普通高等教育电气信息类规划教材

免费电子教案下载

www.cmpedu.com

Verilog HDL

数字系统设计原理与实践

王建民 编著

机械工业出版社
CHINA MACHINE PRESS

普通高等教育电气信息类规划教材

Verilog HDL 数字系统设计原理与实践

王建民　编著

藏书 图书馆

机械工业出版社

本书从应用角度出发，详细介绍了利用硬件描述语言进行数字电路设计的基本原理、基本概念和设计方法，包括 Verilog HDL 语法基础、组合逻辑电路、规则时序逻辑电路、有限状态机及数据通道设计，静态时序分析及跨时钟域数据传输的基本概念、设计方法及应用。全书通过大量、完整、规范的设计实例演示各类数字电路的设计过程和描述方法。每章配有习题，以指导读者深入地进行学习。

本书既可以作为电子科学与技术、集成电路设计相关专业本科、研究生数字集成电路前端设计教材，也可作为电子信息、电气工程和自动化相关专业 FPGA 应用设计课程教材使用。

本书配套授课电子课件，需要的教师可登录 www. cmpedu. com 免费注册，审核通过后下载，或联系编辑索取（QQ：308596956，电话：010 - 88379753）。

图书在版编目（CIP）数据

Verilog HDL 数字系统设计原理与实践/王建民编著. —北京：机械工业出版社，2017. 12
普通高等教育电气信息类规划教材
ISBN 978-7-111-59582-3

Ⅰ. ①V… Ⅱ. ①王… Ⅲ. ①硬件描述语言 - 程序设计 - 高等学校 - 教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2018）第 063820 号

机械工业出版社（北京市百万庄大街 22 号 邮政编码 100037）
责任编辑：时 静 责任校对：张艳霞
责任印制：李 飞

北京铭成印刷有限公司印刷

2018 年 5 月第 1 版第 1 次印刷
184mm × 260mm · 22. 5 印张 · 546 千字
0001—3000 册
标准书号：ISBN 978-7-111-59582-3
定价：59.80 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

电话服务
服务咨询热线：(010)88379833
读者购书热线：(010)88379649

网络服务
机 工 官 网：www. cmpbook. com
机 工 官 博：weibo. com/cmp1952
教育服务网：www. cmpedu. com
金 书 网：www. golden - book. com

封面无防伪标均为盗版

前　言

近年来，硬件描述语言（Hardware Description Language）逐渐取代传统的设计方法，成为数字电路设计的主流方法。本书坚持“用语言、讲设计、重实践”的建设思路，打破传统教材以介绍硬件描述语言语法或者软件使用为重点的传统，以数字电路结构为主线安排教学内容，通过大量完整、规范的设计实例介绍基于 Verilog HDL 的寄存器传输级（Register Transfer Level，RTL）数字电路设计的基本概念和实现方法。

本书从 Verilog HDL 基础语法讲起，由浅入深，系统地介绍组合逻辑电路设计、规则时序逻辑电路、有限状态机、有限状态机 + 数据通道、静态时序分析和跨时钟域设计的基本概念和实现方法；讨论了数字电路结构与面积、速度关系和优化方法及 Verilog HDL 实现方式，内容涵盖数字电路（前端）设计的全部内容。书中对于数字电路基本原理和设计方法的描述，全部是通过难易程度不同的设计实例演示，大部分设计实例通过简单扩展可以直接应用于具体设计，具有很好的参考价值。本书可以作为电子科学与技术、集成电路设计相关专业本科、研究生数字集成电路前端设计基础教材使用，也可作为电子信息、电气工程和自动化相关专业教授 FPGA 应用设计课程教材使用。对于有经验的数字电路设计工程师也会有一定的参考价值。

书中全部设计实例在 Quartus II 13.0 和 ModelSim 10.2 软件环境下编译通过，授课教师在教学过程中可酌情考虑取舍。建议授课学时：48 学时；实验学时：16 学时。授课教师可以通过 wjmfuzzy@126.com 邮箱，申请本教材配套的 CAI 课件、习题答案、实验指导书及实验源代码。由于综合软件限制，本书元器件符号采用 ANSI/IEEE 标准，与国际符号的对照表见附录 B。

