(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 103577379 B
(45)授权公告日 2016.04.13

(21)申请号 201310486505.9

(22)申请日 2013.10.17

(73)专利权人 中国人民解放军国防科学技术大学
地址 410073 湖南省长沙市开福区德雅路109号

(72)发明人 乔寓然 董辛楠 文梅 任巨 杨乾明 张春元 荀长庆 柴俊 贾文涛 黄达飞 薛云刚 蓝强

(74)专利代理机构 国防科技大学专利服务中心 43202
代理人 郭敏

(51)Int.Cl.
G06F 15/173(2006.01)
G06F 11/267(2006.01)

(56)对比文件
CN 102333038 A,2012.01.25,
EP 0676703 A2,1995.10.11,
CN 103729331 A,2014.04.16,
US 7809006 B2,2010.10.05,
宋朝晖,马光胜,宋大雷.一种新的基于NoC的死锁检测算法.《微电子学与计算机》.2009,第26卷(第3期),第30-33页.
乔寓然,伍楠,杨乾明,文梅,张春元.片上网络中基于拓扑排序的死锁检测与恢复方法.《上海交通大学学报》.2013,第47卷(第1期),第92-97页.

审查员 杨霜雪

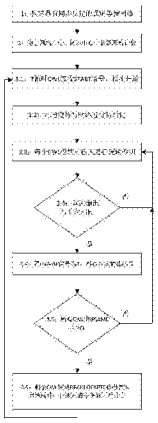
权利要求书3页 说明书9页 附图5页

(54)发明名称

一种检测片上网络中死锁的方法

(57)摘要

本发明公开了一种检测片上网络中死锁的方法,目的是提供一种既能在死锁一旦发生时可立刻检测出死锁,且发现的死锁准确无误的方法。技术方案是先根据路由网络的拓扑互连结构,构建具有同步功能的通道等待网络;在路由网络启动后,每间隔时间T基于通道等待网络对死锁进行检测:每个CWL模块获取对应的通道与相邻通道之间的初始等待关系,通过逻辑检测去除与相邻通道不存在等待关系的通道,即通道等待网络中输入或输出信号都为零的CWL模块,达到稳态网络后所有剩余CWL所对应的通道存在于一个死锁结中。本发明在最多等于通道数的时钟节拍时间内就可完成死锁检测,效率较高;且实现了精确检测,不会出现误检的情况。



CN 103577379 B

1.一种检测片上网络中死锁的方法，其特征在于包括以下步骤：

第一步，根据路由网络的拓扑互连结构，构建具有同步功能的通道等待网络：为网络路由节点间互连的N条通道建立N个与通道一一对应的通道等待标识模块CWL，根据路由网络的拓扑结构，将N个CWL互连形成通道等待网络；将每个CWL模块与同步中心相连，且与n+m个路由节点相连，实现通道等待网络的同步，N为片上网络中的通道个数，n为CWL的输出位数，m为CWL的输入位数，m和n均为正整数；

每个CWL代表网络节点间互连的一个通道，这一通道可以被m个相邻的通道申请使用，也能够申请使用n个相邻的通道，由路由网络原有的拓扑互连结构决定；每个CWL有n位输出、n位反馈输入、m位反馈输出、m位输入；m位输入连接m个相邻CWL，这m个相邻的CWL称为输入CWL，表示在网络中有m条相邻通道可以申请使用该通道；n位输出连接n个相邻CWL，这n个相邻CWL的称为输出CWL，表示在网络中有n条相邻通道可以被该通道申请使用；与输入反向的m位反馈输出用来向申请使用该通道的m条相邻通道反馈该通道的输出信息；与输出反向的n位反馈输入用来向该CWL反馈该CWL要申请使用的n条相邻通道的信息；m位输入中，某一位为1表示两个CWL模块之间存在依赖边，即在某网络通道申请使用其相邻的某个通道时，这一通道被其它CWL模块占用，二者之间具有等待关系；某一位为0表示所属CWL模块未被其它CWL模块占用，两个CWL模块之间不存在依赖边，网络通道之间没有等待关系；

每个CWL模块另通过START，IFSAME，END三个信号线与同步中心相连，实现网络的同步功能，并通过DEADLOCKPTR信号线与该CWL所代表的网络通道相连接；

通道等待标识模块CWL由初始化单元、输入判定单元、输入反馈单元、选择器、输出寄存器、输出判定单元、结果输出单元、缓存栈和比较器构成；初始化单元由输入缓存和置寄存器信号部件组成，对外与路由网络中的n个路由节点相连，并通过START信号线与同步中心相连，对内与选择器相连；置寄存器信号部件从同步中心接收START信号，当START信号为1时，将置寄存器信号置为1，送往选择器；在死锁检测开始时，输入缓存进行初始化，暂存该CWL模块所对应的通道与相邻的可被申请使用的通道之间的等待关系，当该CWL模块申请使用某一相邻通道时，二者之间存在等待关系，输入缓存对应位的值置为1，否则置为0；输入判定单元与为该CWL提供输入的m个CWL相连，对内与选择器相连；当从m个输入CWL接收的信号全部为0时，输入判定单元将n位全0信号送往选择器，当从m个输入CWL接收的信号非全0时输入判定单元无动作；选择器是一个2选1选择器，其数据输入端与初始化单元的输入缓存和输入判定单元相连，控制输入端与初始化单元的置寄存器信号部件相连，输出端与输出寄存器相连；当选择器接收到的置寄存器信号为1时，选择器从输入缓存获得n位结果，送往输出寄存器，否则从输入判定单元获得结果送往输出寄存器；输出寄存器共有n位，输出寄存器输入端与选择器相连，并通过反馈输入信号线和n位输出CWL相连，输出寄存器输出端与输出判定单元、缓存栈和比较器相连，并通过输出信号线和输出CWL相连；当从反馈输入接收的反馈信号中有1时，输出寄存器n位全部置0，没有1时则输出寄存器接收选择器的选择结果；输出寄存器将n位值送至输出判定单元、缓存栈和比较器，并向n个输出CWL模块传送输出信号，每个输出CWL模块接收一位对应的输出信号；缓存栈为n位，与输出寄存器和比较器相连，并通过CLK信号线接收来自路由网络的时钟信号，缓存栈在时钟信号每拍的上升沿处，寄存上一拍输出寄存器值；比较器对内与输出寄

存器和缓存栈相连，对外通过 IFSAME 信号线与同步中心相连，比较器将缓存栈的内容与输出寄存器的内容进行比较，当输出寄存器和缓存栈的值相同时，将 IFSAME 信号置 1 并送至同步中心；输出判定单元是一个 n 位或门，与输出寄存器、结果输出单元和输入反馈单元相连，接收来自输出寄存器的信号；若 n 位输出寄存器信号全为 0，输出判定单元将 0 送至结果输出单元和输入反馈单元，否则将 1 送至结果输出单元和输入反馈单元；结果输出单元是一个三态门逻辑结构，对内与输出判定单元相连，对外通过 DEADLOCKPTR 和 END 信号线与同步中心相连；当结果输出单元接收到来自于同步中心的 END 信号时，根据从输出判定单元接收到的信号产生 DEADLOCKPTR 信号送至同步中心，DEADLOCKPTR 为 1 标识该 CWL 所代表的通道为死锁通道；输入反馈单元对内与输出判定单元相连接，对外与 m 个相连的输入 CWL 相连，输入反馈单元从输出判定单元接收判定信号，当信号为 0 时，产生反馈输出信号将 m 个输入 CWL 模块的输出寄存器全部置为 0，当信号为 1 时不进行操作；

