

EDA先锋工作室 编著

# 轻松成为 设计高手

——Verilog HDL实用精解



北京航空航天大学出版社  
BEIHANG UNIVERSITY PRESS

# 轻松成为设计高手

## ——Verilog HDL 实用精解

EDA 先锋工作室 编著

北京航空航天大学出版社

## 内 容 简 介

本书主要分为3部分:第1~3章描述了Verilog HDL语言的基本概念、设计流程、语法与建模方式;第4~6章讨论如何合理使用Verilog HDL描述高性能的可综合电路;第7、8章重点描述了如何编写测试激励以及Verilog的仿真原理。另外,第9章对Verilog HDL语言的发展趋势做了展望。

本书可作为高等院校通信工程、电子工程、计算机、微电子与半导体学等理工专业的教材,也可作为FPGA/CPLD逻辑设计工程师、IC工程师、硬件电路设计工程师的实用工具书。

### 图书在版编目(CIP)数据

轻松成为设计高手: Verilog HDL实用精解 /  
EDA先锋工作室编著. —北京:北京航空航天大学出版社,2012.6

ISBN 978-7-5124-0790-9

I. ①轻… II. ①E… III. ①硬件描述语言,  
Verilog HDL—程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2012)第075863号

版权所有,侵权必究。

### 轻松成为设计高手

#### ——Verilog HDL实用精解

EDA先锋工作室 编著

责任编辑 何 献 叶建增 王国兴

\*

北京航空航天大学出版社出版发行

北京市海淀区学院路37号(邮编100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱:emsbook@gmail.com 邮购电话:(010)82316936

涿州市新华印刷有限公司印装 各地书店经销

\*

开本:710×1 000 1/16 印张:16 字数:350千字

2012年6月第1版 2012年6月第1次印刷 印数:4 000册

ISBN 978-7-5124-0790-9 定价:36.00元

# 关于本书

## 本书内容和特点

Verilog HDL (Hardware Description Language, 硬件描述语言)是目前全球最流行的硬件描述语言之一,广泛应用于 FPGA(Field Programmable Gate Array, 现场可编程逻辑阵列)、CPLD (Complex Programmable Logic Device, 复杂可编程逻辑器件)以及 ASIC(Application Specific Integrated Circuit, 专用集成电路)的设计与验证中。据统计,美国大约有 90%以上,而中国大约有 70%以上的逻辑电路设计人员选择使用 Verilog HDL。

目前国内关于 Verilog HDL 的书籍大多数仅仅介绍 Verilog HDL 的语法和简单建模,并没有体现实用性和必要的理论指导。HDL 语言与软件设计语言(如 C 语言等)的差距很大,核心在于 HDL 语言的实质是对硬件电路的描述与建模,一切设计与验证需要从硬件电路的角度思考!因此需要真正掌握硬件电路的设计和验证的核心,简单的语法书是远远不够的,广大读者真正需要的是一本理论结合实际、设计与验证并重,思想性强、指导性强、实用性高的书籍。因此,EDA 先锋工作室创作了本书。

本书作者有 15 年以上的专业设计经验,曾担任著名 FPGA/CPLD 厂商的技术经理,所以本身具有了其他语言书籍无法比拟的独特优势——丰富的实际经验。这点正是本书与其他语法书的核心区别。

本书主要分为 3 部分:第 1~3 章描述了 Verilog HDL 语言的基本概念、设计流程、语法与建模方式;第 4~6 章讨论如何合理使用 Verilog HDL 描述高性能的可综合电路;第 7、8 章重点描述了如何编写测试激励以及 Verilog 的仿真原理。另外,在第 9 章对 HDL 语言的发展趋势做了展望。

本书的主要内容如下:

- 第 1 章首先介绍 HDL 的设计方法,以及 Verilog、VHDL 和 C 等语言的区别,最后描述了 HDL 语言的一般设计和验证流程。
- 第 2 章介绍了 Verilog 的语法基础。
- 第 3 章重点介绍 Verilog 的 3 种基本描述方法和不同的设计层次。
- 第 4 章介绍了 RTL 建模的概念和一些常用电路的 Verilog 设计方法,最后引申出 Verilog 语言的可综合子集。
- 第 5 章总结了常用的 RTL 同步设计原则,逐一阐述了从设计模块划分到具

