

高等学校电子信息类专业  
“十二五”规划教材

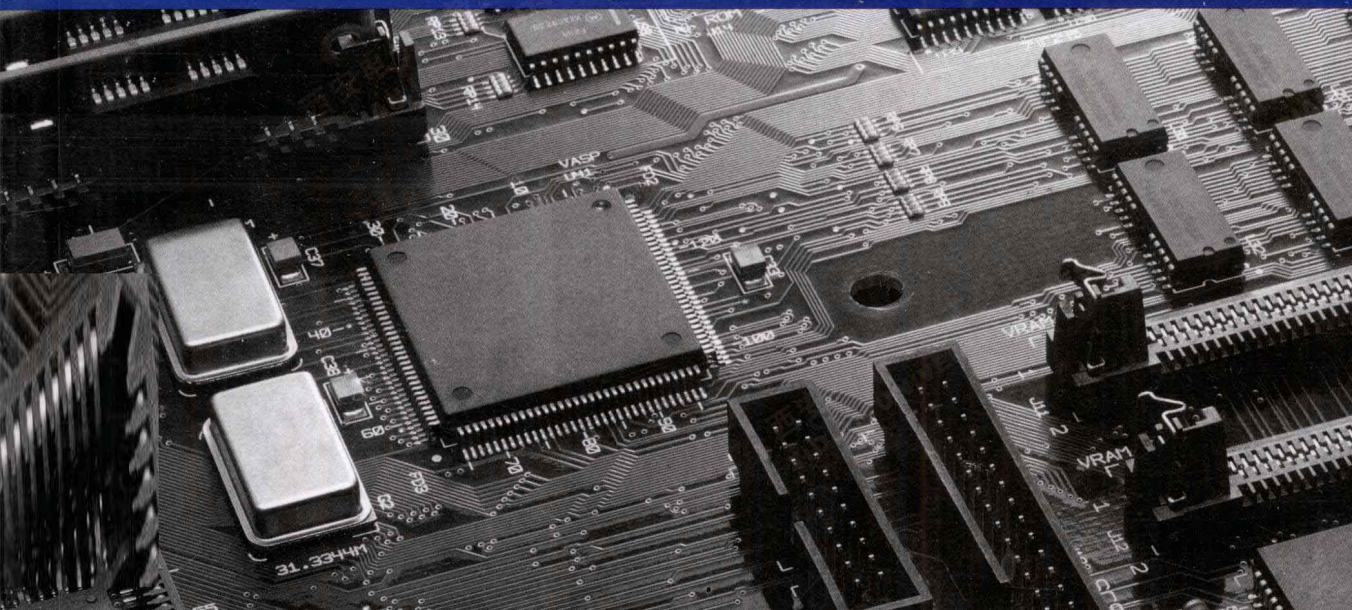
*ELECTRONIC*  
*INFORMATION SPECIALTY*

# Verilog HDL数字集成电路设计原理与应用

蔡觉平 何小川 李道楠 编著



西安电子科技大学出版社  
<http://www.xdph.com>



高等学校电子信息类专业“十二五”规划教材

# Verilog HDL 数字集成电路 设计原理与应用

蔡觉平 何小川 李道楠 编著

西安电子科技大学出版社

## 内 容 简 介

本书系统地对 Verilog HDL 语法和程序设计进行了介绍,明确了数字可综合逻辑设计和测试仿真程序在 Verilog HDL 语言中的不同,通过对典型的组合逻辑电路、时序逻辑电路和测试程序的设计举例,较为完整地说明了 Verilog HDL 语言在数字集成电路中的使用方法。

全书共 8 章,主要内容包括硬件描述语言和 Verilog HDL 概述, Verilog HDL 的基本语法, Verilog HDL 程序设计语句和描述方式, Verilog HDL 对组合逻辑和时序逻辑的设计和举例, Verilog HDL 集成电路测试程序和测试方法,较为复杂的数字电路和系统的设计举例,数字集成电路中 Verilog HDL 的 EDA 工具和使用,以及对 Verilog HDL 发展的分析等。

本书可作为电子信息类相关专业本科生和研究生的教材,也可作为数字集成电路设计工程师的参考书。

### 图书在版编目(CIP)数据

Verilog HDL 数字集成电路设计原理与应用/蔡觉平,何小川,李道楠编著.

