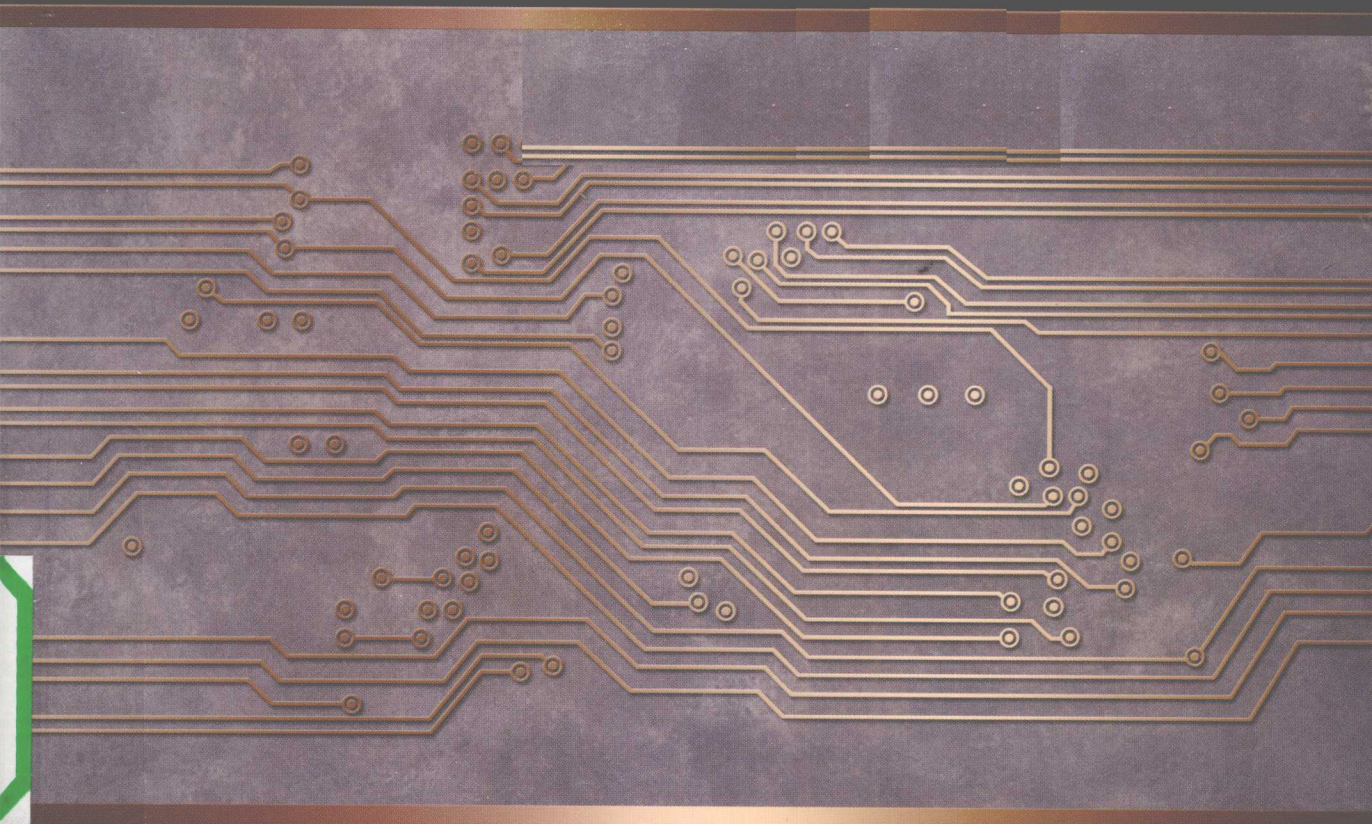


新编电气与电子信息类本科规划教材·电子电气基础课程

# 数字逻辑电路基础

江国强 编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

新编电气与电子信息类本科规划教材·电子电气基础课程

# 数字逻辑电路基础

江国强 编著

電子工業出版社

**Publishing House of Electronics Industry**

北京·BEIJING

## 内 容 简 介

全书共 10 章, 包括数制与编码、逻辑代数、门电路、组合逻辑电路、触发器、时序逻辑电路、脉冲单元电路、数模和模数转换、半导体存储器和可编程逻辑器件, 各章后附有思考题和习题。

