

X-1 机上用的统计预报方法

(一)

湖北省气象局
武汉大学数学系 编

一九七四年五月

前 言

我们编的这个手册性质的资料，目的是给使用 X—1 型电子计算机进行数理统计方法预报工作的同志提供一点方便；也作为我们学员《计算实习》的部分讲义；同时还想通过这份资料的交流更好地学习同志们的经验，以便把我们的工作搞得更好一些。

资料上介绍的方法和编制的程序不仅用于天气预报，其中大部分方法和程序也可用来计算其他领域（如水文、地震、冶金、纺织、化工等方面）中类似的问题。

第一章概括介绍 X—1 机的指令系统、使用说明和上机注意事项等。

第二章叙述的概率统计基本知识是以后各章要用到的。

第三章内容是武大计算数学专业 70 级气象小组的学员毕业实践小结的一部分，我们只对个别地方作了改动。

第三、四、五章的方法和程序我们用得最多，第六、七章的方法算题较少。特别是第八、九章的方法我们使用更少。

参加这份资料编写工作的有湖北省气象局科研究所、武汉中心气象台和武汉大学数学系统计预报小组。我们还在继续编写其他的方法和程序，准备陆续付印。在编写这份资料时，由于时间仓促，特别是我们算题经验不多，资料中一定有许多不妥和错误之处，希望同志们批评指正。来信请寄武汉中心气象台统计预报小组。

一九七四年四月

目 录

第一章 X—1 电子计算机的一般使用方法

§ 1. 概貌介绍.....	(1)
§ 2. 指令和数的表示.....	(3)
§ 3. 指令系统的功能.....	(6)
§ 4. 地址系统的使用.....	(17)
§ 5. 外部设备的使用.....	(18)
§ 6. 控制台操作.....	(22)
§ 7. 出错信息及其处理方法.....	(27)
§ 8. 上机的准备.....	(29)
§ 9. 标准子程序和公用常数.....	(30)
附表: 1. 常用数据的指令形式.....	(39)
2. X—1 机应用符号说明.....	(43)
3. 程序设计实例——主元消去法程序.....	(44)

第二章 概率统计基本知识

§ 1. 概率的基本概念.....	(49)
§ 2. 事件和概率的运算.....	(49)
§ 3. 随机变量及分布函数.....	(51)
§ 4. 随机变量的数字特征.....	(55)
§ 5. 统计检验.....	(59)

第三章 方差分析

§ 1. 方法简介.....	(72)
§ 2. 程序框图.....	(77)
§ 3. 实例计算.....	(79)
§ 4. 几点体会.....	(81)

第四章 回归分析

§ 1. 方法简介.....	(84)
§ 2. 程序框图.....	(95)
§ 3. 实例计算.....	(97)
§ 4. 几点说明.....	(100)

第五章 逐步回归分析

§ 1. 方法简介	(102)
§ 2. 程序框图	(110)
§ 3. 实例计算	(113)
§ 4. 几点说明	(115)

第六章 天气事件概率的回归估计

§ 1. 方法简介	(116)
§ 2. 程序框图	(120)
§ 3. 实例计算	(122)

第七章 判别分析

§ 1. 方法简介	(126)
§ 2. 程序框图	(130)
§ 3. 几点说明	(132)

第八章 序列回归外推法

§ 1. 方法简介	(134)
§ 2. 程序框图	(136)

第九章 自然正交分解

§ 1. 方法简介	(137)
§ 2. 程序框图	(147)

附 录

方差分析程序	附 1-10
逐步回归程序	11-24
概率回归程序	25-33
回归分析程序	34-46
判别分析程序	47-56
自然正交分解程序	57-72

第一章 X—1 电子计算机的一般使用方法

§ 1. 概 貌 介 紹

X—1 机是一架晶体管快速通用数字电子计算机。以单地址为指令结构，以浮点、二进制数为数的表示方式，共有64个操作码，字长45位。平均运算速度为 10 万次/秒。

1. 机器的主要设备

① 变址存贮器：容量64，字长16位。

② 内存贮器：磁心存贮器 $2 \times 16384 + 4096 + 2560 = 39424$ 字。

常数存贮器 4096字。

③ 外存贮器：磁鼓 4 台，鼓号为 0，1，2，3

3 鼓的容量为 0、1、2 鼓容量的 2 倍；0、1、2 鼓每台容量为 32640 字。

磁带 4 台，带号为 1，2，3，4。每台容量为 204800 字。

④ 输入设备：五单位光电输入机 2 台，台号为 1，2

八单位光电输入机 2 台

⑤ 输出设备：即印机 1 台 28 行快速打印机，打印速度 15 行/秒。

组印机 2 台 28 行快速打印机，

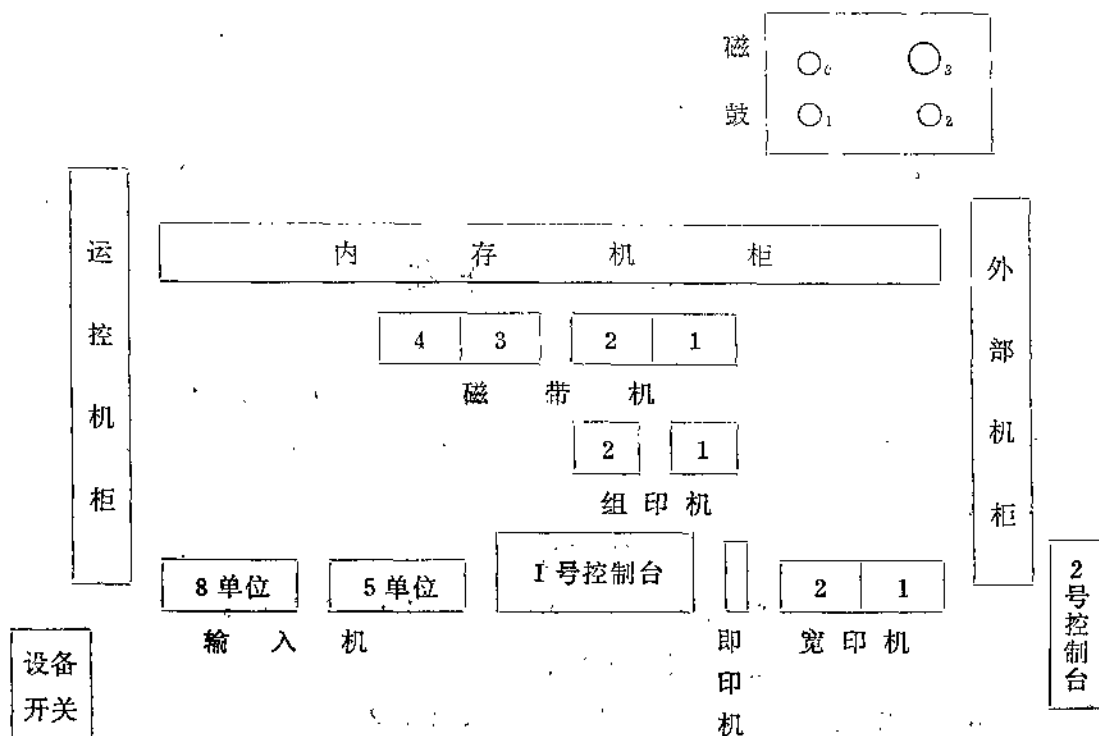
台号为 1，2。打印速度 15 行/秒。

宽印机 2 台 70 行宽行打印机，

台号为 1，2。打印速度 8 行/秒。

⑥ 控制台：2 个 1 号控制台为算题者控制操作用。

2 号控制台为机房工作人员外部调机用。



机房主要设备示意图

2. 中央控制部件

- ① 指令控制器 包括指令寄存器 ($ZL-J$)，指令地址计数器 (L)，地址加法器 (F)，上下界地址寄存器 (A, b) 等。
- ② 变址存贮器 字长16位二进制数，容量64，从 $(00)_2$ — $(77)_2$ ，00 固定为零状态。46~77 单元的使用，应严格遵照管理程序规定方案。运算时，如最高位 (第16位) 有进位则去掉，因此要减去一数可加以该数的补码。

用户使用变存可见下表说明：

编 号	说 明
0 0	固定为 0
01—45	用户使用
46—55	标准子程序专用
56—71	外部设备线路固定用的地址和个数寄存器
72—77	管理程序专用工作单元

③ 辅助存贮器

A 寄存器：(编号140011)字长45位，为运算器中的累加器，称为累加寄存器。

Q 寄存器：(编号140012)字长45位，也是运算器中的累加器，称为乘商寄存器。

P 寄存器：字长45位，从内存单元 M 中取出的数，先存放寄存器 P 中，然后和 A 或 Q 的内容进行运算。

ω 寄存器：字长1位。用于存放运算结果的标志。如结果的符号或转移信号等。

- ④ 二极管寄存器 ($\phi_1 - \phi_7$)：编号为110001—140007，是设在控制台上七排45字长的拨码开关，可由人工拨出信息、数或指令。作为控制用，只能读出、运算，但不能写入。

⑤ 内存贮器

名 称	编 码	容量(字数)	说 明
2N 号磁心存贮器	000000—077777	32768	用户使用共 39424 字，数据和程序最好分放在两个不同的存贮器中。
0N 号磁心存贮器	100000—107777	4096	
1N 号磁心存贮器	110000—114777	2560	打印缓冲区，用户一般禁用，宽印缓冲区为115000—117777，即印组印缓冲区为116000—117777。
	115000—117777	1536	
	120000—137777	8192	空 白
辅助存贮器	140000—140017	12	控制台上专用
	140020—157777	8180	空 白
常数存贮器	160000—167777	4096	只可取用，不可写入
	170000—177777	4096	空 白

§ 2 指令和数的表示

1. 二进制数和八进制数

机器的运算是采用二进制数，而指令和数的填写又往往是采用二进制和八进制数。二进制数就是“逢二进一”的数的表示方法；八进制数则是“逢八进一”，如：

十进制	0	1	2	3	4	5	6	7	8	9	10	11	12
八进制	0	1	2	3	4	5	6	7	10	11	12	13	14
二进制	0	1	10	11	100	101	110	111	1000	1001	1010	1011	1100

下面介绍不同进制数之间的转换方法，这里只讲整数的转换。如十进制数12写成八进制数就是14，写成二进制数就是1100，即 $(12)_{10} = (14)_8 = (1100)_2$ 。

因为
$$\begin{array}{r} 8 \overline{) 12} \\ \underline{8} \\ 4 \end{array}$$
 余 4

即 $(12)_{10} = 1 \times 8^1 + 4 = (14)_8$

也就是说，12被8除后的余数4就是八进制数的个位，而其商1（若它小于8时）是八进制数的8¹位。

例： $(26)_{10} = (?)_8$

因
$$\begin{array}{r} 8 \overline{) 26} \\ \underline{24} \\ 2 \end{array}$$
 余 2

故 $(26)_{10} = 3 \times 8^1 + 2 = (32)_8$

又例： $(117)_{10} = (?)_8$

因
$$\begin{array}{r} 8 \overline{) 117} \\ \underline{144} \end{array}$$
 余 5，故 $(117)_{10} = 14 \times 8 + 5$

而
$$\begin{array}{r} 8 \overline{) 14} \\ \underline{8} \\ 6 \end{array}$$
 余 6 $(14)_{10} = 1 \times 8 + 6$ 代入上式

得
$$\begin{aligned} (117)_{10} &= 14 \times 8 + 5 \\ &= (1 \times 8 + 6) \times 8 + 5 \\ &= 1 \times 8^2 + 6 \times 8^1 + 5 = (165)_8 \end{aligned}$$

例： $(8532)_{10} = (?)_8$

因
$$\begin{array}{r} 8 \overline{) 8532} \\ \underline{81066} \\ 8 \overline{) 133} \\ \underline{816} \\ 8 \overline{) 16} \\ \underline{16} \\ 0 \end{array}$$
 余数 $\begin{array}{l} 4 \\ 2 \\ 5 \\ 0 \end{array}$ $\uparrow$ 即 $(8532)_{10} = (20524)_8$

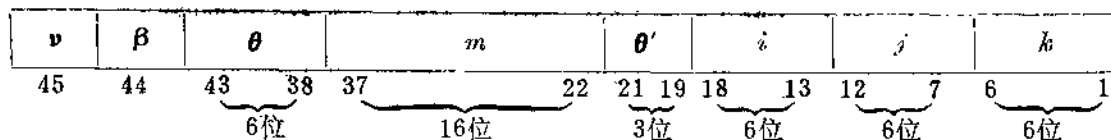
实际上，
$$\begin{aligned} (8532)_{10} &= 1066 \times 8 + 4 \\ &= (133 \times 8 + 2) \times 8 + 4 \\ &= [(16 \times 8 + 5) \times 8 + 2] \times 8 + 4 \\ &= \{[2 \times 8 + 0] \times 8 + 5\} \times 8 + 2 \times 8 + 4 \\ &= 2 \times 8^4 + 0 \times 8^3 + 5 \times 8^2 + 2 \times 8^1 + 4 \\ &= (20524)_8 \end{aligned}$$

由八进制数再化成二进制数，只需将0, 1, 2, 3, 4, 5, 6, 7相应地写成000, 001, 010, 011, 100, 101, 110, 111就可以了。如

$$(117)_{10} = (165)_8 = (001\ 110\ 101)_2 = (1110101)_2$$

$$(8532)_{10} = (20524)_8 = (010\ 000\ 101\ 010\ 100)_2 = (10000101010100)_2$$

2. 指令形式



ν : 间接地址特征位[1位]。 $\nu=1$ 表示间接地址, $\nu=0$ 表示非间接地址。

β : 加下界(b)的特征位[1位]。 $\beta=1$ 表示加下界(b), $\beta=0$ 表示不加下界(b)。

(b)是在下界的专用寄存器。

θ : 主操作码[6位]。由2个八进制数表示。

m : 基本地址或常数[16位]。由6个数字(1个二进制数和5个八进制数)表示。

θ' : 变址操作码也称副操作码[3位]。由一个八进制数表示, 可实现8种地址修改和变址操作。

i, j, k : 变存地址、某种专用信息或地址修改常数[各6位]。各由2个八进制数表示。

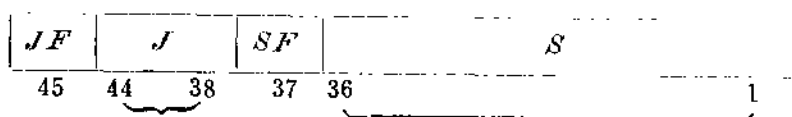
3. 数的表示

① 在 $A-1$ 机上, 数被表示为如下的浮点二进制形式,

$$N = \pm 2^{\pm J} \cdot S$$

其中 J 为阶, S 为尾数。

存贮单元中位数的分配为:



其中 JF 为阶的符号位, SF 为数的符号位, 均以“0”表示正, 以“1”表示负。尾数 S 用原码表示。阶 J 用补码表示(正阶的补码即原码本身)。

② 数的二进制表示范围是

$$0 \leq |N| \leq 2^{127} \cdot (1 - 2^{-36})$$

$$-2^7 \leq J \leq 2^7 - 1$$

$$0 \leq S \leq 1 - 2^{-36}$$

相应于十进制数的范围是 $0 \leq |N| \leq 10^{39} \cdot 0.170215851$

数的表示范围为 $[\pm 10^{-39}, \pm 10^{39}]$ 或 $[\pm 10^{-128}, \pm 2^{127}]$

因此在写十进制数时 J 不能超过 $(38)_{10}$

③ 数的书写格式

i 浮点十进制数书写格式

编码	JF	J	SF	S	说 明
0		03	1	456 . 789 . 321	-456.789321
1	1	02		345 . 678 . 987	0.00345678987
2				100 . 000 . 000	$0.1 = \frac{1}{10}$

ii 八进制数的指令形式

如上述 $\frac{1}{10}$ 该数为例。

$$\frac{1}{10} = (0.1)_{10} = (0.0631463146315)_8$$

经规格化，写成 $2^{-8} \cdot (0.631463146315)$ ，负阶用补码表示，即得该数的指令书写形式：

编码	v	β	θ	m	θ'	i	j	k	说 明
2	1	1	75	063146	3	14	63	15	$\frac{1}{10} = 0.1$

§ 3 指令系统的功能

1. 指令系统说明

00 不操作指令[R0]：本指令对运算器不操作，但变址运算照常可以进行，也可以进行间接地址操作。这里间接地址不是为了产生执行地址，而是取其其中的变址运算。该指令不能访问辅助存储器。

01 必转指令[转]：本指令即为无条件转移指令。在非间接地址时，就按执行地址 M 转移；但在间接地址时，不立即转移，先执行间接地址操作，直至 $v=0$ 时，再按最后形成的执行地址转移。在间接过程中，变址运算都按变址运算规则进行。该指令不能访问辅助存储器。

02 零转指令[Z0]：本指令为条件转移指令。当 $\omega=0$ ，又非间接地址情况下，则立即按执行地址 M 转移；若 $\omega=0$ ，且为间接地址情形时，则在第二个基本周期，改为必转指令执行下去。当 $\omega=1$ 时，则不执行变址运算和间接地址操作，而顺次地执行下一条 $(L)+1$ 指令。

03 定转指令[Z1]：本指令为条件转移指令。当 $\omega=1$ ，又非间接地址情况，则立即按执行地址 M 转移；若 $\omega=1$ ，且为间接地址情形时，则在第二个基本周期，改为必转指令执行下去。当 $\omega=0$ 时，则不执行变址运算和间接地址操作，而顺次执行下一条 $(L)+1$ 指令。

04 $j > 0$ J 转指令 [ZJ]：本指令为记数转移指令，也是一种条件转移指令。其算法如下：① $(j) - 1 \rightarrow j$ ，② 判别修改后的 (j) 是否大于零，若 $(j) > 0$ ，又非间接地址情形，则按 $M = m + (i) + \beta(b)$ 转移，此时 θ' 不起作用。若 $(j) > 0$ ，且为间接地址情形，则首先按 $M = m + (i) + \beta(b)$ 去取间接地址（此时 θ' 也不起作用，但是以后间接过程中 θ' 起作用），直至 $v = 0$ 时，按执行地址转移。若 $(j) \leq 0$ （由于 (j) 均为正整数，不可能为负，此时实为 $[j] = 0$ ），则程序顺次执行下去，此时不做任何变址运算。注意本指令约定开始时 $[j] \neq 0$ ，如果开始 $[j] = 0$ ，则顺次执行下一条，而不按执行地址转移，且 $[j]$ 变为 177777。

05 停转指令：即停机指令。在程序中必须写为

v	β	θ	m	θ'	i	j	k
0	0	05	160000	2	00	00	51

06 执行指令[执行]：本指令按形成的执行地址，取一个全字长单元的内容作为指令来执行。执行完毕后，只要执行的不是转移指令，程序仍按原顺序继续做下去。该指令不能访问辅助存储器。

07~11 与“01”[必转]的功能相同，一般都不用。

12 进退指令 [JT]：管理程序用。

13 启动指令 [QD]：一般用户很少用。

14 送变存指令 [SB]：本指令用来向变存单元送数或修改变存单元内容，亦可进行两个变存单元内容的大小比较，执行过程中 A, Q 内容不变。其算法为：

$$m + (i) - (j) \rightarrow k,$$

$$\text{若 } m + (i) \leq (j) \quad \text{则 } 1 \rightarrow \omega$$

$$\text{若 } m + (i) > (j) \quad \text{则 } 0 \rightarrow \omega$$

本指令的 v, β, θ' 无意义，不起任何作用，且不能访问辅助存储器。

15 变存送 Q 指令 [BQ]：本指令用来把变存内容送到运算器 Q 中，即

$$[(i) + k] \rightarrow Q_{37 \sim 22}, \quad [Q]_{47 \sim 38, 21 \sim 1} \equiv 0$$

即只传送16位的内容到 Q 的第22位到37位上，其余位为零。该指令的 v, β, θ' 无意义，不起任何作用，且不能访问辅助存储器。

16 变存送 A 指令 [BA]：本指令用来把变存内容送到运算器 A 中，即

$$[(i) + k] \rightarrow A_{37 \sim 22} \quad [A]_{47 \sim 38, 21 \sim 1} \equiv 0$$

该指令的 v, β, θ' 无意义，不起任何作用，且不能访问辅助存储器。

17 A 送变存 [AB]：本指令用来将 A 中第22~37位内容送入变存单元，即

$$[A]_{37 \sim 22} \rightarrow (i) + k$$

注意此时 (i) 也只有最后六位起作用。 A 中其余位可以是零也可以不是零，皆无妨碍。

该指令的 v, β, θ' 无意义, 不起任何作用, 且不能访问辅助存储器。

20 内存送变存[MB]: 本指令用来把内存某单元中22—37位送至某一变存单元中, 即

$$[m + (i) + (j) + \beta(b)]_{37 \sim 22} \rightarrow k$$

注意该内存单元其余位可以是零也可以不是零, 皆无妨碍。该指令的 v, θ' 无意义, 不起任何作用, 且不能访问辅助存储器。

21 变存送内存[BN]: 本指令用来把变存内容送入内存某单元的第22~37位, 即

$$[k] \rightarrow m + (i) + (j) + \beta(b) \quad [m + (i) + (j) + \beta(b)]_{45 \sim 38, 21 \sim 1} \equiv 0$$

该指令 v, θ' 无意义, 不起任何作用, 且不能访问辅助存储器。

22 即印指令[JY]: ①本指令是为了满足程序执行中要求及时输出少量全字长信息或16位字长信息之用。它不象成组输出需要填写控制字, 也不需要等待分时, 可以及时输出。但大量输出不宜采用本指令, 因为在执行本指令的整个过程[包括打印的机械过程]不能分时, 等于串行工作。所以只适用输出少量信息, 如迭代误差、次数等使用, 而大量输出结果最好用控制字转管理程序执行。

② 本指令可以输出三种不同形式的信息, 即指令形式, 数形式和地址形式。分别由特征位 k_6 和 k_5 (即由 k 的第6位和第5位) 来控制。当 $k_6 = 0, k_5 = 1$ 时, 按浮点数十进制形式输出, 即数形式印刷出执行地址 M 的内容。当 $k_6 = 1, k_5 = 0$ 时, 按地址形式输出, 即印刷出执行地址 M 。[但当 $m = 0$ 时, 可即印变存 i 中的内容]

③ 标志码: 在印刷出的信息前面, 由 k_4, k_3, k_2, k_1 控制印刷出不同的标志码。对应关系列表如下:

$k_4 k_3 k_2 k_1$	00	01	02	03	04	05	06	07	10	11	12	13	14	15	16	17
标志码	0	A	B	C	D	E	F	G	H	I	J	K	L	M	N	ϕ

本指令不能直接访问辅助存储器。即是说不能直接即印 A, Q 中的信息。该指令的 v, θ' 无意义, 不起作用, $M = m + (i) + (j) + \beta(b)$ 。

23 内存发送[NS]: 作备用。

24 内存送 A [NA]: 本指令将内存信息传送到运算器 A , 即 $[M] \rightarrow A$ 。

注意 M 不能为 140011 如 140012, 亦即不能实现 $[A] \rightarrow A$ 和 $[Q] \rightarrow A$ 。

25 A 送内存[AN]: 本指令将 A 内容传送到内存, 即: $[A] \rightarrow M$

注意 M 不能为: 140001~140012, 140015~140017等, 亦即不能实现 $[A] \rightarrow A, [A] \rightarrow Q, [A] \rightarrow e_1, \dots, e_7$ 等拨数开关单元, $[A]$ 送警报信号寄存器等。

26 内存送 Q [NQ]: 本指令将内存信息传送到运算器 Q , 即: $[M] \rightarrow Q$

注意 M 不能为: 140011, 140012, 亦即不能实现 $[A] \rightarrow Q, [Q] \rightarrow Q$ 。

27 Q 送内存[QN]: 本指令将 Q 内容传送到内存某单元, 即: $[Q] \rightarrow M$

注意 M 不能为 140001~140012, 140015~140017, 亦即不能实现 $[Q] \rightarrow A, [Q] \rightarrow Q, [Q] \rightarrow e_1, \dots, e_7, [Q]$ 送警报信号寄存器。

30 送 A 符号[SA]: 本指令用以实现按内存某单元 M 的数符改变 A 的数符。即

$$[M]_H [A] \rightarrow A.$$

其中 $[M]_H$ 表示 M 单元中数的数符。本指令不能访问辅助存储器。

31 ω 送内存[WN]: 本指令为保存 ω 信号而设, 其算法如下:

$$\omega \cdot [A] \rightarrow M$$

本指令不能访问辅助存储器。

32~37 与“00”[不操作]的功能相同。

40~47 对 A 加法[+]、对 A 减法[-]、对 A 反减[F1]、对 $|A|$ 减法[A1], 对 Q 加法[Q0], 对 Q 减法[Q1]、对 Q 反减[Q2]、对 $|Q|$ 减法[Q3]。

以上八种操作均为规格化操作, 因此运算结果肯定为规格化的数。如为零则45位全为零。本机器没有正零、负零或机器零之分, 如操作数为非规格化数, 则结果仍然正确。

50(52) 对 A 乘法[*] (对 Q 乘法[Q4])

本指令算法为:

$$[A] \rightarrow Q, [A] \times [M] \rightarrow A, ([Q] \text{ 不变}, [Q] \times [M] \rightarrow A)$$

亦即 A 的内容首先放入 Q 中保存, 然后再与内存中的数相乘, 结果送入 A 。本指令也是有规有舍操作。如果乘数和被乘数皆为规格化数, 则乘积亦为规格化数; 若乘数或被乘数有为非规格化数时, 乘积结果可能为非规格化数, 因为乘法操作只规格化一位。

51(53) 对 A 除法[/] (对 Q 除法[Q5])

本指令算法为:

$$[A] \div [M] \rightarrow A \rightarrow Q \quad ([Q] \div [M] \rightarrow A \rightarrow Q)$$

本指令要求除数必须是规格化数, 否则结果不正确。如果被除数为非规数, 而除数为规格化数, 则结果正确, 但不一定为规格化数, 如两数皆为规格化数, 则结果肯定为规格化数。

54~63 为双倍长四则运算, 见 $X-1$ 机指令系统表。

64 对 A 比大[AD]: 本指令算法为:

当 $[A] > [M]$ 时 $1 \rightarrow \omega$; 当 $[A] \leq [M]$ 时, $0 \rightarrow \omega$, $[A]$, $[Q]$ 不变。

本操作是将 A 和 M 中的两数进行比较, 得出 ω 信号, 以便程序控制转移时用。参与比较的两个数, 应该是规格化数, 则比较结果方能得出正确的信号, 如果是非规格化数进行比较, 则可能得出不正确的信号。

65 对 A 比小[<]: 本指令算法为:

当 $[A] < [M]$ 时, $1 \rightarrow \omega$; 当 $[A] \geq [M]$ 时, $0 \rightarrow \omega$, $[A]$, $[Q]$ 不变。

本指令同样要求参加比较的数为规格化数, 否则结果可能不正确。

66 对 A 全同比较[≡]: 本指令算法为:

当 $[A] \equiv [M]$ 时, $1 \rightarrow \omega$; 当 $[A] \neq [M]$ 时, $0 \rightarrow \omega$, $[A]$ 不变, $[A] \oplus [M] \rightarrow Q$ 。

本操作为45位全同比较, 若全同则 $1 \rightarrow \omega$, 否则 $0 \rightarrow \omega$ 。同时将 $[A]$ 和 $[M]$ 按位加的结果送入 Q 中。($0 \oplus 0 = 0$, $0 \oplus 1 = 1$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$)。

67 A 大 Q 小指令[48]: 本指令算法为:

当 $|[A]| \geq |[M]|$ 时, $1 \rightarrow \omega$; A 不变, $[M] \rightarrow Q$;

当 $|[A]| < |[M]|$ 时, $0 \rightarrow \omega$, 不规不舍。

本操作是将 A 中之数的绝对值和内存中之数的绝对值进行比较, 比较结果, 将绝对值大的那个数(包括符号)送进 A 中, 小的那个数送进 Q 中。如果两者绝对值相等, 则 A 中之数不动, 只将 M 中之数送进 Q 中。

本指令要求参与比较之两数均为规格化数。否则, 比较结果不一定正确。

70 对 A 逻辑和[V]: 本指令算法为:

$[A] \rightarrow Q, [A] \vee [M] \rightarrow A$ 。当 $[A] \equiv 0$ 时, $1 \rightarrow \omega$; 当 $[A] \neq 0$ 时, $0 \rightarrow \omega$, 不规不舍。
 $[0 \vee 0 = 0, 0 \vee 1 = 1, 1 \vee 0 = 1, 1 \vee 1 = 1]$

本操作先将 A 中之信息送入 Q 中, 然后与 M 中的数作逻辑和运算, 结果送入 A 中。如果操作结果为零, 则 $1 \rightarrow \omega$, 否则 $0 \rightarrow \omega$ 。

71 对 A 逻辑乘[∧]: 本指令算法为:

$[A] \rightarrow Q, [A] \wedge [M] \rightarrow A$ 。当 $[A] \equiv 0$ 时, $1 \rightarrow \omega$; 当 $[A] \neq 0$ 时, $0 \rightarrow \omega$, 不规不舍。

本操作先将 A 中之信息送入 Q 中, 然后与 M 中的数作逻辑乘运算, 结果送入 A 中。如果操作结果为零, 则 $1 \rightarrow \omega$, 否则 $0 \rightarrow \omega$ 。 $[0 \wedge 0 = 0, 1 \wedge 0 = 0, 0 \wedge 1 = 0, 1 \wedge 1 = 1]$

72 对 A 指令加[L0]: 本指令算法为:

$$\begin{cases} [A]_{45 \sim 38} + [M]_{45 \sim 38} \rightarrow A_{45 \sim 38} \pmod{2^{45}} \\ [A]_{37 \sim 1} + [M]_{37 \sim 1} \rightarrow A_{37 \sim 1} \pmod{2^{37}}, \end{cases} \quad [Q] \text{ 不变, 不规不舍, } [A]_H \rightarrow \omega$$

本操作将指令分为两段(1~37位和38~45位两段), 分别进行代码求和。如果37位有进位, 则并不进到38位上, 而予以舍弃掉。同样, 若45位有进位, 也并不循环进到第1位上, 而予以舍弃掉。

73 对 A 指令减[L1]: 本指令算法为:

$$\begin{cases} [A]_{45 \sim 38} - [M]_{45 \sim 38} \rightarrow A_{45 \sim 38} \pmod{2^{45}} \\ [A]_{37 \sim 1} - [M]_{37 \sim 1} \rightarrow A_{37 \sim 1} \pmod{2^{37}}, \end{cases} \quad [Q] \text{ 不变, 不规不舍, } [A]_H \rightarrow \omega$$

本操作将指令分为两段(1~37位和38~45位)分别进行代码相减。实际实现是将 M 中内容的两段分别用补码形式来和 A 的对应两段相加。在相加过程中, 第37位, 第45位有进位都舍掉。

74 对 A 加阶[JO]: 本指令算法为:

$[A]_J + M_{8 \sim 1} \rightarrow A_J$ 。且当 $[A]_J \leq -128$ 时, $0 \rightarrow A_J$; 当 $[A]_J > -128$ 时, $[A]_J$ 不变。
 $[Q]$ 不变, $[A]_H \rightarrow \omega$ 。

本操作是将本指令的执行地址的第1~8位 $[M_{1 \sim 8}]$ 代码与 A 中的阶码部分的内容进行代数求和, 结果放在 A_J 中。

本指令主要用于对 A 中的二进制数, 乘以2的 n 次方, 既方便又省时间, 其中 $M_{7 \sim 1}$ 就为这个以2为底的幂数的指数值, M_8 就为指数的符号标志, 对应于运算器中数的阶部分来说, M_8 就为阶的符号标志, $M_{7 \sim 1}$ 为阶的值。如果对 A 减阶, 幂指数为负指数, 则必须以

补码形式填写 $M_{8\sim 1}$, 运算结果才正确。

注意, 如果原来 $[A]_8$ 为零, 则结果为零。

75 对 A 移位 $[YA]$

本操作将 A 中代码按本指令地址部分的 $M_{7\sim 1}$ 所指出的值进行移位, M_8 为左移或右移标志。若 $M_8=0$, 则左移; 若 $M_8=1$, 则右移。执行指令后 $[Q]$ 不变, 且 $[A]_H \rightarrow \omega$ 。

76 对 A 分离 $[FA]$ 。本指令算法为:

$2^{-36}[A]_{28} \rightarrow A$, 不规不舍; $[A]_{xs} \rightarrow Q$, 规格化, $[A]_H \rightarrow \omega$ 。

本操作将 A 中之数分离为整数和小数两部分, 整数存放在 A 中, 小数为规格化之数存放在 Q 中。

注意 Q 中始终为正小数。

77 双倍长规格化 $[SG]$ 见 X-1 机指令系统表。

2. X-1 机 指 令 系 统 表

名 称	编号	符号	算 法 内 容	v	β	θ'	A, Q 状态	ω 状态
不 操 作	00	B0	运算器不做任何操作, 状态保持不变, 但可做变址运算或间接地址				不变	不变
必 转	01	转	按所形成的地址 M 转移, 即 $M \rightarrow L$				不变	不变
零 转	02	Z0	当 $\omega=0$ 时, $M \rightarrow L$; 当 $\omega=1$ 时, 变址运算不进行, 顺序执行下一条				不变	不变
一 转	03	Z1	当 $\omega=1$ 时, $M \rightarrow L$; 当 $\omega=0$ 时, 变址运算不进行, 顺序执行下一条				不变	不变
$(j) \rightarrow 0$ 转	04	ZJ	$(j)-1 \rightarrow j$, 当 $(j) > 0$ 时 $\begin{cases} v=0 & M=m+(i)+\beta(b) \rightarrow L \\ v=1, & \text{实现间接直至 } v=0 \text{ 时 } M \rightarrow L \end{cases}$ 当 $(j)=0$ 时, 顺序执行下一条			$\times$	不变	不变
停 机 转	05		当 $\sum Z_{C_i} \neq 0$ 时, $M \rightarrow L$; 当 $\sum Z_{C_i} = 0$ 时停机				不变	不变
执 行	06	执行	将 $[M]$ 作为指令执行之				不变	不变
	07							

X-1 机 指 令 系 统 表(续)

名 称	编 号	符 号	算 法 内 容	ν	β	θ'	A, Q 状态	ω 状态
	10	WZ						
	11	FJ						
进 退	12	JT	j_1 进入不允许中断状态, j_2 进入管理程序状态, j_3 退出不允许中断状态, j_4 退出管理程序状态, j_5 则 $m+(i) \rightarrow L$	$\times$	$\times$	$\times$	不变	不变
启 动	13	QD		$\times$	$\times$	$\times$		
送 变 存	14	SB	$m+(i)-(j) \rightarrow k$ 若 $m+(i) \leq (j)$, 则 $1 \rightarrow \omega$ 若 $m+(i) > (j)$, 则 $0 \rightarrow \omega$	$\times$	$\times$	$\times$	不变	
变存送 Q	15	BQ	$[(i)+k] \rightarrow Q_{87 \sim 22}$, $[Q]_{45 \sim 38} \equiv [Q]_{21 \sim 1} \equiv 0$	$\times$	$\times$	$\times$	[A] 不变	不变
变存送 A	16	BA	$[(i)+k] \rightarrow A_{87 \sim 22}$, $[A]_{45 \sim 38} \equiv [A]_{21 \sim 1} \equiv 0$	$\times$	$\times$	$\times$	[Q] 不变	不变
A 送变存	17	AB	$[A]_{87 \sim 22} \rightarrow (i)+k$	$\times$	$\times$	$\times$	不变	不变
内存送变存	20	NB	$[m+(i)+(j)+\beta(b)]_{87 \sim 22} \rightarrow k$	$\times$	$\times$	$\times$	不变	不变
变存送内存	21	BN	$(k) \rightarrow m+(i)+(j)+\beta(b)$, $[m+(i)+(j)+\beta(b)]_{45 \sim 38} \equiv 0$ $21 \sim 1$	$\times$	$\times$	$\times$	不变	不变
即 印	22	JY	(见说明)	$\times$		$\times$		
内存发送	23	NS	(备用)					
内存送 A	24	NA	$[M] \rightarrow A$				[Q] 不变	$[M]_H \rightarrow \omega$
A 送内存	25	AN	$[A] \rightarrow M$				不变	不变

Y-1 机 指 令 系 统 表(续)

名 称	编号	符号	算 法 内 容	ν	β	θ'	A, Q 状态	ω 状态
内存送 Q	26	NQ	$[M] \rightarrow Q$				$[A]$ 不变	$[M]_H \rightarrow \omega$
Q 送内存	27	QN	$[Q] \rightarrow M$				不变	不变
送 A 符号	30	SA	$[M]_H \cdot [A] \rightarrow A$				$[Q]$ 不变	不变
ω 送内存	31	WN	$\omega \cdot [A] \rightarrow M$				不变	不变
	32							
	33							
	34							
	35							
	36							
	37							