



全国计算机技术与软件专业技术资格（水平）考试指定用书

软件设计师教程

褚华 霍秋艳 主编

全国计算机专业技术资格考试办公室 组编



清华大学出版社

第5版

全国计算机技术与软件专业技术资格（水平）

软件设计师教程

（第5版）

褚华 霍秋艳 主编

常州大学图书馆
藏书章

清华大学出版社
北京

内 容 简 介

本书作为中级职称的全国计算机技术与软件专业技术资格（水平）考试（简称“软考”）指定教材，具有比较权威的指导意义。本书根据《软件设计师考试大纲》（2018年审定通过）的重点内容，组织了12章的内容，考生在学习教材内容的同时，还须对照考试大纲，认真学习和复习大纲的知识点。

本书是在《软件设计师考试大纲》的指导下，对《软件设计师教程（第4版）》进行了认真修编，部分章节是重写后形成的。在本书中，强化了软件工程部分的知识，增加了Web应用系统分析与设计知识。

本书适合参加本级别考试的考生和大学在校生作为教材。

本书扉页为防伪页，封面贴有清华大学出版社防伪标签，无上述标识者不得销售。

版权所有，侵权必究。侵权举报电话：010-62782989 13701121933

图书在版编目（CIP）数据

软件设计师教程/褚华，霍秋艳主编。—5版。—北京：清华大学出版社，2018（2018.3重印）
（全国计算机技术与软件专业技术资格（水平）考试指定用书）

ISBN 978-7-302-49122-4

I. ①软… II. ①褚… ②霍… III. ①软件设计-资格考试-自学参考资料 IV. ①TP311.5

中国版本图书馆CIP数据核字（2017）第313210号

责任编辑：杨如林 柴文强

封面设计：常雪影

责任校对：徐俊伟

责任印制：沈 露

出版发行：清华大学出版社

网 址：<http://www.tup.com.cn>, <http://www.wqbook.com>

地 址：北京清华大学学研大厦A座 邮 编：100084

社总机：010-62770175 邮 购：010-62786544

投稿与读者服务：010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈：010-62772015, zhiliang@tup.tsinghua.edu.cn

印 刷 者：清华大学印刷厂

装 订 者：北京市密云县京文制本装订厂

经 销：全国新华书店

开 本：185mm×230mm 印 张：42.75 防伪页：1 字 数：930千字

版 次：2004年7月第1版 2018年2月第5版 印 次：2018年3月第2次印刷

印 数：5001～10000

定 价：119.00元

产品编号：075520-01

前 言

全国计算机技术与软件专业技术资格（水平）考试实施至今已经历了二十余年，在社会上产生了很大的影响，对我国软件产业的形成和发展做出了重要的贡献。为了适应我国计算机信息技术发展的需求，人力资源和社会保障部、工业和信息化部决定将考试的级别拓展到计算机信息技术行业的各个方面，以满足社会上对各种计算机信息技术人才的需要。

编者受全国计算机专业技术资格考试办公室委托，对《软件设计师教程（第4版）》进行改写，以适应新的考试大纲要求。在考试大纲中，要求考生掌握的知识面很广，每个章节的内容都能构成相关领域的一门甚至多门课程，因此编写的难度很高。考虑到参加考试的人员已有一定的基础，所以本书中只对考试大纲中所涉及的知识领域的要点加以阐述，但限于篇幅所限，不能详细地展开，请读者谅解。

全书共分12章，各章节内容安排如下：

第1章主要介绍计算机系统基础知识、计算机体系结构以及安全性、可靠性和系统性能评测基础。

第2章主要介绍程序设计语言的基本概念与基本成分，阐述了汇编程序、编译程序与解释程序的基本原理。

第3章主要介绍数据结构的基础知识，包括线性结构、数组、广义表、树和图，以及查找和排序的基本算法。

第4章主要介绍操作系统基本概念与分类及特点、进程管理、存储管理、设备管理、文件管理、作业管理等。

第5章主要介绍软件工程中软件过程与过程模型、需求分析与需求工程、系统设计、系统测试、系统运行与维护、软件项目管理、软件质量、软件度量、软件工具与软件开发环境基础知识。

