

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 12/10 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200610121834.3

[43] 公开日 2007年1月31日

[11] 公开号 CN 1904860A

[22] 申请日 2004.5.6

[21] 申请号 200610121834.3

分案原申请号 200480011894.9

[30] 优先权

[32] 2003.5.12 [33] US [31] 10/435,919

[71] 申请人 国际商业机器公司

地址 美国纽约

[72] 发明人 T·J·施莱格尔 L·海勒

E·普费弗 K·普兰贝克

[74] 专利代理机构 北京市中咨律师事务所
代理人 于静 李峥

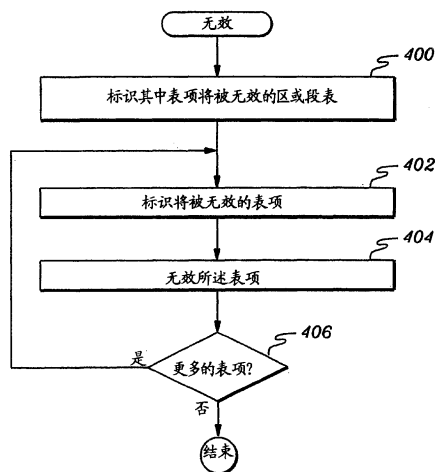
权利要求书 11 页 说明书 22 页 附图 10 页

[54] 发明名称

无效存储器，清除缓冲区表项

[57] 摘要

选定的存储单元(例如存储段或存储区)被无效。通过设置位于与将被无效的存储单元对应的数据结构表项中的无效指示符来有助于所述无效。此外，与所述被无效的存储单元或其他选定存储单元关联的缓冲区表项被清除。提供了指令以执行所述无效和/或清除。此外，与特定地址空间关联的缓冲区表项被清除而不进行任何无效。这也由所述指令来执行。所述指令可以在软件、硬件、固件或它们的某些组合中实现，或者其可以被仿真。



1. 一种用于在计算机系统中清除地址转换缓冲区表项的方法，所述计算机系统包括一个或多个地址转换表，所述地址转换表提供转换信息以将虚拟地址转换为存储器地址，每个地址转换缓冲区表项都动态地缓存了地址转换信息，所述方法包括以下步骤：

从要执行的机器可执行指令的操作码判定所述指令被配置为开始执行清除操作；

执行所述指令，所述执行步骤包括以下步骤：

根据第一转换表的第一转换表起始地址，从所述地址转换缓冲区选择性地清除所述地址转换缓冲区的一个或多个第一表项，所述一个或多个第一表项与所述第一转换表起始地址关联。

2. 如权利要求1中所述的方法，其中所述清除步骤不会清除所述地址转换缓冲区中所述地址转换缓冲区的一个或多个第二表项，所述一个或多个第二表项没有与所述第一转换表起始地址关联。

3. 如权利要求1中所述的方法，其中所述第一转换表起始地址包括地址空间控制元素（ASCE）地址、区表起始地址或段表起始地址中的任何一个地址。

4. 如权利要求1中所述的方法，其中所述第一转换表起始地址包括一系列两个或更多转换表表项的第一转换表表项。

5. 如权利要求3中所述的方法，还包括无效所述系列的两个或更多转换表表项的步骤。

6. 如权利要求1中所述的方法，其中所述计算机系统包括架构的通用寄存器，其中所述地址转换表包括页表和以下各项中的任何一项：

一个或多个段表，

一个或多个段表和一个或多个第一区表，

一个或多个段表和一个或多个第一区表以及一个或多个第二区表或

一个或多个段表和一个或多个第一区表和一个或多个第二区表以及一

个或多个第三区表;

其中段表表项还包括页表起始地址;

所述方法还包括以下步骤:

从所述指令指定的位置检索第一信息, 所述第一信息包括对所述指令的操作的指示; 以及

根据所述第一信息, 判定所述指令被配置为开始执行所述清除操作。

7. 如权利要求6中所述的方法, 其中所述第一信息指示当所述第一信息的位52是1时, 所述指令被配置为开始执行清除操作。

8. 如权利要求1中所述的方法, 还包括以下步骤:

所述计算机系统从主存储器取出所述指令, 所述指令包括:

用于定位第二信息的第二字段, 所述第二信息包括所述清除操作的清除信息;

从所述取出的指令的所述操作码判定所述取出的指令是具有所述清除操作的指令;

i) 判定地址转换缓冲区的地址转换缓冲区表项的预定地址转换缓冲区表项部分是否包括所述第二信息; 以及

j) 清除判定为包括所述第二信息的所述地址转换缓冲区表项; 以及

k) 为多个地址转换缓冲区表项重复步骤i) - j)。

9. 如权利要求8中所述的方法, 其中所述第二字段在第二架构的通用寄存器中定位所述第二信息。

10. 如权利要求8中所述的方法, 其中当所述第二字段是0时:

清除任何地址转换缓冲区表项, 或者

所述第二信息包括所述第一转换表起始地址。

11. 如权利要求8中所述的方法, 还包括以下步骤:

当所述第二字段是0时, 从所述地址空间控制元素形成所述第二信息;

以及

当所述第二字段大于0时, 从架构的通用寄存器检索第二信息。

12. 如权利要求1中所述的方法, 还包括以下步骤:

解释所述指令以标识用于仿真所述指令的操作的预定软件例程，所述预定软件例程包括多个指令；以及
执行所述预定软件例程。

13. 如权利要求 1 中所述的方法，其中所述地址转换缓冲区是转换旁视缓冲区（TLB）。

14. 如权利要求 1 中所述的方法，其中所述地址转换缓冲区是以下缓冲区中的任何一个缓冲区：

与所述计算机系统的第一中央处理单元关联的地址转换缓冲区，所述第一中央处理单元执行所述指令，

与所述计算机系统的第二中央处理单元关联的地址转换缓冲区，或
与所述计算机系统关联的所有地址转换缓冲区。

15. 如权利要求 1 中所述的方法，其中所述计算机系统还包括页表，所述页表具有实主存储器地址，其中实主存储器地址包括绝对地址、实地址或物理地址中的任何一个地址。

16. 一种用于在计算机系统中清除地址转换缓冲区的系统，所述系统包括：

主存储器；

一个或多个与所述主存储器通信的中央处理单元，所述中央处理单元包括：

架构的通用寄存器；

地址转换缓冲区，地址转换缓冲区具有保存地址转换信息的地址转换缓冲区表项；

多个地址转换表，所述地址转换表用于将所述一个或多个处理单元中的一个中央处理单元的虚拟主存储器地址转换成所述计算机系统的主存储器地址；

一个或多个指令执行单元，所述指令执行单元与所述架构的通用寄存器通信，所述指令执行单元执行从所述主存储器取出的机器指令；

其中所述计算机系统包括指令以执行包括以下步骤的方法：

从要执行的机器可执行指令的操作码判定所述指令被配置为开始执行清除操作;

执行所述指令, 所述执行步骤包括以下步骤:

根据第一转换表的第一转换表起始地址, 从所述地址转换缓冲区选择性地清除所述地址转换缓冲区的一个或多个第一表项, 所述一个或多个第一表项与所述第一转换表起始地址关联。

17. 如权利要求 16 中所述的系统, 其中所述清除步骤不会清除所述地址转换缓冲区中所述地址转换缓冲区的一个或多个第二表项, 所述一个或多个第二表项没有与所述第一转换表起始地址关联。

18. 如权利要求 16 中所述的系统, 其中所述第一转换表起始地址包括地址空间控制元素 (ASCE) 地址、区表起始地址或段表起始地址中的任何一个地址。

19. 如权利要求 16 中所述的系统, 其中所述第一转换表起始地址包括一系列两个或更多转换表表项的第一转换表表项。

20. 如权利要求 19 中所述的系统, 还包括无效所述系列的两个或更多转换表表项的步骤。

21. 如权利要求 16 中所述的系统, 其中所述计算机系统包括架构的通用寄存器, 其中所述地址转换表包括页表和以下各项中的任何一项:

一个或多个段表,

一个或多个段表和一个或多个第一区表,

一个或多个段表和一个或多个第一区表以及一个或多个第二区表或

一个或多个段表和一个或多个第一区表和一个或多个第二区表以及一个或多个第三区表;

其中段表表项还包括页表起始地址;

所述系统还包括以下步骤:

从所述指令指定的位置检索第一信息, 所述第一信息包括对所述指令的操作的指示; 以及

根据所述第一信息, 判定所述指令被配置为开始执行所述清除操作

22. 如权利要求 21 中所述的系统, 其中所述第一信息指示当所述第一信息的位 52 是 1 时, 所述指令被配置为开始执行清除操作。

23. 如权利要求 16 中所述的系统, 还包括以下步骤:

所述计算机系统从主存储器取出所述指令, 所述指令包括:

用于定位第二信息的第二字段, 所述第二信息包括所述清除操作的清除信息;

从所述取出的指令的所述操作码判定所述取出的指令是具有所述清除操作的指令;

i) 判定地址转换缓冲区的地址转换缓冲区表项的预定地址转换缓冲区表项部分是否包括所述第二信息; 以及

j) 清除判定为包括所述第二信息的所述地址转换缓冲区表项; 以及

k) 为多个地址转换缓冲区表项重复步骤 i) - j)。

24. 如权利要求 23 中所述的系统, 其中所述第二字段在第二架构的通用寄存器中定位所述第二信息。

25. 如权利要求 23 中所述的系统, 其中当所述第二字段是 0 时:

清除任何地址转换缓冲区表项, 或者

所述第二信息包括所述第一转换表起始地址。

26. 如权利要求 23 中所述的系统, 还包括以下步骤:

当所述第二字段是 0 时, 从所述地址空间控制元素形成所述第二信息; 以及

当所述第二字段大于 0 时, 从架构的通用寄存器检索第二信息。

27. 如权利要求 16 中所述的系统, 还包括以下步骤:

解释所述指令以标识用于仿真所述指令的操作的预定软件例程, 所述预定软件例程包括多个指令; 以及

执行所述预定软件例程。

28. 如权利要求 16 中所述的系统, 其中所述地址转换缓冲区是转换旁视缓冲区 (TLB)。

29. 如权利要求 16 中所述的系统, 其中所述地址转换缓冲区是以下

缓冲区中的任何一个缓冲区:

与所述计算机系统的第一中央处理单元关联的地址转换缓冲区, 所述第一中央处理单元执行所述指令,

与所述计算机系统的第二中央处理单元关联的地址转换缓冲区, 或与所述计算机系统关联的所有地址转换缓冲区。

30. 如权利要求 16 中所述的系统, 其中所述计算机系统还包括页表, 所述页表具有实主存储器地址, 其中实主存储器地址包括绝对地址、实地址或物理地址中的任何一个地址。

31. 一种用于在计算机系统中清除地址转换缓冲区表项的方法, 所述计算机系统包括一个或多个地址转换表, 所述地址转换表提供转换信息以将虚拟地址转换为存储器地址, 每个地址转换缓冲区表项都动态地缓存了地址转换信息, 所述方法包括以下步骤:

从要执行的机器可执行指令的操作码判定所述指令被配置为开始执行清除操作;

执行所述指令, 所述执行步骤包括以下步骤:

根据第一转换表的第一转换表起始地址, 其中所述第一转换表起始地址包括地址空间控制元素 (ASCE) 地址、区表起始地址或段表起始地址中的任何一个地址, 从所述地址转换缓冲区选择性地清除所述地址转换缓冲区的的一个或多个第一表项, 所述一个或多个第一表项与所述第一转换表起始地址关联。

32. 如权利要求 31 中所述的方法, 其中所述清除步骤不会清除所述地址转换缓冲区中所述地址转换缓冲区的的一个或多个第二表项, 所述一个或多个第二表项没有与所述第一转换表起始地址关联。

33. 一种用于在计算机系统中清除地址转换缓冲区表项的方法, 所述计算机系统包括一个或多个地址转换表, 所述地址转换表提供转换信息以将虚拟地址转换为存储器地址, 每个地址转换缓冲区表项都动态地缓存了地址转换信息, 所述方法包括以下步骤:

