

## 基于扩展层次有色 Petri 网创建 SoC 高层可执行规约\*

余金山<sup>1+</sup>, 李 瞰<sup>1</sup>, 张耀鸿<sup>2</sup>, 谭庆平<sup>1</sup>

<sup>1</sup>(国防科学技术大学 计算机学院,湖南 长沙 410073)

<sup>2</sup>(国防科学技术大学 信息系统与管理学院,湖南 长沙 410073)

### Creating System-on-Chip High-Level Executable Specification Based on Extended Hierarchical Colored Petri Net

YU Jin-Shan<sup>1+</sup>, LI Tun<sup>1</sup>, ZHANG Yao-Hong<sup>2</sup>, TAN Qing-Ping<sup>1</sup>

<sup>1</sup>(School of Computer Science, National University of Defense Technology, Changsha 410073, China)

<sup>2</sup>(School of Information System and Management, National University of Defense Technology, Changsha 410073, China)

+ Corresponding author: Phn: +86-731-4515764, E-mail: yujinshan@yeah.net

Yu JS, Li T, Zhang YH, Tan QP. Creating system-on-chip high-level executable specification based on extended hierarchical colored Petri net. *Journal of Software*, 2006,17(Suppl.):148-153. <http://www.jos.org.cn/1000-9825/17/s148.htm>

**Abstract:** This paper presents a method for creating system-on-chip high-level executable specification based on extended colored Petri net (EHCPN). Based on EHCPN model, a modeling and simulation integrated environment: SoC-P/TMSE is implemented, which can efficiently support for creating high-level specification and executing analysis for system-on-chip, i.e., function analysis, performance evaluation and communication protocol verification.

**Key words:** extended hierarchical colored Petri net; specification; system-on-chip

**摘 要:** 提出一种基于扩展层次有色 Petri 网(EHCPN)模型创建 SoC 高层可执行规约的方法,给出了 EHCPN 的形式化定义.设计并实现了 EHCPN 建模与模拟环境:SoC-P/TMSE,环境能够有效支持 SoC 高层规约创建与模拟,如功能分析、性能评估和通信协议检查.

**关键词:** 扩展层次有色 Petri 网;规约;系统芯片

为了处理系统芯片(SoC)设计的复杂性,减少上市时间,提高设计效率,设计抽象已经应用于系统层.概括来讲,在 SoC 设计过程中存在两种风险:第一种是无法正确理解顾客的需求所导致的功能风险,第二种风险是所设计的产品无法达到预定的性能,这两种风险都将提高 SoC 设计的成本.为了更早地处理这两种风险,研究人员对更高抽象级别的设计和验证技术产生了浓厚的兴趣.目前,主要的系统级规约语言有 SpecC 语言<sup>[5]</sup>, SystemC<sup>[6]</sup>, SystemVerilog<sup>[7]</sup>, UML<sup>[1,2]</sup>.另一方面,基本 Petri 网及其扩展模型,具有强大的模拟能力、描述和分析并发行为独到的优越之处、直观的图形表示、严格的数学基础以及能够反映系统的动态性能等特点,使得它在许多领域得到应用,如对通信协议的验证、软件、实时和硬件系统的功能验证和性能分析.我们认为,采用 Petri 网

\* Supported by the National Natural Science Foundation of China under Grant Nos.60403048, 60573173 (国家自然科学基金)

Received 2006-04-15; Accepted 2006-09-11

及其扩展模型创建 SoC 高层可执行规约是另一种好的选择,这是因为可以利用其强大的描述能力、大量的模拟、性能分析和形式化验证工具,对 SoC 设计进行早期验证与评估,如支持 SoC 体系结构开采、性能分析、形式化验证、通信协议确认等.基于 Petri 网的 SoC 验证与分析技术,目前部分文献对其进行了讨论,如文献[3]采用 CPN 对 SoC 执行形式化验证.文献[4]利用确定和随机 Petri 网支持 SoC 通信性能分析.但是,利用 Petri 网及其扩展模型创建 SoC 的高层可执行规约,相关研究还相对比较薄弱.为此,本文提出了一种支持创建 SoC 系统高层可执行规约的扩展层次有色 Petri 网(EHCPN)模型.利用该模型,能够捕获 SoC 系统的层次化、模块化的体系结构、并发、并行、时延等特性,支持多种模型(如时间自动机、有限状态机)和语言(如 C++,SystemC,VHDL, Verilog)实现的 IP 芯核建模.我们设计并实现了基于 EHCPN 的建模和模拟环境:SoC-P/TMSE,支持创建基于 EHCPN 的 SoC 系统功能、时延、通信、同步等行为规约.

