

J. Bhasker 著
孙海平 等译

Verilog[®]

HDL 综合

实用教程

Verilog[®] HDL Synthesis
A Practical Primer



清华大学出版社

TP312
1262

J. Bhasker 著

*Distinguished Member of Technical Staff
Bell Labs, Lucent Technologies*

孙海平 等译

Verilog[®] HDL 综合 实用教程

Verilog[®] HDL Synthesis
A Practical Primer

清华大学出版社
北京

J. Bhasker

Verilog® HDL Synthesis, A Practical Primer

EISBN: 0-9650391-5-3

Original English language edition published by Star Galaxy Publishing.

Copyright © 1998 Lucent Technologies. All rights reserved.

Chinese translation edition is published and distributed exclusively by Tsinghua University Press under the authorization by Star Galaxy Publishing, in the territories throughout the world.

本书中文翻译版由美国 Star Galaxy Publishing 授权清华大学出版社在全球范围内独家出版发行。

北京市版权局著作权合同登记号 图字 01-2003-4338

本书封面贴有清华大学出版社激光防伪标签, 无标签者不得销售。

图书在版编目(CIP)数据

Verilog® HDL 综合实用教程/(美)巴斯克尔(Bhasker, J.)著; 孙海平等译. —北京: 清华大学出版社, 2004. 1

书名原文: Verilog® HDL Synthesis, A Practical Primer

ISBN 7-302-07714-2

I. V… II. ①巴… ②孙… III. 硬件描述语言, Verilog HDL—程序设计—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2003)第 108656 号

出 版 者: 清华大学出版社

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

社 总 机: 010-62770175

地 址: 北京清华大学学研大厦

邮 编: 100084

客户服务: 010-62776969

责任编辑: 张 靓

封面设计: 常雪影

印 刷 者: 北京密云胶印厂

装 订 者: 北京市密云县京文制本装订厂

发 行 者: 新华书店总店北京发行所

开 本: 185×230 印张: 11.75 字数: 241 千字

版 次: 2004 年 1 月第 1 版 2004 年 1 月第 1 次印刷

书 号: ISBN 7-302-07714-2/TP·5649

印 数: 1~4000

定 价: 24.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题, 请与清华大学出版社出版部联系调换。联系电话: (010)62770175-3103 或 (010)62795704

译者序

数字集成电路在过去 30 年里得到了长足发展,EDA(电子设计自动化)技术起到了至关重要的作用。其中,用于表达设计对象的硬件描述语言(HDL)采用形式化方法,不仅可以准确、直观地对数字电路进行建模和仿真,而且极大地提高了电子设计的效率和产出,为顺应半导体工艺技术和应用需求的急速发展提供了可能。目前我国 ASIC(专用集成电路)设计业的基本状况是芯片设计开发工作远远滞后于电子产品发展的需求,滞后于芯片生产线的吞吐能力。为了适应目前系统芯片(System-on-a-Chip, SoC)时代的设计需求,提升设计能力和效率,国内推广和开展基于 HDL 和各种 EDA 工具的设计方法已经成为迫切的需要。

Verilog 作为当今国际主流的 HDL 语言,20 世纪 90 年代初成为 IEEE 标准之后,在数字电路和芯片的前端设计中得到了更为广泛的应用。Verilog 最初是为了仿真和模拟数字电路和数字系统的运行而设计出的一种形式化语言,随着半导体工艺的迅猛发展,电路规模和芯片集成密度迅猛扩张,手工绘制电路图已不能满足设计需求了,此时人们就把作为模拟工具的硬件描述语言发展成电路设计的手段,在 EDA 工具的辅助下把采用硬件描述语言描述的设计对象自动综合成实际电路。

Verilog 的发展历程决定了其语义的丰富性。这种丰富性一方面决定了其模拟能力十分强大,另一方面决定了并不是每种描述出来的现象都可以直接生成实际电路,语言结构的仿真功能与所综合出的电路的功能可能不一致,各种语言结构搭配起来使用产生的似是而非的微妙差异都会导致综合出的电路功能完全不同。为避免歧义和误解,2002 年 11 月 IEEE 正式制定了 Verilog 寄存器传输级综合标准。