第6章主要介绍系统分析与设计、结构化分析与设计、Web应用系统分析与设计、用户界面设计基础知识。

第7章主要介绍面向对象的基本概念和面向对象开发技术，包括面向对象的分析与设计方法，UML以及设计模式的概念和应用。

第8章主要介绍算法设计与分析的基本概念，包括分治法、动态规划法、贪心法、回溯法、分支界限法、概率算法、近似算法、数据挖掘算法及智能优化算法。

第9章主要介绍数据库的基本概念、数据模型、关系代数、SQL语言、规范化理论和事务处理等控制功能。

第10章主要介绍网络与信息安全基础知识，包括网络体系结构、网络互连设备、网络构件、网络协议、网络应用、信息安全和网络安全方面的基础知识。

第11章主要介绍标准化与知识产权基础知识。

第12章主要介绍结构化分析与设计、数据库分析与设计、面向对象分析与设计、算法分析与设计以及面向过程、面向对象的程序设计与实现。

本书第1章由张淑平、马志欣编写，第2章由张淑平编写，第3章由张淑平、陈静玉、宋胜利编写，第4章由王亚平编写，第5章、第6章、第7章由霍秋艳、褚华编写，第8章由覃桂敏、褚华编写，第9章由王亚平编写，第10章由严体华编写，第11章由刘强编写，第12章由王亚平、褚华、霍秋艳、覃桂敏、张淑平编写，最后由霍秋艳、褚华统稿。

在本书的编写过程中，参考了许多相关的书籍和资料，编者在此对这些参考文献的作者表示感谢。同时感谢清华大学出版社在本书出版过程中所给予的支持和帮助。

因水平有限，书中难免存在欠妥之处，望读者指正，以利改进和提高。

编者
2018年1月

目 录

第1章 计算机网络概论	1	3.2 数组、矩阵和广义表	113
1.1 计算机系统基础知识	1	3.2.1 数组	113
1.1.1 计算机系统硬件基本组成	1	3.2.2 矩阵	115
1.1.2 中央处理单元	1	3.2.3 广义表	116
1.1.3 数据表示	4	3.3 树	118
1.1.4 校验码	10	3.3.1 树与二叉树的定义	118
1.2 计算机体系结构	12	3.3.2 二叉树的性质与存储结构	119
1.2.1 计算机体系结构的发展	12	3.3.3 二叉树的遍历	122
1.2.2 存储系统	20	3.3.4 线索二叉树	125
1.2.3 输入/输出技术	31	3.3.5 最优二叉树	126
1.2.4 总线结构	35	3.3.6 树和森林	130
1.3 安全性、可靠性与系统性能评测		3.4 图	133
基础知识	38	3.4.1 图的定义与存储	134
1.3.1 计算机安全概述	38	3.4.2 图的遍历	138
1.3.2 加密技术和认证技术	40	3.4.3 生成树及最小生成树	140
1.3.3 计算机可靠性	48	3.4.4 拓扑排序和关键路径	143
1.3.4 计算机系统的性能评价	51	3.4.5 最短路径	146
第2章 程序设计语言基础知识	56	3.5 查找	149
2.1 程序设计语言概述	56	3.5.1 查找的基本概念	149
2.1.1 程序设计语言的基本概念	56	3.5.2 静态查找表的查找方法	150
2.1.2 程序设计语言的基本成分	61	3.5.3 动态查找表	154
2.2 语言处理程序基础	67	3.5.4 哈希表	161
2.2.1 汇编程序基本原理	67	3.6 排序	165
2.2.2 编译程序基本原理	69	3.6.1 排序的基本概念	165
2.2.3 解释程序基本原理	96	3.6.2 简单排序	165
第3章 数据结构	99	3.6.3 希尔排序	168
3.1 线性结构	99	3.6.4 快速排序	169
3.1.1 线性表	99	3.6.5 堆排序	170
3.1.2 栈和队列	104	3.6.6 归并排序	173
3.1.3 串	108	3.6.7 基数排序	174

3.6.8	内部排序方法小结	175	4.6.1	作业与作业控制	235
3.6.9	外部排序	176	4.6.2	作业调度	236
第4章 操作系统知识			4.6.3	用户界面	238
4.1	操作系统概述	180	第5章 软件工程基础知识		
4.1.1	操作系统的基本概念	180	5.1	软件工程概述	239
4.1.2	操作系统分类及特点	181	5.1.1	计算机软件	240
4.1.3	操作系统的发展	185	5.1.2	软件工程基本原理	241
4.2	进程管理	185	5.1.3	软件生存周期	243
4.2.1	基本概念	185	5.1.4	软件过程	245
4.2.2	进程的控制	189	5.2	软件过程模型	247
4.2.3	进程间的通信	189	5.2.1	瀑布模型(Waterfall Model)	248
4.2.4	管程	193	5.2.2	增量模型(Incremental Model)	249
4.2.5	进程调度	195	5.2.3	演化模型(Evolutionary Model)	250
4.2.6	死锁	198	5.2.4	喷泉模型 (Water Fountain Model)	252
4.2.7	线程	202	5.2.5	基于构件的开发模型 (Component-based Development Model)	252
4.3	存储管理	202	5.2.6	形式化方法模型(Formal Methods Model)	253
4.3.1	基本概念	203	5.2.7	统一过程(UP)模型	253
4.3.2	存储管理方案	204	5.2.8	敏捷方法(Agile Development)	254
4.3.3	分页存储管理	205	5.3	需求分析	256
4.3.4	分段存储管理	208	5.3.1	软件需求	256
4.3.5	段页式存储管理	209	5.3.2	需求分析原则	257
4.3.6	虚拟存储管理	211	5.3.3	需求工程	257
4.4	设备管理	216	5.4	系统设计	260
4.4.1	设备管理概述	216	5.4.1	概要设计	261
4.4.2	I/O软件	217	5.4.2	详细设计	262
4.4.3	设备管理采用的相关技术	218	5.5	系统测试	262
4.4.4	磁盘调度	221	5.5.1	系统测试与调试	262
4.5	文件管理	224	5.5.2	传统软件的测试策略	264
4.5.1	文件与文件系统	224	5.5.3	测试面向对象软件	271
4.5.2	文件的结构和组织	225			
4.5.3	文件目录	227			
4.5.4	存取方法和存储空间的管理	229			
4.5.5	文件的使用	231			
4.5.6	文件的共享和保护	231			
4.5.7	系统的安全与可靠性	233			
4.6	作业管理	234			

