

面向实时数字信号系统关键链路延时的 NoC 映射方法研究

陈庚生 陈亦欧 胡剑浩

(电子科技大学通信抗干扰技术国家级重点实验室 成都 610054)

摘要: 该文在面向功耗优化的经典 NoC 设计平台和映射算法基础上, 针对实时数字信号处理电路固有的实时性特征, 提出了一种新的面向最小化系统关键链路延时的 NoC 自主映射模型 MM-Map。该模型在满足处理单元处理容限和链路带宽的约束下, 采用基本遗传算法完成延时目标的优化求解。实验结果表明, 该模型能节约一定硬件资源的消耗, 得到近似全局最优延时解, 映射过程简单, 收敛效果好。

关键词: 数字信号处理; 关键链路延时; NoC 映射; 遗传算法

中图分类号: TN911.72

文献标识码: A

文章编号: 1009-5896(2010)07-1638-06

DOI: 10.3724/SP.J.1146.2009.00806

A Novel Critical Delay-Aware Mapping Method for Real-Time Digital Signal Systems with NoC Platform

Chen Geng-sheng Chen Yi-ou Hu Jian-hao

(National Key Laboratory of Science and Technology on Communications,
the University of Electronic Science and Technology of China, Chengdu 610054, China)

Abstract: Based on energy-aware mapping algorithm for NoC regular architectures, a new method, MM-Map is proposed, which can automatically map a circuit of real-time digital signal processing system onto the NoC platform with the minimum critical delay for the system in this paper. The MM-Map platform uses generic algorithm to optimize the delay-aware objective which can satisfy the specified constrains such as limitation of processing units and link bandwidth of NoC. The simulation results show that the new method can reduce the hardware cost and achieve an approximate global optimum with a simple procedure and good convergence.

Key words: Digital signal processing; Critical delay; NoC mapping; Generic Algorithm(GA)

1 引言

SoC 作为下一代集成电路主流设计技术近年来得到飞速发展, 作为复杂 SoC 片上数据传递与交换技术之一的片上网络(Network-on-Chip, NoC)也得到日益广泛的关注和重视^[1]。NoC 的核心思想是将计算机网络技术移植到芯片设计中来, 从而实现片上多处理器(multi-processor on chip)高效并行运算。由此产生 NoC 设计的两大核心内容: NoC 平台架构研究和基于 NoC 平台的映射方法研究。关于 NoC 平台的架构描述由文献[2]给出, 分为 3 种: Hard 平台, Firm 平台和 Soft 平台。而基于 NoC 平台的映射过程一般要经历从任务图到 IP 核的调度和从 IP 核到 NoC 结构的映射两个步骤^[1]。过去的大部分研究^[1,3,4]采用了 Firm 平台, 文献[2,3]完成了从 IP 核到 NoC 结构的映射过程, 而文献[4]完成了任务图到 IP 核和 IP 核到 NoC 结构的两步映射。映射的优化

目标通常是最小化通信成本^[5]或者功耗^[6,7]; 不同的映射结果, 对于系统的执行时间、通信时延、通信能耗等性能有着重要的影响。

本文从实时数字信号处理的角度出发, 考虑较大规模通信系统和通信信号处理过程在 NoC 架构下的实时运行情况, 提出的映射模型能够满足数字信号处理实时性的要求, 即满足数据到达率和数据处理时延等基本约束。本文采用基于 Hard 平台的 NoC 同构(isomorphic)处理单元平台, 如图1所示, 即 NoC 结构上所有的处理单元、通信单元和接口都被提前设计和完成分配, 且全部处理单元属于同一类型(如 DSP)。每个 NoC 块(Tile)代表一个具有一定处理能力的处理器、交换节点、缓存等器件的集合, Tile 之间仍然在 NoC 网络层上进行相互通信。由于对 NoC 处理单元的同构简化, 略去了 IP 核的到 NoC 结构的映射过程; 同时为了能够更准确地表示数字信号处理过程, 我们利用数据流图(Data Flow Graph, DFG)代替传统的任务图(Task Graph, TG), 完成数据流图到 NoC 结构的多到多映射(MM-Map, Multi-to-Multi Mapping)。映射过程中, 在满足一定带宽、

