QINGSONG XUEHUI

FPGA

SHEJI YU

KAIFA

轻松学会

FPGA

设计与开发

周 新 主 编 刘 杰 张海洁 副主编

一看就懂 一学就会

助你全面掌握FPGA设计与开发

化学工业出版社

QINGSONG XUEHUI
FPGA
SHEJI YU
KAIFA

轻松学会
FPGA
设计与开发

周 新 主 编 刘 杰 张海洁 副主编

化学工业出版社
·北京·

本书从FPGA开发入门和工程实践角度出发，深入浅出，逐步引导读者学习FPGA设计所需的基础理论基础和工具应用。书中针对Verilog HDL的基础语法进行了系统的介绍，对Verilog HDL中一些常接触并容易出错的概念进行了详细说明。同时，书中还介绍了在数字电路设计中常用的EDA工具，全书内容介绍深入浅出，结合作者多年来使用Verilog HDL的心得体会和积累，列举了丰富的设计实例，展现了许多仿真设计流程，全面总结和深入阐述了在Verilog HDL中一些设计技巧、设计理念，使读者快速、全面地掌握FPGA的设计思路和设计细节。

本书适合广大电路设计开发人员阅读，也可作为相关专业师生的教材。

图书在版编目（CIP）数据

轻松学会FPGA设计与开发/周新主编. —北京：化学工业出版社，2014.9
ISBN 978-7-122-21004-3

Ⅰ.①轻… Ⅱ.①周… Ⅲ.①可编程序逻辑器件-系统设计 Ⅳ.①TP332.1

中国版本图书馆CIP数据核字（2014）第133811号

责任编辑：刘丽宏　　文字编辑：吴开亮
责任校对：宋　玮　　装帧设计：刘丽华

出版发行：化学工业出版社（北京市东城区青年湖南街13号　邮政编码100011）
印　　装：化学工业出版社印刷厂
787mm×1092mm　1/16　印张18½　字数504千字　　2015年2月北京第1版第1次印刷

购书咨询：010-64518888（传真：010-64519686）　售后服务：010-64518899
网　　址：http://www.cip.com.cn
凡购买本书，如有缺损质量问题，本社销售中心负责调换。

定　　价：69.00元

前言

随着电子技术的不断发展，现在数字电路设计思想和方法都进入了一个全新阶段。FPGA/CPLD 可编程逻辑器件的功能越来越强大，应用也越来越广泛，为了能够快速地应用 FPGA/CPLD 进行项目开发，那么选择一种快速的开发方式很重要，而基于硬件描述语言的设计是最高效的一种。所以要想应用 FPGA/CPLD 进行系统设计，掌握一种硬件描述语言是必不可少的。

硬件描述语言中 Verilog HDL 和 VHDL 是首选。这两种语言都是 IEEE 标准的描述语言。本书就是基于 Verilog HDL 进行讲解的。Verilog HDL 与 C 语言语法相似，具有简洁、易掌握的特点。本书针对 Verilog HDL 的基础语法进行了系统的介绍，同时，对 Verilog HDL 中一些常接触并容易出错的概念进行了详细的说明。深入的讨论。书中详细说明了 Verilog HDL 的基础语法、常用语句、常用函数和任务，深入讲解了 Verilog HDL 的有关设计理念、设计技巧和使用要点。此外，书中还介绍了在数字电路设计中常用的 EDA 工具，这些工具可以帮助我们高效、高质量的将设计代码转换为物理电路，同时一些工具例如 Modelsim 还可以为我们提供模拟仿真手段，它们的应用极大地方便了系统设计。

本书在对基础语法讲解的同时结合作者使用 Verilog HDL 的心得体会和积累，以通俗易懂的方式与读者共享，摒弃晦涩难懂的说教方式，结合实例以简单明了的方式呈现给读者。相信可以很好地帮助读者熟悉设计开发流程并轻松进行设计。

