

一种新型 NoC 互连算法及连线冗余的修正*

刘 一^{1,2},段成华¹

(1. 中国科学院研究生院 信息科学与工程学院,北京 100049; 2. 中国民航管理干部学院,北京 100102)

摘 要:片上网络(NoC)是一种新兴的以包交换为通信方式的芯片互连结构。NoC 的互连问题可以抽象为在有障碍曼哈顿平面生成最小森林的图论问题,本文提出了一种新型的 NoC 互连算法,该算法通过对连线边长权重进行更改后再调用最小生成树算法,并针对连线冗余进行修正。实验表明,该算法使得片上网络的全局连线长度最小,从而解决传统片上总线结构中连线延时长、信号完整性差等缺点。

关键词:Steiner 树;片上网络;连线冗余

中图分类号:TP331.1 **文献标识码:**A

A New Algorithm for Connection in NoC and Redundancy Correction

LIU Yi^{1,2},DUAN Cheng-hua¹

(1. School of Information Science and Engineering, Graduate Univ. of Chinese Academy of Sciences, Beijing 100049, China;

2. Civil Aviation Management Institute of China, Beijing 100102, China)

Abstract:Network on Chip (NoC) adopts the packages exchanging interconnection framework for communications on chip. The interconnection problem of NoC can be abstracted as a minimal spanning forest problem in Manhattan plane. A new algorithm based on obstacle avoiding rectilinear Steiner minimal tree by changing the edge weights is proposed for the connection, an optimization method for interconnection redundancies removing is also proposed. Experiments show that the total connection length for NoC is minimized. So the traditional bus communication framework's disadvantages, such as long global connect delay, weak signal integrity may be eliminated.

Key words:steiner tree; Network on Chip (NoC); connection redundancy

芯片集成度的提高导致片上计算资源快速增加,片上各模块间通信对片上系统的性能影响也随之增大。如何有效地连接片上资源,包括处理器、控制器、存储阵列等,是影响片上系统性能的关键因素。随着当今集成电路发展从微电子学进入纳米集成电路时代(nanoscale integrated circuits),片上网络的概念被引入^[1]。如果按拓扑结构分类,NoC 大体可以分为规整的网络(regular network)和定制的网络(custom network),其中规整的网络是先把互连的拓扑结构固定,再把单元映射到结构。而定制的网络是最近才提出的概念^[2],它根据芯片单元的通信任务生成特定的拓扑结构,即最终的结构和芯片单元的通信任务相关联,这样实现性能的优化。如图 1 分别是定制的排列和规整 Mesh 的排列^[2]。定制的 NoC 的互连问题可以抽象成为一个在有障碍曼哈顿(Manhattan)平面的生成最小森林的图论问题^[3],考虑到在集成电路版图设计布局、布线过程中的已有算法与之类似,特别是基于斯坦纳树(Steiner tree)的算法更为接近。这因为基于斯坦纳树的算法需要在互连节点之外再加入新的节点,而这新的节点的地位恰与 NoC 互连中的路由器(router)相类似。因此在我们的研究中,通过改进已有的基于斯坦纳树的互连算法,把它用于定制的片上网络,得到最小的全局连线长度。

