DOI: 10.7652/xjtuxb201608009

采用集簇方法的片上网络动态映射算法

戴启华，刘勤让，沈剑良，孙淼，吴凤阳
（国家数字交换系统工程技术研究中心，450002，郑州）

摘要：针对片上网络(NoC)传统一一对应映射关系造成的资源节点利用率不高和通信功耗大等缺陷进行了改进，提出了一种采用集簇方法的 NoC 动态映射算法(DMA)。首先利用分枝界定算法完成通信量大且相连任务节点的集簇，减小了任务图通信总量；然后在此基础上借助自适应粒子群算法完成最优映射结果的获取；最后利用动态迁移策略对最优映射结果中单独占用资源节点的任务节点进行集簇。仿真实验表明，与随机映射、动态螺旋映射算法和最优邻居算法相比，DMA 算法的通信功耗分别下降了 73.93%、46.37%和 14.55%，NoC 面积占用率分别下降了 50%、50%和 33.3%。

关键词：片上网络；动态映射；粒子群算法；动态迁移；集簇
中图分类号：TN409 **文献标志码**：A **文章编号**：0253-987X(2016)08-0052-07

A Dynamic Mapping Algorithm for Network-on-Chip Based on Cluster

DAI Qihua, LIU Qinrang, SHEN Jianliang, SUN Miao, WU Fengyang
(China National Digital Switching System Engineering & Technological R&D Center, Zhengzhou 450002, China)

Abstract: A dynamic mapping algorithm (DMA) for network-on-chip (NoC) based on cluster is proposed to solve the problems that the traditional one-to-one mapping relationship causes low utilization ratio of resource nodes and high communication power. Firstly, connected task nodes with high traffic load are clustered using a branch-and-bound approach to reduce the total communication volume of task graphs. Then an adaptive particle swarm algorithm is used to obtain the optimal mapping results. Finally, the dynamic migration strategy is used to cluster the task nodes that occupy an individual resource node in the optimal mapping results. Experimental results and comparisons with the random mapping, the dynamic spiral mapping algorithm (DSMA) and the nearest neighbor (NN) algorithm show that the proposed algorithm respectively reduces the communication power of the task graph by about 73.93%, 46.37% and 14.55%, and the occupancy rate of the NoC area by about 50%, 50% and 33.3%.

Keywords: network-on-chip; dynamic mapping; particle swarm algorithm; dynamic migration; cluster

随着晶体管尺寸的不断缩小，单个芯片上集成的元器件数量越来越多[1-3]。基于总线架构的片上系统(System-on-Chip，SoC)受制于时钟频率和有限的地址空间，芯片性能提升代价很大[4-5]。片上网络(Network-on-Chip，NoC)的出现极大地缓解了 SoC 的困境，有效地支持了 IP 核之间高效可靠、高吞吐量、低功耗的通信[6]。

合理有效的映射算法能够使系统整体性能得到

收稿日期：2015-12-18。 作者简介：戴启华(1990—)，男，硕士生；刘勤让(通信作者)，男，研究员。 基金项目：国家自然科学基金资助项目(61572520)。
网络出版时间：2016-05-17 网络出版地址：http://www.cnki.net/kcms/detail/61.1069.T.20160517.1725.004.html

提升，降低 NoC 运行中的代价成本，如功耗、面积需求等等。如何在给定设计约束的基础上，将已知通信关系的任务节点分配到特征结构图（architecture characteristic graph，ARCG）中合适的资源节点上，并且使目标函数最小化，已成为 NoC 领域需要解决的首要问题。NoC 映射过程分为静态映射和动态映射[7]：静态映射要求在给定任务图、拓扑结构和带宽等约束条件下，任务节点到资源节点映射过程中所有的通信特征（如资源节点之间数据传输延迟，通信量等）不会变，所有的映射结果不会改变，也不能进行迁移；动态映射指的是任务节点与 ARCG 上资源节点的对应关系可以随着通信需求或者 ARCG 资源占用情况进行改变。然而，NoC 任务映射是一个 NP 问题。为了更有效地利用资源节点，很多学者对 NoC 动态映射都给予了关注。针对任务节点之间的通信量不同，文献[8]采用 DSMA 将通信量最大的任务节点分配至 NoC 映射平台中间节点，以螺旋形的顺序依次分配其余任务节点直至完成映射。文献[9-10]采用最优邻居（NN）策略将具有高通信量的任务节点映射到同一个或者相近的资源节点上，以此减小任务节点之间的跳变数，减小运行功耗。文献[11-12]结合 NoC 运行过程中任务节点之间的通信请求和资源节点之间的链路负载来进行动态映射。上述算法均存在评价函数单一的缺陷，动态映射结果在功耗、面积和链路负载上仍有可继续优化的空间。

