

第 1 章 数控机床概述

数控机床与普通机床相比,不仅具有零件加工精度高、生产效率高、产品质量稳定、自动化程度极高的特点,而且它还可以完成普通机床难以完成或根本不能加工的复杂曲面的零件加工。因此,数控机床被广泛地应用于制造业,极大地推动了社会生产力的发展,是制造业实现自动化、网络化、柔性化、集成化的基础。

1.1 数控机床的产生与发展

1.1.1 数控机床概念

数字控制(Numerical Control)技术,简称为数控(NC)技术,是指用数字指令控制机器动作的一门技术。采用数控技术的控制系统称为数控系统。数控技术是近代发展起来的一种自动控制技术。数字控制是相对于模拟控制而言的,数字控制系统中的控制信息是数字量,而模拟控制系统中的控制信息是模拟量。

数字控制系统有如下特点:

(1)可用不同的字长表示不同的精度信息,表达信息准确。

(2)可以进行逻辑运算、数学运算,也可以进行复杂的信息处理。

(3)由于有逻辑处理功能,可以根据不同的指令进行不同方式的信息处理,从而可以用软件来改变信息处理的方式或过程,而不用改动电路或机械结构,因而具有柔性化。

由于数字控制系统具有上述优点,故被广泛应用于机械运动的轨迹控制。轨迹控制是机床控制系统和工业机器人的主要控制内容。此外,数字控制系统的逻辑处理功能可以方便地用于机械系统的开关量控制。

数字控制系统的硬件基础是数字逻辑电路。最初的数控系统是数字逻辑电路构成的,因而被称为硬件数控系统。随着微型计算机的发展,硬件数控系统已逐渐被淘汰,取而代之的是采用通用计算机硬件结构,用控制软件来实现数控功能的数控系统,称为计算机数控系统(Computer Numerical Control),简称 CNC。由于计算机可以完全由软件来确定数字信息的处理过程,从而具有真正的“柔性”,可以处理硬件逻辑电路难以处理的复杂信息。使数字控制系统的性能大大提高。当前微机技术发展很快,性价比不断提高,所以微机在数字控制系统中得到广泛的应用。

总而言之,用数字化信息进行控制的自动控制技术称为数字控制技术。装备了数控系统或采用数控技术来控制机床的运动或加工过程的机床,称为数控机床。

1.1.2 数控机床的产生与发展过程

随着科学技术和社会生产力的迅速发展,机械产品日趋复杂,社会对机械产品的质量和

生产率提出了越来越高的要求。在航空航天、造船、汽车、军事计算机等工业中使用的零件精度高、形状复杂、批量小、经常改动、加工困难、生产效率低、劳动强度大、质量难以保证。为了解决上述问题,一种灵活、通用、高精度、高效率的“柔性”自动化设备——数控机床应运而生。

数控机床的数控系统已先后经历了两个阶段、六个时代的发展。第一阶段称为硬件连接数控(NC系统),包括电子管、晶体管、集成电路,其特点是具有很多硬件电路和连接结点,电路复杂,可靠性不好。第二阶段称作计算机软件系统(CNC系统),包括小型计算机、微处理器及其与PC适用的CNC系统,最突出的特点是利用存储器里的软件控制系统工作,这种系统容易扩展功能,柔性好,可靠性高。现在,一种开放式的数控系统(ONC系统)正得到快速发展和应用。

数控机床是在机械制造技术和控制技术的基础上发展起来的,其过程大致如下:

1948年,美国帕森斯公司接受美国空军的委托,研制直升飞机螺旋桨叶片轮廓检验用的样板加工设备。由于样板形状复杂多样,精度要求高,一般加工设备难以适应,于是提出采用数字脉冲控制机床的设想。

1949年,帕森斯公司与美国麻省理工学院(MIT)开始共同研究,并于1952年试制成功第一台三坐标数控铣床,当时的数控装置采用电子管元件。

1959年,数控装置采用了晶体管元件和印刷电路板,出现带自动换刀装置的数控机床,称为加工中心(Machining Center,简称MC),使数控装置进入了第二代。

1965年,出现了第三代的集成电路数控装置,不仅体积小,功率消耗少,且可靠性提高,价格进一步下降,促进了数控机床品种和产量的发展。

20世纪60年代末,先后出现了由一台计算机直接控制多台机床的直接数控系统(简称DNC),又称群控系统;采用小型计算机控制的计算机数控系统(简称CNC),使数控装置进入了以小型计算机为特征的第四代。

1974年,研制成功使用微处理器和半导体存储器的微型计算机数控装置(简称MNC),这是第五代数控系统。

20世纪80年代初,随着计算机软、硬件技术的发展,出现了能进行人机对话式自动编制程序的数控装置。数控装置愈趋小型化,可以直接安装在机床上;数控机床的自动化程度进一步提高,具有自动监控刀具破损和自动检测工件等功能。

20世纪90年代后期,出现了PC+CNC智能数控系统,即以PC机为控制系统的硬件部分,在PC机上安装NC软件系统,此种系统维护方便,易于实现网络化制造。

数控机床的类型,已从最初单一的铣床类数控机床,发展到如今的金属切削类、金属成型类、特种加工类和特殊用途类数控机床,其品种多达千余种。

数控机床由程序介质、数控系统、伺服驱动和机床主体四大部分组成,它综合了计算机、自动控制、精密测量、机床制造及其配套技术的最新成果,成功地解决了现代产品多样化、零件形状复杂化、产品研制生产周期短、精度要求高的难题,是现代制造业的主流设备,也是关系国计民生、国防尖端建设的战略物资。

近年来,带有刀库并能够自动更换刀具的数控机床——加工中心的发展速度十分迅速。相继出现的双托盘和多托盘自动交换的加工中心和柔性制造单元(FMC),由多台加工中心、物流系统、工业机器人及相应的信息流和中央控制系统组成的柔性制造系统(FMS),可实现24~120小时无人化运转。办公自动化(OA)与柔性制造系统(FMS)集成,实现工厂自动化(FA)生产。这些都改变了传统的制造模式,使制造业朝着自动化、柔性化、集成化方向发展。

1.1.3 我国数控机床的发展

我国从1958年开始研究数控机床,一直到60年代中期处于研制、开发阶段。1965年,国内开始研制晶体管数控系统。20世纪70年代初,我国研制出数控铣床和数控插齿机,数控技术在车、铣、镗、磨、齿轮加工、电加工等领域全面展开,数控加工中心在上海、北京研制成功。但由于电子元件的质量和制造工艺水平低,致使数控系统的可靠性、稳定性问题没有得到解决,因此未能广泛推广。这一时期,数控线切割机床由于结构简单、方便实用、价格低廉,在模具加工中得到了推广。

20世纪80年代,我国先后从日本、美国等国家引进了一部分数控装置和伺服系统技术,并于1981年在我国开始批量生产。在此期间,我国在引进、消化吸收的基础上,跟踪国外先进技术的发展,开发出了一些高档的数控系统,如多轴联动数控系统、数字仿形系统、为柔性单元配套的数控系统等。为了适应机械工业生产不同层次的需要,我国开发出了多种经济型数控系统,并得到了广泛应用。现在,我国已建立了高、中、低档数控机床为主的产业体系,在满足国内需求的同时,有大批量的数控机床远销海外。

1.1.4 数控机床的发展趋势

数控系统技术的突飞猛进为数控机床的技术进步提供了条件,为了满足市场的需要,达到现代制造技术对数控机床提出的更高要求,当前,数控技术及数控机床的发展方向主要体现在以下几方面。

1. 高速、高效

机床向高速化方向发展,不但可大幅度提高加工效率,降低加工成本,而且还可提高零件的表面加工质量和精度。超高速加工技术对制造业实现高效、优质、低成本生产有广泛的适用性。

目前,在超高速加工中,车削和铣削的切削速度已达到 $5000\sim 8000\text{ m/min}$ 以上;主轴转速在 30000 r/min (有的高达 10^6 r/min)以上;工作台的移动速度(进给速度)在分辨率为 $1\text{ }\mu\text{m}$ 时,达到 100 m/min (有的到 200 m/min)以上,在分辨率为 $0.1\text{ }\mu\text{m}$ 时,达到 24 m/min 以上;自动换刀速度在 1 s 以内;小线段插补进给速度达到 12 m/min 。

2. 多功能

在零件加工过程中有大量的无用时间消耗在工件搬运、上下料、安装调整、换刀和主轴的升降上,为了尽可能降低这些无用时间,人们希望将不同的加工功能整合在同一台机床上,因此,复合功能的机床成为近年来发展很快的机种。

3. 智能化

智能化是21世纪制造技术发展的一个大方向。智能加工是一种基于神经网络控制、模糊控制、数字化网络技术和理论控制机床的加工,它是要在加工过程中模拟人类专家的智能活动,以解决加工过程许多不确定性的、要由人工干预才能解决的问题。

智能化的内容体现在数控系统中的各个方面:

(1)为追求加工效率和加工质量的智能化,如自适应控制,工艺参数自动生成。

(2)为提高驱动性能及使用连接方便的智能化,如前馈控制、电机参数的自适应运算、自动识别负载自动选定模型、自整定等。

(3)简化编程、简化操作的智能化,如智能化的自动编程,智能化的人机界面等。

(4)智能诊断、智能监控,方便系统的诊断及维修等。

4. 高精度

在机械加工高精度的要求下,普通级数控机床的加工精度已由 $\pm 10 \mu\text{m}$ 提高到 $\pm 5 \mu\text{m}$;精密级加工中心的加工精度则从 $\pm 3\sim 5 \mu\text{m}$,提高到 $\pm 1\sim 1.5 \mu\text{m}$,甚至更高;超精密加工精度进入纳米级($0.001 \mu\text{m}$),主轴回转精度要求达到 $0.01\sim 0.05 \mu\text{m}$,加工圆度为 $0.1 \mu\text{m}$,加工表面粗糙度 $Ra=0.003 \mu\text{m}$ 等。这些机床一般都采用矢量控制的变频驱动电主轴(电机与主轴一体化),主轴径向跳动小于 $2 \mu\text{m}$,轴向窜动小于 $1 \mu\text{m}$,轴系不平衡度达到G0.4级。

5. 高可靠性

随着数控机床网络化应用的发展,数控机床的高可靠性已经成为数控系统制造商和数控机床制造商追求的目标。

6. 柔性化

柔性自动化技术是制造业适应动态市场需求及产品迅速更新的主要手段,是各国制造业发展的主流趋势,是先进制造领域的基础技术。其重点是以提高系统的可靠性、实用化为前提,以易于联网和集成为目标;注重加强单元技术的开拓、完善;CNC单机向高精度、高速度和高柔性方向发展;数控机床及其构成柔性制造系统能方便地与CAD、CAM、CAPP、MTS联结,向信息集成方向发展;网络系统向开放、集成和智能化方向发展。

1.2 数控机床的主要组成与工作过程

1.2.1 数控机床的组成

现代计算机数控机床由程序、输入/输出设备、计算机数控装置、可编程控制器、主轴控制单元即速度控制单元等部分组成,如图 1-1 所示。

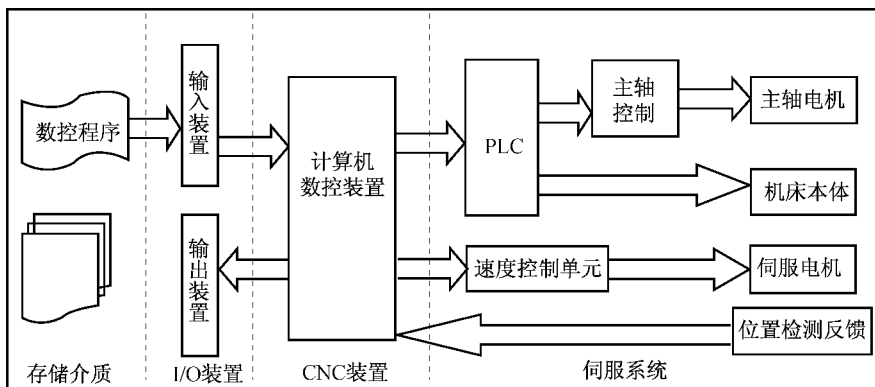


图 1-1 CNC 系统图

1. 程序的存储介质

数控程序是数控机床自动加工零件的工作指令。在对加工零件进行工艺分析的基础

上,确定零件坐标系在机床坐标系上的相对位置,即零件在机床上的安装位置;刀具与零件相对运动的尺寸参数;零件加工的工艺路线、切削加工的工艺参数以及辅助装置的动作等。得到零件的所有运动、尺寸、工艺参数等加工信息后,用由文字、数字和符号组成的标准数控代码,按规定的方法与格式编制零件加工的数控程序单。编制程序的工作可由人工进行,对于形状复杂的零件,则要在专用的编程机或通用计算机上进行自动编程(APT)或CAD/CAM设计。编好的数控程序存放在便于输入到数控装置的一种存储载体上,它可以是软盘、磁盘、U盘以及通信网络。

2. 输入/输出(I/O)装置

输入装置的作用是将程序载体(信息载体)上的数控代码传递并存入数控系统内。根据控制存储介质的不同,输入装置可以是光电阅读机、磁带机或软盘驱动器等。数控机床加工程序也可以通过键盘手工方式直接输入数控系统;数控加工程序还可通过编程计算机用RS-232或网络通信方式传送到数控系统中。

3. 数控装置

数控系统是机床实现自动加工的核心,是整个数控机床的灵魂所在。数控系统主要由输入元件、监视器、主控制系统、可编程控制器、输入/输出接口等组成。主控制系统主要由CPU、存储器、控制器等组成。数控系统的主要控制对象是位置、角度、速度等机械量,以及温度、压力、流量等物理量,其控制方式可分为数据运算处理控制和时序逻辑控制两大类。其中主控制器内的插补模块就是根据所读入的零件程序,通过译码、编译等处理后,进行相应的刀具轨迹插补运算,并通过与各坐标伺服系统的位置、速度反馈信号的比较,从而控制机床各坐标轴的位移。而时序逻辑控制通常由可编程控制器PLC来完成,它根据机床加工过程中各个动作要求进行协调,按各检测信号进行逻辑判别,从而控制机床各个部件有条不紊地按顺序工作。

4. 伺服系统

伺服系统包括伺服驱动电机、各种伺服驱动元件和执行机构等,它是数控系统的执行部分。它的作用是把来自数控装置的脉冲信号转换为机床移动部件的运动。每一个脉冲信号使机床移动部件的位移量称为脉冲当量(也称为最小设定单位或机床的分辨率)。常用的脉冲当量为0.001 mm/脉冲。每个进给运动的执行部件都有相应的伺服驱动系统,整个机床的性能主要取决于伺服系统。常用伺服驱动元件有直流伺服电机、交流伺服电机、电液伺服电机等。

5. 检测反馈系统

检测反馈装置的作用是对机床的实际运动速度、方向、位移量以及加工状态加以检测,把检测结果转化为电信号反馈给数控装置,通过比较,计算出实际位置与指令位置之间的偏差,并发出纠正误差指令。位置检测装置包括光栅、旋转编码器、磁栅等,用于检测执行部件的角度和位移量。

6. 机床本体

数控机床的机床本体与传统机床相似,由主传动装置、进给传动装置、床身、工作台以及辅助运动装置、液压气动系统、润滑系统、冷却装置等组成。但数控机床在整体布局、外观造型、传动系统、刀具系统的结构以及操作机构等方面都已发生了很大的变化。这种变化的目的是为了满足不同数控机床的要求和充分发挥数控机床的特点。

7. 辅助装置

辅助装置主要包括自动换刀装置 ATC(Automatic Tool Changer)、自动交换工作台机构 APC(Automatic Pallet Changer)、工件夹紧放松机构、回转工作台、液压控制系统、润滑装置、切削液装置、排屑装置、过载和保护装置等。

1.2.2 数控机床的工作过程

数控机床就是将加工过程的各种机床动作,由数字化的代码表示,通过某种载体将信息输入数控系统,控制计算机对输入的数据进行处理,来控制机床的伺服系统或其他执行元件,使机床加工出所需的工件。利用数控机床完成零件的数控加工过程如图 1-2 所示,其主要包括以下内容:

(1)根据零件的加工图样进行工艺分析,确定加工方案、工艺参数、位置数据、夹具选用和刀具类型选择等。

(2)用规定的程序代码和格式编写零件加工程序单;或用自动编程软件进行计算机辅助设计与制造(CAD/CAM)工作,直接生成零件的加工程序文件。

(3)程序的输入或传输。由手工编写的程序,可以通过数控机床的操作面板输入程序;由自动编程软件生成的程序,通过计算机的串行口直接传输到数控机床的数控单元(MCU)。

(4)将输入传输到数控单元的加工程序,进行试运行、刀具路径模拟等。

(5)通过对机床的正确操作,运行调试程序。

(6)完成零件的加工。

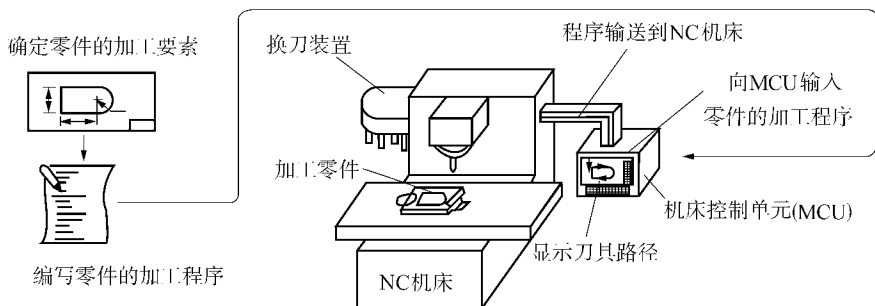


图 1-2 数控加工过程

1.3 数控机床的分类与特点

1.3.1 数控机床的分类

数控机床的品种规格较多,从不同角度对其进行考查,就有不同的分类方法,一般可根据其功能和结构,按以下原则进行分类。

1. 按机床运动的控制功能分类

1) 点位控制数控机床

这类机床仅能实现刀具相对于工件从一点到另一点的精确定位运动,对点与点之间的

运动轨迹不作控制要求,在运动过程中不进行任何加工,各坐标轴之间的运动是不相关的,可以同时移动,也可以依次运动。为了实现快速精确的定位,两点间位移的移动一般是先快速移动,然后慢速趋近定位点,以确保定位精度。如图 1-3 所示为点位控制的运动轨迹。

具有点位控制功能的机床主要有数控钻床、数控坐标镗床、数控冲床和数控测量机等。随着数控技术的发展,单纯用于点位控制的数控系统已不多见。

2) 直线控制数控机床

这类机床除了要求控制点与点之间的准确位置外,还需控制两相关点之间的移动速度和移动轨迹,一般是沿与坐标轴平行的方向作切削运动,也就是说同时控制的坐标轴只有一个,如图 1-4 所示。具有直线控制功能的机床主要有比较简单的数控车床、数控铣床、数控磨床等,用于加工外形简单的矩形、台阶形零件。同样,单纯用于直线控制的数控机床也不多见。

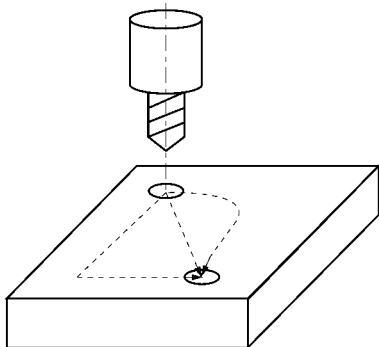


图 1-3 点位控制加工

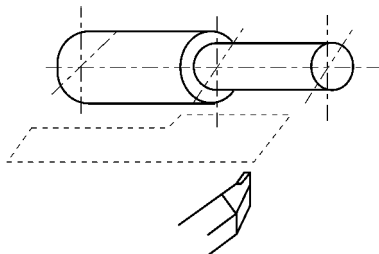


图 1-4 直线控制加工

3) 连续控制数控机床

连续控制数控机床也称为轮廓控制数控机床,能同时控制两个或两个以上坐标轴联动,使工件相对于刀具按程序规定的轨迹和速度运动,在运动过程中进行连续切削加工。这就要求数控装置必须具有插补运算功能,控制各坐标轴的联动位移量与要求的轮廓相符合。

具有连续控制功能的机床有数控车床、数控铣床、数控线切割机床、加工中心等用于加工曲线和曲面的机床,根据机床所控制的联动轴数不同,又可分为以下几种形式:

(1) 二轴联动。主要用于数控车床加工旋转曲面或数控铣床加工曲线柱面,如图 1-5 所示。

(2) 二轴半联动。主要用于三轴以上机床的控制,其中任意两根轴联动,第三根轴作周期性进给。如图 1-6 所示为二轴半联动行切法加工三维空间曲面。

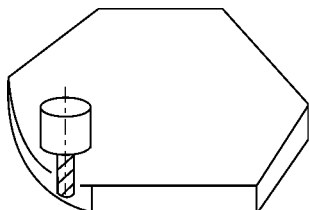


图 1-5 二轴联动加工

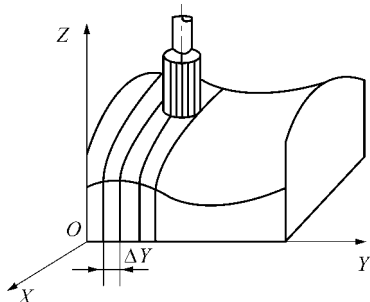


图 1-6 二轴半联动加工

(3)三轴联动。三轴联动一般分两类,一类是 X 、 Y 、 Z 三个直线坐标轴联动,多用于数控铣床和加工中心,如图 1-7 所示用球头铣刀铣切三维空间曲面;另一类是除了同时控制 X 、 Y 、 Z 中两个直线轴外,还同时控制绕某一直线坐标轴旋转的旋转坐标轴,如车削加工中心,除了实现纵向(Z 轴)、横向(X 轴)两个直线坐标轴的联动外,还需同时控制绕 Z 轴旋转的主轴联动。

(4)四轴联动。同时控制 X 、 Y 、 Z 三个直线坐标轴与某一旋转坐标轴联动,如图 1-8 所示为同时控制 X 、 Y 、 Z 三个直线坐标轴与一个工作台回转轴联动的数控机床。

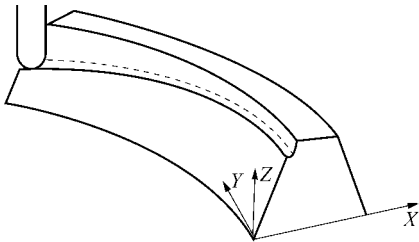


图 1-7 三轴联动

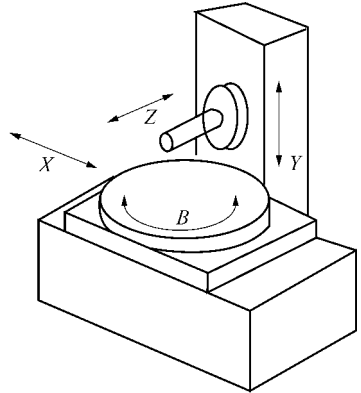


图 1-8 四轴联动

(5)五轴联动。除同时控制 X 、 Y 、 Z 三个直线坐标轴联动外,还需同时控制绕这些直线坐标轴旋转的 A 、 B 、 C 坐标轴中的两个坐标轴,如图 1-9 所示。

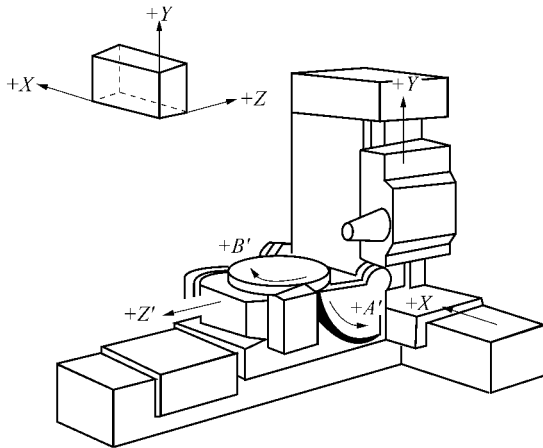


图 1-9 五轴联动

2. 按伺服控制的类型分类

按数控系统的进给伺服系统有无位置测量装置可分为开环数控系统和闭环数控系统,在闭环数控系统中根据位置测量装置安装的位置又可分为全闭环和半闭环两种。

1) 开环控制数控机床

如图 1-10 所示,数控装置发出信号的流程是单向的,没有位置测量装置,其精度主要取决于伺服驱动系统和机械传动机构的性能和精度,一般以功率步进电机作为伺服驱动元件。

开环控制数控机床具有结构简单、工作稳定、调试方便、维修简单、价格低廉等优点,在精度和速度要求不高、驱动力矩不大的场合得到广泛应用,一般用于经济型数控机床。

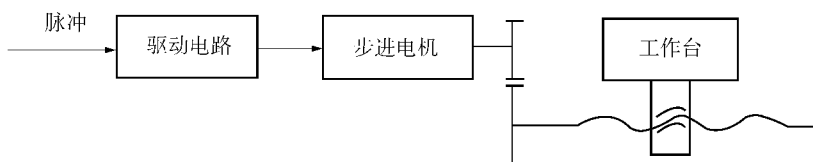


图 1-10 开环控制系统框图

2) 全闭环控制数控机床

如图 1-11 所示,全闭环数控系统是直接检测机床坐标的直线位移量,通过反馈可以消除整个驱动和传动环节的误差、间隙和矢动量,具有很高的位置控制精度。但由于位置环内的许多机械传动环节的摩擦特性、刚性和间隙都是非线性的,很容易造成系统的不稳定,使闭环系统的设计、安装和调试都相当困难。该控制系统主要用于精度要求很高的镗铣床、超精车床、超精磨床以及较大型的数控机床。

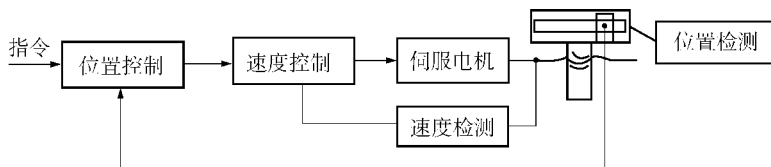


图 1-11 全闭环控制系统框图

3) 半闭环控制数控机床

如图 1-12 所示,半闭环控制系统并不检测移动部件的实际位置,而是检测伺服电机或传动丝杠的转角,由此推算出工作台实际位置,半闭环环路内不包括或只包括少量机械传动环节,因此可获得稳定的控制性能,其系统的稳定性虽不如开环系统,但比闭环系统要好。由于丝杠的螺距误差和齿轮间隙引起的运动误差难以消除,其精度比闭环系统差,比开环系统好。若对这类误差进行补偿,可获得满意的精度。半闭环控制系统结构简单、调试方便、精度也较高,在现代数控机床中得到了广泛应用。

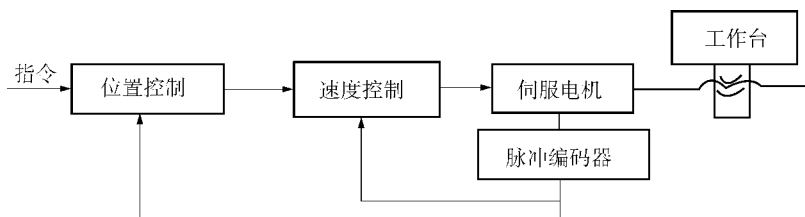


图 1-12 半闭环控制系统框图

3. 按数控系统的功能水平分类

按数控系统的功能水平,通常把数控系统分为低、中、高三类。三档的界限是相对的,不同时期划分的标准也会有所不同。就目前的发展水平而言,可以根据如表 1-1 所示的一些功能及指标进行划分。高档数控系统一般称为全功能型或标准型数控系统。在我国还有经济型数控系统的说法,经济型数控系统属于低档数控系统,是由单片机和步进电机组成的数

控系统,功能简单、成本低廉,主要用于数控车床、数控线切割机床以及旧机床改造等。

表 1-1 数控系统不同档次的功能及指标

功 能	低 档	中 档	高 档
分辨率/ μm	10	1	0.1
进给速度/(m/min)	8~15	15~24	15~100
驱动进给类型	开环	半闭环或闭环的直流或交流伺服系统	
联动轴数/(轴)	2~3	2~4	3~5 以上
通信功能	一般无	RS-232 或 DNC 接口	可有 MAP 通信接口 ^① ,有联网能力
显示功能	LED 或简单的 CRT	较齐全的 CRT 显示 ^②	还有三维图形显示
内装 PLC	无	有	有强功能的 PLC
主 CPU	8 位、16 位	32 位以上或 32 位以上的多 CPU	

注:① MAP——Manufacturing Automation Protocol,制造自动化协议。

② 较齐全的 CRT 显示是指具有字符、图形、人机对话、自诊断等功能的显示。

4. 按加工工艺及机床用途的类型分类

1) 切削加工类

采用车、铣、镗、磨、刨、齿轮加工等各种切削工艺的数控机床。它又可分为以下两类:

(1) 普通型数控机床。如数控镗铣床、数控车床、数控磨床、数控齿轮加工机床等。

(2) 加工中心。加工中心是带有刀库和自动换刀装置的数控机床。工件经一次装夹后,通过自动更换各种刀具,在同一台机床上对工件各加工表面连续进行铣(车)、镗、铰、钻、攻螺纹等多种工序的加工,如镗/铣类加工中心、车削中心、钻削中心等。

2) 成型加工类

采用挤、冲、压、拉等成型工艺的数控机床。常用的有数控压力机、数控折弯机、数控弯管机、数控旋压机等。

3) 特种加工类

主要有数控线切割机、数控电火花加工机、数控火焰切割机、数控激光加工机等。

4) 其他类型

主要有三坐标测量仪、数控装配机、数控测量机、数控绘图仪、机器人等。

1.3.2 数控机床的特点

1. 具有高柔性

在数控机床上加工零件,主要取决于加工程序,它与普通机床不同,不必制造、更换许多工具、夹具,不需要经常调整机床。因此,数控机床适用于零件频繁更换的场合。也就是适合单件、小批生产及新产品的开发,缩短了生产准备周期,节省了大量工艺设备的费用。

2. 加工精度高

数控机床的加工精度,一般可达到 0.005~0.1 mm。数控机床是按数字信号形式控制的,数控装置每输出一个脉冲信号,则机床移动部件移动一个脉冲当量(一般为 0.001 mm),

而且机床进给传动链的反向间隙与丝杠螺距平均误差可由数控装置进行补偿,因此,数控机床定位精度比较高。

3. 加工质量稳定、可靠

加工同一批零件,在同一机床和相同加工条件下,使用相同刀具和加工程序,刀具的走刀轨迹完全相同,零件的一致性,质量稳定。

4. 生产率高

数控机床可有效地减少零件的加工时间和辅助时间,数控机床的主轴转速和进给量的范围大,允许机床进行大切削量的强力切削,数控机床目前正进入高速加工时代,数控机床移动部件的快速移动和定位及高速切削加工,减少了半成品的工序间周转时间,提高了生产效率。

5. 改善劳动条件

数控机床经调整后,输入程序并启动,就能自动连续地进行加工,直至加工结束。操作者主要进行程序的输入、编辑、装卸零件、刀具准备、加工状态的观测,零件的检验等工作,劳动强度极大降低,机床操作者的劳动趋于智力型工作。另外,数控机床一般是封闭式加工,既清洁,又安全。

6. 利于生产管理现代化

数控机床的加工,可预先精确估计加工时间,所使用的刀具、夹具可进行规范化、现代化管理。数控机床使用数字信号与标准代码为控制信息,易于实现加工信息的标准化,目前已与计算机辅助设计与制造(CAD/CAM)有机地结合起来,是现代集成制造技术的基础。

本章小结

本章引入了数控机床的一些基本概念,概述了数控机床的产生和发展历程。

数控机床的核心问题是如何用数字代码的方式记载、输入并控制机床工作台移动的距离、方向、轨迹和速度。数控机床一般由控制介质、数控装置、伺服系统、机床和测量反馈装置等组成。在CNC和MNC系统中,其数控装置的功能由一台计算机来实现。

数控机床可以按多种方式进行分类。若按工艺用途,则可分为一般数控机床、数控加工中心机床、多坐标机床等;若按加工路线分类,则可分为点位控制、点位直线控制和轮廓控制的数控机床;若按测量装置的有无及位置,则可分为开环、闭环和半闭环等系统;若按数控装置,则还可分为普通数控(NC)机床、计算机数控(CNC)机床和微处理机数控(MNC)机床等。

习题 1

1-1 什么是数控机床?

