

数学通报丛书

关于电子数字
计算机的一些问题

中国数学会数学通报编委会编

科学技术出版社

关于电子数字计算机的 一些問題

中国数学会数学通报編委会編

科学技术出版社

1980年·北京

关于电子数字计算机的一些問題

中国数学会数学通报編委会編

*

科学技术出版社出版

(北京市西直門外郝家灣)

北京市書刊出版業營業許可証出字第091號

北京市印刷一厂印刷

新华書店科技發行所發行 各地新华書店經售

*

开本: $850 \times 1168 \frac{1}{32}$ 印張: $3\frac{1}{2}$ 字数: 86,000

1960年4月第1版 1960年4月第1次印刷

印数: 6,590

总号: 1477 統一書号: 13051·302

定价: (9) 4角1分

出版者的話

数学通报自創刊以来，發表了不少对讀者进修数学、扩大数学知識領域有帮助的文章，特別是發表了不少有关中等学校数学教学的文章，受到了讀者的欢迎，并經常收到讀者来信要求将这些文章分类編成單行本出版。数学通报編委会为了满足讀者这一需要，特將該刊自創刊号起至 1959 年中的文章，选其質量較好的并按性質分成数学知識介紹和中等学校教学两部分分类出書。前一部分已选編成“綫性代数多項式”、“实数、極限、近似計算”、“几何作圖、非欧几何、邏輯初步”、“概率和数理統計”、“关于电子数字计算机的一些問題”、“初等数学史”、“趣味的数学問題”和“中等数学習題解答集”；后一部分已选編成“中学数学教学的一般問題”、“初中数学教材和教法分析”和“高中数学教材和教法分析”。

这十一本書的內容，有适合初等数学的，或适合在初等数学基础上进一步提高的，因此，具有相当高中数学水平的讀者、中等学校数学教师、大、中学生及数学爱好者均可閱讀。

“关于电子数字计算机的一些問題”是这套丛書中的一本。

目 次

1. 程序設計.....Д.Ю. 潘諾夫 (1)
2. 在电子计算机上解题的基本知識... А.А.李雅普諾夫 (13)
Г.А.塞斯特巴尔
3. 控制过程的算法描述..... А.А.李雅普諾夫 (33)
Г.А.塞斯特巴尔
4. 模型计算机的簡單介紹..... 莫紹揆 (49)
5. 计算机及其在铁路运输中的应用... В.Д.莫依謝列夫 (72)
Н.С.尼可拉也夫
6. 自动司机..... В.И.羅斯庫托夫 (87)
7. 电子计算机在国民經济计划和統計
工作中的应用..... 伏·別尔金 (93)

程序設計^①

А.Ю. 潘諾夫

最近十年来，在計算技术方面起了很大的变革，这就是数字电子计算机的出現。它工作的速度快到我們不可能用手来操縱它，而要由机器本身进行自动的操縱。

数字电子计算机的重要組成部分是輸入、存儲、运算、控制、輸出(圖1)。

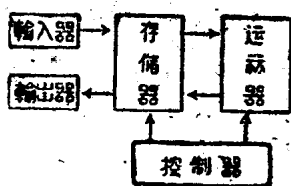


圖 1

数字由輸入器送入存儲器，再到运算器去运算。运算結果暂时送入存儲器，或由輸出器送出。所有这些运算都是由控制器掌握，最后將結果輸出。用紙上打孔方法記錄下来。

存儲器可以有不同的形式。現在常用的是阴極射綫管、磁轉子或磁帶。要求存儲量多時就用磁帶。



圖 2

数字是用代碼存儲在存儲器內，放代碼的地方叫單元，它們都編有號碼，叫做地址。机器里按地址寻找存位，就好像我們在旅館里根据號碼找房間一样(圖2表示磁轉子存儲数字代碼的號碼大致情况)。

运算器和手搖的計算机原則是类似的，不同的是这里不

① 此文是根据潘諾夫教授1956年在我国的报告記錄整理而成。虽經报告人同意發表，但未經他詳細审查，所以有关本文的一切問題由数学通报編委会負責。由于程序設計理論發展很快，这个报告說的是当时情况，有些看法和現在的情况已不符合了，但所介紹的基本理論，至今仍然是正确的。——編者

