

FORTRAN 语言程序设计基础

《FORTRAN 语言程序设计基础》编写组 编

内 容 简 介

本书简要介绍 FORTRAN 算法语言及其程序设计的基本问题。全书共分七章。第一章简略介绍有关电子数字计算机的一些常识。第二章至第六章系统讲述 FORTRAN 的基本内容。第七章通过两个例题阐明应用 FORTRAN 解算工程技术问题的程序设计的梗概和要点。

这是一本入门性读物。它的主要读者对象是初学 FORTRAN 语言、并着手用它编写程序的一般工程技术人员。也可用做大专院校相应课程的教材或教学参考书。

FORTRAN 语言程序设计基础

《FORTRAN 语言程序设计基础》编写组 编

*

国防工业出版社 出版

北京市书刊出版业营业许可出字第 074 号

新华书店北京发行所发行 各地新华书店经售

国防工业出版社印刷厂印装

*

787×1092¹/₁₆ 印张 10³/₄ 249 千字

1978年7月第一版 1978年7月第一次印刷 印数：00,001—82,000册

统一书号：15034·1688 定价：1 15元

目 录

第一章 绪论	1
§ 1-1 电子计算机的组成	1
§ 1-2 电子计算机中数的表示及数的运算	5
§ 1-3 电子计算机的基本电路常识及有关术语	9
§ 1-4 数值计算方法的概念	15
§ 1-5 程序设计概念	16
第二章 FORTRAN 语言的基本组成和表达式	19
§ 2-1 FORTRAN 语言的基本组成	19
§ 2-2 表达式	25
第三章 基本语句(上)	31
§ 3-1 简单程序举例及程序书写格式	31
§ 3-2 赋值语句	34
§ 3-3 输入语句	37
§ 3-4 输出语句	39
§ 3-5 FORMAT 语句	42
§ 3-6 STOP 语句和 PAUSE 语句	48
第四章 基本语句(下)	50
§ 4-1 包含有控制语句的程序的例子	50
§ 4-2 GO TO 语句	54
§ 4-3 IF 语句	57
§ 4-4 DO 语句	61
§ 4-5 CONTINUE 语句	67
§ 4-6 输入输出语句的补充	67
第五章 函数和子程序	72
§ 5-1 标准函数	72
§ 5-2 语句函数	74
§ 5-3 子程序	78
§ 5-4 函数和子程序的总结	91
§ 5-5 EXTERNAL 语句	92
§ 5-6 数学库	94
第六章 其他语句及 FORTRAN 语言小结	98
§ 6-1 其他语句和数据块子程序	98
§ 6-2 FORTRAN 语言小结	105
第七章 程序设计举例	111
§ 7-1 飞行器航迹计算程序设计举例	111
§ 7-2 飞行器结构分析程序设计举例	125
§ 7-3 关于改善程序质量的若干意见	153

第一章 绪 论

§ 1-1 电子计算机的组成

为了便于理解电子数字计算机的基本组成，我们首先考查一下人是怎样用纸、笔和算盘进行计算的。例如，要计算 $365 - 30 \times 6 = ?$ 这道算术题，其步骤大致如下：

第一步——考虑好计算方法和次序，并把所用公式和原始数据等写在纸上，

第二步——在算盘上先作乘法，算出 $30 \times 6 = 180$ ，写在纸上，然后再在算盘上打上 365，减去 180，得 185，其中称 180 为中间结果，而称 185 为最后结果，

第三步——把最后结果 185 写在纸上。至此，这道题的计算就完成了。

从上面繁琐的叙述中可以看出，要完成一道算题，必须具有诸如算盘、纸等东西，还要由人的脑子指挥人的手去操作这些东西，这样才能完成这道算题。

上面只是一个很简单的例子，当然我们还可以举出很多类似的例子，其步骤也大体如此。从这些例子中，可以概括出算题所必需的装置如下：

1. 进行运算的装置——例如上面所举的算盘；
2. 存放题目、计算顺序、原始数据、中间结果以及最后结果的装置——例如上面所举的纸和笔；
3. 进行控制的装置——在上面的例子中，人的脑子就是进行控制的装置。

电子数字计算机的基本组成，与上面概括出的三类装置大体相对应，它是由运算器、存储器、控制器以及输入、输出装置等部分组成的，其框图如图 1-1 所示。运算器就是进行运算的装置，它能自动地进行加、减、乘、除及其他运算。存储器能记存题目、计算顺序、原始数据、中间结果和最后结果。控制器就是进行控制的装置，它部分地代替了人脑的工作，能按照人事先规定的计算顺序或根据中间计算结果，决定下一步的计算。输入装置和输出装置是实现人与计算机之间交际的设备。除以上基本组成部分外，还必须有电源、控制台等组成部分。在

