第50篇　数字计算机控制系统

主编单位：

国家仪器仪表总局重庆工业自动化仪表研究所

主　　编：

苏松基

常 用 符 号 表

A——累加器
ALU——算术逻辑部件
CPU——中央处理器
H——高电平
L——低电平
PC——指令计数器
RAM——随机存贮器
ROM——只读存贮器(控制存贮器)
SP——堆栈计数器
T_1——节拍脉冲

A——有效使用率
C_{mr}——共模抑制比(倍)
CMRR——共模抑制比(dB)
I_p——字电流
I_w——位电流
I_X——X方向驱动电流
I_Y——Y方向驱动电流
I_Z——禁止电流

K_u——临界比例度
K_P——比例放大系数
L——过程的纯滞后时间
MTBF——平均故障间隔时间
MTTR——平均故障修理时间
R——可靠性
S——维修效率或给定值
T——过程的等价时间常数
T_D——微分时间
T_I——积分时间
T_L——滤波器时间常数
T_u——临界振荡周期

U_{CM}——共模干扰电压
X——测量值

Y——阀门开度;计算机输出量;目的函数
ΔT——采样周期
ΔY——计算机输出增量

第1章 概 述

电子数字计算机(以下简称计算机)是一种能自动高速进行大量计算工作的电子设备。它具有信息存贮、逻辑判断、精确、快速计算和自动化程度高等特点，因此将计算机应用到工业自动化控制系统中可以提高劳动生产率，改善劳动条件，降低成本，是实现综合自动化和生产过程控制最优化的有力工具。

1 电子数字计算机的产生和发展

世界上第一台电子数字计算机问世于1946年，三十年来计算机及其有关技术得到了突飞猛进的发展。

1956年出现了以晶体管为主要元件的计算机，可靠性有很大提高。1958年计算机开始进入工业生产过程控制的行列，逐渐成为自动控制系统的重要工具，构成了新的计算机控制系统，打开了工业生产过程自动化的新局面。

计算机的技术发展已经经历了三代更新，发展过程如图50·1-1所示。

从七十年代开始，计算机进入了第四代。亿位的存贮容量，毫微秒的操作速度成为第四代计算机的标志。在这些计算机中采用了一些如大规模集成电路存贮器等新技术，结构上以分布式计算概念来组织，操作系统引进数据库，大大加强了计算机的功能，提高了计算机的性能；另一方面由于微型机的发展，使它在生产过程控制、测量仪器终端装置、数控机床、机械手等方面得到日益广泛的应用。

2 计算机在工业控制上的应用

计算机在发展初期，由于结构庞大，价格昂贵，可靠性不高，仅用于科学技术计算方面。随着主机的不断改进与外部设备品种的增多及完善，计算机在信息处理和工业控制等方面得到越来越广泛的应用。

生产技术的发展使得生产规模愈来愈大，速度和强度也愈来愈高，对自动化的要求也日益提高和多样化。表现在检查测量方面，从单一参数的局部测量逐渐发展到多参数或间接指标的测量和计算，巡回检测和数据处理；在控制方面由简单的顺序控制、单回路反馈控制逐渐发展到集中管理、相互关联的反馈控制、前馈控制以至于最优化控制等；此外还提出了某些管理与调度自动化的要求。这不仅要求提高反映工艺状态信息的速度，而且尚须对前后工序给以综合平衡。显然，单纯采用常规仪表控制难于取得预期的效果，这些都促进了计算机在控制上的应用。另一方面，在各种控制装置中，电子数字装置的应用与集成电路的发展又为这种应用提供了方便。这种发展情况如图50·1-2所示。

在生产自动化方面，计算机可以用来进行：

a. 巡回检测与数据处理 对数以百计的过程物理参数周期性地或随机地进行测量并显示、打印记录，对于间接指标或参数可进行计算处理；

b. 顺序控制与数值控制 对复杂的生产过程可按一定顺序进行启、停、开、关等操作，或对工件加工的尺寸进行精密数值控制；

c. 操作指导 对生产过程进行测量，根据测量结果与预期目的作出判断决定下一步应该怎样改变生产进程，将这种决定打印或显示出来供操作人员参考，采纳执行；

d. 直接控制 对生产过程直接进行反馈或前馈控制，代替常规的自动调节器或控制装置，采用分时的形式，一个计算机可以同时控制大量的生产环节；

e. 监督控制 对生产过程不进行直接控制，只是监督生产过程的进行，根据生产过程的状态、环境、原料等因素，按照过程的数学模型(或控制算法)，计算出它的最优状况或当时应采取的控制措施，把这种措施交给在第一线起直接控制的计算机或常规控制工具进行(整定它们的给定值)；

