

“十三五”国家重点出版物出版规划项目

卓越工程能力培养与工程教育专业认证系列规划教材（电气工程及其自动化、自动化专业）

数字电路 与逻辑设计

万国春 编著

Digital Circuits and Logic Design



“十三五”国家重点出版物出版规划项目
卓越工程能力培养与工程教育专业认证系列规划教材
(电气工程及其自动化、自动化专业)

数字电路与逻辑设计

万国春 编著

机械工业出版社

本书主要介绍了数字逻辑设计的基本理论和基本概念、数字电路分析方法、逻辑设计方法以及工程实践。全书内容安排由浅入深、循序渐进,理论与实践结合,第1部分讲述数字逻辑设计基础,主要内容包括数字逻辑基础、逻辑代数;第2部分讲述数字电路基础,主要内容包括逻辑门电路、组合逻辑电路、时序逻辑与存储电路、脉冲波形产生与数-模转换电路;第3部分讲述逻辑设计方法,包括硬件描述语言 VHDL、数字逻辑设计基础、数字系统设计与 FPGA。本书注重前后内容的连贯性,注重理论联系实际,紧跟数字电子系统技术的最新发展,强调新技术的使用以及分析问题和解决问题的能力培养。

本书借鉴了国内外经典教材和最新的相关专业文献,内容精练、实例丰富,应用性强,可作为高等学校电子信息类专业、电气信息类专业、计算机类专业的基础课教材,也可供相关专业科技人员参考。

图书在版编目(CIP)数据

数字电路与逻辑设计/万国春编著. —北京:机械工业出版社, 2019. 7
“十三五”国家重点出版物出版规划项目 卓越工程能力培养与工程教育专业认证系列规划教材. 电气工程及其自动化、自动化专业
ISBN 978-7-111-62452-3

I. ①数… II. ①万… III. ①数字电路-逻辑设计-高等学校-教材
IV. ①TN79

中国版本图书馆 CIP 数据核字(2019)第 063759 号

机械工业出版社(北京市百万庄大街 22 号 邮政编码 100037)
策划编辑:路乙达 责任编辑:路乙达 张珂玲 王小东
责任校对:郑 婕 封面设计:鞠 杨
责任印制:郜 敏
北京富生印刷厂印刷
2019 年 9 月第 1 版第 1 次印刷
184mm×260mm·25.5 印张·632 千字
标准书号:ISBN 978-7-111-62452-3
定价:59.80 元

电话服务

客服电话:010-88361066
010-88379833
010-68326294

封底无防伪标均为盗版

网络服务

机工官网:www.cmpbook.com
机工官博:weibo.com/cmp1952
金书网:www.golden-book.com
机工教育服务网:www.cmpedu.com

前 言

现代科学技术发展日新月异，数字技术已经成为新技术发展的一个重要标志。数字电路除了不能用于模拟信号采集、微弱信号放大和高频功率放大等局部领域外，它在电子科学、通信、计算机、自动控制领域的应用越来越广泛，随着数字技术的普及、大数据科学和人工智能的快速发展和应用，数字技术已成为微电子、计算机科学和自动控制领域中不可缺少的专业基础知识和技术。

随着微电子技术的不断进步，数字电路与逻辑系统分析与设计也在快速进步与发展，一般数字系统设计普遍在可编程逻辑器件上进行，设计方法从传统的单纯硬件设计，发展到用计算机软硬件辅助设计，本书的内容注重适应数字电路与系统分析和设计技术的发展，适应新一代电子通信人才培养的需要，同时也注重与本书相关课程内容的前后连贯性，突出数字逻辑电路的基础理论、分析方法和设计方法，加强设计例题分析和应用方面的介绍。