研究生兰风宇、崔新莹和姚博文校对了部分书稿和代码，李晓和李喆绘制了部分插图，在此表示真诚的感谢。感谢机械工业出版社时静编辑在本书出版过程中给予的无私帮助。

由于时间仓促，书中难免存在不妥之处，请读者原谅，并提出宝贵意见。

作　者

目　录

前言
第1章　数字系统设计概述 …… 1
1.1　引言 …… 1
1.2　模拟电路和数字电路 …… 1
1.2.1　模拟信号和数字信号 …… 1
1.2.2　模数转换 …… 2
1.2.3　模拟电路和数字电路 …… 3
1.3　数字电路设计 …… 3
1.3.1　数字电路与系统 …… 3
1.3.2　数字电路设计流程 …… 5
1.4　硬件描述语言 …… 6
1.5　习题和思考题 …… 7
第2章　数字电路基础 …… 8
2.1　变量和函数 …… 8
2.2　基本逻辑关系 …… 8
2.2.1　逻辑与 …… 8
2.2.2　逻辑或 …… 9
2.2.3　逻辑反 …… 9
2.3　逻辑门和数字电路 …… 10
2.3.1　晶体管 …… 10
2.3.2　逻辑门 …… 11
2.3.3　逻辑电路的表示 …… 11
2.4　布尔代数和卡诺图 …… 12
2.4.1　布尔代数 …… 13
2.4.2　最小项的定义及其性质 …… 14
2.4.3　卡诺图法化简逻辑函数 …… 15
2.5　CMOS 逻辑门电路 …… 17
2.5.1　NMOS 逻辑门 …… 17
2.5.2　CMOS 逻辑门 …… 19
2.6　设计实现 …… 20
2.6.1　标准芯片 …… 21
2.6.2　可编程逻辑器件 …… 22

试读结束：需要全本请在线购买：www.ertongbook.com

2.6.3 全定制芯片、标准单元和门阵列 …… 27
2.7 习题和思考题 …… 29
第3章 Verilog HDL硬件描述语言 …… 30
3.1 基本概念 …… 30
3.1.1 模块 …… 30
3.1.2 空白和注释 …… 33
3.1.3 关键字 …… 34
3.1.4 标识符 …… 34
3.2 数据类型 …… 35
3.2.1 四值逻辑系统 …… 35
3.2.2 线网和变量 …… 35
3.2.3 有符号和无符号数 …… 36
3.3 层次化设计 …… 37
3.3.1 设计方法学 …… 37
3.3.2 模块实例 …… 38
3.3.3 端口连接规则 …… 39
3.4 门级描述 …… 39
3.4.1 多输入门 …… 39
3.4.2 多输出门 …… 40
3.4.3 三态门 …… 40
3.4.4 门阵列实例 …… 41
3.5 任务和函数 …… 42
3.5.1 任务 …… 42
3.5.2 函数 …… 43
3.5.3 任务和函数的区别 …… 44
3.5.4 设计实例：格雷码计数器 …… 44
3.6 可重用设计 …… 48
3.6.1 宏定义 …… 48
3.6.2 条件编译 …… 49
3.6.3 参数 …… 50
3.7 习题和思考题 …… 52
第4章 组合逻辑电路设计 …… 54
4.1 组合逻辑电路 …… 54
4.2 连续赋值语句 …… 54
4.3 组合逻辑 always 块 …… 54
4.4 Verilog HDL操作符 …… 56
4.4.1 表达式 …… 56
4.4.2 操作数 …… 56
4.4.3 操作符 …… 57
4.4.4 操作符优先级 …… 60

4.5 if 语句 …… 60
4.5.1 基本语法 …… 60
4.5.2 设计实例 …… 61
4.6 case 语句 …… 64
4.6.1 基本语法 …… 64
4.6.2 设计实例 …… 65
4.7 决策树 …… 67
4.7.1 full case 和 parallel case …… 67
4.7.2 优先结构路由网络 …… 68
4.7.3 并列结构路由网络 …… 70
4.8 组合逻辑电路设计实例 …… 72
4.8.1 有符号加法器 …… 72
4.8.2 移位器 …… 73
4.8.3 三态逻辑 …… 75
4.8.4 浮点数加法器 …… 76
4.8.5 组合逻辑乘法器 …… 79
4.9 设计优化 …… 81
4.9.1 操作符共享 …… 81
4.9.2 布局相关的电路 …… 83
4.9.3 功能共享 …… 88
4.10 组合逻辑电路的设计要点 …… 91
4.10.1 组合逻辑电路设计的常见错误 …… 91
4.10.2 组合逻辑电路设计规则 …… 94
4.11 组合逻辑电路 Testbench …… 95
4.11.1 仿真逻辑的构成 …… 95
4.11.2 组合逻辑电路 Testbench 实例 …… 95
4.12 习题和思考题 …… 97
第 5 章 规则时序逻辑电路设计 …… 99
5.1 时序逻辑电路 …… 99
5.1.1 时序逻辑电路结构及工作过程 …… 99
5.1.2 时序逻辑电路的描述 …… 100
5.2 基本存储元件 …… 100
5.2.1 D 锁存器 …… 100
5.2.2 D 触发器 …… 101
5.2.3 寄存器和寄存器文件 …… 104
5.3 规则时序逻辑电路设计实例 …… 105
5.3.1 计数器 …… 105
5.3.2 移位寄存器 …… 108
5.3.3 线性反馈移位寄存器 …… 109
5.3.4 同步 FIFO …… 112

