



计算机导论和程序设计

语言

★ ★
★ ★
★ ★
★ ★
★ ★
★ ★
★ ★
苑森淼

★ 刘长欢 李振文 主编
吉林科学技术出版社

计算机导论和程序设计语言

苑森森 刘长欢 李振文 主编

吉林科学技术出版社

【吉】新登字 03 号

计算机导论和程序设计语言

苑森森 刘长欢 李振文 主编

责任编辑：王维义

封面设计：史殿生

出版 吉林科学技术出版社
发行

787×1092 毫米 16 开本

26.75 印张

670,000 字

印刷 吉林工学院印刷厂

1992 年 8 月第 1 版 1992 年 8 月第 1 次印刷

印数：1—3500 册 定价：13.70 元

ISBN 7-5384-1031-7/TP·11

内容简介

本书由计算机导论、BASIC 语言、FORTRAN 语言、PASCAL 语言组成。

适用于计算机专业的“计算机导论”课和相应“算法语言”课以及非计算机专业的“计算机基础”课的教材。全书各部分内容完整，例题、习题安排适当。

本书的主要特点是从教学实际出发，深入浅出地讲述计算机的基础知识和编程技术技巧，不仅可以作为大专院校相应课的教材，也可作为科技人员学习编程技术的参考书。

序 言

本教材分为四篇：计算机及计算导论、BASIC 语言程序设计、FORTRAN 语言程序设计和 PASCAL 语言程序设计。适用于非计算机类各专业的“计算机基础”或计算机专业的“计算机导论”和相应的算法语言课程教学使用。第一篇与第二至四篇之一均可构成上述课程的讲授体系，适用于 60~70 学时的教学大纲。

第一篇，计算机及计算导论以计算机使用为目的，以计算机硬件和软件的基本概念及它们相互支持构成基本的计算机系统为线索，讲述计算机的基本工作原理和软、硬件支持环境的构成，以及用程序求解问题的基本方法。第二至四篇分别讲述 BASIC、FORTRAN 和 PASCAL 语言的基本语句和程序设计方法。三种语言在程序设计方法的讨论中，都注意了结构程序设计概念和技巧的论述。在应用举例和内容安排方面，三种语言尽量相互呼应，使学生学习一种语言后，可以容易地举一反三，学习其他两种语言。这样安排也希望学生通过用不同语言为解决同一或类似问题所写程序的比较，掌握各语言语句使用和编程技巧上的特点。提高学生对语言构成的认识和针对不同问题选择语言进行编程的能力。

各章后均有相当数量的习题，第二至四篇配有一定数量的带 [上机作业] 字样的习题，供教师在指导学生上机实验中选用。也可将其中一部分留给学生做为上机作业，让其在计算机上自行完成。这类习题是相应章节所讲内容的综合运用和编程技巧的进一步深化，是教材的有机组成部分，望教师和同学注意使用。

本教材是在吉林工业大学、东北师范大学、长春邮电学院长期教学实践基础上共同编写的。由苑森森、刘长欢、李振文共同主编。第一、四两篇由吉林工业大学组织编写，其中：第一篇由苑森森（第 1、4 章）、付宏（第 2、3 章）编写；第四篇由李雄飞（第 1、2 章）、刘震（第 3、6 章）、陈思国（第 4、5 章）编写。第二篇由长春邮电学院组织编写，其中：王晓飞编写第 1~5 章，李振文编写第 6~9 章。第三篇由东北师范大学组织编写，其中：王辉编写 1~6 章，刘长欢编写第 7 章。俞勇同志也参加了本书部分章节的编写。

本书由吉林工业大学吴治衡教授主审，提出指导性意见，作者深表感谢。本书缺点、错误在所难免，望使用本教材的老师和同学们予以指正，以图改进。

编 者 1992. 3.

目 录

第一篇 计算机及计算导论	(1)	习题	(76)
第一章 计算机及系统构成	(2)	第二章 简单程序设计	(76)
§ 1. 1 社会对计算机的需求	(2)	§ 2. 1 赋值语句	(76)
§ 1. 2 计算机的基本结构及工作	(5)	§ 2. 2 键盘输入语句	(78)
§ 1. 3 存储器	(9)	§ 2. 3 读数据语句和置数据语句	(79)
§ 1. 4 外存储器的文件组织与访问	(13)	§ 2. 4 恢复数据区语句	(80)
§ 1. 5 计算机通用 I/O 设备	(15)	§ 2. 5 打印(输出)语句	(81)
§ 1. 6 计算机系统构成	(19)	§ 2. 6 注释语句	(86)
小结	(22)	§ 2. 7 简单程序设计	(87)
习题	(23)	小结	(88)
第二章 数据	(23)	习题	(88)
§ 2. 1 计算机算术基础	(23)	第三章 分支程序设计	(89)
§ 2. 2 数的定点、浮点表示及运算	(27)	§ 3. 1 分支概念	(89)
§ 2. 3 数的原码、反码、补码表示及运算	(28)	§ 3. 2 流程图	(90)
§ 2. 4 字符及字符串表示方法	(32)	§ 3. 3 简单条件语句	(91)
小结	(34)	§ 3. 4 多分支条件语句	(95)
习题	(34)	§ 3. 5 无条件转移语句	(98)
第三章 计算过程的表示方法	(35)	§ 3. 6 暂停语句	(99)
§ 3. 1 程序设计语言及进化	(35)	小结	(100)
§ 3. 2 程序中的变量和数据类型说明	(37)	习题	(100)
§ 3. 3 基本运算和表达式	(42)	第四章 循环程序设计	(101)
§ 3. 4 语句级控制结构	(48)	§ 4. 1 循环概念	(101)
小结	(51)	§ 4. 2 计数循环语句	(102)
习题	(52)	§ 4. 3 条件循环语句	(107)
第四章 算法和数据结构	(52)	§ 4. 4 多重(嵌套)循环概念	(108)
§ 4. 1 算法	(52)	小结	(111)
§ 4. 2 数据结构和基底语言提供的数据 结构工具	(54)	习题	(112)
§ 4. 3 基本数据结构	(59)	第五章 自定义函数和子程序	(113)
§ 4. 4 过程和函数	(63)	§ 5. 1 自定义函数及其调用	(113)
小结	(66)	§ 5. 2 子程序的定义及其调用	(115)
习题	(67)	§ 5. 3 应用举例	(118)
第二篇 BASIC 语言与结构程序设计	(68)	小结	(120)
第一章 BASIC 语言的基础知识	(69)	习题	(120)
§ 1. 1 BASIC 语言简介	(69)	第六章 结构化程序设计方法	(121)
§ 1. 2 基本符号	(69)	§ 6. 1 概述	(121)
§ 1. 3 常量和变量	(70)	§ 6. 2 结构化程序的三种基本结构	(122)
§ 1. 4 标准函数	(71)	§ 6. 3 结构化程序的特征	(124)
§ 1. 5 表达式	(72)	§ 6. 4 结构化程序设计方法	(125)
§ 1. 6 BASIC 语言源程序基本结构	(74)	小结	(129)
小结	(75)	习题	(129)
		第七章 数组和下标变量	(130)

