



高等学校电子信息类专业规划教材

计算机导论

祁亨年 主 编

方陆明 吕海涛 副主编



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



北方交通大学出版社
<http://press.njtu.edu.cn>



21 世纪高等学校电子信息类专业规划教材

计算机导论

祁亨年 主编

方陆明 吕海涛 副主编

清华大学出版社
北方交通大学出版社

• 北京 •

内 容 简 介

本书从直观的图灵机模型出发,分析了现代计算机设计的基本思想,并以通俗易懂的语言从计算机系统结构、计算机软件、数据组织和计算机安全四个方面论述计算机专业知识。全书共安排 10 章内容,涉及了计算机专业相关的主要课程的重要知识点,每一章节内容都紧密衔接,每一部分都自成体系。全书内容主要包括:图灵机模型、二进制位的存储及数据表示、压缩和差错控制,计算机硬件结构、指令系统、流水线、存储体系和 I/O 系统、多处理器系统;操作系统、算法、程序设计语言和软件工程;数据结构、文件和数据库,以及计算机系统安全所涉及的密码学、病毒、网络安全和通信安全等内容。通过学习,不仅可以掌握计算机最基本的知识点,更重要的是可以从总体上把握各门专业课和专业基础课的关系,有助于学生明确进一步学习的目标,为后续专业课程的学习做好准备。为了提高学习效果,每章都提供了关键技术、思考和练习题。

本书可作为高等学校计算机专业、信息技术专业和其他相关专业计算机引论或计算机科学导论课程的教材或教学参考书。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

计算机导论 / 祁亨年主编. —北京:北方交通大学出版社, 2003. 10

(21 世纪高等学校电子信息类专业规划教材)

ISBN 7-81082-197-0

I. 计… II. 祁… III. 电子计算机—高等学校—教材 IV. TP3

中国版本图书馆 CIP 数据核字(2003)第 077203 号

责任编辑:陈 川

印 刷 者:北京鑫海金澳胶印有限公司

出版发行:北方交通大学出版社

邮编:100044

电话:010-51686045, 62237564

清 华 大 学 出 版 社

邮编:100084

经 销:各地新华书店

开 本:787×1092 1/16 印张:16.25 字数:392 千字

版 次:2003 年 10 月第 1 版 2003 年 10 月第 1 次印刷

印 数:8000 册

定价:23.00 元

前 言

近年来,各高校对于计算机专业教学大纲的看法逐步趋向一致,陆续开设了计算机引论或相似的课程。但作为导引性课程,其教学内容究竟应该着重于什么方面,却还是仁者见仁,智者见智,争论不休。这种争论的结果产生了两种截然不同的观点,表现在教材的编写上也采用两种不同的模式:一类是对计算机专业知识的概述,试图总揽全局;另一类是从方法论入手,对计算学科的主要内容进行系统化、形式化的概括。笔者认为,这两种模式各有优劣,彼此不可全盘否定,根据因材施教的一般教学规律,要看教学的具体对象而定。对专业知识基本不了解且抽象思维训练较少的学生来说,第一类教材比较适合,因为讲授的知识比较具体、形象,容易接受;而对于具备一定专业知识,抽象思维能力较强的学生来说,第二类教材将为进一步深入地学习专业知识提供方法论的指导,并进行形式化方法的训练,这对于深入理解和掌握计算机专业知识体系将有很大帮助。

综上所述,对普通高校刚入学的计算机专业或相关专业的学生来说,计算机引论课程的教材还是选择第一类教材比较适宜。但很遗憾,现在这类教材还比较匮乏,仅有的几本,基本上都是20世纪90年代的老教材,知识比较陈旧,有的甚至于偏向了计算机文化基础方向,对计算机专业学习基本没有帮助。受此影响,一些学校甚至就将计算机引论课程当成计算机文化基础课上,这是很不合适的。实际教学的要求,迫切需要一本适合的教材,因此,我们决定组织编写本书。

本书力图以通俗易懂的语言阐述计算机专业知识,全书除绪论外,分为计算机系统结构、计算机软件、数据组织和计算机安全四个部分共10章,从计算学科最核心理论——图灵机模型入手,在形象直观地论述该模型后自然地展开数据编码、系统结构等内容,全书涉及了计算机专业几乎所有主要专业课程和专业基础课程的内容,使学生通过学习本课程后能对计算机专业有个整体的认识,并掌握各专业课程的内在关系,明确进一步学习的目标。

本书绪论部分及第1、2章由祁亨年编写,第3章由吕海涛编写,第4、5、7章由姜真杰编写,第8、9章由方陆明编写,第6、10章由汪杭军编写。浙江大学计算机学院博士生导师杨建刚教授对全书进行了审读,并提出了很好的建议,邓超、蔺杰和陈倩三位博士参与了书稿的审校工作,最后由祁亨年负责统编定稿。

本书的编写得到了浙江林学院教务处石道金处长、信息工程学院寿韬书记和方陆明院长的关心和支持,并获得了学校教材建设基金的资助,方陆明院长本人还亲自参与了本书的编写工作并联系出版事宜,教务处教务科王亚伟科长也对该教材的出版提供了帮助。编者在此对他们的帮助和支持表示衷心的感谢!

