

JS-10A 小型工控机 程序

上海纺织工学院

翻印上海科技

一九七七·六

目 录

≈≈≈≈≈≈≈

第一章：程序和程序设计

第一节 引 言

第二节 算术公式，控制过程的直接程序设计

第三节 单循环的程序设计

第四节 多重循环的程序设计

第五节 指令的修改和使用

第六节 分支程序

第七节 子程序

第二章 程序介绍

第一节 表演程序

第二节 验机程序

第三节 外围分调程序

第三章 JS-10A机应用介绍

第一节 60台手套机群控程序及应用

第二节 标准钢样子数计算（新跃仪表厂）

第三节 硅单晶拉制温度自动控制（上海电子器材厂）

附录： 指令表(JS-10A)

包括：传送指令、输入、输出指令、算术逻辑指令、转移指令、移位指令等共有三十七条(其中二十二条为微指令)。

操作码	名称	操 作 内 容	A	C	备 注
00	按位加	$(A) \oplus (D) \rightarrow A$	变	不变	
01	加 法	$(A) + (D) \rightarrow AA$	变	有进位 变反	结果 $\geq 2^{18}$ 有进位
02	减 法	$(A) - (D) \rightarrow A$ (实际为 $(A) + (D) + 1 \rightarrow A$)	变	有进位 变反	$(A) \geq (D)$ 有进位
03	取 数	$(D) \rightarrow A$	变	不变	
04	存 数	$(A) \rightarrow (D)$	不变	不变	
05	D右移	(D) 带C右循环移一位	不变	变	“4”时右移一位
06	全 等	$(A) \neq (D)$ 则 $pc+2$, 否则 $pc+1$	不变	不变	利用 $(A) + (D) = 0$ 则 $A = (D)$
10	比 较	$(A) \geq (D)$ 且 $C=1$ 或 $(A) < (D)$ 且 $C=0$ 则 $pc+2$, 否则 $pc+1$	不变	不变	把 (A) 和 (D) 看成无符 号之数, $(A) \geq (D)$ 则 $J_1 = 1$ $(A) - (D)$ 时 若 $(A) \geq (D)$ 则 $J_1 = 1$ “15”时; 若 $(A) < (D)$ 则 $J_1 = 0$ “15”时
11	加 零 转	$(D) + 1 \rightarrow D$ 若 $(D) = 0$ 则 $pc+2$, 否则 $pc+1$	不变	$(D) = 0$ 时变反	
12	A右移	A循环右移, 右移位数为 (D) 中1之个数	变	不变	
13	逻辑乘	$(A) \cdot (D) \rightarrow A$	变	不变	
14	带返转	(1) $D \rightarrow pc$ (2) 在A中形成 $D7 \sim pc+1$	变	不变	
15	D左移	(D) 带C左循环移一位	不变	变	
16	输 出	$(D) \rightarrow A$, $(A) \rightarrow I/O$ 总 线, 对外发一接收脉冲 SC_2	变	不变	K周期 $(D) \rightarrow A$, K周期 $(A) \rightarrow I/O$ 总线。 “14~“7” 发输出脉冲 SC_2
17	微指令				

17 微指令表

操作特征 码C 码D	符号	操作内容	操作时间	备 注
170000		A 清 0	K 周期 16 拍	
170001	17D ₀	(A)→A	K 周期 16 拍	
170002	17D ₁	C 置 0	"0"前半部分	D ₀ =0 则 A 清 0, D ₀ =1 则 A 保存
170004	17D ₂	C 置 1	" "	" "
170010	17Y ₁	(A)带 C 右循环移 1 位	"4"后沿	不能和 17D ₀ 复合
170020	17Y ₂	(A)带 C 左循环移 1 位	K 周期 16 拍	" "
170030	17Y ₃	2D→A, ZD 清 0	"0"~"3" ZD→A "4"清 0	" "
170040	17Y ₄	B→A	K 周期 16 拍	" "
170050	17Y ₅	(A) (H)	" "	" "
170060	17Y ₆	(A)+1→A	" "	" "
170070	17Y ₇	ZD=0 则 pc+2 否则 pc+1	" "	D ₀ =0 则 A 清 0 D ₀ =1 则 A 保存
170100	17D ₆	输入总线→A, 对 外发一脉冲 sr ₁	K"0"~"15" 发脉冲"4"~ "7"总线→A 并列输入	K 周期时可以复合其 它指令
170200	17D ₇	H 清 0	K 周期 16 拍	D ₀ =0 则 A 清 0 D ₀ =1 则 A 保存
170300	17D ₇	启动光电机	K 周期前沿	不能复合
170400	17D ₈	停机	K 周期后沿	D ₀ =0 则 A 清 0 D ₀ =1 则 A 保存
171000	17X ₁	无条件跳过一条指 令	K 周期 16 拍	" "
172000	17X ₂	C=0 则 pc+2 否则 pc+1	" "	" "
173000	17X ₃	C=1 则 pc+2 否则 pc+1	" "	" "
174000	17X ₄	A=0 则 pc+2 否则 pc+1	" "	" "
175000	17X ₅	A≠0 则 pc+2 否 则 pc+1	" "	" "
176000	17X ₆	Af=0 则 pc+2 否则 pc+1	" "	" "

