[19] 中华人民共和国国家知识产权局

[51] Int. Cl[7]
G06F 12/08

[12] 发明专利说明书

[21] ZL 专利号 00132829.8

[45] 授权公告日 2005 年 1 月 26 日

[11] 授权公告号 CN 1186729C

[22] 申请日 2000.10.31 [21] 申请号 00132829.8

[30] 优先权

[32] 1999.10.31 [33] US [31] 09/430793

[71] 专利权人 显露结构开发研究所

地址 美国加利福尼亚州

[72] 发明人 W·R·布赖格 S·G·伯格
G·N·哈蒙德 J·O·哈伊斯
J·C·胡克 J·K·罗斯
S·萨克塞纳 K·雅马达

审查员 石 岗

[74] 专利代理机构 中国专利代理(香港)有限公司

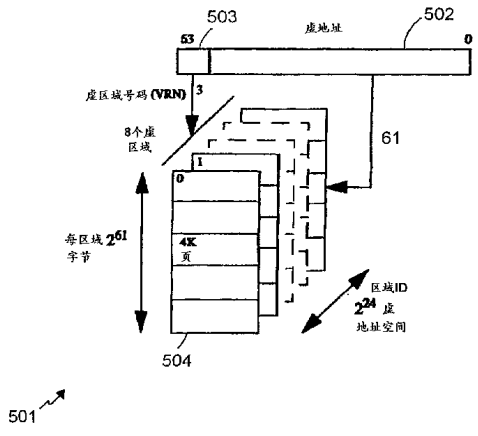
代理人 吴立明 王忠忠

权利要求书 6 页 说明书 22 页 附图 6 页

[54] 发明名称 从虚地址计算页表索引的方法和装置

[57] 摘要

一种从虚地址计算页表索引的方法和装置。本发明像用一种支持两种不同散列页表配置的组合散列算法。“短格式”页表为每一虚拟区域提供，是线性的，它为在该区域内的每一转换有一个线性条目，而不存储标签或链连接。单一“长格式”页表为整个系统提供，它支持链段，并包括散列标签字段。



ISSN1008-4274

1. 一种从虚地址形成引用页表条目的一个入口地址的方法，其中，虚地址包括一个区域部分，它引用标识一个区域的活动区域标识符，该页表能够采取长格式和短格式，所述方法包括:

通过右移该虚地址 J 位从该虚地址形成一个散列页号，其中，与该虚地址的区域部分关联的区域的页大小是 2^J 字节;

如果该页表的格式设定为长格式的话，结合该散列页号和由该虚地址的区域部分引用的区域标识符形成散列索引;

如果该页表的格式设定为长格式的话，左移该散列索引 K 位形成一个表偏移，其中每一长格式页表条目为 2^K 字节长;

如果该页表的格式设定为短格式的话，设定散列索引等于散列页号，形成一个散列索引;

如果该页表的格式设定为短格式的话，左移该散列索引 L 位形成一个表偏移，其中，每一短格式页表条目为 2^L 字节长;

根据该页表的大小形成一个掩码;

使用该页表的基地址和该掩码形成一个第一地址部分;

使用页表偏移和该掩码形成一个第二地址部分;

通过结合第一和第二地址部分形成入口地址。

2. 如权利要求 1 所述方法，另外包括:

如果页表的格式设定为长格式的话，通过从该页表的基地址中减去区域部分形成一个页表区域;

如果页表的格式设定为短格式的话，通过从虚地址中减去区域部分形成一个页表区域;

其中，形成入口地址包括:

通过组合该页表区域与第一和第二地址部分形成入口地址，其中，该页表区域插入该入口地址的区域部分中。

3. 如权利要求 1 所述方法，其中，从虚地址形成散列页号包括:

通过只右移该虚地址已经实现的部分 J 位而从该虚地址形成散列页号，其中，与该虚地址的区域部分相关的区域的页大小为 2^K 字节。

4. 如权利要求 1 所述方法，其中，如果页表格式设定为长格式

的话，通过组合散列页号和由该虚地址的区域部分引用的区域标识符而形成散列索引的步骤包括组合该虚地址的区域部分与该散列页号。

5. 如权利要求 4 所述方法，其中，组合虚地址的区域部分与散列页号的步骤包括在该散列页号中根据右移该虚地址 J 位已知其为空的散列页号的位位置插入该虚地址的区域部分的位。

6. 如权利要求 1 所述方法，其中，根据页表大小形成掩码包括:

设定该掩码等于 2^M 减 1，这里 2^M 是页表大小。

7. 如权利要求 6 所述方法，其中，使用页表的基地址和掩码形成一个第一地址部分包括:

通过对页表的基地址和掩码的逆执行“与”运算形成一个第一地址部分;

使用表偏移和掩码形成一个第二地址部分包括:

通过对该表偏移和掩码执行“与”运算形成一个第二地址部分。

8. 如权利要求 1 所述方法，其中，通过组合第一和第二地址部分形成入口地址包括:

通过对该第一和第二地址部分执行“或”运算形成入口地址。

9. 如权利要求 1 所述方法，其中，已经定义 2^N 字节的最小页表大小，且使用页表的基地址和掩码形成第一地址部分包括:

使用页表的基地址，不包括该页表基地址的低 N 位，和掩码，不包括该掩码的低 N 位，形成第一地址部分;

使用表偏移和掩码形成第二地址部分包括:

使用表偏移，不包括该表偏移的低 N 位，和掩码，不包括该掩码的低 N 位，形成第二地址部分; 和

组合第一和第二地址部分形成入口地址包括:

组合第一和第二地址部分形成一个结果，左移该结果 N 位，组合该结果与表偏移的低 N 位，通过上述步骤形成入口地址。

10. 一种从虚地址形成引用页表条目的一个入口地址的方法，其中，虚地址包括一个区域部分，它引用标识一个区域的活动区域标识符，页表的最小大小为 2^N 字节，该页表能够采取长格式和短格式，所述方法包括:

通过只右移虚地址已经实现的那些部分 J 位而形成该虚地址的

一个散列页号，其中，与该虚地址的区域部分关联的区域的页大小是 2^J 字节；

如果页表的格式设定为长格式的话，通过结合散列页号、虚地址的区域部分、和由该虚地址的区域部分引用的区域标识符而形成一个散列索引，其中，在散列页号中根据右移虚地址 J 位知其为空的散列页号的位位置插入该虚地址的区域部分的位；

如果页表的格式设定为长格式的话，左移该散列索引 K 位形成一个表偏移，其中每一长格式页表条目为 2^K 字节长；

如果页表的格式设定为长格式的话，通过从页表的基地址抽取区域部分形成一个页表区域；

如果页表的格式设定为短格式的话，设定散列索引等于散列页号，形成一个散列索引；

如果页表的格式设定为短格式的话，左移该散列索引 L 位形成一个表偏移，其中，每一短格式页表条目为 2^L 字节长；

如果页表的格式设定为短格式的话，通过从虚地址中抽取区域部分形成一个页表区域；

通过用 2 的 M 次幂减 1 形成一个掩码，这里，2^M 是该表的大小；

通过对页表的基地址，不包括该页表基地址的低 N 位，和掩码的逆，不包括该掩码的逆的低 N 位，执行“与”运算，形成第一地址部分；

通过对表偏移，不包括该表偏移的低 N 位，和掩码，不包括该掩码的低N位，执行“与”运算，形成第二地址部分；和

通过对该第一和第二地址部分执行“与”运算形成一个第一结果，左移该第一结果 N 位形成一个第二结果，对页表区域、该第二结果、和该表偏移的低 N 位执行“或”运算，形成入口地址，其中，页表区域插入该入口地址的一个区域部分。

11. 一个计算机系统，具有定义一个由虚地址寻址的虚地址空间的结构，所述虚地址包括引用标识一个区域的一个活动区域标识符的一个区域部分，所述计算机系统包括：

存储器单元，它包括一个由页表基地址定位的页表，其中，该页表能够采取长格式和短格式；

用于执行指令的处理器，其中，该处理器包括一个页表入口地

址产生单元，它能从虚地址产生到该页表的入口地址，该入口地址产生单元包括:

散列页号产生电路，它通过右移虚地址 J 位形成该虚地址的一个散列页号，其中，与该虚地址的区域部分关联的区域的页大小是 2^J 字节;

散列索引产生电路，它在页表的格式设定为长格式时，通过结合该散列页号和由该虚地址的区域部分引用的区域标识符形成一个散列索引，而在页表的格式设定为短格式时，通过设定散列索引等于该散列页号，形成一个散列索引;

表偏移产生电路，它在页表的格式设定为长格式时，通过左移该散列索引 K 位形成一个表偏移，其中每一长格式页表条目为 2^K 字节长，而在页表的格式设定为短格式时，通过左移该散列索引 L 位形成一个表偏移，其中每一短格式页表条目为 2^L 字节长;

掩码发生电路，它根据页表大小形成一个掩码;

第一地址部分发生电路，它使用页表基地址和掩码形成一个第一地址部分;

第二地址部分发生电路，它使用表偏移和掩码形成一个第二地址部分;

入口地址发生电路，它通过组合第一和第二地址部分形成入口地址。

12. 如权利要求 11 所述计算机系统，其中，页表入口地址发生单元和处理器另外包括:

页表区域发生电路，它在页表的格式设定为长格式时通过从页表基地址中抽取区域部分而形成一个页表区域，而在页表的格式设定为短格式时通过从虚地址中抽取区域部分而形成该页表区域部分;

入口地址发生电路，它通过组合该页表区域和第一和第二地址部分形成入口地址，其中，页表区域插入入口地址的区域部分。

13. 如权利要求 11 所述计算机系统，其中，散列页号发生电路通过只右移虚地址已经实现的那些部分 J 位而从虚地址形成散列页号，其中，与该虚地址的区域部分关联的区域的页大小是 2^J 字节;

