中国科学: 信息科学 2018 年 第 48 卷 第 1 期: 100–114
SCIENTIA SINICA Informationis
论文

基于 S_3 变换的 TriBA-Net 最短路径路由机制

石峰*†, 陈旭†, 尹飞, 王小军, 胡森森, 计卫星, 王一拙, 高玉金, 卫晋

北京理工大学计算机学院, 北京 100081
* 通信作者. E-mail: bitsf@bit.edu.cn
† 同等贡献

收稿日期: 2017–03–30; 接受日期: 2017–05–27; 网络出版日期: 2017–12–28
国家自然科学基金 (批准号: 61300011, 61300010) 资助项目

摘要 片上网络中路由算法的设计对芯片的性能有直接的影响. 本文针对 TriBA-Net 网络提出一种新颖的最短路径路由算法. 首先, 基于 TriBA-Net 网络设计了一种编码方法, 该编码方法中所用到的文字 1, 3, 2 的集合与群论中的三文字集 S_3 群具有相同的含义. 其次, 设计了一种相隔节点间的通信模型, 根据通信路径端点的可能状态, 将通信划分为 6 种宏观数据流动模式. 最后, 利用 S_3 群的循环置换特性对通信模型进行简化, 在 XC6VLX550TL 芯片上完成了 SPR4T 路由器的设计实现. 实验结果表明, 在 27 个节点的 TriBA-Net 网络性能测试中, 在均衡负载模式下, 与 SPORT 路由算法相比, SPR4T 路由算法的饱和注入率提升 7.5%, 吞吐率提升 7.7%, 而且有效降低了硬件以及功耗的开销.

关键词 片上网络, 路由算法, 编码方法, 拓扑结构, 性能评估

1 引言

随着众核处理器技术 [1] 的进步, 内核数目不断上升, 这种硬件能力的提升保证了计算和数据处理能力的持续提高, 然而这也使得以往未能很好解决的技术和理论问题更加尖锐, 例如 “存储墙” [2] 问题, 当芯片上集成的内核增多, 处理器对主存的访问速度以及带宽的要求随之扩大. 此外, 新的问题也不断涌现, 例如众核管理 [3]. 研究表明, 随着众核处理器核数的增加, 当前众核运行时系统的核资源利用效率较低, 导致系统的可扩展性较差, 应用程序的性能不能与核数成正比增长 [4]. 如何将众核处理器的硬件能力转变为应用性能的提升, 是众核时代面临的严峻挑战之一.

针对上述问题, 本文提出面向对象的基三多核计算体系 TriBA (triplet based architecture) [5], 该体系在多核处理器片上网络 (内核间) 以及更高层次的集群计算 (计算节点间) 采用了同一拓扑结构 TriBA-Net.

引用格式: 石峰, 陈旭, 尹飞, 等. 基于 S_3 变换的 TriBA-Net 最短路径路由机制. 中国科学: 信息科学, 2018, 48: 100–114, doi: 10.1360/N112017-00065
Shi F, Chen X, Yin F, et al. A shortest path routing mechanism based on S_3 for TriBA-Net (in Chinese). Sci Sin Inform, 2018, 48: 100–114, doi: 10.1360/N112017-00065

TriBA-Net 的拓扑源于递归拓扑结构 $WK_{d,h}$ [6], 是其 $d=3$ 时的特例. $WK_{d\geqslant 4,h}$ 是被广泛认为适用于集群计算的一类拓扑结构, 而 $WK_{3,h}$ 在泛圈等拓扑特征方面劣于 $WK_{d\geqslant 4,h}$, 因此几乎无一例外地被所有研究者所忽略. 然而, 经过本文作者的深入研究发现, 被忽略的 $WK_{3,h}$ 适于用作片上网络的互连拓扑结构, 与本文所提出的一种层次化分组共享存储网络 [7] 融合后, 可以形成一个全新的高性能多核处理器架构 TriBA-CMPs. 该架构的优势如下:

(1) TriBA-CMPs 能够使用 2D-mesh 类结构常用的瓷砖 (tile) 布局布线策略实现. 从布局布线的效率来说, 2D-mesh 的布局布线策略是效率最高的一种结构.

(2) TriBA-CMPs 能够在二维平面上实现具有抽象三维结构特征的层次化分布式片上存储, 并且在相当程度上体现和简化了 PIM (processing in memory) 结构 [8] 的思想和实现机制, 进而在克服一直制约多核处理器性能的 "存储墙" 问题具有明显优势.

(3) TriBA-CMPs 通过实现多端口分组共享片上存储, 减少共享存储访存冲突, 使得核间流水性能大幅提升.

(4) 由于 $WK_{3,h}$ 也可用于片外集群互连, 尽管性能劣于 $WK_{d\geqslant 4,h}$, 但可以实现整个计算系统各层次互连拓扑的统一化, 对软件任务映射、调度和管理带来巨大便利, 这一点对基于众核处理器的计算系统十分重要.

拓扑结构和路由算法是区别不同片上网络的两个主要特征 [9]. 作为片上网络的关键技术之一, 路由算法 [10,11] 的选择对整个网络的时延、功耗 [12~14] 以及负载均衡等性能具有至关重要的影响. 针对 TriBA-Net 的特性, 现阶段已提出的路由算法有 DDRA [15], Min-DDRA [16] 和 SPORT [17]. DDRA 和 Min-DDRA 二者都是利用网络的层次特性在每个节点处为消息确定相应的输出端口, 但 DDRA 不一定是最短路径, Min-DDRA 存在重复计算的问题. SPORT 算法在路由计算时先根据源节点和目的节点的位置关系选取一个中转节点, 通过分别比较源节点和目的节点到中转节点的距离来确定输出端口, 实验结果表明 SPORT 算法在传输延迟以及吞吐率等性能指标上均优于 DDRA 和 Min-DDRA 算法, 但由于 SPORT 算法中所涉及的中转节点的选取会随着源节点和目的节点的位置关系的不同而不同, 从而导致算法较为复杂, 速度较慢 [17].

本文构建一种基于 S_3 变换的最短路径路由算法 SPR4T (shortest-path routing for TriBA-Net), 该算法既可用于采用 TriBA-Net 拓扑的片上网络, 也可用于片外计算节点的集群计算. 由于 SPR4T 本身及其实现机制最大限度地利用了 TriBA-Net 拓扑特征, 因此路由原理科学严谨、计算速度快、硬件开销较小, 能够满足低传输延迟的应用要求.

注意, 除非特别声明, 本文使用的图论术语 "顶点" 和网络术语 "节点" 含义相同.

2 TriBA-Net 的拓扑特征

2.1 TriBA-Net 的构造

TriBA-Net 是一种层次化通信网络, 除去它每个节点都连接的计算节点外, L 层 TriBA-Net 的拓扑以图的形式命名为 TG^L, 它可以看作是通过递归方式构成的.

