



21世纪

高等院校电子信息类本科规划教材

数字系统的 Verilog HDL设计

江国强 编著



机械工业出版社
China Machine Press

21世纪

高等院校电子信息类本科规划教材



数字系统的 Verilog HDL设计

江国强 编著



机械工业出版社
China Machine Press

本书以硬件描述语言为工具,介绍了数字电路及系统的设计方法。本书内容包括数制与编码、逻辑代数与 Verilog HDL 基础、门电路、组合逻辑电路、触发器、时序逻辑电路、半导体存储器、数模与模数转换、数字系统设计、可编程逻辑器件、Verilog HDL 仿真和 Verilog HDL 综合与优化。书中还列举了大量的基于 Verilog HDL 的门电路、触发器、组合逻辑电路、时序逻辑电路、存储器和数字系统设计的实例,供读者参考。每个设计实例都经过了电子设计自动化(EDA)软件的编译和仿真,确保无误。每章后均附有思考题和习题。

本教材可作为高等院校电子、信息、通信、自动化类专业的数字电子技术、EDA 技术、硬件描述语言等相关课程的教材和相关工程技术人员的参考资料。

版权所有,侵权必究。

本书法律顾问 北京市展达律师事务所

图书在版编目(CIP)数据

数字系统的 Verilog HDL 设计/江国强编著. —北京:机械工业出版社,2007.8
(21 世纪高等院校电子信息类本科规划教材)

ISBN 978-7-111-21622-3

I. 数… II. 江… III. ①数字系统—系统设计—高等学校—教材 ②硬件描述语言, VHDL—程序设计—高等学校—教材 IV. TP271 TP312

中国版本图书馆 CIP 数据核字(2007)第 091844 号

机械工业出版社(北京市西城区百万庄大街 22 号 邮政编码 100037)

责任编辑:秦燕梅

北京慧美印刷有限公司印刷·新华书店北京发行所发行

2007 年 8 月第 1 版第 1 次印刷

184mm×260mm·17.75 印张

定价:26.80 元

凡购本书,如有倒页、脱页、缺页,由本社发行部调换
本社购书热线:(010)68326294

前 言

在 20 世纪 90 年代,国际上电子技术和计算机技术先进的国家,一直在积极探索新的电子电路设计方法,在设计方法、工具等方面进行了彻底的变革,并取得了巨大成功。在电子设计技术领域,可编程逻辑器件(PLD)的应用,已得到很好的普及,为数字系统的设计带来了极大的灵活性。由于 PLD 可以通过软件编程而对其硬件结构和工作方式进行重构,使得硬件设计可以如同软件设计那样方便快捷,因而极大地改变了传统的数字系统设计方法、设计过程和设计理念。随着可编程逻辑器件集成规模不断扩大、自身功能不断完善,以及计算机辅助设计技术的提高,现代电子系统设计领域的电子设计自动化(Electronic Design Automation, EDA)技术便应运而生。

目前,EDA 技术这门课程已经成为很多高等院校本科生的必修课程,但在此之前同学们得先修数字电子技术课程。然而这两门课程的内容是相互关联的,衔接是十分紧密的。为了协调这种关系,本书把两门课程的内容紧密地融为一体,在教学中可以把数字电子技术和 EDA 技术合并为一门课程,在此课程学习的基础上,增加一门 EDA 实训课程,使学生们在 EDA 技术的学习中,既学到了理论,又可得到实际设计的锻炼。

《数字系统的 Verilog HDL 设计》是基于硬件描述语言(Hardware Description Language, HDL)编写的。目前,国际最流行的、并成为 IEEE 标准的两种硬件描述语言是 VHDL 和 Verilog HDL。两种 HDL 各具特色,但 Verilog HDL 是在 C 语言的基础上演化而来的,只要具有 C 语言的编程基础,就很容易学会并掌握这种语言,而且国内外 90% 的电子公司都把 Verilog HDL 作为企业标准设计语言,因此本书以 Verilog HDL 作为数字电路与系统的设计工具。

本书保持了数字电路及系统内容的完整性和理论的系统性,在每章的开始都给出了教学要点和教学安排,全部内容可安排 64~72 学时完成。

第 1 章数制与编码。介绍了脉冲信号和数字信号的特点、数制及其转换、二-十进制编码和字符编码,还介绍了现代数字系统的设计方法。

第 2 章逻辑代数与 Verilog HDL 基础。介绍分析和设计数字逻辑电路功能的数学方法。首先介绍了逻辑代数的基本概念、逻辑函数及其表示方法、基本公式、常用公式和重要定理,然后介绍了 Verilog HDL 硬件描述语言的基本知识,作为数字系统的设计基础。

第 3 章门电路。门电路是构成数字系统的基本单元,本章介绍各种类型(TTL 和 CMOS 等)门电路的结构、逻辑功能和电气特性,为实际使用这些器件打下基础。

第 4 章组合逻辑电路。介绍了组合逻辑电路的特点、组合逻辑电路的分析方法和设计方法,以及加法器、编码器、译码器、数据选择器、数据比较器、奇偶校验器等常用组合逻辑电路的电路结构、工作原理和使用方法。通过对组合逻辑电路分析方法的介绍,让读者了解一些常用组合逻辑器件的功能及用途。在组合逻辑电路设计内容中,主要介绍基于 Verilog HDL 的设计方法。最后介绍组合逻辑电路中的竞争-冒险。

