

<<SystemC入门>>

图书基本信息

书名：<<SystemC入门>>

13位ISBN编号：9787811242492

10位ISBN编号：7811242494

出版时间：2008-9

出版时间：北京航空航天大学

作者：巴斯克

页数：239

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## 前言

J. Bhasker在数字集成电路设计界具有很好的声誉。

包括美国和中国在内的各国数字电路和系统设计者，无论是新手，还是从VHDL转到Verilog的老手，其中很多人都阅读过J. Bhasker编写的A Verilog HDL Primer。

J. Bhasker的教材以简明扼要、清晰易懂著名。

A SystemC Primer一书是介绍SystemC基础知识的教科书，符合他编书的一贯风格，特别适合作为入门教材。

SystemC对许多中国读者来说，还是一种新的硬件描述语言。

近年来，由于基于平台设计方法学的推广，系统验证已成为设计工作中的瓶颈。

而SystemC由于是一种基于C++的语言，具有学习方便的优点，更重要的是其高速的仿真性能已得到多家设计公司的认可，因此被广泛用做SoC系统验证语言。

由OSCI组织的推荐，2004年SystemC 2.0被正式批准为IEEE标准，与SystemVerilog并列成为最主要的系统验证语言之一。

本书是根据J. Bhasker编写的A SystemC Primer第2版翻译的。

2004年起原书在数字集成电路验证界逐渐走红并非偶然。

在世界各地的大学中，教师和学生所熟悉的主要编程语言和环境是C++。

而SystemC是C++的一个子集合，不存在编程环境和学习基础的问题，所以比SystemVerilog更便于推广和应用。

本书的翻译工作安排如下：第2版序言、序言、前言、第1~6章及附录A、B和索引等由夏宇闻负责，第7~9章由神州龙芯IC设计公司的甘伟工程师负责。

全书最后的审校与定稿由夏宇闻负责。

在神州龙芯IC设计公司工作的工程师和实习研究生樊荣、洪雷、周鹏飞、刘家正、陈岩、李鹏、宋成伟、邢志成、管丽、徐伟俊、杨鑫、苏宇、张云帆、邢小地、李鹏、李琪、陈岩等认真阅读了最后完成的翻译稿，并提出了许多改进意见，使翻译工作的质量有了显著提高。

在翻译稿最后完成之际，谨向他们表示诚挚的感谢。

## <<SystemC入门>>

### 内容概要

SystemC既是系统级语言，也是硬件描述语言。

本书介绍的是systemC 2.0标准，主要介绍systemC有关硬件建模方面的语法特性，换言之，是介绍systemC的RTL可综合子集。

其主要内容包括：SystemC数据类型、组合逻辑建模、同步逻辑建模、三态驱动器建模、常用的设计函数模型、测试平台的编写及系统级建模的功能等。

随书附带1张光盘，内含本书所有例子的代码。

本书所有例子都经过Systemc 2.0.1的验证。

本书可作为想要了解和学习SystemC的设计工程师和系统工程师的参考书，也可用做大学讲授体系结构、数字设计或系统设计课程的教材。

## 作者简介

Bhasker, Bhasker是eSilicon公司的体系结构设计师。

他曾是朗讯技术公司和贝尔实验室技术部门的杰出成员，并在朗讯科技公司教授VHDL和Verilog HDL课程四年多。

他还编写了四本其他有关硬件描述语言和综合的书籍，其中包括最畅销的书A VHDL Primer和Verilog HDL Synthesis, A Practical Primer。