首先, TG^1 由 3 个顶点两两互连构成 (图 1), 3 个顶点从左、中 (上)、右依次命名为 1, 3, 2, 此命名规律称为 IDC-132.

其次, 将 TG^{L-1} 视为超节点, 按 TG^1 的构造方式将 3 个这样的节点两两互连构成 TG^L. 按 IDC-132 命名 3 个超节点. TG^L 节点名由该节点原本所在 TG^{L-1} 中的名字左侧添加超节点名构成.

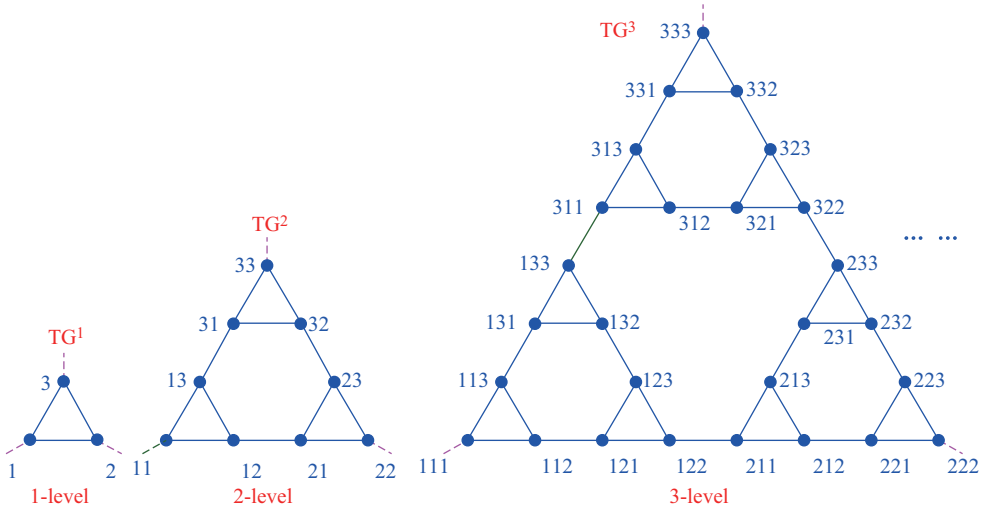

图 1 (网络版彩图) TG^L 的平面图表示

Figure 1 (Color online) Plan of TG^L

文字 1, 3, 2 的集合标记为 $\mathsf{T}=\{1,2,3\}$, 该集合与群论中常见的三文字集具有相同的含义, 本文也常不加特别说明地使用其二进制形式 $\overline{\mathsf{T}}=\{01,11,10\}$.

TG^L 顶点名显然就是 T 上长度为 L 的文字串, 因此 TG^L 的顶点集可表示为

$$V\left(\mathrm{TG}^L\right)=\{x_L\cdots x_2x_1|x_n\in\mathsf{T},\ n\in N,\ n\leqslant L\}.$$

另外, 本文称 TG^L 最底层 3 个节点构成的 TG^1 子图为基本组, 基本组内的节点名 (一位文字) 称为组内编码.

2.2 TriBA-Net 的图论定义

显然, 在图论中, TG^L 是一个无重边、无环边的简单图, 可以由顶点集和由顶点对表示的边的集合所定义, 限于篇幅这里不加讨论和证明地给出 TG^L 的边集, 进而给出 TG^L 的图论定义如下.

定义1 $\mathrm{TG}^L=\{V(\mathrm{TG}^L),E(\mathrm{TG}^L)\}$, 其中顶点集合 $V(\mathrm{TG}^L)$ 和边集合 $E(\mathrm{TG}^L)$ 分别定义如下:

$$V\left(\mathrm{TG}^L\right)=\{x_L\cdots x_l\cdots x_1|x_l\in\mathsf{T},l\in\mathbb{N},l\leqslant L\},$$

$$E\left(\mathrm{TG}^L\right)=\{\bar{x}_{L..l+1}ab^{l-1}\leftrightarrow\bar{x}_{L..l+1}ba^{l-1}|a,b\in\mathsf{T},l\in\mathbb{N},l\leqslant L,\bar{x}_{L..l+1}\in\mathsf{T}^{L-l}\},$$

其中, $\bar{x}_{L..l+1}$ 表示 $x_L\cdots x_{l+1}$, a^{l-1} 表示 l 个文字 a 的文字串, $\mathbb{N}$ 为自然数集合, $\bar{x}\leftrightarrow\bar{y}=\bar{y}\leftrightarrow\bar{x}$.

TG^L 的名字分别为 1^L, 3^L, 2^L 的顶点, 称作尖端; TG^L 中节点的数量为 3^L 两尖端间最短路径所含边数, 称为边长, TG^L 的边长为 2^L-1.

由于路由起点 (当前路由节点) 与目标节点间的最短路径总是被包含在 TG^L 的某个规模最小的 n 层子图中, 因此路由计算与该子图外其他节点和边的信息无关, 这个子图在本文称为焦点路由图 (记为 FRG^n). 例如, 图 1 中 TG^3 上部 9 个顶点和相应边构成的子图即为顶点 313 到 321 路由的 FRG^2.

FRG^n 可以看作由 3 个最大的子图拼接而成, 它们被称为最大焦路子图, 并以 MSG_1^l, MSG_3^l 和 MSG_2^l 标记. 例如前述 FRG^2 的最大焦路子图分别是顶点集 $V(\mathrm{MSG}_1^2)=\{311,313,312\}$, $V(\mathrm{MSG}_3^2)=\{331,333,332\}$ 和 $V(\mathrm{MSG}_3^2)=\{321,323,322\}$ 以及相应边组成的 3 个子图.

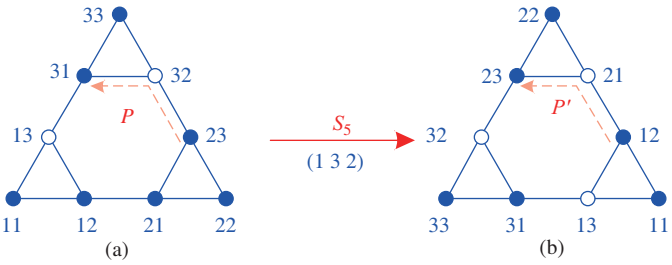

图 2 (网络版彩图) 对 TG^2 实施变换 $s_3=(123)$ 的示意图

Figure 2 (Color online) Diagram of using $s_3=(123)$ to transform TG^2

2.3 TG^L 的特征

2.3.1 TG^L 的循环置换

群论中, 三文字集上全部循环置换的集合构成一个被特别称作为 S_3 的群. 本文三文字集 $\intercal$ 上 S_3 群元素被赋予确定顺序: $S_3=\{s_i|0\leqslant i\leqslant 5\}$, 其中 $s_0=(1)$, $s_1=(12)$, $s_2=(23)$, $s_3=(123)$, $s_4=(13)$, $s_5=(132)$.

