



(12) 发明专利申请

(10) 申请公布号 CN 101819518 A

(43) 申请公布日 2010.09.01

(21) 申请号 200910008371.3

(22) 申请日 2009.02.26

(71) 申请人 国际商业机器公司  
地址 美国纽约阿芒克

(72) 发明人 戈弋 侯锐 王华勇

(74) 专利代理机构 北京市柳沈律师事务所  
11105

代理人 邸万奎

(51) Int. Cl.

G06F 9/38(2006.01)

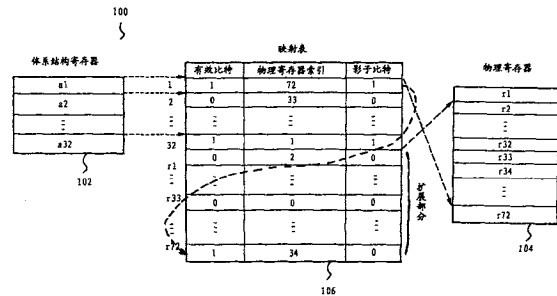
权利要求书 1 页 说明书 5 页 附图 3 页

(54) 发明名称

在事务内存中快速保存上下文的方法和装置

(57) 摘要

本发明提供了一种在事务内存中快速保存上下文的方法和装置,事务内存包括多个体系结构寄存器和数量多于体系结构寄存器的物理寄存器,该方法包括:建立映射表,其包括与体系结构寄存器一一对应的多个表项,每个表项包含映射到多个体系结构寄存器的多个第一物理寄存器的索引和影子比特;当在事务处理中检测到体系结构寄存器的更新、且影子比特为无效值时,将影子比特设为有效值,并利用第一物理寄存器的索引为体系结构寄存器设置影子寄存器;和向影子寄存器映射一第二物理寄存器,以保存更新处理产生的修改值,并由体系结构寄存器对应的第一物理寄存器保存更新处理前的原值。本发明的方法减少了上下文保存/恢复的开销,提高了程序的简易性。



1. 一种用于在事务内存中快速保存上下文的方法,所述事务内存包括多个体系结构寄存器以及数量多于所述体系结构寄存器的物理寄存器,所述方法包括:

建立映射表,其包括与多个体系结构寄存器一一对应的多个表项,每个表项包含映射到多个体系结构寄存器的多个第一物理寄存器的索引以及影子比特;

响应于在事务处理中检测到体系结构寄存器发生更新、并且其影子比特为无效值,将所述影子比特设为有效值,并利用所述第一物理寄存器的索引为该体系结构寄存器设置影子寄存器;以及

向所述影子寄存器映射第二物理寄存器,用来保存更新处理产生的修改值,并由该体系结构寄存器对应的第一物理寄存器保存更新处理前的原值。

2. 如权利要求 1 所述的方法,还包括步骤:响应于事务处理期间发生退回,重置影子比特,并清除影子寄存器及第二物理寄存器,以将体系结构寄存器恢复为原值。

3. 如权利要求 1 所述的方法,还包括步骤:响应于所述事务处理的完成,用影子寄存器的修改值替换相应的体系结构寄存器的原值,并将影子寄存器及第二物理寄存器释放为可用状态。

4. 如权利要求 1-3 中任一所述的方法,还包括步骤:响应于在事务处理中检测到所述体系结构寄存器发生更新、并且其影子比特为有效值,直接用新的修改值更新所述第二物理寄存器中的修改值。

5. 如权利要求 1-3 中任一所述的方法,其中所述表项还包括有效比特,用于将在事务处理中被使用的所述体系结构寄存器标记为有效。

6. 一种事务内存装置,用于快速保存上下文,所述事务内存装置包括多个体系结构寄存器以及数量多于所述体系结构寄存器的物理寄存器,并包括:

映射表,其包括与多个体系结构寄存器一一对应的多个表项,每个表项包含映射到所述多个体系结构寄存器的多个第一物理寄存器的索引以及影子比特;

用于响应于在事务处理中检测到体系结构寄存器发生更新、并且其影子比特为无效值,将所述影子比特设为有效值,并利用所述第一物理寄存器的索引为该体系结构寄存器设置影子寄存器的模块;以及

