高等院校计算机实验与实践系列示范教材

数字逻辑电路设计学习指导与实验教程

马汉达 赵念强 编著
鲍可进 主审

清华大学出版社

高等院校计算机实验与实践系列示范教材

数字逻辑电路设计
学习指导与实验教程

马汉达 赵念强 编著

清华大学出版社
北京

内 容 简 介

从学生课程内容的学习和提高实验技能的角度出发，本书分为两部分：第一部分是学习指导，根据鲍可进主编的《数字逻辑电路设计》教材的内容，主要从课程的要点指导、例题精讲、习题参考答案3个方面对每一章的重点内容进行概括和总结，方便学生学习；第二部分是实验教程，主要介绍数字逻辑电路设计课程实验涉及的相关内容，如EDA技术的基本概念、开发方法，VHDL语言的主要语法，Quartus Ⅱ开发工具，以提高学生实际动手能力和工程设计能力。精心选择了若干个基础实验和综合设计性实验，实验具有一定的层次性、综合性、设计性、实用性和趣味性，能够引起学生的学习兴趣，激发他们内在的学习动力。

本书可作为高等院校电子信息、通信工程、计算机科学与技术、软件工程、网络工程、自动化等电气信息类专业数字逻辑电路设计课程和EDA技术课程的实验教学用书，同时也可作为高等院校相关专业的教学参考书。

图书在版编目(CIP)数据

数字逻辑电路设计学习指导与实验教程/马汉达，赵念强编著．—北京：清华大学出版社，2012.8
(高等院校计算机实验与实践系列示范教材)
ISBN 978-7-302-28814-5

Ⅰ．①数… Ⅱ．①马… ②赵… Ⅲ．①数字电路－逻辑电路－高等学校－教学参考资料 Ⅳ．①TN79

中国版本图书馆CIP数据核字(2012)第101715号

责任编辑：高买花　王冰飞
封面设计：傅瑞学
责任校对：梁　毅
责任印制：沈　露

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质 量 反 馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课 件 下 载：http://www.tup.com.cn,010-62795954
印 装 者：北京嘉实印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：13.25　　**字　　数**：325千字
版　　次：2012年8月第1版　　**印　　次**：2012年8月第1次印刷
印　　数：1～3000
定　　价：25.00元

产品编号：045679-01

出版说明

当前，重视实验与实践教育是各国高等教育界的发展潮流，我国与国外教学工作的差距也主要表现在实践教学环节上。面对新的形式和新的挑战，完善实验与实践教育体系成为一种必然。为了培养具有高质量、高素质、高实践能力和高创新能力的人才，全国很多高等院校在实验与实践教学方面进行了大力改革，在实验与实践教学内容、教学方法、教学体系、实验室建设等方面积累了大量的宝贵经验，起到了教学示范作用。

实验与实践性教学与理论教学是相辅相成的，具有同等重要的地位。它是在开放教育的基础上，为配合理论教学、培养学生分析问题和解决问题的能力以及加强训练学生专业实践能力而设置的教学环节；对于完成教学计划、落实教学大纲，确保教学质量，培养学生分析问题、解决问题的能力和实际操作技能更具有特别重要的意义。同时，实践教学也是培养应用型人才的重要途径，实践教学质量的好坏，实际上也决定了应用型人才培养质量的高低。因此，加强实践教学环节，提高实践教学质量，对培养高质量的应用型人才至关重要。

近年来，教育部把实验与实践教学作为对高等院校教学工作评估的关键性指标。2005年1月，在教育部下发的《关于进一步加强高等学校本科教学工作的若干意见》中明确指出："高等学校要强化实践育人的意识，区别不同学科对实践教学的要求，合理制定实践教学方案，完善实践教学体系。要切实加强实验、实习、社会实践、毕业设计（论文）等实践教学环节，保障各环节的时间和效果，不得降低要求。"，"要不断改革实践教学内容，改进实践教学方法，通过政策引导，吸引高水平教师从事实践环节教学工作。要加强产学研合作教育，充分利用国内外资源，不断拓展校际之间、校企之间、高校与科研院所之间的合作，加强各种形式的实践教学基地和实验室建设。"