图 2 展示了对 TG^2 进行 s_5 变换的情况. 显然, 变换效果是对源图的逆时针旋转, 源图路径 $P:23\to 32\to 31$ 被变换为像图路径 $P':12\to 21\to 23$. 实际上 S_3 元素对 TG^L 的变换效果就是旋转、反射等.

可以证明, TG^L 上的 S_3 变换是一种同构变换, 而且 TG^L 对这种变换具有对称性. 这个同构关系保证了所关联的两个路径 (例如上文 P 和 P') 的长度相同, 变换对称性保证了两个长度的计算算法相同.

2.3.2 最短路径路由分析

在给出 SPR4T 前, 首先分析图 3 所示的路由特例, 图中当前路由节点 S 和通信目标节点 T 分别位于 MSG_1^l 和 MSG_2^l 中, 这里 $l\leqslant L$ 是此时 FRG^l 的规模.

显然, 从 S 出发到达 T 的最短路径必为路径 P_{A} 和 P_{B} 之一, 当路径长度差 $d(P_{\mathrm{A}})-d(P_{\mathrm{B}})\leqslant 0$ 选择路径 P_{A}, 否则选择 P_{B} 即可.

由于这两条路径都是由多个片段构成, 因此有

$$d(P_{\mathrm{A}})-d(P_{\mathrm{B}})=(d_{\mathrm{SA}}+1+d_{\mathrm{A'T}})-(d_{\mathrm{SB}}+d_{\mathrm{BB'}}+d_{\mathrm{B'T}}).$$

因此, 上式中 d_{SA} 和 d_{SB} 是 S 到 MSG_1^l 尖端 3^{l-1} 和 2^{l-1} 的距离, $d_{\mathrm{A'T}}$ 和 $d_{\mathrm{B'T}}$ 是 T 到 MSG_2^l 尖端 1^{l-1} 和 3^{l-1} 的距离, 而 $d_{\mathrm{BB'}}$ 为 MSG_3^l 边长 + 2.

2.3.3 节点到尖端距离

前文对于最短路径的路由分析表明, 路由的核心计算是顶点到尖端的距离, 如图 4 所示. 关于该距离有下述定理.

定理1 $\forall\bar{x}\in V(\mathrm{TG}^L)$, 如果其二进制名为 $x''_Lx'_L\cdots x''_1x'_1$, 则其到尖端 1^L, 2^L 和 3^L 的距离分别为二进制数字 $x''_L\cdots x''_1$, $x'_L\cdots x'_1$ 和 $(\overline{x''_L}\cdots\overline{x''_1})+(\overline{x'_L}\cdots\overline{x'_1})$, 其中 $+$ 表示逻辑位或运算.

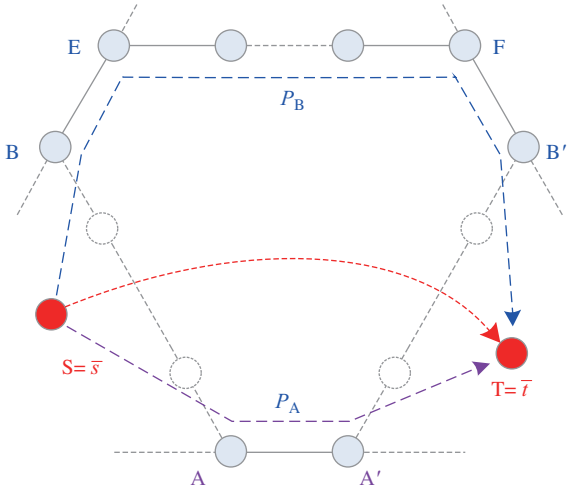

图 3 (网络版彩图) FM_0 模式的最短路径路由方法

Figure 3 (Color online) The shortest path routing approach in FM_0 mode

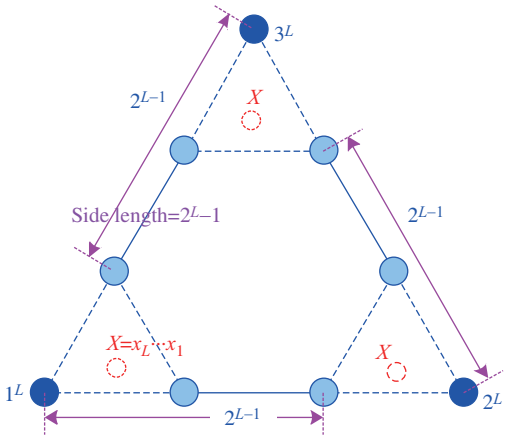

图 4 (网络版彩图) 顶点到尖端距离求值

Figure 4 (Color online) Distance evaluation from vertex to tip

证明 为简化证明令 $\mathrm{FRG}^L=\mathrm{TG}^L$ (图 3, 它由 $\mathrm{MSG}_i^L\,(i\in\mathsf{T})$ 3 个 TG^{L-1} 构成, TG^{L-1} 边长为 $2^{L-1}-1$. TG^L 顶点 $\bar{x}=x_L\cdots x_1$ 到尖端 i^L 的距离 $d\,(x_L\cdots x_1,i)$ 必属下述 3 种情况之一.

(1) $x_L=i$, 此时 $\bar{x}$ 同属 TG^L 和 TG^{L-1}(即 MSG_i^L), 但名字分别为 $x_L\cdots x_1$ 和 $x_{L-1}\cdots x_1$. 注意到此时 TG^L 的尖端 i^l 与 TG^{L-1} 的尖端 i^{l-1} 重叠, 因此 $\bar{x}$ 到 i^l 的距离满足 $d\,(x_L\cdots x_1,i)=d\,(x_{L-1}\cdots x_1,i)$.

(2) $x_L\neq i$, 顶点 $\bar{x}$ 位于 MSG_j^L 中且 $j\neq i$, 例如图 3 位于 MSG_1^L (左下三角区) 中顶点 $\bar{x}$ 到 3^L 的情形. 此时顶点 $\bar{x}$ 到 i^{L-1} 距离为 $d\,(x_{L-1}\cdots x_1,i)$, 而 i^{L-1} 到 i^L 的距离是 2^{L-1}, 因此 $\bar{x}$ 到 i^l 的距离满足 $d\,(x_L\cdots x_1,i)=2^{L-1}+d\,(x_{L-1}\cdots x_1,i)$.

