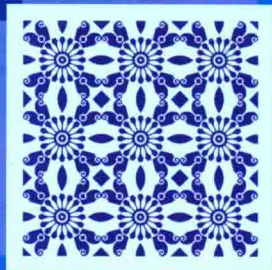




高等院校计算机教材系列

计算机系统 基础概念及编程实践

钱晓捷 编著

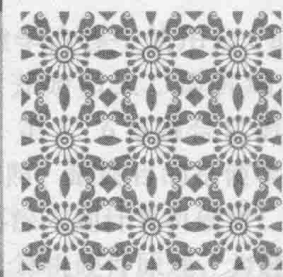


机械工业出版社
China Machine Press

高等院校计算机教材系列

计算机系统 基础概念及编程实践

钱晓捷 编著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

计算机系统：基础概念及编程实践 / 钱晓捷编著. —北京：机械工业出版社，2018.8
(高等院校计算机教材系列)

ISBN 978-7-111-60809-7

I. 计… II. 钱… III. 计算机系统 - 高等学校 - 教材 IV. TP303

中国版本图书馆 CIP 数据核字 (2018) 第 201670 号

本书融合计算机学科的“计算机组成原理”“微机原理”和“汇编语言程序设计”课程的基本内容，同时补充“数字逻辑”课程基础知识，延伸“计算机系统结构”课程核心概念，结合 C 语言和汇编语言编程实践，从软件角度理解计算机系统的工作原理。

本书可以作为“计算机系统基础”或“计算机硬件技术基础”课程的教材或参考书，也就是适合综合“计算机组成原理”与“汇编语言”等教学内容的课程，该课程还可以替代“汇编语言”课程，或者作为“计算机组成与结构”“嵌入式系统”等需要硬件基础知识的前导课程。本书具有以“软”带“硬”、浅显易懂、突出实践等特点，也满足计算机应用开发人员对计算机硬件核心知识的要求。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号 邮政编码：100037）

责任编辑：张梦玲

责任校对：李秋荣

印刷：北京文昌阁彩色印刷有限责任公司

版次：2018 年 9 月第 1 版第 1 次印刷

开本：185mm×260mm 1/16

印张：22.25

书号：ISBN 978-7-111-60809-7

定价：59.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换

客服热线：(010) 88378991 88361066

投稿热线：(010) 88379604

购书热线：(010) 68326294 88379649 68995259

读者信箱：hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问：北京大成律师事务所 韩光 / 邹晓东

前 言

我国计算机科学与技术专业，尤其是计算机工程方向，往往开设多门有关计算机组成与结构的课程，一般包括“数字逻辑”“计算机组成原理”“汇编语言程序设计”“微机原理及接口技术”和“计算机系统结构”等。然而，计算机相关的其他专业并不要求全面深入的硬件技术知识，也没有足够的学时展开如此众多的教学内容。本书基于软件工程专业的课程教学实践，融合上述硬件技术相关课程的基本内容，从技术应用角度通过软件编程介绍计算机硬件组成和计算机工作原理。

在多门课程内容的融合过程和具体的教学实践中，需要努力解决好诸多教学问题，这也就形成了本书内容的特点。

1. 融合计算机组成原理和微机技术实例

传统上，计算机组成原理面向计算机学科，主要介绍计算机硬件的组成结构和工作原理。而微机原理主要针对电子、通信等机电类专业，从应用角度介绍通用微型计算机（简称为微机）的应用技术。本书采取通过实例理解原理的基本思路，即以计算机组成原理为主体，结合微机原理（IA-32 处理器和 PC）实例。这样，一方面利于学生掌握原理，避免重复学习；另一方面使学生熟悉广泛应用的通用微机系统，为应用奠定基础。

2. 以 C 和汇编语言实践贯穿逻辑主线

虽然本书以计算机工作原理和硬件技术为主体，但教学内容中使用 C（或 C++）高级语言、底层汇编语言编程作为实践环节。从第 1 章开始引入 C 语言编程环境（DEV C++），第 2 章主要以 C 语言编程体会数据表示的原理，第 5、6 章融合 C 语言编译程序生成的汇编语言代码和 MASM 汇编语言程序，相互对照，最终目的是使学生掌握汇编语言编程。这使得本书内容从高级语言到低级语言，再深入到计算机硬件，贯穿计算机层次结构；也使得学生能够自然地由软件编程过渡到计算机硬件原理，为进一步学习计算机组成、微机接口技术、嵌入式系统应用奠定基础。

3. 面向软件开发和系统应用取舍课程内容

融合多门课程，需要在内容上进行合理取舍，本书的主要原则是：面向软件开发和系统应用，不以设计处理器、硬件电路为目标，侧重工作原理、硬件电路的外特性。例如，简述运算原理、微程序和硬布线特点，舍弃运算器、微程序和硬布线控制器的设计实现；重点介绍基本指令，突出汇编语言程序结构；只依靠计数器体会接口技术，简介其他接口，引入指令流水线、指令级并行、数据级并行和线程级并行等系统结构先进技术。具体教学内容的选择则采取删繁就简的基本思路。例如，数据编码主要介绍定点整数格式、IEEE 754 标准的浮点格式，不展开定点小数格式、非标准浮点格式相关内容。再如，对于存储器芯片，说明各种存储器芯片特点，而不是内部工作原理；阐明地址译码原理，而不是连接细节。

4. 补充数字逻辑基础知识

对于缺乏硬件电路知识的学生, 本书补充了数字逻辑基础知识。这使得无须单独开设“数字逻辑”先修课程, 数字逻辑只作为本书的一章。教学内容涉及基本概念和核心原理, 具体包括: 逻辑代数, 门电路 (含三态门), 组合逻辑电路的编码器、译码器、加法器, 时序逻辑电路的触发器、寄存器、计数器, PLD 和电子设计自动化 (EDA)。教学要求以理解为主, 满足后续内容的需求即可。