§ 7. 1 数组和下标变量的概念	(130)	第四章 数组	(226)
§ 7. 2 数组的维数说明语句	(131)	§ 4. 1 数组与数组说明语句	(227)
§ 7. 3 数组的基本操作	(132)	§ 4. 2 数组的输入/输出	(231)
§ 7. 4 下标变量的应用	(133)	§ 4. 3 字符型数组与逻辑型数组	(236)
小结	(138)	§ 4. 4 数组应用举例	(240)
习题	(139)	小结	(244)
第八章 字符串与字符型变量	(140)	习题	(245)
§ 8. 1 字符串常量和字符串变量	(141)	第五章 语句函数及子程序	(245)
§ 8. 2 字符串的输入/输出	(141)	§ 5. 1 语句函数	(246)
§ 8. 3 字符串运算	(143)	§ 5. 2 函数子程序	(247)
§ 8. 4 字符串函数	(146)	§ 5. 3 子例程子程序	(252)
§ 8. 5 字符串数组	(151)	§ 5. 4 ENTRY 语句和 SAVE 语句	(254)
§ 8. 6 字符串的应用例题	(153)	§ 5. 5 可变返回	(256)
小结	(153)	§ 5. 6 外部语句和内部语句	(257)
习题	(154)	§ 5. 7 等价语句	(259)
第九章 文件	(155)	§ 5. 8 公用语句	(261)
§ 9. 1 文件的基本概念	(155)	§ 5. 9 数据块子程序	(263)
§ 9. 2 源程序文件的操作命令	(156)	§ 5. 10 程序举例	(264)
§ 9. 3 数据文件的基本操作	(159)	小结	(271)
小结	(171)	习题	(272)
习题	(171)	第六章 结构化程序设计简介	(273)
第三篇 FORTRAN 77 语言及程序设计	(172)	§ 6. 1 结构化程序设计原理	(273)
第一章 基础知识	(172)	§ 6. 2 结构化程序设计流程图	(276)
§ 1. 1 FORTRAN 语言的基本符号	(172)	第七章 文件	(277)
§ 1. 2 数据	(173)	§ 7. 1 文件的基本概念	(278)
§ 1. 3 内部函数	(178)	§ 7. 2 文件输入/输出语句	(279)
§ 1. 4 表达式	(179)	§ 7. 3 文件的打开和关闭语句	(280)
§ 1. 5 FORTRAN 程序结构及书写格式	(183)	§ 7. 4 其他文件语句	(283)
§ 1. 6 框图符号和程序框图	(186)	§ 7. 5 文件应用举例	(285)
小结	(187)	小结	(287)
习题	(188)	习题	(287)
第二章 赋值语句及输入/输出语句	(188)	第四篇 Pascal 语言程序设计	(288)
§ 2. 1 赋值语句	(189)	第一章 Pascal 语言基础	(288)
§ 2. 2 输入/输出语句	(190)	§ 1. 1 Pascal 语言基本符号	(289)
§ 2. 3 PARAMETER 语句及 DATA 语句	(204)	§ 1. 2 标准数据类型	(291)
小结	(205)	§ 1. 3 直接量、常量和变量	(293)
习题	(205)	§ 1. 4 基本运算、表达式和赋值语句	(294)
第三章 控制语句	(206)	§ 1. 5 Pascal 语言基本程序结构	(296)
§ 3. 1 STOP, PAUSE 和 END 语句	(206)	§ 1. 6 输入语句	(298)
§ 3. 2 GOTO 语句	(208)	§ 1. 7 输出语句	(299)
§ 3. 3 IF 语句	(210)	§ 1. 8 标准函数	(302)
§ 3. 4 循环语句	(217)	§ 1. 9 简单程序举例	(305)
§ 3. 5 CONTINUE 语句	(219)	小结	(306)
§ 3. 6 多重循环	(220)	习题	(306)
§ 3. 7 程序举例	(221)	第二章 判定和循环语句	(306)
小结	(225)	§ 2. 1 IF 语句和复合语句	(307)
习题	(225)	§ 2. 2 CASE 语句	(310)