—西安:西安电子科技大学出版社,2011.9

高等学校电子信息类专业“十二五”规划教材

ISBN 978 - 7 - 5606 - 2652 - 9

I. ① V… II. ① 蔡… ② 何… ③ 李… III. ① 数字集成电路—电路设计—高等学校—教材

② VHDL 语言—程序设计—高等学校—教材

IV. ① TN431.2 ② TP312

中国版本图书馆 CIP 数据核字(2011)第 155039 号

策划编辑 云立实

责任编辑 买永莲 云立实

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2011 年 9 月第 1 版 2011 年 9 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 18

字 数 421 千字

印 数 1~3000 册

定 价 30.00 元

ISBN 978 - 7 - 5606 - 2652 - 9/TN · 0620

**XDUP 2944001-1**

\*\*\* 如有印装问题可调换 \*\*\*

本社图书封面为激光防伪覆膜,谨防盗版。

# 前 言

随着集成电路技术的飞速发展，集成电路的制造工艺已经达到 32 nm 甚至更小尺寸，数字集成电路的规模越来越大，复杂度越来越高。为了提高设计的效率和可靠性，融合了电子技术、计算机技术和智能化技术的 EDA(Electronics Design Automation)工具在高速复杂的数字集成电路中得到了广泛应用。

硬件描述语言(HDL)是现代专用集成电路(ASIC)EDA 设计的重要设计和仿真语言。目前，大部分数字集成电路设计者都在使用 HDL 的设计方法创建高层次、结构化、基于语言的抽象电路描述，利用已有的设计技术综合出所需的硬件电路，并对其功能验证和时序分析。

对于准备从事集成电路和 FPGA 设计的读者来说，需要了解如何在设计流程的关键阶段正确使用 HDL，并在综合后获得期望的电路。因此需要在 HDL 基本语法结构的基础上，深入理解电路的综合特性和测试仿真方法。本书就是为这样一个目标而撰写的。

Verilog HDL 是被广泛采用的一种硬件描述语言，目前有关 Verilog HDL 的书籍多数重点讲解语言和语法，较少分析 Verilog HDL 语言和相应数字电路之间的关系以及如何通过设计得到与目标相符合的电路系统。本书的着眼点主要在于 Verilog HDL 的设计方法上，这是编写的基本出发点。

本书主要根据 Verilog HDL 国际标准 IEEE 1364-2001，对基本语法、设计方法、综合方法和验证方法等进行讲解和分析，并通过大量的实用电路的设计，使读者能够对基于 Verilog HDL 的数字集成电路设计技术有一个全面的了解。

本书重点集中于如何在数字电路的设计、综合和验证过程中合理使用 Verilog HDL。由于 Verilog HDL 本质上是对数字电路的一种抽象描述方法，因此学习本书时需具备较深入的数字电路设计基础，同时至少须熟悉一种编程语言。在本书中，我们仅讨论了 Verilog HDL 核心设计方法和验证方法，采用了典型的设计例程，帮助读者快速掌握相关知识。由于数字电路多样而灵活，因此希望这些例程能够为复杂电路的设计提供帮助。

在数字电路中通常采用真值表、状态转移图和算法状态图对组合电路和时序电路进行分析和表示，本书即采用了这些方法用于 Verilog HDL 的设计和分析，以提高读者对设计方法的理解。同时，对于目前在信号处理、自动控制、数值计算等应用中所采用的一些设计方法(如查找表(LUT)、级数展开和有限状态机)进行了说明和举例，希望能够帮助读者扩展设计思路。

本书列举大量实例的目的主要是希望读者在使用 Verilog HDL 进行超大规模集成电路(VLSI)设计中，学习如何应用关键步骤进行设计和验证。所列举的实例是完整的，并在 ModelSim 和 Synplify 软件中进行了编译、综合和仿真。

本书适用于学习集成电路设计和硬件描述语言课程的本科生和研究生，可作为课程教材和设计参考，对于希望通过实例学习 Verilog HDL 并将这种语言应用于集成电路设计和测试的专业工程师，也会有一定的帮助。

全书共 8 章。第 1 章主要介绍 Verilog HDL 在数字集成电路设计中的作用、发展演变的过程和相关国际标准；第 2 章讲述 Verilog HDL 基本语言结构和语法规则；第 3 章通过例程对 Verilog HDL 设计描述方式和高级程序语句进行介绍；第 4 章对组合电路和时序电路的 Verilog HDL 程序设计进行深入分析，并给出典型的设计例程；第 5 章介绍 Verilog HDL 仿真测试程序和相关系统任务的知识；第 6 章给出功能较为复杂的数字电路、系统和总线设计；第 7 章简要介绍采用 Verilog HDL 进行集成电路设计的 EDA 工具和例程；第 8 章从集成电路设计面临的问题出发，分析 Verilog HDL 完善和发展的方向。

十分感谢为本书的出版作出贡献的老师和学生们。感谢何小川和李涛同学在集成电路设计流程过程、代码质量评估等方面所做的大量的实际工作；感谢李道楠和张泽同学对于本书中实例的整理和在 ModelSim 和 Synplify 中所做的验证工作；感谢耿剑波同学对于已有著作中语法的重新整理，并提出了在教学中应用的建议。此外，还要感谢西安电子科技大学出版社的云立实编辑，他的支持和鼓励以及对本书创作过程的指导，确保了本书的出版。