为了配合开展实践教学及适应教学改革的需要，我们在全国各高等院校精心挖掘和遴选了一批在计算机实验与实践教学方面具有潜心研究并取得了富有特色、值得推广的教学成果的作者，把他们多年积累的教学经验编写成教材，为开展实践教学的学校起一个抛砖引玉的示范作用。

为了保证出版质量，本套教材中的每本书都经过编委会委员的精心筛选和严格评审，坚持宁缺毋滥的原则，力争把每本书都做成精品。同时，为了能够让更多、更好的实践教学成果应用于社会和各高等院校，我们热切期望在这方面有经验和成果的教师能够加入到本套丛书的编写队伍中，为实践教学的发展和取得成效做出贡献；也衷心地期望广大读者对本套教材提出宝贵意见，以便我们更好地为读者服务。

清华大学出版社

联系人：索梅 suom@tup.tsinghua.edu.cn

前　言

为贯彻落实教育部《关于进一步加强高等学校本科教学工作的若干意见》和《教育部关于以就业为导向深化高等职业教育改革的若干意见》的精神，加强教材建设，确保教材质量，编写了此教材。

本书分两部分，共11章。第一部分是"数字逻辑电路设计"课程的学习指导，共分6章，是根据鲍可进教授主编的《数字逻辑电路设计》教材的第1章～第6章的主要内容，从要点指导、例题精讲、习题参考答案3个方面进行归纳和总结，对于学生学习该课程具有很好的指导价值。第二部分是"数字逻辑电路设计"课程的实验教程，共分5章，主要介绍数字逻辑电路设计和EDA技术课程实验涉及的相关内容。其中，EDA概述介绍EDA的基本概念、设计流程、设计方法和常用开发工具；VHDL语言概述主要对VHDL语言的语言要素、程序基本结构和语句进行了归纳总结，便于学生的学习；Quartus Ⅱ基本使用方法介绍了Quartus Ⅱ的设计流程、文本输入设计过程和原理图设计方法。基础实验部分以提高学生实际动手能力和工程设计能力为目的，精心选择了15个不同难度的基础实验，供不同专业、不同学时的学生选用；综合设计性实验部分设计了5个典型工程应用设计案例。所有的实验项目在内容安排上由浅入深，循序渐进，便于读者学习和教学使用。

本书可作为高等院校电子信息、通信工程、计算机科学与技术、软件工程、网络工程、自动化等电气信息类专业数字逻辑电路设计课程和EDA技术课程的实验教学用书，同时也可作为高等院校相关专业的教学参考书。

本书由马汉达、赵念强编著，鲍可进主审。其中，第一部分(第1章～第6章)由赵念强编写，第二部分第7章～第10章由马汉达编写，第11章由马汉达、曾宇编写，附录由马汉达编写。鲍可进为本书提供了第1章、第5章的习题答案，赵不贿为本书提供了第4章的习题答案。由于作者水平有限，书中难免存在不当之处，敬请广大读者批评指正。

编　者

2012年5月

目　　录

第一部分　学习指导

第二部分 实验教程

第一部分　学习指导

第1章　数字系统与编码

【学习要求】

本章讲述数字系统中最基础的知识——数制与编码，主要包括各种进制数的表示方法与相互转换、带符号二进制数的编码表示方法及运算、十进制数的二进制编码、各种可靠性编码及字符编码。应重点掌握各种进制数之间的相互转换、真值与3种机器数(原码、反码、补码)之间的相互转换、补码的运算、十进制数与8421码和余3码之间的相互转换以及8421码和余3码的加减运算、二进制数与格雷码之间的相互转换。

1.1　要点指导

1. 数制

1）进位计数制

数制是指用一组固定的符号和统一的规则进行计数的方法。如果按照进位规则计数，则称为进位计数制。任何一种进位计数制都有基数和权两个基本要素。

基数是指某种进位计数制中使用的数字符号的个数。权是指在某种进位计数制中，数字符号根据其所处的位置不同所代表的单位大小。

数有并列表示法和多项式表示法两种表示方法。并列表示法是将各位数字简单罗列的方法。如$(N)_R=(r_{n-1}r_{n-2}\cdots r_1 r_0.r_{-1}r_{-2}\cdots r_{-m})_R$，$r_i\in\{0,1,2,\cdots,R-1\}$，表示有$n$位整数和$m$位小数的$R$进制数据。多项式表示法又称为按权展开式，即将各位的数字与其对应的权相乘然后再求和的一种表示方法。如：