全书共分为3部分9章。第1部分为数字逻辑设计基础，第2部分为数字电路基础，第3部分为逻辑设计方法。具体而言，本书第1章介绍了几种常用的数制和数制转换、二进制算术运算、几种常用编码。第2章分析了逻辑代数中的基本运算和基本定理、逻辑函数及其描述方法、卡诺图与奎恩-麦克拉斯基化简方法、具有无关项的逻辑函数及其化简、多输出逻辑函数的化简与逻辑函数形式的变换。第3章介绍了数字逻辑抽象、半导体开关器件及其门电路分析、CMOS和TTL门电路、NMOS和PMOS晶体管以及未来半导体技术可能的发展方向。第4章介绍了组合逻辑电路的分析方法和基本设计方法、组合逻辑电路模块和时序问题等。第5章介绍了锁存器与触发器的组成和结构，分析了时序逻辑电路和常用的时序逻辑电路模块、时序逻辑电路的设计方法以及“竞争-冒险”、ROM和RAM的结构和工作原理等。第6章介绍了脉冲波形产生电路、D-A转换器、A-D转换电路的结构和工作原理等。第7章介绍了VHDL结构体的描述方式、结构体的子结构形式、顺序语句和并发语句、VHDL中的信号和信号处理等。第8章讲述了组合逻辑电路和时序电路的VHDL设计方法、有限状态机的VHDL设计，分析了多边沿触发问题。第9章介绍了基于FPGA的数字系统设计过程和方法，以及典型的数字系统综合实验。

本书内容全面系统，除介绍数字电路和逻辑设计方法之外，还结合基础项目实践案例，让读者了解数字电路的相关基础理论以及数字逻辑设计方法，帮助读者逐步掌握数字系统设计的基本方法，以满足未来课程基础要求。本书许多逻辑符号采用的国际上流行的符号，因此在书末附上了逻辑符号国外与国标对照表。

本书可作为高等院校电子信息类、电气信息类和计算机类专业教材，也可作为相关专业教师、科研人员及数字集成电路设计工程师的学习参考资料。

本书在写作过程中，得到了同济大学电子科学与技术系教师的许多宝贵的修改意见，对于他们的帮助和支持，编者在此表示衷心的感谢！

特别感谢课题组的刘雯静、陈晨晨、李蒙蒙、唐令怡、邝永康、周健及陈怡等研究生，他们在专业技术资料收集、整理、案例分析和验证等方面做了大量卓有成效的专业技术工作。

由于编者水平有限，书中具体内容可能有疏漏、欠妥和错误之处，恳请各界读者多加指正，以便今后不断改进。

目 录

前言

第 1 部分 数字逻辑设计基础

第 1 章 数字逻辑基础	2	2.2.1 定律	23
1.1 几种常用的数制和数制转换	2	2.2.2 单变量定理	24
1.1.1 数字抽象	2	2.2.3 多变量定理	25
1.1.2 十进制、二进制、八进制和十六进制	3	2.2.4 定理的证明方法	26
1.1.3 字节	8	2.2.5 等式化简	27
1.2 二进制算术运算	9	2.3 逻辑函数及其描述方法	29
1.2.1 二进制加法和有符号的二进制数	9	2.3.1 逻辑函数	29
1.2.2 原码、反码、补码及其运算	11	2.3.2 逻辑函数的两种标准形式	29
1.3 几种常用编码	14	2.3.3 逻辑函数的描述方法	31
1.3.1 几种常见的二—十进制代码	14	2.4 卡诺图与奎恩-麦克拉斯基化简方法	34
1.3.2 格雷码	15	2.4.1 卡诺图化简逻辑	34
1.3.3 ASCII 码	15	2.4.2 奎恩-麦克拉斯基化简方法	40
1.3.4 奇偶校验码	17	2.5 具有无关项的逻辑函数及其化简	42
1.3.5 量子编码	17	2.5.1 约束项、任意项和逻辑函数式中的无关项	42
1.3.6 赫夫曼编码	17	2.5.2 无关项在化简逻辑函数中的应用	44
习题	17	2.6 多输出逻辑函数的化简与逻辑函数形式的变换	44
第 2 章 逻辑代数	19	2.6.1 多输出函数的化简	44
2.1 逻辑代数中的基本运算	19	2.6.2 不同逻辑函数形式的变换	46
2.1.1 “与”“或”“非”及其复合逻辑	19	习题	50
2.1.2 “与或”“或与”表达式	22		
2.2 逻辑代数的基本定理	23		

