



(12) 发明专利

(10) 授权公告号 CN 101911025 B

(45) 授权公告日 2012. 11. 07

(21) 申请号 200980101961. 9

(51) Int. Cl.

(22) 申请日 2009. 01. 05

G06F 12/14 (2006. 01)

G06F 12/10 (2006. 01)

(30) 优先权数据

11/972, 688 2008. 01. 11 US

(56) 对比文件

US RE37305 E, 2001. 07. 31, 全文.

US 5008811 A, 1991. 04. 16, 全文.

CN 1896972, 2007. 01. 17, 全文.

(85) PCT申请进入国家阶段日

2010. 07. 09

IBM Corp. Architecture Principles of Operation. 《Architecture Principles of Operation》. 2007, 第 3 章.

(86) PCT申请的申请数据

PCT/EP2009/050050 2009. 01. 05

(87) PCT申请的公布数据

W02009/087133 EN 2009. 07. 16

审查员 吴瑶

(73) 专利权人 国际商业机器公司

地址 美国纽约

(72) 发明人 D·格雷纳 C·小盖尼 L·海勒

D·奥西塞克 E·普费弗

T·什莱格尔 C·韦布

(74) 专利代理机构 北京市中咨律师事务所

11247

代理人 于静 杨晓光

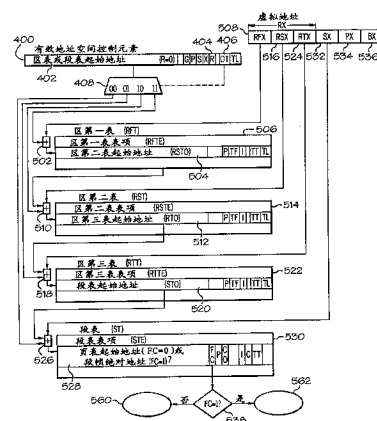
权利要求书 5 页 说明书 36 页 附图 15 页

(54) 发明名称

带有取回保护的动态地址转换的方法和系统

(57) 摘要

提供了一种增强的动态地址转换工具。在一个实施例中, 首先获取要转换的虚拟地址, 并且获取转换层次结构的转换表的初始起始地址。根据所获取的初始起始地址, 获取段表表项。所述段表表项被配置为包含格式控制字段和访问有效性字段。如果已启用所述格式控制字段和访问有效性字段, 则所述段表表项还包含访问控制字段、取回保护字段以及段-帧绝对地址。仅当所述访问控制字段与程序状态字或所执行程序指令的操作数中的任意一个所提供的程序访问键匹配时, 才允许存储操作。当与所述虚拟地址关联的程序访问键与所述段访问控制字段相同时, 允许取回操作。



1. 一种用于保护具有转换表层次结构的计算机系统的数据的方法,所述转换表用于将虚拟地址转换成主存储装置中的数据块的转换后的地址,所述方法包括:

获取要转换的虚拟地址;

获取所述转换表层次结构的转换表的初始起始地址,所述转换表层次结构包括段表;

根据所获取的初始起始地址,从所述段表获取段表表项,所述段表表项被配置为包含格式控制字段;

响应于已启用增强的动态地址转换 DAT 工具,判定是否已启用所述段表表项中的所述格式控制字段;以及

响应于已启用所述格式控制字段,执行以下操作:

判定是否已启用所述段表表项中的访问有效性字段,所述段表表项包括所述访问有效性字段、段访问控制字段、段取回保护字段以及主存储装置中的大数据块的段-帧绝对地址;

响应于已启用所述访问有效性字段,执行以下操作:

允许到所需存储块的存储,其中所述存储与所述虚拟地址关联,仅响应于所述段访问控制字段与程序状态字或所执行程序指令的操作数中的任意一个所提供的程序访问键匹配,才允许所述存储;以及

仅响应于 a) 已禁用所述段取回保护字段,或者 b) 已启用所述段取回保护字段并且与所述虚拟地址关联的所述程序访问键与所述段访问控制字段相同,才允许从所需数据块执行取回,其中所述取回与所述虚拟地址关联;以及

响应于未启用所述增强的动态地址转换 DAT 工具,执行以下操作:

从所述段表表项获取页表的起始地址并使用所述虚拟地址的页索引部分引用所述页表中的页表表项;以及

从所述页表表项获取主存储装置中的小数据块的页-帧实地址,所述小数据块的大小小于所述大数据块并包括 4KB 的数据块,所述小数据块与存储键关联,所述存储键包括存储访问控制字段和存储取回保护字段,其中主存储装置中的所需真实数据块的转换后的地址包括所述页-帧实地址与所述虚拟地址的字节-索引部分的组合。

2. 如权利要求 1 中所述的方法,其中所述转换后的地址是实地址或绝对地址中的任意一个,其中向所述实地址添加前缀以形成所述绝对地址,其中所述转换表层次结构包括区第一表、区第二表、区第三表以及所述段表中的一个或多个,其中从所述段表获取所述段表表项包括 A、B、C、D 中的任意一个:

A) 根据所获取的所述区第一表的初始起始地址,使用所述虚拟地址的第一索引部分引用所述区第一表中包含所述区第二表的起始地址的区第一表表项;根据所述区第二表的所述起始地址,使用所述虚拟地址的第二索引部分引用所述区第二表中包含所述区第三表的起始地址的区第二表表项;根据所述区第三表的所述起始地址,使用所述虚拟地址的第三索引部分引用所述区第三表中包含所述段表的起始地址的区第三表表项;以及根据所述段表的所述起始地址,使用所述虚拟地址的段索引部分引用所述段表中的段表表项;

B) 根据所获取的所述区第二表的初始起始地址,使用所述虚拟地址的第二索引部分引用所述区第二表中包含所述区第三表的起始地址的所述区第二表表项;根据所述区第三表的所述起始地址,使用所述虚拟地址的第三索引部分引用所述区第三表中包含所述段表的

起始地址的所述区第三表表项；以及根据所述段表的所述起始地址，使用所述虚拟地址的所述段索引部分引用所述段表中的所述段表表项；

C) 根据所获取的所述区第三表的初始起始地址，使用所述虚拟地址的第三索引部分引用所述区第三表中包含所述段表的所述起始地址的所述区第三表表项；以及根据所述段表的所述起始地址，使用所述虚拟地址的所述段索引部分引用所述段表中的所述段表表项；以及

D) 根据所获取的所述段表的初始起始地址，使用所述虚拟地址的所述段索引部分引用所述段表表项。

3. 如权利要求 2 中所述的方法，其中响应于已启用所述增强的动态地址转换 DAT 工具以及未启用所述格式控制字段，执行以下操作：

从所述段表表项获取页表的起始地址并使用所述虚拟地址的页索引部分引用所述页表中的页表表项；

从所述页表表项获取主存储装置中的小数据块的页 - 帧实地址，所述小数据块的大小小于所述大数据块并包括 4KB 的数据块，所述小数据块与包括存储访问控制字段和存储取回保护字段的存储键关联，其中主存储装置中的所需真实数据块的转换后的地址包括所述页 - 帧实地址与所述虚拟地址的字节 - 索引部分的组合；

仅响应于所述程序访问键与所述存储访问控制字段匹配，才允许到所需真实存储块的存储，其中所述存储与所述虚拟地址关联，所述程序访问键由包括程序状态字或所执行程序指令的操作数中的任意一个的程序实体提供；以及

仅响应于 a) 已禁用所述存储取回保护字段，或者 b) 已启用所述存储取回保护字段并且与所述虚拟地址关联的程序访问键与所述存储访问控制字段相同，才允许从所需数据块执行取回，其中所述取回与所述虚拟地址关联，所述程序访问键由程序状态字或所执行程序指令的操作数中的任意一个提供。

4. 如权利要求 2 中所述的方法，其中响应于未启用所述增强的动态地址转换 DAT 工具，执行以下操作：

从所述段表表项获取页表的起始地址并使用所述虚拟地址的页索引部分引用所述页表中的页表表项；

从所述页表表项获取主存储装置中的小数据块的页 - 帧实地址，所述小数据块的大小小于所述大数据块并包括与存储键关联的 4KB 的数据块，所述存储键包括存储访问控制字段和存储取回保护字段，其中主存储装置中的所需真实数据块的转换后的地址包括所述页 - 帧实地址与所述虚拟地址的字节 - 索引部分的组合；

仅响应于所述程序访问键与所述存储访问控制字段匹配，才允许到所需真实存储块的存储，其中所述存储与所述虚拟地址关联；以及

仅响应于 a) 已禁用所述存储取回保护字段，或者 b) 已启用所述存储取回保护字段并且与所述虚拟地址关联的程序访问键与所述存储访问控制字段相同，才允许从所需数据块执行取回，其中所述取回与所述虚拟地址关联，所述程序访问键由程序状态字或所执行程序指令的操作数中的任意一个提供。

5. 如权利要求 1 中所述的方法，其中响应于所述程序访问键为零，允许到所述所需存储块的存储以及从所述所需存储块的取回，其中所述存储和取回与所述虚拟地址关联。

6. 如权利要求 1 中所述的方法,其中响应于已启用所述访问有效性字段和所述取回保护字段,并响应于尝试从所述所需存储块执行取回操作以及所述段访问控制字段与所述程序访问键不匹配,指示保护异常,所述取回操作与所述虚拟地址关联。

7. 如权利要求 1 中所述的方法,其中判定是否已启用所述访问有效性字段进一步包括:判定以下项中的任意一个:a) 是否已禁用控制寄存器的存储-保护-忽略字段,或者 b) 是否已启用所述存储-保护-忽略字段但未启用所述段访问控制字段。

8. 如权利要求 1 中所述的方法,还包括:

将在所述虚拟地址的所述转换中使用的信息存储在至少一个转换后备缓冲器中;以及使用来自所述转换后备缓冲器的所存储的信息而不是所述转换表层次结构来执行后续虚拟地址到主存储装置中的所述数据块的所述绝对地址的后续转换。

9. 如权利要求 1 中所述的方法,其中响应于所述转换不是机器体系结构所固有的,还包括:

标识用于仿真所述转换的预定软件例程,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。

10. 如权利要求 1 中所述的方法,其中所述大数据块包括大小至少为 1MB 的块。

11. 一种用于保护具有转换表层次结构的计算机系统的数据的系统,所述转换表用于将虚拟地址转换成主存储装置中的数据块的转换后的地址,所述系统包括:

被配置为获取要转换的虚拟地址的装置;

被配置为获取所述转换表层次结构的转换表的初始起始地址的装置,所述转换表层次结构包括段表;

被配置为根据所获取的初始起始地址、从所述段表获取段表表项的装置,所述段表表项被配置为包含格式控制字段;

被配置为响应于已启用增强的动态地址转换 DAT 工具、判定是否已启用所述段表表项中的所述格式控制字段的装置;以及

被配置为响应于已启用所述格式控制字段、执行以下操作的装置:

判定是否已启用所述段表表项中的访问有效性字段,所述段表表项包括所述访问有效性字段、段访问控制字段、段取回保护字段以及主存储装置中的大数据块的段-帧绝对地址;

响应于已启用所述访问有效性字段,执行以下操作:

允许到所需存储块的存储,其中所述存储与所述虚拟地址关联,

仅响应于所述段访问控制字段与程序状态字或所执行程序指令的操作数中的任意一个所提供的程序访问键匹配,才允许所述存储;以及

仅响应于 a) 已禁用所述段取回保护字段,或者 b) 已启用所述段取回保护字段并且与所述虚拟地址关联的所述程序访问键与所述段访问控制字段相同,才允许从所需数据块执行取回,其中所述取回与所述虚拟地址关联;以及

被配置为响应于未启用所述增强的动态地址转换 DAT 工具、执行以下操作的装置:

从所述段表表项获取页表的起始地址并使用所述虚拟地址的页索引部分引用所述页表中的页表表项;以及

从所述页表表项获取主存储装置中的小数据块的页-帧实地址,所述小数据块的大小

小于所述大数据块并包括 4KB 的数据块,所述小数据块与存储键关联,所述存储键包括存储访问控制字段和存储取回保护字段,其中主存储装置中的所需真实数据块的转换后的地址包括所述页-帧实地址与所述虚拟地址的字节-索引部分的组合。

12. 如权利要求 11 中所述的系统,其中所述转换后的地址是实地址或绝对地址中的任意一个,其中向所述实地址添加前缀以形成所述绝对地址,其中所述转换表层次结构包括区第一表、区第二表、区第三表以及所述段表中的一个或多个,其中被配置为从所述段表获取所述段表表项的装置执行包括 A、B、C、D 操作中的任意一个:

A) 根据所获取的所述区第一表的初始起始地址,使用所述虚拟地址的第一索引部分引用所述区第一表中包含所述区第二表的起始地址的区第一表表项;根据所述区第二表的所述起始地址,使用所述虚拟地址的第二索引部分引用所述区第二表中包含所述区第三表的起始地址的区第二表表项;根据所述区第三表的所述起始地址,使用所述虚拟地址的第三索引部分引用所述区第三表中包含所述段表的起始地址的区第三表表项;以及根据所述段表的所述起始地址,使用所述虚拟地址的段索引部分引用所述段表中的段表表项;

B) 根据所获取的所述区第二表的初始起始地址,使用所述虚拟地址的第二索引部分引用所述区第二表中包含所述区第三表的起始地址的所述区第二表表项;根据所述区第三表的所述起始地址,使用所述虚拟地址的第三索引部分引用所述区第三表中包含所述段表的起始地址的所述区第三表表项;以及根据所述段表的所述起始地址,使用所述虚拟地址的所述段索引部分引用所述段表中的所述段表表项;

C) 根据所获取的所述区第三表的初始起始地址,使用所述虚拟地址的第三索引部分引用所述区第三表中包含所述段表的所述起始地址的所述区第三表表项;以及根据所述段表的所述起始地址,使用所述虚拟地址的所述段索引部分引用所述段表中的所述段表表项;以及

D) 根据所获取的所述段表的初始起始地址,使用所述虚拟地址的所述段索引部分引用所述段表表项。

13. 如权利要求 12 中所述的系统,其中包括,被配置为响应于已启用所述增强的动态地址转换 DAT 工具以及未启用所述格式控制字段、执行以下操作的装置:

从所述段表表项获取页表的起始地址并使用所述虚拟地址的页索引部分引用所述页表中的页表表项;

从所述页表表项获取主存储装置中的小数据块的页-帧实地址,所述小数据块的大小小于所述大数据块并包括 4KB 的数据块,所述小数据块与包括存储访问控制字段和存储取回保护字段的存储键关联,其中主存储装置中的所需真实数据块的转换后的地址包括所述页-帧实地址与所述虚拟地址的字节-索引部分的组合;

仅响应于所述程序访问键与所述存储访问控制字段匹配,才允许到所需真实存储块的存储,其中所述存储与所述虚拟地址关联,所述程序访问键由包括程序状态字或所执行程序指令的操作数中的任意一个的程序实体提供;以及

仅响应于 a) 已禁用所述存储取回保护字段,或者 b) 已启用所述存储取回保护字段并且与所述虚拟地址关联的程序访问键与所述存储访问控制字段相同,才允许从所需数据块执行取回,其中所述取回与所述虚拟地址关联,所述程序访问键由程序状态字或所执行程序指令的操作数中的任意一个提供。

14. 如权利要求 12 中所述的系统,其中包括:被配置为响应于未启用所述增强的动态地址转换 DAT 工具、执行以下操作的装置:

从所述段表表项获取页表的起始地址并使用所述虚拟地址的页索引部分引用所述页表中的页表表项;

从所述页表表项获取主存储装置中的小数据块的页-帧实地址,所述小数据块的大小小于所述大数据块并包括与存储键关联的 4KB 的数据块,所述存储键包括存储访问控制字段和存储取回保护字段,其中主存储装置中的所需真实数据块的转换后的地址包括所述页-帧实地址与所述虚拟地址的字节-索引部分的组合;

仅响应于所述程序访问键与所述存储访问控制字段匹配,才允许到所需真实存储块的存储,其中所述存储与所述虚拟地址关联;以及

仅响应于 a) 已禁用所述存储取回保护字段,或者 b) 已启用所述存储取回保护字段并且与所述虚拟地址关联的程序访问键与所述存储访问控制字段相同,才允许从所需数据块执行取回,其中所述取回与所述虚拟地址关联,所述程序访问键由程序状态字或所执行程序指令的操作数中的任意一个提供。

15. 如权利要求 11 中所述的系统,其中包括被配置为响应于所述程序访问键为零、允许到所述所需存储块的存储以及从所述所需存储块的取回的装置,其中所述存储和取回与所述虚拟地址关联。

16. 如权利要求 11 中所述的系统,其中包括被配置为响应于已启用所述访问有效性字段和所述取回保护字段、并响应于尝试从所述所需存储块执行取回操作以及所述段访问控制字段与所述程序访问键不匹配、指示保护异常的装置,所述取回操作与所述虚拟地址关联。

17. 如权利要求 11 中所述的系统,其中判定是否已启用所述访问有效性字段进一步包括:判定以下项中的任意一个:a) 是否已禁用控制寄存器的存储-保护-忽略字段,或者 b) 是否已启用所述存储-保护-忽略字段但未启用所述段访问控制字段。

18. 如权利要求 11 中所述的系统,还包括:

被配置为将在所述虚拟地址的所述转换中使用的信息存储在至少一个转换后备缓冲器中的装置;以及

被配置为使用来自所述转换后备缓冲器的所存储的信息而不是所述转换表层次结构来执行后续虚拟地址到主存储装置中的所述数据块的所述绝对地址的后续转换的装置。

19. 如权利要求 11 中所述的系统,其中还包括响应于所述转换不是机器体系结构所固有的、执行以下操作的装置:

标识用于仿真所述转换的预定软件例程,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。

20. 如权利要求 11 中所述的系统,其中所述大数据块包括大小至少为 1MB 的块。

带有取回保护的动态地址转换的方法和系统

技术领域

[0001] 本发明一般地涉及用于在计算机系统中转换虚拟地址的系统和方法,更具体地说,涉及在具有动态地址转换工具的计算机系统中将虚拟地址转换成数据块的实地址或绝对地址的系统和方法,其中虚拟地址转换通过转换表层次结构实现。

背景技术

[0002] 动态地址转换提供这样的能力:在任意时刻中断程序的执行,将程序及其数据记录到诸如直接存取存储设备之类的辅助存储装置中,并在以后将程序及其数据返回到其他主存储位置以恢复执行。程序及其数据在主存储装置和辅助存储装置之间的转移可逐步执行,并且可发生将信息返回到主存储装置以响应在执行需要时 CPU 访问该信息的尝试。这些功能可在不更改或不检查程序及其数据的情况下执行,它们不需要重新定位的程序中的任何显式编程约定,并且除了所涉及的时间延迟以外,不会干扰程序的执行。

[0003] 借助操作系统的适当支持,可使用动态地址转换工具为用户提供一种其中存储装置似乎大于配置中可用的主存储装置的系统。此明显的主存储装置通常被称为虚拟存储装置,并且用于指定虚拟存储装置中的位置的地址通常被称为虚拟地址。用户的虚拟存储装置远大于配置中可用的主存储装置的大小并且一般在辅助存储装置中维护。虚拟存储装置被视为包括通常被称为页(也被称为段和区)的数据块。仅将虚拟存储装置的最近被引用的页分配为占用物理主存储装置的数据块。当用户引用的虚拟存储装置的页不在主存储装置中时,将调入这些页以替换主存储装置中引用次数较少的页。在某些情况下,虚拟存储装置被长期(或永久)分配给主存储装置而不考虑是否引用该存储装置。操作系统可在用户不知情的情况下执行存储装置页的调换。

[0004] 程序使用地址(或虚拟地址)访问虚拟存储装置。程序可使用虚拟地址从虚拟存储装置取回指令或从虚拟存储装置加载数据或存储数据。与虚拟存储装置的范围关联的虚拟地址定义了地址空间。借助操作系统的适当支持,可使用动态地址转换工具提供若干地址空间。这些地址空间可用于提供用户间的隔离程度。此类支持可针对每个用户包含完全不同的地址空间,从而提供完全的隔离,或者可通过将每个地址空间的一部分映射到单独的公共存储区域来提供共享区域。另外,还提供了允许半特权程序访问多个此类地址空间的指令。

[0005] 动态地址转换提供了来自多个不同地址空间的虚拟地址的转换。这些地址空间被称为主地址空间、二级地址空间以及存取寄存器指定的地址空间。特权程序还可导致访问起始地址空间。可针对 CPU 生成的指令和数据地址指定动态地址转换。

[0006] 需要一种提供本领域迄今为止尚未了解的附加功能、能力和保护的增强的动态地址转换工具。

发明内容

[0007] 提供了一种用于增强的动态地址转换工具的系统、方法和计算机程序产品,所述

工具将虚拟地址转换成具有机器体系结构的计算机系统的主存储装置中的所需数据块的实地址或绝对地址,所述机器体系结构具有用于转换的转换表层次结构。可向实地址添加前缀以形成绝对地址。

[0008] 在一个示例实施例中,获取要转换的虚拟地址以及所述转换表层次结构的转换表的初始起始地址。所述转换表包括区第一表(region first table)、区第二表、区第三表或段表。根据所获取的初始起始地址,从所述段表获取段表表项。所述段表表项被配置为包含格式控制字段。如果已启用增强的 DAT 工具以及所述格式控制字段,则判定是否已启用所述段表表项中的访问有效性字段。如果已启用所述访问有效性字段,则所述段表表项还包含访问控制字段、保护字段以及主存储装置中的大数据块的段-帧绝对地址。所述大数据块是大小至少为 1MB 的块。如果已启用所述访问控制字段,则仅当所述访问控制字段与程序状态字或者所执行程序指令的操作数所提供的程序访问键(program access key)匹配时,才允许到所需数据块的存储操作。仅当与所述虚拟地址关联的程序访问键与所述访问控制字段相同时,才允许从所需数据块执行取回操作。

[0009] 在另一实施例中,如果已启用所述访问有效性字段,则响应于在所需存储块处尝试取回操作或存储操作并且段访问控制字段与程序访问控制字段不匹配,指示保护异常。

[0010] 下面将结合示出的特定实施例对本发明进行描述。应该理解,本领域的技术人员可以在不偏离本发明的精神或范围的情况下做出各种更改和修改。

附图说明

[0011] 附图连同下面的详细说明一起被结合到说明书中或构成说明书的一部分,用于全部根据本发明进一步示出各种实施例和解释各种原理和优点,在所述附图中,相同的标号在各个独立的视图之间表示相同或功能相似的元素,这些附图是:

[0012] 图 1 示出了其中执行增强的动态地址转换的主计算机系统的一个实施例;

[0013] 图 2 提供了仿真具有主机体系结构的主计算机系统的示例仿真主计算机系统;

[0014] 图 3 示出了如何使用程序状态字针对虚拟地址的动态地址转换确定有效 ASCE 的一个实施例;

[0015] 图 4 示出了其中使用在图 3 中确定的有效 ASCE 来确定在虚拟地址转换中使用的转换表层次结构中的最高转换表的一个实施例;

[0016] 图 5A 示出了使用转换表层次结构使虚拟地址被动态地址转换到段表级别的过程的一个实施例;

[0017] 图 5B 示出了图 5A 中的动态地址转换的后续过程,其中段表表项(STE)格式控制(FC)为 0;

[0018] 图 5C 示出了图 5A 中的动态地址转换的后续过程,其中段表表项(STE)格式控制(FC)为 1;

[0019] 图 6 示出了增强的动态地址转换(eDAT)获取段表表项中的格式控制字段的一个实施例的流程图;

[0020] 图 7 示出了从图 6 的节点 614 开始的流程图的后续过程;

[0021] 图 8 示出了从图 6 的节点 616 开始的流程图的后续过程;