5.4 循环语句 …… 117
5.4.1 for 循环语句 …… 117
5.4.2 while 语句 …… 120
5.5 生成语句 …… 121
5.5.1 循环生成语句 …… 121
5.5.2 条件生成语句 …… 123
5.5.3 case 生成语句 …… 123
5.6 时序逻辑电路 Testbench …… 124
5.7 设计陷阱 …… 127
5.7.1 阻塞赋值和非阻塞赋值 …… 127
5.7.2 组合逻辑环 …… 128
5.7.3 异步信号的误用 …… 128
5.7.4 门控时钟的误用 …… 130
5.7.5 导出时钟的使用 …… 131
5.8 习题和思考题 …… 132
第 6 章 有限状态机设计原理 …… 134
6.1 有限状态机 …… 134
6.1.1 米利状态机和摩尔状态机 …… 134
6.1.2 边沿检测电路 …… 135
6.1.3 米利状态机和摩尔状态机的比较 …… 137
6.2 状态转换图和算法状态机图 …… 138
6.2.1 状态转换图 …… 138
6.2.2 算法状态机图 …… 139
6.3 有限状态机的时序 …… 142
6.4 状态赋值 …… 143
6.4.1 未用状态的处理 …… 144
6.4.2 状态赋值对电路的影响 …… 144
6.4.3 超前输出电路 …… 148
6.5 有限状态机的实现 …… 152
6.5.1 代码风格 …… 152
6.5.2 Verilog HDL 状态赋值 …… 152
6.5.3 两段式 always 块 …… 155
6.5.4 多段式 always 块 …… 158
6.5.5 一段式 always 块 …… 161
6.6 设计实例 …… 164
6.6.1 序列检测器 …… 164
6.6.2 键盘扫描电路 …… 166
6.6.3 仲裁电路 …… 170
6.6.4 BCD 码余 3 码转换电路 …… 174
6.7 习题和思考题 …… 177

第7章 有限状态机设计实践 …… 179
7.1 轨道车控制器 …… 179
7.1.1 问题描述 …… 179
7.1.2 轨道车运行方向输出信号 …… 179
7.1.3 开关位置输出信号 …… 180
7.1.4 传感器输入信号 …… 180
7.1.5 设计实现 …… 180
7.2 飞机起落架控制器 …… 184
7.2.1 问题描述 …… 184
7.2.2 设计实现 …… 184
7.3 存储器控制器 …… 188
7.3.1 SRAM 读写时序 …… 188
7.3.2 SRAM 控制器数据通道 …… 191
7.3.3 SRAM 控制器控制通道 …… 191
7.4 通用异步收发器 UART …… 198
7.4.1 接收模块 …… 199
7.4.2 发送模块 …… 205
7.5 习题和思考题 …… 208
第8章 时序分析基础 …… 209
8.1 组合逻辑电路的传播延迟 …… 209
8.1.1 组合逻辑电路传播延迟的定义 …… 209
8.1.2 传播延迟产生的后果 …… 210
8.1.3 传播延迟的计算 …… 210
8.2 时序逻辑电路的传播延迟 …… 211
8.2.1 引脚到引脚延迟路径 …… 211
8.2.2 输入到寄存器数据输入延迟路径 …… 212
8.2.3 时钟到输出延迟路径 …… 213
8.2.4 寄存器到寄存器延迟路径 …… 214
8.2.5 时序逻辑电路的最高工作频率 …… 215
8.2.6 建立时间和保持时间的调整 …… 215
8.3 提高电路的最高工作频率 …… 217
8.4 调整电路的建立时间和保持时间 …… 218
8.5 同步电路的时序分析方法 …… 220
8.5.1 建立时间和最高工作频率 …… 220
8.5.2 保持时间 …… 221
8.5.3 输出相关的时序参数 …… 221
8.5.4 输入相关的时序参数 …… 222
8.6 带有时钟偏斜情况的时序分析 …… 222
8.6.1 时钟偏斜对同步设计的影响 …… 222
8.6.2 时钟偏斜对于建立时间和最高时钟频率的影响 …… 223

