应用电子计算机编制建筑工程预算

中国建筑科学研究院调查研究室

中国建筑工业出版社

本书是一本应用441B-Ⅲ型电子计算机编制建筑工程预算的应用技术读物。

全书包括两大部分：第一部分介绍441B-Ⅲ型电子计算机的基本知识；第二部分介绍如何设计建筑工程预算程序，并通过程序设计实例，具体介绍从确定设计方案到程序使用等全过程。

本书可供建筑工程预算人员和有关部门的管理人员自学时参考。

应用电子计算机编制建筑工程预算

中国建筑科学研究院调查研究室

*

中国建筑工业出版社出版(北京西郊百万庄)

新华书店北京发行所发行　各地新华书店经售

中国建筑工业出版社印刷厂印刷(北京阜外南礼士路)

*

开本：787×1092毫米 1/16 印张：12 字数：291千字

1980年9月第一版　1980年9月第一次印刷

印数：1— 8,800 册　定价：0.98元

统一书号：15040·3579

前　言

基本建设概预算，是建筑部门安排工程计划、准备施工用料、进行经济核算的一项必不可少的依据，是建筑工程施工管理的一个重要环节。但是，长期以来，由于我国采用手工方法，编制工程概预算，不仅较难及时编出概预算，而且还容易出现漏项、重算和错算，有的地方还由于编制预算的力量薄弱，预算赶不上施工的需要，有时甚至只好“房子造好再算”，使预算成了决算，起不到预算的应有作用。预算编制的不及时和质量不高，使基本建设有关环节的工作无据可循，或依据不准，使国家资金无法控制，施工企业也无法进行正常的管理，给国家经济建设造成了很大损失，影响了多快好省地进行基本建设。

为了充分发挥预算在基本建设中的作用，及时、准确地编出工程预算书，近年来，不少地区和单位，曾经推广应用统筹法编制工程预算，有效地简化了预算编制工作，缩短了编算时间。但是，它仍然不能满足生产建设发展和经济工作越做越细，对预算工作提出的新要求，还不能从根本上解决问题。因此，彻底改变预算的编制方法，成了预算部门进行技术革命和技术革新的一项重要课题。

二十世纪以来，随着电子科学技术的发展，电子计算机已由基本电路采用电子管结构的第一代计算机发展到采用超大规模集成电路的第五代计算机。我国计算机的水平已由第二代的晶体管跨入第三代的集成电路，并正向第四代大规模集成电路迈进。电子计算机的高度发展及其在生产领域中的广泛应用，已不仅可以模拟人的感觉和思维，把人们从大量的繁重的简单的劳动中解放出来，而且可以逾越人体机能的限制，在检测、计算、判断、控制等方面，完成人们无法承担的任务，这就给彻底改革旧的预算编制方法提供了可能。

近几年来的实践表明，电子计算机对于编制工程预算这样一种大量、简单、重复、繁琐的手工计算，是完全有效的。编制工程概、预算的程序一旦设计完成，预算人员只要将图纸尺寸和应选用的定额号，填写在事先设计好的工程初始数据表内，经制成穿孔纸带，即可上机，通过一、二秒钟的运算，在二、三分钟内就能打印出完全符合我们需要的整个工程的工程预算表，大大提高了预算工作效率。

为了促进应用电子计算机编制工程概、预算这方面的工作，进而向建筑管理领域的其它方面发展，我们曾根据几次为预算人员举办441B-Ⅲ型机程序设计学习班讲课材料，编写成讲义（征求意见稿），现在作进一步修订和补充，向初学者提供应用手编程序编制工程预算程序的一些基本知识。

本书共分两部分，第一部分为电子计算机基本知识，为适应建筑安装企业基层人员的需要，在举例中尽可能结合本部分人员的专业实际，在文字上力求通俗，与本专业关系不大的在正文中从略，如有需要，可参阅本书附录。为了部分专业人员深入学习的需要，在这一部分中还增加了少量解释性叙述。第二部分为编制工程预算程序设计实例，介绍一个

