

高等院校教材

SystemC

片上系统设计

陈曦 徐宁仪 编著
周祖成 审定



科学出版社

www.sciencep.com

高等院校教材

SystemC 片上系统设计

陈曦 徐宁仪 编著

周祖成 审定

科学出版社

北京

内 容 简 介

SystemC 是被实践证明的优秀的系统设计描述语言,它能够完成从系统到门级、从软件到硬件、从设计到验证的全部描述。SystemC 2.01 已作为一个稳定的版本提交到 IEEE,申请国际标准。

本书为配合清华大学电子工程系 SystemC 相关课程的教学而编写。全书分 9 章,内容包括:硬件描述语言的发展史; SystemC 出现的历史背景和片上系统设计方法学概述; SystemC 的基本语法; SystemC 的寄存器传输级设计和 SystemC 的可综合语言子集,以及根据作者设计经历归结的 RTL 设计准则和经验;接口、端口和通道等 SystemC 行为建模的基础、交易级建模和通信细化; SystemC 的 Master-Slave 通信库;一个 SystemC TLM 建模实例——片上总线系统; SystemC 与 VHDL/Verilog HDL 的比较; SystemC 的验证标准和验证方法学; SystemC 开发工具 SystemC_win、WaveViewer 等,以及使用 MATLAB 进行 SystemC 算法模块的验证。每一章都精心编写了课后习题以配合教学的需要。

本书可作为大学电子设计自动化(EDA)相关课程教材,也可供电子工程技术人员作为 SystemC 设计、应用开发的技术参考书。本书丰富的实例源代码特别适合初学者根据内容实际运行、体会,举一反三,以掌握 SystemC 进行应用系统设计。书中全部源代码可以从 <http://www.sciencecp.com> 上获得。

图书在版编目(CIP)数据

SystemC 片上系统设计/陈曦,徐宁仪编著. —北京:科学出版社, 2003.10

高等院校教材

ISBN 7-03-012292-5

I. S… II. ①陈…②徐… III. SystemC 语言—程序设计—高等学校—教材 IV.TP312

中国版本图书馆 CIP 数据核字(2003)第 091861 号

责任编辑 杨凯 筱戎 责任制作 魏 谨
责任印制 刘士平 封面设计 李 力

科学出版社 出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencecp.com>

源海印刷有限责任公司印刷

北京东方科龙图文有限公司 制作

<http://www.okbook.com.cn>

科学出版社发行 各地新华书店经销

2004年1月第 一 版 开本: B5(720×1000)

2004年1月第一次印刷 印张: 20

印数: 1~5 000 字数: 367 000

定 价: 30.00 元

(如有印装质量问题,我社负责调换〈新欣〉)

序

随着集成电路制造技术的发展,人们已经可以将包括存储器、信号采集和转换电路、DSP、CPU 核等模拟、数字和混合电路构成的一个完整的电子系统集成到一个芯片上,从而产生了片上系统(System on Chip)的概念。片上系统的核心技术是深亚微米的甚大规模集成电路制造技术和系统设计技术。

一方面,电子系统越来越复杂;另一方面,其市场寿命也越来越短,一个新产品往往上市几个月就被功能更新、更强大的产品代替了。这给电子设计带来了严峻的挑战。

在传统的设计方法中,首先由系统工程师进行系统规范,进行系统分割,接着硬件与软件分开设计。硬件使用硬件描述语言如 VHDL 或者 Verilog HDL 实现,软件使用 C、C++ 或者汇编语言实现。软件的设计往往要迟于硬件。这种设计方法的缺点是系统设计、硬件设计和软件设计使用不同的语言,无法进行软硬件协同验证,系统验证成为制约设计效率的主要因素,迫切需要一种通用语言能够完成从系统、软件到硬件、门级各个层次的设计描述和验证,这就是系统设计语言(System Level Description Language)。

当前系统设计语言的发展方向有两个。一是扩展传统的硬件描述语言 VHDL 和 Verilog HDL,使它们支持抽象数据类型,从而具有系统描述能力。这方面比较突出的工作之一是由 Accellera 组织(Open Verilog International 组织和 VHDL International 组织合并成立的硬件描述语言发展组织)的 SystemVerilog。二是扩展传统的软件语言 C 和 C++,使它们支持硬件描述。在 SystemC 出现之前已经有很多这方面的工作比较成功。但这方面的工作最突出的就是 SystemC。SystemC 在 Synopsys 等一批大公司的支持下得到了很大的发展,它已经成为很多 EDA(电子设计自动化)工具除 VHDL 和 Verilog HDL 之外的第三种支持语言。由于其开放性等特点,已经得到全世界工程师的欢迎和认可,许多大公司都推出了 SystemC 的开发工具。

