

TP-801单板机控制线切割机的程序设计

刘人彦 董明亮 张新学

(电 机 系)

提 要

本文系统介绍对采用相对座标系统“三B”语言编程的线切割机和采用绝对座标系统的加工指令的线切割机,实现Z80—TP801单板机控制,讨论了用Z80汇编语言编制控制程序的设计思想、程序框图及各种控制功能程序的调试方法等。

数控线切割机已广泛地用于模具加工,而且由于采用数字控制,提高了模具的加工精度和质量。但原有的线切割机控制部分多采用分立元件、中小规模的集成电路,线路复杂、可靠性低、成本高、维修困难,操作工人掌握不便。为了对原有采用相对座标系统“三B”语言指令格式的线切割机及新式采用绝对座标系统的加工指令的线切割机用TP801单板机进行控制,用软件实现逻辑功能,我们进行了用Z80汇编语言全部控制程序的设计及调试工作,初步设计出了整套适合两种控制的程序。该程序便于对原有数控线切割机的改造,并扩大加工功能。现将主要设计思想、程序框图及控制功能作概括介绍。

一、“三B”语言编程线切割机控制软件系统

采用传统“三B”语言格式编程的线切割机,编出了整套控制程序,写入EPROM中,用TP801单板机取代原有的运算控制柜,简单易行。原采用的逐点比较法插补原理、插补公式、计数方向、终点判别方法等均保持不变。在加工过程中单板机能正确选择偏差计算公式。加工指令中要体现加工曲线是圆弧或直线,所在象限,顺逆情况等;每条加工指令由4个字组成:LG(加工方式,计数方向),加工方式12种,计数方向2种。X、Y座标值,对圆弧式是以圆心为座标原点的起点座标,对直线是以起点为座标原点的终点座标。J进给步数,END·D停机,以上各用一个代码表示,加工指令的有关内容相应送入用户定义键2FBA、2FBC、2FBE、2FB8,用户可以很方便的启动加工程序。

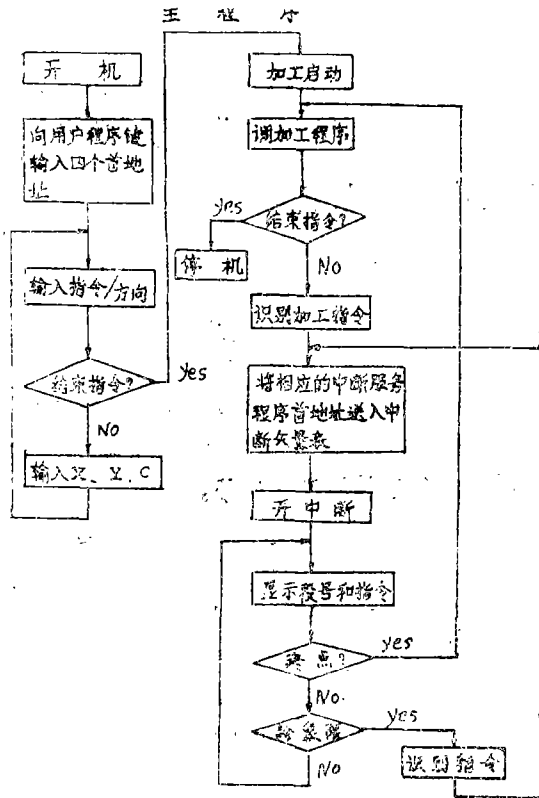
程序框图包括主程序,控制功能,执行一条加工指令的准备工作,显示序号,加工公式,判断终点后转入执行下条加工指令,调用有关子程序。

子程序包括:显示子程序(显示加工序号,加工方式),插补计算子程序,判终点子程序,输入输出控制子程序,指针赋值子程序等。

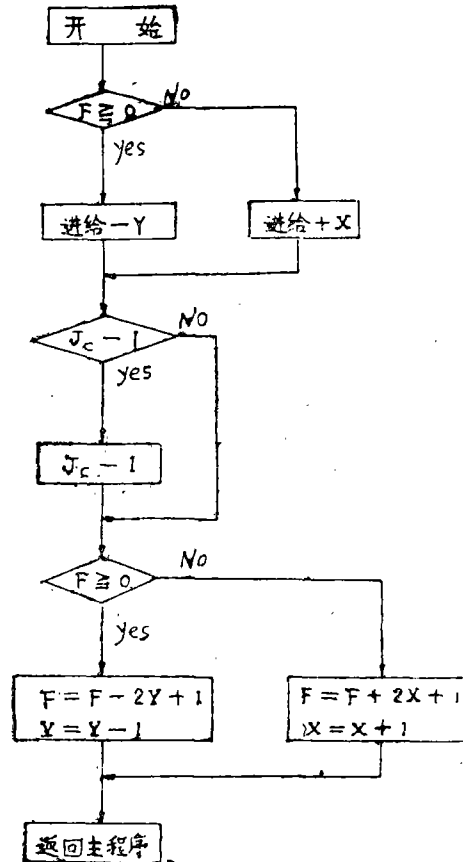
初始化程序,CTC初始化设定为定时器工作方法,PIO初始化,设定A口, B口, 为输出方式。