本文针对 NoC 上传统一一对应映射关系造成的资源节点利用率不高和任务图通信功耗大等缺陷进行了改进，提出了一种基于集簇的 NoC 动态映射算法——DMA。在任务图映射过程中，利用分枝界定算法将通信量大的父子节点划分到一个簇中，减小了任务图通信总量；利用自适应粒子群算法获取局部集簇后任务图的最优映射结果；获取最优映射结果中单独占用一个资源节点的任务节点集合 S 和优先迁移集合 S 内通信量大的任务节点，利用评价函数判断迁移是否成功。实验表明：与随机映射、最优邻居和动态螺旋映射（DSMA）算法相比，本文所提算法可以有效地减小通信功耗和 NoC 映射平台面积需求，提高资源节点利用率。

1　问题定义和模型描述

1.1　问题定义

本文主要研究 NoC 局部集簇动态映射策略。在介绍问题定义和模型描述之前，提出 2 个假设并作出说明。

假设 1　资源节点最多能够运行 2 个任务节点。

假设 2　物理链路带宽远远超过任务节点之间的通信需求。

本文采用资源节点为异构体的 NoC 映射平台，根据文献[9]可知资源节点在系统约束下可以运行多个任务节点。本文中将每个资源节点的任务节点最高承载量设置为 2，既保证簇的映射可能性，又保证簇的集合不会太大，超过资源节点的处理能力。

物理链路带宽不足引起的链路竞争会影响最后的功耗计算，而链路拥塞造成的功耗损失没有确切的数学模型，无法获取其较为准确的功耗值。为减小客观因素对映射结果造成的影响，设定物理链路带宽远超任意任务节点之间的通信带宽，满足簇与簇之间的通信需求。

在上文 2 个假设下，给出如下定义。

定义 1　任务图（task graph，TG）是一个非循环有向图 $G_{\mathrm{TG}}(n,w)$，顶点 $n_i \in N$ 表示一个任务节点，有向弧 $w_{i,j} \in W$ 表示任务节点 i、j 之间的通信量。以实际通信任务图 MPEG-4 为例，其 TG 如图 1 所示，任务节点之间连线上的数字即为 $w_{i,j}$。

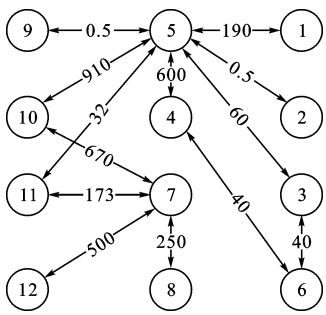


图 1　MPEG-4 任务图

定义 2　NoC 映射平台可以看成是一个特征结构图 $G_{\mathrm{ARCG}}(p,h)$，由资源节点、路由节点、网络接口和资源节点的物理链路组成。其中顶点 $p_i \in P$ 表示 NoC 规则映射平台上的一个资源节点，有向弧 h_{p_i,p_j} 为节点 p_i、p_j 间的物理链路，其值为资源节点 p_i、p_j 之间的曼哈顿距离。NoC 是一个参数化，二维网格结构的模型，假设其处于一个 xy 坐标轴中，1 号路由节点处于坐标原点，则 h_{p_i,p_j} 大小为 $|x_i-x_j|+|y_i-y_j|$。资源节点编号自下往上、从左往右均依次增大，如图 2 所示。

定义 3　TG 映射结果用 1 个一维数组表示，数组元素值为对应任务节点占用的资源节点编号。任务图的映射结果如图 3 所示，1 号任务节点占用 9 号资源

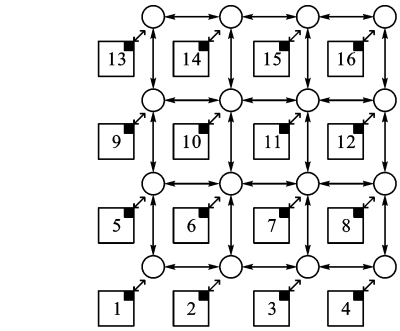
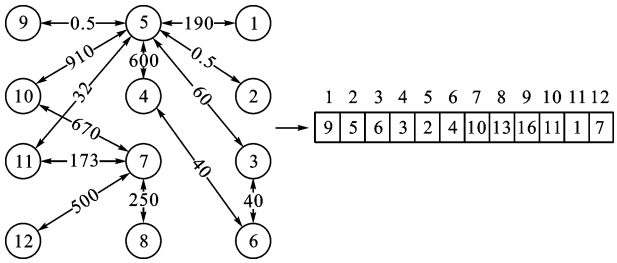