本文第 1 节给出 EHCPN 模型的形式化定义.第 2 节分析如何利用 EHCPN 创建 SoC 高层可执行规约.第 3 节设计并实现了基于 EHCPN 的建模和模拟环境:SoC-P/TMSE.第 4 节比较了相关研究工作.第 5 节总结全文.

## 1 EHCPN 模型定义

SoC 系统具有复杂的结构,需要良好的模型对其进行描述.传统的有色 Petri 网是一种平板结构,在创建 P/T 模型时,模型规模太大,不易理解.因此,我们提出了一种扩展的层次化有色 Petri 网(EEHCPN)模型,用于描述 SoC 设计:采用层次化的结构和面向对象的思想,能够有效捕获 SoC 的体系结构;利用 Petri 网的并发特性可以捕获 IP 核内部的并发行为;利用迁移上的时延捕获 SoC 中的时间延迟,利用带类型和数据的有色 Token 对 SoC 的 IP 芯核之间及其内部的通信行为建模.下面给出支持创建 SoC 可执行模型的扩展层次有色 Petri 网的形式化定义.

定义 1. 扩展的层次有色 Petri 网 (EHCPN) 为一个六元组  $EHCPN = (P, T, \Pi, I, O, M_0)$ , 其中:(1)  $P = \{p_1, p_2, \dots, p_n\}$  为有限非空库所的集合;(2)  $T = \{t_1, t_2, \dots, t_n\}$  为迁移的有限集合;(3)  $\Pi = \{S_1, S_2, \dots, S_r\}$  为超迁移 (super transition) 的有限集合,每个  $S_i \in \Pi$  都表示一个扩展的层次有色 Petri 网  $EHCPN_i = (P_i, T_i, \Pi_i, I_i, O_i, M_{0i})$ ;(4)  $I$  是与外界进行消息交换的输入有色库所有限集;(5)  $O$  为与外界进行消息交换的有色输出库所集;(6)  $M_0$  为网的初始标识集.

图 1 给出了一个 EHCPN 的简单示例,模型包含一个超迁移.

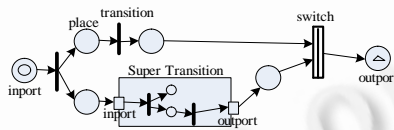


Fig.1 An EHCPN sample

图 1 EHCPN 示例

在 EHCPN 的内部结构中,Token 是结构化的数据,迁移对 Token 中数据的修改常代表特定的语义,Token 本身有类型,只能自动在有相同类型的库所上传递.Token 本身还带一个属性表,使得 Token 在网中流动时能真正携带信息.属性表本身是动态的,可以在 P/T 网执行过程中增加和修改.每一个属性包括名称和值两项,用名称进行存取,其定义如下:

定义 2. 扩展 Token (extended token) 为一个两元组  $T = (Type, PropertyLists)$ , 其中 Type 表示 Token 的类型,对任意 Token 的属性  $P_i \in PropertyList$ , 都表示一个名称与值的二元组:  $P_i = (Name_i, Value_i)$ , 其中  $Name_i$  为 Token 的属性名称,  $Value_i$  为属性值.

扩展迁移用来描述系统发生的行为和执行的动作,与 CPN 相比,扩展迁移增加了时间延迟、优先级、附加的点火条件(谓词)、执行的动作函数以及针对迁移的事件处理函数,并在运行时能统计关于迁移的各种参数(平均服务时间、最大服务时间、最小服务时间、点火次数等),这样可以更好地描述迁移的时间特性和解决冲突的控制问题.扩展迁移将收到的 Token 和完成迁移处理功能之后,将 Token 放置到后置的每个库所中去.