5.5.4	测试 Web 应用	272	6.3.1	结构化设计的步骤	337
5.5.5	测试方法	273	6.3.2	数据流图到软件体系结构的映射	338
5.5.6	调试	276	6.4	WebApp 分析与设计	340
5.6	运行和维护知识	278	6.4.1	WebApp 的特性	341
5.6.1	系统转换	278	6.4.2	WebApp 需求模型	341
5.6.2	系统维护概述	279	6.4.3	WebApp 设计	344
5.6.3	系统评价	283	6.5	用户界面设计	346
5.7	软件项目管理	284	6.5.1	用户界面设计的黄金原则	346
5.7.1	软件项目管理涉及的范围	284	6.5.2	用户界面的分析与设计	348
5.7.2	软件项目估算	287	6.5.3	用户界面设计问题	349
5.7.3	进度管理	289	第 7 章	面向对象技术	351
5.7.4	软件项目的组织	292	7.1	面向对象基础	351
5.7.5	软件配置管理	294	7.1.1	面向对象的基本概念	351
5.7.6	风险管理	296	7.1.2	面向对象分析	354
5.8	软件质量	300	7.1.3	面向对象设计	355
5.8.1	软件质量特性	300	7.1.4	面向对象程序设计	357
5.8.2	软件质量保证	302	7.1.5	面向对象测试	362
5.8.3	软件评审	304	7.2	UML	363
5.8.4	软件容错技术	306	7.2.1	事物	364
5.9	软件度量	307	7.2.2	关系	365
5.9.1	软件度量分类	307	7.2.3	UML 中的图	366
5.9.2	软件复杂性度量	309	7.3	设计模式	378
5.10	软件工具与软件开发环境	314	7.3.1	设计模式的要素	378
5.10.1	软件工具	311	7.3.2	创建型设计模式	379
5.10.2	软件开发环境	313	7.3.3	结构型设计模式	384
第 6 章	结构化开发方法	315	7.3.4	行为设计模式	394
6.1	系统分析与设计概述	315	7.3.5	应用举例	407
6.1.1	系统分析概述	315	第 8 章	算法设计与分析	416
6.1.2	系统设计的基本原理	317	8.1	算法设计与分析的基本概念	416
6.1.3	系统总体结构设计	319	8.1.1	算法	416
6.1.4	系统文档	323	8.1.2	算法设计	416
6.2	结构化分析方法	325	8.1.3	算法分析	417
6.2.1	结构化分析方法概述	325	8.1.4	算法的表示	417
6.2.2	数据流图	325	8.2	算法分析基础	417
6.2.3	数据字典 (DD)	335	8.2.1	时间复杂度	417
6.3	结构化设计方法	337			

