**Software Engineering and Applications** 软件工程与应用, 2015, 4(6), 136-147
Published Online December 2015 in Hans. http://www.hanspub.org/journal/sea
http://dx.doi.org/10.12677/sea.2015.46018

Hans汉斯

# Research and Implementation of Reconfigurable SoC Transaction Level Architectural Performance Analysis Based on Transaction Data Stream

**Minhui Hu, Sikun Li, Guanwu Wang, Kongfei Du**

College of Computer, National University of Defense Technology, Changsha Hunan
Email: huminhui94@163.com

Received: Dec. 9th, 2015; accepted: Dec. 23rd, 2015; published: Dec. 28th, 2015



## Abstract

**Coarse grained reconfigurable architecture has become an important solution for high performance reconfigurable SoC coprocessor because of its high configuration speed, fast calculation speed, good adaptability and low power consumption. However, the traditional performance analysis method of register transfer level modeling is still widely used. To overcome the shortage of traditional performance analysis in flexibility and efficiency, we propose a new method for reconfigurable SoC performance analysis based on transaction data flow graph. This method combined application data flow graph with architectural characteristics through constructing the transaction dataflow graph of reconfigurable coprocessor and labeling hardware performance parameters corresponding to functional properties. Using the breadth first search algorithm based on hierarchical search to analyze the tree-like transaction dataflow graph, we can get the performance of reconfigurable coprocessor running applications. The result of the experiment indicates that the proposed method calculates the structure of the reconfigurable SoC quite accurately, and benefits the design of architecture and design space exploration.**

## Keywords

**Reconfigurable SoC, Reconfigurable Coprocessor, Transaction Dataflow Graph, Performance Analysis Model, Search Algorithm**

文章引用: 胡敏慧, 李思昆, 王观武, 杜孔飞. 基于事务数据流的可重构 SoC 性能分析方法研究与实现[J]. 软件工程与应用, 2015, 4(6): 136-147. http://dx.doi.org/10.12677/sea.2015.46018

# 基于事务数据流的可重构SoC性能分析方法研究与实现

**胡敏慧，李思昆，王观武，杜孔飞**

国防科学技术大学计算机学院，湖南 长沙

Email: huminhui94@163.com

收稿日期：2015年12月9日；录用日期：2015年12月23日；发布日期：2015年12月28日

## 摘 要

**粗粒度可重构体系结构由于其配置速度快、计算加速比高、适应性好、低功耗特性等优点已成为高性能可重构SoC协处理器的重要解决方案。可重构SoC的性能分析，普遍使用传统的寄存器传输级代码仿真分析方法，存在使用不方便、分析效率较低等不足之处。本文提出一种基于事务数据流图的可重构SoC性能分析方法，该方法结合应用程序数据流图与体系结构特征，构建可重构协处理器的事务数据流图并标注图节点功能属性相应的硬件性能参数；然后采用基于层次搜索的广度优先性能搜索算法，通过分析类树形结构的事务数据流图，得到可重构协处理器运行应用程序的性能分析结果。实验表明所提出的方法可以较为准确高效的分析可重构SoC体系结构应用计算性能，有助于可重构SoC早期体系结构设计和设计空间探索。**

## 关键词

**可重构SoC，可重构协处理器，事务数据流图，性能分析模型，搜索算法**

## 1. 引言

由于集成电路的快速发展，芯片复杂度急剧增长，系统架构设计模块化趋势日益明显，IP 复用是片上系统(System on Chip, SoC)的明显特征，如何在 IP 复用的基础上进行架构设计成为了 SoC 体系结构设计的一个重要课题[1]。对于一种特定的应用功能需求，设计者面临种类众多的总线结构、IP 规格、编程模型的选择，单凭经验已经很难确定那种架构方案更合理；决策者也越来越趋向于采用量化评价指标来衡量一个架构设计的优劣，量化指标主要包括性能(如吞吐量、响应时间等)和代价(如面积、功耗等)两个方面，其中性能指标又尤为重要。

可重构 SoC 普遍采用传统嵌入式微处理器+可重构协处理器组成的系统结构。由于可重构协处理器具有配置速度快、计算加速比高、适应性好、低功耗特性等优点，可重构 SoC 的计算密集型应用程序几乎全部在可重构协处理器上运行，因而，可重构协处理器的计算性能决定了可重构 SoC 的计算密集型应用计算性能。判断可重构 SoC 体系结构设计方案是否满足总体性能指标，其核心依据是可重构协处理器的性能分析结果。

传统的 SoC 性能分析方法主要可以分为基于模拟的方法和基于分析的方法。Barreteau [2]，Gerstlauer [3]等都对基于模块的 SoC 框架进行了研究。在这些框架中，最为广泛研究的还是基于模拟的方法，基于模拟的方法模拟时间长，对于体系结构的建模时间也比较长。因为运行的边界节点很难定义，吞吐量和延时很难得到。现在已经有了很多基于模块的框架的数据流网络的研究，例如 Thiele [4]等提出了一种基于有环数

据流图的模块化框架的性能分析方法，其中性能分析被集成到分布式操作层(distributed opratinglayer, DOL)之中。然而在这些框架中，性能评估都是通过在不同抽象层次上进行模拟来实现的，并没有改变其基于模拟的本质。