由于2003级新生教学需要,本书从2002年底开始组织编写,历时短短半年,期间,每位编者都承担着大量的其他教学和科研任务,时间非常仓促,因此,书中的谬误和不完善之处在所难免,敬请各位读者提出宝贵意见和建议,以便在随后的修订版中能一并改正。

编 者

2003年9月

目 录

绪论	(1)
0.1 计算机的产生	(1)
0.1.1 算法 (Algorithm)	(2)
0.1.2 计算机的产生	(2)
0.2 计算机的发展	(6)
0.3 计算机科学与技术学科的构成	(10)
0.3.1 应用层	(10)
0.3.2 专业基础层	(10)
0.3.3 专业基础的理论基础层	(11)

第 1 篇 计算机系统结构

第 1 章 图灵机模型及数据编码	(15)
1.1 图灵机	(16)
1.1.1 图灵机的直观描述	(16)
1.1.2 图灵机的形式化描述	(16)
1.1.3 计算“ $x+1$ ”的图灵机	(17)
1.1.4 通用图灵机	(20)
1.2 位的存储	(22)
1.2.1 逻辑运算和门	(22)
1.2.2 触发器	(23)
1.2.3 其他存储技术	(23)
1.3 存储器	(24)
1.3.1 主存储器	(24)
1.3.2 辅助存储器	(25)
1.4 数据在计算机中的表示	(25)
1.4.1 二进制	(25)
1.4.2 数值的表示	(27)
1.4.3 字符的表示	(31)
1.4.4 图形和图像的表示	(34)
1.4.5 音频数据的表示	(35)
1.5 数据压缩	(36)
1.5.1 无损压缩	(36)

1.5.2 有损压缩	(39)
1.6 数据传输误码及对策	(40)
1.6.1 奇偶校验	(40)
1.6.2 纠错码	(41)
第2章 计算机结构	(45)
2.1 中央处理器	(47)
2.1.1 运算器	(48)
2.1.2 控制器	(48)
2.1.3 寄存器	(49)
2.2 指令系统	(50)
2.2.1 指令系统的概念	(50)
2.2.2 CISC 和 RISC	(51)
2.2.3 指令系统的设计要求	(52)
2.2.4 指令格式	(52)
2.2.5 指令字长度	(54)
2.2.6 寻址方式	(54)
2.3 指令执行	(57)
2.3.1 非访存指令的指令周期	(57)
2.3.2 直接访存指令的指令周期	(60)
2.3.3 间接访存指令的指令周期	(61)
2.3.4 程序控制指令的指令周期	(63)
2.4 流水线技术	(65)
2.4.1 什么是流水线	(65)
2.4.2 理想流水线性能分析	(65)
2.4.3 流水操作中的主要障碍	(66)
2.5 存储系统	(67)
2.5.1 存储器的层次结构	(67)
2.5.2 存储体系涉及的问题	(68)
2.5.3 高速缓冲存储器 (Cache)	(69)
2.5.4 主存储器的设计	(73)
2.5.5 虚拟存储器	(74)
2.6 外围设备	(76)
2.6.1 交互设备	(76)
2.6.2 存储设备	(81)
2.6.3 数据通信设备	(84)
2.7 输入输出方式	(85)
2.7.1 程序查询方式	(86)

2.7.2 中断方式	(86)
2.7.3 直接内存访问方式	(88)
2.8 多处理器	(89)
2.8.1 并行系统结构的分类	(89)
2.8.2 并行处理的困难	(91)