图 2 NoC 4×4 特征结构图

节点，2 号任务节点占用 5 号资源节点。任务图通信矩阵为 $\boldsymbol{C}$，大小为 $N\times N$，如式(1)所示。



(a)任务图 (b)映射结果

图 3 任务图及映射结果

$$\boldsymbol{C}=\begin{bmatrix}0&0&0&0&190&0&0&0&0&0&0&0\\0&0&0&0&0.5&0&0&0&0&0&0&0\\0&0&0&0&60&40&0&0&0&0&0&0\\0&0&0&0&600&40&0&0&0&0&0&0\\190&0.5&60&600&0&0&0&0&0.5&910&32&0\\0&0&40&40&0&0&0&0&0&0&0&0\\0&0&0&0&0&0&0&250&0&670&173&500\\0&0&0&0&0&0&250&0&0&0&0&0\\0&0&0&0&0.5&0&0&0&0&0&0&0\\0&0&0&0&910&0&670&0&0&0&0&0\\0&0&0&0&32&0&173&0&0&0&0&0\\0&0&0&0&0&0&500&0&0&0&0&0\end{bmatrix}\tag{1}$$

1.2 模型描述

(1)功耗模型。由文献[13-14]可知，单位比特数据从资源节点 p_i 传输到 p_j 所消耗的能量为

$$E_{\mathrm{b}}^{p_i,p_j}=(h_{p_i,p_j}+1)E_{\mathrm{S}}+h_{p_i,p_j}E_{\mathrm{L}}\tag{2}$$

式中：E_{S} 为单位比特数据在路由结构所消耗的能量；E_{L} 为单位比特数据在物理链路上传输所消耗的能量。TG 任务节点数为 M，映射结果所消耗的总能量为

$$E_N=\sum_{i=1}^{M}\sum_{j=1}^{M}W_{n_i,n_j}E_{\mathrm{b}}^{p_i,p_j}\tag{3}$$

式中：W_{n_i,n_j} 为任务节点 n_i 到 n_j 之间的通信量；p_i、p_j 分别是任务节点 n_i、n_j 在 ARCG 上的映射资源节点。

(2)链路负载模型。链路负载是 ARCG 物理链路承载的通信量，合理的链路负载可以有效提高链路带宽的利用率，减小数据传输中因链路竞争所产生的拥塞概率，故将链路负载方差作为评价映射结果好坏的一个指标，其计算公式为

$$Q=\sum_{i=1}^{L}(Q_i-(\sum_{j=1}^{L}Q_j)/L)^2/L\tag{4}$$

式中：L 为链路总数；Q_i 为链路 i 的负载量；$(\sum_{j=1}^{L}Q_j)/L$ 为平均链路负载量。

(3)面积模型。相同任务图在更小面积的 NoC 映射平台上成功运行，表明该映射平台上资源节点利用率更高。任务图的面积模型可以用已映射资源节点的多少进行表征，故将映射结果资源节点需求量作为衡量映射结果优劣的一个指标，其计算公式为

$$N_{\mathrm{R}}=N_{\mathrm{T}}+N_{\mathrm{C}}\tag{5}$$

式中：N_{R} 为已映射资源节点数；N_{T} 为单个任务节点映射对应的资源节点数；N_{C} 为任务节点簇映射对应的资源节点数。

2 算法描述

2.1 局部集簇

在任务图与 NoC 映射平台产生映射关系之前，首先对任务图进行预处理，采用分枝界定算法，按照任务节点之间的通信量关系和连接关系将各任务节点划分到各层上，具体流程如下。

