高等职业教育优质校建设轨道交通通信信号技术专业群系列教材

数字电子技术及其应用

（活页式）

主　编◎张　莉　吴　昕

副主编◎高　辉

西南交通大学出版社

·成　都·

图书在版编目（CIP）数据

数字电子技术及其应用：活页式 / 张莉，吴昕主编
. —成都：西南交通大学出版社，2023.1
ISBN 978-7-5643-9129-4

Ⅰ. ①数… Ⅱ. ①张… ②吴… Ⅲ. ①数字电路－电子技术－高等职业教育－教材 Ⅳ. ①TN79

中国版本图书馆 CIP 数据核字（2022）第 256971 号

Shuzi Dianzi Jishu ji Qi Yingyong（Huoyeshi）
数字电子技术及其应用（活页式）

主　编 / 张 莉　吴 昕　　责任编辑 / 李华宇
封面设计 / 吴　兵

西南交通大学出版社出版发行
（四川省成都市金牛区二环路北一段 111 号西南交通大学创新大厦 21 楼　610031）
发行部电话：028-87600564　028-87600533
网址：http://www.xnjdcbs.com
印刷：四川玖艺呈现印刷有限公司

成品尺寸　185 mm×260 mm
印张　20　　字数　462 千
版次　2023 年 1 月第 1 版　　印次　2023 年 1 月第 1 次

书号　ISBN 978-7-5643-9129-4
定价　45.00 元

课件咨询电话：028-81435775
图书如有印装质量问题　本社负责退换

前言
PREFACE

数字电子技术是当前发展最快的学科之一。数字逻辑器件从 20 世纪 60 年代的小规模集成电路（SSI），经历中规模集成电路（MSI）和大规模集成电路（LSI）等阶段，发展到今天的超大规模集成电路（VLSI）。相应地，数字逻辑电路的分析和设计方法也在不断演变和发展。为适应现代电子技术飞速发展的需要，根据高等职业教育的培养目标和能力要求，结合高等职业技术教育培养技术应用型人才的特点，本书在内容编写上，贯彻“理论与工程实践相结合，以应用为目的”的原则，淡化烦琐复杂的数学计算，重视单元电路的工作原理；淡化数字部件的内部结构，重视逻辑功能的外部分析；淡化理论推导，重视实际应用。

本书以数字信号的产生、传输和处理过程为主线，介绍了数字电路的基础知识、数字系统分析和设计的基本方法、数字电路的功能特性以及常用中小规模集成电路的综合应用等内容。全书共分为 8 个项目，分别为数字电路基础知识、集成逻辑门电路、组合逻辑电路、触发器、时序逻辑电路、脉冲产生与信号变换、集成电路发展史和 Multisim 仿真软件的基本操作等。全书内容精炼，逻辑清晰，深浅合理，图表规范，主要有以下特点：

（1）包含中文和英文两个版本，双语结合，培养学生阅读专业外文文献的能力，提高学生专业知识水平。

（2）以党的二十大精神为引领，深耕教材，挖掘课程的思政元素，以“科技强国、团结合作、严以律己、民族自信、规矩方圆”为核心内容构建了课程思政育人体系，并将其细微融入项目内容中，实现立德树人、德技并升的育人目标。

（3）应用实践环节采用传统线下+智能线上的方式，引入可视化仿真技术，创新在线实验新模式。利用 Multisim 软件和云实验平台可对每个项目后的应用实践项目进行仿真和在线实验，弥补了实验环节无法在线进行的缺点，实验报告采用线下活页式报告单+线上云实验平台智能报告单形式，便于学生灵活选择实验形式，使学生的动手实践能力、创新思维能力得到充分发挥与施展，实现了从“要我学”到“我要学”的转变。

（4）与课程内容紧密结合的微课、动画等颗粒化新媒体资源同步上线，使基础性、预备性、理论性较强的知识点更易被学生吸收和掌握，以便更好地实现“便于教，易于学”的目的。

（5）增加“应用拓展”模块，加强实用电路的举例，突出功能应用电路的分析，

拓宽学生的知识面，引导学生将理论与实践相结合，体现高等职业教育特色，实现了将理论知识由“是什么”到“怎么用”的转化。

（6）校企共同开发教材，把企业案例融入教材，利于学生掌握实践技能，使学习内容和工作内容具有连接性，学习要求体现了岗位的需求，为学生将来顺利投入工作打下基础，增加教材的实用性，也体现了职业教育的特点。

本书由郑州铁路职业技术学院张莉、吴昕担任主编，高辉担任副主编。具体编写分工：吴昕编写项目一和项目六，高辉编写项目二和项目三，任全会编写应用实践二，张莉编写项目四、项目五及附录 A，刘海燕编写项目四的应用实践，龙芯中科技术股份有限公司叶骐宁和河南国芯联合教育科技服务有限公司朱冉编写项目七，高基豪编写项目八，陈志红编写附录 B、附录 C、附录 D 和附录 E，马蕾编写应用实践三。全书由张莉、吴昕负责统稿，微课由陈志红、刘海燕、马蕾、张莉、高基豪、吴昕、罗丽宾、孙逸洁、朱力宏、任全会等提供，动画由张莉设计和整理。

本书在编写过程中得到了西南交通大学出版社的支持及帮助，龙芯中科技术股份有限公司、河南国芯联合教育科技服务有限公司提供了部分技术资料，在此表示衷心感谢。

由于作者水平有限，书中难免存在错误和疏漏之处，敬请广大读者批评指正。

编　者

2022 年 10 月

数字资源索引

序号	资源名称	类型	页码
32	非门电路	动画	35
33	数字集成器件的选用原则	微课	38
34	常见集成逻辑门芯片介绍	微课	38
35	TTL 与非门	微课	39
36	TTL 与非门	动画	39
37	TTL 与非门主要参数	微课	41
38	TTL 与非门主要参数	动画	41
39	OC 门	微课	43
40	OC 门	动画	43
41	三态门	微课	45
42	三态门	动画	45
43	集成逻辑门多余输入端处理	微课	46
44	数字集成器件的选用原则	微课	46
45	CMOS 门电路	微课	47
46	CMOS 门电路	动画	47
47	集成 CMOS 与非门和或非门	微课	49
48	CMOS 与 TTL 间的接口电路	微课	50
49	门电路与其他电路的连接	微课	50
50	组合逻辑电路的分析	微课	66
51	组合逻辑电路的分析	动画	66
52	组合逻辑电路的设计	微课	68
53	组合逻辑电路的设计	动画	68
54	二进制编码器	微课	71
55	优先编码器	微课	72
56	二进制译码器	微课	74
57	二进制译码器	动画	74
58	译码器的应用	微课	76
59	半导体数码管	微课	79
60	半导体数码管	动画	79
61	显示译码器	微课	80
62	显示译码器	动画	80
63	数据选择器	微课	81
64	数据选择器	动画	81