14. 如权利要求 11 所述计算机系统，其中，如果页表的格式设

定为长格式的话，则散列索引发生单元通过组合散列页号、由虚地址的区域部分引用的区域标识符、和虚地址的区域部分形成散列索引。

15. 如权利要求 14 所述计算机系统，其中，散列索引发生单元另外在散列页号中根据右移虚地址 J 位知其为空的散列页号的位位置插入该虚地址的区域部分的位。

16. 如权利要求 15 所述计算机系统，其中，掩码发生单元通过设定掩码等于 2^M 减 1 形成该掩码，这里，2^M 是页表的大小。

17. 如权利要求 16 所述计算机系统，其中，第一地址部分发生电路通过对页表的基地址和掩码的逆执行“与”运算形成第一地址部分，第二地址部分发生电路通过对表偏移和掩码执行“与”运算形成第二地址部分。

18. 如权利要求 11 所述计算机系统，其中，入口地址发生电路通过对第一和第二地址部分执行“或”运算形成入口地址。

19. 如权利要求 11 所述计算机系统，其中，定义了 2^N 字节的最小页表大小，以及第一地址部分发生电路使用页表的基地址，不包括该页表基地址的低 N 位，和掩码，不包括该掩码的低 N 位形成第一地址部分，而入口地址发生电路通过组合第一和第二地址部分形成一个结果，左移该结果 N 位，组合该结果与表偏移的低 N 位，形成入口地址。

20. 一种计算机系统，具有定义一个由虚地址寻址的虚地址空间的结构，所述虚地址包括引用标识一个区域的一个活动区域标识符的一个区域部分，所述计算机系统包括:

存储器单元，它包括一个由页表基地址定位的页表，其中，该页表能够采取长格式和短格式，且具有 2^N 位的最小大小;

用于执行指令的处理器，其中，该处理器包括一个页表入口地址发生单元，它能从虚地址产生到页表的入口地址，该入口地址发生单元包括:

散列页号发生电路，它通过只右移虚地址已经实现的部分 J 位而从该虚地址形成一个散列页号，其中，与该虚地址的区域部分相关的区域的页大小为 2^J 字节;

散列索引发生电路，它在页表的格式设定为长格式时，通过组

合散列页号、虚地址的区域部分、和由该虚地址的区域部分引用的区域标识符形成一个散列索引，其中，在该散列页号中根据右移虚地址 J 位知其为空的散列页号的位位置组合插入该虚地址的区域部分，而在页表的格式设定为短格式时，通过设定散列索引等于该散列页号，形成一个散列索引；

页表偏移发生电路，它在页表的格式设定为长格式时，通过左移该散列索引 K 位形成一个表偏移，其中每一长格式页表条目为 2^K 字节长，而在页表的格式设定为短格式时，通过左移该散列索引 L 位形成一个表偏移，其中每一短格式页表条目为 2^L 字节长；

页表区域发生电路，它在页表的格式设定为长格式时通过从页表基地址中抽取区域部分而形成一个页表区域，而在页表的格式设定为短格式时通过从虚地址中抽取区域部分而形成该页表区域部分；

掩码发生电路，它通过用 2 的 M 次幂减 1 形成一个掩码，这里，2^M 是该表的大小；

第一地址部分发生电路，它通过对页表的基地址，不包括该页表基地址的低 N 位，和掩码的逆，不包括该掩码的逆的低 N 位，执行“与”运算，形成第一地址部分；

第二地址部分发生电路，它通过对表偏移，不包括该表偏移的低 N 位，和掩码，不包括该掩码的低 N 位，执行“与”运算，形成第二地址部分；

条目发生电路，它通过对该第一和第二地址部分执行“与”运算形成一个第一结果，左移该第一结果 N 位形成一个第二结果，对页表区域、该第二结果、和表偏移的低 N 位执行“或”运算，形成入口地址，其中，页表区域插入该入口地址的一个区域部分。

从虚地址计算页表索引的方法和装置

技术领域

本发明涉及计算机系统中的存储器组织。更具体说，本发明涉及具有可通过散列函数访问的页表的虚拟存储器系统。

背景技术

常规计算机系统使用一种称为虚拟存储器的技术，虚拟存储器模拟比实际存在更多的逻辑存储器，并允许计算机并列运行几个程序，而不管它们的大小。并发用户程序通过由操作系统指定的虚地址访问主存地址。虚地址向主存的物理地址的映射是称为虚地址转换的过程。可以使用任何数目的技术实现虚地址转换，从而允许处理器访问主存中希望的信息。

虚地址和物理地址空间通常分成称为页的等大的存储器块，而页表提供虚地址和物理地址之间的转换。每一页表条目通常包含虚地址和／或物理地址，和有关该页的保护和状态信息。状态通常包括关于该页曾经经受的访问的类型的信息。例如，已修改位指示曾经对该表内的数据进行过修改。因为页表通常很大，因此它们存储在存储器中。所以，正常的存储器访问实际需要至少两次访问，一次是得到该转换，再一次是访问物理存储器位置。

许多支持虚地址转换的计算机系统使用一种转换查阅缓冲器（TLB）。TLB 通常是一个小而快的关联存储器，它通常位于处理器单元上或在其邻近处，并存储最近使用的虚拟和物理地址对。TLB 包含页表中转换的一个子集，可以被非常快地访问。当处理单元需要来自主存的信息时，它把虚地址发送给 TLB。TLB 接受该虚地址页号，而返回一个物理页号。该物理页号与低阶地址信息结合来访问主存中希望的字节或字。

在大多数场合，TLB 不能包含整个页表，所以需要实现若干步骤来更新该 TLB。当访问一个虚页，而该转换不在 TLB 内时，则访问页表来确定该虚页号对物理页号的转换，这一信息输入到 TLB 中。对页表的访问需要的时间可以比对 TLB 的访问长 20 倍，因此通过保持

在 TLB 中使用转换来优化程序执行速度。

当今大多数计算机系统使用某种海量存储器，通常是磁盘，来扩大计算机中的物理随机访问（RAM）存储器。主存的这一扩大允许实现比只用主存所能执行的程序更大的程序。另外，磁盘存储器要比 RAM 便宜很多，但是也相当慢。取决于程序长度和与其它程序对主存的竞争，在任何特定的时间点，部分程序可以驻留在主存中，而部分驻留在磁盘上。需要被立即访问的程序部分被取入主存，而当前不用的部分留在磁盘上。

例如，考虑一个两兆字节长且在具有 1 兆字节主存的计算机上执行的程序。该程序将需要两兆字节虚地址空间。由于主存只能容纳 1 兆字节，在任何给定时间最多一半程序可以驻留在主存中，剩余的虚地址空间存储在磁盘上。对在主存中的信息的访问正常发生。亦即首先查阅 TLB，看是否有该转换，如果不在 TLB 中，则使用页表中的信息更新 TLB，然后再次引用 TLB 来获得希望的转换信息。

如果发生对不在主存内的信息进行访问的话，则首先访问 TLB，检索不在其内的转换。然后引用页表以获得转换信息来更新 TLB。然而该页表只有为在主存内的信息的转换，因此没有需要的转换信息。这一条件称为页错误。响应页错误，页错误处理程序寻找一个空物理页，用存储在磁盘上的需要的虚页加载该物理页。如果所有物理页已与其它虚页关联，则页错误处理程序需要选择当前存储在物理存储器中的哪一个虚页与磁盘交换。有许多执行这一任务的算法，诸如先进先出和最后使用算法。页错误处理程序通常以软件实现，而 TLB 更新处理可以由硬件或者软件处理，其在技术中公知。

图 1 说明上述处理。在步骤 112 给 TLB 提交一个虚地址。如果为该虚地址的转换存在于 TLB（TLB 命中），则从 TLB 中导出相关物理地址，并用来访问物理存储器（步骤 114）。如果为该虚地址的转换在 TLB 中不存在（TLB 未命中），则为该转换访问页表（步骤 116）。如果该转换在页表中，则将这一信息插入 TLB（步骤 118），并再次提交虚地址（步骤 112）。这一次将会是 TLB 命中，于是使用结果物理地址访问物理存储器。

如果该虚地址在一个无物理地址与之相关的虚地址页内，则在页表内没有为该页的条目，于是发生页错误。在这种情况，软件页

错误处理程序（步骤 120）将给该虚页分配一个物理页，从磁盘复制该页到该物理页，并更新页表。然后再次把该虚地址提交给 TLB。因为 TLB 尚未有该转换，因此会发生另一 TLB 未命中，TLB 将从页表更新。之后，把该虚地址再次提供给 TLB，而这一次保证 TLB 命中，使用结果物理地址访问物理存储器。

图 2 说明响应虚地址的提交访问转换查阅缓冲器（TLB）中一个条目的简化方法。为简化该例，图示 TLB 只有一个条目，而 TLB 通常有多得多的条目。把虚地址加载到寄存器 201。这一虚地址包括两部分，虚页号 203 和物理偏移 205。物理偏移相应于页大小。对于具有页大小 4 千字节的计算机系统，物理偏移 205 是地址的低 12 位（位 11－0），指定一页内的一个特定字节。寄存器内的剩余位指示虚页号。术语“页偏移”是工业中常用的一个术语，其与术语“物理偏移”是同义词。虚地址可以包括用于唯一指定向一个物理页号转换的其它位，诸如“地址空间标识符”位或“区域标识符”位。

对于所述例子，虚页号成为虚标签，它提供给 TLB 比较器 207 的一个输入。TLB209 有两个连接的部分，TLB 标签 211 和一个相关的物理页号 213。TLB 标签 211 给 TLB 比较器 207 提供第二输入，该比较器比较 TLB 标签和虚标签。如果两个标签匹配，则比较器指示 TLB 命中，并把物理页号 213 与物理偏移 205 结合来提供物理（真实）存储器地址。如果两个标签不匹配，则出现 TLB 未命中，则使用参考图 1 说明的 TLB 未命中处理更新 TLB。

