湖南大学现代远程教育
全国税务系统远程教育
系列教材

数字逻辑

学习指导

湖南电子音像出版社

编 写 说 明

现代远程教育是20世纪80年代以来国际教育发展的共同趋势。1998年9月,教育部批准湖南大学等四所大学首批试办现代远程教育,标志着我国现代远程教育已正式启动。湖南大学的现代远程教育，在探索中不断前进，特别是与国家税务总局合作开办的主要面向行业的财税远程教育，在办学模式、教学手段等方面正在实现跨越式发展。

在全国税务系统远程学历教育领导小组的领导下和在全国税务系统远程学历教育教学指导委员会的指导下,我们根据湖南大学本科学历教育教学大纲和新形势下社会对财经类人才素质的要求,组织全国相关专业的著名教授、学者、专家编写了这套系列教材及学习指导书,并配有电子光盘、VCD光盘、网络课件等教学资源。

本书由周南良、汪诗林主编。

由于时间原因，错漏之处在所难免，敬请同行专家批评指正。

目　　录

教学目的与要求

“数字逻辑”是高等院校计算机、自动控制、数字通讯和数字测量等专业的一门重要专业基础课程。学习该课程的主要目的是使学生掌握有关数字逻辑设计的基本概念、基本理论以及数字逻辑电路分析和设计的基本方法，为数字计算机和其他数字系统的硬件分析和设计打下坚实的基础。

通过本课程的学习，学生应熟练掌握计算机中数的表示方法；掌握数字系统分析和设计的重要工具——布尔代数，学会运用这个工具去分析和设计数字逻辑电路；通过典型例题的解析，掌握数字电路分析和设计的一般步骤和方法。举一反三，理论联系实际，独立解决工作中遇到的有关数字逻辑问题，提高学生分析问题和解决问题的能力。

教学重点与难点

本课程的教学重点有两个：一是数字逻辑设计的基础知识，包括计算机算术运算的基础(二进制、八进制与十六进制，原码与补码，定点与浮点表示)以及计算机逻辑设计的基础(布尔代数、布尔函数化简)，这部分内容是基本的，也是最重要的。二是数字逻辑网络的分析和设计方法，包括组合逻辑网络，同步时序网络和中、大规模集成电路。这一部分是本课程的主体内容，也是重要内容。

本课程中涉及的基本门电路、触发器和集成电路，重点从逻辑功能的角度来介绍，而不是从电路内部构成、电性能的角度来介绍，这是学习中要注意的。

随着集成电路技术的发展，研究数字系统逻辑设计的方法和标准也在不断地发展变化。但是，本课程重点讲述的经典最小化设计的基本方法和原则，仍应是数字逻辑设计的基础；用逻辑门和触发器为基本单元电路来构成逻辑部件，仍应作为数字逻辑设计的基本技术。

本课程的难点在时序逻辑网络的分析和设计。因为组合逻辑网络的输入和输出，与时间没有关系，当前有什么样的输入，就有确定的输出；而时序逻辑网络的输出不仅与当前输入有关，还与以前输入有关。因此，时序网络的概念比组合网络要复杂，其设计方法和步骤也比组合网络复杂得多。在时序网络中，异步时序网络又比同步时序网络难掌握，尤其是电平异步时序网络。这是在教学中需引起注意的。

对于教材中每一章的重点与难点,我们将在下面的教学内容提要中分别加以说明。

教学内容提要

第一章　数制和编码

1.1 进位计数制

1.基数和位权

基数表示某种进位制所具有的数字符号的个数。位权(或权)表示某种进位制的数中不同位置上数字的单位数值。

2.数的并列表示法和多项式表示法

设 N 为 R 进制的数,则该数可表示成

$(N)_R=(r_{n-1}\ r_{n-2}\cdots r_1 r_0 \cdot r_{-1} r_{-2}\cdots r_{-m})_R$(并列表示)

$$
\begin{aligned}
(N)_R &= r_{n-1}R^{n-1}+r_{n-2}R^{n-2}+\cdots+r_1R^1+r_0R^0 \\
&\quad +r_{-1}R^{-1}+r_{-2}R^{-2}+\cdots+r_{-m}R^{-m}\text{(多项式表示)} \\
&= \sum_{i=-m}^{n-1} r_iR^i
\end{aligned}
$$