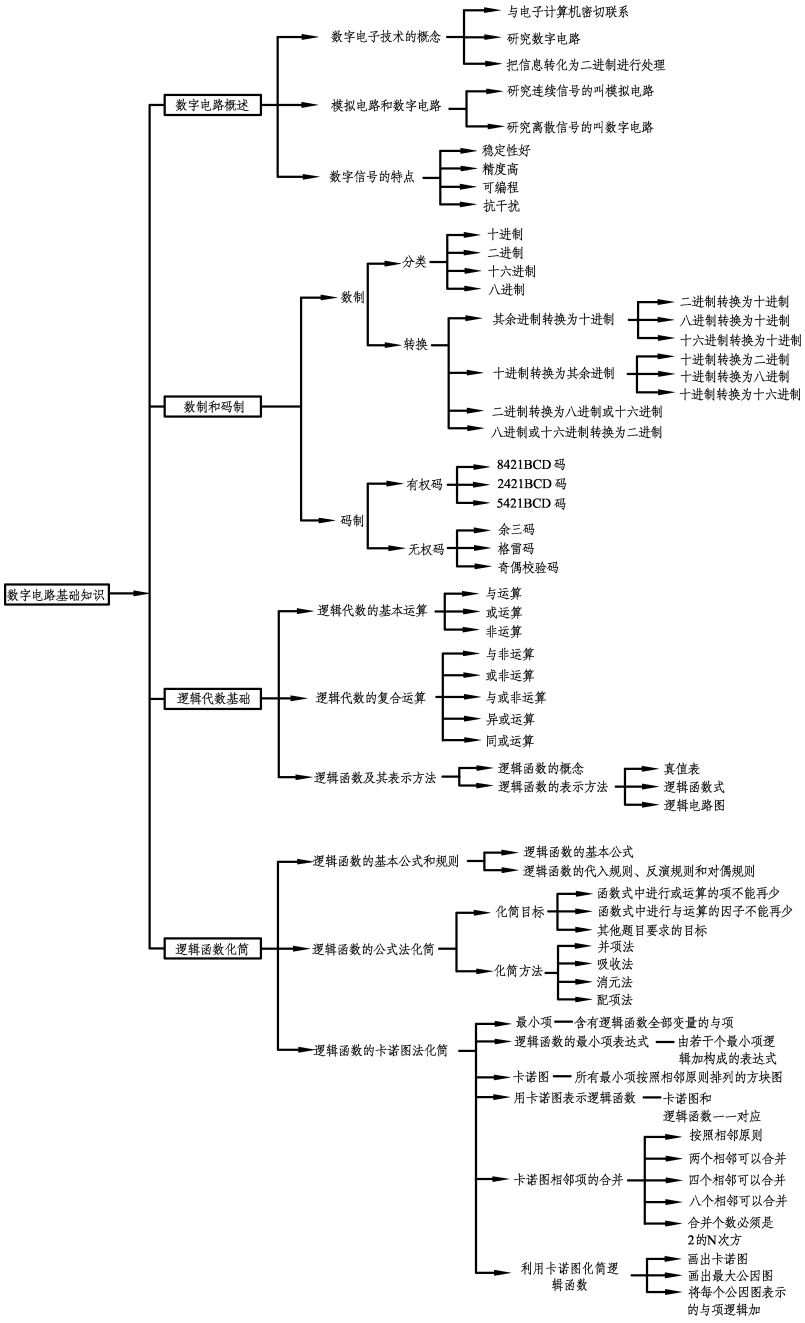
序号	资源名称	类型	页码
65	数据选择器通道扩展	微课	83
66	数据分配器	微课	84
67	数值比较器	微课	87
68	多位数值比较器	微课	88
69	门电路逻辑功能测试	微课	94
70	设计一个三输入表决器	微课	94
71	集成译码器逻辑功能测试	微课	98
72	用集成译码器组成一位全加器	微课	99
73	基本 RS 触发器	微课	105
74	基本 RS 触发器	动画	105
75	触发器逻辑功能的表示方法	微课	106
76	同步 RS 触发器	微课	110
77	同步 RS 触发器	动画	110
78	主从 JK 触发器	微课	121
79	主从 JK 触发器的主从结构	动画	121
80	主从 JK 触发器的一次翻转	微课	122
81	D 触发器和 T 触发器	微课	123
82	触发器使用注意事项	微课	135
83	组合逻辑电路与时序逻辑电路的区别	动画	158
84	时序逻辑电路的概念	微课	158
85	异步二进制计数器	微课	167
86	同步二进制计数器	微课	171
87	十进制计数器	微课	172
88	异步集成计数器 74LS290 逻辑功能	微课	175
89	异步计数器 74LS290 的结构和功能	动画	175
90	集成计数器 74LS290 功能扩展	微课	178
91	同步集成计数器 74LS161 逻辑功能	微课	180
92	同步计数器 74LS161 的结构与功能	动画	180
93	集成计数器 74LS161 功能扩展	微课	182
94	寄存器	微课	188
95	单向移位寄存器	微课	191
96	集成移位寄存器及其应用	微课	196
97	移位寄存器 74HC194 的功能	动画	196

序号	资源名称	类型	页码
98	555 定时器逻辑功能	微课	238
99	集成 555 定时器内部电路结构	动画	238
100	555 定时器构成的施密特触发器	微课	241
101	555 定时器构成的单稳态触发器	微课	245
102	555 定时器构成的多谐振荡器	微课	247
103	ADC 的基本概念及组成	微课	251
104	ADC 主要技术指标和集成 ADC 器件	微课	254
105	DAC 主要技术指标和集成 DAC 器件	微课	258
106	555 定时器的功能检查	微课	269
107	多谐振荡器的连接和观测	微课	269

目 录
CONTENTS

项目一 数字电路基础知识

思维导图



学习目标

（1）掌握数字电子技术和模拟电子技术的区别，理解数字信号的特点。
（2）掌握十进制、二进制等四种数制的表示方法和相互转换。
（3）培养严谨认真的学习态度，学会“对应”与“统一”的辩证思维方法。
（4）掌握 BCD 码和有权码、无权码的概念。
（5）掌握与或非三种基本逻辑关系及其运算规则。
（6）掌握五种复合逻辑关系及其运算规则。
（7）掌握逻辑函数的四种表示方法。
（8）熟练运用逻辑函数的基本公式。
（9）熟练运用公式法化简的四种方法。
（10）掌握最小项的概念。
（11）会用卡诺图法化简逻辑函数。

【引言】

电子技术分为模拟电子技术和数字电子技术两部分，在时间和幅值上连续的信号叫作模拟信号，模拟电子技术主要研究模拟信号的处理方法；在时间和幅值上离散的信号叫作数字信号，数字电子技术主要研究数字信号的处理方法。本项目讲解了数字电路的基础知识，包括常用的数制、码制、数字信号的表示方法、逻辑关系的分析、逻辑函数的运算等。由“0”和“1”构成的二进制是最常用的数字信号。它既可以代表万物的“无”和“有”，也蕴含了“对应”和“统一”的哲学辩证思想。

任务一　数字电路概述

一、数字电子技术的概念

科学技术迅速发展的今天，我们进入了“数字时代”，什么是数字电子技术呢？

数字电子技术是一项与电子计算机联系密切的科学技术，它是指借助一定的设备将图、文、声、像等各种信息转化为电子计算机能识别的二进制数字“0”和“1”，再进行运算、加工、存储、传送、还原等操作的技术。

电子技术的发展与电子器件的发展密不可分。1947 年，第一只晶体三极管的问世开创了电子技术的新领域，晶体管不仅为电子技术开辟了道路，也为计算机世界开辟了道路。各种类型的计算机开始在市场上出现。随后的 20 世纪 60 年代，模拟集成电路和数字集成电路相继上市，使电子技术进入一个迅速发展的时代。集成电路是一种半导体器件，它把晶体管、电容、电阻等元件及它们之间的连线制作在一块硅片上，使电路的体积迅速降低，使设备小型化成为现实。集成电路的飞速发展使电子技术尤其是数字电子技术日新月异。

数字电子技术最典型的应用是计算机和微处理器，数字电子技术的快速发展推动了处理器功能的不断发展和完善，掀起了一场“数字革命”。以电影为例，传统的胶片电影是以感光原料（如卤化银等），在胶片上成像，之后再快速拉动胶片形成动态图像。胶片电影清晰度低，不易保存和传输。而数字电影是把影像的光信号转换为电信号，把光强度、颜色、位置等信号都以数字信号的形式进行存储，可以永远保持质量稳定，方便存储和传输。

随着数字电子技术的发展，工业自动化、农业现代化、办公自动化、通信网络化已经成为现实，但是数字电子技术不能代替模拟电子技术，因为物理世界的信号绝大多数是模拟信号，我们需要把模拟信号采集，转换为数字信号之后才可以用数字电子技术处理，因此实际的电路系统通常是模拟电路和数字电路的结合。

二、模拟电路和数字电路

动画 模拟信号与数字信号　　微课 数字信号及数字电路

信号的形式是多种多样的，例如时间、温度、压强、路程等都是在连续的时间范围内有定义且幅度连续变化的信号。如图 1-1（a）所示，这种连续变化的信号称为模拟信号，它在一定的时间范围内可以有无限多个不同的取值，处理模拟信号电路就是模拟电路。为了便于处理，要先使用转换设备（如传感器）等把物理量转换为变化完全一致的电信号再进行处理。

在时间上和幅值上都是离散的信号称为数字信号。用于产生、传输和处理数字信号的电路称为数字电路。数字电路的主要研究对象是电路的输入和输出之间的逻辑关系。数字电路只有两种状态，如电位的高与低、电流的有与无、开关的通与断等，分别用“1”和“0”表示，而“1”和“0”分别对应着数字电路中的高电平和低电平。一个 1 或一个 0 通常称作 1 比特。图 1–1（b）所示为数字信号 111010111111；图 1–1（c）所示为用高电平代表 1、低电平代表 0 的数字电路的信号波形。

