

高等学校理工类计算机基础课程教材

计算机硬件技术基础

(立体化教材)

白中英 主编

周锋 戴志涛 倪辉 张天乐 编著

陈国良 主审



高等教育出版社
Higher Education Press

高等学校理工类计算机基础课程教材

计算机硬件技术基础

(立体化教材)

白中英 主编

周锋 戴志涛 倪辉 张天乐 编著

陈国良 主审



高等教育出版社

内容提要

本书是作者对“计算机硬件技术基础”课程体系、教学内容、教学方法、教学手段进行综合研究的具体成果,内容分 10 章,包括计算机系统概论、运算方法和运算器、存储系统、指令系统、中央处理器、总线系统、外围设备、输入输出系统、安腾高性能处理机体系结构和片上系统。

本书特色:基础性、时代性、系统性、启发性、实践性、实用性融为一体,本书与副教材、多媒体 CAI 软件、教学课件、习题答案库、自测试题库、教学仪器、实验设计、课程设计综合配套,形成“理论、实验、设计”三个过程相统一的立体化教学体系。

本书文字流畅,内容通俗易懂,循序渐进,有广泛的适应面,是高等学校理工类计算机基础课程教材。

图书在版编目(CIP)数据

计算机硬件技术基础/白中英主编. —北京:高等教育出版社,2009.7
立体化教材
ISBN 978-7-04-026144-8

I. 计… II. 白… III. 硬件-高等学校-教材
IV. TP303

中国版本图书馆 CIP 数据核字(2009)第 038626 号

策划编辑 武林晓 责任编辑 关旭 封面设计 于文燕 责任绘图 尹莉
版式设计 陆瑞红 责任校对 王超 责任印制 宋克学

出版发行 高等教育出版社
社 址 北京市西城区德外大街 4 号
邮政编码 100120
总 机 010-58581000

经 销 蓝色畅想图书发行有限公司
印 刷 高等教育出版社印刷厂

开 本 787×1092 1/16
印 张 21.25
字 数 520 000
插 页 1

购书热线 010-58581118
免费咨询 800-810-0598
网 址 <http://www.hep.edu.cn>
<http://www.hep.com.cn>
网上订购 <http://www.landaco.com>
<http://www.landaco.com.cn>
畅想教育 <http://www.widedu.com>

版 次 2009 年 7 月第 1 版
印 次 2009 年 7 月第 1 次印刷
定 价 33.00 元(含光盘)

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究
物料号 26144-00

前 言

“微型计算机原理”是高等学校理工类非计算机专业本科生的一门必修课程,但是由于微型机更新换代速度太快,因此以型号为背景所写的教材往往使任课教师无所适从。存在的问题是:(1) 教学内容缺乏基础性和时代性;(2) 技术细节较多,即使学生花费很多的时间和精力,也难以消化教学内容;(3) 难以开展有效的实践教学。在此情况下,进行课程改革的呼声日益强烈。高等教育出版社《计算机硬件技术基础》(立体化教材)一书,正是响应了这种呼声而出版的。

作者认为,为了保持教学内容的基础性和时代性,应当避免只讲某种型号的处理机,一本好的《计算机硬件技术基础》教材主要应具备:

- (1) 基础性强,知识结构合理,为学生建立终生知识体系打下良好基础;
- (2) 系统性强,知识模块彼此交互,使学生能清晰地建立计算机整机概念;
- (3) 时代性强,及时反映前沿方向,以适应计算机科学技术快速发展的需要;
- (4) 实践性强,理论教学与实践教学结合,注重学生的智力开发和能力培养;
- (5) 实用性强,有较广的适应面,以适应学生在各类计算机上从事开发应用的需要;
- (6) 启发性强,结合计算机科学技术的重大进展,培养学生的创新思维和创新意识。

本书作者多年从事计算机硬件课程的理论教学和实践教学,从传授知识和培养能力的目标出发,并结合本课程教学的特点、难点和要点,进行了课程体系、教学内容、教学方法和教学手段的研究,在此基础上编写的文字教材与多媒体 CAI 软件、教学课件、习题答案库、自测试题库、教学仪器、实验设计综合配套,力求形成“理论、实验、设计”三个过程相统一的立体化教学体系。本书理论教学建议安排 64 学时,实践教学学时单独安排。因学时所限,带*的内容可以自学。

覃健诚、冯一兵、杨春武、陈楠、王莉、靳秀国、杨秦、张杰、祁之力、王军德、于艳丽、刘俊荣、张振华、宗华丽、李娇娇、王晓梅、胡文发、李贞、王坤山、肖炜、崔洪浚、高明、常培、张果桃、赵玉霞、郝静、吴璇、杨孟柯等参与了文字教材、CAI 软件、教学课件、习题答案库、试题库、教学仪器、实验设计等的编写和研制工作,限于版面,未能在封面上一一署名。

本书由中国科学院院士、中国科学技术大学国家高性能计算中心主任陈国良教授主审。清华大学科教仪器厂李鸿儒教授为配套教学仪器的研制给予了大力帮助,在此,作者一并向他们表示衷心感谢。

