

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.  
H04L 12/40 (2006.01)



# [12] 发明专利申请公布说明书

[21] 申请号 200680007121.2

[43] 公开日 2008年2月27日

[11] 公开号 CN 101133597A

[22] 申请日 2006.3.2

[21] 申请号 200680007121.2

[30] 优先权

[32] 2005.3.4 [33] EP [31] 05101716.8

[86] 国际申请 PCT/IB2006/050649 2006.3.2

[87] 国际公布 WO2006/092768 英 2006.9.8

[85] 进入国家阶段日期 2007.9.4

[71] 申请人 皇家飞利浦电子股份有限公司

地址 荷兰艾恩德霍芬

[72] 发明人 K·G·W·古森斯 J·迪利森

A·拉杜勒斯库 E·里普克马

P·韦拉格

[74] 专利代理机构 中国专利代理(香港)有限公司

代理人 李静岚 谭祐祥

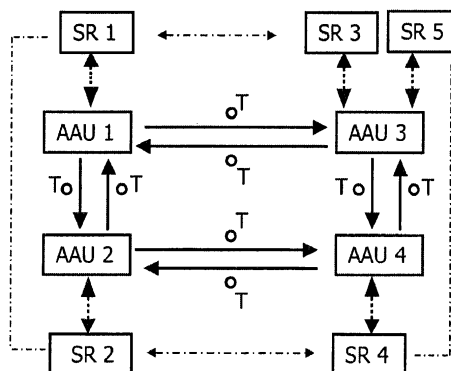
权利要求书 3 页 说明书 17 页 附图 16 页

## [54] 发明名称

用于仲裁共享资源的电子设备和方法

## [57] 摘要

本发明提供一种电子设备，其包括：多个第一共享资源(SR1 - SR4)；以及多个仲裁器单元(AAU1 - AAU4)，每一个仲裁器单元用于执行对所述多个第一共享资源(SR1 - SR4)的至少其中之一的仲裁。所述仲裁器单元(AAU1 - AAU4)之间的通信是在异步的基础上执行的，并且所述第一共享资源之间的数据通信也是在异步的基础上执行的。每一个仲裁器单元(AAU1 - AAU4)适于向至少一个相邻仲裁器单元(AAU1 - AAU4)发送第一令牌(T)，并且适于从至少一个相邻仲裁器单元(AAU1 - AAU4)接收第二令牌(T)，以便实现第一全局时间概念。



1、电子设备，包括：

- 多个第一共享资源 (SR1-SR4)；以及
- 多个仲裁器单元 (AAU1-AAU4)，每一个仲裁器单元用于执行对所述多个第一共享资源 (SR1-SR4) 的至少其中之一的仲裁；
- 其中，所述仲裁器单元 (AAU1-AAU4) 之间的通信是在异步的基础上执行的，并且第一共享资源之间的数据通信也是在异步的基础上执行的；
- 其中，每一个仲裁器单元 (AAU1-AAU4) 适于向至少一个相邻仲裁器单元 (AAU1-AAU4) 发送第一令牌 (T)，并且适于从至少一个相邻仲裁器单元 (AAU1-AAU4) 接收第二令牌 (T)，以便实现第一全局时间概念。

2、根据权利要求1的电子设备，其中，

所述仲裁器单元 (AAU1-AAU4) 适于发送和接收第一和第二令牌 (T) 以实现全局仲裁方案，以用于为所有所述第一共享资源 (SR1-SR4) 提供所需的端到端服务质量。

3、根据权利要求1的电子设备，还包括：

- 多个端口 (OPCU, IPCU)；
- 作为用于耦合所述多个端口 (OPCU, IPCU) 的第一共享资源 (SR1-SR4) 的异步互连装置 (IM, NOC)；
- 其中，所述互连装置 (IM, NOC) 包括：每个均作为第二共享资源的多个互连单元 (NI, R)；以及多个仲裁器单元，每个仲裁器单元均用于执行对所述多个第二共享资源的至少其中之一的仲裁并且用于向至少一个相邻仲裁器单元发送第一令牌 (T)，以及用于从至少一个相邻仲裁器单元接收第二令牌 (T)，以便在所述互连装置 (IM, NOC) 内实现第二全局时间概念。

4、根据权利要求3的电子设备，

所述仲裁器单元用来实现全局仲裁方案，以用于在所述多个端口之间提供所需的端到端服务质量。

5、根据权利要求1的电子设备，

其中，所述第一共享资源 (SR1-SR4) 的至少其中之一是通信资源、存储资源和/或计算资源。

- 6、根据权利要求1的电子设备，其中，  
仲裁器单元（AAC1-AAC4）基于时分多址方案、基于速率控制的仲裁或者基于绝限仲裁来执行。
- 7、根据权利要求1或3的电子设备，其中，  
所述仲裁器单元（AAC1-AAC4）或者所述第一和/或第二共享资源（SR1-SR4）包括D类型端口。
- 8、根据权利要求1或3的电子设备，其中，  
所述仲裁器单元（AAC1-AAC4）或者所述第一和/或第二共享资源（SR1-SR4）包括P类型端口。
- 9、根据权利要求1或3的电子设备，其中，  
所述仲裁器单元（AAC1-AAC4）或者所述第一和/或第二共享资源（SR1-SR4）包括S类型端口。
- 10、根据权利要求3的电子设备，其中，  
所述互连单元（NI, R）是第二共享资源并且包括网络接口（NI）、路由器（R）、桥接器和/或总线。
- 11、根据权利要求1的电子设备，其中，  
所述第一共享资源的至少其中之一包括网络接口（NI）、路由器（R）、桥接器和/或总线。
- 12、根据权利要求1的电子设备，其中，  
所述第一共享资源的其中之一是存储器，并且所述仲裁器单元是存储器控制器。
- 13、根据权利要求1的电子设备，其中，  
所述第一共享资源的其中之一是计算单元，并且所述仲裁器单元是用于硬件或软件多线程的任务调度器。
- 14、根据权利要求3的电子设备，其中，  
所述第一和第二全局时间概念是相同的。
- 15、根据权利要求3的电子设备，其中，  
所述第二全局时间概念是所述第一全局时间概念的倍数或因数。
- 16、根据权利要求1或3的电子设备，其中，  
所述第一和第二令牌（T）指示基于非零增量的逻辑时间的经过，所述增量是静态的或者是动态变化的。
- 17、根据权利要求1或3的电子设备，其中，

所述数据通信与同步通信相组合。

18、用于在具有多个共享资源的电子设备内仲裁共享资源的方法，包括以下步骤：

- 对于多个第一共享资源的至少其中之一执行多个仲裁；
- 其中，仲裁之间的通信是在异步的基础上执行的，并且其中共享资源之间的数据通信是在异步的基础上执行的；
- 其中，每个仲裁包括以下步骤：向至少一个相邻仲裁发送第一令牌，以及从至少一个相邻仲裁接收第二令牌，以便实现第一全局时间概念。

19、使用令牌在仲裁器单元之间传送时间概念，所述仲裁器单元用于在电子设备内对多个第一共享资源的至少其中之一执行多个仲裁，

其中，所述仲裁单元之间的通信是在异步的基础上执行的，并且其中共享资源之间的数据通信是在异步的基础上执行的。

## 用于仲裁共享资源的电子设备和方法

### 技术领域

本发明涉及一种用于仲裁共享资源的电子设备和方法。

### 背景技术

在具有多跳互连的多种新颖的芯片上系统 SoC 体系结构当中，芯片上网络（NOC）已经被证明是可以扩展的互连基础设施，其包括路由器（或交换机）和网络接口（NI 或适配器）、一个或多个管芯（“封装内系统”）或芯片。然而，其中只有几种所提出的体系结构提供受保障的服务（或服务质量 QoS），比如受保障的吞吐量、等待时间或抖动。

这种体系结构的一个例子是由 E. Rijpkema、K. Goossens 和 P. Wielage 的“A router architecture for networks on silicon”（Proceedings of Progress 2001, 2<sup>nd</sup> Workshop on Embedded Systems, Veldhoven, 荷兰, 2001 年 10 月）所描述的 AEthereal 体系结构，其具有无争用的路由或者分布式 TDMA。另一个例子是由 M. Millberg、E. Nilsson、R. Thid 和 A. Jantsch 的“Guaranteed bandwidth using looped containers in temporally disjoint networks within the Nostrum network on chip”（Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE), 2004）给出的 Nostrum 体系结构，其具有利用容器的热土豆算法（hot-potato）路由。J. Liang、S. Swaminathan 和 R. Tessier 的“aSOC: A scalable, single-chip communications architecture”（Proc. Int’l Conference on Parallel Architectures and Compilation Techniques, 2000）给出了一种 aSOC，其在分布式 TDMA 上具有一种变型。

然而，这些芯片上网络 NOC 要求全局的同步性概念，以便通过对分组注入进行调度来避免在芯片上网络 NOC 上的分组争用。一般来说，这些芯片上网络已经按照同步方式实现（即具有一个全局时钟，其是 100%同步的或者均步的）。

已经知道许多其他芯片上网络 NOC 不具有与时间相关的（吞吐量、等待时间、抖动）服务质量 QoS。因此，这些芯片上网络不需要全局的同步性概念，从而它们的实现方式可以是同步的或者是异步的。其例子包括 P. Guerrier 的“Un Réseau D’Interconnexion pour Systèmes Intégrés”（博士论文，Université Paris VI，2000 年 3 月）所提出的同步 SPIN 体系结构、Felicijan 的异步路由器、Arteris 的异步 NOC ([www.arteris.net](http://www.arteris.net)) 和 Sonics 的硅底板 ([www.sonicsinc.com](http://www.sonicsinc.com))。所述同步实现方式（例如 SPIN 和 Sonics）可以很容易地实现全局仲裁方案。所述异步方案（Arteris、Felicijan）不使用全局仲裁方案。

对于服务质量 QoS（即受保障的吞吐量和受保障的等待时间）的实现，对于诸如芯片上网络的多跳互连需要端到端仲裁。这些多跳互连需要多个仲裁器，其中在主设备与从属设备之间（即请求器与响应器之间）的所有仲裁器必须协作以便允许进行端到端仲裁。换句话说，在所述主设备与从属设备之间需要全局的时间概念。这种全局的时间概念可以很容易地在包括同步时钟的芯片上系统 SOC 内实现。然而，芯片上系统不可能 100%同步地实现。这已经导致一种全局异步、局部同步的 GALS 设计的方法。在 Jens Mutersbach 的“Globally-asynchronous locally-synchronous architecture for VLSI systems”（Series in Microelectronics, 120 卷, Hartung-Gorre Verlag Konstanz, 2001）中描述了所述 GALS 体系结构的基本概念。

图 23 示出了根据现有技术不同互连的图示。在图 23a 中示出了通过互连 IM 连接的具有三个 IP 块的芯片上系统。在图 23b 中示出了类似于芯片上网络 NOC 的多跳互连。所述各 IP 模块被耦合到网络 N，该网络 N 包括多个路由器 R 和网络接口 NI。在图 23c 中示出了具有多条总线 B 的多跳互连。该互连包括两条总线 B 并且被耦合到各 IP 块 IP。

在图 24 中示出了 GALS 构建块的一般体系结构。其包括处在局部同步模块 LSM（岛）周围的异步包装器（wrapper）AW。该包装器 AW 允许与该模块 LSM 的环境进行通信，并且生成用于该同步模块 LSM 的局部时钟。在芯片上网络 NOC 的情境中，所述路由器节点 R 和网络接口 NI 以及 IP 块/集群通过这种经过包装的模块 AW 实现。所述时钟的局部生成允许在正在与所述环境进行通信或者要求与该环境进行通信

时延迟下一个时钟周期。提供了端口控制器 IPCU、OPCU 以用于管理在 GALS 系统内的一个块的特定端口上的所有数据传送。所述端口控制器由所述模块 LSM 启用，并且用来同步数据传输和局部时钟相位。为了快速且高效地传输数据，所述端口控制器 IPCU、OPCU 的动作需要独立于所述局部时钟信号。这是通过将它们实施为异步有限状态机而实现的。

为了覆盖针对模块间通信的多样需求，两个端口控制器系列是有用的，即轮询类型和要求类型端口。轮询类型（P 类型）端口仅仅发出针对时钟伸展（clock stretching）的请求，以便防止亚稳定性并且从而确保数据正确性。对于所述时钟的影响尽可能小。要求类型（D 类型）端口也确保所述传送信道上的数据完整性，但是还添加类似于时钟选通的特征。一旦所述要求类型端口被启用，其停止所述局部时钟，并且一旦已经发生了所要求的传送，该要求类型端口就释放该局部时钟。

