

基于多代理系统的软硬件协同设计

陈 芸, 王遵彤, 凌 毅

(同济大学电子与信息工程学院, 上海 201804)

摘 要: 为使软硬件协同设计过程更具分布性、自主性及并行性, 在软硬件协同设计中引入多代理(MAS)技术, 提出软硬件协同设计的 MAS 模型, 包括系统描述 Agent、软硬件划分及映射 Agent、软硬件设计 Agent、协同通信 Agent、性能评估 Agent 和硬件系统测试 Agent 的构建和应用。采用多个目标代理映射、协商的方法协调整个协同设计过程。实际应用表明, 该方法能优化系统级芯片设计方案、软硬件结构和功能, 并提高系统整体性能。

关键词: 软硬件协同设计; 软硬件划分; 多代理系统

Hardware and Software Co-design Based on Multi-Agent System

CHEN Yun, WANG Zun-tong, LING Yi

(College of Electronics & Information Engineering, Tongji University, Shanghai 201804)

【Abstract】 In order to make the hardware and software co-design process more distributive, autonomous and parallel, this paper presents a novel methodology of applying Multi-Agent technology into co-design process and the Multi-Agent System(MAS) model. The model includes system description Agent, hardware and software partition and mapping Agent, hardware and software design Agent, communication Agent, performance analysis Agent and the hardware validation Agent. The MAS model describes how to construct and apply these Agents and how to use Multi-Agent mapping and cooperating algorithm to coordinate the co-design process. Application results show that this methodology works better than traditional process in system level chip design, and it can optimize the software and hardware architecture and improve the whole system performance.

【Key words】 hardware and software co-design; hardware and software partition; Multi-Agent System(MAS)

1 概述

在传统的 SoC 设计中, 一般先进行软硬件划分, 然后分别设计和实现硬件系统与软件系统, 这使得硬件和软件的开发过程缺少沟通, 不能及时进行协调优化, 导致设计效率低下, 开发周期增长。现代 SoC 设计趋于采用软硬件协同技术, 在系统设计的初始阶段, 同步进行软件设计和硬件设计, 在缩短开发周期的同时提升整体设计效果。目前, 在一些需要大量数据处理的应用中, 鉴于不同的任务可以在很大程度上相互独立运行, 可将具有内在执行并行性的复杂任务分解为一系列紧密联系的子任务, 以分布式的形式并行实现需完成的任务。

多代理系统(Multi-Agent System, MAS)起源于分布式人工智能的分布式问题求解, 体现出了人类社会的智能, 适合于开放、动态的环境^[1]。在软硬件协同设计中引入多代理技术, 用多个 Agent 分别封装各个子任务, 通过各代理间的通信、合作、协调、调度、管理及控制来表达系统的结构、功能及行为特性, 使所构造的系统具有 MAS 的特点, 从而体现出了软硬件协同设计过程中的分布性、自主性和并行性。

2 多代理系统

Agent 是一种具有主动行为能力的实体或应用程序, 能够对过程运行中的决策或控制任务进行抽象, 且具有一定的智能特性。它具有 9 个常见的特征: 适应性, 自治性, 协作行为, 推理能力, 基于语言层次的通信能力, 机动性, 拟人性, 反应性, 稳定性^[2]。

根据环境要求的不同, Agent 的结构有所不同, 常见的

主要有慎思型结构、反应型结构和混合型结构等^[3]。在目前的应用中日渐推广的是混合型结构。一个 Agent 的混合型结构如图 1 所示^[4]。它主要由传感器、效应器、信息处理、智能模块和通信器模块组成。

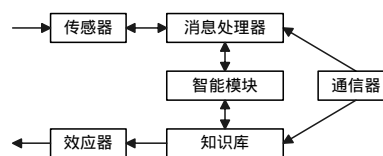


图 1 Agent 的混合型结构

MAS 是一个由多个代理协调合作形成的问题求解网络, 能够解决单个代理不能解决的复杂问题^[5]。Agent 映射是 MAS 技术研究的基本问题之一, 将所要解决的实际问题通过一定操作, 映射到 MAS 对应 Agent 的内部符号推理系统中。例如一个对象或对象之间的关系, 都应该在代理的域模型中有相应的表示。在 Agent 内部通过领域模型执行推理过程, 然后再把所得到的结果返回到实际问题中。