8.2.2	渐进符号	418	9.3.2	5种基本的关系代数运算	478
8.2.3	递归式	419	9.3.3	扩展的关系代数运算	481
8.3	分治法	422	9.4	关系数据库 SQL 语言简介	489
8.3.1	递归的概念	422	9.4.1	SQL 数据库体系结构	490
8.3.2	分治法的基本思想	423	9.4.2	SQL 的基本组成	490
8.3.3	分治法的典型实例	423	9.4.3	SQL 数据定义	491
8.4	动态规划法	427	9.4.4	SQL 数据查询	496
8.4.1	动态规划法的基本思想	427	9.4.5	SQL 数据更新	504
8.4.2	动态规划法的典型实例	428	9.4.6	SQL 访问控制	505
8.5	贪心法	433	9.4.7	嵌入式 SQL	507
8.5.1	贪心法的基本思想	433	9.5	关系数据库的规范化	508
8.5.2	贪心法的典型实例	434	9.5.1	函数依赖	508
8.6	回溯法	437	9.5.2	规范化	509
8.6.1	回溯法的算法框架	437	9.5.3	模式分解及分解应具有的特性	511
8.6.2	回溯法的典型实例	440	9.6	数据库的控制功能	512
8.7	分支限界法	445	9.6.1	事务管理	512
8.8	概率算法	446	9.6.2	数据库的备份与恢复	513
8.9	近似算法	448	9.6.3	并发控制	514
8.10	数据挖掘算法	448	第10章	网络与信息安全基础知识	517
8.11	智能优化算法	450	10.1	网络概述	517
第9章	数据库技术基础	455	10.1.1	计算机网络的	517
9.1	基本概念	455	10.1.2	计算机网络的	520
9.1.1	数据库与数据库系统	455	10.1.3	网络的拓扑	521
9.1.2	数据库管理系统的功能	456	10.1.4	ISO/OSI 网	523
9.1.3	数据库管理系统的特征及分类	457	10.2	网络互连硬	526
9.1.4	数据库系统的体系结构	458	10.2.1	网络的设备	526
9.1.5	数据库的三级模式结构	461	10.2.2	网络的传输	529
9.1.6	大数据	463	10.2.3	组建网络	531
9.2	数据模型	466	10.3	网络的协议	534
9.2.1	基本概念	466	10.3.1	网络的标准	534
9.2.2	数据模型的三要素	466	10.3.2	局域网协议	536
9.2.3	E-R 模型	466	10.3.3	广域网协议	541
9.2.4	数据模型	472	10.3.4	TCP/IP 协	544
9.2.5	关系模型	473	10.4	Internet 及	549
9.3	关系代数	474	10.4.1	Internet 概	550
9.3.1	关系数据库的基本概念	474	10.4.2	Internet 地	550

10.4.3	Internet 服务	558	12.2	数据库分析与设计	618
10.5	信息安全基础知识	564	12.2.1	数据库设计的策略与步骤	618
10.6	网络安全概述	568	12.2.2	需求分析	619
第 11 章	标准化和软件知识产权基础知识	573	12.2.3	概念结构设计	621
11.1	标准化基础知识	573	12.2.4	逻辑结构设计	623
11.1.1	基本概念	573	12.2.5	数据库的物理设计	625
11.1.2	信息技术标准化	579	12.2.6	数据库的实施与维护	628
11.1.3	标准化组织	581	12.2.7	案例分析	631
11.1.4	ISO 9000 标准简介	584	12.3	面向对象分析与设计	635
11.1.5	ISO/IEC 15504 过程评估 标准简介	587	12.3.1	面向对象分析与设计的步骤	636
11.2	知识产权基础知识	588	12.3.2	需求说明	637
11.2.1	基本概念	589	12.3.3	建模用例	637
11.2.2	计算机软件著作权	592	12.3.4	建模活动	638
11.2.3	计算机软件的商业秘密权	603	12.3.5	设计类图	640
11.2.4	专利权概述	605	12.3.6	建模对象状态	642
11.2.5	企业知识产权的保护	610	12.3.7	建模交互	643
第 12 章	软件系统分析与设计	612	12.4	算法分析与设计	645
12.1	结构化分析与设计	612	12.4.1	C 程序设计语言与实现	646
12.1.1	需求说明	614	12.4.2	算法设计与实现	659
12.1.2	结构化分析	614	12.5	面向对象的程序设计 with 实现	672
12.1.3	总体设计	616	12.5.1	设计与实现方法	672
12.1.4	详细设计	617	12.5.2	设计模式的应用	672

第 1 章 计算机网络概论

1.1 计算机系统基础知识

1.1.1 计算机系统硬件基本组成

计算机系统是由硬件和软件组成的，它们协同工作来运行程序。计算机的基本硬件系统由运算器、控制器、存储器、输入设备和输出设备 5 大部件组成。运算器、控制器等部件被集成在一起统称为中央处理单元（Central Processing Unit, CPU）。CPU 是硬件系统的核心，用于数据的加工处理，能完成各种算术、逻辑运算及控制功能。存储器是计算机系统记忆设备，分为内部存储器和外部存储器。前者速度快、容量小，一般用于临时存放程序、数据及中间结果。而后者容量大、速度慢，可以长期保存程序和数据。输入设备和输出设备合称为外部设备（简称外设），输入设备用于输入原始数据及各种命令，而输出设备则用于输出计算机运行的结果。

1.1.2 中央处理单元

中央处理单元（CPU）是计算机系统的核心部件，它负责获取程序指令、对指令进行译码并加以执行。

1. CPU 的功能

- （1）程序控制。CPU 通过执行指令来控制程序的执行顺序，这是 CPU 的重要功能。
 - （2）操作控制。一条指令功能的实现需要若干操作信号配合来完成，CPU 产生每条指令的操作信号并将操作信号送往对应的部件，控制相应的部件按指令的功能要求进行操作。
 - （3）时间控制。CPU 对各种操作进行时间上的控制，即指令执行过程中操作信号的出现时间、持续时间及出现的时间顺序都需要进行严格控制。
 - （4）数据处理。CPU 通过对数据进行算术运算及逻辑运算等方式进行加工处理，数据加工处理的结果被人们所利用。所以，对数据的加工处理也是 CPU 最根本的任务。
- 此外，CPU 还需要对系统内部和外部的中断（异常）做出响应，进行相应的处理。