体组合逻辑、时序逻辑设计的注意事项,以及 RTL 代码优化的方法。

- 第 6 章介绍了状态机的设计方法和技巧。
- 第 7 章阐述了如何搭建测试平台对设计进行验证,介绍了普通的测试平台和结构化测试平台实例。
- 第 8 章详细描述了 Verilog 语言的语义和仿真原理,是 Verilog 语言的精髓所在。
- 第 9 章总结并展望了 HDL 和 HVL 的发展趋势。

本书主要特点如下:

- 创作思想独到,设身处地为读者考虑:由于作者长期在 IC 设计领域工作,对广大读者的真正需求非常了解,而且自身也在该行业内摸爬滚打了多年,深刻体会到初学者的不易,于是希望创作一本能够真正帮助到他们的书籍,使他们在工作和学习中少走弯路。
- 讲述了设计和验证这两大主题,是一本比较完备的书籍:把 Verilog 语言在这两个方面的应用描述得非常到位,使读者真正洞悉 Verilog 语言的精髓和两用性。在 IC 设计领域,验证往往比设计花费更多的时间,工作量更大,而大部分书籍都对此介绍得比较少,这是一对明显的供需矛盾。
- 突出实用性:本书的目标是在读者读完后,就可以独自去设计和验证电路。理论是为实用服务的,本书在深刻的理论基础上,重点突出实用性,为每种设计方法给出一套读者易于遵循的设计法则或模板。只要读者遵循这个模板,就可以设计出正确高效的电路,同时高效地验证电路的功能。

### 读者对象

本书可作为高等院校通信工程、电子工程、计算机、微电子与半导体学等理工专业的教材,也可作为 FPGA/CPLD 逻辑设计工程师、IC 工程师、硬件电路设计工程师的实用工具书。


### 本书配套资料


本书提供了所有示例的工程文件、设计源文件和说明文件,读者可以从北京航空航天大学出版社([www.buaapress.com.cn](http://www.buaapress.com.cn))网站的“下载专区”免费下载。示例按照章节编号和出现的先后顺序排列,例如“Example-2-1”表示基础篇第 2 章的第 1 个示例。


工程示例文件夹中包括了该工程的项目文件、源文件、报告文件和生成结果等文件。对于一些相对复杂的示例,说明文件中包含了示例的详细信息和操作指南。而对于一些简单的实例,只是给出源代码,读者可以在此基础上引申开去,做出更大的工程。


## 本书约定

为了方便读者阅读,书中设计了4个小图标,它们代表的含义如下:

 行家指点:用于介绍使用经验和心得,或罗列重要的概念。

 注意事项:用于提醒读者应该注意的问题。

 多学一招:用于介绍实现同一功能的不同方法。

 操作实例:用于引出一个操作题目和相应的一组操作步骤。

本书第1、4、5、6章由王诚编写,第2、3、7、8、9章由吴继华编写。全书由吴继华、王诚统筹定稿。

有兴趣的读者可以发送电子邮件到:westorwang@gmail.com,与作者进一步交流;也可发送电子邮件到:xdhydc5@sina.com,与本书策划编辑沟通。