MAS 不是多个 Agent 之间简单的线性叠加, 而是通过各 Agent 之间的相互协作来解决复杂问题。通过构建一个描述 Agent 之间相互作用的形式化框架, 即所谓的协调模型, 可以体现各个 Agent 之间的协调机制。协调模型通常由协调体

基金项目: 国家自然科学基金资助项目(60674076)

作者简介: 陈 芸(1984 -), 女, 硕士研究生, 主研方向: 软硬件协同设计; 王遵彤, 副教授; 凌 毅, 硕士研究生

收稿日期: 2009-10-15 E-mail: yunshangcao.yun@163.com

(Entity)、协调规则(Law)、协调媒介(Medium)3个部分构成^[6]。

Agent 之间的合作在统一的协调机制上需要通过协商来完成。协商指 Agent 之间进行相互讨论, 试图改变对方的意图来维护自身的意图, 通过对各方论点进行评价, 决定是否应该改变自己的意图的一个反复过程。Agent 之间采取的协商方式取决于目标系统的结构。本文采用的是基于功能 Agent 的过程模型, 系统中的 Agent 不直接映射实现系统中的实体, 而是在每个 Agent 中封装具有局部和相对简单求解能力的功能模块, 通过 Agent 之间的相互协作来实现软硬件协同设计的过程。

3 基于 MAS 的软硬件协同设计

MAS 技术用于软硬件协同设计是指将设计任务分解为多个相对简单且易于管理的子任务, 映射为不同的 Agent: 系统描述 Agent, 划分及映射 Agent, 软硬件设计 Agent, 协同通信 Agent, 硬件系统测试 Agent 及性能评估 Agent。每个 Agent 能够作用于自身和周围的环境, 希望达到自身利益最大化, 并可与其他 Agent 通信。不同 Agent 之间进行协调、综合, 最后产生最优的设计结果。软硬件协同设计的 MAS 模型如图 2 所示。

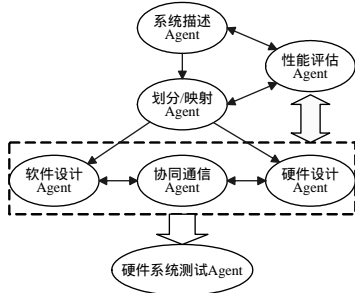


图 2 软硬件协同设计的 MAS 模型

3.1 系统描述 Agent

系统描述 Agent 通过在较高的抽象层次上描述整个系统的行为, 对系统的需求进行分析说明, 是后续设计的基础。系统描述 Agent 的结构如图 3 所示。

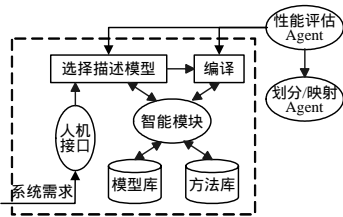


图 3 系统描述 Agent 的结构

当有一个系统需求消息到达时, 系统描述 Agent 在人机接口接收消息后对消息进行处理。消息处理主要涉及系统的描述模型和描述方法。描述模型指的是建立适合描述系统行为功能的抽象模型。目前常采用的描述模型有数据流图、有限状态自动机、Petri 网、并发通信进程、程序状态机等几大类, 包括其中针对于特定需求的变种。系统描述 Agent 将所有的描述模型封装存储于模型库中, 当需要建立描述模型时, 向智能模块发出请求, 由智能模块调用模型库中的描述模型以满足需求, 并在此基础上给出具体的描述方法, 实现编译过程。编译是指针对已选的描述模型选用文本描述语言或图形语言的形式来进行描述。编译开始之前向智能模块发出编译请求, 智能模块根据接收到的请求从方法库中选择合适的描述方法。方法库大致包括 2 种描述^[7]: (1)同构建模方法,

即用一种语言来描述整个系统; (2)异构建模方法, 对于硬件和软件采用不同的语言来描述^[8]。实际上, 智能模块相当于系统描述 Agent 内的子 Agent, 承担构建模型与确定方法 2 项任务。当有建立描述模型或编译请求时, 智能模块就动态的被分派去执行相应的任务。模型库与方法库随不断产生的新模型和新方法而得到扩展及丰富。系统描述 Agent 不涉及系统如何实现和软硬件划分的问题, 但需要通过性能评估 Agent 对整个系统的行为(包括建立描述模型和编译)进行验证, 以期在设计开始阶段就发现系统行为要求中的错误。

3.2 划分及映射 Agent

划分及映射 Agent 将确认后的系统描述分解成为与实现有关的软件部分和硬件部分, 并映射到相应的软件和硬件上, 经性能评估 Agent 反复仿真验证后, 决定如何在软硬件各种部件之间优化系统配置。划分及映射 Agent 的结构见图 4。