5. 浅显易懂、图文并茂的写作风格

为了使得抽象的计算机工作原理易于理解, 本书努力做到描述清晰准确、浅显易懂, 尽量使用图表提供形象化的释义。重点内容常结合程序示例, 让学生在上机实践中体会问题所在, 激发学生探究的兴趣, 然后再答疑解惑、详细讲解。每章之后编排有较多习题, 分成两种类型: 一类包括简答题、判断题和填空题, 用于使学生掌握基本概念和要点, 通过课堂提问与交互方式进行, 便于了解学生自习情况; 另一类包括问答、计算、编程等应用题, 重点考察学生对计算机工作原理的理解和应用能力, 在学生提交作业后进行课堂解答。部分题目有一定难度, 适合学生深入研讨。

本书由钱晓捷编著, 感谢程楠、石磊、关国利、张青、穆玲玲、姚俊婷等同事的帮助, 感谢华章公司的支持。限于水平, 书中难免存在不当之处, 欢迎广大师生交流指正 (作者的电子邮箱: iexjqian@zzu.edu.cn)。

编者

2018 年 5 月

教学建议

本书遵循“计算机系统能力”的培养要求，比较全面地介绍计算机硬件系统中各部件的工作原理、组成结构以及连接方式，使学生建立计算机系统的整机概念，理解计算机系统层次结构，熟悉硬件与软件之间的接口，掌握指令集结构和汇编语言编程的基本知识，能够运用计算机的基本原理和基本方法，分析有关计算机系统的应用问题。

本书以 C 语言引出汇编语言，用软件编程贯穿计算机组成与结构的逻辑主线，汇集计算机硬件的重点知识和基本原理，各章主要教学内容参见下表，具体的课时安排以 16 周、每周 4 授课学时（64 总学时）为例（未计算上机实验学时）。

章节	内容简介	参考学时
第 1 章 计算机系统概述	通过计算机和 Intel 80x86 系列处理器的发展了解计算机基本概念，结合冯·诺依曼计算机结构熟悉计算机核心工作原理，展开计算机硬件组成、总线结构和软件组成内容，理解计算机系统的层次结构	4
第 2 章 数据表示	理解计算机内部如何表达整数、字符、实数，即掌握定点整数编码、字符 ASCII 码和浮点实数编码，并结合 C 语言基本数据类型的程序示例加深对数值编码、数据存储和运算规则的认知	12
第 3 章 数字逻辑基础	从逻辑代数的基本概念引出常用门电路原理和功能，通过编码器、译码器、触发器、寄存器等常用器件说明数字电路的设计、分析过程，结合可编程逻辑器件引出硬件描述语言和电子设计自动化	4
第 4 章 处理器	简介处理器内部的控制器和运算器基本组成，说明 8 位处理器、16 位 8086、32 位 80386 和 Pentium 的功能结构，重点展开 IA-32 处理器通用寄存器、工作方式和逻辑地址等处理器编程结构	2
第 5 章 指令系统	以 IA-32 处理器指令系统为例并结合汇编语言编程，学习指令编码、数据寻址、常用指令功能，熟悉汇编语言的语句格式、程序框架、开发方法，掌握汇编语言的常量表达和变量应用	14
第 6 章 汇编语言程序设计	围绕顺序、分支、循环和子程序结构，结合汇编语言和 C 语言程序，熟悉指令寻址、控制转移类指令，掌握汇编语言程序设计方法，理解 C 语言基本语句的汇编语言实现	18
第 7 章 存储系统	以存储层次结构中的主存储器、高速缓冲存储器为主体，熟悉半导体存储器的类型、特点、连接，理解 Cache 的工作原理和组成结构，了解虚拟存储器以及 IA-32 处理器的存储管理机制	6
第 8 章 输入/输出接口	在熟悉 I/O 接口的特点、编址和指令基础上，结合 I/O 接口电路理解进行查询传送、中断传送和 DMA 传送的原理，并了解常用的定时控制接口、并行接口、异步串行通信接口和模拟接口	2
第 9 章 处理器性能提高技术	熟悉技术成熟的精简指令集和指令流水线思想，以 IA-32 处理器为例了解高性能处理器运用的并行处理技术，如数据级并行的 SIMD 指令、指令级并行的动态执行，以及线程级并行的多核思想	2

本书需要读者具有 C 语言编程基础，教学宜采用小班化形式，建议融课堂讲授与编

程实践于同一个实验室，以便于适时转换、相互结合，更有助于编程指导和问题研讨，提高教学效果。课前应布置学生自学内容，根据学生对课程内容的掌握情况，适时安排课堂讨论、答疑解惑，这样可以减少授课学时，提升教学效率。

本书强调通过软件编程掌握计算机硬件工作原理，较好的授课形式是教师知识讲解与学生编程实践相结合，并通过有关现象引导学生研讨，进而使其掌握教学内容。如果条件不具备，也建议教师通过示例程序的开发、运行，让学生观察程序开发过程和运行结果，主动参与交流，推动教学内容的展开。如果单独开设上机实验，不妨从高级语言和汇编语言两个方面布置任务：

- 高级语言编程：基于C语言（高级语言）程序理解硬件原理对软件应用的支持作用，主要包括定点整数的表达与存储、整数表达的范围和溢出、字符的表达与存储、浮点实数的表达与存储、程序访问的局部性原理、地址边界对齐问题等。
- 汇编语言编程：基于汇编语言（低级语言）程序熟悉计算机常用指令的功能与应用，体会高级语言如何编译为低级语言，主要包括MASM开发过程及常用指令功能、分支结构程序、循环结构程序、过程调用程序及堆栈作用、高级语言生成汇编代码等。

