

面向特定应用的片上网络低能耗拓扑生成方法

葛芬, 吴宁

(南京航空航天大学信息科学与技术学院, 江苏 南京 210016)

摘要:为解决面向特定应用的片上网络(network on chip, NoC)设计中的低能耗问题,提出基于遗传算法的层次化拓扑生成方法(genetic algorithm based topology generation, GATG)。该方法以降低网络通信能耗为目标,在满足带宽和延时约束下,根据给定应用的通信需求和路由器结构特征,自动将 IP 核映射到所选路由器单元。同时提出一种基于递归实现的路径构建算法嵌入 GATG,用于确定 IP 核间的通信路径,从而构建路由器之间的链路,最终生成专用的不规则的 NoC 拓扑结构。实验结果表明,GATG 生成结果与采用规则二维 Mesh 结构相比平均降低能耗 46.5%,并可节省 52.7%的路由器资源开销。

关键词:片上网络; 拓扑生成; 映射; 路由; 低能耗; 遗传算法

中图分类号: TN 47

文献标志码: A

DOI:10.3969/j.issn.1001-506X.2010.08.43

Low energy topology generation approach for application-specific network on chip

GE Fen, WU Ning

(Coll. of Information Science and Technology, Nanjing Univ. of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract: A genetic algorithm based hierarchical topology generation approach (GATG) is proposed to solve the problem of low energy design of application-specific network on chip (NoC). The aim is to reduce the network communication energy. Under the constraints of the bandwidth and latency, GATG automatically maps IP cores onto the selected routers according to the communication requirements of given application and the characteristics of router architectures. In addition, a recursion based routing path construction algorithm embedded in GATG is proposed to construct links between routers, and finally the application-specific irregular NoC topology is formed. Experimental results show that GATG achieves a reduction by an average of 46.5% in energy consumption and 52.7% in router resource overhead compared with regular 2D Mesh topology.

Keywords: network on chip; topology generation; mapping; routing; low energy; genetic algorithm

0 引言

随着半导体与集成电路技术的发展,成百甚至上千的 IP 核将集成到单个系统芯片(system on chip, SoC)上,计算资源的增加造成了片上通信需求的巨大增长,而特征尺寸的缩小和设计复杂度的提高使 SoC 内部互连结构的吞吐量、能耗、信号完整性、延时等问题变得更加复杂。传统的基于总线的片上通信结构全局连线长,可扩展性差,不适合处理复杂系统中的数据传输,成为系统性能的瓶颈^[1]。为此,研究者借鉴并行计算和计算机网络的设计思想,提出在单个硅片上构建一个基于数据报文交换的微网络,称为片上网络(network on chip, NoC)^[2],以实现微处理器、DSP、存储器等众多 IP 核之间的数据高速传输。

NoC 必须服务于系统芯片的通信需求。系统芯片的功能、设计条件的约束以及芯片的划分结构决定着 NoC 的形态和设计方法;而 NoC 的通信性能也直接影响片上系统的性能与代价。所以,面向特定应用的专用 NoC 设计方法是研究 NoC 设计的重要目标之一^{—[3-4]}。

面向特定应用设计 NoC,首先需要分析和提取应用系统的通信特征,确定执行各任务 IP 核间的通信数据量和数据依赖关系,构成核通信图;在给定核通信图、NoC 组件库(高度参数化的路由器、网络接口与链路)以及设计约束(能耗、延时、面积等)的基础上,综合得到 NoC 通信结构;将综合所得系统级设计结果输入至布局规划工具,生成最终的版图^[5]。这个过程的关键问题是,根据系统的通信需求生成满足性能约束且最小化网络通信能耗、面积开销的拓

收稿日期:2009-04-17; 修回日期:2009-09-18。

基金项目:江苏省自然科学基金(BK2008387);江苏省研究生创新基金(CX07B_105z)资助课题

作者简介:葛芬(1981-),女,博士后,主要研究方向为片上网络、电子系统集成与专用集成电路设计。E-mail:gefen_g1@hotmail.com

扑结构(确定路由器的个数及其互连形式),并完成 IP 核至路由器的映射以及通信路径分配。

文献[6-13]研究了 IP 核在 NoC 体系结构中的映射和路由问题。文献[6]提出了一种基于分支定界算法的映射算法,并构建了一种避免死锁的确定性路由函数,在满足带宽约束下最小化通信能耗。文献[7]介绍了一种满足带宽约束且最小化系统平均延时的 NMAP 映射和路由算法。文献[8]使用分为 FM 核映射和层次化路由由两步的 MOCA 技术。文献[9]介绍了一种有效映射和生成可靠路由的方法。文献[10]提出了一种基于 PSO 低功耗 IP 核映射和链路平衡的路由分配算法 PLBMR。文献[11-13]采用了基于遗传算法的技术来获得 IP 核在 Mesh 结构 NoC 中的最佳映射,但均没有涉及核间通信的路由路径分配问题。上述论文的研究中均假设 NoC 体系结构已经存在,且都采用规则的 Mesh 拓扑结构。规则的拓扑结构可以很好地控制电气参数和通信噪声,但可能导致大规模冗余路由器、较低的链路利用率或者局部的拥塞,不能满足专用 NoC 的设计要求^[4]。

