装载页表数据,并且从第三级高速缓冲存储器填充 TLB。TLB 比 SLB 大得多,它经常用于高速缓存 TLB 数据。SLB 很小,以至于在第三级高速缓冲存储器中高速缓存 SLB 数据是不必要的。

在越高的层(越靠近处理器),高速缓冲存储器变得更快,并且存储越少的数据。在此所述的示范实施例中,在典型的大型计算机系统中,每个第一级高速缓冲存储器具有 128 字节大小的高速缓冲存储器行,存储 64KB 数据,并且需要 3 个周期来存取。第二级高速缓冲存储器具有 256 字节大小的高速缓冲存储器行,存储 2MB 数据,并具有 20 个周期的存取时间。第三级高速缓冲存储器具有 512 字节大小的高速缓冲存储器行,存储 32MB 数据,并具有 100 周期的存取时间。对于主存储器的存取时间更慢,可能接近 1000 个周期。但是可以明白,这些参数仅表示在使用当前技术的大型系统中的典型高速缓冲存储器。这些典型的参数可以随着技术的发展而改变。较小的计算机系统一般具有相应较小的高速缓冲存储器,并具有较少的高速缓冲存储器层。本发明不限于任何特定的高速缓冲存储器尺寸、存取时间、高速缓冲存储器行的大小、高速缓冲存储器层的数量、是否高速缓冲存储器在特定层被多个处理器共享或被专用于单处理器、或者类似的参数。

除了可执行的程序代码和由应用程序操纵的其它数据之外,存储器 102 还包括用于支持地址翻译的某些数据结构。特别是,存储器包括页表 402 和多个段表 403A-403C(在此统称为部件 403)。图 4 中图解了三个段表,可以明白段表的数量可以不同,并且通常大得多。页表 402 用于记录向当前在主存储器 102 中的实际页面的虚拟地址范围的分配。页表 402 还包括用于支持操作系统页面算法和其它功能所需要的数据。具体上,页表 402 包括用于支持预取的、页面的可高速缓存子单元的引用历史数据,如以下更详细所述。段表 403 记录在有效地址中的段 ID 到虚拟地址中的段 ID 的分配。因为每个执行任务具有其本身的、从有效地址向虚拟地址的映射,并且不同任务的有效地址可以重叠(虽然虚拟地址是唯一的,并且不重叠),因此每个任务具有其本身的段表 403。

在运行中,每个 SLB 217 高速缓存来自对应于 SLB 定位其中的处理器正在执行的任务的段表 403 的信息,用于将有效地址翻译为虚拟地址。每个 TLB 218 高速缓存来自页表 402 的信息,用于将虚拟地址翻译为实际地址。ERAT 207 是较高层的寻址高速缓冲存储器,包括用于直接将有效地址翻译为实际地址的项。

这些项是使用来自 TLB 218 和 SLB 217 的信息构造的。

图 5 是页表 402 结构的图示，示出了某些重要部件，其中包括用于页面的可高速缓存子单元的历史引用数据。参见图 5，页表 402 包括多个页表项 (PTE) 501。在运行中，页表用于响应于输入虚拟页号 (即，虚拟段 ID 314 和页号 312 的连接) 而提供实际页号 315。索引机制 502 确定与输入的虚拟地址的虚拟页号部分对应的页表项。索引机制的操作细节与本发明不相关，并且可以使用现在已知的或以后开发的任何形式的页表索引。

每个页表项包括虚拟页号字段 511、实际页号字段 512、各种附加的控制和维护数据 513、高速缓冲存储器行引用计数器阵列 514、页引用计数器 516。在阵列 514 内的每个计数器 515 是 3 位的计数器，分别对应于由页表项引用的实际页面各可高速缓存子单元。在所述优选实施例中，页面大小是 4KB，每个阵列 514 包括 16 个计数器 515，每个计数器对应于 256 字节的可高速缓存的子单元。例如，在阵列 514 中的第一个 3 位计数器 515 对应于字节地址范围 (在页面内的字节地址 313 的范围) '000'X 到 '0FF'X；阵列 514 中的第二个 3 位计数器对应于字节地址范围 '100'X 到 '1FF'X，等等。可以观察到，在这个实施例中，可高速缓存子单元的大小与第二级高速缓冲存储器行的大小相同。但是，可高速缓存的子单元也可以更大或更小，并且可以是第一级高速缓冲存储器、第三级高速缓冲存储器行的大小或一些更大的尺寸。页面引用计数器 516 是对应于整个页面的计数器 (优选的是 3 位)。

计数器 515 和 516 表达了历史数据引用模式。在所述优选实施例中，每个计数器是升或降计数器，它在发生某些事件时增量，而在发生其它事件时减量。计数器不能被增量到超过其上限 (例如 '111'B) 或被减量到低于其下限 ('000'B)。优选的是，如果作为高速缓存未中的结果、对应的高速缓冲存储器行被请求装载到高速缓冲存储器中，或者如果对应的高速缓冲存储器行被预取到高速缓冲存储器中并在以后被从高速缓冲存储器存取，则行引用计数器 515 增量。如果对应的高速缓冲存储器行被预取到高速缓冲存储器中，但是在被从高速缓冲存储器删除之前未被处理器存取，则行引用计数器减量。如果对于页面存在 ERAT 未中，或如果页面的项被预取到 ERAT 中并且随后用于引用数据，则页面引用计数器 516 增量。如果页面项被预取到 ERAT 中但是未被使用，则页面引用计数器减量。因此，计数器作为反馈机制，通过增量计数器来“奖赏”正确的预取猜测，而通过减量计数器来“惩罚”不正确的猜测。

在所述优选实施例中，所有的增量和减量是 1，但是其它的值也可以用于调整预取的数量。例如，如果增量的量大于减量的量，则计数趋向于较大，并且较多的行或页面将满足预取的标准，而如果增量的量小于减量的量，则较少的行或页面将满足标准。

优选的是，当在存储器中建立页表项时，行引用计数器 515 被初始化为零。可以选择当删除页表项（在下面的页面被调出）时将这些存储到存储器中，并且当对于同一虚拟存储器空间建立新的项时从存储器检索它们。但是，由于许多引用模式的迅速改变的特性，尤其是当引用商业交易数据时，不清楚在保存这些数字中存在足够的附加益处来判断附加的存储器存取。页面引用计数器 516 被初始化为段表阵列 716 中对应的页面计数器 718 的值——如果它存在的话；如果不存在这样的项，则页面计数器也被初始化为零。

图 6 是 TLB 218 的结构图示。TLB 实质上是某些页表数据的高速缓冲存储器。参见图 6，TLB 218 包括多个页地址项（PE）601，它们被布置在使用虚拟地址的散列函数来寻址的行 602 中。通常，散列函数包括虚拟页号的若干低序位。在每行 602 中的项 601 的数量是 TLB 的关联性（associativity）。在所述优选实施例中，TLB 218 具有关联性 8，可以明白，这个数可以不同。

每个项 601 包括虚拟页号字段 611、实际页号字段 612、各种附加的控制和维护位 613（诸如最近最少使用的（LRU）位，用于确定当需要装载新的项时去除哪个项）、行引用计数器 615 的阵列、页面引用计数器 616 和页面重写标志位（dirty bit）617。通常，虚拟页号 611 仅仅包括虚拟页号的不包括在散列函数中使用的位的部分，因为那些位是冗余的。通过下述方式来确定对应于虚拟地址的页面项：使用散列值来选择 TLB 行，并且将每个被选择的项的虚拟页号 611 与虚拟地址的剩余位相比较以找到正确的项。如果没有项相匹配，则地址不在 TLB 中（TLB 未中），并且必须从主存储器（或高速缓冲存储器）装载。虽然图 6 中（类似地在图 8、9 和 10 中）示出了通常使用的相关索引方案，可以使用索引方法。

在阵列 614 内的每个行引用计数器 615 是与由页面项引用的实际页面的相应可高速缓存子单元对应的 3 位计数器。当页面项被装载到 TLB 218 中时从在页表 402 中的对应页表项获得计数器 615 和 616，并且它们分别与在页表项内的计数器 515 和 516 具有相同的意义。

图 7 是存储器中段表 403 结构的图示，示出了某些重要的部件，其中包括

用于在段中选择页面的历史引用数据。在所述优选实施例中，对于在系统 100 中执行的每个任务存在独立的段表 403。参见图 7，段表 403 包括多个段表项 (STE) 701。在运行中，段表用于响应于输入的有效段 ID 来提供虚拟段 ID 314，该输入有效段 ID 与段表 403 对应的执行任务相关联。索引机制 702 确定对应于输入有效段 ID 的段表项 710。索引机制的操作细节与本发明无关，可以使用任何形式的索引。

每个段表项 701 包括有效段 ID 字段 711、虚拟段 ID 字段 712、各种附加的控制和维护数据 713、自动引用使能位 714、页面重写标志位 720 和页面引用历史阵列 716。阵列 716 包括多个页号 717 (每个 16 位)，和各相应的页面引用计数器 718。每个页面引用计数器 718 最好是 3 位计数器。

在阵列 716 中的页号 717 表示在对应于段表项的虚拟段 ID 内的可能的预取目标页。当初始要求存取段表项时，启动在阵列中的页号预取，如在此更详细描述。页面引用计数器与页表中的页面引用计数器 516 具有相同的意义，并且可以用于段表维护和/或预取决策。

在段表项中的页号阵列可以被类推到在页表项中的页面的可高速缓存子单元。但是，与可高速缓存子单元不同，页号的列表很稀疏。即，在每个段中存在 64K 的页面，但是这些页面中仅有小部分被表示在阵列 716 中。典型的任务在每个段中存取较小量的页面，即 10-20 个页面。在阵列 716 中的页面列表意欲表示页号的这个小子集。理想上，阵列 716 包括由任务使用的所有页号。但是，难于在不对阵列 716 的大小施加一些固定限制的情况下实现段表，这必然意味着，对于一些任务，阵列 716 可能不包括由该任务使用的所有页面。在所述优选实施例中，阵列 716 包括 8 个页号和 8 个对应的页面引用计数器。优选的是，将页号阵列与其它段表数据一起存储在存储器中，并且当建立或填充段表时将页调入存储器中。页面阵列因此而具有长的生命周期。

在阵列 716 中的页号列表可由处理器 101 自动维护，或由一些用户定义的软件算法维护。段表仅仅是在存储器中的数据结构，它可以被软件改变。用户可能希望执行他自己的对任务执行的分析以确定可能要引用哪些页面。在数据库和类似数据的情况下，这样的分析一般是不实用的，但是在程序指令的情况下，对典型执行路径的分析可以提供对于要存取的页面的准确预测，由此可以产生在阵列 716 中的页面列表。如果用户产生这样的列表，则他可能希望防止处理器改变它。自动引用使能位 714 用于指定由处理器对阵列 716 的自动维护。

当这个位被复位时，处理器将不改变阵列 716 中的项。当这个位被置位时，处理器将按照所感知的引用模式自动更新阵列 716。简而言之，处理器以被要求装载到高速缓冲存储器中的前 8 个页面来初始装载阵列 716。随后，当新页面被要求装载到 ERAT 的任何时间，将其页面引用计数器与阵列中页面的页面引用计数器相比较，并替换掉具有较低引用计数的页面。在此更详细地描述这个过程。

图 8 是 SLB 217 结构的图示。SLB 实质上是来自段表 403 的特定数据的高速缓冲存储器。参见图 8，SLB 217 包括多个段 ID 项 (SE) 801，它们排列在行 802 中，其中使用有效段 ID 的散列函数来寻址行 802。典型地，散列函数包括有效段 ID 的若干低序位。在每行 802 中的项 801 的数量是 SLB 的关联性。在所述优选实施例中，SLB 217 具有关联性 8，应当明白这个数量可以改变。

每个项 801 包括线程 ID 字段 811、有效段 ID 字段 812、虚拟段 ID 字段 813、各种附加控制和维护数据 814 (诸如 LRU 位，用于确定当需要装载新的项时去除哪个项)、自动引用使能位 815、页面索引指针 816、页面重写标志位 820、以及页面引用历史阵列 817。有效段 ID 812 仅仅包括未被用于散列函数的 ID 的部分，因为那些位是冗余的。通过下列方式来确定与输入的有效段 ID 对应的段 ID 项：使用散列值来选择 SLB 行，并且将在所选择的行中的每个项的线程 ID 811 和有效段 ID 812 与产生引用的线程的 ID 和引用的有效段 ID 的剩余位相比较。如果没有项匹配，则地址不在 SLB 中 (SLB 未中)，并且必须从主存储器中的段表装载。

页面引用历史阵列 817 包括在虚拟地址段内的一组页号 818 和对应的页面引用计数器 819。页号 818 和计数器 819 在段项 801 被初始地装载在 SLB 217 中时被从段表 403 中的对应段表项复制，并且分别与段表项内的页号 717 和计数器 718 具有相同的作用。

图 9 是 ERAT 207 的结构表示。ERAT 207 是高速缓存的、有效地址到实际地址的翻译，它是从 SLB 217 和 TLB 218 得到的。像 SLB 217 和 TLB 218 一样，ERAT 207 是使用散列函数寻址的相联高速缓冲存储器。参见图 9，ERAT 207 包括多个页地址项 (PE) 901，它们被排列在使用散列函数寻址的行 902 中，散列函数典型地包括若干有效页号的低序位，优选的是，ERAT 207 具有关联性 4，可以明白这个数量可以是不同的。

每个项 901 包括线程 ID 字段 911、有效页号字段 912、实际页号字段 913、用于确定要去除那个项的 LRU 位 914、预取位 (P 位) 阵列 915、单个页面引用

位 916。优选的是，单个 ERAT 207 被当前在处理器 101 中执行的所有线程共享，并且因为不同线程的有效地址空间可能重叠，因此要求线程 ID 911 识别 ERAT 项对应的线程。在优选实施例的 2 线程处理器中，线程 ID 911 是单个位。可以明白，还可以对每个线程使用独立的 ERAT，在这种情况下，线程 ID 是不必要的。优选的是，有效页号字段 912 仅包括有效页号中不包含在散列函数中使用的位的那个部分，因为那些位是冗余的。通过下列方式来确定与有效地址对应的页面项：使用散列值来选择 ERAT 行 902，并且将每个所选择的项的有效页号 912 与有效地址的剩余位相比较以找到正确的项。如果没有项匹配，则地址不在 ERAT 中（ERAT 未中），并且必须从 TLB 和 SLB（或从在主存储器中的表）装载。