第 2 篇 计算机软件

第 3 章 操作系统	(97)
3.1 操作系统的发展	(98)
3.1.1 串行处理	(98)
3.1.2 简单批处理	(98)
3.1.3 多道批处理	(99)
3.1.4 分时系统	(101)
3.1.5 实时系统	(102)
3.2 操作系统的结构	(102)
3.2.1 操作系统的功能	(102)
3.2.2 操作系统的组成	(103)
3.2.3 系统的引导	(105)
3.3 现代操作系统处理的难题	(106)
3.3.1 进程	(106)
3.3.2 有效的分配处理器	(108)
3.4 提供充足和高效的存储器	(109)
3.4.1 存储器的有效利用	(109)
3.4.2 存储器的拓展: 虚拟存储器	(112)
3.4.3 I/O 设备: 高效并通用	(114)
3.4.4 并发控制的难题	(114)
3.5 网络操作系统和分布式操作系统	(116)
3.5.1 网络操作系统	(116)
3.5.2 分布式操作系统	(120)
第 4 章 算法	(123)
4.1 算法的概念	(125)
4.1.1 定义	(125)
4.1.2 性质	(126)
4.2 算法的表示	(126)
4.3 常用算法介绍	(129)
4.3.1 递归	(129)

4.3.2 迭代	(131)
4.3.3 分治法	(132)
4.4 并行算法	(133)
4.5 算法的评价	(135)
4.5.1 算法的时间复杂度	(135)
4.5.2 算法的空间复杂度	(136)
4.5.3 算法的易理解性	(137)
4.6 算法设计的要求	(137)
第5章 程序设计语言	(139)
5.1 程序设计概念	(140)
5.2 程序设计语言的发展	(141)
5.2.1 计算机语言的发展历程	(141)
5.2.2 组件技术	(143)
5.3 程序设计语言中的一些术语	(144)
5.3.1 汇编语言	(144)
5.3.2 过程性语言	(145)
5.3.3 说明性语言	(145)
5.3.4 脚本语言	(145)
5.3.5 高级语言	(147)
5.3.6 编译性语言	(147)
5.3.7 解释性语言	(147)
5.3.8 面向对象的程序设计语言 (OOPL)	(147)
5.4 结构化程序设计方法	(147)
5.5 面向对象的程序设计方法	(150)
5.5.1 对象	(150)
5.5.2 类	(150)
5.5.3 封装性	(151)
5.5.4 继承性	(151)
5.5.5 多态性	(151)
5.6 当前流行语言介绍	(152)
5.6.1 BASIC	(152)
5.6.2 C#	(152)
5.6.3 Delphi	(152)
5.6.4 Java	(152)
5.6.5 PowerBuilder	(152)
5.6.6 ASP.NET	(153)
5.7 程序设计语言的发展趋势	(153)

第 6 章 软件工程	(155)
6.1 软件工程与软件过程	(156)
6.1.1 软件危机	(156)
6.1.2 软件工程	(157)
6.1.3 软件过程	(157)
6.2 传统方法学	(158)
6.2.1 结构化分析	(158)
6.2.2 结构化设计	(158)
6.2.3 结构化实现	(159)
6.3 面向对象方法学	(159)
6.3.1 面向对象分析	(160)
6.3.2 面向对象设计	(160)
6.3.3 面向对象实现	(160)
6.3.4 标准建模语言 UML	(161)
6.4 软件项目管理	(165)
6.4.1 计划	(165)
6.4.2 组织	(166)
6.4.3 控制	(166)
6.4.4 国际标准	(167)

第 3 篇 数据组织

第 7 章 数据结构	(171)
7.1 数据结构的概念	(172)
7.2 几种典型的数据结构	(174)
7.2.1 线性表	(174)
7.2.2 堆栈	(175)
7.2.3 队列	(175)
7.2.4 树	(176)
7.2.5 图	(178)
7.3 查找	(180)
7.3.1 基本概念与术语	(180)
7.3.2 查找的方法	(181)
7.4 排序	(182)
7.4.1 排序分类	(182)
第 8 章 文件	(185)
8.1 文件简介	(186)

8.1.1 文件的含义	(186)
8.1.2 文件及其组成	(186)
8.1.3 文件目录结构与文件夹	(188)
8.2 文件系统分类	(188)
8.2.1 FAT16 文件系统	(189)
8.2.2 FAT32 文件系统	(190)
8.2.3 NTFS 文件系统	(190)
8.2.4 HPFS 文件系统	(191)
8.2.5 EXT2 文件系统	(191)
8.3 文件组织和访问	(192)
8.3.1 堆文件	(192)
8.3.2 顺序文件	(194)
8.3.3 索引顺序文件	(194)
8.3.4 索引文件	(195)
8.3.5 直接文件	(195)
第 9 章 数据库	(197)
9.1 数据库概述	(198)
9.1.1 数据库发展史	(198)
9.1.2 数据库系统的概念	(198)
9.1.3 数据库的研究内容	(202)
9.1.4 数据库研究的难点与展望	(204)
9.2 关系型数据库	(205)
9.2.1 关系模型的基本概念	(205)
9.2.2 关系模型的三类完整性规则	(207)
9.2.3 关系模型的形式定义	(208)
9.2.4 关系代数	(208)
9.2.5 关系数据库语言 and 数据库的建立	(210)
9.3 面向对象的数据库	(214)
9.3.1 面向对象的基本概念	(214)
9.3.2 面向对象的数据库系统	(214)