8.6.3 时钟偏斜对保持时间约束的影响 …… 224
8.7 习题和思考题 …… 225
第9章 数据通道设计原理 …… 226
9.1 数据通道 …… 226
9.2 寄存器传输级设计 …… 227
9.2.1 算法 …… 227
9.2.2 数据流模型 …… 227
9.2.3 寄存器传输级设计 …… 229
9.3 FSMD 设计原理 …… 229
9.3.1 寄存器传输操作 …… 229
9.3.2 数据通道 …… 231
9.3.3 控制通道 …… 231
9.4 FSMD 设计 …… 231
9.4.1 ASMD 图 …… 232
9.4.2 FSMD 设计方法 …… 233
9.4.3 在条件判断框中使用寄存器 …… 238
9.4.4 FSMD 的 Verilog HDL 描述 …… 239
9.4.5 FSMD 设计的资源共享 …… 243
9.5 流水线设计 …… 246
9.5.1 吞吐率和 Latency …… 246
9.5.2 流水线设计 …… 247
9.5.3 流水线逻辑电路设计 …… 248
9.6 设计实例：访问 RAM …… 255
9.6.1 问题描述 …… 255
9.6.2 数据通道 …… 255
9.6.3 控制通道设计 …… 259
9.7 习题和思考题 …… 261
第10章 数据通道设计实践 …… 262
10.1 问题描述 …… 262
10.2 定点数的表示 …… 262
10.3 饱和算术运算 …… 263
10.3.1 饱和加法运算 …… 263
10.3.2 饱和乘法运算 …… 264
10.4 设计实例：混合方程 …… 264
10.4.1 混合方程 …… 264
10.4.2 混合方程的直接实现 …… 267
10.5 流水线设计 …… 270
10.6 面积优化方法 …… 273
10.6.1 资源共享数据通道的设计 …… 273
10.6.2 握手信号 …… 277

10.6.3 输入总线数据通道 …… 279
10.7 递归和初始化 …… 284
10.8 寄存器的 Schedule …… 288
10.9 设计优化 …… 294
10.9.1 电路结构与速度 …… 294
10.9.2 电路结构与面积 …… 307
10.10 习题和思考题 …… 312
第 11 章 跨时钟域数字设计 …… 313
11.1 时钟域 …… 313
11.2 亚稳态 …… 313
11.3 基本同步电路 …… 314
11.3.1 电平同步电路 …… 314
11.3.2 边沿同步电路 …… 316
11.3.3 脉冲同步电路 …… 317
11.4 同步多个控制信号 …… 318
11.4.1 两个控制信号 …… 318
11.4.2 具有相位关系的控制信号 …… 319
11.4.3 编码控制信号 …… 320
11.5 握手协议 …… 322
11.5.1 四步握手协议 …… 322
11.5.2 两步握手协议 …… 326
11.6 跨时钟域数据传输 …… 328
11.6.1 四步握手协议数据传输 …… 328
11.6.2 两步传输协议数据传输 …… 335
11.6.3 单步握手协议数据传输 …… 336
11.7 通过存储器传输数据 …… 337
11.7.1 异步 FIFO 缓冲器 …… 337
11.7.2 共享存储器实现交换数据 …… 338
11.8 异步 FIFO 设计 …… 338
11.8.1 异步 FIFO 设计——方式 1 …… 338
11.8.2 异步 FIFO 设计——方式 2 …… 342
11.9 习题和思考题 …… 347
附录 …… 348
附录 A Verilog HDL 关键字（IEEE Std. 1364 - 2001） …… 348
附录 B 常用逻辑符号对照表 …… 349
参考文献 …… 350

试读结束：需要全本请在线购买：www.ertongbook.com

第1章　数字系统设计概述

本章介绍数字系统设计的基本概念、发展趋势以及基于硬件描述语言的数字电路设计流程，并对 Verilog HDL 和 VHDL 进行比较和分析。

1.1　引言

数字电路凭借成本低、抗干扰能力强等诸多优点已经成为各类电子系统的首选。事实上，不仅是控制系统、通信系统，甚至某些机械系统都被“数字化”，采用数字化的方式存储、处理以及传输信息。

过去几十年，数字电路制造技术和设计能力都发生了巨大改变。数字集成电路芯片的集成度不断提高，芯片内部集成的晶体管（Transistor）个数甚至多达上亿。与此同时，芯片体积变得更小，速度更快，成本更低，功能也更丰富。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）逐渐成为数字系统的主流器件。曾经广泛应用的以 74 系列为代表的标准芯片已经逐渐退出历史舞台。

集成电路技术突飞猛进，对数字电路设计方法提出更高的要求，传统设计方法无法满足。从 20 世纪 70 年代末开始，各大 EDA（Electronic Design Automation）公司、教育和科研机构纷纷提出了自己的硬件描述语言（Hardware Description Language，HDL），用来描述数字系统的结构或者功能。采用硬件描述语言，设计者可以从更高的抽象层次对数字系统进行描述（建模），采用综合软件获得实际电路结构，对电路功能和时序进行仿真和验证。设计者不必过分关心电路实现的具体细节，从而能够将主要精力集中在电路功能设计上。

目前，硬件描述语言已经成为数字系统设计的主流方法，其使用范围也不仅限于集成电路设计。随着现场可编程门阵列技术的不断进步和发展，越来越多的电路设计人员开始采用硬件描述语言进行数字电路的设计。事实上，硬件描述语言已经成为数字电路设计的基本工具，成为从事电路设计相关人员必须掌握的基本工具。

