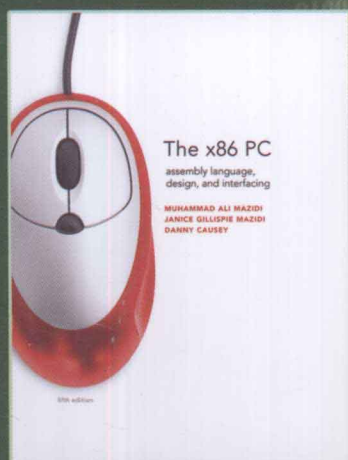


x86 PC 汇编语言、设计与接口 (第五版)

The x86 PC
Assembly Language, Design, and Interfacing
Fifth Edition



Muhammad Ali Mazidi
[美] Janice Gillispie Mazidi 著
Danny Causey

高 升 译
王筱珍 审



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

<http://www.phei.com.cn>

x86 PC 汇编语言、 设计与接口 (第五版)

The x86 PC: Assembly Language, Design,
and Interfacing, Fifth Edition

Muhammad Ali Mazidi
[美] Janice Gillispie Mazidi 著
Danny Causey

高 升 译
王筱珍 审

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

汇编语言与程序设计是计算机类学科专业重要的专业技术基础课程,是以应用为主的工程技术基础类课程。作者以简单易懂、分步介绍的方式,讲授了x86 PC 汇编语言程序设计及PC 体系结构。全书包括两大部分内容:(1)汇编语言程序设计;(2)IBM PC 兼容计算机接口设计。在介绍汇编语言程序的章节中,以汇编方法为引导,逐步加入各种语句及指令,给出了很多程序实例,并通过Debug 实用工具展示出程序指令执行的具体动作。在接口设计方面,从PC 所应用的芯片到设备,从电路设计到编程都进行了详细的阐述。

本书内容新颖,示例丰富,可作为高校计算机专业学生的教材,也是x86 嵌入式设计人员的理想参考书。

Original edition, entitled The x86 PC: Assembly Language, Design, and Interfacing, Fifth Edition, 978-0-13-502648-2 by Muhammad Ali Mazidi, Janice Gillispie Mazidi, Danny Causey, Published by Pearson Education, Inc., publishing as Prentice Hall, Copyright © 2010 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Chinese simplified language edition published by PEARSON EDUCATION ASIA LTD. and PUBLISHING HOUSE OF ELECTRONICS INDUSTRY, copyright © 2011.

本书中文简体字翻译版由 Pearson Education (培生教育出版集团) 授予电子工业出版社。未经出版者预先书面许可,不得以任何方式复制或抄袭本书的任何部分。

本书封面贴有 Pearson Education (培生教育出版集团) 激光防伪标签,无标签者不得销售。

版权贸易合同登记号 图字: 01-2009-6045

图书在版编目(CIP)数据

x86 PC 汇编语言、设计与接口: 第5版/(美)马兹迪(Mazidi, M. A.), (美)马兹迪(Mazidi, J. G.), (美)考西(Causey, C.)著;高升译. —北京:电子工业出版社, 2011.1

(国外计算机科学教材系列)

书名原文: The x86 PC: Assembly Language, Design, and Interfacing, 5e

ISBN 978-7-121-12441-9

I. ①x… II. ①马…②马…③考…④高… III. ①汇编语言—程序设计—教材 IV. ①TP313

中国版本图书馆 CIP 数据核字(2010)第 233947 号

策划编辑: 谭海平

责任编辑: 史 平

印 刷: 北京丰源印刷厂

装 订: 三河市鹏成印业有限公司

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: 36.25 字数: 928 千字

印 次: 2011 年 1 月第 1 次印刷

定 价: 75.00 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。

质量投诉请发邮件至 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010)88258888。

译 者 序

经过一年的辛勤工作, *The x86 PC Assembly Language, Design and Interfacing* 的中译版终于与读者见面了。这是一本全面介绍 x86 汇编语言程序设计、硬件设计与接口的专业图书。本书的初版于 1992 年面世, 随着硬件技术和 x86 处理器性能的不断进步, 本书历经了从第一版到第五版的变化。