17	7000	17 X ₇	$A_F \neq 0$ 则 $pc+2$	K 周期 16 拍	$D_0=0$ 则 A 清 0
			否则 $pc+1$		$D_0=1$ 则 A 保存

第一章 程序和程序设计

第一节 引言

利用电子计算机解任何一个计算问题或对某一工业过程控制时，总要用机器所能完成的一系列指令，把它们描述出来，这种为完成某个计算问题而编写的一系列指令便称为程序，编写程序的工作通常称做程序设计，一般说来，程序设计包括下列三个部分的工作。

1. 确定控制、计算过程进行的顺序，必要时对某些数字公式与控制过程进行整理和变动。

2. 进行内存单元的分配，将解题过程中所需的原始数据，中间结果，最后结果进行适当的分配，特别在内存单元紧张时，更要重视这一工作，严防重用和错叠使用。

3. 编制程序和写出必须的辅助数，例如在程序中某些固定常数要从十进制化成八进制数，並写成指令形式，以便使用。

本章将概要地介绍程序设计的各种基本方法，这是编制程序的基础，同时也是初学者掌握程序设计的入门。

第二节 算术公式，控制过程的直接程序设计

算术公式，控制过程的程序设计是最简单的，而且又是最基本的，在这一节中，我们来说明如何进行算术公式的程序设计。一般说来，算术公式的程序设计，它只要依照给定的算术公式、过程控制的动作安排好运算次序，便可按步地逐步写出相应的指令，编出程序，下面我们看二个例子。

例1 编制下述系数计算程序

$$f(x) = ax^2 + bx + C$$

1. 公式整理：

为了使程序便于编制，我们将 $f(X) = ax^2 + bx + C$ 转换为

$$f(x) = (ax+b) \cdot x + C$$

2. 单元分配:

其中: 0100 a
 0101 b
 0102 c
 0103 x
 0104 f(x)

程序从 0110 开始, λ_1, λ_2 为乘法子程序的工作单元, 並不破坏, K 为乘法子程序的入口。

3. 程序编写:

地址	指	令
0110	03	0100<a>
1	04	λ_1
2	03	0103<x>
3	04	λ_2
4	14	K
5	01	0101
6	04	λ_1
7	14	K
120	01	0102<c>
1	04	0104 f(x)

我们可以从上看, 通过了指令将上述的数学公式的每一个动作实现, 而以上指令组成一段实现 $f(x)$ 的函数式的直接工作程序, 从而用计算机来进行计算代替了人工的计算, 同时, 我们从这个例子可以看出, 如上述公式不加以修改, 指令肯定还要用得更多, 这一点读者可以自己试编一下。人们往往为了程序设计方便而改变计算公式的方法是值得注意的, 它也说明了最直接明显的程序, 並不是最有效的, 稍微考虑一下, 往往可以节省存储单元, 对于加快计算速度也是有好处的。因此常常在编制程序前, 认真推敲一下公式, 是完全有必要的, 当然存放指令的单元号码也不一定放在 0110~0121, 原始数据, 中间结果, 工作单元, 最后结果也不一定要放在 0100~0104 中, 重要

的是要知道程序有多长，数据有多少，因而需要多少单元，以便把内存的单元进行合理的分配。

例2：外围通道程序中对模拟量的某一点输出

1. 过程决定：

对模拟量输出需要二个信息

信息一：为正反转号、状态号、单元号 0-000... 0-0000000
正 状
反 态
号 号

信息二：为十位二进制数据 00... 0000-000-000
S

工作过程：(1)送对应的设备号，启动设备，再送其对应的状态号。

(2)送对应的设备号、启动模拟量输出数据寄存器，输入相应的控制量。

2. 单元分配：

0100	信息一
1	信息二
2	设备号 k_1
3	设备号 k_2

程序从 0110 里开始

3. 程序编写：

0110	03	0102	} 送设备号
1	17	0050	
2	16	0100	
3	03	0103	} 送设备号
4	17	0050	
5	16	0101	

这样，从上述程序中，我们就可以看到利用计算机在必要的时刻，对不同的设备发出控制信息，实现了主机对外围设备的控制。

我们从以上二个例子，可以看到算术公式，控制过程的实现是由一些直接程序来组成和实现的，为了以后叙述循环程序，分支程序方便起见，我们把按算术公式，控制过程进行计算与控制的程序称为算

子，在编制算子时一般应注意以下几点：

1. 算式公式，控制过程必须算术化，特别控制过程必须演变成算术语言，或信息输出输入形式，以便程序实现。
2. 程序编制要简短，在许可的条件下，力求程序简便，应尽量把算式过程简化或作适当的改写，以便指令的精简。
3. 节省机器的计算时间，除用简化公式，缩短程序来减少运算量外，在编制程序的过程中，应尽量避免重复计算相同的量。
4. 占用工作单元少，编制程序时，充分利用工作单元，力求少用工作单元。

第三节 单循环的程序设计

1. 循环程序的引入：

解题时常会遇到运算量较大，但运算过程又具有一定规律的计算和控制过程，对于这种计算，如果按运算顺序逐条写出相应的指令来编制程序，不仅编写费事，而且程序十分冗长，有时还会因为内存的存贮量的限制使问题得不到解决。故我们根据计算本身所具有的规律性，我们可以找出一些办法来缩短程序，这样我们就可以引进循环程序的概念。

例一。我们要对开关量控制单元进行逐点采样，一共为64点，试用循环程序实现，采入结果放于0100~0177，开入设备号0200，程序0300开始。

其中：	0200	<设备号>	
	1	<000000>	
	2	<177700>循环量	
程序：	0300	03	0200
	1	16	0201
	2	17	0100
	3	(04	0100)
	4	11	0303
	5	11	0201
	6	11	0202
	7	07	0300

我们可以从以上例子看出，如不采用循环的方法，要实现以上目的，指令至少要100多条，而现在全过程只要十条左右，这就说明利用循环程序给我们带来了很大的方便。它的特点就是采用重复执行某段算子的方法来实现多次重复的计算，这种编制程序的方法称为循环程序设计，而编制的程序称为循环程序，循环程序设计是一种比较重要的方法，它具有程序简短的特点，这样不仅在一定程序上解决了存储量有限的矛盾，而且提高了计算机的解题能力，因此在实际工作中，应尽量设法把计算过程归纳为循环过程，用循环程序来加以实现。

一般说来，循环程序包括准备部分、工作部分（算子）、改变部分和控制部分，工作部分是循环程序的主体，完成主要的计算操作，工作等，改变部分的作用是对指令进行必要的修改，在JS-10A机上通常是用机器修改指令的功能加以实现（操作码不变，仅地址发生改变），控制部分是用控制循环来判定是否结束，准备部分则是对一些改变指令的单元送入必要的内容，保证循环程序正确地工作。

在实际工作中通常是按下述步骤进行编制的。

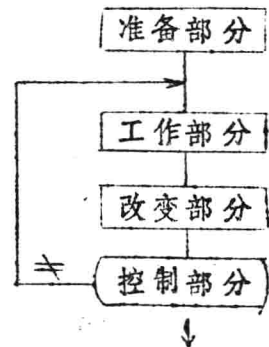
1. 改写计算公式和控制过程，排列有关数据和过程步骤，使之宜于编制循环程序。

2. 确定循环的组成部分和各部分的实现方法。

3. 为了便于编制程序，把确定的方案用框图描述出来。

这里我们把框图的概念介绍一下：

在编题过程中，对于简单的问题，我们可以直接编出程序，但是对于复杂的问题，特别对于某些科题的总控制图、逻辑图，直接编出程序就比较困难，因此人们利用框图比较直观地给出整个解题过程的进行顺序，有了框图不仅可以迅速编写程序，而且大大地减少了逻辑上的错误，框图一般可分粗框图、细框图二类，粗框图主要是给出解题过程中各部分之间的联系，细框图则是对某些算子设想的实现，通常我们用方框表示算子过程，用圆框表示逻辑算子，用“→”表示各框之间的联系。

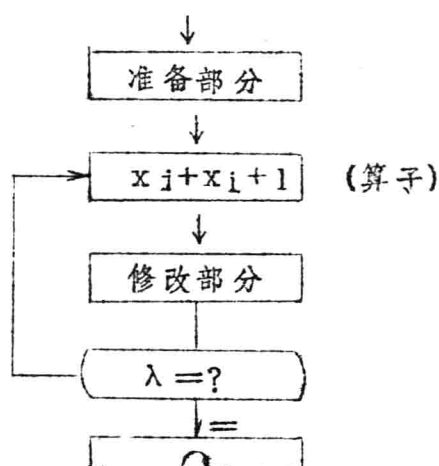


2. 循环的分类:

在实际工作中, 我们可对循环程序进行分类, 一种是循环次数已知的, 对于这一类, 在JS-10A机中一般可利用加1零转<11D>或利用某一单元作为计数器, 每循环一次, 计数器加1, 最后判断是否满足给定数来实现。

$$\text{例如: } y = \sum_{i=0}^m x_i$$

框图



单元分配:

x_i : 0100~0200

y : 0201

m : 0202

程序 0300 开始

以下我们用二种方法来加以实现

(1) 利用加1零转实现

0300	17	0000	入清“0”	} 准备部分
1	04	0201		
2	03	0316	循环次 → m	
3	04	0202		
4	03	0315	恢复程序	} 算子
5	04	0307		
6	03	0201	$x_i + x_{i+1}$	
7	(01	0100)		
10	04	0201		} 修改部分
1	11	0307	修改指令	
2	11	0202		} 控制部分
3	07	0306	判到否	
4	17	0401	Ω	
5	01	0100	常数	
6	17	7700		

(2) 利用计数器实现

0300	17	0000	入清“0”	准备部分
1	04	0201		
2	03	0314		
3	04	0305	恢复程序	
4	03	0201		
5	(01	0100)	$x_i + x_i + 1$	算子
6	04	0201	修改指令	
7	11	0305		修改部分
10	03	0305		
1	06	0315	判等否	控制部分
2	07	0304		
3	17	0401		
4	01	0100	常数	
5	01	0201		

从以上二个程序来看，都达到循环的目的，尽管他们实现方法不同，但大体上讲还是一致的，而且指令用得也有多少，我们可以根据实际情况，编制不同的循环程序。

第二类循环程序，由于它们的循环次数是未定的，它不能用固定记数的办法加以解决，这种循环程序是通过利用某个数学条件来解决，而在本机中则用判跳指令来实现循环的判别。

例如，计算 $y = \sqrt{x}$ ，设迭代误差为任意整数 ε 。

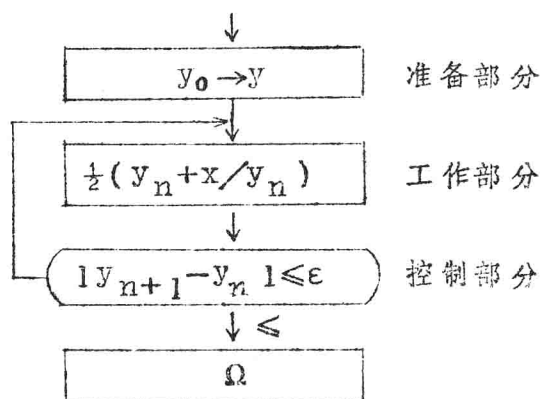
采用牛顿迭代公式：

$y_{n+1} = \frac{1}{2}(y_n + x/y_n)$ ，计算 y_n

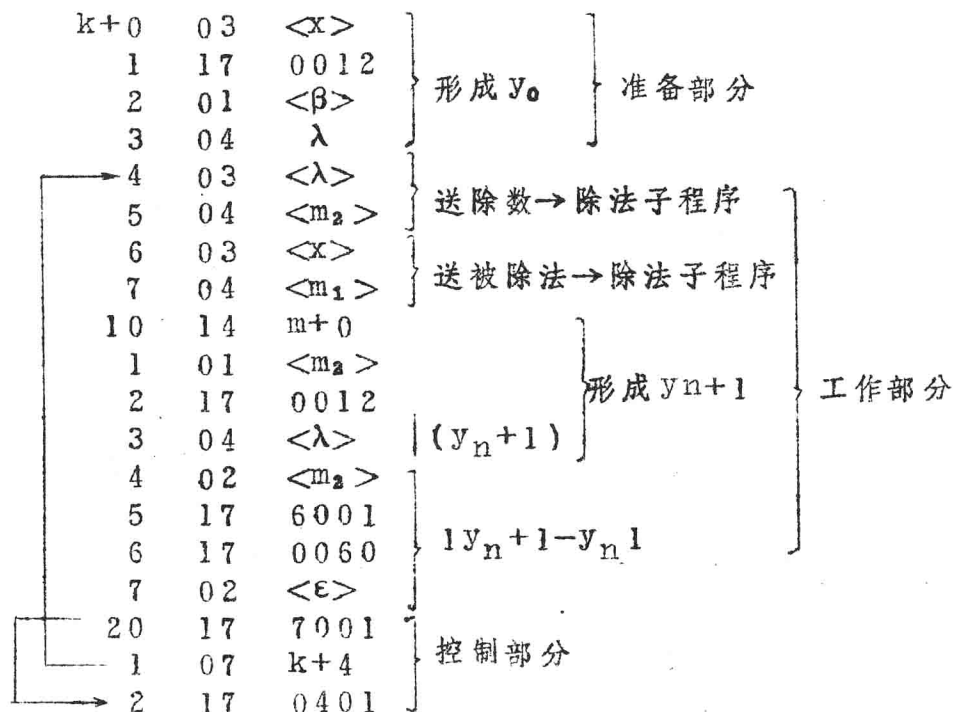
当 $|y_{n+1} - y_n| \leq \varepsilon$ ，则 $y_{n+1} = y_n$

其中 $y_0 = \frac{1}{2}x + \beta$ ， $\beta = 0.4258$

框图



本程序采用符号文字程序：

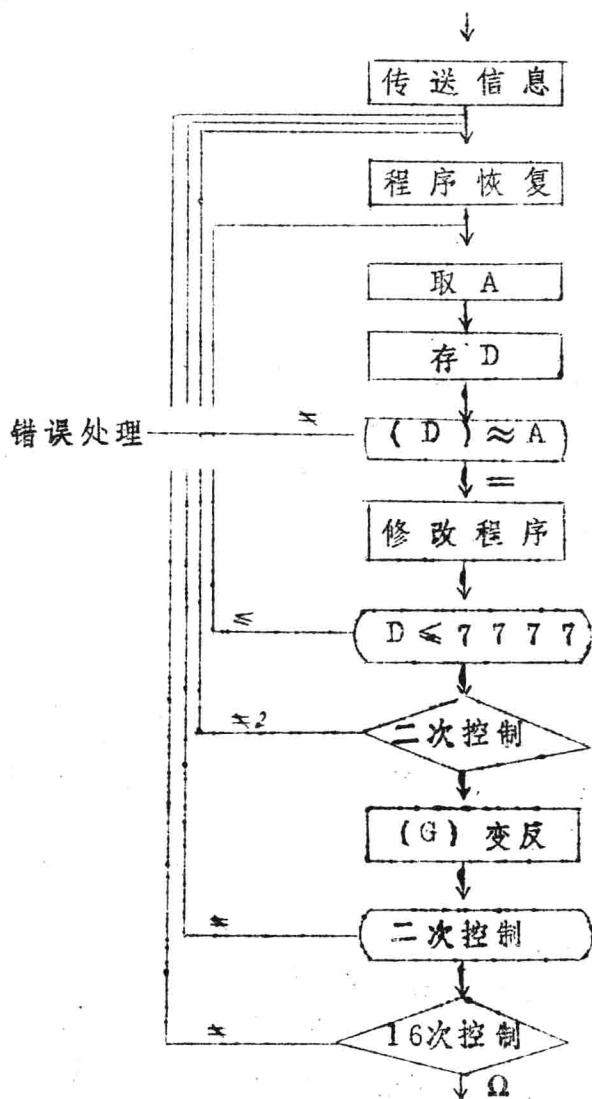


以上介绍了二种不同类型的循环，它们实现的方法虽然不同，但是，总的都实行了达到算子的目的，不但节约了指令，并把某些视不可能实现的课题变成现实，循环是手编程序的一个重要手段，希读者在熟悉的基础上，不断总结经验，使其成为自己一个重要的工具。

第四节 多重循环的程序设计

在上节介绍了单循环的编制方法，但在实际计算中，最终常遇到的不是单重循环能解决的，而是多重循环的问题，也就是说，是一重循环，套着一重循环，甚至几重套用。但我们只要掌握了单重循环的编制方法，对多重循环的编制是不会有有多大困难的。

例如：内存检查位电程序，它就是一个多重循环程序，现把其框图简介如下：



我们为了达到对内存每一单元的16位进行检查，看是否能进行读写“0”与“1”的功能，为之配备了这一检查程序，我们从这一简框图上可以看出，这个程序共有4重。

其中：第一重是对每一个内存单元进行检查；

第二重是为了检查磁蕊在再生过程中，是否会产生误码，故对磁蕊单元检查二边；

第三重：是对送入检查数（G）变反，进行变反状态检查；

第四重：是对送入数移位后，对第二位位电流进行检查，故必须进行16次的控制。

这样使程序对内存的每一颗磁蕊进行了“0”与“1”读写检查，並由于在检查的过程中，单元中的16位，只有一位为“1”，其它位为“0”或1位为“0”其它位为“1”，这样使数据在存贮过程中处于较困难的状态，因此通过此程序的检查之后，一般来讲，对内存的可靠准有一定的保证，由于程序在后面附上，就不在此处列出了。

但我们从上例的框图中可以看出，多重循环是单重循环的重复，但由于JS-10A机是利用单元作记数单元，所以在编制多重循环时，必须一重重地从内到外逐次进行，並搞清各重之间的关系，这样才能使程序在处理过程中正确无误，並做到结构明晰，在工作实践中，不断注意总结和积累编制程序的技巧和经验。

第五节 指令的修改和使用

在JS-10A机中，由于本机在设计过程中，硬件力求简单，因此在使用过程中，对程序进行循环，改变地址等均需由软件实现，也就是说，需要运用对指令进行修改来实现变址、判别、开关、转移等功能，以下为了便于初学者掌握对本机使用的特点，故在此把指令修改强调一下。

(1) 利用指令来实现变址

由于本机没有变址寄存器等硬件，故在程序里实现变址是利用修改指令的方法来实现，一般来讲，当地址修改量大于1时，就用指令加的方法来实现，当地址逐渐加1时，则可利用11指令来实现，前面需要三条指令（取数、定点加、存数），后者只要1条就可，这样实

现了变址。

(2) 利用指令修改来实现程序的变动

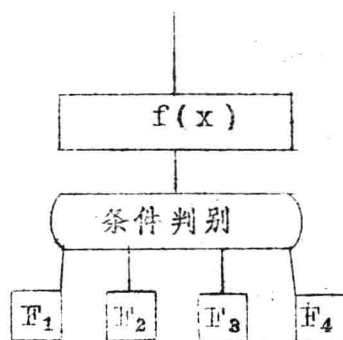
在工作中，有时需要对某一指令的操作码等发生变动，这时也需要利用指令修改的办法，来进行变化和修改。

(3) 总之，在计算机中指令、数、均可以看作成为一个16位二进制的代码，因此为了便于程序对任务的实现，利用加减法指令对某些指令、实现修改和改变，以便于实现变址和程序的控制是有很大的作用。

