清华大学计算机系列教材

清华大学计算机系列教材

数字逻辑与数字集成电路

（第 2 版）

王尔乾
杨士强
巴林凤
编著

清华大学出版社

清华大学计算机系列教材

数字逻辑与数字集成电路

（第2版）

王尔乾　杨士强　巴林凤　编著

清华大学出版社

(京)新登字 158 号

内容简介

本书系统地阐述数制和码制、逻辑代数及逻辑函数化简、基本逻辑电路及触发器、各种集成化组合逻辑电路的设计与应用、同步时序电路及异步时序电路的设计与分析、集成化时序电路、逻辑电路的参数、可编程逻辑电路等内容。

本书可作为高等学校计算机专业"数字逻辑"课程的教材,亦可供从事计算机、自动化及电子学方面生产、科研人员及有关人员参考,本书还是学习"逻辑电路"课的参考书。

书　　名:数字逻辑与数字集成电路(第 2 版)
作　　者:王尔乾　杨士强　巴林风　编著
出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)
http://www.tup.tsinghua.edu.cn
责任编辑:马瑛珺
印 刷 者:北京顺义振华印刷厂
发 行 者:新华书店总店北京发行所
开　　本:787×1092 1/16　**印张**:19.25　**字数**:464 千字
版　　次:2002 年 8 月第 2 版　2002 年 8 月第 1 次印刷
书　　号:ISBN 7-302-05036-8/TP・2940
印　　数:0001～5000
定　　价:29.00 元

作者简历

王尔乾，1935年生。1953年就读于上海交通大学电机系，1956年转学至清华大学电子计算机专业，1957年毕业后留校任教。期间曾任清华大学计算机系副主任、主任，计算机研究所所长等职，1987年受聘为教授。曾任北京市人民政府第一至第四届专业顾问，国家教委高等学校计算机科学专业首届教学指导委员会副主任，中国计算机学会理事，中国软件行业协会常务理事等职。现任澳门科技大学资讯科技学院副院长。

长期从事电子线路、数字逻辑、超大规模集成电路系统设计等方面的教学工作，为本科生及研究生开设相关的课程，编著教材多本。

作为主要技术负责人，早期参加数控铣床研制、计算机用晶体管可靠性研究、晶体管参数自动测试等科研工作。20世纪70年代主持了TTL小规模集成电路系列、TTL中规模集成电路系列共24种电路的设计研制，参加了四种型号小型计算机系统的研制。80年代参加“面向大规模集成电路的计算机辅助设计系统”等三项集成电路CAD系统的研制，以上项目分别获得国家科技进步二等奖、国家教委科技进步二等奖、电子部及国防科工委科技进步一等奖、北京市科技进步二等奖。与研究生、合作者共同发表论文十余篇。

序

清华大学计算机系列教材已经出版发行了近30种，包括计算机专业的基础数学、专业技术基础和专业等课程的教材，覆盖了计算机专业大学本科和研究生的主要教学内容。这是一批至今发行数量很大并赢得广大读者赞誉的书籍，是近年来出版的大学计算机教材中影响比较大的一批精品。

该系列教材的作者都是我熟悉的教授与同事，他们长期在第一线担任相关课程的教学工作，是一批很受大学生和研究生欢迎的任课教师。编写高质量的大学（研究生）计算机教材，不仅需要作者具备丰富的教学经验和科研实践，还需要对相关领域科技发展前沿的正确把握和了解。正因为该系列教材的作者们具备了这些条件，才有了这批高质量优秀教材的出版。可以说，教材是他们长期辛勤工作的结晶。系列教材出版发行以来，无论从其发行的数量、读者的反映、已经获得的许多国家级与省部级的奖励以及在各个高等院校教学中所发挥的作用上，都可以看出该系列教材所产生的社会影响与效益。