为了减少基于模拟的方法的执行时间，一些结合模拟和分析的方法被开发出来，Le Nours [5]针对多核处理器提出一种基于同步瞬态的模拟方法，Kunzli [6]等提出了一种在模拟器中将子系统替代为对应的分析模块的方法。这些方法可以提高执行时间，但是不能解决边界点覆盖率的问题。

由于这些方法的限制，设计者往往采用寄存器传输级(Register Transfer Level, RTL)代码模拟仿真或使用现场可编程门阵列(Field Programmable Gate Array, FPGA)原型验证系统的方法来验证设计方案的性能，然而这种传统方法受限于功能的正确性，需要在通过功能仿真的前提下才能得到性能分析结果，需要编写大量 RTL 级代码和测试数据，影响性能分析效率。

事务级(Transaction Level)系统性能分析是一种在 SoC 设计的早期进行更高抽象层次的事务级体系结构建模和性能分析的方法，既可提高系统建模、模拟验证和性能分析的效率，又可使软件设计和硬件设计在更早的设计阶段协同开始。杨汉侨[7]设计了一种基于事务级模型的 SOC 芯片性能分析平台，孟昕[8]提出一种基于数据包处理的事务级 SoC 性能分析方法，但由于分析粒度没有细化到数据运算，都不适用于可重构 SoC 性能分析。

本文首先简要介绍课题组开发的应用定制可重构流水线协处理器的体系结构框架，以及典型应用程序数据流图，然后，重点论述了所提出的基于事务数据流图的可重构 SoC 性能分析方法，该方法结合应用程序数据流图与体系结构特征，构建可重构协处理器的事务数据流图并标注图节点功能属性相应的硬件性能参数；然后采用基于层次搜索的广度优先性能搜索算法，通过分析类树形结构的事务数据流图，得到可重构协处理器运行应用程序的性能分析结果。实验表明所提出的方法可以较为准确的分析可重构 SoC 体系结构应用计算性能，有助于可重构 SoC 体系结构早期设计的性能分析和设计空间探索。

## 2. 可重构 SoC 体系结构

### 2.1. 可重构 SoC 体系结构框架

可重构 SoC 普遍采用传统嵌入式微处理器+可重构协处理器组成的系统结构，如图 1 所示。

可重构 SoC 的可重构协处理器的体系结构和处理单元指令集根据领域应用需求定制，计算加速比高、适应性好、功耗低，计算密集型应用程序大都分配到可重构协处理器上运行。嵌入式微处理器主要负责 SoC 系统程序运行的总控和 I/O 设备管理，计算开销很小。计算密集型应用程序的运行性能取决于可重构协处理器的运行性能。因此，可重构 SoC 体系结构的性能分析主要是可重构协处理器的性能分析。以下重点研究可重构协处理器的性能分析方法。

### 2.2. 应用定制可重构流水线协处理器框架

嵌入式数字语音图像处理 SoC 设计中，为了减小数字语音图像处理算法计算开销，提高性能，普遍采用专用应用定制处理器硬件结构。可重构流水线(Application Customization Reconfigurable Pipelining, ACRP)协处理器，是课题组面向高性能低功耗数字语音图像处理应用需求，而提出的一种简单专用可重构协处理器结构框架[9]，其体系结构如图 2 所示。

ACRP 由三部分组成：1) 应用定制处理单元集 CPE；2) 可重构互连数据通道 RDC；3) 数据缓存部件 DBM。CPE 的指令集根据对领域应用程序的运算操作功能类型、运算操作次数等计算特征进行定制；RDC 采用由多路选择器构成的交叉开关网络，可以实现 CPE 之间任意的数据交换，并且在运行过程中可以对不同 CPE 之间的数据通路进行实时重构；DBM 采用多功能寄存器文件(Multiple Functional Register