为了解决专用 NoC 设计中的低能耗问题,本文面向应用定制专用拓扑结构,提出基于遗传算法的拓扑生成方法(genetic algorithm based topology generation, GATG)。该方法包含路由器选择、核映射与路径构建三个层次,以降低网络通信能耗为目标,在满足带宽和延时约束下,根据给定应用的通信需求和路由器结构特征,将 IP 核合理映射至所选路由器,并提出一种基于递归实现的路径构建算法嵌入 GATG,用于生成路由器间的拓扑连接,最终生成优化的专用不规则 NoC 拓扑结构。

1 问题描述

NoC 体系结构由互连的路由器组成。路由器负责路由由全局通信结构上的数据报文,其结构如图 1(a)所示。路由器由四部分组成:交换开关、路由和仲裁模块、输入和输出端口模块。各种计算 IP 核与路由器通过端口的输入输出通道相连,通道由两条单向的链路组成。每条链路可以通过一个基于 OCP 协议的网络接口(network interface, NI)连接一个 IP 核,也可以直接连接到其他路由器来扩展 NoC 体系结构^[14],如图 1(b)所示。如此,设计者可以基于应用需求和约束条件构建不同的规则或不规则的 NoC 拓扑结构。

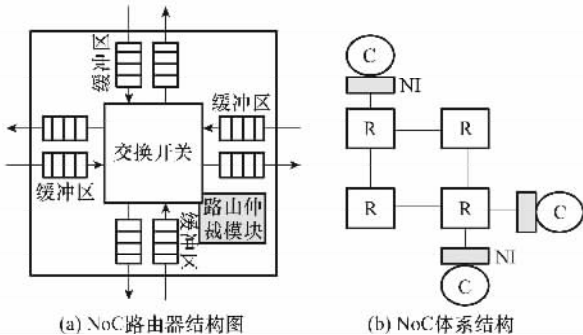


图 1 NoC 路由器结构图与体系结构

能耗优化的拓扑生成问题的描述如下。

给定一个核通信图(core communication graph, CCG) $CCG(C, A)$,其中顶点 $c_i \in C$ 表示一个 IP 核,有向边 $a_{i,j} = (c_i, c_j) \in A$ 表示从 c_i 到 c_j 的通信踪迹,权重 $b(a_{i,j})$ 表示通信踪迹 $a_{i,j}$ 的带宽需求(Mbps),权重 $l(a_{i,j})$ 表示从 c_i 到 c_j 的延时约束(hops)。

给定路由器结构特征库 \mathcal{L} ,对每一个路由器属性 η 表示路由器输入输出端口数目,每个端口可以映射一个 IP 核或者与邻接路由器建立一条链路;属性 Ω 表示路由器任意一个端口可以支持的最大带宽。

求一个网络拓扑结构 $T(R, E)$,其中 $R \in \mathcal{L}$ 表示在拓扑生成中从库 \mathcal{L} 中选择使用的路由器集合, E 表示路由器之间的连接通道集合,使得:

- (1) 存在一组多对一映射 $Map(c)$

$$Map: C \rightarrow R \Rightarrow r_i = Map(c_i), \forall c_i \in C, \exists r_i \in R$$

表示 IP 核 c_i 被映射到路由器 r_i 上,且每个路由器上可布局的 IP 核数目须小于端口数 η_r 。

- (2) 存在一个路由函数 $Routing$

$Routing: A \rightarrow P$,能为每一个 $a_{i,j} \in A$ 在 T 中找到一条唯一的路径 $p_{i,j} = \{(r_i, r_k), (r_k, r_m), \dots, (r_n, r_j)\} \in P$ 满足通信延时约束

$$\forall a_{i,j} \in A, l(a_{i,j}) \geq d(p_{Map(c_i), Map(c_j)}) \quad (1)$$

式中, (r_i, r_k) 表示在 T 中路由器 r_i 与 r_k 间通过端口建立拓扑连接 $e_{i,k}$,构成物理链路 l_k ; $d(p_{i,j})$ 表示沿路径 $p_{i,j}$ 路由器 r_i 与 r_j 间的路由距离,用跳数(hops)表示,相邻路由器 r_i 到 r_k 表示一跳。

