



(12) 发明专利

(10) 授权公告号 CN 101911022 B

(45) 授权公告日 2012. 10. 10

(21) 申请号 200980102001. 4

(51) Int. Cl.

(22) 申请日 2009. 01. 05

G06F 12/10 (2006. 01)

(30) 优先权数据

11/972, 706 2008. 01. 11 US

(56) 对比文件

US 5008811 , 1991. 04. 16,

US 4992936 , 1991. 02. 12,

US 5617554 A, 1997. 04. 01,

(85) PCT申请进入国家阶段日

2010. 07. 09

审查员 武晓冬

(86) PCT申请的申请数据

PCT/EP2009/050049 2009. 01. 05

(87) PCT申请的公布数据

W02009/087132 EN 2009. 07. 16

(73) 专利权人 国际商业机器公司

地址 美国纽约

(72) 发明人 D·格雷纳尔 L·海勒

D·奥西塞克 T·斯莱格尔

E·费菲尔 C·威伯

(74) 专利代理机构 中国国际贸易促进委员会专

利商标事务所 11038

代理人 李镇江

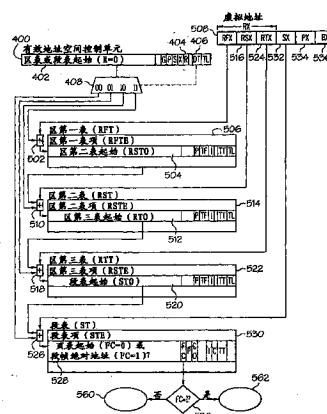
权利要求书 3 页 说明书 34 页 附图 14 页

(54) 发明名称

具有格式控制的动态地址转换

(57) 摘要

本发明涉及具有格式控制的动态地址转换。提供了一种增强型动态地址转换设备。在一个实施例中, 获取要进行转换的虚拟地址和转换表层级结构转换表的最初的起始地址。所述虚拟地址的索引部分被用于引用转换表中的项。如果格式控制字段被启用, 则从转换表项获取主存储中的大的数据块的帧地址。所述大的数据块是大小至少为 1M 字节的块。然后, 所述帧地址与所述虚拟地址的偏移部分结合以形成主存储中的大的数据块内的期望数据块的转换地址。然后, 存取由所述转换地址寻址的期望的大的数据块。



1. 一种用于将计算机系统的主存储中的数据块的虚拟地址转换成转换地址的方法,所述计算机系统具有用于转换所述虚拟地址的转换表层级结构机器架构,所述方法包括:

获取要进行转换的虚拟地址;

获取所述转换表层级结构的一转换表的起始地址;

使用所述虚拟地址的索引部分来引用所述转换表中的项;

确定所述转换表项中的格式控制字段是否被启用;和

响应所述格式控制字段被启用,执行:

从所述转换表项获取主存储中的大的数据块的帧地址;

将所述帧地址与所述虚拟地址的偏移部分结合以形成所述主存储中的所述大的数据块中的期望的数据块的所述转换的绝对地址,其中所述期望的数据块包括数据的一个或多个字节;和

存取由所述转换地址寻址的所期望的大的数据块;

从所述转换表项获取用于限制对所述大的数据块的限制部分的存取的限制信息;和

将对所述转换地址寻址的所述期望的大的数据块的存取限制为仅存取所述大的数据块的所述限制部分;

其中响应所述格式控制字段不被启用,执行:

从所述转换表项获取所述转换表层级结构的另一个表的起始地址;

使用所述虚拟地址的索引部分来引用所述另一个表中的项,从所述另一个表项获取主存储中的小的数据块的小的帧地址,所述小的数据块比所述大的数据块小;

将所述小的帧的实地址与所述虚拟地址的偏移部分结合以形成主存储中的所期望的小数据块的转换的实地址,所述转换的实地址受到加前缀操作以形成绝对地址;和

存取由所述转换地址寻址的主存储中的所述期望的小数据块。

2. 根据权利要求1的方法,其中所述限制信息包括表偏移或表长度中的任何一个。

3. 根据权利要求1的方法,其中,转换表层级结构包括以下中的一个或多个:具有用于定位区第二表的区第一表项的区第一表、具有用于定位区第三表的区第二表项的区第二表、具有用于定位段表的区第三表项的区第三表和具有段表项的段表,所述方法还包括:

以所述转换表层级结构预定的转换表开始,按顺序检索转换表层级结构转换表项中的一个或多个,直至具有被启用格式控制字段的转换表项被找到,其中具有转换表项的转换表的转换表起始地址是所述转换表层级结构的一转换表的获得的起始地址。

4. 根据权利要求1的方法,还包括:

将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;和

使用来自所述转换后备缓冲器的存储的信息而不是所述转换表层级结构来执行主存储中的所述数据块的随后的虚拟地址到绝对地址的随后的转换。

5. 根据权利要求1的方法,还包括,响应于所述虚拟地址转换不是原生于所述机器架构,则:

识别用于仿真所述虚拟地址转换的预定的软件例程,所述预定的软件例程包括多个指令;和

执行所述预定的软件例程。

6. 根据权利要求1的方法,其中所述大的数据块包括大小至少为1兆字节的块。

7. 一种用于将计算机系统的主存储中的数据块的虚拟地址转换成转换地址的装置,所述计算机系统具有用于转换所述虚拟地址的转换表层级结构机器架构,所述装置包括:

用于获取要进行转换的虚拟地址的部件;

用于获取所述转换表层级结构的一转换表的起始地址的部件;

用于使用所述虚拟地址的索引部分来引用所述转换表中的项的部件;

用于确定所述转换表项中的格式控制字段是否被启用的部件;和

用于响应所述格式控制字段被启用的部件包括:

用于从所述转换表项获取主存储中的大的数据块的帧地址的部件;

用于将所述帧地址与所述虚拟地址的偏移部分结合以形成所述主存储中的所述大的数据块中的期望的数据块的所述转换的绝对地址的部件,其中所述期望的数据块包括数据的一个或多个字节;和

用于存取由所述转换地址寻址的所期望的大的数据块的部件;

用于从所述转换表项获取用于限制对所述大的数据块的限制部分的存取的限制信息的部件;和

用于将对所述转换地址寻址的所述期望的大的数据块的存取限制为仅存取所述大的数据块的所述限制部分的部件;

用于响应所述格式控制字段不被启用的部件包括:

用于从所述转换表项获取所述转换表层级结构的另一个表的起始地址的部件;

用于使用所述虚拟地址的索引部分来引用所述另一个表中的项,从所述另一个表项获取主存储中的小的数据块的小的帧地址的部件,所述小的数据块比所述大的数据块小;

用于将所述小的帧的实地址与所述虚拟地址的偏移部分结合以形成主存储中的所期望的小数据块的转换的实地址的部件,

所述转换的实地址受到加前缀操作以形成绝对地址;和

用于存取由所述转换地址寻址的主存储中的所述期望的小数据块的部件。

8. 根据权利要求7的装置,其中所述限制信息包括表偏移或表长度中的任何一个。

9. 根据权利要求7的装置,其中,转换表层级结构包括以下中的一个或多个:具有用于定位区第二表的区第一表项的区第一表、具有用于定位区第三表的区第二表项的区第二表、具有用于定位段表的区第三表项的区第三表和具有段表项的段表,所述装置还包括:

用于以所述转换表层级结构预定的转换表开始,按顺序检索转换表层级结构转换表项中的一个或多个,直至具有被启用格式控制字段的转换表项被找到的部件,其中具有转换表项的转换表的转换表起始地址是所述转换表层级结构的一转换表的获得的起始地址。

10. 根据权利要求7的装置,还包括:

用于将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部件;和

用于使用来自所述转换后备缓冲器的存储的信息而不是所述转换表层级结构来执行主存储中的所述数据块的随后的虚拟地址到绝对地址的随后的转换的部件。

11. 根据权利要求7的装置,还包括,用于响应于所述虚拟地址转换不是原生于所述机器架构,则:

用于识别用于仿真所述虚拟地址转换的预定的软件例程的部件,所述预定的软件例程

包括多个指令 ;和

用于执行所述预定的软件例程的部件。

12. 根据权利要求 7 的装置,其中所述大的数据块包括大小至少 为 1 兆字节的块。

具有格式控制的动态地址转换

技术领域

[0001] 本发明一般涉及用于转换计算机系统内的虚拟地址的系统和方法,更具体地涉及将计算机系统内的数据块的虚拟地址转换为实地址或绝对地址的系统和方法,该计算机系统具有其中通过转换表层级结构发生虚拟地址转换的动态地址转换设备。

背景技术

[0002] 动态地址转换提供了如下的能力:在任意时刻中断程序执行、在诸如直接存取存储设备的辅助存储中记录它及其数据、并在随后的时间将程序和数据返回到不同的主存储位置以便重新执行。将程序及其数据在主存储和辅助存储之间转换可以零碎 (piecemeal) 进行,将信息返回到主存储可响应于 CPU (中央处理单元) 需要执行它时对其试图存取而发生。这些功能可以不改变或不检验程序及其数据而执行,不需要在重定位的程序中有任何明确的程序设计约定,并且除涉及时间延迟外不干扰程序执行。

[0003] 利用操作系统的适当支持,动态地址转换设备可被用来向用户提供一种系统,其中存储看上去比在配置中可用的主存储大。这种明显的主存储通常被称为虚拟存储,用来标识虚拟存储中的位置的地址通常被称为虚拟地址。用户的虚拟存储可以远远超过主存储的大小,该主存储在配置中可用且通常是维持在辅助存储中。虚拟存储被认为是由通常称作页 (也被称为段和区) 的数据块组成。只有最近被称作虚拟存储的页被分配占据物理主存储的块。当用户引用未在主存储中出现的虚拟存储的页时,它们被带入以替换主存储中较小可能需要的页 (pace)。在某些情况下,虚拟存储被分配给主存储达一长段时间 (或永久性地),而不论所述存储是否被引用。存储的调换 (swapping) 页可以在用户不知晓的情况下由操作系统执行。

[0004] 程序使用地址 (或虚拟地址) 来存取虚拟存储。所述程序可使用虚拟地址从虚拟存储提取 (fetch) 指令或加载数据或存储来自虚拟存储的数据。与虚拟存储范围相关联的虚拟地址定义了一个地址空间。在操作系统的适当支持下,动态地址转换设备可用于提供多个地址空间。这些地址空间可以被用来提供用户之间的隔离度。这种支持可以包括对每个用户完全不同的地址空间,从而提供了完全的隔离,或者可通过将每个地址空间的一部分映射到单一的公共存储区来提供共享区。还有,提供了允许半特许程序存取多于一个这种地址空间的指令。

[0005] 动态地址转换提供了来自多个不同的地址空间的虚拟地址的转换。这些地址空间被称作主地址空间、二次地址空间和存取寄存器指定的地址空间。特许程序也可导致初始地址空间 (home address space) 被存取。动态地址转换可以被指定由 CPU 生成的指令和数据地址。

[0006] 所需要的是提供现有技术迄今未知的额外的功能、能力和保护的增强型动态地址转换设备。

发明内容

[0007] 提供了用于将计算机系统的主存储中的期望数据块的虚拟地址转换成实地址或绝对地址的增强型动态地址转换设备的系统、方法和计算机程序产品，该计算机系统具有带有用于转换的转换表层级结构的机器架构。实地址可以受到加前缀而形成绝对地址。

[0008] 在一个实施例中，首先获取要进行转换的虚拟地址和获取转换表层级结构转换表的最初的起始地址。所述虚拟地址的索引部分被用于引用转换表中的项。确定在转换表项中的格式控制字段 (field) 是否被启用。如果格式控制字段被启用，则从转换表项获取主存储中的大的数据块的帧地址。所述大的数据块包括大小至少为 1M 字节的块。然后，所述帧地址与所述虚拟地址的偏移部分结合以形成主存储中的大的数据块内的期望数据块的转换地址。然后，存取由所述转换地址寻址的期望的大的数据块。在另一实施例中，从转换表项获取无效的表项 (entry) 字段。如果相应的无效表项字段被启用，则异常动作开始。

[0009] 在另一实施例中，从转换表项获取限制信息。所述限制信息用于限制对所述大的块的限制部分的存取。对由转换地址寻址的期望的大的数据块的存取随后被允许仅对数据块的受限制部分进行存取。

[0010] 下面将结合某些例示的实施例描述本发明。应该理解，本领域的技术人员可进行各种变更和修改，而不背离本发明的精神或范围。

附图说明

[0011] 在附图中，各视图中的相似的附图标记是指功能相同或类似的部件，各视图连同以下的详细说明被并入和形成本说明书的一部分，用于进一步例示各实施例并解释根据本发明的各种原理和全部优势。

[0012] 图 1 例示了其中增强型动态地址被执行转换的主机计算机系统的实施例；

[0013] 图 2 提供了一个仿真主机架构的主机计算机系统的仿真主机计算机系统的例子；

[0014] 图 3 例示了程序状态字是如何被用来为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 例示了其中将在图 3 中确定的有效 ASCE 用于确定在虚拟地址转换中使用的转换表层级结构中的最高转换表的一个实施例；

[0016] 图 5A 例示了使用转换表层级结构到段表级进行虚拟地址的动态地址转换的过程的一个实施例；

[0017] 图 5B 例示了其中的段表项 (STE) 格式控制 (FC) 为 0 的图 5A 的动态地址转换的继续；

[0018] 图 5C 例示了其中的段表项 (STE) 格式控制 (FC) 为 1 的图 5A 的动态地址转换的继续；

[0019] 图 6 例示了增强型动态地址转换 (eDAT) 以获取段表项中的格式控制字段的一个实施例的流程图；

[0020] 图 7 例示了从图 6 的节点 614 的流程图的继续；

[0021] 图 8 例示了从图 6 的节点 616 的流程图的继续；

[0022] 图 9 例示了确定要被应用到由转换的虚拟地址寻址的期望的数据块的 DAT 保护级的实施例的流程图；

[0023] 图 10 例示了从图 8 的节点 822 的流程图的继续，其确定要被应用到由转换的虚拟

地址寻址的期望的数据块的存取控制保护级别；

[0024] 图 11 例示了从图 8 的节点 822 的流程图的继续,其确定要被应用到由转换的虚拟地址寻址的期望的数据块的提取保护级；

[0025] 图 12 例示了从图 8 的节点 822 的流程图的继续,其中从一个段表项获取改变记录重写字段 (change recording override field) ;和

[0026] 图 13 例示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,在本申请的说明书中进行的声明不会限制各项权利要求的发明的任何一项。此外,某些声明可以适用于一些发明特征但不适用于另一些。除非另有说明,单个元件可以是多个形式,反之亦然,而不会丧失一般性。

[0028] 本领域的普通技术人员容易熟悉在计算环境中寻址存储和使用寄存器或地址字段中的位来指示不同的状态和对这些状态采取行动。此外,本领域的一般技术人员将有计算机程序领域的知识,并有关于计算机系统部件之间的运作及其相互关系的知识。

[0029] 概述

[0030] 所提供的是增强型动态地址转换 (DAT) 设备的实施例的例子。当增强型 DAT 设备已安装并启用时,DAT 转换可能会产生页帧的实地址或段帧的绝对地址,这由段表项中的段表项 (STE) 格式控制确定。这里使用的术语“增强型 DAT 适用”是指所有的下述条件为真: 1) 安装了 EDAT 设备 ;2) 通过控制寄存器 0 (CR0) 位 40 启用 EDAT 设备 ;和 3) 通过 DAT 表项转换地址。

[0031] 当增强型 DAT 应用时,下述额外的功能可以在 DAT 过程中得到:

[0032] DAT 保护位被添加到区表项,提供类似于段和页表项中的 DAT 保护位的功能。

[0033] STE 格式控制被添加到段表项。当 STE 格式控制为 0 时,除非页表项中的改变记录重写指示是否可为该页绕过改变位的设置,否则 DAT 如当前所定义地继续。

[0034] 当 STE 格式控制为 1 时,该段表项还包括以下内容:

[0035] 段帧绝对地址 (而不是页表起始),指定 1 兆字节块的绝对存储位置。

[0036] 存取控制位和提取保护位,可以选择性地替换段的各存储密钥中的相应位来使用。

[0037] 确定段表项中的存取控制位和提取保护位的有效性的位。

[0038] 改变记录重写,指示是否在段的各存储密钥中可绕过改变位。

[0039] 主机计算机系统

[0040] 参考图 1,绘出了主机计算机系统 100 的代表性组件。组件的其它布置也可以实施在计算机系统中,这是本领域所公知的。

[0041] 主机计算机环境优选基于纽约州阿蒙克 (Armonk) 的国际商业机器公司 (IBM®) 所提供的 **z/Architecture®**。z/Architecture® 在 (2007 年 4 月) IBM® 出版物编号 SA22-7832-05 第六版的《**z/Architecture®** Principles of Operation》中有更完全的描述,这里通过引用将其整体并入。基于 **z/Architecture®** 的计算环境包括例如 eServer 和 **zSeries®**,二者皆由 IBM® 提供。

[0042] 代表性的主机计算机 100 包括与主存储（计算机存储器 102）和 I/O 接口通信的一个或多个 CPU 101, 该 I/O 接口到存储设备 111 和网络 110, 用于与其它的计算机或存储区网络（SAN）等通信。CPU 可具有动态地址转换（DAT）设备（功能或单元）103 用于将程序地址（虚拟地址）转换到存储器的实地址。DAT 设备通常包括用于缓存转换的转换后备缓冲器（translation lookaside buffer）107, 以便对计算机存储器 102 的块的较后的存取不需要地址转换迟滞。通常是在计算机存储器 102 和处理器 101 之间采用缓存 109。缓存 109 可以是层级结构, 具有可为多于一个 CPU 所用的大的缓存以及在大缓存和每一个 CPU 之间的更小、更快的（低级）缓存。在一些实现（实施方式）中, 较低级的缓存被分割（split）以为指令提取和数据访问提供单独的低级缓存。在一个实施例中, 由指令提取单元 104 通过缓存 109 从存储器 102 提取指令。该指令在指令解码单元（106）中解码, 并被分派（在一些实施例中, 和其它指令一起）到指令执行单元 108。通常采用几个执行单元 108, 例如算术执行单元、浮点执行单元和分支指令执行单元。指令由执行单元执行, 根据需要从指令指定的寄存器或存储器存取操作数。如果从存储器（内存）（memory）102 存取（加载或存储）操作数, 则负载存储单元 105 通常在正执行的指令的控制下处理存取。

[0043] 在一个实施例中, 本发明可由软件（有时称为授权内码（LIC）、固件、微码（micro-code）、毫码（milli-code）、微微码（pico-code）等, 其中任何一个与本发明一致）实现。实施本发明的软件程序代码通常是由计算机系统 100 的处理器（也称作 CPU（中央处理单元））101 从诸如 CD-ROM 驱动器、磁带驱动器或硬盘驱动器的长期储存介质 111 存取。软件程序代码可以在与数据处理系统一起使用的多种已知介质中的任一个上实施, 例如软盘、硬盘驱动器或 CD-ROM。可以在这种介质上分发代码, 或者可以分发给用户, 通过网络 110 从计算机存储器 102 或计算机系统的存储分发到其它计算机系统, 以便由这种其它系统的用户使用。

[0044] 可替换地, 程序代码可以在存储器 102 上实施, 并由处理器 101 使用处理器总线存取。这种程序代码包括控制各计算机部件和一个或多个应用程序的功能和相互作用（交互）的操作系统。程序代码通常是从高密度存储介质 111 分页到高速存储器 102, 在那里可用于由处理器 101 处理。在存储器中、在物理介质上实施软件程序代码和 / 或通过网络分发软件代码的技术和方法是众所周知的, 在这里不做进一步的讨论。当在有形介质（包括但不限于电子存储器模块（RAM）、快闪存储器、光盘（CD）、DVD、磁带等）上创建和存储程序代码时, 它往往被称为“计算机程序产品”。计算机程序产品介质通常是由优选在计算机系统处理电路可读, 用于由处理电路执行。

[0045] 在图 2 中, 提供了仿真主机架构的主机计算机系统 100 的仿真的主机计算机系统 201 的一个例子。在仿真的主机计算机系统 201 中, 主处理器（CPU）208 是仿真的主处理器（或虚拟主机处理器）, 包括具有与主机计算机 100 的处理器 101 所用的指令集架构不同的原生指令集架构的仿真处理器 207。仿真的主机计算机系统 201 具有可由仿真处理器 207 存取的存储器 202。在这个示例的实施例中, 存储器 207 分区（partition）为主机计算机存储器 102 部分和仿真例程 203 部分（提供仿真的例程可以是主机存储器的一部分）。根据主机计算机架构, 主机计算机存储器 102 对仿真主机计算机 201 的程序可用。仿真处理器 207 执行架构的架构指令集的原生指令而不是仿真处理器 208 的原生指令、从仿真例程存储器 203 获取的原生指令, 并可以通过采用序列与存取 / 解码例程中获取的一个或多个