希望本书能为致力于集成电路设计的读者和工程师提供帮助。

编者  
2011 年 1 月

# 目 录

<b>第 1 章 Verilog HDL 数字集成电路设计方法概述</b> .....	1
1.1 数字集成电路的发展和设计方法的演变 .....	1
1.2 硬件描述语言 .....	3
1.3 Verilog HDL 的发展和国际标准 .....	3
1.4 Verilog HDL 和 VHDL .....	5
1.5 Verilog HDL 在数字集成电路设计中的优点 .....	7
1.6 功能模块的可重用性 .....	9
1.7 IP 核和知识产权保护 .....	9
1.8 Verilog HDL 在数字集成电路设计流程中的作用 .....	10
本章小结 .....	11
思考题和习题 .....	11
<b>第 2 章 Verilog HDL 基础知识</b> .....	13
2.1 Verilog HDL 的语言要素 .....	13
2.1.1 空白符 .....	13
2.1.2 注释符 .....	13
2.1.3 标识符和转义标识符 .....	14
2.1.4 关键字 .....	14
2.1.5 数值 .....	15
2.2 数据类型 .....	17
2.2.1 物理数据类型 .....	18
2.2.2 连线型和寄存器型数据类型的声明 .....	21
2.2.3 存储器型 .....	22
2.2.4 抽象数据类型 .....	22
2.3 运算符 .....	24
2.3.1 算术运算符 .....	24
2.3.2 关系运算符 .....	25
2.3.3 相等关系运算符 .....	26
2.3.4 逻辑运算符 .....	27
2.3.5 按位运算符 .....	27
2.3.6 归约运算符 .....	28
2.3.7 移位运算符 .....	29

2.3.8 条件运算符 .....	30
2.3.9 连接和复制运算符 .....	30
2.4 模块 .....	31
2.4.1 模块的基本概念 .....	31
2.4.2 端口 .....	33
本章小结 .....	33
思考题和习题 .....	34
<b>第 3 章 Verilog HDL 程序设计语句和描述方式</b> .....	<b>35</b>
3.1 数据流建模 .....	35
3.2 行为级建模 .....	37
3.2.1 过程语句 .....	38
3.2.2 语句块 .....	41
3.2.3 过程赋值语句 .....	44
3.2.4 连续赋值语句 .....	47
3.2.5 条件分支语句 .....	49
3.2.6 循环语句 .....	53
3.3 结构化建模 .....	57
3.3.1 模块级建模 .....	57
3.3.2 门级建模 .....	64
3.3.3 开关级建模 .....	66
本章小结 .....	68
思考题和习题 .....	68
<b>第 4 章 Verilog HDL 数字逻辑电路设计方法</b> .....	<b>71</b>
4.1 Verilog HDL 语言的设计思想和可综合特性 .....	71
4.2 组合电路的设计 .....	74
4.2.1 数字加法器 .....	77
4.2.2 数据比较器 .....	80
4.2.3 数据选择器 .....	81
4.2.4 数字编码器 .....	82
4.2.5 数字译码器 .....	87
4.2.6 奇偶校验器 .....	89
4.3 时序电路的设计 .....	91
4.3.1 触发器 .....	96
4.3.2 计数器 .....	99
4.3.3 移位寄存器 .....	100
4.3.4 序列信号发生器 .....	101
4.4 有限同步状态机 .....	105
本章小结 .....	115

思考题和习题 .....	115
<b>第 5 章 仿真验证与 Testbench 编写 .....</b>	<b>119</b>
5.1 Verilog HDL 电路仿真和验证概述 .....	119
5.2 Verilog HDL 测试程序设计基础 .....	120
5.2.1 Testbench 及其结构 .....	120
5.2.2 测试平台举例 .....	123
5.2.3 Verilog HDL 仿真结果确认 .....	126
5.2.4 Verilog HDL 仿真效率 .....	128
5.3 与仿真相关的系统任务 .....	129
5.3.1 \$display 和 \$write .....	129
5.3.2 \$monitor 和 \$strobe .....	131
5.3.3 \$time 和 \$realtime .....	133
5.3.4 \$finish 和 \$stop .....	134
5.3.5 \$readmemh 和 \$readmemb .....	136
5.3.6 \$random .....	137
5.3.7 值变转储文件系统任务 .....	138
5.4 信号时间赋值语句 .....	142
5.4.1 时间延迟的语法说明 .....	142
5.4.2 时间延迟的描述形式 .....	143
5.4.3 边沿触发事件控制 .....	147
5.4.4 电平敏感事件控制 .....	151
5.5 任务和函数 .....	152
5.5.1 任务 .....	152
5.5.2 函数 .....	156
5.5.3 任务与函数的区别 .....	159
5.6 典型测试向量的设计 .....	160
5.6.1 变量初始化 .....	160
5.6.2 数据信号测试向量的产生 .....	161
5.6.3 时钟信号测试向量的产生 .....	162
5.6.4 总线信号测试向量的产生 .....	164
5.7 用户自定义元件模型 .....	166
5.7.1 UDP 的定义与调用 .....	166
5.7.2 UDP 应用实例 .....	167
5.8 基本门级元件和模块的延迟建模 .....	170
5.8.1 门级延迟建模 .....	170
5.8.2 模块延迟建模 .....	174
5.8.3 与时序检查相关的系统任务 .....	177
5.9 编译预处理语句 .....	178