本书是结合传统数字设计技术与最新数字设计技术编写的, 书中保留了传统的卡诺图的逻辑化简手段、布尔方程表达式设计方法和相应的中小规模集成电路的堆砌技术等方面内容, 新增了以硬件描述语言 (HDL)、可编程逻辑器件 (PLD) 的现代数字电路设计技术方面的内容。书中列举了大量的基于 HDL 的门电路、触发器、组合逻辑电路、时序逻辑电路、半导体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。

本教材图文并茂、通俗易懂, 并配有电子课件和习题与实验辅导教材, 可作为高等学校工科有关专业的教材和相关工程技术人员的参考书。

未经许可, 不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有, 侵权必究。

### 图书在版编目 (CIP) 数据

数字逻辑电路基础 / 江国强编著. —北京: 电子工业出版社, 2010.5

新编电气与电子信息类本科规划教材·电子电气基础课程

ISBN 978-7-121-10672-9

I. ①数… II. ①江… III. ①数字电路: 逻辑电路—高等学校—教材 IV. ①TN79

中国版本图书馆 CIP 数据核字 (2010) 第 061609 号

责任编辑: 韩同平 特约编辑: 李佩乾

印 刷: 北京季蜂印刷有限公司

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编: 100036

开 本: 787×1092 1/16 印张: 17 字数: 440 千字

印 次: 2010 年 5 月第 1 次印刷

印 数: 4 000 册 定价: 29.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

# 前 言

在 20 世纪 90 年代,国际上电子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,这些器件为数字系统的设计带来极大的灵活性。由于该器件可以通过软件编程而对其硬件结构和工作方式进行重构,使得硬件的设计可以如同软件设计那样方便快捷,极大地改变了传统的数字系统设计方法、设计过程和设计观念。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,使现代电子系统设计领域的电子设计自动化(EDA)技术应运而生。传统的数字电路设计模式,如利用卡诺图的逻辑化简手段、布尔方程表达式设计方法和相应的中小规模集成电路的堆砌技术正在迅速地退出历史舞台。

本书是基于硬件描述语言 HDL (Hardware Description Language) 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,因此本教材以 Verilog HDL 为主作为数字电路与系统的设计工具。

考虑到以卡诺图为逻辑化简手段和相应设计技术这种传统的数字电路设计模式仍然在国内多数高等院校使用,因此本书保留了这部分内容,同时新增了基于 Verilog HDL 现代的数字电路设计技术。读者通过两种设计技术的比对,更能体会现代数字电路设计技术的优越性与高效率性。

本书共 10 章:

第 1 章数制与编码,介绍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。

第 2 章逻辑代数和硬件描述语言基础,介绍分析和设计数字逻辑电路的数学方法。首先介绍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,然后介绍硬件描述语言的基本知识,作为数字逻辑电路的设计基础。

第 3 章门电路,介绍晶体管的开关特性,TTL 集成门电路和 CMOS 集成门电路。对于每一种门电路,除了介绍其电路结构、工作原理和逻辑功能外,还着重讨论它们的电气特性,为实际使用这些器件打下基础,最后介绍基于 Verilog HDL 的门电路设计。

第 4 章组合逻辑电路,介绍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。在组合逻辑电路分析内容方面,以加法器、编码器、译码器、数据选择器、数据比较器、奇偶校验器等电路的分析为例,介绍常用组合逻辑电路的结构、工作原理、逻辑功能、使用方法和主要用途,为读者使用这些器件打下基础。在组合逻辑电路设计内容方面,除了介绍传统的设计方法外,还重点介绍了基于 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, 常用集成时序逻辑部件的功能及使用方法, 时序逻辑电路的分析方法, 传统时序逻辑电路的设计方法和基于 Verilog HDL 的时序逻辑电路的设计方法。

第 7 章脉冲单元电路, 介绍矩形脉冲信号的产生和整形电路。555 定时器是一种多用途的数字/模拟混合集成电路, 本章以 555 定时器为主, 介绍用它构成的多谐振荡器、施密特触发器和单稳态触发器电路, 同时还介绍用其他方式构成的脉冲单元电路。

第 8 章数模与模数转换, 介绍 D/A 转换器和 A/D 转换器的原理、电路结构和主要技术指标, 还介绍了集成 D/A 转换芯片 DAC0832 和集成 A/D 转换芯片 ADC0809 的内部结构、工作原理和使用方法。

