

数字逻辑系统 分析与设计

崔琛 主编

王振宇 解明祥 陈翔 余剑 参编

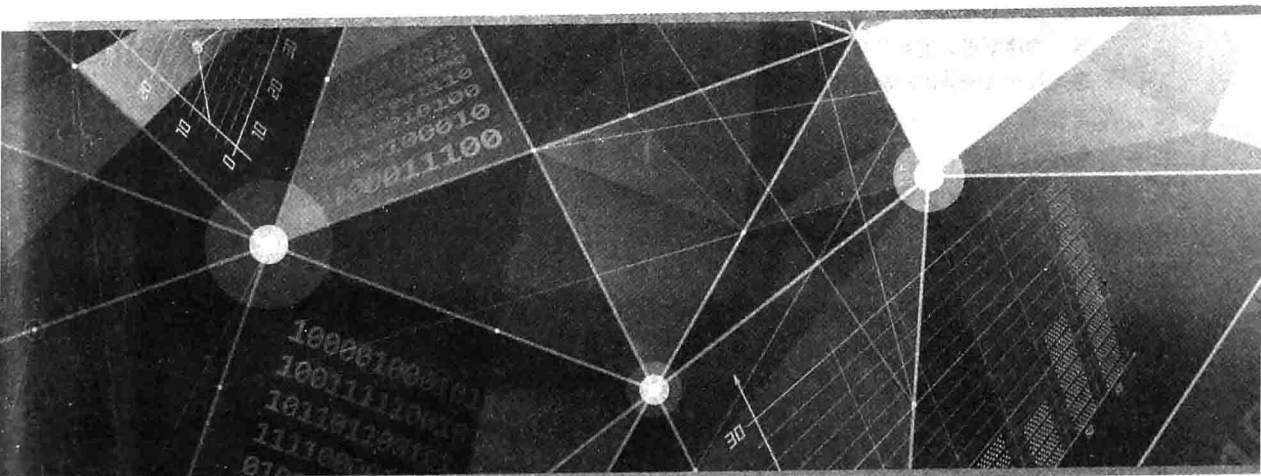
Digital Logic System
Analysis and Design



机械工业出版社
China Machine Press

高等院校电子信息与电气学科系列规划

数字逻辑系统 分析与设计



崔琛 主编
王振宇 解明祥 陈翔 余剑 参编



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

数字逻辑系统分析与设计 / 崔琛主编. —北京: 机械工业出版社, 2015.2
(高等院校电子信息与电气学科系列规划教材)

ISBN 978-7-111-49343-3

I. 数… II. 崔… III. ①数字逻辑—高等学校—教材 ②数字系统—高等学校—教材
IV. TP302.2

中国版本图书馆 CIP 数据核字 (2015) 第 029658 号

本书全面介绍了数字系统的基本理论、常用逻辑电路及基本模块的工作原理、分析方法和设计方法。全书分为 8 章, 内容包括数字逻辑系统基础、逻辑代数与逻辑函数、组合逻辑电路分析与设计、时序逻辑电路分析与设计、半导体存储器、数模与模数转换器、可编程逻辑器件, 以及应用数字系统设计。每章都选用了较多的典型实例, 配有相当数量的习题, 便于读者联系实际, 灵活运用, 提高分析问题、解决问题的能力。

本书贯彻理论与实践相结合的理念, 遵循保证基础、强化应用、便于教学实施的原则, 按照精选内容、循序渐进、前后呼应、突出重点、强化实践的思路进行编写, 可作为高等院校电子信息类、电气信息类及相关专业本科教材和教学参考书, 也可供其他专业师生和相关工程技术人员参考。

出版发行: 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: 张梦玲

责任校对: 董纪丽

印 刷: 北京市荣盛彩色印刷有限公司

版 次: 2015 年 4 月第 1 版第 1 次印刷

开 本: 185mm×260mm 1/16

印 张: 20

书 号: ISBN 978-7-111-49343-3

定 价: 45.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

前 言

本书是数字逻辑系统分析与设计的基础教材，可作为高等院校电子信息类专业开设的“数字逻辑系统”课程的教材和教学参考书。本书在编写时力求教学内容和课程体系符合教学改革与课程整体优化的总要求，使之兼具基础性、系统性、先进性、科学性与可读性，并贯彻理论与实践相结合的理念，遵循保证基础、强化应用、便于教学实施的原则，主要体现在以下几点。

① 注重先进的数字逻辑系统的分析与设计手段，并将这些手段有序地融入本书的内容中。

VHDL 描述和 Quartus II 仿真是现代数字逻辑系统分析与设计的两大重要方法与工具，两者的有机结合代表着现代数字逻辑电路分析与设计的发展方向。本书将两者有机地融合，并注重数字逻辑系统在描述与分析、设计手段上的先进性，同时将 VHDL 与 Quartus II 有机地融入各个章节中。随着内容从器件到系统逐步展开，由浅入深，循序渐进地讲解了现代分析方法和手段的运用。

② 切实贯彻理论分析与实际运用融为一体的先进教学理念。

在本书中，针对每一个实际数字逻辑器件与芯片，除给出逻辑符号与功能表外，每个器件与芯片都配有引脚图，在详细描述其用途和功能后，都给出了来源于实际应用中的实例或例题，并尽量辅以仿真分析，缩小理论分析与实际应用的差距。