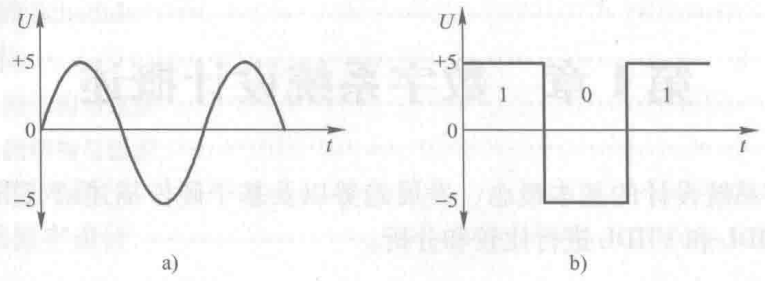
1.2　模拟电路和数字电路

1.2.1　模拟信号和数字信号

电信号是指随时间变化的电压或者电流。通过传感器，非电物理量可以转换成电信号，同时由于方便传输和处理等特性，电信号成为应用最为广泛的信号。电信号分为模拟信号和数字信号两类。模拟信号的时间和幅值都连续变化；数字信号的时间和幅值都是离散的，如图 1-1 所示。

时间离散是指数字信号只能在某些时间点上取值，这些时间点之外的其他时间点上信号无

定义。注意：是无定义，而并非为0。幅值离散指采用有限字长数值表示信号值的过程，因此数字信号只能取预先定义的某些值（图1-1b中数字信号只能取-5或者+5）。



a) b)

图1-1 典型模拟信号和数字信号

a）典型的模拟信号 b）典型的数字信号

1.2.2 模数转换

模拟信号经过采样得到对应的数字信号。采样过程可以分为两个步骤：时域采样和量化编码。图1-2给出了模拟数字信号的转换过程。图1-2a所示为模拟电压信号，为了将其转换为数字信号，首先对模拟信号采样（采样定理），得到的采样信号（离散时间信号）时间离散、幅值连续，t_0、t_1为取样点，如图1-2b所示。所谓幅值连续，是指信号取值与原来模拟信号一样。接下来对幅值进行量化，即数字化。选取一个量化单位，将采样信号除以量化单位并取整，得到时间、幅值都离散的数字量，最后对得到的数字量进行编码，产生用0和1表示的数字信号，如图1-2c所示。

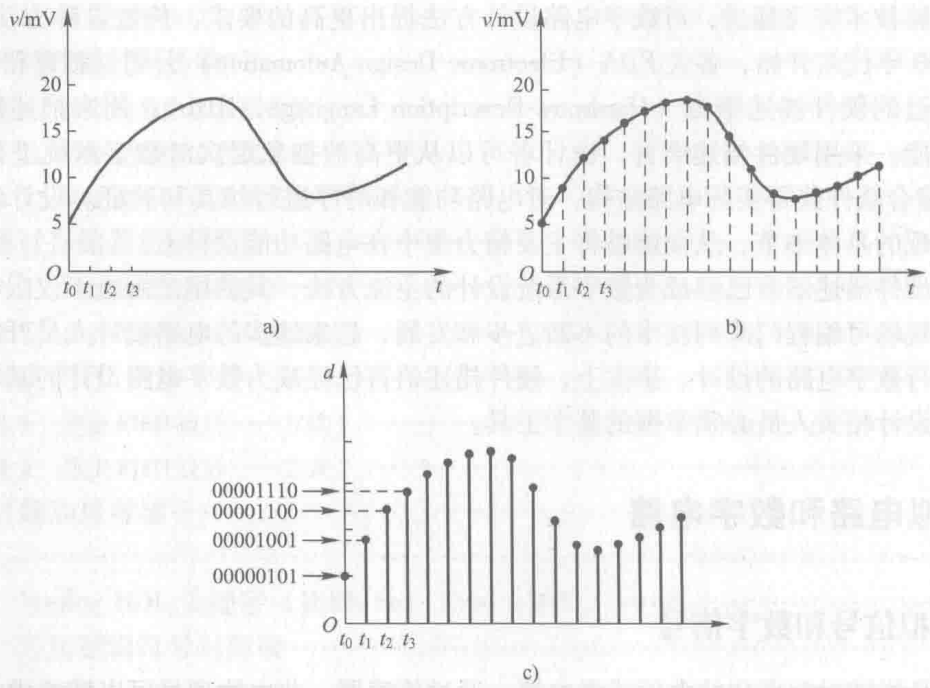


图1-2 模数转换过程

数字信号可以表示事物的不同状态（即数字信号表示逻辑变量），如灯的开与关、开关的导通与关断以及任务的成功与失败等。下面考虑采用数字信号表示二进制逻辑变量的方法，二进制逻辑变量指只能取逻辑0和逻辑1两个值的逻辑变量。

为了采用两个值表示电压，首先需要定义一个阈值电压（Threshold voltage），低于阈值电压的电压值用一个逻辑值（0或者1）表示，高于阈值电压的电压值用另一个逻辑值（1或者0）表示。如果逻辑0表示低于阈值电压的电压值，逻辑1表示高于阈值电压的逻辑值，称为正逻辑系统（Positive logic system）；逻辑0表示高于阈值电压的电压值，逻辑1表示低于阈值电压的电压值则称为负逻辑系统（Negative logic system）。如无特别指出，本书考虑正逻辑系统。