1-2 数控机床的发展经过了哪几个阶段?

1-3 数控机床由哪几部分组成?

- 1-4 数控机床有哪些分类方式？
- 1-5 点位控制、点位直线控制和轮廓控制机床的最主要的区别是什么？
- 1-6 开环、闭环和半闭环控制系统由哪些部分组成？
- 1-7 数控机床的发展趋势是什么？
- 1-8 数控机床的特点是什么？

第 2 章 数控系统

CNC(Computer Numerical Control,简称 CNC)即计算机数控,是用数字化的信息对机床运动及加工过程进行控制的一种方法。CNC 是在早期硬件数控系统的基础上发展起来的,这种靠硬件逻辑电路控制的系统,其通用性、灵活性、功能性方面都较差。随着技术的发展,CNC 的内部结构变为软接线系统,机床的控制功能也有了很大的提升。

2.1 CNC 装置

2.1.1 CNC 装置的结构

1. CNC 系统及 CNC 装置

CNC 系统是用计算机控制加工功能,实现数值控制的系统。CNC 系统是根据计算机存储器中存储的控制程序,执行部分或全部数值控制功能,并配有接口电路和伺服驱动装置的专用计算机系统。CNC 系统由硬件和软件两部分组成。如图 2-1 所示为典型的两坐标 CNC 系统框图,由输入/输出装置、CNC 装置、主轴驱动装置和进给伺服装置系统等组成。

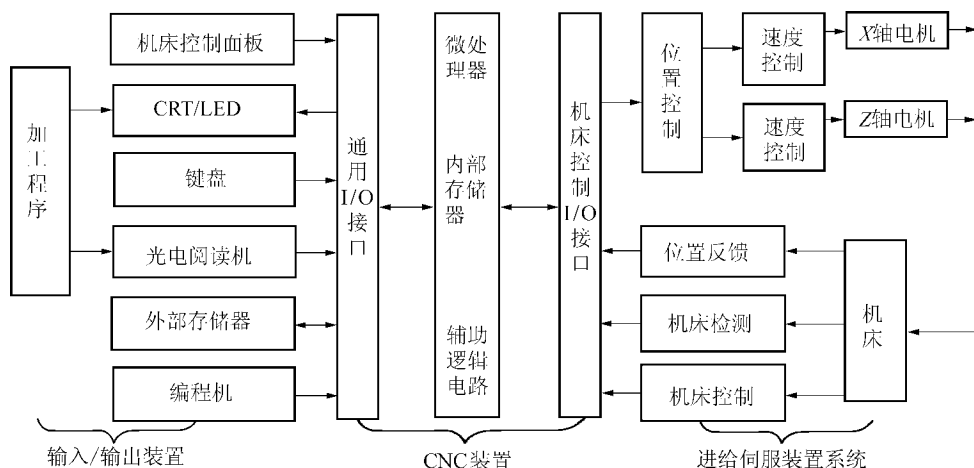


图 2-1 典型的两坐标 CNC 系统框图

1) CNC 装置

CNC 装置是 CNC 系统的核心部分。CNC 装置由通用 I/O 接口、机床 I/O 接口、微处理器、内部存储器、辅助逻辑电路等组成。微处理器执行存储在内部存储器中的加工程序,完成所需要的逻辑分析和数值计算,产生协调整个系统的各类控制信号指令。通用 I/O 接

口是微处理器与外界联系的通路,它完成数据的格式和信号形式的转换,实现程序的输入输出及人机对话。机床控制 I/O 接口连接专用的控制和检测装置,实现机床的位置和工作状态的控制和检测。

2)输入/输出装置

外部输入装置可以实现数字信号的输入。数控系统的加工程序可以通过机床操作面板上的键盘输入到机床数控系统,也可以通过光电阅读器输入机床数控系统,还可以通过外部计算机传输导线与数控机床的 I/O 接口连接输入到机床数控系统。

输出装置可以实现数字信息的输出,即按照指令信号驱动数控设备的执行机构,根据要求的轨迹作严格的相对运动和完成精确的定位。执行机构一般有步进电机驱动系统、直流伺服系统、交流伺服系统等。数控系统中的信息可以通过机床操作面板上的显示器输出,也可以通过数控机床的 I/O 接口输出到外部计算机上。

3)主轴驱动和进给伺服系统

主轴驱动和进给伺服系统主要由位置控制装置组成。通过位置检测装置获得数控设备的实际位移信息和坐标的定位信息。位置控制装置中包括伺服机构和执行元件。执行元件可以是交流或直流伺服电动机,也可以是步进电机。伺服机构包括速度控制和位置控制两部分,一般构成闭环控制。速度环通常包含在伺服电动机驱动器中,而位置控制的反馈信号则送入微处理器,经微处理器构成回路。对于闭环数控系统,其定位精度和加工精度主要由测量装置的精度所决定。

CNC 系统不仅包括 CNC 装置、输入/输出装置、主轴驱动和进给伺服系统,还包括可编程控制器(PLC)。PLC 在数控机床中主要完成各执行机构的逻辑顺序(M、S、T 功能)控制,如更换刀具、主轴的启动停止、主轴转速的变换、主轴转向的变换、工件的装夹、冷却泵的开关等。CNC 系统由于使用了计算机,系统具有了软件功能,又用 PLC 代替了传统的机床电器逻辑控制装置,使系统更小巧,其灵活性、通用性、可靠性更好,易于实现复杂的数控功能,使用和维护也方便,并具有与上位机连接及进行远程通信的功能。数控机床中的 PLC 有内装性和独立性两种。CNC 装置中的 PLC 为内装型,其硬件电路与微处理器的电路制作在同一块印制板上。如图 2-2 所示为内装型 PLC 的 CNC 装置。

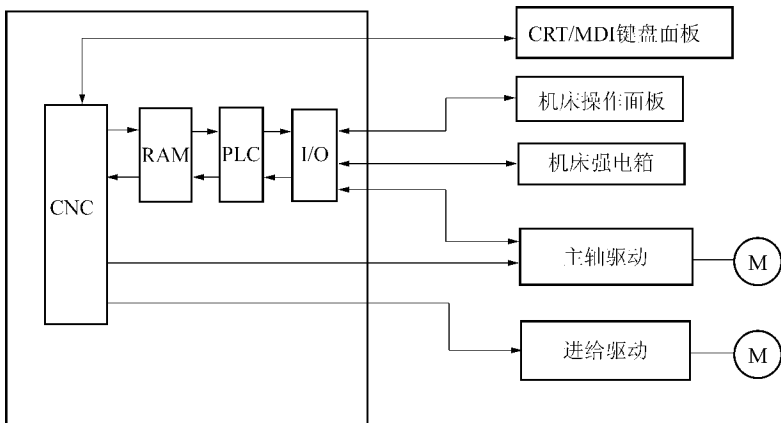


图 2-2 内装型 PLC 的 CNC 装置

2. CNC 装置的功能

CNC 装置的功能是指满足用户操作和机床控制要求的方法及手段,包括基本功能和选择功能。基本功能是必备的功能,用于满足数控系统基本配置的要求;选择功能是根据实际要求选择的功能。CNC 装置主要功能有以下几方面。

1) 控制功能

控制功能能够控制和联动控制的进给轴数,包括移动轴、回转轴、基本轴、附加轴的控制。控制的进给轴数越多,表明 CNC 装置的功能越强。

2) 准备功能

准备功能即 G 代码功能,其作用是使机床准备好某种加工方式,包括的指令有基本移动、程序暂停、平面选择、坐标设定、刀具补偿、固定循环加工、公英制转换、子程序调用等。

3) 插补功能和固定循环功能

插补功能是指实现零件轮廓加工轨迹运算的功能。一般的 CNC 装置具有直线插补、圆弧插补功能;高档的 CNC 装置还具有椭圆插补、正弦线插补、抛物线插补、螺旋线插补、样条曲线插补等功能。

固定循环功能是指在加工一些特定表面(如车削台阶、切削螺纹、钻孔、镗孔、攻丝)时,加工动作按照一定的循环模式多次重复进行,实现上述加工轨迹运算的功能,即把若干有关的典型固定动作顺序用一个指令来表示,用 G 代码定义,直接调用,可大大简化编程。

4) 进给功能、进给速度的控制功能

进给功能、进给速度的控制功能,主要包括:

(1) 进给速度是指控制刀具相对工件的进给速度,所用单位一般是 mm/min。

(2) 同步进给速度是指实现切削速度和进给速度的同步,用于加工螺纹,所用单位一般是 mm/r。

(3) 进给倍率又称为进给修调率,是指通过操作面板上的波段开关,人工实时修调的进给速度。

5) 主轴功能

主轴功能指主轴的控制功能,主要包括:

(1) 切削速度即主轴转速控制的功能,所用单位一般是 m/min 或 r/min。

(2) 恒线速度是指刀具切削点的切削速度为恒速的控制功能,主要用于车端面或磨削加工,可获得较高的表面质量。

(3) 主轴定向控制是指主轴在径向(周向)的某一位置准确停止的功能,常用于换刀。

(4) C 轴控制是指主轴在径向(周向)的任一位置准确停止的功能。

(5) 切削倍率(主轴修调率)是指通过操作面板上的波段开关,人工实时修调切削速度的功能。

6) 辅助功能

辅助功能是指机床的辅助操作功能,即 M 指令功能。辅助功能包括主轴正转、反转、停止,冷却泵的打开和关闭、工件夹紧、松开和换刀等功能。

7) 刀具管理功能

刀具管理功能是指实现刀具几何尺寸、刀具寿命、刀具号的管理的功能。其中刀具几何尺寸一般指刀具半径、长度参数,常用于刀具半径补偿、长度补偿;刀具寿命一般指刀具的使

用时间；刀具号的管理是用于标识、选择刀具，常和 T 指令连用。

8) 补偿功能

补偿功能主要包括以下几个方面：

- (1) 刀具半径补偿和长度补偿。
- (2) 传动链误差补偿。一般有螺距误差补偿和反向间隙补偿。
- (3) 智能补偿。

9) 人机对话功能

人机对话功能是指通过显示器进行字符、图形的显示，从而方便用户的操作和使用。

10) 自诊断功能

自诊断功能是指利用软件诊断程序，在故障出现后，可迅速查明故障的类型和部位，以便及时排除。

11) 通信功能

通信功能是指 CNC 装置与外界进行信息和数据交换的功能。一般 CNC 装置具有 RS-232C 接口，可与上级计算机相连；若具有 DNC 接口则可实现直接数控；若具有 FMS 接口则可按 MAP(制造自动化协议)通讯，实现车间和工厂的自动化。

除以上各项功能外，CNC 装置还可配置选择功能。如增加一个仿形测量头和相应的驱动模块等，可实现数字化仿形加工功能。

2.1.2 CNC 装置的工作过程

机床数控系统是一种位置控制系统。数控机床的任务是依照操作者的意愿完成所要加工零件。操作者根据被加工零件的尺寸要求、外形要求、表面指令要求编制零件加工程序。加工程序通过输入装置输入到数控系统中，存储到数控机床存储介质上。数控系统接收到零件的加工程序后，对其进行译码和数据计算，将编程语言转换为数控机床可以直接执行的机器代码，根据这些数据计算出理论的刀具运动轨迹，然后将计算过的结果输送到执行元件，使刀具按轨迹加工出所需要的形状。因此，CNC 在工作过程中完成以下任务。

1. 加工程序的输入

CNC 装置中有较大容量的存储空间，有些数控机床还配有较大的外部扩展存储，因此，CNC 装置有较强的存储信息的能力。它不但可把一个加工程序一次装入内存，而且可以在内存中同时保存多个零件的加工程序，可以把以后需要加工的零件程序提前输入并保存到数控装置中，同时还可以通过键盘和显示器现场编辑和修改零件加工程序。现在大多数数控机床都配有掉电保护装置，来保护存储器中的零件加工程序。CNC 配有磁盘外部存储器，可以对磁盘上的加工程序进行存入、调出、查找、删除。通常数控机床中有专门的存储器存储零件加工程序，在机床加工零件时，再从存储器中将程序一段一段调出，并执行程序指令。

2. 数据的译码和计算

数控机床加工时，微机从零件加工程序存储区中逐条调取出加工指令，由数控装置的译码程序 and 数据处理程序完成零件加工程序的译码及数据计算工作。译码程序的主要功能是将用文本格式(通常为 ASCII 码)表达的零件加工程序，以一个程序段为单位，根据一定的语

法规则解释、翻译成后续程序(补偿计算程序),即机床能够识别的数据形式,并以一定的数据格式存放在指定的内存专用区内。该数据结构用来描述一个程序段解释后的数据信息。它主要包括 X、Y、Z 等坐标值,进给速度,主轴转速,G 代码,M 代码,刀具号,子程序处理和循环调用处理等数据或标志的存放顺序和格式。一个译码缓冲区数据结构的例子如下:

```
Struct PROG_BUFFER
```

```
char bur-state;(指定缓冲区的状态,0 表示缓冲区为空;1 表示缓冲区准备好)
```

```
int block-num;(以 BCD 码的形式存放本段的程序段号)
```

```
double COOR [20];(以二进制的形式存放 X、Y、Z、I、J、K、R、A、B、C 等)
```

```
char G0;(以标志形式存放 G 指令)
```

如图 2-3 所示为译码缓冲区数据结构。

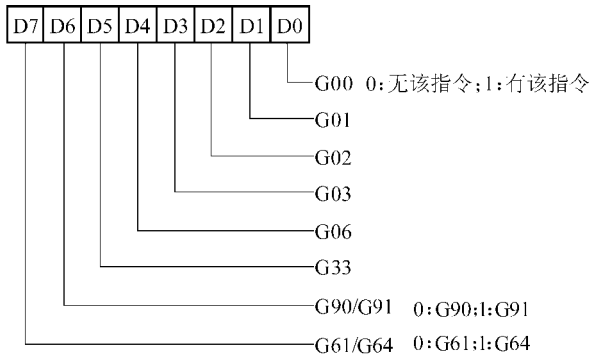


图 2-3 译码缓冲区数据结构

```
char G1;(根据 G 指令的个数设置字符变量的个数)
```

```
char M0;(以标志的形式存放 M 指令,存放形式同 C 代码)
```

```
char M1;(根据 M 指令的个数设置字符变量的个数)
```

```
char T;(以 BCD 码的形式存放本段换刀的刀具号)
```

```
char D;(以 BCD 码的形式存放刀具补偿的刀具半径值)
```

在程序中一般都有若干个由这种结构组成的程序缓冲区组,当前程序段被解释完后便将该段的数据信息送入缓冲区组中空闲的一个。后续程序(如补偿计算程序)从该缓冲中获取程序信息进行工作。下面以一个程序段为例简要说明译码的过程:

```
N06 G90 G01 X200 Y300 F200;
```

译码程序以程序段为单位进行解释,在解释中它从零件程序存储区中逐一读出指令。当译码程序解释 N06 程序段时,首先读入的是 N06 指令,它将“06”按 BCD 码的形式存放在译码缓冲区的“block-num”里;接下去读入 G90,译码程序将“G0”的 D6 位清 0;当读入 G01 后就将“G0”的 D1 位置 1;当读入 X200、Y300、F200 后,将其中的数据转换成二进制码后分别存放在“COOR[1]”、“COOR[2]”和“F”中;当读到程序段的结束符“;”时,译码程序就结束该程序段解释工作,并准备下一程序段的解释工作,直至整个缓冲区组被填满,然后译码程序就进入休眠状态。当缓冲区组中有若干缓冲区置空时(具体个数在系统设计时确定),系统将再次激活译码程序,按此方式重复进行,直到整个加工程序解释完毕(如发现 M02 或 M03 指令)为止。

3. 刀具补偿计算

操作者编制的零件加工程序通常是按零件轮廓编制的,而数控机床在加工过程中控制的是刀具(铣刀)中心轨迹或假想的虚拟刀尖(车刀刀尖)位置点。刀具补偿处理就是完成这种转换的程序,它主要进行以下几项工作:

(1)根据绝对坐标(G90)或增量坐标(G91)计算零件轮廓的终点坐标值。

(2)根据刀具半径和刀具半径补偿的方向(G41/42),计算刀具补偿后本段刀具中心轨迹的终点坐标值。

(3)根据本段与前段的连接关系,进行段间连续处理。

经过刀具补偿程序转换的数据存放在刀具补偿缓冲区中,以供后续程序之用。

4. 插补计算

插补计算的任务是确定某个坐标在规定的位移范围内进给的规律,从而获得所需要的轨迹。数控装置一般采用软件来实现插补计算,因此,要求微处理器有较高的运行速度。插补计算是周期进行的,这个周期称为插补周期。每进行一次插补计算,即形成一个微小的位移量,该位移量是对应轴的进给距离,这个进给距离称为脉冲当量。

插补计算程序完成 CNC 系统中插补器的功能,即实现坐标轴脉冲分配的功能。脉冲分配包括点位、直线以及曲线三个方面,由于现代微机具有完善的指令系统和相应的算术子程序,所以给插补计算提供了许多方便。可以采用一些更方便的数学方法提高轮廓控制的精度,而不必顾虑会增加硬件线路。插补计算是实时性很强的程序,要尽可能减少该程序中的指令条数,即缩短进行一次插补运算的时间。因为这个时间直接决定了插补进给的最高速度。在有些系统中还采用粗插补与精插补相结合的方法,软件只作粗插补,即每次插补一个小线段;硬件再将小线段分成单个脉冲输出,完成精插补。这样既可提高进给速度,又能使计算机空出更多的时间进行必要的数据处理。

插补计算程序以系统规定的插补周期定时运行,其主要功能是:

(1)根据操作面板上“进给修调”开关的设定值,计算本次插补周期的实际合成位移量 $\Delta L_i = \Delta L \times \text{修调值}$ 。

(2)将 ΔL_i 按插补的线形(直线、圆弧等)和本插补点所在的位置分解到各个进给轴,作为各进给轴的位置控制指令。

经插补计算后的数据存放在运行缓冲区中,以供位置控制程序之用。

插补计算的结果输出,经过位置控制部分(这部分工作既可由软件完成,也可由硬件完成),去带动伺服系统运动,控制刀具按预定的轨迹加工。位置控制的主要任务是在每个采样周期内,将插补计算出的理论位置与实际反馈位置相比较,用其差值去控制进给电机。在位置控制中,通常还要完成位置回路的增益调整、各坐标方向的螺距误差补偿和反向间隙补偿,以提高机床的定位精度。

5. 位置控制处理

位置控制数据转换流程,如图 2-4 所示。位置控制主要进行各进给轴跟随误差(ΔX_3 , ΔY_3)的计算,并进行调节处理,其输出为位移速度控制指令值(v_x, v_y)。其计算步骤为:

(1)计算新的位置指令坐标值,即

$$X_{1\text{新}} = X_{1\text{旧}} + \Delta X_1$$

$$Y_{1\text{新}} = Y_{1\text{旧}} + \Delta Y_1$$

(2) 计算新的位置实际坐标值, 即

$$X_{2\text{新}} = X_{2\text{旧}} + \Delta X_2$$

$$Y_{2\text{新}} = Y_{2\text{旧}} + \Delta Y_2$$

(3) 计算跟随误差, 即

$$\text{跟随误差} = \text{指令位置值} - \text{实际位置值}$$

(4) 计算速度指令值, 即

$$v_X = f(\Delta X_3)$$

$$v_Y = f(\Delta Y_3)$$

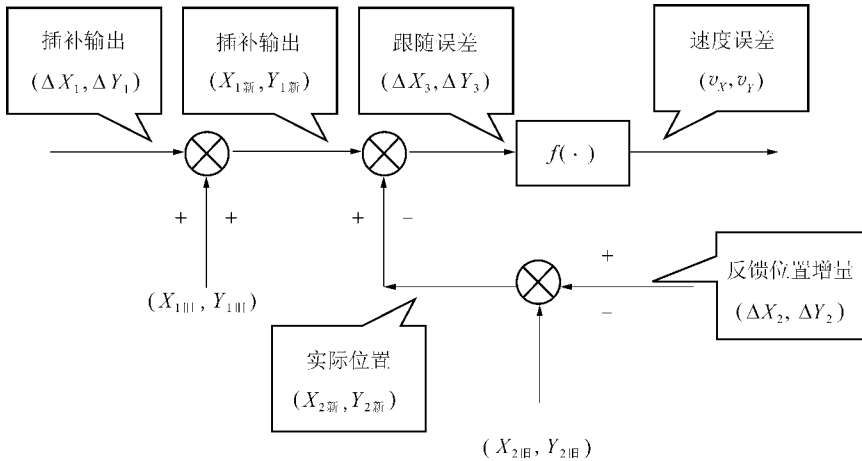


图 2-4 位置控制转换流程

2.1.3 CNC 装置的特点

CNC 是在与 NC 的基础上发展起来的, 它不但继承了 NC 硬件的优点, 又充分利用了微处理器结合控制软件的优势。CNC 系统与 NC 系统相比, 具有以下主要优点。

1. 灵活性

这是 CNC 系统的突出优点。对于传统的 NC 系统, 一旦提供了某些控制功能, 就不能被改变, 除非改变相应的硬件。而对于 CNC 系统, 只要改变相应的控制程序就可以补充和开发新的功能, 并不必制造新的硬件。在 CNC 设备安装之后, 新的技术还可以补充到系统中去, 这就延长了系统的使用期限。因此, CNC 系统具有很大的“柔性”——灵活性。

2. 通用性

在 CNC 系统中, 硬件系统采用模块结构, 依靠软件变化来满足被控设备的各种不同要求。采用标准化接口电路, 给机床制造厂和数控用户带来了许多方便。于是, 用一种 CNC 系统就可能满足大部分数控机床 (包括车床、铣床、加工中心、钻镗床等) 的要求, 还能满足某些其他设备的要求。当用户要求某些特殊功能时, 仅需改变某些软件即可。由于在工厂中使用同一类型的控制系统, 培训和学习也十分方便。

3. 可靠性

在 CNC 系统中, 加工程序常常是一次送入计算机存储器内, 避免了在加工过程中由于

纸带输入机的故障而产生的停机现象(普通数控装置的故障有一半以上发生在逐段光电输入时)。同时,由于许多功能都由软件实现,硬件系统所需元器件数目大为减少,整个系统的可靠性大大改善,特别是随着大规模集成电路和超大规模集成电路的采用,系统可靠性更为提高。据美国第13届NCS年会统计的世界上数控系统平均无故障时间是:硬线NC系统为136小时,小型计算机CNC系统为984小时,而微处理机CNC系统据日本发那科公司宣称已达23000小时。

4. 易于实现许多复杂的功能

CNC系统可以利用计算机的高度计算能力,实现一些高级的或复杂的数控功能。刀具偏移、英公制转换、固定循环等都能用适当的软件程序予以实现;复杂的插补功能,如抛物线插补、螺旋线插补等也能用软件方法来解决;刀具补偿可以在加工过程中进行计算;大量的辅助功能都可以被编程;子程序概念的引入,大大简化了程序编制。

5. 使用维修方便

CNC系统的一个吸引人的特点是有一套诊断程序,当数控系统出现故障时,能显示出故障信息,使操作和维修人员能了解故障部位,减少了维修的停机时间。另外,还可以备有数控软件检查程序,防止输入非法数控程序或语句,这将给编程带来许多方便。有的CNC系统还有对话编程、蓝图编程,使程序编制简便,不需要很高水平的专业编程人员。零件程序编好后,可显示程序,甚至通过空运行,将刀具轨迹显示出来,检验程序是否正确。

2.2 CNC系统硬件

2.2.1 CNC系统硬件的结构与分类

1. CNC系统硬件的结构

随着大规模集成电路技术和表面安装技术的发展,CNC系统硬件模块及安装方式不断改进。从CNC系统的总体安装结构看,有整体式结构和分体式结构两种。

所谓整体式结构是把CRT和MDI面板、操作面板以及功能模块板组成的电路板等安装在同一机箱内。这种方式的优点是结构紧凑,便于安装,但有时可能造成某些信号连线过长。分体式结构通常把CRT和MDI面板、操作面板等做成一个部件,而把功能模块组成的电路板安装在一个机箱内,两者之间用导线或光纤连接。许多CNC机床把操作面板也单独作为一个部件,这是由于所控制机床的要求不同,操作面板相应地有所改变,做成分体式有利于更换和安装。

CNC操作面板在机床上的安装形式有吊挂式、床头式、控制柜式、控制台式等。

从组成CNC系统的电路板的结构特点来看,有两种常见的结构,即大板式结构和模块化结构。

大板式结构的特点是一个系统一般都有一块大板,称为主板。主板上装有主CPU和各轴的位置控制电路等。其他相关的子板(完成一定功能的电路板),如ROM板、零件程序存

存储器板和 PLC 板都直接插在主板上面,组成 CNC 系统的核心部分。由此可见,大板式结构紧凑,体积小,可靠性高,价格低,有很高的性价比,也便于机床的一体化设计。大板结构虽有上述优点,但它的硬件功能不易变动,不利于组织生产。

另外一种柔性比较高的结构就是总线模块化的开放系统结构,其特点是将微处理机、存储器、输入/输出控制分别做成插件板(称为硬件模块),甚至将微处理机、存储器、输入/输出控制组成独立微计算机级的硬件模块,相应的软件也是模块结构,固化在硬件模块中。硬软件模块形成一个特定的功能单元,称为功能模块。功能模块间有明确定义的接口,接口是固定的,成为工厂标准或工业标准,彼此可以进行信息交换。于是,可以以积木式组成 CNC 系统,使设计简单,有良好的适应性和扩展性,试制周期短,调整维护方便,效率高。

2. CNC 系统硬件的分类

从 CNC 系统使用的微机及结构来分,CNC 系统的硬件结构一般分为单一微处理器和多微处理器结构两大类。初期的 CNC 系统和现有一些经济型 CNC 系统采用单微处理器结构。而多微处理器结构可以满足数控机床高进给速度、高加工精度和许多复杂功能的要求,也适应于并入 FMS 和 CIMS 运行的需要,从而得到了迅速的发展,它反映了当今数控系统的新水平。

2.2.2 CNC 系统中的微处理器

CNC 系统的微处理器从构成形式上可分为两类:一类是单微处理器处理系统,由一个处理器完成输入/输出、插补计算、伺服控制等各项任务。这类系统硬件和软件的结构比较简单。另一类是由多个微处理器构成的多处理器系统,也称为分布式处理系统。这类系统有较强的计算插补能力,有效地解决了机械加工的高速化、高精度化问题。但是这类系统的硬件结构较为复杂,软件设计也需要考虑各处理器间通信、同步问题,因此整个系统的设计也较为复杂。单微处理器系统主要用在经济型、一般型的数控装置中;而多微处理系统多用在高级型的数控装置中。

1. 单微处理器系统的组成和特点

单微处理器系统的 CNC 装置的特点是整个 CNC 装置中只有一个 CPU,通过该 CPU 来集中管理和控制整个系统的资源(包括存储器、总线),并通过分时处理的方法,实现各种数控功能。有些 CNC 装置中,虽然有两个或两个以上的 CPU,但只有一个 CPU 对系统的资源拥有控制权和使用权,该 CPU 称为主 CPU,其他 CPU(称为从 CPU)无权控制和使用系统资源,只能接受主 CPU 的控制命令和数据,或向主 CPU 发请求信号以获取所需要的数据,从而完成某一辅助功能,该结构称为主从结构,也可归为单机结构。

在单微处理机结构中,只有一个微处理机,实行集中控制,并分时处理数控的各个任务。其结构特点如下:

(1) CNC 装置内仅有一个微处理机,由它对存储、插补运算、输入/输出控制、CRT 显示等功能集中控制,分时处理。

(2) 微处理机通过总线与存储器、输入输出控制等各种接口相连,构成 CNC 装置。

(3) 结构简单,容易实现。

(4) 正是由于只有一个微处理机集中控制,其功能将受微处理机字长、数据宽度、寻址能

力和运算速度等因素的限制。

单微处理器系统的 CNC 装置由微处理器和总线、存储器(RAM、ROM)、I/O 接口、MDI 接口、位置控制器、纸带阅读机接口等组成。如图 2-5 所示为单微处理器系统框图。