作 者

2008 年 10 月

目 录

第一章 计算机系统概论	1	同异性	30
1.1 计算机的分类	1	2.3.2 并行乘法器	31
1.2 计算机的发展简史	2	2.4 定点除法运算	36
1.2.1 计算机的五代变化	2	2.4.1 原码除法算法原理	36
1.2.2 半导体存储器的发展	3	2.4.2 并行除法器	37
1.2.3 微处理器的发展	3	2.5 定点运算器的组成	40
1.2.4 计算机的性能指标	5	2.5.1 逻辑运算	40
1.3 计算机的硬件	6	2.5.2 多功能算术/逻辑运算单元	
1.3.1 硬件组成要素	6	(ALU)	42
1.3.2 运算器	7	2.5.3 内部总线	46
1.3.3 存储器	7	2.5.4 定点运算器的基本结构	47
1.3.4 控制器	8	2.6 浮点运算方法和浮点运算器	48
1.3.5 适配器与输入输出设备	11	2.6.1 浮点加法与减法运算	48
1.4 计算机的软件	11	2.6.2 浮点乘法与除法运算	51
1.4.1 软件的组成与分类	11	2.6.3 浮点运算流水线	53
1.4.2 软件的发展演变	12	2.6.4 浮点运算器实例	55
1.5 计算机系统的层次结构	13	本章小结	57
1.5.1 多级组成的计算机系统	13	习题	57
1.5.2 软件与硬件的逻辑等价性	14	第三章 存储系统	59
本章小结	15	3.1 存储器概述	59
习题	15	3.1.1 存储器的分类	59
第二章 运算方法和运算器	16	3.1.2 存储器的分级	60
2.1 数据与文字的表示方法	16	3.1.3 主存储器的技术指标	60
2.1.1 数据格式	16	3.2 SRAM 存储器	61
2.1.2 数的机器码表示	19	3.2.1 基本的静态存储元阵列	61
2.1.3 字符与字符串的表示方法	23	3.2.2 基本的 SRAM 逻辑结构	62
2.1.4 汉字的表示方法	24	3.2.3 读/写周期波形图	63
2.1.5 校验码	25	3.3 DRAM 存储器	64
2.2 定点加法与减法运算	26	3.3.1 DRAM 存储元的记忆原理	64
2.2.1 补码加法	26	3.3.2 DRAM 芯片的逻辑结构	65
2.2.2 补码减法	27	3.3.3 读/写周期、刷新周期	66
2.2.3 溢出概念与检测方法	28	3.3.4 存储器容量的扩充	67
2.2.4 基本的二进制加法/减法器	29	3.4 只读存储器和闪速存储器	69
2.3 定点乘法运算	30	3.4.1 只读存储器 ROM	69
2.3.1 人工算法与机器算法的		3.4.2 FLASH 存储器	72

3.5 并行存储器	75	4.5.3 精简指令系统	118
3.5.1 双端口存储器	75	本章小结	120
3.5.2 多模块交叉存储器	77	习题	121
3.6 cache 存储器	81	第五章 中央处理机	123
3.6.1 cache 基本原理	81	5.1 CPU 的功能和组成	123
3.6.2 主存与 cache 的地址映射	82	5.1.1 CPU 的功能	123
3.6.3 替换策略	86	5.1.2 CPU 的基本组成	123
3.6.4 cache 的写操作策略	86	5.1.3 CPU 中的主要寄存器	124
3.6.5 Pentium 4 的 cache 组织	87	5.1.4 操作控制器与时序发生器	126
3.7 虚拟存储器	88	5.2 指令周期	126
3.7.1 虚拟存储器的基本概念	88	5.2.1 指令周期的基本概念	126
3.7.2 虚拟存储器分类	89	5.2.2 MOV 指令的指令周期	128
3.7.3 替换算法	92	5.2.3 LAD 指令的指令周期	130
3.7.4 虚拟存储器实例	94	5.2.4 ADD 指令的指令周期	131
3.8 存储保护	95	5.2.5 STO 指令的指令周期	132
3.8.1 存储区域保护	95	5.2.6 JMP 指令的指令周期	133
3.8.2 访问方式保护	97	5.2.7 用方框图语言表示指令 周期	135
本章小结	98	5.3 时序发生器	136
习题	98	5.3.1 时序信号的作用和体制	136
第四章 指令系统	100	5.3.2 时序信号发生器	137
4.1 指令系统的发展与性能要求	100	5.4 微程序控制器	139
4.1.1 指令系统的发展	100	5.4.1 微程序控制原理	139
4.1.2 对指令系统性能的要求	101	5.4.2 微程序设计技术	145
4.1.3 低级语言与硬件结构的 关系	101	5.5 传统 CPU	149
4.2 指令格式	102	5.5.1 Intel 8088 CPU	149
4.2.1 操作码	102	5.5.2 IBM 370 系列 CPU	150
4.2.2 地址码	103	5.6 流水 CPU	151
4.2.3 指令字长度	104	5.6.1 并行处理技术	151
4.2.4 指令助记符	104	5.6.2 流水 CPU 的结构	152
4.2.5 指令格式举例	105	5.6.3 流水线中的主要问题	154
4.3 操作数类型	107	5.6.4 奔腾 CPU	156
4.3.1 一般的数据类型	107	5.7 RISC CPU	160
4.3.2 Pentium 数据类型	108	5.7.1 RISC 机器的特点	160
4.3.3 PowerPC 数据类型	108	5.7.2 RISC CPU 实例	162
4.4 指令和数据的寻址方式	109	5.8 多媒体 CPU	165
4.4.1 指令的寻址方式	109	5.8.1 多媒体技术的主要问题	165
4.4.2 操作数基本寻址方式	110	5.8.2 MMX 技术	166
4.4.3 寻址方式举例	113	本章小结	169
4.5 典型指令	116	习题	169
4.5.1 指令的分类	116	第六章 总线系统	173
4.5.2 基本指令系统的操作	117	6.1 总线的概念和结构形态	173