本书作者 Bhasker 博士是 Verilog 综合互操作性工作组的主席,长期从事有关 HDL 的研究。他不仅对各种 HDL 的语义有着深入、准确的理解,而且直接主持制定了 Verilog HDL 综合的 IEEE 标准。本书是这方面的权威著作。鉴于国内介绍 HDL 的各种书籍囿于语法层面的介绍,而没有着眼于综合的角度来讲解 Verilog 语言的语义本质,不能解决初学者和许多设计人员在设计、调试时的困惑,清华大学出版社适时引进并出版这本“Verilog® HDL Synthesis, A Practical Primer”以及 Bhasker 博士撰写的另一本著作“A SystemC™ Primer”的中文翻译版,对国内迫切需要此类图书的读者犹如久旱逢甘霖。

本书的鲜明特色在于帮助读者全面、正确地理解 Verilog 硬件描述语言的综合。本书以电路综合为目标,针对各种语言结构逐一讨论了其可综合性、仿真与综合时的语义差别以及相关的各种用法,给出了大量示例,对各种似是而非的用法作了对比,指出了其语义差别和所综合出的电路在功能上的差异。本书的另一特色在于详细介绍了设计模型的

优化技术和验证技术。

本书内容全面、深入浅出、适用面广,对于已经采用或打算采用 Verilog 语言作为电路设计手段的学生和工程人员而言是一本不可多得的好书。

全书由孙海平、刘方海和郑静翻译,孙海平统稿,何伟、徐学迅和郑静等作了修订和校对。在此,谨向为本书出版付出辛勤劳动的所有人员致以诚挚的感谢!

在翻译过程中我们力求译意准确,但限于水平,必然存在错误和不足之处,恳请读者通过电子邮件 hp.sun@263.net 批评指正。

译 者

2003 年 10 月

原 书 序

关于 Verilog HDL 综合的讨论早在 1988 年就已经展开。但时至今日,此领域的优秀教材尚未囊括其基本概念。这本关于 Verilog HDL 综合实用教程全面地介绍了这一新技术。它通过提供便于理解的、与综合技术相关的 Verilog 语义,揭开了 HDL 综合的神秘面纱。本书作者 Bhasker 是综合领域的专家,在此领域已研究了 14 年多。身为 IEEE 工作组主席,他利用自己的专长领导了 Verilog RTL 综合标准(PAR 1364.1)^①的开发,此项工作是以 1998 年 4 月公布的 OVI^② RTL 综合子集为基础开展的,而 Bhasker 正是 OVI 的 RTL 综合标准的缔造者之一。

Bhasker 的这本著作作为学生和刚刚从事逻辑设计的人提供了系统掌握 Verilog HDL 综合的捷径。此书文字浅显易懂,列举了大量可综合的 Verilog HDL 模型示例。读者可以系统地了解 Verilog HDL 的语言结构、它们在综合时的含义、综合设计技术如何把这些结构转换成逻辑门电路,以及它们对设计验证的影响。此书给出了大量 Verilog HDL 模型示例及其等价的逻辑门电路。这些示例虽然简单,但展示了不同的逻辑建模方式,如组合逻辑、时序逻辑、基于寄存器和锁存器的设计、有限状态机、算术单元等。

本书不仅为初学者提供了 HDL 综合方面的知识,还讨论了一些高级论题,诸如如何从综合模型得出优化的逻辑等。其中,资源共享和分配是模型优化的论题之一。另一个独特的论题是对设计的验证。本书阐述了编写综合模型以确保得到可预测和可验证的结果的基本原则。尽管有一章立足于仿真,但是所讨论的那些概念同样可用于形式化验证。

本书是第一本对 Verilog HDL 综合进行全面论述的著作。作者 Bhasker 在 Lucent Technologies(朗讯科技)讲授 Verilog HDL 语言和 Verilog HDL 综合长达三年多。这本书是他 14 年来在 Verilog HDL 综合方面的研究成果的总结。尽管本书定位于初学者,但是专业人员也可从基本原则及综合建模的高级论题中获益。不容置疑,知识产权(IP)开发人员应遵循本书所推荐的建模方式。

