众邦教育

面向21世纪课程教材配套辅导

数字电子技术基础

第四版

习题精解

甄 俊 刘松龄 主编

阎石主编《数字电子技术基础》第四版同步辅导

知识串讲 考点点睛

题型解法 一一归纳

课后习题 详尽解答

SWJUP 西南交通大学出版社
Http://press.swjtu.edu.cn

面向21世纪课程教材配套辅导

数字电子技术基础

（第四版）

习题精解

阎石主编《数字电子技术基础》第四版同步辅导

甄　俊　刘松龄　主　编

西南交通大学出版社
·成都·

图书在版编目(CIP)数据

数字电子技术基础习题精解/甄俊,刘松龄主编.
成都:西南交通大学出版社,2006.9
ISBN 7-81104-386-6

Ⅰ.数… Ⅱ.①甄…②刘… Ⅲ.数字电路—电子技术—高等学校—解题
Ⅳ.TN79

中国版本图书馆 CIP 数据核字(2006)第 085758 号

数字电子技术基础(第四版)习题精解

甄 俊 刘松龄 主 编

*

责任编辑 万 方
特邀编辑 何森东
封面设计 晨 宇

西南交通大学出版社出版发行
(成都二环路北一段 111 号 邮政编码:610031 发行部电话:028-87600533
http://press.swjtu.edu.cn
武汉武铁印刷厂印刷

*

成品尺寸:170 mm×230 mm 印张:18.5
字数:352 千字 印数:1-5 000 册
2006 年 9 月第 1 版 2006 年 9 月第 1 次印刷
ISBN 7-81104-386-6
定价:18.00 元

前　言

数字电子技术是高等学校电器类、自控类和电子类专业在电子技术方面入门的技术基础课。本课程的任务是使学生获得数字电子技术方面的基本理论、基础知识和基本技能，培养学生分析问题和解决问题的能力，为深入学习计算机、数控类有关课程以及为今后从事专业工作打下良好的基础。而《〈数字电子技术基础〉习题精解》是与阎石主编的《数字电子技术基础》（第四版）配套的辅导书，可以用于配合此教材的课堂教学。

本书每章均分为三部分：

一、知识要点：归纳了各章的基本内容、基本概念、重要理论、重点难点；

二、例题解析：精选了具有代表性的且与各章的基本概念、重要理论、重点难点相关的典型例题，进行了详细的解析；

三、习题精解：对阎石主编的《数字电子技术基础》（第四版）中的全部习题都做出了详细的分析和解答，给出了解题思路、解题分析方法和计算方法，给出了解题的具体步骤和较为完善的答案。

本书的第一、二、三、四、五章由甄俊老师负责编写，第六、七、八章由刘松龄老师负责编写，第九章由喻胜辉老师负责编写。

由于编者水平有限加之时间仓促，书中之不足和错误在所难免。我们诚恳地希望读者能给我们提出宝贵意见，在此我们向各位读者及在编写过程中曾给予我们关心和帮助的领导、编辑、同仁、朋友表示衷心的感谢。

编　者

2006 年 6 月

目　录

第一章　逻辑代数基础

一　知识要点

1.1　概　述

1.1.1　数字量和模拟量

1. 数字量和模拟量

数字量是时间上、数量上均为离散的物理量。其特点是值域内只可取某些特定值。模拟量是时间上、数量上均为连续的物理量。其特点是值域内可取任意值。

2. 数字信号和模拟信号

数字信号是时间上、数量上均为离散的信号。其特点是大幅值的突变信号，仅研究其有无及出现的次数，而对其大小要求不太严格。模拟信号是时间上、数量上均为连续的信号。其特点是小幅值的缓变信号，不仅要研究其有无，且要研究其大小及对时间的变化规律，而且对其大小的要求亦较严格。