指令,存取用于执行来自主机计算机存储器 102 中的程序的主机指令,该例程解码所存取的主机指令以确定用于仿真存取的主机指令的功能的原生指令执行例程。

[0046] 为主机计算机系统 100 架构定义的其它设备(设施)可以用架构的设备(设施)例程来仿真,例如,包括诸如通用目的寄存器、控制寄存器、动态地址转换和 I/O 子系统支持和处理器缓存等设备。仿真例程还可以利用在仿真处理器 207 中可用的功能(诸如通用寄存器和虚拟地址的动态转换)来改善仿真例程的性能。也可以提供特殊的硬件和减负载引擎(Off Load Engine)来协助处理器 207 仿真主机计算机 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通信。通信总线可以是在 CPU 的内部或外部。一些寄存器可以是只读的。其它硬件和/或软件还可以读/写可以由 CPU 存取(访问)的一个或多个寄存器。指令操作代码(或运算码)(opcode)确定将在任何特定的机器指令操作中使用的寄存器的类型。

[0049] 通用寄存器

[0050] 指令可以标识(指定)(designate)一个或多个 16 通用寄存器中的信息。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并作为通用算术和逻辑操作中的累加器。每个寄存器包括 64 位位置。通用寄存器由数字 0-15 识别(标识)(identify),并由指令中的 4 位 R 字段标识。一些指令通过有几个 R 字段来提供寻址(address)多个通用寄存器。对于一些指令,使用一个具体的通用寄存器是由指令的 R 字段暗示而不是明确标识的。

[0051] 对于某些操作,两个相邻的通用寄存器的位 32-63 或位 0-63 耦合,分别提供了 64-位或 128 位的格式。在这些操作中,程序必须标识包括最左边(高序位)32 或 64 位的偶数编号的寄存器。下一个更高编号的寄存器包括了最右边(低序位)的 32 或 64 位。除了它们在通用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 个还被用作地址生成中的基址寄存器和索引寄存器。在这种情况下,寄存器被指令中的 4 位 B 字段或 X 字段标识。B 或 X 字段中的 0 值指定不施加基址或索引,因此,通用寄存器 0 不能被标识为包括基址或索引。

[0052] 控制寄存器

[0053] 控制寄存器提供在程序状态字之外维护和操作控制信息。CPU 有 16 个控制寄存器,每一个控制寄存器有 64 位位置。寄存器中的位位置被分配给系统中的诸如程序事件记录的特定设备,并被用于指定操作能够进行或供给设备所需的特别信息。控制寄存器由数字 0-15 识别,由指令 LOAD CONTROL 和 STORE CONTROL 中的四位 R 字段来标识。多个控制寄存器能够由这些指令来寻址。

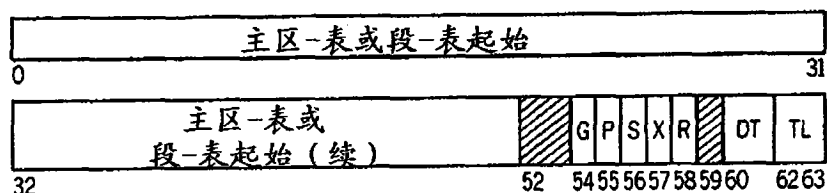
[0054] 控制寄存器 1

[0055] 控制寄存器 1 包括主地址空间控制单元(PASCE)。在一个实施例中,根据寄存器中的实空间控制位(R),控制寄存器 1 具有如下的两种格式之一:

[0056] 主区-表或

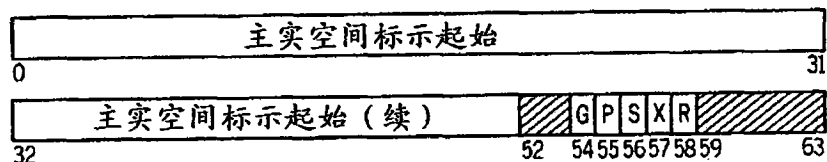
[0057] 段-表标识(R=0)

[0058]



[0059] 主实空间标识 ($R = 1$)

[0060]



[0061] 控制寄存器 1 的格式

[0062] 在主地址空间控制单元 (PASCE) 中选择的字段分配如下：

[0063] 主区表或段表起始：控制寄存器 1 中的主区表或段表标识的位 0-51 在右侧加 12 个 0 形成标识主区表或段表的开始的 64 位地址。不可预知，地址是否为实地址或绝对地址。因为它用来转换主地址空间中的虚拟地址，所以此表被称为主区表或段表。

[0064] 主实空间控制 (R)：如果控制寄存器 1 的位 58 为 0，则寄存器包括区表或段表标识。如果位 58 为 1，则寄存器包括实空间标识。当位 58 是 1 时，转换后备缓冲器中代表段表项的公共段位的 1 值阻止所述项和它标识的转换后备缓冲器的页表副本在将引用转换到主地址空间时被使用，即使控制寄存器 1 中的标识起始与转换后备缓冲器项的表起始之间匹配。

[0065] 主指定类型控制 (DT)：当 R 是 0 时，控制寄存器 1 中的表标识的类型由寄存器中的位 60 和 61 指定如下：

[0066] 位 60

[0067] 和 61 标识类型

[0068] 11 区第一表

[0069] 10 区第二表

[0070] 01 区第三表

[0071] 00 段表

[0072] 主标识类型 (DT) 控制位

[0073] 当 R 为 0 时，当试图使用 PASCE 转换最左边的一位是在地址的位位置 0-10 中的虚拟地址时，位 60 和 61 必须是二进制 11。类似地，当最左边的一位是在地址的位位置 11-21 中时，位 60 和 61 必须是二进制 11 或 10，当最左边的一位是在地址的位位置 22-32 中时，它们必须是二进制 11、10 或 01。否则，认定 (recognize) ASCE 类型异常 (exception)。

[0074] 主区表或段表长度 (TL)：控制寄存器 1 中主区表标识或段表标识的位 62 或 63 以 4,096 字节单元指定主区表或段表的长度，从而使主区表或段表的长度是可以以 512 的倍数项而可变。以 4,096 字节单元的主区表或段表的长度比 TL 值多 1。长度字段的内容被用来证实 (establish) 要通过所述表转换的虚拟地址 (RFX、RSX、RTX 或 SX) 是否标识落在所述表中的项。

[0075] 主实空间标示起始：控制寄存器 1 中的主实空间标识的位 0-51 在右边加 12 个 0 形成一个 64 位地址，可用于形成和使用提供虚拟的等同实转换以引用主地址空间的转换后备缓冲器项。虽然这个地址只是用作一个标示而不用用于执行存储引用，但它仍须是有效地址；否则，当控制寄存器 1 的内容被使用时，不正确的转换后备缓冲器项可能会被使用。

[0076] 控制寄存器 1 的以下位未被分配并被忽略：位 52、53 和 59，如果寄存器包括区表标识或段表标识；和位 52、53 和 59-63，如果寄存器包括实空间标识。

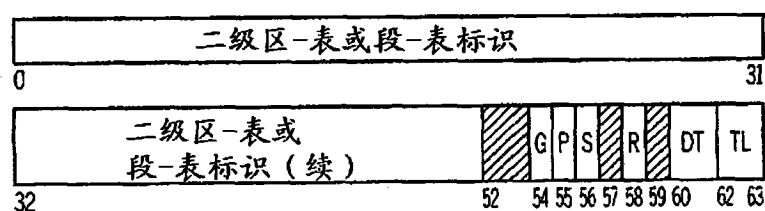
[0077] 控制寄存器 7

[0078] 控制寄存器 7 包括二级地址空间控制单元 (SASCE)。在一个实施例中，依据寄存器中的实空间控制位 (R)，控制寄存器 7 有以下两种格式之一：

[0079] 二级区 - 表或

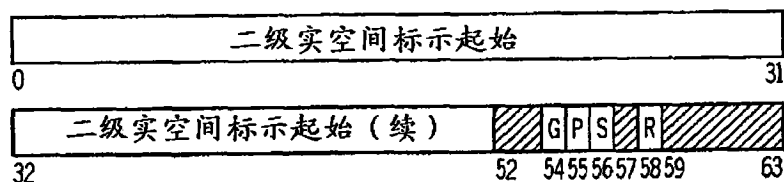
[0080] 段 - 表标识 (R = 0)

[0081]



[0082] 二级实空间标识 (R = 1)

[0083]



[0084] 控制寄存器 7 的格式

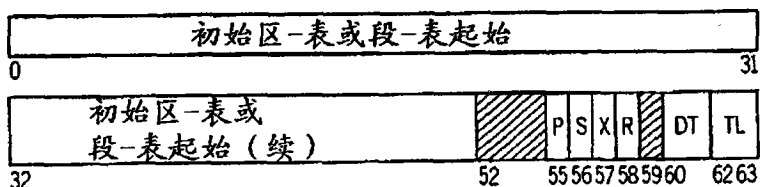
[0085] 控制寄存器 13

[0086] 控制寄存器 13 包括初始 (Home) 地址空间控制单元 (HASCE)。在一个实施例中，依据寄存器中的实空间控制位 (R)，控制寄存器 13 有以下两种格式之一：

[0087] 初始区 - 表或

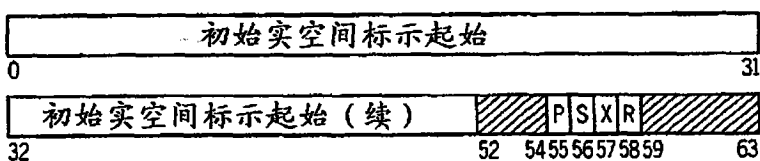
[0088] 段 - 表标识 (R = 0)

[0089]



[0090] 初始实空间标识 (R = 1)

[0091]



[0092] 控制寄存器 13 的格式

[0093] 存取寄存器

[0094] CPU 有 16 个编号为 0-15 的存取寄存器。存取寄存器包括含有 ASCE 的间接规范 (specification) 的 32 位位置。ASCE 是动态地址转换 (DAT) 机制用来将引用转换到相应的地址空间的一个参数。当 CPU 处于称作存取寄存器模式 (由程序状态字中的位控制) 的模式时, 用于为存储操作数引用指定逻辑地址的指令 B 字段标识存取寄存器, 并且存取寄存器指定的 ASCE 由 DAT 使用以进行引用。对于一些指令, 使用 R 字段而不是 B 字段。提供指令来加载和存储存取寄存器的内容, 并将一个存取寄存器的内容移到另一个存取寄存器。

[0095] 存取寄存器 1-15 中的每一个都能标识任何地址空间, 包括当前指令空间 (主地址空间)。存取寄存器 0 标识主指令空间。当寄存器 1-15 中的一个被用来标识一个地址空间时, CPU 确定哪个地址空间是通过转换存取寄存器的内容来标识。当存取寄存器 0 被用于标识地址空间时, CPU 将存取寄存器处理成标识主指令空间, 并且它不检查存取寄存器的实际内容。因此, 16 个存取寄存器能够在任何一个时间标识主指令空间和最多 15 个其它空间。

[0096] 程序状态字 (PSW)

[0097] 程序状态字包括指令地址、状态码和用来控制指令排序 (sequencing) 和确定 CPU 的状态的其它信息。活动或控制程序状态字被称作当前程序状态字。它管理正在执行的程序。

[0098] CPU 具有中断功能, 它允许 CPU 响应于异常条件和外部刺激而快速切换到另一个程序。当中断发生时, CPU 为该特定类别的中断将当前程序状态字放在称作旧程序状态字位置的分配的存储位置。CPU 从第二个分配的存储位置提取一个新的程序状态字。这个新的程序状态字确定要执行的下一个程序。当它已经完成处理中断时, 处理中断的程序可能会重新加载旧程序状态字, 使其再次成为当前程序状态字, 以便中断的程序能够继续。

[0099] 有 6 类中断: 外部、I/O、机器检查、程序、重新启动和监督 (supervisor) 调用。每个类有在实存储中永久分配的不同的一对旧程序状态字和新程序状态字位置。

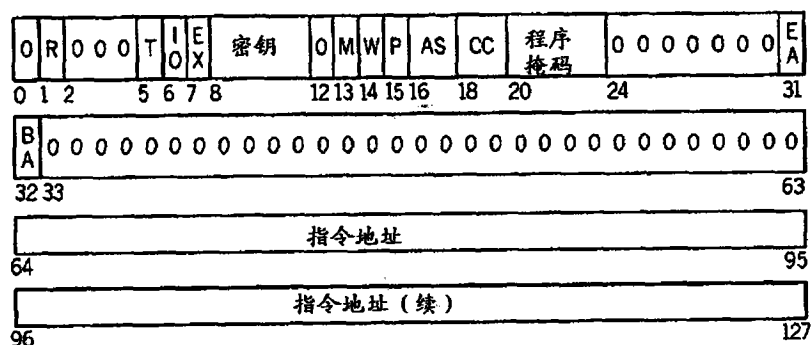
[0100] 当前程序状态字

[0101] CPU 中的当前程序状态字包括执行当前活动程序所需的信息。程序状态字是 128 位长, 包括指令地址、状态码和其它控制字段。一般地, 程序状态字是用来控制指令排序并保持和指示与当前正在执行的程序有关的 CPU 的很多状态。额外的控制和状态信息包括在控制寄存器和永久分配的存储位置中。CPU 的状态可以通过加载新的程序状态字或程序状态字的一部分而改变。

[0102] 通过存储当前的程序状态字以保存 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。LOAD PSW 或 LOAD PSWEXTENDED 的执行或成功结束最初程序加载顺序 (sequence) 引入新的程序状态字。指令地址通过顺序指令执行而更新, 为成功的分支所替代。提供了在一部分程序状态字上操作的其它指令。

[0103] 当中断或执行改变程序状态字的指令完成时, 新的或修改的程序状态字变为活动 (即, 引入当前程序状态字的信息对 CPU 承担控制)。对于与改变程序状态字的指令相关的程序事件记录 (PER) 的中断在 PER 掩码 (mask) 的控制下发生, PER 掩码在操作开始时有效。程序状态字的位 0-7 统称为系统掩码。在一个实施例中, 程序状态字具有如下的格式:

[0104]



[0105] 程序状态字的格式

[0106] 下面是选择的程序状态字字段的功能的简要概述。

[0107] DAT 模式 (T) :位 5 控制用于存取存储的隐含的逻辑和指令地址的动态地址转换是否发生。当位 5 是 0 时, DAT 是关闭的, 逻辑和指令地址被处理成实地址。当位 5 是 1 时, DAT 是开通的, 调用动态地址转换机制。

[0108] PSW 密钥 :位 8-11 形成 CPU 用于存储引用的存取密钥。如果引用受密钥控制的保护, 当存储信息或当信息从提取受保护的位置提取时, PSW 密钥与存储密钥匹配。但是, 对于 MOVE TO PRIMARY、MOVE TO SECONDARY、MOVE WITH KEY、MOVE WITHSOURCE KEY 和 MOVE WITH DESTINATION KEY 中的每一个的操作数中的一个, 使用指定为操作数的存取密钥而不是 PSW 密钥。

[0109] 地址空间控制 (AS) :位 16 和 17 与程序状态字位 5 结合控制转换模式。

[0110] 状态码 (CC) :位 18 和 19 是两位状态码。状况码依据执行某些指令获取的结果而设置为 0、1、2 或 3。绝大多数算术和逻辑操作以及一些其它操作设置状态码。指令“BRANCH ON CONDITION”能够指定状态码值的任何选择作为分支标准。

[0111] 指令地址 :程序状态字的位 64-127 是指令地址。这个地址标识要执行的下一个指令最左边字节的位置, 除非 CPU 是在等待状态 (程序状态字的位 14 是 1)。

[0112] 地址类型和格式

[0113] 为了寻址主存储的目的, 认定三种基本类型的地址 :绝对地址、实地址和虚拟地址。地址是基于在存储存取期间施加到地址上的转换而区分的。地址转换将虚拟地址转换成实地址。加前缀将实地址转换为绝对地址。除了三种基本类型的地址之外, 还定义了另外的类型, 依据指令和当前模式, 所述另外的类型被处理成三种基本类型的一种或另一种。

[0114] 绝对地址

[0115] 绝对地址是分配给主存储位置的地址。绝对地址是用于存储存取而不必在其上进行任何转变。配置中的信道子系统和所有的 CPU 通过使用相同的绝对地址来引用共享的主存储位置。可用主存储通常是从 0 开始的分配的连续的绝对地址, 地址被以整个 4K 字节块分配在整数边界上。当试图使用还没有被分配物理位置的块中的绝对地址时, 被认定异常。在某些模型上, 可提供允许操作者改变绝对地址和物理位置之间的对应的存储重构控制。但是, 在任何一个时间, 物理位置并不与多于一个的绝对地址相关联。根据其绝对地址排序的字节位置构成的存储被称为绝对存储。

[0116] 实地址

[0117] 实地址识别实存储中的位置。当使用实地址存取主存储时, 它通过加前缀

(prefix) 而转换为形成绝对地址。在任何时间,对配置中的每一个 CPU,有从实地址到绝对地址的映射。当 CPU 使用实地址来存取主存储时,该实地址可能会被加前缀而转换为绝对地址。特别的转换由 CPU 的前缀寄存器中的值定义。由根据其实地址排序的字节位置构成的存储被称为实存储。

[0118] 虚拟地址

[0119] 虚拟地址识别虚拟存储中的位置。当使用虚拟地址来存取主存储时,该虚拟地址通过动态地址转换被转换为或者受到加前缀而形成绝对地址的实地址或者直接转换为绝对地址。

[0120] 主虚拟地址

[0121] 主虚拟地址是通过主地址空间控制单元 (PASCE) 转换的虚拟地址。当处于主空间模式时,逻辑地址被处理成主虚拟地址。当处于主空间模式、二级空间模式或存取寄存器模式时,指令地址被处理成主虚拟地址。MOVE TO PRIMARY 的第一个操作数地址和 MOVE TOSECONDARY 的第二个操作数地址被处理成主虚拟地址。

[0122] 二级虚拟地址

[0123] 二级虚拟地址是通过二级地址空间控制单元 (SASCE) 转换的虚拟地址。当处于二级空间模式时,逻辑地址被处理成二级虚拟地址。MOVE TO PRIMARY 的第二个操作数地址和 MOVE TOSECONDARY 的第一个操作数地址被处理成二级虚拟地址。

[0124] AR 标识的虚拟地址

[0125] AR 标识的虚拟地址是通过存取寄存器指定的地址空间控制单元转换的虚拟地址。在处于存取寄存器模式时,逻辑地址被处理成 AR 标识的虚拟地址。

[0126] 初始 (home) 虚拟地址

[0127] 初始虚拟地址是通过初始地址空间控制单元 (HASCE) 转换的虚拟地址。在处于初始空间模式时,逻辑地址和指令地址都被处理成初始虚拟地址。

[0128] 指令地址

[0129] 用于从存储提取指令的地址被称作指令地址。

[0130] 指令地址在实模式下被处理成实地址,在主空间模式、二级空间模式或存取寄存器模式下被处理成主虚拟地址,在初始空间模式中被处理成初始虚拟地址。在当前程序状态字中的指令地址和 EXECUTE 的目标地址是指令地址。

[0131] 有效地址

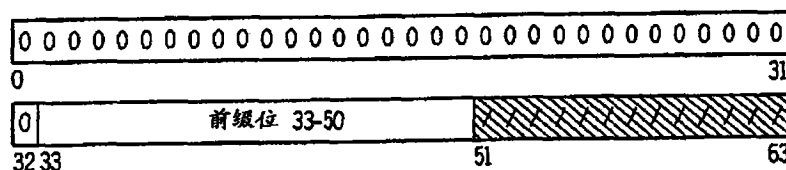
[0132] 在某些状况下,可以方便地使用术语“有效地址”。有效地址是在用动态地址转换或加前缀进行任何的转换之前存在的地址。有效的地址可以在寄存器中直接指定,或者可以从地址算术产生。地址算术是基址和位移的相加或基址、索引和位移的相加。

[0133] 加前缀

[0134] 加前缀提供了为每一个 CPU 将实地址范围 0-8191 分配到绝对存储的不同的块中的能力,因此允许多于一个 CPU 共享主存以便以最小的干扰来同时运行,特别是在中断处理过程中。加前缀导致范围 0-8191 中的实地址一一对应到由 CPU 的前缀寄存器的位位置 0-50 中的值识别的 8K 字节的绝对地址的块 (前缀区) 和由前缀寄存器中的值识别的实地址块一一对应到绝对地址 0-8191。其余的实地址与相应的绝对地址相同。这种转换允许每一个 CPU 存取所有的主存储,包括第一个 8K 字节和由其它 CPU 的前缀寄存器标识的位置。

[0135] 前缀是包含在前缀寄存器的位位置 0-50 中的 51 位的数量。在一个实施例中,前缀寄存器具有如下的格式:

[0136]



[0137] 前缀寄存器的格式

[0138] 当应用加前缀时,依据实地址的位 0-50,通过使用下述规则之一,实地址被转变(transform)为绝对地址:

[0139] 如果全为零,地址的位 0-50 用前缀的位 0-50 代替。

[0140] 如果等于前缀的位 0-50,地址的位 0-50 用 0 代替。

[0141] 如果不全为零且不等于前缀的位 0-50,地址的位 0-50 保持不变。

[0142] 只有提交给存储的地址才通过前缀来转换。该地址的源的内容保持不变。

[0143] 即使在前缀寄存器包括全部的 0 时,实地址和绝对地址之间也有区别,在上述情况下,实地址和其相应的绝对地址相同。图 13 例示了实地址和绝对地址之间的关系。

[0144] 地址空间是连续序列的整数(虚拟地址);还有允许每个数与存储中的字节位置相关联的具体的转换参数。所述序列始于 0,并从左到右进行。

[0145] 当 CPU 使用虚拟地址存取主存储器时,先通过动态地址转换(DAT)将虚拟地址转换为实地址或绝对地址。实地址可进一步受到加前缀以形成绝对地址。DAT 可以使用区第一表、区第二表、区第三表、段表和页表作为转换参数。用于特定地址空间的最高级表的标识(起始和长度)被称作地址空间控制单元(ASCE),并发现它由控制寄存器中的 DAT 使用或由存取寄存器指定。可替换地,用于地址空间的 ASCE 可以是实空间标识,这指示 DAT 将通过将虚拟地址处理成实地址且不使用任何表来转换虚拟地址。

[0146] DAT 在不同时间使用不同的控制寄存器中的 ASCE 或由存取寄存器指定的 ASCE。该选择由当前的程序状态字中指定的转换模式来确定。可利用四种转换模式:主空间模式、二级空间模式、存取寄存器模式和初始空间模式。依据转换模式,不同的地址空间是可寻址的。

[0147] 在任何时候,当 CPU 在主空间模式或二级空间模式时,CPU 能够转换属于两个地址空间-主地址空间和二级地址空间的虚拟地址。在任何时候,当 CPU 处于存取寄存器模式时,它能转换多达 16 个地址空间-主地址空间和多达 15 个 AR 指定的地址空间的虚拟地址。在任何时候,当 CPU 处于初始空间模式时,它能转换初始空间模式的虚拟地址。

[0148] 主地址空间是这样识别的,因为它包括主虚拟地址,该主虚拟地址通过主地址空间控制单元(PASCE)来转换。类似地,二级地址空间包括通过二级地址空间控制单元(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单元(AR 指定的 ASCE)转换的 AR 指定的虚拟地址,初始地址空间包括通过初始地址空间控制单元(HASCE)转换的初始虚拟地址。主 ASCE 和二级 ASCE 分别位于控制寄存器 1 和 7 中。AR 指定的 ASCE 可以在控制寄存器 1 和 7 中或在称作 ASN 二级表项的表项中。所述 HASCE 是在控制寄存器 13 中。

[0149] 动态地址转换

[0150] 动态地址转换是（例如，在存储引用期间）将虚拟地址转换为相应的主存储器地址（在实施例中为实地址或绝对地址）的过程。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初始虚拟地址。这些地址分别通过 PASCE、SASCE、AR 指定的 ASCE 或 HASCE 转换。在选择适当的 ASCE 后，转换过程对所有的四种类型的虚拟地址是相同的。

[0151] 地址转换模式

[0152] 有效地址是在由动态地址转换或加前缀进行任何的转换之前存在的地址（虚拟地址）。程序状态字中的控制动态地址转换的 3 个位是位 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 是关闭的，并且 CPU 处于实模式中。当 DAT 模式位为 1 时，DAT 是开通的，并且 CPU 是处于由地址空间控制位标识的转换模式中：二进制 00 标识主空间模式，二进制 01 标识存取寄存器模式，二进制 10 标识二级空间模式，二进制 11 标识初始空间模式。各种模式连同每种模式中的地址的处理方式如下所示。

[0153]

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	开通	主空间模式	主虚拟	主虚拟
1	0	1	开通	主空间模式	主虚拟	AR 指定的虚拟
1	1	0	开通	主空间模式	主虚拟	二级虚拟
1	1	1	开通	初始空间模式	初始虚拟	初始虚拟

[0154] 转换模式

[0155] 程序状态字是一个 128 位的字，它部分提供了指示寻址模式的 2 个位。在一个实施例中，位 31 是扩展寻址模式 (EA) 位，位 32 是基址寻址模式 (BA) 位。这两个位指示地址的大小。这两位中的每一个的状态是二进制 (1 或 0)。如果 EA 位是 0 和 BA 位是 0，指示 24 位寻址。如果指示 24 位寻址，64 位字的位 40-63 (64 位的实体通常被称为双字) 是地址所在的位置。在指令地址占据了 128 位实体 (四倍字) 的第二个 64 位的情况下，程序状态字中的位位置如下所述。在 24 位模式中，指令地址是在程序状态字的位 104-127 中。在 31 位模式中，指令地址是在程序状态字的位 97-127 中。在 64 位模式中，指令地址是在程序状态字的位 64-127 中。如果 EA 位是 0 和 BA 位是 1，指示 31 位寻址。适当的 64 位字包括位于位位置 33-63 的 31 位地址。如果 EA 位是 1 和 BA 位是 1，那么，64 位字的全部 64 位的位 0-63 包括地址。否则，指示异常情况。一旦获取寻址模式，需要确定 ASCE。

[0156] 地址空间控制单元 (ASCE)

[0157] 现在参考图 3, 其中例示了程序状态字是如何用来为虚拟地址的动态地址转换确定有效地址空间控制单元 (ASCE) 的一个实施例。例如 ASCE 可以指定 $2G$ ($G = 2^{30}$) 字节的地址空间。或者, 它可指定例如 $4T$ ($T = 2^{40}$) 字节、 $8P$ ($P = 2^{50}$) 字节或 $16E$ ($E = 2^{60}$) 字节的地址空间。或者, 它可以指定实空间标识。实空间标识导致虚拟地址被处理成存储中的实地址而不引用一个或多个地址转换表。

[0158] 程序状态字 300 包括转换 (T) 位 302 和地址空间 (AS) 位 304。在 306, 如果转换 (T) 位为 0, 地址是实地址 326。在 308, 如果地址空间 (AS) 等于 0 (二进制 00), 那么这个虚拟地址的有效 ASCE 是主地址空间控制单元 (PASCE) 310。在 312, 如果地址空间 (AS) 等于 1 (二进制 01), 那么有效 ASCE 是存取寄存器指定的地址空间控制单元 314。在 316, 如果地址空间 (AS) 等于 2 (二进制 10), 那么有效 ASCE 是二级地址空间控制单元 (SASCE) 318。否则, 地址空间 (AS) 等于 3 (二进制 11), 有效 ASCE 是起始地址空间控制单元 (HASCE) 322。

[0159] 在选择有效 ASCE 之后, 动态地址转换的过程优选是对所有四种类型的虚拟地址都相同。

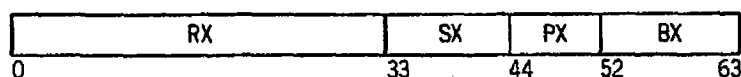
[0160] 段表标识或区表标识通过由操作系统在实或绝对存储中建立的表导致转换被执行。实空间标识引起虚拟空间被简单处理成实地址而不使用存储中的表。

[0161] 在转换过程中, 当使用段表标识或区表标识时, 三种类型的信息单元得以认定 - 区、段和页。区是跨越 $2G$ 字节和始自 $2G$ 字节的边界的连续虚拟地址块。段是跨越 1 兆字节和始自 1 兆字节的边界的顺序虚拟地址块。页是跨越 $4K$ 字节和始自 $4K$ 字节的边界的连续虚拟地址块。

[0162] 虚拟地址格式

[0163] 虚拟地址转换可涉及引用转换表层级结构的多个转换表以获取实地址或绝对地址。实地址会进一步受到加前缀操作以形成绝对地址。虚拟地址包括转换表层级结构中的转换表的项的索引。相应地, 虚拟地址分为四个主要字段。位 0-32 被称作区索引 (RX), 位 33-43 被称作段索引 (SX), 位 44-51 被称作页索引 (PX), 位 52-63 被称作字节索引 (BX)。在一个实施例中, 虚拟地址具有如下格式:

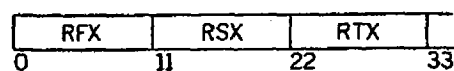
[0164]



[0165] 虚拟地址的格式

[0166] 如 ASCE 所确定的, 虚拟地址空间可以由一个区组成的 $2G$ 字节的的空间, 或者它可以是由高达 $8G$ 区组成的高达 $16E$ 字节的的空间。应用到 $2G$ 字节的地址空间的虚拟地址的 RX 部分必须全部为零; 否则, 认定有异常。虚拟地址的 RX 部分自身划分为三个字段。位 0-10 被称作区第一索引 (RFX), 位 11-21 被称作区第二索引 (RSX), 位 22-32 被称作区第三索引 (RTX)。在一个实施例中, 虚拟地址的位 0-32 有如下格式:

[0167]



[0168] 虚拟地址的 RX 的格式

[0169] 其中 RTX 是最左边的有效部分 (42 位地址) 的虚拟地址能够寻址 $4T$ 字节 (2048 个区), 其中 RSX 是最左边的有效 (significant) 部分 (53 位地址) 的虚拟地址能够寻址

8P 字节 (4193044 个区), 其中 RFX 是最左边的有效部分 (64 位地址) 的虚拟地址能够寻址 16E 字节 (8, 589, 934, 592 个区)。

[0170] 其中 RX 为零的虚拟地址可通过两个转换表: 段表和页表来转化实地址。随着 EDAT 设备被启用, 转换可只用段表来完成。RFX 可以不为零, 在这种情况下需要区第一表、区第二表和区第三表。如果 RFX 为零但 RSX 不为零, 需要区第二表和区第三表。如果 RFX 和 RSX 为零但 RTX 不为零, 需要区第三表。

[0171] 如果用于地址空间的 ASCE 不标识将引用转换到地址空间所需要的最高的表 (始于区第一表并向下继续到段表), 认定异常情况。

[0172] 虚拟地址的动态转换

[0173] 现在参考图 4, 例示了其中在图 3 中确定的有效 ASCE 被用于确定在虚拟地址转换中使用的转换表层级结构中的第一转换表的一个实施例。

[0174] 在一个实施例中, 控制寄存器 1 (CR1) 包括 PASCE。控制寄存器 7 (CR7) 包括 SASCE。控制寄存器 13 (CR13) 包括 HASCE, 并且由存取寄存器转换 (ART) 进程导出 (derive) 的地址空间第二表项 (ASTE) 包括存取寄存器指定的地址空间控制单元。有效 ASCE 400 是从这些位置之一选择的。

[0175] 有效 ASCE 400 的第一部分包括表起始 402, 表起始 402 包括标识或者区第一表、区第二表、区第三表或者段表的起始地址。所述表起始 (位 0..51) 被加 12 个二进制零以形成在虚拟地址转换中使用的转换表层级结构中的最高转换表的 64 位起始地址。有效 ASCE 400 还包括实空间控制 (R) 位 404 和 DT 位 406。如果实空间控制 (R) 位为零, 则选择器 408 解码 DT 位以确定哪一个特定的起始地址是表起始 402。如果 DT 位等于 3 (二进制 11), 表起始 402 标识区第一表 410。如果 DT 位等于 2 (二进制 10), 表起始 402 标识区第二表 412。如果 DT 位等于 1 (二进制 01), 表起始 402 标识区第三表 414。否则, 如果 DT 位等于零 (二进制 00), 表起始 402 标识段表 416。

[0176] 区第一表、区第二表或区第三表有时简称为区表。类似地, 区第一表标识、区第二表标识或区第三表标识有时称为区表标识。区、段和页表反映实存储的当前分配。页是用于虚拟存储分配的术语。实存储配给在固定块中。即便被分配给一组顺序的虚拟地址, 页也不需要实存储中相邻。

[0177] 当用于转换的 ASCE 是区第一表标识时, 转换过程包括使用例如区第一表、区第二表、区第三表、段表和可选的页表的多级查找。这些表驻留在实或绝对存储中。当 ASCE 是区第二表标识时, 区第三表标识、段表标识、在高于标识级的级别的表中的查找被省略, 较高级的表本身也被省略。

[0178] 现在参考图 5A, 其例示了使用转换表层级结构进行虚拟地址的动态地址转换的一个实施例。

[0179] 图 4 的有效 ASCE 400 包括标识类型 (DT) 位 406。如果 ASCE 的实空间控制 (R) 404 位为零, 则选择器 408 解码 DT 位以确定表起始 402 标识了哪一个起始地址。如果实空间控制 (R) 位是 1, 则如图 5B 所示, 在节点 D 564 发生动态地址转换。

[0180] 如果选择器 408 中的 DT 位等于 3 (二进制 11), 则转换表层级结构中的标识的第一个表是区第一表。表起始 402 在 502 与虚拟地址的区第一索引 (RFX) 508 部分算术相加以引用区第一表中的区第一表项 506。所述地址表起始 (在右边加 12 个 0 或乘以 4096) 被加

到索引与 8 的乘积（或索引右侧加 3 个 0）。所述区第一表项包括到在转换中使用的转换表层级结构中的下一较低级的表的区第二表起始 504。区第一表的下一较低级的表是区第二表。如果区第一表项的无效 (I) 位等于 1, 则区第一表项是无效的, 并且不能在转换中使用。指示异常情况。

[0181] 如果选择器 408 中的 DT 位等于 2 (二进制 10), 则转换表层级结构中的标识的第一个表是区第二表。表起始 402 在 510 与虚拟地址的区第二索引 (RSX) 516 部分算术相加以引用区第二表中的区第二表项 514。表起始 (在右边加 12 个 0 或乘以 4096) 被加到索引与 8 的乘积（或索引右侧加 3 个 0）。所述区第二表项包括到在转换中使用的转换表层级结构中的下一较低级的表的区第三表起始 512。区第二表的下一较低级的表是区第三表。如果区第二表项的无效 (I) 位等于 1, 则区第二表项是无效的, 并指示异常情况。

[0182] 如果选择器 408 中的 DT 位等于 1 (二进制 01), 则转换表层级结构中的标识的第一个表是区第三表。表起始 402 在 518 与虚拟地址的区第三索引 (RTX) 524 部分算术相加以引用区第三表中的区第三表项 522。表起始 (在右边加 12 个 0 或乘以 4096) 被加到索引与 8 的乘积（或索引右侧加 3 个 0）。所述区第三表项包括到在转换中使用的转换表层级结构中的下一较低级的表的段表起始 520。区第三表的下一较低级的表是段表。如果区第三表项的无效 (I) 位等于 1, 则区第三表项是无效的, 并指示异常情况。

[0183] 如果选择器 408 中的 DT 位等于 0 (二进制 00), 则转换表层级结构中的标识的第一个表是段表。表起始 402 在 526 与虚拟地址的段索引 (SX) 532 部分算术相加以索引段表中的段表项 530。表起始 (在右边加 12 个 0 或乘以 4096) 被加到索引与 8 的乘积（或索引右侧加 3 个 0）。所述段表项包括或在 528 所示的到页表的起始地址或段帧绝对地址 (SFAA)。如果段表项的无效 (I) 位等于 1, 则段表项是无效的, 并指示异常情况。

[0184] 在 538, 检查段表的 STE 格式控制 (FC) 位。如果 STE 格式控制是 1, 则段表项 530 包括段帧绝对地址 (SFAA) 552, 并且动态地址转换继续引用图 5C 中的节点 562。否则, 从段表获取的段表项包括页表起始地址, 并且动态地址转换继续引用图 5B 中的节点 560。

[0185] 现在参考图 5B。如果在段表项中的 STE 格式控制为 0, 则从段表中获得的段表项包括到转换表层级结构中的下一较低级的表的起始地址。段表的下一较低级的表是页表。从图 5A 的段表项 530 获取的页表起始 528 在 538 与虚拟地址的页索引 (PX) 534 部分算术相加以引用页表中的页表项 542。页表项包括页帧实地址 (PFRA) 546。当页帧实地址的最左边的位在 548 与虚拟地址的字节索引 (BX) 536 部分连结时, 获取 64 位的实地址 550。该 64 位实地址可以进一步受到加前缀操作以形成绝对地址。转换的虚拟地址引用在主存储或存储器中的期望的 4K 字节 (4096 字节) 的数据块。

[0186] 优选地, 在将虚拟地址动态转换到存储器地址中所用的信息与和虚拟地址关联的存储器块的地址一起被存储在转换后备缓冲器项标签 (tag) 中。随后的存储存取能够通过比较 ASCE 信息和虚拟地址信息与虚拟地址转换后备缓冲器标签而快速转换虚拟地址。如果标签被发现是虚拟地址的, 则能够使用存储器块的转换后备缓冲器地址而不执行对涉及的每一个转换表的缓慢顺序存取。在一个实施例中, 页帧实地址 (PFRA) 和由例如 ASCE 以及虚拟地址的 RX、SX 和 PX 部分组成的标签一起存储在转换后备缓冲器 544 的项中。这个虚拟地址的随后的转换之后从转换后备缓冲器中存储的信息导出 (derive)。

[0187] 现在参考图 5C。如果在段表项 530 中的 STE 格式控制是 1, 则段表项包括段帧绝

对地址 (SFAA) 552。当段帧绝对地址的最左边的位在 554 与虚拟地址的页索引 534 部分和字节索引 536 部分连结时, 获取 64 位的绝对地址 556。转换的虚拟地址引用在主存储或存储器中的期望的大的数据块。大的数据块大小至少为 1M 字节 (1,048,576 字节)。

[0188] 在一个实施例中, 段帧绝对地址 (SFAA) 以及虚拟地址的 RX 和 SX 部分被存储在转换后备缓冲器 544 中。这个虚拟地址的随后的转换之后从转换后备缓冲器中存储的信息导出。

[0189] 转换表项格式

[0190] 在转换中使用的转换表层级结构中的各种转换表项的实施例如下所示。

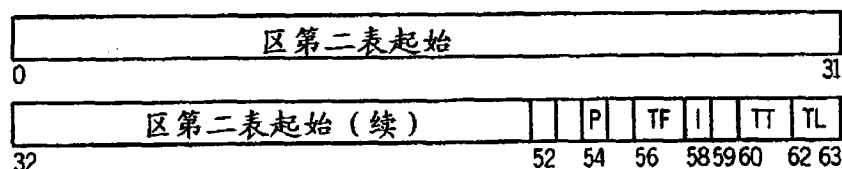
[0191] 区表项

[0192] 术语“区表项”是指区第一表项、区第二表项或区第三表项。从区第一表、区第二表或区第三表提取的项具有如下格式。包括项的表的级 (1、2 或 3) 由项中的表类型 (TT) 位识别。

[0193] 在一个实施例中, 区第一表项、区第二表项和区第三表项的格式如下所示:

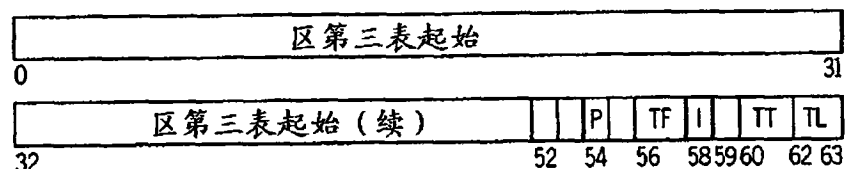
[0194] 区第一表项 (TT = 11)

[0195]



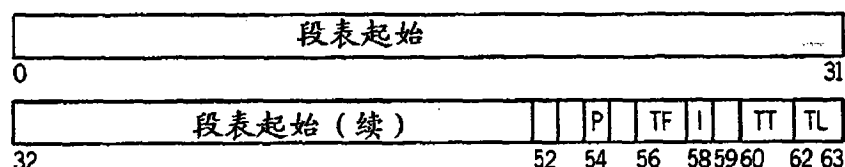
[0196] 区第二表项 (TT = 10)

[0197]



[0198] 区第三表项 (TT = 01)

[0199]



[0200] 区表项的格式

[0201] 区第二表起始、区第三表起始及段表起始: 区第一表项包括区第二表起始。区第二表项包括区第三表起始。区第三表项包括段表起始。下面的说明适用于这三个表起始中的每一个。项的位 0-51 具有在右侧加的 12 个零, 形成标识下一较低级的表的开始的 64 位地址。

[0202] DAT 保护位 (P): 当增强型 DAT 应用时, 位 54 被处理成与每一个随后的在转换中使用的区表项、段表项和适用时的页表项中的 DAT 保护位进行 OR 或操作。因此, 当该位为 1 时, DAT 保护应用于整个区或由区表项指定的区。当增强型 DAT 设备未安装时或当设备已安装但增强型 DAT 启用控制为 0 时, 区表项的位 54 将被忽略。

