高级电工培训教材

微机原理及应用

劳动部培训司组织编写

中国劳动出版社

高级电工培训教材

微机原理及应用

劳动部培训司组织编写

中国劳动出版社

26304

(京)新登字114号

微机原理及应用
劳动部培训司组织编写
责任编辑：张秉淑
中国劳动出版社出版
(北京市和平里中街12号)
北京大兴包头营印刷厂印刷
新华书店总店科技发行所发行
787×1092毫米 16开本 16.75印张 413千字
1992年5月北京第1版 1992年5月北京第1次印刷
印数：6800册
ISBN 7-5045-0952-3/TP·007(课) 定价：5.90元

本书是由劳动部培训司组织编写的全国高级电工培训教材。主要内容：微型计算机基础；Z80的常用指令；汇编语言程序设计初步；接口技术；微型计算机的工业应用及单片机。

本书内容少而精，重点突出，原理与应用相互渗透，但又突出应用，适应高级电工培训的要求。

本书也可作为具有一定电子技术基础知识的工人自学用书及工程技术人员参考用书。

本书由肖世锦、郑盛世编写，肖世锦主编；陈连贵、刘美轮审稿，陈连贵主审。

前　言

随着科技进步与经济发展，电气设备使用广泛，特别是自动控制技术的应用推广，一些新型复杂的机电设备日益增多。对这些设备的安装、调试与维修任务越来越大，需要合格的高级电工越来越多。为使培训高级电工的工作逐步规范化，我们会同有关部门和地方组织编写了这套高级电工培训教材。

这套教材的编写是从企业生产实际出发，主要依据《工人技术等级标准》，既考虑到工人的实际技术状况，又适当兼顾今后生产发展的需要，使其不仅满足目前各行业培训高级电工的需要，又为培训对象进一步掌握新知识、新技能奠定基础。本套教材具有工人培训教材的讲求实际、实用、实效的特点。在内容上，努力做到理论与实践紧密结合，操作技能方面以培养工人掌握复杂操作的技能技巧和增强分析、判断、排除各种复杂故障的能力为重点；理论知识方面力求突出针对性、实用性，与技能训练紧密配合。文字叙述尽量做到深入浅出、通俗易懂，可供培训高级电工使用，也可供工人自学使用。

此套教材计有：《电气管理知识》、《微机原理与应用》、《电工基础》、《电子技术》、《电气测量》、《电机原理与维修》、《工厂电气控制技术》、《液压传动》、《工厂变配电技术》、《电气安装技术》、《高级维修电工技能训练》、《高级电工技能训练》等共12种。

教材的编写得到了航空航天部、建设部、轻工部、天津市机械局、上海、江苏、湖南、辽宁、河南、山东省（市）劳动厅（局）的支持。

由于高级工人培训教材的编写，目前尚无成熟经验可循，教学思想、教学内容、教学方法的改革都在研究探讨之中，书中存在一些缺点和不足在所难免。恳切希望广大读者提出宝贵意见，以便在适当的时候进行修订，使之更加完善。

劳动部培训司

绪　言

电子计算机的研制成功及应用开辟了人类进步史的新纪元。世界上第一台电子计算机是1946年在美国研制成功的，虽然这台电子计算机还只是一个雏形（使用了18000个电子管、60000个继电器，重量30吨，耗电150kW，机房面积170m²），然而它却奠定了电子计算机的基础。短短的四十几年，电子计算机经历了电子管、晶体管、集成电路和大规模集成电路四代，发展极为迅速。

70年代以来，随着大规模集成电路技术的发展，出现了把电子计算机的核心部分——中央处理器（运算器和控制器）集成在一个芯片上的微处理器以及存贮器、接口电路等集成器件。将这些集成器件配以少量其他设备（如键盘、显示器等）就组成了微型电子计算机（简称微型计算机或微机）。微型计算机异军突起，很快占领了市场，大有取代原有的小型、中型甚至大型计算机的趋势。微型计算机的发展经历了五个阶段：

第一阶段（1971～1973年）：典型的产品有Intel公司的4004和8008，字长4位或8位，平均指令周期约为20μs，芯片集成度约为2000个晶体管/片。

第二阶段（1973～1976年）：典型的产品有Intel公司8080和Motorola公司的M6800等，字长8位，平均指令周期2μs，集成度约为5000个晶体管/片。