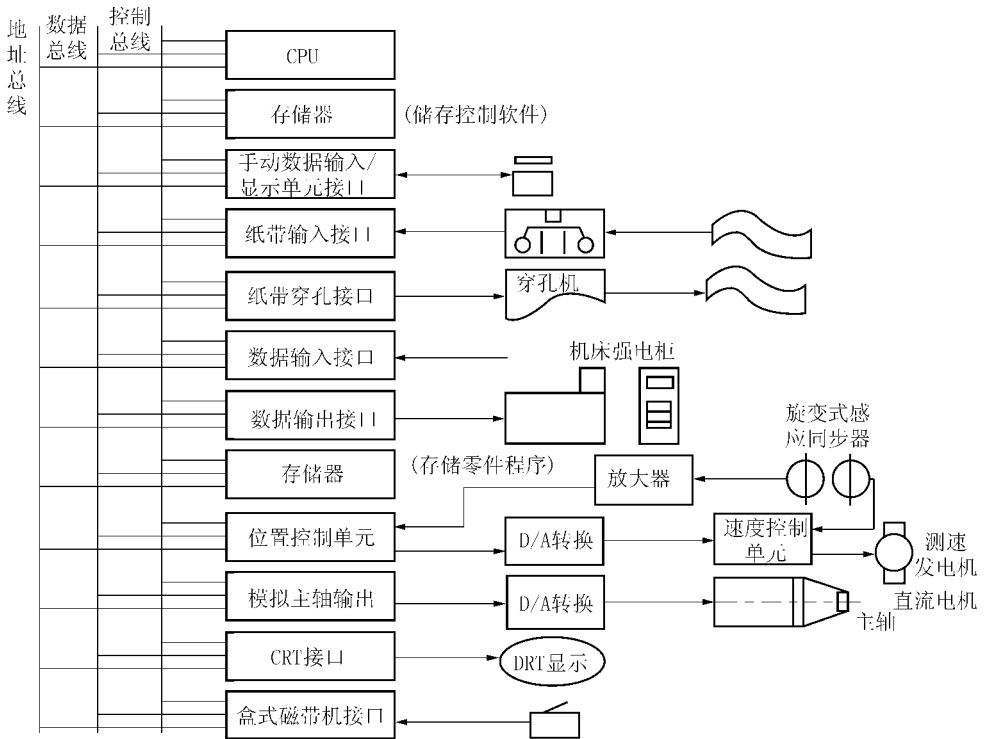


图 2-5 单微处理器系统框图

2. 多微处理器系统的组成和特点

单微处理器系统的 CNC 装置中只有一个主 CPU,因此数控功能的提高和扩展与 CNC 装置的处理速度存在矛盾。为进一步提高 CNC 装置性能,有必要采用多(微处理)机结构的 CNC 装置。

多微处理器系统的 CNC 装置中有两个或两个以上带 CPU 的功能部件可对系统资源(存储器、总线)有控制权和使用权,又分为多主结构和分布式结构两种。多主结构是指带 CPU 的功能部件之间采用紧耦合方式连接,有集中的操作系统用总线仲裁器解决总线争用通过公共存储器交换系统信息。分布式结构是指各个带 CPU 的功能模块有独立的运行环境(总线、存储器、操作系统),各功能模块之间采用松耦合方式连接,用通讯方式交换信息。

从数据处理的角度来看,多微处理机结构的 CNC 是把机床数字控制这个总任务划分为子任务(也称为子功能模块)。在硬件方面,以多个微处理机配以相应的接口形成多个子系统,把划分的子任务分配给不同的子系统承担,由各子系统之间的协调动作完成数控。在多微处理机的结构中,有两个或两个以上的微处理机构成的子系统,子系统之间采用紧耦合,有集中的操作系统,共享资源;或者有两个或两个以上的微处理机构成的功能模块,功能模块之间采用松耦合,有多重操作系统有效地实现并行处理。

1) 多微处理器系统的特点

(1) 计算处理速度快。多微处理机结构中的每一个微处理器完成系统中指定的一部分

功能,独立执行程序,并行运行,比单微处理机提高了计算处理速度。它适应多轴控制、高进给速度、高精度、高效率的数控要求。由于系统共享资源,性能价格比也较高。

(2)可靠性高。由于系统中每个微处理器分管各自的任务,形成若干模块插件,模块更换方便,可使故障对系统影响减到最小。共享资源省去了重复机构,不但降低造价,也提高了可靠性。

(3)有良好的适应性和扩展性。多微处理机的 CNC 装置大都采用模块化结构。

(4)硬件易于组织规模生产。一般硬件是通用的,容易配置,只要开发新软件就可构成不同的 CNC 装置,便于组织硬件规模生产,保证质量,形成批量。

2)多微处理器系统的基本功能模块

多微处理器交流的主要基本功能模块有以下几种:

(1)CNC 管理模块。负责管理和组织整个 CNC 系统的工作,如系统的初始化、中断管理、总线仲裁、系统软硬件诊断等。

(2)存储器模块。包括主存储器(用于存放程序和数据,是各功能模块之间进行数据传送的共享存储器)和局部存储器(在每个 CPU 控制模块中)。

(3)CNC 插补模块。该模块可进行零件程序的译码、刀补计算、坐标位移量的计算、进给速度处理等插补前的预处理,然后进行插补计算,为各个坐标轴提供位置给定量。

(4)位置控制模块。其作用是将插补后的坐标位置给定值与反馈的位置实际值进行比较,并自动加减速,回基准点,监控伺服系统滞后量,补偿漂移,最后得到速度控制的模拟电压,以驱动进给电机。

(5)数据输入输出和显示模块。主要是用于输入、输出、显示零件加工程序、各种参数和数据、各种操作命令所需的接口电路。

(6)PLC 模块。其作用是对零件加工程序中的辅助功能代码、由机床传来的信号进行逻辑处理,从而实现各功能和操作方式之间的连接,机床电器设备的启停,刀具交换,工件数量和运转时间的计算等。

3)多微处理器 CNC 装置的典型结构

在多微处理机组成的 CNC 装置中,由 CNC 管理模块、CNC 插补模块、位置控制模块、PLC 模块、数据输入输出和显示模块、存储器模块这 6 种功能模块组成,若需要扩充功能,再增加相应的模块。这些模块之间互连与通信是在机柜内耦合,典型的有以下两种:

(1)共享总线结构。在共享总线结构中,将各功能模块插在配有总线插座的机框内,由系统总线把各个模块有效地连接在一起,按照要求交换各种控制指令和数据,实现各种预定的功能。如图 2-6 所示为多微处理机共享总线结构。

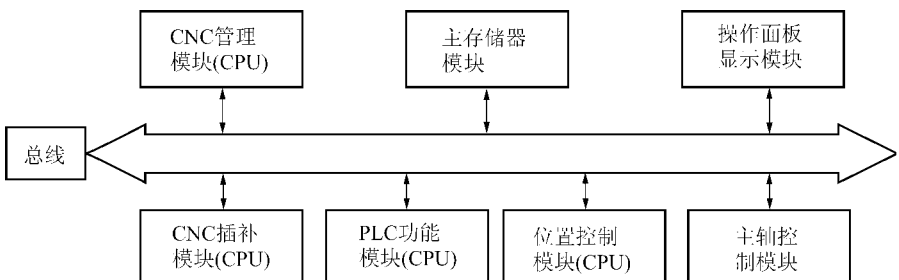


图 2-6 多微处理器共享总线结构框图

共享总线结构的优点是配置灵活、结构简单、容易实现、造价低。不足之处是会引起“竞争”(多个主模块同时请求)而导致信息传输率低,总线一旦出问题会影响全局。

(2)共享存储器结构。这种多微处理机结构,采用多端口存储器来实现各微处理机之间的互联和通信。由多端口控制逻辑电路来解决访问冲突。由于同一时刻只能有一个微处理机对多端口存储器读或写,所以功能复杂而要求微处理机数量增多时,会因争用共享而造成信息传输的阻塞,降低系统效率,因此扩展功能很困难。

如图 2-7 所示是一个双端口存储器结构框图,它配有两套数据、地址和控制线,可供两个端口访问,访问优先权预先安排好。两个端口同时访问时,由内部硬件裁决其中一个端口优先访问。

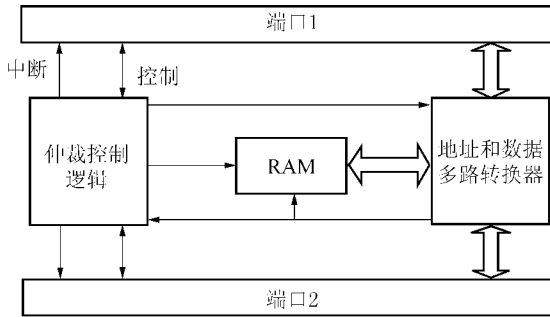


图 2-7 双端口存储器结构框图

如图 2-8 所示是多微处理机共享存储器结构框图。在这种多微处理器结构,采用多端口存储器来实现各微处理器之间的互连和通信,每个端口都配有一套数据、地址、控制线,以供端口访问。由专门的多端口控制逻辑电路解决访问的冲突问题。当微处理器数量增多时,往往会由于争用共享而造成信息传输的阻塞,降低系统效率。

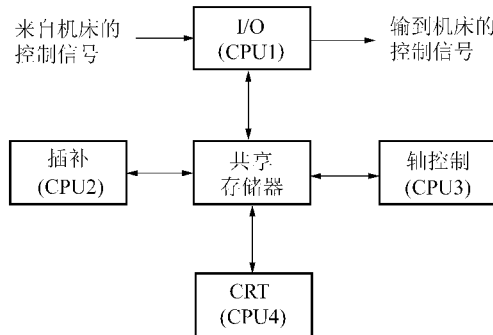


图 2-8 多微处理机共享存储器结构框图

如图 2-9 所示是 GE 公司的 MTC1 数控系统的机构框图,这是一种共享存储器型结构,它采用多端口存储器实现各微处理机之间的互连和通讯。功能模块之间通过公用的存储器连接耦合在一起。CPU1 为中央处理器,其任务是控制数控程序的编辑、译码、刀具和机床参数的输入。此外,作为主处理器它还控制 CPU2、CPU3,并与之交换信息。CPU2 为显示处理器、键盘管理、倍率开关。CPU3 作为插补处理器。

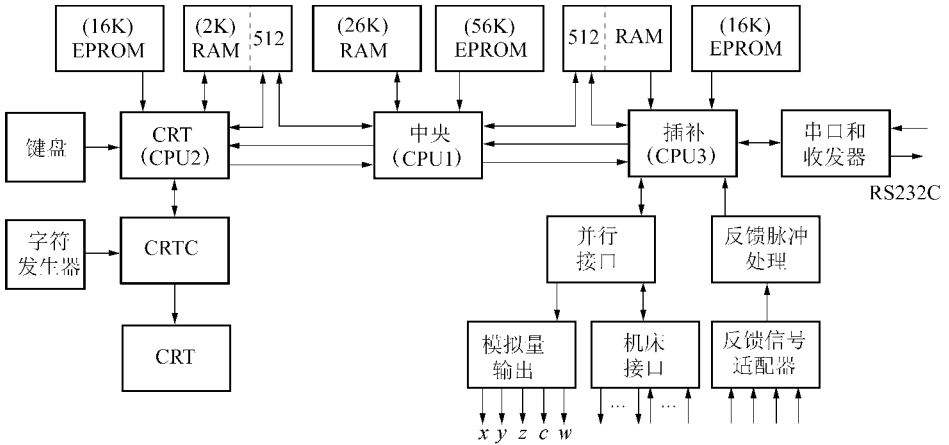


图 2-9 共享存储器型结构

2.2.3 硬件各组成部分的功能和原理

1. 微处理器和总线

微处理器是微机处理系统的中枢，系统都要在它的指挥下协调工作。微机主要由微处理器(CPU)、内部存储器和 I/O 接口电路组成。这三部分由数据总线、地址总线、控制总线的信号线连接，微处理的工作是给予一个基准时钟，基准时钟频率越高，其运算速度就越快。常用的微处理器数据宽度为 8 位、16 位、32 位，相应的微机称为 8 位机、16 位机和 32 位机。

微处理器由控制器和运算器组成。控制器主要完成控制任务，运算器主要完成算术运算、逻辑运算。微处理器选择要根据 CNC 装置进行实时控制和处理速度的要求，并考虑 CPU 在字长、寻址能力、运算速度方面的性能。

总线是由一组传送数字信息的物理导线组成。系统总线上传送的信息包括数据信息、地址信息、控制信息，因此，系统总线包含有三种不同功能的总线，即数据总线 DB(Data Bus)、地址总线 AB(Address Bus)和控制总线 CB(Control Bus)。

数据总线 DB 用于传送数据信息。数据总线是双向三态形式的总线，它既可以把 CPU 的数据传送到存储器或 I/O 接口等其他部件，也可以将其他部件的数据传送到 CPU。数据总线的位数是微型计算机的一个重要指标，通常与微处理的字长相一致。如 Intel 8086 微处理器字长 16 位，其数据总线宽度也是 16 位。需要指出的是，数据的含义是广义的，它可以是真正的数据，也可以指令代码或状态信息，有时甚至是一个控制信息。因此，在实际工作中，数据总线上传送的并不一定仅仅是真正意义上的数据。

地址总线 AB 是专门用来传送地址的，由于地址只能从 CPU 传向外部存储器或 I/O 端口，所以地址总线总是单向三态的，这与数据总线不同。地址总线的位数决定了 CPU 可直接寻址的内存空间大小，比如 8 位微机的地址总线为 16 位，则其最大可寻址空间为 $2^{16} = 64\text{KB}$ ，16 位微型机的地址总线为 20 型位，其可寻址空间为 $2^{20} = 1\text{MB}$ 。一般来说，若地址总线为 n 位，则可寻址空间为 2^n 字节。

控制总线 CB 用来传送控制信号和时序信号。控制信号中，有的是微处理器送往存储器和 I/O 接口电路的，如读/写信号、片选信号、中断响应信号等；也有的是其他部件反馈给

CPU 的,如中断申请信号、复位信号、总线请求信号、限备就绪信号等。因此,控制总线的传送方向由具体控制信号而定,一般是双向的,控制总线的位数要根据系统的实际控制需要而定。实际上控制总线的具体情况主要取决于 CPU。

2. 存储器

按存取速度和用途可把存储器分为两大类:把具有一定容量,存取速度快的存储器称为内部存储器,简称为内存,它是计算机的重要组成部分,CPU 可对它进行访问;把存储容量大而速度较慢的存储器称为外部存储器,简称为外存,在微型计算机中常见的外存有软磁盘、硬磁盘、盒式磁盘等。近年来,由于多媒体计算机的发展,普遍采用了光盘存储器。外存容量很大,如 CD-ROM 光盘可达 650 MB,硬盘可达到 320 GB 的容量(1 GB=1024 MB),而且容量还在增加,故也称外存为海量存储器。不过,要配备专门的设备才能完成对外存的读写功能,如软盘和硬盘要配有驱动器,磁带要有磁带机。通常将外存归入到计算机外部设备一类,它所存放的信息调入内存后 CPU 才能使用。

早期的内存使用磁芯,随着大规模集成电路的发展,半导体存储器集成度大大提高,成本迅速降低,存取速度大大加快,所以在微型计算机中,内存一般都使用半导体存储器。

半导体存储器从制造工艺的角度可分为双极型、CMOS 型、HMOS 型等;从应用角度可将其分为只读存储器(ROM)和随机读写存储器(RAM)两大类。如图 2-10 所示为半导体存储器的分类。

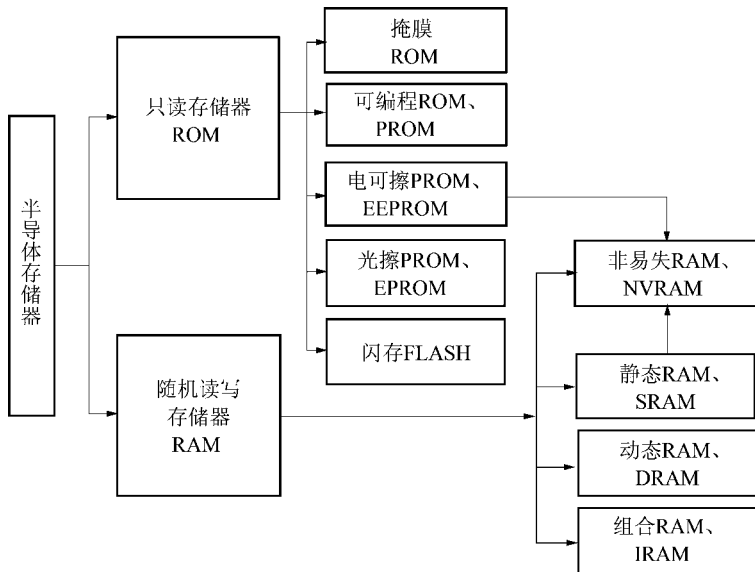


图 2-10 半导体存储器的分类

1) 只读存储器(ROM)

只读存储器是在使用过程中,只能读出存储的信息而不能用通常的方法将信息写入的存储器,其中又可分为以下几种:

(1)掩膜 ROM。利用掩膜工艺制造,一旦做好,不能更改,因此只适合于存储成熟的固定程序和数据。工厂大量生产时,成本很低。

(2)可编程 ROM,简称为 PROM(Programmable ROM)。由厂家生产出的“空白”存储器,根据用户需要,利用特殊方法写入程序和数据,即对存储器进行编程,但只能写入一次,

写入后信息是固定的,不能更改,它类似于掩膜 ROM,适合于批量使用。

(3)可擦除的 PROM,简称为 EPROM(Erasable Programmable ROM)。这种存储器可由用户按规定的方法多次编程,如编程之后想修改,可用紫外线灯制作的擦抹器照射 20 分钟左右,使存储器复原,用户可再编程,这对于研制和开发特别有利,因此应用十分广泛。

(4)电擦除的 PROM,简称为 EEPROM 或 E² PROM(Electrically Erasable PROM)。这种存储器的特点是能以字节为单位擦除和改写,而且不须把芯片拔下插入编程器编程,在用户系统即可进行。随着技术的进步,EEPROM 的擦写速度将不断加快,可作为不易失的 RAM 使用。除了并行读取的 EEPROM 外,现在也经常使用串行的 EEPROM。

2) 随机读写存储器(RAM)

随机读写存储器(RAM)是在使用过程中利用程序随时可写入信息,又可随时读出信息的存储器,分为 bipolar 型和 MOS 型两种。前者读写速度快,但功耗大,集成度低,故在微型计算机中几乎都用后者。MOS 又可分为以下 3 种:

(1)静态 RAM,即 SRAM(Static RAM)。其存储电路以双稳态触发器为基础,状态稳定,只要不掉电,信息就不会丢失。其优点是不需刷新,缺点是集成度低。适于不需要大存储容量的微型计算机,例如单板机和单片机组成的嵌入式系统。

(2)动态 RAM,即 DRAM(Dynamic RAM)。其存储单元以电容为基础,电路简单,集成度高。但也存在问题,即电容中电荷由于漏电会逐渐丢失,因此 DRAM 必须定时刷新。它适于大存储容量的计算机。80x86 计算机中的内存条就是用这种 DRAM 制作的。

(3)非易失 RAM(掉电自保护 RAM),即 NVRAM(Non-Volatile RAM)。这种 RAM 是由 SRAM 和 EEPROM 共同构成的存储器,正常运行时和 SRAM 一样,而在掉电或电源有故障的瞬间,它把 SRAM 的信息保存在 EEPROM 中,从而使信息不会丢失,NVRAM 多用于存储非常重要的信息和掉电保护。

3) 存储器的指标

衡量半导体存储器的指标很多,诸如可靠性、功耗、价格、电源种类等,但从接口电路来看,最重要的指标是存储器芯片的容量和存取速度。

存储器芯片的容量是以存储 1 位(bit) 二进制数为单位的,因此存储器的容量即指每个存储器芯片所能存储的二进制数的位数。如 1024 位/片,即指芯片内集成了 1024 位的存储器。由于在微型计算机中,数据大都是以字节(Byte)为单位并行传送的,同样,对存储器的读写也是以字节为单位寻址的。然而存储器芯片因为要适用于 1 位、4 位、8 位计算机的需要,或因工艺上的原因,其数据线也有 1 位、4 位、8 位之分。例如,Intel 2116 为 1 位,2114 为 4 位,6264 为 8 位,所以在标定存储器容量时,经常同时标出存储单元的数目和位数,因此存储器芯片容量=单元数×数据线位数。

如 Intel 2114 芯片容量为 1 KB×4 位/片,6264 为 8 KB×8 位/片。虽然微型计算机的字长已经达到 16 位、32 位甚至 64 位,但其内存仍以一个字作为一个单元,不过在这种微型计算机中,一次可同时访问 2、4、8 个单元。

存储器芯片的存取速度是用存取时间来衡量的。它是指从 CPU 给出有效的存储器地址到存储器给出有效数据所需要的时间。存取时间越短,则速度越快。超高速存储器的存取时间已小于 20 ns,中速存储器在 100~200 ns 之间,低速存储器的存取时间在 300 ns 以上。现在 80586 CPU 时钟已达 100 MHz 以上,这说明存储器的存取速度已非常高。随着半导体技术的进步,存储器的容量越来越大,速度越来越高,而体积却越来越小。

3. I/O 接口

微型计算机上的所有部件都是通过总线互联的,外部设备也不例外。I/O 接口就是将外部设备连接到总线上的一组逻辑电路的总称,也称为外设接口。在一个实际的计算机控制系统中,CPU 与外部设备之间常需要进行频繁的信息交换,包括数据的输入输出、外部设备状态信息的读取及控制命令的传送等,这些都是通过接口来实现的。

CPU 通过接口与外部设备的连接示意图如图 2-11 所示。通过接口传送的除数据外,还有反映当前外设工作状态的状态信息以及 CPU 向外设发出的各种控制信息。

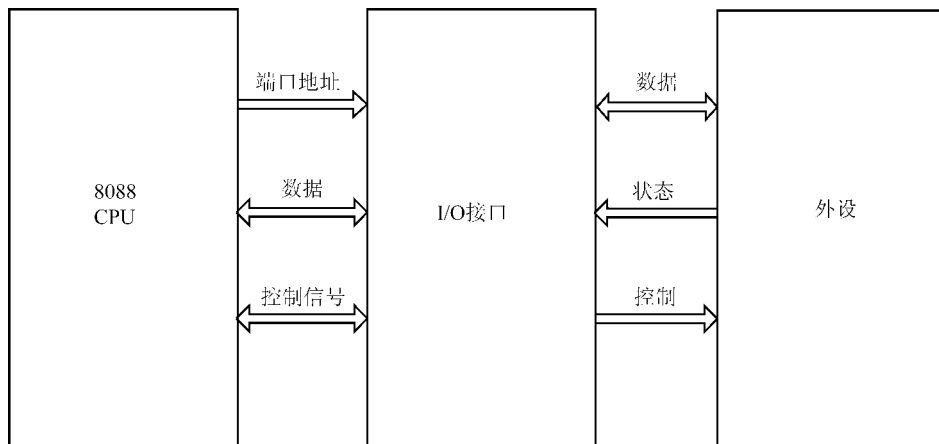


图 2-11 CPU 与外设之间的接口

负责把信息从外部设备送入 CPU 的接口叫做输入接口(端口),而将信息从 CPU 输出到外部设备的接口则称为输出接口(端口)。在需要从外设输入数据时,由于外设处理数据的时间一般要比 CPU 长得多,数据在外部总线上保持的时间相对较长,所以要求输入接口必须要具有对数据的控制能力。即要在外部数据准备好,CPU 可以读取时才将数据送到系统数据总线。

若外设具有数据保持能力,通常可以仅用一个三态门缓冲器作为输入接口,当其控制端信号有效时,三态门导通,该外设就与数据总线连通,CPU 将外设准备好的数据读入;当其控制端信号无效时,三态门断开,该外设就从数据总线脱离,数据总线又可用于其他信息的传送。

在数据输出时,同样由于外设的速度比较慢,要使数据能正确写入外设,CPU 输出的数据一定要能够保持一段时间。如果这个“保持”的工作由 CPU 来完成,则对其资源就必然是个浪费。实际上,CPU 送到总线上的数据只能保持几微秒,因此,要求输出接口必须要具有数据的锁存能力,CPU 输出的数据通过总线送入接口锁存,由接口将数据一直保持到被外设取走。简单的输出接口一般由锁存器构成。

以上三态门和锁存器的控制端一般与 I/O 地址译码输出信号线相连,当 CPU 执行 I/O 指令时,指令中指定的 I/O 地址经译码后即可使控制信号有效,打开三态门(对外设读时)或将数据锁入锁存器(对外设写时)。

一个典型的三态门芯片 74LS244 如图 2-12 所示。

该芯片由 8 个三态门构成,74LS244 有两个控制端: $\overline{E_1}$ 和 $\overline{E_2}$ 。每个控制端各控制 4 个三态门。当某一控制端有效(低电平)时,相应的 4 个三态门导通;否则,相应的三态门呈现高阻状态(断开)。实际使用中,通常是将两个控制端并联,这样就可用一个控制信号来使 8 个三态门同时导通或同时断开。由于三态门具有“通断”控制能力的这个特点,所以可利用其

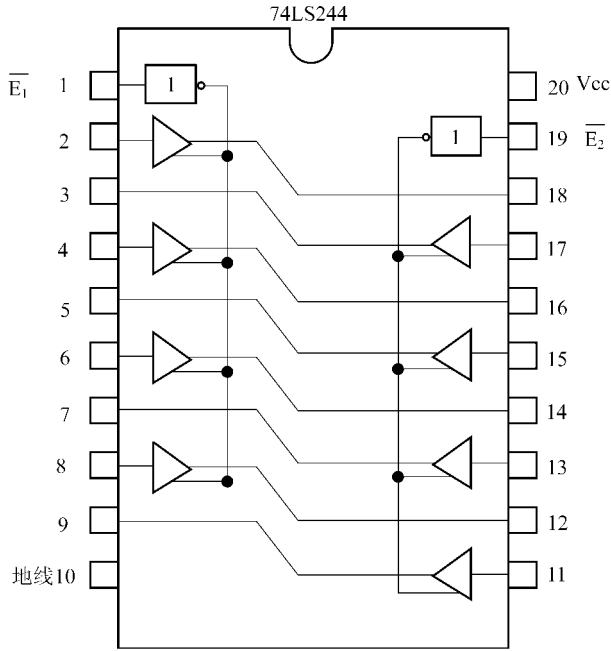


图 2-12 74LS244 芯片引脚图

作为输入接口。利用三态门作为输入信号接口时,要求信号的状态是能够保持的。这是因为三态门本身没有对信号的保持或锁存能力。如图 2-13 所示是一个利用三态门 74LS244 作为开关量输入接口的例子。

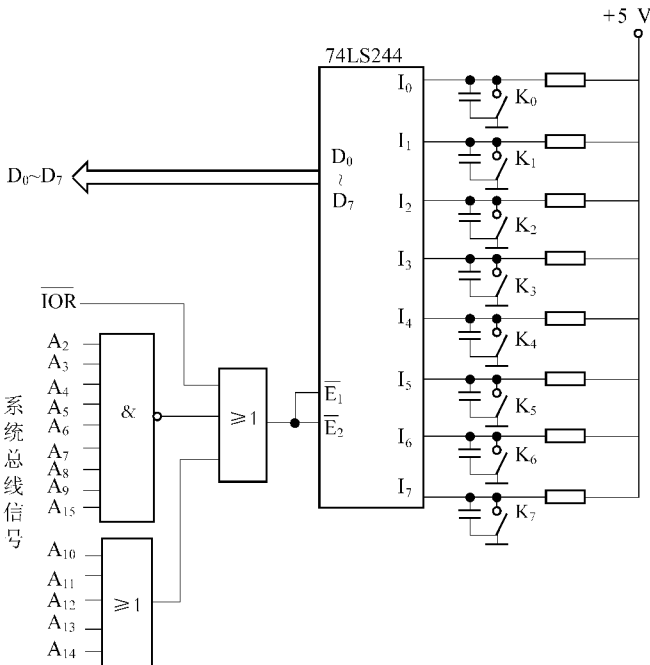


图 2-13 三态门作为输入接口

74LS244 的输入端接有 8 个开关 K_0, K_1, \dots, K_7 。当 CPU 读该接口时,总线上的 16 位地址信号通过译码使 $\overline{E_1}$ 和 $\overline{E_2}$ 有效,三态门导通,8 个开关的状态经数据线 $D_0 \sim D_7$ 被读入到

CPU 中,这样,就可测量出这些开关当前的状态是打开的还是闭合的。当 CPU 不读此接口地址时, $\overline{E_1}$ 和 $\overline{E_2}$ 为高电平,则三态门的输出为高阻状态,使其与数据总线断开。

用一片 74LS244 芯片作为输入接口最多可以连接 8 个开关或其他具有信号保持能力的外设。当然也可只接一个外设而让其他端悬空,对空着未用的端,其对应位的数据是任意值,在程序中常用逻辑“与”指令将其屏蔽掉。

如果有更多的开关状态(或其他外设)需要输入时,可用类似的方法用两片或更多的芯片并联使用。

74LS244 芯片除用作输入接口外,还常用来作为信号的驱动器。

编写程序判断如图 2-13 所示的开关的状态。如果所有的开关都闭合,则程序转向标号为 NEXT1 的程序段执行,否则转向标号为 NEXT2 的程序段执行。

作为输入接口的三态门 74LS244,其 I/O 地址采用了部分地址译码,地址线 A_1 和 A_0 未参加译码,所以它所占用的地址为 83FCH~83FFH。我们可以用其中任何一个地址,而其他重叠的 3 个地址空着不用。另外,当开关闭合时输入低电平(=0),程序段如下:

```
MOV  DX,83FCH
IN   AL,DX
AND  AL,0FFH
JZ   NEXT1
JMP  NEXT2
```

可见,利用三态门作为输入接口,使用和连接都是很容易的。由于三态门器件没有数据的保持能力,所以它一般只用作输入接口,不能直接用作数据输出接口。数据输出接口通常是用具有信息存储能力的双稳态触发器来实现。最简单的输出接口可用 D 触发器构成。如常用的锁存器 74LS273,它内部包含了 8 个 D 触发器。其引线图及真值表如图 2-14 所示。

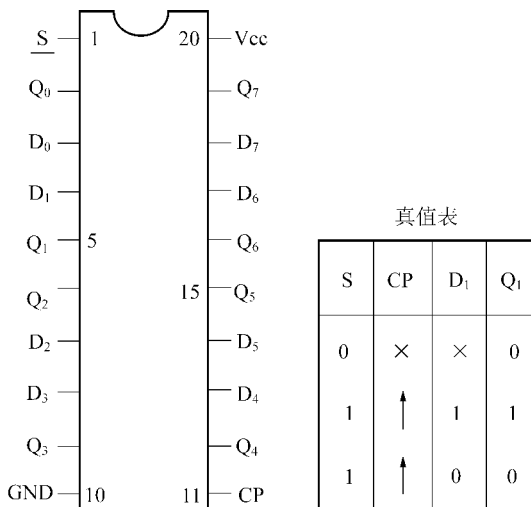


图 2-14 74LS273 引线图和真值表

74LS273 常用来作为并行输出接口。另外,使用其中的某一个 D 触发器也可通过软件编程实现简单的串行输出。

如图 2-15 所示为应用 74LS273 作为输出接口的例子。8 个 Q 端与 8 个发光二极管相连接,要使接到 Q_0 端和 Q_6 端的发光二极管发光,其对应的 Q_0 、 Q_6 端须为“1”状态,而其他

