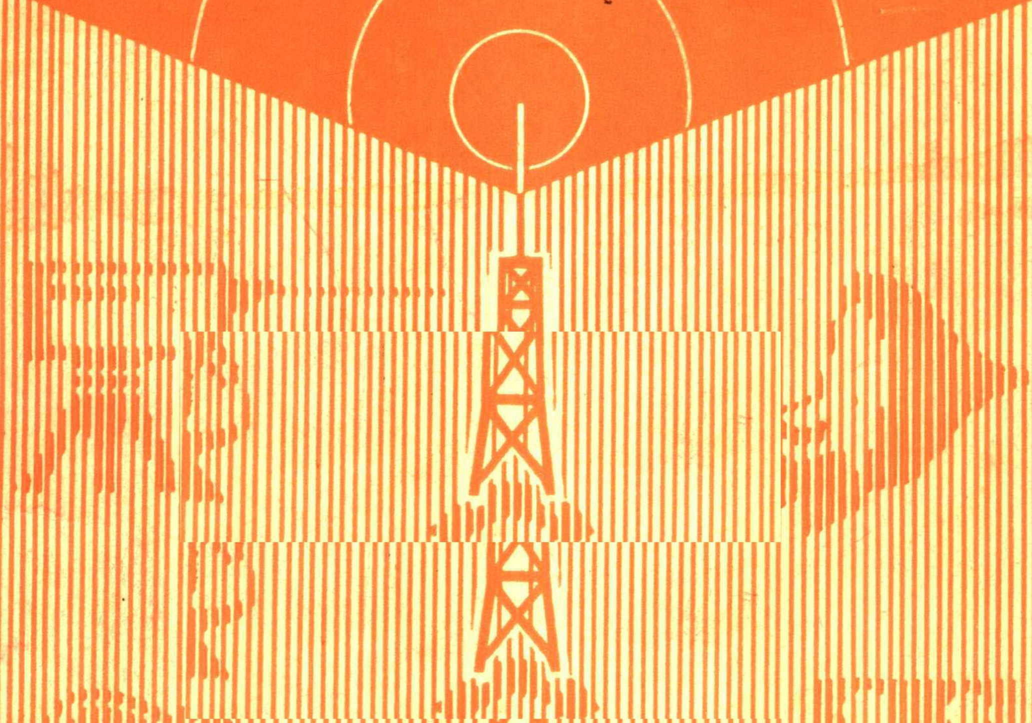
高等学校教材

微型计算机原理

薛钧义　姚燕南



西安电子科技大学出版社

高 等 学 校 教 材

微型计算机原理

(修 订 版)

薛钧义　姚燕南

西安电子科技大学出版社

1992

(陕)新登字 010 号

内容简介

本书共分十二章，主要介绍微型计算机的基本概念和原理．它从教学模型机入门，介绍了整机工作原理；以 Z80 为背景机，从应用角度出发，用软硬件相结合，系统与部件相结合的方法介绍了微型计算机系统中各功能部件的特点和使用方法以及系统的组成方法．本书还以一定篇幅介绍了汇编语言及其程序设计的基本技能，并对单板机、单片机、16 位微处理器 Intel 8086 及 IBM-PC 机作了简要介绍．

本书系高等院校工科自动控制类专业统编教材之一，由计算机与自动控制专业教材编委会推荐再版．本教材也适用于相近专业．由于注意了由浅入深，循序渐进，故也可作为科技人员的自学用书．

高等学校教材

微型计算机原理

(修订版)

薛钧义　姚燕南

责任编辑　谭玉瓦

西安电子科技大学出版社出版

陕西省富平县印刷厂印刷

陕西省新华书店发行　　各地新华书店经售

开本 787×1092　1/16　印张 33　字数 784 千字

1985 年 6 月第 1 版　1990 年 6 月修订版　1992 年 4 月第 8 次印刷　印数 74 001-85 000

ISBN7-5606-0115-4 / TP · 0040(课)　　　定价：8.45 元

出版说明

根据国务院关于高等学校教材工作分工的规定，我部承担了全国高等学校、中等专业学校工科电子类专业教材的编审、出版的组织工作．由于各有关院校及参与编审工作的广大教师共同努力，有关出版社的紧密配合，从1978年至1985年，已编审、出版了两轮教材，正在陆续供给高等学校和中等专业学校教学使用．