第 2 部分 数字电路基础

第 3 章 逻辑门电路	54	3.2.1 半导体	58
3.1 数字逻辑抽象	54	3.2.2 半导体二极管	61
3.1.1 电源电压	55	3.2.3 半导体晶体管	63
3.1.2 逻辑电平	55	3.2.4 MOS 场效应晶体管	66
3.1.3 噪声容限	55	3.3 CMOS 门电路	71
3.1.4 电压传输特性与静态约束	56	3.3.1 CMOS 反相器	74
3.2 半导体开关器件及其门电路分析	58	3.3.2 其他 CMOS 逻辑门	76

3.3.3 传输门	82	相互转换	152
3.4 NMOS 和 PMOS 晶体管	84	5.2.8 CD4027 芯片介绍	152
3.5 TTL 门电路	89	5.3 时序逻辑电路分析	154
3.5.1 晶体管反相器	89	5.3.1 同步逻辑电路分析	154
3.5.2 TTL 反相器的电路结构和工作 原理	89	5.3.2 异步逻辑电路分析	158
3.5.3 TTL 反相器特性分析	90	5.4 常用的时序逻辑电路模块分析	161
3.6 未来半导体技术可能的发展方向	92	5.4.1 移位寄存器	161
习题	93	5.4.2 计数器	166
第 4 章 组合逻辑电路	96	5.4.3 顺序脉冲发生器	178
4.1 概述	96	5.4.4 序列信号发生器	180
4.1.1 组合逻辑电路的基本概念	96	5.5 时序逻辑电路的设计方法	181
4.1.2 组合逻辑电路的框图及特点	96	5.5.1 基于集成芯片的任意进制计数 电路的设计方法	181
4.1.3 组合逻辑电路逻辑功能的表示 方法	97	5.5.2 同步时序逻辑电路的设计方法	184
4.2 组合逻辑电路的分析方法	97	5.5.3 异步时序逻辑电路的设计方法	187
4.3 组合逻辑电路的基本设计方法	98	5.5.4 时序逻辑电路的自启动设计	189
4.4 组合逻辑电路模块	103	5.6 时序逻辑电路中的“竞争-冒险”	191
4.4.1 编码器和译码器	103	5.7 ROM 和 RAM	192
4.4.2 数据选择器、数据分配器	117	5.7.1 半导体存储器的一般结构形式	192
4.4.3 数值比较器	119	5.7.2 RAM 的结构、类型和工作 原理	195
4.4.4 加法器	123	5.7.3 存储器容量的扩展	197
4.5 时序	128	5.7.4 用 PROM 实现组合逻辑函数	198
4.5.1 传播延迟和最小延迟	128	习题	199
4.5.2 组合逻辑电路中的“竞争- 冒险”	129	第 6 章 脉冲波形产生与数-模转换 电路	203
4.5.3 毛刺及其处理方法	131	6.1 脉冲波形产生电路	203
4.6 基于 Multisim 的组合逻辑电路设计 简介	132	6.1.1 描述矩形脉冲的主要参数	203
习题	133	6.1.2 施密特触发电路	204
第 5 章 时序逻辑与存储电路	135	6.1.3 集成单稳态触发器	206
5.1 引言	135	6.1.4 多谐振荡电路	207
5.2 锁存器与触发器	136	6.2 D-A 转换器	212
5.2.1 SR 锁存器	136	6.2.1 D-A 转换器的结构和工作原理	212
5.2.2 电平触发的触发器	139	6.2.2 D-A 转换器的转换精度与转换 速度	219
5.2.3 脉冲触发的触发器	142	6.3 A-D 转换器	220
5.2.4 边沿触发的触发器	146	6.3.1 A-D 转换的基本原理	220
5.2.5 触发器的逻辑功能及其描述 方法	148	6.3.2 A-D 转换器的结构和工作原理	222
5.2.6 触发器的动态特性	151	6.3.3 A-D 转换器的转换精度与转换 速度	234
5.2.7 不同逻辑功能触发器之间的 习题	152	习题	235