在阵列 915 中的各单个 P 位分别对应于页面项引用的实际页面内的每个可高速缓存子单元。该 P 位用于确定是否预取对应的可高速缓存子单元（它等同于第二级高速缓冲存储器的单个高速缓冲存储器行）。该位当页面项被装载到 ERAT 中时被设置为适当的值，并且在 ERAT 中时不改变。如果计数器 615 等于或超过某个门限，则该位被设置为 1（指示数据应当被预取），其中计数器 615 对应于从其得到 ERAT 页面项的、TLB 项 601 中的同一可高速缓存子单元。在一个简单的实施例中，所述门限被固定在 1，以便在计数器非零的任何时间，将对应的 P 位置位，用于指示应当预取该可高速缓存子单元。但是，该门限也可以是可变的。

页面引用位 916 用于记录 ERAT 页地址项 901 是否已实际上被引用，即处理器是否已经产生对于在 ERAT 页地址项所对应的有效页面内的地址的任何引用。页面引用位 916 当项被装载到 ERAT 中时被设置为 0，随后根据数据引用而被更新。

图 10 是按照优选实施例的、第一级高速缓冲存储器的结构的表示。L1 I 高速缓冲存储器 205 和 L1 D 高速缓冲存储器 206 的结构是类似的，都可以用图 10 表示，除了 L1 I 高速缓冲存储器通常不要求页面重写标志位 1013，指令通常不可写入 L1 I 高速缓冲存储器中。参见图 10，第一级高速缓冲存储器包括高速缓冲存储器数据表 1005 和索引 1004。数据表 1005 包括多行数据 1001，它们在所述优选实施例中各包含 128 字节。索引 1004 包括索引项 1002 的行。每个索引项包括一个高速缓冲存储器行的实际地址和某些控制位，特别是引用位 1012、页面重写标志位 1013 和 LRU 位 1014。通过下列方式来引用一个高速缓冲存储

器行:对所期望数据的实际地址的一部分进行散列(hashing)以获得索引行 1003,将该行中的索引项的实际地址与输入的被引用地址相比较,并且如果存在匹配则选择与该行和索引项位置对应的高速缓冲存储器行。

引用位 1012 用于指示一个高速缓冲存储器行是否已经被引用。该位当所述行初始装载时被置为 0,而如果所述行在高速缓冲存储器中时曾经被引用则被置为 1。在响应于高速缓存未命中而装载的情况下,根据硬件实现方式,该位可以被立即置为 1 而不用首先被置为 0。除了引用位 1012,第一级高速缓冲存储器的元素都是传统的。

图 11 是按照所述优选实施例的预取引擎 220 的结构图示,其中示出了预取引擎如何与各种其它硬件交互。在所述优选实施例中,预取引擎具有用于执行两个独立但是相关联的功能的硬件。第一功能将数据预取到高速缓冲存储器,诸如第一级或第二级高速缓冲存储器。第二功能将地址翻译信息预取到地址翻译结构,特别是 ERAT 207。虽然按照所述优选实施例在单个机器中实现这两个功能,但是本发明不必须要求实现在此所述的第二功能。第二功能是下述 David A. Luick 的相关共同待决申请的主题:序号____,题目为“Apparatus and Method for Pre-Fetching Page Data Using Segment Table Data(用于使用段表数据预取页面数据的装置和方法)”(受让方的编号 ROC920030255US1),与本申请在同一日期被提交。

参见图 11,对于页面内的高速缓冲存储器行 ERAT 项包含一组预取位 915。当一个新项被装载到 ERAT 中时,或在一些情况下,当第一次引用现有的 ERAT 项(先前已经被装载)时,实际地址和对应的 P 位被放置在高速缓冲存储器行预取队列 1101 中。在所述优选实施例中,高速缓冲存储器行预取队列是 FIFO 队列,其中保有 16 个实际地址和对应的 P 位组,尽管这个数可以不同。

预取引擎以低优先级运行以盗取通常不使用的总线周期。当总线空闲时,在屏蔽寄存器 1102 中使用 P 位的增量硬件 1103 中和屏蔽指针 1104 中产生地址增量。屏蔽指针 1104 增加 1,如果对应于屏蔽指针的值的屏蔽寄存器 1102 的位位置是'1',则地址产生器 1106 将来自寄存器 1105 的实际页号与屏蔽寄存器值和与'00'X 的固定低位值连接,以产生在 256 字节边界上的实际地址,即一个第二级高速缓冲存储器行或一对第一级高速缓冲存储器行的地址。该地址被输入到高速缓冲存储器行提取请求器 1107,它启动向被寻址的高速缓冲存储器行的高速缓冲存储器装载。

预取引擎的另一部分向页面预取寄存器 1110 中复制一个 SLB 项。选择器 1112 根据当前页面索引从 SLB 项的页面引用列表中选择一个页面。将所选择的页地址与其对应的虚拟段 ID 连接以形成虚拟页地址来供 ERAT 预取请求器 1111 预取。ERAT 预取请求器启动从 TLB (或在必要时的页表) 预取对应的 ERAT 项。一旦一个页地址项被预取到 ERAT, 则所述 ERAT 项可以使得预取引擎启动页面中所选择的高速缓冲存储器行的高速缓冲存储器行预取。

在所述优选实施例中, 高速缓冲存储器行提取请求器 1107 可以向选择的高速缓冲存储器层装载。数据被装载到的高速缓冲存储器层可以依据不同的因素, 诸如机器状态、数据类型、什么引起预取行为、页面引用计数等。在在此所述的示范实施例中, 对于商业交易处理通常是, 如果从其得到预取地址的 ERAT 项在未实际被引用的情况下被预取到 ERAT, 则数据被装载到第二级高速缓冲存储器中, 如果 ERAT 项被请求装载 (即响应于 ERAT 未中而被装载)、或如果它早期被预取并且已经第一次被引用, 则数据被装载到第一级高速缓冲存储器中。但是, 在诸如科学计算的特定类型的应用中, 以高速缓冲存储器行的粒度进行有意义的预取要困难得多。在这样的环境中, 优选的是向诸如第三级高速缓冲存储器这样的某些较低的高速缓冲存储器层预取较大的数据块 (诸如一个页面或更多)。为此, 屏蔽寄存器 1102 具有替代输入 'FFFF'X, 它具有预取在一个页面中的所有高速缓冲存储器行的效果。典型地, 当一个页面的所有高速缓冲存储器行被预取时, 它们不被预取到第一级高速缓冲存储器, 而是被预取到较低层的高速缓冲存储器。预取的进一步的变化可以依据数据类型而改变预取的层或粒度。一般, 指令数据比数据库或类似的数据更可以被预测, 并且可以被更积极地预取 (即预取到较高的层)。预取的其它变化可以根据页面引用计数来改变预取的层或粒度。高引用计数意味着, 一旦一个页面被装载到 ERAT 中, 则它通常被引用。在这种情况下, 诸如到第一级高速缓冲存储器的较高层的高速缓冲存储器预取可以相对于低页面引用计数的情况更为合理。按照本发明的预取引擎可以单独地或组合地实现任何一个或所有的这些变化, 或可以实现其它变化。

图 12-18 是图解按照所述优选实施例的、由各种系统部件执行来支持数据的高速缓冲存储器寻址和预取的步骤。虽然图 12-18 是以流程图的形式, 但是可以明白, 在所述优选实施例中, 这些步骤由硬件部件执行。还可以明白, 在实际实施例中, 某些步骤与其它步骤被并行执行, 这些步骤可能以与所图示的不

同的次序被执行，并且在附图中所示的步骤的次序仅用于容易理解逻辑流程，而不必然意味着指示时间关系。在流程图中所示的一些步骤是常规的，但是仍然被示出以提供用于本发明的上下文环境。还可以明白，为了清楚起见，省略了对于理解本发明不必要的一些步骤。

参见图 12，处理器 101 产生数据引用的有效地址（方框 1201），它可以是它所要求的指令的地址、或作为操作数要被装载到寄存器中的数据的地址、或要从寄存器存储回存储器中的数据。处理器产生有效地址的散列函数以存取 ERAT 207 的行 902 以获得实际页号（方框 1202）。在被存取的行中的项 901 被与有效地址相比较以找到匹配（方框 1203）。如果未找到匹配（ERAT 未中），则从方框 1203 取“是”分支到方框 1204。在这种情况下，所需要的 ERAT 项 901 必须被产生和装载。ERAT 项的产生和装载在图 12 中被图示为方框 1204，并且在图 14 中更详细地被图解。

如果所需要的地址在 ERAT 中（从方框 1203 的“否”分支），或在如图 14 所示将其装载到 ERAT 中后，处理器使用来自 ERAT 项的实际页号的若干位来存取可用的第一级高速缓冲存储器（I 高速缓冲存储器 205 或 D 高速缓冲存储器 206，视情况而定）的一行（方框 1205）。如果所需要的数据不在第一级高速缓冲存储器中，则从方框 1206 取“否”分支。在这种情况下，必须从更低层高速缓冲存储器或从存储器获得所需要的数据。获得和向第一级高速缓冲存储器装载所需要的数据在图 12 中被示为方框 1207，并且在图 13 中被更详细地地图解。

在获得第一级高速缓冲存储器中正确的数据项后，处理器执行所需要的操作，例如向指令缓冲器或寄存器装载数据，或将寄存器中的数据存储在一级高速缓冲存储器（方框 1208）。处理器还置位第一级高速缓冲存储器被存取行的引用位 1012 和 ERAT 中被存取页面的引用位 916（方框 1209）。这些位用于指示事实上已对对应的高速缓冲存储器行或页面进行了引用。处理器还在必要时更新第一级高速缓冲存储器行的 LRU 位 1014 和/或页面重写标志位 1013（方框 1210）。这最后三个操作可以同时或以任何顺序被执行。数据引用于是结束。

如果在方框 1206 所需要的数据不在第一级高速缓冲存储器中，则它将被装载到第一级高速缓冲存储器。这个过程被示出在图 13 中。

参见图 13，在第一级高速缓冲存储器中存在高速缓存未中的情况下，处理器需要做两件事情：为第一级高速缓冲存储器获得一个包含所请求的数据引用的数据行，和去除一现有的高速缓冲存储器行以为新数据腾出空间。这两个操

作可以同时或并行进行，如图 13 所示。

为了去除现有行，首先使用任何可用的方法来选择一行以去除，在优选实施例中采用的方法是由 LRU 位 1014 指示的最近最少使用项（方框 1301）。如果被选择去除的行中的引用位 1012 被置位（方框 1302 的“是”分支），则处理器将与要被去除的高速缓存的行对应的、TLB 218 中的引用计数器 615 增量（方框 1303）。如果引用位 1012 未置位（方框 1302 的“否”分支），则处理器将 TLB 中的对应引用计数器 615 减量（方框 1304）。虽然 TLB 存取可能较慢，但是引用计数器的更新仅仅影响保存在 TLB 中的记录，而其它操作、特别是向第一级高速缓冲存储器装载新数据则不必等待引用计数器更新的完成。如果被选择去除的行中的页面重写标志位(dirty bit)1013 被置位（方框 1305 的“是”分支），则所选择的行被写出到第二级高速缓冲存储器（方框 1306）。如果页面重写标志位未被置位（数据还未被改变），则绕过方框 1306。优选的是，L1 I 高速缓冲存储器不包含页面重写标志位，因此在 L1 I 高速缓存未中的情况下，跳过步骤 1305 和 1306。

为了获得新数据，从 ERAT 先前获得的实际地址被用于存取第二级高速缓冲存储器。如果所需要的数据在第二级高速缓冲存储器中（方框 1307 的“是”分支），则数据被立即提取并装载到第一级高速缓冲存储器的所选择的行中（方框 1313）。如果数据不在第二级高速缓冲存储器中，则处理器查看第三级高速缓冲存储器（方框 1308），如果数据在那里则从第三级高速缓冲存储器装载数据（方框 1312），并且继续。如果数据不在第三级高速缓冲存储器中（方框 1308 的“否”分支），则存取主存储器。如果数据在主存储器中（方框 1309 的“是”分支），则将其装载到第三级高速缓冲存储器中（方框 1311），然后装载到第二级高速缓冲存储器中（方框 1312）和第一级高速缓冲存储器中（方框 1313）。如果所请求的数据不在主存储器中（方框 1309 的“否”分支）。则产生一个页面错误（方框 1310）以从存储器获得数据。由于从存储器获得数据的长等待时间，执行的任务一般被换出。

与提取和向第一级高速缓冲存储器装载新数据同时，查看 ERAT 中被引用页面的页面引用位 916（方框 1314）。如果该位被复位，则当前数据引用是自从装载 ERAT 项起对该页面的第一次数据引用。即，作为预取行为装载 ERAT 项。一般，在这种情况下，页面内的数据被预取到第二级或更低的高速缓冲存储器。既然发生了对页面的实际引用，因此更紧迫地需要同一页面其它高速缓冲存储

器行中的数据，于是应当将其预取到第一级高速缓冲存储器。因此，启动预取引擎，这在图 13 中示为方框 1315，并且在图 17 中更详细地被示出。预取引擎以低优先级运行，它可能需要等待其它操作以实际完成数据的预取。

在次参见图 12，如果在方框 1203 在 ERAT 207 中不能找到引用的有效地址，则必须将地址装载到 ERAT。该过程被示出在图 14 中。与高速缓存未命中类似，处理器需要去除一现有的 ERAT 项以为新的项腾出空间，并且通过存取 SLB 217 和 TLB 218 来产生一新的 ERAT 项。这两个操作可以同时或并行进行，如图 14 所示。

为了去除现有的 ERAT 项，首先使用 LRU 位 914 选择一个项以去除（方框 1401）。如果在所选择的项中的页面引用位 916 被置位（方框 1402 的“是”分支），则处理器将对应于要被去除的页面项增量 TLB 218 中的页面引用计数器 616（方框 1403）。如果引用位 916 未被置位（方框 1402 的“否”分支），则处理器将 TLB 中的对应页面引用计数器 616 减量（方框 1404）。

为了产生和装载新的 ERAT 项，将处理器产生的数据引用的有效段 ID 用于索引 SLB 217 中的项（方框 1405）。如果所需的段在 SLB 中（方框 1406 的“否”分支），则 SLB 项被用于将有效段 ID 翻译为它的对应的虚拟段 ID，它然后被用于索引 TLB 218 中的项（方框 1408）。如果段不在 SLB 中（方框 1406 的“是”分支），则所需的 SLB 项将首先被从在主存储器中的段表 403 装载到 SLB 中（在图 14 中被示为方框 1407，在图 15 中更详细地被示出）。