第三阶段（1976～1978年）：典型的产品是Zilog公司的Z80和Apple公司的6502等，字长8位，基本指令周期为1μs，集成度达到10000个晶体管/片。在此期间，还研制出了将一台微型计算机的主要部件——中央处理器、存贮器、接口电路等都集成到一个芯片上的单片微型计算机，如Intel公司的MCS—48，Zilog公司的Z8等。

第四阶段（1978～1981年）：这一阶段的代表产品有Motorola公司的M6800、Zilog公司的Z8000、Intel公司的8086、8088等，字长增加到16位，基本指令周期<0.5μs，芯片集成度达到60000个晶体管/片，性能也有了大幅度的提高，在很多方面已接近或达到小型计算机的水平。

第五代（1982年以后）：典型的产品有Intel公司的IAP×432、HP公司的HP32、Motorola公司的68020等系列，字长32位，指令周期接近0.1μs，芯片集成度高达450000个晶体管/片。由这类微处理器组成的微型计算机，其性能已接近或达到中型计算机的水平。

尽管微型计算机的性能迅速提高，但从目前的应用情况来看，8位机和16位机仍占主要地位，而且在今后若干年内不会有多大的变化，这是因为8位机和16位机在技术上已趋于成熟，硬件结构合理，配套齐全，并已积累了丰富的软件，价格也较低廉，能满足一般应用的要求。

由于微型机具有体积小、功耗低、可靠性高、组装灵活、扩展方便、适应性强等许多优点，加之其价格便宜，因此获得了非常广泛的应用。

微型计算机的应用主要有以下几个方面：

1．科学计算　无论是一般工程设计的计算还是尖端科学的计算，都离不开电子计算机。随着微机性能的不断提高，很多计算任务都能由微机担任。

2. 过程控制　微机技术的不断发展，使工业、交通运输等生产过程广泛采用计算机控制成为可能。用微机进行自动控制、最优化控制，用带有微机的智能仪表进行自动测试、自动记录、自动显示打印，使工业生产的产量提高、质量上升、成本降低、劳动强度减轻，为企业带来了明显的经济效益。例如一台带钢热轧机，改用计算机控制后，产量可达人工控制的一百倍，而且质量显著提高。

3. 数据处理及信息加工　利用微机对大量的数据、信息进行记录、分类、排序、检索、存贮和转换等处理工作。

4. 事务管理　微机用于企业管理时，企业的生产计划制订、生产统计、技术经济分析、经营决策等都用微机来完成，可以用最优的方案对企业进行管理。此外微机还广泛用于财务、仓库、档案、图书、教学等方面的管理以及商业、银行等部门，大大提高了工作效率和质量。

我国微机的研制应用与发达国家相比还有一定的差距，但自从改革开放以来，发展也相当迅速。目前微机的应用已深入到我国国民经济的各个部门，如工业、农业、国防军事、交通运输、航空航海、教学、科研、商业、银行等各方面。特别是在工业生产中，微机已成为技术改造、技术革新的有力工具。国产设备、仪器仪表中也普遍应用微机来提高产品的性能。可见，学习并掌握微机技术，具有十分重大的意义。

《微机原理及应用》是高级电工培训的技术基础课，其内容是微机的硬件、软件及工业应用。本课程的任务是使学生了解微机的基本结构、原理，掌握常用指令、汇编语言程序设计的基本方法以及微机的工业应用，熟悉工业微机控制系统的安装、调试和维修。

本课程的实践性较强，学习本课程时要注意理论紧密联系实际。在学懂原理的前提下，实践是一个关键环节，包括动手编制各种程序，上机调试运行程序，简单微机控制系统接线、调试、检修等，只有通过反复实验和实习，才能真正掌握。