③ 在保留经典内容的同时，弱化传统教学内容中过时的以及与其他课程重复的内容。

这主要表现在如下几个方面：

- 对于已淡出使用的独立门电路、基本触发器以及一些小规模集成数字逻辑芯片，本书只将其作为中、大规模集成芯片或可编程器件的基本构成单元来讨论，不再关注其内部组成和工作过程。因此本书不像传统教材将门电路与触发器分别作为一章来介绍，而是将其分别作为组合逻辑与时序逻辑电路中的一节来讨论，并且不再讨论其内部的电压、电流关系。
- 数制与码制及其转换的大部分内容在前期课程中都有讨论与介绍，为了避免重复，本书将其大幅精简，同时将数制、码制及其转换以及基本逻辑门全部压缩到第 1 章中的一个小节中。
- 在现代大规模逻辑电路中，由于可编程阵列与可编程芯片中的资源十分丰富，逻辑函数化简的重要性已经减弱，取而代之的是稳定性与可靠性，因此本书弱化了逻辑函数化简的内容。
- 降低了对手动分析与设计技巧的要求，保留其核心精华内容，强调分析与设计方法的规律性，为充分理解自动化分析与设计奠定了基础。

④ 注重培养学生的综合运用能力。

为培养学生综合利用所学知识的能力，编写了第 8 章——应用数字系统设计。该章以设计“等精度频率计”与“信号发生器”为例，通过方案设计、模块设计、系统实现和系统仿真，使学生能够综合利用所学的组合逻辑电路、时序逻辑电路、可编程芯片、存储器和数模转换等知

识，构建与实现数字逻辑系统。

教学建议

本书适用于“数字逻辑系统”、“数字逻辑电路”和“数字技术”等相关课程，它介绍了数字技术的基本理论和方法、组合逻辑电路分析和设计、时序逻辑电路分析和设计、半导体存储器、数/模及模/数转换、可编程逻辑器件组成和开发(用 VHDL 编写单元模块程序并进行数字系统设计)。

与本书有关课程的教学目的、教学目标以及教学建议等分述如下。

1. 教学目的

- ① 掌握数制、码制、逻辑代数的基本理论和常用公式、逻辑关系常用表达方式、逻辑函数化简；了解 TS 门、OD(OC)门的基本工作原理及典型应用；了解 COMS 和 TTL 器件的基本参数。
- ② 掌握基于门电路的组合逻辑电路的分析与设计，掌握常用型号的中规模组合逻辑电路的组成、工作原理及典型应用；掌握基于触发器的时序逻辑电路的分析与设计；掌握常用型号的时序逻辑电路的组成、工作原理及典型应用；掌握 VHDL 的基本语法、文件框架和用 VHDL 编写常用组合逻辑和时序逻辑模块程序，掌握 Quartus II 开发软件的使用并用其调试 VHDL 程序；了解触发器的结构和工作原理、异步时序逻辑电路的分析与设计。
- ③ 了解半导体存储器的组成及工作原理；了解可编程逻辑器件的组成。
- ④ 掌握常用数/模转换和模/数转换器件的电路组成、工作原理，了解新型器件的电路组成、工作原理，了解等效采样技术的原理和具体实现。
- ⑤ 掌握数字系统的一般设计方法，掌握用 VHDL 设计 DDS 信号源和等精度频率计的方法。

2. 教学目标

以逻辑代数为基础、逻辑电路的分析与设计为主线、培养电子信息类、电气信息类及相近专业本科生在现代数字技术领域中的基本理论、基本分析方法以及实际应用方面的思维方式与研究方法，强调逻辑性、实用性和系统性。希望学生通过该门课程的学习，能深刻地理解数字逻辑的基本概念，掌握基本分析和设计方法，提高分析问题、解决问题的能力，培养科研能力以及创新能力。

3. 学时建议

教学内容	教学要点	课时安排	
		长学时	短学时
第 1 章 数字逻辑系统基础	• 数字逻辑系统简介； • 数制、码制、基本逻辑门*； • 数字逻辑系统分析与设计的基本概念#	4	4
第 2 章 逻辑代数与逻辑函数	• 逻辑代数*； • 逻辑函数及其表示方法*； • 逻辑函数的化简*	8	8
第 3 章 组合逻辑电路分析与设计	• 小规模组合逻辑电路分析与设计*； • 常用组合逻辑功能器件*； • 组合逻辑电路的 VHDL 描述*#； • 大规模组合逻辑电路的计算机辅助设计流程#	10	6

(续)

教学内容	教学要点	课时安排	
		长学时	短学时
第 4 章 时序逻辑电路分析与设计	• 时序逻辑电路的基本简介； • 触发器； • 小规模时序逻辑电路的一般分析方法*； • 小规模时序逻辑电路的一般设计方法*； • 常用时序逻辑功能器件*； • 时序逻辑电路的计算机辅助分析与设计举例*#	12	8
第 5 章 半导体存储器	• 只读存储器*； • 随机存取存储器*； • 存储器的应用*	4	4
第 6 章 数模与模数转换器	• 数模转换器*； • 模数转换器*	4	4
第 7 章 可编程逻辑器件	• 低密度可编程逻辑器件#； • 高密度可编程逻辑器件#； • 硬件测试与编程#	4	0
第 8 章 应用数字系统设计	• 等精度频率计的设计*#； • 信号发生器的设计*#	10	0
教学总学时建议		56	34

4. 说明

① 本教材为“数字逻辑系统”、“数字逻辑电路”和“数字技术”等相关课程的基本教材，授课时分长学时和短学时两种(长学时为 56 学时，短学时为 34 学时)，长学时用于不开“可编程逻辑系统设计”课程的电子信息类、电气信息类及相近本科专业，短学时用于非上述本科专业和开设“可编程逻辑系统设计”课程的上述本科专业，以及专科院校相关专业。

② “数字电路”课程的特点之一就是教学内容涉及大量的设计，由于设计思路不同，即使是采用相同器件进行设计，设计结果也有较大差异。因此，在课堂教学中，应尽可能使用仿真软件进行功能性验证，增加学生的感性认识，进一步深化学生对理论知识的理解。

③ 本教材不包含实验方面的内容，“数字系统设计”课程必须安排有关实验，建议与长学时课程配套的实验学时数为 20 学时(以设计性实验为主)，而短学时课程可安排实验学时数为 10 学时(以验证性实验为主)。