第3部分 逻辑设计方法

第7章 硬件描述语言 VHDL	241	7.10.1 默认配置	321
7.1 概述	241	7.10.2 元件配置	323
7.2 VHDL 的基础知识	242	7.10.3 块的配置	326
7.2.1 VHDL 程序的结构	242	7.10.4 结构体的配置	328
7.2.2 VHDL 常用资源库中的程序包 ..	249	习题	329
7.2.3 VHDL 的词法单元	254	第8章 数字逻辑设计基础	330
7.2.4 数据对象和类型	257	8.1 组合逻辑电路的 VHDL 设计	330
7.2.5 表达式与运算符	263	8.1.1 加法器	330
7.3 VHDL 结构体的描述方式	265	8.1.2 多路选择器	334
7.3.1 结构体的行为描述	265	8.1.3 编码器与译码器	336
7.3.2 结构体的 RTL 描述	267	8.1.4 设计实践	340
7.3.3 结构体的结构化描述	269	8.2 时序电路的 VHDL 设计	344
7.4 结构体的子结构形式	272	8.2.1 基础时序逻辑模块	345
7.4.1 进程	272	8.2.2 计数器的 VHDL 设计	349
7.4.2 复杂结构体的多进程组织方法 ..	275	8.2.3 堆栈与 FIFO	352
7.4.3 块	276	8.2.4 多边沿触发问题分析	355
7.4.4 子程序	278	8.2.5 设计实践	357
7.5 顺序语句和并发语句	281	8.3 有限状态机的 VHDL 设计	362
7.5.1 顺序语句	281	8.3.1 VHDL 状态机的一般形式	363
7.5.2 并发语句	290	8.3.2 有限状态机的一般设计方法	368
7.6 VHDL 中的信号和信号处理	297	8.3.3 有限状态机的 VHDL 描述	370
7.6.1 信号的驱动源	297	8.3.4 设计实践	374
7.6.2 信号的延迟	298	习题	377
7.6.3 仿真周期和信号的 δ 延迟	300	第9章 数字系统设计与 FPGA	379
7.6.4 信号的属性函数	302	9.1 数字系统设计自动化技术	379
7.6.5 带属性函数的信号	304	9.2 数字系统的设计流程	380
7.7 VHDL 的其他语句	309	9.3 基于 FPGA 的数字系统设计	383
7.7.1 ATTRIBUTE 描述与定义语句	309	9.3.1 可编程逻辑器件的发展历史	383
7.7.2 ASSERT 语句	312	9.3.2 基于 FPGA 的数字系统设计	
7.7.3 TEXTIO	312	流程	384
7.8 多值逻辑	313	9.4 数字系统综合试验	386
7.8.1 三态数值模型	313	9.4.1 直接数字频率合成技术的设计与	
7.8.2 多值逻辑	313	实现	386
7.9 元件例化	316	9.4.2 基于 FPGA 的 FIR 数字滤波器的	
7.9.1 设计通用元件	316	设计	391
7.9.2 构造程序包	319	9.4.3 数字下变频器的设计	395
7.9.3 元件的调用	320	附录	399
7.10 配置	321	参考文献	401

第 1 部分

数字逻辑设计基础

第1章 数字逻辑基础

1.1 几种常用的数制和数制转换

在自然界中，存在着各种各样的物理量，这些物理量可以分为两大类：数字量和模拟量。图 1.1.1 为一种常见的数字和模拟混合系统。该系统中，传感器采集自然界的物理量信号，模拟系统通过模-数转换器把模拟信号转换为数字信号，数字计算机对数字信息进行处理，包括存储、分析与控制。数-模转换器把需要输出的数字量转换为模拟控制信号，通过模拟系统输出给被控对象。

一般来说，在时间上和数量上都是离散的物理量称为数字量，表示数字量的信号叫数字信号，工作在数字信号下的电子电路叫数字电子电路。在时间上或数量上都是连续的物理量称为模拟量，表示模拟量的信号叫模拟信号，工作在模拟信号下的电子电路叫模拟电子电路。从图 1.1.1 系统中可以看出，与模拟信号相比，数字信号具有传输可靠、易于存储、抗干扰能力强、稳定性好等优点。因此，数字电子电路获得了越来越广泛的应用。

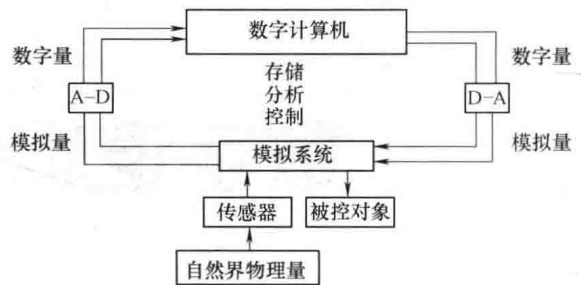


图 1.1.1 一种常见的数字和模拟混合系统

1.1.1 数字抽象

大部分的物理变量值都是连续的，例如正弦交流电压、模拟音频信号、振动的频率、物体位置的数值等都是连续的。而数字电子系统中使用离散变量（Discrete—Variables）来表示信息的变化。一般来说，离散变量是有限数目的不同离散值。