[0203] 区第二表偏移、区第三表偏移及段表偏移 (TF) : 区第一表项包括区第二表偏移。区第二表项包括区第三表偏移。区第三表项包括段表偏移。下面的说明适用于这三个表偏移中的每一个。项的位 56 和 57 指定在表的开始处丢失的下一较低级的表的部分的长度, 即, 这些位指定实际存在于下一较低级表中的第一项的位置。所述位以 4096 个字节为单位指定丢失部分的长度, 从而使丢失部分的长度 512 的倍数项可变。以 4096 个字节为单位的丢失部分的长度等于 TF 值。偏移字段的内容结合长度字段、位 62 和 63, 被用于证实通过下一较低级表转换的虚拟地址的部分 (RSX、RTX 或 SX) 是否标识在表中实际存在的项。

[0204] 区无效位 (I) : 区第一表项或区第二表项中的位 58 控制是否与该项相关联的区集 (set) 是可用的。区第三表项中的位 58 控制是否与该项相关联的单区是可用的。当位 58 是 0 时, 地址转换通过使用区表项进行。当该位是 1 时, 不能将该项用于转换。

[0205] 表类型位 (TT) : 区第一表项、区第二表项和区第三表项中的位 60 和 61 识别包括项的表的级别如下: 考虑到是转换中正在使用的 ASCE 的表标识的类型和至此已使用的表级数目, 位 60 和 61 必须识别正确的表级; 否则, 认定转换规范 (specification) 异常。下表显示了表类型位:

[0206]	位 60 和 61	区 - 表级
[0207]	11	第一
[0208]	10	第二
[0209]	01	第三

[0210] 用于区表项的表类型位

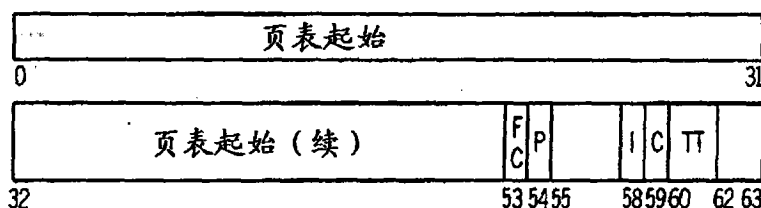
[0211] 区第二表长度、区第三表长度和段表长度 (TL) : 区第一表项包括区第二表长度。区第二表项包括区第三表长度。区第三表项包括段表长度。下面的说明适用于这三个表长度中的每一个。项的位 62 和 63 以 4096 个字节为单位指定下一较低级的表的长度, 从而使表的长度以 512 的倍数项可变。以 4096 个字节为单位的下一较低级的表的长度比 TF 值多 1。长度字段的内容结合偏移字段、位 56 和 57, 被用于证实通过下一低级表转换的虚拟地址的一部分 (RSX、RTX 或 SX) 是否标识在表中实际存在的项。区表项的所有其它位位置保留给可能的将来扩展并应包含零; 否则, 该程序可能将来不会兼容操作。当增强型 DAT 应用时, 区表项的保留位位置应该包含零, 即使该表项是无效的。

[0212] 段表项

[0213] 当增强型 DAT 不应用时或当增强型 DAT 应用且 STE 格式控制、段表项的位 53 为零时, 在一个实施例中, 从段表提取的项具有如下格式:

[0214] 段 - 表项 (T = 00, FC = 0)

[0215]

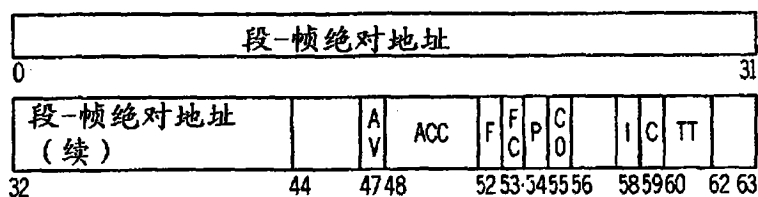


[0216] 段表项的格式 I

[0217] 当增强型 DAT 应用且 STE 格式控制是 1 时, 在一个实施例中, 从段表提取的项具有如下格式:

[0218] 段 - 表项 (T-00, FC = 1)

[0219]



[0220] 段表项的格式 II

[0221] 在段表项中的选择的区分配如下：

[0222] 页表起始：当增强型 DAT 不应用或增强型 DAT 应用但 STE 格式控制、段表项的位 53 为零时，位 0-52 右侧加 11 个零形成标识页表开始的 64 位地址。不可预知，该地址是实地址还是绝对地址。

[0223] 段帧绝对地址 (SFAA)：当增强型 DAT 应用和 STE 格式控制是 1 时，在项的位 0-43 右侧加 20 个零，形成段的 64 位绝对地址。

[0224] ACCF 有效性控制 (AV)：当增强型 DAT 应用和 STE 格式控制是 1 时，位 47 是存取控制位和提取保护位 (ACCF) 有效性控制。当 AV 控制为 0 时，段表项的位 48-52 被忽略。当 AV 控制为 1 时，如下所述地使用位 48-52。

[0225] 存取控制位 (ACC)：当增强型 DAT 应用、STE 格式控制是 1 和 AV 控制是 1 时，段表项的位 48-51 包括可用于施加到该地址的任何密钥控制的存取检查的存取控制位。

[0226] 提取保护位 (F)：当增强型 DAT 应用、STE 格式控制是 1 和 AV 控制是 1 时，段表项的位 52 包括可用于应用到该地址的任何密钥控制的存取检查的提取保护位。

[0227] STE 格式控制 (FC)：当增强型 DAT 应用时，位 53 为段表项的格式控制，具体如下：

[0228] 当 FC 位是 0 时，项的位 0-52 形成页表起始，位 55 保留。

[0229] 当 FC 位是 1 时，项的位 0-43 形成段帧绝对地址，位 47 是 ACCF 有效控制，位 48-51 是存取控制位，位 52 是提取保护位，位 55 是改变记录重写。当增强型 DAT 不应用时，位 53 被忽略。

[0230] DAT 保护位 (P)：当位 54 为 1 时，指示 DAT 保护适用于整个段。

[0231] 当增强型 DAT 不应用时，位 54 被处理成与转换中使用的页表项中 DAT 保护位进行 OR 或操作。

[0232] 当增强型 DAT 应用时，转换中使用的任何区和所有的区表项中的 DAT 保护位被处理成与段表项中的 DAT 保护位进行 OR 或操作；当 STE 格式控制为 0 时，在 STE 中的 DAT 保护位被进一步处理成与页表项中的 DAT 保护位进行 OR 或操作。

[0233] 改变记录重写 (CO)：当增强型 DAT 应用和 STE 格式控制是 1 时，段表项的位 55 是改变记录重写。当增强型 DAT 不应用或增强型 DAT 适用但 STE 格式控制为 0 时，段表项的位 55 被忽略。

[0234] 段无效位 (I)：位 58 控制是否与段表项关联的段是可用的。

[0235] 当该位为 0 时，地址转换通过使用段表项进行。

[0236] 当该位为 1 时，段表项不能用于转换。

[0237] 公共段位 (C)：位 59 控制段表项的转换后备缓冲器的使用。当增强型 DAT 不应用时或当增强型 DAT 应用但格式控制为零时，位 59 还控制段表项标识的页表的转换后备缓冲

器副本的使用。

[0238] 零识别秘密 (private) 段 ;在这种情况下,段表项和它标识的任何页表只可以与标识段表项驻留的段表的段表起始关联使用。

[0239] 1 标识公共段 ;在这种情况下,段表项和它标识的任何页表可继续用于转换对应于段索引的地址,即便指定了不同的段表。

[0240] 然而,如果秘密空间控制、位 55 在转换中使用的 ASCE 中是 1 或如果 ASCE 是实空间标识,则段表项的转换后备缓冲器副本和公共段的任何页表不能使用。当秘密空间控制在使用的 ASCE 中为 1 时,如果段表项是在转换期间从存储提取,则公共段位必须为 0。否则,认定转换规范异常。

[0241] 表类型位 (TT) :段表项的位 60 和 61 是二进制 00 以识别含有项的表的级。在区表项或段表项中的位 60 和 61 的所有可能值的含义如下 :

[0242] 位 60

[0243] 和 61 表级

[0244] 11 区第一

[0245] 10 区第二

[0246] 01 区第三

[0247] 00 段

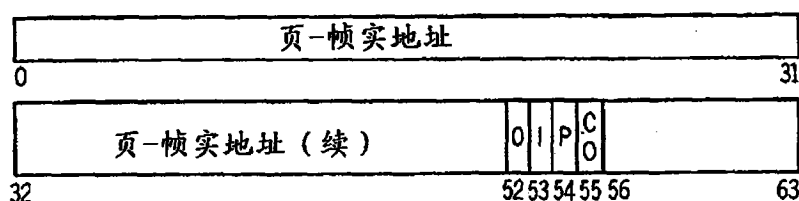
[0248] 表类型位 60、61

[0249] 考虑到转换中正在使用的 ASCE 的表标识的类型和至此已使用的表级数目,位 60 和 61 必须识别正确的表级 ;否则,认定转换规范异常。段表项的所有其它位位置保留用于将来可能的扩展并应包含零 ;否则,该程序将来不会兼容操作。当增强型 DAT 应用时,段表项的保留位位置应该包含零,即使该表项是无效的。

[0250] 页表项

[0251] 在一个实施例中,从页表提取的项具有如下格式 :

[0252]



[0253] 页表项的格式

[0254] 在页表项中的选择字段分配如下 :

[0255] 页帧实地址 (PFRA) :位 0-51 提供实存储地址最左边的位。当这些位在右侧与虚拟地址的 12 位字节的索引字段连结 (concatenate) 时,获取 64 位的实地址。

[0256] 页无效位 (I) :位 53 控制是否与页表项相关联的页是可用的。当该位是 0 时,地址转换通过使用页表项进行。当该位是 1 时,该页表项不能用于转换。

[0257] DAT 保护位 (P) :位 54 控制是否可以在页内进行存储存取。这种保护机制是在密钥控制的保护和低地址保护机制之外的。该位对提取存取没有影响。如果该位为 0,允许向页存储,但受如下的额外的约束 :

[0258] 在转换中使用的段表项中,DAT 保护位为零。

[0259] 当增强型 DAT 应用时,在转换中使用的所有区表项中,DAT 保护位都为零。

[0260] 其它保护机制

[0261] 如果该位为 1,不允许存储。如果没有更高优先异常状况存在,在 DAT 保护位为 1 时试图存储导致认定保护异常。当确定 DAT 保护是否应用于该页时,段表项中的 DAT 保护位被处理成与位 54 进行 OR 或操作。当增强型 DAT 应用时,当确定 DAT 保护是否应用时,在转换中使用的任何区表项中的 DAT 保护位也被处理成与位 54 进行 OR 或操作。

[0262] 改变记录重写 (CO):当增强型 DAT 不应用时,页表项的位 55 必须包括零;否则,作为使用该项进行地址转换执行指令的一部分,认定转换规范异常。当增强型 DAT 适用和 STE 格式控制为 0 时,页表项的位 55 是页的改变记录重写。

[0263] 项的位位置 52 必须包括零;否则,作为使用该项进行地址转换执行指令的一部分,认定转换规范异常。位位置 56-63 不被分配和被忽略。

[0264] 动态转换的另一个实施例

[0265] 本节介绍一种在使用虚拟地址存取主存储之前隐含执行的转换过程。

[0266] 虚拟地址转换是受程序状态字中的 DAT 模式位和地址空间控制位以及控制寄存器 1、7 和 13 中的 ASCE 控制和由存取寄存器指定的。当转换中使用的 ASCE 是区第一表标识时,通过都贮存在实或绝对存储中的区第一表、区第二表、区第三表、段表和页表执行转换。当 ASCE 是较低级类型的表标识(区第二表标识、区第三表标识或段表标识)时,转换只通过以标识的级开始的表级进行,并且虚拟地址位如果不为 0,会需要使用更高单个或多个级表的虚拟地址位必须全为 0,否则,认定 ASCE 类型异常。当 ASCE 是实空间标识时,虚拟地址被处理成实地址,并且不使用在实或绝对存储中的表项。

[0267] 用于特定地址转换的 ASCE 被称作有效 ASCE。相应地,当主虚拟地址被转换时,控制寄存器 1 的内容被用作有效 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,使用存取寄存器指定的 ASCE;对于初始虚拟地址,使用控制寄存器 13 的内容。

[0268] 当在有效 ASCE 中的实空间控制为 0 时,在 ASCE 中的标识类型指定表标识类型:区第一表、区第二表、区第三表或段表。虚拟地址的相应的部分(区第一索引、区第二索引、区第三索引或段索引)与标识中的表长度字段核查(check),它被加到标识中的起始以选择在指定表中的项。如果选择的项在其表之外,如标识中的表长度字段所确定的,或者如果所选择的项中的 I 位为 1,则依据标识所指定的表级,认定区第一转换、区第二转换、区第三转换或段转换异常。如果选择的项中的表类型位没有指示预期的表级,认定转换规范异常。

[0269] 通过有效 ASCE 选择的表项标识要使用的下一较低级的表。如果当前表是区第一表、区第二表或区第三表,那么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)与当前表项中的表偏移和表长度字段进行核查,它被加到项中的起始以选择在下一较低级表中的项。如果在下一表中的选择的项在其表之外,如当前表项中的表偏移和表长度字段所确定的,或者如果选择的项中的 I 位为 1,则依据下一表的级,识别区第二转换、区第三转换或段转换异常。如果在选择的项中的表类型位没有指示预期的表级,认定转换规范异常。

[0270] 通过连续的表级处理虚拟地址的部分继续进行,直到段表项已被选择为止。段表项包括应用于指定段的所有页的页保护位。

[0271] 虚拟地址的页索引部分被加到段表项中的页表起始,以选择页表中的项。如果页表项中的 I 位为 1,认定页转换异常。页表项包括表示虚拟地址转换的实地址的最左边的位,并且它包括仅应用于由页表项指定的页的页保护位。

[0272] 虚拟地址的字节索引字段未改变地用作实地址的最右边的位位置。

[0273] 为了消除与参考实或绝对存储中的转换表相关联的延迟,通常也将从表提取的信息放在一个特别的缓冲区、转换后备缓冲器中,并且涉及相同的表项的随后的转换可通过使用记录在转换后备缓冲器中的信息来执行。转换后备缓冲器也可以记录与实空间标识相关的虚拟等同的实转换。

[0274] 每当在地址转换期间出于从区表、段表或页表提取项的目的存取实或绝对存储时,密钥控制的保护不应用。

[0275] ASCE 标识的表中的查找

[0276] DT 控制、有效 ASCE 的位 60-61 指定 ASCE 的表标识类型和虚拟地址部分,该虚拟地址部分将利用如下的标识表进行转换:

[0277]	位 60 和 61	标识类型	由表转换的虚拟地址部分
[0278]	11	区第一表	区第一索引(位 0-10)
[0279]	10	区第二表	区第二索引(位 11-21)
[0280]	01	区第三表	区第三索引(位 22-32)
[0281]	00	段表	段索引(位 33-43)

[0282] 利用标识表的转换

[0283] 当位 60 和 61 具有二进制值 11 时,虚拟地址的区第一索引部分结合包含在 ASCE 中的区第一表起始,被用来从区第一表中选择项。通过向区第一表标识的位 0-51 的右侧加 12 个 0 并与区第一索引相加,该第一索引最右侧加 3 个 0 且最左侧加 50 个 0,获取实或绝对存储中的区第一表项的 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(是区第一索引的位 0 和 1)与表长度、区第一表标识的位 62 和 63 进行比较以证实寻址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相应位位置中的值,认定区第一转换异常。如果在转换中使用转换后备缓冲器中的区第一表项的等同物,与表的长度的比较可以被省略。从区第一表提取的项标识开始并指定相应的区第二表的偏移和长度。

[0284] 当 ASCE 的位 60 和 61 具有二进制值 10 时,虚拟地址的区第二索引部分结合包含在 ASCE 中的区第二表起始,被用来从区第二表中选择项。将虚拟地址的位 11 和 12(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相应位位置中的值,认定区第二转换异常。如果在转换中使用转换后备缓冲器中的区第二表项的等同物,与表的长度的比较可以被省略。区第二表查找过程在其他方面与区第一表查找过程相同;从区第二表提取的项标识开始并指定相应的区第三表的偏移和长度。

[0285] 当 ASCE 的位 60 和 61 具有二进制值 01 时,虚拟地址的区第三索引部分结合包含在 ASCE 中的区第三表起始,被用来从区第三表中选择项。将虚拟地址的位 22 和 23(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相应位位置中的值,认定区第三转换异常。区第三表查找过程在其他方面与区第一表查找过程相同,包括检查区第三表项中的表类型位。从区第三表提取的项标识开始并指定相应的段表的偏移和长度。

[0286] 当ASCE的位60和61具有二进制值00时,虚拟地址的段索引部分结合包含在ASCE中的段表起始,被用来从段表中选择项。将虚拟地址的位33和34(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相应位位置中的值,认定段转换异常。如果在转换中使用转换后备缓冲器中的段表项的等同物,与表的长度的比较可以被省略。段表查找过程在其他方面与区第一表查找过程相同,包括检查段表项中的表类型位。处理如下:

[0287] 当增强型DAT不应用时或当增强型DAT应用但STE格式控制为0时,从段表提取的项标识相应页表的开始,并且处理如在下文的“页表查找”中所述地继续进行。

[0288] 当增强型DAT应用和STE格式控制为1时,从段表提取的项包括段帧绝对地址的最左边的位。如果在转换中使用的任何区表项或段表项中的DAT保护位为1并且正在为其执行转换的存储引用是存储时,认定保护异常。

[0289] 区表项标识的表中的查找

[0290] 当有效ASCE是区表标识时,如前一节中所述地选择区表项。然后,选定(选择的)项的内容和虚拟地址的下一索引部分被用于选择可以是另一个区表或段表的下一较低级表中的项。当通过ASCE选择的表项是区第一表项时,虚拟地址的区第二索引部分结合包含在区第一表项中的区第二表起始,被用来从区第二表中选择项。通过向区第一表项的位0-51的右侧加12个0并与区第二索引相加,该区第二索引最右侧加3个0且最左侧加50个0,获取实或绝对存储中的区第二表项的64位地址。

[0291] 当形成区第二表项、区第三表项或段表项的地址时,不可预知是否加前缀(如果有的话)在加表索引值前被应用到包含在更高级的表项中的各表起始,还是加前缀被应用到表项地址,该表项地址通过将表起始和表索引值相加而形成。

[0292] 作为区第二表查找过程的一部分,将虚拟地址的位11和12(是区第二索引的位0和1)与表偏移、区第一表项的位56和57以及表长度、区第一表项的位62和63进行比较以证实寻址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相应位位置中的值,或者如果表长度字段中的值小于虚拟地址的相应位位置中的值,认定区第二转换异常。

[0293] 区第二表标识开始,并指定对应区第三表的偏移量和长度。

[0294] 当通过ASCE选择的表项是区第二表项时,或如果已通过区第一表项的内容选择了区第二表项,则使用虚拟地址的区第三索引部分结合包含在区第二表项中的区第三表起始来从区第三表中选择项。将虚拟地址的位22和23(是区第三索引的位0和1)与区第二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,或如果表长度小于位22和23,认定区第三转换异常。区第三表查询过程在其他方面与区第二表查询过程相同。从区第三表提取的项标识开始,并指定对应段表的偏移和长度。

[0295] 当通过ASCE选择的表项是区第三表项时,或如果已通过区第二表项的内容选择了区第三表项,则使用虚拟地址的段索引部分结合包含在区第三表项中的段表起始来从段表中选择项。将虚拟地址的位33和34(是段索引的位0和1)与区第三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,或如果表长度小于位33和34,则认定段转换异常。如果(1)ASCE中的秘密空间控制、位55为1,以及(2)从段表提取的项中的公共段位、位59为1,认定转换规范异常。段表查询过程在其他方面与区第二表查询过程相同。处理如下:

[0296] 当增强型 DAT 不应用时,或当增强型 DAT 应用但 STE 格式控制为 0 时,从段表提取的项标识对应页表的开始,并且处理会在如下“页表查找”中所述地继续进行。

[0297] 当增强型 DAT 应用并且 STE 格式控制为 1 时,从段表提取的项包含段帧绝对地址的最左边的位。如果在转换中使用的任何区表项或段表项中的 DAT 保护位为 1 并且正在为其执行转换的存储引用是存储时,认定保护异常。

[0298] 页表查询

[0299] 当增强型 DAT 不应用时,或当增强型 DAT 适用但 STE 格式控制为 0 时,使用虚拟地址的页索引部分结合段表项中包含的页表起始,以从页表选择一个项。

[0300] 通过向页表起始的右侧附加(加)11 个 0 并与页索引相加,该页索引最右侧加 3 个 0 且最左侧加 53 个 0,获取实存储或绝对存储中的段表项的 64 位地址。位位置 0 的执行不会发生。