从要执行的机器可执行指令的操作码判定所述指令被配置为开始执行

清除操作;

执行所述指令, 所述执行步骤包括以下步骤:

根据第一转换表的第一转换表起始地址, 其中所述第一转换表起始地址包括一系列两个或更多转换表表项的第一转换表表项, 从所述地址转换缓冲区选择性地清除所述地址转换缓冲区的一个或多个第一表项, 所述一个或多个第一表项与所述第一转换表起始地址关联。

34. 如权利要求 33 中所述的方法, 其中所述清除步骤不会清除所述地址转换缓冲区中所述地址转换缓冲区的一个或多个第二表项, 所述一个或多个第二表项没有与所述第一转换表起始地址关联。

35. 如权利要求 34 中所述的方法, 还包括无效所述系列的两个或更多转换表表项的步骤。

36. 一种用于在计算机系统中清除地址转换缓冲区的系统, 所述系统包括:

主存储器;

一个或多个与所述主存储器通信的中央处理单元, 所述中央处理单元包括:

架构的通用寄存器;

地址转换缓冲区, 地址转换缓冲区具有保存地址转换信息的地址转换缓冲区表项;

多个地址转换表, 所述地址转换表用于将所述一个或多个处理单元中的一个中央处理单元的虚拟主存储器地址转换成所述计算机系统的主存储器地址;

一个或多个指令执行单元, 所述指令执行单元与所述架构的通用寄存器通信, 所述指令执行单元执行从所述主存储器取出的机器指令;

其中所述计算机系统包括指令以执行包括以下步骤的方法:

从要执行的机器可执行指令的操作码判定所述指令被配置为开始执行清除操作;

执行所述指令, 所述执行步骤包括以下步骤:

根据第一转换表的第一转换表起始地址，其中所述第一转换表起始地址包括地址空间控制元素（ASCE）地址、区表起始地址或段表起始地址中的任何一个地址，从所述地址转换缓冲区选择性地清除所述地址转换缓冲区的一个或多个第一表项，所述一个或多个第一表项与所述第一转换表起始地址关联。

37. 如权利要求 36 中所述的系统，其中所述清除步骤不会清除所述地址转换缓冲区中所述地址转换缓冲区的一个或多个第二表项，所述一个或多个第二表项没有与所述第一转换表起始地址关联。

38. 一种用于在计算机系统中清除地址转换缓冲区的系统，所述系统包括：

主存储器；

一个或多个与所述主存储器通信的中央处理单元，所述中央处理单元包括：

架构的通用寄存器；

地址转换缓冲区，地址转换缓冲区具有保存地址转换信息的地址转换缓冲区表项；

多个地址转换表，所述地址转换表用于将所述一个或多个处理单元中的一个中央处理单元的虚拟主存储器地址转换成所述计算机系统的主存储器地址；

一个或多个指令执行单元，所述指令执行单元与所述架构的通用寄存器通信，所述指令执行单元执行从所述主存储器取出的机器指令；

其中所述计算机系统包括指令以执行包括以下步骤的方法：

从要执行的机器可执行指令的操作码判定所述指令被配置为开始执行清除操作；

执行所述指令，所述执行步骤包括以下步骤：

根据第一转换表的第一转换表起始地址，其中所述第一转换表起始地址包括一系列两个或更多转换表表项的第一转换表表项，从所述地址转换缓冲区选择性地清除所述地址转换缓冲区的一个或多个第一表项，所述一

个或多个第一表项与所述第一转换表起始地址关联。

39. 如权利要求 38 中所述的系统，其中所述清除步骤不会清除所述地址转换缓冲区中所述地址转换缓冲区的一个或多个第二表项，所述一个或多个第二表项没有与所述第一转换表起始地址关联。

40. 如权利要求 38 中所述的方法，还包括无效所述系列的两个或更多转换表表项的步骤。

41. 一种用于在计算机系统中清除转换旁视缓冲区 (TLB) 地址转换缓冲区的地址转换缓冲区表项的方法，所述计算机系统包括一个或多个地址转换表，所述地址转换表提供转换信息以将虚拟地址转换为存储器地址，每个地址转换缓冲区表项都动态地缓存了地址转换信息，所述方法包括以下步骤：

所述计算机系统从主存储器取出动态地址转换表表项 (IDTE) 指令，所述指令包括操作码和用于确定第二信息的第二字段，所述第二信息包括地址空间控制元素 (ASCE) 地址或地址转换表的起始地址，所述地址转换表包括：

第一区表，所述第一区表具有包括第二区表起始地址的表项，

第二区表，所述第二区表具有包括第三区表起始地址的表项，

第三区表，所述第三区表具有包括段表起始地址的表项；或

段表，所述段表具有包括页表起始地址的表项，其中页表包括存储器地址；

从所述操作码判定所述指令被配置为开始执行清除操作；以及

执行所述指令，所述执行步骤包括以下步骤：

i) 判定地址转换缓冲区的地址转换缓冲区表项是否与所述第二信息相关；以及

j) 清除判定为与所述第二信息相关的所述地址转换缓冲区表项；以及

k) 为多个地址转换缓冲区表项重复步骤 i) - j)。

42. 如权利要求 41 中所述的方法，其中所述计算机系统包括多个转换旁视缓冲区，还包括为所述计算机系统的所述多个转换旁视缓冲区执行

步骤 i) -j) 的步骤。

43. 如权利要求 41 中所述的方法, 还包括以下步骤:

解释所述指令以标识用于仿真所述方法的步骤的预定软件例程, 所述预定软件例程包括多个指令; 以及

执行所述预定软件例程。

44. 一种用于在计算机系统中清除地址转换缓冲区的系统, 所述系统包括:

主存储器;

一个或多个与所述主存储器通信的中央处理单元, 所述中央处理单元包括:

架构的通用寄存器;

一个或多个转换旁视缓冲区 (TLB) 地址转换缓冲区, 地址转换缓冲区具有保存地址转换信息的地址转换缓冲区表项;

多个地址转换表, 所述地址转换表用于将所述一个或多个处理单元中的一个中央处理单元的虚拟主存储器地址转换成所述计算机系统的主存储器地址;

一个或多个指令执行单元, 所述指令执行单元与所述架构的通用寄存器通信, 所述指令执行单元执行从所述主存储器取出的机器指令;

其中所述计算机系统包括指令以执行包括以下步骤的方法:

所述计算机系统从主存储器取出动态地址转换表表项 (IDTE) 指令, 所述指令包括操作码和用于确定第二信息的第二字段, 所述第二信息包括地址空间控制元素 (ASCE) 地址或地址转换表的起始地址, 所述地址转换表包括:

第一区表, 所述第一区表具有包括第二区表起始地址的表项,

第二区表, 所述第二区表具有包括第三区表起始地址的表项,

第三区表, 所述第三区表具有包括段表起始地址的表项; 或

段表, 所述段表具有包括页表起始地址的表项, 其中页表包括存储器地址;

从所述操作码判定所述指令被配置为开始执行清除操作；以及
执行所述指令，所述执行步骤包括以下步骤：

i) 判定地址转换缓冲区的地址转换缓冲区表项是否与所述第二信息相关；以及

j) 清除判定为与所述第二信息相关的所述地址转换缓冲区表项；以及

k) 为多个地址转换缓冲区表项重复步骤 i) - j) 。

45. 如权利要求 44 中所述的系统，其中所述计算机系统包括多个转换旁视缓冲区，还包括为所述计算机系统的所述多个转换旁视缓冲区执行步骤 i) - j) 的步骤。

46. 如权利要求 44 中所述的系统，还包括以下步骤：

解释所述指令以标识用于仿真所述系统的步骤的预定软件例程，所述预定软件例程包括多个指令；以及

执行所述预定软件例程。

47. 一种包括指令的计算机程序，当所述计算机程序在计算机系统上执行时，所述指令用于执行如上述任何方法权利要求所述的方法的所有步骤。

无效存储器，清除缓冲区表项

技术领域

本发明一般地涉及计算环境内的处理，具体地说，涉及一次或多次无效存储器、清除缓冲区表项以及用于上述目的指令。

背景技术

现有计算环境使用虚拟存储器（通常在辅助存储器中对其进行维护）来增加可用存储器的大小。当某一虚拟存储页被引用时，用于引用该页的虚拟地址通过动态地址转换被转换成实存储器地址。如果转换成功，则该虚拟存储器是有效的；否则，该虚拟存储器是无效的。通过页表表项中的无效位将虚拟存储页指示为有效或无效，如果该页有效，则所述表项包括实存储器地址。

多个存储页一次可以被无效一页。例如，在基于 z/Architecture（由国际商业机器公司提供）的计算环境中，指令（称为无效页表表项（IPTE）指令）被用来无效存储页。无效包括在控制结构（与虚拟存储器关联并位于物理主存储器内）中设置无效指示符，以指示虚拟存储器中某一位置的虚拟地址不能通过动态地址转换被转换成物理主存储器地址（也称为实地址）。进而，所述指令被用来清除与页关联的内部机器缓冲区表项，在动态地址转换过程中可以使用该表项来避免存取物理主存储器中的控制结构。例如，在 z/Architecture 中，在虚拟地址到实地址的动态地址转换过程中使用了转换旁视缓冲区（Translation Lookaside Buffers）。当存储页被无效时，相应缓冲区的与该页关联的表项也将被清除。

还存在允许清除整个转换旁视缓冲区的指令。例如，在 z/Architecture 中，指令（称为比较、交换和清除（CSP）指令）将清除转换旁视缓冲区的所有表项。

因此,尽管存在无效存储页并清除缓冲区的与该页关联的表项的指令,以及清除整个缓冲区的指令,但是仍然需要这样的指令,所述指令可以有选择地无效各种大小的存储单元和/或清除缓冲区的与之关联的表项。例如,需要这样的指令,所述指令允许指定大于将被无效的页的选定存储单元和/或使与之关联的缓冲区表项被清除。还需要这样的指令,所述指令允许清除选定存储单元的缓冲区表项,而不影响其他存储单元的缓冲区表项。

发明内容

本发明提供了一种如权利要求 1 所要求的无效计算环境的存储器的方法。

优选地,本发明提供了一种无效计算环境的存储器的方法。所述方法包括例如指定将被无效的存储单元,所述存储单元包括多个存储页,并且其中所述指定包括提供所述存储单元的描述符;以及无效所述存储单元。

优选地,本发明提供了一种有助于清除计算环境的缓冲区的方法。所述方法包括例如标识存储单元,将为该存储单元执行对一个或多个缓冲区的一个或多个表项的清除,其中所述存储单元包括多个存储页;以及清除与所述标识的存储单元关联的一个或多个缓冲区的一个或多个表项,其中与其他存储单元关联的一个或多个表项不会被清除。

优选地,提供了将在计算环境中执行的指令。所述指令包括例如操作码,以标识将被执行的指令;无效标志,以指定与将被无效的存储单元关联的信息,所述存储单元包括多个存储页;其他标志,以指定所述指令所使用的附加信息;以及清除标志,以指定与所述存储单元和其他存储单元中的一个存储单元关联的信息,将为该存储单元清除一个或多个缓冲区的一个或多个表项,其中一个或多个所述无效标志的一个或多个组件、所述其他标志以及所述清除标志可以被配置为无操作。

优选地,提供了将在计算环境中执行的指令。所述指令包括例如操作码,以标识将被执行的指令;以及以下各项中的至少一个:第一无效标志,以指定与将被无效的存储单元关联的信息,所述存储单元包括多个存储页;

第二无效标志，以指定与将被无效的所述存储单元关联的附加信息；以及清除标志，以指定与所述存储单元和其他存储单元中的一个存储单元关联的信息，将为该存储单元清除一个或多个缓冲区的一个或多个表项，其中所述由所述清除标志所指定的信息不与将不会为其清除一个或多个表项的至少一个其他存储单元关联。

