

大专学校教学用书

微机应用与 数控技术

孙奎武 张 弛 王子贵 编

下 册

华中理工大学出版社

微机应用与数控技术

(下册)

孙奎武 张 弛 王子贵编

华中理工大学出版社

内 容 简 介

本书较详细、全面地阐述了微机原理、应用及其与数控技术的结合。分上、下两册出版。

上册，主要阐述计算机基础知识、微机原理及如何应用于数控方面，并附有大量实用程序。

下册，主要阐述数控技术中的插补原理及其实现方法、程序编制、输入、输出和检测装置，最后以MNC数显装置、MNC数控车床和小型计算机CNC数控车床为实例作了综合的论述。

本书内容丰富、实用性强，能使读者获得微机应用与数控技术较全面的知识，为设计微机软硬件打下良好的基础。

本书可作大专院校、职工大学、电视大学的教材，也可供广大工程技术人员参考。

微机应用与数控技术

(下册)

孙奎武 张 弛 王子贵编

责任编辑 韩瑞根

华中理工大学出版社出版发行

(武昌喻家山)

新华书店湖北发行所经销

武汉市汉桥印刷厂印刷

开本：787×1092 1/16 印张：8 插页：1 字数：192 000

1987年9月第1版 1990年4月第2次印刷

印数：2501—5500

ISBN 7-5609-0042-9/TP·7

定价：2.90元

目 录

第二篇 数控技术

第七章 数控的插补原理

§ 7-1 数字乘法器	(2)
§ 7-2 逐点比较法的插补运算	(5)
§ 7-3 数字积分器	(27)

第八章 输入装置

§ 8-1 概述	(32)
§ 8-2 穿孔纸带的规格与编码	(32)
§ 8-3 加工程序编制	(42)
§ 8-4 数值计算	(46)
§ 8-5 加工程序编制实例	(51)
§ 8-6 典型零件加工程序	(53)
§ 8-7 输入装置	(59)

第九章 输出装置

§ 9-1 概述	(67)
§ 9-2 常用的伺服驱动装置	(67)
§ 9-3 输出装置	(80)

第十章 位置检测装置

§ 10-1 概述	(82)
§ 10-2 旋转变压器	(82)
§ 10-3 感应同步器	(84)
§ 10-4 光栅	(87)
§ 10-5 光电盘	(90)
§ 10-6 激光检测	(92)
§ 10-7 位置检测装置的选择	(94)

第十一章 数控机床的实例

§ 11-1 MNC数控车床	(95)
§ 11-2 CNC数控机床	(114)

第二篇 数控技术

数控技术是随着电子计算机的普及和提高而发展起来的。它是促进机械加工自动化的一种现代化手段。目前工厂所用的现代化机床多是数控机床。使用这种机床不但能提高加工精度和生产率，而且能经济、安全和自动地进行加工。

数控机床的加工适应性特别广。它既适于形状复杂的工件加工，也适于简单形状工件的加工。既能应用于单件和小批量生产。也适于大批量生产。

任何形状复杂的工件，就其形状构成来说，不外乎是由直线、圆弧或其它曲线所构成。因而，研究数控机床的加工，就是根据工件表面的形成原理，研究出控制切削刀具移动轨迹的方法，以便加工出合乎精度要求的工件。控制机床加工轨迹的问题对数控装置来说就是实施插补运算的问题。

我们知道数控机床的数控装置是由输入装置、运算器、控制器和输出装置等构成。其中运算器就担任插补运算的任务。如果使用微机来代替传统的数控装置，则其插补运算需靠软件来实现。下面详细介绍数控技术中的插补原理和输入输出等数控机床中所用的数控装置。

第七章 数控的插补原理

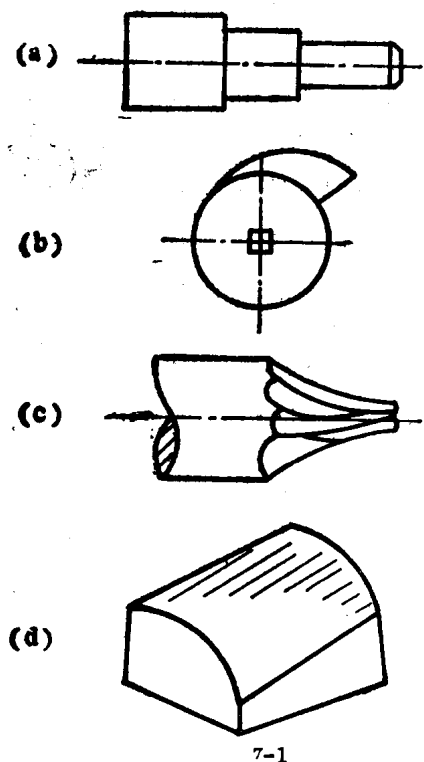
数控机床一般是用步进电机或伺服电机驱动进给机构来完成切削加工的。当加工具有直线轮廓的工件时,只沿一个坐标轴进给就能满足加工要求。例如加工图7-1(a)所示的阶梯轴就是这样。如果加工平面凸轮、手把或成形铣刀(图7-1(b)、(c)),须在两个坐标轴方向

控制进给才能加工出指定的形状。若加工图7-1(d)所示的曲面工件,应在三个坐标方向控制进给才能满足加工要求。

数控技术是靠数字脉冲来控制加工进给距离的。一个脉冲能使工作台沿一个坐标移动的距离,叫作脉冲当量。因而知道了沿一个轴线的加工长度,就可换算出加工脉冲总数。但问题是在加工曲面时,如何使数控装置沿几个坐标轴合理地分配脉冲,这就是如何进行插补运算的问题。例如只沿一个坐标轴分配脉冲就加工出直线;若沿两个坐标轴按比例分配脉冲就得出斜面;若按其它一定规律分配脉冲就可得出各种曲线轮廓的表面等。数控技术中采用的插补运算方法有很多种,最常用的方法有以下几种:

1. 数字乘法器
2. 逐点比较法
3. 数字积分器
4. 最小偏差法

下面将详细讨论前三种方法的原理和应用。



7-1

§ 7-1 数字乘法器

数字乘法器也叫作数字脉冲分频器,它是最早采用的一种简单插补运算方法。第一台数控铣床就是采用这种插补运算方法,它是控制机床完成直线和圆弧加工的最简单加工方法。数字乘法器以前是靠逻辑电路来构成数字脉冲分频器的。如果了解了硬件结构,不难用微机的软件来实现。我们知道一般的触发器,就是起着分频的作用。一个触发器能完成二分频,即每输入两个脉冲,触发器就有一个脉冲(2^0)输出。如果有 n 个触发器,显然就起着 2^{n-1} 的分频作用。用这样的分频器,再配上一些门电路,就组成了数字乘法器,如图7-2所示。

该图是由4个触发器和两个4位的 X 、 Y 寄存器,4个单稳DW及“与非”门电路构成。分频器输出的脉冲数 2^3 、 2^2 、 2^1 、 2^0 是否能从 S_x 或 S_y 输出,它取决于寄存器 X 或 Y 的状态,也就是取决于寄存器的预置的代码。因而

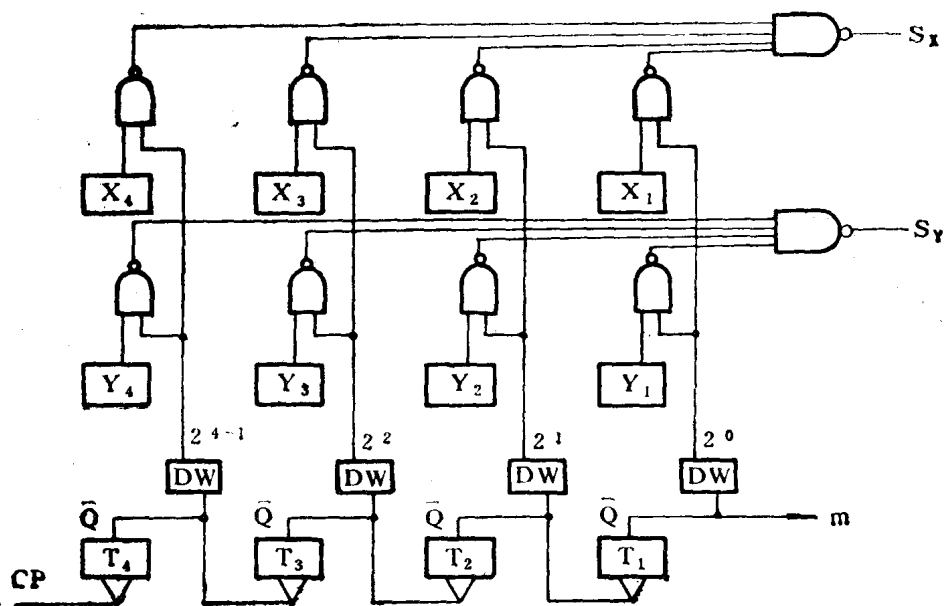


图7-2

$$S_x = X_4 \cdot 2^3 + X_3 \cdot 2^2 + X_2 \cdot 2^1 + X_1 \cdot 2^0$$

$$S_y = Y_4 \cdot 2^3 + Y_3 \cdot 2^2 + Y_2 \cdot 2^1 + Y_1 \cdot 2^0$$

这就是说，4位分频器中，当输入脉冲数为 2^4 时， T_1 端有一个溢出脉冲即 $m=1$ ；而 S_x 、 S_y 的脉冲数是由寄存器X或Y决定。显然，当有 m 个溢出脉冲时，输出总脉冲将为：

$$S = mS_x \quad \text{或} \quad S = mS_y$$

这是一乘法关系，也就是数字乘法器的由来。这一脉冲总数输出给伺服电机，就可驱动机床的工作台上的X轴和Y轴，使刀具走出斜线轨迹。

例如要使刀具走一斜线轨迹，令其斜率为 $\Delta Y / \Delta X = \frac{6}{10}$ 。这时寄存器 $Y = 0110$ ， $X = 1010$ ，在输入16个脉冲的情况下，经分频输出所得的 S_x 、 S_y 的脉冲分布，如图7-3所示。按这样的脉冲数即可走出如图7-4所示的斜线。