2. CPU 的组成

CPU 主要由运算器、控制器、寄存器组和内部总线等部件组成，如图 1-1 所示。

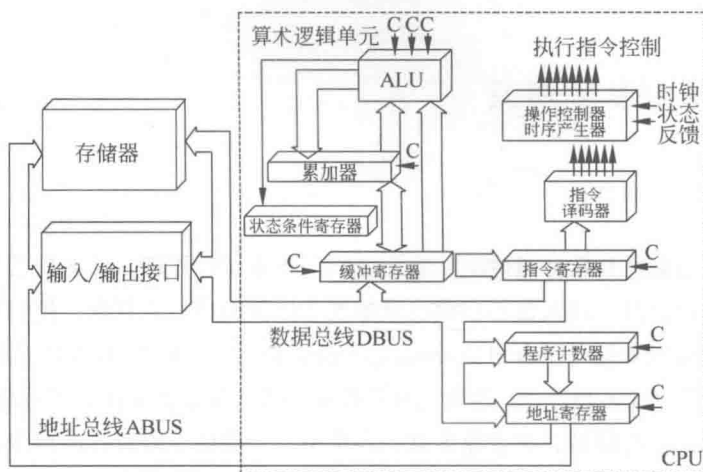


图 1-1 CPU 基本组成结构示意图

1) 运算器

运算器由算术逻辑单元（Arithmetic and Logic Unit, ALU）、累加寄存器、数据缓冲寄存器和状态条件寄存器等组成，它是数据加工处理部件，用于完成计算机的各种算术和逻辑运算。相对控制器而言，运算器接受控制器的命令而进行动作，即运算器所进行的全部操作都是由控制器发出的控制信号来指挥的，所以它是执行部件。运算器有如下两个主要功能。

- (1) 执行所有的算术运算，例如加、减、乘、除等基本运算及附加运算。
 - (2) 执行所有的逻辑运算并进行逻辑测试，例如与、或、非、零值测试或两个值的比较等。
- 下面简要介绍运算器中各组成部件的功能。

(1) 算术逻辑单元（ALU）。ALU 是运算器的重要组成部分，负责处理数据，实现对数据的算术运算和逻辑运算。

(2) 累加寄存器（AC）。AC 通常简称为累加器，它是一个通用寄存器，其功能是当运算器的算术逻辑单元执行算术或逻辑运算时，为 ALU 提供一工作区。例如，在执行一个减法运算前，先将被减数取出暂存在 AC 中，再从内存储器中取出减数，然后同 AC 的内容相减，将所得的结果送回 AC 中。运算的结果是放在累加器中的，运算器中至少要有一个累加寄存器。

(3) 数据缓冲寄存器 (DR)。在对内存储器进行读/写操作时, 用 DR 暂时存放由内存储器读/写的一条指令或一个数据字, 将不同时间段内读/写的数数据隔离开来。DR 的主要作用为: 作为 CPU 和内存、外部设备之间数据传送的中转站; 作为 CPU 和内存、外围设备之间在操作速度上的缓冲; 在单累加器结构的运算器中, 数据缓冲寄存器还可兼作为操作数寄存器。

(4) 状态条件寄存器 (PSW)。PSW 保存由算术指令和逻辑指令运行或测试的结果建立的各种条件码内容, 主要分为状态标志和控制标志, 例如运算结果进位标志 (C)、运算结果溢出标志 (V)、运算结果为 0 标志 (Z)、运算结果为负标志 (N)、中断标志 (I)、方向标志 (D) 和单步标志等。这些标志通常分别由 1 位触发器保存, 保存了当前指令执行完成之后的状态。通常, 一个算术操作产生一个运算结果, 而一个逻辑操作产生一个判决。

2) 控制器

运算器只能完成运算, 而控制器用于控制整个 CPU 的工作, 它决定了计算机运行过程的自动化。它不仅要保证程序的正确执行, 而且要能够处理异常事件。控制器一般包括指令控制逻辑、时序控制逻辑、总线控制逻辑和中断控制逻辑等几个部分。

指令控制逻辑要完成取指令、分析指令和执行指令的操作, 其过程分为取指令、指令译码、按指令操作码执行、形成下一条指令地址等步骤。

(1) 指令寄存器 (IR)。当 CPU 执行一条指令时, 先把它从内存储器取到缓冲寄存器中, 再送入 IR 暂存, 指令译码器根据 IR 的内容产生各种微操作指令, 控制其他的组成部件工作, 完成所需的功能。