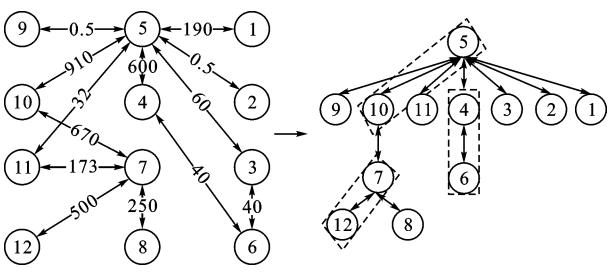
步骤 1 计算 TG 各任务节点通信量，找到通信量最大的任务节点 A 作为根(父)节点。

步骤 2　选择与上层父节点相连的任务节点作为子节点，子节点可能与多个父节点相连，选择与其相连且通信量最大的上层节点作为父节点。如任务节点 7 同时和父节点 10 和 11 相连，由于 $\boldsymbol{C}_{10,7}>\boldsymbol{C}_{11,7}$，故选择任务节点 10 作为节点 7 的父节点。

步骤 3　重复步骤 2，直至所有任务节点都划分完毕。

根据上文假设可得，每个资源节点可运行 2 个任务节点。为提高资源节点利用率，需要对部分任务节点进行局部集簇。从根节点向下寻找任务节点，由于每个节点都可能连接多个任务节点，需要找出每一层中具有最大通信量的父节点和子节点，将两者划分到一个簇中。如图 4b 所示，节点 5 是多个分枝的父节点，对比多个与其相连子节点的通信量，节点 5 和节点 10 的通信量最大，故将两者划分到一个簇中。第 1 层根节点已经集簇，以第 2 层任务节点为父节点，寻找通信量最大的父子节点划分到一个簇中，直到第 2 层所有任务节点都遍寻完毕。依次类推，直至找到最后一层，局部集簇结果如图 4b 虚线框所示。集簇造成任务节点数量及连接关系改变，通信矩阵 $\boldsymbol{C}$ 也相应改变为

$$\boldsymbol{C}=\begin{bmatrix}
0 & 0 & 0 & 0 & 190 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 40 & 60 & 0 & 0 & 0 & 0\\
0 & 0 & 40 & 0 & 600 & 0 & 0 & 0 & 0\\
190 & 0.5 & 60 & 600 & 0 & 670 & 0 & 0.5 & 32\\
0 & 0 & 0 & 0 & 670 & 0 & 250 & 0 & 173\\
0 & 0 & 0 & 0 & 0 & 250 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0.5 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 32 & 173 & 0 & 0 & 0
\end{bmatrix} \tag{6}$$



(a)任务图　　(b)集簇结果

图 4　分枝界定流程图

2.2　映射结果获取

利用自适应粒子群算法获取预处理任务图在 NoC 映射平台上的最优映射结果。粒子群算法（particle swarm optimization，PSO）是一种群体智能的优化算法，最早由 Kennedy 和 Eberhart 在 1995 年提出[15]，快速的收敛速度和简单易实现的特点使其迅速成为解决 NoC 映射问题的一个重要方法。标准粒子群算法无法随着解空间的缩小进而动态调整收敛速度，易陷入早熟陷阱。本文构建了一种自适应粒子群算法，根据迭代次数的增加动态调整粒子速度的惯性权重值，消除了寻优过程中粒子速度惯性分量的不良影响，减少了算法的迭代次数。

自适应粒子群算法利用粒子自身历史最优位置 z_i 和群体最优位置 g_i 来更新下一代粒子的速度和位置。用 t 表示算法的迭代次数，粒子根据下式完成速度和位置的更新

$$V_i^{t+1}=\omega V_i^t+c_1r_1(z_i-P_i^t)+c_2r_2(g_i-P_i^t) \tag{7}$$

$$P_i^{t+1}=P_i^t+V_i^{t+1} \tag{8}$$

$$\omega=\omega_s-(\omega_s-\omega_e)(t/T_m)^2 \tag{9}$$

式中：c_1、c_2 为非负常数，称为加速度因子；r_1、r_2 是分布在[0,1]之间的随机数；ω 为惯性权重，表示当前粒子继承先前粒子速度的能力；ω_s 为初始惯性权重；ω_e 为迭代次数最大时的惯性权重；T_m 为最大迭代次数。ω 随着迭代次数变化而变化，算法前期 ω 较大，维持算法较好的全局搜索能力；算法中期，ω 变化较快，粒子迅速靠近最优解区域；算法后期，ω 较小且变化缓慢，极大的提高了算法的局部寻优能力，从而取得了很好的求解效果。算法主要包括以下步骤。