用于向所述影子寄存器映射第二物理寄存器,以保存更新处理产生的修改值,并用该体系结构寄存器对应的第一物理寄存器保存更新处理前的原值的模块。

7. 如权利要求 6 所述的事务内存装置,还包括:用于响应于事务处理期间发生退回,重置影子比特,并清除影子寄存器及第二物理寄存器,以将体系结构寄存器恢复为原值的模块。

8. 如权利要求 6 所述的事务内存装置,还包括:用于响应于所述事务处理的完成,用影子寄存器的修改值替换相应的体系结构寄存器的原值,并将影子寄存器及第二物理寄存器释放为可用状态的模块。

9. 如权利要求 6-8 中任一所述的事务内存装置,还包括:用于响应于在事务处理中检测到所述体系结构寄存器发生更新、并且其影子比特为有效值,直接用新的修改值更新所述第二物理寄存器的修改值的模块。

10. 如权利要求 6-8 中任一所述的事务内存装置,其中所述表项还包括有效比特,用于将在事务处理中被使用的所述体系结构寄存器标记为有效。

## 在事务内存中快速保存上下文的方法和装置

### 技术领域

[0001] 本发明涉及一种在处理器的事务内存中使用的方法和装置,更具体地涉及一种在处理器的事务内存中快速保存和恢复上下文的方法和装置。

### 背景技术

[0002] 为了更有效地使用多核资源,越来越多地应用采取了并程序。但是,用于数据共享管理的复杂编程模型使得难于开发并程序,因此建议由事务内存来提供一种简易的使用机制,以定义和管理并程序中的重要区段。

[0003] 在事务内存模型中,线程上下文应该保存于事务处理的开始阶段。如果在事务处理期间由于出现特定事件而发生退回(rollback),则恢复事务处理之前被保存的上下文。直到目前为止,所有的程序上下文都是根据负载/存储指令来保存的,包括体系结构寄存器、程序计数器、状态寄存器、堆栈指针等,它们原来是被保存在处理器的通用寄存器中。将所有这些保存在现代微体系结构的主存储器中会花费数千个时间周期,相同情况也发生在事务处理的退回阶段。

[0004] 寄存器重命名机制消除了 WAR(write-after-read) 和 WAW(write-after-write) 冲突,被广泛用于现代处理器的流水线处理模式。伴随某种映射方案,寄存器重命名机制可以动态地分配物理寄存器给体系结构寄存器。

[0005] 图 1 显示了体系结构寄存器和物理寄存器之间的映射的基本关系。

[0006] 当一个指令试图修改一个体系结构寄存器(例如 a1)时,为了避免与之前发布的访问该体系结构寄存器 a1 的指令产生冲突,重命名机制自动为新指令分配一个新的物理寄存器(r72),将该指令的修改值存储在新物理寄存器 r72 中。如果有多个指令访问同一个体系结构寄存器,则该体系结构寄存器会存在多个对应的物理寄存器。因此,物理寄存器的数目需要大于体系结构寄存器的数目。

[0007] 在现有技术中,包括被修改和未被修改的所有寄存器都需要在上下文保存和恢复过程中被写入存储器和从存储器读出,这可能要花费数千个时间周期。但是,大多数事务处理在整个过程中仅修改了几个体系结构寄存器,而绝大多数体系结构寄存器是未经修改而被保存和恢复的,这种方式导致浪费了大量的存储器资源。

### 发明内容

[0008] 本发明提出了一种新的方法,其利用对重命名寄存器机制的扩展,在事务处理期间仅仅保存/恢复被修改的体系结构寄存器,而不再对未被修改的寄存器进行保存/恢复,体系结构寄存器的原值将被保存在重命名寄存器中而不是存储器中,因此上下文保存/恢复的开销缩短为数十个周期,并且在事务处理开始阶段不再需要明显的上下文保存操作。

[0009] 根据本发明的一个方面,提供了一种用于在事务内存中快速保存上下文的方法,所述事务内存包括多个体系结构寄存器以及数量多于所述体系结构寄存器的物理寄存器,所述方法包括:建立映射表,其包括与多个体系结构寄存器一一对应的多个表项,每个表项