[0022] 图 9 示出了确定要应用于由转换后的虚拟地址寻址的所需数据块的 DAT 保护级别

的一个实施例的流程图；

[0023] 图 10 示出了从图 8 的节点 822 开始的流程图的后续过程，其确定要应用于由转换后的虚拟地址寻址的所需数据块的访问控制保护级别；

[0024] 图 11 示出了从图 8 的节点 822 开始的流程图的后续过程，其确定要应用于由转换后的虚拟地址寻址的所需数据块的取回保护级别；

[0025] 图 12 示出了从图 8 的节点 822 开始的流程图的后续过程，其中从段表表项获取更改记录忽略 (override) 字段；以及

[0026] 图 13 示出了实地址和绝对地址之间的关系。

具体实施方式

[0027] 应该理解，本申请的说明书中做出的陈述并非一定限制各种要求保护的发明中的任何发明。而且，某些陈述适用于某些发明特性，但不适用于其他发明特性。除非另外指出，否则在不失一般性的情况下，单数元素可以表示复数元素，并且反之亦然。

[0028] 本领域中的技术人员很容易熟悉在计算环境中为存储装置寻址以及使用寄存器或地址字段中的位指示不同的状态以及针对这些状态执行操作。进而，本领域中的技术人员了解计算机程序领域并且了解计算机系统组件之间的工作方式和相互关系。

[0029] 概述

[0030] 提供了增强的动态地址转换 (DAT) 工具的示例实施例。当安装并启用增强的 DAT 工具之后，DAT 转换可产生由段表表项中的段表表项 (STE) 格式控制确定的页帧实地址或段帧绝对地址。如在此使用的，术语“应用增强的 DAT”表示满足下面所有条件：1) 已安装 EDAT 工具；2) 已通过控制寄存器 0 (CR0) 位 40 启用 EDAT 工具；以及 3) 借助 DAT- 表表项转换地址。

[0031] 应用增强的 DAT 之后，可以在 DAT 过程中提供以下附加功能：

[0032] 将 DAT 保护位添加到区表表项，提供与段表表项和页表表项中的 DAT 保护位类似的功能。

[0033] 将 STE 格式控制添加到段表表项。当 STE 格式控制为 0 时，DAT 按照当前定义的那样执行，除非页表表项中的更改记录忽略指示对于该页可绕过是否设置更改位。

[0034] 当 STE 格式控制为 1 时，段表表项还包含以下内容：

[0035] 段帧绝对地址（而非页表起始地址），其指定 1MB 块的绝对存储位置。

[0036] 访问控制位和一个取回保护位，其可选地可代替段的单个存储键中的对应位而使用。

[0037] 一个位，其确定段表表项中的访问控制位和取回保护位的有效性。

[0038] 更改记录忽略，其指示在段的单个存储键中是否可绕过更改位设置。

[0039] 主计算机系统

[0040] 参考图 1，示出了主计算机系统 100 的代表组件。也可以在计算机系统使用其他组件布置，这在本领域中是公知的。

[0041] 主计算环境优选地基于由位于纽约阿蒙克的国际商业机器公司 (IBM®) 所提供的 **z/Architecture®**。z/Architecture® Principles of Operation(

z/Architecture[®] 操作原理, **IBM[®]** 出版号:SA22-7832-05, 第六版, 2007 年 4 月) 中更全面地描述了 **z/Architecture[®]**, 其全部内容在此引入作为参考。基于 **z/Architecture[®]** 的计算环境例如包括 eServer 和 **zSeries[®]**, 均由 **IBM[®]** 提供。

[0042] 代表性主计算机 100 包括一个或多个 CPU 101, 其与主存储装置 (计算机存储器 102) 通信以及与存储设备 111 和网络 110 的 I/O 接口通信以实现与其他计算机或存储区域网络 (SAN) 等通信。CPU 可具有用于将程序地址 (虚拟地址) 转换为存储器的实地址的动态地址转换 (DAT) 工具 (功能或单元) 103。DAT 工具通常包括用于对转换进行缓存的转换后备缓冲器 107, 这样, 稍后对计算机存储器 102 的块的访问没有地址转换延迟。通常在计算机存储器 102 和处理器 101 之间使用高速缓存 109。高速缓存 109 可具有层次结构, 其中较大的高速缓存可用于多个 CPU, 而较小、较快 (较低级别) 的高速缓存位于较大的高速缓存和每个 CPU 之间。在某些实施方式中, 将分割较低级别的高速缓存以针对指令取回和数据访问提供单独的低级别高速缓存。在一个实施例中, 指令取回单元 104 通过高速缓存 109 从存储器 102 取回指令。所述指令在指令解码单元 (106) 中进行解码并被分派到 (在某些实施例中, 与其他指令一起被分派到) 指令执行单元 108。通常采用若干执行单元 108, 例如算术执行单元、浮点执行单元和分支指令执行单元。执行单元执行指令, 根据需要访问指令指定的寄存器或存储器中的操作数。如果要从存储器 102 访问 (加载或存储) 操作数, 则通常由加载存储单元 105 在被执行的指令的控制下处理访问。

[0043] 在一个实施例中, 本发明通过软件 (有时称为许可内码 (LIC)、固件、微代码、毫代码、微微代码等, 它们均与本发明一致) 实现。体现本发明的软件程序代码通常由计算机系统 100 中被称为 CPU (中央处理单元) 101 的处理器从诸如 CD-ROM 驱动器、磁带或硬盘驱动器之类的长期存储介质 111 进行访问。所述软件程序代码可体现在诸如磁盘、硬盘驱动器或 CD-ROM 之类的各种已知介质上以便与数据处理系统一起使用。所述代码可通过此类介质分发, 也可从计算机存储器 102 分发给用户, 或通过网络 110 从某个计算机系统的存储装置分发给其他计算机系统以供此类其他系统的用户使用。

[0044] 备选地, 所述程序代码可以位于存储器 102 中, 由处理器 101 使用处理器总线进行访问。此类程序代码包括控制各种计算机组件的功能及交互以及一个或多个应用程序的操作系统。程序代码一般从密集存储介质 111 分页到高速存储器 102, 在高速存储器 102 处, 此类程序代码可由处理器 101 进行处理。将软件程序代码包含在存储器、物理介质中以及/或者通过网络分发软件代码的技术和方法是公知的, 此处不做进一步讨论。当创建程序代码并将其存储在有形介质 (包括但不限于电子存储器模块 (RAM)、闪存、光盘 (CD)、DVD、磁带等) 上时, 通常被称为“计算机程序产品”。计算机程序产品介质一般可由优选地在计算机系统内的处理电路读取以便由所述处理电路进行处理。

[0045] 在图 2 中, 提供了仿真主机体系结构的主计算机系统 100 的示例仿真主计算机系统 201。在仿真主计算机系统 201 中, 主机处理器 (CPU) 208 为仿真主机处理器 (或虚拟主机处理器) 并包括仿真处理器 207, 仿真处理器 207 具有的本机指令集体系结构不同于主计算机 100 的处理器 101 使用的指令集体系结构。仿真主计算机系统 201 包括可由仿真处理器 207 访问的存储器 202。在示例实施例中, 存储器 202 被划分成主计算机存储器 102 部分和仿真例程 203 部分 (提供仿真的例程可为主机存储器的一部分)。根据主计算机体系结构, 主计算机存储器 102 可供仿真的主计算机 201 的程序使用。仿真处理器 207 执行体系

结构不同于被仿真的处理器 208 的架构指令集的本机指令（从仿真例程存储器 203 获取所述本地指令），并可通过使用在序列与访问 / 解码例程中获取的一个或多个指令来从主计算机存储器 102 中的程序访问主机指令以便执行，所述序列与访问 / 解码例程可解码所访问的主机指令（多个）以确定用于仿真所访问的主机指令的功能的本机指令执行例程。

[0046] 其他针对主计算机系统 100 体系结构定义的工具可以由架构的工具例程进行仿真，其中包括诸如通用寄存器、控制寄存器、动态地址转换和 I/O 子系统支持以及处理器高速缓存之类的工具。仿真例程还可利用仿真处理器 207 中可提供的功能（例如通用寄存器和虚拟地址的动态转换）以提高自身的性能。还可提供专用硬件和卸载引擎来帮助处理器 207 仿真主计算机 100 的功能。

[0047] 计算机处理器和寄存器

[0048] 在一个实施例中，CPU 的程序指令功能通过通信总线与多个寄存器进行通信。所述通信总线可以是 CPU 的内部总线，也可以是其外部总线。某些寄存器可以是只读的。其他硬件和 / 或软件也可以对 CPU 所访问的一个或多个寄存器执行读 / 写操作。指令操作码 (opcode) 确定在任意特定机器指令操作中使用哪种类型的寄存器。

[0049] 通用寄存器

[0050] 指令可以在 16 个通用寄存器中的一个或多个寄存器中指定信息。所述通用寄存器可用作地址算术中的基址寄存器和索引寄存器以及用作一般算术和逻辑操作中的累加器。每个寄存器都包含 64 个位位置。通用寄存器通过数字 0-15 来识别，并且由指令中的 4 位 R 字段指定。某些指令通过具有若干 R 字段来提供对多个通用寄存器的寻址。对于某些指令，将隐含使用特定的通用寄存器，而不是由指令的 R 字段显式地指定。

[0051] 对于某些操作，将两个相邻通用寄存器的位 32-63 或位 0-63 相连，从而分别提供 64 位或 128 位格式。在这些操作中，程序必须指定偶数编号的寄存器，其包含最左侧的（高阶）32 或 64 个位。下一较高编号的寄存器包含最右侧（低阶）的 32 或 64 个位。除了在一般算术和逻辑操作中用作累加器之外，16 个通用寄存器中的 15 个通用寄存器还被用作地址生成中的基址寄存器和索引寄存器。在这些情况下，寄存器由指令中的 4 位 B 字段或 X 字段指定。B 或 X 字段中的零值指定不应用任何基址或索引，因此，通用寄存器 0 不能被指定为包含基址或索引。

[0052] 控制寄存器

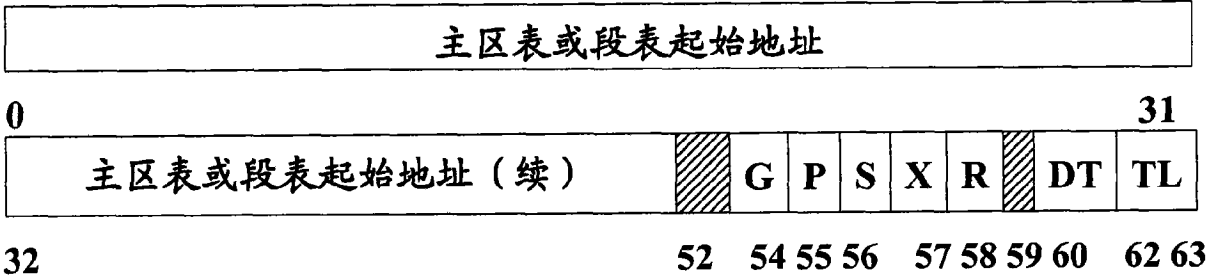
[0053] 控制寄存器提供程序状态字以外的控制信息的维护和操纵。CPU 具有 16 个控制寄存器，每个控制寄存器都具有 64 个位位置。所述寄存器中的位位置被分配给系统中诸如程序事件记录之类的特定工具，并用于指定操作可以发生或用于提供工具所需的特殊信息。控制寄存器通过数字 0-15 来识别并且由指令 LOAD CONTROL 和 STORE CONTROL 中的 4 位 R 字段指定。可以通过这些指令寻址多个控制寄存器。

[0054] 控制寄存器 1

[0055] 控制寄存器 1 包含主地址空间控制元素 (PASCE)。在一个实施例中，控制寄存器 1 具有以下两种格式之一，具体取决于寄存器中的实空间控制位 (R)：

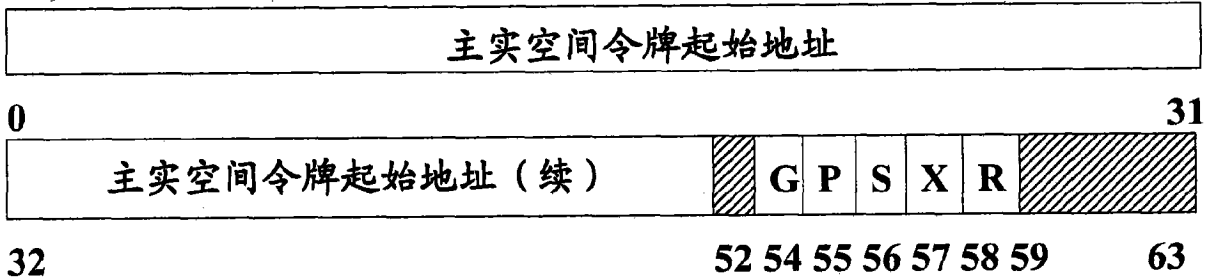
[0056] 主区表或段表指定 (R = 0)

[0057]



[0058] 主实空间指定 (R = 1)

[0059]



- [0060] 控制寄存器 1 的格式
- [0061] 主地址空间控制元素 (PASCE) 中的选定字段按如下方式进行分配：
- [0062] 主区表或段表起始地址：控制寄存器 1 中的主区表或段表指定的位 0-51 及右侧附加的 12 个 0 形成指定主区表或段表的开头的 64 位地址。无法预测此地址是实地址还是绝对地址。该表被称为主区表或段表，因为它用于在主地址空间中转换虚拟地址。
- [0063] 主实空间控制 (R)：如果控制寄存器 1 的位 58 为 0，则所述寄存器包含区表或段表指定。如果位 58 为 1，则所述寄存器包含实空间指定。当位 58 为 1 时，在转换对主地址空间的引用时，即使控制寄存器 1 中的令牌起始地址与转换后备缓冲器表项中的起始地址匹配，段表表项的转换后备缓冲器表示中的公共段位的 1 值也会阻止使用表项以及它所指定的转换后备缓冲器页表副本。
- [0064] 主指定类型控制 (DT)：当 R 为 0 时，控制寄存器 1 中的表指定类型由所述寄存器中的位 60 和 61 指定，如下所示：
- [0065] 位 60
- [0066] 和 61 指定类型
- [0067] 11 区第一表
- [0068] 10 区第二表
- [0069] 01 区第三表
- [0070] 00 段表
- [0071] 主指定类型 (DT) 控制位
- [0072] 当 R 为 0 时，在尝试使用 PASCE 转换虚拟地址（其中最左侧的一个位在地址的位置 0-10 中）时，位 60 和 61 必须为二进制 11。类似地，在最左侧的一个位在地址的位置 11-21 中时，位 60 和 61 必须为二进制 11 或 10，并且在最左侧的一个位在地址的位置 22-32 中时，它们必须为二进制 11、10 或 01。否则，将识别到 ASCE- 型异常。
- [0073] 主区表或段表长度 (TL)：控制寄存器 1 中的主区表指定或段表指定的位 62 和 63

以 4,096 字节为单位指定主区表或段表的长度,从而使区表或段表的长度以 512 个表项的倍数来变化。以 4,096 字节为单位的主区表或段表的长度大于 TL 值。长度字段的内容用于确定通过表转换的虚拟地址部分 (RFX、RSX、RTX 或 SX) 是否指定落入表中的表项。

[0074] 主实空间令牌起始地址:控制寄存器 1 中的主实空间指定的位 0-51 以及右侧附加的 12 个 0 形成 64 位地址,该 64 位地址可以在形成和使用提供虚等于实 (virtual equals real) 转换以引用主地址空间的转换后备缓冲器表项中使用。尽管该地址仅用作令牌而不用于执行存储引用,但是其仍必须为有效地址;否则,当使用控制寄存器 1 的内容时,可能使用不正确的转换后备缓冲器表项。

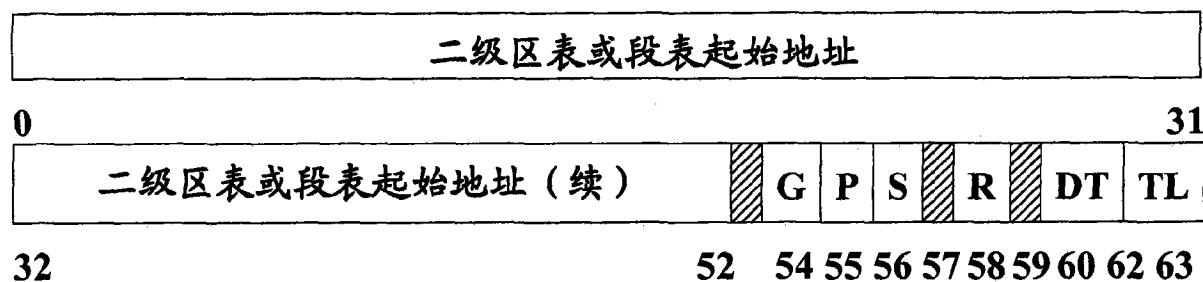
[0075] 将不分配并忽略控制寄存器 1 的以下位:当寄存器包含区表指定或段表指定时不分配并忽略位 52、53 和 59,当寄存器包含实空间指定时不分配并忽略位 52、53 以及 59-63。

[0076] 控制寄存器 7

[0077] 控制寄存器 7 包含二级地址空间控制元素 (SASCE)。在一个实施例中,控制寄存器 7 具有以下两种格式之一,具体取决于寄存器中的实空间控制位 (R):

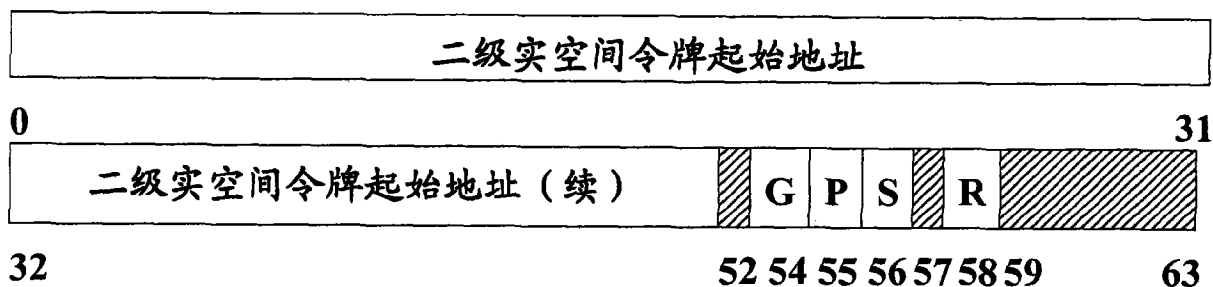
[0078] 二级区表或段表指定 ($R = 0$)

[0079]



[0080] 二级实地址指定 ($R = 1$)

[0081]



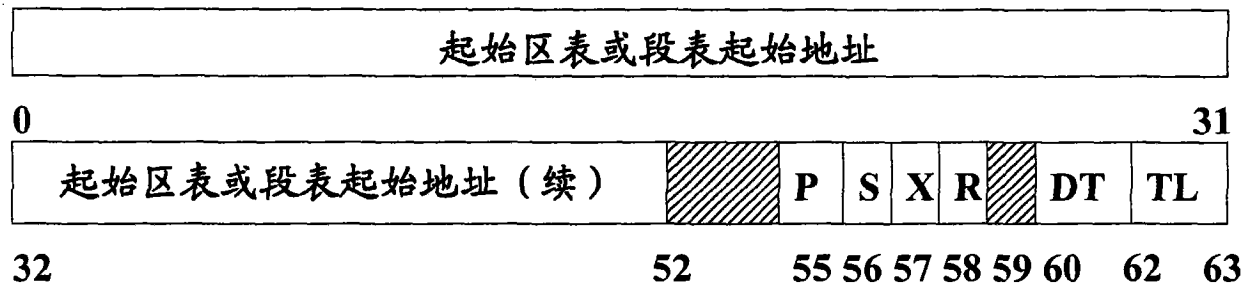
[0082] 控制寄存器 7 的格式

[0083] 控制寄存器 13

[0084] 控制寄存器 13 包含起始地址空间控制元素 (HASCE)。在一个实施例中,控制寄存器 13 具有以下两种格式之一,具体取决于寄存器中的实空间控制位 (R):

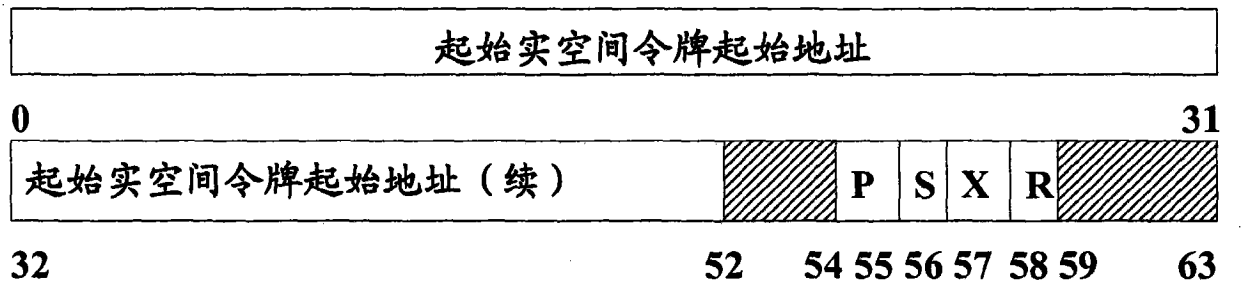
[0085] 起始区表或段表指定 ($R = 0$)

[0086]



[0087] 起始实空间指定 (R = 1)

[0088]



[0089] 控制寄存器 13 的格式

[0090] 存取寄存器

[0091] CPU 具有 16 个以 0-15 进行编号的存取寄存器。存取寄存器包括 32 个位位置,所述位位置包含对 ASCE 的间接指定。ASCE 是动态地址转换 (DAT) 机制在转换对相应地址空间的引用时所使用的参数。当 CPU 处于被称为存取寄存器模式 (由程序状态字中的位控制) 的模式下时,用于指定存储操作数引用的逻辑地址的指令 B 字段将指定存取寄存器,并且 DAT 针对所做的引用使用由存取寄存器指定的 ASCE。对于某些指令,将使用 R 字段,而非 B 字段。提供指令以加载和存储存取寄存器的内容以及将某个存取寄存器的内容移至另一存取寄存器。

[0092] 存取寄存器 1-15 中的每个都指定任意地址空间,其中包括当前指令空间 (主地址空间)。存取寄存器 0 指定主指令空间。当使用存取寄存器 1-15 之一指定地址空间时,CPU 通过转换存取寄存器的内容来确定指定哪个地址空间。当使用存取寄存器 0 指定地址空间时,CPU 将存取寄存器视为指定主指令空间,并且它不会检查存取寄存器的实际内容。因此,16 个存取寄存器在任意时刻可以指定主指令空间以及最多 15 个其他空间。

[0093] 程序状态字 (PSW)

[0094] 程序状态字包括指令地址、条件码以及其他用于控制指令排序及确定 CPU 状态的信息。活动的或执行控制的程序状态字被称为当前程序状态字。它管理当前执行的程序。

[0095] CPU 具有中断功能,此功能允许 CPU 迅速切换到另一程序以响应异常状况和外部激励。当中断发生时,CPU 针对特定的中断类别,将当前程序状态字置于被称为旧程序状态字位置的指定存储位置。CPU 从第二指定存储位置取回新的程序状态字。此新的程序状态字确定下一要执行的程序。完成中断处理之后,处理中断的程序会重新加载旧程序状态字,使其重新成为当前程序状态字,以便继续被中断的程序。

[0096] 存在六个中断类别:外部、I/O、机器检查、程序、重新启动以及系统管理程序调用。每个类别都具有在实际存储装置中永久指定的一对独特的旧程序状态字和新程序状态字

位置。

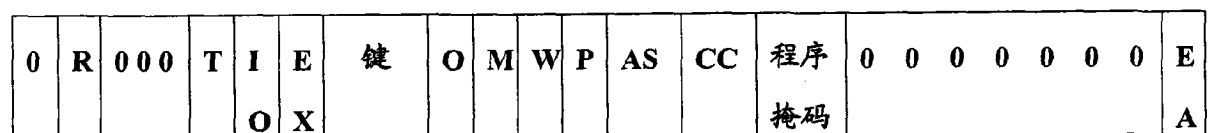
[0097] 当前程序状态字

[0098] CPU 中的当前程序状态字包含执行当前活动的程序所需的信息。所述程序状态字的长度为 128 位,其中包括指令地址、条件码以及其他控制字段。一般而言,程序状态字用于控制指令排序以及用于保存和指示 CPU 中与当前执行的程序有关的多数状态。其他控制和状态信息包含在控制寄存器和永久指定的存储位置中。可通过加载新的程序状态字或部分程序状态字来更改 CPU 的状态。