(2) 程序计数器 (PC)。PC 具有寄存信息和计数两种功能, 又称为指令计数器。程序的执行分两种情况, 一是顺序执行, 二是转移执行。在程序开始执行前, 将程序的起始地址送入 PC, 该地址在程序加载到内存时确定, 因此 PC 的内容即是程序第一条指令的地址。执行指令时, CPU 自动修改 PC 的内容, 以便使其保持的总是将要执行的下一条指令的地址。由于大多数指令都是按顺序来执行的, 所以修改的过程通常只是简单地对 PC 加 1。当遇到转移指令时, 后继指令的地址根据当前指令的地址加上一个向前或向后转移的位移量得到, 或者根据转移指令给出的直接转移的地址得到。

(3) 地址寄存器 (AR)。AR 保存当前 CPU 所访问的内存单元的地址。由于内存和 CPU 存在着操作速度上的差异, 所以需要 AR 保持地址信息, 直到内存的读/写操作完成为止。

(4) 指令译码器 (ID)。指令包含操作码和地址码两部分, 为了能执行任何给定的指令, 必须对操作码进行分析, 以便识别所完成的操作。指令译码器就是对指令中的操作码字段进行分析解释, 识别该指令规定的操作, 向操作控制器发出具体的控制信号, 控制各部件工作, 完成所需的功能。

时序控制逻辑要为每条指令按时间顺序提供应有的控制信号。总线逻辑是为多个功能部件

服务的信息通路的控制电路。中断控制逻辑用于控制各种中断请求，并根据优先级的高低对中断请求进行排队，逐个交给 CPU 处理。

3) 寄存器组

寄存器组可分为专用寄存器和通用寄存器。运算器和控制器中的寄存器是专用寄存器，其作用是固定的。通用寄存器用途广泛并可由程序员规定其用途，其数目因处理器不同有所差异。

3. 多核 CPU

核心又称为内核，是 CPU 最重要的组成部分。CPU 中心那块隆起的芯片就是核心，是由单晶硅以一定的生产工艺制造出来的，CPU 所有的计算、接收/存储命令、处理数据都由核心执行。各种 CPU 核心都具有固定的逻辑结构，一级缓存、二级缓存、执行单元、指令级单元和总线接口等逻辑单元都会有合理的布局。

多核即在一个单芯片上面集成两个甚至更多个处理器内核，其中，每个内核都有自己的逻辑单元、控制单元、中断处理器、运算单元，一级 Cache、二级 Cache 共享或独有，其部件的完整性和单核处理器内核相比完全一致。

CPU 的主要厂商 AMD 和 Intel 的双核技术在物理结构上有所不同。AMD 将两个内核做在一个 Die（晶元）上，通过直连架构连接起来，集成度更高。Intel 则是将放在不同核心上的两个内核封装在一起，因此将 Intel 的方案称为“双芯”，将 AMD 的方案称为“双核”。从用户的角度来看，AMD 的方案能够使双核 CPU 的管脚、功耗等指标跟单核 CPU 保持一致，从单核升级到双核，不需要更换电源、芯片组、散热系统和主板，只需要刷新 BIOS 软件即可。

多核 CPU 系统最大的优点（也是开发的最主要目的）是可满足用户同时进行多任务处理的要求。

单核多线程 CPU 是交替地转换执行多个任务，只不过交替转换的时间很短，用户一般感觉不出来。如果同时执行的任务太多，就会感觉到“慢”或者“卡”。而多核在理论上则是在任何时间内每个核执行各自的任务，不存在交替问题。因此，单核多线程和多核（一般每核也是多线程的）虽然都可以执行多任务，但多核的速度更快。

虽然采用了 Intel 超线程技术的单核可以视为是双核，4 核可以视为是 8 核。然而，视为是 8 核一般比不上实际是 8 核的 CPU 性能。

要发挥 CPU 的多核性能，就需要操作系统能够及时、合理地给各个核分配任务和资源（如缓存、总线、内存等），也需要应用软件在运行时可以把并行的线程同时交付给多个核心分别处理。

1.1.3 数据表示

各种数值在计算机中表示的形式称为机器数，其特点是采用二进制计数制，数的符号用 0

和1表示,小数点则隐含,表示不占位置。机器数对应的实际数值称为数的真值。

机器数有无符号数和带符号数之分。无符号数表示正数,在机器数中没有符号位。对于无符号数,若约定小数点的位置在机器数的最低位之后,则是纯整数;若约定小数点的位置在机器数的最高位之前,则是纯小数。对于带符号数,机器数的最高位是表示正、负的符号位,其余位则表示数值。