路径 $p_{i,j}$ 还需满足路由器端口带宽约束,即路由器端口所能提供的带宽不小于所有经过该端口构建的路径的通信带宽需求之和

$$\sum_{\forall a_{i,j} \in A} b(a_{i,j}) f(l_k, p_{Map(c_i), Map(c_j)}) \leq \Omega \quad (2)$$

式中

$$f(l_k, p_{Map(c_i), Map(c_j)}) = \begin{cases} 0, & l_k \notin p_{Map(c_i), Map(c_j)} \\ 1, & l_k \in p_{Map(c_i), Map(c_j)} \end{cases}$$

- (3) 核间通信能耗最小

$$\min E(A) = \sum_{\forall a_{i,j} \in A} b(a_{i,j}) \times E_{bit}^{a_{i,j}} \quad (3)$$

式中

$$E_{bit}^{a_{i,j}} = \sum_{r \in p_{Map(c_i), Map(c_j)}} E_{Rbit} + \sum_{e \in p_{Map(c_i), Map(c_j)}} E_{Lbit} =$$

$$(d(p_{Map(c_i), Map(c_j)}) + 1) \times E_{Rbit} + d(p_{Map(c_i), Map(c_j)}) \times E_{Lbit}$$

式中, $E_{bit}^{a_{i,j}}$ 表示沿路径 $p_{Map(c_i), Map(c_j)}$ 传输 1 bit 数据的能耗; E_{Rbit} 表示传输在路由器上的能耗; E_{Lbit} 表示路由器之间链路路上的能耗^[14]。

式(3)是一个线性方程, E_{Rbit} 和 E_{Lbit} 是常数,可见最小化通信能耗与最小化 $d(p_{Map(c_i), Map(c_j)})$ 等价。因此,能耗优化的公式可改写为

$$\min E(A) = \sum_{\forall a_{i,j} \in A} b(a_{i,j}) \times d(p_{Map(c_i), Map(c_j)}) \quad (4)$$

论文所提出方法的基本思想是在满足带宽和延时约束条件下,将尽可能多的核(特别是核间通信量多的核)分配到相同或相邻的路由器单元,这样大多数数据交换只需经过短的路径。

2 专用 NoC 拓扑生成

基于映射和路径分配的专用 NoC 拓扑生成属于二次分配问题的范畴,而二次分配问题已被证明是 NP 完全的,其搜索空间随着网络尺寸的增长呈阶乘递增。因此,本文

提出了 GATG,通过将 IP 核合理映射至路由器以及构建核间最优通信路径,来定制满足性能约束且最小化能耗的面向特定应用的 NoC 拓扑结构。

2.1 GATG 基本思想

我们将专用 NoC 拓扑生成问题划分为三个关键步骤:从特征库中选择合适数目的路由器;将 IP 核映射至所选路由器各端口;为核间通信分配最优的路由路径以构建路由器之间的拓扑连接链路。