计算机科技发展异常迅速、内容更新很快。作为教材，一方面要反映本领域基础性、普遍性的知识，保持内容的相对稳定性；另一方面，又需要跟踪科技的发展，及时地调整和更新内容。该系列教材都能按照自身的需要及时地做到这一点，如《计算机组成与结构》一书十年中共发行了三版，其他如《数据结构》等也都已发行了第二版，使教材既保持了稳定性，又达到了先进性的要求。该系列教材内容丰富、体系结构严谨、概念清晰、易学易懂，符合学生的认识规律，适合于教学与自学，深受广大读者的欢迎。系列教材中多数配有丰富的习题集和实验，有的还配备多媒体电子教案，便于学生理论联系实际地学习相关课程。

随着我国进一步的开放，我们需要扩大国际交流，加强学习国外的先进经验。在大学教材建设上，我们也应该注意学习和引进国外的先进教材。但是，计算机系列教材的出版发行实践以及它所取得的效果告诉我们，在当前形势下，编写符合国情的具有自主版权的高质量教材仍具有重大意义和价值。它与前者不仅不矛盾，而且是相辅相成的。

我希望今后有更多、更好的我国优秀教材的出版。

清华大学计算机系教授，中科院院士

张钹

2002年6月25日

第一版前言

“数字逻辑”是计算机专业本科学生的一门主要课程。它是“计算机组成原理”课程的主要先导课之一，是计算机及其应用专业关于计算机系统结构方面四门主干课程（数字逻辑、计算机组成原理、微机与接口技术、计算机系统结构）的第一门课。本课程的主要目的是使学生了解和掌握从对数字系统提出要求开始，一直到用集成电路实现所需逻辑功能为止的整个过程的完整知识。课程的基本要求是系统掌握逻辑电路（重点是组合逻辑电路和同步时序电路）的分析、设计与应用。

数字集成电路是数字系统也是计算机功能实现的物质基础。由于数字集成技术的发展，迄今人们已不再用分立器件去实现逻辑功能部件了，而是用标准集成电路去构成系统。各种标准数字集成电路本身就是优美的逻辑设计作品。因此把数字逻辑和数字集成电路结合起来讲授学习，既使读者掌握数字逻辑部件的分析与设计方法，又使他们了解标准数字集成电路的原理与使用方法，无疑，这是理论结合实际的学习方法。由于篇幅及学时限制，本书不去讨论集成电路、单元电路的线路设计技术和集成电路制造工艺，只限于介绍集成电路的逻辑结构与其应用，并以介绍广泛应用的 TTL 集成电路为主。

本书是按上述思路并结合多年来教学实践经验来编写的。为突出集成电路在功能部件实现中的重要性，本书取名为“数字逻辑及数字集成电路”。本课程的参考学时数为 40～60 学时，其先修课程是电子线路。

由于编者水平有限，书中肯定会有许多缺点和错误，殷切希望广大读者批评指正。

编　者

1992 年 12 月

第二版前言

本书自出版以来，已经10次印刷了。令我们欣慰的是本书已被不少高等学校计算机专业选为教材。在倾听读者意见以及编者教学实践的基础上，考虑以下因素决定对原书作了修改和补充：

(1) 考虑到随机存储器(RAM)器件与计算机系统的关系十分密切，结合“计算机组成原理”课程来讲解 RAM 器件，能更深刻地理解 RAM 器件的结构、时序及其在系统中的应用，从而事半功倍。

事实上，不少高校在“计算机组成原理”课程中是要重新讲述 RAM 器件的。为了减少课程间内容不必要的重叠，经征求部分教师意见，决定从本版中删去有关 RAM 器件的内容。建议这部分内容在“计算机组成原理”课程中讲授。

(2) 为了适应可编程逻辑器件发展迅速和应用广泛的趋势，在本版中增加“通用阵列逻辑(GAL)器件”的内容，并把原书中删去 RAM 后的第 7 章，改名为“可编程逻辑器件”。虽然规模更大的可编程器件不断涌现，但在本版中只选当前已得到广泛应用的 GAL 来介绍。我们认为，掌握数字逻辑及数字集成电路基础仍是同学们的主要任务。若有了计算机编程及计算机系统的有关知识，再通过自学器件的技术资料是完全能掌握其原理和使用的，没有必要也不可能在本书中对它们作一一介绍。