图 1-1 中，实线箭头表示“数”的传送方向，虚线箭头表示“控制命令”的传送方向。

现在，我们仍以 $365 - 30 \times 6 = ?$ 这道算题为例，用图 1-1 来说明电子数字计算机的工作过程：

第一步——由输入装置将计算程序（即事先编好的计算顺序等）和原始

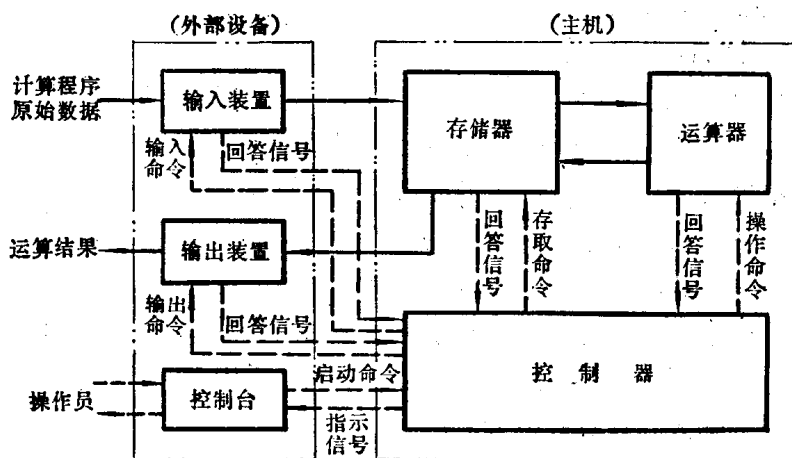


图1-1 电子数字计算机的组成框图

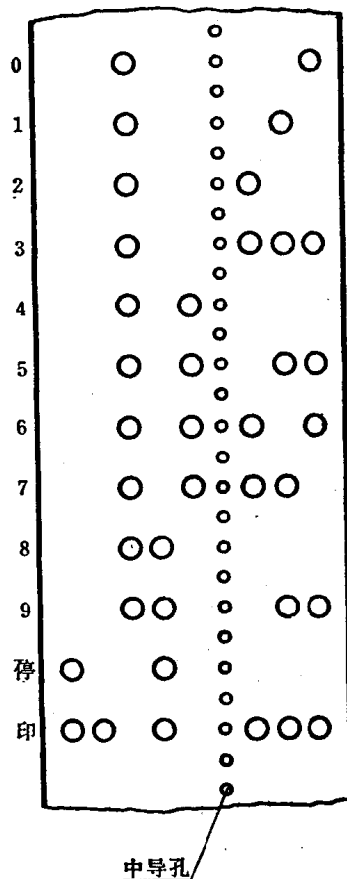
数据 365、30、6 输入到存储器存放起来；

第二步——由控制台启动计算机，在控制器的控制下，按计算程序自动完成以下操作：从存储器取出被乘数30、乘数6送到运算器进行乘法操作，求得乘积180，将运算器中所得的中间结果180送回到存储器；从存储器取出被减数365、减数180到运算器，进行减法操作，求得减法结果185，将运算器中所得到的最后结果送回存储器；

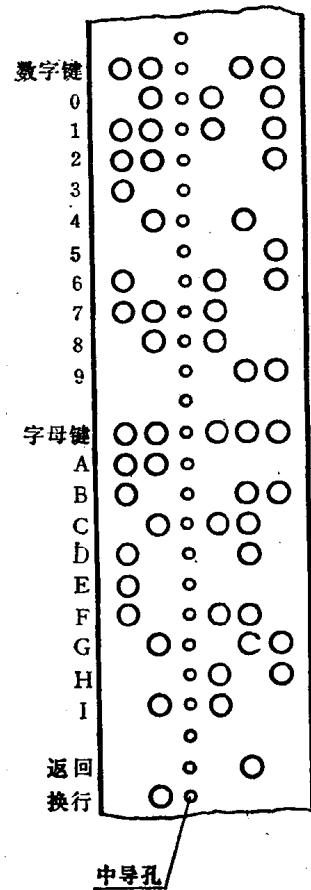
第三步——由输出装置将最后结果185打印在纸上。本题计算完毕，可以停机。

电子计算机是由专门的工作人员进行操作的，作为需要算题的技术人员，他只要按照规定的算法语言（或称程序语言）编写好计算程序并给出原始数据，就可送去计算。因此，本书并不介绍计算机，而只是提一提有关计算机的一些基本概念，以便于理解算法语言和编写程序的规定。由于输入、输出装置与用户的关系比较密切，因此下面将简略介绍一下一般计算机的输入、输出装置。

目前常用的输入方式是穿孔纸带输入和穿孔卡片输入。穿孔纸带如图1-2所示，一般是黑色的纸带，中间有一行小孔，称为同步孔（或中导孔），每横排可以穿八个



(a) 八单元纸带



(b) 五单元纸带

图1-2 穿孔纸带

孔(八单元纸带)或五个孔(五单元纸带)，孔的多少和有无是根据二进制编码穿制的，每横排代表一个字符，这些字符就组成我们编写的计算程序和数据。输入装置有各种型式，通常都是利用光电效应制造的，图1-3就是一种光电纸带输入机的原理示意图。输入时，纸带卷在卷筒上，通过传动系统使它向另一卷筒上运动，当纸带经过光束时，纸带上有一个孔就通过一束光线，使光敏元件导电，表示二进制的1；无孔则不通过光线，表示二进制的0。光敏元件组的导电情况取决于纸带上孔的多少和排列方式，这样就把计算程序和数据输入到计算机中去了。

另一种输入方式是穿孔卡片输入。图1-4是FELIX C-256计算机上使用的输入卡片，上面穿上了该机所用的字符，即数字0、1、2、3、……9，英文字母A；B、C、……Z，专用

