(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101305354 B
(45)授权公告日 2011.08.31

(21)申请号 200680037568.4

(22)申请日 2006.08.11

(30)优先权数据
11/201,581 2005.08.11 US

(85)PCT申请进入国家阶段日
2008.04.09

(86)PCT申请的申请数据
PCT/US2006/031520 2006.08.11

(87)PCT申请的公布数据
WO2007/022018 EN 2007.02.22

(73)专利权人 苹果公司
地址 美国加利福尼亚

(72)发明人 西达·P.·苏布拉马尼昂
詹姆斯·B.·凯勒 乔治·孔·游
吕奇·沃德万 雷米希·冈纳

(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 李颖

(51)Int.Cl.
G06F 13/40(2006.01)
G06F 13/362(2006.01)

(56)对比文件
US 2003/0065843 A1,2003.04.03, 说明书0004至0037段、图1-3.
EP 1308862 A1,2003.05.07, 全文.
PCI Express Base Specification Revision 1.0a.PCI-SIG,2003, 全文.

审查员 魏峰

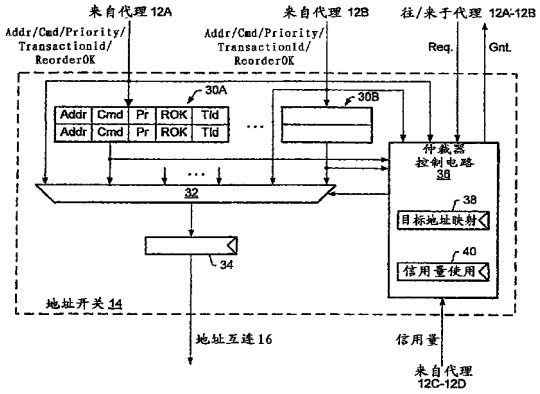
权利要求书 2 页 说明书 9 页 附图 5 页

(54)发明名称

在多个请求之间进行仲裁的系统和方法

(57)摘要

在一个实施例中,开关被配置成与互连耦接。所述开关包括多个存储位置和与所述多个存储位置耦接的仲裁器控制电路。所述多个存储位置被配置成保存由多个代理传送的多个请求。所述仲裁器控制电路被配置成在保存于所述多个存储位置中的多个请求之间进行仲裁。选择的请求是仲裁的获胜者,所述开关被配置成把选择的请求从所述多个存储位置之一传送到所述互连上。在另一实施例中,系统包括多个代理,一个互连和与所述多个代理和互连耦接的开关。在另一实施例中,设想了一种方法。



CN 101305354 B

1. 一种在多个请求之间进行仲裁的系统，包括：

多个代理；

一个互连，其包括通信介质；和

与所述多个代理及互连耦接的开关，其中所述开关包括多个存储位置，其中所述多个存储位置被配置成保存由所述多个代理传给所述开关的多个请求，以及其中所述开关被配置成在保存于所述多个存储位置中的多个请求之间进行仲裁，以及其中所述开关被配置成在所述互连上传送选择的请求，其中选择的请求是仲裁的获胜者；

其中所述互连包括耦接在所述开关和在所述互连上接收请求的多个代理中的每一个之间的一个或多个定时存储装置，其中所述一个或多个定时存储装置的数目以请求到离所述开关最远的接收代理的飞行时间为基础，并且其中在所述互连上到达最远的接收代理的飞行时间超过所述多个定时存储装置的一个时钟的时钟周期，以及其中在所述开关和所述多个代理中的每一个代理之间设置相同数目的定时存储装置，即使某些代理物理上离所述开关更近且请求物理上能够在更短的飞行时间内到达更近的代理。

2. 按照权利要求 1 所述的系统，其中所述多个存储位置包括多个队列，其中所述多个队列中的每个队列对应于所述多个代理中的一个相应代理，并被配置成保存由所述相应代理传送的请求，其中所述多个队列中的每个队列包括所述多个存储位置中的至少两个。

3. 按照权利要求 1 所述的系统，其中所述开关被配置成在选择第二请求之前，把由所述多个代理中的第一代理传送的第一请求选为选择的请求，其中第二请求由第一代理在第一请求之前传送。

4. 按照权利要求 3 所述的系统，其中每个请求具有对应的优先级，以及其中所述开关被配置成如果第一请求的优先级高于第二请求的话，则在选择第二请求之前选择第一请求。

5. 按照权利要求 3 所述的系统，其中所述开关被配置成如果第一和第二请求可按照一组排序规则重新排序，那么在选择第二请求之前选择第一请求，并且其中所述开关被配置成如果即使第一请求的优先级高于第二请求，第一和第二请求也不能按照所述一组排序规则重新排序，那么不在第二请求之前选择第一请求。

6. 按照权利要求 5 所述的系统，其中第一代理被配置成随同第一请求传送表示第一请求是否可与第二请求重新排序的指示，以及其中所述开关被配置成如果所述指示表示第一请求可与第二请求重新排序，那么在选择第二请求之前选择第一请求。

7. 按照权利要求 1 所述的系统，其中所述开关被配置成确定所述多个请求中的每个请求的多个代理的目标代理，以及其中如果根据所述多个请求的一个请求的目标代理、所述请求被阻塞，那么所述开关被配置成把给另一目标代理的另一请求选为所选请求。

8. 按照权利要求 7 所述的系统，其中所述开关被配置成对所述多个请求中的每个请求的地址的一部分解码，以确定每个请求的目标代理。

9. 一种在多个请求之间进行仲裁的方法，包括：

使来自多个代理的请求在多个存储位置中排队；

在所述多个存储位置中的请求之间进行仲裁，以选择所述多个请求中的所选请求；和

在包括通信介质的互连上传送选择的请求，其中所述互连包括耦接在所述开关和在所述互连上接收请求的多个代理中的每一个之间的一个或多个定时存储装置，以及其中所述

一个或多个定时存储装置的数目以请求到离所述开关最远的接收代理的飞行时间为基础，并且其中在所述互连上到达最远的接收代理的飞行时间超过所述多个定时存储装置的一个时钟的时钟周期，以及其中在所述开关和所述多个代理中的每一个代理之间设置相同数目的定时存储装置，即使某些代理物理上离所述开关更近且请求物理上能够在更短的飞行时间内到达更近的代理。