$$\begin{aligned}(N)_R&=(r_{n-1}r_{n-2}\cdots r_1r_0.r_{-1}r_{-2}\cdots r_{-m})_R\\&=(r_{n-1}\times R^{n-1}+r_{n-2}\times R^{n-2}+\cdots r_1\times R^1+r_0\times R^0+r_{-1}\times R^{-1}+\\&\quad r_{-2}\times R^{-2}+\cdots+r_{-m}\times R^{-m})_R\\&=\sum_{i=-m}^{n-1}r_i\times R^i,\quad r_i\in\{0,1,2,\cdots,R-1\}\end{aligned}$$

常用的进位计数制有二进制、八进制、十进制、十六进制等，其特点如表1-1所示。

表1-1　4种常用进位计数制的特点

数制	基数	使用的字符	进位规则	表示形式	权
二进制	2	0、1	逢二进一	$(N)_2=(r_{n-1}\cdots r_0.r_{-1}\cdots r_{-m})_2=\sum_{i=-m}^{n-1}r_i\times 2^i$	2^i
八进制	8	0～7	逢八进一	$(N)_8=(r_{n-1}\cdots r_0.r_{-1}\cdots r_{-m})_8=\sum_{i=-m}^{n-1}r_i\times 8^i$	8^i

续表

数制	基数	使用的字符	进位规则	表示形式	权
十进制	10	0～9	逢十进一	$(N)_{10}=(r_{n-1}\cdots r_0.\ r_{-1}\cdots r_{-m})_{10}=\sum_{i=-m}^{n-1}r_i\times 10^i$	10^i
十六进制	16	0～9，A～F	逢十六进一	$(N)_{16}=(r_{n-1}\cdots r_0.\ r_{-1}\cdots r_{-m})_{16}=\sum_{i=-m}^{n-1}r_i\times 16^i$	16^i

2）数制转换

各种进制数之间的相互转换方法汇总如表 1-2 所示。

表 1-2　进制转换方法汇总

进制转换	方法	要点
α 进制→十进制	按权展开法	第 i 位的权为 α^i，而非 10^i
十进制整数→α 进制	除 α 取余法	直到商为 0 为止，第一次得到的余数为最低位，最后一次得到的余数为最高位
十进制小数→α 进制	乘 α 取整法	直到小数部分为 0 或达到要求的精度为止，第一次得到的整数为最高位，最后一次得到的整数为最低位
α 进制↔β 进制	α 进制↔十进制↔β 进制	以十进制为桥梁进行转换
2^i 进制↔二进制	1 位 2^i 进制↔i 位二进制	牢记 1 位 2^i 进制与 i 位二进制之间的对应关系
2^i 进制↔2^j 进制	2^i 进制↔二进制↔2^j 进制	以二进制为桥梁进行转换，牢记 1 位 2^i、2^j 进制与 i、j 位二进制之间的对应关系

表中 2^i 进制↔2^j 进制的转换常见的是八进制和十六进制之间的相互转换，以二进制为桥梁要比以十进制为桥梁更为方便，但要牢记 1 位 2^i、2^j 进制与 i、j 位二进制之间的对应关系。可以采用按权展开的方法进行记忆。

十进制→α 进制的转换也可以按照“按权累加”的方法直接进行，这样比采用“除 α 取余”法和“乘 α 取整”法更方便快捷。如：

$$\begin{aligned}(169.375)_{10}&=(128)_{10}+(32)_{10}+(8)_{10}+(1)_{10}+(0.25)_{10}+(0.125)_{10}\\&=(10101001.011)_2\end{aligned}$$

2. 编码

1）带符号数的代码表示

这部分首先要掌握真值和机器数的概念。真值是人们直接用“＋”号和“－”号表示数据正、负的一种带符号数的表示方法。机器数是计算机或其他数字系统中有符号数的表示方法，即将“＋”号和“－”号转换成数字“0”和“1”，并将数据位做适当变换（或不变换）的表示方法。根据数据位变换规则的不同，机器数有原码、反码和补码 3 种常见的表示形式，表 1-3 对这 3 种机器数进行了总结。