2009-05-25 收到, 2010-03-31 改回

国家 863 计划项目(2007AA01Z291)和教育部博士点新教师基金(200806141015)资助课题

通信作者: 陈庚生 taryc@163.com

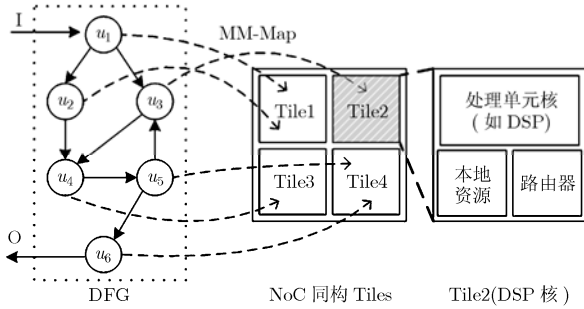


图1 NoC同构处理单元平台

处理器可用资源和处理能力门限的基本约束下,使用遗传算法配合足够的离线运算操作,高效求出近似全局最小的关键链路延时及其相应的映射方案。从而,保证映射后的NoC平台能够处理实时到达的信号,同时信号从输入到输出的延时也得以保证。仿真实验表明,本文提出的映射模型能节约一定硬件资源的消耗,并得到近似全局最优延时解;且映射过程简单,收敛效果好。

接下来,本文首先从映射平台,映射数学模型和映射算法 3 个方面介绍面向实时数字信号系统关键链路延时的 NoC 映射方法,然后给出仿真实例,最后对该映射方法作出评价和展望。

2 映射平台

本节从数据流图, NoC 以及 NoC 映射 3 个基本概念的描述开始,简单介绍数字信号系统到 NoC 的映射过程及其关键技术。

2.1 数据流图(DFG)

定义 1 数据流图 $DG = (U, A)$, 节点数 $M = |U|$, 直连链路数 $IN = |A|$ 。数据流图是由数据节点 $u_i (\forall u_i \in U, i = 1, 2, \dots, M)$ 和有向边 $a_k = a_{i,j} (\forall a_k \in A, k = 1, 2, \dots, IN)$ 组成的能实时表征数字信号处理过程的赋权图。每个数据节点表示基本运算单元, 或基本功能单元, 或一个子任务; 每条有向边则表示相应的数据通路。 $a_{i,j}$ 表示存在从 u_i 到 u_j 的直连链路, 即 u_i 和 u_j 为相邻节点。数据节点的权值 $ut_i (i = 1, 2, \dots, M)$ 表征节点 u_i 运算时间大小, 有向边的权值 $at_k (k = 1, 2, \dots, IN)$ 表征通道的传输延时大小。

对于任意一个数字信号处理系统, 都可以建立与之对应的 DFG, 本文研究的映射方法是建立在一个给定数据流图的基础上, 因此如何从实际系统得到相应的 DFG 这个过程和方法不在本文的论述范围之内。

2.2 片上网络(NoC)

由于 2 维 Mesh NoC 具有结构简单、应用广泛等特点, 因此本文只针对 2 维 Mesh 同构处理单元平台结构进行分析讨论。但是本文的研究结果可以

方便地应用到其它 NoC 拓扑结构上。2 维 Mesh 拓扑结构的 NoC 以其带宽宽、布局布线简单、时钟周期短和可扩展性强等优点^[8]成为 NoC 研究领域内最常用的一种结构, 如图 2 所示。

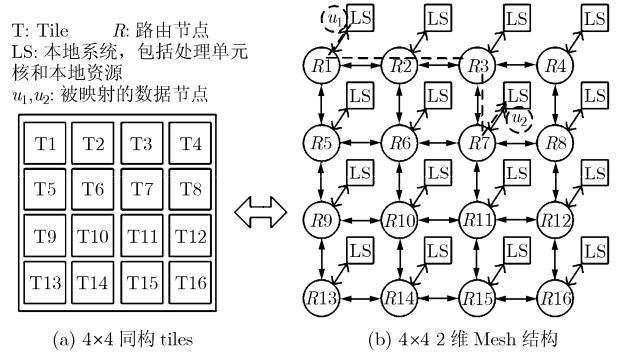


图 2 2 维 4 × 4 Mesh NoC 结构

图 2 给出了具有 16 个 Tile 的二维 4 × 4 Mesh NoC 结构。路由节点(router)是 NoC 网络结构的重要部件, 根据本地路由信息表, 通过内部接口与处理单元进行数据交互和资源的访问, 通过外部接口与其他路由节点完成全局网络的数据传输。当数据流图映射到 NoC 以后, 每个处理单元上的数据节点就是通过路由节点来实现与其他数据节点间的实时通信的。如图 2, 假设数据节点 u_1 和 u_2 分别被映射到 T1 和 T7, 则数据从 u_1 到 u_2 可能经过的所有路由节点有 R1, R2, R3 和 R7。