此外，在图 24 中示出了所述端口类型在一个输入和输出变型中的实现方式。这些端口控制器具有两个握手对：其中一个是在控制器与时钟发生器之间，另一个是在控制器与相应的模块之间。所述端口控制器采用四阶段握手（电平信令）。此外，所述端口使得线路能够采用两阶段协议（转换信令）。Ta 是从端口控制器到 LS 模块的确认信号。其电平表明是否已经发生了数据字的传送。

在图 25 中示出了图 24 的可暂停时钟发生器的方框图。该可暂停时钟发生器 PCG 是 GALS 模块的关键元件。这里示出了一种没有任何测试和调试措施的实现方式。

图 26 示出了根据现有技术的两个局部同步的岛（LSM1，LSM2）之间的单向信道的实现方式。其中采用了上述握手协议。所述端口控制器 PCU 之间的连接是通过握手信号 Ap 和 Rp 建立的。由该握手确认信号 Ap 控制的数据线数据 1 和数据 2 上的锁存器 L 把通信模块 LSM1、LSM2 尽可能地去耦合。向传送信道添加存储器允许发送方在接收块尚未对数据进行采样的情况下恢复操作。

图 27 示出了从 D 输出端到 P 输入端的数据传送的波形。首先，该 D 输出端被启用，其停止其时钟并且发出 Rp+。此时，接收端口尚未被启用。一旦该接收端口被启用，其检测到所述待决握手、停止其时钟

并且确认所述握手。在已经处理了所述外部握手之后，所述两个端口及其相应的模块 LSM 可以恢复其操作。

图中的灰色阴影区域标记出所述数据锁存器 L 的透明阶段 ( $A_p=1$ )。此时，该锁存器 L 打开，接收时钟无效 ( $A_{i2}=1$ )，并且其保持无效的时间远远长于所述锁存器的传播延迟。这样确保了所述数据线上的事件安全地到达接收触发器并且不会发生亚稳定性。保持发送时钟停止 ( $A_{i1}=1$ ) 确保了在所述锁存器是透明时数据 1 保持稳定。

图 28 示出了传统的异步芯片上系统的方框图。其中示出了三个异步电路 AC1-AC3。每一个异步电路 AC1-AC3 只有在数据实际存在于其至少一个输入端上时才被激活。相应地，异步电路 AC1-AC3 不具有任何时间概念，或者仅仅具有其自身的局部时间概念。

图 29 示出了具有三个异步电路 AC1-AC3 的传统的异步系统的执行追踪。这里，所述异步电路 AC1-AC3 被单独地以及独立地触发而没有任何时间概念。在  $t_1$  处，对应于电路 AC1 的输入到达第一电路 AC1。在  $t_2$  处，对应于第二电路 AC2 的输入从第一电路 AC1 到达。在  $t_3$  处，对应于第三电路 AC3 的输入从第二电路 AC2 到达。

## 发明内容

本发明的一个目的是提供一种用于在缺少全局同步时钟的情况下实现服务质量的电子设备和相应方法。

该目的是通过根据权利要求 1 的电子设备、根据权利要求 18 的用于仲裁共享资源的方法以及根据权利要求 19 通过使用令牌在仲裁器单元之间传送时间概念而解决的。

因此，提供一种电子设备，其包括：多个第一共享资源；以及多个仲裁器单元，每一个仲裁器单元用于执行对所述多个第一共享资源的至少其中之一的仲裁。所述仲裁器单元之间的通信是在异步的基础上执行的，并且所述第一共享资源之间的数据通信也是在异步的基础上执行的。每一个仲裁器单元适于向至少一个相邻仲裁器单元发送第一令牌，并且适于从至少一个相邻仲裁器单元接收第二令牌，以便实现第一全局时间概念。

因此，所提出的全局仲裁方案在仲裁单元的数目方面是可扩展的，这是优于在仲裁单元之间使用同步通信的一个优点，其中后一方

案是不可扩展的。

根据本发明的一个方面，所述电子设备还包括多个端口以及作为用于耦合所述多个端口的第一共享资源的异步互连装置。所述互连装置包括：分别作为第二共享资源的多个互连单元；以及多个仲裁器单元，其用于执行对所述多个第二共享资源的至少其中之一仲裁并且用于向至少一个相邻互连组件发送第一令牌，以及用于从至少一个相邻互连组件接收第二令牌，以便在所述互连装置内实现第二全局时间概念。相应地，所述全局时间概念也可以在所述互连中实现，从而允许在异步互连内（从而在各端口之间）实现服务质量。

本发明还涉及一种用于在具有多个第一共享资源的电子设备内仲裁共享资源的方法。对于所述多个第一共享资源的至少其中之一执行多个仲裁。所述仲裁之间的通信是在异步的基础上执行的。所述第一共享资源之间的数据通信也是在异步的基础上执行的。每个仲裁包括以下步骤：向至少一个相邻仲裁发送第一令牌，以及从至少一个相邻仲裁接收第二令牌，以便实现第一全局时间概念。

本发明还涉及使用令牌在仲裁器单元之间传送时间概念，所述仲裁器单元用于在电子设备内对多个第一共享资源的至少其中之一执行多个仲裁。所述仲裁单元之间的通信是在异步的基础上执行的。所述第一共享资源之间的数据通信也是在异步的基础上执行的。这样做是有利的，因为令牌通常仅仅传送数据而不传送时间。

本发明是基于提供分布式全局仲裁方案的异步实现方式的思想（例如存储器控制器和芯片上网络 NOC 仲裁方案，基于贴片的方法中的通信辅助和芯片上网络 NOC 仲裁方案）。提供了一种可以按照分布式方式异步地实现的全局同步性概念（或仲裁方案）。其可以被应用来实现具有同样需要全局同步性概念的其他仲裁方案的芯片上网络 NOC（或者更一般来说是通信基础设施，比如分级/桥接总线），比如速率控制的方案（例如虚拟电路排队或输出排队）和基于绝限的方案。基本上，所述基本思想是芯片上网络 NOC 可以通过由在每个逻辑同步单元（或者时间步或数据流发射）交换令牌的组件（例如路由器、网络接口）构成而实现全局同步性概念（或全局时间表）。

本发明主要针对以下情况：a) 异步芯片上网络 NOC 以芯片上网络 NOC 同步速率的倍数或因数耦合 IP 块，即需求驱动；b) 异步芯片上网

络 NOC 耦合不以芯片上网络 NOC 同步速率的倍数或因数操作的 IP 块 IP，即数据驱动；以及 c) 异步芯片上网络 NOC 耦合不以芯片上网络 NOC 同步速率的倍数或因数操作的 IP 块 IP，即事件驱动。

在从属权利要求中描述了本发明的其他方面。

将参照下面描述的实施例并且参照附图来描述本发明的这些和其他方面。

#### 附图说明

图 1 示出了根据本发明的第一实施例的异步系统的方框图；

图 2 示出了根据第一实施例耦合几个 IP 块的多跳互连的方框图；

图 3a-d 示出了具有路由器 R 和作为互连的网络接口 NI 以及 IP 块的芯片上网络；

图 4 示出了用于根据第二实施例耦合三个 IP 块 IP 的芯片上网络 NOC 的方框图；

图 5 示出了 IP 块 IP、网络接口 NI 和路由器 R 的方框图；

图 6 示出了根据图 5 的 IP 块 IP、网络接口 NI 和路由器 R 的方框图；

图 7 示出了图 4 的两个相邻路由器的更详细的方框图；

图 8 示出了图 4 的两个相邻路由器的另一个详细方框图；

图 9 示出了根据第二实施例的图 4 的路由器 R 的方框图；

图 10 示出了所述芯片上网络的一部分的方框图；

图 11 示出了根据第三实施例的芯片上网络的一部分的方框图；

图 12 示出了所述 IP 块 IP 和网络接口 NI 的更详细的方框图；

图 13 示出了图 4 的网络接口的更详细的方框图；

图 14 示出了根据第四实施例的芯片上网络的一部分的方框图；

图 15 示出了根据第四实施例的图 14 的 IP 块 IP 和网络接口的更详细的方框图；

图 16 示出了图 14 的网络接口的更详细的方框图；

图 17 示出了根据第五实施例的耦合到 IP 块的芯片上网络的一部分的方框图；

图 18 示出了图 17 的 IP 块 IP 和网络接口 NI 的更详细的方框图；

图 19 示出了图 17 的网络接口的更详细的方框图；

图 20 示出了根据第七实施例的两个局部同步的岛 (LSM1, LSM2) 之间的单向信道的实现方式的方框图;

图 21 示出了对应于事件驱动的同步的定时信号的图示;

图 22 示出了根据第六实施例的耦合几个 IP 块的芯片上网络;

图 23 示出了根据现有技术不同互连的图示;

图 24 示出了 GALS 构建块的一般体系结构;

图 25 示出了图 24 的可暂停时钟发生器的方框图;

图 26 示出了根据现有技术的两个局部同步的岛之间的单向信道的实现方式;

图 27 示出了从 D 输出端到 P 输入端的数据传送的波形;

图 28 示出了传统的异步芯片上系统的方框图; 以及

图 29 示出了具有三个异步电路的传统异步系统的执行追踪。

### 具体实施方式

提供 QoS (特别是有界限等待时间) 的本方法在于构成无争用路由的基础的数据流模型, 正如 E. Rijpkema、K. Goossens 和 P. Wielage 的 “A router architecture for networks on silicon” (Proceedings of Progress 2001, 2<sup>nd</sup> Workshop on Embedded Systems, Veldhoven, 荷兰, 2001 年 10 月) 中所记载的那样。所述逻辑同步单元可以是微片 (flit), 正如 E. Rijpkema、K. G. W. Goossens、A. Radulescu、J. Dielissen、J. van Meerbergen、P. Wielage 和 E. Waterlander 的 “Trade offs in the design of a router with both guaranteed and best-effort services for networks on chip” (Proc. Design, Automation and Test in Europe Conference and Exhibition (DATE), 第 350-355 页, 2003 年 3 月) 中所解释的那样。正如所引用的文章中所解释的那样, 该方案可以在同步的基础上实现, 但是也可以根据本发明被实现为异步实现方式。

图 1 示出了根据本发明的第一实施例的异步系统的方框图。该系统包括几个共享资源 SR1-SR4 和几个仲裁器单元 AAU1-AAU4。仲裁器间通信 (即所述仲裁器之间的通信) 是彼此异步地执行的。所述共享资源 SR1-SR4 可以在它们之间传送数据。每个仲裁器单元 AAU1-AAU4 在令牌 T 存在于其输入端上时激活。相应地, 所述异步仲裁器 AAU1-

AAU3 具有全局的共享时间概念。结果，所述仲裁器单元 AAU 可以仲裁（见虚线）与所述仲裁器单元相关联的共享资源。特别地，仲裁器单元 AAU1 与共享资源 SR1 相关联并且对其进行仲裁。仲裁器单元 AAU2 与共享资源 2 相关联并且对其进行仲裁。仲裁器单元 AAU3 与共享资源 SR3 和 SR5 相关联并且对其进行仲裁。仲裁器单元 AU4 与共享资源 4 相关联并且对其进行仲裁。对于所述仲裁器单元 AU1-AU4 的仲裁是按照全局同步或协调的方式执行的。所述共享资源 SR1-SR4 可以在它们之间传送数据。仲裁器单元 AAU1-AAU4 仅仅与相邻的仲裁器单元通信，以便实现全局时间概念。因此，所提出的全局仲裁方案在仲裁单元的数目方面是可扩展的，这是优于在仲裁单元之间使用同步通信的一个优点，其中后一方案是不可扩展的。

所述全局时间概念描述了其中（可能是每一个）仲裁器单元知道（所有）其他仲裁器单元的状态或状况的情形。因此，如果一个仲裁器单元处在步骤 3，则所有其他仲裁器也都将处在步骤 3。

图 2 (a) 和 2 (b) 示出了根据第一实施例耦合几个 IP 块的多跳互连 IM 的方框图。该互连 IM 包括几个路由器 R 和网络接口 NI，所述网络接口作为用于把所述路由器连接到所述 IP 块 IP 的互连组件或互连节点。

在启动/重置时，路由器 R（或其他芯片上网络 NOC 组件）的异步实现方式首先导致在每个输出端（即如图 2a 中所示的到其他芯片上网络 NOC 组件的每条链接）上产生令牌 T，并且随后（永久性地或者直到重置）从每个输入端读取令牌，如图 2b 中所示的那样处理所述令牌，并且随后在每个输出端上产生令牌 T。这样，所有路由器以锁定步进的方式前进，例如处在相同的 TDMA 时隙中。这样做的效果是实现了一种全局仲裁方案，其仅仅具有与往往是局部的邻居的异步握手。产生和消耗令牌对应于需求驱动（请求-确认）的交互风格（握手）。