图 3 表示给定虚页号检索物理页信息的处理，需要这一处理来在 TLB 未命中后更新 TLB。如上所述，在页表中维持虚拟到物理的映射。为把一个给定的虚地址转换为物理地址，一种方法是对该虚地址执行一种多对单（散列）函数以形成对页表的一个索引。这给出对一个链接的条目表的指针。然后检索这些条目寻找匹配。为确定一个匹配，比较该虚页号和在页表中的一个条目（虚标签）。如果这两个相等，则该页表条目提供物理地址转换。

在所述例子中，对虚页号 203 执行散列函数 301 形成一个索引。该索引是对页表 303 的偏移。如图所示，索引是 0，亦即，该索引指向页表 303 中的第一条目 305。页表中的每一条目包括多个部分，但是通常至少包括有虚标签 307、物理页 309 和指针 311。如果虚页号

203 等于虚标签 307，则物理页 309 给出希望的物理（真实）存储器页地址。如果虚标签不匹配，则指针 311 指向包含虚拟对物理转换信息的存储器中的一个条目链。需要在该链中包含诸如可以散列多于一个的虚页号到同一页表条目的另外的信息。

如图所示，指针 311 指向一个链段。该链段包括同一类型的信息作为开始的页表条目。和前面一样，比较虚页号与下一虚标签 315，看是否匹配。如果匹配，则相关的物理页 317 给出希望的物理存储器页的地址。如果不匹配，则检查指针 309 来定位下一链段，如果存在的话。如果指针 319 不指向另一链段，如图所示，则出现页错误。于是使用页错误软件程序来更新页表，其参考图 1 已说明。

上述方法对于虚标签少于或等于计算机的基本数据路径大小的系统工作的很好。然而，如果虚标签大于该数据路径大小，则需要两次比较来测试虚标签和虚页号是否相同。

因此转让给 Dale Morris 等人的美国专利 NO5724538 在此作为参考，其名称为“使用页表中的散列地址标签的计算机存储器地址控制装置，该散列地址标签与一个组合地址标签和索引比较，其长于相关计算机的基本数据宽度”，它公开了一种减少虚标签大小的方案，从而较少了测试虚标签和虚页号是否相同所需要的比较数目。基本上，Morris 等人认识到部分虚地址已经由散列索引表示，因此部分地址不需要由虚标签表示。

图 4 表示由 Dale Morris 公开的一个实施例的简化方框图。在图 4 中，页表 413 包括“散列标签” 421 和 423。通过取虚页号位 401 并对这些位执行索引散列函数 405 形成散列索引 409，其结果不大于该计算机的基本数据宽度。相似地，通过取虚页号位 401 并执行标签散列函数 427，其结果散列标签不大于该计算机的基本数据宽度。注意，虽然图 4 未表示在标签散列函数 427 和散列标签 421 和 423 之间的明显的连接，但是当产生散列标签 421 和 423 并插入页表 413 中时使用由标签散列函数 427 表示的算法。

索引散列函数 405 和标签散列函数 427 是值得称道的，因为对于任何给定的虚页号，结果散列索引和结果散列标签的组合是唯一的。因此，当访问一个虚页时，对索引散列函数 405 应用该虚页的号码 401 来产生散列索引，其指向页表 413 中的一个散列标签（诸

如散列标签 421 或 423)。从表 413 提供的散列标签导向比较函数 429。同时，也把虚页号 401 提供给标签散列函数 427 来产生散列标签 425。如果散列标签 425 和来自页表 413 的散列标签匹配，则使用该物理页（诸如在条目 317 和 417 存储的物理页）完成存储器访问操作。如果这两个标签不匹配，则访问页表条目的指针（诸如指针 319 和 419）看是否存在一个链段。如果不存在链段，或者检索过所有链段而没有发现有匹配，则调用操作系统的页错误处理程序，如上所述。

注意，既用硬件也用软件访问索引散列函数 405 和标签散列函数 427。当转换一个虚页号为物理页号时硬件必须访问该散列函数，而当初始化该页表并访问和修改该页表时软件必须访问该散列函数，诸如服务于页错误时所需要的。在现有技术中，散列算法基本上以两种形式提供。计算机硬件包括的基于硬件的散列算法版本允许虚拟对物理的转换迅速处理，而操作系统包括的基于软件的散列算法版本在初始化、访问和修改页表时产生虚拟对物理的转换。

某些计算机通过支持多区域扩展虚拟寻址概念。多区域通过把虚地址空间分成等大的区域而提供有效产生在该虚地址空间内独立的局部、共享和全局寻址空间的能力。通常，在任何时间只有多区域的一个子集可以激活。与每一区域相关的有一个区域标识符，它唯一标记给定区域的地址转换。如果把为一个区域的区域标识符分配给一个特定处理，则该区域空间成为对该处理的局部空间。如果为一个区域的区域标识符在多个处理之间共享，则该区域空间成为共享空间。如果为一个区域的区域标识符为所有处理共享，则该区域成为全局空间。改变为局部区域的区域标识符有效交换一个处理的局部空间的虚地址为另一处理的局部空间的虚地址。这样，多区域事实上消除了在切换处理时刷新 TLB 的需要，从而改善了总系统性能。

发明内容

根据本发明的一个方面，提供一种从虚地址形成引用页表条目的一个入口地址的方法，其中，虚地址包括一个区域部分，它引用标识一个区域的活动区域标识符，该页表能够采取长格式和短格式，所述方法包括:

通过右移该虚地址 J 位从该虚地址形成一个散列页号，其中，

与该虚地址的区域部分关联的区域的页大小是 2^J 字节;

如果该页表的格式设定为长格式的话，结合该散列页号和由该虚地址的区域部分引用的区域标识符形成散列索引;

如果该页表的格式设定为长格式的话，左移该散列索引 K 位形成一个表偏移，其中每一长格式页表条目为 2^K 字节长;

如果该页表的格式设定为短格式的话，设定散列索引等于散列页号，形成一个散列索引;

如果该页表的格式设定为短格式的话，左移该散列索引 L 位形成一个表偏移，其中，每一短格式页表条目为 2^L 字节长;

根据该页表的大小形成一个掩码;

使用该页表的基地址和该掩码形成一个第一地址部分;

使用页表偏移和该掩码形成一个第二地址部分;

通过结合第一和第二地址部分形成入口地址。

根据本发明的一个方面，提供一种从虚地址形成引用页表条目的一个入口地址的方法，其中，虚地址包括一个区域部分，它引用标识一个区域的活动区域标识符，页表的最小大小为 2^N 字节，该页表能够采取长格式和短格式，所述方法包括:

通过只右移虚地址已经实现的那些部分 J 位而形成该虚地址的一个散列页号，其中，与该虚地址的区域部分关联的区域的页大小是 2^J 字节;

如果页表的格式设定为长格式的话，通过结合散列页号、虚地址的区域部分、和由该虚地址的区域部分引用的区域标识符而形成一个散列索引，其中，在散列页号中根据右移虚地址 J 位知其为空的散列页号的位位置插入该虚地址的区域部分的位;

如果页表的格式设定为长格式的话，左移该散列索引 K 位形成一个表偏移，其中每一长格式页表条目为 2^K 字节长;

如果页表的格式设定为长格式的话，通过从页表的基地址抽取区域部分形成一个页表区域;

如果页表的格式设定为短格式的话，设定散列索引等于散列页号，形成一个散列索引;

如果页表的格式设定为短格式的话，左移该散列索引 L 位形成一个表偏移，其中，每一短格式页表条目为 2^L 字节长;

如果页表的格式设定为短格式的话，通过从虚地址中抽取区域部分形成一个页表区域;

通过用 2 的 M 次幂减 1 形成一个掩码，这里，2^M是该表的大小;

通过对页表的基地址，不包括该页表基地址的低 N 位，和掩码的逆，不包括该掩码的逆的低 N 位，执行“与”运算，形成第一地址部分;

通过对表偏移，不包括该表偏移的低 N 位，和掩码，不包括该掩码的低N位，执行“与”运算，形成第二地址部分; 和

通过对该第一和第二地址部分执行“与”运算形成一个第一结果，左移该第一结果 N 位形成一个第二结果，对页表区域、该第二结果、和该表偏移的低 N 位执行“或”运算，形成入口地址，其中，页表区域插入该入口地址的一个区域部分。

根据本发明的一个方面，提供一个计算机系统，具有定义一个由虚地址寻址的虚地址空间的结构，所述虚地址包括引用标识一个区域的一个活动区域标识符的一个区域部分，所述计算机系统包括:

存储器单元，它包括一个由页表基地址定位的页表，其中，该页表能够采取长格式和短格式;

用于执行指令的处理器，其中，该处理器包括一个页表入口地址产生单元，它能从虚地址产生到该页表的入口地址，该入口地址产生单元包括:

散列页号产生电路，它通过右移虚地址 J 位形成该虚地址的一个散列页号，其中，与该虚地址的区域部分关联的区域的页大小是2^J字节;

散列索引产生电路，它在页表的格式设定为长格式时，通过结合该散列页号和由该虚地址的区域部分引用的区域标识符形成一个散列索引，而在页表的格式设定为短格式时，通过设定散列索引等于该散列页号，形成一个散列索引;

表偏移产生电路，它在页表的格式设定为长格式时，通过左移该散列索引 K 位形成一个表偏移，其中每一长格式页表条目为2^K字节长，而在页表的格式设定为短格式时，通过左移该散列索引 L 位形成一个表偏移，其中每一短格式页表条目为2^L字节长;

掩码发生电路，它根据页表大小形成一个掩码;

第一地址部分发生电路，它使用页表基地址和掩码形成一个第一地址部分；

第二地址部分发生电路，它使用表偏移和掩码形成一个第二地址部分；

入口地址发生电路，它通过组合第一和第二地址部分形成入口地址。

根据本发明的一个方面，提供一种计算机系统，具有定义一个由虚地址寻址的虚地址空间的结构，所述虚地址包括引用标识一个区域的一个活动区域标识符的一个区域部分，所述计算机系统包括：

