

面向能耗和延时的 NoC 映射方法

杨盛光, 李 丽, 高明伦, 张宇昂

(南京大学微电子设计研究所, 江苏南京 210093; 江苏省光电信息功能材料重点实验室, 江苏南京 210093)

摘 要: 随着对 NoC 平台研究的逐步深入, 如何将规模庞大的应用合理地映射到 NoC 平台上成为亟待解决的问题之一. 本文基于二维网格结构 NoC 平台, 建立了旨在优化系统通信能耗和执行时间的统一目标函数. 提出了通过优化链路负载分布间接优化延时的方法, 避免了 NoC 等待延时精确建模的难题. 并且采用蚁群算法实现了面向能耗和延时的 NoC 映射. 调整参数 λ , 可以选择单一目标或者联合目标优化. 本文还对映射结果进行了执行时间模拟. 实验结果显示: 与随机映射相比, 单一目标优化在通信能耗和执行时间上分别能节省 (30%~47%) 和 (20%~39%), 而联合目标优化则能在能量支配的映射方案中进一步挖掘时间维度的潜力.

关键词: 片上网络; 映射; 能耗; 延时; 蚁群算法

中图分类号: TN47 **文献标识码:** A **文章编号:** 0372-2112 (2008) 05-0937-06

An Energy- and Delay-aware Mapping Method of NoC

YANG Sheng-guang, LI Li, GAO Ming-lun, ZHANG Yu-ang

(Institute of VLSI design, Nanjing University, Nanjing, Jiangsu 210093, China;

Key Laboratory of Advanced Photonic and Electronic Materials, Nanjing, Jiangsu 210093, China)

Abstract: With the research on the NoC deeper and deeper, it becomes a stringent task to map a complex application onto an appointed NoC platform efficiently. Based on 2-D mesh NoC platform, this paper sets up a united object function in order to optimize communication energy and execution time. It is also proposed that the delay can be optimized indirectly by means of optimizing distribution of link load, which can avoid the difficulty to model waiting time in NoC accurately. Ant colony algorithm is used to realize the energy- and delay-aware NoC mapping. Single object (energy/ delay) or collaborate object can be selected to optimize the mapping by modifying parameter λ . Furthermore, the optimized solutions are simulated on NoC platform finally. Experimental results show single object optimization mapping can reduce communication energy and execution time by (30%~47%) and (20%~39%) respectively compared with random mapping. And collaborate object optimization mapping can provide a seductive tradeoff between energy and execution time.

Key words: network on a chip (NoC); mapping; energy; delay; ant colony algorithm

1 引言

在愈来愈短的面市时间内完成一款复杂系统芯片 (System on a chip, SoC) 的设计对芯片设计效率提出了巨大的挑战. IP (Intellectual Property) 核与互连结构重用将成为未来芯片设计方法学最主要的特征. 片上网络 (Network on a chip, NoC) 为设计未来复杂 SoC 提供了一种新的范例^[1~4].

NoC 设计两大内容分别为: NoC 平台设计和应用到 NoC 平台的映射. 本文旨在研究应用到 NoC 平台的映射方法. 一个特定的任务总是可以通过某种形式的任务图 (Task graph, TG) 来表示. 而映射就是按照一定的规则将

TG 指定的任务分配到目标 NoC 平台上, 从而实现该应用.

NoC 映射问题是 NoC 领域的一个开放性课题^[5~10]. 映射的优化目标通常是最小化通信成本^[5]或者能耗^[6,7], 而任务执行时间是另一个重要目标. 单纯地考虑能耗最小化, 可能导致局部区域网络拥塞, 从而影响执行时间.

文献[8]采用两步遗传算法来实现任务映射问题, 它以 IP 核之间的曼哈顿 (Manhattan) 距离来估计两者之间的通信延时, 而没有考虑到拥塞导致的等待时间, 因此不尽合理. 文献[9]采用排队理论, 用类似于共享总线等待时间建模的方法来近似估计 NoC 等待时间, 但 NoC

收稿日期: 2007-07-31; 修回日期: 2008-01-18

基金项目: 国家自然科学基金 (No. 90307011, No. 60576034); 江苏省高技术研究项目 (No. BG2005030)

通信比共享总线复杂许多,其精确度需要关注.该模型计算复杂,优化结果可使任务执行时间平均下降 20%.类似的,文献[10]也采用直接建模方法考虑映射中的能耗和延时问题.