第5章触发器。介绍触发器的类型、电路结构和功能的表示方法，并介绍基于 Verilog HDL 的触发器设计，为时序逻辑电路的学习打下基础。

第6章时序逻辑电路。介绍时序逻辑电路的结构及特点、常用集成时序逻辑器件的功能及使用方法、时序逻辑电路的分析方法和基于 Verilog HDL 的时序逻辑电路的设计方法。在时序逻辑电路设计内容中，介绍了数码锁存器、移位寄存器、同步计数器、异步计数器、序列信号发生器、序列信号检测器等电路的设计。

第7章半导体存储器。介绍半导体存储器的结构、工作原理和使用方法，并介绍基于 Verilog HDL 的半导体存储器的设计，最后介绍存储器的应用。

第8章数模与模数转换。介绍 A/D 转换器和 D/A 转换器的原理、电路结构和主要技术指标，并以集成 A/D 转换器和 D/A 转换器为例，介绍 A/D 转换器和 D/A 转换器的应用。

第9章数字系统设计。介绍基于 Verilog HDL 的数字系统电路的设计，包括在电子、信息、通信、计算机、自动控制等技术领域的设计实例。

第10章可编程逻辑器件。介绍可编程逻辑器件的分类、结构和编程方法，为使用这类器件打下基础。

第11章 Verilog HDL 的仿真。介绍 Verilog HDL 的仿真语句、仿真测试软件的设计方法和基于 ModelSim(EDA)工具的仿真的方法。

第12章 Verilog HDL 综合与优化。介绍综合的概念、Verilog HDL 综合工具、数字电路与系统设计的硬件实现方法和设计优化。

书中列举了大量的基于 Verilog HDL 的门电路、触发器、组合逻辑电路、时序逻辑电路、存储器和数字系统设计的实例，供读者参考。每个设计实例都经过了 EDA 工具软件的编译和仿真，确保无误。

全书逻辑电路图尽可能地采用国标 GB4728.12-85(即国标标准 IEC617-12)符号，为了与国际教材和 EDA 软件接轨，也提供了国际 IEEE 标准符号。

本书由桂林电子科技大学江国强编著，对于书中的错误和不足之处，恳请读者指正。

编著者
2007年3月

目 录

前言	
第 1 章 数制与编码	1
1.1 概述	1
1.1.1 模拟电子技术和数字电子技术	1
1.1.2 脉冲信号和数字信号	1
1.1.3 数字电路的特点	2
1.2 数制	2
1.2.1 概述	2
1.2.2 数制之间的转换	3
1.3 编码	5
1.3.1 二-十进制编码	5
1.3.2 字符编码	6
1.4 现代数字系统的设计方法	7
1.4.1 设计准备	7
1.4.2 设计输入	7
1.4.3 设计处理	8
1.4.4 设计校验	8
1.4.5 器件编程	8
1.4.6 器件测试和设计验证	8
本章小结	8
思考题和习题	9
第 2 章 逻辑代数和 Verilog HDL 基础	10
2.1 逻辑代数基本概念	10
2.1.1 逻辑常量和逻辑变量	10
2.1.2 基本逻辑和复合逻辑	10
2.1.3 逻辑函数的表示方法	14
2.1.4 逻辑函数的相等	16
2.2 逻辑代数的运算法则	17
2.2.1 逻辑代数的基本公式	17
2.2.2 逻辑代数的基本定理	18
2.2.3 逻辑代数的常用公式	19
2.2.4 异或运算公式	20
2.3 逻辑函数的表达式	20
2.3.1 逻辑函数常用表达式	20
2.3.2 逻辑函数的标准表达式	21
2.3.3 约束及其表示方法	23
2.4 逻辑函数的公式简化法	24
2.4.1 逻辑函数简化的意义	24
2.4.2 逻辑函数的公式简化法	25
2.5 Verilog HDL 基础	25
2.5.1 Verilog HDL 设计模块的结构	26
2.5.2 Verilog HDL 的词法	27
2.5.3 Verilog HDL 的语句	33
2.5.4 不同抽象级别的 Verilog HDL 模型	39
2.5.5 关于 Verilog 2001	39
本章小结	41
思考题和习题	41
第 3 章 门电路	43
3.1 概述	43
3.2 TTL 集成门	44
3.2.1 TTL 集成与非门	44
3.2.2 TTL 与非门的外部特性	45
3.2.3 TTL 与非门的主要参数	48
3.2.4 TTL 与非门的改进电路	49
3.2.5 TTL 集成电路多余输入端的处理	51
3.2.6 TTL 其他类型的集成电路	51
3.2.7 TTL 电路的系列产品	53
3.3 其他类型的双极型集成电路	54