5.9.1	宏定义 .....	178
5.9.2	文件包含处理 .....	180
5.9.3	仿真时间标度 .....	181
5.9.4	条件编译 .....	182
5.9.5	其它语句 .....	183
5.10	Verilog HDL 测试方法简介 .....	183
	本章小结 .....	184
	思考题和习题 .....	184
<b>第 6 章</b>	<b>Verilog HDL 高级程序设计举例</b> .....	<b>188</b>
6.1	数字电路系统设计的层次化描述方法 .....	188
6.2	典型电路设计 .....	192
6.2.1	加法器树乘法器 .....	192
6.2.2	Wallace 树乘法器 .....	196
6.2.3	复数乘法器 .....	198
6.2.4	FIR 滤波器的设计 .....	199
6.2.5	片内存储器的设计 .....	203
6.2.6	FIFO 设计 .....	208
6.2.7	键盘扫描和编码器 .....	212
6.2.8	log 函数的 Verilog HDL 设计 .....	219
6.2.9	CORDIC 算法的 Verilog HDL 实现 .....	223
6.3	总线控制器设计 .....	231
6.3.1	UART 接口控制器 .....	231
6.3.2	SPI 接口控制器 .....	236
	本章小结 .....	240
	思考题和习题 .....	240
<b>第 7 章</b>	<b>仿真测试工具和综合工具</b> .....	<b>242</b>
7.1	数字集成电路设计流程简介 .....	242
7.1.1	设计规范 .....	242
7.1.2	设计划分 .....	243
7.1.3	设计输入 .....	243
7.1.4	仿真 .....	243
7.1.5	综合 .....	244
7.1.6	适配布线 .....	244
7.1.7	时序分析 .....	244
7.1.8	物理验证 .....	245
7.1.9	设计结束 .....	245
7.2	测试和仿真工具 .....	245
7.2.1	ModelSim 的使用 .....	246

7.2.2 NC-Verilog 的使用.....	254
7.3 综合工具 .....	257
7.3.1 Synplify 的使用.....	257
7.3.2 Design Compiler 的使用 .....	262
7.4 测试和综合举例 .....	265
7.4.1 自动布局布线 .....	265
7.4.2 后仿真 .....	266
7.4.3 在 ModelSim 中加入仿真库.....	268
本章小结 .....	270
思考题和习题 .....	270
<b>第 8 章 设计与验证语言的发展趋势 .....</b>	<b>272</b>
本章小结 .....	276
思考题和习题 .....	276
<b>参考文献.....</b>	<b>277</b>

# 第1章 Verilog HDL 数字集成电路设计方法概述

## 1.1 数字集成电路的发展和设计方法的演变

从20世纪60年代开始,数字集成电路的工艺、制造和设计技术飞速发展,数字集成电路从最早的真空管和电子管电路,发展到以硅基半导体为主的集成电路。集成电路的规模从开始的几十个逻辑门的小规模集成电路(Small Scale Integrated, SSI)发展到单芯片数千万个逻辑门的极大规模集成电路(Ultra Large Scale Integrated, ULSI),单芯片可以集成几十亿只晶体管(见图1.1-1)。集成电路设计单元从起初的分立元件发展到IP复用;系统级别由早期的印刷版系统发展到当下最为流行的片上系统(System on Chip, SoC);采用的65 nm和45 nm工艺技术已成熟,并迅速向更小尺寸的产品方向发展;功能方面也从开始的简单布尔逻辑运算发展到可以每秒处理数十亿次计算的复杂运算,使数字集成电路在计算机、通信、图像等领域得到了广泛应用。

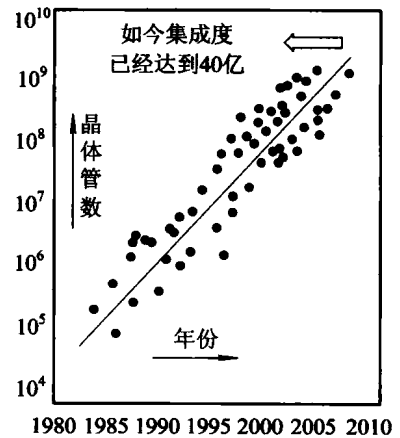


图 1.1-1 数字集成电路复杂度趋势

集成电路工艺制造水平的提高和芯片规模的扩大,使芯片的设计方法和设计技术发生了很大的变化,如图1.1-2所示。早期的数字系统大多采用搭积木式的原理图设计方法,通过一些固定功能的器件加上一定的外围电路构成模块,再由这些模块进一步形成功能电路。这种设计方式的灵活性差,只适合于中小规模的集成电路,当电路和模块的规模增大时,设计效率会降低。