6.1.1 总线的基本概念	173	7.6.2 字符/图形显示器	211
6.1.2 总线的连接方式	174	7.6.3 图像显示设备	212
6.1.3 总线的内部结构	176	7.6.4 VESA 显示标准	213
6.2 总线接口	177	7.7 输入设备和打印设备	216
6.2.1 信息传送方式	177	7.7.1 输入设备	216
6.2.2 总线接口的基本概念	178	7.7.2 打印设备	217
6.3 总线的仲裁	180	本章小结	218
6.3.1 集中式仲裁	180	习题	219
6.3.2 分布式仲裁	182	第八章 输入输出系统	221
6.4 总线的定时和数据传送模式	183	8.1 外围设备的速度分级及与 CPU 进行 数据交换的方式	221
6.4.1 总线的定时	183	8.1.1 外围设备的速度分级	221
6.4.2 总线数据传送模式	185	8.1.2 外设与 CPU 进行数据交换的 方式	222
6.5 HOST 总线和 PCI 总线	186	8.2 程序查询方式	224
6.5.1 多总线结构	186	8.3 程序中断方式	227
6.5.2 PCI 总线信号	187	8.3.1 中断的基本概念	227
6.5.3 总线周期类型	188	8.3.2 程序中断方式的基本 I/O 接口	228
6.5.4 总线周期操作	189	8.3.3 单级中断和多级中断	230
6.5.5 总线仲裁	191	8.3.4 中断控制器	234
6.6 InfiniBand 标准	191	8.3.5 Pentium 中断机制	235
本章小结	192	8.4 DMA 方式	238
习题	193	8.4.1 DMA 的基本概念	238
第七章 外围设备	195	8.4.2 DMA 传送方式	238
7.1 外围设备概述	195	8.4.3 基本的 DMA 控制器	240
7.1.1 外围设备的一般功能	195	8.4.4 选择型和多路型 DMA 控制器	242
7.1.2 外围设备的分类	195	8.5 通道方式	245
7.2 磁盘存储设备	196	8.5.1 通道的功能	245
7.2.1 磁记录原理	196	8.5.2 通道的类型	247
7.2.2 硬磁盘的组成和分类	199	8.5.3 通道结构的发展	248
7.2.3 磁盘驱动器和控制器	200	8.6 通用 I/O 标准接口	248
7.2.4 磁盘上信息的分布	201	8.6.1 并行 I/O 标准接口 SCSI	248
7.2.5 磁盘存储器的技术指标	202	8.6.2 串行 I/O 标准接口 IEEE1394	249
7.3 磁盘存储设备的技术发展	203	8.6.3 串行 I/O 标准接口 USB	251
7.3.1 磁盘 cache	203	本章小结	252
7.3.2 磁盘阵列 RAID	204	习题	253
7.3.3 可移动存储设备	205	第九章 安腾高性能处理机体系结构	255
7.4 磁带存储设备	205	9.1 高性能处理机体系结构的演变	255
7.5 光盘和磁光盘存储设备	207	9.1.1 IA 体系结构的历史演变	255
7.5.1 光盘存储设备	207	9.1.2 Intel 64 位处理机的两种体系	
7.5.2 磁光盘存储设备	209		
7.6 显示设备	210		
7.6.1 显示设备的分类与有关 概念	210		

结构	256	10.3.1 CIP-51 内部结构	280
9.2 安腾体系结构的基本设计思想	257	10.3.2 存储器组织	281
9.3 安腾指令系统结构	259	10.3.3 特殊功能寄存器 SFR	285
9.3.1 执行单元与指令类型	259	10.4 CIP-51 指令系统	288
9.3.2 安腾寄存器结构	260	10.4.1 指令格式	288
9.3.3 安腾指令格式	261	10.4.2 寻址方式	289
9.3.4 安腾汇编语言格式	262	10.4.3 指令类型	292
9.4 指令级并行机制	263	10.5 模拟量数字量转换与 I/O 接口	297
9.4.1 推断执行技术	264	10.5.1 数字信号处理系统	297
9.4.2 推测技术	266	10.5.2 模拟数字转换原理	297
9.5 双核安腾处理机的组成	269	10.5.3 数字模拟转换原理	300
9.5.1 双核安腾处理机的基本 特性	269	10.5.4 I/O 接口	302
9.5.2 双核安腾处理机的组织 结构	271	10.6 编程应用	306
本章小结	273	10.6.1 计算机编程的相关概念	307
习题	274	10.6.2 汇编语言程序	310
第十章 片上系统	276	10.6.3 C 语言程序	315
10.1 嵌入式系统与片上系统	276	本章小结	320
10.1.1 嵌入式系统	276	习题	321
10.1.2 片上系统	276	附录 A C8051F 指令系统	324
10.2 C8051F 片上系统体系结构	277	附录 B 配套教材与教学设备	329
10.3 CIP-51 内核	280	参考文献	330
		参考网站	330