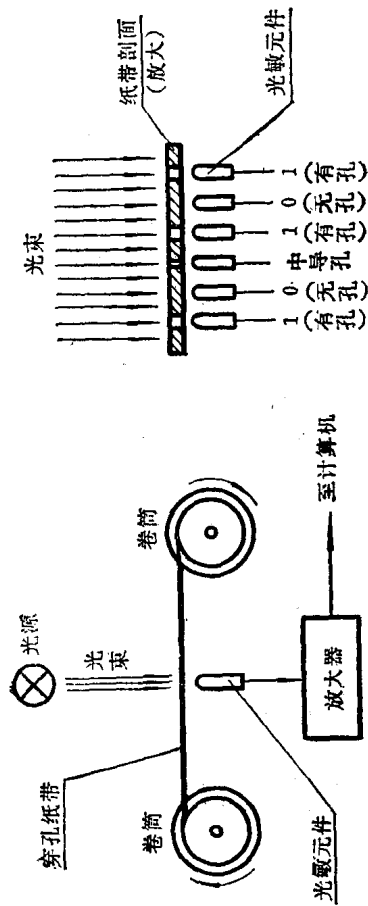


图1-3 光电纸带输入机原理示意图

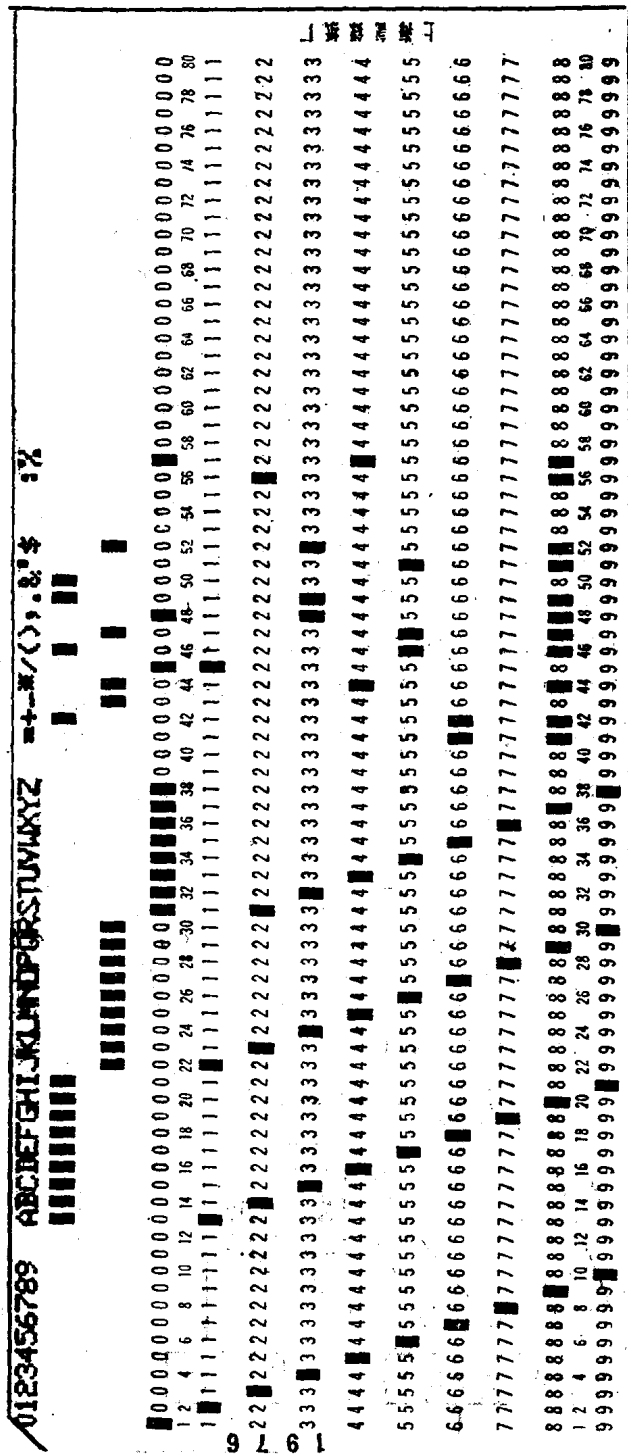


图1-4 穿孔卡片

字符=、+、-、*等。卡片划分为80列12行，列从左向右编号，依次为1、2、3、……80，行从上向下编号，依次为12、11、0、1、……9。行与列的交点处即为可穿孔的位置，每个位置上均可穿一个矩形孔，如果穿满了总共可穿960个孔。每一列上可穿一个、两个或几个孔，代表一个字符。字符的穿孔（代码）组合将在第二章中介绍。在卡片上除12、11行外，每行都印上了数码。

穿孔卡片是由读卡机自动输入的。用FORTRAN语言编写的计算程序和数据，按照规定的代码预先在卡片上穿孔，计算时将穿好孔（需经过校对，保证无误）的卡片置入读卡机，启动机器，就将计算程序和数据输入到计算机了。

同纸带相比较，卡片的优点是便于阅读，便于更改和修正错误，便于保存。其缺点是卡片价格较纸带要高些。

目前常用的输出装置有行式打印机、电传打字机、电灼印刷机等。下面我们介绍行式打印机的一般原理。行式打印机的种类虽然很多，但其工作原理基本相同，图1-5是它的原理示意图。其主要组成部分有：字符轮2、榔头6、电磁铁8、走纸链轮5及其控制机构、编码盘11、字符控制电路、位控制电路以及打印机控制电路等。字符轮表面上制有阳文字符（数字、字母、专用字符等）。例如，132行64字符的宽行打印机，其字符轮的圆周方向制有64个字符，字符轮轴向每排有132个相同的字符。沿字符轮轴向有132个榔头，每一个榔头对准一个字符的位置。它可由电磁铁的推动打击字符轮。在字符轮与榔头之间装有色带和白纸。字符轮旋转，榔头按一定规律打击，就可把欲打印的字符打印在白纸上。每打印一排，字符轮旋转一周，走纸控制机构即将纸向前推进一格（也可推进两格——隔行打印，或不推进——迭印），色带则由专门的机构操纵，使其自动往返运行。例