定义 3. 扩展迁移 (extended transition), 表示引入如下函数集和功能定义的迁移,其函数集和功能定义为:(1) 优先级:当多个迁移发生冲突时,优先级高的优先点火;(2) 动作函数:当点火发生时,调用该函数,可以用该函数

完成处理动作;(3) 延时函数:点火从进入点火状态到实施点火动作之间的一段时间,该时间长短由延时函数返回;(4) 谓词函数:附加谓词函数.谓词函数返回 0 时迁移可以进入点火状态,否则不能够进入点火状态.引入谓词函数,加强了网的描述能力,简化了网的结构,减少了大量辅助元素,有助于模型的理解;(5) 事件处理函数:当迁移进入点火状态时,系统调用事件处理函数.可以使用该函数进行统计或附加的处理动作.

定义 4. 开关(switch),表示收到 Token,并不立即将 Token 放置到后续库所的扩展迁移.

超迁移的引入为 Petri 网提供了层次化的结构,可以简化模型的构造,并引入了面向对象的建模思想,便于模型的理解.

定义 5. 超迁移(super transition),一个超迁移对应一个扩展的层次有色 Petri 网  $EHCPN=(P,T, \Pi, I, O, M_0)$ .

库所增加了类型和事件处理函数,提供了队列长度、等待时间和 Token 的统计特性值等规定.其定义如下:

定义 6. 扩展的库所(extended place),表示引入了事件处理函数和类型的库所.

当扩展库所收到 Token 或输出 Token 时,系统调用库所的事件处理函数.库所带有类型,使网带“色”.只有类型与库所相同的 Token 才能在 Token 自动传递机制中被传递到相应的库所上去.

## 2 利用 EHCPN 创建 SoC 芯核可执行规约

本节介绍如何利用 EHCPN 创建 SoC 芯核可执行规约.基本的 EHCPN 图形建模元素如图 2 所示.其中:(a) 用于顺序行为建模;(b) 用于并发行为建模;(c) 用于同步行为建模,其语义是只有两个事件都到达的时候,才执行迁移;(d) 用于异或同步行为建模,当其中的一个事件到达之后就执行迁移;(e) 用于选择某条路径执行的行为建模;(f) 用于循环行为建模;(g) 表示输入端口;(h) 表示输出端口;(i) 表示 Token 和库所;(j) 表示 EHCPN 子模型,用于构造层次化的 EHCPN 模型.

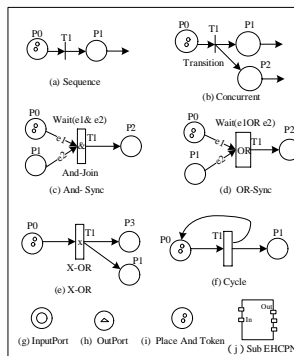


Fig.2 Base EHCPN graphical modeling elements

图 2 基本的 EHCPN 图形建模元素

利用 EHCPN 的特性,很容易对 SoC 芯核内部结构进行建模,可以根据其结构图对应创建 EHCPN 模型.图 3 给出了一个 RISC 处理器结构及其在 SoC-P/TSME 建模环境中创建的 EHCPN 模型.

通过分析硬件设计中每个进程的行为,利用 EHCPN 对进程行为建模.EHCPN 提供了丰富的建模元素,集成了脚本语言,支持在模型模拟过程中调用外部函数库、输出数据、提供初始化操作等功能.而且,这些建模元素,可以用于构造复杂的嵌套结构(如 if 和循环结构).同时,SoC 芯核之间的同步行为可以利用同步迁移建模,通信采用有色 Token 实现.图 4 给出了一种捕获进程内部及其同步行为的 EHCPN 建模示例.

时钟同步行为采用如图 5 所示的方式建模.其中迁移 T1,T2,T3 在收到时钟 Token 之后同步触发执行,实现时钟同步.迁移 T4 在收到来自库所 P4,P5 和 P6 的 Token 以及时钟 Token 之后触发执行,实现事件和时钟的同步执行.