f. 自动管理或调度 对车间或全厂的自动生产线或生产过程进行调度管理。

在计算机控制系统发展的初期，通常是独立地实现上述几种机能，或采用较大型的计算机统一处理这几种机能的。随着生产的发展(大型化、复杂

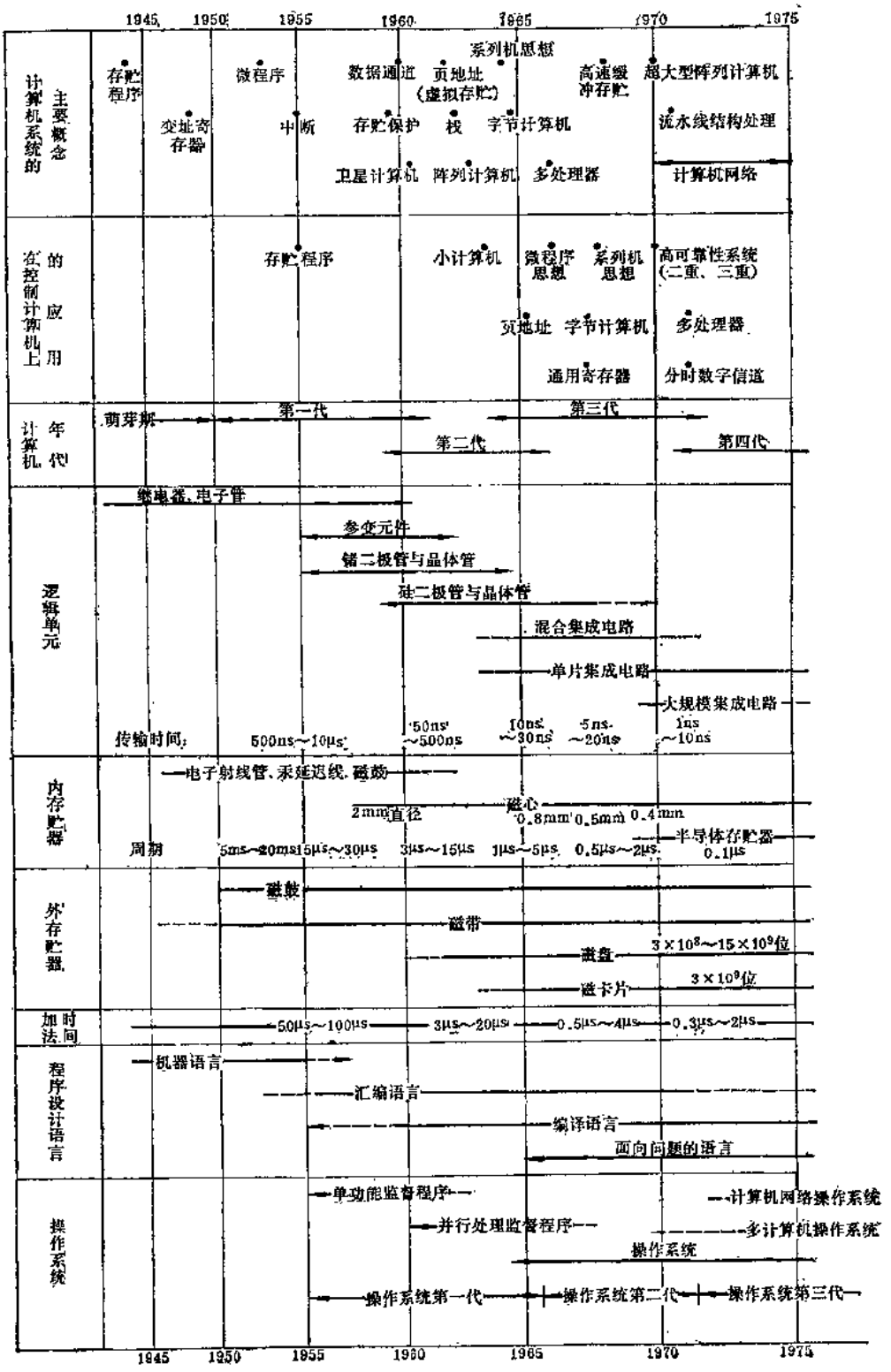


图 50·1-1　计算机系统发展过程

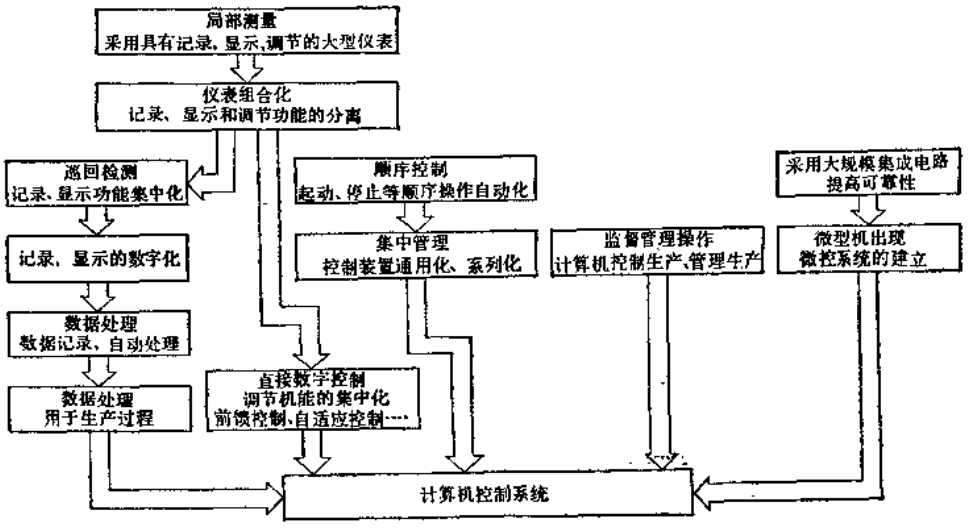


图 50·1-2 自动控制系统的发展

化、集中化)如果用单一的计算机企图实现全部功能,则在机器的可靠性、计算机的应用范围和设施的周期等等方面均带来不少困难。因此目前往往按控制系统的机能在系统内分级,而在每一级各自采用最合适的计算机,组成分级控制系统,充分发挥各类型计算机的特点和效率,提高系统的可靠性,这是一种新的动向。

计算机分级控制的示意图如图 50·1-3。

计算机控制系统是比较复杂的控制系统,它不但可以完成其他控制系统所具有的控制功能。而且有它的独到之处:

a. 从速度和精度来看,常规控制工具达不到的控制质量,计算机控制系统比较容易达到;

b. 由于计算机具有分时操作的功能,所以一台计算机能替代许多台常规控制工具;

c. 计算机的记忆和判断功能使计算机能够综合生产的各方面情况,在环境或生产参数变化时及时作出判断,选择最合理、最有利的方案和对策,这是常规控制工具所不能胜任的;

d. 有些生产过程,例如具有大滞后的对象,各参数相互作用比较大的对象,被控制量是经过计算才能得出的间接指标的对象,采用常规控制工具往往是得不到满意的效果的,这时计算机就更能发挥它独特的优点了。

总之,计算机控制系统的特点是容易实现任意的控制算法,只要按人们的要求改变程序或修正控制算式(数学模型)的某些参数,就能得到不同的控制效果。随着生产技术的进一步发展,它的优越性

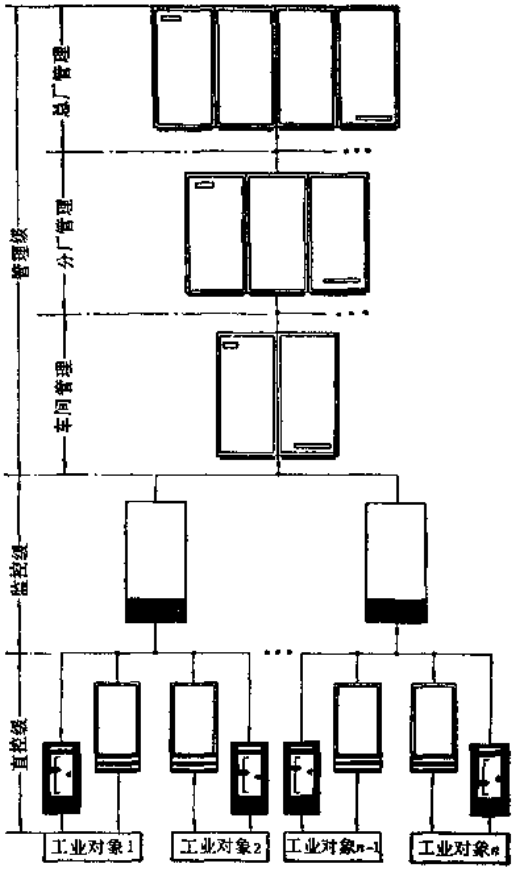


图 50·1-3 计算机分级控制的概念

愈来愈显示出来，但应用计算机控制需要设备比较多，投资较大，设备的可靠性对系统的可靠性影响较大；此外，设备的使用、维护、修理和编程序也比较复杂，需要一定熟练程度的使用维修人员，而且大多数计算机还需要特殊的环境条件。因此在实现生产自动化决定究竟使用计算机还是常规控制工具时，要从实际出发，选择适宜的方案，特别要把二者有机地结合起来。