程序的完整设计过程。

本书在编写过程中，得到一些单位的支持，在此表示感谢。

由于我们对电子计算机的学习和了解还很粗浅，程序设计的工作实践更是有限，本书内容难免有许多缺点和错误，希望读者批评指正。

编　者

一九七八年三月

目　录

第一篇　电子计算机基本知识

第二篇　工程预算的程序设计

第一篇　电子计算机基本知识

第一章　电子计算机简介

第一节　电子计算机的功用和分类

由于电子计算机具有高速度、自动化和“记忆”的功能，使过去很多无法解决的计算问题和自动控制问题得到了解决。如精确预报天气的计算工作，以往为了预报第二天的天气，计算工作却需要一、两个星期才能完成，应用电子计算机后，几分钟时间就能计算出全部结果；在工业上，电子计算机被用于控制一台机床或一条生产线甚至控制整个工厂的生产过程；在国防上，它可以解决导弹的自动瞄准、自动发射并随时调整飞行方向，等等；至于电子计算机与原子能研究、人造卫星等现代科学技术的关系，就更加密切了。

电子计算机的研究制造，还在不断发展，目前已有的电子计算机种类很多，一般按它的工作原理分，有“模拟计算机”和“数字计算机”两大类。模拟计算机也叫做连续式计算机。在模拟计算机中，所有数据都是用连续变化着的物理量（如电压、电流、轴的旋转角度、长度等）来表示的。数字计算机又叫不连续式计算机，它直接对数字进行算术运算。电子数字计算机中的“数”是用电脉冲之类的物理状态表示的。此外还有数字、模拟混合式的计算机。按计算机的用途来分类，电子计算机基本上可以分成“通用计算机”和“专用计算机”两大类。其中电子数字计算机按照应用和特点，还可以细分成“通用数字计算机”、“数据处理机”、“逻辑机”、“控制计算机”等等。本书着重介绍的国产441B-Ⅲ型机如图1-1-1就是属于通用数字计算机一类。

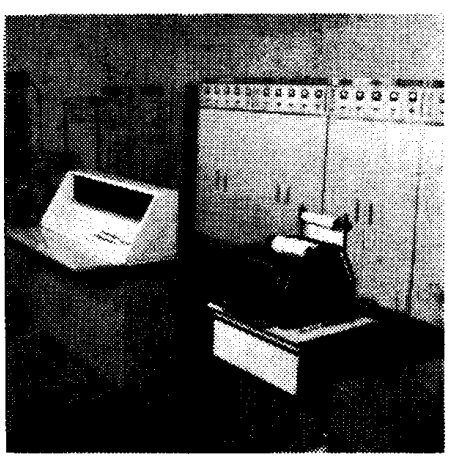

图 1-1-1　441B-Ⅲ型机外貌

电子计算机的计算过程是完全自动化的。当我们把机器准备好之后，只要拍入一个启动命令，机器就能按照事先存入的操作程序自动地执行全部运算，而且在运算过程中还能进行一些人们预先给定的逻辑判断工作，按所规定的指示进行“分析”和“比较”，找出最好的答案。

目前我国自行设计制造的通用数字计算机已有数十种之多，由于各种不同型号的电子计算机的构造不同，所用的指令形式和指令系统内容大部分都不相同，因此给程序设计的学习造成一定的困难（一般都要求针对具体的电子计算机所特定的指令系统作专门的学习）。近年来，为了克服程序设计的这方面困难，并简化程序设计工作，我国已逐步形成若干“语言程序”系统，在一定程都上解决了不同型机之间“语言不通”的障碍，同时大大简化了程序设计。但是，由于生产管理方面应用电子计算机有它自己的特点，学习手编程序仍有一定意义。本书将以国产441B-Ⅲ型通用数字计算机为主，讲解程

序设计的基本问题，介绍手编程序的基本方法。

第二节　电子计算机构造简介

电子计算机为什么能高速度并自动解算各种问题呢？它有哪些部件？它们又是如何进行工作的？

在解决这些问题之前，先让我们分析一下人是怎样算题的。

有这样一个计算问题：

某工程公司月月超额完成施工任务，其中五月份各工程队生产指标超额情况分别是：四个队各超额54%，六个队各超额46%，二个队各超额35%。要求计算全公司十二个队五月份平均的超额指标。

假定由一位统计员用算盘进行这个问题的计算。他算题的过程是：

先按照计算的内容在纸上列出计算式，写上有关数据：

用“A”代表平均超额指标，则

$$A=(4\times0.54+6\times0.46+2\times0.35)\div12$$

第二步，用算盘逐项计算，并将中间结果记录在纸上(或寄放在算盘上指定的地方)，具体是：

$$4\times0.54=2.16\Rightarrow(\text{寄存})2.16$$
$$6\times0.46=2.76\Rightarrow2.76+2.16=4.92$$
$$2\times0.35=0.70\Rightarrow0.70+4.92=5.62$$
$$5.62\div12=0.4683$$