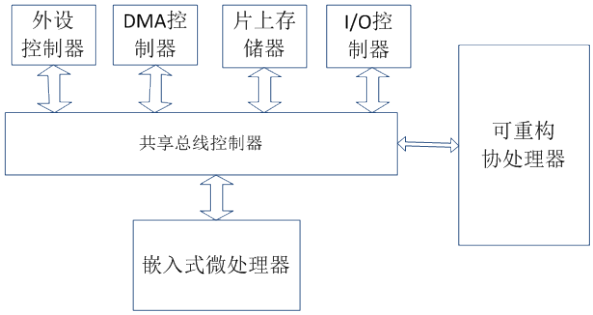


**Figure 1.** System structure of reconfigurable SoC
**图 1.** 可重构 SoC 系统结构

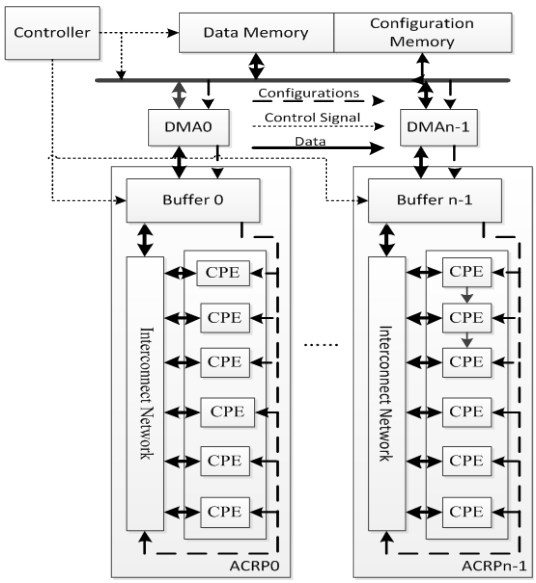


**Figure 2.** Architecture of ACRP
**图 2.** ACRP 体系结构

File, MFRF) [10]实现，为可重构流水线提供多端口数据读写功能。可重构流水线条数根据领域应用程序的并行计算特征和流水计算长度定制。

## 3. 事务数据流图定义与生成

### 3.1. 程序数据流图分析

可重构 SoC 是面向特定领域应用需求设计的，对可重构 SoC 的性能评价和性能分析不能脱离于应用程序。程序数据流图以图形方式表达了应用算法程序运行过程中，数据传输和运算操作处理功能的逻辑结构关系。既描绘了数据从节点输入移动到输出的过程中所经过的计算处理功能，又描述了数据传输的路径和方向。虽然数据流图与具体硬件实现无关，而且没有时间参数信息，但是程序数据流图准确描述了可重构 SoC 的应用程序计算处理功能、运算操作执行先后顺序和运行过程。这些要素是构建可重构 SoC 性能分析模型不可或缺的。

以数字语音增强处理领域应用为例，主流的谱减法、维纳滤波法、基于字典训练算法中，计算密集型基本算法主要是 FFT、一阶平滑滤波、矩阵乘等。矩阵乘法运算是两个矩阵的行向量与列向量的点积运算的循环，基于字典训练的算法中矩阵乘的规模为([300 × 256] × [256 × 1]，[1 × 256] × [256 × 400]，进

行一次矩阵乘法需要几百次点积运算。点积运算的数据流图如图 3 所示。由数据流图中可以看出，矩阵乘法运算的运算操作类型为乘法和加法，对缓存访问频繁。

FFT 算法中基本运算操作单元是蝶形单元，对于 256 点的基 2 FFT 需要循环执行 1024 个蝶形单元。一个蝶形运算是三个复数进行复数乘加运算，其中一个为旋转因子 $W=\cos(2\pi r/N)-j\sin(2\pi r/N)$。式中，N 为 FFT 点数，W 为 FFT 点数决定的常数。蝶形单元的数据流图如图 4(a)所示，主要运算操作类型是乘法、加法和减法。一阶平滑滤波的计算公式为 $Y[n]=aY[n-1]+(1-a)|X[n]|^2$，计算简单，但是对于每一帧数据都要循环 256 次。其数据流图如图 4(b)所示，主要运算操作类型是乘法、加法和减法。

以上三种密集型计算虽然基本运算的结构简单，但是运行过程是对这些基本运算的嵌套循环，而且循环次数相当大，尤其是乘法运算需要多次进行，而乘法运算占用相当多的运算资源，因此粗粒度可重构协处理器的基本运算单元主要进行这三种运算，结合这三种典型运算研究可重构 SoC 体系结构性能分析具有代表性。

数据流图中的数据从节点输入移动到输出的过程中数据传输的路径和方向是一种类树型结构关系，若对数据流图进行扩展，生成与可重构协处理器硬件模块计算功能和性能参数有对应关系的事务数据流图，采用合理的自动搜索统计算法，可以快速得出可重构协处理器运行该应用程序的性能参数。这是构建可重构 SoC 性能分析模型和分析方法的有效技术途径。