Vassilios C. Gerousis

Motorola 高级工程师 Phoenix, Arizona

Open Verilog International 技术协作委员会主席

^① 译者注:该工作组提交的“Verilog 寄存器传输级综合 IEEE 标准”已于 2002 年 11 月 10 日被批准成为 IEEE 标准,编号是 IEEE Std 1364.1TM-2002。

^② Open Verilog International

前 言

本书是 Verilog HDL 寄存器传输级综合方面的实用指南，提供了大量可综合的 Verilog HDL 示例，详细介绍了 Verilog HDL 综合所支持的各种语法结构，并且用示例说明了如何把这些可综合的结构搭配起来对各种硬件元件进行建模。本书还详细讲解了造成设计模型和综合出的网表功能不一致的常见原因，并给出了避免产生这些错误的建议。

对很多人而言，综合看起来像是一个黑箱，输入的是 Verilog HDL 描述的设计，而输出的是门级网表。这种黑箱方式看起来似乎存在着某些奥妙。正确理解综合过程中出现的各种变换，才能充分利用综合系统并充分发挥它的长处。Verilog HDL 终究是一种建模语言，本书的目的就是要通过介绍从硬件描述语言模型到网表这一综合过程中出现的各种变换来揭示黑箱中的奥秘。

Verilog 硬件描述语言通常指的是作为 IEEE 标准 (IEEE Std 1364) 的 Verilog HDL。它可以用来描述时序的和并发的行为，也可用来描述模型的结构。它支持在从体系结构级到开关级的多个抽象层次上描述设计。该语言支持对设计进行层次化建模，此外还提供了大量内建的基本元件，包括逻辑门和用户自定义的基本元件。各种语言结构都具有精确的仿真语义，因此可以用 Verilog HDL 仿真器来验证采用该语言编写出的模型。

通常，对于不同的人而言，“综合”有不同的含义。本书中，综合指的是对 Verilog HDL 描述的设计进行综合，该设计描述了组合逻辑和（或）时序逻辑。对于时序逻辑，清晰地描述了其受时钟控制的行为。这排除了讨论逻辑综合（用逻辑门基本元件描述的设计）和高层次综合（不用时钟信息来指定设计对象的行为）。综合过程把 Verilog HDL 模型转换成门级网表。通常假定目标网表是被模拟的逻辑与工艺无关的表示形式。目标工艺包含诸如逻辑门之类与工艺无关的通用功能块，以及诸如算术逻辑单元和比较器之类的寄存器传输级功能块。对于综合流程的后续阶段，如工艺转换（即从通用门到库中特定部分的映射）和模块绑定（即采用逻辑门基本元件来构建寄存器传输级功能块），本书均未涉及。

之所以很难编写一本关于综合的书，是因为其具有发展迅速的特性。因此，本书所提供的是大体上成立的基本信息，尽可能避开了模棱两可的论题（包括与特定实现相关的问题）。由于 Verilog HDL 语言的丰富性，描述一种行为可能存在着不止一种方式，本书提出了一两种可综合的建模方式。另外，并非该语言中的所有结构都是可综合的，因为 Verilog HDL 最初是被设计成一种仿真语言。因此，本书将介绍主流综合系统所支持的各种结构。

同样,本书还避免提及特定综合工具厂商提供的不同特性。然而,某些特定情况下也有必要介绍某种实现示例。此时,所介绍的特性都在朗讯科技的贝尔实验室开发的 ArchSyn (14.0 版本) 综合工具中得到了实现。

注意:不是所有的综合系统都支持本书中描述的 Verilog HDL 结构。任何关于综合系统的专有特性的更多细节,读者都需要及时参考相应厂商的文档资料。

笔者是 Verilog 综合互操作性工作组的主席,该工作组目前正在开发 RTL 级综合的 IEEE 标准。

本书假定读者已经具备了 Verilog HDL 语言的基础知识。Star Galaxy 出版社的“A Verilog® HDL Primer”是一本关于 Verilog HDL 语言入门的好书。

本书面向电子工程师,尤其是那些对于理解综合的技巧感兴趣的电路与系统设计人员。本书不打算解释任何综合算法。作者相信一旦理解了综合结果会是什么,就能够编写出有效的设计模型,从而对综合出的设计对象的品质能有所控制。这是因为综合出的电路结构易受编写模型的方式的影响。