相对于原码和反码来讲，补码有表示形式唯一、运算速度快等优势，因此在计算机等数字系统中广泛使用。教材中有关补码的快速求法和特殊数据的补码求法需要熟练掌握。另外，要熟练掌握真值、原码、反码和补码之间的相互转换，已知其中任意一种代码，应能熟练

地写出其他代码。代码之间的相互转换方法是相同的,如从负数的原码到补码的转换是数值位变反加 1,符号位不变,那么从负数的补码到原码的转换也是数值位变反加 1,符号位不变。

表 1-3　3 种常见机器数的比较

机器数	原　码	反　码	补　码
编码规则	符号位:"+"和"-"转换成"0"和"1",数值位不变	符号位:"+"和"-"转换成"0"和"1",正数的数值位不变,负数的数值位按位取反	符号位:"+"和"-"转换成"0"和"1",正数的数值位不变,负数的数值位变反加 1
0 的表示形式	$[+0]_{原}=0.00\cdots00$ $[-0]_{原}=1.00\cdots00$	$[+0]_{反}=0.00\cdots00$ $[-0]_{反}=1.11\cdots11$	$[+0]_{补}=0.00\cdots00$ $[-0]_{补}=0.00\cdots00$
n 位代码的表示范围	整数:$-2^{n-1}<N<2^{n-1}$ 小数:$-1<N<1$	整数:$-2^{n-1}<N<2^{n-1}$ 小数:$-1<N<1$	整数:$-2^{n-1}\leqslant N<2^{n-1}$ 小数:$-1\leqslant N<1$
加减运算规则	数值位进行加减运算,符号位需单独处理	符号位一起参与运算,加减运算统一成加法运算,需要循环进位 $[N_1+N_2]_{反}=[N_1]_{反}+[N_2]_{反}$ $[N_1-N_2]_{反}=[N_1]_{反}+[-N_2]_{反}$	符号位一起参与运算,加减运算统一成加法运算,不需要循环进位 $[N_1+N_2]_{补}=[N_1]_{补}+[N_2]_{补}$ $[N_1-N_2]_{补}=[N_1]_{补}+[-N_2]_{补}$

这里还要注意,教材中讲述已知$[N]_{补}$求$[-N]_{补}$时,引入了求补的概念,即:

$$[-N]_{补}=[[N]_{补}]_{求补}$$

这里的求补运算是连同符号位一起变反加 1,不区分数据的正负,与求补码运算是两个不同的概念。

另外,在计算机等数字系统中机器数的表示可以采用不同的位数,如 8 位、16 位等,因此同一个数的真值,因为使用的位数不同会有多种不同的表示。比如教材中讲到 $N=-2^{n-1}$(n 为代码的长度)时,给出的结论是$[N]_{补}=2^{n-1}$,如 $N=-10000$,则$[N]_{补}=10000$。有的同学会提出质疑,因为采用变反加 1 法得到的结论是$[N]_{补}=110000$。这里的问题就是代码位数的问题,教材中这里讲的是 $N=-2^{n-1}$这样的特殊数的补码的求法,而且这里 n 就是代码的位数(包括符号位),而用变反加 1 法求得的$[N]_{补}=110000$ 是用 $n+1$ 位表示的,同样若用 $n+2$ 位表示,则$[N]_{补}=1110000$。

2) 十进制数的二进制编码

十进制数的二进制编码通常称为二-十进制编码,即 BCD 码。常用的 BCD 码有 8421 码、2421 码和余 3 码共 3 种,因 8421 码是其中最常用的一种,所以 8421 码也常简称 BCD 码。按照代码的各位是否有固定的权,BCD 码分为有权码(各位有固定的权,如 8421 码、2421 码等)、偏权码(在有权码的基础上加上一个偏值,如余 3 码)和无权码(如格雷(Gray)码等)3 种类型。

3 种 BCD 码与十进制数之间的相互转换,是以 4 位对应 1 位直接进行变换的。一个 n 位十进制数对应的 BCD 码为 $4n$ 位。

这里应特别强调的是，BCD码不是二进制数，而是用二进制编码表示的十进制数，因此每种BCD码都仅有与十进制的‘0’～‘9’对应的10组有效代码，另外6组为非法代码。