3.3.1	ECL 电路	54	本章小结	100
3.3.2	I ² L 电路	54	思考题和习题	100
3.4	MOS 集成门	54	第 5 章 触发器	103
3.4.1	MOS 管	55	5.1 概述	103
3.4.2	MOS 管开关的电路结构和 工作原理	56	5.2 基本 RS 触发器	103
3.4.3	MOS 非门	57	5.2.1 由与非门构成的基本 RS 触发器	104
3.4.4	MOS 门	58	5.2.2 由或非门构成的基本 RS 触发器	106
3.4.5	CMOS 门的外部特性	61	5.3 钟控触发器	107
3.5	基于 Verilog HDL 的门电路设计	63	5.3.1 钟控 RS 触发器	107
3.5.1	用 assign 语句建模方法实现 门电路的描述	63	5.3.2 钟控 D 型触发器	109
3.5.2	用门级元件例化建模方式来 描述门电路	64	5.3.3 钟控 JK 触发器	109
本章小结		65	5.3.4 钟控 T 型触发器	111
思考题和习题		65	5.3.5 钟控 T' 触发器	112
第 4 章 组合逻辑电路		67	5.4 集成触发器	112
4.1 概述		67	5.4.1 边沿 JK 触发器	112
4.1.1	组合逻辑电路的结构和 特点	67	5.4.2 维持-阻塞结构集成 触发器	113
4.1.2	组合逻辑电路的分析方法	67	5.5 触发器之间的转换	114
4.1.3	组合逻辑电路的设计方法	68	5.5.1 用 JK 触发器实现其他类型 触发器	114
4.2 若干常用的组合逻辑电路		71	5.5.2 用 D 触发器实现其他类型 触发器的转换	115
4.2.1	算术运算电路	71	5.6 基于 Verilog HDL 的触发器设计	116
4.2.2	编码器	74	5.6.1 基本 RS 触发器的设计	116
4.2.3	译码器	77	5.6.2 D 锁存器的设计	118
4.2.4	数据选择器	81	5.6.3 D 触发器的设计	118
4.2.5	数值比较器	83	5.6.4 JK 触发器的设计	119
4.2.6	奇偶校验器	85	本章小结	120
4.3 基于 Verilog HDL 的组合逻辑 电路设计		87	思考题和习题	120
4.3.1	加法器的设计	87	第 6 章 时序逻辑电路	123
4.3.2	编码器	89	6.1 概述	123
4.3.3	译码器的设计	92	6.1.1 时序逻辑电路功能的描述 方法	123
4.3.4	数据选择器的设计	94	6.1.2 时序逻辑电路的分析 方法	124
4.3.5	数值比较器的设计	95		
4.3.6	奇偶校验器的设计	96		
4.4 组合逻辑电路的竞争-冒险现象		98		

6.1.3	同步时序逻辑电路和异步 时序逻辑电路	126	7.5.1	RAM 设计	170
6.2	寄存器和移位寄存器	126	7.5.2	ROM 的设计	172
6.2.1	数码寄存器	126		本章小结	173
6.2.2	移位寄存器	127		思考题和习题	173
6.2.3	集成移位寄存器	128	第 8 章	数模和模数转换	174
6.3	计数器	130	8.1	概述	174
6.3.1	同步计数器的分析	130	8.2	数模(D/A)转换	175
6.3.2	异步计数器	134	8.2.1	D/A 转换器的结构	175
6.3.3	集成计数器	137	8.2.2	D/A 转换器的主要技术 指标	178
6.4	同步时序逻辑电路的设计	140	8.2.3	集成 D/A 转换器	178
6.4.1	数码寄存器的设计	140	8.3	模数(A/D)转换	180
6.4.2	移位寄存器的设计	142	8.3.1	A/D 转换器的基本原理	180
6.4.3	同步计数器的设计	143	8.3.2	A/D 转换器的类型	182
6.4.4	顺序脉冲发生器的设计	148	8.3.3	A/D 转换器的主要技术 指标	187
6.4.5	序列信号发生器的设计	149	8.3.4	集成 ADC 芯片	188
6.4.6	序列信号检测器的设计	152		本章小结	189
6.4.7	一般同步时序逻辑 电路的设计	153		思考题和习题	190
6.5	异步时序逻辑电路的设计	156	第 9 章	数字系统设计	191
	本章小结	158	9.1	数字系统的设计方法	191
	思考题和习题	159	9.1.1	4 位二进制计数器的 设计	192
第 7 章	半导体存储器	162	9.1.2	设计七段显示 译码器 dec7s	192
7.1	概述	162	9.1.3	计数译码显示系统 电路的设计	194
7.1.1	半导体存储器的结构	162	9.2	系统设计实例	195
7.1.2	半导体存储器的分类	163	9.2.1	8 位频率计的设计	196
7.2	随机存储器	164	9.2.2	交通灯控制电路的设计	200
7.2.1	静态随机存储器	164	9.2.3	数字电压表的设计	202
7.2.2	动态随机存储器	165	9.2.4	信号发生器的设计	206
7.2.3	随机存储器的典型芯片	166		本章小结	210
7.3	只读存储器	167		思考题和习题	210
7.3.1	固定 ROM	167	第 10 章	可编程逻辑器件	212
7.3.2	可编程只读存储器	168	10.1	PLD 的基本原理	212
7.3.3	可擦除可编程只读 存储器	168			
7.3.4	快闪存储器	169			
7.4	半导体存储器的应用	169			
7.5	基于 Verilog HDL 的存储器设计	170			