在索引 TLB 中的项时，如果所需的页地址在 TLB 中（方框 1409 的“否”分支），则根据 TLB 和 SLB 中的信息建立 ERAT 项，并将其装载到 ERAT 中选定的位置（方框 1411）。如果页地址不在 TLB 中（方框 1409 的“是”分支），则所需的 TLB 项将被首先从页表 402 装载到 TLB 中（在图 14 中被示为方框 1410，在图 16 中更详细地被示出），其后建立和装载 ERAT 项（方框 1411）。

在以新的项装载 ERAT 时，处理器继续执行到方框 1205 以尝试存取第一级高速缓冲存储器。虽然这被示出为附图中的下一个逻辑步骤，但实际上，ERAT 命中几乎总是也意味着高速缓存命中，并且在一些处理器的设计中，可能在 ERAT 装载结束之前在高速缓存命中的假设下开始取数据到第一级高速缓冲存储器中。

与装载 ERAT 同时，在必要时更新 SLB 中的页面引用历史阵列 817（方框 1412）。即，如果自动引用使能位 815 被置位，则将被装载到 ERAT 中的页地址

与 SLB 中的阵列项相比较。如果阵列 817 中已经有被装载到 ERAT 中的页面，则其页面引用计数器 819 通过下述方式被更新：从 TLB 218 中取出页面引用计数器 616 并且将其增量，以说明 ERAT 未中（TLB 中的计数器 616 此时不被更新）。如果在阵列 817 中未已有该页面，则将对于被装载页面的、TLB 218 的页面引用计数器 616 的增量值与当前在阵列 817 中的页面引用计数器 819 相比较以确定是否在阵列 817 中的任何页面应当被替换为装载到 ERAT 中的页面。如果被装载的页面的页面引用计数器的增量值大于阵列 817 中任何页面的页面引用计数器 819，则具有最低计数器值的页面在阵列 817 中被替换为装载到 ERAT 中的页面。以这种方式，页面引用历史阵列处于持续的维护下，并且将响应于实际的数据引用模式而改变。

理想地，阵列 817 中的页面次序对应于由执行的任务使用页面的顺序。可以单独或组合地使用两种可能的技术来保持正确的顺序。在第一种技术中，如果由于 ERAT 未中而被调入 ERAT 中的页面已经在列表上，则可以假定它未被足够快地预取。在这种情况下，它在列表上的位置与该列表中紧在其前的项交换。随着时间的连续交换将趋向于在该列表中将较早引用的页面向前移动。在第二种技术中，当向阵列 817 增加新的页面时，该页面被插入到与页面索引指针 816 的当前值对应的位置，将其它页号根据情况向右或左移动覆盖要被替换的页面项所需要多的位置。页面索引指针通过页号阵列 817 指示向 ERAT 预取的进程。在当前页面索引指针之前的页面已经被预取到 ERAT 中（或至少已经开始了预取）。在当前页面索引指针之后的页面还没有被预取。因此，页面索引指针的当前值提供了页面预取序列中新页面次序的大致近似。这两种技术都需要附加的硬件支持，并且被考虑为选用的特征。

在方框 1411 向 ERAT 装载数据还启动了预取引擎，这在图 14 中被示出为方框 1413，并且在图 17 中更详细地被示出。预取引擎以低优先级工作，它可能必须等待一些时间来实际完成数据的预取。

如果在方框 1406 所需要的段项不在 SLB 217 中，则该项必须被装载到 SLB 中。这个过程被示出在图 15 中。与 ERAT 未中的情况类似，处理器需要为新的项腾出空间，并且通过存取主存储器中的段表 403 来产生新的 SLB 项。这两个操作可以同时或并行进行，如图 15 所示。

因为通常仅仅存在少量有效段项，因此在 SLB 中可能存在未使用的空间用于装载新项，在这种情况下，方框 1502-1504 被绕过。如果在 SLB 中没有未使

用的空间,则自方框 1501 采取“否”分支。处理器然后使用控制位 815 来选择要去除的 SLB 项,控制位 815 可能是 LRU 位或支持一些其它去除技术的数据(方框 1502)。如果页面重写标志位 820 被置位(方框 1503 的“是”分支),则被选择来去除的 SLB 项将被首先存储到主存储器中对应的段表 403 中(方框 1504)。页面重写标志位可能因为当项在 SLB 中时页面引用历史阵列 817 被更新而被置位。如果页面重写标志位未被置位,这不必进一步作出任何事情来去除 SLB 项;新项将覆盖旧项。

为了获得 SLB 217 的新项,使用数据引用的有效段 ID 来存取与引起 SLB 未中的执行线程对应的、主存储器中的段表 403 (方框 1505)。如果所需要的段表项不在段表中(方框 1506 的“否”分支),则必须从存储器首先装载该段表项,并且产生缺页,它一般使执行的任务被换出(方框 1507)。实际上,因为由典型任务使用的段项的数量较小(经常仅仅占用单个页面),因此通常当启动一个任务时在段表中装载该任务的所有段项,以使通常不发生后续的缺页。但是,存在一些具有广泛引用模式的应用,其中可能出现段表中的页面错误。假定所需要的项在段表中(方框 1506 的“是”分支),则它被装载到 SLB (方框 1508),并且处理器继续到图 14 的方框 1408 以尝试访问 TLB。

在向 SLB 中装载新的项时,启动预取引擎的 ERAT 预取功能,这在图 15 中被表示为方框 1509,并且在图 18 中被更详细地示出。ERAT 预取功能使用新装载的 SLB 项的页面引用阵列 817 预取页面项到 ERAT 中,如在此更详细地所述。这个功能以低优先级工作,并且可能需要在完成前等待其它事件。

再次参见图 14,如果在方框 1410 所需要的页面项不在 TLB 218 中,则该项必须被装载到 TLB。这个过程被示出在图 16 中。与 ERAT 未中的情况类似,处理器需要为新的项腾出空间,并且通过存取存储器中的页表 401 或第三级高速缓冲存储器中的页表的被高速缓存的部分来获得新的 TLB 项。这两个操作可以同时或并行进行,如图 16 所示。

处理器按照任何常规技术使用控制位 613 和/或其它数据来选择要去除的 TLB 项(方框 1601)。如果页面重写标志位 617 被置位(方框 1602 的“是”分支),则首先存储被选择来去除的 TLB 项。作为页面引用计数器 616 和/或行引用计数器 615 改变的结果,页面重写标志位将经常被置位。如果页面重写标志位被置位,则处理器确定 TLB 项是否在第三级高速缓冲存储器中(方框 1603)。如果是,则所选择的项被写出到第三级高速缓冲存储器中的页表的某部分(方

框 1604); 否则, 它被写出到在主存储器中的页表(方框 1605)。如果页面重写标志位未被置位, 则不必在方框 1602 后进一步进行任何事情以去除 TLB 项, 因为新的项将覆盖旧的项。

为了获得 TLB 218 的新的项, 处理器使用虚拟页号(来自 SLB 的虚拟段 ID 和由处理器产生的页号)来试图存取在第三级高速缓冲存储器中的页表项(方框 1606)。如果页表项不在第三级高速缓冲存储器中(方框 1607), 则处理器试图存取在主存储器中的该项(方框 1608)。如果该项不在主存储器中(方框 1609), 则产生页面错误, 并且执行的任务被换出(方框 1610)。如果从主存储器返回该页面项(方框 1609 的“是”分支), 则它可以可选地被装载到第三级高速缓冲存储器中(方框 1611)。不论是从第三级高速缓冲存储器还是主存储器返回, 页表项都将响应于虚拟页号被作为新的页面项装载到 TLB 中(方框 1612), 并且处理器继续到图 14 的方框 1411。

在优选实施例中, 在 ERAT 未命中时, 和在某些第一级高速缓存未命中时, 预取引擎 220 被“启动”, 以便预取高速缓冲存储器行, 如在图 13 中的方框 1315 和在图 14 中的方框 1413 所示。“启动”预取引擎以预取高速缓冲存储器行意味着一项被加入到高速缓冲存储器行预取队列 1101。预取引擎以先入先出的顺序来从该队列检索项。由处理器、特别是由预取引擎所采取的预取高速缓冲存储器行的步骤被更详细地示出在图 17 中。

参见图 17, 预取引擎在方框 1701 等待直到在预取队列 1101 中存在至少一个页地址项。虽然方框 1701 被示出为软件典型的空循环, 但是可以明白, 因为预取引擎是硬件, 因此它仅仅在一个信号存在时响应它并且在其不存在时不响应它。在队列 1101 中的每个页地址项包括从 ERAT 项得到的数据, 特别是实际页号和 P 位阵列 916。队列项最好包括至少一位, 用于指示预先装载的目的高速缓冲存储器。特别是, 在所述优选实施例中, 作为在被预取的页面上任何位置的 ERAT 未命中或第一级高速缓存未命中的结果被启动的预取导致到第一级高速缓冲存储器的预取, 而作为基于段表数据的到 ERAT 的预取项的结果被启动的预取则导致到第二级高速缓冲存储器的预取。队列项还可以包括其它数据, 用于更复杂的预取决定。

如果在队列中存在项, 则最旧的项被选择并装载到实际页面寄存器 1105 和屏蔽寄存器 1102(方框 1702)。屏蔽指针 1104 初始被设置为 0。如果屏蔽指针指向的屏蔽寄存器 1102 中的屏蔽位是“1”(方框 1703), 则地址产生器 1106 将

来自寄存器 1105 的实际页地址与屏蔽指针值和与'00'X 的固定低序位值连接,以产生预取地址(方框 1704)。这个预取地址是在 256 字节边界上的实际地址,即一个第二级高速缓冲存储器行或一对第一级高速缓冲存储器行的地址。

高速缓冲存储器行预取请求器 1107 然后等待直到它被授权进行预取操作(方框 1705)。高速缓冲存储器预取应当发生在后台,而不严重中断其它行为。预取仅仅是预测的,自然必须比响应于实际高速缓存未中的数据提取操作采取更低的优先级。因此,如果其它数据提取操作正在使用可用的总线或其它资源,则预取请求器将等待。而且,因为预取操作通常需要多个资源和多个周期来完成,因此不总是可能提前确定预取是否将使其它操作在随后的周期内空闲。为了避免对其它处理的过多干扰,按照任何现在公知的或以后开发的调步技术,最好将预取操作“调步”。调步仅仅意味着对预取操作的频率有一定的限制,即使资源看起来可以获得以满足预取的请求。在简单的调步技术中,提取请求器将等待直到自最后预取操作被分派起、预定的周期计数已经过去。

当提取请求器被授权执行时,它使用先前产生的预取地址(方框 1706)和用于指示预取目的的队列中的位来分派预取请求。分派请求意味着产生信号,使得在被请求的地址的数据被提取到指定的高速缓冲存储器。一般,这遵循与任何其它数据提取相同的过程。

在分配预取请求时(或如果方框 1704-1706 由于在方框 1703 屏蔽位是'0'而被绕过),屏蔽指针被增量(方框 1707)。如果增量不引起指针转向 0(方框 1708 的“否”分支),则预取引擎继续到方框 1703 以处理当前页中的下一个高速缓冲存储器行。如果指针转向 0(方框 1708 的“是”分支),则在当前页中的所有高速缓冲存储器行已经被处理,并且预取引擎返回方框 1701。

在所述优选实施例中,预取引擎还执行到 ERAT 的页面项预取,这可以进一步触发在这些页面内的高速缓冲存储器行项的预取。通过预取 SLB 项中页面引用历史阵列 817 中的页面来完成向 ERAT 预取页面项。被选择来用于启动预取的 SLB 项在循环的基础上旋转。即,每个 SLB 项可以指定多个阵列 817 中的虚拟页面,其中每个虚拟页面是可能的预取目标。预取操作被从第一 SLB 项分派,然后选择下一个 SLB 项,并且从其分派下一个操作。具有未完成的预取页面的每个相继的 SLB 项将被选择,并且在返回到第一 SLB 项之前使得一个操作被分派。

用于提取页面项的预取引擎的“开始”仅仅意味着在 SLB 项中的一些控制

位或其它信息被置位，用于指示可以获得来预取到 ERAT 的页面。在适当的时候，SLB 项应当被选择，并且在阵列 817 中的其页地址的一个或多个被预取到 ERAT。图 18 中更详细地示出了由处理器、特别是预取引擎用于预取页面项而采取的步骤。

参见图 18，预取引擎在方框 1801 等待直到在 SLB 中存在至少一个段项具有到预取队列的页面。虽然方框 1801 被示出为软件中典型的空循环，但是可以明白，因为预取引擎是硬件，因此它仅仅在一个信号存在时响应它而如果它不存在则不响应它。在典型的操作中，当一个任务开始时，一个段表立即被装载，并且多个项将被放置到 SLB 中以启动该任务。这多个项将初始建立预取操作的积压待办事项。

如果在 SLB 中存在至少一个用于预取的项，则下一个项被选择(方框 1802)。选择以循环的方式工作，并且可以例如通过下列方式来完成：增量 SLB 中的指针，并且查看适当的位以看是否存在任何未完成的页面预取操作。所选择的项被装载到寄存器 1110。

选择器 1112 从在所选择的项中的页号阵列 818 选择虚拟页号(方框 1803)。所选择的虚拟页号是对应于页面索引指针 816 的值的数量。于是通过将选择的项的虚拟段 ID、所选择的虚拟页号和低位的零位 ('000'X) 连接以形成页面边界来构成虚拟地址(方框 1804)。

ERAT 预取请求器 1111 然后等待直到它被授权来进行 ERAT 预取操作(方框 1805)。如在高速缓冲存储器行预取的情况一样，ERAT 预取发生在后台，应当不仅服从其它操作，而且被调步，如上对于高速缓冲存储器行请求所述。一般，ERAT 预取具有比在第一级高速缓冲未中或 ERAT 未中的情况下的高速缓冲存储器行预取更低的优先级。

当 ERAT 提取请求器被授权执行时，它使用先前产生的预取地址来分派预取请求(方框 1806)。分派请求意味着产生信号，用于使得对于 ERAT 提取和产生地址项。一般，这遵循与 ERAT 未中时的提取相同的过程。

在分派预取请求时，在 SLB 中的所选择的项的页面索引指针增量(方框 1807)。如果页面索引指针已经超过其极限(方框 1808)，则在这个项的页表上的所有页面已经被预取到 ERAT。在这种情况下，该项不再排队，即一个控制位或其它数据被设置来指示所有页面已经被预取(方框 1809)。如果未达到该极限，则当指针绕回到 SLB 中该项的位置时，该项将被再次选择。