④ 表中“*”表示重点教学内容，“#”表示短学时课程不讲授的内容。

本书第 1 章由余剑编写，第 2 章由陈翔编写，第 3、4 章由崔琛编写，第 5、6 章由解明祥编写，第 7、8 章由王振宇编写。崔琛、王振宇对全书所有的 VHDL 代码进行了仿真验证，同时崔琛还负责全书的内容组织与统稿工作。

本书在编写过程中，参考、引用了众多专家与学者编写的著作与研究成果，编者已在书后的参考文献中尽量全部列出，在这里对他们表示衷心的感谢。

由于编者的能力和水平有限，书中难免有不妥之处和错误，恳请广大师生和读者提出批评和改进意见。

编者

2014 年 11 月

目 录

前言

第 1 章 数字逻辑系统基础	1
1.1 数字逻辑系统简介	1
1.1.1 模拟信号与数字信号	1
1.1.2 模拟系统与数字逻辑系统的区别	1
1.1.3 数字逻辑系统的特点	2
1.2 数制、码制、基本逻辑门	3
1.2.1 数制及其转换	3
1.2.2 常用码制及其特点	7
1.2.3 基本逻辑关系与逻辑门	12
1.3 数字逻辑系统分析与设计的基本概念	19
习题	21
第 2 章 逻辑代数与逻辑函数	23
2.1 逻辑代数	23
2.1.1 逻辑代数的基本公理	23
2.1.2 逻辑代数的基本定律	23
2.1.3 逻辑代数的基本定理	24
2.2 逻辑函数及其表示方法	25
2.2.1 逻辑函数	25
2.2.2 逻辑函数的表示方法	26
2.2.3 逻辑函数的标准形式	29
2.3 逻辑函数的化简	31
2.3.1 公式化简法	32
2.3.2 卡诺图化简法	33
习题	41
第 3 章 组合逻辑电路分析与设计	43
3.1 小规模组合逻辑电路分析与设计	43

3.1.1 组合逻辑电路的定义与描述	43
3.1.2 小规模组合逻辑电路的分析	43
3.1.3 小规模组合逻辑电路的设计	44
3.2 常用组合逻辑功能器件	47
3.2.1 编码器	47
3.2.2 译码器	52
3.2.3 数据选择器与数据分配器	59
3.2.4 数据比较器	62
3.2.5 加法器	63
3.2.6 8421BCD 码与二进制码转换器	68
3.2.7 广义译码器	70
3.3 组合逻辑电路的 VHDL 描述	70
3.3.1 硬件描述语言与 VHDL 概述	70
3.3.2 常用逻辑器件的 VHDL 描述	75
3.4 大规模组合逻辑电路的计算机辅助设计流程	85
3.4.1 计算机辅助设计的一般流程	86
3.4.2 Quartus II 简介	87
3.4.3 设计实例	88
习题	103
第 4 章 时序逻辑电路分析与设计	107
4.1 时序逻辑电路简介	107
4.1.1 时序逻辑电路的基本概念	107
4.1.2 时序逻辑电路的分类	107

4.2 触发器	108	5.3.2 双口 RAM 与先入先出 RAM	185
4.2.1 触发器及其分类	108	5.3.3 动态 RAM	188
4.2.2 触发器的描述方法	108	5.3.4 铁电 RAM	191
4.2.3 RS 触发器	109	5.4 存储器的应用	193
4.2.4 D 触发器	111	5.4.1 字扩展和位扩展	193
4.2.5 JK 触发器	112	5.4.2 实现组合逻辑函数	194
4.2.6 锁存器与触发器的 VHDL 描述	113	习题	197
4.2.7 T(T') 触发器	116	第 6 章 数模与模数转换器	198
4.3 小规模时序逻辑电路的一般 分析方法	117	6.1 概述	198
4.3.1 同步时序逻辑电路的分析 方法	117	6.2 数模转换器	198
4.3.2 异步时序逻辑电路分析 举例	119	6.2.1 DAC 主要性能	198
4.4 小规模时序逻辑电路的一般 设计方法	120	6.2.2 电阻串 DAC	199
4.4.1 设计步骤及设计举例	120	6.2.3 电阻解码 DAC	201
4.4.2 有限状态机的 VHDL 描述	125	6.2.4 恒流源 DAC	203
4.5 常用时序逻辑功能器件	129	6.2.5 电容 DAC	204
4.5.1 计数器	129	6.2.6 其他类型 DAC	206
4.5.2 寄存器与移位寄存器	149	6.2.7 新型集成 DAC 简介	207
4.5.3 顺序脉冲发生器与序列信号 发生器	161	6.3 模数转换器	210
4.6 时序逻辑电路的计算机辅助分析 与设计举例	164	6.3.1 ADC 主要性能	210
4.6.1 时序逻辑电路的计算机 辅助分析举例	165	6.3.2 快闪型 ADC	212
4.6.2 时序逻辑电路的计算机 辅助设计举例	166	6.3.3 反馈比较型 ADC	213
习题	167	6.3.4 双积分型 ADC	215
第 5 章 半导体存储器	173	6.3.5 其他类型 ADC	217
5.1 概述	173	6.3.6 新型集成 ADC 简介	219
5.2 只读存储器	173	6.3.7 等效采样技术	223
5.2.1 组成框图	174	习题	224
5.2.2 掩膜 ROM	175	第 7 章 可编程逻辑器件	226
5.2.3 可编程 ROM	175	7.1 概述	226
5.2.4 电擦除 EPROM	176	7.1.1 PLD 发展历程	226
5.2.5 闪存	180	7.1.2 PLD 分类	226
5.3 随机存取存储器	183	7.1.3 PLD 常用逻辑符号	227
5.3.1 静态 RAM	183	7.2 低密度可编程逻辑器件	227
		7.2.1 PAL 器件	227
		7.2.2 GAL 器件	228
		7.3 高密度可编程逻辑器件	232
		7.3.1 CPLD 器件	232
		7.3.2 FPGA 器件	234
		7.4 硬件测试与编程	244
		7.4.1 硬件测试技术	244
		7.4.2 配置与编程	245

