



普通高等教育**计算机类**特色专业系列规划教材

# 计算机组成与系统结构

(第五版·立体化教材)

	白中英	主编	
戴志涛	张天乐	覃健诚	编著
	陈鸿安	主审	

第四版 2008年北京市高等教育教学成果二等奖  
2011年普通高等教育精品教材奖



科学出版社

普通高等教育计算机类特色专业系列规划教材

# 计算机组成与系统结构

(第五版·立体化教材)

白中英 主编

戴志涛 张天乐 覃健诚 编著

陈鸿安 主审

科学出版社

北京

## 第五版前言

“计算机组成与系统结构”是计算机科学与技术学科的一门核心专业基础课程。

“计算机组成与系统结构”将“计算机组成原理”、“计算机系统结构”两门课程合二为一，重点讲授单处理机系统的组成和工作原理，在此基础上扩展讲授并行计算机的体系结构。课程教学具有知识面广、内容多、难度大、更新快等特点。另一方面，体现课程特点的教材对于提高教学水平和培养人才的质量起着十分重要的作用。作者认为，一本好的《计算机组成与系统结构》教材主要应具备下述特点：

- (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的培养。
- (5) 应用性强，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。

按计算机学科教指委 2008 教学计划，本课程的先修课是“数字逻辑”。

作者根据多年从事计算机硬件课程理论教学和实践教学的经验，从传授知识和培养能力的目标出发，结合本课程教学的特点、难点和要点，使文字教材、多媒体 CAI 软件、教学课件、习题库、试题库、教学仪器、实验设计、课程设计综合配套，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。

理论教学计划 64 学时，授课学时建议：第一章 3（学时），第二章 9，第三章 10，第四章 5，第五章 13，第六章 4，第七章 3，第八章 6，第九章 3，第十章 8。课程实验单独安排 16 学时，可与理论教学同步进行。

杨春武、冯一兵、张杰、靳秀国、杨秦、陈楠、于艳丽、刘敬晗、祁之力、王军德、张振华、刘俊荣、李娇娇、李贞、王坤山、肖炜、崔洪浚、白媛、吴琨等参与了第五版文字教材、CAI 课件、习题库、试题库、教学仪器、实验设计、课程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。

本书由中国科学院计算技术研究所国家智能计算机研究开发中心陈鸿安研究员主审。在此，作者表示衷心感谢。

作者  
2011 年 8 月于北京

# 目 录

## 第五版前言

### 第一章 计算机系统概论 /1

- 1.1 计算机的分类 /1
  - 1.2 计算机的发展简史 /2
    - 1.2.1 计算机的五代变化 /2
    - 1.2.2 半导体存储器的发展 /3
    - 1.2.3 微处理器的发展 /3
    - 1.2.4 计算机的性能指标 /5
  - 1.3 计算机的硬件 /6
    - 1.3.1 硬件组成要素 /6
    - 1.3.2 运算器 /7
    - 1.3.3 存储器 /8
    - 1.3.4 控制器 /8
    - 1.3.5 适配器与输入输出设备 /11
  - 1.4 计算机软件 /11
    - 1.4.1 软件的组成与分类 /11
    - 1.4.2 软件的发展演变 /12
  - 1.5 计算机系统的层次结构 /13
    - 1.5.1 多级组成的计算机系统 /13
    - 1.5.2 软件与硬件的逻辑等价性 /14
- 本章小结 /14  
习题 /15

### 第二章 运算方法和运算器 /16

- 2.1 数据与文字表示方法 /16
  - 2.1.1 数据格式 /16
  - 2.1.2 数的机器码表示 /19
  - 2.1.3 字符与字符串表示方法 /23
  - 2.1.4 汉字的表示方法 /24
  - 2.1.5 校验码 /25
- 2.2 定点加法、减法运算 /26
  - 2.2.1 补码加法 /26
  - 2.2.2 补码减法 /27
  - 2.2.3 溢出概念与检测方法 /28
  - 2.2.4 基本的二进制加法/减法器 /29
- 2.3 定点乘法运算 /31