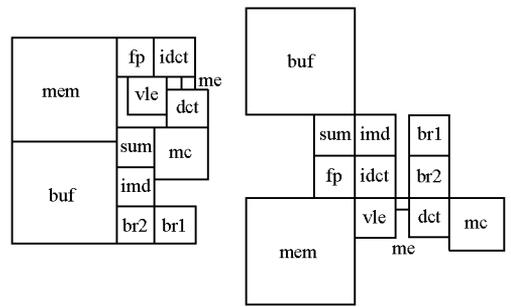


图 1 片上网络的规整结构和定制结构
Fig. 1 Regular NoC and custom NoC

* 收稿日期:2009 - 09 - 01
基金项目:国家 863 高技术资助项目
作者简介:刘一(1981—),男,博士。

1 Steiner 树理论背景

定义 平面上的 Steiner 最小树给定二维平面上的点集 P (其中的点称为原点或正则点), 要求设法添加新点集 S (其中的点称为 Steiner 点, 简称 s -点), 来构造平面上集合 $P \cup S$ 的一棵最小生成树 T , 使其总长尽可能小。设 $p_1 = (x_1, y_1), p_2 = (x_2, y_2)$ 为集合 P 中任意两点, 则 p_1 与 p_2 间的距离可表示为 $p_1 p_2 = (|x_1 - x_2|^d + |y_1 - y_2|^d)^{1/d} (d=1, 2)$ 。根据节点间连接方式的不同, 可以分为两种: 即当 $d=1$ 时, 节点间的距离为欧氏距离, 此时的问题称为欧氏 Steiner 最小树 (Euclidean Steiner Minimum Tree — ESMT) 问题; 而当 $d=2$ 时, 结点间的距离为绝对值距离 (也称 Manhattan 距离, 结点间只能由直线段或折线段相连), 此时称为绝对值距离 Steiner 最小树 (Rectilinear Steiner Minimum Tree — RSMT) 问题^[4]。下面是一些常用的重要性质:

性质 1 设平面上原点数为 n 个, 则 ESMT 和 RSMT 均满足, s -点数 $n - 2$ 。

性质 2 (Hanan 定理^[5]) 任意原点集 P 的绝对值距离 Steiner 最小树 T 的 s -点均在 P 的 Hanan 网格 G_p 的格点 (即经过 P 中所有结点的水平线与垂直线的交点) 上, 生成的图也称为 Escape Graph (见图 2)。

已有的解决 RSMT 问题的算法主要有: Maze-routing (迷宫) 算法、线探索算法、遗传进化算法 (包括蚁群优化算法等)、模拟退火算法及一类基于构造最小生成树的算法。RSMT 问题是 NPC 问题, 精确求解是困难的。1992 年, 关于 Steiner 树的 GP 猜想获得证明: 把 Steiner 树的边长和与最小生成树的边

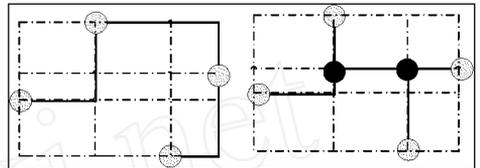


图 2 Hanan 定理
Fig. 2 Hanan theorem

长和的比记为 Steiner 比, 则 $\frac{\sqrt{3}}{2}$ 。这说明, 虽然 Steiner 问

题的精确求解非常难做到, 但其得到的结果并不比最小生成树好很多。这也为使用构造最小生成树的方法取得最小 Steiner 树提供了理论基础。如文献 [6] 就是按以下步骤生成 RSMT: 先生成 MST, 再结合障碍物四角生成 Spanning graph, 在 Spanning graph 上绕线生成 RSMT, 最后修正冗余的连线。同理文献 [7] 给出的算法也是先考虑生成最小生成树, 再将其直角化, 并且连线冗余的去除大多放在算法的最后步骤。考虑到算法中各步骤的执行顺序对算法结果的影响, 我们提出一种新型的基于构造最小生成树的 RSMT 算法, 并对连线中形成的冗余进行分类修正, 以使连线长度最短。

2 算法实现与连线冗余的修正

对于 $M(x_m, y_m), N(x_n, y_n)$ 两点, 若满足 $x_{left} < x_m, x_n < x_{right}$ 且 $y_{bottom} > y_m, y_n > y_{top}$ 则称 M, N 两点间存在垂直方向的障碍, 如图 3 所示, 同理可以定义水平方向有障碍。

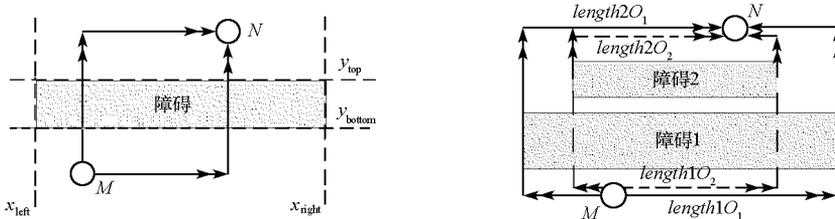


图 3 对节点间障碍的定义并对有障碍的节点距离重新赋值
Fig. 3 Definition of obstacle and evaluation of distance of points that have obstacle