第六节 分支程序

在程序的编制过程中，常常需要根据变量满足某些因条件不同，选择不同的计算，控制程序段的方法，这样就提出了分支程序的设计要求。

分支程序一般框图，表现为右图，其中 $P_1, P_2 \dots P$ 分别表示各分支所对应的程序段。它根据算子的条件判定来实现分支的，一般来讲，一个条件可形成二个分支，但当算子变化较多，我们可以运用循环程序实行对多

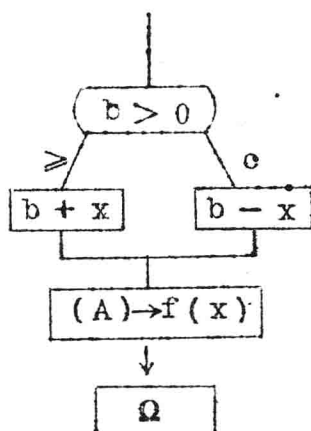


种条件用一块程序段来实现多次分支，这一点对于编写工业控制程序的同志应引起重视。例如在中断处理的问题上，本机通过中扩可扩充到64级中断，如用一般分支方法实行分支，则程序太长，而在中断程序中采用“伪入口”的方法结果，使程序比较灵活、简短，由于此程序在程序介绍上附出，故在此不再详述了。

$$\text{举例} \quad f(x) = \begin{cases} b+x & b \geq 0 \\ b-x & b < 0 \end{cases}$$

程序:

k+0	03	
1	17	6001
2	07	k+6
3	01	<x>
4	04	f(x)
5	17	0401
6	02	<x>
7	07	k+4



这里我们可以看到实现了二个分支，但运算完毕又回到主程序上。归纳分支程序，有以下几种编制方法：

1. 对算子的某个结果，可能出现的条件，常量进行比较和判断，使之满足于某一条件的与否，转向某段分支程序，这就是常用的比较法。

2. 某些程序，由于要修改和恢复，在这样的状态下，在判断式的出口放置一条指令，使程序时而转向某一段程序，时而转向另一段程序（条件是满足某一判断式的要求），而这类指令的位置起了开关的作用，所以称为程序开关法。

3. 采用简单分支组合的办法来组合实现多分支，这实际上是在判别某一条件实行分支的基础上，再对某一分支再进行判断，以实现分支，依次类推，实现多分支程序。

第七节 子 程 序

在解算大量实用问题时，有一些程序经常被反复使用，只是变量数值不同而已，而计算、控制方法是一致的。例如计算中，有 $10 \rightarrow 2$ ， $2 \rightarrow 10$ ， $\sin x$ ， $\cos x$ ， $x \dots\dots$ 等。控制过程中，中断信息的判断和处理，十进制数制表输出的打印程序，输入程序……等等。这些程序如每用一次都重新编制，就必定浪费大量的人力和机时，工作的实践告诉我们有必要把这些常用的程序独立起来构成一个单独套用程序，一旦使用时，只需套用和转向使用即可，工作完毕自动返回程序断点、继续工作，这种要经常用到的小程序一般称为子程序，子程序必需自带常数，工作单元，使子程序具有完整性和独立性，子程序必须设进口与出口，JS-10A中进子程序是用14指令，出口由于在14指令执行的同时形成一条返回指令，所以返回指令必须放在子程序里，并在进入子程序时，首先把出口返回指令存在子程序的最后一条这一工作做好，以便执行完子程序后退回主程序的断点上去，由于子程序在主程序里，可能要大量使用或重复使用，因此一般来讲，子程序应满足下列条件。

1. 指令要求精简，常数，工作单元要少，最好自带，以保持其独立性。
2. 计算方法，控制方法要力求简便，以便在满足精度的要求上，减少程序的长度和计算时间，但要保证精度。
3. 套用力求简单、方便。
4. 要程序自带恢复性，即能反复使用。
5. 子程序最好能浮动，但JS-10A寻址方法均是直接地址，因此不能实行浮动，如不能浮动，需代真也必须方便、简单，每个标准子程序应有完整的说明书，包括使用方法、算法、框图、程序说明，以便于使用者正确使用和工作参考查阅。

例如，踏步程序，在表演程序与调机程序中，由于对某一数据的输出，需要时间长一点，故套用踏步时间，延长时间，它先把返回地址送入0305，即工作程序的最后一条，随后再执行工作程序，程序自己带有常数，工作单元。