在上述的说明中，已经假定每个 SLB 项包括用于预取的页面的列表。事实上，一些 SLB 项可以不包含列表中的页面或包含比页面的完整列表少的页面。因为该列表长度固定，因此，可以通过页面引用计数为零或一个或一些控制位来指示可预取页面的缺少或比阵列 817 的固定长度少的一些可预取页面。

当一个 ERAT 项响应于预取请求而被装载到 ERAT 中的时候，这个行为可能进一步引起预取在由要预取的 ERAT 项寻址的页面内的各高速缓冲存储器行，如上所述。

在上面的说明中，可以注意到，仅对未使用的预取、即如果数据被预取但在高速缓冲存储器或 ERAT 中从未被使用的情况下，减量行引用计数和页面引用计数。这意味着，如果计数低于预取门限，则它可以从不被减量。结果，它将缓慢地增量直到达到门限值（假定数据在存储器中）。当门限是 1 时，这是在优选实施例中用于高速缓冲存储器行预取的门限，没有关系。但是，它破坏了较高门限的目的，并且可以干扰要预取到 SLB 项中的 ERAT 的页面阵列的正确操作。因此，一种可选的老化机制可以被应用到用于预取 ERAT 项的页面引用计数。这样的机制可能包括例如：当页表项被从存储器向 TLB 装载时或当出现一些其它较不频繁的事件的时候，减量页面计数。这样的老化机制将使得不常使用的数据的页面计数保持近乎为零，即使数据从未被预取。类似的老化机制可用于减量与在页表项内的高速缓冲存储器行相关联的行引用计数。

虽然为了清楚起见，在此示出的各种数据结构、特别是页表和段表具有类似表格的结构，但是事实上这些结构可以更复杂，并且在这些结构中的数据“项”可以被分割，以使一个项的部分在一个位置，而其它部分在另一个位置。

在所述优选实施例中，页表中的引用历史数据包括简单的升或降计数器，它们在一行被装载和使用时增量，而在装载但不使用时减量。但是，可以明白，这样的计数器可以对不同的事件进行增量或减量，并且可以对于不同的事件增量或减量不同的数量。而且，页面的每个可高速缓存子单元的引用历史数据不必是升或降计数器。它可以简单为单个位，该位依赖于一些近期事件而被设置为 1 或 0。例如，如果最近对高速缓冲存储器的装载是成功的（即数据被使用），则该位可以被置为 1，否则被置为 0。它还可以是一组位，其中每位对应于一些不同的近期事件。

在所述优选实施例中，如果可高速缓存子单元各自的计数器大于 0，则它们被预取。但是，预取决定可以更复杂，并且以其它的方式来考虑计数器的值。

例如，如果计数器超过低门限则处理器可以向一层高速缓冲存储器预取，如果计数器超过较高的门限，则处理器可以向较高层高速缓冲存储器预取。而且，门限可以根据系统的当前负载和其它因素而变化。

在所述优选实施例中，计算机系统使用三级寻址方案，其中当存取高速缓冲存储器或存储器时，有效地址被翻译为虚拟地址，虚拟地址继而又被翻译为实际地址。但是，本发明并不限于任何特定的寻址方案，并且可以应用到具有少于三级寻址或多于三级寻址的系统，而且，按照本发明的计算机系统可以具有不同形式的地址翻译结构，尤其是不必具有 ERAT。地址翻译结构与在特定结构中使用的寻址级有关。

在所述优选实施例中，当一个页面项被装载到 ERAT 时该页面项被装载到预取缓冲器。但是，将适当的项装载到预取缓冲器或类似的结构以启动预取也可以响应于任何数量的其它事件。当计算机系统没有 ERAT 时尤其是这样。例如，当页面项被装载到 TLB 中或一些其它的或替代的寻址翻译结构中时，该项可以被装载到预取缓冲器中。

虽然在此所述的升或降计数器最好被保持用于少于全页的可高速缓存子单元，但是也可以使用单个页面引用计数器来进行对于整个页面的预取决定。这样的技术对较低层的高速缓冲存储器（例如第三级高速缓冲存储器）更为有益。