同步中心由计数器、同步状态机、全 1 判别部件、START 扇出部件和 END 扇出部件组成；全 1 判别部件是或门逻辑结构，对外通过 IFSAME 信号与 N 个 CWL 模块相连，对内与同步状态机相连；全 1 判别部件接收 N 位 IFSAME 信号进行全 1 判别，若 N 位 IFSAME 信号全为 1，判别信号为 1，指示所有 CWL 模块当前状态不再变化，检测结束，否则判别信号为 0 继续检测；计数器与同步状态机相连；当每次死锁检测结束时，计数器清 0，重新开始计时，到 T 拍时启动死锁检测；T 由网络规模和死锁发生的频率设定，使得计数器每到 T 拍进行一次死锁检测，计数器的每拍与路由网络的时钟节拍同步；同步状态机与全 1 判别部件、计数器、START 扇出部件和 END 扇出部件相连；同步状态机由空闲和检测两个状态组成；START 扇出部件对内与同步状态机相连，对外通过 START 信号线与所有 CWL 模块相连；当同步状态机从空闲状态进入检测状态时，通过 N 位 START 信号线向 N 个 CWL 模块发送启动信号，通道等待网络开始进行死锁检测；END 扇出部件对内与同步状态机相连，对外通过 END 信号线与所有 CWL 模块相连，当同步状态机从检测状态进入空闲状态时，通过 N 位 END 信号线将结束信号送至 N 个 CWL 模块；

第二步，启动路由网络，同步状态机处于空闲状态，向 START 扇出部件发送值为 0 的 START 扇出信号，通道等待网络的同步中心的计数器开始计数，节拍与路由网络的时钟节拍一致；

第三步，基于通道等待网络每隔时间 T 拍进行一次死锁检测，流程如下：

3.1 在同步中心的计数器到达第 T 拍时，同步状态机由空闲状态转为检测状态，向 START 扇出部件发送值为 1 的 START 扇出信号，START 扇出部件向所有的 CWL 模块发送 START 信号，N 个 CWL 模块接收到 START 扇出部件发送的 START 信号，死锁检测开始；

3.2 对通道等待网络进行初始化：每个 CWL 的输入缓存从相连的 n 个路由节点获取该 CWL 模块所对应的通道与相邻通道之间的等待关系：当该 CWL 所对应的通道申请使用某一相邻通道时，二者间存在通道等待关系，输入缓存的对应位为 1；当该 CWL 所对应的通道没有申请使用某一通道时，二者间不存在等待关系，输入缓存的对应位为 0；输出寄存器在接收来自输入缓存的信号，相应位也为 1 或 0；

3.3 每个 CWL 模块对自身的 m 位输入进行逻辑检测，方法是：

3.3.1 若 m 位输入全为 0，输出寄存器的 n 位输出信号线也全部为 0，转 3.5；

3.3.2 若 m 位输入不全为 0，则输出判定单元对从输出寄存器接收的 n 位输出信号进行

判定，如果 n 位输出全为 0，则由输入反馈单元通过反馈信号线对与所属 CWL 相连的 m 个输入 CWL 发送反馈信号，m 个输入 CWL 接收到反馈信号后，分别将各自对该 CWL 的输入也置为 0，转 3.4 ；如果 n 位输出不全为 0，则直接转 3.4 ；

3.4 每个 CWL 模块的比较器分别比较各自 n 位输出与缓存栈中上一拍的输出是否相同，如果相同，由 CWL 模块中的比较器置 IFSAME 信号为 1，并由缓存栈暂存这一拍 n 位输出信号的值，转 3.5 ；如果不同跳转到 3.3 ；

3.5 同步中心的全 1 判别部件若发现所有 CWL 模块的 IFSAME 均为 1，代表网络状态没有变化了，达到稳态网络，同步状态机由检测状态转为空闲状态，向 END 扇出部件发送值为 1 的 END 扇出信号，同步中心的 END 扇出部件向所有 CWL 模块发送 END 信号，计数器清零，检测结束，将此时输入信号或输出信号不为 0 的 CWL 模块称为剩余 CWL 模块，所有剩余的 CWL 模块所对应的通道存在于一个死锁结中 ；剩余 CWL 模块的结果输出单元置 DEADLOCKPTR 信号为 1，标识死锁通道并且通知路由节点启动死锁解锁，计数器清零重新开始计时，转 3.1 ；若全 1 判别部件发现存在有某个 CWL 模块的 IFSAME 值不为 1，则向同步状态机发送值为 0 的判别信号，同步状态机仍然处于检测状态并向 END 扇出部件发送值为 0 的 END 扇出信号，跳转到 3.3 步。

2. 如权利要求 1 所述的一种检测片上网络中死锁的方法，其特征在于当计数器没有到 T 时，同步状态机处于空闲状态，向 START 扇出部件发送值为 0 的 START 扇出信号，计数器增 1 ；当计数器到 T 拍时，同步状态机由空闲状态进入检测状态，此时向 START 扇出部件发送值为 1 的 START 扇出信号，START 扇出部件向所有的 CWL 模块发送 START 信号，检测开始 ；当从全 1 判别部件接收的判别信号为 0 时，同步状态机处于检测状态，向 END 扇出部件发送值为 0 的 END 扇出信号 ；当同步状态机从全 1 判别部件的判别信号为 1 时，全 1 判别成立，由检测状态进入空闲状态，向 END 扇出部件发送值为 1 的 END 扇出信号，此时 END 扇出部件向所有 CWL 发送 END 信号，计数器清零。

一种检测片上网络中死锁的方法

技术领域

[0001] 本文涉及一种检测片上网络中死锁的方法。

背景技术

[0002] 超大规模集成电路 (VLSI) 技术继续遵循摩尔定律，单位面积可集成的晶体管数目以每 18 个月翻一番的速度迅速增长，可以说微处理器已经进入了超 10 亿支晶体管时代。但是由于近年来 VLSI 技术进入纳米级工艺时，继续缩小器件尺寸遇到了前所未有的阻力。为了高效利用片上各种资源，片上多处理器技术 CMP(Chip Multi-Processor) 逐渐流行起来。随着单片芯片内集成的核数的增加，芯片的可扩展性也受到了限制。传统处理器的片上互连机制多采用总线结构，系统硬件开销很低，但带宽较窄，在当前 CMP 中，随着处理器核心数目的增加，总线已经无法提供足够的带宽。总线的另一个问题是容易产生故障，一旦总线故障，整个 CMP 系统就会崩溃，容错性较差。当 CMP 技术广为应用后，传统的多处理器互连方式——计算机互连网络技术也相应的移植到了片上，成为片上网络 (Network on Chip)，简称 NOC。片上网络相比总线提供了足够的带宽并比交叉开关拥有更好的扩展性与更少的硬件代价。但是设计片上网络需要考虑多个方面的问题，从网络模块设计实现角度来讲，由于信号在芯片内传输时，往往要经过数个功能部件，多个时钟周期，所以在 CMP 中片上互连已经同时主导了性能和功耗两个研究者最关心的方面，设计带宽充足，结构精巧的片上网络是提高处理器性能，减少处理器功耗，简化处理器设计复杂度不可忽视的重要研究点。此外随着片上集成的核数逐步增加，片上的结构更加复杂，组件数量的增加也造成了发生错误概率的增加，引起了片上平均失效时间 MTFB(Mean Time Between Failures) 的减少，尤其对于长时间进行密集型运算的系统来说，发生错误的概率会更高。所以 CMP 系统相对传统单核处理器更容易受到各种故障的威胁。片上网络的容错能力和处理器总体性能表现及正常工作能力密切相关。

[0003] 死锁问题同样出现在片上网络中，会导致网络性能的大幅下降，如果不对其进行专门的处理就会导致系统崩溃。当网络中的一组报文，由于申请其他报文占用而未释放的资源，就会进入等待状态而不能继续前进。如果由于这种等待占用的报文构成了一个循环，而造成所有的报文永远无法前进，就称这种情况为死锁。