对于有障碍的两点, 我们要重新定义两点间的距离长度来替代两点间的 Manhattan 距离。定义 $newlength = \max(\min(length1 O_1, length2 O_1), \dots, \min(length1 O_n, length2 O_n))$, 这样可以确保新的长度是可以绕过所有障碍物的最短距离。再调用 Prim 算法得到一棵 MST (最小生成树)。在生成 MST 的过程中, 依据 Hanan 定理对于每个新加入的节点, 调用 Dijkstra 算法来确定一条新加入节点到已有生成树的最短路径, 同时确定 Steiner 点即 router 的位置。图 4 可以说明把边的权重根据是否有障碍进行更改后

再执行 Prim 算法,或先使用 Prim 算法再绕过障碍,对最终结果是有影响的。

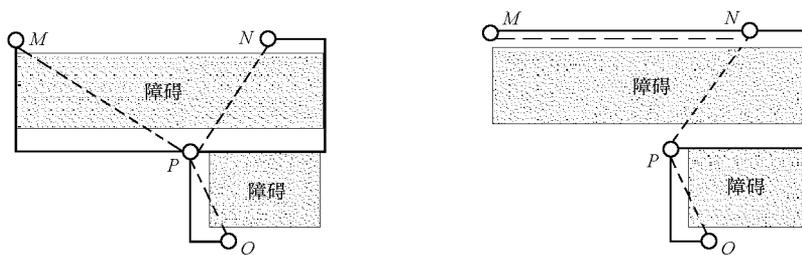


图 4 是否先考虑障碍对结构的影响

Fig. 4 Effect of whether considering obstacle firstly

如果不是先改变边的权重,而是先生成 MST(左侧虚线),然后对通过障碍的边进行重新连接,则得到的是左侧实线连接。而如果首先改变有障碍物的两点间边的权重,则可以得到新的 MST,再把它直角化,可以使总的连线长度缩短。这是因为节点 M 与 P 之间的边长在调用 Prim 算法前被修改,新的 MST 中不再选取这条边的缘故。而在 Prim 算法的执行过程中,每一步加入一个节点到已有的树中,这是一个确定起点和终点的连接问题,可以利用 Dijkstra 算法得到这两点间的最短 Manhattan 路径。同时在这个过程中可以确定 router 的位置(即加入的 Steiner 点)。如图 5 所示,在执行 Prim 算法的过程中,假设 P 是要加入的新节点,新加入的边是 PM ,则在 Escape 图上调用 Dijkstra 算法,可以寻找到从 P 到 M 的最短路径。对于所有可能的路径,选取最先遇到现有 MST 的那条路径,即如图 5 的下方的虚线,交点 S 即为 Steiner 点,也是设置 router 的位置。

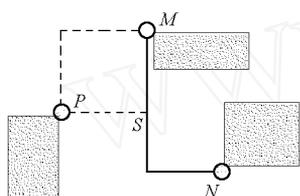


图 5 Steiner 点位置的确定

Fig. 5 Confirm the position of S-point

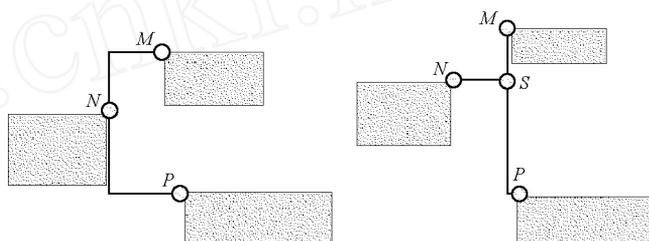


图 6 需要引入新 Steiner 点消除连线冗余

Fig. 6 Correct redundancy by adding new S-point

在算法的执行过程中,连线冗余的修正也非常重要。如图 6 所示, N 点的引入可能产生连线 MN , MN 与 PN 都是最优的,但原节点 M , P 之间却产生了冗余。这需要对 MN 与 PN 连线进行平移,并引入新的 Steiner 点来优化连接。而下述情况是已经有 Steiner 点时,加入新的节点,仍可能产生连线冗余。