第三步，将算盘上的最后计算结果写在纸上：

$$A=0.4683=46.83\%$$

利用电子数字计算机来计算任何数学问题，运算操作的过程同上面的情况很相似。为了能进行上列各个运算步骤和有关的操作，数字计算机设计了如下一些设备：

（1）**运算器**——进行加、减、乘、除等算术运算的设备。它相当于一把算盘。

441B-Ⅲ型机的运算器包括“累加器”（代号“L”）和“积商寄存器”（代号“S”）等一些组成部分。

（2）**存储器**——是保存和记录原始数据、运算操作指令以及各种运算结果的“记忆”装置。它相当于记录用的纸和笔。

一般电子计算机的存储器能存放成千上万、几十万个数据和指令代码，就象是一座大旅馆，有成千上万个床位。我们把电子计算机的这些“床位”叫做“存储单元”（简称“单元”）。因为它是在主机内部的存储器中，为了同外部设备中的存储器相区别，通常也叫它为“内存单元”。要找住在旅馆里的人，先要知道他住在第几号房间，否则就不容易找到。同样道理，电子计算机的存储单元也按一定规则顺序编上号码，叫做“单元号”或“地址”（这种内存单元的地址号码在441B-Ⅲ型机中是用十六进制的数来表示的）。

（3）**控制器**——是控制运算器自动进行计算操作，并使各部分设备能协调进行工作的一个“总指挥”装置。它相当于计算人员本身。

除了上面三种基本的设备之外，因为机器是按照人们事先给定的运算步骤一“指令”

和有关的数据进行操作的。另一方面，机器算出来的计算结果最后还要表达给人们知道，所以还必须配备：

（4）**输入和输出设备**——作为人与机器之间的桥梁，它们好比是“传递员”和“翻译员”，负责把计算人员的意图送入机器并“翻译”成机器能够识别的“语言”，又将机器所计算的结果和计算过程中产生的问题“翻译”出来给人看。这类设备一般包括光电输入机、电传打字机、窄行打印机、宽行打印机、绘图仪等等。