第 9 章半导体存储器, 首先介绍半导体存储器的结构与分类, 然后介绍半导体存储器 (RAM 和 ROM) 的工作原理和使用方法, 还介绍了只读存储器 ROM 和可编程逻辑阵列 PLA 在组合逻辑电路设计方面的应用, 最后介绍基于 Verilog HDL 的半导体存储器的设计。

第 10 章可编程逻辑器件, 介绍可编程逻辑器件 (PLD) 的基本原理、电路结构和编程方法。

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采用国标 GB4728.12—85 (即国标标准 IEC617—12), 为了读者习惯, 保留了国际和国内的惯用符号。

本书由桂林电子科技大学江国强教授编著, 如有不足之处, 恳请读者指正。

E-mail: hmjgq@gliet.edu.cn

地 址: 桂林电子科技大学信息科技学院 (541004)

电 话: (0773) 5601095, 13977393225

编著者

# 目 录

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 数制及其转换 .....	(2)
1.3 编码 .....	(5)
1.3.1 二-十进制编码 .....	(5)
1.3.2 字符编码 .....	(6)
本章小结 .....	(7)
思考题和习题 .....	(7)
第 2 章 逻辑代数和硬件描述语言基础 .....	(9)
2.1 逻辑代数基本概念 .....	(9)
2.1.1 逻辑常量和逻辑变量 .....	(9)
2.1.2 基本逻辑和复合逻辑 .....	(9)
2.1.3 逻辑函数的表示方法 .....	(13)
2.1.4 逻辑函数的相等 .....	(15)
2.2 逻辑代数的运算法则 .....	(16)
2.2.1 逻辑代数的基本公式 .....	(16)
2.2.2 逻辑代数的基本定理 .....	(16)
2.2.3 逻辑代数的常用公式 .....	(17)
2.2.4 异或运算公式 .....	(19)
2.3 逻辑函数的表达式 .....	(19)
2.3.1 逻辑函数常用表达式 .....	(19)
2.3.2 逻辑函数的标准表达式 .....	(20)
2.4 逻辑函数的简化法 .....	(22)
2.4.1 逻辑函数简化的意义 .....	(22)
2.4.2 逻辑函数的公式简化法 .....	(23)
2.4.3 逻辑函数的卡诺图简化法 .....	(24)
2.5 Verilog HDL 基础 .....	(28)
2.5.1 Verilog HDL 设计模块的基本结构 .....	(29)
2.5.2 Verilog HDL 的词法 .....	(30)
2.5.3 Verilog HDL 的语句 .....	(36)

2.5.4 不同抽象级别的 Verilog HDL 模型 .....	(41)
本章小结 .....	(42)
思考题和习题 .....	(43)
<b>第3章 门电路</b> .....	<b>(45)</b>
3.1 概述 .....	(45)
3.2 晶体二极管和三极管的开关特性 .....	(46)
3.2.1 晶体二极管的开关特性 .....	(46)
3.2.2 晶体三极管的开关特性 .....	(50)
3.3 分立元件门 .....	(54)
3.3.1 二极管与门 .....	(54)
3.3.2 二极管或门 .....	(55)
3.3.3 三极管非门 .....	(56)
3.3.4 复合逻辑门 .....	(56)
3.3.5 正逻辑和负逻辑 .....	(58)
3.4 TTL 集成门 .....	(58)
3.4.1 TTL 集成与非门 .....	(59)
3.4.2 TTL 与非门的外部特性 .....	(60)
3.4.3 TTL 与非门的主要参数 .....	(64)
3.4.4 TTL 与非门的改进电路 .....	(65)
3.4.5 TTL 其他类型的集成电路 .....	(66)
3.4.6 TTL 集成电路多余输入端的处理 .....	(68)
3.4.7 TTL 电路的系列产品 .....	(69)
3.5 其他类型的双极型集成电路 .....	(69)
3.5.1 ECL 电路 .....	(69)
3.5.2 I <sup>2</sup> L 电路 .....	(70)
3.6 MOS 集成门 .....	(70)
3.6.1 MOS 管 .....	(70)
3.6.2 MOS 反相器 .....	(72)
3.6.3 MOS 门 .....	(74)
3.6.4 CMOS 门的外部特性 .....	(77)
3.7 基于 Verilog HDL 的门电路设计 .....	(78)
3.7.1 用 assign 语句建模方法实现门电路的描述 .....	(79)
3.7.2 用门级元件例化建模方式来描述门电路 .....	(80)
本章小结 .....	(81)
思考题和习题 .....	(81)
<b>第4章 组合逻辑电路</b> .....	<b>(85)</b>
4.1 概述 .....	(85)
4.1.1 组合逻辑电路的结构和特点 .....	(85)