[0301] 从页表提取的项指示页的可用性,并且包含页帧的实地址的最左边的位。检查页无效位、位 53,以证实对应页是否可用。如果该位为 1,认定页转换异常。如果位位置 52 包含 1,认定转换规范异常。当增强型 DAT 不应用时,或当增强型 DAT 应用但 STE 格式控制为 0 时,如果位位置 55 包含 1,则认定转换规范异常。当增强型 DAT 不适用时,或当增强型 DAT 适用并且 STE 格式控制为 0 时,如果位位置 55 包含 1,则也认定转换规范异常。如果在转换所使用的段表项中,在页表项中,或在转换期间使用的任何区表项中(当增强型 DAT 应用时),DAT 保护位为 1,以及如果为其执行转换的存储引用是存储,则认定保护异常。

[0302] 实地址与绝对地址的形成

[0303] 当有效 ASCE 是实空间标识时,虚拟地址的位 0-63 直接用作实存储地址。实地址可能进一步受到加前缀,而形成绝对地址。当有效 ASCE 不是实空间标识和在转换过程中没有遇到异常情况时,下述状况应用:

[0304] 当增强型 DAT 不应用时,或当增强型 DAT 应用但 STE 格式控制为 0 时,从页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结,该页帧实地址形成最左边的部分。该结果是对应于虚拟地址的实存储地址。实地址可能进一步受到加前缀,而形成绝对地址。

[0305] 当增强型 DAT 应用并且 STE 格式控制为 1 时,段帧绝对地址以及虚拟地址的页索引和字节索引部分分别从左到右连结,而形成对应于虚拟地址的绝对地址。

[0306] 转换期间的异常认定

[0307] 无效地址和无效格式能导致在转换过程期间认定出异常。当表项中包含的信息用于转换且发现其不正确时,认定存在异常。

[0308] 现在参考图 6,例示了到从段表项获取格式控制字段的点的动态地址转换的一个实施例的流程图。

[0309] 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来引用转换表中的合适表项。如果在 608 处从转换表提取的项不是段表项,那么在转换表的层级结构中的段表尚未被引用。在这种情况下,在 610,从表项中获取转换表层级结构中的下一较低级表的起始地址(origin)。使用虚拟地址的适当部分来引用转换中所使用的下一较低级表中的对应表项。

[0310] 例如,如果将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 部分来引用区第一表内的区第一表项。如果表起始地址是到区第二表,那么使用虚拟地址的 RSX 部分来引用区第二表内的区第二表项。如果表起始地址是到区第三表,那么使用虚拟地址的 RTX 部分来引用区第三表内的区第三表项。如果表起始地址是到段表,那么使用虚拟地址的 SX 部分来引用段表内的段表项。引用接连的表格,直到已提取到段表项。

[0311] 一旦已经提取了段表项,在 612 处,检查段表项 (STE) 格式控制位,以确定是否针对该特定的虚拟地址启用格式控制。如果 STE 格式控制为 0,则关于节点 614 发生动态地址转换。如果 STE 格式控制为 1,则关于节点 616 发生动态地址转换。

[0312] 动态地址转换 (STE 格式控制为 0)

[0313] 现在参考图 7,例示了当 STE 格式控制为 0 时从图 6 节点 614 开始的后续流程图。

[0314] 在 710,从段表项获取页表的起始地址。在 712,使用虚拟地址的 PX 部分来引用页表中的页表项。在 714,从页表项获取页帧实地址 (PFRA)。从页表项获取一个无效 (I) 位。如果在 716 处无效 (I) 位为 1,那么由于该项已被标记为无效,所以在 718 处,不能使用该页表项继续进行虚拟地址的转换。使用该页表项的虚拟地址的进一步转换在 722 处停止。如果在 716 处无效 (I) 位为 0,那么在 720 处,页帧实地址 (PFRA) 与虚拟地址的 BX 部分组合以生成实地址。实地址可能会进一步受到加前缀操作,以形成绝对地址。在 724 处,使用实地址来存取转换的虚拟地址寻址的数据块。

[0315] 动态地址转换 (STE 格式控制为 1)

[0316] 现在参考图 8,例示了从图 6 节点 616 的后续流程图。

[0317] 在 810,从段表项的一部分获取段帧绝对地址 (SFRA)。从段表项获取一个无效 (I) 位。如果在 812 处无效 (I) 位为 1,则由于段表项已被标记为无效,所以在 814 处,不能使用该段表项进行虚拟地址的进一步转换。在一个实施例中,将一个异常代码返回给请求转换的程序实体。使用该段表项的虚拟地址的进一步转换在 818 处停止。

[0318] 如果在 812 处无效 (I) 位为 0,那么在 816,段帧绝对地址 (SFRA) 与虚拟地址的 PX 和 BX 部分组合,以生成主存储或存储器中所期望的大的数据块的绝对地址。在 820,存取由转换的虚拟地址所寻址的期望的大的数据块。

[0319] 在每一个表项处,对无效位进行检查以确定所获取表项的有效性。本文进一步论述由保护转换后虚拟地址所寻址的数据块的其它转换保护机制。

[0320] 在另一实施例中,从转换表项获取限制信息。所述限制信息用于限制对虚拟地址范围的受限部分的存取。此后,对转换地址所寻址的期望的大数据块的存取被允许仅对地址范围的受限部分进行存取。限制信息是表偏移或表长度中的任何一个。

[0321] 还在另一实施例中,将虚拟地址转换中所使用的信息存储在至少一个转换后备缓冲器中。随后,使用来自转换后备缓冲器而非来自转换表层级结构的存储信息,执行从主存储中的数据块的虚拟地址到绝对地址的随后的转换。

[0322] 还在另一实施例中,如果转换并非原生于机器架构,则要识别预定的软件例程,以便进行仿真转换。预定的软件例程中包含多个指令。预定的软件例程被执行。

[0323] 对寻址数据块的保护

[0324] 一旦使用增强型 DAT 设备转换了虚拟地址,如本文所述,转换的虚拟地址所寻址

的主存储或存储器中的期望数据块可能会受到附加的保护机制。

[0325] DAT 保护

[0326] DAT 保护功能通过使用每个页表项和段表项中以及当增强型 DAT 设备安装时的每一个区表项中的 DAT 保护位来控制对虚拟存储的存取。这提供了对于不正当存储的保护。

[0327] 页表项的 DAT 保护位（位 54）对是否允许存储到对应的 4K 字节的页进行控制。当该位为 0 时，提取和存储都被允许；当该位为 1 时，只允许提取。当试图存储到受保护的页时，该页的内容保持不变，指令的操作单元或执行单元受抑制，并且保护的程序中断发生。

[0328] 段表项的 DAT 保护位（位 54）对是否允许存储到对应的 1M 字节的段进行控制，具体如下：

[0329] 当增强型 DAT 不应用时，或当增强型 DAT 应用并且 STE 格式控制为 0 时，段表项的 DAT 保护位被处理成进行 OR 或操作，进入段表项标识的页表中的每一个项的 DAT 保护位位置。因此，当段表项 DAT 保护位为 1 时，其效果如同是所标识的页表中的每个项中的 DAT 保护位都是 1。

[0330] 当增强型 DAT 应用并且 STE 格式控制为 1 时，段表项的 DAT 保护位对是否允许存储到对应的 1M 字节的段进行控制。当该位为 0 时，提取和存储都被允许，当该位为 1 时，只允许提取。当试图存储到受保护的段时，该段的内容保持不变，指令的操作单元或执行单元受抑制，并且保护的程序中断发生。

[0331] 当增强型 DAT 应用时，区表项的 DAT 保护位对是否允许存储到对应的区进行控制。区表项中的 DAT 保护位被处理成进行或操作，进入任何随后的区表项或转换中使用的段表项的 DAT 保护位位置。当 STE 格式控制位为 0 时，DAT 保护位进一步传到页表项。

[0332] DAT 保护应用于使用虚拟地址的所有存储类型的引用。

[0333] 现在参考图 9，例示了从转换中使用的段表项获取 DAT 保护位的一个实施例的流程图。

[0334] 在 902 处，获取要转换的虚拟地址。在 904，获取虚拟地址转换中使用的第一转换表的起始。在 906，以前面讨论的方式，使用虚拟地址的一部分来引用转换表中的项。在 908，由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。

[0335] 在 910，从虚拟地址转换中所使用的转换表的每一项获取 DAT 保护（P）位。应当理解，DAT 保护的有效级别是从转换中使用的每一个表项获取的多个 P 位的逻辑或。在 912，检查 DAT 保护位以确定转换的虚拟地址所寻址的数据块是否被 DAT 保护。如果在 912，DAT 保护的（P）位为 0，则在 914 处允许到转换虚拟地址所寻址的数据块的存储操作。如果在 912 处，DAT 保护（P）位为 1，则在 916 处阻止到转换虚拟地址所寻址的数据块的存储操作。可选地，将 DAT 保护字段所指示的保护级别应用于使用虚拟地址的所有存储类型的引用。如果启用 DAT 保护字段，则在试图向转换虚拟地址所寻址的期望数据块进行存储的情况下，指示保护异常状态。

[0336] 密钥控制式保护

[0337] 当密钥控制式保护应用于存储存取时，只有当存储密钥匹配于与存储存取请求相关联的存取密钥时才允许存储；当密钥匹配时或当存储密钥的保护位为 0 时，允许提取。当存储密钥的 4 个存取控制位等于存取密钥时或当存取密钥为 0 时，视为这些密钥匹配。保护动作概述如下。

[0338]

状态		是否允许存取存储	
存储密钥的提取保护位	密钥关系	提取	存储
0	匹配	是	是
0	不匹配	是	否
1	匹配	是	是
1	不匹配	否	否

注释:

匹配: 存储密钥的 4 个存取控制位等于存取密钥, 或存取密钥为 0。

是: 允许存取。

否: 不允许存取。提取时, 信息不能供程序使用, 而存储时, 存储位置的内容不变化。

[0339] 保护动作的概述

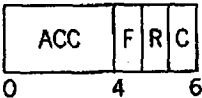
[0340] 当 CPU 开始对存储进行存取并且密钥控制保护应用时,PSW 密钥就是存取密钥,除非在通用寄存器中指定了针对 MOVE TOSECONDARY 和 MOVE WITH DESTINATION KEY 的第一操作数的存取密钥、针对 MOVE TO PRIMARY、MOVE WITH KEY 和 MOVE WITH SOURCE KEY 的第二操作数的存取密钥以及针对 MOVE PAGE 的第一操作数或第二操作数的存取密钥。PSW 密钥占据当前程序状态字的位 8-11。

[0341] 当 CPU 存取因为密钥控制式保护的原因被禁止时,该指令的执行会终止,并且保护异常的程序中断发生。然而,指令的操作单元或执行单元可能被抑制。

[0342] 存储密钥

[0343] 一个存储密钥与所述配置中可用的每个 4K 字节的存储块相关联。存储密钥不是地址存储的一部分。在一个实施例中,存储密钥的格式如下：

[0344]



[0345] 存储密钥的格式

[0346] 存储密钥中的位位置的分配如下：

[0347] 存取控制位 (ACC) :如果引用受到密钥控制式保护,那么当存储信息时以及当从受到提取保护的位置提取信息时,4 位存取控制位与 4 位存取密钥匹配。

[0348] 提取保护位 (F) :如果引用受到密钥控制式保护,那么提取保护位控制密钥控制式保护是否应用于提取类型的引用 :0 指示只监测存储类型的引用,以及允许用任何存取密钥提取 ;1 指示密钥控制式保护应用于提取和存储。

[0349] 引用位 (R) :每次为存储或提取信息而引用对应存储块中的位置时,通常将引用位设为 1。

[0350] 改变位 (C) :每次将信息存储到对应存储块中的一个位置时,将改变位设为 1。

[0351] 当增强型 DAT 应用时,下述附加条件生效:

[0352] 当 STE 格式控制 (FC,转换期间使用的段表项的位 53) 为 0 时,转换期间使用的页表项的位 55 是该页的改变记录重写 (CO)。当页表项中的 CO 位为 1 时,改变记录对于页面的任何存储操作都是不可预知的。

[0353] 当段表项中的格式控制位 (FC) 为 1 时,以下应用:

[0354] 段表项的位位置 47 包含 ACCF 有效性控制。ACCF 有效性控制确定存取控制的有效性和 STE 中的提取保护位。当 ACCF 有效性控制为 0 时,密钥控制式保护会使用对应于地址的 4K 字节块的存取密钥中的存取控制和提取保护位。

[0355] 当 ACCF 有效性控制为 1 时,段表项的位位置 48-52 包含段的存取控制位和提取保护位。当确定关于存储操作数的可存取性时,构成段的 4K 字节块的 STE 的位 48-52 或单个存储密钥的位 0-4 是否已检查是不可预知的。

[0356] 段表项的位 55 是段的改变记录重写 (CO)。当段表项中的 CO 位为 1 时,是否针对段的任何存储操作设置改变位是不可预知的。

[0357] 存储密钥存取

[0358] 存储密钥的引用处理为如下:

[0359] 当作出存储引用并且密钥控制式保护应用于该引用时,同时检查四位存取控制位和与存储位置相关联的提取保护位,并同时存储位置的引用。当 (1) 增强型 DAT 不应用;(2) 增强型 DAT 应用但存储是通过其中 STE 格式控制为 0 的段表项存取;或 (3) 增强型 DAT 应用,并且存储是通过其中 STE 格式控制为 1 的段表项存取,但 ACCF 有效性控制为 0 时,存取控制位和提取保护位处于 4K 字节块的存储密钥的位 0-4 中。当增强型 DAT 应用并且存储是通过其中 STE 格式控制和 ACCF 有效性控制都为 1 的段表项来存取时,存储密钥的 0-4 位或段表项的位 48-52 是否提供存取控制位和提取保护位是不可预知的。此外,当段表项提供存取控制位和提取保护位时,可以使用来自转换后备缓冲器的缓存副本。

[0360] 当增强型 DAT 应用并且,或者 (a)STE 格式控制为 0 并且 DAT 所用页表项中的改变记录重写为 1;或者 (b)STE 格式控制为 1 并且 DAT 所用段表项中的改变记录重写为 1 时,在执行存储操作时 CPU 是否设置了改变位是不可预知的。改变记录重写可以缓存在 PTE 或 STE 的转换后备缓冲器副本中。

[0361] 当条件性 SSKE 特征没有安装时,指令“SET STORAGE KEYEXTENDED”致使所有的 7 位都同时设置在存储密钥中。当条件性 SSKE 特征安装时,指令“SET STORAGE KEY EXTENDED”可用于基于程序指定的标准来设置全部或部分存储密钥。

[0362] 指令“INSERT STORAGE KEY EXTENDED”提供了 4K 字节块的存储密钥的位 0-6 的一致性图像 (consistent image)。类似地,指令“INSERT VIRTUAL STORAGE KEY”和“TESTPROTECTION”提供存取控制位和提取保护位的一致性图像。

[0363] 指令“RESET REFERENCE BIT EXTENDED”只修改引用位。存储密钥的所有其它位保持不变。同时地检查引用位和改变位以便设置状态码。

[0364] 引用位所提供的引用记录未必准确。然而,在大多情况下,引用记录与关联的存储引用近似一致。在没有存储发生的情况下,可设置改变位。

[0365] 如其它 CPU 所观测到的,因对存储密钥进行显性操作的指令 (INSERT STORAGE KEY

EXTENDED、INSERT VIRTUALSTORAGE KEY、RESET REFERENCE BIT EXTENDED 和 SETSTORAGE KEY EXTENDED) 而发生的存储密钥提取和存储在它们彼此之间以及在操作数引用之间是有顺序的,如同存储密钥存取自身分别是存储操作数提取和存储。

[0366] 存取控制和提取保护 (ACC/F)

[0367] 主存储或存储器中的数据块能通过段表项中提供的存取控制和提取保护位被进一步保护。

[0368] 存取有效性 (AV) 和存取控制 (ACC)

[0369] 现在参考图 10,例示了对应用到转换的虚拟地址所寻址的期望数据块的存取控制保护进行确定的流程图的一个实施例。

[0370] 动态地址转换进行到图 6 所述的节点 616,其中已从转换中所使用的段表提取到段表项。地址转换进行到图 8 中所述的节点 822,其中已获取到存储或存储器中的期望数据块的绝对地址。图 10 的流程图开始于节点 822。

[0371] 在 1002,从段表项获取存取有效性 (AV) 字段。在 1004,作出关于是否启用段表项中的存取有效性字段 (AV = 1) 的确定。如果存取有效性字段为 0,那么段表项中的存取控制 (ACC) 字段无效。在 1006,忽略存取控制字段的位。如果启用存取有效性字段,那么存取控制字段有效。在 1008,从段表项获取存取控制字段的位。在 1010,将段表项中的 ACC 字段所指示的存取控制保护级应用到由转换的虚拟地址所寻址的期望数据块。

[0372] 在另一实施例中,只有当存取控制字段与程序实体所提供的存取密钥匹配,才允许存储到期望数据块的存储操作,该程序实体例如是程序状态字或正在执行的程序指令的操作数。如果试图对期望的数据块进行存储并且存取控制字段不匹配存取密钥,则优选指示保护异常。

[0373] 还在另一实施例中,如果启用存取有效性字段,如果试图对期望的存储器块进行提取操作或存储操作并且段存取控制字段不匹配程序存取控制字段,则指示保护异常。

[0374] 存取有效性 (AV) 和提取保护 (F)

[0375] 现在参考图 11,例示了对应用到转换的虚拟地址所寻址的期望数据块的提取保护进行确定的流程图的一个实施例。

[0376] 动态地址转换进行到如图 6 所述的节点 616 的点,其中已从转换中使用的段表提取到段表项。地址转换进行到图 8 所述的节点 822,其中已获取存储或存储器中期望的数据块的绝对地址。图 11 的流程图开始于节点 822。

[0377] 在 1102,从段表项获取存取有效性 (AV) 字段。在 1104,作出关于是否启用段表项中的存取有效性字段 (AV = 1) 的确定。如果在 1104 处存取有效性字段为 0,那么在 1106,忽略提取保护位。如果在 1104 处启用存取有效性字段,那么,在 1108,从段表项获得提取保护位。在 1110,将段表项中的 F 字段所指示的提取保护级应用到转换虚拟地址所寻址的期望数据块。只有当提取保护字段匹配存取密钥,才允许从转换的虚拟地址所寻址的数据块进行提取操作。存取密钥由程序状态字或正在执行的程序指令的操作数提供。只有当提取保护字段被禁用,或当提取保护字段被启用且与虚拟地址相关联的存取控制字段等于存取密钥,才允许从期望数据块进行提取操作。如果存取密钥为 0,则允许从期望数据块进行提取。

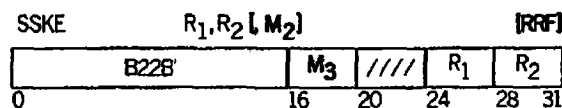
[0378] 在另一实施例中,如果程序存取密钥为 0,则允许进行到期望的存储器块的存储操

作以及从期望的存储器块的提取操作。还在另一实施例中,如果启用了存取有效性字段和提取保护字段,则响应于试图从期望的存储块进行提取操作并且段存取控制字段与程序存取控制字段不匹配,会指示保护异常。

[0379] 设置存储密钥扩展 (SSKE)

[0380] 存储密钥可以通过 SET STORAGE KEY EXTENDED (SSKE) 指令来设置。在一个实施例中, SSKE 指令具有如下格式:

[0381]



[0382] SSKE 指令的格式

[0383] 一个或多个 4K 字节块的存储密钥由第一操作数寄存器中的值替换。当条件性 SSKE 设备安装时,密钥设置操作的某些功能会被绕过。当条件性 SSKE 设备未安装时,或当条件性 SSKE 设备安装并且 M3 字段的 MR 和 MC 位都为 0 时,通用寄存器 R2 的内容所寻址的 4K 字节块的存储密钥由来自通用寄存器 R1 的位替换。该指令在不改变状态码的情况下完成。

[0384] 当条件性 SSKE 设备安装并且 MR 和 MC 位中的一个或两个为 1 时,将通用寄存器 R2 的内容所寻址的存储密钥的存取控制位、提取保护位以及可选的引用位和改变位与通用寄存器 R2 中的相应位进行比较。如果比较的位是相等的,不对密钥进行改变;否则,密钥的选择位由通用寄存器 R1 中的相应位替换。将进行任何修改之前的密钥插入到通用寄存器 R1 中,并且结果由状态码指示。

[0385] 当增强型 DAT 设备安装时,如下所述,可针对受多块控制的控制的同一 1MB 块内的多个 4K 字节块的存储密钥重复进行上述操作。在一个实施例中, M3 字段有如下格式:

[0386]



[0387] M3 字段的格式

[0388] M3 字段的位定义如下:

[0389] 保留位:保留位 0。

[0390] 引用位更新掩码 (MR):如下所述, MR 位、M3 字段的位 1 控制是否可绕过存储密钥中的引用位的更新。

[0391] 改变位更新掩码 (MC):如下所述, MC 位、M3 字段的位 2 控制是否可绕过存储密钥中的改变位的更新。

[0392] 多块控制 (MB):如在多个 4K 字节块中设置存储密钥中所述的, MB 位、M3 字段的位 3 控制是否可以设置多个 4K 字节存储块的存储密钥。

