



“十二五”普通高等教育本科国家级规划教材  
国家优秀教学团队/国家特色专业教学成果  
国家精品在线开放课程教学成果  
系统能力培养教学成果  
新工科建设·计算机类系列教材

# 计算机组成原理与 汇编语言程序设计

## (第5版)

徐洁 叶娅兰 主编

电子工业出版社

Publishing House of Electronics Industry

北京 · BEIJING

## 内 容 简 介

本书为“十二五”普通高等教育本科国家级规划教材。全书从微体系结构层、指令系统层、汇编语言层三个层次，以及CPU、存储系统、输入/输出系统及其互连三大系统出发，建立整机的概念，并体现软硬结合的思想。全书共7章，分为三篇，系统介绍基础知识（绪论、计算机中的信息表示）、计算机系统结构（微体系结构层——CPU组织、指令系统层、汇编语言层）、存储系统与输入/输出系统（存储系统、输入/输出系统）。本书为教师免费提供电子教案。

本书系统全面，实例丰富，适合作为高等学校计算机及相关专业教材，也可作为IT技术人员的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有，侵权必究。

## 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 / 徐洁, 叶妍兰主编. —5 版. —北京: 电子工业出版社, 2023.2

ISBN 978-7-121-44991-8

I. ① 计… II. ① 徐… ② 叶… III. ① 计算机体系结构—高等学校—教材 ② 汇编语言—程序设计—高等学校—教材 IV. ① TP303 ② TP313

中国国家版本馆 CIP 数据核字（2023）第 017576 号

责任编辑：章海涛

印 刷：

装 订：

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编：100036

开 本：787×1092 1/16 印张：24.5 字数：620 千字

版 次：2023 年 2 月第 1 版

印 次：2023 年 2 月第 1 次印刷

定 价：69.80 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888, 88258888。

质量投诉请发邮件至 [zlts@phei.com.cn](mailto:zlts@phei.com.cn)，盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

本书咨询联系方式：192910558 (QQ 群)。

# 前 言

CO

AL

本书为“十二五”普通高等教育本科国家级规划教材，曾入选电子工业部“九五”规划教材和普通高等教育“十一五”国家级规划教材。

本书将传统教学计划中的两门相互依赖程度较深的主干课程《计算机组成原理》和《汇编语言程序设计》有机地组织为一门课程，目的是围绕机器指令级（指令系统层）及紧密相关的微操作级（微体系结构层）和汇编语言级（层），将计算机内部工作机制与编程求解问题结合，从而更好地体现软硬结合的计算机系统思想。本书将输入/输出（I/O）系统作为《计算机组成原理》和《汇编语言程序设计》的结合点，既包含硬件接口设计，又包含I/O驱动程序等软件设计，体现了两部分的紧密关联度和目前汇编语言程序设计的主要应用场合。在保持前几版的基本结构、主要内容和风格的基础上，我们对原书各章内容进行了修订，以便更好地与现代计算机系统相吻合。

下面介绍本书的编写思路、各章内容简介和第5版修订内容。

## 一、编写思路

本书涵盖了 ACM/IEEE-CS 中“AR 计算机组织与体系结构”知识体的 5 个核心知识单元：AR2 数据的机器级表示，AR3 汇编机器组织，AR4 存储系统组织和结构，AR5 接口和通信，AR6 功能组织。

本书还涉及汇编语言层的 PF 程序设计基础知识领域的一个核心知识单元——PF1 程序设计基本结构。因此，本书适用于“计算机学科教学计划”推荐的 16 门核心课程中的《计算机组成基础》课程。

多年来，我们跟踪分析了国外经典教材与国内优秀教材，这些教材一般可分为两种体系：“分层体系”和“功能部件组成体系”。本书的框架体系采用这两种体系相结合的方案，从三个层次（微体系结构层、指令系统层和汇编语言层）和三大子系统（CPU、存储系统、输入/输出系统及其互连）出发建立整机概念。本书第二篇按照微体系结构层、指令系统层和汇编语言层三个层次，讨论计算机系统的组成，从而在 CPU 级建立整机概念。第三篇介绍存储系统、输入/输出系统，帮助读者在系统级上建立整机概念。

本书以一个 CPU 教学模型机的设计为例，从寄存器级描述 CPU 的内部逻辑组成结构，在指令流程和微操作两个层次上深入分析 CPU 的工作原理；从存储原理、高速缓冲 - 主存 - 外存三级存储体系、虚拟存储器阐述存储系统的工作机制与组织；从输入/输出接口控制方式、

总线、外部设备阐述输入/输出系统的原理与结构。同时，本书以实际 Intel 80x86、Pentium 系列微处理器为背景，阐述和分析其 CPU 的微体系结构、指令系统、汇编语言、主存、高速缓存（Cache）、虚拟存储器、总线、I/O 接口、I/O 驱动程序和 I/O 软件调用等。

而且，本书介绍了典型精简指令集计算机（Reduced Instruction Set Computer, RISC）微处理器 MIPS 和 ARM 的微体系结构和指令系统，它们与复杂指令集计算机（Complex Instruction Set Computer, CISC）的 80x86 微处理器形成了对比。

## 二、各章内容简介

全书由基础知识、计算机系统分层结构、存储系统和输入/输出系统 3 篇共 7 章组成，体现了下述教学思路。

第一篇基础知识，包括第 1~2 章，介绍计算机系统组成的基本概念和信息表示。

第 1 章仍然强调计算机的两个重要基本概念：信息数字化、存储程序工作方式。在介绍计算机系统的硬件、软件组成后，引入了现代计算机的分层结构模型，自下而上分为 5 层：微体系结构层、指令系统层、操作系统层、汇编语言层和面向问题的语言层，本书第二篇内容就是根据其中的三层来组织的。第 1 章还对计算机的工作过程、特点、性能指标和发展史等做了介绍。

第 2 章讲述数据信息和指令信息的表示，包括带符号数、小数点及字符的表示方法，以及指令格式、寻址方式和指令类型，并介绍 Pentium II 指令格式、RISC 基本概念、典型 RISC 微处理器 MIPS 和 ARM 的指令系统。

第二篇计算机系统结构，包括第 3~5 章，分别从微体系结构层、指令系统层和汇编语言层深入讨论计算机系统的组成和工作机制。

第 3 章介绍微体系结构层，将 CPU 作为整体来讨论，弱化运算方法与运算器；主要以一个教学模型机为例，从寄存器级描述 CPU 的内部逻辑组成，在指令流程和微操作两个层次上分析 CPU 的工作原理。根据当前计算机的发展趋势，更强调组合逻辑控制方式，对微程序控制方式只做一般原理性介绍，并以 MIPS 和 ARM 为典型例子，说明 RISC 的微体系结构。

第 4 章以 80x86 CPU 为背景讨论指令系统层，将 80x86、Pentium 系列 CPU 的微体系结构做了类比，用归纳和概括的方式介绍该系列 CPU 的寻址方式和指令系统。本章作为第 3 章与第 5 章的桥梁，由 CPU 模型过渡到实际机器 80x86，并为学习汇编语言奠定了相关基础。

第 5 章汇编语言层，以 80x86、Pentium 系列 CPU 为背景，精练地阐述了汇编语言的基础知识、伪指令与宏指令、汇编语言程序设计基本技术。

第三篇包括第 6~7 章，讨论存储系统和输入/输出（I/O）系统。

第 6 章简介各种存储器存储信息的基本原理，芯片级主存储器的逻辑设计方法。按三级存储体系（高速缓存、主存、外存）分别介绍其工作原理，并从物理层和虚拟层讨论存储系统的组织；以 Pentium 为实例，深入分析其主存储器、Cache 和虚拟存储器的工作机制；还引入了高级 DRAM、磁盘冗余阵列 RAID、并行存储技术等。

第 7 章介绍输入/输出系统。采用硬软结合的方式，既讨论硬件接口与 I/O 设备的逻辑组成及工作原理，也介绍软件调用方法与相应的 I/O 程序设计；详细阐述 I/O 接口的主要控制方式：直接程序控制方式、程序中断方式和 DMA 方式的工作原理，并以 80x86 为背景深入分析了三种控制方式的接口组成和 I/O 驱动程序设计；系统介绍总线分类、标准、时序和实例；简

介了常用 I/O 设备：键盘、鼠标器、打印机和液晶显示器的工作原理，并以键盘为例分析其驱动程序的设计；另外，引入典型外设接口 ATA 接口和 SCSI 接口。

### 三、第 5 版主要修订内容

第 1 章增加了国产芯片龙芯和华为海思麒麟的发展史，使读者对我国自主研制 CPU 芯片的情况和应用场合有所了解，还对计算机的分类和提高 CPU 性能技术进行了补充。

目前，国际上市场份额较大的主流 CPU 架构有 3 种：Intel 80x86、ARM 和 MIPS。本书的实际机器背景就是 CISC 的 Intel 80x86，因此在第 2 章增加了典型的 RISC 微处理器 MIPS 和 ARM 的指令系统，删除了非主流的 SPARC 指令系统；第 3 章相应增加了 MIPS R4000 和 ARM7 的微体系结构，删除了 SPARC 微体系结构，以利于读者在了解 RISC 的 MIPS 和 ARM 的同时与 CISC 的 80x86 进行对比。

第 6 章修订了 Cache 基本原理内容，增加了 AMD Opteron 数据 Cache 的组织结构并深入分析了该数据 Cache 读、写数据的过程；增加了半导体 ROM 和 Flash 存储器的存储原理、U 盘和固态硬盘的内容，补充和修订了高级 DRAM 内容；删除了光盘存储器的相关内容。

第 7 章根据微机的总线结构发展演变过程，重新梳理了其单总线、双总线和多总线的结构变化和特点；增加了 80x86 中断指令及其应用场合，对中断接口举例和 DMA 接口举例内容重新进行了梳理和修订，使其内容更具可读性；删除了通道和 IOP 的相关内容。

本书内容丰富，覆盖与融合了传统“计算机组成原理”“微机原理”和“汇编语言程序设计”课程的主要内容，知识的系统性和教学的实用性更强，也使学时数得到了合理压缩。

由于篇幅所限，与本书有关的一些教学资料将以电子文档方式提供给读者，欢迎免费下载 (<http://www.hxedu.com.cn>)。教学资料包括：80x86 指令系统一览表，DOS 系统功能调用 (INT 21H)，BIOS 功能调用等。

本课程的参考教学时数为 64~80，全书内容可能比教学时数允许的稍多一些，教师可以选取或让学生自学，部分实例可随技术发展而更新。本书为教师免费提供电子教案，欢迎到 <http://www.hxedu.com.cn> 注册后下载。

参与本书编写工作的有徐洁、叶娅兰、李晶晶和鲁珂，全书由徐洁、叶娅兰主编并负责全书的组织和统稿。重庆大学袁开榜教授担任主审，他认真、仔细地审阅了全稿，提出了许多宝贵的修改意见。本书编辑们热情、专业和细致的工作态度保证了教材的编辑质量和水平。教材的编写还得到了北京航空航天大学杨文龙教授、电子科技大学龚天富教授的热情指导和帮助，以及电子科技大学计算机学院领导和老师们的热情支持。在此，谨向所有给予我们支持和帮助的人们表示衷心的感谢。

书中还会存在错误与不足之处，恳请读者与同行给予批评指正。

作者 E-mail：[xujie@uestc.edu.cn](mailto:xujie@uestc.edu.cn)，或者加入 QQ 群：192910558。

作 者  
2022 年于成都



# 目 录

CO

AL

## 第一篇 基础知识

第 1 章 绪论 .....	3
1.1 计算机的基本概念 .....	3
1.1.1 存储程序工作方式 .....	4
1.1.2 信息的数字化表示 .....	5
1.2 计算机系统的硬件和软件组成 .....	7
1.2.1 计算机硬件系统 .....	7
1.2.2 计算机软件系统 .....	10
1.3 层次结构模型 .....	13
1.3.1 从计算机系统组成角度划分层次结构 .....	13
1.3.2 从语言功能角度划分层次结构 .....	15
1.3.3 软件和硬件在逻辑上的等价 .....	16
1.4 计算机的工作过程 .....	17
1.4.1 处理问题的步骤 .....	17
1.4.2 指令执行过程 .....	18
1.5 计算机的特点和性能指标 .....	19
1.5.1 计算机的特点 .....	19
1.5.2 计算机的性能指标 .....	20
1.6 计算机的发展与应用 .....	22
1.6.1 计算机的发展历程 .....	22
1.6.2 计算机性能提高的技术 .....	26
1.6.3 计算机应用举例 .....	29
习题 1 .....	30
第 2 章 信息表示 .....	32
2.1 数值型数据的表示 .....	32
2.1.1 带符号数的表示 .....	32
2.1.2 定点数与浮点数 .....	37

2.2	字符的表示.....	42
2.2.1	ASCII .....	42
2.2.2	Unicode 编码 .....	43
2.2.3	汉字编码简介 .....	44
2.3	指令信息的表示.....	45
2.3.1	指令格式.....	45
2.3.2	常用寻址方式.....	49
2.3.3	指令类型 .....	54
2.3.4	Pentium II 指令格式.....	57
2.3.5	RISC 概述 .....	59
2.3.6	MIPS 指令系统.....	60
2.3.7	ARM 指令系统.....	64
	习题 2.....	70