### 3.2. 事务数据流图的定义

在电子系统级(ESL)设计方法领域，国际 OCP-IP 标准化组织已明确论述，事务是从更高抽象层次表

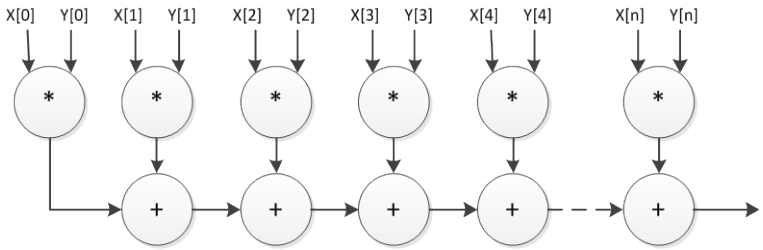


**Figure 3.** Dataflow graph of point multiply
**图 3.** 点积运算的数据流图

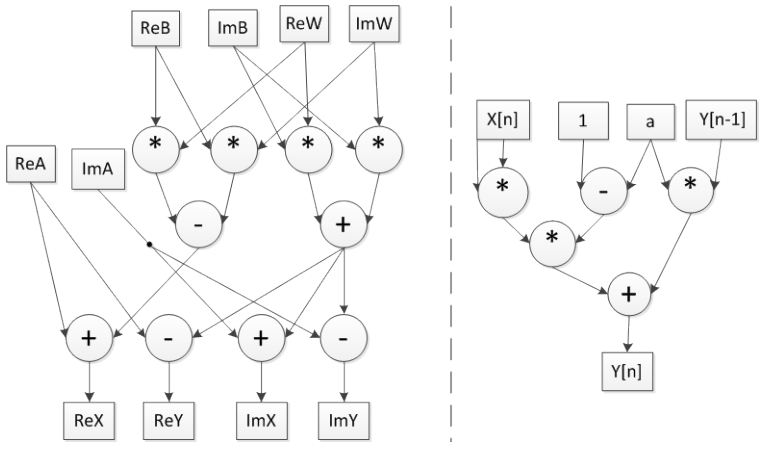


**Figure 4.** Dataflow graph of butterfly and first-order smooth filtering
**图 4.** 蝶形运算和一阶平滑滤波数据流图

示数据处理与数据通信的交换过程，SoC 事务建模的核心概念是将运算功能和通信功能分开，模块之间的通信通过通道实现。一个系统的事务模型不但和数据流图一样描述了数据在模块之间传输和运算操作处理功能的逻辑结构关系，而且还描述了通道数据通信和模块数据处理的时间性能参数信息。根据事务模型概念和特征，为了在数据流图中体现可重构 SoC 硬件模块的功能和性能，对数据流图进行如下扩展，定义新的事务数据流图。

事务数据流图(TDFG)是一个超图(hydra-graph)，是由一个超节点(Hydra-node)集合 TN、超边(Hydra-edge)集合 TE、以及超节点输入端口集合 IP 和超节点输出端口集合 OP 所组成的四元组构成，TDFG = (TN, TE, IP, OP)。

首先，将数据流图的节点扩展定义为超节点，如图 5 所示。

“n1”，“n2”和“n3”是三个超节点(hydra-node)，节点“n1”IP(n1) = {In1, In2, In3}，OP(n1) = {Out1, Out2}。节点“n3”IP(n3) = {a, b, c}，OP(n3) = {d}。超节点(hydra-node)可包含有多个运算操作功能表示这些运算操作同时进行。运算操作功能包括算术运算，逻辑运算，数据读/写操作等。

一个节点可有多个输入输出端口，为了区分不同节点的输入输出端口，节点 n 的输入或输出端口 p 可以写为 pn.p。上图中有两个被命名为“In1”的端口。为了区分，一个记为“n1. In1”，另一记为“n2. In1”。对节点 n 的端口 p，$\mathbf{p} \in \mathbf{IP(n)}$或者 $\mathbf{p} \in \mathbf{OP(n)}$，使用 Node(p)来表示 p 所属的节点，因此，Node(n. p) = n。上图中，Node(a) = n3。对输入端口的集合 $\mathbf{IP} = \{\mathbf{p1}, \mathbf{p2}, \cdots, \mathbf{pn}\}$，使用 NODE(IP)来表示节点集合，同理，对输出端口集合 $\mathbf{OP} = \{\mathbf{p1}, \mathbf{p2}, \cdots, \mathbf{pn}\}$，使用 NODE(OP)可以表述节点集合。如对于输入端集合 IP = {n1. In1, n2. In1, n3. a}，NODE(IP) = {n1, n2, n3}。

考虑到硬件实现中的计算和访存特性，进一步将超节点分为计算节点和存储节点，计算节点的功能属性赋以运算操作类型值和时间性能参数值。数据访问的性能参数对 SoC 系统的性能有重要的影响，为了描述存储模块的功能和性能，将存储节点的功能赋以功能类型值(静态存储器 RAM、动态随机存储器 SRAM、寄存器文件 RF 等)和对应的时间性能值，存储节点的输出端口赋以读延迟性能参数，输入端口赋以写延迟性能参数。计算节点与存储节点是相互独立而又紧密联系的，存储节点是计算节点的数据来源之一，也是计算节点的最终输出。

进一步扩展定义数据流图的边为超边(Hydra-edge)。超边集合 TE 是数据通路连接关系的集合，对应于数据流图来说，表示的是数据的流出方向，$\mathbf{PE} = (\mathbf{te1}, \mathbf{te2}, \cdots)$，te∈TE 每条超边表示了一条数据通路，超边的头端/尾端表示数据通路的数据流向和数据来源。超边的尾端连接超节点的一个输出端口，超边头端连接另一超节点的一个输入端口，表示建立了两个节点间的一条数据通路。超边 te = (te, He)有一个尾端 te 和非空头端集合 He：