[0393] 当增强型 DAT 设备未安装时, M3 字段的位位置 3 保留。当条件性 SSKE 设备安装时,处理如下:

[0394] 当 MR 和 MC 位以及 M3 字段的位 1 和 2 都为 0 时,指令完成,如同条件性 SSKE 设备没安装。通用寄存器 R2 的内容所寻址的 4K 字节块的存储密钥由来自通用寄存器 R1 的位所替换(替代),并且指令在不改变状态码的情况下完成。

[0395] 当 MR 和 MC 位中的一个或两个为 1 时,处理如下:

[0396] 在任何修改之前,通用寄存器 R2 所寻址的 4K 字节块的存储密钥的内容放置在通用寄存器 R1 中的位位置 48-54 中,通用寄存器 R1 的位 55 设置为 0。该寄存器的位 0-47 和 56-63 保持不变。如果提取存储密钥时检测到无效检查块码 (CBC),则 (a) 4K 字节块的整个存储密钥由通用寄存器 R1 的位 56-62 替换, (b) 通用寄存器 R1 的位位置 48-55 的内容是不可预知的,并且 (c) 指令通过设置状态码 3 来完成。

[0397] 所标识的 4K 字节块的存储密钥的存取控制位和提取保护位与通用寄存器 R1 的位 56-60 中的相应字段进行比较。如果各个字段不相等,那么 4K 字节块的整个存储密钥由来自通用寄存器 R1 的位替换,指令通过设置状态码 1 来完成。当存储密钥中的存取控制位和提取保护位等于通用寄存器 R1 中的相应位时,处理继续进行,如下所述。

[0398] 当 MR 和 MC 位都为 1 时,指令通过设置状态码 0 来完成。在这种情况下,存储密钥保持不变。

[0399] 当 MR 为 0 和 MC 为 1 时,将用于标识的 4K 字节块的存储密钥的引用位与通用寄存器 R1 的位 61 进行比较。如果位是相等的,那么指令通过设置状态码 0 来完成。在这种情况下,存储密钥保持不变。如果位不相等,那么,或者 (a) 用于标识的 4K 字节块的整个存储密钥被通用寄存器 R1 中的位替换,指令通过设置状态码 1 来完成;或者 (b) 用于存储密钥的引用位被通用寄存器 R1 的位 61 替换,密钥的改变位是不可预知的,指令通过设置状态码 2 来完成。不可预知,是否设置了状态码 1 或 2。

[0400] 当 MC 位为 0 且和 MR 位为 1 时,将用于标识的 4K 字节块的存储密钥的改变位与通用寄存器 R1 的位 62 进行比较。如果位是相等的,那么指令通过设置状态码 0 来完成。在这种情况下,存储密钥保持不变。如果位不相等,那么,或者 (a) 所标识的 4K 字节块的整个存储密钥由通用寄存器 R1 中的位替换,指令通过设置状态码 1 来完成;或者 (b) 存储密钥的改变位由通用寄存器 R1 的位 62 替换,密钥的引用位是不可预知的,并且指令通过设置状态码 2 来完成。不可预知,是否设置了状态码 1 或 2。

[0401] 当增强型 DAT 设备未安装时或当该设备已安装但多块控制为 0 时,通用寄存器 R2 包括实地址。当增强型 DAT 设备安装并且多块控制为 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 标识实存储或绝对存储中的 4K 字节块,寄存器的位 0-39 和 52-63 被忽略。在 31 位寻址模式下,通用寄存器 R2 的位 33-51 标识实存储或绝对存储中的 4K 字节块,并且寄存器的位 0-32 和 52-63 被忽略。在 64 位寻址模式下,通用寄存器 R2 的位 0-51 标识实存储或绝对存储中的 4K 字节块,寄存器的位 52-63 被忽略。因为它是实或绝对地址,所以标识存储块的地址不会受到动态地址转换。对存储密钥的引用不会受到保护异常。

[0402] 新的 7 位存储密钥值或其选定 (选择的) 位从通用寄存器 R1 的位位置 56-62 获取。寄存器的位位置 0-55 和 63 的内容被忽略。当条件性 SSKE 设备安装以及 MR 和 MC 位中的一个或两个为 1 时,位位置 63 应该包含 0;否则,该程序将来不会兼容操作。

[0403] 系列化和检点同步的功能在操作开始前执行,并在操作完成后再次执行,除了条件性 SSKE 设备安装并且所产生的状态码为 0 时,不可预知,在操作完成后是否执行系列化和检点同步的功能。对于任何 CPU 或通道程序在由该指令设置密钥之前或之后完成的到标识的 4K 字节块的任何存储存取,为块在存储密钥中将引用位和改变位相关设置为 1 也是分

别在这个指令执行之前或之后完成的。

[0404] 在多个 4K 字节块中设置存储密钥

[0405] 当增强型 DAT 设备未安装时,或当该设备安装但多块控制为 0 零时,如上所述,用于单个 4K 字节块的存储密钥被设置。当增强型 DAT 设备安装并且多块控制为 1 时,可对 1M 字节块内部的多个 4K 字节块的存储密钥进行设置,从由第二个操作数地址指定的块开始,继续向右进行,每一个接续的块直至下一个 1M 字节的边界。在这种情况下,SET STORAGE KEY EXTENDED 是可中断的,并且处理如下:

[0406] 当发生中断(终止之后的中断除外)时,通用寄存器 R2 已经更新,使得指令在重新执行时从中断点重新开始。如果 MR 或 MC 位中的一个或两个为 1,状态码是不可预知的;否则,状态码不变。

[0407] 当指令在不中断的情况下完成时,通用寄存器 R2 已经更新到下一个 1M 字节的边界。如果 MR 或 MC 位中的一个或两个为 1,状态码 3 被设置;否则,状态码不变。

[0408] 在上述两种情况的任一情况中,当 MR 或 MC 位中的一个或两个为 1 时,通用寄存器 R1 中的位 48-55 是不可预知的。

[0409] 当发生多块处理并且 R1、R2 字段标识同一个寄存器时,第二操作数地址放置在寄存器中。当多块处理发生在 24 位或 31 位寻址模式下时,最左边的位(不是通用寄存器 R2 的位位置 32-63 中的地址的一部分)被设置为 0;寄存器的位 0-31 不变。

[0410] 所得到的状态码:

[0411] 当条件性 SSKE 设备未安装时,或当 M3 字段的 MR 和 MC 位都为 0 时,状态码保持不变。当条件性 SSKE 设备安装且 MR 和 MC 位中的一个或两个为 1 时,状态码被如下设置:

[0412] 0- 存储密钥未被设置

[0413] 1- 整个存储密钥被设置

[0414] 2- 部分存储密钥被设置

[0415] 3- 整个存储密钥被设置;通用寄存器 R1 的位 48-55 是不可预知的。

[0416] 程序异常:

[0417] 寻址(由通用寄存器 R2 指定的地址)

[0418] 特许操作。

[0419] 改变记录

[0420] 改变记录提供关于在主存储中更换页时哪些页必须存放在辅助存储中的信息。改变记录使用存储密钥的改变位(位 6)。每当存储存取导致相应的存储块的内容改变,并且或者 (a) 增强型 DAT 不应用或者 (b) 增强型 DAT 应用并且下述之一为真时,改变位被设为 1:

[0421] DAT 使用的段表项中的 STE 格式控制为 0,且 DAT 使用的页表项中的改变记录重写(C0)为 0。

[0422] DAT 使用的段表项中的 STE 格式控制为 1 且 DAT 使用的段表项中的改变记录重写(C0)为 0。

[0423] 不改变存储内容的存储存取可以或不将改变位设置为 1。如果存取被禁止,不会为试图存储而将改变位设为 1。特别是:

[0424] 对于 CPU,每当对该存取存在存取异常时,或每当存在优先级高于该存取的存取异

常优先级的异常时,存储存取被禁止。

[0425] 对于通道子系统,每当该存取存在密钥控制的保护违犯时,存储存取被禁止。

[0426] 改变记录总是活动的,对寄存器的所有存储存取都发生,包括由任何 CPU(除在如本文所述被改变记录重写抑制时以外)、任何操作设备或通道子系统做出的存储存取。改变记录因机器所作出的隐含引用而发生,例如作为中断的一部分的隐含引用。

[0427] 改变记录不因下列指令的操作数而发生,因为它们不改变存储地址而直接改变存储密钥:

[0428] RESET REFERENCE BIT EXTENDED

[0429] SET STORAGE KEY EXTENDED(改变位被设置为指定值)

[0430] 在 CPU 重试时,已经从 0 改为 1 的改变位未必恢复到 0。

[0431] 改变记录重写 (CO)

[0432] 每当存储存取导致相应存储块的内容改变时,存储密钥的改变位被设置为 1。不改变存储内容的存储存取可以或可以不将改变位设置为 1。如果存取被禁止,那么不因试图存储而将改变位设置为 1。改变记录重写允许绕过存储密钥的改变位的设置。

[0433] 当增强型 DAT 应用并且和虚拟地址通过 DAT 表项来转换时,在段表项和页表项二者的位 55 中提供改变记录重写 (CO)。当段表项的 STE 格式控制 (FC) 位 53 为 0 时,在页表项中的改变记录重写应用。当 PTE 中的改变记录重写为 0 时,为 4K 字节块的存储操作发生改变记录。当改变记录重写为 1 时,不可预知,为 4K 字节块的存储操作是否发生改变记录。当 STE 格式控制为 1 时,STE 中的改变记录重写应用。当 STE 中的改变记录重写为 0 时,为该段的 256 个 4K 字节块中的任何一个的存储操作都发生改变记录。当 STE 中的改变记录重写为 1 时,不可预知,该段的 256 个 4K 字节块中的任何一个是否发生改变记录。

[0434] 改变记录重写并不应用于实地址或绝对地址,也不应用于通过实空间标识转换的虚拟地址。

[0435] 现在参考图 12,例示了从虚拟地址的动态地址转换中使用的段表项获取改变记录重写字段的一个实施例的流程图。

[0436] 动态地址转换进行到图 6 所述的节点 616 的点,其中已经从转换中使用的段表提取了段表项。地址转换继续到如图 8 中说明的节点 822,其中已经获取到存储或存储器中的期望数据块的绝对地址。图 12 的流程图开始于节点 822。

[0437] 在 1210,从转换中使用的段表项获取改变记录重写 (CO) 字段。改变记录重写使得存储密钥的改变位被重写。在 1212,做出一个确定,关于对转换的虚拟地址所寻址的期望数据块执行的存取是存储类型操作还是提取类型操作。如果存取是提取类型的存取,则在 1213,改变位的设置不会发生,并且改变记录重写不应用。如果存取是存储类型的操作,则在 1214,做出关于在段表项中的改变记录重写是否被启用的确定。如果在 1214 处改变记录重写未被启用,则在 1218 将与期望的块相关联的存储密钥的改变位设置为 1。如果在 1214 处改变记录重写被启用,则在 1216 处,不设置相关联的存储密钥的改变位。在 1220 处,做出关于期望的数据块已被修改的指示。

[0438] 在另一实施例中,如果增强型 DAT 应用并且 STE 中的格式控制没有启用,则从段表项获取页表的起始。虚拟地址的页索引部分被用于引用页表中的页表项。从页表项获取主存储器中的小数据块的页帧实地址和页改变重写字段。每个小的数据块都有一个含有改变

位的相关联的存储密钥。小的数据块的大小比大的数据块小,并且由 4K 字节(4096 个字节)的数据块组成。主存储器中的期望实数据块的转换地址包括页帧实地址和虚拟字节索引部分的组合。作出关于对期望的实数据块进行的是存储类型的操作还是提取类型的操作的确定。如果存取操作是提取类型的操作,那么改变记录重写不应用。如果存取是存储类型的操作,那么,在与虚拟地址相关联的改变记录重写字段被禁止的情况下,与期望的 4K 块相关联的存储密钥的改变位被设置为 1。还提供了期望的 4K 块已被修改的指示。

[0439] 在另一实施例中,如果启用改变记录重写字段,则与期望的 4K 数据块相关联的存储密钥的改变位不可预知。

[0440] 商业的实现

[0441] 虽然本文提及了 **IBM**® 的 z/Architecture 架构,但本发明的一个或多个方面同样适用于其它的机器架构和 / 或采用可分页实体或类似构造的计算环境。

[0442] 本文公开的 eDAT 设备和其它格式、指令和属性的商业实现可以由硬件来实现,或可通过诸如由编程人员(例如操作系统编程人员)用(例如)汇编语言写的操作系统程序的来实现。可将这种编程指令存储在存储介质上,用于在(诸如) **IBM**® System z 服务器的计算环境中以原生方式执行,或者可选地在执行其它架构的机器中执行。这些指令能够在现有和将来的 **IBM**® 服务器和其它机器或大型机上进行仿真。它们可以在其中执行通常处于仿真模式的机器上执行。

[0443] 本发明的一个或多个方面也同样适用于(例如)虚拟机仿真,其中一个或多个分页实体(例如,客机(guest))在一个或多个处理器上执行。作为一个例子,分页客机由如下出版物中定义:**IBM**® 出版物 No. SA22-7095(1985)《**IBM**® System/370 Extended Architecture》中描述的开始解释性执行(Start Interpretive Execution(SIE))架构,该出版物通过整体引用并入本文。

[0444] 在仿真模式中,对正处于仿真的具体指令解码,并例如在子程序或驱动器中执行子例程以便实现各指令,或使用一些其它技术来为具体硬件提供驱动器,如本发明技术人员在理解了本发明说明之后的能力范围之内。各种软件和硬件仿真技术在美国专利中都有所描述,其中包括:5,551,013、5,574,873、5,790,825、6,009,261、6,308,255 以及 6,463,582,上述每个专利通过引用并入本文。其它的许多教示进一步例示了实现为目标机构造的指令集的仿真的多种方式。

[0445] 其它的变化和架构

[0446] 本文所述的各种实施例只是例子。这些实施例可以有多种变化而不背离本发明的精神。

[0447] 本发明的一个或多个性能可以以软件、固件、硬件或其一些组合来实现。本发明的各方面有利于各种类型的环境,包括具有多个区的其它环境和非分区环境。此外,可以没有中央处理器复合体,但可以有连在一起的多个处理器。本发明的各个方面也适用于单处理器环境。

[0448] 虽然本文中描述了特定的环境,但同样地,可以对环境进行许多变化,而不背离本发明的精神。例如,如果环境是逻辑分区的,那么环境中所包含的逻辑分区可多可少。此外,可以将多个中央处理复合体连一起。这些只不过是背离本发明精神的变化中的一部分。此外,还可能存在其它变化。

[0449] 虽然术语“页”是用来指固定大小或预定义大小的存储区,但页的大小能够改变。类似地,块的大小能够改变。可以存在不同大小的块和 / 或页。一个页可以等于一个块。可选地,可使用其它结构或另外可以通过软件和 / 或硬件来实现其它结构。此外,在本文所述的例子中,可以有多种变化,这些变化包括但不限于: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;不同大小的字段等等。同样地,这些只是提供作为例子。可能存在多种变化。

[0450] 处理单元包括诸如客机、主机、其它处理器、仿真器,虚拟机和 / 或其它类似构造的可分页实体。缓冲器包括存储区和 / 或内存区(memory)以及不同类型的数据结构,包括但不限于数组或分页实体。表还可以包括其它数据结构。指令可以引用其它寄存器。此外,页、段和 / 或区变化的大小可以是不同于本文所述的大小。

[0451] 本发明的一个或多个方面可以包含在具有(例如)计算机可用介质或机器可读介质的制造品(例如,一个或多个计算机程序产品)中。所述介质中实施了(例如)计算机可读程序代码装置或逻辑(例如指令、代码和命令等),以提供和促进本发明的功能。制造品可被包括作为计算机系统的一部分或单独出售。另外,可以提供至少一个程序存储设备,该程序存储设备可由实施至少一个指令程序的机器读取,该至少一个指令程序可由执行本发明功能的机器来执行。

[0452] 本文描绘的流程图是例示性的。可以对这些描述的图示或步骤或操作进行许多变化,而不背离本发明的精神。例如,步骤可以以不同的顺序执行,或者可对步骤进行添加、删除或修改。所有这些变化都视为本发明要求保护的一部分。

[0453] 虽然在这里已经详细描绘和说明了其中的实施例,但显然相关领域的技术人员可作出各种修改、添加和替换等,而不背离本发明精神,并因此,这些被视为在如下权利要求书定义的本发明的范围内。

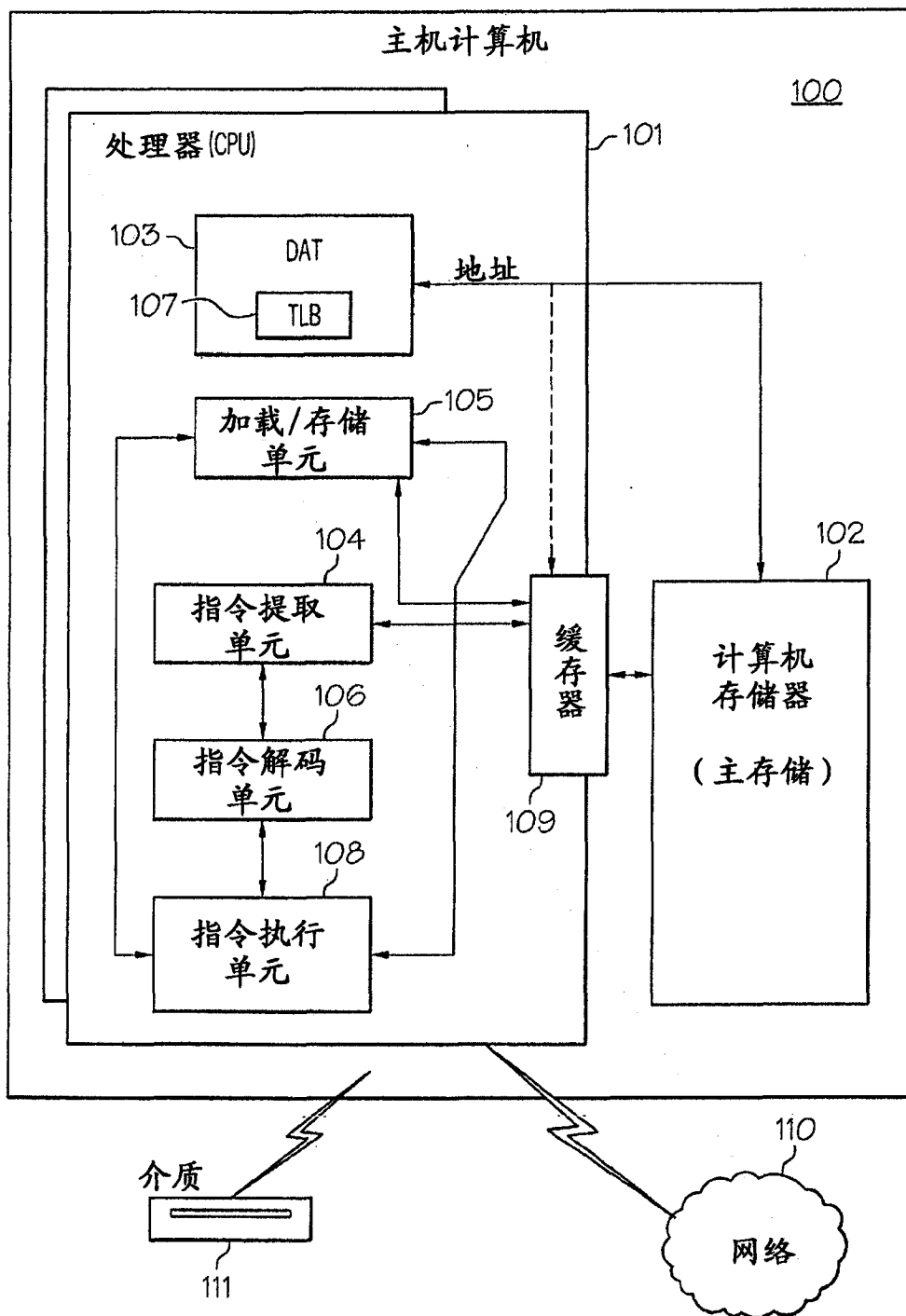


图1(现有技术)

