(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 101326505 B
(45)授权公告日 2011.06.01

(21)申请号 200680037536.4

(22)申请日 2006.08.11

(30)优先权数据
11/201,573 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/US2006/031521 2006.08.11

(87)PCT申请的公布数据
WO2007/022019 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

(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 13/36(2006.01)

(56)对比文件
CN 1321274 A,2001.11.07, 全文.
US 6460174 B1,2002.10.01, 全文.
US 2005/0071533 A1,2005.03.31, 全文.
US 2002/0159474 A1,2002.10.31, 全文.

审查员 何明伦

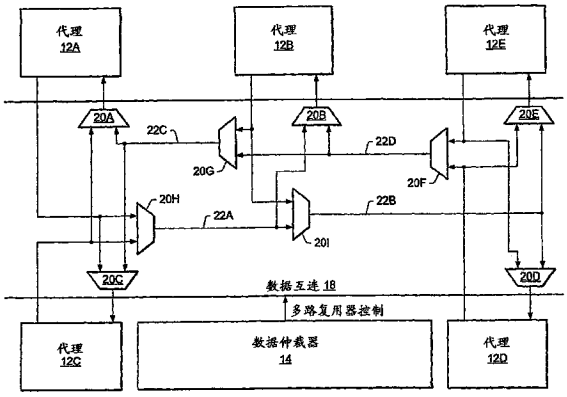
权利要求书 2 页 说明书 8 页 附图 6 页

(54)发明名称

部分填充的分级交叉开关

(57)摘要

在各个实施例中，设备多个代理和一个互连。在一个实施例中，所述多个代理包括第一到第四代理。所述互连包括多个区段，所述多个区段是可切换的（例如，利用多个选择电路），以形成代理之间的通信路径，第一区段包括在从第一代理到第二代理的第一通信路径中，还包括在从第三代理到第四代理的第二通信路径中。在另一实施例中，每个区段由一个选择电路驱动。至少一个选择电路以至少一个区段和来自至少一个代理的输出端作为输入端。在另一实施例中，仲裁器被配置成为每个请求代理确定所述互连上通过各个区段到目的地代理的通信路径。仲裁器被配置成在对其来说，对应通信路径中的每个区段都可用的请求的子集中进行仲裁。



CN 101326505 B

1. 一种用于在集成电路内和 / 或集成电路之间进行互连的设备，包括 ：

多个代理，其中每个代理配置为通过互连进行通信 ；

所述互连，其定义为包括通过多个选择电路耦接的一组区段，所述多个代理与所述互连耦接，所述多个选择电路可配置成在所述多个代理之间的多个区段上建立通信路径，其中所述区段是指一个或多个导体，并且如果一个区段中包括一个以上的导体，那么所述导体可被并行驱动，以把多比特值从导体的一端传送到另一端 ；和

与所述多个代理和所述互连耦接的仲裁器，其中所述仲裁器被耦接成接收来自所述多个代理的使用所述互连的请求，其中每个请求代理被配置成识别所述请求的目的地代理，并且其中仲裁器被配置成为每个请求确定所述互连上越过所述多个区段中的一个或多个区段、从请求代理到目的地代理的通信路径，其中所述仲裁器被配置成在这样的请求的子集中进行仲裁 ：对所述请求来说，对应通信路径中的每个区段都可用。

2. 按照权利要求 1 所述的设备，其中所述仲裁器被配置成选择所述多个代理中的第一代理作为仲裁获胜者，其中所述仲裁器被配置成通过控制所述多个选择电路，建立从第一代理到所述多个代理中的目的地代理的通信路径。

3. 按照权利要求 1 所述的设备，其中所述多个选择电路中的每个选择电路被配置成驱动所述多个区段之一。

4. 按照权利要求 3 所述的设备，其中每个选择电路的输入端与所述多个区段中的至少一个区段和 / 或所述多个代理中的至少一个代理的输出端耦接。

5. 按照权利要求 1 所述的设备，其中所述区段中的一些区段由不同的通信路径共用，并且如果这些通信路径之一正在使用中，那么不能建立使用共用区段的其它通信路径。

6. 按照权利要求 1 所述的设备，其中所述多个区段中的每个区段由所述多个选择电路中的相应选择电路驱动，其中至少一个选择电路以所述多个区段中的至少一个区段和来自所述多个代理中的至少一个代理的输出端作为输入端，其中通过配置所述多个选择电路中的一个或多个选择电路，以耦接请求代理的输出端与目的地代理的输入端，建立从所述多个代理中的请求代理到所述多个代理中的目的地代理的通信路径。

7. 按照权利要求 6 所述的设备，其中所述多个区段中的第一子集沿第一方向在所述多个代理之间传送通信，其中所述多个区段中的第二子集沿第二方向在所述多个代理之间传送通信，其中第一子集与第二子集不相重叠，其中从请求代理沿目的地代理的方向驱动从请求代理到目的地代理的通信。

8. 按照权利要求 6 所述的设备，其中所述多个选择电路物理分布在所述互连内。

9. 一种用于在集成电路内和 / 或集成电路之间进行互连的方法，包括 ：

从与互连耦接的多个代理接收使用所述互连的请求，其中每个代理配置为通过所述互连进行通信，并且其中所述互连定义为包括通过多个选择电路耦接的一组区段，所述多个选择电路可配置成在所述多个代理之间的多个区段上建立通信路径，所述区段是指一个或多个导体，并且如果一个区段中包括一个以上的导体，那么所述导体可被并行驱动，以把多比特值从导体的一端传送到另一端，其中每个请求代理被配置成识别所述请求的目的地代理 ；

为每个请求确定所述互连上越过所述多个区段中的一个或多个区段、从请求代理到目的地代理的通信路径 ；和

在这样的请求的子集中进行仲裁：对所述请求来说对应通信路径中的每个区段都可用。

10. 按照权利要求 9 所述的方法，还包括：

选择所述多个代理中的第一代理作为仲裁获胜者；和

建立从第一代理到所述第一代理识别的所述多个代理中的目的地代理的通信路径，其中所述建立包括控制所述多个选择电路。

部分填充的分级交叉开关

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。

背景技术

[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。

[0003] 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。地址总线和数据总线的带宽相当有限，因为每次在地址总线上只能够传送一个地址传输，在数据总线上只能够传送一个数据传输。即使两个或更多的传输可以并行地进行，总线结构也不允许这种并行性。另外，总线往往具有高的功耗，因为每个通信器与总线耦接，对总线的驱动器带来电容性负载。