本书译者在多年的计算机专业教学实践中, 一直坚持使用世界著名英文原版教材授课, 并取得较好的教学效果。尽管如此, 心里仍一直存在着翻译英文原版教材的心愿, 电子工业出版社翻译英文原版的计划与译者的心愿不谋而合, 并且最终使得这一心愿得以实现, 为此应诚挚地感谢电子工业出版社。之所以想翻译英文原版教材, 是因为我国大学计算机类教材知识老化, 与现代知识理论严重脱节, 而翻译世界先进水平的大学教科书, 站在巨人的肩膀上起步, 是我国大学教育与国际先进标准接轨的最快、最直接的方法之一。这不仅促进了我国大学的专业教育以最快的渠道与世界接轨, 也使得人才培养符合社会需求, 而且少走弯路。

本书译者长期从事计算机专业的教学与科研工作, 主讲汇编语言、操作系统、单片机、微机原理与接口等课程, 多年以来一直从事操作系统课程的双语教学, 具有多个项目的开发经历, 这些均为本书的翻译工作奠定了良好的基础。

在此要感谢王筱珍老师, 她对本书的译稿进行了校对工作, 对本书的翻译提出了许多颇具价值的改进意见。广东海洋大学吴志君等同学也参加本书的部分译稿校对及编辑排版工作, 在此一并表示感谢。正是由于各位出色的工作, 使得本书更加完美。

概括起来, 本书具有如下特点:

1. 内容安排循序渐进, 便于自学。
2. 语言生动, 深入浅出, 实例丰富, 且每个章节之后都附有大量的复习思考题和习题。
3. 内容丰富, 本书共 28 章, 内容涉及 x86 PC 的汇编语言、中断技术、ISA 总线、PCI 总线、USB 接口、存储器设计、保护模式、流水线、芯片引脚等各方面。
4. 能够紧随硬件和 x86 处理器性能的发展不断充实新内容, 使得内容与实际紧密结合。

本书适用于大中专院校计算机、电气工程、自动化等专业学生阅读, 既可以作为教材使用, 也可以作为技术手册为工程实践提供参考资料。

由于译者水平有限, 加上时间仓促, 错误和不足之处敬请读者批评指正。

译者
于广东海洋大学

前 言

编写目的

本书用于汇编语言程序设计和 x86 PC 接口的大学课程。它不仅为汇编语言程序设计的学习打下基础,同时也为从事工程与计算机学科的学生提供关于 x86 PC 设计与接口方面的综合训练。这本书旨在帮助读者对 x86 PC 的内部原理进行深入理解。它为现实世界中基于微处理器的系统中的设计与接口的理解奠定基础。此外,本书也适用于从事微机接口和数据采集的技术人员、硬件工程师、计算机研究人员以及业余爱好者。

阅读本书所需要的背景知识

阅读本书的读者应该具备初级的数字知识。虽然编程语言的知识也很有帮助,但不是必需的。尽管当前绝大多数 PC 使用奔腾等 x86 作为它们的微处理器,但是它们仍然是根据 IBM PC/AT 这个在 1984 年推出的 80286 微处理器系统而设计的。PC/AT 既有可取之处,也有一些缺点,早在 1981 年推出基于 8088 微处理器的 IBM PC 的时候,这些优点和缺点就已经确定了。换句话说,只有先理解了 80286 PC/AT 及其子集 PC/XT 之后,才能充分理解 x86 PC 的组成和内部构成。因此,本书将在第 9 章讲述 8088 和 80286 微处理器。

内容介绍

汇编语言程序设计的学习往往采用系统的、渐进的方法。本书提供许多例子以及示例程序,以便澄清相关概念,并促使学生边学边做。每节的后面都设有测验题,以巩固该节主要知识点。我们认为,一本教科书的作用之一,是让学生熟悉文献及行业中的术语,本书的编写也遵循了同样的规则。

第 0 章叙述数字系统(二进制、十进制和十六进制)中相关概念及计算机结构。大多数学生可能在以前的课程中学过这些概念,但本章为那些没有学过这些概念的学生提供一个快速的教程。对那些已经学习了这些概念的学生而言,本章引领他们进行复习。

第 1 章简要介绍 x86 微处理器的发展史,讲解作为所有 x86 处理器基础的 8086 的内部工作情况。

第 2 章主要介绍如何使用汇编语言创建程序。尽管书中的程序是基于 Microsoft MASM 汇编器建立并测试的,但它们同样适用于 Borland TASM 等其他兼容的汇编器。