习题	246	8.2.5 系统仿真	275
第 8 章 应用数字系统设计	247	8.3 信号发生器的设计	277
8.1 概述	247	8.3.1 设计任务	277
8.1.1 系统设计	247	8.3.2 方案论证	277
8.1.2 开发环境与设计方法	247	8.3.3 创建设计工程	279
8.1.3 器件选型	247	8.3.4 功能模块设计	280
8.2 等精度频率计的设计	247	8.3.5 系统仿真	296
8.2.1 设计任务	247	8.3.6 优化改进	298
8.2.2 方案论证	248	8.3.7 功能扩展	307
8.2.3 创建设计工程	251	习题	308
8.2.4 功能模块设计	251	参考文献	310

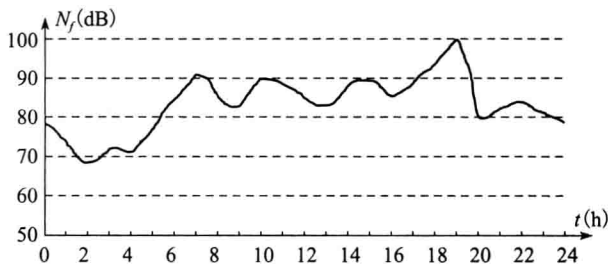
数字逻辑系统基础

1.1 数字逻辑系统简介

1.1.1 模拟信号与数字信号

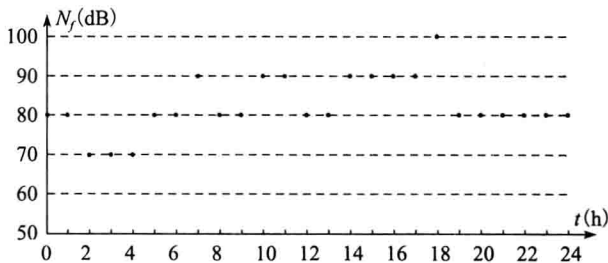
物理量一般可以分为两大类：一类是在时间与强度上都连续的物理量——模拟量；另一类是在时间与强度上都离散的物理量——数字量。

自然界中的大部分物理量都是模拟量，例如温度、压力、电压（或电流）、声音等。图 1-1a 给出了一个声音强度监测点在某天 24 小时内的环境声音强度的变化情况。可以看出它是一条平滑的曲线，如果将该声音强度作为一个信号来看，它就是一个模拟信号。从图 1-1 中可以看出，该监测点的环境声音强度在凌晨 2 点左右达到最小值（车流很小且基本上无过往人群），而在 18—19 时具有很高的峰值（车流很大且过往人群密集）。



a) 某监测点全天的声音强度监测值

当用数字技术来存储、处理该环境声音强度时，往往可能只记录某些特殊时刻的声音强度，比如每天只在整点时刻对测量的声音强度进行记录，且为了存储方便，对测量值进行了分档（例如以 10dB 为 1 档），如果在该时刻的实际声音强度值与分档的分贝值之间有误差，则以误差最小的准则进行归档（这是量化的概念，在本书的第 6 章有详细阐述），那么这时再来观察该信号，就如图 1-1b 所示。可以明晰地看出，此时的环境声音强度在时间与强度上都是离散的，如果将之视为信号，则该信号就是数字信号（其实与真正的数字信号相比，它还缺乏一个编码的过程，这也将第 6 章中阐述）。



b) 整点时刻分档离散的声音强度离散值

图 1-1 环境声音强度监测的模拟信号与整点时刻分档离散的声音强度值

相较于模拟信号，数字信号更易于传输、存储与处理，并具有良好的可再现性。

1.1.2 模拟系统与数字逻辑系统的区别

处理模拟信号的系统称为模拟系统，处理数字信号的系统称为数字逻辑系统（当系统的规模较小时，又称为电路。有时可混用电路与系统这两个概念而不用严格区分）。

随着集成电路与数字处理技术的飞速发展，目前越来越多的系统表现出：除了与物理世界的接口信号处理是由模拟系统完成的外，其余大量对信号的分析与处理都由数字逻辑系统来完成。数字逻辑系统的处理不仅仅包括通过硬件电路的处理，还包括通过软件的处理以及两者的

结合。

模拟系统和数字逻辑系统的主要区别如下。

(1) 处理信号的不同

模拟电路处理的是在时间与幅度上都连续的信号，而数字逻辑电路处理的是在时间上与幅度上都离散的信号。如果数字信号是由模拟信号经变换(模拟—数字转换)得来的信号，则可以认为模拟电路处理的是信号的全部，而数字逻辑电路处理的是模拟信号离散化后的信号。

模拟信号通常可以直接感知，而数字信号往往需要通过特定的变换(数字—模拟转换)才能感知。如以数字方式存储的音乐需要经过相应的数—模转换电路变成模拟信号后，才能被听到。

需要注意的是，采样定理确保了只要信号在时间上离散化的密度足够高，就可以通过数字信号无失真地再现模拟信号。

(2) 分析与设计手段的不同

模拟电路主要关注的是输出与输入的电压、电流、功率、频谱、波形之间的关系。其分析手段涉及线性代数、微积分、常用变换(傅里叶变换、拉普拉斯变换)、电路分析理论等，其设计与后期调试上的自动化程度较低，研发周期较长。