这种思想也可以用于速率控制的和基于绝限的全局仲裁方案。应当注意，令牌 T 或者包含数据，或者是空的。即使在没有数据的情况下也必须发送所述令牌，以便保持所述同步性概念。

现在描述对于异步互连 IM 的服务质量的实现方式。所述芯片上网络 NOC 组件的前进速度将与最慢的组件相同，从而构成作为整体的该芯片上网络 NOC 的同步速率。每秒的迭代次数与“实际时钟速度”相

关。举例来说，一个同步步骤可以对应于三个时钟周期。所述同步速率是在所述芯片上网络 NOC 内部生成的（即由所述最慢组件生成，而不是由外部已知时钟施加（对于完全同步的芯片上网络 NOC 就是后一种情况））这一事实并不成为问题，并且不会使得 QoS 的思想无效，这是因为该网络内的所有异步组件都是利用所考虑的特定目标操作频率来设计的。

作为用于说明的例子，所述目标频率可以是 166M 同步/秒或者 166 兆微片/秒；其中一个微片可以是 3 个分别具有 32 比特的字。通过采取适当的余量（或者“过度设计”），比如 20%，所述组件应当运行在 200M 同步/秒或者 200M 微片/秒，但是所述最慢组件的运行速度将肯定快于预定的 166M 同步/秒或 500M 字/秒，从而导致至少 166 M 同步/秒或 500M 字/秒的受保障吞吐量以及潜在地操作更快的芯片上网络 NOC。所述实际余量将取决于芯片处理的精度、最坏情况操作条件等等。这一系列推理对于同步和异步模块/IC 都是同样可以接受的。

图 3a-d 示出了根据第二实施例的芯片上网络，其具有路由器 R、作为互连的网络接口 NI 以及耦合到相应的网络接口 NI 的 IP 块 IP。所述 IP 块可以利用不同的令牌速率而操作在倍数速率（或因数速率）下。相应地，图中示出了具有运行在芯片上网络 NOC 同步速率的倍数或因数下的 IP 块 IP 的异步多跳互连 IM 的服务质量（QoS）。在图 3a 中，所述 IP 块 IP 运行在所述互连的双倍速率下，因此产生两个同步令牌 T，而所述路由器 R 和网络接口 NI 仅仅产生单一令牌 T。

在所述两种情况下，所述解决方案仅仅适用于运行在芯片上网络 NOC 频率的倍数或因数下的 IP 块。此外，在同步情况下，不再能够用单一同步时钟服务于属于芯片上网络 NOC 的所有 IP 块。

在同步情况下，对于 IP 和芯片上网络 NOC（其操作在一个时钟上）使用多个独立时钟依赖于数据同步，也就是说，使用两个串联的触发器从（所述 IP 的）一个时钟域跨越到（所述芯片上网络 NOC 的）另一个时钟域，或者反之亦然。这可以被称作数据驱动的同步。尽管这样的解决方案将有效果，但是其并不是最佳的，因为在对来自另一个时钟域的数据进行采样时有可能会发生错误。随着这两个频率提高，该解决方案变得更差。

在异步情况下，对应于所述 IP 和芯片上网络 NOC（其利用逻辑同

步性概念操作)的多个独立时钟的同步可以通过需求驱动的同步、数据同步或者事件驱动的同步来解决。第一种解决方案无法处理所有的时钟比值、可变时钟等等。第二种解决方案引入了错误数据的可能性。第三种解决方案没有上述问题。

在数据驱动的同步的情况下,每一个模块在其每一条连接到其他模块的通信线上、在其推进其时钟时对所述通信线进行采样。这可以利用双触发器方案来实现。引入了具有错误数据样本的潜在问题。特别地,利用所述两个触发器采样的比特有一定的概率是错误的。通过使用更多的触发器可以降低该概率,其代价是等待时间延长。现在,应当注意到对于所述系统上的每一个数据驱动的端口/链接,该错误概率都存在,并且这些概率相加,其中所述错误不彼此抵消或彼此补偿。

在图2和图3中示出了需求驱动的同步,并且其构成各芯片上网络NOC模块(NI和路由器)之间的一个实施例。在所传输的数据中将不会发生错误。

图4示出了用于根据第二实施例耦合三个IP块IP的芯片上网络NOC的方框图。该芯片上网络包括三个网络接口NI以及三个路由器R。所述路由器R以及网络接口NI通过D类型端口D通信。

图5示出了IP块IP、网络接口NI和路由器R的方框图。IP块IP与网络接口NI之间的接口是基于可暂停时钟方案实现的,而网络接口NI与路由器R之间的接口则是基于需求驱动的同步实现的。从IP块IP到网络接口NI的通信是通过来自IP块的请求信号ip2ni\_有效以及和请求数据reqdata一起来来自网络接口的响应信号ip2ni\_确认实现的。从网络接口NI到IP块IP的通信是通过来自网络接口NI的请求信号ni2ip\_有效以及和响应数据respdata一起来自IP块IP的响应信号ni2ip\_确认实现的。此外,从网络接口NI到路由器R的通信是通过来自网络接口NI的请求信号ni2r\_有效以及与数据ni2r\_数据一起来自路由器R的响应信号r2ni\_确认实现的。从路由器R到网络接口NI的通信是通过来自路由器的请求信号r2ni\_有效以及与数据r2ni\_数据一起来来自网络接口的响应信号r2ni\_确认实现的。

网络接口NI包括连接到互斥单元mutex的异或单元XOR,该互斥单元又连接到翻转单元TU。该翻转单元TU的输出连接到逻辑单元LU,并且构成响应信号ip2ni\_确认。具有延迟线和反相器DLI的反馈环路

耦合到该互斥单元 mutex。该具有两个输入端的互斥元件 mutex 是一个标准的异步构建块。

网络接口 NI 的响应部分按照相应的方式设置，其中没有所述延迟和反相器 DLI。

基本上，每当来自 IP 的外部事件到达 NI 时就翻转一个状态元件，以便存储（该 IP 所传送的）该信息，从而该信息可以被逻辑块使用。随后通过信号 ip2ni\_确认向该 IP 块 IP 确认该事件。向该 IP 块的确认处在关键路径内，并且必须尽可能地快。为此原因，翻转元件 TU 立即降低（进入互斥元件的）请求线，而不要求与可能非常慢的 IP 块进行任何交互。随后该 IP 块可以在空闲时对该确认做出响应。逻辑单元 LU 使用关于所述请求线 ip2ni\_有效已经为高的信息，以便例如读出请求数据。

图 6 示出了根据图 5 的 IP 块 IP、网络接口 NI 以及路由器 R 的方框图。然而，根据图 6，可以重复使用同步 NI 核心 NSNI。图 6 的其他设置对应于图 5 的设置。换句话说，如果将要实现异步网络接口，则这可以通过以下措施实现：使用同步网络接口的典型结构并且提供某种内壳，以便允许在这种典型结构之上与 IP 块 IP 进行通信。

应当注意到，上面提到的操作通常根本不停止 NI 的内部生成的时钟。

图 7 示出了图 4 的两个相邻路由器的更为详细的方框图。各路由器 R 之间的接口是基于需求驱动的同步实现的。各路由器之间的通信是通过请求信号“有效”以及与请求数据“数据”一起的响应信号“确认”来实现的。

所述路由器包括连接到互斥单元 mutex 的异或单元 XOR，该互斥单元又连接到翻转单元 TU。该翻转单元 TU 的输出连接到同步路由器核心 NSR。具有延迟线和反相器 DLI 的反馈环路耦合到该互斥单元 mutex。该具有两个输入端的互斥元件 mutex 是一个标准的异步构建块。

图 8 示出了图 4 的两个相邻路由器的另一个详细方框图。该路由器包括普通的同步路由器核心 NSR 以及可暂停时钟发生器 PCG。

图 9 示出了根据第二实施例的图 4 的路由器 R 的方框图。该路由器 R 包括需求驱动接口，所述需求驱动接口把该路由器 R 耦合到

各相邻路由器 R 并且还有可能耦合到各相邻网络接口 NI。该路由器 R 包括作为核心的普通同步路由器 NSR，其具有输入端口控制单元 IPCU 和输出端口控制单元 OPCU。该输入端口控制单元 IPCU 以及该输出端口控制单元 OPCU 被实现为 D 类型端口。这两个端口控制单元 IPCU、OPCU 被耦合到可暂停时钟发生器 PCG。该路由器 R 与相邻路由器之间的通信在其输入侧是通过握手信号 AP1 和 RP1 执行的，并且该路由器接收输入数据 data1。在该路由器 R 的输出侧，与相邻路由器 R 的通信是通过握手信号 AP2 和 RP2 执行的，并且数据 data2 被前送到后续路由器。

在图 10 的上半部分中示出了芯片上网络的一部分的方框图。图 10 示出了根据第二实施例的芯片上网络的一部分。这里示出了（充当主设备的）主 IP 块 MIP、主网络接口 mNI、一个或多个路由器 R、从属网络接口以及（充当从属设备的）从属 IP 块 SIP。这些单元通过链接 L1、L2、L3、L4 相连，所述链接是逻辑同步的，也就是说它们处在相同的时钟域内或者在固定速率下同步。换句话说，所述 IP 块 MIP、SIP 以及所述互连 mNI、R、sNI 是逻辑同步的。任何时间相关的 QoS 都可以从主 IP 块 MIP 延伸到从属 IP 块 SIP。

图 10 的下半部分示出了所述芯片上网络的相同部分，但是这里仅仅有互连 IM、主网络接口 mNI、路由器 R 和从属网络接口 sNI 是逻辑同步的。任何时间相关的 QoS 都将从主网络接口 MNI 延伸到从属网络接口 SNI，即不是从主 IP 块 MIP 延伸到从属 IP 块 SIP，这是由于链接 L1 和 L4 不是同步的。必须对通过这两个链接 L1、L4 的通信的数据进行采样以便实现数据驱动的同步，或者必须对相应的时钟进行同步以便实现事件驱动的同步。

现在考虑（同步的或异步的）芯片上网络 NOC 与各 IP 块之间的交互。由所述芯片上网络 NOC 实现的 QoS（例如受保障的等待时间）将仅仅从主 mNI 延伸到从属 mNI。如果主设备（从属设备）与芯片上网络 NOC（即主（从）NI）同步地操作，即处在相同的或所导出的时钟域内（即没有时钟域跨越），则所述 QoS 保障将从主设备延伸到从属设备。类似地，如果所述芯片上网络 NOC 是异步的，并且所述主设备（从属设备）把每个（固定倍数）时间步与主（从）NI 同步，则该 QoS 将从主 MIP 延伸到从属 SIP。相应地，这将对应用于异步（多速率 SDF）情况，

即需求驱动的不同步。

在图 11 中示出了根据第三实施例的芯片上网络的一部分的方框图。应当注意，为了说明本发明，其中仅仅示出了一个 IP 块 IP、一个网络接口 NI 以及仅仅一个路由器 R。IP 块 IP 与网络接口之间的通信是通过 D 类型接口来执行的，该 D 类型接口具有处在 IP 块 IP 以及网络接口 NI 内的 D 类型端口 D。网络接口 NI 与其相关路由器 R 之间的通信也是基于具有 D 类型端口 D 的 D 类型接口来执行的。这同样适用于路由器间通信。相应地，在芯片上网络 NOC 与 IP 块 IP 之间示出了需求驱动的通信。这里，IP 块按照与芯片上网络相同的速率或者其倍数-因数速率来执行其处理。

在图 12 中示出了 IP 块 IP 和网络接口 NI 的更加详细的方框图。该 IP 块 IP 包括一个普通同步 IP 核心 NSIP。输入端口控制单元 IPCU 以及输出端口控制单元 OPCU 耦合到该普通同步 IP 单元 NSIP 的端口控制单元 OPCU 和 IPCU。二者都被实现为 D 类型端口。所述端口控制单元耦合到可暂停时钟发生器 PCG。该网络接口 NI 包括一个普通同步网络接口核心 NSNI，其具有输入端口控制单元 IPCU 以及输出端口控制单元 OPCU。所述两个端口控制单元都耦合到可暂停时钟发生器 PCG。从 IP 块到网络接口 NI 的通信是通过握手信号 AP1 和 RP1 以及数据 data1 处理的，其中所述数据 data1 被从 IP 块 IP 传送到网络接口 NI。从网络接口到 IP 块的通信是通过第二握手信号 AP2 和 RP2 以及数据 data2 控制的，其中所述数据 data2 被从网络接口 NI 传送到 IP 块 IP。相应地，在 IP 块 IP 与网络接口 NI 之间实现需求驱动的接口。