本书由周新主编，刘杰 、张海洁副主编，参加编写的还有王佳、侯永恒、孙玉倩、贾永翠、张海洁、戴斌、邹全、张杰、孔海颖等。

因编者水平所限，书中不足之处难免，恳请广大读者及同行批评指正，以便改进。

编者

目录

第一章

Verilog HDL设计入门

第一节　Verilog HDL语言概述

Verilog HDL（Hardware Description Language）即Verilog硬件描述语言，它主要应用于数字电路和系统设计、数字电路和系统仿真等，即利用计算机和相关软件对用Verilog HDL等硬件语言建模的复杂数字逻辑电路设计进行仿真验证，再利用综合软件将设计的数字电路自动综合，以得到符合功能需求并且在相应的硬件电路结构上可以映射实现的数字逻辑网表（Netlist），然后布局布线，根据网表和选定的实现器件工艺特性自动生成具体电路，同时软件生成选定器件的延时模型，经过仿真验证确定无误后，用来制造ASIC芯片或写入FPGA和CPLD器件中，最终实现电路设计。

在EDA领域中把用Verilog HDL或VHDL等其他硬件设计语言建立的数字模型称为Soft Core即软核，用Verilog HDL或VHDL等其他硬件设计语言建模和综合后生成的网表称为Hard Core即固核。在数字电路和系统设计时可以将这些设计封装为模块，重复利用，以缩短开发时间，提高产品的开发效率和设计效率。

目前常用的有两种标准的硬件描述语言是Verilog和VHDL。1998年我国国家技术监督局正式将《集成电路/硬件描述语言Verilog》列入了国家标准，国家标准编号为GB/T 18349—2001，从2001年10月1日起正式实施。近年来，该标准的制定对我国集成电路设计技术的发展起到了重要的推动作用，现在EDA技术在我国集成电路设计领域越来越普遍，扮演的角色越来越重要。

硬件描述语言是继EDA原理图（Shematic）设计输入方式之后出现的，它是目前应用于ASIC、FPGA、CPLD等开发的最重要的一种设计输入方式，它与原理图（Shematic）设计输入方式相比，具有以下特点：

① 可以自顶向下设计，利于模块划分及模块复用；

② 可移植性好、通用性强，设计不因芯片的工艺和结构变化而变化；

③ 便于维护升级；

④ 易于实现较复杂的设计。

硬件描述语言是一种分层次的设计语言，这样对于设计开发更灵活方便，最常用的层次概念有系统级（System Level）、功能模块级（Functional Model Level）、行为级（Behavioral Level）、寄存器传输级（Register Transfer Level，RTL）和门级（Gate Level）。其概念如下。

（1）系统级　在进行大型复杂系统的设计与实现之前，首先对要实现的设计进行详细的分析和系统规划并利用硬件设计语言进行描述。此阶段硬件设计描述侧重于整体系统划分和实现，同时对本阶段的仿真则侧重于对整个系统的功能和性能指标验证。系统级常用语言包括C语言和抽象程度较高的HDL语言，例如SystemC、CoWareC、SystemVerilog、Superlog等。

（2）功能模块级　功能模块级主要完成描述系统的功能划分，其主要工作内容是将系统整体功能按需求进行划分，并完成实现具体功能模块，大致确定模块间的接口，例如时钟、控制信号、读写信号、数据流等。在某些情况下，还需要根据系统需求，完成每个模块或进程的时序约束描述。通常每个系统功能都可以利用多种方式实现，所以在本层次中需要对多种实现方式进行对比，以便筛选出性能指标最优和实现效率最高的设计方案。这个层次的仿真主要是考察每个功能的功能和基本时序情况。这个层次适合用抽象程度较高的HDL语言描述。本层次仿真主要是针对每个功能的时序是否满足设计要求。功能模块级适合利用较高层次的硬件描述语言实现。

（3）行为级　经过行为级模块描述可以得到每个模块的具体设计，此时模块间的所有接口，例如时钟、控制信号、读写信号、数据流等都已清晰地确定，同时每个模块的功能已经明确，在FPGA和CPLD的设计流程中，常用行为级描述方式编写测试激励、延时、输入输出监测描述等，以便进行仿真。行为级描述常用HDL语言如Verilog和VHDL等。

（4）寄存器传输级　寄存器传输级描述通常是利用综合工具将设计自动综合为门级间网表，此时模块间的引脚接口、功能结构、时钟等都已经明确，通过相关综合工具的自动编译、映射等步骤，可以将模块中描述的功能用FPGA内部的查找表（LUT）、寄存器（Register）和基本的硬件原语（Primitive）实现。硬件原语是FPGA的基本功能模块和实现单元，如Block RAM、DLL/PLL等，不同厂家的FPGA具有不同的硬件原语结构。目前最常见FPGA、CPLD设计流程如下。