本文综合考虑了通信能耗和延时,抓住 NoC 通信特征,将两者优化重点定位于加权距离之和与链路负载分布,建立了旨在优化通信能耗和执行时间的目标函数,并且采用蚁群算法实现了面向能耗和延时的 NoC 映射.调整相应参数,可以选择单一目标或者混合目标进行优化.

2 NoC 平台概述

2.1 NoC 平台结构

NoC 平台是一个参数化的、二维网格结构的 SystemC 模型. NoC 由路由器(R)、链路和本地子系统(LS)组成(4×4 规模,见图 1). 路由器是网络的核心部件,构成开关(Switch)节点;本地子系统包含 IP 核和网络接口(NI),构成资源(Resource)节点.

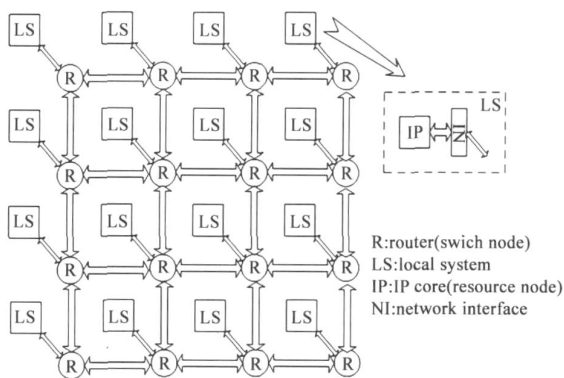


图 1 NoC 平台结构

平台采用虫孔交换方式、输入通道循环 FIFO (First in First out) 缓存策略、轮转优先级仲裁机制和静态 XY 路由算法。静态路由可以保证通信路径的可预测性,以便开展 NoC 映射问题研究.

2.2 路由器结构

上述路由器设计为五个双向端口:北(North)、东(East)、南(South)、西(West)和本地(Local). 每个端口具有输入和输出通道. 输入通道包含缓存和路由逻辑,输

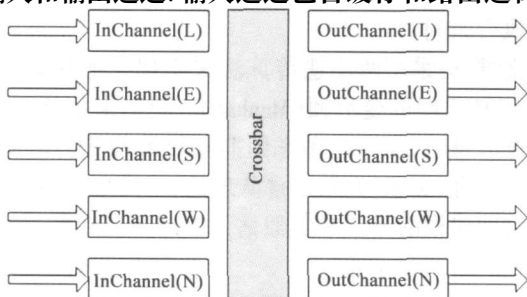


图 2 路由器结构

出通道包含仲裁逻辑. 路由器结构如图 2 所示.

3 能耗和延时模型

3.1 能耗模型

文献[11]定义了一个能耗模型,用于估计单比特数据经过路由器所消耗的能量.而文献[6]则修改了该模型,使之适用于二维网格结构 NoC. 模型中,单比特数据由节点 r_i 传输到节点 r_j 平均消耗的能量如式(1)所示.

$$E_{bit}^{i,j} = (h_{i,j} + 1) \times E_{sbit} + h_{i,j} \times E_{lbit} \quad (1)$$

式中 E_{sbit} 和 E_{lbit} 分别表示在开关和链路上消耗的能量. $h_{i,j}$ 表示节点 r_i 到节点 r_j 的步(hop)数,即两节点之间的曼哈顿距离. 式(1)是一个线性方程, E_{sbit} 和 E_{lbit} 为常系数,可见最小化能耗与最小化 $h_{i,j}$ 等价. 因此,可以将两节点之间的加权曼哈顿距离作为它们的通信能耗指标.

3.2 延迟模型

采用虫孔交换和静态 XY 路由的二维网格 NoC 中,从源节点 r_i 到目标节点 r_j 传输一个数据包的延时($T_{i,j}$)定义为这样一个时间段: r_i 处包头产生开始,包尾到达 r_j 结束. $T_{i,j}$ 模型如式(2)所示.

$$T_{i,j} = (T_b + T_w) \times h_{i,j} + T_b \times (B - 1) \quad (2)$$

式中, T_b 为无拥塞时一帧(flit)数据通过一个开关和一条链路所需要的时间; T_w 为存在拥塞时包头在开关节点处平均等待时间; $h_{i,j}$ 为 r_i 到 r_j 之间的曼哈顿距离; B 为数据包包含的数据帧数. 其中, T_b 和 B 是常系数. 式(2)中,第一项为包头到达目的地所需时间,第二项为后继帧到达目的地所需时间,为常数项. 因此, $T_{i,j}$ 的优化空间在于第一项,取决于参数 T_w 和 $h_{i,j}$. $h_{i,j}$ 由 r_i 和 r_j 的相对位置关系决定,而 T_w 则取决于网络的拥塞状况.