图 1-1-2　光电输入机

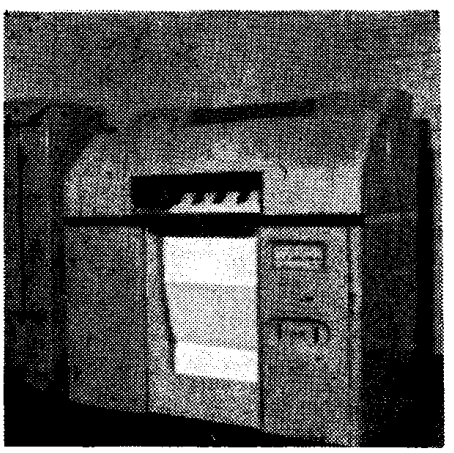

图 1-1-3　宽行打印机

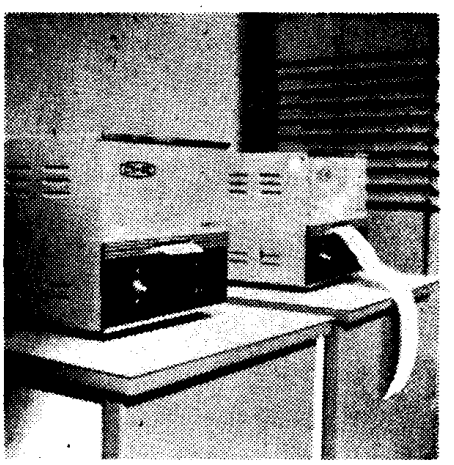

图 1-1-4　窄行打印机

下图表示一般电子数字计算机的组成和各部分设备之间的联系：

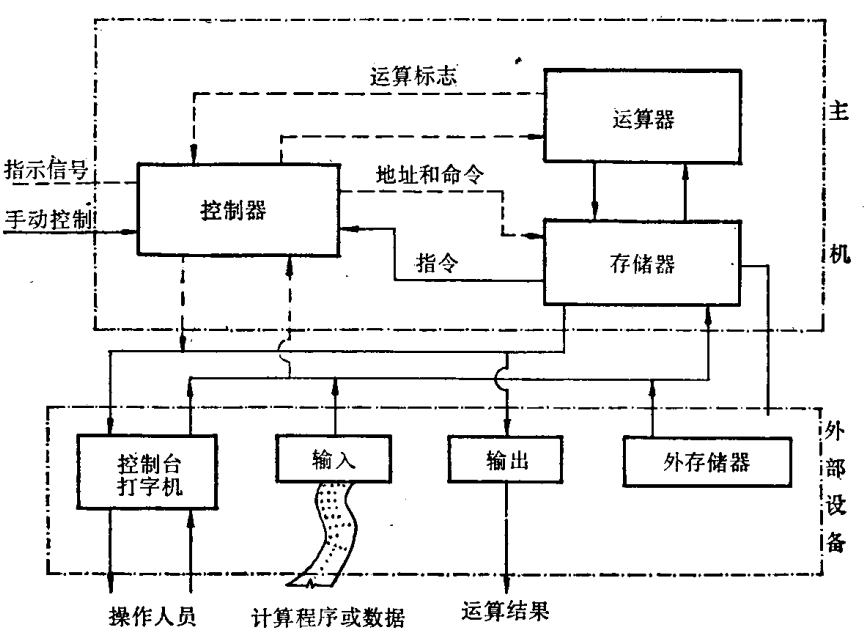

图 1-1-5　电子数字计算机框图

上面框图中的运算器、存储器（指内存储器）、控制器以及输入输出设备是电子计算机的基本组成部分。一般电子计算机还配备有外存储器（如磁带机、磁鼓、磁盘等）。各

部分设备之间的联系，图中实线箭头表示代码传送途径，虚线箭头表示控制信号的传输途径。

控制器、运算器和内存储器构成主机部分，而输入输出设备和外存储器等叫做计算机的**外部设备**。

电子数字计算机的主机设备，主要有以下几类基本电路，如逻辑电路、存储电路、辅助电路等(这些基本电路是由半导体集成电路、厚膜电路、晶体管以及电阻电容等元件组成的)。利用这些基本电路再构成更复杂的电路，如加法器、计数器、译码器、寄存器等等。最后才由这些复杂的电路和基本电路组合为运算器、控制器和存储器这些主要部件。

第二章 数 制

数制是人们在劳动实践中根据实际需要和计算条件逐步形成的。我们所接触到的数制除最常用的“十进制”外还有“十二进制”（十二个为一打）、“十六进制”（旧秤十六两为一斤）、“六十进制”(六十秒为一分，六十分为一小时)等等。选择什么样的数制来表示数，对于电子计算机的性能和设计都有很大的关系。在电子计算机中二进制数制是最基本的数制。根据441B-Ⅲ型机的特点，本章将重点介绍二进制和十六进制，讲解它们的表示形式，彼此之间的换算，它们与常用十进制的关系和换算，以及在机器中数码的表示法。

第一节 电子数字计算机的数制

在日常生活和劳动中，人们最习惯使用的是十进制的数。十进制数就是由十个数符表达的，它们是：

0； 1； 2； 3； 4； 5； 6； 7； 8； 9。

用这些数符，加上其他一些符号，如正负号、小数点等，便可以来表达各种各样的数字。如1976.28就是一个完整的数字。由于长期熟悉了，我们一看这个数字，就知道它是表示“一千九百七十六点二八”。如果细细分析一下，这个数就是：

$$\begin{aligned}1976.28&=1000+900+70+6+0.2+0.08\\&=1\times1000+9\times100+7\times10+6+2\times0.1+8\times0.01\\&=1\times10^{3}+9\times10^{2}+7\times10^{1}+6\times10^{0}+2\times10^{-1}\\&\quad+8\times10^{-2}\end{aligned}$$

通常也可以这么说，“10^{n}”表示了数的“位置”，如10^{3}表示千位数，10^{2}表示百位数，等等。其中的“10”（即十）就表明这种数的“数制”值。在十进制中，数逢十进一（即一个数位满十之后就向比它高的一位进一）。

根据这个道理，可以把任何一个十进制的数目字N概括成如下的表达式：

$$N=\pm[K_{n}\times10^{n}+K_{n-1}\times10^{n-1}+K_{n-2}\times10^{n-2}+\cdots+K_{0}\times10^{0}+K_{-1}\times10^{-1}+K_{-2}\times10^{-2}+\cdots+K_{-m}\times10^{-m}]$$