3　计算机控制系统的组成

根据工业生产控制的特点和要求而设计的成套的计算机有时称之为工业控制计算机（简称控制机），由控制机组成的控制系统如图 50·1-4 所示。

计算机部分是它的主体，通常也叫作主机。控制机的主机和一般通用数字计算机的主机在原理和构造上是一样的，多功能的小型通用机就可直接作控制机的主机用。

主机包括运算器、控制器和内存贮器，如图 50·1-5 所示。

a. 运算器　由具有逻辑功能的电子线路和部件组成，主要用来对以数码形式表示的数据或信息进行加、减、乘、除等算术运算和逻辑运算。绝大多数的计算机都使用二进制数码，这样可以很方便地用逻辑电路进行运算（参看本手册第 5 篇自动控制理论），这是计算机的基本特点之一。

b. 内存贮器　存贮器用来存贮各种信息，具有

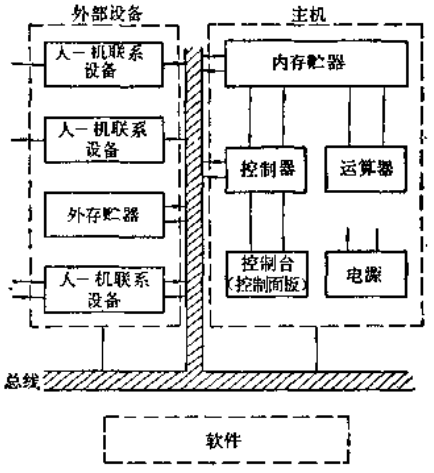


图 50·1-5　计算机（主机）的组成

“记忆”的功能，其中放在主机内部的称为内存贮器（简称内存）或主存贮器（简称主存）。

c. 控制器　是指挥和控制的机构，它联系计算机的各个部分，向各部分发出协调工作的命令。运算器与控制器在一起叫做中央处理器（CPU）。

计算机所进行的运算或者控制动作，是按人们事先规定的顺序和步骤来做的，这种步骤叫做“程序”，是预先存贮在计算机存贮器内的由一条条基本单元——“指令”组成。计算机工作时，依次取出一条条指令执行，并按执行结果决定接着执行哪一条指令。因此，按预存在机内的程序自动工作是计算机的另一个基本特点。

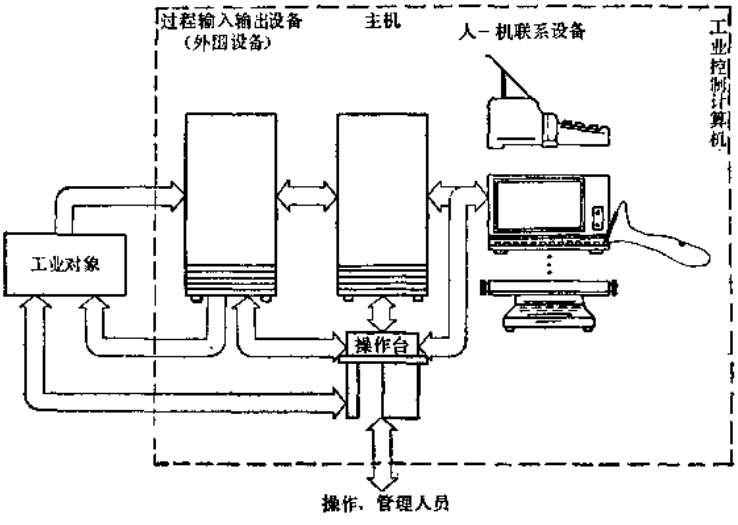


图 50·1-4　工业控制计算机系统的组成

人-机联系设备包括纸带输入机、电传打字机(控制台打字机)、宽行打字机、屏幕显示器、操作控制台、数传机(数据通信用)等,是操作人员和计算机进行联系的工具。

过程输入输出设备又称外围设备,是由许多与工业对象相互作用的装置组成。它一方面把工业对象的生产过程参数取出,经过转换,变成计算机能够接收和识别的代码,以便计算机进行处理;另一方面,又把计算机作出的控制决定,变成操作执行器的控制信号。

人-机联系设备和过程输入输出设备连同装在主机之外的大容量存贮器——外存贮器,统称为计算机的外部设备。

过程输入输出设备还必须通过工业自动化仪表才能和控制对象发生联系,进行这种联系的有检测仪表、显示仪表、调节仪表、执行器等。

主机、外部设备、工业自动化仪表,构成了控制计算机系统的硬设备,它是实现计算机控制的物质基础。

由于数字计算机的特点之一是按程序自动工作,所以控制计算机还必须具备完善的程序系统。程序系统又叫作软件(以区别于由电子、机械等设备构成的硬件)。

软件可分为系统软件和应用软件两大类。系统软件通常包括程序设计系统、诊断程序、操作系统以及与计算机密切相关的程序;应用软件则视计算机的应用场合而异。控制计算机的应用软件包括描述生产过程和控制规律以及实现控制动作的那些程序,它广泛涉及到生产工艺、生产设备、控制原理、控制工具等各个方面。

系统软件由计算机制造厂提供,一般总带有一定的通用性,应用软件则由配套厂或使用单位自行配置。

控制机的有关内容见图 50·1-6 和表 50·1-1。

因此,控制机系统事实上是由系统工程人员以计算机为主体、根据控制对象提出的要求及计算机所提供的能力而配置成的一个计算机控制系统。

计算机控制系统不仅包括由成套控制计算机与控制对象组成的整体,而且还包括反映生产过程和控制规律、体现控制功能和控制动作的程序系统。因此在建立一个计算机控制系统时,除了要配齐硬件之外,还要着重研究生产装置和工艺流程,建立数学描述关系(数学模型),确定控制规律和控制作用,把它们编成相应的程序,并配齐必需的软件。因此可以说,一个计算机控制系统的实现最终要把注意力集中在应用软件方面。