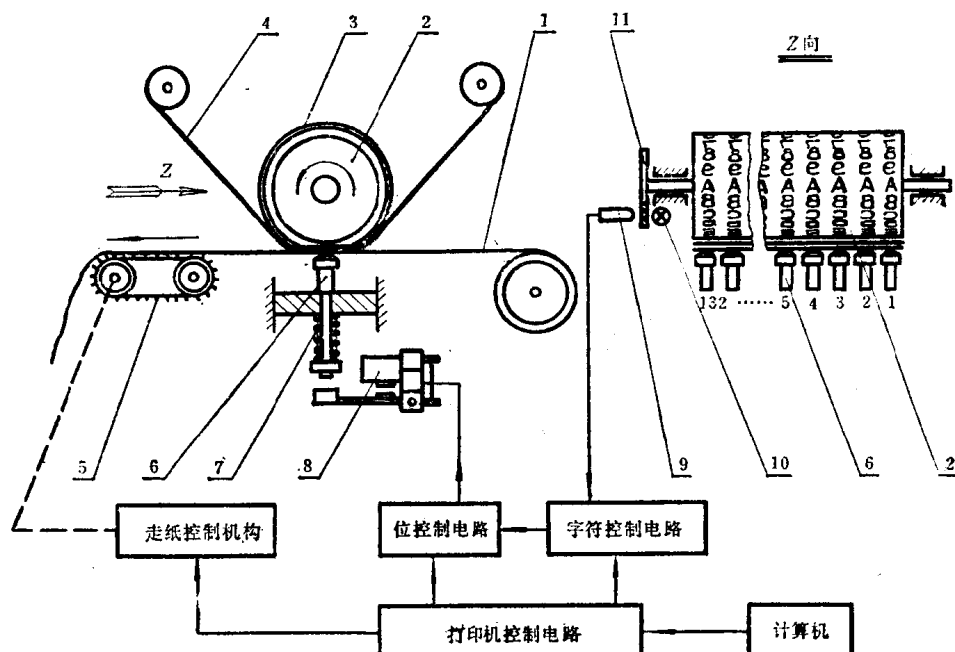


图1-5 行式打印机原理示意图

1—纸；2—字符轮；3—轮罩；4—色带；5—走纸链轮；6—榔头；7—压缩弹簧；8—电磁铁；9—光敏元件；10—光源；11—编码盘。

如,要打印 $A=7097$ 这一排字符,其过程如下:计算机将这一排字符的信号送到打印机控制电路中的缓冲寄存器中;当字符轮旋转到 A 时,由编码盘通过光敏元件给出 A 的代码信号输送到字符控制电路,与缓冲寄存器中的字符信号相比较,当其相一致时,便发出信号给位控制电路,位控制电路使电磁铁通电,于是推动第 1 号榔头打击一次,这样就把 A 打印在纸上了;假如字符在字符轮上沿圆周方向的排列次序为 $A、B、C、\dots、0、1、2、\dots、+、-、\times、\div、=、\dots$,那么,当字符轮旋转到 $B、C、\dots$ 等处时,榔头不打击;当它旋转到 0 时,第 4 号榔头打击一次;旋转到 7 时,第 3 号和第 6 号榔头打击一次;旋转到 9 时,第 5 号榔头打击一次;旋转到字符 $=$ 时,第 2 号榔头打击一次;其他位置不打击。于是,在字符轮旋转一周的过程中,打印出一行字符 $A=7097$ 。其他行的打印过程与此类似。

§ 1-2 电子计算机中数的表示及数的运算

一、电子计算机中数的表示

我们通常都习惯于十进制的计数方法。用 $0、1、2、\dots、9$ 这十个数字的不同组合,可以表示任意一个数。在十进制中共有十个基数,由低位向高位的进位是逢十进一。电子计算机中数的表示方法采用的是二进制,它只有两个基数,即 0 和 1 ,由低位向高位的进位是逢二进一。初看起来似乎很别扭,其实也不陌生,在我们的日常生活和工作中也遇到非十进制的计数方法,例如,一小时等于 60 分,一分等于 60 秒,这是六十进制;还有十六进制——旧秤十六两是一斤;二进制——一双鞋等于两只鞋;等等。

为什么电子计算机中采用二进制呢?这主要是因为二进制基数只有两个,便于用物理元件的不同稳定状态来表示。例如,电灯的亮和暗,电位的高和低,脉冲的有和无,磁性物的正磁化和反磁化等,都可以用来表示二进制的 1 和 0 。显然,这种物理元件在技术上是比较容易实现的。如果采用十进制,那么就要求一个物理元件具有十种不同的稳定状态,这样的物理元件必定很复杂,用这种元件制造出的电子计算机必然十分庞大。另外,二进制数的运算规则比较简单,这也是电子数字计算机采用二进制的原因之一。任何事物都是一分为二的,二进制虽然有以上优点,但它的位数较多,例如十进制数 79 ,用二进制表示则为 1001111 ;另外,人们对二进制用起来很不习惯。为克服以上缺点,在计算机中有的采用 8 进制或 16 进制。另外,大多数电子数字计算机还能够进行二进制与十进制之间的自动转换,解决了使用二进制不习惯的问题。人们可以按十进制编制计算程序并给出数据,输入时计算机能自动将其转换为二进制,在输出时又能自动将二进制的结果转换为十进制数,并打印在纸上,这对使用者自然是十分方便的了。

用 FORTRAN 语言编写计算程序时,并不需要使用二进制,但有时会遇到二进制的数。因此,作为一般常识,下面写出十进制数和二进制数的一般表达式,以供对照。

数 N 用十进制表示可写为:

$$N = a_n a_{n-1} \dots a_1 a_0 . a_{-1} a_{-2} \dots a_{-m}$$

或

$$N = \sum_{k=-m}^n a_k 10^k$$

式中 a_k ——0, 1, …, 9 中的某一数;

m ——小数点之后的总位数;

n ——小数点之前的总位数减去 1。

例如, $m=3$, $n=4$, $a_{-3}=2$, $a_{-2}=3$, $a_{-1}=4$, $a_0=9$, $a_1=7$, $a_2=4$, $a_3=6$, $a_4=1$, 则

$$N = \sum_{k=-3}^4 a_k 10^k = 1 \times 10^4 + 6 \times 10^3 + 4 \times 10^2 + 7 \times 10^1 + 9 \times 10^0 + 4 \times 10^{-1} \\ + 3 \times 10^{-2} + 2 \times 10^{-3} = 16479.432$$

又如, 十进制数 1977.125 可以写成和式的形式如下:

$$1977.125 = 1 \times 10^3 + 9 \times 10^2 + 7 \times 10^1 + 7 \times 10^0 + 1 \times 10^{-1} + 2 \times 10^{-2} + 5 \times 10^{-3}$$

套一下公式可得出: $m=3$, $n=3$, $a_{-3}=5$, $a_{-2}=2$, $a_{-1}=1$, $a_0=7$, $a_1=7$, $a_2=9$, $a_3=1$ 。

二进制数可仿照上面十进制数的表达式写出其表达式如下:

$$N = \sum_{k=-m}^n a_k 2^k$$

式中 a_k ——0, 1 中的某一数;

m ——小数点之后的总位数;

n ——小数点之前的总位数减去 1。

例如, 有一个二进制数 10101.01, 问它相当的十进制数是多少? 利用上面的表达式可得

$$10101.01 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\ + 0 \times 2^{-1} + 1 \times 2^{-2} \\ = 16 + 0 + 4 + 0 + 1 + 0 + \frac{1}{4} = 21.25$$

利用上面的表达式, 可写出二进制数与十进制数的对应关系, 如表 1-1 所示。

表 1-1 二进制与十进制数的对应关系

整 数		小 数	
二 进 制	十 进 制	二 进 制	十 进 制
0	0	0.1111	0.9375
1	1	0.1110	0.8750
10	2	0.1101	0.8125
11	3	0.1100	0.7500
100	4	0.1011	0.6875
101	5	0.1010	0.6250
110	6	0.1001	0.5625
111	7	0.1000	0.5000
1000	8	0.0111	0.4375
1001	9	0.0110	0.3750
1010	10	0.0101	0.3125
1011	11	0.0100	0.2500
1100	12	0.0011	0.1875
1101	13	0.0010	0.1250
1110	14	0.0001	0.0625
1111	15		
10000	16		

