



中国计算机学会教育专业委员会 推荐
全国高等学校计算机教育研究会 出版
高等学校规划教材

计算机组成原理与 汇编语言程序设计

傅廷祯 王正智 徐洁 俸智丹 编
朱怀芳 主审

计算机学科教学计划1993



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
URL: <http://www.phei.co.cn>

高等学校规划教材

计算机组成原理与汇编语言程序设计

俸远祯 王正智 徐 洁 俸智丹 编

朱怀芳 主审

15630

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书系统地介绍计算机组成原理与汇编语言程序设计方法, 强调整机概念与软硬结合。全书分为四篇: 第一篇介绍 CPU 子系统, 首先从机器指令级描述其基本组成, 然后介绍计算机中数据信息的表示方法及指令系统, 再通过一台模型机的设计阐明指令执行流程与微操作。第二篇介绍 8086/8088 汇编语言, 讲解程序设计方法, 并提供较多程序实例。第三篇讨论运算功能的实现方法, 解释运算指令功能的硬件实现方法, 以及通过程序扩展运算功能的方法。第四篇介绍存储器子系统与 I/O 子系统, 包含主存储器设计、存储系统组织、I/O 控制方法、总线、接口、常用 I/O 设备工作原理、I/O 程序设计等内容。

本书可作为高校计算机专业教材, 也可供从事计算机专业的科技人员参考。

图书在版编目(CIP)数据

计算机组成原理与汇编语言程序设计/俸远祯等编. -北京:电子工业出版社, 1999.7

高等学校规划教材

ISBN 7-5053-3863-3

I. 计… II. 俸… III. ①电子计算机-基础理论-高等学校-教材②汇编语言-程序设计-高等学校-教材 IV. TP301

中国版本图书馆 CIP 数据核字(1999)第 30268 号

丛 书 名:高等学校规划教材

书 名:计算机组成原理与汇编语言程序设计

编 著:俸远祯 王正智 徐 洁 俸智丹

审 校 者:朱怀芳

责任编辑:张凤鹏

特约编辑:袁 英

印 刷 者:民族出版社印刷厂

出版发行:电子工业出版社 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话:68214070

经 销:各地新华书店

开 本:787×1092 1/16 印张:33.5 插页:1 字数:883.3 千字

版 次:1999 年 7 月第 2 版 1999 年 7 月第 1 次印刷

书 号:ISBN 7-5053-3863-3
G·303

定 价:38.00 元

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换
版权所有·翻印必究

出版说明

中国计算机学会教育专业委员会和全国高等学校计算机教育研究会(以下简称“两会”),为了适应培养我国 21 世纪计算机各类人材的需要,根据学科技术发展的总趋势,结合我国高等学校教育工作的现状,立足培养的学生能跟上国际计算机科学技术发展水平,于 1993 年 5 月参照 ACM 和 IEEE/CS 联合教程专题组 1990 年 12 月发表的《Computing Curricula 1991》,制定了《计算机学科教学计划 1993》,并组织编写与其配套的首批 18 种教材。现推荐给国内有关院校,作为组织教学的参考。

《计算机学科教学计划 1993》是从计算机学科的发展和社会需要出发提出的最基本的公共要求,不是针对某一具体专业(如计算机软件或计算机及应用专业),因此它适用于不同类型的学校(理科、工科及其它学科)、不同专业(计算机各专业)的本科教学。各校可以根据自己的培养目标和教学条件有选择地组织制定不同的教学计划,设置不同的课程。本教学计划的思想是将计算机学科领域的知识,分解为九个主科目(算法与数据结构、计算机体系结构、人工智能与机器人学、数据库与信息检索、人-机通信、数值与符号计算、操作系统、程序设计语言、软件方法学与工程)作为学科的公共要求;对计算机学科的教学归结为理论(数学)、抽象(实验)和设计(工程)三个过程,并强调专业教学一定要与社会需要相结合。另外,还提出了贯穿于计算机学科重复出现的十二个基本概念,在深层次上统一了计算机学科,对这些概念的理解和应用能力,是本科毕业生成为成熟的计算机学科工作者的重要标志。

为了保证这套教材的编审和出版质量,两会成立了教材编委会,制定了编写要求和编审程序。编委会对编者提出的编写大纲进行了讨论,其中一些关键性和难度较大的教材还进行了多次讨论。并且组织了部分编委对教材的质量和进度分片落实,有的教材在编审过程中召开了部分讲课教师座谈会,广泛听取意见。参加这套教材的编审者都是在该领域第一线从事教学和科研工作多年,学术水平较高,教学经验丰富,治学态度严谨的教师。这套教材的出版得到了电子工业出版社的积极支持。他们把这套教材列为出版社的重点图书出版,并制定了专门的编审出版暂行规定和出版流程,组织了专门的编辑和协调机构。

这套教材的编审出版凝聚了参加这套教材编审教师和关心这套教材的教师、参与编辑和出版工作者、以及编委会成员的汗水,他们为此作出了努力。

这套教材还得到电子工业部计算机专业教学指导委员会的支持,其中 11 本被选入 1996~2000 年全国工科电子类专业规划教材。

限于水平和经验,这套教材肯定还会有缺点和不足,希望使用教材的单位、教师和同学积极提出批评建议,共同为提高教学质量而努力。

中国计算机学会教育专业委员会
全国高等学校计算机教育研究会

教材编审委员会成员名单

- 主任：王义和 哈尔滨工业大学计算机系
副主任：杨文龙 北京航空航天大学计算机系（兼北京片负责人）
委员：朱家鏗 东北大学计算机系（兼东北片负责人）
龚天富 电子科技大学计算机系（兼成都片负责人）
邵军力 南京通信工程学院计算机系（兼南京片负责人）
张吉锋 上海大学计算机学院（兼上海福州片负责人）
李大友 北京工业大学计算机系
袁开榜 重庆大学计算机系
王明君 电子工业出版社
朱毅 电子工业出版社（特聘）