其中,n 表示整数的位数,m 表示小数的位数;R 为基数,r_i 是 R 个数字符号中的任一个,即有　$r_i\in\{0,1,2,\cdots,R-1\}$

3. 二进制，八进制和十六进制

二进制的基数 $R=2$，每位只有 0 和 1 两个数字符号，其进位规则是“逢二进一”。

八进制的基数 $R=8$，每位可取 8 个不同的数字符号（即 0，1，2，3，4，5，6，7），其进位规则是“逢八进一”。

十六进制的基数 $R=16$，每位可取 16 个不同的数字符号（即 0～9，A～F），其进位规则是“逢十六进一”。

1.2 数制转换

1. **多项式替代法**：用于 α 进制→十进制转换。
2. **基数乘除法**：用于十进制→β 进制转换
3. **混合法**：用于任意进制 $\alpha\to\beta$ 之间转换
4. **直接转换法**：用于 2^n 进位制之间转换。

1.3 带符号二进制数的代码表示

1. 原码表示法

原码表示法表示一个二进制数时，其符号部分用代码“0”表示“+”，用代码“1”表示“-”，数值部分以真值形式表示。

设真值 x 为整数，其数值部分为 n 位，则原码的一般定义为

$$[x]_{原}=\begin{cases} x & 当\ 0\leqslant x<2^n \\ 2^n-x & 当\ -2^n<x\leqslant 0 \end{cases}$$

设真值 x 为小数时，则原码定义如下：

$$[x]_{原}=\begin{cases} x & 当\ 0\leqslant x<1 \\ 1-x & 当\ -1<x\leqslant 0 \end{cases}$$

注意原码表示零有两种形式：

$[+00\cdots0]_{原}=000\cdots0$

$[-00\cdots0]_{原}=100\cdots0$

2. 反码表示法

反码表示法的符号位部分与原码表示法相同。数值部分与数的符号位有关:对于正数,反码的数值位与原码相同;对于负数,反码的数值位是将原码数值位按位取反而得。

设真值 x 的数值部分有 n 位,则整数反码的定义如下

$$[x]_{反}=\begin{cases} x & 当\ 0\leqslant x<2^n \\ (2^{n+1}-1)+x & 当\ -2^n<x\leqslant 0 \end{cases}$$

小数反码的定义如下

$$[x]_{反}=\begin{cases} x & 当\ 0\leqslant x<1 \\ (2-2^{-n})+x & 当\ -1<x\leqslant 0 \end{cases}$$

同样,反码也有两种表示零的形式:

$[+00\cdots0]_{反}=0\ 00\cdots0$

$[-00\cdots0]_{反}=1\ 11\cdots1$

3. 补码表示法

补码表示法是目前计算机中用得最多,也是最重要的一种表示法。

补码表示法的符号位部分同原码。数值位部分与数的符号有关:对于正数,补码的数值位同原码;对于负数,补码的数值位是将原码数值位按位取反后再在最低位加 1。

设真值 x 数值部分为 n 位,则整数补码的一般定义如下

$$[x]_{补}=\begin{cases} x & 当\ 0\leqslant x<2^n \\ 2^{n+1}+x & 当\ -2^n\leqslant x<0 \end{cases}$$

需要引起注意的是,当 x 为负数时,x 的定义域可以等于 -2^n,而原码,反码对这个数是没有定义的。例如,当 $n=4$,$x=-2^4$(即二进制 -10000)时

$$\begin{aligned}[x]_{补}&=2^{n+1}+x\\&=100000-10000\\&=10000\end{aligned}$$

这说明当 $n=4$ 时，$x=-10000$ 这个数的补码是有定义的。

引进了补码后，计算机进行加减运算就特别方便，如要完成 x_1+x_2，只要分别求出 $[x_1]_补$ 和 $[x_2]_补$，然后相加，符号位也参加运算，得到 $[x_1+x_2]_补$；如要完成 x_1-x_2，只要分别求出 $[x_1]_补$ 和 $[-x_2]_补$，然后相加，得到 $[x_1-x_2]_补$。

为了判断运算过程中是否出现了溢出，采用两位符号位的变形补码，即用 00 表示正数，用 11 表示负数。两个符号位同数值位一起参加运算，当运算结果的符号位仍是 00 或 11 时，说明没有出现溢出，结果正确；当运算结果的符号位出现 01 或 10 时，说明出现了溢出，且 01 表示有正溢出，10 表示有负溢出。