二、电子计算机中数的运算

电子计算机中数的运算以二进制的算术运算为基础，运算规则很简单。下面我们简略介绍一下它的四则运算。

1. 加法 二进制的加法规则如下：

$$0 + 0 = 0, 0 + 1 = 1, 1 + 0 = 1, 1 + 1 = 10。$$

例：求 $27 + 10 = ?$

	十进制	二进制
被加数	27	11011
加 数	+ 10	1010
结 果	37	100101

2. 减法 二进制的减法规则如下：

$$0 - 0 = 0, 1 - 0 = 1, 1 - 1 = 0, 10 - 1 = 1。$$

例：求 $37 - 27 = ?$

	十进制	二进制
被减数	37	100101
减 数	- 27	11011
结 果	10	1010

3. 乘法 二进制乘法比十进制乘法简单得多，其规则如下：

$$0 \times 0 = 0, 0 \times 1 = 0, 1 \times 0 = 0, 1 \times 1 = 1。$$

例：求 $13 \times 5 = ?$

	十进制	二进制
被乘数	13	1101
乘 数	× 5	× 101
结 果	65	1101 0000 1101 1000001

从上例可见，做二进制乘法时，若乘数为1，则将被乘数照写一遍，其最右边的一位应与相应的乘数位对齐；若乘数为零，实际上没有任何作用。所有乘数位算完后，把各部分乘积加起来即得所求结果。由此可见，二进制的乘法实质上是由加法和移位两种操作来实现的。

4. 除法 二进制除法运算是乘法的逆运算，它是由减法和移位两种操作来实现的。

例：求 $65 \div 5 = ?$

十进制	二进制
13	1101
5) 65	101) 1000001
5	101
15	11001
15	101
0	101
	101
	0

总之，二进制数的四则运算可归结为加法、减法和移位三种操作。为简化设备，在电子数字计算机中，往往不直接进行减法，而是用加负数的方法来代替减法。这样一来，四则运算就简化为加法和移位两种操作了。

电子计算机中怎样表示数的正负，怎样用加负数的方法来代替减法呢？下面简单地加以介绍。

在电子数字计算机中表示数的正负不是在其前面冠以+、-号（因为机器中只能用0和1这两种符号）。而是在二进制数前面增加一位符号位，当符号位为0时，表示此数为正的；当符号位为1时，则表示此数为负的。例如，+37和-27表示为二进制时写为0·100101和1·011011，其中圆点前面的0和1就是符号位。

加负数是通过加补码的办法来实现的。

一个二进制负数可以有三种不同形式的编码，即原码、反码和补码。例如上面写的-27——1·011011，就是原码。所谓反码，就是将原码中除符号位以外其它位数的1换成0、0换成1所得到的编码。例如-27的反码是1·100100。在反码的最低位上加1所构成的编码称为补码。--27的补码是1·100101。

怎样用加补码的办法实现减法？下面以 $37-27=?$ 这道题为例简述如下。

+37的二进制数	0·100101
-27的二进制补码 (+)	1·100101
	10·001010
此位被丢掉	↑

在计算机中符号位为最高位，再往前进位就被丢掉了，因为没有设备给予保证。因此，上例所得结果实际上是0·001010，即+10（十进制），显然是正确的。

当两数相减的结果为负数时，上述方法仍然适用，但其结果为补码形式，应将其变为原码，才便于阅读。

例：求 $27-37=?$

+27的二进制数	0·011011
-37的二进制补码 (+)	1·011011
所得结果的二进制补码	1·110110
(-)	1
所得结果的二进制反码	1·110101
所得结果的二进制原码	1·001010

001010

所得结果为 1.001010, 即十进制数的 -10, 显然是正确的。

§ 1-3 电子计算机的基本电路常识及有关术语

一、电子计算机的基本电路常识

在电子数字计算机中通过什么样的电路来实现对二进制数的运算呢? 如前所述, 二进制数只有两个基本数字 1 和 0, 因此对二进制数的运算实际上可归结为对 1 和 0 的运算。1 和 0 可以同逻辑中的是和非相对应, 所以对 1 和 0 的运算就是对是和非的运算, 这种运算实际上就是一种逻辑的判断。电子计算机中, 就是用一些能进行逻辑判断的电路来实现对二进制数运算的。能够对 1 和 0 进行基本逻辑运算的电路, 称为基本逻辑电路。常用的基本逻辑电路有: 或门逻辑电路、与门逻辑电路、非门逻辑电路和双稳态触发电路等。下面我们对这些电路略加介绍。

1. 或门逻辑电路

我们先从日常生活中所熟悉的电灯线路开始讨论。在图 1-6 所示的电灯线路中, 开关 K_1 和 K_2 并联后再与电源及电灯串联。我们对电灯的亮或暗这两种现象作如下规定, 即电灯亮为 1 状态, 暗为 0 状态。同时对开关 K_1 和 K_2 所处的状态也作相应的规定, 即合上是状态 1, 断开则为状态 0。那么, 开关 K_1 和 K_2 的不同状态的组合将决定电灯具有不同的状态, 如表 1-2 所示。

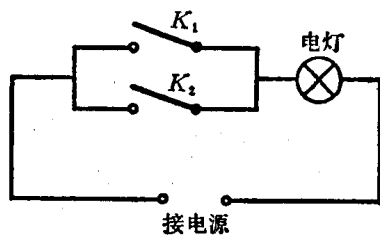


图1-6 说明“或”现象的电灯线路

表 1-2

开关状态		电灯状态	说明
K_1	K_2		
0	0	0	K_1 、 K_2 都断开, 电灯是暗的
0	1	1	K_1 断开, K_2 合上, 电灯是亮的
1	0	1	K_1 合上, K_2 断开, 电灯是亮的
1	1	1	K_1 合上, K_2 合上, 电灯是亮的

如果把开关状态称为输入, 电灯状态称为输出, 那么, 表 1-2 所列的四种情况所代表的输入和输出之间的关系可用下列等式表示:

$$0 + 0 = 0, \quad 0 + 1 = 1, \quad 1 + 0 = 1, \quad 1 + 1 = 1。$$

以上等式就是逻辑加法的关系式。与二进制加法规则是很相近的, 除二进制中 $1 + 1 = 10$ 与逻辑加法中的 $1 + 1 = 1$ 不同外, 其余三式完全相同。

