

[19]中华人民共和国国家知识产权局

[51]Int.Cl⁶

G06F 15/16

G06F 12/08 G06F 13/00

[12]发明专利申请公开说明书

[21]申请号 99104167.4

[43]公开日 1999年11月24日

[11]公开号 CN 1236136A

[22]申请日 99.3.22 [21]申请号 99104167.4

[30]优先权

[32]98.3.23 [33]US[31]09/046,430

[71]申请人 国际商业机器公司

地址 美国纽约州

[72]发明人 G·A·范休本 M·A·布拉克

P·-K·马克

[74]专利代理机构 中国专利代理(香港)有限公司

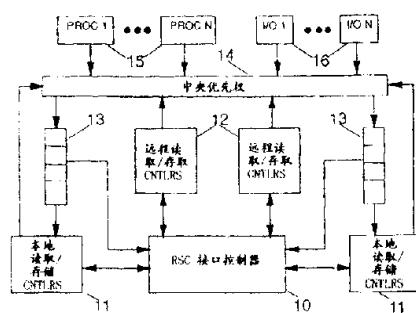
代理人 王 勇 陈景峻

权利要求书 5 页 说明书 29 页 附图页数 10 页

[54]发明名称 高速远程存储簇接口控制器

[57]摘要

用于管理对称多处理中资源的一个远程资源管理系统。包括：在对称多处理器系统中的簇节点之间具有接口的多个对称多处理器簇。每个系统簇具有一个本地接口和接口控制器。存在有都有其本地接口控制器的一个或多个远程存储控制器，和一条本地对远程数据总线。远程资源管理器管理在两个对称多处理器簇之间的接口，每个对称多处理器簇具有多个处理器，一个共享的高速缓冲存储器，多个I/O适配器和可从该簇中接入的一个主存储器。该远程资源管理器管理利用远程控制存储器管理资源以便分配工作到作为代理的一个远程控制器以便执行希望的工作而不必知道发起工作请求的请求人。所述工作只在远程请求者对于该工作处理有效时才被传送，而不需要在该对称多处理器簇之间经常通信。



权 利 要 求 书

1. 对称多路处理环境中用于管理资源的一个远程资源管理系统包括:

5 在对称多路处理器系统中的簇节点之间具有接口的多个对称处理器簇,

一个本地接口和接口控制器,

每个都具有其本地接口控制器的一个或多个远程存储控制器, 和一个本地到远程数据总线,

10 用于管理两个对称处理器簇之间的接口的一个远程资源管理器, 上述两个对称处理器簇具有多个处理器, 一个共享的高速缓冲存储器, 多个 I/O 适配器和从该簇中可接入的主存储器,

15 所述远程资源管理器利用远程存储控制器管理资源以便分配工作给作为代理的所述远程控制器以便执行希望的操作而不需要知道发起工作请求的请求者, 只当远程请求者对工作处理可得到时工作才传送, 而不需要在所述对称多路处理器簇之间经常通信。

20 2. 根据权利要求 1 的远程资源管理系统, 在每簇上具有单独的接口宏, 负责包括优先化排队请求, 通过接口发送新操作, 处理来自其它端的返回响应, 监视簇间所有数据传送的接口任务控制, 并且其中所述本地接口控制器不仅发送工作请求到远程端, 而且管理远程端上的读取/存储控制器, 由此立即发送新操作到可利用的远程控制器, 在那儿所述远程控制器变为以本地接口控制器名义工作的一个代理, 其以请求者名义依次工作, 取消任何发送识别操作主的信息的需要。

25 3. 根据权利要求 2 的远程资源管理系统, 具有允许几个本地操作被组合为单一的原子远程操作的命令重映射操作。

30 4. 根据权利要求 3 的远程资源管理系统, 其中用于数据只读拷贝的处理器读取请求, 和用于只读数据的读取请求包括存储保护密钥, 要求远程簇上的控制器利用相同的状态图和高速缓冲存储器管理操作, 并且所述接口控制器将重映射这两者为单独简化的公知只读线路读取的远程存储器簇 (RSC) 接口控制器命令以便减少必须由远程存储器簇接口控制器 (RSC) 处理的操作数。

5. 根据权利要求 4 的远程资源管理系统，其中当传送存储数据不必占用本地到远程的数据总线时，而且附加控制线路被要求发送目录信息时，所述接口控制器根据目录状态再变换传送命令为“强制舍弃”或“只读无效”命令。

5 6. 根据权利要求 1 的远程资源管理系统，具有包含大量服务一个或多个流水线分级高速缓冲存储器的读取和存储远程控制器的高端存储子系统的接口，并且其中一系列优先站选择一个请求以便通过接口发送和当涉及多条管道时，每个管道中的预优先站选择读取或存储请求以便发送到 RSC IC，而且在同一个周期中，远程存储控制器的接口控制器采用优先操作以便根据命令类型和有效资源选择最佳请求。
10

15 7. 根据权利要求 6 的远程资源管理系统，具有所述优先操作，由于多条管道可以请求在任何给定周期上使用接口，则操作将使读取优先于存储，只要远程读取控制器可利用；否则，存储将优先，只要远程存储控制器是可利用的，并且数据路径对需要一个数据路径的那些存储操作是可利用的，而如果两个请求都是读取的话，而且两个都具有可利用的资源，则一个循环决定允许哪个请求，但在两个请求都是存储的情况下，由可得到资源的管道决定优胜者，而如果两个都有可利用的资源，则使用该循环。

20 8. 根据权利要求 6 的远程资源管理系统，在每个本地接口控制器中具有远程资源管理器，它保证那个接口周期不被浪费地结束在远程端上的排队的发送工作。

25 9. 根据权利要求 6 的远程资源管理系统，其中每个远程存储控制器的本地接口控制器采用同步和异步响应总线二者来维持高速缓冲存储器的相关性，同时最大限度提高性能，并且其中异步响应总线用于标志远程操作正式结束的所有最后响应，并常常送往原始请求者并且其中所述最后响应利用允许本地目录用正确的最后状态更新的变化线信息标记。

30 10. 根据权利要求 9 的远程资源管理系统，其中所述远程控制器的接口控制器管理所有簇对簇数据流并比较来自本地存储控制器的请求与现存的远程读取控制器的请求，试图返回读取数据，并且其中在循环期间，在那里数据路径的两个竞争，最好的一个给予返回读取数据，并且其中在读取数据从远程主存储器中获得情况下，所述远程控制器的

接口控制器管理器监视和从存储器存储体接入数据时管理相应的数据路径，并且当远程存储控制器数据路径可利用时，数据将旁路远程读取缓冲器，因此减小了与暂时缓冲数据有关的正常等待时间。

11. 根据权利要求 9 所述的远程资源管理系统，其中为了改善已被
5 复制以便提高整个系统通过量的远程存储控制器资源的管理，所述远程
控制器的接口控制器利用在复制的远程读取资源之间的交替工作请求管
理命中远程高速缓冲存储器的后续读取请求，并发送第二读取请求到复
制的远程控制器，如果它有效，则允许复制的远程控制器开始加载其缓
冲器，同时第一远程控制器缓冲器仍然完成其数据传送，以便当完成第
10 一缓冲器的传送时，允许第二缓冲器立即通过接口传送其数据。

12. 根据权利要求 9 的远程资源管理系统，其中所述远程控制器的
接口控制器管理器管理所设计的免死锁机构以便监视可以导致交叉簇的
死锁，而且当检测这个方案时，所述远程控制器的接口控制器管理器将
15 通过返回一个特定拒绝响应给初始簇来拒绝未进行的操作，远程控制器
的接口控制器将依次发送该拒绝到原即读取/存储控制器使得操作可以
再试并持续地拒绝以及再试，直止死锁窗口消失。

13. 根据权利要求 9 的远程资源管理系统，其中当在伴随有一个新的
远程存储控制器操作的任何一个控制信息上检测接口优先差错时，在
命令发出后在固定数的周期内同步接口被用于发送接口差错状态，并且
20 一旦有差错，则通知原始读取/存储控制器并随后决定恢复合格，并且
远程存储控制器的接口控制器自动地复位相应的远程存储控制器的资源
以便允许该操作被再次请求。

14. 根据权利要求 1 的远程资源管理系统，其中所述系统包括用于
重映射超集命令为减少远程控制器复杂度的更小有效子集的命令的重映
25 射装置，并通过阻止不必要的数据传送来改善接口有效性。

15. 根据权利要求 14 的远程资源管理系统，其中所述命令重映射利
用交换位 (flip Bits) 来减少要求的门数，还允许该功能能在单独的
逻辑周期中执行以便改善系统性能。

16. 根据权利要求 6 的远程资源管理系统，其中为处理本地发出的
30 操作和远程返回的提供统一控制，它允许带外和带内数据共享所述数据
总线，因此减少接口 I/O 并还能使所述总线对于全系统性能以高度有效
方式进行管理。

17. 根据权利要求 6 的远程资源管理系统，其中通过采用读取请求优先于存储请求，考虑本地发出的请求和对远程发出的操作的响应，所述优先操作动态地分析每个周期的请求和远程存储控制器资源以便利用应用接口有效地平衡系统性能，用于操作的数据路径有效性要求数据路径和只允许操作通过接口发送，如果适宜的资源是可利用的。
5

18. 根据权利要求 1 的远程资源管理系统，其中，远程资源管理器提供同步交叉询问，它允许资源在固定的时间量内自动释放，即使目录在远程高速缓冲存储器中未命中；根据用于 CP 读取的管道快速分支 RSC IC 监视流水线查找 CP 读取，并当发现一个时，试图比正常情况提前一个周期发送它，其中读取必须优先于发现 RSC IC 之前加载到 LFAR 控制器；并支持早先的 PMA 读取 W/删除，其中本地 PMA 以交叉询问通过接口发送的相同时间开始读取数据，但万一在远程高速缓冲存储器中命中，RSC IC 发送信号给本地 LFAR 控制器以便取消早选的 PMA 读取以便开放存储器交错；读取缓冲器在分级高速缓冲存储器接口上旁路，所述 RSC IC 监视簇对簇数据路径，同时数据从分级高速缓冲存储器（PMA）中接收，并且如果所述数据路径可利用，则数据自动地旁路掉读取存储器并从 PMA 接收端口直接流到 RSC 接口。
10
15

19. 根据权利要求 1 的远程资源管理系统，其中远程资源管理器提供利用一个单独的交叉控制器来管理 4 个数据路径，能够 4 路同时数据传送，所述数据路径的每一个多路复用本地发出和远程发出的操作，并且由此可利用的数据路径被发送到优先机构，以决定调度下个操作。
20

20. 根据权利要求 1 的远程资源管理系统，其中所述远程资源管理系统提供：

25 一个加速只读无效操作，其中在远程端完成在只读无效中要求的所有步骤之前能够释放本地 LFAR 控制器，以致于所述 LFAR 控制器自由地开始新的操作，包括可以涉及发送远程操作，甚至随后的只读无效的一种；利用同步接口检测以便能使 RSC 资源自动复位，即使接口奇偶校验差错，包括通知相关 LFAR 或 LSAR 控制器以使所述控制器能重试该操作，
30 如果这样希望的话；

用于交叉簇免死锁的装置，据此远程 RFAR 或 RSAR 控制器检测潜在的死锁并发送送往相应 LFAR 或 LSAR 控制器的拒绝响应以使所述控制器

可以重试该操作；利用成对的 RSC 资源以使连续数据读取以“对” RFAR 或 RSAR 资源方式被分配给交替的资源，当两个成员对都可利用时，保证后面的读取将有一个远程缓冲器以便开始加载，同时前者读取的后尾字节仍然在其它缓冲器中进行处理。

说 明 书

高速远程存储簇接口控制器

本发明涉及计算机系统并特别表示一个高速远程存储簇接口控制
5 器。

过去的系统设计师曾使用各种装置实现在大型紧密耦合的对称多处
理器（SMP）中的高性能。它们从通过单独共享的系统总线耦合单个
处理器或处理器簇变化为将处理器耦合在一起成为一簇，借此簇使用簇
10 到簇的接口与一个中央互联网通信。在该中央互联网中在大量处理器周
围放置的并行系统通过一个中央交换机（即一个纵横制交换机）互联。
10

由于单独的总线协议可以服务多种类型的资源，所以共享的总线方
法通常提供最有效的系统设计。而且，另外的处理器，簇或外设可以经
济地连接到总线以扩展系统。但是，在大型系统中，与仲裁耦合的系统
15 总线上的拥塞，另外导致降低了系统性能并且产生低的SMP效率。这
些问题对于采用运行在超过500兆赫频率的对称多处理器可能是难以克
服的。
15

中央互联的系统通常具有对该系统中的所有处理器的共享资源提供
相同的等待时间的优点。在一个理想的系统中，相同的等待时间使多应
用，或在一个应用内的并行线索在任何系统结构或存储器分层结构不预
20 先知道的情况下，在可得到的处理器之间分配。这些类型的系统一般利
用一个或多个大的纵横交换机实现以在处理器和存储器之间发送数据。
基本的设计经常变换为大封装要求和对昂贵的部件封装的需求。另外，
25 实现有效的共享高速缓冲存储器结构可能是困难的。

紧密耦合的分簇方法作为折衷的解决方法。在这个应用中，术语C L
U S T E R是指共享一个单独的主存储器的处理器的集合，并且借此该
30 系统中的任何处理器可以存取主存储器的任何部分，而不管对一个特定
簇的它的类似性。不像不统一的存储器存取（NUMA）结构，在我们
例子中指的簇使用专用的硬件以保持在存储器和在每个簇内放置的第二
级高速缓冲存储器之间的数据相关性。因此，表现出对软件的统一的单
一的概念，没有任何存储器分层结构或例如存储器存储体交错的物理划
分。这些系统的优点之一是当数据对需要它的处理器保持紧密的似然性
时在一簇内紧密耦合的处理器的特性具有优越的性能。例如，如果数据

在一个簇的第二级高速缓冲器存储器内，则该存储器存储体交错连接到那个簇上。另外，当与建立在中央互联系统中的 N 路纵横制交换机相比时，它通常导致更加经济合算。但是，如果处理器频繁地从其它簇中请求数据，则该分簇方法会导致差的性能，而且随后的等待时间是严重的，
5 或者带宽是不足的。

直到与中央互联系统有关的许多花费高的问题可以以一种经济合算方式解决时，市场对于围绕共享总线或簇设计建立的经济系统将继续存在。本发明消除了传统的簇接口设计具有的许多缺陷，使得该系统可以最大化处理器性能，而不需要昂贵的高级插件和额外的插件板上的 c a
10 e h e。在与本发明有关的领域中的现有技术教导各种方法解决设计一个经济合算的高频率存储控制器的整个问题的孤立的方面。但是，如下面的例子所示，它们不能提供对满足根据本发明给出的目的的一个完整的解决方案。