SystemVerilog 可以粗略地理解为是 Verilog 的超集。业界多数人认为,如果它成为 IEEE 标准,很可能导致 VHDL 逐步退出历史舞台。同时业界也普遍认为, SystemC 与 SystemVerilog 刚好构成互补的关系,“SystemVerilog 的发展将会成为 SystemC 的基石”。由于 SystemC 在系统级设计与验证方面没有对手,我们认为它几乎必然会流行开来并成为 IEEE 的标准。

由于开发工具的不断推出, SystemC 在国外已经十分流行。目前已经有国内外一些大公司利用 SystemC 开发项目成功的案例。我们在深圳清华大学研究院 EDA 与网络应用重点实验室使用 SystemC 开发的数字音频芯片的流片也是使用 SystemC 开发成功的案例之一。我们已经在清华大学开始讲授 SystemC。

SystemC 之所以成功, 关键在于:

(1) 开放性和广泛的支持。由于 SystemC 扩展了 C++ 语言, 所以无数熟悉 C++ 又对硬件描述感兴趣的设计者可以很容易地掌握。对于那些熟悉硬件描述语言的工程师, 学习 SystemC 也很容易, 对于他们来讲, 只需找到他们与所熟悉的语言所对应的语法就可以了。SystemC 几乎受到了所有 EDA 公司的支持, 这也是它成功的关键。

(2) 抽象描述能力。SystemC 是在系统级和交易级建模的最佳语言, 同时它能够很好地支持寄存器传输级设计和验证。

(3) 仿真速度。SystemC 描述的系统是可执行的文件, 仿真速度远远大于逐行解释的硬件描述语言, 这一点对于越来越复杂的电子系统来讲十分重要。

周祖成

2003 年 9 月于清华园

编者序

1 写书背景

到目前为止，成为 IEEE 标准的硬件描述语言有两个：VHDL 与 Verilog HDL。VHDL 的英文全称为 VHSIC Hardware description language，其中 VHSIC 指 Very High Speed Integrated Circuit。VHDL 出现于 1980 年，并于 1987 年开始成为 IEEE 标准，称为 IEEE Std. 1176。Verilog HDL 语言出现于 1983 年，并于 1995 年成为 IEEE 标准，称为 IEEE Std 1364—1995。与 VHDL 和 Verilog HDL 同时存在的语言如 Superlog、SpecC 等，由于它们由专门的公司所有，并非开放标准，所以不是很流行。

VHDL 与 Verilog HDL 并称为传统硬件描述语言。基于它们的设计方法是，在系统级一般采用 C/C++ 等高级语言进行算法描述，然后再手工转换为 VHDL 或者 Verilog HDL。由于两个阶段使用不同的设计语言，不能够很好地结合起来，软硬件协同设计和仿真、验证很困难。随着集成电路工艺水平的提高，现代业界进入了片上系统集成 (System on Chip) 的时代，一大特征就是软件设计的比重越来越大。然而，使用传统设计方法只能在硬件设计接近完成时才能开始软件的设计和调试，从而导致设计成为整个集成电路产业的瓶颈。

基于以上原因，人们致力于开发一种新的设计语言，它能够提供全面的系统描述能力。由于在软件领域 C/C++ 的绝对优势，人们渴望能够利用它们来描述硬件。这方面的研究工作已经开展了 10 多年的时间，一直处于自发的状态，直到 1999 年 SystemC 的出现才改变了这一格局。

1999 年 9 月，一些微电子业内一流的 EDA 公司、IP 提供商、半导体制造商及系统和嵌入式软件设计公司在加利福尼亚州 Saint Jose 举行的“嵌入式系统会议”上，联合创建了开放 SystemC 推动 (OSCI 或者 Open SystemC Initiative) 组织，并推出了基于 C++ 的系统设计语言——SystemC。OSCI 是一个非盈利性组织，它负责维护和发展 SystemC。SystemC 是完全免费的，这使得 EDA 供应商能够充分地了解 SystemC 库的源代码以优化它们的设计工具。