[0004] 通常使用的另一种互连被称为交叉开关(crossbar)。交叉开关一般允许每个发射通信器并行地与任意接收通信器通信，并且允许任何其它发射通信器并行地与任何其它接收通信器通信。但是，交叉开关中为了允许发射器和接收器之间的任意一组并发连接而需要的电路常常导致高的功耗。

发明内容

[0005] 在一个实施例中，一个设备包括第一代理、第二代理、第三代理和第四代理和一个互连。第一到第四代理都与所述互连耦接，所述互连包括多个区段(segment)，所述区段是可切换的，以形成与所述互连耦接的代理之间的通信路径。所述多个区段中的第一区段包括在从第一代理到第二代理的第一通信路径中，还包括在从第三代理到第四代理的第二通信路径中。

[0006] 在另一实施例中，一个设备包括与一个互连耦接的多个代理。所述互连包括多个区段和多个选择电路，其中所述多个区段都由所述多个选择电路中的相应选择电路驱动。至少一个选择电路以所述多个区段中的至少一个区段和来自所述多个代理中的至少一个代理的输出端作为输入端。通过配置所述多个选择电路中的一个或多个选择电路，以耦接第一代理的输出端与第二代理的输入端，建立从所述多个代理中的第一代理到所述多个代理中的第二代理的通信路径。

[0007] 在另一实施例中，一种设备包括与互连耦接的多个代理。所述互连包括多个区段和多个选择电路，所述多个选择电路可配置成在所述多个代理之间的多个区段上建立通信路径。所述设备还包括一个仲裁器，其与所述多个代理和所述互连耦接，并被耦接成接收来

自所述多个代理的使用所述互连的请求。每个请求代理被配置成识别所述请求的目的地代理，仲裁器被配置成为每个请求确定所述互连上越过所述多个区段中的一个或多个区段、从请求代理到目的地代理的通信路径。仲裁器被配置成在对于其来说、对应通信路径中的每个区段都可用的请求的子集中进行仲裁。

[0008] 在另一实施例中，一种方法包括：从与互连耦接的多个代理接收使用所述互连的请求，其中所述互连包括多个区段和多个选择电路，所述多个选择电路可配置成在所述多个代理之间的多个区段上建立通信路径，其中每个请求代理被配置成识别所述请求的目的地代理；为每个请求确定所述互连上越过所述多个区段中的一个或多个区段、从请求代理到目的地代理的通信路径；和在这样的请求的子集中进行仲裁：对所述请求来说对应通信路径中的每个区段都可用。

附图说明

[0009] 下面的说明参考了附图，现在简要说明这些附图。

[0010] 图 1 是集成电路的一个实施例的方框图。

[0011] 图 2 是图 1 中所示的数据互连的一个实施例的方框图。

[0012] 图 3 是图解说明图 1 和 2 中所示的用于数据互连的仲裁的数据仲裁器的一个实施例的操作的流程图。

[0013] 图 4 是图解说明当数据传输将要完成时，图 1 和 2 中所示的数据仲裁器的一个实施例的操作的流程图。

[0014] 图 5 是图解说明在数据互连上配置的第一通信路径的一个例子的方框图。

[0015] 图 6 是图解说明与第一通信路径同时，在数据互连上配置的第二通信路径的一个例子的方框图。

[0016] 图 7 是图解说明数据仲裁器的仲裁接口的一个实施例的方框图。

[0017] 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

[0018] 现在参见图 1，图 1 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 10 还包括地址 / 响应互连 16，数据互连 18 和数据仲裁器 14。代理 12A-12D 与数据仲裁器 14 耦接，并与数据互连 18 耦接。另外，代理 12A-12D 与地址 / 响应互连 16 耦接。数据仲裁器 14 还与数据互连 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中。在其它实施例中，可在独立的集成电路上实现系统 10 的各个组件。在各个实施例中可以使用任意级别的集成。

[0019] 至少一些代理 12A-12D 被配置成启动与系统 10 中的其它代理 12A-12D 通信的事务。所述事务可包括地址 / 响应互连 16 上的地址阶段，和数据互连 18 上的数据阶段。相关事务包括地址 / 响应互连 16 上的响应阶段。

[0020] 地址 / 响应互连 16 可包括任意通信介质。例如，在一个实施例中，可在源代理

(source agent) 之间仲裁地址 / 响应互连 16。可以使用集中仲裁方案，其中源代理向中央仲裁器传送地址阶段请求（包括事务的地址、命令等），中央仲裁器可使请求排队，在所述请求之中进行仲裁，并在地址 / 响应互连 16 上传送获胜的事务。其它实施例可以使用任何其它所需的通信介质。

[0021] 代理 12A-12D 也可对数据互连 18 的使用进行仲裁，以执行事务的数据阶段。代理 12A-12D 可用信号向数据仲裁器 14 通知对数据互连 18 的请求，数据仲裁器 14 可在所述请求之中进行仲裁，并选择仲裁获胜者。数据仲裁器 14 可用信号通知获胜的代理 12A-12D，响应对数据互连 18 的请求的准许，获胜的代理 12A-12D 可在数据互连 18 上驱动数据。可以使用任何仲裁方案（例如，优先权、循环法、加权循环法、优先权和各种循环方案的组合等等）。

[0022] 在一个实施例中，数据互连 18 可包含部分填充的交叉开关 (partially-populated crossbar)。数据互连 18 可支持一定量的数据传输同时性（并行性），但是可能不实现全交叉开关。在一些实施例中，当数据互连 18 中包括简化的电路时，可支持高带宽和低功率的组合。在一种实现中，数据互连 18 可包含由诸如多路复用器之类的选择电路耦接的一组区段。数据仲裁器 14 可控制多路复用器建立经由所述区段中的一些区段，从一个代理 12A-12D 到另一个代理 12A-12D 的通信路径。所述区段中的一些区段可被不同的通信路径共用，从而，如果通信路径之一正在使用中，那么在建立的通信路径上的数据阶段结束之前，不能建立使用共用区段的其它通信路径。数据仲裁器 14 可被配置成为每个请求的数据阶段确定通信路径，并且可以掩蔽对其来说，所需通信路径的至少一个区段正在使用的那些请求。下面关于图 2-7 提供数据互连 18 的一个实施例的补充细节。