早期 Charles Babbage 的分析机（Analytical Engine）使用了具有 10 个离散变量的数字系统。从 1834 年到 1871 年，Babbage 一直在设计和尝试制作这种机械计算机。分析机使用从 0 号到 9 号 10 个齿轮表示 0~9 共 10 个数字，这很像汽车里的机械里程表。这种分析机中每一行表示一个数字，Babbage 使用了 25 行齿轮，因此这台机器的精度为 25 位数字。

与 Babbage 的机器不同的是，大部分电子计算机使用二进制数制，在正逻辑体制中，数

字电子系统用高电平代表“1”，低电平代表“0”。这是因为半导体器件和电路的发展，很容易用电路中晶体管的导通和截止状态得到高电平和低电平输出。

一个有 N 个不同状态的离散变量的信息量 (Amount of Information) D 由比特 (bit) 衡量, N 和 D 之间的关系是

$$D = \log_2 N \text{ bit} \quad (1-1)$$

一个二进制状态变量包含了 $\log_2 2 = 1\text{bit}$ 的信息, 事实上, bit 是 binary digit 的缩写。每一个 Babbage 的齿轮包含了 $\log_2^{10} = 3.322\text{bit}$ 的信息, 这是因为它能够表示 $2^{3.322} = 10$ 种不同状态的一种。

通常, 在正逻辑体制中, 规定高电平为逻辑“1”、低电平为逻辑“0”; 在负逻辑体制中, 规定低电平为逻辑“1”、高电平为逻辑“0”, 本书中如果未加说明, 则按照正逻辑体制分析与理解。

本书着重讲述使用二进制数码“1”和“0”表示的数字电子系统。英国科学家乔治·布尔 (George Boole) 于 19 世纪中叶提出布尔代数 (Boolean Logic), 又称为逻辑代数, 它是一种针对二进制变量进行逻辑操作的系统, 每个布尔变量都可以取值 TRUE 或 FALSE 中的一种。电子计算机普遍使用正电压代表“1”, 使用零电压代表“0”。本书中使用的“1”与 TRUE 和 HIGH 具有同等的含义。

数字抽象 (Digital Abstraction) 的优势在于设计者可以只关注“0”和“1”, 而忽略布尔变量的物理表示到底是特定电压, 还是旋转的齿轮, 或者是液体的高度。计算机编程人员可以不需要了解计算机硬件的细节就能工作。此外, 对硬件细节的理解使得程序员可以针对特定计算机来优化软件。

1.1.2 十进制、二进制、八进制和十六进制

1. 数制

虽然计算机能极快地进行运算, 但其内部并不像人类在实际生活中使用的是十进制数, 而是使用只包含 0 和 1 两个数值的二进制数。当然, 人们输入计算机的十进制数被转换成二进制数进行计算, 计算后的结果又由二进制数转换成十进制数。数制也称计数制, 是用一组固定的符号和统一的规则来表示数值的方法。人们通常采用的数制有二进制 (Binary)、八进制 (Octal)、十进制 (Decimal) 和十六进制 (Hexadecimal), 当然, 在有些研究工作中, 可以根据特殊应用需要自己定义 N 进制, 如 $N=12$ 、 $N=24$ 、 $N=60$ 、 $N=365$ 等。

数码是数制中表示基本数值大小的不同数字符号。例如, 十进制有 10 个数码: 0、1、2、3、4、5、6、7、8、9。表示数码中每一位的构成及进位的规则称为进位计数制, 简称数制 (Number system)。