对应上述三个步骤,GATG 采用层次化方式模拟种群,算法流程如图 2 所示。

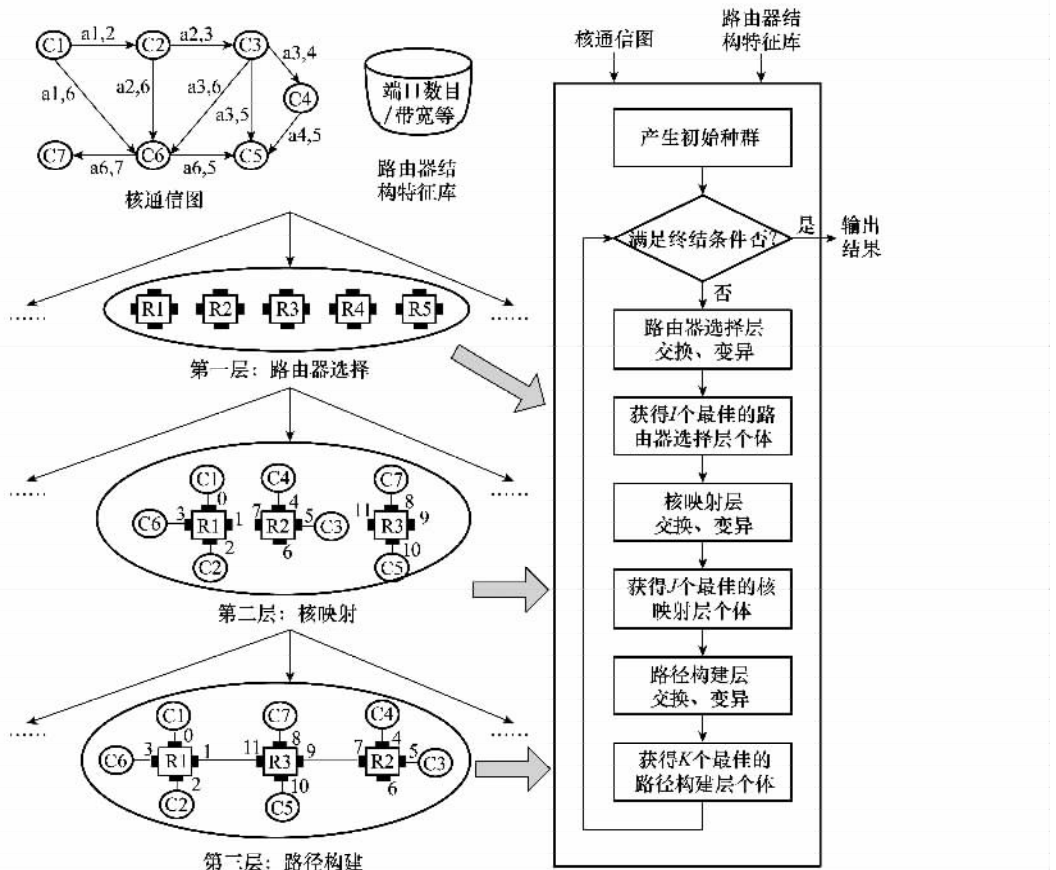


图 2 GATG 算法流程图

第一层为路由器选择层,GATG 设定有 I 种不同数目的路由器选择方式。第二层为核映射层,针对第一层中选择的每个路由器,GATG 设定 J 种不同的 IP 核到路由器各端口的映射。第三层为路径构建层,由从源节点到目标节点经过的路由器端口构建路由路径,针对第二层中每个 IP 核到端口的映射,GATG 设定 K 种不同的通信踪迹路径分配方案。因此,GATG 解集中种群的个数 $P=I \times J \times K$ 。

GATG 首先读取核通信图 CCG(C, A)以及路由器结构特征库信息,生成初始种群。GATG 的优化流程是个反复的迭代过程。在每一代中,个体适应度越大,复制到下一代种群数越多,并通过交换和变异,生成新一代种群。对此种群再进行适应度计算,如此迭代反复,直到满足终止条件为

止。终止条件即最大迭代次数 N 等于 CCG 中顶点和边的数目的总和,即 $N=|C|+|A|$ 。

GATG 中的种群复制、交换和变异依次需在路由器选择层、核映射层和路径构建层这三个层次分别进行。各层次应用的交换概率不同,设路由器选择层的交换概率为 p_r ,核映射层是 p_n ,路径构建层为 p_t ,那么 $p_r \ll p_n \ll p_t$ 。每一层中交换产生的个体数目远远大于变异产生的。在每一层,通过遗传操作产生新的个体,新一代通过在当前代和新的个体中选择 M 个适应度最大的个体来产生,其中在第一层 $M=I$,在第二层 $M=J$,在第三层 $M=K$ 。一旦上一层交换变异产生了新个体,须在下一层中更新相应的个体。

适应度在路径构建层给定,即须在完成路由器选择、核

至路由器端口映射以及通信踪迹路径构建从而形成定制的拓扑结构后计算。在路由器选择层,每种路由器选择对应于 J 种核映射结果,每种核映射又对应于 K 个核间路径分配实例,所以第一层中每个路由器选择个体适应度通过 $J \times K$ 种通信踪迹路径分配个体中最大的适应度获得。类似的,第二层中每个核映射个体的适应度为相应的 K 种通信踪迹路径分配个体中最大的适应度。

专用 NoC 拓扑生成的优化目标定位于核间通信能耗,适应度函数是获得最优化目标的一个代价函数。此外,在路由路径构建中有可能在满足约束条件下从源到目的节点不能找到一条路径,在这种情况下,需对不能分配路径的通信踪迹给予惩罚。根据第 2 节阐述的思想,统一代价函数定义如下

$$\text{cost} = E(A) + \gamma \times N_{\text{unassigned}} \quad (5)$$

式中, $N_{\text{unassigned}}$ 表示未被分配路径的通信踪迹的数目; γ 为惩罚系数,一般取一个较大的值,我们可以设 $\gamma = 10\ 000$ 。GATG 中适应度函数设为代价函数的倒数。