在美国专利 4 5 0 3 4 9 7 (在 85 年 3 月 5 日授与 K r y g o w
15 s k i 等人) 中描述了由两个对称多处理器簇组成的系统。该发明教导利用专用的内置高速缓存器在处理器之间保持高速缓存器相关性的改进方法。但是，它不能解决与驻留在该簇内的内置流水线 2 级 (L 2) 高速缓存器有关的各种问题，仅仅利用连接到那个簇的所有处理器共享。它还不能集中最大化对于所有类型操作 (处理器， I / O， 存储器， 广
20 播信令， 交叉簇同步等) 的簇接口的全部效率。

在 1992 年 12 月 1 日，授与 M i l l e r 等人的美国专利 5 ,
168,547 和在 1993 年 3 月 23 日授与 c h e n 等人的美国专利 5,197,130 中公开了一个超大型 SMP 系统的实例。两者都描述了一种由大量簇组成的计算机系统，每个簇具有大量的 (即 32)
25 处理器和外部接口装置。每个处理器对在所有簇中的全部共享资源具有对称的存取。该计算机系统通过依靠大的纵横制交换机，高度交错共享的主存储器，一系列入站和出站队列以便将事务分级直到源和目的之间的通路成为可得的，和用于同步和共享数据的在该簇仲裁装置内的一组全局资源的组合实现其性能目的。该公开还从使用分级存储器系统 (包括二级高速缓存器) 中教导出一个结构，实现在多个并行处理器中划分工作的一种更有效的方式。
30

还设计出一些方法用于通过将多个 I / O 设备分簇并利用智能控制器管理它们来提高整个系统性能。美国专利 4,156,907 (在 1979 年 5 月 29 日授与 Rawlings 等人) 和 4,200,930 (在 1980 年 4 月 29 日授与 Rawlings 等人) 教导一个改进的自适应簇模块和数据通信子系统，它包括启动 I / O 处理器的固件，I / O 处理器卸载从主机传送的数据和消息。本发明能够与采用多种传输协议的大量远程外设接口。该自适应簇模块主要是有关根据不同协议将“字节”通信操作变换为可以使用一个单独协议更有效地传输到主机系统的整个消息。本发明还采用几种可靠而有效的特性，使通信子系统即使当主机系统出现故障时也能继续处理远程外围传输。尽管分开的技术可以肯定提高 I / O 子系统的性能问题，但是它们不能解决对于在两个处理器或一个处理器与在一个主计算机系统中的主存储器之间高速数据传输的需要。

存在致力于由本发明解决的全部问题的部分问题的几个发明，但是没有一个发明可以解决全部问题。更重要的，在这些发明中公开的方法的结合不能赋予由本发明提供的全部有效性的程度。例如，美国专利 5,392,401 (在 1995 年 2 月 21 日授与 Baruechi 等人) 教导在两个处理器之间传送数据的改进方法。但是，该发明依靠使用一个纵横制交换机，并且没有教导共享的二级高速缓存器的 cache 相关性。美国专利 4,445,174 (在 84 年 4 月 24 日授与 Fleischner) 教导用于利用专用高速缓存器和一个共享的二级 (L2) 高速缓存器联锁处理器的一个方法，但是没有解决带宽和与簇到接口有关的等待时间问题。美国专利 5,185,875 (在 93 年 2 月 9 日授与 Chinawamy 等人) 教导利用平行于将数据装置载到高速缓存器的同时将数据发送到请求的处理器，减少在存储控制单元之间数据传输等待时间的一种方法。尽管现在在计算机系统中广泛采用相似的技术，但是这个发明不能解决当存储控制单元不能提供用于请求访问高速缓存器的每个系统资源 (包括 I / O 和存储器) 的专用针接口时产生的问题。美国专利 4,785,395 (在 88 年 11 月 15 日授与 Keeley) 教导用于至少在一对处理器中共享高速缓存器的一种方法。但是它假定所有处理器可以以相同的等待时间访问高速缓存器。

几个发明描述用于在一个共享总线系统中仲裁通信的技术，在该共享总线系统中各个处理器或处理器簇通过一个共享总线与主存储器和外部 I / O 设备通信。例如，美国专利 4 7 8 5 3 9 4（在 88 年 11 月 15 日授与 F i s c h e r）描述用于仲裁共享总线的使用的方法。它们的技术包括给应答器优先于启动器并且使请求发送给即使处于忙时的接收模块。本发明通过只有当在远端的资源能够适应工作时使簇到簇接口忙来改进这个仲裁操作。另外，动态执行响应器和启动器之间的仲裁，每个周期没有固定的优先权。美国专利 4 5 7 0 2 2 0（在 86 年 2 月 11 日授与 T e t r i c k）采用串行和并行总线的组合构成系统总线。在几个“代理”中共享该总线，其中每个代理必须参加一个握手序列以获得使用总线的权利。本发明跟踪远程资源，使得它能够动态启动在一个单独的时钟周期上的新请求，不需要执行任何类型的总线协商。

本发明描述一种用于管理在双节点 S M P 系统中两个簇之间接口的装置。该优选的实施例结合到对称多处理系统中，该系统包括多个中央处理器，每个处理器具有一个专用 L 1 高速缓存器，多个 I / O 适配器，和一个主存储器，其中任意处理器或 I / O 适配器可以存取存储器的任意部分。处理器和 I / O 适配器的总数被平等地划分为两个簇。另外，主存储器由存储体或交错组成，其部分连接到每个簇。

在每个簇中，存在一个存储控制器，由一个共享第二级高速缓存器，各种控制器，和对每个处理器的离散接口（或端口），I / O 适配器，和主存储器组成，在本实施例表示的高速缓存器由多个存储体或交错组成并且内容（contents）由一个 8 路相关目录管理。图 1 A 描述的存储控制器说明主要功能部分并且将进一步在优选实施例的描述中描述。但是，一个单独簇内的存储控制器的简要概述对理解本发明是有利的。

存储控制器的主要功能是处理从处理器和 I / O 适配器读取并存储到主存储器或来自主存储器的请求的数据。因为存储控制器包括一个共享的第二级高速缓存器，它对于软件和操作系统在体系统结构上是看不见的，存储控制器负责执行目录和高速缓存器存取。所有的输入请求进入存储控制器的一个端口，在那里它们由中央处理器（C F A R）或 I / O 控制器接收。这些控制器产生请求到一个中央优先权单元，在那里在这些请求中仲裁并选择一个请求进入基于该地址的两个多级流水线之一。在流水线的每一级期间，请求器访问和 / 或保存诸如高速缓存器，

本地高速缓存器读取 / 存储控制器，数据通路控制器，数据通路先入先出缓存器，远程高速缓存器，读取/存储控制器等的各种资源。

当请求退出流水线时，本发读取 / 存储控制器之一负起管理通过完成的操作的责任。这经常请求通过流水线的另外通路，因此本地读取 /
5 存储控制器必须还参与中央优先权仲裁，并且还认为是一个请求者。在
本实施例中，我们包括高速缓存器控制器和主存储器控制器，作为本地
读取 / 存储控制器的部分，在它们之间它们包括需要访问来自高速缓存
器交错的数据的所有资源（包括例如 f i f o 缓存器的数据通路单元和
交叉点交换机），当高速缓存器出现没命中时，处理接入主存储器的数
据，执行进入高速缓存器交错的存储操作，并且舍去从高速缓存器进入
10 主存储器的过期数据（使用最近最少使用算法）以便为来自主存储器访
问的输入数据产生空间。

如上所述，在两节点系统的两个簇之间物理分配主存储器存储体。
但是，主存储器作为对位于 S M P 系统中任何地点的任意处理器或 I /
15 O 适配器的一个单独统一的实体。因此，本实施例加入称作远程读取 /
存储控制器的一个另外的控制器组。存储控制器记录哪些主存储器地址
分配给每个簇中的存储器存储体。无论何时数据存取（读取请求）没命
中本地簇中的高速缓存器，（此处术语本地指原始处理器或 I / O 适配
器连接的那个簇），本地读取 / 存储控制器必须询问远程（或“其它”）
20 簇以看数据是否保存在那个高速缓存器中。这些远程询问由远程读取控
制器处理，该远程读取控制器产生中央优先权的请求并且以对本地读取 /
存储控制器的相同方式访问资源。

另外，如果数据访问没命中远程的高速缓存器，但是地址表示它属
于连接到远程簇的存储器存储体，那么远程读取 / 存储控制器也与主存
25 储器控制器配合以便启动主存储器访问。对于必需将数据存储到存储器
的操作（例如舍弃高速缓存器的过期数据），地址再次确定本地读取 /
存储控制器是否可处理整个操作或是必须通过该双节点接口起动远程存
储操作。在这种情况下，该远程存储操作由远程存储控制器处理，该远
程存储控制器也与主存储器控制器配合以便将数据存储到存储器交错
30 中。关于本地读取 / 存储控制器，它们的远程对应物也包括需要处理内
部簇操作的所有资源（包括数据通路， f i f o 缓冲存储器，和交叉点
交换机）。

本发明适合用于管理包括前述的远程读取 / 存储控制器的资源的一个远程管理系统，并且还适合分配工作给这些远程读取 / 存储控制器，它们依次为执行所需操作的代理，无需知道启动工作请求的请求者。只有当远程资源对于处理工作是可以得到的时才分配工作，不需要对称处理器的多个簇之间的恒定通信。它使用最少的接口通信信号。

我们的远程资源管理系统利用减少的输入和输出针数以一种高效方式管理对称多处理器两个簇之间的接口。采用几种技术克服针限制并且仍然允许在例如 S/390 企业服务器的非常复杂的计算机系统内的集成，其中单一簇可以包括大量的甚高频处理器，一个共享的 2 级高速缓存器，若干 I/O 适配器装置，和一个主存储器。在这样一个系统中，性能是最高的，而与高速缓存器未命中有关的等待时间一定是最小化的。这样，本发明设法最大化整个系统性能，而最小化组装的成本。

首先，每个簇上的一个单一接口负责整个接口控制。这包括定排队请求的优先次序，通过该接口发送新的操作，处理来自其它方的返回响应，并监视簇之间所有数据的传送。因为控制 I/O 的有限个数，我们的发明使用与命令重映射耦合的远程资源管理的新组合以最小化需要发送的信息数量。本地接口控制器不仅启动到远端的工作请求，而且管理远端的读取/存储控制器，因此立即将新的操作发送到一个可用的控制器。远程读取/存储控制器干脆成为代表本地接口控制器工作的一个代理，他顺序代表请求者工作。通过以这种方式工作，因为远端无原因要知道，所以减少了发送识别操作所有者的信息的需要。

通过命令重映射操作实现了远程控制器的进一步简化，重映射操作使几种本地操作结合为一种单独的原子远程操作。例如，一个处理器读取用于只读数据拷贝的请求，和读取用于包括存储保护密钥的只读数据的请求，要求远程簇上的读取控制器使用相同的状态图和高速缓存器管理操作。因此，接口控制器将把这些重映射为称做 READ ONLY LINE FETCH (只读行读取) 的一个单独的简化的远程存储簇 (RSC) 接口控制器命令，这样减少了必须由远程存储簇接口控制器 (RSC) 处理的操作数目。

这种重映射操作的另一种好处是通过祛除不必要的数据传输能够更有效地管理接口数据路径。考虑一个 64 字节 I/O 存储需要将输入的 64 字节合并成在存储在主存储器之前的相同数据行的最新拷贝。这个操作

根据目标主存储地址和目前的高速缓存器状态可以导致三种不同的情况：

1. 如果数据目标在于远端的主存储器，并且未命中本地高速缓存器，那么 I/O 存储数据必须被发送到其它端用于合并。这将使执行从本地簇到远程簇的存储操作的 RSC 接口控制器（RSC IC）成为必需的。

2. 如果数据目标在于本地存储器，但是命中在远程高速缓存器，那么需要从远端检索出该行以便在本地产生簇合并。这将使对远端的交叉询问以及可能的数据读取成为必须的。

3. 如果该行拷贝存在于两个高速缓存器，那么因为输入的 64 字节可以与本地高速缓存器上的拷贝合并，所以唯一请求的操作是使远端的行 invalidate 无效。

一个较简单的设计是通过该接口无条件地发送具有 64 字节数据的 I/O 存储命令。另一端的远程读取存储控制器将根据目录状态接着执行必要操作。但是，在三种情况中的两种情况下，传送存储数据将不需要占用本地到远程数据总线。还需要另外的控制线发送目录信息。我们的发明采用一个智能接口控制器，将后两种情况分别重映射为“强制舍弃”和“只读无效”命令。

命令重映射提供了几种好处。首先，因为许多操作可以映射为较简单的原子接口操作的子集，所以它简化了远程读取/存储设计。其次，不要求额外的控制 I/O 发送簇之间的目录信息。第三，为防止等待时间的任意增加，在相同的周期内执行命令重映射，使得为通过接口的新命令分配优先级。

设计远程管理系统与一个高端存储子系统接口，该子系统包括服务一个单独的或多个流水线 2 级高速缓存器的大量的读取和存储控制器。一系列优先权站用于最终选择一个请求通过该接口发送。当包括多个管道时，在每个管道中的预优先权站选择一个读取或存储请求以便发送到 RSC IC。在同一周期期间，RSC IC 根据命令类型和可用资源，采用高性能优先权操作选择最佳的请求。因为多个管道可以在任何给定周期请求使用该接口，所以只要远程读取控制器是可用的，该操作将有助于使读取优先于存储。否则，只要远程存储控制器是可用的，和对于那些需要数据通路的存储操作数据通路是可得到的，则存储将优先。如果两个请求都是读取，并且都是具有可得到的资源，则一个简单的循环

确定赐给哪个请求，在两个请求均为储存的情况下，优胜者由具有可利用资源的无论哪一个流水线确定。再者，如果两者具有所有可利用的资源，则使用简单的循环。这种方法虚拟地保证只要有工作请求和可得到的资源，则命令将被发送，此外对处理器读取提供的优先处置提高了整个系统性能。最后，本地接口控制器内的远程资源保证接口周期将不是浪费结束在远端排队的发送工作。

为进一步减少由于 L 1 高速缓存器未命中的处理器数据访问等待时间，R S C I C 在没有读取或存储控制器请求使用接口期间采用“快速通路”技术。在这些周期期间，监视用于合法 C P 读取的所有管道。如果发现一个管道，则立即分配给远端，同时本地高速缓存器读取控制器被并行装配。这使读取请求得到一个对远端的周期标题开始，由此减少了返回数据的等待时间。

因为大的不均衡可以存在于对于命中远程高速缓存器的数据读取的最好情况等待时间与来自本地主存储器的数据访问之间，R S C I C 具有将远程高速缓存器命中传送给本地读取控制器的功能，因此使主存储访问被删除。这通过使存储器存储体自由工作于另一个请求有利于整个系统性能。