1.4 数的定点与浮点表示

1. 定点表示法

定点小数	数符 S_M	·尾数 M
定点整数	数符 S_M	尾数 M·

2. 浮点表示法

一个浮点数可以用一对二进制定点数来表示：阶码用定点整数表示，尾数用定点小数表示。格式如下：

$$\underbrace{\boxed{S_E}\boxed{\quad E\quad}}_{\text{阶码}}\underbrace{\boxed{S_M}\boxed{\quad M\quad}}_{\text{尾数}}$$

阶码除了用原码、反码和补码表示外，在许多通用机中还常用移码来表示。

设阶码真值 x 为

$x=\pm x_1x_2\cdots\cdots x_n$

则 $[x]_移=2^n+x \quad (-2^n\leqslant x<2^n)$

注意阶码 x 的定义域包含 -2^n 这个端点。当 $x=-2^n$ 时，$[x]_{移}=00\cdots0$。

1.5 二－十进制编码

用若干位二进制代码来表示一位 10 进制数字符号的方法称为二－十编码。

常用的二－十编码：

1.8421(BCD)码

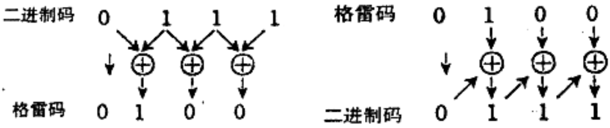
2.余 3 码

3.2421 码

1.6 可靠性编码

1.格雷(Gray)码

格雷码的编码特点是相邻的两个代码之间只有一位不一样。要求掌握格雷码与二进制码之间的转换关系：



2.奇偶校验码

奇偶校验码由 n 位信息位和附加的一位校验位形成的编码构成。表示成

$C_1C_2\cdots C_nP$

其中 $C_1,C_2,\cdots,C_n$ 为信息位，P 为校验位。

编码方法：即确定校验位 P 的取值。

$P=C_1\oplus C_2\oplus\cdots\oplus C_n$　　（构成偶校验码）

$P=C_1 \oplus C_2 \oplus \cdots \oplus C_n \oplus 1$ （构成奇校验码）

校验方法：在发送端将信息位和校验位构成的奇（或偶）校验码一起发送出去。在接收端对接收到的奇（或偶）校验码进行校验。其校验方程为

$S=C_1 \oplus C_2 \oplus \cdots \oplus C_n \oplus P$

根据 S 的值来判断发送过程中有无出错。

3. 汉明校验码

奇偶校验码只是一种检错码，而汉明校验码是一种纠错码。汉明校验的基础是奇偶校验。要求掌握汉明校验的编码方法和校验方法，有关汉明校验码的编码方法，我们将在后面“疑难问题解答”中专门介绍。

第二章　布尔代数基础

2.1 布尔代数的基本概念

1. 布尔变量及其基本运算

布尔变量只有两种取值，即 0 或 1，布尔变量的基本运算只有“或”、“与”和“非”三种。

2. 布尔函数及其表示方法

设$(x_1, x_2, \cdots, x_n)$为布尔代数的一组布尔变量，其中每个变量取值为 0 或 1，则当把 n 序列$(x_1, x_2, \cdots, x_n)$映射到 $B=(0,1)$ 时，这个映射就是一个布尔函数，记为

$F=f(x_1, x_2, \cdots, x_n)$

布尔函数的表示方法有三种：真值表、布尔表达式和卡诺图。

3. 布尔函数的“相等”概念

2.2 布尔代数的基本公式、定理和规则

1. 布尔代数的基本公式

交换律 $A+B=B+A$

$A\cdot B=B\cdot A$

结合律 $(A+B)+C=A+(B+C)$

$(A\cdot B)\cdot C=A\cdot(B\cdot C)$

分配律 $A\cdot(B+C)=A\cdot B+A\cdot C$

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

0－1 律 $A+0=A \quad A+1=1$

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

互补律 $A+\overline{A}=1 \quad A\cdot\overline{A}=0$

等幂律 $A+A=A \quad A\cdot A=A$

吸收律 $A+AB=A \quad A+\overline{A}B=A+B$

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

对合律 $\overline{\overline{A}}=A$

包含律 $AB+\overline{A}C+BC=AB+\overline{A}C$

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

2. 主要定理

定理 1 德·摩根(De Morgan)定理

(1) $\overline{x_1+x_2+\cdots+x_n}=\overline{x_1}\cdot\overline{x_2}\cdot\cdots\cdot\overline{x_n}$