式中m、n都是正整数，K_{n}、K_{n-1}……K_{-m}则可以是0、1、2、3、4、5、6、7、8、9这十个数符中的任何一个，由实际的数决定其中“10”叫做计数制的基数，也可以说它就是该数制所采用的数字符号的数量。

前面提到过存在有许多种数制，如十二进制、十六进制、二进制等等。这些不同数制的数，与十进制数最基本的区别就是基数不同，它们的基数不是十，而是十二、十六、二等等，只要将“10^{n}”换成相应的基数“R^{n}”，（R是十二、十六、二等等数制的值）那么任何一种数制的数都可以用下面的方式来表示：

$$N=\pm[K_{n}\times R^{n}+K_{n-1}R^{n-1}+\cdots\cdots+K_{-m}\times R^{-m}]$$

式中K_{n}、K_{n-1}、……K_{-m}代表0、1、2……(R－1)等相应数符中的任何一个，R为数制的值。如果是二进制的数，K_{n}、K_{n-1}、……K_{-m}就只代表0和1两个数符，即R＝2，它的进位是逢二进一。

二进制是电子计算机采用的最基本的数制，因此下面先介绍二进制的问题。

按前面的表达式，任何一个二进制数N可以写成：

$$N=\pm[K_n\times 2^n+K_{n-1}\times 2^{n-1}+K_{n-2}\times 2^{n-2}+\cdots\cdots+K_0\times 2^0+K_{-1}\times 2^{-1}+K_{-2}\times 2^{-2}+\cdots\cdots+K_{-m}\times 2^{-m}]$$

其中的K只能是0或1。必须注意的是，这个式子中R＝2，但二进制是没有“2”这个数符的。准确的表示，“二”＝1＋1＝10(逢二进一)。因此上式的正确写法应该是：

$$N=\pm[K_n\times 10^n+K_{n-1}\times 10^{n-1}+\cdots\cdots+K_{-m}\times 10^{-m}]$$

此时的“10”已不是通常概念中的“十”，而是二进制中的“10”（即相当于十进制中的“二”）。因为各种数制的“10”写法无法区别，为了说明方便起见，可用一种办法将各种数制的写法区别开来：

十进制的十写成$(10)_{十}$；

二进制的二写成$(10)_{二}$；

八进制的八写成$(10)_{八}$；

十六进制的十六写成$(10)_{十六}$。

为了进一步了解二进制的有关问题，这里先举例看一看十进制中十个不同的数字是如何用二进制来表示的。

$(0)_{十}=(0)_{二}$　　$(1)_{十}=(1)_{二}$

$(2)_{十}=(10)_{二}$　　$(3)_{十}=(11)_{二}$

$(4)_{十}=(100)_{二}$　　$(5)_{十}=(101)_{二}$

$(6)_{十}=(110)_{二}$　　$(7)_{十}=(111)_{二}$

$(8)_{十}=(1000)_{二}$　　$(9)_{十}=(1001)_{二}$

可能有人要问：人们使用十进制已经习惯，书写也方便，而用二进制写起来既长又难懂，为什么电子数字计算机要采用二进制？回答是，采用二进制对于电子数字计算机的设计制造和使用，有许多重要的优点：

第一，二进制只需两个数字符号（0和1），这就可以利用任何具有两个不同稳定状态的元件来表示数的每一位，给机器的设计制造带来非常大的方便。同时，用二进制，电子计算机中数的存储和传送也可以用比较简单而可靠的表示方法，如“有”与“没有”电脉冲，“正极”与“负极”，“高电位”与“低电位”等等，都可以用来代表“0”与“1”这两个不同的数符。

第二，二进制的数进行四则运算是最简单的。在十进制的算术运算中，必须熟记两数相加和相乘的大量规则（如九九乘规等），而二进制就简单多了。如：

加法：

$0+0=0$

$0+1=1$

$1+1=10$

乘法：

$0\times 0=0$

$0\times 1=0$

$1\times 1=1$

减法：

$0-0=0$

$1-0=1$

$1-1=0$

除法：

$0\div 0$，$1\div 0$ 无意义（非法除）

$0\div 1=0$

$1\div 1=1$

很明显，运算法则越是简单，电子计算机中进行运算的线路就越简单了。

第三，电子计算机采用二进制表示数，还可以节省存储设备。关于这一点这里就不详细说了。

为了既方便机器的设计和制造，又能照顾人的习惯方便，电子计算机的设计考虑了一种“翻译”的功能，使用时，人们仍然可按十进制的方式书写数据，输入机器之后，由机器自动“翻译”成二进制的数，然后才进行运算操作。同样，机器算出结果后，也可以先自动将二进制数“翻译”成十进制的形式才打印出来。