数字逻辑电路主要关注的是输出与输入信号之间的逻辑关系。其分析手段是逻辑代数，也正是逻辑代数的简单性与规律性使数字逻辑电路在设计与后期调试上的自动化程度很高。

(3) 构成电路的元器件工作状态的不同

无论是模拟电路还是数字逻辑电路，其核心构成单元(未必体现为分立的元器件)都是三极管或场效应管。对模拟电路而言，除了非线性变换的特殊需要外，大部分应用电路中的大量核心构成单元都工作在准线性区，元器件需要较大的偏置电压、电流，存在较大的静态功耗，因此其集成度受到了一定的制约。而在数字逻辑电路中，电路中的大量核心构成单元都工作在开关状态，因此静态功耗可以很小，这也是数字逻辑电路集成度高的重要原因之一。由于需要在两个状态之间进行切换，因此数字逻辑电路的工作速度受限于开关状态的切换时间。

尽管模拟电路与数字逻辑电路有很多的不同，但在构成一个完整的系统时，往往两者都是不可或缺的，应相互协调以实现系统的性能指标。合理地划分好两者在系统中承担的任务，充分发挥各自的优势，是系统整体设计中需要均衡考虑的。

1.1.3 数字逻辑系统的特点

数字逻辑系统具有如下显著的特点。

(1) 算术/逻辑运算双功能

数字逻辑系统处理的是二进制信号，除了能进行算术运算外，更为重要的是，二进制信号能方便地进行诸如与、或、非及其组合逻辑运算，因此数字逻辑系统具有算术/逻辑运算双功能，极其适合于运算、比较、存储、传输、控制、决策等用途。

(2) 稳定性好，抗干扰能力强

在数字逻辑系统中，受关注的是信号经过分档处理(量化)与编码后的逻辑0与1，而不是每一点精确的电压值、电流值。在实际电路中，所处理的信号都叠加有噪声，只要噪声产生的影响不超过量化的分辨率，就不会影响逻辑状态，因而数字逻辑系统具有较好的稳定性和抗干扰能力。

至于处理的精度，则可通过增加量化的位数来确保满足要求。

(3) 便于软硬件融合实现

数字逻辑系统的许多功能完全可以交给软件来实现(只要软件处理的时间消耗在可允许的范围内)，并由于能用软件实现使得电路的规模更小、结构更加紧凑、可靠性更高、灵活性更强，因此由微处理器乃至计算机参与构成的数字逻辑系统已是一种普遍的现象。

(4) 分析与设计的可编程性

数字逻辑系统只采用了逻辑代数的分析手段为数字逻辑系统分析与设计的可编程性奠定了基础；硬件描述语言的发展与成熟为数字逻辑系统分析与设计的可编程性铺垫了通路；大规模可编程逻辑器件的面世为数字逻辑系统分析与设计的可编程性提供了硬件平台；分析与设计自动化软件的出现与完善为数字逻辑系统分析与设计的可编程性提供了工具。数字逻辑系统分析与设计的可编程性大大提高了数字逻辑系统设计的效率，缩短了数字逻辑系统的研发周期，进一步促成了数字逻辑系统的蓬勃发展。

1.2 数制、码制、基本逻辑门

1.2.1 数制及其转换

数制是指进位计数的体制。在日常生活中最习惯的是十进制。但在电子世界中很难找到一个存在 10 个稳定状态的器件，而具有两个稳定状态的器件比比皆是(如继电器、二极管、三极管、场效应管等)，因此在数字系统中常采用的是二进制。为了使计数表现形式紧凑以及与存储方式(字节或字)相呼应，还经常采用八进制与十六进制。

数制及其转换所要讨论的就是上述几种数制的计数方式以及之间的转换，如将日常生活中的十进制数转换为可用于机器操作的二进制数，或者是将机器操作结果的二进制数转换为便于理解的十进制数等。

1. 任意进制数的表示方法

任何一个数制必须有两大要素：基数与进位规则。以十进制为例，它的基数为 10，有 0、1、2、3、4、5、6、7、8、9 共 10 个数字，其进位规则为“逢十进一”，而二进制的基数为 2，具有 0、1 两个数字，进位规则为“逢二进一”。

任意进制数的表示通式是：

$$(N)_b = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \cdots + a_0b^0 + a_{-1}b^{-1} + \cdots + a_{-m}b^{-m} = \sum_{i=-m}^{n-1} a_i \times b^i \quad (1-1)$$

式中： b 为数制的基数； n 为该数整数部分的位数； m 为小数部分的位数； a_{n-1} 、 a_{n-2} 、 $\cdots$ 、 a_{-1} 、 $\cdots$ 、 a_{-m} 为每个位置上的系数； b^{n-1} 、 b^{n-2} 、 $\cdots$ 、 b^{-1} 、 $\cdots$ 、 b^{-m} 为每个位置的权重，其对应基数的幂次。

该通式又称为按权展开表达式。

以十进制数 473.85 为例，按照式(1-1)，其可表示为：

$$(473.85)_{10} = 4 \times 10^2 + 7 \times 10^1 + 3 \times 10^0 + 8 \times 10^{-1} + 5 \times 10^{-2}$$

其从左至右的系数依次为 4、7、3、8、5，即百位的系数为 4；十位的系数为 7；个位的系数为 3；第一位小数的系数为 8；第二位小数的系数为 5。

根据上述描述，不难看出，对于十进制数而言，式(1-1)可以转化为：

$$(N)_{10} = a_{n-1}10^{n-1} + a_{n-2}10^{n-2} + \cdots + a_010^0 + a_{-1}10^{-1} + \cdots + a_{-m}10^{-m} = \sum_{i=-m}^{n-1} a_i \times 10^i \quad (1-2)$$

依照上述规则，很容易推断出二进制数按权展开的表达式为：

$$(N)_2 = a_{n-1}2^{n-1} + a_{n-2}2^{n-2} + \cdots + a_02^0 + a_{-1}2^{-1} + \cdots + a_{-m}2^{-m} = \sum_{i=-m}^{n-1} a_i \times 2^i \quad (1-3)$$