- 2.3.1 原码并行乘法 /31
  - 2.3.2 直接补码并行乘法 /36
  - 2.4 定点除法运算 /38
    - 2.4.1 原码除法算法原理 /38
    - 2.4.2 并行除法器 /40
  - 2.5 定点运算器的组成 /43
    - 2.5.1 逻辑运算 /43
    - 2.5.2 多功能算术/逻辑运算单元 (ALU) /45
    - 2.5.3 内部总线 /49
    - 2.5.4 定点运算器的基本结构 /49
  - 2.6 浮点运算方法和浮点运算器 /51
    - 2.6.1 浮点加法、减法运算 /51
    - 2.6.2 浮点乘法、除法运算 /54
    - 2.6.3 浮点运算流水线 /55
    - 2.6.4 浮点运算器实例 /58
- 本章小结 /60  
习题 /60

### 第三章 多层次的存储器 /62

- 3.1 存储器概述 /62
  - 3.1.1 存储器的分类 /62
  - 3.1.2 存储器的分级 /63
  - 3.1.3 主存储器的技术指标 /63
- 3.2 SRAM 存储器 /64
  - 3.2.1 基本的静态存储元阵列 /64
  - 3.2.2 基本的 SRAM 逻辑结构 /65
  - 3.2.3 读/写周期波形图 /66
- 3.3 DRAM 存储器 /67
  - 3.3.1 DRAM 存储元的记忆原理 /67
  - 3.3.2 DRAM 芯片的逻辑结构 /68
  - 3.3.3 读/写周期、刷新周期 /69
  - 3.3.4 存储器容量的扩充 /70
  - 3.3.5 高级的 DRAM 结构 /71
  - 3.3.6 DRAM 主存读/写的正确性校验 /75
- 3.4 只读存储器和闪速存储器 /76

3.4.1	只读存储器 ROM	/76
3.4.2	FLASH 存储器	/80
3.5	并行存储器	/82
3.5.1	双端口存储器	/82
3.5.2	多模块交叉存储器	/85
3.6	cache 存储器	/88
3.6.1	cache 基本原理	/88
3.6.2	主存与 cache 的地址映射	/90
3.6.3	替换策略	/95
3.6.4	cache 的写操作策略	/95
3.6.5	Pentium 4 的 cache 组织	/96
3.7	虚拟存储器	/97
3.7.1	虚拟存储器的基本概念	/97
3.7.2	页式虚拟存储器	/99
3.7.3	段式虚拟存储器和段页式虚拟存储器	/101
3.7.4	虚存的替换算法	/104
3.8	奔腾系列机的虚存组织	/104
3.8.1	存储器模型	/104
3.8.2	虚地址模式	/105
3.8.3	分页模式下的地址转换	/106
3.9	存储保护	/107
3.9.1	存储区域保护	/107
3.9.2	访问方式保护	/109
	本章小结	/109
	习题	/110
<b>第四章</b>	<b>指令系统</b>	<b>/113</b>
4.1	指令系统的发展与性能要求	/113
4.1.1	指令系统的发展	/113
4.1.2	对指令系统性能的要求	/113
4.1.3	低级语言与硬件结构的关系	/114
4.2	指令格式	/115
4.2.1	操作码	/115
4.2.2	地址码	/116
4.2.3	指令字长度	/117
4.2.4	指令助记符	/117
4.2.5	指令格式举例	/118
4.3	操作数类型	/120
4.3.1	一般的数据类型	/120
4.3.2	Pentium 数据类型	/120

4.3.3	Power PC 数据类型	/120
4.4	指令和数据的寻址方式	/121
4.4.1	指令的寻址方式	/121
4.4.2	操作数基本寻址方式	/122
4.4.3	寻址方式举例	/125
4.5	典型指令	/129
4.5.1	指令的分类	/129
4.5.2	基本指令系统的操作	/130
4.5.3	精简指令系统	/130
4.6	ARM 汇编语言	/133
	本章小结	/134
	习题	/134