为了使工科电子类专业教材能更好地适应"三个面向"的需要，贯彻"努力提高教材质量，逐步实现教材多样化，增加不同品种、不同层次、不同学术观点、不同风格、不同改革试验的教材"的精神，我部所属的七个高等学校教材编审委员会和两个中等专业学校教材编审委员会，在总结前两轮教材工作的基础上，结合教育形势的发展和教学改革的需要，制订了1986～1990年的"七五"(第三论)教材编审出版规划．列入规划的教材、实验教材、教学参考书等近400种选题．这批教材的评选推荐和编写工作由各编委会直接组织进行．

这批教材的书稿，是从通过教学实践、师生反映较好的讲义中经院校推荐，由编审委员会(小组)评选择优产生出来的．广大编审者、各编审委员会和有关出版社为保证教材的出版和提高教材的质量，作出了不懈的努力．

限于水平和经验，这批教材的编审、出版工作还会有缺点和不足之处，希望使用教材的单位，广大教师和同学积极提出批评建议，共同为不断提高工科电子类专业教材的质量而努力．

电子工业部教材办公室

前　言

(第一版)

本教材系由《计算机与自动控制》教材编委会《自动控制》教材编审小组评选审定，并推荐出版.

本教材由西安交通大学薛钧义、姚燕南同志主编，西北工业大学康继昌教授主审. 编审者均依据《自动控制》教材编审小组的编写大纲进行编写和审阅的.

本课程的参考学时数为 90 学时，其主要内容为: 1. 计算机概述(第一、二两章)，介绍了计算机的数制和码制，并从教学模型机入门，讲解了计算机的整机工作原理，以 Z80 为背景机介绍了微型计算机的结构和组成. 2. Z80 微处理器的指令系统、寻址方式、时序波形、汇编语言程序设计、各种 I／O 接口电路以及中断处理和输入／输出技术(第三、七、八、九章). 3. 微型计算机系统及 16 位微处理机简介(第十、十一、十二章). 使用本教材时应注重实践和应用.

本教材由姚燕南同志编写第一、二、三、四、十一章；薛钧义同志编写第五、七、八、九、十二章，并统编全稿. 此外，虞鹤松同志编写了第六章；武自芳、王民培同志也参加了部分章节的编写工作.

承蒙康继昌教授审阅了全部书稿，并提出了许多宝贵意见. 在编写过程中还得到了本校胡保生教授、李人厚及宣国荣副教授的关心和帮助，这里一一表示诚挚的感谢. 由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正.

编者于西安交通大学

1984.6

修订版前言

本教材系由《计算机与自动控制》专业教材编委会《自动控制》教材编审小组评选审定，并推荐再版。

根据《自动控制》教材编审小组的意见，除对原稿基础部分作文字上的压缩和修订外，又新增加如下内容：1. 较新的常用存贮器芯片. 2. Intel 系列串行通讯接口芯片 8251 及其应用. 3. 8 位单片机 MCS-51. 此外，考虑到 Intel 8086 及 IBM-PC 机在国内的应用范围愈来愈大，故将原稿中关于 MC68000 芯片简介改为 Intel 8086 和 IBM-PC 机简介. 关于微型计算机操作系统简介及开发系统简介已分别并入 IBM-PC 机简介和 MCS-51 单片机介绍中，不再另列一章.

本课程参考学时数为 100. 其主要内容为 1. 计算机概述(第一、二章)，介绍计算机的数制和码制，并从教学模型机入门，讲解计算机的整机工作原理；以 Z80 为背景机介绍微型计算机的结构和组成. 2. Z80 微处理机的指令系统、寻址方式、时序波形及汇编语言程序设计(第四、五、六章). 3. 各种存贮器和 I／O 接口电路以及中断处理和输入／输出技术(第三、七、八、九章). 4. 微型机系统及单板机、单片机和 IBM-PC 简介(第十、十一、十二章). 使用本教材时应注重实践和应用.