按照此规则， $(11010.101)_2$ 可以表示为：

$$(11010.101)_2 = 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$$

八进制数的基数为 8，有 0、1、2、3、4、5、6、7 共 8 个数字，其进位规则为“逢八进一”，八进制数按权展开的表达式为：

$$(N)_8 = a_{n-1}8^{n-1} + a_{n-2}8^{n-2} + \cdots + a_08^0 + a_{-1}8^{-1} + \cdots + a_{-m}8^{-m} = \sum_{i=-m}^{n-1} a_i \times 8^i \quad (1-4)$$

如: $(146.5)_8 = 1 \times 8^2 + 4 \times 8^1 + 6 \times 8^0 + 5 \times 8^{-1}$

十六进制的基数为 16, 有 1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 共 16 个数字, 其中 A~F 分别对应十进制的 10~15。进位规则为“逢十六进一”。十六进制数按权展开的表达式为:

$$(N)_{16} = a_{n-1}16^{n-1} + a_{n-2}16^{n-2} + \cdots + a_016^0 + a_{-1}16^{-1} + \cdots + a_{-m}16^{-m} = \sum_{i=-m}^{n-1} a_i \times 16^i \quad (1-5)$$

如: $(7AE.F)_{16} = 7 \times 16^2 + A \times 16^1 + E \times 16^0 + F \times 16^{-1}$

用十六进制数表示时, 首先, 一个很长的二进制数可以表现得非常紧凑(比十进制还要紧凑), 读者在下一小节的进制转换过程中就能够比较清楚地看到。其次, 它还与一个 16 位存储单元相对应, 因此作为二进制的一种辅助表示方式常在数字系统的软、硬件中使用, 并常用后缀“H”来表示一个十六进制数。如 3FH 的意思就是 $(3F)_{16}$ 。

需要指出的是, 在数字逻辑系统中, 八进制、十六进制是作为二进制的辅助表达形式而存在的, 在实际的系统中, 真正使用的是二进制。

式(1-1)的表达方式是任意进制数按权展开的通式, 它同样可以用来表达三进制、五进制、七进制乃至其他进制数。只是那些进制在数字逻辑系统中没有什么实用意义, 故在本书中不予讨论。

2. 任意进制之间的转换

(1) 任意进制数转换为十进制数

式(1-1)不仅给出了将任意进制数按权展开的通式, 还给出了将任意进制数转换为十进制数的方法——按权展开、计算求和, 即可得到对应的十进制数。

【例 1-1】 分别将 $(11010.101)_2$ 、 $(146.5)_8$ 与 $(7AE.F)_{16}$ 转换为对应的十进制数。

解: 在前面的讨论过程中, 已经分别给出了上述 3 个数的按权展开表达式, 只要对其求和即可得到对应的十进制数。

$$(11010.101)_2 = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^1 + 1 \times 2^{-1} + 1 \times 2^{-3} = (26.625)_{10}$$

在上述的计算过程中, 由于余数为零的项不影响计算结果, 因此可将其直接去除。

$$(146.5)_8 = 1 \times 8^2 + 4 \times 8^1 + 6 \times 8^0 + 5 \times 8^{-1} = (102.625)_{10}$$

$$\begin{aligned} (7AE.F)_{16} &= 7 \times 16^2 + A \times 16^1 + E \times 16^0 + F \times 16^{-1} \\ &= 7 \times 16^2 + 10 \times 16^1 + 14 \times 16^0 + 15 \times 16^{-1} = (1966.9375)_{10} \end{aligned}$$

在计算过程中, 要注意将 A~F 改为对应的 10~15 进行计算。

(2) 十进制数转换为任意进制数

该问题的实质就是 $(M)_{10} = (N)_b$, 其中 b 为除了 10 以外的其他数制的基数。通过前面的转换知, 式(1-1)可以改写为:

$$(N)_b = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \cdots + a_0b^0 + a_{-1}b^{-1} + \cdots + a_{-m}b^{-m} = \sum_{i=-m}^{n-1} a_i \times b^i = (M)_{10} \quad (1-6)$$

由于在转换的过程中, 对整数与小数部分的处理方式不一样, 因此可将问题划分为两部分, 分别为纯整数部分的转换与纯小数部分的转换。整个数的转换结果是将两部分的转换结果合在一起。

① 十进制整数转换为其他进制的整数。

如果仅考虑整数, 则式(1-6)可简化为:

$$(N)_b = a_{n-1}b^{n-1} + a_{n-2}b^{n-2} + \cdots + a_0b^0 = \sum_{i=0}^{n-1} a_i \times b^i = (M)_{10} \quad (1-7)$$

等式两边分别除以基数 b 可得到:

$$(M)_{10}b^{-1} = a_{n-1}b^{n-2} + a_{n-2}b^{n-3} + \cdots + a_1b^0 + a_0b^{-1}$$

可见, 如果要将 $(M)_{10}$ 转换为 b 进制数(假设 $(M)_{10} > b$, 否则转换结果只有 1 位, 且系数就

是 $(M)_{10}$), 则首先要将 $(M)_{10}$ 除以 b , 得到的余数就是 a_0 (注意: a_0 可以为0), 所得的商如果大于 b , 则用这个商再除以 b , 所得余数即为 a_1 , 依次类推, 直至商为0, 得到 a_{n-1} 。

该方法又称为“连除法”, 即不断重复地除基数、取余数, 直至商为零。然后将余数从最后一个余数至第一个余数进行排列, 即得到将十进制整数转换为 b 进制整数的各权值对应的系数: $a_{n-1}, a_{n-2}, \dots, a_0$ 。

下面以 $(100)_{10}$ 转换为二进制数为例, 其转换过程如图 1-2 所示。

即 $(100)_{10} = (1100100)_2$ 。

读者可以依照该过程, 将 $(100)_{10}$ 分别转换为对应的八进制数与十六进制数, 其转换结果为: $(100)_{10} = (144)_8 = (64)_{16}$ 。