第二节　二进制与十进制

机器是如何进行二进制和十进制的换算呢？下面先介绍二进制换算成十进制的问题。

依照二进制的定义，利用前面关于任意数N的表达式，便可以将任何二进制的数换算成相应的十进制数。例如：

$$\begin{aligned}(10110.01)_{=} &= 1\times2^4+0\times2^3+1\times2^2+1\times2^1+0\times2^0+0\times2^{-1}+1\times2^{-2}\\ &=1\times16+0\times8+1\times4+1\times2+0\times1+0\times0.5+1\times0.25\\ &=16+0+4+2+0+0+0.25\\ &=(22.25)_{+}\end{aligned}$$

这种换算的办法是比较繁琐的，通常人们用手工换算时都是采用一些比较简便的办法，这里介绍其中的一种，即“乘2递加法”和“除2累进法”。

“乘2递加法”用在整数位的换算。它的换算方法是，从最前一位开始，先将该位置的数乘2，再将乘积同下一位的数相加；所加的和又同2相乘，乘积同样与再下一位的数相加，如法顺序往下进行，直到前面的累积同个位数上的数相加，得到的和就是相应的十进制的数。例如：

$(10110)_{=}\Rightarrow(22)_{+}$

$1\times2=2$

$(0+2)\times2=4$

$(1+4)\times2=10$

$(1+10)\times2=22$

$(0+22)=(22)_{+}-$(结果)

或写成

1	0	1	1	0
$\times2$	$+2$	$+4$	$+10$	$+22$
2	2	5	11	$(22)_{+}$
$\times2$	$\times2$	$\times2$		
4	10	22		

“除2累进法”专用于小数位的换算。与“乘2递加法”的不同在于：①顺序相反，是从最后的一位开始顺次向前累进；②把乘2改为除2。例如：

$$(0.011)_{=}\Rightarrow(0.375)_{+}$$

$$1\div2=0.5$$

$$(1+0.5)\div2=0.75$$

$$(0+0.75)\div2=(0.375)_{+}$$

除了上面介绍的方法外，有的人还可以利用八进制（或十六进制）与十进制的换算方法，间接将二进制数换算成十进制的数。

关于十进制换算成二进制的方法，因为利用任意数N表达式的原理进行换算比较难掌握，这里只介绍手工换算时常用的办法，即“除2取余法”和“乘2取整法”。

“除2取余法”用在把十进制的整数换算成二进制整数。方法是将十进制整数连续被2除，每除一次所得的商再给2除，一直进行到所得的商等于“0”为止。而每次除2后

的余数（除不尽时余数为“1”，除尽时余数为“0”），用反方向顺序排列成串，便是所换算的二进制数。例如：

$$(22)_{十}\Rightarrow(10110)_{二}$$

$$\begin{array}{r|l} 2 & 22 \quad \text{余数 } 0 \\ 2 & 11 \quad \text{余数 } 1 \\ 2 & 5 \quad \text{余数 } 1 \\ 2 & 2 \quad \text{余数 } 0 \\ 2 & 1 \quad \text{余数 } 1 \\ & 0 \end{array}$$

将所有的余数按反方向顺序排列出来，就得到二进制的数$(10110)_{二}$。

“乘2取整法”用在小数位的换算。办法是拿十进制的小数乘2，将所得乘积中的整数记录下来，再拿乘积中小数点后的数去乘2，一直进行到乘积中小数点以后的数全部为0结束。最后将所有的整数顺序排列起来，前面加上小数点，便是相应的二进制的数。例如：

$$(0.375)_{十}\Rightarrow(0.011)_{二}$$

方法是：

$$\begin{array}{rl} 0.375 & \\ \times \quad 2 & \\ \hline 0.750 & \text{整数为 } 0 \\ \times \quad 2 & \\ \hline 1.500 & \text{整数为 } 1 \\ \times \quad 2 & \\ \hline 1.000(\text{结束}) & \text{整数为 } 1 \end{array}$$