SystemC 出身名门，是绝对的宠儿，它一开始出现就受到业界前所未有的欢迎。又由于 SystemC 是开放的标准，所以它很快就流行开来。目前业界知名的公司包括 Synopsys、Cadence、Frontier Design、ARM、Eriassion、Lucent、Sony、Synopsys、TI 等都支持 SystemC。

SystemC 是一种完成电子系统从软件到硬件的全部建模过程的语言，比起

VHDL/Verilog HDL, 其在系统设计方面的优势明显。

SystemC 最近的版本是 2.01, 目前许多 EDA 厂商都在开发或者扩展它们的设计工具以支持 SystemC。现在已经推出的商用 SystemC 软件有 CCSS、SPW、SystemC ART 等 50 多种。已经有很多著名公司(包括西门子、ARM、Alcatel、富士通, 中国的东方通信、大唐飞利浦等)开始使用 SystemC 作为其系统级开发语言。一些归国的留学生在他们自己的小公司里也已经开始使用 SystemC 进行开发和验证。作者所在的实验室之一的清华大学深圳研究院 EDA 实验室已经将其用 SystemC 设计的第一款数字音频芯片进行了流片。

业界一般认为, SystemC 可望在 1~2 年之内成为 IEEE 标准。目前 SystemC 2.01 作为一个稳定的版本已经提交到 IEEE 进行标准化。SystemC 的标准化会加速 SystemC 的应用和发展。

目前, 另外一种被业界看好的硬件描述语言是 SystemVerilog, 它是 Verilog 2001 标准的扩展, 增加了类似 C 语言的结构和增强的验证功能, 它与 SystemC 恰好形成互补的关系。SystemC 和 SystemVerilog 在不同的设计领域中, 分别有各自的特点。两种语言的结合, 将从系统规范、门电路布局直至设计验证, 为设计人员提供一种可供选择的综合语言环境。这种结合可以看作是工具和语言相统一的转变潮流中的重要部分, 这种统一是为了满足最终用户的需求, 而不仅仅是为了方便工具供应商。

需要特别澄清的是, SystemC 语言的主要目的是实现用单一语言完成设计规范、系统体系结构分析、验证平台和行为模型, 而寄存器传输级和门级建模并非 SystemC 所长。现在的很多设计都从算法开始, SystemC 比 C/C++ 更适合对算法的描述和分析。传统硬件描述语言和 SystemVerilog 的 C 接口不支持硬件设计中的并发性、层次性和互连, 所以并不适合描述复杂多样的系统行为。通常用 SystemC 的行为模型描述的模块比 RTL 模块仿真速度快 10~100 倍, 这是 SystemC 的价值之一。SystemC 的更大价值在于其提供的高层次设计流程。基于渐趋成熟的行为级综合工具和 SystemC 行为模型的高层次设计流程将会大大提高设计和验证效率。SystemC 特别为行为综合设计的语法将使它成为非常适合行为级综合的设计语言。

2 本书内容安排

本书内容安排如下: 第 1 章着重介绍硬件描述语言的发展史, 以及 SystemC 的出现和成为 IEEE 标准的必然趋势。第 2 章介绍 SystemC 的基本语法, 包括模块、端口、信号、进程、基本数据类型、定点数据类型、sc.main() 函数以及波形跟踪等。该章内容是所有后续章节的基础, 读完了第 2 章就可以完成全部的 SystemC 寄存器传输级建模。第 3 章着重讨论 SystemC 的寄存器传输级设计和 SystemC 的可综合语言子集, 并根据作者的设计经验讲述 RTL 设计的一些准则和经验。第 4

章将围绕 SystemC 行为建模展开,前半部分着重介绍接口、端口和通道等 SystemC 行为建模的基础。后半部分介绍 SystemC 行为建模中的交易级建模(transaction level modeling),内容涉及系统建模的分层模型、交易级建模的基本概念和通信细化。第 5 章讲述 SystemC 提供的 Master-Slave 通信库,这个库尤其适用于包含一个或者多个处理器核、数字信号处理器、周边设备和用户定制集成电路而且是利用一组总线相互通信的系统建模。第 6 章集中介绍一个 SystemC TLM 建模实例——片上总线系统,包括主从设备接口、快速/慢速存储器通道、通用串口通道、仲裁器接口、仲裁器模块、仲裁算法的 SystemC 实现。第 7 章写给那些熟悉传统硬件描述语言 VHDL/Verilog HDL 或者需要将传统硬件描述语言写成的 IP 向 SystemC 进行手工转换的读者。通过 SystemC 与 VHDL/Verilog HDL 的直接比较,读者可以很快地掌握和更好地理解 SystemC。第 8 章将简要介绍 SystemC 的验证标准和验证方法学,主要描述 SystemC 的验证标准的基本思路和主要功能。通过该章,读者可以对基于交易的验证 SystemC 方法有一个初步的了解,并且可以利用 SystemC 库进行简单的验证平台(Testbench)的编写。第 9 章是本书最后一章,介绍一些基于个人电脑的 SystemC 开发工具,包括 SystemC_win、WaveViewer,并详细介绍了如何使用 MATLAB 进行 SystemC 模块,尤其是 SystemC 算法模块的验证。