3. 数字电路和模拟电路

数字电路是工作在数字信号下的电子电路。模拟电路是工作在模拟信号下的电子电路。数字电路与模拟电路相比，数字电路主要具有以下优点：

① 结构简单，易于实现，便于集成和系列化生产；

② 不仅能够完成算术运算，而且能够完成逻辑运算（故又称之为数字逻辑电路，简称逻辑电路）；

③ 数字信号便于长期存储；

④ 抗干扰能力强，可靠性高，精确性和稳定性好；

⑤ 使用灵活简便，且便于维护和故障诊断；

⑥ 通用性强，保密性好。

1.1.2　数制和码制

1. 数　制

数制是计数进位制的简称，即多位数中每一位的构成方法以及从低位向高位的进位规则。

1）常用数制

（1）十进制（Decimal）：有 0 ～ 9 十个数码，计数基数为十，进位规则为逢十

进一；

(2) 二进制(Binary)：有 0 和 1 两个数码，计数基数为二，进位规则为逢二进一；

(3) 八进制(Octal)：有 0 ～ 7 八个数码，计数基数为八，进位规则为逢八进一；

(4) 十六进制(Hexadecimal)：有 0～9 十个数码和 A、B、C、D、E、F 六个特殊的数字符号，此六个数字符号依次表示十进制数的 10、11、12、13、14、15，计数基数为十六，进位规则为逢十六进一。

数字电路中常用进制有十进制，二进制。

2) 数制间的相互转换

(1) 十 — 二进制之间的相互转换：

① 十进制数转换为二进制数：

若将带小数十进制数转换为二进制数，则首先将整数部分、小数部分分别转换，然后再合起来。

• 十进制整数转换为二进制数，采取除 2 取余，先余为低法，即将十进制整数逐次用 2 除，直至商为 0，而将余数 0 或 1，按先余为低位数，后余为高位数，依次排列。

• 十进制小数转换为二进制数，采取乘 2 取整，先整为高法，即将十进制小数部分逐次乘以基数 2，取其整数部分，直至小数部分为 0，或按要求的精度取够位数，而将所得整数 0 或 1，按先整为高位数，后整为低位数，依次排列。

同理：若将带小数的十进制数转换为任意 N 进制数，则首先将整数部分采取除以基数 N 取余，先余为低法；小数部分采取乘以基数 N 取整，先整为高法，分别转换，然后再合起来。

② 二进制数转换为十进制数：

采取按权展开式展开求和法，即首先写出二进制数的按权展开式，然后按十进制数的运算规则求和，亦即将每一位二进制数乘以其位权，然后相加。

注意：若系数为 0，则和式中可省略。

同理：任意 N 进制数转换为十进制数均可采取按权展开式展开求和法。

(2) 八 — 二进制之间的相互转换：

① 八进制数转换为二进制数：

采取一位拆三位，依权定码法，即将各位八进制数用与其等值的三位二进制数代之。

注意：小数点对齐，两头的 0 可省略。

② 二进制数转换为八进制数：

采取三位合一位，依权相加法，即将二进制数依其小数点为界，整数部分从右向左、小数部分从左向右，每三位分为一组，两头不足三位时，添 0 补足三位，再将

各组的三位二进制数用与其等值的一位八进制数代之。

(3) 十六 — 二进制之间的相互转换：

① 十六进制数转换为二进制数：

采取一位拆四位，依权定码法，即将各位十六进制数用与其等值的四位二进制数代之。

注意：小数点对齐，两头的 0 可省略。

② 二进制数转换为十六进制数：

采取四位合一位，依权相加法，即将二进制数依其小数点为界，整数部分从右向左、小数部分从左向右，每四位分为一组，两头不足四位时，添 0 补足四位，再将各组的四位二进制数用与其等值的一位十六进制数代之。

(4) 十六 — 八进制之间的相互转换：

① 十六进制数转换为八进制数：

若将十六进制数转换为八进制数，则首先将十六进制数转换为二进制数，然后再将相应的二进制数转换为八进制数。

② 八进制数转换为十六进制数：

若将八进制数转换为十六进制数，则首先将八进制数转换为二进制数，然后再将相应的二进制数转换为十六进制数。

2. 码　制

表示某种特定事物的数码为代码。编制代码时遵循的规则为码制。

常用的 BCD 码：

BCD 码是二 — 十进制码的简称，实为用四位二进制数表示一位十进制数的十进制数码。因为四位二进制数有 0000 ～ 1111 十六种组合状态(即 16 个码组)，所以各种 BCD 码从中任意选取十种组合状态(即 10 个码组)，依次表示十进制数的 0 ～ 9 十个数字符号，多余的六种组合状态(即 6 个码组) 则为禁止码。亦即选用的十种组合状态(即 10 个码组) 称之为有效状态，为其真码，多余的六种组合状态(即 6 个码组) 称之为无效状态，为其伪码。