步骤 1　相关参数设置，粒子速度和位置初始化，并且计算粒子适应值。

步骤 2　根据式(7)(8)更新粒子速度和位置，并计算适应值。

步骤 3　对比新粒子适应值，更新 z 和 g。在连续 50 次算法迭代中，z 未更改，则最优解已寻到并显示结果，否则转步骤 2。算法最大迭代次数为 200。

预处理任务图映射结果如图 5 所示。

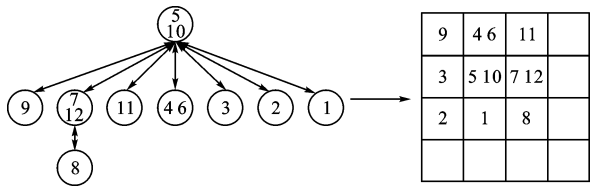


图 5　预处理任务图映射结果

2.3　映射迁移

由预处理任务图映射结果可以看到，任务节点 1、2、3、8、9、11 都单独占用一个资源节点，可能造成硬件资源的浪费。借助任务节点通信量 W_{t_i,t_j} 和对

应映射资源节点的曼哈顿距离 h_{p_i,p_j}，构建迁移评价函数 $R=\sum_{j=1}^{N}h_{p_i,p_j}W_{t_i,t_j}$，将未集簇且通信量高的任务节点迁移到通信量低的任务节点上，对比迁移过后评价函数大小来判断迁移是否成功。动态迁移主要步骤如下。

步骤 1 找到自适应粒子群算法最优映射结果中一一对应映射的任务节点集合 S。

步骤 2 在 S 中找到通信量最大的任务节点，依次迁移到集合 S 中剩余的任务节点映射资源节点上。

步骤 3 若迁移过后的 R 小于等于迁移之前的 R，迁移成立。同一个任务节点若有多个迁移成立，选取 R 最小的作为本次迁移路径。迁移成功，S 集合减去集簇成功的 2 个任务节点；迁移失败，S 集合减去该任务节点。重复步骤 2、步骤 3，直至迁移结束。

迁移结果如图 6 所示，任务节点 8、1、2 进行了映射迁移，任务节点 811、13、29 分别映射到一个资源节点上。自适应粒子群算法映射结果占用 9 个资源节点，迁移过后的映射结果占用 6 个资源节点，大大减小了资源节点需求量，既减小了映射结果对于 NoC 映射平台的面积需求，同时也提高了资源节点的利用率。

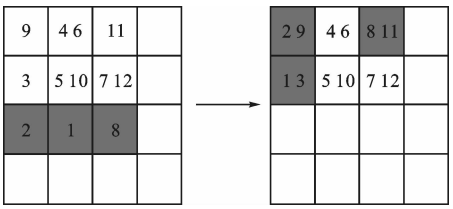


图 6 任务节点迁移

利用自适应粒子群算法获取 MPEG-4 原任务图和局部集簇后任务图映射结果，并与动态迁移结果进行对比，如表 1 所示。可以看到，动态迁移操作相对于原任务图和局部集簇后映射结果在通信量、资源节点需求量和跳变数方面都有不同程度的减小。

表 1 自适应粒子群算法对 3 种任务图映射结果对比

任务图类型	通信总量	资源节点需求量	跳变数
原任务图	6 932	12	32.0
局部集簇后任务图	4 032	9	26.7
动态迁移后任务图	4 032	6	20.7

3 实验与分析

为了验证上文中的结论，本文的映射算法程序源代码用 C++编写，在 Linux 环境下进行编译和仿真。为体现算法的实用性，将实际任务通信图 MPEG-4 作为测试样本，实验基于 4×4 NoC 系统级模型展开，其特性如 1.1 节所述，同时借助 1.2 节所提模型获取映射结果相关测试参数。由于采用启发式智能算法，实验结果不可避免地有一定的随机性，以下数据均为 3 次实验结果的平均值。