§ 2. 3 FOR 循环语句	(311)	§ 4. 4 数组类型	(372)
§ 2. 4 WHILE 循环和 REPEAT 循环语句	(313)	§ 4. 5 记录类型	(376)
§ 2. 5 标号说明和 GOTO 语句	(319)	§ 4. 6 应用举例	(381)
§ 2. 6 循环的嵌套问题	(323)	小结	(384)
§ 2. 7 应用举例和程序设计问题的讨论	(326)	习题	(385)
小结	(331)	第五章 文件和动态结构	(385)
习题	(331)	§ 5. 1 文件概念和说明	(386)
第三章 过程和函数	(332)	§ 5. 2 文件的标准操作和过程	(387)
§ 3. 1 过程的编写和调用	(332)	§ 5. 3 TEXT 文件	(391)
§ 3. 2 函数说明与函数调用	(339)	§ 5. 4 指针	(394)
§ 3. 3 关于过程和函数说明中的变量和 参数的讨论	(343)	§ 5. 5 单链表	(396)
§ 3. 4 过程嵌套和递归	(349)	小结	(403)
§ 3. 5 过程参数和函数参数	(357)	习题	(403)
§ 3. 6 外部过程和外部函数	(361)	第六章 Pascal 程序的运行	(403)
小结	(364)	小结	(414)
习题	(365)	附录	(415)
第四章 用户自定义数据类型	(367)	1. 1 ASCII 表	(415)
§ 4. 1 枚举类型	(367)	3. 1 FORTRAN77 内部函数表	(416)
§ 4. 2 子界类型	(370)	3. 2 程序单位中语句和注释行的顺序	(418)
§ 4. 3 集合类型	(371)	4. 1 Pascal 语言标准函数表	(419)

第一篇 计算机及计算导论

现代电子数字计算机，是一种高度自动化的，以计算程序存储和顺序执行为特征的，对各种数字化信息进行高速处理的电子设备。

第一台电子计算机 ENIAC (Electronic Numerical Integrator and Calculator) 是 1945 年美国毛奇来 (Mauchley) 和艾克 (Eckert) 在宾夕法尼亚大学的摩尔电工学院为马里兰州阿具火炮试验场研制的，它标志着计算机进入其第一代的发展时期 (1945 年~1954 年)。第一代计算机主要特征是：以电子管取代了原来计算设备使用继电器、半机械装置构成基本逻辑电路；使用磁鼓或延迟线做主存；用机器语言或汇编语言编制程序；用途主要为科学计算。关于第一代计算机的结构和性能，以 ENIAC 为例说明如下：该机共使用 18 800 支电子管，占地 170m²，重 30t。运算速度加法每秒 5 000 次；乘法每秒 56 次。

半导体物理和晶体管制造技术的发展，推动计算机进入第二代 (1955 年~1964 年)，其代表机型为 IBM 公司 1958 年推出的全晶体管 7090 计算机。它使用 43 000 支晶体管，主存使用磁芯存储器。体系结构从以中央处理器为中心，改为以主存储器为中心。运算速度每秒几十万次至一百万次。在此期间，另一关键进展是面向过程的高级程序设计语言及其编译程序的产生和发展，其代表为 1957 年 IBM 公司的一个研究小组关于 FORTRAN 语言的研究报告和相继开发的 COBOL 和 ALGOL 语言。由于晶体管和磁芯均在“冷”工作状态下运行，从而硬件可靠性大幅度提高，加之程序设计的简化，使计算机应用范围，从纯科学计算扩大到数据处理、自动控制和事务处理。

集成电路的发展，使计算机进入以中、小规模集成电路构成基本控制、运算逻辑电路为主要特征的第三代发展时期 (1964 年~1974 年)。其代表机型为 1967 年 IBM 公司宣布的 IBM360 系统，运算速度提高到每秒几百万次至几千万次。在这一发展时期，计算机软件研究迅猛发展，使软件研究开发费用逐渐超过硬件开发费用。计算机操作系统、数据库及交互式语言相继出现和完善，使计算机应用几乎深入社会各个领域。

随大规模和超大规模集成电路的发展，一块硅片上由集成上千个晶体管提高到上亿个元件，使计算机进入以使用大规模集成电路和以半导体存储器做为主存储器的第四代 (1975 年以后)，从而形成大、中、小、微系列机型和规模。尤其是微型计算机的发展，由于它体积小、价格低、使用灵活，使计算机的应用范围，从机房深入人们的办公室和家庭。值得特别指出的是，目前软件发展迅速，各种语言、操作系统、数据库、人工智能和门类繁多的应用程序的发展，使软件已经成为一个重要的产业部门。

本篇讨论计算机系统构成，计算机内数的表示，计算过程的实现和表示方法，及计算过程中的算法设计和数据结构方法等一系列计算机及其应用方面的基本概念和方法。

第一章 计算机及系统构成

§ 1. 1 社会对计算机的需求

本节通过计算机在几个重要领域的应用, 讨论计算机的使用特点。

一、计算机作为信息处理机

在某种意义上, 计算机是一种信息处理设备。它可以接收信息, 对信息进行加工, 使用信息进行逻辑推理和判断, 从而做出某种辅助决策, 并把信息转换成不同形式进行输出或实施某种控制。由于现代信息处理系统, 如文献检索, 工厂、企业的生产管理等, 要求计算机处理的信息量很大, 所以计算机使用一个专门的数据管理软件——数据库管理系统对数据进行组织和管理。因此, 计算机信息管理系统通常由负责管理和存储数据的数据库系统和负责对数据信息进行加工的应用程序两个基本部分组成。以工厂的生产管理系统为例, 一个工厂要协调地完成其生产任务, 必须有如下两方面的活动: ①物质流: 就是生产实体。包括从原材料采购、库存, 各车间、加工工序到成品库直至销售的整个生产过程。②信息流: 就是生产调度管理实体。从班组和车间的统计人员对生产状况的统计, 经生产管理职能部门汇总, 提供给厂级生产作业会议根据生产实体统计来的信息, 做出相应决策, 调度协调生产活动。