## 第五章 中央处理器 /137

5.1	CPU 的功能和组成	/137
5.1.1	CPU 的功能	/137
5.1.2	CPU 的基本组成	/137
5.1.3	CPU 中的主要寄存器	/138
5.1.4	操作控制器与时序产生器	/139
5.2	指令周期	/140
5.2.1	指令周期的基本概念	/140
5.2.2	MOV 指令的指令周期	/141
5.2.3	LAD 指令的指令周期	/143
5.2.4	ADD 指令的指令周期	/145
5.2.5	STO 指令的指令周期	/145
5.2.6	JMP 指令的指令周期	/147
5.2.7	用方框图语言表示指令周期	/147
5.3	时序产生器和控制方式	/150
5.3.1	时序信号的作用和体制	/150
5.3.2	时序信号产生器	/151
5.3.3	控制方式	/153
5.4	微程序控制器	/153
5.4.1	微程序控制原理	/153
5.4.2	微程序设计技术	/160
5.5	硬连线控制器	/163
5.6	传统 CPU	/166
5.6.1	Intel 8088 CPU	/166
5.6.2	IBM 370 系列 CPU	/167
5.7	流水 CPU	/168
5.7.1	并行处理技术	/168
5.7.2	流水 CPU 的结构	/169

5.7.3	流水线中的主要问题	/171
5.7.4	奔腾 CPU	/173
5.8	RISC CPU	/177
5.8.1	RISC 机器的特点	/177
5.8.2	RISC CPU 实例	/178
	本章小结	/182
	习题	/183
<b>第六章</b>	<b>总线系统</b>	<b>/185</b>
6.1	总线的概念和结构形态	/185
6.1.1	总线的基本概念	/185
6.1.2	总线的连接方式	/186
6.1.3	总线的内部结构	/188
6.1.4	总线结构实例	/189
6.2	总线接口	/190
6.2.1	信息传送方式	/190
6.2.2	总线接口的基本概念	/192
6.3	总线的仲裁	/194
6.3.1	集中式仲裁	/194
6.3.2	分布式仲裁	/196
6.4	总线的定时和数据传送模式	/196
6.4.1	总线的定时	/196
6.4.2	总线数据传送模式	/199
6.5	HOST 总线和 PCI 总线	/199
6.5.1	多总线结构	/199
6.5.2	PCI 总线信号	/201
6.5.3	总线周期类型	/202
6.5.4	总线周期操作	/203
6.5.5	总线仲裁	/204
6.6	InfiniBand 标准	/204
6.6.1	InfiniBand 的体系结构	/204
6.6.2	InfiniBand 的通信协议	/205
	本章小结	/207
	习题	/207
<b>第七章</b>	<b>外围设备</b>	<b>/209</b>
7.1	外围设备概述	/209
7.1.1	外围设备的一般功能	/209
7.1.2	外围设备的分类	/209
7.2	磁盘存储设备	/210
7.2.1	磁记录原理	/210
7.2.2	磁盘的组成和分类	/212
7.2.3	磁盘驱动器和控制器	/213
7.2.4	磁盘上信息的分布	/215
7.2.5	磁盘存储器的技术指标	/216
7.3	磁盘存储设备的技术发展	/217
7.3.1	磁盘 cache	/217
7.3.2	磁盘阵列 RAID	/218
7.3.3	可移动存储设备	/219
7.4	磁带存储设备	/219
7.5	光盘和磁光盘存储设备	/220
7.5.1	光盘存储设备	/220
7.5.2	磁光盘存储设备	/222
7.6	显示设备	/223
7.6.1	显示设备的分类与有关概念	/223
7.6.2	字符/图形显示器	/224
7.6.3	图像显示设备	/226
7.6.4	VESA 显示标准	/227
7.7	输入设备和打印设备	/229
7.7.1	输入设备	/229
7.7.2	打印设备	/230
	本章小结	/232
	习题	/232
<b>第八章</b>	<b>输入输出系统</b>	<b>/234</b>
8.1	外围设备的速度分级与信息交换方式	/234
8.1.1	外围设备的速度分级	/234
8.1.2	信息交换方式	/235
8.2	程序查询方式	/237
8.3	程序中断方式	/240
8.3.1	中断的基本概念	/240
8.3.2	程序中断方式的基本 I/O 接口	/241
8.3.3	单级中断	/243
8.3.4	多级中断	/244
8.3.5	中断控制器	/247
8.3.6	Pentium 中断机制	/248
8.4	DMA 方式	/250
8.4.1	DMA 的基本概念	/250
8.4.2	DMA 传送方式	/251
8.4.3	基本的 DMA 控制器	/253
8.4.4	选择型和多路型 DMA 控制器	/255
8.5	通道方式	/258