存储器单元，它包括一个由页表基地址定位的页表，其中，该页表能够采取长格式和短格式，且具有 2^N 位的最小大小；

用于执行指令的处理器，其中，该处理器包括一个页表入口地址发生单元，它能从虚地址产生到页表的入口地址，该入口地址发生单元包括：

散列页号发生电路，它通过只右移虚地址已经实现的部分 J 位而从该虚地址形成一个散列页号，其中，与该虚地址的区域部分相关的区域的页大小为 2^J 字节；

散列索引发生电路，它在页表的格式设定为长格式时，通过组合散列页号、虚地址的区域部分、和由该虚地址的区域部分引用的区域标识符形成一个散列索引，其中，在该散列页号中根据右移虚地址 J 位知其为空的散列页号的位位置组合插入该虚地址的区域部分，而在页表的格式设定为短格式时，通过设定散列索引等于该散列页号，形成一个散列索引；

页表偏移发生电路，它在页表的格式设定为长格式时，通过左移该散列索引 K 位形成一个表偏移，其中每一长格式页表条目为 2^K 字节长，而在页表的格式设定为短格式时，通过左移该散列索引 L 位形成一个表偏移，其中每一短格式页表条目为 2^L 字节长；

页表区域发生电路，它在页表的格式设定为长格式时通过从页表基地址中抽取区域部分而形成一个页表区域，而在页表的格式设定为短格式时通过从虚地址中抽取区域部分而形成该页表区域部分；

掩码发生电路，它通过用 2 的 M 次幂减 1 形成一个掩码，这里，

2^M是该表的大小;

第一地址部分发生电路，它通过对页表的基地址，不包括该页表基地址的低 N 位，和掩码的逆，不包括该掩码的逆的低 N 位，执行“与”运算，形成第一地址部分;

第二地址部分发生电路，它通过对表偏移，不包括该表偏移的低 N 位，和掩码，不包括该掩码的低 N 位，执行“与”运算，形成第二地址部分;

条目发生电路，它通过对该第一和第二地址部分执行“与”运算形成一个第一结果，左移该第一结果 N 位形成一个第二结果，对页表区域、该第二结果、和表偏移的低 N 位执行“或”运算，形成入口地址，其中，页表区域插入该入口地址的一个区域部分。

本发明是一种从虚地址计算页表索引的方法和装置。本发明通过一个组合的散列算法实现，该散列算法通过配置寄存器和预先定义的常数支持在单一计算机结构中的两种不同的散列页表配置。

本发明例如可以与具有 64 位虚地址的虚拟寻址模式结合使用，其前 3 位形成虚区域部分。相应地，在任何给定时间可以由一个虚地址指定 8 个区域。虚地址剩余的 61 位用于寻址每一区域内的存储器，从而提供给每一区域 2^{61} 字节的虚拟存储器。与每一存储器页相关的有一个 24 位的区域标识符。因此，操作系统可以指定直到 2^{24} 个单独虚地址空间。存储器页可以从 4 千字节到 256 兆字节范围变化。

第一散列页表配置支持一个基于区域的线性页表，在此处将称为“短格式”页表。短格式页表为每一虚区域提供，它是线性的，并对在该区域中的每一转换有一个线性条目。短格式页表不需要链段，并且短格式页表不包括散列标签条目。第二散列页表配置支持为整个计算机系统的单一页表，在此处将称为“长格式”页表。长格式页表支持链段，长格式页表条目包括散列标签字段。

在一个实施例中，本发明的方法从一个虚地址形成一个入口地址，该入口地址引用页表的一个条目。为形成该入口地址，首先从虚地址通过右移该虚地址 J 位形成一个散列页号，其中与该虚地址的区域部分关联的区域的大小优选为 2^J 字节。

如果计算机系统以长格式页表操作，则下一步骤是通过组合该

散列页号和由虚地址的区域部分引用的区域标识符而形成一个散列索引，和通过左移该散列索引 K 位形成一个表偏移，其中每一长格式页表条目为 2^K 字节长。

然而，如果计算机系统以短格式页表操作，则下一步是通过设定散列索引等于散列页号而形成一个散列索引，和通过左移该散列索引 L 位而形成一个表偏移，其中每一短格式页表条目为 2^L 字节长。

接着，根据页表大小形成一个掩码。然后使用页表的基地址和该掩码形成第一地址部分，使用表偏移和该掩码形成第二地址部分。最后，结合第一和第二地址部分形成入口地址。

在一个实施例中，在入口地址中插入一个区域部分。如果设定格式为长格式，则该区域部分从页表的基地址的区域部分导出。然而，如果该区域设定为短格式，则从虚地址的区域部分导出该区域部分。

在另一个实施例中，长格式页表的最大尺寸通过在设定格式为长格式时在散列页号中插入虚地址的区域部分而增加。

本发明还包括几个实施例，它们基于某些实现独立的参数减少用于实现本发明的逻辑数量。通过提供能够为长格式和短格式页表两者产生一个页表条目的单一算法，本发明减少了访问这两种页表格式需要的逻辑数量，而不明显影响执行速度。

附图说明

图 1 表示现有技术响应在一个程序执行期间提供的虚地址的处理过程。

图 2 表示现有技术访问转换查阅缓冲器（TLB）中一个条目的方法。

图 3 表示现有技术在 TLB 未命中后检索物理页信息更新 TLB 的方法。

图 4 表示一个现有技术页表模式，其中在一个页表中存储散列标签。

图 5 表示由本发明支持的一个虚拟寻址模式。

图 6 表示一个“短格式”虚散列的页表的条目，其可以由本发明的散列函数的应用程序访问。

图 7 表示一个“长格式”虚散列的页表的条目，其可以由本发

明的散列函数的应用程序访问。

具体实施方式

本发明是一种为从虚地址计算页表索引和散列标签的方法和装置。本发明由一个组合的散列算法和一个从虚地址产生散列标签的算法实现，所述组合散列算法通过配置寄存器支持在单一计算机结构中的两个不同的散列表结构。

在详细说明本发明之前，首先考虑本发明在其内可以实现的结构框架。为此，结合由 Stephen Burger 等人提交的、名称为“为把基于硬件的虚拟存储器散列模式暴露给软件的方法和装置”的待审美国专利申请作为参考。该申请转让给本申请的同一受让人，在 1998 年 10 月 12 日申请，分配的美国序列号为 09／170143。Burger 等人公开了两个把由硬件使用的散列算法暴露给软件来访问一个页表的指令。第一指令是转换散列的入口地址（THASH）指令，它从一个虚地址产生指向在页表中的一个条目的一个散列索引。第二指令是转换散列的条目标签（TTAG）指令，它从一个虚地址产生存储在由该散列索引引用的页表条目中的一个散列标签。通过提供这两个指令，Burger 等人教导，计算机操作系统（或其它系统软件）不需用由计算机硬件使用的散列算法编码。相反，THASH 和 TTAG 指令提供一个允许软件访问由硬件使用的散列算法的接口。本发明与上述申请在下面一点上相关，即本发明提供一种可能的算法，其可以由 THASH 指令使用。另外，一种可以由 TTAG 指令使用的可能算法在下面公开。

本发明支持在图 5 所示的虚拟寻址模式 501。虚地址 502 是 64 位地址。前 3 位形成一个虚拟区域号码（VRN）503。因此，在任何给定时间可以由一个虚地址指定 8 个区域。使用虚地址 502 的剩余 61 位来寻址每一区域内的存储器，从而为每一区域提供 2^{61} 字节的虚拟存储器。与每一存储器页（诸如页 504）相关的有一个 24 位的区域标识符（RID）。因此，操作系统可以指定直到 2^{24} 个单独的虚地址空间。存储器页可以在大小从 4 千字节到 256 兆字节的范围内变化，其在下面详述。说明虚区域的附加信息可以在 Stephen Burger 等人的美国待审专利申请、名称为“用于虚拟寻址模式中预验证区域的方法和装置”中找到。该申请因此结合在这里作为参考，其转让给和本申请同样的受让人，在 1998 年 10 月 12 日申请，分配有美国序

列号 09／170140。

本发明支持提供两个页表格式的结构。第一格式支持一个基于区域的线性页表，此处将称为“短格式”页表，短格式页表为每一虚区域提供，如图 5 所示。短格式页表是线性的，并为在该区域中的每一转换有一个线性条目。相应地，短格式页表不需要链段，并且短格式页表不包括散列标签条目。

第二格式支持为整个计算机系统的一个单一的大页表，在此处将称为“长格式”页表。长格式页表支持链段，长格式页表条目包括散列标签字段。

图 6 表示一个短格式页表条目 601。注意，短格式页表条目包括单一 64 位字，因此总共大小为 8 字节。短格式条目 601 中的字段在下面的表 1 中说明。

表 1

条目字段	说明
P	预设定位，指示映射的物理页是否实际在存储器内。
Rv	保留。
Ma	存储器属性－说明所映射的物理页的超高速缓冲存储能力，相干性，写策略和推测性。
A	被访问的位－指示如何处理页错误。
D	已修改位－指示如何处理由对该页写数据引起的错误。
Pl	特权级－指示该页的特权级。
Ar	访问权限－页级读、写和执行许可和特权控制。
ppn	物理页号－映射物理地址的最高有效位。取决于在该映射中所用的页大小，忽略某些最小意义的 PPN 位。
ig	可用于操作系统的软件字段，被 CPU 忽略。
ed	例外延迟－指示是否应该延迟一个例外或错误。

注意，短格式条目为在一个区域中的每一页存在，而一个转换的虚页号（vpn ）由在虚散列页表（VHPT）中的该短格式条目的位

置隐含。还应该注意，页大小在一个区域内是常数。因此，页大小可以通过访问与该区域关联的一个配置寄存器的 preferred_page_size 字段得到，其在下面讨论。