程序框图举例。如图一所示

本文1984年6月15日收到



图一



图二

顺园 1 (SR1) 中断服务程序流程图如图二所示。

二、采用绝对坐标系统的控制程序设计

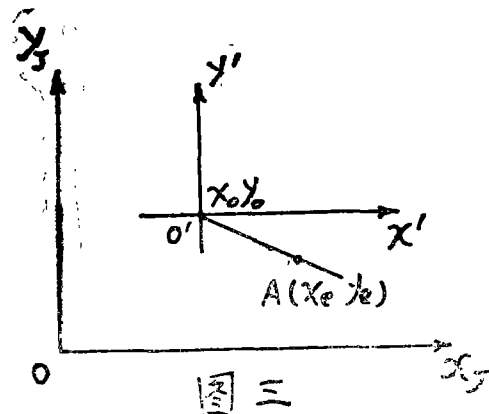
为了克服“三B”语言存在的缺点，如计算线段交点坐标，加工程序编制困难，加工线段存在积累误差等，我们根据北工大沈以清老师为克服上述缺点提出采用绝对坐标系统和新的终点判别准则，进行了系统软件设计工作。

座标变换，由相对座标系统变成绝对座标系统，必须进行插补公式的座标变换。

相对座标系 $X'O'Y'$

绝对座标系 X_0Y_0

座标变换是座标轴平移后，加工线 $O'A$ 上任意点 A 在 X_0Y_0 中的座标为 (X_0, Y_0) ， A 点在相对座标系为 $(X_0 - X_0, Y_0 - Y_0)$ ，插补公式中原式



图三

$$F_{i+1,j} = F_{i,j} - Y_e$$

$$\text{变成 } F_{i+1,j} = F_{i,j} - (Y_e - Y_0)$$

座标变换后的插补计算公式如附表所示。

修改后的插补计算公式: $F_{i,j}$ 为进给偏差

附表:

加工指令	$F_{i,j} \geq 0$			$F_{i,j} < 0$		
	进给	座标	插补公式	进给	座标	插补公式
SR ₁	-y	y _j -1	$F_{ij} - 2(y_j - y_0) + 1$	+x	x _j +1	$F_{ij} + 2(x_j - x_0) + 1$
SR ₂	+x	x _j -1	$F_{ij} + 2(x_j - x_0) + 1$	+y	y _j +1	$F_{ij} + 2(y_j - y_0) + 1$
SR ₃	+y	y _j +1	$F_{ij} + 2(y_j - y_0) + 1$	-x	x _j -1	$F_{ij} - 2(x_j - x_0) + 1$
SR ₄	-x	x _j -1	$F_{ij} - 2(x_j - x_0) + 1$	-y	y _j -1	$F_{ij} - 2(y_j - y_0) + 1$
NR ₁	-x	x _j -1	$F_{ij} - 2(x_j - x_0) + 1$	+y	y _j +1	$F_{ij} + 2(y_j - y_0) + 1$
NR ₂	-y	y _j -1	$F_{ij} - 2(y_j - y_0) + 1$	-x	x _j -1	$F_{ij} - 2(x_j - x_0) + 1$
NR ₃	+x	x _j -1	$F_{ij} + 2(x_j - x_0) + 1$	-y	y _j -1	$F_{ij} - 2(y_j - y_0) + 1$
NR ₄	+y	y _j +1	$F_{ij} + 2(y_j - y_0) + 1$	+x	x _j +1	$F_{ij} + 2(x_j - x_0) + 1$
L ₁	+x	x _j +1	$F_{ij} - (y_e - y_0)$	+y	y _j +1	$F_{ij} + (x_e - x_0)$
L ₂	+y	y _j +1	$F_{ij} + (x_e - x_0)$	-x	x _j -1	$F_{ij} + (y_e - y_0)$
L ₃	-x	x _j -1	$F_{ij} + (y_e - y_0)$	-y	y _j -1	$F_{ij} - (x_e - x_0)$
L ₄	-y	y _j -1	$F_{ij} - (x_e - x_0)$	+x	x _j +1	$F_{ij} - (y_e - y_0)$