在电子数字计算机中, 能够实现逻辑加法的电路, 称为或门逻辑电路, 简称或门。

最简单的一种或门电路, 如图 1-7 (a) 所示。其中 D_1 和 D_2 为晶体二极管, A 、 B 为输入端, Y 为输出端, R 为电阻。规定: 输入端 A 或 B 为高电位时, 代表 1 状态, 低电位则代表 0 状态。同样, 输出端 Y 为高电位时, 代表输出为 1 状态, 低电位则代表输出为 0 状态。当输入端 A 和 B 中有一个为高电位时, 电流通过二极管经过电阻 R 流向地, 在电阻 R 上产生电压降, 使输出端 Y 为高电位; 自然, 当 A 和 B 都是高电位时, Y 也是高电位。

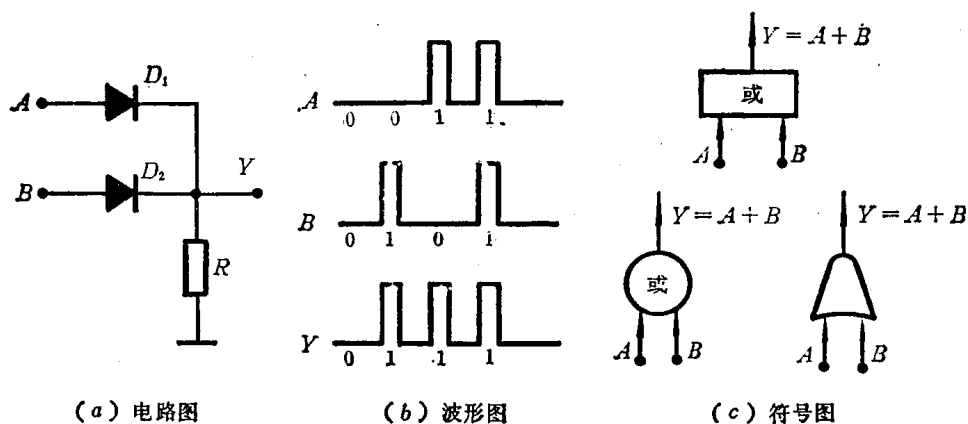


图1-7 或门逻辑电路

当 A 和 B 都是低电位时，电流被 D_1 和 D_2 所阻隔，没有电流流通，这时输出端 Y 为低电位。输出和输入的电位关系用矩形波形图表示，如图 1-7 (b) 所示。另外，输入端的数目不一定只有两个，也可以有几个，一般最多为四到五个。输出与输入的关系，可写为一般的表达式：

$$Y = A + B + C + \dots$$

在电子数字计算机的逻辑图上，或门逻辑电路常用图 1-7 (c) 所示的符号图来表示，在矩形框或圆圈中，有时也写汉语拼音字母 H (即 HUO 字第一个字母)，或者写英文字母 OR。

2. 与门逻辑电路

图 1-8 所示的串联开关电灯线路，可以形象地帮助我们理解什么是与门逻辑电路。由图可见，只有当开关 K_1 和 K_2 同时合上，电灯才亮，否则总是暗的。引用讲述或门逻辑电路的有关规定，则开关状态与电灯状态的关系如表 1-3 所示。

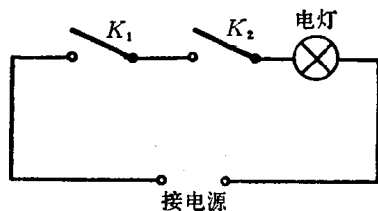


图1-8 说明“与”现象的电灯线路

表 1-3

开关状态		电灯状态	说 明
K_1	K_2		
0	0	0	K_1 、 K_2 都断开，电灯是暗的
0	1	0	K_1 断开， K_2 合上，电灯是暗的
1	0	0	K_1 合上， K_2 断开，电灯是暗的
1	1	1	K_1 、 K_2 都合上，电灯是亮的

与门逻辑电路输入和输出的关系，可用下列等式表示：

$$0 \times 0 = 0, 0 \times 1 = 0, 1 \times 0 = 0, 1 \times 1 = 1。$$

这就是逻辑乘法，与二进制乘法规则完全相同。

在电子数字计算机中，能够实现逻辑乘法的电路称为与门逻辑电路，简称与门。最简单的与门电路如图 1-9 (a) 所示。当输入端 A 、 B 中只要有一个是低电位，电流便从 $+E$

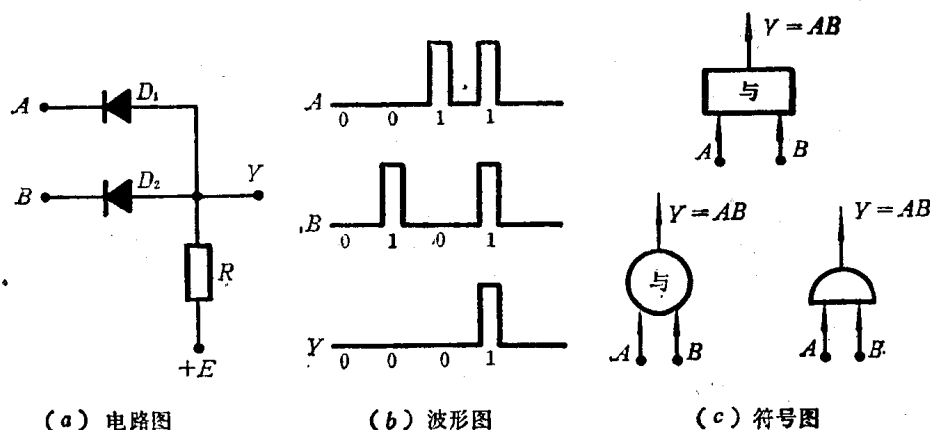


图1-9 与门逻辑电路

端通过电阻 R 并流经二极管 D_1 或 D_2 到低电位输入端，由于二极管的正向电阻很小，故输出端 Y 是低电位。只有当输入端 A 和 B 都是高电位时，二极管 D_1 和 D_2 都不通，输出端 Y 才是高电位。若规定高电位为 1 状态，低电位为 0 状态，则此电路所实现的输入输出关系恰好与逻辑乘法相同。其输入输出关系可用波形图表示，如图 1-9 (b) 所示。