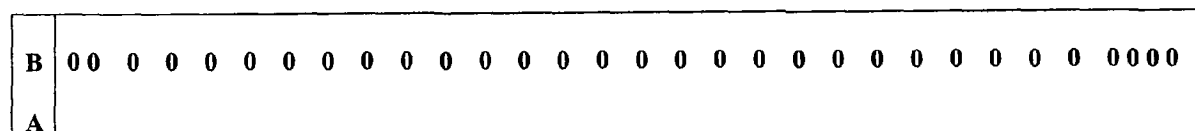
[0099] 通过存储当前程序状态字以保留 CPU 的状态,然后加载新的程序状态字,在 CPU 中断期间切换控制。执行 LOAD PSW 或 LOAD PSW EXTENDED,或成功地结束初始程序加载序列将引入新的程序状态字。指令地址通过顺序指令执行来更新,并且由成功的分支进行替换。提供了对程序状态字的一部分执行操作的其他指令。

[0100] 当更改程序状态字的指令中断或执行完成时,新的或修改后的程序状态字将变为活动(即,引入到当前程序状态字内的信息开始控制CPU)。与更改程序状态字的指令关联的程序事件记录(PER)在PER掩码的控制下发生中断,所述掩码在操作开始时生效。程序状态字的0-7位统称为系统掩码。在一个实施例中,程序状态字具有以下格式:

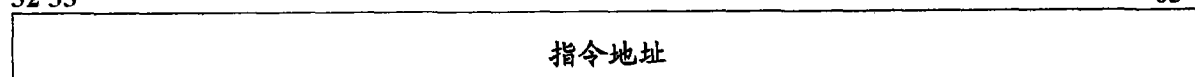
[0101]



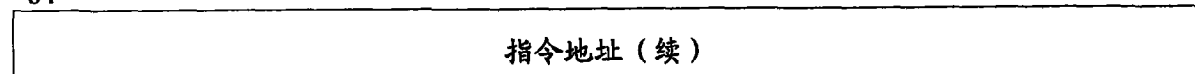
0 1 2 5 6 7 8 12 13 14 15 16 18 20 24 31



32 33 63



64 95



96 127

[0102] 程序状态字格式

[0103] 下面是对选定程序状态字字段功能的简要描述。

[1014] DAT 模式 (T) :位 5 控制是否发生用于访问存储装置的逻辑和指令地址的隐式动态地址转换。当位 5 为 0 时,DAT 处于关闭状态并且逻辑和指令地址被视为实地址。当位 5 为 1 时,DAT 处于打开状态并且调用动态地址转换机制。

[0105] PSW 键:位 8-11 形成 CPU 引用存储装置时使用的访问键。如果引用受到键控保护,则当存储信息或从具有取回保护的位置取回信息时,会将 PSW 键与存储键进行匹配。但是,

对于 MOVE TO PRIMARY、MOVE TO SCNDARY、MOVE WITH KEY、MOVE WITH SOURCE KEY 以及 MOVE WITH DESTINATION KEY 中的每个的操作数之一而言,将使用被指定为操作数的访问键而不是 PSW 键。

[0106] 地址空间控制 (AS) :位 16 和 17,它们与程序状态字位 5 一起控制转换模式。

[0107] 条件码 (CC) :位 18 和 19 是两个条件码位。所述条件码被设为 0、1、2 或 3,具体取决于在执行特定指令时获取的结果。多数算术和逻辑运算以及某些其他运算都设置条件码。指令 BRANCH ON CONDITION 可以将条件码值中的任何选择的值指定为分支准则。

[0108] 指令地址 :程序状态字的位 64-127 为指令地址。该地址指定要执行的下一指令的最左侧字节的位置,除非 CPU 处于等待状态 (程序状态字的位 14 为 1)。

[0109] 地址类型和格式

[0110] 为了寻址主存储装置,将识别三种基本类型的地址 :绝对地址、实地址和虚拟地址。根据在存储访问期间施加于地址的变换来区分所述地址。地址转换将虚拟地址转换成实地址。通过添加前缀将实地址转换成绝对地址。除了三种基本地址类型以外,还定义了附加类型,这些附加类型被视为三种基本类型中的一个或另一个,具体取决于指令和当前模式。

[0111] 绝对地址

[0112] 绝对地址是分配给主存储位置的地址。绝对地址用于存储访问,不需要对其执行任何变换。配置中的通道子系统以及所有 CPU 通过使用相同的绝对地址来访问共享主存储位置。通常为可用的主存储装置分配从 0 开始的连续绝对地址,以及在整数界限上以完整的 4KB 数据块分配地址。当尝试在未被分配给物理位置的块中使用绝对地址时,会识别到异常。在某些模型上,可提供存储重新配置控制,它允许操作者更改绝对地址与物理位置之间的对应关系。但是,在任意时刻,一个物理位置都不与一个以上的绝对地址关联。包括根据字节位置的绝对地址排序的字节位置的存储装置被称为绝对存储装置。

[0113] 实地址

[0114] 实地址标识实存储装置中的位置。当使用实地址访问主存储装置时,借助添加前缀来转换实地址以形成绝对地址。对于配置中的每个 CPU,任何时刻都存在一个实地址到绝对地址映射。当 CPU 使用实地址访问主存储装置时,可通过添加前缀将此实地址转换成绝对地址。由 CPU 的前缀寄存器中的值定义特定的变换。包括根据字节位置的实地址排序的字节位置的存储装置被称为实存储装置。

[0115] 虚拟地址

[0116] 虚拟地址标识虚拟存储装置中的位置。当使用虚拟地址访问主存储装置时,会借助动态地址转换将虚拟地址转换成需要添加前缀以形成绝对地址的实地址,或直接转换为绝对地址。

[0117] 主虚拟地址

[0118] 主虚拟地址是要通过主地址空间控制元素 (PASCE) 转换的虚拟地址。在主空间模式下,逻辑地址被视为主虚拟地址。在主空间模式下、二级空间模式下或存取寄存器模式下,指令地址被视为主虚拟地址。MOVE TO PRIMARY 的第一操作数地址和 MOVE TO SCNDARY 的第二操作数地址被视为主虚拟地址。

[0119] 二级虚拟地址

[0120] 二级虚拟地址是要通过二级地址空间控制元素 (SASCE) 转换的虚拟地址。在二级空间模式下, 逻辑地址被视为二级虚拟地址。MOVE TO PRIMARY 的第二操作数地址和 MOVE TO SCNDARY 的第一操作数地址被视为二级虚拟地址。

[0121] AR 指定的虚拟地址

[0122] AR 指定的虚拟地址是要通过存取寄存器指定的地址空间控制元素转换的虚拟地址。在存取寄存器模式下, 逻辑地址被视为 AR 指定的地址。

[0123] 起始虚拟地址

[0124] 起始虚拟地址是要通过起始地址空间控制元素 (HASCE) 转换的虚拟地址。在起始空间模式下, 逻辑地址和指令地址被视为起始虚拟地址。

[0125] 指令地址

[0126] 用于从存储装置取回指令的地址被称为指令地址。指令地址在实模式下被视为实地址, 在主空间模式下、二级空间模式下或存取寄存器模式下被视为主虚拟地址, 以及在起始空间模式下被视为起始虚拟地址。当前程序状态字中的指令地址和 EXECUTE 的目标地址为指令地址。

[0127] 有效地址

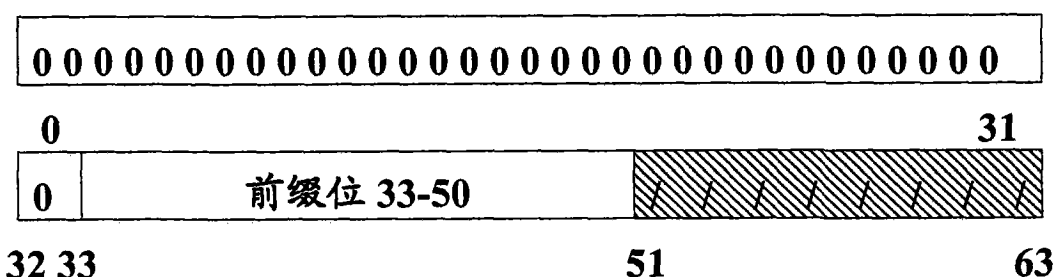
[0128] 在某些情况下, 使用术语“有效地址”非常方便。有效地址是通过动态地址转换或添加前缀执行任何变换之前存在的地址。有效地址可以在寄存器内直接指定, 也可以通过地址算术得出。地址算术是对基址和位移执行加法或对基址、索引和位移执行加法。

[0129] 添加前缀

[0130] 添加前缀提供了这样的能力: 对于每个 CPU, 将实地址范围 0-8191 分配给绝对存储装置中的不同块, 从而允许共享主存储装置的一个以上 CPU 在干扰最小, 尤其是在处理中断的情况下并行运行。添加前缀使范围 0-8191 内的实地址以一对一的方式对应于通过 CPU 的前缀寄存器的位位置 0-50 中的值所标识的 8KB 绝对地址块 (前缀区域), 并使通过所述前缀寄存器中的值标识的实地址块以一对一的方式对应于绝对地址 0-8191。其余的实地址与对应的绝对地址相同, 该变换允许每个 CPU 访问所有主存储装置, 包括第一 8KB 字节以及由其他 CPU 的前缀寄存器指定的位置。

[0131] 所述前缀是前缀寄存器的位位置 0-50 中包含的 51 位量。在一个实施例中, 前缀寄存器具有以下格式:

[0132]



[0133] 前缀寄存器的格式

[0134] 在应用前缀时, 取决于实地址的位 0-50, 通过使用以下规则之一将实地址变换成绝对地址:

[0135] 如果地址的位 0-50 全部为 0, 则地址的位 0-50 被前缀的位 0-50 所替换。

- [0136] 如果地址的位 0-50 等于前缀的位 0-50,则地址的位 0-50 被 0 所替换。
- [0137] 如果地址的位 0-50 不全部为 0 且不等于前缀的位 0-50,则地址的位 0-50 保持不变。
- [0138] 只有提供给存储装置的地址才通过添加前缀进行变换。地址的源的内容保持不变。
- [0139] 即使在前缀寄存器包含的位全部为 0(在此情况下实地址及其对应的绝对地址完全相同)时也将区分实地址和绝对地址。
- [0140] 图 13 示出了实地址和绝对地址之间的关系。
- [0141] 地址空间是连续的整数序列(虚拟地址);另外还包括特定的变换参数,所述参数允许将每个数字与存储装置中的字节位置关联。所述序列以 0 开始,从左到右进行排列。
- [0142] 当 CPU 使用虚拟地址访问主存储装置时,首先通过动态地址转换(DAT)将虚拟地址转换成实地址或绝对地址。可进一步向实地址添加前缀以形成绝对地址。DAT 可使用区第一表、区第二表、区第三表、段表以及页表作为变换参数。特定地址空间的最高级别表的指定(起始地址和长度)被称为地址空间控制元素(ASCE),需要找到该元素以供控制寄存器中的 DAT 使用或者由存取寄存器指定该元素。备选地,地址空间的 ASCE 可以是实空间指定,后者指示 DAT 不使用任何表并且仅通过将虚拟地址视为实地址来转换虚拟地址。
- [0143] DAT 在不同的时间使用不同的控制寄存器中的 ASCE 或使用由存取寄存器指定的 ASCE。此选择由当前程序状态字中指定的转换模式确定。可提供四种转换模式:主空间模式、二级空间模式、存取寄存器模式以及起始空间模式。可根据转换模式寻址不同的地址空间。
- [0144] 在 CPU 处于主空间模式或二级空间模式中的任何时刻,CPU 可以转换属于两个地址空间(即,主地址空间和二级地址空间)的虚拟地址。在 CPU 处于存取寄存器模式中的任何时刻,它可以转换最多 16 个地址空间(即,主地址空间和最多 15 个 AR 指定的地址空间)的虚拟地址。在 CPU 处于起始空间模式中的任何时刻,它可以转换起始地址空间的虚拟地址。
- [0145] 这样标识主地址空间是因为其包括主虚拟地址,通过主地址空间控制元素(PASCE)转换所述主虚拟地址。类似地,二级地址空间包括通过二级地址空间控制元素(SASCE)转换的二级虚拟地址。AR 指定的地址空间包括通过存取寄存器指定的地址空间控制元素(AR 指定的 ASCE)转换的 AR 指定的虚拟地址,以及起始地址空间包括通过起始地址空间控制元素(HASCE)转换的起始虚拟地址。主 ASCE 和二级 ASCE 分别位于控制寄存器 1 和 7 中。AR 指定的 ASCE 可在控制寄存器 1 和 7 中,或者在被称为 ASN 第二表表项的表表项中。HASCE 在控制寄存器 13 中。
- [0146] 动态地址转换
- [0147] 动态地址转换是将虚拟地址(例如,在存储引用期间)转换成对应的主存储器地址(所述实施例中的实地址或绝对地址)的过程。所述虚拟地址可以是主虚拟地址、二级虚拟地址、存取寄存器指定的虚拟地址或起始虚拟地址。这些地址分别通过 PASCE、SASCE、AR 指定的 ASCE 或 HASCE 进行转换。选择适当的 ASCE 之后,对于所有四种虚拟地址类型而言,转换过程都是相同的。
- [0148] 寻址转换模式

[0149] 有效地址是通过动态地址转换或添加前缀执行任意变换之前存在的地址（虚拟地址）。程序状态字中控制动态地址转换的三个位分别为位 5 (DAT 模式位)、位 16 和 17 (地址空间控制位)。当 DAT 模式位为 0 时，DAT 处于关闭状态，CPU 处于实模式下。当 DAT 模式位为 1 时，DAT 处于打开状态，CPU 处于由地址空间控制位指定的转换模式下：二进制 00 指定主空间模式、二进制 01 指定存取寄存器模式，二进制 10 指定二级空间模式，以及二进制 11 指定起始空间模式。下面将示出各种模式以及在每种模式下对地址的处理。

[0150]

PSW 位			DAT	模式	地址处理	
5	16	17			指令地址	逻辑地址
0	0	0	关闭	实模式	实	实
0	0	1	关闭	实模式	实	实
0	1	0	关闭	实模式	实	实
0	1	1	关闭	实模式	实	实
1	0	0	打开	主空间模式	主虚拟	主虚拟
1	0	1	打开	存取寄存器模式	主虚拟	AR 指定的虚拟
1	1	0	打开	二级空间模式	主虚拟	二级虚拟
1	1	1	打开	起始空间模式	起始虚拟	起始虚拟

[0151] 转换模式

[0152] 程序状态字是一个 128 位字，其一部分提供了 2 个指示寻址模式的位。在一个实施例中，位 31 是扩展寻址模式 (EA) 位并且位 32 是基本寻址模式 (BA) 位。这两个位指示地址的大小。这两个位中每个位的状态都以二进制 (1 或 0) 表示。如果 EA 位为 0 且 BA 位为 0，则指示 24 位寻址。如果指示 24 位寻址，则 64 位字 (64 位实体一般称为双字) 的位 40-63 是地址所在的位。当指令地址占据 128 位实体 (四倍长字) 的第二个 64 位时，程序状态字中的位位置如下所示。在 24 位模式下，指令地址位于程序状态字的位 104-127 中。在 31 位模式下，指令地址位于程序状态字的位 97-127 中。在 64 位模式下，指令地址位于程序状态字的位 64-127 中。如果 EA 位为 0 且 BA 位为 1，则指示 31 位寻址。相应的 64 位字包含位于位位置 33-63 中的 31 位地址。如果 EA 位为 1 且 BA 位为 1，则 64 位字的位 0-63 (全部 64 位) 包含地址。否则，将指示异常情况。一旦获取了寻址模式，就需要确定 ASCE。

[0153] 地址空间控制元素 (ASCE)

[0154] 现在参考图 3，其中示出了如何使用程序状态字针对虚拟地址的动态地址转换确定有效的地址空间控制元素 (ASCE) 的一个实施例。ASCE 可以例如指定 2GB ($G = 2^{30}$) 的地址空间。或者，可以例如指定 4TB ($T = 2^{40}$)、8PB ($P = 2^{50}$) 或 16EB ($E = 2^{60}$) 的地址空间。或者，其可以指定实空间指定。实空间指定使虚拟地址被视为存储装置中的实地址而不引用一个或多个地址转换表。

[0155] 程序状态字 300 包含转换 (T) 位 302 和地址空间 (AS) 位 304。在 306，如果转换

(T) 位为 0, 则地址为实地址 326。在 308, 如果地址空间 (AS) 等于 0 (二进制 00), 则该虚拟地址的有效 ASCE 为主地址空间控制元素 (PASCE) 310。在 312, 如果地址空间 (AS) 等于 1 (二进制 01), 则有效 ASCE 为存取寄存器指定的地址空间控制元素 314。在 316, 如果地址空间 (AS) 等于 2 (二进制 10), 则有效 ASCE 为二级地址空间控制元素 (SASCE) 318。否则, 地址空间 (AS) 等于 3 (二进制 11) 并且有效 ASCE 为起始地址空间控制元素 (HASCE) 322。

[0156] 选择有效 ASCE 之后, 动态地址转换过程优选地对于所有四种虚拟地址类型均相同。

[0157] 段表指定或区表指定将使转换通过操作系统在实或绝对存储装置中建立的表来执行。实空间指定将使虚拟地址简单地被视为实地址而不使用存储装置中的表。

[0158] 在使用段表指定或区表指定进行转换的过程中, 将识别三种类型的信息单位: 区、段和页。区是从 2GB 边界开始的跨 2GB 的顺序虚拟地址块。段是从 1MB 边界开始的跨 1MB 的顺序虚拟地址块。页是从 4KB 边界开始的跨 4KB 的顺序虚拟地址块。

[0159] 虚拟地址格式

[0160] 虚拟地址转换可涉及引用转换表层次结构的多个转换表以获取实地址或绝对地址。可以进一步对实地址执行添加前缀操作以形成绝对地址。虚拟地址包含对转换表层次结构中的转换表内的表项的索引。因此, 虚拟地址被分为四个主要字段。位 0-32 被称为区索引 (RX), 位 33-43 被称为段索引 (SX), 位 44-51 被称为页索引 (PX), 以及位 52-63 被称为字节索引 (BX)。在一个实施例中, 虚拟地址具有以下格式:

[0161]

RX	SX	PX	BX
0	33	44	52
			63

[0162] 虚拟地址的格式

[0163] 如其 ASCE 确定的, 虚拟地址空间可以是包括一个区的 2GB 空间, 也可以是包括最多 8GB 的区的最多 16EB 空间。应用到 2GB 地址空间的虚拟地址的 RX 部分必须全部为 0, ; 否则, 将识别到异常。虚拟地址的 RX 部分本身被分为三个字段。位 0-10 被称为区第一索引 (RFX), 位 11-21 被称为区第二索引 (RSX), 以及位 22-32 被称为区第三索引 (RTX)。在一个实施例中, 虚拟地址的位 0-32 具有以下格式:

[0164]

RFX	RSX	RTX
0	11	22
		33

[0165] 虚拟地址的 RX 的格式

[0166] 其中 RTX 为最左侧有效部分 (42 位地址) 的虚拟地址能够寻址 4TB (2048 个区), 其中 RTX 为最左侧有效部分 (53 位地址) 的虚拟地址能够寻址 8PB (4, 193, 044 个区), 并且其中 RTX 为最左侧有效部分 (64 位地址) 的虚拟地址能够寻址 16EB (8, 589, 934, 592 个区)。

[0167] 其中 RX 为 0 的虚拟地址可以通过以下两种转换表转换为实地址: 段表和页表。启用 EDAT 工具之后, 可以仅使用段表完成所述转换。RFX 可以为非零, 在这种情况下, 需要区

第一表、区第二表和区第三表。如果 RFX 为 0,但是 RSX 为非零,则需要区第二表和区第三表。如果 RFX 和 RSX 为 0,但是 RTX 为非零,则需要区第三表。

[0168] 如果地址空间的 ASCE 未指定转换对地址空间的引用时所需的最高级别的表(从区第一表开始,然后继续向下到段表),将识别到异常。

[0169] 虚拟地址的动态转换

[0170] 现在参考图 4,示出了其中使用在图 3 中确定的有效 ASCE 来确定在虚拟地址转换中使用的转换表层次结构中的第一转换表的一个实施例。

[0171] 在一个实施例中,控制寄存器 1(CR1) 包含 PASCE。控制寄存器 7(CR7) 包含 SASCE。控制寄存器 13(CR13) 包含 HASCE,以及由存取寄存器转换(ART)过程得到的地址空间第二表表项(ASTE)包含存取寄存器指定的地址空间控制元素。将从这些位置之一选择有效 ASCE400。

[0172] 有效 ASCE 400 的第一部分包含表起始地址 402,表起始地址 402 包含指定区第一表、区第二表、区第三表或段表的起始地址。向表起始地址(位 0...51)附加 12 个二进制 0 以形成在虚拟地址转换中使用的转换表层次结构中的最高转换表的 64 位起始地址。有效 ASCE 400 还包含实空间控制(R)位 404 和 DT 位 406。如果实空间控制(R)位为 0,则 DT 位由选择器 408 解码以确定哪个特定的起始地址是表起始地址 402。如果 DT 位等于 3(二进制 11),则表起始地址 402 指定区第一表 410。如果 DT 位等于 2(二进制 10),则表起始地址 402 指定区第二表 412。如果 DT 位等于 1(二进制 01),则表起始地址 402 指定区第三表 414。否则,如果 DT 位等于 0(二进制 00),则表起始地址 402 指定段表 416。

[0173] 区第一表、区第二表或区第三表有时简称为区表。类似地,区第一表指定、区第二表指定或区第三表指定有时被称为区表指定。区表、段表和页表反映实存储装置的当前指定。页是用于分配虚拟存储装置的术语。实存储装置以固定块的形式进行分配。即使被分配给一组顺序虚拟地址,页在实存储装置中也无需相邻。

[0174] 当转换中使用的 ASCE 是区第一表指定时,转换过程包含多级查找,所述查找例如使用区第一表、区第二表、区第三表、段表以及可选地使用页表。这些表驻留在实存储装置或绝对存储装置中。当 ASCE 为区第二表指定、区第三表指定或段表指定时,将忽略级别在指定级别以上的表中的查找,并且忽略更高级表本身。

[0175] 现在参考图 5A,示出了使用转换表层次结构执行虚拟地址的动态地址转换的一个实施例。

[0176] 图 4 的有效 ASCE 400 包含指定类型(DT)位 406。如果 ASCE 的实空间控制(R)404 位为 0,则 DT 位由选择器 408 解码以确定表起始地址 402 指定哪个起始地址。如果实空间控制(R)位为 1,则动态地址转换如图 5B 中的节点 D 564 处所示那样发生。

[0177] 如果选择器 408 中的 DT 位等于 3(二进制 11),则转换表层次结构中指定的第一表为区第一表。在 502,将表起始地址 402 与虚拟地址的区第一索引(RFX)508 部分进行算术相加以引用区第一表中的区第一表表项 506。表起始地址(在右侧附加 12 个 0,或乘以 4096)与索引乘以 8 的乘积(或右侧附加三个 0 的索引)进行相加。区第一表表项包含转换中使用的转换表层次结构中的下一低级表的区第二表起始地址 504。区第一表的下一低级表为区第二表。如果区第一表表项的无效(I)位等于 1,则区第一表表项无效,并且不能在转换中使用。将指示异常状况。

[0178] 如果选择器 408 中的 DT 位等于 2(二进制 10),则转换表层次结构中指定的第一表为区第二表。在 510,将表起始地址 402 与虚拟地址的区第二索引 (RSX) 516 部分进行算术相加以引用区第二表中的区第二表表项 514。表起始地址(在右侧附加 12 个 0,或乘以 4096)与索引乘以 8 的乘积(或右侧附加三个 0 的索引)进行相加。区第二表表项包含转换中使用的转换表层次结构中的下一低级表的区第三表起始地址 512。区第二表的下一低级表为区第三表。如果区第二表表项的无效(I)位等于 1,则区第二表表项无效,并且将指示异常状况。

