



## (12) 发明专利

(10) 授权公告号 CN 101978659 B

(45) 授权公告日 2013. 11. 06

(21) 申请号 200980110393. 9

(22) 申请日 2009. 03. 31

(30) 优先权数据

12/061302 2008. 04. 02 US

(85) PCT申请进入国家阶段日

2010. 09. 21

(86) PCT申请的申请数据

PCT/US2009/038874 2009. 03. 31

(87) PCT申请的公布数据

W02009/146025 EN 2009. 12. 03

(73) 专利权人 英特尔公司

地址 美国加利福尼亚州

(72) 发明人 A·库马 P·P·孔杜

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

代理人 王岳 王洪斌

(51) Int. Cl.

H04L 12/70 (2013. 01)

H04L 29/02 (2006. 01)

(56) 对比文件

US 20040165536 A1, 2004. 08. 26, 说明书第 0031-0036 段, 0053-0054 段, 图 2, 7.

US 6674720 B1, 2004. 01. 06, 说明书第 6 列第 19-22 行, 41-45 行.

US 20020150056 A1, 2002. 10. 17, 全文.

CN 1705297 A, 2005. 12. 07, 全文.

审查员 刘爽

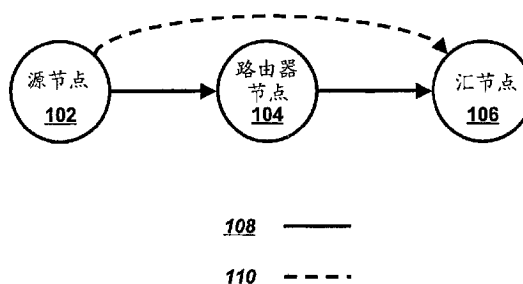
权利要求书1页 说明书7页 附图14页

(54) 发明名称

分组交换的片上互连网络中的快速虚拟通道

(57) 摘要

公开了用于在芯片上部件网络中使用快速虚拟通道的方法、路由器节点和指令集。输入链路 302 可以经由快速虚拟通道 110 从分组交换网络中的源节点 102 接收快速流控制单元。输出链路 306 可以把快速流控制单元发送到汇节点 106。交换机分配器 322 可以把快速流控制单元直接转发到输出链路 306。



100

1. 一种在分组交换的片上互连网络中使用快速虚拟通道的方法,包括:  
经由快速虚拟通道从分组交换网络的源节点接收快速流控制单元;  
把快速流控制单元直接转发到输出链路以发送到汇节点;  
跟踪从快速虚拟通道接收的快速流控制单元;  
如果达到资源缺乏阈值,则进入资源缺乏补救模式;以及  
把正常流控制单元传输到相邻节点。
2. 权利要求 1 的方法,其中快速虚拟通道具有静态长度。
3. 权利要求 1 的方法,其中快速虚拟通道具有动态长度。
4. 权利要求 1 的方法,其中快速虚拟通道是在最优快速虚拟通道不能用的情况下供替换的快速虚拟通道。
5. 权利要求 1 的方法,其中快速虚拟通道具有静态分派的缓冲器。
6. 权利要求 1 的方法,其中快速虚拟通道具有动态分派的缓冲器。
7. 一种路由器节点,包括:  
输入链路,用于经由快速虚拟通道从分组交换网络中的源节点接收快速流控制单元;  
输出链路,用于把快速流控制单元发送到汇节点;  
交换机分配器,用于把快速流控制单元直接转发到输出链路;以及  
资源缺乏计数器,用于确定是否达到资源缺乏阈值,以便正常的流控制单元能够到达相邻节点。
8. 权利要求 7 的路由器节点,其中快速虚拟通道具有静态长度。
9. 权利要求 7 的路由器节点,其中快速虚拟通道具有动态长度。
10. 权利要求 7 的路由器节点,其中快速虚拟通道是在最优快速虚拟通道不能用的情况下供替换的快速虚拟通道。
11. 权利要求 7 的路由器节点,其中快速虚拟通道具有静态分派的缓冲器。
12. 权利要求 7 的路由器节点,其中快速虚拟通道具有动态分派的缓冲器。
13. 权利要求 7 的路由器节点,其中分组交换网络是单芯片多处理器。
14. 一种在分组交换的片上互连网络中使用快速虚拟通道的装置,包括:  
经由快速虚拟通道从分组交换网络的源节点接收快速流控制单元的部件;  
把快速流控制单元直接转发到输出链路以发送到汇节点的部件;  
跟踪从快速虚拟通道接收的快速流控制单元的部件;  
如果达到资源缺乏阈值,则进入资源缺乏补救模式的部件;以及  
把正常的流控制单元传输到相邻节点的部件。
15. 权利要求 14 的装置,其中快速虚拟通道具有动态长度。
16. 权利要求 14 的装置,其中快速虚拟通道是在最优快速虚拟通道不能用的情况下供替换的快速虚拟通道。
17. 权利要求 14 的装置,其中快速虚拟通道具有动态分派的缓冲器。

## 分组交换的片上互连网络中的快速虚拟通道

### 技术领域

[0001] 本发明一般涉及单芯片多处理器 (chip multi-processor) 上的部件之间的通信的领域。本发明还具体地涉及在单芯片多处理器上的部件之间创建快速 (express) 虚拟通信通道。

### 背景技术

[0002] 单芯片多处理器 (CMP) 或专用片上系统 (ASOC) 可能具有需要彼此通信的若干部件。这些部件可以是诸如处理器、高速缓存、数据寄存器或其他芯片部件之类的零件。芯片部件可以使用总线系统或专用布线系统进行通信。这些系统中的每一个可能要求大量的布线,其可能用尽芯片上的宝贵空间并且在信号之间产生干扰。

[0003] 这些部件可以通过使用分组交换系统来以空间更高效的方式进行通信。分组交换系统可以通过在部件之间传送数据分组来起作用,从而减小任意两个部件之间的布线量。

### 附图说明

[0004] 要理解这些附图仅描绘了本发明的典型实施例,而因此不被视为对其范围的限制,将通过使用附图另外详细且具体地描述和解释本发明,在附图中:

[0005] 图 1 以框图示出单芯片多处理器部件数据分组传输的一个实施例。

[0006] 图 2 作为二维网状网络示出芯片上的部件网络的一个实施例。

[0007] 图 3 以框图示出实施快速虚拟通道的路由器节点微架构的一个实施例。

[0008] 图 4 作为二维网状网络示出芯片上的动态快速虚拟通道部件网络的一个实施例。

[0009] 图 5 作为框图示出芯片上的路由灵活 (route-flexible) 的动态快速虚拟通道部件网络的一个实施例。

[0010] 图 6 以流程图示出路由器节点处理来自虚拟通道的数据分组的方法的一个实施例。

[0011] 图 7 以网状网络示出快速虚拟通道网络中的资源缺乏 (starvation) 事件的一个实施例。

[0012] 图 8 以时间线示出正常虚拟通道基于信用 (credit) 的缓冲器管理系统的一个实施例。

[0013] 图 9 以时间线示出快速虚拟通道基于信用的缓冲器管理系统的一个实施例。

[0014] 图 10 以框图示出静态缓冲器管理系统的缓冲器计数的一个实施例。

[0015] 图 11 以框图示出动态共享的缓冲器管理系统的缓冲器计数的一个实施例。

[0016] 图 12 以框图示出可获得的进出给定节点的路径的组合的一个实施例。

[0017] 图 13 以框图示出虚拟通道的静态分区的一个实施例。

[0018] 图 14 以框图示出可以充当路由器节点的计算机系统的一个实施例。

### 具体实施方式

[0019] 本发明的附加特征和优点将在以下的说明书中进行阐述,并且部分地将从说明书中显而易见,或者可以通过实践本发明而获悉。本发明的特征和优点可以借助于在所附权利要求书中具体指出的工具和组合来实现和获得。本发明的这些及其他特征将从以下的说明书和所附的权利要求书中变得更彻底地清楚,或者可以通过实践如本文所阐述的发明而获悉。

[0020] 下面详细讨论本发明的各个实施例。虽然讨论了具体的实施方式,但是应当理解这样做仅仅是为了说明目的。相关领域的技术人员将意识到在不偏离本发明的精神和范围的情况下可以使用其他部件和配置。