信息流中的信息, 源于生产实体, 又反过来调动生产实体永远落后于物质的流动, 又要调动、指挥物质的流动。这种滞后越小, 其指挥越近于实际的生产状态, 也就越有效。为此可以建立如图 1-1 所示的计算机生产管理系统。其工作过程可简述如下: 首先由生产实体的各班组对每班生产情况按规定要求进行统计, 形成描述本班组生产进度的各种表格, 使生产活动得以规范化, 数据化描述。输入计算机, 以数据形式存于数据库中, 形成一种反映生产实体生产状态的数据映象。当厂级生产作业会议要研究、调度生产时, 根据需要向计算机发出命令, 计算机通过各种应用程序向数据库发出相应询问, 并根据数据库提供的数据进行统计、分析或辅助决策, 供决策人员分析生产实体的现行状态。生产管理人员则根据计算机提供的报告, 加上他们自身的生产和管理经验, 对生产实体的运行状况做出比较及时正确的判断, 从而形成指挥生产的各种决策, 以计划的形式下达生产实体, 实施指挥。构成一个从生产实体到计算机再到生产决策机构又回到生产实体的信息循环。

计算机作为信息处理设备的其他应用, 也大抵如此。如文献检索系统, 它把来自社会的生产和科学研究文献, 如论文、报告及其他情报抽象化、规范化为文摘、主题词等, 以数据形式存于数据库中, 而应用程序则分别在主机和通过国际通信卫星进行通信的检索终端上运行。根据用户提供的检索主题词及其逻辑运算表达式, 对数据库中的数据进行处理, 达到相应的检索目的, 并按题目、发表年月、所载期刊及内容摘要等项目分级提供给用户。检索结果, 经科研、技术人员消化理解, 用于生产或研究工作, 其成果又以论文、报告等形式发表并进入检索系统数据库, 从而构成信息的循环, 达到服务于社会的目的。

二、计算机作为控制设备

计算机作为控制设备，广泛用于石油、化工、冶金、电力等工业的生产过程控制和机械、轻工行业的数控和群控。另外机械、航天航空等工业对其整机及另部件进行可靠性或其他动、静态性能测试，也广泛使用计算机实现其测试系统的快速检测控制和动态数据处理。

以生产过程控制为例，现阶段工业控制机其主机就是一台小型或超级微型通用电子数字计算机。实现工业过程控制，单依靠通用计算机本身当然是不够的，所以工业控制机这个概念应包括用于工业控制的计算机和它与被控现场联系的所谓外围通道两部分。通常外围通道包括：模拟量输入和模拟量输出通道，开关量输入和开关量输出通道，以及对生产或测试现场进行人工干预的系统运行操作台，如图 1-2 所示。

生产现场被测控的参数往往是多种形式的物理量，可归纳为两类：①连续变化参数。如温度、压力、流量等，称为模拟量。②数字量参数。如阀门的开与关，某种物理参数的限内与限外，移动件行程的开与关，或脉冲串信号。这些参数统称为开关量。计算机只能接收和识别数字信息，故模拟信号须先经相应变送器，变成与对应物理量成正比的标准电流信号（0~10mA）或电压信号（0~5 或 10mV），然后在计算机控制下，经采样开关电路对各被测信号进行巡回或定点检测。采样开始，将各路检测信号分时间片送入模拟/数字（A/D）转换器，把与被检测物理量成正比的标准电流（或电压）信号转换成数字量，送入计算机，做相应的识别和处理。数字信号则经寄存器（对于开关量）或计数器（对于脉冲信号）直接送入计算机。计算机控制两类输入通道，接收各检测点检测的数字信息后，分别做出判断或根据某种数学模型进行计算得出最佳回控值，实施回控，以便使生产现场运行于最佳状态。

由于计算机给出的回控信号，其调整量也都是以数字的形式表示的，但生产或试验现场的执行元件，就其对输入信号的要求，也分为两类：一类要求提供连续的电流或电压信号；另一类则要求开关或数字量信号。前者由模拟量输出通道先经数字/模拟（D/A）转换装置，产生相应模拟控制电流或电压，经输出切换装置控制送入正确的执行元件；后者则经开关量输出通道切换，送入对应执行机构，实施回控。