[0004] 在网络中解决死锁问题对于维护网络的正常运行至关重要。目前解决死锁问题有三种策略，死锁预防，死锁避免和死锁恢复。在死锁预防策略中，网络资源以一种绝不会导致死锁的方式分配，多在数据传输之前，先保留所有传输所需的资源，然后再开始传输，当传输完全结束后，释放所有申请到的资源。死锁预防策略往往被认为太过保守，提前保留所有传输所需要的资源虽然解决了死锁问题，但是却会导致低下的网络资源利用率，事实证明，大多数被保留的网络资源在整个传输过程中都存在严重的闲置情况，无法被其他需要传输的数据所利用。死锁避免策略允许报文在网络中传输时动态的申请资源，当然这种申请只有在不会引起死锁的安全状态才会被应答。显然，在死锁避免策略中当报文通过后，会立即释放申请到的资源供其他报文使用而不必等到传输结束。相比死锁预防策略，死锁避

免策略明显提高了网络资源的利用率。当然，如何保证网络始终运行在无死锁的安全状态并不是一个简单的问题，目前主流的做法是对网络的路由函数进行一些限制，删除那些可能导致不安全状态的路由选项，这样一来，如何尽量减少对路由的限制并避免死锁的发生就成了一个研究的方向。

[0005] 上述两种策略都会在死锁发生之前就将其产生的可能性消除，也就是说阻止死锁的出现。而死锁恢复策略则不同，不会对网络资源的申请进行任何限制，也不会在分配资源时进行额外的检查来防止死锁出现，而是放任死锁出现，然后对其进行恢复。所有死锁恢复策略均基于一个基本的假设，死锁发生的频率是很小的，两次死锁发生的间隔不能大于一次解锁过程所需要的时间，否则不停地死锁检测和解锁将会造成极大地性能下降。死锁恢复策略一般都会提供一种检测手段来检测死锁，一旦检测出死锁的存在，就会采取某种手段将其从网络中移除。死锁恢复策略需要搭配死锁检测机制来确定死锁的存在，由于硬件条件的限制，往往无法获得精确死锁检测所需要的全局信息，所以精确死锁检测往往被认为不可实现。传统的死锁检测方法一般都是基于启发式算法和本地信息的非精确算法，而这些非精确算法有着其局限性，限制了死锁恢复技术的应用。

[0006] 当网络平台转移到片上以后，由于片上具有通信资源较为丰富和信号传递延迟较小的特点，为精确死锁检测传递所需的全局信息提供了有利条件。在此基础上，存在一种利用专用线路传递全局信息基于传递闭包网络的实时精确死锁检测机制。实验表明，与传统的基于本地信息的启发式非精确方法相比，这种实时精确死锁检测机制在死锁检测精度上有了较大提升，同时在性能和功耗上也比传统方法有很大优势。但是这种方法需要消耗大量的通信资源，其中有很大一部分是片上也较为珍贵的全局通信资源，硬件开销较大。

[0007] 在某一个特定的时刻，网络中的资源申请与等待情况可以用通道等待图 (Channel Wait-for Graph，CWG) 来描述，CWG 中的节点代表网络中的各条通道，如果通道 1 中的报文正在申请并等待通道 2，那么在 CWG 中，代表通道 1 的节点就会发出一条有向边，指向代表通道 2 的节点。死锁检测的目标就是在 CWG 中找到环 (cycle) 或者结 (knot)。由于检测往往需要全局信息以及大量的计算时间，所以精确的死锁检测算法实现起来非常困难。采用通道等待图 CWG 来描述某一个特定时刻网络中的资源申请与等待情况时，CWG 中的节点代表网络中的各条通道。图 $G = (V,E)$，点集 $V = \{v_0, v_1, \ldots, v_n\}$ 代表网络中的通道，边集 E 是边序偶 $\langle i,j \rangle$ 的集合，$i,j \in V$。$\langle i,j \rangle \in E$ 当且仅当在当前时刻，通道 i 正在等待通道 j。一个通道等待图中的节点可以有多条出度边，这是因为自适应路由中，路由函数 R 所提供的选择往往有多个，如果可以选择的所有输出通道都被占用了，则报文进入阻塞状态，同时等待所有可以选择的输出通道，这时一个节点的多条出度边就形成了。当通道等待图中出现结死锁就出现了，结是这样一个节点的集合 N，从 N 中的任何一个节点出发，所能到达的所有节点的集合依然是 N。结是死锁出现的充分必要条件，直观上看来，由于结内的节点代表的通道都被占用了，正在被其他节点等待，而这些通道所申请的输出通道又都是结内的其他通道，这样所有的通道都不可能被释放，这种等待关系就会持续下去形成死锁。

[0008] 目前，广为应用的检测算法是只用本地信息的非精确的启发式算法。基于时间阈值的启发式算法是一种典型的非精确死锁检测算法。其思想是一个报文在某一个通道中被阻塞时，该通道为之提供一个计数器，用以计算该报文停留的时间，当这个时间大于一个时间阈值的时候，该报文就被认为处于死锁环中。

[0009] 这种算法并不能立刻检测出死锁的存在，而且会把某些仅仅因为等待过长时间的情况错判为死锁。这种错判会激活解锁机制，而这往往会带来额外的开销。事实上，一些解锁机制是独占的，如果两个死锁同时发生，不能同时进行解锁处理。所以当该解锁机制忙于为一个错判的死锁进行解锁时，真正的死锁也许已经发生并且无法马上进行解锁。另外还有一个主要的缺陷是，时间阈值的选取是非常困难的。报文长度、网络负载、通信模式等变化的网络状态将导致与阈值时间的不匹配，致使性能下降。正是这种检测方法的低效限制了当前死锁恢复机制的应用，找到更加高效的检测算法是十分有必要的。

发明内容

[0010] 本发明要解决的技术问题是提供一种精确检测片上网络中死锁的方法，使得既能在死锁一旦发生时即可立刻检测出死锁，并且发现的死锁准确无误。

[0011] 本发明的技术方案是：

[0012] 第一步，根据路由网络的拓扑互连结构，构建具有同步功能的通道等待网络。

[0013] 为网络路由节点间互连的N条通道建立N个与通道一一对应的通道等待标识模块CWL(Channel Wait Label)，根据路由网络的拓扑结构，将N个CWL互连形成通道等待网络。将每个CWL模块与同步中心相连，且与n+m个路由节点相连（连接这些路由节点的通道与该CWL模块所对应的通道之间，存在着通道等待关系），实现通道等待网络的同步。N为片上网络中的通道个数，n为CWL的输出位数，m为CWL的输入位数，m和n均为正整数，由网络中节点间互连通道的设计决定。

[0014] 每个CWL代表网络节点间互连的一个通道，这一通道可以被m个相邻的通道申请使用，也能够申请使用n个相邻的通道，由路由网络原有的拓扑互连结构决定。每个CWL有n位输出、n位反馈输入、m位反馈输出、m位输入。m位输入连接m个相邻CWL，这m个相邻的CWL称为输入CWL，表示在网络中有m条相邻通道可以申请使用该通道；n位输出连接n个相邻CWL，这n个相邻CWL的称为输出CWL，表示在网络中有n条相邻通道可以被该通道申请使用。与输入反向的m位反馈输出用来向申请使用该通道的m条相邻通道反馈该通道的输出信息；与输出反向的n位反馈输入用来向该CWL反馈该CWL要申 请使用的n条相邻通道的信息。m位输入中，某一位为1表示两个CWL模块之间存在依赖边，即在某网络通道申请使用其相邻的某个通道时，这一通道被其它CWL模块占用，二者之间具有等待关系；某一位为0表示所属CWL模块未被其它CWL模块占用，两个CWL模块之间不存在依赖边，网络通道之间没有等待关系。每个CWL模块另通过START，IFSAME，END三个信号线与同步中心相连，实现网络的同步功能，并通过DEADLOCKPTR信号线与该CWL所代表的网络通道相连接。