10.1.1	PLD 的分类	212	11.3.1	组合逻辑电路测试平台 软件的设计	246
10.1.2	阵列型 PLD	215	11.3.2	时序逻辑电路测试平台 软件的设计	248
10.1.3	现场可编程门阵列 FPGA	218	11.3.3	数字系统电路测试平台 软件的设计	249
10.1.4	基于查找表的结构	220	本章小结	250	
10.2	PLD 的设计技术	223	思考题和习题	251	
10.2.1	PLD 的设计方法	223	第 12 章 Verilog HDL 综合与优化	252	
10.2.2	PLD 的设计流程	224	12.1	综合的概念	252
10.2.3	在系统可编程技术	227	12.2	Verilog HDL 设计的硬件实现	253
10.2.4	边界扫描技术	229	12.2.1	编辑设计文件	253
10.3	PLD 的编程与配置	230	12.2.2	编译设计文件	256
10.3.1	CPLD 的 ISP 方式编程	230	12.2.3	仿真设计文件	257
10.3.2	使用 PC 的并口配置 FPGA	231	12.2.4	编程下载设计文件	262
	本章小结	233	12.3	设计优化	265
	思考题和习题	233	12.3.1	面积与速度的优化	265
第 11 章	Verilog HDL 仿真	234	12.3.2	时序约束与选项设置	266
11.1	Verilog HDL 仿真支持语句	234	12.3.3	Fitter 设置	267
11.1.1	系统任务和系统函数	234	12.4	Quartus II 的 RTL 阅读器	267
11.1.2	编译指令	237	本章小结	268	
11.2	Verilog HDL 的仿真方法	238	思考题和习题	269	
11.2.1	ModelSim 的命令式 仿真	239	附录	270	
11.2.2	ModelSim 的波形仿真	241	附录 A	国产半导体集成电路型号命名法 (GB3430—82)	270
11.2.3	ModelSim 交互命令方式 仿真	244	附录 B	Altera DE2 开发板使用方法	271
11.2.4	ModelSim 批处理工作 方式	245	参考文献	276	
11.3	Verilog HDL 测试平台软件的 设计	246			

第 1 章 数制与编码

📖 知识要点

- 脉冲信号和数字信号的特点。
- 数制及其转换。
- 二-十进制编码和字符编码。
- 现代数字系统的设计方法。

📖 本章教学安排2学时

1.1 概述

1.1.1 模拟电子技术和数字电子技术

电子技术分为模拟电子技术和数字电子技术。模拟电子技术是分析和处理模拟信号的技术，模拟信号具有在数值上和时间上都是连续的特点，正弦波是模拟信号的典型代表。在模拟电路中，使用的主要器件是晶体管，而且晶体管工作在线性区(即放大区)，构成信号的放大和正弦振荡电路。

数字电子技术是分析和处理数字信号的技术。数字信号具有在数值和时间上都不连续的特点，矩形波是数字信号的典型代表。在数字电路中，使用的主要器件也是晶体管，但晶体管工作在非线性区(即截止区和饱和区)，构成信号的开关电路。

1.1.2 脉冲信号和数字信号

从狭义上讲，脉冲信号是指在短时间内突然作用的信号。从广义上讲，除了正弦波或若干个正弦波合成的信号以外的信号都可以称为脉冲信号(如图 1-1a ~ 1-1f)。由图 1-1 可见，脉冲波形是不连续的，但一般都有周期性。

数字信号是指由高低两种电平构成的矩形波，通常用“1”表示高电平，用“0”表示低电平。把矩形波按周期划分，就可以得到由 0 和 1 构成的符号组合，如图 1-2 所示。图中的符号组合是“110100011”，代表二进制数字，所以把矩形波称为数字信号。数字信号也是一种脉冲信号。

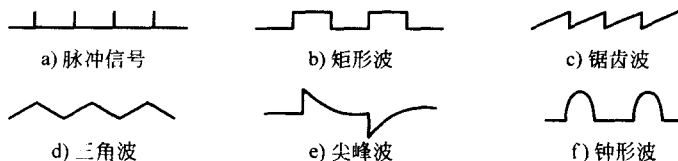


图 1-1 各种脉冲信号波形

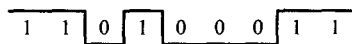


图 1-2 数字信号波形

1.1.3 数字电路的特点

1) 数字电路只有“与”“或”“非”三种基本电路, 这些电路对元件的精度要求不高, 允许有较大的误差, 只要在工作时能可靠地区分高、低两种电平状态就可以了, 因此电路简单, 而且容易实现。

2) 数字电路用 0 和 1 两种状态来表示信息, 便于信息的存储、传输和处理。因此, 许多现代技术都向着数字技术发展, 如数字电话、数字电视等。

3) 数字电路能够对输入的数字信号进行各种算术运算和逻辑运算。所谓逻辑运算, 就是按照人们设计好的规则, 进行逻辑推理和逻辑判断, 得出相应的输出结果。因此, 数字电路具有逻辑思维能力, 它是计算机及智能控制电路的基本电路。