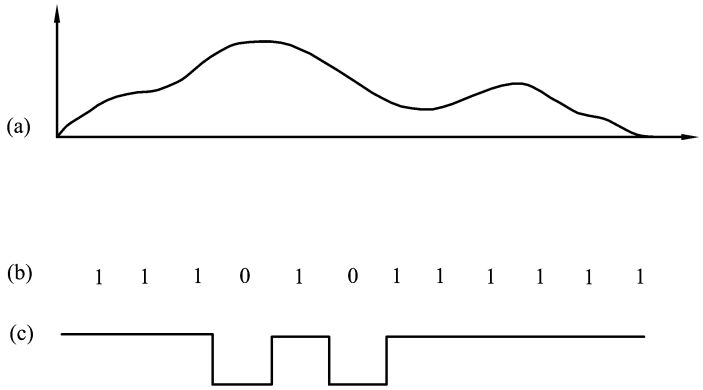


图 1-1　模拟信号和数字信号

三、数字信号的特点

1. 稳定性好

模拟电路中元件、电阻、电容、运算放大器的特性会随着温度、湿度的变化而变化，而数字电路只需分辨出信号的有与无，或者电压信号的高和低，因此电路的组件参数允许有较大的变化范围，在相同的工作条件下，数字电路更稳定。

2. 精度高

模拟电路中元件的误差比较大，如电阻器有 5%的公差，电容器公差在 20%左右；而数字电路中提高 ADC 位数、CPU 字宽和算法均可提高精度，且稳定不受干扰。

3. 可编程

电路的设计组成只需采用一些标准的集成电路块单元连接而成。对于非标准的特殊电路还可以使用可编程序逻辑阵列电路，通过编程的方法实现任意的逻辑功能。相对来说，模拟电路如果需要改变电路功能就需要改变硬件设计。

4. 抗干扰

数字电路处理的是二进制信息 0 和 1，分别对应着电路的低电平和高电平，只要外界干扰在电路的噪声允许范围内，电路就可以正常工作，因此数字电路抗干扰能力很强。模拟信号和数字信号的抗干扰性如图 1-2 所示。

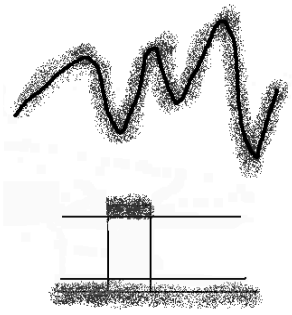


图 1-2　模拟信号和数字信号的抗干扰性

任务二　数制和码制

一、几种常用数制

微课 数制　　动画 数制

我们生活中常用的数制是十进制，任何一个数都可以用 0～9 的一个或者几个数码来表示。十进制的特点是“逢十进一”，但是数字系统中广泛采用的是二进制，为了方便存储数据，还用到十六进制和八进制，下面我们来学习这些数制。

1. 十进制

十进制是我们最熟悉的数制，它使用 0、1、2、3、4、5、6、7、8、9 十个数码，把其中的一个或者几个按照规律排列起来表示不同大小的数字。例如，同样使用 5、6、7 三个数码组成的两个数 567 和 765，大小是不同的，其中 5 这个数字在 567 中表示 5 个 100，而在 765 中则表示 5 个 1。每个数码处在不同数位时所代表的数值是不同的，例如十进制数 737 可表示为

$$(737)_{\mathrm{D}} = 7\times10^2 + 3\times10^1 + 7\times10^0$$

这里用下标“D”来表示十进制数，其中 10^2、10^1、10^0 分别为百位、十位、个位的权（Weight），也就是相应位的 1 所代表的实际数值。显然，位数越高权值越重；相邻两位的权值相差正好为该数制的基数，即相邻位间按“逢十进一”或“借一当十”的规律排列。

系数：0、1、2、3、4、5、6、7、8、9

权：10^i

基数：10

任意一个十进制数都可以表示为系数和权的乘积再相加的形式：

$$\begin{aligned}(N)_{\mathrm{D}} &= K_n\cdot10^n + K_{n-1}\cdot10^{n-1} + \mathrm{L} + K_1\cdot10^1 + K_0\cdot10^0 + K_{-1}\cdot10^{-1} + \mathrm{L} + K_{-m}\cdot10^{-m}\\ &= \sum_{i=-m}^{n-1} K_i\cdot10^i\end{aligned}$$

式中，K_i 为十进制基数 10 的 i 次幂的系数，它可取 0～9 中任一个数字。

2. 二进制

在数字电路中应用最广泛的是二进制，用下标“B”表示二进制数。二进制数的系数 K_i 只取二个数码“0”和“1”，与十进制的区别在于基数和权上。二进制的基数是 2，权是 2^i，二进制的“进”“退”位规律是“逢二进一”或“借一当二”。

任意一个二进制数 $(N)_{\mathrm{B}}$ 都能分解成按系数和权的乘积再相加的形式：

$$\begin{aligned}(N)_{\mathrm{B}} &= K_n\cdot2^n + K_{n-1}\cdot2^{n-1} + \mathrm{L} + K_1\cdot2^1 + K_0\cdot2^0 + K_{-1}\cdot2^{-1} + \mathrm{L} + K_{-m}\cdot2^{-m}\\ &= \sum_{i=-m}^{n-1} K_i\cdot2^i\end{aligned}$$

二进制最突出的优点是简单，只需用 0 和 1 两个数码，在电路中有最简洁的实现方案，即用“开”和“关”两种状态表示。

3. 十六进制

二进制虽简单且电路实现方便，但与十进制比较，人们使用不大习惯，与等值十

进制数相比，它所需要的位数较多，不便于书写和记忆，因此在计算机系统中经常用十六进制数，用“H”表示十六进制数。

十六进制采用 0～9、A（对应十进制数 10）、B（11）、C（12）、D（13）、E（14）、F（15）十六个数码作为系数，其基数为 16，权为 16^i，相邻位间计数规律是“逢十六进一”或“借一当十六”。

同上，十六进制数可按系数和权展开为

$$(N)_{\mathrm{H}}=\sum_{i=-m}^{n-1}K_i\cdot 16^i$$

4. 八进制

同理可表述，八进制数的基数 K_i 为 0、1、2、3、4、5、6、7 八个数码，计数规律是“逢八进一”或“借一当八”，用“O”表示八进制数。

八进制数可按权展开为

$$(N)_{\mathrm{O}}=\sum_{i=-m}^{n-1}K_i\cdot 8^i$$

二进制、八进制、十进制及十六进制四种不同数制的对照关系如表 1-1 所示。

表 1-1　数制对照表

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

二、数制之间的转换

1. 二进制、十六进制、八进制转换为十进制

动画 数制转换　　微课 不同数制之间的转换

二进制转换为十进制是把每一位的二进制数和它的权相乘再相加，便得到相应的十进制数。

【例 1-1】 将二进制数（11101）$_{\mathrm{B}}$ 转换成十进制数。

解 （11101）$_{\mathrm{B}}=1\times2^4+1\times2^3+1\times2^2+0\times2^1+1\times2^0=$（29）$_{\mathrm{D}}$

而把十六进制和八进制转换为十进制也是相同的方法。

【例 1-2】 将八进制数（64.72）$_O$ 转换成十进制数。

解 （64.72）$_O = 6\times8^1 + 4\times8^0 + 7\times8^{-1} + 2\times8^{-2} = 48 + 4 + 0.875 + 0.125 =$（53）$_D$

【例 1-3】 将十六进制数（1F6.B2）$_H$ 转换成十进制数。

解 （1F6.B2）$_H = 1\times16^2 + 15\times16^1 + 6\times16^0 + 11\times16^{-1} + 2\times16^{-2}$

$= 256 + 240 + 6 + 0.6875 + 0.0078125 =$（502.6953125）$_D$

2. 十进制转换为二进制、十六进制、八进制

将一个十进制数转换成二进制数，要将十进制数的整数部分和小数部分分开进行转换，可采用“基数乘除法，存余取整”，再将整数、小数部分合并起来。

1）整数转换

将十进制数的整数部分除以 2，保存余数并作为二进制数的最低位数，并把前一步的商再除以 2，还保存余数作为次低位数，重复上一步过程，保存住余数，直至最后商为 0，这时最后的余数才为 N 进制的最高位数。

【例 1-4】 将十进制数（92）$_D$ 转换成二进制数。

解：

```
2 | 92 ……0   ……最低位
  2 | 46 ……0   ……次低位
    2 | 23 ……1
      2 | 11 ……1
        2 | 5 ……1
          2 | 2 ……0
            2 | 1 ……1   ……最高位
              | 0   最后的商必须为 0
```