Q 端则为“0”状态。

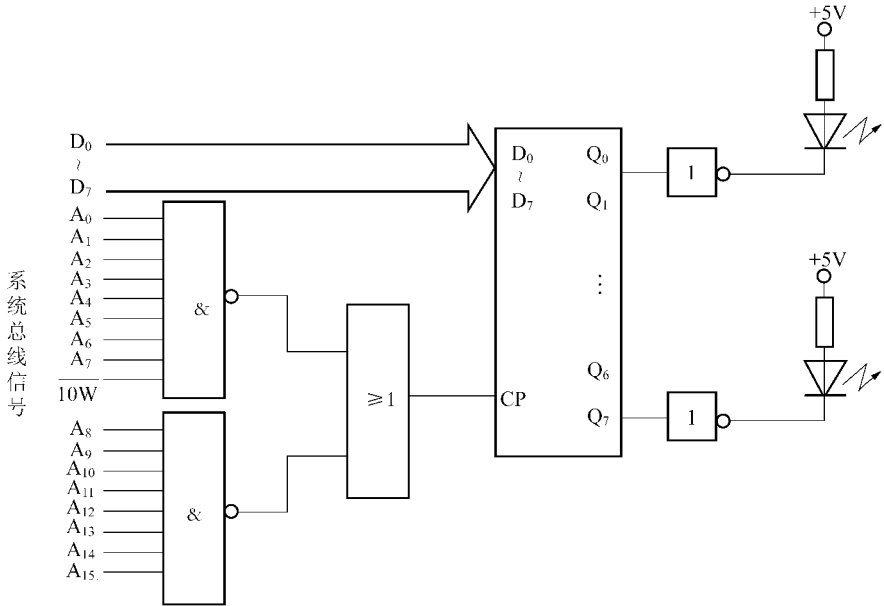


图 2-15 74LS273 作为输出接口

假定该输出接口的地址为 0FFFFH, 则程序段如下:

```
MOV DX,0FFFFH
MOV AL,01000001B
OUT DX,AL
```

74LS273 的数据锁存输出端 Q 是通过一个一般的门(二态门)输出的。也就是说,只要 74LS273 正常工作,其 Q 端总有一个确定的逻辑状态(0 或 1)输出。因此,74LS273 无法直接用作输入接口,即它的 Q 端绝对不允许直接与系统的数据总线相连接。也有既可作输入接口又能作输出接口的芯片,如下面介绍的带有三态输出的锁存器 74LS374。这也是经常用到的一种电路芯片,其引线图和真值表如图 2-16 所示。

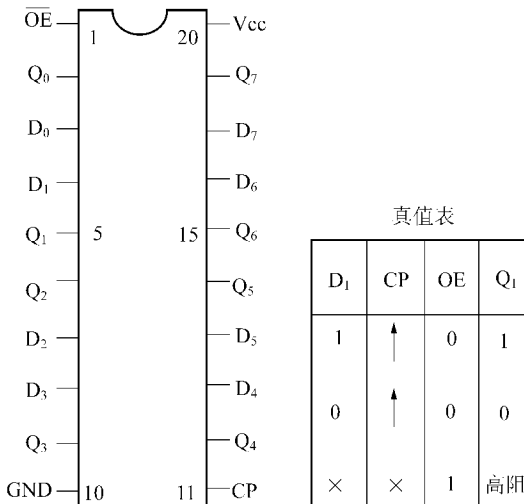


图 2-16 74LS374 引线图和真值表

从引线上可以看出,它比 74LS273 多了一个输出允许端 \overline{OE} 。只有当 $\overline{OE}=0$ 时,74LS374 的输出三态门才导通。当 $\overline{OE}=1$ 时,则呈高阻状态。如图 2-17 所示为 74LS374 中一个锁存器的结构图,由图可知,74LS374 在 D 触发器输出端加有一个三态门。

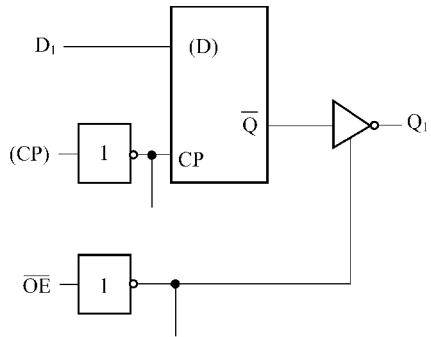


图 2-17 74LS374 内部结构

74LS374 在用作输入接口时,端口地址信号经译码电路接到 \overline{OE} 端,外设数据由外设提供的选通脉冲锁存在 74LS374 内部。当 CPU 读该接口时,译码器输出低电平,使 74LS374 的输出三态门打开,读出外设的数据;如果用作输出接口,也可将 OE 端接地,使其输出三态门一直处于导通状态,这样就与 74LS273 一样使用了。

用 74LS374 作为输入和输出接口的电路如图 2-18 所示。

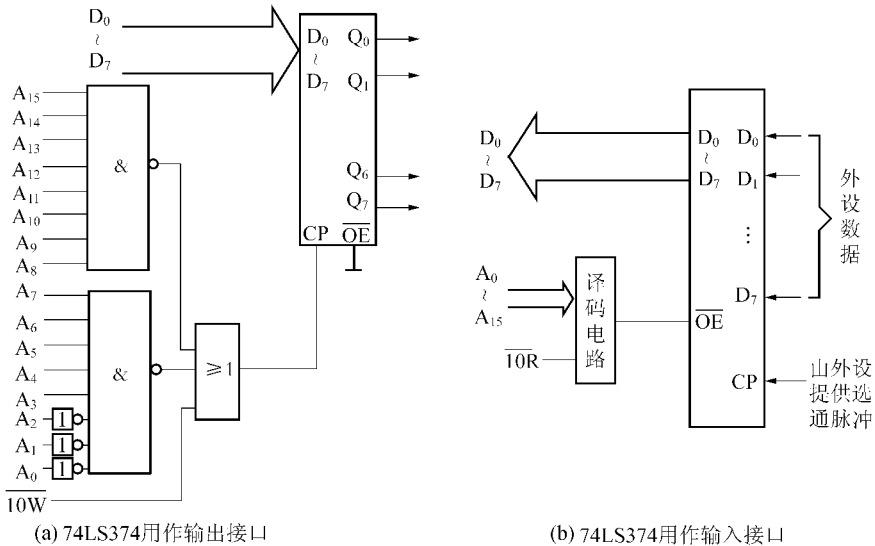


图 2-18 74LS374 用作输入和输出接口

另外还有一种常用的带有三态门的锁存器芯片 74LS373,它与 74LS374 在结构和功能上完全一样,区别是数据锁存的时机不同,带有三态门基片 74LS373 是在 CP 脉冲的高电平期间将数据锁存。

总之,简单接口电路芯片在构造上比较简单,使用也很方便,常作为一些功能简单的外部设备的接口电路,但由于它们的功能有限,对较复杂的功能要求就难以胜任。

1) I/O 接口需要解决的问题

外部设备的种类繁多,有机械式、电动式、电子式和其他形式。它们涉及的信息类型也不相同,可以是数字量、模拟量或开关量。因此,CPU 与外设之间交换信息时需要解决以下问题:

(1)速度匹配问题。CPU 的速度很高,而外设的速度有高有低,而且不同的外设速度差异甚大。

(2)信号电平和驱动能力问题。CPU 的信号都是 TTL 电平(一般在 0~5 V 之间),而且提供的功率很小,而外设需要的电平要比这个范围宽得多,需要的驱动功率也较大。

(3)信号形式匹配问题。CPU 只能处理数字信号,而外设的信号形式多种多样,有数字

量、开关量、模拟量(电流、电压、频率、相位),甚至还有非电量,如压力、流量、温度、速度等。

(4)信息格式问题。CPU在系统总线传送的是8位、16位或32位并行二进制数据,而外设使用的信号形式、信息格式各不相同。有些外设是数字量或开关量,而有些外设使用的是模拟量;有些外设采用电流量,而有些是电压量;有些外设采用并行数据,而有些则是串行数据。

(5)时序匹配问题。CPU的各种操作都是在统一的时钟信号作用下完成的,各种操作都有自己的总线周期,而各种外设也有自己的定时与控制逻辑,大都与CPU时序不一致。因此各种各样的外设不能直接与CPU的系统总线相连。上述问题是通过在CPU与外设之间设置相应的I/O接口电路来予以解决的。

2) I/O接口的功能

(1)I/O地址译码与设备选择。所有外设都通过I/O接口挂接在系统总线上,在同一时刻,总线只允许一个外设与CPU进行数据传送。因此,只有通过地址译码选中的I/O接口允许与总线相通,而未被选中的I/O接口呈现为高阻状态,与总线隔离。

(2)信息的输入输出。通过I/O接口,CPU可以从外部设备输入各种信息,也可将处理结果输出到外设;CPU可以控制I/O接口的工作(向I/O接口写入命令),还可以随时监测与管理I/O接口和外设的工作状态;必要时,I/O接口还可以通过接口向CPU发出中断请求。

(3)命令、数据和状态的缓冲与锁存。因为CPU与外设之间的时序和速度差异很大,为了能够确保计算机和外设之间可靠地进行信息传送,要求接口电路应具有信息缓冲能力。接口不仅应缓存CPU送给外设的信息,也要缓存外设送给CPU的信息,以实现CPU与外设之间信息交换的同步。

(4)信息转换。I/O接口还要实现信息格式变换、电平转换、码制转换、传送管理以及联络控制等功能。

3) 数控机床上的接口规范

按ISO4336-1982(E)标准,数控机床上的接口分为4类:

- (1)与驱动命令有关的连接电路。
- (2)数控系统与检测系统和测量传感器的连接电路。
- (3)电源及保护电路。
- (4)通断信号和代码信号连接电路。

其中,前两类接口传送的是数控系统与伺服驱动单元(即速度控制环)、伺服电机、位置检测和速度检测之间的控制信息及反馈信息,它们属于数字控制、伺服控制及检测控制。

电源及保护电路由数控机床强电线路中的电源控制电路构成。强电线路由电源变压器、控制变压器、各种断路器、保护开关、接触器、功率继电器及熔断器等连接而成。强电线路的作用是为驱动单元、主轴电机、辅助电机、电磁铁、电磁阀、离合器等功率执行元件供电。强电线路不能与低压下工作的控制电路或弱电线路直接连接,只能通过中间继电器、断路器、热动开关等器件转换成在直流低压下工作的触点的开、合动作,才能成为继电器逻辑电路和PLC可接收的电信号。反之,由CNC系统输出的信号,应先去驱动小型中间继电器,然后用中间继电器的触点接通强电线路中的功率继电器/接触器,从而接通主回路(强电回路)。

通断信号和代码信号连接电路是CNC系统与机床参考点、限位、面板开关等的连接信号。当数控机床没有PLC时,上述信号在CNC装置和机床之间直接传送。当数控机床有PLC时,上述信号除极少数的高速信号外,均通过PLC传送。

4) I/O 信号的分类

从机床(MT)向 CNC 装置传送的信号称为输入信号;从 CNC 装置向机床(MT)传送的信号称为输出信号。I/O 信号的类型有:

(1) 直流数字量输入/输出信号。

(2) 直流模拟量输入/输出信号。主要用于进给坐标轴、主轴的伺服控制,或其他接收、发送模拟量信号的设备。

(3) 交流输入/输出信号。主要用于直接控制功率执行器件。

后两类信号传送需要专门的接口电路。在实际应用中,一般都用 PLC,并配置专门的接口模板或插座才能实现。

5) I/O 接口电路的任务

(1) 进行电平转换和功率放大。由于 CNC 装置内部信号为 TTL 电平,而机床设备信号是非 TTL 电平,因此两者需要进行电平转换,在重负载情况下,还要进行功率放大。

(2) 防止噪声引起的误动作。一般用光电隔离器、继电器将 CNC 系统和机床之间的信号在电气上加以隔离。

(3) 进行模拟量和数字量的转换。由于 CNC 装置的微处理器只能处理数字量,对于需要模拟量控制的地方,需要 D/A(数/模)转换电路;反之将模拟量输入到 CNC 装置,需要 A/D(模/数)转换电路。

6) 数字量输入/输出接口

输入接口分为触点输入和电压输入。触点输入中又包含无源触点输入和有源触点输入。在触点输入时要采取措施消除触点抖动。常用的防抖动的方法是用施密特触发器或 R-S 触发器来整形。

输出接口分为继电器输出和无触点输出。负载可为指示灯或继电器。对于交流负载或大功率直流负载驱动输出时,一般要通过中间继电器进行“二次驱动”;当 CNC 装置中带有 PLC 装置时,并且 PLC 本身具有交流输入、输出接口,或有用于直流大负载驱动的专用接口时,输出信号就不必经过中间继电器过渡,即可直接驱动负载器件。

直流数字量输入/输出信号(开关量 I/O 信号)在 CNC 和机床之间的传送通过接口存储器进行。机床上各种 I/O 信号均在存储器中占有一位,该位的状态是 0、1,分别表示开/关或继电器通/断状态。CNC 装置中的 CPU 定时从接口存储器中回收状态,并由软件进行相应处理;同时又向接口电路输出各种控制命令,控制强电箱的动作。

4. MDI 接口

MDI 是指手动数据输入。在该方式下,可直接通过数控机床操作面板上的键盘,输入单段程序,或进行各种参数的设置和修改等操作。MDI 接口的任务之一是对键盘按键进行处理。在 CNC 装置中一般配有键盘扫描程序。

5. 位置控制器

位置控制器的作用主要是对数控机床的进给运动坐标轴的位置进行控制。如工作台的前、后、左、右移动,主轴箱的上、下移动,绕某一直线轴的旋转运动等。进给运动坐标轴的位置是数控机床上要求最高的位置控制,其不仅对单坐标轴的运动和位置精度有严格要求,而且在多轴联动时,要求各坐标轴有很好的动态配合。

位置控制器采用的硬件可以是大规模专用集成电路位置控制芯片,如 FANUC 公司的 MB8720、MB8739、MB87103,也可以是位置控制模块,如 SIEMENS 公司的 MS230、MS250、

MS300 等。如图 2-19 所示为位置控制模块结构框图。

位置控制模块由软件和硬件两部分组成。软件部分包括跟随误差计算和进给速度指令数值的计算；硬件部分包括位置控制输出组件(其功能是接收进给指令,进行 D/A 转换,为速度单元提供指令电压)和位置测量组件(其功能是接收并处理反馈信号,送至跟随误差计数器,与位移指令值进行比较,得到进给指令)。

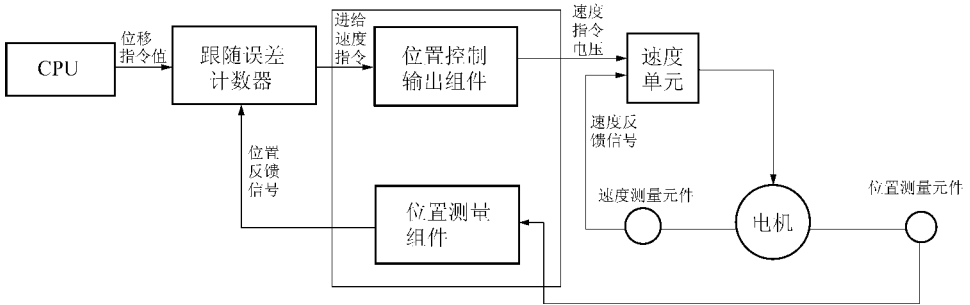


图 2-19 位置控制模块结构框图

2.3 CNC 系统软件

CNC 装置的软件是为完成数控系统的各项功能而专门设计的一种专用软件,又称为系统软件(或系统程序),其管理作用类似于计算机操作系统的功能。CNC 装置的软件结构取决于硬件和软件的分工及软件本身的工作特点。为满足制造业发展的要求,一些厂商生产的 CNC 装置,硬件设计好后基本不变,而软件功能不断升级,这也是 CNC 装置灵活性的体现。

2.3.1 CNC 系统软件的结构与分类

1. CNC 系统软硬件组合类型

CNC 装置由软件和硬件组成。在信息的处理方面,两者在逻辑上是等价的。一些由硬件完成的任务可由软件完成;反之亦然。硬件和软件在实现同一功能时各有不同的特点:硬件处理速度较快,造价较高,实现复杂控制的功能比较困难,而软件设计灵活,适应性强,但处理速度较慢。在 CNC 装置中软件和硬件的功能如何分工,即如何划分软件和硬件的功能界面,有如图 2-20 所示的几种方案。

当今数控技术发展趋势是用相对较少且标准化程度较高的硬件,配以功能丰富的软件模块构成 CNC 装置。CNC 系统中实时性要求最高的任务就是插补和位控,即在一个采样周期中必须完成控制策略的计算,而且还要留有一定的时间去做其他的事。CNC 系统的插补器既可面向软件也可面向硬件。归结起来,主要有以下 3 种类型:

- (1)不用软件插补器,插补完全由硬件完成的 CNC 系统。
- (2)由软件插补器完成粗插补,由硬件插补器完成精插补的 CNC 系统。
- (3)带有完全用软件实施的插补器的 CNC 系统。

2. CNC 装置软件结构的特点

CNC 系统是一个专用的实时多任务系统,其中的 CNC 装置通常作为一个独立的过程

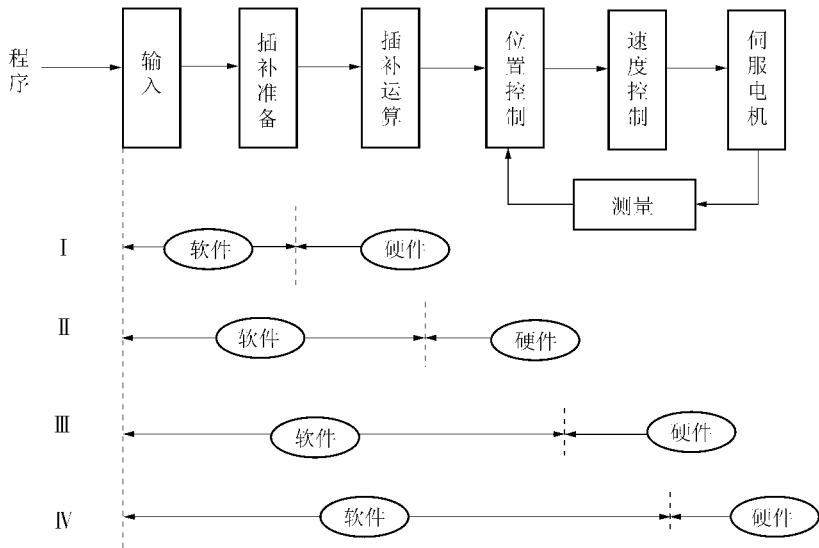


图 2-20 CNC 系统软硬件组合类型

控制单元用于控制各种对象，它的系统软件必须完成管理和控制两大任务。多任务并行处理和多重实时中断是 CNC 装置软件结构的两大特点。

1) 多任务并行处理

如图 2-21 所示为 CNC 系统的软件任务分解图，反映了数控系统软件的多任务性。CNC 系统通常作为一个独立的过程控制单元用于工业自动化生产中，因此它的系统软件必须完成管理和控制两大任务。系统的管理部分包括输入、I/O 处理、显示和诊断。系统的控制部分包括译码、刀具补偿、速度处理、插补和位置控制。在许多情况下，管理和控制的某些工作必须同时进行。如当 CNC 系统工作在加工控制状态时，为了使操作人员能及时地了解 CNC 系统的工作状态，管理软件中的显示模块必须与控制软件同时运行。当 CNC 系统工作在 NC 加工方式时，管理软件中的零件程序输入模块必须与控制软件同时运行。而当控制软件运行时，其本身的一些处理模块也必须同时运行。如为了保证加工过程的连续性，刀具需要在各程序段之间不停刀，译码、刀具补偿和速度处理模块必须与插补模块同时运行，而插补又必须与位置控制同时进行。

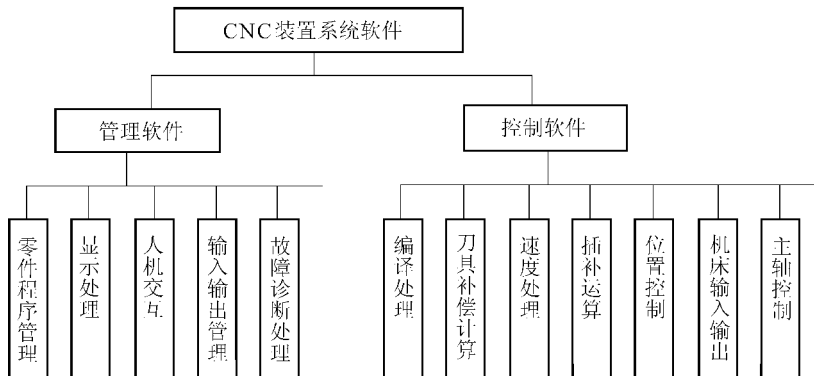


图 2-21 CNC 系统的软件任务分解图

并行处理是指计算机在同一时刻或同一时间间隔内完成两种或两种以上性质相同或不同的工作。并行处理最显著的优点是提高了运算速度。如图 2-22 所示为 CNC 系统并行处理图。拿 n 位串行运算和 n 位并行运算来比较,在元件处理速度相同的情况下,后者运算速度几乎提高为前者的 n 倍。这是一种资源重复的并行处理方法,它是根据“以数量取胜”的原则大幅度提高运算速度的。但是并行处理还不仅仅是设备的简单重复,它还有更多的含义,如时间重叠和资源共享。所谓时间重叠是根据流水线处理技术,使多个处理过程在时间上相互错开,轮流使用同一套设备的几个部分。而资源共享则是根据“分时共享”的原则,使多个用户按时间顺序使用同一套设备。

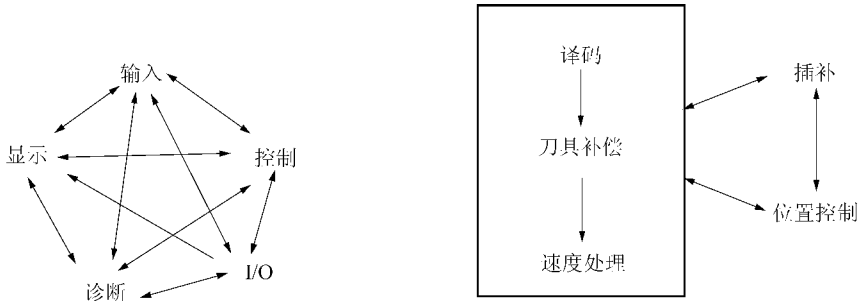


图 2-22 CNC 系统并行处理图

目前在 CNC 系统的硬件设计中,已广泛使用资源重复的并行处理方法,如采用多 CPU 的系统体系结构来提高系统的速度。而在 CNC 系统的软件设计中则主要采用资源分时共享和资源重叠的流水线处理技术。

(1)资源分时共享。在单 CPU 的 CNC 系统中,主要采用 CPU 分时共享的原则来解决多任务的同时运行。一般来讲,在使用分时共享并行处理的计算机系统中,首先要解决的问题是各任务占用 CPU 时间的分配原则,这里有两方面的含义:其一是各任务何时占用 CPU;其二是允许各任务占用 CPU 的时间长短。在 CNC 系统中,对各任务使用 CPU 是用循环轮流和中断优先相结合的方法来解决。如图 2-23 所示为一个典型 CNC 系统各任务分时共享 CPU 的时间分配图。系统在完成初始化以后自动进入时间分配环中,在环中依次轮流处理各任务。而对于系统中一些实时性很强的任务则按优先级排队,分别放在不同中断

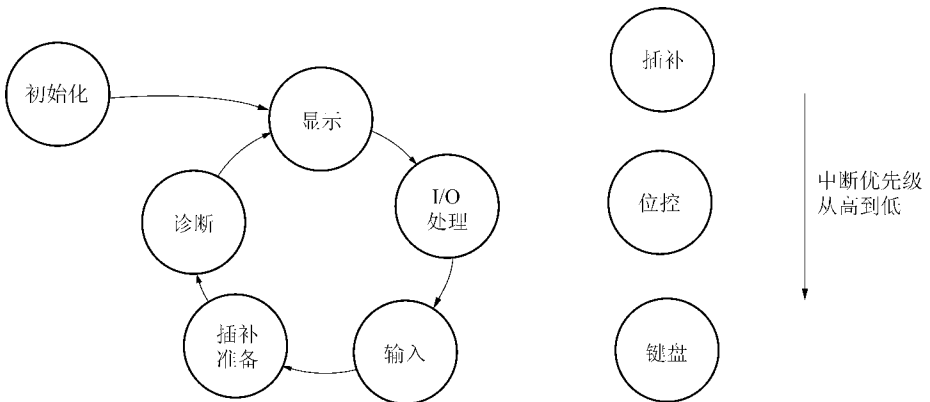


图 2-23 CPU 分时共享和中断优先

优先级上,环外的任务可以随时中断环内各任务的执行。每个任务允许占有 CPU 的时间受到一定限制,通常是这样处理的,对于某些占有 CPU 时间比较多的任务,如插补准备,可以在其中的某些地方设置断点,当程序运行到断点处时,自动让出 CPU,待到下一个运行时间里自动跳到断点处继续执行。

(2)资源重叠流水处理。当 CNC 系统处在 NC 工作方式时,其数据的转换过程将由零件程序输入、插补准备(包括译码、刀具补偿和速度处理)、插补、位置控制 4 个子过程组成。如果每个子过程的处理时间分别为 Δt_1 、 Δt_2 、 Δt_3 、 Δt_3 ,那么一个零件程序段的数据转换时间将是 $t = \Delta t_1, t = \Delta t_2, t = \Delta t_3, t = \Delta t_3$ 。如果以顺序方式处理每个零件程序段,即第一个零件程序段处理完以后再处理第二个程序段,以此类推,这种顺序处理方式的时间空间关系如图 2-24(a)所示。从图上可以看出,如果等到第一个程序段处理完之后才开始对第二个程序段进行处理,那么在两个程序段的输出之间将有一个时间长度为 t 的间隔。同样在第二个程序段与第三个程序段的输出之间也会有时间间隔,以此类推。这种时间间隔反映在电机上就是电机的时转时停,反映在刀具上就是刀具的时走时停。不管这种时间间隔多么小,这种时走时停在加工工艺上都是不允许的,消除这种间隔的方法是用流水处理技术,采用流水处理后的时间空间关系如图 2-24 (b)所示。

流水处理的关键是时间重叠,即在一段时间间隔内不是处理一个子过程,而是处理两个或更多的子过程。从图 2-24 (b)可以看出,经过流水处理后从时间 t_4 开始,每个程序段的输出之间不再间隔,从而保证了电机转动和刀具移动的连续性。流水处理要求每一个处理子程序的运算时间相等,而在 CNC 系统中每一个子程序所需的处理时间都是不相等的,解决的办法是取最长的子程序处理时间为处理时间间隔。这样当处理时间较短的子程序时,处理完成之后就进入等待状态。

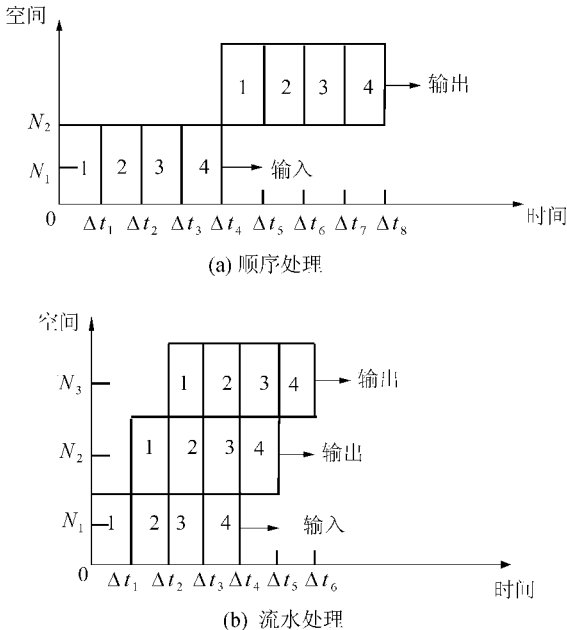


图 2-24 资源重叠流水处理

在单 CPU 的 CNC 装置中,流水处理的时间重叠只有宏观的意义,即在一段时间内,CPU 处理多个子程序,但从微观上看,各子程序分时占用 CPU 时间。

(3)并行处理中的信息交换和同步。并行处理中的信息交换是通过各种缓冲区(缓冲存储区)来实现的,缓冲存储区是在存储区中开辟的系统专用区域。在自动加工方式中,CNC装置通过缓冲区交换信息的情况如图 2-25 所示。各缓冲区数据交换和更新的同步是靠同步信号指针来实现的。

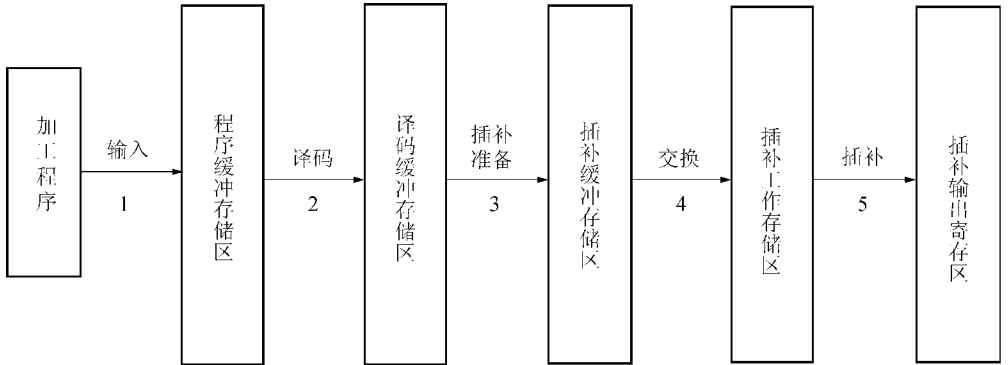


图 2-25 CNC 中的信息交换

2)实时中断处理

CNC 系统是一个专用的实时多任务系统,该系统中各任务的执行或强或弱都具有实时性要求,即任务必须在规定时间内完成或响应,否则将导致执行结果错误或系统故障。

实时性任务可以分为强实时性任务和弱实时性任务。在强实时性任务中,又有实时突发性任务和实时周期性任务。实时突发性任务是指任务的发生具有随机性和突发性,是一种异步中断事件,有很强的实时性要求,主要包括故障中断(急停、机械限位、硬件故障等)、机床 PLC 中断、硬件(按键)操作中断等。实时周期性任务是指任务是精确地按一定时间间隔发生,包括插补运算、位置控制等任务,这类任务处理的实时性是保证加工精度和加工过程的连续性,在任务的执行过程中,除系统故障外,不允许被其他任何任务中断。弱实时性任务是指这类任务的实时性要求相对较弱,只需要保证某一段时间内得以运行即可,如背景程序或优先级较低的任务,其中包括 CRT 显示、零件程序的编辑、加工状态和加工轨迹的动态显示、译码、刀补等。

