



普通高等教育“十一五”国家级规划教材

普通高等院校

电子信息类系列教材

*Shuzi DianLu Yu
LuoJi SheJi*

数字电路与 逻辑设计

© 徐惠民 安德宁 延明 编著



人民邮电出版社
POSTS & TELECOM PRESS

普通高等教育“十一五”国家级规划教材

普通高等院校电子信息类系列教材

数字电路与逻辑设计

徐惠民 安德宁 延明 编著

人民邮电出版社

北 京

图书在版编目 (CIP) 数据

数字电路与逻辑设计 / 徐惠民, 安德宁, 延明编著.
北京: 人民邮电出版社, 2009. 9
(普通高等院校电子信息类系列教材)
ISBN 978-7-115-19767-2

I. 数… II. ①徐…②安…③延… III. 数字电路—逻辑设计—高等学校—教材 IV. TN79

中国版本图书馆CIP数据核字 (2009) 第055395号

内 容 提 要

本书是普通高等教育“十一五”国家级规划教材。本书系统地介绍了数字电路与逻辑设计的基本概念及分析和设计方法。本书的编写本着经典和现代相结合的原则, 对于经典的概念、原理和方法进行准确的、深入浅出的讲解; 对于现代的数字技术和设计方法, 则是有针对性地选择了一些最新的设计理念和方法。本书普遍采用了逻辑仿真的波形图, 使得分析和设计的结果更有真实感。

全书共分 9 章, 包括数字电路中的数和编码、逻辑代数基础、集成门电路、组合逻辑电路的分析和设计、集成触发器、时序逻辑电路的分析和设计、大规模数字集成电路、数/模和模/数转换、VHDL 描述逻辑电路等内容。

本书可作为高等院校通信、信息、电子工程、计算机、自动化等相关专业的本科教材, 也可供相关专业的研究生和工程技术人员阅读参考。

普通高等教育“十一五”国家级规划教材

普通高等院校电子信息类系列教材

数字电路与逻辑设计

-
- ◆ 编 著 徐惠民 安德宁 延 明
责任编辑 滑 玉
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号
邮编 100061 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 20.25
字数: 495 千字 2009 年 9 月第 1 版
印数: 1—3 000 册 2009 年 9 月河北第 1 次印刷

ISBN 978-7-115-19767-2/TN

定价: 33.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

前 言

“数字电路与逻辑设计”是各高等院校多数工科专业的一门必修的专业基础课。课程的特点可以用“经典”、“现代”来概括。“经典”是指课程的基本概念和基本理论是非常经典的；“现代”是指数字技术是现代发展最快的技术之一，必须以最新的数字技术成果充实到“数字电路与逻辑设计”课程和教材中来，保持课程和教材与现代数字技术的同步推进。

本书的编写就是遵循了经典和现代相结合的原则。对于经典的概念、原理和方法既要保证讲解深入浅出、通俗易懂，同时对于有些传统的说法作了一些修正，使概念能够更加准确。对于一些以前强调不够的内容，如缓冲器、锁存器等，增加了篇幅，以便为后续课程的学习打下良好的基础。本书保持了 1990 年版《数字电路与逻辑设计》文字流畅、易学易懂的风格，力求使读者看得懂，学得会。

在力图使得本书符合现代数字技术发展要求方面，做了以下几方面的工作。首先是引入了硬件描述语言（VHDL）及相应的技术。VHDL 描述是用一章的篇幅来介绍的，内容上是 3 个独立的部分：语言基础、组合电路描述和时序电路描述。在教学使用时可以分散在其他相应章节后面来讲，也可以集中来讲，或者根据实验的需要来选用。本书所使用的 VHDL 例子，全部在 Quartus II 平台上进行了验证，并给出了仿真波形。

其次是引入了逻辑仿真。逻辑仿真不仅仅是重要的设计技术，也是一种现代的学习方法。书中对于许多组合和时序电路都给出了输入和输出的仿真波形，非常有利于读者直观地感知理论分析的结果，并且从理论与实际结果的差别上进一步学习正确的分析方法。

另外，书中还注意引入最新的技术成果。例如，在介绍模数转换器时引入了 $\Sigma-\Delta$ 模数转换器。在介绍可编程逻辑器件时，引入了 altera 公司的 FPGA，并给出了 CPLD 和 FPGA 的比较等。它们既是一些最新的数字技术发展的成果，也是对于课程内容的新的补充。

本书共分 9 章，包括数字电路中的数和编码、逻辑代数基础、集成逻辑门电路、组合电路的分析和设计、集成触发器、时序逻辑电路的分析和设计、大规模数字集成电路、模数和数模转换、VHDL 描述逻辑电路等内容。