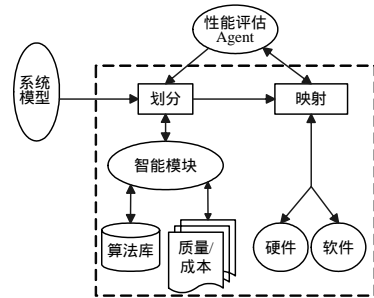


图 4 划分及映射 Agent 的结构

在软硬件划分请求到达时, 智能模块根据性能、面积、功耗、成本、重量等方面的综合要求, 在算法库中进行解的搜索, 为系统的每个功能模块节点选择一种合适的实现方式(软件或硬件)。算法库中包含启发式算法、整数线性规划、遗传算法等, 可最大限度地加快计算速度和改进计算精度。在完成软硬件划分后, 通过性能评估 Agent 对系统的性能参数进行预测, 以评估软硬件划分合理性。如果划分不合理, 就需要再次进行软硬件划分, 重新进行评估。如此反复, 直至获得最优的解决方案。

将划分的结果一一在特定的系统架构上映射, 建立系统的事物级模型, 即搭建系统的虚拟平台。在这个虚拟平台上通过性能评估 Agent 进行性能评估, 多次优化系统架构。

3.3 软硬件设计 Agent

完成软硬件划分和映射后, 进入软硬件协同综合阶段。软硬件设计 Agent 的结构如图 5 所示。

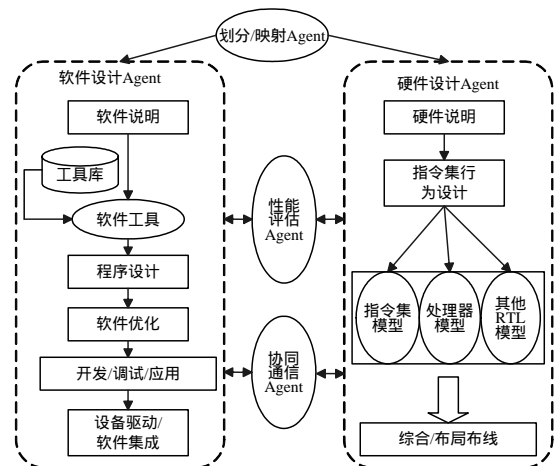


图 5 软硬件设计 Agent 的结构

软件设计 Agent 的功能主要包括任务综合以及操作系统、接口驱动和应用软件的开发。任务综合就是根据软件说明生成代码,并用编译器对其进行编译和优化,进行相应的应用开发。硬件设计 Agent 在进行硬件说明的前提下,定义指令集,产生指令集模型,形成基于周期的指令集仿真模型(RTL级)。然后完成处理器 BIM 模块、可优化的存储器模块与硬件逻辑部分的 HDL 模块之间的布局连线工作。协同通信 Agent 负责软硬件设计 Agent 之间的实时沟通交流。整个设计流程要求性能评估 Agent 多次对底层的软硬件协同设计系统验证和仿真,确认设计是否满足功能要求和条件约束。甚至还应对系统的性能再次进行评估,以确定前面的先验估计是否准确。如果后验评估与先验评估相差太大,可能还需要重新进行结构划分和软硬件划分。

3.4 其他 Agent

在软硬件协同设计 MAS 中,还涉及了协同通信 Agent、性能评估 Agent 和硬件系统测试 Agent,限于篇幅,本文只对其进行简单介绍。

(1)协同通信 Agent:在软硬件协同设计过程中,协同通信 Agent 是软件设计 Agent 和硬件设计 Agent 之间沟通交流的桥梁。当一方 Agent 发起呼叫组织实时协同时,呼叫会首先通知协同通信 Agent,通过协同通信 Agent 发布消息,另一方的 Agent 收到消息会做出相应的合作请求。握手成功后,软件设计 Agent 和硬件设计 Agent 以协同通信 Agent 为依托进行交流协商。在通信综合步骤中,通过选择协议和分配内存来确定通信机制。