8.5.1	通道的功能	/258	10.1.4	多处理机系统中并行性的发展	/289
8.5.2	通道的类型	/259	10.1.5	并行处理机的体系结构类型	/291
8.5.3	通道结构的发展	/260	10.2	超长指令字处理机	/292
8.6	通用 I/O 标准接口	/261	10.2.1	VLIW 处理机的特点	/292
8.6.1	并行 I/O 标准接口 SCSI	/261	10.2.2	VLIW 处理机的结构模型	/293
8.6.2	串行 I/O 标准接口 IEEE 1394	/262	10.2.3	典型处理机结构	/294
	本章小结	/264	10.3	多线程与超线程处理机	/296
	习题	/265	10.3.1	指令级并行与线程级并行	/296
<b>第九章</b>	<b>安腾高性能处理机体系结构</b>	<b>/267</b>	10.3.2	同时多线程结构	/298
9.1	高性能处理机体系结构的演变	/267	10.3.3	超线程处理机结构	/298
9.1.1	IA 体系结构的历史演变	/267	10.4	向量处理机	/300
9.1.2	英特尔 64 位处理机的两种体系结构	/268	10.4.1	向量处理的基本概念	/300
9.2	安腾体系结构的基本设计思想	/269	10.4.2	向量处理机的结构	/302
9.3	安腾指令系统结构	/271	10.4.3	并行向量处理机	/306
9.3.1	执行单元与指令类型	/272	10.5	多处理机	/308
9.3.2	安腾寄存器结构	/272	10.5.1	多处理机系统的分类	/308
9.3.3	安腾指令格式	/273	10.5.2	SMP 的基本概念	/309
9.3.4	安腾汇编语言格式	/275	10.5.3	SMP 的结构和实例	/309
9.4	指令级并行机制	/275	10.5.4	多处理机操作系统	/312
9.4.1	推断执行技术	/276	10.5.5	多处理机的 cache 一致性	/313
9.4.2	推测技术	/278	10.6	机群系统	/314
9.5	双核安腾处理机的组成	/281	10.6.1	机群系统的定义和特点	/314
9.5.1	双核安腾处理机的基本特性	/281	10.6.2	机群系统的体系结构	/315
9.5.2	双核安腾处理机的组织结构	/283	10.6.3	IBM SP2 系统	/317
	本章小结	/285	10.6.4	超级刀片系统	/318
	习题	/286		本章小结	/321
<b>第十章</b>	<b>并行体系结构</b>	<b>/288</b>		习题	/322
10.1	体系结构中的并行性	/288	<b>附录 A</b>	<b>《计算机组成与系统结构》(第五版·立体化教材) 配套教材与教学设备</b>	<b>/324</b>
10.1.1	并行性的概念	/288	<b>附录 B</b>	<b>计算机组成原理研究生入学统考大纲</b>	<b>/325</b>
10.1.2	提高并行性的技术途径	/289		<b>参考文献</b>	<b>/327</b>
10.1.3	单处理机系统中并行性的发展	/289			

# 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，是用数目字来表示数量的大小。数字计算机的主要特点是按位运算，并且不连续地跳动计算。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 数字计算机与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。

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、最经济和最快速的计算机，但是它的适应性很差。通用计算机适应性很大，但是牺牲了效率、速度和经济性。