[0179] 如果选择器 408 中的 DT 位等于 1(二进制 01),则转换表层次结构中指定的第一表为区第三表。在 518,将表起始地址 402 与虚拟地址的区第三索引 (RTX) 524 部分进行算术相加以引用区第三表中的区第三表表项 522。表起始地址(在右侧附加 12 个 0,或乘以 4096)与索引乘以 8 的乘积(或右侧附加三个 0 的索引)进行相加。区第三表表项包含转换中使用的转换表层次结构中的下一低级表的段表起始地址 520。区第三表的下一低级表为段表。如果区第三表表项的无效(I)位等于 1,则区第三表表项无效,并且将指示异常状况。

[0180] 如果选择器 408 中的 DT 位等于 0(二进制 00),则转换表层次结构中指定的第一表为段表。在 526,将表起始地址 402 与虚拟地址的段索引 (SX) 532 部分进行算术相加以引用段表中的段表表项 530。表起始地址(在右侧附加 12 个 0,或乘以 4096)与索引乘以 8 的乘积(或右侧附加三个 0 的索引)进行相加。段表表项包含在 528 处示出的页表的起始地址或段帧绝对地址 (SFAA)。如果段表表项的无效(I)位等于 1,则段表表项无效,并且将指示异常状况。

[0181] 在 538,检查段表的 STE 格式控制 (FC) 位。如果 STE 格式控制为 1,则段表表项 530 包含段帧绝对地址 (SFAA) 552 并且参考图 5C 中的节点 562 继续动态地址转换。否则,从段表获取的段表表项包含页表起始地址并且参考图 5B 中的节点 560 继续动态地址转换。

[0182] 现在参考图 5B。如果段表表项中的 STE 格式控制为 0,则从段表获取的段表表项包含转换表层次结构中的下一低级表的起始地址。段表的下一低级表为页表。在 538,将从图 5A 的段表表项 530 获取的页表起始地址 528 与虚拟地址的页索引 (PX) 534 部分进行算术相加以引用页表中的页表表项 542。页表表项包含页帧实地址 (PFRA) 546。当在 548 处将页帧实地址的最左侧位与虚拟地址的字节索引 (BX) 536 部分连接时,将获得 64 位实地址 550。可以进一步对实 64 位地址执行添加前缀操作以形成绝对地址。转换后的虚拟地址引用主存储装置或存储器中所需的 4KB(4096 字节)数据块。

[0183] 优选地,在将虚拟地址动态转换为存储器地址时使用的信息随同与虚拟地址关联的存储块地址一起存储在转换后备缓冲器表项标签中。后续存储访问可通过将 ASCE 信息和虚拟地址信息与转换后备缓冲器标签进行比较来迅速转换虚拟地址。如果发现所述标签为虚拟地址的标签,则可使用存储块的转换后备缓冲器地址,而不是执行对所涉及的每个转换表的缓慢顺序访问。在一个实施例中,页帧实地址 (PFRA) 与包含例如 ASCE 以及虚拟地址的 RX、SX 和 PX 部分的标签一起存储在转换后备缓冲器 544 的表项中。此后,从转换后备缓冲器中存储的信息得到对该虚拟地址的后续转换。

[0184] 现在参考图 5C。如果段表表项 530 中的 STE 格式控制为 1,则段表表项包含段帧绝对地址 (SFAA) 552。当在 554,将段帧绝对地址的最左侧位与虚拟地址的页索引 534 部分

和字节索引 536 部分连接时,将获得 64 位绝对地址 556。转换后的虚拟地址引用主存储装置或存储器中所需的大数据块。大数据块的大小至少为 1MB(1,048,576 字节)。

[0185] 在一个实施例中,段帧绝对地址(SFAA)与虚拟地址的RX和SX部分一起存储在转换后备缓冲器 544 中。此后,从转换后备缓冲器中存储的信息得到对该虚拟地址的后续转换。

[0186] 转换表表项格式

[0187] 下面将示出在转换中使用的转换表层次结构中的各种转换表表项的实施例。

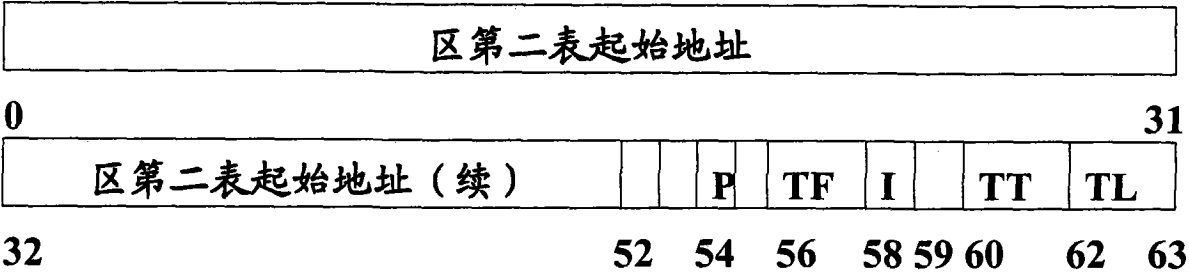
[0188] 区表表项

[0189] 术语“区表表项”表示区第一表表项、区第二表表项或区第三表表项。从区第一表、区第二表和区第三表取回的表项具有下面的格式。包含表项的表的级别(第一、第二或第三)通过表项中的表类型(TT)位来标识。

[0190] 在一个实施例中,区第一表表项、区第二表表项和区第三表表项的格式如下:

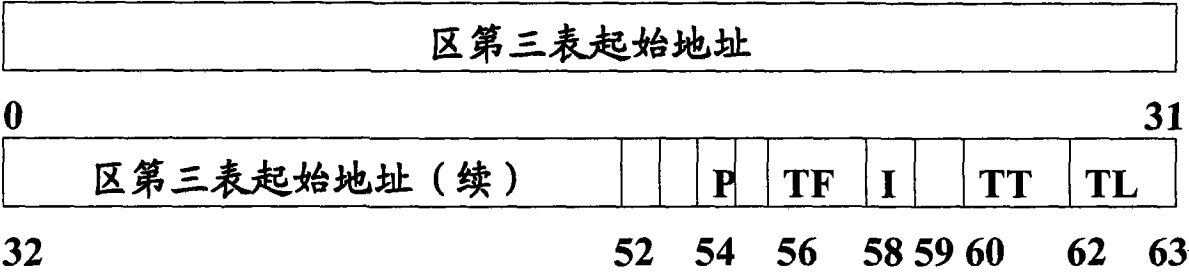
[0191] 区第一表表项(TT = 11)

[0192]



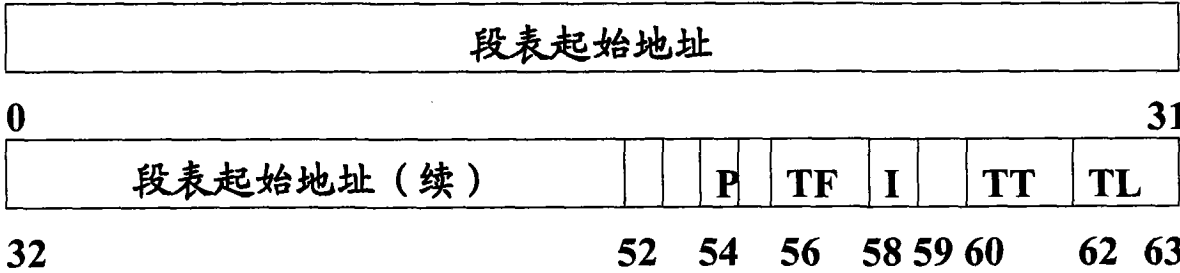
[0193] 区第二表表项(TT = 10)

[0194]



[0195] 区第三表表项(TT = 01)

[0196]



[0197] 区第二表起始地址、区第三表起始地址和段表起始地址:区第一表表项包含区第二表起始地址。区第二表表项包含区第三表起始地址。区第三表表项包含段表起始地址。以下描述适用于这三个表起始地址中的每一个。表项的位 0-51 以及在右侧附加的 12 个 0

形成指定下一低级表的开头的 64 位地址。

[0198] DAT 保护位 (P) :当应用增强的 DAT 时,位 54 被视为与在转换中使用的每个后续区表表项、段表表项以及页表表项 (当适用时) 中的 DAT 保护位进行“或”操作。因此,当此位为 1 时,DAT 保护应用到由区表表项指定的整个区或多个区。当未安装增强的 DAT 工具,或者已安装此工具但增强的 DAT 启用控制为 0 时,将忽略区表表项的位 54。

[0199] 区第二表偏移、区第三表偏移和段表偏移 (TF) :区第一表表项包含区第二表偏移。区第二表表项包含区第三表偏移。区第三表表项包含段表偏移。以下描述适用于三个表偏移中的每一个。表项的位 56 和 57 指定下一低级表的在该表开头处缺少的部分的长度,即,所述位指定实际存在于下一低级表中的第一表项的位置。所述位以 4,096 字节为单位指定缺少部分的长度,从而使缺少部分的长度可以以 512 个表项的倍数变化。以 4,096 字节为单位表示的缺少部分的长度等于 TF 值。使用偏移字段以及长度字段 (位 62 和 63) 的内容确定要通过下一低级表转换的虚拟地址部分 (RSX、RTX 或 SX) 是否指定实际存在于该表中的表项。

[0200] 区无效位 (I) :区第一表表项或区第二表表项中的位 58 控制与该表项关联的一组区是否可用。区第三表表项中的位 58 控制与该表项关联的单个区是否可用。当位 58 为 0 时,将使用区表表项进行地址转换。当此位为 1 时,表项无法用于转换。

[0201] 表类型位 (TT) :区第一表表项、区第二表表项和区第三表表项的位 60 和 61 按如下方式标识包含表项的表的级别:位 60 和 61 必须标识正确的表级别,其中要考虑作为在转换中使用的 ASCE 的表指定类型以及迄今为止使用的表级别的数量;否则,将识别到转换规范异常。下表示出了表类型位:

[0202] 位 60

[0203] 和 61 区表级别

[0204] 11 第一

[0205] 10 第二

[0206] 01 第三

[0207] 区表表项的表类型位

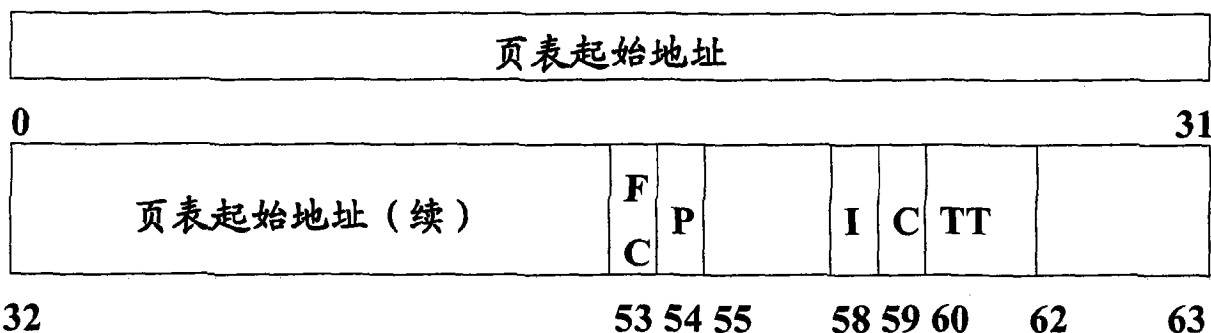
[0208] 区第二表长度、区第三表长度和段表长度 (TL) :区第一表表项包含区第二表长度。区第二表表项包含区第三表长度。区第三表表项包含段表长度。以下描述适用于三个表长度中的每一个。表项的位 62 和 63 指定以 4,096 字节为单位表示的下一低级表的长度,从而使表长度以 512 个表项的倍数变化。以 4,096 字节为单位表示的下一低级表的长度大于 TL 值。使用长度字段以及偏移字段 (位 56 和 57) 的内容确定要通过下一低级表转换的虚拟地址部分 (RSX、RTX 或 SX) 是否指定实际存在于表中的表项。区表表项的所有其他位位置均被保留以供将来可能扩展并且应包含 0;否则,将来程序可能无法兼容运行。当应用增强的 DAT 时,即使表项无效,区表表项的保留位位置也应包含 0。

[0209] 段表表项

[0210] 当不应用增强的 DAT,或者应用增强的 DAT 但段表表项的 STE 格式控制 (位 53) 为 0 时,在一个实施例中,从段表取回的表项具有以下格式:

[0211] 段 - 表表项 (T = 00, FC = 0)

[0212]

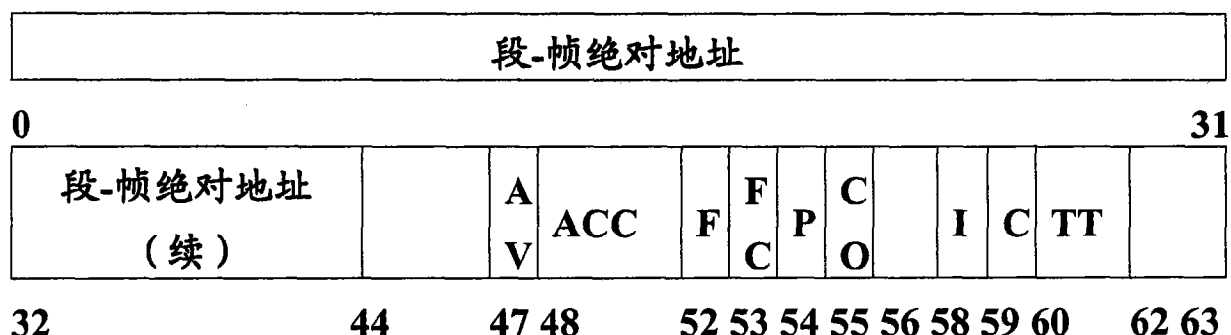


[0213] 段表表项的格式 I

[0214] 当应用增强的 DAT 并且 STE 格式控制为 1 时, 在一个实施例中, 从段表取回的表项具有以下格式:

[0215] 段 - 表表项 (T = 00, FC = 1)

[0216]



[0217] 段表表项的格式 II

[0218] 段表表项中的选定字段按如下方式进行分配:

[0219] 页表起始地址: 当不应用增强的 DAT, 或者应用增强的 DAT 但段表表项的 STE 格式控制 (位 53) 为 0 时, 位 0-52 以及在右侧附加的 11 个 0 形成指定页表的开头的 64 位地址。无法预测此地址是实地址还是绝对地址。

[0220] 段帧绝对地址 (SFAA): 当应用增强的 DAT 并且 STE 格式控制为 1 时, 表项的位 0-43 以及在右侧附加的 20 个 0 形成段的 64 位绝对地址。

[0221] ACCF 有效性控制 (AV): 当应用增强的 DAT 并且 STE 格式控制为 1 时, 位 47 是访问控制位和取回保护位 (ACCF) 有效性控制。当 AV 控制为 0 时, 将忽略段表表项的位 48-52。当 AV 控制为 1 时, 将按下文所述使用位 48-52。

[0222] 访问控制位 (ACC): 当应用增强的 DAT, STE 格式控制为 1 并且 AV 控制为 1 时, 段表表项的位 48-51 包含访问控制位, 这些控制位可用于应用到地址的任何键控访问检查。

[0223] 取回保护位 (F): 当应用增强的 DAT, STE 格式控制为 1 并且 AV 控制为 1 时, 段表表项的位 52 包含取回保护位, 此保护位可用于应用到地址的任何键控访问检查。

[0224] STE 格式控制 (FC): 当应用增强的 DAT 时, 位 53 是段表表项的格式控制, 如下所示:

[0225] 当 FC 位为 0 时, 表项的位 0-52 形成页表起始地址, 而位 55 被保留。

[0226] 当 FC 位为 1 时, 表项的位 0-43 形成段帧绝对地址, 位 47 是 ACCF 有效性控制, 位 48-51 是访问控制位, 位 52 是取回保护位, 而位 55 是更改记录忽略。当不应用增强的 DAT

时,将忽略位 53。

[0227] DAT 保护位 (P) :位 54,当为 1 时,指示 DAT 保护应用到整个段。

[0228] 当不应用增强的 DAT 时,位 54 被视为与在转换中使用的页表表项中的 DAT 保护位进行“或”操作。

[0229] 当应用增强的 DAT 时,在转换中使用的任何和全部区表表项中的 DAT 保护位被视为与段表表项中的 DAT 保护位进行“或”操作;当 STE 格式控制为 0 时,STE 中的 DAT 保护位还被视为与页表表项中的 DAT 保护位进行“或”操作。

[0230] 更改记录忽略 (CO) :当应用增强的 DAT 并且 STE 格式控制为 1 时,段表表项的位 55 是段的更改记录忽略。当不应用增强的 DAT,或者应用增强的 DAT 但 STE 格式控制为 0 时,将忽略段表表项的位 55。

[0231] 段无效位 (I) :位 58 控制与段表表项关联的段是否可用。

[0232] 当此位为 0 时,将使用段表表项进行地址转换。

[0233] 当此位为 1 时,段表表项无法用于转换。

[0234] 公共段位 (C) :位 59 控制段表表项的转换后备缓冲器副本的使用。当不应用增强的 DAT,或者当应用增强的 DAT 但格式控制为 0 时,位 59 还控制由段表表项指定的页表的转换后备缓冲器副本的使用。

[0235] 0 标识专用段;在此情况下,段表表项及其指定的任何页表仅可与指定段表表项所在的段表的段表起始地址结合使用。

[0236] 1 标识公共段;在此情况下,段表表项及其指定的任何页表可以继续用于转换与段索引对应的地址,即使指定了其他段表也是如此。

[0237] 但是,如果在转换中使用的 ASCE 内的专用空间控制 (位 55) 为 1,或者如果此 ASCE 是实空间指定,则公共段的段表表项和任何页表的转换后备缓冲器副本不可用。当所使用的 ASCE 中的专用空间控制为 1 时,如果在转换过程中从存储装置取回段表表项,则公共段位必须为 0。否则,将识别到转换规范异常。

[0238] 表类型位 (TT) :段表表项的位 60 和 61 是用于标识包含该表项的表的级别的二进制 00。区表表项或段表表项中的位 60 和 61 的所有可能值的含义如下所示:

[0239] 位 60

[0240] 和 61 表级别

[0241] 11 区第一

[0242] 10 区第二

[0243] 01 区第三

[0244] 00 段

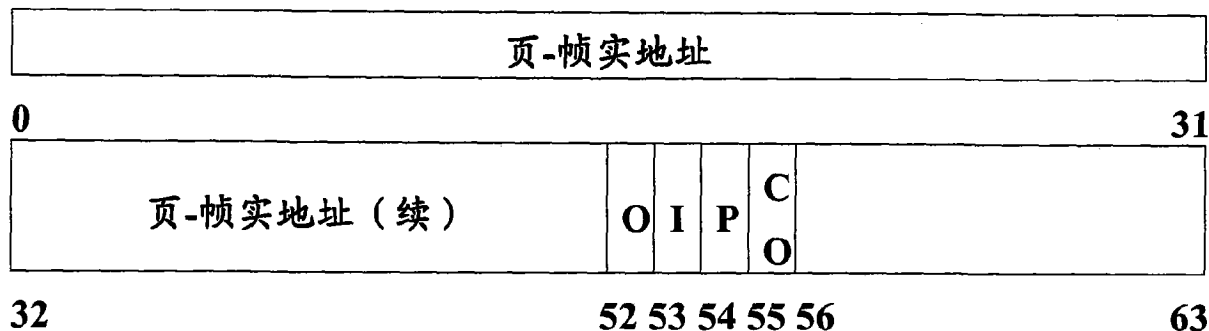
[0245] 表类型位 60、61

[0246] 位 60 和 61 必须标识正确的表级别,其中要考虑作为在转换中使用的 ASCE 的表指定类型以及迄今为止使用的表级别的数量;否则,将识别到转换规范异常。段表表项的所有其他位位置均被保留以供将来可能扩展,并且应包含 0;否则,将来程序可能无法兼容运行。当应用增强的 DAT 时,段表表项的保留位位置应包含 0,即使表项无效也是如此。

[0247] 页表表项

[0248] 在一个实施例中,从页表取回的表项具有以下格式:

[0249]



[0250] 页表表项的格式

[0251] 页表表项中的选定字段按如下方式进行分配：

[0252] 页帧实地址 (PFRA) :位 0-51 提供实存储地址的最左侧位。将这些位与右侧的虚拟地址的 12 位字节索引字段连接时,将获得 64 位实地址。

[0253] 页无效位 (I) :位 53 控制与页表表项关联的页是否可用。当此位为 0 时,将使用页表表项进行地址转换。当此位为 1 时,页表表项无法用于转换。

[0254] DAT 保护位 (P) :位 54 控制是否可以在页中进行存储访问。此保护机制是除了键控保护和低地址保护机制以外的机制。此位对取回访问没有影响。如果此位为 0,则允许存储到页,但受以下附加约束限制：

[0255] DAT 保护位在转换中使用的段表表项中为 0。

[0256] 当应用增强的 DAT 时,DAT 保护位在转换中使用的所有区表表项中为 0。

[0257] 其他保护机制

[0258] 如果此位为 1,则不允许存储。如果不存在较高优先级异常状况,则当 DAT 保护位为 1 时,尝试存储将导致识别到保护异常。当确定 DAT 保护是否应用到页时,段表表项中的 DAT 保护位被视为与位 54 进行“或”操作。如果应用增强的 DAT,则当确定是否应用 DAT 保护时,在转换中使用的任何区表表项中的 DAT 保护位也被视为与位 54 进行“或”操作。

[0259] 更改记录忽略 (CO) :当不应用增强的 DAT 时,页表表项的位 55 必须包含 0 ;否则,将作为执行使用此表项进行地址转换的指令的一部分而识别到转换规范异常。当应用增强的 DAT 并且 STE 格式控制为 0 时,页表表项的位 55 是页的更改记录忽略。

[0260] 表项的位位置 52 必须包含 0 ;否则,将作为执行使用此表项进行地址转换的指令的一部分而识别到转换规范异常。位位置 56-63 未被分配并被忽略。

[0261] 动态转换的另一实施例

[0262] 此部分描述在使用虚拟地址访问主存储装置之前隐式执行的转换过程。

[0263] 虚拟地址的转换由程序状态字中的 DAT 模式位和地址空间控制位控制,以及由控制寄存器 1、7 和 13 中的 ASCE 及存取寄存器所指定的 ASCE 控制。当在转换中使用的 ASCE 是区第一表指定时,通过区第一表、区第二表、区第三表、段表以及页表执行转换,所有这些表都位于实存储装置或绝对存储装置中。当 ASCE 是较低级类型的表指定 (区第二表指定、区第三表指定或段表指定) 时,仅通过以指定级别开始的表级别执行转换,并且如果为非零,则需要使用较高级表的虚拟地址位必须全部为 0 ;否则,将识别到 ASCE 型异常。当 ASCE 是实空间指定时,虚拟地址被视为实地址,并且不使用实存储装置或绝对存储装置中的表项。

[0264] 用于特定地址转换的 ASCE 被称为有效 ASCE。相应地,当转换主虚拟地址时,使用控制寄存器 1 的内容作为有效 ASCE。类似地,对于二级虚拟地址,使用控制寄存器 7 的内容;对于 AR 指定的虚拟地址,使用由存取寄存器指定的 ASCE;以及对于起始虚拟地址,使用控制寄存器 13 的内容。

[0265] 当有效 ASCE 中的实空间控制为 0 时,ASCE 中的指定类型指定表指定类型:区第一表、区第二表、区第三表或段表。根据指定中的表长度字段检查虚拟地址的对应部分(区第一索引、区第二索引、区第三索引或段索引),并将其添加到指定中的起始地址以便在已指定表中选择表项。如果指定中的表长度字段确定所选表项在其表之外,或者如果所选表项中的 I 位为 1,则取决于由指定所指出的表级别,将识别到区第一转换、区第二转换、区第三转换或段转换异常。如果所选表项中的表类型位未指示预期的表级别,将识别到转换规范异常。