从图7-3可知， S_x 、 S_y 的脉冲分布是不均匀的，这将影响加工质量，因而必须将它细化。用数字乘法器这一方法很简单。而且容易实现。

用软件来实现16分频的方法比较简单，它是利用寄存器反复加上某数来完成的。实际

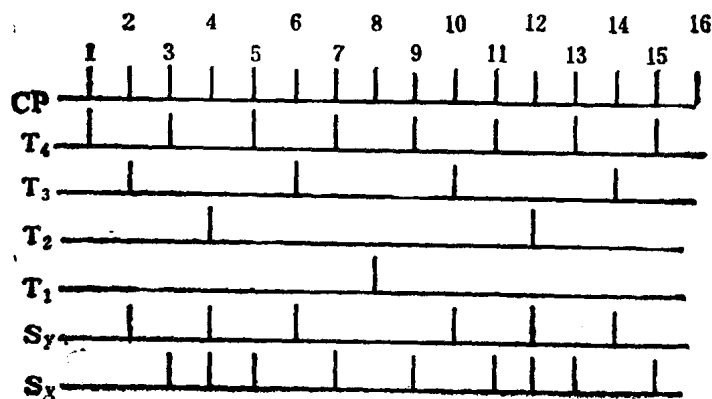


图7-3

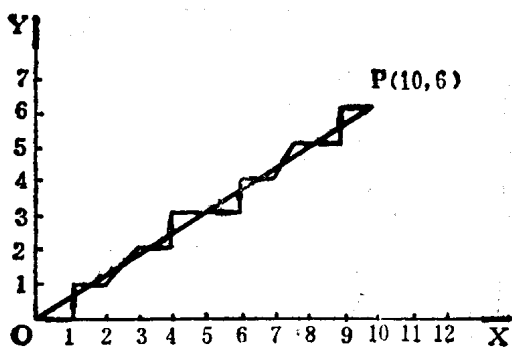


图7-4

16位寄存器的低4位,正好相当于4位计数器。如在低4位反复加上 ΔX 或 ΔY ,且进行16次,也可完成分频任务。现设在HL寄存器对中装入脉冲总数;使 $(2040H) \leftarrow \Delta X$ 、 $(2041H) \leftarrow \Delta Y$ 。每次都要作累加计算,并测低4位有无向 D_4 (上一位)进位。如有则驱动伺服电机;若没有就继续计算。这时总脉冲数作 $HL \leftarrow HL - 1$ 计算,并判别是否到终点,以决定继续加工或停止。根据这一思路可编制能作16分频的数字乘法器如下:

加工直线存程

```

2200 215505
2203 3A4020
2206 47
2207 3A4120
220A 4F
220B 110000
220E 7H AGAIN
220F 80
2210 57
2211 CB67
2213 2806
2215 CBA7
2217 57
2218 CD0024
221B 7B NEXT
221C 81
221D 5F
221E CB57
2220 2806
2222 CBA7
2224 5F
2225 CD002C
2228 2B NEXT1:
2229 7C
222A 5B
222B 20E1
222D CD002F
2230 76

```

```

ORG 2200H
LD HL, DATA ; 置总步数
LD A, (2040H) ; 置 $\Delta X$ 
LD B, A
LD A, (2041H) ; 置 $\Delta Y$ 
LD C, A
LD DE, 00H
LD A, D
ADD A, B ;  $A \leftarrow A + B$ 
LD D, A
BIT 4, A ; 测 $D_4=1?$ 
JR Z, NEXT ; 否、转置NEXT
RES 4, A ; 是 $D_4=0$ 
LD D, A
CALL MOTORX ; 置X方向走一步
LD A, E
ADD A, C ;  $A \leftarrow A + C$ 
LD E, A
BIT 4, A ; 测 $D_4=1?$ 
JR Z, NEXT1 ; 否, 转至NEXT1
RES 4, A ; 是, 使 $D_4=0$ 
LD E, A
CALL MOTORY ; 置Y方向走一步
DCE HL
LD A, H
OR L
JR NZ, AGAIN
CALL STOP ; 完, 停止
HALT

```

这样的程序不但能作16分频,还能作32、64、128、256分频。脉冲分布也比较均匀,脉冲总数也可扩大。至于驱动步进电机的子程序和停止加工子程序,可参考以后的输出装置一节中的内容,操作时将加工总步数DATA值,直接送给寄存器对HL即可。

§7-2 逐点比较法的插补运算

逐点比较法是我国数控机床中经常采用的一种插补运算方法,采用这种方法的数控机床既能加工直线轮廓、又能加工圆弧曲线轮廓。逐点比较法的特点是加工每走一步,要作一次运算,并使加工点同要求的加工轨迹相比较,以确定下一步的走向。加工时,一般按偏差判别、进给、偏差计算和终点判别等四步进行,如图7-5所示。

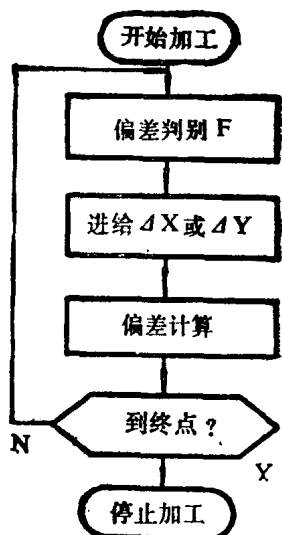


图7-5

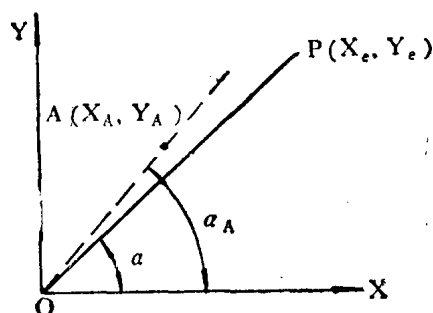


图7-6

现对加工直线轨迹和圆弧轨迹的逐点比较法介绍如下。

(一) 直线加工的插补运算

1. 偏差判别

设加工斜角为 α 的直线OP,若加工点A在OP的上边时有 $\alpha_A > \alpha$,若A在直线下边时,有 $\alpha_A < \alpha$;当 $\alpha_A = \alpha$ 时 (见图7-6)有

$\text{tg} \alpha_A = \text{tg} \alpha$, 由此得:

$$\frac{Y_A}{X_A} = \frac{Y_e}{X_e}$$

$$\text{或 } F = Y_A \cdot X_e - Y_e \cdot X_A = 0$$

上式中的F表示偏差,根据F可以判断加工点A偏离直线OP的情况。即:

$$\text{当 } F \begin{cases} > 0 \text{ 时} & \text{A点在直线的上边} \\ = 0 \text{ 时} & \text{A点在直线上} \\ < 0 \text{ 时} & \text{A点在直线的下边} \end{cases}$$

2. 进给

知道偏差F就可以决定加工的进给方向,例如当加工的一个点A在OP线的上边时,为了使其加工时不偏离直线太远,它应向X方向走一步即进给为 $\Delta X + 1$ (图7-7)。而在到达A点后,如再进给应是Y方向,即进给为 $\Delta Y + 1$ 。也就是当加工点位置已知时,根据偏差F就可决定进给方向。即

