



数字电子技术

冯全源 胡香荣 康萍 等编著

Digital
Electronic Technology

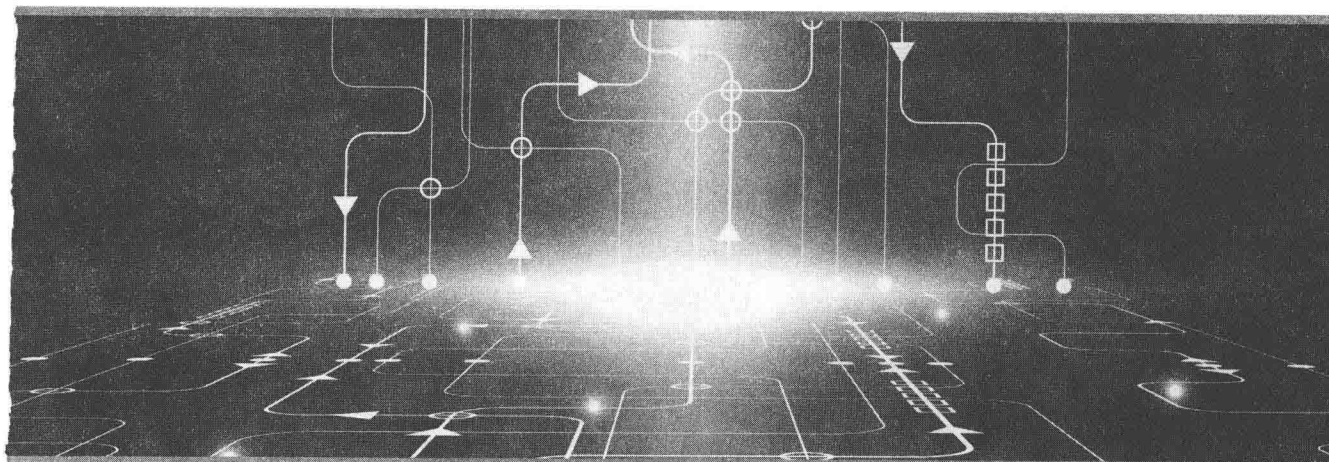


机械工业出版社
China Machine Press

高等院校

规划教材

数字电子技术



冯全源 胡香荣 康萍 等编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

数字电子技术 / 冯全源等编著. —北京: 机械工业出版社, 2015.8
(高等院校电子信息与电气学科系列规划教材)

ISBN 978-7-111-50961-5

I. 数… II. 冯… III. 数字电路—电子技术—高等学校—教材 IV. TN79

中国版本图书馆 CIP 数据核字 (2015) 第 170575 号

本书在编写过程中, 贯彻由浅入深、学以致用原则, 系统地介绍数字电子技术理论的基础知识——数字逻辑基础, 并对半导体晶体管及其开关特性作了一定的说明, 以此引入构成数字电子技术的基本部件——基本逻辑门电路、组合逻辑电路、时序逻辑电路及存储器。本书的主要内容包括组合逻辑电路与时序逻辑电路的分析和设计、可编程逻辑器件设计等。此外, 为了适应当前数字电子技术系统化的需求和趋势, 本书还对数字脉冲波形变换与生成以及数字系统进行了讲解和说明, 并给出了应用实例分析。

本书可作为高等院校电子科学与技术、微电子学与固体电子学、通信工程类、信息工程类、计算机类、网络工程类、测控技术与仪器、应用物理、电气工程及自动化等专业的本科生教材, 也可供其他专业相关教师和学生参考。

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: 秦秀真

责任校对: 董纪丽

印刷: 藁城市京瑞印刷有限公司

版次: 2015 年 8 月第 1 版第 1 次印刷

开本: 185mm×260mm 1/16

印张: 13.5

书号: ISBN 978-7-111-50961-5

定价: 35.00 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: hzsj@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

在过去近一个世纪,以计算机技术、通信技术和微电子技术为代表的电子信息技术得到了迅猛发展。随之造就了今天这样一个繁荣、便捷、多彩的世界,处在这个世界的我们,无时无刻不被电子信息技术的成果所包围,并享受着电子信息技术带来的馈赠。而电子信息技术得以快速发展的一项重要推动技术就是数字化技术。数字化技术使得信息能够大规模并长时间地存储,使得电子设备能够高度集成并具备较强的抗噪声干扰能力,更为重要的是它奠定了计算机技术与各项技术交叉和融合的基础。而本书所讲述的数字电子技术是数字化技术的重要基础。

“数字电子技术”是电子信息类各专业的一门技术基础课,主要研究数字逻辑代数、逻辑器件和电路,以及由各种逻辑电路构成的数字系统。本书依据获评四川省首批精品课程“电子技术基础”的讲义,结合教育部有关专业的大纲要求编写,面向计算机类专业、网络工程类专业等不开设模拟电路或者先开设数字电路的学生,同时也面向微电子专业、电子科学与技术专业等同时开设数字电路和模拟电路或者先开设模拟电路的学生。