[0266] 通过有效 ASCE 选择的表项指定要使用的下一低级表。如果当前表是区第一表、区第二表或区第三表,则根据当前表项中的表偏移字段和表长度字段检查虚拟地址的下一部分(分别为区第二索引、区第三索引或段索引),并将其添加到表项中的起始地址以选择下一低级表中的表项。如果当前表项中的表偏移字段和表长度字段确定下一表中的所选表项在其表之外,或者如果所选表项中的 I 位为 1,则取决于下一表的级别,将识别到区第二转换、区第三转换或段转换异常。如果所选表项中的表类型位未指示预期的表级别,将识别到转换规范异常。

[0267] 继续借助连续表级别处理虚拟地址的各部分,直到已选择段表表项。段表表项包含应用到指定段中所有页的页保护位。

[0268] 将虚拟地址的页索引部分添加到段表表项中的页表起始地址,以便选择页表中的表项。如果页表表项中的 I 位为 1,将识别到页转换异常。页表表项包含表示虚拟地址转换的实地址的最左侧位,并且包含仅应用到由页表表项指定的页的页保护位。

[0269] 虚拟地址的字节索引字段不作改变地用作实地址的最右侧位位置。

[0270] 为了消除与引用实存储装置或绝对存储装置中的转换表关联的延迟,从表取回的信息通常还放置在特殊缓冲区(即转换后备缓冲器)中,并且可以使用转换后备缓冲器中记录的信息执行涉及相同表项的后续转换。转换后备缓冲器还可以记录与实空间指定相关的虚等于实转换。

[0271] 在地址转换过程中,每当为了从区表、段表或页表取回表项而对实存储装置或绝对存储装置进行访问时,将不应用键控保护。

[0272] 在 ASCE 指定的表中查找

[0273] 有效 ASCE 的 DT 控制(位 60-61)指定 ASCE 的表指定类型以及要借助指定的表转换的虚拟地址部分两者,如下所示:

[0274]	位 60		通过表转换的虚
[0275]	和 61	指定类型	拟地址部分
[0276]	11	区第一表	区第一索引
[0277]			(位 0-10)
[0278]	10	区第二表	区第二索引
[0279]			(位 11-21)

[0280]	01	区第三表	区第三索引
[0281]			(位 22-32)
[0282]	00	段表	段索引
[0283]			(位 33-43)

[0284] 借助指定表的转换

[0285] 当位 60 和 61 具有二进制值 11 时,使用虚拟地址的区第一索引部分以及 ASCE 中包含的区第一表起始地址从区第一表中选择表项。通过向区第一表指定的位 0-51 的右侧附加 12 个 0,并添加最右侧附加 3 个 0 而最左侧附加 50 个 0 的区第一索引,获得实存储装置或绝对存储装置中的区第一表表项的 64 位地址。作为区第一表查找过程的一部分,将虚拟地址的位 0 和 1(它们是区第一索引的位 0 和 1)与区第一表指定的表长度(位 62 和 63)进行比较,以确定寻址的表项是否在区第一表内。如果表长度字段中的值小于虚拟地址的对应位位置中的值,将识别到区第一转换异常。如果在转换中使用转换后备缓冲器中的区第一表表项的等价表项,则可以省略与表长度进行的比较。从区第一表取回的表项指定对应区第二表的开头以及指定其偏移和长度。

[0286] 当 ASCE 的位 60 和 61 具有二进制值 10 时,使用虚拟地址的区第二索引部分以及 ASCE 中包含的区第二表起始地址从区第二表中选择表项。将虚拟地址的位 11 和 12(它们是区第二索引的位 0 和 1)与 ASCE 中的表长度进行比较。如果表长度字段中的值小于虚拟地址的对应位位置中的值,将识别到区第二转换异常。如果在转换中使用转换后备缓冲器中的区第二表表项的等价表项,则可以省略与表长度进行的比较。否则,区第二表查找过程与区第一表查找过程相同;从区第二表取回的表项指定对应的区第三表的开头并指定其偏移和长度。

[0287] 当 ASCE 的位 60 和 61 具有二进制值 01 时,使用虚拟地址的区第三索引部分以及 ASCE 中包含的区第三表起始地址从区第三表中选择表项。将虚拟地址的位 22 和 23(它们是区第三索引的位 0 和 1)与 ASCE 中的表长度进行比较。如果表长度字段中的值小于虚拟地址的对应位位置中的值,将识别到区第三转换异常。否则,区第三表查找过程与区第一表查找过程相同,包括检查区第三表表项中的表类型位。从区第三表取回的表项指定对应的段表的开头并指定其偏移和长度。

[0288] 当 ASCE 的位 60 和 61 具有二进制值 00 时,使用虚拟地址的段索引部分以及 ASCE 中包含的段表起始地址从段表中选择表项。将虚拟地址的位 33 和 34(它们是段索引的位 0 和 1)与 ASCE 中的表长度进行比较。如果表长度字段中的值小于虚拟地址的对应位位置中的值,将识别到段转换异常。如果在转换中使用转换后备缓冲器中的段表表项的等价表项,则可以省略与表长度进行的比较。否则,段表查找过程与区第一表查找过程相同,包括检查段表表项中的表类型位。处理如下所示:

[0289] 当不应用增强的 DAT,或者应用增强的 DAT 但 STE 格式控制为 0 时,从段表取回的表项指定对应页表的开头,并且按照下面“页表查找”中所述继续处理。

[0290] 当应用增强的 DAT 并且 STE 格式控制为 1 时,从段表取回的表项包含段帧绝对地址的最左侧位。如果在转换中使用的任何区表表项中或在段表表项中 DAT 保护位均为 1,并且针对其执行转换的存储引用为存储,将识别到保护异常。

[0291] 在由区表表项指定的表中查找

[0292] 当有效 ASCE 是区表指定时,按照上一部分中所述选择区表表项。然后,使用所选表项的内容以及虚拟地址的下一索引部分在下一低级表中选择表项,此下一低级表可以是另一区表或段表。当借助 ASCE 选择的表表项是区第一表表项时,使用虚拟地址的区第二索引部分以及区第一表表项中包含的区第二表起始地址从区第二表中选择表项。通过向区第一表表项的位 0-51 的右侧附加 12 个 0,并添加最右侧附加 3 个 0 而最左侧附加 50 个 0 的区第二索引,获得实存储装置或绝对存储装置中的区第二表表项的 64 位地址。

[0293] 当形成区第二表表项、区第三表表项或段表表项的地址时,无法预测是在添加表索引值之前向较高级别表表项中包含的相应表起始地址添加前缀(如果有),还是向通过添加表起始地址和表索引值形成的表表项地址添加前缀。

[0294] 作为区第二表查找过程的一部分,将虚拟地址的位 11 和 12(它们是区第二索引的位 0 和 1)与区第一表表项的表偏移(位 56 和 57)进行比较,并与区第一表表项的表长度(位 62 和 63)进行比较,以确定寻址的表项是否在区第二表内。如果表偏移字段中的值大于虚拟地址的对应位位置中的值,或者如果表长度字段中的值小于虚拟地址的对应位位置中的值,将识别到区第二转换异常。

[0295] 区第二表指定对应的区第三表的开头并指定其偏移和长度。

[0296] 当借助 ASCE 选择的表表项是区第二表表项时,或者如果已借助区第一表表项的内容选择区第二表表项,则使用虚拟地址的区第三索引部分以及区第二表表项中包含的区第三表起始地址从区第三表中选择表项。将虚拟地址的位 22 和 23(它们是区第三索引的位 0 和 1)与区第二表表项中的表偏移和表长度进行比较。如果表偏移大于位 22 和 23,或者如果表长度小于位 22 和 23,将识别到区第三转换异常。否则,区第三表查找过程与区第二表查找过程相同。从区第三表取回的表项指定对应的段表的开头并指定其偏移和长度。

[0297] 当借助 ASCE 选择的表表项是区第三表表项时,或者如果已借助区第二表表项的内容选择区第三表表项,则使用虚拟地址的段索引部分以及区第三表表项中包含的段表起始地址从段表中选择表项。将虚拟地址的位 33 和 34(它们是段索引的位 0 和 1)与区第三表表项中的表偏移和表长度进行比较。如果表偏移大于位 33 和 34,或者如果表长度小于位 33 和 34,将识别到段转换异常。如果(1)ASCE 中的专用空间控制(位 55)为 1 以及(2)从段表取回的表项中的公共段位(位 59)为 1,将识别到转换规范异常。否则,段表查找过程与区第二表查找过程相同。处理如下所示:

[0298] 当不应用增强的 DAT,或者应用增强的 DAT 但 STE 格式控制为 0 时,从段表取回的表项指定对应页表的开头,并且按照下面“页表查找”中所述继续处理。

[0299] 当应用增强的 DAT 并且 STE 格式控制为 1 时,从段表取回的表项包含段帧绝对地址的最左侧位。如果在转换中使用的任何区表表项中或在段表表项中 DAT 保护位均为 1,并且针对其执行转换的存储引用为存储,将识别到保护异常。

[0300] 页表查找

[0301] 当不应用增强的 DAT,或者应用增强的 DAT 但 STE 格式控制为 0 时,使用虚拟地址的页索引部分以及段表表项中包含的页表起始地址从页表中选择表项。

[0302] 通过向页表起始地址的右侧附加 11 个 0,并添加最右侧附加 3 个 0 而最左侧附加 53 个 0 的页索引,获得实存储装置或绝对存储装置中的页表表项的 64 位地址。不会发生位位置 0 的进位输出。

[0303] 从页表取回的表项指示页的可用性,并且包含页帧实地址的最左侧位。将检查页无效位(位 53)以确定对应页是否可用。如果此位为 1,将识别到页转换异常。如果位位置 52 包含 1,将识别到转换规范异常。当不应用增强的 DAT,或者应用增强的 DAT 并且 STE 格式控制为 0 时,如果位位置 55 包含 1,也将识别到转换规范异常。如果在转换中使用的段表表项中、在页表表项中 DAT 保护位为 1,或者当在转换过程中使用的任何区表表项中应用增强的 DAT,并且针对其执行转换的存储引用为存储时,将识别到保护异常。

[0304] 实地址和绝对地址的形成

[0305] 当有效 ASCE 是实空间指定时,直接将虚拟地址的位 0-63 用作实存储装置地址。可以进一步向实地址添加前缀以形成绝对地址。当有效 ASCE 不是实空间指定并且在转换过程中未遇到异常时,适用以下情况:

[0306] 当不应用增强的 DAT,或者应用增强的 DAT 但 STE 格式控制为 0 时,从页表表项获取页帧实地址。将页帧实地址和虚拟地址的字节索引部分连接,并且页帧实地址形成最左侧部分。结果是与虚拟地址对应的实存储装置地址。可以进一步向实地址添加前缀以形成绝对地址。

[0307] 当应用增强的 DAT 并且 STE 格式控制为 1 时,从左到右分别将段帧绝对地址以及虚拟地址的页索引和字节索引部分连接,以形成与虚拟地址对应的绝对地址。

[0308] 转换过程中的异常识别

[0309] 无效地址和无效格式可导致在转换过程中识别到异常。当表表项中包含的信息用于转换并且被发现不正确时,将识别到异常。

[0310] 现在参考图 6,其示出了到从段表表项获取格式控制字段时的动态地址转换的一个实施例的流程图。

[0311] 在 602,获取要转换的虚拟地址。在 604,获取在虚拟地址转换中使用的最高转换表的起始地址。在转换中使用的第一转换表的起始地址取决于 ASCE 和 DT 位。在 606,使用虚拟地址的一部分引用转换表中的适当表表项。在 608,如果从转换表取回的表项不是段表表项,则尚未引用转换表层次结构中的段表。在这种情况下,在 610,从所述表表项获取转换表层次结构中的下一低级表的起始地址。使用虚拟地址的适当部分引用在转换中使用的下一低级表中的对应表表项。

[0312] 例如,如果要在转换中使用的第一转换表的表起始地址是区第一表,则使用虚拟地址的 RFX 部分引用区第一表内的区第一表表项。如果表起始地址是区第二表,则使用虚拟地址的 RSX 部分引用区第二表内的区第二表表项。如果表起始地址是区第三表,则使用虚拟地址的 RTX 部分引用区第三表内的区第三表表项。如果表起始地址是段表,则使用虚拟地址的 SX 部分引用段表内的段表表项。将一直引用连续的表,直到取回段表表项为止。

[0313] 一旦已取回段表表项,在 612,检查段表表项(STE)格式控制位以判定是否针对此特定虚拟地址启用了格式控制。如果 STE 格式控制为 0,则根据节点 614 发生动态地址转换。如果 STE 格式控制为 1,则根据节点 616 发生动态地址转换。

[0314] 动态地址转换(STE 格式控制为 0)

[0315] 现在参考图 7,其示出了当 STE 格式控制为 0 时从图 6 的节点 614 继续的流程图。

[0316] 在 710,从段表表项获取页表的起始地址。在 712,使用虚拟地址的 PX 部分引用页表中的页表表项。在 714,从页表表项获取页帧实地址(PFRA)。从页表表项获取无效(I)

位。在 716, 如果无效 (I) 位为 1, 则在 718 无法使用此页表表项继续虚拟地址转换, 因为此表项已被标记为无效。使用此页表表项的进一步虚拟地址转换将停止 (722)。在 716, 如果无效 (I) 位为 0, 则在 720 将页帧实地址 (PFRA) 与虚拟地址的 BX 部分组合以生成实地址。可以进一步对实地址执行添加前缀操作以形成绝对地址。在 724, 使用实地址访问由转换后的虚拟地址寻址的数据块。

[0317] 动态地址转换 (STE 格式控制为 1)

[0318] 现在参考图 8, 其示出了从图 6 的节点 616 继续的流程图。

[0319] 在 810, 从段表表项的一部分获取段帧绝对地址 (SFAA)。从段表表项获取无效 (I) 位。在 812, 如果无效 (I) 位为 1, 则在 814, 无法使用此段表表项进行进一步虚拟地址转换, 因为此表项已被标记为无效。在一个实施例中, 向请求转换的程序实体返回异常代码。使用此段表表项的此虚拟地址的进一步转换将停止 (818)。

[0320] 在 812, 如果无效 (I) 位为 0, 则在 816, 将段帧绝对地址 (SFAA) 与虚拟地址的 PX 和 BX 部分组合以生成指向主存储装置中或存储器中的所需大数据块的绝对地址。在 820, 访问由转换后的虚拟地址寻址的所需大数据块。

[0321] 在每个表表项处, 检查无效位以确定所获取的表表项的有效性。在此将进一步讨论保护通过转换后的虚拟地址寻址的数据块的其他转换保护机制。

[0322] 在另一实施例中, 从转换表表项获取限制信息。所述限制信息用于将访问限于虚拟地址范围的限定部分。此后, 仅对于地址范围的限定部分才允许访问由转换后的地址寻址的所需大数据块。所述限制信息是表偏移或表长度中的任意一个。

[0323] 在另一实施例中, 在虚拟地址转换中使用的信息被存储在至少一个转换后备缓冲器中。使用转换后备缓冲器中存储的信息而不是转换表层次结构来执行后续虚拟地址到主存储装置中数据块的绝对地址的后续转换。

[0324] 在另一实施例中, 如果转换不是机器体系结构所固有的, 则标识预定软件例程以仿真转换。所述预定软件例程包含多个指令。将执行所述预定软件例程。

[0325] 寻址后的数据块的保护

[0326] 如此处所述, 使用增强的 DAT 工具转换虚拟地址之后, 可以对由转换后的虚拟地址寻址的主存储装置或存储器中的所需数据块执行其他保护机制。

[0327] DAT 保护

[0328] DAT 保护功能通过使用每个页表表项和段表表项以及 (当安装了增强的 DAT 工具时) 每个区表表项中的 DAT 保护位来控制对虚拟存储装置的访问。它可防止不正确的存储。

[0329] 页表表项的 DAT 保护位 (位 54) 控制是否允许存储到对应的 4KB 页。当此位为 0 时, 允许取回和存储两者; 当此位为 1 时, 仅允许取回。当尝试存储到被保护页时, 此页的内容保持不变, 将取消操作单元或指令执行, 并发生程序保护中断。

[0330] 段表表项的 DAT 保护位 (位 54) 控制是否允许存储到对应的 1MB 段, 如下所示:

[0331] 当不应用增强的 DAT, 或者应用增强的 DAT 并且 STE 格式控制为 0 时, 段表表项的 DAT 保护位被视为与由段表表项指定的页表中每个表项的 DAT 保护位位置进行“或”。因此, 当段表表项 DAT 保护位为 1 时, 效果就像 DAT 保护位在指定页表中的每个表项内都为 1 那样。

[0332] 当应用增强的 DAT 并且 STE 格式控制为 1 时,段表表项的 DAT 保护位控制是否允许存储到对应的 1MB 段。当此位为 0 时,允许取回和存储两者 ;当此位为 1 时,仅允许取回。当尝试存储到被保护段时,此段的内容保持不变,将取消操作单元或指令执行,并发生程序保护中断。

[0333] 当应用增强的 DAT 时,区表表项的 DAT 保护位控制是否允许存储到对应的区 (多个)。区表表项中的 DAT 保护位被视为与在转换中使用的任何后续区表表项和段表表项的 DAT 保护位位置进行“或”。当 STE 格式控制位为 0 时,DAT 保护位进一步传播到页表表项。

[0334] DAT 保护应用到使用虚拟地址的所有存储型引用。

[0335] 现在参考图 9,其示出了从转换中使用的段表表项获取 DAT 保护位的一个实施例的流程图。

[0336] 在 902,获取要转换的虚拟地址。在 904,获取在虚拟地址转换中使用的第一转换表的起始地址。在 906,通过先前讨论的方式使用虚拟地址的 一部分引用转换表中的表项。在 908,从所获取的段帧绝对地址 (SFAA) 或页帧实地址 (PFRA) 生成实地址或绝对地址。

[0337] 在 910,从虚拟地址转换中使用的转换表的每个表项获取 DAT 保护 (P) 位。应该理解,DAT 保护的有效级别是从转换中使用的每个表项获取的多个 P 位的逻辑“或”。在 912,检查 DAT 保护位以判定由转换后的虚拟地址寻址的数据块是否已被 DAT 保护。在 912,如果 DAT 保护 (P) 位为 0,则在 914 允许执行到由转换后的虚拟地址寻址的数据块的存储操作。在 912,如果 DAT 保护 (P) 位为 1,则在 916 阻止执行到由转换后的虚拟地址寻址的数据块的存储操作。备选地,由 DAT 保护字段指示的保护级别应用到使用虚拟地址的所有存储型引用。如果启用了 DAT 保护字段,则在尝试存储到由转换后的虚拟地址寻址的所需数据块时,将指示保护异常状况。

[0338] 键控保护

[0339] 当键控保护应用到存储访问时,仅当存储键与存储访问请求所关联的访问键匹配时,才允许存储 ;当键匹配或者当存储键的取回保护位为 0 时,允许取回。当存储键的四个访问控制位与访问键相同或者当访问键为 0 时,则键被称为匹配。下表概述了保护操作。

[0340]

条件		是否允许对存储装置的访问	
存储键的取回保护位	键关系	取回	存储
0	匹配	是	是
0	不匹配	是	否
1	匹配	是	是
1	不匹配	否	否
解释:			
匹配 存储键的四个访问控制位与访问键相同, 或者访问键为 0。			

[0341]

是	允许访问。
否	不允许访问。取回时，信息不可用于程序；存储时，存储位置的内容不变。

[0342] 保护操作概述

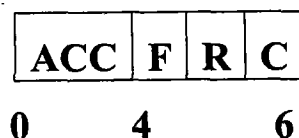
[0343] 当 CPU 启动对存储装置的访问并且应用键控保护时,PSW 键是访问键,但对于 MOVE TO SECONDARY 和 MOVE WITH DESTINATION KEY 的第一操作数,对于 MOVE TO PRIMARY、MOVE WITH KEY 和 MOVE WITH SOURCE KEY 的第二操作数以及对于 MOVE PAGE 的第一或第二操作数,访问键在通用寄存器中指定。PSW 键占用当前程序状态字的位位置 8-11。

[0344] 当由于键控保护而禁止 CPU 访问时,指令执行被终止,并发生程序保护异常中断。但是,可以取消操作单元或指令执行。

[0345] 存储键

[0346] 存储键与配置中可用的存储装置的每个 4KB 块关联。存储键不是可寻址的存储装置的一部分。在一个实施例中,存储键具有以下格式:

[0347]



[0348] 存储键格式

[0349] 存储键中的位位置按如下方式进行分配:

[0350] 访问控制位 (ACC):如果引用受到键控保护,则当存储信息以及从具有取回保护的位置取回信息时,需要将四个访问控制位与四位访问键进行匹配。

[0351] 取回保护位 (F):如果引用受到键控保护,则取回保护位控制是否将键控保护应用到取回型引用:0 指示仅监视存储型引用并允许具有任何访问键的取回;1 指示键控保护同时应用到取回和存储。

[0352] 引用位 (R):引用位通常在每次引用对应存储块中的位置以存储或取回信息时被设为 1。

[0353] 更改位 (C):更改位在每次将信息存储到对应存储块中的位置时被设为 1。

[0354] 当应用增强的 DAT 时,以下附加条件生效:

[0355] 当 STE 格式控制 (FC,在转换中使用的段表表项的位 53) 为 0 时,在转换中使用的页表表项的位 55 是页的更改记录忽略 (CO)。当页表表项中的 CO 位为 1 时,无法针对到页的任何存储操作预测更改记录。

[0356] 当段表表项中的格式控制位 (FC) 为 1 时,应用以下内容:

[0357] 段表表项的位位置 47 包含 ACCF 有效性控制。ACCF 有效性控制确定 STE 中的访问控制位和取回保护位的有效性。当 ACCF 有效性控制为 0 时,键控保护针对与地址对应的 4K 字节块使用存储键中的访问控制位和取回保护位。

[0358] 当 ACCF 有效性控制为 1 时,段表表项的位位置 48-52 包含段的访问控制位和取回保护位。在确定对存储操作数的可访问性时,对于组成段的 4K 字节块,无法预测是检查 STE

的位 48-52, 还是检查单个存储键的位 0-4。

[0359] 段表表项的位 55 是段的更改记录忽略 (CO)。当段表表项中的 CO 位为 1 时, 无法预测是否针对到段的任何存储操作设置更改位。

[0360] 存储键访问

[0361] 对存储键的引用的处理方式如下:

[0362] 每当引用存储装置并且对所述引用应用键控保护时, 四个访问控制位和与存储位置关联的取回保护位被同时检查并该检查与引用所述存储位置同时。当 (1) 不应用增强的 DAT, (2) 应用增强的 DAT 但借助其中 STE 格式控制为 0 的段表表项访问存储装置, 或 (3) 应用增强的 DAT, 借助其中 STE 格式控制为 1 的段表表项访问存储装置, 但 ACCF 有效性控制为 0, 则对于 4K 字节块, 访问控制位和取回保护位位于存储键的位 0-4 中。当应用增强的 DAT 并且借助其中 STE 格式控制和 ACCF 有效性控制均为 1 的段表表项访问存储装置时, 无法预测是存储键的位 0-4 还是段表表项的位 48-52 提供访问控制位和取回保护位。此外, 当段表表项提供访问控制位和取回保护位时, 可以使用来自转换后备缓冲器的缓冲的副本。

[0363] 当应用增强的 DAT, 并且 (a) 在由 DAT 使用的页表表项中, STE 格式控制为 0 而更改记录忽略为 1 或 (b) 在由 DAT 使用的段表表项中, STE 格式控制为 1 而更改记录忽略为 1 时, 无法预测在执行存储操作时, CPU 是否设置更改位。更改记录忽略可缓冲在 PTE 或 STE 的转换后备缓冲器副本中。

[0364] 当未安装条件型 SSKE 特性时, SET STORAGE KEY EXTENDED 指令导致在存储键中同时设置所有 7 个位。当安装了条件型 SSKE 特性时, SET STORAGE KEY EXTENDED 指令可用于根据程序指定的准则设置全部或部分存储键。