本书给出了丰富的例题和习题。每章的习题中都包含相当数量的选择题，读者可以使用这些选择题来复习检查本章所学习的基本概念和基本知识。

本书由徐惠民、安德宁、延明编写。彭家浚、李春宜、徐晶、龚乃绪等也参与了本书编写的有关工作。

由于编者水平有限，书中存在的问题一定不少，恳请广大读者批评指正。

编 者
2009 年 1 月

目 录

第 1 章 数字电路中的数和编码	1	2.5 卡诺图法化简逻辑函数	29
1.1 十进制数的二进制编码	1	2.5.1 卡诺图及其构成	29
1.1.1 有权码和无权码	1	2.5.2 卡诺图化简逻辑函数的	
1.1.2 格雷码	3	基本原理	30
1.1.3 二一十进制码的运算	5	2.5.3 如何将逻辑函数填入	
1.2 用补码表示负数	5	卡诺图	31
1.2.1 补码	6	2.5.4 卡诺图化简的步骤及举例	33
1.2.2 补码加法	6	2.6 硬件描述语言及逻辑仿真	37
本章小结	7	2.6.1 硬件描述语言	38
习题和思考题	7	2.6.2 逻辑仿真	38
第 2 章 数字逻辑基础	9	本章小结	39
2.1 逻辑变量和逻辑函数	9	习题和思考题	39
2.1.1 逻辑变量和逻辑系统	9	第 3 章 集成逻辑门电路	46
2.1.2 基本逻辑运算和布尔		3.1 数字集成电路的发展	46
代数公理	10	3.2 二极管门电路	48
2.1.3 其他常用逻辑运算	12	3.2.1 二极管与门	48
2.2 布尔代数的定律和规则	16	3.2.2 二极管或门电路	49
2.2.1 布尔代数的基本定律	16	3.2.3 正逻辑和负逻辑	50
2.2.2 布尔代数的常用公式	17	3.3 三极管反相器	51
2.2.3 布尔代数的三个规则	19	3.3.1 三极管非门电路	51
2.3 逻辑函数的标准表达式	21	3.3.2 三极管反相器的负载电流	52
2.3.1 逻辑函数的最小项		3.4 TTL 集成逻辑门电路	52
表达式	21	3.4.1 TTL 集成与非门	53
2.3.2 逻辑函数的最大项		3.4.2 TTL 逻辑门的特性参数	55
表达式	23	3.5 其他 TTL 集成门电路	59
2.3.3 最小项表达式和最大项		3.5.1 74 系列集成电路	59
表达式的关系	24	3.5.2 抗饱和 TTL 电路	59
2.3.4 非标准表达式到标准		3.5.3 TTL 或非门电路	60
表达式的转换	25	3.5.4 集电极开路门	61
2.3.5 任意项及其表示	25	3.5.5 TTL 三态门	63
2.4 代数法化简逻辑函数	27	3.6 CMOS 集成电路	65
2.4.1 逻辑函数化简的标准	27	3.6.1 CMOS 反相器	65
2.4.2 代数化简法	27	3.6.2 其他 CMOS 逻辑电路	67

3.6.3 CMOS 漏极开路门和三态门	70	5.3.2 可控 RS 触发器	135
3.6.4 CMOS 传输门	70	5.3.3 其他可控触发器	136
3.6.5 CMOS 集成电路的使用	71	5.3.4 电位型触发器的局限性	138
3.7 ECL 集成电路	72	5.3.5 电位型触发器的应用: 锁存器	139
3.7.1 基本 ECL 门的组成	73	5.4 钟控型触发器	140
3.7.2 ECL 电路的特点	74	5.4.1 主从触发器	140
本章小结	75	5.4.2 边沿触发器	144
习题和思考题	75	5.5 触发器的逻辑符号	146
第 4 章 组合逻辑电路的分析和设计	81	5.6 CMOS 触发器	147
4.1 组合逻辑电路的特点	81	5.6.1 带使能端 D 触发器	147
4.2 组合逻辑电路的分析	82	5.6.2 CMOS 主从 D 触发器	147
4.2.1 组合逻辑电路的分析步骤	82	5.6.3 CMOS JK 触发器	149
4.2.2 组合逻辑电路分析举例	82	5.7 触发器的转换	150
4.3 组合逻辑电路的设计	85	5.8 集成触发器的时间参数	151
4.3.1 组合逻辑电路的设计步骤	85	5.8.1 建立时间和保持时间	151
4.3.2 组合逻辑电路的实现方式	86	5.8.2 时钟信号的时间参数	152
4.3.3 组合逻辑电路设计举例	87	5.9 钟控触发器构成的常用 时序电路	152
4.4 中规模组合逻辑电路	91	5.9.1 寄存器	152
4.4.1 加法器和减法器	91	5.9.2 移位寄存器	153
4.4.2 编码器	93	5.9.3 计数器	153
4.4.3 译码器	97	本章小结	154
4.4.4 数据选择器	104	习题和思考题	155
4.4.5 数值比较器	107	第 6 章 时序逻辑电路的分析和设计	160
4.4.6 奇偶校验器/发生器	110	6.1 时序电路的分类和描述	160
4.4.7 中规模组合电路用于 逻辑设计	112	6.1.1 时序电路的一般分类	160
4.5 组合逻辑电路的竞争和 冒险	117	6.1.2 同步时序电路的分类	161
4.5.1 冒险的分类	117	6.1.3 同步时序电路的描述	162
4.5.2 冒险的识别和消除	120	6.2 常用同步时序电路的分析	163
本章小结	123	6.2.1 同步时序电路分析的 步骤	163
习题和思考题	124	6.2.2 同步计数器的分析	164
第 5 章 集成触发器	130	6.2.3 移位寄存器及其应用 电路的分析	166
5.1 时序逻辑电路的特点	130	6.3 常用时序电路的设计	172
5.2 触发器的基本特性及其 记忆作用	131	6.3.1 基本的设计步骤	172
5.3 电位型触发器	132	6.3.2 同步计数器的设计	172
5.3.1 基本 RS 触发器	132	6.3.3 序列信号发生器	176