习　题

1.将下列十进制数换成二进制数：

$(32)_{十}$；　$(128)_{十}$；　$(256)_{十}$；　$(1024)_{十}$；

$(1959)_{十}$；　$(0.25)_{十}$；　$(0.75)_{十}$；　$(92.22)_{十}$。

2.将下列二进制的数，换成十进制的数：

1000000；　10000000；　100000000；

1000000000000；　1111010001111；　0.01；

0.11；　1011100.00111。

第三节　十六进制和二进制、十进制关系

介绍电子计算机的数制和十进制与二进制的换算，目的是让大家了解电子计算机所采用的数制以及各种数制之间换算的基本方法，这对于程序设计工作是有一定关系的，例如程序设计中会遇到少量的常用数需要进行换算，在控制台或输出设备打印出原样的二进制

数码时，都必须会辨认。但大量输入数据或输出的计算结果，都是由机器自动进行“翻译”的，不需要我们处理。

在程序设计时，有一些数，如存储单元地址码的编号，指令中地址码的使用都是要按二进制的方式处理的。但是大家已经感到，用二进制表示一个数写起来会很长，看起来也很吃力，特别是数值比较大的时候，这个矛盾就更突出了。因此，在程序设计中，通常采用另外一些数制来代替二进制作为书写方式，如十六进制、八进制、四进制等等。

为什么要用十六进制之类的方式呢？为了解这个问题，先看看下面几个情况：

$(1)_{+}=(1)_{二}$　　$(2)_{+}=(1,0)_{二}$

$(3)_{+}=(1\ 1)_{二}$　　$(4)_{+}=(1,0\ 0)_{二}$

$(7)_{+}=(1\ 1\ 1)_{二}$　　$(8)_{+}=(1,0\ 0\ 0)_{二}$

$(15)_{+}=(1\ 1\ 1\ 1)_{二}$　　$(16)_{+}=(1,0\ 0\ 0\ 0)_{二}$

从上面的几个数制关系中，可以看到一个规律：十进制1、3、7、15这几个数，用二进制的写法，分别为一位、二位、三位和四位的“1”，当它们各增加1时，都必须相应地再进一位。这说明二进制数与十进制数4、8、16等有一个对应关系，如果采用4，8，16为数制，则它们可以分别用二位二进制、三位二进制或四位二进制的数来代表各自的一个位数。如：

四进制：

$(0)_{四}=(0\ 0)_{二}$

$(1)_{四}=(0\ 1)_{二}$

$(2)_{四}=(1\ 0)_{二}$

$(3)_{四}=(1\ 1)_{二}$

(4)即$(10)_{四}=(1,00)_{二}$或$(01,00)_{二}$

(5)即$(11)_{四}=(1,01)_{二}$或$(01,01)_{二}$

⋮

八进制：

$(0)_{八}=(0\ 0\ 0)_{二}$

$(1)_{八}=(0\ 0\ 1)_{二}$

$(2)_{八}=(0\ 1\ 0)_{二}$

$(3)_{八}=(0\ 1\ 1)_{二}$

$(4)_{八}=(1\ 0\ 0)_{二}$

$(5)_{八}=(1\ 0\ 1)_{二}$

$(6)_{八}=(1\ 1\ 0)_{二}$

$(7)_{八}=(1\ 1\ 1)_{二}$

$(10)_{八}$[即八]$=(1,000)_{二}$或$(001,000)_{二}$

可以用四个二进制数来表示。这样，我们在具体书写时，用十六进制或八进制的方式，写起来数字就可以简短多了，既能适应机器需要的二进制要求，又能简化书写。因此，一般电子计算机程序设计中，通常采用了十六进制或八进制（也有用四进制）作为一种过渡的数制来使书写简化。

下表列出几种常用数制基本数符的对应关系：

各种常用数制数符的对照表

十 进 制	二 进 制	十六进制	八 进 制	四 进 制
一	0 0 0 1	1	1	1
二	0 0 1 0	2	2	2
三	0 0 1 1	3	3	3
四	0 1 0 0	4	4	1 0
五	0 1 0 1	5	5	1 1
六	0 1 1 0	6	6	1 2
七	0 1 1 1	7	7	1 3
八	1 0 0 0	8	1 0	2 0
九	1 0 0 1	9	1 1	2 1
十	1 0 1 0	$\overline{0}$或A	1 2	2 2
十一	1 0 1 1	$\overline{1}$或B	1 3	2 3
十二	1 1 0 0	$\overline{2}$或C	1 4	3 0
十三	1 1 0 1	$\overline{3}$或D	1 5	3 1
十四	1 1 1 0	$\overline{4}$或E	1 6	3 2
十五	1 1 1 1	$\overline{5}$或F	1 7	3 3
十六	1 0 0 0 0	1 0	2 0	1 0 0
十七	1 0 0 0 1	1 1	2 1	1 0 1