本书在编写过程中贯彻由浅入深、学以致用原则,在注重基础和细节的同时,还对数字系统级概念和应用进行了讲述。本书主要内容安排如下。

第1章主要介绍数字逻辑基础,以二进制的概念为讲述起点,重点讲述以二进制为基础的逻辑代数。

第2章主要介绍半导体晶体管及基本逻辑门电路,重点讲述CMOS逻辑和双极性逻辑的特性,并给出了常用逻辑产品系列的规格,使读者建立逻辑电路的基础。

第3章和第4章是本书的重点,主要讲述构成数字电路的两个重要组成部分:组合逻辑电路和时序逻辑电路。其中,重点讲述组合逻辑电路和时序逻辑电路的分析和设计方法,并给出大量设计实例,使读者能够熟练地进行数字电路的设计和分析。

第5章首先讲述数字信息存储装置,即存储器(它也是数字电路的一个重要组成部分),并介绍存储器的常见分类,同时详细说明各类存储器的内部结构和应用方法。其次,该章也对可编程逻辑器件与Verilog HDL语言做了详细讲述。

第6章讲述数字脉冲波形的变换与生成,其中单稳态触发器和多谐振荡器是学习重点。此外,还讲述了常用数字脉冲产生芯片555定时器及其典型应用。

第7章讲述数字系统的构成,重点讨论数字系统的设计和描述方法,并给出一个数字系统的完整设计实例,通过该设计实例讲述以Verilog HDL为代表的描述方法在数字系统设计中的应用。

本书由西南交通大学冯全源教授主编,负责全书的策划、组织和定稿。西南交通大学信息科学与技术学院电子系部分教师参与了本书的编写工作,其中:第1章与第3章由胡香荣老师编写;第2章与第6章由向乾尹老师编写;第4章由康萍老师编写;第5章由张

辉波老师编写；第7章由邸志雄老师编写。

由于本书涉及的内容比较多，很多知识点无法详细展开，读者可以自行对感兴趣的内容进行更深入的阅读和学习。

本书在编写过程中参阅了各章所列出的参考文献，在此对参考文献的原作者表示敬意和感谢！

由于时间仓促和作者水平有限，书中的错误和不妥之处在所难免，希望广大读者批评和指正。

目 录

前言

第 1 章 数字逻辑基础	1
1.1 数字逻辑概念	1
1.1.1 数字电路和数字信号	1
1.1.2 数字电路的发展	2
1.2 数制	2
1.2.1 十进制	2
1.2.2 二进制	2
1.2.3 数制间的转换	3
1.3 码制	6
1.3.1 BCD 码	6
1.3.2 典型格雷码	8
1.3.3 奇偶校验码	8
1.3.4 ASCII 码	9
1.4 基本逻辑运算	9
1.4.1 与运算	9
1.4.2 或运算	10
1.4.3 非运算	11
1.4.4 复合逻辑运算	11
1.5 逻辑表达形式的变换	13
1.6 布尔代数	17
1.6.1 基本定律	17
1.6.2 基本规则	19
1.7 逻辑函数形式变换	20
1.8 逻辑函数化简	22
1.8.1 代数法化简	22
1.8.2 卡诺图化简	23
小结	27
习题	27
参考文献	28

第 2 章 半导体晶体管及基本

逻辑门电路	29
2.1 CMOS 逻辑	29
2.1.1 MOS 管开关特性	29
2.1.2 CMOS 反相器及 CMOS 逻辑基本电气特性	32
2.1.3 其他常用 CMOS 基本 门电路	36
2.2 双极性逻辑	41
2.2.1 二极管开关特性及 二极管逻辑	41
2.2.2 三极管开关特性	43
2.2.3 典型 TTL 门电路	45
2.2.4 其他典型双极性 逻辑类型	47
2.3 常用逻辑产品系列规格	49
小结	50
习题	50
参考文献	52
第 3 章 组合逻辑电路	53
3.1 基本概念	53
3.2 组合电路分析	53
3.3 组合电路设计	54
3.4 组合逻辑模块	55
3.4.1 逻辑门电路	56
3.4.2 74 系列中规模集成 电路	57
3.5 竞争与冒险	68
3.5.1 概念介绍	68
3.5.2 判断方法	69
3.5.3 消除方法	69