$F \geq 0$ 时,沿X方向的进给为 $\Delta X \leftarrow \Delta X + 1$; $F < 0$ 时,沿Y方向的进给为 $\Delta Y \leftarrow \Delta Y + 1$ 。

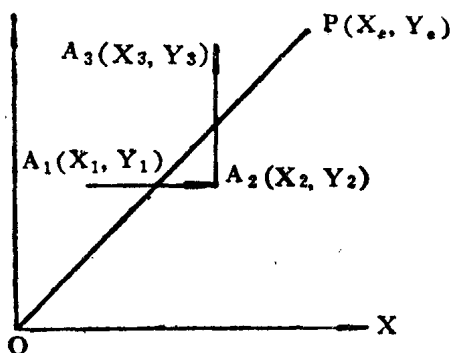


图7-7

3. 偏差计算

加工时每走一步要作一次偏差计算。由此得出F后,再确定进给方向。为了插补运算方便,偏差计算可用下述方法导出的简便公式进行。

设OP直线的终点坐标为 X_e, Y_e ; 点 A_1 的坐标为 X_1, Y_1 , 由此可计算出 A_1 点的偏差:

$$F_1 = Y_1 X_e - Y_e X_1 \quad (7-1)$$

如果 $F_1 > 0$, 进给应是向X方向走一步到达 A_2 点。这时 A_2 的坐标为 $X_2 = X_1 + 1, Y_2 = Y_1$ 。因而 A_2 点的偏差为:

$$\begin{aligned} F_2 &= Y_2 X_e - Y_e X_2 \\ &= Y_1 X_e - Y_e (X_1 + 1) \\ &= (Y_1 X_e - Y_e X_1) - Y_e \\ &= F_1 - Y_e \end{aligned} \quad (7-2)$$

这时因为 $F_2 < 0$ (A_2 点在直线下方), 加工时应向Y方向进给。因而可求得 A_3 点的偏差如下:

$$\begin{aligned} F_3 &= Y_3 X_e - Y_e X_3 \\ &= (Y_2 + 1) X_e - Y_e X_2 \\ &= (Y_2 X_e - Y_e X_2) + X_e \\ &= F_2 + X_e \end{aligned} \quad (7-3)$$

由以上公式可得出结论: 若直线的终点坐标 X_e, Y_e 已知, 加工点的偏差由F加上 $\pm X_e$ 或 $\pm Y_e$ 求得。进给的方向可根据F值的正负确定。

表7-1 L_1 直线的偏差计算

偏差符号	$F \geq 0$	$F < 0$
偏差计算	$F \leftarrow F - Y_e$	$F \leftarrow F + X_e$
进给	$+\Delta X$	$+\Delta Y$

根据以上的结论, 可归纳出处于第1象限的直线 L_1 的加工计算公式和进给方向如表7-1所示。基于这样的方法不难推出第2、3、4象限的直线偏差计算的公式。这如图7-8和表7-2所示。

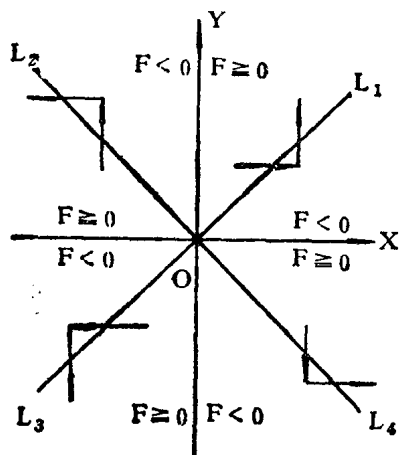


图7-8

表7-2 直线加工的偏差计算

线型	$F \geq 0$		$F < 0$	
	进给	偏差计算	进给	偏差计算
L_1	$+\Delta X$	$F \leftarrow F - Y_e$	$+\Delta Y$	$F \leftarrow F + X_e$
L_2	$-\Delta X$	$F \leftarrow F - Y_e$	$-\Delta Y$	$F \leftarrow F + X_e$
L_3	$+\Delta Y$	$F \leftarrow F - X_e$	$-\Delta X$	$F \leftarrow F + Y_e$
L_4	$-\Delta Y$	$F \leftarrow F - X_e$	$+\Delta X$	$F \leftarrow F + Y_e$

由上可知：逐点计算偏差的方法，可把 $F = Y_A \cdot X - Y \cdot X_A$ 的运算公式化为 $F \leftarrow F \pm X$ 或 $F \leftarrow F \pm Y$ 的简单公式，它能简化逻辑电路。当然对微机来说，也能简化加工程序。只要根据表7-2，对不同象限的直线加工，采用不同的计算公式和进给就可以了。

4. 终点判别

最后谈一下加工直线的终点判别。逐点比较法是每走一步、作一次计算和判断。因而，如控制X向或Y向的总步数，就能判别出是否到达终点。问题是到底采用X方向计数 G_x ，还是采用Y方向计数 G_y ？，一般是以 45° 为界，可按图7-9确定。也就是说，对 L_1 来说如 $\alpha < 45^\circ$ ，应采用X方向的 $\angle X$ 的总步数 J_x 来控制终点。如 $\alpha > 45^\circ$ 则用Y方向控制总步数 J_y ，以此判断加工是否到达终点。

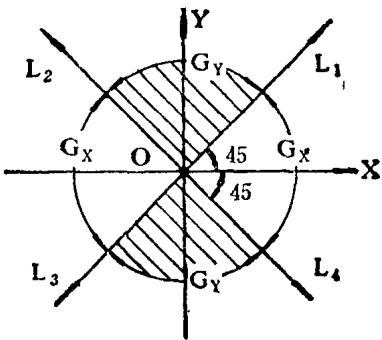


图7-9

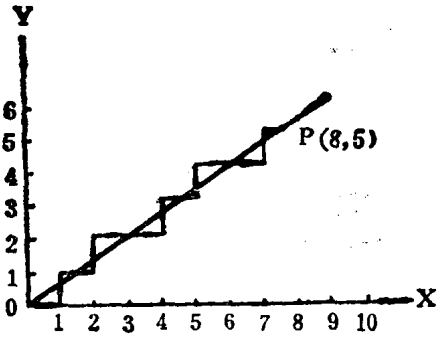


图7-10

(二) 直线加工偏差计算的举例

例如，加工直线 L_1 ，其终点坐标如图7-10所示。这时终点判别的计数方向为 G_x 。设 $\angle X$ 的总脉冲为8。它每走一步就减1，当为零时就表示到了终点。现将这一直线的插补运算列表如下：

表7-3

序 号	偏 差 判 别	进 给	偏 差 计 算 $F \leftarrow F - Y$ 或 $F \leftarrow F + X$	终 点 判 别
1	$F = 0$	$+\Delta X$	$F = 0 - 5 = -5, \Delta X + 1$	$8 - 1 = 7 \neq 0$
2	$F = -5$	$+\Delta Y$	$F = -5 + 8 = 3, \Delta Y + 1$	$7 \neq 0$
3	$F = 3$	$+\Delta X$	$F = 3 - 5 = -2, \Delta X + 1$	$7 - 1 = 6 \neq 0$
4	$F = -2$	$+\Delta Y$	$F = -2 + 8 = 6, \Delta Y + 1$	$6 \neq 0$
5	$F = 6$	$+\Delta X$	$F = 6 - 5 = 1, \Delta X + 1$	$6 - 1 = 5 \neq 0$
6	$F = 1$	$+\Delta X$	$F = 1 - 5 = -4, \Delta X + 1$	$5 - 1 = 4 \neq 0$
7	$F = -4$	$+\Delta Y$	$F = -4 + 8 = 4, \Delta Y + 1$	$4 \neq 0$
8	$F = 4$	$+\Delta X$	$F = 4 - 5 = -1, \Delta X + 1$	$4 - 1 = 3 \neq 0$
9	$F = -1$	$+\Delta Y$	$F = -1 + 8 = 7, \Delta Y + 1$	$3 \neq 0$
10	$F = 7$	$+\Delta X$	$F = 7 - 5 = 2, \Delta X + 1$	$3 - 1 = 2 \neq 0$
11	$F = 2$	$+\Delta X$	$F = 2 - 5 = -3, \Delta X + 1$	$2 - 1 = 1 \neq 0$
12	$F = -3$	$+\Delta Y$	$F = -3 + 8 = 5, \Delta Y + 1$	$1 \neq 0$
13	$F = 5$	$+\Delta X$	$F = 5 - 5 = 0, \Delta X + 1$	$1 - 1 = 0$
				终 点

(三) 直线加工的程序设计

逐点比较法的直线加工，过去都用中小规模集成电路组成的运算器和控制器来实现其直

线加工的插补运算。这在一般数控书中都有详细介绍。但这种逻辑电路式的插补器的结构复杂、需用的逻辑元件较多、造价高、可靠性差，造成数控机床不易被工厂采用。因而目前都在利用微机来实现逐点比较法的直线加工插补计算。但在专业书中还没有这方面的介绍。为此，本章力图结合前面讲的插补原理，详细介绍这种插补计算程序设计的步骤和方法。

由表7-2和图7-8可知：直线插补计算是和加工偏差 F 、线型 $L_1 \sim L_4$ 、计数方向及进给等参数有关的。同时它要根据 F 和线型的不同，选择不同插补计算公式进行计算。因而按下述步骤，就可画出程序框图（图7-11），并能编出直线加工的程序。

1. 向微机键入直线加工的数据。

(2800H)和(2801H) $\leftarrow$ X值（终点坐标）

(2802H)和(2803H) $\leftarrow$ Y值（终点坐标）

(2804H)和(2805H) $\leftarrow$ J值（计数长度）

(2806H)高4位 $\leftarrow G_x$ 或 G_y ，用0代表 G_x ，1代表 G_y

(2806H)低4位 $\leftarrow L_1 \sim L_4$ 。分别用01、02、03、04代表。

(2807H)和(2808H) $\leftarrow F$ 值（加工偏差）

(280AH)和(280BH) $\leftarrow$ X或Y的补码

2. 进行程序初始化，先使 $F \leftarrow 0$

设(2809H)的 $D_0 = 0$ 表示 G_x (2809H)的 $D_1 = 0$ 表示线型 L_1 和 L_2

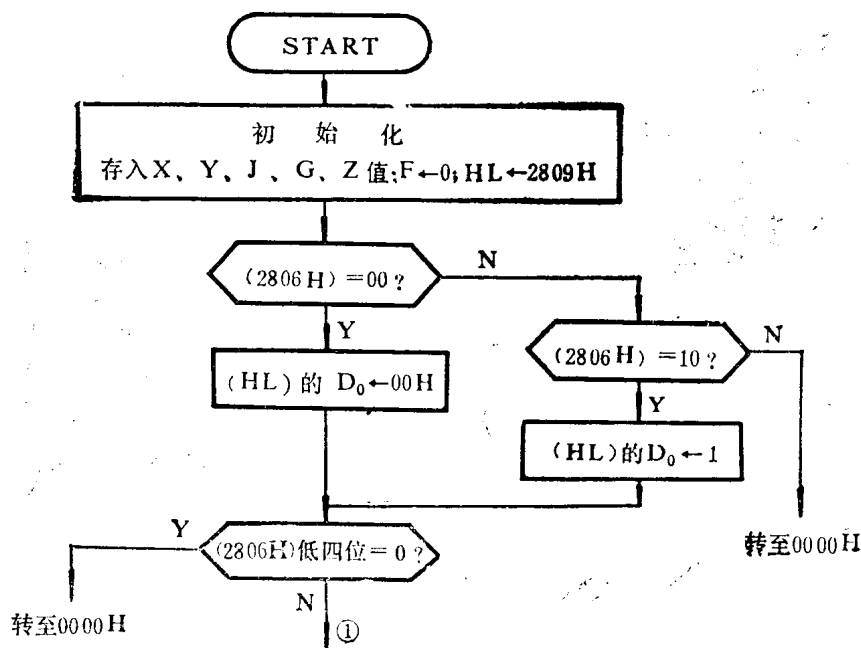
(2809H)的 $D_0 = 1$ 表示 G_y (2809H)的 $D_1 = 1$ 表示线型 L_3 和 L_4

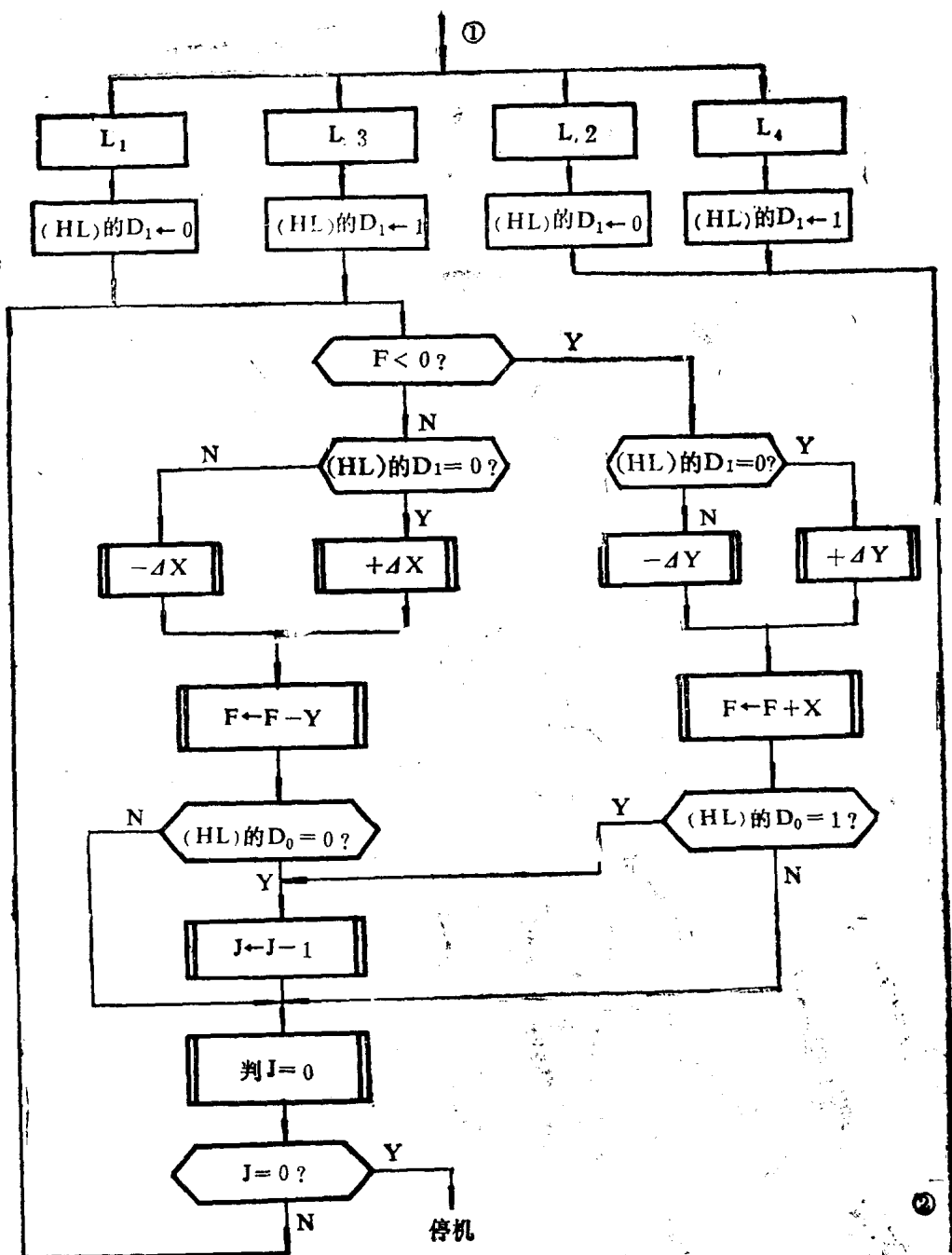
3. 由程序区分出线型 $L_1 \sim L_4$ 。根据 F 的不同选择不同的插补公式和进给 ΔX 或 ΔY 进行加工。

4. 计数长度 J 是根据 G_x 或 G_y 的不同，分别存贮的直线终点坐标 X 、或 Y 值。当 $J-1=0$ 时，表示已到终点加工完了。

5. 偏差计算，即 $F \leftarrow F - X$ 或 $F \leftarrow F - Y$ 的减法计算。本程序采用了十进制数的补码运算。

直线加工插补计算框图





本例附有这一补码换算的子程序 Z_{sx} 和 Z_{sy} 。

6. 进给 ΔX 或 ΔY 是用作电机驱动。这应另编有步进电机驱动子程序，本例中未列出。可参看以后的章节。

7. 偏差计算都有专用的子程序。因 F 存入2807H、2808H单元，用(2808H)的 D_7 可判断 F 的正负，即当 $D_7 = 0$ 时，说明 $F \geq 0$ ， $D_7 = 1$ 时、 $F < 0$ 。

根据以上的分析要点可画出逐点比较法的插补计算程序框图如图7-11所示。

下面是根据图7-11的直线插补程序框图编制的逐点比较法加工程序。

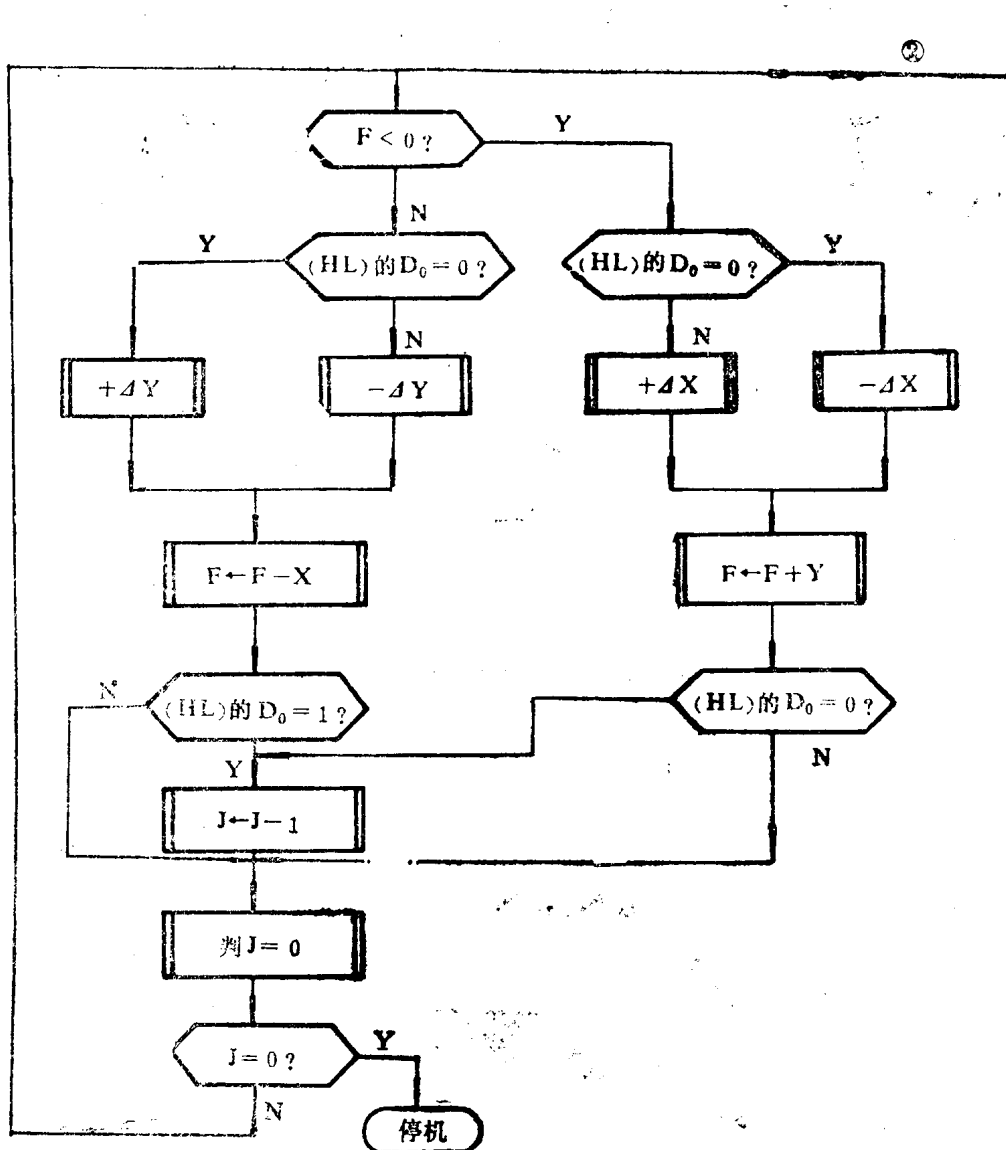


图7-11

直线加工插补计算程序

		ORG	2000H	
2000	21000	LD	HL, 0000H	, F置0
2003	220728	LD	(2807H), HL	
2006	210928	LD	HL, 2809H	, 确定Z、G存储单元
2009	3A0628	LD	A, (2806H)	, 取G及L
200C	47	LD	B, A	
200D	E6F0	AND	F0H	, 判G _x 或G _y
200F	2004	JR	NZ, LOOP2	, 是G _y 转至LOOP2
2011	77	LD	(HL), A	, 是G _x 时 (HL) ← 0
2012	78 LOOP1:	LD	A, B	
2013	180B	JR	LOOP4	

2015	FE10	LOOP2,	CP	10H	
2017	2803		JR	Z , LOOP3	
2019	C30000	AGAIN,	JR	0000H	
201C	CBC6	LOOP3,	SET	0 , (HL)	
201E	18F2		JR	LOOP1	
202D	E60F	LOOP4,	AND	0F	, 判加工指令Z
2022	FE01		CP	01H	
2024	280E		JR	Z, LOOP5	, 是L ₁ 转至LOOP5
2026	FE02		CP	02H	
2028	280E		JRZ	LOOP6	, 是L ₂ 转至LOOP6
202A	FE03		CP	03H	
202C	280E		JR	Z , LOOP7	, 是L ₃ 转至LOOP7
202E	FE04		CP	04H	
2030	280F		JR	Z , LOOP8	, 是L ₄ 转至LOOP8
2032	18E5		JR	AGAIN	, 不是, 则返回
2034	CB8E	LOOP5,	RES	1, (HL)	, (HL) 的D ₁ 置0
2036	180C		JR	LOOP9	
2038	CB8E	LOOP6,	RES	1, (HL)	, (HL) 的D ₁ 置0
203A	184C		JR	LOOP10	
203C	CBCE	LOOP7,	SET	1, (HL)	, (HL) 的D ₁ 置1
203E	1804		JR	LOOP9	
204D	CBCE	LOOP8,	SET	1 , (HL)	, (HL) 的D ₁ 置1
2042	1844		JR	LOOP10	
2044	3A0828	LOOP9,	LD	A, (2808H)	, 判F
2047	CB7F		BIT	7 , A	
2049	2022		JR	NZ , QA4	
204B	3A0928		LD	A, (2809H)	
204E	CB4F		BIT	1 , A	
2050	2805		JR	Z , QA0	
2052	CD002B		CALL	FEED2	, 进给 - ΔX
2055	1803		JR	QA1	
2057	CD002A	QA0,	CALL	FEED1	, 进给 + ΔX
205A	CD0321	QA1,	CALL	FOML1	, 插补 $F \leftarrow F - Y$
205D	3A0928		LD	A , (2809H)	
2060	CB47		BIT	0 , A	
2062	2003		JR	NZ , QA3	,
2064	CD0821	QA2,	CALL	TERM1	, 计数 $J \leftarrow J - 1$
2067	CD1721	QA3,	CALL	TERM2	, 判 $J = 0$
206A	20D8		JR	NZ , LOOP9	
206C	76		HALT		
206D	3A0928	QA4,	LD	A , (2809H)	
2070	CB4F		BIT	1 , A	
2072	2805		JR	Z , QA5	
2074	CD002C		CALL	FEED4	进给 - ΔY