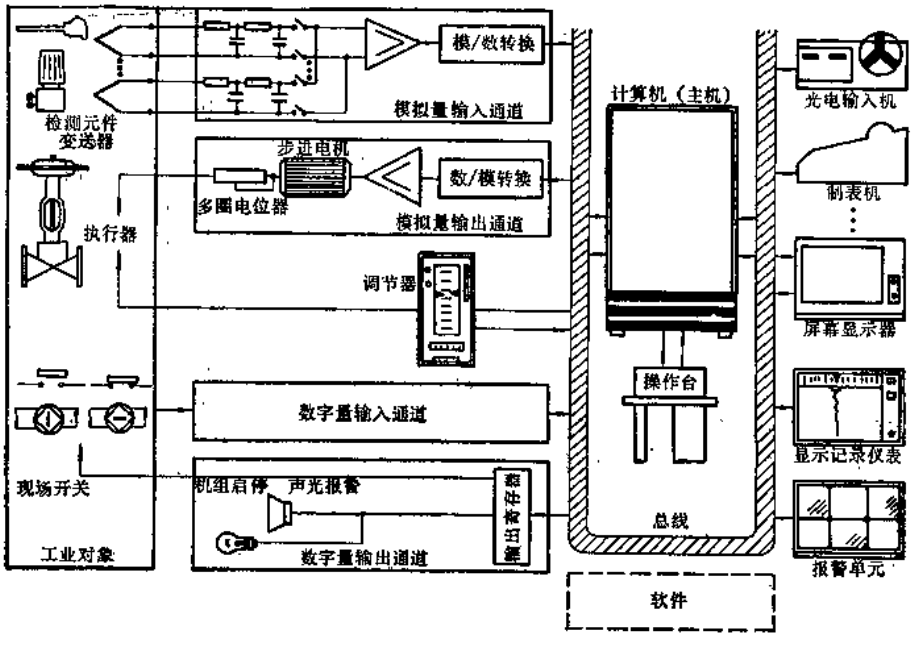
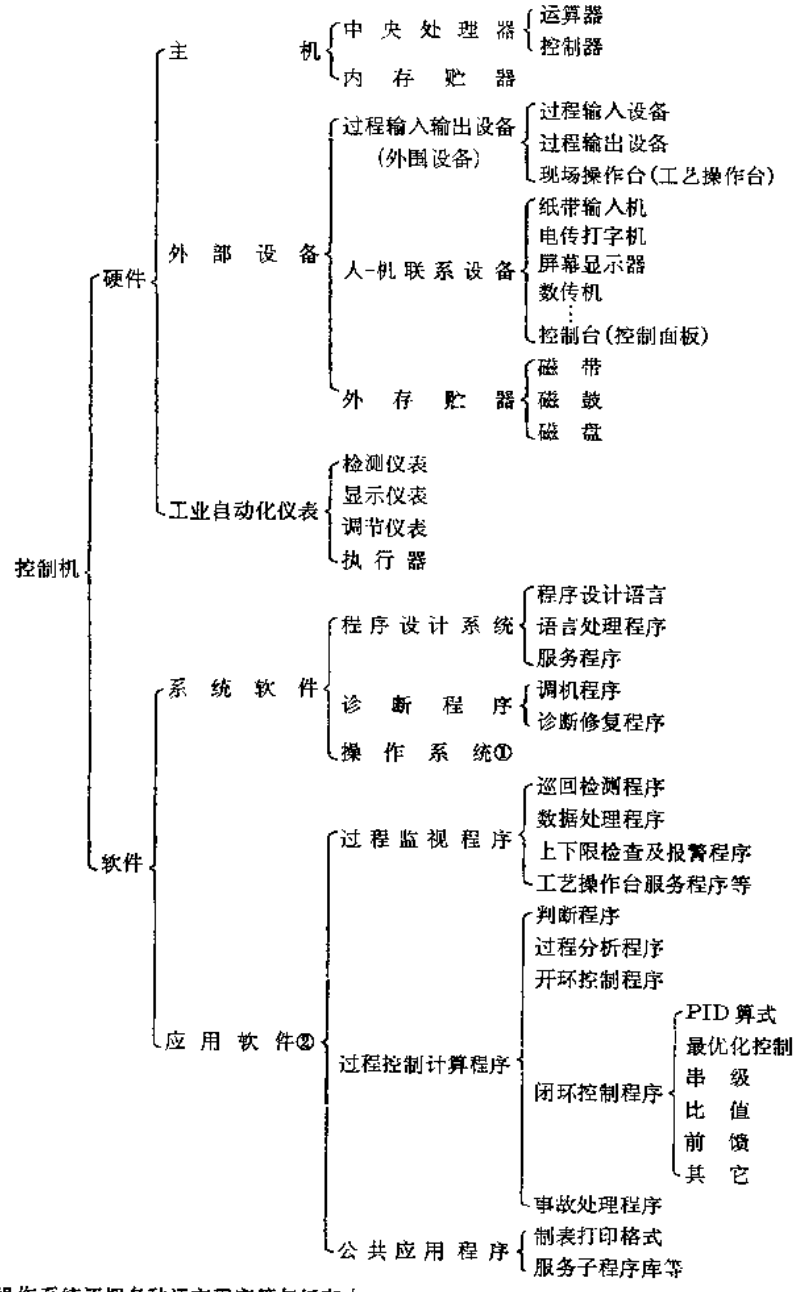


图 50·1-6 控制机的有关内容

表 50·1-1　控制机的有关内容



① 广义的操作系统还把各种语言程序等包括在内。

② 这里的应用软件只列举了与生产过程控制有关的。

第2章　计算机控制系统的硬件

硬件是计算机控制系统的物质基础，针对不同的工业对象采用不同的硬件可以组成功能和规模不同的控制系统。

1　控制用计算机的特点

应用于工业自动控制的计算机和有关设备一般称为工业控制机。它的主机就是通用数字计算机（常为小型机或微型机），但在系统中却专作自动控制使用。因此在结构性能上与通用机有共性，而在使用上有它的特殊性。

控制机与一般的计算机比较有如下几个特点：

a．可靠性高　因为生产过程往往是昼夜连续的，一般的生产装置几个月甚至一年才大修一次，这就要求控制生产的计算机可靠性尽可能的高。

b．相对而言对计算机的精度、速度要求较低　当然，随着工业自动化的发展，对控制机的规模、速度、精度的要求也在不断提高。

c．实时响应性好　为及时对付被控对象随时发生的事态，并保持诸多数据的同时性，要求计算机在某一限定时间内必须完成规定处理的工作。

d．要求有比较完善的中断系统　控制系统或计算机本身常常发生一些情况（例如要输入输出控制信号，要处理异常现象或故障等）需要计算机暂时停止原来工作去处理，为了适应这一要求设置了中断系统，它能使计算机暂时中断原来工作去处理临时发生的事件。对控制计算机来说要比其它类型计算机有更完善的中断系统。

e．有较丰富的指令系统　为适应生产过程的自动控制的需要，计算机必须具备有较丰富的指令系统，尤其是逻辑判断指令和外围设备控制指令。

f．要求有较完善的外部设备　特别是过程输入输出设备和各种工业自动化仪表要配套。

g．有正确反映生产过程规律的数学模型（或控制算法）　为寻找生产过程的最优工况，实现最优控制，达到多快好省的目的，要求为计算机建立正确反映生产过程规律的数学模型。

h．要有比较完善的软件　包括具有比较完整的操作系统，配备有工业自动化所需要的应用软件，使机器的使用更加合理，控制质量更高。

2　主机的总体结构和指令系统

计算机的性能很大程度取决于它的总体结构和指令系统。

计算机的总体结构是指计算机设计的基本思想和由此而产生的逻辑上的构造。总体结构与硬件和软件都有密切的联系。即使使用同样元器件和外部设备，总体结构不同，机器的性能会有很大的差别。而指令系统是与总体结构相呼应的。两者有很密切的关系。

2·1　总体结构

计算机总体结构涉及到计算机的几乎所有基本的内容，如指令的形式、寻址方式、数据形式、输入输出通道、程序中断、内存管理方式等等。它们组成了结构的骨架，也是构成系统的基础。

由于控制机的主机多半是小型机或微型机，这里先介绍小型机的总体结构。

计算机的主机由中央处理器和内存贮器、输入输出接口等部分组成。每一部分都有暂存数据或指令的寄存器，寄存器由触发器组成。数据或指令在寄存器间传送。早期的小型机各寄存器间都有控制门来控制传送过程，以中央处理器为中心，现在的小型机则以内存贮器为中心，多采用总线。总线是传送信息的公共通路，各寄存器通过控制门接在统一的一组总线上，分时进行传送。

图 50·2-1 是一台小型机主要部分的总体结构图。

图中 S、R、T 是三组总线，通过控制门（图中未画出）把各部件连接起来，使得计算机的操作过程就变成进出总线、内存贮器及算术逻辑部件的过程。

ALU 是算术逻辑部件，可进行算术、逻辑运算；

A 是累加器，它不仅能将本身内容和另一数进行代数相加，而且能存放相加结果（由于能进行累次相加而叫累加器），还可以执行寄存、移位和求补等

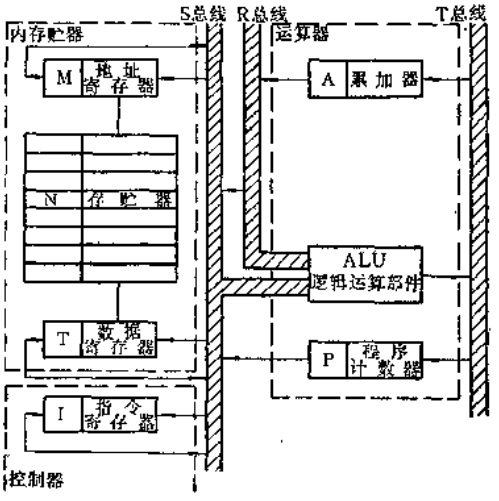


图 50·2-1　一台小型机的总体结构(主要部分)