第 3 章介绍无符号逻辑运算和算术运算指令,以及如何用 C 语言进行位操作。

第 4 章介绍 DOS 和 BIOS 中断。汇编程序允许学生从键盘输入数据并发送到显示器上。此外,本章还将涉及宏汇编。

第 5 章介绍如何进行键盘和鼠标编程。

第 6 章介绍带符号数的算术运算和逻辑运算以及串操作。

第 7 章讨论模块化编程方法,以及如何将可以独立编程和调试的小程序模块合并成大程序。此外,还将介绍如何在 C 语言中嵌入汇编语句。

第 8 章介绍 x86 32 位程序设计的一些概念。尽管此书重点介绍的是 16 位编程,但是 386 的引入可帮助同学们体验 32 位处理器的强大功能。

第 9 章详细介绍 8088 和 286 微处理器及支撑芯片和它们在早期的 IBM PC/XT/AT 中的使用。同时介绍 ISA 扩展槽的地址、数据和控制信号的产生和功能。

第 10 章介绍各类 RAM 和 ROM 存储器、它们与微处理器的接口、x86 PC 的存储映射、存储器接口的时序, 以及为确保 RAM 和 ROM 中数据完整性而使用的校验字节和奇偶校验位技术。

第 11 章主要介绍 I/O 接口、x86 中 IN 指令和 OUT 指令的使用, 以及 8255 芯片的接口与编程。书中将用若干种语言对 I/O 编程进行描述。

第 12 章讲述 PC 与 LCD、步进电机、ADC、DAC 和传感器等数据采集设备的连接。

第 13 章讨论 x86 PC 中 8253/54 定时器的使用, 音乐演奏和生成时序的方法。

第 14 章介绍硬件中断和软件中断、8259 中断控制器的使用、ISA 总线扩展槽上 IRQ 符号的产生和分配。

第 15 章介绍直接存储器访问(DMA)的概念及其在 x86 PC 中的应用、DMA 通道以及相关的 ISA 总线信号。

第 16 章介绍视频显示器的基本知识、PC 的各种显示模式以及图形模式下显卡的存储器需求。

第 17 章讨论串行通信的原理和 PC 上 COM 端口的编程。

第 18 章介绍 PC 键盘、打印机的接口和编程, 讨论 EPP 和 ECP 等并行接口。

第 19 章讨论硬磁盘储器的组织和相关术语。

第 20 章介绍 x87 数字协处理器及编程、IEEE 单精度和双精确浮点数据类型。

第 21 章探讨 386 处理器的硬件及编程, 比较实模式和保护模式, 讨论虚拟存储器的实现。

第 22 章讲述高速存储器的连接, 描述各种类型的 DRAM, 包括 EDO 和 SDRAM, 并且详细解释高速缓存、高速缓存的组成及术语。

第 23 章介绍 486、Pentium 和 Pentium Pro 的主要特征, 并将它们同 RISC 处理器进行比较。同时, 这一章还讨论了 MMX 技术, 以及如何编写程序来检测 CPU 型号。

第 24 章讲解 Intel 和 AMD 的新一代 64 位处理器。

第 25 章概述集成电路工艺及故障分析, 描述 IC 接口和系统设计问题, 同时讲解错误检测和校正。

第 26 章讨论 ISA 和 PC104 等各类 PC 总线、性能比较以及 PCI 总线的特征。

第 27 章详细介绍 USB 接口, 说明如何使用 C 语言编程来访问连接到 x86 PC 上的 USB 设备。

实验指导书

实验指导书中提供了一些非常基本的实验操作, 这些内容可以在 www.MicroDigitalEd.com 站点上找到。教师可根据教学目标、学生水平以及听课对象是研究生还是本科生等因素, 设置一些更为高级的、有一定难度的实验。相关支撑材料和作者编写的其他书籍也可以在这个站点上找到。

习题解答/PowerPoint 演示文稿

每章后面的习题涵盖了一些基本的概念。教师可根据具体情况设置一些难度较大的课后习题。习题解答是在 Sepehr Naimi 先生的帮助下完成的。习题解答和 PowerPoint 演示文稿只供教师从网上下载^①。

在线教学资源

要想访问网上资源, 教师需要申请一个密码。方法如下: 进入 www.prenhall.com 网站、单击 “Instructor Resource Center” (教学资源中心), 接着单击 “Register Today” (注册) 来获取密码。48 小时后你将收到带有密码的确认邮件。获得密码后可立即登录, 按照提示下载你想要的材料。