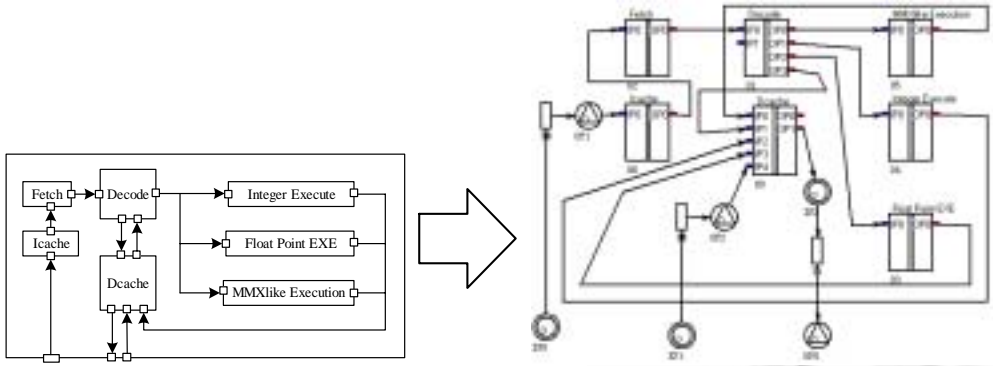


Fig.3 RISC processor structure and its EHCPN model

图 3 RISC 处理器结构及其 EHCPN 模型

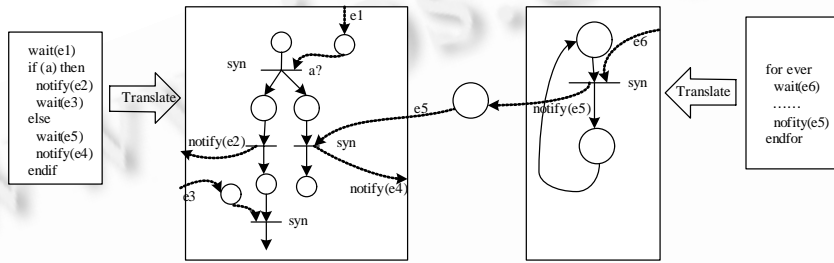


Fig.4 Process internal behavior and synchronization modeling

图 4 进程内部行为及同步建模

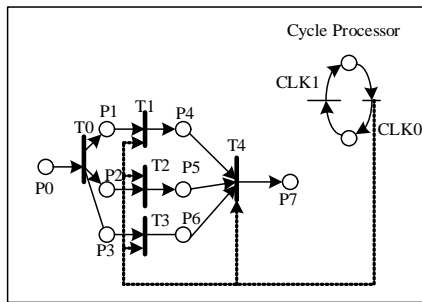


Fig.5 Clock synchronization behavior modeling

图 5 时钟同步行为建模

当获得 SoC 系统的 EHCPN 模型之后,为了支持在模拟验证环境中集成验证,需对模型添加模拟接口,进行封装,获得模拟模型.封装的接口包括:(1) SimInit(),属性初始化,将数据装入内存等;(2) SimStart(),设置模拟时钟,实例化模型;(3) SimPause(),暂停其模拟进程;(4) SimContinue(),推进模拟进程;(5) SimStop(),停止模拟进程;(6) SimAdvance(),推进本地时钟,执行本地事件;(7) GetSimClock(),报告自己的下次事件时钟;(8) SendMsg(),向消息发送端口发送消息.

定义 7. 模拟模型为一个 8 元组  $M = \{X, S, s_0, Y, \delta_{int}, \delta_{ext}, \lambda, ta\}$ ,其中:(1)  $X$  为外部输入事件集;(2)  $s_0$  为初始状态集;(3)  $S$  为状态集;(4)  $Y$  为外部输出事件集;(5)  $\delta_{int} : S \times \{i\} \rightarrow S$  为内部转移函数, $i$  代表下一预定时间产生的内部事件;(6)  $\delta_{ext} : Q \times X \rightarrow S$  为外部转移函数, $Q$  代表状态集,满足条件:  $\{(s,e) | s \in S, 0 \leq e \leq ta(s)\}$ ; (7)  $\lambda : S \times \{i\} \rightarrow Y$  为输出函数;(8)  $ta : S \rightarrow R_0^+$  为时钟推进函数.