CNC 系统的多任务性和实时性决定了中断是整个系统必不可少的重要组成部分。CNC 装置的中断管理主要靠硬件完成,而系统的中断结构决定了 CNC 装置软件的结构。CNC 装置的中断类型有:

- (1)外部中断,如纸带光电阅读机读孔中断、外部监控中断(如急停)、键盘操作面板输入中断等。
- (2)内部定时中断,如插补周期定时中断、位置采样定时中断等。
- (3)硬件故障中断,是指 CNC 装置各种硬件故障检测装置发出的中断,如存储器出错、定时器出错、插补运算超时等。
- (4)程序性中断,是指程序中出现的各种异常情况的报警中断,如各种溢出、除零等。

CNC 装置中断结构模式又称为 CNC 装置软件结构模式,是指系统软件的组织管理方式。系统软件通过一定的结构模式,来组织和协调各个任务的执行,使各任务满足一定的时序配合要求和逻辑关系,从而实现 CNC 系统的各种控制要求。CNC 装置软件结构模式主要有:

- (1)前后台型结构模式。如图 2-26 所示为前后台软件结构,该模式将 CNC 系统软件划

分成前台程序和后台程序两部分。前台程序主要完成插补运算、位置控制、故障处理等实时性很强的任务，它是一个实时中断服务程序。后台程序(又称背景程序)主要完成显示、零件加工程序的编辑管理、系统的输入/输出、插补预处理(译码、刀补处理、速度预处理)等弱实时性的任务，它是一个循环运行的程序。后台程序在运行过程中，不断地定时被前台实时中断程序打断，前后相互配合来完成零件的加工任务。其缺点是实时性较差。

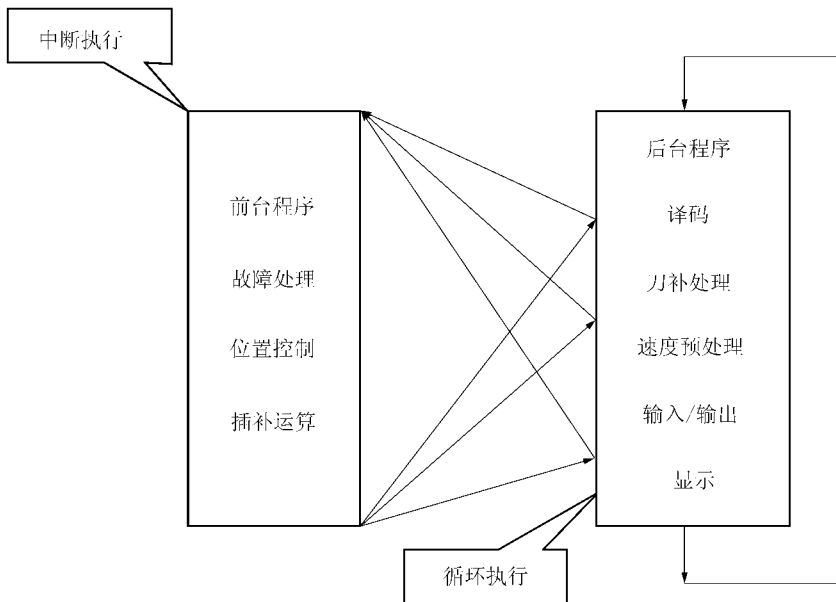


图 2-26 前后台软件结构

(2)中断型结构模式。如图 2-27 所示为中断型结构模式，在该结构中，除了初始化程序之外，整个系统软件的各个任务模块分别安排在不同级别的中断服务程序中，然后由中断管理系统(由硬件和软件组成)对各级中断服务程序实施调度管理。整个软件就是一个大的中断管理系统。

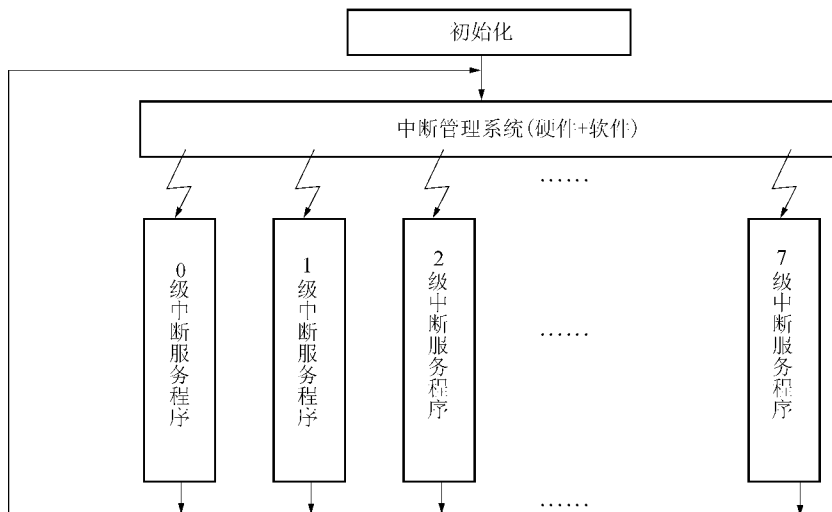


图 2-27 中断型结构模式

中断型结构模式的优点是实时性好,由于中断级别较多,强实时性任务可安排在优先级较高的中断服务程序中。其缺点是模块间的关系复杂,耦合度大,不利于对系统的维护和扩充。

2.3.2 CNC 系统软件的功能特点

控制软件是为完成特定 CNC 系统各项功能所编制的专用软件,又称为系统软件(或系统程序)。因此,CNC 系统的功能设置与控制方案各不相同,各种系统软件在结构和规模上差别很大。系统程序的设计与各项功能的实现及其将来的扩展有最直接的关系,是整个 CNC 系统研制工作中关键性和工作量最大的部分。

尽管不同的控制软件在结构和控制性能上大相径庭,但是系统软件一般由输入、译码、数据处理(预计算)、插补运算、速度控制、输出控制、管理程序及诊断程序等部分构成。下面分别加以介绍。

1. 输入

CNC 系统中一般通过纸带阅读机、磁带机、磁盘及键盘输入零件程序,且其输入大都采用中断方式。在系统程序中有相应的中断服务程序,如纸带阅读机中断服务程序及键盘中断服务程序等。当纸带阅读机读入一个字符至接口中时,就向主机发出中断,由中断服务程序将该字符送入内存。同样,每按一个键则表示向主机申请一次中断,调出一次键盘服务程序,对相应的键盘命令进行处理。

从阅读机及键盘输入的零件程序,一般是经过缓冲器以后,才进入零件程序存储器的。零件程序存储器的规模由系统设计员确定。一般有几 KB 字节,可以存放许多零件程序。如 7360 系统的零件程序存储器为 5 KB,可存放 20 多个零件程序。

键盘中断服务程序负责将键盘上打入的字符存入 MDI 缓冲器,按一下键就是向主机申请一次中断,如图 2-28 所示。

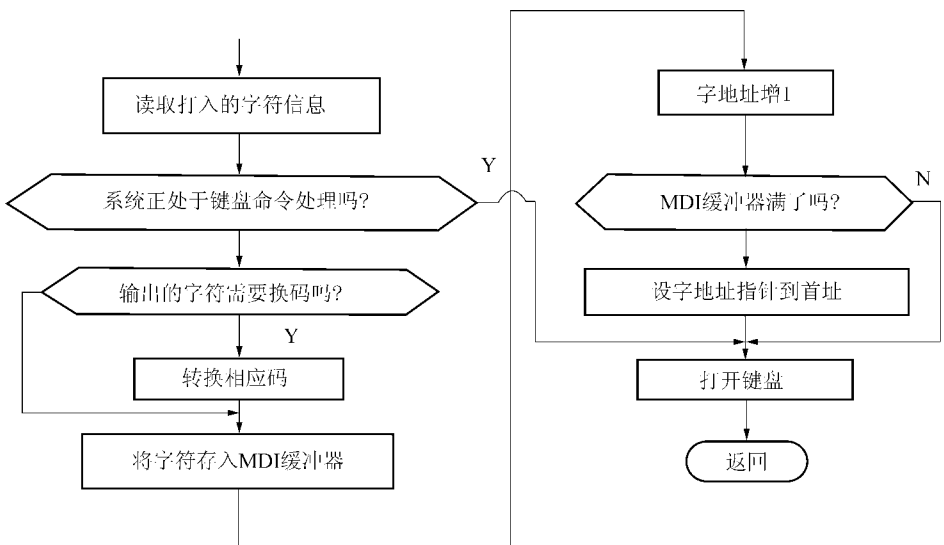


图 2-28 键盘中断服务程序

2. 译码

由前面的讨论可知,经过输入系统的工作,已将数据段送入零件程序存储器。下一步就是由译码程序将输入的零件程序数据段翻译成本系统能识别的语言。一个数据段从输入到传送至插补工作寄存器需经过如图 2-29 所示的几个环节。

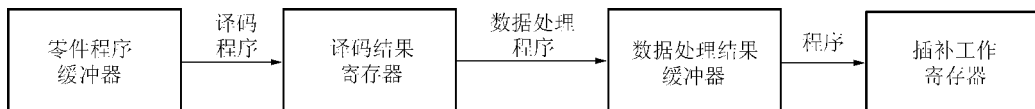


图 2-29 数据(程序)段传送过程

从原理和本质上说,软件译码与硬件译码相同。对于 8 单位的纸带程序,一个字符占 8 位。在 16 位字长的缓冲器中,一个字可存放两个字符。数据段长的则占的字数多。译码程序按次序将一个个字符和相应的数字进行比较,若相等了,则说明已输入了该字符。它就如同在硬件译码线路中一样,一个代码输入时只打开相应的某一个与门一样。所不同的是译码程序是串行工作的,即一个一个地比较,一直到相等时为止。而硬件译码线路则是并行工作的,因而速度较快。以 ISO 码为例, M 为 $(01,001,101)_2$,即 M 为八进制的 $(115)_8$, S 为 $(123)_8$, T 为 $(124)_8$, F 为 $(106)_8$ 等,因此,在判定数据段中是否已编入 M、S、T 或 F 字时,就可以将输入的字符和这些八进制数相比较,若相等了,则说明相应的字符已输入,立即设立相应的标志。某一个字符输入以后的处理过程包括:

(1)建立格式标志。如果是位格式,则每个字符所占的格式字的位数不同。

(2)根据输入字符的不同,确定相应的存放数值的地址。如 M 码的值存放在 1000H, S 码的值存放在 1002H。有的系统则对于各专用地址码(如 N、X、Y、G、M、F 等)在存放区域中都有一个位移量,该区域的首址加上地址码所对应的位移量,就可得到该地址码所存放的区域。

(3)确定调用“数码转换程序”的次数。一个代码后总有数字相接,如 M02、S11、X1000000。M 码的值最多为 2 位, S 码最多为 3 位(或 2 位), X 码的值最多为 7 位等。各个系统不尽相同,但对某一个具体系统而言,有一个规定值。如果某一个代码,它的值最多为 2 位,那么只需调用数码转换程序两次。所谓数码转换,即把输入的字符(如 ASCII 码)转换成二进制码在内存中存放。

将不同字符的处理器程序合并起来需要一张信息表。该表中每一个字符均有相应的一栏。栏中内容包括地址偏移量、在格式标志字中的位数及调用数码转换程序的次数,经过一次算术和逻辑运算即可以完成译码工作。在进行译码的同时,系统要对零件程序作语法检查,如输入的数字个数是否大于允许值,不允许带负号的地址码是否带了负号等。

译码的结果存放在规定的存储区内,存放译码结果的地方叫做译码结果存储器。译码结果存储器以规定的次序存放各代码的值(二进制),且包括一个程序格式标志单元,在该格式标志单元中某一位为 1,即表示指定的代码(如 F、S、M 等)已经被编入。为了使用方便,有时对 G 码、M 码的每一个值或几个值单独建立标志字。例如,对关于插补方式的 G00、G01、G02、G03 建立一个标志字,该标志字为 0 时代表已编入了 G00,为 1 时代表编入了 G01…。译码程序流程如图 2-30 所示。

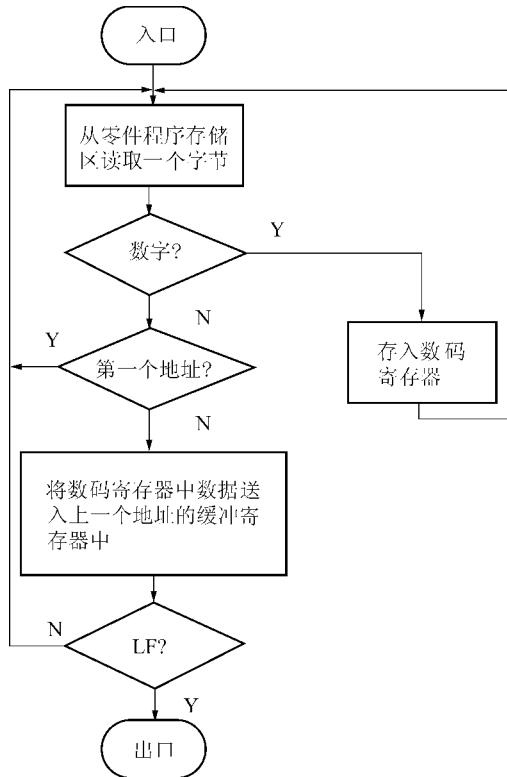


图 2-30 译码程序流程图

3. 预计算

为了减轻插补工作的负担,提高系统的实时处理能力,常常在插补运算前先进行数据的预处理,如确定圆弧平面、刀具半径补偿的计算等。当采用数字积分法时,可预先进行左移规格化的处理和积分次数的计算等,这样,可把最直接、最方便的数据形式提供给插补运算。

数据预处理即预计算,通常包括刀具长度补偿计算、刀具半径补偿计算、象限及进给方向判断、进给速度换算和机床辅助功能判断等。

在轮廓加工过程中,由于刀具总有一定的半径(如铣刀半径或线切割机的钼丝半径),刀具中心的运动轨迹并不等于所需加工零件的实际轨迹,也就是说,数控机床进行轮廓加工时必须考虑刀具半径。现以铣床为例,如图 2-31 所示,若要用半径为 r 的刀具加工外形轮廓为 A 的工件,那么刀具中心必须沿着与轮廓 A 偏离 r 距离的轨迹 B 移动,即铣削时刀具中心轨迹和工件的廓形是不一致的。我们可以根据轮廓 A 的坐标参数和刀具半径 r 的值计算出轨迹 B 的坐标参数,再编制成程序进行加工,但这很不方便。因为当材料、工艺变化或刀具磨损需要更换刀具时,需要重新制作程序。如果不考虑刀具半径,直接按照工件的廓形编程是比较方便的,但是这时刀具中心是按工件廓形运动,加工出来的零件比图纸要求缩小了。为了既能使编程方便,又能使刀具中心沿轨迹运动,加工出合格的零件来,就需要有刀具半径补偿(或称刀具偏移计算,简称刀偏)功能。

需要指出的是,刀具半径补偿通常不是由程序编制人员完成的,程序编制人员只是按零件的加工轮廓编制程序,同时用指令 G41、G42、G40 控制 CNC 系统,使刀具沿零件内轮廓运动或沿外轮廓运动。实际的刀具半径补偿是在 CNC 系统内部由计算机自动完成的。CNC

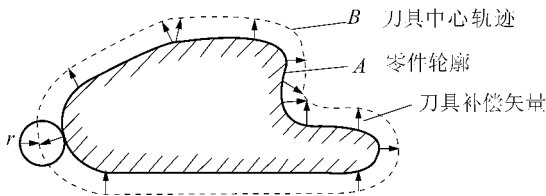


图 2-31 立铣刀半径补偿

系统根据零件轮廓尺寸(直线或圆弧以及其起点和终点)和刀具运动的方向指令(G41、G42、G40),以及实际加工中所用的刀具半径值自动地完成刀具半径补偿计算。

根据 ISO 标准,当刀具中心轨迹在编程轨迹(零件轮廓)前进方向右边时称为右刀具补偿,简称右刀补,用 G42 表示;反之,则称为左刀补,用 G41 表示;当不须要进行刀具半径补偿时用 G40 表示。

在实际轮廓加工过程中,刀具半径补偿执行过程一般可分为 3 步:

- (1)刀具补偿建立。
- (2)刀具补偿进行。
- (3)刀具补偿撤销。

进给速度的控制方法与系统采用的插补算法有关,也与伺服系统有关。在开环系统中,常常采用基准脉冲插补法,其坐标轴的运动速度控制是通过控制插补运算的频率,进而控制向步进电机输出脉冲的频率来实现的,速度计算的方法是根据编程 F 值来确定这个频率值。通常有程序延时法和中断法两种。

程序延时法又称为程序计时法。这种方法先根据系统要求的进给频率,计算出两次插补运算之间的时间间隔,用 CPU 执行延时子程序的方法控制两次插补之间的时间。改变延时子程序的循环次数,即可改变进给速度。

中断法又称为时钟中断法,是指每隔规定的时间向 CPU 发中断请求,在中断服务程序中进行一次插补运算并发出一个进给脉冲。因此,改变中断请求信号的频率,就等于改变了进给速度。中断请求信号可通过 F 指令设定的脉冲信号产生,也可通过可编程计数器/定时器产生。如采用 Z80CTC 作定时器,由程序设置时间常数,每到规定时间,就向 CPU 发中断请求信号,改变时间常数 T 。就可以改变中断请求脉冲信号的频率。因此,进给速度计算与控制的关键就是如何给定 CTC 的时间常数 T_c 。

在半闭环和闭环系统中,则是采用时间分割的思想,根据编程的进给速度 F 值将轮廓曲线分割为采样周期,即迭代周期的进给量——轮廓步长的方法。速度计算的任务是:当直线插补时,计算出各坐标轴的采样周期的步长;当插补圆弧时,为插补程序计算好步长分配系数(有时也称之为角步距)。另外,在进给速度控制中,一般也都有一个升速、恒速(匀速)和降速的过程,以适应伺服系统的工作状态,保证工作的稳定性。

4. 插补计算

插补计算是 CNC 系统中最重要计算工作之一。在传统的 NC 装置中,采用硬件电路(插补器)来实现各种轨迹的插补。为了在软件系统中计算所需的插补轨迹,这些数字电路必须由计算机程序来模拟。利用软件来模拟硬件电路的问题在于:三轴或三轴以上联动的系统具有三个或三个以上的硬件电路(如每轴一个数字积分器),计算机是用若干条指令来实现插补工作的,但是计算机执行每条指令都需要花费一定的时间,而当前有的小型或微型

计算机的计算速度难以满足 NC 机床对进给速度和分频率的要求。因此,在实际的 CNC 系统中,常常采用粗、精插补相结合的方法,即把插补功能分为软件插补和硬件插补两部分,计算机控制软件把刀具轨迹分为若干段,而硬件电路在段的起点和终点之间进行数据的“密化”,使刀具轨迹在允许的误差之内,即软件实现粗插补,硬件实现精插补。有关插补的详细计算原理将在 2.4 节详细介绍。

5. 输出

输出程序有以下几种功能:

(1)进行伺服控制。

(2)当进给脉冲改变方向时,要进行反向间隙补偿处理。若某一轴由正向变成负向运动,则在反向前输出 Q 个正向脉冲;反之,若由负向变成正向运动,则在反向前输出 Q 个负向脉冲(Q 为反向间隙值,可由程序预置)。

(3)进行丝杠螺距补偿。当系统具有绝对零点时,软件可显示刀具在任意位置上的绝对坐标值。若预先对机床各点精度进行测量,作出其误差曲线,随后将各点修正量制成表格存入数控系统的存储器中,这样,数控系统在运行过程中就可对各点坐标位置自动进行补偿,从而提高机床的精度。

(4)M、S、T 等辅助功能的输出。在某些程序段中需要启动机床主轴、改变主轴速度、换刀等,因此要输出 M、S、T 代码,这些代码大多数是开、关控制,由机床强电执行。但哪些辅助功能是在插补输出之后才执行,哪些辅助功能是在插补输出前执行,需要在软件设计前预先确认。

6. 管理与诊断软件

一般 CNC(MNC)系统中的管理软件只涉及两项,即 CPU 管理和外部设备管理。由于数控机床的加工是以单个零件为对象的,一个零件程序可以分成若干程序段。每个程序段的执行又分成数据分析、运算、走刀控制、其他动作的控制等步骤。通常情况下,这些加工步骤之间多是顺序关系,因此实际的过程就是这些预定步骤的反复执行。在实际系统中,通常多是采用一个主程序将整个加工过程串起来,主控程序对输入的数据分析判断后,转入相应的子程序处理,处理完毕后再返回对数据的分析、判断、运算……。在主控程序空闲时(如延时),可以安排 CPU 执行预防性诊断程序,或对尚未执行程序段的输入数据进行预处理等。

在 CNC 系统中,中断处理部分是重点,工作量也比较大。因为大部分实时性较强的控制步骤如插补运算、速度控制、故障处理等都要由中断处理来完成。有的机床将行程超程和报警、阅读机请求、插补等分为多级中断,根据其优先级决定响应的次序。有的机床则只设一级中断,只是在中断请求同时存在时,才用硬件排队或软件询问的方法来定一个顺序。

前台程序为实时中断程序,承担了几乎全部的实时功能,这些功能都与机床动作直接相关,如位置控制、插补、辅助功能处理、监控等。后台程序主要用来完成准备工作和管理工作,包括输入、译码、插补准备及管理,通常称为背景程序。背景程序是一个循环运行程序,在其运行过程中实时中断程序不断插入,前后台程序相互配合完成加工任务。如图 2-32 所示为前后台式软件结构示意图。

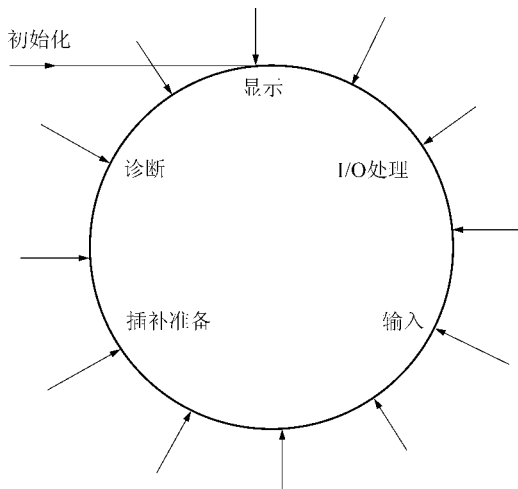


图 2-32 前后台结构

能够方便地设置各种诊断程序也是 CNC 和 MNC 系统的特点之一。有了较完善的诊断程序可以防止故障的发生或扩大,在故障出现后可以迅速查明故障的类型和部位,减少故障停机时间。各种 CNC(MNC)系统设置诊断程序的情况差别也很大。诊断程序可以包括在系统运行过程中进行检查和诊断;也可以作为服务性程序,在系统运行前或故障停机后进行诊断,查找故障的部位。国外一些公司的 CNC 系统还可以进行通信(海外)诊断,由通信诊断中心指示系统或操作者进行某些试运行,以查找故障隐患或故障部位。

1) 运行中的诊断

在普通 NC 系统中已包含有在运行中进行诊断的趋向,如纸带输入时的横向与纵向(水平与垂直)奇偶校验、同步孔丢失校验、非法指令码检查等。此外,还有超程报警等措施。在 CNC 和 MNC 系统中做这些工作更加方便。而且还可以用打字机指示各种项目诊断的结果和用统一编号表示的故障部位。一般来说,运行中进行诊断的程序比较零散,常包含在主控程序及中断处理程序的分支中,常见的方法有:

(1)用代码和检查内存。此法只能对程序中那些不变区域进行检查,而且必须是在恢复系统程序的初态后进行的。

(2)格式检查。此法一般用在纸带输入时,对零件加工源程序进行检查,包括奇偶校验、非法指令代码(本系统中不使用的指令代码)、数据超限等。

(3)双向传送数据校验。此法常用在间接型 CNC 系统或群控系统中。手动数据输入也可用此法校验。

(4)清单校验。即利用所配备的打印设备打印程序清单及某些中间数据,综合性地诊断主机、接口及软件的故障。

2) 停机诊断

停机诊断是指在系统开始运行前,或发生故障(包括故障先兆)系统停止运行后,利用计算机进行诊断。它一般是用软件控制进行阶段性的运行,如传送数据或模拟进行单项目的控制动作,逐项检查硬件线路的功能,有选择地查找故障部位。这种诊断程序可以与运行用的系统程序分开,在需要进行诊断时再输入计算机。必要时,还可冲掉部分系统程序而装入内存。

3) 通信诊断

通信诊断是由用户经电话线路与通信诊断中心联系,由该中心的计算机给用户的计算机发送诊断程序,程序指示 CNC 系统进行某种运行,同时收集数据,分析系统的状态。将系统状态与存储的应有工作状态以及某些极限参数作比较,以确定系统的工作状态是否正常,故障的部位及故障的趋势。可见,通信诊断既可用作诊断异常状态的工具,又可用于预防性检修的手段。因为用户与通信诊断中心之间可使用越洋电话系统,故又被称为海外诊断。

2.4 数控插补原理

2.4.1 插补的概念和分类

在数控加工中,若已知运动轨迹的起点坐标、终点坐标和曲线方程,则数控系统会根据这些信息实时地计算出各个中间点的坐标,使切削加工运动沿着预定轨迹移动,通常把这个过程称为“插补”。

所谓插补亦可以看做是数据密化的过程。在对数控系统输入有限坐标点(如起点、终点)的情况下,计算机根据线段的特征(直线、圆弧、椭圆等),运用一定的算法,自动地在有限坐标点之间生成一系列的坐标数据,即数据密化,从而自动地对各坐标轴进行脉冲分配,完成整个线段的轨迹运行,以满足加工精度的要求。在机床的实际加工中,被加工工件的轮廓形状千差万别,严格说来,为了满足几何尺寸精度的要求,刀具中心轨迹应该准确地依照工件的轮廓形状来生成。然而,对于简单的曲线,如直线和圆弧,数控装置易于实现,但对于较复杂的形状,若直接生成,势必会使算法变得很复杂,计算机的工作量也相应地大大增加。因此,在实际应用中,常常采用一小段直线或圆弧去进行逼近,有些场合也可以用抛物线、椭圆、双曲线和其他高次曲线去逼近(或称为拟合)。因此,数控机床在加工时,刀具的运动轨迹不是严格的直线或圆弧曲线,而是以折线轨迹逼近所要加工的曲线运动。

机床数控系统轮廓控制的主要问题就是怎样控制刀具或工件的运动轨迹。无论是硬件数控(NC)系统,还是 CNC 系统,都必须有完成插补功能的部分,只是采取的方式不同而已。在 CNC 中以软件(程序)或软、硬件结合实现插补,而在 NC 中有一个专门完成脉冲分配计算(即插补计算)的计算装置——插补器。无论是软件数控还是硬件数控,其插补的运算原理基本相同,其作用都是根据给定的信息进行数字计算,在计算过程中不断向各个坐标发出相互协调的进给脉冲,使被控机械部件按指定的路线移动。

关于插补算法问题,除了要保证插补计算的精度之外,还要求算法简单。这对于硬件数控来说,可以简化控制电路,采用较简单的运算器。而对于计算机数控系统来说,则能提高运算速度,使控制系统较快且均匀地输出进给脉冲。

插补工作可由硬件逻辑电路或执行软件程序来完成,在 CNC 系统中,插补工作一般由软件完成,软件插补结构简单、灵活易变、可靠性好。

目前普遍应用的两类插补方法为基准脉冲插补和数据采样插补。

1) 基准脉冲插补

基准脉冲插补又称为行程标量插补或脉冲增量插补,这类插补算法是以脉冲形式输出,

每进行一次插补运算,最多给每一轴一个进给脉冲,再把每次插补运算产生的指令脉冲输出到伺服系统,以驱动工作台运动。每发出一个脉冲,工作台移动一个基本长度单位,也叫脉冲当量,脉冲当量是脉冲分配的基本单位。

这种插补算法的特点是每次插补结束,数控装置向每个运动坐标输出基准脉冲序列,每个脉冲插补的实现方法比较简单(只有加法和移位),可以用硬件实现。目前,随着计算机技术的迅猛发展,多采用软件完成这类算法。脉冲的累积值代表运动轴的位置,脉冲产生的速度与运动轴的速度成比例。由于脉冲增量插补的转轴最大速度受插补算法执行时间限制,所以它仅适用于一些中等精度和中等速度要求的经济型计算机数控系统。

基准脉冲插补方法有以下几种:数字脉冲乘法器插补法、逐点比较法、数字积分法、矢量判别法、比较积分法、最小偏差法、目标点跟踪法、直接函数法、单步跟踪法、加密判别和双判别插补法、Bresenham 算法。

早期常用的脉冲增量式插补算法有逐点比较法、单步跟踪法、DDA 法等。插补精度常为一个脉冲当量,DDA 法还伴有运算误差。20 世纪 80 年代后期,插补算法有改进逐点比较法、直接函数法、最小偏差法等,使插补精度提高到半个脉冲当量,但执行速度不很理想,在插补精度和运动速度均较高的 CNC 系统中应用不广泛。近年来的插补算法有改进的最小偏差法、映射法,都有插补精度高和插补速度快的特点。

2) 数据采样插补

数据采样插补又称为时间标量插补或数字增量插补,这类算法插补结果输出的不是脉冲,而是标准二进制数。根据编程中的进给速度,把轮廓曲线按插补周期将其分割为一系列微小直线段,然后将这些微小直线段对应的位置增量数据进行输出,以控制伺服系统实现坐标轴的进给。

插补计算是计算机数控系统中实时性很强的一项工作,为了提高计算速度,缩短计算时间,按以下 3 种结构方式进行改进:

- (1) 采用软/硬件结合的两级插补方案。
- (2) 采用多 CPU 的分布式处理方案。
- (3) 采用单台高性能微型计算机方案。

数控插补的常用方法有逐点比较插补法、数值积分法和数据采样插补法。

2.4.2 逐点比较插补法

所谓逐点比较插补法,就是机床每走到一个坐标位置,都要和给定的轨迹上的坐标值比较一次,看实际加工点在给定轨迹的什么位置,判断其偏差,然后决定下一步的走向,如果加工点走到图形外面去了,那么下一步就要向图形里面走;如果加工点在图形里面,那么下一步就要向图形外面走,以缩小偏差。逐点比较法是以阶梯折线来逼近直线和圆弧的,最大偏差不超过一个脉冲当量,因此,只要把脉冲当量控制得足够小,就能达到加工精度的要求。