所以，（92）$_D$ =（1011100）$_B$

将一个十进制整数转换为十六进制和八进制，只需要把整数部分除以 2 改为除以 16 和 8 再由下至上取余数即可，方法一致。

2）小数转换

十进制纯小数转换成二进制数，采用“连乘基数取整法”，逐次乘 2，逐次取出整数，直至最后乘积为 0 或达到某个精度为止。当乘至第 n 位时，若乘积不等于 0，但看成近似为 0，这就存在了转换误差。转换误差的估算：取到小数点后第几位，转换误差就为（基数）$^{-n}$。

【例 1-5】 将十进制小数（0.3721）$_D$ 转换成二进制数（取到小数点后八位）。

解：

```
  0．3721
  ×      2
(0)．7442      B-1=0……最高位
  ×      2
(1)．4884      B-2=1
  ×      2
(0)．9768      B-3=0
  ×      2
(1)．9536      B-4=1
```

$$\begin{array}{r} \times \quad 2 \\ \hline (1).\ 9072 \end{array} \qquad B_{-5}=1$$

$$\begin{array}{r} \times \quad 2 \\ \hline (1).\ 8144 \end{array} \qquad B_{-6}=1$$

$$\begin{array}{r} \times \quad 2 \\ \hline (1).\ 6288 \end{array} \qquad B_{-7}=1$$

$$\begin{array}{r} \times \quad 2 \\ \hline (1).\ 2576 \end{array} \qquad B_{-8}=1\cdots\cdots\cdots\text{最低位}$$

所以（0.3721）$_D$ =（0.01011111）$_B$

转换误差为 2^{-8} = 1/256 = 0.0039

将一个十进制小数转换为十六进制数和八进制数，只需要把小数部分乘 2 改为乘 16 和 8 再取整数即可，方法一致。

3. 二进制和十六进制、八进制之间的转换

因为八进制数、十六进制数严格说来都可归于二进制数，所以它们之间的相互转换就显得十分方便，并且也不存在转换误差。

如八进制数每位的基数为 $8 = 2^3$，相当于三位二进制数。因此，就有“每一位八进制相当于三位的二进制；每三位二进制相当于八进制的一位”。

【例 1-6】 将（115. 734）o 转换成二进制数。

解 （115. 734）$_O$ =（001 001 101.111 011 100）$_B$

【例 1-7】 将（111 001.011 101）$_B$ 转换成八进制数。

解 （111 001.011 101）$_B$ =（71.35）$_O$

再如，十六进制数每位的基数为 $16 = 2^4$，相当于四位二进制数。因此，就又有“十六进制的每一位相当于四位二进制；每四位二进制相当于十六进制的一位”。

【例 1-8】 将（A6D.8F）$_H$ 转换成二进制数。

解 （A6D.8F）$_H$ =（1010 0110 1101.1000 1111）$_B$

【例 1-9】 将（1001.1011 0101 0011）$_B$ 转换成十六进制数。

解 （1001.1011 0101 0011）$_B$ =（9.B53）$_H$

将八进制和十六进制相互转换时，可借助二进制来完成。

三、几种常用码制

数字系统中使用 0 和 1 组成的代码表示数值或者特定的信息。当表示数值时，该二进制代码表示数量的大小；当表示特定信息时，该二进制代码不表示数量大小，仅仅用于区别不同的事物。这些代码都是以一定规则编制的，编制的过程就成为编码，编制的规则就称为码制。

微课 码制

编码的形式有很多，码制也有很多，BCD 码是其中的一类，用四位二进制数来表示十进制数中的 0 ~ 9 十个数码，称为二-十进制代码，简称 BCD（Binary Coded Decimal）代码。

当采用不同的编码方案时，可以得到不同形式的 BCD 码。下面介绍几种常用的 BCD 码。

1. 常用 BCD 码

1）8421BCD 码

8421BCD 码是最基本、最常用的有权 BCD 码，即其 4 位二进制代码中，每位二进制数码都对应有确定的位权值，即 $B_4 = 8$，$B_3 = 4$，$B_2 = 2$，$B_1 = 1$。

例如，8421BCD 码中的 1001 代表：$8\times1 + 4\times0 + 2\times0 + 1\times1 =（9）_D$

应当指明的是，在 8421BCD 码中不允许出现 1010～1111 这 6 个代码，它们是没有意义的。

2）2421BCD 码和 5421BCD 码

只要满足最低权位值为 1、四位权值之和大于等于 9、且能区分开 0～9 十个数码的均可构成有权的 BCD 码。

2421BCD 码和 5421BCD 码也是有权码，它们的 0 和 9、1 和 8、2 和 7、3 和 6、4 和 5 恰好互为反码，这种特性称为具有自补性，在数字系统的信号传输中是很有用的。

3）余 3BCD 码

余 3BCDBCD 码，是在每个 8421BCD 代码上加上（0011）$_B$ 而得到的。余 3BCD 码各位无固定的位权，也称为无权码，用余 3BCD 码进行加减运算比 8421BCD 码方便、快捷。

2. 格雷码、奇偶校验码的特点及应用

代码在形成、传输的过程中，由于偶然因素会产生误码，为了减少这种误码，就要对代码的形式进行筛选，挑选出在实际传输过程中不容易出错的代码，这就是可靠性编码。

可靠性编码有许多，其中最突出的就是格雷（Gray）码和奇偶校验码。

1）格雷码

格雷（Gray）码，又称循环码。它利用了所有的十六种组合，因此也是一种无权码。它的编码特点是任两相邻代码间只有一位数码不同，所以在传输过程中易被机器识别而不容易出错。

2）奇、偶校验码

可靠性编码中还有一种常用的代码，就是奇、偶校验码（Parity Code）。它除了表示传输信息的代码外，又增加了一位奇、偶校验位，用来标记传输信息的代码中“1”的奇、偶数。

它的编码有两种方式：使得一个代码组中信息位和校验位中“1”的总个数为奇数的叫奇校验；“1”的总个数为偶数的叫偶校验。奇校验和偶校验在计算机中获得广泛应用。通常，对接收到的奇偶校验码要进行检查，查看码中“1”的个数的奇偶是否正确。如果不对，就是错误代码（或称非法码），也就说明信息传递有错。

常用码制见表 1-2。

表 1-2　常用码制

十进制数	编码					
	有权码			无权码		
	8421BCD	2421 BCD 码	5421BCD 码	余 3 码	格雷码	奇偶校验码（奇校验）
0	0000	0000	0000	0011	0000	10000
1	0001	0001	0001	0100	0001	00001
2	0010	0010	0010	0101	0011	00010
3	0011	0011	0011	0110	0010	10011
4	0100	0100	0100	0111	0110	00100
5	0101	1011	1000	1000	0111	10101
6	0110	1100	1001	1001	0101	10110
7	0111	1101	1010	1010	0100	00111
8	1000	1110	1011	1011	1100	01000
9	1001	1111	1100	1100	1101	11001
10					1111	
11					1110	
12					1010	
13					1011	
14					1001	
15					1000	

任务三　逻辑代数基础

在数字系统中，当 0 和 1 表示的是逻辑状态时，二进制数码按照某种特定的因果关系进行逻辑运算，逻辑运算与数学运算规则不同，它使用的数学工具是逻辑代数（又叫布尔代数）。逻辑代数也有逻辑函数、逻辑变量和逻辑运算这几个要素，不同的是，普通代数中变量的取值是任意的，而逻辑变量的取值只有 0 和 1，逻辑代数中的 0 和 1 不表示数值的大小，只代表事物的两种状态，如开关的断开和闭合、灯的亮和灭等。

逻辑代数的基本运算有三种，我们来看下面的例子。

一、逻辑代数的基本运算

微课 与逻辑及与门　　动画 与逻辑及与门

1. 与运算

如图 1-3 所示，指示灯控制电路由两个开关 *A* 和 *B* 来控制灯 *P* 的亮和灭，每个开关都有两种状态：开和关；灯也有两种状态：亮和灭。显然两个开关和灯之间形成了

一种逻辑关系，只有当两个开关全部闭合时，灯才会亮，两个开关任意一个断开或者两个同时断开，灯不亮。图 1-3 所示电路显然是开关的状态决定了灯的状态，因此开关的状态是条件，灯的状态是结果，这个电路表示了这样一种逻辑关系：当决定某一事件的条件全部具备时，这一事件才发生；有任一条件不具备，事件就不发生，我们把这种因果控制关系称为“与逻辑”。开关 A、B 和灯 P 的因果关系用表达式来表示，可写成 $P = A \cdot B$。