[0021] 本发明包括诸如方法、设备和计算机指令集之类的各种各样的实施例、以及与本发明的基本概念有关的其他实施例。公开了用于在芯片上的部件网络中使用快速虚拟通道的方法、路由器节点和指令集。输入链路可以经由快速虚拟通道从分组交换网络中的源节点接收快速流控制单元(express flow control unit)。输出链路可以把快速流控制单元发送到汇节点。交换机分配器可以把快速流控制单元直接转发到输出链路。

[0022] 单芯片多处理器(CMP)或专用芯片上系统(ASOC)可能具有许多需要彼此通信的部件。这些部件可以使用分组交换来传送数据而不使用过量的芯片空间。图1以框图示出单芯片多处理器部件数据传输100的一个实施例。充当源节点102的部件可以把数据集组织成数据分组。数据分组可以被分成一个或多个流控制单元(FLIT)。源节点102可以把每个FLIT传输到最近可用的部件。充当路由器节点104的接收部件可以把数据传输到下一个最近可用的部件,直到到达充当汇节点106的部件为止。FLIT从源节点102到汇节点106所遵循的路径可以被组织成虚拟通道。在正常虚拟通道(NVC)108中,源节点102可以把FLIT传输到路由器节点104,该路由器节点104可以缓冲FLIT、组织该FLIT以及其他经历该管线的FLIT、然后把它们发送到输出链路以将其发送到下一个路由器节点104或汇节点106。

[0023] 快速虚拟通道(EVC)110可以通过接近专用链路的等待时间和能量而用来缩短封装化的片上网络和理想互连结构之间的性能和能量差距。EVC 110可以允许FLIT沿节点对之间的预定义路径绕过中间路由器节点104。与其他FLIT相比,可以给予在EVC 110上行进的FLIT优待,从而使得它们直接使用输出链路而不用进行缓冲,且不必通过每个路由器节点104处的路由器管线。EVC 110可以以类似于电路交换的方式促进流在网络中的创建而不涉及任何电路建立阶段,从而显著地减少分组传输等待时间。此外,通过允许FLIT实际上绕过中间路由器节点104,EVC 110可以减小缓冲量、路由器交换活动和每跳资源竞争,从而增加能量和面积效率。

[0024] 图2作为二维网状网络示出静态EVC分组交换网络200的一个实施例。虽然引用分组交换芯片多处理器网络,但是EVC 110可以用于任何分组交换网络。请注意,对本发明而言,网状网络的大小和二维性质是说明性的而决不是限定性的。每个节点202可以充当源节点102、路由器节点104或汇节点106。网络200中的每个节点202可以通过NVC108而连接到相邻节点202。而且,节点202可以通过EVC 110而连接到非相邻节点,从而允许FLIT沿预定义路径穿过多个中间节点或路由器节点104,而不用在任何路由器节点104处停止或缓冲。例如,从节点B行进到节点G的FLIT可以采取从节点B到节点D的规则NVC路径108并且然后采取从节点D到节点G的EVC 110或链路。EVC 110可以消除通过诸如

节点 E 和 F 之类的路由器节点 104 的等待时间,并且通过消除路由器节点 104 上的所有缓冲操作来节约功率。在静态 EVC 网络中,具体的源节点 102 和汇节点 106 可以通过 EVC 110 连接。虽然在本示例中使用四个节点的静态 EVC 110,但是根据选择也可以使用更大或更小长度的静态 EVC 110。

[0025] 路由器可以实施在虚拟通道流控制之上的快速链路。路由器可以指定虚拟通道,该虚拟通道可以仅由在快速路径上行进的数据分组预订。图 3 以框图示出实施 EVC 110 的路由器节点微架构 300 的一个实施例。路由器节点 104 可以经由输入链路 302 上的虚拟通道接收 FLIT,并且把 FLIT 引导至流控制模块 304。用于流控制的虚拟通道可以被布置为 NVC108 和 EVC 110。流控制模块 304 可以使用虚拟通道来确定 FLIT 优先级以选择哪个 FLIT 转发到由纵横式交换机 308 所选择的输出链路 306。

[0026] 路由计算逻辑模块 310 可以在每个路由器节点 104 处计算路由计算或数据分组在源节点 102 和汇节点 106 之间行进的路径。路由计算逻辑模块 310 可以负责验证数据分组是否可以被放置在 EVC110 上。另外,路由计算逻辑模块 310 可以执行所有数据分组的先行 (look-ahead) 路由,以便一旦数据分组到达节点或 EVC 路径 110 端点,在路由器管线的关键路径中就不需要路由计算。路由计算不必以任何方式影响所用的基本路由算法。EVC 分配器 312 可以给打算使用快速通路来在节点对之间行进的数据分组分配 EVC 110,而 NVC 分配器 314 可以给打算行进到邻近节点的另一个数据分组分配 NVC 108。

[0027] 流控制模块 304 可以经由 EVC 输入缓冲器 (EVC BUF) 316 从 EVC110 或者经由 NVC 输入缓冲器 (NVC BUF) 318 从 NVC 108 接收 FLIT。当 FLIT 通过其快速路径 110 上的任何路由器节点 104 时,EVC 锁存器 320 可以实行 (stage) FLIT。通过 EVC 锁存器 320 行进的 FLIT 可以优先于 (pre-empt) 相同输出端口 306 上的任何其他数据分组流。EVC 锁存器 320 可以中断两个节点之间正在进行的数据分组传输而免于被源节点 102 发送,诸如较低优先级 EVC 流。

[0028] 交换机分配器 322 可能喜欢 EVC 流 110 胜过 NVC 流 108。交换机分配器 322 可以中断源节点 102 处具有较少跳的 FLIT 的 EVC 流 110,以便具有更多跳的 FLIT 的 EVC 流 110 可以使用该路由器节点 104。交换机分配器 322 可以中断较近源节点 102 处的 FLIT 的 EVC 流 110,以便来自更远源节点 102 的 FLIT 的 EVC 流 110 可以使用该路由器节点 104。

[0029] 网络 200 可以被静态地划分成 EVC 110 和 NVC108,或者可以具有更多动态创建的虚拟通道。图 4 作为二维网状网络示出动态 EVC 分组交换网络 400 的一个实施例。在动态 EVC 分组交换网络 400 中,节点 202 之间的任何跳集可以被指定为动态 EVC 402。EVC 路径 402 的长度可以在由部件网络约束确定的最大 EVC 长度内针对给定数据路径的路径来根据需要定制大小。例如,在静态 EVC 网络 200 中,源节点 102 可以经由单跳 NVC 路径 108 从节点 B 遍历到节点 C 并从节点 C 遍历到节点 D,并且经由静态分配的 EVC 路径 110 从节点 D 遍历到节点 G。在允许最大三跳 EVC 路径 402 的动态网络中,源节点 102 可以把 FLIT 从节点 B 直接发送到节点 E,然后可以使用两跳长度的子集 EVC 路径 402 而去往节点 G,诸如此类。

[0030] 图 5 作为框图示出芯片 500 上的路由灵活的动态 EVC 部件网络的一个实施例。有时,数据路径可能由于资源约束而不能使用最优 EVC 路径 502,诸如最长的 EVC 路径 502。在这种情况下,源节点 102 可以经由具有可用资源的可选 EVC 路径 504 (诸如下一个较短 EVC 路径 504) 发送 FLIT。例如,对于源节点 A 102 把 FLIT 发送到汇节点 G 106 的最高效的数

据路径可以是把 FLIT 沿第一最优 EVC 路径 502 从源节点 A 102 直接发送到节点 D, 然后沿第二最优 EVC 路径 502 从路由器节点 D 104 发送到汇节点 G 106。然而, 最高效的数据路径可能没有可用的资源。源节点 A 102 可以沿可选路径 504 把 FLIT 发送到汇节点。源节点 A 102 可以沿可选 EVC 路径 504 把 FLIT 发送到路由器节点 C 104, 然后从路由器节点 C 104 发送到路由器节点 E 104, 并然后从路由器节点 E 104 发送到汇节点 G 106。EVC 分配器 212 可以通过考虑数据分组的路由信息来确定最佳路径。