本教材由姚燕南同志编写第一、二、三、四、十一章；虞鹤松同志编写第六章；薛钧义同志编写五、七、八、九、十、十二章并统编全稿.

该书再版修订过程中得到胡保生教授、李人厚教授及谢剑英副教授的大力支持和指导. 武自芳副教授及张仁强讲师、张彦斌讲师对该书的修订再版也提出不少宝贵意见. 在此表示诚挚的感谢.由于编者水平有限，书中难免还存在一些缺点和错误，殷切希望广大读者批评指正.

编者于西安交通大学　1989.9.12

目　录

第一章 计算机中的数制和码制

客观事物存在着多少和大小的差别，数就是客观事物的量在人们头脑中的反映．一个数可以用不同的计数制度表示它的大小，虽然形式不同，但数的“量”则是相等的．

用一串数字或一串符号表示某个数时，这串数字或符号就是这个数的码或编码．表达一个数的大小和正负的不同方法称为码制。

§1.1 数和数制

一、数的位置表示法及各种进位制数

用一组数字(或符号)表示数时，如果每个数字表示的量不但决定于数字本身，而且决定于它所在的位置，就称为位置表示法．在位置表示法中，对每一个数位赋以一定的位值，称为权．每个数位上的数字所表示的量是这个数字和权的乘积．相邻两位中高位的权与低位的权之比如果是常数,则此常数称为基数.用X表示，则数$a_{n-1}, a_{n-2}, \cdots, a_0, a_{-1}, \cdots, a_{-(m-1)}, a_{-m}$所表示的量$N$为

$$N = a_{n-1}X^{n-1} + a_{n-2}X^{n-2} + \cdots + a_0X^0 + a_{-1}X^{-1} + \cdots + a_{-(m-1)}X^{-(m-1)} + a_{-m}X^{-m}$$

式中从a_0X^0起向左是数的整数部分，向右是数的小数部分．a表示各数位上的数字，称为系数．它可以在 0，1，…，$X-1$ 共X种数中任意取值．m 和 n为幂指数，均为正整数．正由于相邻高位的权与低位的权相比是个常数，因而在这种位置记数法中，基数(或称底数)X的取值不同便得到不同进位制数的表达式。

当$X=10$ 时，得十进制数的表达式为

$$(N)_{10} = \sum_{i=-m}^{n-1} a_i 10^i$$

其特点是：系数a_i只能在 0～9 这十个数字中取值；每个数位上的权是 10 的某次幂；在加、减运算中，采用“逢十进一”“借一当十”的规则。

例如 $(1392.67)_{10} = 1\times10^3 + 3\times10^2 + 9\times10^1 + 2\times10^0 + 6\times10^{-1} + 7\times10^{-2}$

十进制计数制是人们日常生活中最常用的一种计数制．

当$X=2$ 时，得二进制数的表达式为

$$(N)_2 = \sum_{i=-m}^{n-1} a_i 2^i$$

其特点是：系数a_i只能在 0 和 1 这两个数字中取值；每个数位上的权是 2 的某次幂；在加、减法运算中，采用“逢二进一”和“借一当二”的规则。

例如 $(10111.011)_2 = 1\times2^4 + 0\times2^3 + 1\times2^2 + 1\times2^1 + 1\times2^0$

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

二进制计数制中，各数位上的系数只有 0 和 1 两种取值，用电路实现时最为方便，因而它是电子计算机内部采用的计数制．以后还可以看到，除了物理实现方便以外，二进制计数制的运算也特别简单．

当$X=8$时，得八进制数的表达式为

$$(N)_8=\sum_{i=-m}^{n-1}a_i 8^i$$

其特点是：系数a_i只能在 0～7 这 8 个数字中取值；每个数位上的权是 8 的某次幂；在加、减法运算中，采用“逢八进一”和“借一当八”的规则．

例如　$(137.56)_8=1\times 8^2+3\times 8^1+7\times 8^0+5\times 8^{-1}+6\times 8^{-2}$