EDA 先锋工作室

2012年3月

# 本书导读

为了方便读者更高效地使用本书,下面的表格明示了每一章节的学习重点、难点和适合的读者层次。

章号	章节标题	重点	难点	读者准备
1	HDL 设计方法简介	① Verilog HDL 和 C 语言的区别 ② HDL 设计与验证流程	HDL 是分层次的,不同层次对应不同的设计和验证流程	本章适合初学者
2	Verilog 语言基础	基本语法释义	驱动和赋值	读者需要掌握基本的 Verilog 语法
3	描述方式和设计层次	① 3 种基本的描述方式 ② 设计层次	阻塞赋值、非阻塞赋值和过程连续赋值	这章的很多概念非常重要,要求读者能掌握常用的描述方式和设计层次
4	RTL 概念与常用 RTL 建模	① RTL 级的概念、要素和设计步骤 ② RTL 级常用的建模方法	非阻塞赋值、阻塞赋值、连续赋值;寄存器电路建模;组合逻辑建模;双向端口与三态信号建模;mux 建模;存储器建模;同步复位和异步复位;用 case 和 if... else 建模讨论	这章是笔者的经验总结,希望读者能仔细体会、学习 RTL 的常用建模方法
5	RTL 设计与编码指导	① “面积与速度互换”、“硬件原则”、“系统原则”等设计指导原则 ② 同步设计原则和多时钟处理 ③ 一些具体的代码风格	“面积与速度互换”、“硬件原则”、“系统原则”和“同步设计原则”在笔者的其他书籍、期刊和网络作品中被称为“4 大指导原则”	本章是笔者多年的设计经验总结,希望读者能仔细体会其中的 4 大指导原则
6	如何写好状态机	① 状态机的设计思想、方法与要素 ② 推荐的状态机设计方法	两段式和三段式状态机设计方法	本章详细讨论了状态机的设计方法与技巧,要求读者掌握两段式和三段式状态机设计方法

续表

章号	章节标题	重点	难点	读者准备
7	逻辑验证与 testbench 编写	① 编写仿真测试激励 ② 结构化 testbench 设计	结构化 testbench 的编写	本章需要读者有一定的逻辑设计经验。认真学习本章,培养好的测试激励设计习惯
8	Verilog 语义和仿真原理	① Verilog 的仿真语义与仿真原理 ② 事件队列与仿真参考模型 ③ 时序模型与延时	仿真的原理和仿真参考模型;阻塞与非阻塞赋值的本质	本章为 Verilog 高级用户详细剖析了仿真的语义与原理,并再次深入探讨了阻塞和非阻塞赋值
9	设计与验证语言发展趋势	① 硬件设计语言发展的现状和走向 ② 验证语言发展的现状和走向	硬件设计语言发展的现状和走向	本章作为补充知识,可做一般性了解





<b>第 1 章 HDL 设计方法简介</b> .....	1
1.1 设计方法的变迁 .....	1
1.2 Verilog 语言的特点 .....	2
1.2.1 Verilog 的由来 .....	2
1.2.2 HDL 与原理图 .....	2
1.2.3 Verilog 和 VHDL .....	3
1.2.4 Verilog 和 C .....	4
1.3 HDL 设计与验证流程 .....	5
1.4 问题与思考 .....	8
<b>第 2 章 Verilog 语言基础</b> .....	9
2.1 Top-Down 和 Bottom-Up .....	9
2.2 从一个实例开始 .....	10
2.2.1 实例 .....	10
2.2.2 3 种描述方法 .....	14
2.3 基本词法 .....	14
2.4 模块和端口 .....	15
2.5 编译指令 .....	17
2.6 逻辑值与常量 .....	18
2.6.1 逻辑值 .....	18
2.6.2 常量 .....	19
2.7 变量类型 .....	20
2.7.1 线网类型 .....	20
2.7.2 寄存器类型 .....	20
2.7.3 变量的物理含义 .....	21
2.7.4 驱动和赋值 .....	22
2.8 参数 .....	23
2.9 Verilog 中的并发与顺序 .....	24
2.10 操作数、操作符和表达式 .....	24
2.10.1 操作符 .....	24
2.10.2 二进制数值 .....	28