## 第二篇 计算机系统结构

第 3 章	微体系结构层 .....	73
3.1	CPU 的组成和功能 .....	74
3.1.1	CPU 的组成 .....	74
3.1.2	指令执行过程 .....	77
3.1.3	时序控制方式 .....	79
3.1.4	指令流水线 .....	80
3.2	ALU 和运算方法 .....	84
3.2.1	ALU 介绍 .....	84
3.2.2	定点数运算方法 .....	88
3.2.3	浮点数运算方法 .....	95
3.2.4	十进制数加、减运算 .....	97
3.3	CPU 模型机的组成及其数据通路 .....	98
3.3.1	基本组成 .....	98
3.3.2	数据传输 .....	101
3.4	组合逻辑控制器原理 .....	102
3.4.1	模型机的指令系统 .....	103
3.4.2	模型机的时序系统 .....	106
3.4.3	指令流程 .....	108
3.4.4	微命令的综合与产生 .....	114
3.4.5	小结 .....	115
3.5	微程序控制器原理 .....	115
3.5.1	微程序控制概念 .....	115
3.5.2	微指令编码方式 .....	117
3.5.3	微程序的顺序控制 .....	119

3.5.4 微指令格式 .....	121
3.5.5 典型微指令举例——模型机微指令格式 .....	121
3.6 典型 RISC 处理器微体系结构 .....	122
3.6.1 MIPS R4000 的微体系结构 .....	122
3.6.2 ARM7 的微体系结构 .....	126
习题 3 .....	128
<b>第 4 章 指令系统层 .....</b>	<b>130</b>
4.1 80x86 CPU .....	131
4.1.1 8086/8088 CPU .....	131
4.1.2 80386/80486 CPU .....	133
4.1.3 Pentium 系列 CPU .....	135
4.2 80x86 CPU 的寄存器和主存储器 .....	139
4.2.1 80x86 CPU 的寄存器 .....	139
4.2.2 80x86 的主存储器 .....	143
4.3 80x86 CPU 指令系统 .....	154
4.3.1 80x86 寻址方式 .....	154
4.3.2 80x86 CPU 指令分类 .....	161
4.3.3 传送类指令 .....	162
4.3.4 算术运算类指令 .....	169
4.3.5 逻辑类指令 .....	177
4.3.5 串操作类指令 .....	182
4.3.6 处理机控制类指令 .....	186
习题 4 .....	187
<b>第 5 章 汇编语言层 .....</b>	<b>192</b>
5.1 汇编语言层概述 .....	192
5.2 汇编语句语句格式 .....	194
5.3 80x86 宏汇编语言数据、表达式和运算符 .....	196
5.3.1 常数 .....	196
5.3.2 变量 .....	196
5.3.3 标号 .....	200
5.3.4 表达式与运算符 .....	201
5.4 80x86 宏汇编语言伪指令 .....	204
5.4.1 符号定义语句 .....	204
5.4.2 处理器选择伪指令 .....	205
5.4.3 段结构伪指令 .....	206
5.4.4 段组伪指令 .....	210
5.4.5 内存模式和简化段定义伪指令 .....	211
5.4.6 定位和对准伪指令 .....	212
5.4.7 过程定义伪指令 .....	213

5.4.8 包含伪指令 .....	214
5.4.9 标题伪指令 .....	214
5.5 宏指令.....	214
5.6 汇编语言程序设计基本技术.....	220
5.6.1 程序设计步骤 .....	220
5.6.2 顺序程序设计 .....	222
5.6.3 分支程序设计 .....	224
5.6.4 循环程序设计 .....	231
5.6.5 子程序设计 .....	237
5.6.6 系统功能子程序的调用 .....	247
5.6.7 汇编语言程序的开发 .....	248
习题 5.....	254

### 第三篇 存储系统和输入/输出系统

第 6 章 存储系统.....	261
6.1 存储系统概述.....	261
6.1.1 存储器的分类 .....	262
6.1.2 主存的主要技术指标 .....	263
6.2 存储原理.....	264
6.2.1 半导体存储器的存储原理 .....	264
6.2.2 磁表面存储器的存储原理 .....	269
6.3 主存储器的组织.....	273
6.3.1 主存储器的逻辑设计 .....	273
6.3.2 主存储器与 CPU 的连接 .....	277
6.3.3 Pentium CPU 与存储器组织 .....	279
6.3.4 高级 DRAM.....	282
6.4 高速缓冲存储器.....	285
6.4.1 Cache 的工作原理 .....	286
6.4.2 Cache 的组织 .....	287
6.4.3 Pentium II CPU 的 Cache 组织 .....	293
6.5 外部存储器.....	294
6.5.1 硬磁盘存储器 .....	295
6.5.2 U 盘和固态硬盘 .....	300
6.6 物理存储系统的组织.....	301
6.6.1 存储系统的层次结构 .....	301
6.6.2 磁盘阵列 .....	302
6.6.3 多体交叉存取技术 .....	305
6.7 虚拟存储系统的组织.....	306
6.7.1 概述 .....	306

6.7.2	虚拟存储器的组织方式	306
6.7.3	Pentium CPU 支持的虚拟存储器	309
6.7.4	存储管理部件	311
习题 6		313
<b>第 7 章</b>	<b>输入/输出系统</b>	<b>315</b>
7.1	输入/输出系统概述	315
7.1.1	主机与外围设备间的连接方式	315
7.1.2	I/O 接口的功能和分类	317
7.1.3	接口的编址和 I/O 指令	320
7.2	直接程序控制方式	322
7.3	程序中断方式	324
7.3.1	中断的基本概念	324
7.3.2	中断的过程	330
7.3.3	80x86 中断指令	332
7.3.4	中断接口模型	334
7.3.5	中断接口举例	339
7.4	DMA 方式	343
7.4.1	DMA 方式的一般概念	343
7.4.2	DMA 工作过程	345
7.4.3	DMA 接口组成	346
7.4.4	DMA 控制器编程及应用	349
7.5	总线	356
7.5.1	总线的功能与分类	356
7.5.2	总线标准及信号组成	358
7.5.3	总线操作时序	361
7.5.4	典型总线举例	363
7.6	典型外设接口	364
7.6.1	ATA 接口	364
7.6.2	SCSI 接口	365
7.7	I/O 设备与 I/O 程序设计	365
7.7.1	概述	366
7.7.2	键盘	368
7.7.3	鼠标器	372
7.7.4	打印机	373
7.7.5	液晶显示器	375
习题 7		376
<b>附录 A</b>	<b>ASCII 字符表</b>	<b>377</b>
<b>参考文献</b>		<b>378</b>



# 第1章

CO

AL

## 绪 论

通常所讲的计算机，全称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。这个定义中包含两个重要的基本概念：信息数字化和存储程序工作方式。本书一开始就强调它们，希望读者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、软件组成后，将分别从系统内部的有机组成和程序设计语言功能的角度，介绍两种常用的层次结构模型；再通过对解题过程的描述，说明计算机的应用方式与工作过程；在上述知识的基础上，分析计算机的特点，说明其性能指标的含义；最后，简要介绍 Intel、龙芯和华为微处理器的发展，以及提高计算机性能的技术和主要应用领域。

### 1.1 计算机的基本概念

初学者提出的第一个问题常常是：计算机是什么？简单地讲，计算机是一种能够存储程序，能够自动连续地执行程序，对各种数字化信息进行算术运算或逻辑运算的快速工具。我们先对这个定义进行初步解释。首先，计算机是能够运算的设备，运算可以分为两大类：算术运算和逻辑运算。算术运算的对象是数值型数据，以四则运算为基础，许多复杂的数学问题可通过相应的算法最终分解为若干四则运算。逻辑运算用来解决逻辑型问题，如信息检索、判断分析和决策等。所以，我们常将计算机的工作泛称为对信息进行运算处理。那么，计算机中的信息用什么形式来表示呢？简单地讲，是用数字代码来表示各类信息，所以称为数字计算机。计算机又是怎样对这些数字化的信息进行运算处理呢？它采用的是一种存储程序工作方式，即把编写好的程序输入计算机并存储，然后通过连续、快速地执行程序实现各种运算处理。为了存储程序与数据，需要存储器；为了进行运算处理，需要运算器；为了输入程序、数据、输出运算结果，需要输入设备和输出设备；控制器则对计算机的各项工作进行控制管理。

这些要点是由计算机技术的先驱者冯·诺依曼首先提出的，他在 1945 年提出了数字计算机的若干设计思想，被称为冯·诺依曼体制，这是计算机发展史上的一个里程碑。采用冯·诺依曼体制的计算机被称为诺依曼机。几十年来，计算机的体系结构发生了许多演变，但冯·诺依曼体制的核心概念仍沿用至今，绝大多数实用的计算机仍属于冯·诺依曼机。冯·诺依曼体制中那些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，即信息（数据和指令）的数字化。
- ② 采用存储程序工作方式，即事先编制程序，事先存储程序，自动、连续地执行程序。
- ③ 由存储器、运算器、控制器、输入设备、输出设备五大部件组成计算机硬件系统。

本节先阐述其中的两点：存储程序工作方式和信息的数字化表示。硬件组成部分则放在 1.2 节与系统结构一起讨论。

### 1.1.1 存储程序工作方式

计算机的工作最终体现为执行程序，采用存储程序工作方式，这是冯·诺依曼体制中最核心的思想，体现了用计算机求解问题的过程，包括三点含义。

#### 1. 事先编制程序

为了用计算机求解问题，需要事先编制程序，也就是将求解问题的处理过程用程序来实现。程序规定计算机需要做哪些事，按什么步骤去做，还包括需要运算处理的原始数据，或者规定计算机在什么时候从输入设备获得数据。一件事往往要分很多步骤去完成，要求计算机硬件在一步中执行的操作命令称为一条指令，如加法指令。计算机最终执行的程序，其形态就是指令序列，即若干指令的有序集合，每一步将执行一条指令。换句话说，我们预先编好的程序最终变成：指令序列和有关的原始数据。

#### 2. 事先存储程序

编好的程序经由输入设备送入计算机，存放在存储器中。编写的程序是用字符书写的，通过键盘将字符变成二进制编码，然后输入计算机。二进制编码中的每一位不是 0 就是 1，可以保存在存储器中。

最早的电子计算机是靠许多开关和拔插连接线来体现程序的，被称为台外程序式，意思是程序不在计算机内部。后来，冯·诺依曼机采用了事先存储程序的工作方式，这有重要意义。

#### 3. 自动、连续地执行程序

程序存储在存储器中后，启动计算机并运行程序，计算机就可以依照一定顺序从存储器中逐条读取指令，按照指令的要求执行操作，直到运行的程序执行完毕。原则上，程序运行不需通过人工操作逐条读取指令，所以是自动、连续地执行程序，使得计算机可以高速运行。当然，有些工作本身要求以人机对话方式进行，如通过计算机进行查询，计算机通过显示屏向用户询问：需查询什么项目？用户通过键盘或鼠标进行选择。这种情况要求计算机分段执行程序，中间允许用户进行人工干预。所以，计算机在自动、连续地执行程序的同时，往往允许使用者以外部请求方式进行干预。

上面描述了计算机的基本工作方式。冯·诺依曼机的这种工作方式被称为控制流驱动方

式，是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。在这个过程中，逐步发出的控制信息成为一种控制信息流，简称控制流，它是驱动计算机工作的因素。而依次处理的数据信息则成为一种数据信息流，简称数据流，它是被调用的对象，或者说这是被驱动的部分。

## 1.1.2 信息的数字化表示

前面谈到，计算机中的信息可以分为两大类：控制信息和数据信息。随着程序的逐步执行，依次取出的指令代码序列，以及在此基础上产生的微命令等，就成为控制信息流，它们是控制计算机工作的有关信息。而依据指令要求依次取出的数据和运算处理的结果等成为数据信息流，它们是计算机加工处理的对象。数据可以分为两大类：数值型数据和非数值型数据。前者有数值大小及正负之分，如四则运算的对象等；后者指字符、文字、图像、声音等一类信息，以及条件、命令、状态等逻辑信息。这就需要解决一个问题：怎样表示上述信息？

现在广泛使用的计算机是电子式数字计算机。“电子式”指计算机的主要部件由电子电路构成，计算机内传输与处理的信息是电子信号。那么，为什么称为数字计算机呢？因为计算机中的信息（控制流、数据流）都采用数字化表示方法，简单地讲，它有两层含义。

### 1. 在计算机中各种信息用数字代码表示

下面通过一组例子来说明如何用数字代码表示各类信息，这是了解计算机工作原理的又一重要基础，希望大家熟练掌握，能够举一反三。

#### 【例 1-1】用二进制数字代码表示数值的大小。

用一组数字代码表示一个数值型数据，其中每一位数字只有两种，不是 0 就是 1，逢 2 进位，所以称为二进制。数的正负数符也用一位数字代码表示，称为符号位，如约定符号位为 0 表示正数，符号位为 1 表示负数。例如，11001 表示 -9。

#### 【例 1-2】用数字编码表示中、西文字符。

例如，01000001 表示 A，01000010 表示 B，就像发电报时邮局将汉字编为一组数字电报码一样。以字符为基础可以表示范围广泛的各种文字，编写程序时所用的程序设计语言也是用字符组成的。

#### 【例 1-3】用数字代码表示图像。

与字符相比，图像信息变化多，哪里亮哪里暗，是随时变化的。但是我们可以将一幅图像细分为许多像点（或像素），用这些像点的组合逼近真实图像。如果分得足够细，即点数足够多，那么在人的视觉中这幅由许多像点组成的图像几乎是连续的。相应地，用 1 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。再按照一定的扫描规律，如逐线从左到右地扫描，就可以将这些像点的信息以数字代码形式组织，并存入计算机，这样就可以用计算机对图像进行处理了。