NoC 传输延时具有动态特征,其复杂性远高于共享总线延时,因此几乎不可精确预测. 通过忽略或者人为设定 T_w 来进行延时估计的映射策略都是不够准确的. 考虑到精确建模 T_w 的困难和复杂性,我们转向了通过优化影响 T_w 的关键因素来间接优化 T_w 的方法. 只要关键因素定位合理,优化效果是可以保障的,甚至可以比直接优化 T_w 近似模型的方法还好.

减小 T_w 最有效的途径就是缓解网络拥塞,而缓解拥塞的关键则在于平衡链路负载. 基于该思想,本文选择平衡链路负载来缓解拥塞,优化 T_w , 减小传输延时,从而达到降低任务执行时间的目的.

3.3 目标函数

NoC 优化目标定位于通信能耗和执行时间. 根据 3.1 节阐述的思想,能耗指标定义如式(3),

$$E(C) = \sum_{i=1}^N \sum_{j=1}^N w_{i,j} \times h_{i,j} \quad (3)$$

式中 $w_{i,j}$ 为 r_i 与 r_j 之间的通信量, N 为节点数. 可见,通

信能耗的优化目标就是最小化各节点间的加权曼哈顿距离之和,即: $\text{minimize}(E(C))$.

如式(2)第一项所示,可优化的传输延时项本质上分为两部分:拥塞相关的等待时间和距离相关的固有传输时间.固有传输时间由 $h_{i,j}$ 决定,等待时间则是由 T_w 和 $h_{i,j}$ 之积决定,优化通信能耗(式(3))时已经考虑了 $h_{i,j}$,而 T_w 还未纳入优化范围. T_w 对延时影响很大,其优化空间可观(无拥塞时 $T_w = 0$;严重拥塞时 T_w 很大).根据 3.2 节阐述的思想,我们通过平衡链路负载来达到优化延时的目的.优化时以链路负载方差(Variance)作为链路负载失衡指标,其定义如式(4)所示.

$$\text{VAR}(L) = \sum_{i=1}^M [Load(l_i) - Load(l)_{avg}]^2 / M \quad (4)$$

式中, l_i 为 NoC 中第 i 条链路, M 为链路总数, $Load(l_i)$ 为链路 l_i 的负载量, $Load(l)_{avg}$ 为平均链路负载量.方差表征分布的离散程度,方差越大分布越不均匀,因此通信延时的优化目标转化为最小化链路负载方差,即: $\text{minimize}(\text{VAR}(L))$.

由上可知,优化通信能耗需要优化 $E(C)$,而优化延时则需要同时优化 $\text{VAR}(L)$ 和 $E(C)$,其中以 $\text{VAR}(L)$ 为主导.统一成本函数定义如下:

$$\text{cost} = \lambda \times E(C) + (1 - \lambda) \times \text{VAR}(L) \quad (5)$$

式中, λ 是比例系数,用于调节通信能耗和延时在成本函数中的比重,取值范围为: $0 \sim 1$.于是,本文 NoC 映射方法的优化目标即为: $\text{minimize}(\text{cost})$. $\lambda = 1$ 时,表示通信能耗优化, $\lambda = 0$ 时,表示通信延时优化, λ 取 $0 \sim 1$ 中间某值时,优化体现出通信能耗和通信延时的折衷.

4 映射问题

NoC 映射就是将各 IP 核按照一定的优化规则分配到 NoC 平台上,实现特定应用,并且使目标成本最小化.为了清晰地描述映射问题,假设任务和 IP 核一一对应,即:每个任务只能由一个核完成,而每个核只完成一个任务.这样,核通信图就完全等价于经典的应用特征图(Application Characteristic Graph, APCG)了,而 NoC 映射问题也就抽象成 APCG 到 NoC 结构特征图(Architecture Characteristic Graph, ARCG)之间的映射问题.

