

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06F 11/20 (2006.01)



[12] 发明专利申请公布说明书

[21] 申请号 200480043678.2

[43] 公开日 2007 年 7 月 4 日

[11] 公开号 CN 1993681A

[22] 申请日 2004.7.27

[21] 申请号 200480043678.2

[86] 国际申请 PCT/EP2004/008402 2004.7.27

[87] 国际公布 WO2006/010374 德 2006.2.2

[85] 进入国家阶段日期 2007.1.26

[71] 申请人 西门子公司

地址 德国慕尼黑

[72] 发明人 F·哈纳 P·佩勒斯卡

[74] 专利代理机构 中国专利代理(香港)有限公司
代理人 刘春元 魏 军

权利要求书 2 页 说明书 7 页 附图 2 页

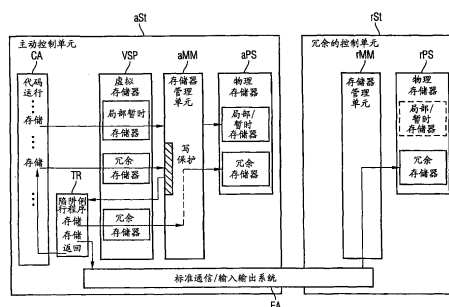
[54] 发明名称

对保持冗余的存储单元中的一致的存储内容进行保护的方法和装置

[57] 摘要

本发明首先涉及一种用于在电信或者数据处理系统内部对保持冗余的存储单元中的一致的存储内容进行保护的方法,所述电信或者数据处理系统具有至少一个主动控制单元(aSt)和至少一个冗余的被动控制单元(rSt),所述至少一个主动控制单元和至少一个冗余的被动控制单元分别用至少一个存储单元构成。这里执行以下的步骤:(a)如果对所述主动控制单元(aSt)的存储单元中的存储区(VSP)写访问,所述存储区的存储内容在存储器写访问的情况下应当被复制到至少一个冗余的被动控制单元(rSt)的存储单元中,则调用反射例程序(TR);(b)在执行所述反射例程序(TR)期间把要写的存储内容存放在所述至少一个冗余的被动控制单元的存储单元(rPS)中的存储区中;(c)在所述反射例程序中

重新在另一虚拟存储区上进行已经导致调用反射例程序的对所述主动控制单元的写访问,所述虚拟存储区被映射到与其写已经导致调用所述反射例程序的存储区相同的物理地址上。



1. 用于在电信或者数据处理系统内部对保持冗余的存储单元中的一致存储内容进行保护的方法，所述电信或者数据处理系统具有至少一个主动控制单元（aSt）和至少一个冗余的被动控制单元（rSt），所述至少一个主动控制单元和至少一个冗余的被动控制单元分别用至少一个存储单元构成，其中执行以下的步骤：

-如果对所述主动控制单元（aSt）的存储单元中的一个存储区（VSP）进行写访问，所述存储区的存储内容在存储器写访问的情况下应当被复制到至少一个冗余的被动控制单元（rSt）的存储单元（rPS）中，则调用反射例行程序（TR），

-所述反射例行程序（TR）的调用通过适当地设置存储器管理单元（aMM）来引起，而不是通过从运行的程序（CA）的显性调用引起，

-在执行所述反射例行程序（TR）期间，把要写的存储内容复制到所述至少一个冗余的被动控制单元的存储单元（rPS）中的存储区中，

-在所述反射例行程序中重新在另一虚拟存储区上进行已经导致调用反射例行程序的对所述主动控制单元的写访问，所述另一虚拟存储区被映射到与其写已经导致调用所述反射例行程序的存储区相同的物理地址上。

2. 如权利要求1所述的方法，其特征在于，所述虚拟存储区在存储器写访问之前被表明是受写保护的。

3. 如以上权利要求中任一项所述的方法，其特征在于其它步骤：

-在更换或者开始使用冗余的控制单元（rSt）之后，通过复制例行程序（KR）把主动控制单元（aSt）的存储单元的存储内容一次完全地复制到所述至少一个冗余的被动控制单元（rSt）的存储单元中的存储区中，