此处还描述并要求保护对应于上述方法的系统和计算机程序产品。

通过本发明的技术实现了其他特征和优点。此处详细描述了本发明的其他实施例和方面，并且它们被看作所要求保护的发明的一部分。

附图说明

在本说明书结尾处的权利要求书中特别指出并明确要求了被看作本发明的主题。通过以下结合附图的详细说明，本发明的上述和其他目标、特性和优点是显而易见的，这些附图是：

图 1 示出了结合并使用本发明的一个或多个方面的计算环境的一个实施例；

图 2 示出了根据本发明的一个方面的与图 1 的控制器关联的进一步细节的一个实施例；

图 3 示出了根据本发明的一个方面的可以仿真另一计算机的主机的一个实施例；

图 4 示出了根据本发明的一个方面的与无效存储器关联的逻辑的一个实施例；

图 5 示出了根据本发明的一个方面的与清除缓冲区表项关联的逻辑的一个实施例；

图 6a 示出了根据本发明的一个方面的无效动态地址转换表表项 (IDTE) 指令的格式的一个实施例；

图 6b 示出了根据本发明的一个方面的与图 6a 的 IDTE 指令的通用寄存器 R1 关联的字段的一个实施例；

图 6c 示出了根据本发明的一个方面的与图 6a 的 IDTE 指令的通用寄

寄存器 R2 关联的字段的一个实施例；

图 6d 示出了根据本发明的一个方面的与图 6a 的 IDTE 指令的通用寄存器 R3 关联的字段的一个实施例；

图 7a - 7b 示出了根据本发明的一个方面的与 IDTE 指令关联的逻辑的一个实施例；

图 8a 示出了根据本发明的一个方面的用于 IDTE 指令的可替代操作的通用寄存器 R1 的字段的一个实施例；

图 8b 示出了根据本发明的一个方面的用于 IDTE 指令的可替代操作的通用寄存器 R2 的字段的一个实施例；

图 8c 示出了根据本发明的一个方面的用于 IDTE 指令的可替代操作的通用寄存器 R3 的字段的一个实施例；

图 9 示出了根据本发明的一个方面的与 IDTE 指令的可替代操作关联的逻辑的一个实施例。

具体实施方式

根据本发明的一个方面，存储器（例如，虚拟存储器）的指定部分被无效。作为实例，存储段（其包括多个存储页）或存储区（其包括多个存储段）被选择以便被无效。通过设置无效指示符（其位于与将被无效的存储单元对应的数据结构表项中）可以有助于所述无效。在本发明的进一步的方面中，与正在被无效的存储器或其他存储器关联的缓冲区表项（例如，转换旁视缓冲区表项）被清除。更进一步地，在本发明的另一个方面中，在不影响未选定地址空间的缓冲区表项和不执行无效的情况下，与选定地址空间关联的缓冲区表项被从缓冲区清除。在一个实例中，来自 IBM[®] z/Architecture 的指令（此处被称为无效动态地址转换（DAT）表表项（IDTE）指令）被用来执行一个或多个上述操作。

参考图 1 描述了纳入并使用本发明的一个或多个方面的计算环境 100 的一个实施例。计算环境 100 基于例如由位于纽约阿蒙克的国际商业机器公司提供的 z/Architecture。在标题为《z/Architecture 操作原理》

（“z/Architecture Principles of Operation”，IBM 出版号：SA22-7832-00，2000 年 12 月，其全部内容在此引入作为参考）的 IBM® 出版物中描述了所述 z/Architecture。（IBM® 是位于美国纽约阿蒙克的国际商业机器公司的注册商标。此处所使用的其他名称可以是国际商业机器公司或其他公司的注册商标、商标或产品名称。）在一个实例中，基于 z/Architecture 的计算环境包括由位于纽约阿蒙克的国际商业机器公司所提供的 eServer zSeries。

作为一个实例，计算环境 100 包括与控制器 120 相连的中央处理器复合体（CPC）102。中央处理器复合体 102 包括例如一个或多个分区 104（例如，逻辑分区 LP1-LPn）、一个或多个中央处理器 106（例如，CP1-CPm）以及系统管理程序 108（例如，逻辑分区管理器），它们中的每一个都将在下面进行描述。

每个逻辑分区 104 都能够用作单独的系统。即，每个逻辑分区都可以被独立地重置、初始地加载操作系统（如果需要）并与不同的程序一起运行。在逻辑分区中运行的操作系统或应用程序似乎可以访问整个系统，但实际上，只能访问系统的一部分。硬件和许可的内部代码（通常称为微代码）的组合防止了一个逻辑分区中的程序干扰不同逻辑分区中的程序。这使得多个不同的逻辑分区可以按照时间分片的方式在一个或多个物理处理器上运行。在此特定的实例中，每个逻辑分区都具有驻留的操作系统 110，其对于一个或多个逻辑分区来说可以是不同的。在一个实施例中，操作系统 110 是由位于纽约阿蒙克的国际商业机器公司提供的 z/OS 操作系统。

中央处理器 106 是分配给逻辑分区的物理处理器资源。例如，逻辑分区 104 包括一个或多个逻辑处理器，每个逻辑处理器代表全部或一部分分配给该分区的物理处理器资源 106。特定分区 104 的逻辑处理器既可以专用于该分区以便为该分区保留底层处理器资源，也可以与其他分区共享以便底层处理器资源潜在地可用于其他分区。逻辑分区 104 由系统管理程序 108（其由处理器 106 上运行的微代码来实现）来管理。逻辑分区 104 和系统管理程序 108 都包括一个或多个程序，所述程序驻留在与中央处理器关联的中央存储器的各个部分中。系统管理程序 108 的一个实例是由位于纽

约阿蒙克的国际商业机器公司所提供的处理器资源/系统管理器(PR/SM)。

与中央处理器复合体相连的控制器 120 包括集中逻辑, 所述集中逻辑负责在发出请求的不同处理器之间进行仲裁。例如, 当控制器 120 接收到请求时, 它确定请求者是该请求的主处理器并且其他处理器是从处理器; 它广播消息; 以及在其他情况下处理请求。美国专利 No. 6,199,219 中描述了控制器的一个实例。还参考图 2 描述了进一步的细节。图 2 示出了与多个中央处理器(CPU) 201 相连的控制器 200 的一个实例。在此实例中, 示出了两个中央处理器。但是, 应当理解, 可以有两个以上的处理器与控制器 200 相连。

控制器 200 包括各种控制器, 包括例如系统串行化控制器 202。所述系统串行化控制器被用来确保将被串行化的操作(例如无效指令)被串行化, 因为在某一时刻在计算环境中只有一个此类指令被执行。它还监视该操作的事件顺序。

控制器 200 通过各种接口与每个中央处理器相连。例如, 中央处理器中许可的内部代码使用接口 204 将“控制”命令(其指定将采取的行动)发送给控制器并发送“读出”命令(其从控制器返回信息)。另一个接口是响应总线 206, 其被用来为“读出”命令从控制器返回信息。所述响应总线还被用来为“控制”命令传送命令状况, 并且可以从所述控制器内的多个源(包括系统串行化控制器)来设置。中央处理器可以使用此接口来读出控制器 200 中的系统串行化控制器的状态。

进一步的接口是接口 208, 其由控制器使用以将命令发送给每个 CPU。其也可以从控制器内的多个源(包括系统串行化控制器 202)来控制。更进一步的接口是接口 210, 其向中央处理器 201 的高速缓存控制器 212 提供信号。响应于所述信号, 高速缓存控制器 212 处理命令。在一个实例中, 如以下更详细地说明的, 高速缓存控制器 212 处理影响一个或多个缓冲区(例如转换旁视缓冲区(TLB) 213)的命令。

除了高速缓存控制器 212 之外, 中央处理器 201 包括各种其他控制器, 包括例如中断控制器 220 和执行控制器 222。响应于特定事件, 中断控制

器 220 导致内部中断在 CPU 中被挂起，这依次导致执行控制器 222 在下一个中断点处暂停程序指令处理。响应于所述中断，执行控制器 222 调用许可的内部代码例程来设置广播操作允许锁存器 224，使得高速缓存控制器 212 能够处理挂起的命令。

中央处理器 201 还包括 CPU 静止锁存器 226，其指示中央处理器是否静止。

上述计算环境只是一个实例。例如，一个或多个分区可以在不同的体系结构模式中运行。进而，作为另一个实例，所述环境无需基于 z/Architecture，而是可以基于由 Intel、Sun Microsystems 以及其他公司提供的其他体系结构。此外，环境可以包括仿真器（例如，软件或其他仿真机制），其中特定的体系结构或其子集被仿真。在这种环境中，仿真器的一个或多个仿真功能可以实现本发明的一个或多个方面，尽管执行仿真器的计算机可能具有与正在被仿真的能力不同的体系结构。作为一个实例，在仿真模式中，正在被仿真的特定指令或操作被解码，并且合适的仿真功能被建立以实现单独的指令或操作。

参考图 3 描述了仿真环境的进一步的细节。作为一个实例，主机 300 能够仿真其他计算机的其他体系结构、计算机和/或处理能力。例如，主机 300 基于 Intel 体系结构、RISC 体系结构（如 PowerPC）、Sun Microsystems 提供的 SPARC 体系结构或其他体系结构，并且能够仿真 IBM® 的 z/Architecture 或 IBM® 的其他体系结构或其他实体。

主机 300 包括例如存储器 302，以存储指令和数据；指令取出单元 304，以从存储器 302 取出指令并可选地为取出的指令提供本地缓冲；指令解码单元 306，以接收来自指令取出单元 304 的指令并确定已取出的指令的类型；以及指令执行单元 308，以执行所述指令。执行可以包括将数据从存储器 302 加载到寄存器中；将数据从寄存器存回存储器；或如所述解码单元所确定的，执行某种类型的算术或逻辑运算。

在一个实例中，在软件中实现上述每个单元。例如，由所述各单元执行的操作被实现为仿真器软件内的一个或多个子例程。在另一个实例中，

在固件、硬件、软件或它们的某些组合中实现一个或多个所述操作。

进而，尽管参考仿真描述了图 3，但是图 3 的环境无需是仿真环境。在另一个实例中，在本地环境中执行指令，并且在硬件、固件、软件或它们的某些组合中实现所述操作。

计算环境可以包括虚拟存储器以及主存储器。虚拟存储器可能远远超过配置中可用的主存储器的大小并且通常在辅助存储器中进行维护。虚拟存储器被看作由地址块（称为页）组成。最新访问的虚拟存储页被分配给物理主存储器的占用块。当用户访问没有出现在主存储器中的虚拟存储页时，虚拟页被引入以取代主存储器中不太可能被使用的页。存储器中的页交换可以在用户不知道的情况下由操作系统来执行。

用来指定虚拟存储器中的位置的地址被称为虚拟地址。范围例如最高到 4K 字节的顺序虚拟地址块被称为页。类似地，范围例如最高到 1M 字节的顺序虚拟页块被称为段；并且范围例如最高到 2G 字节的顺序虚拟段块被称为区。进而，与虚拟存储器关联的、由地址空间控制元素（ASCE）所指向的虚拟地址序列被称为地址空间。地址空间可以被用来提供用户间的隔离程度。地址空间可以包括一个或多个区、一个或多个段、一个或多个页或者它们的某些组合。

与不同类型的存储单元（例如区、段、页）关联的是将被用在与存储单元关联的处理中的数据结构。例如，与区关联的是区表；与段关联的是段表；而与页关联的是页表。这些表被用在例如用于访问主存储器的虚拟地址到实地址的转换（例如，动态地址转换）过程中。将用在转换中的表（此处称为转换表）由地址空间控制元素（ASCE）来指定。在标题为《z/Architecture 操作原理》（“z/Architecture Principles of Operation”，IBM 出版号：SA22-7832-00，2000 年 12 月）的 IBM® 出版物中对此进行了进一步的详细说明。当前没有被分配给主存储器的虚拟存储单元被称为无效。虚拟存储单元的无效状态由与该单元关联的数据结构中的无效指示符来指示。

在一个实施例中，动态地址转换机制被实现，使得通过所述 DAT 过

程从转换表（例如，区表、段表和/或页表）得到的信息在位于处理器内的一个或多个缓冲区（此处称为转换旁视缓冲区）中被维护，以便提高地址转换性能。即，在转换过程中，检查缓冲区以便获得所需的信息，如果该信息不在缓冲区中，则存取一个或多个转换表。