正逻辑系统中，逻辑0和逻辑1分别称为“低电平”和“高电平”。为了表示阈值电压，首先定义高、低电平的电压范围，如图1-3所示。图1-3给出电路的最低和最高电压，分别用V_{SS}和V_{DD}表示。通常，V_{SS}等于0 V，用Gnd表示，V_{DD}即电源电压，一般在1～5 V之间。图1-3中，Gnd到$V_{0,max}$电压范围表示逻辑0，$V_{0,max}$表示低电平（逻辑0）最高电压。类似地，$V_{1,min}$到V_{DD}电压范围表示逻辑1，$V_{1,min}$表示高电平最小电压。$V_{0,max}$和$V_{1,min}$取值与具体电路有关。典型情况下，$V_{0,max}$取V_{DD}的40%，$V_{1,min}$取V_{DD}的60%。$V_{0,max}$和$V_{1,min}$之间电压值未定义，除非电平转换过程（从逻辑0变为逻辑1，或者从逻辑1变为逻辑0），逻辑信号实际电压值不能位于该范围。

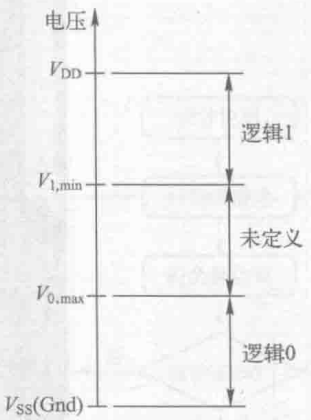


图1-3　逻辑电平定义

1.2.3　模拟电路和数字电路

传输和处理数字信号的电子电路称为逻辑电路。二进制逻辑电路中，信号只能取0和1；十进制逻辑电路中，信号可取0～9。由于信号值都表示数字，逻辑电路也称为数字电路。传输和处理模拟信号的电子电路称为模拟电路。模拟电路中，信号在某个范围内连续取值。

与模拟电路相比，数字电路具有诸多优点。数字电路将某一电压范围识别为高电平或者低电平，具有更好的抗干扰性。同时，更高的集成度有助于降低制造和设计难度，电路结构简单，容易制造，便于集成和系列化生产。数字电路不仅可以完成逻辑运算、数值运算，在数字控制系统中也不可缺少。因此，数字电路在通信、自动控制、自动化测量及计算机等各个学科领域都得到了广泛的应用。

1.3　数字电路设计

设计是一个复杂的过程，更像一个艺术创作过程，而不是一个科学发现的过程。设计过程绝不是简单地执行事先确定的工作步骤，实践是学习设计的最好方法，也是唯一途径。

1.3.1　数字电路与系统

典型数字系统由一块或多块印制电路板（Printed circuit board，PCB）[⊖]组成，PCB由若干芯片和其他元器件（电源及各种连接器等）组成，如图1-4所示。电路核心功能一般由PCB板上的芯片实现，芯片分为两种类型：专用集成电路和可编程逻辑器件（Programmable Logic Device，PLD）。图1-5给出了一块集成电路芯片的内部结构。典型情况下，无论是专用集成电