本书可以用作高校教材。在电子工程专业的教学大纲中,本书可以在计算机辅助设计方面的 VLSI (超大规模集成电路) 课程中使用。学生可使用本书编写多种模型,并在任何可用的综合系统对它们加以综合,以研究综合过程中出现的各种变换。在计算机科学专业的课程(如计算机辅助设计的算法课程)中,学生可以编写简单的综合程序来识别 Verilog HDL 语法的一个子集并生成综合的网表。本书中的示例可用作测试用例以供理解所生成的网表。

专业工程师将本书作为参考书也可以从中获益。工程师们可以在大量模型示例及其综合出的网表中直接寻找自己感兴趣的部分加以研究。

本书的组织结构

第 1 章介绍综合过程的基础知识,诸如什么是连线、触发器和状态以及如何确定对象的大小之类的内容。

第 2 章介绍 Verilog HDL 结构向逻辑门的映射。通过组合逻辑的示例说明如何把 Verilog HDL 结构变换成基础逻辑门以及它们的互连结构。还通过异步置位和清零、同步置位和清零、多时钟、多相位时钟等建模示例介绍了各种模拟时序逻辑设计的方式。

有时也有必要使用预先设计的功能块,因此第 2 章进一步介绍了如何对结构进行建模,包括在行为模型中采用部分结构建模。

第 3 章介绍如何把 Verilog HDL 的各种结构搭配起来对硬件元件进行建模。虽然第 2 章介绍了 Verilog HDL 向逻辑门的映射,但本章介绍的是另一方面,即如何用 Verilog HDL 来建立硬件元件的可综合模型。本章提供了许多通用硬件元件的详尽示例,如多路选择器、计数器、译码器以及算术逻辑单元等。

第 4 章介绍可应用于 Verilog HDL 模型以综合出优质网表的各种有效技术。

本章介绍的各种优化手段如果不能由综合系统自动实现,则需要由设计者手工加以实现。

有了 Verilog HDL 综合模型,通常还有必要用输入的设计模型来验证综合出的网表。第 5 章提供了编写检验综合结果的测试平台的策略。因为 Verilog HDL 不是为了综合而专门设计的语言,设计出的模型与综合出的网表可能会出现功能上的不一致。本章解释了产生那些分歧的原因。

为了说明典型的综合系统所支持的 Verilog HDL 可综合子集,附录 A 介绍了 ArchSyn 综合系统所支持的结构。但是,不同综合系统的可综合子集之间是存在差别的。

附录 B 给出了本书中综合出的网表用到的各种逻辑门的说明。

注意:本书展示的综合出的网表不是优化过的网表,因此在某些情况下这些逻辑可能不是最理想的。这是可以接受的,因为本书的目的是体现 Verilog HDL 到逻辑门之间的变换,而不是用来说明各种逻辑优化技术。本书中的有些网表已经被有目的地优化过了,因此可以把那些网表作为经典记录下来。

约定

本书的很多地方使用了词语“设计者”,它泛指任何阅读本书的读者。此外,术语“综合工具”和“综合系统”在本书中互换使用,它们指的都是读入 Verilog RTL 模型并生成门级网表的程序。

本书出现的所有 Verilog HDL 描述,保留字都用黑体印刷。有时候 Verilog HDL 源程序中出现的省略号 (...) 用来表示与当时讨论的内容不相关的代码。

本书采用的所有示例都已使用 ArchSyn 综合系统加以综合。附录 B 介绍了综合出的网表中使用的各种逻辑门。

致谢

衷心感谢以下为本书审稿和提供建设性意见的个人,他们提出了很多发人深省的评论,对于改进本书有着直接的帮助。真诚地感谢他们在百忙之中抽出时间和精力审阅本书。

- a) Cliff Cummings, Sunburst Design 公司
 - b) Joe Pick, Synopsys 公司
 - c) Doug Smith, VeriBest 公司
 - d) Egbert Molenkamp, 荷兰 Twente 大学
 - e) Carlos Roman、Jenjen Tiao、Jong Lee 和 Sriram Tyagarajan, 朗讯科技, 贝尔实验室
 - f) Jim Vellenga、Ambar Sarkar, Viewlogic Systems 公司
- 非常感谢他们!