目　录

第一章　微型计算机基础

§ 1-1　微型计算机的数制与码制

一、进位计数制

按进位的原则进行计数的方法，称为进位计数制。在日常生活中，经常使用的是十进制。在计算机中，常使用的是二进制、八进制和十六进制。

1. 十进制

十进制有两个主要特点：

（1）采用十个不同的数字符（数码），即：0、1、2、3、4、5、6、7、8、9。

（2）采用逢“十”进位。同一个数字符号在不同的数位（权位）代表的数值是不同的。例如在555.55这个数中，小数点左边第一个“5”代表个位，其值为5×10^0；小数点左边第二个“5”代表十位，其值为5×10^1；小数点左边第三个“5”代表百位，其值为5×10^2；小数点右边第一个“5”，其值为5×10^{-1}；小数点右边第二个“5”，其值为5×10^{-2}。故可将这个数字写成：

$$555.55=5\times10^2+5\times10^1+5\times10^0+5\times10^{-1}+5\times10^{-2}。$$

又如数字3978.26可写成：

$$3978.26=3\times10^3+9\times10^2+7\times10^1+8\times10^0+2\times10^{-1}+6\times10^{-2}。$$

一般地说，任意一个十进制数S都可以表示为：

$$\begin{aligned}S&=\pm(S_{n-1}\times10^{n-1}+S_{n-2}\times10^{n-2}+\cdots+S_1\times10^1+S_0\times10^0\\&\quad+S_{-1}\times10^{-1}+S_{-2}\times10^{-2}+\cdots+S_{-m}\times10^{-m})\\&=\pm\sum_{i=n-1}^{-m}S_i\times10^i\end{aligned}$$

其中i表示数的某一位；S_i表示第i位的数码，它可是0～9中的任意一个，由具体的数S确定；n和m为正整数，n为小数点左边的位数，m为小数点右边的位数。10^{n-1}、…、10^1、10^0、10^{-1}、…10^{-m}称为十进制数的“位权”，而“10”为十进制数的基数，它表示“逢十进一”，因此，我们将具有“逢十进一”的这种计数制称为十进制。

2. 二进制

与十进制类似，它也有两个主要特点：

（1）只有两个不同的数字符，即：0和1。

（2）采用逢“二”进位。因此，不同的数字符在不同的数位所代表的值也是不同的。例如二进制数 $(1101)_2$和$(110101.101)_2$可分别写成：

$$\begin{aligned}(1101)_2&=1\times2^3+1\times2^2+1\times2^0\\&=8+4+1\\&=(13)_{10}\end{aligned}$$

此处的（　$)_2$和（　$)_{10}$，表示圆括号里的数分别为二进制数和十进制数。以后还会遇到（　$)_8$

和（　）$_{16}$，表示圆括号里的数分别为八进制数和十六进制数。

$$
\begin{aligned}
(110101.101)_2 &= 1\times2^5+1\times2^4+1\times2^2+1\times2^0+1\times2^{-1}+1\times2^{-3}\\
&= 32+16+4+1+0.5+0.125\\
&= (53.625)_{10}
\end{aligned}
$$

一般地说，任意一个二进制数B，都可以表示为：

$$
\begin{aligned}
B &= \pm(B_{n-1}\times2^{n-1}+B_{n-2}\times2^{n-2}+\cdots+B_1\times2^1+B_0\times2^0\\
&\quad +B_{-1}\times2^{-1}+B_{-2}\times2^{-2}+\cdots+B_{-m}\times2^{-m})\\
&= \pm\sum_{i=n-1}^{-m} B_i\times2^i
\end{aligned}
$$

其中，B_i只能取1或0，由具体的数B确定。n和m为正整数，n为小数点左边的位数，m为小数点右边的位数。2是二进制的基数，它表示“逢二进一”，故称为二进制。

3．八进制

类似地，它也有两个主要特点：

（1）八个不同的数码，即：0、1、2、3、4、5、6、7。

（2）采用逢“八”进位。因此，在不同的数位，数码所表示的值是不同的。例如，八进制数（327）$_8$和（5134.72）$_8$可分别写成：

$$
\begin{aligned}
(327)_8 &= 3\times8^2+2\times8^1+7\times8^0\\
&= 192+16+7\\
&= (215)_{10}
\end{aligned}
$$

$$
\begin{aligned}
(5134.72)_8 &= 5\times8^3+1\times8^2+3\times8^1+4\times8^0+7\times8^{-1}+2\times8^{-2}\\
&= 2560+64+24+4+0.875+0.03125\\
&= (2652.90625)_{10}
\end{aligned}
$$

如上所述，对于任意一个八进制数Q，可表示为：

$$
\begin{aligned}
Q &= \pm(Q_{n-1}\times8^{n-1}+Q_{n-2}\times8^{n-2}+\cdots+Q_1\times8^1+Q_0\times8^0\\
&\quad +Q_{-1}\times8^{-1}+Q_{-2}\times8^{-2}+\cdots+Q_{-m}\times8^{-m})\\
&= \pm\sum_{i=n-1}^{-m} Q_i\times8^i
\end{aligned}
$$