一种数制中允许使用的数码个数称为该数制的基数。

数的一般展开式表示法如式 (1-2) 所示。

$$\begin{aligned} D &= (a_{n-1} a_{n-2} \cdots a_1 a_0 a_{-1} a_{-2} \cdots a_{-m})_R = \sum_{i=-m}^{n-1} a_i \times R^i \\ &= a_{n-1} \times R^{n-1} + a_{n-2} \times R^{n-2} + \cdots + a_0 \times R^0 + a_{-1} \times R^{-1} + \cdots + a_{-m} \times R^{-m} \end{aligned} \quad (1-2)$$

式中, n 是整数部分的位数; m 是小数部分的位数; a_i 是第 i 位的系数; R 是基数, R^i 称为第 i 位的权。

(1) 十进制

十进制是人们日常生活中最熟悉的进位计数制。在十进制中, 基数 R 为 10, 它有 0、1、2、3、4、5、6、7、8、9 共 10 个有效数码, 低位向相邻高位“逢十进一, 借一为十”。十进制数一般用下标 10 或 D 表示, 如 $(2018)_{10}$ 、 $(2019)_D$ 。

(2) 二进制

二进制是在计算机系统中采用的进位计数制。基数 R 为 2 的进位计数制称为二进制, 它只有 0 和 1 两个有效数码, 低位向相邻高位“逢二进一, 借一为二”。二进制数一般用下标 2 或 B 表示, 如 $(10101100)_2$ 、 $(11110000)_B$ 等。

(3) 八进制

基数 R 为 8 的进位计数制称为八进制, 它有 0、1、2、3、4、5、6、7 共 8 个有效数码, 低位向相邻高位“逢八进一, 借一为八”。八进制数一般用下标 8 或 O 表示, 如 $(722)_8$ 、 $(713)_O$ 等。

(4) 十六进制

十六进制是人们在计算机指令代码和数据的书写中经常使用的数制。在十六进制中, 基数 R 为 16, 十六进制有 0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F 共 16 个有效数码, 低位向相邻高位“逢十六进一, 借一为十六”。十六进制数一般用下标 16 或 H 表示, 如 $(08EF)_{16}$ 、 $(16DE)_H$ 等。

2. 不同数制间的转换

一个数可以表示为不同进制的形式。在日常生活中, 人们习惯使用十进制数, 而在计算机等设备中则使用二进制数和十六进制数, 因此有时需要在不同数制之间进行转换。

(1) 二—十转换

在求二进制数的等值十进制数时, 将所有值为 1 的数位的位权相加即可。

[例 1.1.1] 将二进制数 $(10010001.11)_2$ 转换为等值的十进制数。

解: 二进制数 $(11001101.11)_2$ 各位对应的位权如下:

位权: 2^7 2^6 2^5 2^4 2^3 2^2 2^1 2^0 2^{-1} 2^{-2}

二进制数 1 0 0 1 0 0 0 1 . 1 1 的等值十进制数为

$$2^7 + 2^4 + 2^0 + 2^{-1} + 2^{-2} = 128 + 16 + 1 + 0.5 + 0.25 = (145.75)_D$$

(2) 十—二转换

十—二转换采用除 2 取余法, 即每次将整数部分除以 2, 余数为该位权上的数, 而商继续除以 2, 余数又为上一个位权上的数, 这个步骤一直持续下去, 直到商为 0 为止, 进行读数时候, 从最后一个余数读起, 一直到最前面的一个余数。将十进制数转换为二进制数时, 要分别对整数部分和小数部分进行转换。

进行整数部分转换时, 先将十进制整数除以 2, 再对每次得到的商除以 2。直至商等于 0 为止。然后将各次余数按倒序写出来, 即第一次的余数为二进制整数的最低有效位 (LSB), 最后一次的余数为二进制整数的最高有效位 (MSB), 所得数值即为等值二进制整数。

[例 1.1.2] 将 $(12)_D$ 转换为二进制数。

解：转换过程如下：

$$\begin{array}{r}
 2 \overline{) 120} \\
 \underline{24} \\
 60 \\
 \underline{48} \\
 120 \\
 \underline{120} \\
 0
 \end{array}$$

因此，对应的二进制整数为 $(1100)_B$ 。

进行小数部分转换时，先将十进制小数乘以2，积的整数作为相应的二进制小数，再对积的小数部分乘以2。如此类推，直至小数部分为0，或按精度要求确定小数值数。第一次积的整数为二进制小数的最高有效位，最后一次积的整数为二进制小数的最低有效位。