^① 国内教师申请教辅材料的方法, 请参见本书末尾的 “教学支持说明” 页。

致谢

此书是许多人士奉献、工作和爱的结晶。在此必须感谢诸多学生和教授，正是由于自 1992 年本书首次出版以来他们的精辟评论才造就了这本书。特别感谢 Dimitri Moonen，他详细地阅读了此书并指出了其中的一些错误。感谢 Pedran Mazidi 重新制作了本书新版中的图表。还要感谢 Prentice Hall 的工作人员，尤其是 Wyatt Morris 编辑，他一直对本书的写作提供支持与鼓励。感谢项目经理 Rex Davidson，使此书终成现实。我们很荣幸能请到世上最优秀的编辑 Bret Workman，感谢你精妙的工作。最后，真诚地感谢德克萨斯大学计算机科学与工程学院的 Roger S. Walker 博士，感谢他的不断鼓励。我们为编写此书而感到无比高兴，希望你喜欢这本书，并在课程学习和项目开发中使用。如果您想对本书提出改进建议，或发现本书中存在错误，请及时与我们联系，以便我们不断改进和完善。

作者简介

Muhammad Ali Mazidi 曾就读于美国大布里士大学，并获南卫理公会大学和德州大学达拉斯分校的硕士学位。现今他在南卫理公会大学的电气工程学院攻读博士学位。他曾经编写过《8051 微控制器与嵌入式系统》、《PIC 微控制器和嵌入式系统》、《HCS12 微控制器和嵌入式系统》等多部教材，这些教材被广泛使用，并可从 Prentice Hall 出版公司购买。他在得克萨斯州达拉斯的德锐大学讲授微处理器设计课程。他是 MicroDigitalEd.com 网站的创始人，该网站为本书提供支持。

Janice Gillispie Mazidi 获得北德克萨斯大学的计算机科学硕士学位。她具有在达拉斯从事软件工程工作的若干年经验。现今，她在当地的一所高中讲授计算机科学和数学课程。

Danny Causey 毕业于德锐大学，并获计算机工程技术理学学士学位。他是《PIC 微控制器和嵌入式系统》和《HCS12 微控制器和嵌入式系统》的作者。他是 MicroDigitalEd.com 网站的合伙人之一。他致力于软件开发，研制旨在提高日常工作效率的软件工具。

如果您有评论和建议，或是发现了书中的错误，可通过以下方式与作者取得联系：

mdebooks@yahoo.com

mmazidi@microdigitaled.com

dcausey@microdigitaled.com

目 录

第 0 章 计算导论	1
0.1 计数与编码系统	1
0.2 数字系统入门	6
0.3 进入计算机内部	9
本章习题	14
复习题答案	15
第 1 章 x86 微处理器	17
1.1 x86 系列演变的简要历史	17
1.2 走进 8088/86 的内部	20
1.3 汇编语言简介	22
1.4 程序段介绍	25
1.5 堆栈	31
1.6 标志寄存器	34
1.7 x86 寻址模式	37
本章习题	40
复习题答案	43
第 2 章 汇编语言程序设计	45
2.1 伪指令和示例程序	45
2.2 程序的汇编、链接与执行	48
2.3 示例程序	51
2.4 控制转移指令	56
2.5 数据类型和数据定义	60
2.6 完整段定义	64
2.7 流程图和伪代码	69
本章习题	71
复习题答案	73
第 3 章 算术与逻辑运算指令及编程	76
3.1 无符号数的加减法	76
3.2 无符号数的乘法和除法	82
3.3 逻辑指令	85
3.4 BCD 和 ASCII 的转换	94
3.5 循环移位指令	101
3.6 C 语言的位操作	104

本章习题	106
复习题答案	109
第 4 章 INT 21H 和 INT 10H 编程及宏指令	111
4.1 BIOS 中 INT 10H 的编程	111
4.2 DOS 中断 21H	117
4.3 什么是宏以及如何使用宏	127
本章习题	137
复习题答案	139
第 5 章 键盘和鼠标编程	141
5.1 INT 16H 键盘编程	141
5.2 利用 INT 33H 进行鼠标编程	145
本章习题	152
复习题答案	152
第 6 章 带符号数、字符串和表	153
6.1 带符号数的算术运算	153
6.2 字符串和表操作	163
本章习题	168
复习题答案	169
第 7 章 模块及模块化程序设计	170
7.1 编写和链接模块	170
7.2 一些非常有用的模块	179
7.3 在模块之间传递参数	184
本章习题	187
复习题答案	188
第 8 章 x86 的 32 位程序设计	189
8.1 x86 的 32 位编程	189
本章习题	196
复习题答案	197
第 9 章 8088、80286 微处理器与 ISA 总线	198
9.1 8088 微处理器	198
9.2 8284 和 8288 支撑芯片	203
9.3 ISA 总线的 8 位区段	206
9.4 80286 微处理器	210
9.5 16 位 ISA 总线	213
本章习题	217
复习题答案	219
第 10 章 存储器及其接口	220
10.1 半导体存储器	220