如果具备硬件实验条件，可以另行引入简单的数字逻辑基础实验、基本的计算机组成部件实验或者简单的微机接口实验。

1	第1章 计算机的发展及其应用	1
2	第2章 计算机系统的组成	10
3	第3章 指令系统	15
4	第4章 中央处理器	25
5	第5章 总线系统	35
6	第6章 指令流水线	45
7	第7章 指令流水线	55
8	第8章 指令流水线	65
9	第9章 指令流水线	75
10	第10章 指令流水线	85
11	第11章 指令流水线	95
12	第12章 指令流水线	105
13	第13章 指令流水线	115
14	第14章 指令流水线	125
15	第15章 指令流水线	135
16	第16章 指令流水线	145
17	第17章 指令流水线	155
18	第18章 指令流水线	165
19	第19章 指令流水线	175
20	第20章 指令流水线	185
21	第21章 指令流水线	195
22	第22章 指令流水线	205
23	第23章 指令流水线	215
24	第24章 指令流水线	225
25	第25章 指令流水线	235
26	第26章 指令流水线	245
27	第27章 指令流水线	255
28	第28章 指令流水线	265
29	第29章 指令流水线	275
30	第30章 指令流水线	285
31	第31章 指令流水线	295
32	第32章 指令流水线	305
33	第33章 指令流水线	315
34	第34章 指令流水线	325
35	第35章 指令流水线	335
36	第36章 指令流水线	345
37	第37章 指令流水线	355
38	第38章 指令流水线	365
39	第39章 指令流水线	375
40	第40章 指令流水线	385
41	第41章 指令流水线	395
42	第42章 指令流水线	405
43	第43章 指令流水线	415
44	第44章 指令流水线	425
45	第45章 指令流水线	435
46	第46章 指令流水线	445
47	第47章 指令流水线	455
48	第48章 指令流水线	465
49	第49章 指令流水线	475
50	第50章 指令流水线	485
51	第51章 指令流水线	495
52	第52章 指令流水线	505
53	第53章 指令流水线	515
54	第54章 指令流水线	525
55	第55章 指令流水线	535
56	第56章 指令流水线	545
57	第57章 指令流水线	555
58	第58章 指令流水线	565
59	第59章 指令流水线	575
60	第60章 指令流水线	585
61	第61章 指令流水线	595
62	第62章 指令流水线	605
63	第63章 指令流水线	615
64	第64章 指令流水线	625
65	第65章 指令流水线	635
66	第66章 指令流水线	645
67	第67章 指令流水线	655
68	第68章 指令流水线	665
69	第69章 指令流水线	675
70	第70章 指令流水线	685
71	第71章 指令流水线	695
72	第72章 指令流水线	705
73	第73章 指令流水线	715
74	第74章 指令流水线	725
75	第75章 指令流水线	735
76	第76章 指令流水线	745
77	第77章 指令流水线	755
78	第78章 指令流水线	765
79	第79章 指令流水线	775
80	第80章 指令流水线	785
81	第81章 指令流水线	795
82	第82章 指令流水线	805
83	第83章 指令流水线	815
84	第84章 指令流水线	825
85	第85章 指令流水线	835
86	第86章 指令流水线	845
87	第87章 指令流水线	855
88	第88章 指令流水线	865
89	第89章 指令流水线	875
90	第90章 指令流水线	885
91	第91章 指令流水线	895
92	第92章 指令流水线	905
93	第93章 指令流水线	915
94	第94章 指令流水线	925
95	第95章 指令流水线	935
96	第96章 指令流水线	945
97	第97章 指令流水线	955
98	第98章 指令流水线	965
99	第99章 指令流水线	975
100	第100章 指令流水线	985

目 录

前言

教学建议

第 1 章 计算机系统概述 1

- 1.1 计算机的发展 1
 - 1.1.1 计算机发展概况 1
 - 1.1.2 微型计算机的发展 3
 - 1.1.3 Intel 80x86 系列处理器 5
- 1.2 冯·诺依曼计算机结构 6
 - 1.2.1 二进制编码 7
 - 1.2.2 存储程序和程序控制 8
 - 1.2.3 顺序执行 9
 - 1.2.4 组成部件 9
- 1.3 计算机系统的组成 10
 - 1.3.1 计算机的硬件组成 10
 - 1.3.2 计算机的总线结构 14
 - 1.3.3 计算机系统的层次结构 19
 - 1.3.4 计算机系统的软件组成 23
- 习题 28

第 2 章 数据表示 29

- 2.1 数制 29
 - 2.1.1 二进制和十六进制 29
 - 2.1.2 数制之间的转换 31
- 2.2 整数编码 33
 - 2.2.1 定点整数格式 33
 - 2.2.2 有符号整数编码 34
 - 2.2.3 整数的类型转换 38
 - 2.2.4 整数的加减运算及溢出 41
 - 2.2.5 整数的移位运算 46

- 2.2.6 整数运算的数学性质 47
- 2.3 字符编码 48
 - 2.3.1 ASCII 48
 - 2.3.2 Unicode 53
- 2.4 实数编码 53
 - 2.4.1 浮点数据格式 54
 - 2.4.2 浮点运算的数学性质 61
- 习题 64