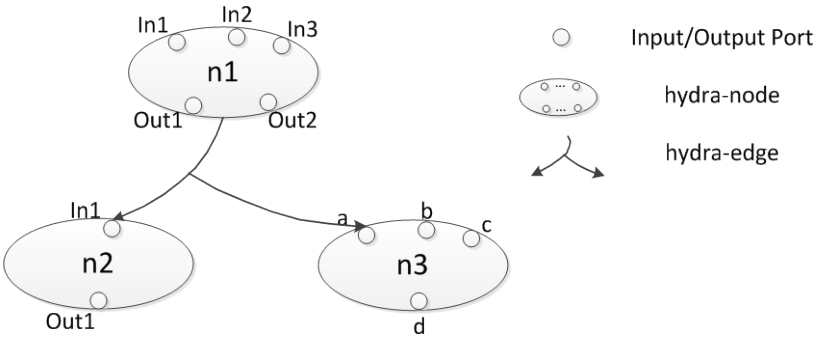


**Figure 5.** Graph of hydra-node and hydra-edge
**图 5.** 超节点超边示意图

$$t_e \in \bigcup OP(pn)$$

$$H_e \in \bigcup IP(pn)$$

超边从它的尾指向它的头。边“e”可写为(n1. Out2, {n2. In1, n3. a})，这里 te = n1. Out2 和 He = {n2. In1, n3. a}。当节点的输入端是边的头端时，边 ed 称为节点 pn 的输入边，即，存在一个端口 $\mathbf{p} \in \mathbf{He}$ 且 $\mathbf{ip} \in \mathbf{IP(pn)}$，相应的节点 n 称为尾节点。当节点的输出端是边的尾端时，边 ed 称为节点 n 的输出边，即，$\mathbf{te} \in \mathbf{OP(n)}$，相应的节点 n 称为头节点。在图 3.2 中，边“e”是“n1”的一条输出边并且是“n2”或“n3”的输入边，n1 是头节点，“n2”和“n3”是尾节点。超边的属性包括边的头端连接的端口、尾端连接的端口、边的数据通信时间参数或重构时间性能参数。超边的属性包括边的头端连接的端口、尾端连接的端口、边的数据通信延迟时间参数或重构延迟时间性能参数。

### 3.3. 事务数据流图生成

1) 基于 C++的事务数据流图生成方法

由于程序数据流图和事务数据流图描述的应用程序数据传输和运算操作处理功能的逻辑结构关系是一致的，因此由程序数据流图扩展生成事务数据流图，实质上是实现可重构协处理器硬件模块功能和性能参数到程序数据流图节点的映射。在映射中数据流图节点的记录域必须扩展，以保证准确完整表示事务数据流图超节点所具有的端口、处理功能、性能参数等属性。以蝶形运算为例，图 6 为蝶形运算数据流图映射到可重构流水线协处理器上产生的事务数据流图。图中，n0 是映射了缓存部件 MFRF 的存储节点，缓存部件有 4 个读数据端口(A B C D)和两个写数据端口(E F)。n1~n10 是映射了可重构处理单元的计算节点。

可重构 SoC 的性能参数主要有以下几个部分：

- 计算模块的计算时间(Operation Time, OT)；
- 存储部件存取时间，包括取数时间(Load Time, LT)和存数时间(Store Time, ST)；
- 可重构通信通道的配置时间(Configuration Time, CT)，配置时间包括配置信息读取时间和可重构数据通路的重构时间。

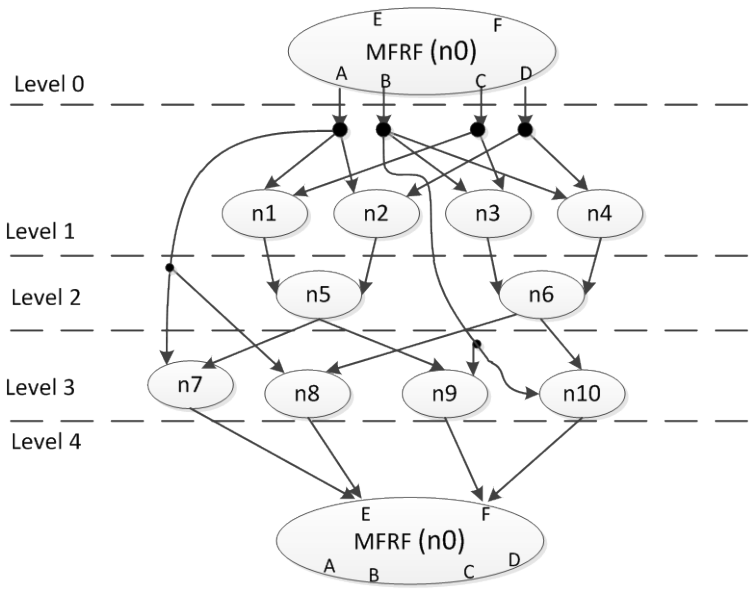


**Figure 6.** Transaction dataflow graph of butterfly extended from dataflow graph
**图 6.** 由蝶形运算数据流图扩展生成的事务数据流图

不同的可重构协处理器设计方案，对各模块功能可以有多种不同选择，通常 CPE 是同构的，但是在不同的方案中 CPE 指令集可能不同。存储部件有 MFRF、RAM、ROM 等多种选择，可重构数据通路也有开关网络、路由网络等多种实现方式。每种功能都有相应的性能参数，功能性能参数查找表如表 1 所示。

C++描述中，数据流图和事务数据流图都是有向图，都可以采用图 7 所示的邻接链表表示法表示。

在事务数据流图的邻接链表表示法中，每一个节点包含三个域：节点编号；可扩展的性能分析相关结构 PM，LS；指向头节点的指针。编号为 0 的节点是存储节点，具有一定的特殊性，节点 0 的性能分析相关结构 LS 包含了存储节点的功能：取数 Load 和存数 Store；其他节点是计算节点，计算节点相关结构 PM 中包含了超节点包含的运算的种类与个数，以及与功能对应的性能。由于运算最开始的数据来源是存储器，运算完成之后最终要将结果存在存储器中，因此存储节点 0 表现了运算流程的开始和结束。对于以存储节点为头结点的超边，对应的尾节点为运算的起始节点，如图 7 中的节点 1 和节点 2；对于以存储节点为尾结点的超边，对应的头节点为运算的终止节点，如图 7 中的节点 4 和节点 5。