同理，当$X=16$时，得十六进制数的表达式为

$$(N)_{16}=\sum_{i=-m}^{n-1}a_i 16^i$$

表 1.1　十进制、二进制、八进制、十六进制数码对照表

十进制	二进制	八进制	十六进制
0	0000B	0Q	0H
1	0001B	1Q	1H
2	0010B	2Q	2H
3	0011B	3Q	3H
4	0100B	4Q	4H
5	0101B	5Q	5H
6	0110B	6Q	6H
7	0111B	7Q	7H
8	1000B	10Q	8H
9	1001B	11Q	9H
10	1010B	12Q	AH 或$\bar{0}$H
11	1011B	13Q	BH 或$\bar{1}$H
12	1100B	14Q	CH 或$\bar{2}$H
13	1101B	15Q	DH 或$\bar{3}$H
14	1110B	16Q	EH 或$\bar{4}$H
15	1111B	17Q	FH 或$\bar{5}$H
16	10000B	20Q	10H

其特点是：系数a_i只能在 0～15 这 16 个数字中取值(其中 0～9 这十个数字借用十进制中的数码，10～15 这 6 个数可用两种方法表示，即$\bar{0}$、$\bar{1}$、$\bar{2}$、$\bar{3}$、$\bar{4}$、$\bar{5}$或 A、B、C、D、E、F)；每个数位上的权是 16 的某次幂；在加、减法运算中，采用"逢十六进一"和"借一当十六"的规则．

例如
$$(32AF.EB)_{16}=3\times16^{3}+2\times16^{2}+10\times16^{1}+15\times16^{0}+14\times16^{-1}+11\times16^{-2}$$

八进制计数制和十六进制计数制常常在人们书写计算机程序时被采用．

表 1.1 列出了四种进位制中数的表示法，其中 B 是 Binary 缩写，表示该数为二进制数，Q 表示该数为八进制数(Octal 的缩写应为字母"O"，为区别于数字"0"写为"Q")，H 是 Hexadecimal 的缩写,表示该数是十六进制数,十进制数可用符号 D(Decinal),也可不写．

二、各种进位制数的换算方法

1. 任意进位制数与十进制数之间的相互转换

(1) 任意进位制数转换成十进制数．最简单的方法是根据任意进位制数的表达式按权展开后相加即可．例如

$$1011.110B=1\times2^{3}+0\times2^{2}+1\times2^{1}+1\times2^{0}+1\times2^{-1}+1\times2^{-2}+0\times2^{-3}=11.75$$

$$732.14Q=7\times8^{2}+3\times8^{1}+2\times8^{0}+1\times8^{-1}+4\times8^{-2}=474.187\,5$$

$$3ABF.E6H=3\times16^{3}+10\times16^{2}+11\times16^{1}+15\times16^{0}+14\times16^{-1}+6\times16^{-2}=15\,039.898\,437\,5$$

也可按下列方法将任意进位制数的整数部分和小数部分分开转换．对整数部分：

设 N 为 n 位的任意进位制整数，即

$$N=a_{n-1}X^{n-1}+a_{n-2}X^{n-2}+\cdots+a_{1}X^{1}+a_{0}X^{0}$$

将此式改写如下：

$$N=\{\cdots\{[(a_{n-1}X+a_{n-2})X+a_{n-3}]X+a_{n-4}\}X\cdots+a_{1}\}X+a_{0}$$

由此得转换步骤为：先将最高位乘以基数 X，加上次高位，令其结果为Y_1．再将Y_1乘以 X，加上第三位，结果为Y_2，又将Y_2乘以 X，加上第四位，结果为Y_3．如此一直进行下去，直至加上最低位为止，便得到所要求的任意进制结果．如对上述各例中三个整数部分转换如下：

$$1011B=[(1\times2+0)\times2+1]\times2+1=11$$
$$732Q=(7\times8+3)\times8+2=474$$
$$3ABFH=[(3\times16+10)\times16+11]\times16+15=15\,039$$