[0023] 代理 12A-12D 中的一个或多个可被配置成在系统 10 中发起事务。这样的代理可被称为源代理。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) 缓存区块发起 (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）。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提供的写入数据。例如，目标代理可包括存储器控制器和 I/O 桥接器。从而，对于读取事务来说，目标代理可仲裁数据互连 18，并在数据互连 18 上传送数据。对于写入事务来说，源代理可仲裁数据互连 18，并在数据互连 18 上传送数据。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 / 响应互连 16 和数据互连 18 上的事务进行通信的任何电路。这里使用的区段指的是诸如导线之类的一个或多个导体。如果一个区段中包括一个以上的导体，那么所述导体可被并行驱动，以把多比特值从导体的一端传送到另一端。例如，在一个实施例中，每个时钟周期，数据互连 18 可传送 16 字节的数据。在这样的实施例中，128 个并行驱动的导体可构成一个区段。在一个特殊的实施例中，在每个区段中还可包括用于纠错码 (ECC) 位的额外导体。另外，命令、事务 ID、有效信号等等可以或者作为区段的一部分或者单独地以广播方式在数据互连 18 上被传送给每个代理 12A-12D。

[0024] 注意在下面的例子中，多路复用器将被用作选择电路。但是一般来说，接收两个或

者更多的输入以及一个选择控制输入，以便在所述两个或更多的输入之中进行选择的任何电路可被用于构成选择电路。

[0025] 现在参见图 2，图中更详细地表示了数据互连 18 的一个实施例的方框图。图 2 中表示了附加的代理 12E 以及代理 12A-12D。在各个实施例中可以包括任意数目的代理。代理 12E 可按照与其它代理 12A-12D 类似的方式（图 2 中未示出）与数据仲裁器 14 耦接。在图解说明的实施例中，数据互连 18 包括一组多路复用器 20A-20I。在图 2 中，多路复用器 20A-20I 被耦接以便分别驱动代理 12A-12E 的数据输入端。多路复用器 20A-20I 均被耦接，以便驱动数据互连 18 的某一区段（例如，图 2 中所示的区段 22A-22D）。多路复用器 20A-20I 的每个输入端或者是区段 22A-22D，或者是代理 12A-12D 之一的数据输出端。特别地，在图解说明的实施例中，多路复用器输入端如下所示：多路复用器 20A 以代理 12C 的数据输出端和区段 22C 作为输入端；多路复用器 20B 以区段 22A 和 22D 作为输入端；多路复用器 20C 以代理 12A 的数据输出端和区段 22C 作为输入端；多路复用器 20D 以代理 12E 的数据输出端和区段 22B 作为输入端；多路复用器 20E 以代理 12D 的数据输出端和区段 22B 作为输入端；多路复用器 20F 以代理 12D-12E 的数据输出端作为输入端；多路复用器 20G 以代理 12B 的数据输出端和区段 22D 作为输入端；多路复用器 20H 以代理 12A 和 12C 的数据输出端作为输入端；多路复用器 20I 以代理 12B 的数据输出端和区段 22A 作为输入端。每个多路复用器 20A-20I 都具有在其输入端之中进行选择的多路复用器控制器。数据仲裁器 14 可提供多路复用器控制器，以便在数据互连 18 上建立通信路径。为了使图 2 简明，图 2 中未示出各个多路复用器 20A-20I 的独有控制器。

[0026] 数据仲裁器 14 可控制多路复用器 20A-20I 在数据互连 18 上建立从作为数据仲裁器 14 确定的仲裁获胜者（“获胜的代理”）12A-12E 到目的地代理 12A-12E 的通信路径，所述目的地代理 12A-12E 将接收由获胜的代理 12A-12E 驱动的数据。即，数据仲裁器 14 可控制多路复用器 20A20I 耦接获胜代理的数据输出端与目的地代理的数据输入端。在一个实施例中，每个请求代理可识别对数据仲裁器 14 的数据传输以及所述请求的目的地代理。例如，对于读取操作来说，请求代理可提供源自在事务的地址阶段中，由源代理提供的事务 ID 的代理 ID。对于写入操作来说，源代理可被提供目的地代理的指示（例如，在上面关于图 1 说明的集中实施例中由地址仲裁器提供，或者由目标代理提供）。源代理可把目的地代理 ID 连同所述请求一起提供给数据仲裁器 14。

[0027] 例如，在图 2 的实施例中，通过控制多路复用器 20H 选择代理 12A 的数据输出端；控制多路复用器 20I 选择区段 22A；和控制多路复用器 20D 选择区段 22B，可建立从代理 12A 到代理 12D 的通信路径。作为另一例子，通过控制多路复用器 20G 选择代理 12B 的数据输出端和控制多路复用器 20C 选择区段 22C，可建立从代理 12B 到代理 12C 的通信路径。

[0028] 数据互连 18 可被看作部分填充的交叉开关。即，每个代理 12A-12E 在其数据输出端驱动数据，数据互连 18 通过一个或多个区段把数据发送到目的地代理 12A-12E。数据互连 18 可提供部分并行性，但是不提供全交叉开关的全并行性。例如，在图 2 的实施例中，如果代理 12B 正在向代理 12D 传送数据，那么代理 12A 和 12C 不能并行地向代理 12E 传送数据（因为区段 22B 被用于从代理 12B 到代理 12D 的通信，还被用于从代理 12A 和 12C 到代理 12E 的通信）。但是，代理 12E 可与正在向代理 12D 传送数据的代理 12B 并行地向代理 12A 或 12C 传送数据（通过区段 22C-22D）。

[0029] 在数据互连 18 提供的并行性的数量因实施例而异。一般来说，增大包括在数据互连 18 中的区段和多路复用器的数目，可增大并行性。可按照任何所需的方式增加多路复用器和区段，以便在给定代理和另一给定代理，给定代理和任何其它代理，或者给定代理和其它代理的子集之间形成另外的路径。提供的并行性的数量可由所需的性能水平（例如，就增大的带宽而论）和 / 或由在各种工作负荷下预期的并发通信量确定。例如，在一个实施例中，系统 10 可包括两个处理器，一个 L2 高速缓存，一个 I/O 桥接器，和两个存储器控制器。在该实施例中，最好至少提供下述并行性 :(i) 在一个处理器正在向 L2 高速缓存传送数据的时候，所述 L2 高速缓存能够向存储器控制器传送数据 ;(ii) 在 L2 高速缓存正在向一个处理器传送数据的时候，存储器控制器能够向其它处理器传送数据 ;和 (iii) 在 L2 高速缓存正在向 I/O 桥接器传送数据的时候，存储器控制器能够向处理器传送数据。

[0030] 在图解说明的实施例中，至少一些区段 22A-22D 在从不同的请求代理到不同的接收代理的通信路径之中被共用。例如，区段 22B 在从任意代理 12A-12C 到代理 12D-12E 任意之一的路径中被共用。即，从任意代理 12A-12C 到代理 12D-12E 任意之一的通信路径使用区段 22D。另外，从任意代理 12A-12C 到代理 12E 的通信路径还使用区段 22B。类似地，区段 22A 被用在从代理 12A 或 12C 到代理 12B、12E 或 12D 任意之一的通信路径中。区段 22A 未被代理 12B 使用。在一些实施例中，从不同的请求代理到不同的目的地代理的通信路径之中区段的共用可减少用于实现数据互连 18 的布线的数量，这可简化布线，并且可降低总的能耗。