图 13 示出了图 11 的网络接口的更加详细的方框图。该网络接口包括被实现为 D 类型端口的到 IP 和路由器的需求驱动的接口。

图 14 示出了根据第四实施例的芯片上网络的一部分的方框图。该芯片上网络的基本结构对应于根据图 11 的结构。然而，IP 块 IP 与网络接口 NOC 之间的接口是 P 类型接口。因此，该 IP 块包括两个 P 类型端口，该网络接口 NI 也包括两个 P 类型端口。网络接口与路由器之间的通信以及路由器间通信是基于具有 D 类型路由器的 D 类型接口。

图 15 示出了根据第四实施例的图 14 的 IP 块 IP 和网络接口的更加详细的方框图。图 15 的 IP 块和网络接口的基本结构对应于根据图 12 的网络接口和 IP 块的结构。然而，端口控制单元 OPCU 和 IPCU 被

实现为 P 类型端口控制单元，从而在 IP 块与网络接口之间实现了 P 类型接口。相应地，在 IP 块 IP 与网络接口 NI 之间实现了事件驱动的接口。从 IP 块到网络接口的通信是通过第一握手信号 AP1 和 RP1 以及数据 data1 控制的，从网络接口到 IP 块的通信是通过第二握手信号 AP2 和 RP 以及数据 data2 控制的，其中所述数据 data2 被从网络接口 NI 传送到 IP 块 IP。

图 16 示出了图 14 的网络接口的更加详细的方框图。该网络接口包括分别被实现为 P 类型端口和 D 类型端口的一个事件驱动的接口（用于到 IP 的通信）和需求驱动的接口（用于到路由器的通信）。

图 17 示出了根据第五实施例的耦合到 IP 块的芯片上网络的一部分的方框图。该芯片上网络 and IP 块的结构对应于图 11 和图 14 的结构。网络接口 NI 之间的通信以及路由器间通信是基于具有 D 类型端口的 D 类型接口。然而，IP 块与网络接口之间的通信是利用数据驱动的接口执行的，其中该 IP 块包括 S 类型端口，该网络接口包括 P 类型端口。这里，该 IP 块的运行速率可以独立于该芯片上网络的速率。

图 18 示出了图 17 的 IP 块 IP 和网络接口 NI 的更加详细的方框图。图 18 的 IP 块以及网络接口的基本结构对应于图 12 和图 16 的基本结构。然而，该 IP 块包括 S 类型端口控制单元 OPCU、PCU，而该网络接口则包括 P 类型端口控制单元 IPCU、OPCU。

图 19 示出了图 17 的网络接口的更加详细的方框图。该网络接口包括一个被实现为 S 类型端口的需求驱动的接口和被实现为 D 类型端口的需求驱动的接口。

图 20 示出了根据第七实施例的两个局部同步的岛 (LSM1, LSM2) 之间的单向信道的实现方式的方框图。输出端口控制器 OPCU 与输入端口控制器 PCU 之间的连接是通过握手信号 Ap 和 Rp 建立的。由握手确认信号 Ap 控制的数据线 data1、data2 上的锁存器 L 尽可能地把通信模块 LSM1、LSM2 去耦合。

这里，S 类型端口被用于局部同步的岛 LSM1、LSM2 的输出和输入端口控制器 OPCU、PCU，所述岛运行在一个无法被停止的时钟下。这种时钟典型地是外部生成的时钟。这种局部同步的岛 LSM1、LSM2 不具有可暂停时钟发生器 PCG。该局部同步的岛 LSM1、LSM2 可以（通过翻转 En 信号）而使得所述 S 类型端口能够执行数据通信。当信号 Ta 随

之翻转时，数据通信已经被执行。S 类型端口的实现方式基本上是自由运行的 P 类型端口，这是因为 S 类型端口不会干扰任何时钟。使用触发器 FF 以使得信号 Ta 与 LSM 时钟信号同步。因此，取代由 P 和 D 类型端口采用的时钟同步，采用了数据同步。

图 21 示出了对应于事件驱动的同步的定时信号图。图 21 中示出的时钟 C 是由延迟线和反相器 DLI 生成的。如果事件 E1 在时钟沿很久之前到达，则时钟 C 不被延迟，这是由于互斥单元 mutex 接收该事件和该时钟沿的间隔足够远（事件已经在最小（恒定）时间内发生），从而避免了亚稳定性。只有当到来的事件 E2 的到达时间靠近时钟沿（同时，在极限内）时，该互斥元件才需要仲裁哪一个先到（或者在严格重合的情况下仲裁允许哪一个先通过）。这可能会花费一些时间（由于亚稳定性），并且可能会因此而把时钟延迟 ED，即图 14 中的第二事件。这种情况很少发生。对时钟进行延迟的时刻之间的时间可以被计算，其取决于 IP 和 NI 的时钟速度（并且随着更高的速度而减小）。

响应路径以类似的方式工作。按照所述方式实现请求和响应路径，以便确保 NI 是可暂停的（即其局部时钟可以被停止），但是仅仅可以暂停很短的时间。应当注意，仅有 NI 被停止，任何相连路由器的时钟都不被停止，只是它们的需求驱动的握手可能会花费略微更长的时间。如果被停止很短时间的 NI 连接到一个快速路由器（例如由于工艺变化或温度差异），则该 NI 的短暂停滞可以由该路由器补偿。这样，与全局钟控的同步网络相比，分布式异步芯片上网络 NOC 可以更好地处理暂停，在所述全局钟控的同步网络中，由于停滞的 NI 所招致的所有任何延迟都无法再被补偿。这仅仅影响等待时间而不影响吞吐量，所述吞吐量总是被减少到最慢反馈环路。

如果我们把由于到来的事件所造成的时钟延迟考虑为错误，则与上面描述的数据驱动的同步情况相反，这些错误不会相加。也就是说，如果多个 NI 同时被延迟，则所述芯片上网络 NOC 作为一个整体将仅仅被这些延迟当中的最坏的一个所延迟，而不是被这些延迟的总和所延迟。这是事件驱动的同步优于数据驱动的方案的一个优点。

如果我们把 NI 速度的大小确定得过大，例如大 5%，则对应于单一时钟周期的失败之间的平均时间被缩短，这是因为所述互斥元件 mutex 有 5% 的附加时间来解决。如果考虑多个连续的时钟周期（例如 3 个），

则 NI 在 3 个时钟周期之后过慢的概率要低于该 NI 在 1 个时钟周期之后过慢的概率，这是因为如果一个延迟事件发生在 3 个时钟周期内，则有 3x5%的宽松度 (slack) 来解决，而不是仅有 5%。对于在 3 个周期期间的两个延迟事件也是类似的（它们各具有 1.5x5%的宽松度）。对于三个延迟事件，没有附加的宽松度可用。这是事件驱动的同步优于数据驱动的方案的一个优点。

相应地，芯片上网络 NOC 的物理（定时和钟控）方面被放松：对于芯片上网络 NOC 不需要全局时钟。芯片上网络 NOC 在组件数目方面的可扩展性更好，因此性能也更好。所述 IP 和芯片上网络 NOC 可以在任何独立的速度下运行（对于事件驱动的 IPNOC 同步）而不需要害怕错误数据，而是具有在错失时间绝限方面的失败之间的先验已知的平均时间。

另一方面，异步电路的测试要难于同步电路。标准硬件后端流程（合成、定时检验等等）更适于同步设计而不是异步设计。

图 22 示出了根据第六实施例耦合几个 IP 块的芯片上网络。各网络接口与路由器之间的通信以及路由器间通信是基于具有 D 类型端口的 D 类型接口，也就是说，该芯片上网络的各组件之间的接口是需求驱动的。相应的 IP 块与其相关网络接口之间的接口示出了根据第三（左）、第四（中）和第五（右）实施例的接口。相应地，根据第三、第四和第五实施例的接口也可以被应用在单一芯片上网络中。

在根据第五实施例的基于所介绍的 GALS 技术的芯片上网络 NOC 中，为了实现 NOC 与各 IP 之间的需求驱动的通信，在各 NI 与各 IP 之间的信道的两侧都使用 D 类型端口。由于所有的信道都使用 D 类型端口，从而保障了所有块的连贯进展。由于 D 类型端口是 100%确定性的，因此所得到的数量性能也是如此。

可以从文献中得知用于提供 QoS 的其他方法（从一般网络）（特别是由 H. Zhang 的“Service disciplines for guaranteed performance service in packet-switching networks” (Proceedings of the IEEE, 83(10):1374-96, 1995 年 10 月) 所描述的速率控制的方案；以及由 J. Rexford 的“Tailoring Router Architectures to Performance Requirements in Cut-Through Networks” (博士论文, University of Michigan, department of

Computer Science and Engineering, 1999 年) 描述的基于绝限的方案), 但是还没有关于实现这些方案的芯片上网络 NOC 的记录。这些方法也依赖于全局同步性概念。

应当注意到, 上面描述的实施例说明而不是限制本发明, 在不背离所附权利要求书的范围的情况下, 本领域技术人员将能够设计出许多替换实施例。在权利要求书中, 括号中的任何附图标记不应被理解成限制该权利要求。“包括”一词不排除未在权利要求中列出的其他元件或步骤。元件前面的“一个”不排除多个元件的存在。在列举几种装置的设备权利要求中, 这些装置当中的几个可以由同一硬件项具体实现。在彼此不同的从属权利要求中阐述某些措施并不意味着不能使用这些措施的组合来获益。