[0365] INSERT STORAGE KEY EXTENDED 指令针对 4K 字节块提供存储键的位 0-6 的一致映像。类似地, 指令 INSERT VIRTUAL STORAGE KEY 和 TEST PROTECTION 提供访问控制位和取回保护位的一致映像。

[0366] 指令 RESET REFERENCE BIT EXTENDED 仅修改引用位。存储键的所有其他位保持不变。同时检查引用位和更改位以设置条件代码。

[0367] 由引用位提供的引用记录不一定准确。但是, 在大多数情况下, 引用记录大致与相关的存储引用一致。可以在未发生存储时设置更改位。

[0368] 如其他 CPU 所观察到的, 由显式操纵存储键的指令 (INSERT STORAGE KEY EXTENDED、INSERT VIRTUAL STORAGE KEY、RESET REFERENCE BIT EXTENDED 以及 SET STORAGE KEY EXTENDED) 执行的存储键取回和存储在它们本身之间以及在存储操作数引用之间被排序, 就像存储键访问本身分别是存储操作数取回和存储那样。

[0369] 访问控制和取回保护 (ACC/F)

[0370] 可以进一步借助段表表项中提供的访问控制位和取回保护位来保护主存储装置或存储器中的数据块。

[0371] 访问有效性 (AV) 和访问控制 (ACC)

[0372] 现在参考图 10, 其示出了确定要应用于由转换后的虚拟地址寻址的所需数据块的访问控制保护的流程图的一个实施例。

[0373] 动态地址转换继续到其中已从转换中使用的段表取回段表表项的点, 如图 6 中所述继续到节点 616。地址转换如图 8 中所述继续到节点 822, 其中已获取存储装置或存储器

中所需数据块的绝对地址。图 10 中的流程从节点 822 开始。

[0374] 在 1002, 从段表表项获取访问有效性 (AV) 字段。在 1004, 判定是否启用段表表项中的访问有效性字段 (AV = 1)。如果访问有效性字段为 0, 则段表表项中的访问控制 (ACC) 字段无效。在 1006, 忽略访问控制字段的各位。如果启用了访问有效性字段, 则访问控制字段有效。在 1008, 从段表表项获取访问控制字段的各位。在 1010, 将由段表表项中的 ACC 字段指示的访问控制保护级别应用到由转换后的虚拟地址寻址的所需数据块。

[0375] 在另一实施例中, 仅当访问控制字段与程序实体 (例如程序状态字或所执行程序指令的操作数) 所提供的访问键匹配时, 才允许到所需数据块的存储操作。如果在所需数据块处尝试执行存储并且访问控制字段与访问键不匹配, 则优选地指示保护异常。

[0376] 在另一实施例中, 如果启用了访问有效性字段, 则当在所需数据块处尝试执行取回操作或存储操作并且段访问控制字段与程序访问控制字段不匹配时, 将指示保护异常。

[0377] 访问有效性 (AV) 和取回保护 (F)

[0378] 现在参考图 11, 其示出了确定要应用于由转换后的虚拟地址寻址的所需数据块的取回保护的流程图的一个实施例。

[0379] 动态地址转换继续到其中已从转换中使用的段表取回段表表项的点, 如图 6 中所述继续到节点 616。地址转换如图 8 中所述继续到节点 822, 其中已获取存储装置或存储器中所需数据块的绝对地址。图 11 中的流程从节点 822 开始。

[0380] 在 1102, 从段表表项获取访问有效性 (AV) 字段。在 1104, 判定是否启用段表表项中的访问有效性字段 (AV = 1)。如果在 1104 访问有效性字段为 0, 则在 1106 忽略取回保护位。如果在 1104 启用了访问有效性字段, 则在 1108 从段表表项获取取回保护位。在 1110, 将由段表表项中的 F 字段指示的取回保护级别应用到由转换后的虚拟地址寻址的所需数据块。仅当取回保护字段与访问键匹配时, 才允许从由转换后的虚拟地址寻址的数据块执行取回操作。访问键由程序状态字或所执行程序指令的操作数提供。仅当已禁用取回保护字段或者仅当已启用取回保护字段并且与虚拟地址关联的访问控制字段与访问键相同时, 才允许从所需数据块执行取回操作。如果访问键为 0, 则允许从所需数据块执行取回。

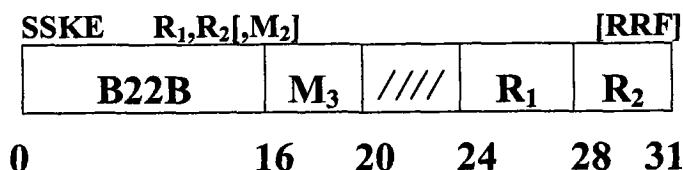
[0381] 在另一实施例中, 如果程序访问键为 0, 则允许执行到所需存储块的存储操作以及从所需存储块执行取回操作。

[0382] 在另一实施例中, 如果启用了访问有效性字段和取回保护字段, 则响应于从所需存储块尝试执行取回操作并且段访问控制字段与程序访问控制字段不匹配, 将指示保护异常。

[0383] SET STORAGE KEY EXTENDED (SSKE)

[0384] 可以借助 SET STORAGE KEY EXTENDED (SSKE) 指令设置存储键。在一个实施例中, SSKE 指令具有以下格式:

[0385]



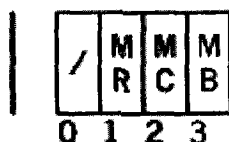
[0386] SSKE 指令的格式

[0387] 由第一操作数寄存器中的值替换一个或多个 4K 字节块的存储键。当安装了条件型 SSKE 工具时,可以绕过键设置操作的特定功能。当未安装条件型 SSKE 工具,或者已安装条件型 SSKE 工具并且 M3 字段的 MR 和 MC 位均为 0 时,由来自通用寄存器 R1 的位替换通过通用寄存器 R2 的内容寻址的 4K 字节块的存储键。指令在不更改条件代码的情况下完成。

[0388] 当安装了条件型 SSKE 工具并且 MR 位和 MC 位为 1 时,将由通用寄存器 R2 的内容寻址的存储键的访问控制位、取回保护位以及可选地引用位和更改位与通用寄存器 R1 中的对应位进行比较。如果所比较的位相同,则不对键进行更改;否则,由通用寄存器 R1 中的对应位替换键的选定位。将任何修改之前的存储键插入通用寄存器 R1,并且由条件代码指示结果。

[0389] 当安装了增强的 DAT 工具时,可以在多块控制(如下所述)的控制下,针对同一 1MB 块内的多个 4K 字节块的存储键重复上述操作。在一个实施例中,M3 字段具有以下格式:

[0390]



[0391] M3 字段的格式

[0392] M3 字段的位定义如下:

[0393] 保留:位 0 被保留。

[0394] 引用位更新掩码(MR):MR 位(M3 字段的位 1)控制是否可以绕过对存储键中引用位的更新,如下所述。

[0395] 更改位更新掩码(MC):MC 位(M3 字段的位 2)控制是否可以绕过对存储键中更改位的更新,如下所述。

[0396] 多块控制(MB):MB 位(M3 字段的位 3)控制是否可以设置存储装置的多个 4K 字节块的存储键,如“在多个 4K 字节块中设置存储键”中所述。

[0397] 当未安装增强的 DAT 工具时,M3 字段的位位置 3 被保留。当安装了条件型 SSKE 工具时,按以下方式处理:

[0398] 当 MR 位和 MC 位(M3 字段的位 1 和 2)均为 0 时,指令如同未安装条件型 SSKE 工具那样完成。由来自通用寄存器 R1 的位替换通过通用寄存器 R2 的内容寻址的 4K 字节块的存储键,并且指令在不更改条件代码的情况下完成。

[0399] 当 MR 位和 MC 位为 1 时,按以下方式处理:

[0400] 在任何修改之前,将通过通用寄存器 R2 寻址的 4K 字节块的存储键的内容放置在通用寄存器 R1 的位位置 48-54 中,并且将通用寄存器 R1 的位 55 设为 0。寄存器的位 0-47 和 56-63 保持不变。如果当取回存储键时检测到无效的检查块代码(CBC),则(a)由通用寄存器 R1 的位 56-62 替换 4K 字节块的整个存储键,(b)无法预测通用寄存器 R1 的位位置 48-55 的内容,以及(c)指令通过设置条件代码 3 完成。

[0401] 将指定的 4K 字节块的存储键的访问控制位和取回保护位与通用寄存器 R1 的位 56-60 中的对应字段进行比较。如果相应字段不同,则由来自通用寄存器 R1 的位替换 4K 字节块的整个存储键,并且指令通过设置条件代码 1 完成。当存储键中的访问控制位和取回

保护位与通用寄存器 R1 中的相应位相同时,将按以下方式继续处理。

[0402] 当 MR 位和 MC 位均为 1 时,指令通过设置条件代码 0 完成。在这种情况下存储键保持不变。

[0403] 当 MR 位为 0 并且 MC 位为 1 时,将指定的 4K 字节块的存储键的引用位与通用寄存器 R1 的位 61 进行比较。如果位相同,则指令通过设置条件代码 0 完成。在这种情况下存储键保持不变。如果位不同,则 (a) 由通用寄存器 R1 中的位替换指定的 4K 字节块的整个存储键,并且指令通过设置条件代码 1 完成;或 (b) 由通用寄存器 R1 的位 61 替换存储键的引用位,键的更改位无法预测,并且指令通过设置条件代码 2 完成。无法预测 是设置条件代码 1 还是 2。

[0404] 当 MC 位为 0 并且 MR 位为 1 时,将指定的 4K 字节块的存储键的更改位与通用寄存器 R1 的位 62 进行比较。如果位相同,则指令通过设置条件代码 0 完成。在这种情况下存储键保持不变。如果位不同,则 (a) 由通用寄存器 R1 中的位替换指定的 4K 字节块的整个存储键,并且指令通过设置条件代码 1 完成;或 (b) 由通用寄存器 R1 的位 62 替换存储键的更改位,键的引用位无法预测,并且指令通过设置条件代码 2 完成。无法预测是设置条件代码 1 还是 2。

[0405] 当未安装增强的 DAT 工具,或者已安装此工具但多块控制为 0 时,通用寄存器 R2 包含实地址。当安装增强的 DAT 工具并且多块控制为 1 时,通用寄存器 R2 包含绝对地址。在 24 位寻址模式下,通用寄存器 R2 的位 40-51 指定实存储装置或绝对存储装置中的 4K 字节块,并且忽略寄存器的位 0-39 和 52-63。在 31 位寻址模式下,通用寄存器 R2 的位 33-51 指定实存储装置或绝对存储装置中的 4K 字节块,并且忽略寄存器的位 0-32 和 52-63。在 64 位寻址模式下,通用寄存器 R2 的位 0-51 指定实存储装置或绝对存储装置中的 4K 字节块,并且忽略寄存器的位 52-63。由于是实地址或绝对地址,因此指定存储块的地址不进行动态地址转换。对存储键的引用不会出现保护异常。

[0406] 从通用寄存器 R1 的位位置 56-62 获取新的 7 位存储键值或其选定位。忽略寄存器的位位置 0-55 和 63 的内容。当安装了条件型 SSKE 工具并且 MR 位和 MC 位均为 1 时,位位置 63 应包含 0;否则,将来程序可能无法兼容运行。

[0407] 在操作开始之前执行串行化和检查点同步功能并且在操作完成之后再次执行此功能,只是当安装了条件型 SSKE 工具并且结果条件代码为 0 时,无法预测是否在操作完成之后执行串行化和检查点同步功能。对于在此指令设置键之前或之后由任何 CPU 或通道程序完成的对指定 4K 字节块的任何存储访问,也分别在此指令的执行之前或之后,完成此块的存储键中的引用位和更改位到 1 的关联设置。

[0408] 在多个 4K 字节块中设置存储键

[0409] 当未安装增强的 DAT 工具,或者已安装此工具但多块控制为 0 时,如上所述设置单个 4K 字节块的存储键。当安装了增强的 DAT 工具并且多块控制为 1 时,可以设置 1MB 块内的多个 4K 字节块的存储键,从由第二操作数地址指定的块开始,并向右继续到每个连续的块,直到下一个 1MB 边界。在这种情况下,可中断 SET STORAGE KEY EXTENDED,并且按以下方式处理:

[0410] 当发生中断(不同于终止之后的中断)时,通用寄存器 R2 已被更新,因此指令在重新执行时将在中断点继续。如果 MR 位和 / 或 MC 位为 1,则条件代码无法预测;否则,条

件代码不变。

[0411] 当指令在不中断的情况下完成时,通用寄存器 R2 已被更新到下一个 1MB 边界。如果 MR 位和 / 或 MC 位为 1,则设置条件代码 3 ;否则,条件代码不变。

[0412] 对于上述两种情况之一,当 MR 位和 / 或 MC 位为 1 时,通用寄存器 R1 的位 48-55 无法预测。

[0413] 当发生多块处理并且 R1 字段和 R2 字段指定同一寄存器时,将第二操作数地址放置在此寄存器中。当在 24 位或 31 位寻址模式下发生多块处理时,将通用寄存器 R2 的位位置 32-63 中不是地址一部分的最左侧位设为 0 ;寄存器的位 0-31 不变。

[0414] 结果条件代码 :

[0415] 当未安装条件型 SSKE 工具,或者 M3 字段的 MR 位和 MC 位均为 0 时,条件代码保持不变。当安装了条件型 SSKE 工具并且 MR 位和 / 或 MC 位为 1 时,条件代码设置如下 :

[0416] 0- 未设置存储键

[0417] 1- 设置整个存储键

[0418] 2- 设置部分存储键

[0419] 3- 设置整个存储键 ;通用寄存器 R1 的位 48-55 无法预测。

[0420] 程序异常 :

[0421] 寻址 (由通用寄存器 R2 指定的地址)

[0422] 特权操作

[0423] 更改记录

[0424] 更改记录提供有关在主存储装置中替换页时必须将哪些页保存到辅助存储装置中的信息。更改记录使用存储键的更改位 (位 6)。每当存储访问导致对应存储块的内容被更改,以及 (a) 不应用增强的 DAT 或 (b) 应用增强的 DAT 并且满足以下两种条件之一时,将更改位设为 1 :

[0425] DAT 所使用的段表表项中的 STE 格式控制为 0,并且 DAT 所使用的页表表项中的更改记录忽略 (CO) 为 0。

[0426] DAT 所使用的段表表项中的 STE 格式控制为 1,并且 DAT 所使用的段表表项中的更改记录忽略 (CO) 为 0。

[0427] 不更改存储装置内容的存储访问可以将更改位设为 1,也可以不设为 1。如果禁止访问,则对于进行的存储尝试,不将更改位设为 1。具体而言 :

[0428] 对于 CPU,每当存储访问存在访问异常,或者每当存在优先级高于存储访问的访问异常优先级的异常时,将禁止存储访问。

[0429] 对于通道子系统,每当存储访问存在键控保护违反时,将禁止存储访问。

[0430] 更改记录始终处于活动状态并且针对所有对存储装置的存储访问发生,包括由任何 CPU (当由在此所述的更改记录忽略取消时除外)、任何操作者工具或通道子系统执行的存储访问。它针对机器所进行的隐式引用 (例如作为中断一部分的那些引用) 发生。

[0431] 对于以下指令的操作数不发生更改记录,因为这些指令直接修改存储键而不修改存储位置 :

[0432] RESET REFERENCE BIT EXTENDED

[0433] SET STORAGE KEY EXTENDED (更改位设为指定值)。

[0434] 当 CPU 重试时,不必将已从 0 更改为 1 的更改位恢复为 0。

[0435] 更改记录忽略 (CO)

[0436] 每当存储访问导致对应存储块的内容更改时,存储键的更改位被设为 1。不更改存储装置内容的存储访问可以将更改位设为 1,也可以不设为 1。如果禁止访问,则对于进行的存储尝试,不将更改位设为 1。更改记录忽略允许绕过存储键的更改位的设置。

[0437] 当应用增强的 DAT 并且通过 DAT 表表项转换虚拟地址时,在段表表项和页表表项两者的位 55 中提供更改记录忽略 (CO)。当段表表项的 STE 格式控制 (FC) 位 53 为 0 时,应用页表表项中的更改记录忽略。当 PTE 中的更改记录忽略为 0 时,针对到 4K 字节块的存储操作发生更改记录。当更改记录忽略为 1 时,无法预测是否针对到 4K 字节块的存储操作发生更改记录。当 STE 格式控制为 1 时,应用 STE 中的更改记录忽略。当 STE 中的更改记录忽略为 0 时,针对到段的 256 个 4K 字节块中任何一个的存储操作发生更改记录。当 STE 中的更改记录忽略为 1 时,无法预测是否针对段的 256 个 4K 字节块中的任何一个发生更改记录。

[0438] 更改记录忽略不应用到实地址或绝对地址,也不应用到通过实空间指定转换的虚拟地址。

[0439] 现在参考图 12,其示出了从在虚拟地址的动态地址转换中使用的段表表项获取更改记录忽略字段的一个实施例的流程图。

[0440] 动态地址转换继续到其中已从在转换中使用的段表取回段表表项的点,如图 6 中所述继续到节点 616。地址转换如图 8 中所述继续到节点 822,其中已获取存储装置或存储器中所需数据块的绝对地址。图 12 的流程从节点 822 开始。

[0441] 在 1210,从在转换中使用的段表表项获取更改记录忽略 (CO) 字段。更改记录忽略使能忽略存储键的更改位。在 1212,判定针对通过转换后的虚拟地址寻址的所需数据块执行的访问是存储型操作还是取回型操作。如果访问是取回型访问,则在 1213 不发生更改位的设置并且不应用更改记录忽略。如果访问是存储型操作,则在 1214 判定是否启用段表表项中的更改记录忽略。如果在 1214 未启用更改记录忽略,则在 1218 将与所需块关联的存储键的更改位设为 1。如果在 1214 启用了更改记录忽略,则在 1216 不设置关联的存储键的更改位。在 1220,指示已修改所需数据块。

[0442] 在另一实施例中,如果应用增强的 DAT 并且未启用 STE 中的格式控制,则从段表表项获取页表的起始地址。使用虚拟地址的页索引部分引用页表中的页表表项。从页表表项获取主存储装置中小数据块的页帧实地址以及页更改忽略字段。每个小数据块都具有包含更改位的关联存储键。小数据块的大小小于大数据块并包括 4KB(4096 字节)数据块。主存储装置中的所需实数据块的转换后地址包含页帧实地址和虚拟地址的字节索引部分的组合。判定对所需实数据块是执行存储型操作还是取回型操作。如果访问操作是取回型操作,则不应用更改记录忽略。如果访问是存储型操作,并且如果禁用与虚拟地址关联的更改记录忽略字段,则将与所需 4K 块关联的存储键的更改位设为 1。将提供已修改所需 4K 块的指示。

[0443] 在另一实施例中,如果启用了更改记录忽略字段,则无法预测与所需 4K 块关联的存储键的更改位。

[0444] 商业实现

[0445] 尽管此处提及了 **IBM**[®] 的 **z/Architecture**[®], 但是本发明的一个或多个方面可同样适用于其他采用可分页实体或类似结构的机器体系结构和 / 或计算环境。

[0446] 此处披露的 eDAT 工具以及其他格式、指令和属性的商业实现可以通过硬件实现, 也可以由例如使用汇编语言的编程人员 (如操作系统编程人员) 实现。此类编程指令可存储在旨在在诸如 **IBM**[®] System z server 之类的计算环境中本地地执行, 或备选地在执行其他体系结构的机器中本地地执行的存储介质中。所述指令可以在现有和未来的 **IBM**[®] 服务器中以及在其他机器或主机上进行仿真。它们可以在其中总体执行处于仿真模式的机器中执行。

[0447] 本发明的一个或多个方面可同样适用于例如虚拟机仿真, 其中一个或多个可分页实体 (例如客户机) 在一个或多个处理器上执行。作为一个实例, 可分页客户机由 “**IBM**[®] Syetem/370Extended Architecture (**IBM**[®] 系统 /370 扩展体系结构, **IBM**[®] 出版号 :SA22-7095, 1985, 其全部内容在此引入作为参考)” 中描述的 Start Interpretive Execution (SIE) 体系结构定义。

[0448] 在仿真模式中, 解码被仿真的特定指令, 然后执行子例程以便如子例程或驱动程序中那样实现各个指令, 或者使用其他某种技术提供特定硬件的驱动程序, 本领域的技术人员在理解了此处的说明之后便可理解此技术。各种软件和硬件仿真技术在多个美国专利中进行了描述, 这些专利包括 :5, 551, 013、5, 574, 873、5, 790, 825、6, 009, 261、6, 308, 255 和 6, 463, 582, 在此引入上述每个专利作为参考。许多其他教导进一步示出了各种仿真具有目标机器体系结构的指令集的方法。

[0449] 其他变型和体系结构

[0450] 此处所述的各种实施例仅作为示例。这些实施例具有多种不偏离本发明的精神的变型。可以通过软件、固件、硬件或它们的某种组合实现本发明的一种或多种功能。本发明的各方面对于许多类型的环境是有利的, 其中包括具有多个区的其他环境以及非分区环境。进而, 可以没有中央处理器复合体, 但是仍存在多个耦合在一起的处理器。此处所述的各个方面可适用于单处理器环境。

[0451] 尽管此处描述了特定环境, 但再次地, 可以在不偏离本发明的精神的情况下实现这些环境的多种变型。例如, 如果对环境执行逻辑分区, 则此环境中可包括更多或更少的逻辑分区。进而, 可以存在多个耦合在一起的中央处理复合体。在不偏离本发明的精神的情况下只能做出某些变型。此外, 其他变型是可能的。

[0452] 尽管术语 “页” 用于指固定大小或预定大小的存储区域, 但是页大小 是可变的。类似地, 块大小是可变的。可存在不同大小的块和 / 或页。页也可以等同于块。可以备选地使用其他结构或以其他方式在软件和 / 或硬件中实现其他结构。进而, 在此处所述的示例中, 存在许多变型, 包括但不限于不同大小的字或地址 ; 不同数量的位 ; 不同顺序的位 ; 更多、更少或不同的位 ; 更多、更少或不同的字段 ; 不同顺序的字段 ; 不同大小的字段等。再次地, 这些仅提供为示例。许多变型是可能的。

[0453] 处理单元包括可分页实体, 例如客户机、主机、其他处理器、仿真器、虚拟机和 / 或其他类似的结构。缓冲器包括存储装置和 / 或存储器区域, 以及其他类型的数据结构, 其中包括但不限于阵列或可分页实体。表也包括其他数据结构。指令可以引用其他寄存器。而

且,页、段和 / 或区的大小可以不同于此处所述的大小。

[0454] 本发明的一个或多个方面可以包括在具有例如计算机可用或机器可读介质的制品(例如,一个或多个计算机程序产品)中。所述介质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)以提供或促进本发明的功能。所述制品可以包括为计算机系统的一部分或单独出售。此外,可提供可由机器读取的至少一个程序存储设备,所述程序存储设备包含至少一个可由所述机器执行以实现本发明的功能的指令程序。

[0455] 此处描述的流程图是示例性的。所述的这些图形或步骤或操作可以具有多种不偏离本发明的精神的变型。例如,所述步骤可以按照其他顺序执行,也可以添加、删除或修改步骤。所有这些变型均被视为要求保护的发明的一部分。

[0456] 尽管此处详细地描述和说明了本发明的实施例,但是对于本领域的技术人员显而易见的是,可以在不偏离本发明精神的情况下做出各种修改、添加、替换等,因此它们被视为在以下权利要求限定的发明范围之内。