对小数部分，设 N 是 m 位的任意进位制纯小数，即

$$N=a_{-1}X^{-1}+a_{-2}X^{-2}+\cdots+a_{-m+1}X^{-m+1}+a_{-m}X^{-m}$$

将此式改写如下：

$$N=X^{-1}\{a_{-1}+X^{-1}[a_{-2}+X^{-1}(a_{-3}+\cdots+X^{-1}(a_{-m+1}+X^{-1}a_{-m}))]\}$$

由此得转换步骤为：先将最低位除以 X，加次低位，令结果为R_1。再将R_1除以 X，加上第三低位，结果为R_2。又把R_2除以 X，加上第四低位，结果为R_3。如此一直进行下去，直到加上最高位后被X除为止，便得到所要求的任意进位制结果。对上述三例中三个小数部分转换如下：

$$0.110\text{B}=2^{-1}[1+2^{-1}(1+2^{-1}\times 0)]=0.75$$

$$0.14\text{Q}=8^{-1}(1+8^{-1}\times 4)=0.187\,5$$

$$0.\text{E6H}=16^{-1}(14+16^{-1}\times 6)=0.898\,437\,5$$

可以看出，两种转换方法得到了同样的结果。

(2) 十进制整数转换为任意进位制整数。设N是要转换的十进制整数，它相应的任意进位制整数共有n位，即

$$N=a_{n-1}X^{n-1}+a_{n-2}X^{n-2}+\cdots+a_1X^1+a_0X^0$$

等式两边同除以基数X，得商Q_1和余数(都为整数)，即

$$\frac{N}{X}=\underbrace{a_{n-1}X^{n-2}+a_{n-2}X^{n-3}+\cdots+a_1X^0}_{Q_1}\qquad \text{余：}a_0$$

余数正是所要求的任意进位制数的最低位a_0，再将Q_1除以X，得商Q_2和余数，即

$$\frac{Q_1}{X}=\underbrace{a_{n-1}X^{n-3}+a_{n-2}X^{n-4}+\cdots+a_2X^0}_{Q_2}\qquad \text{余：}a_1$$

余数正好是任意进位制数的次低位。如此一直进行下去，直到商等于 0 为止，就得到一系列余数，它正好是所要求的任意进位制数各位。例如，若需将 17、289、3 910 分别转换成相应的二进制数、八进制数、十六进制数，则可列竖式进行计算和转换如下：

$$\begin{array}{r|l l l}
2 & 17 & & \\
2 & 8 & \text{余数为 1} & \text{低位} \\
2 & 4 & \text{余数为 0} & \uparrow \\
2 & 2 & \text{余数为 0} & \\
2 & 1 & \text{余数为 0} & \\
 & 0 & \text{余数为 1} & \text{高位}
\end{array}
\qquad
\begin{array}{r|l l l}
8 & 289 & & \\
8 & 36 & \text{余数为 1} & \text{低位} \\
8 & 4 & \text{余数为 4} & \uparrow \\
 & 0 & \text{余数为 4} & \text{高位}
\end{array}$$

∴　17 = 10001B　　　　∴　289 = 441Q

$$\begin{array}{r|l l l}
16 & 3910 & & \\
16 & 244 & \text{余数为 6} & \text{低位} \\
16 & 15 & \text{余数为 4} & \uparrow \\
 & 0 & \text{余数为 15} & \text{高位}
\end{array}$$

∴　3910 = F46H

(3) 十进制小数转换为任意进制小数．设N为要转换的十进制小数，它相应的任意进位制小数共m位，则

$$N = a_{-1}X^{-1} + a_{-2}X^{-2} + \cdots + a_{-m}X^{-m}$$

等式两边同乘以基数，得到

$$XN = a_{-1} + (a_{-2}X^{-1} + \cdots + a_{-m}X^{-m+1}) = a_{-1} + D_1$$

其中a_{-1}为整数部分，它正好等于所要求的任意进位制数的最高位；D_1为所剩的小数部分．若再将D_1乘以X便得到

$$XD_1 = a_{-2} + (a_{-3}X^{-1} + \cdots + a_{-m}X^{-m+2}) = a_{-2} + D_2$$

整数部分正好是所要求的任意进位制数的次高位．如此继续进行下去，直到$D_m = 0$，即可得到所要求的任意进位制小数各位．例如，若要将0.687 5、0.156 25、0.656 25三个十进制小数分别转换为二进制小数、八进制小数和十六进制小数，则可列竖式计算和转换如下：