操作。

N、M、T 是内存贮器，由地址寄存器 M 决定存贮器 N 中某一地址，将该地址所存信息(指令或数据)取到数据寄存器 T，或将信息由 T 存入该地址。

指令寄存器 I 以及相应的一些控制逻辑电路和控制门构成了控制器，控制协调各部件工作。

例如当我们要把 A 中的数 X 与内存贮器某地址 D 中的数 Y 相加时，X 通过 R 总线，Y 从 N 中取到数据寄存器 T 经过 S 总线，两者进入 ALU 相加，相加结果通过 T 总线送到累加器 A 中暂存。

在有些小型机(如上例)中，在中央处理器中用于运算的寄存器(或累加器)只有一个；而有些小型机为了减少从内存取数存数的次数，提高工作速度，设置了多个寄存器以存放中间结果和进行累加运算等。后来发展到这些寄存器作为通用寄存器来使用，同一寄存器按程序的调度可以作不同的用途(例如存放中间结果、变址等)。这样就提高了使用效率和灵活性。

计算机的进一步发展，还要求有容量更大的寄存器组，能作为通用寄存器、输入输出工作寄存器等使用，为此在有些计算机的中央处理器中设置了便笺存贮器，它是用半导体存贮器构成的，比磁心存贮器工作速度高得多。此外还要求一些寄存器或存贮单元的信息出入管理比较方便，堆栈(堆阵)就是为了满足这一要求而发展起来的。

堆栈是一种快速的寄存器或存贮器(有时占用内存贮器的一部分作堆栈)。它是用来暂时存放中间结果或子程序与中断服务程序的联结，或存贮断点，状态字等。

各个数据可按时序存入堆栈中，而以相反的次序从栈中取回或消去。这种操作称为"后进先出"。

例如计算机在处理程序的过程中，常要转入子程序(见第 4 章)，子程序执行完后又返回原程序，这就要把原程序的返回地址存放好。但子程序在执行过程中又可能转入另一子程序并返回，亦即可能发生多重嵌套，这时，采用堆栈放置多重返回地址，让后放入的返回地址先取用，这对程序的编制是十分有利的。

进出栈的操作如图 50·2-2 所示。

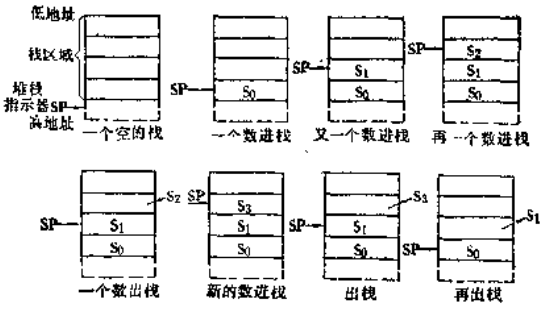


图 50·2-2　进出栈操作图

除了中央处理器中的总线外，小型机在输入输出通道中也采用总线，故称为多总线结构。近年来又出现了单总线结构。这种结构的特点是使中央处理器、内存贮器、外部设备等均并挂在总线上，使各部件都处于平行关系。对外存贮器的控制可以象对内存贮器控制一样灵活，便于外部设备的扩充。摆脱过去各部件的信息都要经过中央处理器，大部分只能进行串行操作的情形。从而使中央处理器、外部设备能同时操作，并可采用分时系统和远距离使用计算机(结合数字通信线路和数据终端设备)。如果采用异步工作还可使中央处理器能适应各种速度的设备，这样就可配置存贮周期不同的各种类型的存贮器，而无须对中央处理器进行控制方式的改变，便于更换部件及采用先进技术。

图 50·2-3 是一台具有单总线结构、使用堆栈的小型机的总体结构图。

这小型机的中央处理器由下面几部分组成：

a. 累加器 A_0～A_7　A_0～A_7 为 8 个累加器，其中 A_0～A_5 为通用寄存器，寄存操作数及结果，也可以作为变址指示器及寻址修饰等。A_6 为堆栈指示

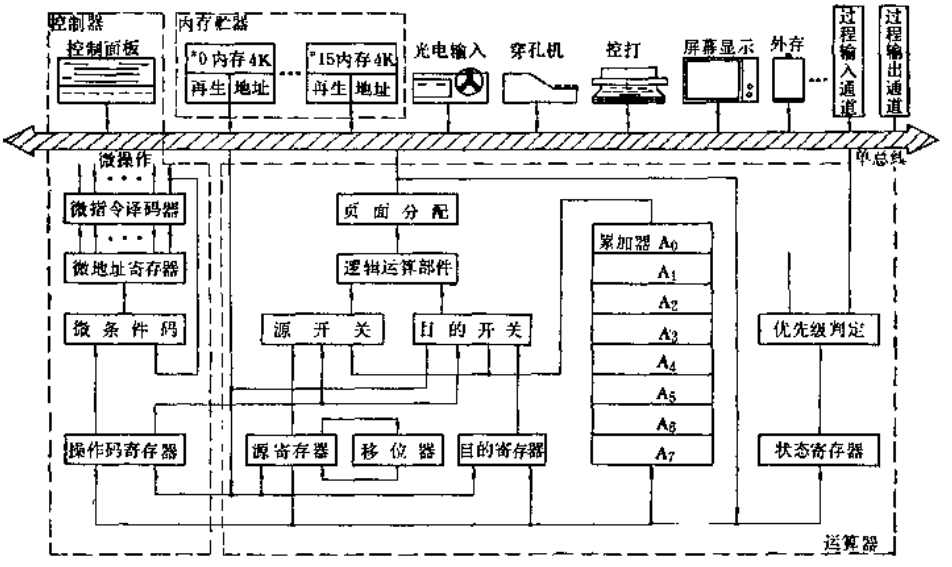


图 50·2-3 具有单总线结构、使用堆栈的小型机总体结构图

器，A_7 为指令计数器。

b. 源寄存器及目的寄存器 源寄存器寄存源操作数及中间结果暂存。目的寄存器寄存目的操作数及中间结果暂存。

c. 源开关及目的开关 源开关的作用是转换来自累加器的输出、源寄存器的输出以及操作码寄存器的输出。目的开关的作用是转换来自累加器的输出、目的寄存器的输出、总线的输入、以及操作码寄存器的输出。

d. 移位器 源寄存器的输出可进行左移、右移、交换、不移等操作。

e. 逻辑运算部件 即算术逻辑部件，具有必要的算术操作及逻辑操作。

f. 操作码寄存器 操作码寄存器寄存 16 位字长的指令代码，以执行相应的操作。

g. 微程序控制器 微程序控制器包括微地址寄存器、微指令译码器和微条件码。微指令译码器依指令译码出相应的微操作，而微条件码依指令条件及相应运算结果状态形成相应的返回程序地址。

h. 状态寄存器 寄存计算机的状态字（包括中断的优先级、运算结果状态、进位位、负数、结果及管态等等）。

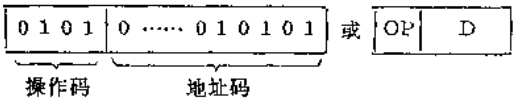
i. 优先级判定 优先级判定的逻辑作用是判断中断请求级别与中央处理器正在执行的程序的优先级何者优先。若中断请求级别比中央处理器正在执行的程序的优先级别高，则响应中断操作，否则执行中央处理器原来的程序。

j. 页面分配 目的是自动将目的程序的虚地址转换成实地址。

2·2 指令系统

规定计算机的基本操作种类、操作数值或其地址的命令称为指令。计算机是按程序自动工作的，而程序是由一系列（成串）指令组成的。对一台计算机或一个计算机系列来说，它所能够执行的各种不同类型指令的总和称为该机或该系列机的指令系统。