第 4 篇 计算机安全

第 10 章 计算机安全	(219)
10.1 计算机安全的研究内容和发展过程	(220)
10.1.1 研究内容	(220)
10.1.2 发展过程	(220)

10.2 计算机密码学	(221)
10.2.1 基本概念	(221)
10.2.2 密码编码学	(223)
10.2.3 密码分析学	(226)
10.3 计算机病毒	(226)
10.3.1 计算机病毒概述	(226)
10.3.2 病毒分类	(227)
10.3.3 病毒的检测	(228)
10.3.4 病毒的防护	(229)
10.4 计算机网络安全	(230)
10.4.1 网络的特点及安全问题	(230)
10.4.2 网络的数据加密	(231)
10.4.3 防火墙技术	(232)
10.4.4 虚拟专网技术	(232)
10.4.5 入侵检测	(232)
10.4.6 了解黑客	(233)
10.5 通信安全	(233)
10.5.1 通信基本概念	(234)
10.5.2 搭线窃听	(234)
10.5.3 通信安全技术	(234)
附录 A 参考实验目录	(236)
附录 B 计算机发展大事记	(238)
参考文献	(245)

绪 论

计算机从产生到现在，也不过短短几十年时间，却已经极大地影响并将更广泛而深远地影响和改变人类社会的生活。计算机科学的发展日新月异，计算机科学与技术学科已经发展为一个口径很宽、有多研究方向的学科，成为诸如计算机设计、程序设计与实现、信息处理、算法设计与实现等学科的基础。

0.1 计算机的产生

现在人们公认的世界上第一台电子计算机 ENIAC (Electronic Numerical Integrator And Calculator, 电子数字积分计算机, 见图 0.1) 是 1946 年美国宾西法尼亚大学研制成功的。而早在 20 世纪初, 数学家们就提出了假想的抽象计算机 (也称之为模型机), 当时并不叫计算机而称之为自动机, 并在此基础上进一步提出功能更强的自动机如图灵机 (Turing Machine, TM)。正是这种抽象计算机理论的深入研究, 促进了现代计算机的产生和发展, 而且, 它们至今仍在计算机理论研究方面起着不可替代的作用。但若追溯人们研究执行算法任务的机器的历史, 那将更为久远。

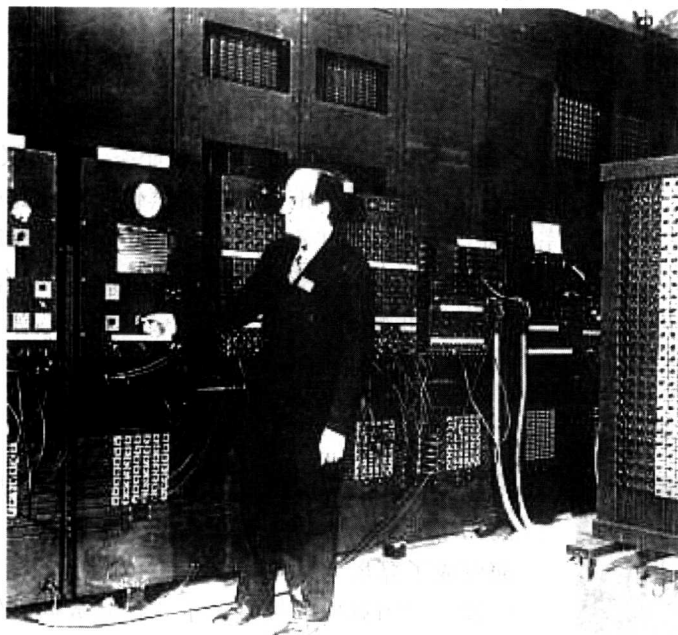


图 0.1 世界上第一台电子计算机 ENIAC

0.1.1 算法 (Algorithm)

通俗地讲，一个算法就是完成一个任务的步骤。例如：

任务：将 3、74、23、89、22、99、65、109、55、45 十个数按从小到大的顺序排列

结果：3、22、23、45、55、65、74、89、99、109

对此任务读者计算非常熟练，可能感觉不到需要按什么特定的步骤进行，而实际上对于一个计算速度较慢的小学生，或者当需要排序的数很多的情况下，必须制定一个规则，按特定的步骤进行。比如，可每次找出未排序的 n 个数中的最小的数，并将其排列在这些数的最前面，接着再同样排列其余 $n-1$ 个数，具体步骤如下：