$$\begin{array}{r l}
 & 0.6875 \\
 & \times\ 2 \\ \hline
\text{高位} & \boxed{1}.3750 \\
 & \times\ 2 \\ \hline
 & \boxed{0}.7500 \\
 & \times\ 2 \\ \hline
 & \boxed{1}.5000 \\
 & \times\ 2 \\ \hline
\text{低位} & \boxed{1}.0000
\end{array}
\qquad
\begin{array}{r l}
 & 0.15625 \\
 & \times\ 8 \\ \hline
\text{高位} & \boxed{1}.25000 \\
 & \times\ 8 \\ \hline
\text{低位} & \boxed{2}.00000
\end{array}
\qquad
\begin{array}{r l}
 & 0.65625 \\
 & \times\ 16 \\ \hline
\text{高位} & \boxed{10}.50000 \\
 & \times\ 16 \\ \hline
\text{低位} & \boxed{8}.00000
\end{array}$$

故得转换结果为：0.687 5 = 0.1011B，0.156 25 = 0.12Q，0.656 25 = 0.A8H．

必须注意，在进行任意进位制数和十进制数的相互转换时，由于整数部分和小数部分的转换方法截然不同，当整数部分和小数部分在形式上相同时，它们的转换结果在形式上却完全不同．如 1101B = 13，但 0.1101B = 0.812 5 ≠ 0.13；25 = 11001B，但 0.25 = 0.01000B ≠ 0.1101B．又如 AH = 10，但 0.AH = 0.625 ≠ 0.10；75 = 4BH，但 0.75 = 0.C0H ≠ 0.4BH．因此，若一个数由整数和小数两部分组成，必须分开进行转换．

此外还必须注意，一个二进制小数能够完全准确地转换成十进制小数，但是一个十进制小数不一定能完全准确地转换成二进制小数．例如，0.1 = 0.000110011001100…B，这就是说十进制小数 0.1 转换成二进制后成为一个无限循环的小数，不能准确地被表示出来．

有时不能用有限位的二进制小数去表示任意一个有限位的十进制小数，这是二进制计数制的一个缺点．

2. 八进制数与二进制数之间的相互转换

由于$8 = 2^3$，故一位八进制数相当于三位二进制数．八进制与二进制之间的相互转换是十分简便的．

首先，若需将一个八进制数转换成二进制数，只要将每位八进制数用三位二进制数表示即可，如

$$467Q = 100110111B$$
$$0.532Q = 0.101011010B$$

其次，若需将一个二进制整数转换成八进制整数，则只要从最低位开始，每三位分为一组，不够三位的以 0 补足三位，然后将每组二进制数分别用相应的八进制数表示即可．如

$$010 \quad 111 \quad 000 \quad 101B = 2705Q$$

最后，若需将一个二进制小数转换成八进制小数，则只要从最高位开始，每三位分为一组，不足三位的以 0 补足，然后把每一组二进制数分别用相应的八进制数表示即可，如

$$0.100 \quad 101 \quad 011 \quad 110B = 0.4536Q$$

3. 十六进制数与二进制数之间的转换

由于$16 = 2^4$，故一位十六进制数相当于四位二进制数，与八进制数相类似，十六进制数与二进制数之间的相互转换也是十分简便的．

首先，对一个十六进制数，不论其整数部分还是小数部分，只要把每一位十六进制数用相应的四位二进制数代替，就可被转换为二进制数．如

$$EFB.3DAH = 1110 \ 1111 \ 1011.0011 \ 1101 \ 1010B$$

其次，对一个二进制数，若将其整数部分由小数点向左，每四位一组，最后不足四位的前面补 0；小数部分则由小数点向右，每四位一组，最后不足四位的后面补 0，而后再把每组四位二进制数用相应的十六进制数代替，即可转换成十六进制数．如

$$0001 \quad 1010 \quad 0101 \quad 1111.1000 \quad 0101 \quad 1100B = 1A5F.85CH$$