还要向 Hao Nham 表示感谢，他为我在贝尔实验室创建了良好的工作氛围，并鼓励我在日常工作之外完成本书的编写。

当然，如同笔者撰写其他书那样，如果没有生活、家庭、我的妻子 Geetha 以及两个孩子 Arvind 和 Vinay 不断带给我欢乐、喜悦和继续写作的动力，本书就不可能得以完成。

J. Bhasker

1998 年 8 月

目 录

译者序	I
原书序	III
前言	IV
第1章 基础知识	1
1.1 什么是综合?	1
1.2 设计流程中的综合	2
1.3 逻辑值体系	4
1.4 位宽	5
1.4.1 数据类型	5
1.4.2 常量	7
1.4.3 参数	8
1.5 值保持器的硬件建模	8
第2章 从 Verilog 结构到逻辑门	11
2.1 持续赋值语句	11
2.2 过程赋值语句	12
2.2.1 阻塞式过程赋值	12
2.2.2 非阻塞式过程赋值	13
2.2.3 赋值对象	13
2.2.4 赋值限制	14
2.3 逻辑算符	15
2.4 算术算符	15
2.4.1 无符号算术	16
2.4.2 有符号算术	17
2.4.3 进位的建模	18
2.5 关系算符	18
2.6 相等性算符	20
2.7 移位算符	20
2.8 向量运算	22

2.9	部分选取	23
2.10	位选取	24
2.10.1	常量下标	24
2.10.2	表达式中的非常量下标	25
2.10.3	赋值对象中的非常量下标	26
2.11	条件表达式	27
2.12	always 语句	27
2.13	if 语句	30
2.13.1	从 if 语句推导出锁存器	31
2.14	case 语句	34
2.14.1	casez 语句	37
2.14.2	casex 语句	39
2.14.3	从 case 语句推导出锁存器	40
2.14.4	case 分支的全列举	41
2.14.5	并行 case 分支	44
2.14.6	非常量分支项	46
2.15	再谈锁存器推导	47
2.15.1	带异步预置位和清零的锁存器	52
2.16	循环语句	53
2.17	触发器的建模	55
2.17.1	多个时钟	61
2.17.2	多相位时钟	62
2.17.3	使用异步预置位与清零	63
2.17.4	使用同步预置位和清零	67
2.18	再谈阻塞式和非阻塞式赋值	69
2.19	函数	72
2.20	任务	73
2.21	使用 x 值和 z 值	76
2.21.1	x 值	77
2.21.2	z 值	77
2.22	门级建模	80
2.23	模块实例化语句	81
2.23.1	使用预定义功能块	82
2.24	参数化的设计	85

第 3 章 建模示例	88
3.1 组合逻辑的建模	88
3.2 时序逻辑的建模	90
3.3 存储器的建模	91
3.4 编写布尔等式	93
3.5 有限状态机的建模	94
3.5.1 Moore 有限状态机	94
3.5.2 Mealy 有限状态机	97
3.5.3 状态编码	101
3.6 通用移位寄存器的建模	102
3.7 ALU 的建模	105
3.7.1 参数化的 ALU	105
3.7.2 简单 ALU	108
3.8 计数器的建模	108
3.8.1 二进制计数器	108
3.8.2 模 N 计数器	109
3.8.3 约翰逊计数器	110
3.8.4 格雷码计数器	111
3.9 参数化加法器的建模	114
3.10 参数化的比较器的建模	114
3.11 译码器的建模	116
3.11.1 简单译码器	116
3.11.2 二进制译码器	117
3.11.3 约翰逊译码器	117
3.12 多路选择器的建模	119
3.12.1 简单多路选择器	119
3.12.2 参数化的多路选择器	119
3.13 参数化的奇偶校验生成器的建模	121
3.14 三态门的建模	122
3.15 数据流检测模型	123
3.16 阶乘模型	125
3.17 UART 模型	126
3.18 纸牌 21 点模型	132
第 4 章 模型的优化	135