在使达到最大限度性能的同时 R S C I C 采用同步和异步响应总线，保持高速缓存器相关性。因为不可能确定在请求测试远端的高速缓存器状态以前完成操作的管道周期的准确数，使用异步响应总线用于所有最后的响应。这些响应标记远程操作正式结束并且被传送到原始的请求者。另外，标记最后响应与数据修改信息（行改变状态）结合，允许利用正确的最终状态更新本地目录。因为最后响应可以由于对单独的最后响应接口总线的争用被延迟。能通过在一旦知道原始远程高速缓存器交叉询问（X I）的报告提高性能。R S C 总能够使用同步 X I 响应总线在最小的时间帧中报告这个 X I 结果。由于 S C 的中央优先权站保证从远端接收的任何新的请求将立刻输入适当的流水线，所以这是可能做的。在一个固定的管周期执行原始目录查找并且结果通过 X I 响应总线发送回来。启动该操作的接口控制器知道 X I 响应在固定个数的未来周期里将是可得到的，所以使用一种简单的分级机制将命中 / 未命中结果传送给请求读取 / 存储控制器。

除管理控制接口以外，R S C I C 还管理所有簇到簇数据流。本发明包括足够的数据流 I / O，以便使每个管道每个周期以两个方向传送 16 字节，并且 R S C I C 能够利用潜在的 64 字节的每周期最大通过量。尽管有两个单向数据通路，一个给定的数据通路必须用作用于
5 存储从簇 A 至簇 B 的数据管道，和从簇 B 到簇 A 返回读取数据的管道。

R S C I C 不仅防止冲突，而且在不预先保留总线的情况下防止冲突。这依次使 R S C I C 将来自本地存储控制器的请求与来自常驻远程读取控制器的请求作比较以便试图返回读取数据。在竞争数据通路期间给返回读取数据优先。这再次有助于整个系统性能。而且，在从远程主存储器获得读取数据的情况下，当数据从存储器存储体访问时，R S C I C 监视相应的数据通路。如果该 R S C 数据通路是可得到的，则数据将旁路远程读取缓存器，因此减少了与暂时缓存该数据有关的正常等待时间。
10

我们已经证明用于改进远程存储控制器资源的管理的方法已重现提高整个系统吞吐量。例如，用于最大限度增加命中远程高速缓存器的后续读取请求效率的一种技术是准备在复制的远程读取资源之间选择工作请求。所有的读取以与被传送的前 16 个字节数据一致的最后响应结束。优点是它能使在远程读取控制器的缓存器仍是空的同时远程读取资源立即分配给一件新的工作。当新的一件工作也是命中远程高速缓存器的数据读取时，缺点就出现了。因为缓存器可以仍然是空的，所以远程
15 读取控制器必须通过管道反复循环该请求，直到缓存器变为可用时为止。R S C I C 通过将第二读取请求发送给如果是可利用的复制的远程读取控制器减少了这种情况。这使第二远程读取控制器在第一缓存器仍然完成它的数据传送的同时开始装载它的缓存器。这使在完成第一缓存器的传送时，第二缓存器立即通过该接口传送其数据。
20

远程接口通过使许多操作当操作在远端不能成功完成的情况下被重新试验还表现出高度的可靠性和可用性。这些类型的问题归入两种主要的分类：远程拒绝和接口错误。操作可以由远程簇拒绝以避免潜在的交叉簇死锁。这些死锁当存在比服务于它们的 R S C 资源多的存储子系统
25 读取和存储控制器时出现。远程读取和存储控制器包括设计为监视可以导致交叉簇锁的操作序列的死锁避免机制。在检测到这种情况时，控制器通过返回一个特定的拒绝响应给初始簇来描绘挂起操作。R S C I

C 将依次发送该拒绝给初始读取 / 存储控制器所以可以重试该操作。可以继续拒绝和重试操作直到死锁窗口消失。当在伴随新的 RSC 操作的任何控制信息中检测到接口奇偶错时，其它类型的恢复出现。同步接口用于在该命令发送以后的固定周期数内发送接口错误状态。在错误的情况下，通知原始的读取 / 存储控制器并且接着确定恢复的适当性。RSC IC 自动重新设置相应的 RSC 资源以允许操作再次被请求。这些和其它改进在下面的详细描述中阐述。为了更好地理解本发明的优点和特点，参照描述和附图。

尽管结合本发明优选实施例描述了本发明，但是本领域技术人员会懂得这里公开的原理适合于包括多于两个簇，和使用不同于我们的本发明的存储簇的系统。另外，本发明试图替换不同数量和结构的功能部件的储存控制器实施例，包括但不限于高速缓存器结构，主存储器构造，数据通路资源的个数和大小（例如，缓存器，控制总线等），各种控制器的组成，和流水线的个数和尺寸。

图 1 a 描述双节点对称多处理器系统的一个单独存储控制器簇。

图 1b 描述负责处理从远程簇返回的所有响应通信量的响应处理器，构成远程存储控制器 RSC IC (10) 的详细接口和子部件，并且还表示远程读取/存储控制器 (12) 与 RSC IC 之间的接口。

图 2 表示硬件的命令变换的优选实施例实现。

图 3 描述具有馈送主命令优先权站 (33)) 的它的流水线站 (32) 的所有优先权站。

图 4 表示如何将命令分级到适当的资源保持寄存器。

图 5 描述提供禁止位，有效位的资源寄存器 (59)，命令寄存器，原始请求者 ID 寄存器，LFSAR 控制器 ID 寄存器和 LFSAR 缓存器寄存器的一个单独组的详细图。

图 6 表示分级机制和响应处理器 (22) 如何利用分级机制跟踪 XI 响应。

图 7 描述工作在与其它描述的分级流水线相同方式下的只读无效分级流水线 (75)。

图 8 表示在一个 RSC IC 内的一个流水线接收和驱动交叉点的内部逻辑。

(注：为说明方便，图可以分成部分并且在使用多个图表的情况下，作为惯例，当看图时我们将图的顶部作为第一图表，接下来和经过的是后续图表。)

图 1a 描述一个双节点对称多处理器系统的单独存储控制器簇。本发明表示为与本地读取/存储控制器（11）的双组，远程读取存储控制器 / (12) 和集中的多级流水线 (13) 交互的远程存储簇接口控制器 (RSC IC 10)。这些流水线由一个中央优先权 (14) 站馈送，该中央优先权 (14) 站优先从 N 个中央处理单元处理器 (15) 或 N 个 I/O 适配器控制器 (16) 发出的异步工作请求。每个 CPU 控制器具有对 SMP 系统中特定中央处理器 (CPU) 的亲和性。流水线包括所有中央处理单元，I/O 适配器和本地和远程读取存储控制器已经读取和写入访问的一个共享的 2 级存入高速缓存器。在优选实施例中采用的高速缓存器相关性操作使数据驻留在只读状态的高速缓存器，只读状态使所有簇中的所有处理器和 I/O 适配器可以访问数据拷贝，或独占状态，使一个单独的处理器可以在任意时间占有该数据。任意簇中的任意处理器可以在任意时间请求数据的所有权，即使另一个处理器目前正占有那个数据。

优选实施例使用一个双流水线设计，其中中央流水线 (13)，远程读取/存储控制器 (12) 和本地读取/存储控制器 (11) 都是双倍的。本发明中公开的原理独立于存储控制器 (SC) 的周围结构并且可以仅有效实现为更传统的单独流水线 SC 设计或采用三个或更多流水线的较不传统的 SC 设计。本领域技术人员会知道如何能够容易地度量 RSC IC (10) 接口和内部功能模块以适合几乎任意的 SC 结构。

RSC 接口控制器 (10) 由与各种 SC 功能单元联系的若干子部件组成。因为只有一个单独的远程存储簇接口服务来自流水线 (13) 的双组和读取/存储控制器 (11, 12) 的 请求，单独的 RSC IC 必须适合多个本地接口和簇到簇接口。另外，它必须管理从本地簇到远程簇的通信流以及来自远程簇的通信流。图 1b 描述详细接口和组成 RSC IC (10) 的子部件。

由于在优选实施例中存储控制器的特性，请求的绝大部分来自本地读取/存储控制器 (11)。这些请求被分成叫做 LFAR 请求的读取请求，和叫做 LSAR 请求的存储请求。在优选实施例中，每个流水线有 4 个 LFAR 和 4 个 LSAR 请求者，这样总共有竞争使用 RSC 接口的 16 个可能的 LFSAR

请求者。在 LFSAR 控制器内的一个预先优先权站从每个管道中选择一个请求者并且在任意一个时钟周期最多传送两个请求给 RSC IC。再者，本发明精神允许在任意个数的流水线中分配的任意个数的 LFAR 和 LSAR 请求者。另外，在 LFSAR 控制器 (11) 内的预先优先权站可以直接结合到 RSC IC 内的命令优先权单元中。

将我们的注意转向 LFSAR 控制器 (11) 和 RSC IC (10) 之间的接口，存在对于与一个流水线有关的每组控制器的一个接口。每个接口由请求总线，C3 命令总线，C3 请求者 ID 总线和 C3 地址总线组成。8 位请求总线包括用于每个 LFAR 和 LSAR 请求者的每个位，在任意周期只有一位是有效的。这位指定在那个周期由预先优先权站选择的 LFAR 或 LSAR 控制器的 ID。响应的总线将命令，地址和 CPU 的 ID 传送到启动流水线中命令的 I/O 适配器或 SC 控制器。所有这种信息在相应于流水线第三级的 C3 周期提供给 RSC IC。如果 RSC IC 不能承诺在它被提交的周期中的请求，则 LFSAR 控制器可以继续提供相同的请求或动态选择在下一个周期中的一个新的请求。

除了 LFSAR 控制器，流水线 (13) 本身还作为允许叫做快速通路的操作的请求者。快速通路使 RSC 能监视所有管道并且在需要操作和没有从任意一个 LFSAR 控制器 (11) 挂起的工作时开始一个远程读取操作。从每个管道的第一级 (C1 周期) 获得 C1 命令，C1 地址，和 C1 请求者 ID 并且发送到 RSC IC 的优先权 (21) 子单元内的一个优先权站。预先优先权站的输出被传送到主优先权站 (也位于 RSC 优先权 (21) 子单元内。在那它与来自 LFSAR 控制器 (11) 的请求竞争)。

在每个周期，RSC 优先权 (21) 站检查挂起的工作请求和使用一个高性能操作确定哪个请求应该被允许通过接口。当选择一个请求时，允许被发送到相应于选择的操作的流水线的 LFSAR 控制器 (11)。允许表示选择的操作是否是来自 LFSAR 控制器 (11) 的一个挂起请求或是一个从流水线 (13) 启动的快速通路操作。当允许被发送到 LFSAR 控制器时，选择的命令以及它的相关地址，请求者 ID，和表示 RSC 资源的标记线通过 RSC 接口发送到远程簇。

所有的 RSC 操作请求来自远程簇的某些类型的完成响应。另外，数据读取还请求表示该请求的数据是否存在在远程高速缓存器的一个交叉询问 (XI) 响应。所有的处理通过 RSC IC (10) 处理并且或者直接发送

或者间接发送给原始请求者。大多数时间响应被解码并且适当状态，释放和高速缓存器相关性信息被传送到 LFAR 或 LSAR 控制器。但是，许多操作包括将响应发送回原始的 CPU 控制器 (CFAR 15)。不管响应的最后目的地，RSC IC 跟踪对于每个操作的所有必要信息以便在一个单编码响应总线上可以多路复用用于所有 RSC 操作的响应信息。RSC IC 接收输入的响应到响应处理器 (22)，其工作是解码该响应并发送适当的信息给 LFSAR 控制器 (11) 或 CPU 控制器 (15)。

图 1b 还表示远程读取/存储控制器 (12) 与 RSC IC 之间的接口。远程读取/存储控制器被分为单独的远程读取 (RFAR 12a) 和远程存储 (RSAR 12b) 控制器。远程读取控制器 (12a) 负责接收来自其它簇的读取请求，通过流水线 (13) 处理它们并返回该数据 (如果可能) 以及必要的响应信息。远程存储控制器 (12b) 负责接收输入的存储操作 (和任意伴随的数据)，通过流水线 (13) 处理它们，并且返回必要的响应信息。每个流水线具有与它相关的的一个 RFAR (12a) 和一个 RSAR (12b) 控制器，因此四个请求的最大数可以发送给 RSC IC 以便在给定的周期返回编码响应总线上的信息。RSC IC 内的响应优先权 (23) 子单元在这些请求之间仲裁并且管理编码响应总线上的通信量。如果一个 RFAR 控制器需要返回读取数据，则响应优先权 (23) 站与 RSC 优先权 (21) 和 XPT 控制器 (25) 通信以便保证数据路径是可用的。

本发明的关键方面之一是使用资源寄存器 (24) 跟踪在本地端的所有的远程活动。为了最小化接口 I/O 和最大化吞吐量，RSC IC 作为在本地端的原始请求者的代理。它跟踪远程端的 RFAR 和 RSAR 资源，减少了对固定簇到簇通信的需要。资源寄存器 (24) 与 RSC 优先权 (21) 站联系以便保证只有在 RSC 资源是可得到的情况下才启动操作。在启动操作时，RSC IC 标记选择的 RFAR 或 RSAR 作为“正在使用”，并且资源保持正在使用直到接收到表示操作完成的响应为止。接着使用这些响应复位该资源并且使该资源对于新的操作是可用的。

在优选实施例中，有总共由用于每个流水线的 2 个 RFAR 和 2 个 RSAR 组成的 8 个 RSC 资源寄存器 (24)。两个 RFAR/RSAR 的每一个是彼此相同的并且存在主要是通过使多个远程读取和存储操作由每个管道同时处理来提高性能。再次注意到本发明既不需要也不限于每个管道 2 个 RFAR 和 2 个 RSAR。以相同的方式而不管存在的个数跟踪所有远程资源。

在选择操作之前，原始流水线命令执行对 RSC 命令的变换。在许多情况下，产生的 RSC 命令与原始命令是相同的，但在某些情况下，重映射命令代码点以便使类似的操作共享一个单独的 RSC 代码点。这个步骤还保证所有 RSC 读取命令落入连续的范围（优选实施例中的‘01’ x-‘1F’ x），同时所有存储命令落入不同的连续范围（优选实施例中的‘20’ x-‘3F’ x）。在启动操作时，RSC IC 使用两个选择位表示 8 个资源应该服务新选择的操作的另一端。该两个位指的是流水线和在那个流水线内的双资源的位应该处理该命令。命令的位 0 确定该命令是读取类型（位 0=0）还是存储类型（位 0=1）。所有的读取命令是 RFAR 服务的，而存储是由 RSAR 处理的。因为命令和地址必须总被发送，这个方法只需要仅发送一次的两个另外的接口控制位，以便同步两端。还应该注意到该优选的实施例描述了一个 RSC 接口总线发送除命令，地址和选择线路以外的请求 ID。这个请求 ID 仅仅是通过 RSC 接口传送和发送到远程端的诸如 CPU 控制器或 I/O 适配器控制器的一个请求者。由本发明采用的该技术不需要知道原始请求者的 ID 以便实现权利要求书阐述的目的。

最后的子单元是交叉点（XPT）控制器（25），负责管理连接簇的四个数据路径。在优选实施例中，对于每个管道有两个单向数据路径，因此允许四个数据传输同时发生。每个数据路径具有它的 XPT 总线使得四个操作可以同时发生。数据路径是 16 字节宽，并且每个周期可以传输四倍长字（16 字节）。