第 3 章 数字逻辑基础 68

- 3.1 逻辑代数 68
 - 3.1.1 逻辑关系 68
 - 3.1.2 逻辑代数运算规则 73
 - 3.1.3 逻辑函数的形式、转换及化简 76
- 3.2 逻辑门电路 78
 - 3.2.1 门电路的实现 78
 - 3.2.2 集成电路 80
 - 3.2.3 三态门 82
- 3.3 组合逻辑电路 83
 - 3.3.1 编码器 84
 - 3.3.2 译码器 85
 - 3.3.3 加法器 87
 - 3.3.4 多路开关 88
- 3.4 时序逻辑电路 88
 - 3.4.1 触发器 89
 - 3.4.2 寄存器 93
 - 3.4.3 计数器 94
- 3.5 可编程逻辑器件 94
 - 3.5.1 PLD 概述 95
 - 3.5.2 电子设计自动化 96
- 习题 99

第4章 处理器	101	习题	172
4.1 处理器的组成	101	第6章 汇编语言程序设计	177
4.1.1 控制器	101	6.1 顺序程序结构	177
4.1.2 运算器	103	6.2 分支程序结构	181
4.2 处理器的结构	103	6.2.1 指令寻址	181
4.2.1 处理器的基本结构	104	6.2.2 无条件转移指令	182
4.2.2 8086的功能结构	105	6.2.3 条件转移指令	184
4.2.3 80386的功能结构	106	6.2.4 单分支程序结构	190
4.2.4 Pentium的功能结构	107	6.2.5 双分支程序结构	191
4.3 寄存器	109	6.2.6 多分支程序结构	192
4.3.1 通用寄存器	109	6.3 循环程序结构	195
4.3.2 专用寄存器	111	6.3.1 循环指令	195
4.4 存储器组织	113	6.3.2 计数控制循环	198
4.4.1 存储模型	113	6.3.3 条件控制循环	199
4.4.2 工作方式	114	6.3.4 多重循环	200
4.4.3 逻辑地址	115	6.4 子程序	201
习题	119	6.4.1 子程序指令	202
第5章 指令系统	121	6.4.2 子程序设计	205
5.1 指令格式	121	6.4.3 寄存器传递参数	208
5.1.1 指令编码	121	6.4.4 共享变量传递参数	210
5.1.2 IA-32指令格式	124	6.4.5 堆栈传递参数	212
5.2 汇编语言基础	126	6.4.6 堆栈帧	215
5.2.1 汇编语言的语句格式	126	习题	223
5.2.2 汇编语言的源程序框架	128	第7章 存储系统	229
5.2.3 汇编语言的开发过程	132	7.1 存储系统的层次结构	229
5.2.4 DEVC中C语言的开发过程	137	7.1.1 技术指标	230
5.3 汇编语言的常量和变量	141	7.1.2 层次结构	231
5.3.1 常量表达	141	7.1.3 局部性原理	232
5.3.2 变量应用	143	7.2 主存储器	235
5.4 数据寻址	150	7.2.1 读写存储器	236
5.4.1 立即数寻址	150	7.2.2 ROM	241
5.4.2 寄存器寻址	151	7.2.3 半导体存储器的连接	245
5.4.3 存储器寻址	151	7.3 高速缓冲存储器	249
5.4.4 数据寻址的组合	158	7.3.1 高速缓存的工作原理	249
5.5 通用数据处理指令	159	7.3.2 地址映射	252
5.5.1 数据传送类指令	160	7.3.3 替换算法	257
5.5.2 算术运算类指令	164	7.3.4 写入策略	258
5.5.3 位操作类指令	168	7.4 存储管理	261

7.4.1 虚拟存储器	261	第 9 章 处理器性能提高技术	308
7.4.2 段式存储管理	262	9.1 精简指令集计算机技术	308
7.4.3 页式存储管理	264	9.1.1 复杂指令集和精简指令集	308
习题	267	9.1.2 MIPS 处理器	311
第 8 章 输入 / 输出接口	269	9.2 指令流水线技术	313
8.1 I/O 接口概述	269	9.2.1 指令流水线思想	313
8.1.1 I/O 接口的典型结构	269	9.2.2 80486 的指令流水线	316
8.1.2 I/O 端口的编址	271	9.3 并行处理技术	318
8.1.3 I/O 指令	273	9.3.1 并行性概念	318
8.2 外设数据的传送方式	274	9.3.2 数据级并行	319
8.2.1 无条件传送	275	9.3.3 指令级并行	321
8.2.2 程序查询传送	277	9.3.4 线程级并行	325
8.2.3 中断传送	279	习题	329
8.2.4 中断控制系统	283	附录 A 32 位通用指令列表	331
8.2.5 DMA 传送	286	附录 B MASM 伪指令和操作符列表	336
8.3 常用接口技术	289	附录 C I/O 子程序库	337
8.3.1 定时控制接口	289	附录 D 列表文件符号说明	339
8.3.2 并行接口	297	附录 E 常见汇编错误信息	340
8.3.3 异步串行通信接口	300	参考文献	343
8.3.4 模拟接口	304		
习题	305		

了 ENIAC (Electronic Numerical Integrator And Calculator)。ENIAC 最初用于为军队编制各种武器的弹道表, 后经多次改进, 成为能进行各种科学计算的通用计算机。在 1946 年得以公开。

1 计算机发展简史

计算机的发展突飞猛进, 经历了电子管、晶体管、中小规模集成电路和超大规模集成电路 4 个阶段。

- 第一代计算机 (以第一台计算机 ENIAC 问世开始到 20 世纪 50 年代末期)。第一代计算机的主要特点是使用电子管作为逻辑器件, 软件还处于初始阶段, 使用机器语言与符号语言编制的程序。

第一代计算机是计算机发展的初级阶段, 其体积比较大, 运算速度也比较慢, 存储容量不大。为了解决一个问题, 所编制的程序很复杂。这一代计算机主要用于科学计算。

- 第二代计算机 (从 20 世纪 50 年代末期到 60 年代初期)。第二代计算机的主要特点是使用晶体管作为电子器件, 在软件方面开始使用计算机高级语言, 为更多的人学习和使用计算机铺平了道路。