在一个实施例中，缓冲区表项被看作以下三种类型之一：TLB 组合区和段表表项；TLB 页表表项；或 TLB 实空间表项，每种类型都将在下面进行描述。

TLB 组合区和段表表项（CRSTE）同时包括从实存储器或绝对存储器中的一个或多个表表项获得的信息以及用于从存储器取出此信息的属性。在一个实例中，TLB 组合区和段表表项（CRSTE）包括以下字段：

TO 用于形成表项的地址空间控制元素中的表起始地址；

RX 用于形成表项的虚拟地址的区索引；

SX 用于形成表项的虚拟地址的段索引；

PTO 从实存储器或绝对存储器中的段表表项取出的页表起始地址；

C 从实存储器或绝对存储器中的段表表项取出的公共段位；以及

P 从实存储器或绝对存储器中的段表表项取出的页保护位。

V 指示 TLB 中的表项是否有效的有效位，并且因此可以被用于查找随后的表项。

TLB 页表表项包括从实存储器或绝对存储器中的一个或多个表表项获得的信息以及用于从存储器取出此信息的属性。在一个实例中，TLB 页表表项包括：

TO 用于形成表项的地址空间控制元素或 TLB 组合区和段表表项中的表起始地址（取决于表项如何被形成）；

PTO 用于形成表项的页表起始地址；

PX 用于形成表项的虚拟地址的页索引；

PFRA 从实存储器或绝对存储器中的页表表项取出的页帧实地址；以及

P 从实存储器或绝对存储器中的页表表项取出的页保护位。

V 指示 TLB 中的表项是否有效的有效位，并且因此可以被用于查找随后的表项。

有时，某一存储单元将被无效。如此处所描述的，为了便于此无效，采用了与该存储单元关联的转换表。进而，响应于无效所述存储器，还可以清除相应的缓冲区表项。例如，为了无效一个存储页，将设置页表表项的无效位。此外，在一个实例中，一个或多个 TLB 中的相应表项被自动清除。在一个特定实例中，由无效页表表项 (IPTE) 指令来执行与存储页关联的表项的无效和清除。

除了无效存储页之外，根据本发明的一个方面，可以无效存储段和/或存储区。作为一个实例，段和/或区表被用在此无效中。参考图 4 描述了与无效特定的存储单元关联的逻辑的一个实施例。

初始地，在步骤 400，标识其中一个或多个表项将被无效的区或段表。在一个实例中，这包括提供区表起始地址或段表起始地址。然后，在步骤 402，指定区或段表内的表项被标识以便进行无效，在步骤 404，所述表项被无效。在一个实例中，通过提供选择被标识的表内的表项的区索引或段索引（虚拟地址位）来完成标识，并且所述无效包括在表项内设置无效指示符。响应于设置无效指示符，相应的存储单元（例如，区或段）被无效。

此后，在查询 406，判定是否有更多的表项将被无效，如果有更多的表项将被无效，则过程继续步骤 402、否则，过程完成。

除了无效区或段表中的一个或多个表项并因此无效那些存储单元之外，还可以从所述环境内的一个或多个处理器上的一个或多个缓冲区中清除或清空相应的表项（或其他表项）。参考图 5 进一步详细描述了与此清除关联的逻辑的一个实施例。

初始地，在查询 500，判定区或段表起始地址是否被指定。如果区或段表起始地址被指定，则在步骤 502，由与所述区或段起始地址匹配的区或段索引指定的表项被从一个或多个缓冲区清除。随后，在查询 504，判定是否有更多的表项将被清除。如果有，则过程继续步骤 502、否则，过程完成。

返回查询 500，如果区或段表起始地址没有被指定，则在步骤 508，由所述区或段索引指定的表项被从一个或多个缓冲区清除。此后，在查询 510，判定是否有更多的表项将被清除。如果有更多的表项将被清除，则过程继续步骤 508。否则，过程完成。

在一个实例中，无效和清除由一个指令来执行。该指令无效段或区表的选定表项并自动从配置的处理器上的 TLB 中至少清除相应的表项(或其他表项)。在一个实例中，从所有处理器上的所有 TLB 中清除所述表项，无论是物理的、虚拟的或由软件仿真的。在一个实例中，如果存在多级 TLB 结构，则所有级都被清除。如果系统中存在多个逻辑分区，则在此和其他处理器上、为逻辑分区(当前 IDTE 指令正在其中被执行)形成的 TLB 表项被清除。

此指令可以在许多体系结构中被实现并可以被仿真。作为实例，所述指令可以在硬件中由处理器来执行；或者通过在具有不同本地指令集的计算机上执行的软件来仿真包括此指令的指令集来执行。在一个特定实例中，所述指令在 z/Architecture 中被实现，并且在此处被称为无效动态地址转换(DAT)表表项(IDTE)指令。

IDTE 提供例如无效和清除选项，其中通过区表或段表的一个或多个表项来无效一个或多个存储区或段，并且至少清除来自配置中的 TLB(例如，所有 TLB)的相应表项(或其他表项)。该指令指定将被清除的 TLB 表项的最小集合。在进一步的实现中，当执行 IDTE 指令时，可以清除更多甚至全部 TLB 表项。

参考图 6a 描述了无效 DAT 表表项(IDTE)指令的格式的一个实施例。在一个实例中，IDTE 指令 600 包括例如指定无效 DAT 表表项指令的操作码 602(例如，‘B98E’ x)和多个寄存器 604-608。以下参考图 6b-6d 进一步详细描述了每个寄存器。

参考图 6b，寄存器 606(其是通用寄存器 R1)具有例如这样的格式：带有表起始地址的地址空间控制元素(例如，位 0-51)，并使用了标志类型控制(位 60 和 61)。位 52-59、62 以及 63 被忽略。具体地说，R1 包括

例如用于无效的区表或段表起始地址 610，其指定了转换表（该转换表的一个或多个表项将被无效）；以及标志类型控制（DT）612，其指定了所述表的类型（例如，区表或段表）。

寄存器 608（图 6c）（其是通用寄存器 R2）包括区第一索引 614，其指定了第一级区表；区第二索引 616，其指定了第二级区表；以及区第三索引 618，其指定了第三级区表。即，为了容纳较大的虚拟地址，使用了三级的区表。寄存器 R2 还包括段索引 620，其是段表中的索引；模式指示符 621，其确定将被执行的功能；以及附加表项指示符 622，其指示多少表项将被无效。例如，通用寄存器 R2 的位 53-63 包括一个二进制整数，其指定了将被无效的附加表表项的数量。在一个实例中，对应于位 53-63 中的值 0-2047，将被无效的表项的数量是 1-2048。通用寄存器 R2 的位 44-51 是零；否则，识别为指定异常。因此，通用寄存器 R2 的位 0-43 具有虚拟地址的区索引和段索引的格式。由 DAT 用来在由通用寄存器 R1 指定的类型的表中选择表项的位 0-43 的部分被称为有效的无效索引，以下将对其进行进一步的详细描述。

寄存器 604（图 6d）（其是通用寄存器 R3）具有例如这样的格式：带有表起始地址的地址空间控制元素，如果 R3 不是零，则位 0-51 被使用。此表起始地址被用来选择将被清除的 TLB 表项，并且其指定的表的类型与通用寄存器 R1 的位 60 和 61 无关。通用寄存器 R3 的位 52-63 被忽略。如果 R3 是零，则通用寄存器 0 的全部内容被忽略。在一个实例中，将被清除的表项可以与被无效的表项相同或不同。

参考图 7a-7b 描述了与无效和清除操作关联的逻辑的一个实施例。在一个实例中，当通用寄存器 R2 的位 52 是零时，无效和清除操作被指定。借助此操作，存储器中指定的区表表项或段表表项或者从该指定的表项开始的某一范围的表项被无效，并且配置的 CPU 中的转换旁视缓冲区（TLB）的关联表项（或其他表项）被清除。在一个实例中，所有 CPU 中的缓冲区的所有级（如果是多级的）都被清除。

参考图 7a，初始地，在步骤 700，指定区表或段表起始地址，所述起

始地址指定了从其将无效一个或多个表项的区表或段表。在一个实例中，这在通用寄存器 R1 中指定。此外，在步骤 702，选择将被无效的存储器中的表表项。在一个实例中，如此处所描述的，通过添加通用寄存器 R1 中的表起始地址和通用寄存器 R2 中的有效的无效索引以及以下各种规则来指定表表项。

通用寄存器 R1 的位 60 和 61 指定了表类型和有效的无效索引，如下表所示：

寄存器 R1 的位 60 和 61	表类型	寄存器 R2 中的有效的无效索引
11	区第一表	区第一索引 (位 0-10)
10	区第二表	区第二索引 (位 11-21)
01	区第三表	区第三索引 (位 22-32)
00	段表	段索引 (位 33-43)

(有效的无效索引右侧的位 0-43 的部分被忽略。)

参考上表，当位 60 和 61 具有二进制值 11 时，R2 的区第一索引部分，结合 R1 的区第一表起始地址，被用来选择区第一表中的表项。表起始地址被看作 64 位地址。

通过将十二个零附加到指定的区第一表的位 0-51 的右侧并且将三个最右侧和五十个最左侧的附加的零添加到区第一索引，可以获得实存储器或绝对存储器中的区第一表表项的 64 位地址。根据用于 64 位地址运算的规则来执行此添加，而与当前寻址模式（由当前程序状态字 (PSW) 的位 31 和 32 来指定）无关。当在添加期间发生位单元零的进位输出时，其将被忽略。由这两个部分形成的地址是实地址或绝对地址。所述地址的所有 64 位都被使用，与当前 PSW 是否指定了 24 位、31 位或 64 位寻址模式无关。

在一个实例中，如其他 CPU 所观察到的，从存储器中同时取出整个区第一表表项。存取表项的取出受到密钥控制的保护，而存储器存取受到

密钥控制的保护和低地址保护。当为取出区第一表表项生成的存储器地址指定了配置中不可用的位置时，寻址异常被识别并且操作单元被禁止。

当位 60 和 61 具有二进制值 10 时，R2 的区第二索引，结合 R1 的区第二表起始地址，被用于从区第二表选择表项。区第二表查找过程与区第一表查找过程相类似。当没有识别到异常时，从区第二表中取出的表项指定了开始位置并指定了相应的区第三表的偏移量和长度。

当位 60 和 61 具有二进制值 01 时，R2 的区第三索引部分，结合 R1 的区第三表起始地址，被用于从区第三表中选择表项。区第三表查找过程与区第一表查找过程相类似。当没有识别到异常时，从区第三表中取出的表项指定了开始位置并指定了相应的段表的偏移量和长度。

当位 60 和 61 具有二进制值 00 时，R2 的段索引部分，结合 R1 的段表起始地址，被用于从段表中选择表项。段表查找过程与区第一表查找过程相类似。当没有识别到异常时，从段表中取出的表项指定了相应的页表的开始位置。

在一个实例中，从存储器中同时取出整个表表项。随后，在步骤 704，该表项被无效。在一个实例中，通过将所述表项的位 58 设置为二进制 1 来无效该表项。通过无效此表项，相应的存储单元被无效。

此后，在查询 706，判定是否将无效更多的表项。在一个实例中，通过检查通用寄存器 R2 的位 53-63 来做出此判定。如果通用寄存器 R2 的位 53-63 不全是零，则有更多的表项将被无效。因此，将 1 添加到先前使用的有效的无效索引的值并且过程继续步骤 700。此过程被重复，以便多个表项（比位 53-63 所指定的数量多一）被无效。有效的无效索引的最左侧的位单元的进位输出被忽略，并且在此情况下，表中出现环绕（wraparound）。通用寄存器 R2 的内容维持不变。

在无效一个或多个表项之后，根据本发明的进一步的方面，在步骤 708，此 CPU 从其 TLB 中清除选定的表项并且发送信号给配置中的其他 CPU 以从它们的 TLB 中清除选定的表项。在一个实例中，至少清除每个 TLB 的符合以下条件的那些组合区和段表表项（CRSTE）：