前 言

ACM/IEEE-CS《计算教学计划 1991》发表后,对我国计算机专业的教学体系也产生了重大的影响。以此为依据,中国计算机学会教育专业委员会和全国高等学校计算机教育研究会组织了数十位专家,经过多次讨论,制订了《计算机学科教学计划 1993》,并提出了相应的配套教材编写计划,本书就是其中的一本。此后,本书由全国工科电子类专业专业教材《计算机》教材编审委员会评选择优,作为“九五”规划教材,推荐出版。

本书将传统教学计划中的两门主干课程《计算机组成原理》与《汇编语言程序设计》,有机地组织成为一门课程,这是建立新教学体系的一种努力。其目的是围绕机器指令级(传统机器级),及与此紧密相关的微操作级和汇编语言级,更好地体现软硬结合,了解内部工作机制与编程求解问题相结合。从 CPU 和整机系统两个层次,从逻辑组成、内部工作机理、程序设计等三个方面,使学生建立比较清晰的整机概念。

本课程的参考教学时数约为 100~120 学时,全书内容可能比教学时数所允许的分量稍多一些,可供选取或让学生自学,部分实例可随技术发展而更新。由于涉及面较宽,我们将全书分为四篇,以体现模块性,具体教学顺序则可以适当改变。

本书的编写顺序体现了下述的教学思路:第一篇讲解 CPU 的组织及其内部工作机理,并从 CPU 的角度说明主存储器的基本特性,及主机与外部进行信息交换的基本方法,这就为讲解程序设计奠定比较坚实的硬件基础。第二篇在此基础上阐明程序设计方法。在第三篇讨论运算方法时,就可以既讨论硬件实现也讨论软件实现,既解释运算指令功能的硬件实现,也提供通过软件扩展运算功能的方法。第四篇介绍存储器子系统与 I/O 子系统,从 CPU 级发展为整机系统级。在讨论 I/O 子系统时可以软硬结合,既讨论硬件接口与 I/O 设备的逻辑组成及工作机理,也介绍软件调用方法与相应的 I/O 程序设计。因此,这样的组织方法并非简单地将《计算机组成原理》与《汇编语言程序设计》相拼合,而是一种有机的结合,在一定程度上避免了以往在课程先后顺序上所遇到的困难。

本书由俸远祯、王正智主编。俸远祯执笔第一章,并负责全书的统稿;王正智编写第五、六、八章;徐洁编写第二、三、四、七章;俸智丹编写第九、十、十一章。西南交大朱怀芳教授担任主审,他认真地审阅了书稿,提出了许多宝贵而重要的意见。《计算机学科教学计划 1993》配套教材编委会编委,北京航空航天大学杨文龙教授、电子科技大学龚天富教授、重庆大学袁开榜教授等,对本书的编写自始至终地给予了热情的指导和帮助。在编写过程中,还得到了电子科技大学计算机学院有关领导和老师们的关怀和支持。全国工科电子类专业教材《计算机》教材编委会(课程指导委员会)对本书的出版给予了支持和鼓励。谨向上述同志表示衷心的感谢。

作为建立新教学体系的一种尝试,本书肯定是不成熟的,而且编者水平有限。对书中的错误和不足,恳切希望大家批评指正。

编 者
1996 年 9 月

开 篇

第一章 绪 论

通常所讲的计算机，其全名是电子式数字计算机，它是一种能存储程序，能自动连续地对各种数字化信息进行算术、逻辑运算的快速工具。在这一定义中包含两个重要的基本概念：信息数字化，存储程序工作方式。本章一开始就强调它们，作为了解计算机组成及工作机制的基本出发点。

计算机系统是由硬件与软件组成的综合体，人们常采用层次结构观点去描述系统的组成与功能，分层次地分析与设计计算机系统。本章在简要叙述计算机系统的硬件、软件组成之后，将分别从程序设计角度及系统内部有机组成角度，介绍两种常用的层次结构模型。然后从教学组织的角度，再进一步强调有关的几种层次。

第一节 基本概念

一、存储程序工作方式与诺依曼机

1945年，冯·诺依曼在领导设计EDVAC计算机的过程中，提出了现代数字计算机的若干设计思想，被后人称为诺依曼体制，这是计算机发展史中的一个里程碑。采用诺依曼体制的计算机就被称为诺依曼机。几十年来计算机体系结构发生了许多演变，但诺依曼体制的核心概念——存储程序工作方式，仍是普遍采用的原则，绝大多数实用的计算机仍属于诺依曼机。我们将诺依曼体制中至今仍广泛采用的要点归纳如下。

1. 采用二进制代码表示数据和指令

计算机是处理信息的工具，所处理的信息泛称为数据。计算机的具体工作表现为执行程序，而程序的最终可执行形态是指令序列，即若干用数字代码表示的机器指令，它们是产生控制命令的信息基础。数据和指令都采用二进制代码表示，则它们在外形上并无区别，都是由0或1组成的代码序列，只是各自约定的含义不同而已。采用二进制，使信息数字化容易实现，可以用二值逻辑电路进行处理，这可称之为处理功能逻辑化。不仅数据，程序本身也可以作为被处理的对象，例如对源程序进行编译，就是将源程序当作被加工处理的对象。