[0031] 图 6 以流程图示出路由器节点处理来自虚拟通道的 FLIT 的方法 600 的一个实施例。路由器节点 104 可以经由虚拟通道 (VC) 接收 FLIT (块 602)。如果 FLIT 是正常 FLIT (NF) (块 604), 则路由器节点 104 可以把 NF 存储在输入缓冲器中 (块 606) 并且然后处理 NF (块 608)。NF 可以是路由器节点 104 是源节点 102 的 FLIT、从 NVC 108 上的邻近节点到达路由器节点 104 的 FLIT、或者路由器节点 104 是 EVC 110 的终点但不是汇节点 106 的 FLIT。输入端口可以是 NVC 输入缓冲器 318 (如果到达 NVC 108 的话) 或者是 EVC 输入缓冲器 316 (如果到达 EVC 110 的话)。路由器节点 104 可以把处理的 NF 转发到输出链路 306 (块 610)。如果 FLIT 是快速 FLIT (EF) 110 (块 604), 则路由器节点 104 可以把 EF 转发到输出链路 306 (块 612)。快速 FLIT 可以是由在 EVC 110 中部的路由器节点 104 接收的 FLIT。

[0032] 在为具体的消息流预留带宽的任何网络中, 当在预先建立的电路上行进的消息阻挡其他消息时可能出现资源缺乏。图 7 以网状网络示出 EVC 网络中的资源缺乏事件 700 的一个实施例。给予 EF 702 的较高优先级可能导致资源缺乏情形 704。更具体地, 如果沿 EVC 702 路径的路由器节点 D 104 总是具有即将到来的 EF 要服务, 则在路由器节点 D 104 处局部缓冲的 NF 可能从未得到使用物理通道的机会。来自更上游的快速数据分组也可以使源自下游路由器的不同 EVC 110 上的快速数据分组连续资源缺乏。

[0033] 为对此进行补救, 每个路由器节点 104 可以保存该路由器节点 104 已服务了具体 EVC 路径 110 的连续循环的数量的计数。在给快速 FLIT 服务了阈值数量的连续循环之后, 路由器节点 104 可以把诸如“开始资源缺乏补救 (Begin Starvation Remediation)”消息或“资源缺乏开 (Starvation On)”令牌之类的资源缺乏消息沿信用通道向上游发送到 EVC 源节点 A 102。在接收到这个令牌后, 源节点 A 102 可以停止在对应的链路上发送 EF, 以允许服务局部资源缺乏的 FLIT。在这样做设定数量的循环后, 路由器节点 D 104 可以发送“结束资源缺乏补救 (End Starvation Remediation)”消息或“资源缺乏关 (Starvation Off)”令牌到 EVC 源节点 A 102, 以信号通知可以恢复 EVC 通信。

[0034] 如图 6 中所示, 对于 EF, 如果去往相同输出端口 306 的 NF 被缓冲 (块 614), 则路由器节点 104 可以递增资源缺乏计数 (SCT) (块 616)。路由器节点 104 可以检查 SCT 以查看其是否达到了资源缺乏阈值 (STHR) (块 618)。如果 SCT 达到了 STHR (块 618), 则路由器节点 104 可以把“开始资源缺乏补救”消息 (BSRM) 向上游发送到所有 EVC 源节点 102 (块 620)。在允许所有适当的上游节点接收 BSRM 的一段时间之后, 路由器节点 104 可以进入资源缺乏补救模式, 在资源缺乏补救模式期间, 路由器节点不接收 EF。虽然路由器节点 104 处于资源缺乏补救模式, 但是 EVC 源节点 102 可以暂停 EF 的传输、重新路由 EF、或者采取其他补偿动作。路由器节点 104 可以把资源缺乏补救周期计数器 (SRP) 设定为零 (块 622)。

[0035] 对于 NF, 如果路由器节点 104 不处于资源缺乏补救模式 (块 624), 则路由器节点

104 可以把 SCT 复位为零 (块 626)。如果路由器节点 104 处于资源缺乏补救模式 (块 624), 则一旦 NF 已经被成功地路由到输出端口 306, 路由器节点 104 就可以递增 SRP (块 628)。一旦资源缺乏补救周期结束 (块 630), 路由器节点 104 就可以把“结束资源缺乏补救”消息 (ESRM) 发送到所有上游 EVC 源节点 102, 以信号通知路由器节点 104 再次接收 EF (块 632)。

[0036] 芯片上部件网络可以使用缓冲流控制技术来管理缓冲器, 并且在相邻的路由器节点 104 之间传送其可用性。下游节点可以通过背压把可用的自由缓冲器数量传送到上游节点。基于信用的缓冲器管理可以使用上游路由器来保存下游可用的自由缓冲器数量的计数。每当发出 FLIT 就可以使计数递减, 从而消耗下游缓冲器。当 FLIT 离开下游节点并且释放其相关缓冲器时, 可以向上游发回信用并且可以递增对应的自由缓冲器计数。

[0037] 部件网络可以使用简单的缓冲器管理方案来给每个虚拟通道静态地分配等于该虚拟通道的信用往返延迟的缓冲。图 8 以时间线示出 NVC108 基于信用的缓冲器管理系统 800 的一个实施例。在时间  $t_1$ , 当节点 B 发送 FLIT 到下一个节点 (箭头 802) 时, 节点 B 可以在  $t_1$  和  $t_2$  之间发送信用到节点 A (箭头 804)。节点 A 可以在  $t_2$  和  $t_3$  之间处理信用 (箭头 806)。节点 A 可以发送信用到先前节点 (箭头 804) 同时在  $t_3$  和  $t_4$  之间发送 FLIT 到节点 B (箭头 802)。节点 B 可以从  $t_4$  到  $t_5$  处理非快速管线中的 FLIT (箭头 808)。因而, 从  $t_1$  到  $t_5$  可以得到信用往返延迟 ( $T_{CRN}$ )。

[0038] 当使用 EVC 110 时, 缓冲器可用性的传送可以跨越若干跳来完成, 以确保在下游 EVC 汇节点 106 处的自由缓冲器。缓冲器可用性的多跳传送可能导致较长的缓冲周转时间, 对通过量有不良影响。图 9 以时间线示出 EVC 110 基于信用的缓冲器管理系统 900 的一个实施例。在时间  $t_1$ , 当节点 D 发送 FLIT 到下一个节点 (箭头 902) 时, 节点 D 可以在  $t_1$  和  $t_2$  之间发送信用到节点 A (箭头 904)。节点 A 可以在  $t_2$  和  $t_3$  之间处理信用 (箭头 906)。节点 A 可以发送信用到先前节点 (箭头 904) 同时在  $t_3$  和  $t_4$  之间发送 FLIT 到节点 B (箭头 902)。节点 B 可以从  $t_4$  到  $t_5$  把 FLIT 转发到作为部分快速管线的输出链路 (箭头 908)。节点 B 可以在  $t_5$  和  $t_6$  之间发送 FLIT 到节点 C (箭头 902)。节点 C 可以从  $t_6$  到  $t_7$  把 FLIT 转发到作为部分快速管线的输出链路 (箭头 908)。节点 C 可以在  $t_7$  和  $t_8$  之间发送 FLIT 到节点 D (箭头 902)。节点 C 可以从  $t_8$  到  $t_9$  处理非快速管线中的 FLIT (箭头 910)。因而, 从  $t_1$  到  $t_9$  可以得到信用往返延迟 ( $T_{CRE}$ )。

[0039] 缓冲器管理系统可以静态地组织缓冲器。图 10 以框图示出静态缓冲器管理系统 1000 的缓冲器计数的一个实施例。缓冲器的静态数量可以被分派给每个虚拟通道, 被保留用于到达每个虚拟通道的 FLIT。NVC 缓冲器集 1002 可以被保留用于每个 NVC 108 并且 EVC 缓冲器集 1004 可以被保留用于每个 EVC 110。当设定该集中缓冲器的数量时, 每个 EVC 缓冲器集 1004 的大小可以把  $T_{CRE}$  考虑在内。

[0040] 静态缓冲器管理可能在对立业务的情况下在分配缓冲器方面效率低。例如, 如果大多数网络业务只有近邻, 则 EVC 110 也许从未被使用, 并且静态分派给 EVC 110 的缓冲器空间也许变得闲置。代替把缓冲器静态地分区成 VC (诸如 NVC 108 和 EVC 110) 的是, 缓冲器可以按需被分派给 VC。当缓冲器变得可用时, 向上游节点信号通知 EVC 的存在的高水位线 (high-water mark) 方案可以优先考虑短跳 EVC 402 或 NVC 108 而不是长跳。例如, 如果两跳 EVC 402 要求至少八个缓冲器可用, 则当在下游路由器处的缓冲器计数等于八时, 下游路由器可以给上游路由器信号通知离开两个节点 (two nodes away) 的 EVC 402 可用。