2.2 路由器选择

路由器选择层完成从路由器结构特征库中选择一定数目的路由器。种群个体用二进制数组 $\text{router}[M_R]$ 表示,数组第 i 个元素值为“1”表示第 i 个路由器被选中,“0”表示该路由器没有被采用。其中 M_R 是可利用的路路由器的最大数目, M_R 等于 CCG 中顶点和边数目之和的两倍,即为 $M_R = 2 \times (|C| + |A|)$ 。如果路由器库 \mathcal{L} 由多种类型的路由器组成(例如,四端口的路由器和五端口的路由器),那么 $M_R = |\mathcal{L}| \times 2 \times (|C| + |A|)$,每一种类型有 $M_R / |\mathcal{L}|$ 个路由器。我们用 r_i 表示每个路由器,其中 $1 \leq i \leq M_R$ 。此外,还需为每个被选用的路由器各端口分配一个唯一的标号。例如,若 $\text{router}[0]$ 对应的路由器 r_1 被选用,则其四个端口标为 0、1、2、3, $\text{router}[1]$ 对应的被选用路由器 r_2 的四个端口标为 4、5、6、7,依此类推。

初始路由器选择通过为 router 数组每个元素随机分配“0”和“1”来产生。交换操作采用一点交换。变异操作作为在 router 数组中随机选择一个元素,将相应的位反转。如果元素值“0”反转为“1”,表示在 NoC 拓扑结构中新增加了一个路由器,下面的核映射层和路径构建层不需要发生变化。但如果元素值“1”反转为“0”,表示移除一个先前使用的路由器,则包含属于被移除路由器端口的相应的核到路由器端口的映射以及通信踪迹路径构建的种群个体都需要重新产生。

2.3 核映射

在核映射层,核到路由器端口的映射用 $\text{port}[|C|]$ 整型数组表示,数组第 i 个元素表示顶点 $c_i \in C$ 的端口分配结果,元素值为上述可用路由器端口的标号值,且一个端口只能分配一个 IP 核(一个端口标号只能被选择一次)。例如,图 2 中核 c_1 映射至路由器 r_1 的端口 0。

初始核至路由器端口的映射通过为 CCG 中每个顶点在可用端口中随机选择一个来获得,交换操作仍采用一点交换。在核映射层变异,随机选择一个核映射个体,同时在

其对应的路由器选择个体中随机选择两个可用的路由器端口 (u, v) 。在当前选择的核映射结果中,如果有两个不同的 IP 核映射到端口 u 和 v ,则将这两个 IP 核布局交换;如果只有一个 IP 核映射到其中一个端口,比如 u ,那么改变该 IP 核映射至 v ;如果没有任何 IP 核映射到 u 或者 v ,则反复重新选择两个新的端口,直到存在 IP 核映射到所选端口。一旦生成了一个新的核映射个体,则表示 IP 核的布局发生了变化,需要重新生成路径构建层的种群个体。

2.4 路径构建

在确定了路由器及 IP 核的布局后,拓扑生成问题的最后一步为生成路由器之间的互连关系,论文通过确定 IP 核间的通信路径来构建路由器之间的拓扑连接。GATG 中路径构建层的种群个体用一个整型数组 trstr_i 的链表 tracelist 表示,第 i 个数组表示第 i 条踪迹的路由路径。每个 trstr 数组是端口的顺序序列,表示传输的流向。例如,图 2 中一条通讯踪迹 $a_{2,3}$ 顺序穿过端口 1、11、9、7,则用数组 $[1, 11, 9, 7]$ 表示。由为通信踪迹分配的路由路径构成路由器之间的物理链路。

相比路由器选择和核映射,初始路由路径分配相对复杂些,需在 A 中随机选择一条通信踪迹 $a_{i,j}$,采用一种路径构建算法产生从源节点到目的节点的传输路由。如前所述,一条通信路径通过一个端口标号的数组来表示。经过多个路由器端口的传输路径建立了不同路由器两个端口之间的一条物理链路。典型的最短路径算法(如 Dijkstra)都是基于已知网络拓扑结构分配通信路径的,显然,在本文讨论的拓扑生成问题中,不能采用这些最短路径算法为每条通信踪迹分配路由路径。因此,论文提出一种基于递归实现的路径构建算法来解决拓扑生成中的路由路径分配问题,其基本思路是在满足带宽和延时约束下,通过先前通信踪迹建立的链路完成所有踪迹的路径分配。该算法能够尽可能建立短而少的链路来为每条通信踪迹分配路径,同时降低所构建的拓扑结构的复杂性,以达到节省通信延时和能耗的目的。

算法首先根据当前路由器选择和端口映射的结果获得通信踪迹对应两个 IP 核所布局的源和目的路由器,然后在满足带宽和延时约束条件下尝试采用三种方式:① 利用源和目的路由器之间已有链路;② 在源和目的路由器之间通过路由器没有布局 IP 核的空余端口构建新的链路;③ 利用源或者目的路由器与其他路由器之间由先前通信踪迹建立的链路,从源节点开始逐步递归寻找,直至达到目的节点。在这个过程中为每条通信踪迹分配路径,从而形成定制的拓扑结构。算法描述及伪代码如下:

算法输入:通信踪迹 $a_{i,j} = (c_i, c_j)$ 对应 IP 核 c_i 和 c_j 映射到的路由器 r_{src} 与 r_{dest} 。

算法输出:生成 $a_{i,j}$ 的路径 $p_{rsrc, rdest} = \{(r_{src}, r_{next}), \dots, (r_n, r_{dest})\}$ 。

递归结束条件:当 $r_{src} = r_{dest}$ 或找不到可分配路径。

递归函数: $\text{route_construction}(r_{src}, r_{dest})$

```

{if ( $r_{src} == r_{dest}$ ) exit;
if (no link between  $r_{src}$  and  $r_{dest}$ )
    {if (existing empty port  $port_i$  and  $port_j$  in  $r_{src}$  and  $r_{dest}$ )
        {construct link between  $port_i$  and  $port_j$ , let  $r_{src} = r_{dest}$ , add  $b(a_{i,j})$  to  $RA_{dj}[r_{src}][r_{dest}]$ , and update  $PortList$ , exit;}
    else if (no empty port in  $r_{src}$ )
        { $r_{next} = get\_next\_rtr(r_{src})$ ;
        if ( $r_{next} \neq NULL$ )
            {let  $r_{src} = r_{next}$ , add  $b(a_{i,j})$  to  $RA_{dj}[r_{src}][r_{next}]$ ;
            route\_construction( $r_{src}$ ,  $r_{dest}$ );}
        else add  $a_{i,j}$  to  $PathUnAssignedSet$ , exit;}
    else if (no empty port in  $r_{dest}$ )
        { $r_{next} = get\_next\_rtr(r_{dest})$ ;
        if ( $r_{next} \neq NULL$ )
            {let  $r_{dest} = r_{next}$ , add  $b(a_{i,j})$  to  $RA_{dj}[r_{next}][r_{dest}]$ ;
            route\_construction( $r_{src}$ ,  $r_{dest}$ );}
        else add  $a_{i,j}$  to  $PathUnAssignedSet$ , exit;};}
else if (existing link between  $r_{src}$  and  $r_{dest}$ )
    {if ( $RA_{dj}[r_{src}][r_{dest}] + b(a_{i,j}) \leq \Omega$  &&  $d(p_{r_{src}, r_{dest}}) \leq l(a_{i,j})$ )
        {let  $r_{src} = r_{dest}$ , add  $b(a_{i,j})$  to  $RA_{dj}[r_{src}][r_{dest}]$ , exit;}
    else {
         $r_{next} = get\_next\_rtr(r_{src})$ ;
        if ( $r_{next} \neq NULL$  &&  $r_{next} \neq r_{dest}$ )
            {let  $r_{src} = r_{next}$ , add  $b(a_{i,j})$  to  $RA_{dj}[r_{src}][r_{next}]$ ;
            route\_construction( $r_{src}$ ,  $r_{dest}$ );}
        else if (existing empty port  $port_i$  and  $port_{next}$  in  $r_{src}$  and  $r_{next}$ )
            {construct link between  $port_i$  and  $port_{next}$ ;
            let  $r_{src} = r_{next}$ , add  $b(a_{i,j})$  to  $RA_{dj}[r_{src}][r_{next}]$ ;
            update  $PortList$ ;
            route\_construction( $r_{src}$ ,  $r_{dest}$ );}
        else add  $a_{i,j}$  to  $PathUnAssignedSet$ , exit;};}
}

```

其中,函数 $get_next_rtr(r_i)$ 返回与路由器 r_i 相连,且构建的链路满足带宽和延时约束的相邻路由器 r_{next} 。邻接矩阵 $RA_{dj}[M_R][M_R]$ 表征当前路由器分配方案下路由器之间互连关系。矩阵元素值为 0 表示对应两个路由器在路由选择层被选中,可以根据路径分配情况建立连接;元素值为 ∞ 表示对应两个路由器不能建立任何连接;元素值介于 0 和 ∞ 之间表示对应两个路由器在路径分配中建立了连接,且元素值为通信踪迹经过该链路占用的带宽。分配完所有通信踪迹的路径后,检查 $RA_{dj}[M_R][M_R]$ 中的每个元素值,就可以获得当前路径分配下需占用的最大链路带宽,其不应超过路由器端口支持最大带宽 Ω 。端口信息链表 $PortList$ 用于详细记录路由器各个端口的状态,包括该端口是否为空,不为空时连接的是哪个 IP 核或是与另外哪个路由器端口建