式中，A 和 B 为逻辑变量，每个变量都可以取 0 和 1 两个值，可以分别代表开关的断开和闭合；P 为与逻辑函数，也可以取 0 和 1 两个值代表灯的灭和亮，P 的结果由 A 和 B 决定。根据电路中的逻辑关系可以看出，只有当 A、B 均为 1 时，即两个开关都闭合时，灯才会亮，此时 P 为 1，其余情况 P 的取值均为 0，我们可以用一个表格来表示出所有的取值情况，见表 1-3。

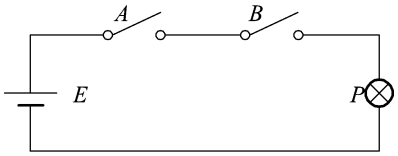


图 1-3　与逻辑示意图

表 1-3　与逻辑真值表

A	B	P
0	0	0
0	1	0
1	0	0
1	1	1

如表 1-3 所示，这种将逻辑变量（用字母 A、B、C…来表示）的各种可能的取值（每个逻辑变量只能有 0 与 1 两种取值）和相应的函数值 P 排列在一起所组成的表叫作真值表。

与逻辑有多个输入变量时可写成

$$P = A \cdot B \cdot C \cdots \tag{1-1}$$

式（1-1）称为与逻辑表达式。符号“·”读作“与”或“乘”，在不致混淆的情况下，“·”可省略，写成 $P = AB$。

在逻辑代数中，“与逻辑”也称作“与运算”或“逻辑乘”（Logic Multplication）。

与逻辑的基本运算规则为

$0 \cdot 0 = 0$　　$0 \cdot 1 = 0$　　$1 \cdot 0 = 0$　　$1 \cdot 1 = 1$

与逻辑的运算规则可归纳为：有 0 得 0，全 1 得 1。

在数字电路中表示与逻辑的电路叫与门，用图 1-4 所示符号表示。

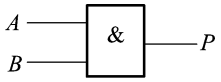


图 1-4　与逻辑符号

2. 或运算

微课 或逻辑及或门　　动画 或逻辑及或门

如图 1-5 所示，指示灯控制电路同样是两个开关 A 和 B 来控制灯 P 的亮和灭，每个开关都有两种状态：开和关；灯也有两种状态：亮和灭。但是这两个开关是并联的，两个开关和灯之间形成了另外一种逻辑关系，当两个开关有任意一个闭合时，灯就会亮，只有当两个开关都断开时，灯不亮。这个电路表示了第二种逻辑关系：在决定某一事件的各条件中，只要具备一个以上的条件，这一事件就会发生；条件全部不具备时，事件不发生。这种因果控制关系称为“或逻辑”。

我们把开关 A、B 和灯 P 的因果关系用表达式来表示，可写成：$P = A+B$ 。

我们也可以列出或逻辑的真值表，见表 1-4。

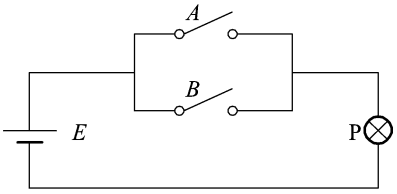


图 1-5　或逻辑示意图

表 1-4　或逻辑真值表

A	B	P
0	0	0
0	1	1
1	0	1
1	1	1

当有多个输入变量时：

$P = A + B + C$（1-2）

符号“+”表示“或逻辑”，也称为“或运算”或“逻辑加”，读作“或”或者“加”。

或逻辑的基本运算规则为

0+0 = 0　　0+1 = 0　　1+0 = 0　　1+1 = 1

显然，或逻辑的运算规则可归纳为：有 1 得 1，全 0 得 0。

必须指出的是，二进制运算和逻辑代数有本质的区别，二者不能混淆。

（1）二进制运算中的加法、乘法是数值的运算，所以有进位问题，如 1 + 1 = 10。

（2）“逻辑或”研究的是“0”“1”两种逻辑状态的逻辑加，所以 1 + 1 = 1。

或门的逻辑符号如图 1-6 所示。

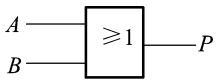


图 1-6　或逻辑符号

3. 非逻辑

微课 非逻辑及非门　　动画 非逻辑及非门

图 1-7 所示为非逻辑的示意图，当开关 S 闭合时，灯不亮，当开关 S 断开时，灯亮。灯亮以开关 S 不闭合为条件。由此电路可以得出一种逻辑关系：只要某一条件具备了，结果便不发生；而此条件不具备时，结果一定发生。这种逻辑关系称为“非逻辑”，用变量 A 代表开关 S 的状态，非逻辑表达式可写成：$P=\bar{A}$。

式中，“ – ”表示“非逻辑”，也称为“非运算”，读作“非”或者“反”。

非逻辑真值表见表 1-5。

表 **1-5**　非逻辑真值表

A	P
0	1
1	0

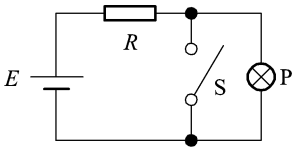


图 1-7　非逻辑示意图

非逻辑的基本运算规则为

$\bar{0}=1$　　$\bar{1}=0$

非门的逻辑符号如图 1-8 所示。

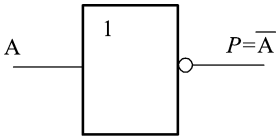


图 1 – 8　非门逻辑符号

二、逻辑代数的复合运算

实际的逻辑问题往往比“与”“或”“非”复杂得多，但是都可以用“与”“或”“非”组合实现，这类逻辑统称为“复合”逻辑，如“与非”“或非”“与或非”“异或”“同或”等，如表 1-6 所示。

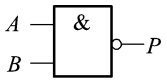
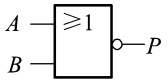
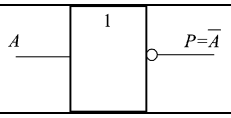
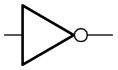
微课 复合逻辑门

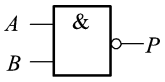
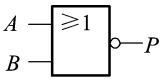
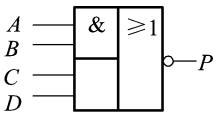
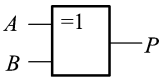
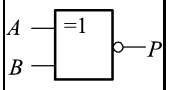
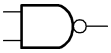
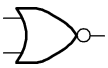
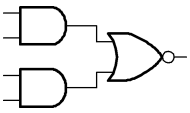
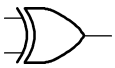
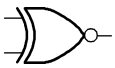
表 1-6　复合逻辑关系

名称	与非门	或非门	与或非门	异或门	同或门
逻辑表达	$P=\overline{ABC}$	$P=\overline{A+B+C}$	$P=\overline{AB+CD}$	$P=A\oplus B$ $=\overline{A}B+A\overline{B}$	$P=A\mathrm{e}\ B$ $=AB+\overline{A}\overline{B}$
逻辑口诀	有 0 得 1 全 1 得 0	全 0 得 1 有 1 得 0	先与再或后非	相异得 1 相同得 0	相同得 1 相异得 0

国际通用符号和我们常用逻辑符号不一样，为了方便使用，将常见的逻辑符号在表 1-7 中列出，以便对照。

表 **1-7**　常用逻辑门符号对照表

	“与”逻辑	“或”逻辑	“非”逻辑
国家标准			
通用符号			

	与非逻辑	或非逻辑	与或非逻辑	异或逻辑	同或逻辑
国家标准					
通用符号					

三、逻辑函数及其表示方法

1. 逻辑函数的概念

图 1-3 所描述的与逻辑电路，由两个开关 A 和 B 来控制灯 P 的亮和灭，每个开关都有两种状态：开和关；灯也有两种状态：亮和灭。显然两个开关和灯之间形成了一种逻辑关系，如果我们把这种逻辑关系用代数式的形式表示出来，就是一个逻辑代数式。

就像普通代数一样，我们可以定义自变量和变量，根据之前的分析，灯的状态由开关的状态决定，因此 P 是关于 A 和 B 的函数，可以用表达式 $P=f(A，B)$ 来表示，这就是逻辑函数，这个电路的逻辑函数表达式就是与逻辑表达式：$P=AB$。

任何一个具体事物的因果关系都可以用一个逻辑函数来描述，比如一个三人裁判电路，其中一个主裁判，两个副裁判，规定一个主裁判在内的两个裁判同意则通过。定义三个裁判的意见为 A、B、C，令 A 为主裁判，B、C 为副裁判，1 和 0 分别代表同意和不同意，裁决结果为 F，1 和 0 分别代表通过和不通过，裁决结果是由三个裁判