6.3.4 M 序列发生器	178	7.5 通用阵列逻辑	241
6.4 异步计数器	180	7.5.1 GAL 和 PAL 的区别	241
6.4.1 异步计数器的基本形式	180	7.5.2 输出逻辑宏单元	241
6.4.2 异步计数器的分析	181	7.5.3 GAL 芯片示例	244
6.5 中规模时序集成电路	184	7.6 复杂可编程逻辑器件	245
6.5.1 中规模集成计数器	184	7.6.1 CPLD 器件的基本体系 结构	245
6.5.2 中规模计数器的应用	189	7.6.2 CPLD 器件结构举例	246
6.5.3 中规模移位寄存器	197	7.6.3 宏单元的构成	247
6.5.4 中规模移位寄存器的 应用	199	7.6.4 PIA 和 I/O 控制块	248
6.6 一般时序电路的分析	205	7.6.5 CPLD 产品举例	249
6.6.1 一般时序电路的特点	205	7.7 现场可编程门阵列	250
6.6.2 一般时序电路分析举例	205	7.7.1 FPGA 芯片的基本结构	250
6.7 一般时序电路的设计	208	7.7.2 Altera 公司 FPGA 芯片 基本结构	250
6.7.1 设计步骤	208	7.7.3 逻辑阵列块	251
6.7.2 状态表的建立	208	7.7.4 逻辑单元	252
6.7.3 状态表的简化	211	7.7.5 嵌入式 RAM 块	253
6.7.4 状态分配	213	7.7.6 输入输出单元	254
本章小结	216	7.7.7 FPGA 芯片的编程	255
习题和思考题	216	7.7.8 FPGA 产品举例	255
第 7 章 大规模数字集成电路	224	7.8 CPLD 和 FPGA 的比较	256
7.1 大规模数字集成电路概述	224	本章小结	258
7.1.1 大规模集成电路的分类	224	习题和思考题	258
7.1.2 专用集成电路的分类	225	第 8 章 数模和模数转换	261
7.1.3 可编程逻辑器件及其 发展	226	8.1 数模转换器	261
7.1.4 PLD 的分类	227	8.1.1 数模转换器的基本要求	261
7.1.5 PLD 的性能特点	228	8.1.2 数模转换器的主要参数	262
7.2 存储器	229	8.1.3 数模转换器的一般框图	263
7.2.1 随机存储器	229	8.1.4 权电阻网络数模转换器	264
7.2.2 只读存储器	231	8.1.5 倒 T 形网络数模转换器	265
7.2.3 ROM 作为逻辑器件	233	8.1.6 权电流源网络数模 转换器	267
7.2.4 存储器容量的扩展	234	8.1.7 单电流源网络数模 转换器	268
7.3 可编程逻辑阵列	236	8.1.8 数模转换器的选用	269
7.3.1 PLA 结构的特点	237	8.2 模数转换器	269
7.3.2 用 PLA 设计逻辑电路	237	8.2.1 模数转换的一般过程	269
7.4 可编程阵列逻辑	238	8.2.2 模数转换器的主要参数	270
7.4.1 PAL 的逻辑结构	238		
7.4.2 PAL 芯片示例	240		