Step 1 取未排序的数中的第 1 个数，假设它是其中最小的数。

Step 2 将它依次与其后的数相比较，若发现某个数更小，则目前为止发现的最小数是该数，并假设它是最小数，重复 Step 2 直到比较到最后一个数，此时，假设的最小数就是未排序的数中的最小数。

Step 3 将该最小数与未排序的数中的第 1 个数交换位置。回到 Step 1，直到所有数都排序。

执行该步骤 Step 1~3，产生以下的排序过程：

第 1 遍：3、74、23、89、22、99、65、109、55、45

第 2 遍：3、22、23、89、74、99、65、109、55、45

第 3 遍：3、22、23、89、74、99、65、109、55、45

第 4 遍：3、22、23、45、74、99、65、109、55、89

第 5 遍：3、22、23、45、55、99、65、109、74、89

第 6 遍：3、22、23、45、55、65、99、109、74、89

第 7 遍：3、22、23、45、55、65、74、109、99、89

第 8 遍：3、22、23、45、55、65、74、89、99、109

第 9 遍：3、22、23、45、55、65、74、89、99、109

经过共 9 轮的比较选择，产生了最终的排序结果，完成任务。则步骤 Step 1~3 就是完成该排序任务的一个算法。显然，还可以用其他的规则制定其他的步骤完成该任务，从而定义其他的算法。也就是说，完成同一任务的算法不是惟一的。实际上，排序问题是算法学中很重要的一类问题，刚才定义的算法称为选择排序法，其他的排序算法还有诸如交换排序法、插入排序法等。不同算法的效率不同，可在诸多算法中选择最优算法。关于算法将在第四章进一步介绍。

0.1.2 计算机的产生

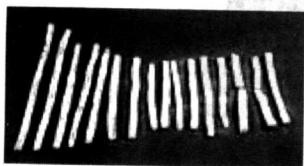


图 0.2 西汉年间的金属算筹

最早的计算设备之一是由中国人发明的算盘。公元前 3000 年，中国人就发明了算筹（见图 0.2）。后来，算筹逐渐演变成了算盘（见图 0.3），至 17 世纪，算盘已在世界上广为传播。它用算珠的位置表示和存储数据，但它不能控制算法的执行，控制算法执行的是人。因此，算盘本身仅仅是一个

据存储装置，有了人的参与才构成完整的计算机（Computational Machine，称为计算机，以区别于今天的计算机—Computer）系统。在电子计算机出现以前，计算机的发展是一个漫长的渐进的过程，每次细微的改进可能谁也不能确切考证。此处从思想的演进连续性角度，将期间较重大的演进过程进行阐述，并于书末列出计算机发展大事记的附录。

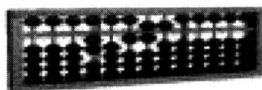


图 0.3 算盘

1620~1630 年，英国人发明了计算尺。至 17 世纪，钟表业的发展特别是齿轮传动装置技术的发展为机械传动装置计算机的产生提供了重要的技术基础。

1623 年，德国久宾根大学的威尔海姆·什卡尔教授设计了第一个带有进位机构执行四则运算的计算设备的模型。该模型的设计思想和以后的机械式计算机的设计思想有类似之处，在此稍做描述。

什卡尔机由三部分组成：进行加减法的加法器、乘法器和记录中间结果的机械。每根轴上都有一个十个齿的齿轮和只有一个齿的辅助轮，辅助轮的作用是把进位送到下一位：当前一个齿轮转了一周后，使下一个齿轮转 $1/10$ 。减法时可以反方向转齿轮。计算加法或减法时，只要排好相加或相减数，然后就可以读结果。除法是重复地从被除数中减除数。乘法按下列方式进行：什卡尔机的结构如图 0.4 (a) 所示，在并行的六个轴上绕上乘法表，每个表的展开形式如图 0.4 (b) 所示，在卷有表格的轴前安上一块有九排窗的板，每排六个窗，和卷着的乘法表数目相同。窗口可由把手控制打开和关闭。假定需要做“387 乘以 27”，所有表格都用把手旋转，使得最上一排窗口出现 387 (000387)。打开第 7 排窗口就可以读到 387 和 7 相乘的结果 2709，如图 0.4 (c) 所示。同样打开第二排窗口得到 387 乘以 2 的结果 774，然后用加法器将 2709 和 7740 相加就得到最终结果 10449。但很遗憾，该模型并未广泛流传，因此，人们一般都以为第一台计算器是由布莱斯·帕斯卡发明的。