(3) 增加非计算机专业关注的一些内容，如序列信号发生器等。

(4) 改正了第一版中的错误。

编著者虽然从事于计算机教学和科研多年，积累了本学科方面一些理论和经验，但当今计算机科学发展很快，对本课程的一些看法仍有不全面之处，本书中的错误之处，恳请广大读者给予批评和指正，以便不断提高。

编著者
2001 年 9 月于清华大学

目　录

第1章 数制和编码

1.1 数　　制

数制是人们对数量计数的一种统计规律。日常生活中最常遇到的是十进制的进位计数制，在数字系统中，广泛采用的则是二进制、八进制和十六进制。

一种进位计数包含着两个基本的因素：

(1) 基数：它是计数制中所用到的数码的个数，一般地说，基数为 R 的计数制(简称 R 进制)中，包含的是 $0,1,\cdots,R-2,R-1$ 等数码，进位规律是“逢 R 进一”，即每个数位计满 R 就向高位进1，称为 R 进位计数制。

(2) 位权：在一个进位计数制表示的数中，处在不同数位的数码，代表着不同的数值，某一个数位的数值是由这一位数码的值乘上处在这位的一个固定常数。不同数位上的固定常数称为位权值，简称位权。不同数位有不同的位权值。例如，十进制数个位的位权值是1，十位的位权值是 10^1，百位的位权值是 10^2。

广义地说，一个 R 进制数 N，可以有两种表示方式：

① 并列表示方式，也称位置计数法：

$$(N)_R = (K_{n-1}K_{n-2}\cdots K_1K_0.K_{-1}K_{-2}\cdots K_{-m})_R \tag{1-1}$$

其中，n 为整数部分的数位；m 为小数部分的数位；R 表示基数；K_i 为不同数位的数值：

$$0 \leqslant K_i \leqslant (R-1)$$

② 多项式表示法，也称以权展开式：

$$(N)_R = (K_{n-1}R^{n-1} + K_{n-2}R^{n-2} + \cdots + K_1R^1 + K_0R^0 + K_{-1}R^{-1} + \cdots + K_{-m}R^{-m})_R \tag{1-2}$$

或者写成和式：

$$(N)_R = (\sum_{i=-m}^{n-1} K_iR^i)_R$$

式(1-2)中：R 代表进位制的基数；m、n 为正整数，n 代表整数部分的位数，m 代表小数部分的位数；K 代表 R 进位制中 R 个数字符号中的任何一个：

$$0 \leqslant K_i \leqslant (R-1)$$

1.1.1 二进制

1. 二进制数的表示

基数 $R=2$ 的数制为二进制。二进制数的数值表示只有“1”和“0”，进位规律是“逢二进一”，任意一个二进制数 N 的多项式表示为：

$$\begin{aligned}(N)_2 &= K_{n-1}2^{n-1} + K_{n-2}2^{n-2} + \cdots + K_12^1 + K_02^0 + K_{-1}2^{-1} + \cdots + K_{-m}2^{-m} \\ &= (\sum_{i=-m}^{n-1} K_i2^i)_2\end{aligned} \tag{1-3}$$

其中：K_i 为 0 或 1，2 为位权。

例如：二进制数 1011.101 可以展开为：

$$1011.101 = 1\times 2^3 + 0\times 2^2 + 1\times 2^1 + 1\times 2^0 + 1\times 2^{-1} + 0\times 2^{-2} + 1\times 2^{-3}$$

2. 二进制数的运算

二进制数的算术运算比较简单，只要记住两个二进制整数的和与积的运算规律就可以了。

加法规律为：

$$0+0=0 \qquad 0+1=1+0=1 \qquad 1+1=10$$

乘法规律为：

$$0\times 0=0 \qquad 0\times 1=1\times 0=0 \qquad 1\times 1=1$$

由于二进制数每位只可能有两种数值 0 或者 1，在数字系统中，可用电子器件的两种不同的状态来表示一位二进制数，因此实现起来非常方便。例如，在数字系统中，用晶体管的导通表示“0”，而晶体管的截止表示“1”；或用低电位表示“0”、高电位表示“1”。所以二进制数的物理实现简单、易行和可靠，并且存储和传送也方便。其运算规则也很简单。但二进制书写位数太多，不便记忆。为此，通常用八进制和十六进制数作为二进制数的缩写。