第一章 计算机系统概论

计算机系统不同于一般的电子设备,它是一个由硬件、软件组成的复杂的自动化设备。本章先说明计算机的分类,然后采用自上而下的方法,简要地介绍硬件、软件的概念和组成,目的在于使读者先有一个粗略的总体概念,以便于展开后续各章内容。

1.1 计算机的分类

电子计算机从总体上来说分为两大类。一类是电子模拟计算机。“模拟”就是相似的意思,例如计算尺是用长度来标示数值;时钟是用指针在表盘上的转动来表示时间;电表是用角度来反映电量大小。这些都是模拟计算装置。模拟计算机的特点是,数值由连续量来表示,运算过程也是连续的。

另一类是电子数字计算机,它是在算盘的基础上发展起来的,是用数字来表示数量的大小。电子数字计算机的主要特点是按位运算。表 1.1 列出了电子数字计算机与电子模拟计算机的主要区别。

表 1.1 电子数字计算机与电子模拟计算机的主要区别

比较内容	数字计算机	模拟计算机
数据表示方式	数字 0 和 1	电压
计算方式	数字计数	电压组合和测量值
控制方式	程序控制	盘上连线
精度	高	低
数据存储量	大	小
逻辑判断能力	强	无

电子模拟计算机由于精度和解题能力都有限,所以应用范围较小。电子数字计算机则与电子模拟计算机不同,它是以近似于人类的“思维过程”来进行工作的,所以有人把它叫做**电脑**。它的发明和发展是 20 世纪人类最伟大的科学技术成就之一,也是现代科学技术发展水平的主要标志。习惯上所称的**电子计算机**,一般是指现在广泛应用的电子数字计算机,简称数字计算机。

数字计算机进一步又可分为专用计算机和通用计算机。专用和通用是根据计算机的效率、速度、价格、运行的经济性和适应性来划分的。专用机是最有效、最经济或最快速的计算机,但是它的通用性很差。通用计算机适应性很强,但是牺牲了效率、速度或经济性。

通用计算机可分超级计算机、大型机、服务器、工作站、微型机和单片机六类,它们的区别在于体积、简易性、功(率)耗、性能(指标)、数据存储(容)量、指令系统(规模)和(机器)价格,见

图 1.1。一般来说,超级计算机主要用于科学计算,其运算速度在每秒万亿次以上,数据存储容量很大,结构复杂,价格昂贵。而单片机是只用一片集成电路作成的计算机,体积小,结构简单,性能指标较低,价格便宜。介于超级计算机和单片机之间的是大型机、服务器、工作站和微型机,它们的结构规模和性能指标依次递减。但是随着超大规模集成电路的迅速发展,微型机、工作站等彼此之间的概念也在发生变化,因为今天的工作站可能就是明天的微型机,而今天的微型机可能就是明天的单片机。专用计算机是针对某一任务设计的计算机,一般来说,其结构要比通用机简单。目前已经出现了多种型号的单片专用机及嵌入式单片机,用于测试或工业控制。

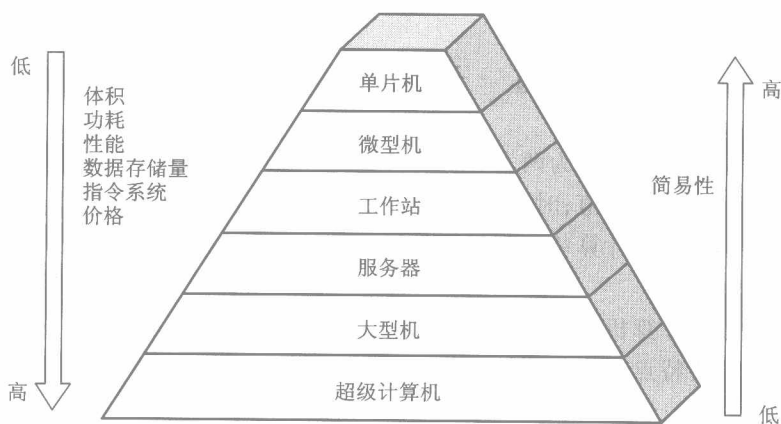


图 1.1 单片机、微型机、工作站、服务器、大型机、超级计算机之间的区别

1.2 计算机的发展简史

1.2.1 计算机的五代变化

世界上第一台电子数字计算机是 1946 年在美国宾夕法尼亚大学制成的。这台机器用了 18 000 多个电子管,占用超过 30m^2 的房间,重量达 30 吨,而运算速度只有 5000 次/秒。从今天的目光来看,这台计算机体积庞大且功能简单,但却是科学史上一次划时代的创新,它奠定了电子计算机的基础。自从这台计算机问世 60 多年来,从所用器件的角度来说,计算机的发展大致经历了五代的变化。

第一代为 1946—1957 年,电子管计算机。计算机运算速度为每秒几千次至几万次,体积庞大,成本很高,可靠性较差。在此期间,形成了计算机的基本体系,确定了程序设计的基本方法,数据处理机开始得到应用。

第二代为 1958—1964 年,晶体管计算机。运算速度提高到几万次至几十万次,可靠性提高,体积缩小,成本降低。在此期间,工业控制机开始得到应用。

第三代为 1965—1971 年,中小规模集成电路计算机。可靠性进一步提高,体积进一步缩小,成本进一步下降,运算速度提高到几十万次至几百万次。在此期间形成机种多样化,生产系列化,使用系统化的发展格局,小型计算机开始出现。