10.2	存储器地址译码	228
10.3	IBM PC 的内存映射	231
10.4	RAM 和 ROM 的数据完整性	234
10.5	16 位存储器的接口	238
	本章习题	242
	复习题答案	244
第 11 章	8255 I/O 编程	246
11.1	8088 输入/输出指令	246
11.2	I/O 地址的译码和设计	248
11.3	x86 PC 的 I/O 地址映射	251
11.4	8255 编程与接口	254
	本章习题	264
	复习题答案	265
第 12 章	LCD、电动机、ADC 及传感器接口	267
12.1	LCD 接口	267
12.2	步进电动机接口	275
12.3	DAC 接口	279
12.4	ADC 芯片和传感器接口	283
	本章习题	290
	复习题答案	292
第 13 章	8253/54 定时器	293
13.1	8253/54 定时器简介	293
13.2	x86 PC 中 8253/54 定时器的连接与编程	297
13.3	在 x86 PC 上播放音乐	301
	本章习题	306
	复习题答案	307
第 14 章	x86 PC 的中断系统	308
14.1	8088/86 中断	308
14.2	x86 PC 及其中断分配	313
14.3	8259 可编程中断控制器	316
14.4	8259 芯片在 x86 PC 中的应用	323
14.5	x86 PC 中断的进一步讨论	328
	本章习题	331
	复习题答案	333
第 15 章	x86 PC 直接存储器存取和 DMA 通道	335
15.1	DMA 的概念	335
15.2	8237 DMA 芯片编程	336
15.3	IBM PC 8237 DMA 接口	344
15.4	x86 PC DMA	347

本章习题	350
复习题答案	351
第 16 章 视频和视频适配器	353
16.1 显示器的原理和视频模式	353
16.2 文本模式编程及显存	359
16.3 图形和图形编程	366
本章习题	369
复习题答案	370
第 17 章 汇编语言和 C# 串口编程	371
17.1 串行通信的基本知识	371
17.2 x86 PC COM 端口的汇编和 C# 编程	376
本章习题	381
复习题答案	381
第 18 章 键盘与打印机接口	383
18.1 键盘与主机的接口	383
18.2 PC 键盘接口与编程	387
18.3 IBM PC 打印机及接口	394
本章习题	402
复习题答案	403
第 19 章 硬盘	405
19.1 硬盘组织与性能	405
本章习题	412
复习题答案	412
第 20 章 IEEE 浮点数和 x87 数字协处理器	413
20.1 数字协处理器和 IEEE 浮点标准	413
20.2 x87 指令和编程	416
20.3 x87 指令	428
本章习题	433
复习题答案	434
第 21 章 386 微处理器: 实模式与保护模式	435
21.1 实模式下的 80386	435
21.2 80386 硬件结构视图	442
21.3 80386 保护模式	447
本章习题	454
复习题答案	456
第 22 章 高速存储器设计和缓存	458
22.1 x86 的存储器周期时间	458
22.2 页式 DRAM 和静态列 DRAM	460

22.3 高速缓存	466
22.4 SDRAM、DDR RAM 和 RAMBUS 存储器	472
本章习题	477
复习题答案	480
第 23 章 奔腾和 RISC 处理器	482
23.1 80486 微处理器	482
23.2 Intel 奔腾处理器	486
23.3 RISC 架构	491
23.4 Pentium Pro(高能奔腾)处理器	497
23.5 MMX 技术	500
本章习题	505
复习题答案	508
第 24 章 x86 的演化: 从 32 位到 64 位	510
24.1 x86 奔腾系列处理器的演化	510
24.2 x86 的 64 位处理器和 Vista	514
本章习题	518
复习题答案	519
第 25 章 系统设计与故障分析	520
25.1 集成电路技术概述	520
25.2 IC 连接和系统设计问题	525
本章习题	534
复习题答案	535
第 26 章 ISA、PC104 和 PCI 总线	537
26.1 ISA 总线的内存访问信号	537
26.2 ISA 的 I/O 总线时序	543
26.3 PCI 总线	551
本章习题	557
复习题答案	558
第 27 章 USB 编程	559
27.1 USB 端口概要	559
27.2 USB 口的扩展与电源管理	560
27.3 USB 编程	563
本章习题	565
复习题答案	566