[0015] 通道等待标识模块CWL由初始化单元、输入判定单元、输入反馈单元、选择器、输出寄存器、输出判定单元、结果输出单元、缓存栈和比较器构成。初始化单元由输入缓存和置寄存器信号部件组成，对外与路由网络中的n个路由节点相连（这n个路由节点的通道与该CWL模块所对应的通道相邻，并且可能被该CWL对应的通道申请使用），并通过START信号线与同步中心相连，对内与选择器相连；置寄存器信号部件从同步中心接收START信号，当START信号为1时，将置寄存器信号置为1，送往选择器；死锁检测开始时，输入缓存进行初始化，暂存该CWL模块所对应的通道与相邻的可被申请使用的通道之间的等待关系，当该CWL模块申请使用某一相邻通道时，二者之间存在等待关系，输入缓存对应位的值

置为 1，否则置为 0。输入判定单元与为该 CWL 提供输入的 m 个 CWL 相连，对内与选择器相连；当从 m 个输入 CWL 接收的信号全部为 0 时，输入判定单元将 n 位全 0 信号送往选择器，当从 m 个输入 CWL 接收的信号非全 0 时输入判定单元无动作。选择器是一个 2 选 1 选择器，其数据输入端与初始化单元的输入缓存和输入判定单元相连，控制输入端与初始化单元的置寄存器信号部件相连，输出端与输出寄存器相连；当选择器接收到的置寄存器信号为 1 时，选择器从输入缓存获得 n 位结果，送往输出寄存器，否则从输入判定单元获得结果送往输出寄存器。输出寄存器共有 n 位，输出寄存器输入端与选择器相连，并通过反馈输入信号线和 n 位输出 CWL 相连；输出寄存器输出端与输出判定单元、缓存栈和比较器相连，并通过输出信号线和输出 CWL 相连；当从反馈输入接收的反馈信号中有 1 时，输出寄存器 n 位全部置 0，没有 1 时则输出寄存器接收选择器的选择结果；输出寄存器将 n 位值送至输出判定单元、缓存栈和比较器，并向 n 个输出 CWL 模块传送输出信号，每个输出 CWL 模块接收一位对应的输出 信号。缓存栈为 n 位，与输出寄存器和比较器相连，并通过 CLK 信号线接收来自路由网络的时钟信号；缓存栈在时钟信号每拍的上升沿处，寄存上一拍输出寄存器值。比较器对内与输出寄存器和缓存栈相连，对外通过 IFSAME 信号线与同步中心相连，比较器将缓存栈的内容与输出寄存器的内容进行比较，当输出寄存器和缓存栈的值相同时，将 IFSAME 信号置 1 并送至同步中心。输出判定单元是一个 n 位或门，与输出寄存器、结果输出单元和输入反馈单元相连，接收来自输出寄存器的信号；若 n 位输出寄存器信号全为 0，输出判定单元将 0 送至结果输出单元和输入反馈单元，否则将 1 送至结果输出单元和输入反馈单元。结果输出单元是一个三态门逻辑结构，对内与输出判定单元相连，对外通过 DEADLOCKPTR 和 END 信号线与同步中心相连；当结果输出单元接收到来自于同步中心的 END 信号时，根据从输出判定单元接收到的信号产生 DEADLOCKPTR 信号送至同步中心，DEADLOCKPTR 为 1 标识该 CWL 所代表的通道为死锁通道；输入反馈单元对内与输出判定单元相连接，对外与 m 个相连的输入 CWL 相连；输入反馈单元从输出判定单元接收判定信号，当信号为 0 时，产生反馈输出信号将 m 个输入 CWL 模块的输出寄存器全部置为 0，当信号为 1 时不进行操作。

[0016] 同步中心由计数器、同步状态机、全 1 判别部件、START 扇出部件和 END 扇出部件组成。全 1 判别部件是或门逻辑结构，对外通过 IFSAME 信号与 N 个 CWL 模块相连，对内与同步状态机相连；全 1 判别部件接收 N 位 IFSAME 信号进行全 1 判别，若 N 位 IFSAME 信号全为 1，判别信号为 1，指示所有 CWL 模块当前状态不再变化，检测结束，否则判别信号为 0 继续检测。计数器与同步状态机相连；当每次死锁检测结束时，计数器清 0，重新开始计时，到 T 拍时启动死锁检测；T 由网络规模和死锁发生的频率设定，使得计数器每到 T 拍进行一次死锁检测，计数器的每拍与路由网络的时钟节拍同步。同步状态机与全 1 判别部件、计数器、START 扇出部件和 END 扇出部件相连；同步状态机由空闲和检测两个状态组成：当计数器没有到 T 时，同步状态机处于空闲状态，向 START 扇出部件发送值为 0 的 START 扇出信号，计数器增 1；当计数器到 T 拍时，同步状态机由空闲状态进入检测状态，此时向 START 扇出部件发送值为 1 的 START 扇出信号，START 扇出部件向所有的 CWL 模块发送 START 信号，检测开始； 当从全 1 判别部件接收的判别信号为 0 时，同步状态机处于检测状态，向 END 扇出部件发送值为 0 的 END 扇出信号；当同步状态机从全 1 判别部件的判别信号为 1 时，全 1 判别成立，由检测状态进入空闲状态，向 END 扇出部件发送值为 1 的 END 扇出信号，此时 END 扇出部件向所有 CWL 发送 END 信号，计数器清零。START 扇出部件对内与同步状态机相连，对

外通过START信号线与所有CWL模块相连；当同步状态机从空闲状态进入检测状态时，通过N位START信号线向N个CWL模块发送启动信号，通道等待网络开始进行死锁检测；END扇出部件对内与同步状态机相连，对外通过END信号线与所有CWL模块相连，当同步状态机从检测状态进入空闲状态时，通过N位END信号线将结束信号送至N个CWL模块。

[0017] 第二步，启动路由网络，同步状态机处于空闲状态，向START扇出部件发送值为0的START扇出信号，通道等待网络的同步中心的计数器开始计数，节拍与路由网络的时钟节拍一致。

[0018] 第三步，基于通道等待网络对死锁进行检测。路由网络启动之后，每隔时间T拍进行一次死锁检测，检测流程如下：

[0019] 3.1在同步中心的计数器到达第T拍时，同步状态机由空闲状态转为检测状态，向START扇出部件发送值为1的START扇出信号，START扇出部件向所有的CWL模块发送START信号，N个CWL模块接收到START扇出部件发送的START信号，死锁检测开始；

[0020] 3.2对通道等待网络进行初始化。每个CWL的输入缓存从相连的n个路由节点获取该CWL模块所对应的通道与相邻通道之间的等待关系：当该CWL所对应的通道申请使用某一相邻通道时，二者间存在通道等待关系，输入缓存的对应位为1；当该CWL所对应的通道没有申请使用某一通道时，二者间不存在等待关系，输入缓存的对应位为0；输出寄存器在接收来自输入缓存的信号，相应位也为1或0；

[0021] 3.3每个CWL模块对自身的m位输入进行逻辑检测，方法是：

[0022] 3.3.1若m位输入全为0，输出寄存器的n位输出信号线也全部为0，转3.5；

[0023] 3.3.2若m位输入不全为0，则输出判定单元对从输出寄存器接收的n位输出信号进行判定，如果n位输出全为0，则由输入反馈单元通过反馈信号线对与 所属CWL相连的m个输入CWL发送反馈信号，m个输入CWL接收到反馈信号后，分别将各自对该CWL的输入也置为0，转3.4；如果n位输出不全为0，则直接转3.4；