事务数据流图生成方法就是从数据流图的邻接链表向事务数据流图的邻接链表映射的方法。具体是由事务数据流图设计者，采用 C++根据应用程序执行流程和功能性能参数查找表填充完事务数据流图链表的各个域，即可生成事务数据流图。

2) 基于编译前端工具的事务数据流图自动生成方法

多种 C 或 C++编译器，如 GCC、LLVM 等的前端工具具有自动生成程序数据流图的功能。在编译前端生成的程序数据流图基础上，根据事务数据流图定义和可重构 SoC 事务级体系结构特征，设计一个转

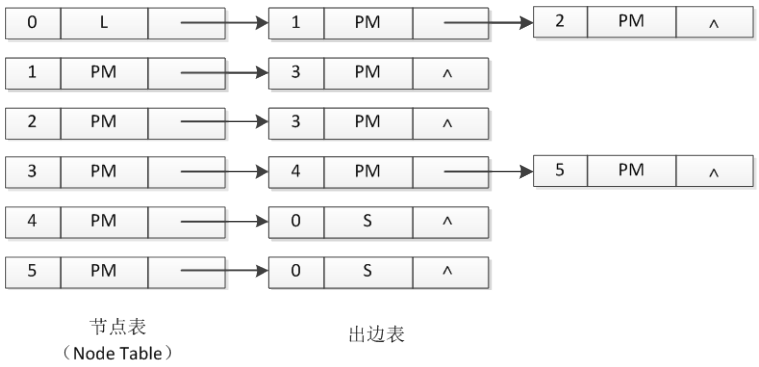


**Figure 7.** Adjacency list representation of transaction dataflow graph and dataflow graph
**图 7.** 事务数据流图和数据流图的邻接链表表示

**Table 1.** Lookup table of function and performance parameters
**表 1.** 功能性能参数查找表

| 模块名 | 功能 | 性能参数 |
|---|---|---|
| CPE0-CPE7 | 加法<br>减法<br>乘法<br>… | Add_Time<br>Sub_Time<br>Mul_Time<br>… |
| MFRF<br>RAM<br>ROM | 读数据时间<br>写数据时间 | Read_Time<br>Write_Time |
| 可重构开关网络 | 配置信息读取时间 | CR_Time |
| 可重构路由网络 | 数据通路重构时间 | DC_Time |

换工具，实现自动生成事务数据流图。目前课题组正在开展此项研究工作。

## 4. 基于事务数据流图的可重构 SoC 性能分析模型与分析算法

### 4.1. 可重构 SoC 性能分析模型构建

在生成了应用程序事务数据流图的基础上，根据可重构流水线协处理器的流水计算和并行计算等特征，对事务数据流图进行归并，即可构建出可重构 SoC 性能分析模型。归并步骤如下：

1) 根据数据流图中的数据依赖关系和数据依赖关系的传递性，将所有节点分为 N 层，同一层中的节点原则上可以并行计算。各层的节点组成 N 个集合 S1，S2…，Sn，每一个集合中包含的节点个数分别为 a1，a2，…，an；

2) 逐层对节点合并。若可重构协处理器中 CPE 个数为 M，将数据流图中同一层的计算节点分为若干个组，每组包含 M 个节点，同一组的节点归并为事务数据流图中的超节点，性能参数标注以执行时间最长的运算为准。则第 p 层节点转换后生成的超节点个数为$b_p = \lceil a_p / M \rceil$将超节点排序为 n1，n2，…，$n_{bp}$；

3) 超节点重新排列。如果硬件资源太少不足以支持两个超节点的并行，这些超节点是串行工作的。将同一层中的各超节点按照步骤(2)中的排序链接到前一节点的出边表中，层间的数据流向关系不变。

生成基于 C++的邻接链表表示的性能分析模型之后，结合 4.2 节的性能分析算法即可得出性能。

蝶形运算应用事务数据流图图 6 经过节点归并后生成的性能分析模型如下图 8 所示。

### 4.2. 基于层次遍历的性能搜索统计算法

依据以上性能分析模型，本文采用基于层次遍历的性能搜索统计算法得出体系结构性能。典型应用中循环的类型主要是 for 循环，其基本语句为 for(loop_index = low_bound; loop_index+ = step; loop_index <= up_bound){循环体}，循环开始和结束条件以及每一循环体前进的步进已知故可以得出循环次数。由图 6 可以发现由一个输出由下至上的追溯可以将基本块的事务数据流图视为类树形结构，在树中位于同一层次的计算/存储节点同时计算，因此采用广度优先的层次遍历搜索算法。