⊖ PCB板上也可能包括部分模拟电路或者其他辅助电路，比如电源等。

路还是可编程逻辑器件，内部都是若干相互连接的逻辑电路。逻辑电路实现的功能各不相同，如算术运算、存储数据以及数据流控制等。

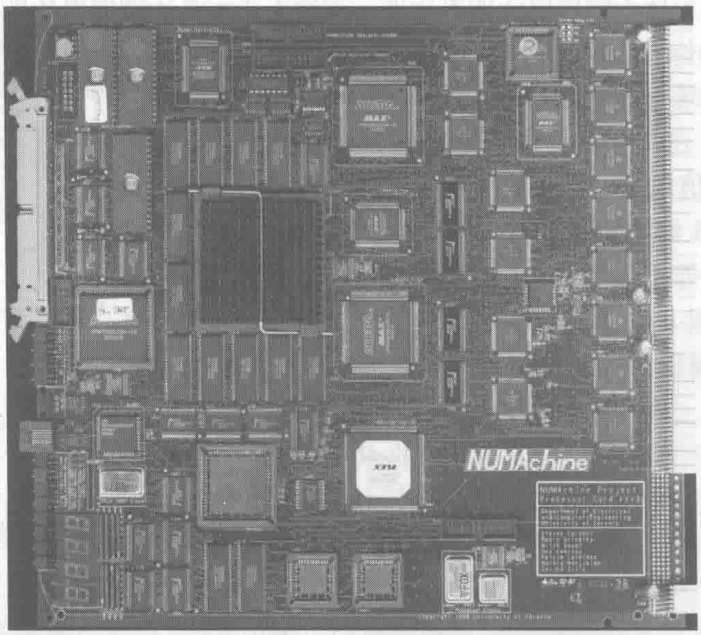


图 1-4　典型数字系统

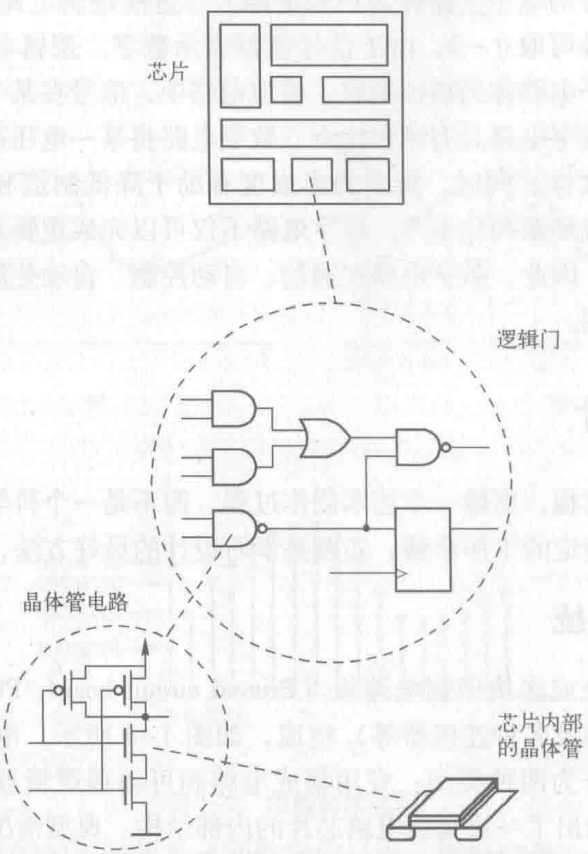


图 1-5　数字电路分层次描述

逻辑电路（逻辑网络）由逻辑门（Logic gate）互联形成，逻辑门执行基本的逻辑功能，是逻辑电路的基本组成单元，逻辑门由更基本的功能单元晶体管组成。

本书重点关注逻辑电路设计问题，即专用集成电路或可编程逻辑器件内部逻辑电路功能设计问题，逻辑电路和逻辑门的实现方式不属于本书的讨论范围。考虑到完整性，在2.5节和2.6节将简单讨论逻辑电路的实现问题。

1.3.2 数字电路设计流程

图1-6给出数字系统设计的典型流程。根据设计目标的不同，设计流程可能会稍有不同。

任何设计都从编写详细的设计说明（Specifications）开始，设计说明必须详细给出设计需要满足的性能指标和功能，包括电路功能、输入/输出信号（接口）以及总体结构。此外，某些设计说明还会详细说明电路的时序、逻辑资源、功耗、可测性以及故障覆盖率等内容。

根据设计说明对系统进行功能模块划分，将复杂系统划分为规模不同的子系统，模块划分需要遵循的一些设计原则，比如功能原则、时钟域（Timing domain）划分原则等，关于模块划分原则将会在后续章节具体讨论。

完成功能模块划分后，需要设计系统的行为级模型（行为级描述）。行为级描述主要用于电路的功能、性能及兼容性等问题的分析和验证，进一步确定后续的开发和设计过程。现代硬件描述语言（Verilog HDL和VHDL）支持行为级描述，行为级描述中允许使用硬件描述语言的高级语法结构（不考虑综合问题）。

仿真是正确实现设计的关键环节，验证设计者的设计是否正确（符合设计说明）及在设计实现过程中引入各种参数后，设计功能是否依然正确无误。仿真主要分为功能仿真和时序仿真两种。功能仿真在设计输入完成后进行，时序仿真是在逻辑综合后或布局布线后进行。逻辑综合或者布局布线以前的仿真称作功能仿真，分为综合前仿真（Pre-Synthesis Simulation）和综合后仿真（Post-Synthesis Simulation），目的是在不考虑时序信息的情况下，验证设计功能正确与否。时序仿真利用逻辑综合或者布局布线器件时序信息，在最坏情况下对电路行为做出评估。通常情况下，时序仿真和功能仿真使用相同的仿真软件、仿真流程和激励信号，区别在于时序仿真包括基于实际布局布线设计的最坏情况的布局布线延时。