通用计算机可分超级计算机、大型机、服务器、PC 机、单片机和多核机六类，它们的区别在于体积、简易性、功率损耗、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。而单片计算机是只用一片集成电路做成的计算机，体积小，结构简单，性能指标较低，价格便宜。介于超级计算机和多核机之间的是大型机、服务器、PC 机和单片机，它们的结构规模和性能指标依次递减。但是随着巨大规模集成电路的迅速发展，单片机、多核机等彼此之间的概念也在发生变化，因为今天的单片机可能就是



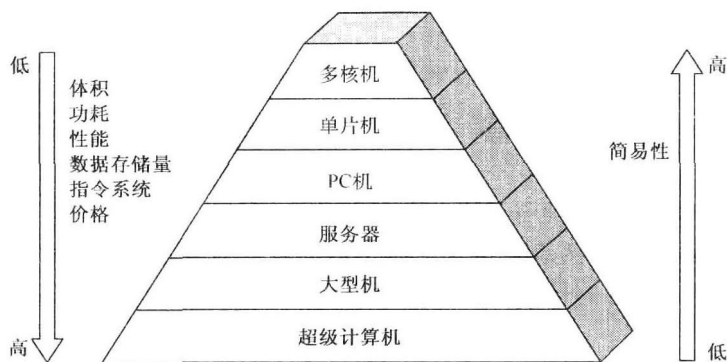


图 1.1 多核机、单片机、PC 机、服务器、大型机、超级计算机之间的区别

明天的多核机。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，用于测试或控制，成为计算机应用领域中最热门的产品。多核计算机是多于一个处理器（核）的计算机芯片。

## 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 18000 多个电子管，占用长度超过 30 米的房间，重量达 30 吨，而运算速度只有 5000 次/秒。从今天的眼光来看，这台计算机耗费既大又不完善，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，从使用器件的角度来说，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，可靠性较低。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，**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 1000 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。

计算机从第三代起，与集成电路技术的发展密切相关。LSI 的采用，一块集成电路芯片

上可以放置 1000 个元件，VLSI 达到每个芯片 1 万个元件，现在的 ULSI 芯片超过了 100 万个元件。1965 年摩尔观察到芯片上的晶体管数量每年翻一番，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。

在国际超级计算机 500 强排序中，中国 2004 年“曙光 400A”位居第 10；2009 年“星云号”位居第 2；2010 年“天河 1 号”位居第 1，运算速度达 2500 万亿次/秒。

### 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(磁芯)做成，每个磁芯直径约 1mm。这些小磁芯处在计算机内用三条细导线穿过网格板上。每个磁芯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。磁芯存储器速度相当快，读存储器中的一位只需 1 个微秒。但是磁芯存储器价格昂贵，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。更重要的在于工艺复杂，甚至手工制作。

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。一个相当于单个磁芯大小的芯片，包含了 256 位的存储器。这种芯片是非破坏性的，而且读写速度比磁芯快得多，读出一位只要 70 纳秒，但是其价格比磁芯要贵。

1974 年每位半导体存储器的价格低于磁芯。这以后，存储器的价格持续快速下跌，但存储密度却不断增加。这导致了新的机器比它之前的机器更小、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 年里改变了计算机的生命力。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 1KB、4KB、16KB、64KB、256KB、1MB、4MB、16MB、64MB、256MB 和现在的 1GB。其中  $1K=2^{10}$ ， $1M=2^{20}$ ， $1G=2^{30}$ 。每一代比前一代存储密度提高 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越来越少。表 1.2 列出了 Intel 公司微处理器的演化。

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奠基者。

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。这一发展趋势中的另一阶段是在 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。

表 1.2 Intel 微处理器的演化

(a) 20 世纪 70 年代的处理器					
型 号	4004	8008	8080	8086	8088
发布时间	1971	1972	1974	1978	1979
时钟速率	108kHz	108kHz	2MHz	5MHz, 8MHz, 10MHz	5MHz, 8MHz
总线宽度	4 位	8 位	8 位	16 位	8 位
晶体管数	2300	3500	6000	29000	29000
特征尺寸(微米)	10		6	3	3
可寻址存储器	640B	16KB	64KB	1MB	1MB
虚拟存储器	—	—	—	—	—