从电子系统建模层次的角度看,第 1、2、3 章描述 SystemC 的寄存器传输级建模;第 4、5、6 章讨论 SystemC 的行为级建模;第 7 章阐明 SystemC 与传统硬件描述语言的等效性;第 8 章介绍 SystemC 的验证;第 9 章给出 SystemC 的设计工具。其中第 5 章和第 8 章又分别围绕 SystemC 的 Master-Slave 库和验证库展开。

本书第 2.4 节描述的是 SystemC 定点数据类型,这一节是与后续章节无关的,读者如果在工作中用不到,可以略过本节。在本书作者看来,这一节放在第 2 章是否合适还有待讨论。

本书的每一章节中都给出了较多的典型设计实例以加深读者对这些概念的理解,读者可以对照着这些代码学习。

作为教材,作者建议在课堂上集中讲授第 1、2(不包括 2.4 节)、4、5、8 章,其他章节可以作为课外阅读内容。本书每一章都安排了习题,可以作为课后作业。同时可以考虑以第 6 章的片上总线模型为起点,安排一个复杂度和任务量适中的课程设计。

3 写书目的与读者对象

清华大学电子工程系已经开始在其《电子系统仿真和 VHDL》和《通信系统仿真与 ASIC 设计》两门课中讲授 SystemC,本书写作的主要目的是作为这两门课程的教材。本书作者的导师,也是这两门课的授课教授,对于本书的编写给予了细

致入微的指导。

本书出版的另外一个目的是配合 SystemC 在国内发展的需要。目前,国外已经有若干本 SystemC 方面的图书,而据作者所知,中文的相关出版物却一直没有。本书作为国内 SystemC 方面的第一本书,希望能够方便国内工程师对 SystemC 的了解、学习和使用。

作为教材,本书在注重对基本概念的解释的同时,还引入了大量的实例以加深读者对所述内容的理解。

本书可以作为大学相关课程的高年级本科生和研究生教材,也可以作为工程技术人员的参考手册。本书读者需要有 C++ 的基础知识,包括 C++ 中的纯虚函数和模板类。如果读者已经掌握了一种硬件描述语言(如 VHDL 或者 Verilog HDL),同时又具有 C/C++ 的基本知识,那么阅读本书将会如鱼得水。

如果读者是一位软件工程师,阅读本书可以迅速掌握数字电路的设计和建模。如果读者是一位硬件工程师,通过阅读本书,将可以掌握新一代描述语言,从而更加有效地与软件工程师合作。作为 IC 设计公司的负责人,可以借助本书了解业界最新的设计方法学和电子系统设计领域的动向,缩短公司产品上市时间(time to market),以在竞争中获得先发优势。

本书所有实例都有源代码,读者可以从科学出版社的网站(<http://www.sciencep.com>)下载或者向本书作者索取。

4 作者简介

本书以两个高年级博士生为主完成,他们的研究方向都是片上系统集成(SoC)。他们都有丰富的 Verilog HDL 和 VHDL 的开发经验,参与过骨干网路由器项目的开发,独立完成过百万门级 FPGA 的设计,并于 2002 年下半年和 2003 年上半年参与国家 863 项目期间使用了 CoCentric System Studio 和 SystemC 进行了系统建模和仿真,最终系统满足了设计要求。本书作者还多次在国内外期刊和会议上发表有关集成电路设计和 SystemC 方面的文章,他们都是中国通信学会学生会员、IEEE 学生会员。

本书第 1、2、4、6、7、9 章由陈曦完成;第 3、5、8 章由徐宁仪完成。

本书的编写得到了作者导师周祖成教授的大力支持和指导。作者导师长期从事 EDA 相关领域的研究,并在清华大学主讲《电子系统仿真和 VHDL》和《通信系统仿真与 ASIC 设计》这两门课。