在多个请求之间进行仲裁的系统和方法

技术领域

[0001] 本发明涉及集成电路的领域，更具体地说，涉及集成电路内和／或集成电路之间的互连的仲裁机构。

[0002] 系统中的集成电路，或者集成电路内的各种电路一般需要相互通信。许多情况下，系统／集成电路中的通信器可通过存储器映射中的各个地址通信。即，各个通信器被分配存储器映射中的地址，对所述地址的读取／写入被用于通信。一般来说，这样的通信器使用在通信器之间的互连上传送的读取／写入事务。例如，常见的是具有一条地址总线，在所述地址总线上，传送地址、命令和其它事务信息以发起事务。另外，可使用数据总线来传送与事务对应的数据（如果有的话）。如果为事务实现高速缓存一致性，那么可以提供一个响应接口以便按照由通信器实现的一致性方案，保持一致状态。

[0003] 在互连或其一部分在通信器之间被共享的条件下，需要关于互连的使用在通信器之间进行仲裁的某种机制。过去，使用的是集中式和分布式仲裁机制。在集中式仲裁机制中，所有通信器向中央仲裁器传送请求信号，中央仲裁器确定哪个通信器被准许使用互连（“仲裁获胜者”）。中央仲裁器向获得批准的通信器返回许可信号，获得批准的通信器随后在互连上驱动其事务。分布式仲裁方案中，每个通信器实现一个本地仲裁器（或者附近包括一个本地仲裁器）。每个通信器向所有本地仲裁器宣称其请求信号。本地仲裁器用于独立地确定相同的仲裁获胜者。获得批准的通信器的本地仲裁器通知该获得批准的通信器，该获得批准的通信器把其事务驱动到互连上。

[0004] 与分布式仲裁机制相比，集中式仲裁机制一般实现起来较简单。 但是，集中式仲裁机制一般来说也是一种等待时间较长的机制。集中式仲裁机制包括可能较长距离的请求信号传输，随后是同样距离的许可信号传输，接下来是得到批准的通信器驱动其事务。另一方面，更复杂的分布式仲裁机制只涉及一次长距离传输（对每个本地仲裁器的请求信号传输）。分布式仲裁机制中的复杂性一般包括关于特定通信器的许可的更复杂“停车(parking)”，控制通信器的流程的复杂性，以及以每个源通信器为基础的目标通信器中的缓冲器的分配。

发明内容

[0005] 在一个实施例中，开关（swtich）被配置成与互连耦接。所述开关包括多个存储位置和与所述多个存储位置耦接的仲裁器控制电路。所述多个存储位置被配置成保存由多个代理传送的多个请求。仲裁器控制电路被配置成在保存于所述多个存储位置中的多个请求之间进行仲裁。选择的请求是仲裁的获胜者，开关被配置成把选择的请求从所述多个存储位置之一传送到所述互连上。在另一实施例中，系统包括多个代理，一个互连，和与所述多个代理及互连耦接的开关。

[0006] 在另一实施例中，方法包括使来自多个代理的请求在多个存储位置中排队；在所述多个存储位置中的请求之间进行仲裁，以选择所述多个请求中的一个请求；并在互连上传送选择的请求。

附图说明

[0007] 下面的说明参考了附图，现在简要说明这些附图。

[0008] 图 1 是集成电路的一个实施例的方框图。

[0009] 图 2 是图 1 中所示的仲裁器 / 地址开关的一个实施例的方框图。

[0010] 图 3 是图解说明图 2 中所示的在请求之间进行仲裁的仲裁器控制单元的一个实施例的操作的流程图。

[0011] 图 4 是图解说明一个实施例的排序规则的表。

[0012] 图 5 是在地址互连上传递请求的方法的高级流程图。

[0013] 尽管本发明容许各种修改和备选形式，不过附图中举例表示了本 发明的具体实施例，并且这里将详细说明所述具体实施例。但是，附图及其详细说明显然并不意图把本发明局限于公开的特定形式，相反，本发明将包括落入由附加权利要求限定的本发明的精神和范围内的所有修改、等同物和替换物。

具体实施方式

[0014] 现在参见图 1，图中表示了系统 10 的一个实施例的方框图。在图解说明的实施例中，系统 10 包括多个代理，比如代理 12A-12D。系统还包括地址开关 14，地址互连 16 和响应 / 数据互连 18。代理 12A-12B 与地址开关 14 耦接（在图解说明的实施例中，代理 12B 通过触发器 (flop)20A 被耦接）。地址开关 14 还与地址互连 16 耦接，地址互连 16 与代理 12A-12D 耦接（在图解说明的实施例中，通过触发器 20B-20I）。按照另一种方式来看，触发器 20B-20I 可以是地址互连 16 的一部分。代理 12A-12D 还与响应 / 数据互连 18 耦接。在一个实施例中，系统 10 可被集成到单一的集成电路芯片上。在其它实施例中，系统 10 的各个组件可在独立的集成电路上实现。在各个实施例中，可以使用任意级别的集成。

[0015] 代理 12A-12B 被配置成把将在地址互连 16 上传送的请求传送给地址开关 14。每个请求可包括事务的地址和命令（所述命令识别待执行的事务）。可以支持各种命令，例如相关 (coherent) 读取和写入命令，非相关读取和写入命令，相关所有权命令，探测命令，同步命令，高速缓存管理命令等等。在各个实施例中，请求还包括其它信息。例如，在下面更详细说明的一个实施例中，请求可包括请求的优先级（供仲裁之用），和该请求的数据是否将被复制到 2 级高速缓存的指示。

[0016] 代理 12A-12B 可被称为源代理，因为它们可通过传送对地址互连 16 的请求，在系统中发起事务。例证的源代理可包括处理器，外部回写式高速缓存（它向已被改变成内存的写入收回 (write evicted) 高速缓存块发起 (source) 写入事务），和输入 / 输出 (I/O) 桥接器（它代表它们耦接的外围设备发起事务）。如图 1 中的省略号所示，各个实施例中 可包括两个以上的源代理（或者下面说明的源 / 目标代理）。其它代理可能不发起 (source) 事务，但是可以是事务的目标（即，接收事务并且对事务的数据负责的代理）。这样的代理被称为目标代理。对于读取事务来说，目标代理提供数据，除非另一代理具有数据的最新（修改的）高速缓存副本。对于写入事务来说，目标代理接收源代理供给的写入数据。例如，目标代理可包括存储器控制器和 I/O 桥接器。一些代理可以既是某些事务的源代理，又是其它事务的目标代理。例证的源 / 目标代理可包括上面提及的 I/O 桥接器或者外部高速