是用齒輪，而是用電子計算綫路。蘇聯科學院 B3CM 計算機進行一次加法運算，只需 70 微秒。運算進行必須由控制器控制。為了使控制器能正確控制，就必須有程序，程序就是保證進行一連串運算的一連串指令。今天簡單而且簡短的講程序的基本要點和怎樣編制程序。

每一個程序是由許多指令組成。指令必須指明：一、進行哪種運算；二、數字從哪里去取；三、運算完畢的結果要送到哪里去。如果指令不是相連時，也必須指出，下一道指令進行哪一個運算。詳細的下面再講。

電子計算機中數字以二進位數表示，例如：

1	2	3	4	5	6	7	8	9	10
1	10	11	100	101	110	111	1000	1001	1010
	2^1		2^2				2^3		

十進位數 $10^2=100$, $10^3=1000$, ...

二進位數 $2^2=100$, $2^3=1000$, ...

採取二進位數時，數字雖然寫長了一些，但運算非常簡單。譬如說，採取二進位數，九九乘法表就用不着了，只有 1 和 1 相乘，或 0 和 1 相乘。

因為大家都熟悉二進位數，所以我就不多介紹了。

在計算機內，每一個數的形式如下（下面是取 B3CM 的實例，每一個數用 39 個碼位）：

1	P	1	A
---	---	---	---

$$N = A \cdot 2^P$$

6 位（表示階） 33 位（也叫真數）

P 和 A 前空一格，表示運算符號。

1 ~ (+), 0 ~ (-).

這樣 N 的範圍是 $\pm 2^{+11111} \cdot 111 \dots 1$ 或 $10^{-9} \rightarrow 10^9$ ，這對

於一般數學、物理方面的問題已經夠用了。

機器的指令形式如下：

+	A_1	A_2	A_3
运算碼	第一地址	第二地址	第三地址
6 位	11 位	11 位	11 位

这是 B3CM 所用的三地址指令，指令本身亦用二进位数表示。其中第一部分表示应该执行的运算(用它的代号来表示)。第一地址及第二地址表示应该从哪两个存位来取数进行运算，第三地址表示运算的结果应该送到哪里。

例如：000100 00000000101 00000001001 00000001010

6 位 11 位 11 位 11 位

即

:	5	9	10
---	---	---	----

这表示第 5 单元里的数被第 9 单元里的数除，将结果送入第 10 单元内。即

(5):(9)送入第 10 单元

[单元 α 内的数用符号(α)表示]。

必须注意，地址里的数可以很大，只要是在机器的可能运算范围之内就行。如

(5) = 1230.09632.

(9) = -2345.69973.

(10) = -0.5... 机器可以计算得很准确。

程序是进行运算的指令的总体。解决某一个数学问题，每一步算术运算有它自己的指令通常是不可能的，因为如数学、物理以及技术上的问题，有时要用到 10^9 个算术运算，而把这些运算记录下来所花的时间比用人工计算这个问题所花的时间还要多。所以要求解决任何一个问题，总要求指令的总数不要太多。编制程序时就是要尽量使之简单。有时是能做到这一点的，如外弹道问题，共有约 10000 个运算，我们用 99 个指令所组成的程序就指导了运算。

必须说明，编制程序的一般方法现在还没有，现在程序设计的情况正像中世纪时正在成长中的代数学的原始状态一样。

那时是各人有单独寻求解决个别算术问题的方法，甚至相互間是保密的，老师只傳給他的学生。解决了某个問題的人不一定能解决另一問題，因为当时一般的理論还未建成，沒有解决問題的一般方法。目前程序設計正是这样^①。現在我們已开始利用已知的程序，进行建立程序的一般理論的研究。今天向大家介紹一些簡單的关于程序設計中的一些情况。

表1. 计算机的运算

1. 加~田

+	α	β	γ	$(\alpha) + (\beta)$ 送入 (γ) .
---	----------	---------	----------	--------------------------------------

其意是 α 存位里的数加 β 存位里的数，然后放在 γ 单元內。

2. 减~日

-	α	β	γ	$(\alpha) - (\beta)$ 送入 (γ) .
---	----------	---------	----------	--------------------------------------

3. 乘~田

$\times$	α	β	γ	$(\alpha) \times (\beta)$ 送入 (γ) .
----------	----------	---------	----------	---

4. 除~日

:	α	β	γ	$(\alpha) : (\beta)$ 送入 (γ) .
---	----------	---------	----------	--------------------------------------