① 首先根据功能需求，设计RTL级代码。

② 使用综合工具，综合生成网表。

③ 利用厂商的相关布局布线工具，实现布局布线。

④ 利用厂商相关开发软件生成器件下载配置文件。

其中，RTL代码设计是整个设计的第一步，直接决定着设计的功能和效率。好的RTL代码的判定标准是既能满足逻辑功能又能使最终设计实现速度和面积最优平衡。好的代码设计不仅需要最优的实现方案、匹配的综合开发软件，同时还需要具有良好的Coding Style即代码设计规范，一个具有良好Coding Style的代码设计不仅有助于高效的设计开发并且方便以后的升级维护。关于Coding Style，在后面章节进行详细介绍。

（5）门级　在目前FPGA、CPLD的设计中，大部分设计依靠专业综合工具自动完成从RTL级代码向门级代码的转换，高效的综合工具替代设计者完成复杂烦琐门级描述，所以现在设计者直接利用硬件描述语言设计门级电路的情况很少。目前只有在ASIC和FPGA设计中有些速度和时序要求较高的情况下会直接使用门级描述实现。门级描述即在整个设计中利用逻辑门实现，通过不同逻辑门的组合来实现设计的所有功能。

第二节 数字电路设计方法简介

一、 布尔方程设计

很多基于门和触发器的逻辑电路传统上都是用布尔方程设计的，如果不清楚门和触发器这些基本的组件，是很难设计数字电路系统的。利用布尔方程式可以化简设计，同时由此产生了很多技术来优化设计，包括简化方程式来节约设计中所使用的门和触发器。

由于利用布尔方程设计对于每一个触发器和逻辑门组成的块都需一个方程进行标示，因此当进行拥有成百上千的触发器的设计时实现起来非常不实际，因为这将导致很多的逻辑方程。

理论上任何系统都可以用布尔方程设计方法进行设计，但是要处理描述系统的上千个逻辑方程是非常困难的，所以在实际操作中并不适用。

二、 原理图的设计

原理图的设计方法扩展了布尔方程设计方法的设计能力，它既可以利用逻辑门和触发器，也可以利用封装好的电路模块进行设计。原理图设计方法突出了设计的层次结构，层次结构设计使得进行上千元件的电路设计变得更简单。对于设计人员来说，原理图设计方法比布尔方程设计方法更友好和更加容易实现电路设计。

由于原理图提供图形表示的设计方法，它更清楚地显示不同设计模块的关系，所以大部分设计人员喜欢用原理图设计方法来进行电路设计。在较早时期，原理图设计得到了发展，被设计人员广泛应用，原理图设计方法被认为是一种最优的设计方法。但是近些年来，电子技术不断发展，系统功能越来越强大，电路结构越来越复杂，器件密度越来越大，设计人员发现，利用原理图设计已不能满足研发的需求，原理图设计方法在当今的设计中显得能力有限，而且这种设计方法非常耗时，不易升级和维护。

以上介绍的两种传统设计方法尽管容易使用，但都有一些缺点，最重要的缺点是一个系统常被定义为一个内部相连的网络，但这不是建立系统的真正定义。

另外一个缺点很难处理太复杂的设计。处理上百个的逻辑方程虽然很困难但还是可行的，但是处理上千个逻辑方程就很难想像了。一个普遍接受的事实是：超过 6000 个门的原理图就很难理解。由于最新的集成电路包含上百万个门，而且这种密度还在增加，这就需要一种新的设计方法。

三、 硬件描述语言

相对于以上提到的几种传统的设计方法，硬件描述语言将使电路设计方案的实现变得更加容易和方便，通用性和可行性更好，因为 HDL 的灵活性和可操作性强，所以在对设计进行修改时相对于传统的设计方法更加方便。传统设计的一个最主要的缺点就是需要将设计转换成布尔方程或原理图，而通常这一步对于复杂电路来说不易实现。硬件描述语言可以替代传统设计方法方便地进行电路设计。大部分 HDL 允许对时序电路使用有限状态机，而对于组合电路可以使用真值表，这种设计描述可以自动被转化为 HDL，然后经过自动综合实现电路设计。