2.3 NoC 多到多映射(MM-Map)

传统的 NoC 映射问题一般归结为将各 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。从 IP 核到 NoC 平台的映射过程为同一规模映射, 即 IP 核数小于等于 NoC Tile 个数, 一个 IP 核必须且只能映射到一个 NoC Tile 上, 不同 IP 核分配到不同的 NoC Tile 上, 属于经典的二次分配问题, 其解空间随网络尺寸的增长呈阶乘递增。而本文提出的 MM-Map, 是数据流图到 NoC 平台的不等规模映射, 即数据节点数可能大于、等于或者小于 NoC Tile 数。一个数据节点也必须且只能映射到一个 NoC Tile 上, 但不同数据节点可以分配到相同 NoC Tile 上。这种映射的复杂度已经超越了二次分配问题。假设网络尺寸为 N , 数据流图尺寸为 M , 则其无约束解空间尺寸达 N^M 。这种映射问题类似于经典装箱问题(Bin Packing Problem)^[9], 属于 NP 难问题, 因此不能使用常规的算法, 而只能使用启发式算法寻求其局部最优解或全局近似最优解。

由于本文提出的映射算法是延时敏感的, 且在

映射过程中, 数据节点间的延时会由于网络通信延时的加入而增长, 为了反映这种实际情况, 我们根据 NoC 通信延时的计算方法提出了虚拟交换节点和映射后数据流图的概念。

定义 2 虚拟交换节点(virtual node)集 $C_v = \{c_{i,j} \mid i, j = 1, 2, \dots, M, i \neq j, a_{i,j} \in A(DG)\}$, 表示 u_i 和 u_j 由于映射到不同处理单元上而引入了附加的通信延时, $t(c_{i,j})$ 表示 $c_{i,j}$ 的延时值, 由 NoC 相关参数决定。之所以称之为虚拟节点是因为 $c_{i,j}$ 本身只存在延时值的意义, 而不是真正添加了多余的运算或路由交换节点。

定义 3 映射后数据流图 $MG = DG(U, V) \cup C_v$, 其中 DG 为原始 DFG, C_v 为虚拟交换节点集。称从数据流图到 NoC Tiles 的映射关系为 MM-Map, 记做 $MG: DG \xrightarrow{\text{MM-Map}} T$ 。每个 Tile 上所分配的数据节点集记做 $T_y = \{u_i \mid \forall u_i \in U(DG)\}$, $|T_y|$ 为第 y 个 Tile 上分配的数据节点数, 满足 $1 \leq y \leq N$ 。对应相邻 Tile 之间的每一条直连通道定义一个通道共享值 B_l , $1 \leq l \leq 2(N - \sqrt{N})$ 。 $2(N - \sqrt{N})$ 为 N (N 为平方数) 个 Tiles 的 2 维 Mesh 结构的直连通道数 (X 方向有 $\sqrt{N}(\sqrt{N} - 1)$ 条, Y 方向上有 $(\sqrt{N} - 1)\sqrt{N}$ 条), B_l 的值由映射后各相邻数据节点间链路经过通道 l 的次数之和确定。 B_l 越大代表通道 l 使用频率越高, 即发生冲突和带宽过大的机会也越高。

这里采用图 1 的 6 节点 DFG 和 2×2 NoC 作为例子, 映射的一种可能结果(mapping result)如图 3 所示。

M-DFG 是原 DFG 的映射结果, 在原 DFG 的基础上添加了虚拟节点。被分配到同一处理器的相邻数据节点之间(如图 3(d)的 $u_1 \rightarrow u_2$, $u_5 \rightarrow u_6$)的通信延时与映射之前(如图 3(a))一样, 由 DFG 相应向边权值确定, 因此在 M-DFG 上这些相邻节点的边上不会添加虚拟节点, 相邻节点间进行处理器内通信; 而被分配到不同处理器上的相邻数据节点之间(如图 3(d)的 $u_1 \rightarrow u_3$, $u_2 \rightarrow u_4$, $u_3 \rightarrow u_4$, $u_4 \rightarrow u_5$, $u_5 \rightarrow u_3$)的通信延时则比映射前(如图 3(a))要大, 由 DFG 相应向边权值与添加的虚拟节点的延时值之和确定。 $c_{i,j}$ 代表相邻节点 u_i 和 u_j 处于两个不同的处理单元上(如图 3(d)的 $u_1 \rightarrow u_3$, u_1 分配到 P_1 上, u_3 分配到 P_2 上, 因此在 M-DFG 中 u_1 与 u_3 之间加入 $c_{1,3}$)。由图 3 可以得出, $T_1 = \{u_1, u_2\}$, $T_2 = \{u_3\}$, $T_3 = \{u_4\}$, $T_4 = \{u_5, u_6\}$; $B_1 = 2(u_1 \rightarrow u_3, u_3 \rightarrow u_4)$, $B_2 = 2(u_2 \rightarrow u_4, u_3 \rightarrow u_4)$, $B_3 = 1(u_5 \rightarrow u_3)$, $B_4 = 1(u_4 \rightarrow u_5)$ 。