下面的例子说明怎样用数字代码表示逻辑型信息。

#### 【例 1-4】用数字代码表示机器指令。

程序在计算机中的最终（可执行）形态是指令序列，按照事先约定的指令格式，每条指令用一组数字代码表示。一条指令往往分为几个字段，如操作码字段、地址字段等，我们约定用

不同的编码表示不同的指令含义，如操作码的编码含义是：0000 表示传输，0001 表示相加，0010 表示相减等。

### 【例 1-5】用数字代码表示设备的状态。

计算机在控制打印机、显示器等设备时，常常需要根据设备的工作状态来决定操作。这些状态可以被抽象化，然后用数字代码表示。例如，用 00 表示设备现在空闲，用 01 表示设备忙，用 10 表示设备已完成一次操作，等等。

## 2. 用数字型电信号表示数字代码

从物理实现层次看，数字代码需要用电信号去体现，这样才能用电子电路部件实现信息的传送和运算处理。电信号分为两类：模拟信号和数字信号。

模拟信号是用信号的某些参量去模拟信息，如用电信号的幅值模拟数值的大小，所以称为模拟信号、模拟量。许多物理量，如压力、温度等，需要先通过传感器变为模拟信号，再转换为数字信号，才可以用计算机处理。

数字信号是这样一种信号，它的单个数字信号仅取有限的几种状态，一般只取两种状态，如高电平或低电平、有脉冲或无脉冲，这两种状态可用数字代码 1 或 0 来表示，称为二值逻辑。相应地，数字信号也有两种形式：电平信号和脉冲信号。依靠多位数字信号的组合，可以表示多位数字代码。换句话说，1 位数字信号表示 1 位数字代码。

例如，在计算机传输数据时，常用多根传输线同时传输，每根线传送 1 位，这称为并行传送。如果某一根线的电平为高，则该位为 1；若另一根线的电平为低，则该位为 0。各线之间相互分离，可独立传输电平信号。

用数字代码可以表示各种信息。用数字信号表示数字代码，就是信息数字化的含义。计算机是用来处理信息的，可以处理的信息类型极其广泛。要了解计算机的工作原理，并在今后工作中灵活地进行设计，首先需要深刻理解和熟练掌握信息的数字化表示方法。

采用数字化方法表示信息的优点如下。

① 抗干扰能力强，可靠性高。因为每位数字的取值非 1 即 0，相应地，表示数字的电信号也只需两种状态。假定电源为 +5 V，用高电平表示 1，用低电平 (0 V) 表示 0，则在 1 与 0 之间有比较大的差别，即使受到一定干扰也能够区分是 0 还是 1。

② 依靠多位数字信号的组合，在表示数值时可以获得很宽的表示范围和很高的精度。理论上，对信息表示的位数增加并无限制，这取决于使用者愿意付出的硬件代价。

③ 数字化信息容易存储，信息传输也比较容易实现。因为每位数字非 0 即 1，相应地，在物理实现上也只需取两种可能的极端状态来表示 0 或 1，因而可以有多种方法来体现，如开关连通或断开，晶体管导通或截止，电容上有电荷或无电荷，磁性材料的正向磁饱和或反向磁饱和，磁化状态的变或不变，等等。相应地，双稳态触发器可以用来存储信息，电容存储的电荷可以用来存储信息。

④ 可以表示的信息类型和范围极其广泛，几乎没有限制，这在前面已经举例说明。

⑤ 能用逻辑代数等数字逻辑技术进行信息处理，这形成了计算机硬件设计的基础。

计算机的各项具体操作最终是用数字逻辑电路来实现的，可以称为处理功能逻辑化。由于采用二进制数字代码来表示各类信息，因此我们能用种类非常有限的几种逻辑单元（与、或、非门）构造出变化无穷的计算机系统。

## 1.2 计算机系统的硬件和软件组成

计算机系统的组成可分为两大范畴：硬件和软件。计算机硬件是指系统中可触摸到的设备实体，如运算器、控制器、存储器、输入设备、输出设备，以及将它们组织为一个计算机系统的总线、接口等。计算机软件是指系统中的各类程序和文件，它们在计算机中体现为一些不能直接触摸到的二进制信息，所以被称为软件。下面分别介绍计算机系统的硬件组成和软件组成，帮助读者初步建立整机的概念。

### 1.2.1 计算机硬件系统

早期，冯·诺依曼将计算机的硬件组成为五大部件。几十年来，计算机硬件系统已有了许多重大变化。首先，现在采用的大规模及超大规模集成电路可将运算器和控制器集成在一块芯片上，称为中央处理器（Central Process Unit, CPU），是负责执行程序、实现运算处理、控制整个系统的部件。相应地，原来的运算器现在作为 CPU 中的运算部件（又称为算术逻辑部件），与控制器之间的界限已不像原来那样分明。其次，存储器分为高速缓存（Cache）、主存储器、外存储器三个层次。其中，高速缓存常集成在 CPU 内部，作为 CPU 的一部分，也可以在 CPU 外再设置一级高速缓存。通常，CPU 与主存储器合在一起，被称为主机，主存储器（简称主存）就是因为位于主机之内而得名，有的文献称之为内存储器（内存）。位于主机之外的磁盘、光盘、磁带等，则作为外存储器（外存）。输入设备的任务是将外部信息输入主机，输出设备则是将主机的运算处理结果或其他信息从主机输出。但从信息传输和控制的角度看，它们并无多大区别，不过是传输方向不同而已。有些设备还兼有输入、输出两种功能，所以在描述系统结构时常将它们合称为输入/输出设备，简称 I/O（Input/Output）设备。

图 1-1 用框图的形式描述了一种简单的单总线硬件系统结构，将 CPU、主存、I/O 设备等都画为一个框，在框内标注其名称。部件、设备之间的连接线也采用示意方式来表现，暂不画出全部连线，也没有具体标明各条连线的细节。这种画法可大大简化细节，以突出其系统结构。

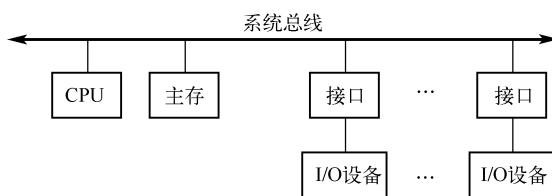


图 1-1 单总线计算机硬件系统结构

下面以图 1-1 为基础，先分别说明各组成部件的功能和相应特性，再介绍以总线为基础构成整机系统的方法。

#### 1. CPU

CPU 是硬件系统的核心部件，负责读取并执行指令，也就是执行程序。冯·诺依曼体制的要点之一即采用二进制代码表示数据和指令，那么，怎样区分它们呢？在 CPU 中有一个程序计数器（Program Counter, PC），它存放着当前指令所在存储单元的地址。如果程序是顺序执行，在读取一个或连续几个存储单元的指令代码后，PC 的内容就加 1 或加几，以指出下一条

指令的地址；如果程序需要转移，就将转移地址送入 PC。因此，PC 像一个指针，指引着程序的执行顺序。指令和数据都采用二进制代码表示，程序可以按照 PC 中的地址信息去读取指令，再按照指令给出的操作数地址去读取数据。后面讲到主存储器时会解释地址的概念。

CPU 内有一个或多个算术逻辑部件（Arithmetic Logic Unit, ALU），通常按照指令的要求将有关数据送入 ALU，进行指定的算术或逻辑运算，然后将运算结果送到主存单元，或暂存在 CPU 内的寄存器中。

CPU 的主要部分是控制器（又称为控制部件），它的任务是控制整个系统的工作，决定在什么时候、根据什么、发出什么命令、做什么操作。例如，控制器控制着从主存中读取指令，根据指令代码，分时地发出一些最基本的控制信号即微命令，控制存储器的读写、ALU 的运算处理、数据的输入/输出等。

按照产生微命令的方式不同，控制器可分为两大类：组合逻辑控制器、微程序控制器。

组合逻辑控制器完全靠若干组合逻辑电路（即硬件）产生微命令序列，将指令代码、状态信息、时序信号等输入这些组合逻辑电路，电路将分时产生相应的微操作命令。这种控制器又称为硬连逻辑控制器，其优点是形成微命令的速度快，常用于精简指令系统计算机（Reduced Instruction Set Computer, RISC）。

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；当 CPU 执行指令时，通过读取并执行对应的一段微程序，产生微命令序列，以完成指定的操作。微程序控制方式比较规整，硬件代价较小，易于扩充功能，但速度较慢。

在微型计算机系统中，CPU 又被称为微处理器。并行计算机系统通常包括若干 CPU，CPU 又被称为处理器或处理机（Processor）。

## 2. 主存储器

我们总是希望计算机系统中的存储器容量大，存取速度快，但这往往是有矛盾的，所以存储器常常分为几级。其中，主存储器用半导体存储器构成，工作速度较高，也有一定的存储容量。外存储器为主存提供后援，由磁盘、光盘、磁带等构成，其存储容量很大，但速度较主存储器低。以后要介绍的高速缓冲存储器、主存储器和外存储器，组合成一个完整的多级存储系统，使得整个系统既有很大的存储容量，而 CPU 可以高速地访问存储器。

主存储器用来存放需要执行的程序及需要处理的数据，能由 CPU 直接读出或写入。

主存储器划分为许多单元，通常每个单元存放 8 位二进制数，称为 1 字节。每个单元都有唯一的编号，称为存储单元地址，简称地址。向主存储器送出某个地址编码，就能根据地址选中对应的一个单元。这就像一幢学生宿舍楼，分为若干房间，每个房间可住 8 个人，每个房间的房号相当于存储单元地址。程序的最终（可执行）形态是指令序列，通常按照执行顺序依次存放在连续的存储单元中，通过程序计数器 PC 提供的指令地址，就可以逐条地读取指令。一条指令按其长度不同可存放在一个或相邻的几个单元中。有的指令需要处理的数据（又称为操作数）存放在主存的一个或相邻的几个单元中，指令执行时，就提供地址去寻找对应单元，从中读取操作数。通过以上描述，读者可以初步了解主存中的信息存放情况，可见主存储器的一项重要特性是：能按地址（单元编号）存放或读取内容，就是允许 CPU 直接编址访问，通常以字节为编址单位。对主存储器来说，寻找存储单元（寻址）的依据是地址码，所存取的内容是指令或数据。

### 3. 外存储器

外存储器用来存放那些需要联机存放，但暂不执行的程序和数据，当需要运行它们时再由外存调入主存。例如，硬盘中存放着多个应用的软件，当前暂时只用得着其中的一个，我们先将它调入主存，其余软件仍存放在硬盘上。又如，信息管理系统保存的数据很多，可将它们存放在磁盘中，只将当前需要查询的部分调入主存，以后再调换。这样，主存的容量就不需要很大，可以做到速度比较快。而由磁盘、光盘等构成的外存储器容量很大，可为整个系统提供后援支持，其速度要求可以比主存低一些。

由于外存储器不由 CPU 直接编址访问，也就是说，不需要按字节地从外存储器读取或写入，因此外存储器中的内容一般按文件的形式进行组织。一个文件常分解为若干数据块，可以包含许多字节的信息。用户按文件名进行调用，CPU 找到该文件在外存中的存放位置，以数据块为单位进行读写。

从功能上，外存储器是整个存储系统的一部分，是一种存储器。从信息传输的角度，外存储器又是一种输入/输出设备。将磁盘中的文件调入主存时，磁盘是输入设备；将主存中的内容以文件的形式写入磁盘时，磁盘是输出设备。

### 4. 输入/输出设备

计算机系统大多配备了键盘、鼠标、显示器、打印机等常规输入/输出（I/O）设备，有的还配备了图形、音频输入/输出设备。

输入设备用来将计算机外部的信息输入计算机。外部信息的形式可能有多种，因此输入设备常需进行信息形式的转换，将外部信息变换为计算机能识别和处理的形式。例如，用程序设计语言编写的程序，是以字符为基础，通过按键将它变为机器能够识别和存储的二进制代码。

输出设备将计算机的处理结果以我们能看得懂的形式输出。目前，多数计算机将信息输出到显示器，关机后显示信息会丢失，所以显示器被称为软拷贝设备。打印机可将有关信息打印在纸上，长期保管，所以被称为硬拷贝设备。

### 5. 总线

怎样将 CPU、主存、多台 I/O 设备连接成整机系统呢？一种简单的方法是采用总线结构。所谓总线，是指一组能为多个部件分时共享的信息传输线。如图 1-1 用一组系统总线连接 CPU、主存及多台 I/O 设备，它们之间可以通过系统总线传输信息，连接在总线上的部件都可以使用这组总线。注意，某时刻只能有一个部件或设备向总线发送数据，如果有两个或两个以上的部件同时向总线发送数据，就会产生冲突，使数据混乱，这就是分时共享的含义。但总线上的数据既可以只向某个部件发送，也可以同时向几个部件发送。

系统总线可分为三组，即地址总线、数据总线和控制总线，如某微型计算机的系统总线有 32 位地址线、32 位数据线、20 多根控制信号线。CPU 如果需要访问主存，就向地址总线送出地址码以选择某个主存单元；通过数据总线送出数据，写入主存；或从主存读出数据，通过数据总线送入 CPU 的寄存器。大部分控制信号是由 CPU 提供的，它们通过控制总线送往主存和 I/O 设备；有些信号是 I/O 设备提供的，其中有些也送往 CPU。