5. 轉送

+	α		β	$(\alpha) + 0 = (\alpha)$ 送入 (β) .
---	----------	--	---------	--

(0) = 0，空格表示 0。

或

+		α	β
---	--	----------	---------

6. 有負号的轉送

-		α	β	$-(\alpha)$ 送入 (β)
---	--	----------	---------	--------------------------

7. 清除单元內数字

+			α	将 α 存位內的数洗掉。
---	--	--	----------	---------------------

① 作者說的是当时情况，目前已有所不同。——譯者

8. 使机器停止 0 是使机器停止的符号。若运算碼內未写 0, 則格內是空的, 但机器还是在工作。

0			
---	--	--	--

9. 印出結果

π	α		
-------	----------	--	--

將 α 單元內的數輸出印刷。

10. 条件轉移 也叫比較运算。

$\geq$	α	β	γ
--------	----------	---------	----------

有两种可能性:

① 若 $(\alpha) < (\beta)$, 則轉移到实现按編号的下一指令。

② 若 $(\alpha) \geq (\beta)$, 則轉移到实现指令 (γ) 。

11. 无条件轉移

$>$			α
-----	--	--	----------

轉移到实现指令 (α) 。

表2. 存儲器內地址的符号

I. 存放指令的地址表为 $K+1, K+2, K+3, \dots$
(为方便計, 今后把指令本身的編号和与存放这一指令的單元編号等同起来。)

II. 存放計算过程中所必需的一定的量的地址表为 $A, B, C, D, \dots$

III. 供暂时存儲任意的量的地址表为 $U, V, W, Z, \dots$
(操作單元)。

表3. 程序記錄示意表

I. 所需各量在存儲器中存放情况

存儲器內的單元 $\left\{ \begin{array}{c} A \\ B \\ C \\ D \end{array} \left[\begin{array}{c} a \\ b \\ c \\ d \end{array} \right] \right\}$ 其中所存放的量

I. 指令一覽表

指令編號	存放指令的地址	指 令			
		運算碼	第一地址	第二地址	第三地址
1	$K+1$	$\times$	A	B	U
2	$K+2$	$+$	C	U	V
3	$K+3$	$\times$	F	A	V

$$(V) = (ab+c)a.$$

程 序 № 1

复数的除法

$$z_1 = a + bi, \quad z_2 = c + di,$$

$$\frac{z_1}{z_2} = \frac{ac + bd}{c^2 + d^2} + \frac{bc - ad}{c^2 + d^2}i.$$

I. 存儲器單元中的存放情況:

A	a	U	Var
B	b	V	Var
C	c	W	Var
D	d	Z	Var

操作單元.

II. 指令一覽表

指令的地址	運算碼	第一地址	第二地址	第三地址	結果(在第三地址所示的單元內)
$K+1$	$\times$	C	e	U	c^2
$K+2$	$\times$	D	D	V	d^2
$K+3$	$+$	U	V	U	$c^2 + d^2$
$K+4$	$\times$	A	C	V	ac
$K+5$	$\times$	B	D	W	bd
$K+6$	$+$	V	W	V	$ac + bd$
$K+7$	$\times$	B	C	W	bc
$K+8$	$\times$	A	D	Z	ad
$K+9$	$-$	W	Z	W	$bc - ad$
$K+10$	$:$	V	U	V	$\frac{ac + bd}{c^2 + d^2}$
$K+11$	$:$	W	U	W	$\frac{bc - ad}{c^2 + d^2}$
$K+12$	0				停止

这个例子显示程序设计中的一个要点：單元尽量少用，不必存儲的数字立刻去掉，以节省存儲器的存儲量的使用。

程 序 № 2

e^x 值的近似計算

$$e^x = 1 + x + \frac{x^2}{1 \cdot 2} + \frac{x^3}{1 \cdot 2 \cdot 3} + \dots + \frac{x^n}{1 \cdot 2 \cdot \dots \cdot n}$$

S_n

n 为我們要求計算的項數。

第一方案：I. 存放情况

A	x
B	1
C	1
D	1
E	x

II. 指令一覽表

K+1	+	A	C	C	$1+x$
K+2	+	B	D	D	2
K+3	$\times$	A	E	A	x^2
K+4	:	A	D	A	$\frac{x^2}{2}$
K+5	+	A	C	C	$1+x+\frac{x^2}{2}$
K+6	+	B	D	D	3
K+7	$\times$	A	E	A	$\frac{x^3}{2}$
K+8	:	A	D	A	$\frac{x^3}{2 \cdot 3}$
K+9	+	A	C	C	$1+x+\frac{x^2}{1 \cdot 2}+\frac{x^3}{1 \cdot 2 \cdot 3}$
K+10	+	B	D	D	4