图 7 表示一个长格式页表条目 701。注意，长格式条目包括 4 个 64 位字，因此总大小为 32 字节。长格式条目 701 的第一字和短格式条目 601 的相同，因此，第一字中的字段用上述表 1 说明。下面的表 2 说明长格式条目 701 的剩余字段。

表 2

条目字段	说明
rv	保留。
ps	页大小－映射的页大小。对于页大小大于 4K 字节的，忽略 PPN 和 VPN 的低阶位。页大小定义为 2^{PS} 字节。
key	保护键字－唯一标记对一个保护域的转换。
tag	转换标签。该标签与长格式索引结合，用于唯一标识该转换。
ti	标签无效位。指示该标签无效。软件可以使用该位使长格式条目无效。
ig	可用于操作系统的软件字段，被 CPU 忽略。注意，长格式 VHPT 条目的最后 64 位（从偏移＋24 开始）通常由操作系统使用来存储对另一长格式 VHPT 条目的连接，如果两个或者多个虚拟对物理的转换散列到该长格式 VHPT 的同一起始条目话。

注意，对所有虚地址使用单一长格式页表，条目可以链接到一起，对每一页，通常没有起始条目。因此，长格式页表条目包括附加信息，诸如页大小（ps）和标签。VPN 由散列索引和标签唯一表示。

最后，在下面讨论本发明的算法之前，下述字段可用于本发明的算法。注意，这些字段的某些表示可编程变量，它们存储在配置寄存器中，因此可以由在一个特定的计算机系统中的软件执行改变。其它字段表示常数，它们在计算机系统的特定实现中不改变，因此可以硬编码为本发明的算法的特定实现。这些字段示于下面的表 3。

表 3

配置寄存器字段或常数	说明
page_table_format	指示是在使用长格式还是短格式页表。这一可编程字段是全局字段，应用于存储器内的所有页。
preferred_page_size	指定一页中的字节数。页大小编码为 N，其中页大小为 2^N 字节。这一可编程字段可以为每一区域指定。注意，该 preferred_page_size 被复制到每一长格式页表条目的(ps)字段。
page_table_size	这一可编程字段指示在该页表的线性部分内的字节数。在短格式页表中，为每一虚区域提供 page_table_size ，它决定该虚区域的大小，因为短格式页表必须为在该虚区域内的每一页有一个条目。因为短表格式是线性的，不能增长，因此 page_table_size 表示短格式页表的精确大小。在长格式页表中，page_table_size 指示该页表的线性部分的长度，该页表可以随添加链段向更深处增长。页表大小编码为 N，其中该页表的大小为 2^N 字节。
page_table_base	这一可编程字段指示存储器中第一页表条目的地址。当使用短格式页表时，每一虚区域包括它自己的页表并为每一虚区域提供 page_table_base。当使用长格式页表时，提供单一页表，且单一页表基指示第一长格式页表条目的地址。注意，只需要存储位{63:min_pt_size}（见下面）。还要注意，page_table_base 必须位于 $2^{page\_table\_size}$ 边界。
impl_va_msb	这一常数指示由该特定计算机系统支持的虚地址的最高有效位。
min_pt_size	这一常数指示长和短格式页表两者的最小大小（以字节计）。该最小页表大小表示为 N，其中

一个页表的最小大小为 2^N 字节。

如上所述，在基于区域的短格式中，为每一区域的线性页表存在于它自己引用的区域内。其结果，短格式 VHPT 包括单独的每区域页表，其由 page_table_base 的位{60:min_pt_size}固定在每一区域。对在其内允许 VHPT 的区域，需要操作系统来维护一个每区域的线性页表。如在下面的短格式算法中所定义的，使用要被转换的虚地址（VA）、该区域的 preferred_page_size、page_table_base 和 page_table_size 来计算对该短格式 VHPT 的线性索引。

短格式 VHPT 的大小（page_table_size ）定义映射的虚地址空间的大小。在短格式中的最大结构表大小为每区域 2^{52} 字节。为使用 4 千字节的页映射整个区域（2^{61} 字节），必须有 $2^{(61-12)}$（或者另写为，2^{49}）个页是可映射的。一个短格式 VHPT 条目为 8 字节（或者另写为，2^3 字节）大。其结果，最大表大小为每区域 $2^{(61-12+3)}$（或者另写为，2^{52}）字节。如果使用该短格式映射一个小于 2^{61} 的地址空间，则可以使用较小的短格式表（page_table_size <52）。使用 4 千字节的页映射 2^N 的地址空间需要（N－9）的最小 page_table_size 。

当使用短格式 VHPT 时，THASH 指令（上面已说明）返回一个基于区域的短格式索引。TTAG 指令，其也已在上面说明，不用于短格式。在下面的短格式散列算法中，把希望为其得到一个 VHPT 入口地址的虚地址（VA）传送给函数 tlb_vhpt_hash_short 。该函数返回相应于该虚地址的条目的地址（vhpt_addr）。

短格式散列算法

```
1:    tlb_vhpt_hash_short(VA)
2:    {
3:
hash_page_number=VA{impl_va_msb:0}u>>preferred_page_size;
4:    hash_index=hash_page_number;
5:    vhpt_offset=hash_index<<3;
6:    vhpt_region=VA{63:61};
7:    pmask=2^page_table_size-1;
8:    vhpt_addr=(vhpt_region<<61)|
```

```
    9:
(((page_table_base{60:min_pt_size}&~pmask{60:min_pt_size})|
    10:
(vhpt_offset{60:min_pt_size}&pmask{60:min_pt_size}))<<min_p
t_size)|
    11:     vhpt_offset{min_pt_size-1:0};
    12:     return vhpt_addr;
    13:     }
```

在短格式散列算法的第一行，调用叫作函数 tlb_vhpt_hash_short， 虚地址（VA）传送给该函数。在第三行，通过使用 preferred_page_size 除 VA 计算 hash_page_number。注意，只使用由特定计算机系统的一种实现使用的 VA 的那些位（由常数 impl_va_msb 定义）。通过右移 VA N 位实现该除法运算，这里页大小为 2^N。该右移是无符号的。如上所述，在一个实施例中，页大小可以在 4 千字节到 256 兆字节的范围之间变化，所以 VA 将右移 12 到 28 位。

在第 4 行，设定 hash_index 等于 hash_page_number。在该短格式算法中，这一步骤有点冗余，但是包含它进来是为了协调短格式和长格式算法，下面将会看到。如上所述，短格式 VHPT 中的每一条目是 8 字节宽。因此在第 5 行，通过用 8 乘 hash_page_number 计算对该页表的偏移（vhpt_offset ）。这通过左移 hash_index3 位执行。

在第六行，计算 VHPT 的区域（vhpt_region ）。如上所述，当使用短格式 VHPT 时，每一区域包括它自己的 VHPT，所以该 VHPT 的区域和 VA 的区域相同。因此，该 VHPT 的区域只是 VA 的位{63:61} 。

在第七行，通过取 2 的相应于 page_table_size 的位数的幂减 1 形成一个掩码(pmask) 。例如，为使用最小 4 千字节的 preferred_page_size 映射整个区域（2^{61} 字节），必须有 $2^{(61-12)}$（或者另写为：2^{49}）个页可以映射。由于每一短格式 VHPT 条目是 8（或另写为 2^3）字节，因此最大 page_table_size 是 2^{52}。在该第一例子中，pmask 的前 12 位将是“0”，而低 52 位将是“1”。相似地，为使用最大 256 兆字节的 preferred_page_size 映射整个区域（2^{61} 字节），必须有 $2^{(61-28)}$（或者另写为：2^{33}）个页可以映射。由于每一

短格式 VHPT 条目是 8(或另写为 2^3)字节，因此最小 page_table_size（当映射一个完全的区域时）是 2^{36}。在该第二例子中，pmask 的前 28 位将是“0”，而低 36 位将是“1”。当然，还可以映射小于整个 2^{61} 字节的区域，取决于 preferred_page_size，其可以产生 page_table_size 小于 2^{36}。该掩码用于选择形成相应于 VA 的 VHPT 条目的结果地址的分量，其在下面说明。

在第 8－11 行，通过“或”运算一些分量计算相应于该 VA 的 VHPT 的条目的地址（vhpt_addr）。首先，在第 8 行通过左移 vhpt_region 61 位计算该区域分量，从而把 vhpt_region 定位在 vhpt_addr 的合适位置。

在讨论第 9－11 行之前，考虑 min_pt_size 是为计算机系统的每一实现定义的常数。常数 min_pt_size 表示为 N，这里该页表的最小大小是 2^N 字节。因此，总知道 vhpt_addr 的位{min_pt_size-1:0}将由 vhpt_offset 提供。然而，位{60:min_pt_size}根据 page_table_size 既可由 page_table_base 也可由 vhpt_offset 提供。相应地，使用在第七行计算的 pmask 根据 page_table_size 来选择 page_table_base 和 vhpt_offset 的合适位。

定义最小页表大小的确减少（在某种程度上）为实现按照本发明的计算机系统需要的逻辑的数量。例如，存放每一 page_table_base 的寄存器只需要存储位{63:min_pt_size}。另外，下面参考第 9 和第 10 行讨论的“与”和“或”运算的宽度可以减少 min_pt_size 位。在一个实施例中，min_pt_size 是 15，导致 32 千字节的最小页表大小。

相应地，在第 9 行，page_table_base 的位{60:min_pt_size}是和 pmask 的位{60:min_pt_size}的逆的“与”，在第 10 行，vhpt_offset 的位{60:min_pt_size}是和 pmask 的位{60:min_pt_size}的“与”。两个“与”运算的结果“或”在一起，并将其结果左移 min_pt_size 位的位置。相应地，第 9 和第 10 行使用 pmask 和 min_pt_size 形成 vhpt_addr 的分量，其根据 VHPT 的大小变化，且已知不是唯一地由 vhpt_offset 根据 min_pt_size 提供。注意，把这一分量与在第 8 行计算的区域分量进行“或”运算。