小结	69	6.1.1 单稳态触发器基本 原理	115
习题	69	6.1.2 集成单稳态触发器	117
参考文献	71	6.2 多谐振荡器	121
第4章 时序逻辑电路	72	6.2.1 基于CMOS反相器门的 多谐振荡器电路	121
4.1 触发器	72	6.2.2 基于施密特电路的 多谐振荡器电路	122
4.1.1 RS锁存器	72	6.2.3 石英晶体多谐振荡器	123
4.1.2 时钟电平触发的触发器	73	6.3 555定时器及其典型应用	124
4.1.3 时钟边沿触发的触发器	73	6.3.1 555定时器原理	124
4.1.4 其他类型的触发器	74	6.3.2 555定时器的应用	125
4.2 时序逻辑电路的分析与 设计	76	小结	128
4.2.1 时序逻辑电路的特点和 逻辑功能的描述	76	习题	128
4.2.2 同步时序逻辑电路的 分析	76	参考文献	130
4.2.3 异步时序电路的分析	80	第7章 数字系统设计	131
4.2.4 同步时序电路的设计	81	7.1 典型数字系统的构成	131
4.3 常用的时序逻辑集成电路	84	7.2 数据转换器	132
4.3.1 寄存器	84	7.2.1 DAC	132
4.3.2 移位寄存器	85	7.2.2 ADC	136
4.3.3 计数器	86	7.2.3 数据转换器主要技术 指标	141
小结	90	7.2.4 常用集成数据转换 器件	143
习题	90	7.3 数字系统的描述和设计 方法	143
参考文献	93	7.3.1 数字系统的描述	143
第5章 存储器及可编程逻辑器件	94	7.3.2 数字系统的设计方法	144
5.1 半导体存储器	94	7.3.3 数字系统的常用评价 参考指标	145
5.1.1 只读存储器	95	7.4 数字系统设计实例	146
5.1.2 随机存取存储器	98	7.4.1 功能需求	146
5.1.3 存储器容量的扩展	99	7.4.2 设计分析	146
5.2 可编程逻辑器件	101	7.4.3 各模块实现	146
5.2.1 PLD电路的描述方式	102	7.4.4 仿真结果	153
5.2.2 简单可编程器件	102	7.4.5 FPGA实现结果	156
5.2.3 复杂可编程逻辑器件和 现场可编程门阵列	105	小结	157
5.3 Verilog HDL语言简介	107	习题	157
5.3.1 组合逻辑电路的实例	107	参考文献	161
5.3.2 时序逻辑电路的实例	110	附录 部分代码清单	162
小结	112	习题参考答案	179
习题	112		
参考文献	114		
第6章 数字脉冲波形变换与生成	115		
6.1 单稳态触发器	115		

数字逻辑基础

数字技术是研究数字电路及其应用的一门学科,在计算机、交通灯等电子电路中有着广泛应用。元器件技术的发展及电子技术的自动化使得数字电路渗透到生活的各个领域。因此对工科专业的学生,有必要对数字电路和数字信号进行了解、分析和使用。

1.1 数字逻辑概念

1.1.1 数字电路和数字信号

电子电路分为模拟电路和数字电路。日常生活中,计算机、数码相机、交通灯和智能仪器等都包含数字电路。简单地说,数字信号就是时间和数值上离散的信号,是可以用高低电平表示的信号(如图 1.1.1 所示),处理这类信号的电路就是数字电路。对应地,模拟信号就是时间和数值上连续的信号(例如图 1.1.2 所示的正弦波信号、温度、电压、声音等),处理这类信号的电路就是模拟电路。

图 1.1.1 所示的数字波形是理想情况,非理想情况下的数字波形通常存在渡越时间的问题(即上升时间和下降时间,如图 1.1.3 所示)。已知波形的最大幅值,定义从最大幅值 10% 位置处上升到 90% 位置处所用的时间称上升时间 t_r ;从最大幅值 90% 位置处下降到 10% 位置处所用的时间称下降时间 t_f ;上升沿 50% 位置到下降沿 50% 位置,这两点之间的时间差称脉冲宽度 t_w ,周期性数字信号的脉冲宽度和低电平宽度所占的时间称为周期 T 。定义脉冲宽度占整个周期 T 的比例称为占空比 q ,即

$$q = \frac{t_w}{T} \times 100\% \quad (1-1-1)$$

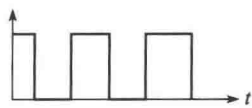


图 1.1.1 数字信号波形

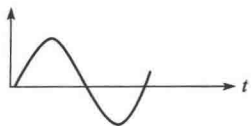


图 1.1.2 模拟信号波形

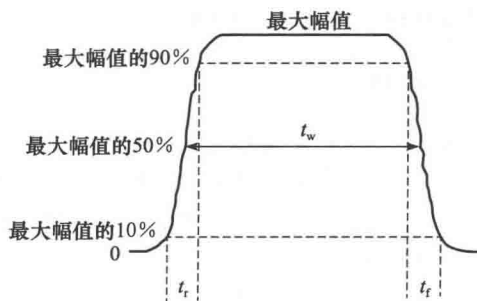


图 1.1.3 数字信号波形与参数

【例 1.1.1】 设周期性数字波形的高电平持续 10ms,低电平持续 10ms,求占空比 q 。

解: 因数字波形的脉冲宽度 $t_w = 10\text{ms}$,周期 $T = 10 + 10 = 20\text{ms}$ 。

$$q = \frac{10}{20} \times 100\% = 50\%$$

占空比为 50% 的数字波形为方波。

1.1.2 数字电路的发展

随着电子电路和电子器件的发展,数字电路经历了电子管电路、晶体管电路和集成电路阶段(如图 1.1.4 所示)。电子电路体积越来越小,功能却越来越强大。

根据集成电路规模的大小(即一片芯片上包含的门或者元器件的个数),通常将其分为小规模集成(Small Scale Integration, SSI)电路、中规模集成(Medium Scale Integration, MSI)电路、大规模集成(Large Scale Integration, LSI)电路及超大规模集成(Very Large Scale Integration, VLSI)电路等。