其中Q_i可取0～7中的任何一个，由具体的数Q确定；n和m为正整数，n为小数点左边的位数，m为小数点右边的位数。8是八进制的基数，它表示“逢八进一”，故称为八进制。

4．十六进制

它也有两个主要特点：

（1）有十六个不同的数码，即：0～9、A、B、C、D、E、F。其中：A表示10，B表示11…F表示15。

（2）采用逢“十六”进位。因此，在不同的数位，数码所表示的值是不同的。例如，十六进制数(4F)$_{16}$和(17B.3E)$_{16}$可分别表示为：

$$
\begin{aligned}
(4F)_{16} &= 4\times16^1+F\times16^0\\
&= 64+15\\
&= (79)_{10}
\end{aligned}
$$

$$
(17B.3E)_{16} = 1\times16^2+7\times16^1+B\times16^0+3\times16^{-1}+E\times16^{-2}
$$

$$=256+112+11+0.1875+0.0546875$$
$$=(379.2421875)_{10}$$

所以，对于任意一个十六进制数H，可表示为：

$$H=\pm(H_{n-1}\times16^{n-1}+H_{n-2}\times16^{n-2}+\cdots+H_1\times16^1+H_0\times16^0$$
$$+H_{-1}\times16^{-1}+H_{-2}\times16^{-2}+\cdots+H_{-m}\times16^{-m})$$
$$=\pm\sum_{i=n-1}^{-m}H_i\times16^i$$

其中，H_i可取0～F中的任意一个，由具体的数H确定；n和m为正整数，n为小数点左边的位数，m为小数点右边的位数；16为十六进制的基数，它表示“逢十六进一”，故称十六进制。

表 1-1　十进制与二进制、八进制、十六进制对照表

十进制	二进制	八进制	十六进制	十进制	二进制	八进制	十六进制
0	0	0	0	9	1001	11	9
1	1	1	1	10	1010	12	A
2	10	2	2	11	1011	13	B
3	11	3	3	12	1100	14	C
4	100	4	4	13	1101	15	D
5	101	5	5	14	1110	16	E
6	110	6	6	15	1111	17	F
7	111	7	7	16	10000	20	10
8	1000	10	8				

综上所述，对于任意(R)进位计数制其特点为：

（1）有R个不同的数码，即：0～(R－1)。

（2）逢“R”进一。对于R进制数的任意一数N可表示为：

$$N=\pm(N_{n-1}\times R^{n-1}+N_{n-2}\times R^{n-2}+\cdots+N_1\times R^1+N_0\times R^0$$
$$+N_{-1}\times R^{-1}+N_{-2}\times R^{-2}+\cdots+N_{-m}\times R^{-m})$$
$$=\pm\sum_{i=n-1}^{-m}N_i\times R^i$$

其中，N_i可取0～(R－1)中的任意一个；n和m为正整数，n为小数点左边的位数，m为小数点右边的位数；R为R进制的基数，它表示“逢R进一”，故称R进制。

另外，在以后的叙述中，会经常用到二进制数、八进制数、十进制数及十六进制数，为了区别这些不同进位制的数，其结尾通常用字母B、Q、D及H来分别标识它们。如，1011B、27Q、19D及5EH，分别表示1011、27、19及5E为二进制数、八进制数、十进制数及十六进制数。

二、不同进位制数之间的转换

1．十进制数与二进制数之间的转换

（1）十进制整数转换成二进制整数　十进制整数转换成二进制整数，通常采用除2取余法。即将已知十进制数反复除以2，从低位到高位逐次进行，直到商是0为止。若相除之后余数为1，则对应于二进制数的相应位为1；余数为0，则相应位为0。第一次相除得到的余数是