4) 数字电路容易实现集成化, 数字集成电路具有体积小、功耗低、可靠性高等特点。

由于具有这些显著的特点, 数字电路已广泛地应用在计算机、数字通信、智能仪器仪表、自动控制、汽车电子、家用电器和航天航空等领域中。

1.2 数制

在数字电路和计算机中, 只用“0”和“1”两种符号来表示信息, 参与运算的数也是由“0”和“1”构成的, 称为二进制数。考虑到计数习惯, 在计算机操作时, 一般都要把输入的十进制数转换为二进制数后再让计算机处理; 而计算机处理的二进制结果也需要转换为便于人们识别的十进制数后再显示出来。因此, 我们需要学习不同的数制及其转换方法。

1.2.1 概述

用数字量表示物理量的大小时, 仅用一位数码往往不够, 因此经常需要用进位的方法组成多位数码来表示数的量。把多位数码中每一位的构成方法及从低位到高位进位的规则称为数制。常用的数制有十进制、二进制、八进制和十六进制。

1. 十进制

十进制用 0~9 十个符号表示数, 计数的基数是 10(即使用的符号个数)。超过 9 的数必须用多位数表示, 其中低位和相邻高位之间的关系是“逢十进一”或“借一当十”, 故称为十进制。任意一个十进制数 D 均可展开为

$$(D)_{10} = \sum_{i=-m}^{n-1} k_i \times 10^i \quad (1-1)$$

式中, k_i 是第 i 位的系数, 它可以是 0~9 这十个数码中的任何一个。若整数部分的位数是 n , 小数部分的位数是 m , 则 i 包含从 $n-1$ 到 0 的所有正整数和从 -1 到 $-m$ 的所有负整数。 10^i 称为第 i 位的权值(即基数的幂次)。例如, 按权展开十进制数为

$$(125.625)_{10} = 1 \times 10^2 + 2 \times 10^1 + 5 \times 10^0 + 6 \times 10^{-1} + 2 \times 10^{-2} + 5 \times 10^{-3}$$

若以 N 取代式(1-1)中的 10, 即可得到任意进制(N 进制)数展开式的普遍形式为

$$(D)_N = \sum_{i=-m}^{n-1} k_i \times N^i \quad (1-2)$$

2. 二进制

二进制用 0 和 1 两个符号来表示数，计数的基数是 2，权值为 2^i 。低位和相邻高位之间的关系是“逢二进一”或“借一当二”，故称为二进制。

根据式(1-2)的规则，任意一个二进制数 D 均可展开为

$$(D)_2 = \sum_{i=-m}^{n-1} k_i \times 2^i \quad (1-3)$$

并可计算出它所表示的十进制数的大小。例如

$$(1101.101)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} = (13.625)_{10}$$

3. 八进制

八进制是用 0~7 八个符号来表示数，计数的基数是 8，权值为 8^i 。低位和相邻高位之间的关系是“逢八进一”或“借一当八”，故称为八进制。

任意一个八进制数 D 均可展开为

$$(D)_8 = \sum_{i=-m}^{n-1} k_i \times 8^i \quad (1-4)$$

并可计算出它所表示的十进制数的大小。例如

$$(376.65)_8 = 3 \times 8^2 + 7 \times 8^1 + 6 \times 8^0 + 6 \times 8^{-1} + 5 \times 8^{-2} = (254.828125)_{10}$$

4. 十六进制数

十六进制是用 0~9 和 A~F 十六个符号来表示数，计数的基数是 16，权值为 16^i 。低位和相邻高位之间的关系是“逢十六进一”或“借一当十六”，故称为十六进制。

任意一个十六进制数 D 均可展开为

$$(D)_{16} = \sum_{i=-m}^{n-1} k_i \times 16^i \quad (1-5)$$

并可计算出它所表示的十进制数的大小。例如

$$(1FD.6C)_{16} = 1 \times 16^2 + 15 \times 16^1 + 13 \times 16^0 + 6 \times 16^{-1} + 12 \times 16^{-2} = (509.421875)_{10}$$

在数字电路中，为了区别不同数制表示的数，可以用括弧加数制基数下标的方式，但在计算机的编程语言中不能使用这种方式，而是使用加数制后缀或数字前缀的方式来表示不同的数制数。数制表示方式随计算机的编程语言的不同而不同，例如在 Verilog HDL 中，用在数的前面加前缀的方式来区别不同数制数，其中，十进制数的前缀为 D(Decimal)或 d，二进制数的前缀为 B(Binary)或 b，八进制数的前缀为 O(Octonary)或 o，十六进制数的前缀为 H(Hexadecimal)或 h。例如

$$(25)_{10} = 'd25 = 25 \quad (\text{十进制数的前缀可略})$$

$$(1101.101)_2 = 'b1101.101$$

$$(76.56)_8 = 'o76.56$$

$$(1FD.6C)_{16} = 'h1FD.6$$

都是不同数制的表示形式。

1.2.2 数制之间的转换

把一种数制数转换为另一种数制数的过程称为数制之间的转换。十进制与二进制数之间的转换是最常用的转换。为了方便表示二进制数，有时也需要在二进制数与八进制或二进制

数与十六进制数之间进行转换。

1. 十进制数到 N 进制数的转换

十进制数的整数和小数部分到 N 进制数的转换方法是不同的，整数部分按除以 N 看余数的方法进行，小数部分按乘以 N 看整数的进位进行。下面以十进制数转换为二进制数为例来讨论这个问题。