(2) $\overline{x_1\cdot x_2\cdot\cdots\cdot x_n}=\overline{x_1}+\overline{x_2}+\cdots+\overline{x_n}$

定理 2 香农(Snannon)定理

$\overline{f(x_1,x_2,\cdots,x_n,0,1,+,\cdot)}=f(\overline{x_1},\overline{x_2},\cdots,\overline{x_n},1,0,\cdot,+)$

定理 3 展开定理

(1) $f(x_1,\cdots,x_i,\cdots,x_n)$

$=x_i f(x_1,\cdots,1,\cdots,x_n)+\overline{x_i} f(x_1,\cdots,0,\cdots,x_n)$

(2) $f(x_1,\cdots,x_i,\cdots,x_n)$

$$=[x_i + f(x_1,\cdots,0,\cdots,x_n)]\cdot[\overline{x_i} + f(x_1,\cdots,1,\cdots,x_n)]$$

3. 重要规则

(1)代入规则

对逻辑等式中的任意变量 x,若将所有出现 x 的位置都代之以同一个布尔函数 F,则等式仍然成立。

(2)对偶规则

任何一个布尔函数表达式 F,若将表达式中的所有的"+"改成"·","·"改成"+","1"改成"0","0"改成"1",而变量保持不变,则可得到一个新的函数表达式 F_d,称 F_d 为 F 的对偶函数。

(3)反演规则

任何一个布尔函数表达式 F,若将表达式中的所有的"+"改成"·","·"改成"+","1"改成"0","0"改成"1",原变量改为反变量,反变量改为原变量,则可得函数 F 的反函数(或称补函数)$\overline{F}$,这一规则称为反演规则。

2.3 布尔函数的基本形式

1. 函数的"积之和"与"和之积"形式

2. 函数的"标准积之和"与"标准和之积"形式

最大项与最小项的概念。

2.4 不完全确定的布尔函数

随意项 d 的概念。

第三章　布尔函数的简化和实现

3.1 代数化简法

运用布尔代数的基本公式、定理和规则对布尔函数表达式进行恒等变换,消去表达式中的多余项和多余变量,以获得最简表达式。常用的方法有并项法,吸收法,配项法,消去冗余项法等。

3.2 卡诺图法

将布尔函数用卡诺图来表示,在卡诺图上进行函数的化简。这是一种常用的手工方法,后面设计中经常用到它。

1. 卡诺图的构成
2. 布尔函数在卡诺图上的表示
3. 卡诺图化简的基本步骤

几个概念:蕴涵项,质蕴涵项和必要质蕴涵项。

3.3 列表化简法($Q-M$ 法)

1. 用列表法确定布尔函数的所有质蕴涵
2. 用质蕴涵表确定必要质蕴涵
3. 求函数的最小覆盖

行列消去法:用优势行和优势列规则化简质蕴涵表。

布尔代数法:从简化质蕴涵表列出布尔表达式,从中选出最小覆盖方案。

3.4 布尔函数的实现

1. 用与非门实现布尔函数

首先将函数化成最简“与或”形式；然后对表达式二次取反，得到函数的“与非－与非”形式；最后用与非门画出逻辑图。

2. 用或非门实现布尔函数

首先将函数化成最简“或与”形式；然后对表达式二次取反，得到函数的“或非－或非”形式；最后用或非门画出逻辑图。

3. 用与或非门实现布尔函数

首先将函数化简成原函数 F 的最简“与或”形式以及补函数 $\overline{F}$ 的最简“与或”形式；然后对 F 的表达式二次取反，对 $\overline{F}$ 的表达式一次取反，即得 F 的“与或非”的两种表达式；最后比较两者取较简单的一种，用与或非门画出逻辑图。

第四章　基本逻辑门电路

4.1 开关器件及其逻辑描述

4.2 分立元件的门电路

1. 二极管“与”门
2. 二极管“或”门
3. “非”门(反相器)
4. “与非”门