### 6. 接口

一台计算机系统需连接哪些 I/O 设备，这要根据该系统的应用场合而定，因此通过系统总

线连接的设备，其类型与数量都应当可以扩充。某种型号的计算机系统，其系统总线往往是标准的，也就是说，有多少根地址线、数据线，有哪些控制信号线，每个信号的名称及作用等，都是规定好的。但是它所连接的 I/O 设备却是类型各异。怎样使标准的系统总线与各种类型的设备相连接呢？这就需要在系统总线与 I/O 设备之间设置一些逻辑部件，约定它们之间的界面，这种逻辑部件称为 I/O 接口，在微型计算机中又称为适配卡。在实际工作中，常常需要我们去设计各类接口，需要具备接口的有关知识和设计方法。

概括地说，计算机硬件系统是由三大子系统——CPU、存储系统（包括高速缓存、主存和外存）、输入/输出系统（包括输入/输出设备和接口）以及连接它们的总线构成的。本书后续将深入讨论这些子系统的内部结构和功能。1.3 节将介绍计算机的存储程序工作方式和指令的执行过程。

## 1.2.2 计算机软件系统

计算机软件就是程序，规定计算机如何去完成某个任务，是某种算法的体现。计算机保存着一些以文件形式组织的信息，如对系统的说明，对编程工具与运行环境的说明，为用户提供帮助的提示与其他参考信息等，所以有人将软件的定义描述得更广泛，即软件是程序和文件。

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看，软件可以分为系统软件和应用软件两大类。下面结合计算机的工作，介绍这些软件的主要内容，使读者了解计算机是如何工作的。

### 1. 系统软件

系统软件是一组为使计算机系统良好运行而编制的基础软件。从软件配置角度，系统软件是用户所购置的计算机系统的一部分，是一种软设备，是提供给用户的系统资源。当我们购买一台计算机系统时，除了购买硬件还要购买一些系统软件，有时软件的费用可能超过硬件费用。从功能角度，系统软件是负责计算机系统的调度管理，提供程序的运行环境和开发环境，向用户提供各种服务的一类软件。下面介绍常见的系统软件及其作用。

#### (1) 操作系统

操作系统是软件系统的核心，如 UNIX、Linux、Windows 和 Mac。操作系统负责管理和控制计算机系统的硬件、软件资源及运行的程序，合理地组织计算机的工作流程，是用户与计算机之间的接口，为用户提供软件的开发环境和运行环境。下面进行初步解释，后续课程如操作系统课程会深入讨论。

一个完备的操作系统包括 CPU 调度管理、存储器管理、I/O 设备管理、文件管理、作业管理等几部分。在计算机系统中，大量的信息以文件形式组织并保存，操作系统的文件管理模块提供了信息管理机构。用户程序及其所需的数据常以作业的形式存放在外存储器中，由操作系统进行调度管理，调入主存储器后方可由 CPU 运行。如果计算机系统具备多道程序运行环境，就需要操作系统对 CPU 的分配和运行实施有效管理，还需要为各道程序分配内存空间，并使它们互不干扰，即提供内存保护。I/O 设备管理模块负责为用户程序分配 I/O 设备，提供良好的人机界面，其中含有对各种设备的驱动程序，完成有关的 I/O 操作。所以说操作系统控制和管理着计算机的硬件、软件资源，合理地组织计算机的工作流程。

在配置操作系统后，用户可以通过操作系统提供的用户界面去使用、操作计算机。例如，曾经广泛使用的单用户操作系统 PC-DOS 为用户提供了两种界面，一种是通过键盘操作执行的 DOS 命令，另一种供用户程序调用的系统功能调用。所以说，操作系统提供了计算机与用户之间的接口。现在的操作系统为用户提供了更丰富、更方便的人机图形界面。

在用户程序的开发和执行过程中，可能需要用到许多其他的系统软件程序，即软件资源，它们是作为文件被操作系统管理调度的。我们所编制的用户程序也作为文件纳入操作系统的管理之下。所以，操作系统为用户提供了软件的开发环境和运行环境。正如人们常常说的，某应用软件是在 Windows 环境下开发、运行的。

## (2) 编译程序与解释程序

计算机硬件能够直接识别的是数字代码，所以让计算机硬件执行的基本命令，如传输、加法、减法等，必须用 0、1 这样的数字编码来表示。由硬件执行的程序的最终形态是由若干指令组成的序列，即指令是程序（可执行形态）的基本单位。通常，一条指令规定了一种基本操作（如传输、加、减），并提供操作数地址或直接提供操作数，这些信息都由数字代码表示。一台计算机可以执行的各种指令的集合，称为这种计算机的指令系统。显然，不同的机型往往具有不同的指令系统，以及相应的指令格式约定，所以机器指令代码又称为机器语言，即面向特定机器结构的一种内部语言。

如果直接用机器语言（0、1 代码）编制程序，将非常不便，于是人们想到用一些约定的符号，如英文缩写的字符串，去表示操作含义、操作数、地址等，这就产生了汇编语言。汇编语言是一种用符号表示的与机器指令基本对应的程序设计语言，专属于某种机型或某种系列机，其他计算机不能直接使用，所以是一种面向机器结构的程序设计语言，不是通用语言。

为了便于编制程序，现在所使用的绝大多数编程语言是高级程序设计语言，这是一些面向用户、与特定机器属性分离的语言。高级程序设计语言与机器指令之间没有直接的对应关系，所以可以在各种机型中通用，编程者使用高级语言也不必了解具体的机器指令系统及其他硬件属性。高级程序设计语言需要遵循一定的严格语法规规定与格式，才能为语言处理程序（编译、解释）所识别。现在已出现了许多种高级语言，各具特色，还在继续发展之中。

机器语言是机器内部使用的、用数字代码表示的指令代码，面向某一特定机型，可由硬件直接识别并执行。汇编语言是一种用符号表示的，面向某一特定机型的程序设计语言，它的指令语句与机器指令一一对应。高级程序设计语言则是面向用户，与特定机器属性相分离的程序设计语言，具有通用性。

大多数情况下，用户采用高级程序设计语言编写程序，个别情况采用汇编语言编写程序。用这些程序设计语言编写出的程序称为源程序，它们由一些语句组成。将源程序输入计算机后，计算机先执行一种语言处理程序，将源程序转换为机器语言代码序列，即机器语言程序，然后由计算机硬件执行这些用机器语言代码表示的指令序列，从而完成用户程序的执行过程。这种语言处理程序也是必需的系统软件。

语言处理方式有两种类型：解释、编译。

解释方式是边解释边执行，为此需要一种针对某种程序设计语言的解释程序（又称解释器）作为系统软件的组成部分之一。将源程序输入计算机后，启动并执行相应的解释程序，它的作用是逐步分析源程序中的语句，按照源程序描述的过程，执行一个与此等价的机器语言指令序列，直到整个源程序都被扫描一遍，并被解释执行完毕为止。这有点像口译外语的情形，

边说边翻译。解释方式适用于比较简单的程序设计语言，如 BASIC。它的优点是支持人机对话方式的程序设计，可以边执行边修改；需要的主存空间较小。但是这种方式的执行速度较慢，不能解释那些前后关联较多、较难理解的程序设计语言。

大多数程序设计语言采用编译方式。将源程序输入计算机后，先启动并执行相应的编译程序（又称编译器），将源程序全部翻译成目标程序（目标代码）的机器语言指令序列。执行时，计算机将直接执行目标程序，不再需要源程序与翻译程序。因此，这种编译方式有点像笔译，得到完整的译文后就可以不要原文与译者了。编译需要的主存空间比解释方式多，既要容纳源程序，又要容纳一个比较大的编译程序；花费的时间也要长；但运行用户程序时，所需的主存空间比较小，执行速度也较快。

将汇编语言源程序转换为机器代码的目标程序的过程也是一种编译，被称为“汇编”，相应的翻译程序被称为汇编程序（又称为汇编器）。它的逆过程被称为“反汇编”，即将用机器代码表示的目标程序（指令序列）反向编成用汇编语言描述的程序。为利于二次开发，在剖析一些已有的重要软件时常常需要进行反汇编。

将一种程序设计语言的源程序转换为不同机器语言的目标程序，需要不同的编译程序或解释程序。例如，Pentium 机上的 C 语言编译程序就不同于 ARM 机上的 C 语言编译程序。

### （3）各种软件平台

为了方便用户，常将开发及运行过程中所需的各种软件集成为一个综合的软件系统，称为软件平台，这已成为软件开发中的一种重要趋势。我们在构建一个应用系统时，首先要考虑：需要购买怎样的一套硬件系统，配置什么软件平台。

有些软件平台是以某种操作系统为核心，增加一些常用的基本功能，特别是人机界面功能，如窗口软件、提示系统等。有些平台属于通用的开发环境型，以某种高级语言编译系统为核心，加上输入程序、编辑工具、调试工具，以及一些常用的基本功能程序模块。有些软件平台面向某种应用领域的开发、运行需要，如信息管理领域所需要的数据库管理系统，为用户提供一种数据库语言用于编制数据管理软件，并提供一些数据库系统所需的基本功能，如用于多媒体制作的多媒体平台、用于中文处理的软件平台等。

## 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，无所不在，因此应用软件不胜枚举，它一般包括：科学计算类、数据处理类、自动控制类、计算机辅助设计类、人工智能类等。有关内容在 1.5 节中介绍。

当然，对系统软件与应用软件的划分并不是一成不变的，一些具有通用价值的应用软件也可以纳入系统软件中，作为一种软件资源提供给用户。前面提到的许多软件平台包含的一些常用基本功能模块，就其功能来说，属于应用软件，但就系统配置来说，它们又可算作系统软件的一部分。

从功能角度，系统软件是负责系统调度管理，提供开发环境和运行环境，向用户提供各种服务的一类软件，而应用软件是用户在各自应用领域中为解决各类问题所编写的程序。从配置角度，系统软件是用户购置的系统资源之一，而应用软件是用户自身开发的、直接面向应用需要的程序。

## 1.3 层次结构模型

上面介绍了计算机系统的硬件、软件组成。现在按照层次结构的观点去分析这些硬件、软件组成之间的关系，从而建立计算机系统的整机概念。计算机系统是相当复杂的，所以我们在分析、设计、开发时往往采用层次结构的观点和方法，也就是将系统分成若干层，逐层分析、设计、构建。当我们购买一台计算机时，可以逐层配置软件资源或扩展其功能。分析计算机的工作原理时，可以根据自己工作的需要选取某一层，如汇编语言层或某种高级语言层，然后观察、分析它是如何工作的。在开发软件时，也常常是分成若干层、若干模块去进行，如面向用户的人机界面、用户所需的功能模块、公用的基础性软件、面向机器硬件的物理层等。

为达到不同的目的，通常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。

### 1.3.1 从计算机系统组成角度划分层次结构

图 1-2 是从计算机系统组成的角度来划分的一种层次结构模型，给出了构成计算机系统的硬件层和多个软件层，以及它们之间的关系。图 1-2 分层描述了计算机系统的主要硬件、软件组成，自下而上表明了设计和构建一台计算机时的逐层生成过程，每层都在下一层的基础上增加功能。

#### 1. 微体系结构层

微体系结构层是具体存在的硬件层次，我们看到的不是由大部件（即 CPU、存储系统、输入/输出系统及互连机构）构成的计算机，而是更细微的机器结构。

由于 CPU 是计算机的核心部件，因此微体系结构层主要从寄存器级观察 CPU 的结构，分析 CPU 执行指令的详细过程。在这一层，我们看到的是 CPU 内部的数据通路，即一些寄存器和算术逻辑部件 ALU 相连构成的数据通路。数据通路的基本功能是传输和加工数据，如选择一个或两个寄存器的内容作为 ALU 的操作数，将它们进行运算，如相加；然后将结果存回某寄存器。在一些机器上，这些功能是由微程序产生的控制信号控制的，相应的控制部件被称为微程序控制器；而有些机器是直接由硬件产生的控制信号来控制的，相应的控制部件被称为组合逻辑控制器。

微体系结构层执行机器指令，可以看作指令系统层指令的解释器。在由微程序控制数据通路的计算机上，微程序就是上一层指令的解释器，通过数据通路逐条对指令进行取指、译码和执行。例如，对加法 ADD 指令，将首先取出指令进行译码分析，然后找到操作数送入寄存器，由 ALU 求和，最后存结果到指定地方，如某存储器单元。而在硬件直接控制数据通路的计算机上，执行的步骤与此类似，但是由硬件直接解释执行指令，并不存在一个真正的程序来解释上一层的指令。

微体系结构层的内容即 CPU 结构和功能，是本课程的一个重点。

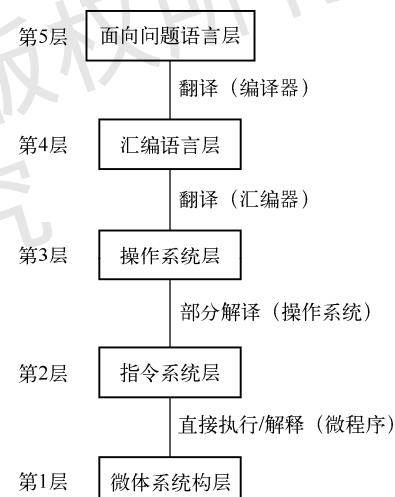


图 1-2 从计算机系统组成角度划分的层次结构模型

如果从硬件组成的角度进一步分析微体系结构层中寄存器、ALU、控制电路等部件的构成，那么可知几种数字逻辑单元（与、或、非门）组合成了这一层的部件。由于门电路的结构和工作原理是属于“数字逻辑电路”课程的内容，本书不再赘述。

