



(12) 发明专利

(10) 授权公告号 CN 109561034 B

(45) 授权公告日 2021.06.04

(21) 申请号 201811587617.2

H04L 12/931 (2013.01)

(22) 申请日 2018.12.25

H04L 12/947 (2013.01)

(65) 同一申请的已公布的文献号

申请公布号 CN 109561034 A

(56) 对比文件

CN 103763171 A, 2014.04.30

US 9331958 B2, 2016.05.03

CN 107612746 A, 2018.01.19

(43) 申请公布日 2019.04.02

(73) 专利权人 中科曙光信息产业成都有限公司

地址 610213 四川省成都市天府新区华阳

街道天府大道南段846号

专利权人 曙光信息产业(北京)有限公司

审查员 白红昌

(72) 发明人 苏勇 万伟 李斌

(74) 专利代理机构 北京德恒律治知识产权代理

有限公司 11409

代理人 章社杲 卢军峰

(51) Int. Cl.

H04L 12/933 (2013.01)

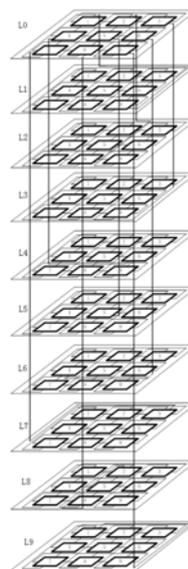
权利要求书2页 说明书12页 附图10页

(54) 发明名称

三维网络拓扑结构及其路由算法

(57) 摘要

本发明公开了一种三维网络拓扑结构及其路由算法,该三维网络拓扑结构包括:多个第一层虚拟子网,各个第一层虚拟子网为Torus拓扑结构;其中,多个第一层虚拟子网之间通过TSV互连链路形成全互连拓扑结构,全互连拓扑结构为第二层虚拟子网。本发明的上述技术方案,通过Torus拓扑和全互连拓扑相结合,基于三维集成技术和TSV技术来构建层次化网络拓扑结构,至少能够有效压缩网络直径,降低通信延迟。



1. 一种三维片上网络拓扑结构,其特征在于,包括:

多个第一层虚拟子网,各个第一层虚拟子网为Torus拓扑结构;

其中,多个第一层虚拟子网之间通过全互连拓扑结构连接,所述全互连拓扑结构为第二层虚拟子网,所述第一层虚拟子网在垂直方向的链路对应于三维集成技术中的硅通孔。

2. 根据权利要求1所述的三维片上网络拓扑结构,其特征在于,所述第一层虚拟子网为N端口虚拟交换机,第i个虚拟交换机的第j个端口的端口号为 $\langle i, j \rangle$;

其中,端口号为 $\langle i, j \rangle$ 的端口与端口号为 $\langle i+j, N+1-j \rangle$ 的端口之间的互连链路对应于三维集成技术中的硅通孔。

3. 根据权利要求2所述的三维片上网络拓扑结构,其特征在于,所述硅通孔对应的互连链路在所述多个第一层虚拟子网间垂直延伸。

4. 根据权利要求2所述的三维片上网络拓扑结构,其特征在于,所述硅通孔互连链路是由端口号为 $\langle i, j \rangle$ 的端口至端口号为 $\langle i+j, N+1-j \rangle$ 的端口的直达链路。

5. 一种如权利要求1-4任一项所述的三维片上网络拓扑结构的路由算法,其特征在于,包括:

判断目标核心与源核心是否处于相同的超级核心,其中,多个第一层虚拟子网构造为多个所述超级核心;

当所述目标核心与所述源核心处于相同的超级核心时,则通过维序路由算法路由至所述目标核心;

当所述目标核心与所述源核心处于不同的超级核心时,则通过全局路由算法路由至所述目标核心,

其中,所述第一层虚拟子网内部X、Y、Z维度的位置信息用标号 N_x 、 N_y 、 N_z 表示,其中 $N_x \in [0, K_x-1]$, $N_y \in [0, K_y-1]$, $N_z \in [0, K_z-1]$,其中 K_x 、 K_y 和 K_z 分别是所述第一层虚拟子网在X维度、Y维度和Z维度的核心数,所述第一层虚拟子网构成超级核心,

设所述源核心所在位置为 $(N_{ss}, N_{xs}, N_{ys}, N_{zs})$,所述目标核心所在位置为 $(N_{sd}, N_{xd}, N_{yd}, N_{zd})$,其中 N_{ss} 和 N_{sd} 分别是所在源和目标超级核心的编号, N_{xs} 、 N_{xd} 和 N_{ys} 、 N_{yd} 以及 N_{zs} 、 N_{zd} 分别是所述源核心和所述目标核心所在的超级核心内部X、Y和Z维度的编号。

6. 根据权利要求5所述三维片上网络拓扑结构的路由算法,其特征在于,通过维序路由算法路由至所述目标核心包括:

按照维度长度递减的顺序遍历各个维度,其中,每路由一步当前维度上的偏移量减1;

在所述当前维度的偏移量减为0后计算下一个维度的偏移量;

遍历各个维度直至所述目标核心。

7. 根据权利要求5所述三维片上网络拓扑结构的路由算法,其特征在于,通过全局路由算法路由至所述目标核心包括:

如果目标核心编号比源核心编号高,则将数据路由至本地跳转核心,所述本地跳转核心所在位置为 $(N_{ss}, N_{xm}, N_{ym}, N_{zm})$,其中, N_{ss} 是所在源核心的编号, N_{xm} 、 N_{ym} 、 N_{zm} 分别是源核心的X、Y和Z维度的编号, $N_{zm} = (N_{sd} - N_{ss} - 1) / (K_x \times K_y)$, $N_{ym} = (N_{sd} - N_{ss} - 1) \% (K_x \times K_y) / K_x$, $N_{xm} = (N_{sd} - N_{ss} - 1) \% (K_x \times K_y) \% K_x$,根据全互连的连接关系, $N_{sd} = N_{ss} + P_g$, $P_g = N_z \times K_x \times K_y + N_y \times K_x + N_x + 1$, N_x 、 N_y 表示第一层虚拟子网构中X、Y维度的位置信息, K_x 、 K_y 和 K_z 表示第一层虚拟子网构中X、Y和Z维度的核心数;

由所述本地跳转核心将所述数据路由至所述目标核心。

8. 根据权利要求5所述三维片上网络拓扑结构的路由算法, 其特征在于, 通过全局路由算法路由至所述目标核心包括:

如果目标核心编号比源核心编号低, 则将数据路由至本地跳转核心, 所述本地跳转核心所在位置为 $(N_{ss}, N_{xn}, N_{yn}, N_{zn})$, 其中, N_{ss} 是所在源核心的编号, N_{xn} 、 N_{yn} 、 N_{zn} 分别是源核心的X、Y和Z维度的编号, 所述本地跳转核心的全局端口编号为 $P_g = N + 1 - (N_{ss} - N_{sd})$, 其中 $N = K_x \times K_y \times K_z$, 根据所述第一层虚拟子网位置关系, $N_{zn} = P_g / (K_x \times K_y)$, $N_{yn} = P_g \% (K_x \times K_y) / K_x$, $N_{xn} = P_g \% (K_x \times K_y) \% K_x$;

由所述本地跳转核心将所述数据路由至所述目标核心。

三维网络拓扑结构及其路由算法

技术领域

[0001] 本发明涉及计算机技术领域,具体来说,涉及一种三维网络拓扑结构及其路由算法。

背景技术

[0002] 随着半导体器件技术的高速发展,大规模集成电路的特征尺寸越来越小,芯片内部集成电路的晶体管数量越来越多。处理器集成的核心数量也越来越多。为了将这些处理器核连接起来,协同工作,提出了专门用于芯片内部互连的通信网络,也就是片上网络(Network-on-Chip, NoC)。片上网络是指在单一芯片上实现的基于网络通信的多处理器系统。多核处理器对资源和功耗等特殊限制条件要求片上网络具有布线简单,互连线少,更低延迟,更低功耗,高吞吐率等特征。NoC具有很好的能耗效率,但随着通信量的增大,功耗约束越发成为片上网络面临的重要问题。从底层的物理设计到拓扑结构的选择都要考虑功耗、面积、通信时延和吞吐率等性能。

[0003] 拓扑结构规定了NoC中计算资源是如何在芯片中分布和连接的。对于二维平面互连网络,节点的拓扑相邻一定是空间相邻的。NoC网络要求拓扑结构具有可扩展性和可重用性。随着三维集成电路(three-dimensional integrated circuit, 3D IC)技术的发展,2D NoC逐渐发展到3D NoC。

[0004] 三维片上网络体系结构的最大特点是三维集成堆叠以及与此相适应的垂直通信链路。这些三维堆叠都是通过纵向的垂直互连进行通信。纵向垂直互连技术包括引线接合法(wirebonding)、微焊点(microbump)、无触点(contactless)以及硅通孔(Through Silicon Vias, TSV)连接等几种方式。其中,采用TSV的垂直互连技术通过在硅晶圆上通孔并以钨金属填注进行芯片间互连。

[0005] 由表1和表2所示对长互连线和TSV进行了建模对比,长互连线的功耗和时延参数见表1。3D路由器的垂直通路TSV参数如表2所示,可见采用TSV的三维堆叠的导通能力很强,在1个时钟周期内可快速互通多层,说明对于跨越不同层之间的通信延迟极低,几乎可忽略不计,而且传输延迟与通过的层数无关。当然,随着传输层数的增加,消耗的能耗也相应增加,但是功耗要远远低于长连线。

[0006] 表1互连线参数

[0007]

长度(hops)	时延(cycles)	能耗(fJ)
1	1	238
2	1	256
3	2	750
4	2	1449
5	2	2182
6	3	1641
7	3	3881