包含映射到多个体系结构寄存器的多个第一物理寄存器的索引以及影子比特；响应于在事务处理中检测到体系结构寄存器发生更新、并且其影子比特为无效值，将所述影子比特设为有效值，并利用所述第一物理寄存器的索引为该体系结构寄存器设置影子寄存器；以及向所述影子寄存器映射一第二物理寄存器，用来保存更新处理产生的修改值，并由该体系结构寄存器对应的第一物理寄存器保存更新处理前的原值。

[0010] 根据本发明的一个方面的方法还包括步骤：响应于事务处理期间发生退回，重置影子比特，并清除影子寄存器及第二物理寄存器，以将体系结构寄存器恢复为原值。

[0011] 根据本发明的一个方面的方法还包括步骤：响应于所述事务处理的完成，用影子寄存器的修改值替换相应的体系结构寄存器的原值，并将影子寄存器及第二物理寄存器释放为可用状态。

[0012] 根据本发明的一个方面的方法还包括步骤：响应于在事务处理中检测到所述体系结构寄存器发生更新、并且其影子比特为有效值，直接用新的修改值更新所述第二物理寄存器中的修改值。

[0013] 在根据本发明的一个方面的方法中，所述表项还包括有效比特，用于将在事务处理中被使用的所述体系结构寄存器标记为有效。

[0014] 根据本发明的另一个方面，提供一种事务内存装置，用于快速保存上下文，所述事务内存装置包括多个体系结构寄存器以及数量多于所述体系结构寄存器的物理寄存器，并包括：映射表，其包括与多个体系结构寄存器一一对应的多个表项，每个表项包含映射到所述多个体系结构寄存器的多个第一物理寄存器的索引以及影子比特；用于响应于在事务处理中检测到体系结构寄存器发生更新、并且其影子比特为无效值，将所述影子比特设为有效值，并利用所述第一物理寄存器的索引为该体系结构寄存器设置影子寄存器的模块；以及用于向所述影子寄存器映射一第二物理寄存器，以保存更新处理产生的修改值，并用该体系结构寄存器对应的第一物理寄存器保存更新处理前的原值的模块。

[0015] 根据本发明的前述另一个方面的事务内存装置还包括：用于响应于事务处理期间发生退回，重置影子比特，并清除影子寄存器及第二物理寄存器，以将体系结构寄存器恢复为原值的模块。

[0016] 根据本发明的前述另一个方面的事务内存装置还包括：用于响应于所述事务处理的完成，用影子寄存器的修改值替换相应的体系结构寄存器的原值，并将影子寄存器及第二物理寄存器释放为可用状态的模块。

[0017] 根据本发明的前述另一个方面的事务内存装置还包括：用于响应于在事务处理中检测到所述体系结构寄存器发生更新、并且其影子比特为有效值，直接用新的修改值更新所述第二物理寄存器的修改值的模块。

[0018] 根据本发明的前述另一个方面的事务内存装置，所述表项还包括有效比特，用于将在事务处理中被使用的所述体系结构寄存器标记为有效。

[0019] 本发明的优点在于：当发生寄存器重命名时，仅仅将被改变的上下文保存到重命名的寄存器中，以此来减少缓冲的需要和上下文保存 / 恢复的开销。本发明的方法由硬件控制，无需软件介入，提高了程序的简易性。

## 附图说明

[0020] 通过结合附图对本发明各个实施例的详细描述,本领域的技术人员可以更好地理解本发明的上述和其它方面、特点和优点,其中在所有附图中使用相同或相似的附图标记来表示相同或者相似的部件,在附图中:

[0021] 图 1 示出了体系结构寄存器和物理寄存器之间的映射的基本关系。

[0022] 图 2 示出了根据本发明实施例的方法的工作原理的视图。

[0023] 图 3(a) 是示出了根据本发明实施例用于在事务内存中快速保存上下文的方法的流程图。

[0024] 图 3(b) 示出了根据本发明实施例的在事务内存中快速保存上下文后进行恢复或设置的方法的流程图。

## 具体实施方式

[0025] 下面将结合附图详细描述本发明的具体实施例。如果考虑到对某些相关现有技术详细描述会混淆本发明的要点,则不会在这里提供其详细描述。

[0026] 如图 2 所示,根据本发明实施例的事务内存 100 包括多个体系结构寄存器 102 和多个物理寄存器 104,物理寄存器 104 的数量多于体系结构寄存器 102 的数量,例如图 2 中的体系结构寄存器 102 包括  $a_1, a_2, \dots, a_{32}$ ,物理寄存器 104 包括  $r_1, r_2, r_3, \dots, r_{72}$ 。

[0027] 事务内存 100 还包括映射表 106,该映射表 106 按照如下方式构成:映射表在从上到下方向由多个表项组成,每个表项表示体系结构寄存器 102 中的一个,例如表项 1 表示体系结构寄存器  $a_1$ ,表项 2 表示体系结构寄存器  $a_2, \dots$ ,表项 32 表示体系结构寄存器  $a_{32}$ 。

[0028] 映射表在从左到右方向由三列构成,第一列是有效比特,第二列是物理寄存器索引,第三列是影子比特。也就是说,每个表项包括三个部分,即有效比特、物理寄存器索引、和影子比特。可以将事务处理前已被使用的体系结构寄存器 102 的对应表项中的有效比特设置为有效值,例如 1,表示其在该事务处理前已被使用,而如果有效比特的值是无效值,例如 0,则表示其还没有在事务处理中被使用。物理寄存器索引用来表示在事务处理中被映射到体系结构寄存器 102 的物理寄存器(第一物理寄存器)104。影子比特表示在事务处理中该体系结构寄存器 102 的值被改变,并且为该体系结构寄存器 102 创建了重命名寄存器——影子寄存器,并为新创建的影子寄存器(例如  $r_{72}$ )映射新的物理寄存器(第二物理寄存器),例如用物理寄存器索引(标号)34 表示,用来代替原始的体系结构寄存器来存储改变值。

[0029] 在映射表 106 的下部,包括多个增加的表项,这些表项由为体系结构寄存器 102 所创建的影子寄存器构成,作为该体系结构寄存器 102 的重命名寄存器使用,例如影子寄存器  $r_1, r_2, \dots, r_{33}, \dots, r_{72}$ 。代表影子寄存器的表项与代表体系结构寄存器 102 的表项的构成方式相同。

[0030] 根据本发明一个实施例的示例,表项 1 代表体系结构寄存器  $a_1$ ,其中的有效比特是 1,表示该体系结构寄存器  $a_1$  在事务处理前已被使用,物理寄存器索引是 72,表示在该事务处理前被映射到体系结构寄存器  $a_1$  的物理寄存器(第一物理寄存器)是  $r_{72}$ 。如果影子比特是 1,表示在该事务处理中该体系结构寄存器  $a_1$  的值已经被改变,即在该事务处理中存在至少一个指令访问同一个体系结构寄存器  $a_1$ ,由此产生了寄存器更新操作。这时,在映

射表中为体系结构寄存器 a1 创建新的表项 r72, 来代表体系结构寄存器 a1 的重命名寄存器, 即影子寄存器, 并为影子寄存器 r72 映射新的物理寄存器 (第二物理寄存器), 例如该新物理寄存器的索引是 34, 用来代表原始的体系结构寄存器来存储该事务处理中的改变值。

[0031] 由于代表体系结构寄存器 a1 的表项 1 中的影子比特的值是 1, 并且物理寄存器索引在这个表项中为 72, 所以利用影子寄存器 r72 代表体系结构寄存器 a1 来记录体系结构寄存器 a1 的重命名状态, 直到在事务处理期间发生退回或由于事务处理完成而重置影子比特时为止。体系结构寄存器 a1 的表项内容在事务处理过程期间不会改变。从寄存器角度来看, 影子寄存器表项 r72 不仅将原始体系结构寄存器 a1 的值保存在寄存器 (第一物理寄存器 r72) 中, 其也记录了在事务处理中被改变的寄存器值 (利用第二物理寄存器, 例如 r34)。