### 3 基于 EHCPN 的建模和模拟环境:SoC-P/TMSE

基于所提出的 EHCPN 建模元素和建模机制,我们设计并实现了支持创建 SoC 高层可执行规约的建模和模拟的集成环境:SoC-P/TMSE. SoC-P/TSME 集成环境的体系结构如图 6 所示.

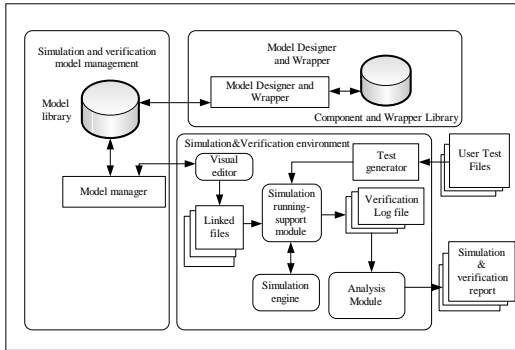


Fig.6 SoC/PTMSE architecture

图 6 SoC-P/TMSE 体系结构

环境主要包括 3 个部分:

(1) EHCPN 模型设计工具,提供了一个可视化建模环境,支持自动封装模拟模型.模型设计工具提供了模型库和封装库,支持重用.其 GUI 如图 6 所示.在建模工具中,EHCPN 模型采用类库方式组织:一个模型由若干个类组成,其中某个类为模型的主类,将其他类关联起来,逻辑上描述系统的总体结构,其他各个类则用来描述 SoC 系统各个组成部分的结构.

(2) 模型管理,提供了模型库和 GUI,用于存储、查询、更新和删除模拟模型.

(3) 模拟环境.它是 P/TMSE 环境的核心部分,其 GUI 如图 7 所示.模拟环境包括可视化 SoC 集成编辑工具、测试生成器、运行支持模块、模拟器.可视化编辑工具从模拟模型库中提取模拟模型,根据 SoC 的体系

结构进行集成,使得集成后的模型满足 SoC 的体系结构规约.可视化编辑工具将模型集成之后,联编为单个文件,传递给运行支持模块;测试生成器根据模型的配置和用户自定义的测试请求文件,生成测试配置,作为模拟的输入;运行支持模块同模拟器进行交互,模拟器支持对模型的模拟,获得模拟数据,存储到数据库或者日志文件中;最后,利用结果分析与覆盖率评估模块对数据进行分析,获得模拟报告.

作为模拟环境核心部分的模拟器,维护一个离散事件链表,并以该链表为中心,按照事件的偏序关系对模拟链表中事件进行调度.在模拟验证环境中,设计并实现了一个类似于总线的模拟总线结构.模拟总线由 5 条数据传输线组成:接收消息总线、发送消息总线、时钟推进通知总线、请求时钟推进总线和内部消息管理总线.负责模型之间的时钟同步和消息接收.模拟环境支持 SoC 系统高层模拟分析,包括其性能评估、通信协议的分析与验证.

### 4 相关工作比较

目前,用于支持 SoC 系统级模型的语言主要包括 SystemC<sup>[6]</sup>,SystemVerilog<sup>[7]</sup>,SpecC<sup>[5]</sup>和 UML<sup>[1,2]</sup>.SystemC<sup>[6]</sup>是基于 C++ 类库的一种系统级描述语言,支持 SoC 的事务级建模,提供了比 RTL 级更快的模拟速度.SpecC<sup>[5]</sup>是对 ANSI-C 进行的扩充,支持创建 SoC 的规约模型、体系结构模型、通信模型、实现模型.文献[8]对 SystemC 和 SpecC 在支持 SoC 系统级建模的能力方面进行了比较.SystemVerilog<sup>[7]</sup>是对 Verilog 语言的扩展,减少了硬件逻辑建模的代码量,为嵌入 SystemC 代码提供了标准的编程接口.UML<sup>[1,2]</sup>是一种可视化建模语言,可以创建图形化的 SoC 规约模型,但由于它主要应用于软件系统的建模,当前还需要对 UML 支持 SoC 的硬件设计和软/硬件协同设计作深入的研究.相对 SystemC, SystemVerilog, SpecC 和 UML 来说,本文所提出的 Petri 网模型是一种轻量级的 SoC 高层可执行规约模型(EHCPN).在建模方面,EHCPN 支持 SoC 的体系结构开采与体系结构精化、事务开采与事务精化、通信开采与通信精化、实现开采与实现精化.同时,EHCPN 能够有效支持 SoC 高层模型(体系结构模型、通信模型和实现模型)的模拟分析(如支持性能评估、功能分析)与形式化验证(如支持协议检查).