最后，在第 11 行，把根据唯一的 vhpt_offset（位

{min_pt_size-:0})的 vhpt_addr 的分量和上面计算的其它两个分量进行“或”运算以形成 vhpt_addr 。在第 12 行，函数 tlb_vhpt_hash_short 结束，并返回 vhpt_addr 给调用子例程。

在短格式 VHPT 中，每一 VHPT 条目唯一相应于一个虚地址。然而，在长格式 VHPT 中，多个虚地址可以共享 VHPT 的一个初始条目，其存储在 VHPT 条目中的后面的转换由操作系统链接到该初始条目。在访问该初始条目后，通过检索该初始和链接的条目找到合适的虚拟对物理的转换而形成相应于该虚拟对物理转换的标签（在图 7 中表示）。长格式算法在下面叙述。注意，为避免混淆，对所有算法使用唯一行号。

长格式散列算法

```
14:     tlb_vhpt_hash_long(VA, region_id)
15:     {
16:
hash_page_number=VA{impl_va_msb:0}u>>preferred_page_size;
17:
hash_index=((VA{63:61}<<52|hash_page_number)^region_id;
18:     vhpt_offset=hash_index<<5;
19:     vhpt_region=page_table_base{63:61};
20:     pmask=2^page_table_size-1;
21:     vhpt_addr=(vhpt_region<<61)|
22:
(((page_table_base{60:min_pt_size}&~pmask{60:min_pt_size})|
23:
(vhpt_offset{60:min_pt_size}&pmask{60:min_pt_size}))<<min_pt_size)|
24:     vhpt_offset{min_pt_size-1:0};
25:     return vhpt_addr;
26:     }
```

在长格式散列算法的第14行，叫作函数 tlb_vhpt_hash_long，虚地址（VA）和 24 位的 region_id 传送给该函数。在第 16 行，通过使用 preferred_page_size 除 VA，计算 hash_page_number。注意，

只使用由特定计算机系统的一种实现使用的 VA 的那些位（由常数 impl_va_msb 定义）。通过右移 VA N 位实现该除法运算，这里页大小为 2^N。该右移是无符号的。如上所述，在一个实施例中，页大小可以在 4 千字节到 256 兆字节的范围之间变化，所以 VA 将右移 12 到 28 位。

在第 17 行，形成 hash_index。如上所述，通过右移 VA 至少 12 位形成该 hash_page_number。因此，散列页号的最大数目是 2^{52}，而 hash_page_number 的位{64:52}为“0”。第 17 行的第一部分左移 VA 的位{63:61}（VA 的区域部分）52 位，并且与 hash_page_number 进行“或”运算。这增加了长格式 VHPT 的最大可能大小，从 2^{52} 个条目（散列页号的最大值）到 2^{55} 个条目。最后，第 17 行的第一部分的结果与 24 位的 region_id 进行“异或”运算而形成 hash_index。

如上所述，长格式 VHPT 条目是 32（或另写为 2^5）字节。因此，在第 18 行通过左移 hash_index5 位的位置形成 vhpt_offset 。在第 19 行，通过检索 page_table_base 的位{63:61}形成 vhpt_region。与存在于每一区域中的短格式 VHPT 不同，为整个系统只定义一个长格式 VHPT。

已经在第 17－19 行计算出 hash_index、vhpt_offset、和 vhpt_region，于是在第 20－24 行计算 pmask 和 vhpt_addr。注意，长格式算法的第 20－24 行和短格式算法的第 7－11 行相同。因此，以和上面参考短格式算法叙述的同样方式形成 vhpt_addr 。最后，在第 25 行，结束函数 tlb_vhpt_hash_long，给调用子例程返回 vhpt_addr。

注意，当使用长格式 VHPT 时，vhpt_addr ，结合存储在长格式 VHPT 条目中的标签一起，唯一标识一个虚拟对物理的转换。下面叙述与长格式散列算法一起保证唯一性的标签算法。

标签算法

```
27:    tag(VA, region_id)
28:    {
29:    pmask=2^page_table_size-1;
30:    tpn=VA & ~pmask;
31:    tag_for_entry=(region_id<<40)|(tpn>>12);
```

```
32      return tag_for_entry;
33:     }
```

按照本发明设计的计算机系统支持长短两种格式的 VHPT。如同上面讨论的，优选用硬件实现长和短格式散列算法。如在本技术中公知，总希望使实现一个特定函数而需要的事务处理数目最小，同时使该函数的执行速度最快。

在检查长和短格式算法时，可以发现在这两种算法之间有许多地方相似。按照本发明，下面提供一种组合长短格式算法。通过把长短格式算法组合，实现两种算法所需要的晶体管数目最少，而不明显影响任一种算法的执行速度。下面叙述组合散列算法:

组合散列算法

```
34:     tlb_vhpt_hash_combined(VA,region_id)
35:     {
36:
hash_page_number=VA{impl_VA_msb:0}u>>preferred_page_size;
37:     if(page_table_format==long){
38:
hash_index=((VA{63:61}<<52)|hash_page_number)^region_id;
39:     vhpt_offset=hash_index<<5;
40:     vhpt_region=pafe_table_base{63:61};
41:     }
42:     else{
43:      hash_index=hash_page_number;
44:      vhpt_offset=hash_index<<3;
45:      vhpt_region=VA{63:61};
46:     }
47:      pmask=2^page_table_size-1;
48:      vhpt_addr=(vhpt_region<<61)|
49:
(((page_table_base{60:min_pt_size}&~pmask{60:min_pt_size})|
50:
(vhpt_offset{60:min_pt_size}&pmask{60:min_pt_size}))<<min_p
```

```
t_size)|
    51:     vhpt_offset{min_pt_size-1:0};
    52:     return vhpt_addr;
    53:     }
```

基本上，组合散列算法组合了长和短格式散列算法的公共元件，而算法的不同部分在一个 IF-THEN-ELSE 框内提供，该框测试 page_table_format 是否设定为“长”格式。因此，在组合散列算法的第 34 行，调用函数 tlb_vhpt_hash_combined，并把虚地址（VA）和 24 位的 region_id 传送给该函数。注意，如果 page_table_format 未设定为“长”格式，则将不使用 region_id。在第 36 行，通过用 preferred_page_size 去除 VA 计算 hash_page_number，如同在短格式散列算法的第 3 行和在长格式散列算法的第 16 行一样。

在第 37 行，测试 page_table_format，看其是否设定为“长”格式。如果是，则分别在第 38、39、和 40 行计算 hash_index，vhpt_offset，和 vhpt_region，如同在长格式散列算法的第 17、18、和 19 行分别进行的那样。如果 page_table_format 未设定为“长”格式，则分别在第 43、44、和 45 行计算 hash_index，vhpt_offset，和 vhpt_region，如同在短格式散列算法的第 4、5、和 6 行分别进行的那样。之后，在第 47－52 行计算 pmask 和 vhpt_addr 并结束该函数（给调用子例程返回 vhpt_addr）。注意，第 47－52 行和短格式散列算法的第 7－12 行相同，也和长格式散列算法的第 20－25 行相同。

因此，本发明提供一种组合散列算法，它既能为短格式 VHPT（每一 VHPT 条目唯一标识一个虚拟对物理的转换）产生索引，也能为长格式 VHPT（每一初始的 VHPT 条目与一个存储的标签唯一标识一个虚拟对物理的转换）产生索引。注意，实现本发明的组合算法的人也会发现在该组合散列算法中另外的公共的地方，它们单独为长格式和短格式执行。例如，在第 39 和 44 行执行的左移可以通过单一移位电路实现，该电路在 page_table_format 设定为“长”格式时另外左移两位的位置。类似地，在第 40 和 45 行对 vhpt_region 的计算可以使用多路转换器根据 page_table_format 从 page_table_base 或 VA 中选择位 63－61。设计逻辑电路实现本发明的组合散列算法的