图 7 为自适应粒子群算法迭代过程中惯性权重 ω 的变化过程。可以明显地看到，在算法的起始阶段 ω 较大，确保了算法前期较强的全局搜索能力，收敛速度快，求解精度低；在算法的过渡阶段，惯性权重 ω 随着迭代次数的增加，逐渐减小，表明粒子向全局最优区域靠近，空间距离越近，ω 越小；在算法后期，惯性权重值 ω 减小，速度变缓，在 0.2～0.3 趋于一个较小的稳定值。此时算法全局搜索能力弱，收敛速度变慢，局部搜索能力强，利于算法跳出局部最优而求得全局最优。图 8 为自适应粒子群算法映射结果适应度变化过程，以功耗值为适应度，适应度越小，映射结果越好。可以看到，自适应粒子群算法映射结果适应度值收敛速度快，下降趋势明显，在较小的迭代次数里就能找到最优映射，算法性能较好。图 9 是自适应粒子群算法运算过程中迁移评价函数 R 的变化过程。对比图 8 和图 9 可以发现，适应度和 R 变化保持一致，R 越小映射结果越好，任务节点在 NoC 映射平台上布局越合理，这也证明了 R 作为任务节点迁移评价函数的有效性。

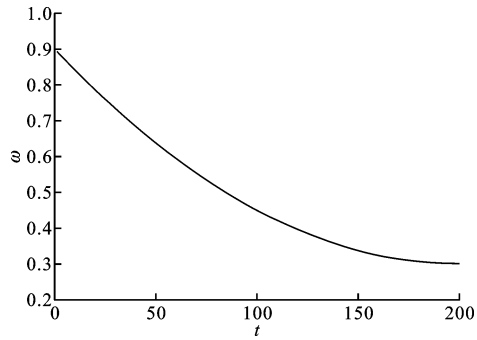


图 7 自适应粒子群算法惯性权重 ω 的变化过程

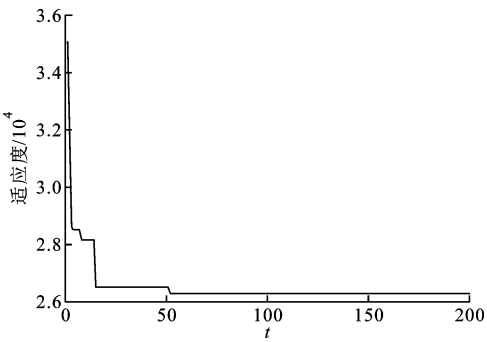


图 8 自适应粒子群算法映射结果适应度的变化过程

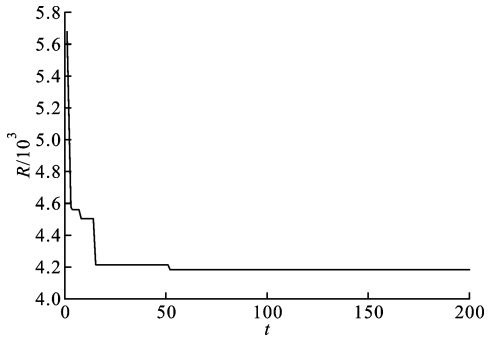


图 9　迁移评价函数 R 的变化过程

定义 θ 为各算法映射结果占随机映射结果比例。图 10 为上述 4 种算法在功耗值、映射面积和链路负载方差的仿真结果，其中 DSMA、NN 算法的映射结果是根据原文献算法的描述，在同一实验环境下自行建模获取的。

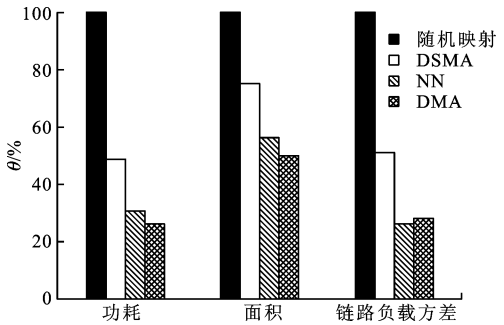


图 10　4 种算法仿真结果对比

(1)功耗代价。NN 和 DMA 算法在映射策略中允许通信量大且连接的任务节点能够集簇并映射到同一个资源节点上。在 NoC 映射平台上运行的时候，同一个簇内的任务节点通信不需要通过交换开关和资源节点之间物理链路，大大减小了任务运行时消耗的功耗，故功耗下降优势比较明显。DMA 算法相对于 NN 算法，更深层次地考虑了无通信关系任务节点之间的集簇可能，构建 R 评价函数，扩大了任务节点集簇范围，且自适应粒子群算法寻优效果较好，进一步减小了通信功耗。相对于随机映射、DSMA 和 NN 映射算法，本文 DMA 算法映射后通信功耗下降了 73.93%、46.37%和 14.55%，本文 DMA 算法在功耗代价上优势明显。