2.10.3	操作数 .....	28
2.11	系统任务和系统函数 .....	29
2.11.1	显示任务 .....	29
2.11.2	文件输入/输出任务 .....	30
2.11.3	其他系统任务和函数 .....	30
2.12	小结 .....	31
2.13	问题与思考 .....	31
<b>第3章</b>	<b>描述方式和设计层次 .....</b>	<b>32</b>
3.1	描述方式 .....	32
3.2	数据流描述 .....	32
3.2.1	数据流 .....	32
3.2.2	连续赋值语句 .....	33
3.2.3	延 时 .....	35
3.2.4	多驱动源线网 .....	36
3.3	行为描述 .....	38
3.3.1	行为描述的语句格式 .....	38
3.3.2	过程赋值语句 .....	42
3.3.3	语句组 .....	45
3.3.4	高级编程语句 .....	47
3.4	结构化描述 .....	52
3.4.1	实例化模块的方法 .....	54
3.4.2	参数化模块 .....	56
3.5	设计层次 .....	59
3.5.1	系统级和行为级 .....	59
3.5.2	RTL 级 .....	62
3.5.3	门 级 .....	62
3.5.4	晶体管级 .....	63
3.5.5	混合描述 .....	63
3.6	CRC 计算与校验电路实例 .....	63
3.6.1	CRC10 校验,行为级 .....	64
3.6.2	CRC10 计算电路,RTL 级 .....	64
3.7	小结 .....	67
3.8	问题与思考 .....	67
<b>第4章</b>	<b>RTL 概念与常用 RTL 建模 .....</b>	<b>68</b>
4.1	RTL 和综合的概念 .....	68
4.2	RTL 级的基本要素和设计步骤 .....	69
4.3	常用 RTL 级建模 .....	70

4.3.1	非阻塞赋值、阻塞赋值、连续赋值	70
4.3.2	寄存器电路建模	71
4.3.3	组合逻辑建模	74
4.3.4	双向端口与三态信号建模	76
4.3.5	mux 建模	77
4.3.6	存储器建模	78
4.3.7	简单的时钟分频电路	79
4.3.8	串/并转换建模	81
4.3.9	同步复位和异步复位	81
4.3.10	用 case 和 if... else 建模	85
4.3.11	可综合的 Verilog 语法子集	91
4.4	CPU 读/写 PLD 寄存器接口设计实例	91
4.5	小 结	96
4.6	问题与思考	96
<b>第 5 章</b>	<b>RTL 设计与编码指导</b>	<b>97</b>
5.1	一般性指导原则	97
5.1.1	面积和速度的平衡与互换原则	98
5.1.2	硬件原则	107
5.1.3	系统原则	110
5.2	同步设计原则和多时钟处理	112
5.2.1	同步设计原则	112
5.2.2	亚稳态	114
5.2.3	异步时钟域数据同步	116
5.3	代码风格	118
5.3.1	Coding Style 的分类	118
5.3.2	Coding Style 的重要性	119
5.4	结构层次设计和模块划分	119
5.4.1	结构层次化编码	119
5.4.2	模块划分的技巧	120
5.5	组合逻辑的注意事项	121
5.5.1	always 组合逻辑信号敏感表	122
5.5.2	组合逻辑环路	122
5.5.3	脉冲产生器	123
5.5.4	慎用锁存器	124
5.6	时钟设计的注意事项	125
5.6.1	内部逻辑产生的时钟	126
5.6.2	Ripple Counter	126