5 致谢

本书作者首先感谢恩师周祖成教授将自己带入电子设计自动化和集成电路设

计这一前景广阔、研究活跃的科学领域，感谢在同一实验室工作的罗飞同学、贺光辉博士、刘付娥博士等为本书的出版所做的努力。

本书第一作者在此感谢妻子李光在本书出版过程中所给予的理解和支持。

本书第二作者在此感谢杜爽同学对其工作的大力支持和无私的帮助。

感谢 OSCI (Open SystemC Initiative) 的广大工程师在 SystemC 发展中所付出的不懈努力。

感谢科学出版社为我们出版了本书。没有出版社同仁细致认真的工作，本书也无法与读者见面。

6 再版目标

作者希望在 SystemC 进一步发展以后能够再版本书，修正本书第一版中的错误，吸收第一版读者的意见，增加新的语法内容，并给出在工作站和个人电脑上都能使用的源代码，添加更加详尽的寄存器传输级设计的经验方法和系统级设计方法学，还可能引入 SystemVerilog 的部分内容。

7 反 馈

本书作者力求将本书写好，但一则工作中纰漏在所难免，二则作者能力有限。对于本书的任何意见和建议，欢迎读者联系本书作者。

本书作者联系方式：

清华大学东主楼 9 区 324 房间 清华大学微波与数字通信国家重点实验室
CAD 中心

陈曦 徐宁仪

邮编 100084

作者的校内 Email: chenxie@mails.tsinghua.edu.cn

xuny97@mails.tsinghua.edu.cn

作者的公众网 Email: chenxi01@tsinghua.org.cn

您的建议和意见将融入本书，成为本书第二版出版的最直接的动力。本书作者就此诚谢。

编 者

2003 年 9 月于清华园

目 录

第 1 章 电子系统设计方法学和系统级描述语言概述

- 1.1 片上系统对设计描述语言的要求 1
- 1.2 传统硬件描述语言 Verilog HDL 和 VHDL 2
- 1.3 SystemC 的历史 3
- 1.4 SystemC 到底是什么 4
- 1.5 基于 SystemC 的设计流程 5
- 1.6 一个“Hello, SystemC!”建模实例 5
- 1.7 SystemC 的系统描述能力 6
- 1.8 SystemC 的开发工具 9
- 1.9 使用 Visual C++编辑和编译 SystemC 设计 9
- 1.10 利用 ModelSim 查看 SystemC 产生的波形文件 12
- 习 题 14

第 2 章 SystemC 基本语法

- 2.1 模 块 15
 - 2.1.1 模块的定义 15
 - 2.1.2 模块的端口 16
 - 2.1.3 模块的信号 17
 - 2.1.4 位置关联 18
 - 2.1.5 名字关联 21
 - 2.1.6 模块内部数据 22
 - 2.1.7 模块的构造函数 23
 - 2.1.8 模块的析构函数 25
- 2.2 端口和信号 26
 - 2.2.1 端口和信号的基本概念 26
 - 2.2.2 端口和信号的读写 27
 - 2.2.3 端口和信号类型 28
 - 2.2.4 端口和信号的多驱动处理 29
 - 2.2.5 端口和信号的绑定 31
 - 2.2.6 SystemC 时钟模型 31

2.2.7	SystemC 时间模型	33
2.3	基本数据类型	34
2.3.1	sc_bit 和 sc_logic 数据类型	35
2.3.2	固定精度整型数据类型 sc_int 和 sc_uint	36
2.3.3	任意精度整型数据类型 sc_bigint 和 sc_biguint	39
2.3.4	任意长度比特和逻辑向量	41
2.3.5	用户自定义类型	43
2.4	定点数据类型	43
2.4.1	SystemC 定点数据类型	43
2.4.2	定点数据的字长和整数部分字长	44
2.4.3	定点数据的量化模式	45
2.4.4	量化模式 SC_RND	46
2.4.5	量化模式 SC_RND_ZERO	46
2.4.6	量化模式 SC_RND_MIN_INF	47
2.4.7	量化模式 SC_RND_INF	48
2.4.8	量化模式 SC_RND_CONV	49
2.4.9	量化模式 SC_TRN	50
2.4.10	量化模式 SC_TRN_ZERO	50
2.4.11	定点数据的溢出模式	51
2.4.12	溢出模式 SC_SAT	52
2.4.13	溢出模式 SC_SAT_ZERO	52
2.4.14	溢出模式 SC_SAT_SYM	53
2.4.15	溢出模式 SC_WRAP	54
2.4.16	溢出模式 SC_WRAP_SM	57
2.4.17	定点数据类型支持的运算符	58
2.4.18	定点数据类型的状态信息	59
2.4.19	将定点数据类型转换为字符串	59
2.4.20	一个定点 FIR 滤波器设计实例	60
2.5	进 程	61
2.5.1	SystemC 进程基础	61
2.5.2	方法进程 SC_METHOD	62
2.5.3	线程进程 SC_THREAD	63
2.5.4	钟控线程进程	65
2.5.5	wait_until()、wait()和 next_trigger()	68
2.5.6	watching 结构	70
2.5.7	局部 watching	71
2.6	仿真与波形跟踪	73