[0031] 数据互连 18 可以是分布式交叉开关。多路复用器 20A-20I 可沿着数据互连 18 经过的物理距离物理分布。例如，以代理 12A-12E 的数据输出端作为输入端的多路复用器可物理地位于该代理附近。从而，多路复用器 20F 可物理地位于代理 12E 和 12D 附近 ;多路复用器 20G 和 20I 可物理地位于代理 12B 附近 ;多路复用器 20H 可物理地位于代理 12A 和 12C 附近。驱动代理 12A-12E 的数据输出端的多路复用器可物理地位于该代理附近。例如，多路复用器 20A-20E 可分别物理地位于代理 12A-12E 附近。区段 22A-22D 可把来自源多路复用器的数据沿着数据互连 18 的一部分物理距离向下传送到下一个多路复用器，如图 2 中所示。

[0032] 多路复用器 20A-20I 可形成分级的多路复用结构。在传统的交叉开关中，一组输入多路复用器从给交叉开关的输入中进行选择，另外的多路复用器可被用于在输入多路复用器的输出中进行选择，以产生给指定目的地的交叉开关输出。至少一些多路复用器 20A-20I 在多路复用器输出（区段 22A-22D) 和来自某一代理的输出之间进行选择。例如，在图 2 的实施例中，多路复用器 20I 在区段 22A（在先多路复用器 20H 的输出）和来自代理 12B 的输出之间进行选择。类似地，多路复用器 20D-20E 分别在区段 22B 和代理 12E 及 12D 的输出之间进行选择。

[0033] 数据互连 18 本质上也可以是定向的。即，按照目的地代理的方向（并不是按照其它方向）在数据互连 18 上从给定代理 12A-12E 驱动数据。多路复用器 20F-20G 和区段 22C-22D 提供朝左的数据传输（如图 2 中所示），多路复用器 20H-20I 和区段 22A-22B 提供朝右的数据传输（如图 2 中所示）。即，区段 22A-22D 的非重叠子集被用于每个方向的通信。例如，如果代理 12B 正在向代理 12A 或 12C 传送数据，那么通过多路复用器 22C，区段 22C，和多路复用器 20A 或 20C 驱动数据。类似地，如果代理 12B 正在向代理 12E 或 12D 传送数

据，那么通过多路复用器 20I，区段 22B 和多路复用器 20E 或 20D 驱动数据。可以只在用于把数据从请求代理传给目的地代理的那部分数据互连 18 上驱动数据。例如，在区段 22D 或 22A 上并不驱动来自代理 12B 的数据。

[0034] 注意在其它实施例中可以包括附加的代理。每个附加的代理可由类似于多路复用器 20G 和 20I 的附加多路复用器，以及由这些多路复用器驱动的附加区段支持。

[0035] 在一些实施例中，数据互连 18 可被流水线化。例如，如果从数据互连 18 的一端到另一端的飞行时间超过用于数据互连 18 的时钟的一个时钟周期，那么可能需要流水线技术。可随意地在数据互连 18 中插入触发器 (flop) 或者其它时钟存储装置，以使互连流水线化。

[0036] 注意尽管在图解说明的实施例中，数据仲裁器 14 被表示成提供对多路复用器 20A-20I 的多路复用器控制，不过产生多路复用器控制的电路可与数据仲裁器 14 分离。该电路可被告知给获胜代理的许可，以及被告知对应的目的地代理，并且可控制多路复用器建立从获胜代理到目的地代理的通信路径。

[0037] 要注意的是，尽管在图 2 的实施例中表示了两个输入端的多路复用器，不过其它实施例可包括具有两个输入端以上的多路复用器。这样的多路复用器可与作为输入端的两个或更多区段和 / 或代理的两个或更多数据输出端耦接。

[0038] 下面参考图 3，图中表示了图解说明执行数据互连 18 的仲裁的数据仲裁器 14 的一个实施例的操作。尽管为了便于理解，在图 3 中按照特殊的顺序表示各个方框，不过也可使用其它顺序。此外，所述方框可由数据仲裁器 14 中的组合逻辑电路并行执行。如果需要的话，各个方框，流程图的多个部分，或者整个流程图可在多个时钟周期内被流水线化。

[0039] 对于每个宣称的请求，数据仲裁器 14 可确定从请求代理到目的地代理的路径（方框 30）。每个请求代理可具有它自己的请求信号，从而数据仲裁器 14 可根据对应请求信号的宣称，识别请求代理。请求代理还可提供目的地代理的识别。数据仲裁器可掩蔽对应通信路径的至少一个区段不可以被分配给请求代理的通信路径的每个请求（方框 32）。从而，只有在仲裁中只考虑通信路径可被建立的那些请求。如果某一区段被分配用在另一数据传输的通信路径中，那么该区段不可用。

[0040] 在图 2 的实施例中，从指定的请求代理到指定的目的地代理只有一条路径。在为了得到额外的并行性而提供额外的区段和多路复用器的实施例中，如果从指定的请求代理到指定的目的地代理可能存在一条以上的路径，那么数据仲裁器 14 在确定通信路径的时候，会考虑每条可能的路径。从仲裁的观点来看，如果可能的话，数据仲裁器 14 会选择一条所有区段都可用的通信路径。

[0041] 如果至少一个宣称的请求未被掩蔽（判定方框 34，“yes”支路），那么数据仲裁器 14 可选择获胜的代理，并向获胜的代理发出许可（方框 36）。获胜的代理选自请求代理的子集，其中，所述子集的成员是其宣称的请求信号未由于从请求代理到目的地代理的用于数据传输的通信路径中的至少一个区段的不可用性而被掩蔽的请求代理。利用由数据仲裁器 14 实现的仲裁方案确定获胜的代理。数据仲裁器 14 也可把用在从获胜的代理到其目的地代理的通信路径中的区段记录成“忙”（或者“不可用”），供后续仲裁中通信路径的考虑之用（方框 38）。例如，从仲裁的观点来看，数据仲裁器 14 可为每个区段保持一位，指示该区段是忙还是可用。数据仲裁器 14 可产生对多路复用器 20A-20I 的多路复用器控制，以建