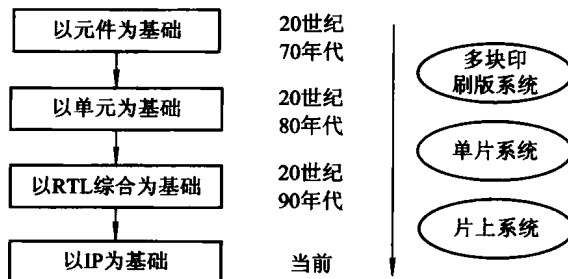


图 1.1-2 数字集成电路设计方法的演变



集成电路的发展可分为三个主要阶段。20 世纪 70 年代(第一次变革时期), 是以加工制造为主导的 IC(Integrated Circuit)产业发展的初级阶段, 主流产品是简单微处理器(Micro Processor Unit, MPU)、存储器以及标准通用逻辑电路。这一时期, IC 整合元件厂(Integrated Device Manufacturer, IDM)在 IC 市场中充当主要角色, 设计只作为附属部门而存在。芯片设计和半导体工艺密切相关, 设计主要以人工为主, 计算机辅助设计(Computer Aided Design, CAD)系统仅作为数据处理和图形编程之用。

20 世纪 80 年代(第二次变革时期)是标准工艺加工线(Foundry)公司与 IC 设计公司共同发展的阶段, 主流产品是 MPU、微控制器(Micro Control Unit, MCU)及专用 IC(Application-Specific IC, ASIC)。这时, Foundry 和 IC 设计公司相结合的方式开始成为集成电路产业发展的新模式。这一时期, IC 产业开始进入以客户为导向的阶段。首先, 标准化功能的 IC 已难以满足整机客户对系统成本、可靠性等的要求; 其次, 由于小尺寸加工技术的进步, 软件的硬件化已成为可能, 超大规模集成电路(Very Large Scale Integrated, VLSI)开始成为主流芯片; 再次, 随着电子设计自动化(Electronic Design Automation, EDA)工具软件的发展, 采用了元件库、工艺模拟参数及其仿真概念等方法, 芯片设计开始进入以计算机为主的抽象化软件阶段, 使设计过程可以独立于生产工艺而存在。无生产线的 IC 设计公司(Fabless)和设计部门纷纷建立起来并得到迅速的发展, 同时以制造为主的 Foundry 工厂也迅速发展起来。1987 年, 全球第一个 Foundry 工厂——台湾积体电路公司成立, 它的创始人张忠谋被誉为“芯片加工之父”。

20 世纪 90 年代(第三次变革时期), IC 产业的“四业”开始分离, 功能强大的通用型中央处理器(Central Processing Unit, CPU)和信号处理器(Digital Signal Processing, DSP)成为产业新的增长点。在这个阶段, 芯片厂商认识到, 越来越庞大的集成电路产业体系并不利于整个 IC 产业的发展, “分”才能精, “整合”才成优势。于是, IC 产业结构向高度专业化转化成为一种趋势, 开始形成了设计业、制造业、封装业、测试业独立成行的局面, 全球 IC 产业的发展越来越显示出这种结构的优势。

进入 21 世纪, IC 产业的发展速度更是惊人, 基于市场和社会发展的需要, 数字集成电路正向多元化发展。在芯片的市场需求方面, 移动通信、多媒体技术等应用的迅速发展, 使具有特定功能的差异化专用芯片取代通用型芯片, 逐渐成为数字 IC 的主要增长点。在技术方面, 出现了新的发展方向。首先, CMOS 模拟技术的发展使得数模混合单芯片集成技术迅速发展, 在设计和成本方面体现了巨大优势; 其次, 应用需求使得存储器在 ULSI 芯片中的作用越来越明显, 高密度存储器及其 SoC 设计成为设计的热点; 再次, 单芯片规模的扩大使得单纯依靠提升频率的发展路线出现技术瓶颈, 大规模多内核处理器结构成为通用型芯片和 SoC 芯片的主流设计方式。在设计方法方面, 采用功能复用 IP(Intelligent Property)的设计方式成为 IC 设计和商业化的一种主要方式, 极大提高了 ULSI 芯片的设计效率和可扩展性。

随着集成电路规模的迅速扩大和复杂度的不断提高, 芯片设计和制造成本不断提高, 设计、测试和制造工艺中的环节增加, 使相应的设计过程变得越来越复杂, 因此, 设计者希望通过某种手段提高数字集成电路的设计、验证的效率和可靠性。

集成电路单元从起初的分立元件到单元, 然后到寄存器传输级, 再到 IP 复用技术; 系统级别由原先的印刷版系统到当下最为流行的 SoC 片上系统。由图 1.1-1 可以看出, 数字集成电路技术的发展速度基本符合摩尔定律, 芯片上晶体管的集成数目以每三年翻两番的速度增长。



超大规模集成电路的发展给设计者和开发者提出了一系列问题,如高层次综合、数模混合电路描述、仿真验证与形式验证等自动验证手段、数字电路的超深亚微米效应,以及设计重用等。这些问题给 EDA 技术的发展提出了一系列新的课题。为了从更高的抽象层次开展设计工作,增强元件模型的可重用性,提高硬件描述设计效率,提出了采用硬件描述语言(Hardware Description Language, HDL)进行数字集成电路设计。如何自动化、高效率地进行数字电路的设计,是 HDL 语言产生的出发点,也是其进一步完善和发展的目标。