通用寄存器 R2 中的有效的无效索引和其左侧的各位与 TLB 表项中的区和段索引字段的同一部分相匹配。(在一个实施例中,如果这些字段包括段索引字段,则可以清除 CRSTE (与其区索引字段无关),也可以清除所有 CRSTE。如果所述字段不包括段索引字段,则可以清除所有 CRSTE。)注意,当由于通用寄存器 R2 的位 53-63 而无效多个表项时,有效的无效索引被递增,并且该索引的最左侧位单元的进位输出被丢失。

或者 R3 是零,或者通用寄存器 R3 中的表起始地址字段与所述 TLB 表项中的表起始地址字段相匹配。

如果被无效的表项是段表项,则所述被无效的表项中的页表起始地址字段与所述 TLB 表项中的页表起始地址字段相匹配。(在一个实施例中,可以清除 TLB 表项,而与该表项中的页表起始地址无关。)

还至少清除每个 TLB 的这些页表项:其中页表起始地址字段与(1)被无效的段表项,如果段表项被无效的话,或(2)同一 TLB 中被清除的任何组合区和段表项(其在将被清除的集合内)中的页表起始地址字段相匹配。(在另一个实施例中,所述实现可以清除页表项,而与该表项中的页表起始地址无关。即,其可以清除所有页表项。)

参考图 7b 描述了与从 TLB 清除表项关联的逻辑的一个实施例。作为一个实例,这是由 IDTE 执行的过程的继续,并且其被自动执行到无效,如系统中的此 CPU 和其他 CPU 所观察到的那样。

参考图 7b,在查询 720,初始地判定 R3 是否是零。如果 R3 不是零,则在步骤 722,在寄存器 R3 中指示用于清除的区表或段表起始地址以清除 TLB 表项,寄存器 R3 中指定的区/段表起始地址要与用来形成该表项的区/段表起始地址相匹配。如果 R3 是零,则在步骤 724,寄存器 R3 (寄存器 0)的内容被忽略,并且表项被清除而与用于形成该表项的区/段表起始地址无关。

在步骤 726,从所述系统中的所有处理器上的 TLB 清除由区/段索引(在寄存器 R2 中指定)指定的那些表项。在一个实例中,至少清除 TLB 的这些组合区和段表项:其中例如通用寄存器 R2 中有效的无效索引及

其左侧的各位与所述 TLB 表项中的区和段索引表的同一部分相匹配。作为一个实例，为了清除表项，V 位被关闭。

进而，在步骤 728，如果段表表项正在被无效，则系统中所有处理器上的 TLB 被清除这样的页表表项：其中该表项中的页表起始地址等于 TLB 中的页表起始地址。进而，在步骤 730，如果段表表项正在被无效，则至少清除 TLB 的这些页表表项：其中页表起始地址字段与同一 TLB 中被清除（例如，必要地）的任何 CRSTE 中的页表起始地址字段相匹配。在步骤 732，为多个表项（比图 6c 的附加表项字段 622 中指定的数量多一）重复此清除过程。

此后，在查询 734，判定无效和清除操作是否完成。在一个实例中，发布处理器上的 IDTE 指令将不会被完成，直到该 CPU 上指定的 TLB 表项已被清除并且系统中所有其他 CPU 已完成了使用在此指令中指定的 TLB 表项的任何存储器存取为止。在一个实例中，是静止控制器向此发布处理器指示其他处理器是否已完成了存取。即，该发布处理器被看作主处理器并且受 IDTE 指令影响的任何从处理器都向控制器指示它们的状态。响应于控制器接收到来自从处理器的适当状态，控制器向主处理器指示完成。假如过程完成，则在步骤 736，指令结束。否则，在查询 734，处理器等待预定的时间量并再次检查状态或被提供信号以便完成。

在一个实施例中，如果无效位在区或段表表项中已经是 1，则区或段表表项中的存储以及 TLB 表项的清除可能发生也可能不发生。

尽管在上述实施例中，在无效所需的表项之后执行 TLB 表项的清除，但是在其他实施例中，TLB 表项的清除可以发生在每次无效之后，在选定数量的无效之后，等等。

在本发明的进一步的方面中，可以可替代地使用 IDTE 指令来清除与地址空间（例如，由地址空间控制元素（ASCE）指定）关联的 TLB 表项。在一个实例中，通过将通用寄存器 R2 的位 52 设置为 1 来指示此选择。根据此选择，通用寄存器 R1 的内容（见图 8a）以及通用寄存器 R2 的位 0-43 和 53-63（见图 8b）被忽略。通用寄存器 R3 的内容具有这样的格式：带

有表起始地址的地址空间控制元素，位 0-51 被使用（见图 8c）。通用寄存器 R3 的位 52-63 被忽略。R3 可以是零或非零；即，可以指定任何通用寄存器，包括寄存器 0。此操作不执行存储器的任何无效，但是它确实从配置中的 CPU 的 TLB 中至少清除这些组合区和段表表项以及页表表项：其中表起始地址字段与通用寄存器 R3 中的表起始地址字段相匹配。（在进一步的实施例中，所述实现可以清除页表表项，而与该表项中的表起始地址无关。即，其可以清除所有页表表项。）

参考图 9 描述了与通过 ASCE 清除操作关联的逻辑的一个实施例。初始地，在步骤 900，指定区表或段表起始地址，所述起始地址指定了将为其清除相应的 TLB 表项的区表或段表。在一个实例中，这在通用寄存器 R3 的位 0-51 中指定。此后，在步骤 902，在系统中的所有处理器上至少清除对应于 R3 中指定的表起始地址的组合区和段表表项和页表表项。

随后，在查询 904，判定 IDTE 指令的执行是否完成。在一个实施例中，发布处理器上的 IDTE 指令将不会被完成，直到对应于指定参数的 TLB 表项已在发布 CPU 上被清除并且配置中的其他 CPU 已经完成了任何存储器存取（包括利用对应于指定参数的 TLB 表项来更新变更和参考位）为止。如果表项已被清除并且存储器存取已经完成，则在步骤 906，IDTE 指令的执行完成。否则，在查询 904，处理器等待预定的时间量并再次检查状态或被提供信号以便完成。

以上详细描述了与 IDTE 指令关联的过程。在一个实施例中，所述操作不必对 TLB 实空间表项具有任何影响。下面提供了与 IDTE 指令的一个实施例的过程关联的附加信息、增强和/或实施例。

1. 取决于模型或所选择的实施例，可以以不同方式实现 TLB 表项的有选择的清除。例如，一般地说，可以清除多于所指定的最小数量的表项。当无效和清除操作被执行时，所有包含段索引（其等于是段索引的有效无效索引）的组合区和段表表项（CRSTE）都可以被清除，或者当有效的无效索引不是段索引时，所有 CRSTE 都可以被清除。进而，CRSTE 或页表表项可以被清除，而与该表项中的页表起始地址无关。当执行由 ASCE

清除的操作时，页表表项可以被清除，而与该表项中的页表起始地址无关。在一个实施例中，当执行任一操作时，可以清除所要求的精确的最小数目的表项。

2. 地址空间控制元素可以包含实空间标记起始地址，而不是区表或段表起始地址。通用寄存器 R3 中的表起始地址与该寄存器中的标记起始地址没有区别。

3. 当指定了无效和清除操作时，可以利用段表表项中的页表起始地址来进行 TLB 表项的清除。因此，如果段表表项（当处于附加状态时）包含不同于当前值的页表起始地址，则包含先前值的表项的副本仍可以保留在 TLB 中。

4. 当指定了无效和清除操作时，用于 IDTE 的 DAT 表表项的地址是 64 位地址，并且按照用于 64 位地址运算的正常规则来执行地址运算，并且在 $2^{64}-1$ 处环绕。此外，不使用偏移量和长度字段。因此，不能指定 DAT 表从最大存储位置环绕到位置 0，并且第一个指定的表项与由通用寄存器 R3 的位 63-63 指定的所有附加表项应位于指定的表内。

5. IDTE 指令在 SIE 下受到拦截。在一个实施例中，系统被串行化，以使执行 IDTE 指令的处理器不能开始操作，直到其他处理器已停止使用将被指定的 TLB 表项为止。此外，执行 IDTE 指令的处理器不继续进一步的执行，直到从系统中的所有处理器的 TLB 中清除了适当的表项为止。

6. 在操作开始前执行串行化功能，并且在操作完成后再次执行串行化功能。将串行化应用到此 CPU，其他 CPU 不必被串行化。

以上详细说明的是可以无效段或区表的表项并自动地从一个或多个转换旁视缓冲区至少清除相应的表项（或其他表项）的指令的一个实例。尽管此处描述了一个特定的指令，但是可以做出许多变化而不偏离本发明的精神。例如，可以使用不同的操作码、不同的字段、不同的寄存器甚至无寄存器、不同的位等。例如，其他指令集体系统结构可以以略微不同的方式（例如，不同的操作码、指令中的不同的字段、使用不同的寄存器等）定义等效的指令，但本发明的一个或多个方面仍然适用。此外，如上所述，

本发明的一个或多个方面可以很好地同等地应用到目标体系结构（例如，IBM 的 z/Architecture）的软件仿真。在这种情况下，上述 TLB 可以是矩阵或由软件仿真器实现的其他数据结构，以便保存最近的虚拟到实地址映射。还可以是所述仿真器软件不实现任何与 TLB 相似的结构。在此情况下，只有在存储器中无效表项可以适用。

因此，此处提供了能够无效存储器、无效存储器并自动清除缓冲区表项或清除缓冲区表项的指令。因此，所述指令的一个或多个寄存器可以被配置为无操作（no-ops）。即，在与没有被提供的特定操作（例如，无效、清除等）关联的那些寄存器中的信息（如果有）被忽略或指示操作没有被执行。

尽管以上描述了特定的指令（其中提供了无效、无效并清除或清除），但是所述指令可以被修改或者可以实现只执行由 IDTE 实现的操作的一个或多个方面的其他指令。例如，指令可以只实现存储器的无效。作为进一步的实例，指令可以只从与特定段或区表关联的缓冲区来清除表项。作为更进一步的实例，可以实现只是基于地址空间来清除表项的指令。

在无效和/或清除设备计算机架构的指令格式的商业实现中，所述指令由程序员（例如使用汇编语言进行编写的操作系统程序员）来使用。存储在存储介质中的这些指令格式可以在 z/Architecture IBM 服务器中被本地执行，或者可替代地，在执行其他体系结构的计算机中执行。在现有和将来的 IBM 主机服务器和 IBM 的其他计算机（例如，pSeries Server 和 xSeries Server）中可以对它们进行仿真。它们可以在运行 Linux 的计算机中或使用由 IBM、Intel、AMD、Sun Microsystems 以及其他公司生产的硬件的各种计算机上执行。除了在 z/Architecture 下的硬件上执行外，还可以使用 Linux 以及使用由 Hercules、UMX、FXI 或 Platform Solutions 仿真的计算机（其中通常执行处于仿真模式）。

在仿真模式中，解码被仿真的特定指令，并建立子例程以实现各个指令，如在 C 子例程或驱动程序中那样，或者使用某些其他技术以便为特定硬件提供驱动程序，如在理解了本发明的实施例的描述之后，在本领域的

技术人员的范围之内。各种软件和硬件仿真专利包括，但不限于，美国专利 No. 5,551,013、美国专利 No. 6,009,261、美国专利 No. 5,574,873、美国专利 No. 6,308,255、美国专利 No. 6,463,582、以及美国专利 No. 5,790,825，这些专利示出了实现用于不同计算机架构的指令格式的仿真（用于本领域的技术人员可得到的目标计算机）的各种已知方法，以及示出了由以上参考所使用的那些商业软件技术。

有利地，本发明的一个或多个方面提高了系统性能。例如，通过从缓冲区中有选择地清除表项而不是完全清除缓冲区，提高了系统性能（无需不必要地重新创建表项）。即，需要 20 到 100 个周期来创建 TLB 项，并且有选择的清除避免了不必要地重新创建表项。

作为进一步的优点，如果需要完整的 TLB 清除，则所述实现允许过度清除。进而，如果例如 TLB 没有被使用，则 TLB 部分可以被看作无操作。有利地，可以在硬件、固件、软件或它们的组合中实现所述 IDTE 指令，或者所述 IDTE 指令可以用在目标计算机指令集的仿真器中。