性能搜索统计算法 TimeCount()主要流程为：

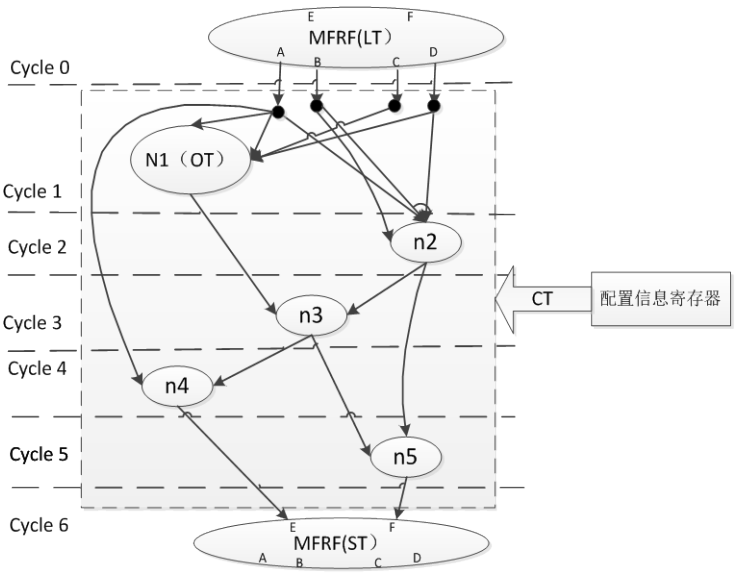


**Figure 8.** Performance model of butterfly on reconfigurable coprocessor
**图 8.** 蝶形运算的可重构协处理器性能分析模型

Input：计算/存储节点编号与前驱后继关系，各运算时钟周期数，性能参数查找表

TimeCount()

{根据输入由下至上建立性能搜索树，树的节点为事务数据流图中的计算/存储节点；

为计算节点赋予性能参数变量，表征计算类型；

建立一个数组存放已经遍历到的节点，节点以层次遍历顺序排放，不同层次节点之间放置标记位；

由节点计算类型参数和性能参数查找表得出节点运算时间；

比较同一层次各节点运算时间，选出最大值视为并行计算时间 levelTime；

累计各层 levelTime 得到基本块执行时间；

计算循环次数：(up_bound - low_bound)/step；

由基本块执行时间和循环次数得出总执行时间

}

Output：应用程序总执行时间

## 5. 实验与结果分析

为了验证本文方法的有效性和精确性，采用本文方法对 FFT、一阶平滑滤波、矩阵乘运算三种典型运算针对 ACRP 体系结构进行事务级性能建模与性能分析。同时对所分析的 ACRP 体系结构和上述应用程序进行 RTL 级建模，使用 ModelSim 软件模拟根据波形图得出硬件模拟运行时间。将两者进行比对进行验证，体系结构性能使用运行特定应用程序的时钟周期数 cycle 加以表征。实验的软件环境为 Windows 7 系统，VisualC++6.0 和 ModelSim10.1；硬件环境为 Intel CPU i5 3.20 GHz。

首先进行 RTL 级建模实验，可重构流水线协处理器的体系结构采用由 8 个基本运算单元 CPE 组成的一条可重构流水线，一个多功能寄存器文件(Multiple Function Register File, MFRF) [10]做缓存部件、可重构互连通信网络 RCN 采用交叉开关网络。其硬件模块划分结构如图 9 所示，采用 Verilog HDL 对 ACRP 的模块和系统建模。

ACRP 的工作流程由配置信息控制，配置信息主要控制：(1) CPE 运算功能；(2) CPE，MFRF,主存等各组件的数据来源和去向。设置 10 个配置信息数据 CR0-CR9，其中 CR0-CR7 都为 12 比特，分别配置 8 个 CPE 的输入端口的数据来源和功能，CR8 配置 MFRF 的两个输入端口和主存储器的数据来源，数据宽度为 12 比特；CR9 配置 MFRF 的工作模式，数据宽度为 6 比特。

在 RTL 级建模中，应用程序的运行过程隐含于配置信息之中，通过 testbench 的方式实现，总运行时间和对应的时钟周期数可以由 ModelSim 模拟生成的波形图得到。由于粗粒度可重构协处理器暂无可重构编译其支持，需要手工编写配置信息代码，工作量很大，本文采用 8 点基 2FFT 运算实验，编写配置信息代码量约两千多行。大型应用程序的配置信息很难生成。

为了使用本文提出的性能分析模型预测 ACRP 的应用程序执行时间，首先需提取相关性能参数，性能参数主要分为存取时间，计算时间和配置时间。根据 ACRP 体系结构的缓存系统特征，存数据、取数据分别需要 2 个 cycle 和一个 cycle。根据对三种典型运算的数据流图计算特征分析，计算时间考虑加法、减法和乘法的运算时间，这三种运算操作需要的时钟周期数使用 ModelSim 和 Verilog 模拟实验得出并作为基本性能参数，列于表 2 中。在性能搜索算法执行中通过查找基本性能参数表得出每个运算节点功能相应的运算时间。由于在 ACRP 体系结构中配置信息较少而且采用配置信息预取技术，以及典型应用程序不需每个时钟周期都对协处理器进行重构，在基本性能参数查找表中设置 ACRP 配置时间为 0。

针对 FFT、一阶平滑滤波、矩阵乘运算三种典型运算，采用 C++编写程序使用本文提出的可重构 SoC 事务级体系结构性能分析方法进行性能预测，与 RTL 级仿真模拟的性能实测结果比较如表 3 所示。其中

性能分析模型预测时间与 RTL 模拟运行时间都以 ACRP 体系结构运行一个完整应用程序并将输出结果储存于缓存器中所需时钟周期数表征，矩阵乘应用程序为大小分别为[16 × 256]和[256 × 16]的两个矩阵相乘。