2. 采用存储程序工作方式

这是数字计算机不同于其他计算工具之处，它意味着：

(1) 事先编制程序。为了用计算机求解问题，需要编制程序，它告诉计算机需要做哪些事，按什么步骤去做，并提供所要加工处理的数据。一件事往往要分步去做，每一步要求计算机执行的操作命令称为一条指令。计算机最终执行的程序，包含指令序列和有关的原始数

据。

(2) 将程序存储于计算机的存储器之中。

(3) 计算机在运行时将自动地、连续地从存储器中依次取出指令加以执行。因此，计算机的工作体现为执行程序，计算机功能的扩展在很大程度上取决于所存储程序的扩展。

存储程序工作方式是所有计算机都需采用的方式，是其他具体工作方式的基础，但传统诺依曼机在执行程序时，始终是以控制信息流（指令流）为驱动程序执行的因素，而数据信息流（数据流）则被动地被调用处理，因而称为控制流驱动方式。即按照指令的执行序列，依次读取指令，根据指令所含的控制信息，调用数据进行运算处理。

为了控制指令序列的执行顺序，可设置一个程序（指令）计数器 PC (Program Counter)，让它存放当前指令所在存储单元的地址。如果是顺序执行，则读取一个存储单元的指令代码之后，PC 内容加 1。如果需要程序转移，则将转移后的地址送入 PC。因此 PC 就像一个指针，指引着程序的执行。虽然程序与数据都采用二进制代码，在外形上没有区别，但可按照 PC 的内容作为地址读取指令，再按照指令给出的操作数地址去读取数据。由于多数情况下程序将顺序执行，所以大多数指令需要依次地紧挨着存放，除了个别立即使用的数据可以紧挨着指令存放外，一般将指令与数据分别存储于该程序区中的不同区域。

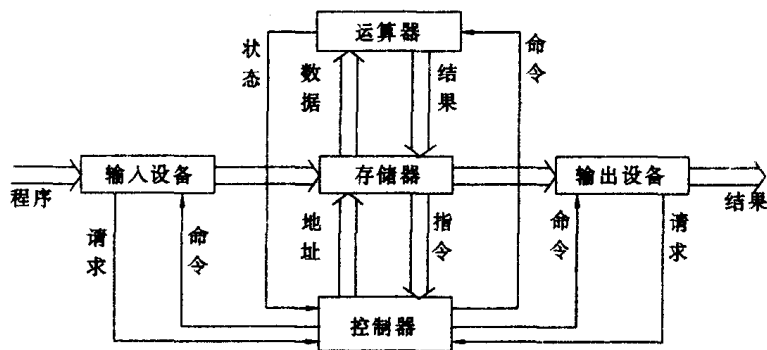


图 1-1 数字计算机的简单框图

3. 由运算器、存储器、控制器、输入装置、输出装置等五大部件，构成计算机硬件系统

如图 1-1 示，所编好的程序（包含指令与数据）通过输入设备送入计算机的存储器中。例如用键盘，将程序语句的字符转换为机器所能识别的二进制字符编码，然后送入存储器。

存储器按一定顺序存放所输入的程序，计算机处理所获得的结果也作为数据存放在存储器中。这就要求将存储器分成若干单元，并给每个单元分配一个地址，因此存储器的一个重要特性是能按地址存放或读取内容。对存储器而言，寻址的依据是地址码，存取的内容称为数据。

启动运行后，计算机将自动连续地执行程序。控制器向存储器送出某个地址码，从指定的存储单元中读取一条指令，送入控制器进行识别，分析该指令要求做何操作，操作数在何处。根据该指令的要求，控制器发出相应的操作命令，例如将某存储单元中存放的操作数取出，送往运算器进行所指定的运算处理，再将运算结果送回存储器中指定单元。运算过程中的某些状态信息，如有无进位、结果是否为 0、结果正负、是否溢出等，可送回控制器作为下一步操作的参考依据。

计算机执行完一段程序后，将获得所需的处理结果，再通过输出设备将结果送出。例如通过显示器在屏幕上显示程序运行情况与结果，或通过打印机将结果打印出来。

信息的输入、输出，存储器的读写，信息的运算处理等，都必须在控制器的控制下有序地进行。控制器控制着整个计算机的工作，决定在什么时候发出什么命令、进行什么操作。例如向存储器发出读命令或写命令及单元地址码，向运算器发出某种四则运算命令。在输入/输出设备与控制器之间常采取这样一种协调方式：当输入/输出设备做好相应准备之后，向中央控制器发出请求信号，然后控制器发出输入命令或输出命令。

早期的计算机常设置一个专门的控制台，操作人员可通过控制台启动或停止计算机的运行，或设置有关的运行参数，或在运行过程中进行某些必要的干预。现在的计算机则往往通过键盘实现上述操作，取代了传统的控制台。

二、信息的数字化表示

计算机是处理信息的工具。如前所述，在计算机中的信息可分为两大类：控制信息流、数据信息流。前者是指指令代码序列，及以此为基础所产生的控制信号（微命令）序列，它们是控制计算机工作的有关信息。数据流是计算机加工处理的对象，可分为数值型与非数值型（如字符、文字、图像、声音、逻辑信息等）。

为什么称为数字计算机呢？就是因为计算机中的上述信息都采用数字化表示，简单地讲它有两层含义。

1. 在计算机中的各种信息用数字代码表示

例 1-1 用数字代码表示数值大小，正负数符也用代码表示，如约定符号位为 0 表示数为正，符号位为 1 表示数为负。

例 1-2 用数字编码表示字符，以字符为基础可表示范围广泛的各种文字，以及程序设计语言等。西文字符、常用算符与控制符等，可用七位编码表示，每个字符编码占一个 8 位存储单元（即一个字节单元）。对于中文字符，可用二个字节的编码表示。