1)8421BCD 码

8421BCD 码是最常用的一种 BCD 代码，其特点是：它与十进制数码的四位等值二进制数完全相同，从左向右各位的权依次是 8、4、2、1，故又称之为自然 BCD 码。又因为其各位的权固定不变，故又称之为恒权代码，其真码是 0000 ～ 1001，其伪码是 1010 ～ 1111。

2)5421BCD 码

5421BCD 码的特点是从左向右各位的权依次是 5、4、2、1 固定不变，故亦为恒

权代码。

3)2421BCD 码

2421BCD 码的特点是从左向右各位的权依次是 2、4、2、1 固定不变，故亦为恒权代码。

4)5211BCD 码

5211BCD 码的特点是从左向右各位的权依次是 5、2、1、1 固定不变，故亦为恒权代码。

5) 余 3 码

余 3 码的特点是总比对应的 8421BCD 码多 3(0011)，因为其各位的权不能固定，故称之为变权代码，或称之为无权码。

6) 余 3 循环码

余 3 循环码的特点是具有相邻性。相邻性是指任意两个相邻的代码中仅有一位取值不同，因此将余 3 循环码计数器的输出状态译码时，不会产生竞争－冒险现象。又因为其各位的权不能固定，故亦为变权代码，或称之为无权码。

注意：

(1) 用 BCD 码表示多位十进制数时，其高位数及小数点后末尾两头的 0 一般不省略。

(2) 带符号数表示法：

① 原码表示法：＋用 0 表示，－用 1 表示。

n 位二进制原码可表示十进制数范围：

$-(2^{n-1}-1) \sim +(2^{n-1}-1)$

② 补码表示法：对于正数，补码与原码相同；对于负数，符号位仍为 1，但数值部分要按位取反，末位加 1。

n 位二进制补码可表示十进制数范围：

$-2^{n-1} \sim +(2^{n-1}-1)$

1.1.3 算术运算和逻辑运算

逻辑是指思维的规律，在数字电路中指一定的因果关系的判断。

逻辑运算是指遵循一定的规则进行逻辑推理和逻辑判断。

1.2 逻辑代数中的三种基本运算

逻辑代数亦称为开关代数或者布尔代数，用来解决数字逻辑电路的分析与设计问题。

参与逻辑运算的变量叫逻辑变量，用字母 A,B,…，表示。每个变量的取值非 0 即 1，故称逻辑变量之为二值变量。0、1 不表示数的大小，而是代表两种不同的逻辑

状态。

1.2.1　基本逻辑运算

1. 与逻辑运算(亦称为逻辑乘)

1) 与逻辑关系

当决定某一事件的所有条件都具备时，事件才能发生。这种决定事件的因果关系称为与逻辑关系。

2) 与逻辑函数式

$Y = A \cdot B$

3) 与逻辑运算

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

注意:与运算符号可省略。

2. 或逻辑运算(亦称为逻辑加)

1) 或逻辑关系

当决定某一事件的条件有一个或多个具备时，事件就能发生。这种决定事件的因果关系称为或逻辑关系。

2) 或逻辑函数式

$Y = A + B$

3) 或逻辑运算

$0 + 0 = 0 \quad 0 + 1 = 1 \quad 1 + 0 = 1 \quad 1 + 1 = 1$

3. 非逻辑运算(亦称为逻辑非)