立了链路。路径构建完后,根据 $RA_{dj}[M_R][M_R]$ 和 $PortList$ 中记录的信息,便可获得一种定制的拓扑结构。

在路径构建层,变异操作为在通信踪迹路径分配链表中随机选择一个已分配了路径的踪迹,添加到未被分配集合 $PathUnAssignedSet$,然后在未被分配集合中随机选择一条通信踪迹调用路径构建算法为其分配路径,直到尽可能多的未分配踪迹被处理。

3 实验结果及分析

专用 NoC 拓扑生成算法程序源代码用 C++ 编写,在 Microsoft Visual C++ 6.0 环境下完成编译和仿真。实验所使用的 CPU 为 Pentium(R) D 3.4GHz,内存 1GB。实验中使用的测试应用包括音频、视频编解码器及其各种组合,如表 1 所示,这些应用的通信踪迹见文献[8,15]。

表 1 测试应用核通信图

核图	核图 ID	顶点数	边数
MP3 dec	G1	6	6
H. 263 dec	G2	7	8
MP3 enc	G3	7	8
H. 263 enc	G4	8	11
MWD	G5	12	13
VOPD	G6	12	15
MPEG4	G7	12	26
H. 263 enc-MP3 dec	G8	12	17
H. 263 enc-MP3 enc	G9	14	19
H. 263 enc-H. 263 dec	G10	15	19

实验方案是将表 1 所述的各种应用采用 GATG 方法生成结果与其映射至规则二维 Mesh 结构的方案作比较,其中 G1~G4 被映射至 3×3 的 Mesh 结构,G5~G10 被映射至 4×4 的 Mesh 结构。路由器结构特征包括:设置路由器端口数为 $\eta=4$,端口可以支持的最大带宽 $\Omega=1$ Gb/s。在采用 GATG 生成 NoC 拓扑结构过程中,路由器选择个体值 $I=10$,核到端口映射个体值 $J=10$,路径构建个体值 $K=10$;交换概率 $p_r=0.1, p_n=0.25, p_e=0.9$;变异概率 $p_m=0.01$ 。

实验中对各种测试应用进行了拓扑生成,算法均在合理的 CPU 时间(<15 min)内求得了近似最优解。以各种应用在规则二维 Mesh 结构中的随机映射方案(Random-Mesh)作为参考方案(通信能耗归一化为 1),各种应用在 NoC 结构中的通信能耗开销如图 3 所示。

“Optimal-Mesh”表示采用文献[12]中介绍的基于遗传算法的映射方法将各测试应用映射至二维 Mesh 结构以获得最小通信能耗,该类常用方法采用一对一映射和维序路由,实现简单。由图 3 可知,在不同规模的测试应用下,与采用规则 Mesh 结构相比,GATG 生成的专用不规则 NoC 体系结构的能耗最小,与 Optimal-Mesh 相比平均降低了 46.5%。

图 4 描述了由 GATG 生成的 NoC 拓扑结构与规则二维 Mesh 结构使用的路由器数目比较,GATG 方法可平均节省 52.7% 的路由器资源。我们可以看出,尽管 MWD

(G5)和 VOPD(G6)应用中的 IP 核数目均为 12,但生成的优化的专用拓扑结构却不同,分别需要 7 个和 8 个路由器,与规则 Mesh 网络(16 个路由器)相比需要较少的路由器资源。

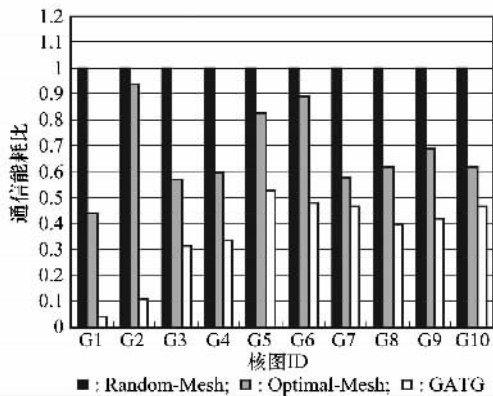


图 3 通信能耗实验结果

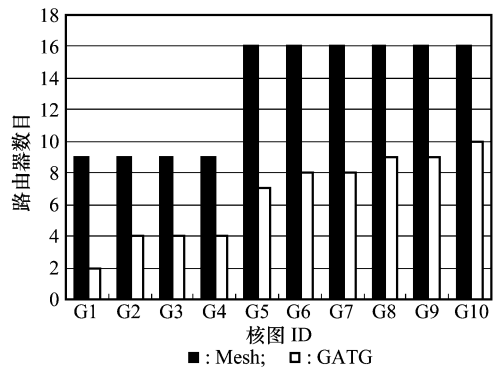


图 4 路由器资源开销实验结果

由实验看出,GATG 对于解决基于映射和路径分配思想的专用 NoC 拓扑生成问题是非常有效的。该方法通过遗传技术实现 IP 核到路由器的多对一映射,并采用基于递归实现的路径构建算法建立短而少的链路生成专用拓扑结构,从而比基于一对一映射及循序路由的规则二维 Mesh 结构获得更小的通信能耗和路由器资源开销。