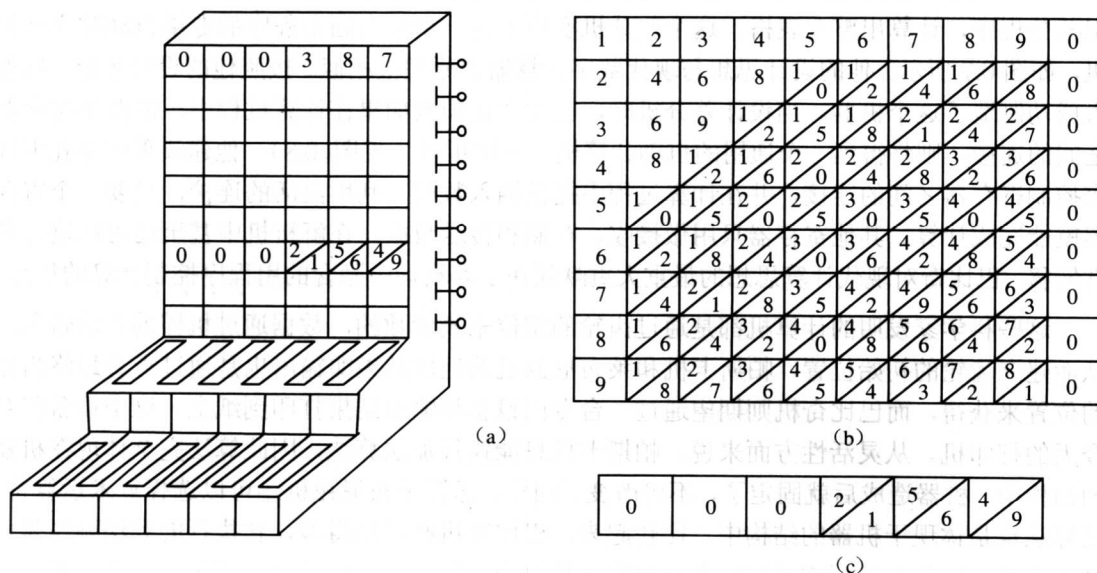


图 0.4 什卡尔机模型

1642 年, 法国哲学家、数学家和物理科学家布莱斯·帕斯卡 (Blaise Pascal, 1623—1662) 发明了加法机, 如图 0.5 所示。这种计算机在今天看来很简单, 它的设计思想比较符合人们的思维习惯。1649 年, 帕斯卡为自己的演算机取得皇家专利权, 证书中谈到: “主要的发明和重大的进展在于: 某一位的小轮或轴完成了十个数字的转动, 才强使下一个只转动一个数字”。为纪念这位伟大的科学家对计算机科学的贡献, 由语言大师沃斯 (N.Wirth) 设计的一种高级程序设计语言命名为 Pascal。

德国哲学家和自然科学家戈特弗里德·威廉·莱布尼兹 (Gottfried Wilhelm Leibniz, 1646—1716) 在帕斯卡的思想和工作的影响下, 改进了计算机的设计思想, 并于 1672 年成功设计完成了改进的计算机, 如图 0.6 所示。它是第一台不仅能加减而且能乘除的演算机, 机器的关键部件是个梯形轴, 即齿长不同的圆柱。莱布尼兹的这一发明第一次实现了带有可变齿数的齿轮, 正是这样的数字轮保证了完成乘除法运算。