HDL 在各种不同密度可编程逻辑器件中得到了应用，从简单的 PLD 到复杂的可编程逻辑器件 CPLD 和现场可编程阵列 FPGA 都能较好地应用。目前有好几种硬件描述语言在使用，最受欢迎的有在小规模器件中使用的 Abel、在大规模的 FPGA 和 CPLD 中使用的 VHDL 及 Verilog HDL。

HDL 可以处理不同的描述层次。从硅片级到复杂的系统有几个级别的系统描述，这些级可用它们的结构和行为描述，今天的技术已经相当复杂，因此，传统的设计方法不能覆盖不同设计层的方方面面。近些年来，一些面向低层（硅）和设计处理集成化的工具已经出现。

前面所描述的设计方法好像很完美，但是它们大部分在整个设计流程中只起很小的作用。要完成一个设计，可能需要从一种设计描述转到另一种，或从一种工具转到另一种。这就产生了工具兼容及学习新环境的问题。

这个问题可由一个覆盖所有设计层和类型的可能描述方法解决。Verilog 就是一个能处理不同设计层的设计工具，它可在不同设计阶段形成标准化数据传输。

第三节　Verilog HDL 与 VHDL 对比

Verilog HDL 很容易理解且方便使用。在近些年，Verilog HDL 已成为需要仿真和综合的 EDA 行业应用中的首选语言，但是它同时也存在一缺点，就是在系统级别的描述结构比较匮乏。

相对于 Verilog HDL，VHDL 相对复杂些，难以学习和使用。但是它可允许的编码风格丰富，提供了许多便利。由于 VHDL 特别适合处理复杂设计，因此它也获得了广泛使用。

Verilog HDL 和 VHDL 都是应用于逻辑设计的硬件描述语言，并且都已成为 IEEE 标准。VHDL 在较早的 1987 年成为 IEEE 标准，而 Verilog HDL 则在 1995 年才正式成为 IEEE 标准。Verilog HDL 是从一个普通的民间公司的“私有财产”转化而来的，凭借其优越性和更强的生命力，才成为 IEEE 标准；而 VHDL 是美国军方组织开发的，VHDL 的英文全名为“VHSIC Hardware Description Language”，而 VHSIC 则是“Very High Speed Integrated Circuit”的缩写词，意为“甚高速集成电路”，故 VHDL 准确的中文译名为“甚高速集成电路的硬件描述语言”。

Verilog HDL 和 VHDL 同为描述硬件电路设计的语言，其共同的特点在于：

① 能形式化地抽象表示电路的行为和结构；

② 支持逻辑设计中层次与范围的描述；

③ 可借用高级语言的精巧结构来简化电路行为的描述；

④ 具有电路仿真与验证机制以保证设计的正确性；

⑤ 支持电路描述由高层到低层的综合转换；

⑥ 硬件描述与实现工艺无关；

⑦ 便于文档管理；

⑧ 易于理解和设计重用。

Verilog HDL 和 VHDL 又各有其自己的特点。由于 Verilog HDL 早在 1993 年就已推出，至今已有 20 多年的应用历史，因而 Verilog HDL 拥有更广泛的设计群体，成熟的资源也远比 VHDL 丰富。与 VHDL 相比，Verilog HDL 的最大优点为：它是一种非常容易掌握的硬件描述语言，与 C 语言的语法相似，只要有 C 语言的编程基础，通过几十学时的学习，再经过一段时间的实际操作，一般可在 2～3 个月内掌握这种设计技术。而掌握 VHDL 设计技术比较困难，这是因为 VHDL 不很直观，需要有 Ada 编程基础，一般认为至少需要半年以上的专业培训，才能掌握 VHDL 的基本设计技术。目前版本的 Verilog HDL 和 VHDL 在行为级抽象建模的覆盖范围方面也有所不同，一般认为 Verilog HDL 在系统级抽象方面比 VHDL 略差一些，而在门级开关电路描述方面比 VHDL 强得多。图 1-1 所示为 Verilog HDL、VHDL 及由 HDL 衍生出的几种硬件描述语言如 System Verilog、Superlog、SystemC 等的使用层次对比图，其中 System Verilog、Superlog、SystemC 这些高级 HDL 语言的语法结构更加丰富，