立从获胜的代理到其目的地代理的通信路径（方框 40）。在一个实施例中，通信路径的实际建立可因获胜代理的确定而被延迟。例如，在一个实施例中，获胜代理在收到许可之后两个时钟周期开始其数据传输。通信路径的建立可类似地被延迟。另一方面，所述许可和数据传输的开始之间的时钟周期可被用于建立通信路径。如果不存在宣称的且未被掩蔽的请求（判定方框 34，“no”支路），那么不宣布任何获胜的代理。

[0042] 在一个实施例中，每个时钟周期，数据仲裁器 14 可准许一个请求代理。在存在独立的通信路径（即，没有共用区段）的情况下，借助在后续时钟周期中，数据仲裁器 14 发出的额外许可，可以产生数据传输的同时性。在其它实施例中，数据仲裁器 14 可被配置成每个时钟周期发出多个许可。

[0043] 现在参见图 4，图中表示了图解说明响应数据互连 18 上的数据传输的结束，数据仲裁器 14 的一个实施例的操作。在一些实施例中，数据仲裁器 14 可预先知道数据传输将在指定的时钟周期结束，并且可以探测数据传输将在所述指定的时钟周期结束。例如，在图 7 中，表示了一个仲裁接口，其中请求代理可指示数据传输将以已知数目的连续时钟周期的脉冲串的形式发生（例如，对于每个时钟 16 字节的传输和 64 字节的高速缓存块传输来说，4 个时钟周期）。在这种情况下，数据仲裁器可探测数据传输将在后续传输的开始之前结束，并且可响应这样的探测执行图 4 中图解说明的操作。这样的操作允许关于共用一个或多个区段的通信路径的许可的流水线操作。在其它情况情况下，数据仲裁器 14 可跟踪有效的数据传输（例如，数据有效信号可指示数据互连 18 上的数据传输），以探测数据传输的结束，或者传输代理可用信号通知数据仲裁器 14 数据传输将要结束。如果数据仲裁器 14 发现某一数据传输将要结束，那么数据仲裁器 14 可把在该传输中用于形成通信路径的区段记录成“不忙”或者“可用”（方框 50）。从而，这些区段可分配给用于请求代理的通信路径。在一些实施例中，数据仲裁器 14 还可产生对多路复用器 20A-20I 的多路复用器控制，以便即使在该数据路径上的数据传输结束之后仍然保持通信路径（例如，如果这样做能够节电的话）。

[0044] 图 5 是类似于图 2 的方框图，图解说明数据互连 18 上的第一通信路径的例子。在图 5 的例子中，从代理 12C 到代理 12D 建立该通信路径，如图 5 中的虚线所示。数据仲裁器 14 控制多路复用器 20H 选择代理 12C 的数据输出端到区段 22A。数据仲裁器 14 控制多路复用器 20I 选择区段 22A 到区段 22B。数据仲裁器 14 控制多路复用器 20D 选择区段 22 到代理 12D 的数据输入端。按照这种方式，从代理 12C 的输出端开始经多路复用器 20H-20I 和 20D，并越过区段 22A-22B 建立通信路径。

[0045] 图 6 是类似于图 2 和 5 的方框图，图解说明数据互连 18 上与图 5 中图解说明的第一通信路径同时的第二通信路径的例子。在图 6 的例子中，第二通信路径从代理 12E 到代理 12A，如图 6 中较小的虚线所示（与第一通信路径的虚线相比）。数据仲裁器 14 控制多路复用器 20F 选择代理 12E 的数据输出端到区段 22D。数据仲裁器 14 控制多路复用器 20G 选择区段 22D 到区段 22C。数据仲裁器控制多路复用器 20A 选择区段 22C 到代理 12A 的数据输入端。按照这种方式，与第一通信路径同时地，从代理 12E 的数据输出端开始经多路复用器 20F-20G 和 20A，并越过区段 22C-22D 建立通信路径。

[0046] 现在参见图 7，图中表示了图解说明数据仲裁器 14 的一个实施例的例证仲裁接口。在图解说明的实施例中，该接口包括数据仲裁器 14 和每个代理之间的各个信号（对于

每个代理，例如图 7 中的代理 12A-12B 来说，所述各个信号是相同的）。在图解说明的实施例中，所述信号包括请求信号 (Req.)，脉冲串信号 (Brst.)，目的地标识符 (DestID) 和许可信号 (Gnt.)。

[0047] 每个代理可通过宣称其请求信号，请求数据互连 18。与请求信号同时，代理可利用目的地 ID，识别正在请求的数据传输的目的地代理，也可利用脉冲串信号指示数据是否将以连续时钟周期的脉冲串的形式传送。在一些实施例中，数据仲裁器 14 可使用脉冲串信号确定数据传输预期将在何时结束。在其它实施例中，可以不包括脉冲串信号。数据仲裁器 14 可通过宣称其许可信号，用信号把所述许可通知仲裁获胜者。