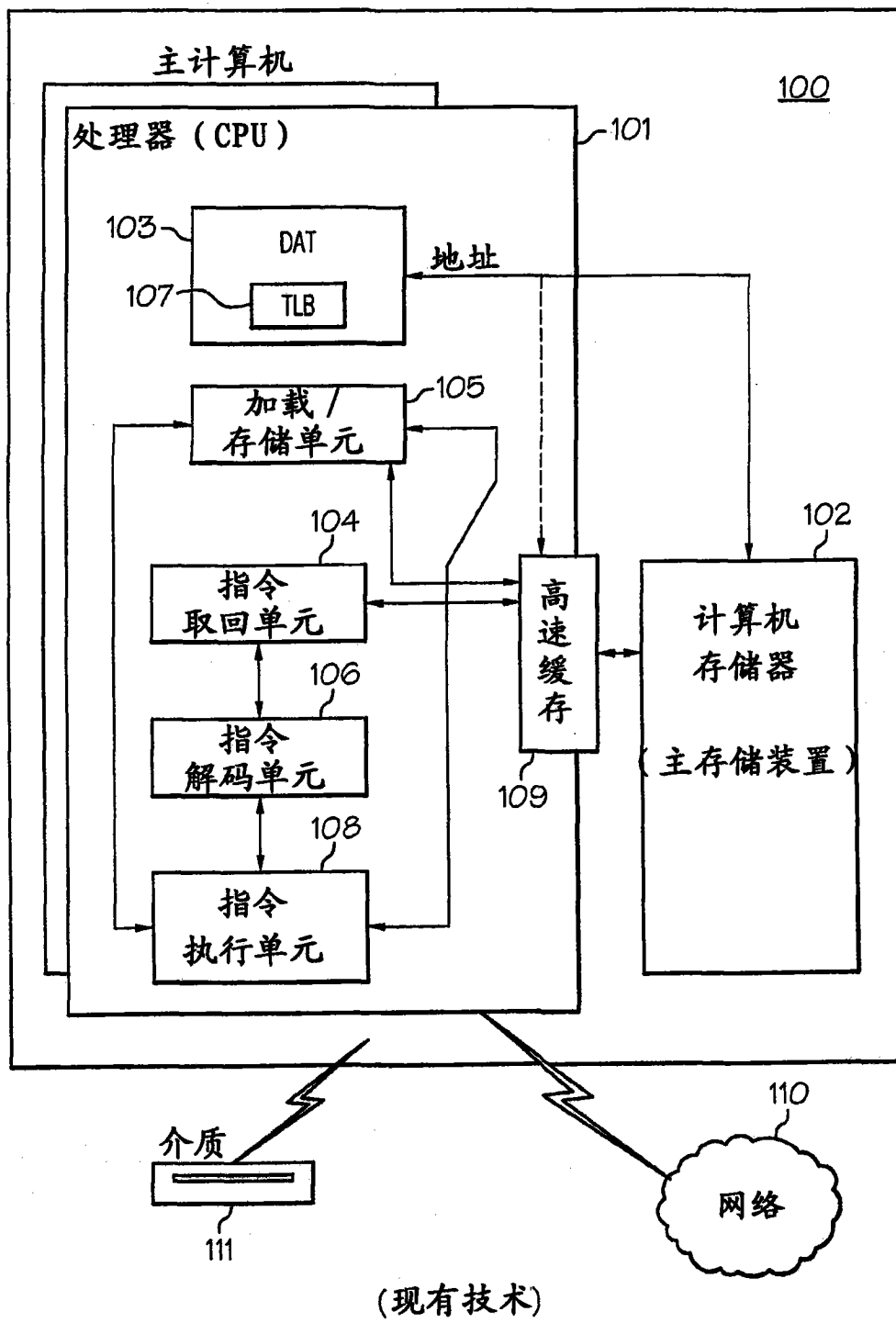


图 1

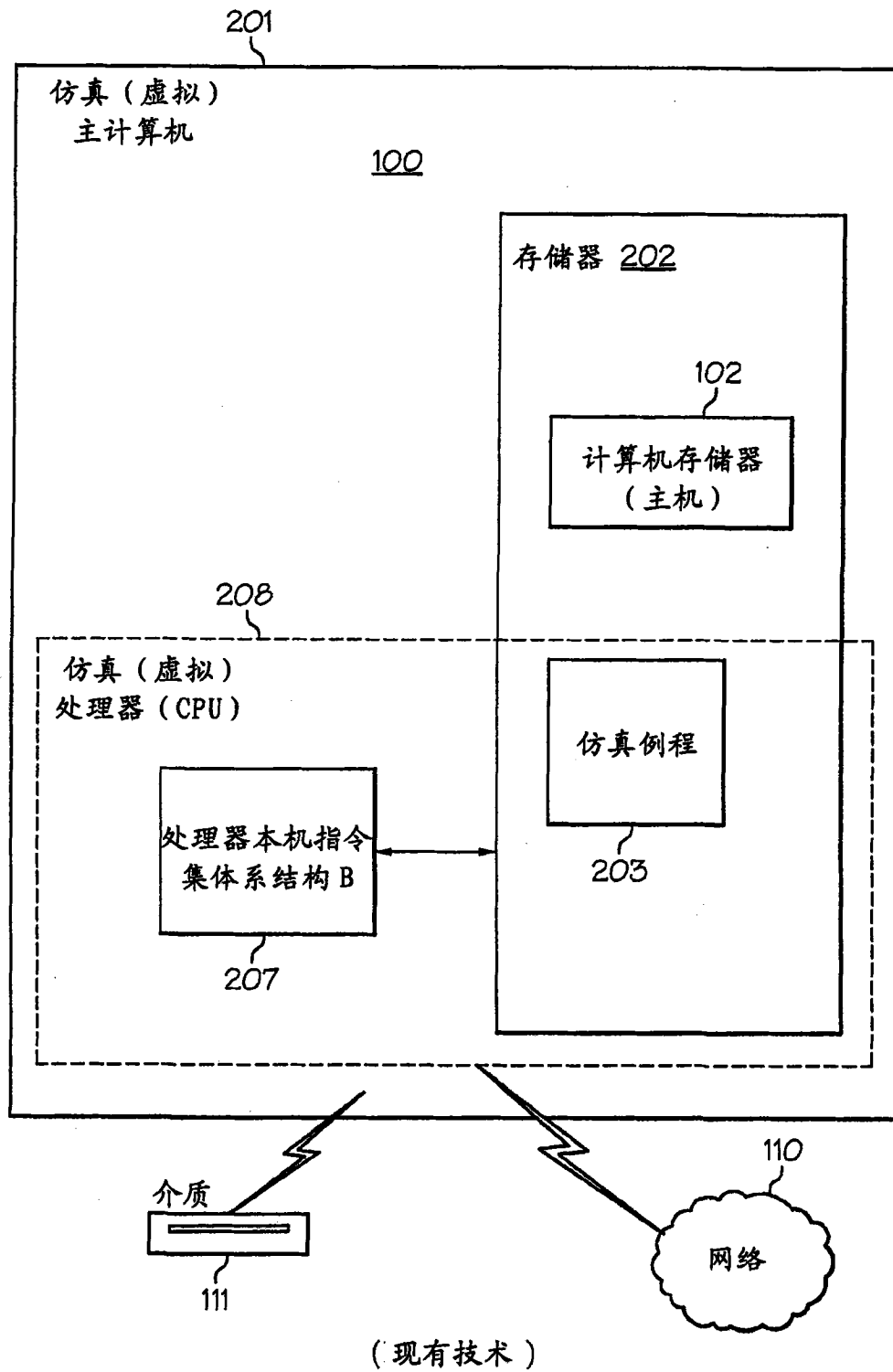
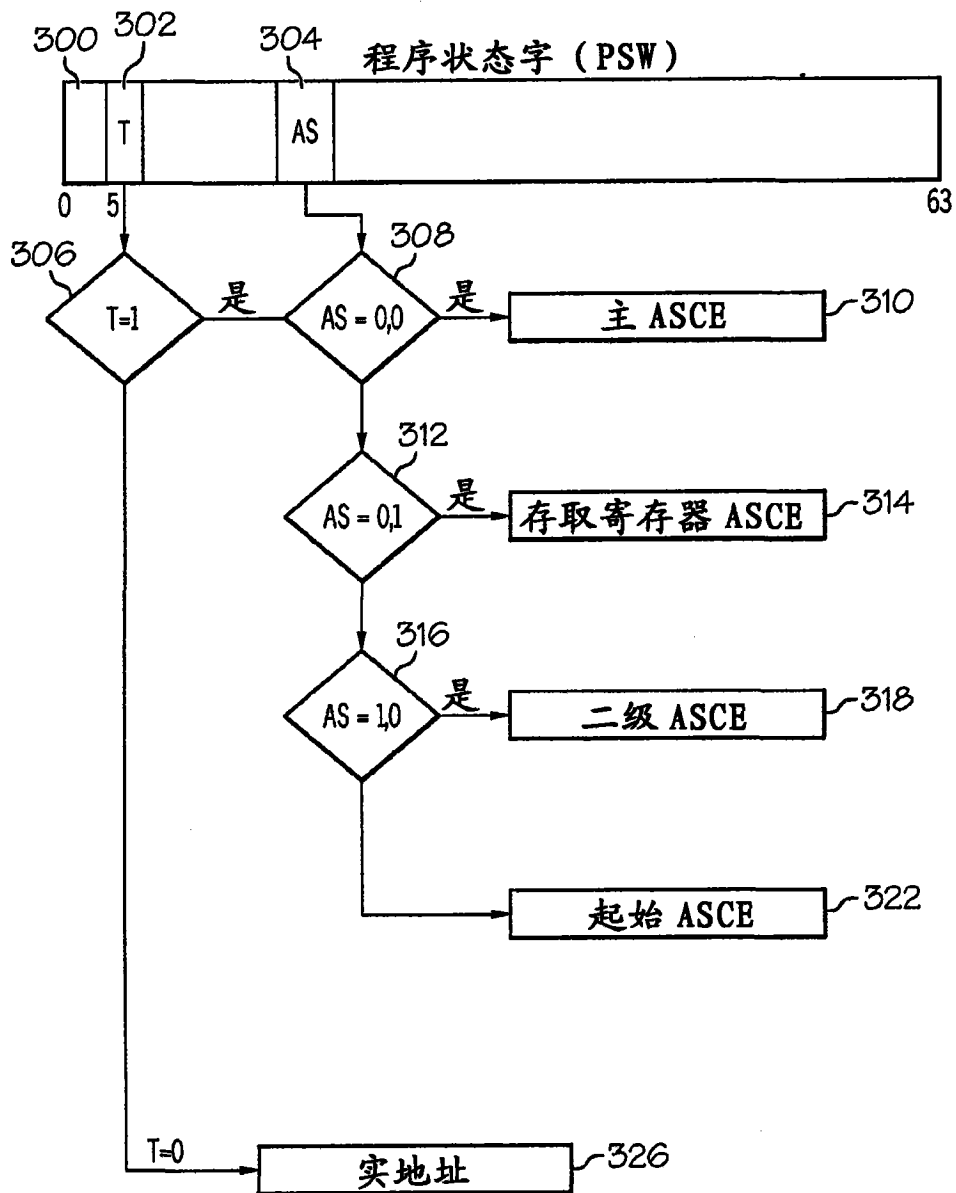
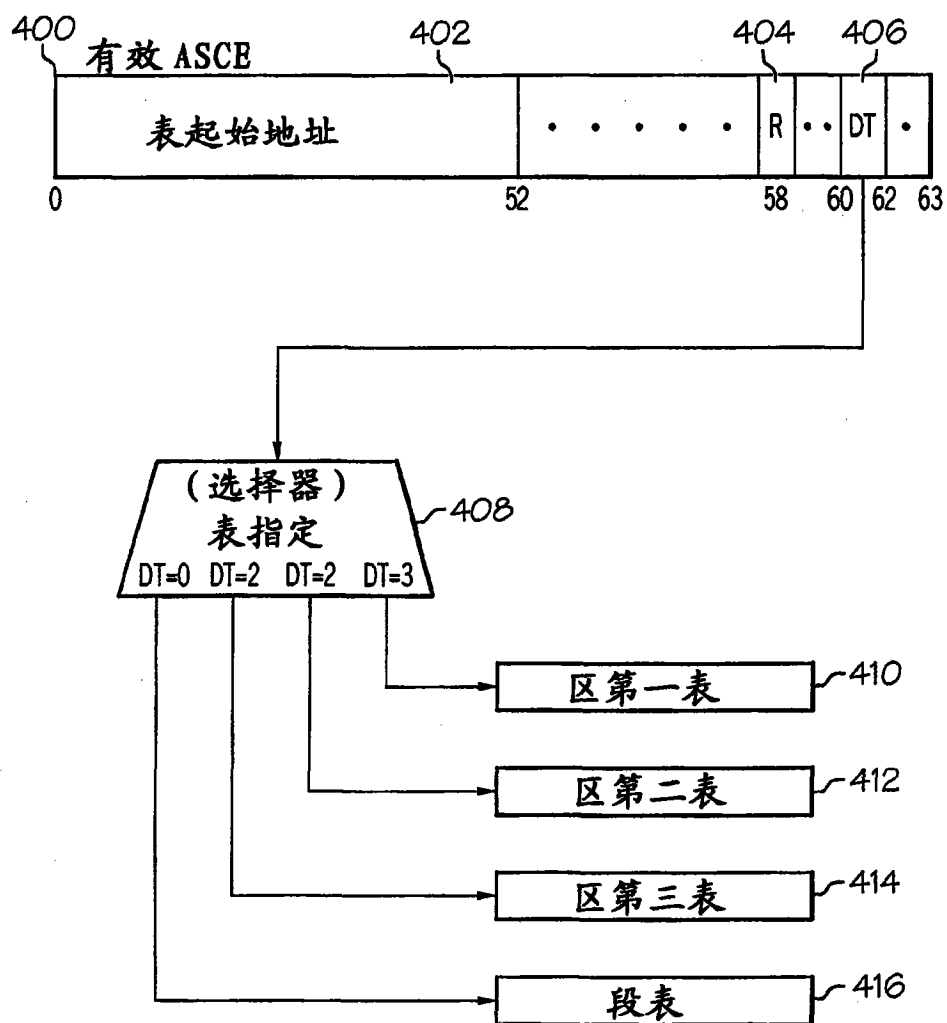


图 2



(现有技术)

图 3



(现有技术)