虽然BCD码表示的是十进制数据，但因其编码采用的是相当于1位十六进制的4位二进制，内部运算时也就按照十六进制的进位规则进行。因此用BCD码进行加减运算时，需要对运算结果进行适当的修正。

8421码的加法修正规则是：当两个8421码相加的结果无进位且小于或等于9时，不需要修正(或加0修正)；当相加的结果大于9或有进位时，则该位需加6修正；低位修正的结果使高位大于9或有进位时，则高位也应加6修正。

余3码的加法修正规则是：当两个余3码相加的结果无进位时，和需要减3进行修正，否则和需要加3进行修正。

3）可靠性编码

可靠性代码可以提高信息传输的准确性。有两种方法可以实现代码的可靠性：一种是基于代码本身的某种特征(如相邻代码间仅有一位数字不同)，使得代码在形成过程中不易出错；另一种是代码出错时可以被发现，甚至能对错误进行定位并予以纠正，这种代码称为校验码。

教材讲述了格雷码、奇偶校验码和汉明码3种常用的可靠性编码。格雷码就是具备上述第一种特性的可靠性编码，对于格雷码，学习时应重点掌握其代码的特点、编码方式以及与普通二进制码之间的相互转换方法。奇偶校验码和汉明码属于校验码，其中，奇偶校验码因为只有一位校验位，相对比较简单，掌握其编码方法和校验规则即可。汉明码实际上是多重奇偶校验码(有多个校验位，进行分组奇偶校验)，其功能比奇偶校验码更强大，可以对单个错误进行定位，学习时应重点理解其编码的方法和校验的过程。

4）字符编码

这部分内容重点掌握常见字符的ASCII代码，后续的计算机课程中会经常用到。

数字‘0’～‘9’：30H～39H(后缀H表示是十六进制数据)

大写字母‘A’～‘H’：41H～5AH

小写字母‘a’～‘h’：61H～7AH，与大写字母相差20H

空格：20H

回车：0DH

换行：0AH

1.2　例题精讲

例1-1　将十进制数80.125转换成二进制数和十六进制数。

解：十进制数转换成任意进制数的基本方法是“基数乘除法”，用该方法进行转换的过程如下：

$$
\begin{array}{r|l l}
2 & 80 & \\
2 & 40 & \cdots 0 \\
2 & 20 & \cdots 0 \\
2 & 10 & \cdots 0 \\
2 & 5 & \cdots 0 \\
2 & 2 & \cdots 1 \\
2 & 1 & \cdots 0 \\
 & 0 & \cdots 1
\end{array}
\uparrow
\qquad
\begin{array}{r l}
 & 0.125 \\
\times & \quad\ 2 \\ \hline
 & 0.250 \quad \cdots 0 \\
\times & \quad\ 2 \\ \hline
 & 0.500 \quad \cdots 0 \\
\times & \quad\ 2 \\ \hline
 & 1.000 \quad \cdots 1
\end{array}
\downarrow
$$

$$
\begin{array}{r|l l}
16 & 80 & \\
16 & 5 & \cdots 0 \\
 & 0 & \cdots 5
\end{array}
\uparrow
\qquad
\begin{array}{r l}
 & 0.125 \\
\times & \quad 16 \\ \hline
 & 2.000 \quad \cdots 2
\end{array}
$$

即$(80.125)_{10}=(1010000.001)_2=(50.2)_{16}$。

在掌握基本方法的基础上，针对具体问题可灵活处理，尽可能采用简单、快速的方法。本例采用“按权累加”法更快捷，具体过程如下：

因为$(80.125)_{10}=64+16+0.125=2^6+2^4+2^{-3}$，又因为$2^6=(1000000)_2$，$2^4=(10000)_2$，$2^{-3}=(0.001)_2$，所以$(80.125)_{10}=(1010000.001)_2$。