终点判别,直线与圆弧相交交点有一个或两个,每进给一步计算一下直线上点到园心的距离,等于半径时说明为第一个交点,大于半径点落园外,小于半径点落在园内,第二次等于半径时,到达第二个交点。直线与直线相交,圆弧与直线相交,则用进给点到直线的距离来判交点,该距离为零时,即到达交点。如图四所示:



图四

采用交点 1 或 2 由实际加工图形而定,加工时利用设置标志位来区别。

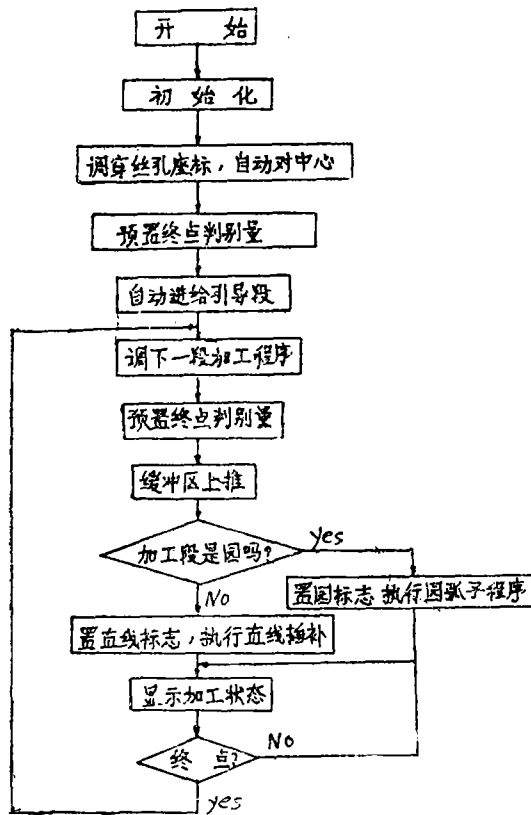
根据控制的算法及终点判别方法来编制控制程序,各项基本功能都由子程序进行,如插补迭代计算,穿丝孔定位,短路回退、驱动输出,显示断丝处理等。主程序中则设置各种判断标志,调用子程序的执行,把子程序按加工要求组织在一起,以完成应有的加工。系统控制原理:线切割是一种电蚀加工设备,它是在工件与钼丝之间加一高频电源,通过高频脉冲放电对金属的电蚀作用,对金属工件进行切割加工的。为了保证加工质量应维持钼丝与工件之间的放电间隙适当。变频电路可以用所发脉冲频率的高低来反映放电隙的大小。我们把变频电路的脉冲输出作为单板机的中断请求信号,当放电间隙增大时,变频电路就向计算机发出中断请求,要求计算机发出进给脉冲缩小放电间隙。计算机接到中断请求后就立即响应,根据用户加工图形的要求进行插补计算,根据计算结果决定向X方向或向Y方向的步进电机发进给脉冲,使钼丝与工件之间的间隙变小,这样往复下去就可以加工成一个完整的闭合曲线。

在加工工件时要求用户按切割指令的格式,编出加工图形的程序,并输入计算机内存,然后运行控制主程序,主程序框图为图五。

控制主程序:首先对I/O口及工作单元进行初始化。为了提高加工精度进行自动对中心,使钼丝位于穿丝孔的中心位置,然后调用户切割指令,分析切割指令,根据不同的指令,置不同状态标志,作为插补运算,发进给脉冲及终点判别的依据。然后开中断进行显示等待,一旦变频电路有中断请求信号,则CPU就暂停显示转向中断服务程序进行插补运算,向步进电机发进给脉冲,作终点判别等,中断服务结束再返回主程序,进行显示,判终点标置。若本加工段已加工完毕,则返回再调下一段切割指令,若没有加工完则继续显示等待。显示程序是利用单板机上现有的六位显示器,显示出当前加工段号,直线或圆弧与下条线段几次交,加工点所在象限等。