图 4

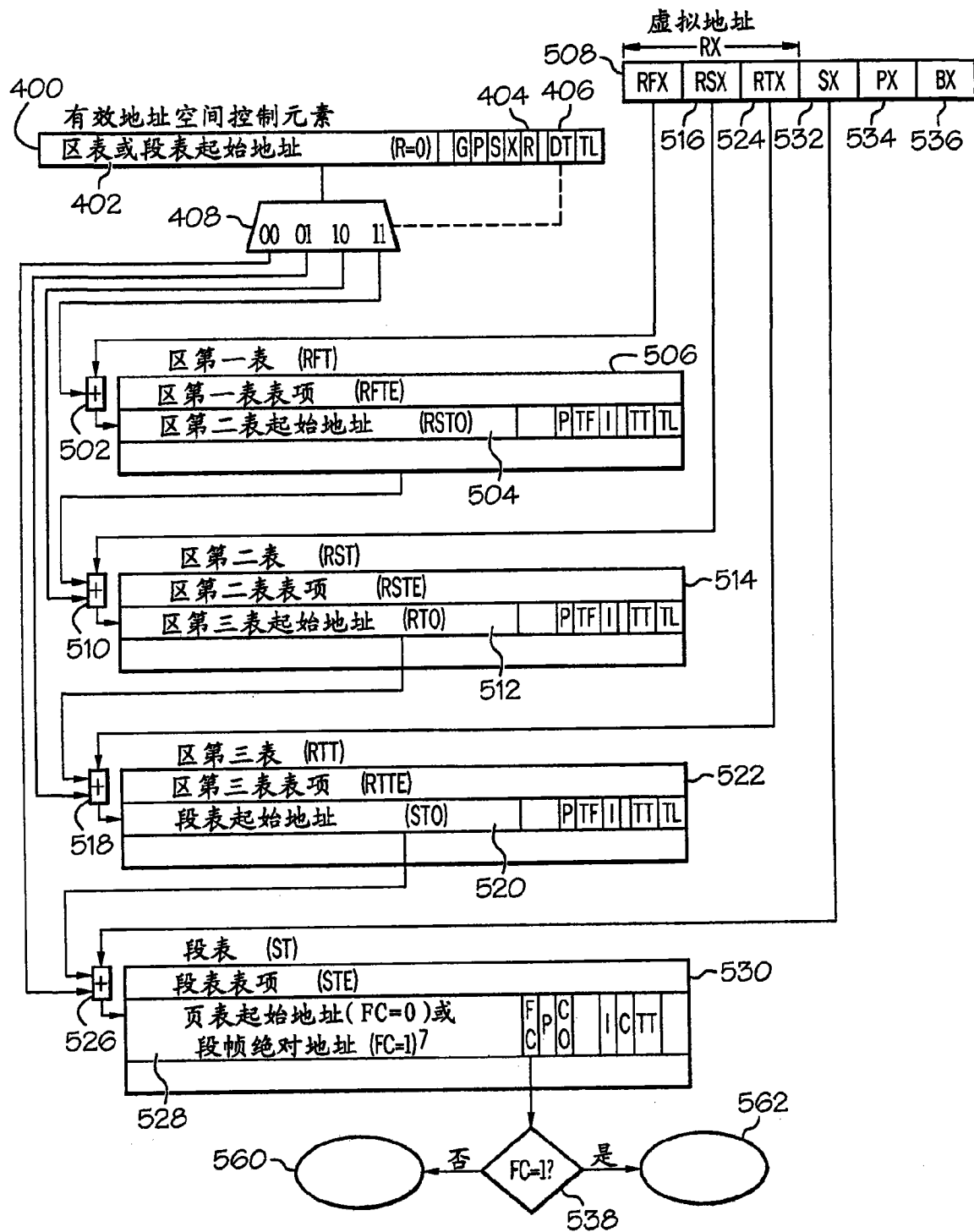


图 5A

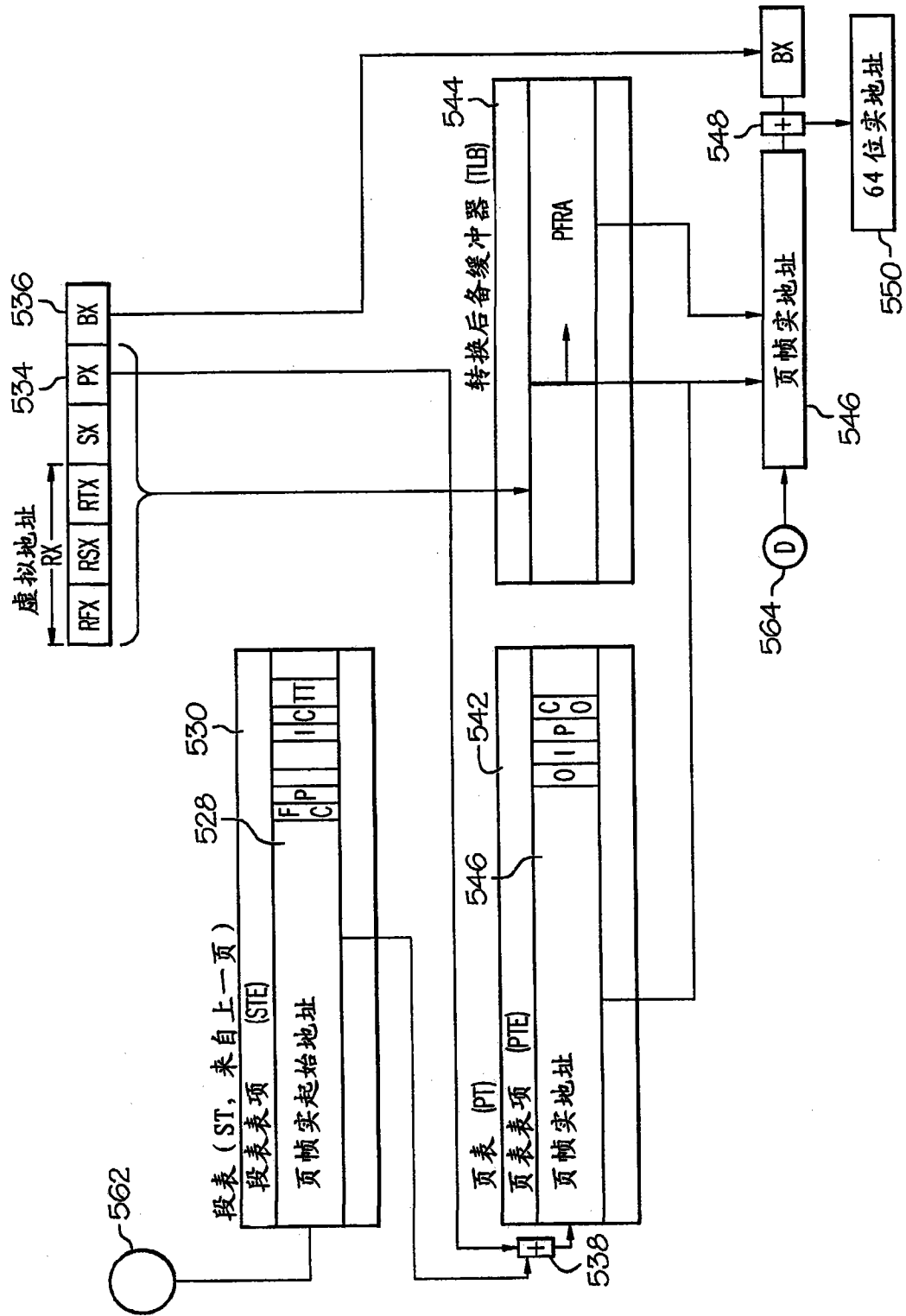


图 5B

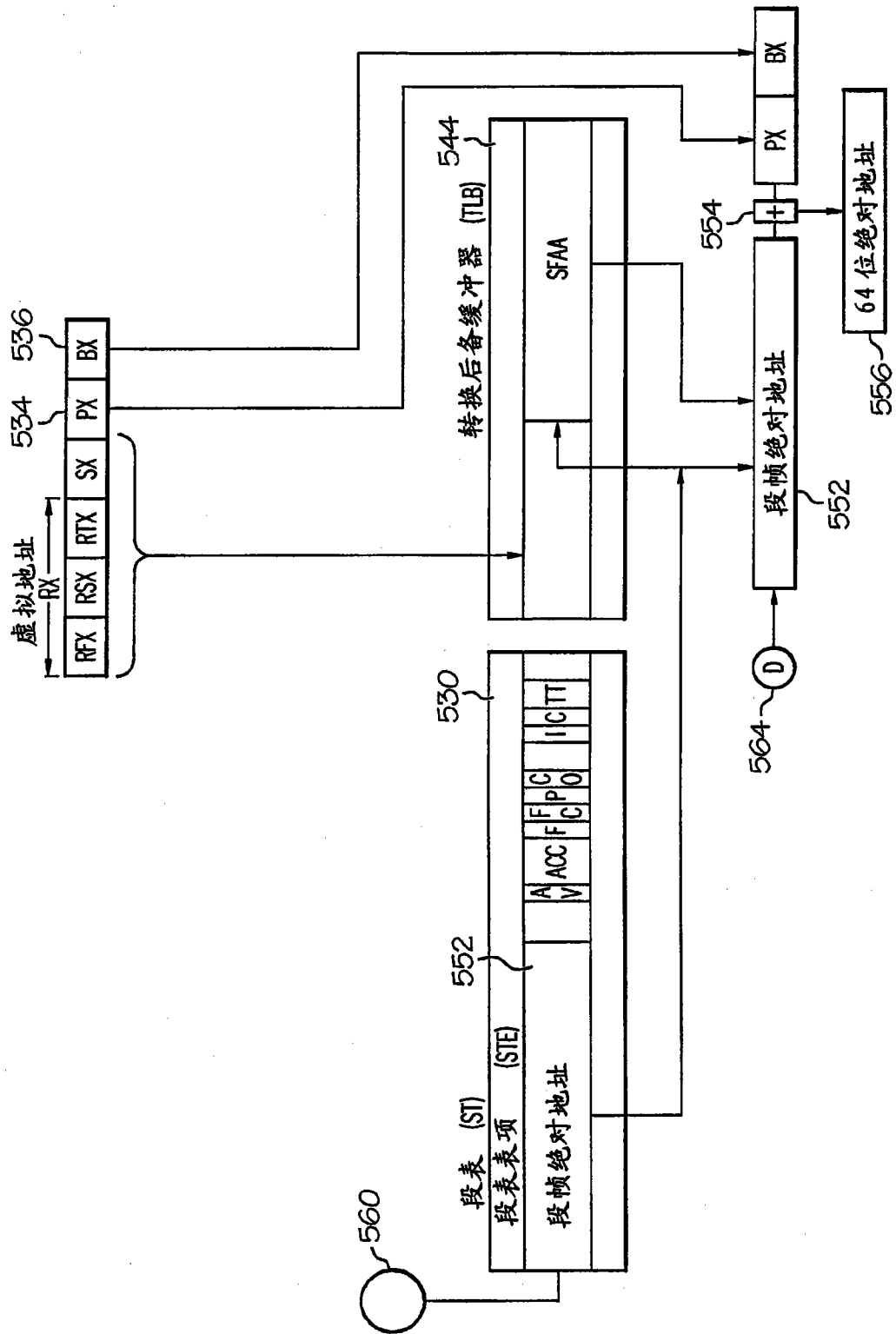


图 5C

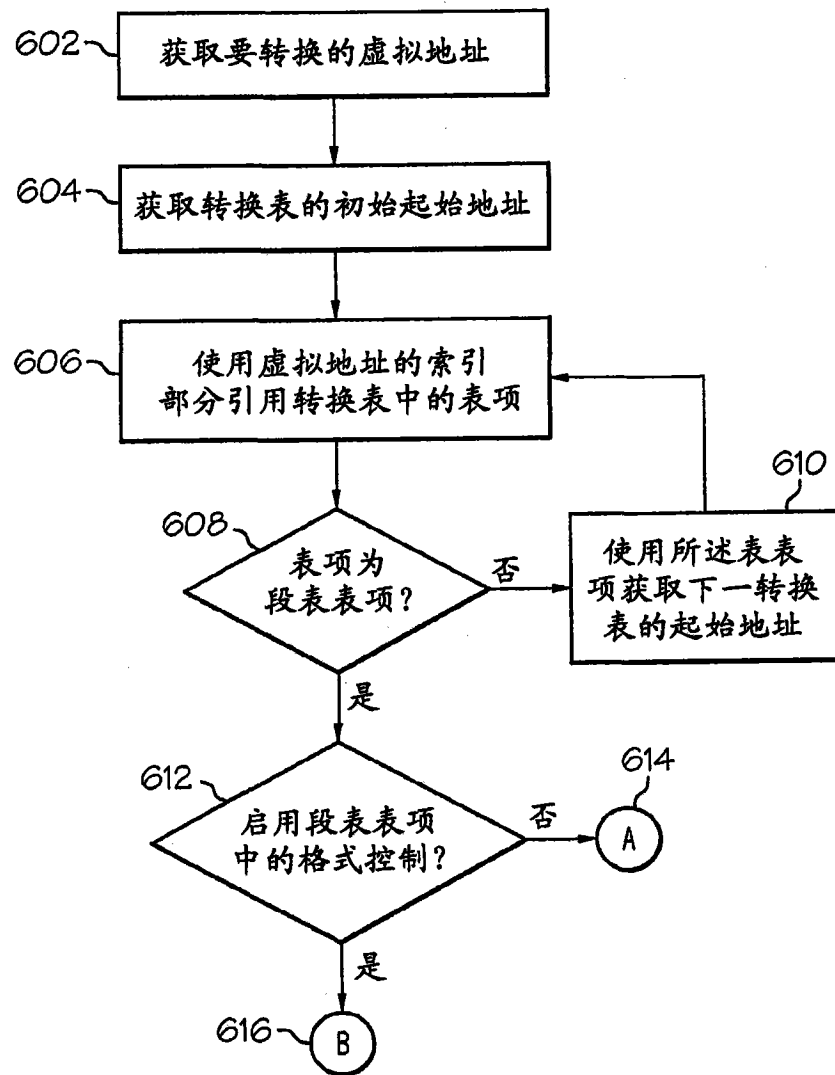
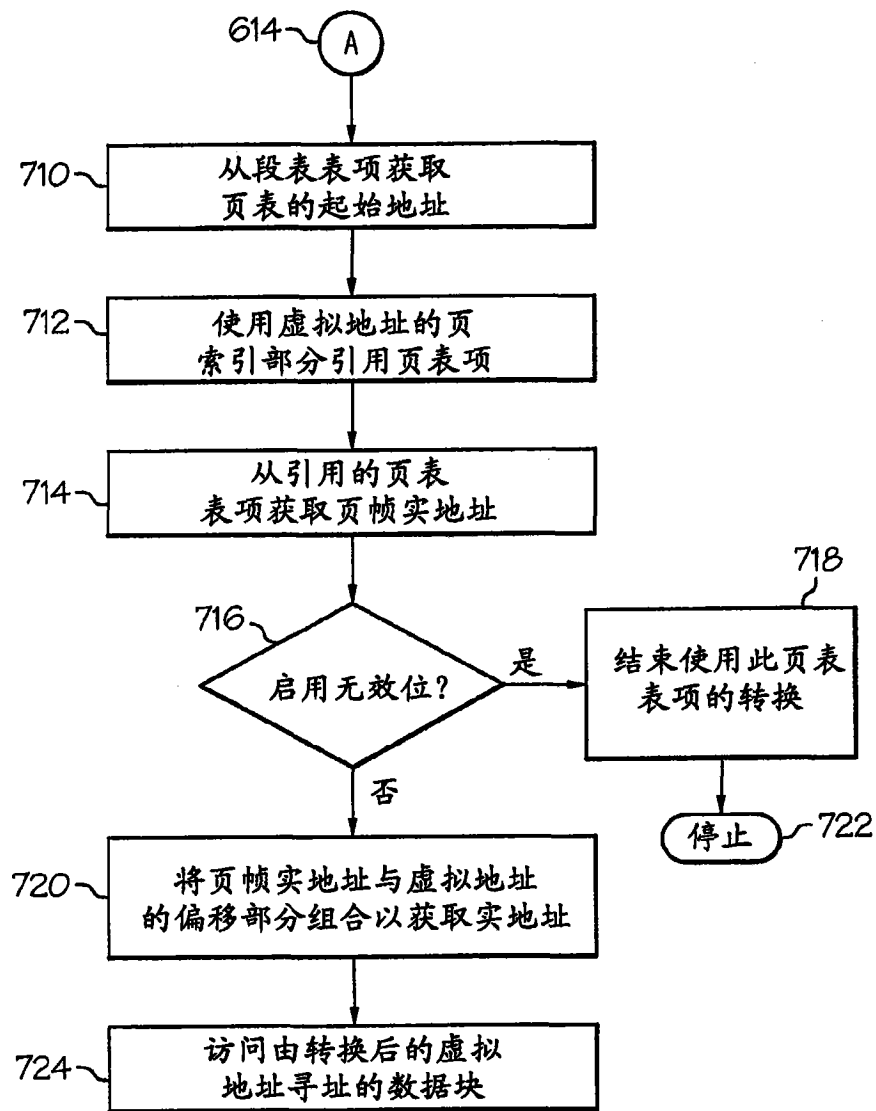


图 6



(现有技术)

图 7

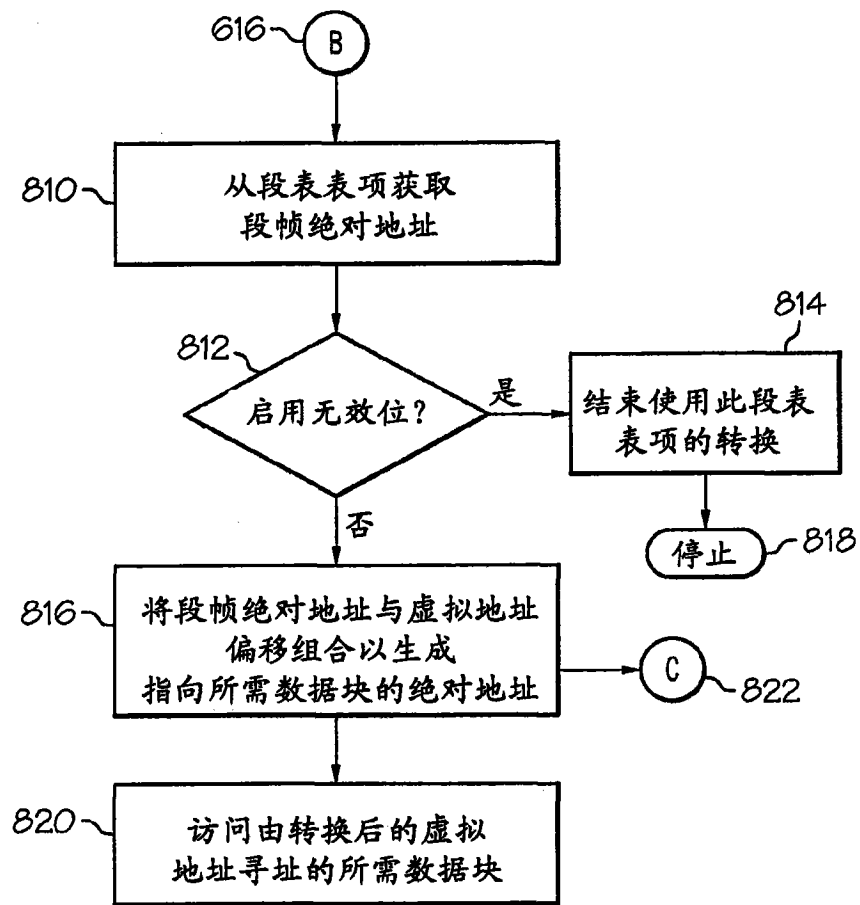


图 8

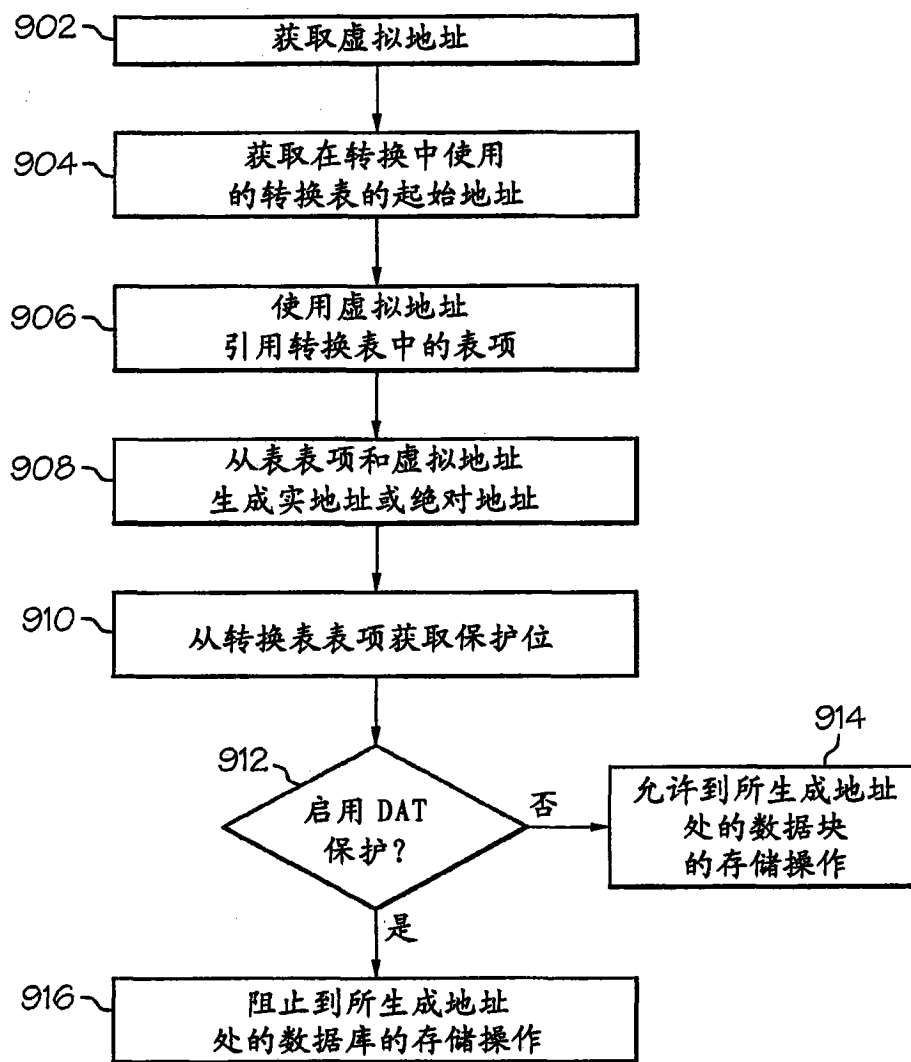


图 9

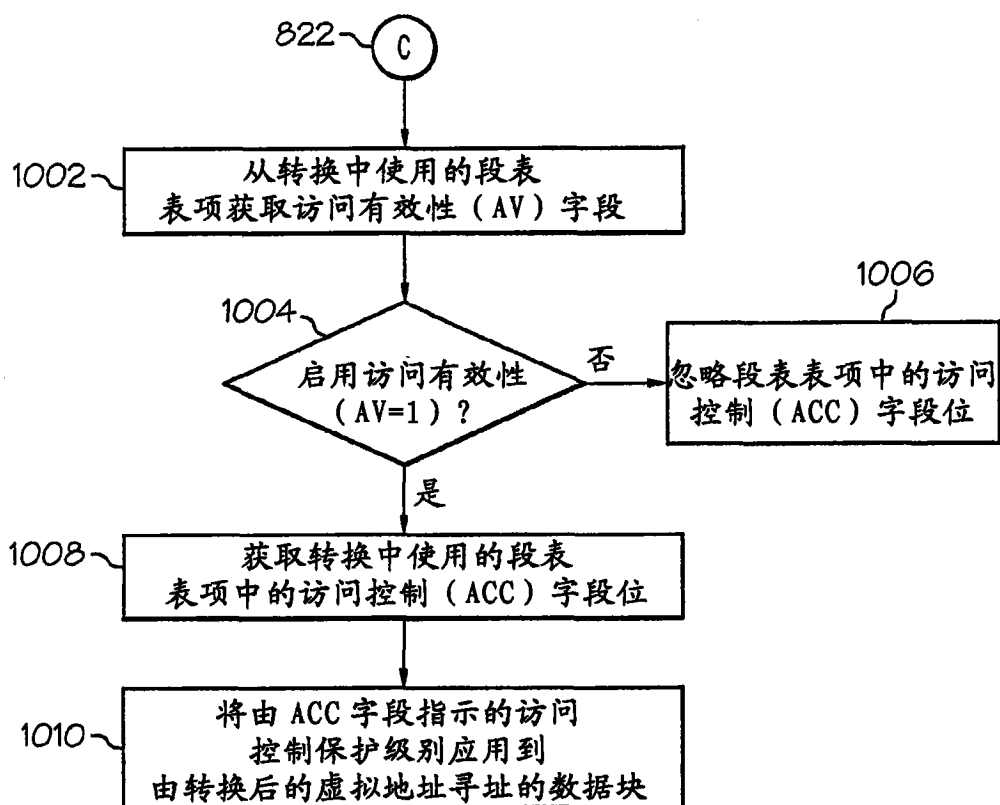


图 10

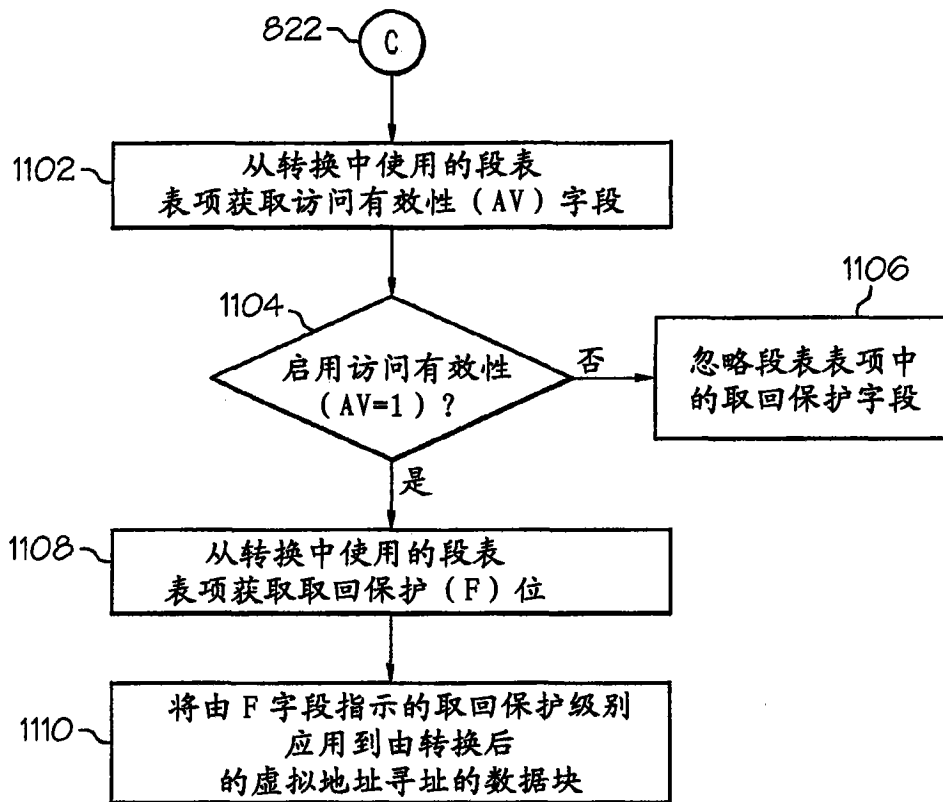


图 11

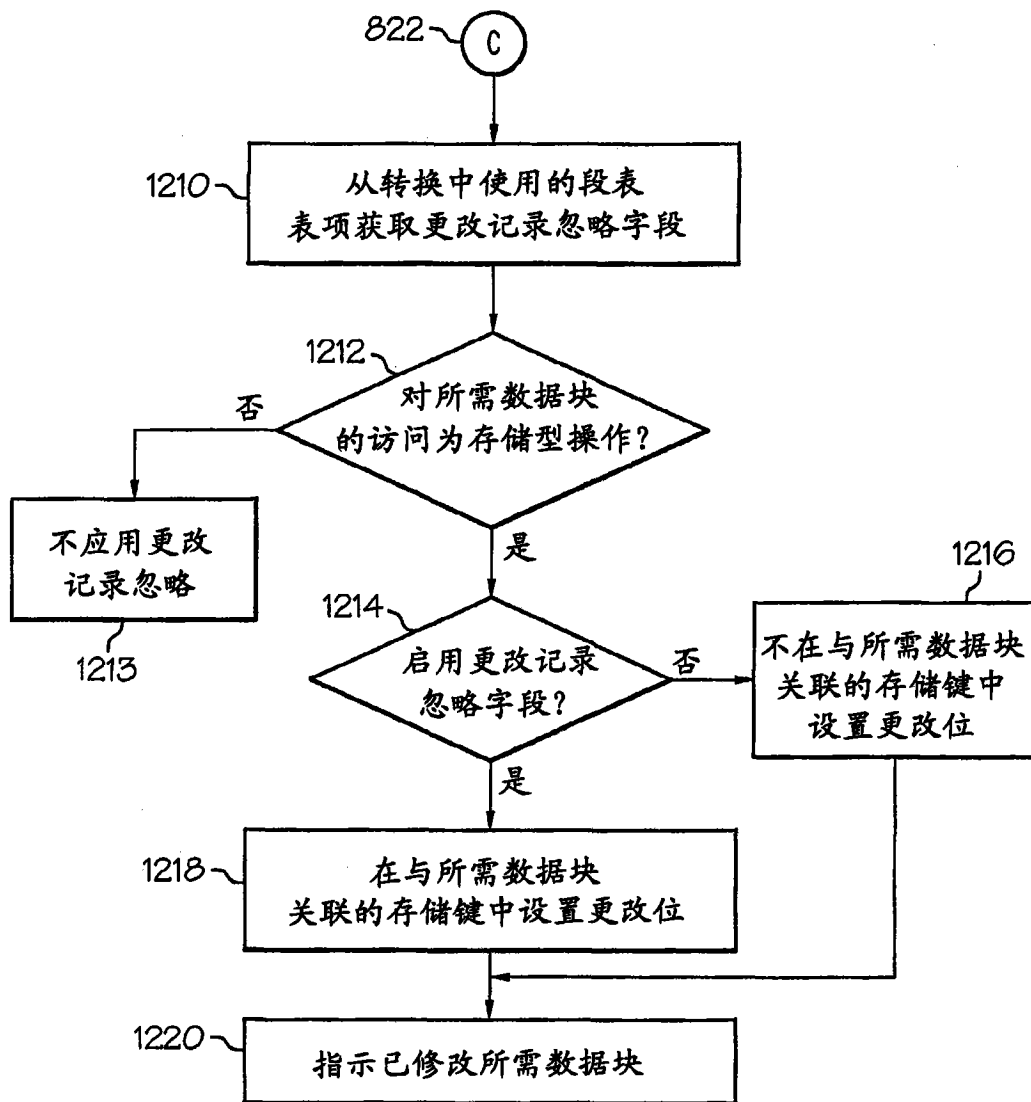


图 12

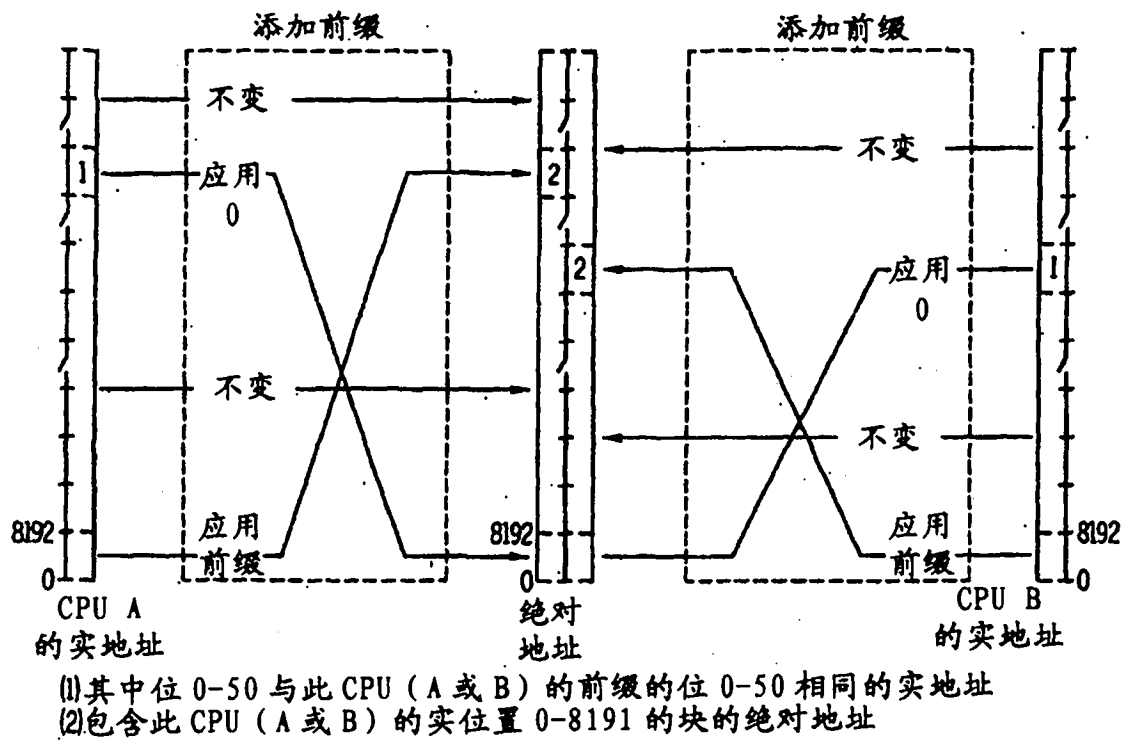


图 13