[0032] 当在事务处理期间由于特定事件的出现而发生了退回时, 重置影子比特的值, 即将它们的值重新设置为零, 并清除影子寄存器及其对应的第二物理寄存器, 以将体系结构寄存器 102 恢复为事务处理前的原值。

[0033] 或者, 当事务处理完成时, 将各个影子寄存器对应的第二物理寄存器中保存的修改值复制到相应的体系结构寄存器 102 中, 以替换其中保存的原值, 并将影子寄存器及对应的第二物理寄存器释放为可用状态。

[0034] 需要注意的是, 体系结构寄存器 102 的有效比特不对本发明的技术范围构成任何限制, 本发明的实施例也可以不包括任何的有效比特。

[0035] 下面结合图 3(a) 和图 3(b) 来描述根据本发明实施例的用于在事务内存中快速保存和恢复上下文的方法。

[0036] 图 3(a) 是示出根据本发明实施例的用于在事务内存中快速保存上下文的方法的流程图, 而图 3(b) 示出了根据本发明实施例的在事务内存中保存了上下文后进行恢复或设置的方法的流程图。

[0037] 如上所述, 根据本发明实施例的事务内存包括多个体系结构寄存器 102 以及数量多于体系结构寄存器 102 的物理寄存器 104, 并且还包含映射表 106。该映射表 106 包括与多个体系结构寄存器 102 一一对应的多个表项, 每个表项包含映射到该多个体系结构寄存器 102 的多个 (第一) 物理寄存器 104 的索引以及影子比特。

[0038] 在正常状态下, 只有体系结构寄存器 102 在事务处理中被使用, 物理寄存器的表项和影子比特部分处于待用状态。

[0039] 参照图 3(a), 当程序开始事务处理后, 步骤进入 S301。在步骤 S301 中, 执行事务处理指令, 并在步骤 S302 判断在事务处理中, 体系结构寄存器 102 是否发生更新。当在步骤 S302 判断在该事务处理中体系结构寄存器 102 不存在更新时, 返回步骤 S301, 并保持正常使用的寄存器状态, 不发生上下文保存操作。在步骤 S301 中, 可以设定一个事务内存标志来指示该事务处理的状态 (可选)。体系结构寄存器 102 在事务处理中发生更新意味着事务内存处理中存在至少一个指令对同一个体系结构寄存器 102 进行了访问, 产生了访问更新。

[0040] 当在步骤 S302 中判断在事务处理中体系结构寄存器 102 (例如 a1) 发生更新时, 前进到步骤 S303。在步骤 S303, 判断映射表 106 中代表该体系结构寄存器 102 的表项中的影子比特是否为零。如果在步骤 S303 中判断代表该体系结构寄存器 102 的表项中的影子比特为零, 意味着这是在事务处理中该体系结构寄存器 102 的值的的第一次改变, 于是处理

前进到步骤 S304, 否则处理前进到步骤 S305。

[0041] 在步骤 S304 中, 将该影子比特设置为有效值, 例如设置为 1, 利用代表该体系结构寄存器 102(例如 a1) 的表项中的物理寄存器索引(例如是 72, 代表与体系结构寄存器 a1 对应的第一物理寄存器), 创建该体系结构寄存器 102(例如 a1) 的影子寄存器(例如 r72), 并向该影子寄存器(例如 r72) 映射一个新的物理寄存器(第二物理寄存器, 例如是 r34, 用其索引 34 表示)。在该新的物理寄存器(r34) 中保存更新处理的修改值, 并在该体系结构寄存器 102(例如 a1) 对应的原始物理寄存器(第一物理寄存器) 中保存更新处理前的原值。

[0042] 如果在步骤 S303 中判断代表该体系结构寄存器 102(a1) 的表项中的影子比特不为零, 即该影子比特已经为有效值, 则意味着该体系结构寄存器 102(a1) 的值在事务处理中不是第一次改变, 而且已经存在与该体系结构寄存器 102(a1) 对应的影子寄存器, 这时, 在步骤 S305 中, 只需将该影子寄存器所映射的(第二)物理寄存器中的值更新为新的改变值即可。