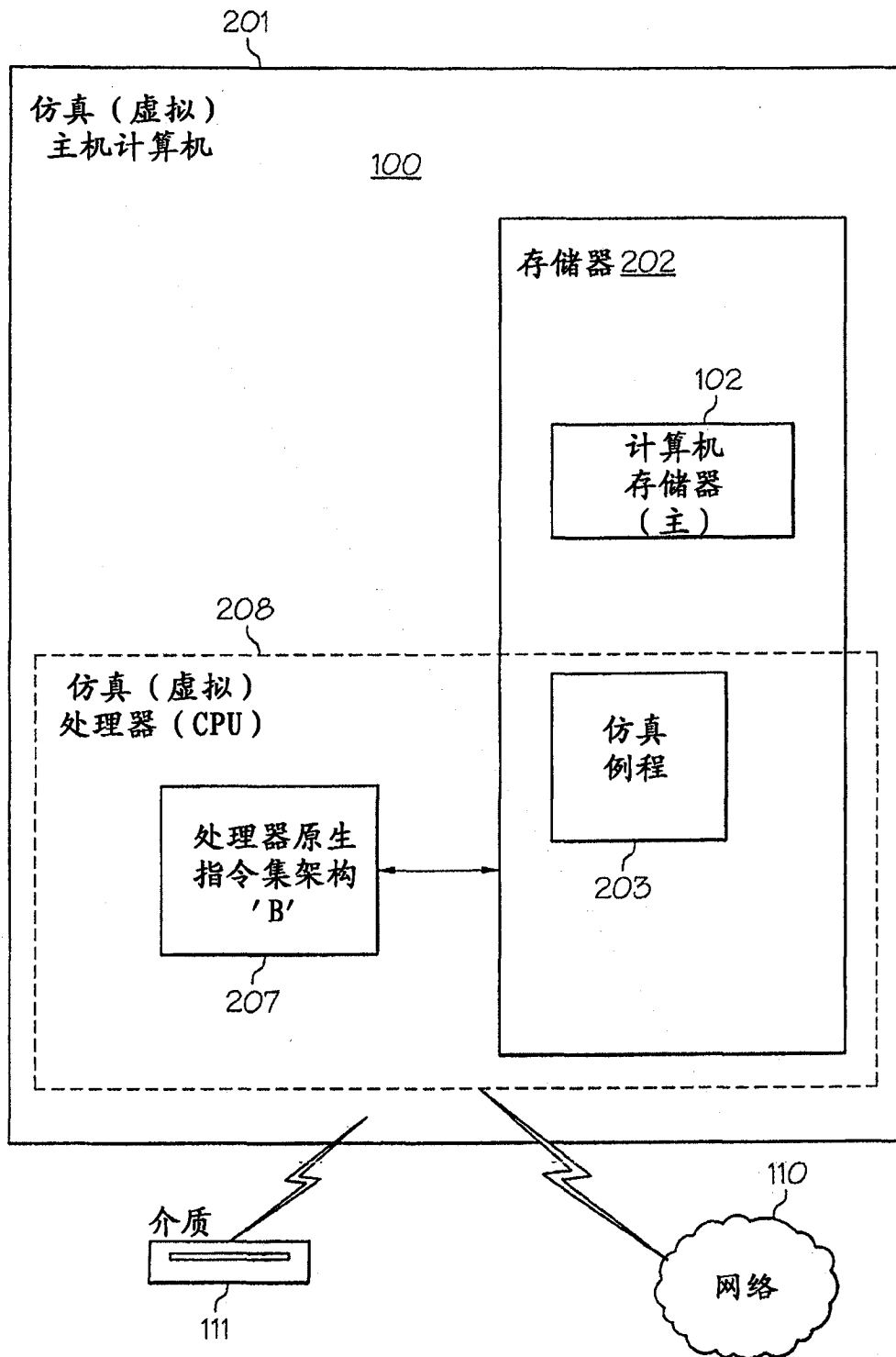


图 2(现有技术)

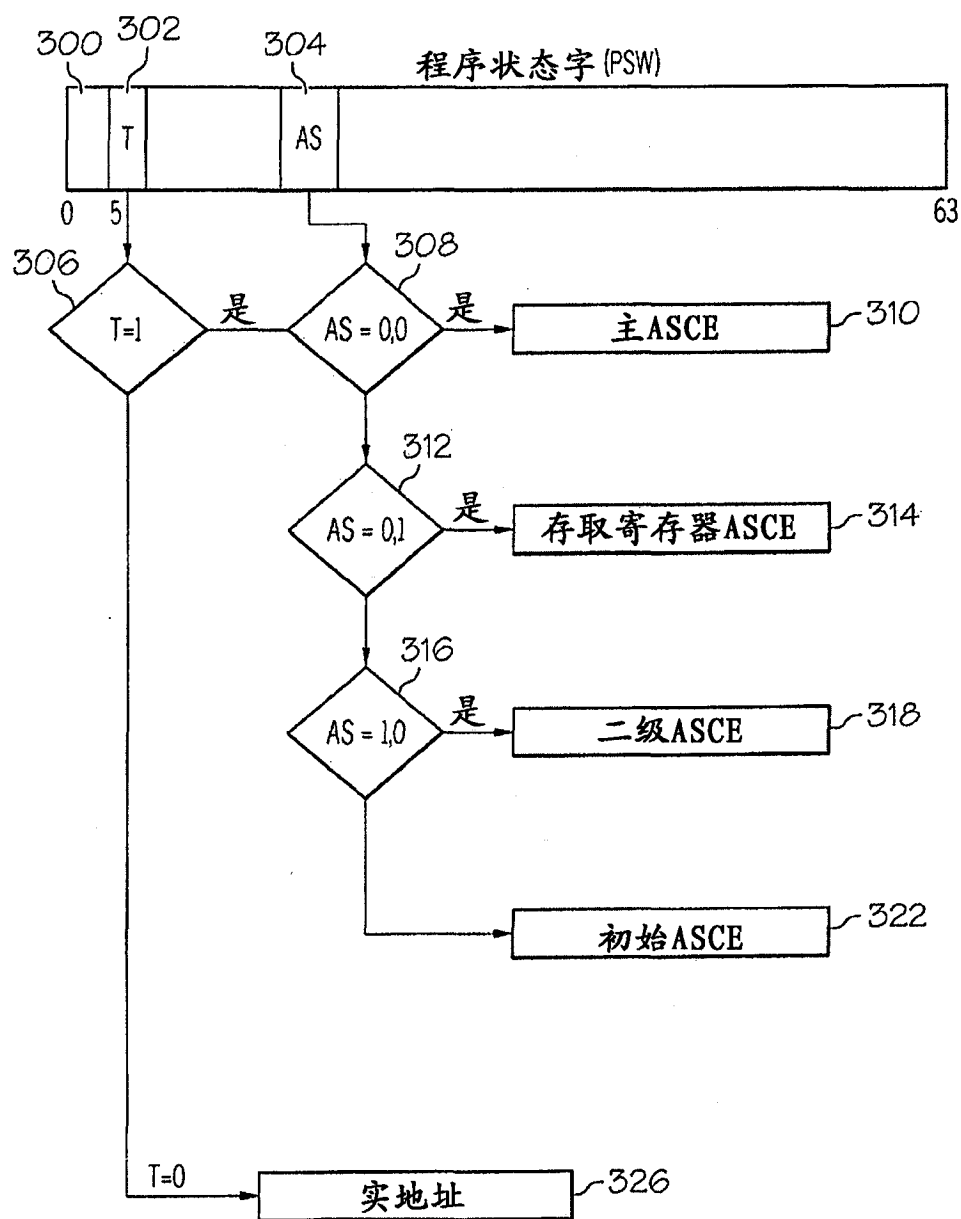


图 3(现有技术)

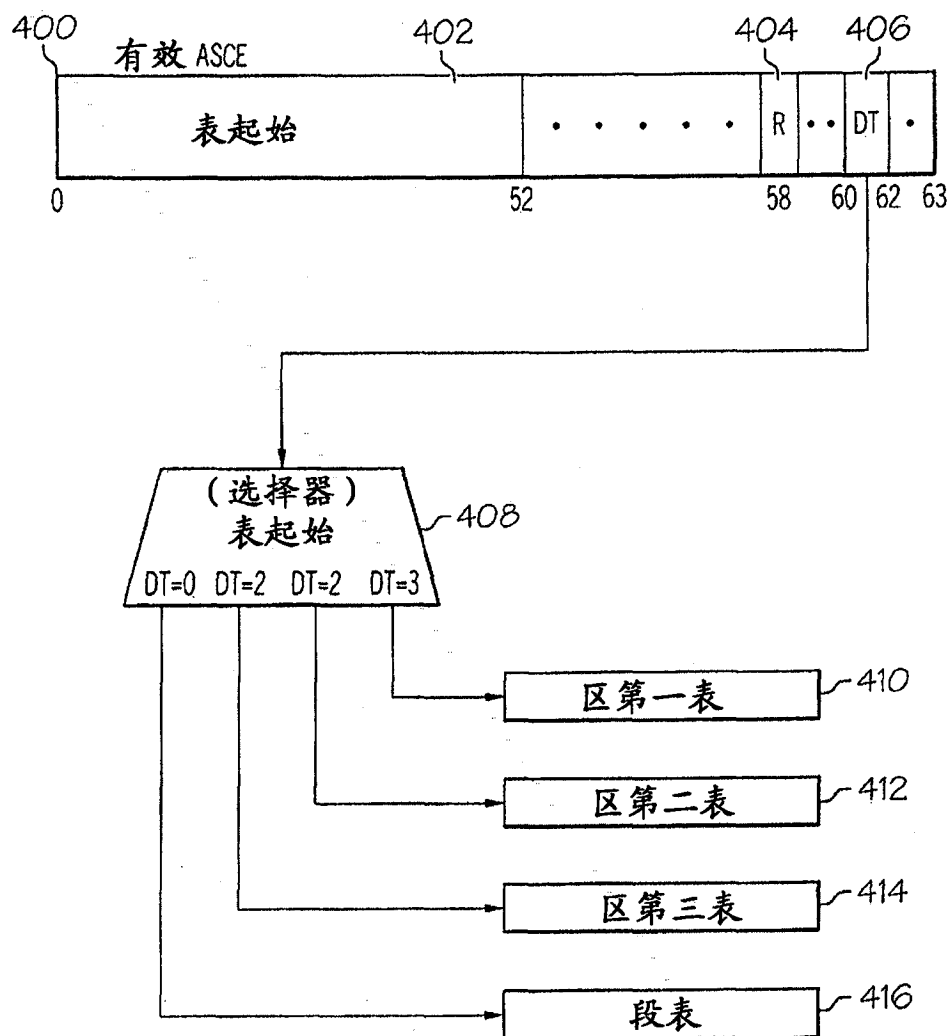


图 4(现有技术)