-随着每个复制过程提高全局变量中的存储器地址，

-在执行所述反射例行程序（TR）时把要写的存储区的存储器地址与全局变量中的存储器地址相比较，

-如果在比较时存储器地址一致，则延迟执行反射例行程序（TR）。

4.如以上权利要求中任一项所述的方法，其特征在于，如果在重新比较时存储器地址不相同，则继续进行反射例程序（TR）。

5.如以上权利要求中任一项所述的方法，其特征在于，把反射例程序（TR）实施为微处理器代码中的功能序列。

6.如以上权利要求中任一项所述的方法，其特征在于，反射例程序（TR）由协处理器执行。

7.用于执行如以上权利要求中任一项所述方法的控制单元（aSt），具有虚拟存储单元（VSP）和物理存储单元（aPS），其存储内容在存储器写访问的情况下可以被复制到至少另一冗余的被动控制单元（rSt）的存储单元中，

-用于调用反射例程序（TR）的装置，用以在对所述存储区（VSP）的存储器写访问的情况下使至少一个冗余的被动控制单元（rSt）的存储单元中的存储区中的存储内容加倍，和

-用于执行已经导致调用反射例程序（TR）的对另一虚拟存储区的写访问的装置，所述另一虚拟存储区被映射到与其写已经导致调用所述反射例程序的存储区相同的物理地址上。

对保持冗余的存储单元中的 一致的存储内容进行保护的方法和装置

本发明涉及一种用于在电信或者数据处理系统内部对保持冗余的存储单元中的一致的存储内容进行保护的方法和装置。

每年出现由 CPU、芯片组、主存储器 and 外围器件所组成的典型处理器板的硬件故障的概率处于一位数的百分比范围中。电信系统或者也为所谓的数据中心一般由多个这样的板构成。诸如交换站之类的系统典型地由高达几百个这样的处理器板构成，由此相对于年时间间隔而言任意硬件组件失灵的的概率变得非常高。尤其是对电信系统、但是也越来越多地对数据中心提出高度的系统可用性的要求，例如 > 99.999% 的可用性或者说一年中几分钟的不可使用性的要求。因为通常在硬件故障的情况下在系统层面上更换处理器板并且重新建立服务一般持续数 10 分钟直至几小时，所以对于硬件故障情况必须采取相应的防护措施以便能够满足对系统可用性的要求。

原则上对这种可用性的要求只能够通过冗余的系统组件来满足。在此存在通过软件措施控制冗余的途径，这里典型地使用所谓的中间件，或者存在以下途径，即在硬件层面上压缩（kapseln）冗余，使得所述冗余对软件是透明的。由软件措施所控制的冗余的基本缺点是，在这样一种系统中只能够采用为该特别的冗余方案所开发的（应用）软件。这显著地限制了可用的（应用）软件的范围。此外一般可以耗费大地开放和测试软件冗余原理的应用软件。

尤其是常规的交换系统的控制软件是非常大规模的（高达几百万代码行）。因此只能够以非常高的耗费和风险匹配这样的软件。

因此迄今对于这样的交换系统需要一种专门开发的硬件。于是所述硬件支持所要求的冗余，使得在单个硬件模块失灵时具有足够的信息用于在冗余的单元上无间断地继续操作。为此往往采用一种自动的、总是在后台一起运行的复制装置，所述复制装置把主动控制单元的工作存储器的可能已定义的部分连续地复制到冗余单元的存储器上（常常被称为“反射存储”）。

直接的结果是：交换设备用的控制要求非常高昂的开发耗费并

且其革新周期大多非常长。在微处理器（例如快速的、往往也是专有的且受保护的总线系统和具有回写（write-back）功能的内部高速缓冲存储器）的开发中的当前优化也大大使实施反射存储日益变得困难。

本发明的任务是克服上述的缺点。

该任务在方法方面和在装置方面通过独立权利要求的特征来解决。本发明的改进在从属权利要求中得以说明。

本发明的一个基本方面在于一种用于在电信或者数据处理系统内部对保持冗余的存储单元中的一致的存储内容进行保护的方法，所述电信或者数据处理系统具有至少一个主动控制单元和至少一个冗余的被动控制单元，所述至少一个主动控制单元和至少一个冗余的被动控制单元分别用至少一个存储单元构成，其中执行以下的步骤：

- 如果对所述主动控制单元的存储单元中的存储区进行写访问，其中所述存储区的存储内容在存储器写访问的情况下被反射到至少一个冗余的被动控制单元的存储单元中，则调用反射例行程序，