Bhasker拥有美国明尼苏达大学计算机科学系的博士学位和计算机技术学院的技术硕士学位，以及新德里印度技术学院电子工程系的学士学位。

## 书籍目录

## 第1章 绪论 1.1 什么是SystemC ?

## 1.2 为什么使用SystemC ?

1.3 设计方法学 1.4 SystemC的功能 1.5 SystemCRTL 1.6 本书的组织 1.7 练习题第2章 起步 2.1 基础知识 2.2 再举一个例子 2.3 描述的层次 2.4 功能的验证 2.5 练习题第3章 数据类型 3.1 值保持器 3.2 类型的总结 3.3 位类型 3.4 任意位宽类型 3.5 逻辑类型 3.6 任意位宽的逻辑类型 3.7 有符号的整数类型 3.8 无符号的整数类型 3.9 任意精度有符号的整数类型 3.10 任意精度无符号的整数类型 3.11 判断类型 3.12 用户定义的数据类型 3.13 推荐的数据类型 3.14 练习题第4章 组合逻辑建模 4.1 SC—MODULE 4.2 一个例子 4.3 端口和信号的读/写 4.4 逻辑操作符 4.5 算术操作符 4.5.1 无符号的算术运算 4.5.2 有符号的算术运算 4.6 关系操作符 4.7 向量和范围 4.7.1 常数索引 4.7.2 非常数索引 4.8 条件语句 4.9 开关语句 4.10 循环 4.11 方法 4.12 结构 4.13 多进程和 延迟 4.14 小结 4.15 练习题第5章 同步逻辑建模 5.1 触发器建模 5.2 多进程 5.3 带异步置位和清零端的触发器 5.4 带同步置位和清零端的触发器 5.5 多时钟和多相位时钟 5.6 锁存器建模 5.6.1 条件语句 5.6.2 开关语句 5.6.3 避免生成锁存器 5.7 小结 5.8 练习题第6章 各色各样的逻辑 6.1 三态驱动器 6.2 多个驱动源 6.3 无关项的处理 6.4 层次 6.5 模块的参数化 6.6 变量和信号的赋值 6.7 练习题第7章 建模示例 7.1 具有三态输出的参数化寄存器 7.2 存储器模型 7.3 有限状态机的建模 7.3.1 摩尔型有限状态机 7.3.2 米利型有限状态机 7.4 通用移位寄存器 7.5 计数器 7.5.1 模N计数器 7.5.2 Johnson计数器 7.5.3 格雷码双向计数器 7.6 Johnson译码器 7.7 阶乘的模型 7.8 只读存储器(ROM)的建模 7.9 练习题第8章 测试平台的编写 8.1 编写测试平台 8.2 仿真控制 8.2.1 scclock 8.2.2 sctrace 8.2.3 scstart 8.2.4 scstop 8.2.5 sctime—stamp 8.2.6 scsimulatio\_n\_time 8.2.7 scycle和scinitialize 8.2.8 sctime 8.3 波形 8.3.1 任意类型的波形 8.3.2 复杂的重复波形 8.3.3 生成一个衍生的时钟 8.3.4 从文件中读取激励 8.3.5 响应性激励 8.4 监视行为 8.4.1 判断仿真结果是否正确 8.4.2 把结果保存到文本文件中 8.5 更多示例 8.5.1 触发器 8.5.2 带同步输出的多路选择器 8.5.3 全加器 8.5.4 周期节拍级仿真 8.6 在sc—main内的语句顺序 8.7 跟踪记录集合类型 8.8 跟踪记录枚举类型 8.9 练习题第9章 系统级建模 9.1 SC—THREAD进程 9.2 动态敏感性 9.3 构造函数的参数 9.4 更多示例 9.4.1 求最大公约数 9.4.2 滤波器 9.5 端口、接口和通道 9.6 高级话题 9.6.1 共享的数据成员 9.6.2 定点类型 9.6.3 模块 9.6.4 其他方法 9.7 仿真算法 9.8 练习题附录A 运行环境 A.1 软件的安装 A.2 设计的编译 A.3 设计的仿真 A.4 设计的调试附录B SystemCRTL : SystemC的可综合子集 B.1 SystemC的特性 B.2 C++的特性参考文献索引

## &lt;&lt;SystemC入门&gt;&gt;

## 章节摘录

本章描述了什么是SystemC，为什么要用SystemC，并且叙述了使用SystemC的方法学。

本章还提供了从高层次角度观察到的SystemC能力。

1.1 什么是SystemC? SystemC是基于C++的编程语言。

C++是一种应用广泛的面向对象的建模语言。

由于添加了硬件建模功能，SystemC扩展了C++的能力。

SystemC在C++的基础上增加了一些重要概念，如并发（多个处理器同时执行）、定时事件和数据类型等概念。

SystemC还增加了一个类库，扩展了C++的能力。

该类库并不是C++的修改，而是一个由使用合法C++代码编写的函数、数据类型和其他语言结构函数所组成的库。

该类库提供了功能强大的新机制，这种机制可以为具有硬件时序、并发和响应行为的系统结构建模。

该机制只不过是在C++编程语言的类构造的高级层面上对该语言进行扩展而已。

该SystemC库提供许多构造函数，描述了硬件工程师们经常用到的概念，诸如信号、模块和端口等。

此外，该库还提供了像进程以及等待时钟负跳变沿到来等常见的能力。

SystemC并没有在C++编程语言上增加新的语法。

它只是定义了一个新的类库，因此仍旧是C++。

借助于类库中的这些类，用户便可以定义模块、进程以及通过端口的通信和信号。

这些信号可以处理范围广泛的数据类型，从位、位向量和标准的C++类型一直到用户定义的数据类型，诸如枚举类型和结构类型。

## <<SystemC入门>>

### 编辑推荐

《SystemC入门(第2版)》为国外数字系统设计经典教材系列之一，由北京航空航天大学出版社出版。

#### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>