这一代计算机的体积大大减小, 具有重量轻、寿命长、耗电少、运算速度快、存储容量比较大等优点。因此, 这一代计算机不仅用于科学计算, 还用于数据处理

第1章 计算机系统概述

数字电子计算机经历了电子管、晶体管、集成电路为主要部件的时代。随着大规模集成电路的应用,计算机的功能越来越强大、体积却越来越小,微型计算机(简称微型机或微机)应运而生,并成为通用计算机的主要应用形式。本书基于 Intel 80x86 处理器和个人微机(PC)硬件平台,结合 C 语言和汇编语言的软件编程,介绍计算机系统的工作原理。

1.1 计算机的发展

计算机的诞生和发展是 20 世纪重要的科技成果之一。微型计算机的应用深入人类社会的方方面面,极大地改变了人们的工作、学习和生活方式,成为 21 世纪不可或缺的通用性工具。

1.1.1 计算机发展概况

美国宾夕法尼亚大学摩尔学院的莫克利(J.W.Mauchly)和埃克特(J.P.Eckert)制造了世界上第一台电子数字通用计算机——ENIAC(Electronics Numerical Integrator And Calculator)。ENIAC 最初用于为军队编制各种武器的弹道表,后经多次改进,成为能进行各种科学计算的通用计算机,在 1946 年得以公开。

1. 计算机发展简史

计算机的发展突飞猛进,经历了电子管、晶体管、中小规模集成电路和超大规模集成电路 4 个阶段。

- 第一代计算机(以第一台计算机 ENIAC 问世开始到 20 世纪 50 年代末期)。第一代计算机的主要特征是使用电子管作为逻辑器件,软件还处于初始阶段,使用机器语言与符号语言编制程序。

第一代计算机是计算机发展的初级阶段,其体积比较大,运算速度也比较低,存储容量不大。为了解决一个问题,所编制的程序很复杂。这一代计算机主要用于科学计算。

- 第二代计算机(从 20 世纪 50 年代末期到 60 年代初期)。第二代计算机的主要特征是使用晶体管作为电子器件,在软件方面开始使用计算机高级语言,为更多的人学习和使用计算机铺平了道路。

这一代计算机的体积大大减小,具有质量轻、寿命长、耗电少、运算速度快、存储容量比较大等优点。因此,这一代计算机不仅用于科学计算,还用于数据处理

和事务处理，并逐渐用于工业控制。

- 第三代计算机（从 20 世纪 60 年代中期到 70 年代初期）。第三代计算机的主要特征是使用中小规模集成电路作为电子器件，出现了操作系统，计算机的功能越来越强，应用范围越来越广。

使用中小规模集成电路制成的计算机，其体积与功耗都得到了进一步减小，可靠性和运算速度等指标也得到了进一步提高，并且为计算机的小型化、微型化提供了良好的条件。在这一时期中，计算机不仅用于科学计算，还用于文字处理、企业管理、自动控制等领域，出现了计算机技术与通信技术相结合的管理信息系统，可用于生产管理、交通管理、情报检索等领域。

- 第四代计算机（20 世纪 70 年代初期至今）。第四代计算机是指用大规模与超大规模集成电路作为电子器件制成的计算机。这一代计算机的各种性能都得到了大幅度提高，应用的软件也越来越丰富，其应用涉及国民经济的各个领域，已经在办公室自动化、数据库管理、图像识别、语音识别、专家系统等众多领域大显身手，并且进入了家庭。

1971 年以来，作为第四代计算机重要产品的微型计算机得到了飞速的发展，对计算机的普及起到了决定性的作用。计算机的应用有力地推动了国民经济的发展和科学技术的进步，同时也对计算机技术提出了更高的要求，从而促进了计算机进一步发展。现代计算机系统以超大规模集成电路为基础，具有运算速度快、存储容量大、功能强劲的特点，不仅实现了微型化和网络化，也将进一步实现智能化。

2. 摩尔定律

从利用算盘实现机械式计算到电子计算机出现，期间经历了千年历史。但从 1946 年第一台通用数字电子计算机 ENIAC 开始到计算机广泛应用的信息时代，却只有短短的几十年时间。大规模集成电路生产技术的不断提高推动了计算机的飞速发展。摩尔定律 (Moore's Law) 很好地说明了这个现象。

1965 年，美国 Intel 公司的创始人之一摩尔 (G. Moore) 预言：集成电路的晶体管密度每年将翻倍。现在这个预言通常被表达为：每隔 18 个月硅片密度（晶体管容量）将翻倍。也常被表达为：每 18 个月，集成电路的性能将提高一倍，而其价格将降低一半。这个预言就是所谓的摩尔定律。摩尔预计这个规律会持续 10 年，而事实上这个规律已经持续了 50 多年，也许还将继续维持 5 年或 10 年。

伴随着摩尔定律，我们看到原来封闭在机房的庞大计算机系统已经走入普通家庭，成为人们日常使用的桌面微机、平板电脑和智能终端等。事实上，以微处理器为基础的计算机在整个计算机设计领域占据统治地位。工作站和个人计算机 (Personal Computer, PC) 成为计算机工业的主要产品，使用微处理器的服务器取代了传统的小型机，大型机则几乎由通用微处理器组成的多处理器系统取代，甚至高端的巨型机也采用大量微处理器构成。更不用说形影难离的智能手机、无处不在的嵌入式计算机，它们正改变着我们应用计算机的方式。因此，体型微小的微处理器从性能等方面来讲已经不再“微弱”。于是，人们自然地将微处理器简称为处理器，而本书论述的计算机系统也主要指微型计