虽然已经公开了本发明的具体实施例以及一些替代实施例，本领域的技术人员可以认识到，可以在本发明的范围内进行各种形式和细节上的附加改变。

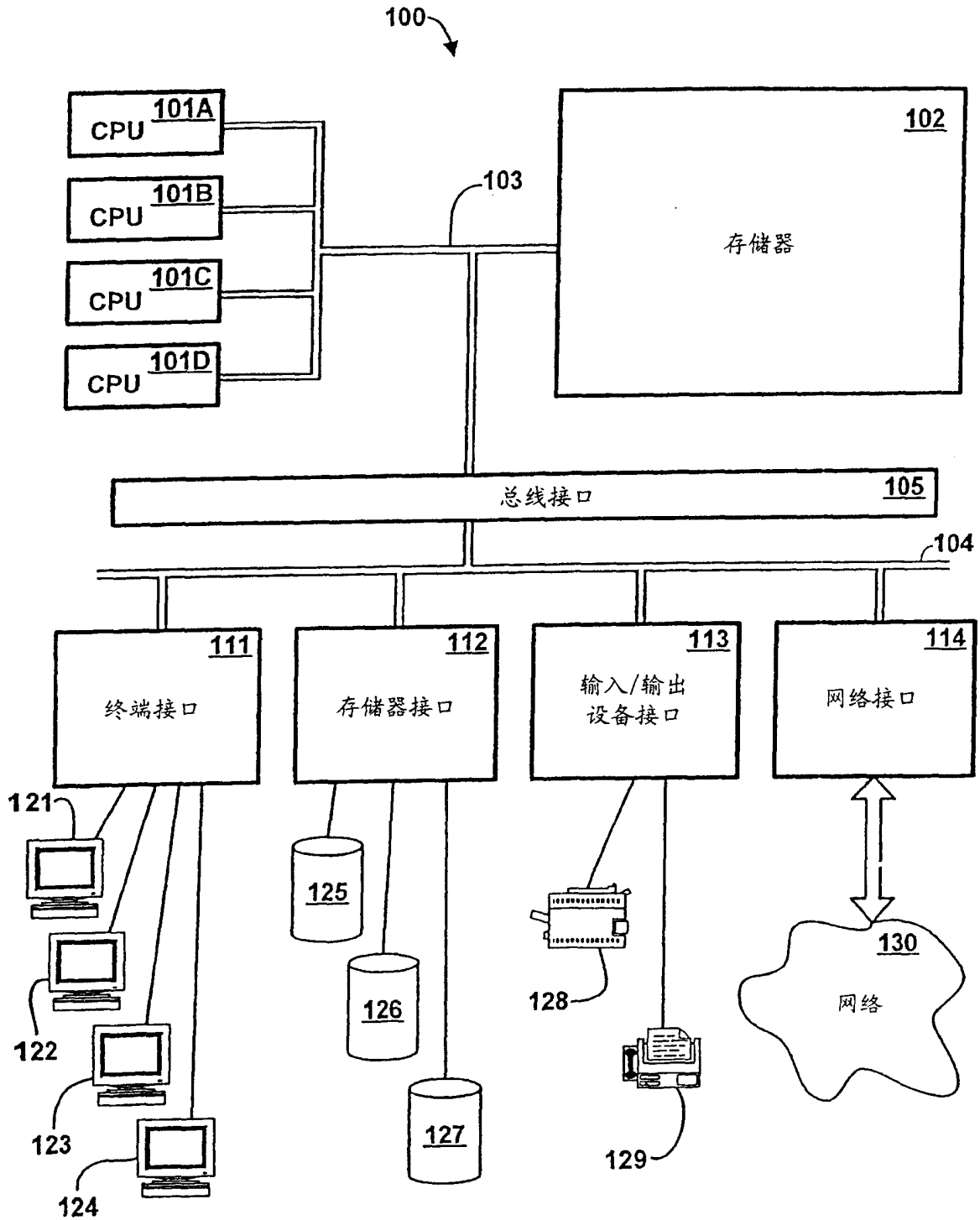


图 1

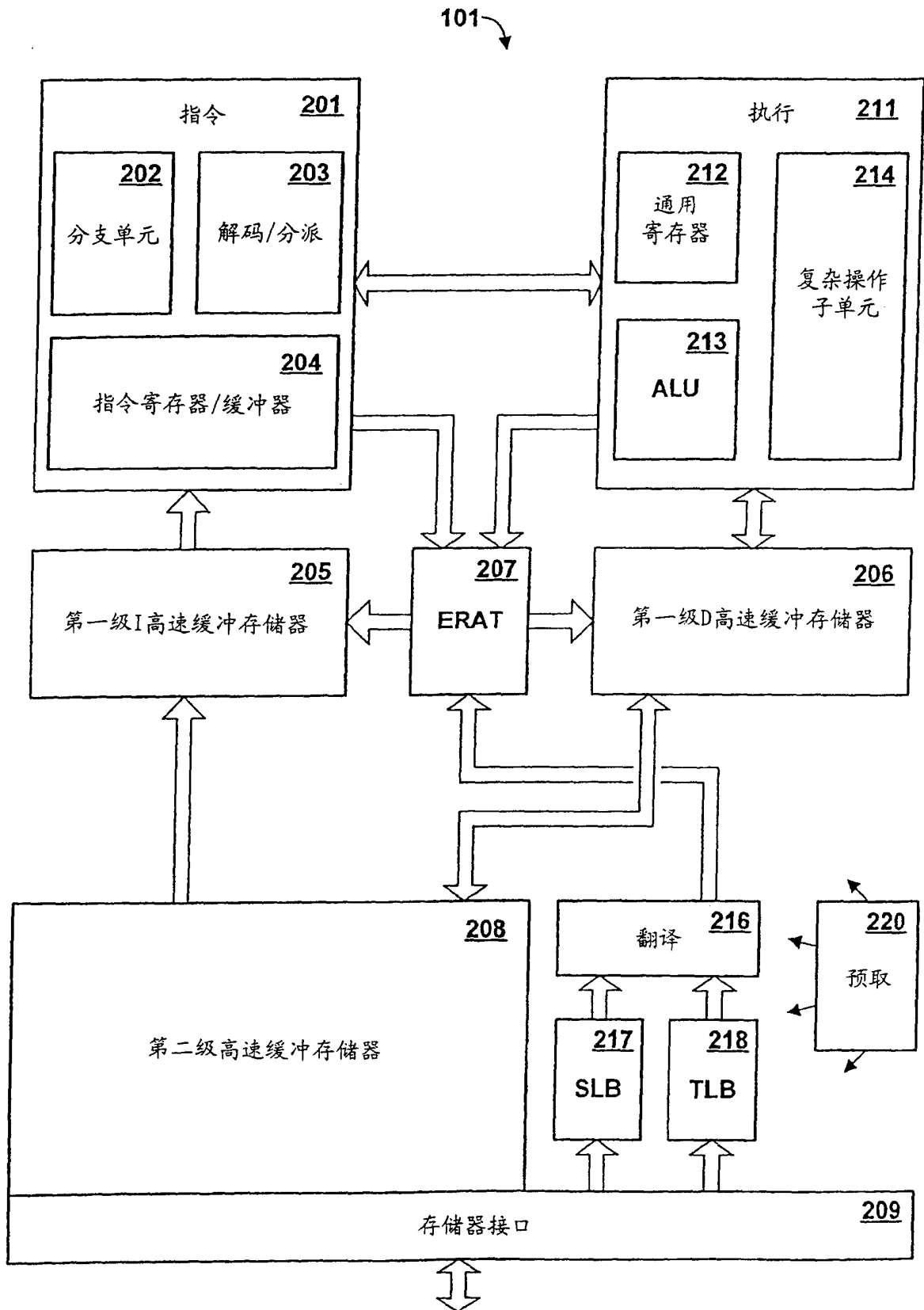


图 2

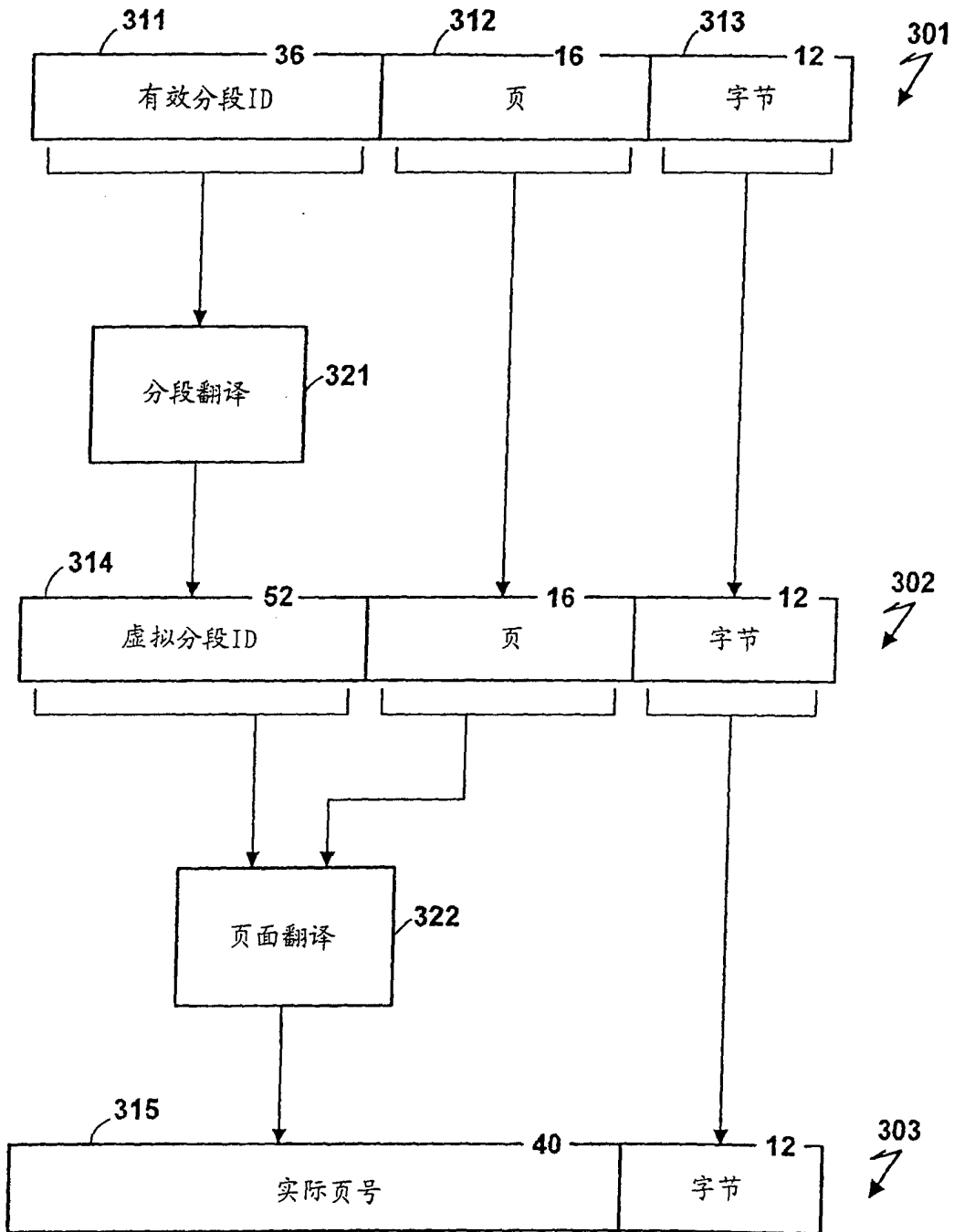


图 3

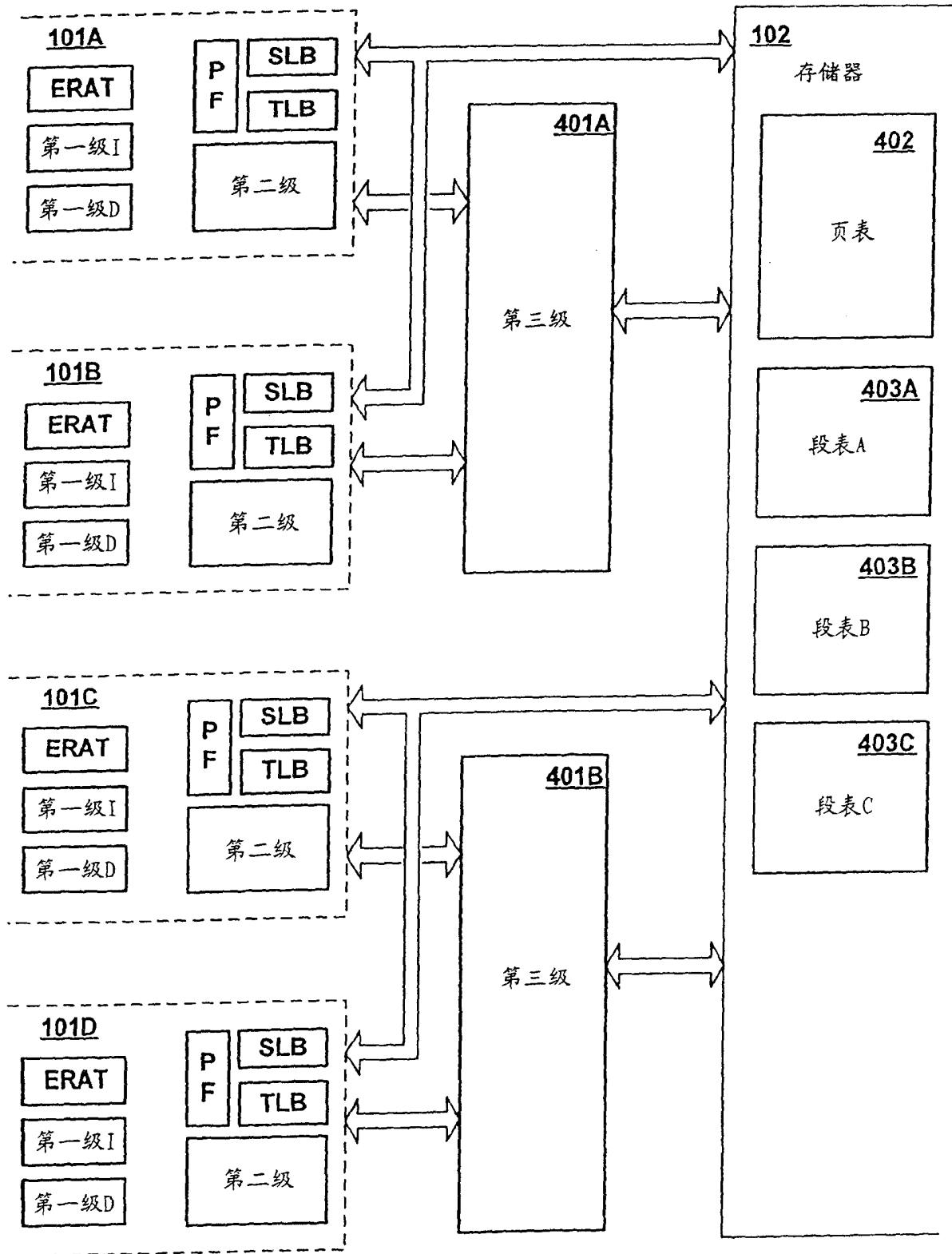