(b) 20 世纪 80 年代的处理器				
型 号	80286	386TM DX	386TM SX	486TM DX
发布时间	1982	1985	1988	1989
时钟速率	6~12.5MHz	16~33MHz	16~33MHz	25~50MHz
总线宽度	16 位	32 位	16 位	32 位
晶体管数	134 000	275 000	275 000	1 200 000
特征尺寸(微米)	1.5	1	1	0.8~1
可寻址存储器	16MB	4GB	16MB	4GB
虚拟存储器	1GB	64TB	64TB	64TB

(c) 20 世纪 90 年代的处理器				
型 号	486TM SX	Pentium	Pentium Pro	Pentium Ⅱ
发布时间	1991	1993	1995	1997
时钟速率	16~33MHz	60~166MHz	150~220MHz	200~300MHz
总线宽度	32 位	32 位	64 位	64 位
晶体管数	1.185 百万	3.1 百万	5.5 百万	7.5 百万
特征尺寸(微米)	1	0.8	0.6	0.35
可寻址存储器	4MB	4GB	64GB	64GB
虚拟存储器	64TB	64TB	64TB	64TB

(d) 21 世纪的处理器				
型 号	Pentium Ⅲ	Pentium 4	Itanium	Itanium 2
发布时间	1999	2000	2001	2002
时钟速率	450~600MHz	1.3~1.8GHz	733~800MHz	0.9~1GHz
总线宽度	64 位	64 位	64 位	64 位
晶体管数	9.6 百万	42 百万	25 百万	220 百万
特征尺寸(微米)	0.25	0.18	0.18	0.18
可寻址存储器	64GB	64GB	64GB	64GB
虚拟存储器	64TB	64TB	64TB	64TB

### 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所占的比率，用百分比表示。

**处理机字长** 指处理机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二进制位数。

**存储器容量** 存储器中所有存储单元的总数目，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，一般用字节数/秒表示。

**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 ( $f$ ) 叫 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 ( $T$ )， $T=1/f$ ，度量单位是微秒、纳秒。

**CPU 执行时间** 表示 CPU 执行一般程序所占用的 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：

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

**MIPS** 平均每秒执行多少百万定点指令数，用下式计算：

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

**FLOPS** 每秒执行浮点操作的次数，用来衡量机器浮点操作的性能。用下式计算：

$$\text{MFLOPS} = \text{程序中的浮点操作次数} \div \text{程序执行时间 (秒)}$$

**【例 1】** 对于一个给定的程序， $I_N$  表示执行程序中的指令总数， $t_{\text{CPU}}$  表示执行该程序所需的 CPU 时间， $T$  为时钟周期， $f$  为时钟频率 ( $T$  的倒数)， $N_C$  为 CPU 时钟周期数。设 CPI 表示每条指令的平均时钟周期数，MIPS 表示每秒钟执行的百万条指令数，请写出如下四种参数的表达式：

(1)  $t_{\text{CPU}}$     (2) CPI    (3) MIPS    (4)  $N_C$

**解** (1)  $t_{\text{CPU}} = N_C \times T = N_C f = I_N \times \text{CPI} \times T = \left( \sum_{i=1}^n \text{CPI}_i \times I_i \right) \times T$

(2)  $\text{CPI} = \frac{N_C}{I_N} = \sum_{i=1}^n \left( \text{CPI}_i \times \frac{I_i}{I_N} \right)$   $I_i/I_N$  表示  $i$  指令在程序中所占比例

(3)  $\text{MIPS} = \frac{I_N}{t_{\text{CPU}} \times 10^6} = \frac{f}{\text{CPI} \times 10^6}$

(4)  $N_C = \sum_{i=1}^n (\text{CPI}_i \times I_i)$

上式中， $I_i$  表示  $i$  指令在程序中执行的次数， $\text{CPI}_i$  表示  $i$  指令所需的平均时钟周期数， $n$  为指令种类。