第四代为1972—1990年,大规模和超大规模集成电路计算机,可靠性更进一步提高,体积更进一步缩小,成本更进一步降低,速度提高到每秒1 000万次至1亿次。由几片大规模集成电路组成的微型计算机开始出现。

第五代为1991年开始的巨大规模集成电路计算机,运算速度提高到每秒10亿次。由一片巨大规模集成电路实现的单片计算机开始出现。

总之,从1946年计算机诞生以来,大约每隔五年运算速度提高10倍,可靠性提高10倍,成本降低10倍,体积缩小10倍。而20世纪70年代以来,计算机的生产数量每年以25%的速度递增。

值得一提的是,计算机从第三代起,与集成电路技术的发展密切相关。大规模集成电路(LSI)的采用,使得一块集成电路芯片上可以放置1 000个元件;超大规模集成电路(VLSI)达到每个芯片1万个元件,现在的甚大规模集成电路(ULSI)芯片超过了100万个元件。1965年摩尔观察到芯片上的元件数量每年翻一番,1970年这种态势减慢成每18个月翻一番,这就是人们所称的摩尔定律。直到目前,摩尔定律依然有效。

1.2.2 半导体存储器的发展

20世纪50—60年代,所有计算机存储器都是由微小的铁磁体环(磁心)作成,每个磁心直径约1 mm。这些小磁心处在计算机内用三条细导线穿过网格板,每个磁心的一种磁化方向代表一个1,另一个磁化方向则代表一个0。磁心存储器速度相当快,读存储器中的一位只需1 μ s。但是磁心存储器价格昂贵,体积大,而且读出是破坏性的,因此必须有读出后立即重写数据的电路。更重要的是,它工艺复杂,甚至需要手工制作。

1970年,仙童半导体公司生产出了第一个较大容量半导体存储器。一个相当于单个磁心大小的芯片,包含了256位的存储器。这种芯片是非破坏性的,而且读写速度比磁心快得多,读出一位只要70 ns,但是其价格比磁心要贵。

1974年每位半导体存储器的价格低于磁心。这以后,存储器的价格持续快速下降,但存储密度却不断增加,这导致了新的计算机比它之前的计算机更小、更快、存储容量更大,价格更便宜。存储器技术的发展与处理器技术的发展一起,在不到10年里为计算机的发展赋予了旺盛的生命力。虽然庞大、昂贵的计算机仍然存在,但计算机已经走向了个人电脑时代。

从1970年起,半导体存储器经历了11代:单个芯片1 KB、4 KB、16 KB、64 KB、256 KB、1 MB、4 MB、16 MB、64 MB、256 MB和1 GB。其中 $1\text{ K}=2^{10}$, $1\text{ M}=2^{20}$, $1\text{ G}=2^{30}$ 。每一代是前一代存储密度的4倍,而每位价格和存取时间都在下降。

1.2.3 微处理器的发展

同存储器芯片一样,处理器芯片的单元密度也在不断增加。随着时间的推移,每块芯片上的单元个数越来越多,因此构建一个计算机处理器所需的芯片数量越来越少。表1.2列出了Intel公司微处理器的演化过程。

表 1.2 Intel 微处理器的演化

(a) 20 世纪 70 年代的处理器的					
型 号	4004	8008	8080	8086	8088
发布时间	1971 年	1972 年	1974 年	1978 年	1979 年
时钟速率	108 kHz	108 kHz	2 MHz	5 MHz, 8 MHz, 10 MHz	5 MHz, 8 MHz
总线宽度	4 位	8 位	8 位	16 位	8 位
晶体管数	2 300 个	3 500 个	6 000 个	29 000 个	29 000 个
特征尺寸	10 μm	—	6 μm	3 μm	3 μm
可寻址存储器	640 B	16 KB	64 KB	1 MB	1 MB
虚拟存储器	—	—	—	—	—
(b) 20 世纪 80 年代的处理器的					
型 号	80286	386TM DX	386TM SX	486TM DX	
发布时间	1982 年	1985 年	1988 年	1989 年	
时钟速率	6 ~ 12.5 MHz	16 ~ 33 MHz	16 ~ 33 MHz	25 ~ 50 MHz	
总线宽度	16 位	32 位	16 位	32 位	
晶体管数	134 000 个	275 000 个	275 000 个	1 200 000 个	
特征尺寸	1.5 μm	1 μm	1 μm	0.8 ~ 1 μm	
可寻址存储器	16 MB	4 GB	16 MB	4 GB	
虚拟存储器	1 GB	64 TB	64 TB	64 TB	
(c) 20 世纪 90 年代的处理器的					
型 号	486TM SX	Pentium	Pentium Pro	Pentium II	
发布时间	1991 年	1993 年	1995 年	1997 年	
时钟速率	16 ~ 33 MHz	60 ~ 166 MHz	150 ~ 220 MHz	200 ~ 300 MHz	
总线宽度	32 位	32 位	64 位	64 位	
晶体管数	1.185 百万个	3.1 百万个	5.5 百万个	7.5 百万个	
特征尺寸	1 μm	0.8 μm	0.6 μm	0.35 μm	
可寻址存储器	4 GB	4 GB	64 GB	64 GB	
虚拟存储器	64 TB	64 TB	64 TB	64 TB	
(d) 最近的处理器的					
型 号	Pentium III	Pentium 4	Itanium	Itanium 2	
发布时间	1999 年	2000 年	2001 年	2002 年	
时钟速率	450 ~ 600 MHz	1.3 ~ 1.8 GHz	733 ~ 800 MHz	0.9 ~ 1 GHz	
总线宽度	64 位	64 位	64 位	64 位	