[0043] 由此, 以上参照图 3(a) 描述了用于在事务内存中快速保存上下文的方法, 下面参照图 3(b) 描述在事务内存中保存了上下文后进行恢复或设置的方法。

[0044] 处理从步骤 S304 或者 S305 前进到步骤 S306。在步骤 S306 中, 判断在事务处理中是否由于出现特定事件而发生退回。如果在步骤 S306 中判断在事务处理中发生了退回, 则处理前进到步骤 S307, 否则处理前进到步骤 S308。

[0045] 在步骤 S307 中, 响应于在事务处理期间发生了退回, 重置影子比特的值, 即将它们的值重新设置为零, 并清除影子寄存器及其对应的第二物理寄存器, 以将体系结构寄存器 102 恢复为事务处理前的原值。然后, 事务处理结束。

[0046] 在步骤 S308 中, 判断事务处理是否完成。如果在步骤 S308 中判断事务处理已经完成, 则处理前进到步骤 S309, 否则处理返回到步骤 S306。

[0047] 在步骤 S309 中, 响应于所述事务处理的完成, 将各个影子寄存器对应的第二物理寄存器中保存的修改值复制到相应的体系结构寄存器 102 中, 以替换其中保存的原值, 并将影子寄存器及对应的第二物理寄存器释放为可用状态。然后, 事务处理结束。

[0048] 根据本发明实施例的以上各个步骤的执行顺序不对本发明的技术范围构成限制, 例如上述步骤 S306 与步骤 S308 的执行顺序可以互换, 且所有步骤均可以并行的顺序来执行。

[0049] 本发明的优点在于: 当发生寄存器重命名时, 仅仅将被改变的上下文保存到重命名寄存器, 以此来减少缓冲器的需求和上下文保存/恢复的开销。该机制由硬件控制, 无需软件介入, 因此提高了程序的简易性。

[0050] 虽然上面已经结合附图示出并描述了本发明的一些实施例, 但是本领域的技术人员应当理解, 在不偏离本发明的原则和精神的情况下, 可以对这些实施例进行变化和修改, 但它们仍然落在本发明的权利要求及其等价物的范围之内。