[0048] 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。

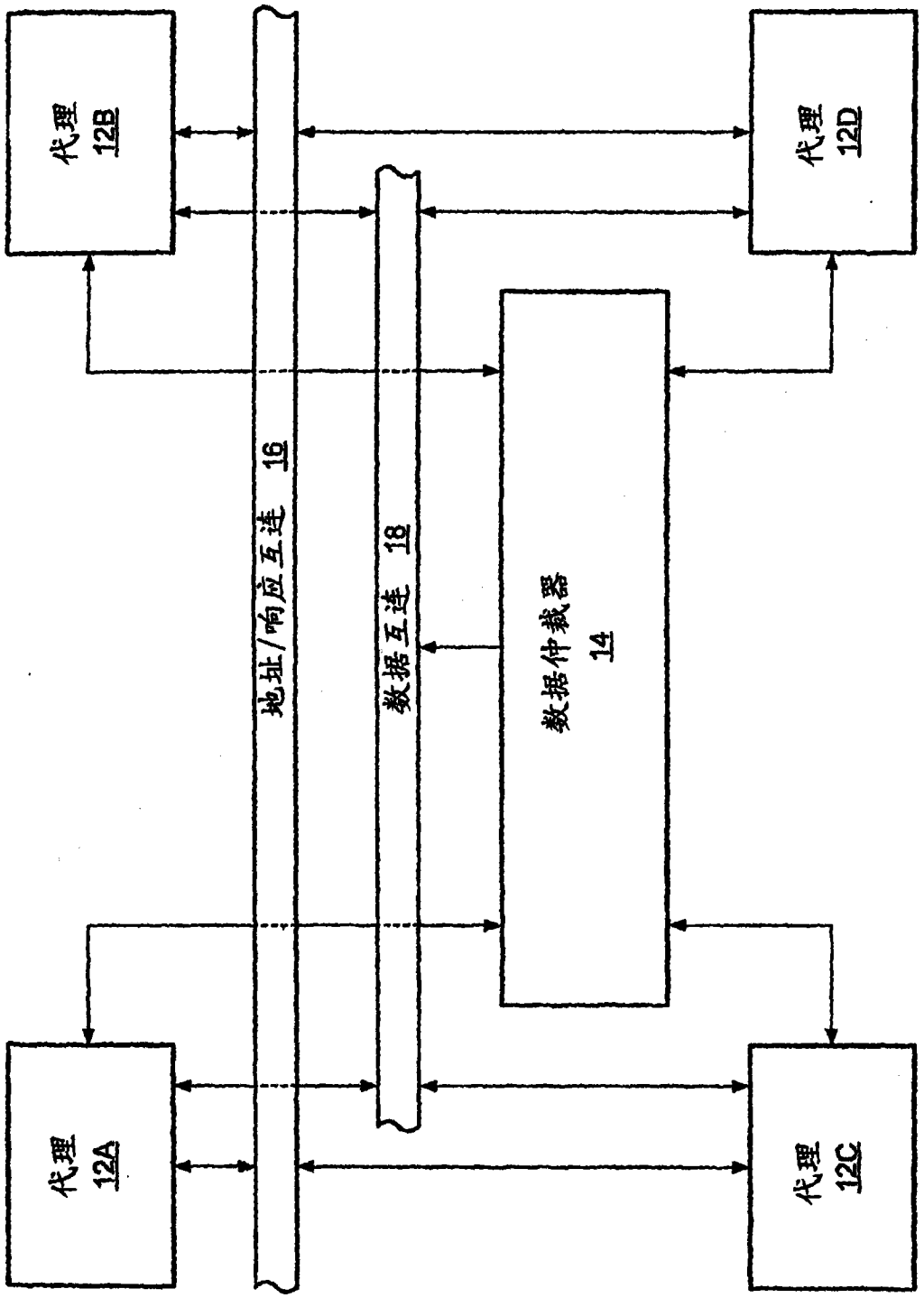


图 1

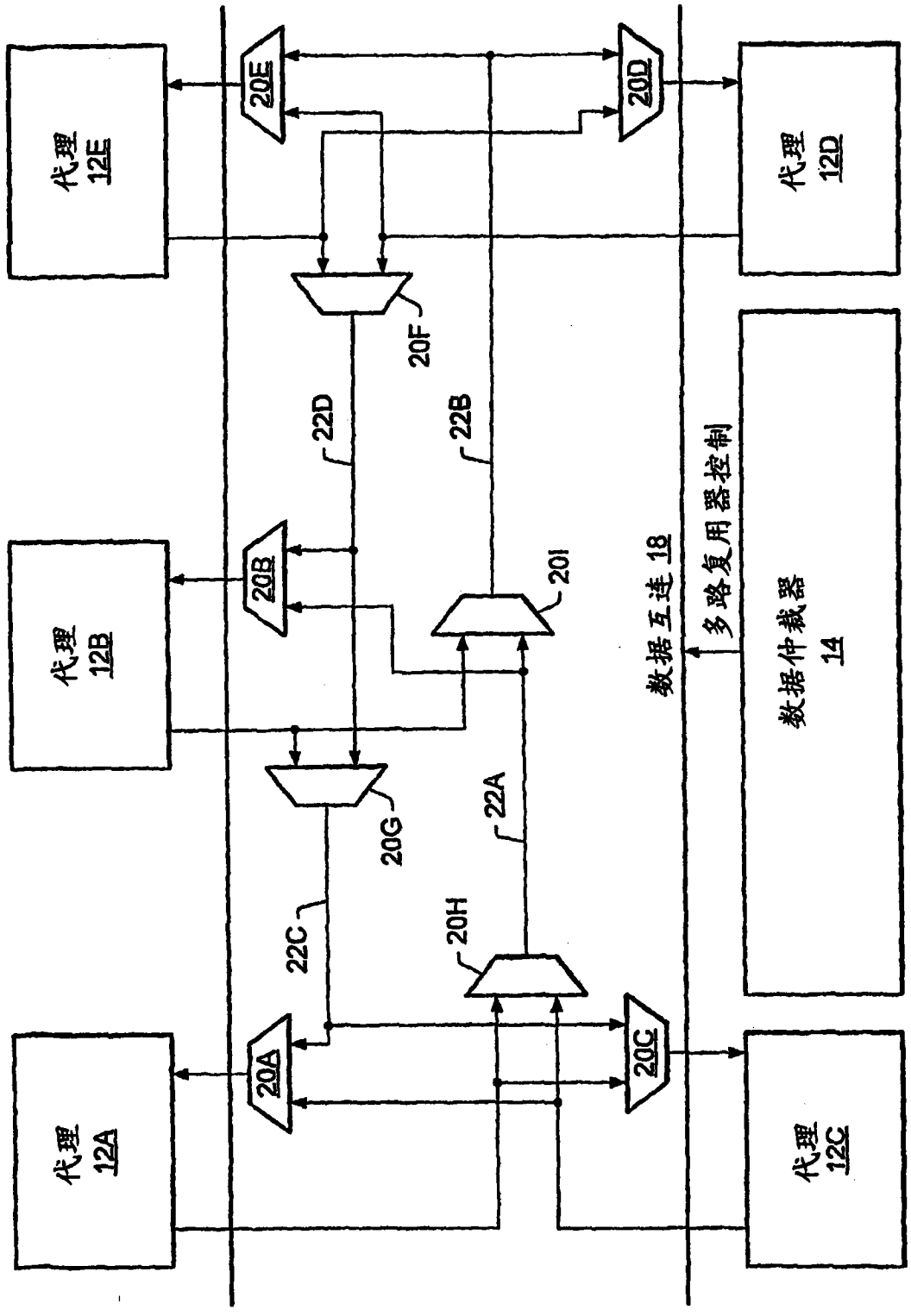

代理 12E
代理 12D
代理 12B
代理 12A
代理 12C
20E
20D
20F
20B
20I
20G
20H
20A
20C
22D
22B
22A
22C
数据互连 18
多路复用器控制
数据仲裁器 14