可以允许缓冲器自由浮动跨越 VC, 以致 VC 可能只有一个 FLIT。

[0041] 图 11 以框图示出动态共享的缓冲器管理系统 1100 的缓冲器计数的一个实施例。缓冲器计数 1102 可以表示可用的或自由的缓冲器的数量。如果缓冲器计数 1102 指示可用缓冲器的数量等于 NVC 阈值数量 1104、或者为操作 NVC 108 所需的缓冲器数量, 则节点可以使用 NVC 108。如果缓冲器计数 1102 指示可用缓冲器的数量等于 EVC 阈值数量 1106、或者为操作 EVC 402 所需的缓冲器数量, 则节点可以使用 EVC 402。EVC 阈值 1106 可以与 EVC 路径 402 的长度有关。

[0042] 为了实现灵活的缓冲, 下游路由器可能必须向上游路由器发送适当的信号。图 12 以框图示出可获得的进出给定节点的路径 1200 的组合的一个实施例。虽然示出八节点网络, 但是可以使用其他大小的网络。

[0043] VC 可以被静态地分区。图 13 以框图示出虚拟通道的静态分区 1300 的一个实施例。VC 的总数 1302 或分组状态跟踪器 (packet state tracker) 可以在 NVC 108 和不同长度的 EVC 402 两者之间进行划分。这种划分可以在设计时完成。对于具有 EVC 长度高达  $I_{\text{MAX}}$  的能力的部件网络而言, 总计  $(k \times I_{\text{MAX}})$  的虚拟通道集可以被供应。部件网络可以给 NVC 108 分派  $k_1$  个虚拟通道 1302, 给长度为二的 EVC 402 分派  $k_2$  个虚拟通道, 给长度为三的 EVC 402 分派  $k_3$  个虚拟通道, 诸如此类。

[0044] 图 14 示出用于充当分组交换网络中的路由器节点 104 的计算系统 1400 的可能配置。路由器节点 104 可以包括通过总线 1440 连接的控制器 / 处理器 1410、存储器 1420、网络接口 1430。路由器节点 104 可以实施任何操作系统, 例如诸如 Microsoft **Windows®**、UNIX 或 LINUX。客户端和服务端软件可以以任何编程语言来编写, 例如诸如 C、C++、Java 或 Visual Basic。服务端软件可以运行在应用程序框架上, 例如诸如 **Java®** 服务器或 **NET®** 框架。

[0045] 控制器 / 处理器 1410 可以是为本领域技术人员所已知的任何编程处理器。然而, 决策支持方法也可以在通用或专用计算机、编程微处理器或微控制器、外围集成电路元件、专用集成电路或其他集成电路、硬件 / 电子逻辑电路, 诸如分立元件电路、可编程逻辑器件 (诸如可编程逻辑阵列、现场可编程门阵列等等) 上实施。一般而言, 能够实施如本文所描述的 EVC 路由方法的任一装置或多个装置可以用来实施本发明的决策支持系统功能。

[0046] 存储器 1420 可以包括易失性和非易失性数据存储器, 包括一个或多个电、磁或光存储器, 诸如随机存取存储器 (RAM)、高速缓存、硬驱或其他存储器装置。存储器可以具有高速缓存以加速对特定数据的访问。存储器 1420 也可以连接到致密盘只读驱动器 (CD-ROM)、数字多功能光盘只读存储器 (DVD-ROM)、DVD 读写输入、磁带驱动或者允许直接加载数据到系统中的其他可移动存储器装置。

[0047] 网络连接接口 1430 可以连接到通信装置、调制解调器、网络接口卡、收发器或者任何其他能够从网络发射和接收信号的装置。网络连接接口 1430 可以用来把路由器节点 104 连接到网络。路由器节点 104 的部件可以例如经由电总线 1440 连接或者被以无线方式链接。

[0048] 客户端软件和数据库可以被控制器 / 处理器 1410 从存储器 1420 进行访问, 并且例如可以包括数据库应用、字处理应用以及体现本发明的决策支持功能的部件。路由器节点 104 可以实施任何操作系统, 例如诸如 Microsoft **Windows®**、UNIX 或 LINUX。客户端

和服务软件可以以任何编程语言来编写,例如诸如 C、C++、Java 或 Visual Basic。尽管不要求,但是在诸如程序模块之类的计算机可执行指令的一般背景下,至少部分地描述本发明,所述计算机可执行指令由诸如通用计算机之类的电子装置执行。一般而言,程序模块包括执行特定任务或实施特定抽象数据类型的例程程序、对象、部件、数据结构等等。此外,本领域的技术人员将明白,本发明的其他实施例可以在具有许多类型的计算机系统配置的网络计算环境中加以实施,包括个人计算机、手持式装置、多处理器系统、基于微处理器的或可编程消费者电子器件、网络 PC、小型计算机、大型计算机、等等。

[0049] 尽管不要求,但是在诸如程序模块之类的计算机可执行指令的一般背景下至少部分地描述本发明,所述计算机可执行指令由诸如通用计算机之类的电子装置执行。一般而言,程序模块包括执行特定任务或实施特定抽象数据类型的例程程序、对象、部件、数据结构等等。此外,本领域的技术人员将明白,本发明的其他实施例可以在具有许多类型的计算机系统配置的网络计算环境中加以实施,包括个人计算机、手持式装置、多处理器系统、基于微处理器的或可编程消费者电子器件、网络 PC、小型计算机、大型计算机、等等。

[0050] 实施例也可以在分布式计算环境中加以实施,其中任务由通过通信网络(由硬布线链路、无线链路或由其组合)链接的本地和远程处理装置执行。

[0051] 本发明范围内的实施例也可以包括用于承载计算机可执行指令或数据结构或者在其上存储计算机可执行指令或数据结构的计算机可读介质。这样的计算机可读介质可以是任何可用介质,所述介质可以由通用或专用计算机访问。作为示例而非限制,这样的计算机可读介质可以包括 RAM、ROM、EEPROM、CD-ROM 或其他光盘存储器、磁盘存储器或其他磁存储器装置、或者任何其他可以用来以计算机可执行指令或数据结构的形式承载或存储期望的程序代码构件的介质。当信息通过网络或另一通信连接(硬布线的、无线的或其组合)而被转移或提供到计算机时,计算机把该连接恰当地视为计算机可读介质。因而,任何这样的连接被恰当地称作计算机可读介质。上面的组合也应当被包括在计算机可读介质的范围内。

[0052] 计算机可执行指令包括例如使通用计算机、专用计算机或专用处理装置执行特定功能或功能组的指令和数据。计算机可执行指令也包括由独立或网络环境中的计算机执行的程序模块。一般而言,程序模块包括执行特定任务或实施特定抽象数据类型的例程、程序、对象、组件和数据结构等等。计算机可执行指令、相关数据结构以及程序模块代表用于执行本文公开的方法的步骤的程序代码构件的示例。这样的可执行指令或相关数据结构的特定序列代表用于实施在此类步骤中描述的功能的对应动作的示例。

[0053] 尽管上面的描述可以包含特定细节,但是它们不应当被解释为以任何方式限制本发明。本发明的所描述实施例的其他配置是本发明的范围的一部分。例如,本发明的原理可以应用于每个单独用户,其中每个用户可以单独部署这样的系统。即使大量可能应用中的任何一个不需要本文描述的功能,这也使得每个用户能够利用本发明的益处。电子装置的多个实例中的每一个可以以各种可能的方式处理内容。不必需要所有终端用户都使用一个系统。因此,所附的权利要求书及其法律等效物应当唯一地限定本发明,而不是由所给出的任何特定实例来限定本发明。