4.1 问题定义

首先对 APCG 和 ARCG 进行定义:

定义 1 给定应用特征图 $G1(V, E)$ 为有向非循环加权图,图中顶点 $v_i \in V$,表示一个执行特定任务的 IP 核;有向弧 $e_{i,j} \in E$,表示 v_i 与 v_j 之间的通信关系,其权重 $w_{i,j}$ 表示 v_i 与 v_j 之间的通信量:以数据包(packet)为单位计数通信量; $b_{i,j}$ 为 v_i 与 v_j 之间通信带宽要求.

定义 2 给定 NoC 结构特征图 $G2(R, P)$ 为有向图,图中顶点 $r_i \in R$,表示 NoC 中的一个资源节点;有向

弧 $p_{i,j} \in P$ 表示从 r_i 到 r_j 的路径,由节点间的链路串接而成; $h_{i,j}$ 表示从 r_i 到 r_j 之间的曼哈顿距离; $B_{i,j}$ 为路径 $p_{i,j}$ 能提供的最大通信带宽.

记 $G1$ 到 $G2$ 的映射函数为 $\text{map}()$,根据以上定义和假设,映射的通信能耗为:

$$E(C) = \sum_{i=1}^N \sum_{j=1}^N w_{i,j} \times h_{\text{map}(v_i), \text{map}(v_j)} \quad (6)$$

其中, $\text{map}(v_i), \text{map}(v_j) \in R$.

链路负载可以通过式(7)计算:

$$Load(l_k) = \sum_{i=1}^N \sum_{j=1}^N w_{i,j} \times \text{pass}_{i,j}^k \quad (7)$$

式中,

$$\text{pass}_{i,j}^k = \begin{cases} 1, & \text{when } p_{i,j} \text{ pass by } l_k \\ 0, & \text{else} \end{cases} \quad (8)$$

平均链路负载为:

$$Load(l)_{avg} = \sum_{i=1}^M Load(l_i) / M \quad (9)$$

由式(4)~(9)可得到目标函数 cost 的具体数学表达式.

面向能耗和延时的 NoC 映射问题定义为:给定 $G1$ 和 $G2$,寻找 $G1 \rightarrow G2$ 的映射函数 $\text{map}()$,优化目标为: $\text{minimize}(\text{cost})$;约束条件为式(10)~(13):

$$\forall v_i \in V \Rightarrow \text{map}(v_i) \in R \quad (10)$$

$$\forall v_i \neq v_j \Leftrightarrow \text{map}(v_i) \neq \text{map}(v_j) \quad (11)$$

$$\text{size}(G1) \leq \text{size}(G2) \quad (12)$$

$$\forall b_{i,j} \leq B_{i,j} \quad (13)$$

其中,式(10)、(11)用以确保 IP 核与资源节点一一对应,而式(12)、(13)用以保证规模和带宽要求.

5 面向能耗和延时的映射方法

将 4.1 节中定义的面向能耗和延时的 NoC 映射问题用蚁群算法来实现,即为本文提出的面向能耗和延时的映射方法.

5.1 蚁群算法

自 1994 年 V. Maniezzo、A. Colorini 和 M. Dorigo 首次应用 AS (Ant System) 算法来解决二次分配问题以来^[12,13],很多研究人员都在此基础上提出了改进算法,相关介绍可参考文献[14].NoC 映射属于二次分配问题,本文采用 AS 算法,为了抑制搜索过早陷入局部最优解,引入了 MMAS 算法中采用的信息素上下限.整个算法包括三个主要步骤:(1)初始化;(2)解的构造;(3)信息素更新.

5.1.1 初始化

基本参数定义与设置见表 1.参数值在文献[15]的基础上根据任务特征进行调整.

$\eta_{i,j}$ 称为先验知识或能见度,是指 IP 核 v_j 分配到资源节点 r_i 上的启发信息,它与信息素强度一起确定分配

概率. 启发信息对于算法的收敛速度有重要影响, 本文对此慎重考虑. 在此, 先定义两个函数 $center()$ 和 $vip()$:

$$center(i) = \sum_{j=1}^N h_{i,j} \quad (14)$$

$$vip(i) = \sum_{j=1}^N w_{i,j} + \sum_{j=1}^N w_{j,i} \quad (15)$$

表 1 参数定义及初始化

参数	定义	设置
M	蚁群中蚂蚁的数量	N (等于 NoC 的规模)
α	路径上信息量的重要程度	1
β	启发因子的重要程度	3.5
ρ	信息量随时间推移而衰减的程度	0.7
$\tau(0)$	信息素强度的初始值	1
$BestSolution(0)$	最佳方案成本的初始值	10000000000 (∞)

$center(i)$ 表示 r_i 在 NoC 中的居中程度, 代表通信能力, 其值越小通信能力越强; $vip(i)$ 代表 v_i 在应用特征图中的重要程度, 其值越大越重要. 显然, 一个优化的映射总是希望把最重要的 IP 核分配到通信能力最强的资源节点上. 因此, 定义启发信息如式(16).

$$\eta_{i,j} = vip(j) / center(i) \quad (16)$$

$\eta_{i,j}$ 体现出将 IP 核 v_j 分配给资源节点 r_i 的合理程度.

5.1.2 解的构造

$p_{i,j}^k(t)$ 表示在第 t 次循环时蚂蚁 k 将 v_j 分配到 r_i 上的概率:

$$p_{i,j}^k(t) = \begin{cases} \frac{[\tau_{i,j}(t)]^\alpha \times [\eta_{i,j}]^\beta}{\sum_{j \in tabu_k} [\tau_{i,j}(t)]^\alpha \times [\eta_{i,j}]^\beta}, & j \notin tabu_k \\ 0, & j \in tabu_k \end{cases} \quad (17)$$

$tabu_k (k = 1, 2, \dots, M)$ 用于记录蚂蚁 k 已经分配过的 IP 核, 称为禁忌表. $\tau_{i,j}(t)$ 表示在第 t 次循环时, v_j 到 r_i 这条分配路径上的信息素强度.

求解步骤如下:

(1) 从可选核集合 V 中按概率 $p_{i,j}^k(t)$ 选择一个核分配到 r_1 上, 并将该核添加到 $tabu$ 中;

(2) 按概率 $p_{i,j}^k(t)$ 选择一个未分配的核分配到 r_2 上, 再将该核添加到 $tabu$ 中;

(3) 执行 N 步, 直到所有的核都分配到相应的资源上, $tabu$ 满.

一次循环中, 上述过程每只蚂蚁都要执行一次, 得到 M 个映射问题的解.

5.1.3 信息素更新

随着进化进程的推进, 以前留下的信息素逐渐消逝, 用参数 ρ 表示信息素的持久程度 ($0 < \rho < 1$). 在所有蚂蚁完成一次循环以后, 更新信息素. 此时, 要根据式(18)对各分配路径 ($v_j \rightarrow r_i$) 上的信息量作更新.

$$\tau_{i,j}(t+1) = \rho \times \tau_{i,j}(t) + \Delta \tau_{i,j} \quad (18)$$

式中 $\Delta \tau_{i,j}$ 为信息增量, 表示为:

$$\Delta \tau_{i,j} = \sum_{k=1}^M \Delta \tau_{i,j}^k \quad (19)$$

$\Delta \tau_{i,j}^k$ 表示本次循环中蚂蚁 k 在分配路径 ($v_j \rightarrow r_i$) 上留下的信息量, 其计算公式为:

$$\Delta \tau_{i,j}^k = \begin{cases} \frac{1}{cost(k)}, & map(k) \text{ includes } (v_j \rightarrow r_i) \\ 0, & \text{else} \end{cases} \quad (20)$$

式中, $cost(k)$ 为蚂蚁 k 完成的分配方案的成本, 见 3.3 节中定义. 可见, 最优解具有最小的成本, 因此对信息量更新的贡献最大.

为了避免出现搜索停滞现象, 对信息素强度规定了上下限: T_{max} 和 T_{min} . 计算公式由(21)和式(22)给出.

$$T_{max} = \frac{1}{1 - \rho} \times \frac{1}{BestSolution} \quad (21)$$

$$T_{min} = T_{max} / 5 \quad (22)$$

式中, $BestSolution$ 为当前时刻最优解的成本.

6 实验与分析

6.1 映射算法实现

映射算法程序源代码用 C++ 编写, 在 Microsoft Visual C++ 6.0 环境下完成编译和仿真. 本文采用自编程随机产生实验所需的任务图. 实验采用三个不同规模的应用特征图. 表 2 给出了每个应用特征图的属性. 三个应用对应的 NoC 目标平台分别为 3×3 、 4×4 和 5×5 的规则二维网络结构.