二进制数的低位，最后一次相除所得到的余数是二进制数的高位。

例 1-1　将 $(125)_{10}$转换成二进制数。

解：

$$
\begin{array}{r|l}
2 & 125 \\
\hline
2 & 62 \quad \cdots\cdots \text{余数为 } 1 \\
\hline
2 & 31 \quad \cdots\cdots \text{余数为 } 0 \\
\hline
2 & 15 \quad \cdots\cdots \text{余数为 } 1 \\
\hline
2 & 7 \quad \cdots\cdots \text{余数为 } 1 \\
\hline
2 & 3 \quad \cdots\cdots \text{余数为 } 1 \\
\hline
2 & 1 \quad \cdots\cdots \text{余数为 } 1 \\
\hline
 & 0 \quad \cdots\cdots \text{余数为 } 1
\end{array}
$$

$\therefore (125)_{10}=(1111101)_2$。

（2）十进制纯小数转换成二进制纯小数　十进制纯小数转换成二进制纯小数，通常采用乘2取整法。即将已知的十进制纯小数反复乘以2，从高位到低位逐次进行，直到满足精度要求或乘2后的小数部分是0为止。每次乘2之后，所得新数的整数部分为1，则相应位为1；整数部分为0，相应位为0。第一次乘2所得的整数部分为二进制小数的高位。

例 1-2　将 $(0.825)_{10}$转换成二进制数。

解：

$$
\begin{array}{rl}
0.825 & \\
\times)\quad 2 & \\
\hline
1.650 & \cdots\cdots \text{整数部分为 } 1 \\
0.650 & \\
\times)\quad 2 & \\
\hline
1.300 & \cdots\cdots \text{整数部分为 } 1 \\
0.300 & \\
\times)\quad 2 & \\
\hline
0.600 & \cdots\cdots \text{整数部分为 } 0 \\
0.600 & \\
\times)\quad 2 & \\
\hline
1.200 & \cdots\cdots \text{整数部分为 } 1
\end{array}
$$

如只取四位小数，能满足精度要求，则得

$$(0.825)_{10}\approx(0.1101)_2\ 。$$

由此可见，十进制纯小数不一定能转换成完全等值的二进制纯小数。遇到这种情况，根据精度要求取近似值。

（3）二进制数转换成十进制数　二进制数转换成十进制数的方法是：将二进制数用计数制通用形式表示出来，计算出结果，便得到相应的十进制数。

例 1-3　将 $(110111.101)_2$转换成十进制数。

解：

$$
\begin{aligned}
(110111.101)_2 &= 1\times2^5+1\times2^4+1\times2^2+1\times2^1+1\times2^0+1\times2^{-1}+1\times2^{-3} \\
&= 32+16+4+2+1+0.5+0.125 \\
&= (55.625)_{10}
\end{aligned}
$$

$\therefore (110111.101)_2=(55.625)_{10}$。

2. 二进制数与八进制数之间的转换

（1）二进制数转换成八进制数　二进制数转换成八进制数的方法是：从二进制数的最低位（小数点之左第一位）开始，每三位分为一组，不够三位的用0补足三位，然后把每三位二进制数用相应的八进制数表示就可以了。

例 1-4　将 $(11010111)_2$转换成八进制数。

解：

$$\begin{array}{c}(\underbrace{011}\underbrace{010}\underbrace{111})_2\\(\ 3\ \ \ 2\ \ \ 7\ )_8\end{array}$$

$\therefore (11010111)_2=(327)_8$ 。

二进制小数转换为八进制时，从小数点右边第一位开始，每三位分为一组，最后不足三位的用0补足，然后把每一组二进制数用相应的八进制数表示即可。

例 1-5　将 $(0.01010111)_2$转换成八进制数。

解：

$$\begin{array}{c}(0.\underbrace{010}\underbrace{101}\underbrace{110})_2\\(0.\ 2\ \ \ 5\ \ \ 6\ )_8\end{array}$$

$\therefore (0.01010111)_2=(0.256)_8$ 。

（2）八进制数转换成二进制数　八进制数转换成二进制数，只要将每位八进制数用相应的三位二进制数代替即可。

例 1-6　将$(751.62)_8$转换成二进制数。

解：

$$\begin{array}{c}(\ 7\ \ \ 5\ \ \ 1\ .\ 6\ \ \ 2\ )_8\\(\overbrace{111}\overbrace{101}\overbrace{001}.\overbrace{110}\overbrace{010})_2\end{array}$$