的意见决定，因此 F 可以表示为关于 A、B、C 的逻辑函数。

$$F=f(A,B,C)$$

2. 逻辑函数的表示方法

表示逻辑函数的方法有真值表、逻辑函数式、逻辑电路图和卡诺图四种方法，这里只讲解前三种，卡诺图在本项目稍后讲解。

1）真值表

仍以前面裁判电路为例，A 为主裁判，分别用 1 和 0 代表其同意和不同意；B、C 为副裁判，也可以用 1 和 0 来代表其同意和不同意；F 为裁决结果，分别用 1 和 0 代表裁决通过和不通过。根据逻辑关系可以列出真值表，见表 1-8。

表 1-8　裁判电路真值表

A	B	C	F
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

从真值表中可以看出裁决结果和裁判的意见之间的逻辑关系。

2）逻辑函数式

在前述裁判电路中，两个副裁判任意一个同意都可以，但是同时要主裁判同意，因此主裁判的意见和任意一个副裁判的意见（这里是或逻辑）形成与逻辑函数，则根据与逻辑和或逻辑的概念可以得到

$$F=A(B+C)$$

3）逻辑电路图

将逻辑函数式中的与、或、非等各种逻辑关系用相应的逻辑符号表示出来就可以得到逻辑电路图，式 $F=A(B+C)$ 中，B 和 C 是与逻辑关系，可以画在同一个与逻辑门的输入端；它们与的结果和 A 形成了或逻辑，后面再接一个或门即可，逻辑电路如图 1-9 所示。

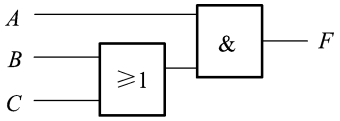


图 1-9　逻辑电路

逻辑函数的各种表达方式之间是可以相互转换的。

任务四　逻辑函数的化简

一、逻辑函数的基本公式和规则

微课 逻辑代数的基本定律及基本规则

1. 基本公式

表 1-9 给出了逻辑代数中最基本的定律，这些定律反映了逻辑代数运算的基本规律，可以作为公式使用，均可用真值表一一加以验证。

表 1-9　逻辑代数基本规律

定律名称	逻辑代数表达式
重迭律	$A+A=A$；　$A\cdot A=A$
交换律	$A\cdot B=B\cdot A$；　$A+B=B+A$
互补律	$A\cdot\overline{A}=0$；　$A+\overline{A}=1$
0-1 律	$A\cdot 1=A$；　$A\cdot 0=0$；　$A+1=1$；　$A+0=A$
结合律	$A\cdot(B\cdot C)=(A\cdot B)\cdot C$　$A+(B+C)=(A+B)+C$
分配律	$A\cdot(B+C)=AB+AC$；　※ $A+BC=(A+B)+(A+C)$
吸收律	$(A+B)(A+\overline{B})=A$；　$AB+A\overline{B}=A$；　$A+AB=A$ ※ $A(A+B)=A$；　$A(\overline{A}+B)=AB$；　※ $A(\overline{A}+B)=AB$
多余项吸收律（消除冗余项）	※ $(A+B)(\overline{A}+C)(B+C)=(A+B)(\overline{A}+C)$ ※ $AB+\overline{A}C+BC=AB+\overline{A}C$
反演律（狄·摩根定律）	※ $\overline{AB}=\overline{A}+\overline{B}$　※ $\overline{A+B}=\overline{A}\cdot\overline{B}$
否定律	※ $\overline{\overline{A}}=A$

注：※表示在逻辑代数中特有的定律，使用时需要特别引起注意。

基本公式都可以用真值表或基本逻辑关系来证明。

【例 1-10】 证明 $A\cdot B=B\cdot A$。

表 1-10　例 1-10 真值表

A　B	$A\cdot B$	$B\cdot A$
0　0	0	0
0　1	0	0
1　0	0	0
1　1	1	1

【例 1-11】 证明 $A\cdot(B+C)=AB+AC$。

表 1-11　例 1-11 真值表

A	B	C	$A\cdot(B+C)$	$AB+AC$
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	0
1	0	1	1	1
1	1	0	1	1
1	1	1	1	1

由真值表可以看出，两个表达式相等。

【例 1-12】 证明反演律：$\overline{AB}=\overline{A}+\overline{B}$。

表 1-12　例 1-12 真值表

A	B	$\overline{AB}$	$\overline{A}$	$\overline{B}$	$\overline{A}+\overline{B}$
0	0	1	1	1	1
0	1	1	1	0	1
1	0	1	0	1	1
1	1	0	0	0	0

【例 1-13】 证明 $AB+A\overline{B}=A$。

解

$$
\begin{aligned}
&AB+A\overline{B}\\
&=A(B+\overline{B})\\
&=A
\end{aligned}
$$

【例 1-14】 证明 $A+\overline{A}B=A+B$。

解

$$
\begin{aligned}
&A+\overline{A}B\\
&=A+AB+\overline{A}B\\
&=A+(A+\overline{A})B\\
&=A+B
\end{aligned}
$$

在进行逻辑函数化简时，可以使用这些基本公式。

2. 基本规则

在逻辑代数中有三个重要规则，依据规则能用已知公式推出更多的公式，为公式法化简提供便利。

1）代入规则

在任何一个含有变量 A 的等式中，如果将所有出现变量 A 的地方都用逻辑函数 F 来取代，则等式仍然成立，此规则称为代入规则。

利用代入规则可扩大等式的应用范围。

例如在 $AB+A\overline{B}=A$ 中，用 $F=AC$ 来代替所有的 A，则可得

$$ABC+A\overline{B}C=AC$$

2）反演规则

对逻辑函数 P 取“非”（即求其反函数 $\overline{P}$ $\overline{P}$ ）称为“反演”。运用反演规则可很方便地求出反函数。

反演规则规定：

将逻辑函数 P 中所有的：

$$P\begin{cases}\text{运算符}\begin{cases}\cdot \rightarrow +\\ + \rightarrow \cdot\end{cases}\\ \text{变量}\begin{cases}\text{原变量} \rightarrow \text{反变量}\\ \text{反变量} \rightarrow \text{原变量}\end{cases}\\ \text{常量}\begin{cases}0 \rightarrow 1\\ 1 \rightarrow 0\end{cases}\end{cases} \xrightarrow{\text{置换后}} \overline{P}$$

利用反演规则求反函数十分方便，但要注意两点：

（1）置换时要保持原式中的运算顺序；

（2）不在“单个”变量上面的“非”号应保持不变。

【例 1-15】 已知 $P=\overline{A}\overline{B}+CD$，求 $\overline{P}$。

根据反演规则直接求出 $\overline{P}=(A+B)(\overline{C}+\overline{D})$。

反演规则实际上是反演律的推广，但反演规则更广泛、更方便。

3）对偶规则

设 P 是一个逻辑函数表达式，将 P 中所有的：

$$P\begin{cases}\text{运算符}\begin{cases}\cdot \rightarrow +\\ + \rightarrow \cdot\end{cases}\\ \text{常量}\begin{cases}0 \rightarrow 1\\ 1 \rightarrow 0\end{cases}\end{cases} \xrightarrow{\text{置换后}} P^*$$

置换后，得到一个新的逻辑函数表达式 P^*，P^* 就是 P 的对偶式。

在置换时要注意两点：

（1）保持原式中的运算顺序；

（2）P 的对偶式 P^* 没有变量的变换，与反函数 $\overline{P}$ 不同。

【例 1-16】 已知 $P=A\cdot\overline{B}+A(C+0)$，求 P^*。

解 $P^*=(A+\overline{B})(A+C\cdot 1)$

值得注意的是，如果两个逻辑函数 P 和 G 相等，那么它们的对偶式 P^* 和 G^* 必相等，这就是对偶规则。

例如，$A+\overline{A}B=A+B$ 成立，则它的对偶式 $A(\overline{A}+B)=AB$ 也成立。

二、逻辑函数的公式化简法

微课 逻辑函数的公式法化简

对于同一个逻辑函数，其表达式不是唯一的，表达式越简单，它表示的逻辑关系越明显，实现时所需的电子器件就越少，这样既可以降低成本，又可以减少故障源，这就是逻辑函数化简的意义。