## 2. 指令系统层

原则上，指令系统层是机器语言程序员眼中看到的计算机，当然现在人们并不使用机器语言编程。指令系统层位于微体系结构层之上，是一个抽象的层次，其主要特征是指令系统。

指令系统（又称指令集）是指一台计算机所能执行的全部指令的集合。其指令是由微体系结构层的微程序解释执行或硬件电路直接执行的。每台计算机都有自己的“指令系统参考手册”，用来描述各种指令的格式和功能。

指令系统是计算机软件与硬件之间的一种接口。硬件系统的基本任务是实现指令系统所规定的各种指令功能，而各种程序只有最终转化为用机器语言（即代码表示的指令序列）才能被硬件执行。尽管现在已广泛应用各种高级语言编程，但需通过编译器或解释器将高级语言程序转换为硬件可以识别与执行的机器指令序列。

指令系统层定义了硬件和编译器之间的接口，是一种硬件和编译器都能理解的语言。一方面，指令系统表明了一台计算机具有哪些硬件功能，是硬件逻辑设计的基础。因此，在指令系统层，应该定义一套在当前和将来的技术条件下能够高效率实现的指令集，从而使高效率的设计可用于今后的若干代计算机中。另一方面，指令系统层应该为编译器提供明确的编译目标，使编译结果具有规律性和完整性。

对指令系统层的讨论也是本课程的重点之一。

## 3. 操作系统层

从程序员的观点，操作系统是一个在指令系统层提供的指令和特性上又增加了新指令和特性的程序。这一层有新的指令集，有不同的存储器结构，有同时运行两个或多个程序的能力，以及其他特性。

尽管操作系统层和指令系统层都是抽象层次，但它们有重要的区别。操作系统层指令集是系统程序员完全可用的指令集，包括几乎所有的指令系统层的指令和操作系统层增加的新指令。这些新指令被称为系统调用（System Call），如 DOS 操作系统的系统功能调用“INT 21H”用于设备、文件和目录等管理，Linux 操作系统的系统调用“fork()”用来创建一个进程。一个系统调用使用一条新指令调用一个预先定义好的操作系统服务，这样效率很高。一个典型的系统调用是从一个文件中读取数据。

操作系统层增加的系统调用是由运行在指令系统层上的操作系统解释执行的。当一个用户程序执行一个系统调用时，如从一个文件中读取数据，操作系统将一步步执行这个调用。但是，那些与指令系统层相同的本层指令将直接交给微体系结构层执行，而不是由操作系统执行。换句话说，操作系统层的新增指令由操作系统解释，而其他指令由微体系结构层直接执行，因此操作系统层又被称为“混合层”。

操作系统层并不是为普通程序员的使用而设计的，主要是为支持高层所需的解释器或翻译器运行而设计的。

操作系统层包含的具体内容会在操作系统课程中详细介绍。

#### 4. 汇编语言层

微体系结构层可看作指令系统层指令的解释器，使指令系统的功能得以实现。但是，直接用机器指令代码编程是非常困难的，而让微体系结构层直接执行高级语言也不是好办法。因此，人们为计算机设计了汇编语言层，它位于指令系统层、操作系统层与面向问题语言层之间。从这一层看去，每种计算机都有一套自己的汇编语言、解释它的汇编器，以及相应的程序设计与开发方法。

汇编语言层以及上层是提供给解决应用问题的应用程序员使用的。低三层提供的机器语言都是二进制代码，适合机器执行，但不容易被人理解。从汇编语言层开始，其提供的语言是人们能理解的单词和缩略语。汇编语言实际就是“符号化”的机器语言，每条汇编指令语句都对应一条机器语言指令，它是面向机器结构的语言。用汇编语言编写的程序先由汇编器翻译成机器语言程序，然后由微体系结构层解释执行。

汇编语言层支持上层的方法与低层不同，通常用的是编译，而指令系统层和操作系统层主要用的是解释。

#### 5. 面向问题语言层

面向问题语言层的语言通常是为解决现实问题的应用程序员使用的，这些语言通常称为高级语言。目前已开发的高级语言有几百种，业界比较知名的有 Visual BASIC、C、C++、Java、Python 和 Fortran 等。用这些语言编写的程序一般先由编译器翻译成指令系统层和操作系统层语言，偶尔也有解释执行的。例如，用 Java 语言写的程序可以采用解释执行。

在某些情况下，面向问题语言层由针对某特别领域（如符号数学）的解释器组成。解释器提供该领域专业人员熟悉的运算和数据以解决该领域的问题。

总之，用分层方式设计和分析计算机系统可以忽略一些无关紧要的细节，使复杂的问题变得更容易理解。本书第二篇各章就是按图 1-2 中层次模型的微体系结构层、指令系统层、汇编语言层顺序组织的，并分别讨论计算机系统的组成。

### 1.3.2 从语言功能角度划分层次结构

如果将计算机功能描述为“能执行用某些程序设计语言编写的程序”，那么用户看到的就是图 1-3 所示的语言功能层次模型。计算机硬件的物理功能是执行机器语言，称为机器语言物理机，从这一级看到的是一台实际的机器。而用户看到的是能执行某种语言程序的虚拟机，即通过配置某种语言处理程序后所形成的一台计算机。

与机器语言最接近的是汇编语言，它的基本成分是与指令系统一一对应的用助记符描述的汇编语句。与算法、数学模型甚至自然语言接近的编程语言称为高级语言，具有较强的通用性，如多种通用的高级程序设计语言。针对某些特定应用领域与用户，也可使用某种专用语言，它们一般面向应用，如所要求解的问题。

大多数计算机都是先将用程序设计语言编写的程序翻译为机器语言，然后才能执行，一般是直接翻译为机器语言。但当高级语言较复杂时也可能分级编译，即先翻译为层次低些的某种中间语言，再将中间语言进一步翻译为机器语言，如图 1-3 中虚线所示。

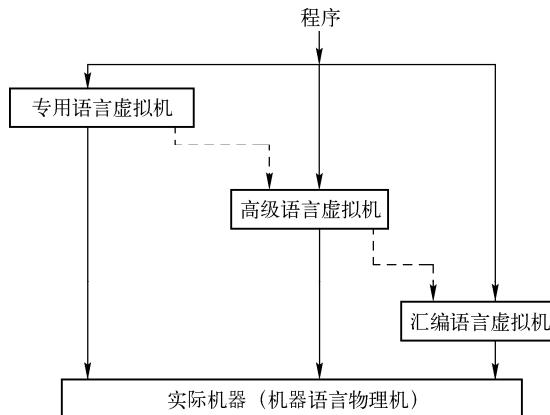


图 1-3 从语言功能角度划分的层次结构模型

所谓虚拟机，通常是指通过配置软件（如某种语言的编译器或解释器）扩充机器功能后所形成的一台计算机。实际硬件在物理功能级上并不具备这种机器功能，因而称为虚拟机。虚拟机概念是计算机设计的又一重要策略，将提供给用户的功能抽象出来，使其脱离具体的物理机器，这有利于让用户摆脱真实物理机细节的束缚，获得超越物理机的功能。

目前，Internet 上广泛使用 Java 虚拟机（Java Virtual Machine，JVM）。为了使 Java 程序能在不同的计算机上运行，Sun 公司定义了一种称为 Java 虚拟机的虚拟体系结构，有 32 位字组成的内存，能执行 226 条指令。大多数指令很简单，只有一些比较复杂，需要多次内存循环。

Sun 公司提供了一个将 Java 语言程序编译成 Java 虚拟机指令序列（又称 Java 执行程序，JVM 程序）的编译器，以实现程序的跨平台运行，还实现了能解释执行 Java 程序的解释器。该解释器用 C 语言编制，可在任何一台有 C 语言编译器的计算机上运行。实际上意味着可在世界上所有计算机上编译和运行。因此，要使一台计算机能够执行 Java 程序，所要做的就是找一个能在该平台（如 Pentium II 和 Windows、SPARC 和 UNIX 等）上运行的 JVM 解释器及相关的支持程序和库。而且，大多数 Internet 浏览器已包含了 JVM 解释器，能容易地运行万维网页中的 Java Applet。这些 Applet 主要是为网页提供语音和动画。

解释 JVM 程序的速度要慢一些。运行 Applet 或其他新收到的 JVM 程序的另一条途径是先将它编译成用户计算机能执行的机器指令代码，再运行编译好的程序。这就要求浏览器中有一个 JVM 到该机器语言的编译器，并能在需要时激活它。这种随时编译的编译器被称为 JIT (Just In Time)，已被普遍使用。

除了软件实现的 JVM (JVM 解释器和 JIT 编译器)，Sun 和其他公司还设计了硬件的 JVM 芯片，即设计出可直接执行 JVM 程序的 CPU，不再需要一层软件来解释或用 JIT 来编译 JVM 程序。这种体系结构的新芯片 picoJava-I 和 picoJava-II 已经在嵌入式系统市场上出现。

### 1.3.3 软件和硬件在逻辑上的等价

计算机系统以硬件为基础，通过软件扩充其功能，并以执行程序方式体现其功能。一般来说，硬件只完成最基本的功能，而复杂的功能则通过软件来实现。但是，硬件与软件之间的界面，如功能分配关系，常随技术发展而变化。有许多功能既可以直接由硬件实现，也可以在硬件支持下靠软件实现，对用户来说在功能上是等价的，称为软件和硬件在功能上的逻辑等价。

例如，乘法运算可由硬件乘法器实现，也可在加法器与移位器支持下由乘法子程序实现。又如，JVM 既可以用软件（JVM 解释器或 JIT 编译器）实现，也可以用硬件（JVM 芯片）实现。

从设计者角度，指令系统是硬件与软件之间的界面。硬件的基本任务是识别与执行指令代码，因此指令系统所规定的功能一般可由硬件实现。所编制的程序最终需要转换成指令序列，由指令代码表示才能执行，因此指令系统是编程的基础（直接或间接）。问题是如何设计指令系统，选择恰当的软件、硬件功能分配，这取决于所选定的设计目标、系统的性能价格比等因素，并与当时的技术水平有关。

早期曾采用的一种技术策略是硬件软化。刚出现数字计算机时，人们依靠硬件实现各种基本功能，随后为了降低造价，只让硬件实现较简单的指令系统，如加、减、移位与基本逻辑运算功能，依靠软件实现乘、除、浮点运算等更高级一些的功能，这导致了在当时条件下小型计算机的出现，它们结构简单又具有较强的功能，推动了计算机的普及与应用。

随着集成电路技术的飞速发展，我们可以在一块芯片上集成相当强的功能模块，于是出现了另一种技术策略：软件硬化，即将原来依靠软件才能实现的一些功能，改由大规模、超大规模集成电路直接实现，如浮点运算、存储管理等。这样，系统将有更高的处理速度，在软件支持下具有更强的功能。

如果说系统设计者必须关心软件、硬件之间的界面，即哪些功能由硬件实现，哪些由软件实现，那么用户更关心系统究竟能提供哪些功能。至于这些功能是由硬件还是软件实现，在逻辑功能上则是等价的，只是执行速度有差别而已。

## 1.4 计算机的工作过程

前面在介绍计算机系统的硬件、软件组成时，已经从多个角度涉及计算机的工作方式。现在我们沿着用户应用计算机求解问题这一线索，分析计算机的工作过程。

### 1.4.1 处理问题的步骤

处理问题的步骤可归纳为：系统分析，建立数学模型，设计算法，编写应用程序，编译为目标代码，由硬件执行目标程序。

有的用户使用计算机处理工作时并不需要自己编写程序，只需要面对显示器，用键盘、鼠标调用有关功能即可。例如，学籍管理系统包含了有关的多个应用程序模块，并提供良好的人机界面，学校的管理人员打开系统后，可以从屏幕上读到该系统的使用方法，可从有关菜单中选取所需的功能，进行查询、输入、删除、修改、统计等工作。又如，多媒体制作人员使用多媒体制作平台设计广告，只需输入文字、图像、声音等基本素材，调用系统的有关功能对这些素材进行编辑加工，制作动画，就可以生产出所希望的广告。学籍管理系统的用户只需使用，基本上不需要程序设计；多媒体制作人员是设计者，只需运用计算机系统的已有功能，不需要自己编程。

当然，应用系统的研制就复杂得多了。本节主要讨论应用计算机处理（求解）问题的基本步骤。不同应用领域的处理方法会有所不同，这里讨论共性的内容。

## 1. 系统分析

如果构造一个比较复杂的应用系统，首先要进行需求分析：确定该系统应具备哪些功能并据此划分功能模块；了解需存储、处理哪些数据、数据量、调用数据时的流向等。然后根据需求分析结果选择硬件平台和软件平台。如果准备购置的平台不能完全满足需要，可能需要自己设计一些硬件部件和系统软件模块。总体设计中的这些分析工作常称为系统分析。

## 2. 建立数学模型、设计算法

应用计算机求解、处理问题的方法被泛称为算法。早期计算机主要用于数学计算，那时的算法主要是指一些求解数学方程的公式之类的方法。后来计算机广泛应用于各种信息处理，算法的具体含义就推广为处理各种问题的方法，如信息的检索方法、调度策略、逻辑判别等。

如果需处理的问题比较复杂，包含多项分析、计算，或多种类型的数据信息，就需要建立相应的数学模型，可能是一组算法的有机组合，如一种桥梁应力分析的数学模型；也可能是一些数据信息的组织结构，如某种信息管理系统的数学模型；也可能是一组逻辑判断规则的有机组合，如某种决策系统等。