(2)面积代价。DSMA 采用一一对应的映射关系，相同的任务图运行比 NN、DMA 算法需要更多的资源节点。集簇策略使得单个资源节点利用率提高，减小了资源节点的需求量。相对于 Random、DSMA 和 NN 映射算法，DMA 算法映射后资源节点需求量减少了 50%、50%和 33.3%。

(3)链路负载方差。DSMA、NN 和 DMA 算法链路负载方差明显小于随机映射，其中 NN 链路负载方差最小。NN 映射策略减小了通信量大且相连的任务节点对物理链路的需求量，集簇的任务节点不需要物理链路为载体来传递数据信息，直接在资源节点中进行了交互和处理。DMA 算法利用 R 评价函数将无连接关系的任务节点映射到同一个资源节点上，增大了某条物理链路上通信负载。相对于集簇之前的链路负载方差，集簇之后链路负载方差可能会增大或减小。DMA 与 NN 算法相比，MPEG-4 链路负载方差提升了 6.93%。由上文假设可知，物理链路带宽满足簇与簇之间的通信需求，物理链路负载的增加不会造成物理链路竞争堵塞。

4　结　论

针对现存映射算法大都是基于一一对应的静态映射，无法根据资源节点利用情况而对映射关系进行动态的调整，本文提出了一种基于集簇的 NoC 动态映射算法——DMA，完成了任务图与 NoC 映射平台资源节点新型映射关系的设计，减小了 NoC 映射面积需求和通信功耗输出，优化了硬件资源配置。本文算法利用分枝界定算法完成通信量大且相连任务节点的集簇，接着利用自适应粒子群算法完成最优映射结果的获取，最后利用动态迁移策略完成最优映射结果中单独占用资源节点的任务节点的集簇。与随机映射、DSMA 和 NN 算法相比，DMA 算法虽然在链路负载方差考量上稍有欠缺，但是在功耗和 NoC 面积占用率上都有较大的提升。综合考虑来看，DMA 算法的优化优势明显，为 NoC 映射高性能实现提供了新的数学模型和努力方向。

参考文献：

[1] BENINI L, MICHELI G D. Networks on chips: a new SoC paradigm [J]. IEEE Transactions on Computer, 2002, 35(1): 70-78.

[2] KUMAR S, JANTSCH A, SOININEN J P, et al. A network on chip architecture and design methodology [C]// Proceedings of IEEE Computer Society Annual Symposium on VLSI. Los Alamitos, CA, USA: IEEE Computer Society, 2002: 117-124.

[3] BJERREGAARD T, MAHADEVAN S. A survey of research and practices of network-on-chip [J]. ACM Computing Surveys, 2006, 38(1): 1-51.

[4] 杨盛光，李丽，高明伦，等. 面向能耗和延时的 NoC 映射方法 [J]. 电子学报，2008，36(5)：937-942.

YANG Shengguang, LI Li, GAO Minglun, et al. An energy- and delay-aware mapping method of NoC [J]. Acta Electronica Sinica, 2008, 36(5): 937-942.

[5] 白海. 片上网络映射算法研究和设计 [D]. 成都：电子科技大学, 2009.

[6] MAQSOOD T, ALI S, MALIK S U R, et al. Dynamic task mapping for network-on-chip based systems [J]. Journal of Systems Architecture, 2015, 61(7): 293-306.

[7] 杨鹏飞，王泉. 片上网络异构多核系统任务调度与映射 [J]. 西安交通大学学报, 2015, 49(6): 72-76.
YANG Pengfei, WANG Quan. An effective scheduling and mapping algorithm of tasks for heterogeneous NoC based MPSoC [J]. Journal of Xi'an Jiaotong University, 2015, 49(6): 72-76.

[8] MEHRAN A, KHADEMZADEH A, SAEIDI S. DSM: a heuristic dynamic spiral mapping algorithm for network on chip [J]. IEICE Electronics Express, 2008, 5(13): 464-471.

[9] CARVALHO E, CALAZANS N, MORAES F. Heuristics for dynamic task mapping in NoC-based heterogeneous MPSoCs [C]// Proceedings of 18th IEEE/IFIP International Workshop on Rapid System Prototyping. Piscataway, NJ, SUA: IEEE, 2007: 34-40.