输入端不只限于 A 、 B 两个，还可有 C 、 D ……多个，故其输出与输入之间的关系可写成一般表达式：

$$Y = ABCD \dots$$

与门的符号图如图 1-9 (c) 所示，在矩形框或圆图中，也可写 Y 或 AND 字样。

3. 非门逻辑电路

此种电路的原理可以用图 1-10 所示的电灯线路加以说明。当开关 K 断开时，电灯是亮的；当开关 K 合上时，电流从开关流过，电灯是暗的。引用上述关于开关状态与电灯状态的规定，可以列出其关系如表 1-4 所示。

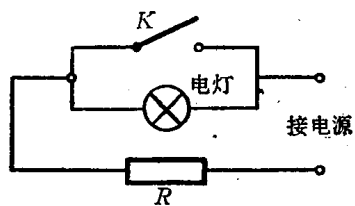


图1-10 说明“非”现象的电灯线路

表 1-4

开关状态	电灯状态	说 明
0	1	K 断开，电灯是亮的
1	0	K 合上，电灯是暗的

开关 K 对电灯起了否定的作用，称为非。对于一般情况，假定输入状态是 A ，则输出状态是 $\bar{A}$ 。 $\bar{A}$ 读做非 A 或 A 非。

在电子数字计算机中能够实现“非”这一逻辑关系的电路称为非门逻辑电路，简称非门。通常又称为反相器，因为它的输出总是输入的反相。

图 1-11 (a) 为反相器的一种典型电路图。当输入端 A 是高电位时（即输入为 1），晶体管 G 导通，电流由 $+E_c$ 流经电阻 R_c 和三极管 G 到地，由于晶体管 G 导通时的内阻很小，输出端 $\bar{A}$ 为低电位（即输出为 0）。当 A 是低电位时，晶体管截止，这时二极管 D 是导通的，输出端 $\bar{A}$ 为高电位。在这里，晶体管 G 起着开关的作用，当输入为 1 时，开关

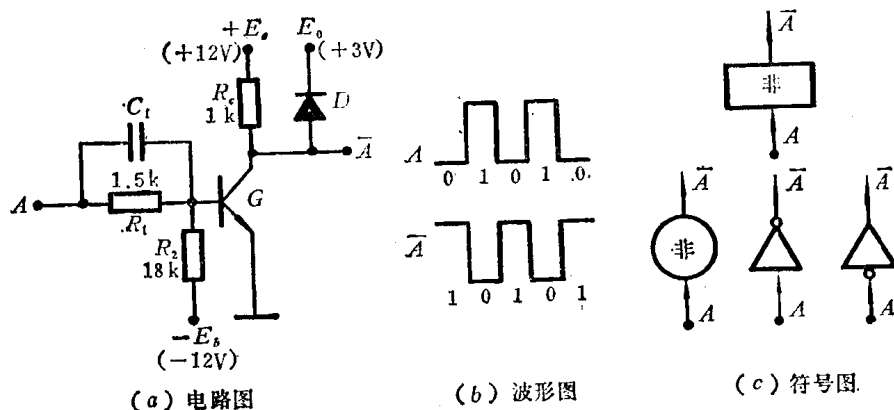


图1-11 非门逻辑电路

接通，输出为0；当输入为0时，开关断开，输出为1。实现了输出对输入的反相。反相器的波形图如图1-11(b)所示。在逻辑图上，非门的符号图如图1-11(c)所示，在矩形框或圆圈中，也可写FEI或NOT。

除以上三种基本逻辑电路外，还有一些基本电路。比如，双稳态触发器，它的作用是暂时和长期保存信息1、0，这里就不讨论了。

有了以上一点基本电路常识，或许可以进一步阅读有关电子计算机基本部件的组成和工作原理的资料。电子数字计算机的基本部件有数码寄存器、移位寄存器、计数器、译码器和全加器等。

二、电子数字计算机的有关术语

在编制算题的计算程序和联系计算工作时，常常遇到有关计算机的一些术语，例如，机器的容量、存储单元、字节、单字等等，如果对他们一点也不了解，那么工作就很困难。为此我们在这里简略地加以介绍。

在§1-1中讲到，电子数字计算机主要是由运算器、存储器、控制器等部分组成的。计算机的功能，主要取决于这些组成部分的性能。

1. 字长和运算速度

运算器的主要技术指标是字长和运算速度。字长是由运算器中寄存器的触发器个数所决定的，因为每一个触发器可以保存一位二进制数。例如，寄存器是由48个触发器所组成的，则称机器的字长是48位。字长越长，有效数字的位数越多，计算精度就越高。一般计算机的字长为20~50位。

运算速度是指机器进行四则运算的快慢程度。通常用做加法或乘法的时间来衡量电子计算机的运算速度。例如，153型机做加法的时间是6~9微秒，做乘法的时间是29~32微秒。另外还有一种通俗的说法，即每秒完成的算术运算的平均次数，例如，153型计算机的定点运算速度为每秒10万次。虽然平均速度不能确切地反映计算机的运算速度，但它大体上给出运算快慢的概念。

2. 定点和浮点运算

上面提到定点运算这一概念，与它对应的是浮点运算。这是为解决数字运算时小数点定位问题而提出的两种办法。计算机在对定点数进行运算时，要求参加运算的数的绝对值

都小于1，把小数点固定在数码最前面。对于大于1的数，要选择一个适当的比例因子，用此比例因子去除所有数据，使它们都不超过1（即化为二进制的真分数）去参加运算。输出结果时，当然应乘以同一比例因子。

浮点运算则是小数点位置不固定，即浮动的。例如3.1416这个数，可写成 3.1416×10^0 、 0.31416×10^1 、 0.031416×10^2 、 31.416×10^{-1} 等多种形式。如果小数点后第一位为有效数字的最高位，则这个浮点数称为规格化的浮点数，例如上面所写的 0.31416×10^1 就是一个规格化的浮点数。概括起来说就是：在十进制中，一个数 N 可以表示为