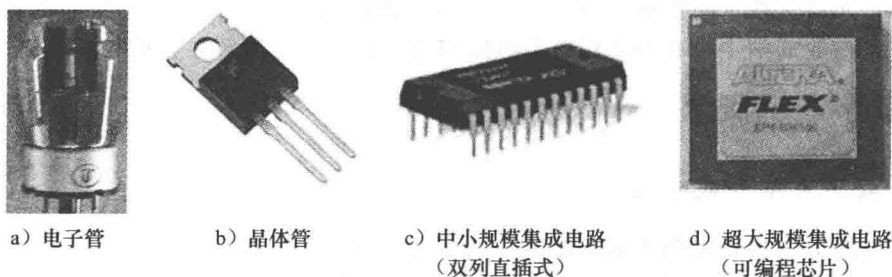


图 1.1.4 电子器件的发展历程

数字集成电路的芯片上标有公司代号、产品类型和产品系列等,例如 Altera 指 Altera 公司, NEC 指日本电气, RCA 指美国无线电等; 74LS 指 TTL 产品, 74HC 指 MOS 产品。一般产品型号的中间数字相同的产品均可替换使用,但如果电路对元器件要求比较严格就要区别对待。

1.2 数制

俗语“半斤八两”指的就是计数数制。日常生活中人们习惯用十进制数,日历中用的是三十或者三十一进制数,计时用到六十进制数。而计算机和数字电路中用二进制数,有时也采用八进制或者十六进制数等。总而言之,数制就是在计数过程中,表示加法运算时的进位或减法运算时的借位规律。 N 进制表示有 N 个数码, N 称为基数; N^i 为对应的位权。下面分别讨论常用的几种数制。

1.2.1 十进制

十进制(Decimal)是指以 10 为基数的计数方式,使用较为广泛。十进制数包括 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 这十个数码,加法运算时“逢十进一”,减法运算时“借一抵十”,即 $9+1=10$, $10-1=9$ 。数码的位置不同代表的实际大小也不同,这和位权 10^i 有关。例如十进制数 11,前一个数码 1 代表 10^1 ,后一个数码 1 代表 10^0 。实际大小就是数码和位权相乘之后再相加,即 $1 \times 10^1 + 1 \times 10^0 = 11$ 。

1.2.2 二进制

二进制(Binary)是指以 2 为基数的计数方式,在计算机技术中经常使用。以电子管为基础的计算机,因为工作状态只有两个,即开和关,所以用十进制数表示不太方便,就采用了包括两个数码 0 和 1 的数制,加法运算时“逢二进一”,减法运算时“借一抵二”,即 $1+1=10$, $10-1=1$ 。数码的位置不同代表的实际大小也不同,这和位权 2^i 有关。例如二进制数

11, 前一个数码 1 代表 2^1 , 后一个数码 1 代表 2^0 。实际大小就是 $1 \times 2^1 + 1 \times 2^0 = 3$, 即二进制数 11 和十进制数 3 对应相等。其他数制也类似, 可以用类似的方式定义和表示。数字电路中常用数制见表 1.2.1。

表 1.2.1 常用数制

进制	十进制(Decimal)	二进制(Binary)	八进制(Octal)	十六进制(Hexadecimal)
数码	0,1,2,3,4,5,6,7,8,9	0,1	0,1,2,3,4,5,6,7	0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
基数	10	2	8	16
位权	10^i	2^i	8^i	16^i
进位规则	逢 10 进 1	逢 2 进 1	逢 8 进 1	逢 16 进 1

不同数制可以对应相互转换, 0~15 以内的数参照表 1.2.2。

表 1.2.2 常用数制的对应关系

十进制数	二进制数	八进制数	十六进制数	十进制数	二进制数	八进制数	十六进制数
0	0000	0	0	8	1000	10	8
1	0001	1	1	9	1001	11	9
2	0010	2	2	10	1010	12	A
3	0011	3	3	11	1011	13	B
4	0100	4	4	12	1100	14	C
5	0101	5	5	13	1101	15	D
6	0110	6	6	14	1110	16	E
7	0111	7	7	15	1111	17	F

【例 1.2.1】 把下面不同进制的数按照位权展开式展开。

解:

$$(55.55)_{10} = 5 \times 10^1 + 5 \times 10^0 + 5 \times 10^{-1} + 5 \times 10^{-2}$$

$$(101.011)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2} + 1 \times 2^{-3} = (5.375)_{10}$$

$$(36.4)_8 = 3 \times 8^1 + 6 \times 8^0 + 4 \times 8^{-1} = 3 \times 8 + 6 + 0.5 = (30.5)_{10}$$

$$(7B.11)_{16} = 7 \times 16^1 + 11 \times 16^0 + 1 \times 16^{-1} + 1 \times 16^{-2} = (123.0664)_{10}$$

从例 1.2.1 中可以看出: 不同数制按照位权展开计算的结果为对应十进制数。为了区别不同数制, 一般要加数字或者符号来对应区分。比如例 1.2.1 中的表示, 也可以用符号。例如, $10010001B=145D$, D 可以省略。