例 1-3 程序在计算机中的最终可执行形态是指令序列，每种指令按约定的指令格式，用一组数字代码表示。其中往往分为：操作码字段，操作数（或地址）字段。例如用 0000 表示传送，0001 表示加，0010 表示减……等操作命令。用某种程序设计语言编写的程序，先以字符编码形式输入计算机，再转换为指令代码。

例 1-4 将设备的工作状态抽象化，用不同编码分别表示，如：00 表示空闲，01 表示忙，10 表示完成一次操作。计算机根据某设备的状态编码，即可判定该设备所处状态。

例 1-5 用点阵近似地表示图像。

与字符相比，图像信息是随机分布的，规律性差。但可以将一幅图像细分为若干点，用这些点的组合逼近真实图像。如果分得足够细即点数很多，则在人的视觉中，这幅由许多点组成的图像几乎是连续的。相应地在计算机中用一位数字代码表示一个像点，如用 1 表示一个亮点，用 0 表示一个暗点。再按一定的扫描规律，将这些位（点）的信息以数字形式组织并存储起来，就可以对图像进行处理。

例 1-6 用数字表示声音。

在传统的电话装置中，受话器将声波转换为模拟信号形式的电流波，如要将其数字化，则可按一定频率对电流波进行采样，即等时间间隔地对电流波进行测量，将每次测得的电流幅

值用一个相应的数字量表示。如果采样频率足够高，则所获得的数字信息能逼真地保持声波信息，还原后发话器能逼真地再现原来的声音。采用数字化转换后，就可以用计算机对声音信息进行各种处理。

例 1-7 用数字代码表示逻辑信息。

计算机中的运算可分为：算术运算、逻辑运算。常用二值逻辑描述与处理逻辑问题，如用逻辑变量表示逻辑条件，1 表示该条件成立，0 表示条件不成立。将逻辑处理的可能结果分解为若干个逻辑命题，1 表示命题真，0 表示命题假，则计算机可以用数字逻辑部件实现其各种功能，对数据信息实行逻辑运算。例如一个多数表决器，每个投票者的意见可用一个逻辑变量表示，1 表示赞成，0 表示反对；表决结果也用一个逻辑变量表示，为 1 表示通过，为 0 表示否决。

计算机可以处理的信息类型极其广泛，我们不可能也没有必要穷举所有的例子。通过上面的几个例子，希望读者能举一反三。请注意，善于用约定的数字代码去表示各种需要描述的信息，这是从事计算机技术工作的重要基础。

2. 在物理机制上，数字代码以数字型信号表示

从物理实现这一层次看，数字代码需用电信号去体现，才能用电路部件实现信息的传送与处理。电信号可分为两大类型，即模拟信号及数字信号。

模拟信号是一种在时间上连续的信号，用信号的某些参量去模拟信息，例如用电信号的幅值去模拟数值的大小，所以称为模拟信号或模拟量。当通过传感器将一些非电量转换为电信号时，最初获得的常常是模拟信号，例如受话器将声音转换为电流波，温度传感器将温度转换为相应的电压或电流信号等。需要将模拟信号转换为数字信号，数字计算机才能进行处理。

数字信号是一种在时间上或空间上离散的信号。单个数字信号仅取有限的几种状态，目前常用的是二值逻辑，非 0 即 1。依靠彼此离散的多位信号的组合，能表示广泛的信息，而处理时则可以逐位处理。

例 1-8 用一串脉冲型数字信号表示一个四位数 1101。

数字信号的一种形式是脉冲，如图 1-2 示，它呈脉动形状，随时间的分布不连续。脉冲信号的特点是，信号电平向正方向（或负方向）跳变，并维持较短的时间，然后回到原来的状态。脉冲信号依靠边沿作用，可利用其上升边沿或下降边沿表示某一时刻，起定时作用。也可利用脉冲的有无表示 1 或 0，如在约定时间内有脉冲为 1，无脉冲为 0。如图 1-2 示，通过同一根信号线在一段时间内发出一组脉冲序列，表示数字代码 1101。

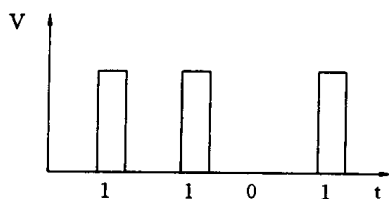


图 1-2 用一串脉冲表示数字代码

例 1-9 用一组电平型数字信号表示四位数 1101。

数字信号的另一种形式是电平信号，利用信号电平的高低表示不同状态值，通常定义高电平（如 +5V）表示 1，低电平（如 0V）表示 0。由于是依据电平的高低起作用，所以电平信号维持某一状态的时间相对要长一些。如图 1-3 示，用一组信号线表示多位数，每一根线产生一位电平信号，各线（位）之间在空间上是独立的，即离散的。事先约定各位的高低顺序，如图表示 1101。