在不偏离本发明的精神的情况下，可以对上述环境进行许多改变。例如，本发明的一个或多个方面可以同等地应用于例如虚拟机仿真，其中一个或多个可分页实体（例如，客户）在一个或多个处理器上执行。作为一个实例，由起始解释执行 (STE) 体系结构来定义可分页客户，标题为“IBM 系统/370 扩展体系结构”（“IBM Syetem/370 Extended Architecture”，IBM 出版号 SA22-7095（1985））的 IBM® 出版物中描述了一个实例。

尽管上面提到了 SIE 和 z/Architecture，但是本发明的一个或多个方面可以同等地应用于采用可分页实体或类似构造的其他体系结构和/或环境。

此外，上述各种实施例只是实例。在不偏离本发明的精神的情况下，可以对这些实施例进行许多改变。例如，尽管此处描述了逻辑分区环境，但是这只是一个实例。本发明的诸方面有益于许多类型的环境，包括具有多个区的其他环境以及未分区的环境。进而，可以没有中央处理器复合体，但是多个处理器被连接到一起。更进一步地，本发明的一个或多个方面可应用于单处理器环境。

尽管此处描述了特定的环境，但是再次地，在不偏离本发明的精神的情况下，可以对这些实施例实现许多改变。例如，如果环境被逻辑地分区，则环境中可以包括更多或更少的逻辑分区。进而，可以有多个连接到一起的中央处理复合体。这些只是可以做出的某些改变。此外，其他变化也是可能的。例如，尽管此处描述的控制器的串行化了指令以便一次执行一条IDTE指令，但是在另一个实施例中，一次可以执行多条指令。进而，所述环境可以包括多个控制器。更进一步地，多个静止请求（来自一个或多个控制器）可以同时地在系统中未完成。其他变化也是可能的。

如此处所使用的，术语“处理单元”包括可分页实体，例如客户；处理器；仿真器；和/或类似组件。此外，术语“通过处理单元”包括代表处理单元。术语“缓冲区”包括存储区域以及不同类型的数据结构，包括但不限于矩阵；并且术语“表”可以包括除表类数据结构之外的其他数据结构。进而，指令可以包括除了指定信息的寄存器之外的其他寄存器。此外，页、段和/或区可以具有不同于此处所描述的大小。

本发明的一种或多种能力可以在软件、固件、硬件或它们的组合中实现。进而，一种或多种能力可以被仿真。

本发明的一个或多个方面可以被包括在一件制品（例如，一个或多个计算机程序产品）中，所述制品具有例如计算机可用介质。所述介质其中包含例如计算机可读程序代码装置或逻辑（例如，指令、代码、命令等）以提供和促进本发明的能力。所述制品可以被包括为计算机系统的一部分或单独出售。