本发明的目的之一是利用远程资源管理的使用以便最小化读取/存储控制器（12）的尺寸和复杂度，以及必须通过簇到簇接口交换的信息量。在具有复杂的存储控制器的高端 SMP 系统中，实际上在一个本地簇中可以启动的每个命令还可以通过用于在远程簇处理的接口发送。当这些命令被分成一系列的原子操作时，一个操作发现在远端的 RSC 读取/存储控制器（12）可以使用相同状态机处理几种相似的命令。因此，为了简化 RSC 设计，本地端的一些原始命令被重映射为相等的“基” RSC 命令。例如，一个“具有存储保护密钥的排它读取”导致相同的流水线序列和目录修改动作作为“没有密钥的排它读取”。因此，RSC 接口控制器在通过接口发送它之前将有密钥命令（‘06’ x）的排它读取重映射为简单的排它读取命令（‘02’ x）。

优选实施例以图 2 所示的硬件实现命令变换。原始命令输入为实现命令变换表 (28) 的交换 (FLIP) 位要求的由逻辑门组成的交换 FLIP 位产生器 (26)。原始命令与目录状态和目标 L3 组合以便确定如果存在情况下的那些位，是否需要被交换。产生的 FLIP 位与 XLAT (27) 块中的原始命令异或以便产生在命令变换表 (28) 的 RSC 命令列中所示的所需的 RSC 基命令。设计 RSC 命令变换器以执行在一个单独时钟周期内的变换和作为一个独立的功能单元。因此，本领域技术人员可以知道采用这个方案的灵活性。例如，命令变换器可以物理地实现为 RSC 接口控制器 (10) 的部分或它可以包括在启动对 RSC 的工作请求的控制器中。而且，该变换器可以与相同的逻辑周期结合在一起作为 RSC 命令优先权站，或者如果原始命令是可得到的，则它可以在更早的周期中执行。例如，在优选实施例中，在第二流水线级 (C2) 流水线快速通路命令是可以得到的，因此它可以在 RSC 命令优先权周期之前变换。

命令变换的使用以几种方式提高了接口效率。首先，如果所需的数据驻留在本地高速缓存器，许多操作不需要通知远程端。因此，命令变换器将使用目录状态以便从请求 RSC 接口的使用中排除这些类型的操作。第二，命令变换保证数据传送只需要在一个方向上发送，而不是原来先处理，再返回原始端。例如，没有变换的一个 I/O 存储 64 字节 (命令 28) 将导致无条件的 64 字节数据传输，即使数据的最终目的地是本地 L3 存储器。这意味着 64 字节将被传送到远端，合并成目标行，并且修改的行将不得不通过接口再返回，所以它将存储在连接到本地簇的 L3 存储器中。我们的发明通过使用本地 L3 和目录状态优化了数据传输以便如果目的地址是远程 L3 存储器和数据不驻留在该高速缓存器，则仅通过接口发送 64 字节。如果数据未命中该高速缓存器，并且目标 L3 是本地端，那么将一个通知发送到请求从远程高速缓存器传送的数据的另一端。在这种情况下，原始数据不进行传送，并且如果目标行保持在远程高速缓存器上，则数据将仅通过接口返回。即使数据处于远程高速缓存器，整个操作只使从远端到本地端的单独的数据传送是必要的，在那它可以与 I/O 存储数据合并并且放置在本地 L3 存储器。最后，I/O 存储的第三种可能的情况是目标数据驻留在两个高速缓存器的只读状态的情况。再次，在这种情况下，I/O 存储数据可以与本地的数据副本合并，所以不必要通过接口传送任何数据。代之以原始命令被变换为传送到远

程读取/存储控制器 (12a) 的只读无效命令，所以在远程高速缓存器上的数据副本可以标记无效。

为了尽可能有效和便利地处理工作请求，RSC 接口控制器 (10) 采用多级智能优先权站。图 3 表示由馈送主命令优先权 (33) 的流水线预先优先权站 (32) 构成的整个优先权站。流水线预先优先权站监视寻找快速通路选择物的两个流水线的第一级 (C1)。任何 CPU 读取命令 ('01' x- '07' x) 被认为是快速通路的选择物。如果任意一个管道命令是选择物，则它输入预先优先权站 (32) 并且和其它流水线命令竞争用于选择进入 C2 分级区域。如果在那个周期只有一个管道具有有效选择物，则自动地选择它。如果两个管道都有有效选择物，则一个简单的循环确定下一个轮到谁。

无论何时选择管道命令进入 C2 分级区域，接着比较与第二流水线级 (C2) 有关的各种接口信号。这些 C2 拒绝信号由目录状态，来自各种 CPU 控制器 (15) 的拒绝信号，和来自 LFSAR 控制器 (11) 的快速通路信号组成。这些信号的组合确定在 C2 分级区域中的目前操作是否应该被完全拒绝还是发送到主命令优先权站 (33)。用于拒绝操作的可能原因是：

- 命中本地目录的 CPU 读取，具有适当状态，否认需要通知远程端数据。
- 来自任意的 CPU CFAR 控制器 (15) 的拒绝信号。
- C2 流水线无效被复位。
- 来自 LFSAR 控制器 (11) 的块快速通路信号。
- 来自 L3 存储器结构阵列的无效地址表示。

如果没有出现拒绝条件，则命令被传送到主命令优先权站 (33)，
25 在那它与来自两个 LFSAR 控制器 (11) 的请求竞争。

如图 3 所示，命令优先权站 (33) 从每个 LFSAR 控制器 (11) 以及从流水线预先优先权站 (32) 发送的快速通路信息中接收一组信号。另外，它还与 8 个资源寄存器 (24) 和 XPT 控制器 (25) 联系，所以它可以智能地选择一个适当的操作。

30 基本地，如果一个操作被挂起和 RSC 资源是可得到的，则该操作将总是试图选择一个 LFSAR 操作。如果只有一个单独的 LFSAR 控制器 (11) 请求，和 RSC 资源是可得到的，则选择它。如果两个 LFSAR 控制器 (11)

请求，并且只有一个控制器具有可得到的资源，那么它将获胜。如果两个 LFSAR 都请求，并且两个都有可得到的资源，则一个读取类型的操作将优先于存储类型操作。在两个请求具有相同类型的情况下，一个简单的循环确定下一个轮到谁。最后，如果没有 LFSAR 控制器 (11) 请求，
5 或没有资源可以得到以便承诺 LFSAR 请求，则选择一个快速通路请求。

资源的可用性根据操作类型变化。因为只有请求的资源是相应于处理该读取操作的流水线的读取控制器 (RFAR 12a)，所以读取是最简单的情况。因为 RFAR 忙于处理其它的读取操作，或者因为资源禁止开关可以是有效的，所以 RFAR 可能是不可得到的。命令优先权站 (33) 监视每个 RSC 资源禁止开关和无效位以便确定可用性。
10

为进一步提供系统性能，优先权操作结合响应处理器 (22) 工作以便最大化远程读取效率。通常读取将分配到下一个可用的 RFAR 控制器 (12a)。远程簇的 RFAR 将处理流水线中的读取请求和开始装载它的数据缓冲器。同时它将对远端的 RSC IC (10) 发出请求以便通过接口返回最后的响应和数据。最后的响应一被发送，RFAR 资源就被认为是可得到的和可以接收新的工作。如果一个新的非数据操作发送到那个 RFAR，则在来自前面读取的结尾字节仍然从数据缓存器中读出的同时可以处理它。但是，如果新的操作是第二数据读取，则通过远端的流水线继续再循环直到该缓存器变为可用的。
15

本发明通过保证如果两个资源在第二读取请求到达的同时是可得到的，则连续的数据读取将被发送给交替的 RFAR 控制器来防止这种情况。例如，如果 RFAR A0 正在处理第一读取，而在 RFAR A0 仍然忙的同时第二读取到达，则它将被发送给 RFAR A1 (假定 RFAR A1 是可得到的)。另外，如果 RFAR A0 由于最后响应被发送而变为可用的，并且接着第二读取到达，它将还被发送给 RFAR A1 (因为 RFAR A0 缓存器仍然在传送数据)。但是，如果一个类似只读无效的非数据操作在 RFAR A0 仍然忙的同时到达，则它将发送到 RFAR A1。如果这后跟一个数据读取的第三操作，并且 RFAR A0 是可得到的，这个新的数据读取将发送到 RFAR A0，即使缓存器仍然忙于传送结尾字节。换句话说，交替将请求发送到不同的 RFAR 的机制对于成对资源的任意一个是有帮助的。
20
25
30

图 3b 描绘了表示在上述装置和 RSC 优先站 (21) 之间交互的逻辑方框图。一个源触发功能存在于用于每对 RSC 源 (24) 的 RSC IC 之内。

表示用于流水线 A 的 RFAR 对的一个单独的源触发器 (35) 在图 3b 中描述。它从每个 RFAR 源 (A0 和 A1) 中接收有效信号。这些有效信号，连同来自 6 个其它 RSC 源的那些信号一起，也送到 RSC 优先站 (21)。此外，由 RSC 优先站产生的流水线 A 读取许可信号也送到源触发器 (35)。
5 最后，源触发器利用触发占有锁定 (36) 控制哪个源对之一将被分配下一个操作，如果有正确条件。源触发器产生与读取同意信号“与”两次的一个单独选择信号以便产生送往 RFAR A0 和 A1 源寄存器的负载 RFAR A0 和负载 RFAR A1 信号。

图 3b 中显示的触发器真实表 (37) 描述了怎样更改选择信号和触发占有锁定 (36)。如果两个源中只有一个是可以得到的，则选择信号将不使用该可得到的源，而不管触发占有锁定 (36) 的状态。如果两个源都是可得到，并且对该流水线发出读取同意，则触发占有锁定的目前状态驱动选择信号。另外，如果是可得到，则触发锁定在下一循环被更新以待应该操纵“其它”源的随后的读取。正如在触发器真实表 (37) 的底部所示，可得到的信号是源有效位 (59a)，无效位 (59f) 的一种功能和表示最后响应的 RST-RFAR-A0 锁定为此源所接收并且被认为在这个循环“可得到”。

存储类型操作比读取更复杂，因为它们可以涉及一个初始数据传送以陪伴命令。通过测试无效开关和有效位，命令优先站 (33) 以与 RFARS 相同的方式决定远程存储控制器 (RSAR 12b) 的有效部分。命令优先站 (33) 解码 RSC 基本命令以便看它是否请求数据传送。如果命令不要求数据传送，则必须检验数据总线有效性。为通过这个检验，必须满足两个条件：

1. 对应于发出存储的流水线的数据总线不总是繁忙传送数据。
- 25 2. 相应的 RFAR 控制器 (12a) 不被请求利用数据路径以便返回从其它簇发出的读取操作的数据。

如果两个条件都满足，或者如果存储命令不要求使用数据路径，诸如 ISK 操作 (命令 30)，则满足存储操作的源标准。

上述优先操作保证只要资源可用，新操作将进到远程端。另外，通过赞成 LFSAR 控制器 (11)，它提高了 LFSAR 资源的通过量，由此降低源阻和由等待彼此间完成联锁的资源造成的死锁。一旦选择一种操作，则用整 27 位地址跨越 RSC 接口发送命令 (或在其初始或再变换代码指

针)。远程端的 RSC 命令分配器根据 C0-C1-CMD 命令总线的位 0 的值发送该命令到 RFAR 或 RSAR。此外, C0-C1-管道-SEL 和 C0-C1-REQ-REG-SEL 被用来操纵对与处理该操作的流水线有关的选择的 RFSAR 控制器 (12a 或 12b) 的命令。这个远程资源管理技术使得大量的存储控制器资源在 5 多个流水线之间扩展以便利用有限的 I/O 数使用共享的 RSC 接口。

如前所述, C0-C1-REQID 是通过接口被发送的原始请求器的 ID, 但不参加远程管理操作。这个 ID 纯粹作为发送到远程端的信息来对待, 并沿着需要它的存储控制器资源通过。

当发出新的 RSC 操作时, 命令优先站 (33) 对相应的 LFSAR 控制器 10 (11) 发出许可。在优选实施例中, 通过延迟这个许可一个周期来缓和临界定时路径。然而这个延迟意味着来自 LFSAR 控制器的请求线将被启动多于必须的一个周期。当分析下一个操作请求时, RSC 优先站考虑这种情形诸如它不再次浪费选择相同操作的一个周期。有两种类型的许可发出向每个 LFSAR 控制器 (11), 所说的控制器 (11) 清楚地指在 RSC 优先 15 站中已经泄漏的。无论何时选择 LFSAR 请求, 常规的许可就发出, 同时如果选择流水线快速路径命令则发出特定快速路径许可。

一旦 RSC 优先站 (21) 选择命令去通过接口发送, 则命令和其相关信息 (LFSAR ID, 请求者 ID 和 LFSAR 缓冲器 ID) 被加载到合适的 RSC 资源寄存器。图 4 说明该命令是怎样进入到适当的保持寄存器。CLC 命令 (42) 和 C3 管道命令 (43) 分级寄存器的目的是通过图 4 右上角的 20 3 向多路器 (41) 缓和临界路径的定时。所有 CLC 命令从 LFSAR 控制器 (11) 中发出并是定时临界。为了改善全部的系统性能, 远程操作在单周期中经过接口发送。由于这个相同的命令必须通过优先逻辑和小的交叉点开关以达到适宜的 RSC 次源寄存器, 这就出现了询问路径。优选实 25 施例通过分级来话 CLC 命令和通过交叉开关先发送的管道命令解决了这个问题。

转移我们注意力到图 4, 更详细地, 与每个 LFSAR 控制器 (11) 请示相关的命令被分簇为 CLC 命令分级寄存器 (42)。并行地, 由流水线预优先站 (32) 选择的 C2 流水线命令被分级为 C3 管道命令分级寄存器 30 (43)。还以并行, CLC 和管道命令通过 3 向 MUX (41) 流动。这个 MUX 受来自命令优先站 (33) 的许可线控制。每个 CLC 命令可被加载为或者两个 RFAR 命令 (45) 或与 CLC' S 流水线有关的 RSAR 命令 (47) 寄存

器。这意味着每个 CLC 命令具有 4 个可能的目的。由于快速路径命令被限制为 CPU 读取操作，则它们只能加载为两个 RFAR 命令（45）寄存器。该命令通过包括 2 路 MUX（44）和网关（GATE WAY）（46）的交叉点开关发送。2 路 MUX 由在 CLC 命令（42）和 C3 管道命令（43）寄存器之间选择的信号控制。网关（46）由允许 CLC 命令（42）通过的单独门线控制。所有这些控制信号被正交并且由组合具有选择下一个有效资源的优先逻辑的选择 RSC 操作的许可来造成。
5

RSC 包含所有交叉簇操作的 8 个资源寄存器（24）。这就允许对每条流水线进行两次读取和两个存储操作以便同时发生。由于整个操作被本地 RSC 接口控制器（10）跟踪，则要求完成操作的所有信息必须保持在源寄存器（24）之内。图 5 描绘了包含无效位，有效位、命令寄存器、原始请求者 ID 寄存器、LFSAR 控制 ID 寄存器和 LFSAR 缓冲寄存器的单一组资源寄存器（59）。在优选实施例中，LFSAR 控制器（11）内的每条流水线有两个 LFARS 和两个 LSARS，但每个流水线只有两个 LFSAR 缓冲器。因此对于每个新操作，LFSAR 控制器（11）必须动态地分配两个缓冲器之一到每个 LFSAR ID。因此，缓冲器 ID 必须发送到具有每个新请求的 RSC IC（10）。
10
15