每一位数，每一根线产生一位电平信号，各线（位）之间在空间上是独立的，即离散的。事先约定各位的高低顺序，如图表示 1101。

采用数字化方法表示信息，具有许多重要优点：

(1) 抗干扰能力强，可靠性高。因为每位数字信号只有两种状态：非 0 即 1，即使信号受到一定程度的干扰，仍能比较可靠地鉴别出它的有无或高低。

(2) 依靠较多位数字信号的组合，可以获得很宽的表示范围及很高的精度。理论上位数的增加并无限制，取决于愿意付出的硬件代价。

(3) 在物理上容易实现，并可存储信息。因为每一位只取其两种可能的极端状态，因而可以有多种方法来体现，如：开关的通或断，晶体管的通导或截止，电容上有电荷或无电荷，磁性材料的正向磁化或反向磁化，磁化状态的变或不变、变化相位、变化频率等。相应地可用双稳态触发器存储信息，或利用电容上存储电荷来存储信息。

(4) 可表示信息的类型与范围极其广泛。

(5) 能用逻辑代数等数字逻辑技术进行处理，这就形成了计算机硬件设计的基础。通过处理功能逻辑化这一思想，能用种类非常有限的几种逻辑单元构造出变化无穷的计算机系统及其他数字系统。

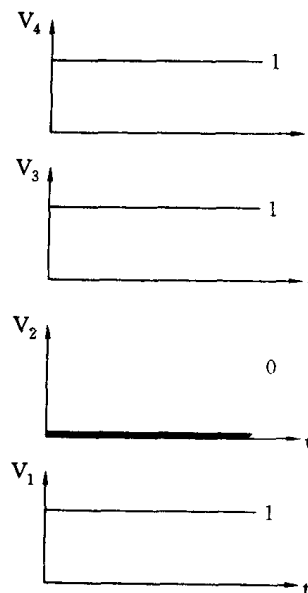


图 1-3 用一组电平信号表示数字代码

三、数字计算机的特点

基于数字化的信息表示方法与存储程序的工作方式，数字计算机具有如下一些重要特点：

1. 能在程序控制下自动连续地工作

由于采用存储程序工作方式，一旦输入所编好的程序，只要给定程序运行条件和起始地址，启动后计算机就能自动连续地执行程序。这是数字计算机的一个基本特点，也是与其他计算工具最本质的区别所在。

2. 运算速度快

目前的计算机采用高速电子线路组成硬件，能以很高的速度工作，这不仅极大地提高了工作效率，还使许多复杂问题得以实际解决。现在普通的微型计算机每秒可执行数十万到数百万次指令，大规模并行处理系统每秒可完成千亿次浮点运算。

3. 运算精度高

如前所述，只要增加位数就能提高运算精度，这在理论上几乎没有限制。现在普通的微型计算机基本运算字长 32 位，在软件上稍加变化便能处理多字长运算，从而获得更高的精度。

4. 具有很强的信息存储能力

数字计算机设置有存储器，因而具有存储信息能力，这是它的又一主要特点，计算机的许多功能与特点也是由此派生的。能存储程序，所以能自动连续工作。存储容量大，可存储的程序多，计算机功能就强，而且可以继续扩充功能。能存储大量数据，使许多信息处理得

以实现。现在一块存储芯片可以存储几百页英文书籍的内容；一台普通的微型计算机，其主存储器可存储数百万个字符，其联机外存储器可存储数亿个字符；至于巨型计算机系统，其存储能力更是高出成百上千倍。

5. 通用性强

基于信息表示数字化，数字计算机能够处理范围极其广泛的各类信息。基于数字信号的特点，可用逻辑代数作为硬件设计的基本数学工具，用数字逻辑部件处理数字信号，即前面所说的“处理功能逻辑化”，这就使计算机具有统一的逻辑基础。在此基础上，计算机不仅能实现算术运算，也能实现逻辑运算；不仅能实现数值运算，也能对各类信息作非数值计算性质的运算处理，如信息检索、图像处理、逻辑判断、知识处理等。因此计算机具有极强的通用性，能应用于各个科学技术领域，并渗透到社会生活的各个方面。

第二节 计算机系统的硬软组成与层次结构

一、硬件系统

计算机的硬件是指系统中可触摸得到的物理设备实体，如运算器、存储器、控制器、输入设备、输出设备等，以及将它们组织为一个计算机系统的体系结构。按照这一概念，一个计算机系统的硬件包括哪些部件，应当具备哪些基本功能；采用何种电路、器件构成，工艺上如何组装；这些部件按什么体系结构连接成有机的整体等，都属于硬件设计的范畴。

在早期的计算机中，如经典诺依曼机所描述，运算器与控制器曾是相对独立的两大部分。现在则常将它们组成一个整体，称为中央处理机 CPU (Central Processing Unit)。在微型计算机中，常将 CPU 集成于一块芯片之中。CPU 是硬件系统的核心，它是执行指令功能的核心部件，控制着数据流与控制信息流的操作。此外，它向计算机系统中的其他部件发出各种控制信息，收集各部件的状态信息，与各部件交换数据信息等。

按照产生微命令的方式，控制器可分为两大类：组合逻辑控制器及微程序控制器。前者是靠若干组合逻辑电路产生微命令序列，将指令代码、状态信息、时序信号等输入到这些组合逻辑电路，电路将分时产生相应的微操作命令。这种控制器又称硬连逻辑控制器，优点是形成微命令的速度快，用于高速计算机；但所用电路较多，且繁琐、不规整、不易修改，硬件代价较高。微程序控制器是将微命令序列以代码形式编制成微程序，存入一个控制存储器中；当 CPU 运行程序时，通过读取并执行微程序，产生微命令序列，解释执行机器指令序列。微程序控制方式比较规整，硬件代价较小，易于扩充功能，但速度稍慢。当指令功能比较丰富，而且对工作速度的要求不是特别高时，广泛采用微程序控制方式。