8.2.3 逐次比较型模数转换器·····	272	9.5 数据流描述·····	292
8.2.4 双积分型模数转换器·····	273	9.5.1 并行赋值语句·····	292
8.2.5 并联比较型模数转换器·····	274	9.5.2 条件赋值语句·····	293
8.2.6 Σ - Δ 模数转换器·····	275	9.5.3 选择信号赋值语句·····	294
8.2.7 流水线型模数转换器·····	279	9.6 性能描述·····	295
8.2.8 模数转换器的选用·····	280	9.6.1 PROCESS 语句·····	296
本章小结·····	280	9.6.2 信号和变量赋值语句·····	296
习题和思考题·····	281	9.6.3 分支语句·····	296
第9章 VHDL 描述逻辑电路·····	283	9.6.4 循环语句·····	298
9.1 基于硬件描述语言的设计		9.7 结构描述·····	299
过程·····	283	9.7.1 部件声明语句·····	299
9.1.1 VHDL 的基本特点·····	283	9.7.2 部件描述语句·····	300
9.1.2 设计过程·····	284	9.8 VHDL 描述组合逻辑电路·····	301
9.1.3 Quartus II 设计软件·····	284	9.8.1 译码电路的描述·····	301
9.2 VHDL 描述的基本结构·····	285	9.8.2 编码器的描述·····	302
9.2.1 实体描述·····	285	9.9 触发器的 VHDL 描述·····	304
9.2.2 结构体描述·····	286	9.9.1 电位型触发器的 VHDL	
9.2.3 VHDL 的 3 种描述·····	287	描述·····	304
9.3 数据类型、运算符和表达式·····	287	9.9.2 钟控型触发器的 VHDL	
9.3.1 枚举类型·····	287	描述·····	304
9.3.2 数组类型·····	288	9.10 时序部件的 VHDL 描述·····	307
9.3.3 VHDL 运算符·····	289	9.10.1 计数器的 VHDL 描述·····	307
9.3.4 常量的定义·····	290	9.10.2 移位寄存器的 VHDL	
9.3.5 VHDL 表达式·····	290	描述·····	309
9.4 VHDL 的库和包·····	291	9.10.3 时序机的 VHDL 描述·····	310
9.4.1 VHDL 库的种类和使用·····	291	本章小结·····	311
9.4.2 程序包·····	291	习题和思考题·····	312
9.4.3 库和程序包的引用·····	292	参考文献·····	316

第 1 章 数字电路中的数和编码

数字电路中要用到不同数制中的数，如二进制数、十进制数等。对于这些不同数制的数以及它们之间的互相转换，应该在计算机基础课程中有详细介绍，这里不再重复。

数字电路中要使用不同的二进制代码来表示十进制数，需要掌握相应的编码知识。此外，数字电路中还要用到负数在机器中的表示方法，以便方便地构成加法器/减法等最基本的运算电路。通过本章学习，应掌握：

- 十进制数的二进制编码；
- 用补码表示负数及其相应的运算。

1.1 十进制数的二进制编码

数字电路和计算机中最常用的是二进制数，有时候也会使用十进制数。但是 0~9 这样的十进制数在二值的数字电路中是不能直接输入和运算的，需要用某种二进制代码表示以后才可以操作。

用一组特定的二进制代码来表示一组特定的事物，称为二进制编码。一组字符、一组颜色、一批货物、一队参赛队员，等等，都可以是二进制编码的对象。

对于 0~9 来说，每个数字需要用 4 位二进制代码来表示。这种用二进制代码表示的十进制数，称为二—十进制编码（Binary-Coded-Decimal），简称为 BCD 码。

1.1.1 有权码和无权码

4 位二进制代码有 0000~1111 共 16 种组合。从 16 种代码组合中取出 10 种作为 BCD 码有许多方案。这些编码方案可以分为两种类型：有权码和无权码。

1. 有权码

如果二进制代码的每一位都有固定的权值，这样的代码称为有权码。

如果 4 位有权码用 $b_3b_2b_1b_0$ 表示，每一位的权值用 $w_3w_2w_1w_0$ 表示，则有权码所代表的十进制数 N 可以写为：

$$N = \sum_{i=0}^3 b_i w_i \quad (1-1)$$

对于有权码,通常就使用其权值 $w_3w_2w_1w_0$ 作为编码的名称。如果权值 $w_3w_2w_1w_0$ 分别是 8、4、2、1,这样的编码就称为 8421BCD 码,或者直接称为 8421 码。具体的编码如表 1-1 所示。

如果已经知道具体的代码,则使用式 (1-1) 就可以计算出它所代表的数值。例如,8421 码中的 1001 代码所代表的十进制数是:

$$N = 1 \times 8 + 0 \times 4 + 0 \times 2 + 1 \times 1 = 9$$

有权码的种类也是很多的。例如 5421 码、2421 码、7421 码等。如果不同位的权值相同,如 2421 码,有两位的权值是 2;或者若干位权值的和等于另一个权值,如 5421 码,4+1 和 5 相同,这样的有权码的具体编码方案就不是唯一的。也就是说,某一个十进制数所对应的编码可以不止一种。例如,对于 2421 码来说,数值 3 的编码可以是 0011,也可以是编码 1010。而对于 5421 码而言,数值 7 可以是编码 1010,也可以是编码 0111。在表 1-1 中,只列出了它们的一种编码方案。