## 3. 编写应用程序

在建立数学模型并设计算法后，就可以选择合适的程序设计语言和开发工具，着手编写应用程序，然后在相应的调试环境下进行调试和修改。

## 4. 编译为目标代码

在大多数情况下，计算机是采用编译方式处理源程序的。源程序输入计算机后（或者是直接在计算机中编制而成），调用相应的编译程序进行编译，形成用机器语言代码表示的目标程序，即目标代码。如果这种程序需要多次使用，就可将它们作为独立的文件保存，并冠以文件名，以便今后直接使用。

## 5. 由硬件执行目标程序

通常，目标程序存储在磁盘中，用户需执行时给出文件名，操作系统按文件名调出目标程序并送入主存，再将它在主存中的首址送入程序计数器 PC，从该地址开始依序执行目标程序。

## 1.4.2 指令执行过程

目标程序的实体就是用代码表示的指令序列，因此掌握了每条指令的执行过程也就掌握了程序的执行过程。下面以一条加法指令为例，说明指令执行过程的一般情况。

加法指令“ADD AX, [1000H]”的执行过程如下。

### 1. 取指令与指令分析

1.2.1 节中提到，CPU 中有一个程序计数器 PC，存放着当前指令（取指时）所在主存单元的地址码。因此每当读取指令时，就先将 PC 的内容送入主存储器的地址寄存器，据此访问主存单元，从中读出指令，送入指令寄存器 IR。一条指令代码可能分别存放在几个地址连续的主存单元中，每读出一个单元的指令代码，PC 内容就相应地加 1，如果这条指令占 n 个主存单元，那么在该指令代码都读取后，PC 内容就加了 n，这时 PC 指示的就是下一条指令在主存

的位置。

当指令代码读入 IR 后，相应的逻辑电路（可称为指令译码器）自动分析：这是一条什么指令？有关的操作数存于何处？控制器分步发出微操作命令，以实现该指令的功能。例如，ADD 表示加，因此这是一条加法指令。相加的两个操作数，一个来自主存，地址码是用十六进制数表示的 1000，即该操作数存放在主存的 1000H 单元中；另一个操作数存放在名为 AX 的寄存器中。相加后，其结果仍存入 AX 寄存器，不再保留原来的内容。

读取指令是需要时间的，由译码电路对指令代码进行分析的时间很快，一般不需要安排专门的时间段，所以我们将读取指令与指令分析作为一步来阐述。

## 2. 读取操作数

根据本例需要，先将指令提供的操作数地址码 1000H 送入主存的地址寄存器，从中读出操作数，送入 CPU 中的一个暂存器。读取操作数后，1000H 单元中仍保留着原来的内容，换句话说，从主存中读出的信息具有复制性质，不影响原来的内容。这种操作数被称为源操作数。

另一个操作数存放在 CPU 的 AX 寄存器中，运算后，AX 改为存储运算结果，原来存放的操作数不再保留。AX 寄存器既是一个操作数的来源地，又是存放运算结果的目的地，所以它提供的操作数被称为目的操作数。

本例中，读取操作数只需访问一次主存，有些指令可能需要访问多次。

## 3. 运算

本例中，ADD 是指令的操作码，表明要进行加法运算。将已从主存单元中读出的源操作数和 AX 中的目的操作数都送往运算器相加，然后将运算结果送回 AX 寄存器。

## 4. 后继指令地址

在读取指令时 PC 的内容已自动修改，本例不需要转移指令地址，所以 PC 修改后的内容就是后继指令地址，即下一条待执行指令所在存储单元的地址。

为了让初学者容易理解，前面分步阐述了一条指令的读取与执行过程。在实际的 CPU 中，为了提高工作速度，需采取一些措施，如不等一条指令执行完毕就预取下一条指令，让几条指令的部分操作重叠执行，甚至让两条指令并行地执行，等等。

# 1.5 计算机的特点和性能指标

## 1.5.1 计算机的特点

由于采用数字化的信息表示方法及存储程序工作方式，计算机具有如下重要特点。

### 1. 能在程序控制下自动连续地工作

计算机能执行程序，而且采用存储程序工作方式，一旦输入可执行的目标程序，只要给出运行条件和起始地址，启动后就能自动、连续地执行程序。这是数字计算机的基本特点，也是与其他计算工具（如计算器）的本质区别。

## 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，这不仅极大提高了工作效率，还使许多复杂问题得以实际解决。目前，可采用若干 CPU 构成多处理器系统，有效地提高计算机的运算速度。

## 3. 运算精度高

由于采用数字代码表示信息，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。当然，出于成本上的考虑，实际计算机的基本字长会有一定限制，但在软件上稍加变化便能实现多字长运算，从而获得更高的精度。

## 4. 具有很强的信息存储能力

二进制代码容易被存储，如采用双稳态触发器，电容上有无电荷、不同的磁化状态等多种物理机制都可保存或暂存 0、1 代码，因此计算机中设置有各类存储器，具有很强的信息存储能力。计算机的许多功能和特点也是由此而派生的。计算机能够存储程序，所以能自动、连续地工作；存储容量越大，存储的程序就越多，计算机的功能就越强，而且可以继续扩充功能；计算机能够存储大量的信息，是使许多信息处理得以实现的前提条件。

## 5. 通用性强，应用领域广泛

基于信息表示的数字化，计算机能够处理范围广泛的各类信息，因此可以应用在所有的领域。计算机不仅能实现算术运算，也能实现逻辑运算，即可对各类信息进行非数值性质的运算处理，如图像处理、语音处理、文字处理、知识处理、信息检索、逻辑判断和决策等。

## 1.5.2 计算机的性能指标

全面衡量一台计算机的性能需要考虑多种指标，下面是一些基本的性能指标。

### 1. 基本字长

CPU 的基本字长是指参与一次运算的二进制数的位数。对此有两点需要注意：

其一，计算机中数的表示有定点数与浮点数之分，相应地有定点运算与浮点运算之分。习惯上，CPU 字长是一次定点运算的字长。例如，目前微型计算机的字长分为 8、16、32、64 位，就是指它的一次定点运算的位数。

其二，8 位二进制为 1 字节，一个字符可以用 1 字节的代码来表示。为了能灵活地处理字符串类信息及其他以字节为单位的信息，大多数计算机既能进行全字长运算，又能支持以字节为单位的运算。另外，通过软件可以实现多倍字长的运算。

### 2. 数据通路宽度

数据总线一次所能并行传输的二进制数的位数称为数据通路宽度。它体现了信息传输的能力，会影响计算机的有效处理速度。一台计算机至少有两处需考虑数据通路宽度问题，一是 CPU 内部，二是 CPU 外部的总线。CPU 内部的数据通路宽度一般等于基本字长，而外部的数据通路宽度取决于系统总线。有些计算机的内、外数据通路宽度相等，如 Intel 80386 CPU 的都是 32 位，称为 32 位机。有些计算机的外部数据通路宽度小于内部，如 Intel 8088 的内部为

16 位，外部为 8 位，称为准 16 位机。

### 3. 运算速度

即使是同一台计算机，执行不同运算所需的时间可能不同，因而对运算速度的描述通常采用不同方法。

#### (1) CPU 时钟频率与主频

主频即 CPU 的时钟频率。时钟通常是指晶体振荡器的输出经相关处理后提供给 CPU 的脉冲序列，其频率称为主频。

CPU 执行一条指令需要分若干步操作完成，如何确定一步操作的起始和结束呢？这就需要使用一个时钟信号来定时控制，一个时钟周期完成一步操作，所以时钟频率的高低在很大程度上反映了 CPU 速度的快慢。

注意，主频与实际的运算速度并不是一种简单的线性关系。由于现代 CPU 芯片内部通常采用多种并行技术如流水线（一个时钟周期通常有多条指令操作重叠执行），因此主频只是 CPU 性能表现的一方面，而不代表 CPU 的整体性能。

#### (2) 每秒平均执行指令的条数

常用表示单位为 MIPS（Million Instructions Per Second，每秒执行的百万条指令）。此处的指令是指单字长定点指令。

#### (3) 分别标明几种典型四则运算所需时间

如标明定点加减、乘、除、浮点运算所需时间等。现在常以浮点运算速度（FLOPS，Floating Point Operations Per Second，每秒执行的浮点运算次数；MFLOPS，每秒执行的百万次浮点运算次数）作为计算机的速度指标。

## 4. 主存储器容量

CPU 可直接编址访问的存储器是主存储器，需要执行的程序与需要处理的数据就存放在主存中。如果主存容量大，就可以运行比较复杂的大程序，存放大量的信息，利用更完善的软件环境。所以，计算机处理能力的大小与主存容量的大小有紧密的关系。

主存储器容量的表示方法主要有如下两种。

#### (1) 字节数

大多数计算机的主存是按字节编址的，即每个编址单元存放 8 位二进制数，称为 1 字节，表示为 1 B（Byte）。相应地，用字节数表示存储容量的大小。通常，表示主存容量有以下几个常用的单位： $2^{10}$  B=1024 B=1 KB， $2^{20}$  B= $2^{10}$  KB=1 MB， $2^{10}$  MB=1 GB， $2^{10}$  GB=1 TB。

例如，Intel 8086 微处理器的主存按字节编址，有 20 位主存地址总线，则可访问的主存单元数为  $2^{20}$  个，其主存容量为 1 MB。Pentium 微处理器有 32 位地址总线，则可访问的主存单元数为  $2^{32}$  个，其主存容量为 4 GB。

#### (2) 单元数（字数）×位数

有些计算机的主存可以按字编址，即每个编址单元存放 1 个字，其位数等于基本字长。在这种情况下，采用单元数×位数来表示存储容量的大小。例如，64K×16 位表示该存储器有  $64 \times 1024 = 65536$  个单元，每个单元 16 位。

## 5. 外存容量

外存储器容量一般是指联机的磁盘容量，以字节数表示。大量的软件存放在外存中，需要

运行时再调入主存。

## 6. 外围设备及其性能

外围设备的配置情况也是影响系统性能的重要因素。一般在系统的技术说明书中给出了允许配置及典型配置的规格。实际上，用户在配置外围设备时，除了要考虑系统的性能，还要考虑自己的需要与支付能力，以求二者之间的最佳结合。

## 7. 系统软件配置

理论上，计算机可以无止境地扩充其软件，但购置系统软件时需要考虑所需的硬件支持，如存储容量够不够。考察某个系统当前配置了哪些软件，可以了解它具有的功能。

# 1.6 计算机的发展与应用

## 1.6.1 计算机的发展历程

从 1946 年出现第一台电子计算机，计算机技术得到了迅速发展，可谓日新月异。这表现在许多方面，如硬件方面的逻辑器件和体系结构，软件方面的程序设计语言、操作系统、网络软件、人工智能等，这些发展是相辅相成的。

### 1. 计算机的换代

由于计算机的发展极为迅速，人们将取得重大突破后的计算机称为新一代计算机。

① 第一代（电子管计算机）。其主要特征是采用电子管构成逻辑电路，运算速度约为几千次每秒到几万次定点加法运算每秒，生存时期大约是 1946 至 1954 年。这段时间是用机器语言或汇编语言编程，后期出现了一些简单的 I/O 管理程序。

② 第二代（晶体管计算机）。其主要特征是采用分立式晶体管构成逻辑电路，运算速度为几万次每秒到几十万次每秒，生存时期约为 1955 至 1964 年。软件方面出现了高级程序设计语言，相应地出现了编译程序、子程序库、批处理管理程序等系统软件。

③ 第三代（中、小规模集成电路计算机）。其主要特征是采用中、小规模集成电路，开始用半导体存储器作为主存，生存时期约为 1965 至 1974 年。硬件方面采用了流水线技术、微程序控制技术，提出了系列机概念。软件方面操作系统逐渐成熟，出现了虚拟存储技术、信息管理系统、网络通信软件等，开始出现独立的软件企业。

④ 第四代（大规模、超大规模集成电路计算机）。在集成电路中，每块芯片内含有的门电路数或元件数称为集成度。每片几百门至几千门称为大规模集成电路（Large Scale Integration, LSI），更高的称为超大规模集成电路（Very Large Scale Integration, VLSI）。随着 LSI 和 VLSI 的出现，计算机的发展又出现了一次飞跃，进入第四代。一般认为，第四代约从 1975 年开始，直至今天，当前大部分实用的计算机都属于第四代。

后来又出现了第五代、第六代的提法，但尚未得到更多的认可。

在使用 VLSI 后，一个重大的飞跃是出现了微型计算机，从而打破了原有计算机体系结构，为计算机的应用拓展了广阔的空间。

进入第四代后，计算机的发展更为迅速。硬件方面，在系统结构上发展了并行处理、多机系统、分布式计算机、计算机网络等技术。软件方面提出了软件工程概念，出现了一些更完善的高级语言、操作系统、数据库系统、网络软件、多媒体技术等。

## 2. Intel 80x86 系列微处理器的进展

Intel 系列微处理器的演变历史是计算机技术演变的一个写照。过去，Intel 公司几乎每 4 年开发出一种新的处理器，目前已将每代开发时间缩短到 1~2 年，以继续保持领先优势，不断推出最新产品。此外，Intel 80x86 系列微处理器占领市场的另一个重要原因是，该系列在基本结构上采用向后兼容的设计思想，即新开发的微处理器与以前的微处理器在指令系统层兼容，以保证用户原来开发的软件仍可在新的系列微处理器上运行。下面介绍 Intel 80x86 系列微处理器的进展。