我们总是希望系统中的存储容量大、存取速度快，但这往往是有矛盾的，为此常将存储器分为几级。可以由 CPU 按地址直接访问的，称为主存储器，由半导体存储器构成，速度较快并具有适当的存储容量。习惯上将 CPU 与主存储器合称为主机，相应地又将主存储器叫做内存储器。作为主存后援的一级，叫后援存储器或称为辅助存储器，由于它位于主机范畴之外，又叫做外存储器，简称为外存。目前多采用磁盘、磁带、光盘构成外存，容量可以很大，但速度低于主存，CPU 不能直接访问。为了提高 CPU 访存速度，许多计算机在 CPU 与主存之间增设一级高速缓存，将当前正在使用的程序与数据区复制于其中，作为主存当前活跃区

的副本。

从信息传送角度看，输入设备与输出设备可视为一类，输入设备将数据输入主机，而输出设备由主机向外输出数据，传送方向不同而已。有一些设备既可以作为输入设备又可以作为输出设备。因此常将输入设备与输出设备合称为 I/O 设备，由于它们在逻辑划分上也是位于主机之外，又称为外围设备或外部设备。外存储器也属于 I/O 设备的范畴。

主机通过一组总线连接各种外围设备，在总线与各种外围设备之间往往有一些起缓冲与连接作用的部件，称为外围接口或 I/O 接口。

下面以系统粗框图（部件、设备级）形式描述一些常见的硬件系统结构形态。

1. 以总线为基础的系统结构

计算机的操作基本上可归结为信息传送，其逻辑结构的关键在于如何实现数据信息的传送，即数据通路结构。现在普遍采用总线结构实现。

总线是一组能为多个部件分时共享的公共信息传送线路。所谓共享，即可通过总线挂接多个部件，它们之间通过这一公共总线传送信息。但同一时刻同一组总线只能接受一个发送源，否则会发生冲突，即分时地接受与发送各部件的信息，换句话说各次传送需要分时占用总线进行。

在微、小型计算机系统中，常以一组系统总线为连接的基础，这组总线包括：数据线、地址线、控制信号线，通过它们连接计算机系统的各组成部件，如 CPU 模块、主存储器模块、各种外围接口，再通过这些接口部件分别连接外围设备。如图 1-4 示。接口相当于外围设备与主机之间的适配器，如果要增加新的外围设备，只需增加相应的适配器，就可往总线上挂接，扩展很方便。

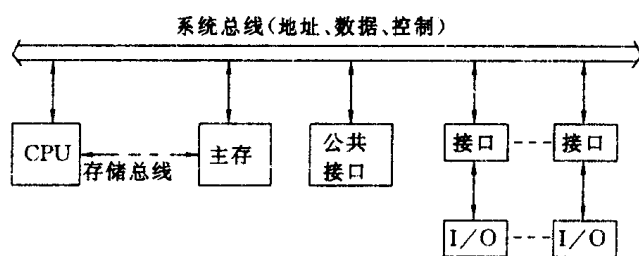


图 1-4 以总线为基础的系统结构

不同的计算机系统，可能采用不同的总线结构、信号组成与工作方式。

例 1-10 PDP-11 是一种典型的 16 位小型计算机系列，其系统总线称为单总线，意为只有一组总线，如图 1-4 结构。CPU 与主存间、CPU 与各外围设备间、外围设备与主存间、各外围设备间，都可通过这组单总线交换信息。PDP-11 对外围设备采取统一编址方式，将接口中有关寄存器与主存储器各单元统一编址，称为总线地址，CPU 可用传送指令，像访问主存一样地访问外围设备接口。

例 1-11 IBM-PC 系列是现在应用最广泛的微型计算机。与 CPU 直接连接的是一组局部总线，它位于 CPU 与主存所在的系统母板上，CPU 通过局部总线访问主存与某些属于主机范畴内的其他部件，这组局部总线连接了一个最小规模的系统。为了连接外围设备，局部总线上的信息经总线控制器与锁存器再与外部相连，形成 I/O 扩展总线，即系统总线，在 PC 机中

将它称为 PC 总线。通过 PC 总线可挂接各种接口适配器，再通过它们连接各种外围设备，从而形成一个较大规模的系统。PC 机一般对外围设备采用单独编址方式，为接口中有关寄存器分配一种 I/O 端口地址，用 I/O 指令按端口地址访问接口，从而实现访问 I/O 设备。

例 1-12 Multibus II 是一种用于 32 位微机的互连总线，它将系统总线分为若干组子总线。其中有一组 32 位并行系统总线 ipsB，用来实现多 CPU 之间的互连。另有一组称为“局部总线扩充”的存储总线 iLBX II，它包含 26 位地址线、32 位数据线，可高速访问存储器。此外还有系统扩充 I/O 总线 iSBX、多通道 DMAI/O 总线等，用于访问 I/O 设备。访问主存速度对于系统的工作速度是至关重要的，因此在速度较高的系统中常在 CPU 与主存间设置专门的高速存储总线。

2. 采用通道或 IOP 的大型系统结构

大、中型计算机系统的构成，更着重于系统功能的扩大与效率的提高。相应地，CPU 内部采取多个运算处理部件，可实现流水处理，甚至可同时执行几条指令；采用多级存储体系，主存储器采取多存储体交叉访问工作方式；外围设备的种类与数量增多，采用通道控制器或输入输出处理机 IOP 管理 I/O 操作；此外，大型机的系统软件庞大。