此外，可以提供至少一个程序存储装置，所述装置可由计算机读取，所述计算机包含至少一个可由该计算机执行的指令程序以实现本发明的能力。

此处示出的流程图只是实例。此处描述的这些图或步骤（或操作）可以有許多变化。例如，可以以不同的顺序执行所述步骤，或者可以增加、删除或修改步骤。

一种用于无效计算环境的存储器的装置，所述装置包括：

用于通过指令指定将被无效的存储单元的装置，所述存储单元包括多个存储页；以及

用于无效所述存储单元的装置。

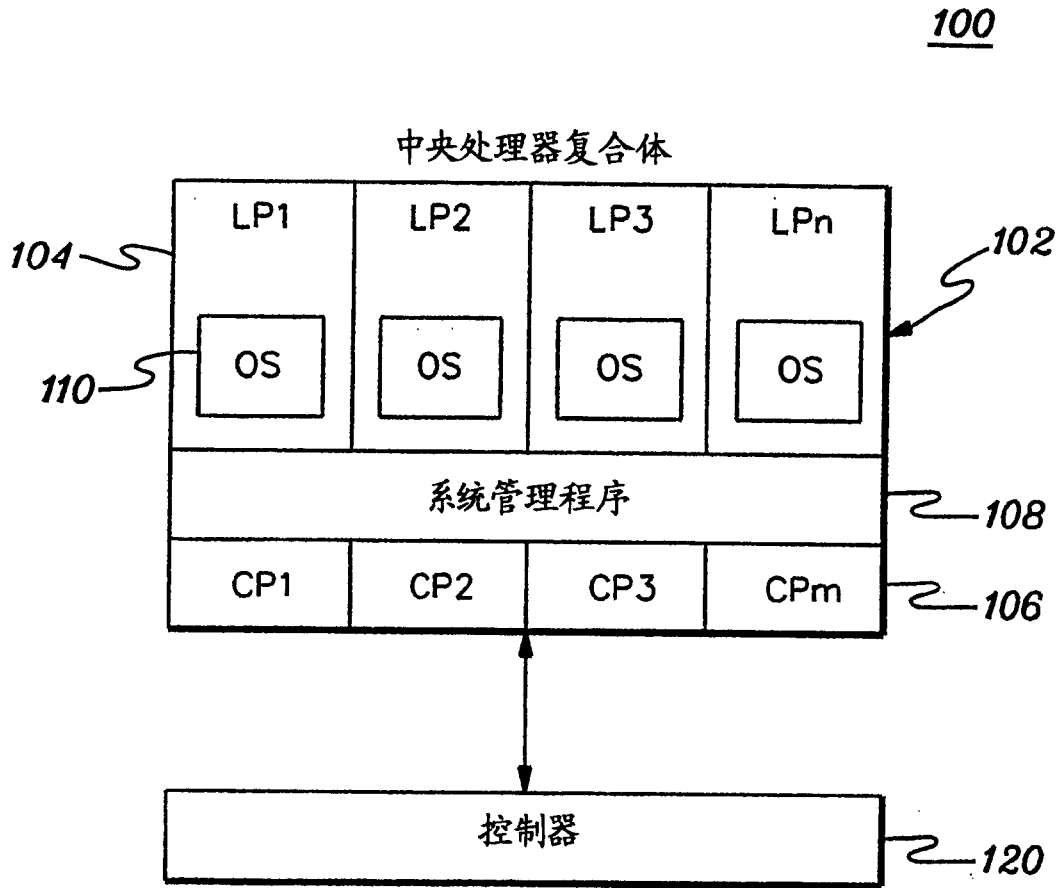


图 1

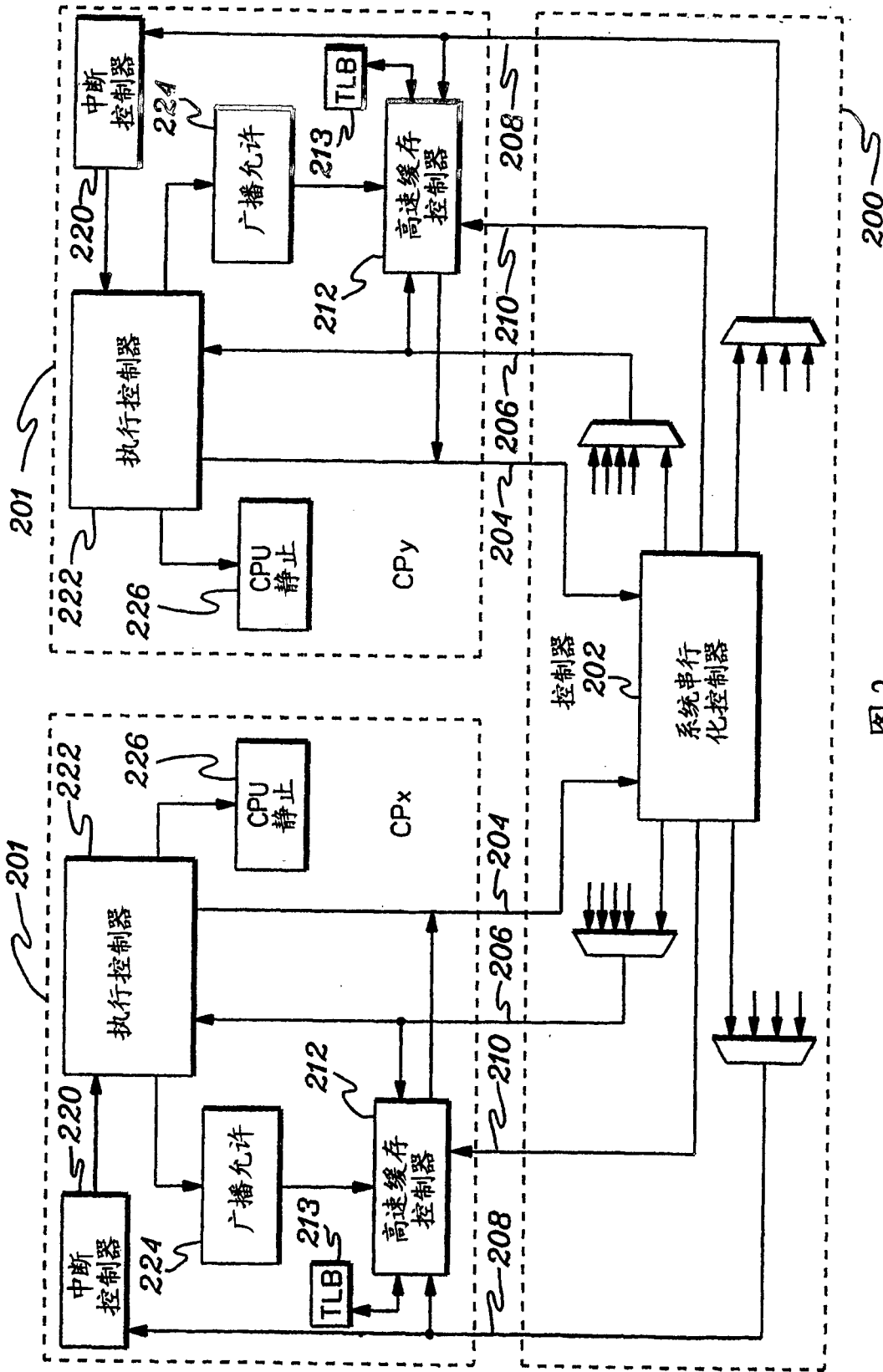


图2

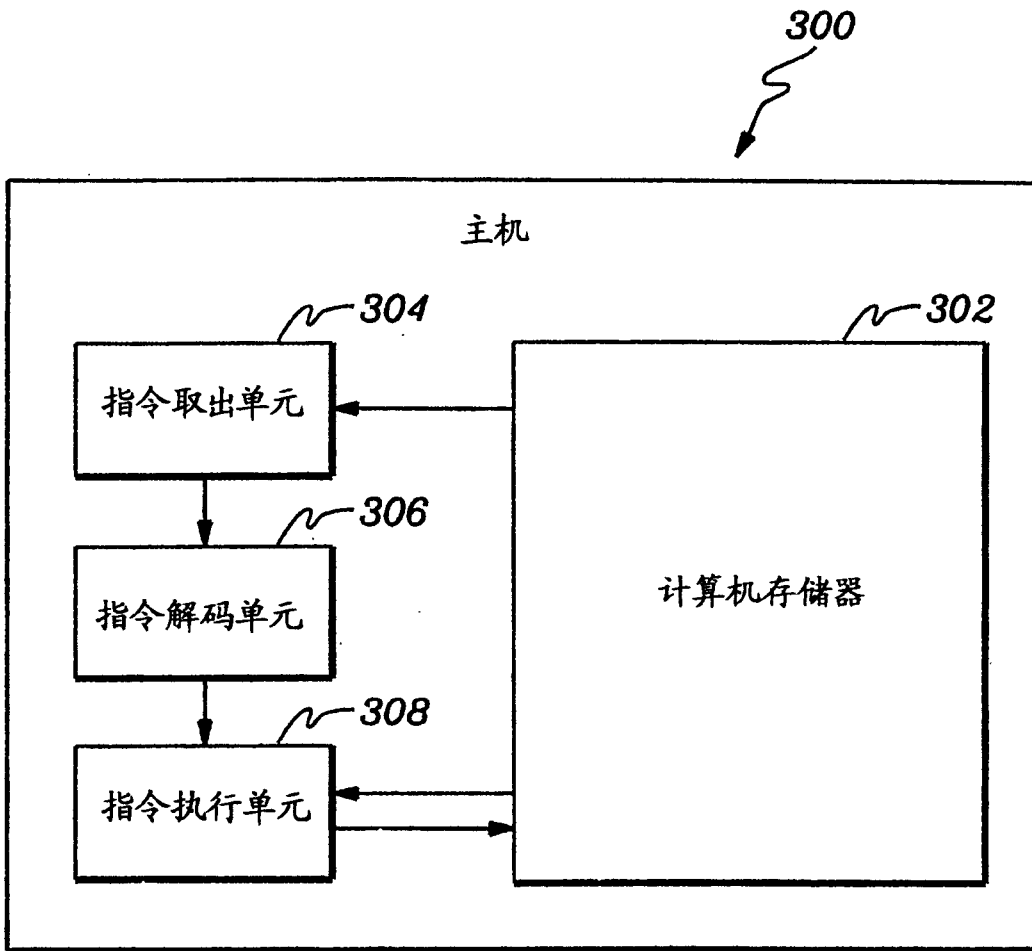


图 3

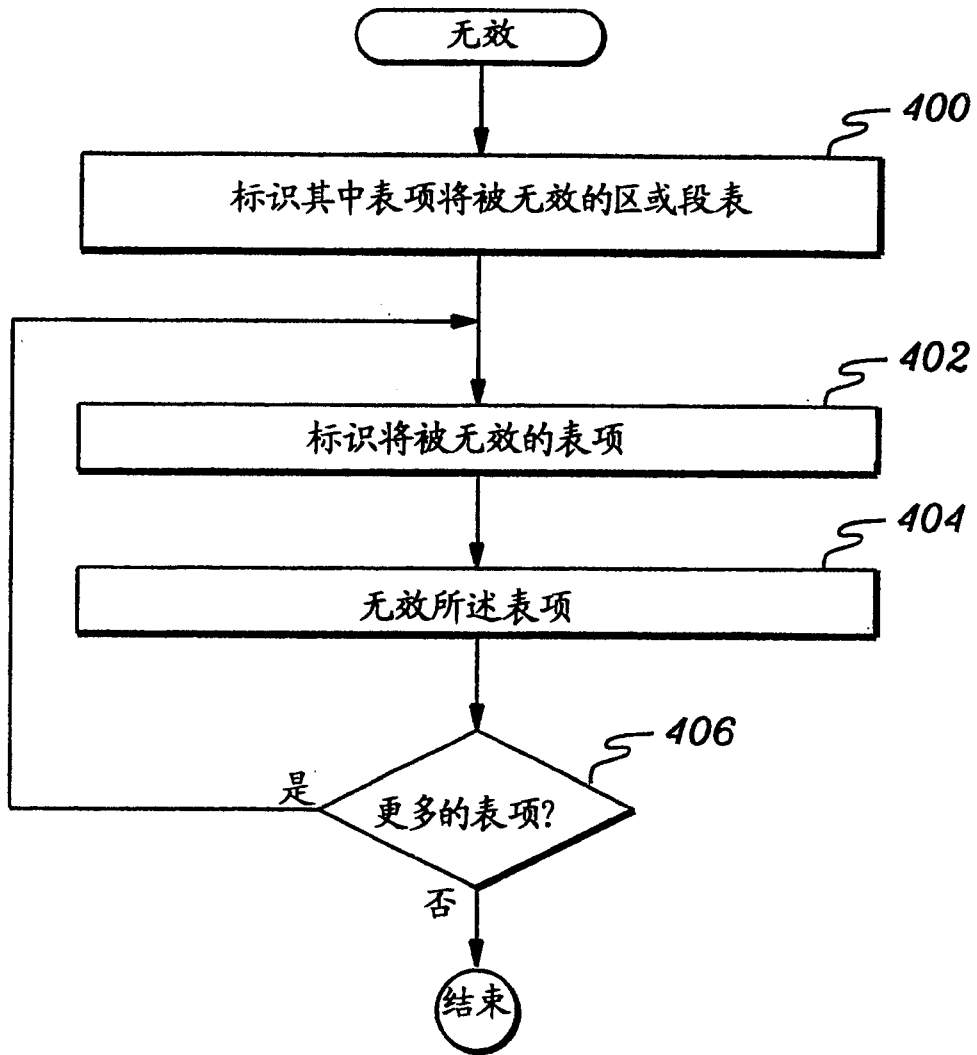


图 4

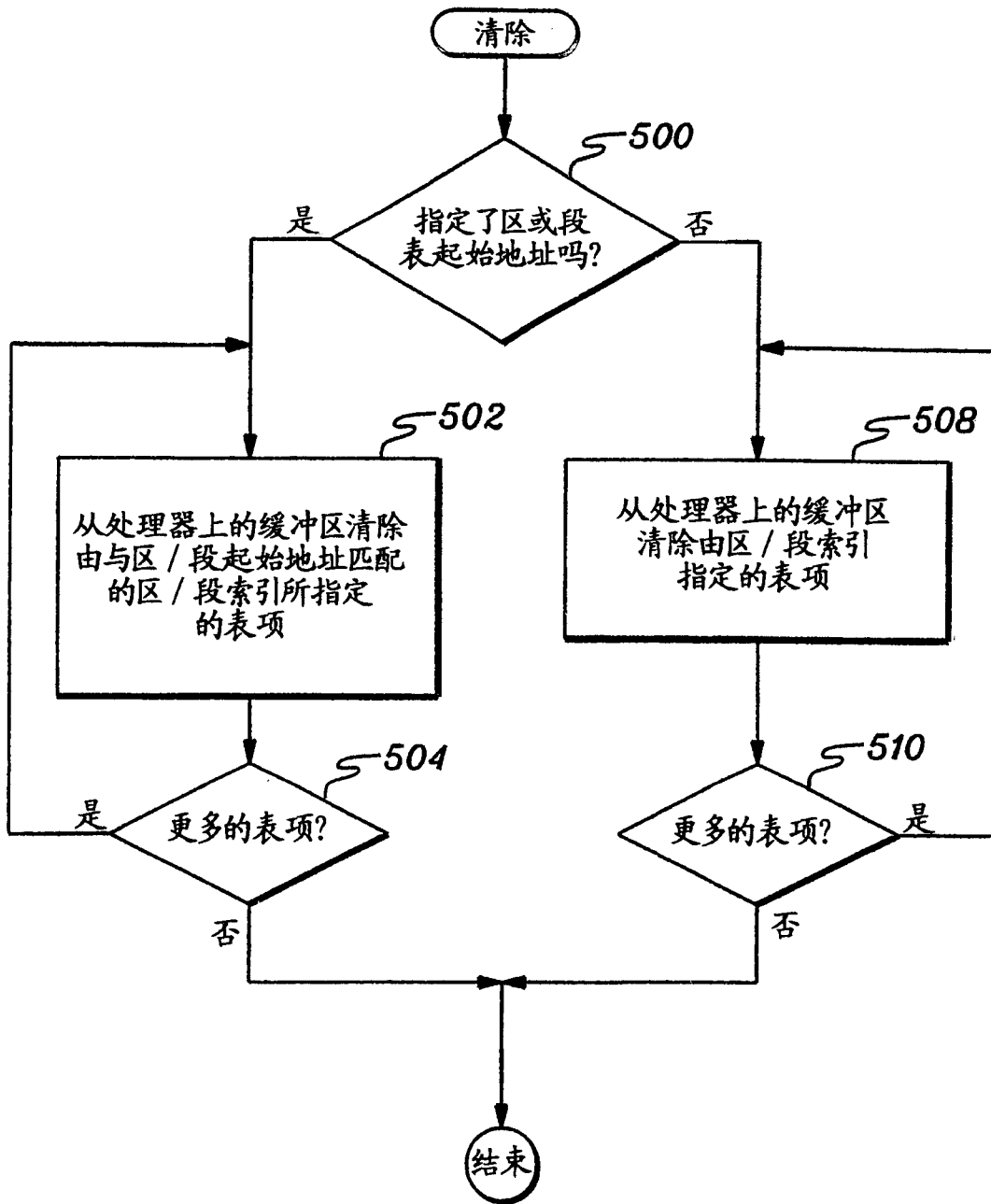


图 5