② 十进制小数转换为其他进制小数。

如果仅考虑小数部分, 则式(1-6)可简化为

$$(N)_b = a_{-1}b^{-1} + \dots + a_{-m}b^{-m} = \sum_{i=-m}^{-1} a_i \times b^i = (M)_{10} \quad (1-8)$$

如果等式两边分别乘以基数, 那么

$$(M)_{10}b = a_{-1}b^0 + a_{-2}b^{-1} + \dots + a_{-m}b^{-m+1}$$

第一次相乘后的整数部分即系数 a_{-1} , 将 a_{-1} 从整数部分中拿出(其值可以为零), 如果积的剩余部分不为0, 则令其再乘以基数 b , 得到的整数部分则为 a_{-2} , 依次类推, 直至乘积的小数部分为0或位数满足转换精度的要求, 从而得到 a_{-m} 。

该方法又称“连乘法”, 即不断重复地乘基数、取整数位, 直至积为零或位数满足转换精度的要求。然后从第一个整数位排列至最后一个整数位, 即得将十进制小数转换为 b 进制小数的各权值对应的系数 $a_{-1}, a_{-2}, \dots, a_{-m}$ 。

下面以 $(0.625)_{10}$ 转换为二进制数小数为例, 其转换过程如图 1-3 所示。

即 $(0.625)_{10} = (0.101)_2$ 。

按照此方法, 读者可以自行完成 $(0.625)_{10}$ 向八进制与十六进制的转换, 其转换结果为: $(0.625)_{10} = (0.5)_8 = (0.A)_{16}$ 。

读者可能已经感觉到, 如果将 $(0.3)_{10}$ 这样的数转换为二进制数(或八进制数与十六进制数), 无论执行多少次连乘都无法使小数部分为零, 此时规定连乘的次数(也是转换后小数的位数)达到满足转换精度的要求即可。

下面以 $(0.265)_{10}$ 转换为八进制数为例来说明转换过程, 假设转换到小数点后5位, 则满足转换精度的要求, 其过程如图 1-4 所示。

因此可知: $(0.265)_{10} \approx (0.20753)_8$

【例 1-2】 将 $(25.625)_{10}$ 转换为二进制数。

解: 对该数的整数与小数部分分别采用“连除法”与“连乘法”, 然后将两者合并即可得到转换后的数。

在前述过程中已知: $(0.625)_{10} = (0.101)_2$, 因此这里只进行整数部分的变换, 其过程如图 1-5 所示。

即: $(25)_{10} = (11001)_2$ 。再将上述两者合并, 则有:

2	100	余数
2	50	0
2	25	0
2	12	1
2	6	0
2	3	0
2	1	1
	0	1

图 1-2 用“连除法”将十进制数 100 转换为二进制数的过程

$\times 2$	0.625	进位
$\times 2$	0.25	1
$\times 2$	0.5	0
	0	1

图 1-3 用“连乘法”将十进制数 0.625 转换为二进制数的过程

$\times 8$	0.265	进位
$\times 8$	0.12	2
$\times 8$	0.96	0
$\times 8$	0.68	7
$\times 8$	0.44	5
	0.52	3

图 1-4 满足一定转换精度的进制转换过程

2	25	余数
2	12	1
2	6	0
2	3	0
2	1	1
	0	1

图 1-5 将十进制数 25 转换为二进制数的过程

$$(25.625)_{10} = (11001.101)_2$$

【例 1-3】 将 $(1966.9375)_{10}$ 转换为十六进制数。

解：其分为整数与小数部分的处理，处理过程如图 1-6 所示。

因此： $(1966.9375)_{10} = (7AE.F)_{16}$

(3) 二进制、八进制、十六进制数之间的互换

采用上述方法，将十进制作为中间桥梁，实现二进制、八进制、十六进制数之间的互换是完全可行的。比如要将 $(7AE.F)_{16}$ 转换为二进制数，则可以通过上面的方法，先将 $(7AE.F)_{16}$ 转换为十进制的 $(1966.9375)_{10}$ ，然后再通过十进制数转为二进制数的方法，将其转换为 $(11110101110.1111)_2$ 。

然而，由于 2、8、16 均为 2 的整数次幂，即 1 位的八进制数可以用 3 位的二进制数来表示，1 位的十六进制数可用 4 位二进制数来表示，其对应的转换表分别如表 1-1 与表 1-2 所示。



图 1-6 $(1966.9375)_{10}$ 转换为十六进制数的转换过程

表 1-1 八进制/二进制转换表

八进制	二进制	八进制	二进制
0	000	4	100
1	001	5	101
2	010	6	110
3	011	7	111

表 1-2 十六进制/二进制转换表

十六进制	二进制	十六进制	二进制
0	0000	8	1000
1	0001	9	1001
2	0010	A	1010
3	0011	B	1011
4	0100	C	1100
5	0101	D	1101
6	0110	E	1110
7	0111	F	1111

由于存在上述对应关系，只要根据上述对照表则可直接进行转换，无需经过十进制的中间环节，可避免繁琐的计算过程。

① 将八进制、十六进制数转换为二进制数。这种转换比较方便，只要根据表 1-1 或表 1-2 将所要转换的数的每一位数转换为对应的 3 位二进制数(八进制转二进制)或 4 位二进制数(十六进制转二进制)即可。

如将八进制数 $(432.7)_8$ 转换为二进制数的过程为：

$$\begin{array}{cccc} 4 & 3 & 2. & 7 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 100 & 011 & 010. & 111 \end{array}$$

即： $(432.7)_8 = (100011010.111)_2$ 。

再比如将十六进制数 $(4FC2)_{16}$ 转换为二进制数的过程为：

$$\begin{array}{cccc} 4 & F & C & 2 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 0100 & 1111 & 1100 & 0010 \end{array}$$