图 1-5 给出了大、中型计算机的一种典型结构，在系统连接上大致可分为四级：主机、通道、I/O 控制器、I/O 设备。

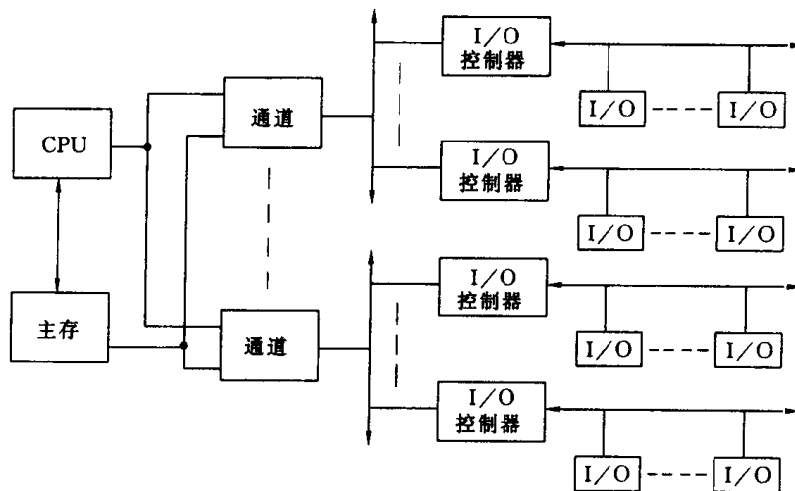


图 1-5 带通道的大中型计算机系统结构

主机范畴的主体是 CPU 与主存储器。由于整个系统的信息传送量大，为了实现高速工作，CPU 与主存之间设置了专门的存储总线，在 CPU 与通道间、主存与通道间也都有独立的数据通路。

通道是一种管理 I/O 操作的控制部件，CPU 在启动通道后可继续执行本身的程序以进行运算处理，由通道执行专门编制的一种通道程序，管理 I/O 操作，使 CPU 中的数据处理与 I/O 操作可以并行地执行，从而提高系统效率。每个通道可以连接一台或数台 I/O 设备控制器，每个 I/O 控制器又可连接一台或数台相同类型的 I/O 设备，如几台键盘、几台打印机或几台磁盘等，这样，整个系统可连接数量可观的多种外围设备。

图 1-5 所示的系统结构具有较大的变化与扩展余地。对较小一些的系统，可将通道与

CPU 合并为一级,即将通道部件纳入 CPU 内,称为结合型通道。对较大一些的系统,可将通道设置为独立的一级,如图所示。对更大的系统,可将通道发展为功能更强的输入输出处理机,简称 IOP。在有些巨型机中,用通用小型计算机作为前端机,负责管理 I/O 操作并可进行预处理。现在通道的概念已进入微型计算机范畴,有集成化的 IOP 芯片负责管理 I/O 操作。

I/O 设备控制器负责对某种设备的控制,可以是独立于设备的控制部件,也可能与 I/O 设备合为一体。

3. 提高系统性能的若干技术

为了提高计算机系统的整体性能,一是提高单 CPU 性能,二是用多个 CPU 构成多机系统。其中一个重要的趋势是增强并行处理能力。

(1) 流水处理。在 CPU 内部设置指令栈、数据栈、多个处理部件,使指令的执行呈流水线方式,使每秒平均执行指令数大为增加。这是利用时间重叠提高并行能力的策略,现已成为高速 CPU 的一种标准技术。

(2) 向量机。将某些适于向量化的任务如矩阵运算等,予以向量化,用向量运算器并行处理。

(3) 精简指令系统计算机 (RISC)。在微型计算机领域出现了 RISC 化的趋势,即只采用比较简单、执行速度快、使用频繁的指令,并采取一系列配套技术、使单机运算速度得以提高。

(4) 超标量方式。CPU 可同时执行数条指令,如 2~4 条。这已成为高速微处理器的一种标准技术。

(5) 超长指令字 (VLIW) 方式。一条超长指令相当于多条常规指令,以提高并行度。

(6) 阵列机。这是并行处理机的又一种模式,系统中有多个处理部件,组成一个阵列,在统一的控制器管理下,并行地处理各自的数据,合作实现一条复杂指令。一般称为单指令流多数据流方式。

(7) 多存储体交叉访问技术。每一条指令的执行都要从访存读取指令开始,然后根据指令要求读取操作数,很可能还需将运算结果存入存储器。因此存储器速度是影响整机速度的关键,在当前技术条件下,存储器速度的提高跟不上 CPU 速度的提高,约相差一个数量级,被称为诺依曼瓶颈。因此在高速系统中广泛采用多存储体技术,CPU 可以不等一个存储体的存取结束,就可交叉地访问其他存储体,从而使整体的访存速度提高。这样的主存称为并行主存系统。

(8) 多机系统。在同一技术条件下,单个 CPU 的功能与速度毕竟有限,因此计算机系统结构的一种发展方向是多处理器化,即用多个 CPU 构成多机系统。由于大规模集成技术的发展,CPU 可集成于一块很小的芯片之中,而多机系统的连接模式灵活多样,多机系统成为一种很有前途的模式,甚至有人主张取消传统的微、小、大型机的划分,代之以 CPU 数来标志系统规模的大小。

二、软件系统

计算机软件通常泛指各类程序和文件,由于它们在计算机中的表示是一些不能直接触摸到的二进制信息,所以称为软件。软件的实体主要表现为程序,说明如何完成某个任务的指令序列,就是某种算法的程序体现。因此有人主张采取一种简单的定义:软件即程序。而有