为了便于运算,带符号的机器数可采用原码、反码和补码等不同的编码方法,机器数的这些编码方法称为码制。

1) 原码、反码、补码和移码

(1) 原码表示法。数值 X 的原码记为 $[X]_{\text{原}}$, 如果机器字长为 n (即采用 n 个二进制位表示数据), 则原码的定义如下:

$$\text{若 } X \text{ 是纯整数, 则 } [X]_{\text{原}} = \begin{cases} X & 0 \leq X \leq 2^{n-1} - 1 \\ 2^{n-1} + |X| & -(2^{n-1} - 1) \leq X \leq 0 \end{cases}$$

$$\text{若 } X \text{ 是纯小数, 则 } [X]_{\text{原}} = \begin{cases} X & 0 \leq X < 1 \\ 2^0 + |X| & -1 < X \leq 0 \end{cases}$$

【例 1.1】 若机器字长 n 等于 8, 分别给出 +1, -1, +127, -127, +45, -45, +0.5, -0.5 的原码表示。

$$\begin{aligned} [+1]_{\text{原}} &= 0\ 0000001 & [-1]_{\text{原}} &= 1\ 0000001 \\ [+127]_{\text{原}} &= 0\ 1111111 & [-127]_{\text{原}} &= 1\ 1111111 \\ [+45]_{\text{原}} &= 0\ 0101101 & [-45]_{\text{原}} &= 1\ 0101101 \\ [+0.5]_{\text{原}} &= 0\ \diamond 1000000 & [-0.5]_{\text{原}} &= 1\ \diamond 1000000 \end{aligned}$$

(其中, \diamond 是小数点的位置)

在原码表示法中,最高位是符号位,0表示正号,1表示负号,其余的 $n-1$ 位表示数值的绝对值。数值 0 的原码表示有两种形式: $[+0]_{\text{原}} = 0\ 0000000$, $[-0]_{\text{原}} = 1\ 0000000$ 。

(2) 反码表示法。数值 X 的反码记作 $[X]_{\text{反}}$, 如果机器字长为 n , 则反码的定义如下:

$$\text{若 } X \text{ 是纯整数, 则 } [X]_{\text{反}} = \begin{cases} X & 0 \leq X \leq 2^{n-1} - 1 \\ 2^n - 1 + X & -(2^{n-1} - 1) \leq X \leq 0 \end{cases}$$

$$\text{若 } X \text{ 是纯小数, 则 } [X]_{\text{反}} = \begin{cases} X & 0 \leq X < 1 \\ 2 - 2^{-(n-1)} + X & -1 < X \leq 0 \end{cases}$$

【例 1.2】 若机器字长 n 等于 8, 分别给出 +1, -1, +127, -127, +45, -45, +0.5, -0.5 的反码表示。

$$\begin{aligned} [+1]_{\text{反}} &= 0\ 0000001 & [-1]_{\text{反}} &= 1\ 1111110 \\ [+127]_{\text{反}} &= 0\ 1111111 & [-127]_{\text{反}} &= 1\ 0000000 \\ [+45]_{\text{反}} &= 0\ 0101101 & [-45]_{\text{反}} &= 1\ 1010010 \end{aligned}$$

$$[+0.5]_{\text{反}}=0 \diamond 1000000 \quad [-0.5]_{\text{反}}=1 \diamond 0111111 \quad (\text{其中, } \diamond \text{ 是小数点的位置})$$

在反码表示中,最高位是符号位,0表示正号,1表示负号,正数的反码与原码相同,负数的反码则是其绝对值按位求反。数值0的反码表示有两种形式: $[+0]_{\text{反}}=0 \ 0000000$, $[-0]_{\text{反}}=1 \ 1111111$ 。

(3) 补码表示法。数值 X 的补码记作 $[X]_{\text{补}}$, 如果机器字长为 n , 则补码的定义如下:

$$\text{若 } X \text{ 是纯整数, 则 } [X]_{\text{补}} = \begin{cases} X & 0 \leq X \leq 2^{n-1} - 1 \\ 2^n + X & -2^{n-1} \leq X \leq 0 \end{cases}$$

$$\text{若 } X \text{ 是纯小数, 则 } [X]_{\text{补}} = \begin{cases} X & 0 \leq X < 1 \\ 2 + X & -1 \leq X < 0 \end{cases}$$

【例 1.3】 若机器字长 n 等于 8, 分别给出 +1, -1, +127, -127, +45, -45, +0.5, -0.5 的补码表示。

$$[+1]_{\text{补}}=0 \ 0000001 \quad [-1]_{\text{补}}=1 \ 1111111$$

$$[+127]_{\text{补}}=0 \ 1111111 \quad [-127]_{\text{补}}=1 \ 0000001$$

$$[+45]_{\text{补}}=0 \ 0101101 \quad [-45]_{\text{补}}=1 \ 1010011$$

$$[+0.5]_{\text{补}}=0 \ \diamond 1000000 \quad [-0.5]_{\text{补}}=1 \ \diamond 1000000 \quad (\text{其中, } \diamond \text{ 是小数点的位置})$$