[0024] 3.4每个CWL模块的比较器分别比较各自n位输出与缓存栈中上一拍的输出是否相同，如果相同，由CWL模块中的比较器置IFSAME信号为1，并由缓存栈暂存这一拍n位输出信号的值，转3.5；如果不同跳转到3.3；

[0025] 3.5同步中心的全1判别部件若发现所有CWL模块的IFSAME均为1，代表网络状态没有变化了，达到稳态网络，同步状态机由检测状态转为空闲状态，向END扇出部件发送值为1的END扇出信号，同步中心的END扇出部件向所有CWL模块发送END信号，计数器清零，同步中心的END扇出部件向所有CWL模块发送END信号，检测结束，将此时输入信号或输出信号不为0的CWL模块称为剩余CWL模块，所有剩余的CWL模块所对应的通道存在于一个死锁结中；剩余CWL模块的结果输出单元置DEADLOCKPTR信号为1，标识死锁通道并且通知路由节点启动死锁解锁，计数器清零重新开始计时，转3.1；若全1判别部件发现存在有某个CWL模块的IFSAME值不为1，则向同步状态机发送值为0的判别信号，同步状态机仍然处于检测状态并向END扇出部件发送值为0的END扇出信号，跳转到3.3步。

[0026] 由于路由网络启动后，网络持续运行的过程中死锁都有可能会发生，所以本发明的死锁检测是每间隔时间T拍进行一次死锁检测。在路由网络停止运行后，通道等待网络也停止运行，不再进行死锁的检测。

[0027] 采用本发明可以达到以下技术效果：

[0028] 在本发明第二步的 3.3 中，由于每时钟节拍将去除至少一条通道，所以，当存在死锁时，所用时钟节拍数目少于通道数目；当不存在死锁时，所用时钟节拍数等于通道数目。因此本发明在最多等于通道数的时钟节拍时间内就可完成死锁检测，效率较高。

[0029] 由于网络中总的发送报文的数量是一定的，所以可以用检测出的正确死锁数目来说明检测的精度。本发明第二步采用的死锁检测方法实现了精确检测，不会出现误检的情况。若检测结束时通道等待网络中不存在剩余的 CWL，即所有的 CWL 模块的输入和输出均为 0，则没有死锁存在；若通道等待网络中存在输入 或输出不为 0 的剩余 CWL 模块存在，则说明网络中有回路，死锁存在。通过检测出的死锁的正确数目与其他非精确检测方法对比，体现出了精确检测算法的优势。由于本发明的死锁检测机制效率较高且不会出现误检的情况，应用后最终会转化为片上路由网络性能上的优势。

附图说明

[0030] 图 1 是一种简单死锁示例图。

[0031] 图 2 是本发明第一步构建具有同步功能的通道等待网络整体互连示例图。

[0032] 图 3 是本发明第一步构建的通道等待网络局部连接图。

[0033] 图 4 是本发明第一步构建的通道等待标识模块逻辑结构图。

[0034] 图 5 是本发明第一步构建的同步中心模块逻辑结构图。

[0035] 图 6 是图 5 中同步状态机逻辑结构图。

[0036] 图 7 是本发明总体流程图。

[0037] 图 8 是本发明第三步 CWL 进行精确死锁检测的示例图。

具体实施方式

[0038] 图 1 为一种简单的死锁示例。其中：通道 A 中的报文目的地是节点 5，它占有通道 A 并申请通道 B；通道 B 中的报文目的地是节点 4，它占有通道 B 并申请通道 C；通道 C 中的报文目的地是节点 3，它占有通道 C 并申请通道 D；通道 D 中的报文目的地是节点 2，它占有通道 D 并申请通道 E；通道 E 中的报文目的地是节点 1，它占有通道 E 并申请通道 A。这种请求与占用的循环依赖关系造成了回路中的所有报文处于阻塞状态而永远无法前进。

[0039] 图 2 是本发明第一步构建的具有同步功能的通道等待网络整体互连示例图。在路由网络中，每个路由节点均对应通道等待网络中的一个 CWL 模块，路由网络中节点间拓扑互连结构决定了通道等待网络中 CWL 模块的互连，并且每个 CWL 模块均通过同步信号线与通道等待网络的同步中心相连。

[0040] 图 3 是本发明第一步构建的通道等待网络局部连接图。每个 CWL 代表网络节点间互连的一个通道，每个 CWL 代表网络节点间互连的一个通道，这一通道可以被 m 个相邻的通道所申请使用，也能够申请使用 n 个相邻的通道。因此， 每个 CWL 的 n 位输出，将分别送到这个 CWL 可能申请等待的对应通道的 CWL 模块的输入，而作为被申请通道对应的 CWL 模块的输入中的 1 位。与输入反向的 m 位反馈输出，向申请使用该通道的 m 条相邻通道反馈该通道的信息；与输出反向的 n 位反馈输入，向该 CWL 模块反馈它所要申请使用的 n 条相邻通道的反馈信息。这样就将 N 个 CWL 模块连接成为了通道等待网络。

[0041] 图 4 是本发明第一步构建的通道等待标识模块图。通道等待标识模块 CWL 由初始

化单元、输入判定单元、输入反馈单元、选择器、输出寄存器、输出判定单元、结果输出单元、缓存栈和比较器构成。初始化单元由输入缓存和置寄存器信号部件组成，对外与路由网络中的 n 个路由节点相连（这 n 个路由节点的通道与该 CWL 模块所对应的通道相邻，并且可能被该 CWL 对应的通道申请使用），并通过 START 信号线与同步中心相连，对内与选择器相连；置寄存器信号部件从同步中心接收 START 信号，当 START 信号为 1 时，将置寄存器信号置为 1，送往选择器；死锁检测开始时，输入缓存进行初始化，暂存该 CWL 模块所对应的通道与相邻的可被申请使用的通道之间的等待关系，当该 CWL 模块申请使用某一相邻通道时，二者之间存在等待关系，输入缓存对应位的值置为 1，否则置为 0。输入判定单元与为该 CWL 提供输入的 m 个 CWL 相连，对内与选择器相连；当从 m 个输入 CWL 接收的信号全部为 0 时，输入判定单元将 n 位全 0 信号送往选择器，当从 m 个输入 CWL 接收的信号非全 0 时输入判定单元无动作。选择器是一个 2 选 1 选择器，其数据输入端与初始化单元的输入缓存和输入判定单元相连，控制输入端与初始化单元的置寄存器信号部件相连，输出端与输出寄存器相连；当选择器接收到的置寄存器信号为 1 时，选择器从输入缓存获得 n 位结果，送往输出寄存器，否则从输入判定单元获得结果送往输出寄存器。输出寄存器共有 n 位，输出寄存器输入端与选择器相连，并通过反馈输入信号线和 n 位输出 CWL 相连；输出寄存器输出端与输出判定单元、缓存栈和比较器相连，并通过输出信号线和输出 CWL 相连；当从反馈输入接收的反馈信号中有 1 时，输出寄存器 n 位全部置 0，没有 1 时则输出寄存器接收选择器的选择结果；输出寄存器将 n 位值送至输出判定单元、缓存栈和比较器，并向 n 个输出 CWL 模块传送输出信号，每个输出 CWL 模块接收一位对应的输出信号。缓存栈为 n 位，与输出寄存器和比较器相连，并通过 CLK 信号线接收来自路由网络的时钟信号；缓 存栈在时钟信号每拍的上升沿处，寄存上一拍输出寄存器值。比较器对内与输出寄存器和缓存栈相连，对外通过 IFSAME 信号线与同步中心相连，比较器将缓存栈的内容与输出寄存器的内容进行比较，当输出寄存器和缓存栈的值相同时，将 IFSAME 信号置 1 并送至同步中心。输出判定单元是一个 n 位或门，与输出寄存器、结果输出单元和输入反馈单元相连，接收来自输出寄存器的信号；若 n 位输出寄存器信号全为 0，输出判定单元将 0 送至结果输出单元和输入反馈单元，否则将 1 送至结果输出单元和输入反馈单元。结果输出单元是一个三态门逻辑结构，对内与输出判定单元相连，对外通过 DEADLOCKPTR 和 END 信号线与同步中心相连；当结果输出单元接收到来自于同步中心的 END 信号时，根据从输出判定单元接收到的信号产生 DEADLOCKPTR 信号送至同步中心，DEADLOCKPTR 为 1 标识该 CWL 所代表的通道为死锁通道；输入反馈单元对内与输出判定单元相连接，对外与 m 个相连的输入 CWL 相连；输入反馈单元从输出判定单元接收判定信号，当信号为 0 时，产生反馈输出信号将 m 个输入 CWL 模块的输出寄存器全部置为 0，当信号为 1 时不进行操作。