下面介绍将十进制数转换成其他进制数的方法。

1.2.3 数制间的转换

1. 十进制整数转换成二进制整数

十进制整数与二进制整数之间的对应转换为: 十进制整数除以 2 得余数, 先得到的余数为二进制整数的低位, 直至商为 0 结束, 或者按位权对应法进行转换。

【例 1.2.2】 将十进制数 $(179)_{10}$ 转换为二进制数。

解:

$$179 \div 2 = 89 \cdots \cdots 1(\text{LSB})$$

$$\div 2 = 44 \cdots \cdots 1$$

$$\div 2 = 22 \cdots 0$$

$$\div 2 = 11 \cdots 0$$

$$\div 2 = 5 \cdots 1$$

$$\div 2 = 2 \cdots 1$$

$$\div 2 = 1 \cdots 0$$

$$\div 2 = 0 \cdots 0 (\text{MSB})$$

即 $(179)_{10} = (10110011)_2$ 。

也可以按位权对应法来转换。十进制数 179 大于 2^7 小于 2^8 ，故转换为二进制数之后在 2^7 位置上有 1，即 $179 - 128 = 51 = 32 + 16 + 2 + 1 = 2^5 + 2^4 + 2^1 + 2^0$ ，因此转换为对应二进制数时在位权 2^7 ， 2^5 ， 2^4 ， 2^1 ， 2^0 的位置上的数码为 1，其他位置数码为 0，即 $(179)_{10} = (10110011)_2$ 。

2. 十进制小数转换成二进制小数

十进制小数与二进制小数之间的对应转换为：十进制小数乘 2 取整数，先得到的整数为二进制数小数点后的第一位，直到乘积为 0 结束或者积不为 0 但可得到需要的精度。

【例 1.2.3】 将十进制数 $(0.5)_{10}$ 转换为二进制数。

解：

$$0.5 \times 2 = 1.0 \cdots 1$$

即 $(0.5)_{10} = (0.1)_2$ 。

【例 1.2.4】 将十进制数 $(0.45)_{10}$ 转换为二进制数(保留 8 位有效数字)。

解：

$$0.45 \times 2 = 0.9 \cdots 0 (\text{MSB})$$

$$\times 2 = 1.8 \cdots 1$$

$$\times 2 = 1.6 \cdots 1$$

$$\times 2 = 1.2 \cdots 1$$

$$\times 2 = 0.4 \cdots 0$$

$$\times 2 = 0.8 \cdots 0$$

$$\times 2 = 1.6 \cdots 1$$

$$\times 2 = 1.2 \cdots 1 (\text{LSB})$$

即 $(0.45)_{10} = (0.01110011)_2$ 。

综上所述，十进制数转换成对应二进制数分别把整数部分和小数部分转换，再合到一起即可。

【例 1.2.5】 将十进制数 $(179.45)_{10}$ 转换为二进制数。

解：

$$179 \div 2 = 89 \cdots 1 (\text{LSB})$$

$$\div 2 = 44 \cdots 1$$

$$\div 2 = 22 \cdots 0$$

$$\div 2 = 11 \cdots 0$$

$$\div 2 = 5 \cdots 1$$

$$\div 2 = 2 \cdots 1$$

$$\div 2 = 1 \cdots 0$$

$$\div 2 = 0 \cdots 1 (\text{MSB})$$

$$0.45 \times 2 = 0.9 \cdots \cdots 0 (\text{MSB})$$

$$\times 2 = 1.8 \cdots \cdots 1$$

$$\times 2 = 1.6 \cdots \cdots 1$$

$$\times 2 = 1.2 \cdots \cdots 1$$

$$\times 2 = 0.4 \cdots \cdots 0$$

$$\times 2 = 0.8 \cdots \cdots 0$$

$$\times 2 = 1.6 \cdots \cdots 1$$

$$\times 2 = 1.2 \cdots \cdots 1 (\text{LSB})$$

即 $(179.45)_{10} = (10110011.01110011)_2$ 。

3. 二进制数转换成八进制数/十六进制数

二进制数和八进制/十六进制数的转换相对比较简单。因为 $8=2^3$, $16=2^4$, 所以只需要将一组二进制数转换为对应一位八(十六)进制数码即可。以小数点为界往前三(四)位作为一组, 前面不够一组则补零, 小数点往后三(四)位作为一组, 后面不够一组补零; 然后每组对应写出一个八(十六)进制数码即可。

【例 1.2.6】求 $(1010.1011)_2$ 的八进制、十六进制数。

解:

$$\begin{array}{ccccccc} 1010.1011 & & & & & & 1010.1011 \\ =\underline{001} & \underline{010} & . & \underline{101} & \underline{100} & & =\underline{1010} & . & \underline{1011} \\ \downarrow & \downarrow & & \downarrow & \downarrow & & \downarrow & & \downarrow \\ 1 & 2 & & 5 & 4 & & A & & B \end{array}$$

即 $(1010.1011)_2 = (12.54)_8 = (A.B)_{16}$ 。

4. 八进制数/十六进制数转换成二进制数

将八进制数和十六进制数转换为二进制数时, 把每一位数码对应写出三(四)位二进制数即可。

【例 1.2.7】求 $(12.54)_8$ 的二进制数。

解:

$$\begin{array}{ccccccc} & 1 & & 2 & & & 5 & & 4 \\ & \downarrow & & \downarrow & & & \downarrow & & \downarrow \\ 001 & & 010 & & & & 101 & & 100 \end{array}$$