[例 1.1.3] 将 $(0.25)_D$ 转换为二进制小数。

解：转换过程如下：

$$\begin{array}{r}
 0.25 \times 2 = 0.5 \\
 \\
 \\
 0.5 \times 2 = 1.0
 \end{array}$$

因此，对应的二进制小数为 $(0.01)_B$ 。

(3) 八—十转换

求八进制数的等值十进制数时，将各数位的值和相应的位权相乘，然后相加。

[例 1.1.4] 将八进制数 $(71.5)_O$ 转换为等值的十进制数。

解：八进制数 $(71.5)_O$ ，各位对应的位权如下：

$$\text{位权：} 8^1 \quad 8^0 \quad 8^{-1}$$

$$\text{八进制数：} 7 \quad 1 \quad 5$$

$$\text{等值十进制数为 } 7 \times 8^1 + 1 \times 8^0 + 5 \times 8^{-1} = 7 \times 8 + 1 \times 1 + 5 \times 0.125 = (57.625)_D$$

(4) 十一—八转换

将十进制数转换为八进制数时，要分别对整数部分和小数部分进行转换。

进行整数部分转换时，先将十进制整数除以8，再对每次得到的商除以8，直至商等于0为止。然后将各次余数按倒序写出来，即第一次的余数为八进制整数的最低有效位，最后一次余数为八进制整数的最高有效位，所得数值即为等值八进制整数。

[例 1.1.5] 将 $(1225)_D$ 转换为八进制数。

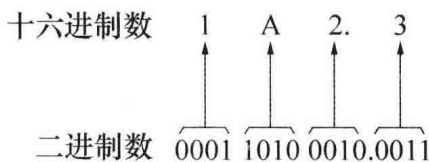
解：转换过程如下：

$$\begin{array}{r}
 8 \overline{) 1225} \\
 \underline{96} \\
 265 \\
 \underline{224} \\
 41 \\
 \underline{32} \\
 9 \\
 \underline{8} \\
 1
 \end{array}$$

将十六进制数转换为二进制数时，将每位十六进制数展开成 4 位二进制数即可。

[例 1.1.13] 将十六进制数 $(1A2.3)_H$ 转换为二进制数。

解：转换过程如下：



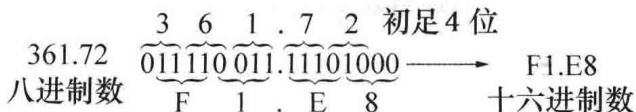
因此，对应的二进制数为 $(110100010.0011)_B$ 。

(11) 八—十六转换

将八进制数转换为十六进制数时，先将八进制数转换为二进制数，再将所得的二进制数转换为十六进制数。

[例 1.1.14] 将八进制数 $(361.72)_O$ 转换为十六进制数。

解：转换过程如下：



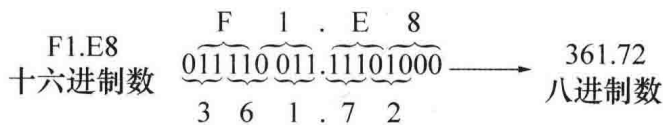
因此，对应的十六进制数为 $(F1.E8)_H$ 。

(12) 十六—八转换

将十六进制数转换为八进制数时，先将十六进制数转换为二进制数，再将所得的二进制数转换为八进制数。

[例 1.1.15] 将十六进制数 $(F1.E8)_H$ 转换为八进制数。

解：转换过程如下：



因此，对应的八进制数为 $(361.72)_O$ 。

1.1.3 字节

字节 (Byte) 是计算机信息技术中用于计量存储容量的一种计量单位，也在一些计算机编程语言中表示数据类型和语言字符。字节是指一小组相邻的二进制数码。通常是 8 位作为一个字节。它是构成信息的一个小单位，作为一个整体来参加操作，比字小，是构成字的单位。

在微型计算机中，通常用字节数来表示存储器的存储容量。在不同的编码方式下，英文字母和中文汉字会占用不同的存储空间。比如，ASCII 码：一个英文字母（不分大小写）占一个字节的空间，一个中文汉字占两个字节的空间。一个二进制数字序列，在计算机中作为一个数字单元，一般为 8 位二进制数，换算为十进制数后，最小值为 0，最大值为 255，如