4.1	资源分配	135
4.2	公共子表达式	138
4.3	代码移位	138
4.4	公因子提取	139
4.5	交换律和结合律	140
4.6	其他优化手段	141
4.7	触发器和锁存器的优化	141
4.7.1	消除触发器	141
4.7.2	消除锁存器	142
4.8	设计规模	143
4.9	使用括号	144
第 5 章	验证	146
5.1	测试平台	146
5.2	赋值语句中的延迟	148
5.3	悬空的端口	149
5.4	遗失的锁存器	150
5.5	再谈延迟	152
5.6	事件表	153
5.7	综合指令	154
5.8	变量的异步预置位	155
5.9	阻塞式和非阻塞式赋值	156
5.9.1	组合逻辑	157
5.9.2	时序逻辑	158
附录 A	可综合的语言结构	161
附录 B	通用库	164
参考文献		172

第1章 基础知识

Verilog HDL 是一种硬件描述语言,它不仅可以在门级和寄存器传输级(Register-Transfer Level,RTL)描述硬件,也可以在算法级对硬件加以描述。因此,将采用 Verilog HDL 语言描述的设计转变成逻辑门构成的电路绝非简单的处理过程。

本章将介绍 Verilog HDL 模型映射成逻辑门所涉及的基础知识。

1.1 什么是综合?

综合就是从采用 Verilog HDL 语言描述的寄存器传输级电路模型构造出门级网表的过程。^① 图 1-1 对此综合过程作了说明。综合可能是个中间步骤,它生成的网表是由用导线相互连接的寄存器传输级功能块(如触发器、算术逻辑单元和多路选择器)组成的。此时,就有必要使用被称为 RTL 模块构造器的程序了。该构造器用来针对用户指定的目标工艺从预定义元件库中构造或获取每一个必需的 RTL 功能块。

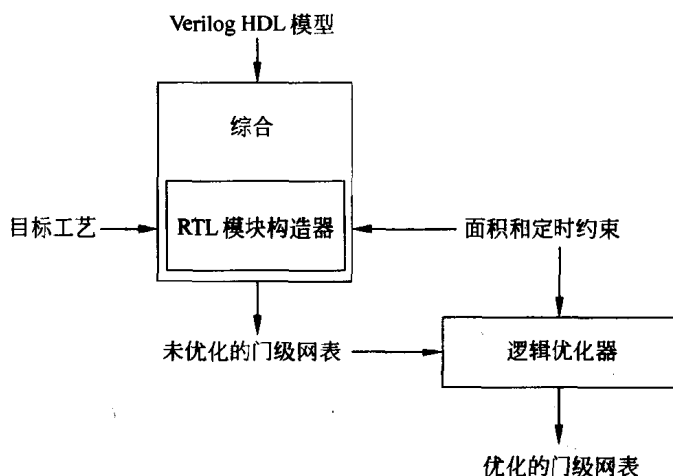


图 1-1 综合过程

^① 这是本书中使用的定义。

产生门级网表之后,逻辑优化器读入网表并以用户指定的面积和定时约束为目标优化网表。这些面积和定时约束也可以用来指导模块构造器恰当地选取或生成寄存器传输级功能块。

本书中,假定目标网表是门级的。附录 B 对综合出的网表中所使用的各种逻辑门作了介绍。但本书并不涉及模块构造和逻辑优化阶段的内容。

图 1-2 列出了 Verilog HDL 的各种基本元素和硬件中所使用的各种基本元件。必须使用某种映射机制或者构造机制将 Verilog HDL 元素转变成相应的硬件元件。由此,产生以下问题:

- a) 数据类型如何转变成硬件?
- b) 常量如何映射成逻辑值?
- c) 语句如何转变成硬件?