第0章 计算导论

目 标

学习完本章之后，你能够：

- 将一个数在二进制、十进制和十六进制之间做转换
- 做十六进制加减法运算
- 做二进制加法运算
- 用 2 的补码表示二进制数
- 用 ASCII 码表示字符串
- 描述 AND、OR、NOT、XOR、NAND 和 NOR 的逻辑操作
- 使用逻辑门绘制简单的线路图
- 解释位、半字节、字节以及字之间的区别
- 给出千字节、兆字节、吉字节和太字节等术语准确的数学定义
- 解释 RAM 与 ROM 的区别，描述它们的用途
- 描述计算机系统中主要构件的用途
- 能够列举计算机中的三种总线，描述每种类型总线的用途
- 描述计算机系统中 CPU 的作用
- 列举构成 CPU 的主要部件，描述每个部件的功能

为了理解基于微控制器的计算机系统的软件和硬件，首先必须掌握最基本的概念。本章(按照数字计算机的传统，称之为第 0 章)展示计数和编码系统。在介绍逻辑门之后，给出计算机内部工作原理的概述。作为结尾，在最后一节给出 CPU 体系结构进展的简要历史。可能许多读者对本章的一些内容已经很熟悉，尽管如此，仍然建议大家进行简要浏览。

0.1 计数与编码系统

人类使用十进制进行算术运算，而计算机使用二进制系统。本节解释十进制和二进制之间的转换、二进制数的十六进制表示，以及字母和数字的二进制编码形式——ASCII 码。

0.1.1 十进制和二进制计数系统

人类使用十进制系统的推测依据源自于人类有十个手指这一事实。然而，却找不到推测依据来说明为什么计算机中使用二进制系统。实际上，计算机中使用二进制系统是因为 1 和 0 表示电平的开与关。在基数为 10 的系统中有 10 个独立的符号：0, 1, 2, ..., 9，在基数为 2 的系统中只有 0 和 1 两个符号，这些符号用于组成数字。十进制数中包含 0 到 9 十个符号，二进制数中包含 0 和 1 两个符号，而这两个二进制符号就是通常所说的“位”。

0.1.2 十进制转换为二进制

十进制转换为二进制的方法之一是将十进制数重复地除以 2，记下所得的各余数。这一过程持续到商变成 0 为止。所得余数序列的反序即为对应的二进制值。请参见例 0.1。

例 0.1

将 25_{10} 转换为二进制。

解:

	商	余数	
$25/2$	$= 12$	1	LSB(最低有效位, least significant bit)
$12/2$	$= 6$	0	
$6/2$	$= 3$	0	
$3/2$	$= 1$	1	
$1/2$	$= 0$	1	MSB(最高有效位, most significant bit)
所以, $25_{10} = 11001_2$			

0.1.3 二进制转换为十进制

二进制转换为十进制时, 重要的是要理解每一个数字所对应位置的权重的概念。首先, 作为类比, 回顾一下基数为 10 系统中的数码的权重, 如下图所示。同样, 基数为 2 的系统中数据的每一个码位都对应一个权重。

$740683_{10} =$	
$3 \times 10^0 =$	3
$8 \times 10^1 =$	80
$6 \times 10^2 =$	600
$0 \times 10^3 =$	0000
$4 \times 10^4 =$	40000
$7 \times 10^5 =$	700000
	740683

$110101_2 =$	十进制	二进制
$1 \times 2^0 =$	$1 \times 1 = 1$	1
$0 \times 2^1 =$	$0 \times 2 = 0$	00
$1 \times 2^2 =$	$1 \times 4 = 4$	100
$0 \times 2^3 =$	$0 \times 8 = 0$	0000
$1 \times 2^4 =$	$1 \times 16 = 16$	10000
$1 \times 2^5 =$	$1 \times 32 = 32$	100000
	53	110101

已知二进制数中每一个位的权重之后, 就可以将它们相加以求得十进制的值, 如例 0.2 所示。

例 0.2

将 11001_2 转换为十进制。

解:

权重:	16	8	4	2	1
数码:	1	1	0	0	1
和:	$16+$	$8+$	$0+$	$0+$	$1 = 25_{10}$

知道了每一个二进制位的权重之后, 就可以直接将十进制转换为二进制, 而不需要进行反复的除法运算。如例 0.3 所示。

例 0.3

使用权重的概念将 39_{10} 转换为二进制。

解:

权重:	32	16	8	4	2	1
数码:	1	0	0	1	1	1
和:	$32+$	$0+$	$0+$	$4+$	$2+$	$1 = 39$

所以, $39_{10} = 100111_2$

0.1.4 十六进制系统

基数 16, 亦即计算机科学中所称的十六进制系统, 可以便捷地表示二进制数。例如, 对于人类来说, 如果将 0 和 1 构成的串 100010010110 表示为等价的十六进制形式 896H, 则更容易表达。二进制系统有两个符号(0 和 1); 十进制系统有十个符号, 0 至 9; 而十六进制(基数 16)系统中有十六个

符号。其中的前十个符号0~9同十进制系统一样，余下的六个符号使用字母A、B、C、D、E和F表示。表0.1显示出数值0~15等价的二进制、十进制和十六进制表示。

表0.1 基数为16的计数系统

十进制	二进制	十六进制	十进制	二进制	十六进制	十进制	二进制	十六进制
0	0000	0	6	0110	6	12	1100	C
1	0001	1	7	0111	7	13	1101	D
2	0010	2	8	1000	8	14	1110	E
3	0011	3	9	1001	9	15	1111	F
4	0100	4	10	1010	A			
5	0101	5	11	1011	B			

0.1.5 十六进制和二进制间的转换

要用等价的十六进制数表示二进制数，从右边开始每次以4位一组。如表0.1所列出的，用等价的十六进制代替这4位二进制数。要把十六进制转换为二进制，用等价的4位二进制数代替每位十六进制数。如例0.4和例0.5所示。

例0.4

用十六进制数表示二进制数100111110101

解：

首先把该二进制数分为每4位一组：1001 1111 0101

然后用十六进制数表示每一组：

1001 1111 0101
9 F 5

所以， $100111110101_2 = 9F5_{16}$

例0.5

把十六进制数29B转换为二进制数。

解：

2 9 B
= 0010 1001 1011

去掉第一个有效数字前的0，得：1010011011

0.1.6 十进制转换为十六进制

十进制转换为十六进制有两种方法：

1. 首先把十进制数转换为二进制，再转换为十六进制，例0.6就是用这种方法把十进制数转换为十六进制。
2. 把要转换的十进制数直接不断地除以16，并记下余数。这种方法由读者自己去证明。

例0.6

(a) 把 45_{10} 转换为十六进制。

32 16 8 4 2 1 首先转换为二进制
1 0 1 1 0 1 $32 + 8 + 4 + 1 = 45$
 $45_{10} = 0010\ 1101_2 = 2D_{16}$

(b) 把 629_{10} 转换为十六进制。

512 256 128 64 32 16 8 4 2 1
1 0 0 1 1 1 0 1 0 1
 $629_{10} = (512 + 64 + 32 + 16 + 4 + 1) = 0010\ 0111\ 0101_2 = 275_{16}$

(c) 把 1714_{10} 转换为十六进制。

$$\begin{array}{cccccccccccc}
 1024 & 512 & 256 & & 128 & & 64 & & 32 & & 16 & & 8 & & 4 & & 2 & & 1 \\
 1 & 1 & 0 & & 1 & & 0 & & 1 & & 1 & & 0 & & 0 & & 1 & & 0 \\
 1714_{10} = (1024 + 512 + 128 + 32 + 16 + 2) = 0110\ 1011\ 0010_2 = 6B2_{16}
 \end{array}$$

0.1.7 十六进制转换为十进制

十六进制和十进制间的转换也有两种方法:

1. 先把十六进制转换为二进制, 然后再转换为十进制, 例0.7证明了这种十六进制转换为十进制的方法。
2. 通过把各位十六进制数与其对应的权值的乘积相加得到的和, 就是十六进制转换为十进制的结果。

例 0.7

把下列十六进制数转换为十进制。

(a) $6B2_{16} = 0110\ 1011\ 0010_2$

$$\begin{array}{cccccccccccc}
 1024 & 512 & 256 & & 128 & & 64 & & 32 & & 16 & & 8 & & 4 & & 2 & & 1 \\
 1 & 1 & 0 & & 1 & & 0 & & 1 & & 1 & & 0 & & 0 & & 1 & & 0 \\
 1024 + 512 + 128 + 32 + 16 + 2 = 1714_{10}
 \end{array}$$