## 1.2 硬件描述语言

C、FORTRAN、Pascal 等程序化设计语言,极大地提高了计算机软件程序设计的效率和可靠性。因此,在硬件设计领域,设计人员也希望采用程序化设计语言来进行硬件电路的设计。为此,产生了硬件描述语言 HDL。HDL 是一种高级程序设计语言,通过对数字电路和系统的语言描述,可以对数字集成电路进行设计和验证。利用 HDL 语言,数字集成电路设计工程师可以根据电路结构的特点,采用层次化的设计结构,将抽象的逻辑功能用电路的方式进行实现。为了提高 HDL 对数字电路设计、综合和仿真的能力,Mentor、Cadence、Synopsys 等公司提供了功能强大的电子设计自动化(Electronic Design Automation, EDA)工具,可以将 HDL 程序综合成为网表,通过自动布局布线工具把网表转换为具体电路布线结构,用于专用集成电路(Application Specific Integrated Circuit, ASIC)和现场可编程门阵列(Field Programmable Gate Array, FPGA)的实现。

HDL 语言发展至今,产生了很多种对于数字集成电路的描述性设计语言,并成功地应用于设计的各个阶段(建模、仿真、验证和综合等)。20 世纪 80 年代至今,已出现了上百种硬件描述语言,它们对设计自动化起到了极大的促进和推动作用,主要有 Gateway Design Automation 公司提出的 Verilog HDL、美国国防部高级研究计划局(DARPA)设计的 VHDL、美国国防部 RPASSP(Rapid Prototyping of Application Specification Signal Processing)计划提出的基于面向对象的 OO VHDL(Object Oriented VHDL)、美国杜克大学的 DE VHDL(Duke Extended VHDL)和美国电气和电子工程师协会(Institute of Electrical and Electronics Engineers, IEEE)支持的 VITAL 等。Verilog HDL 和 VHDL 语言是目前主要的两种 HDL 语言,并分别在 1995 年和 1987 年被采纳为 IEEE 国际标准,广泛用于数字集成电路的设计和验证领域。

## 1.3 Verilog HDL 的发展和国际标准

Verilog HDL 是一种常用的硬件描述语言,可以从系统级、电路级、门级到开关级等抽象层次,进行数字电路系统的建模、设计和验证工作。利用该语言可以设计出简单的门级电路,甚至功能完整的数字电路系统。

从 Verilog HDL 的设计初始到目前的广泛应用,经历了近 30 年的发展历程,其功能也由最初的模拟集成电路设计发展到数字和模拟电路设计(见图 1.3-1),它已经成为数字电路和数字集成电路中使用最为广泛的设计语言。

Verilog HDL 语言最初是由 Gateway Design Automation(GDA)公司于 1983 年为其模拟器



产品开发的硬件建模语言。作为一种便于使用的专用设计语言，Verilog HDL 被广泛用于模拟集成电路和仿真器中，并逐渐为众多设计者所接受。在随后的几年，Verilog HDL 开始在数字电路设计领域广泛使用。1987 年，Synopsys 公司开始使用 Verilog HDL 语言作为综合工具的输入，为在数字集成电路上的应用提供了 EDA 综合工具，提高了电路描述性设计方式的效率。

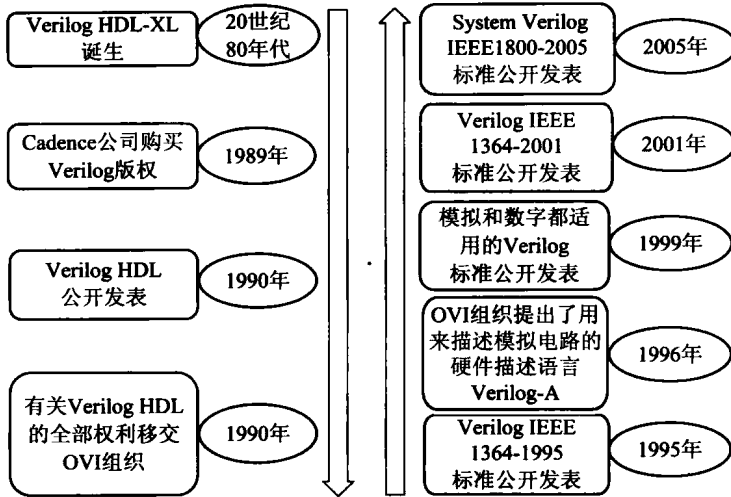


图 1.3-1 Verilog HDL 的发展历史

1989 年 Cadence 公司收购了 GDA 公司，Verilog HDL 语言成为 Cadence 公司的专有设计语言。为了在更大范围内推广和使用 Verilog HDL，1990 年 Cadence 公司决定公开 Verilog HDL 语言，于是成立了 OVI(Open Verilog International)组织，负责促进 Verilog HDL 语言的发展。