应注意，本领域技术人员能意识到 LFSAR 资源总数、缓冲器和其彼此关系是怎样对本发明不起作用。在缓冲器总数等于 LFSAR 资源数并遵循固定关系的情况下，RSC IC 将不要求额外的资源寄存器来跟踪这个信息。然而，也有其它实施例，其中 RSC IC 需要跟踪超出在优选实施例中说明的额外信息。不考虑在资源寄存器的必须跟踪的信息量，而仍然采用在此公开的原理。
20

返回图 5，该图说明稍微复杂一些的细节，因为它包括 C3 快速路径寄存器（55a）和 CLC 分级寄存器（55b）。图 4 说明的正是资源寄存器的命令部分，并描绘所有 8 个资源寄存器。另一方面，图 5 显示一个资源但显示要求跟踪给定 RSC 操作的所有信息是怎样加载到资源寄存器。从命令优先站（33）发出所有控制信号。这个图只用了优先许可（所用实际信号的子组）来描述各种资源寄存器怎样加载的逻辑定时。
25

30 在操作发出许可之后有效位（59a）开始加载在该周期上。或门（54）保证 有效位被加载而不管该操作是规则 CLC 操作还是流水线快速路径 OP。由于有效位在决定资源是否有效时起到重要作用，所以这保证在下

一优先周期资源被标记为无效。为了减轻通过驱动许可出去到完整的组资源寄存器 (59) 而造成的定时临时路径，我们的发明采取这种事实的优势，即剩余的信息可以在加载之前延迟。

从 LFSAR 控制器 (11) 发出的那些 CLC 操作，按照逻辑定时是最简单的。⁵ 对于这些操作，分级的 CLC 许可 (53) 经过或门 (54) 控制 2 路 MUX (56) 和网关 (58)。因此当许可启动时，包含 CLC 分级寄存器 (55b) 的所有信息在有效位 (59a) 的加载之后的一个周期上加载为剩余组资源寄存器 (55b)。

流水线快速路径操作被分级为只有 C3 命令和请求者 ID 寄存器的 C3 快速路径分级寄存器 (55a)。¹⁰ 在这个情况中，分级的快速路径 GRANTLC (52a) 通过 2 路 MUX (56) 选择 C3 流水线命令和请求者 ID 并加载它们为命令 (59b) 和请求者 ID (59c) 资源寄存器。一旦这些快速路径操作达到流水线第 3 级，它们就被加载为 LFAR 资源并分配一个 LFAR 缓冲器。¹⁵ 这是必须要的因为许多 CPU 读取操作要求多条流水线穿过使得要求一个 LFAR 管理整个读取操作。因此，一旦知道这个分配，LFSAR 控制器 (11) 将对紧跟流水线快速路径操作的周期上的 RSC IC (10) 产生一个特别请求。²⁰ 随着这个请求，LFSAR 接口将包含 LFAR ID 和 LFAR 缓冲器 ID。RSC IC 对信息将在 CLC BFR AND CLC REA 寄存器 (55b) 的时间上使用快速路径许可 LC2 触发器。这将允许这个信息通过 GA TEWAY (58) 通过并在紧跟着命令 (59b) 和请求者 ID (59c) 寄存器后的周期上加载到 LFSAR 缓冲器 (59d) 和 LFSAR ID (59c)。

优选实施例也描绘了利用执行位于 CLC REQ 寄存器的 8 个请求信号的 8-3 译码的译码器 (57)，并在 LFSAR ID 寄存器 (59e) 中存储该 ID 作为 3 位值。表示无效寄存器 (59f) 的 1 个附加位也被完全包括。²⁵ 这个单独位寄存器是也可以通过优选实施例的 UBUS 加载的一个可扫描寄存器。每个 RSC 源寄存器具有这样一个无效位，它允许资源经过微代码、稳定器，系统预置等等或者永久或者暂时无效。虽然这个位在 RSC IC (10) 常规系统操作中不起作用，但在工程预算和接口性能分析中起到帮助作用。

³⁰ 图 1b 所示的响应处理器 (22) 负责从远程簇返回的和送往适当完成信号给起始端的所有响应话务。存在有由 RSC IC (10) 接收的响应的两个主要类型许多 RSC 操作包括远程交叉询问 (XI) 以便决定数据是否

存在于远程高速缓冲存储器中。无论何时请求这些操作之一，在另一侧接收该命令并利用保证的优先级输入远程线。该确保的输入流水线允许在操作通过接口发出的时间和知道命中/未命中结果的时间之间存在同步关系。在本实施例中，在命令在 RSC 接口出现之后 XI 响应返回 4 个周期。
5

从 RSC 优先 (21) 周期开始，响应处理器 (22) 利用图 6 所示的分级结构跟踪 XI 响应。为了改善丢失本地高速缓冲存储器的某一读取类型的性能 RSC IC 利用表示读取是否指本地 L3 或远程 L3 存储器的地址位与同步接口耦合，以便确定是否自动离开操作。例如，如果该取指向 10 远程 L3，则 RSC 资源必须保持有效直止操作完成，因为希望的数据会从远程高速缓冲存储器或者远程 L3 来。然而，如果地址指向本地 L3，并且数据不存在于远程高速缓冲存储器则资源可以自由工作在新的操作上，因为数据读取可以被本地 LFAR 控制器处理。

每个 CLC 命令寄存器 (55b) 的每个循环由解码器 (61) 分析以便看是否它是要求交叉询问 (XI) 的读取命令之一。这个结果与 CLC 本地 L3 位 (60) 组合并且来自表示这个 CLC OP 的 RSC 优先站 (21) 的信号发出许可。按并行 C3 管道本地 L3 位与来自表示快速路径操作发出许可的 RSC 优先站 (21) 的类似信号进行比较。通过限定，所有快速路径操作要求交叉询问。由于许可相互排斥，则在一任何周期上只有一个分支可以启动。这些信号以所示的方式在 2 路和/或 MUX (63) 内进行组合，并且如果条件正确则导致加载 4 位 L3 分级流水线 (64) 的 1 位。这个流水线包括 4 位分级寄存器用于每个周期开始于 2 级而结束于 6 级。每个 4 位表示 RFAR 资源 (12a) 之一。虽然在图 6 未说明，但包括刚才描述功能的所有单元利用馈送每个 2 级位的结果输出被重复 4 次。又一次，20 由于这些 RFAR 资源中只有一个能加载到任何给定周期上，则 L3 分级流水线 (64) 中的每级的 4 个位被正交。当 XI 响应由响应处理器 (22) 接收时，流水线第 6 级对应这个周期。如果任何的 4 位被启动并且 XI 响应未命中，则通过并掉有效位 (59a) 预置相应的 RSC 资源寄存器。
25

除了只在数据子组读取期加载的特定 L3 分级流水线 (64) 之外，30 图 6 还显示了利用新开始的 RSC 操作加载的 RFSAR 分级流水线 (67)。每个 RSC 资源寄存器 (24) 包含表示资源加载在最后周期的单一位锁定。这些 8 个资源负载锁定 (65) 是正交的，因为只有一个资源可以利用新

操作每个周期加载。这 8 个寄存器的输出用 8-3 译码器 (66) 译码并且 3 位译码值存储到 RFSAR 分级线 (67)。这条流水线也从 2 级开始 6 级结束。与有效位耦合的 3 位 RSC 资源 ID 通过分级每个周期被流水线化直至达到 6 级。逻辑定时是这样的，即这个 ID 在先级到达，这个 6 级是 XI 响应和远程接口差错信号被收到的相同周期。

在接口出错的情况下，3 位 RFSAR ID 被解码并用于复位涉及操作的 RSC 资源寄存器 (21)。此外，采用硬件查找表 (68) 以便利用 3 位 RSC ID 索引到那个资源寄存器的 LFSAR ID (59e) 寄存器。LFSAR ID 寄存器内容还被解码并用于发送接口差错信号给适当的 LFAR 或 LSAR 控制器。例如，如果 RFSAR 分级流水线 (67) 的 6 级包括“010”值，则这就表示线 A 的 RSAR 0 是 RSC 资源。查找表 (68) 然后会在线 A RSAR 0 资源之内解码 LFSAR ID 寄存器，并且这个值将指向与这个特定操作有关的 LSAR。相联一个具有相应本地 LFAR 或 LSAR 控制器的给定操作的能力允许许多 RSC 操作重试。经常是接口差错被中断，因此重试操作的能力阻止了不必要的系统停止。

采用 RFSAR 分级流水线 (67) 的一种附加系统性能改善是快速只读无效。RSC 操作之一是只读无效，其中存在于高速缓冲存储器中的数据只读拷贝必须在远程端无效。这就出现了，例如当 CPU 希望读取具有排斥关系的数据，并且其它 CPU 具有数据的只读拷贝。如果其它 CPU 在远程端，则 RSC IC 将发送只读无效命令，在其它簇的远程读取控制器 (12a) 将处理。通常 这个结果导致简单流水线通过以便无效掉目录输入。有时这些初始流水线通过导致地址比较试图接入在远程 CPU 上表现的相同线路的另一个控制器。当这些冲突出现时，就有几次其安全地允许请求排除关系的 CPU 在冲突完全解决以前具有该数据。本发明的远程读取控制器 (12a) 具有能力在起始流水线通过期间检测这些“安全”情况，并经由同步 XI 通知 RSC IC 响应安全进行的总线。

图 7 描述了以与上述其它两个分级流水线类似方式工作的只读无效分级流水线 (75)。用于两个流水线的 CLC 命令寄存器 (55b) 利用滤出只读无效操作的 ROI 解码器 (73) 解码。这样，与 CLC 许可 (53) 耦合表示开始有效只读操作。再次，这些中只有一个可以在任何给定周期上开始。这个结果送给 ROI MUX(74) 并用于设定只能无效分级流水线 (75) 的 2 级。这个位向下送往 6 级，在那它并入在响应处理器 (22) 中接收

的 XI 响应。如果 R0 无效 6 级位启动并且 XI 响应丢失，则采用 RFSAR 6 级寄存器 (63) 和 LFSAR 查找表 (68) 以释放相关 LFAR 控制器并允许它完成初始操作。远程读取控制器继续处理只读无效使得 RSC 资源有效位保持启动。一旦远程读取控制器 (12a) 完成操作，则它返回到退出操作的最后响应并允许资源接受新一份工作。同时，与只读无效有关的 LFAR 可以开始新的操作。为了防止只读无效的最后响应对新的 LFAR 操作的最后响应出错，RSC IC 包含维持每个 RFAR 资源的寄存器。无论何时快速只读无效原理被用于释放 LFAR 时设定适当的维持寄存器，并且它阻碍了来自 LFAR 信令的下次的最后响应。一旦接收到最后响应，并且该操作正式完成，则维持寄存器随着剩余资源寄存器复位。

涉及远程操作，而不是交叉询问的本发明的所有操作以编码的最后响应结束。响应处理器 (22) 利用编码响应 ID 总线匹配具有开始操作的 LFSAR ID 响应。最小情况下，RSC IC (10) 向完成操作的初始 LFSAR 控制器发信号使得它们可以释放其资源。在数据从远程端读取的情况下，前方数据发送到相应的本地 LFAR 控制器使得它能更改本地目录状态。此外，信号被发送到 XPT 控制器 (25) 以便允许 RSC XPT 代码指针发送到数据流片中。

子集远程操作也要求整个响应代码前送到 CFAR 控制器 (20)。例如，CFAR 控制器得利用这个单独响应去发回一个最先和最后响应给中央处理单元。对于本发明中定义的 7 个响应位，位 0 和 1 不包括在实际的响应值中，而是，它们具有以下特定意义：

位 0 表示远程操作被拒绝，通常防止死锁状态。这个位导致重试信号正发向适当的 LFSAR。LFSAR 将试图在最后时刻重试操作。

位 1 表示线在 CHANGED 改变状态命中远程高速缓冲存储器。通过 LFAR 在数据读取期间运用这份信息以便计算本地目录。

剩余位根据开始的操作被编码以便表示各种完成代码。

除了处理从远程端返回的响应之外，RSC IC 也采用了响应优先功能来发送响应给远程端。这些响应以交叉询问 (XI) 的形式并最后响应从远程簇开始的操作并在本地簇进行处理。来自每个流水线 (总共 4 个请求者) 的本地 RFAR (12a) 和 RSAR (12b) 控制器表示现存 XI 响应并请求发送最后的响应到 RSC IC (10) 中。交叉询问只从属于读取操作。因此只有 RFAR 控制器 (12a) 能代表 XI 响应。此外，因为只有一个交叉

询问能在某一时刻由远程端开始，并且由于它们保证通过流水线以固定周期数处理，则只有 4 个可组织的 RFAR XI 响应之一能够在任何给定周期启动。因此，响应优先（23）逻辑简易化 ORS 共同的 4 个 XI 响应并前去输出到接口。

5 最后的响应请示可以从 RFAR（12a）和 RSAR（12b）控制器发出，并且远程操作在长度上变化很大，该响应非同步出现。响应优先（23）逻辑与 RSC 优先站（21）交互以便决定任何最后响应请示是否兑付。对于非数据读取的操作，响应逻辑利用简易优先操作选择 4 个 RFAR 之一并通过接口发送响应。如果多于一个 RFSAR 在同一周期发出请求，则操作更倾向于 RFAR 而不是 RSAR。这就通过保证 CPU 要求的读取数据不由 RSAR 响应话务不必要延长来改善系统性能。在两个 RFAR 表示同一周期请求的情况下，操作使用循环选择一个 RFARS。在没有 RFAR 能够兑付而多于一个 RSAR 被请求的情况下，简单的循环选择 RSAR。

10 本发明新的一个方面是在响应优先功能（23）和 RSC 优先站（21）之间的交互以便最大化共享数据总线的效力。由于返回读取数据（从属于远程开始的读取）必须共享如本地开始的存储操作相同的数据路径，则当等待存储传送完成时读取的电势存在要延迟。响应优先通过执行试图返回读取数据的最后响应请求的以下步骤来减少这个电势：

15 1、它检测看是否对应于请求 RFAR 的流水线的数据路径有效。如果无效，则优先逻辑将立即选择 RSAR 请求，如果未决定的话。

20 2、如果数据路径有效，则优先逻辑将选择 RFAR 并通知 RSC 优先站阻止包括数据传送的任何未定 LSAR 存储操作。

25 以两个优先功能（21 和 23）的操作周期地改变，这意味着它们估价当前环境每个周期并使所有决事实上在一个周期之内。因此，在因为数据路径无效而延长请求的情况下，它将在可以服务的第一个周期上服务。一旦请求被选择了用于通过接口发送，则许可被发送到请求 RFSAR 使得它将停止当前请求并在下一周期发出新的一个。除了实际响应外，RSC IC 也发送 3 位编码响应 ID 总线，它表示 RFSAR 正返回响应。另一端上的响应处理器（22）解码这个 3 位 ID 以便解决哪个 RSC 资源寄存器需要复位。