2. 余三码

如果二—十进制编码的每一位没有固定的权值,这样的编码称为无权码。

最典型的无权码之一就是余三码。对于同样的十进制数值,它的余三码是相应数值的 8421 码加 3 而获得的。例如,数值 9 的 8421 码是 1001,相应的余三码就是在 1001 的基础上再加 0011。所以,数值 9 的余三码就是 1100。

8421 码的范围是从 0000~1001。余三码的范围就是从 0011~1100。

余三码也列出在表 1-1 中。

表 1-1 几种常见的二—十进制码

十进制数	8421 码	2421 码	7421 码	余三码
0	0000	0000	0000	0011
1	0001	0001	0001	0100
2	0010	0010	0010	0101
3	0011	0011	0011	0110
4	0100	0100	0100	0111
5	0101	0101	0101	1000
6	0110	0110	0110	1001
7	0111	0111	1000	1010
8	1000	1110	1001	1011
9	1001	1111	1010	1100

虽然余三码很有规律,但它却属于无权码。因为余三码不可能为每一位代码指定一个固定的权值,因此也不能用式 (1-1) 计算余三码所代表的数值。

例 1.1 将十进制数 548 用 8421 码和余三码表示。

解: 要将十进制数中的每一位数,用相应的二—十进制码表示。对于十进制数 548 来说,要用 BCD 码分别表示 5、4 和 8。

$$\begin{aligned}(548)_{10} &= (010101001000)_{8421} \\ &= (100001111011)_{\text{余三码}}\end{aligned}$$

例 1.2 将 7421 码 100100111000 所表示的十进制数，用 8421 码表示出来。

解：先写出 7421 码所表示的十进制数，再将这个十进制数的每一位用 8421 码表示。

$$\begin{aligned}(100100111000)_{7421} &= (837)_{10} \\ &= (100000110111)_{8421}\end{aligned}$$

1.1.2 格雷码

格雷码本身并不是二—十进制码。格雷码的长度并没有限制，可以是 2 位格雷码、3 位格雷码等。它是二进制编码的一种。

在一组二进制代码中，两个相邻代码对应位置上不同码值的数目称为码距。例如，代码 010 和 100 的码距为 2。

在实际应用中，有时需要一种码距为 1 的代码，也就是一组代码中，所有相邻的代码的码距都是 1。

每一位代码，在实际的电路中都会对应一个物理的器件，代码转换就是物理器件的状态转换。码距大于 1 的代码转换时，由于物理器件状态转换时间上的差别，可能产生暂时的过渡状态。例如从 0111 转换到 1000，4 个物理器件都要改变状态，如果最高位变的最快，就会出现过渡状态 1111。而对于码距为 1 的代码，相邻代码转换时，就不会出现这样的问题。

格雷码是码距为 1 的代码的一种。所以，一组格雷码的两个相邻代码，只有一位代码有 0、1 的差别，其余各位代码都是相同的。

表 1-2 列出了 4 位二进制码和 4 位格雷码。

表 1-2 二进制码和格雷码

十 进 制 数	二 进 制 码	格 雷 码	码距为 1 非格雷码
0	0000	0000	0000
1	0001	0001	1000
2	0010	0011	1001
3	0011	0010	1011
4	0100	0110	1010
5	0101	0111	1110
6	0110	0101	1111
7	0111	0100	1101
8	1000	1100	1100
9	1001	1101	0100
10	1010	1111	0101
11	1011	1110	0111
12	1100	1010	0110
13	1101	1011	0010
14	1110	1001	0011
15	1111	1000	0001

除了码距为 1 的特征外,格雷码还是一种反射码。将一组格雷码分为长度相等的两部分,处在镜像对称位置上的两个代码除了第一位有 0、1 差别外,其余各位码都是相同的。具有这样特征的代码属于反射码。从表 1-2 上可以看出,处于对称位置上的代码 0100 和 1000, 0001 和 1001 等都具有这样的特征。

码距为 1, 但是不具有反射特征的代码不是格雷码。表 1-2 中的最右边一列代码就是一种码距为 1 的非格雷码。

可以按以下的方法来产生各种码长的格雷码:

- 码长 $n = 1$ 的格雷码只有两个: 0 和 1, 即代码数为 $m = 2$;
- 码长 $n = n+1$ 格雷码都是在码长为 n 的格雷码基础上产生: 将码长为 n 的格雷码作一次镜像对称, 使得代码数为 $2m$ 个, 前 m 个代码添加最高位为 0, 后 m 个代码添加最高位为 1, 这 $2m$ 个代码就是码长 $n = n+1$ 的格雷码;
- 重复第二步的操作, 就可以不断得到更大码长的格雷码。

图 1-1 所示给出了从 1 位~3 位格雷码产生的过程。

一位格雷码	二位格雷码	三位格雷码	
		000	
		001	
		011	
		010	
0	00		
1	01		
	11	110	
	10	111	
		101	
		100	

镜像对称轴
(不包括最高位)

图 1-1 格雷码的产生

还有一种从一般二进制码产生格雷码的方法。步骤如下:

- 设二进制码是 $b_{n-1}b_{n-2}\cdots b_1b_0$, 产生的格雷码是 $g_{n-1}g_{n-2}\cdots g_1g_0$;
- 格雷码最高位和二进制码最高位相等;
- 若 $b_i = b_{i-1}$, 则 $g_{i-1} = 0$; 若 $b_i \neq b_{i-1}$, 则 $g_{i-1} = 1$ 。其中的 $i = n-1, \cdots, 1, 0$ 。

这个规则也可以叙述如下:

- 若码长等于 n , 令 $i = n-1$, 格雷码的最高位和二进制码最高位相同, 即 $g_{n-1} = b_{n-1}$;
- 若 b_{i-1} 和 b_i 相同, 则 $g_{i-1} = 0$; 若 b_{i-1} 和 b_i 不相同, 则 $g_{i-1} = 1$;
- 令 $i = i-1$, 重复第二步, 直到 $i = 0$ 时结束。

使用这个方法, 马上就可以知道和二进制码 010 对应的格雷码是 011。也容易验证表 1-2 最右边一列代码不是格雷码。

如果要从 4 位格雷码中选择 10 个来表示十进制数 0~9, 则可以去掉 4 位格雷码的前 3 个和后 3 个码, 只取中间 10 个代码。这 10 个代码仍然保持格雷码的所有特性, 包括仍具有镜像对称的特性。这样的二—十进制码也称为余三格雷码。

例 1.3 若用 4 位格雷码 0101 表示十进制数 3, 1100 表示十进制数 5, 写出表示十进制数 4 的余三格雷码。

解: 这个代码和 0101 及 1100 的码距都应该是 1。这样的代码不止一个, 例如, 可以是 0100 或者 1101。当然, 不能有两个答案, 还要从中进行选择。

由于表示 4 的代码和表示 5 的代码处于镜像对称的位置上。相应的两个代码的低三位应该相同, 最高位则分别是 0 和 1。只有代码 0100 可以满足这样的要求。

所以, 此时表示十进制数 4 的余三格雷码是 0100。

有的资料上有“模 10 格雷码”的说法, 如“用模 10 格雷码设计一个十进制计数器”。意思是从 4 位格雷码中取出 10 个来表示十进制数 0~9。但是要注意: 这种“模 10 格雷码”的选取不是唯一的, 而是有多种可能。表 1-3 所示列出了余三格雷码, 以及一种“模 10 格雷码”。

“模 10 格雷码”只能保持码距为 1 的特点，但是，并不具有反射的特点。

表 1-3 用格雷码表示十进制数

十进制数	二进制码	余三格雷码	一种模 10 格雷码
0	0000	0010	0000
1	0001	0110	0001
2	0010	0111	0011
3	0011	0101	0010
4	0100	0100	0110
5	0101	1100	0111
6	0110	1101	0101
7	0111	1111	0100
8	1000	1110	1100
9	1001	1010	1101

1.1.3 二—十进制码的运算

二—十进制码本身是二进制码，运算按二进制运算规律进行。但是，二—十进制码代表的是十进制数，运算结果要和十进制数运算结果一致。

在计算机系统中经常用 8421BCD 码来表示十进制数，并且会直接称之为 BCD 码。

用这样的 BCD 码直接运算的结果，往往会和十进制数运算结果不一致。例如， $6 + 5 = 11$ ，而 $(0110)_{8421} + (0101)_{8421} = (1011)$ 。代码 1011 在 8421 码中并不存在。又如， $8 + 9 = 17$ ，而 $(1000)_{8421} + (1001)_{8421} = (00010001)$ 。00010001 对应的十进制数是 11，和正确结果 17 也不一致。

因此，要对二—十进制码运算的结果进行调整才能得到正确的结果，调整的规则是：

- 如果两个二—十进制码相加结果在 1010~1111，则要对结果作加 6 调整；
- 如果两个二—十进制码相加后产生进位，也要对结果作加 6 调整。

如： $(0110)_{8421} + (0101)_{8421} = (1011)$ ，作加 6 调整后，结果为 $(00010001)_{8421}$ ，正确。