1.1.2 八进制

八进制数的数位符号有八个，即 0～7，进位规律是“逢八进一”，基数 $R=8$，任意的一个八进制数 N 的多项展开式为：

$$\begin{aligned}(N)_8 &= K_{n-1}8^{n-1} + K_{n-2}8^{n-2} + \cdots + K_1 8^1 + K_0 8^0 + K_{-1}8^{-1} + \cdots + K_{-m}8^{-m} \\ &= \Big(\sum_{i=-m}^{n-1} K_i 8^i\Big)_8 \qquad (1\text{-}4)\end{aligned}$$

式(1-4)中 K_i 表示为 0～7 中的任意一个。

1.1.3 十六进制

一个十六进制数的表示符号有 16 个，即 0～9，以及用 A,B,C,D,E 和 F 分别表示 10～15。进位规律为“逢十六进一”，基数 $R=16$，任意的一个十六进制数 N 的多项表示式为：

$$\begin{aligned}(N)_{16} &= K_{n-1}16^{n-1} + K_{n-2}16^{n-2} + \cdots + K_1 16^1 + K_0 16^0 + K_{-1}16^{-1} + \cdots + K_{-m}16^{-m} \\ &= \Big(\sum_{i=-m}^{n-1} K_i 16^i\Big)_{16} \qquad (1\text{-}5)\end{aligned}$$

式(1-5)中，K_i 表示为 0～9 以及 A、B、C、D、E 和 F 中的任意一个。

1.1.4 二进制与八进制、十六进制之间的转换

1. 八进制转换为二进制

把每位八进制数用三位二进制数表示。例如$(312.64)_8$，3、1、2、6、4 各用 3 位二进制数表示：

$$
\begin{array}{cccccc}
3 & 1 & 2 & . & 6 & 4 \\
011 & 001 & 010 & . & 110 & 100
\end{array}
$$

所以

$$(312.64)_8=(11001010.1101)_2$$

2. 二进制数转换为八进制

二进制转换为八进制时，整数部分从低位向高位每三位分为一组，最高一组不够时，前面用 0 补足；小数部分从高位向低位每三位一组，最后不足三位的，在低位补 0，然后把每三位的二进制数用相应的八进制数表示。

例如$(10110.11)_2$

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

$$(10110.11)_2=(26.6)_8$$

3. 十六进制转换为二进制

把每位十六进制数用相应的 4 位二进制数表示。

例如$(21A.5)_{16}$

$$
\begin{array}{ccccc}
2 & 1 & A & . & 5 \\
0010 & 0001 & 1010 & . & 0101
\end{array}
$$

$$(21A.5)_{16}=(1000011010.0101)_2$$

4. 二进制转换为十六进制

整数部分由小数点向左，每 4 位一组，最高一组不足 4 位数时，前面补 0；小数部分由小数点向右，每 4 位一组，最后不足 4 位的，在低位补 0，然后把每 4 位二进制数用相应的十六进制数表示。

例如：$(1100101.101)_2$

$$
\begin{array}{cccc}
0110 & 0101 & . & 1010 \\
6 & 5 & . & A
\end{array}
$$

$$(1100101.101)_2=(65.A)_{16}$$

由于八进制、十六进制比二进制书写简短、容易读写，也便于记忆，并且转换为二进制也较方便，因此，在数字系统中得到普遍应用。

1.1.5 二进制与十进制之间的转换

1. 二进制转换为十进制

把二进制数按权展开，利用十进制运算法则，求出其值，即可将二进制数转换为十进制数。

例如：

$$
\begin{aligned}
(101101.101) &= 1\times 2^5+1\times 2^3+1\times 2^2+1\times 2^0+1\times 2^{-1}+1\times 2^{-3} \\
&= 32+8+4+1+0.5+0.125 \\
&= (45.625)_{10}
\end{aligned}
$$