30 所有响应在单独一个周期中交叉到 RSC 接口，而不管两个操作利用附加信息的响应总线。在关键操作期间，实际关键紧跟着响应。在检测

字节绝对 (TBA) 操作期间, TBA 状态在下一周期也跟着响应。在任一情况, RFSAR 控制器发送特别信号给伴随着表示这是两个周期操作的响应请求的 RSC IC。这允许响应优先防止任何新 RFSAR 最终响应在那个第二周期期间选择。

5 在我们优先实施例中的 RSC 接口支持总共 4 个 QW 数据路径。对每个管道的每个方向中 (本地到远程以及远程到本地) 有一个数据路径。物理上, 每条数据路径要求 2 个 SCD 片来实施, 每个集成片携带数据 DW。这种设计在共享总数结构和真实专用点对点数据流之间兼顾。虽然在每个方向上有单向数据路径, 但每条数据路径必须多路复用从接口两端发出的数据。例如, 在任一时刻连接远程 SC 到本地 SC 的数据路径能够用于返回数据以响应本地发出的读取请求, 或者它能够用来送出随着由远程端发出的存储操作的存储数据。理想地, 这些操作会用分离的数据路径分隔开, 但分包限制阻止这样做。无论如何, 单向总线对于每条线存在两路的事实不允许每周期同时移动 4 个 QW (64 字节)。
10

15 RSC IC 包含负责监督所有 4 个数据路径的交叉 (XPT) 控制器 (25)。事实上, 每个数据路径的一半在每簇中受 XPT 控制器的控制。例如, 从本地 SC 到远程 SC 传输的数据被本地 RSC IC 驱动并由远程 RSC IC 接收。因此, XPT 总线的驱动部分从本地 RSC IC 发出, 同时接收部分来自远程 RSC IC。所有 4 个数据通道由位 (0: 5) 控制接收端和位 (6: 10) 控制驱动端的 11 位控制总线控制。这部分 XPT 总线下文被称为接收 XPT (RCV
20 XPT) 和驱动 XPT (DRV—XPT)。
25

图 8 表示在一个 RSC IC 之内一个流水线的接收和驱动 XPT。为了缓和临界定时路径, 无论何时有可能则提前设定 XPT 信息。XPT GEN (81a) 逻辑的作用是利用 RSC 资源信息和外部信号的组合来建立适当的数据路径控制。触发器 (82a 和颜色 2b) 作为一个门以便根据移动数据在确切的正确时刻释放 XPT 信息到 RSC XPT 总线。RSC XPT 总线的位由各种缓冲控制器接收并在数据片上交叉转换。这个逻辑组成启动缓冲器数据和写控制以及选择器的简易解码器。在数据传送之后由于数据片没有逻辑操作知识, 则 RSC XPT 总线必须“脉动”, 一次用于每个 QW 传送。因此, 如果数据线要求移动, 则 RSC IC 在 RSC XPT 总线上为 16 个连续周期保持适当的值。
30

把我们的注意力转向 XPT 控制器 (25) 的 RCV——XPT 部分，我们看到 RCV XPT GEN (81a) 逻辑由 RFAR 源寄存器、方框——XPT 信号和一个 RSC——CMD 缓送。如上所述，数据由这个 SC 接收出于两个原因：返回来自本地发出的请求的读取数据或来自远程开始的存储 OP 的来话存储数据。在前一种情况中，这个 RSC IC 负责读取操作并在一组 RFAR 资源寄存器中具有所有信息。XPT GEN 逻辑利用在 CMD (59b)、REQ ID (59c)、LFAR 缓冲器 (59d) 和 LFAR ID (59e) 寄存器中的信息来计算 RCV——XPT 的值并决定数据传送长度。如果长度大于 1 个 QW，则 XPT——CNTR (83) 利用适当的循环数加载。所有这些在通过接口调用命令之后很快发生。当数据返回时，该第一 QW 总是伴随着 ‘03’ X, ‘023’ X、‘05’ X 或 ‘18’ – ‘1B’ X 的编码响应。接收这些响应(匹配 enc-resp-id)之一到触发器逻辑 (82a)，触发 RCV——XPT 的释放到 RSC——XPT 总线。在涉及多个 QW 情况下，XPT——CNTR 将继续支持 RCV——XPT 值直到计算完毕。RCV——XPT 将根据操作引导数据到合适的 CPU 口，I/O 口，LSAR 缓冲器和/或 LFAR 缓冲器(用于后来包括的高速缓冲存储器)。在某种条件下，指事实上本地 CPU 的数据必须阻止在最后的时刻被发往该 CPU。RSC IC 从用于抑制触发器的每个 LFAR (11) 和 CFAR (15) 中接收几个块信号。

关于利用 RCV——XPT 总线的第二种方案涉及接收从远程簇发出的存储数据。由于这是一个完全的异步事件，则同时要求 XPT GEN (81a) 触发逻辑 (82a)。RSC IC 监视来话 RSC——CMD 总线部分以及来自本地 RSAR 控制器 (12b) 的数据前置触发器。则 RSAR——DADV 被启动并且命令子集解码为合适值，则 RCV——XPT 被设定并且被送到 RSC XPT 总线使得本地数据路径片能被导引以便接收来话数据并将之送到 RSAR 缓冲器。

驱动 XPT (DRV——XPT) 以类似方式工作。再次，两种方案需要利用这个数据路径。第一个涉及本地开始的存储操作。在这种情况下，RSAR 资源寄存器保持所有必要的信息以便设定 DRV——XPT 并加载 XPT——CNTR (83)，如果必要的话。由触发逻辑 (82b) 接收的 RSAR——ST——OP 信号根据通过接口发送的命令控制数据应该开始移动的定时。其它端上的 RSAR 将启动 RSAR——DADY 以便移动 RSC IC，使得它能“睡醒”并接收存储数据。RSAR——ST——OP 信号是简单的 RSAR 命令 (59b) 寄存

器的解码以便决定当前操作是否要求存储数据传送。所有存储数据从本地端的 LSAR 缓冲器中发出，并且 DRV——XPT 将控制这些缓冲器的读取。

其它方案涉及返回由其它端请求的读取数据。由于这个数据源可以是 RFAR 缓冲器、PMA 接口或任意的 CPU 遥感寄存器，则 XPT GEN (81b) 逻辑利用了除 MBA——ID 之外的来自 RFAR 控制器 (12a) 的信号组合。

非零 MBA——ID 指数据将从对应于该 ID 的遥感寄存器中来。如果该 ID 为零，则各种 RFAR 信号号用来确定数据是从 RFAR 缓冲器还是从 PMA 接口中来。这些信号之一，PMA——DATA——RDY 信号在数据正从 L3 存储器到存储控制器中传送的时间窗口期间由 RFAR 控制器 (12a) 产生。如果响应优先 (23) 可以在这个窗口期间处理 RFAR 编码响应请求，则该数据可以旁路过 RFAR 缓冲器并直接传送到 RSC 接口。另一方面，如果在响应优先 (23) 发出许可给请求 RFAR 之前到达窗口末端，则停止 PMA——DATA——RDY 信号。XPT GEN (81b) 逻辑将随后发送数据到缓冲器址到 RSC 接口有效并能从缓冲器移动数据到接口的这样的时刻为止。本发明的这一方面通过在 CP 读取到远程簇期间取消不必要的缓冲器加载或不加载来改善系统性能。

除了设定 DRV——XPT 以外，来自 RFAR 控制器 (12a) 的几个信号包含允许导出数据传送长度的 CRF——XFR——LEN 总线。对于返回的数据情况，触发逻辑 (82b) 由来自响应优先站的 RFAR 许可启动，所说的触发逻辑与表示“返回读取数据的 ENC-RESP 值相耦合。这使得 DRV——XPT 释放到 RSC——XPT 总线的第二个一半上。再一次，如果传送长度大于一个 QW，则 XPT——CNTR (83) 连续地启动 RSC——XPT 直止计数器用尽。

应注意，RSC 的异步特征导致频繁的冲突，诸如本地 RSC 试图开始存储操作同时返回远程读取操作数据。为了在数据路径上避免冲突，而体现最佳性能，XPT 控制器 (25) 与优先站密切交互以保证无论何时可能的许返回读取数据具有优先权。而且，一旦一条路径在使用中，优先站立即集中在不要求数据路径的发送新操作，致力于总是保持工作在簇间移动。

数据路径由称为驱动器 (诸如 RCV——XPT 和 DRV——XPT) 的 RSC IC 管理，所说的驱动器具有分别用于目的和源的相应代码指针定义。

每个 RSC 资源寄存器包含一个图 5 中给出的单独位无效 (59f) 锁定。这个锁定可被扫描到 ‘1’ 到永久地无效资源的任意组合。此外，这些锁定也可以在 UBUS 寄存器中通过使用 4 个位设定。在优选实施例中描绘的存储控制器包含可以读取，写入和 CP 微码修改的一系列这些 UBUS 寄存器。由于无效锁定可以经由这些微代码可控 UBUS 寄存器之一控制，RSC 资源的动态无效可以得到作为一部分微代码程序或暂时的插入码。人们这样使用可能是可比性能分析以便决事实上在各种工作负载上的双倍资源。代码指针控制将在这些不能转换的 RSC IC 出现的结果行为。

下代码指针（‘1’ X 到 ‘6’ X）与代码指针 ‘8’ X 到 ‘F’ X 的工作不同。通过启动 RSC IC 内的相关无效位，要求代码指针 ‘8’ X 到 ‘F’ X 不能简单选择资源。连续 UBUS 写操作能用于以任何希望的组合无效多个资源。下代码指针导致无效方案因此 RSC IC 内的优先逻辑将监视该无效模式以便限制以任何适宜方式的接口启动。例如，如果选择模式 ‘2’ X，则优先逻辑保证第二操作将不发送直至第一个完成为止。

虽然我们已经描述了本发明的优选实施例，但可以理解，本领域的技术人员，现在或将来可以做出各种改进和完善，这将落入本发明的保护范围之内。这些权利要求应该构成维持本发明第一次公开的适当保护。

流器出图

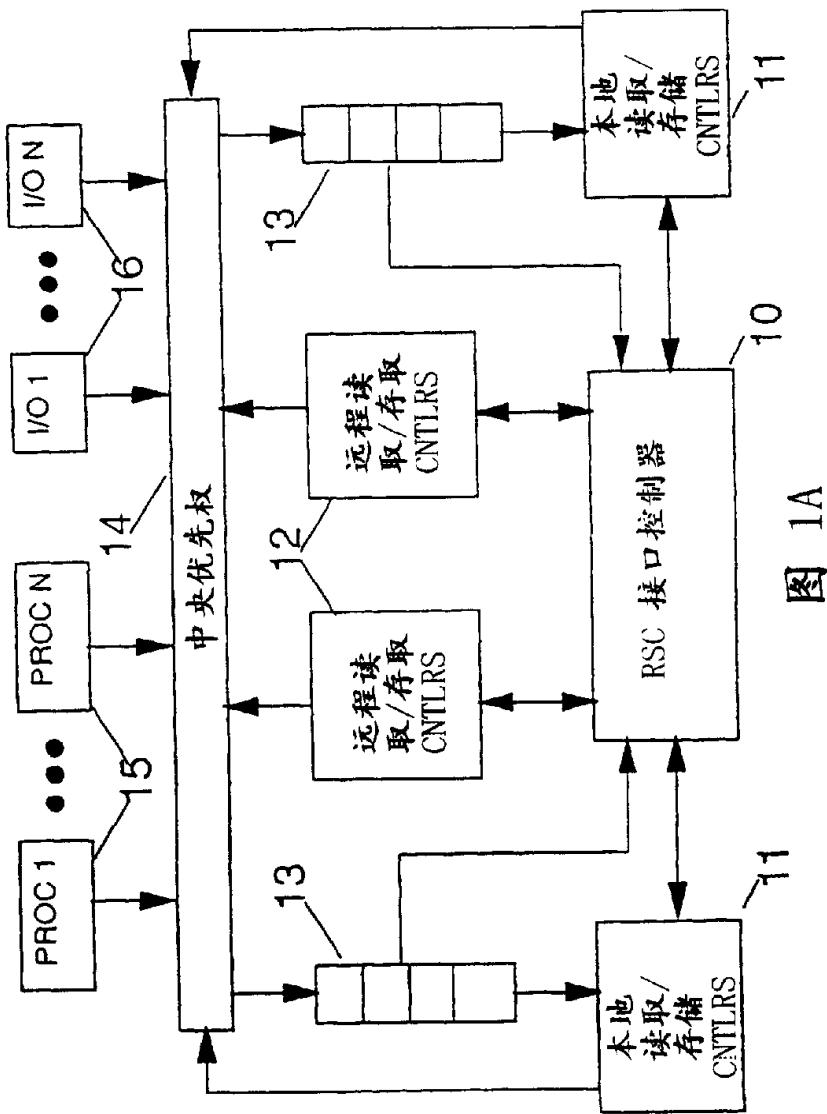


图 1A

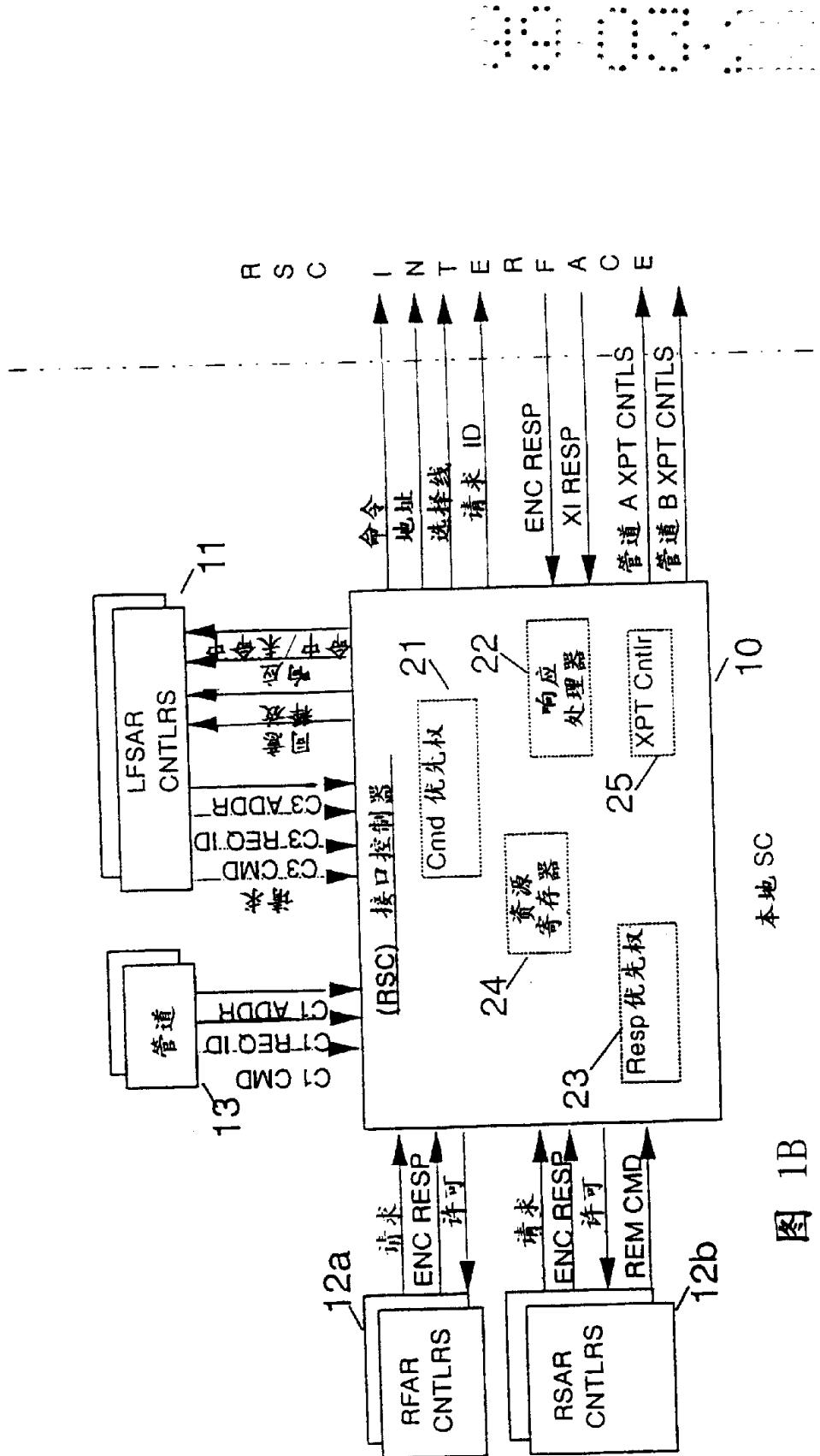
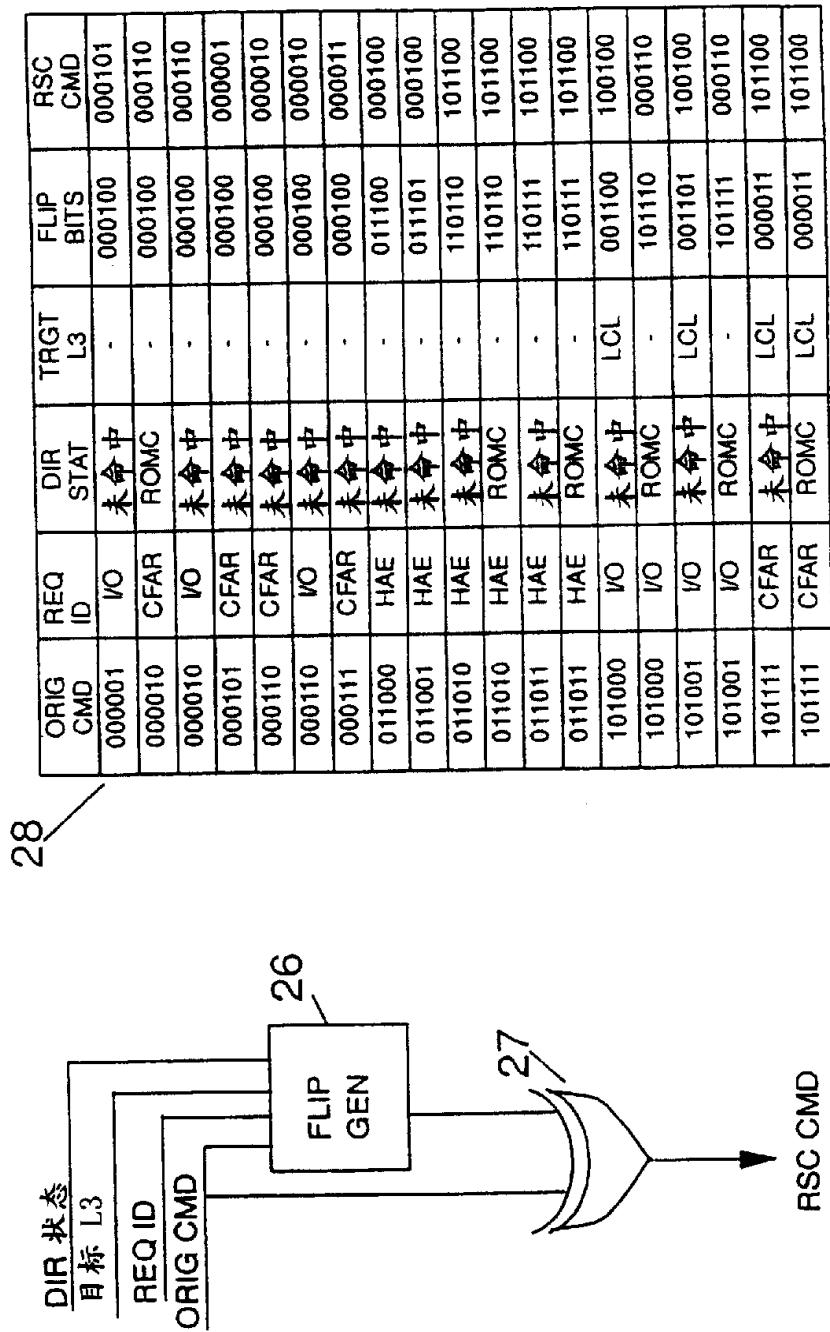


图 1B

图 2



3

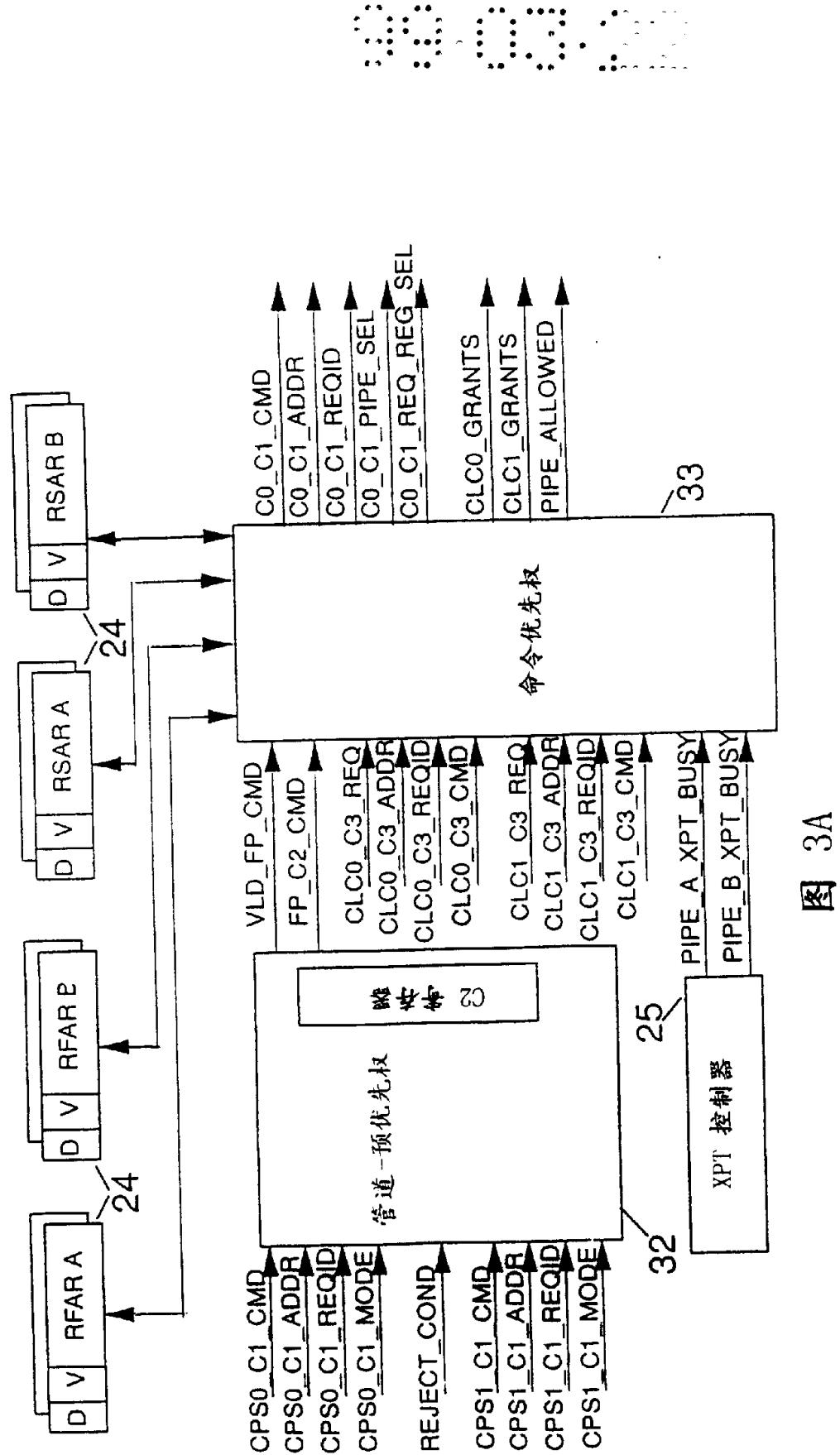
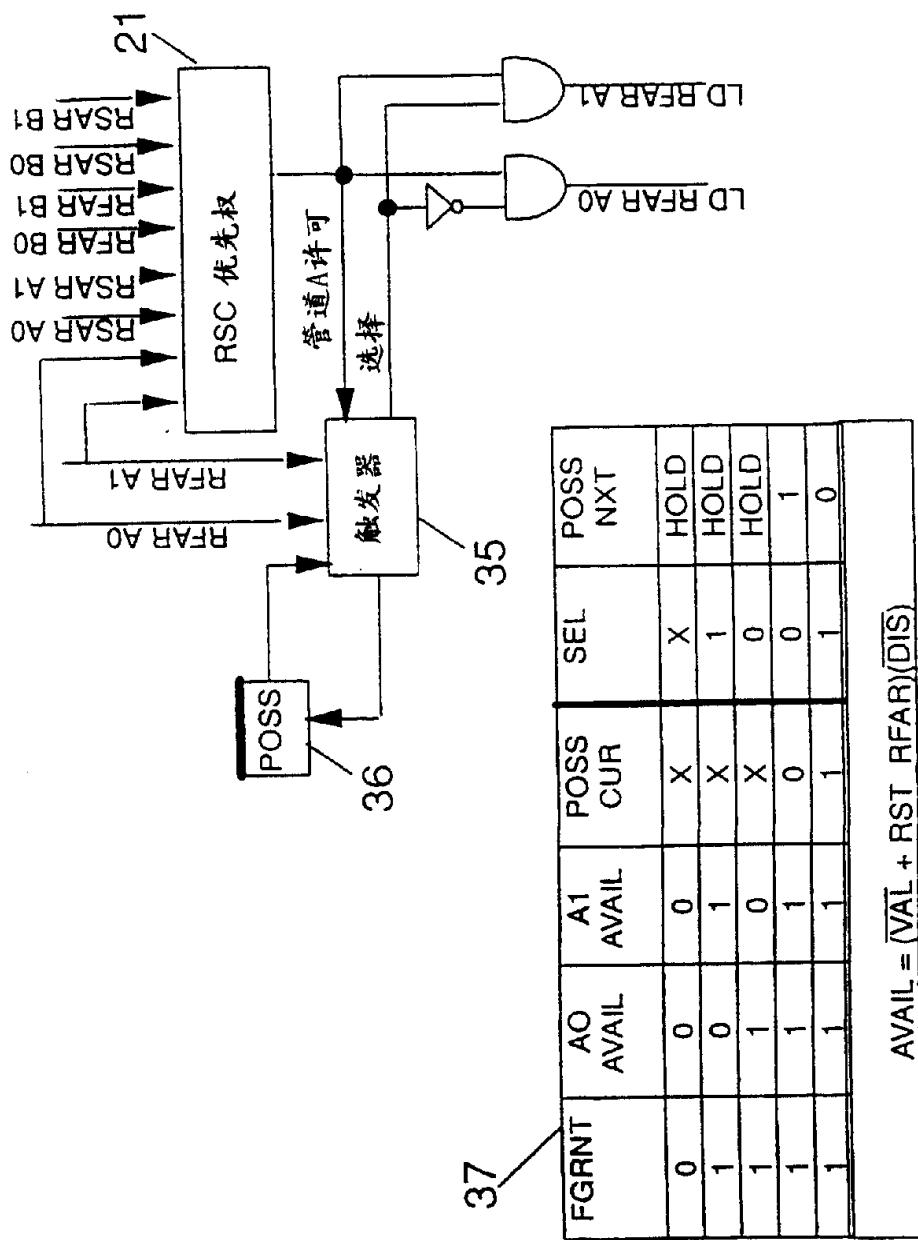
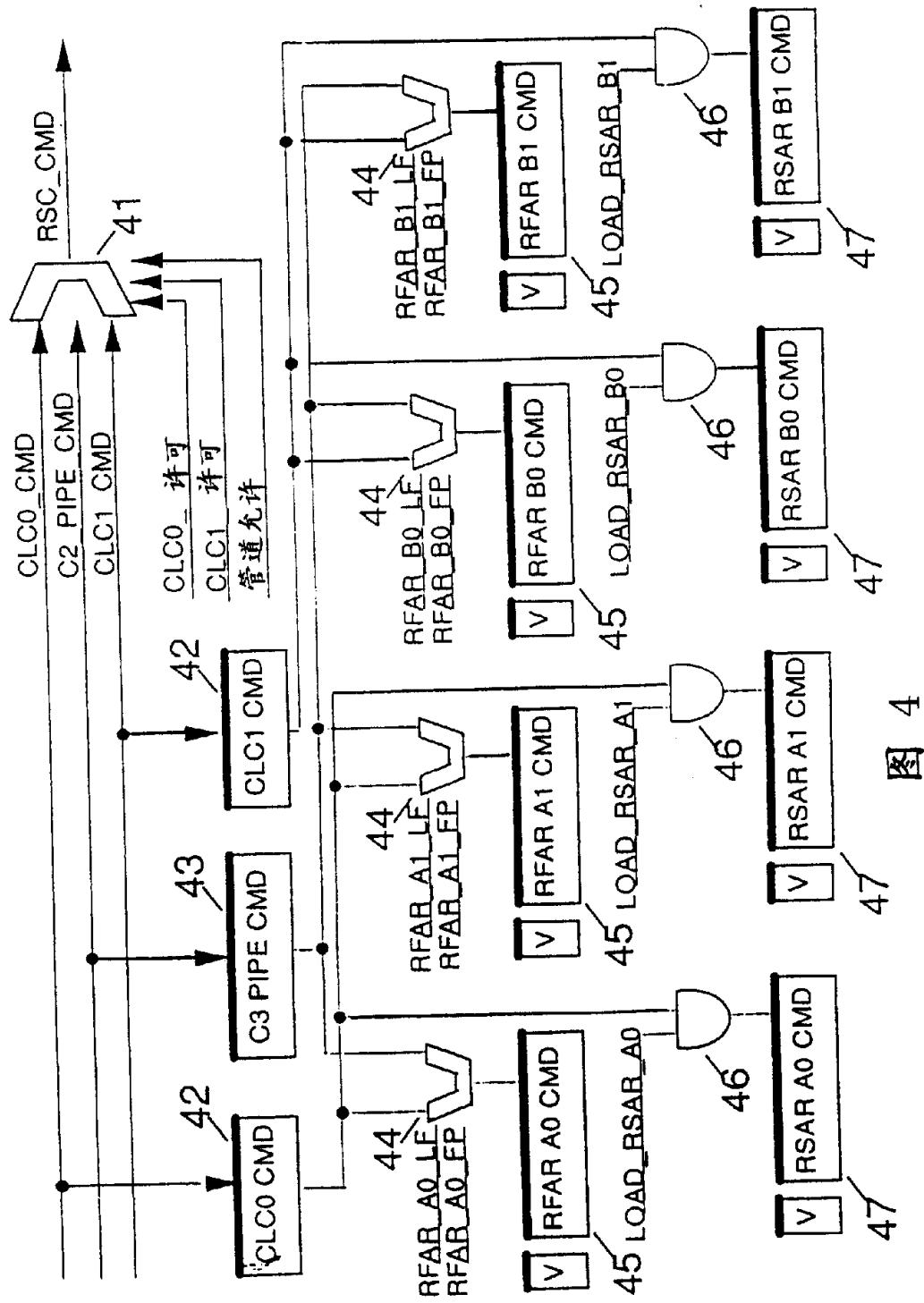