1) 非逻辑关系

当决定某一事件的条件具备时，事件不能发生，条件不具备时，事件就能发生。这种决定事件的因果关系称为非逻辑关系。

2) 非逻辑函数式

$Y = \overline{A}$

3) 非逻辑运算

$\overline{0} = 1 \quad \overline{1} = 0$

1.2.2　常用复合逻辑运算

(1) 与非运算逻辑函数式　　$Y = \overline{AB}$

(2) 或非运算逻辑函数式　　$Y = \overline{A + B}$

(3) 与或非运算逻辑函数式　　$Y = \overline{AB + CD}$

(4) 异或运算逻辑函数式　　$Y = \overline{A}B + A\overline{B} = A \oplus B$

(5) 同或运算逻辑函数式　　$Y = AB + \overline{A}\,\overline{B} = A \odot B$

1.3 逻辑代数的基本公式和常用公式

逻辑代数的基本公式亦为布尔恒等式。

(1)0－1 律(常量与变量之间的运算规则)　$0\cdot A=0$，$1+A=1$

(2) 自等律(常量与变量之间的运算规则)　$1\cdot A=A, 0+A=A$

(3) 重叠律(同一变量之间的运算规则)　$A\cdot A=A$，$A+A=A$

(4) 互补律(变量与反变量之间的运算规则)　$A\cdot\overline{A}=0$，$A+\overline{A}=1$

(5) 交换律(因形同普通代数,故称之为代数律)$AB=BA$，$A+B=B+A$

(6) 结合律(因形同普通代数,故称之为代数律)

$A(BC)=(AB)C$，　$A+(B+C)=(A+B)+C$

(7) 分配律(因形同普通代数,故称之为代数律)

$A(B+C)=AB+AC$，　$A+BC=(A+B)(A+C)$

(8) 反演律(亦称为德·摩根定理)　$\overline{AB}=\overline{A}+\overline{B}$，　$\overline{A+B}=\overline{A}\,\overline{B}$

(9) 还原律(变量两次求反,还原为原变量)$\overline{\overline{A}}=A$

(10) 吸收律 $A(A+B)=A$，　$A+AB=A$；　$A+\overline{A}B=A+B$，

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

(11) 包含律 $AB+\overline{A}C+BC=AB+\overline{A}C$，

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

1.4 逻辑代数的基本定理

1.4.1 代入定理

对于任何一个逻辑等式,若以某个逻辑变量或逻辑函数同时取代等式两端的任何一个逻辑变量 A 后,则等式依然成立。

代入定理可以用来扩展公式。

1.4.2 反演定理

1. 反演式

逻辑函数 Y 的反函数$\overline{Y}$称为 Y 的反演式。

2. 反演定理

若将任何一个逻辑函数 Y 中的“·”、“+”相互转换,常量 0、1 相互转换,原变量、反变量相互转换,则所得的式子即为 Y 的反演式或反函数$\overline{Y}$。

反演定理可以用来快速求得逻辑函数 Y 的反逻辑函数$\overline{Y}$。

1.4.3 对偶定理

1. 对偶规则

若将任何一个逻辑函数 Y 中的“·”、“+”相互转换,常量 0、1 相互转换,原变

量、反变量保持不变，则所得的式子即为 Y 的对偶式 Y'。

Y 的对偶式有时也用 Yd 表示。

2. 对偶定理

如果两个逻辑函数 $F = G$，则有 $F' = G'$；反之亦然。

对偶定理可以用来减少公式记忆量。

注意：使用逻辑代数的基本公式和常用公式及基本定理时：

① 运算顺序为先括号，然后乘，最后加；

② 由乘积项变为相加时必须加括号；

③ 对乘积项、逻辑函数式求反时，其外边的括号可省略；

④ 不属于单一变量上的反号应保留。

1.5 逻辑函数及其表示方法

1.5.1 逻辑函数

因为逻辑函数中的逻辑变量及函数只可取 0 和 1 二个值(每个逻辑变量及函数的取值非 0 即 1)，故我们仅讨论二值函数。且 0、1 不表示数的大小，而是代表两种不同的逻辑状态。

1.5.2 逻辑函数的表示方法

(1) 真值表表示法(唯一的)

(2) 逻辑函数式表示法(多种的)

(3) 逻辑图表示法(多种的)

(4) 卡诺图表示法(唯一的)

1.5.3 逻辑函数的两种标准形式

逻辑函数有两种标准形式：最小项之和形式和最大项之积形式。这里重点介绍最小项之和形式。

1. 最小项的定义

在 n 变量逻辑函数中，若 m 是包含 n 个因子的乘项积，而且这 n 个变量均以原变量或反变量的形式在 m 中出现一次，则称 m 为该组变量的最小项。

2. 最小项及其编号的表示方法

通常用符号 mi 来表示最小项及其编号，用符号 m 来表示最小项，用符号 i 来表示最小项的编号。最小项编号的确定：把最小项中的原变量记为 1，反变量记为 0，当变量顺序确定后，可以按顺序排列成一个二进制数，则与这个二进制数相对应的十进制数，就是这个最小项的编号。