[0042] 图 5 是本发明第一步构建的同步中心模块逻辑结构图。同步中心由计数器、同步状态机、全 1 判别部件、START 扇出部件和 END 扇出部件组成。全 1 判别部件是或门逻辑结构，对外通过 IFSAME 信号与 N 个 CWL 模块相连，对内与同步状态机相连；全 1 判别部件接收 N 位 IFSAME 信号进行全 1 判别，若 N 位 IFSAME 信号全为 1，判别信号为 1，指示所有 CWL 模块当前状态不再变化，检测结束，否则判别信号为 0 继续检测。计数器与同步状态机相连；当每次死锁检测结束时，计数器清 0，重新开始计时，到 T 拍时启动死锁检测；T 由网络规模和死锁发生的频率设定，使得计数器每到 T 拍进行一次死锁检测，计数器的每拍与路由网

络的时钟节拍同步。同步状态机与全 1 判别部件、计数器、START 扇出部件和 END 扇出部件相连；同步状态机由空闲和检测两个状态组成：当计数器没有到 T 时，同步状态机处于空闲状态，向 START 扇出部件发送值为 0 的 START 扇出信号，计数器增 1；当计数器到 T 拍时，同步状态机由空闲状态进入检测状态，此时向 START 扇出部件发送值为 1 的 START 扇出信号，START 扇出部件向所有的 CWL 模块发送 START 信号，检测开始；当从全 1 判别部件接收的判别信号为 0 时，同步状态机处于检测状态，向 END 扇出部件发送值为 0 的 END 扇出信号；当同步状态机从全 1 判别部件的判别信号为 1 时，全 1 判别成立，由检测状态进入空闲状态，向 END 扇出部件发送值为 1 的 END 扇出信号，此时 END 扇出部件向所有 CWL 发送 END 信号，计数器清零。START 扇出部件对内与同步状态机相连，对外通过 START 信号线与所有 CWL 模块相连；当同步状态机从空闲状态进入检测状态时，通过 N 位 START 信号线向 N 个 CWL 模块发送启动信号，通道等待网络开始进行死锁检测；END 扇出部件对内与同步状态机相连，对外通过 END 信号线与所有 CWL 模块相连，当同步状态机从检测状态进入空闲状态时，通过 N 位 END 信号线将结束信号送至 N 个 CWL 模块。

[0043] 图 6 是本发明第一步同步中心状态转换图。同步状态机与全 1 判别部件、计数器、START 扇出部件和 END 扇出部件相连；同步状态机由空闲和检测两个状态组成：当计数器没有到 T 时，同步状态机处于空闲状态，向 START 扇出部件发送值为 0 的 START 扇出信号，计数器增 1；当计数器到 T 拍时，同步状态机由空闲状态进入检测状态，此时向 START 扇出部件发送值为 1 的 START 扇出信号，START 扇出部件向所有的 CWL 模块发送 START 信号，检测开始；当从全 1 判别部件接收的判别信号为 0 时，同步状态机处于检测状态，向 END 扇出部件发送值为 0 的 END 扇出信号；当同步状态机从全 1 判别部件的判别信号为 1 时，全 1 判别成立，由检测状态进入空闲状态，向 END 扇出部件发送值为 1 的 END 扇出信号，此时 END 扇出部件向所有 CWL 发送 END 信号，计数器清零。

[0044] 图 7 是本发明总体流程图。

[0045] 第一步构建具有同步功能的通道等待网络；

[0046] 第二步启动路由网络，同步中心计数器开始计数；

[0047] 第三步基于第一步构建的通道等待网络，每隔时间 T 拍进行一次死锁检测，检测流程如下：

[0048] 第 3.1 步，T 拍时 CWL 从同步中心接收 START 信号检测开始；

[0049] 第 3.2 步，对通道等待网络进行初始化；

[0050] 第 3.3 步，每个 CWL 对输入进行逻辑检测，产生相应动作；

[0051] 第 3.4 步，判断本次输出结果与上次输出结果是否相同，若相同置 IFSAME

[0052] 信号为 1，暂存输出信号值，转 3.5，否则转 3.3；

[0053] 第 3.5 步，若所有 CWL 模块 IFSAME 信号为 1，将剩余 CWL 模块的 DEADLOCKPTR 信号置为 1，标识死锁通道并启动死锁解锁，计数器清零重新开始计时，转 3.1；否则，则跳转到 3.3 步。

[0054] 图 8 是本发明第二步 CWL 进行精确死锁检测的示例图。(a) 为有向图 G 最初的状态。依次检测节点 A 至 G。第 1 步检测出节点 A 的出度为 0，将其及其所有的边删除，从图中去掉该节点，如图 8(b) 所示；第 2 步检测出节点 E 的入度为 0，将其及其所有的边删除，如图 8(c) 所示；第 3 步检测出节点 G 的入度为 0，将其及其所有的边删除，如图 8(d) 所示；

得到图 8(d) 后发现没有入度为 0 的节点也没有出度为 0 的节点，算法停止，但是依然有节点没有输出，说明节点 A、B、C、D、E、F 这五个节点之间存在回路。其实，(d) 表示了一个结，从 B、C、D、F 任意一个节点出发，最后均只能到达其中的某个节点。