一般来说，逻辑函数化简的目标如下：

（1）函数式中进行或运算的项不能再减少；

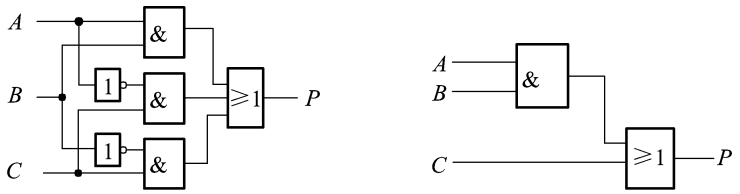
（2）各项中进行与运算的因子也不能再减少。

此时函数式就是一个最简单的与-或表达式。

例如，对于逻辑函数式：

$$
\begin{aligned}
P &= AB + \overline{A}C + \overline{B}C \\
P &= AB + (\overline{A} + \overline{B})C \\
&= AB + \overline{AB}C \\
&= AB + C
\end{aligned}
$$

逻辑函数化简前后的电路图如图 1-10 所示。



（a）化简前　　（b）化简后

图 1-10　化简前后的电路图

1. 逻辑函数的形式

逻辑函数的形式各种各样，但是使用最多的是与-或表达式，因此化简的时候常常把逻辑函数化成与-或表达式的形式，也可以根据电路和实际情况的需要化简成其他形式。

常见逻辑函数表达式的形式见表 1-13。

表 1-13　逻辑函数的形式

表达式	类型
$F = AB + \overline{A}C$	与或型
$F = \overline{A\overline{B} + \overline{A}\overline{C}}$	与或非型
$F = \overline{\overline{AB} \cdot \overline{\overline{A}C}}$	与非与非型
$F = (\overline{A} + B)(A + C)$	或与型
$F = \overline{\overline{\overline{A} + B} + \overline{A + C}}$	或非或非型

它们之间可以相互转换，例如：

$$F = AB + \overline{A}C = \overline{\overline{AB + \overline{A}C}} = \overline{\overline{AB} \cdot \overline{\overline{A}C}}$$

2. 逻辑函数的化简方法

公式化简法就是运用逻辑代数的基本定律和常用公式化简逻辑函数，是最常用的化简法之一，要求在熟练掌握逻辑函数的基本定律和基本公式的基础上进行。

逻辑函数化简的目标是：逻辑函数中“与”项数最少，每个“与”项中变量数最少，则逻辑门数最少，需要的集成电路数量最少，且集成电路之间连线最少。

常用的公式化简方法如下：

（1）并项法：利用公式 $A+\overline{A}=1$，将两项合并，并消去一个变量。

【例 1-17】 试用并项法化简下列逻辑函数式为与-或逻辑函数表达式。

① $L_1 = \overline{A}\overline{B}C + \overline{A}\overline{B}\overline{C}$

② $L_2 = A(BC + \overline{B}\overline{C}) + A(B\overline{C} + \overline{B}C)$

解 ① $L_1 = \overline{A}\overline{B}(C + \overline{C}) = \overline{A}\overline{B}$

② $L_2 = ABC + A\overline{B}\overline{C} + AB\overline{C} + A\overline{B}C$

$= AB(C + \overline{C}) + A\overline{B}(C + \overline{C})$

$= A(B + \overline{B}) = A$

（2）吸收法：利用公式 $A + AB = A$，吸收多余的项（冗余项），根据代入规则，A、B 还可以是任何一个复杂的逻辑式。

【例 1-18】 试用吸收法化简下列逻辑函数。

① $Y_1 = \overline{AB} + \overline{A}D + \overline{B}E$

② $Y_2 = AB + \overline{A}C + BC$

解 ① $Y_1 = \overline{A} + \overline{B} + \overline{A}D + \overline{B}E$

$= \overline{A} + \overline{B}$

② $Y_2 = AB + \overline{A}C + (A + \overline{A})BC$

$= AB + \overline{A}C + ABC + \overline{A}BC$

$= AB(1 + C) + \overline{A}C(1 + B)$

$= AB + \overline{A}C$

（3）消元法：利用公式 $A + \overline{A}B = A + B$，消去多余的变量。

【例 1-19】 试用消元法化简下列逻辑函数。

① $L_1 = AB + \overline{A}C + \overline{B}C$

② $L_2 = \overline{AB} + AC + BD$

解 ① $L_1 = AB + \overline{A}C + \overline{B}C$

$= AB + \overline{AB}C$

$= AB + C$

② $L_2 = \overline{A} + \overline{B} + AC + BD$

$= \overline{A} + \overline{B} + C + D$

（4）配项法：当不能直接利用基本定律时，可先利用基本定律配项后再化简。

【例 1-20】 试用配项法化简下列逻辑函数表达式。

① $L = AB + \overline{A}\overline{C} + B\overline{C}$

② $F = AB\overline{C} + (\overline{A} + C)D + BD$

解 ① $= AB + \overline{A}\overline{C} + AB\overline{C} + \overline{A}B\overline{C}$

$= (AB + AB\overline{C}) + (\overline{A}\overline{C} + \overline{A}\overline{C}B)$

$= AB + \overline{A}\overline{C}$

解 ② $F = AB\overline{C} + \overline{A}\overline{C}D + BD$

$= AB\overline{C} + \overline{A}\overline{C}D$

下面来做一组化简练习。

【例 1-21】 化简 $Y = ACE + \overline{A}BE + \overline{B}\overline{C}\overline{D} + B\overline{C}E + D\overline{C}E + \overline{A}E$ 。

解 $Y = E\,(AC + \overline{A}B + B\overline{C} + D\overline{C} + \overline{A}) + \overline{B}\,\overline{C}\,\overline{D}$

$= E\,(C + B + D + \overline{A}) + \overline{B}\overline{C}\overline{D}$（吸收法、消元法）

$= E\,(B + C + D) + \overline{A}E + \overline{B}\,\overline{C}\,\overline{D}$（分配律）

$= E\,\overline{\overline{B}\overline{C}\overline{D}} + \overline{A}E + \overline{B}\overline{C}\overline{D}$（反演律）

$= E + \overline{A}E + \overline{B}\overline{C}\overline{D}$（消元法）

$= E + \overline{B}\,\overline{C}\,\overline{D}$

【例 1-22】 化简 $L = AD + A\overline{D} + AB + \overline{A}C + BD + A\overline{B}EF + \overline{B}EF$

解 $L = A + AB + \overline{A}C + BD + A\overline{B}EF + \overline{B}EF$（利用 $A + \overline{A} = 1$）

$= A + \overline{A}C + BD + \overline{B}EF$（利用 $A + AB = A$）

$= A + C + BD + \overline{B}EF$（利用 $A + \overline{A}B = A + B$）

思考：化简 $F = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$ 。

逻辑函数化简的途径并不是唯一的，上述方法可以任意组合或综合运用。

三、逻辑函数的卡诺图化简法

1. 最小项

微课 最小项与逻辑函数　　动画 最小项

所谓最小项是这样一个乘积项：在该乘积项中含有输入逻辑变量的全部变量，每个变量以原变量或反变量的形式出现且仅出现一次。

对于包含 n 个变量的函数来说，共有 2^n 个不同取值组合，所以有 2^n 个最小项。对于三变量 A、B、C 来讲，有 $2^3=8$ 个最小项，分别为：$\bar{A}\bar{B}\bar{C}$、$\bar{A}\bar{B}C$、$\bar{A}B\bar{C}$、$\bar{A}BC$、$A\bar{B}\bar{C}$、$A\bar{B}C$、$AB\bar{C}$ 和 ABC。例如，A、B、C 变量取 0、1、0 时，对应的最小项为 $\bar{A}B\bar{C}$；A、B、C 取 1、0、1 时，对应的最小项为 $A\bar{B}C$ …共 8 组。表 1-14 列出了三变量的所有最小项。

表 1-14　三变量对应的最小项

A	B	C	对应最小项（m_i）
0	0	0	$\bar{A}\bar{B}\bar{C}=m_0$
0	0	1	$\bar{A}\bar{B}C=m_1$
0	1	0	$\bar{A}B\bar{C}=m_2$
0	1	1	$\bar{A}BC=m_3$
1	0	0	$A\bar{B}\bar{C}=m_4$
1	0	1	$A\bar{B}C=m_5$
1	1	0	$AB\bar{C}=m_6$
1	1	1	$ABC=m_7$

为表达和书写方便，通常用“m_i”表示最小项，下标 i 代表由输入变量组合的二进制数所对应的十进制数。若变量取值组合为 $A\bar{B}C=101$，则有 $1\times2^2+0\times2^1+1\times2^0=5$ 来表示对应的最小项 $A\bar{B}C$，记作 m_5。