缓存。一般来说，代理可包含被配置成借助地址互连 16 和响应 / 数据互连 18 上的事务进行通信的任何电路。

[0017] 每个源代理 12A-12B（或者源 / 目标代理，不过在本说明中为了简洁起见，将使用源代理）可使用请求信号来指示源代理 12A-12B 正在传送请求。地址开关 14 还可向指定的源代理 12A-12B 断言许可信号，以指示源代理 12A-12B 传送的对地址互连 16 的请求已被准许。

[0018] 地址开关 14 可包括多个存储位置，所述存储位置被配置成保存源代理传送的请求，直到对地址互连 16 的请求被准许为止。在一个实施例中，存储位置可包含多个队列。每个队列可对应于特定的源代理，并且可专用于保存由该源代理传送的请求。即，在队列和源代理之间存在一一对应。指定源代理的队列可保存由该指定源代理传给地址开关 14 的多个请求。每个源代理可以知道与该源代理对应的队列中的队列项的数目，不能传送比队列项更多的请求。

[0019] 地址开关 14 还可被配置成在队列中的请求之间进行仲裁，以选择在地址互连 16 上进行传输的请求。可以采用任何仲裁方案。例如，在一些实施例中，每个请求可具有分配给它的优先级。仲裁方案可以是具有饥饿预防机制的严格优先级方案（选择优先级最高的请求），以避免使优先级较低的请求饿死。地址开关 14 可在地址互连 16 上驱动选择的请求。

[0020] 从而，地址开关 14 可对地址互连 16 采用集中式仲裁。但是，由于请求被传给地址开关 14，并由地址开关 14 驱动到地址互连 16 上， 因此在一些实施例中，与向选择的在仲裁中获胜的源代理返回许可（和源代理响应所述许可，驱动地址互连 16）关联的等待时间可被缩短。地址开关 14 可并行于把选择的请求驱动到地址互连 16 上，向源代理返回许可。另外，在一些实施例中，地址开关 14 中的仲裁电路具有和请求有关的更多信息，因为请求本身在地址开关 14 中排队（例如，与一般在常规的集中式仲裁器中实现的请求 / 许可结构相比）。

[0021] 当源代理从地址开关 14 收到许可时，源代理被告知某一队列项可用于保存另一请求。在一个实施例中，可按照传送顺序准许来自指定源代理的请求。从而，收到许可的源代理可使所述许可与对应的请求联系起来。在其它实施例中，地址开关 14 可被配置成在一些情况下对请求重新排序（在较早从源代理接收的请求之前传送稍后从相同源代理接收的请求）。在这样的实施例中，源代理可与地址互连 16 耦接，并且可接收传紧接着传送的请求，以确定哪个请求被准许。例如，在一些实施例中，源代理可用源标记标识每个请求，源代理可从地址互连 16 接收源标记，以确定哪个请求被准许。

[0022] 在各个实施例中，地址互 16 可包括任何通信介质。例如，地址互连 16 可包括分组接口，其中在地址互连 16 上在一个或多个时钟周期内以分组的形式传送请求。特别地，在一个实施例中，可在地址互连 16 上在一个时钟周期内传送地址分组。这样的实施例可使地址开关 14 多少与事务的地址阶段的协议隔离。其它实施例可被地址互连 16 实现成总线，地址连同各种控制信号一起被传送，以指示在地址阶段内传送的命令和其它控制信息。

[0023] 请求被广播给地址互连 16 上的代理 12A-12D。在一些实施例中，在地址互连 16 上到达最远的代理 12A-12D（就物理距离来说）的飞行时间可能超过与地址互连 16 关联的时钟的一个时钟周期。包括在地址开关 14 和指定代理 12A-12B 之间的触发器 (Flop)20B-20I

的数目可以到最远的代理的飞行时间（以用于地址互连 16 的时钟信号的时钟周期的数目表示）为基础。在图解说明的实施例中，飞行时间超过两个时钟周期，从而使用两个触发器。其它实施例可以包括零个触发器（如果飞 行时间小于一个时钟周期），一个触发器（如果飞行时间超过一个时钟周期，但是小于两个时钟周期），或者两个以上的触发器（取决于飞行时间）。为了确保指定的请求逻辑上在相同的时钟周期被每个代理 12A-12D 接收，即使某些代理物理上离地址开关 14 更近，请求能够在更短的飞行时间内到达所述更近的代理，仍然在地址开关 14 和每个代理 12A-12D 之间设置相同数目的触发器 20B-20I。到更远的代理的触发器可沿地址开关 14 和所述更远的代理之间的距离物理分布。为了简化附图，图 1 并不试图图解说明触发器 20B-20I 的物理分布。

[0024] 由于每个代理 12A-12D 逻辑上在相同的时钟周期接收在地址互连 16 上传送的请求，在一些实施例中，地址互连 16 可以是相关 (coherent) 事务的空间相关点。即，在地址互连 16 上成功传送的请求的顺序可为了相关性的目的定义事务的顺序。

[0025] 类似地，在一些实施例中，请求从源代理 12A-12B 到地址开关 14 的飞行时间可能超过一个时钟周期。在一些实施例中，地址开关 14 可被物理布置成最接近预期具有最高的请求带宽的源代理（例如，处理器代理一般具有比高速缓存代理和 I/O 代理更高的请求带宽）。在图 1 中的实施例中，来自源代理 12B 的请求的飞行时间可超过一个时钟周期，从而触发器 20A 可被用于捕获所述请求，并继续所述请求对地址开关 14 的传送。类似地，地址开关 14 返回的许可信号可被触发器 20A 捕捉，并在下一时钟周期传送。

[0026] 在本实施例中，由于地址互连 16 是相关事务的相关点（也可整体定义请求的顺序），因此在从不同的代理传给地址开关 14 的请求之中不存在任何排序。因此，如果诸如触发器 20A 之类的触发器被用于自一个源代理的飞行时间，那么对于其请求的飞行时间小于一个时钟周期的其它代理来说，不需要插入触发器。

[0027] 如上所述，在一些实施例中，源代理可在地址互连 16 上接收请求，以确定实际向地址互连 16 批准停留在地址开关 14 中的来自指定代理的多个请求中的哪个请求。另外，在一些实施例中，也可高速缓存数据（从而可以参与相关事务）的源代理也可出于相关性的目的在地 址互连 16 上探听其它源代理的请求。诸如代理 12C-12D 的目标代理与地址互连 16 耦接，以便接收以它们为目标的请求。

[0028] 在一个实施例中，地址开关 14 也可被配置成管理对各个目标代理 12C-12D 的流控制。例如，地址开关 14 可被配置成确定每个请求寻址哪个目标代理（例如，借助请求地址的粗粒 (coarse grain) 解码，并根据所述解码把请求地址映射到目标代理）。地址开关 14 可以知道在目标代理中排队的请求的数目（在从地址互 (16) 收到请求之后），并且可以确保目标代理的输入队列不会由于请求而溢出。如果给定请求以其输入队列已满的指定目标代理为目标，那么地址开关 14 可确保该给定请求不会被选作仲裁的获胜者，直到在给定的目标代理中，一个输入队列项可用为止。在这种情况下，地址开关 14 可以不阻塞其它请求。即，即使由于目标代理不能接收请求，较早的请求或较高优先级的请求没有资格赢得仲裁，地址开关 14 仍然能够选择以另一目标代理为目标的另一请求。在一些实施例中，地址开关 14 也可努力在源代理之中实现对目标代理的公平或优化访问。

[0029] 代理 12A-12D 也可与响应 / 数据互 18 耦接，以便传递在地址互连 16 上借助请求发起的事务的响应阶段和数据阶段。在各个实施例中，一些事务可能不包括数据阶段。响

应阶段可包括来自高速缓存代理的对相关事务的响应。所述响应可提供在与事务对应的数据的接收器中，应建立哪种相关状态。事务的数据阶段涉及对源代理的数据传输（用于读取）或者对目标代理的数据传输（用于写入）。在各个实施例中，响应 / 数据互连 18 可包括一个通信介质。

[0030] 尽管在上面的一些实施例中，地址开关 14 中的存储位置被描述成每个代理的队列，不过其它实施例可按照其它方式实现存储位置。例如，存储位置可以是源代理把请求保存于其中的单一队列。队列项可由地址开关 14 灵活地分配给源代理，存在指示每个源代理可用的队列项的数目的其它信令（例如，从地址开关 14 到每个代理的指示队列项的数目的信令，或者至少一个另外的队列项可供该代理之用的信令等等）。代理可被分组，并且可以共用队列，或者可根据每个请求的 目标分配队列。

[0031] 注意尽管在图 1 的实施例中图解说明了触发器 20A-20I，不过任何定时存储装置可被用作所述装置 20A-20I。例如，可以使用寄存器、锁存器等等。定时存储装置可以包括配置成响应时钟信号，捕捉供存储的数值的任何存储装置。在本实施例中，触发器 20A-20I 的时钟信号输入可以是用于地址互连 16 的时钟。一些代理在内部可以时钟的倍数工作。也可用任何定时存储装置实现这里描述的其它触发器。一般来说，每个触发器 20A-20I 具有与其输入的宽度相等的位宽。例如，触发器 20A 可以是与地址开关 14 的请求 / 准许接口的宽度，触发器 20B-20I 的宽度可以是地址互连 6 的宽度。

[0032] 虽然上面严格优先级仲裁方案被用作一个例子，不过其它实施例可以实现其它仲裁方案。例如，其它仲裁方案可以包括循环法、优先级加权循环法、循环方案和优先级的组合等等。

[0033] 虽然上面的讨论涉及的是接收对地址互连 16 的请求，并在所述请求之中进行仲裁，以确定将在地址互连 16 上传送的仲裁获胜者的地址开关，不过其它实施例可以实现用于数据互连的类似开关。对数据开关的请求可以接收自数据源。请求可包括事务的数据，以及识别数据所对应的地址请求的标记。请求可包括优先级，在各个实施例中，所述优先级可以是和地址请求相同的优先级，或者是不同的优先级。数据开关可在排队的数据请求中进行仲裁，以选择在数据互连上驱动的仲裁获胜者。

[0034] 现在参见图 2，图中表示了地址开关 14 的一个实施例的方框图。在图解说明的实施例中，地址开关 14 包括队列，比如分别对应于代理 12A 和 12B 的队列 30A 和 30B。地址开关 14 还包括多路复用器 32，输出触发器 34 和仲裁器控制电路 36。队列 30A 和 30B 被耦接，以便从它们各自的代理 12A-12B 接收请求。另外，在图解说明的实施例中，仲裁器控制电路 36 和多路复用器 32 被耦接，以接收所述请求。队列 30A 和 30B 还与仲裁器控制电路 36 和多路复用器 32 耦接。在包括另外的源代理的实施例中，多路复用器 32 还可被耦接成接收所述请求 和地址开关 14 中的对应队列的输出。仲裁器控制电路 36 被耦接成接收来自代理 12A-12B 的请求信号，并向代理 12A-12B 提供许可信号。仲裁器控制电路 36 还被耦接成提供对多路复用器 32 的选择控制。多路复用器 32 的输出端与输出触发器 34 耦接，输出触发器 34 再与地址互连 16 耦接。仲裁器控制电路 36 还被耦接成接收来自代理 12C-12D（在一些实施例中，其它目标代理或源 / 目标代理）的信用指示 (credit indication)。

[0035] 在图解说明的实施例中，每个请求包括地址 (Addr)，命令 (Cmd)，优先级 (Pr)，事务 ID(TId) 和 ReorderOK 位 (ROK)。队列 30A-30B 可被配置成保存所述请求。即，每个队列

项可包括足以保存请求的存储空间。在图解说明的实施例中，每个队列 30A-30B 包括两项。在其它实施例中，在每个队列 30A-30B 中可以包括更多的项。地址是存储器映射中受所述请求影响，并且识别目标代理的地址。命令识别正在发起的事务。优先级指示请求的优先级。在一个实施例中，使用三个优先级。当请求的等待时间至关重要时，使用最高优先级。例如，在一个实施例中，当与分组接口，比如以太网接口耦接的一些 I/O 桥接器或者驱动分组接口的电路中的缓冲器不断减少时，所述 I/O 桥接器可以使用最高优先级来读取描述符或者分组数据。中等优先级可由 I/O 桥接器用于处理器读取和直接存储器访问 (DMA) 描述符读取。如果源代理的写缓冲器正在接近于充满，那么中等优先级也可被用于写入。低优先级可被用于所有其它请求（例如，处理器写入，诸如 DAM 读取和写入之类的高带宽读取和写入等等）。其它实施例可以使用更多或更少的优先级。reorderOK 位可被用于指示该请求是否可被重新安排在来自相同代理的某一在先请求之前，如果所述在先请求仍然在代理的队列 30A-30B 中（即，所述在先请求还未被准许到地址互连 16 上）。源代理可以按照由代理实现的一组排序规则产生 reorderOK 位。图 4 中图解说明了一个例子，并在下面更详细地讨论。

[0036] 当向地址开关 14 传送请求时，代理 12A-12B 可宣称对应的请求信号。好，宣称的请求信号可以充当请求的有效位，用于写入与代理 12A-12B 对应的队列 30A-30B，以及用于向仲裁器控制电路 36 指示该请求。仲裁器控制电路 36 还可产生许可信号（每个源代理 12A-12B 一个许可信号）。仲裁器控制电路 36 可向指定的源代理 12A-12B 宣称许可信号，以指示来自源代理 12A-12B 的请求已被准许，并将在地址互连 16 上被驱动。宣称的许可信号可向代理 12A-12B 指示代理的队列 30A-30B 中的一个队列项正在释放以接受另一请求。

[0037] 每个代理 12A-12B 可被配置成传送数目与其队列 30A-30B 中的队列项的数目（在图解说明的实施例中，两个）相同的请求。在一个实施例中，每个代理 12A-12B 可把停留在队列 30A-30B 中的请求的数目限制为项数。即，代理可传送两个请求，随后禁止传送另外的请求，直到宣称的许可信号指示一个队列项正在被释放为止。在另一实施例中，每个代理 12A-12B 可填充其队列 30A-30B，并传送又一个请求，代理 12A-12B 可继续传送所述又一个请求，直到前一请求被准许，从而宣称的请求被写入队列项中为止。

[0038] 仲裁器控制电路 36 可仲裁队列 30A-30B 中的请求，并选择将在地址互连 16 上传送的请求。仲裁器控制电路 36 可产生对多路复用器 32 的选择控制，以选择请求，并把选择的请求提供给输出触发器 34。输出触发器 34 把请求驱动到地址互连 16 上。可提供输出触发器 34，以保证在时钟周期开始时，请求被驱动到地址互连 16 上。在其它实施例中，可以除去输出触发器 34，可依据通过多路复用器 32 的选择驱动请求。选择的请求同样从其队列 30A-30B 中被删除，仲裁器控制电路 36 向对应的源代理 12A-12B 宣称许可信号。

[0039] 仲裁器控制电路 36 可实现任意仲裁方案在请求中进行选择。例如，如上所述，仲裁器控制电路 36 可实现具有饥饿预防机制的严格优先级选择。在这种方案中，最高优先级请求通常被选为仲裁的获胜者。但是，如果众多的较高优先级请求导致较低优先级请求长时间停留在队列 30A-30B 中（即，使较低优先级请求“饥饿”），那么可以选择较低优先级请求。可按照各种方式实现饥饿预防机制。例如，每个请求可具有与之相关的计时器或者时间戳记，所述计时器或者时间戳记 指示该请求在队列 30A-30B 中的时间有多长。如果该请求在队列 30A-30B 中的时间超过时间阈值（所述时间阈值可以是固定的或者可编程的），那

么可以选择该请求。实际上，请求的优先级可因其在队列 30A-30B 中的时间而增大。在另一例子中，如果连续选择规定数目的较高优先级请求（这里所述数目可以是固定的或者可编程的），那么可自动选择一个较低优先级请求。如果对于指定仲裁，在队列 30A-30B 中，一个以上的请求具有最高优先级，那么可以使用任何机制在请求中进行选择（例如，源代理之间的固定优先级，源代理之间的循环法，可以选择最老的请求等）。其它实施例可以实现其它仲裁方案（例如，无优先级的循环法，基于优先级的加权循环法等等）。

[0040] 如果在队列 30A-30 中，高优先级请求在另一请求之后，那么如果该高优先级请求的 RrorderOK 位未被设置成指示允许把该请求重新安排在前一请求之前，那么该高优先级请求没有仲裁资格。即，如果 RrorderOK 位不指示允许重新排序，那么仲裁器控制电路 36 不会在相同队列中的前一较低优先级请求之前选择该高优先级请求。如果如 RrorderOK 位所示，允许在在先请求之前的重排，那么当较高优先级请求在队列中的较低优先级请求之后时，可选择高优先级请求。即，在传送高优先级请求之前，较低优先级请求可能已由代理传给地址开关。

[0041] 在一些实施例中，仲裁器控制电路 36 还可实现对每个目标代理（例如，在图 1 的实施例中，代理 12C-12D）的流量控制。仲裁器控制电路 36 可确定每个请求的目标代理（例如，在本实施例中利用地址）。就地址到目标代理的映射来说，仲裁器控制电路 36 是可编程的。例如，一个或多个寄存器 38 可被编程，以把地址空间映射到目标代理。根据地址映射，地址控制电路 36 可执行某些最高有效位的粗粒解码，以确定目标代理。在本实施例中，所述解码是粗粒的，因为预期将对相同的目标代理规划较大的连续地址范围。其它实施例可以使用细粒解码。此外，尽管在本实施例中，解码是可编程的，不过其它实施例可以具有固定的地址映射，仲裁器控制电路 36 可按照固定的地址映 射解码地址。

[0042] 每个目标代理具有接受高达一定数目的事务的能力（例如，与在目标代理中实现的缓冲器的数目一致）。在一些实施例中，事务可依据事务类型分组，可关于每个目标代理规定每组事务的数目。例如，在一个实施例中，事务可被分组成相关读取，相关写入，非公告 (non-posted) 非相关命令，和公告的非相关命令。对于上述各组事务中的每一组，每个目标代理可以实现一定数目的缓冲器。

[0043] 仲裁器控制电路 36 可被配置成实现对目标代理的流量控制，以保证目标代理的缓冲器不会溢出。例如，可以使用基于信用量的系统，其中每个缓冲器由对应事务类型的信用量表示。仲裁器控制电路 36 可跟踪可用的信用量（例如，使用图 2 中的一个或多个寄存器 40）。如果仲裁器控制电路 36 选择指定类型的以指定目标代理为目标的请求，那么仲裁器控制电路 36 可把对应的信用计数减 1。当缓冲器空闲时，目标代理也可传达信用量的恢复（图 2 中表示成信用量）。从而，在任意指定时刻，仲裁器控制电路 36 知道对于每种事务类型，每个目标代理中的缓冲器可用性。如果某一请求会消耗的信用量不存在，那么仲裁器控制电路 36 会防止选择该请求。改为选择另一请求（甚至较低优先级的请求），如果所述另一请求的对应信用量可用的话。

[0044] 在一些实施例中，仲裁器控制电路 36 可努力保证源代理对指定目标代理的访问的公平性。仲裁器控制电路 36 可跟踪每个目标代理的各种信用量的总体使用，以及每个源代理对信用量的使用。如果目标代理的信用量的总体使用较高（表示目标代理忙于事务），并且特定的源代理正在以较高的速率与目标代理通信（由该目标代理的信用量的使用指

示），那么仲裁器控制电路36可限止该对源/目标代理对信用量的使用，以允许其它源代理更好地访问该目标代理。

[0045] 如图2的实施例中所示，多路复用器32和仲裁器控制电路36被耦接，以接收代理12A-12B当前传给地址开关14的请求。仲裁器控制电路36可被配置成关于指定的请求绕过队列30A-30B，并通过多路复用器32选择该请求，如果当传送该请求时队列30A-30B为空（并且目标代理的对应信用量可供消耗）。这种情况下能够避免通过队列的等待时间。在其它实施例中，只对一个源代理，或者源代理的子集提供旁路（例如，处理器可具有旁路，其它代理不具有旁路）。在其它实施例中，可不实现旁路，输入的位流与多路复用器32和仲裁器控制电路36的连接可被除去。

[0046] 注意尽管本实施例使用ReorderOK位来指示指定的请求是否可以重排在先前从相同代理传送的请求之前，不过其它实施例可以使用其它指示。例如，如果在队列30A-30B中实现两个以上的队列项，那么存在对应于队列中的每一项的ReorderOK位。每个ReorderOK位可指示该请求是否可相对于对应队列项听请求重排。另一方面，仲裁器控制电路36可实现当确定指定请求是否可被重排在先前传送的请求之前时，应用于代理的一组排序规则。

[0047] 图3是图解说明仲裁控制电路36的一个实施例的操作的流程图。尽管为了易于理解，按照特定的顺序表示各个方框，不过可以使用任何顺序。此外，仲裁控制电路36中的组合逻辑电路可以并行实现各个方框。根据需要，其它方框，方框的组合，或者整个流程图可在多个时钟周期内被流水线化。

[0048] 如果对于当前仲裁周期启动了饥饿控制（判定方框50），那么仲裁控制电路36可超越“正常”（例如，基于优先级的）仲裁。如上所述，如果指定的较低优先级的请求已长时间留在队列中，那么可启动饥饿控制。另一方面，如果在许多连续仲裁内都选择高优先级的请求，那么可启动饥饿控制。如果启动了饥饿控制（判定方框50，“yes”支路），那么仲裁控制电路36可把年老的请求（或者较低优先级的请求）选为仲裁获胜者（方框52）。

[0049] 如果未启动饥饿控制（判定方框50，“no”支路），那么仲裁器控制电路36可掩蔽没有资格仲裁的各个请求，并在未被掩蔽的请求之中进行仲裁。例如，如果队列中的某个请求使其ReorderOK位指示不允许重排，并且在同一队列中存在在先请求，那么该请求可被掩蔽，以避免在所述在先请求之前选择该请求（例如，如果该请求的优先级高 于所述在先请求）（方框54）。另外，如果请求的目标是不可得到用于该请求的任何适当类型的信用量的目标代理，那么该请求可被掩蔽（方框56）。如果仲裁控制电路36限制可被与某一请求对应的源代理消耗的信用量，并且已达到该限度，那么即使存在可供消耗的信用量，该请求也可被掩蔽（方框56）。仲裁控制电路可把未被掩蔽的优先级最高的请求选为仲裁获胜者（方框58）。

[0050] 仲裁控制电路36可通过多路复用器32把仲裁获胜者选到地址互连16上。另外，仲裁控制电路36可向发起所选请求的源代理宣称许可信号，并从队列30A-30B中删除选择的请求。

[0051] 图4是在各个实施例中，可由源代理，或者仲裁控制电路36，或者这两者实现的一组排序规则的一个实施例的方框图。事务类型示于表的上部和表的左侧。行列的交点处是关于该行中的那种事务是否被允许重排在该列中的那种在先事务之前的规则。

[0052] 因此，相关读取和写入请求可被自由重排，只有不存在地址匹配。在这个意义上，在保持相关性的粒度上检测地址匹配（例如，高速缓存块）。相关读取请求可被重排在非相关公告 (posted) 请求和非相关完成 (completion) 之前，但是不可重排在非相关非公告请求（相关读取行和非相关公告，非相关非公告以及非相关完成列）之前。但是，在一些情况下，请求是否可被重排取决于请求的具体实例（包括 a) 和 b) 答案的交叉点）。下面在图 5 中的表中定义了关于 a) 和 b) 的请求的类型。Y/N 意味重排是允许的，但是并不需要被许可。从而，如果图 4 指示 yes 或 Y/N，那么请求可被重排。

[0053] 下面参见图 5，图中表示了在地址互连上传递请求的方法的高级流程图。来自源代理的请求在地址开关中排队（方框 70）。地址开关在排队的请求中进行仲裁，以选择某一请求（方框 72）。选择的请求在地址互连上传送（方框 74）。

[0054] 一旦充分理解上述公开内容，对本领域的技术人员来说，众多的变化和修改将是显而易见的。下面的权利要求意图包含所有这样的变化和修改。

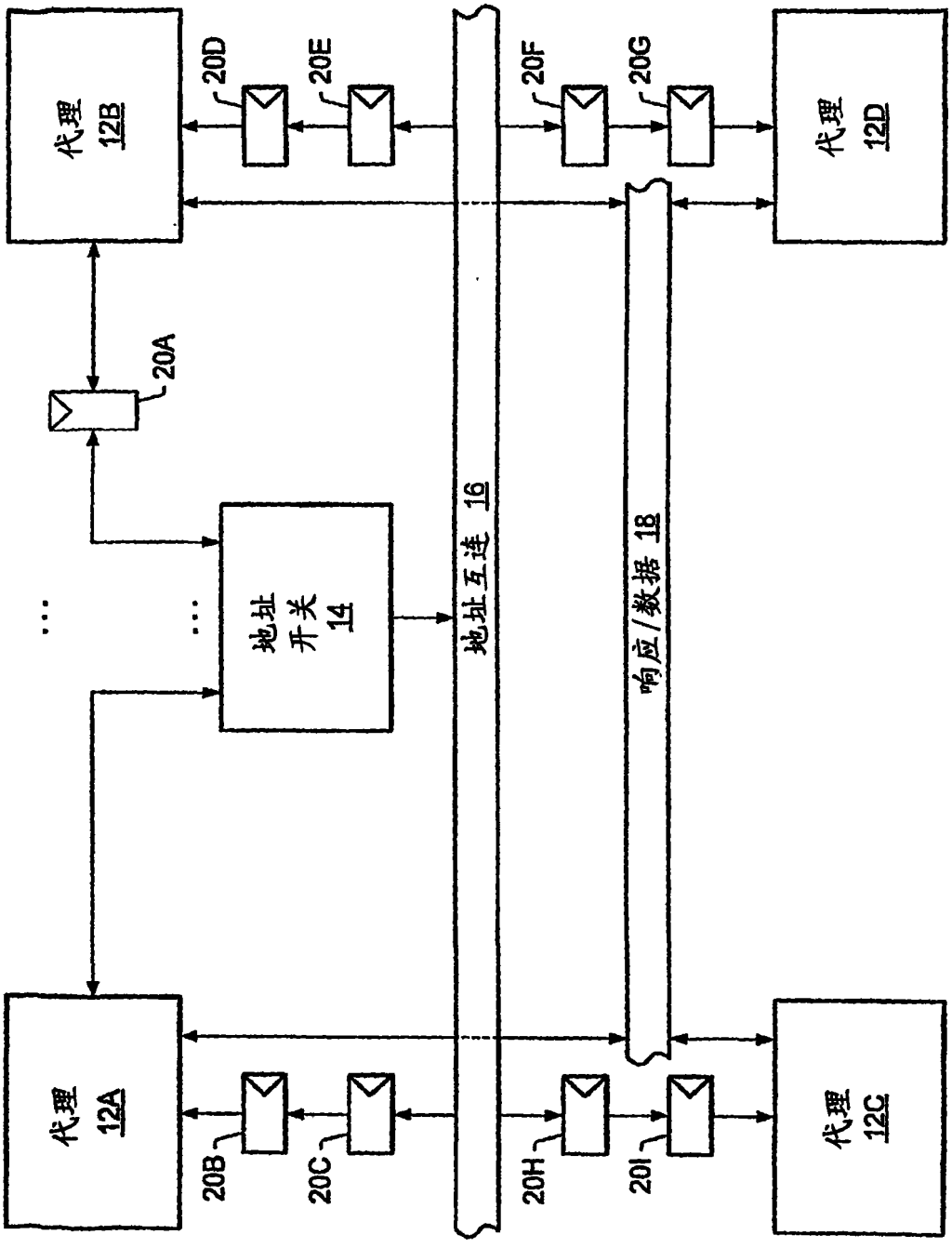

代理 12B
20D
20E
20F
20G
代理 12D
20A
地址 开关 14
地址互连 16
响应/数据 18
代理 12A
20B
20C
20H
20I
代理 12C
10