- 所述反射例行程序的调用通过适当地设置存储器管理单元来引起，而不是通过从运行的程序中的显性调用来引起，

- 在执行所述反射例行程序期间把要写的存储内容存放在所述至少一个冗余的被动控制单元的存储单元中的存储区中，并且

- 在所述反射例行程序中重新地在另一虚拟存储区上进行已经导致调用反射例行程序的对所述主动控制单元的写访问，所述另一虚拟存储区被映射到与其写已经导致调用所述反射例行程序的存储区相同的物理地址上。

本发明的中心优点是，程序可以在几乎每种商业上可获得的处理器平台上运行，并且仍然达到与高耗费的特定解决方案相同程度高可用性，其中所述程序不包含针对冗余的预先支付并且此外只与特定硬件一起构成失灵安全的、高度可用的系统。

从而将来也有可能在不具有高耗费的情况下引入新代的控制计算机，也就是说可以显著地缩短革新周期。

由于调用反射例行程序（也被称陷阱例行程序）所引起的动态缺点通过以下各种途径来减弱：

-只在对其数据内容必须确实被反射到冗余单元上的页面进行写时才触发陷阱例行程序。从而许多用于暂时和局部的数据的写周期不受影响并且一如既往以全性能方式运行。也不限制所有的读访问。

-陷阱例行程序的功能可以被实施为微代码中的功能序列，这减少了在进入和离开陷阱例行程序时的动态损失。

-许多处理器提供协处理器用的特定接口。使对所述冗余的和主动的页面的存储访问的加倍可以在通过存储器管理单元发起之后也由硬件形式的协处理器实现。由此消除通过陷阱例行程序以及微代码例行程序所引起的动态缺点。

根据本发明，在连续操作中把主动单元的存储器的所有改变都通过陷阱例行程序反射到冗余的单元上。于是在差错的情况下可以在无信息损失的情况下切换到所述冗余的单元。

但是，该方法只是足以把存储器的连续改变反射到冗余单元上。然而如果在操作期间例如由于故障而更换单元，则由此不再达到主动板的相同的存储状态。在此情况下也必须把所有的静态数据、也即在操作中不再改变的这些数据传输给已更换的冗余的单元。

据此，本发明的有利的扩展方案在于一种对应于以下过程的方法：

-在更换或者开始使用所述被动控制单元以后，通过复制例行程序把主动控制单元的存储单元的存储内容一次完全地复制到所述至少一个冗余的被动控制单元的存储单元中，

-随着每个复制过程提高全局变量中的要复制的存储器地址，

-在执行所述第一反射例行程序时把要写的存储区的存储器地址与所述全局变量中的存储器地址相比较，

-如果在比较时存储器地址一致，则延迟执行所述第一复制例行程序。

所述主动单元的静态存储内容到冗余的单元的所述复制不是无意义的，因为在连续操作期间所述主动单元必须被复制。也即在数据恰好通过复制过程被传输给冗余的单元期间，所述数据可以被改变。复制通常通过在处理器寄存器中的加载和回写实现。在数据已

被加载到处理器寄存器中之后、但是在所述数据能够被存储到冗余的单元上之前，现在主动单元于是可以改变该数据。主动的和冗余的单元上的数据的由此可能的不一致性在迄今的系统中通过适当的专有硬件电路已被排除。

然而需要特定的硬件却使得不能够使用如在市场上可获得的标准组件。由此该解决方案造成高的开发耗费，所述高的开发耗费可以通过下面所建议的软件实现来取消。

本发明的中心优点是，在没有硬件形式的冗余支持的情况下可以在连续操作中使由标准组件所组成的系统同步，并且因此也可以在更换单元之后以更快的切换可能性使所述系统重新处于冗余式操作中。

如果在主动的与冗余的单元之间实现一致的存储器，则在正常操作中可以通过检验存储区变量来忽略动态开销。

本发明的另一方面在于一种用于执行所述方法的控制单元，具有（VSP）和物理存储单元（aPS），所述物理存储单元的存储内容在存储器写访问的情况下可以被反射到至少另一冗余的被动控制单元（rSt）的存储单元中，

-用于调用反射例程序的装置，用以在对存储区的存储器写访问的情况下使所述至少一个冗余的被动控制单元的存储单元中的存储区中的存储内容加倍，和

-用于执行已经导致调用反射例程序的对另一虚拟存储区的写访问的装置，所述另一虚拟存储区被映射到与其写已经导致调用所述反射例程序的存储区相同的物理地址上。

本发明的其它优点从借助于附图描述的以下所述实施形式中得出。

在附图中：

图 1 和图 2 示出主动的和冗余的控制单元的可能的结构。

图 1 示出在电信系统内部的主动控制单元的结构。示出一种主动处理器操作，包括代码序列 CA 的运行、所谓的陷阱例程序 TR、具有虚拟存储区 VSP 的存储单元、存储器控制单元 aMM 以及具有物理存储区 aPS 的存储单元，其相互通信（由箭头线表示）。此外还示出包含冗余的存储单元 rPS 的冗余控制单元以及输入/输出系统