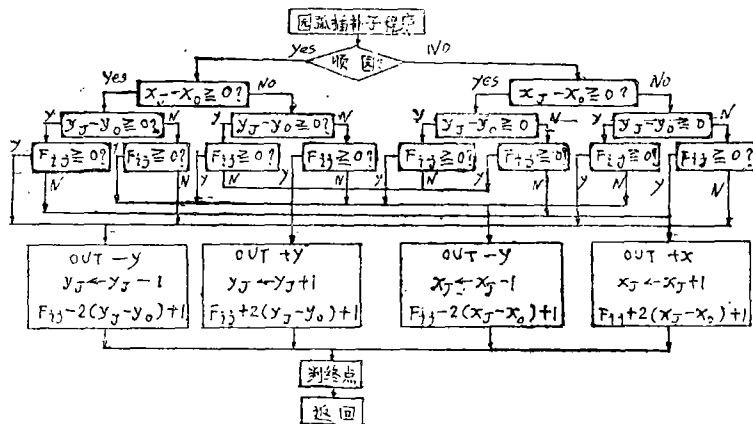
插补计算子程序框图举例如图六、图七所示。

系统主程序的简化框图



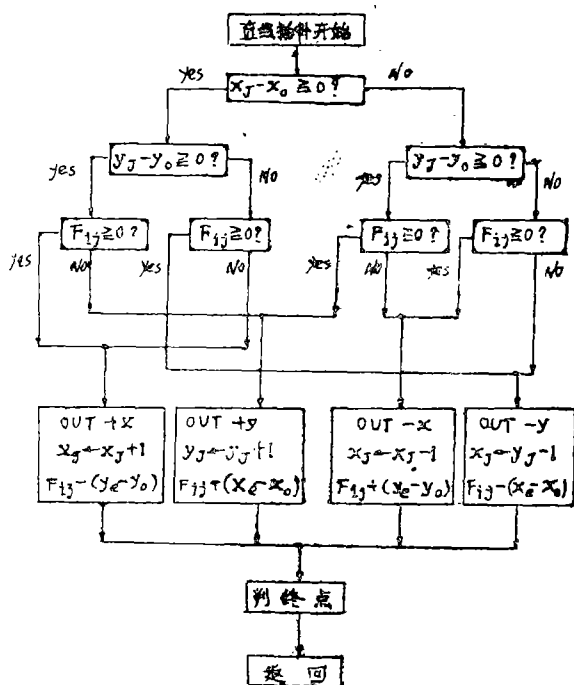
图五

圆弧插补子程序



图六

直线插补子程序



图七

三、加工指令

加工指令格式

操作码	数据
-----	----

具体规定如下表:

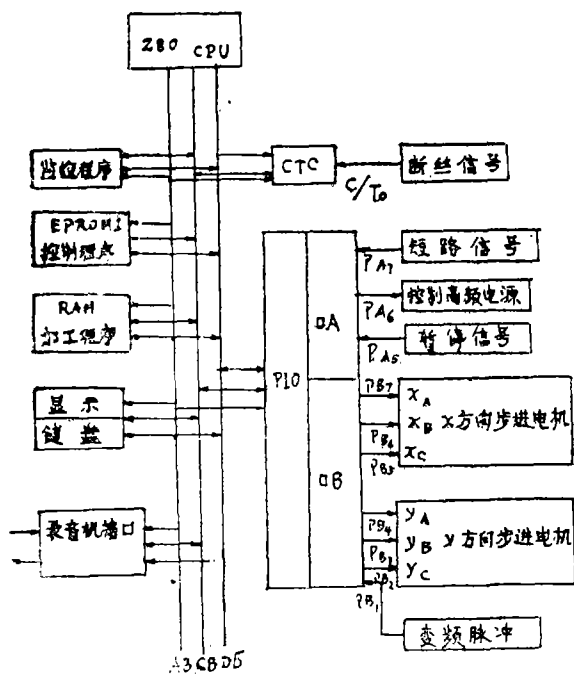
加工指令	操作码	加工指令	操作码
直线一次交	11	逆园一次交	41
直线二次交	12	逆园二次交	42
顺园一次交	21	穿丝孔定位	80
顺园二次交	22	结 束	84

直线加工指令数据为直线或其延长线上任意两点座标, 园弧加工指令数据为园心座标, 半径R, 有园弧过渡时只给出半径R。数据均由十进制数输入, 由十翻二子程序转为二进制。

四、系统硬件连接简图

单板机通过PIO口与机床联接, A口B口均置成位控方式, B口的PB₂~PB₄通过Y方向步进电机的驱动电路分别接步进电机的A、B、C三相, PB₅~PB₇接X方向步进电机的A、B、C三相, 用软件代替原来的环形分配器、A口接三个系统控制信号, 断丝信号接CTC的通道, 用计数方式向CPU发中断。全部控制程序设计过程都在TP801单板机上及模拟实验电路进行了调试, 并对加工图形进行加工曲线的切割模拟实验, 控制程序工作效果良好, 加工精度控制在1μ以内。以上工作使我们初步掌握了微机控制程序设计与调式方法。

硬件框图如图八所示



图八