求出对应的二进制数之后，可以直接按照二进制与十六进制的对应关系，采用“4 位变 1 位”的方法直接求出对应的十六进制数据，具体过程如下：

$$
\begin{array}{ccccc}
\underline{0\,1\,0\,1} & \underline{0\,0\,0\,0} & . & \underline{0\,0\,1\,0} \\
\downarrow & \downarrow & & \downarrow \\
5 & 0 & & 2
\end{array}
$$

所以$(80.125)_{10}=(50.2)_{16}$。

例 1-2　已知 $N=-\dfrac{13}{64}$，求$[N]_补$和$[-N]_补$。

解：一般同学拿到该题后，可能首先会想到先把分数转换成十进制小数，然后再转换成二进制数据，最后再求补码。这种方法步骤虽然正确但比较烦琐，而且有时会因为除不尽而影响转换精度。

实际上，联想到教材中介绍的根据$[N]_补$求$\left[\dfrac{N}{2}\right]_补$的方法，不难找出该题的简便解法。因为 $N=-\dfrac{13}{64}=-\dfrac{13}{2^6}$，所以只要写出十进制$-13$对应的二进制数$-1101$，然后将小数点左移 6 位（整数的小数点默认在最低位的右边），便可得到 N 的二进制小数-0.001101，再求其补码即可得到$[N]_补=1.110011$。

当然也可以通过先求-13，即二进制数-1101的补码$[-1101]_补=1110011$（用 7 位代码表示），然后再将小数点左移 6 位，同样得到$[N]_补=1.110011$。

注意，教材中所讲的已知$[N]_补$求$\left[\dfrac{N}{2}\right]_补$的方法，是在假设$\dfrac{N}{2}$仍然为整数的情况下进行

的，此时将$[N]_{补}$右移一位，保持符号位不变，将最右边的位忽略即可（如略去的是1则有精度损失）。而此处$N=-\frac{13}{64}$为小数，我们是通过左移-13对应的二进制数的小数点进行的（对于定点小数，小数点左移一位即相当于原数据右移一位）。

本例的另外一个问题是，求$[-N]_{补}$可以通过对$[N]_{补}$进行求补运算得到，即：

$$[-N]_{补}=[[N]_{补}]_{求补}=0.001101$$

当然也可以直接对$-N=0.001101$直接求补码得到。这里应注意求补码运算和求补运算是两个完全不同的概念。

例 1-3 已知$[N_1]_{反}=10110101$，$[N_2]_{补}=10000000$，求N_1和N_2对应的十进制真值。

解：前面已经强调，大家要掌握3种机器数与真值之间的相互转换，已知其中任意一种代码，就应该能快速地写出其余代码。任意两种代码之间的相互转换方法是相同的，比如已知原码求反码的方法是符号位不变，负数时数值位取反，那么从反码回到原码也是同样的过程。

本例因为$[N_1]_{反}=10110101$，所以$[N_1]_{原}=11001010$，$N_1=(-1001010)_2=(-74)_{10}$。

对于本例的N_2要特别注意，属于-2^{n-1}（n为代码长度）这样的特殊数字的补码。因为$[N_2]_{补}=10000000$，所以$N_2=(-10000000)_2=(-128)_{10}$。

例 1-4 求二进制数B=1000011.101对应的8421码。

解：8421码是用4位二进制编码表示1位十进制数字的BCD码，所以求二进制数的8421码，就要先将给定的二进制数转换成十进制数，然后再求相应十进制数的8421码。

$$(1000011.101)_2=(67.625)_{10}=(01100111.011000100101)_{8421}$$

例 1-5 求余3码100010101001对应的二进制数，并将所得的二进制数转换为典型的格雷码。

解：余3码同样是用4位二进制编码表示1位十进制数字的BCD码，所以求余3码对应的二进制数，也应先求出其表示的十进制数，然后再转换成二进制数。求出二进制数之后，根据二进制数与典型格雷码之间的转换公式，可求出对应的典型格雷码。

$$(100010101001)_{余3}=(576)_{10}=(1001000000)_2=(1101100000)_{Gray}$$

例 1-6 某机器对标准ASCII代码进行了扩充，用一个字节表示，其中最高位P为奇偶校验位，低7位$X_6\ X_5\ X_4\ X_3\ X_2\ X_1\ X_0$为标准ASCII码的数据位。如果校验位P的生成方式为$P=X_6\oplus X_5\oplus X_4\oplus X_3\oplus X_2\oplus X_1\oplus X_0$，问该ASCII码的校验方式是奇校验还是偶校验。若要改成另外一种校验方式，则P的生成表达式应如何修改？