**【例 2】** 用一台 50MHz 处理机执行标准测试程序，它包含的混合指令数和相应所需的平均时钟周期数如下表所示：

指令类型	指令数目	平均时钟周期数
整数运算	45000	1
数据传送	32000	2
浮点运算	15000	2
控制传送	8000	2

求有效 CPI、MIPS 速率、处理机程序执行时间  $t_{CPU}$ 。

$$\begin{aligned} \text{解 } \text{CPI} &= \frac{N_c}{I_N} = \sum_{i=1}^n \left( \text{CPI}_i \times \frac{I_i}{I_N} \right) \quad I_i/I_N \text{ 表示 } i \text{ 指令在程序中的比例} \\ &= \frac{45000 \times 1 + 32000 \times 2 + 15000 \times 2 + 8000 \times 2}{45000 + 32000 + 15000 + 8000} = 1.55 \text{ (周期/指令)} \end{aligned}$$

$$\text{MIPS} = \frac{f}{\text{CPI} \times 10^6} = \frac{50 \times 10^6}{1.55 \times 10^6} = 32.26 \text{ (百万条指令/秒)}$$

$$t_{CPU} = \frac{N_c}{f} = \frac{45000 \times 1 + 32000 \times 2 + 15000 \times 2 + 8000 \times 2}{50 \times 10^6} = 31 \times 10^{-4} \text{ (秒)}$$

## 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，不妨按以下方法把使用算盘进行解题的过程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，……依次类推。

根据表 1.3 所列的解题步骤，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

行数	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，用到了什么东西呢？

首先，用到了带横格且编有序号的纸，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，用到了我们人本身，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完全和我们人用算盘解题的情况相似，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**；而相当于人的大脑，能够自动控制整个计算过程的，称之为**控制器**。图 1.2 示出了数字计算机的主要组成部分，其中双线及箭头表示数据代码传送通路。

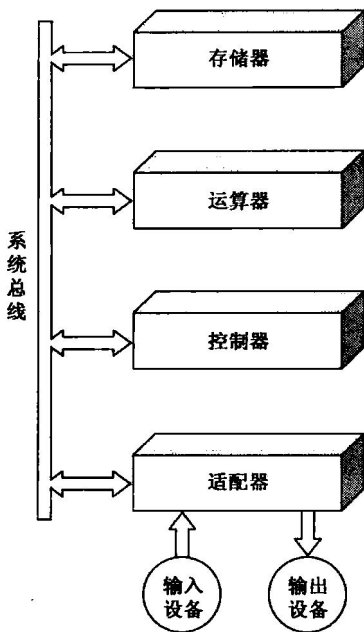


图 1.2 数字计算机的主要组成结构

图 1.2 示出了数字计算机的主要组成部分，其中双线及箭头表示数据代码传送通路。

### 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)。

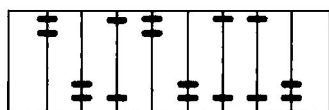


图 1.3 运算器结构示意图

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数。二进制的数是以 2 为基数来计数，也就是“逢二进一”。在二进制数中，只有 0 和 1 两个数目字。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。

二进制数的运算规律非常简单。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律简单，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。但是位数越多，所需的电子器件也越多，因此计算机的运算器长

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一样。图 1.4 为存储器的结构示意图。



图 1.4 存储器结构示意图

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 64KB，128MB。存储容量越大，表示计算机记忆储存的信息就越多。

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**。相对而言，半导体存储器称为**内存储器**，简称内存。

### 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。对于比较复杂的算题，计算机在运算前必须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。解题步骤的每一步，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。

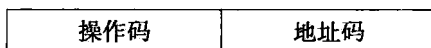
正如我们在横格纸上按行的序号记下解题步骤一样，计算机中为了顺利运算，也必须事先把程序和数据按地址安排到存储器里去。注意，程序中的指令通常按顺序执行，所以这些指令是顺次放在存储器里。这就相当于我们把表 1.3 所示的横格纸的内容原封不动地搬到存储器，因而所编的程序如表 1.4 所示。