8	3	5322
---	---	------

[0008] 表2TSV通路参数

[0009]	互通层数	时延 (cycles)	能耗 (fJ)
	2	1	111
	3	1	211
	4	1	293
	>5	1	340

[0010] 基于三维集成技术,目前对三维片上网络的拓扑结构主要分为2类:一类是将2维片上网络拓扑结构直接应用在三维网络上,发挥三维立体结构的空问优势;另一类是根据三维立体结构的特点进行有针对性的设计,例如,针对垂直方向上相对于二维平面可忽略不计的距离和通信延迟以及垂直方向上TSV技术提供的超高通信带宽等。例如,三维对称网络是直接由2D Mesh拓扑在垂直方向上进行级联扩展构成,如图1所示,该结构要求路由器增加2个端口,连接Z维度的正、负2个方向(垂直方向),这会相应的增加缓存,交叉开关等资源。好处是可压缩网络直径,同时可利用TSV垂直通道,降低延迟,提高Z维度通信带宽。如果增加环回链路,将Mesh拓扑变为Torus拓扑可进一步压缩网络直径,如图2所示,而且Z维度长的环回链路通信延迟不因通信距离的增加而增加,可有效提升通信性能。但是Z维度的TSV链路成本较高,而且发热量较大,因此,有研究采用“总线”结构来对TSV链路进行时分复用,以降低网络成本。但该方法存在资源竞争问题,会影响性能。还有研究采用基于超立方体结构,亦即将三维超立方体再增加一个维度,构成四维超立方体,可压缩网络直径,但代价是增加路由器端口数量,可扩展性不足。

[0011] 高性能计算的应用负载普遍具有局部性特征:空间局部性和(或)时间局部性。基于应用的局部通信特征能够有效利用网络带宽,缩短消息延迟,提高网络吞吐率,通信局部性对计算效率有着重要影响。因此应用的通信模式是网络设计的重要依据。

[0012] 路由算法:路由算法主要负责消息的有效传输,对于网络通信性能有着重要影响。经典的3D Torus路由算法有维序路由算法、转向控制算法和自适应路由等。死锁避免:网络交换机需要一定的缓冲器用来存储数据包。但是缓冲器的容量是有限的。对于那些数据包头还没有到达目的节点的报文,一方面要请求缓冲器资源,同时又要占用当前缓冲报文的缓冲器,这种通道资源的循环占用就可能产生死锁。增加虚通道构建虚拟子网,通过分离流量可有效避免死锁,但是这种方法的缺点是所需资源比较多,对于采用高维度拓扑的超大规模网络会极大增加系统成本。采用逃逸虚通道也可以避免死锁,当发生拥塞时数据包进入逃逸虚通道路由,可化解拥塞情况,其不足之处在于逃逸通道的利用率比较低。

[0013] 针对相关技术中的上述问题,目前尚未提出有效的解决方案。

发明内容

[0014] 针对相关技术中的问题,本发明提出一种三维网络拓扑结构及其路由算法,三维网络拓扑结构采用Torus拓扑和全互连拓扑相结合,基于三维集成技术和TSV技术来构建层次化网络拓扑结构,至少能够有效压缩网络直径,降低通信延迟。

[0015] 本发明的技术方案是这样实现的:

[0016] 根据本发明的一个方面,提供了一种三维网络拓扑结构,包括:多个第一层虚拟子

网,各个第一层虚拟子网为Torus拓扑结构;

[0017] 其中,多个第一层虚拟子网之间通过TSV互连链路形成全互连拓扑结构,全互连拓扑结构为第二层虚拟子网。

[0018] 根据本发明的实施例,第一层虚拟子网为N端口虚拟交换机,第i个虚拟交换机的第j个端口的端口号为<i,j>;其中,端口号为<i,j>的端口与端口号为<i+j,N+1-j>的端口通过TSV互连链路进行连接。

[0019] 根据本发明的实施例,TSV互连链路在多个第一层虚拟子网间垂直延伸。

[0020] 根据本发明的实施例,TSV互连链路是由端口号为<i,j>的端口至端口号为<i+j,N+1-j>的端口的直达链路。

[0021] 根据本发明的另一方面,提供了一种三维网络拓扑结构的路由算法,包括:

[0022] 判断目标核心与源核心是否处于相同的超级核心,其中,多个第一层虚拟子网构造为多个超级核心;

[0023] 当目标核心与源核心处于相同的超级核心时,则通过维序路由算法路由至目标核心;

[0024] 当目标核心与源核心处于不同的超级核心时,则通过全局路由算法路由至目标核心。

[0025] 根据本发明的实施例,通过维序路由算法路由至目标核心包括:按照维度长度递减的顺序遍历各个维度,其中,每路由一步当前维度上的偏移量减1;在当前维度的偏移量减为0后计算下一个维度的偏移量;遍历各个维度直至目标核心。

[0026] 根据本发明的实施例,通过全局路由算法路由至目标核心包括:如果目标核心编号比源核心编号高,则将数据路由至本地跳转核心,本地跳转核心所在位置为(Nss,Nxm,Nym,Nzm),其中,Nss是所在源核心的编号,Nxm、Nym、Nzm分别是源核心的X、Y和Z维度的编号, $Nzm = (Nsd - Nss - 1) / (Kx \times Ky)$, $Nym = (Nsd - Nss - 1) \% (Kx \times Ky) / Kx$, $Nxm = (Nsd - Nss - 1) \% (Kx \times Ky) \% Kx$, $Nsd = Nss + Pg$, $Pg = Nz \times Kx \times Ky + Ny \times Kx + Nx + 1$,Nx、Ny表示第一层虚拟子网构中X、Y维度的位置信息,Kx、Ky和Kz表示第一层虚拟子网构中X、Y和Z维度的核心数;由本地跳转核心将数据路由至目标核心。

[0027] 根据本发明的实施例,通过全局路由算法路由至目标核心包括:如果目标核心编号比源核心编号低,则将数据路由至本地跳转核心,本地跳转核心所在位置为(Nss,Nxn,Nyn,Nzn),其中,Nss是所在源核心的编号,Nxn、Nyn、Nzn分别是源核心的X、Y和Z维度的编号, $Pg = N + 1 - (Nss - Nsd)$, $N = Kx \times Ky \times Kz$, $Nzn = Pg / (Kx \times Ky)$, $Nyn = Pg \% (Kx \times Ky) / Kx$, $Nxn = Pg \% (Kx \times Ky) \% Kx$;由本地跳转核心将数据路由至目标核心。

[0028] 本发明的上述技术方案,采用Torus拓扑和全互连拓扑相结合的方法,基于三维集成IC技术和TSV技术特点来构建层次化网络拓扑结构,针对高性能计算的通信局部性特征,充分发挥Torus拓扑的局部性优势和全互连拓扑的结构优势,有效压缩网络直径,降低通信延迟。基于低端口路由模块构建大规模网络可有效降低系统成本和功耗,并针对层次化网络结构特点提出高效的确定性路由算法和局部自适应路由算法以及基于虚通道的死锁避免方法,具有低延迟,高性能、低成本、低功耗的特点。。

附图说明

[0029] 为了更清楚地说明本发明实施例或现有技术中的技术方案,下面将对实施例中所需要使用的附图作简单地介绍,显而易见地,下面描述中的附图仅仅是本发明的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

- [0030] 图1是3D Mesh拓扑结构三维片上网络;
- [0031] 图2是3D Torus三维片上网络;
- [0032] 图3是层次化片上网络STF拓扑连接关系示意图;
- [0033] 图4是层次化片上网络STF空间结构示意图;
- [0034] 图5是STF拓扑Z维度TSV连接示意图;
- [0035] 图6是STF拓扑Z维度TSV直达连接示意图;
- [0036] 图7是层次化片上网络STF位置信息;
- [0037] 图8是Torus子网维序路由算法流程图;
- [0038] 图9是层次化片上网络STF全局路由算法流程图;
- [0039] 图10A、图10B是全互连拓扑路由死锁示意图;
- [0040] 图11A、图11B是全互连拓扑自适应路由算法死锁避免示意图;
- [0041] 图12是部分自适应路由算法流程图;
- [0042] 图13是均匀随机流量网络性能示意图;
- [0043] 图14是LuTrace流量性能示意图;
- [0044] 图15是OceanTrace流量性能示意图;
- [0045] 图16是片上网络等分宽度对比示意图。

具体实施方式

[0046] 下面将结合本发明实施例中的附图,对本发明实施例中的技术方案进行清楚、完整地描述,显然,所描述的实施例仅仅是本发明一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员所获得的所有其他实施例,都属于本发明保护的范围。

[0047] 本发明提供了一种三维网络拓扑结构,包括:多个第一层虚拟子网,各个第一层虚拟子网为Torus拓扑结构。其中,多个第一层虚拟子网之间通过TSV互连链路形成全互连拓扑结构,全互连拓扑结构为第二层虚拟子网。

[0048] 本发明的上述技术方案,采用Torus拓扑和全互连拓扑相结合的方法,基于三维集成IC技术和TSV技术特点来构建层次化网络拓扑结构,针对高性能计算的通信局部性特征,充分发挥Torus拓扑的局部性优势和全互连拓扑的结构优势,有效压缩网络直径,降低通信延迟。基于低端口路由模块构建大规模网络可有效降低系统成本和功耗,并针对层次化网络结构特点提出高效的确定性路由算法和局部自适应路由算法以及基于虚通道的死锁避免方法,具有低延迟,高性能、低成本、低功耗的特点。

[0049] 具体说来,Torus拓扑结构简单、通信度低,而且与应用的局部通信模式非常匹配,因此以Torus拓扑作为第一层虚拟子网是一种合理的选择,而全互连拓扑网络直径只有1跳,延迟最低,因此采用全互连拓扑作为第二层虚拟子网,Torus虚拟子网间以全互连拓扑

结构连接起来,构成层次化片上网络SuperTorusFabric (STF)。在Torus网络的基础之上至少增加1个端口,用于全互连拓扑连接,即可构建层次化片上网络,可有效降低系统功耗和成本。据此,本发明提出了混合Torus和全互连拓扑的片上网络结构,如图3所示。为了便于说明,以 3×3 的2D Torus为例(3D Torus可进一步压缩网络直径,提升网络性能,增加系统可扩展性),每个维度最少需要2个端口用于该维度的正、负方向路由,2D Torus需要4个端口,Torus拓扑的缺点是网络直径大,因此,采用具有直径最短的全互连结构作为第二层拓扑,至少需要1个端口用于全局路由,最少5个端口即可构建层次化片上网络(这里不考虑连接处理器核心的数据注入端口)。如果以3D Torus构建虚拟子网,则路由器最少需要7个端口。采用三维集成技术,将垂直方向的链路采用TSV通道,可极大压缩通信距离,降低通信延迟。而且将较长的全互连全局通信链路也采用TSV通道,在Z维度构建全互连拓扑,可进一步降低全局通信延迟,提高通信带宽,有效提升网络性能。如图3所示, 3×3 的Torus拓扑构成9核心(计算核心和路由器模块)的虚拟子网,每个路由器模块贡献一条全局链路,相当于构成一个9端口的虚拟交换机。9端口虚拟交换机最多可连接10个虚拟子网,每个虚拟子网占用一层,共10层集成电路,可连接90个核心,其拓扑空间结构如图4所示。网络直径5跳,而2D Torus为9跳,而且TSV通道会进一步降低全局通信延迟。如表3所示,相同规模的3D Torus拓扑网络直径为6跳,且至少需要6个互连端口,因而功耗更高,所需芯片资源更多。这种层次化结构将两种拓扑有机的结合在一起,扬长避短,使网络具有更好的通信性能、降低成本并提高可扩展性。