图 1

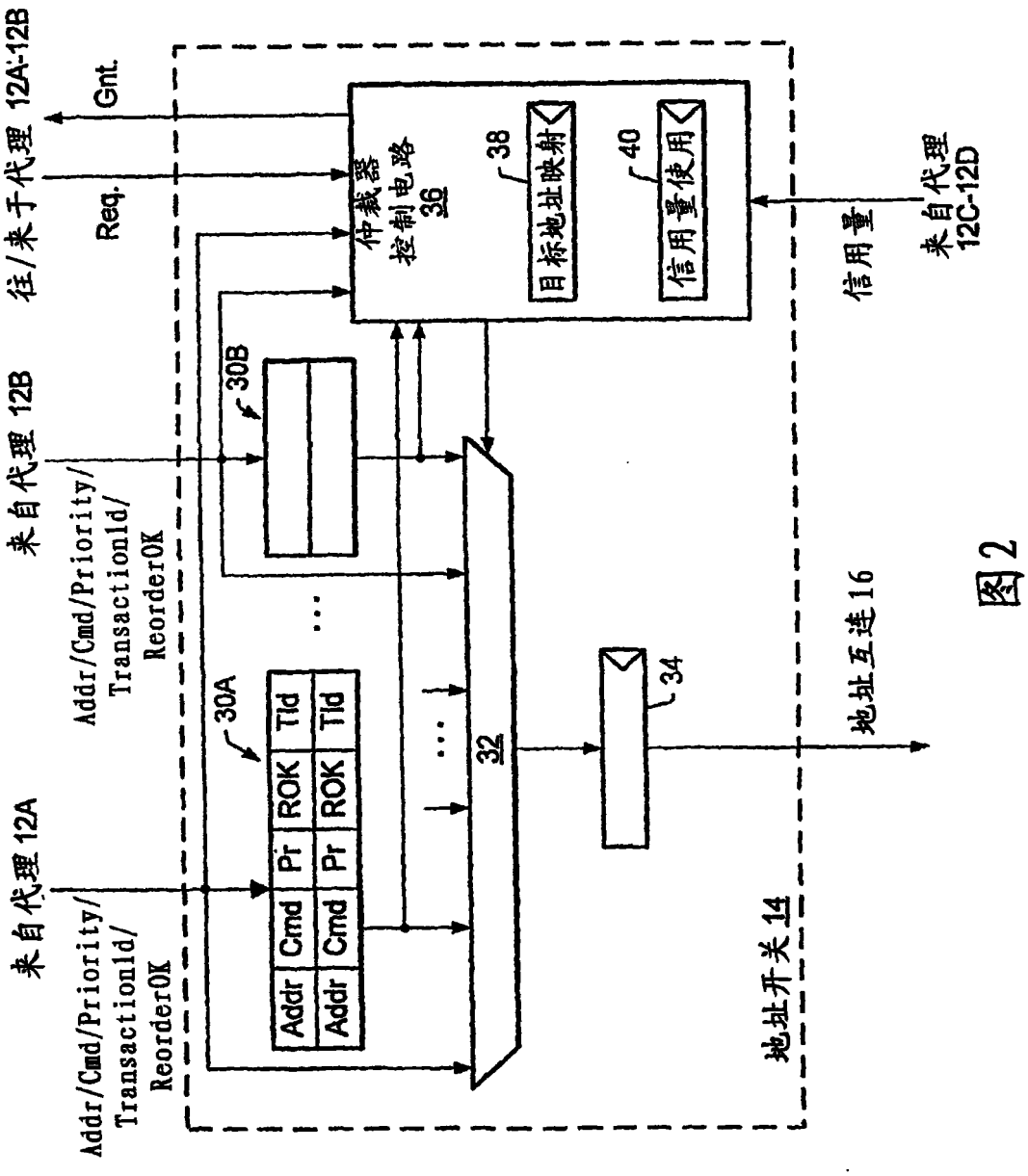

来自代理 12A
Addr/Cmd/Priority/
TransactionId/
ReorderOK
来自代理 12B
Addr/Cmd/Priority/
TransactionId/
ReorderOK
往/来于代理 12A-12B
Gnt.
Req.
仲裁器
控制电路
36
38
目标地址映射
40
信用量使用
信用量
来自代理
12C-12D
30A
30B
Addr
Cmd
Pr
ROK
TId
32
34
地址互连 16
地址开关 14

图 2

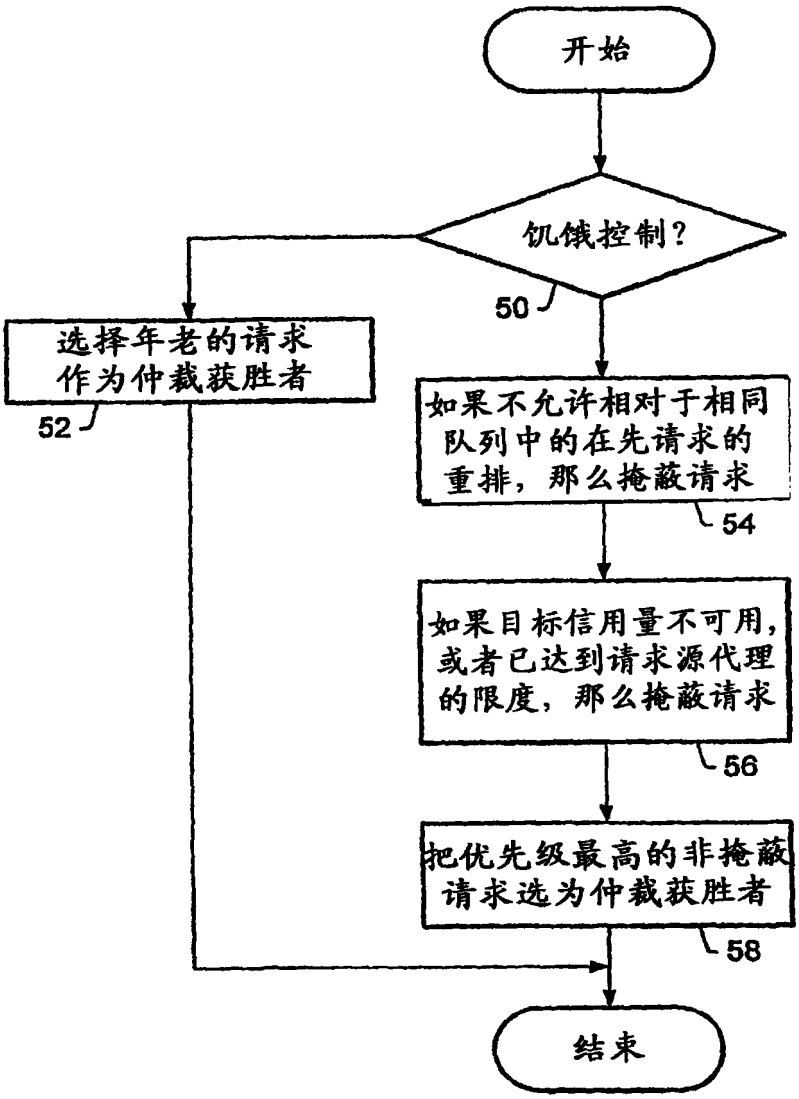

开始
饥饿控制?
50
选择年老的请求
作为仲裁获胜者
52
如果不允许相对于相同
队列中的在先请求的
重排，那么掩蔽请求
54
如果目标信用量不可用，
或者已达到请求源代理
的限度，那么掩蔽请求
56
把优先级最高的非掩蔽
请求选为仲裁获胜者
58
结束

图 3

事务类型	相关读取	相关写入	非相关公告	非相关非公告	非相关完成
相关读取	是(如果无地址匹配)	是(如果无地址匹配)	是	否	是
相关写入	是(如果无地址匹配)	是(如果无地址匹配)	否	是	是
非相关公告	是	否	a) 否 b) Y/N	是	a) Y/N b) 是
非相关非公告	否	否	否	Y/N	Y/N
非相关完成	是	否	a) 否 b) Y/N	是	a) Y/N

a) 读取请求
b) IO写入，配置写入，CR写入

图4

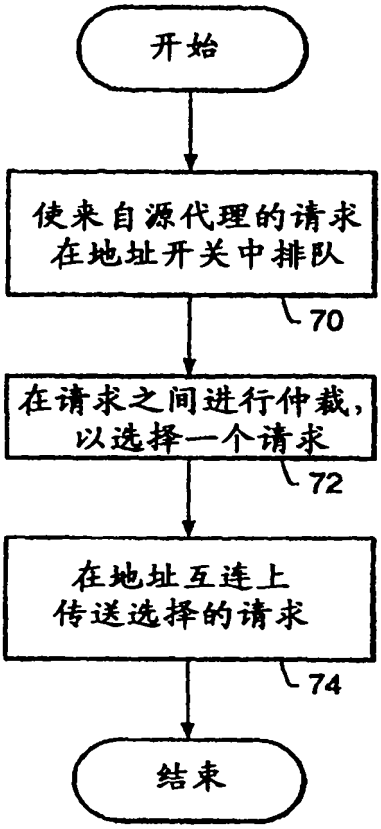

开始
使来自源代理的请求在地址开关中排队
70
在请求之间进行仲裁，以选择一个请求
72
在地址互连上传送选择的请求
74
结束

图 5