人也可以认识到使实现本发明所需要的逻辑为最少的其它方式。

虽然参考优选实施例说明了本发明，但是本技术领域熟练的工作人员了解，在形式和细节上可以进行改变而不离开本发明的精神和范围。

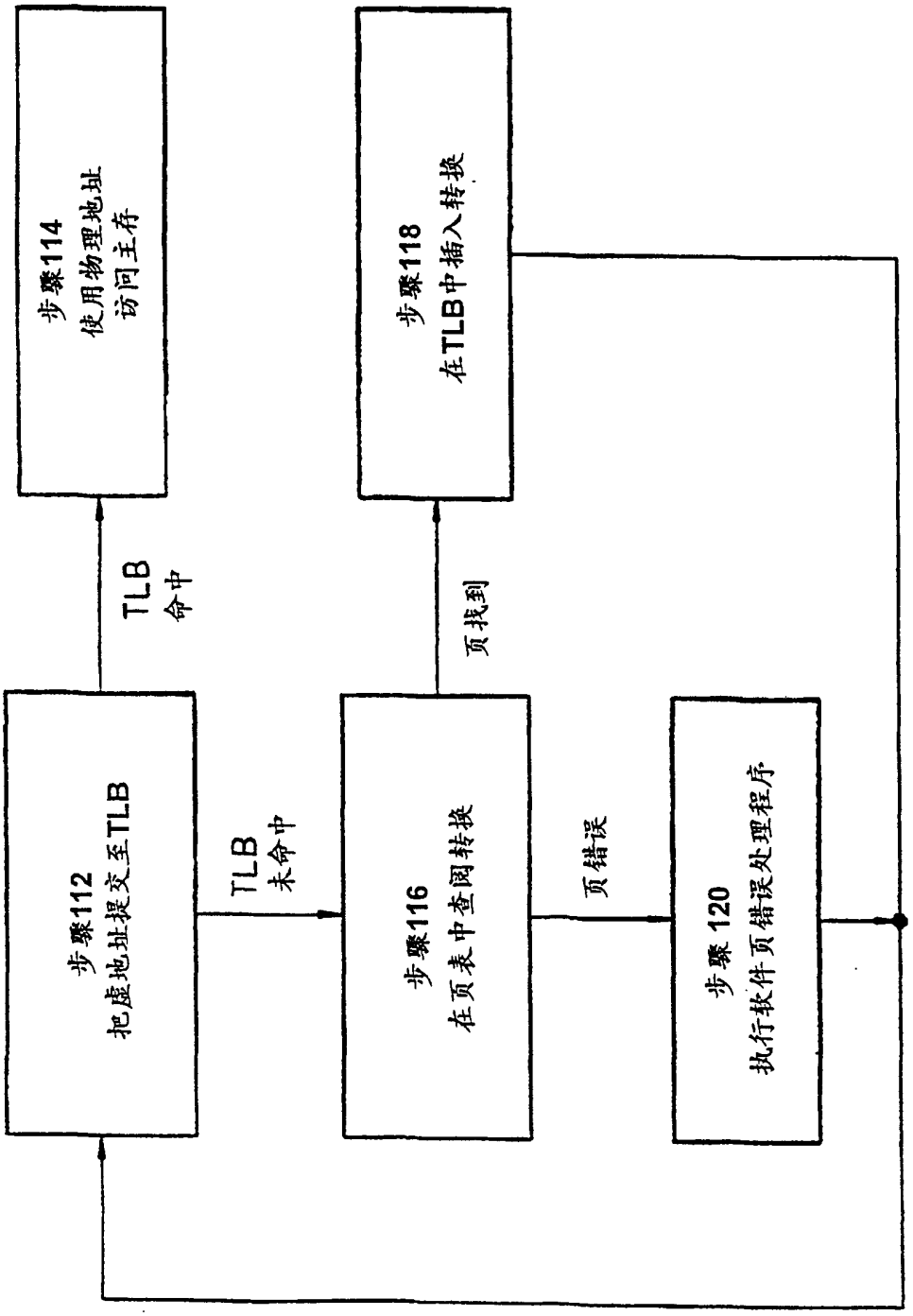

步骤112
把虚地址提交至TLB
TLB
命中
TLB
未命中
步骤114
使用物理地址
访问主存
步骤116
在页表中查阅转换
页找到
步骤118
在TLB中插入转换
页错误
步骤120
执行软件页错误处理程序

图 1 (现有技术)

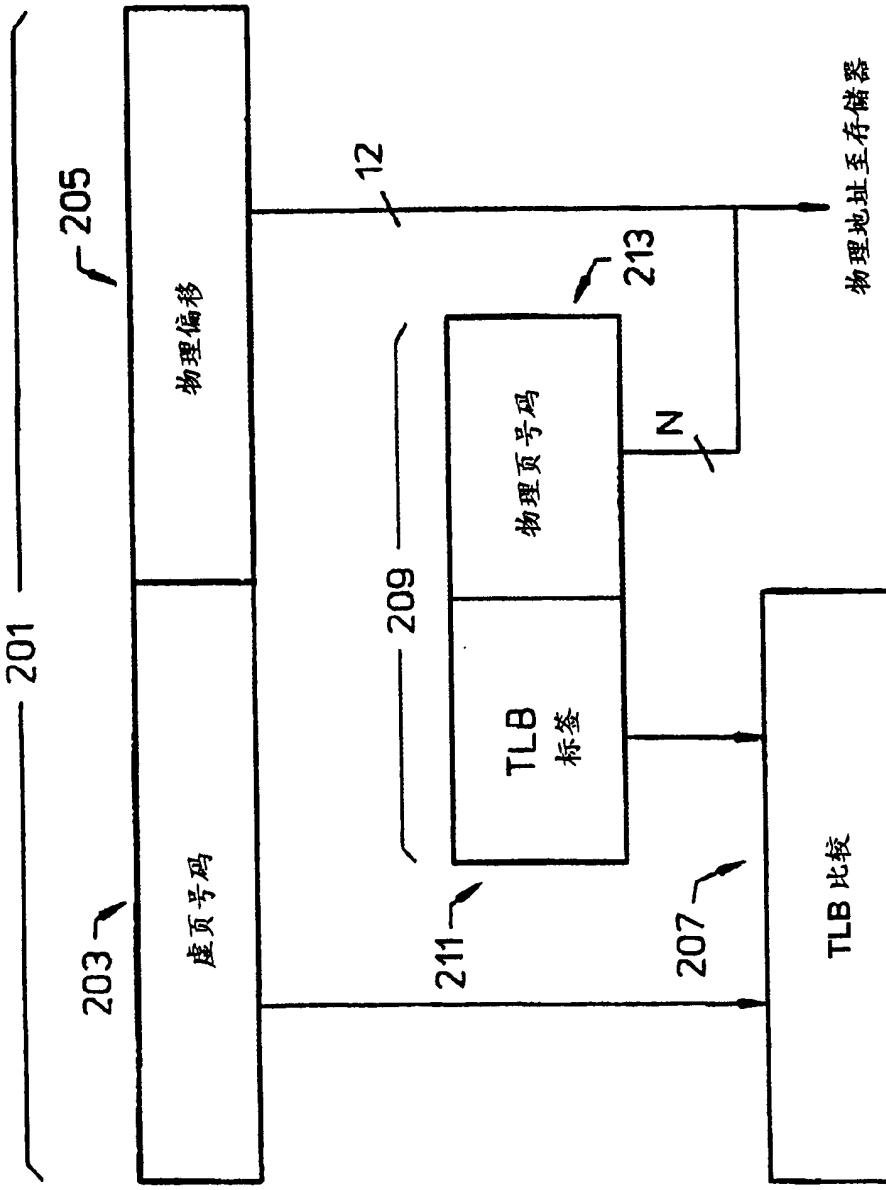

201
205
物理偏移
虚页号码
203
12
物理地址至存储器
213
物理页号码
N
209
TLB
标签
211
207
TLB 比较
TLB 命中

图 2
(现有技术)

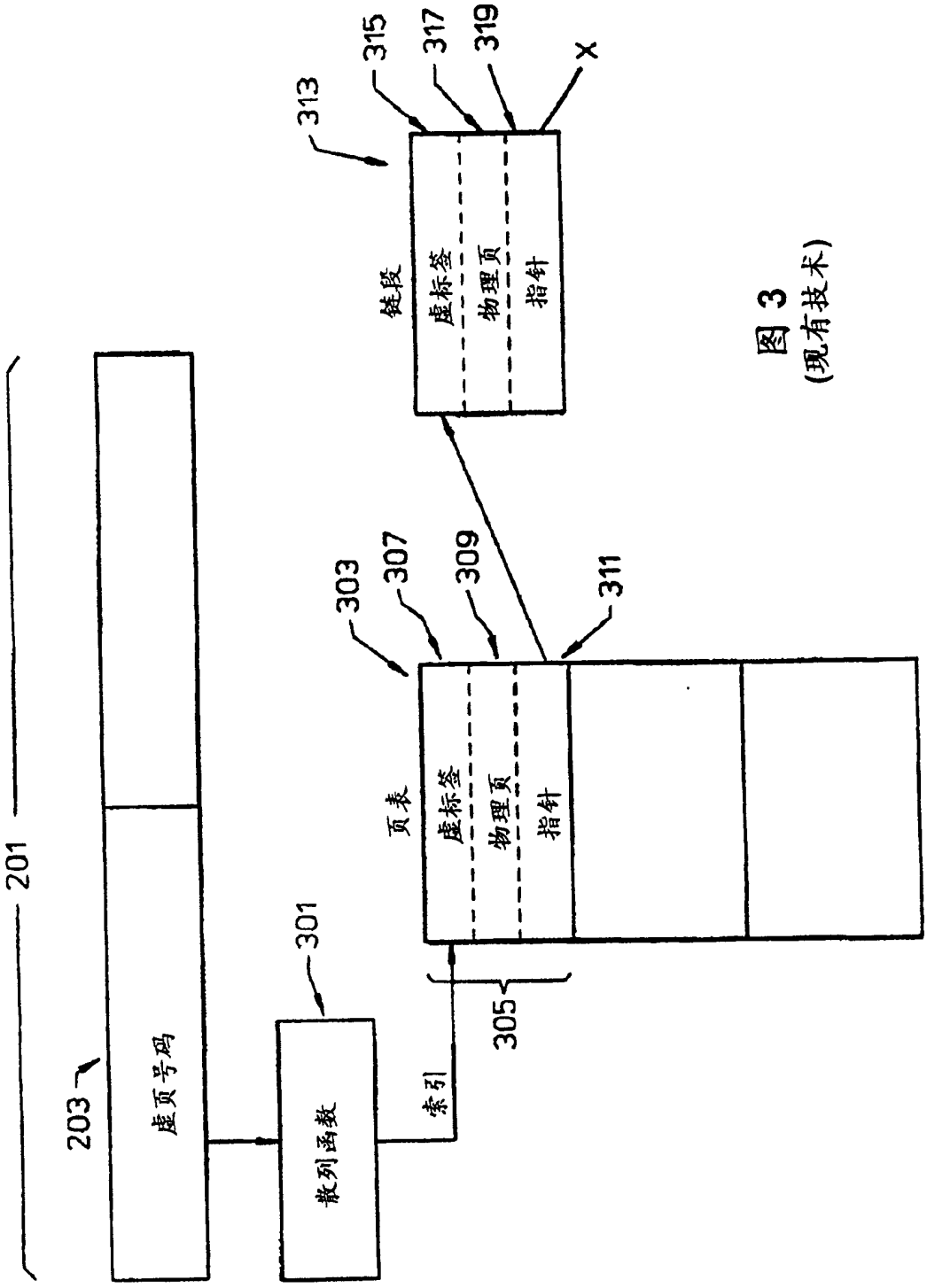

201
203
虚页号码
301
散列函数
索引
305
页表
虚标签
物理页
指针
303
307
309
311
链段
虚标签
物理页
指针
313
315
317
319
X

图 3
(现有技术)