注意：二变量全部最小项有 $m_0 \sim m_3$ 共 $4(2^2)$ 个；三变量全部最小项有 $m_0 \sim$

m_7 共8(2^3)个；四变量全部最小项有 $m_0 \sim m_{15}$ 共16(2^4)个；变量全部最小项有 $m_0 \sim m_{(2^n-1)}$ 共 2^n 个。

3. 最小项的性质

(1) 在输入变量的任何取值下必有一个且仅有一个最小项的值为1；

(2) 全体最小项之和为1；

(3) 任意两个最小项的乘积为0；

(4) 具有相邻性的两个最小项可以合并，并可消去一对因子。(只有一个因子不同的两个最小项是具有相邻性的最小项。)

逻辑函数的最小项之和形式：任何一个逻辑函数都可以表示成唯一的一组最小项之和，称为标准与或表达式，也称为逻辑函数的最小项之和形式(或最小项表达式)。对于不是最小项之和形式而是与或表达式的任何逻辑函数，都可以利用基本公式 $A+\overline{A}=1$ 和 $A(B+C)=AB+BC$ 来配项展开，化为标准的最小项之和形式。

1.6 逻辑函数的公式化简法

1.6.1 逻辑函数的最简形式

1. 最简的标准

(1) 首先逻辑函数式中，包含的或运算的项最少(与门数最少)；

(2) 其次每一项中包含与运算的因子最少(与门的输入端数最少)，则此函数式为最简函数式。

常用的最简形式有：

与－或式；与非－与非式。将与－或式取两次反，再去掉一个反号即可得与非－与非式。

1.6.2 常用的公式化简方法

利用基本公式和常用公式，再配合常用的并项法、吸收法、消项法、消因子法和配项法，消去函数式中多余的乘积项和多余的因子，即可求得逻辑函数式的最简形式。

1.7 逻辑函数的卡诺图化简法

1.7.1 逻辑函数的卡诺图表示方法

1. n 变量卡诺图(n 变量全部最小项的卡诺图)

将 n 变量的全部最小项各用一个小方格表示，并使具有逻辑相邻性的最小项在几何位置上也相邻，所得图形叫 n 变量最小项的卡诺图，简称 n 变量卡诺图。

注意：在卡诺图中，左、右、上、下、每一行的首尾、每一列的首尾及四角的最小项都是具有逻辑相邻性的最小项。

2. 用卡诺图表示逻辑函数

方法一：

(1) 首先把已知逻辑函数式化为最小项之和形式。

(2) 然后将逻辑函数式中包含的最小项在卡诺图上对应的方格中填 1，其余方格中填 0(填 0 可省略)。

方法二：

根据逻辑函数式直接填卡诺图。

方法三：

根据逻辑函数真值表直接填卡诺图。

3. 逻辑函数的卡诺图化简法

(1) 化简依据：具有逻辑相邻性的最小项可以合并，并可以消去互为反变量的因子。

(2) 合并化简(画矩形圈) 规则：

① 能够合并在一起的最小项是 2^n 个[若合并为 1 的最小项(圈 1 法)，则每圈应有 2^n 个 1；若合并为 0 的最小项(圈 0 法)，则每圈应有 2^n 个 0]；

② 圈的数目越少越简(与项最少，与门数最少)；

③ 圈内的最小项越多越简(消去互为反变量的因子数最多，与项的因子数最少，与门的输入端数最少)；

④ 在卡诺图中，左、右、上、下、每一行的首尾、每一列的首尾及四角的最小项都是具有逻辑相邻性的最小项，均可以圈到合并；

⑤ 卡诺图中的 1(或 0) 可以重复圈到，但每次都必须圈到新的 1(或 0)；

⑥ 特别注意：卡诺图中所有的 1(或 0) 都必须圈到，不能合并的 1(或 0) 必须单独画圈。

若卡诺图中 1 的数目远远大于 0 的数目，则可用圈 0(合并为 0 的最小项) 的方法，先求得逻辑函数 Y 的最简反逻辑函数式 $\overline{Y}$，再对 $\overline{Y}$ 求一次反，即可得最简逻辑函数式 Y。若求逻辑函数 Y 的最简反逻辑函数式 $\overline{Y}$，或者求逻辑函数 Y 的最简与一或非式，则用圈 0 法更为简便。