图 4

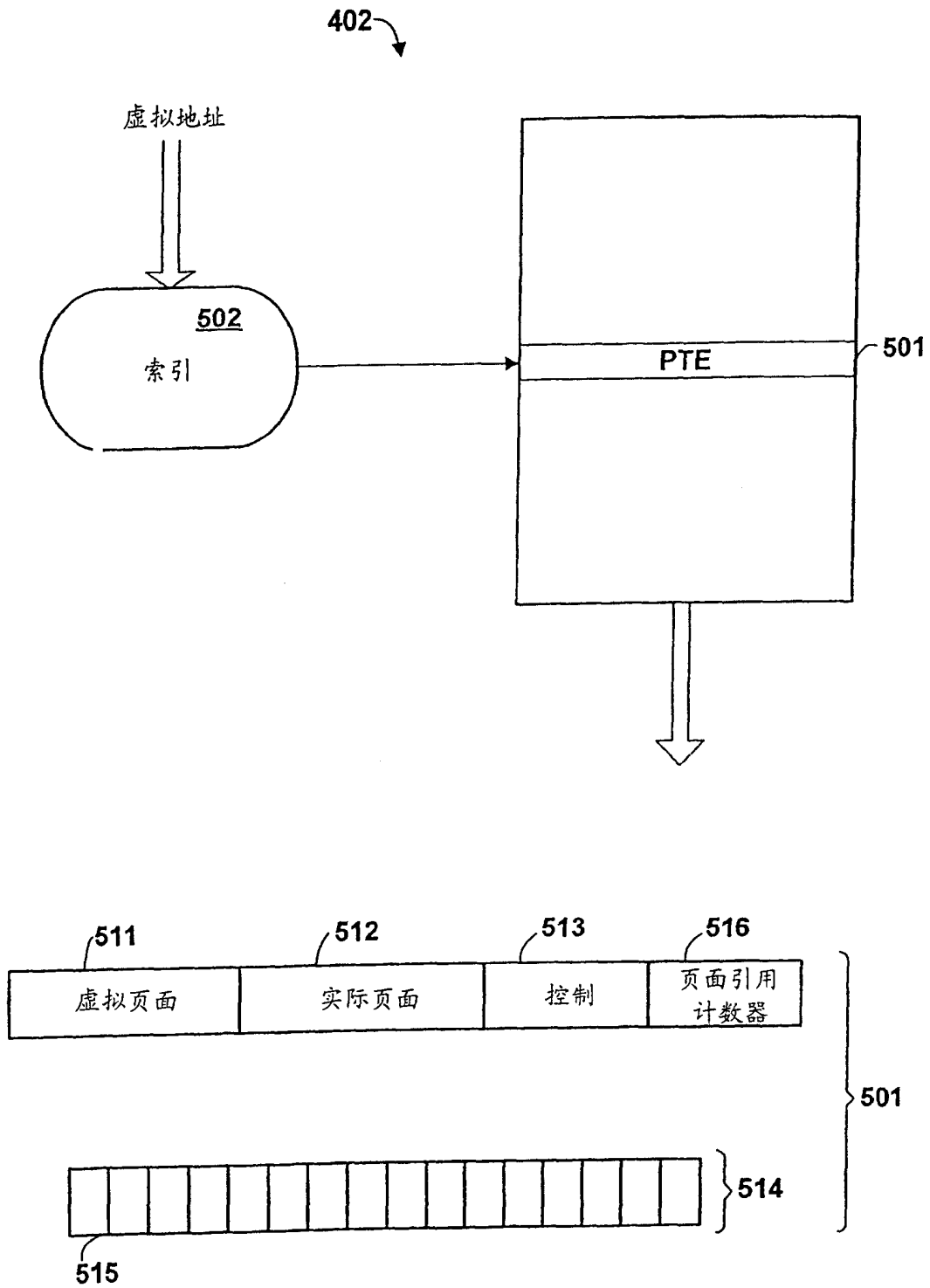


图 5

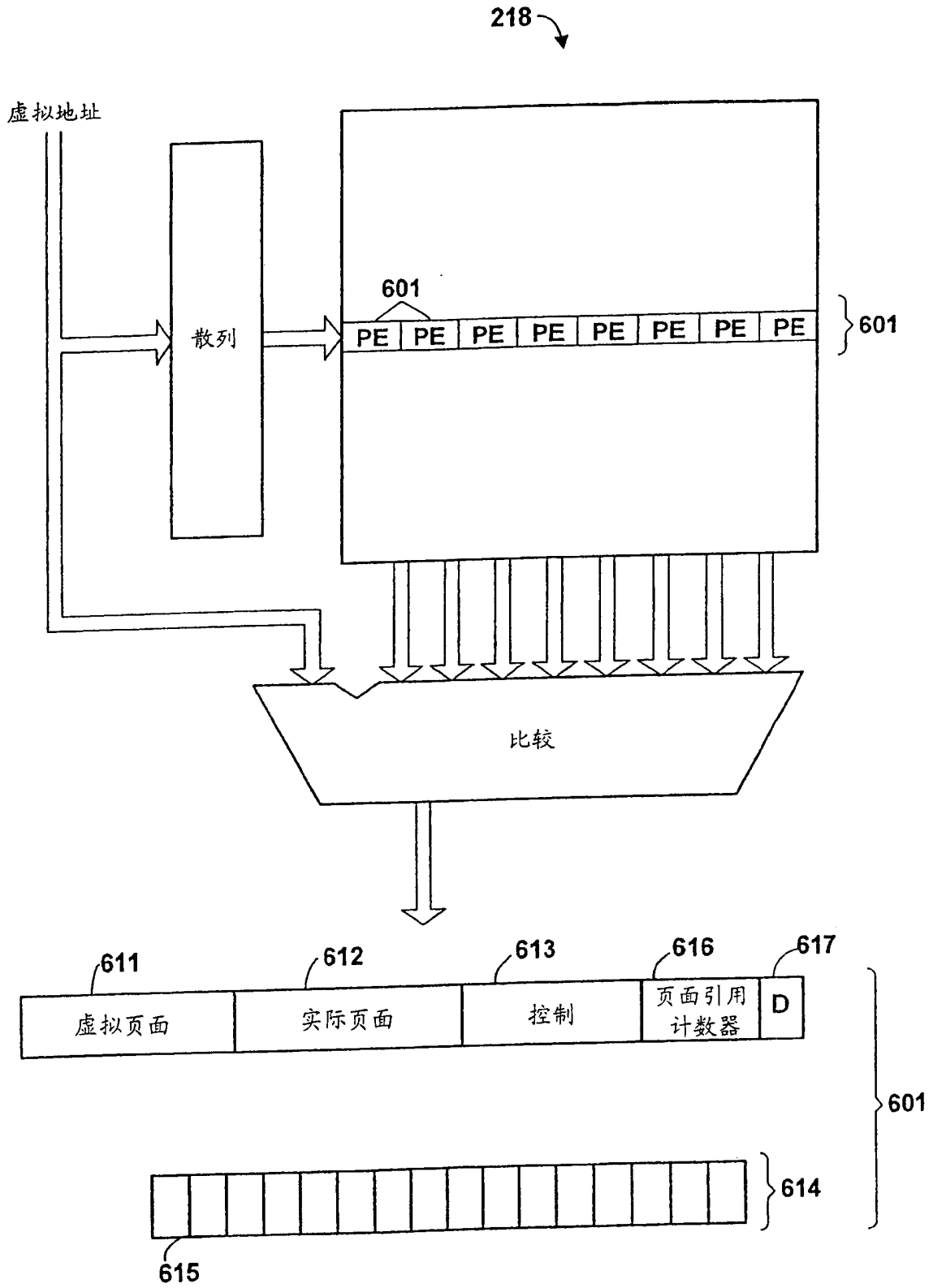


图 6

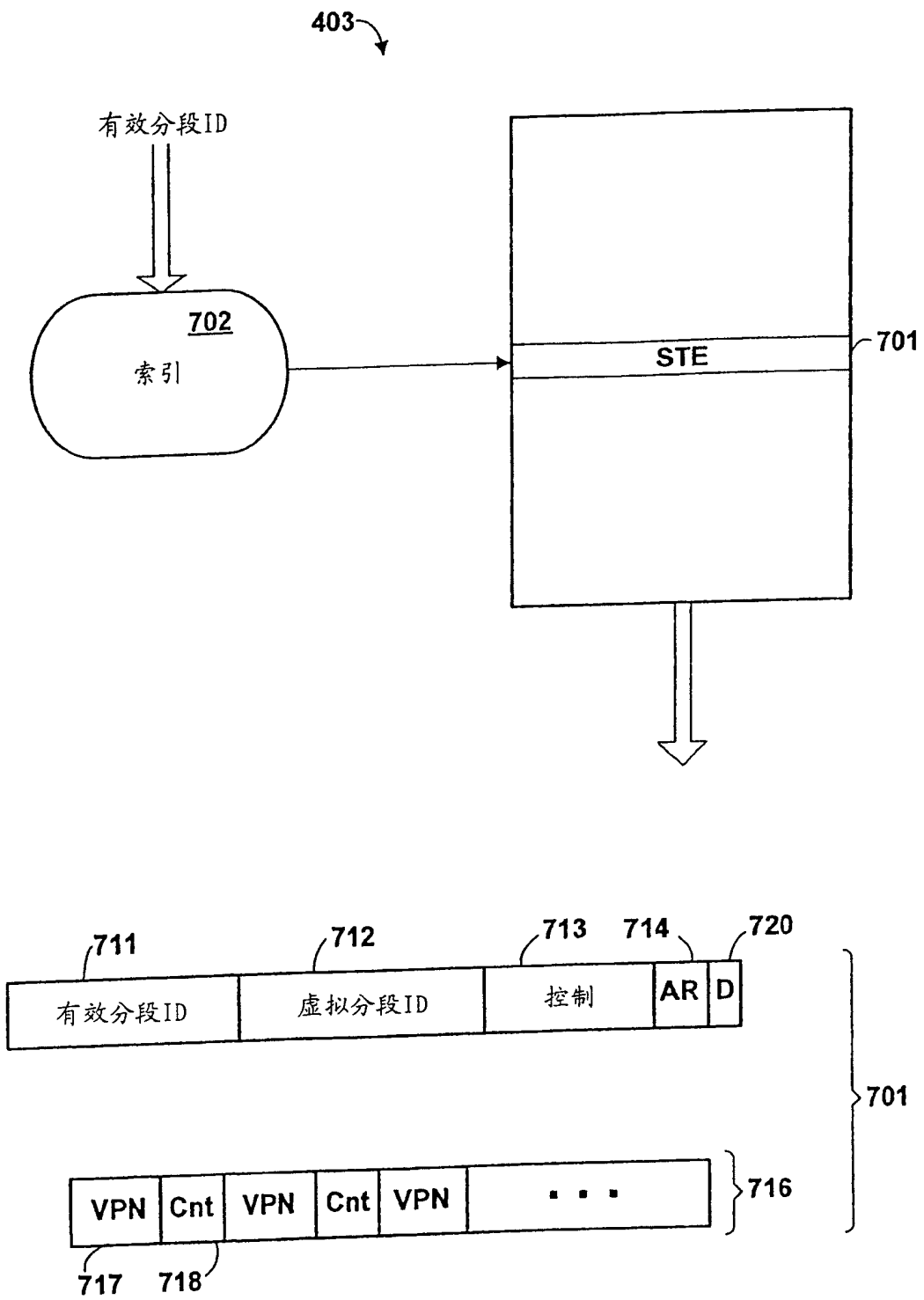


图 7

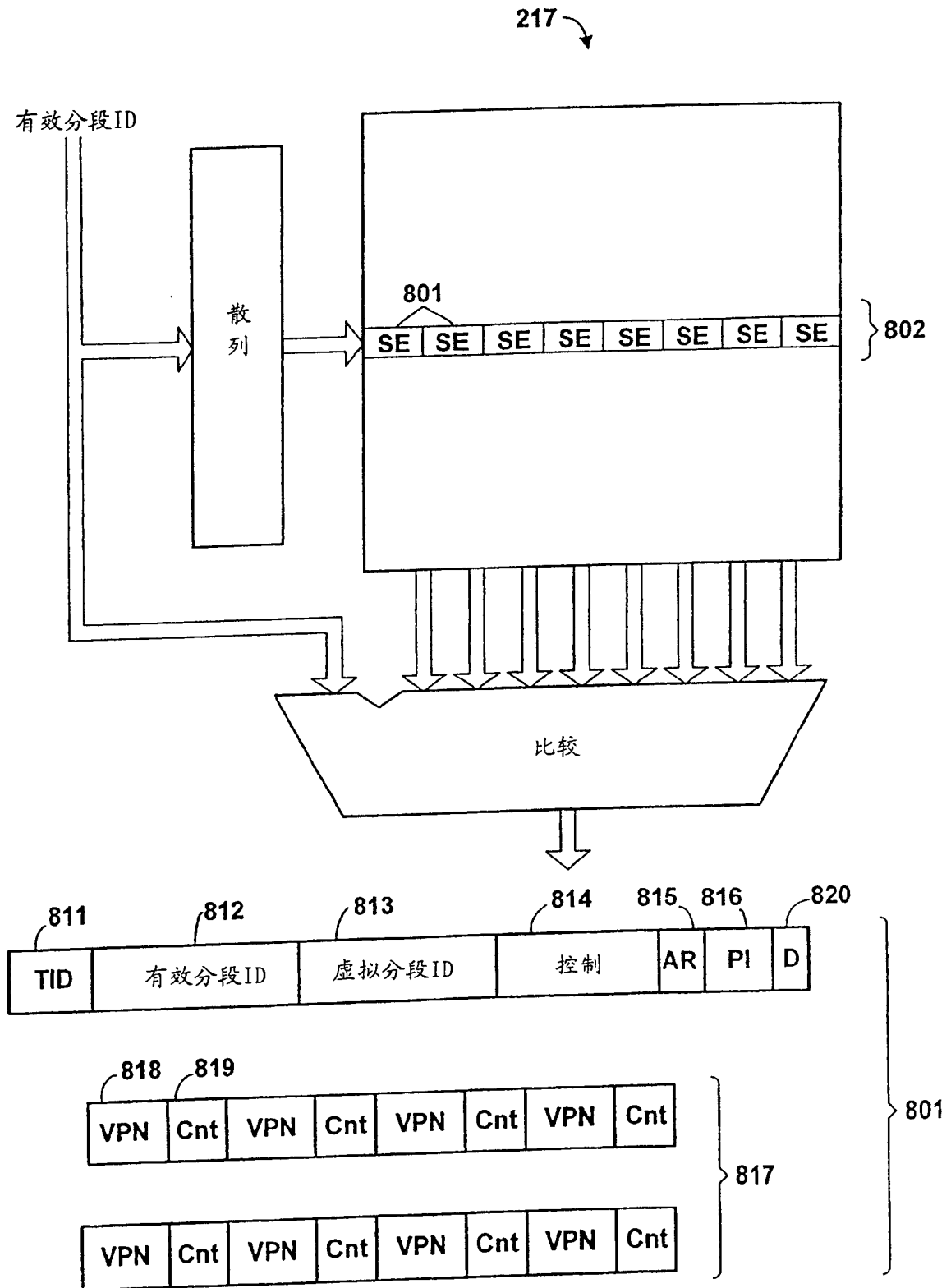


图 8