[0050] 表3 90核心片上网络拓扑结构

网络类型	拓扑结构	网络直径	端口数量
2D Torus	10×9	9	4
3D Torus	$6 \times 5 \times 3$	6	6
STF	$3 \times 3 \times (3 \times 3 + 1)$	5	5

[0052] 由于多核处理器对片上资源和能耗的严格要求限制了互连(路由)模块的端口数目和链路数量,决定了直接以低端口互连模块构造大规模网络难以实现性能和扩展性的兼顾。面对这一挑战,针对应用的局部性通信特征本发明提出了一种混合Torus和全互连拓扑的层次化片上网络拓扑结构,特别是基于TSV快速通道实现全互连长链路,可有效降低全局通信延迟,实现应用通信模式与拓扑结构的匹配,达到性能、扩展性和成本三者的均衡,亦即实现网络与应用的协同设计。

[0053] 根据本发明的实施例,第一层虚拟子网为N端口虚拟交换机,第i个虚拟交换机的第j个端口的端口号为 $\langle i, j \rangle$ 。其中,端口号为 $\langle i, j \rangle$ 的端口与端口号为 $\langle i+j, N+1-j \rangle$ 的端口通过TSV互连链路进行连接。在一个实施例中,TSV互连链路在多个第一层虚拟子网间垂直延伸。

[0054] 具体的,以N端口虚拟交换机为例,描述虚拟子网间的全互连拓扑连接关系。虚拟交换机端口命名: \langle 虚拟交换机号,端口号 \rangle ,这里虚拟交换机号也对应每个2D Torus拓扑所在的集成电路层号,端口号也对应该层核心对应的编号。例如 $\langle 0, 7 \rangle$ 是0号交换机的7号端口,亦即0层芯片7号核心的全局链路端口。交换机间端口连接关系:对于交换机i的端口j: $\langle i, j \rangle$,与之连接的端口是 $\langle i+j, N+1-j \rangle$,即对端交换机(i+j)的端口(N+1-j)。因此,0层芯片7号核心的全局链路端口负责连接7层芯片的3号核心,具体的端口连接关系如表4所示,按此

方法可以快速部署全互连拓扑。图5示出了STF拓扑全互连在Z维度的连接关系示意图,各层(L0,L2,⋯,L9)间采用TSV技术建立直达链路,再通过水平连线连接到对应目标核心接口,即可实现垂直方向的低延迟、高带宽快速通信链路,为清晰起见,图5仅示出了0号虚拟子网到其他各子网的TSV链路。

[0055] 表4全互连拓扑连接关系

	0 : 1 ---> 1 : 9	1 : 1 ---> 2 : 9	2 : 1 ---> 3 : 9	3 : 1 ---> 4 : 9	4 : 1 ---> 5 : 9
	0 : 2 ---> 2 : 8	1 : 2 ---> 3 : 8	2 : 2 ---> 4 : 8	3 : 2 ---> 5 : 8	4 : 2 ---> 6 : 8
	0 : 3 ---> 3 : 7	1 : 3 ---> 4 : 7	2 : 3 ---> 5 : 7	3 : 3 ---> 6 : 7	4 : 3 ---> 7 : 7
	0 : 4 ---> 4 : 6	1 : 4 ---> 5 : 6	2 : 4 ---> 6 : 6	3 : 4 ---> 7 : 6	4 : 4 ---> 8 : 6
[0056]	0 : 5 ---> 5 : 5	1 : 5 ---> 6 : 5	2 : 5 ---> 7 : 5	3 : 5 ---> 8 : 5	4 : 5 ---> 9 : 5
	0 : 6 ---> 6 : 4	1 : 6 ---> 7 : 4	2 : 6 ---> 8 : 4	3 : 6 ---> 9 : 4	5 : 1 ---> 6 : 9
	0 : 7 ---> 7 : 3	1 : 7 ---> 8 : 3	2 : 7 ---> 9 : 3	6 : 1 ---> 7 : 9	5 : 2 ---> 7 : 8
	0 : 8 ---> 8 : 2	1 : 8 ---> 9 : 2	7 : 1 ---> 8 : 9	6 : 2 ---> 8 : 8	5 : 3 ---> 8 : 7
	0 : 9 ---> 9 : 1	8 : 1 ---> 9 : 9	7 : 2 ---> 9 : 8	6 : 3 ---> 9 : 7	5 : 4 ---> 9 : 6

[0057] 在一个实施例中,TSV互连链路是由端口号为 $\langle i, j \rangle$ 的端口至端口号为 $\langle i+j, N+1-j \rangle$ 的端口的直达链路。图5的连接方式还是需要部分水平连线来连接TSV链路,因此,考虑不局限于垂直方向连接TSV链路,扩展到空间直达链路,即允许存在部分倾角,只要不影响芯片布局,尽可能基于直达链路构建TSV通道,最大限度的压缩水平连线,可进一步压缩延迟,提高性能,具体的连接方式如图6所示。图6所示的是一种理想情况,实际布线需要根据集成电路的实际情况调整进行。

[0058] 根据本发明的实施例,还提供了一种三维网络拓扑结构的路由算法,包括以下步骤:

[0059] S10,判断目标核心与源核心是否处于相同的超级核心,其中,多个第一层虚拟子网构造为多个超级核心;

[0060] S20,当目标核心与源核心处于相同的超级核心时,则通过维序路由算法路由至目标核心;

[0061] S30,当目标核心与源核心处于不同的超级核心时,则通过全局路由算法路由至目标核心。

[0062] 根据本发明的实施例,通过维序路由算法路由至目标核心包括以下步骤:按照维度长度递减的顺序遍历各个维度,其中,每路由一步当前维度上的偏移量减1;在当前维度的偏移量减为0后计算下一个维度的偏移量;遍历各个维度直至目标核心。

[0063] 根据本发明的实施例,通过全局路由算法路由至目标核心包括:如果目标核心编号比源核心编号高,则将数据路由至本地跳转核心,本地跳转核心所在位置为 $(N_{ss}, N_{xm}, N_{ym}, N_{zm})$,其中, N_{ss} 是所在源核心的编号, N_{xm} 、 N_{ym} 、 N_{zm} 分别是源核心的X、Y和Z维度的编号, $N_{zm} = (N_{sd} - N_{ss} - 1) / (K_x \times K_y)$, $N_{ym} = (N_{sd} - N_{ss} - 1) \% (K_x \times K_y) / K_x$, $N_{xm} = (N_{sd} - N_{ss} - 1) \% (K_x \times K_y) \% K_x$, $N_{sd} = N_{ss} + P_g$, $P_g = N_z \times K_x \times K_y + N_y \times K_x + N_x + 1$, N_x 、 N_y 表示第一层虚拟子网构中X、Y维度的位置信息, K_x 、 K_y 和 K_z 表示第一层虚拟子网构中X、Y和Z维度的核心数;由本地跳转核心将数据路由至目标核心。

[0064] 根据本发明的实施例,通过全局路由算法路由至目标核心包括:如果目标核心编号比源核心编号低,则将数据路由至本地跳转核心,本地跳转核心所在位置为 $(N_{ss}, N_{xn},$

Nyn, Nzn), 其中, Nss是所在源核心的编号, Nxn、Nyn、Nzn分别是源核心的X、Y和Z维度的编号, $Pg = N + 1 - (Nss - Nsd)$, $N = Kx \times Ky \times Kz$, $Nzn = Pg / (Kx \times Ky)$, $Nyn = Pg \% (Kx \times Ky) / Kx$, $Nxn = Pg \% (Kx \times Ky) \% Kx$; 由本地跳转核心将数据路由至目标核心。

[0065] 以下结合具体实施例对本发明的路由算法进行说明。

[0066] 1. 处理器核心位置信息标识

[0067] 每个Torus子网内部X、Y、Z维度的位置信息用标号Nx、Ny、Nz表示, 其中 $Nx \in [0, Kx - 1]$, $Ny \in [0, Ky - 1]$, $Nz \in [0, Kz - 1]$, 这里Kx、Ky和Kz分是Torus网络X维度、Y维度和Z维度的基数(核心数)。Torus子网构成超级核心(SuperCore), 超级核心间采用全互连的方式连接, 其标号为Ns, $Ns \in [0, Kx \times Ky \times Kz]$, 编号(Ns, Nx, Ny)即可唯一表示处理器核位置信息, 具体如图7所示, 这里给出的是基于3D Torus构建层次化拓扑STF网络的示意图, 每个3D Torus子网可以跨越不同集成电路层, 各层间垂直方向也采用TSV技术连接。图7中突出的粗线代表全局互连TSV链路, 用于构建全互连拓扑, 每个全局链路的端口号可表示为 $\langle Ns, Pg \rangle$, 其中 $Pg = Nz \times Kx \times Ky + Ny \times Kx + Nx + 1$, 亦即位于连接核心(Nx, Ny, Nz)的路由器模块对应的全局链路端口。

[0068] 2. 路由计算

[0069] 路由算法决定了信息的传输路径和通信效率, 对系统性能有重要影响, 好的路由算法应该具有简单高效, 易于工程实现的特点, 还应具备高带宽, 低延迟, 高吞吐率的通信性能, 并能避免死锁。对于多核处理器集成网络的情况, 还应考虑尽量少的占用宝贵的片上资源, 这就要求路由模块控制逻辑简单, 使用的端口数量、通道数量和缓存容量都最小化。根据层次化拓扑的结构特征基于最短路径原则设计了简单高效的路由算法。

[0070] 2.1 Torus虚拟子网内部维序路由算法

[0071] 维序路由算法是最短路径算法, 平均通信距离最短, 具有算法简单, 便于硬件实现等优点。特别是对于本发明所规定的交换芯片命名规则, 可根据交换机命名在每个维度的坐标信息直接判断路由。如果目标核心与源核心位于相同的虚拟子网内, 即位于一个Torus子网内部, 则采用维序路由算法即可实现路由。算法描述如图8所示, 设源处理器核心所在位置为(Nss, Nxs, Nys, Nzs), 目标处理器核心为(Nsd, Nxd, Nyd, Nzd), 路由模块当前位置为(Nsc, Nxc, Nyc, Nzc), 其中Nss和Nsd分别是所在源和目标超级核心的编号, 此时 $Nsd = Nsc$, Nxs, Nxd和Nys, Nyd以及Nzs, Nzd分别是源和目标处理器核心所在的超级核心内部X、Y和Z维度的编号。首先根据超级核心编号判断目标处理器核所在位置是否与源处理器核处在相同的超级核心, 如果相同, 则采取维序路由算法路由至目的地。对于3D Torus可分解相互正交的X、Y、Z维度, 很容易按照各维上偏移量之和计算出当前核心与目的处理器核心之间的距离。按照维度长度递减的顺序遍历每个维度, 每路由一步偏移量减1, 由于Torus拓扑存在环回链路, 可压缩网络直径, 因此每个维度的偏移量一定小于或等于该维度长度的一半, 可选择距离最短的输出方向, 在当前维度的偏移量减为0后才计算下一维的偏移量。维序路由依次走完每个维度直至目标处理器核心。否则, 则采用STF全局路由算法。

[0072] 2.2 STF全局路由算法

[0073] 如果目标处理器核心与源处理器核心位于不同的虚拟子网, 则需要通过全互连的全局链路才能路由到目标虚拟子网, 之后再按虚拟子网内部维序路由即可到达目的地。将与目标虚拟子网有直连通路的本地核心命名为跳转核心, 该核心负责将数据从全局链路端

口输出。因此,首先将数据路由至与目标超核心相连的本地跳转核心 ($N_{ss}, N_{xm}, N_{ym}, N_{zm}$)。如果目标超处理器核心编号比源处理器核心编号高,根据全互连的连接关系 $N_{ss} + P_g = N_{sd}$, $P_g = N_z \times K_x \times K_y + N_y \times K_x + N_x + 1$,可知与目标超核心连接的本地跳转核心编号符合如下规律: $N_{zm} = (N_{sd} - N_{ss} - 1) / (K_x \times K_y)$, $N_{ym} = (N_{sd} - N_{ss} - 1) \% (K_x \times K_y) / K_x$, $N_{xm} = (N_{sd} - N_{ss} - 1) \% (K_x \times K_y) \% K_x$ 。需将数据路由至该处理器核心 ($N_{ss}, N_{xm}, N_{ym}, N_{zm}$),再由该核心路由模块通过全局链路路由至目标超核心,此后再通过维序路由即可达到目的地。如果目标超处理器核心编号比源处理器核心编号低,首先需要将数据路由到核心编号为 ($N_{ss}, N_{xn}, N_{yn}, N_{zn}$)的本地跳转核心,跳转核心的全局端口编号为: $P_g = N + 1 - (N_{ss} - N_{sd})$,其中 $N = K_x \times K_y \times K_z$ 。根据Torus子网位置关系,可知 $N_{zn} = P_g / (K_x \times K_y)$, $N_{yn} = P_g \% (K_x \times K_y) / K_x$, $N_{xn} = P_g \% (K_x \times K_y) \% K_x$,该核心的路由模块将数据通过全局链路路由至目标超核心,再采用Torus虚拟子网内部维序路由算法将数据路由至目标处理器核心。具体算法流程如图9所示。

[0074] 2.3死锁避免算法:

[0075] 维序路由算法对于Mesh拓扑是没有死锁的,但是Torus拓扑存在环回链路,对于维度长度大于4的Torus环路是存在死锁的。这里选取Torus环路中的4个路由器为例,阐述如何造成死锁的。如图10A、图10B所示,假设有4个路由器间的数据传输是单向逆时针的,从R0到R2;R1到R3;R2到R0;R3到R1。如果同时发生数据传输,则路由器R0会占用数据通道VL01发送数据给路由器R1,而路由器R1会请求数据通道VL12,因此,VL01和VL12间具有相关性。路由器R1会占用数据通道VL12发送数据给路由器R2,而路由器R2会请求数据通道VL23,因此,VL12和VL23间具有相关性。路由器R2会占用数据通道VL23发送数据给路由器R3,而路由器R3会请求数据通道VL30,因此,VL23和VL30间具有相关性。路由器R3会占用数据通道VL30发送数据给路由器R0,而路由器R0会请求数据通道VL01,因此,VL30和VL01间具有相关性。根据Duato定理,如果通道扩展相关图连接成环,说明存在环回链路,则拓扑存在死锁,否则,拓扑是无死锁的。如图10A、图10B所示,通道扩展相关图成环,因此,虚拟子网内部存在死锁。

[0076] 将物理链路分离为多个虚通道可以去除通道间的循环依赖关系,是死锁避免的有效手段。因此,考虑增加虚通道资源来避免死锁,打破通道间的循环依赖关系,提高链路利用率。针对Torus拓扑,采用2条虚通道即可避免死锁。将物理链路的通信资源分离为2条虚通道VL0和VL1。规定如果目标核心的坐标编号高于当前核心的坐标编号,则在VL0号虚通道路由;反之,在VL1号虚通道路由。这样可去除通道间的循环依赖关系,去除死锁。以前面描述的4个路由器间通信情况为例,说明分离虚通道是如何去除死锁的。如图11A、图11B所示,对于从路由器R0到R2的路由,根据自适应路由算法规定,首先路由器R0会通过虚通道VL0_01发送数据,而对应的路由器R1会请求虚通道VL0_12,因此,虚通道VL0_01和VL0_12具有相关性;对于从路由器R1到R3的路由,首先路由器R1会通过虚通道VL0_12发送数据,而对应的路由器R2会请求虚通道VL0_23,因此,虚通道VL0_12和VL1_23具有相关性;对于从路由器R2到R0的路由,首先路由器R2会通过虚通道VL1_23发送数据,而对应的路由器R3会请求虚通道VL1_30,因此,虚通道VL1_23和VL1_30具有相关性;对于从路由器R3到R1的路由,首先路由器R3会通过虚通道VL1_30发送数据,而对应的路由器R0会请求虚通道VL0_01,因此,虚通道VL1_30和VL0_01具有相关性。对应的扩展通道相关图如图11A、图11B所示,可见并不成环,根据Duato定理可知网络不存在死锁。

[0077] 由于全互连拓扑本身存在环路,全局链路间存在循环依赖,也会导致死锁。因此,在全局链路间再增加2条虚通道(VC:Virtual Channel)破除循环依赖,当目标超核心标号大于当前超核心标号使用3号虚通道,反之,使用2号虚通道,全局链路拓展通道的相关图是无环的,因此,不存在死锁。由于全局链路和Torus内部链路都需要2条虚通道避免死锁,因此,层次化拓扑需要4条虚通道即可避免死锁。

[0078] 2.4局部自适应路由算法

[0079] 维序路由是确定性路由算法,路由不受网络状态影响,具有简单高效,易于实现的优点,但容易出现资源竞争的情况。自适应路由可根据网络状态提供多种路径选择,灵活适应网络变化,有利于提高链路带宽利用率并且具有一定容错能力。因此,在确定性最短路径路由算法的基础之上提出一种自适应路由路由算法,可根据应用的负载特征和网络特征灵活选择路由策略,最大限度的发挥网络性能。在实际应用中根据并行应用的通信特征和网络规模可以灵活采取不同的路由算法来实现快速、灵活的路由策略,以实现高效的网络通信。根据Duato定理“互连网络I的连接路由函数是无死锁的,如果存在一个通道子集 $C_1 \subseteq C$,使得路由子函数 $R_1(x, y) = R(x, y) \cap C_1 \forall x, y \in N$ 是连通的,并且其通道相关图 D_1 中没有环路”,设计部分自适应路由算法。在前面的维序路由基础之上,增加通道子集,在Mesh/Torus虚拟子网中,可以在其中根据网络负载情况自适应路由,当自适应子网通道资源不可用时再进入维序路由子网,因为,维序路由是无死锁的,因此,这种部分自适应路由也是无死锁的。其代价是需要额外增加虚通道资源,例如,最低增加1条虚通道VL4,优先在VL4虚通道路由,采用自适应路由算法:基于最短路径原则,可以在X、Y、Z维度选择选择负责最轻的端口输出,只要保证每次路由都向目标核心步进;否则,当所有维度的V4虚通道资源都不可用时,转向维序路由子网,采用维序路由。部分自适应算法流程如图12所示。

[0080] 3测试结果

[0081] 本发明已经在大规模网络模拟器SiNetSim上通过测试验证,验证路由算法正确,不存在死锁。

[0082] 本发明采用系统模拟的方法,对STF网络进行性能预测和分析。为了准确模拟真实网络的通信性能,我们在中科院计算所的并行模拟器为的基础上开发了层次化网络模拟器HyperNetSim。目前体系结构的模拟方法主要有数学建模分析、执行驱动和踪迹驱动等方式。Trace驱动模拟简单易懂,实验数据可重现,便于模拟器调试,因此本文采用合成Trace来驱动模拟器。Trace的模式有3种:首先模拟了像FFT这类同所有节点都进行通信的均匀随机流量:Uniform Random Trace,源节点以等概率向其他节点发送数据。 λ_{sd} 表示源节点和目标节点间发送通信的概率,如公式(1)表示,式中N是节点规模, N_s, N_d 分别表示源和目标节点,公式(2)、(3)含义相同;根据Lu这类主要在局部范围内发生通信的应用,抽象了LuSim Trace来模拟。源节点以较高概率向局部范围内的节点通信,以较低概率向其他全局节点通信,其通信概率由公式(2)表示。参数 ρ 表示发送全局通信的概率,本实验设置为10%,G是与源节点发送频繁通信的节点数量;对于像“洋流模拟”和“3D Stencil”这类以“近邻通信”为主要通信模式的应用,抽象了OceanSim Trace来模拟:其通信主要是在与其直接相连的节点间发生,仅有少量的全局通信,如公式(3)所示。这些Trace类型涵盖了典型高性能应用的通信模式,具有普遍意义,能反映拓扑在实际应用中的真实性能。

$$[0083] \quad \lambda_{sd} = \frac{1}{N^2} (0 \leq N_s, N_d \leq N-1) \quad (1)$$

$$[0084] \quad \lambda_{sd} = \begin{cases} \rho / N^2 & \{ \&(N_s / G \neq N_d / G) \\ & \&(|N_d - N_s| \% G \neq 0) \} \\ (1 - \rho) / N^2 & (N_s / G = N_d / G) \\ (1 - \rho) / N^2 & (|N_d - N_s| \% G = 0) \\ 0 & N_s = N_d \end{cases} \quad (2)$$

$$[0085] \quad \lambda_{sd} = \begin{cases} \rho / N^2 & \{ \&(N_s / G \neq N_d / G) \\ & \&(|N_d - N_s| \% G \neq 0) \} \\ (1 - \rho) / N^2 & \{ (|N_{dx} - N_{sx}| = 1) \\ & \&(|N_{dy} - N_{sy}| = 1) \} \\ (1 - \rho) / N^2 & \{ (|N_{dy} - N_{sy}| = 1) \\ & \&(N_{dx} = N_{sx}) \} \\ 0 & N_s = N_d \end{cases} \quad (3)$$

[0086] 为了比较STF层次化拓扑性能,这里基于 $4 \times 4 \times 4$ 的3D Torus以全互连的方式构建STF网络,命名为3DTorusFly。将3D Torus Fly拓扑与标准3D Torus拓扑和Dragonfly拓扑进行比较。在4K ($4 \times 4 \times 4 \times 65 = 4160$) 节点规模下,对三种流量进行了性能测试。如图13、图14和图15所示,其中,横轴是归一化吞吐率(Normalized Throughput),纵坐标是通信延迟(Latency),单位为时钟周期(cycles);曲线S1、S2、S3和S4分别是3D Torus、3D Torus Fly/DOR、3D Torus Fly/Adaptive和Dragonfly的变化曲线。对于均匀随机流量,由于系统规模较大,3D Torus拓扑网络直径较大,饱和吞吐率不到20%,而3D Torus Fly能达到40%,仅增加一个端口,性能提升一倍,Dragonfly拓扑性能最好,吞吐率最高;在LuSim流量的情况下,如图14所示,由于局部性特征匹配,3DTorus拓扑性能有了提升,吞吐率超过50%,而3D Torus Fly吞吐率相对于均匀随机流量增加了100%,高达80%,延迟更低,特别是自适应路由,接近Dragonfly,Dragonfly拓扑延迟更低;在Ocean Sim这种近邻通信流量的情况下,如图15所示,3DTorus拓扑的包含吞吐率最高,但是网络未到饱和状态前延迟却较高,而3DTorus Fly具有非常低的通信延迟,自适应路由的饱和吞吐率也接近80%。可见3DTorus Fly拓扑对于有通信局部性的应用具有良好的性能优势和可扩展性,Dragonfly拓扑延迟很低,但是当网络达饱和状态是延迟却快速增加,性能相对于LuSim流量略显下降是因为Ocean Trace流量的局部通信特征,造成局部链路拥塞,进而影响整体性能。

[0087] 本发明的效果:

[0088] 在现有技术背景下,如何根据众核设计制约条件,研究扩展性强的片上网络拓扑是很有意义的。本文应用三维堆叠技术优势,提出了一种层次化三维片上网络,其所需互连线少,功耗低,布线简单,具有良好的可扩展性。本文所提出的三维片上网络在有效减低网络直径的同时,利用TSV技术,可降低全局互连通信延迟,有效提升网络性能,并且有效降低了布线规模数和复杂度。

[0089] 网络拓扑的静态度量主要包括:网络的度、链路数、网络直径和等分宽度,这些特性决定拓扑的通信性能、规模、成本和可扩展性等,是评价网络性能的重要参数。表5描述了各种拓扑的静态参数。

[0090] 表5片上网络拓扑静态参数

拓扑结构	度 (Degree)	链路数 (Number of Links)	网络直径 (Diameter of Network)	等分宽度 (Bisection Width)
2D Mesh	4(基本上)	$2 \times (N - \sqrt{N})$	$2 \times (\sqrt{N} - 1)$	\sqrt{N}
3D Mesh	6(基本上)	$3 \times (N - \sqrt[3]{N^2})$	$3 \times (\sqrt[3]{N} - 1)$	$N^{2/3}$
2D Torus	4	2N	\sqrt{N} (N 为偶数); $\sqrt{N} - 1$ (N 为奇数)	$2 \times \sqrt{N}$ (N 为偶数); $2 \times (\sqrt{N} + 1)$ (N 为奇数)
3D Torus	6	3N	$3 \times (\sqrt[3]{N} / 2)$ (N 为偶数); $3 \times (\sqrt[3]{N} - 1) / 2$ (N 为奇数)	$2 \times N^{2/3}$
STF	7	$N \times 7 / 2$	$3 \times \left(\frac{\sqrt{1+4N}-1}{2}\right)^{1/3} + 1$ (N 为偶数); $3 \times \left(\frac{\sqrt{1+4N}-1}{2}\right)^{1/3} - 2$ (N 为奇数)	$\frac{(\sqrt{1+4N}+1)^2}{16}$ (N 为奇数); $\frac{(\sqrt{1+4N}+1)^2 - 4}{16}$ (N 为偶数)
Dragonfly	$\sqrt{2 \times \sqrt{1+4N} - 2}$ -1	$N \times \left(\sqrt{\frac{\sqrt{1+4N}-1}{2}} - 1/2\right)$	5	$\frac{(\sqrt{1+4N}+1)^2 - 4}{16}$ (N 为偶数); $\frac{(\sqrt{1+4N}+1)^2}{16}$ (N 为奇数)

[0092] 节点度反映节点成本,应尽量降低。Mesh网络节点度最低,但网络直径最大;Torus网络相对Mesh网络降低了网络直径,但节点度增加2;全互连拓扑节点度最大,可扩展性最差;而层次化片上网络STF拓扑仅比对应的Torus拓扑多一条全局链路,因而节点度低,系统成本低,便于模块化设计,可扩展性强。链路数量表示了网络成本和复杂度。STF拓扑的链路总数平缓增加且比Torus拓扑要少,因而系统成本低。网络直径表征网络最长距离,是衡量网络性能的重要指标,层次化拓扑STF的网络直径相对标准Mesh/Torus网络具有很大优势。等分宽度是沿等分网络最大通信带宽,对于网络性能有重要影响。如图16所示,其中S11、S12、S13、S14和S15分别表示2D Mesh、3D Mesh、3D Torus、3D Torus Fly和Dragonfly的变化曲线。相同系统规模条件下,层次化拓扑STF的等分宽度远远高于Mesh和Torus拓扑,而且规模越大,差距越明显,由于具有相同的全互连结构,STF拓扑等分宽度与Dragonfly一致,但路由器所需端口数量却少很多,可节省大量片上资源。

[0093] 片上网络的规模越大越能体现层次化片上网络STF的优越性。Adapteva Epiphany 1024核心的处理器采用2D Mesh ($32 \times 32 = 1024$) 片上网络拓扑结构,网络直径高达62跳,如此长的通信距离将严重影响通信性能,降低处理器的并行计算能力。如果采用本发明提出的层次化片上网络STF拓扑结构,基于 $4 \times 3 \times 3$ 的Torus子网构建STF拓扑,可互连 $36 \times 37 =$

1332个处理器核心,而网络直径只有9跳,仅为Adapteva Epiphany的14.5%。网络规模为1024情况下,2D Mesh拓扑的等分宽度为32链路,而STF拓扑的等分宽度高达264,是2D Mesh拓扑的8.25倍,可见其通信性能将远远优于2D Mesh拓扑。表6给出了1K核心规模的不同片上网络拓扑结构特性,可见STF拓扑的网络直径远远低于Mesh/Torus拓扑,尽管比Dragonfly拓扑高,但其所需的路由器端口数量却低很多,所需片上资源为Dragonfly的63%,可极大的降低NoC成本和功耗。

[0094] 表6 1K规模片上网络拓扑结构特征

	拓扑结构	规模	结构	网络直径	端口数
	2D Mesh	1024	32×32	62	4
[0095]	2D Torus	1024	32×32	32	4
	3D Mesh	1024	$16 \times 8 \times 8$	29	6
	3D Torus	1024	$16 \times 8 \times 8$	16	6
	STF	1332	$(4 \times 3 \times 3) \times (4 \times 3 \times 3 + 1)$	9	7
[0096]	Dragonfly	1332	$(6 \times 6) \times (6 \times 6 + 1)$	5	11

[0097] 以上所述仅为本发明的较佳实施例而已,并不用以限制本发明,凡在本发明的精神和原则之内,所作的任何修改、等同替换、改进等,均应包含在本发明的保护范围之内。

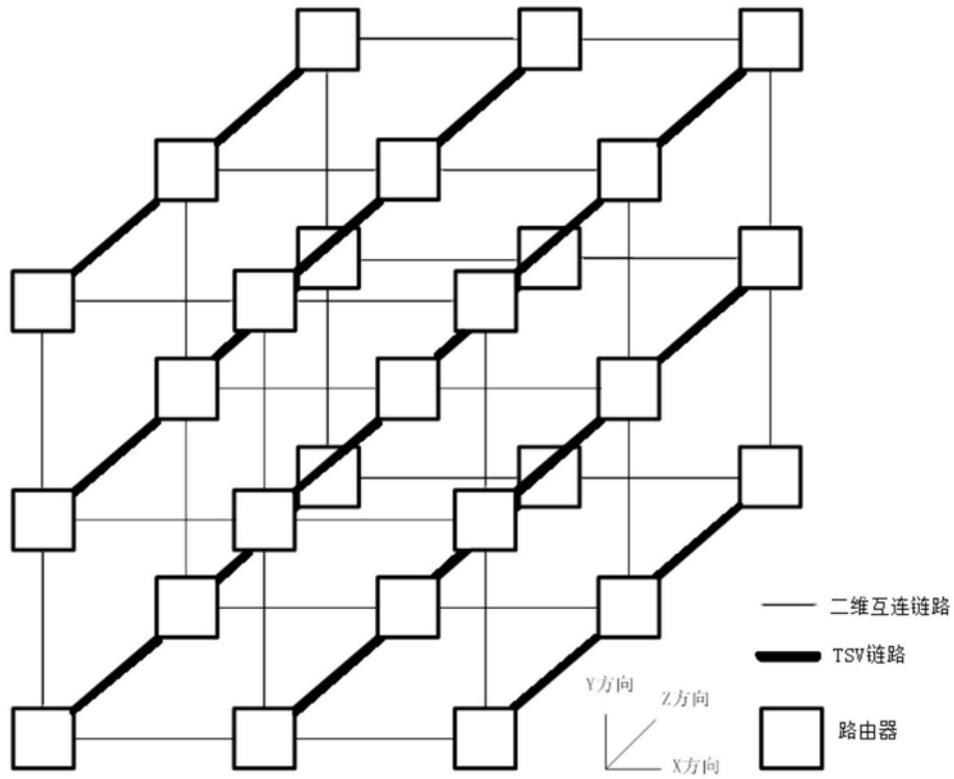


图1

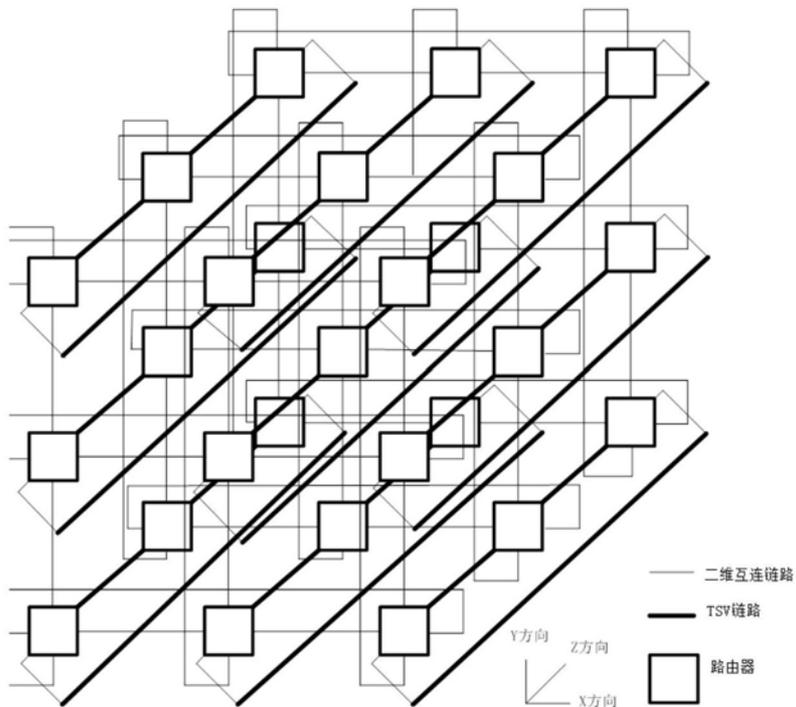


图2

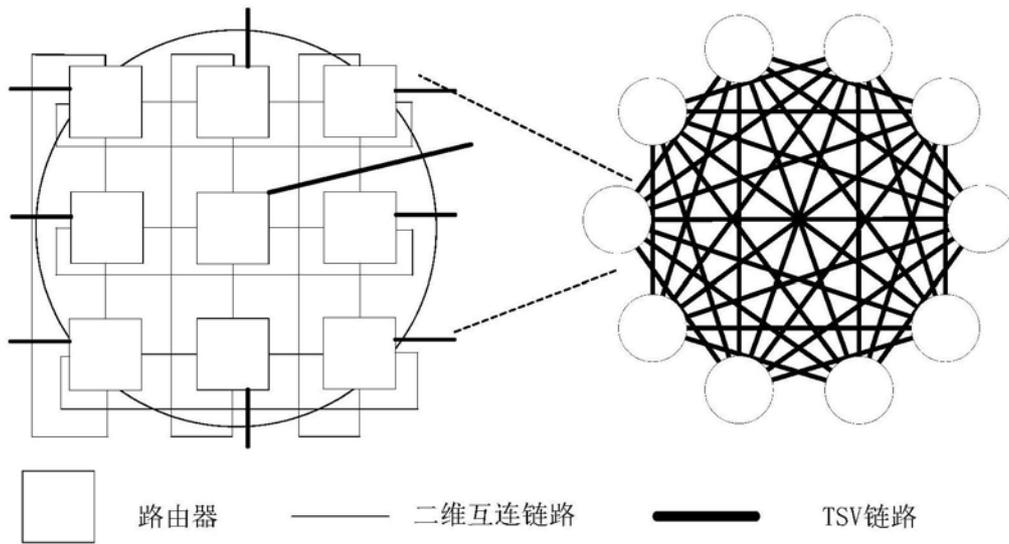


图3

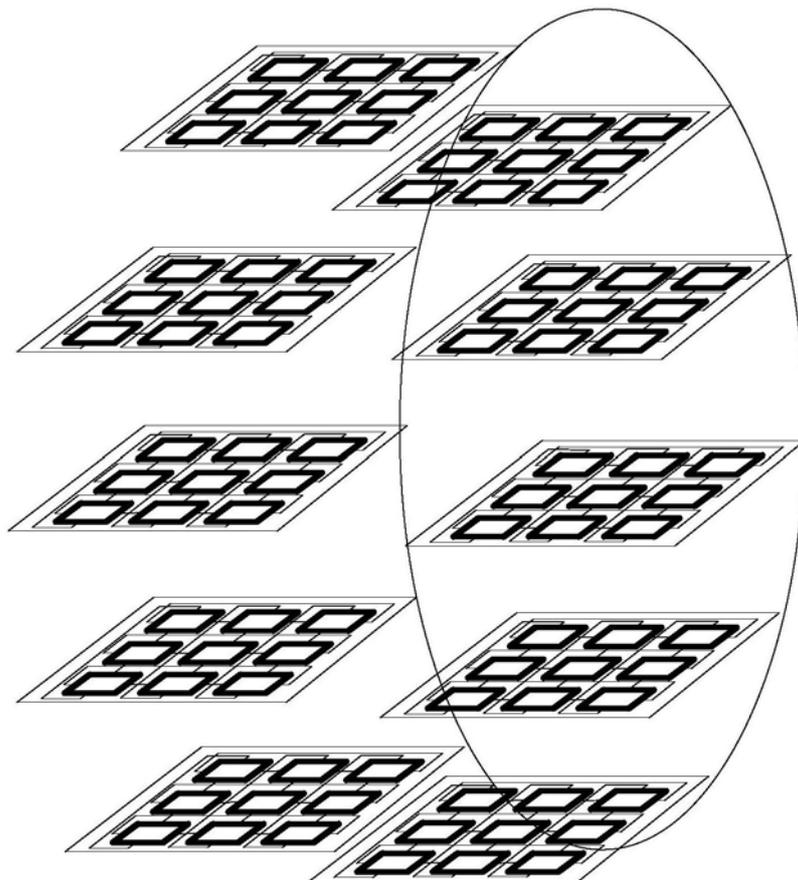


图4

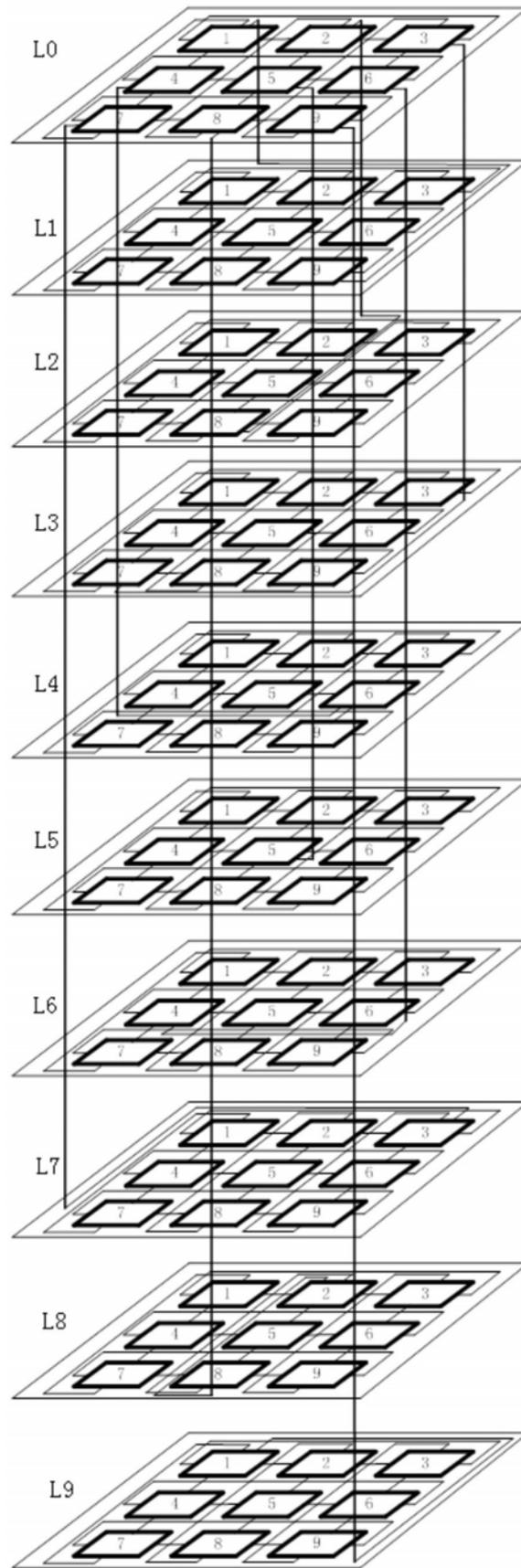


图5

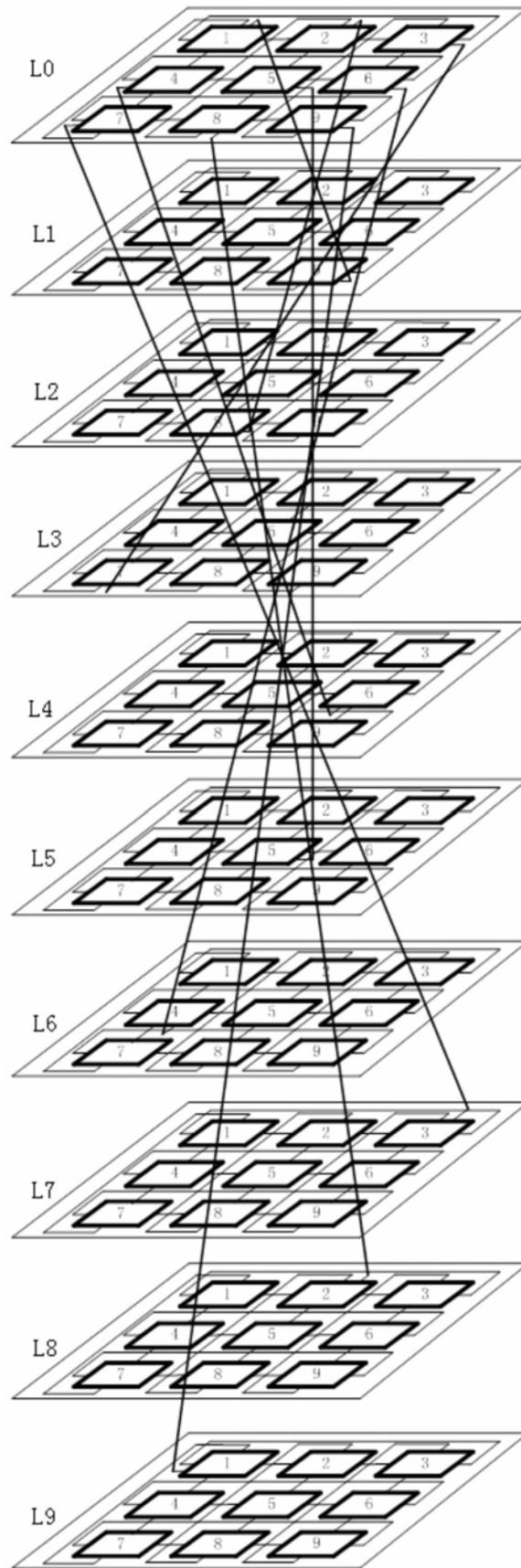


图6

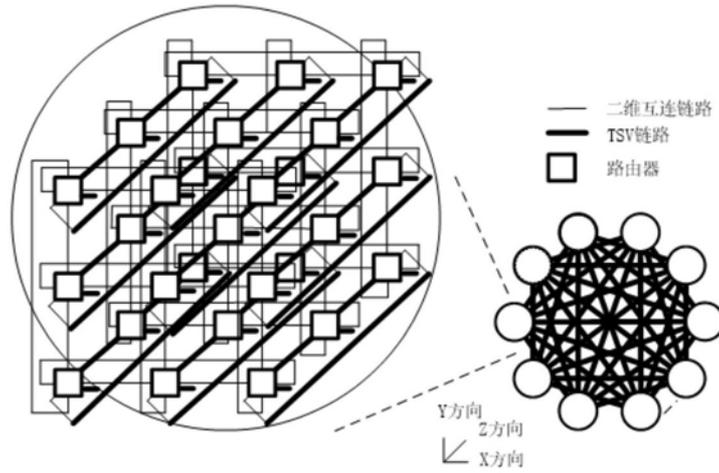


图7

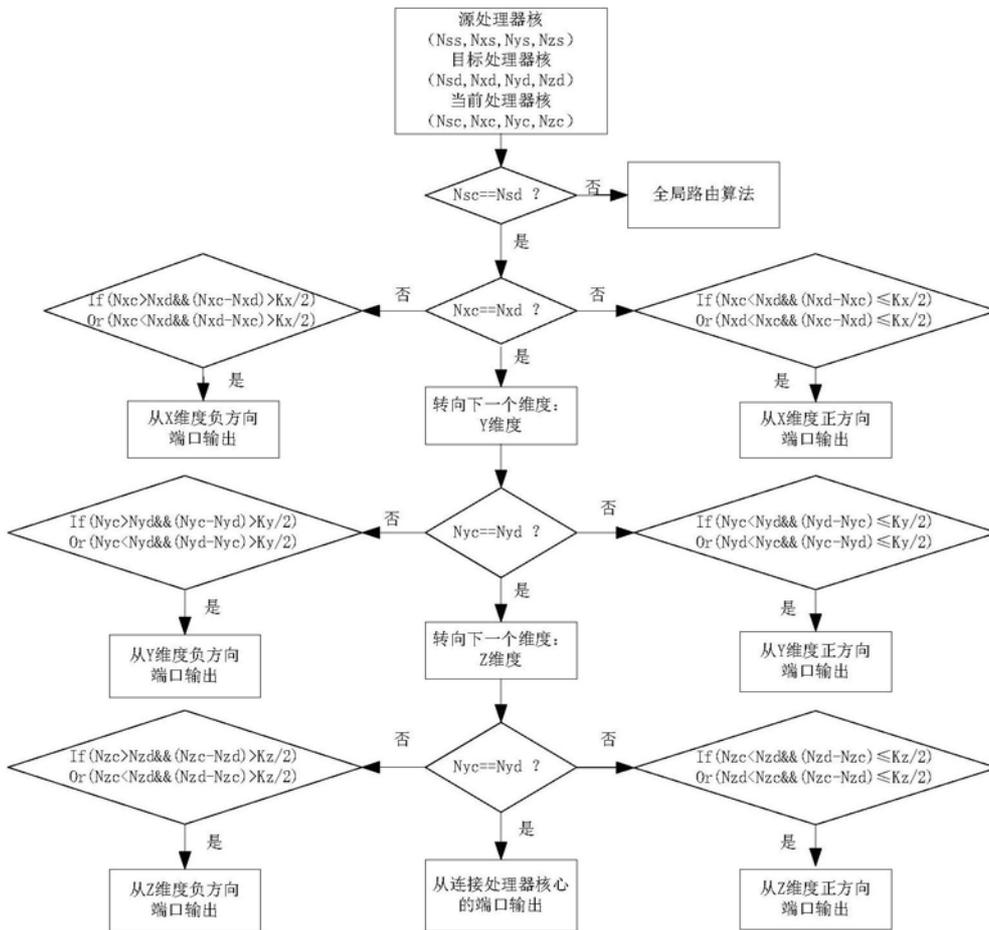


图8

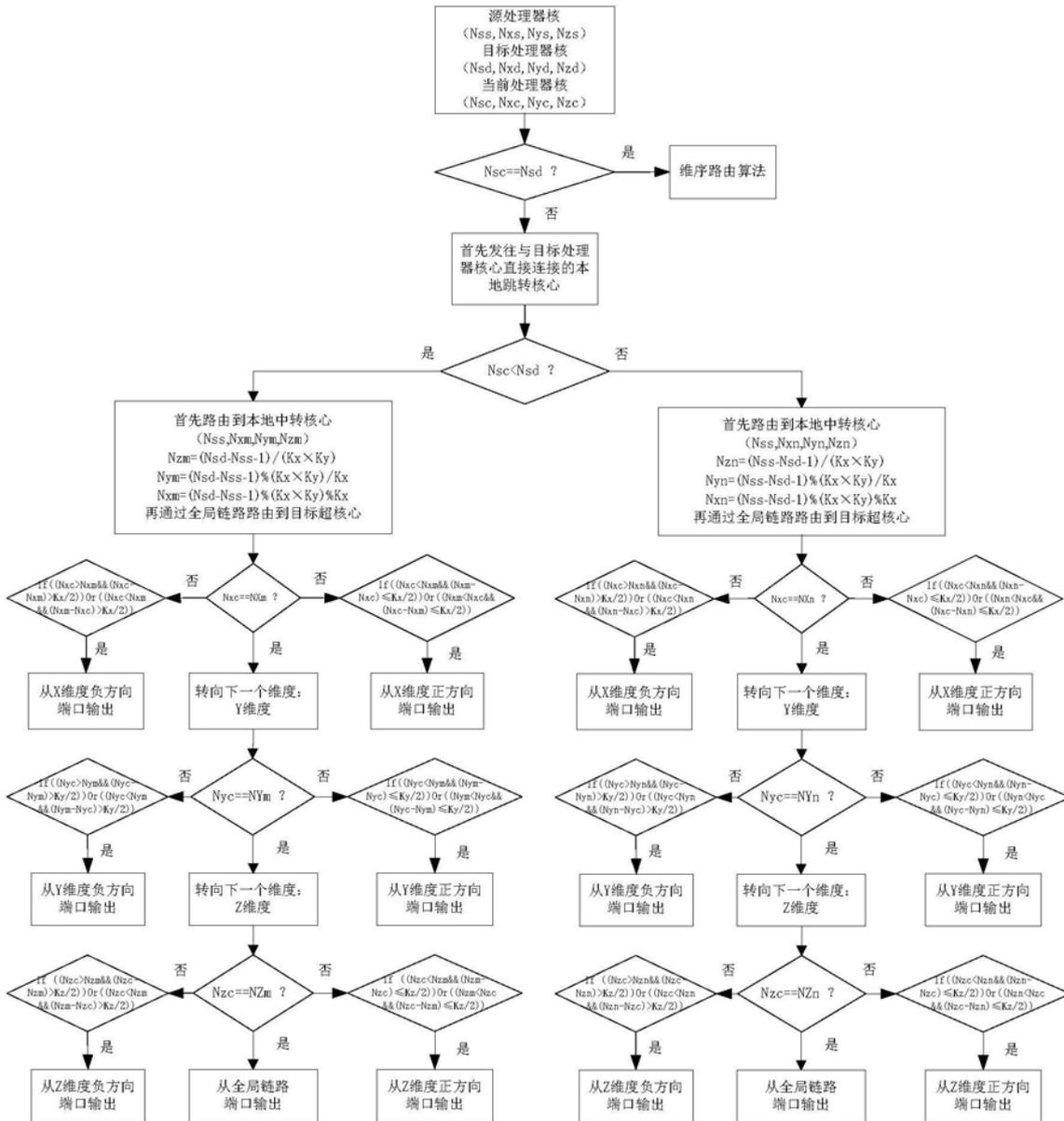


图9

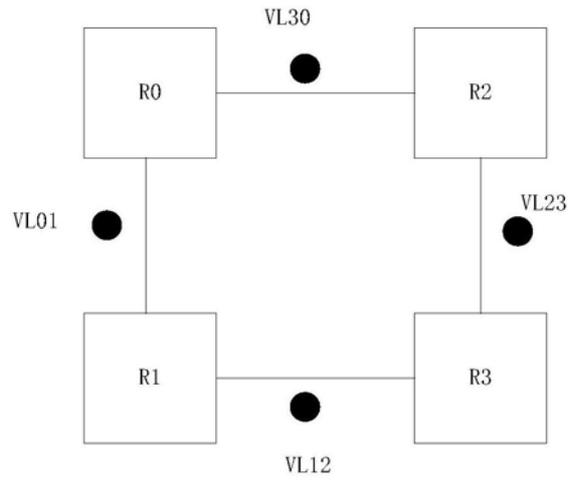


图10A

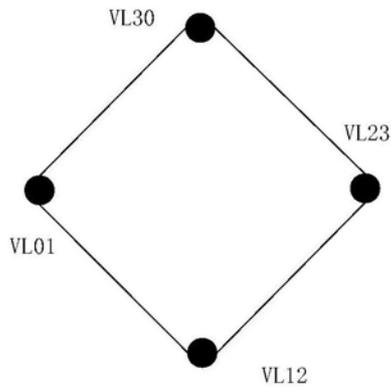


图10B

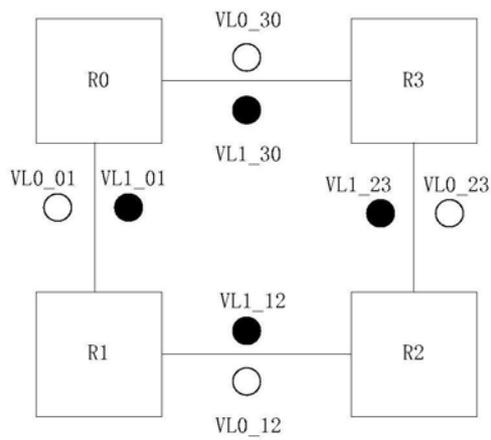


图11A