(3) $d\,(\varepsilon,i)=0$, 这里 ε 是空文字串. 于是, 顶点到尖端的距离满足如下递推公式:

$$d\,(x_L\cdots x_1,i)=\begin{cases}d(x_L\cdots x_1), & x_L=i,\\ 2^{L-1}+d(x_{L-1}\cdots x_1,i), & x_L\neq i.\end{cases}$$

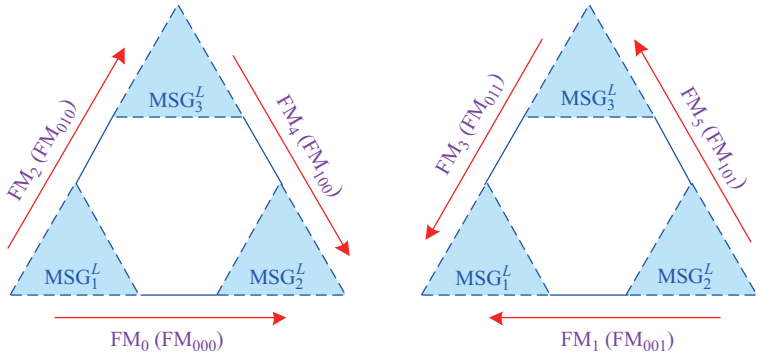

图 5 (网络版彩图) TriBA-Net 通信数据的宏观流动模式

Figure 5 (Color online) Flow modes of communication data in TriBA-Net

该式可展开为

$$d(x_L\cdots x_1,i)=\sum_{n=1}^{L}(x_n\neq i)\times 2^{n-1},$$

其中 $x_n\neq i$ 为关系运算表达式, 其值只能为 0 或者 1, 显然该式右侧恰好是某个十进制数字的二进制求值公式.

下面以顶点 $x_L\cdots x_1$ 到 1^L 的距离的求值为例 (其他情况类似) 来证明定理 1. 此时, 如果使用二进制节点名则上述关系表达式变形为 $x_n''x_n'\neq 01$, 根据该式真值表可得该式的值为 x_n'', 于是有

$$d(x_L\cdots x_1,i)=\sum_{n=1}^{L}x_n''\times 2^{n-1},$$

该式右侧恰好是二进制数字 $x_L''\cdots x_1''$ 的十进制求值表达式, 于是定理得证.

2.3.4 流模式

根据当前路由节点和目标节点名, 依照定理 1 的方法即可完成 2.3.2 小节路由计算, 但该路由只适合两节点位于所述区域的特定情况. 实际通信中两节点位置关系存在多种情况, 虽然可根据 2.3.2 小节的原理为每种情况设计专用算法, 但这样做硬件实现效率较低, 算法的逻辑较乱.

由于 TG^L 对于 S_3 群所含变换具有同构和对称的性质, 图 2(a) 中路径 P 的端点不符合 2.3.2 小节算法要求, 因此不能用该算法求其长度, 但是如果对图 2(a) 实施变换 s_5 后 P 被转换成图 2(b) 路径 P', 它满足 2.3.2 小节算法对端点位置关系的要求, 由于 TG^L 对 S_3 元素的变换具有同构性和不变性保证了两条路径长度相同以及在图 2(b) 中 2.3.2 小节算法仍有效.

也就是说, 对于不满足 2.3.2 小节端点要求的通信, 可用 S_3 元素的变换将之转换为 2.3.2 小节通信等价的状态进行路由计算, 所得结果通过相应逆变换后回到原本通信状态即可等价地得到所需路由计算结果.

为此, 分析通信路径端点的可能状态, 通信被划分为 6 种如图 5 所示的宏观数据流动模式 $FM_0\sim FM_5$. 显然, 2.3.2 小节通信的流模式为 FM_0, 而且通过简单地验证即可得知对于处于 FM_i 模式的通信, 只要对其 FRG 实施变换 $s_i\in S_3$, 即可将之变换为等价的 FM_0 模式, 进而在该模式下进行相关距离计算.

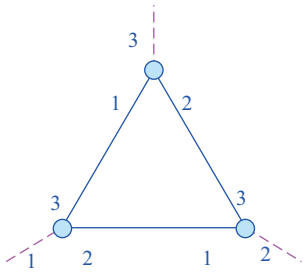

图 6 (网络版彩图) TriBA-Net 节点端口命名

Figure 6 (Color online) Port naming of nodes in TriBA-Net

3 TriBA-Net 的路由算法

根据前文所述原理, TriBA-Net 的最短路径路由算法 SPR4T (shortest-path routing algorithm for TriBA) 被设计为两个层次, 其核心是 FM_0 的路由算法 SPR0, 其他模式路由计算通过 S_3 转换到 FM_0 后调用 SPR0 得到等价路由结果, 然后通过相应逆变换将之转换为原通信模式的路由计算结果.

3.1 关于数据转发方向

考虑到 $\forall \bar{x}_{L..1} \in V(\mathrm{TG}^L)$ 其名必然形如 $\bar{x}_{L..2}a$, 定义 1 表明它应该有两个邻接顶点: $\bar{x}_{L..2}b$ 和 $\bar{x}_{L..2}c$, 其中 $a \neq b \neq c$; 实际上 $\bar{x}_{L..2}a$ 还可写为 $\bar{x}_{L..l+2}xa^l$ 的形式, 其中 $x \neq a$ 且 $l \geqslant 1$, 定义 1 表明该顶点拥有邻接顶点 $\bar{x}_{L..l+2}ax^l$.

上述规律提示我们, 在物理实现 TG^L 时, 可将对应节点连接 $\bar{x}_{L..2}b$ 和 $\bar{x}_{L..2}c$ 的端口编码为 b 和 c, 而将连接 $\bar{x}_{L..l+2}xa^l$ 的端口编码为 a, 图论中以有向边标识的路由方向在节点内部等价地标记为数据转发的目标端口, 只要数据转发到某端口即可实现其所连通道 (有向边) 的通信 (路由).

根据上述规则画出位于某个基本组的 3 个节点端口的编码情况如图 6 所示.

3.2 FM_0 模式最短路径路由

根据前文的分析和定理 1, 很容易得到 FM_0 模式的最短路径路由算法 SPR0 (算法 1).

算法 1 SPR0

Input: $s''_L s'_L \cdots s''_1 s'_1$, starting point identifier;
$t''_L t'_L \cdots t''_1 t'_1$, ending point identifier;
l, size of FRG^l;

Output: Direction of forwarding at $s''_L s'_L \cdots s''_1 s'_1$;

LenPA $\leftarrow s'_{l-1} \cdots s'_1 + 1 + t''_{l-1} \cdots t''_1$;
LenPB $\leftarrow (\bar{s}''_{l-1} \cdots \bar{s}''_1 + \bar{s}'_{l-1} \cdots \bar{s}'_1) + (\bar{t}''_{l-1} \cdots \bar{t}''_1 + \bar{t}'_{l-1} \cdots \bar{t}'_1) + 2^{l-1} + 1$;
if LenPA $\leqslant$ LenPB **then**
    return 2; //Port2;
else
    return 3; //Port3;
end if

3.3 任意模式最短路径路由

算法 2 所调用的 FM($s''_l s'_l \cdots s''_1 s'_1, t''_l t'_l \cdots t''_1 t'_1$) 是根据路径端点计算流模式的算法 (较简单, 此处从略). 关于算法 1 和 2 中的转发方向请见 3.1 小节.