20 世纪 70 年代 Intel 微处理器的典型产品如下。

① 8080：世界上第一个通用微处理器，8 位机，处理器与主存的数据总线宽度为 8 位，典型时钟频率为 108 kHz，内部集成晶体管数约 6000 个，主存容量可达 64 KB。8080 曾用于第一台个人计算机。

② 8086：比 8080 强大得多的 16 位微处理器，除了采用 16 位的数据总线宽度和寄存器，还开辟了指令高速缓存（或称队列），以存放预取的指令，具有一条两级流水线，使不同指令的读取与执行可以部分重叠执行；具有 20 位地址总线，可寻址 1 MB 存储空间；典型时钟频率为 8 MHz，内部集成晶体管数约 2.9 万个。8088 是 8086 的一个变形，曾用于最初的 IBM PC，并确保了 Intel 公司的成功。

20 世纪 80 年代 Intel 微处理器的典型产品如下。

① 80286：仍为 16 位微处理器，是 8086 的扩展产品，具有 24 位地址总线，可寻址 16 MB 存储空间，典型时钟频率为 10 MHz，内部集成晶体管数约 13.4 万个。

② 80386：Intel 的第一个 32 位处理器，是一个有重大改进的产品，32 位的结构，复杂程度和功能可以与几年前的小型机和大型机相抗衡；采用一条多级流水线结构，具有片内存储管理部件，是 Intel 的第一个支持多任务的处理器，即支持同时运行多个程序；具有 32 位地址总线，可以寻址 4 GB 的存储空间；典型时钟频率为 33 MHz，内部集成晶体管数约 27.5 万个。

③ 80486：相当于增强型 80386、浮点部件 FPU、8 KB 高速缓存（Cache）的集成，采用更为复杂、功能更强的高速缓存技术和指令流水线技术，典型时钟频率为 50 MHz，内部集成晶体管数约 120 万个。

20 世纪 90 年代 Intel 微处理器的典型产品是 Pentium 系列如下。

① Pentium：开始引入超标量（Super Scalar）技术，有两条整数指令流水线与一条浮点指令流水线，允许更多的指令并行执行，时钟频率为 100 MHz，内部集成晶体管数约 310 万个。

② Pentium Pro：继续推进由 Pentium 开始的超标量结构，为了进一步提高处理器的性能，采用寄存器重命名、转移预测、数据流分析、推测执行等技术；具有 36 位地址总线，可以寻址 64 GB 的存储空间；典型时钟频率为 150 MHz，内部集成晶体管数约 550 万个。

③ Pentium II：融入了专门用于有效处理视频、音频和图形数据的 Intel MMX 技术；典型时钟频率为 250 MHz，内部集成晶体管数约 750 万个。

④ Pentium III：融入了新的浮点指令，以支持三维图形软件。

2002 年, Intel 发布了超线程微处理器 P4 (单核 CPU)。超线程(硬件多线程)技术允许多个线程共享单个 CPU 内部的功能部件,可以提高多线程应用程序的性能。时钟频率为 3.06 GHz,采用 0.13 μm 制造工艺,内部集成 5500 万晶体管。

Intel 的首个双核芯片发布于 2005 年。2006 年, Intel 发布了针对个人计算机的双核微处理器产品 Core2, 主频为 3.2 GHz, 内部集成 2.91 亿个晶体管。

2011 年, Intel 重新构建了高、中、低端,即 Core i7、i5、i3 多核微处理器架构。Core i3 针对低端市场,采用双核架构,二级 Cache 为 2 MB; Core i5 针对主流市场,采用四核架构,二级 Cache 为 4 MB; Core i7 针对高端市场,采用四核 8 线程或六核 12 线程架构,二级 Cache 不小于 8 MB。此后, Intel 不断对 Core i7、i5、i3 进行升级。

Intel 微处理器的发展历史表明,处理器的性能提升一方面依赖于集成电路技术的进步,另一方面依赖于体系结构的不断改进(如采用流水线、集成 Cache 在芯片上、超标量、超线程、多核)。

### 3. 国内微处理器的发展情况

我国是世界上第一大芯片市场,但芯片领域曾经长期依赖进口,自给率低。我国微处理器芯片的发展起步较晚,下面着重介绍龙芯和海思麒麟处理器的发展情况。

目前,国际上市场份额较大的主流 CPU 架构有 ARM、x86 和 MIPS。新设计的 CPU 为了软件兼容,通常会选用成熟的 CPU 架构,如龙芯选用 MIPS、华为选用了 ARM。

#### (1) 龙芯系列

2002 年,我国第一款具有完全自主知识产权的通用 CPU 芯片龙芯 1 号正式问世,采用 MIPS 架构,实现了我国集成电路产业“零的突破”,打破了国外芯片在我国的长期垄断地位。

龙芯 1 号采用动态流水线结构,定点和浮点最高运算速度均超过每秒 2 亿次,与 Pentium II 芯片性能大致相当,在总体上达到了 1997 年前后的国际先进水平。

2004 年 6 月,64 位龙芯 2 号发布,性能相当于 P4 水平,比龙芯 1 号性能提高 10~15 倍。

2009 年研制的 3A1000 是我国首个四核 CPU 芯片,标志着国内已首次掌握了多核 CPU 的片间互连及 Cache 一致性技术。3A1000 的第二次改版于 2012 年 8 月流片成功,至今还是龙芯销售的一款重要芯片,尤其在工控领域。

2015 年发射的北斗双星搭载的就是龙芯 CPU,龙芯为国家航宇领域也做出了突出贡献。

#### (2) 海思麒麟系列

2009 年,华为海思推出了首款移动微处理器 K3V1,采用 ARM 架构、110 nm 制造工艺,主要面对中低端市场。

2012 年,华为海思发布了 K3V2,是当时全球最小的四核 ARM A9 架构处理器,集成 GC4000 的 GPU,采用 40 nm 制造工艺,主要用于华为 P6 和 Mate1 等产品上。

其后,华为推出的麒麟处理器则全面采用了 SoC (System on Chip, 片上系统) 架构,即在单个芯片上集成 CPU、通信模块、音视频解码和外围电路等一个完整的系统。

2014 年,麒麟 910 (K3V2 改进版,也是四核 CPU 结构) SoC 芯片发布,从此开始改变了芯片命名方式。麒麟 910 首次集成华为自研的巴龙 Balong710 基带,把 GPU 换成 Mali, 制造工艺升级到 28 nm。

2014 年 6 月,麒麟 920 SoC 芯片发布,采用 28 nm 制造工艺,八核 CPU 结构,将 4 个

ARM Cortex-A15 和 4 个 Cortex-A7 处理器结合在一起，使同一应用程序可以在二者之间无缝切换，解决了高性能和低功耗之间的平衡，在提升性能的同时延长了电池使用时间。麒麟 920 还集成了协处理器 i3，能以极低的功耗运行，持续采集加速计、陀螺仪和指南针等数据，使一些智能应用可以在待机下一直运行。

2015 年 11 月，麒麟 950 SoC 芯片发布，其八核 CPU 结构包括 4 个 Cortex-A72 和 4 个 Cortex-A53，采用 16 nm 制造工艺，集成自研 Balong720 基带、双核 14-bit ISP 和音/视频解码芯片，集成了 i5 协处理器，是一款集成度非常高的 SoC。

2016 年 10 月，麒麟 960 SoC 芯片发布，GPU 为 Mali G71 MP8。

2017 年 9 月，人工智能芯片麒麟 970 SoC 芯片（八核 CPU 结构与麒麟 960 相同）发布，首次采用台积电 10 nm 工艺、集成 NPU（Neural Network Processing Unit，神经元网络）专用硬件处理单元，以处理海量数据。其集成的 55 亿个晶体管远高于高通和苹果的芯片，使华为步入了顶级芯片厂商行列。

2018 年 8 月，麒麟 980 SoC 芯片（八核 CPU 结构包括 4 个 Cortex-A76 和 4 个 Cortex-A55）发布，首次采用台积电 7 nm 制造工艺，集成 69 亿个晶体管，全面升级的 CPU、GPU、新的双核 NPU 使其性能更为优秀。

2019 年 9 月，全球首款 5G SoC 芯片麒麟 990 5G（八核 CPU 结构与麒麟 980 相同）发布，内置巴龙 5000 基带即 5G，采用 7 nm 制造工艺，GPU 为 16 核 Mali-G76，NPU 在双核基础上增加了一个微核。

华为研制手机芯片的时间虽然短，但是进步非常快。

#### 4. 分类

20 世纪 80 年代以前，人们按功能、体积、价格等因素将计算机分为微型机、小型机、大中型机和巨型机。但随着微型计算机技术的迅速发展，这种分类方法正逐渐失去意义。因为现在一个高性能的 CPU 和部分存储器已集成在一块芯片，这样的微处理器其工作速度、可访问的主存容量超过了以前的小型机和大型机，其字长也可以达到以前大型机的水平，甚至在一块芯片上已经集成了多个 CPU，称为多核处理器。用多个微处理器构成的多处理器系统的功能也超过了传统的巨型机，如 2000 年 IBM 公司研制成功的一种超大规模并行处理计算机由 8192 个微处理器组成，工作速度达到 12.3 万亿次浮点运算每秒，用于模拟核爆炸技术。

当前，更符合市场发展的通常是按应用分类。按照计算机的应用特征，现代的计算机可分为个人移动设备、桌面计算机、服务器、集群/仓库级计算机和嵌入式计算机五种。

个人移动设备是指一类带有多媒体用户界面的无线设备，如手机、平板电脑等。目前，手机已成为销量最大的一类计算机，设计的关键问题包括成本、能耗、多媒体应用的响应性能等。

桌面计算机包括个人计算机和工作站。个人计算机主要为一个用户提供良好的计算性能和较低成本的工作环境。工作站是指具有完整人机交互界面、图形处理性能和较高计算性能，可配置大容量的内存和硬盘，I/O 和网络功能完善，使用多任务多用户操作系统的小型通用个人化计算机系统。

服务器是 20 世纪 90 年代迅速发展起来的主流计算机产品，是为网上客户机在网络环境下提供共享资源（包括查询、存储、计算等）的高性能计算机，具有高可靠性、高性能、高吞吐能力、大内存容量等特点，并且具备强大的网络功能和友好的人机界面。其高性能主要体现

在长时间的可靠运行、良好的扩展性、强大的外部数据吞吐能力等方面。

集群是指一组桌面计算机或服务器通过局域网连接在一起，运转方式类似一个更大型的计算机。每个节点都运行自己的操作系统，节点之间通过网络协议进行通信。最大规模的集群称为仓库级计算机，使数万个服务器像一个服务器一样运行。这类计算机可用于互联网搜索、社交网络、视频分享、在线销售等场合。与服务器的主要区别在于，集群/仓库级计算机以很多廉价的组件构建模块，性价比是关键因素。

在很多应用中，计算机作为应用产品的核心控制部件，隐藏在各种装置、设备和系统中，这样的计算机被称为嵌入式计算机。嵌入式计算机系统集软件与硬件于一体，是满足具体应用对功能、可靠性、成本、体积、功耗等综合性严格要求的专用计算机系统。我们以能否运行第三方软件作为区分嵌入式和非嵌入式计算机的分界线。个人移动设备可以运行外部开发软件，与桌面计算机相同；而嵌入式计算机是专用设备，在硬件和软件复杂性方面受到很大限制，通常不能运行第三方软件。

## 5. 计算机网络

需要特别强调的是，当前已进入网络化时代，这是计算机发展的重要方向之一。计算机网络利用通信线路，将分布在不同地点的多个独立计算机连接起来，使多个用户能够共享网络中的硬件、软件和信息等资源。计算机网络是计算机技术与通信技术结合的产物，在信息时代具有极为重要的意义，不仅大大提高了人类的工作能力，也正在改变着人们的生活、工作和学习的方式。

按照网络的分布距离，计算机网络分为局域网和广域网（远程网），而因特网（Internet）可使分布在全球各地的各种网络实现互连。现在，许多机关、学校、企业建立了自己的局域网，将内部的计算机接入，再将局域网接入因特网，实现全球范围的互连。

## 1.6.2 计算机性能提高的技术

目前，计算机性能的提高主要有两方面的因素：常规是改进芯片，包括电子器件和电路的制造工艺，提高芯片的集成度与工作频率，更重要的手段是改进计算机的系统结构。

芯片制造商通过在 CPU 芯片上增加新的电路，缩减芯片尺寸，减少部件间的距离，来提高 CPU 的性能。例如，从 1979 年开始推出的 Intel 80x86 系列，大约每 3 年就要发布新一代的 CPU 芯片，它的晶体管数为上一代的 4 倍，性能提高 4~5 倍。其内存芯片仍旧采用基本的主存储器技术，通过采用新的工艺，动态随机存储器（DRAM）的容量每 3 年提高 4 倍。

近年来，计算机性能大幅提高的更深层次原因是源于对传统冯·诺依曼机系统结构的改进，一是在单 CPU 芯片中采用流水线处理技术、集成 Cache、RISC 技术、超标量、硬件多线程技术等，二是用多个 CPU 或计算机构成并行计算机系统。