1993 年，几乎所有 ASIC 厂商都开始支持 Verilog HDL，并且认为 Verilog HDL-XL 是最好的仿真器。同时，OVI 组织推出 Verilog HDL 2.0 规范，IEEE 接受了将其作为 IEEE 标准的提案。自此，Verilog HDL 正式成为数字集成电路的设计语言标准(见表 1.3-1)。

表 1.3-1 Verilog HDL 国际标准

名称	时间	备注
Verilog IEEE 1364-1995	1995 年 12 月	基于 Verilog HDL 的优越性，IEEE 制定了 Verilog HDL 的 IEEE 标准，即 Verilog HDL 1364-1995
Verilog-A	1996 年	Verilog-A 是由 OVI 组织提出的一种硬件描述语言，是模拟电路行业的标准建模语言，来源于 IEEE 1364 Verilog 规范
	1999 年	模拟和数字都适用的 Verilog 标准公开发表
Verilog IEEE 1364-2001	2001 年	IEEE 制定了 Verilog IEEE 1364-2001 标准，并公开发表；其中 HDL 部分相对于 1995 年的标准有较大增强
System Verilog IEEE 1800-2005	2005 年	此标准是继 VHDL 和 Verilog HDL 之后仿真工具支持的语言，它建立在 Verilog HDL 语言的基础上，是 IEEE 1364 Verilog-2001 标准的扩展，兼容 Verilog-2001，将成为下一代硬件设计和验证的语言

1995 年底, IEEE 制定了第一个 Verilog HDL 语言标准 Verilog IEEE 1364-1995。在此基础上, 于 2001 年又增加了部分功能, 并制定了较为完善的标准 Verilog IEEE 1364-2001。目前在数字集成电路方面主要采用的就是这两个标准所规定的程序语法和设计规范。

Verilog HDL 在数字集成电路设计上的优越性, 使其在硬件设计领域得到了广泛的应用和发展。

在模拟电路设计方面, 基于 IEEE 1364 Verilog HDL 规范, 提出了模拟电路行业的标准建模语言 Verilog-A, 以提高模拟集成电路的程序化设计能力。

在系统级设计方面, 传统的设计方法采用 C 语言等高级软件语言进行数学模型的建立和分析, 通过定点化设计, 将数学模型转变成电路模型, 最后采用 HDL 语言进行电路设计。这种方法的缺点是, 数学模型的建立和电路设计是独立的, 从而导致设计周期长、需要的人员和软件多, 且存在重复性的工作等问题。研究和开发人员希望能将数学模型直接用于数字集成电路的设计, 以提高集成电路的设计效率, 这就给 EDA 工具厂商提出了新的要求。为了满足这一要求, 2005 年诞生了 System Verilog IEEE 1800-2005 标准。该标准建立在 Verilog HDL 语言的基础上, 在系统层次上增强了模型建立和验证的功能, 是 IEEE1364 Verilog-2001 标准的扩展, 向下兼容 Verilog-2001, 并将成为下一代硬件设计和验证的语言。

在功能设计方面, Verilog HDL 采用描述性建模方式, 通过行为描述、数据流描述和结构性描述等方式, 可以对电路、输入信号激励和响应监控方式进行设计。同时, 提供编程语言接口, 通过该接口可以在模拟、验证期间从设计外部访问设计, 包括模拟的具体控制和运行。

Verilog HDL 语言定义了完善的语法规则, 对每个语法结构都定义了清晰的模拟、仿真语义。它从 C 语言中继承了多种操作符和结构, 具有较强的扩展建模能力。Verilog HDL 语言的核心子集相对紧凑, 可以满足大多数建模应用的要求, 容易学习和掌握。当然, 应用于数字集成电路设计的较为完整的 Verilog HDL 语言还有很多的语法规则和使用方式, 需要进一步学习。本书主要针对 Verilog HDL 基本语法规则和数字集成电路设计进行讲述, 更为专业和细致的内容需要参照相关的国际标准和 EDA 工具的功能说明, 以应对越来越复杂的数字集成电路芯片设计和验证工作。

## 1.4 Verilog HDL 和 VHDL

目前, 最为常用的硬件描述语言有两种, 分别是 Verilog HDL 和 VHDL(VHSIC Hardware Description Language)。其中, VHSIC 是 Very High Speed Integrated Circuit 的缩写, 故 VHDL 准确的中文译名应为甚高速集成电路的硬件描述语言。

Verilog HDL 和 VHDL 都是完备的 HDL 设计和验证语言, 具有完整的设计方法和设计规范。它们可以设计和验证超大规模数字集成电路, 并且都已成为 IEEE 标准。选用哪种语言进行数字集成电路开发, 主要取决于设计单位的基础、计划采用的设计方案和 EDA 工具。这两种 HDL 语言具有较多的共同特点:

- (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, 具有电路仿真与验证机制, 以保证设计的正确性;