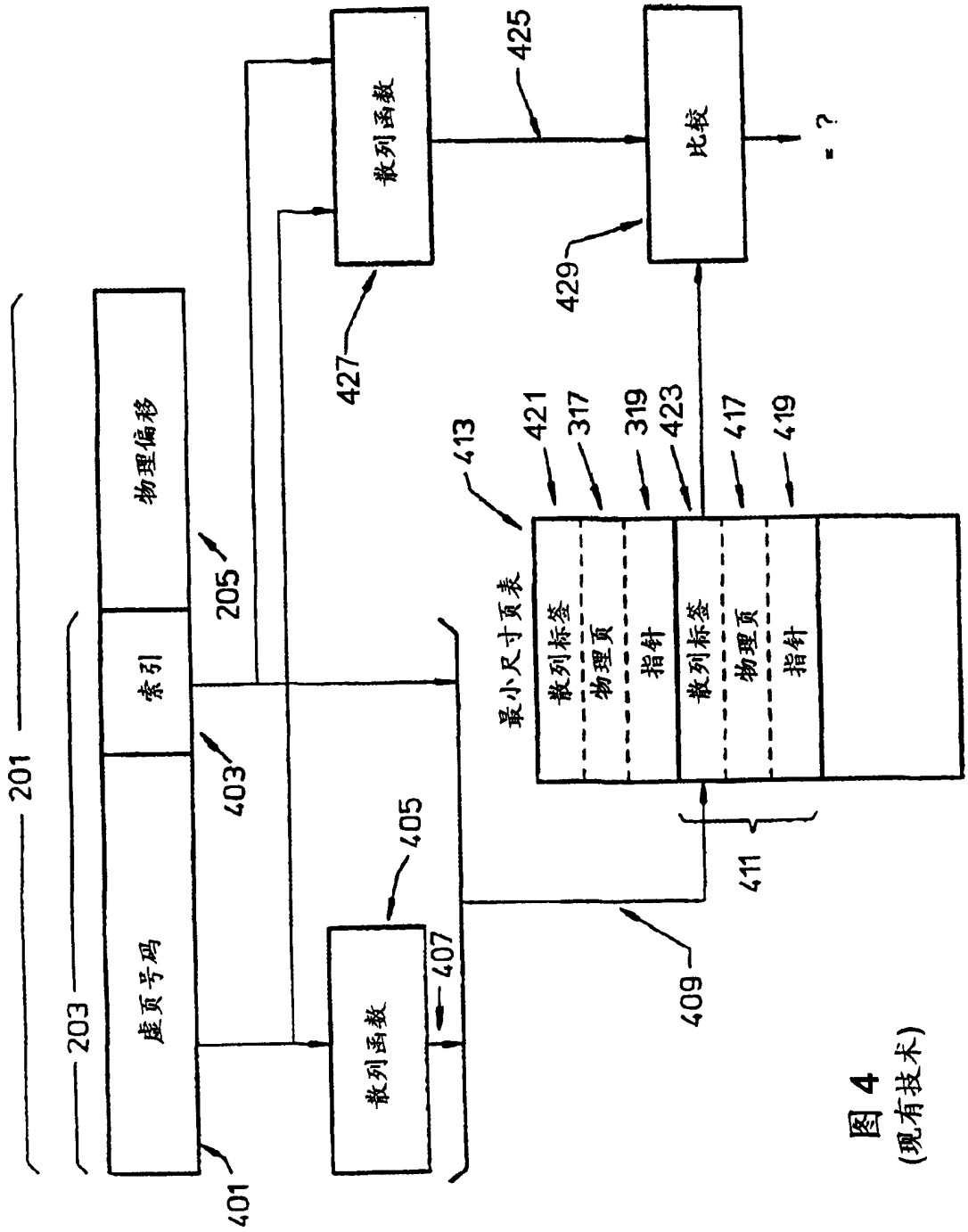

201
203
虚页号码
索引
物理偏移
205
403
401
405
散列函数
407
427
散列函数
425
比较
= ?
429
413
最小尺寸页表
421
317
319
423
417
419
散列标签
物理页
指针
散列标签
物理页
指针
409
411

图 4
(现有技术)

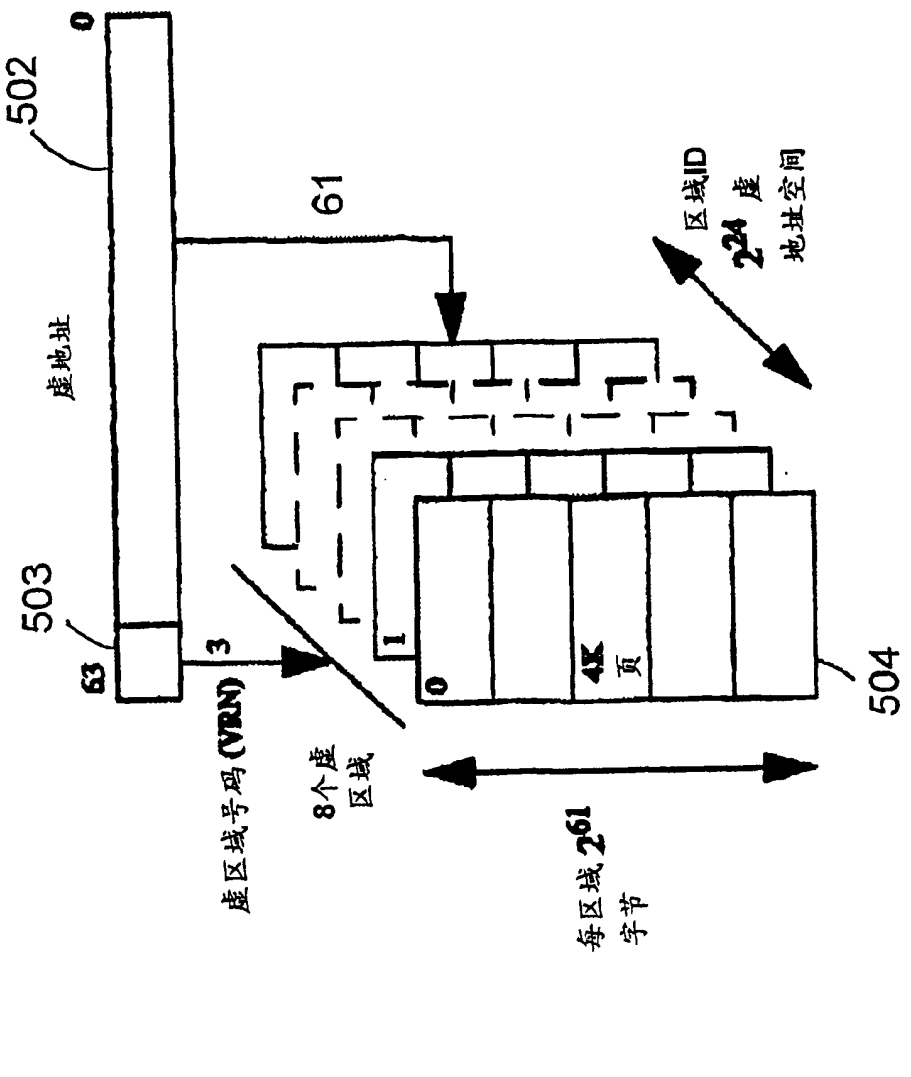

502
0
虚地址
61
区域ID
2^{24} 虚
地址空间
503
63
3
虚区域号码(VRN)
8个虚
区域
1
0
4K
页
504
每区域 2^{61}
字节
501

图 5

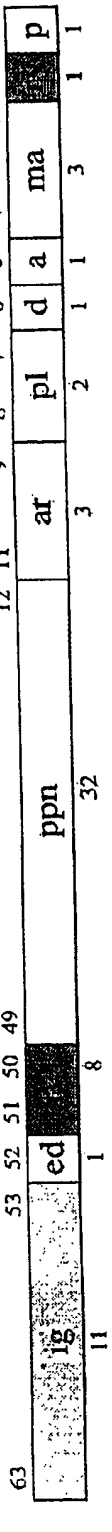

63
53
52
51
50
49
12
11
9
8
7
6
5
4
2
1
0
ig
ed
ppn
ar
pl
d
a
ma
p
11
1
8
32
3
2
1
1
3
1
1
601

图 6

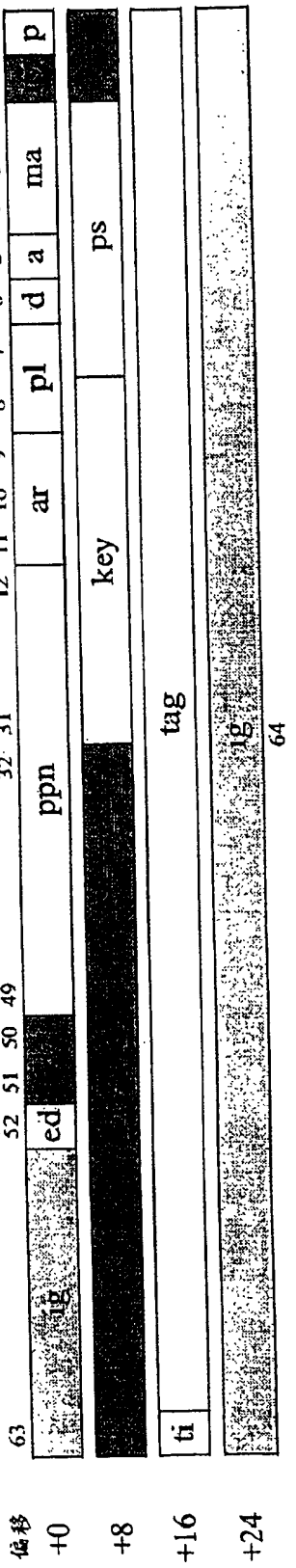

63
52
51
50
49
32
31
12
11
10
9
8
7
6
5
4
3
2
1
0
ig
ed
ppn
ar
pl
d
a
ma
p
key
ps
tag
ti
ig
64
偏移
+0
+8
+16
+24
701

图 7