2. 逻辑函数的最小项表达式

由若干个最小项相加而构成的“与-或”表达式被称为最小项表达式，也是“与-或”表达式的标准形式，且是唯一的。

例如：　$P=ABC+AB\bar{C}+\bar{A}BC+\bar{A}\bar{B}C$

可以简写成：$P(A、B、C)=m_7+m_6+m_3+m_1=\sum m(1,3,6,7)$

逻辑函数展开成最小项之和的表达式形式，其变换方法有两种。

（1）由逻辑函数列出真值表，再写出最小项表达式。

【例 1-23】 将 $P=A\bar{B}+AB\bar{C}$ 展开成最小项表达式。

依题意，列出其真值表，如表 1-15 所示，再由真值表写出最小项表达式为

$$P=A\bar{B}C+A\bar{B}\bar{C}+AB\bar{C}=m_4+m_5+m_6=\sum m(4,5,6)$$

表 1-15　例 1-23 的真值表

A	B	C	P
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

（2）由逻辑函数利用公式法去反、脱括号、配项后写成最小项表达式。

【例 1-24】 将函数 $P=\overline{(AB+\overline{A}\overline{B}+\overline{C})\overline{AB}}$ 展开成最小项表达式

解 第一步“去反”：原式 $=\overline{AB+\overline{A}\overline{B}+\overline{C}}+AB$

$$\overline{AB+\overline{A}\overline{B}+\overline{C}}+AB$$

$$=(\overline{A}+\overline{B})(A+B)C+AB$$

第二步“脱括号”：$=(\overline{A}B+A\overline{B})C+AB$

$$=\overline{A}BC+A\overline{B}C+AB$$

第三步“配项”：$=\overline{A}BC+A\overline{B}C+AB(C+\overline{C})$

$$=\overline{A}BC+A\overline{B}C+ABC+AB\overline{C}$$

$$=m_5+m_3+m_7+m_6=\sum m(3,5,6,7)$$

3. 卡诺图

卡诺图是代表逻辑函数的所有最小项的小方块按相邻原则排列而成的方块图。

相邻原则：几何上邻接的小方格所代表的最小项，只有一个变量互为反变量，其他变量都相同。

制作卡诺图，只需将所有逻辑变量分成纵、横两组，且每一组变量取值组合按循环码排列，即相邻两组之间只有一个变量取值不同。例如，两变量的 4 种取值应按 00→01→11→10 排列。要特别注意的是，头、尾两组取值也是相邻的。

下面给出二变量、三变量和四变量卡诺图，如图 1-11 所示。

A \ B	0	1
0	m_0	m_1
1	m_2	m_3

（a）二变量卡诺图

A \ BC	00	01	11	10
0	m_0	m_1	m_3	m_2
1	m_4	m_5	m_7	m_6

（b）三变量卡诺图

AB \ CD	00	01	11	10
00	m_0	m_1	m_3	m_2
01	m_4	m_5	m_7	m_6
11	m_{12}	m_{13}	m_{15}	m_{14}
10	m_8	m_9	m_{11}	m_{10}

（c）四变量卡诺图

图 1-11 卡诺图

1）二变量卡诺图

设输入变量为 A、B（A 是高位、B 是低位），共有 $2^2=4$ 个最小项。有 4 个小方块分别表示二变量的全部 4 个最小项 $m_0\sim m_3$。这 4 个最小项按逻辑相邻的原则排列。

2）三变量卡诺图

设输入变量为 A、B、C（高位→低位），共有 $2^3=8$ 个最小项，共有 8 个小方块分别表示三变量的全部 8 个最小项 $m_0\sim m_7$。将 A 作为纵轴，BC 作为横轴，BC 取值应符合逻辑相邻排列规则。

3）四变量卡诺图

设输入变量为 A、B、C、D（高位→低位），共有 $2^4=16$ 个最小项。有 16 个小方块分别表示四变量的全部 16 个最小项 $m_0\sim m_{15}$。将 AB 作为纵轴、CD 作为横轴，并且

AB、CD 分别按逻辑相邻规律排列，可得到四变量卡诺图。

4. 用卡诺图表示逻辑函数

微课 用卡诺图表示逻辑函数　　动画 卡诺图表示逻辑函数

由于任意一个 n 变量的逻辑函数都能变换成最小项表达式。而 n 变量的卡诺图包含了 n 个变量的所有最小项，所以卡诺图与逻辑函数存在一一对应的关系，n 变量的卡诺图可以表示 n 变量的任意一个逻辑函数。

例如，表示一个三变量的逻辑变量 $F(A、B、C)=\sum m(2,4,5)$，可以在三变量卡诺图的 m_2、m_4、m_5 的小方格中填写 1 来标记，其余各小方格填 0（或者什么也不填），如图 1-12 所示，填“1”格的含义是当函数的变量取值与该小方格代表的最小项相同时，函数值为 1。

A \ BC	00	01	11	10
0	0	0	0	1
1	1	1	0	0

图 1-12　三变量卡诺图

对于一个非标准的逻辑函数表达式（即不是最小项形式），通常是将逻辑函数变换成最小项表达式再填图。

【例 1-25】 完成逻辑函数 $P=A\bar{B}C+\bar{A}BD+AD$ 的卡诺图。

解： 原式 $=A\bar{B}C(D+\bar{D})+\bar{A}BD(C+\bar{C})+AD(B+\bar{B})$

$=A\bar{B}CD+A\bar{B}C\bar{D}+\bar{A}BCD+\bar{A}B\bar{C}D+ABD(C+\bar{C})+A\bar{B}D(C+\bar{C})$

$=A\bar{B}CD+A\bar{B}C\bar{D}+\bar{A}BCD+\bar{A}B\bar{C}D+ABCD+A\bar{B}\bar{C}D$

$=\sum m(5,7,9,10,11,13,15)$

将上述表达式填入卡诺图，如图 1-13 所示。

AB \ CD	00	01	11	10
00				
01		1	1	
11		1	1	
10		1	1	1

图 1-13　例 1-25 卡诺图

5. 卡诺图相邻项的合并

在公式法化简逻辑函数时，利用公式 $AB+A\bar{B}=A$ 将两个乘积项进行合并。该公式表明两个具有“相邻性”的乘积项，相同部分将被保留，而不同部分被吸收。

由于卡诺图是按循环码的规律排列，使处在相邻位置的最小项都只有一个变量相反，因此，凡是处于相邻位置的最小项均可以合并消去相异的变量，2^N 个相邻项可以

被合并在一起，如 2 个相邻项合并、4 个相邻项合并、8 个相邻项合并等。

图 1-14 列出了 2 个相邻项进行合并的例子。

A \ BC	00	01	11	10
0		1	1	
1				

（a）$F = A\overline{C}$

A \ BC	00	01	11	10
0	1			1
1				

（b）$F = \overline{A}\overline{C}$

A \ BC	00	01	11	10
0		1		
1		1		

（c）$F = B\overline{C}$

AB \ CD	00	01	11	10
00				
01				
11				
10		1	1	

（d）$F = A\overline{B}D$

AB \ CD	00	01	11	10
00			1	
01				
11				
10			1	

（e）$F = \overline{B}CD$

AB \ CD	00	01	11	10
00				
01			1	
11			1	
10				

（f）$F = BCD$

图 1-14　2 个相邻项合并

图 1-15 给出了 4 个相邻项合并的例子。

A \ BC	00	01	11	10
0	1	1	1	1
1				

（a）$F = \overline{A}$

A \ BC	00	01	11	10
0	1			1
1	1			1

（b）$F = \overline{C}$

A \ BC	00	01	11	10
0		1	1	
1		1	1	

（c）$F = C$

AB \ CD	00	01	11	10
00				
01				
11				
10	1	1	1	1

（d）$F = A\overline{B}$

AB \ CD	00	01	11	10
00			1	
01			1	
11			1	
10			1	

（e）$F = CD$

AB \ CD	00	01	11	10
00				
01		1	1	
11		1	1	
10				

（f）$F = BD$

AB \ CD	00	01	11	10
00	1			1
01	1			1
11				
10				

（g）$F = \overline{A}\overline{D}$

AB \ CD	00	01	11	10
00		1	1	
01				
11				
10		1	1	

（h）$F = \overline{B}D$

AB \ CD	00	01	11	10
00	1			1
01				
11				
10	1			1

（i）$F = \overline{B}\overline{D}$

图 1-15　4 个相邻项合并