表 3 说明使用性能分析模型预测计算时间与 RTL 模拟运行时间相比较，预测准确度较高。由于性能分析模型忽略了应用程序运行过程中的控制流的影响，因此随计算规模的增大，预测准确度略有下降。通过采用本文提出的可重构 SoC 事务级体系结构性能分析方法，性能分析不必以实现体系结构功能的正确性为前提，避免建立繁琐的 RTL 级模型，手工编写大量配置信息。可以在设计初期快速对体系结构性能进行预测，指导体系结构设计。

## 6. 结论与展望

本文针对传统的“估计－设计－仿真－原型验证”的设计与验证模式中性能分析的灵活性和效率低的不足，提出了一种基于事务数据流图的可重构 SoC 性能分析方法。实验结果表明，本文方法可以较准确分析预测所设计的可重构协处理器体系结构运行典型应用程序的运行时间，可以对可重构协处理器体系结构的不同设计方案进行性能评测，判断设计方案是否满足总体性能指标，支持可重构 SoC 高层体系结构设计优化。目前，本文方法的不足之处是面向应用程序的事务数据流图和性能参数标注需要用 C++

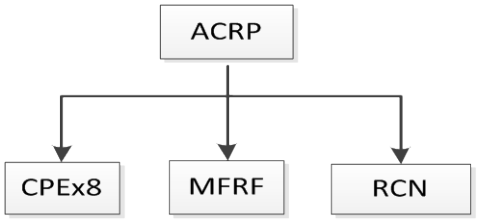


**Figure 9.** ACRP Verilog HDL modeling Parts
**图 9.** ACRP 的 Verilog HDL 建模模块划分

**Table 2.** Lookup table of basic performance parameters
**表 2.** 基本性能参数查找表

| 运算类型 | 计算时间(cycle) |
|---|---|
| 存数据 | 2 |
| 取数据 | 1 |
| 加法 | 1 |
| 减法 | 1 |
| 乘法 | 3 |
| 配置信息读取 | 0 |
| 数据通路重构 | 0 |

**Table 3.** Performance analysis result comparison: performance analysis model vs RTL modeling
**表 3.** 性能分析结果比对：性能分析模型 vs RTL 模拟

| 应用程序 | 性能分析模型<br>预测计算时间 | RTL 模拟运行时间 | 预测准确度(%) |
|---|---|---|---|
| 一阶平滑滤波 | 9 | 9 | 100 |
| 8 点基 2 FFT | 352 | 360 | 97.8 |
| 矩阵乘 | 7206 | 7424 | 97.1 |

手工生成，容易出错。下一步将研究由数据流图自动生成事务数据流图技术方法，并面向用实际计算密集型应用程序进行可重构协处理器性能分析，进一步改进完善所提出的可重构 SoC 性能分析方法。

## 基金项目

本文得到国家自然科学基金重点项目(编号：61133007)资助。

## 参考文献 (References)

[1] 郭炜. SoC 设计方法与实现[M]. 第 2 版. 北京: 电子工业出版社, 2011.

[2] Barreteau, A., Le Nours, S. and Pasquier, O. (2013) A Case Study of Simulation and Performance Evaluation of a SDR Baseband Architecture. *Journal of Signal Processing Systems*, **73**, 267-279. http://dx.doi.org/10.1007/s11265-013-0764-0

[3] Gerstlauer, A., Haubelt, C., Pimentel, A.D., *et al.* (2009) Electronic System-Level Synthesis Methodologies. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **28**, 1517-1530. http://dx.doi.org/10.1109/TCAD.2009.2026356

[4] Haid, W. and Thiele, L. (2007) Complex Task Activation Schemes in System Level Performance Analysis. 5*th IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis* (*CODES+ISSS*), Salzburg, September 30 2007-October 3 2007, 173-178.

[5] Le Nours, S., Postula, A. and Bergmann, N.W. (2014) A Dynamic Computation Method for Fast and Accurate Performance Evaluation of Multi-Core Architectures. *Design, Automation and Test in Europe Conference and Exhibition* (*DATE*), Dresden, 24-28 March 2014, 1-6.

[6] Kunzli, S., Poletti, F., Benini, L. and Thiele, L. (2006) Combining Simulation and Formal Methods for System-Level Performance Analysis. *Proceedings of the Design Automation and Test in Europe* (Vol. 1), Munich, 6-10 March 2006, 1-6. http://dx.doi.org/10.1109/DATE.2006.244109

[7] 杨汉侨. 基于事务级模型的 SOC 芯片性能分析平台的设计[D]: [硕士学位论文]. 南京: 东南大学, 2012.

[8] 孟昕, 沈海斌, 严晓浪. 基于数据流的 SoC 性能建模方法及实现[J]. 浙江大学学报: 工学版, 2011, 45(2): 314-322.

[9] Wang, G., Liu, L. and Li, S. (2014) ACRP: Application Customized Reconfigurable Pipeline. Advanced Computer Architecture. Springer Berlin Heidelberg, 16-30.

[10] 杜孔飞. 骨传导语音增强 SoC 的可重构流水线协处理器设计与实现[D]: [硕士学位论文]. 长沙: 国防科技大学, 2014.