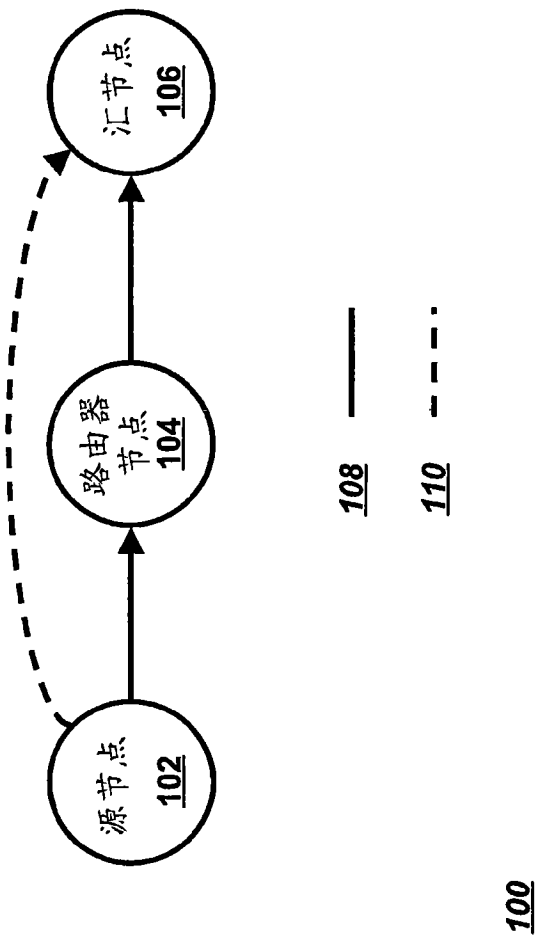


图 1

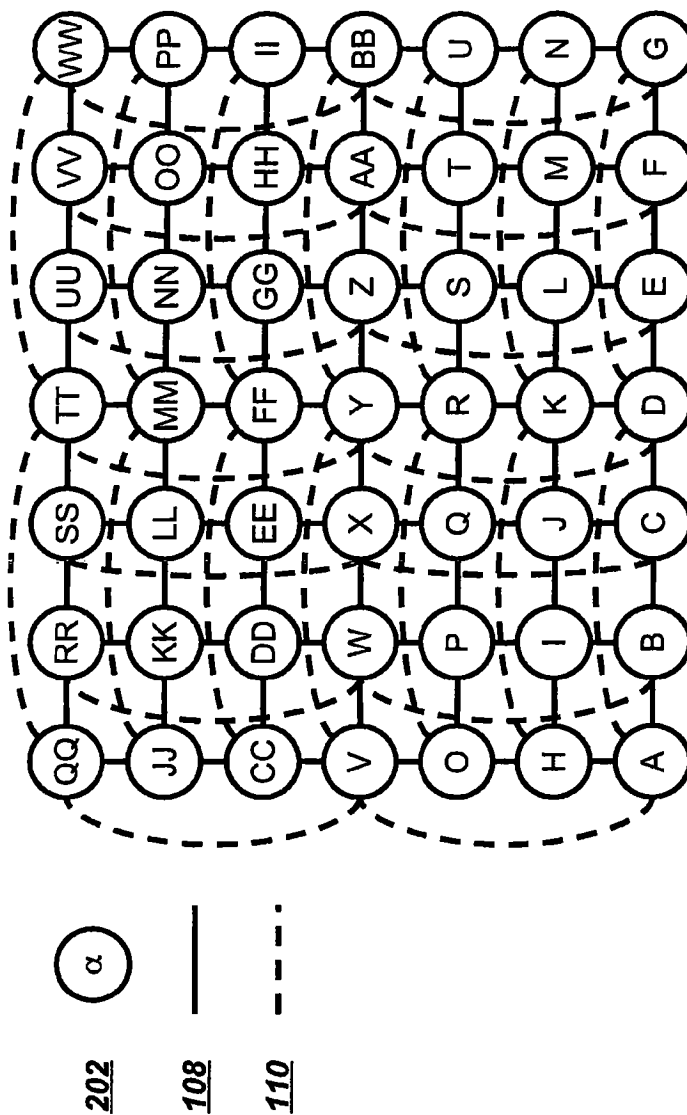


图 2

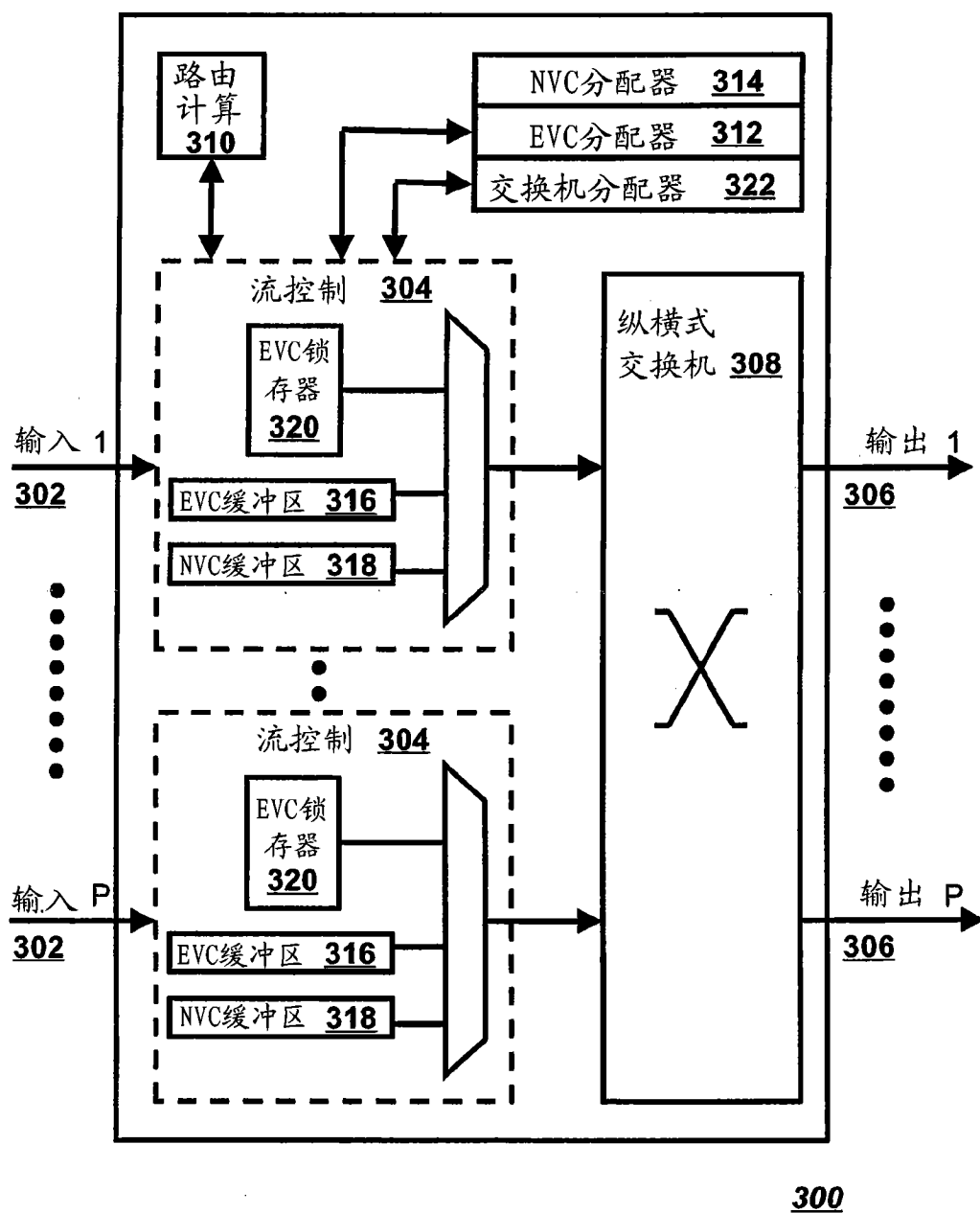
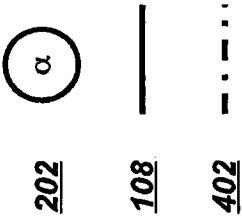
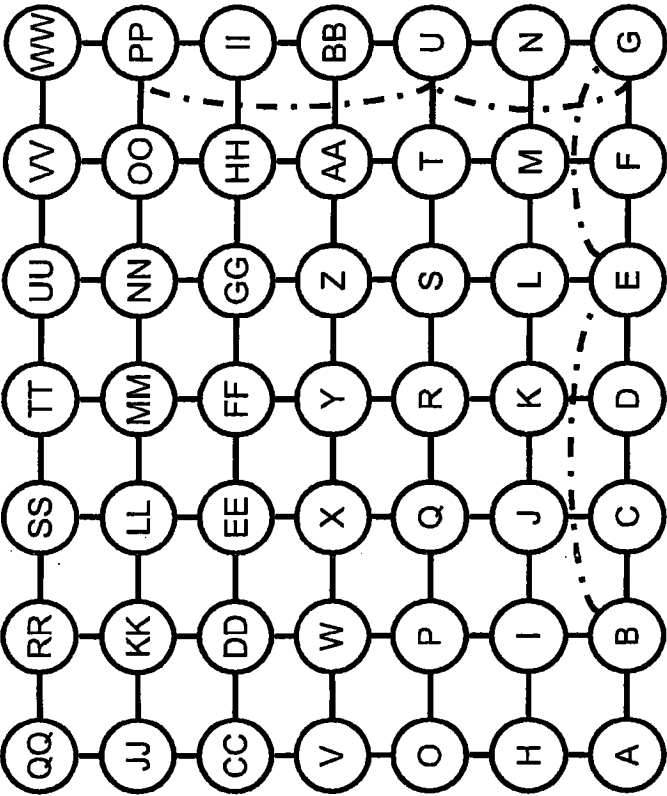