图 2

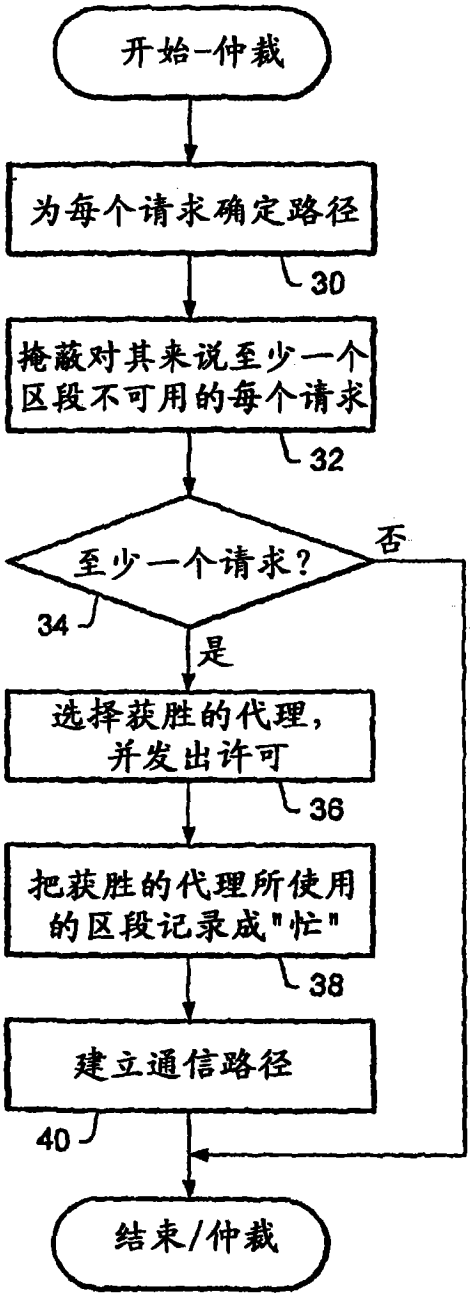

开始-仲裁
为每个请求确定路径
30
掩蔽对其来说至少一个区段不可用的每个请求
32
至少一个请求?
否
34
是
选择获胜的代理，并发出许可
36
把获胜的代理所使用的区段记录成"忙"
38
建立通信路径
40
结束/仲裁

图 3

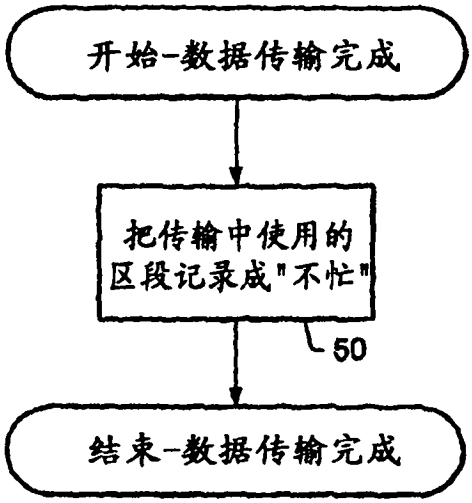

开始-数据传输完成
把传输中使用的区段记录成"不忙"
50
结束-数据传输完成

图 4

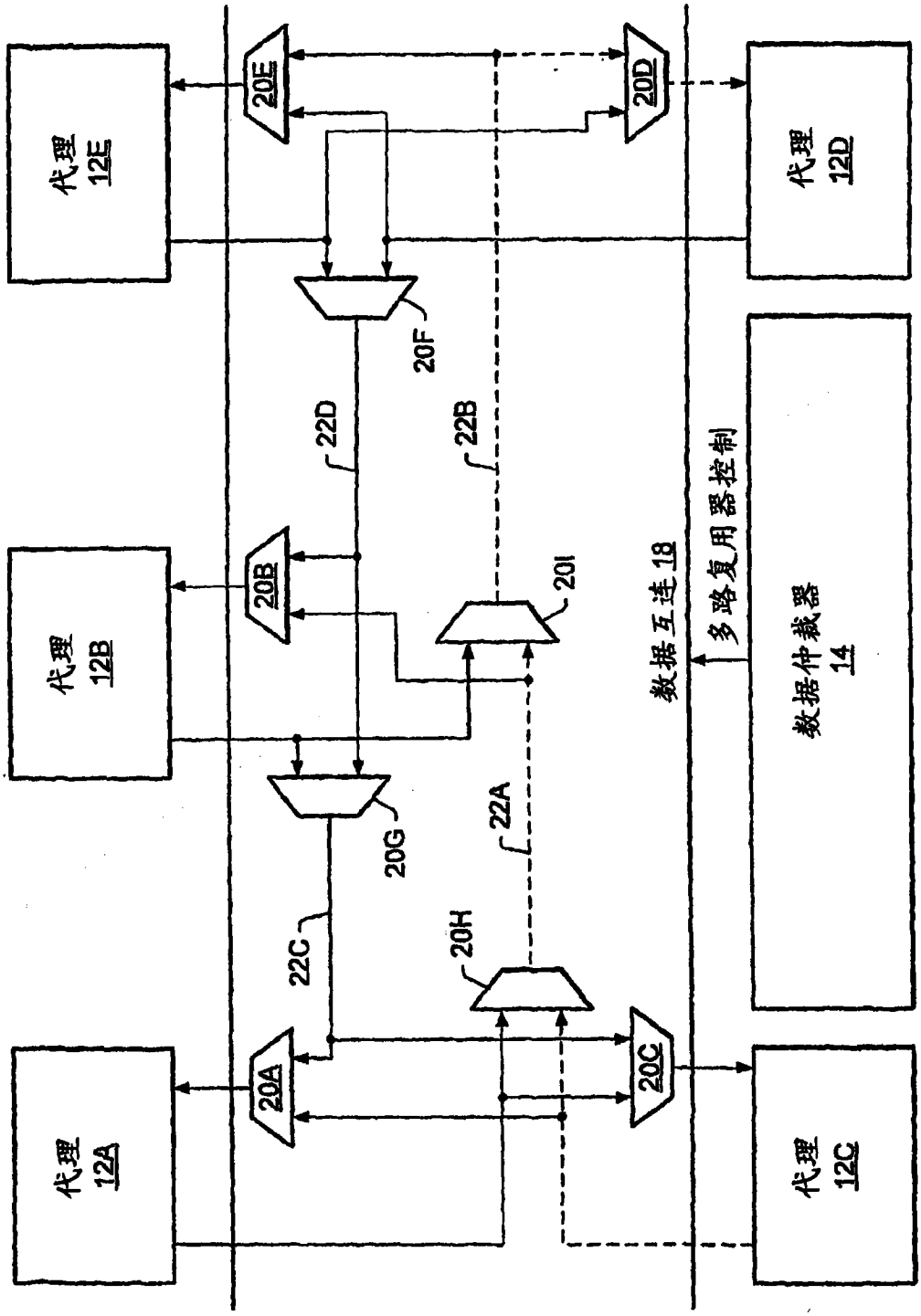

代理 12E
代理 12D
代理 12B
代理 12A
代理 12C
20E
20D
20F
20B
20I
20G
20H
20A
20C
22D
22B
22A
22C
数据互连 18
多路复用器控制
数据仲裁器 14