算法 2 SPR4T

Input: $s''_L s'_L \cdots s''_1 s'_1$, starting point identifier;
$t''_L t'_L \cdots t''_1 t'_1$, ending point identifier;
l, size of TriBA-Net;
Output: Direction of forwarding at $s''_L s'_L \cdots s''_1 s'_1$;

```
if l = 0 then
    return 0; //Port0;
else
    for l = L → 1 do
        if s''_l s'_l ≠ t''_l t'_l then
            break;
        end if
    end for
    i ← FM(s''_l s'_l ··· s''_1 s'_1, t''_l t'_l ··· t''_1 t'_1);
    ū_EQ ← Convert_i(s''_l s'_l ··· s''_1 s'_1);
    v̄_EQ ← Convert_i(t''_l t'_l ··· t''_1 t'_1);
    return s_i^{-1}(SPR0(ū_EQ, v̄_EQ, l));
end if
```

4 路由实现原理与关键环节

4.1 路由计算流水段简介

考虑到算法 SPR0 和 SPR4T 的宏观计算流程, 本文以四级流水对其进行实现, 各流水段主要功能如下:

(1) PREP, 数据准备阶段. 包括计算当前流模式、网络规模、FRG^l 规模以及后续运算需要的屏蔽位.

(2) EDIS, 等价距离计算阶段. 对现模式当前节点名 CID 和目标节点名 TID 进行 S_3 变换得 FM_0 模式等价名 ECID 和 ETID; 然后计算 ECID 到节点 MSG_1^l 尖端 3^{l-1}, 2^{l-1} 的距离, ETID 到 MSG_2^l 尖端 3^{l-1}, 1^{l-1} 的距离; 该段还为后续计算提供相应的数据按位取反等操作.

(3) PCMP, 备选路径比较阶段. 该段进行图 3 中路径 P_A 和 P_B 的比较. 该段操作耗时较长, 是整个流水的瓶颈.

(4) FWDP, 转发端口输出及特殊处理阶段. 该段根据 PCMP 输出判决转发目标端口编号.

另外, 上述流水段中还包括的虚通道即通信安全等计算, 因与本文无关, 相关介绍从略.

4.2 路由计算关键环节

上述流水过程的整个设计规模较大, 因篇幅限制, 这里仅对所涉及的重要环节进行简单介绍.

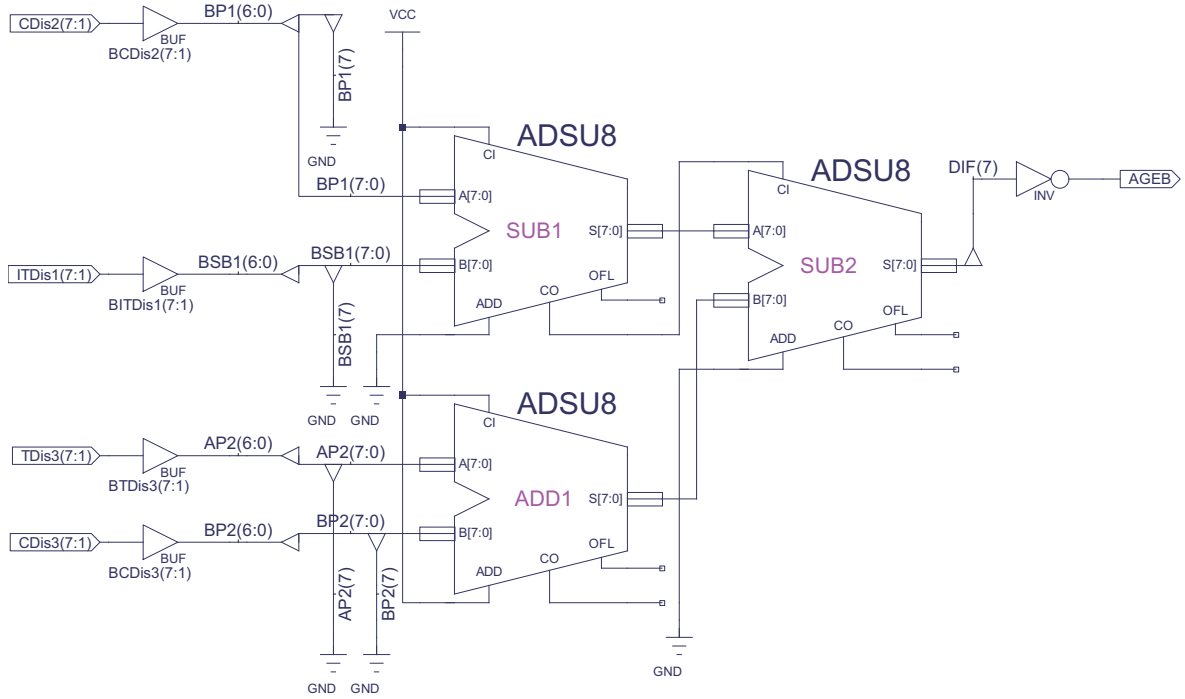

图 7 (网络版彩图) 优化后的路径比较电路

Figure 7 (Color online) Optimized route comparison circuit

4.2.1 关于路径比较

2.3.2 小节给出的路径长度比较计算公式为三级加/减法运算, 实现过程中可优化为两级加/减法:

$$\begin{aligned} d(P_{\mathrm{A}}) - d(P_{\mathrm{B}}) &= (d_{\mathrm{SA}} + 1 + d_{\mathrm{A'T}}) - (d_{\mathrm{SB}} + d_{\mathrm{BB'}} + d_{\mathrm{B'T}}) \\ &= (d_{\mathrm{SA}} + 1 + d_{\mathrm{A'T}}) - (d_{\mathrm{SB}} + d_{\mathrm{EF}} + 2 + d_{\mathrm{B'T}}) \\ &= (d_{\mathrm{SA}} + d_{\mathrm{A'T}}) - d_{\mathrm{EF}} - (d_{\mathrm{SB}} + 1 + d_{\mathrm{B'T}}) \\ &= d_{\mathrm{SA}} - (d_{\mathrm{EF}} - d_{\mathrm{A'T}}) - (d_{\mathrm{SB}} + d_{\mathrm{B'T}} + 1) \\ &= d_{\mathrm{SA}} - \bar{d}_{\mathrm{A'T}} - (d_{\mathrm{SB}} + d_{\mathrm{B'T}} + 1), \end{aligned}$$

上式结果中 +1 在实现时可连接到加法器的进位端, 而 $d_{\mathrm{EF}} - d_{\mathrm{A'T}}$ 恰好为 $d_{\mathrm{A'T}}$ 的按位取反 $\bar{d}_{\mathrm{A'T}}$(其有效位在相应 FRG 的 MSG 范围内, 高位清零即可), 因此上述运算实际为两级加法运算, 如图 7 所示.