3 问题描述与数学模型

定义 4 给定代表某数字信号系统的已知数据

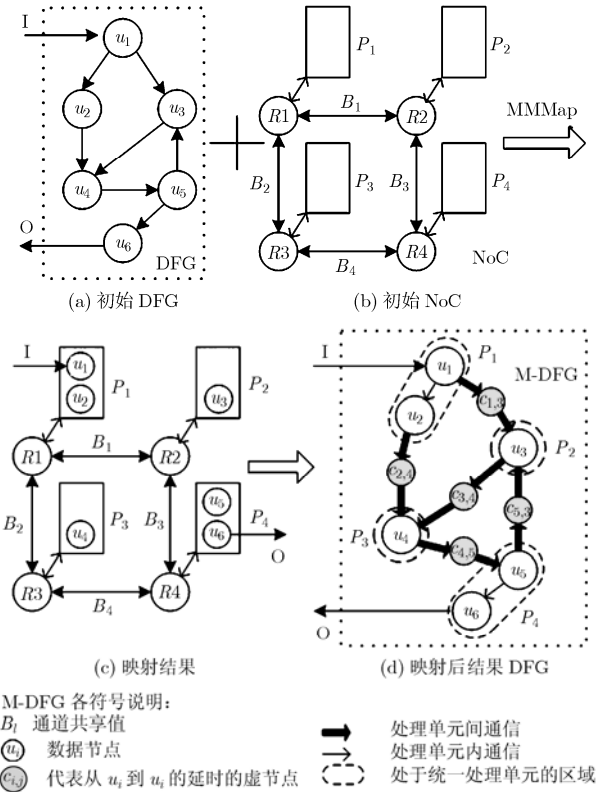


图 3 DFG 到 2×2 NoC 的映射 M-DFG

流图 $DG = (U, A)$, I 为输入节点集, O 为输出节点集, 则关键链路定义为从 I 到 O 的所有链路中延时值最大的链路, 记为 $CL_{I,O}(DG)$; IO 关键链路延时值则记为 $CLt_{I,O}(DG)$, 数学表达式见式(1)。

$$CLt_{I,O}(DG) = \max_{\substack{1 \leq x \leq L \\ L = |L_{I,O}|}} \{Lt_x(I, O)\} \\ = \max_{L_x \in L_{I,O}} \left\{ \sum_{i=1}^{|U(L_x)|} ut_i + \sum_{k=1}^{|A(L_x)|} at_k \right\} \quad (1)$$

$L_{I,O} = \{L_x \mid x = 1, 2, \dots, L\}$ 为 I 到 O 的链路集合, $L = |L_{I,O}|$ 为链路总数, 假设 $I = \{u_i \mid i = 1, 2, \dots, |I|\}$, $O = \{u_j \mid j = 1, 2, \dots, |O|\}$, 且 $\forall i, j$, 有 $i \neq j$, $|I|$ 和 $|O|$ 分别为 I 集和 O 集的节点数。当 $|I| = 1$ 且 $|O| = 1$ 时, 系统为单输入单输出结构; 反之, 为单进多出 ($|I| = 1$ 且 $|O| > 1$) 或多进单出 ($|I| > 1$ 且 $|O| = 1$) 或多进多出结构 ($|I| > 1$ 且 $|O| > 1$)。 $L_x = (U(L_x), A(L_x))$ 代表任意一条从 u_i 到 u_j 的链路, $|U(L_x)|$ 为 L_x 上节点数, $|A(L_x)|$ 为 L_x 上边数, $Lt_x(I, O)$ 为 L_x 的链路延时值。

定义 5 给定 N 个 Tiles 的 $\sqrt{N} \times \sqrt{N}$ 的 2 维 Mesh NoC, 从处理单元 P_m 到处理单元 P_n 的网络延时值, 即对应虚拟交换节点 $c_{i,j}$ 的延时值 $t(c_{i,j})$, 由 P_m 到 P_n 的交换延时 $St_{m,n}$ 决定, 数学表达式见式