解：异或运算$\oplus$具有“奇数个1相异或结果为1，偶数个1相异或结果为0”的性质，所以，当数据位$X_6\ X_5\ X_4\ X_3\ X_2\ X_1\ X_0$中含有奇数个1时，P为1，从而使整个代码中1的个数为偶数，所以该机器中ASCII码的校验方式是偶校验。

不难理解，若要改成奇校验方式，则P的生成表达式应为$P=X_6\oplus X_5\oplus X_4\oplus X_3\oplus X_2\oplus X_1\oplus X_0\oplus 1$。

例 1-7 某机器中十进制数采用8421码表示，试给出十进制数加法$87+74=161$用8421码运算的过程。

解：前面已经讲过8421码加法，需要对结果进行修正。方法是：当两个8421码相加的结果无进位且小于或等于9时，不需要修正（或加0修正）；当相加的结果大于9或有进位时，则该位需加6修正；低位修正的结果使高位大于9或有进位时，高位也应加6修正。

本例的运算过程如下：

		1	0	0	0	0	1	1	1	(87)
+)		0	1	1	1	0	1	0	0	(74)
		1	1	1	1	1	0	1	1	
+)						0	1	1	0	结果的低位大于 9，加 6 调整
	1	0	0	0	0	0	0	0	1	
+)		0	1	1	0					使高位产生进位，高位加 6 调整
	1	0	1	1	0	0	0	0	1	表示十进制的 161，结果正确

1.3　习题参考答案

1.

(1) $(4517.239)_{10}=4\times10^3+5\times10^2+1\times10^1+7\times10^0+2\times10^{-1}+3\times10^{-2}+9\times10^{-3}$

(2) $(10110.010)_2=1\times2^4+0\times2^3+1\times2^2+1\times2^1+0\times2^0+0\times2^{-1}+1\times2^{-2}+0\times2^{-3}$

(3) $(325.744)_8=3\times8^2+2\times8^1+5\times8^0+7\times8^{-1}+4\times8^{-2}+4\times8^{-3}$

(4) $(785.4AF)_{16}=7\times16^2+8\times16^1+5\times16^0+4\times16^{-1}+10\times16^{-2}+15\times16^{-3}$

2.

(1) $(1101)_2=(13)_{10}=(15)_8=(D)_{16}$

(2) $(101110)_2=(46)_{10}=(56)_8=(2E)_{16}$

(3) $(0.101)_2=(0.625)_{10}=(0.5)_8=(0.A)_{16}$

(4) $(0.01101)_2=(0.40625)_{10}=(0.32)_8=(0.68)_{16}$

(5) $(10101.11)_2=(21.75)_{10}=(25.6)_8=(15.C)_{16}$

(6) $(10110110.001)_2=(182.125)_{10}=(266.1)_8=(B6.2)_{16}$

3.

(1) $(27)_{10}=(11011)_2=(33)_8=(1B)_{16}$

(2) $(915)_{10}=(1110010011)_2=(1623)_8=(393)_{16}$

(3) $(0.375)_{10}=(0.011)_2=(0.3)_8=(0.6)_{16}$

(4) $(0.65)_{10}\doteq(0.1010011)_2=(0.514)_8=(0.A6)_{16}$

(5) $(174.25)_{10}=(10101110.01)_2=(256.2)_8=(AE.4)_{16}$

(6) $(250.8)_{10}\doteq(11111010.11001)_2=(372.62)_8=(FA.C8)_{16}$

4.

(1) $(78.8)_{16}=(120.5)_{10}$

(2) $(10.375)_{10}=(1010.011)_2$

(3) $(65634)_8=(6B9C)_{16}$

(4) $(121.02)_3=(100.032)_4$

5.

(1) $[+0.00101]_{原}=[+0.00101]_{反}=[+0.00101]_{补}=0.00101$

(2) $[-0.10000]_{原}=1.10000$，$[-0.10000]_{反}=1.01111$，$[-0.100000]_{补}=1.10000$