$$N = 10^j \times M$$

式中 M 称为数码，也称尾数部分或尾数；

j 称为数 N 的阶码，它是一个任意整数；

10是阶码的底。

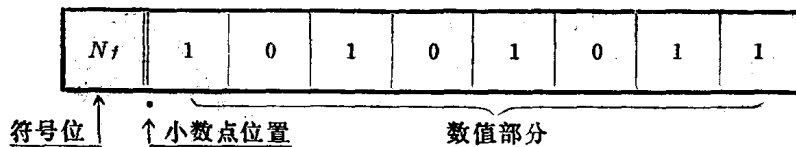
类似地，在二进制中，一个数 N 可以表示为

$$N = 2^j \times M$$

其中 j 、 M 的意义与十进制表达式中的 j 、 M 相同，但他们是二进制数，阶码的底是2。

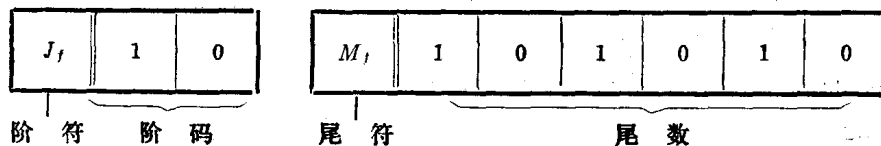
在设计计算机时，通常将尾数定为小于1的数，在定点形式中，将阶码 j 固定为0，而在浮点形式中， j 是可以变化的。为了说明定点运算和浮点运算的区别，我们用框图表示计算机中存放数的单元，每一矩形框代表一位，它有两种状态，分别代表二进制的0和1。

例如有一个定点数 $N = 0.10101011$ ，在机器中的形式可以安排如下：

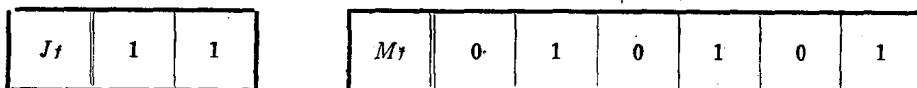


这时存储单元的八位全部用来表示有效数字，小数点永远固定在小数最高位之前。

若是浮点数，比如 $N = 2^{10} \times 0.101010$ ，假定分配两位表示阶码，其余六位表示尾数，则在机器中的形式如下：



在进行浮点运算时，阶码和尾数分别运算。例如两个数 $N_1 = 2^{j_1} \times M_1$ 和 $N_2 = 2^{j_2} \times M_2$ 相加，若 $j_1 = j_2$ ，则其结果为 $N = 2^{j_1} (M_1 + M_2)$ ；若 $j_1 \neq j_2$ ，则首先需将小数点移位，使其阶码相同，然后才能相加。小数点往左移一位，在机器中表现为尾数数码向右移一位而阶码加1。比如将 $N = 2^{10} \times 0.101010$ 的小数点向左移一位，其在机器中的形式变为：



即 $N = 2^{11} \times 0.010101$ 。

从以上例子可以看出，对一定的运算器来说，采用定点运算，其有效数字的位数长，

但数的表示范围小；而采用浮点运算，其有效数字位数较短，但数的表示范围却较大。仍以上述例子说明如下：若只考虑绝对值，那么，对定点运算来说，它的有效数字可达八位，其能表示的数的范围为 $0.00000001 \sim 0.11111111$ ，相当于十进制数 $2^{-8} \sim (1-2^{-8})$ ，即 $\frac{1}{256} \sim \frac{255}{256}$ ；对浮点运算来说，由于阶码占用了两位，尾数的有效数字就少了两位，其能表示的数的范围则为 $2^{-11} \times 0.000001 \sim 2^{11} \times 0.111111$ ，相当于十进制数 $2^{-3} \times 2^{-6} \sim 2^3 \times (1-2^{-6})$ ，即 $\frac{1}{512} \sim 7\frac{7}{8}$ 。当存储器的字长一定时，分配给阶码的位数越多，则能表示的数的范围就越大，但有效数字位数就要相应地减少，因此，阶码与尾数各分配多少位，需视具体情况而定。

另外，从计算机的控制来说，定点运算比浮点运算较为简单，故其运算速度较快。例如，上面提到的153型计算机，其定点运算速度为每秒10万次，而浮点运算速度则为每秒几千次。又如，108-乙型计算机的定点运算速度为每秒10万次，而其浮点运算速度则为每秒2万次。

现代计算机一般都可采用定点和浮点两种运算，对于不同类型的数，选用不同的运算形式。

3. 存储单元和容量

存储器是电子计算机的重要组成部分之一，它的容量大小决定了所能够解算题目的规模，因此要对它加以注意。

存储器的核心部分是磁心体，它是由许多磁心按照一定的规律排列而成的，通常构成一个立体矩阵，如图1-12所示。该图中的磁心体是由32块磁心板组成的，每块磁心板上装有许多个（例如4096个）磁心。每一个磁心能记存一位数1或0，每一串磁心可存放一个二进制数据，这一串磁心好比一个房间，在计算机中称为一个存储单元，它有一个编号（一般按八进制或十六进制），称为存储地址或简称地址。一个单元中磁心的个数，决定能够存几位数据，也就是前面所说的字长。例如109-乙机每单元磁心个数为32个，故其字长为32位（二进制）。每块磁心板上的磁心个数，表明磁心体所能存储数据的多少（图1-12中仅示出了16个）。存储器一般由若干个磁心体组成。

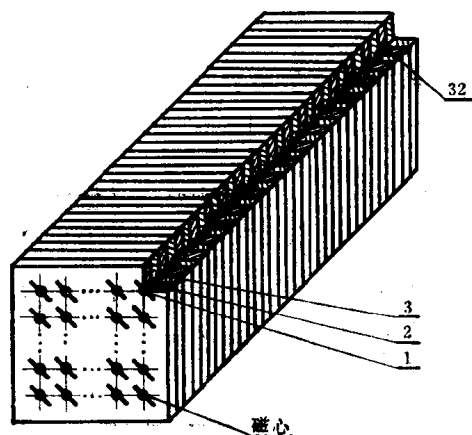


图1-12 磁心体构成示意图

存储器的主要技术指标是存储容量和存取周期。存储容量取决于存储单元的数量。例如108-乙型计算机的磁心存储器的容量是16384个单元（每单元字长为48位）。存储容量越大，计算机解题能力就越强。存取周期或称存取时间，就是存一次或取一次所需的时间。例如，108-乙型计算机的磁心存储器的存取周期是4~5微秒。又如，FELIX C-256计算机的中央存储器的存取周期为：每存或取一个“半字”的时间为950毫微秒，即 $\approx 10^{-6}$ 秒。

4. 字节、单字和半字

有的计算机的存储器容量以字节为单位进行计算。所谓字节，就是由计算机加工的最小信息单位，它是由若干个二进制位所组成的。例如，IBM-360系统，一个字节由九个