三、计算机做为辅助设计设备

计算机辅助设计（CAD）、辅助制造（CAM）是 70 年代以来在生产制造部门发展起来的一

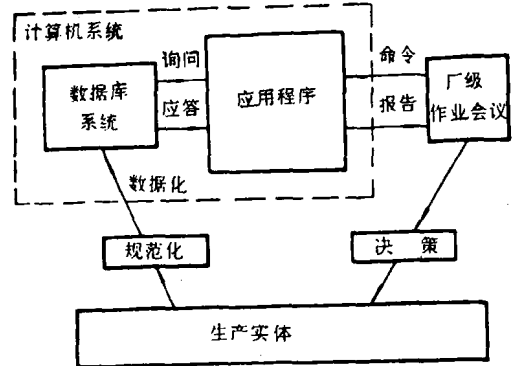


图 1-1 工厂的生产信息管理系统示意图

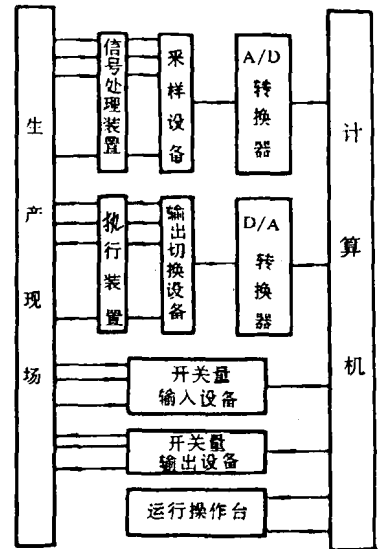


图 1-2 工业过程控制设备

项崭新的计算机应用领域,已广泛应用于机械、电子、建筑等行业。其发展的动力是这些行业为缩短新产品开发周期,提高设计的效率和质量,增加其产品的竞争能力。

早期 CAD 技术是以计算机绘图为中心的。如用计算机产生印刷电路板和集成电路的掩模图,机械另件和装配图,或建筑管道布置图等。为进一步提高设计质量,人们把计算机辅助设计同计算机辅助工程分析和工作过程模拟结合起来,形成计算机辅助工程(CAE)。如机械、建筑使用的有限元方法进行应力和热力场分析,电子领域则广泛采用逻辑模拟和电路模拟,机械行业对产品的工作过程模拟等。CAD 与 CAE 技术的结合,提高了设计的质量和优化设计的能力,往往简化了样机的制造过程,缩短产品的设计周期。70 年代以来,随数控机床和工业机器人的广泛应用,人们开始把计算机辅助产品设计和自动产生其加工程序结合起来,实现了从产品到其加工程序设计的一体化的计算机辅助设计过程,从而人们又提出了计算机辅助制造(CAM)技术。80 年代后期,随柔性加工技术的进一步发展,为适应机械、电子等行业小批量、多品种、短周期,以提高产品市场竞争力的生产特点,出现了以工程数据库技术和人工智能为中心,把 CAD、CAE、CAM、CAPP(计算机辅助工艺设计)、机器人自动编程、计算机辅助信息管理(CIS)和决策(DSS)集成在一起的计算机综合制造系统(CIM)。所以建立大型的 CAD 系统,往往需要一种较复杂的计算机软、硬件支持环境。有两种基本的结构形式:①分布式 CAD 系统,通常由几台 CAD 工作站或通用高档微机组成;②集中式 CAD 系统,由一台多终端高级微型或小型计算机组成。无论那种基本结构形式,其主机实际上也是一台或几台通用电子数字计算机,并要求运算速度在百万次以上,内存应大于 4~16MB,由于 CAD 系统要处理图形,故应配备图形显示终端和其他图形输入、输出设备。

CAD 支持软件主要有如下几种:

1. 工程数据库管理系统。为 CAD 系统提供动态和静态的数据支持环境。静态数据库应包括国家和部颁的各种标准数据,亦称标准数据库;静态数据库的另一部分是用以存放标准件和标准画法的所谓图形数据库。动态数据库亦称设计工作库,它支持动态设计数据、图形的存储和管理。工程数据库是 CAD 系统的中心支持环境。

2. 图形支持软件。图形软件主要任务是建立设计对象的实体几何模型,确定形状及尺寸。在此基础上进行各种分析评价,并交互地实施再设计和再评价,直至形状及尺寸完全确定。可分为两个级别:①二维图形软件:主要用于绘制传统的工程图,其主要功能是产生各种平面图,图形的变换和投影,对图形进行编辑,标注尺寸及技术工艺要求等。②三维图形软件:除二维图形软件的全部功能外,主要特点是支持三维实体造型,所产生的在复杂光照模型下,带浓淡色彩的所谓“真实感”图形,其完美几乎可以乱真。另外,还可模拟刚体运动,弹性、塑性变形及阻尼等对工件的影响,可用来显示设计对象在正常及非正常情况下应力分布,热分布,显示物体变形及损坏过程。有些过程使用常规试制及实验方法也是难以分析和显示的。

3. 工程分析软件。包括:①有限元力学分析程序。用于另部件应力分布分析,热力场分析和显示。②系统动态特性分析及机构分析程序。一般采用模拟方法,分析设计产品的噪声,振动等特性,它能计算产品的自然频率,振动类型及对动载荷的各种影响,给出各另件结合部位的应变,运动能及能量损耗等。③优化设计方法库。主

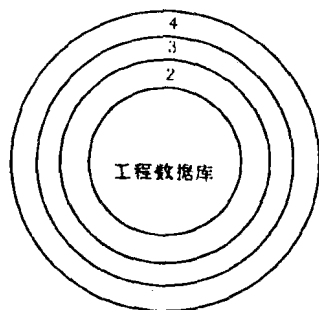


图 1-3 CAD 系统的
软件及数据支持

要包括设计中应用的优化方法（程序）供设计者调用。

图 1-3 给出了 CAD 系统的软件分层结构，其中：1 层为工程数据库，包括各种标准数据及标准图形及画法库；2 层为工程数据库管理系统；3 层为图形支持及工程分析软件层；4 层为各种用户应用设计软件层。

§ 1. 2 计算机的基本结构及工作

一、计算机的基本结构

计算机的基本结构包括如下四个主要部分（图 1-4）：①存储器：程序和被处理的数据都以二进制代码存放在存储器中。②控制器：在它控制下，指令依次从存储器取出，然后对其进行解释并执行。③运算器：在控制器控制下，完成指令所需的算术和逻辑运算。④输入和输出设备：完成人-机间的通信和信息转存。