算机系统,具体的实例则是通用个人微机(PC,包括台式电脑、笔记本电脑等)。

但是,摩尔定律不会永远持续,电子器件的物理极限在悄然逼近。20世纪80年代中期以前,处理器的性能提高主要由工艺技术驱动。此后,处理器的性能提高更多地得益于计算机系统结构的革新。从通用寄存器结构、精简指令集计算机(RISC)、高速缓冲存储器(Cache)、虚拟存储器管理,到指令级并行、线程级并行、单芯片多核心等并行技术,先进的系统结构已经成为提高处理器性能的主要推动力。

1.1.2 微型计算机的发展

在巨型机、大型机、小型机和微型机等各类计算机中,微型计算机(Microcomputer)是性能、价格、体积较小的一类。在科学计算、信息管理、自动控制、人工智能等应用领域中,微型计算机是最常见的一类。工作、学习和娱乐中使用的桌面个人微机(PC)是我们最熟悉且最典型的微型机系统;支撑网络的文件服务器、WWW服务器等各类服务器属于高档微型机系统;生产生活中运用的各种智能化电子设备从计算机系统角度看同样也是微型机系统,只不过作为其控制核心的处理器常被封装在电子设备内部,不易被觉察,因此常称它们为嵌入式计算机系统。桌面系统、服务器和嵌入式计算构成现代计算机的主要应用形式,微型机都是其中的主角。

计算机的运算和控制核心称为处理器(Processor),即中央处理单元(Central Processing Unit, CPU)。微型机中的处理器常采用一块大规模集成电路芯片,所以也被称为微处理器(Microprocessor),它代表着整个微型机系统的性能。因此,通常将以微处理器为核心构造的计算机称为微型计算机。

处理器的性能经常用字长(Word)、时钟频率、集成度等基本的技术参数反映。字长表明处理器每个时间单位可以处理的二进制数据位数,如一次运算、传输的位数。时钟频率表明处理器的处理速度,反映了处理器的基本时间单位。集成度表明处理器的生产工艺水平,通常用芯片上集成的晶体管数量来表达。晶体管只是一个由电子信号控制的电子开关。集成电路在一个芯片上集成成千上万的晶体管完成特定功能。

1. 通用微处理器

1971年,美国Intel公司为日本制造商设计可编程计算器时,把采用多个专用芯片的方案修改成一个通用处理器,于是诞生了世界上第一个微处理器Intel 4004。Intel 4004微处理器字长4位,集成了约2300个晶体管,时钟频率为108kHz。以它为核心组成的MCS-4计算机是世界上第一台微型计算机。随后,Intel 4004被改进为Intel 4040。

1972年,Intel公司研制出字长8位的微处理器芯片8008,其时钟频率为500kHz,集成了约3500个晶体管。这之后的几年当中,微处理器开始走向成熟,出现了以Motorola公司M6800、Zilog公司Z80和Intel公司8080/8085为代表的中、高档8位微处理器。Apple(苹果)公司的Apple机就是这一时期著名的个人微型机。

1978年开始,各公司相继推出一批16位字长的微处理器,如Intel公司的8086和8088、Motorola公司的M68000、Zilog公司的Z8000等。例如,Intel 8086的时钟频率

为 5MHz，集成了多达 2.9 万个晶体管。这一时期的著名微机产品是 IBM 公司采用 Intel 公司微处理器、Microsoft 操作系统开发的 16 位 PC。

1985 年，Intel 公司借助 IBM PC 的巨大成功，进一步推出了 32 位微处理器 Intel 80386，其集成了多达 27.5 万个晶体管，时钟频率达 16MHz。从这时起，微处理器步入快速发展阶段。就 Intel 公司来说，它陆续研制生产了 80486、Pentium（奔腾）、Pentium Pro（高能奔腾）、MMX Pentium（多能奔腾）、Pentium II、Pentium III 和 Pentium 4 等微处理器。例如，2003 年 Intel 公司生产的 Pentium 4 处理器，具有 1.25 亿个晶体管，时钟频率达到 3.4GHz。兼容 IBM PC 的 32 位 PC、Apple 公司的 Macintosh 机等，在这个时期得到飞速发展，伴随着多媒体技术和互联网络，这些 PC 成为我们工作和生活中不可缺少的一部分。

2000 年，Intel 公司在微型机的高端产品服务器中使用了 64 位字长的新一代微处理器 Itanium（安腾）。事实上，其他公司的 64 位微处理器在 20 世纪 90 年代已经出现，但也主要应用于服务器产品中，不能与通用 80x86 微处理器兼容。2003 年 4 月，AMD 公司推出首款兼容 32 位 80x86 结构的 64 位微处理器，称为 x86-64 结构。2004 年 3 月，Intel 公司也发布了首款扩展 64 位能力的 32 位微处理器，它采用扩展 64 位主存技术 EM64T（Extended Memory 64 Technology）。64 位微处理器主要将整数运算和主存寻址能力扩大到 64 位，极大地提高了微型机的处理能力，后被称为 Intel 64 结构。2005 年以后，采用 64 位技术的桌面微型机逐渐获得用户青睐。与此同时，生产厂商已经可以在一个半导体芯片上制作两个微处理器核心，原来面向高端的并行处理器技术开始走向桌面系统，微型计算机系统也进入了一个全新的 64 位多核处理器阶段。