2077	1803	JR	QA6	
2079	CD002D QA5;	CALL	FEED3	, 进给 + ΔY
207C	CDE420 QA6;	CALL	FOML 2	, 插补 $F \leftarrow F + X$
207F	3A0928	LD	A , (2809H)	
2082	CB47	BIT	0 , A	
2084	20DE	JR	NZ , QA2	
2086	18DF	JR	QA3	
2088	3A0828 LOOP10;	LD	A , (2808H)	
208B	CB7F	BIT	7 , A	
208D	2022	JR	NZ , QB4	
208F	3A0928	LD	A , (2809H)	
2092	CB4F	BIT	1 , A	
2094	2805	JR	Z , QB0	
2096	CD002C	CALL	FEED4	, 进给 - ΔY
2099	1803	JR	QB1	
209B	CD002D QB0;	CALL	FEED3	, 进给 + ΔY
209E	CDF6220 QB1;	CALL	FOML3	, 插补 $F \leftarrow F - X$
20A1	3A0928	LD	A , (2809H)	
20A4	CB47	BIT	0 , A	
20A6	2803	JR	Z , QB3	
20A8	CD0821 QB2;	CALL	TERML1	, 计数 $J \leftarrow J - 1$
20AB	CD1721 QB3;	CALL	TERML2	, 判 $J = 0$
20AE	20D8	JR	NZ , LOOP10	
20B0	76	HALT		
20B1	3A0928 QB4;	LD	A , (2809H)	
20B4	CB4F	BIT	1 , A	
20B6	2805	JR	Z , QB5	
20B8	CD002A	CALL	FEED1	, 进给 + ΔX
20BB	1803	JR	QB6	
20BD	CD002B QB5;	CALL	FEED2	, 进给 - ΔX
20C0	CDFE20 QB6;	CALL	FOML4	, 插补 $F \leftarrow F + Y$
20C3	3A0928	LD	A , (2809H)	
20C6	CB47	BIT	0 , A	
20C8	28DE	JR	, Z, QB2	
20CA	18DF	JR	QB3	
		X变补子程序		
20CC	210028 ZSX;	LD	HL , 2800H	, 取X值
20CF	110A28 ZSXY1;	LD	DE , 280AH	, 设存补码单元
20D2	AF	XOR	A	
20D3	0602	LD	B , 02H	
20D5	3E00	LD	A , 00H	
20D7	9E ZSXY2;	SBC	(HL)	, 变补
20D8	27	DAA		
20D9	12	LD	(DE) , A	, 存补码

20DA	23		INC HL	
20DB	13		INC DE	
20DC	10F7		DJ NZ ZSXY 2	
20DE	C9		RET	
			Y变补子程序	
20DF	210228	ZSY,	LD HL , 2802H	, 取Y值
2DE2	18EB		JR ZSXY1	
			F←F + X子程序	
20E4	210028	FOML2,	LD HL , 2800H	, 取X
20E7	110728	MOL2,	LD DE , 2807H	, 取F
20EA	AF		XOR A	
2 EB	0602		LD B , 02H	
20ED	1A	MOL3,	LD A , (DE)	
20EE	8E		ADC (HL)	, F←F + X
20EF	27		DAA	
20F0	12		LD (DE) , A	
20F1	13		INC DE	
20F2	23		INC HL	
20F3	10F8		DJ NZ , MOL3	
20F5	C9		RET	
			F←F - X子程序	
20F6	CDCC20	FOML3,	CALL ZSX	, 调X变补子程序
20F9	210A28	MOL4,	LD HL , 280AH	, 存补码单元
20FC	18E9		JR MOL2	, 转至MOL2
			F←F + Y 子程序	
20FE	210228	FOMLZ,	LD HL 2802H	, 取Y
2101	18E4		JR MOL2	, 转至MOL2
			F←F - Y 子程序	
2103	CDDF20	FOML1,	CALL ZSXY	, 调Y变补子程序
2106	18F1		JR MOL4	, 转至MOL4
			J←J - 1 子程序	
2108	210428	TERM1,	LD HL , 2804H	, 设计数单元
210B	37		SCF	
210C	0602		LD B 02H	
210E	7E	MOL5,	LD A , (HL)	
210F	DE00		SBC 00H	, 带进位位相减
2111	27		DAA	
2112	77		LD (HL) , A	
2113	23		INC HL	
2114	10F8		DJN2 MOL5	
2116	C9		RET	
			判J = 0 子程序	
2117	210428	TERM 2,	LD HL, 2804H	, 设计数单元
211A	AF		XOR A	