(2)。

$$\left. \begin{aligned} t(c_{i,j}) &= St_{m,n} = \text{hop}_{m,n} \cdot \Delta St \\ m &= \text{map}(i), n = \text{map}(j) \end{aligned} \right\} \quad (2)$$

$\text{hop}_{m,n}$ 代表采用 XY 路由^[3]下, 从 P_m 到 P_n 要经过的路由节点的个数(如图 2, $\text{hop}_{1,7} = 4$)。 ΔSt 代表每个路由节点在正常工作状态下完成数据包处理和交换的平均时间, 一般假设 $\Delta St \sim U(t_0, t_1)$ (均匀分布), 取 $\Delta St = (t_0 + t_1)/2$ 。 $m = \text{map}(i), n = \text{map}(j)$ 表明, u_i 映射到 P_m 上, u_j 映射到 P_n 上。

定义 6 给定一种可能的映射数据流图 $\text{MG} = \text{DG} \cup C_v$, 则映射后关键链路记为 $\text{MCL}_{t_{10}}(\text{MG})$, 它的延时值 $\text{MCL}_{t_{10}}(\text{MG})$ 的数学表达式见式(3)。

$$\begin{aligned} \text{MCL}_{t_{10}}(\text{MG}) &= \max_{L_x \in L_{I,O}} \left\{ Lt_x(I, O) + \sum_{c_{i,j} \in L_x} t(c_{i,j}) \right\} \\ &= \max_{L_x \in L_{I,O}} \left\{ \sum_{\substack{u_i \in L_x \\ i=1}}^{|U(L_x)|} ut_i + \sum_{\substack{a_k \in L_x \\ k=1}}^{|A(L_x)|} at_k \right. \\ &\quad \left. + \sum_{c_{i,j} \in L_x} (\text{hop}_{\text{map}(i), \text{map}(j)} \cdot \Delta St) \right\} \quad (3) \end{aligned}$$

由上述定义可见, 每一种可能的映射方案 MG 都会对应一个关键链路延时 $\text{MCL}_{t_{10}}(\text{MG})$; 面向数字信号系统关键链路延时的映射过程, 就是为了寻求一种最优的映射方案, 使得这个映射方案对应的 $\text{MCL}_{t_{10}}(\text{MG})$ 达到最小值。所以, 本映射方法的目标函数定义如式(4)。

$$\min_{1 \leq q \leq N^M} \{ \text{MCL}_{t_{10}}(\text{MG}_q) \} \quad (4)$$

在遍历搜索的情况下式(4)的解空间尺寸达 N^M , 随着网络和数据流图尺寸的增长, 是一个典型的 NP 难问题。这里, 从实际情况出发, 对任意映射 $\text{MG}_q: \text{DG} \xrightarrow{\text{MM-Map}} T$, 首先提出本映射平台的约束条件如式(5)-式(7)。

$$\forall u_i \in U(\text{DG}), P_{\text{map}(i)} \in T \quad (5)$$

$$\forall 1 \leq y \leq N, \sum_{u_i \in T_y} ut_i \leq PT_y \quad (6)$$

$$\forall 1 \leq l \leq 2(N - \sqrt{N}), B_l \leq B_{\max} \quad (7)$$

式(5)表明每个数据节点都会且只会映射到一个处理单元上去; 式(6)指出每个处理单元都有它的处理容限值, 为 PT_y , 因此被分配到同一处理单元上的数据节点的运算时间和不能超过这个值, 这保证了映射不会为了寻求最短关键延时就将数据节点映射到尽可能少的处理单元上去。因为本文假设的数据任务量是非常大的, 一个处理单元是无法独立完成

整个系统的; 式(7)则是 NoC 的带宽约束, 每条通道上的共享值不能超过最大共享值 B_{\max} 。约束条件的存在会在求解问题过程中产生许多不可行解, 因此约束门限应该根据系统的具体特性合理设定。

4 映射算法

由前述内容可见, 对目标函数进行遍历求解是不可行的。在解决现有 NoC 映射模型的算法中比较经典的主要有文献[10]中的分支定界法(branch band), 文献[4]中的两步遗传算法(Two-step GA), 还有文献[11]中的蚁群算法(Ant System)和文献[12]的遗传算法。在诸多约束优化问题的解决方式当中, 遗传算法以其简单、易操作、需求低、并行和全局性等特点受到了普遍的关注, 成为极具潜力的新型优化方法, 本文也将采用基本遗传算法来解决 MM-Map 问题。使用遗传算法解决 MM-Map 问题的基本流程见图 4。