2. 专用微处理器

除了装在 PC、笔记本电脑、工作站、服务器上的通用微处理器（常简称为 MPU）外，还有其他应用领域的专用微处理器：单片机（微控制器）和数字信号处理器。

单片机（Single Chip Microcomputer）是指通常用于控制领域的微处理器芯片，其内部除处理器外还集成了计算机的其他一些主要部件，如主存储器、定时器、并行接口、串行接口，有的芯片还集成了模拟/数字、数字/模拟转换电路等。换句话说，一个芯片几乎就是一个计算机，只要配上少量的外部电路和设备，就可以构成具体的应用系统。

单片机是国内习惯的名称，国际上多称其为微控制器（Micro Controller）或嵌入式控制器（Embedded Controller），简称为 MCU。微控制器的初期阶段（1976～1978 年）以 Intel 公司的 8 位 MCS-48 系列为代表。1978 年以后，微控制器进入普及阶段，以 8 位为主，最著名的是 Intel 公司的 8 位 MCS-51 系列，还有 Atmel 公司的 8 位 AVR 系列、Microchip Technology 公司的 PIC 系列。1982 年以后，出现了高性能的 16 位、32 位微控制器，如 Intel 公司的 MCS-96/98 系列，尤其是基于 ARM（Advanced RISC Machine）核心的微处理器。ARM 核心采用精简指令集 RISC 结构，具有耗电少、成本低、性能高的特点，因此使用 ARM 为核心研制的各种微处理器已经广泛应用于 32 位嵌入式系统，如 Cortex-M3/M4 微控制器。而面向高性能应用领域的 ARM 核心则是 Cortex-A 系列，主要应用于移动通信领域，如智能手机和平板电脑。目前，高端专用微处理器也实现了

64 位处理, 并支持多核技术。

数字信号处理器 (Digital Signal Processor) 简称为 DSP 芯片, 实际上也是一种微控制器 (单片机), 但更专注于数字信号的高速处理, 内部集成有高速乘法器, 能够进行快速乘法和加法运算。自 1979 年 Intel 公司开发 Intel 2920 以后, DSP 芯片也经历了多代发展, 其中美国德州仪器 TI (Texas Instruments) 公司的 TMS320 各代产品具有代表性, 如 1982 年的 TMS32010、1985 年的 TMS320C20、1987 年的 TMS320C30、1991 年的 TMS320C40, 以及 TMS320C2000 / TMS320C5000 / TMS320C6000 系列等。DSP 芯片市场主要分布在通信、消费类电子产品和计算机领域。我国推广和应用较多的是 TI 公司、AD 公司和 Motorola 公司的 DSP 芯片。

利用微控制器、数字信号处理器或通用微处理器, 结合具体应用就可以构成一个控制系统, 如当前的主要应用形式: 嵌入式系统。嵌入式系统融合了计算机软硬件技术、通信技术和半导体微电子技术, 把计算机直接嵌入应用系统之中, 构造信息技术 (Information Technology, IT) 的最终产品。

自从 20 世纪 70 年代微处理器产生以来, 它就一直沿着通用处理器、微控制器和数字信号处理器 (DSP) 3 个方向发展。这 3 类微处理器的基本工作原理一样, 但各有特点, 技术上它们不断地相互借鉴和交融, 应用却不尽相同。

1.1.3 Intel 80x86 系列处理器

美国 Intel 公司是目前世界上最有影响的处理器生产厂家, 也是世界上第一个生产微处理器芯片的厂家。其生产的 80x86 系列处理器一直是个人微机的主流处理器。Intel 80x86 系列处理器的发展就是微型计算机发展的一个缩影。

1. 16 位 80x86 处理器

1971 年, Intel 公司生产的 4 位处理器芯片 Intel 4004 宣告了微型计算机时代的到来。1972 年, Intel 公司开发了 8 位处理器芯片 Intel 8008; 1974 年接着生产了 Intel 8080; 1977 年, Intel 公司将 8080 及其支持电路集成在一块电路芯片上, 形成了性能更高的 8 位处理器 8085。1978 年, Intel 公司在其 8 位处理器基础上, 陆续推出了 16 位结构的 8086、8088 和 80286 等处理器, 它们在 IBM PC 系列中获得广泛应用, 被称为 16 位 80x86 处理器。

Intel 公司于 1978 年推出的 16 位 8086 处理器, 是该公司生产的第一个 16 位芯片。8086 支持的所有指令, 即指令系统 (Instruction Set) 成为整个 Intel 80x86 系列处理器的 16 位基本指令集。其随后推出的 80186 和 80286 处理器, 增加了若干实用指令。

2. IA-32 处理器

IBM PC 系列机的广泛应用推动了处理器芯片的生产。Intel 公司在推出 32 位的 80386 处理器后, 明确宣布 Intel 80386 芯片的指令集结构 (Instruction Set Architecture, ISA) 被确定为以后开发的 80x86 系列处理器的标准, 称为 Intel 32 位结构——IA-32 (Intel Architecture-32)。现在, Intel 公司的 80386、80486 以及奔腾 (Pentium) 各代处

理器被通称为 IA-32 处理器或 32 位 80x86 处理器。

IA-32 指令系统在兼容原 16 位 80286 指令系统基础上，全面升级为 32 位，还新增了有关位操作、条件设置等指令。Intel 80386/80486/Pentium/Pentium Pro 等 IA-32 处理器新增若干实用指令，但非常有限。为了顺应技术向多媒体和通信方向发展的潮流，Intel 公司在后续处理器加入了多媒体扩展指令（统称为 SIMD 指令），形成了 Pentium MMX（多能奔腾）、Pentium II、Pentium III 和 Pentium 4 处理器。