5.6.3	时钟选择 .....	126
5.6.4	门控时钟 .....	127
5.6.5	时钟同步使能端 .....	128
5.7	RTL 代码优化技巧 .....	128
5.7.1	使用 Pipelining 技术优化时序 .....	128
5.7.2	模块复用与 Resource Sharing .....	129
5.7.3	逻辑复制 .....	131
5.7.4	香农扩展运算 .....	132
5.8	小结 .....	135
5.9	问题与思考 .....	136
<b>第 6 章</b>	<b>如何写好状态机</b> .....	<b>137</b>
6.1	状态机的基本概念 .....	137
6.1.1	状态机是一种思想方法 .....	137
6.1.2	状态机的基本要素与分类 .....	139
6.1.3	状态机的基本描述方式 .....	139
6.2	如何写好状态机 .....	140
6.2.1	什么是好的 RTL 级 FSM 描述 .....	141
6.2.2	RTL 级状态机描述常用语法 .....	141
6.2.3	推荐的状态机描述方法 .....	144
6.2.4	状态机设计的其他技巧 .....	158
6.3	使用 Synplify Pro 分析 FSM .....	160
6.4	小结 .....	163
6.5	问题与思考 .....	164
<b>第 7 章</b>	<b>逻辑验证与 testbench 编写</b> .....	<b>165</b>
7.1	概述 .....	165
7.1.1	仿真和验证 .....	165
7.1.2	什么是 testbench .....	166
7.2	建立 testbench, 仿真设计 .....	168
7.2.1	编写仿真激励 .....	168
7.2.2	搭建仿真环境 .....	179
7.2.3	确认仿真结果 .....	180
7.2.4	写 testbench 要注意什么 .....	182
7.3	CPU 接口仿真实例 .....	184
7.3.1	设计简介 .....	184
7.3.2	一种 testbench .....	186
7.3.3	另一种 testbench .....	189
7.4	结构化 testbench 思想 .....	191

7.4.1	任务和函数 .....	192
7.4.2	总线模型重用 .....	192
7.4.3	测试套具 .....	192
7.4.4	测试用例 .....	193
7.4.5	结构化 testbench .....	194
7.5	实例：结构化 testbench 的编写 .....	196
7.5.1	单顶层 testbench .....	196
7.5.2	多顶层 testbench .....	199
7.6	扩展 Verilog 的高层建模能力 .....	201
7.7	小 结 .....	201
7.8	问题与思考 .....	201
<b>第 8 章</b>	<b>Verilog 语义和仿真原理 .....</b>	<b>202</b>
8.1	从一个问题说起 .....	202
8.2	电路与仿真 .....	203
8.2.1	电路是并行的 .....	204
8.2.2	Verilog 是并行语言 .....	204
8.2.3	仿真器串行执行, Verilog 仿真语义 .....	205
8.3	仿真原理 .....	206
8.3.1	Verilog 如何仿真 .....	206
8.3.2	仿真时间 .....	210
8.3.3	事件驱动 .....	211
8.3.4	进 程 .....	211
8.3.5	调 度 .....	212
8.3.6	时序控制 .....	213
8.3.7	进程、事件和仿真时间的关系 .....	213
8.3.8	Verilog 语言的不确定性 .....	213
8.4	分层事件队列与仿真参考模型 .....	214
8.4.1	分层事件队列 .....	214
8.4.2	仿真参考模型 .....	215
8.5	时序模型与延时 .....	215
8.5.1	仿真模型 .....	216
8.5.2	时序模型 .....	216
8.5.3	案例分析 .....	217
8.5.4	如何在 Verilog 语言中正确的增加延时 .....	218
8.6	再谈阻塞与非阻塞赋值 .....	222
8.6.1	本 质 .....	222
8.6.2	案例分析 .....	225

8.7	如何提高代码的仿真效率 .....	228
8.8	防止仿真和综合结果不一样 .....	229
8.9	小 结 .....	230
8.10	问题与思考 .....	230
<b>第 9 章</b>	<b>设计与验证语言发展趋势 .....</b>	<b>231</b>
9.1	设计与验证语言发展历程 .....	232
9.1.1	HDL 作为设计和验证语言 .....	232
9.1.2	C/C++ 和私有的验证语言 .....	232
9.1.3	Accellera 和 IEEE 的标准化工作 .....	232
9.2	硬件设计语言发展的现状和走向 .....	233
9.2.1	HDL 的竞争 .....	233
9.2.2	一些尝试 .....	234
9.2.3	下一代的 Verilog 语言 .....	234
9.2.4	SystemC .....	235
9.3	验证语言发展现状和走向 .....	236
9.3.1	验证方法 .....	236
9.3.2	HVL 标准化进程 .....	236
9.3.3	HVL 的新需求 .....	237
9.4	总结和展望 .....	237
9.5	小 结 .....	237
9.6	问题与思考 .....	238
<b>参考文献</b>	<b>.....</b>	<b>239</b>