图 7 中, 加法器 SUB1:ADSU8 实现了上式中的减法运算 $d_{\mathrm{SA}} - \bar{d}_{\mathrm{A'T}}$. 当 SUB1:ADSU8 设置为减法器 (ADD=0), 此时 CI=1 表明无借位; ADD1:ADSU8 实现了上述公式中的加法运算 $d_{\mathrm{SB}} + d_{\mathrm{B'T}} + 1$, 当 ADD1:ADSU8 设置为加法器 (ADD=1), 此时 CI=1 表明有进位, 即公式中的 +1; SUB2:ADSU8 实现了上述两个中间项的减法.

4.2.2 关于 S_3 变换

节点名的 S_3 变换被分解为对其每位层码 (二进制形式) 的变换. 如 2.3.4 小节所述, 当前模式 FM_i 下层码所进行的变换为 S_3 的元素 s_i, 相关变换计算真值表如表 1 所示.

表 1 层码等价变换真值表

Table 1 Truth table of equivalent transformation of layer codes

Current mode (uvw)	000	001	010	011	100	101
Current code (ab)	Transformed code ($x''x'$)					
01	01	10	01	10	11	11
10	10	01	11	11	10	01
11	11	11	10	01	01	10

表 2 模拟器相关参数配置

Table 2 Configuration of the parameters of the simulator

Configuration	Topology	Network size	Switching	Flit size	Buffer	Packet size	Virtual channel
Parameter	TriBA-Net	27 nodes	Wormhole	32 bits	4 flits	4 flits	4

根据表 1 可得变换下述逻辑表达式:

$$x'' = \overline{\bar{a}b\bar{u}\bar{w} + a\bar{b}\bar{v}w + ab\bar{u}vw + abu\bar{v}\bar{w}},$$

$$x' = \overline{\bar{a}b\bar{u}w + a\bar{b}\bar{v}\bar{w} + ab\bar{u}v\bar{w} + abu\bar{v}w}.$$

这表明, 仅用组合电路即可实现 S_3 变换.

4.2.3 关于有效位

由于路由算法的相关约束, 最短路径总是被包含在某个规模为 $l-1$ 的焦路子图内部, 因此节点名中的第 $l-1$ 位以及低于该位的层码才与最短路径路由计算相关, 属于有效位. 又由于所涉及的距离计算公式均为位逻辑操作, 不同位操作之间无关, 因此在实现具体计算时仅需对所有的数位按位进行操作, 最后将结果中的高于 $l-1$ 位的层码全部置 0.

5 设计验证与仿真评测

5.1 实验环境及配置

在性能分析方面, 本文采用片上网络模拟器 Noxim [18] 搭建仿真实验平台. Noxim 模拟器采用 SystemC 语言进行描述, 具有良好的可扩展性以及时钟模拟精度, 可以详细地模拟片上网络中主要功能部件的行为. 为了使 Noxim 能够模拟 TriBA-Net, 从拓扑结构、网络大小、路由算法等方面对 Noxim 进行修改, 搭建了 27 个节点的 TriBA-Net 拓扑结构. 模拟器的相关参数设置如表 2 所示, 拓扑结构采用 27 个节点的 TriBA 结构, 交换机制采用虫孔交换机制, 仿真环境中数据的最小单元是 flit, 每个数据包由 4 个 flit 构成. 每个输入信道的缓存大小为 4 个 flit, 支持 4 条虚拟通道.

5.2 延迟和吞吐量

传输延迟和吞吐率是评价片上网络性能的重要指标. 本文实验主要针对 SPR4T 和 SPORT 两种算法, 选取 Uniform, Bitreversal, Shuffle 3 种流量模式, 在各种注入率与流量模式下对延迟和吞吐率进

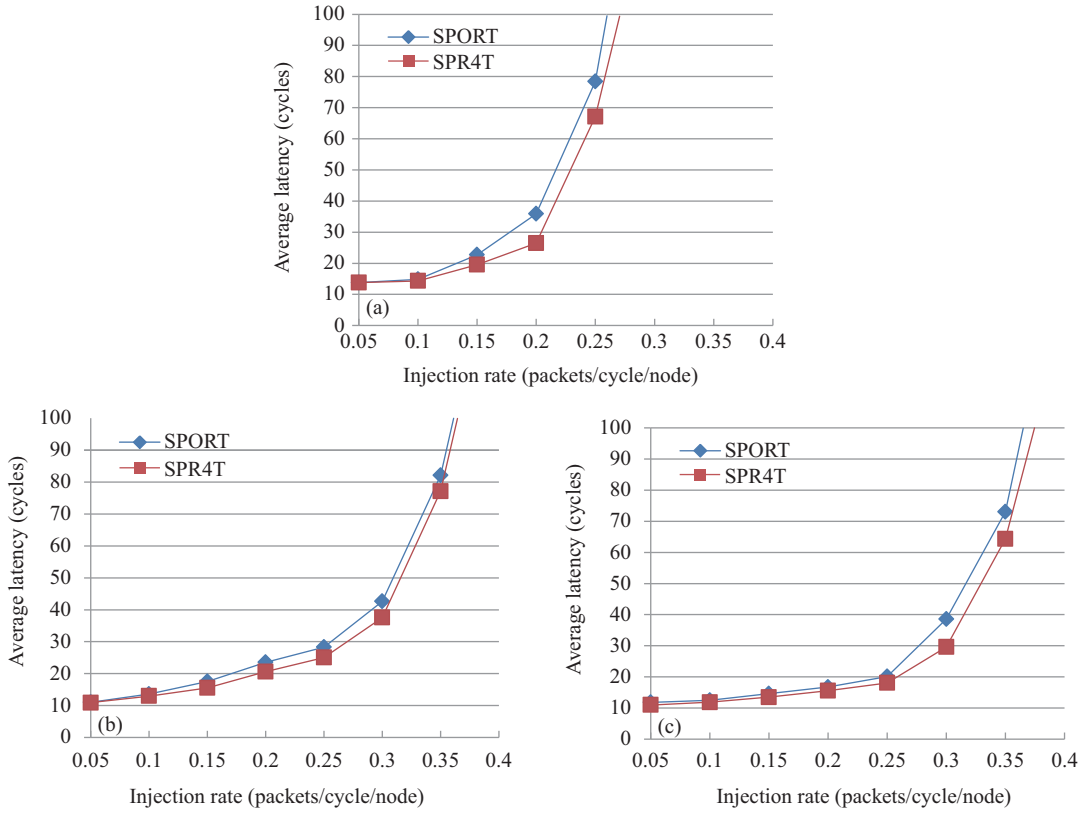

图 8　(网络版彩图) 平均延迟随注入率变化曲线图

Figure 8　(Color online) Curve of average latency versus injection rate. (a) Uniform; (b) Bitreversal; (c) Shuffle

行测试. 每次模拟 1×10^6 个周期, 为了获取稳定的网络性能, 起初 1×10^5 个周期属于预热阶段, 随后的 9×10^5 个周期用于性能统计.