又如： $(1000)_{8421} + (1001)_{8421} = (00010001)$ ，由于产生进位要作加 6 调整，加 6 后的结果是 $(00010111)_{8421}$ ，也是正确的。

将来在用逻辑电路对这样的二—十进制数作加法时，就要考虑在一定条件下，对结果作加 6 调整。

如果用其他的编码来表示十进制数，加法运算的调整规则将有所不同。

1.2 用补码表示负数

为了在数字系统和计算机中表示负数，要将二进制数的最高位用作符号位：符号位为 0 表示正数，符号位为 1，则表示负数。用这种方式表示的正负数，称为原码。

例如，十进制数 100 的原码是 01100100。−100 的原码是 11100100。两者的差别只有符号位的不同。

1.2.1 补码

但是, 实际系统中经常使用的不是原码, 而是补码。也就是用补码表示正负数。

正数的补码就是在对应的二进制数的最高位加 0。也就是正数的补码和正数的原码是一样的。

负数的补码是将负数的绝对值的原码“按位求反, 末位加 1”。

例 1.4 将十进制数 -80 用 8 位补码表示。

解: 先将 80 变为等值的二进制数:

$$(80)_{10} = (1010000)_2$$

再用原码表示: $(1010000)_2 = (01010000)_{\text{原码}}$

对这个原码按位取反, 得到: 10101111

末位再加 1: 10110000

即: $(-80)_{10} = (10110000)_{\text{补码}}$

1.2.2 补码加法

在计算机和数字系统中用补码来表示正负数是因为补码运算比原码运算简单。

在算术运算中 $3 + 4$ 是用加法完成的, $3 + (-4)$ 则是要用减法完成。也就是要根据操作数的正负来决定作什么运算。原码加法也会有类似的情况。

使用补码后, 情况就会简单得多。补码加法的规则是:

两个数 (不论正负) 的补码的和, 结果等于这两个数和的补码。

也就是说, 要作两个数的加法, 只要将它们变成补码, 然后直接相加, 结果一定正确。

例 1.5 用补码加法求 $-50 + (-70)$ 。

解: 先将两个数变成补码:

$$(-50)_{10} = (10110010)_{\text{原码}} = (11001110)_{\text{补码}}$$

$$(-70)_{10} = (11000110)_{\text{原码}} = (10111010)_{\text{补码}}$$

作补码加法:

$$\begin{aligned} (-50)_{10} + (-70)_{10} &= (11001110)_{\text{补码}} + (10111010)_{\text{补码}} \\ &= (10001000)_{\text{补码}} = (-120)_{10} \end{aligned}$$

运用补码加法的规则, 在实际应用中, 可以将两个正数的减法, 用加法运算来实现: 只要将减数变为负数的补码, 再进行相加, 就可以完成两个正数的减法。

例 1.6 用补码加法求 $80 - 70$ 。

解: $80 - 70 + 80 + (-70)$

先将两个数 80 和 -70 变成补码:

$$(80)_{10} = (01010000)_{\text{原码}} = (01010000)_{\text{补码}}$$

$$(-70)_{10} = (11000110)_{\text{原码}} = (10111010)_{\text{补码}}$$

作补码加法:

$$\begin{aligned} (80)_{10} + (-70)_{10} &= (01010000)_{\text{补码}} + (10111010)_{\text{补码}} \\ &= (00001010)_{\text{补码}} = (10)_{10} \end{aligned}$$

最后得到的 $(00001010)_{\text{补码}}$ 就是减法的结果, 相当于十进制数的 10。

本章小结

这一章的主要内容是二—十进制数的编码。要掌握什么是有权码，掌握主要的无权码——余3码，正确掌握格雷码的特点，能熟练地写出不同位数的格雷码。本章介绍的二—十进制数运算和补码运算知识，在后面的运算逻辑电路中要用到，在这里属于一种准备知识。