2.6.1	SystemC 设计的顶层函数 sc_main()	73
2.6.2	仿真控制	74
2.6.3	SystemC 波形跟踪概述	76
2.6.4	创建和关闭波形跟踪文件	77
2.6.5	跟踪标量型变量和信号	77
2.6.6	跟踪聚合型变量和信号	78
2.6.7	仿真和波形跟踪实例	79
习 题		80

第 3 章 寄存器传输级 SystemC 设计

3.1	SystemC 寄存器传输级设计和综合	82
3.1.1	什么是综合	82
3.1.2	为什么要用 SystemC 进行 RTL 建模	83
3.2	RTL 风格的 SystemC 编程	84
3.2.1	定义模块和进程	84
3.2.2	创建模块	85
3.2.3	定义敏感表	86
3.2.4	信号和变量的读写	88
3.3	SystemC 的可综合语言子集和可综合数据类型	90
3.3.1	可综合语言子集详解	90
3.3.2	可综合的数据类型详解	93
3.3.3	可综合修改建议	97
3.4	可综合 RTL 编程参考实例	97
3.4.1	寄存器建模	97
3.4.2	三态逻辑建模	107
3.4.3	组合逻辑建模	109
3.4.4	有限状态机建模	112
习 题		118

第 4 章 SystemC 行为建模

4.1	行为级建模的目的	120
4.2	接口、端口和通道的基本概念	120
4.3	接 口	122
4.3.1	接口的定义	122
4.3.2	存储器接口实例	123

4.3.3	接口基类 <code>sc_interface</code>	124
4.4	端 口	126
4.4.1	自定义端口	126
4.4.2	一个端口实例	127
4.4.3	端口基类 <code>sc_port<IF, N></code>	130
4.4.4	一个连接到多个接口的端口实例	131
4.4.5	直接通道调用	133
4.5	通道基础	136
4.5.1	端口与通道的关联	136
4.5.2	通道的同步规则	138
4.5.3	静态规则检查	138
4.5.4	动态规则检查	140
4.5.5	通道的属性	141
4.6	基本通道	141
4.6.1	<code>sc_signal<T></code> 、 <code>sc_signal_rv<T></code> 和 <code>sc_buffer<T></code>	142
4.6.2	<code>sc_mutex</code>	144
4.6.3	<code>sc_fifo<T></code>	146
4.6.4	<code>sc_semaphore</code>	148
4.7	分层通道	150
4.7.1	分层通道的定义	150
4.7.2	一般分层通道	151
4.7.3	特殊分层通道	158
4.8	系统建模中的分层模型	158
4.8.1	系统建模中通信的抽象层次	158
4.8.2	寄存器传输层	159
4.8.3	传输层	159
4.8.4	交易层	160
4.8.5	消息层	161
4.9	SystemC 的交易级建模	162
4.9.1	交易的概念	162
4.9.2	嵌入式软件开发与交易级建模	162
4.9.3	交易级建模用于系统结构探索	163
4.9.4	SystemC 交易级建模的特点	163
4.10	通信细化	164
4.10.1	通信细化的概念	164
4.10.2	一个通信细化实例	164
习 题		170