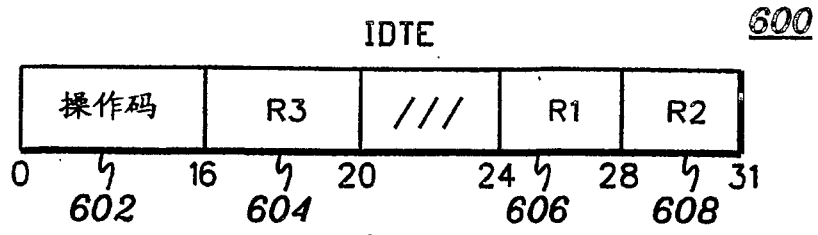


图 6A

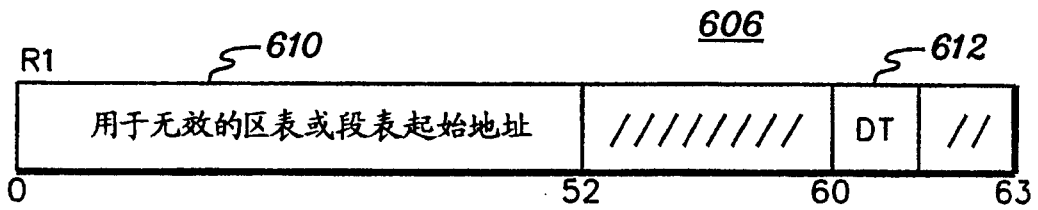


图 6B

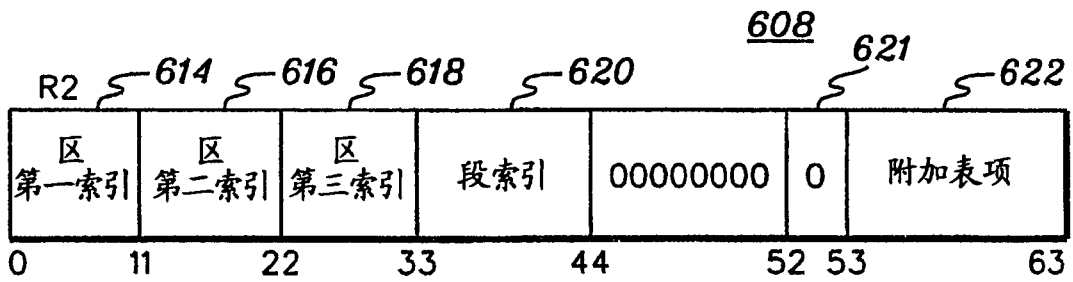


图 6C

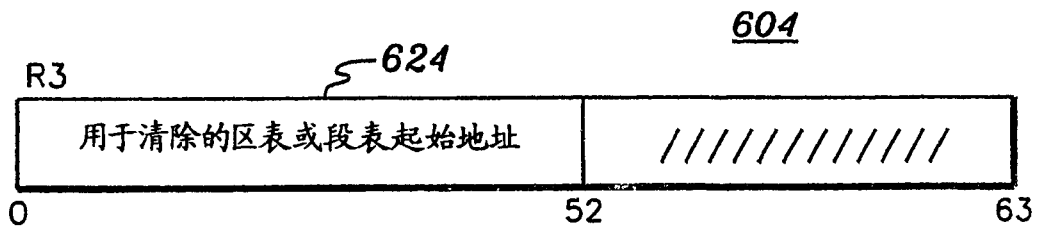


图 6D

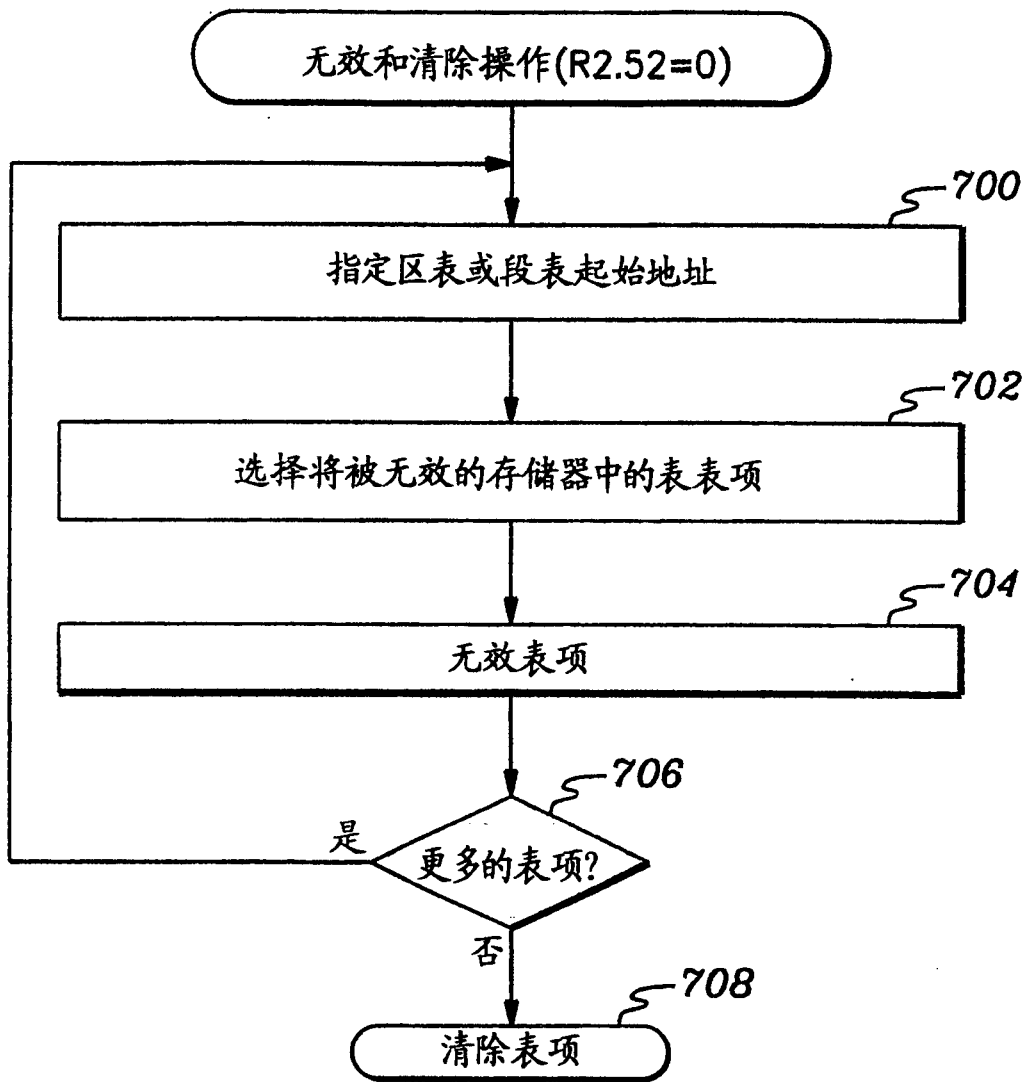


图 7A

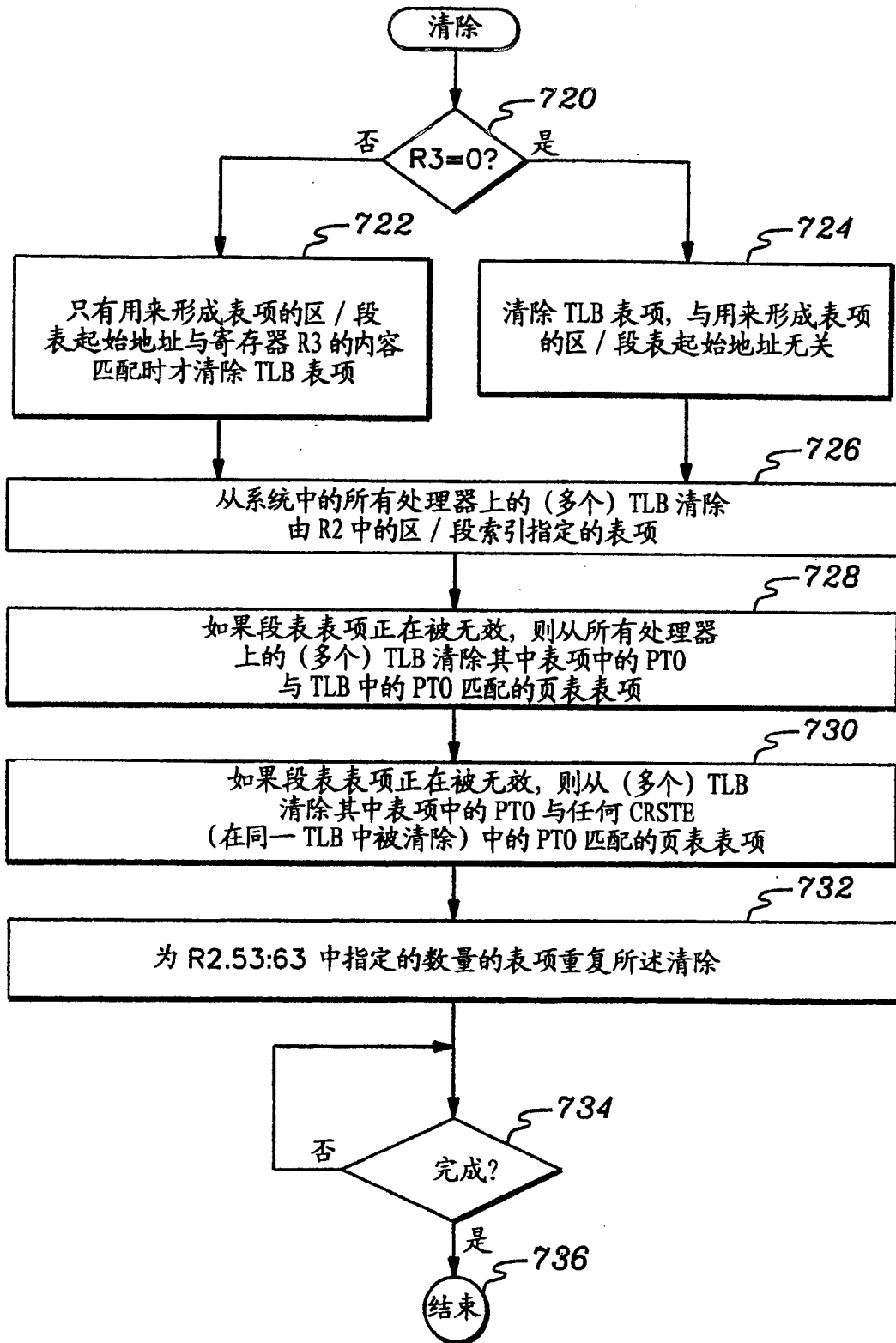


图 7B

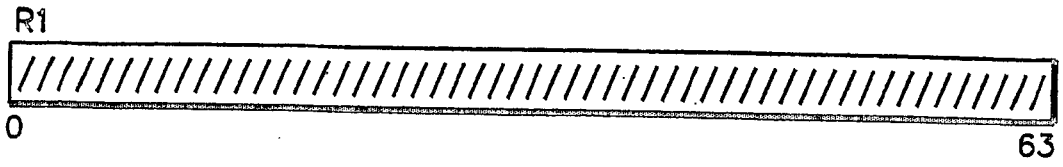


图 8A

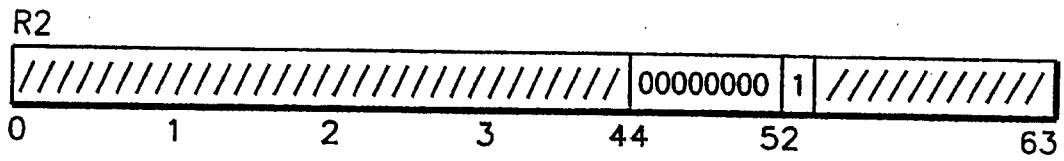


图 8B

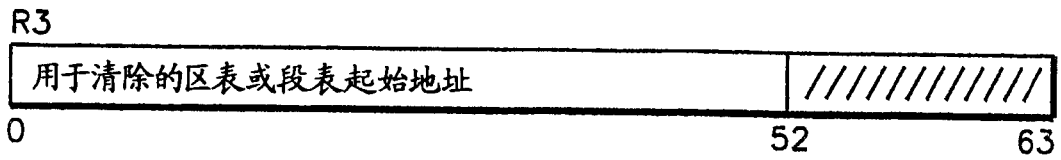


图 8C

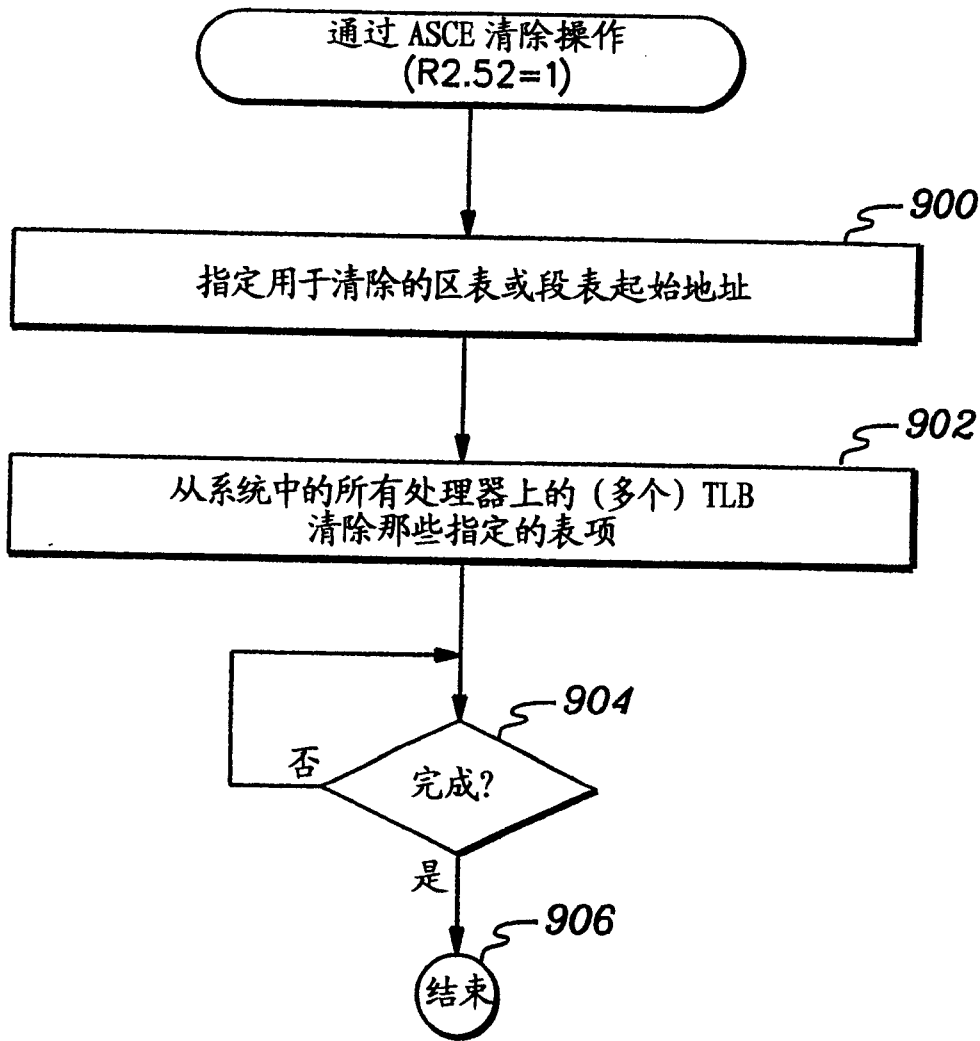


图 9