最前面和最后面的 0 可以省掉, 即 $(12.54)_8 = (1010.1011)_2$ 。

【例 1.2.8】求 $(A.B)_{16}$ 的二进制数。

解:

$$\begin{array}{ccc} A & & B \\ \downarrow & & \downarrow \\ 1010 & & 1011 \end{array}$$

即 $(A.B)_{16} = (1010.1011)_2$ 。

5. 十进制数和八进制数/十六进制数之间的转换

将十进制数转换成八进制数或者十六进制数时, 可以先转换为二进制数, 再将二进制数转换成八进制或者十六进制数。也可以按照十进制整数除以 8(或 16)取余数, 十进制小数乘 8(或 16)取整数的方法得到。而任意进制数转换成十进制数则是按其对应基数的位权

展开即可。

【例 1.2.9】求 $(179)_{10}$ 的八进制、十六进制和二进制数。

解：

$$\begin{aligned} 179 \div 8 &= 22 \cdots 3 (\text{LSB}) \\ \div 8 &= 2 \cdots 6 \\ \div 8 &= 0 \cdots 2 (\text{MSB}) \end{aligned}$$

即 $(179)_{10} = (263)_8$ 。

$$\begin{aligned} 179 \div 16 &= 11 \cdots 3 (\text{LSB}) \\ \div 16 &= 0 \cdots B (\text{MSB}) \end{aligned}$$

即 $(179)_{10} = (B3)_{16}$ 。

$$\begin{aligned} 179 \div 2 &= 89 \cdots 1 (\text{LSB}) \\ \div 2 &= 44 \cdots 1 \\ \div 2 &= 22 \cdots 0 \\ \div 2 &= 11 \cdots 0 \\ \div 2 &= 5 \cdots 1 \\ \div 2 &= 2 \cdots 1 \\ \div 2 &= 1 \cdots 0 \\ \div 2 &= 0 \cdots 1 (\text{MSB}) \end{aligned}$$

即 $(179)_{10} = (10110011)_2$ 。

6. 二进制数的加减运算

无符号二进制数的运算按照与十进制类似的方法，从最低位开始相加，该位之和大于等于 2 时往其高位进一位。有符号二进制数的运算要注意溢出的问题。

【例 1.2.10】求下面两个四位二进制数的加法运算。

解：

$\begin{array}{r} 1001 \\ + 1101 \\ \hline 10110 \end{array}$	$\begin{array}{r} (+4) \quad 0100 \\ + (-7) \Leftrightarrow +1001 \\ \hline -3 \quad 1101 \end{array}$	$\begin{array}{r} (+6) \quad 0110 \\ + (-3) \Leftrightarrow +1101 \\ \hline +3 \quad 10011 \end{array}$
---	--	---

1.3 码制

为了识别 0 和 1 之外的数码和信息，需要采用二进制数字组合的方式，按照一定的规律来识别，这就是编码。对于 N 个信息，要用几位二进制数才能满足编码呢？根据 $2^n \geq N$ ，要用 n 位二进制数组成编码即可。

下面介绍常用的几种代码。

1.3.1 BCD 码

BCD(Binary Code Decimal, 二-十进制)码是用四位二进制数码表示一位十进制数的编码方式。有多种编码方式，常见的有 8421BCD 码、5421BCD 码、2421BCD 码等。编码规则及对应关系见表 1.3.1。表示多位十进制数时，需要多组 BCD 码按位对应。而有权 BCD 码转换成十进制数时，按位权展开即可。余 3 码和余 3 循环码都是无权码^[1]。

表 1.3.1 BCD 码的对应关系

十进制数	8421BCD 码	余 3 码	2421BCD 码	5421BCD 码	余 3 循环码
0	0 0 0 0	0 0 1 1	0 0 0 0	0 0 0 0	0 0 1 0
1	0 0 0 1	0 1 0 0	0 0 0 1	0 0 0 1	0 1 1 0
2	0 0 1 0	0 1 0 1	0 0 1 0	0 0 1 0	0 1 1 1
3	0 0 1 1	0 1 1 0	0 0 1 1	0 0 1 1	0 1 0 1
4	0 1 0 0	0 1 1 1	0 1 0 0	0 1 0 0	0 1 0 0
5	0 1 0 1	1 0 0 0	1 0 1 1	1 0 0 0	1 1 0 0
6	0 1 1 0	1 0 0 1	1 1 0 0	1 0 0 1	1 1 0 1
7	0 1 1 1	1 0 1 0	1 1 0 1	1 0 1 0	1 1 1 1
8	1 0 0 0	1 0 1 1	1 1 1 0	1 0 1 1	1 1 1 0
9	1 0 0 1	1 1 0 0	1 1 1 1	1 1 0 0	1 0 1 0
权值	$2^3 2^2 2^1 2^0$	比 8421 码多 3	$2^1 2^2 2^1 2^0$	$5 2^2 2^1 2^0$	

【例 1.3.1】求十进制数 $(98)_{10}$ 的对应 BCD 码。

解：

$$(98)_{10} = (1001\ 1000)_{8421BCD} = (1111\ 1110)_{2421BCD} = (1100\ 1011)_{5421BCD} = (1100\ 1011)_{\text{余3码}}$$