表 2 应用特征图属性

	Number of tasks	Number of edges	Communication quantity
Application1	9	64	505
Application2	16	176	751
Application3	25	457	1804

本文以顺序 ($v_i \rightarrow r_i$) 映射方案作为参考方案 (Ref), 由于应用特征图是随机产生的, 因此该参考等价于随机映射. 实验中根据不同侧重的优化目标 ($\lambda = 1, 0.5, 0$) 对三种规模的应用进行了映射优化. 各映射方案的通信能耗和链路负载方差代价分别如图 3 和图 4 所示.

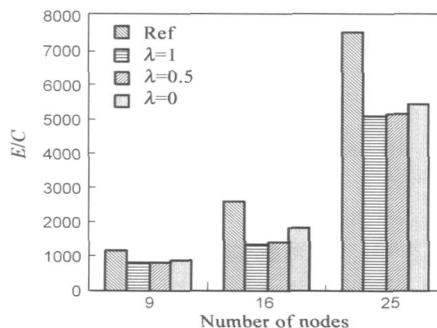


图 3 通信能耗实验结果

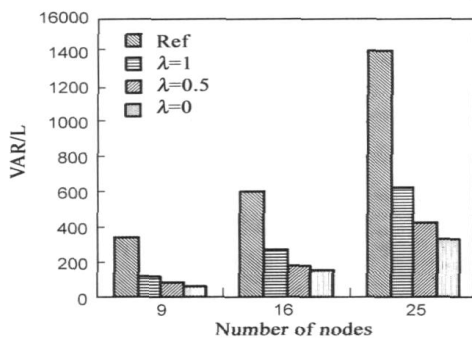


图 4 链路负载方差实验结果

由图 3 和图 4 可知,与参考方案相比, $\lambda=1$ 映射方案 $E(C)$ 和 $VAR(L)$ 分别降低 30~47 % 和 55~65 %; $\lambda=0.5$ 映射方案分别降低 30~45 % 和 70~77 %; $\lambda=0$ 映射方案分别降低 24~30 % 和 75~83 %。实验结果与算法目标一致,单一目标优化能最大限度地最小化相应指标 ($E(C)$ 或 $VAR(L)$),而联合目标优化,则能兼顾两个指标,取得不错的折衷。

实验还显示:在满足特定通信能耗要求的情况下,还有优化负载分布的空间(见图 3、图 4 中 9 节点的情况),联合目标优化恰恰能发掘这种潜力。

能耗优化将通信任务重、联系紧密的 IP 核摆在一起,大大减少了随机映射中大量通信“借道”的情况,所以使得 $VAR(L)$ 有较大的下降,但是其也会导致部分链路严重过载,由于缺乏保障负载平衡的机制,故不能进一步优化 $VAR(L)$ 。

6.2 执行时间模拟

上述实验结果表明,将链路负载方差引入目标函数可以明显地平衡链路负载分布。但是,链路负载平衡是否能够真正改善网络拥塞,降低延时,从而最终减小应用执行时间还需要进一步验证。因此,我们将 IP 核之间的通信量转换成相应数量的数据包(即: $v_i \rightarrow v_j$ 之间的通信量为 $w_{i,j}$ 个数据包),由 IP 核充当数据包的发送和接收单元,对特定映射方案的通信过程进行实时模拟。NoC 平台其它参数:缓存深度为 8,数据包长度为 3 帧,每帧宽度为 35 位(bit)。各映射方案的执行时间如图 5 所示。

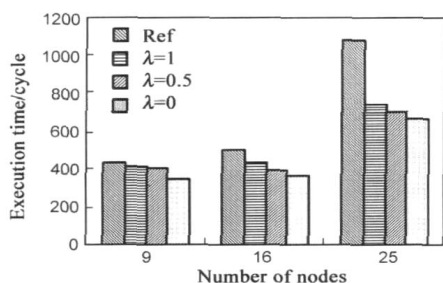


图 5 执行时间实验结果

由图 5 可知,与参考方案相比, $\lambda=1, 0.5, 0$ 三种映

射方案执行时间分别降低了 4~32 %、7~35 % 和 20~39 %。执行时间实验结果与链路负载方差保持趋势一致,印证了 3.2 节中的分析结果。链路负载分布与延时之间虽然不是简单的线性关系,但是平衡链路负载确实能缓解网络拥塞,降低延时,最终减少了应用执行时间,其效果还优于文献[9]中采用等待时间近似模型的映射方案(平均降低 20 %)。