# 第 1 章

## HDL 设计方法简介

本章重点介绍了数字系统的建模和 HDL 语言的基本概念,并引入了主流的设计和验证流程。主要内容如下:

- 设计方法的变迁;
- Verilog 语言的特点;
- HDL 设计与验证流程。

### 1.1 设计方法的变迁

有人将当今时代称为数字化时代,因为数字技术突飞猛进,日新月异。随着微电子设计技术与工艺的发展,数字集成电路从电子管、晶体管、中小规模集成电路、超大规模集成电路(VLSIC)逐步发展到今天的专用集成电路(ASIC),人们发现自己的身边充斥着数字化的产品,比如计算机、手机、数字电视等都运用了复杂的专用数字集成电路。数字逻辑器件从简单的逻辑门到复杂的 SoC(System on Chip,片上系统),种类繁多,提供了对复杂系统的灵活的支撑。

随着数字电路系统的不断发展,系统的逻辑复杂度与规模日益增加,数字系统的设计方法也随之不断演进。早期简单的门逻辑设计阶段,电子辅助设计(EDA)工具应用范围十分有限,工程师们习惯于使用卡诺图简化设计,然后通过面包板等实验系统验证设计;系统相对复杂后,工程师们借助 EDA 工具通过原理图描述数字系统,原理图由元件库中的元件构成,使用 EDA 工具可以对原理图进行仿真并分析其性能;当数字系统发展到 ASIC 与可编程逻辑器件(PLD)设计阶段后,原理图不利于移植,维护费时费力等缺点逐步显现,这时需要抽象度更高、更灵活的设计方式——硬件描述语言(HDL, Hardware Description Language)。

使用 HDL 语言,可以从算法、系统级(System Level)、功能模块级(Functional Model Level)、行为级(Behavioral Level)、寄存器传输级(RTL, Register Transfer Level)、门级(Gate Level)到开关级(Switch Level)等不同层次描述数字电路系统,然

后通过 EDA 工具综合、仿真并实现该系统。可以说 HDL 语言的出现是数字系统设计方法的一个重大飞跃。

数字电路系统设计方法的另一个重大飞跃是随着 EDA 工具的不断推陈出新发生的。早期 ASIC 和 PLD 设计时,人们需要直接描述 CMOS 的开关电路或门级电路,其设计量庞大,仿真速度也非常慢,像当今系统门数量为千万门级的 FPGA (Field Programmable Gate Array, 现场可编程门阵列)用开关级或门级方法描述在当时是不可想象的。这时人们就希望能够使用 HDL 直接从更高的层次描述电路,然后使用 EDA 工具自动将高层次的 HDL 电路描述解析到门级,这些将大大节约设计与仿真时间,这个从高层次的电路描述通过 EDA 工具解析到门级等低层次的电路描述的过程就叫“综合”(Synthesize),或者逻辑综合。综合工具能将高层次的 HDL 语言、原理图等设计描述翻译成由与、或、非门等基本逻辑单元组成的门级连接(网表),并根据设计目标与要求(约束条件)优化所生成的逻辑连接,输出门级网表文件。目前最成熟的综合工具是 RTL 级综合工具,它能将 RTL 级描述翻译并优化为门级网表。综合工具的产生支撑了直接从 RTL 级描述硬件电路,这种从门级描述抽象到 RTL 级描述数字电路是设计方法的又一次伟大飞跃。

从上面描述可以看到 HDL 语言是分层次的,关于上述 HDL 语言设计层次的含义在 1.3 节和 3.5 节有详细论述。

## 1.2 Verilog 语言的特点

本节综述 Verilog HDL 语言的特点以及与其他语言的比较。

### 1.2.1 Verilog 的由来

Verilog 是 Verilog HDL 的简称。Verilog 语言最初于 1983 年由 Gateway Design Automation 公司开发,于 1995 年接受为 IEEE 标准。Verilog 语言不仅定义了语法,而且对每个语法结构都清晰定义了仿真语义,从而便于仿真调试。而且 Verilog 从 C 语言继承了很多操作符和语法结构,对初学者而言易学易用。另外 Verilog 语言具有很强的扩展能力,最新的 Verilog 2001 标准大大扩展了 Verilog 的应用灵活性。