一个 ASCII 码就是一个字节。UTF-8 编码：一个英文字母等于一个字节，一个中文汉字（含繁体）等于三个字节。Unicode 编码：一个英文字母等于两个字节，一个中文汉字（含繁体）等于两个字节。

数据存储是以“字节”（Byte）为单位，数据传输大多是以“位”（bit，又名“比特”）为单位，1 位就代表一个“0”或“1”（即二进制数），每 8 位（bit，简称为 b）组成一个字节（Byte，简称为 B），是最小一级的信息单位。数字电子系统中存储容量的字节单位和换算如表 1.1.1 所示。

表 1.1.1 数字电子系统中存储容量的字节单位和换算

中文单位	中文简称	英文单位	英文简称	进率 (Byte =)
位	比特	bit	b	0.125
字节	字节	Byte	B	1
千字节	千字节	Kilo Byte	KB	2^{10}
兆字节	兆	Mega Byte	MB	2^{20}
吉字节	吉	Giga Byte	GB	2^{30}
太字节	太	Trillion Byte	TB	2^{40}
拍字节	拍	Peta Byte	PB	2^{50}
艾字节	艾	Exa Byte	EB	2^{60}
泽字节	泽	Zetta Byte	ZB	2^{70}
尧字节	尧	Yotta Byte	YB	2^{80}
千亿亿字节	千亿亿字节	Bront Byte	BB	2^{90}
百万亿亿字节	百万亿亿字节	Nona Byte	NB	2^{100}
十亿亿亿字节	十亿亿亿字节	Dogga Byte	DB	2^{110}
万亿亿亿字节	万亿亿亿字节	Corydon Byte	CB	2^{120}

1.2 二进制算术运算

1.2.1 二进制加法和有符号的二进制数

1. 二进制数的运算

电子计算机具有强大的运算能力，它可以进行两种运算：算术运算和逻辑运算。在数字电子电路中，0 和 1 表示逻辑状态，当用于表示数量的大小时，可以进行算术运算。

(1) 二进制算数运算特点

当两个二进制数码表示两个数量大小时，它们之间可以进行数值运算，这种运算称为算术运算。二进制算术运算和十进制算术运算的规则基本相同，唯一的区别在于二进制数是“逢二进一”而不是十进制数的“逢十进一”。

与十进制数的算术运算相比，二进制数的算术运算特点是：

- 1) 运算的规则类似。
- 2) 进位和借位规则不同（逢二进一，借一当二）

在数字电子系统中实现二进制数的加、减、乘、除运算，算法上，可以用加法器和移位寄存器这两种操作实现，这样做有利于简化数字电子电路。因此，在数字电子电路中普遍采用二进制算术运算。

(2) 二进制数加法

运算规则：0+0=0, 0+1=1, 1+1=10（向高位进一）——逢二进一

[例 1.2.1] 计算二进制数 1010 和 0101 的和。

解：计算过程如下：

$$\begin{array}{r} 1010 \\ + 0101 \\ \hline 1111 \end{array}$$

因此，结果为 1111。

(3) 二进制数减法

运算规则：0-0=0, 1-1=0, 1-0=1, 0-1=11（向高位借一）——借一当二

[例 1.2.2] 计算二进制数 1010 和 0101 的差。

解：计算过程如下：

$$\begin{array}{r} 1010 \\ - 0101 \\ \hline 0101 \end{array}$$

因此，结果为 0101。

(4) 二进制数乘法

由左移被乘数（或零）与加法运算构成。

[例 1.2.3] 计算二进制数 1010 和 0101 的积。

解：计算过程如下：

$$\begin{array}{r} 1010 \\ \times 0101 \\ \hline 1010 \\ 0000 \\ 1010 \\ 0000 \\ \hline 110010 \end{array}$$

因此，结果为 110010。

(5) 二进制数除法

由右移被除数与减法运算构成。

[例 1.2.4] 计算二进制数 1010 和 111 之商。

解：计算过程如下：

$$\begin{array}{r} 1010 \\ 111 \overline{) 1010} \\ \underline{111} \\ 1100 \\ \underline{111} \\ 1010 \\ \underline{111} \\ 11 \end{array}$$