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
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	机器停止工作
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:



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, 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。

表 1.5 指令的操作码定义

指令	操作码
加法	001
减法	010
乘法	011
除法	100
取数	101
存数	110
打印	111
停机	000

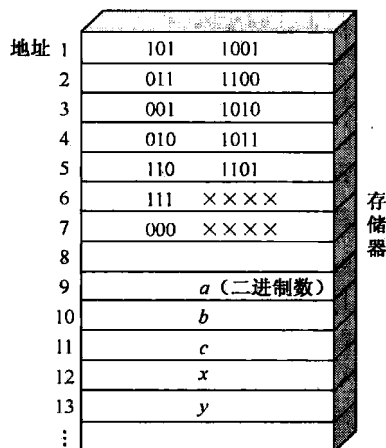


图 1.5 指令和数据在存储器中用二进制码存储



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全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·诺依曼型计算机的设计思想，也是机器自动化工作的关键。由于指令和数据放在同一个存储器，称为**冯·诺依曼结构**，如果指令和数据分放在两个存储器，称为**哈佛结构**。

一台计算机通常有几十种基本指令，从而构成了该计算机的**指令系统**。指令系统不仅是硬件设计的依据，而且是软件设计的基础。因此，指令系统是衡量计算机性能的一个重要标志。

### 3. 控制器的基本任务

由表 1.4 可知，计算机进行计算时，指令必须是按一定的顺序一条接一条地进行。控制器的基本任务，就是按照计算程序所排的指令序列，先从存储器取出一条指令放到控制器中，对该指令的操作码由译码器进行分析判别，然后根据指令性质，执行这条指令，进行相应的操作。接着从存储器取出第二条指令，再执行这第二条指令。依此类推。通常把取指令的一段时间叫做**取指周期**，而把执行指令的一段时间叫做**执行周期**。因此，控制器反复交替地处在取指周期与执行周期之中，如图 1.6 所示。每取出一条指令，控制器中的指令计数器就加 1，从而为取下一条指令做好准备，这也就是指令为什么在存储器中顺序存放的原因。

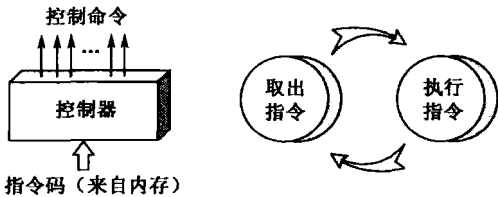


图 1.6 控制器功能示意

早期的计算机术语中，把运算器和控制器合在一起称为**中央处理器**，简称 CPU。目前存储器也放入到 CPU 中，称为**中央处理器**。

### 4. 指令流和数据流

由于计算机仅使用 0 和 1 两个二进制数字，所以使用“位”（bit）作为数字计算机的最小信息单位。当 CPU 向存储器送入或从存储器取出信息时，不能存取单个的“位”，而用 B（字节）和 W（字）等较大的信息单位来工作。一个“字节”由 8 位二进制信息组成，而一个“字”则至少由一个以上的字节组成。通常把组成一个字的二进制位数叫做**字长**。例如微型机的字长可以是 8 位，也可以达到 64 位。

由于计算机使用的信息既有指令又有数据，所以计算机字既可以代表指令，也可以代表数据。如果某字代表要处理的数据，则称为**数据字**；如果某字为一条指令，则称为**指令字**。

我们已经看到，指令和数据统统放在内存中，从形式上看，它们都是二进制数码，似乎很难分清哪些是指令字，哪些是数据字。然而控制器完全可以区分开哪些是指令字，哪些是数据字。一般来讲，取指周期中从内存读出的信息流是**指令流**，它流向控制器；而在执行器周期中从内存读出的信息流是**数据流**，它由内存流向运算器。例如，图 1.5 中从地址 1~7 号单元读出的信息流是指令流，而从地址 9~12 号单元读出的信息流是数据流。显然，某些