(4) 支持电路描述由高层到底层的综合转换;

(5) 硬件描述与实现工艺无关(有关工艺参数可通过语言提供的属性包括进去);

(6) 便于文档管理;

(7) 易于理解 and 设计重用。

但是作为两种不同的标准化 HDL 语言, Verilog HDL 和 VHDL 在设计方法和范围方面也有一些各自的特点。

(1) 在设计方法方面, VHDL 语言语法结构紧凑、灵活性差、设计规则繁琐, 初学者需要用较长时间掌握它。由于语法规则严谨性高, VHDL 可综合性和代码一致性很强, 适用于规模较大的数字集成电路系统设计。而 Verilog HDL 语言的语法结构和设计方式灵活, 初学者对语言掌握的难度较小, 设计也较容易进行综合和验证。但是由于设计代码风格的多样性, 当数字电路规模较大时, 代码的管理和系统设计难度较大。当然, 作为经验丰富的数字电路设计工程师, 采用何种语言进行设计的关键在于对语言和电路的掌握能力和对设计规范的理解程度。

为了发挥两种设计语言在设计方面各自的优势, EDA 工具厂商提供了 Verilog HDL 和 VHDL 语言的混合设计、验证和综合方法。因此, 设计人员只需掌握其中一种 HDL 语言即可。

(2) 在设计范围方面, Verilog HDL 和 VHDL 语言有一个显著的区别: Verilog HDL 可以描述系统级(System)、算法级(Algorithm)、寄存器传输级(RTL)、门级(Gate)和开关级(Switch)电路, 而 VHDL 则不具备开关级电路描述能力。在 FPGA 和 CPLD 等用户可配置数字电路设计中, 由于最小可配置电路是门级电路, 没有开关级可配置电路, 因此两种语言的设计能力相当。但是在专用数字集成电路设计和开关级描述方面, Verilog HDL 语言的设计范围比 VHDL 略大一些。

图 1.4-1 是 Verilog HDL 和 VHDL 在电路建模能力方面的比较。随着数字集成电路工艺和设计方法的快速发展, 这两种设计语言也在不断丰富和改进, 以满足更大、更高速、更复杂的数字集成电路系统设计的要求。

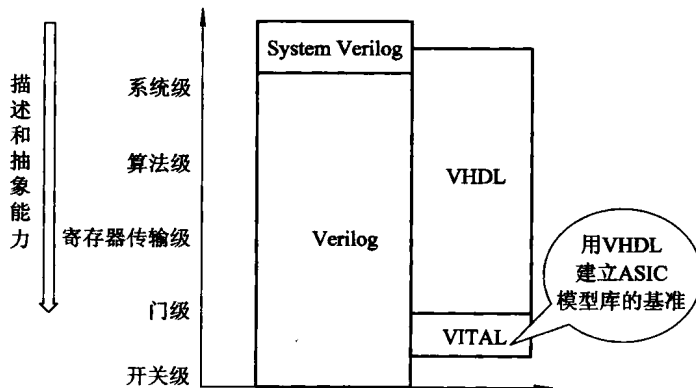


图 1.4-1 Verilog HDL 和 VHDL 建模能力比较



## 1.5 Verilog HDL 在数字集成电路设计中的优点

在数字集成电路出现的最初几十年中，数字逻辑电路和系统的设计规模较小，复杂度也低。ASIC、FPGA 和 CPLD 的设计工作采用厂家提供的专用电路图工具，通过连接线将定制电路单元进行互连实现。随着电路规模的增加，设计人员通常要花费很多的时间做大量重复的手工布线工作，同时为了达到设计目标，对于大量定制单元电路要分厂熟悉。这种低效率的设计方式持续了很长时间。

Verilog HDL 语言和 EDA 工具的出现和发展，通过运用高效率的描述性语言和强大的仿真综合工具，使设计人员将注意力集中于系统、算法和电路结构上，极大地提高了设计输入和验证的效率。

作为最广泛采用的 HDL 语言，Verilog HDL 在硬件描述方面的效率高、灵活性强。图 1.5-1 中的(a)和(b)分别是 4 位和 32 位总线与逻辑的原理图设计和 Verilog HDL 语言描述方式的对比。

原理图方式：



Verilog HDL 描述方式：

```
module aand4(a,b,c);  
    input [3:0]a,b;  
    output [3:0]c;  
    reg [3:0] c;  
    always @(a or b)  
        c=a&b;  
endmodule
```

(a) 4位总线与逻辑

原理图方式：



Verilog HDL 描述方式：

```
module aand32(a,b,c);  
    input [31:0]a,b;  
    output [31:0]c;  
    reg [31:0] c;  
    always @(a or b)  
        c=a&b;  
endmodule
```

(b) 32位总线与逻辑

图 1.5-1 组合逻辑电路原理图设计和 Verilog HDL 语言描述方式对比

图 1.5-2 中的(a)、(b)分别是长度为 4 位和 8 位移位寄存器的原理图设计与 Verilog HDL 语言描述方式的对比。