但我們立刻看出，方格中的符号完全重复，因此我們考虑程序編制可用循环重复方法。

第二方案：

I. 存放情况

A	1
B	1
C	1
D	0
E	α

II. 指令一覽表

K+1	+	B	D	D
K+2	$\times$	A	E	A
K+3	:	A	D	A
K+4	+	A	C	C
K+5	$\geq$			K+1

这里利用了无条件轉移运算，使得机器自动地进行循环的演算。但这样又有一个缺点，机器永远不会停止工作。

第三方案：設法使机器自动进行循环运算，但达到要求后即自行停止工作。

I. 存放情况

A	1
B	1
C	1
D	0
E	α
F	$n-1$

II. 指令一覽表

K+1	+	B	D	D
K+2	$\times$	A	E	A
K+3	:	A	D	A
K+4	+	A	C	C
K+5	$\geq$	F	D	K+1
K+6	0			

这里，在执行了指令 $K+4$ 后， D 单元內的数是 1。指令 $K+5$ 系将 D 单元內的数（此时为 1）与 F 单元的数即固定的 $n-1$ 相比較。最初自然 $n-1 \geq 1$ 。故按照条件轉移的約定又轉回到指令 $K+1$ ，即形式地重复上一循环（但此时被运算的数已經不同了）。经过 n 次重复后， D 单元的数是 $n (> n-1)$ ，于是按照条件轉移的約定，即轉至次一指令 $K+6$ ，即停止工作。

这个例子又显示程序設計中的两个要点：一、尽量搜索运算中的循环过程，利用循环重复法編制程序。这样可以大大地

減縮指令的总数；二、計算过程中所須要的某些系数（如 2, 3, ..., p, p! 等）可由机器本身按照程序自动算出，而不必事先列出存入。

程 序 № 3

用欧拉法解微分方程 $y' = ay$ ，初始条件 $x = x_0, y = y_0$ ；
算至 $x < x_N$ 。

$$\Delta y_n = h \cdot ay_n, \quad h = \Delta x,$$

$$y_{n+1} = y_n + \Delta y_n, \quad x_{n+1} = x_n + h.$$

I. 存放情况

A	ah
B	h
C	x_N
D	x
E	y
Z	Var

II. 指令一覽表

K+1	×	A	E	Z
K+2	+	E	Z	E
K+3	+	D	B	D
K+4	Π	D		
K+5	Π	E		
K+6	≥	C	D	K+1
K+7	0			

这个例子是解微分方程，积分曲线必須描画出来，因此一些中間結果必須輸出印刷。如果不需要印刷的，最好不印刷。因为輸出印刷最費時間，运算过程加上印刷所費的時間，有时为仅运算而不加印刷所費時間的 1,000 倍。

程 序 № 4

六次多項式的計算：

$$p(x) = a_0x^6 + a_1x^5 + a_2x^4 + a_3x^3 + a_4x^2 + a_5x + a_6$$

$$= (((((a_0x + a_1)x + a_2)x + a_3)x + a_4)x + a_5)x + a_6$$

第一方案:

I. 存放情况

A	a_0
A+1	a_1
A+2	a_2
A+3	a_3
A+4	a_4
A+5	a_5
A+6	a_6
B	x
Z	Var

$K+1$
 $K+2$
 $K+3$
 $K+4$
 $K+5$
 $K+6$
 $K+7$
 $K+8$
 $\vdots$
 $K+12$
 $K+13$

II. 指令一覽表

$\times$	A	B	Z
+	Z	A+1	Z
$\times$	Z	B	Z
+	Z	A+2	Z
$\times$	Z	B	Z
+	Z	A+3	Z
$\vdots$	$\vdots$	$\vdots$	$\vdots$
+	Z	A+6	Z
0			