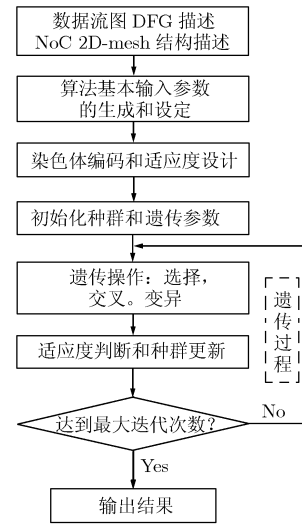


图 4 遗传算法求解 MM-Map 映射问题的流程图

一个染色体代表一种映射方案, 而适应度函数值代表染色体生存的可能性大小, 即对应着映射方案的优劣, 所以适应度函数是目标函数的函数。对于单目标最小化的目标函数, 适应度函数可以简单地设计为目标函数的倒数, 见式(8)。这样, 对应染色体的目标值越小, 其适应度值越大, 即生存并被遗传到下一代的可能性越大, 从而保持了染色体的良性遗传。

$$f = \frac{1}{\min_{1 \leq q \leq N^M} \{ \text{MCL}_{t_{10}}(\text{MG}_q) \}} \quad (8)$$

遗传操作在每次遗传过程中作用于种群, 包括选择, 交叉和变异 3 个过程。选择用于选出优良的父代, 保证遗传的良性发展; 交叉用于得到组合了

不同父代特性的子代，保证繁殖不会停滞在一个局部特性中；变异则以一定概率改变某个个体，与交叉的作用相似，但只是辅助保持新个体的产生。本法采用基于随机遍历采样的选择算子，单点交叉算子和实值变异算子完成遗传过程。

5 仿真实例

使用国家“863”重点项目“B3G-TDD 下行链路设计”中 MIMO/OFDM 接收端^[13]的系统设计为例，将该系统划分成 4 类模块(ABCD)共 32 个数据节点，得到其单输入单输出模式(以下简称 SISO)和 4 输入 4 输出模式(以下简称 4I4O)下等效数据流图，如图 5 所示，其相关延时值列于表 1 和表 2 中。

目标 SoC 平台采用 4×4 的 2 维 Mesh NoC 结

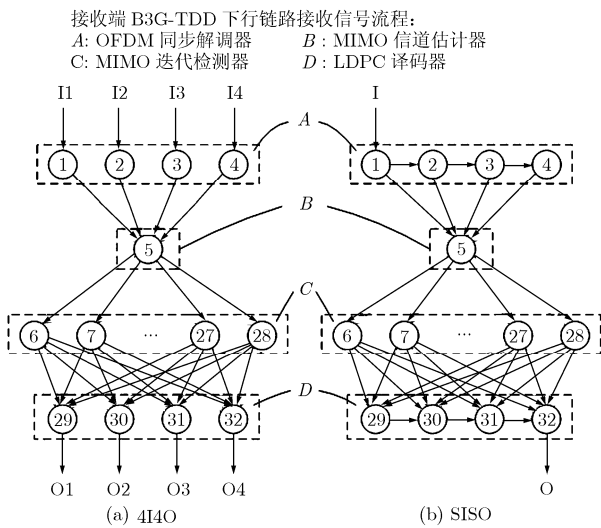


图 5 B3G-TDD 下行链路接收端等效 DFG

表 1 各类数据节点运行时间值

节点类型(节点标号)	执行时间(s)
A (1~4)	0.01011057
B (5)	0.02137344
C (6~28)	0.00632808
D (29~32)	0.01322244

表 2 两种模式下数据节点间各直连链路延时值

链路类型(节点到节点)	传输时延(s)
A to A (1 to 2~4)	0.006
A to B (1~4 to 5)	0.0020332
B to C (5 to 6~28)	0.001056
C to D (6~28 to 29~32)	0.0000528
D to D (29~31 to 32)	0.0000528

构，取 $\Delta St = 0.1 \text{ ms}$ ， $PT_y = 30 \text{ ms}$ ， $B_{\max} = 80$ 。对遗传算法的主要参数设置为：种群大小 $nind = 100$ ，代沟 $gap = 0.9$ ，交叉概率 $pc = 0.5$ ，变异概率 $pm = 0.1$ 。算法所有程序采用 Matlab 语言进行编程，外加英国谢菲尔德(Sheffield)大学开发的遗传算法工具箱 *gatbx*，仿真结果如下：

(1)分别对 SISO 和 4I4O 两种模式使用本法，得到其解收敛过程(见图 6-图 9)。

图 6 给出了在遗传算法迭代过程中 SISO 模式下目标种群均值的变化曲线和最优目标值(最优解)的收敛曲线，图 7 是图 6 最优目标值收敛曲线的纵向放大；相应地，图 8 给出了 4I4O 模式下目标种群变化曲线和最优目标值收敛曲线，图 9 是图 8 最优目标值收敛曲线的放大。可见在此次仿真实验中，SISO 模式下最优解在迭代到第 70 次左右收敛(图 7)，且最优值约为 0.0849 s；而 4I4O 模式下最优解在迭代到第 107 次左右收敛(图 9)，且最优值约为 0.0552 s。同时，由图 6 可以看出，即使目标种群均值的变化起伏比较大，但是最优值仍然有比较好的收敛特性，可见遗传算法的科学性。通过精确显示，可以得到 SISO 模式下局部最小关键延时值为 0.08491339 s，用了 14 个处理单元；4I4O 模式下则为 0.05517653 s，也用了 14 个处理单元。

(2)由于单次执行本法得到的解为局部最优解，因此，多次执行本法，然后取所有局部解的最小值的话就可以得到近似全局最优解。图 10 以 4I4O 模式为例，给出了执行 100 次本法与 100 次随机映射方法得到的两条局部解变化曲线。

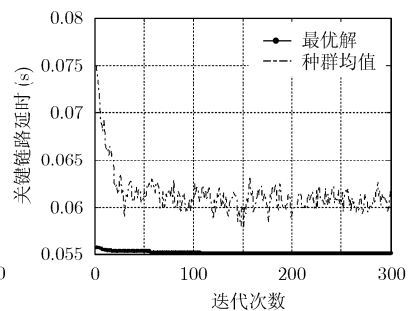
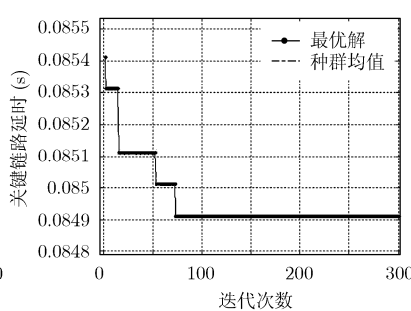
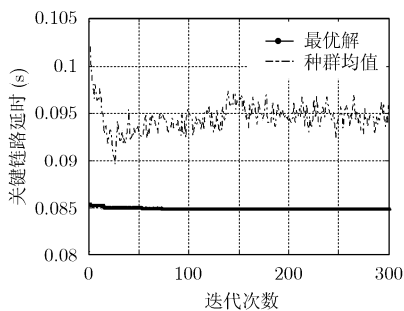


图 6 SISO 模式下最优解和种群均值收敛过程 图 7 最优解收敛过程(图 6)的局部放大 图 8 4I4O 模式下最优解和种群均值收敛过程

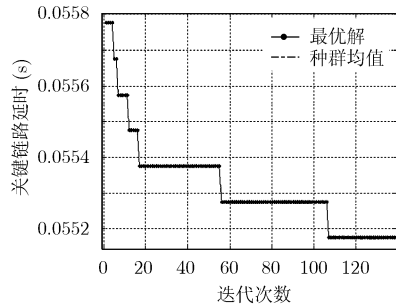


图9 最优解收敛过程(图8)的局部放大

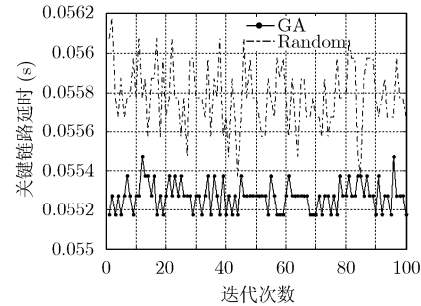


图10 遗传算法与随机映射算法局部解变化曲线对比

这里需要说明的一点是,遗传算法(GA)在每次运行之后都收敛于一个局部最优解,因此得到的解均为可行解(即满足约束的解);而对于随机(random)映射来说,每次映射结果随机产生,可能得到可行解,可能得到不可行解,而且实验证明,出现不可行解的概率要远远大于出现可行解的概率。例如,图10中显示的100随机映射样本,实际上是在接近3000次随机映射之后才提取出来的100个可行解。