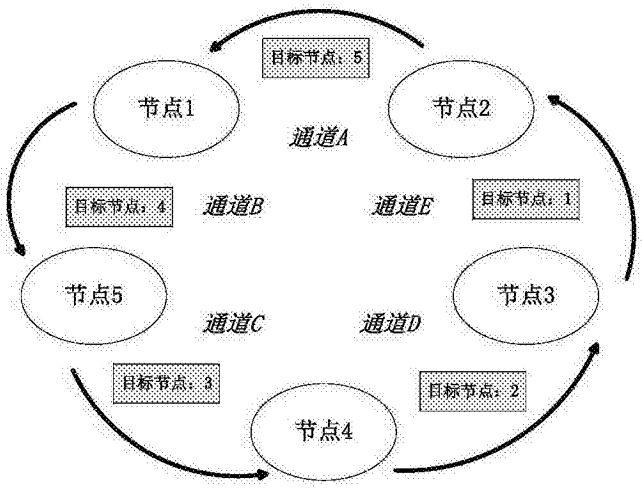

目标节点：5
节点1
节点2
通道A
目标节点：4
通道B
通道E
目标节点：1
节点5
节点3
通道C
通道D
目标节点：3
目标节点：2
节点4

图 1

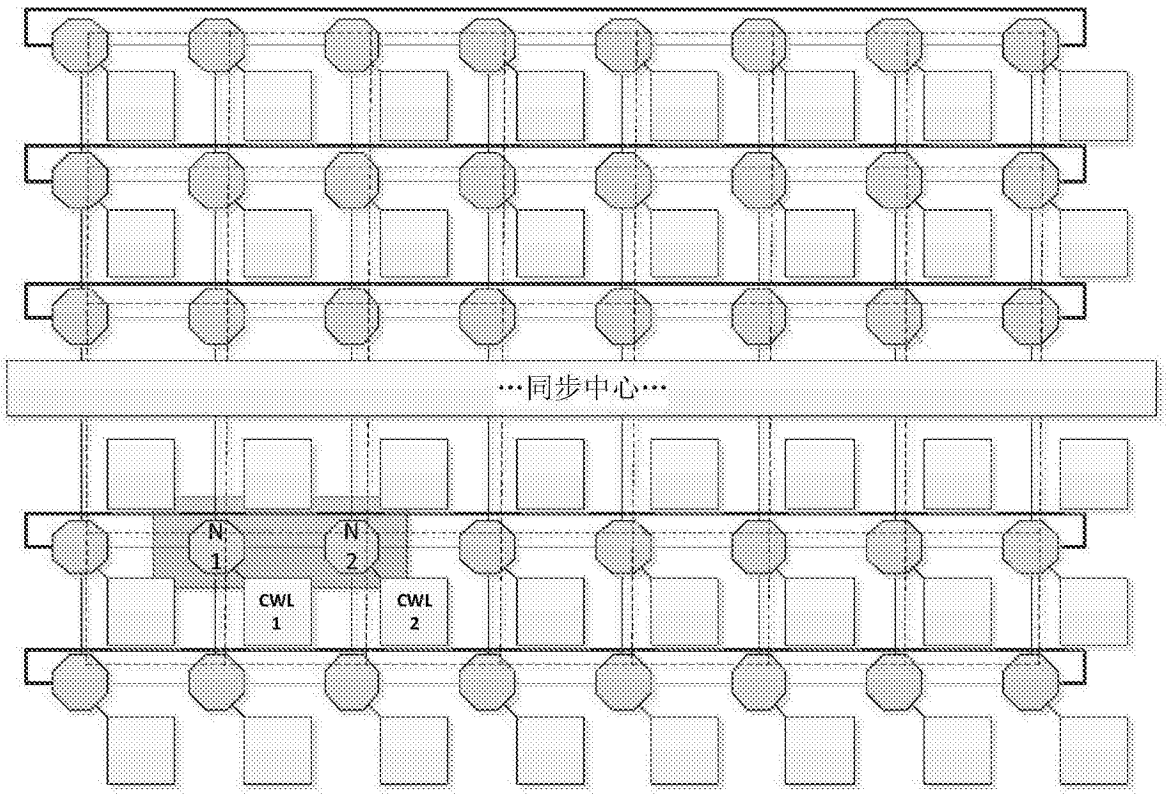

…同步中心…
N 1
N 2
CWL 1
CWL 2

图 2

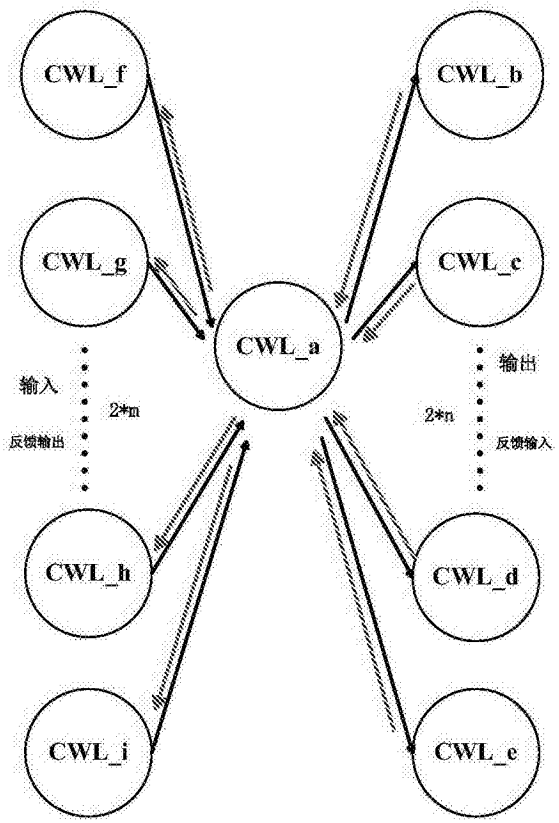

CWL_f
CWL_b
CWL_g
CWL_c
CWL_a
输入
2*m
反馈输出
输出
2*n
反馈输入
CWL_h
CWL_d
CWL_i
CWL_e

图 3

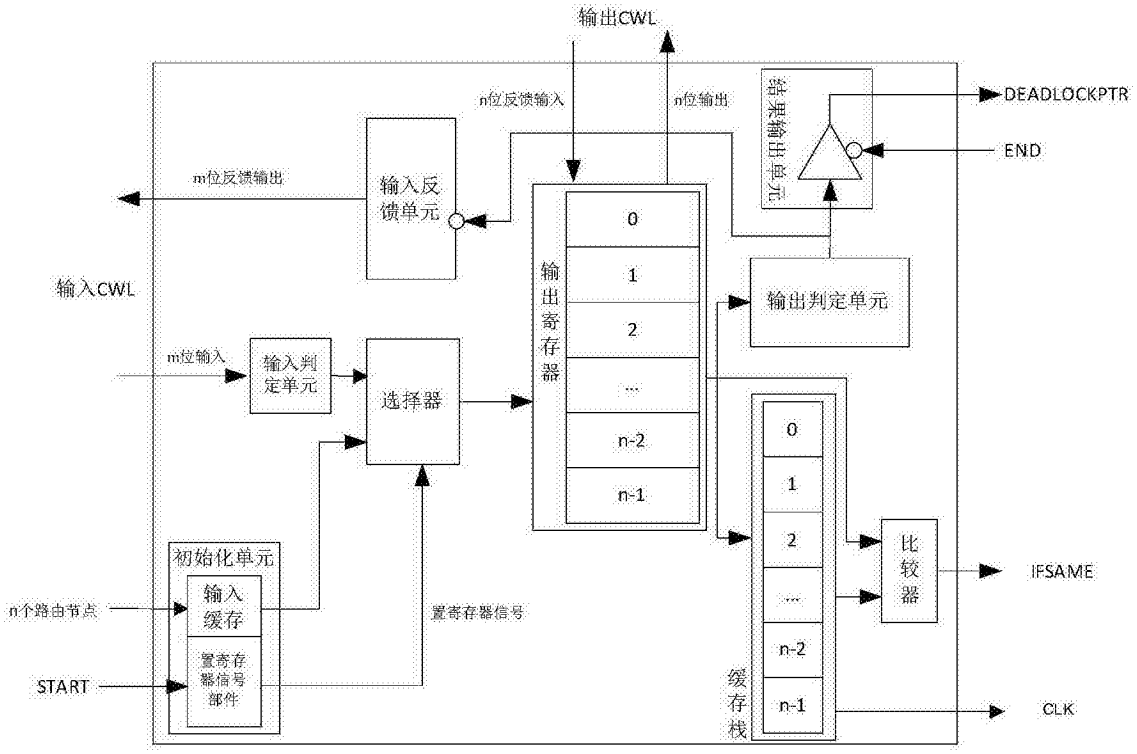

输出CWL
n位反馈输入
n位输出
结果输出单元
DEADLOCKPTR
END
m位反馈输出
输入反馈单元
输入CWL
输出寄存器
0
1
2
...
n-2
n-1
输出判定单元
m位输入
输入判定单元
选择器
初始化单元
输入缓存
置寄存器信号部件
n个路由节点
START
置寄存器信号
缓存栈
比较器
IFSAME
CLK