1. 逐点比较法直线插补

1) 直线插补偏差计算公式

偏差计算是逐点比较法关键的一步。下面以第 I 象限直线为例导出其偏差计算公式。

如图 2-33 所示为动点与直线的位置关系。第 I 象限直线 OE , 起点 O 为坐标原点, 用户编程时, 给出直线的终点坐标为 $E(X_e, Y_e)$, 直线方程为

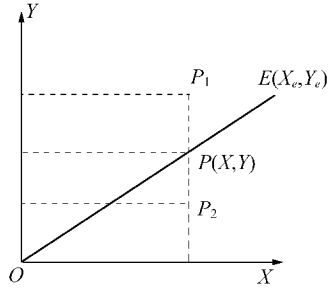


图 2-33 直线插补过程

$$X_e Y - Y_e X = 0$$

直线 OE 为给定轨迹, $P(X, Y)$ 为动点坐标, 动点与直线的位置关系有 3 种情况:

- (1) 若 P_1 点在直线上方, 则有 $X_e Y - Y_e X > 0$;
- (2) 若 P 点在直线上, 则有 $X_e Y - Y_e X = 0$;
- (3) 若 P_2 点在直线下方, 则有 $X_e Y - Y_e X < 0$ 。

因此, 可以构造偏差函数为

$$F = X_e Y - Y_e X$$

对于第 I 象限直线, 其偏差符号与进给方向的关系为:

- (1) 当 $F=0$ 时, 表示动点在 OE 上, 如点 P , 可向 $+X$ 向进给, 也可向 $+Y$ 向进给。
- (2) 当 $F>0$ 时, 表示动点在 OE 上方, 如点 P_1 , 应向 $+X$ 向进给。
- (3) 当 $F<0$ 时, 表示动点在 OE 下方, 如点 P_2 , 应向 $+Y$ 向进给。

这里规定动点在直线上时, 可归入 $F>0$ 的情况一同考虑。插补工作从起点开始, 走一步, 计算一步, 判别一次, 再走一步, 当沿两个坐标方向走的步数分别等于 X_e 和 Y_e 时, 停止插补。下面将 F 的运算采用递推算法予以简化, 动点 $P_i(X_i, Y_i)$ 的 F_i 值为

$$F_i = X_e Y_i - Y_e X_i$$

若 $F_i \geq 0$, 表明 $P_i(X_i, Y_i)$ 点在 OE 直线上方或在直线上, 应沿 $+X$ 向走一步, 假设坐标值的单位为脉冲当量, 走完一步后新的坐标值为 (X_{i+1}, Y_{i+1}) , 且 $X_{i+1} = X_i + 1, Y_{i+1} = Y_i$, 新点偏差为

$$\begin{aligned} F_{i+1} &= X_e Y_{i+1} - Y_e X_{i+1} \\ &= X_e Y_i - Y_e (X_i + 1) \\ &= X_e Y_i - Y_e X_i - Y_e \\ &= F_i - Y_e \end{aligned}$$

即
$$F_{i+1} = F_i - Y_e$$

若 $F_i < 0$, 则表明 $P_i(X_i, Y_i)$ 点在 OE 的下方, 应向 $+Y$ 方向进给一步, 新点坐标值为 (X_{i+1}, Y_{i+1}) , 且 $X_{i+1} = X_i, Y_{i+1} = Y_i + 1$, 新点的偏差为

$$\begin{aligned} F_{i+1} &= X_e Y_{i+1} - Y_e X_{i+1} \\ &= X_e (Y_i + 1) - Y_e X_i \\ &= X_e Y_i - Y_e X_i + X_e \\ &= F_i + X_e \end{aligned}$$

即
$$F_{i+1} = F_i + X_e$$

偏差计算通常采用的是迭代法,或称递推法,即每走一步后新加工点的加工偏差值用前一点的加工偏差递推出来。

机床开始加工工件时,首先将刀具移到起点,刀具正好处于直线上,偏差为零,即 $F=0$,根据这一点偏差可求出新一点偏差,随着加工的进行,依据递推公式每走一步后新加工点的加工偏差值都可以用前一点的加工偏差递推出来。

2) 终点判断

在插补计算、进给的同时还要进行终点判别。常用终点判别方法有两种,一种是设置一个长度计数器,从直线的起点走到终点,刀具沿 X 轴应走的步数为 X_e ,沿 Y 轴走的步数为 Y_e ,计数器中存入 X 和 Y 两坐标进给步数总和 $\Sigma = |X_e| + |Y_e|$,当 X 或 Y 坐标每进给一步,计数长度减一,当计数长度减到零时,即当 $\Sigma = 0$ 时,停止插补,到达终点;另一种是如果在两个轴上的插补数不一样多,则将插补步数较大的轴设为计数轴,步数值设为计数长度,当在计数轴上每进给一步计数长度减一,当 $\Sigma = 0$ 时,停止插补,到达终点。

3) 插补计算过程

如图 2-34 所示为逐点比较法直线插补流程。逐点比较法的直线插补过程为每走一步要进行以下 4 个节拍(步骤),即偏差判别、坐标进给、偏差计算、终点判别。

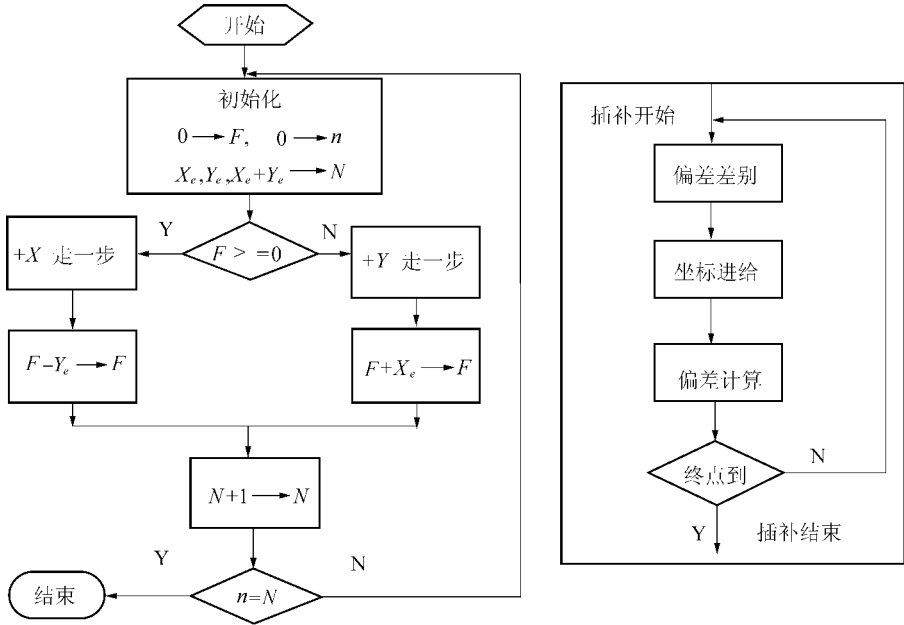


图 2-34 逐点比较法直线插补流程

偏差判别是根据偏差值确定刀具位置是在直线的上方(或线上),还是在直线的下方。坐标进给是根据判别的结果,决定控制哪个坐标(X 或 Y)移动一步。偏差计算是计算出刀具移动后的新偏差,提供给下一步作判别依据。根据公式来计算新加工点的偏差,使运算大大简化。但是,每一新加工点的偏差都是由前一点偏差推算出来的,并且一直递推下去,这样就要知道开始加工时那一点的偏差是多少。当开始加工时,以人工方式将刀具移到加工起点,这一点当然没有偏差,所以开始加工点的 F 值为零。

终点判别是在计算偏差的同时,还要进行一次终点比较,以确定是否到达了终点。若已

经到达,就不再进行运算,并发出停机或转换新程序段的信号。

例 2-1 加工第 I 象限直线 OE ,如图 2-35 所示,起点为坐标原点,终点坐标为 $E(5,3)$ 。试用逐点比较法对该段直线进行插补,并画出插补轨迹。

初始点 $O(0,0)$,终点 $(5,3)$,应用递推公式进行偏差计算。终点判断 $\Sigma = |X_e| + |Y_e| = 5+3=8$,插补需要七个循环,如表 2-1 所示。

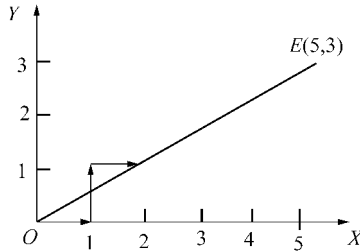


图 2-35 直线插补轨迹过程实例

表 2-1 直线插补运算过程

插补循环	偏差判别	进给方向	偏差计算	终点判别
0			$F_0=0, X_e=5, Y_e=3$	$\Sigma=0, N=8$
1	$F_0=0$	+X	$F_1=F_0-Y_e=0-3=-3$	$\Sigma=0+1=1<N$
2	$F_1=-3<0$	+Y	$F_2=F_1+X_e=-3+5=2$	$\Sigma=1+1=2<N$
3	$F_2=2>0$	+X	$F_3=F_2-Y_e=2-3=-1$	$\Sigma=2+1=3<N$
4	$F_3=-1<0$	+Y	$F_4=F_3+X_e=-1+5=4$	$\Sigma=3+1=4<N$
5	$F_4=4>0$	+X	$F_5=F_4-Y_e=4-3=1$	$\Sigma=4+1=5<N$
6	$F_5=1>0$	+X	$F_6=F_5-Y_e=1-3=-2$	$\Sigma=5+1=6<N$
7	$F_6=-2<0$	+Y	$F_7=F_6+X_e=-2+5=3$	$\Sigma=6+1=7<N$
8	$F_7=3>0$	+X	$F_8=F_7-Y_e=3-3=0$	$\Sigma=7+1=8=N$

4) 四象限插补

假设有第 III 象限直线 OE' ,如图 2-36 所示,起点坐标在原点 O ,终点坐标为 $E'(-X_e, -Y_e)$,在第 I 象限有一条和它对称于原点的直线,其终点坐标为 $E(X_e, Y_e)$,按第 I 象限直线进行插补时,从 O 点开始把沿 X 轴正向进给改为 X 轴负向进给,沿 Y 轴正向改为 Y 轴负向进给,这时实际插补出的就是第 III 象限直线,其偏差计算公式与第 I 象限直线的偏差计算公式相同,仅仅是进给方向不同,输出驱动时,应使 X 和 Y 轴电机反向旋转。

四个象限直线的偏差符号和插补进给方向如图 2-37 所示,用 L_1, L_2, L_3, L_4 分别表示第 I、II、III、IV 象限的直线。为适用于四个象限直线插补,插补运算时用 $|X|$ 和 $|Y|$ 代替 X 和 Y ,偏差符号确定可将其转化到第 I 象限,动点与直线的位置关系按第 I 象限判别方式进行判别。

在图 2-37 中,靠近 Y 轴区域偏差大于零,靠近 X 轴区域偏差小于零。当 $F \geq 0$ 时,进给都是沿 X 轴,不管是 $+X$ 向还是 $-X$ 向, X 的绝对值增大;当 $F < 0$ 时,进给都是沿 Y 轴,不论 $+Y$ 向还是 $-Y$ 向, Y 的绝对值增大。如图 2-38 所示为四象限直线插补流程图,如表 2-2 所示为直线插补公式及进给方向。

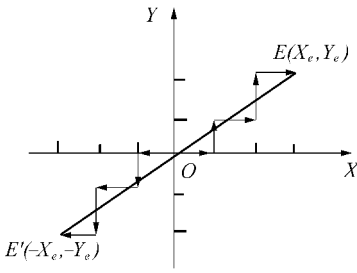


图 2-36 第三象限直线插补

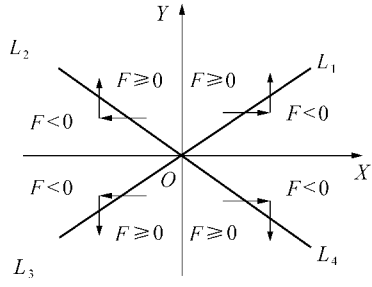


图 2-37 四象限直线偏差符号和进给方向

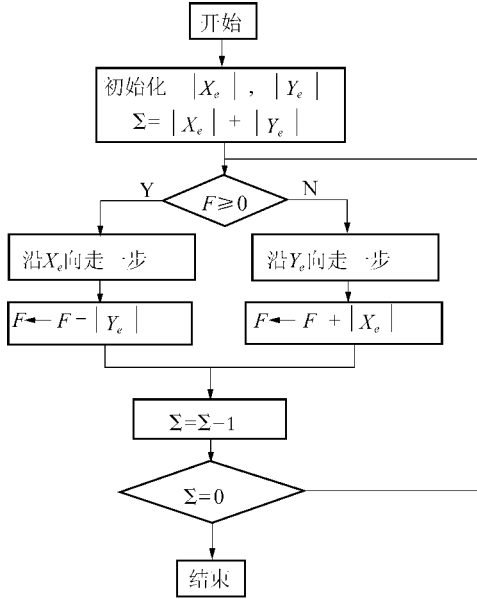


图 2-38 四象限直线插补流程图

表 2-2 直线插补公式及进给方向

$F_m \geq 0$			$F_m \leq 0$		
直线线型	进给方向	偏差计算	直线线型	进给方向	偏差计算
L_1, L_4	+X	$F_{m+1} = F_m - Y_e$	L_1, L_2	+Y	$F_{m+1} = F_m + X_e$
L_2, L_3	-X		L_3, L_4	-Y	

2. 圆弧插补原理

1) 圆弧插补偏差计算公式

在圆弧加工过程中,可用动点到圆心的距离来描述刀具位置与被加工圆弧之间关系。如图 2-39 所示为圆弧插补过程,设圆弧圆心在坐标原点,已知圆弧起点坐标为 $A(X_a, Y_a)$, 终点坐标为 $B(X_b, Y_b)$,圆弧半径为 R 。

加工点可能在 3 种情况出现,即圆弧上、圆弧外和圆弧内。当动点 $P(X_i, Y_i)$ 位于圆弧上时有

$$X^2 + Y^2 - R^2 = 0$$

P 点在圆弧外侧时,则 OP 大于圆弧半径 R , 即

$$X^2 + Y^2 - R^2 > 0$$

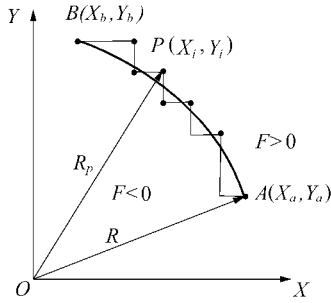


图 2-39 圆弧插补过程

P 点在圆弧内侧时, 则 OP 小于圆弧半径 R , 即

$$X^2 + Y^2 - R^2 < 0$$

用 F 表示 P 点的偏差值, 定义圆弧偏差函数判别式为

$$F = X^2 + Y^2 - R^2$$

当动点落在圆弧上时, 一般约定将其和 $F > 0$ 一并考虑。

如图 2-40(a) 所示 AB 为第 I 象限顺圆弧 SR_1 , 若 $F \geq 0$ 时, 动点在圆弧上或圆弧外, 向 $-Y$ 方向进给, 计算出新点的偏差; 若 $F < 0$, 则表明动点在圆内, 向 $+X$ 向进给, 计算出新一点的偏差, 如此走一步, 计算一步, 直至终点。

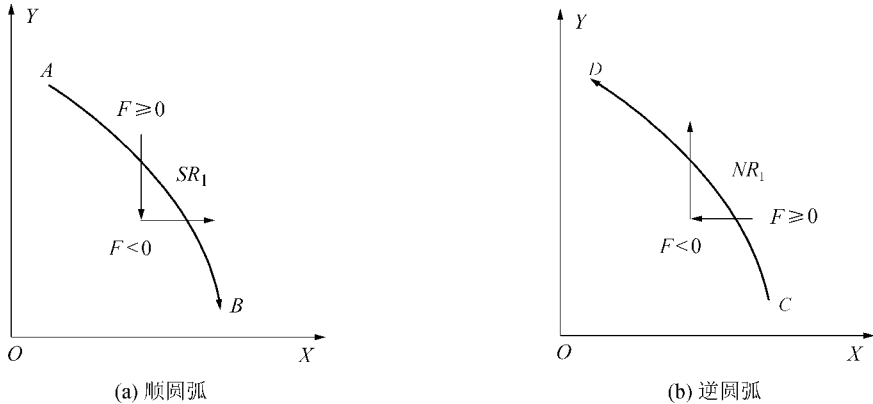


图 2-40 第 I 象限顺、逆圆弧

由于偏差计算公式中有平方值计算, 下面采用递推公式给予简化, 对第一象限顺圆, $F_i \geq 0$, 动点 $P_i(X_i, Y_i)$ 应向 $-Y$ 向进给, 新的动点坐标为 (X_{i+1}, Y_{i+1}) , 且 $X_{i+1} = X_i, Y_{i+1} = Y_i - 1$, 则新点的偏差值为

$$\begin{aligned} F_{i+1} &= X_{i+1}^2 + Y_{i+1}^2 - R^2 \\ &= X_i^2 + (Y_i - 1)^2 - R^2 \end{aligned}$$

即

$$F_{i+1} = F_i - 2Y_i + 1$$

若 $F_i < 0$ 时, 则沿 $+X$ 向前进一步, 到达 (X_{i+1}, Y_{i+1}) 点, 且 $X_{i+1} = X_i + 1, Y_{i+1} = Y_i$ 新点的偏差值为

$$\begin{aligned} F_{i+1} &= X_{i+1}^2 + Y_{i+1}^2 - R^2 \\ &= (X_i + 1)^2 + Y_i^2 - R^2 \end{aligned}$$

即

$$F_{i+1} = F_i + 2X_i + 1$$

进给后新点的偏差计算公式除与前一点偏差值有关外,还与动点坐标有关,动点坐标值随着插补的进行是变化的,所以在圆弧插补的同时,还必须修正新的动点坐标。

2) 终点判断

圆弧插补终点判别:将 X、Y 轴走的步数总和存入一个计数器, $\Sigma = |X_b - X_a| + |Y_b - Y_a|$, 每走一步 Σ 减一,直到 $\Sigma = 0$ 时发出停止信号。

3) 插补计算过程

圆弧插补过程与直线插补过程基本相同,但由于其偏差计算公式不仅与前一点偏差有关,且与前一点坐标有关,故在偏差计算的同时要进行坐标计算,以便为下一点的偏差计算做好准备。即圆弧插补过程为偏差判别、坐标进给、偏差计算、坐标计算及终点判别 5 个步骤。圆弧插补流程图如图 2-41 所示。

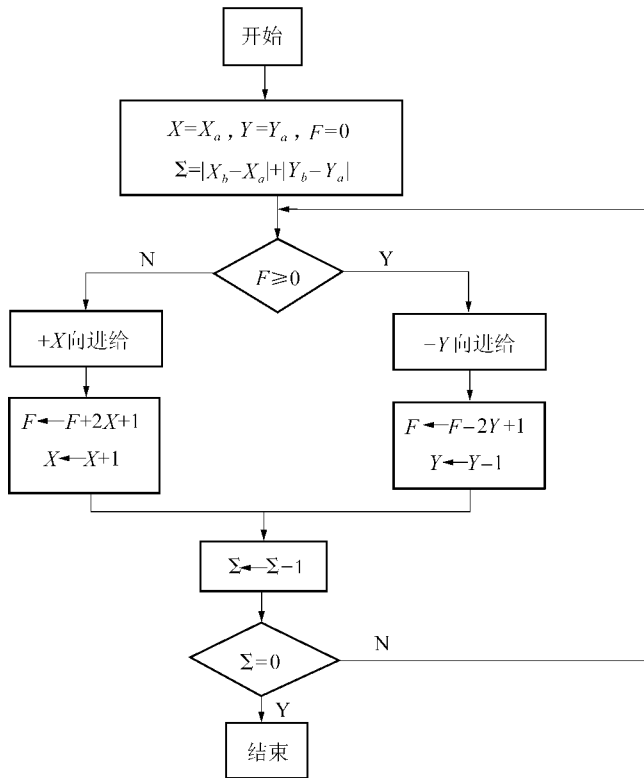


图 2-41 顺圆弧插补流程图

例 2-2 现欲加工第 I 象限顺圆弧 AB, 如图 2-42 所示, 起点坐标为 A(0,4), 终点坐标为 B(4,0), 试用逐点比较法进行插补。计算过程如表 2-3 所示。

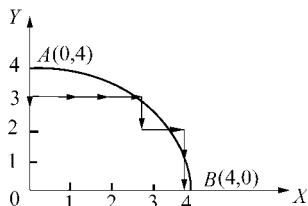


图 2-42 圆弧插补实例

表 2-3 圆弧插补运算过程

步数	偏差判别	坐标进给	偏差计算	坐标计算	终点判别
起点			$F_0=0$	$X_0=0, Y_0=4$	$\Sigma=8$
1	$F_0=0$	$-Y$	$F_1=F_0-2Y_0+1=-7$	$X_1=0, Y_1=3$	$\Sigma=7$
2	$F_1<0$	$+X$	$F_2=F_1+2X_1+1=-6$	$X_2=1, Y_2=3$	$\Sigma=6$
3	$F_2<0$	$+X$	$F_3=F_2+2X_2+1=-3$	$X_3=2, Y_3=3$	$\Sigma=5$
4	$F_3<0$	$+X$	$F_4=F_3+2X_3+1=2$	$X_4=3, Y_4=3$	$\Sigma=4$
5	$F_4>0$	$-Y$	$F_5=F_4-2Y_4+1=-3$	$X_5=3, Y_5=2$	$\Sigma=3$
6	$F_5<0$	$+X$	$F_6=F_5+2X_5+1=4$	$X_6=4, Y_6=2$	$\Sigma=2$
7	$F_6>0$	$-Y$	$F_7=F_6-2Y_6+1=1$	$X_7=4, Y_7=1$	$\Sigma=1$
8	$F_7>0$	$-Y$	$F_8=F_7-2Y_7+1=0$	$X_8=4, Y_8=0$	$\Sigma=0$

4) 四个象限圆弧插补

如图 2-40(b) 所示, 第 I 象限逆圆弧 CD 的运动趋势是沿 X 轴方向绝对值减少, 沿 Y 轴方向绝对值增大, 当动点在圆弧上或圆弧外, 即 $F_i \geq 0$ 时, X 轴沿负向进给, 新动点的偏差函数为

$$F_{i+1} = F_i - 2X_i + 1$$

当 $F_i < 0$ 时, Y 轴沿正向进给, 新动点的偏差函数为

$$F_{i+1} = F_i + 2Y_i + 1$$

如果插补计算都用坐标的绝对值, 将进给方向另做处理, 四个象限的插补公式可以统一起来, 当对第 I 象限顺圆插补时, 将 X 轴正向进给改为 X 轴负向进给, 则走出的是第 II 象限逆圆, 若将 X 轴沿负向、 Y 轴沿正向进给, 则走出的是第 III 象限顺圆。

用 SR_1 、 SR_2 、 SR_3 、 SR_4 分别表示第 I、II、III、IV 象限的顺时针圆弧, 用 NR_1 、 NR_2 、 NR_3 、 NR_4 分别表示第 I、II、III、IV 象限的逆时针圆弧, 四个象限圆弧的进给方向表示如图 2-43 所示。

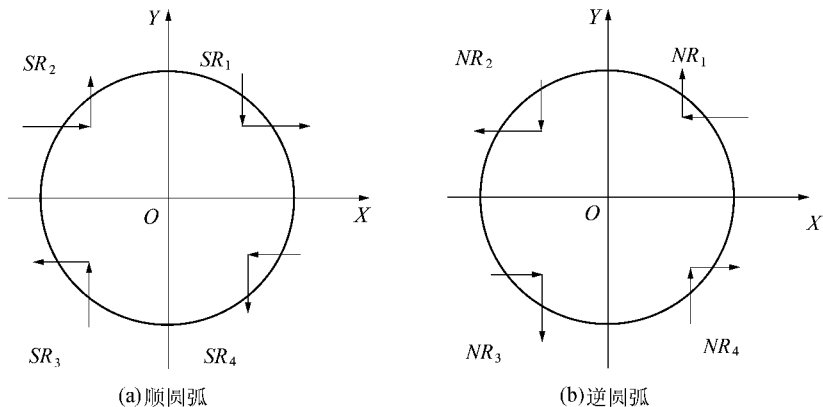


图 2-43 四个象限圆弧进给方向

圆弧过象限, 即圆弧的起点和终点不在同一象限内时, 若坐标采用绝对值进行插补运算, 应先进行过象限判断, 当 $X=0$ 或 $Y=0$ 时过象限。如图 2-44 所示, 需将圆弧 AC 分成两段圆弧 AB 和 BC , 到 $X=0$ 时进行处理, 对应调用顺圆 AB 和顺圆 BC 的插补程序。

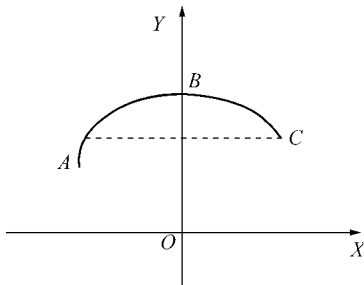


图 2-44 跨象限圆弧

若用带符号的坐标值进行插补计算,在插补的同时,比较动点坐标和终点坐标的代数
值,若两者相等,则插补结束,其计算过程如表 2-4 所示。

表 2-4 圆弧插补计算过程

进给	坐标计算	偏差计算	终点判别
+X	$X_{i+1}=X_i+1$	$F_{i+1}=F_i+2X_i+1$	$X_e-X_{i+1}=0$
-X	$X_{i+1}=X_i-1$	$F_{i+1}=F_i-2X_i+1$	$X_e-X_{i+1}=0$
+Y	$Y_{i+1}=Y_i+1$	$F_{i+1}=F_i+2Y_i+1$	$Y_e-Y_{i+1}=0$
-Y	$Y_{i+1}=Y_i-1$	$F_{i+1}=F_i-2Y_i+1$	$Y_e-Y_{i+1}=0$

3. 逐点比较法合成进给速度

从前面的讨论知道,插补器向各个坐标分配进给脉冲,这些脉冲造成坐标的移动。因此,对于某一坐标而言,进给脉冲的频率就决定了进给速度。以 X 坐标为例,设 f_x 为以“脉冲/秒”表示的脉冲频率, v_x 为以“mm/min”表示的进给速度,它们的比例关系为

$$v_x = 60\delta f_x$$

式中, δ 为脉冲当量,毫米/脉冲。

各个坐标进给速度的合成线速度称为合成进给速度或插补速度。对三坐标系统来说,合成进给速度 v 为

$$v = \sqrt{v_x^2 + v_y^2 + v_z^2}$$

式中, v_x, v_y, v_z 分别为 X、Y、Z 方向的进给速度。

合成进给速度直接决定了加工时的粗糙度和精度。在插补过程中,希望合成进给速度恒等于指令进给速度,或只在允许的范围内变化。但是实际上,合成进给速度 v 与插补计算方法、脉冲源频率及程序段的形式和尺寸都有关系。也就是说,不同的脉冲分配方式,指令进给速度 F 和合成进给速度 v 之间的换算关系各不相同。下面来计算逐点比较法的合成进给速度。

逐点比较法的特点是脉冲源每产生一个脉冲,不是发向 X 轴(ΔX),就是发向 Y 轴(ΔY)。令 f_g 为脉冲源频率,单位为“个脉冲/秒”,则有

$$f_g = f_x + f_y$$

从而 X 和 Y 方向的进给速度 v_x 和 v_y (mm/min) 分别为

$$v_x = 60\delta f_x, \quad v_y = 60\delta f_y$$

合成进给速度 v 为

$$v = \sqrt{v_x^2 + v_y^2} = 60\delta \sqrt{f_x^2 + f_y^2}$$

当 $f_x=0$ (或 $f_y=0$)时,也就是进给脉冲按平行于坐标轴的方向分配时有最大速度,这个速度由脉冲源频率决定,所以称其为脉冲源速度 v_g (实质是指循环节拍的频率,单位为 mm/min)。

$$v_g = 60\delta f_g$$

合成进给速度 v 与 v_g 之比为

$$\frac{v}{v_g} = \frac{\sqrt{f_x^2 + f_y^2}}{f_g} = \frac{\sqrt{X^2 + Y^2}}{X + Y}$$

其插补速度 v 的变化范围为 $v = (1 \sim 0.707)v_g$,最大速度与最小速度之比为

$$k_v = \frac{v_{\max}}{v_{\min}} = 1.414$$

这样的速度变化范围,对一般机床来说已可满足要求,所以逐点比较法的进给速度是较平稳的。

2.4.3 数值积分法

数字积分法又称为数字微分分析法(Digital Differential Analyzer,简称为 DDA),是在数字积分器的基础上建立起来的一种插补算法。数字积分法的优点是,易于实现多坐标联动,较容易地实现二次曲线、高次曲线的插补,并具有运算速度快,应用广泛等特点。

如图 2-45 所示,设有一函数 $Y=f(t)$,求此函数在 $t_0 \sim t_n$ 区间的积分,就是求出此函数曲线与横坐标 t 在区间 (t_0, t_n) 所围成的面积。如果将横坐标区间段划分为间隔为 t 的很多小区间,当 t 取足够小时,此面积可近似地视为曲线下许多小矩形面积之和。

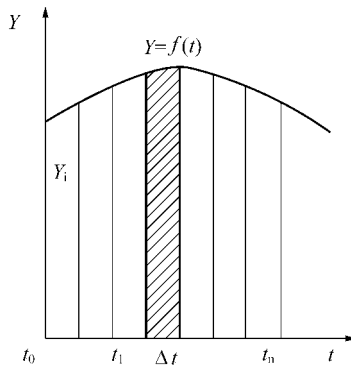


图 2-45 函数 $Y=f(t)$ 的积分

$$S = \int_{t_0}^{t_n} Y dt = \sum_{i=0}^{n-1} Y_i \Delta t$$

式中, Y_i 为 $t=t_i$ 时 $f(t)$ 的值。

这个公式说明,求积分的过程也可以用累加的方式来近似。在数学运算时,取 t 为基本单位“1”,则上式可简化为

$$S = \sum_{i=0}^{n-1} Y_i$$

数字积分器通常由函数寄存器、累加器和与门等组成。其工作过程为:每隔 t 时间发一个脉冲,与门打开一次,将函数寄存器中的函数值送累加器里累加一次,令累加器的容量为

一个单位面积,当累加和超过累加器的容量一个单位面积时,便发出溢出脉冲,这样累加过程中产生的溢出脉冲总数就等于所求的总面积,也就是所求积分值。数字积分器结构框图如图 2-46 所示。