些人则主张将软件的含义描述得更广泛一些，把编制程序、维护与运行程序所需的文件也纳入软件范畴。

在计算机系统中，各种软件的有机组合构成了软件系统。从软件配置与功能的角度进行分类，可将软件分为两大类：系统软件、应用软件。

1. 系统软件

这是一组为使计算机系统良好运行而编制的基础软件，其功能是负责系统的调度管理，提供程序的运行环境和开发环境，并且向用户提供各种服务。从软件配置的角度，它是作为系统资源即软设备提供给用户使用，是用户所使用的计算机系统的一部分。如：

(1) 操作系统

操作系统是软件系统的核心。它是负责管理和控制计算机系统硬、软资源与运行程序的系统软件，是用户和计算机之间的接口，提供了软件的开发环境和运行环境。

操作系统的典型组成包括五大模块：作业调度、处理机调度、存储管理、文件系统、外围设备管理。用户往往是通过操作系统提供的各种功能来操作使用计算机，例如键盘命令解释与执行等，所以我们说它是用户和计算机之间的接口。操作系统的几大功能模块，以尽可能优化的方式来调度管理系统的硬、软资源，合理地组织工作流程，提高系统的工作效率。当我们开发某个软件时，需要调用在操作系统管理下的有关资源，以获取操作系统的最大支持。所编制的软件又往往作为在操作系统管理下的文件，运行时也可能需要利用操作系统的有关功能，调用操作系统管理下的其他资源。所以我们说操作系统提供了软件的开发环境与运行环境。

(2) 语言处理程序（解释、编译）

用户往往是用某一种程序设计语言来编写程序，这样的程序称为源程序。但计算机硬件所能直接识别的是数字代码化的机器语言，即用数字代码表示的指令序列。这就需要通过语言处理程序进行转换，将源程序变化为可执行的指令代码。有两种基本的转换方式：

* 解释方式

针对某种程序设计语言，事先编制解释程序。使用时计算机执行解释程序，将源程序分段地转换为相同功能的指令序列，并执行这段等价的指令序列，即边解释边执行。

* 编译方式

针对某种程序设计语言，事先编制编译程序。使用时计算机先执行编译程序，将源程序完整地翻译为机器语言（代码化）的目标程序，即可由硬件执行的指令序列。然后执行目标程序，而源程序则不再需要。

这些解释程序与编译程序，泛称为语言处理程序。显然，不同的CPU、不同的程序设计语言，需要不同的语言处理程序。

(3) 数据库管理系统

在计算机中，对数据的管理极为重要，特别是在计算机信息管理系统中更为突出，为此出现了数据库技术。数据库是在计算机存储设备上合理存放的、相互关联的数据的集合，能提供给不同用户，共享使用，独立维护。数据库的应用属于编制应用软件范畴，但相应地需事先在计算机系统中配置数据库管理系统软件，负责装配数据、更新内容、查询检索、通信控制及对用数据库语言编写的程序进行翻译、控制有关的运行操作等。

(4) 各种服务性支撑软件

服务性程序是指：为帮助用户使用与维护计算机，提供服务性手段而编制的一类程序。这类程序包含的内容很广泛，一般指输入与装配程序、编辑工具、调试工具、诊断程序、窗口软件、提示系统，以及一些可供调用的通用性应用软件如文字处理软件、表格处理软件、图形处理软件等。为了使用户能更好地操作计算机，现在趋向于为用户提供一种集成环境，这种环境具有良好的人机界面，能提供完善的服务支持。

(5) 各种标准程序库

系统中可事先配置一些通用的、优化的标准子程序，供用户调用。例如许多编译程序中就含有库文件，用户在用高级程序设计语言编写源程序时，只要给出调用名即可，编译时可将有关的库文件调出装配。

2. 应用软件

应用软件是指用户在各自应用领域中，为解决各类问题而编写的程序，也就是直接面向用户需要的一类软件。由于计算机的应用领域极其广泛，这类应用软件可以说是不胜枚举，如：

- (1) 科学计算类程序
- (2) 工程设计类程序
- (3) 数据处理类程序
- (4) 信息管理类程序
- (5) 自动控制类程序
- (6) 情报检索类程序

当然，对系统程序与应用程序的划分并不是一成不变的，一些具有通用价值的应用程序也可以纳入系统程序之中，作为一种资源提供给用户。

三、层次结构模型

如上所述，计算机系统以硬件为基础，通过配置各种软件，形成一个有机组合的系统。我们常采用一种层次结构观点去进行分析或设计，也就是从不同角度将计算机系统分为若干级（层次）。在使用计算机时，根据需要选择某一层次，去观察分析计算机的组成、性能和工作机理，或进行该层次的设计工作。在构造一个完整的系统时，可以分层次地逐级实现，按这种层次结构化设计策略实现的系统，易于建造、调试、维护和扩充。

有多种划分层次的方法，本节将列举两种常见的层次结构模型。

1. 硬软组成的层次结构

图 1-6 所示的层次结构模型，表明了一个计算机系统中由哪些硬件与软件组成，以及它们之间的关系。自下而上，反映了计算机系统的逐级生成过程。

(1) 拟定指令系统，它是 CPU 硬软界面所在。

(2) 制作能实现指令功能的硬件系统，常称为硬核。许多计算机采用微程序控制方式，相应地将硬件部分分为二级。最底层是硬连逻辑部件，即用连线连接的各种逻辑部件。上面一层是微程序控制器，它解释并执行指令，发出微操作命令控制逻辑部件工作。

(3) 配置操作系统，这是系统软件的核心和基础。

(4) 配置所需的语言处理程序及各种软件资源，将它们置于操作系统的管理调度之下。

(5) 输入用户程序，予以处理执行。