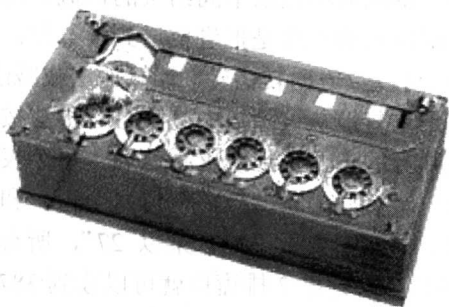


图 0.5 帕斯卡的加法机

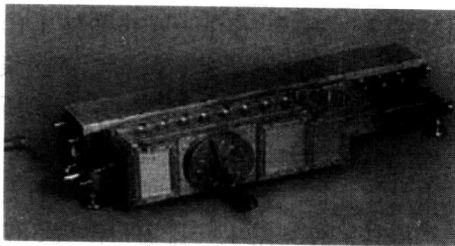


图 0.6 莱布尼兹的演算机

英国数学家查尔斯·巴比奇 (Charles Babbage, 1792—1871) 的工作则更进一步, 他设计了一个差分机来制造航海表。他的思想是先有少量手算的非常精确的值, 然后使用一个有限差分程序, 计算出整个表格。这个差分机实际上是一个带有固定程序的专用自动数字计算机, 如图 0.7 所示。他的设计思想与现代数字计算器的原理很相似。按照他的设计思想, 其他人成功制造了这个机器。当设计差分机时, 巴比奇认识到如果在计算过程中, 能改变寄存器之间的连接, 则将得到一个通用的自动计算机——解析机 (见图 0.8)。他建议采用穿孔卡片来控制寄存器之间的连接, 并在计算过程中提供输入数据。利用灵活的连接, 只要一个寄存器就能完成计算。其他寄存器只用做接收、存储和传送数据。在解析机中甚至还可以进行条件转移。巴比奇对现代计算思想的最重大贡献就在于其设计中隐含的用程序控制计算的思想。

这些科学家发明的计算机都是通过齿轮的定位来表示数的, 数据通过机械的方法输入, 从而建立齿轮的初始位置, 帕斯卡机和莱布尼兹机的处理结果即输出也是通过查看最终齿轮的位置来获得, 而巴比奇机则期望通过一台专门设备将输出结果打印到纸上, 这个设备就是今天的打印机。从灵活性方面来说, 帕斯卡机只能执行加法算法, 因此算法完全体现在机器的结构中, 机器造成后就固定了, 不可改变。同样, 尽管莱布尼兹机还可以选择算术运算符, 其算法也是体现于机器的结构中。比较起来, 巴比奇机则灵活得多, 它执行的算法顺序是可以改变的, 操作人员可通过穿孔卡片来和计算机交换信息, 改变算法。因此, 巴比奇机是可

编程的。巴比奇的助手艾达（Ada, 1815—1852）就是公认的世界第一位程序员，美国军方使用的一门编程语言 ADA 也是以她的名字命名的。

如果追溯用穿孔卡片与机器交换算法的思想的起源，并不是由巴比奇最先提出的，而是借鉴了穿孔卡片式织布机的技术。早在 1801 年，约瑟夫·杰卡德（Joseph Jacquard, 1752—1834, 法国）在她发明的第一台能织出复杂图案的自动织布机上采用了类似的技术，在这台织布机上，织布过程的每一步都是由穿孔卡片上穿孔分布的不同样式来决定的。通过这种方式，可以很容易改变织布机所执行的“算法”，从而织出不同的图案。这时，通过穿孔卡片就向机器表达和传递了“算法”信息。后来，赫尔曼·霍勒瑞斯（Herman Hollerith, 1860—1929）根据类似的思想，发明了能够在穿孔卡片上储存和再现信息的系统（1880 年），并创建了后来发展为计算机界蓝色巨人的 IBM 的公司（1924 年）。