此外, 权利要求书中的附图标记不应被理解成限制权利要求的范围。

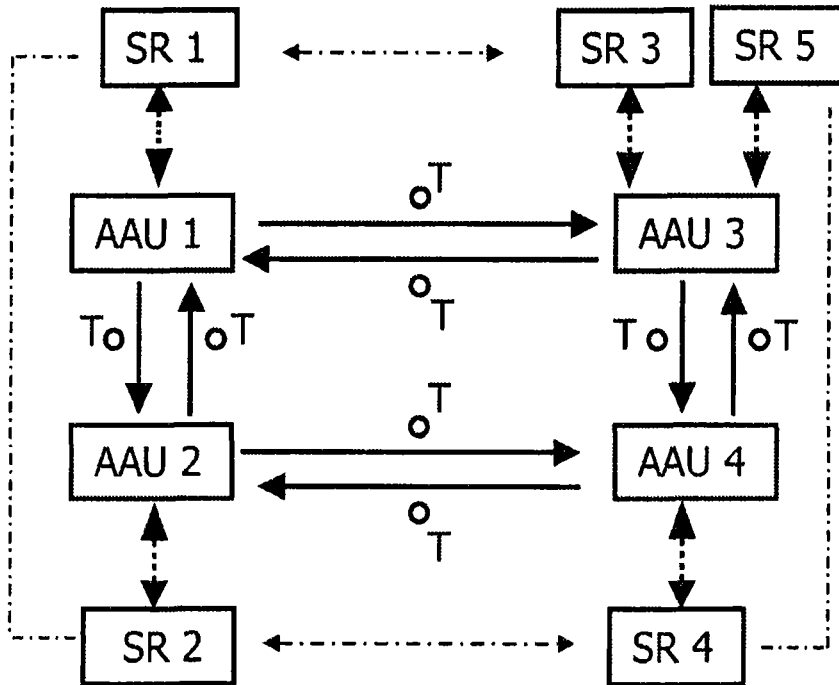


图 1

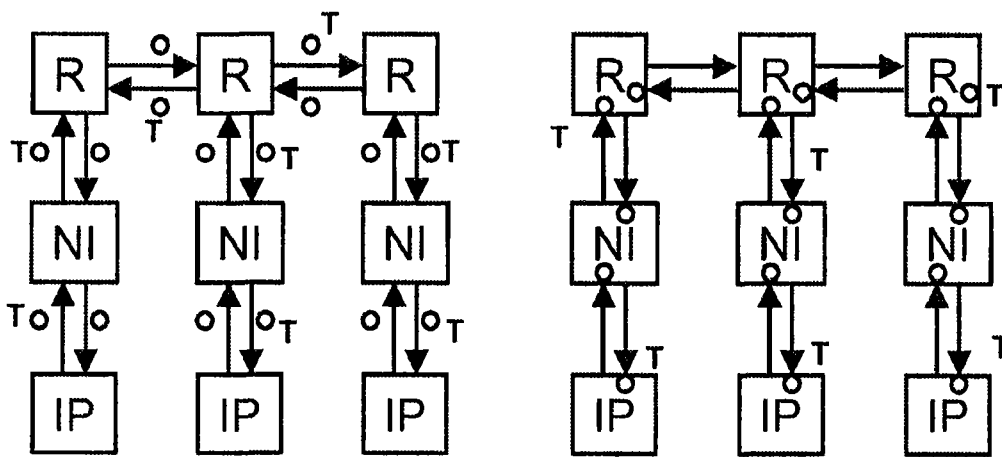


图 2a

图 2b

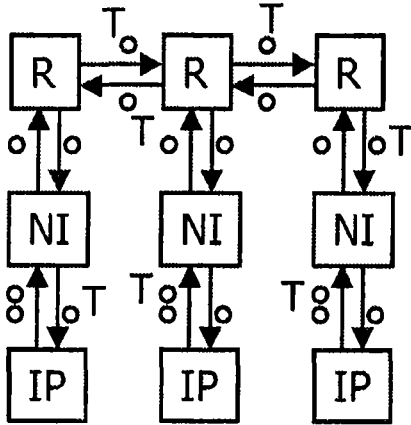


图 3a

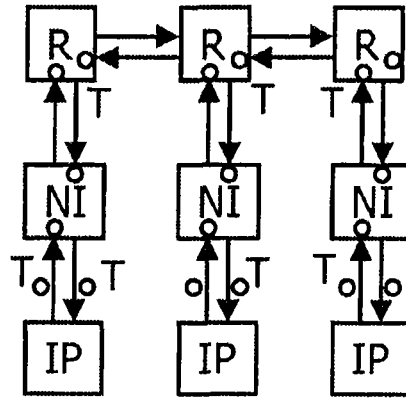


图 3b

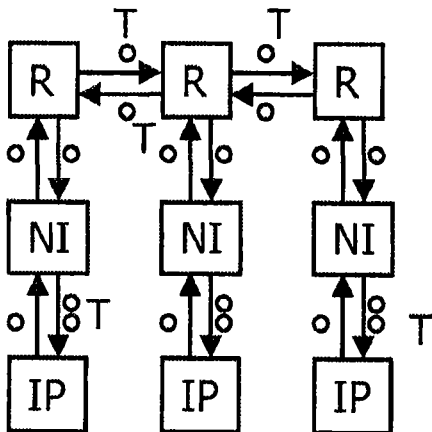


图 3c

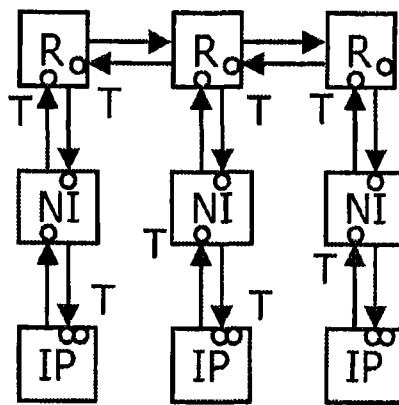


图 3d

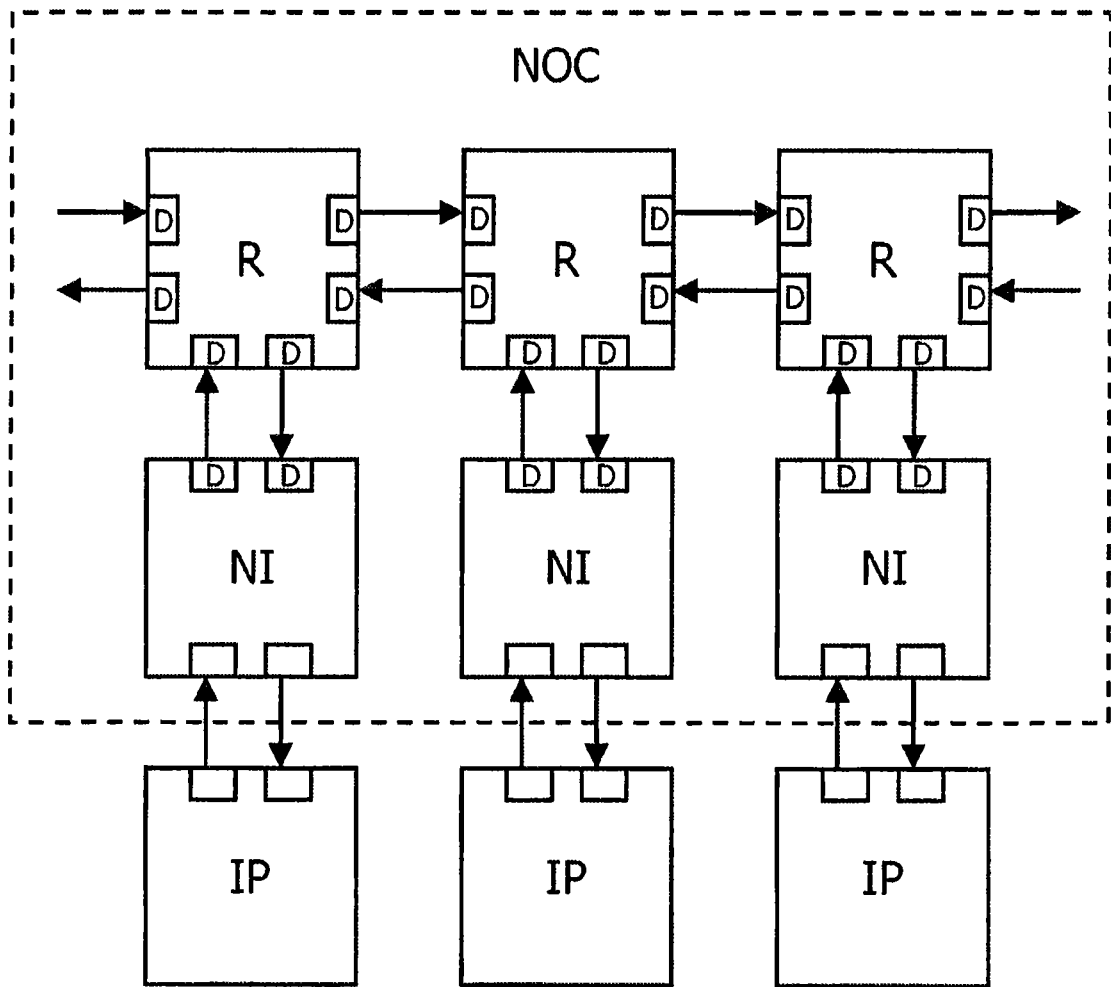


图 4