EA, 所述主动控制单元通过所述输入/输出系统与冗余的控制单元通信。

在图 2 中几乎示出了相同的结构单元。主动控制单元 aST 附加地还具有另一复制例行程序 KR。主动的或者冗余的控制单元的冗余存储区用 aRS 或者 rRS 表示。

图 1:

根据本发明, 不再通过特定的硬件把主动单元的存储内容的当前复制供应给冗余的单元。

取代于此, 使用在所有相关的处理器中所具有的存储器管理单元 (也即存储器管理) 的功能, 以能够在对主动控制单元 AST 的每次存储访问时在运行时间期间检验是否必须从要写的的数据在冗余的控制单元 rST 上存放一个复制。

这通过按存储页面 (典型地 4 千字节) 设置属性为 “写保护” 或者 “只读” 是可能的。

如果然后过程对存储器控制单元 aMM 中的这种存储区进行写访问, 则中断指令执行 CA 并且调用陷阱例行程序 TR。

现在该陷阱例行程序分析存储器指令并且负责把相同的数据写在冗余的存储单元的相同的存储器地址上。这可以通过适当的标准硬件 (例如 PCI-Express) 来支持。

还必须执行在主动控制单元上的局部写过程。这最有效地通过对虚拟存储单元 VSP 的虚拟地址空间中的不受保护的存储页面的写周期来实现, 所述不受保护的存储页面被映射到物理存储单元 aPS 中的与受写保护的存储页面相同的物理页面上。这可以通过所有公知的存储器控制单元 aMM 实现。

在对局部存储器以及也对冗余的控制单元的存储单元 rPS 写访问之后, 陷阱例行程序在 (在陷阱例行程序中已被执行的) 存储器指令之后返回到正常的指令执行中并且继续正常的程序运行。

为了改善动态特性, 还可以把陷阱例行程序的所述功能在处理器中实现为微代码。在此情况下, 陷阱例行程序不会通过存储器控制单元 aMM 中的存储页面的写保护标志而被触发, 而是相应的功能序列作为微代码直接被发起, 所述相应的功能序列使对冗余的控制单元的写访问加倍。由此降低通过进入和离开陷阱例行程序所引起

的动态损失。

作为进一步的优化，可以在处理器上连接协处理器，所述协处理器在由通过存储器控制单元 aMM 发起后负责使写访问加倍。从而可以取消陷阱例行程序也可以取消微代码例行程序。

图 2:

在主动控制单元 aSt 上，软件或者复制例行程序 KR 连续地把存储器复制到冗余控制单元 rSt。这在小的、固定的单元（例如具有高速缓冲存储器线（Cacheline）的大小）中实现。同时在所述主动控制单元的所有其余的处理器上进行主动操作。由此导致不断地调用陷阱例行程序 TR，所述陷阱例行程序把存储内容的当前改变反射到冗余的控制单元 rSt 上。

为了避免与由陷阱例行程序所反射的数据的不一致性，把刚刚所复制的数据块（例如高速缓冲存储器线）的基本地址存放在全局变量中。该全局变量处于主动控制单元的公共存储器 aRS 中，主动控制单元的所有处理器都具有读访问（除了复制过程之外，所述复制过程为每个要复制的数据块继续写基本地址）。

如果现在在对存储器写访问时在任一处理器上调用陷阱例行程序，则必须在该陷阱例行程序中把具有基本地址的全局变量与当前的写地址进行比较。如果所述地址不同（也即，对刚好未被复制的区域进行写访问），则该陷阱例行程序可以正常通过并且写访问被反射到冗余的控制单元的页面上。如果应当在所述陷阱例行程序中确定进行对数据块的访问，所述数据库刚好被反射到冗余的控制单元的存储器上，则陷阱例行程序探测（pollen）全局变量，直到再复制过程结束并且已经将所述全局变量继续计数到下个数据组块为止。于是所述陷阱例行程序同样能够正常运行到结束。