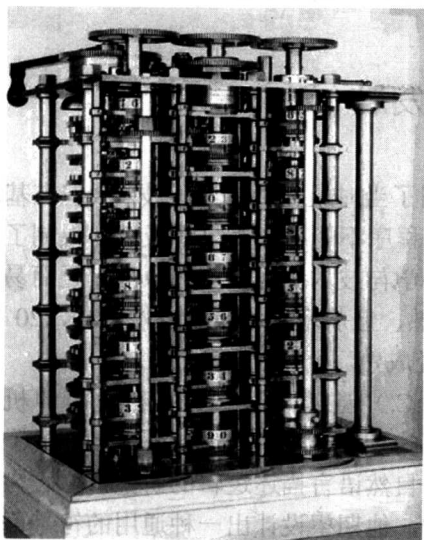


图 0.7 巴比奇差分机

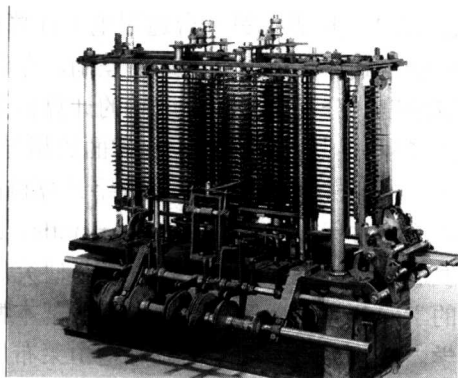


图 0.8 巴比奇解析机

由于准确性很差，齿轮驱动的计算机并未推广，直到有了电动的机械设备。1940 年贝尔实验室完成了采用延迟线的继电器计算机 Model-1，1944 年霍华德·艾肯（Howard Aiken）和一群 IBM 公司的工程师在哈佛大学研制出了机电式自动顺序控制计算机 MARK I（见图 0.9），它也是第一台自动通用数字计算机，其中用了大量的继电器。因当时电子技术发展迅猛，所以 MARK I 计算机几乎在制成的同时就过时了，取而代之的是电子真空管技术，当时其他的研究者已经在采用电子真空管技术研制电子数字计算机了。第一台电子真空管的电子数字计算机是 1937~1941 年在爱荷华州立学院（现在的爱荷华州立大学）建成的 Atanasoff-Berry 计算机。在第二次世界大战快结束时，英国研制了用于破译德国电码的计算机 COLOSSUS，该计算机装有 2500 只真空管。随后，在美国宾夕法尼亚大学莫尔研究所（Moore Institute of University of Pennsylvania），莫克利（J.W. Mauchly）、艾克特（W.J. Eckert）等人研制成功了功能更加灵活的计算机 ENIAC，这就是大家所公认的世界上的第一台采用电

子线路技术的通用电子计算机,该机整机大约有 18000 只真空管,其程序设计是靠部件间转接插线,解算完一个问题转去解算另一个问题时,要拔去所有转接线,再按新的排列顺序重新插入。

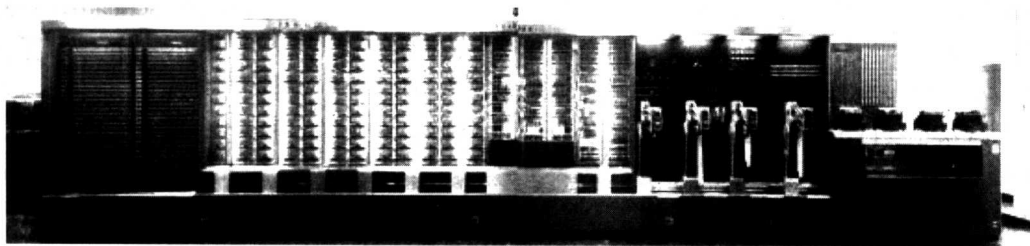


图 0.9 哈佛大学研制的机电式自动顺序控制计算机

0.2 计算机的发展

ENIAC 是世界上第一台通用电子计算机,它采用了当时最先进的电子技术,但在基本结构上与机电式计算机没有本质的差别,存储容量小,程序不能存储等缺陷极大地限制了机器的运算速度。分析随后改进成功的计算机及计算机科学与技术迅猛发展的原因,显而易见,ENIAC 的不足是因为结构设计方面的指导思想不成熟,但从理论和技术条件上看,20 世纪 30~40 年代制造现代通用电子数字计算机的条件已经成熟。

古希腊时代,亚里斯多德(Aristotle, 384—322B.C.)创立了逻辑学说,其对计算机的产生和发展起着很重要作用。逻辑学是研究思维和论证的科学,直观地说,就是研究关于人类推理的学问。亚里斯多德的逻辑学的基本特点是使用自然语言描述逻辑的研究,称之为古典逻辑学。现代逻辑学的最初设想是由莱布尼兹提出的,他期望设计出一种通用的符号语言和通用的数学,在其中将概念、判断和推理进行符号化,从而将人类推理的正确性的验证转化成为形式上的数学演算。莱布尼兹的这一伟大而天才的思想使逻辑学的发展从此开始了一个新的阶段,标志着数理逻辑学科的产生。将推理变成数学演算就是数理逻辑的指导思想,也是这门学科的基本特征。由于莱布尼兹在建立新的逻辑系统中保留了对内涵的解释,因而,他在应用数学方法的过程中不断遇到困难,而英国数学家和逻辑学家布尔(G. Boole, 1815—1864)却采用了外延的方法,取得了重大的突破,发展了一种相当于命题演算的逻辑代数,一般称为布尔代数。数理逻辑研究的发展对计算机的产生有着极为深刻的影响,在计算机科学的硬件和软件理论及设计中发挥了巨大的作用,已经成为计算机科学最重要的理论基础之一。数理逻辑的一般研究方法是从一组逻辑公理出发,依据一些推理规则,通过代数演算来推导逻辑定理。于是许多科学家就试图建立通用的逻辑系统,验证所有反映客观实际规律的定理,或发现和导出所有未知定理。数学家希尔伯特(Hilbert)是所谓数学机械化思想的积极倡导者,他曾希望将整个数学形式化,建立一个协调、完全的大系统。但 1931 年哥德尔发表的不完全性定理(Kurt Gödel's incompleteness theorem)表明只要表达能力足够丰富,这