假定十进制整数为 $(D)_{10}$ ，等值的二进制数为 $(k_{n-1}k_{n-2}\cdots k_0)_2$ ，依式(1-3)可知

$$\begin{aligned} (D)_{10} &= k_{n-1}2^{n-1} + k_{n-2}2^{n-2} + \cdots + k_12^1 + k_02^0 \\ &= 2(k_{n-1}2^{n-2} + k_{n-2}2^{n-3} + \cdots + k_1) + k_0 \end{aligned} \quad (1-6)$$

式(1-6)表明，若将 $(D)_{10}$ 除以 2，则得到的商为 $k_{n-1}2^{n-2} + k_{n-2}2^{n-3} + \cdots + k_1$ ，而余数即 k_0 ，得到转换后的二进制数的最低位(LSB)。

同理，将式(1-6)中的商除以 2 得到新的商，可写成

$$k_{n-1}2^{n-2} + k_{n-2}2^{n-3} + \cdots + k_1 = 2(k_{n-1}2^{n-3} + k_{n-2}2^{n-4} + \cdots + k_2) + k_1 \quad (1-7)$$

由式(1-7)可看出，若将 $(D)_{10}$ 除以 2 的商再次除以 2，则所得的余数即 k_1 。

依此类推，反复将每次得到的商再除以 2，就可以得到二进制数的每一位了。当 $(D)_{10}$ 被除到 0 时，得到的最后一个余数是 k_{n-1} ，即为转换后的二进制数的最高位(MSB)。

例如，将 $(62)_{10}$ 转换为二进制数可按照如下方法进行：

故 $(62)_{10} = (111110)_2$ 。

其次讨论小数的转换。若 $(D)_{10}$ 是一个十进制数的小数，

对应的二进制数为 $(0k_{-1}k_{-2}\cdots k_{-m})_2$ ，依式(1-3)可知：

$$(D)_{10} = k_{-1}2^{-1} + k_{-2}2^{-2} + \cdots + k_{-m}2^{-m}$$

将上式两边同乘以 2 得到

$$2(D)_{10} = k_{-1} + (k_{-2}2^{-1} + k_{-3}2^{-2} + \cdots + k_{-m}2^{-m+1}) \quad (1-8)$$

式(1-8)说明，将小数 $(D)_{10}$ 乘以 2 所得乘积的整数部分即 k_{-1} ，这是转换后的二进制小数的最高位(MSB)。

同理，将乘积的小数部分再乘以 2 又可得到

$$2(k_{-2}2^{-1} + k_{-3}2^{-2} + \cdots + k_{-m}2^{-m+1}) = k_{-2} + (k_{-3}2^{-1} + k_{-4}2^{-2} + \cdots + k_{-m}2^{-m+2}) \quad (1-9)$$

亦即乘积的整数部分就是 k_{-2} 。

依此类推，将每次乘 2 后所得乘积的小数部分再乘以 2，便可求出二进制小数的每一位。例如，将 $(0.625)_{10}$ 转换为二进制数时可按照如下方法进行：

$$\begin{array}{r} 0.625 \\ \times \quad 2 \\ \hline 1.250 \quad \cdots \cdots \cdots \text{整数部分} = 1 = k_{-1}(\text{MSB}) \\ \times \quad 2 \\ \hline 0.500 \quad \cdots \cdots \cdots \text{整数部分} = 0 = k_{-2} \\ \times \quad 2 \\ \hline 1.000 \quad \cdots \cdots \cdots \text{整数部分} = 1 = k_{-3} \end{array}$$

故 $(0.625)_{10} = (0.101)_2$ 。

请读者注意，按除以 2 看余数的方法，将十进制数的整数部分转换为二进制数时，任何十进制整数经过若干次除以 2 的运算后，最终结果都可以达到 0，因此十进制整数转换成二进制整数的结果是精确的。而十进制小数部分按乘以 2 看向整数的进位方法转换为二进制小数时，若经过若干次乘以 2 的运算后，其小数部分变为 0 时结束转换，则这些十进制小数转换成二进制小数的结果也是精确的。但是，大部分十进制小数（例如 0.66）不断乘以 2 后，其小数部分结果将永远不会为 0，因此这部分十进制小数转换为二进制小数的结果是不精确的。在这种情况下，可以按照转换精度的要求，进行若干次乘以 2 的运算后结束转换。

2. N 进制数转换为十进制数

将 N 进制数按权展开后即可转换为十进制数。例如

$$(1101.011)_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3}$$

$$= 8 + 4 + 0 + 1 + 0.0 + 0.25 + 0.125 = (13.375)_{10}$$

$$(376.65)_8 = 3 \times 8^2 + 7 \times 8^1 + 6 \times 8^0 + 6 \times 8^{-1} + 5 \times 8^{-2} = (254.828125)_{10}$$

$$(1FD.6C)_{16} = 1 \times 16^2 + 15 \times 16^1 + 13 \times 16^0 + 6 \times 16^{-1} + 12 \times 16^{-2} = (509.421875)_{10}$$