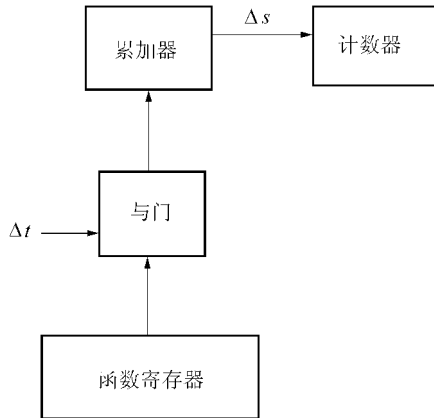


图 2-46 数字积分器结构框图

1) 数字积分法直线插补

如图 2-47 所示,直线 OE 起点为坐标原点 O ,终点坐标为 $E(7,4)$ 。设寄存器和累加器容量为 1,将 $X_e=7, Y_e=4$ 分别分成 8 段,每一段分别为 $7/8, 4/8$,将其存入 X 和 Y 函数寄存器中。

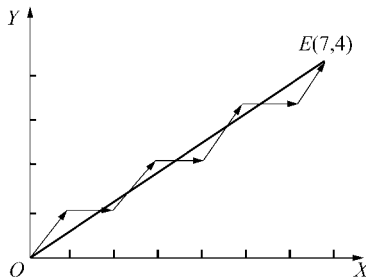


图 2-47 直线插补过程

第一个时钟脉冲来到时,累加器里的值分别为 $7/8$ 和 $4/8$,因不大于累加器容量,没有溢出脉冲。

第二个时钟脉冲来到时, X 累加器累加结果为 $7/8+7/8=1+6/8$,因累加器容量为 1,满 1 就溢出一个脉冲,则往 X 方向发出一进给脉冲,余下的 $6/8$ 仍寄存在累加器里,累加器又称余数寄存器。 Y 累加器中累加为 $4/8+4/8$,其结果等于 1, Y 方向也进给一步。

第三个脉冲到来时,仍继续累加, X 累积器为 $6/8+7/8$,大于 1, X 方向再走一步, Y 累加器中为 $0+4/8$,其结果小于 1,无溢出脉冲, Y 向不走步。

如此下去,直到输入第 8 个脉冲时,积分器工作一个周期, X 方向溢出脉冲总数为 $7/8 \times 8=7$, Y 方向溢出脉冲总数为 $4/8 \times 8=4$,到达终点 E 。

若要加工第 I 象限直线 OE ,如图 2-48 所示,起点为坐标原点 O ,终点坐标为 $E(X_e, Y_e)$,刀具以匀速 v 由起点移向终点,其 X, Y 坐标的速度分量为 v_x, v_y ,则有

$$\frac{v}{OE} = \frac{v_x}{X_e} = \frac{v_y}{Y_e} = k \quad (k \text{ 为常数})$$

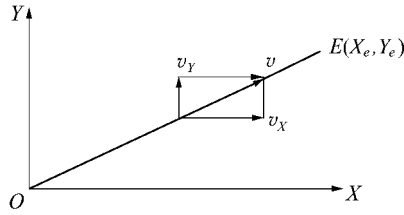


图 2-48 DDA 直线插补

各坐标轴的位移量为

$$X = \int v_x dt = \int kX_e dt$$

$$Y = \int v_y dt = \int kY_e dt$$

数字积分法是求从 O 到 E 区间的定积分。此积分值等于由 O 到 E 的坐标增量,因积分是从原点开始的,所以坐标增量即是终点坐标

$$\int_{t_0}^{t_n} kX_e dt = X_e - X_0$$

$$\int_{t_0}^{t_n} kY_e dt = Y_e - Y_0$$

t_0 对应直线起点的时间, t_n 对应终点时间。用累加来代替积分,刀具在 X 、 Y 方向移动的微小增量分别为

$$\Delta X = v_x \Delta t = kX_e \Delta t$$

$$\Delta Y = v_y \Delta t = kY_e \Delta t$$

动点从原点出发走向终点的过程,可以看作是各坐标轴每经过一个单位时间间隔 t ,分别以增量 kX_e 及 kY_e 同时累加的结果。即

$$X = \sum_{i=1}^m \Delta X_i = \sum_{i=1}^m kX_e \Delta t_i$$

$$Y = \sum_{i=1}^m \Delta Y_i = \sum_{i=1}^m kY_e \Delta t_i$$

取 $\Delta t_i = 1$ (一个单位时间间隔), 则

$$X = kX_e \sum_{i=1}^m \Delta t_i = kmX_e$$

$$Y = kY_e \sum_{i=1}^m \Delta t_i = kmY_e$$

若经过 m 次累加后, X 、 Y 都到达终点 $E(X_e, Y_e)$, 下式成立

$$X = kmX_e = X_e$$

$$Y = kmY_e = Y_e$$

可见累加次数与比例系数之间关系为

$$km = 1 \quad \text{或} \quad m = 1/k$$

两者互相制约,不能独立选择, m 是累加次数,取整数, k 取小数。即先将直线终点坐标 (X_e, Y_e) 缩小到 (kX_e, kY_e) , 然后再经 m 次累加到达终点。此外,还要保证沿坐标轴每次进给脉冲不超过一个,保证插补精度,则应使下式成立,即

$$\Delta X = kX_e < 1$$

$$\Delta Y = kY_e < 1$$

如果存放 X_e, Y_e 寄存器的位数是 n , 对应最大允许数字量为 $2^n - 1$ (各位均为 1), 所以 X_e, Y_e 最大寄存数值为 $2^n - 1$, 则

$$k(2^n - 1) < 1$$

$$k < \frac{1}{2^n - 1}$$

为使上式成立, 令 $k = \frac{1}{2^n}$, 代入得

$$\frac{2^n - 1}{2^n} < 1$$

累加次数为 $m = \frac{1}{k} = 2^n$

上式表明, 若寄存器位数是 n , 则整个直线插补过程要进行 2^n 次累加才能到达终点。

对于二进制数来说, 一个 n 位寄存器中存放 X_e 和存放 kX_e 的数字是一样的, 只是小数点的位置不同罢了, X_e 除以 2^n , 只需把小数点左移 n 位, 小数点出现在最高位数 n 的前面。采用 kX_e 进行累加, 累加结果大于 1, 就有溢出; 若采用 X_e 进行累加, 超出寄存器容量 2^n 有溢出。将溢出脉冲用来控制机床进给, 其效果是一样的。在被积函数寄存器里可只存 X_e , 而省略 k 。

如 $X_e = 100101$ 在一个 6 位寄存器中存放, 若 $k = 1/2^6$, $kX_e = 0.100101$ 也存放在 6 位寄存器中, 数字是一样的; 若进行一次累加, 都有溢出, 余数数字也相同, 只是小数点位置不同而已, 因此可用 X_e 替代 kX_e 。

如图 2-49 所示为平面直线的插补框图, 它由两个数字积分器组成, 每个坐标轴的积分器由累加器和被积函数寄存器组成, 被积函数寄存器存放终点坐标值, 每经过一个时间间隔 t , 将被积函数值向各自的累加器中累加, 当累加结果超出寄存器容量时, 就溢出一个脉冲, 若寄存器位数为 n , 经过 2^n 次累加后, 每个坐标轴的溢出脉冲总数就等于该坐标的被积函数值, 从而控制刀具到达终点。

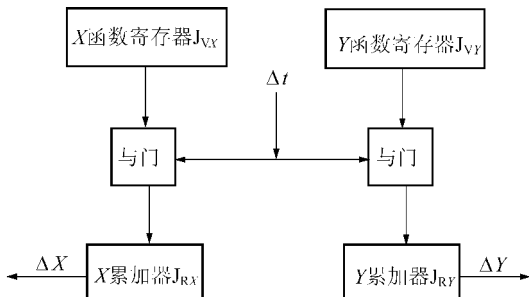


图 2-49 平面直线的插补框图

例 2-3 设有一直线 OE , 如图 2-50 所示。其起点坐标为 $O(0, 0)$, 终点坐标为 $E(4, 3)$, 累加器和寄存器的位数为 3 位, 其最大可寄存数值为 7 ($J \geq 8$ 时溢出)。若用二进制计算, 则起点坐标为 $O(000, 000)$, 终点坐标为 $E(100, 011)$, $J \geq 1000$ 时溢出。试采用 DDA 法对其进行插补。其插补运算过程如表 2-5 所示。

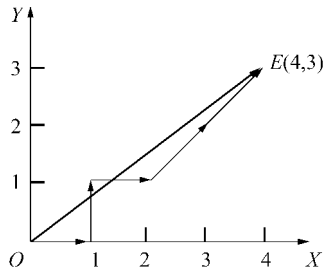


图 2-50 DDA 直线插补实例

表 2-5 DDA 直线插补运算过程

累加次数(Δt)	X 积分器			Y 积分器			终点计数器 J_E
	J_{VX}	J_{RX}	ΔX	J_{VY}	J_{RY}	ΔY	
0	4 100	0		3 011	0		0 000
1	4 100	$0+4=4$ $000+100=100$		3 011	$0+3=3$ $000+011=011$		1 001
2	4 100	$4+4=8+0$ $100+100=\underline{1000}$	1	3 011	$3+3=6$ $011+011=110$		2 010
3	4 100	$0+4=4$ $000+100=100$		3 011	$6+3=8+1$ $110+011=\underline{1001}$	1	3 011
4	4 100	$4+4=8+0$ $100+100=\underline{1000}$	1	3 011	$1+3=4$ $001+011=100$		4 100
5	4 100	$0+4=4$ $000+100=100$		3 011	$4+3=7$ $100+011=111$		5 101
6	4 100	$4+4=8+0$ $100+100=\underline{1000}$	1	3 011	$7+3=8+2$ $111+011=\underline{1010}$	1	6 110
7	4 100	$0+4=4$ $000+100=100$		3 011	$2+3=5$ $010+011=101$		7 111
8	4 100	$4+4=8+0$ $100+100=\underline{1000}$	1	3 011	$5+3=8+0$ $101+011=\underline{1000}$	1	8 <u>1000</u>

2) 数字积分法圆弧插补

第 I 象限顺圆如图 2-51 所示, 圆弧的圆心在坐标原点 O , 起点坐标为 $A(X_a, Y_a)$, 终点坐标为 $B(X_b, Y_b)$ 。

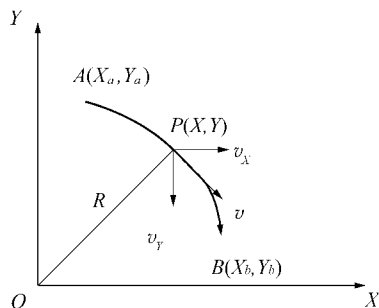


图 2-51 DDA 圆弧插补

圆弧插补时,要求刀具沿圆弧切线作等速运动,设圆弧上某一点 $P(X,Y)$ 的速度为 v ,则在两个坐标方向的分速度为 v_x, v_y ,根据图中几何关系,得则关系式为

$$\frac{v}{R} = \frac{v_x}{Y} = \frac{v_y}{X} = k$$

对于时间增量而言,在 X, Y 坐标轴的位移增量分别为

$$\Delta X = v_x \Delta t = k Y \Delta t$$

$$\Delta Y = v_y \Delta t = -k X \Delta t$$

由于第一象限顺圆对应的 Y 坐标值逐渐减小,所以 v_x 和 v_y 均取绝对值计算。

与 DDA 直线插补类似,DDA 圆弧插补也可用两个积分器来实现圆弧插补,如图 2-52 所示。

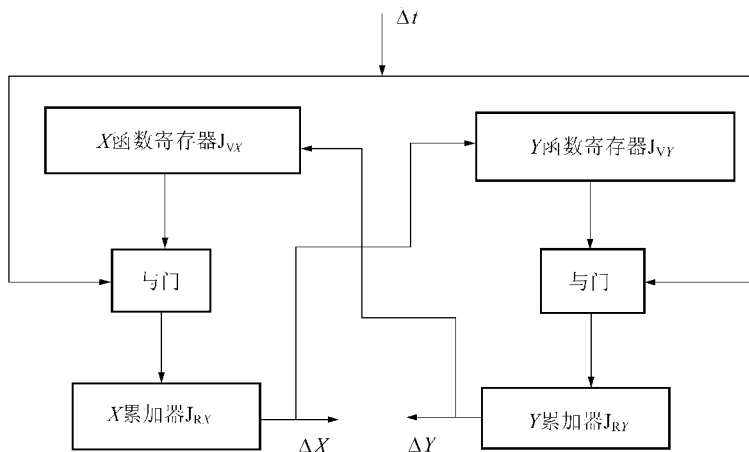


图 2-52 第 I 象限顺圆弧插补器

DDA 圆弧插补与直线插补的主要区别为:

圆弧插补中被积函数寄存器寄存的坐标值与对应坐标轴积分器的关系恰好相反;圆弧插补中被积函数是变量,直线插补的被积函数是常数;圆弧插补终点判别需采用两个终点计数器。对于直线插补,如果寄存器位数为 n ,则无论直线长短都需迭代 2^n 次到达终点。

例 2-4 设有第 I 象限顺圆 AB ,如图 2-53 所示,起点坐标为 $A(0,5)$,终点坐标为 $B(5,0)$,所选寄存器位数 $n=3$ 。若用二进制计算,起点坐标为 $A(000,101)$,终点坐标为 $B(101,000)$,则试用 DDA 法对此圆弧进行插补。其插补运算过程如表 2-6 所示。

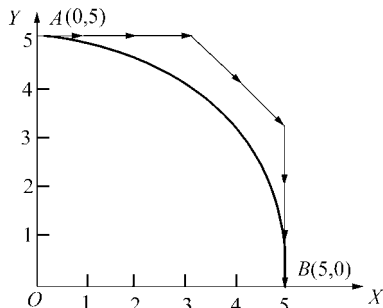


图 2-53 DDA 圆弧插补实例

表 2-6 DDA 圆弧插补运算过程

累加次数(Δt)	X 积分器				Y 积分器			
	J_{VX}	J_{RX}	ΔX	J_{EX}	J_{VY}	J_{RY}	ΔY	J_{EY}
0	5	0		5	0	0		5
	101			101	000	000		101
1	5	0+5=5		5	0	0+0=0		5
	101	000+101=101		101	000	000+000=000		101
2	5	5+5=8+2	1	4	0	0+0=0		5
	101	101+101=1010		100	000	000+000=000		101
					1			
					001			
3	5	5+2=7		4	1	1		5
	101	101+010=111		100	001	001+000=001		101
4	5	5+7=8+4	1	3	1	1+1=2		5
	101	101+111=1100		011	001	001+001=010		101
					2			
					010			
5	5	5+4=8+1	1	2	2	2+2=4		5
	101	101+100=1001		010	010	010+010=100		101
					3			
					011			
6	5	5+1=6		2	3	3+4=7		5
	101	101+001=110		010	011	011+100=111		101
7	5	5+6=8+3	1	1	3	3+7=8+2	1	4
	101	101+110=1011		001	011	011+111=1010		100
	4				4			
	100				100			
8	4	4+3=7		1	4	4+2=6		4
	100	100+011=111		001	100	100+010=110		100
9	4	4+7=8+3	1	0	4	4+6=8+2	1	3
	100	100+111=1011		000	100	100+110=1010		011
	3				5			
	011				101			
10	3	停止			5	5+2=7		3
	011				101	101+010=111		011
11	3				5	5+7=8+4	1	2
	011				101	101+111=1100		010
	2							
	010							

(续表)

累加次数(Δt)	X 积分器				Y 积分器			
	J_{VX}	J_{RX}	ΔX	J_{EX}	J_{VY}	J_{RY}	ΔY	J_{EY}
12	2				5	$5+4=8+1$	1	1
	010				101	$101+100=1001$		001
	1							
	001							
13	1				5	$5+1=6$		1
	001				101	$101+001=110$		001
14	1				5	$5+6=8+3$	1	0
	001				101	$101+110=1011$		000
	0							
	000							
15	0				5	停止		
					101			

3) 数字积分法插补的象限处理

DDA 插补不同象限直线和圆弧时,用绝对值进行累加,把进给方向另做讨论。DDA 插补是沿着工件切线方向移动,四个象限直线进给方向如图 2-54(a)所示;四个象限圆弧进给方向如图 2-54(b)所示。

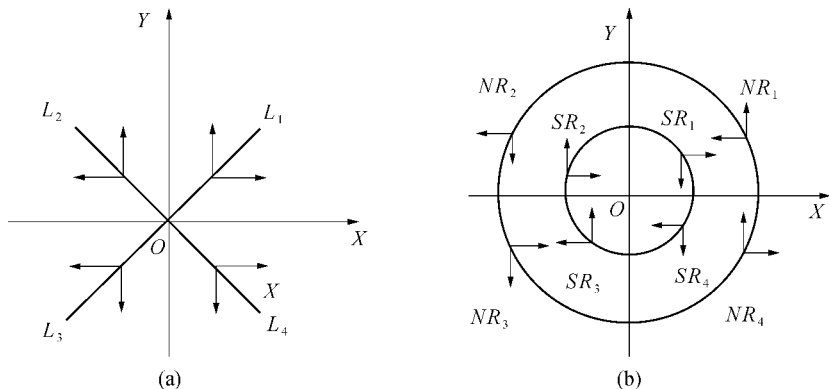


图 2-54 四象限直线和圆弧插补的进给方向

圆弧插补时被积函数是动点坐标,在插补过程中要进行修正,坐标值的修改要看动点运动是使该坐标绝对值是增加还是减少,来确定是加 1 还是减 1。四个象限直线进给方向和以及坐标修正如表 2-7 所示。圆弧插补进给方向及坐标修正如表 2-8 所示。

表 2-7 直线进给方向

		L_1	L_2	L_3	L_4
进给	ΔX	+	-	-	+
修正	J_{VY}				
进给	ΔY	+	+	-	-
修正	J_{VX}				

表 2-8 圆弧插补进给方向及坐标修正

		NR ₁	NR ₂	NR ₃	NR ₄	SR ₁	SR ₂	SR ₃	SR ₄
进给	ΔX	-	-	+	+	+	+	-	-
修正	J_{VY}	-1	+1	-1	+1	+1	-1	+1	-1
进给	ΔY	+	-	-	+	-	+	+	-
修正	J_{VX}	+1	-1	+1	-1	-1	+1	-1	+1

4) 数字积分法合成进给速度

数字积分法的特点是,脉冲源每产生一个脉冲,作一次累加计算,如果脉冲源频率为 f_g (Hz),插补直线的终点坐标为 $E(X_e, Y_e)$,则 X 、 Y 方向的平均进给频率 f_X 、 f_Y 为

$$f_X = \frac{X_e}{m} f_g$$

$$f_Y = \frac{Y_e}{m} f_g$$

$$m = 2^n$$

式中, m 为累加次数。

若已知脉冲当量,可求得 X 和 Y 方向进给速度为

$$v_X = 60 f_X \delta = 60 \delta f_g \frac{X_e}{m}$$

$$v_Y = 60 f_Y \delta = 60 \delta f_g \frac{Y_e}{m}$$

合成进给速度为

$$v = \sqrt{v_X^2 + v_Y^2} = v_g \frac{L}{m}$$

式中, L 为被插补直线长度, $L = \sqrt{X_e^2 + Y_e^2}$,若为圆弧插补, L 应为圆弧半径 R ; v_g 为脉冲源速度, $v_g = 60 \delta f_g$ 。

数控加工程序中 F 代码指定进给速度后, f_g 基本维持不变,这样合成进给速度 v 与被插补直线的长度或圆弧的半径成正比。如图 2-55 所示,如果寄存器位数是 n ,加工直线 L_1 、 L_2 都要经过 $m = 2^n$ 累加运算, L_1 直线短,进给慢,速度低; L_2 直线长,进给快,速度高。加工 L_1 生产效率低;加工 L_2 零件表面质量差。

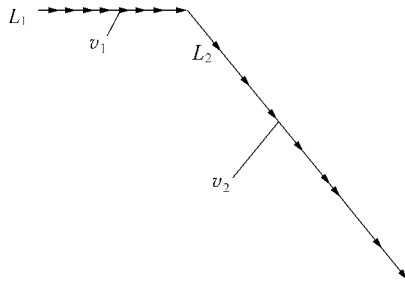


图 2-55 进给速度与直线长度的关系

2.4.4 数据采样插补法

1. 数据采样法原理

数据采样插补又称为时间分割法,与基准脉冲插补法不同,数据采样插补法得出的不是进给脉冲,而是用二进制表示的进给量。这种方法是根据编程进给速度 F ,将给定轮廓曲线按插补周期 T (某一单位时间间隔)分割为插补进给段(轮廓步长),即用一系列首尾相连的微小线段来逼近给定曲线。每经过一个插补周期就进行一次插补计算,算出下一个插补点,即算出插补周期内各坐标轴的进给量,如果相等,则得出下一个插补点的指令位置。

插补周期越长,插补计算误差越大,插补周期应尽量选得小一些。CNC 系统在进行轮廓插补控制时,除完成插补计算外,数控装置还必须处理一些其他任务,如显示、监控、位置采样及控制等。因此,插补周期应大于插补运算时间和其他实时任务所需时间之和。插补周期大约在 8 ms 左右。

采样是指由时间上连续信号取出不连续信号,对时间上连续的信号进行采样,就是通过一个采样开关 K (这个开关 K 每隔一定的周期 T_c 闭合一次)后,在采样开关的输出端形成一连串的脉冲信号。这种把时间上连续的信号转变成时间上离散的脉冲系列的过程称为采样过程,周期 T_c 叫采样周期。

计算机定时对坐标的实际位置进行采样,采样数据与指令位置进行比较,得出位置误差用来控制电动机,使实际位置跟随指令位置。对于给定的某个数控系统,插补周期 T 和采样周期 T_c 是固定的,通常 $T \geq T_c$,一般要求 T 是 T_c 的整数倍。对于直线插补,不会造成轨迹误差。在圆弧插补中,则会带来轨迹误差。

如图 2-56 所示,用弦线逼近圆弧,其最大径向误差 e_r 为

$$e_r = R \left(1 - \cos \frac{\delta}{2} \right)$$

式中, R 为被插补圆弧半径, mm ; δ 为角步距,在一个插补周期内逼近弦所对应的圆心角。

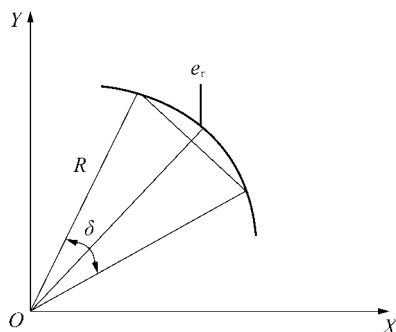


图 2-56 弦线逼近圆弧

将 $\cos(\delta/2)$ 用幂级数展开,得

$$e_r = R \left(1 - \cos \frac{\delta}{2} \right) = R \left\{ 1 - \left[1 - \frac{(\delta/2)^2}{2!} + \frac{(\delta/2)^4}{4!} - \dots \right] \right\} \approx \frac{\delta^2}{8} R$$

设 T 为插补周期, F 为进给速度,则轮廓步长为

$$l = TF$$

用轮廓步长代替弦长,有

$$\delta \approx \frac{1}{R} = \frac{TF}{R}$$

由此可得

$$e_r = \frac{(TF)^2}{8R}$$

可见,圆弧插补过程中,用弦线逼近圆弧时,插补误差 e_r 与程编进给速度 F 的平方、插补周期 T 的平方成正比,与圆弧半径 R 成反比。

2. 直线函数法

1) 直线插补

设要加工如图 2-57 所示直线 OE , 起点在坐标原点 O , 终点为 $E(X_e, Y_e)$, 直线与 X 轴夹角为 α , 则有

$$\tan \alpha = \frac{Y_e}{X_e}, \quad \cos \alpha = \frac{1}{\sqrt{1 + \tan^2 \alpha}}$$

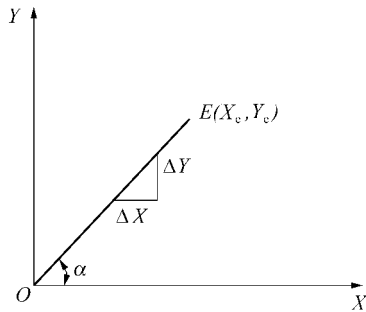


图 2-57 直线插补原理

若已计算出轮廓步长,从而求得本次插补周期内各坐标轴进给量为

$$\Delta X = l \cos \alpha$$

$$\Delta Y = \frac{Y_e}{X_e} \Delta X$$

2) 圆弧插补

圆弧插补,需先根据指令中的进给速度 F , 计算出轮廓步长 l , 再进行插补计算。以弦线逼近圆弧,就是以轮廓步长为圆弧上相邻两个插补点之间的弦长,由前一个插补点的坐标和轮廓步长来计算后一插补点,实质上是求后一插补点到前一插补点两个坐标轴的进给量 ΔX 和 ΔY 。如图 2-58 所示, $A(X_i, Y_i)$ 为当前点, $B(X_{i+1}, Y_{i+1})$ 为插补后到达的点,图中 AB 弦正是圆弧插补时在一个插补周期的步长 l , 需计算 X 轴和 Y 轴的进给量 $\Delta X = X_{i+1} - X_i$, $\Delta Y = Y_{i+1} - Y_i$ 。 AP 是圆弧在 A 点的切线, M 是弦的中点, $OM \perp AB$, $ME \perp AG$, E 为 AG 的中点。圆心角的计算公式为

$$\phi_{i+1} = \phi_i + \delta$$

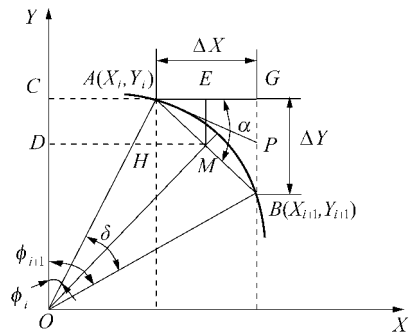


图 2-58 圆弧插补

式中, δ 为轮廓步长所对应的圆心角增量, 也称为角步距。

因为 $OA \perp AP$ (AP 为圆弧切线)

所以 $\triangle AOC \sim \triangle PAG$

则 $\angle AOC = \angle GAP = \phi_i$

因为 $\angle PAB + \angle OAM = 90^\circ$

所以 $\angle PAB = \angle AOM = \frac{1}{2} \angle AOB = \frac{1}{2} \delta$

设 $\alpha = \angle GAB = \angle GAP + \angle PAB = \phi_i + \frac{1}{2} \delta$

$\triangle MOD$ 中 $\tan\left(\phi_i + \frac{1}{2} \delta\right) = \frac{DM}{OD} = \frac{DH + HM}{OC - CD}$

将 $DH = X_i, OC = Y_i, HM = \frac{1}{2} l \cos \alpha = \frac{1}{2} \Delta X, CD = \frac{1}{2} l \sin \alpha = \frac{1}{2} \Delta Y$ 代入上式, 则有

$$\tan \alpha = \tan\left(\phi_i + \frac{1}{2} \delta\right) = \frac{X_i + \frac{1}{2} l \cos \alpha}{Y_i - \frac{1}{2} l \sin \alpha} = \frac{X_i + \frac{1}{2} \Delta X}{Y_i - \frac{1}{2} \Delta Y}$$

又因为 $\tan \alpha = \frac{GB}{GA} = \frac{\Delta Y}{\Delta X}$

由此可以推出 (X_i, Y_i) 与 $\Delta X, \Delta Y$ 的关系式为

$$\frac{\Delta Y}{\Delta X} = \frac{X_i + \frac{1}{2} \Delta X}{Y_i - \frac{1}{2} \Delta Y} = \frac{X_i + \frac{1}{2} l \cos \alpha}{Y_i - \frac{1}{2} l \sin \alpha}$$

上式反映了圆弧上任意相邻两插补点坐标之间的关系, 只要求得 ΔX 和 ΔY , 就可以计算出新的插补点 $B(X_{i+1}, Y_{i+1})$ 的坐标值为

$$X_{i+1} = X_i + \Delta X$$

$$Y_{i+1} = Y_i + \Delta Y$$

且 $\sin \alpha$ 和 $\cos \alpha$ 均为未知, 求解较困难。为此, 采用近似算法, 用 $\sin 45^\circ$ 和 $\cos 45^\circ$ 代替 $\sin \alpha$ 和 $\cos \alpha$, 即

$$\tan \alpha' = \frac{X_i + \frac{1}{2} l \cos 45^\circ}{Y_i - \frac{1}{2} l \sin 45^\circ}$$

$\tan \alpha'$ 与 $\tan \alpha$ 不同, 从而造成了 α 的偏差, α' 在 $\alpha = 0$ 处偏差较大。如图 2-59 所示, 由于角 α 成为 α' , 因而影响到 ΔX 值, 使之成为 $\Delta X'$, 即

$$\Delta X' = l \cos \alpha' = AT$$

为保证下一个插补点仍在圆弧上, $\Delta Y'$ 的计算公式为

$$X_i^2 + Y_i^2 = (X_i + \Delta X')^2 + (Y_i + \Delta Y')^2$$

经展开整理得

$$\Delta Y' = \frac{\left(Y_i + \frac{1}{2} \Delta X'\right) \Delta X'}{Y_i - \frac{1}{2} \Delta X'}$$

用迭代法可解出 $\Delta Y'$ 。

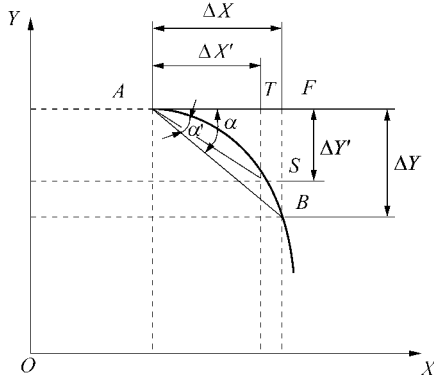


图 2-59 近似处理引起的进给速度偏差

采用近似算法可保证每次插补点均在圆弧上,引起的偏差仅是 $\Delta X \rightarrow \Delta X'$, $\Delta Y \rightarrow \Delta Y'$, $AB \rightarrow AS$ 即 $l \rightarrow l'$ 。这种算法仅造成每次插补进给量的微小变化,而使进给速度有偏差,实际进给速度的变化小于指令进给速度的 1%,在加工中是允许的。