图 5

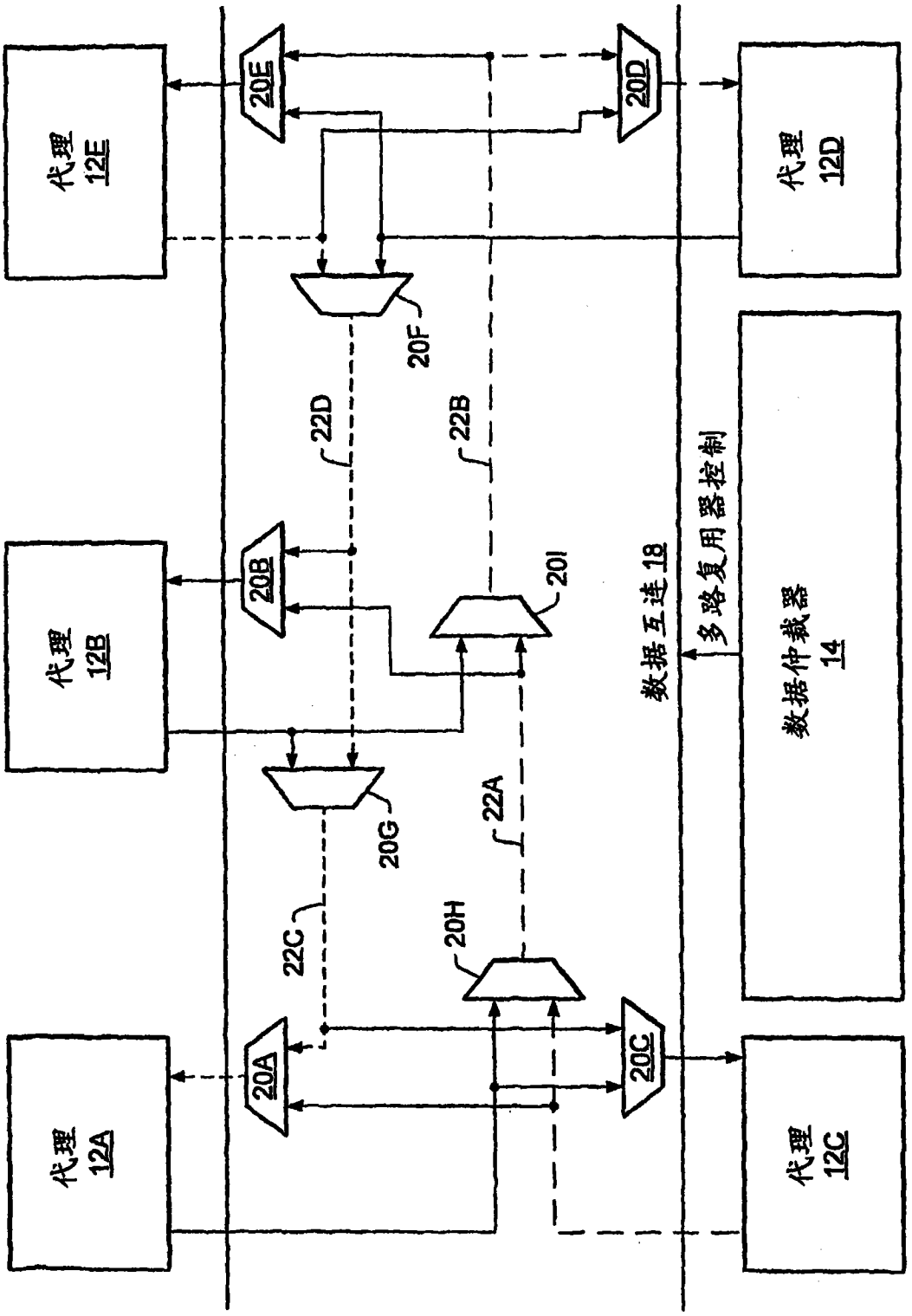

代理
12E
20E
20D
代理
12D
20F
22D
22B
代理
12B
20B
20I
数据互连 18
多路复用器控制
数据仲裁器
14
20G
22A
22C
20H
代理
12A
20A
20C
代理
12C

图 6

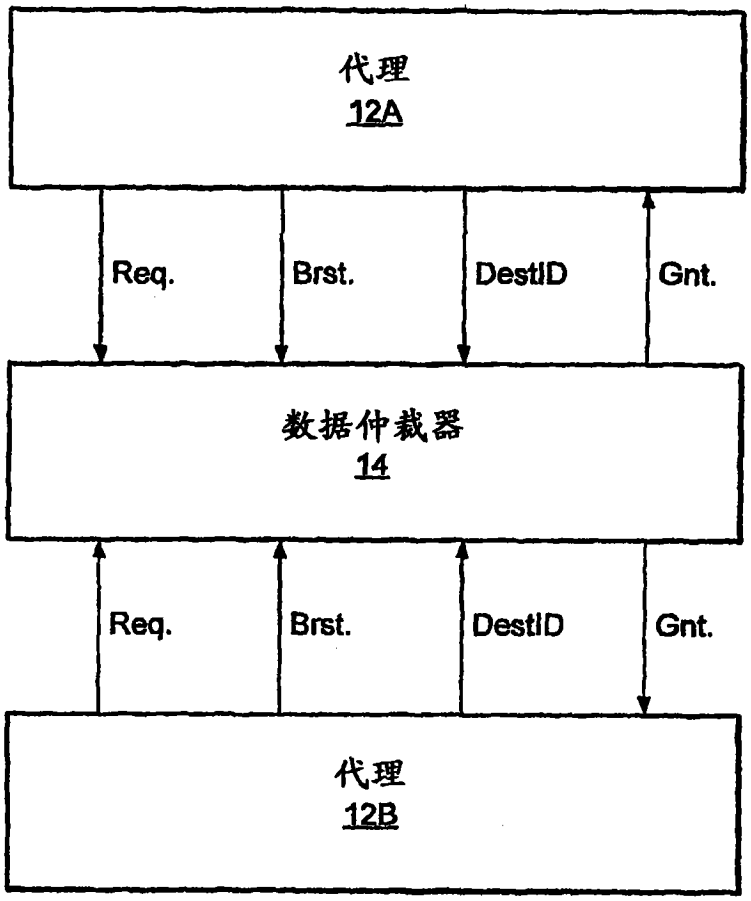


图 7