显然,使用遗传算法得到的局部最优解基本都要比随机映射得到的可行解要小,图11中显示GA解的范围是0.05517653 s到0.05547653 s,而随机映射解的范围是0.05537653 s到0.05617653 s。遗传算法与随机映射算法的简单比较见表3。

表3 遗传算法与随机映射算法性能比较

算法名称	运行时间	收敛效果	解特性
遗传算法	较慢	收敛	可行解;局部最优; 多次运行可达近似 全局最优
随机映射 算法	较快	无	可行或不可行解;多 次运行可达局部最优

6 结束语

本文提出了一种面向IO关键链路延时的NoC映射模型,并采用遗传算法,对B3G-TDD下行链路接收端电路进行建模,得到了优于一般随机映射算法的最优解。该模型实现简单,并以除功耗之外的另一重要因素——延时作为优化目标,为数字电路在NoC处理单元中的布局提供了一定的参考价值。随着映射目标的增多,映射模型的可重配置性将成为一个重要的议题,该模型可以作为一种通用的映射平台,通过添加不同的约束和目标完成不同用户对电路的需求。

参考文献

- [1] Hu Jing-cao and Marculescu R. Energy- and performance-aware mapping for regular NoC architectures[J]. *IEEE Transactions on Computer-aided Design of Integrated Circuits & Systems*, 2005, 24(4): 551-562.
- [2] Hu Jing-cao and Marculescu R. Communication and task

- scheduling of application-specific network-on-chip[J]. *IEEE Processing-Computer Digital Technology*, 2005, 152(5): 643-651.
- [3] Ascia G, Catania V, and Palesi M. Multi-objective mapping for mesh-based NoC architectures[C]. CODES+ISS'04, Stockholm, Sweden, ACM, 2004: 182-187.
- [4] Lei T and Kumar S. A two-step genetic algorithm for mapping task graphs to a network on chip architecture[C]. DSD'03, Turkey, IEEE, 2003: 180-187.
- [5] Zhou Wen-biao, Zhang Yan, and Mao Zhi-gang. An application specific NoC mapping for optimized delay[C]. Design and Test of Integrated Systems in Nanoscale Technology, Sept 5-7, DTIS, 2006: 184-188.
- [6] Modarressi M and Sarbazi-Azad H. Power-aware mapping for reconfigurable NoC architectures[C]. 25th International Conference on Computer Design, ICCD, Oct 7-10, 2007: 417-422.
- [7] Nickray M, Dehyadgari M, and Afzali-Kusha A. Power and delay optimization for network on chip[C]. ECCTD'05, Cork, Ireland, IEEE, 2005: 273-276.
- [8] Gu Hai-yun, Li Chang-wen, and Sun Shu. Research on mapping algorithm of irregular mesh NoC for portable multimedia appliances[C]. IET Conference on Wireless, Mobile and Sensor Networks, 2007 (CCWMSN07), Dec 12-14, 2007: 697-700.
- [9] Cong Liu and Baskiya S. Scheduling mixed tasks with deadlines in grids using bin packing[C]. 14th IEEE International Conference on Parallel and Distributed Systems, 2008(ICPADS '08). Dec. 8-10, 2008: 229-236.
- [10] Hu Jing-cao and Marculescu R. Energy-aware mapping for tilebased NoC architectures under performance constraints[C]. ASP-DAC 2003, Japan, ACM, 2003: 233-239.
- [11] Withironprasert K, Chusanapiputt S, Nualhong D, Jantarang S, and Phoomvuthisarn S. Hybrid ant system/priority list method for unit commitment problem with operating constraints[C]. IEEE International Conference on Industrial Technology, 2009(ICIT 2009), Feb. 10-13, 2009: 1-6.
- [12] Ning Wen, Kumar N, Dechu S, and Soewito B. Mapping task graphs onto Network Processors using genetic algorithm[C]. IEEE/ACS International Conference on Computer Systems and Applications, 2008(AICCSA 2008), March 31-April 4, 2008: 481-488.
- [13] 丁庆. B3G TDD中关键技术——信道估计技术的研究及其硬件实现. [硕士学位论文], 成都: 电子科技大学, 2004: 49-50.

陈庚生: 男, 1984年生, 硕士生, 研究方向为通信与信息系统。
陈亦欧: 女, 1982年生, 博士生, 研究方向为通信与信息工程。
胡剑浩: 男, 1971年生, 教授, 博士生导师, 从事无线通信和VLSI的研究。