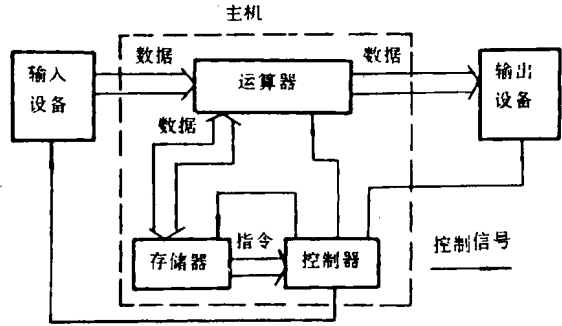


图 1-4 计算机的基本结构

一般把运算器和控制器合在一起称为计算机的中央处理单元（Central Processing Unit），简记为 CPU；把 CPU 与存储器合在一起称计算机的主机；输入、输出设备则统称为计算机的外部设备（peripheral）。现阶段使用的这种结构叫冯·诺依曼型计算机，其最基本的工作特点是“程序存储”：把计算机要执行的命令和待处理的数据经输入设备送入计算机，经由运算器逐条依次转存入存储器。执行时，由控制器再依次将命令逐条取出解释，形成一系列控制信号，控制运算器对存储器中的数据进行加工，并将运算结果和中间运算结果输出或存入寄存器中。所以，在计算机中主要有两类信息在流动：①数据信息。各原始数据，由输入设备输入并存入存储器；运算过程中，数据从存储器被读入运算器参加运算；运算的最终或中间结果，则由运算器传送给存储器存储或输出设备进行输出。②控制信息。由人输入计算机的命令（程序），也以数据的形式存于存储器。但在执行时，计算机却把它们读入控制器，并译码产生相应的控制信号，去控制运算器和计算机其他部件。根据预先编好的次序，命令逐条取出并解释执行，从而控制运算器一步步进行各种运算；控制存储器及时进行读、写；控制输入、输出设备按时启停，从而完成规定的系列操作。

二、指令

要让计算机自动地完成一项复杂处理任务，就要把待完成的任务分解成一系列基本操作，以命令的形式存入存储器，而后逐条执行。描述一项计算机可直接译码（解释）并执行的命令所用的代码，叫指令（instruction），每一条指令对应一串规定的二进制代码。计算机可以解释执行多少条指令，每一条指令执行什么操作以及每条指令的编码，都是设计计算机时事先规定好的。计算机的硬件设计的任务就是如何保证所有指令被读入控制器，经指令译码器译码后，在时钟节拍信号配合下，所产生的操作脉冲信号能够正确控制各种部件完成相应指令规定的运算或操作。一台计算机能直接译码执行的所有指令集合，叫该计算机的指令系统

(instruction system)。

指令代码通常由两部分组成：第一部分表示本指令应执行的操作，叫指令的操作码。它被读入控制器，译码形成执行相应操作的控制信号。第二部分是参加运算的数或其地址，叫指令的操作数 (operand)。由于微机字长较短，所以指令又以其占用的字节 (8 个二进制位) 数分为单字节、双字节和多字节指令。以 Z-80 处理器指令系统的下述三条指令为例，说明指令的编码结构：

(1) 0011 1110 操作码
 0000 0111 操作数 (7)

是一条双字节指令。其操作是把第二字节中的数送入累加器 A。

(2) 1100 0110 操作码
 0000 1010 操作数 (10)

也是一条双字节指令。表示把累加器 A 中的数与本指令第二字节中的数相加，结果存入 A 中。

(3) 0111 0110 操作码

是一条单字节指令，表示让计算机停止所有操作。

上述指令描述方式叫指令的机器码。使用计算机解题时，把要解的问题用指令分解成一项项基本操作，形成一个指令序列，这样的指令代码序列就叫程序 (program)。使用这样的机器指令码编写程序，很不方便，因为由 0 或 1 组成的位串构成的指令难记忆，易出错。于是人们把每一条指令用一个称为助记符的符号代表，就形成了所谓汇编语言。仍以上述三条指令为例，可分别写成如下形式：

LD A, n
ADD A, n
HALT

它们的含义分别与上述指令 (1) ~ (3) 相同，但实际写程序时应把第 1, 2 条中的 n 分别代之以被送入累加器 A 的数和与 A 中的数相加的数的值。应注意，无论使用汇编语言还是使用其他高级程序设计语言编写的程序，都必须先由编译或解释程序，解释成形如指令 (1) ~ (3) 的机器指令序列才能执行，因为机器指令是计算机唯一可以直接译码执行的指令形式。

三、总线

计算机的实际结构是比较复杂的，它需要由多块组件板构成。各板间的信号传递是通过印刷电路板插座间的连线实现的。人们希望各插座之间的连线具有通用性，使计算机的任何一块电路板，插在任何一个同类型插座上，计算机均可正常工作。这样也可使不同厂家生产的组件板具有通用性和互换性，于是产生了总线 (BUS) 的概念与标准。例如 S-100 总线规定采用双面印刷电路板，插座共有 100 个插脚，每面 50 个。对每个插脚的输入或输出信号都做了明确规定：包括输入、输出数据线 16 根，控制用信号线 39 根，地址线 16 根，和电源、地线以及一些未定义线。

因此总线其实就是一组能为多个插件板服务的公共信号传输线，机器中各插座的插脚具有相同定义的都用导线联在一起。在插件板内，对每一插脚，使用该信号的就连入相应逻辑元件，不用的就闲置 (不往板内连)。从而保证各插件板插入任一插座均可正确组成计算机系统。为保证各插件板能正确使用总线传递信号，各总线中的信号线，进入插件板后，都有适当的门控电路控制信号的输入、输出。传递信号时，只要控制器发出信号，打开输出插件和