4.1.2	组合逻辑电路的分析方法	(85)
4.1.3	组合逻辑电路的设计方法	(86)
4.2	若干常用的组合逻辑电路	(90)
4.2.1	算术运算电路	(90)
4.2.2	编码器	(92)
4.2.3	译码器	(94)
4.2.4	数据选择器	(98)
4.2.5	数值比较器	(101)
4.2.6	奇偶校验器	(102)
4.3	组合逻辑电路设计	(104)
4.3.1	采用中规模集成部件实现组合逻辑电路	(104)
4.3.2	基于 Verilog HDL 的组合逻辑电路的设计	(109)
4.4	组合逻辑电路的竞争-冒险现象	(119)
	本章小结	(121)
	思考题和习题	(121)
<b>第 5 章</b>	<b>触发器</b>	<b>(125)</b>
5.1	概述	(125)
5.2	基本 RS 触发器	(125)
5.2.1	由与非门构成的基本 RS 触发器	(126)
5.2.2	由或非门构成的基本 RS 触发器	(127)
5.3	钟控触发器	(129)
5.4	集成触发器	(133)
5.4.1	主从 JK 触发器	(133)
5.4.2	边沿 JK 触发器	(135)
5.4.3	维持-阻塞结构集成触发器	(136)
5.5	触发器之间的转换	(137)
5.6	基于 Verilog HDL 的触发器设计	(139)
5.6.1	基本 RS 触发器的设计	(139)
5.6.2	D 锁存器的设计	(140)
5.6.3	D 触发器的设计	(141)
5.6.4	JK 触发器的设计	(142)
	本章小结	(143)
	思考题和习题	(143)
<b>第 6 章</b>	<b>时序逻辑电路</b>	<b>(146)</b>
6.1	概述	(146)
6.2	寄存器和移位寄存器	(149)
6.2.1	寄存器	(149)
6.2.2	移位寄存器	(149)



6.2.3 集成移位寄存器 .....	(151)
6.3 计数器 .....	(153)
6.3.1 同步计数器的分析 .....	(153)
6.3.2 异步计数器的分析 .....	(156)
6.3.3 集成计数器 .....	(160)
6.4 时序逻辑电路的设计 .....	(163)
6.4.1 同步计数器的设计 .....	(164)
6.4.2 异步计数器的设计 .....	(167)
6.4.3 移存型计数器的设计 .....	(170)
6.4.4 一般同步时序逻辑电路的设计 .....	(173)
6.5 基于 Verilog HDL 的时序逻辑电路的设计 .....	(175)
6.5.1 数码寄存器的设计 .....	(175)
6.5.2 移位寄存器的设计 .....	(177)
6.5.3 计数器的设计 .....	(178)
6.5.4 顺序脉冲发生器的设计 .....	(181)
6.5.5 序列信号发生器的设计 .....	(182)
6.5.6 序列信号检测器的设计 .....	(184)
本章小结 .....	(184)
思考题和习题 .....	(185)
<b>第 7 章 脉冲单元电路 .....</b>	<b>(188)</b>
7.1 概述 .....	(188)
7.1.1 脉冲单元电路的分类、结构和波形参数 .....	(188)
7.1.2 脉冲波形参数的分析方法 .....	(189)
7.1.3 555 定时器 .....	(189)
7.2 施密特触发器 .....	(191)
7.2.1 用 555 定时器构成施密特触发器 .....	(191)
7.2.2 集成施密特触发器 .....	(193)
7.3 单稳态触发器 .....	(194)
7.3.1 用 555 定时器构成单稳态触发器 .....	(194)
7.3.2 集成单稳态触发器 .....	(195)
7.4 多谐振荡器 .....	(198)
7.4.1 用 555 定时器构成多谐振荡器 .....	(198)
7.4.2 用门电路构成多谐振荡器 .....	(200)
7.4.3 石英晶体振荡器 .....	(201)
7.4.4 用施密特电路构成多谐振荡器 .....	(201)
本章小结 .....	(202)
思考题和习题 .....	(202)
<b>第 8 章 数模和模数转换 .....</b>	<b>(204)</b>