指令由机器代码（一般用 8 进制或 16 进制书写）、记忆符或程序设计语言表示（详见第 4 章），而记忆符又有英文与汉语拼音两种。每条指令至少应包括两部分：一是操作码，决定机器运行的性质；二是地址码，表示参加操作的代码的地址或操作结果存贮地址。最简单的单地址指令形式如下：



现在大多数小型机都采用单地址的指令。

例如我们要计算：

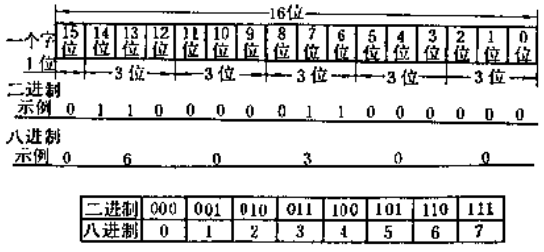
$$X+Y=Z \quad (X=1, Y=2)$$

其中已知 X、Y 存在内存贮器 300、301 地址，算得的结果要存入 302 地址，则需要通过如表 50.2-1 所列的程序进行。

表 50·2-1　求 X+Y=Z 的程序

顺序	操作	指令		备注
		代码形式	记忆符(汇编)形式	
1	把 X 取到 A 寄存器	06 0300	LDA X	06 是"取到 A"（它们都是用 8 进制书写的操作码）
2	把 Y 加到 A 寄存器，得(X+Y)	04 0301	ADA Y	04 是"加到 A"
3	将结果(X+Y=Z)存入 302	07 0302	STA Z	07 是"A 送内存"
4	停机	10 2000	HALT	10 是停机

8 进制和 2 进制的关系可用图 50·2-4 来表示。



二进制	000	001	010	011	100	101	110	111
八进制	0	1	2	3	4	5	6	7

图 50·2-4　用 8 进制数表示由 16 位构成的一个字

在图 50·2-1 的计算机中，ADA　Y 这条指令的执行过程就是将 Y 的地址 301 送到 M，由 N 中 301 地址中将 Y 取出到数据寄存器 T，再由 T 送入总线 S，与 A 中的 X(000001) 经过总线 R 一同进入 ALU，相加的结果经总线 T 再送入 A，同时 P 的内容还要加 1，如图 50·2-5 所示。

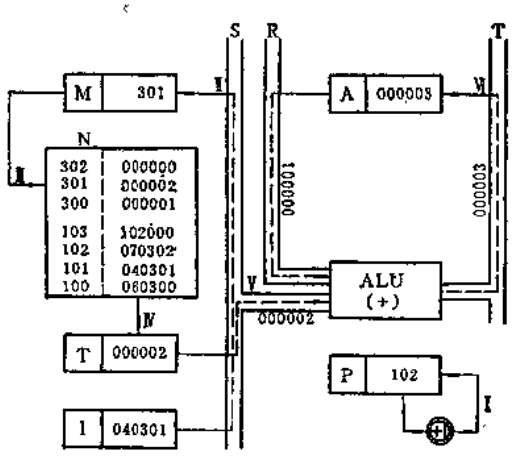


图 50·2-5　ADA　Y 的执行指令阶段
I～VI 表示指令的执行顺序

2·2·1　指令按操作的分类

指令有不同的分类方法。指令从功能上来看有算术运算指令、逻辑运算指令、控制指令、传送指令、输入输出指令和复合指令几种，根据是否与内存贮器打交道又可分为访问内存指令和非访问内存指令。下面仅就功能上的分类加以叙述。

a. 算术运算指令　加减乘除运算指令是计算机的最基本指令，有的小型机为简化设备而省去了乘除指令。根据所处理的数据种类，算术运算指令可分为：

(1) 定点运算和浮点运算；
(2) 二进制运算和十进制运算；
(3) 整数运算和小数运算；
(4) 固定字长运算和可变字长运算。

b. 逻辑运算指令　逻辑运算指令以二进制为单位进行逻辑加、逻辑乘、逻辑非等操作，如表 50·2-2 所示。

表 50·2-2　逻辑运算指令

逻辑运算	符号	相应二进位的关系	例
逻辑加	+ ∨	数据 A 数据 B 结果 $A_1+B_1\rightarrow C_1$ $A_2+B_2\rightarrow C_2$ $\vdots\ \vdots\ \vdots$ $A_n+B_n\rightarrow C_n$	数据 A　10110001 数据 B　11001011 $C=A+B=11111011$
逻辑乘	× ∧ ·	数据 A 数据 B 结果 $A_1\cdot B_1\rightarrow C_1$ $A_2\cdot B_2\rightarrow C_2$ $\vdots\ \vdots\ \vdots$ $A_n\cdot B_n\rightarrow C_n$	数据 A　10110001 数据 B　11001011 $C=A\cdot B=10000001$
逻辑非	−	$\overline{A}_i\rightarrow C_i$	数据 A　10110001 $C=\overline{A}=01001110$
按位加	⊕	$A_i\cdot\overline{B}_i+\overline{A}_i\cdot B_i\rightarrow C_i$	数据 A　10110001 数据 B　11001011 $C=A\cdot\overline{B}+\overline{A}\cdot B=01111010$

c. 控制指令　控制指令在执行程序时进行各种控制动作。最普通的控制指令有各种指示器的置

位/复位指令、改变程序执行顺序的转移指令、停止执行程序的停机指令等等。

d. 传送指令 传送指令常常用来把数据和程序从内存的某个地址传送到别的地址。

e. 输入输出指令 输入输出指令是使外部设备动作的指令。外部设备种类繁多，动作也复杂，而且许多是通过通道与计算机联系的，因此输入输出指令必须能控制种类非常多的动作。

f. 复合指令 复合指令是用一条指令执行多条前述普通指令组合起来的动作，以缩短处理的时间。

2·2·2 基本指令

用于生产过程控制的计算机指令系统应着重于工业控制，同时应考虑到多功能的应用，兼顾系列机各机型间的系列性、向上兼容性、可扩性以及使用维护的方便灵活和便于记忆等。

对于成系列的计算机的指令系统可分为基本指令和扩充指令两类。最低档机的指令系统只包括基本指令，扩充指令是在基本指令的基础上为了增加功能而扩展的，这样就实现了指令系统的向上兼容性，也保证了软件的向上兼容性。