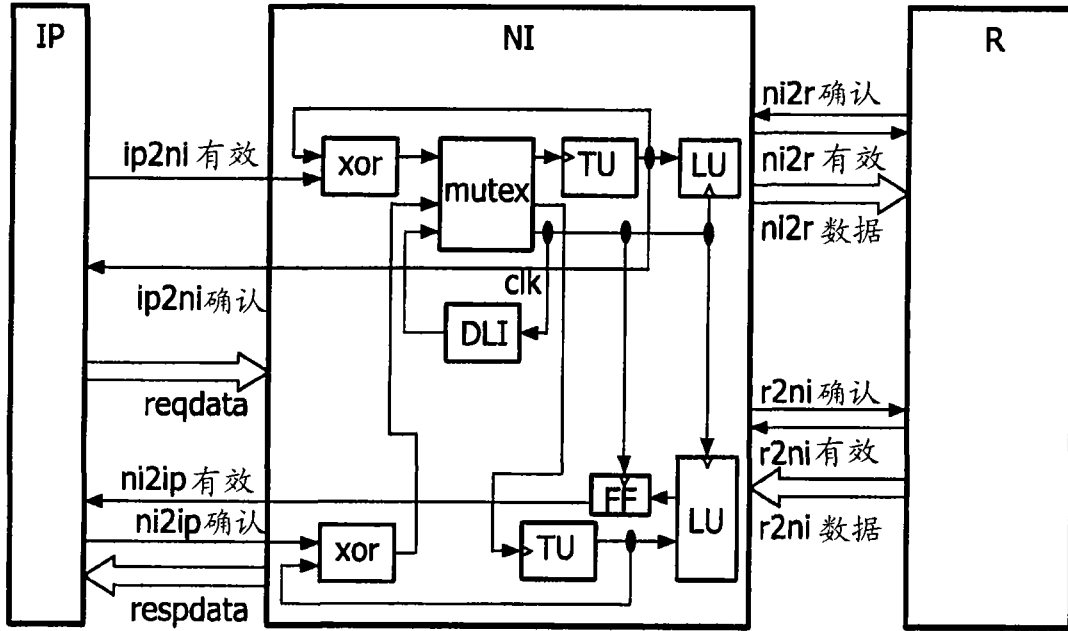


图 5

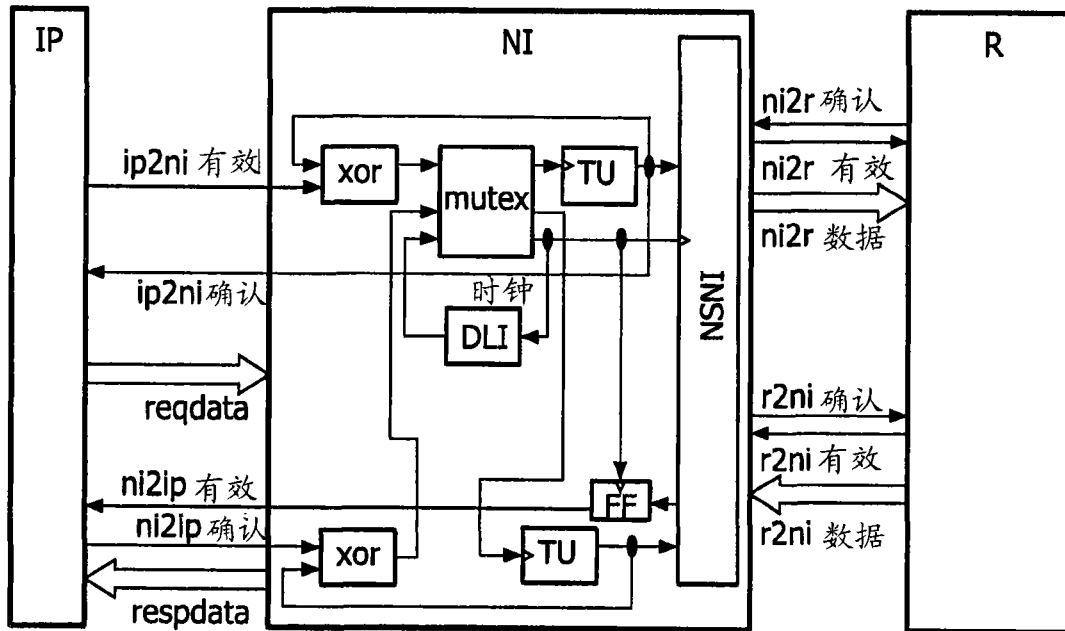


图 6

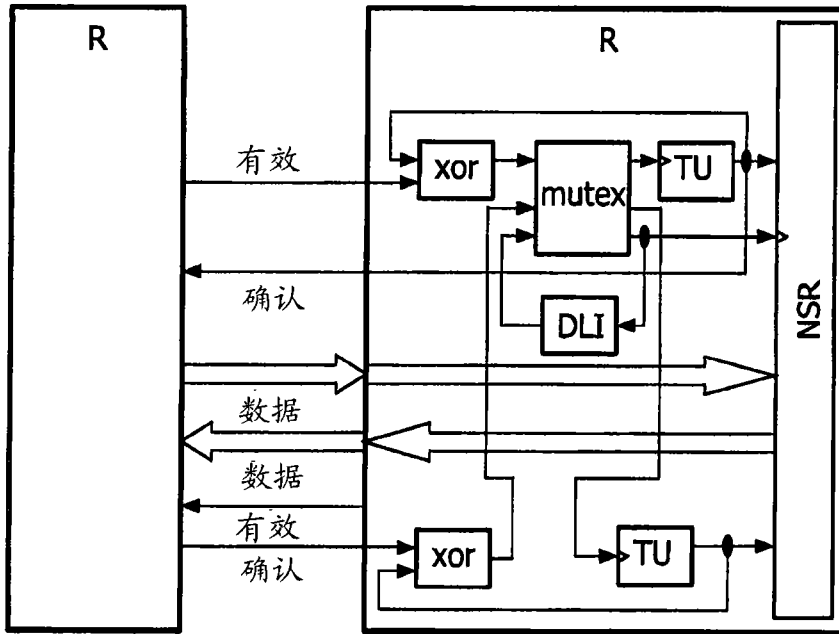


图 7

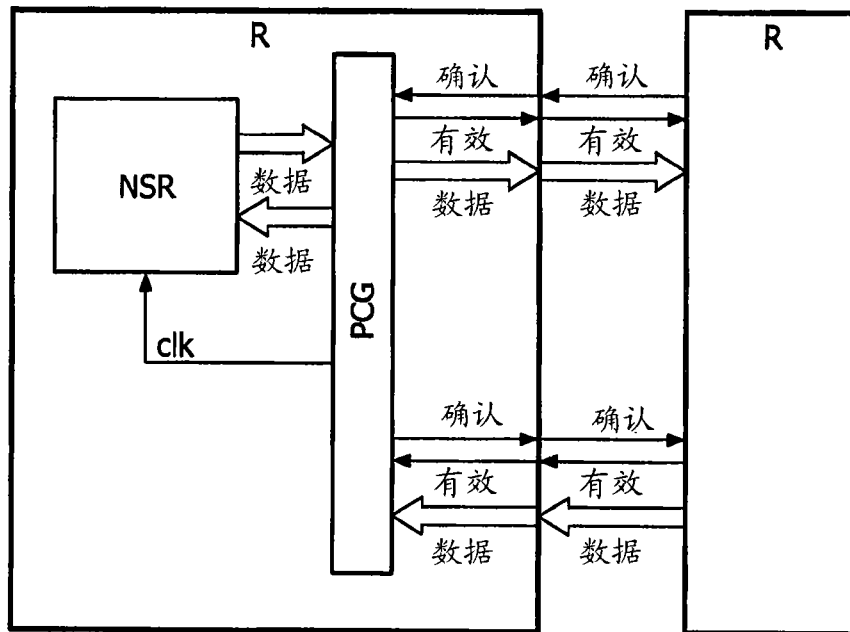


图 8

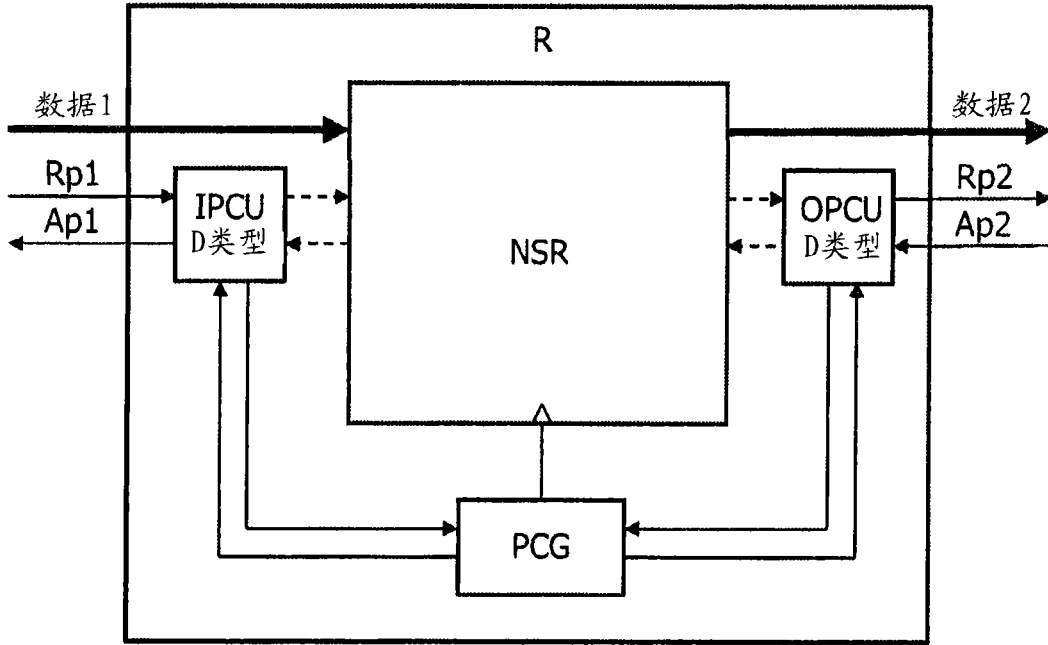


图 9

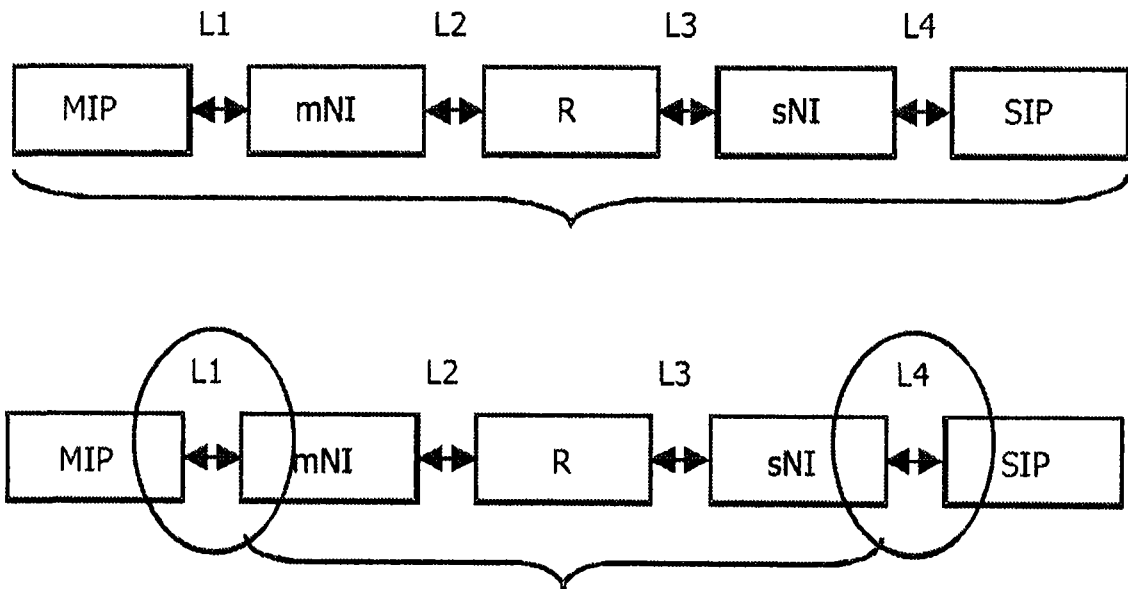


图 10

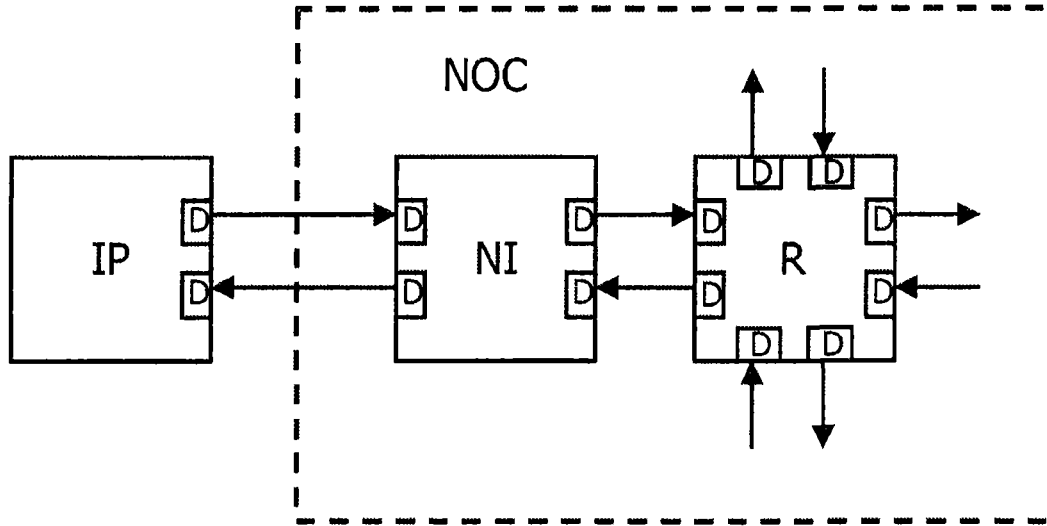


图 11

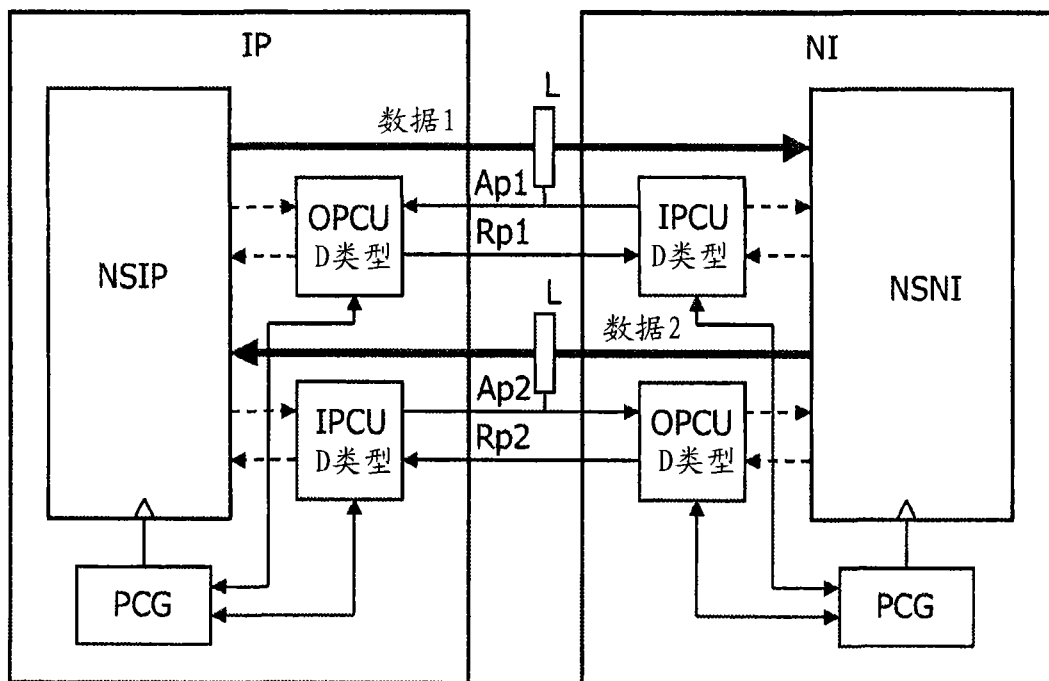


图 12

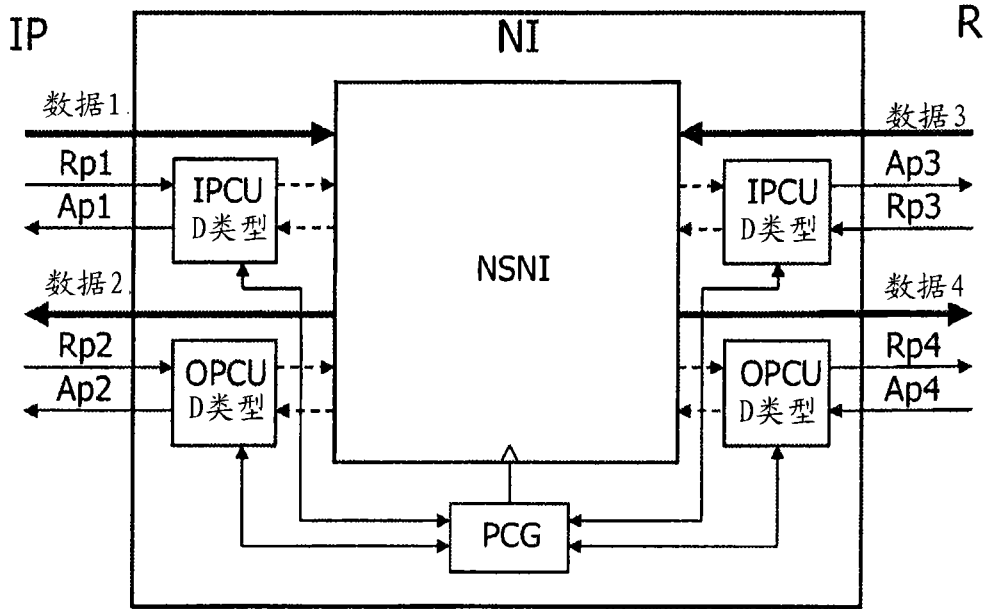