[10] SINGH A K, SRIKANTHAN T, KUMAR A, et al. Communication-aware heuristics for run-time task mapping on NoC-based MPSoC platforms [J]. Journal of Systems Architecture, 2010, 56(7): 242-255.

[11] FALAHATI H, KOOHI S, HESSABI S. Application-based dynamic reconfiguration in optical network-on-chip [J]. Computers & Electrical Engineering, 2015, 45: 417-429.

[12] CARVALHO E, CALAZANS N, MORAES F. Dynamic task mapping for MPSoCs [J]. IEEE Design and Test of Computers, 2010, 27(5): 26-35.

[13] YE T T, MICHELI G D, BENINI L. Analysis of power consumption on switch fabrics in network routers [C]// Proceedings of the 39th Annual Design Automation Conference. Piscataway, NJ, USA: IEEE, 2002: 524-529.

[14] HU J, MARCULESCU R. Energy- and performance-aware mapping for regular NoC architectures [J]. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2005, 24(4): 551-562.

[15] BETHART R, KENNEDY J. A new optimizer using particle swarm theory [C]// Proceeding of the 6th International Symposium on Micro Machine and Human Science. Piscataway, NJ, USA: IEEE, 1995: 39-43.

[本刊相关文献链接]

马亚燕，王保云. 采用博弈论的认知无线网络主用户和不可信任次用户的协作策略. 2016，50(2)：61-67. [doi：10.7652/xjtuxb201602011]

杨鹏飞，王泉. 片上网络异构多核系统任务调度与映射. 2015，49(6)：72-76. [doi：10.7652/xjtuxb201506012]

任向隆，安建峰，高德远，等. 低功耗片上网络映射的遗传及蚂蚁融合算法. 2012，46(8)：65-70. [doi：10.7652/xjtuxb201208012]

唐轶轩，吴俊敏，陈国良，等. 并行片上网络仿真器 ParaNSim 的设计及性能分析. 2012，46(2)：24-30. [doi：10.7652/xjtuxb201202005]

张瑶，张鸿，李梁，等. 时钟数据恢复电路中的线性相位插值器. 2016，50(2)：48-54. [doi：10.7652/xjtuxb201602009]

张尚伟，曾平，罗雪梅，等. 具有细节补偿和色彩恢复的多尺度 Retinex 色调映射算法. 2012，46(4)：32-37. [doi：10.7652/xjtuxb201204006]

魏倩，蔡远利. 一种基于神经网络的中制导改进算法. 2016，50(7)：125-130. [doi：10.7652/xjtuxb201607019]

姜涛，黄伟，王安麟. 多路阀阀芯节流槽拓扑结构组合的神经网络模型. 2016，50(6)：36-41. [doi：10.7652/xjtuxb201606006]

孙立远，周亚东，管晓宏. 利用信息传播特性的中文网络新词发现方法. 2015，49(12)：59-64. [doi：10.7652/xjtuxb201512010]

王安麟，孟庆华，曹岩，等. 液力变矩器的叶片数神经网络模型. 2015，49(7)：11-16. [doi：10.7652/xjtuxb201507003]

王丽霞，曲桦，赵季红，等. 软件定义网络中应用二值粒子群优化的控制器部署策略. 2015，49(6)：67-71. [doi：10.7652/xjtuxb201506011]

吴杰，冯祖仁，刘恒，等. 水下目标多元声传感阵列网络定位方法. 2015，49(4)：40-45. [doi：10.7652/xjtuxb201504007]

孟宪佳，马建峰，王一川，等. 面向社交网络中多背景的信任评估模型. 2015，49(4)：73-77. [doi：10.7652/xjtuxb201504012]

董恩清，刘伟，宋洋. 采用三角形节点块处理无线传感器网络节点定位中节点翻转歧义的迭代方法 2015，49(4)：84-90. [doi：10.7652/xjtuxb201504014]

李季碧，邓科，任智，等. 机会网络中高效低时延多摆渡节点路由算法. 2015，49(4)：91-97. [doi：10.7652/xjtuxb201504015]

李刘强，桂小林，安健，等. 采用模糊层次聚类的社会网络重叠社区检测算法. 2015，49(2)：6-13. [doi：10.7652/xjtuxb201502002]

李季碧，李宾，任智，等. 自适应动态功率控制的机会网络节能高效路由算法. 2014，48(12)：49-56. [doi：10.7652/xjtuxb201412008]

（编辑 刘杨）