图 4

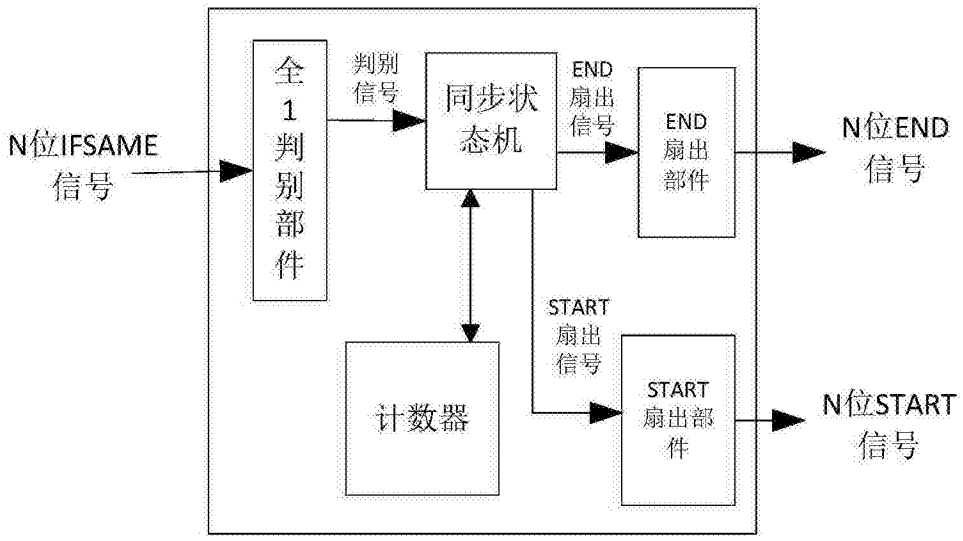

N位IFSAME
信号
全
1
判
别
部
件
判别
信号
同步状
态机
END
扇出
信号
END
扇出
部件
N位END
信号
START
扇出
信号
计数器
START
扇出部
件
N位START
信号

图 5

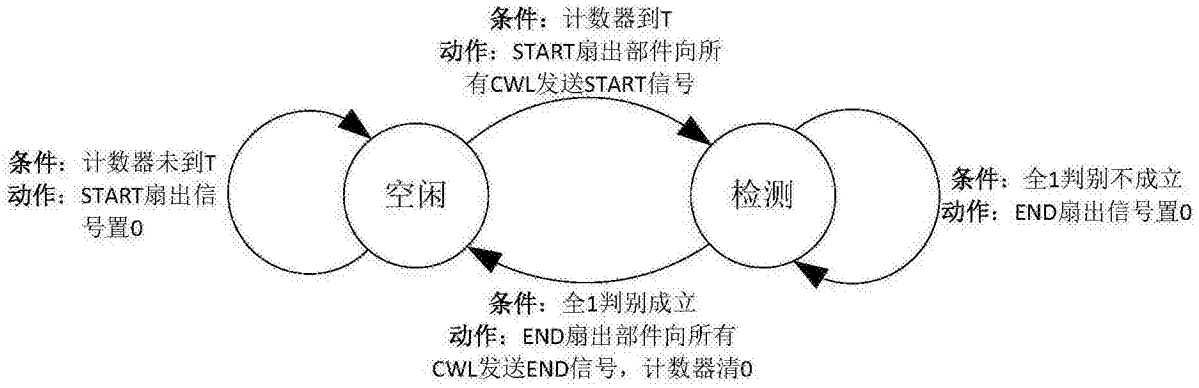

条件：计数器到T
动作：START扇出部件向所
有CWL发送START信号
条件：计数器未到T
动作：START扇出信
号置0
空闲
检测
条件：全1判别不成立
动作：END扇出信号置0
条件：全1判别成立
动作：END扇出部件向所有
CWL发送END信号，计数器清0

图 6

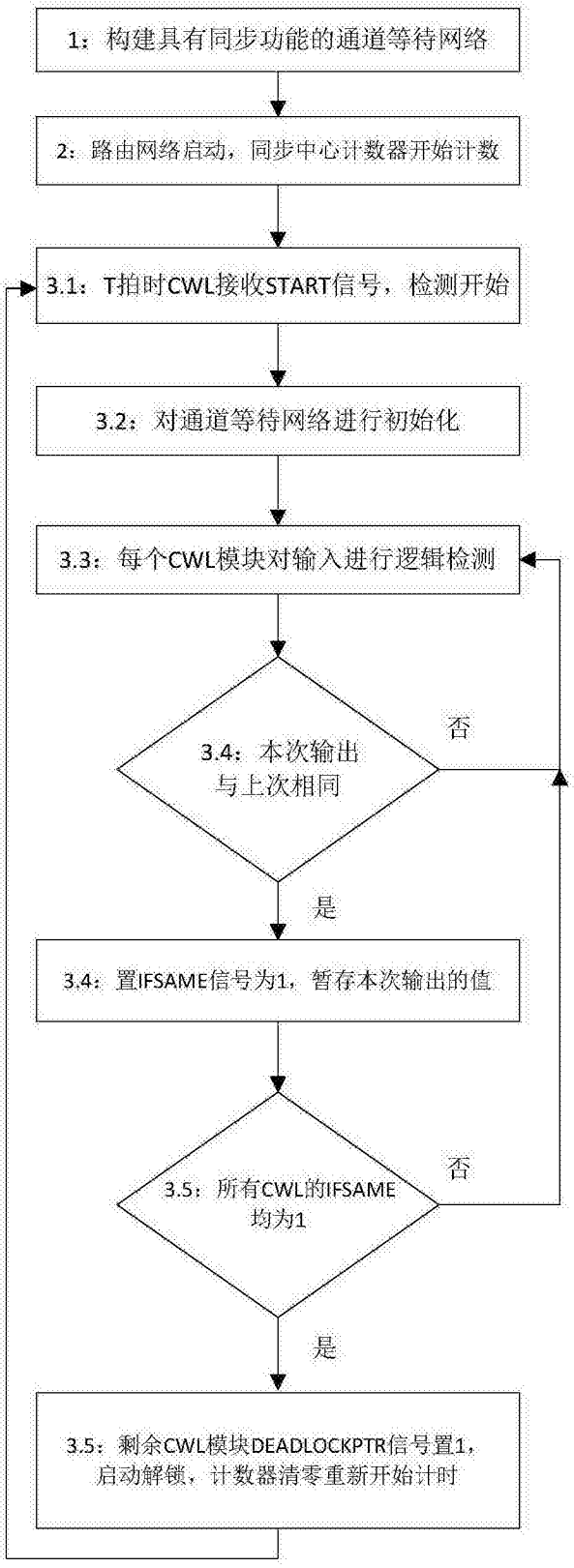

1：构建具有同步功能的通道等待网络
2：路由网络启动，同步中心计数器开始计数
3.1：T拍时CWL接收START信号，检测开始
3.2：对通道等待网络进行初始化
3.3：每个CWL模块对输入进行逻辑检测
3.4：本次输出与上次相同
否
是
3.4：置IFSAME信号为1，暂存本次输出的值
3.5：所有CWL的IFSAME均为1
否
是
3.5：剩余CWL模块DEADLOCKPTR信号置1，启动解锁，计数器清零重新开始计时

图 7

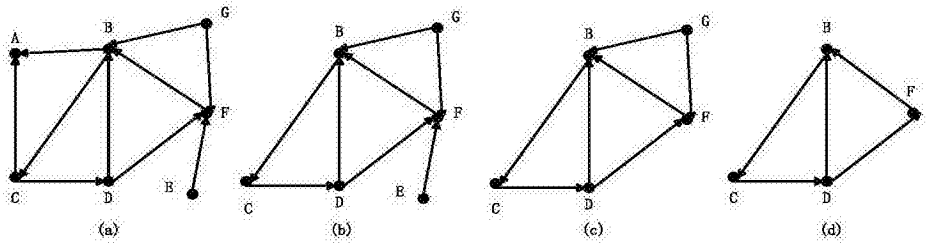


图 8