续表

型 号	Pentium III	Pentium 4	Itanium	Itanium 2
晶体管数	9.6 百万个	42 百万个	25 百万个	220 百万个
特征尺寸	0.25 μm	0.18 μm	0.18 μm	0.18 μm
可寻址存储器	64 GB	64 GB	64 GB	64 GB
虚拟存储器	64 TB	64 TB	64 TB	64 TB

1971 年 Intel 公司开发出 Intel 4004。这是第一个将 CPU 的所有元件都放入同一块芯片内的产品,于是,微处理器诞生了。

4004 能完成两个 4 位数相加,通过重复相加能完成乘法。按今天的标准,4004 虽然过于简单,但是它却成为微处理器的能力和功能不断发展的里程碑。

微处理器演变中的另一个主要进步是 1972 年出现的 Intel 8008,这是第一个 8 位微处理器,它比 4004 复杂一倍。

1974 年出现了 Intel 8080。这是第一个通用微处理器,而 4004 和 8008 是为特殊用途而设计的。8080 是为通用微机而设计的中央处理器。它与 8008 一样,都是 8 位微处理器,但 8080 更快,有更丰富的指令集和更强的寻址能力。大约在同时,16 位微机被开发出来。但是直到 20 世纪 70 年代末才出现强大的通用 16 位微处理器,8086 便是其中之一。1981 年,贝尔实验室和 HP 公司开发出了 32 位单片微处理器。Intel 于 1985 年推出了 32 位微处理器 Intel 80386。

1.2.4 计算机的性能指标

吞吐量:表征一台计算机在某一时间间隔内能够处理的信息量。

响应时间:表征从输入有效到系统产生响应之间的时间度量,用时间单位来度量。

利用率:在给定的时间间隔内,系统被实际使用的时间比率,用百分比表示。

处理机字长:指 CPU 运算器中一次能够完成二进制数运算的位数,如 32 位、64 位。

总线宽度:一般指 CPU 中运算器与存储器之间进行互连的内部总线的二进制位数。

存储器容量:存储器中所有存储单元的数目,通常用 KB、MB、GB、TB 来表示。

存储器带宽:单位时间内从存储器读出的二进制数信息量,一般用 Bps 表示。

主频/时钟周期:CPU 的工作节拍受主时钟控制,主时钟不断产生固定频率的时钟,主时钟的频率(f)称为 **CPU 的主频**。度量单位是 MHz(兆赫兹)、GHz(吉赫兹)。

主频的倒数称为 **CPU 时钟周期**(T),即 $T = 1/f$,度量单位是 μs (微秒)、ns(纳秒)

CPU 执行时间:表示 CPU 执行一段程序所占用的 CPU 时间,可用下式计算:

$$\text{CPU 执行时间} = \text{CPU 时钟周期数} \times \text{CPU 时钟周期}$$

CPI:表示每条指令周期数,即执行一条指令所需的平均时钟周期数。用下式计算:

$$\text{CPI} = \text{执行某段程序所需的 CPU 时钟周期数} \div \text{该程序包含的指令条数}$$

MIPS:每秒百万条指令数,即单位时间内执行的指令数。用下式计算:

$$\text{MIPS} = \text{指令条数} \div (\text{程序执行时间} \times 10^6)$$

MFLOPS:每秒百万次浮点操作次数,用来衡量计算机浮点操作的性能。用下式计算:
$$\text{MFLOPS} = \text{程序中的浮点操作次数} \div (\text{程序执行时间} \times 10^6)$$

1.3 计算机的硬件

1.3.1 硬件组成要素

要了解数字计算机的主要组成和工作原理,可从打算盘说起。假设给一个算盘、一张带有横格的纸和一支笔,要求计算 $y = ax + b - c$ 这样一个题目。为了和下面讲到的内容做比较,我们不妨按以下方法把使用算盘进行解题的步骤事先用笔详细地记录在带横格的纸上。

首先,将横格纸编上序号,每一行占一个序号,如 1,2,3,⋯,n,如表 1.3 所示。其次,把计算式中给定的四个数 a, b, c 和 x 分别写到横格纸的第 9,10,11,12 行上,每一行只写一个数。接着详细列出给定题目的解题步骤,而解题步骤也需要记在横格纸上,每一步也只写一行。第一步写到横格纸的第 1 行,第二步写到第 2 行……依此类推。

根据表 1.3 所列的解题步骤,我们从第 1 行开始一步一步进行计算,最后可得出所要求的结果。

表 1.3 解题步骤和数据记录在横格纸上

行	解题步骤和数据		说 明
1	取数	(9)→算盘	(9) 表示第 9 行的数 a ,下同
2	乘法	(12)→算盘	完成 $a \cdot x$,结果在算盘上
3	加法	(10)→算盘	完成 $ax + b$,结果在算盘上
4	减法	(11)→算盘	完成 $y = ax + b - c$,结果在算盘上
5	存数	$y \rightarrow 13$	算盘上的 y 值记到第 13 行
6	输出		把算盘上的 y 值写出给人看
7	停止		运算完毕,暂停
8			
9	a		数据
10	b		数据
11	c		数据
12	x		数据
13	y		数据

在完成 $y = ax + b - c$ 的计算过程中,我们用到了什么呢?