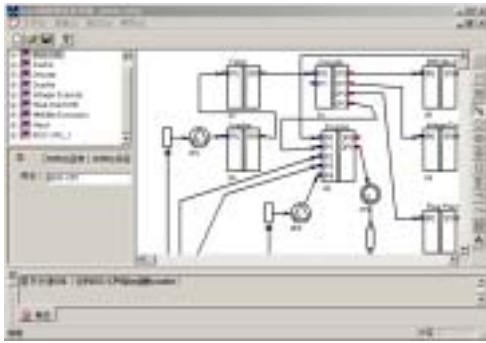


Fig.7 Model tool GUI  
图 7 建模工具 GUI

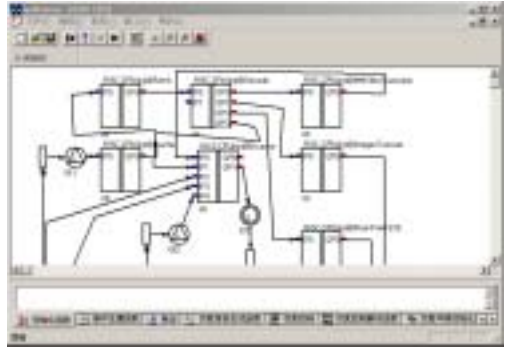


Fig.8 Simulation tool GUI  
图 8 模拟工具 GUI

### 5 结束语

本文提出了一种基于扩展层次有色 Petri 网(简称为 EHCPN)模型创建 SoC 高层可执行规约的方法.给出了 EHCPN 的形式化定义.基于所提出的 EHCPN 模型,设计并实现了 EHCPN 建模与模拟的集成环境:SoC-P/TMSE, 该环境能够有效支持 SoC 高层规约创建与模拟分析(如功能分析、性能评估、协议检查等).

### References:

- [1] Schattkowsky T. UML 2.0 overview and perspectives in SoC design. In: Proc. of the Design, Automation and Test in Europe Conference and Exhibition (DATE 2005). 2005. 832–833.
- [2] Zhu Q, Oishi R, Hasegawa T, Nakata T. Integrating UML into SoC design process. In: Proc. of the Design, Automation and Test in Europe Conference and Exhibition (DATE 2005). 2005. 836–837.
- [3] Zhan JY, Sang N, Xiong GZ. In: Wu GZ, *et al.*, eds. Formal co-verification for SoC design with colored Petri net. In: Proc. of the ICSS 2004. LNCS 3605, Berlin: Springer-Verlag, 2005. 188–195.
- [4] Blume H, von Sydow T, Noll TG. Performance analysis of SoC communication by application of deterministic and stochastic Petri nets. In: Pimentel A, Vassiliadis S, eds. Proc. of the SAMOS 2004. LNCS 3133, Berlin: Springer-Verlag, 2004. 484–493.
- [5] Doemer R, Gerstlauer A, Gajski D. SpecC Language Reference Manual. version 2.0 edition. SpecC Technology Open Consortium, 2001.
- [6] OSCI. SystemC 2.0.1 language reference manual revision 1.0. 2003. <http://www.systemc.org>
- [7] Accellera. SystemVerilog 3.1, extensions to verilog. 2003. <http://www.accellera.org>
- [8] Cai LK, Verma S, Gajski DD. Technical Report, CECS-03-11, Comparison of SpecC and SystemC Languages for System Design, 2003.



余金山(1976 - ),男,四川邻水人,博士生,主要研究领域为分布式软件工程,系统芯片的设计与验证技术.



张耀鸿(1973 - ),博士,副教授,主要研究领域为离散事件仿真.



李杰(1974 - ),博士,副教授,主要研究领域为并行模拟,微处理器设计验证技术,电子CAD 技术.



谭庆平(1965 - ),教授,博士生导师,主要研究领域为分布式软件工程,软件设计与验证技术.