在计算机中，数是以二进制形式表示和运算的，但二进制数书写起来太长，易错，通常用八进制或十六进制数来书写．特别在微型计算机中，目前通用的字长为 8 位，它正好可用两位十六进制数表示，故十六进制计数制在微型机中应用十分普遍．

三、二进制数的运算方法

二进制计数制除物理实现简便外，运算方法也较十进制计数制大为简单．下边介绍二进制数的加、减、乘、除运算．

1. 二进制加法

一位二进制数的加法规则为

$$0 + 0 = 0$$
$$0 + 1 = 1 + 0 = 1$$
$$1 + 1 = 0 \qquad \text{进位}1$$
$$1 + 1 + 1 = 1 \qquad \text{进位}1$$

故两个二进制数相加，便可按下列方法进行．如 1011B+1010B:

被加数	1011
加数	1010
进位	+)1 1
	10101

由此可见，两个二进制数相加时，每一位有三个数(本位被加数和加数以及低位来的进位)相加，可按一位二进制数的加法规则得到本位的和及向高位的进位。

2. 二进制减法

一位二进制数减法的规则为

$$0-0=0 \qquad 0-1=1 \quad 有借位$$
$$1-1=0 \qquad 0-1-1=0 \quad 有借位$$
$$1-0=1$$

故两个二进制数相减，如1100 0000B−0010 1010B，过程如下：

被减数	11000000
减数	00101010
借位	−) 11111
	10010110

与加法相类似，每一位有三个数(本位被减数和减数以及从低位来的借位)参加运算，可按一位二进制数的减法规则得到本位的差及向高位的借位。

3. 二进制乘法

一位二进制数的乘法规则为

$$0\times0=0 \qquad 1\times0=0$$
$$0\times1=0 \qquad 1\times1=1$$

只有当两个1相乘时，积才为1，否则积为0。

两个二进制数相乘与两个十进制数相乘类似，可用乘数的每一位去乘被乘数，乘得的中间结果的最低有效位与相应的乘数位对齐，最后把这些中间结果同时相加即可。如1110B×0110B可进行如下：

被乘数	1110
乘数	×0110
中间结果	0000
中间结果	1110
中间结果	1110
中间结果	+) 0000
积	1010100

每一次的中间结果取决于乘数，若乘数的某一位为 1，中间结果便为被乘数；若某一位为 0，则中间结果为 0. 此外，乘数有几位，就有几个中间结果同时进行相加，因而当乘数位数超过 2 时，计算机实现起来便有困难. 实际上计算机内对两个二进制数进行乘法时，常常是采用边乘边移位边相加的办法. 如上例可重作如下：

被乘数	1110	
乘数	+ 0110	
初始部分积	0000	
乘数最低位为 0，加全 0	+ 0000	
部分积	0000	
部分积右移一位	0000	0
乘数次低位为 1，加被乘数	+ 1110	
部分积	1110	0
部分积右移一位	0111	00
乘数第三低位为 1，加被乘数	+ 1110	
部分积	10101	00
部分积右移一位	1010	100
乘数最高位为 0，加全 0	+ 0000	
部分积	1010	100
部分积右移一位得乘积	0101	0100

4. 二进制除法

二进制除法运算与十进制除法运算类似，对整数除法可先从被除数的最高位开始，将被除数(或中间余数)与除数相比较，若被除数(或中间余数)大于除数，则被除数(或中间余数)减去除数，商 1，并将相减之后得到的中间余数左移一位(中间余数的最低位用下一位被除数补充)作为下一次的中间余数. 若被除数(或中间余数)小于除数，则不作减法，商 0，并将本次的中间余数左移一位(中间余数的最低位用下一位被除数补充)得下一位的中间余数. 如此逐次地进行比较、相减和移位，就可得到所要求的各位商数和最终的余数. 如 100110B÷110B 的过程如下：

$$
\begin{array}{r@{\,}l}
 & 000110 \\
\cline{2-2}
110) & 100110 \\
 & 110 \\
\cline{2-2}
 & 0111 \\
 & 110 \\
\cline{2-2}
 & 10
\end{array}
$$

即　100110B÷110B＝110B　余 10B