首先,用到了带横格且编有序号的纸,我们把原始的数据以及解题步骤记录在纸上,即纸“存储”了算题的原始信息。其次,用到了算盘,它用来对数据进行加、减、乘、除等算术运算。再次,用到了笔,利用笔把原始数据和解题步骤记录到纸上,还可把计算结果写出来告诉人。最后,用到了人本身,这主要是人的脑和手。在人的控制下,按照解题步骤一步一步进行操作,直到完成

全部运算。

电子数字计算机进行解题的过程和人用算盘解题的情况非常相似,也必须有运算工具,解题步骤和原始数据的输入与存储,运算结果的输出以及整个计算过程的调度控制。和打算盘不同的是,以上这些部分都是由电子线路和其他设备自动进行的。在电子计算机里,相当于算盘功能的部件,我们称之为**运算器**;相当于纸那样具有“记忆”功能的部件,我们称之为**存储器**;相当于笔那样把原始解题信息送到计算机或把运算结果显示出来的设备,我们称之为**输入设备**或**输出设备**;而相当于我们人的大脑,能够自动控制整个计算过程的,我们称之为**控制器**。图 1.2 示出了数字计算机的主要组成部分,其中带箭头的双线表示数据代码的传送通路。

1.3.2 运算器

运算器就好像是一个由电子线路构成的算盘,图 1.3 是它的示意图。它的主要功能是进行加、减、乘、除等算术运算。除此以外,还可以进行逻辑运算,因此通常称为 ALU(算术逻辑运算部件)。

人们习惯于十进制数的运算,但是考虑到电子器件的特性,计算机中通常采用二进制进行运算。二进制是以 2 为基数来记数的,也就是“逢二进一”。在二进制数中,只有 0 和 1 两个数字。1 和 0 可以用电压的高低、脉冲的有无来表示。这种电压的高低、脉冲的有无,在电子器件中很容易实现,而且设备也最省。

二进制数的运算规则非常简单。例如加法: $0+0=0$, $0+1=1$, $1+0=1$, $1+1=10$,最后一个加式中等于号右边的“1”表示向上一位的进位。又如乘法: $0\times 0=0$, $0\times 1=0$, $1\times 0=0$, $1\times 1=1$ 。正是由于二进制数运算规则简单,在电子器件中比较容易实现,因此,在电子数字计算机中广泛采用二进制数。

二进制数和十进制数一样,在运算中,当数的位数越多时,计算的精度就越高。理论上讲,数的位数可以任意多。位数越多,所需的电子器件也越多。计算机的运算器长度一般是 8 位、16 位、32 位、64 位。

1.3.3 存储器

存储器的功能是保存或“记忆”解题的原始数据和解题步骤。为此,在运算前需要把参加运算的数据和解题步骤通过输入设备送到存储器中保存起来。

注意,不论是数据,还是解题步骤,在存放到存储器以前,它们全已变成 0 或 1 表示的二进制代码。因此,存储器存储

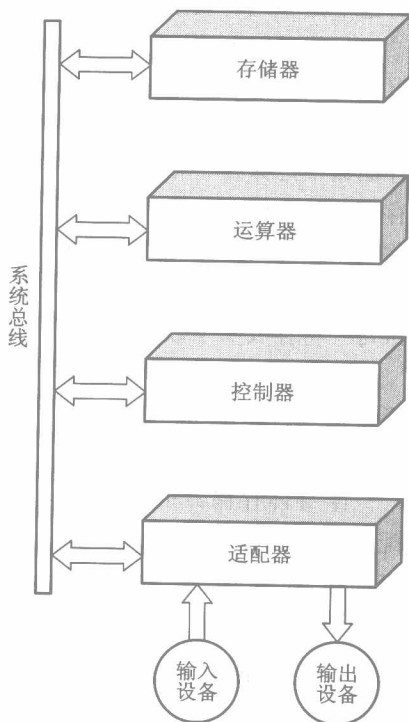


图 1.2 数字计算机的主要组成结构

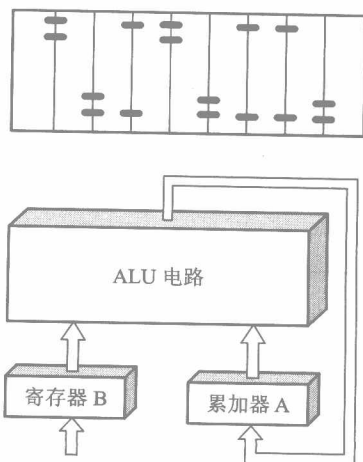


图 1.3 运算器结构示意图

的也全是 0 或 1 表示的二进制代码。那么大量的 0,1 代码在存储器中如何保存呢?