8.1	概述	(204)
8.2	数模 (D/A) 转换	(205)
8.2.1	D/A 转换器的结构	(205)
8.2.2	D/A 转换器的主要技术指标	(209)
8.2.3	集成 D/A 转换器	(210)
8.3	模数 (A/D) 转换	(211)
8.3.1	A/D 转换器的基本原理	(212)
8.3.2	A/D 转换器的类型	(214)
8.3.3	A/D 转换器的主要技术指标	(218)
8.3.4	集成 A/D 转换器	(219)
	本章小结	(220)
	思考题和习题	(221)
<b>第 9 章</b>	<b>半导体存储器</b>	<b>(222)</b>
9.1	概述	(222)
9.1.1	半导体存储器的结构	(222)
9.1.2	半导体存储器的分类	(223)
9.2	随机存储器	(223)
9.2.1	静态随机存储器 (SRAM)	(223)
9.2.2	动态随机存储器 (DRAM)	(224)
9.2.3	随机存储器的典型芯片	(225)
9.2.4	随机存储器的扩展	(226)
9.3	只读存储器	(228)
9.3.1	固定 ROM	(228)
9.3.2	可编程只读存储器	(229)
9.3.3	可擦除可编程只读存储器	(229)
9.3.4	ROM 的应用	(230)
9.3.5	可编程逻辑阵列 PLA	(231)
9.4	基于 Verilog HDL 的存储器设计	(232)
9.4.1	RAM 设计	(232)
9.4.2	ROM 的设计	(234)
	本章小结	(235)
	思考题和习题	(236)
<b>第 10 章</b>	<b>可编程逻辑器件</b>	<b>(237)</b>
10.1	PLD 的基本原理	(237)
10.1.1	PLD 的分类	(237)
10.1.2	阵列型 PLD	(239)
10.1.3	现场可编程门阵列 (FPGA)	(243)
10.1.4	基于查找表 (LUT) 的结构	(245)

10.2 PLD 的设计技术 .....	(247)
10.2.1 PLD 的设计方法 .....	(248)
10.2.2 PLD 的设计流程 .....	(248)
10.2.3 在系统可编程技术 .....	(251)
10.2.4 边界扫描技术 .....	(253)
10.3 PLD 的编程与配置 .....	(254)
10.3.1 CPLD 的 ISP 方式编程 .....	(255)
10.3.2 使用 PC 的并口配置 FPGA .....	(256)
本章小结 .....	(256)
思考题和习题 .....	(257)
附录 A 国产半导体集成电路型号命名法 (GB3430—82) .....	(258)
参考文献 .....	(260)

# 第 1 章 数制与编码

本章介绍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。

## 1.1 概 述

### 1.1.1 模拟电子技术和数字电子技术

电子技术可以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，而且控制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上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，但控制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，如图 1.1 (a) 所示。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，例如，矩形波（参见图 1.1 (b)）、锯齿波（参见图 1.1 (c)）、三角波（参见图 1.1 (d)）、尖峰波（参见图 1.1 (e)）、钟形波（参见图 1.1 (f)）等。由图 1.1 可见，脉冲波形是不连续的，但一般都有周期性。

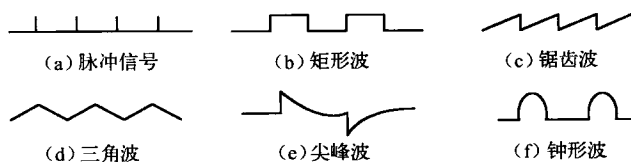


图 1.1 各种脉冲信号波形

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”，它可以代表二进制数字，所以把矩形波称为数字信号。数字信号也是一种脉冲信号，因此一些教材取名为“脉冲与数字电路”。

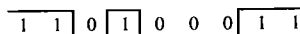


图 1.2 数字信号波形

### 1.1.3 数字电路的特点

数字电路有以下主要特点:

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。

③ 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。

④ 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, 它是计算机以及智能控制电路的基本电路。

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、航天航空等领域中。

## 1.2 数制及其转换

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。考虑到人类计数习惯, 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类识别的十进制数后显示出来。因此, 需要学习不同的数制及其转换方法。