2. 十进制转换为二进制

一个具有整数部分和小数部分的十进制数转换为二进制数时，应当分别将其整数部分和小数部分转换为二进制数，然后用小数点将两部分连接起来。现举例说明。

例 1.1 将十进制数$(157)_{10}$转换为二进制数。

根据式(1-2)：

$$(157)_{10} = K_{n-1}2^{n-1} + K_{n-2}2^{n-2} + \cdots + K_1 2^1 + K_0 2^0$$

显然，等式右边除 K 项外都有 2 的因子。因此，用 2 除$(157)_{10}$，所得余数即为 K_0，

$$\begin{array}{r|l} 2 & 157 \\ \hline & 78 \cdots\cdots \text{余数 } 1 = K_0 \end{array}$$

并得到等式：

$$(78)_{10} = K_{n-1}2^{n-2} + K_{n-2}2^{n-3} + \cdots + K_2 2^1 + K_1$$

同样，再用 2 除$(78)_{10}$，余数则为 K_1，

$$\begin{array}{r|l} 2 & 78 \\ \hline & 39 \cdots\cdots \text{余数 } 0 = K_1 \end{array}$$

再用这样的方法一直继续下去，直到商为 0 为止。

$$\begin{array}{r|ll} 2 & 157 & \\ \hline 2 & 78 & \text{余数为 } 1\text{,所以 } K_0 = 1 \\ \hline 2 & 39 & \text{余数为 } 0\text{,所以 } K_1 = 0 \\ \hline 2 & 19 & \text{余数为 } 1\text{,所以 } K_2 = 1 \\ \hline 2 & 9 & \text{余数为 } 1\text{,所以 } K_3 = 1 \\ \hline 2 & 4 & \text{余数为 } 1\text{,所以 } K_4 = 1 \\ \hline 2 & 2 & \text{余数为 } 0\text{,所以 } K_5 = 0 \\ \hline 2 & 1 & \text{余数为 } 0\text{,所以 } K_6 = 0 \\ \hline & 0 & \text{余数为 } 1\text{,所以 } K_7 = 1 \end{array}$$

得$(157)_{10} = (10011101)_2$

十进制小数转换为二进制小数的方法是：不断用 2 乘要转换的十进制小数，将每次所得的整数(0 或 1)，依次记为 K_{-1}、K_{-2}、…。若乘积的小数部分最后能为 0，那么最后一次乘积的整数部分记作 K_{-m}，则 $0.K_{-1}K_{-2}\cdots K_{-m}$ 即为十进制小数的二进制表达式。因为十进制小数，并不都是能用有限位的二进制小数精确表示，通常则是根据精度要求 m 位，作为十进制小数的二进制的近似表达式。

例 1.2 将$(0.913)_{10}$转换为二进制数，根据式(1-2)：

$$(0.913)_{10} = K_{-1}2^{-1} + K_{-2}2^{-2} + \cdots + K_{-m}2^{-m}$$

两边乘以 2，则得：

$$(1.826)_{10}=K_{-1}+K_{-2}2^{-1}+\cdots+K_{-m}2^{-m+1}$$

所以 $K_{-1}=1$；那么， $(0.826)_{10}=K_{-2}2^{-1}+K_{-3}2^{-2}+\cdots+K_{-m}2^{-m+1}$

两边再乘以 2，则得：

$$(1.652)_{10}=K_{-2}+K_{-3}2^{-1}+K_{-4}2^{-2}+\cdots+K_{-m}2^{-m+2}$$

假如精度要求 $m=4$，转换过程如下：

$$\begin{array}{r} 0.913 \\ \times\quad 2 \\ \hline 1.826 \end{array}\quad \text{整数部分为 1,所以 } K_{-1}=1;$$

$$\begin{array}{r} 0.826 \\ \times\quad 2 \\ \hline 1.652 \end{array}\quad \text{整数部分为 1,所以 } K_{-2}=1;$$

$$\begin{array}{r} 0.652 \\ \times\quad 2 \\ \hline 1.304 \end{array}\quad \text{整数部分为 1,所以 } K_{-3}=1;$$

$$\begin{array}{r} 0.304 \\ \times\quad 2 \\ \hline 0.608 \end{array}\quad \text{整数部分为 0,所以 } K_{-4}=0;$$