### 1. 提高和发挥单 CPU 性能的技术

提高和发挥 CPU 性能的技术有很多，下面讨论主要采用的技术。

#### (1) 流水线处理技术

流水线是指利用执行指令操作的并行性实现多条指令重叠执行的技术。

一条指令的执行可以分为不同的阶段，一般分为取指令、指令译码、取操作数、执行算术

/逻辑操作、存结果，在有的计算机中还可细分。因此，可以建立指令执行的流水线（类似汽车装配线），使多条连续指令在不同的执行阶段重叠执行，即在 CPU 内部设置与指令不同执行阶段相对应的独立部件，以构成多级流水线结构，让每个独立部件完成指令的一个步骤的操作，使多条指令重叠执行。例如，80486 CPU 中设有总线接口部件、指令预取部件、指令译码部件、产生微命令的控制部件、高速缓存 Cache、执行部件和存储管理部件构成的流水线，可以使多条指令重叠执行，使每秒平均执行的指令数大大增加，具有平均每个时钟周期执行 1 条常用指令的功能。流水线处理技术已成为高性能 CPU 采用的一种基本技术。

但是，除非以指令形式源源不断地向 CPU 流水线提供工作流，否则 CPU 将达不到它的潜在速度。任何阻碍工作流的事件都会降低 CPU 的性能，如指令间的数据相关（流水线上多条指令重叠执行会出现：前面一条指令还未执行结束，其后指令就需要使用该指令结果）和控制相关（流水线上执行转移指令）会引起流水线停顿。为了保证流水线能被充分利用，一些相关处理技术，如 Forwarding、转移预测、静态调度（由编译器判断相关后，对有相关的目标代码重新排序以减少流水线停顿的技术）等，已被应用到 CPU（如 Pentium II）的设计中。

#### （2）RISC 技术

RISC 技术主要通过减少指令条数、简化指令寻址方式、采用相同长度的指令格式和 LOAD/STORE 结构等技术来简化指令系统，不同类型指令的执行时间接近，使流水线的结构简化，提高了流水线的使用效率，从而提高了指令执行速度。

其设计思想不仅为 RISC 采用，也为传统的 CISC（Complex Instruction Set Computer，复杂指令系统集计算机）采用。例如，Intel 80486 与以前的 80286 及 80386 相比，已经吸收了很多 RISC 中的思想，其中很重要的一点就是注重常用指令的执行效率，减少常用指令执行所需的时钟周期数。

#### （3）超标量技术

标量（也称为单发射）CPU 是指具有一条指令执行流水线的处理器，设计目标是做到平均每个时钟周期执行一条指令。为了提高处理器的性能，要求处理器具有每个时钟周期发射执行多条指令的能力。超标量（多发射）CPU 是指具有 2 条或 2 条以上相互独立的指令执行流水线，可在每一个时钟周期同时发射执行 2 条或 2 条以上指令的处理器。

例如，Pentium 微处理器采用超标量结构，有两条独立的整数指令流水线与一条浮点指令流水线。每个时钟周期可以同时发射执行两条整数指令，因而相对同一频率下工作的 80486 来说，其性能几乎提高了 1 倍。

在超标量微处理器中，每个时钟周期可同时发射执行多条指令，但指令的高发射率意味着相关发生的频率增加，而且多发射的结构决定了相关的复杂性。因此，为了有效地处理相关，需要采用静态调度与动态调度（即在流水线上执行指令时，由硬件检测指令间的相关性，然后对相关指令重新排序执行，这也称为乱序执行）技术相结合的方法。静态调度可在编译过程中减少相关的产生；而动态调度可根据处理器的动态信息发掘出更多的指令级并行，动态调度简化了编译器的设计，减小了编译代码对硬件的依赖，但代价是大量的硬件开销。

#### （4）硬件多线程

采用流水线和超标量技术可以有效地提高单线程 CPU 的指令执行效率，但是当单线程指令执行遇到阻塞（如相关、Cache 缺失）时，会使 CPU 内部执行功能部件空闲，为了充分利用这些功能部件，就出现了硬件多线程技术。由于线程间的相关性较少，当一个线程遇到阻塞时，

CPU 可以切换到另一个线程执行。

硬件多线程允许多个线程共享单个 CPU 的功能部件。为了允许这种共享，处理器必须能够复制每个线程的独立状态，这样才具备在多个线程之间进行切换的能力。

硬件多线程有两种实现方法：细粒度多线程（CPU 必须能够在每个时钟周期进行线程切换）、粗粒度多线程（CPU 在当前线程遇到阻塞时才切换到其他线程）。

同时多线程（Simultaneous Multithreading, SMT）是多线程的变种，也称为超线程。由于多发射动态调度 CPU 具有多个并行执行功能部件，因此可以通过寄存器换名、动态调度、多发射等技术，实现在一个时钟周期同时发射多个不同线程的指令而不用考虑其相关性。Intel Pentium 4、IBM POWER 5 都采用了 SMT 技术。

但是，硬件多线程通常会损失单个线程的性能。而且，为了减少多个线程同时运行引起的各种资源（如指令预取缓冲、存储器带宽）竞争，CPU 会增加相应硬件资源而使其结构变得更复杂，从而导致功耗大大增加，结果是 CPU 性能的提升与功耗的增大不成正比。

由于单核 CPU 指令级并行的限制、采用硬件多线程的复杂性、时钟频率增加的瓶颈和 CPU 功耗等问题，自 2002 年开始单核 CPU 的性能增长进入缓慢期，因此从 2004 年 Intel 就开始转向研发多核芯片，以在进一步提升 CPU 性能的同时降低功耗的提升。

#### （5）平衡不同子系统的数据吞吐率

组成计算机的三个主要子系统（即 CPU、存储系统和 I/O 系统）之间存在不同的数据吞吐率和处理要求，而且差异很大。这就导致了在它们之间寻求性能平衡的需要：调整组织和结构，以补偿各子系统之间的能力不匹配，提高整个计算机系统的效率。

CPU 与主存储器的接口问题是最重要的。当 CPU 速度和存储器容量快速增长时，主存储器和 CPU 之间的数据传输率却严重滞后。解决该问题的主要技术有：在主存和处理器之间引入更复杂、更有效的高速缓存（Cache）结构，以减少存储器访问频度；采用多模块或多端口主存储器，实现并行存取；增大总线的数据宽度，以增加每次读出或写入主存的数据位数；在主存芯片中加入高速缓存或其他缓冲机制来改进其接口，提高它的效率；采用高速总线和使用分层总线来缓冲和结构化数据流。

I/O 系统的数据吞吐率远比 CPU/主存小得多，很早就有人想到采用中断技术让 I/O 系统的操作与 CPU 的操作重叠进行，以提高整个系统的效率。利用 CPU 在进行算术/逻辑操作或不需访存的操作时，使外设直接与主存交换数据的技术，即直接存储器访问（Direct Memory Access, DMA）技术早已普遍使用，还可采用缓冲和暂存机制以及多级互连总线结构，以提高 CPU 与外设之间的数据传输率。

上述技术可有效地提高和发挥单处理器计算机的性能，但是单个 CPU 的性能与速度毕竟有限，仍有许多应用问题，如复杂的科学计算、系统模拟、天气预测、知识处理等，仅靠单处理器计算机系统是无法解决的。因此，只有通过开发并行计算机与超级计算机来满足高端应用的需求。

## 2. 并行计算机

使用多个处理器或多个计算机组成一个并行计算机（Parallel Computer）是提高系统性能的有效方法。并行计算机是多个处理部件（Processing Element）的集合，所有的处理部件通过相互通信，协同解决复杂问题。

处理部件可以是 CPU，也可以是计算机。处理部件之间（或处理部件与存储器之间）由互连网络连接。这种并行计算机的优点是可以利用现有的高性能处理器或计算机，再加上快速互连网络，构成高性能的并行系统。

由多个处理器及存储器模块构成的并行计算机被称为多处理器系统（Multiprocessor System）。处理器之间的通信通过共享存储器（Shared Memory）进行。目前，最普遍的多处理器组织方式是对称多处理器（Symmetric MultiProcessor, SMP），由多个相同或相似的处理器组成，以总线或某种开关阵列互连成一台计算机。

特别指出，将多个处理器集成在一块芯片上称为多核芯片，目前已经普遍应用在个人移动设备、服务器、桌面机和嵌入式计算机中。从结构上，多核芯片仍然属于多处理器系统。

由多个计算机构成的并行计算机被称为多计算机系统（Multicomputer System），其中每个计算机只能访问自己内部的私有存储器，而无法访问其他计算机内的存储器。计算机之间的通信只能通过消息传递（Message Passing）方式进行。目前常用的多机系统是集群/仓库级计算机，由一组完整的计算机或服务器通过局域网互连而成，作为统一的计算资源一起工作，并能产生像一台机器在工作的印象。

### 1.6.3 计算机应用举例

本质上，计算机的工作就是对信息进行处理，而信息无处不在，所以计算机应用涉及所有的领域。下面根据信息处理任务的性质，分类列举部分典型的应用领域。

#### 1. 科学计算

科学计算一般指这样一种类型的任务：原始数据不太多，而计算量大且比较复杂，如求解数学方程，大坝、桥梁等工程结构的应力分析，航天技术中对卫星轨道的计算，气象预报，对化学反应甚至核爆炸进行计算机模拟等。

#### 2. 信息管理中的数据处理

数据处理一般是指那些数据量很大而操作类型相近的任务，如各种人事管理、企业管理、金融管理、信息情报与文献资料检索等。数据处理中存储数据所需的存储空间远大于处理数据的程序所需的存储空间。大多数计算机被用来为这一类任务服务。以计算机信息管理系统为核心，加上文字处理、通信、分析决策，就形成了办公自动化系统。

#### 3. 科技工程中的数据处理

这类数据处理与信息管理中的数据处理有些不同，数据量比较大，同时分析计算比较复杂，如物理探矿中对振动波形的分析，医疗仪器中的图像处理，卫星遥感数据处理等。

#### 4. 自动控制

计算机应用于各类生产过程控制，极大地提高了生产力和生产质量，如炉温控制、机床控制和各种化工生产过程控制等。以炉温控制过程为例，传感器先将温度值变为电信号，再将电信号转换为数字信号并送入计算机，然后将当前温度值与要求保持的温度值进行比较，得出误差值，接着按照某种控制算法进行调整，调节发热部件使温度回到要求值。过程控制的突出特点是实时性，即计算机做出反应的时间必须与被控制过程的实际需要相适应。

## 5. 计算机辅助设计( CAD )、计算机辅助制造( CAM )、计算机模拟、计算机辅助教学( CAI )

运用计算机设计、监控生产过程，可以使生产进入高度自动化。许多复杂的事物可以在计算机产生的虚拟环境中进行模拟分析，使所需时间缩短，成本降低。例如，驾驶员训练环境的模拟，复杂化学反应过程的模拟，核反应过程的模拟，飞机、车辆、桥梁、大坝等的应力情况的模拟，等等。

近年来，随着信息技术的发展，传统的教学手段受到挑战，计算机辅助教学的应用日益广泛并取得了长足的发展。利用多媒体技术制作的 CAI 课件，将文本、图像、声音、动画集为一体，解决了传统课堂中难以解决的问题，为学生提供了生动、直观的学习素材，而且可实现人机对话。将 CAI 软件应用于网络环境，使一些有经验的教师通过计算机网络对学生给予指导，就可实现远程教育。

## 6. 人工智能

人工智能（ Artificial Intelligence, AI ）是计算机应用中处于前沿地位的一个重要分支，或者说是高层次的应用。人工智能是指用计算机模拟实现人的某些智能行为，包括专家系统、模式识别、机器翻译、自动定理证明、自动程序设计、智能机器人、知识工程等。

专家系统包含知识库和推理机两大部分，能在某特定领域内使用大量专家的知识，去解决专家才能解决的某些问题，如能下国际象棋的著名的“深蓝”系统，某些大型设备的诊断维护系统，中医专家系统，分析物质分子结构的专家系统等。

利用计算机对物体、图像、语音、文字等信息模式进行自动识别，称为模式识别。现在，对西文和汉字的自动识别率已经很高，颇具实用价值。对有限语音的识别能力也已达到可用语言指挥计算机进行某些操作的程度。

## 7. 娱乐活动

随着互联网和计算机的迅速普及，人们可以访问各种各样的网站获取自己需要的信息，随时可以发 E-mail 、在网上和朋友聊天、玩游戏和购物、在线支付等，还可以在计算机或手机上看各种视频节目、听音乐等。

总之，这是一个五彩缤纷的精彩世界，应用实例不胜枚举。更多的辉煌还有待我们大家去创造。

# 习题 1

1-1 简要解释下述概念。

冯·诺依曼机	信息的数字化表示	硬件	软件
操作系统	机器语言	汇编语言	高级程序设计语言
编译	解释	虚拟机	流水线
RISC	超标量	硬件多线程	Cache
并行计算机	个人移动设备	桌面计算机	服务器
集群/仓库级计算机	嵌入式计算机。		

- 1-2 什么是存储程序工作方式?
- 1-3 采用数字化方法表示信息有哪些优点?
- 1-4 如果有  $7 \times 9$  点阵显示出字符 A 的图像, 用 9 个 7 位二进制代码表示 A 的点阵信息。
- 1-5 数字计算机的主要特点是什么?
- 1-6 衡量计算机性能的基本指标有哪些?
- 1-7 某计算机的主存按字节编址, 有 34 位主存地址总线, 其主存容量是多少?
- 1-8 针对一种实际计算机, 列举出其各部件、设备的技术性能及常规软件配置。
- 1-9 软件系统一般包含哪些部分? 列举读者所熟悉的三种系统软件。
- 1-10 对源程序的处理有哪两种基本方式?
- 1-11 提高和发挥单 CPU 计算机性能的主要技术有哪些?
- 1-12 列举两种国产的处理器芯片 (除了教材上介绍的)。