图 3



400

图 4

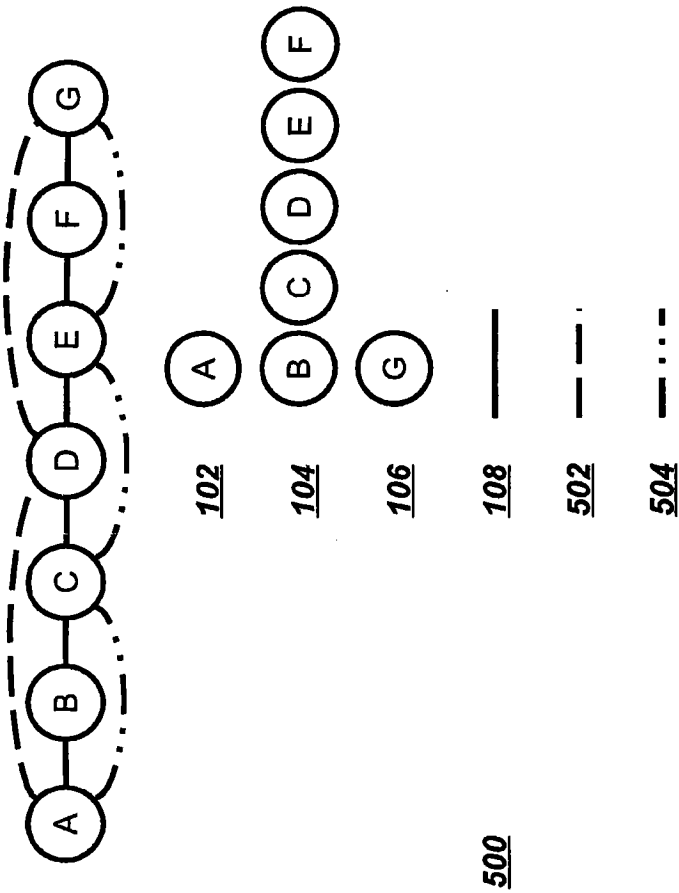


图 5

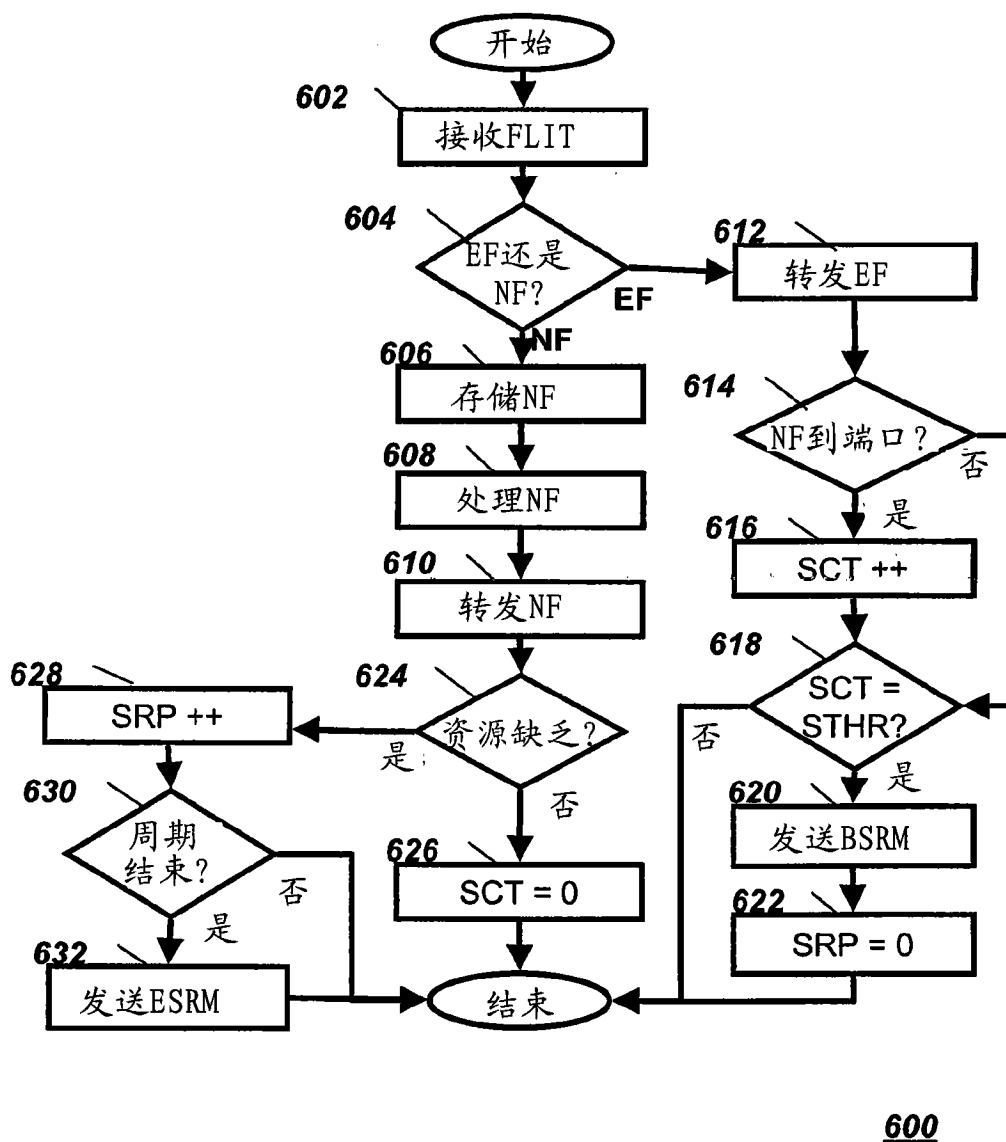


图 6

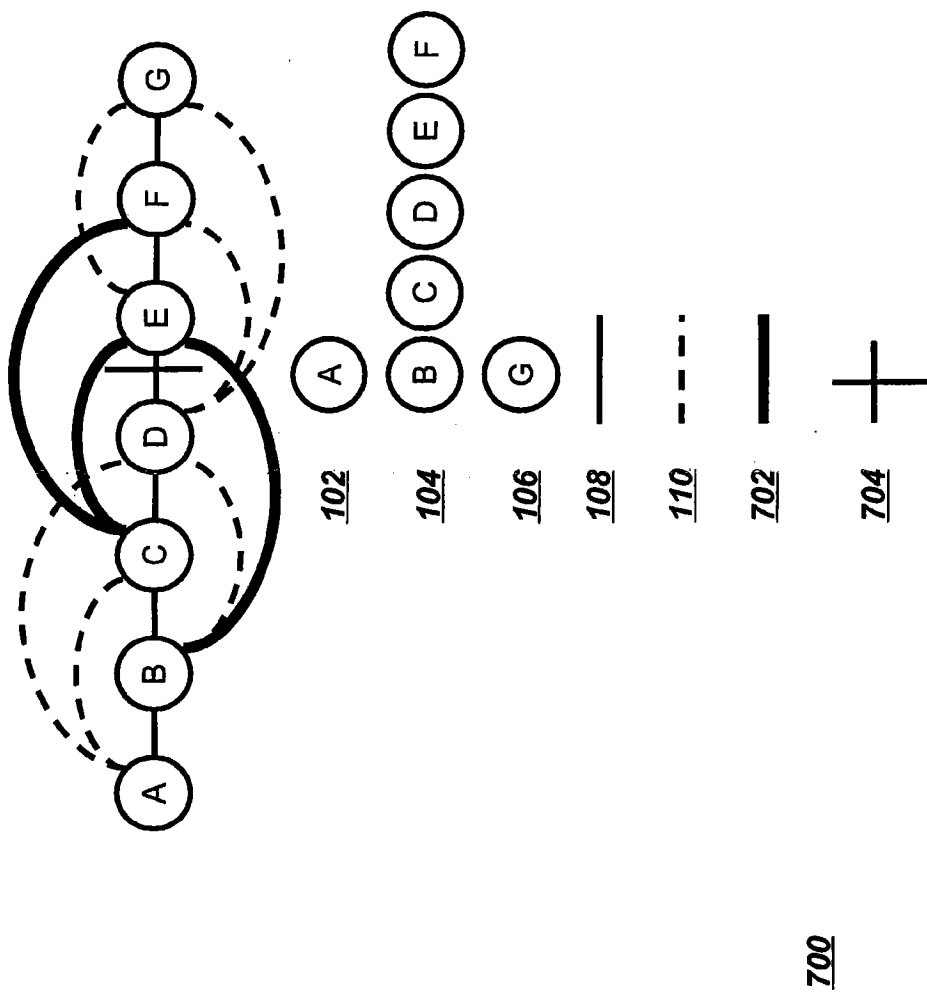


图 7