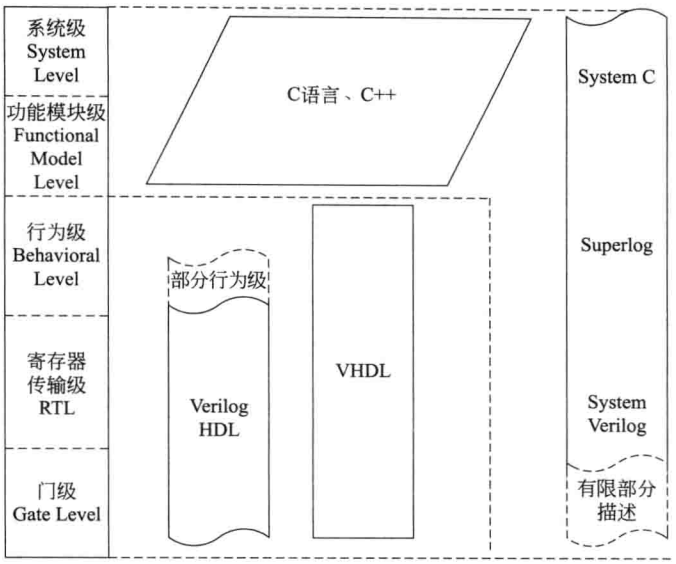

图 1-1 语言使用层次对比图

更适合作系统级、功能模块级等高层次的设计描述和仿真。

硬件描述语言仍处在不断完善的过程中，2000 年公布的 IEEE Verilog 2001 标准，使 Verilog 在系统级和可综合性能方面都有大幅度的提高，因此 Verilog HDL 作为学习 HDL 设计方法的入门和基础是比较合适的。学习掌握 Verilog HDL 建模、仿真和综合技术不仅可以使读者对数字电路设计技术有更进一步的了解，而且可以为以后学习高级的行为综合和物理综合打下坚实的基础。

第四节 Verilog HDL 与 C 语言对比

Verilog HDL 语言是在 C 语言的基础上发展而来的，Verilog HDL 和 C 语言在语法结构上有许多的相似之处，继承和借鉴了 C 语言的很多语法结构。当然，应该明确在 FPGA/CPLD、ASIC 的逻辑设计领域所采用的硬件描述语言（HDL）与软件语言例如 C、C++等是有本质区别的！以 Verilog HDL 语言为例，虽然 Verilog 很多语法规则和 C 语言相似，但是 Verilog 作为硬件描述语言，它的本质作用在于利用语言描述要实现的硬件电路。设计者要清楚硬件描述语言最终实现的硬件电路结构，做到对描述语言的更深层次的理解；设计者应该认识到 Verilog HDL 虽然是采用了 C 语言形式的硬件的抽象，但它的最终实现结果是芯片内部的实际电路。所以评判一段 HDL 代码的优劣的最终标准是：其描述并实现的硬件电路的性能（主要是面积和速度间的最优平衡）。评价一个设计的代码水平较高，仅仅是说这个设计由 HDL 代码向硬件这种表现形式转换得更流畅、合理。而一个设计的最终性能，在更大程度上取决于设计工程师所构想的硬件实现方案的效率以及合理性。

进行硬件电路设计不能按照 C、C++的代码设计规范进行，片面追求代码的整洁、简短是错误的，是与评价 HDL 的标准背道而驰的。正确的编码方法是：首先要做到对所需实现的硬件电路“胸有成竹”，对该部分硬件的结构与连接十分清楚，然后用适当的 HDL 语句表达出来即可。

另外，在前面的小节中已经提到 Verilog HDL 作为一种硬件描述语言是分层次的。主要的设计层次有：系统级、功能模块级、行为级、寄存器传输级、门级等。系统级、功能模块

级和行为级与C语言更相似，可用的语法和表现形式也更丰富。自RTL级以后，HDL语言的功能就越来越侧重于硬件电路的描述，可用的语法和表现形式的局限性也越大。相比之下，C语言与系统级、功能模块级和行为级的Verilog HDL描述更相近一些，而与RTL级、Gate级描述相比，从描述目标和表现形式上都有较大的差异。