直线函数法是用弦线逼近圆弧,因此插补误差主要为半径的绝对误差。因插补周期是固定的,该误差取决于进给速度和圆弧半径,当加工的圆弧半径确定后,为了使径向绝对误差不超过允许值,对进给速度要有一个限制,即

$$F \leq \frac{8e_r R}{T}$$

式中, e_r 为最大径向误差; R 为圆弧半径。

3. 扩展 DDA 数据采样插补

1) 扩展 DDA 直线插补

假设根据编程进给速度,要在时间段 T_1 内走完如图 2-60 所示直线,其起点为坐标原点 O ,终点坐标为 $E(X_e, Y_e)$, v 为进给速度(零件加工程序中记为 F), v_x 与 v_y 分别为 X 、 Y 坐标的分速度。则有

$$v_x = \frac{X_e}{\sqrt{X_e^2 + Y_e^2}} v$$

$$v_y = \frac{Y_e}{\sqrt{X_e^2 + Y_e^2}} v$$

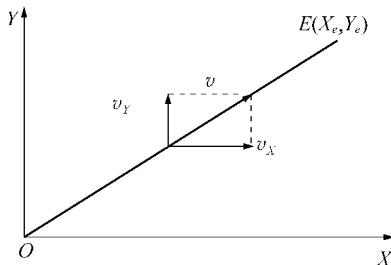


图 2-60 扩展 DDA 直线插补

将时间 T_1 用插补周期 T 分割为 n 个子区间 (n 取 $\geq \frac{T_1}{T}$ 最接近的整数), 从而在每个插补周期 T 内的坐标增量分别为

$$\Delta X = v_x T = \frac{v}{\sqrt{X_e^2 + Y_e^2}} T X_e = \text{FRN} \lambda_t X_e$$

$$\Delta Y = v_y T = \frac{v}{\sqrt{X_e^2 + Y_e^2}} T Y_e = \text{FRN} \lambda_t Y_e$$

式中, v 为编程的进给速度, mm/min; T 为插补周期, ms; λ_t 为根据插补周期换算后的时间常数, $\lambda_t = T \times 10^{-3} / 10$; FRN 为进给速率数, $\text{FRN} = \frac{v}{\sqrt{X_e^2 + Y_e^2}} = \frac{v}{L}$; L 为插补直线长度, mm。

对于同一直线来说, 由于 v 和 X_e 、 Y_e , 以及 T 均为常数, 可以记作 λ_d , 称为步长系数。则

$$\Delta X = \lambda_d X_e$$

$$\Delta Y = \lambda_d Y_e$$

故同一条直线在每个采样周期内增量 ΔX 和 ΔY 的常数均相同, 在每个插补周期内算出后, 就可以得到本插补周期末的刀具位置坐标值

$$X_{i+1} = X_i + \Delta X$$

$$Y_{i+1} = Y_i + \Delta Y$$

由于插补中每次迭代形成的子线段斜率 ($\Delta Y / \Delta X$) 等于给定直线斜率, 从而保证了轨迹要求。

2) 扩展 DDA 圆弧插补

如图 2-61 所示为第一象限顺圆弧 $A_i Q$, 圆心在坐标原点 O , 半径为 R 。设刀具处在 $A_i (X_i, Y_i)$ 点位置。若在一个插补周期 T 内, 用 DDA 插补法沿切线方向进给的步长为 l , 一个插补周期后达到 C 点, 即 $A_i C = l$, 径向误差较大。扩展 DDA 插补算法, 就是将切线逼近圆弧转化为割线逼近圆弧, 以减少插补误差, 具体步骤如下:

先通过 $A_i C$ 微小线段的中点 B 作以 OB 为半径的圆弧的切线 BD , 再通过 A_i 点作 BD 的平行线 $A_i H$, 即 $A_i H \parallel BD$, 并在 $A_i H$ 上截取 $A_i A_{i+1} = A_i C = l$, 如果 OB 与 $A_i H$ 的交点为 M , 在 $\triangle A_i M B$ 中, 斜边 $A_i B = \frac{1}{2} l$, 直角边 $A_i M < \frac{1}{2} l$, 所以 A_{i+1} 点落在圆弧外侧。扩展 DDA 用割线 $A_i A_{i+1}$ 线段代替切线, 使径向误差减少了。

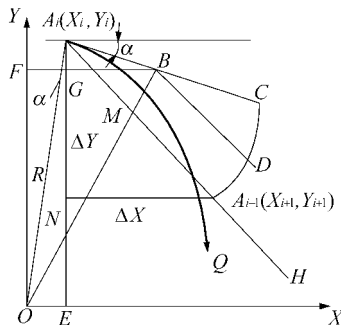


图 2-61 扩展 DDA 圆弧插补

下面计算在插补周期 T 内轮廓进给步长 l 的坐标分量 ΔX 和 ΔY , 就可以得到本次插补周期后到达的位置 A_{i+1} 。

在图 2-61 中, 在 $\triangle OEA_i$ 中, OA_i 和 A_iE 的夹角为 α , 则

$$\sin \alpha = \frac{OE}{OA_i} = \frac{X_i}{R}$$

$$\cos \alpha = \frac{A_iE}{OA_i} = \frac{Y_i}{R}$$

设刀具以恒速进给, 即在每个插补周期 T 内的进给速度均为 v , 则 $A_iA_{i+1} = l = vT$ 。过 B 点作 X 轴的平行线 BF 交 Y 轴于 F 点, 交 A_iE 线段于 G 点。过 A_{i+1} 点作 $A_{i+1}N$ 平行 X 轴, 交 A_iE 于 N 点。在图 2-61 中, $\triangle OFB$ 与 $\triangle A_iNA_{i+1}$ 相似, 有比例关系为

$$\frac{NA_{i+1}}{A_iA_{i+1}} = \frac{OF}{OB}$$

$$NA_{i+1} = \Delta X$$

$$A_iA_{i+1} = l = vT$$

其中

在 $\triangle A_iGB$ 中, 有

$$A_iG = A_iB \sin \alpha = \frac{1}{2}l \sin \alpha$$

$$OF = A_iE - A_iG = Y_i - \frac{1}{2}l \sin \alpha$$

因此, 在 $\triangle OA_iB$ 中, 有

$$OB = \sqrt{(A_iB)^2 + (OA_i)^2} = \sqrt{\left(\frac{1}{2}l\right)^2 + R^2}$$

由以上分析可得

$$\frac{\Delta X}{l} = \frac{Y_i - \frac{1}{2}l \sin \alpha}{\sqrt{\left(\frac{1}{2}l\right)^2 + R^2}}$$

将式 $\sin \alpha = \frac{X_i}{R}$ 代入上式并整理, 得

$$\Delta X = \frac{l \left(Y_i - \frac{1}{2} \frac{X_i}{R} l \right)}{\sqrt{\left(\frac{1}{2}l\right)^2 + R^2}}$$

因为 $l \ll R$, 故将 $\left(\frac{1}{2}l\right)^2$ 略去不计, 则上式变为

$$\Delta X \approx \frac{l}{R} \left(Y_i - \frac{1}{2} \frac{X_i}{R} l \right) = \frac{v}{R} \lambda_t \left(Y_i - \frac{1}{2} \frac{v}{R} \lambda_t X_i \right)$$

$$\lambda_t = \frac{1}{60} T \times 10^{-3}$$

若令

$$\lambda_d = \frac{v}{R} \lambda_t = FRN \lambda_t$$

则

$$\Delta X = \lambda_d \left(Y_i - \frac{1}{2} \lambda_d X_i \right)$$

另外, 从 $\triangle OFB$ 与 $\triangle A_iNA_{i+1}$ 相似, 还可以得出

$$\frac{A_i N}{A_i A_{i+1}} = \frac{FB}{OB}$$

$$\frac{A_i N}{A_i A_{i+1}} = \frac{FG+GB}{OB}$$

已知

$$OB = \sqrt{\left(\frac{1}{2}l\right)^2 + R^2}$$

由直角 $\triangle A_i GB$ 得

$$GB = A_i B \cos \alpha = \frac{1}{2} \frac{Y_i}{R} l$$

$$FG = X_i$$

而

$$\Delta Y = A_i N = \frac{l \left(X_i + \frac{1}{2} \frac{Y_i}{R} l \right)}{\sqrt{\left(\frac{1}{2}l\right)^2 + R^2}}$$

同理,因 $l \ll R$,故 $\left(\frac{1}{2}l\right)^2$ 略去不计,则

$$\Delta Y \approx \frac{1}{R} \left(X_i + \frac{1}{2} \frac{Y_i}{R} l \right)$$

仍令

$$\Delta Y \approx \frac{l}{R} \left(X_i + \frac{1}{2} \frac{Y_i}{R} l \right)$$

$$\lambda_t = \frac{1}{60} T \times 10^{-3}$$

则

$$\Delta Y = \lambda_d \left(X_i + \frac{1}{2} \lambda_d Y_i \right)$$

由于 $A_i(X_i, Y_i)$ 为已知,因而可算出本次插补周期刀具应达到的坐标位置 (X_{i+1}, Y_{i+1}) ,即

$$X_{i+1} = X_i + \Delta X$$

$$Y_{i+1} = Y_i - \Delta Y$$

2.5 可编程控制器 PLC

世界上第一台可编程控制器是1969年美国数字设备公司(DEC)研制成功的。最初,美国通用汽车公司(GM)根据生产的需要提出如下设想:能否把计算机的功能完善、灵活性好、通用性强等优点和继电器控制系统的简单易懂、操作方便、价格便宜等优点结合起来,做成一种通用控制装置,并把计算机的编程方法和程序输入方式加以简化,用面向控制过程、面向问题的“自然语言”编程,使得不熟悉计算机的人也能方便地使用。这样,使用人员不必在编程上花费大量的精力,而是集中力量去考虑如何发挥该装置的功能和作用。这一设想提出后,美国DEC公司首先响应,于1969年首先研制成功第一台可编程控制器,型号为PDP-14。它代替传统的继电器控制系统,在美国GE公司的汽车自动装配上试用获得了成功,并取得了满意的效果。可编程控制器由此诞生。

可编程控制器自问世以来,发展极为迅速。1971年,日本开始生产,1973年,欧洲也开

始生产。现在,世界各国的一些著名的电气工厂几乎都在生产可编程控制器装置。可编程控制器已作为一个独立的工业设备被列入生产中,成为当代电控装置的主导。

可编程控制器是在继电器控制和计算机控制的基础上开发出来的,并逐渐发展成以微处理器为核心,把自动化技术、计算机技术、通讯技术融为一体的新型工业自动控制装置。目前已被广泛地应用于各种生产机械和生产过程的自动控制中。

可编程控制器出现以后,名称很不一致。早期的可编程控制器在功能上只能进行逻辑控制,因此被称为可编程逻辑控制器(Programmable Logic Controller,简称为 PLC)。

随着技术的发展,国外一些厂家开始采用微处理器(Micro Processor Unit,简称为 MPU)来作为可编程控制器的中央处理单元(Center Processing Unit,简称为 CPU),从而扩大了控制器的功能。它不仅可以进行逻辑控制,而且还可以对模拟量进行控制,因此美国电气制造协会(National Electrical Manufacturers Association,简称为 NEMA)于1980年将它正式命名为可编程控制器(Programmable Controller,简称为 PC),并给 PC 作了如下定义:PC 是一种数字式的电子装置,它使用了可程序的存储器以存储指令,能完成逻辑、顺序、计时、计数和算术运算等功能,用以控制各种机械或生产过程。

国际电工委员会(IEC)于1985年在其颁布的可编程控制器标准草案第二稿中,又给 PC 作了如下定义:可编程控制器是一种数字运算操作的电子系统,专为工业环境下应用而设计,它采用可程序的存储器,用来在其内部存储执行逻辑运算、顺序控制、定时、计数和算术运算等操作的指令,并通过数字式、模拟式的输入和输出,控制各种机械或生产过程。

近年来,PC 技术发展很快,每年都推出不少新产品,其功能已超出上述定义范围。PC 这一名称在国外工业界已使用多年,但是近年来 PC 这个名字又成为个人计算机(Personal Computer)的专称,为了区别,现在也常把可编程控制器称为 PLC。

2.5.1 可编程控制器的结构与工作方式

在传统继电器接触器控制系统中,支配控制系统工作的“程序”是由导线将电气元件连接起来实现的,我们把这样的控制系统称之为“硬接线”程序控制系统。在这种接线控制系统中,控制功能的改变必须通过修改控制器件和接线来实现。而可编程控制系统是通过修改 PLC 的程序来完成。可编程控制系统也称为“软接线”程序控制系统。PLC 控制系统与微型计算机控制系统基本相似,它是由硬件和软件两大部分组成。PLC 实质上是一种用于工业控制的专用计算机,但对硬件各部分的定义及工作过程则与 PC 有很大差异。

1. 可编程控制器的结构

PLC 实质上是一种工业控制计算机,只不过它比一般计算机具有更强的与工业过程相连接的接口和更直接的适应于控制要求的编程语言,故 PLC 与计算机的组成十分相似。其硬件构成如图 2-62 所示。

1) 中央处理单元 CPU

与一般计算机一样,CPU 是 PLC 的核心,它按 PLC 中系统程序赋予的功能指挥 PLC 有条不紊地进行工作。其主要任务有:控制从编程器键入的用户程序和数据的接收与存储;用扫描的方式通过 I/O 部件接收现场的状态或数据,并存入输入映像存储器或数据存储器中;诊断 PLC 内部电路的工作故障和编程中的语法错误等;PLC 进入运行状态后,从存储器逐条读取用户指令,经过命令解释后按指令规定的任务进行数据传送、逻辑或算术运算等;根据运算等结果,更新有关标志位的状态和输出映像存储器的内容,再经输出部件实现输出

控制、制表打印或数据通信等功能。

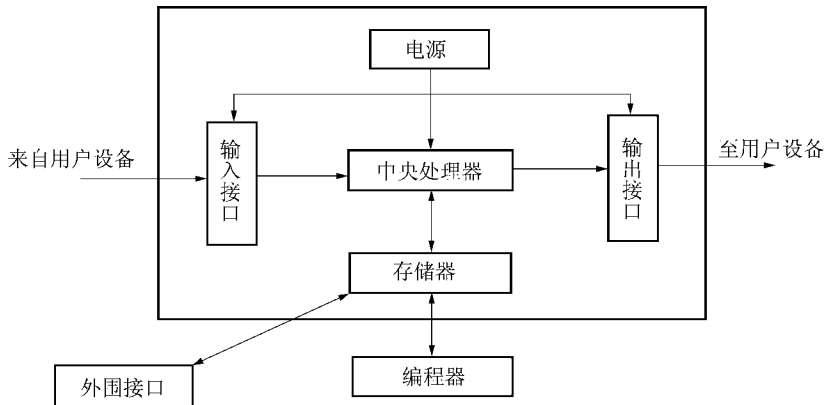


图 2-62 PLC 的硬件结构框图

不同型号 PLC 的 CPU 芯片是不同的,有采用通用 CPU 芯片的,如 8031、8051、8086、80286 等,也有采用厂家自行设计的专用 CPU 芯片的。CPU 芯片的性能关系到 PLC 处理控制信号的能力与速度,CPU 位数越高,系统处理的信息量越大,运算的速度也越快。随着芯片技术的不断发展,PLC 所用的 CPU 芯片也越来越高档。

2) 存储器

PLC 的存储器包括系统和用户存储器两部分。

系统存储器用来存放由 PLC 生产厂家编写的系统程序,并固化在 ROM 内,用户不能直接更改。它使 PLC 具有基本的智能,能够完成 PLC 设计者规定的各项工作。系统程序质量的好坏,很大程度上决定了 PLC 的性能。其内容主要包括 3 部分:

(1) 系统管理程序。它主管控制 PLC 的运行,使整个 PLC 按部就班地工作。

(2) 用户指令解释程序。通过用户指令解释程序,将 PLC 的编程语言变为机器语言指令,再由 CPU 执行这些指令。

(3) 标准程序模块与系统调用程序。它包括许多不同功能的子程序及其调用管理程序,如完成输入、输出及特殊运算等的子程序。PLC 的具体工作都是由这部分程序来完成的,这部分程序的多少决定了 PLC 性能的强弱。

用户存储器包括用户程序存储器(程序区)和功能存储器(数据区)两部分。用户程序存储器用来存放用户针对具体控制任务,用规定的 PLC 编程语言编写的各种用户程序。用户程序存储器根据所选用的存储器单元类型的不同,可以是 RAM(有掉电保护)、EPROM 或 EEPROM 存储器,其内容可以由用户任意修改或增删。用户功能存储器是用来存放(记忆)用户程序中使用的 ON/OFF 状态、数值数据等。它构成 PLC 的各种内部器件,也称“软元件”。用户存储器容量的大小,关系到用户程序容量的大小和内部器件的多少,是反映 PLC 性能的重要指标之一。

3) 输入/输出(I/O)模块

I/O 模块是可编程控制器与现场 I/O 装置或其他外部设备的连结部件,可编程控制器的 I/O 能直接加到执行元件上。为了满足更加复杂的控制功能的需要,PLC 配有多种智能 I/O 接口。例如,满足位置调节需要的位置闭环控制模块,对高速脉冲进行计数和处理的高速计数模块等。这类智能模块都有其自身的处理器系统。

4) 编程器

编程器的作用是供用户进行程序的编制、编辑、调试和监视。编程器有简易型和智能型两类。简易型的编程器只能联机编程,且往往需要将梯形图转化为机器语言助记符(指令表)后才能输入,一般由简易键盘和发光二极管或其他显示器件组成。智能型的编程器又称图形编程器,可以联机或脱机编程,具有 LCD 或 CRT 图形显示功能,能够直接输入梯形图并通过屏幕对话。

5) 电源

小型整体式可编程控制器内部有一个开关式稳压电源。此电源一方面可为 CPU 板、I/O板及扩展单元提供工作电源(DC5 V);另一方面可为外部输入元件提供 DC24 V。

6) 扩展接口

扩展接口用于将扩展单元与基本单元相连,使 PLC 的配置更加灵活。

2. 可编程控制器工作方式

PLC 虽然以微处理器为核心,具有微型计算机的许多特点,但它的工作方式却与微型计算机有很大的不同,微型计算机一般采用等待命令或中断的工作方式,如常见的键盘扫描方式或 I/O 扫描方式,当有键按下或 I/O 动作,则转入相应的子程序或中断服务程序,无键按下,则继续扫描等待。PLC 采用循环扫描的工作方式,即顺序扫描,不断循环这种工作方式是在系统软件控制下进行的。当 PLC 运行时,CPU 根据用户按控制要求编写好并存于用户存储器中的程序,按序号作周期性的程序循环扫描,程序从第一条指令开始,逐条顺序执行用户的程序直到程序结束。然后重新返回第一条指令,再开始下一次扫描;如此周而复始。实际上,PLC 扫描工作除了执行用户程序外,还要完成其他工作,整个工作过程分为自诊断、通信服务、输入处理、输出处理、程序执行 5 段,如图 2-63 所示。

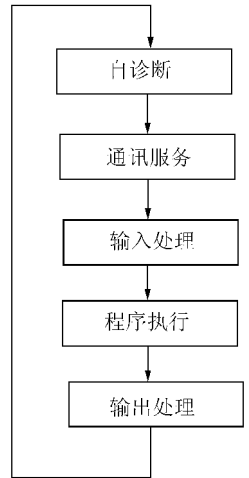


图 2-63 PLC 循环扫描示意图

1) 自诊断

每次扫描用户程序之前,都先执行故障自诊断程序。自诊断内容包括 I/O 部分、存储器、CPU 等,并通过 CPU 设置定时器来监视每次扫描是否超过规定的时间,如果发现异常,则停机并显示出错。若自诊断正常,则继续向下扫描。

2) 通信服务

PLC 检查是否有与编程器、计算机等的通信要求,若有则进行相应处理。

3) 输入处理

PLC 在输入刷新阶段,首先以扫描方式按顺序从输入寄存器中写入所有输入端子的状态或数据,并将其存入内存中为其专门开辟的暂存区——输入状态映像区中,这一过程称为输入采样,或是刷新。随后关闭输入端口,进入程序执行阶段,即使输入端有变化,输入映像区的内容也不会改变。变化的输入信号的状态只能在下一个扫描周期的输入刷新阶段被读入。

4) 输出处理

同输入状态映像区一样,PLC 内存中也有一块专门的区域称为输出状态映像区。当程序的所有指令执行完毕,输出状态映像区中所有输出继电器的状态就在 CPU 的控制下被一

次集中送至输出锁存器中,并通过一定的输出方式输出,推动外部的相应执行器件工作,这就是 PLC 输出刷新阶段。

5) 程序执行

PLC 在程序执行阶段,按用户程序顺序扫描执行每条指令。从输入状态映像区读出输入信号的状态,经过相应的运算处理等,将结果写入输出状态映像区。通常将自诊断和通讯服务合称为监视服务,输入刷新和输出刷新称为 I/O 刷新。可以看出,PLC 在一个扫描周期内,对输入状态的扫描只是在输入采样阶段进行,对输出的赋值也只有在输出刷新阶段才能被送出,而在程序执行阶段输入、输出会被封锁。这种方式称为集中采样、集中输出。

当 PLC 投入运行后,重复完成以上 3 个阶段的工作,即采用循环扫描工作过程。PLC 工作的主要特点是输入输出采样,程序执行,输出刷新“串行”工作方式,这样既可避免继电器、接触器控制系统中的触点竞争和时序失配,也可提高 PLC 的运算速度。这是 PLC 系统可靠性高、响应快的原因。但是又会导致输出对输入在时间上的滞后。

PLC 在执行程序时所用到的状态值不是直接从实际输入口所获得,而是来源于输入映像区和输出映像区。输入映像区的状态取决于上一扫描周期从输入端子中采样取得的数据,并在程序执行阶段保持不变。输出映像区的状态取决于执行程序输出指令的结果。输出锁存器中的状态值是上一个扫描周期的输出刷新结果。

2.5.2 可编程控制器的分类与特点

1. 可编程控制器的分类

目前,PLC 的品种很多,规格性能不一,且没有一个权威的统一分类标准,一般按下面几种情况大致分类。

1) 按结构形式分类

PLC 可分为整体式和模块式两种。

整体式 PLC 是将其电源、中央处理器、输入/输出部件等集中配置在一起,有的甚至全部安装在一块印刷电路板上。整体式 PLC 结构紧凑、体积小、质量小、价格低、I/O 点数固定、使用不灵活。小型 PLC 常使用这种结构。

模块式 PLC 是将各部分以模块形式分开,如电源模板、CPU 模板、输入模板、输出模板等。把这些模板插入机架底板上,组装在一个机架内。这种结构配置灵活,装配方便,便于扩展。一般中型和大型 PLC 常采用这种结构。

2) 按功能分类

按 PLC 功能强弱来分,可分为低档机,中档机,高档机三种。

低档 PLC 具有逻辑运算、定时、计数等功能,有的还增设模拟量处理、算术运算、数据传送等功能。

中档 PLC 除具有低档机的功能外,还具有较强的模拟量输入/输出、算术运算、数据传送等功能,可完成既有开关量又有模拟量控制的任务。

高档 PLC 增设有带符号算术运算及矩阵运算等,使运算能力更强,还具有模拟调节、联网通信、监视、记录和打印等功能,使 PLC 的功能更多更强。能进行远程控制、构成分布式控制系统,成为整个工厂的自动化网络。

3)按输入/输出点数和存储容量分类

在实际应用中,一般都是根据输入/输出的点数多少和程序存储器容量的大小来选择可编程控制器的规模。可编程控制器按规模可分为小型、中型、大型三类,如表 2-9 所示。

表 2-9 可编程控制器的规模

评价指标 PLC 规模	输入/输出的点数	程序存储器容量
小型	小于 128 点	1 KB 以下
中型	128 点~512 点	1~4 KB
大型	512 点以上	4 KB 以上

一般来说,数控车床、数控铣床、加工中心、机器人等单机数控设备,所需的输入/输出的点数都在 128 点以内;少数复杂的数控设备需要 128 点以上;而大型数控设备、FMC、FMS、CIMS 等则需采用中、大型规模的可编程控制器。

2. 可编程控制器的特点

可编程控制器有以下特点:

(1)高可靠性。高可靠性是 PLC 最突出的特点之一。由于工业生产过程是昼夜连续的,一般的生产装置要几个月,甚至几年才大修一次,这就对于工业生产过程的控制器的提出了高可靠性的要求。PLC 之所以具有较高的可靠性是因为它采用了微电子技术,大量的开关动作由无触点的半导体电路来完成,另外还采取了屏蔽、滤波、隔离等抗干扰措施。它的平均故障间隔时间为三至五万小时以上。

(2)灵活性。以前电气工程师必须为每套设备配置专用控制装置。有了可编程控制器,硬件设备采用相同的可编程控制器,只需编写不同应用软件即可,且可以用一台可编程控制器控制几台操作方式完全不同的设备。

(3)便于改进和修正。相对传统的电气控制线路,可编程控制器为改进和修订原设计提供了极其方便的手段。以前也许要花费几周的时间,用可编程控制器只用几分钟就可以完成。

(4)节点利用率提高。传统电路中一个继电器只能提供几个节点用于连锁,在可编程控制器中,一个输入中的开关量或程序中的一个“线圈”可提供用户所需用的任意的连锁节点,也就是说,节点在程序中可不受限制地使用。

(5)具有丰富的 I/O 接口。由于工业控制机只是整个工业生产过程自动控制系统中的一个控制中枢,为了实现对工业生产过程的控制,它还必须与各种工业现场的设备相连接才能完成控制任务。因此 PLC 除了具有计算机的基本部分如 CPU、存储器等以外,还有丰富的 I/O 接口模块。对不同的工业现场信号(如交流、直流、电压、电流、开关量、模拟量、脉冲等),都有相应的 I/O 模块与工业现场的器件或设备(如按钮、行程开关、接近开关、传感器及变送器、电磁线圈、电机启动器、控制阀等)直接连接。另外有些 PLC 还有通信模块、特殊功能模块等。

(6)模拟调试。可编程控制器能对所控功能在实验室内进行模拟调试,缩短现场的调试时间,而传统电气线路是无法在实验室进行调试的,只能花费现场大量时间。

(7)能对现场进行微观监视。操作人员能通过显示器观测到所控每一个节点的运行情况,随时监视事故发生点。

(8)快速动作。传统继电器节点的响应时间一般需要几百毫秒,而可编程控制器里的节点反应很快,内部是微秒级的,外部是毫秒级的。

(9)梯形图及布尔代数并用。可编程控制器的程序编制可采用电气技术人员熟悉的梯形图方式,也可以采用程序员熟悉的布尔代数图形方式。

(10)系统购置的简化。可编程控制器是一个完整的系统,购置了一台可编程控制器,就相当于购买了系统所需要的所有继电器、计数器、计时器等器件。在传统控制中,系统中所需要的继电器、计数器、计时器常常来源于不同的厂家,等货到齐需要很长时间,即使只缺一个继电器,也会推迟整个工期。而可编程控制器能提供足够的备用继电器、计时器、计数器。

(11)体积小,质量轻,功耗低。由于采用半导体集成电路,与传统控制系统相比较,其体积小,质量轻,功耗低。

(12)编程简单,使用方便。PLC采用面向控制过程、面向问题的“自然语言”编程,容易掌握。例如,目前PLC大多数采用的梯形图语言编程方式,它继承了传统控制线路的清晰直观感,考虑到大多数电气技术人员的读图习惯及应用微机的水平,很容易被技术人员所接受,易于编程,程序改变时也易于修改。

当然,PLC也并非十全十美,其价格比继电器控制和一般单片机系统高。

本章小结

本章介绍了CNC的一些基本概念。由于CNC是在普通NC的基础上发展起来的,因此,在实施对机床的数控方面有一些共同之处,如输入格式、插补方式、译码处理等过程。在插补运算方面可以采取一些复杂的高精度算法,还可以采用数据采样方法计算位置增量,而并不直接计算输出脉冲;由于CNC具有存储系统,CNC系统还能预先根据两相邻程序段的数据进行刀补计算及插补预处理等工作;在伺服控制方面又多采取实时性的中断处理等。

CNC系统的硬件结构从使用的微机及结构来分,有单微机和多微机结构。为了实现机床的控制任务,还必须设置一些输入、输出装置,这些装置通常称为外部设备,它们通过相应的接口与数控机床连接,实施信息交换与控制。

为了完成控制机床的任务,CNC系统都有一套专用软件,这就是系统软件。它一般包括输入数据处理、插补计算、位置控制、速度控制、管理和诊断等软件。输入数据处理软件包括对程序段的输入、存储、译码、修改、编辑以及预计算(如刀补计算)等内容。插补计算是CNC系统中实时性很强的一项任务。CNC可以采用逐点比较法、DDA等基准脉冲插补算法外,还常采用数据采样法,即在每个采样周期内计算出下个周期中机床工作台应到达的位置值,将此位置坐标值输出给伺服系统,然后伺服系统再根据具体情况做具体处理(如步进系统时,再算出插补脉冲;闭环系统时,求出跟随误差),以带动机床操作。管理软件和诊断软件的设置使CNC系统的性能更可靠,工作更稳定,提高了使用效率。常见的系统软件结构有前后台型和中断型两种。

本章着重介绍了逐点比较法、数字积分法以及数据采样插补法等多种插补方法。插补方法有多种方式进行分类。以插补输出的信号形式,可以将插补分为基准脉冲插补法和数据采样插补法两类。本章所介绍的数字积分法(DDA)、逐点比较法均属于基准脉冲插补法,

其特点是以脉冲的方式产生输出。数据采样插补法又称为时间分割法,其特点是计算出轮廓线段在每一插补周期内的进给量,边计算边控制加工。

PLC是数控机床的重要控制部分。在本章中主要介绍了PLC的结构、分类、特点及工作方式,详细描述了“循环扫描”工作过程。

习 题 2

2-1 CNC系统有哪些功能?能完成哪些工作?

2-2 单微机处理结构和多微机处理结构各有什么特点?

2-3 CNC系统软件一般包括哪几个部分?

2-4 CNC系统软件处理中的两个突出特征是什么?

2-5 CNC装置的主要功能有哪些?

2-6 利用逐点比较法插补直线 OE ,起点坐标为 $O(0,0)$,终点坐标为 $E(12,15)$,试写出插补过程并绘出轨迹。

2-7 利用逐点比较法插补圆弧 PQ ,起点坐标为 $P(8,0)$,终点坐标为 $Q(0,8)$,半径为 $R8$,试写出插补过程并绘出轨迹。

2-8 数据采样插补和基准脉冲插补的区别何在?

2-9 PLC采取什么工作方式?