第 5 章 SystemC 的 Master-Slave 通信库

- 5.1 SystemC Master-Slave 通信库综述 174
- 5.2 SystemC Master-Slave 通信库的安装 175
- 5.3 功能级的 Master-Slave 通信库 175
 - 5.3.1 一个 Master-Slave 通信的简单实例 175
 - 5.3.2 串行信道 `sc_link_mp<T>` 和内嵌执行语法简介 178
 - 5.3.3 主、从端口语法 179
 - 5.3.4 从进程语法 180
 - 5.3.5 多点通信中的内嵌执行语法 181
 - 5.3.6 并行通信与主-从通信的结合 182
 - 5.3.7 通信端口连接规则和实例 182
 - 5.3.8 抽象端口类详解 184
- 5.4 总线周期精确级的 Master-Slave 通信库 188
 - 5.4.1 利用总线协议进行通信细化 188
 - 5.4.2 模块细化 188
 - 5.4.3 信道细化 191
 - 5.4.4 预定义的总线协议 195
 - 5.4.5 端口追踪 197
 - 5.4.6 用户自定义的总线协议 198
- 5.5 Master-Slave 通信库实例 200
 - 5.5.1 功能级的 FIFO 模型 200
 - 5.5.2 BCA 级的 FIFO 实例 206
- 习 题 208

第 6 章 TLM 设计实例——片上总线

- 6.1 片上总线系统概述 210
- 6.2 从设备接口 211
- 6.3 快速存储器的实现 213
- 6.4 慢速存储器的实现 215
- 6.5 通用串口的实现 217
- 6.6 总线主设备接口 221
- 6.7 总线主设备 222
- 6.8 总线的实现 223
 - 6.8.1 直接接口的实现 224

- 6.8.2 非阻塞型接口的实现 224
- 6.8.3 阻塞型接口的实现 225
- 6.8.4 助手函数 end_of_elaboration 的实现 226
- 6.8.5 助手函数 get_slave 的实现 226
- 6.8.6 助手函数 get_request 的实现 227
- 6.8.7 助手函数 get_next_request 的实现 227
- 6.8.8 助手函数 clear_locks 的实现 228
- 6.8.9 助手函数 handle_request 的实现 228
- 6.9 仲裁器接口 230
- 6.10 仲裁器模块的定义 230
- 6.11 仲裁器策略的实现——函数 arbitrate() 231
- 习 题 232

第 7 章 SystemC 与传统硬件描述语言 VHDL/Verilog HDL 的比较

- 7.1 SystemC 与传统硬件描述语言的关系 233
- 7.2 SystemC 与 VHDL 的语法等效性 235
 - 7.2.1 ENTITY、ARCHITECTURE 与 SC_MODULE 之间的等效性 235
 - 7.2.2 process 与 method 之间的等效性 236
 - 7.2.3 信号定义和端口声明 237
 - 7.2.4 例化和端口映射 237
 - 7.2.5 运算符 238
 - 7.2.6 数据类型 239
- 7.3 SystemC 与 Verilog HDL 的语法等效性 240
 - 7.3.1 基本构成单元 240
 - 7.3.2 进 程 240
 - 7.3.3 时间模型 241
 - 7.3.4 分支控制语句 242
- 7.4 SystemC 与 VHDL/Verilog HDL 等效的设计实例 242
 - 7.4.1 异步复位的 D 触发器 242
 - 7.4.2 移位寄存器 244
 - 7.4.3 计数器 246
 - 7.4.4 有限状态机 249
- 习 题 257

第 8 章 基于 SystemC 的验证方法学

- 8.1 SystemC 验证标准 260
- 8.2 与 SystemC 验证方法学相关的术语 262
- 8.3 SystemC 的验证标准 263
 - 8.3.1 交易器建模的风格 264
 - 8.3.2 动态并发性建模 267
 - 8.3.3 交易处理和记录 269
 - 8.3.4 受约束的随机数产生 271
 - 8.3.5 SCV 标准支持的其他用于功能验证的特性 272
- 8.4 验证实例 272
- 习 题 278

第 9 章 SystemC 的开发工具

- 9.1 利用 SystemC_win 编译和仿真 SystemC 设计 280
- 9.2 使用 WaveViewer 查看波形 281
- 9.3 MATLAB 用于 System 算法模块的验证 283
 - 9.3.1 将 MATLAB 作为 SystemC 验证程序的计算引擎 283
 - 9.3.2 MATLAB 作为计算引擎的例子 283
 - 9.3.3 MATLAB 作为 I/O 与 SystemC 验证程序通信 287
 - 9.3.4 MATLAB 作为验证 I/O 的例子 287
 - 9.3.3 两种方法的优缺点 295
- 习 题 295
- 附录 部分名词术语英汉对照 296
- 参考文献 299