基本指令的"基本"有两种涵义，一是其运算的起码功能，其它运算可以用这些指令来完成，这样硬件可较简单，但软件工作量大，运算速度低；二是为某些对象服务而采用的一些必不可少的功能(如控制机的输入输出指令就是必要的)，其它附带功能亦由这些指令来完成。可见对不同类型、不同用途的计算机其基本指令的选择是不尽相同的。

表 50·2-3 是某系列机所用的基本指令名称及符号。

2·2·3 寻址方式

计算机的每一条指令通常是在寄存器、内存、外部设备之间进行信息的传送或运算的，因此总要指定信息所在的地址。信息的来源可称为原地址，而信息的去处可称为目的地址。指令总得指定源地址与目的地址。有的指令在操作码中已指出了其中一个地址，另一个地址则需通过地址码指定，指定地址的方式称为寻址方式。而按照指令给出的形式地址及特征码、找出存放在内存中需要进行处理的信息的真实地址的过程叫寻址操作。

表 50·2-3 某系列机用的基本指令及符号

指令名称	符号	备注
传送	MOV	MOVE
加法	ADD	ADDITION
减法	SUB	SUBTRACTION
加反	ADC	ADDITION COMPLEMENT
逻辑乘	AND	LOGICAL AND
逻辑或	LOR	LOGICAL OR
交换	SWP	SWAP
乘法	MUL	MULTIPLICATION
除法	DIV	DIVISION
比较转	BCJ	BYTE COMPARE JUMP
加进位	ACA	ADDITION CARRY
减进位反	SCC	SUBTRACTION CARRY COMPLEMENT
字节传送	BMV	BYTE MOVE
取数	LDA	LOAD
存数	STA	STORAGE
无条件转	JMP	JUMP
转子	JSR	JUMP TO SUBROUTINE
零转	JZR	JUMP IF RESULT IS ZERO
非零转	JNR	JUMP IF RESULT IS NO ZERO
进位零转	JZC	JUMP IF CARRY IS ZERO
进位非零转	JNC	JUMP IF CARRY IS NO ZERO
负转	JNN	JUMP IF NEGATIVE IS NO ZERO
正转	JZN	JUMP IF NEGATIVE IS ZERO
左环移	ROL	ROTATE LEFT
右环移	ROR	ROTATE RIGHT
左算移	ASL	ARITHMETICAL SHIFT LEFT
右算移	ASR	ARITHMETICAL SHIFT RIGHT
左长移	LSH	LEFT SHIFT
右长移	RSH	RIGHT SHIFT
送屏蔽	MSK	MASK OUT
送状态	SST	SEND STATE
送优先	SPR	SEND PRIORITY
停机	STP	STOP
进栈	IST	IN STACK
出栈	OST	OUT STACK
中断返回	IRR	INTERRUPT RETURN
清外部	CPE	CLEAR PERIPHERAL EQUIPMENT
开中断	INE	INTERRUPT ENTER
关中断	IND	INTERRUPT DISABLE
进管	EMT	ENTER MONITOR
出管	OMT	OUT MONITOR

为解决小型机字长较短、容量不大、而又要求功能强所带来的矛盾，为扩大计算机的操作功能和使用上的灵活性，恰当地选择寻址方式是十分必要的。

用指令的地址码指定内存地址时，其寻址方式有直接寻址和间接寻址两种。

a. 直接寻址方式　若指令中地址码给出的地址直接指定所求信息的存贮位置，称之为直接寻址，见图50·2-6。

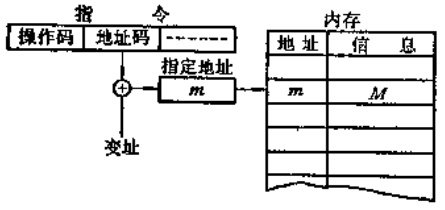


图 50·2-6　直接寻址方式

b. 间接寻址方式　间接寻址是间接地通过内存的某个地址指定所求信息的地址，如图50·2-7所示，所以如果变更间接地址中所存地址的信息，则不必改变原指令的地址码就能作各种地址改变，便于两个程序之间的数据传送，可以简化表格处理程序的编写工作。

间接寻址的次数就叫间接寻址的级数。

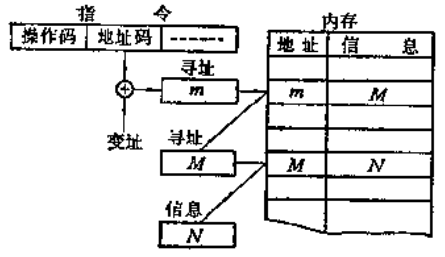


图 50·2-7　间接寻址方式(一级)

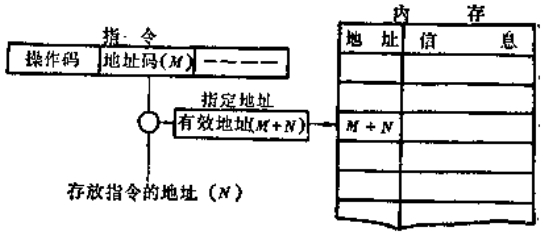
2·2·4　变　址

变址就是改变地址的意思。计算机在执行需要变址的指令时，必须把指令中给出的地址(一般称为形式地址)与变址寄存器中的内容相加(或相减)，才得到操作数的地址。它分下列几种形式：

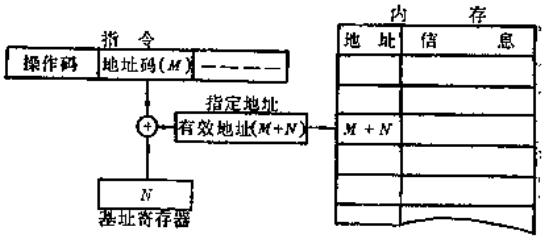
a. 绝对寻址　它是指令地址码中所指定的地址直接地不作任何变更地指定内存地址。

在绝对寻址方式中，内存的全部地址均可由指令任意处理，对程序来说有很大的灵活性。但如果内存容量很大，指令地址码的位数将增加，会使指令本身太长，导致指令的读出时间增加。

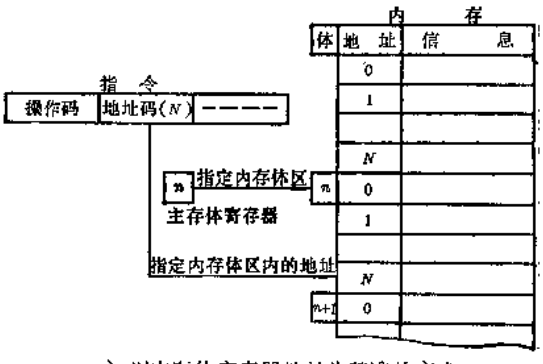
b. 相对寻址　指令中地址码的值仅用来指定从某个基准地址算起的位置，它有三种方式，如图50·2-8所示。