3. 二进制数与八进制数之间的转换

因为 $2^3 = 8$ ，所以 3 位二进制数与 1 位八进制数有直接对应关系，即 3 位二进制数可直接写为 1 位八进制数，而 1 位八进制数也可直接写为 3 位二进制数。例如

$$(11010011.1101101)_2 = (323.664)_8$$

$$(174.536)_8 = (1111100.10101111)_2$$

4. 二进制数与十六进制数的转换

因为 $2^4 = 16$ ，所以 4 位二进制数与 1 位十六进制数有直接对应关系，即 4 位二进制数直接可写为 1 位十六进制数，而 1 位十六进制数也可直接写为 4 位二进制数。例如

$$(11010011.1101101)_2 = (D3.DA)_{16}$$

$$(17C.5F)_{16} = (10111100.01011111)_2$$

十进制数到八进制数（或十六进制数）的转换，可以用整数部分按除 8（或除 16）看余数，小数部分按乘 8（或乘 16）看向整数进位的方法进行，但采用这种转换方法时运算比较繁琐。采用把十进制数先转换为二进制数，再将二进制数写为八进制数或十六进制数的方法则比较简单。例如

$$(62.625)_{10} = (111110.101)_2 = (76.5)_8 = (3E.A)_{16}$$

1.3 编码

在数字电路和计算机中，“0”和“1”两个二进制符号除了可以表示二进制数以外，还可以表示十进制数（符号）、英文字母和一些特殊符号。用二进制符号表示特定信息的过程叫做二进制编码。

1.3.1 二-十进制编码

用 4 位二进制符号表示 1 位十进制数的方法叫做二-十进制编码，也称为 BCD (Binary Coded Decimal) 代码。表 1-1 中列出了几种常见的 BCD 代码，根据编码规则的不同，分为有

权码和无权码两类。

表 1-1 几种常用的 BCD 代码

十进制数 \ 编码种类	8421 码	2421 码	4221 码	5421 码	余 3 码
0	0000	0000	0000	0000	0011
1	0001	0001	0001	0001	0100
2	0010	0010	0010	0010	0101
3	0011	0011	0011	0011	0110
4	0100	0100	1000	0100	0111
5	0101	0101	1001	1000	1000
6	0110	0110	1010	1001	1001
7	0111	0111	1011	1010	1010
8	1000	1110	1110	1011	1011
9	1001	1111	1111	1100	1100
权值	8421	2421	4221	5421	无

1. 有权码

在有权码的编码方式中，每个代码中的“1”都代表一个固定的十进制数值，称为这一位的权值。把每一位的“1”代表的十进制数值加起来，得到的结果就是它所代表的十进制数值。例如在 8421 代码中，从左到右每一位“1”的权值分别为 8，4，2，1，所以这种代码称为 8421 码。此外，还有 2421 码、4221 码和 5421 码等，它们都是有权码。

2. 无权码

在无权码的编码方式中，每个代码中的“1”都不代表固定数值，因此不能按照有权码的方法找到每个代码代表的十进制数值。一般无权码都有一定的编码规则，例如，余 3 码是由每个 8421 码加上 3 后得到的。

1.3.2 字符编码

用若干位二进制符号表示的数字、英文字母、命令及特殊符号叫做字符编码，常用的字符编码是美国国家信息交换标准码，简称 ASCII (American Standard Code for Information Interchange) 码。ASCII 码用 7 位二进制符号 $a_7a_6a_5a_4a_3a_2a_1$ 来表示字符和命令。

ASCII 编码表见表 1-2，表中列出了各种命令、数字、字母(含大小写)和一些特殊符号的 ASCII 编码。例如，数字字符'0'(字符要用单引号括起来)的 ASCII 码是'b0110000 或'h30；'9'是'b0111001 或'h39；大写字母字符'A'的 ASCII 码是'b1000001 或'h41；小写字母字符'a'的 ASCII 码是'b1100001 或'h61。ASCII 码是目前大部分计算机与外设交换信息的字符编码。例如，键盘将按键的字符用 ASCII 码表示送入计算机，而计算机将处理好的数据也是用 ASCII 码传送到显示器或打印机，因此称为信息交换标准码。

另外，ASCII 码是一组数字组合，因此 ASCII 码有大小之分。例如，'0'的 ASCII 码 ('h30) 小于'1' ('h31)、'A' ('h41) 小于'B' ('h42) 等。在计算机编程中，利用 ASCII 码的大小特征，可以对一些符号组合(例如国家名)进行排序。

表 1-2 ASCII 编码表

$a_7 a_6 a_5 \rightarrow$ $\downarrow a_4 a_3 a_2 a_1$	000	001	010	011	100	101	110	111
0000	NUL	DLE	SP	0	@	P	`	p
0001	SOH	DC1	!	1	A	Q	a	q
0010	STX	DC2	"	2	B	R	b	r
0011	ETX	DC3	#	3	C	S	c	s
0100	EOT	DC4	\$	4	D	T	d	t
0101	ENQ	NAK	%	5	E	U	e	u
0110	ACK	SYN	&	6	F	V	f	v
0111	BEL	ETB	'	7	G	W	g	w
1000	BS	CAN	(8	H	X	h	x
1001	HT	EM)	9	I	Y	i	y
1010	LF	SUB	*	:	J	Z	j	z
1011	VT	ESC	+	;	K	[k	{
1100	FF	FS	,	<	L	\	l	
1101	CR	GS	-	=	M]	m	}
1110	SO	RS	.	>	N	^	n	~
1111	SI	US	/	?	O	-	o	DEL