从上表中可知，十六进制的十用$\overline{0}$或A表示，十一用$\overline{1}$或B表示，十二用$\overline{2}$或C表示，十三用$\overline{3}$或D表示，十四用$\overline{4}$或E表示，十五用$\overline{5}$或F表示，十六则进位用10表示。

441B-Ⅲ型机采用的是十六进制，因此本节将着重介绍十六进制数及其与二进制、十进制数的换算问题。

（1）十六进制$\rightleftarrows$二进制：

十六进制数与二进制数的转换非常简单，只要熟记上表十六个基本数的对应关系，并按“四位二进制数为一位十六进制数”的规则，便可以解决任何十六进制数与二进制数的换算。例如：

问：$(1\,0\,1\,1\,0\,1)_{二}=(?)_{十六}$

方法：先将二进制数分解成：

$$(1\,0,1\,1\,0\,1)_{二}$$

其中：$(1\,1\,0\,1)_{二}=(D)_{十六}$

$$(1\,0)_{二}=(2)_{十六}$$

因此：$(1\,0\,1\,1\,0\,1)_{二}=(2D)_{十六}$

问：$(1\,1\,0\,1\,0\,1\,1\,1\,0\,0\,0)_{二}=(?)_{十六}$

先分解$(\underbrace{1\,1\,0}_{\text{用十六进制写 6}},\ \underbrace{1\,0\,1\,1}_{B},\ \underbrace{1\,0\,0\,0}_{8})_{二}$

因此：$(1\,1\,0\,1\,0\,1\,1\,1\,0\,0\,0)_{二}=(6\,B\,8)_{十六}$

问：$(7\,A)_{十六}=(?)_{二}$

因为$(7)_{十六}=(0\,1\,1\,1)_{二}$

$(A)_{十六}=(1\,0\,1\,0)_{二}$

所以$(7A)_{十六}=(1\,1\,1\,1\,0\,1\,0)_{二}$

（2）十六进制⇄十进制：

在程序设计中，既然是用十六进制之类的数制来代替二进制数的书写，那么，我们所必须掌握的主要是十六进制与十进制的换算问题。关于这类问题的原理，这里不作详细介绍，下面只谈谈几个常用的换算方法。

我们已经知道，十六进制数中各数位的关系是："十六进一"和"借一当十六"，即高一位数符的一[$(1)_{+六}$]相当于下一位的十六[$(16)_{+六}$]，每一位十六进制数如果满了$(16)_{+六}$，便向高一位进$(1)_{+六}$。根据这些原理，便可以进行十六进制数与十进制数的换算，例如：

问：$(7\ A)_{+六}=(?\ )_{+}$

分析：其中的"7"是十六进制中"10"位数，即$(7\ 0)_{+六}$，把它化作个位数，用十进制数表达的话，就应当是

$$7\times16=(112)_{+}$$

而个位数$(A)_{+六}$是十进制的$(10)_{+}$，

因此，$(7\ A)_{+六}=(70)_{+六}+(A)_{+六}=(112)_{+}+(10)_{+}$

$$=(122)_{+}$$

用这个道理解决十六进制的数换算为十进制数的办法，就是从最高位开始，逐位先乘十六，乘积同下一位相加，它的和又乘十六，再将乘积依次和下一位累加下去，直到"个位"上为止。（关于小数位的问题，因为很少使用，这里不作介绍）。我们叫它为"乘16递加法"。

例 $(B29)_{+六}=(?\ )_{+}$

$$B\times16=176$$

$$(176+2)\times16=2848$$

$$2848+9=2857$$

结果：$(B29)_{+六}=(2857)_{+}$

十进制数换成十六进制的方法，叫"除十六取余法"，就是将十进制的数给十六除，除得的商再给十六除，一直除到"商"比十六小为止，而每次除后除不尽的部分（即余数）逐次记下来，最后倒一个方向将所有余数排列起来，就是所需要的十六进制数。例：

$$(5\ 7\ 3)_{+}=(?\ )_{+六}$$

方法：

```
      3 5
   ---------
16 | 5 7 3
     4 8
     ---
       9 3
       8 0
       ---
       1 3      余数13（即D）
       2
   ---------
16 | 3 5
     3 2
     ---
       3        余数3
```