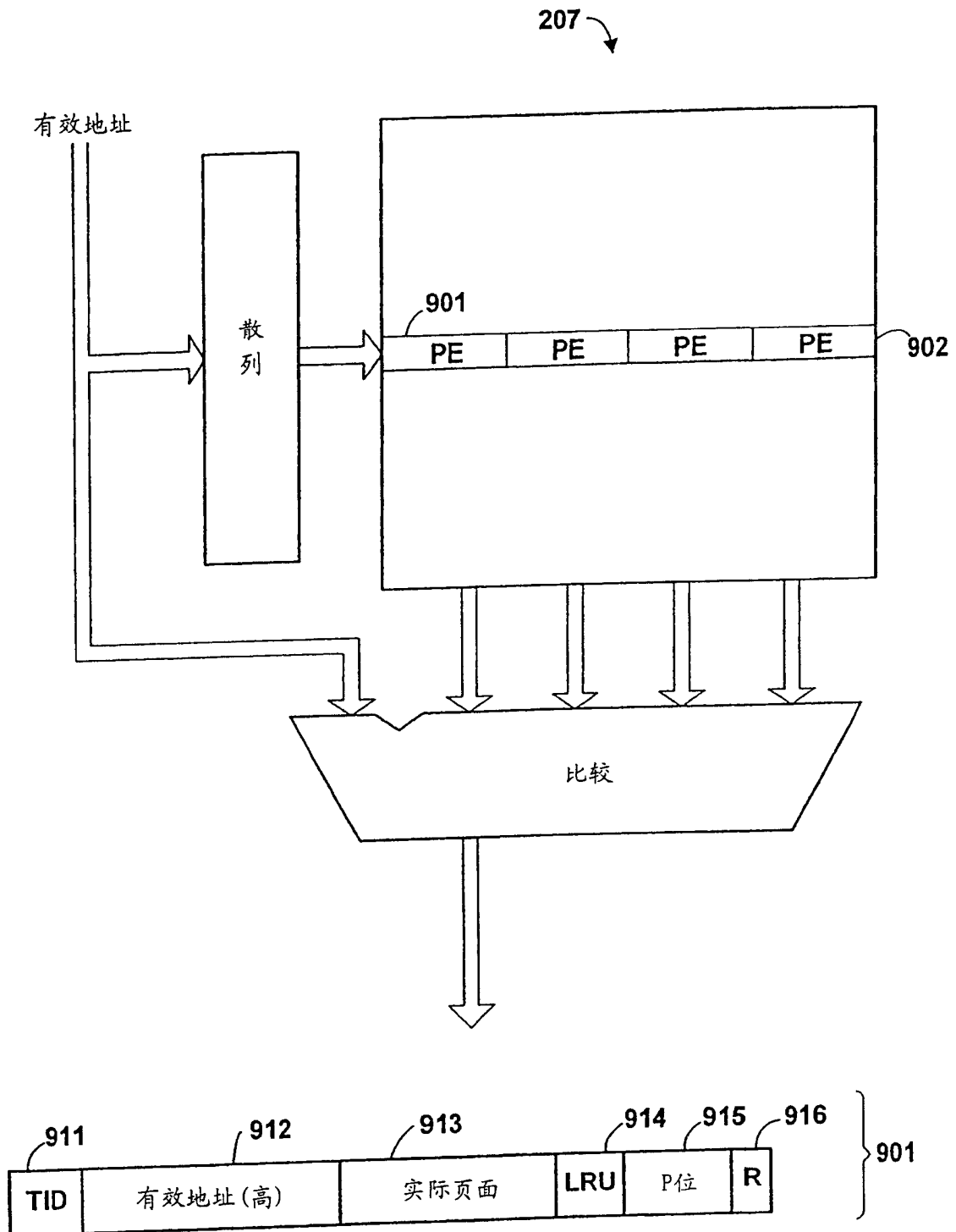


图 9

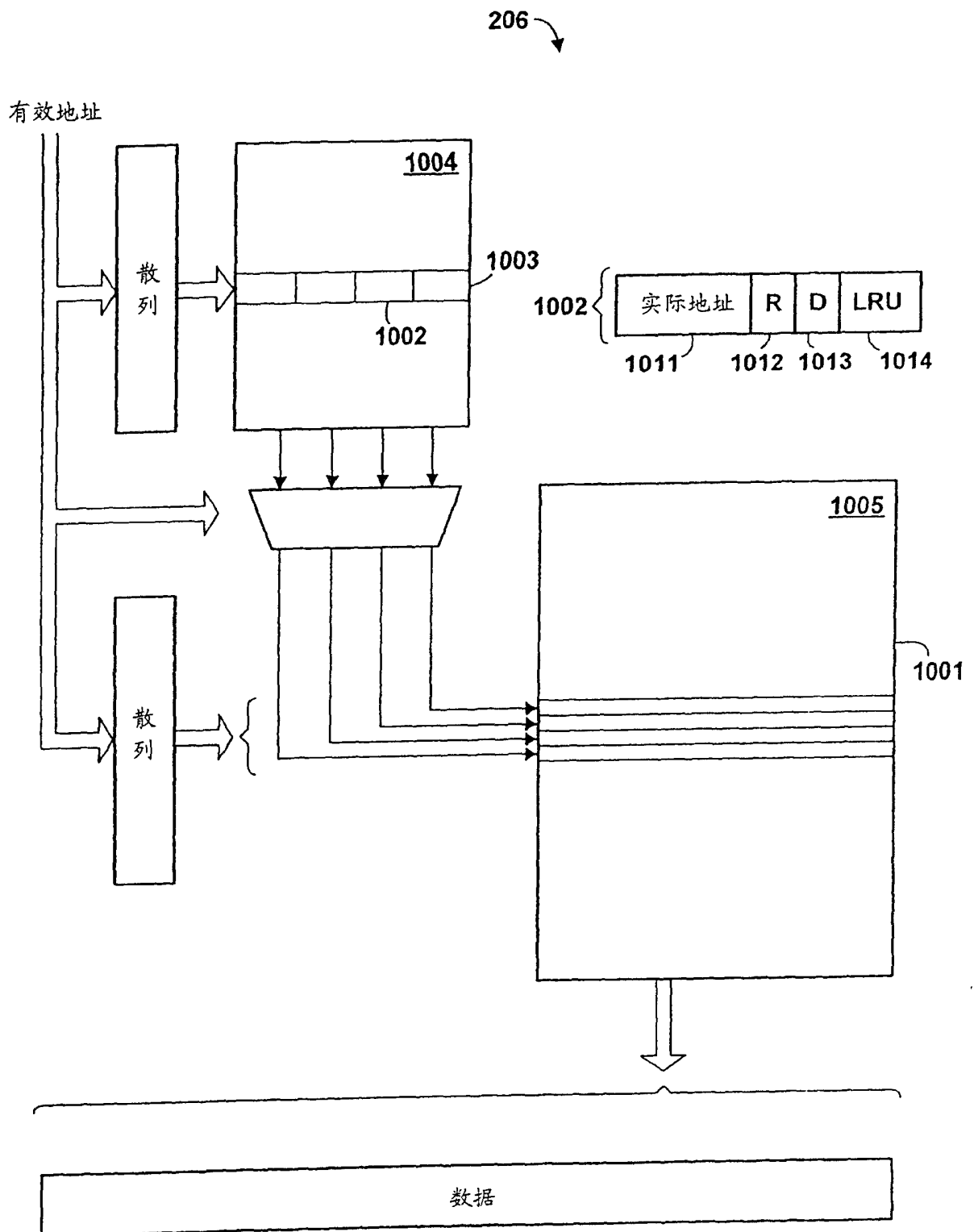


图 10

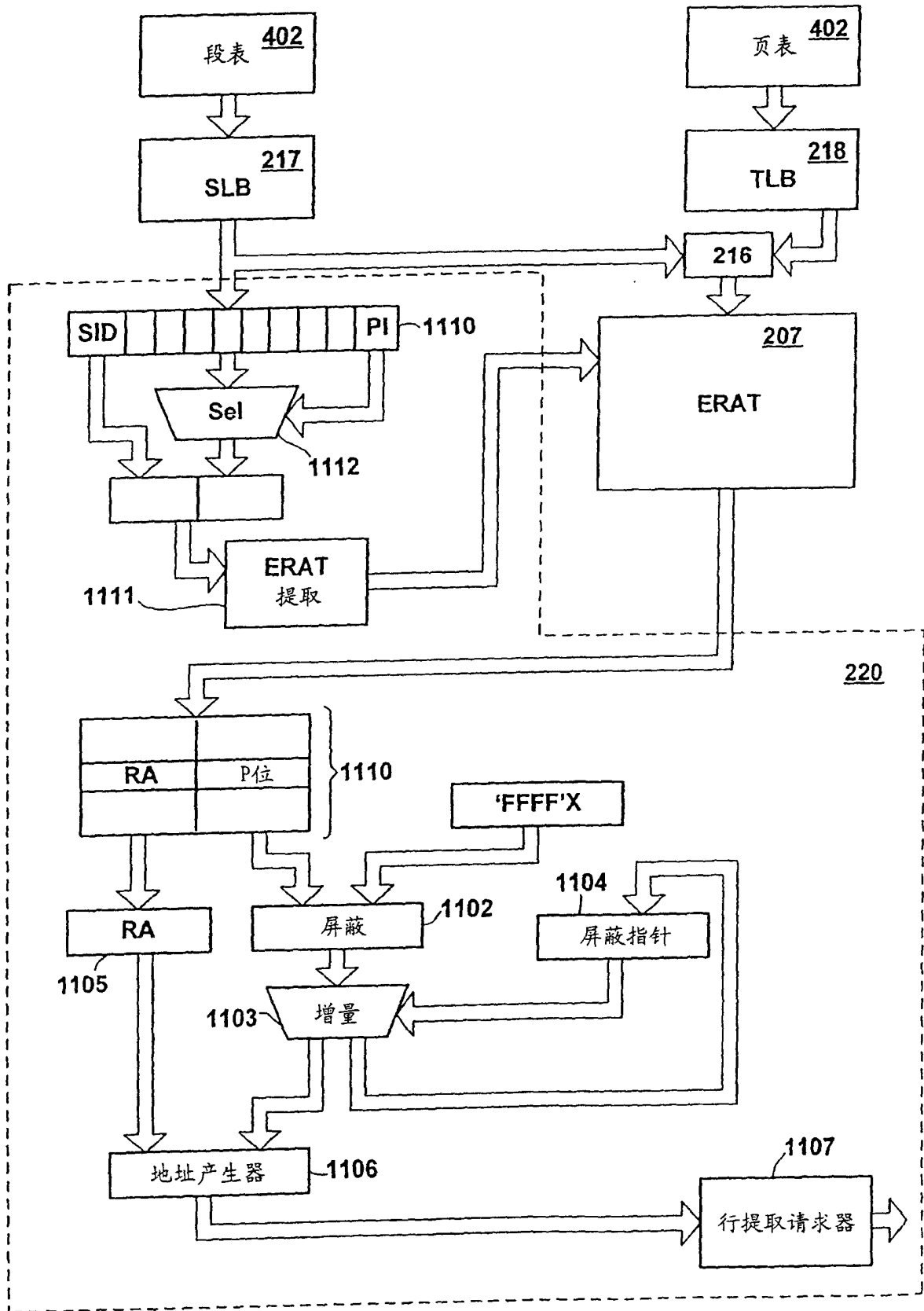


图 11

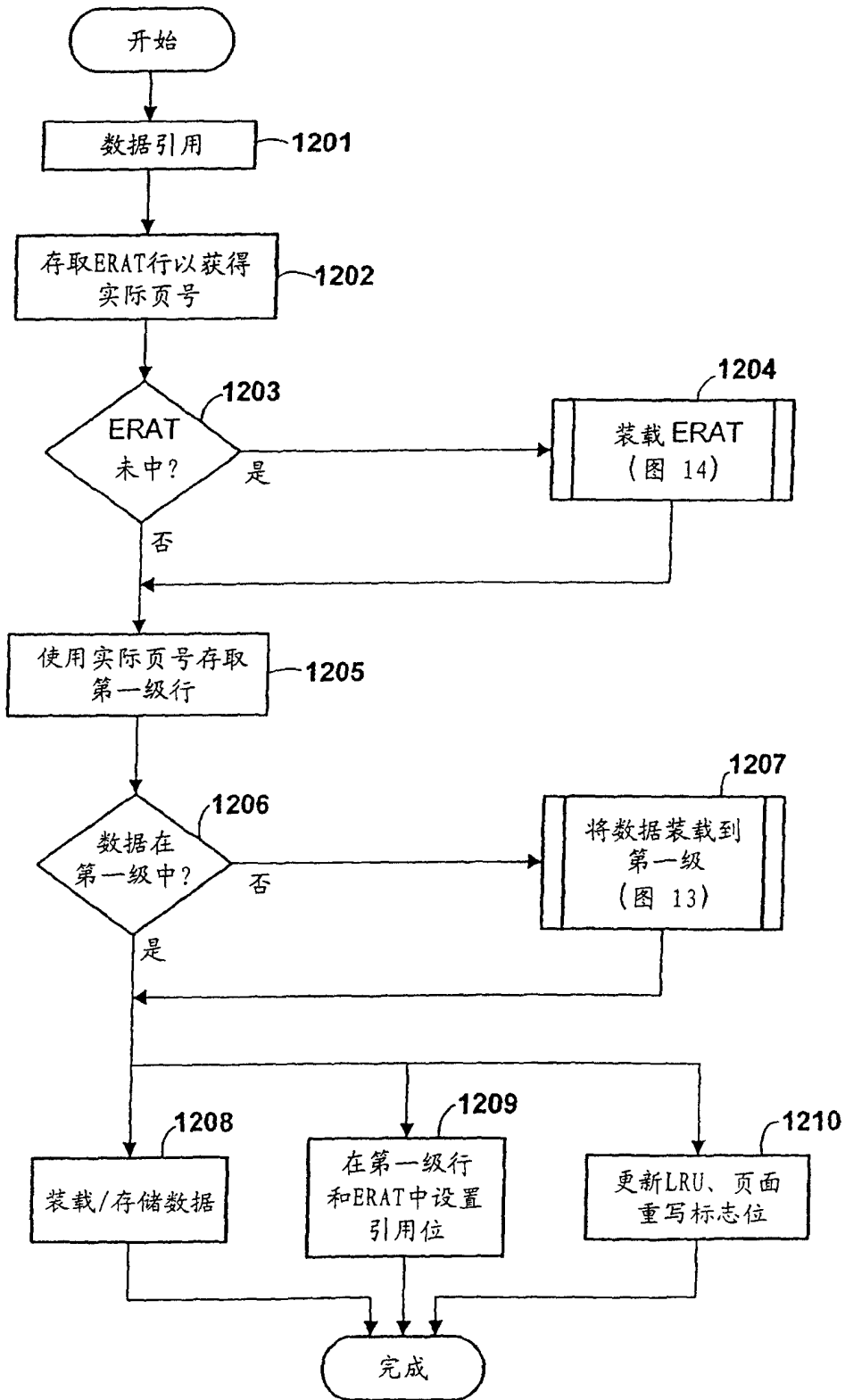


图 12

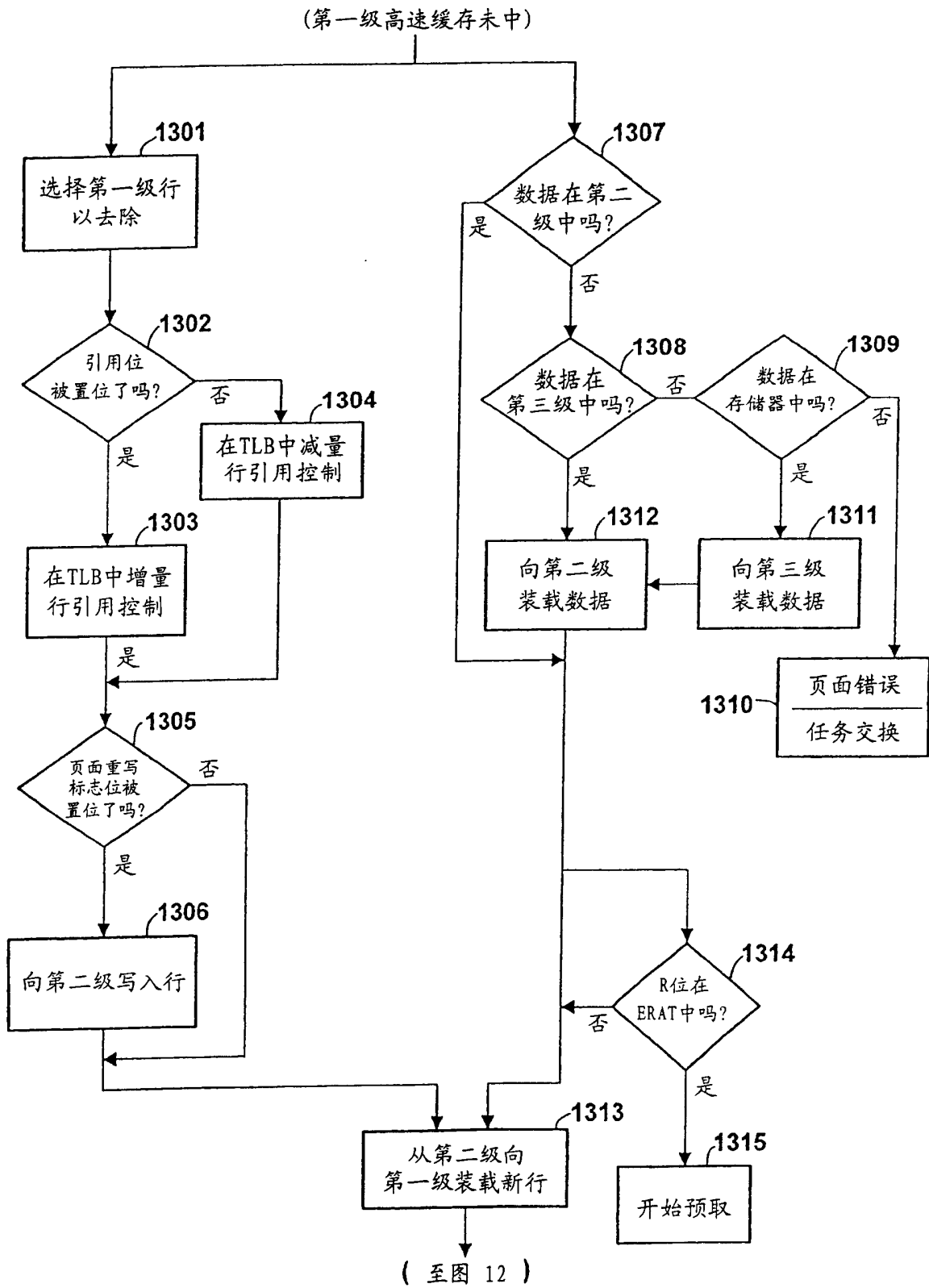


图 13