另外一个流行的 HDL 语言是 VHDL (Very High Speed Integrated Circuit HDL, 超高速集成电路硬件描述语言的缩写),其发展初期受到美国国防部支持,并于 1987 年成为 IEEE 标准。VHDL 语言的特点是描述严谨。

为了使读者加深对 Verilog 语言的理解,下面比较几个设计输入的方法。

### 1.2.2 HDL 与原理图

硬件描述语言(HDL)和原理图是最常用的两种数字硬件电路描述方法。原理图设计输入法在早期应用得比较广泛,它根据设计要求选用器件、绘制原理图、完成



输入过程。这种方法的优点是直观、便于理解、元件库资源丰富。但是在大型设计中,这种方法的维护性较差,不利于模块建设与重用。更主要的缺点是:当所选用芯片升级换代后,所有的原理图都要做相应的改动。

目前最常用的设计方法是 HDL 设计输入法。其中影响最为广泛的 HDL 语言是 Verilog HDL 和 VHDL。它们的共同特点是利于自顶向下的设计,利于模块的划分与复用,可移植性好,通用性好,设计不因芯片的工艺与结构的变化而变化,更利于向 ASIC 的移植。

波形输入和状态机输入方法是两种常用的辅助设计输入方法。使用波形输入法时,只要绘制出激励波形和输出波形,EDA 软件就能自动地根据响应关系进行设计或仿真。而使用状态机输入法时,设计者只须画出状态转移图,EDA 软件就能生成相应的 HDL 代码或者原理图。这两种设计方法往往和某种特定的设计工具相关,应用范围受到局限,而且效率和可维护性不高,仅仅在某些场合作为辅助的设计描述手段。

因此,笔者推荐初学者在描述和仿真数字电路时首选 HDL 语言方式,仅仅在某些要求使用图形描述设计顶层的情况下才使用原理图,不要在设计顶层以外的其他层次使用原理图。另外,不要依赖于波形设计工具,尽量使用 HDL 描述仿真与激励,因为简单的信号虽然用波形描述起来十分方便,但是复杂的测试激励几乎无法用波形工具有效地描述。

### 1.2.3 Verilog 和 VHDL

Verilog 和 VHDL 作为最为流行的 HDL 语言,从设计能力上而言都能胜任数字电路系统的设计任务。

VHDL 最初用作文档来描述数字硬件的行为,因此其描述性和抽象层次更高。也就是说,VHDL 更适合描述更高层次(如行为级、系统级等)的硬件电路。

Verilog 最初是为了更简洁有效地描述数字硬件电路和仿真而设计的,它的许多关键字与语法集成了 C 语言的体系,因此易学易懂,接受度很广。

前面已经提到最流行的 HDL 语言是 Verilog 和 VHDL。在其基础上发展出了许多抽象程度更高的硬件描述语言,如 SystemVerilog、Superlog、SystemC、CoWare C,这些高级 HDL 语言的语法结构更加丰富,更适合做系统级、功能级等高层次的设计描述和仿真。HDL 语言的适用层次示意图如图 1-1 所示。其中实线框表示适用程度较高,虚线框表示适用程度较低。

笔者推荐初学者学习 Verilog 入门。Verilog 语法的寄存器和线网两种数据类型定义清楚,时序与组合电路描述简洁,能够帮助初学者快速建立硬件设计的基本概念,非常容易上手。这也是 Verilog 成为最受欢迎的 HDL 语言的主要原因。

但是请读者明确,Verilog 与 VHDL 语言本身并没有什么优劣之分,而是各有所长。使用 HDL 语言描述数字硬件电路,HDL 仅仅是将真实的硬件设计抽象为语言形式的工具,是真实的硬件电路到 EDA 工具的桥梁,选择何种语言作为桥梁本身并