(2)性能评估 Agent:性能评估 Agent 对系统描述 Agent 所产生的结果进行验证仿真,以便在详细设计开始之前验证需求分析和系统描述的正确性。由于性能评估 Agent 在软硬件协同设计的 MAS 系统中的应用几乎涉及所有的 Agent,因此可以把性能评估 Agent 设计成移动 Agent,这样就可以移动到所需资源的所在处,与资源直接进行交互,省去了中间数据的传输,降低了对网络带宽的要求,减少了延迟。性能评估 Agent 的功能主要包括协同仿真和协同验证,为降低设计风险,在流程的每一步都会进行不同形式的验证和分析。

(3)硬件系统测试 Agent:硬件系统测试 Agent 负责测试功能模块、模块的部分接口以及总线功能,然后在搭建好的芯片上运行实际的应用软件或测试平台。

由上述分析可知,Agent 构建是基于 MAS 的软硬件协同设计系统的基本问题。另外,各 Agent 之间的协商机制及算法也是实现分布式求解软硬件协同设计问题的关键。利用面向对象技术和通信技术可实现 Agent 功能设计及 Agent 间的协商,这为实现基于 MAS 的软硬件协同设计提供了技术支撑,保证了 MAS 技术应用于软硬件协同设计的可行性。

4 结束语

本文应用 MAS 技术的相关理论,对软硬件协同设计过程进行分析,提出一种基于 MAS 的软硬件协同设计模型,将整个软硬件协同设计任务分派给若干个相互合作而又相对独立的 Agent 来完成,使得原本较为复杂的任务变成了几个较为简单任务的组合。未来的 SoC 中将会用到更多的处理器,以便更加灵活地支持不断出现的新应用,这也使得 SoC 设计更为复杂,对设计方法及设计工具提出了新的要求。在软硬件协同设计中引入多 Agent 技术,为面向 SoC 的软硬件协同设计的成功实现提供了一个新的解决方案。

参考文献

- [1] Shen Weiming, Norrie D H. Agent-based Systems for Intelligent Manufacturing: A State-of-the-art Survey[J]. Knowledge and Information Systems, 1999, 1(2): 129-156.
- [2] 李海刚, 吴启迪. 多 Agent 系统研究综述[J]. 同济大学学报, 2003, 31(6): 728-732.
- [3] 马巧云. 基于多 Agent 系统的动态任务分配研究[D]. 武汉: 华中科技大学, 2006.
- [4] 窦 讯, 张广明. 基于多 Agent 的电梯智能群控系统[J]. 东南大学学报, 2005, 35(2): 219-224.
- [5] 范玉顺, 曹军威. 多代理系统理论、方法与应用[M]. 北京: 清华大学出版社, 2002.
- [6] Papadopoulos G A, Arbab F. Coordination Models and Languages[J]. Advances in Computers, 1998, 3(1): 329-400.
- [7] 赵 川, 徐 涛, 孙晓光. 软硬件协同设计方法的研究[J]. 计算机工程与设计, 2003, 24(7): 7-9.
- [8] Allen C, London T. LDAP Directory-based Dynamic Publishing: A Model for Dynamic Web Publishing Using XML and the Sun One Directory Server[Z]. [2009-07-28]. <http://www.sun.com/software>.

编辑 顾姣健

(上接第 255 页)

(4)可以改变 m_v 和 m_w 的值对多阶段任务多态系统的备件问题进行仿真求解和分析。

5 结束语

本文对基于备件方案的多态 PMS 问题进行了可靠性仿真求解和分析,算法时间和空间复杂度小,阶段随机数产生原则提高了仿真精度。然而,模型仍有一些需要改进之处,如可选的备件更换策略太少,缺乏对可修问题的考虑,这些都有待进一步研究。

参考文献

- [1] Tang Zhihua, Dugan J B. BDD-based Reliability Analysis of Phased

Mission Systems with Multimode Failures[J]. IEEE Trans. on Reliability, 2006, 55(2): 350-360.

- [2] 张 涛, 郭 波, 谭跃进. k/N 系统的多阶段任务可靠度模型[J]. 系统工程, 2004, 22(8): 100-103.
- [3] 郭 波, 张 涛, 张 泉, 等. 备件组合方案下的多阶段任务成功性评估模型[J]. 系统工程理论与实践, 2005, 25(2): 94-100.
- [4] 张 涛, 高大化, 郭 波, 等. 多阶段任务系统的备件保障度模型研究[J]. 系统工程学报, 2006, 21(1): 86-92.
- [5] 张玉涛, 唐 俊, 张明清, 等. 基于蒙特·卡罗方法的可靠性仿真过程模型研究[J]. 系统工程与电子技术, 2008, 30(7): 1374-1378.

编辑 张正兴