后续章节将更详细地讨论这些映射过程。

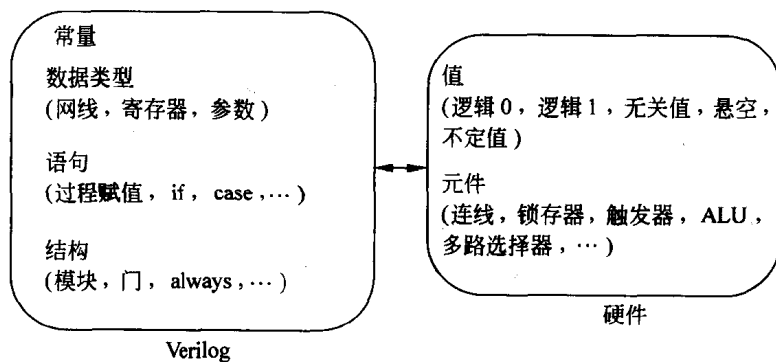


图 1-2 综合涉及的两个领域

1.2 设计流程中的综合

Verilog HDL 这种硬件描述语言允许用户在不同的抽象层次上对电路进行建模,这些层次从门级、寄存器传输级、行为级直至算法级。因此,同一个电路就可以有多种不同的描述方式,但不是每一种描述都是可综合的。事实上,Verilog HDL 原本被设计成一种仿真语言,而不是一种用于综合的语言。结果导致 Verilog HDL 中有很多结构没有相应的硬件可以对应,例如系统调用 `$display`。同样也不存在用于寄存器传输级综合的 Verilog HDL 标准子集。

正是由于存在这些问题,不同的综合系统所支持的 Verilog HDL 综合子集是不同的。由于 Verilog HDL 中不存在单个的对象来表示锁存器或触发器,所以每一种综合系

统都会提供不同的机制以实现锁存器或触发器的建模。因此各种综合系统都定义了自己的 Verilog HDL 可综合子集以及自己的建模方式。

图 1-3 中使用 Verilog HDL 以不同的方式描述了同一个电路。某综合系统支持对方式 A 和方式 B 的综合,但可能不支持对方式 C 的综合。这意味着综合模型在不同的综合系统之间通常是不可移植的。而方式 D 可能根本就不可综合。

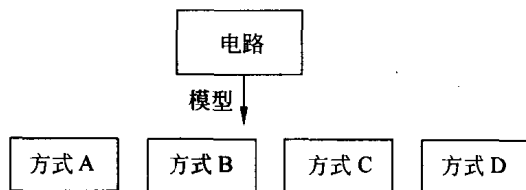


图 1-3 同样的行为,不同的建模方式

这一局限给设计者造成了严重障碍,因为设计者不仅需要理解 Verilog HDL,而且还必须理解特定综合系统的建模方式,才能编写出可综合的模型。图 1-4 是典型的设计流程,但 Verilog HDL 综合也并非总是遵循该流程。

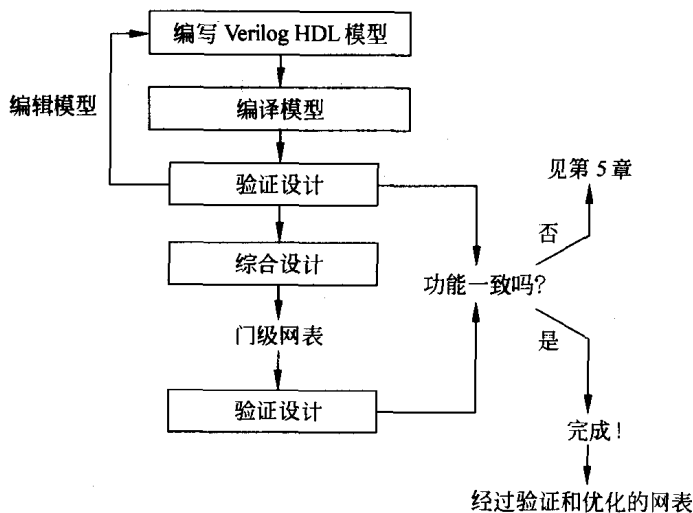


图 1-4 典型的设计流程

此设计流程的问题在于:如果编写 Verilog HDL 模型的时候并不了解综合系统的建模方式(这里假定是为了综合才编写那些模型,否则编写出不可综合的模型也无妨),而只有进入综合阶段设计者才能了解特定综合系统建模方式的限制。这时可能需要改写模型,而且大量的时间可能已经浪费在“编写 Verilog HDL 模型”→“编译模型”→“验证设计”→“编辑模型”的循环之中。因此 Verilog HDL 综合应采用如图 1-5 所示的更实用的