C语言与Verilog HDL间的对比如下。

① C语言由函数组成，Verilog HDL由模块（module）组成。

② C语言通过函数名及其端口变量实现调用，Verilog HDL也通过模块名和端口变量实现调用。

③ C语言有主函数main（），Verilog HDL由一个或几个module组成，但其中必有一个顶层模块，包含芯片系统与外界的所有I/O信号。

④ C语言是顺序执行，而Verilog HDL的所有module均并发执行。

相对于C语言来说，Verilog HDL具有以下特点。

① 既可以进行电路仿真，也可以进行电路设计。

② Verilog HDL可以在同一个电路模块中进行不同抽象层次的描述设计。从门、RTL到行为等各个层次设计者均可以对电路模型进行定义。同时，设计者只需要学习一种语言就能够使用它来描述电路的激励，进行层次化设计。

③ 目前EDA技术领域中绝大部分的综合工具都支持Verilog HDL，这也是设计者首选Verilog HDL作为设计语言的重要原因之一。

④ 各个制造厂商都提供关于Verilog HDL的逻辑仿真元件库，因此使用Verilog HDL进行设计，即可在更广泛的范围内选择委托制造的厂商。

⑤ Verilog语言最重要的特性之一是编程语言接口（PLI），设计者可以利用此功能通过自己编写C代码与Verilog HDL设计的电路模块内部数据结构进行传输访问，同时设计者可以使用PLI按照自己的需求来配置Verilog HDL仿真器，这样极大地方便了验证仿真。

表1-1和表1-2为对比Verilog HDL与C语言的相似之处。

表1-1 Verilog HDL与C语言的对比一

Verilog HDL	C语言	Verilog HDL	C语言
if-else	if-else	case	case
for	for	define	define
while	while	printf	printf

表1-2 Verilog HDL与C语言的对比二

Verilog HDL	C语言	Verilog HDL	C语言
+	+	!	!
−	−	>	>
*	*	<	<
/	/	>=	>=
%	%	<=	<=
&&	&&	==	==
||	||	~	~

下面进行举例说明 RTL 级 Verilog HDL 描述语法和 C 语言描述语法的一些区别。

在 C 语言的代码设计中，为了代码执行效率高，遵循代码表述简洁的原则，经常用到如下所示的 for 循环语句：

```
for(i = 0;i<32;i + + )
DONEXT();//调用函数。
```

但是在用 Verilog HDL 进行硬件电路设计时，除了在进行描述仿真测试激励即 testbench 时常会用到 for 循环语句外，极少在 RTL 级编码中使用 for 循环。其原因是 for 循环会被综合器展开为所有变量情况的执行语句，每个变量独立占用寄存器资源，每条执行语句并不能有效地复用硬件逻辑资源，造成巨大的资源浪费。for 语句在 RTL 级编码设计中的可综合性不强。那么在 RTL 硬件描述中，遇到类似情况该如何实现呢？首先搞清楚设计的时序要求，可以设计一个 reg 型计数器，在每个时钟沿累加，并在每个时钟沿判断计数器情况？根据需求在相应的计数器值执行相应的处理，可以采用 if…else 语句进行判断执行或者在较复杂的情况下利用 case 语句罗列出相应的执行操作。如下所示：

```
reg [4 : 0] COUNTER;//计数寄存器
wire [4 : 0] COUNTER_N;计数寄存器的下一个状态
always @(posedge SYSCLK or negedge RST_B)//时序逻辑电路,系统时钟 SYSCLK 的上升沿或者系统复位
                                          RST_B 的下降沿触发
begin
if (! RST_B)
COUNTER <= 4'b0;
else
COUNTER <= COUNTER_N;
end
assign  COUNTER_N = COUNTER + 4'h1;
always @ (posedge CLK or negedge RST_B)
begin
case (COUNTER)
4'b0000;//需要执行的操作
4'b0001;//需要执行的操作
default;// 需要执行的操作
endcase
end
```

再例如，C 语句描述中有 if…else 和 switch 条件判断语句，其语法如下所示：

```
if(comeon)
... ;                  //若 comeon 为真则执行此操作
else
...;                   //否则执行此操作
```

switch 语句的基本格式是：

```
switch(comeon)
{
case comeon1:...   //若 comeon 的值为 comeon1 则执行此操作
break;
case comeon2:...   //若 comeon 的值为 comeon2 则执行此操作

break;             //...
...
default:...        //...
break;
}
```

Verilog HDL 中也有与 C 语言中相对应的 if…else 语句和 case 语句，if…else 语句的语法