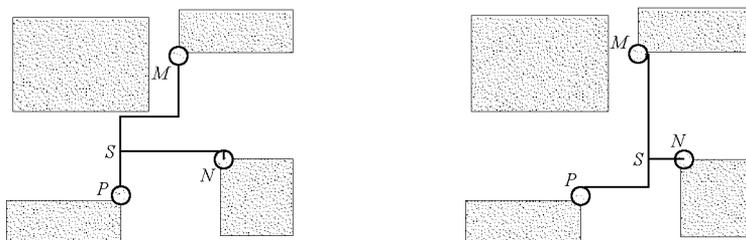


图 7 需要改变 Steiner 点位置以消除连线冗余

Fig. 7 Correct redundancy by moving new S-point

图 7 中,对于新引入的 N 点,原有连线 MP 之间并没有产生冗余,而新节点 N 却与原有节点之一的 M 构成冗余,这需要移动 Steiner 点的位置以消除连线冗余。另外需要强调,在新加入每个节点的时候进行冗余修正与在整个直角 Steiner 树构成之后进行连线冗余的修正,结果是不同的。由于在新加入每个节点时对连线的冗余的修正,可以确保在调用 Prim 算法的每步中都是使用最优的 Steiner 树,这样可

以使得连线长度更短。在一个 $10\ 000 \times 10\ 000$ 的平面定义节点和障碍的坐标作为评价标准,用点的坐标代表待连节点的位置,用障碍物四角的坐标代表障碍物的位置,得到如表 1 所示的对比数据。可以表明,该算法可以缩短全局连线长度,而对连线冗余的处理可以进一步优化结果,见表 2。

表 1 连线长度对比

Tab. 1 Contrast of connection length

节点 (n)	障碍数 (m)	连线长度		
		文献[6]	文献[7]	本文
10	32	-	626	614
30	10	5688	4322	4232
200	500	18 147	11 326	11 239
400	100	20 347	16 957	14 992

表 2 对连线冗余的修正

Tab. 2 Connection of connection redundancy

节点 (n)	障碍数 (m)	连线长度		
		不处理冗余	去除冗余	改进率
10	32	614	614	0
30	10	4271	4232	0.93 %
200	500	11 431	11 239	1.68 %
400	100	18 223	17 843	2.09 %

在上述算法中,对于 n 个节点和 m 个障碍物的输入,建立 Escape 图的时间复杂度为 $O((n + 2m)^2)$,建立障碍物加权图并生成最小生成树的时间复杂度为 $O(n^2)$,执行 Dijkstra 算法的复杂度为 $O(n(\frac{n+2m}{n})^2)$,三者相加化简,最终的时间复杂度为 $O((n + 2m)^2)$ 。

3 研究展望

事实上,在实际电路中,需要考虑的不仅是连线的长度,还要考虑延时的约束。较短的连线长度往往对应于较短的延时,但并不呈严格正比^[8]。电路中的真实延时模型可以用 Elmore 分布 RC 延时模型计算, s 到 t 的延时 $del(s, t)$ 为: $del(s, t) = \sum_{k \in T} R_{kt} C_k$,其中 R_{kt} 是与漏点 t 共享的公共路径的电阻, C_k 是节点 k 上的电容, k 是 T 上的节点。因此,我们的目标函数可以从总连线长度最短转变为线网的全局延时最短,这也是我们下一步研究的重点。

参考文献:

- [1] Dally W J, Towles B. Route Packets, not Wires: On-chip Interconnection Networks [C]//38th Design Automation Conference (DAC) IEEE, 2001: 684 - 689.
- [2] Srinivasan K, et al. A Technique for Design of Application Specific Network-on-Chip Architectures [C]//Proceedings of DATE, Munich, Germany, 2006.
- [3] Srinivasan K, et al. An Automated Technique for Topology and Route Generation of Application Specific On-Chip Interconnection Networks [C]//Proceedings of ICCAD, San Jose, USA, 2005: 231 - 237.
- [4] Gilbert, E. Pllak H. Steiner Minimal Trees [J]. SIAM Journal on Applied Mathematics, 1968: 1 - 29.
- [5] Hanan M. On Steiner's Problem with Rectilinear Distance [J]. SIAM Journal on Applied Mathematics, 1966: 255 - 265.
- [6] Wu P C, Gao J R. A Fast and Stable Algorithm for Obstacle-Avoiding Rectilinear Steiner Minimal Tree Construction [J]. ASP - DAC, 2007: 262 - 267.
- [7] Shen Z, Chu C N, Li Y M. Efficient Rectilinear Steiner Tree Construction with Rectilinear Blockages [J]. ICCD, 2005: 38 - 44.
- [8] 洪先龙. 一种以电性能优化为目标的 Steiner 树算法 [J]. 计算机学报, 1995(4): 266 - 272.