图 13

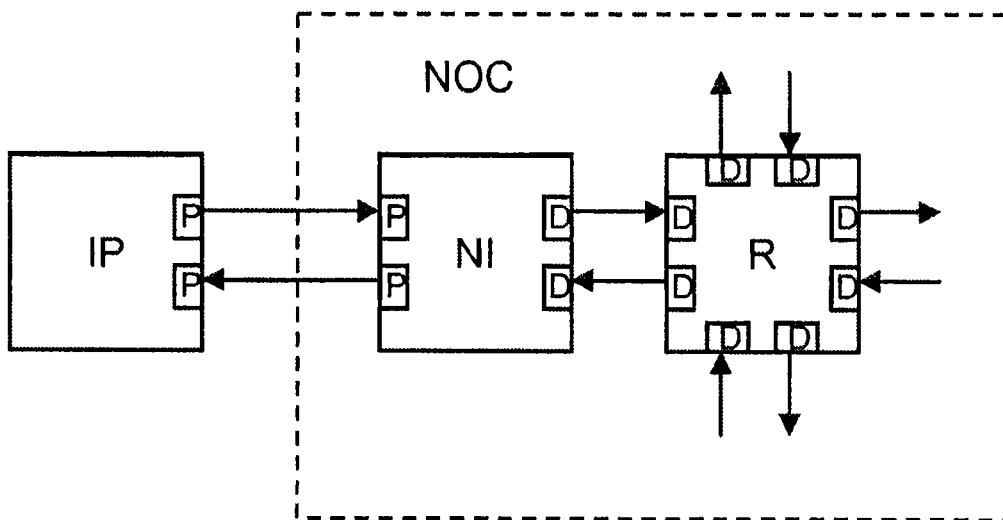


图 14

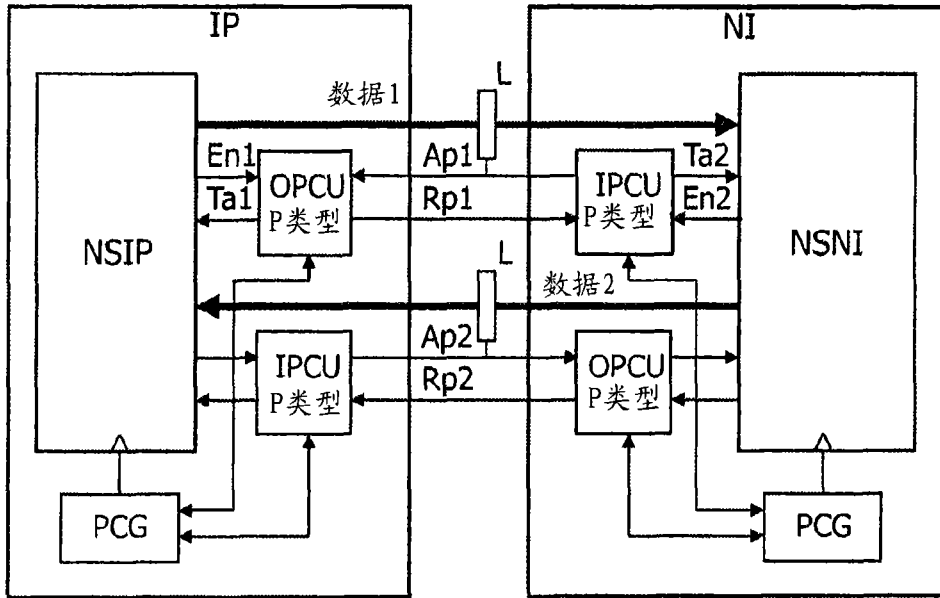


图 15

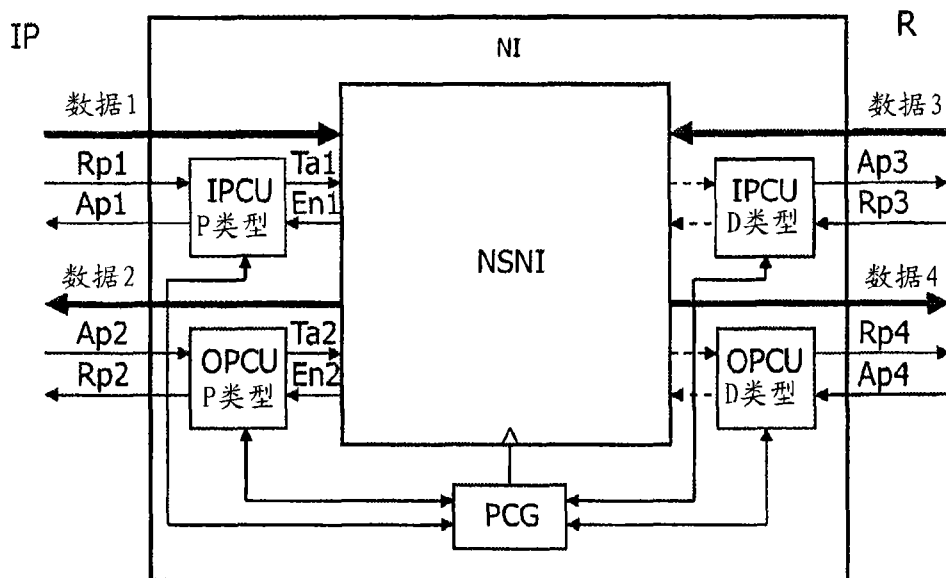


图 16

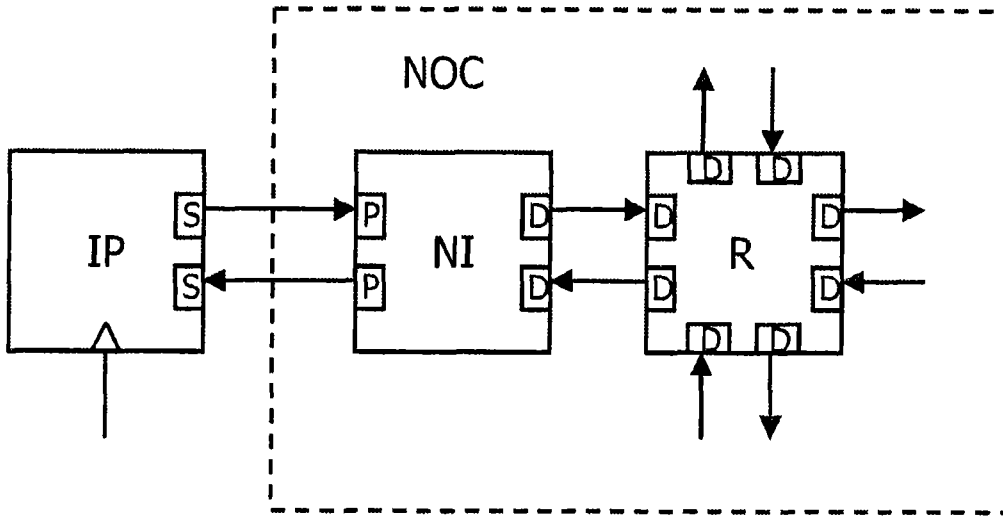


图 17

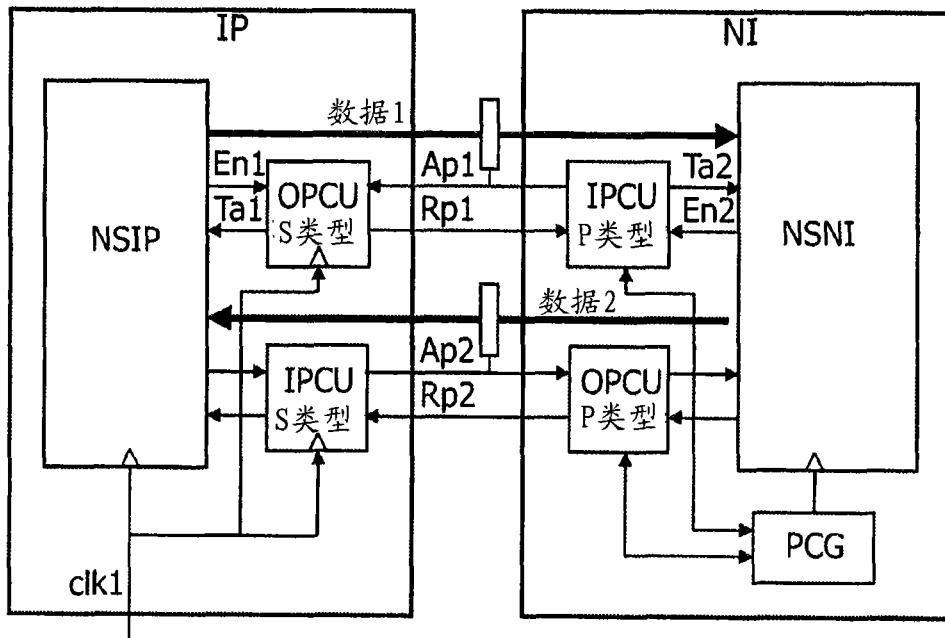


图 18

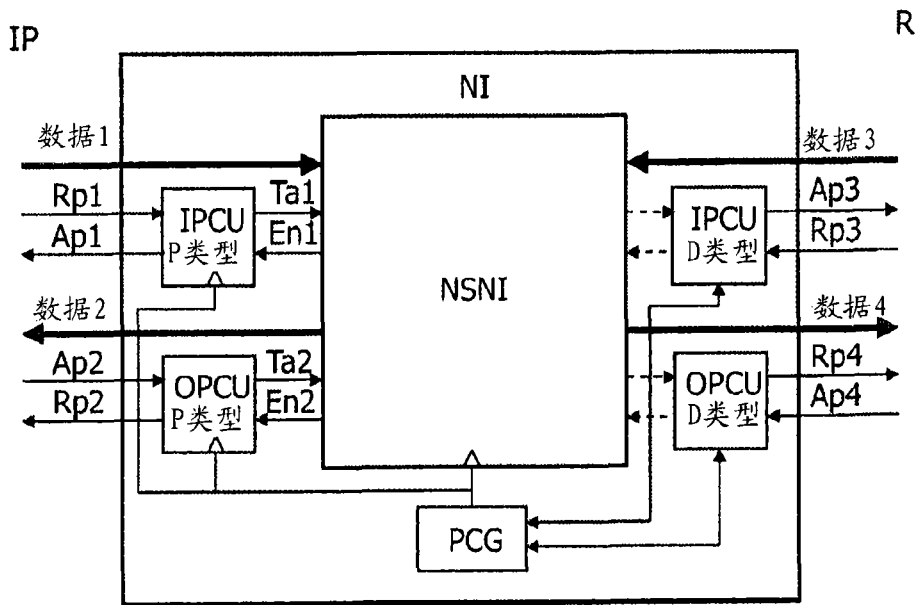


图 19

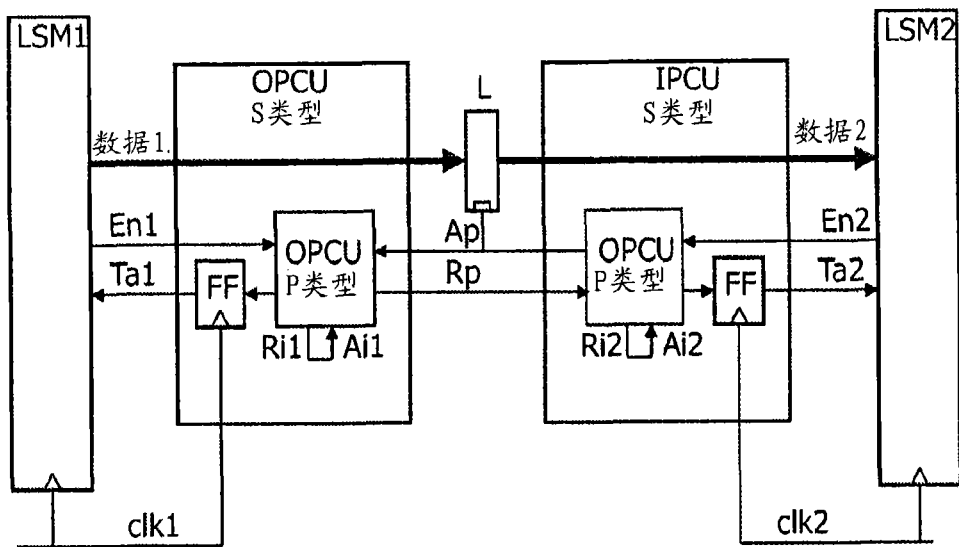


图 20

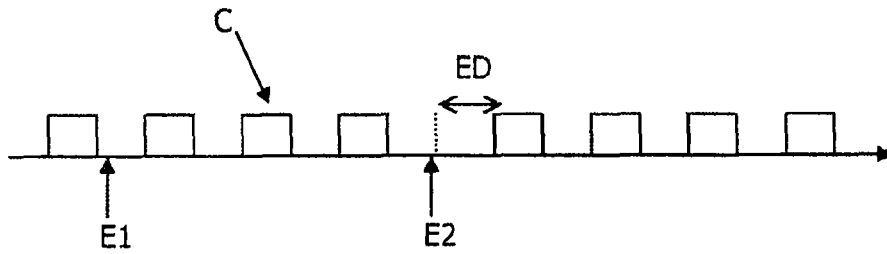