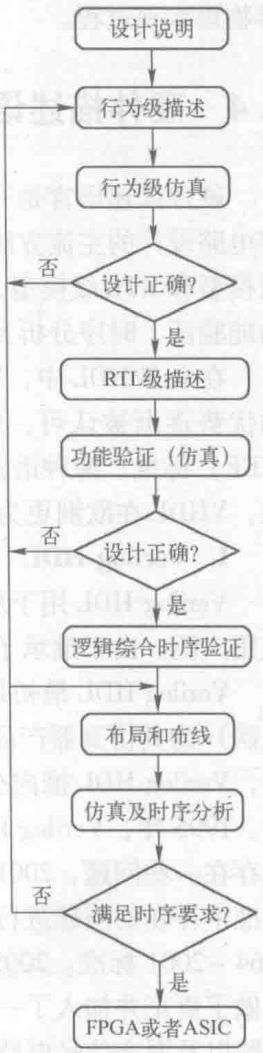


图1-6 数字电路设计流程

逻辑综合将高抽象层次的描述转换为低抽象层次描述。寄存器传输级（Register transfer level，RTL）综合就是将RTL级描述转换为门级电路网表的过程。综合过程需要指定ASIC工艺库或者具体的现场可编程门阵列器件，综合软件采用器件库提供的标准单元将RTL级描述转换为门级网表。综合过程分为三个步骤：

1）翻译（Translation）：综合软件读取RTL代码并将其转换成门级网表；综合过程要确保门级网表的输入/输出关系与RTL级描述的输入/输出关系保持一致。

2）优化（Optimization）：对门级网表进行优化，优化是个迭代搜索过程，并不是求解过

程，因此综合软件的优化只是局部优化。

3）映射（Maping）：采用器件库内的标准元件或者 FPGA 内部的逻辑单元实现优化后的门级网表。

如果数字系统实现的平台是 FPGA，设计者需要采用 FPGA 厂家提供的软件，进行布局（Layout）和布线（Routing）以及适配（Fitter）过程。经过以上过程可以获得更详细的时序信息，进行时序仿真以及时序分析。通过时序仿真，验证综合前后电路功能是否一致，从而确定综合结果是否正确。如果采用 ASIC 实现，则需要进行掩模板设计及布局、布线及时钟树综合等物理实现过程。

1.4 硬件描述语言

硬件描述语言是一种形式化语言，描述数字电路的结构和功能。硬件描述语言已经成为数字电路设计的主流方法，数字电路设计流程的每个关键步骤都采用硬件描述语言。电路的行为级模型和 RTL 级模型，利用综合软件将 RLT 级模型转换成实际的物理电路，以及之后电路的功能验证、时序分析及故障覆盖率的测试都需要硬件描述语言。

在众多 HDL 中，VHDL（Very High Speed Integrated Circuit HDL）和 Verilog HDL 凭借自身的优势逐渐被认可，成为电气电子工师协会（Institute of Electrical and Electronics Engineers，IEEE）标准。两种语言都有自己的使用者，其中 Verilog HDL 在美国、日本以及我国比较受欢迎，VHDL 在欧洲更为普及。

1. Verilog HDL

Verilog HDL 用于从算法级、门级到开关级的多种抽象层次的数字系统建模和验证过程，使用广泛。由于继承了 C 语言的很多语法结构，与 VHDL 相比，Verilog HDL 更容易上手。

Verilog HDL 最初是由 Gateway Design Automation 公司（后来被 Cadence Design Systems 公司收购）为其仿真器产品开发的，初衷是用于公司开发的仿真器产品，是一种专用语言。1990 年，Verilog HDL 推向公众领域，目前由 Open Verilog International（OVI）组织负责升级和维护。1995 年，Verilog HDL 成为 IEEE 标准（IEEE Std1364 - 1995）。IEEE Std 1364 - 1995 标准中存在一些问题，2001 年，IEEE 发布了 IEEE Std 1364 - 2001 标准，对 IEEE Std 1364 - 1995 标准中存在的问题进行了改进并增加了一些新的特性。目前，绝大多数 EDA 软件都支持 Std 1364 - 2001 标准。2005 年，IEEE 发布了 IEEE Std 1364 - 2005 标准，对原标准出现的一些问题做了更正并加入了一些新的语言特征，同时增加了一个独立标准 Verilog - AMS，尝试对模拟电路以及混合信号电路的支持。

2. VHDL

VHDL 是另外一个成为 IEEE 标准的硬件描述语言。最初由美国国防部（Department of Defense）组织开发，旨在提高设计的可靠性和缩减开发周期。最初的 VHDL 是一种小范围使用的设计语言。1987 年底，VHDL 成为 IEEE 标准（IEEE Std 1076 - 1987）。自 IEEE 公布了 VHDL 的标准版本 IEEE Std 1076 - 1987 之后，各 EDA 公司相继推出了自己的 VHDL 设计环境，或宣布自己的设计工具可以和 VHDL 接口。此后 VHDL 在电子设计领域得到了广泛的应用，并逐步取代了原有的非标准的硬件描述语言。1993 年，IEEE 对 VHDL 进行了修订，从更高的抽象层次和系统描述能力上扩展 VHDL 的描述功能，公布了新版本的 VHDL，即 IEEE Std 1076 - 1993 版本。

试读结束：需要全本请在线购买：www.ertongbook.com