1.8 具有无关项的逻辑函数及其化简

1.8.1 约束项和任意项及逻辑函数式中的无关项

1. 约束项

某些逻辑函数对其某些输入变量的取值必须加以限制，所加的限制称为约束，其对应的最小项称为约束项，(用它们对应的最小项恒等于 0 来表示)。

2. 任意项

某些输入变量的取值下，逻辑函数的值是 0 或是 1 均可，并不影响电路的逻辑功能，在这些输入变量的取值下其值是 1 的最小项称为任意项。

3. 逻辑函数式中的无关项

在实际的数字系统中，会出现这样一种情况：逻辑函数式中没有包含的某些最小项，写入或不写入逻辑函数式，都不影响原逻辑函数的值，不影响原逻辑函数表示的逻辑功能，这样的最小项叫“无关项”。无关项由“约束项”和“任意项”形成。因为约束项恒等于 0，任意项的输入变量取值下，逻辑函数的值是 0 是 1 均可，并不影响电路的逻辑功能，所以将约束项、任意项写不写入逻辑函数式中均无关紧要，因此将约束项、任意项统称为逻辑函数式中的无关项，可以用 × 表示。

1.8.2　无关项在逻辑函数中的应用

无关项在卡诺图对应的方格中亦用 × 表示，必须充分利用无关项 × 化简逻辑函数，× 的取值可以是 1，也可以是 0，根据化简时所需而定，圈 1 时用到的 ×，便认为是 1，用不到的 ×，就认为是 0，且无关项 × 亦可以重复圈到。

二　例题解析

例 1－1　将十进制数 13.125 转换成二进制数。

解　若将带小数十进制数转换为二进制数，则首先将整数部分、小数部分分别转换，然后再合起来。

(1) 将十进制整数 13 转换为二进制数：

采取除 2 取余，先余为低法：

即将十进制整数逐次用基数 2 除，直至商为 0，而将余数 0 或 1，按先余为低位数，后余为高位数，依次排列。

	余数	
2 | 13	1	B_0 最低位
6	0	B_1
3	1	B_2
1	1	B_3 最高位
0		

即 $(13)_D = (1101)_B$

(2) 将十进制小数 0.125 转换为二进制数：

采取乘 2 取整，先整为高法：

即将十进制小数部分逐次乘以基数 2，取其整数部分，直至小数部分为 0，或按要求的精度取够位数，而将所得的整数 0 或 1，按先整为高位数，后整为低位数，依次排列。

整数

0.125
× 2
0.250---------0---------B_{-1} 最高位
× 2
0.500---------0---------B_{-2}
× 2
1.000---------1----------B_{-3} 最低位

即 $(0.125)_D = (0.001)_B$

(3) 将转换为二进制数的整数部分、小数部分二者相加即为所求，即

$(13.125)_D = (1101.001)_B$

例 1—2 将十进制数 13.675 转换成八进制数，要求误差 $\varepsilon < 2^{-3}$。

解 (1) 将十进制整数 13 转换为八进制数

采取除 8 取余，先余为低法：

余数

8 | 13-------5----------O_0 最低位
  | 1--------1----------O_1 最高位
   0

即 $(13)_D = (15)_O$

(2) 将十进制小数 0.675 转换为八进制数

采取乘 8 取整，先整为高法：

本例要求误差 $\varepsilon < 2^{-3}$，由于是小数部分的转换，所以将十进制小数0.675逐次乘以基数 8，取其整数部分即可。

整数

0.675
× 8
5.400---------5----------O_{-1} 最高位
0.400
× 8
3.200---------3----------O_{-2}
0.200
× 8
1.600---------1----------O_{-3} 最低位

即 $(0.675)_D = (0.531)_O$