图 21

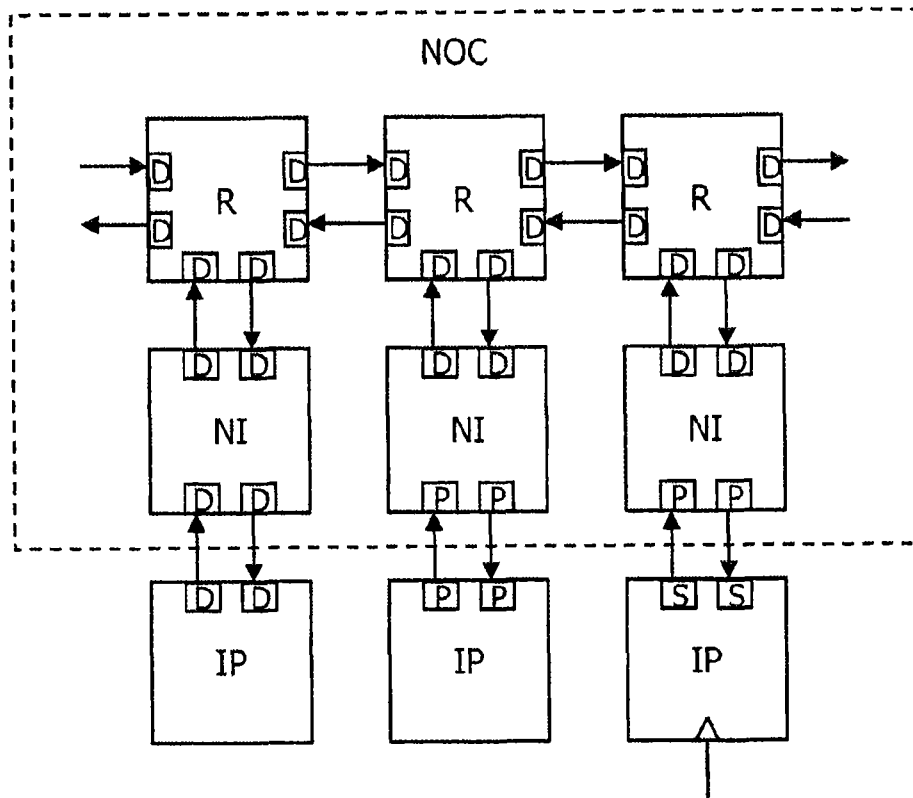


图 22

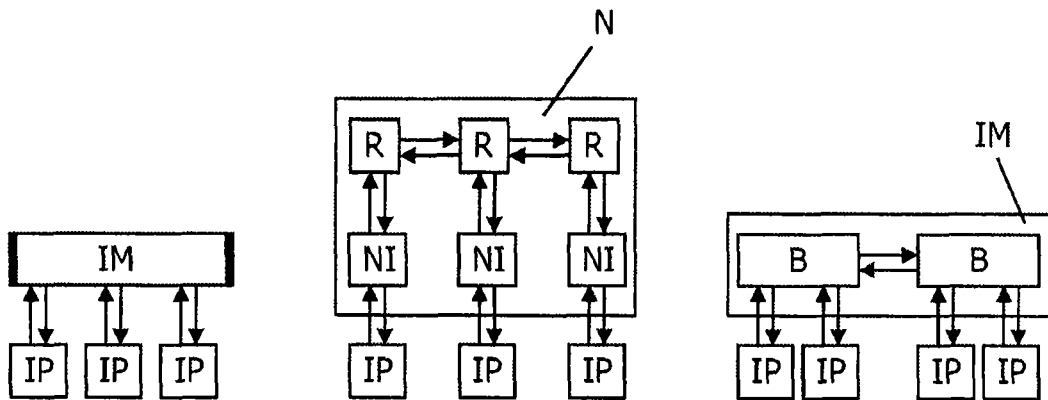


图 23

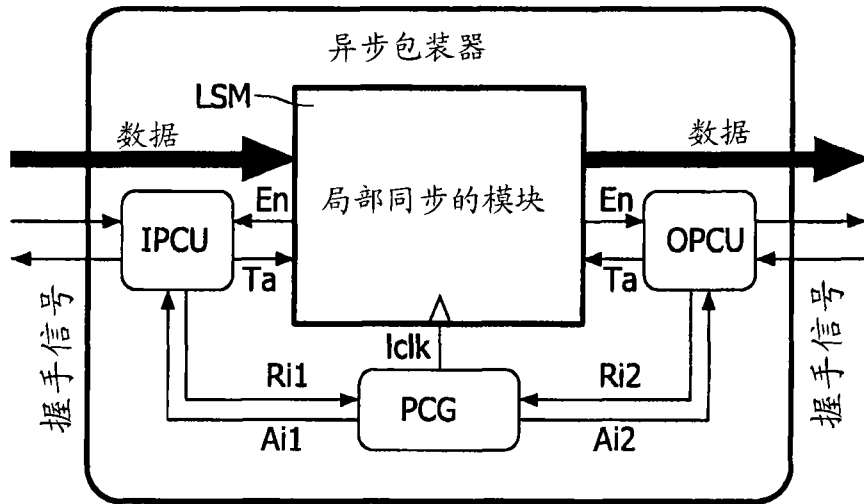


图 24

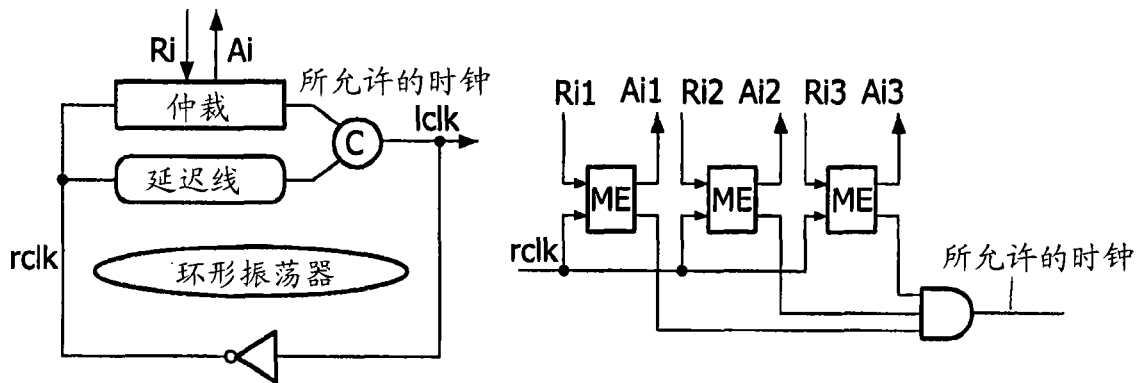


图 25

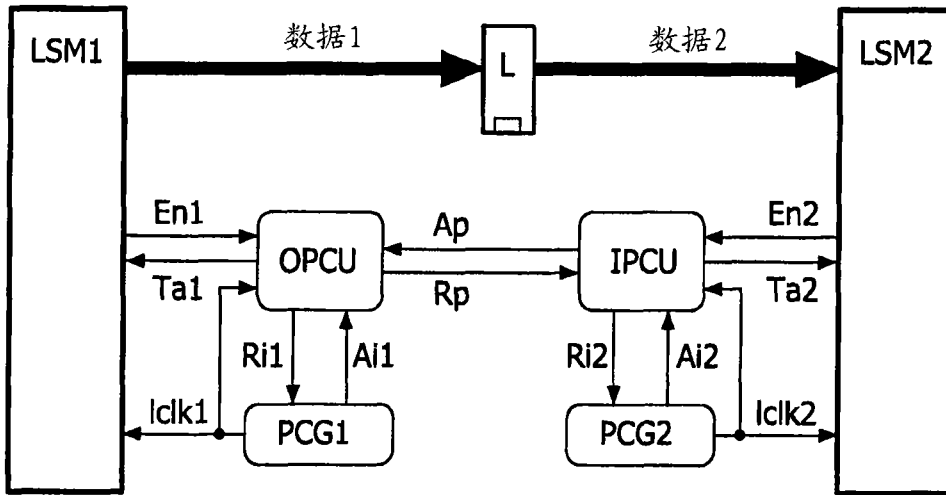


图 26

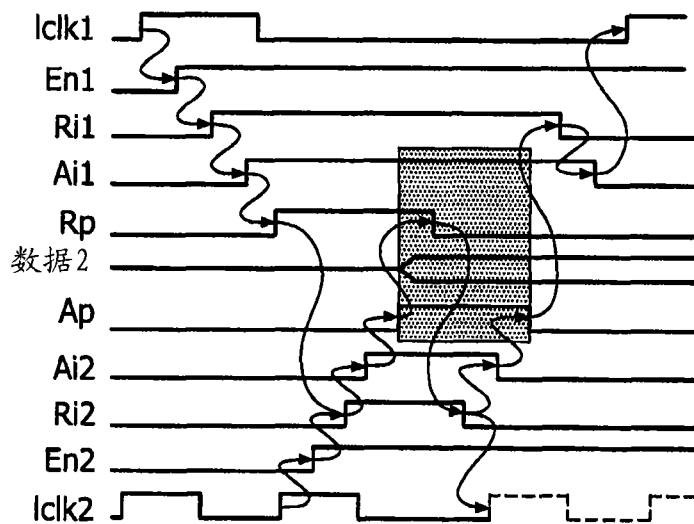


图 27

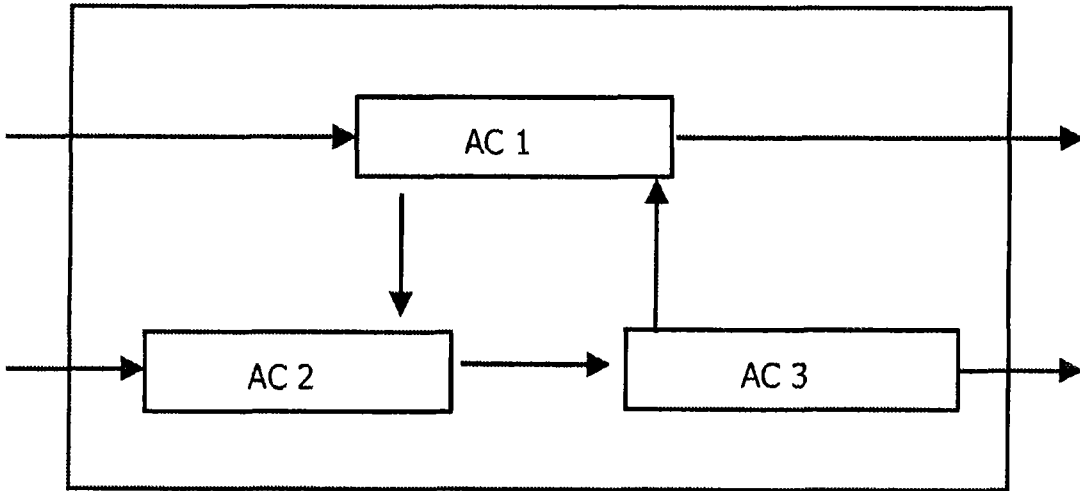


图 28

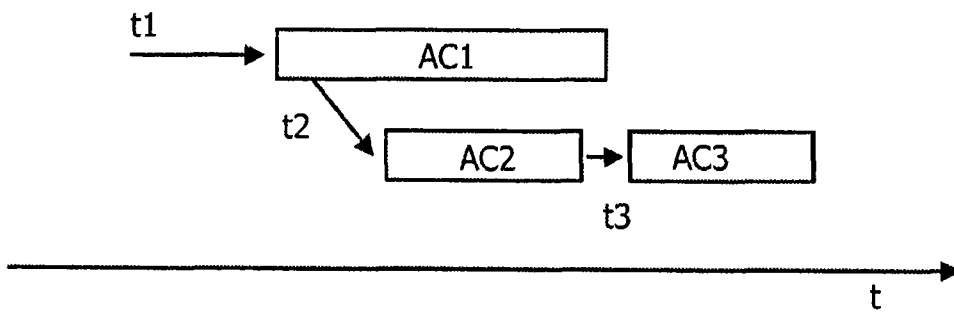


图 29