对动态性能的作用：在陷阱例行程序中的探测很少（只有在刚刚恰好对当前所复制的数据块进行写时才需要），并且该探测持续时间不长（对这样小的区域的复制持续时间不长）。然而由于所需要的地址比较，对全局变量的询问在陷阱例行程序中总是产生开销。但是如果不运行存储器的复制过程，则附加的运行时间在正常操作中是很少的。在此情况下，如果较长时间不改变所述全局变量，则所述全局变量（由于陷阱的频率）在调用陷阱例行程序的情况下

可靠地处于高速缓冲存储器中，并且所述比较只导致很少的运行时间开销。如果并列地运行再复制过程，则所述开销是更关键的。在此持续地改变所述变量并且在高速缓冲存储器中没有当前的复制。在此情况下对存储器的存储访问必须作为陷阱例行程序的用于读取位于全局变量中的地址的开销来进行。

该原理对冗余系统的硬件和软件没有提出特别的前提。只是必须使再复制存储内容的相应例行程序经由包含当前要复制的地址的全局变量与控制装置一体化。

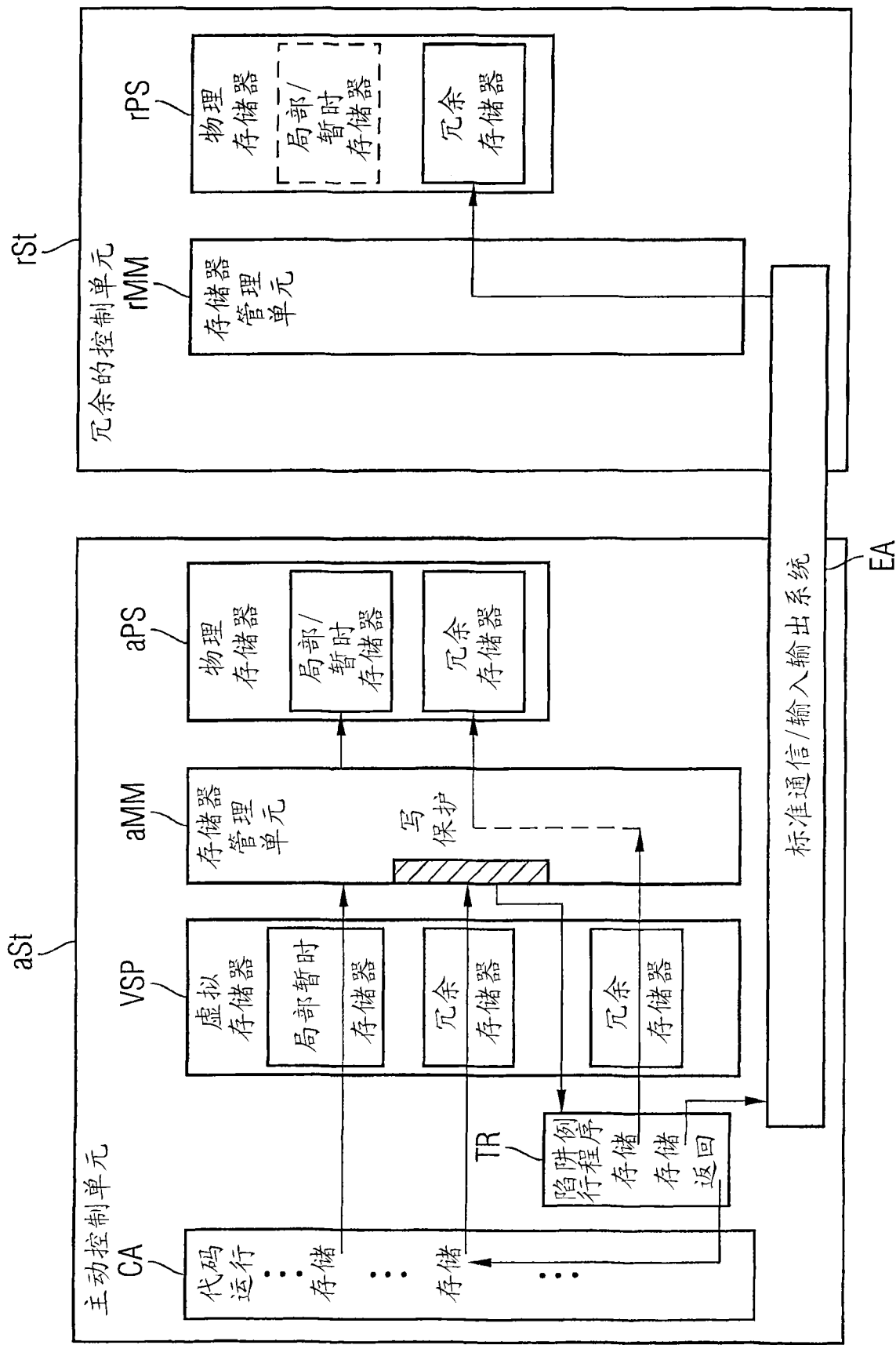


图 1

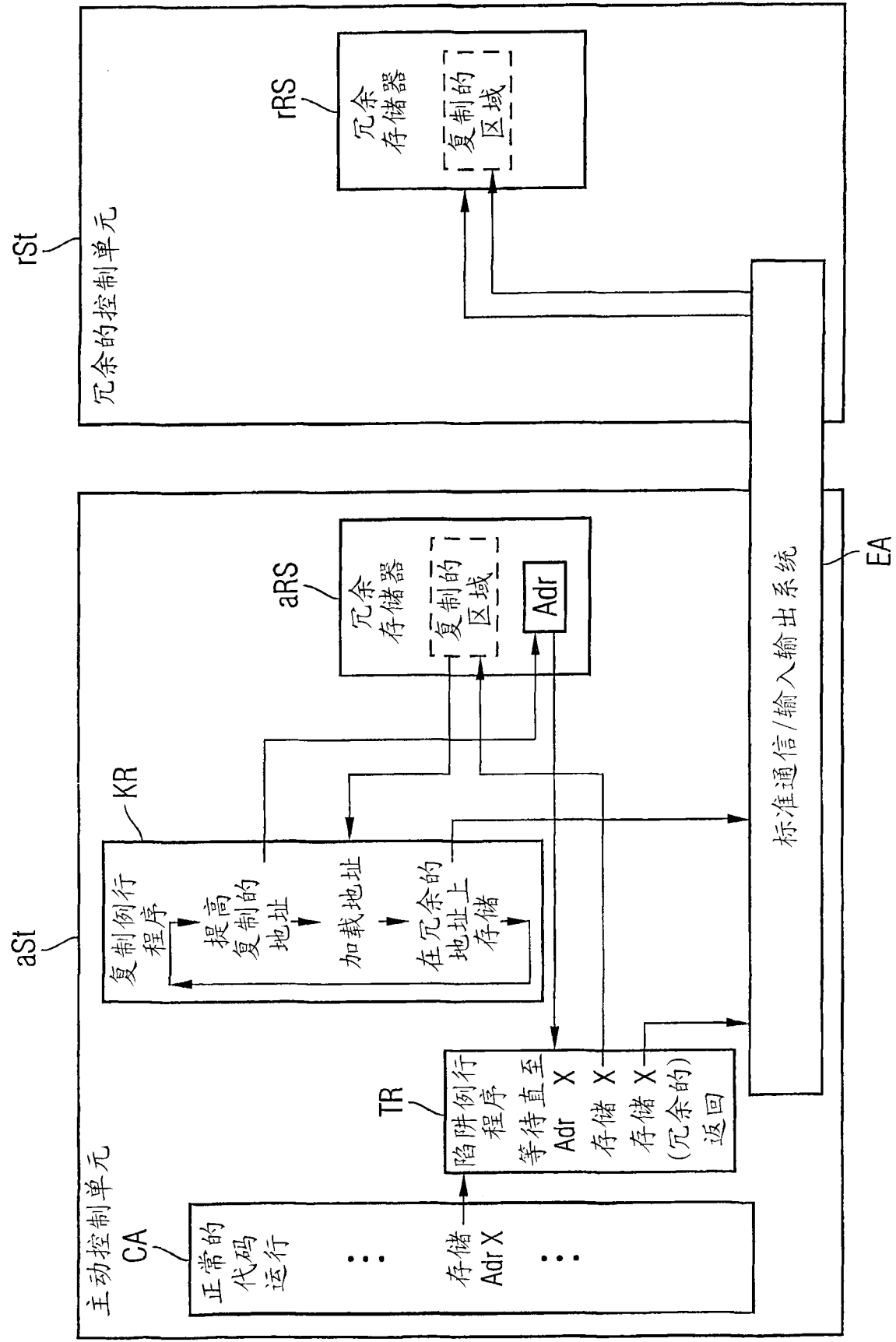


图 2