因此， $(0.913)_{10}=(0.1110)_2$

任意进制与十进制之间的转换原理及方法，同二进制与十进制之间的转换原理及方法相类似，不再重复。

而任意两种进制之间的转换，一般说来是先由一种进位制转换为十进制，再由十进制转换为另一种进制，把十进制作为桥梁。例如，实现 $(N)_a$ 转换为 $(N)_b$ 时，首先是将 $(N)_a$ 转换为 $(N)_{10}$，也就是将 $(N)_a$ 展开为 a 进位制的多项式，在十进制中计算其值；然后再利用基数乘除法，将 $(N)_{10}$ 转换为 $(N)_b$。

1.2 编　　码

1.2.1 带符号的二进制数的编码

在通常的算术运算中，用符号位“+”号表示正数，用符号位“−”号表示负数。而在数字系统中，正、负数的表示方法是：把数值最高位的前一位作为符号位，并用“0”表示“+”；用“1”表示“−”。连同符号位在一起作为一个数，称之为机器数，它的原来的数值形式则称为这个机器数的真值。

例如：　　　　　　　$X_1=+0.1101$；　　$X_2=-0.1101$

表示成机器数为：　　$X_1=0.1101$；　　$X_2=1.1101$

在数字系统中，表示机器数的方法很多，目前常用的有原码、反码和补码。

1. 原码(true form)

原码表示法又称符号-数值表示法。正数的符号位用"0"表示；负数的符号位用"1"表示；数值部分保持不变。

(1) 小数原码的定义：若二进制数 $X=\pm 0.X_{-1}X_{-2}\cdots X_{-m}$

1) $X>0$ 时

$$X=+0.X_{-1}X_{-2}\cdots X_{-m}$$

$$(X)_{原}=0.X_{-1}X_{-2}\cdots X_{-m}$$

2) $X<0$ 时

$$\begin{aligned}X&=-0.X_{-1}X_{-2}\cdots X_{-m}\\(X)_{原}&=1.X_{-1}X_{-2}\cdots X_{-m}\\&=1-(-0.X_{-1}X_{-2}\cdots X_{-m})\\&=1-X\end{aligned}$$

例如：$X_1=+0.1101$　　　则$(X)_{原}=0.1101$

　　　$X_2=-0.1101$　　　则$(X)_{原}=1-(-0.1101)=1.1101$

3) 零的原码有两种表示形式

$$(+0)_{原}=0.00\cdots0$$

$$(-0)_{原}=1.00\cdots0$$

所以小数原码表示为：

$$(X)_{原}=\begin{cases}X & 当\ 0\leqslant X<1\\1-X & 当\ -1<X\leqslant 0\end{cases}$$

(2) 整数原码的定义：若 $X=\pm X_{n-1}X_{n-2}\cdots X_0$

1) $X>0$ 时，则

$$X=+X_{n-1}X_{n-2}\cdots X_0$$

$$(X)_{原}=0X_{n-1}X_{n-2}\cdots X_0$$

2) $X<0$ 时，则

$$\begin{aligned}X&=-X_{n-1}X_{n-2}\cdots X_0\\(X)_{原}&=1X_{n-1}X_{n-2}\cdots X_0\\&=2^n+X_{n-1}X_{n-2}\cdots X_0\\&=2^n-(-X_{n-1}X_{n-2}\cdots X_0)\\&=2^n-X\end{aligned}$$

例如：$X=-1101$

$$\begin{aligned}(X)_{原}&=11101\\&=10000+1101\\&=10000-(-1101)\\&=2^n-X\end{aligned}$$

因此，整数原码定义为：

$$(X)_{原}=\begin{cases}X & 当\ 0\leqslant X<2^n\\2^n-X & 当\ -2^n<X\leqslant 0\end{cases}$$