在补码表示中,最高位为符号位,0表示正号,1表示负号,正数的补码与其原码和反码相同,负数的补码则等于其反码的末位加1。在补码表示中,0有唯一的编码: $[+0]_{\text{补}}=0 \ 0000000$, $[-0]_{\text{补}}=00000000$ 。

(4) 移码表示法。移码表示法是在数 X 上增加一个偏移量来定义的,常用于表示浮点数中的阶码。如果机器字长为 n , 规定偏移量为 2^{n-1} , 则移码的定义如下:

若 X 是纯整数, 则 $[X]_{\text{移}}=2^{n-1} + X$ ($-2^{n-1} \leq X < 2^{n-1}$); 若 X 是纯小数, 则 $[X]_{\text{移}}=1 + X$ ($-1 \leq X < 1$)。

【例 1.4】 若机器字长 n 等于 8, 分别给出 +1, -1, +127, -127, +45, -45, +0, -0 的移码表示。

$$[+1]_{\text{移}}=1 \ 0000001 \quad [-1]_{\text{移}}=0 \ 1111111$$

$$[+127]_{\text{移}}=1 \ 1111111 \quad [-127]_{\text{移}}=0 \ 0000001$$

$$[+45]_{\text{移}}=1 \ 0101101 \quad [-45]_{\text{移}}=0 \ 1010011$$

$$[+0]_{\text{移}}=1 \ 0000000 \quad [-0]_{\text{移}}=10000000$$

实际上,在偏移 2^{n-1} 的情况下,只要将补码的符号位取反便可获得相应的移码表示。

2) 定点数和浮点数

(1) 定点数。所谓定点数,就是小数点的位置固定不变的数。小数点的位置通常有两种约定方式: 定点整数(纯整数,小数点在最低有效数值位之后)和定点小数(纯小数,小数点在

最高有效数值位之前)。

设机器字长为 n ，各种码制下带符号数的范围如表 1-1 所示。

表 1-1 机器字长为 n 时各种码制表示的带符号数的范围

码 制	定 点 整 数	定 点 小 数
原码	$-(2^{n-1}-1) \sim +(2^{n-1}-1)$	$-(1-2^{-(n-1)}) \sim +(1-2^{-(n-1)})$
反码	$-(2^{n-1}-1) \sim +(2^{n-1}-1)$	$-(1-2^{-(n-1)}) \sim +(1-2^{-(n-1)})$
补码	$-2^{n-1} \sim +(2^{n-1}-1)$	$-1 \sim +(1-2^{-(n-1)})$
移码	$-2^{n-1} \sim +(2^{n-1}-1)$	$-1 \sim +(1-2^{-(n-1)})$

(2) 浮点数。当机器字长为 n 时，定点数的补码和移码可表示 2^n 个数，而其原码和反码只能表示 2^n-1 个数 (0 的表示占用了两个编码)，因此，定点数所能表示的数值范围比较小，在运算中很容易因结果超出范围而溢出。浮点数是小数点位置不固定的数，它能表示更大范围的数。

在十进制中，一个数可以写成多种表示形式。例如，83.125 可写成 $10^3 \times 0.083125$ 或 $10^4 \times 0.0083125$ 等。同样，一个二进制数也可以写成多种表示形式。例如，二进制数 1011.10101 可以写成 $2^4 \times 0.101110101$ 、 $2^5 \times 0.0101110101$ 或 $2^6 \times 0.00101110101$ 等。由此可知，一个二进制数 N 可以表示为更一般的形式 $N=2^E \times F$ ，其中 E 称为阶码， F 称为尾数。用阶码和尾数表示的数称为浮点数，这种表示数的方法称为浮点表示法。

在浮点表示法中，阶码为带符号的纯整数，尾数为带符号的纯小数。浮点数的表示格式如下：

阶符	阶码	数符	尾数
----	----	----	----

很明显，一个数的浮点表示不是唯一的。当小数点的位置改变时，阶码也随着相应改变，因此可以用多个浮点形式表示同一个数。

浮点数所能表示的数值范围主要由阶码决定，所表示数值的精度则由尾数决定。为了充分利用尾数来表示更多的有效数字，通常采用规格化浮点数。规格化就是将尾数的绝对值限定在区间 $[0.5, 1]$ 。当尾数用补码表示时，需要注意如下问题。

① 若尾数 $M \geq 0$ ，则其规格化的尾数形式为 $M=0.1 \times \times \times \dots \times$ ，其中， \times 可为 0，也可为 1，即将尾数限定在区间 $[0.5, 1]$ 。

② 若尾数 $M < 0$ ，则其规格化的尾数形式为 $M=1.0 \times \times \times \dots \times$ ，其中， \times 可为 0，也可为 1，即将尾数 M 的范围限定在区间 $[-1, -0.5]$ 。

如果浮点数的阶码 (包括 1 位阶符) 用 R 位的移码表示，尾数 (包括 1 位数符) 用 M 位的补码表示，则这种浮点数所能表示的数值范围如下。