习题和思考题

一、单选题

- 1.1 与二进制数 $(1101000010)_2$ 等值的二—十进制代码是【 】。
 A) $(100001000010)_{8421}$ B) $(101100110100)_{\text{余3码}}$
 C) $(111000110100)_{2421}$ D) $(101100110100)_{2421}$
- 1.2 与余三码 $(011100111001)_{\text{余3码}}$ 等值的十进制数是【 】。
 A) 739 B) 406
 C) 628 D) 517
- 1.3 以下关于格雷码的说法中，正确的是【 】。
 A) 格雷码是一种二—十进制编码
 B) 只要是码距等于1的代码，就是格雷码
 C) 用余三格雷码表示十进制数可以有多种形式
 D) 可以根据一定的算法，将任意位数的二进制码转换为格雷码
- 1.4 与二进制码 $(1010)_2$ 对应的格雷码是【 】。
 A) 1111 B) 1101
 C) 1000 D) 不能确定
- 1.5 十进制数 $(-101)_{10}$ 对应的补码是【 】。
 A) 10011011 B) 10011010
 C) 11100101 D) 11100100
- 1.6 一个十进制数的原码和补码是 11111111 及 10000001，但是不知道哪个是原码，哪个是补码，这个十进制数是【 】。
 A) -127 B) 127
 C) -1 D) 不能确定的
- 1.7 十进制数 0 既可以用原码来表示，也可以用补码来表示。以下说法中正确的是【 】。
 A) 0 的原码和 0 的补码是相同的
 B) 0 的补码有两种形式，0 的原码只有一种形式
 C) 0 的原码有两种形式，0 的补码只有一种形式
 D) 0 的原码有两种形式，0 的补码也有两种形式
- 1.8 用 8421BCD 码表示的十进制数 $(01001001)_{8421} + (01001001)_{8421}$ 的正确结果是十进制数【 】。

A) $(10010010)_{8421}$

B) $(10011000)_{8421}$

C) $(10000010)_{8421}$

D) $(10001000)_{8421}$

二、简答题

1.1 将十六进制数 $(3E)_{16}$ 转换为等值的十进制数，并用 8421 码表示。

1.2 将十六进制数 $(5C)_{16}$ 转换为等值的十进制数，并用 7421 码表示。

1.3 将十六进制数 $(4B)_{16}$ 转换为等值的十进制数，并用余 3 码表示。

1.4 将下列十进制数分别转换成 8421BCD 码和余 3 码。

(1) $(3890)_{10}$

(2) $(7863)_{10}$

(3) $(10952)_{10}$

(4) $(889.01)_{10}$

1.5 将下列 8421 码转换成十进制数和二进制数。

(1) $(1001001010000111)_{8421}$

(2) $(001001101001100000000100)_{8421}$

(3) $(0101000000000100100)_{8421}$

1.6 完成下列二—十进制代码的加法运算。

(1) $(0101)_{8421} + (0011)_{8421}$

(2) $(1010)_{\text{余3码}} + (0110)_{\text{余3码}}$

(3) $(1001)_{8421} + (0111)_{8421}$

(4) $(0100)_{2421} + (1110)_{2421}$

1.7 写出下列二进制数所对应的格雷码。

(1) $(1101)_2$

(2) $(01011)_2$

(3) $(100111)_2$

1.8 写出下列十进制数所对应的原码和补码。

(1) $(125)_{10}$

(2) $(-125)_{10}$

(3) $(1000)_{10}$

(4) $(-1000)_{10}$

第2章 数字逻辑基础

数字电路中经常用 0 和 1 表示变量的值，但是这里的 0 和 1 不是二进制数，而是逻辑值：0 表示逻辑假 (False)，1 表示逻辑真 (True)。取值为逻辑值的变量为逻辑变量，它的运算规则和二进制运算规则是不同的。研究逻辑变量运算规则的数学是布尔代数。布尔代数是 19 世纪中叶英国数学家布尔首先提出来的。本章将以布尔代数（或称逻辑代数）的基本内容为核心，介绍数字逻辑设计的基本理论知识。

通过本章学习，应掌握：

- 各种基本逻辑运算，逻辑函数的表示方法；
- 布尔代数的基本定律、公式和规则；
- 逻辑函数化简的方法。

2.1 逻辑变量和逻辑函数

布尔代数有严密的数学体系，它提出了一整套公理、定理、定律，构成了逻辑系统的数学基础。数字系统中的三种基本逻辑运算（与、或、非）在布尔代数中就是以公理的方式提出来的。

2.1.1 逻辑变量和逻辑系统

数字信号是有具体的数值大小的。在数字电路和数字系统中用逻辑值来表示实际的信号或电路状态。而逻辑值的取值只有两种：0 和 1，0 表示逻辑假，1 表示逻辑真。取值为逻辑值的变量称为**逻辑变量**。

逻辑变量的逻辑值是没有大小的。在逻辑系统中，不能说逻辑 1 比逻辑 0 大。

逻辑变量通常用英文字母表示，如变量 A、B、C 等。在一些实际系统中，也用字符串来表示逻辑变量，如 READY、START 等。

如果逻辑变量是实际数字信号的映射，则逻辑 1 反映信号的值处于某一个范围（例如 3V~5V），逻辑 0 表示信号处于另一个范围（例如 0V~2V）。

逻辑值也可以是电路或信号状态的反映：1 表示状态正常，或者信号有效；0 表示电路状态不正常，或者信号无效。

逻辑变量只有两种取值的逻辑系统称为**二值逻辑系统**。