图 8(a) ~ (c) 所示为不同流量模式下平均延迟随注入率变化曲线图, 性能判别选择饱和注入率作为基准. 饱和注入率定义为平均延迟 3 倍于零负载延迟时的包注入率大小, 其可以间接反映出网络吞吐饱和的趋势 [19,20]. 从图 8 中可以看出, 在网络流量较低的情况下, SPR4T 和 SPORT 两种算法的性能区别不大. 随着流量的不断注入, 平均传输延迟逐渐增大. 不论在哪种流量模式和路由算法下, 当注入率增长到一定时, 网络流量都会达到饱和状态, 网络时延会呈现急剧增长趋势, 引起性能严重下降. 在 Uniform, Bitreversal, Shuffle 3 种流量模式下, 当网络延迟为 35 cycles 时, 相比于 SPORT 算法, SPR4T 算法的饱和注入率分别提升 7.5%, 7.2%, 7.6%. 从仿真结果可以看出, SPR4T 算法的平均延迟性能优于 SPORT 算法.

图 9(a) ~ (c) 所示为不同流量模式下吞吐率与注入率的关系曲线. 结果表明, 当网络流量较低时, 网络未发生拥塞, SPR4T 和 SPORT 两种算法在相同的注入率下单位时间内收到的数据包数量相同, 并且随着注入率的增加呈线性增长. 然而, 当注入率不断增加时, 网络流量逐渐加大, 由于路由资源的限制, 当注入率达到一定值时, 导致单位时间内收到的数据包数量最终趋于一个饱和值, 即饱和吞吐率. 在 Uniform, Bitreversal, Shuffle 3 种流量模式下, 相比于 SPORT 算法, SPR4T 算法的饱和吞吐率分别提升 7.7%, 6.9%, 7.4%. 可以看出, SPR4T 算法在饱和吞吐率的性能指标上优于 SPORT 算法.

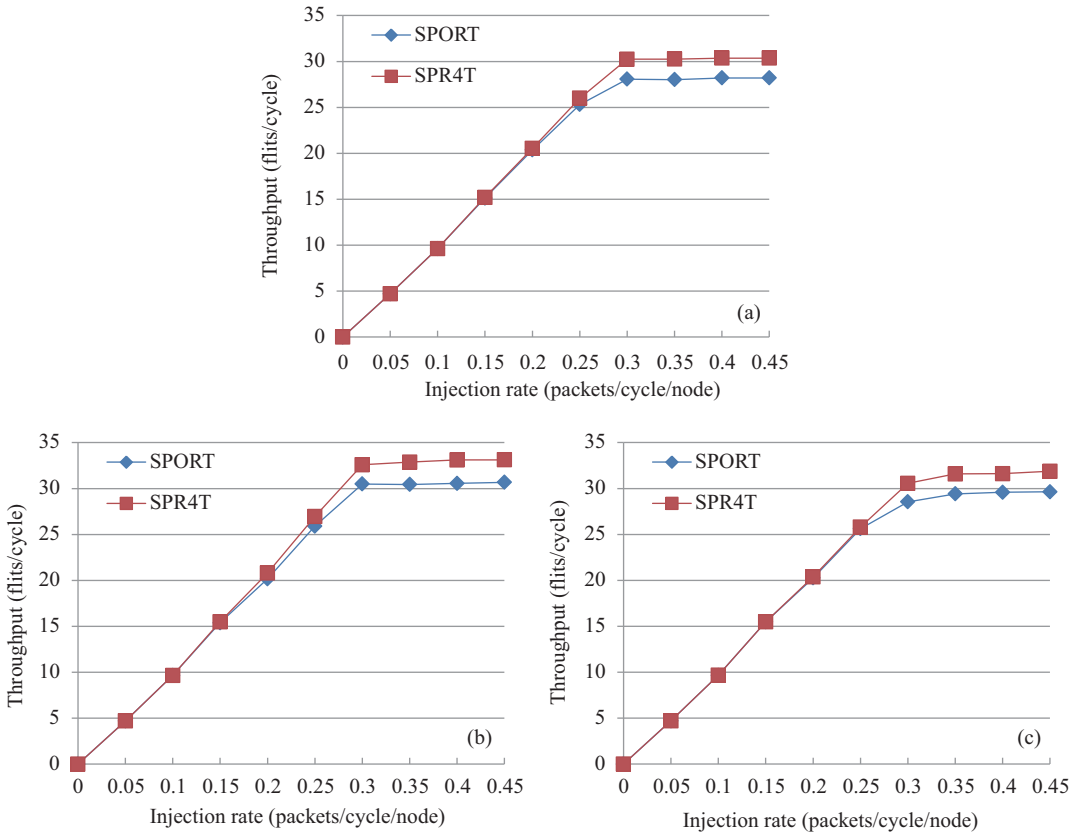

图 9 (网络版彩图) 吞吐率随注入率变化曲线图

Figure 9 (Color online) Curve of throughput versus injection rate. (a) Uniform; (b) Bitreversal; (c) Shuffle

表 3 路由器硬件开销情况表

Table 3 Router hardware overhead list

	Slices	LUTs	Flips-Flops
SPORT router	139	269	114
SPR4T router	118	229	103

5.3 硬件开销和功耗分析

本节主要评估 SPR4T 算法的硬件实现成本和功耗, 仍然以 SPORT 算法为参考. 本文使用 VHDL 硬件描述语言对以上两种基于不同路由算法的路由器模型进行建模, 采用 Xilinx ISE 13.4 工具, 在 Xilinx 公司 Virtex6 系列的 XC6VLX550TL 芯片上进行了设计实现, 综合结果如表 3 所示.

综合结果显示, 与 SPORT 路由器相比, SPR4T 路由器的 Slices, LUTs 和 Flips-Flops 分别减少了 15.1%, 14.9%, 9.7%. 这主要是因为 SPR4T 算法利用 TriBA-Net 的拓扑特征简化了路由计算, 控制逻辑更为简单, 仲裁器、多路选择器数量更少. 因此, SPR4T 算法导致路由器硬件成本开销更小, 是一种性价比较高的算法.

最后, 本文利用 Prime Power 门级功耗分析工具 [21] 对两种不同路由算法的路由器进行功耗评估. 针对配置两种路由器的 27 个节点规模的 TriBA-Net 网络分别展开逻辑综合并利用相同激励进行前端

表 4 SPR4T 和 SPORT 路由器的功耗比较

Table 4 Comparison of power consumption between SPR4T and SPORT routers

	Uniform (nW)	Bitreversal (nW)
SPORT router	32.28	30.42
SPR4T router	29.54	27.78

仿真, 将得到的门级电平翻转模型、工艺库文件以及网表一起输入 Prime Power 工具来评估整个网络的平均功耗. 如表 4 所示, 在两种模型下 SPR4T 路由器的平均功耗分别降低了约 8.5% 和 8.7%.