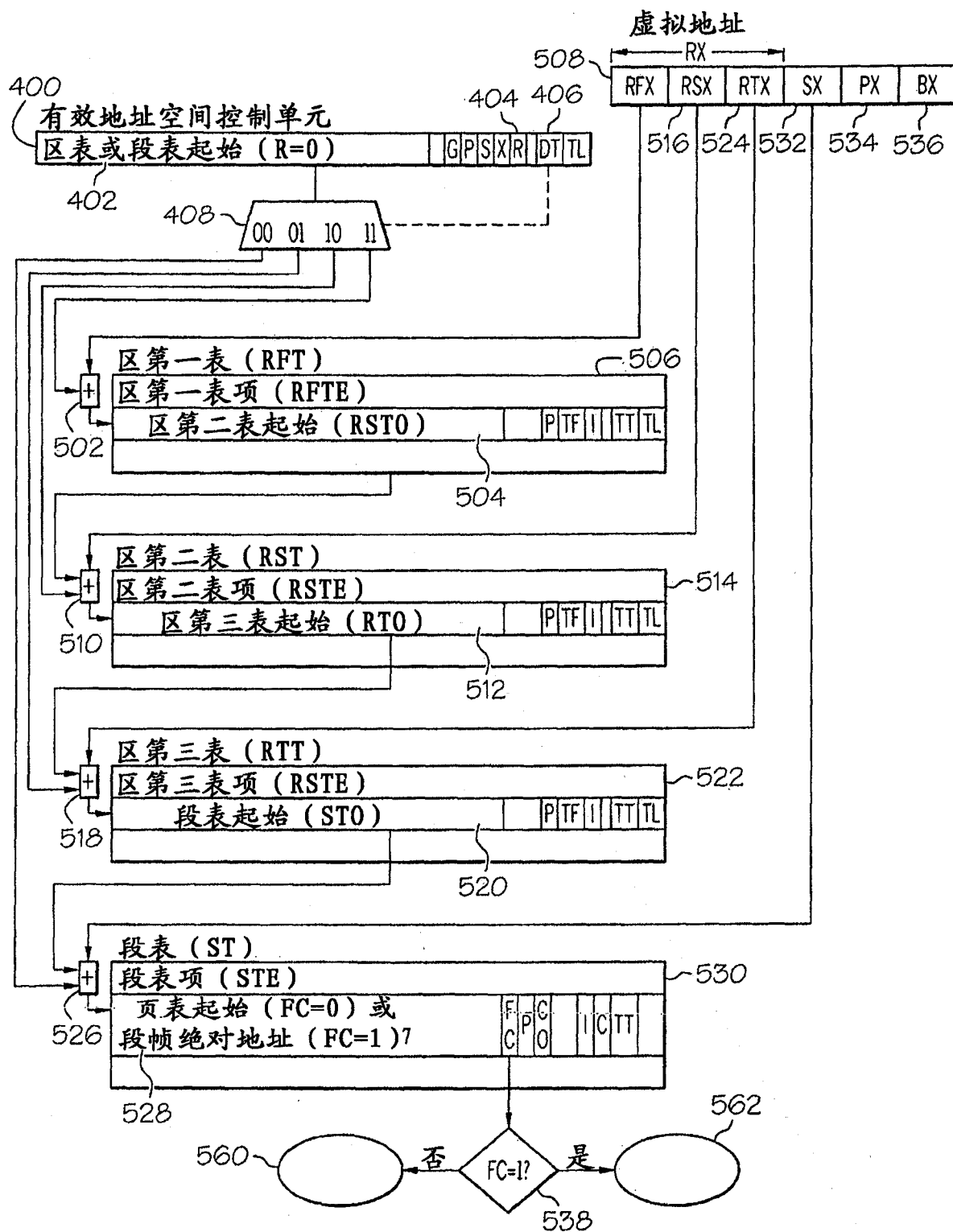


图 5A

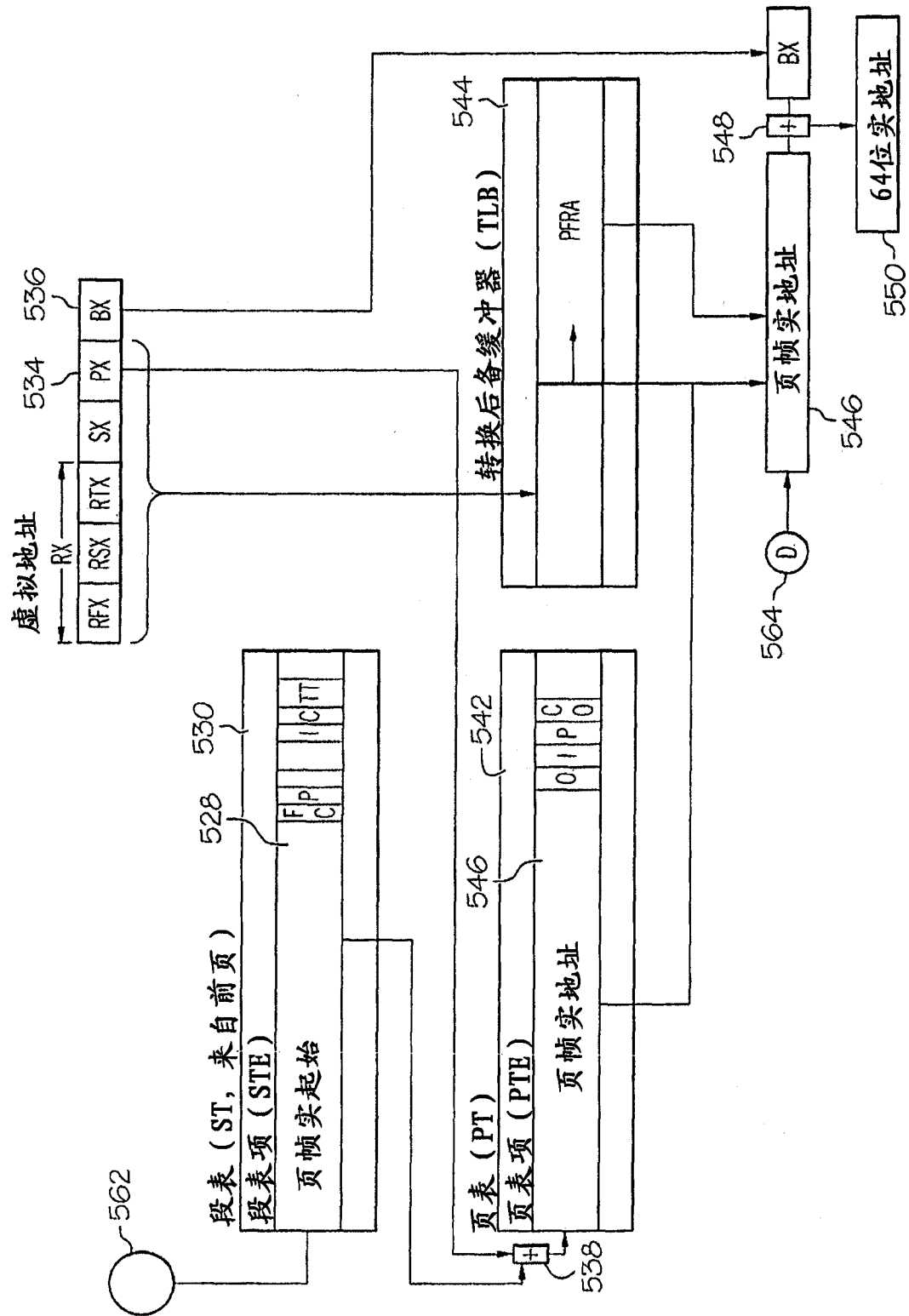


图 5B

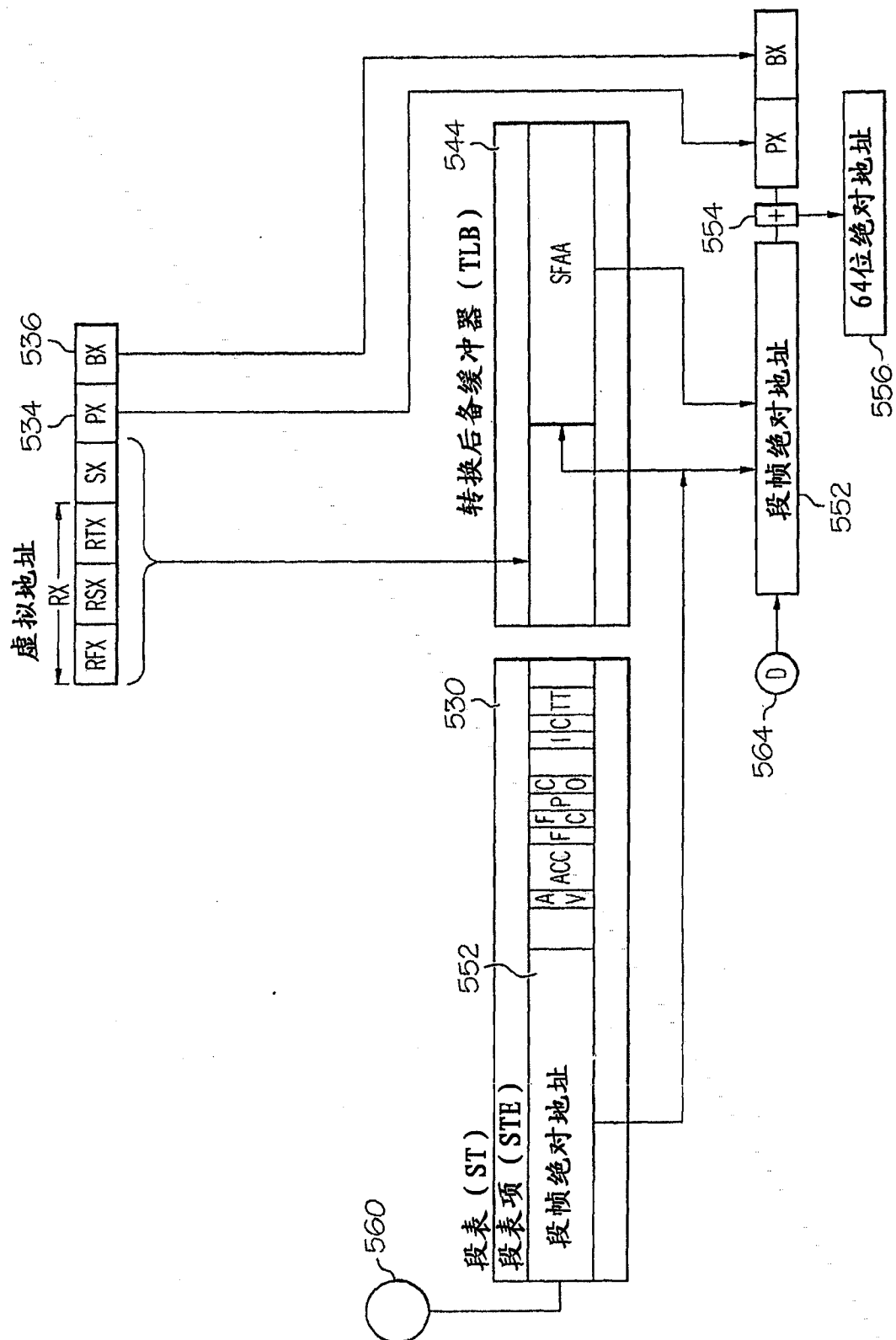


图 5C

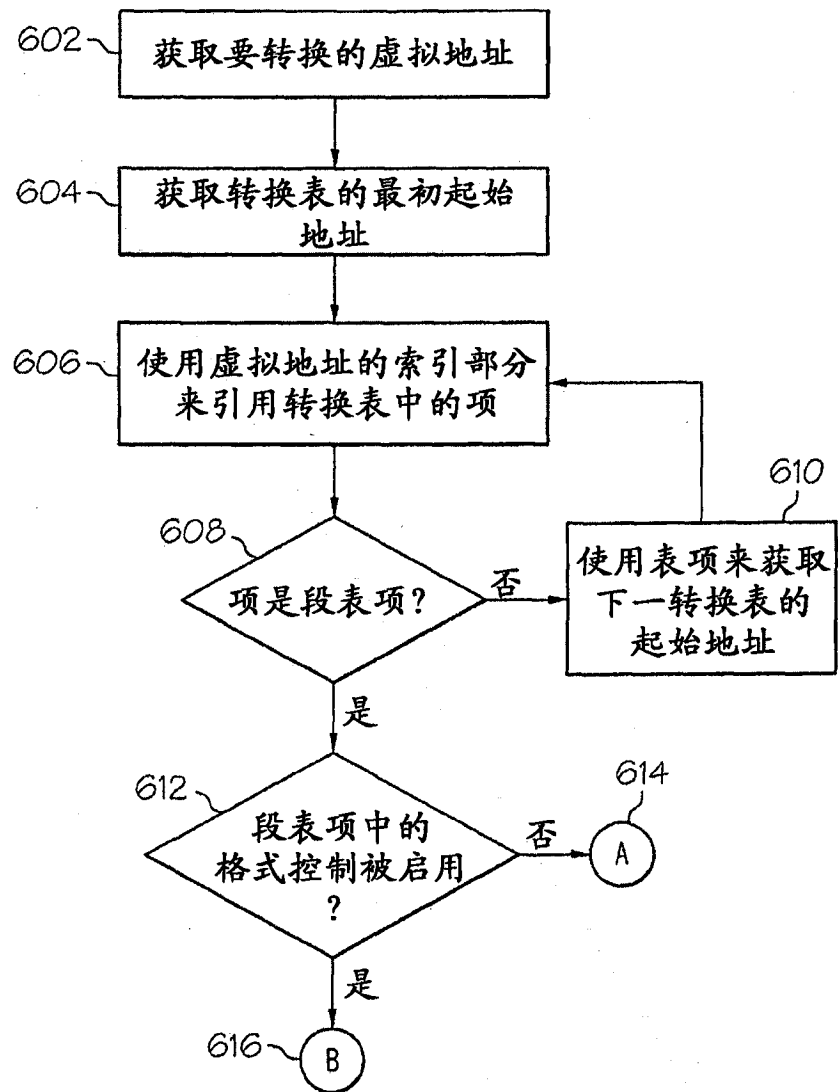


图 6

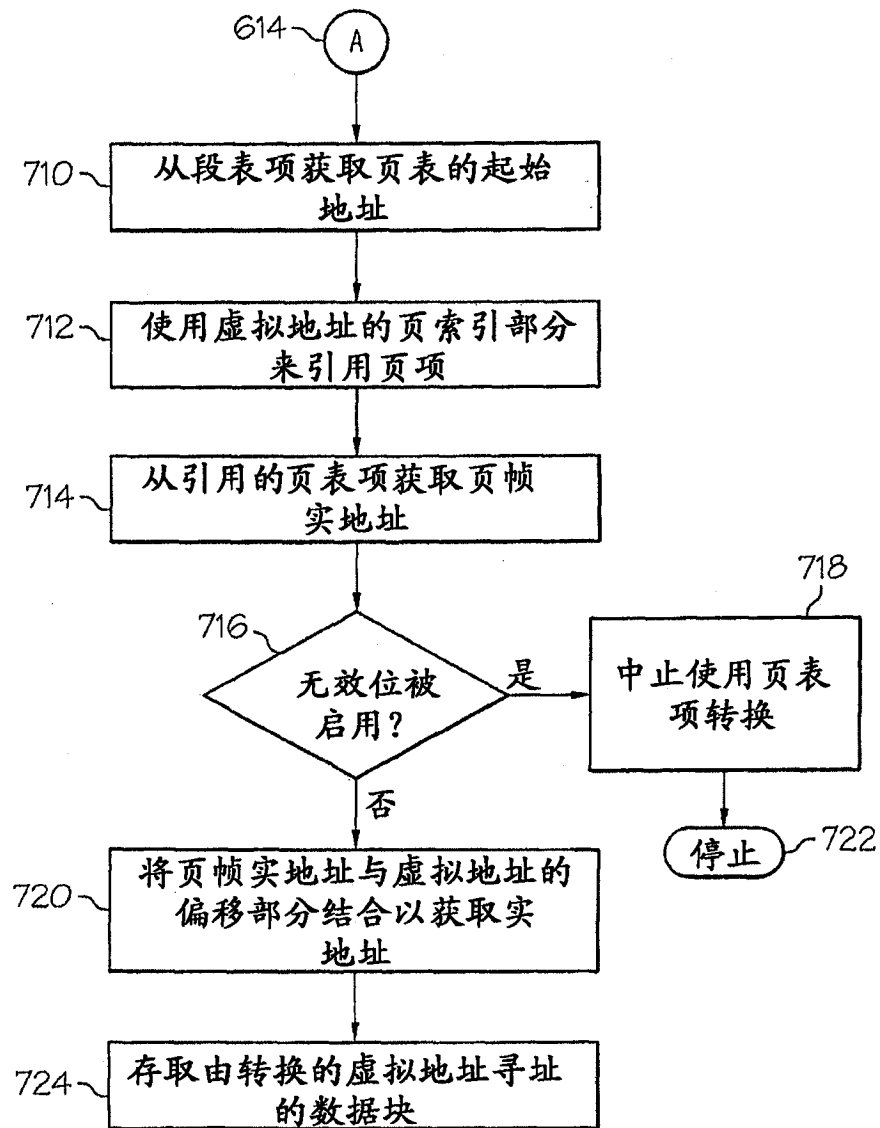


图 7(现有技术)

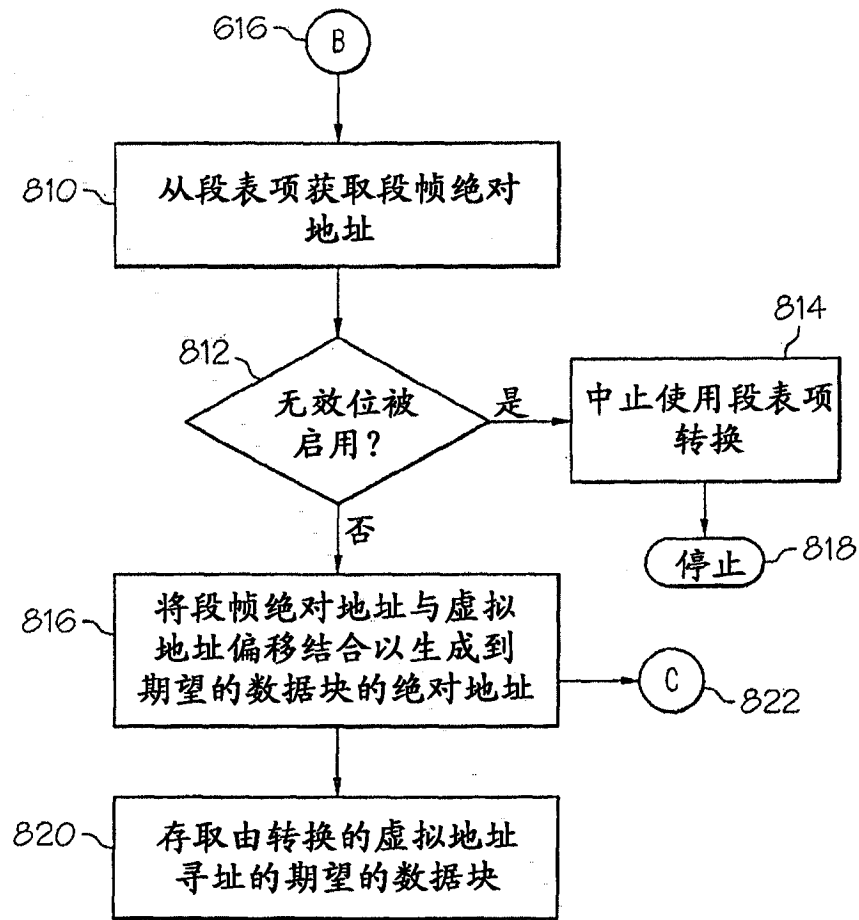


图 8

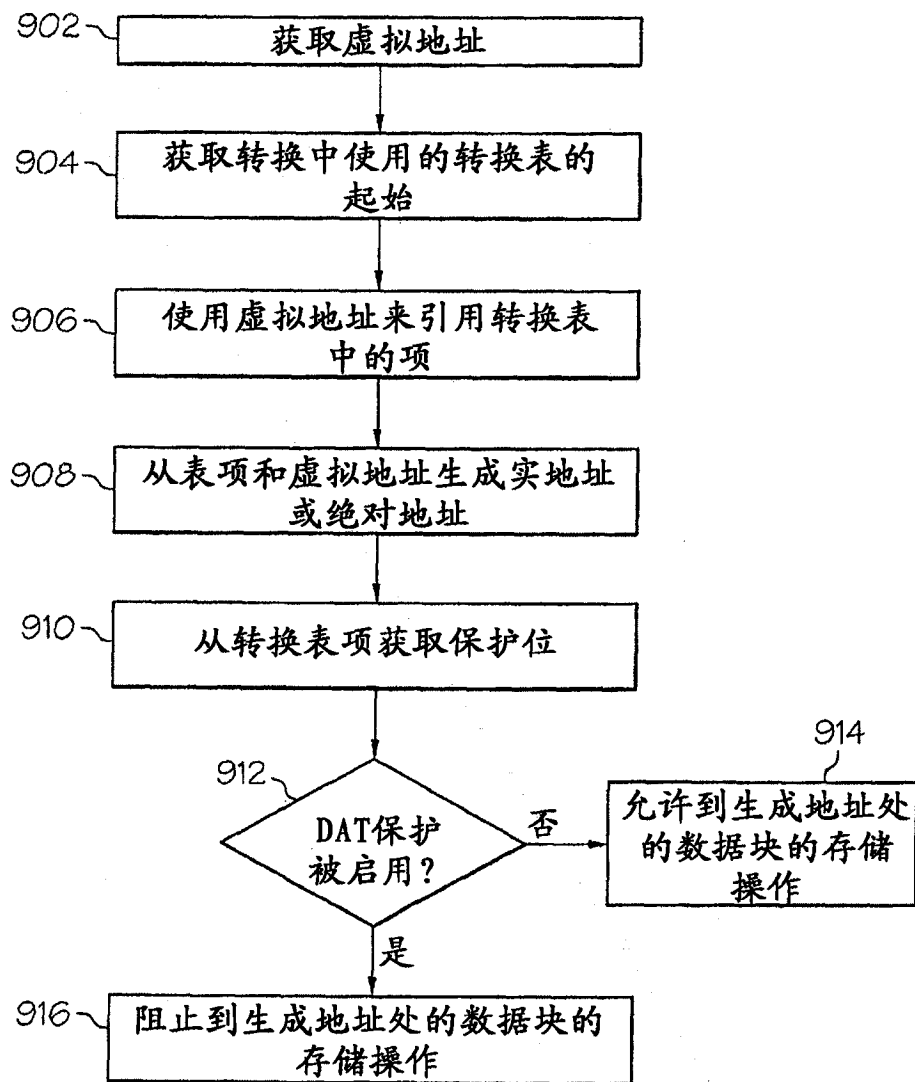


图 9

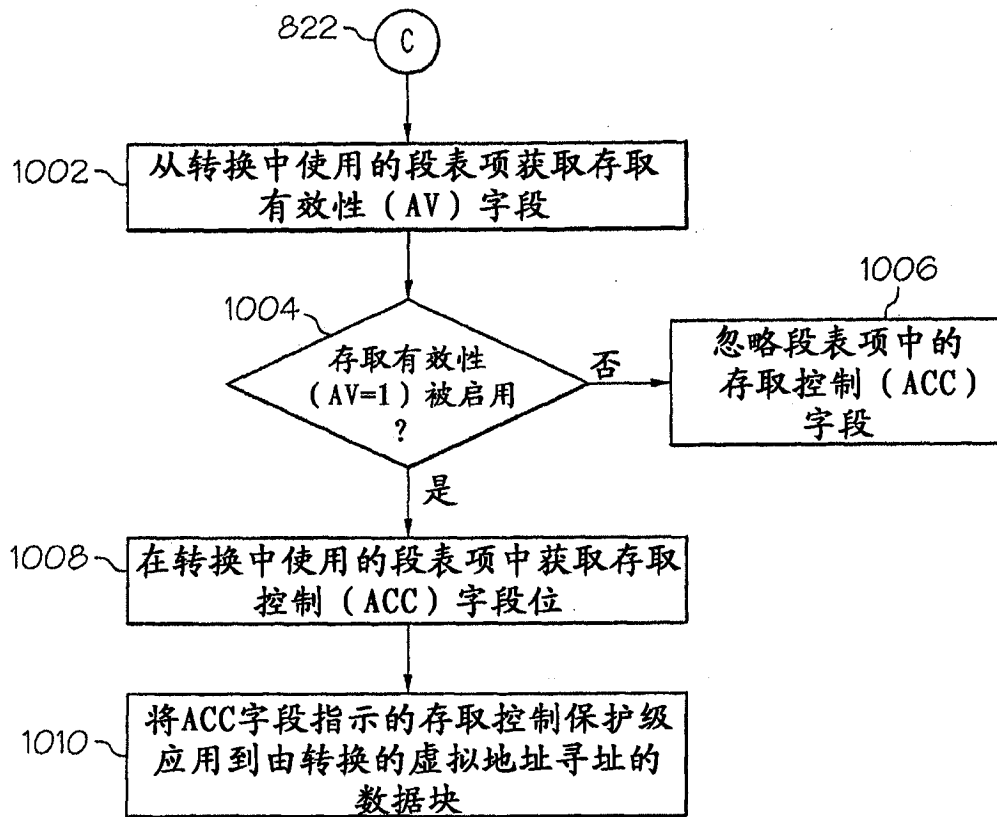


图 10

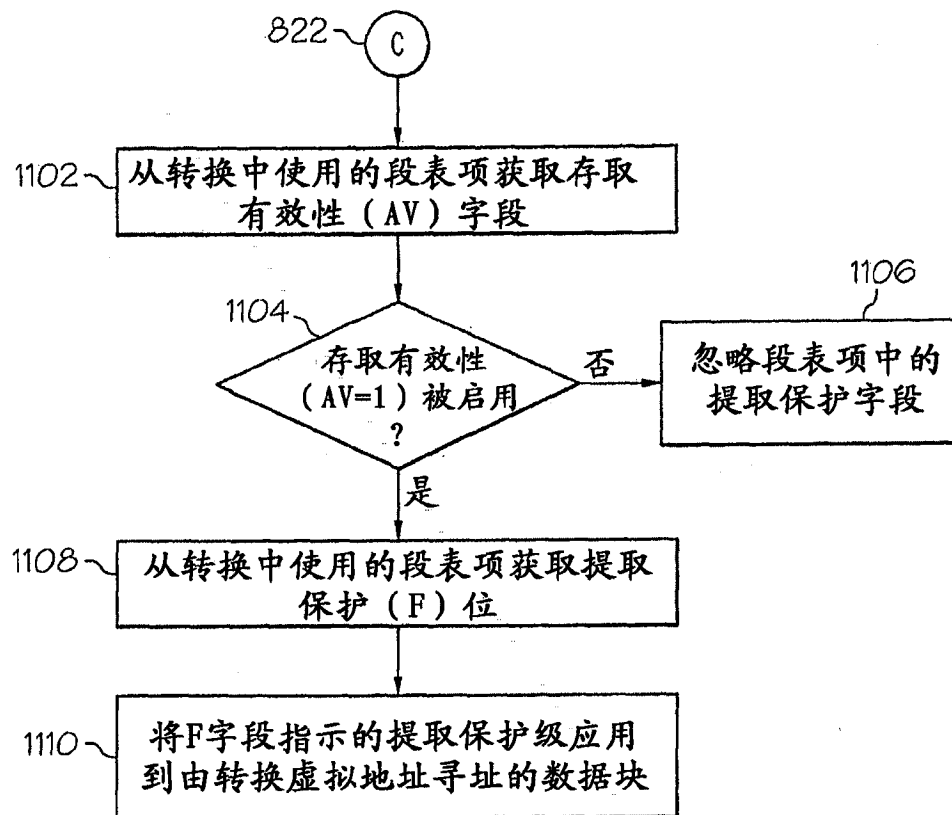


图 11

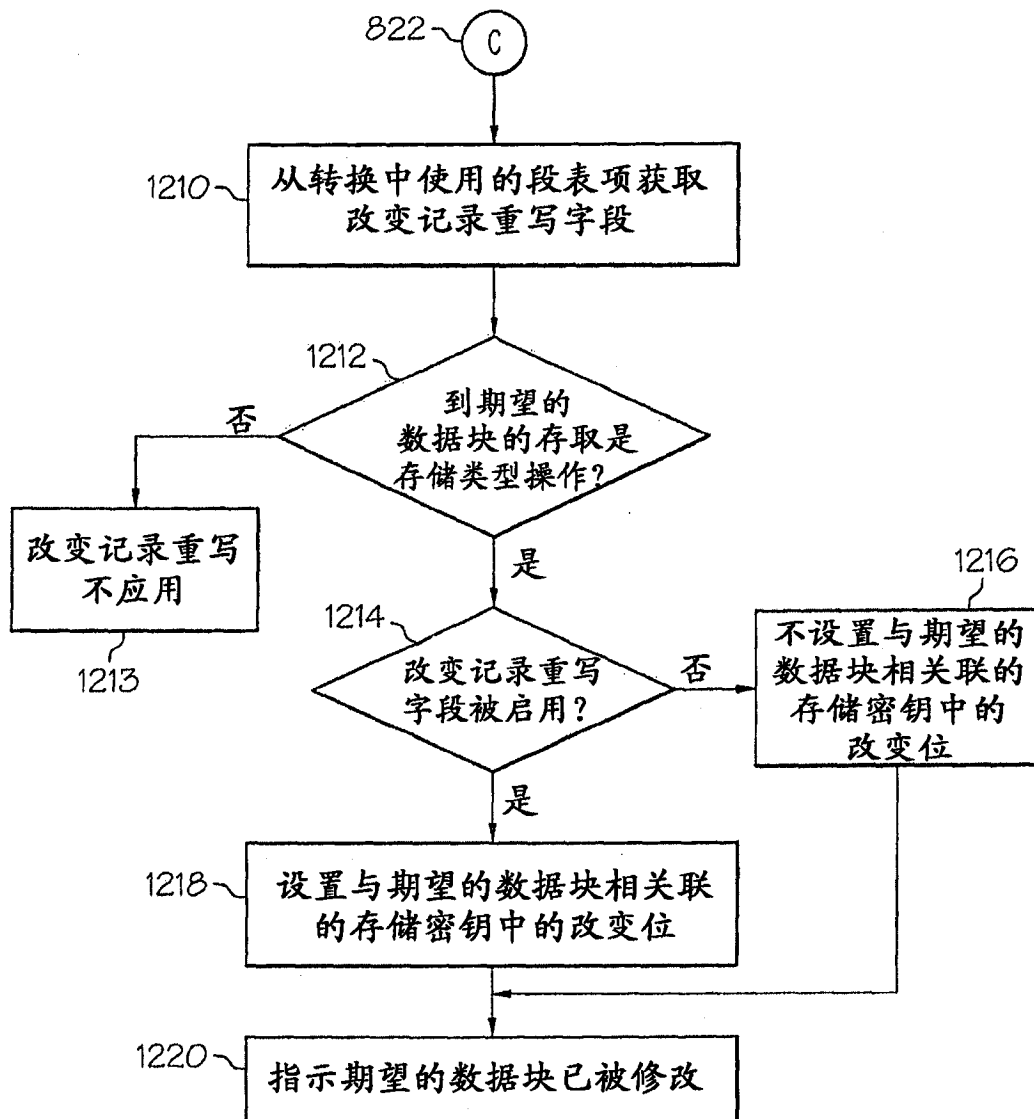


图 12