### 1. 数制

用数字量表示物理量的大小时, 仅用一位数码往往不够用, 因此经常需要用进位的方法组成多位数码来记录数的量。把多位数码中每一位的构成方法以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 这 10 个符号来表示数, 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

$$(D)_{10} = \sum_{i=-m}^{n-1} k_i \times 10^i \quad (1.1)$$

其中,  $k_i$  是第  $i$  位的系数, 它可以是 0~9 这 10 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)。任何数制的数都可以按权展开, 例如, 十进制数按权展开的结果如下:

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

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:

$$(D)_N = \sum_{i=-m}^{n-1} k_i \times N^i \quad (1.2)$$

## (2) 二进制

二进制用 0 和 1 这两个符号来表示数，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：

$$(D)_2 = \sum_{i=-m}^{n-1} k_i \times 2^i \quad (1.3)$$

按权展开可以计算出二进制数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 这 8 个符号来表示数的，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：

$$(D)_8 = \sum_{i=-m}^{n-1} k_i \times 8^i \quad (1.4)$$

按权展开可以计算出八进制数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

十六进制是用 0~9 和 A~F 这 16 个符号来表示数的，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：

$$(D)_{16} = \sum_{i=-m}^{n-1} k_i \times 16^i \quad (1.5)$$

按权展开可以计算出十六进制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数制的数。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的数，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

$$(25)_{10} = \text{'d}25 = 25 \quad (\text{十进制数的前缀可略}) \quad (1101.101)_2 = \text{'b}1101.101$$

$$(76.56)_8 = \text{'o}76.56 \quad (1FD.6C)_{16} = \text{'h}1FD.6C$$

都是不同数制的表示形式。

## 2. 数制之间的转换

把一种数制的数转换为另一种数制的数的过程称为数制之间的转换。十进制数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数或二进制数与十六进制数之间转换。

### (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  $N$  看向整数的进位进行。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  $(S)_{10}$ ，等值的二进制数为  $(k_{n-1}k_{n-2}\cdots k_0)_2$ ，依式 (1.3) 可知

$$\begin{aligned}(S)_{10} &= k_{n-1}2^{n-1} + k_{n-2}2^{n-2} + \cdots + k_12^1 + k_02^0 \\ &= 2(k_{n-1}2^{n-2} + k_{n-2}2^{n-3} + \cdots + k_1) + k_0\end{aligned}\quad (1.6)$$

上式表明，若将  $(S)_{10}$  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) 看出，若将  $(S)_{10}$  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  $(S)_{10}$  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：

2	62	..... 余数 =0= $k_0$ (LSB)
2	31	..... 余数 =1= $k_1$
2	15	..... 余数 =1= $k_2$
2	7	..... 余数 =1= $k_3$
2	3	..... 余数 =1= $k_4$
2	1	..... 余数 =1= $k_5$ (MSB)
	0	

故  $(62)_{10} = (111110)_2$ 。

其次讨论小数的转换。若  $(S)_{10}$  是一个十进制数的小数，对应的二进制数为  $(0.k_{-1}k_{-2}\cdots k_{-m})_2$ ，依式 (1.3) 可知

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

将上式两边同乘以 2 得到

$$2(S)_{10} = k_{-1} + (k_{-2}2^{-1} + k_{-3}2^{-2} + \cdots + k_{-m}2^{-m+1})\quad (1.8)$$

上式说明，将小数  $(S)_{10}$  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：

0.625	
× 2	
1.250	..... 整数部分 =1= $k_{-1}$ (MSB)
× 2	
0.500	..... 整数部分 =0= $k_{-2}$
× 2	
1.000	..... 整数部分 =1= $k_{-3}$

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 位二进制数直接可写为 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

$$(11010011.1101101)_2 = (323.664)_8 \quad (174.536)_8 = (1111100.10101111)_2$$

### (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

$$(11010011.1101101)_2 = (D3.DA)_{16} \quad (17C.5F)_{16} = (101111100.01011111)_2$$

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，小数部分按乘以 8（或乘以 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烦琐。一般采用把十进制数首先转换为二进制数后，再将二进制数写为八进制数或十六进制数方法比较简单。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 中列出了几种常用的 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”的权值依次为 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数字、英文字母、命令以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 编码表如表 1.2 所示，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；大写字符‘A’是‘b1000001 或’h41；小写字符‘a’是‘b1100001 或’h61。ASCII 码是目前大部分计算机与外部设备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的，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，字符‘0’的 ASCII 码（‘h30）小于字符‘1’（‘h31）的、字符‘A’的 ASCII 码（‘h41）小于字符‘B’（‘h42）的等。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。