(b) $9F2D_{16} = 1001\ 1111\ 0010\ 1101_2$

$$\begin{array}{cccccccccccccccccccc}
 32768 & 16384 & 8192 & 4096 & 2048 & 1024 & 512 & 256 & 128 & 64 & 32 & 16 & 8 & 4 & 2 & 1 \\
 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\
 32768 + 4096 + 2048 + 1024 + 512 + 256 + 32 + 8 + 4 + 1 = 40\ 749_{10}
 \end{array}$$

0.1.8 十进制、二进制和十六进制间的运算

为了说明三种进制间的关系, 表 0.2 按顺序列出了 0~31 的十进制数, 以及与其相对应的二进制数和十六进制数。注意到每一个数值最高的位, 当它加1时, 该位就变成0, 并向其下一高位进1。例如, 在十进制中, $9 + 1 = 0$ 有一进位到下一高位; 在二进制中, $1 + 1 = 0$ 有一进位; 同样地, 在十六进制中, $F + 1 = 0$ 也有一进位。

表 0.2 三种进制间的运算

十 进 制	二 进 制	十六进制	十 进 制	二 进 制	十六进制	十 进 制	二 进 制	十六进制
0	0000	0	11	01011	B	22	10110	16
1	00001	1	12	01100	C	23	10111	17
2	00010	2	13	01101	D	24	11000	18
3	00011	3	14	01110	E	25	11001	19
4	00100	4	15	01111	F	26	11010	1A
5	00101	5	16	10000	10	27	11011	1B
6	00110	6	17	10001	11	28	11100	1C
7	00111	7	18	10010	12	29	11101	1D
8	01000	8	19	10011	13	30	11110	1E
9	01001	9	20	10100	14	31	11111	1F
10	01010	A	21	10101	15			

0.1.9 二进制数和十六进制数的加法

表 0.3 二进制加法

A+B	进 位	和
0+0	0	0
0+1	0	1
1+0	0	1
1+1	1	0

二进制加法运算是一个很直观的过程。表 0.3 列出了两个位之间的加法情况。因为计算机用二进制加法来实现减法, 所以这里不再讨论二进制减法。虽然计算机有加法器电路, 却没有单独的减法电路。用 2 的补码在加法器上实现减法运算。也就是说, 为了实现 $x - y$ 运算, 计算机用 y 的补码去与 x 做加法。2 的补码的概念在下文介绍。例 0.8 示出了二进制数加法。

例 0.8

求下列二进制数的加法，检查它们对应的十进制值。

解：

	二进制数	对应十进制
	1101	13
+	1001	9
	10110	22

0.1.10 2 的补码

得到二进制数的 2 的补码的方法是：将所有位按位取反后再加 1。将二进制位取反只需要将所有 0 变为 1，同时将所有 1 变为 0，取反后所得的数称为 1 的补码。见例 0.9。

例 0.9

求二进制数 10011101 关于 2 的补码。

解：

	10011101	二进制数
	01100010	1 的补码
+	<u>1</u>	
	01100011	2 的补码

0.1.11 十六进制数的加法与减法

在研究与计算机硬件和软件相关问题的时候，经常涉及十六进制数的加减法问题。掌握这些技术是很有必要的。下面对十六进制数的加法和减法进行单独讨论。

0.1.12 十六进制数加法

本节描述十六进制数的加法过程。从最低有效位开始，将各位加在一起。如果结果小于 16，则将结果作为对应位置的数字。如果结果大于（或等于）16，则将结果减去 16 并向高位进 1。为了更好地说明问题，最好是举一个例子，见例 0.10。

例 0.10

进行十六进制数加法：23D9 + 94BE。

解：

23D9	LSD:	9 + 14 = 23	23 - 16 = 7 产生进位
+	94BE	1 + 13 + 11 = 25	25 - 16 = 9 产生进位
	B897	1 + 3 + 4 = 8	
	MSD:	2 + 9 = B	

0.1.13 十六进制数减法

在十六进制数减法过程中，如果减数大于被减数，则从相邻的高位借 16。见例 0.11。

例 0.11

进行十六进制数减法：59F - 2B8。

解：

59F	LSD:	15 - 8 = 7
-	2B8	25 (由 9 + 16 得来) - 11 = 14 (E)
	2E7	4 (由 5 - 1 得来) - 2 = 2