$\therefore (751.62)_8=(111101001.11001)_2$ 。

3. 二进制数与十六进制数之间的转换

（1）二进制数转换成十六进制数　与二进制数转换成八进制数类似，对于二进制整数，只要从二进制数的最低位开始，自右至左，将每四位二进制数分为一组，不足四位时，在左边添0补足四位；对于二进制小数，只要自左至右将每四位二进制数分为一组，不足四位时在右边添0补足四位。然后将每组用相应的十六进制数代替即可。

例 1-7　将$(1110001101.1100001)_2$转换成十六进制数。

解：

$$\begin{array}{c}(\underbrace{0011}\underbrace{1000}\underbrace{1101}.\underbrace{1100}\underbrace{0010})_2\\(\ \ 3\ \ \ \ 8\ \ \ \ D\ .\ \ C\ \ \ \ 2\ \ )_{16}\end{array}$$

$\therefore (1110001101.1100001)_2=(38D.C2)_{16}$。

（2）十六进制数转换成二进制数　十六进制数转换成二进制数，只要将每位十六进制数用四位相应的二进制数表示即可。

例 1-8　将$(19B.3E)_{16}$转换成二进制数。

解：

$$\begin{array}{c}(\ \ 1\ \ \ \ 9\ \ \ \ B\ .\ \ 3\ \ \ \ E\ \ )_{16}\\(\overbrace{0001}\overbrace{1001}\overbrace{1011}.\overbrace{0011}\overbrace{1110})_2\end{array}$$

$\therefore (19B.3E)_{16}=(110011011.0011111)_2$ 。

三、二进制数的运算

1. 二进制的四则运算

（1）二进制加法　二进制加法规则为：

$0+0=0$　　　　$0+1=1$

$1+1=0$（进位1）　　　　$1+1+1=1$（进位1）。

例 1-9　$1101+1011=?$

解：

进　位	1 1 1
被加数	1 1 0 1
加　数	+) 1 0 1 1
和	1 1 0 0 0

∴1101＋1011＝11000。

（2）二进制减法　二进制减法规则为：

0－0＝0　　1－0＝1

1－1＝0　　0－1＝1（有借位）。

例 1-10　11000100－00100101＝?

解：

借　　位	111111
借位后的被减数	1011101
被　减　数	11000100
减　　数	－) 00100101
差	10011111

∴11000100－00100101＝10011111。

计算时先用被减数和借位相运算，得到了考虑借位以后的被减数，然后再与减数相减，最后得到每一位的差以及所产生的借位。

（3）二进制乘法　二进制乘法规则为：

0×0＝0　　1×0＝0

0×1＝0　　1×1＝1

例 1-11　1101×1001＝?

解：

```
       1101
   ×)  1001
   ---------
       1101   ……第一次部分积
      0000
   ---------
      01101   ……第二次部分积
     0000
   ---------
     001101   ……第三次部分积
    1101
   ---------
    1110101   ……最后乘积
```

∴1101×1001＝1110101。

在计算机上实现上述乘法时，为了节省存贮设备，对上述乘法过程作如下修改：

①被乘数左移改为部分积右移。

②当乘数某位为0时不进行加法运算，只进行一次部分积右移操作。

（4）二进制除法　二进制除法规则为：

0÷1＝0　　1÷1＝1

1÷0与0÷0无意义。

例 1-12　1000001÷101＝?

解：

```
          1101
     ---------
101 ) 1000001
      101
      -------
       110
       101
       ------
         101
         101
         ----
         000
```

∴1000001÷101＝1101。

与乘法类似，在计算机上实现上述除法时还要作修改，如将每次做减法时除数的右移修

改为余数的左移。

2. 逻辑运算

逻辑运算是指两个操作数按位进行指定的逻辑操作。常用的有"与"、"或"、"非"、"异或"运算等。

(1)"或"运算(OR) "或"运算的规则为:

$$0 \vee 0 = 0 \qquad 0 \vee 1 = 1$$

$$1 \vee 0 = 1 \qquad 1 \vee 1 = 1$$

例 1-13

$$\begin{array}{r} 10110101 \\ \vee\ 00001111 \\ \hline 10111111 \end{array}$$

"或"运算的功能是:凡与0进行"或"运算的位,结果保持不变;凡与1进行"或"运算的位结果为1。所以逻辑"或"用作对某些位置1,其实质是加。

(2)"与"运算(AND) "与"运算规则为:

$$0 \wedge 0 = 0 \qquad 0 \wedge 1 = 0$$

$$1 \wedge 0 = 0 \qquad 1 \wedge 1 = 1$$

例 1-14

$$\begin{array}{r} 10110101 \\ \wedge\ 00001111 \\ \hline 00000101 \end{array}$$

"与"运算的功能是:凡与0进行"与"运算的位结果为0;凡与1进行"与"运算的位结果保持不变。所以"与"运算用作屏蔽一些位(即使一些位置为0),保留一些位。其实质是乘。