6 结论

本文针对 TriBA-Net 网络的拓扑特征, 提出了一种相隔节点间的通信模型, 根据两节点的位置关系, 将通信划分为 6 种数据宏观流动模式. 再利用 S_3 群的循环置换特性对通信模型进行简化, 进而设计出一种基于 S_3 变换的最短路径路由机制, 并且完成了路由器的硬件实现, 给出了性能分析结果. 实验结果表明, 在 27 个节点的 TriBA-Net 网络性能测试中, SPR4T 算法的性能优于 SPORT 算法, 能够降低网络的平均端到端时延, 提高网络吞吐量, 而且硬件以及功耗的开销更小. 后续工作中, 我们将研究基于 Hamilton 路径的路由防死锁机制.

参考文献

1 Borkar S, Chien A A. The future of microprocessors. Commun ACM, 2011, 54: 67–77

2 Wulf W A, McKee S A. Hitting the memory wall: implications of the obvious. Comput Archit New, 1995, 23: 20–24

3 Das R, Mutlu O, Moscibroda T, et al. Aergia: a network-on-chip exploiting packet latency slack. IEEE Micro, 2011, 31: 29–41

4 Bhattacharjee A, Contreras G, Martonosi M. Parallelization libraries: characterizing and reducing overheads. ACM Trans Archit Code Opt, 2011, 8: 1–29

5 Shi F, Ji W X, Qiao B J, et al. A triplet-based computer architecture supporting parallel object computing. In: Proceedings of IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP), Montreal, 2007. 192–197

6 Vecchia G D, Sanges C. A recursively scalable network VLSI implementation. Future Gener Comput Syst, 1988, 4: 235–243

7 Hu S S, Shi F, Ji W X, et al. Exploring grouped coherence for clustered hierarchical cache. J Supercomput, 2017, 73: 4137–4157

8 Sterling T L, Zima H P. Gilgamesh: a multithreaded processor-in-memory architecture for petaflops computing. In: Proceedings of ACM/IEEE Conference on Supercomputing, Baltimore, 2002. 1–23

9 Mubeen S, Kumar S. Designing efficient source routing for mesh topology network on chip platforms. In: Proceedings of the 13th Euromicro Conference on Digital System Design: Architectures, Methods and Tools, Lille, 2010. 181–188

10 Xiang D, Luo W. An efficient adaptive deadlock-free routing algorithm for torus networks. IEEE Trans Parall Distrib Syst, 2012, 23: 800–808

11 Xiang D, Zhang Y, Shan S, et al. A fault-tolerant routing algorithm design for on-chip optical networks. In: Proceedings of the 32nd International Symposium on Reliable Distributed Systems, Braga, 2013. 1–9

12 Hu J, Marculescu R. Energy-aware mapping for tile-based NoC architectures under performance constraints. In: Proceedings of Asia and South Pacific Design Automation Conference, Kitakyushu, 2003. 233–239

13 Marvasti M B, Daneshtalab M, Afzali-Kusha A, et al. PAMPR: power-aware and minimum path routing algorithm for NoCs. In: Proceedings of the 15th IEEE International Conference on Electronics, Circuits and Systems, Kitakyushu, 2008. 418–421

14 Hu J, Marculescu R. Exploiting the routing flexibility for energy/performance-aware mapping of regular NoC architectures. In: Proceedings of Europe Conference and Exhibition on Design, Automation and Test, Munich, 2003. 688–693
15 Qiao B j, Shi F, Ji W X. A new hierarchical interconnection network for multi-core processor. In: Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications, Harbin, 2007. 246–250
16 Wang Z, Shi F. A shortest path routing algorithm in triplet-based network. Trans Beijing Inst Technol, 2009, 29: 410–414
17 Zhang Y, Shi F. Design and evaluation of low-latency and shortest-path routing algorithm for triplet-based hierarchical interconnection network. J Test Eval, 2013, 41: 541–550
18 Catania V, Mineo A, Monteleone S, et al. Cycle-accurate network on chip simulation with noxim. ACM Trans Model Comput Simul, 2016, 27: 1–25
19 Weerasinghe H D, Tackett R, Fu H R. Verifying position and velocity for vehicular ad-hoc networks. Secur Commun Netw, 2011, 4: 785–791
20 Wu L F, Meng Q H, Liang H W, et al. Accurate localization in combination with wireless sensor networks and laser localization. In: Proceedings of IEEE International Conference on Automation and Logistics, Shenyang, 2009. 146–151
21 Synopsys Inc. Data sheet: primePower full-chip dynamic power analysis for multimillion-gate design. 2004. https://www.synopsys.com/

A shortest path routing mechanism based on S_3 for TriBA-Net

Feng SHI*†, Xu CHEN†, Fei YIN, Xiaojun WANG, Sensen HU, Weixing JI, Yizhuo WANG, Yujin GAO & Jin WEI

School of Computer, Beijing Institute of Technology, Beijing 100081*, China*
* Corresponding author. E-mail: bitsf@bit.edu.cn
† Equal contribution

Abstract The routing algorithm of a Network-on-Chip (NoC) is essential to its performance and power consumption. This paper presents a novel shortest path routing algorithm for TriBA-Net. First, the algorithm designs a coding scheme based on the topological features of TriBA-Net. The set of words 1, 3, and 2, used in the coding scheme, has the same meaning as the well-known group S_3 on 3-letters. Second, a communication model, which contains 6 types of flow modes, has been proposed for reflecting the status of the path within two hops. Finally, the algorithm is simplified by the cyclic permutation characteristic of the S_3 group. What's more, the implementation of the SPR4T router is completed under the XC6VLX550TL chip. Experimental results show that under the uniform traffic pattern in the 27-node TriBA-Net performance test, SPR4T routing algorithm has a 7.5% higher saturation injection rate and a 7.7% higher throughput rate, with the obvious savings of hardware overhead and lower power consumption when compared to the SPORT routing algorithm.

Keywords networkon-chip, routing algorithm, coding scheme, topology, performance evaluation

Feng SHI was born in 1961. He received the Ph.D. degree from the Beijing Institute of Technology, Beijing, in 1999. Currently, he is a professor at the Beijing Institute of Technology. His research interests include computer architecture, ASIC design and processor technology.

Fei YIN was born in 1979. He received the Master's degree from the Beijing University of Posts and Telecommunications, Beijing, in 2006. Currently, he is a Ph.D. candidate at the Beijing Institute of Technology. His research interests include computer architecture and parallel computing.

Xu CHEN was born in 1983. He received the Master's degree from the Beijing Institute of Technology, Beijing, in 2010. Currently, he is a Ph.D. candidate at the Beijing Institute of Technology. His research interests include computer architecture, embedded systems, and parallel computing.

Xiaojun WANG was born in 1979. He received the Master of Engineering degree from Harbin Engineering University, Harbin, in 2007. Currently, he is a Ph.D. candidate at the Beijing Institute of Technology. His research interests include computer architecture and embedded systems.