7 结束语

本文以二维网格结构 NoC 平台为载体,首次将通信能耗与延时纳入统一的代价函数,并采用蚁群算法实现了面向能耗和延时的 NoC 映射。实验结果显示:与随机映射相比,单一目标优化映射结果在通信能耗和执行时间上分别能节省(30 %~47 %)和(20 %~39 %),而混合目标优化能取得不错的折衷效果。其中,通过优化链路负载分布间接优化延时的方法,取得了良好的效果,并且避免了 NoC 等待延时精确建模的难题;通信能耗与延时统一的优化方法,在速度支配的系统设计中,可以实现面向延时的映射方案;在能耗支配的系统设计中,可以在满足能量指标的情况下进一步挖掘映射方案时间维度的潜力。本文提出的映射方法为即将步入应用的 NoC 设计提供了有益的参考。

参考文献:

- [1] Luca Benini, Giovanni De Micheli. Network on chips: a new SoC paradigm[J]. IEEE Computer, 2002, 35(1): 70 - 78.
- [2] Axel Jantsch, Hannu Tenhunen (Editors). Networks on Chip [M]. Boston: Kluwer Academic Publishers, 2003. 3 - 18.
- [3] Dally W J, Towles B. Route Packets, Not Wires: On-Chip Interconnection Networks[A]. Proc of DAC[C]. Las Vegas, Nevada, USA: IEEE, 2001. 684 - 689.
- [4] Shashi Kumar et al. A network on chip architecture and design methodology [A]. VLSI '02 [C]. Pittsburgh, Pennsylvania, USA: IEEE, 2002. 105 - 112.
- [5] Murali S, Micheli G De. Bandwidth-constrained mapping of cores onto NoC architectures[A]. DATE '04[C]. Paris: IEEE, 2004. 896 - 901.
- [6] Hu J, Marculescu R. Energy-aware communication and task scheduling for network-on-chip architectures under real-time constraints[A]. Proc DATE '04[C]. Paris: IEEE, 2004. 234 - 239.
- [7] Nickray M, Dehyadgari M, Afzali-kusha. Power and delay optimization for network on chip [A]. ECCTD '05[C]. Cork, Ireland: IEEE, 2005. 273 - 276.
- [8] Tang Lei, Shashi Kumar. A two-step genetic algorithm for mapping task graphs to a network on chip architecture[A]. DSD '03 [C]. Antalya, Turkey: IEEE, 2003. 180 - 187.

- [9] Zhou W B, Zhang Y, Mao Z G. An application specific NoC mapping for optimized delay[A]. DTIS 2006[C]. Gammarth, Tunisia:IEEE,2006. 184 - 188.
- [10] Marcon C, Calazans N, et al. Exploring NoC mapping strategies: an energy and timing aware technique[A]. DATE '2005 [C]. Munich, Germany:IEEE,2005. 1. 502 - 507.
- [11] Ye T T, Benini L, Micheli G De. Analysis of power consumption on switch fabrics in network routers[A]. DAC '02[C]. New Orleans, LA:ACM Press,2002. 524 - 529.
- [12] Maniezzo V, Colomi A. The ant system applied to the quadratic assignment problem[J]. IEEE Tran on Knowledge and Data Engineering, 1999, 11(9): 769 - 778.
- [13] Maniezzo V, Colomi A. The ant system applied to the quadratic assignment problem[J]. IEEE Transactions on Knowledge and Data Engineering, 1999, 11(5): 769 - 778.
- [14] Stützle T, Dorigo M. ACO Algorithms for the Quadratic Assignment Problem, New Ideas in Optimization[M]. McGraw-Hill, 1999.
- [15] 吴春明, 陈治, 姜明. 蚁群算法中系统初始化及系统参数的研究[J]. 电子学报, 2006, 34(8): 1530 - 1533.
Wu Chun-ming, Chen Zhi, Jiang Ming. The research on initialization of ants system and configuration of parameters for different TSP problems in ant algorithm[J]. Acta Electronica Sinica, 2006, 34(8): 1530-1533. (in Chinese)

作者简介:



杨盛光 男, 1979 年生于浙江长兴, 南京大学博士生. 主要研究方向为片上多处理器系统、片上通信方案以及 MPSoC 仿真平台.
E-mail: yangsg@nju.edu.cn



李 丽 女, 1975 年生于山东文登, 博士, 南京大学副教授. 主要研究方向为片上多处理器系统体系结构和信息处理算法向 VLSI 的映射优化. E-mail: lili@nju.edu.cn