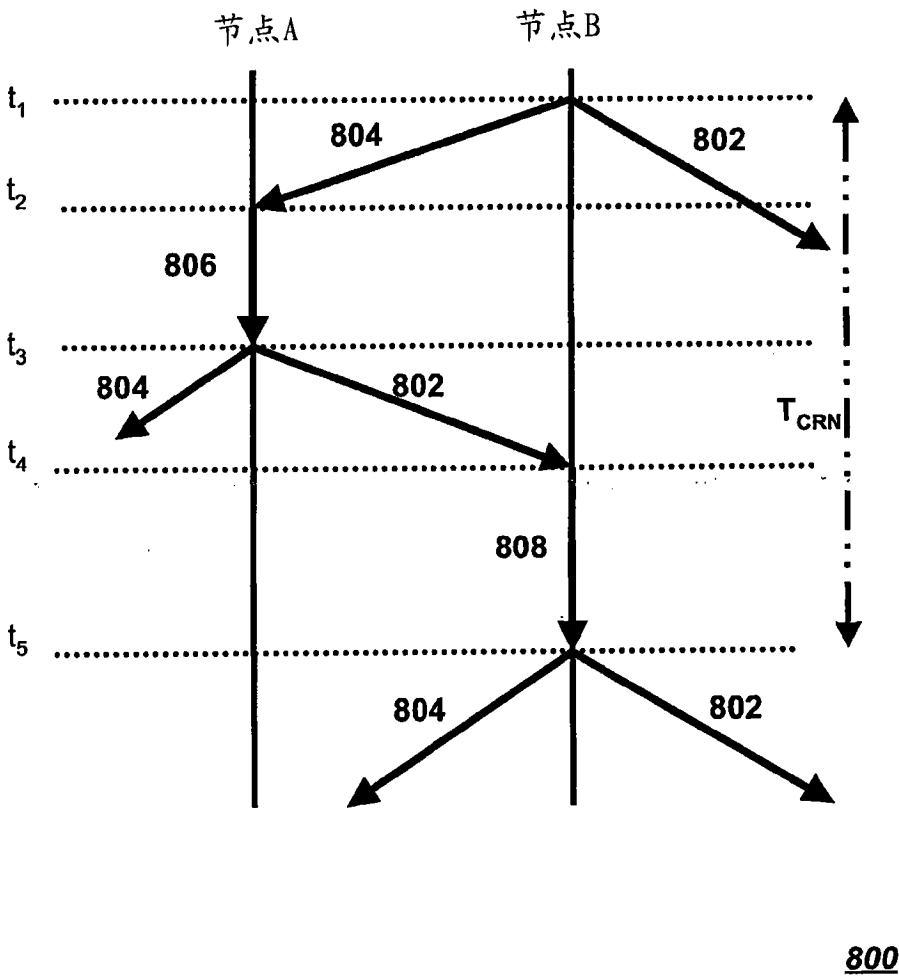


图 8

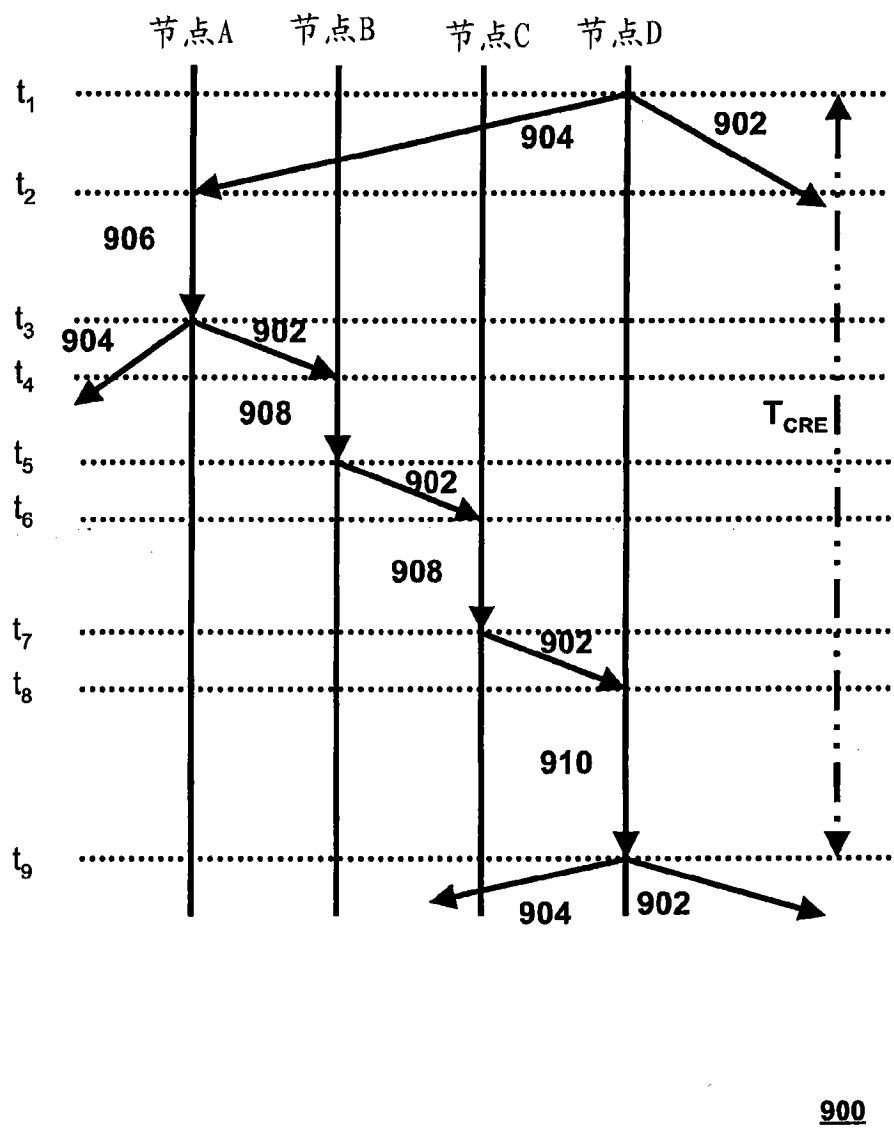


图 9

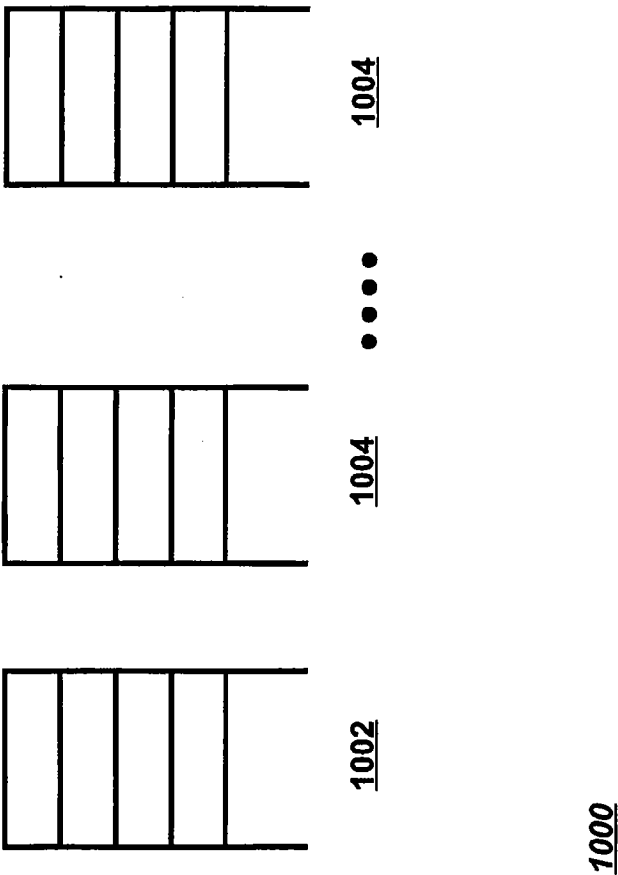


图 10

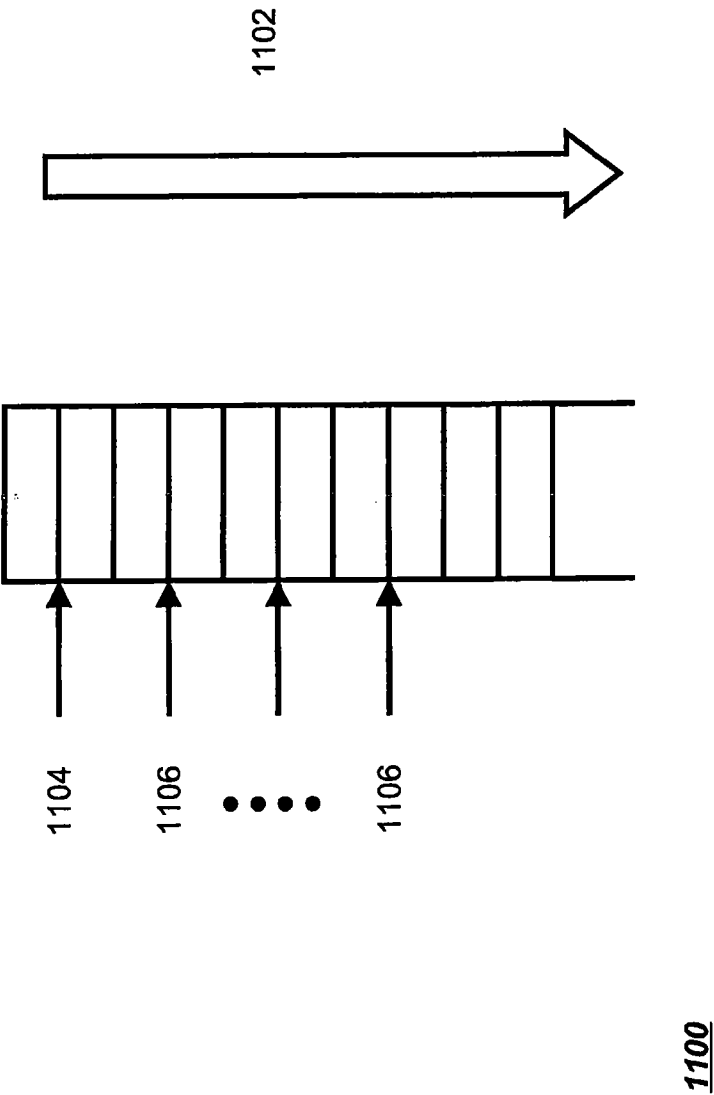