目前采用半导体器件来担当此任务。我们知道,一个半导体触发器由于有 0 和 1 两个状态,可以记忆一个二进制代码。一个数假定用 16 位二进制代码来表示,那么就需要有 16 个触发器来保存这些代码。通常,在存储器中把保存一个数的 16 个触发器称为一个**存储单元**。存储器是由许多存储单元组成的。每个存储单元都有编号,称为**地址**。向存储器中存数或者从存储器中取数,都要按给定的地址来寻找所选的存储单元,这相当于上面所讲的横格纸每一行存放一个数。图 1.4 为存储器的结构示意图。

存储器所有存储单元的总数称为存储器的**存储容量**,通常用单位 KB、MB 来表示,如 64 KB、128 MB。存储容量越大,表示计算机记忆或存储的信息就越多。

半导体存储器的存储容量毕竟有限,因此计算机中又配备了存储容量更大的磁盘存储器和光盘存储器,称为**外存储器**,简称外存。相对而言,半导体存储器称为**内存**存储器,简称内存。

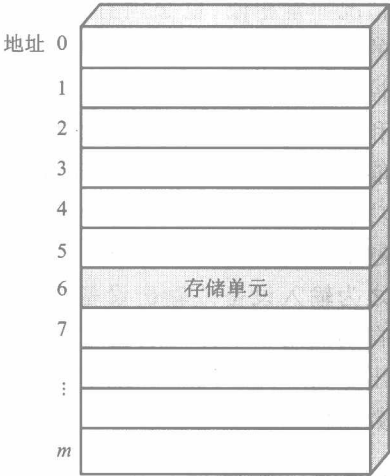


图 1.4 存储器结构示意图

1.3.4 控制器

控制器是计算机中发号施令的部件,它控制计算机的各部件有条不紊地进行工作。更具体地讲,控制器的任务是从内存中取出解题步骤加以分析,然后执行某种操作。

1. 计算程序

运算器只能完成加、减、乘、除四则运算及其他一些辅助操作。对于比较复杂的计算题,计算机在运算前必须转化成一步一步简单的加、减、乘、除等基本操作来做。每一个基本操作就叫做一条**指令**,而解算某一问题的一串指令序列,叫做该问题的**计算程序**,简称为程序。

例如在前述求解 $y = ax + b - c$ 的例子中,我们在横格纸上列出了它的解题步骤。解题的每一步骤,只完成一种基本操作,所以就是一条指令,而整个解题步骤就是一个简单的计算程序。

正如我们在横格纸上按行的序号记下解题步骤一样,计算机中为了顺利运算,也必须事先把程序和数据按地址存放到存储器里去。注意,程序中的指令通常按顺序执行,所以这些指令是顺次放在存储器里的。这就相当于我们把表 1.3 所示的横格纸的内容原封不动地搬到存储器中,因而所编的程序如表 1.4 所示。

表 1.4 计算 $y = ax + b - c$ 的程序

指令地址	指令		指令操作内容	说 明
	操作码	地址码		
1	取数	9	$(9) \rightarrow A$	存储器 9 号地址的数 a 运算器 A
2	乘法	12	$(A) \times (12) \rightarrow A$	完成 $a \cdot x$, 结果保留在运算器 A

续表

指令地址	指令		指令操作内容	说 明
	操作码	地址码		
3	加法	10	$(A) + (10) \rightarrow A$	完成 $ax + b$, 结果保留在运算器 A
4	减法	11	$(A) - (11) \rightarrow A$	完成 $y = ax + b - c$, 结果保留在 A
5	存数	13	$(A) \rightarrow 13$	运算器 A 中的结果 y 送入存储器 13 号地址
6	打印		$(A) \rightarrow \text{Print}$	将 A 中的结果经打印机打印出来
7	停止		Stop	计算机停止工作
8				
数据地址	数据		说 明	
9	a		数据 a 存放在 9 号单元	
10	b		数据 b 存放在 10 号单元	
11	c		数据 c 存放在 11 号单元	
12	x		数据 x 存放在 12 号单元	
13	y		运算结果 y 存放在 13 号单元	

2. 指令的形式

由表 1.4 可知, 每条指令应当明确告诉控制器, 从存储器的哪个单元取数, 并进行何种操作。这样一来, 可知指令的内容由两部分组成, 即操作的性质和操作数的地址。前者称为**操作码**, 后者称为**地址码**。因而上述指令的形式如下:

操作码	地址码
-----	-----

其中操作码指出指令所进行的操作, 如加、减、乘、除、取数、存数等; 而地址码表示参加运算的数据应从存储器的哪个单元中取来, 或运算的结果应该存到哪个单元中去。

指令的操作码和地址码是用二进制代码来表示的, 其中地址码部分和数据一样, 是二进制数的数码, 而操作码部分则是二进制代码的编码。假定我们只有 8 种基本指令, 那么这 8 种指令的操作码可用 3 位二进制代码来定义, 如表 1.5 所示。

这样一来, 表 1.5 中指令的操作码部分就可以变成二进制代码。假如把地址码部分和数据也换成二进制数, 那么整个存储器的内容全部变成了二进制的代码(或数码), 如图 1.5 所示。

由图 1.5 可知, 指令数码化以后, 就可以和数据一样放入存储器。存储器的任何位置既可以存放数据也可以存放指令, 不过一般是将指令和数据分开存放。将解题的程序(指令序列)存放在存储器中称为**存储程序**; 而控制器依据存储的程序来控制计算机协调地完成计算任务叫做**程序控制**。存储程序并按地址顺序执行, 这就是冯·诺依曼型计算机的工作原理, 也是计算机自动化工作的关键。为了提高速度, 现代计算机采用了指令存储器和数据存储器独立配置的设计思想, 称为**哈佛结构**。