1.4 现代数字系统的设计方法

现代数字系统的设计是基于电子设计自动化 (Electronic Design Automation, EDA) 的设计, 它与传统设计方法有很大的区别。现代数字系统设计的大部分工作是在 EDA 软件工作平台上进行的, 因此称为 EDA 设计。EDA 设计流程如图 1-3 所示。EDA 设计流程包括设计准备、设计输入、设计处理和器件编程 4 个步骤及相应的功能仿真、时序仿真和器件测试 3 个设计验证过程。

1.4.1 设计准备

设计准备是指设计者在进行设计之前, 依据任务要求, 确定系统所要完成的功能及复杂程度、器件资源的利用和成本等所要做的准备工作, 如进行方案论证、系统设计和器件选择等。

1.4.2 设计输入

设计输入是指将设计的数字系统或电路按照 EDA 开发软件要求的某种形式表示出来, 并输入计算机的过程。设计输入有多种方式, 包括采用硬件描述语言 (如 VHDL 和 Verilog HDL 等) 进行设计的文本输入方式、图形输入方式和波形输入方式, 或者采用文本、图形两者混合

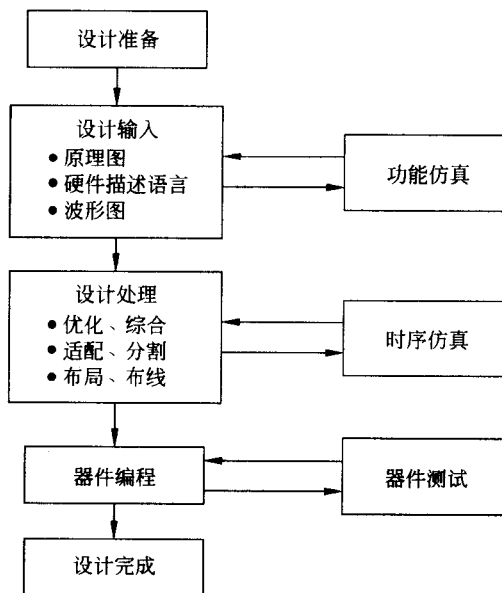


图 1-3 EDA 设计流程

的设计输入方式。

1.4.3 设计处理

设计处理是 EDA 设计中的核心环节。在设计处理阶段,编译软件将对设计输入文件进行逻辑化简、综合和优化,并适当地用一片或多片器件自动地进行适配,最后产生编程用的编程文件。设计处理主要包括设计编译和检查、逻辑优化和综合、适配和分割、布局和布线、生成编程数据文件等过程。

1.4.4 设计校验

设计校验过程包括功能仿真和时序仿真,这两项工作是在设计处理过程中与设计处理同时进行的。功能仿真是在设计输入完成之后,选择具体器件进行编译之前进行的逻辑功能验证,因此又称为前仿真。此时的仿真一般没有电路的延时信息,这对于初步的功能检测非常方便。通过仿真可以观察到电路各个节点的信号变化。仿真中若发现错误,则返回设计输入中修改逻辑设计。

时序仿真是在选择了具体器件并完成布局、布线之后进行的时序关系仿真,因此又称为后仿真或延时仿真。由于不同器件的内部延时不一样,不同的布局、布线方案也会给延时造成不同的影响,因此在设计处理以后,对系统和各模块进行时序仿真,分析其时序关系,估计设计的性能及检查和消除竞争-冒险等是非常有必要的。

1.4.5 器件编程

编程是指将设计处理中产生的编程数据文件通过软件放到具体的可编程逻辑器件中去的过

1.4.6 器件测试和设计验证

器件在编程完毕之后,可以用编译时产生的文件对器件进行检验、加密等工作,或采用边界扫描测试技术进行功能测试,测试成功后才完成其设计。

设计验证可以在 EDA 硬件开发平台上进行。EDA 硬件开发平台的核心部件是一片可编程逻辑器件 FPGA 或 CPLD,再附加一些输入输出设备,如按键、数码显示器、指示灯、喇叭等,还提供时序电路需要的时钟脉冲源。将设计电路编程下载到 FPGA 或 CPLD 中后,根据 EDA 硬件开发平台的操作模式要求,进行相应的输入操作,然后检查输出结果,验证设计电路。

本章小结

数字信号是指由高低两种电平构成的矩形波,通常用“1”代表高电平,用“0”代表低电平。数字电路可以对数字信号进行存储、传输和处理,因此数字电路是计算机的基本电路。用“0”和“1”代表的数称为二进制数,它是计算机惟一能识别和处理的数字。为了方便人与计算机的交流,在操作计算机时需要把十进制数转换为二进制数,或者把二进制数转换为十进制数。