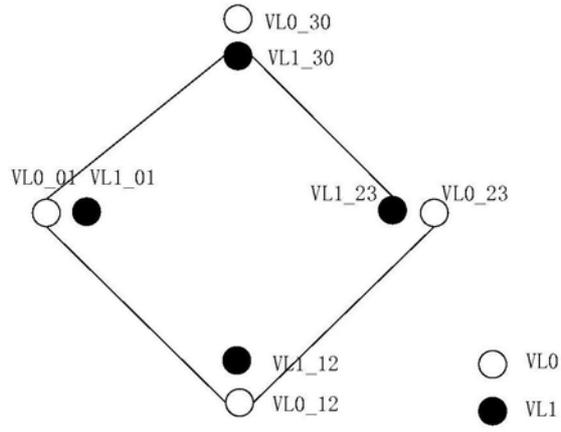


图11B

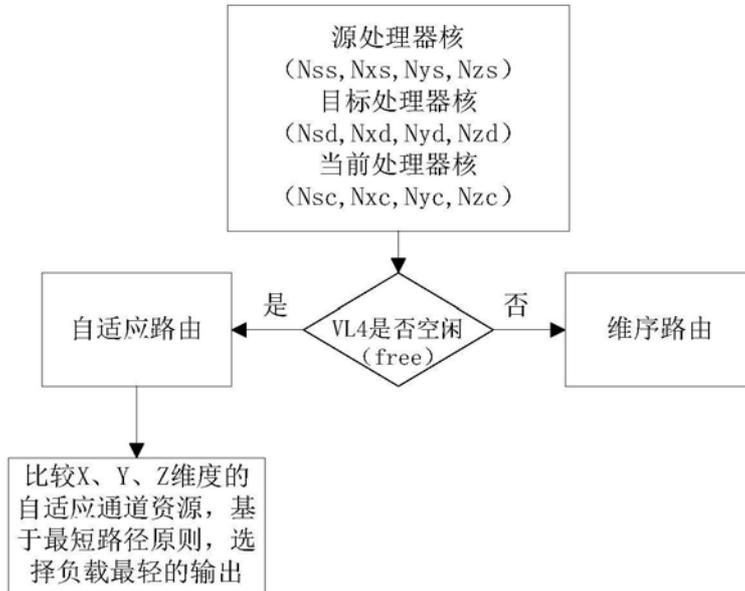


图12

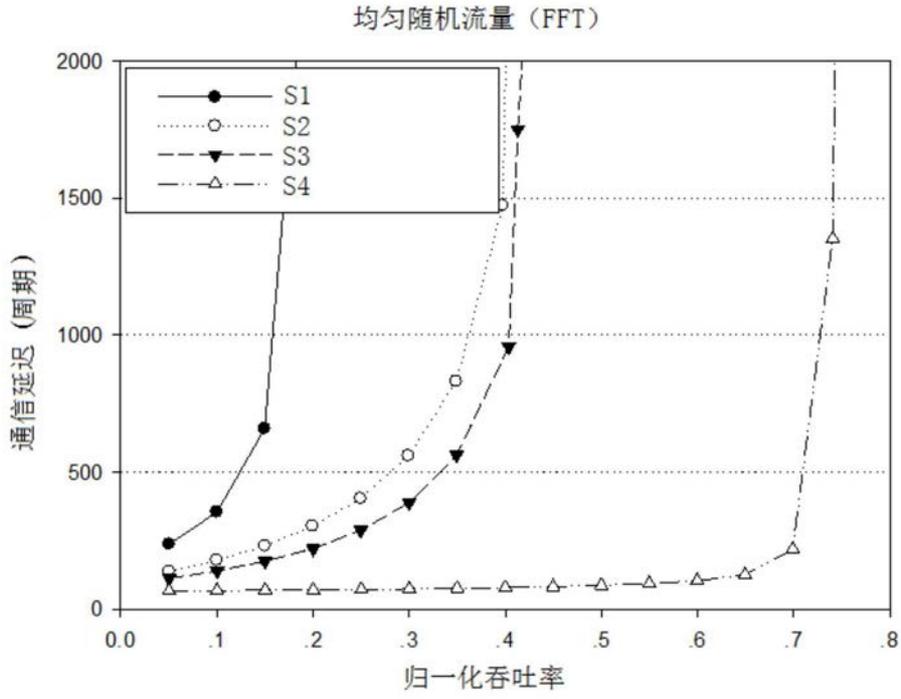


图13

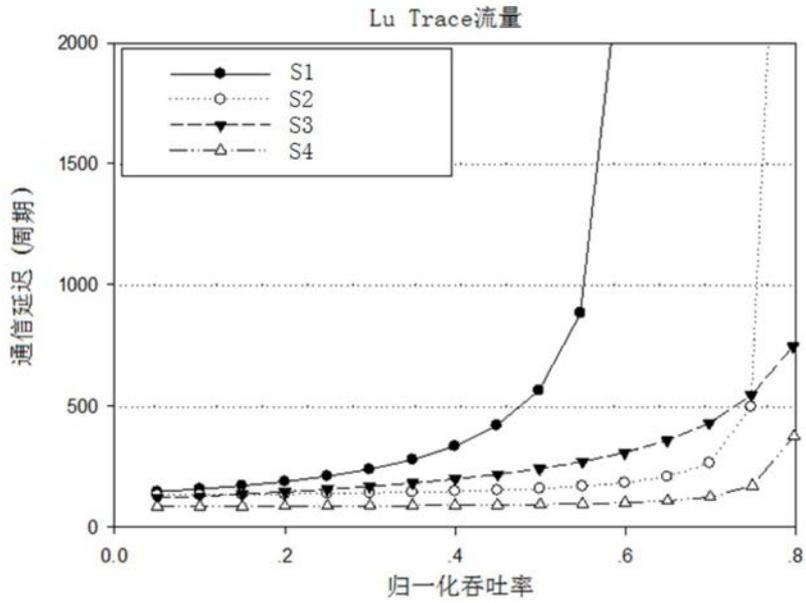


图14

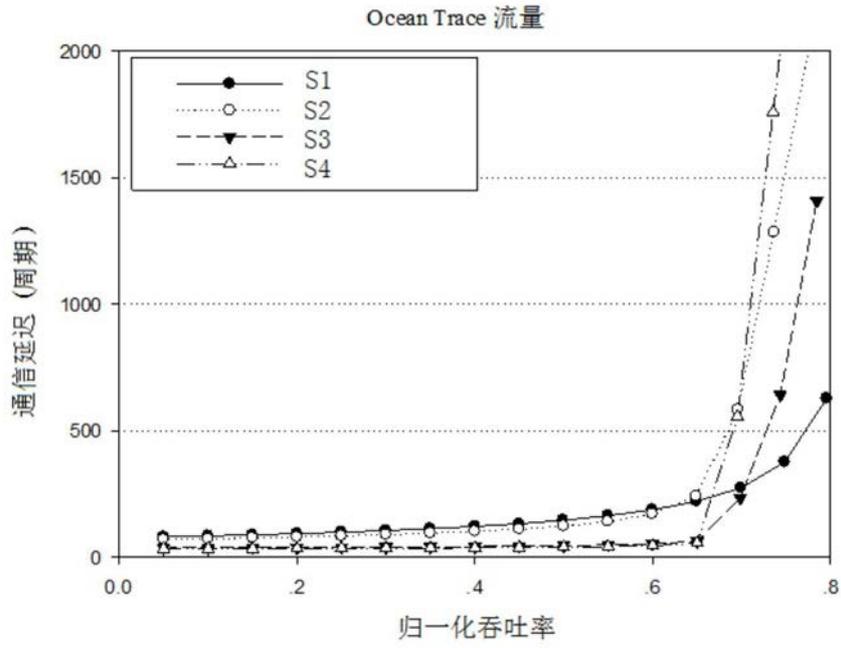


图15

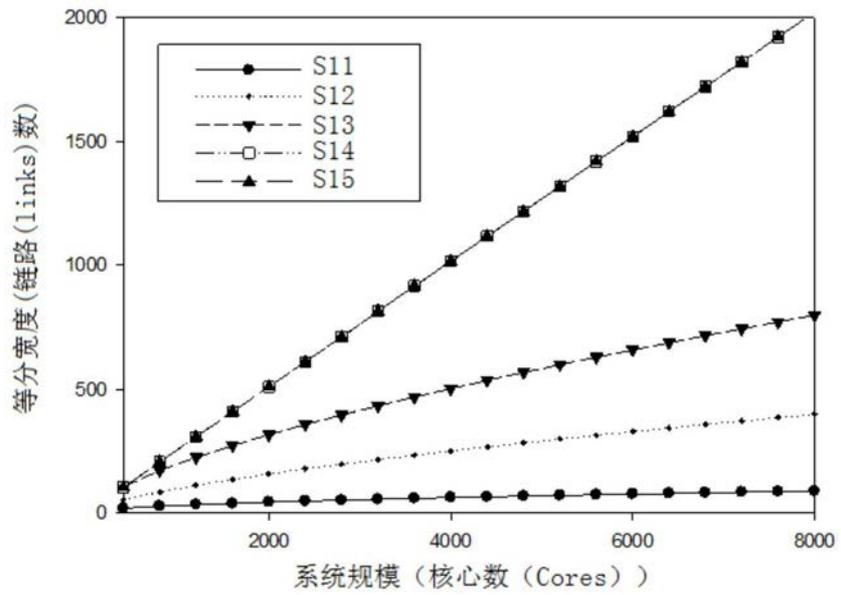


图16