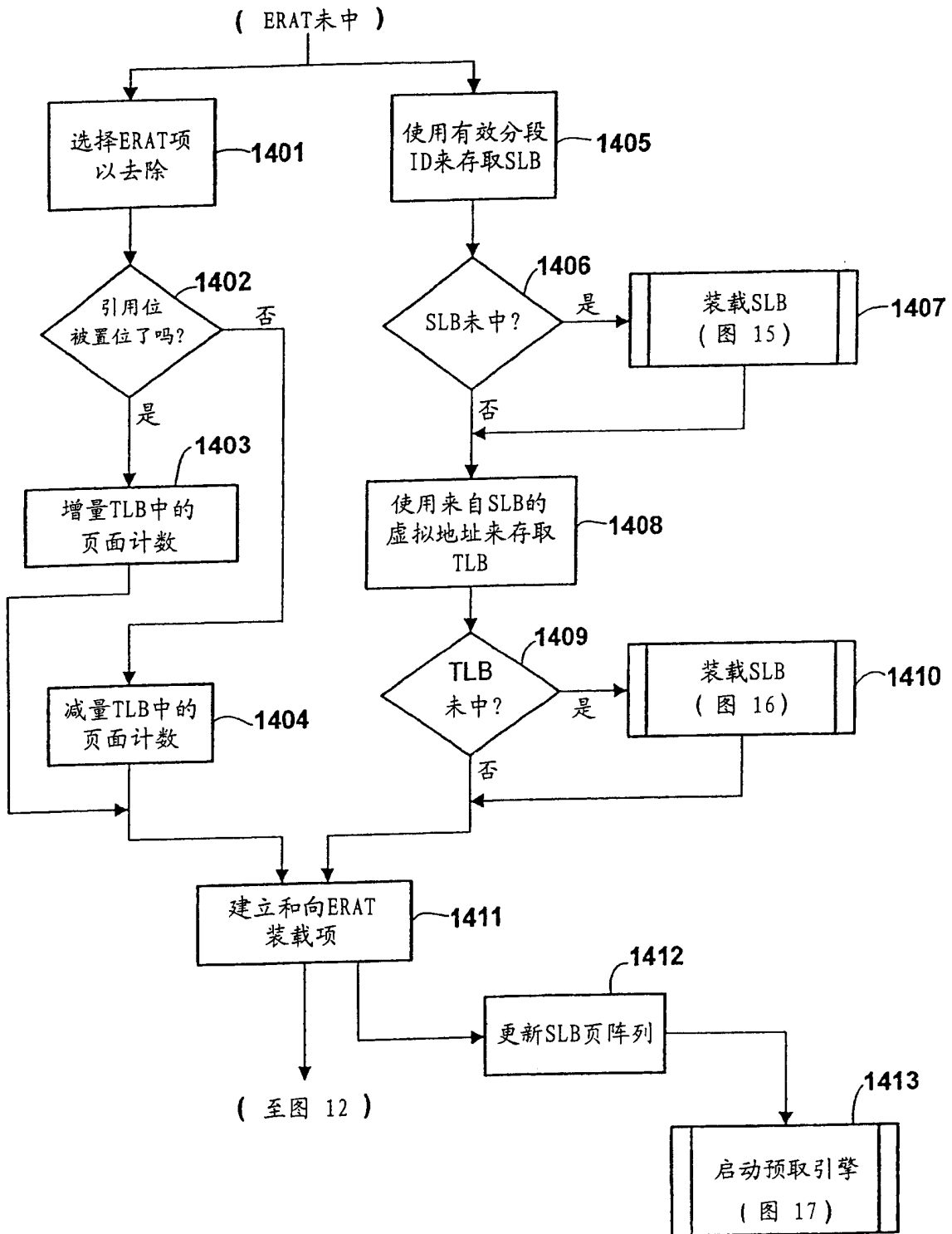


图 14

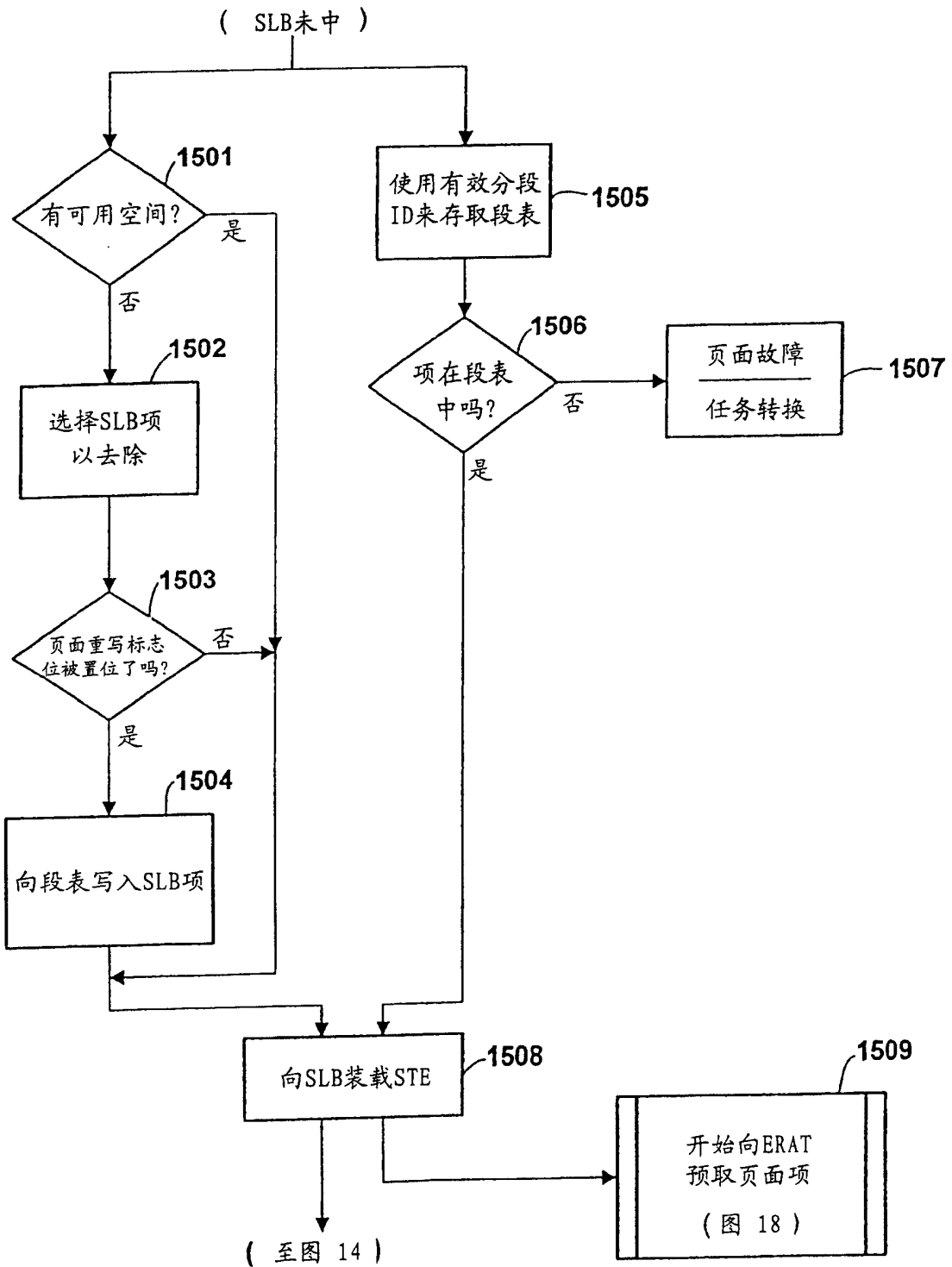


图 15

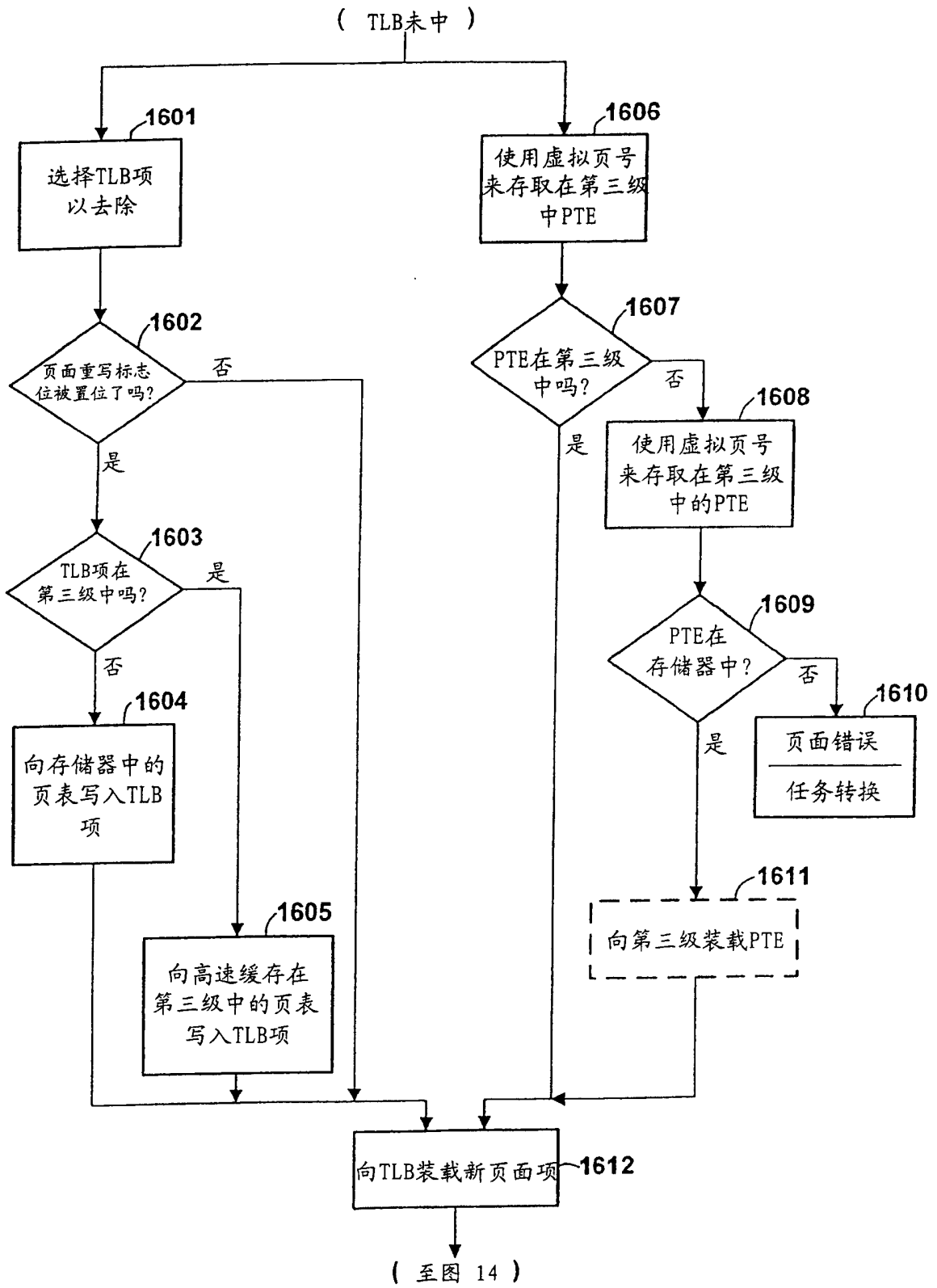


图 16

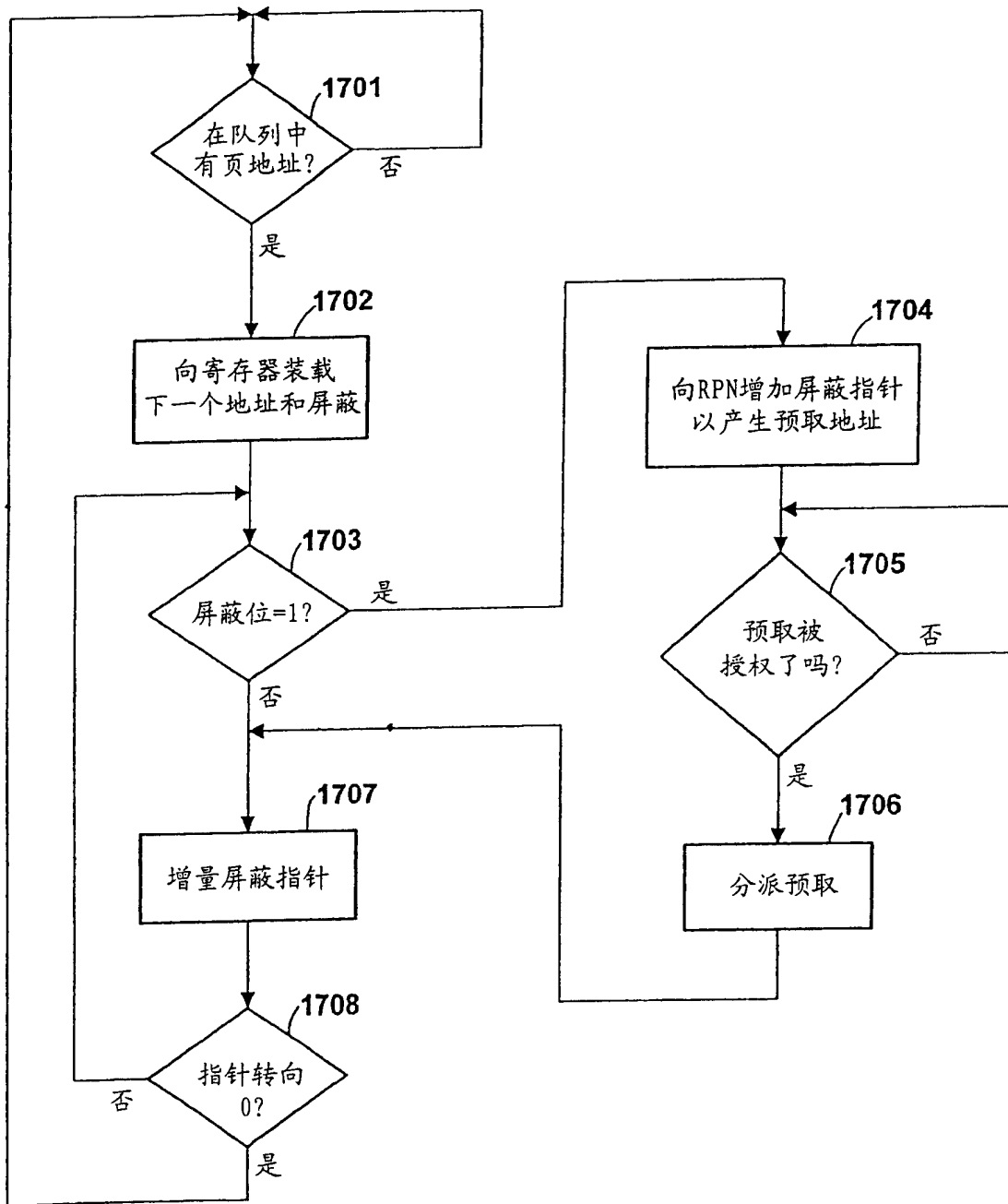


图 17

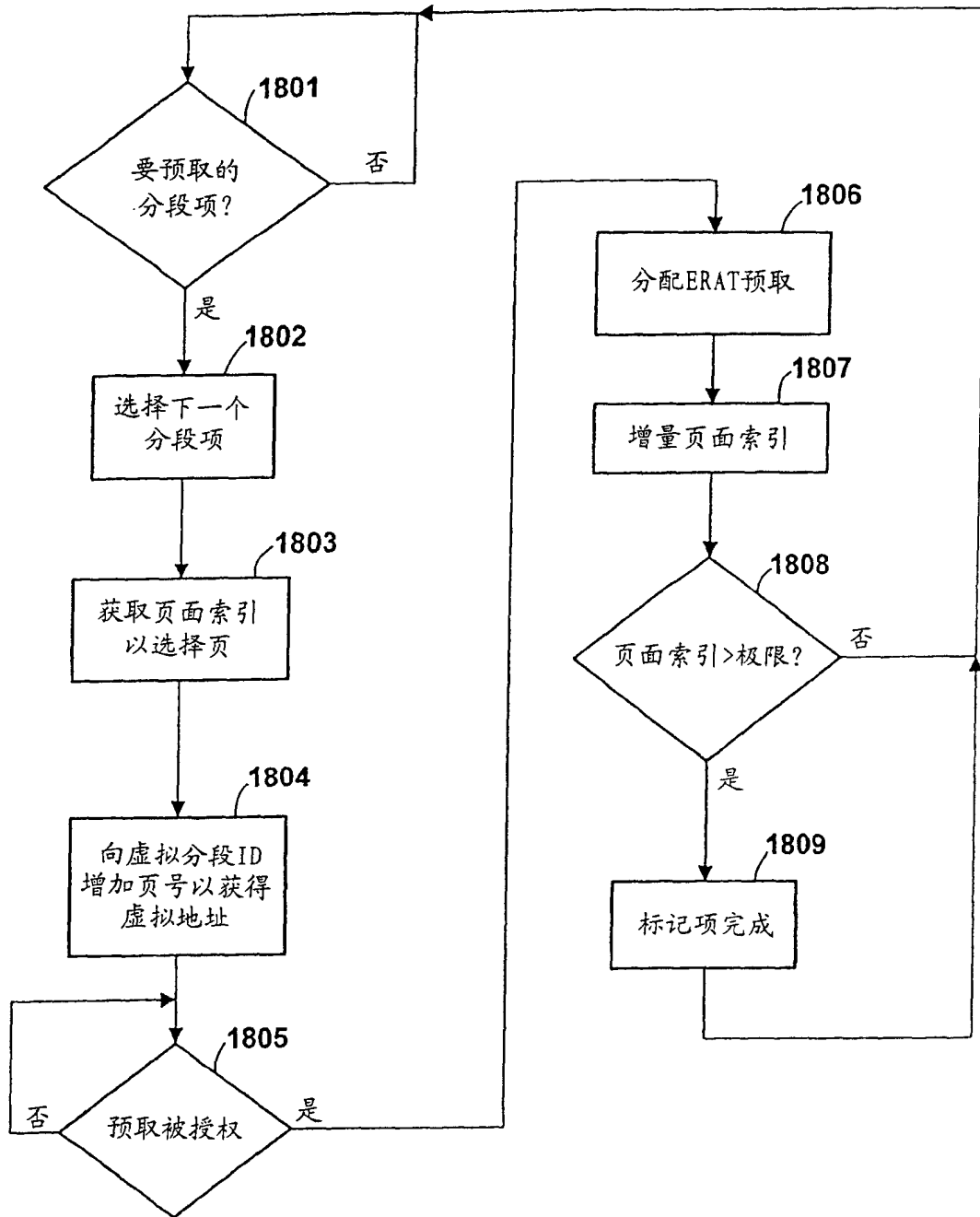


图 18