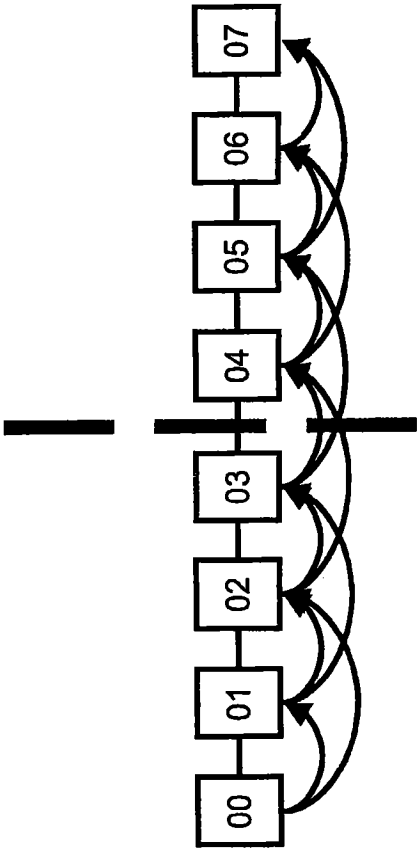
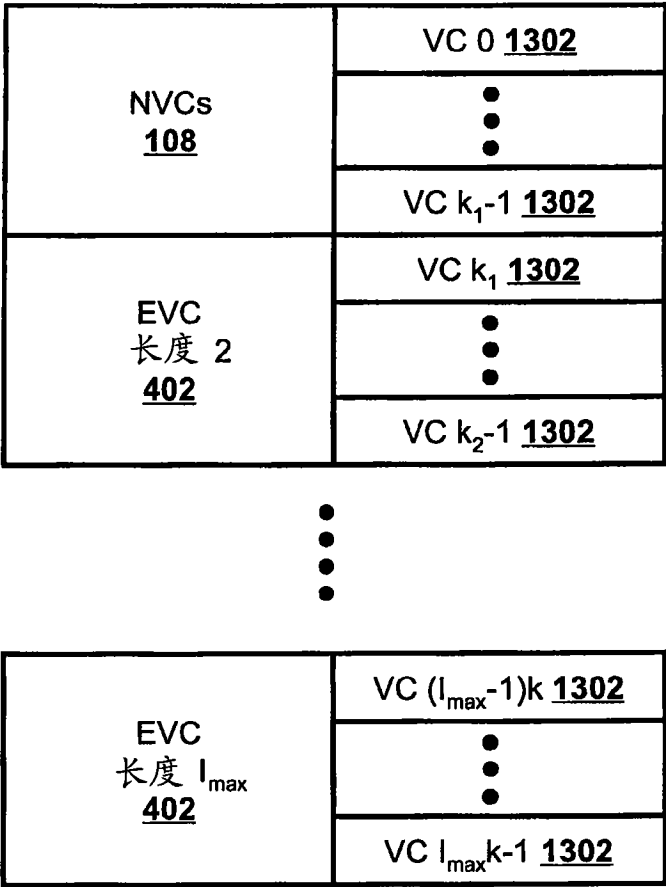


图 11



1200

图 12



1300

图 13

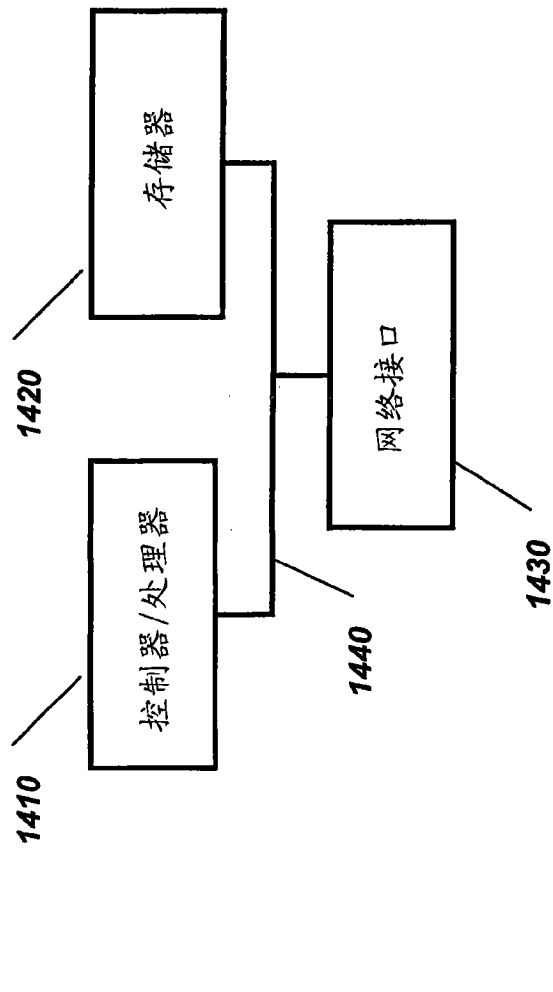


图 14