4 结束语

本文提出了一种 GATG 的方法,采用遗传算法解决了专用 NoC 设计中的拓扑生成问题。GATG 以最小化 NoC 通信能耗为目标,根据实际应用的通信需求和路由器结构特征,在满足设计约束下,为实际应用定制专用拓扑结构,同时实现了 IP 核映射和通信踪迹的路径分配。特别地,GATG 中提出了一种基于递归实现的路径构建算法,通过确定 IP 核间的通信路径来构建路由器之间的拓扑连接。使用多媒体测试应用进行结果评测,实验显示,与采用规则二维 Mesh 结构相比,GATG 生成的专用不规则 NoC 体系结构平均降低通信能耗 46.5%,且可节省 52.7% 路由器资源。未来工作将进一步改进算法,考虑芯片面积约束,使得能以更少的路由器和链路连接生成满足应用需求的专用拓扑结构。

参考文献:

- [1] Jerraya A, Tenhunen H, Wolf W. Multiprocessor system-on-chips [J]. *IEEE Computer*, 2005,38(7):36-40.
- [2] Benini L, Micheli G D. Networks on chip: a new SoC paradigm [J]. *IEEE Computer*, 2002,35(1):70-78.
- [3] 马立伟,孙义和. 片上网络拓扑优化:在离散平面上布局与布线 [J]. *电子学报*, 2007,35(5):906-911.
- [4] Bertozzi D, Jalabert A, Murali S, et al. NoC synthesis flow for customized domain specific multiprocessor systems-on-chip [J]. *IEEE Trans. on Parallel and Distributed Systems*, 2005,16(2):113-129.
- [5] Srinivasan K, Chatha K S. ISIS: a genetic algorithm based technique for custom on-chip interconnection network synthesis [C]// *Proc. of the International Conference on VLSI Design*, Washington: IEEE Press, 2005:623-628.
- [6] Hu J, Marchlescu R. Exploiting the routing flexibility for energy/performance aware mappings of regular NoC architectures [C]// *Proc. of the Design, Automation and Test in Europe Conference and Exhibition*, Munich: IEEE Press, 2003:688-693.
- [7] Murali S, Micheli G D. Bandwidth-constrained mapping of cores onto NoC architectures [C]// *Proc. of the Design, Automation and Test in Europe Conference and Exhibition*, Paris: IEEE Press, 2004:896-901.
- [8] Srinivasan K, Chatha K S. A technique for low energy mapping and routing in network-on-chip architectures [C]// *Proc. of the International Symposium on Low Power Electronics and Design*, San Diego: IEEE Press, 2005:387-392.
- [9] Harmanani H M, Farah R. A method for efficient mapping and reliable routing for NoC architectures with minimum bandwidth and area [C]// *Proc. of the 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference*, Montreal: IEEE Press, 2008:29-32.
- [10] 周文彪,张岩,毛志刚. PSO 基础的链路平衡 NoC 映射与路由 [J]. *系统工程与电子技术*, 2008,30(12):2506-2511. (Zhou Wenbiao, Zhang Yan, Mao Zhigang. Mapping and routing of link-balance NOC based on PSO[J]. *Systems Engineering and Electronics*, 2008,30(12):2506-2511.)
- [11] Ascia G, Catania V, Palesi M. A multi-objective genetic approach to mapping problem on network-on-chip [J]. *Journal of Universal Computer Science*, 2006,12(4):370-394.
- [12] Jena R K, Sharma G K. A multi-objective evolutionary algorithm based optimization model for network-on-chip synthesis [C]// *Proc. of the International Conference on Information Technology*, Rourkela: IEEE Press, 2007:977-982.
- [13] Moein-darbari F, Khademzade A, Gharooni-fard G. CGMAP: a new approach to network-on-chip mapping problem [J]. *IEICE Electronics Express*, 2009,6(1):27-34.
- [14] Chang K C, Chen T F. Low-power algorithm for automatic topology generation for application-specific networks on chips [J]. *IET Computers & Digital Techniques*, 2008,2(3):239-249.
- [15] Hu J, Marchlescu R. Energy-and performance-aware mapping for regular NoC architectures [J]. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 2005,24(4):551-562.