即 $(4FC2)_{16} = (100111111000010)_2$ 。

注意：由于最高位的0在数值上没有意义，故将其省略。

② 将二进制数转换为八进制、十六进制数。这是上述转换的逆变换。不难理解，这里应是将3位二进制数转换为1位对应的八进制数(二进制转八进制)或4位二进制数转换为1位对应的十六进制数(二进制转十六进制)的过程。唯一需要注意的是：对整数部分的处理是从低位向高位进行分组转换(简称从右向左)的，而对小数部分是从高位向低位分组转换(简称从左向右)的。

$(100011010.111)_2$ 转换为八进制数的过程是：

$$\begin{array}{cccc} 100 & 011 & 010. & 111 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 4 & 3 & 2. & 7 \end{array}$$

即 $(100011010.111)_2 = (432.7)_8$ 。

在转换的过程中，当位数不够的时候，要注意补0(这对整数部分不重要，但对小数部分至关重要，否则会出现转换错误)。

以 $(10011010.11)_2$ 转换为八进制数为例，其转换过程为：

$$\begin{array}{cccc} 010 & 011 & 010. & 110 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 2 & 3 & 2. & 6 \end{array}$$

即 $(10011010.11)_2 = (232.6)_8$ 。如果小数部分不补0，就会转换为3，从而导致转换错误。

再以 $(110100011.001)_2$ 转换为十六进制数为例，其转换过程为：

$$\begin{array}{cccc} 0001 & 1010 & 0011. & 0010 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & A & 3. & 2 \end{array}$$

即 $(110100011.001)_2 = (1A3.2)_{16}$ 。

注意：在转换前进行了补零操作。

最后需要说明的是：八进制数与十六进制数之间的互换可以通过二进制为中间环节进行，这也比用十进制做中间环节要方便得多。这方面的内容留给读者自己进行练习。

1.2.2 常用码制及其特点

任何语言都有自己的编码系统，我们日常所使用的语言也不例外(只是其编码规则十分复杂而已)。概括来说，编码就是用一组特定的符号表示数字、字母或文字。

在数字系统中，只使用0、1两个基本量，其对所处理的常规元素(如数字、字母、标点符号、常用运算符等)的编码是通过一定的编码规则、由多位0、1组合实现的。本节将介绍几种常用的编码方式。

1. 常用BCD码

BCD(Binary Coded Decimal, BCD)码是用4位二进制数表示1位十进制数的编码，因此又称二-十进制编码。BCD码分为有权码与无权码两大类。

(1) 常用有权码及其特点

有权码的特点是将编码按权展开后恰好就是所编码数的十进制值。最常用的3种有权BCD码为8421码、5421码和2421码。表1-3给出了它们的编码表。

表 1-3 常用的三种有权 BCD 码

十进制数字	8421 码	5421 码	2421 码
0	0000	0000	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	1000	1011
6	0110	1001	1100
7	0111	1010	1101
8	1000	1011	1110
9	1001	1100	1111

以十进制数 7 的编码为例说明，其对应的 8421 码、5421 码、2421 码分别为： $(0111)_{8421\text{码}}$ 、 $(1010)_{5421\text{码}}$ 和 $(1101)_{2421\text{码}}$ 。现在按照每一位的权值进行展开，可得：

$$(0111)_{8421\text{码}}=0\times8+1\times4+1\times2+1\times1=(7)_{10}$$

$$(1010)_{5421\text{码}}=1\times5+0\times4+1\times2+0\times1=(7)_{10}$$

$$(1101)_{2421\text{码}}=1\times2+1\times4+0\times2+1\times1=(7)_{10}$$

可见，将码按权展开正好就是所代表数字的十进制值，这是有权码的特点，因此它特别容易记忆。

从表 1-3 中可以看出，无论是采用哪一种编码方式，都用唯一的一个编码表示了数字 0~9。

读者可能会发现，对 5421 码与 2421 码来说，编码方式不是唯一的，而表 1-3 所列的只是它们最常用的两种编码方式。

需要说明的是：用 4 位二进制数进行编码时，可以产生 16 个码，而数字 0~9 只用了其中 10 个码，还有 6 个没有用到的码。在实际工作中，如何处理剩下的 6 个码也是需要特别关注的，在后续章节中会看到这一点。

最后还要强调的是：编码不同于数字，其前面的 0 是有意义的。比如数字 3 的 8421 编码是 0011，不能认为前面的两个零无意义而取消它们。

(2) 常用无权码及其特点

无权码的每一位码值无确定的权值，因此不能用按权展开的方法来计算它所代表的十进制数。

最常用的无权码是余 3 码，其构成非常简单，它是在 8421 码的基础上并对每个码值加 $(3)_{10}=(11)_2$ 得到的，这也是其名为余 3 码的原因。表 1-4 给出了余 3 码与 8421 码的编码对照表。

表 1-4 余 3 码与 8421 码编码对照表

十进制数字	8421 码	余 3 码
0	0000	0011
1	0001	0100
2	0010	0101
3	0011	0110
4	0100	0111
5	0101	1000
6	0110	1001
7	0111	1010
8	1000	1011
9	1001	1100

观察余 3 码的编码表可以发现，其 0 和 9、1 和 8、2 和 7、……恰好具有每个对应位都不相同的(即互为反码)的特点，将具有这种特征的编码称为自反编码。观察表 1-3 可以发现，给出的 2421 码的编码也具有这种自反特性。

2. 格雷码及其特点

在数字系统中，格雷(Gray)码是一种非常有用的码。该码不限于码的位数，也不限于对 0~9 数字的编码。表 1-5 给出了 4 位格雷码与二进制代码之间的对应关系。格雷码有相邻性、循环性与镜像性等特点。