3. Intel 64 处理器

互联网、多媒体、3D 视频等的广泛应用，对计算机性能提出了越来越高的要求。Intel、AMD、IBM、Sun 等厂商已陆续设计并推出了多种采用 RISC 结构的 64 位处理器。但是，这些 64 位处理器主要面向服务器和工作站等高端应用，不能兼容通用 PC。例如，Intel 公司于 2000 年推出 64 位 Itanium（安腾）处理器，2002 年又推出 Itanium 2 处理器。Intel 公司称该处理器的指令集结构为 Intel 64 位结构（IA-64），以区别于原来的 Intel 32 位结构（IA-32）。虽然这两个名称似乎有继承性，但实际上，IA-64 结构根本不是 IA-32 结构的 64 位扩展。

一直以来，80x86 处理器的更新换代都保持与早期处理器的兼容，以便继续使用现有的软硬件资源。但是，Intel 公司迟迟不愿将 80x86 处理器扩展为 64 位，这给了 AMD 公司（它是生产 IA-32 处理器兼容芯片的厂商）一个机会。它于 2003 年 9 月率先推出支持 64 位、兼容 80x86 指令集结构的 Athlon 64 处理器（K8 核心），将桌面 PC 引入到了 64 位领域。

2005 年，在 PC 用户对 64 位技术的企盼和 AMD 公司 64 位处理器产品的压力下，Intel 公司推出了扩展存储器 64 位技术（Intel Extended Memory 64 Technology, Intel EM64T）。EM64T 技术是 IA-32 结构的 64 位扩展，首先应用于支持超线程技术的 Pentium 4 终极版（支持双核技术）和 6xx 系列 Pentium 4 处理器。随着 EM64T 技术的出现，IA-32 指令系统也扩展成 64 位，现被称为 Intel 64 结构（Intel 64 Architecture）。这之后的 Pentium 4、酷睿（Core）2 和酷睿 i 等系列多核处理器都支持 Intel 64 结构，可称为 Intel 64 处理器。

Intel 64 结构为软件提供了更大的存储空间。在新增的 64 位工作方式下，大多数指令升级支持对 64 位数据的处理，还新增了一些 64 位指令，但也有少数指令在 64 位方式下不再应用。

1.2 冯·诺依曼计算机结构

第一台电子数字通用计算机 ENIAC 用电子管实现，采用字长为 10 位的十进制计数方式。ENIAC 可以编程，但只有很少的存储空间。其编程通过手工拔插电缆和拨动开关完成，通常需要半小时到一天的时间。莫克利和埃克特提出了改进程序输入方式的设想，希望能够像存储数据那样存储程序代码。

1944 年，冯·诺依曼被 ENIAC 项目吸引，并在一份备忘录中提出了能够存储程序

的计算机设计构想。Herman Goldstine 发表了这份备忘录，并冠以冯·诺依曼的名字。这样，术语“冯·诺依曼计算机”被广泛引用，它代表存储程序计算机结构，并成为现代计算机的基本特征。冯·诺依曼计算机的主要设计思想如下：

- 采用二进制形式表示数据和指令。指令由操作码和地址码组成。
- 将程序和数据存放在存储器中，计算机在工作时从存储器取出指令加以执行，自动完成计算任务。这就是“存储程序”和“程序控制”（简称存储程序控制）的概念。
- 指令的执行是顺序的，即一般按照指令在存储器中存放的顺序执行，程序分支由转移指令实现。
- 计算机由存储器、运算器、控制器、输入设备和输出设备 5 大基本部件组成，并规定了这 5 部分的基本功能。

1.2.1 二进制编码

冯·诺依曼计算机的基本思想之一是采用二进制（Binary）形式表示数据（Data）和指令（Instruction）。这说明现实中的一切信息（数据），包括控制计算机操作的指令，在计算机中都是一串“0”和“1”数码。这串数码是按照一定规律组合起来的，即二进制编码规则。不同的信息用不同的数码表示，同样的信息也可以用不同的编码规则、不同的数码表示（以便计算机进行不同的处理）。

指令是控制计算机操作的基本命令，是处理器不需要翻译就能识别（直接执行）的“母语”，即机器语言。程序虽然可以用 C/C++ 或 Java 等高级语言编写，但需要由编译程序或解释程序翻译成指令，才可以由处理器执行，所以程序是由指令构成的。

指令的二进制编码规则形成了指令的代码格式，它由操作码和地址码组成。指令的操作码（Opcode）表明指令的操作，如数据传送、加法运算等基本操作。操作数（Operand）是参与操作的数据，主要以寄存器或存储器地址形式指明数据的来源，所以也称为地址码。例如，数据传送指令的源地址和目的地址，加法指令的加数、被加数以及和值都是操作数。

二进制只支持“0”和“1”两个数码，可以表示电源的关（Off）和开（On）两种状态，对应数字信号的低电平（Low）和高电平（High）。数字计算机中信息的最基本单位就是一个二进制位（所以，计算机专业书籍中的“位”常常是二进制位，而不是日常生活中的十进制位），或称为比特（Bit）。4 个二进制位称为半字节（Nibble），8 个二进制位构成一个字节（Byte）。IBM PC 系列微机以 16 位结构的 Intel 8086 和 Intel 80286 为处理器，并获得广泛应用，所以 Intel 80x86 系列处理器常称 16 位数据为一个字（Word），这样 32 位数据被称为双字（Double Word），64 位数据被称为 4 字（Quad Word）。

这里应该再次明确一下处理器字长的概念，处理器字长指单位时间处理器可以处理的二进制数据位数。因此，字长（位数）随处理器不同，可以是 8 位、16 位、32 位或 64 位（但字节总是表示 8 位）。例如，Linux 操作系统最初基于 32 位 Intel 80386 处理器开发，就定义一个字的数据位数是 32 位，相应地称 16 位数据为半字（Half Word），而称 64 位数据为双字。

数据用二进制位表达，就是一串 0 和 1 组成的序列。表达数据时，通常按日常书写