a) 以存放指令的地址为基准的方式



b) 以基址寄存器为基准的方式



c) 以内存体寄存器地址为基准的方式

图 50·2-8　相对寻址方式

(1) 以存放指令的地址为基准的方式　指令中的地址码和存放该指令的内存地址码相加，加得的结果用作实际上的地址。

(2) 以基址寄存器为基准的方式　就是用另外的一个独立寄存器给出基准地址，该寄存器的内容可由程序任意改变。

(3) 以内存体寄存器地址为基准的方式　内存体寄存器中的地址用来指定内存的区域(指出是第几个内存体)，然后用指令中的地址码在指定的内存体内寻址。

c. 变址寄存器变址　大多数程序常常会碰到这样的情况，即只要有规则地变更存数的地址码，多

次执行同一条指令，就可以得到运算结果。例如连续做十次加法时，只要把地址码变更十次也就是把进行加法的数据改变十次，用同一指令反复执行十次后就得到了运算结果。

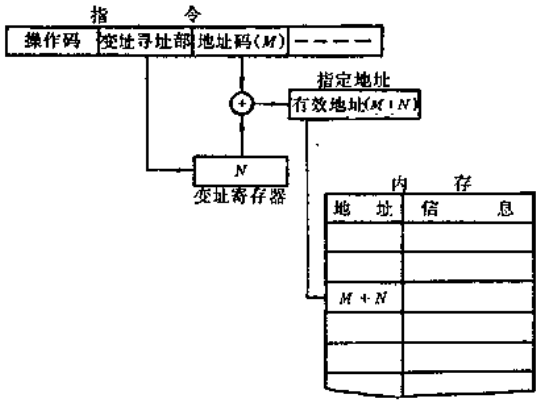


图 50·2-9 有变址寄存器的变址

这种简单改变地址码的方法称之为有变址寄存器的变址方法，如图 50·2-9 所示。

变址寄存器常常用作计算连续运算次数的计数器，或暂时存放其它的运算结果，或作为下一条指令的地址，或用于主程序和子程序的联结，总之用途是很广的。

d. 基址变址 基址变址原理上与变址寄存器变址相同，但它通常由操作系统来处理。

2·2·5 寻址示例

表 50·2-4 是某计算机系列所用的寻址方式。

其中几种寻址方式的示例说明列于表 50·2-5 和 50·2-6。

表 50·2-4 某计算机系列的寻址方式

寻址方式	名 称	汇编程序格式	功 能
直接寻址方式	寄存器方式	R	寄存器内容为操作数
	自增方式	$(R)+$	寄存器内容为操作数的地址，调用后在寄存器内容上增数
	自减方式	$-(R)$	调用前将寄存器内容减数，寄存器内容减数后作为操作数的地址
	变址方式	$x(R)$	把数值 x（存于指令后的一个字内）加到 (R)，产生操作数的地址，x 和 (R) 都不变
间接寻址方式	寄存器间接	$@R$ 或 (R)	寄存器内容是操作数的地址
	自增间接	$@(R)+$	寄存器用作字 A 的指示器，字 A 的内容为操作数的地址，然后寄存器内容增数（总是增加 2，即使字节指令也加 2）
	自减间接	$@-(R)$	寄存器减数（总是减 2，即使字节指令也减 2）然后用作操作数地址的地址
	变址间接	$@x(R)$	将数值 x（存于指令后的一个字内）加到 (R)，加得的和作为操作数地址的地址，x 和 (R) 都不变
PC 寻址方式①	立即方式	$\#n$	指令后面跟的是操作数
	绝对方式	$@\#$	指令后面跟的是绝对地址
	相对方式	A	指令后面跟的是相对于指令的 A 地址
	相对间接方式	$@A$	指令后面跟的是相对于指令的 A 地址的地址

① 当 $R=R_7$ 时，即成为 PC 寻址方式。
PC 是程序计数器，也是通用寄存器，所以也能响应一切寻址方式。

表 50·2 5　几种直接间接寻址方式的作用及示例

寻址方式	名　　称	作　　用	示　　例
R	寄存器方式	减少访内存次数，就是把寄存器R的内容作为操作数	MOV 2 3 操作前：R_2 000120；R_3 000000 操作后：R_2 000120；R_3 000120
@R	寄存器间接方式	减少访内存次数，就是把寄存器R的内容作为操作数的地址	MOV @2 3 操作前：R_2 000120；R_3 000000；120 177777 操作后：R_2 000120；R_3 177777；120 177777
$(R)+$	自增方式	对存取在内存中连续存放的数据十分有用 一方面把寄存器R的内容作为操作数的地址；一方面在操作后R的内容自增1	MOV (2)+ 3 操作前：R_2 000120；R_3 000000；120 177777 操作：R_2 000120；R_3 177777；120 177777；后+1 操作后：R_2 000121；R_3 177777；120 177777
$-(R)$	自减方式	对存取在内存中连续存放的数据十分有用 首先将寄存器R的内容自动减1，然后把R的内容作为操作数的地址	MOV −(2) 3 操作前：R_2 000120；R_3 000000；117 000134 先−1 操作：R_2 000117；R_3 000134；117 000134 操作后：R_2 000117；R_3 000134；117 000134
$x(R)$	变址方式	可连续两次存取数据 寄存器R的内容加上紧跟在指令后面的变址字作为操作数的地址	400 ADD 100(3) 5 401 000100 操作前：R_3 003700；R_5 076540；4000 001240 操作：003700 + 000100 = 004000；076540 + 001240 = 100000 操作后：R_3 003700；R_5 100000；4000 001240；PC=402
@$x(R)$	变址间接方式	可进行两次间接存取数据 把指令后面的字中的x值与R相加，加得的结果作为操作数地址的指示器，x和R都不变	ADD @1000R_2 R_1 操作前：1050 000002 R_1 001234；1100 001050 R_2 000100 操作后：1050 000002 R_1 001236；1100 001050 R_2 000100 1000 + 100 = 1100；001234 + 000002 = 001236

表 50·2-6 指令计数器的四种寻址方式示例

当 $R=R_7$	名 称	作 用	示 例
$(R_7)+$	%n 立即方式	自增方式 $E=R_7$ 时就为立即方式 对于取数十分方便，即把紧接指令的下一条作为操作数	406 ADD % 0 407 000010 操作前 操作后 R_0 001010 R_0 001020 + 407 000010 PC=410
$@R_7$	@% 绝对方式	为在整个存贮空间内存取数据提供便利条件 规定紧接指令的下一条单元内容作为操作数地址	410 MOV @% 0 411 037200 操作前 操作后 R_0 012400 R_0 010101 411 037200 PC=412 37200 010101
$x(R_7)$	A 相对方式 ·±x	变址方式用在 PC 时即为相对方式 即把紧接指令的下一条单元内容加上 PC 的内容作为操作数地址	R_0 104000 1200 004000 176 ADD ·+1000 0 177 1000 200 操作后 R_0 110000 PC=200
$@x(R_7)$	@A 相对间接方式 @·±x	方法同相对方式，不同的是计算出来的地址为操作数地址的间接地址	1000 034402 500 1000 R_0 020202 176 MOV 0 @·+300 177 000300 200 操作后 1000 020202 PC=200

3 中央处理器

计算机系统中能够独立执行程序、对指令和数据进行加工与处理的部分叫中央处理器，也就是通常所说的运算器和控制器部分。它按一定顺序从内存贮器中取出构成程序的一条条指令，取一条执行一条，并根据执行情况决定下一步怎样进行。在简单的计算机中，中央处理器还负责内存与外部设备传送数据，而在比较复杂的计算机中，它只对这些操作给予指示。

为了完成各项任务，中央处理器具有进行算术与逻辑运算的运算器(算术逻辑部件)和许多存放信息的寄存器(早期的计算机各寄存器有各自专一的用途，目前则趋向于设置多用途的通用寄存器，便于程序设计人员编制出更有效的程序)，还有控制和协调各部件工作的控制器，通过总线、传送门与内存贮