图 3A

图 3B





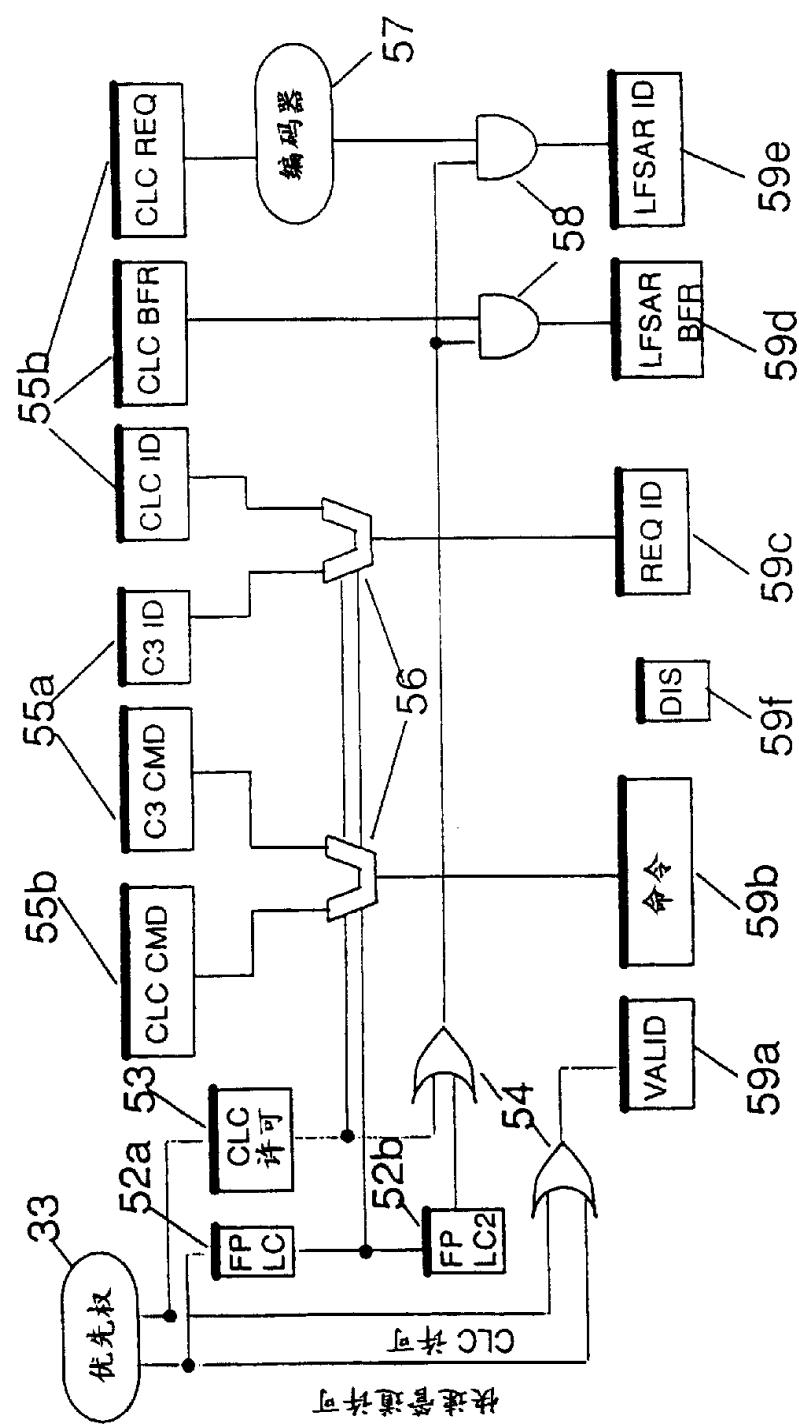


图 5

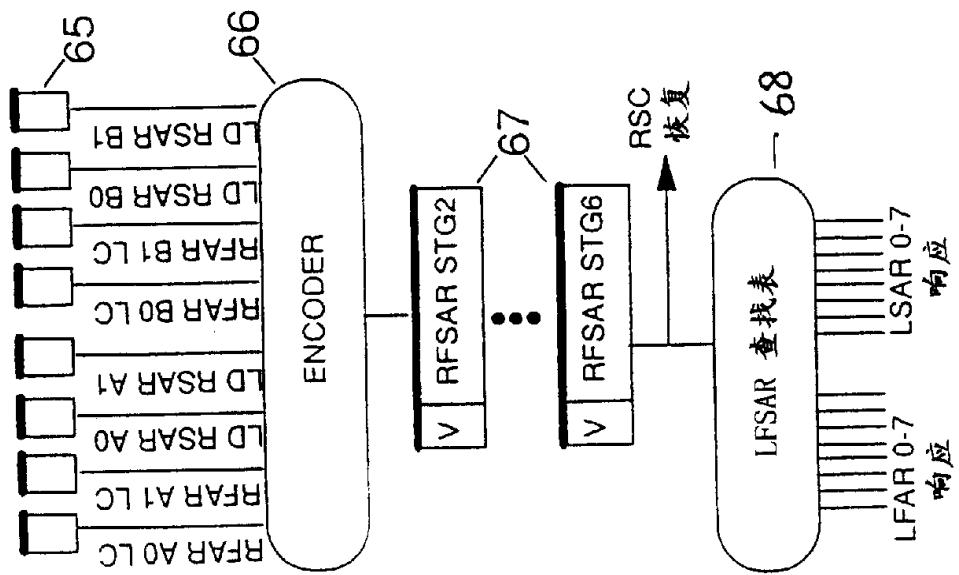
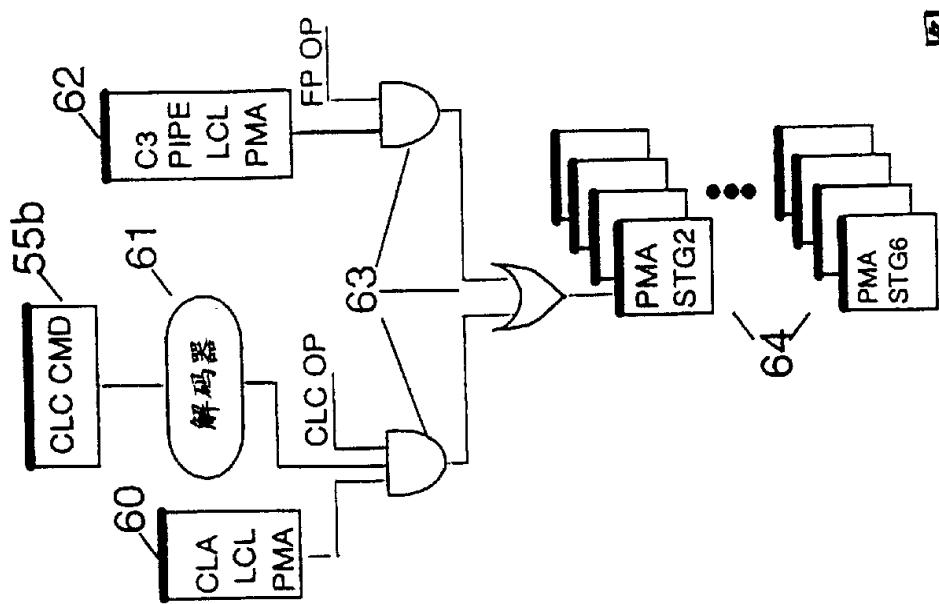


图 6



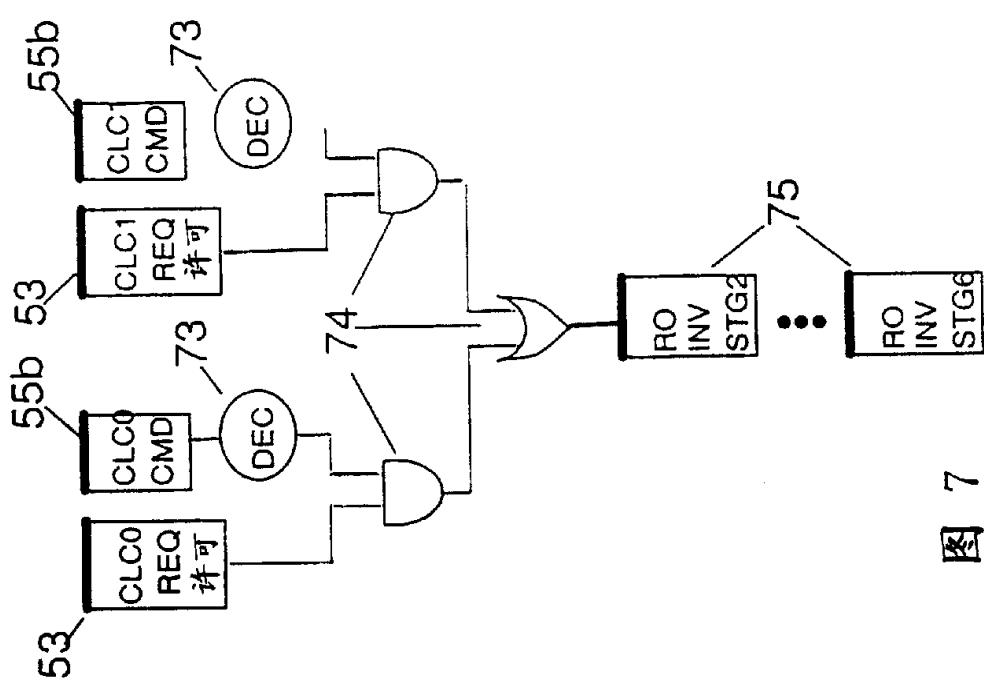


图 7

2000-03-21

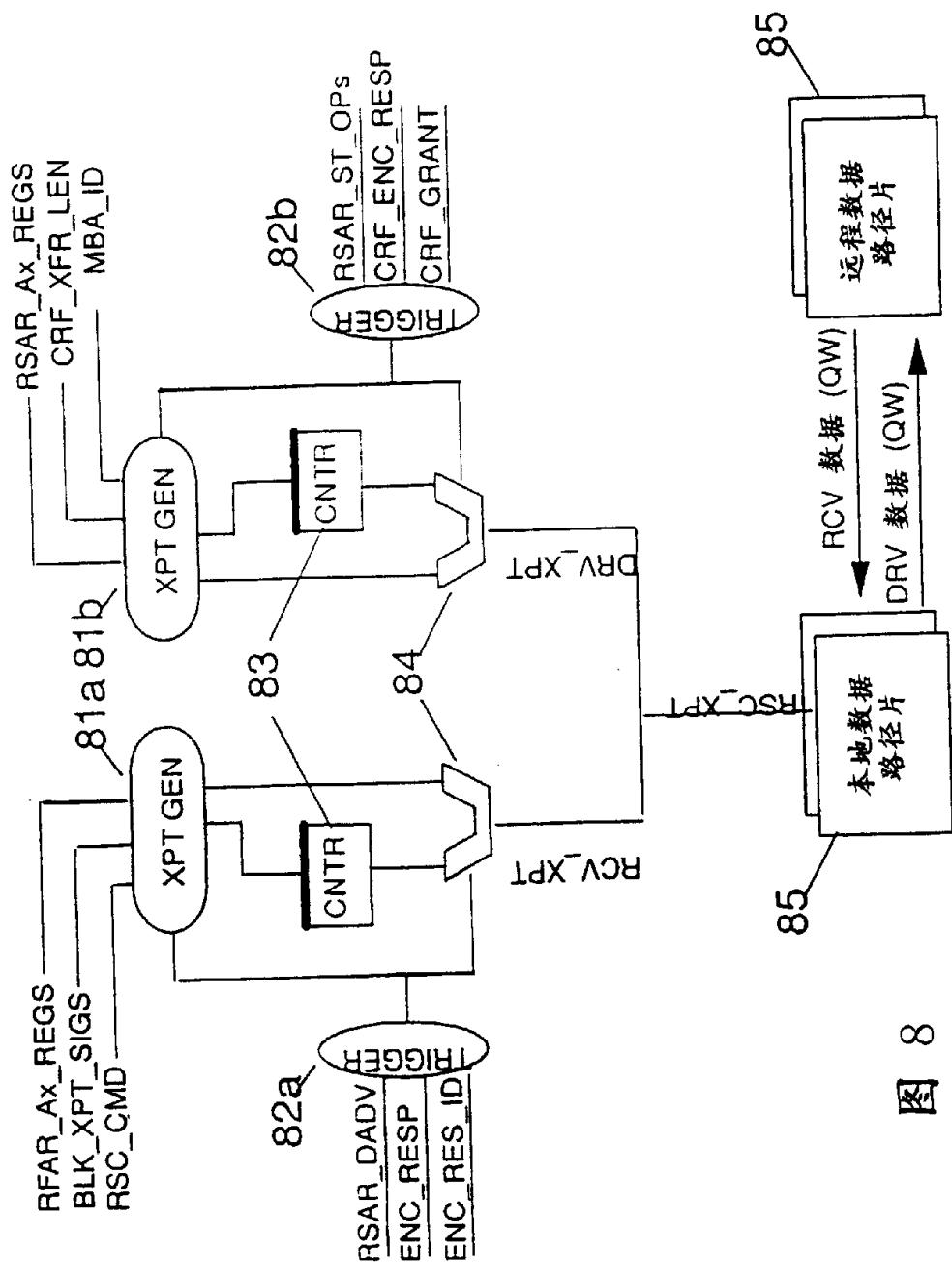


图 8