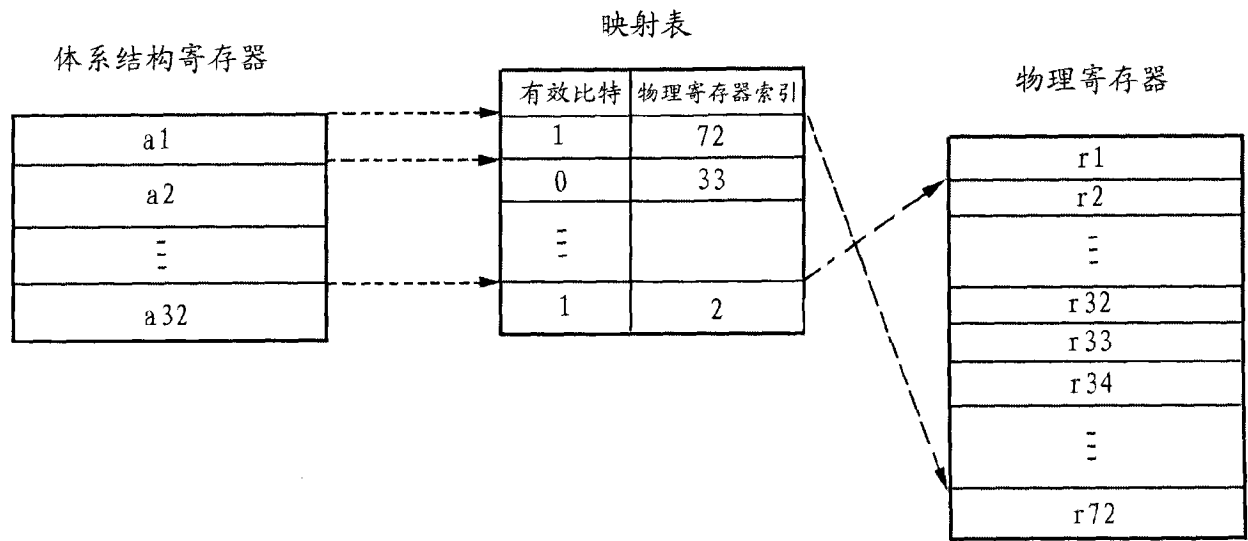


图 1



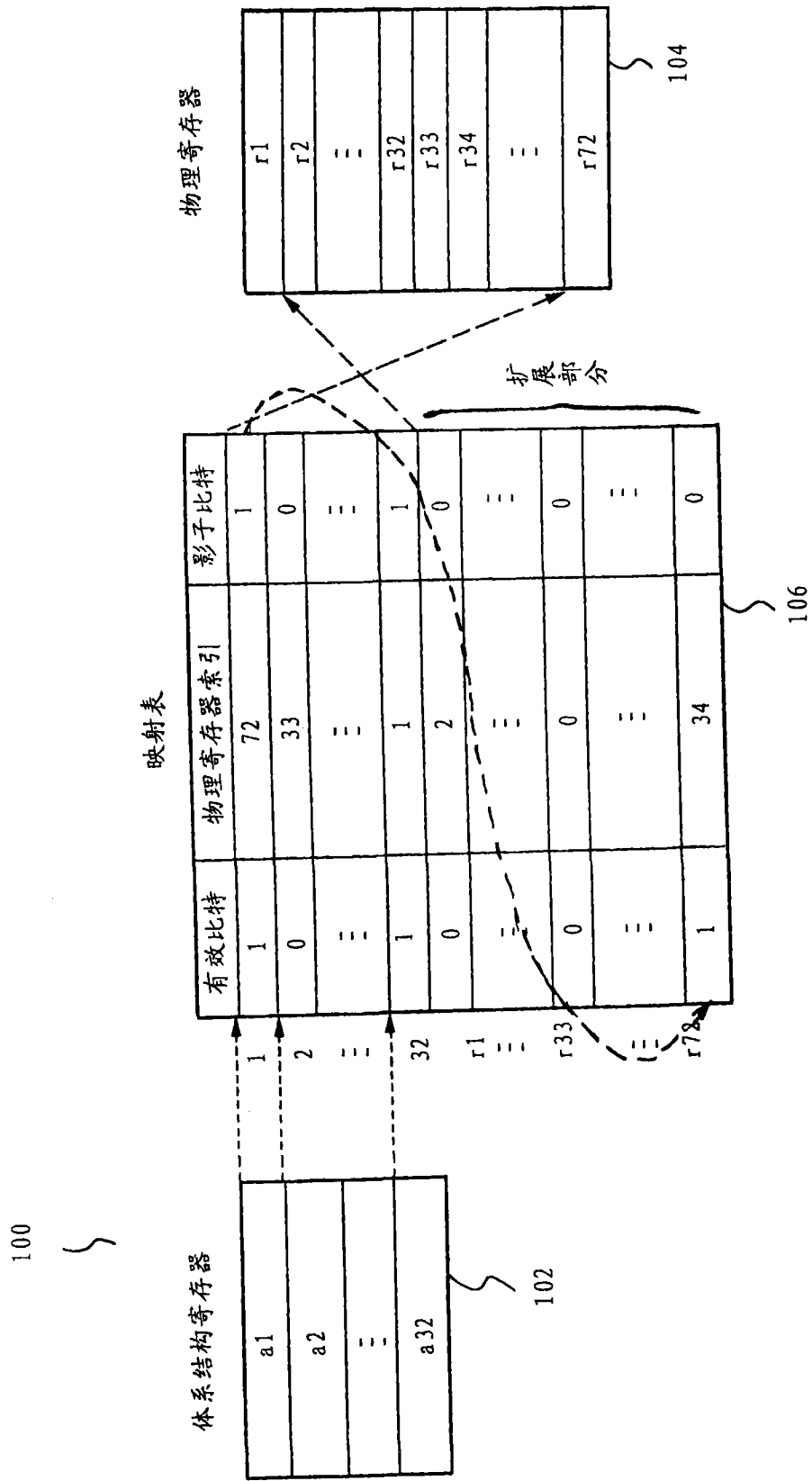


图 2