输入插件相应的输出和输入门控电路，就可以分时地使用总线完成不同插件板间的信号传送任务。

为概念上明确,把总线(如 S-100 总线定义的 100 条线)按用途分为数据总线,地址总线和控制总线三类;根据总线上传送信号的方向是否可逆,又分成单向和双向总线两种。使用总线把主要部件:运算器、控制器、主存储器和输入输出设备连接起来构成计算机的结构方式叫计算机的总线结构方式。

四、计算机主机结构与工作

计算机主机的基本结构如图 1-5 所示。由于计算机只处理二进制数据和开关控制信号,故其所有运算、存储和控制部件,都是由只具两种工作状态的所谓开关电路组成。图中虚线以上为 CPU;虚线以下为主存储器。通常它们分别放在不同的插件板上,故 AB (Address Bus) 和 DB (Data Bus) 分别为总线中的地址和数据总线。在要求运算速度不高的计算机中,为减少印刷电路板内部走线,在 CPU 内部,也使用总线方式联接各主要组件,如图 1-5 中 AR、DR 寄存器以上的总线,叫内部总线。

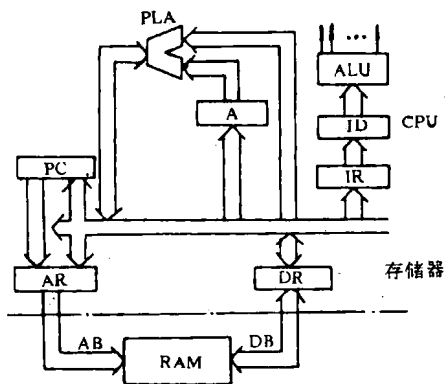


图 1-5 计算机主机结构

算术逻辑运算单元 ALU (Arithmetic Logic Unit) 是执行算术和逻辑运算的部件, 它以累加器 (实际上就是运算器内部的一种数据寄存器) A 的内容做为一个操作数, 另一操作数视指令不同可由内部数据总线提供, 也可以由其他寄存器 (图 1-5 中未画出) 提供, 也可以是数据寄存器 DR (Data Register) 提供的读自主存储器的数据。操作的结果通常放在累加器 A 中。图 1-5 模型机中, ALU 和累加器 A 组成运算器, 虚线上方其余部分均属控制器。

PC 为程序计数器, 它由具有计数 (自动加 1) 功能的一组可预置双稳电路组成。如果要求计算机执行的指令以寻址单元为存储单元按顺序预先存于主存储器内, 则工作开始, 只需将存储指令的首地址予置给 PC。然后每读出一条指令, 则 PC 自动执行加 1 操作, 于是 PC 可自动产生下一条指令的存储地址。AR (Address Register) 是地址寄存器, 它把要寻址的单元地址通过地址总线 AB, 送至主存储器的地址译码器, 实现对指定存储单元的读写操作。要寻址的主存单元, 其内容可以是指令, 地址由 PC 提供; 也可以是数据, 则地址由指令中的数据部分提供。把指令从主存读入数据寄存器 DR 再转入指令寄存器 IR (Instruction Register) 的过程叫计算机取指令阶段。在取指令阶段, 之后, 计算机进入指令执行阶段。IR 中的指令码经指令译码器 ID (Instruction Decoder) 进行译码, 通过控制电路 (PLA), 与节拍信号配合, 产生执行相应指令所需的各种控制信号, 控制运算器等部件完成指令应进行的各种操作。所以, 若程序已存于主存, 则计算机的工作过程就是循环地执行取指令和执行指令这样两个工作阶段。计算机从暂停状态进入运行状态后, 把第一条指令所在地址预置入 PC, 并进入第一个取指令阶段, 从存储器中读出要执行的指令内容, 经 DR 送到 IR, 然后进入指令执行阶段。由指令译码器对指令进行译码, 实现对所读指令的解释。根据译码器译码结果, 由控制部件产生相应信号, 控制计算机执行指令指示的操作。一条指令执行完成后, 再进入下一条指令的取指令阶段, 如此循环, 直到程序结束。

设有如下三条指令组成的程序：

```
LD    A, 7
ADD   A, 10
HALT
```

并已经以机器码的形式依次连续存于从 00 地址开始的主存单元中。以此为例，研究计算机对上述程序是如何逐条执行的。

1. 开始执行，首先把第一条指令的地址 00H 赋给 PC，计算机进入取指令阶段，顺序执行如下操作：①把 PC 内容 (00H) 送入地址寄存器 AR；②待 PC 内容稳定置入 AR 后，PC 内容加 1，变为 01H，为取第一条指令的第二字节做好准备；③地址寄存器 AR 中的地址通过地址总线 AB 送入存储器的地址译码器，译码后选中存储器的 00 号单元；④CPU 发出读命令，00 单元内容 3EH 被读入数据总线 DB；⑤DB 上的内容打入数据寄存器 DR；⑥因处于取指令阶段，取出的数据应按指令处理，CPU 把 DR 内容送入指令寄存器 IR。参考图 1-6。然后计算机进入执行第一条指令的阶段。