我們可以看出运算过程是加法和乘法相間进行，作加法时第一地址总是 Z，第二地址每次加 1，所以我們設法使第二地址的代碼在每进行一次加法运算时自动加 1。

第二方案，引进两道新的指令如下：

I. 存放情况

A	a_0
A+1	a_1
A+2	a_2
A+3	a_3
A+4	a_4
A+5	a_5
A+6	a_6
B	x
Z	Var

II. 指令一覽表

$K+1$	+	A		Z
$K+2$	$\times$	Z	B	Z
$K+3$	+	Z	A+1	Z
$K+4$	+	K+3	M	K+3
$K+5$	$\geq$	N	K+3	K+3
$K+6$	0			

A			1
B	+	Z	A+7
Z			Z

最初当执行了指令 $K+3$ 后，單元 Z 內的数为 a_0x+a_1 。
指令 $K+4$ 系將單元 $K+3$ 內的数即指令 $K+3$ 与單元 M 內的数即 ϵ 相加而將結果送至單元 $K+3$ 。即

+	Z	A+1	Z
---	---	-----	---

加		1	
---	--	---	--

得	+	Z	A+2	Z
---	---	---	-----	---

，被送至單元 $K+3$ 。
故此时 $K+3$ 指令中的第二地址中的数已比前同一指令的相应地址中的数大 1。指令 $K+4$ 系將 N 單元中的固定数 δ 与此时單元 $K+3$ 中的数相比較。因 $A+6 \geq A+2$ ，故按条件轉移的规定而轉至指令 $K+2$ ，即重复原有的循环，但在第二个循环里，指令 $K+3$ 系指定在 $A+2$ 單元內取数，即取 a_2 而不是以前的 a_1 ，并且將新的結果 $(a_0x+a_1)x+a_2$ 送至單元 Z 。这样每經一次循环，指令 $K+3$ 的第二地址自动加 1。經過六个循环后，指令 $K+3$ 的第二地址为 $A+7$ ，它大于 δ 的第二地址 $A+6$ 。于是經過比較指令 $K+5$ 后机器即行停止。此时單元 Z 內所存的结果就是所要的六次多項式的值 $a_0x^6+\dots+a_6$ 。

这就是將指令也加以运算的例子。这种方法以及循环运算方法，对程序的縮减非常有用。这些例子說明程序是可以縮减的，另一种利用現有的标准程序(如复数的除法等)，插入正在求的运算程序中，也是使編制程序時間縮短的一种方法，这叫做子程序的利用。

如上面曾提到的外彈道問題，約 10,000 个运算用了 99 个指令，312 个單元。

程序設計在数字电子計算机的使用中，是不可少的环节，而对于專用数字电子計算机而言它还有进一步的意义。

設計專用机时，就要求从尽可能設計出的程序中選擇最簡

單的程序，再確定機器的構造，因此在專用機的設計中，數學家起決定性的作用。程序的簡單複雜就直接決定機器成本的高低。但不是所有的問題都可造出簡單的程序，如文字的自動翻譯，就要 10,000 以上的指令，因此產生了要求程序也自動化的問題，這方面的工作還不多，但已經做了一些。

目前程序設計工作的發展中存在着兩個方向。1. 一般理論的建立以及編制標準程序，其意義就像形成代數學後能統一地解決各種算術問題一樣。2. 作程序的程序，即程序設計自動化的問題。現在還不能說這樣做出的結果最好，但可以得出這樣的信念，自動編制的程序沒有錯誤，而程序中有錯誤是最糟的。現在人工編制程序錯誤總難避免，每個程序做出後，都不得在機器上試驗、修改。試機往往要花 1—2 小時，和真正計算的時間相當。用以上的標準程序和程序的方法，就可消滅編制程序中的錯誤。現在程序還不能由非編制此程序的人來檢查，因這樣所花時間會超過原來編制程序的時間，同時由於沒有一般的理論，兩個人在一起工作也是困難的^①。

最後，準備對將來參加程序設計工作的數學家說幾句話。要從事程序設計工作當然首先需要掌握數學。要有高度的敏感和機警，就是說在沒有辦法的情況下，能想出辦法、找竅門、找出路。要能堅持、能耐心地听取計算機工程師的意見。但不管怎樣，這是很重要而且也很有趣的工作，程序設計師編成一個好程序所得到的滿足正好像藝術家完成一件藝術創作時所得到的滿足一樣。

我相信程序設計的工作在中國將得到迅速的發展。

(馮康 黃啓晉 羅佩琮 譯)

^① 由於程序設計理論發展很快，這種情況已經有了改變。——譯者