(3)"异或"运算 "异或"运算规则为:

$$0 \oplus 0 = 0 \qquad 0 \oplus 1 = 1$$

$$1 \oplus 0 = 1 \qquad 1 \oplus 1 = 0$$

例 1-15

$$\begin{array}{r} 10110101 \\ \oplus\ 10100100 \\ \hline 00010001 \end{array}$$

"异或"运算的功能是:两个相同的位进行"异或"运算结果为0,否则为1。故可利用A⊕A实现对A清0,也可利用"异或"运算来检验两个操作数是否相同。其实质是比较。

(4)"非"运算 "非"运算规则为:

$$\bar{0} = 1 \qquad \bar{1} = 0$$

"－"表示"非",例如对二进制数01010101进行"非"运算,则得到10101010。其实质是求反。

四、二进制编码

1. 二—十进制编码

在计算机中,二进制数实现容易、可靠,且运算规则也十分简单,所以在计算机中普遍采用二进制。但是二进制数对于人来说不直观,为此在计算机输入和输出时,通常还是采用十进制数。这就要求在输入时,将十进制转换成二进制;输出时将二进制转换成十进制。为了使计算机便于转换,通常将人们习惯的十进制数的每一位写成二进制形式输入到计算机中去,然后由计算机转换成二进制数。

凡是采用若干位二进制数码来表示一位十进制数的方法,统称为十进制数的二进制编码,简称二—十进制编码。这种编码是用四位二进制数表示一位十进制数。但用四位二进制

数可以表示十六种状态，十进制数只有0～9十个数码，因此，用四位二进制数表示一位十进制数时，有六种状态是不用的，究竟取其中哪十种状态，方案是很多的，这里仅介绍标准BCD码（8421码）和余3码。

（1）8421码 这种编码是最自然和最简单的，是选取四位二进制数按计数顺序从0开始的前十种状态与一位十进制数的十个数码相对应。即只采用0000－1001十种状态，1010－1111六种状态不用。而且表示一位十进制数的四位二进制编码中，每位仍然保持二进制数位所有的权位，从左至右为8、4、2、1。8421码的名称由此而来。

例如
$$45=(01000101)_{8421}$$
$$87=(10000111)_{8421}$$

（2）余3码 余3码是由8421码变化而来的，即把每个十进制数字0～9加上3，用这个新数字的8421码作为原十进制数字的代码，就变成了余3码。同一个数字的余3码总比它的8421码多余3，这就是余3码名称的由来。

表 1-2 二—十进制编码

十进制数	8421码	余3码	十进制数	8421码	余3码
0	0000	0011	6	0110	1001
1	0001	0100	7	0111	1010
2	0010	0101	8	1000	1011
3	0011	0110	9	1001	1100
4	0100	0111	10	00010000	01000011
5	0101	1000			

2．字符编码

人和计算机进行通信所用语言不仅是纯数字语言，在多数情况下是字符式的语言，其中包括字符式的数据信息（例如，十六进制数中，用A～F表示十～十五）。这就要求对字符进行编码。

目前，国际上普遍采用的字符编码系统是美国国家信息交换标准字符码，简称ASCII码。

ASCII码总共有128个元素，其中包括32个通用控制字符、10个十进制数码、52个英文大写与小写字母和34个专用符号。因为总共为128个元素，所以用二进制编码需要7位。为了查阅方便，附录一中列出了ASCII字符编码表。

五、带符号数的表示方法

1．机器数与真值

实际的二进制数，是有正、负号的。我们规定用“0”表示正数的符号，用“1”表示负数的符号，这种表示数的方法，称为带符号数的表示方法。而且规定一个数的编码的最高位是符号位。

例 1-16 符号位 数值位

$$Y=(1\underbrace{1011010}_{\text{数值位}})_2=(-90)_{10}$$

这样，连同符号位在内的若干位二进制数称为机器数，它所代表的数值称为机器数的真值。

2．原码