在指令执行阶段中，IR 中的指令代码经指令译码器 ID 译码，由控制逻辑阵列 PLA 发出与执行本指令相应的操作控制信号，控制指令操作的执行。第一条指令的操作是把操作数 (存于指令第二字节) 送入累加器 A。故 PLA 发出的控制信号控制计算机执行下述操作 (参考图 1-7)：①把 PC 内容 (01H) 送至地址寄存器 AR；②待 PC 内容可靠进入 AR 后，令 PC 加 1，变为 02H，指向下一条指令；③把地址寄存器 AR 中的地址送入地址总线 AB，经存储器译码器译码后打开 01H 单元的读出控制门电路；④CPU 发出读命令，01H 单元中的内容 07H 被读到数据总线上；⑤把数据总线上的数据打入数据寄存器 DR；⑥根据 ID 对指令的译码，已知读的为 (立即) 操作数，并应存入累加器 A，故 CPU 令 DR 通过内部数据总线将其内容传至 A。至此，第一条指令执行完毕。

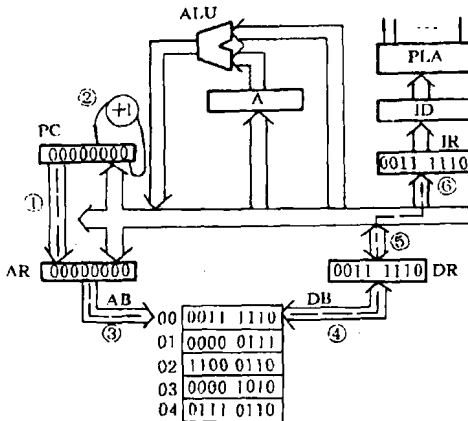


图 1-6 取第一条指令 (操作码)

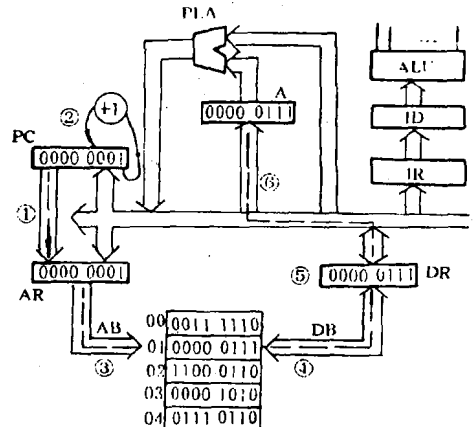


图 1-7 取立即数操作

2. 第一条指令执行完成后，计算机进入下一条指令的取指令阶段，其操作流程与第一条指令取指周期完全相同：①把 PC 内容 02H 送入地址寄存器 AR；②PC 内容可靠进入 AR 后，PC 自动加 1 变成 03H，把地址指向本指令第二字节；③AR 通过地址总线 AB 将地址送存储器译码器，译码选中相应存储单元；④CPU 发出读命令，第二条指令 C6H 被读出 (上数据总线 DB)；⑤CPU 发出控制信号，把 DB 上的数据置入 DR；⑥因处于取指令阶段，故计算机将 DR 内

此指令为加法指令,要求以 A 中内容做第一操作数,指令第二字节所存为第二操作数,执行加法。故指令译码后 PLA 产生的操作控制信号,将依次完成如下操作(参考 1-9):①PC 内容 03H 送入 AR;②PC 内容可靠进入 AR 后,PC 自动加 1 指向第三条指令;③AR 通过地址总线,将地址 03H 送入存储器译码器,选中 03 单元;④CPU 发出读命令,将 03 单元内容 0AH 读入总线 DB;⑤总线 DB 上的数据打入数据寄存器 DR;⑥指令译码时已知读出的为操作数,并要求它与累加器 A 中内容相加,故将 DR 内容通过内部数据总线送至 ALU 的一个输入端;⑦将 A 中内容 07H 送 ALU 另一输入端并执行加法;⑧相加结果回送累加器 A。

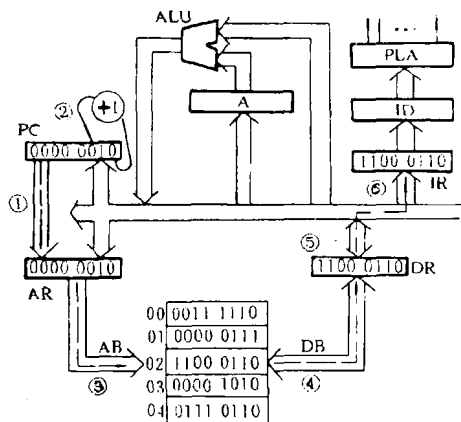


图 1-9 执行 ADD 指令操作示意图

至此，第二条指令执行完毕。计算机进入第三条指令的取指阶段，与上述各取指过程相似的操作序列取出第三条指令，在执行阶段译码后产生停机信号，机器停止一切操作。至此上述三条指令组成的程序执行完毕。

存储器是计算机中具有“记忆”功能的部件。解题程序的全部指令，参与运算的全部数据和部分运算结果都要存放在存储器中，使计算机实现“程序存储”和自动连续执行一系列的规定操作。

二进制数的每一位，只有两种取值状态：0 或 1。故具有两种稳定状态的元件都可做为存储器的基本记忆元件，用以表示二进制数的一位，如继电器、磁芯、双稳触发器等。一个二进制数由若干位组成，它们做为一个整体，应具有统一的存储地址，称为存储字或存储单元。存储单元是存储器的最小寻址单位，所以也是其读出或写入的最小单位。一般一个存储单元，可存放一个字节 (Byte)，也就是 8 个二进制位，所以存储器存放一个字节，要使用 8 个记忆元件。

计算机主存储器简称主存，就是通常人们所说的内存。现阶段主存储器一般都使用半导体随机存储器，简称为 RAM (Random Access Memory)。它是一种可读、可写的半导体存储电路。有两种类型：金属-氧化物-半导体存储器和双极型半导体存储器。前者使用 P 沟道或 N 沟