【例 1.3.2】求十进制数 $(463.5)_{10}$ 对应的 BCD 码。

解：

$$(463.5)_{10} = (0100\ 0110\ 0011.0101)_{8421BCD} = (0100\ 1100\ 0011.1011)_{2421BCD}$$

【例 1.3.3】求下列 BCD 码对应的十进制数。

解：

$$(0111)_{8421BCD} = 2^2 + 2^1 + 2^0 = (7)_{10}$$

$$(1101.0111)_{2421BCD} = (2^1 + 2^2 + 2^0) + 0.(2^2 + 2^1 + 2^0) = (7.7)_{10}$$

注意：8421BCD 码和二进制数的区别及联系，特别是在加法/减法计算时涉及进位/借位时。

【例 1.3.4】将下列十进制数分别转换成二进制数和 8421BCD 码，并进行加法运算： $7+2$ ， $9+2$ ， $9+9$ ，比较结果。

解：如表 1.3.2 所示。

表 1.3.2 二进制数和 8421BCD 码加法运算对应表

计算结果 ≤ 9 时		$10 \leq$ 计算结果 ≤ 15 时		计算结果 ≥ 16 时	
二进制数	8421BCD 码	二进制数	8421BCD 码	二进制数	8421BCD 码
0111	0111	1001	1001	1001	1001
+0010	+0010	+0010	+0010	+1001	+1001
1001	1001	1011	1011	1 0010	1 0010
			+0110		+0110
			1 0001		1 1000

$$7+2=9=(1001)_2=(1001)_{8421BCD}$$

$$9+2=11=(1011)_2=(0001\ 0001)_{8421BCD}$$

$$9+9=18=(10010)_2=(0001\ 1000)_{8421BCD}$$

即在加法运算中，当和数小于等于 9 时，二进制数和 8421BCD 码的表示方式一致；但是在和数大于 9 时，就是无效的 8421BCD 码，需要加 0110 进行修正。

1.3.2 典型格雷码

典型格雷码的编码规则是相邻码组之间只有一位状态不同,即互补,首尾也满足此规律。例如,两位典型格雷码为:00,01,11,10。可以看出,00和01有一位相同,一位互补,01和11也是一位相同,一位互补。而首尾10和00也满足此规律。类似地,三位典型格雷码表示为:000,001,011,010,110,111,101,100,见表1.3.3。

表 1.3.3 三位格雷码的编码规则

十进制数	三位格雷码	十进制数	三位格雷码
0	000	4	110
1	001	5	111
2	011	6	101
3	010	7	100

四位典型格雷码和十进制数的对应关系见表1.3.4。也可以用图1.3.1所示的方格图来表示十进制数和格雷码的对应关系,行列取值是格雷码(先行后列),方格内数字则是对应的十进制数。

表 1.3.4 十进制数对应的四位格雷码

十进制数	四位格雷码	十进制数	四位格雷码	十进制数	四位格雷码	十进制数	四位格雷码
0	0000	8	1100	4	0110	12	1010
1	0001	9	1101	5	0111	13	1011
2	0011	10	1111	6	0101	14	1001
3	0010	11	1110	7	0100	15	1000

	00	01	11	10
00	0	1	2	3
01	7	6	5	4
11	8	9	10	11
10	15	14	13	12

图 1.3.1 四位典型格雷码的方格图

1.3.3 奇偶校验码

有时候为了防止信息因干扰出现错误,需要在信息码的前面或后面加一位校验码。如果加了校验码后,检验码和信息码合起来的1的个数是奇数,则称为奇校验;反之为偶校验。

以四位信息码后所加校验位为例,编码规则见表1.3.5。

表 1.3.5 奇偶校验码编码规则

信息位	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
偶校验位	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
奇校验位	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
偶校验位	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
奇校验位	0	1	1	0	1	0	0	1	1	0	0	1	0	1	1	0
奇校验位	1	0	0	1	0	1	1	0	0	1	1	0	1	0		1

【例 1.3.5】 如果传送的信息码为 011011111010, 采用后面加校验位的方式, 分别写出对应的奇校验码和偶校验码。

解: (011011111010)_{信息码} 包括 8 个 1, 为偶数。因此校验位添加 1 才能凑够奇数个, 即奇校验码为: 0110111110101; (011011111010)_{信息码} 包括 8 个 1, 后面校验位添加 0, 即偶校验码为: 01101111101010。

1.3.4 ASCII 码

ASCII 码是 American Standard Code for Information Interchange 的简称, 即美国信息交换标准代码。它是由美国国家标准学会(American National Standard Institute, ANSI)制定的标准的单字节字符编码方案, 用于基于文本的数据。它起始于 20 世纪 50 年代后期, 在 1967 年定案。它最初是美国国家标准, 供不同计算机在相互通信时共同遵守的西文字符编码标准, 它已被国际标准化组织(International Organization for Standardization, ISO)定为国际标准, 称为 ISO 646 标准^[2]。

该编码适用于所有拉丁文字字母。采用 7 位二进制编码, 表示 2^7 (即 128) 个字符。其中:

0~31 及 127 (共 33 个) 是控制字符或通信专用字符 (其余为可显示字符), 如控制字符: LF (换行)、CR (回车)、FF (换页)、DEL (删除)、BS (退格)、BEL (响铃) 等; 通信专用字符: SOH (文头)、EOT (文尾)、ACK (确认) 等; ASCII 值为 8、9、10 和 13 分别表示退格、制表、换行和回车符。它们并没有特定的图形显示, 但会依不同的应用程序, 对文本显示有不同的影响。

32~126 (共 95 个) 是字符 (32 是空格), 其中 48~57 为 0~9 的阿拉伯数字。

65~90 为 26 个大写英文字母, 97~122 为 26 个小写英文字母, 其余为一些标点符号、运算符号等。

1.4 基本逻辑运算

数字电路用二进制的逻辑运算来表示事件的因果关系。逻辑运算的变量只有 0 和 1 两种取值, 这个变量称为二值逻辑变量。逻辑运算是由逻辑代数 (布尔代数) 来表示并且进行变换或者化简的。逻辑代数可以由逻辑表达式表示, 也可以由编程语言描述或者用真值表、波形图等作为逻辑工具。常用的逻辑运算有“与”“或”“非”三种。下面分别进行介绍。

1.4.1 与运算

与 (AND) 逻辑一般用在只有当全部条件具备时, 结果发生这样的事件中。一个简单的与逻辑电路如图 1.4.1 所示。电源 E 通过开关 A、开关 B 接灯泡, 当开关 A 和开关 B 同时闭合时, 回路形成, 灯泡亮; 如果开关 A 或者开关 B 有一个断开或者两个开关都断开时, 则灯泡不亮。开关为因, 灯亮为果。这种关系可以用如表 1.4.1 所示的功能表来表示。

如果用二值逻辑来表示功能表, 则表 1.4.1 可以转化为表 1.4.2 的形式, 表 1.4.2 为真值表。正逻辑方式为开关闭合和灯亮用 1 代表, 开关断开和灯不亮用 0 代表。反过来, 如果用 0 代表开关闭合和灯亮的状态, 用 1 代表开关断开和灯不亮的状态则为负逻辑。一般默认用正逻辑表示。

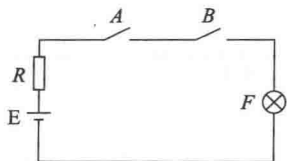


图 1.4.1 与逻辑电路

表 1.4.1 与运算功能表

开关 A	开关 B	灯 F
断开	断开	不亮
断开	闭合	不亮
闭合	断开	不亮
闭合	闭合	亮

表 1.4.2 与运算真值表

开关 A	开关 B	灯 F
0	0	0
0	1	0
1	0	0
1	1	1

根据表 1.4.2, 灯 F 作为输出变量, 开关 A 、开关 B 作为输入变量, 可以用逻辑表达式表示为:

$$F = A \cdot B = A \cap B = A \& B = A \wedge B = AB \quad (1-4-1)$$

式(1-4-1)中, 小圆点“ $\cdot$ ”“ $\cap$ ”“ $\&$ ”“ $\wedge$ ”等代表“与”运算的符号, 不同书中用的符号不同, 小圆点是最常用的, 有时也可以省略。能实现与运算的逻辑电路称为与门, 可以用图 1.4.2 所示的符号来表示。

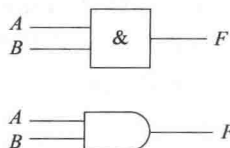


图 1.4.2 与运算符号

从真值表和逻辑表达式可以得出与运算的逻辑规律:

$$0 \cdot 0 = 0, \quad 0 \cdot 1 = 0, \quad 1 \cdot 1 = 1, \quad A \cdot 0 = 0, \quad A \cdot 1 = A \quad (1-4-2)$$

因而可得出与运算的逻辑规律是“有 0 出 0, 全 1 出 1”。

上述与运算可以推广到多变量的情况:

$$F = A \cdot B \cdot C \cdots \quad (1-4-3)$$

1.4.2 或运算

或(OR)运算指所有条件中只要有条件具备结果就会发生。一个简单的或逻辑电路如图 1.4.3 所示。电源 E 通过开关 A 或开关 B 接灯泡, 当开关 A 或者开关 B 闭合时, 回路形成, 灯泡亮; 如果开关 A 或者开关 B 都断开, 则灯不亮。这种关系可以用表 1.4.3 所示的功能表来表示。

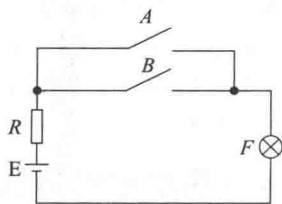


图 1.4.3 或运算示意图

表 1.4.3 或运算功能表

开关 A	开关 B	灯 F	开关 A	开关 B	灯 F
断开	断开	不亮	闭合	断开	亮
断开	闭合	亮	闭合	闭合	亮

如果开关闭合和灯亮用 1 代表, 开关断开和灯不亮用 0 代表, 则功能表可以对应表示成表 1.4.4 所示的真值表。

表 1.4.4 或运算真值表

开关 A	开关 B	灯 F	开关 A	开关 B	灯 F
0	0	0	1	0	1
0	1	1	1	1	1

根据表 1.4.4, 灯 F 作为输出变量, 开关 A 、开关 B 作为输入变量, 可以用逻辑表达式表示为: