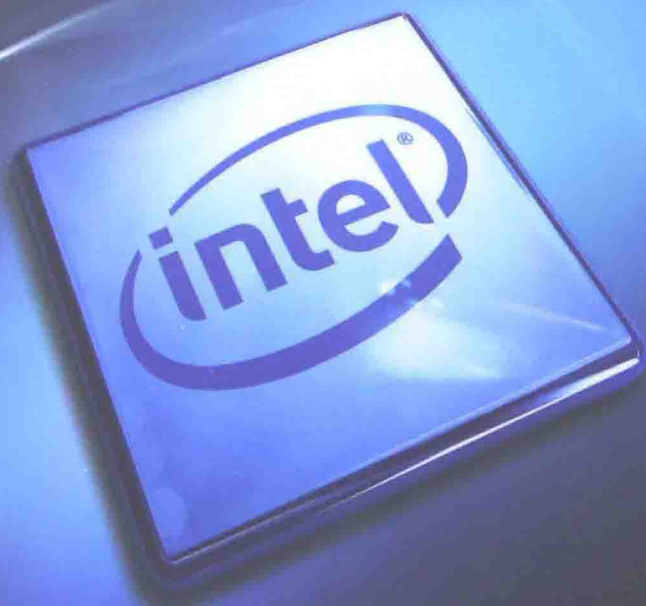


高等院校电子信息科学与工程规划教材

微机原理与 接口技术

——基本原理、实用技术和基于FPGA的SOC技术

潘松 潘明 黄继业 ● 编著



清华大学出版社

高等院校电子信息科学与工程规划教材

微机原理与接口技术

——基本原理、实用技术和基于 FPGA 的 SOC 技术

潘 松 潘 明 黄继业 编著

清华大学出版社

北 京

内 容 简 介

本书系统地讲解了微型计算机系统的结构、工作原理、接口技术及其应用,特别是将这些内容与现代 EDA 技术、FPGA 开发技术和 SOC 片上系统技术有机地融合起来,全方位强化和拓展了这一传统教学领域中的知识与技能传授的深度与广度。本书的基本内容包括 80x86 微处理器结构、指令系统、汇编语言程序设计、存储器系统、总线技术、中断技术、定时/计数接口技术和 DMA 技术、并行接口技术、串行接口技术、模拟接口技术和其他实用的接口技术,以及与这些内容相对应的基于超大规模 FPGA 的 SOC 技术。

本书可作为高等院校电子工程、通信、工业自动化、计算机等专业的本科生或研究生教材,也可用作相关专业技术人员的参考书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

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

微机原理与接口技术:基本原理、实用技术和基于 FPGA 的 SOC 技术/潘松,潘明,黄继业编著.

—北京:清华大学出版社,2015

高等院校电子信息科学与工程规划教材

ISBN 978-7-302-38905-7

I. ①微… II. ①潘… ②潘… ③黄… III. ①微型计算机-理论-高等学校-教材 ②微型计算机-接口技术-高等学校-教材 IV. ①TP36

中国版本图书馆 CIP 数据核字(2015)第 004854 号



责任编辑:苏明芳

封面设计:刘超

版式设计:魏远

责任校对:王云

责任印制:李红英

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

印装者:北京密云胶印厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:23.5

字 数:553 千字

版 次:2015 年 3 月第 1 版

印 次:2015 年 3 月第 1 次印刷

印 数:1~4000

定 价:39.80 元

产品编号:052808-01

前 言

尽管“微机原理与接口技术”在计算机、通信、电子工程、工业自动化和机械制造等专业学科中具有十分重要的地位，但至今，诸如为什么要在如此多的专业中安排学习这门课，为何长期以 8086/8088 微机系统及其接口作为学习的基本内容等问题，一直是热门话题。对此，作者认为传统“微机原理与接口技术”课程的基本内容仍然具有极大的重要性和必要性。

(1) 内容的典型性和完整性。从最早的计算机原型——图灵机，经过长期发展至今，计算机已遍布人类生活生产的各个角落。计算机也分化成了多种类型，如微控制器、手持式系统（手机、平板等）、笔记本电脑、PC 机、工作站、服务器、数据中心、大型计算机、超级计算机等。这些种类的计算机系统间，有些界限模糊，有些差异巨大，但共性的部分也有不少。PC 机是其中最为典型的代表，它比微控制器、手持式系统结构复杂，比工作站、服务器等更常见，比较适合学习。然而面对有限的课时数，现代 PC 机所涉及的原理与接口技术在一门课程中是无法讲述完整的。如现代 PC 机中的内部总线 PCIe 无法在短短的几节课中解释明白，更不用说编程实践了。那么，若把内容进行压缩作简要介绍，又失去了系统性和完整性，势必导致盲人摸象，学而不知其所以然。

于是 IBM PC 就成为了“微机原理与接口技术”这门课程最好的教学模型机。在有限的学时里，可以把 8086/8088 系统讲述完整，给读者一个完整系统的概念。

(2) 良好的兼容性及其原理的可延伸性。Intel 的 8086/8088 微处理器是 IBM PC 的核心，作为 16 位处理器，其存储器结构和指令集在 Intel x86 的历代处理器中被完整保留下来，无论是 32 位还是 64 位 x86，都有 8086 兼容的工作模式，可以做到对 8086 代码的二进制兼容。这种兼容给课程学习带来了方便，在任何一台 x86 的笔记本电脑或 PC 机上，不论是 Windows 系统还是 Linux 系统，简单安装上一个 VM 软件（虚拟机）即可再现 8086 环境。

事实上，虽然 PC 机发展了多年，但兼容性一直是 PC 机的突出优点，而且早期 PC 机——IBM PC 机在本质结构上和现代 PC 机是基本一致的，即虽然从早期的 8086、8088、80x86 到后来的 Pentium 系列以及现在的 Pentium 4 和多核技术，无论其制造工艺还是技术和性能，都有了极大的改进和提高，但从编程的角度看，仍然属于同一系列，且完全兼容，并无本质区别，指令系统从 8086 到 Pentium 系列，除部分保护模式下扩展指令外，80%以上完全相同；在应用程序中所用到的绝大多数指令依然是基本指令集中的指令，即 8086 指令集；虽然微机连接的外部设备越来越丰富，但中断工作的原理和输入/输出控制理念没有变化；此外随着大规模集成电路技术的发展，主板上曾经大量独立的接口芯片都已被集成到少数芯片中，但 8255、8259A、8254、16550 等可编程接口芯片依然存在且用法未变。

(3) 技术开放，资料全面详细。基于 8086/8088 的 IBM PC 系统中的各个部分，无论是 CPU 结构还是总线和外设接口都不止一次地被分析和重构过，导致 8086/8088 系统的技术资料非常容易被找到。而且，由于时效性的缘故，基本上不存在技术保护的壁垒。这样的教学模型机更加有助于对学有余力的同学进行课程内容外的学习。

(4) 课程内容更新和拓展的空间大。8086/8088 的学习者和教师，很容易迁移到 32



位 x86 系统。这是由于 486 以及 Pentium 的开源模拟器早已出现, 486 微机系统的 FPGA 全 HDL 描述也已经实现。因此以 8086/8088 为核心的微机原理与接口技术课程内容很容易更新为 32 位处理器架构的内容。此外, 8086/8088 系统的 HDL 描述完整透明, 可以在单片 FPGA 上重构 8086/8088 系统, 使教学内容更易于与 SOC 技术整合 (这也是本书重点讨论的内容), 提高教学成效, 开拓实用领域, 提高学习者的兴趣。

(5) 知识结构完整。早期 CPU 以 CISC 为主, 而现代新设计的 CPU 基本上都是 RISC 架构的, 例如, Intel 的 8086 指令属于 CISC, 而 ARM 是 RISC 架构的。从当前的计算机发展情况看, RISC 具有明显优势。于是有人简单地得出结论, x86 的架构体系在先进性上不及 ARM 架构。其实不然, 早在 1995 年, Intel 的 David B. Papworth 等设计了 x86 架构的 Pentium Pro 处理器, 在这个处理器中, x86 指令先被解码为类似于 RISC 指令的微操作, 再被 RISC 内核执行, 这种方式一直延续至今。因此, 以 x86 指令属于 CISC 指令集来否定 x86 显然是不对的。Intel 采用 CISC 和 RISC 融合的 x86 内部架构, 实现了兼容性和性能上的完美平衡, 绝不失先进性。

(6) 作为能力培养, 容易触类旁通。仅仅通过学时数并不多的一门课的学习, 学生便能掌握以 8086/8088 为核心的 PC 系统的原理和编程控制方法, 形成对计算机系统全面完整的系统概念。这些内容包括计算机系统的处理器、内存、外部存储器、输入/输出、通信接口以及大量的计算机系统的基本概念和软硬件知识等。而现代 PC 中的 PCIe 总线、SATA 硬盘接口、USB、DDR3、南桥芯片组都是 8086 微机原理与接口技术中的技术升级, 都包含了这些早期技术的影子。显然掌握这些基础内容, 容易触类旁通, 举一反三。

有不少用户曾提出, ARM 系统新颖实用, 应该取代传统的 Intel 8086/8088 系统, 成为“微机原理与接口技术”教学的主要内容, 作者并不认可。

(1) 典型性不够。ARM 公司的各代处理器以微控制器和手持系统为主。微控制器只是一种简化的计算机系统, 即手机与平板等手持式系统, 受制于体积和功耗, 往往是定制系统, 扩展性很差, 不具有典型性和一般性, 其接口知识也不易推广到其他系统。

(2) 开放性不够。ARM 公司本身不生产芯片, 而是通过把 ARM 处理器 IP 核授权给其他厂商进行系统生产, 这导致其尤其重视知识产权的保护, 因此从未公开过 ARM 处理器的详细内部结构, 且严厉打击仿制兼容 ARM 处理器或公开其内容的行为。因此关于 ARM 内部结构资料仅限于有限的公开资料和开发者的猜测。显然, 资料的不系统性、不完整性、应用对象的非一般性以及接口系统的不一致性, 成了此课程教学的最大瓶颈。此外, 普通学习者无法获得 ARM 系统的全 HDL 描述, 因此无法实现基于 FPGA 和全 HDL 的 SOC 系统创建, 也就无法更深入地从微观和宏观两方面认识和学习这一系统。

(3) 实践成本高, 学习有局限性。ARM 官方的软件工具价格昂贵, 对于一般的学习者, 面向修改内部设计只能纸上谈兵。实验设计中有较多开源或者授权的软件开发工具, 导致学习者多数情况下只能局限于软件编程, 对于其中的原理学习和对应的接口实践活动有很大的局限性。

(4) 学习内容无以此及彼的可推广性。各代 ARM 处理器之间不像 x86 那样具有良好的兼容性和继承性。例如, 已过时的 ARM7/ARM9 的接口技术和构架原理属于 ARM v4 架构, 已不同于现在 ARM 公司主推的 Cortex-A 和 Cortex-A50 系列的 ARM v7 架构。

显然, “微机原理与接口技术”中的传统内容仍然拥有不可替代的合理性、必要性和重



要性。当然,这并不意味着针对这些内容的教学和实验只能停留在原有的基础上。

随着超大规模集成电路设计技术、EDA 技术、大规模可编程应用技术、片上系统(SOC)技术和嵌入式系统等应用技术和推广,基于这些全新的技术层次,工程技术学科间的边界变得不断模糊,内容相互渗透,技术手段不断充实、学科知识不断融合、学科内涵不断延拓,应用领域不断扩大。于是,这些赋予了全新生命力的原本独立的领域具备或兼备了更一般化和更高效的研究与设计手段,并更有效快捷地融合或吸收了其他领域的先进技术和方法。这些变化很快在国内外高校的教学和实践中有了相应的反应。

这里不妨从 DSP 开始,对相关教学内容的变化作一简要说明。

早期的数字信号处理(Digital Signal Processing, DSP)可归为一个独立的学科领域,主要研究针对特定对象的 DSP 算法及 DSP 的技术实现。对于后者,最早的以及被广泛接受的是利用 TI 公司的数字信号专用微处理器来实现。而 TI 公司在每一片处理器芯片的表面印上了大大的 DSP 3 个字母,期待芯片使用者在潜移默化中将 TI 与 DSP 等同起来,将 Digital Signal Processing 和 Digital Signal Processor 等同起来,即暗示 DSP 中的 P 是 Processor。使用户逐渐确信 DSP 处理器就是 TI 生产的,而 DSP 算法则必须用 Digital Signal Processor (其实正确的表述应该是 Digital Signal Processing Processor)来完成。

TI 的这种概念偷换确实产生了效果。2002 年,一位负责《现代 DSP 技术》一书审稿的教授认为此书的题目不妥,不应该用 DSP,理由是全书完全没有涉及使用 TI 的 DSP 处理器的内容,甚至不涉及任何处理器的应用。我们给出版社的解释是,所谓 DSP 就是 Digital Signal Processing,绝非 Digital Signal Processor。DSP 只是一种算法,只要能快速完成这种算法,任何方法都能用。《现代 DSP 技术》一书就是讨论使用 FPGA 的纯硬件方式来实现 DSP 算法。由于 DSP 中乘法累加是最基本的算法,所以 Altera 公司在其技术资料中将嵌入于 FPGA 中可编辑硬件乘法器直接称为“DSP 模块”。按照 DSP 的算法要求,全部使用加法器和硬件乘法器等硬件模块搭建算法系统同样可以高效完成 DSP 算法;其工作效率、工作速度和工作可靠性都优于普通的 DSP 处理器,这在图形处理和数字通信中具有广泛应用。显然 DSP 技术与 EDA 技术是相通的。

DSP 的实现不仅可以借助于专用 DSP 处理器或纯硬件的算法系统完成,也能利用嵌入式系统来完成。例如,32 位 Nios II 可重构型嵌入式微处理器同样能完成不同类型的 DSP 算法,条件是针对具体的算法特征用 VHDL 描述对应的算法模块,然后将这些模块为 Nios II 处理器配置成专用指令,再通过 SOPC Builder(现在升级为 Qsys)软件将此指令融入 Nios II 的 C 编译器中,同时重构 Nios II,使其融入新指令的硬件结构。从此,在编程中就可在 C 环境中直接调用并编译具有高速运算性能和具备 DSP 算法功能的硬件指令。这种硬件指令是典型的精简指令(RISC),而对应的处理器就能运行这种指令。于是 DSP 技术、EDA 技术、嵌入式系统应用技术、片上系统技术和计算机组成与设计技术便融为一体。这部分内容我们写入了清华大学出版社出版的《SOPC 技术实用教程》一书中。

其实,此后不久,许多更新的技术和软硬件工具与以上提到的新技术平台产生了密切的交集。例如,著名的 MathWorks 公司推出的被广泛应用于许多研究与工程技术领域的业界杰出的数学集成与仿真工具软件 MATLAB,为 DSP、数字通信、电子工程、EDA、嵌入式系统和 SOC 等领域增添了设计与仿真平台以及与对应工具软件的输出接口。



例如,可以利用 MATLAB 的 Simulink 数学模块图形编辑平台,将数字通信领域的功能模块(如 DDS 模块)或 DSP 的算法模型(如 FIR 或 FFT 等模块)用图形表达出来,或是将能配置于 32 位 Nios II 嵌入式处理器软核的具有特定功能的精简指令(如用于 DSP 领域或电机控制领域的指令)表达出来,甚至将可编程嵌入式系统总线上接插的功能模块(如 VGA 显示控制模块或 PWM 信号发生模块或 LED 屏幕扫描显示控制模块等)表达出来。所有这些功能模块都可以在 MATLAB 平台进行功能测试和系统仿真,甚至利用后台的 Modelsim 软件进行功能仿真。基于 MATLAB/Simulink 的功能模块或系统模块的设计文件可向与之接口的转换软件输出,利用转换软件转换成指定的目标文件继续处理。例如,可以利用第三方的 DSP-Builder 将 MATLAB 输出的设计文件转换成便于逻辑综合的 VHDL 代码,再由与 DSP-Builder 接口的 Quartus II 软件针对选定的 FPGA 目标器件进行逻辑综合与结构综合,最后在芯片中实现既定的硬件功能。

由于 MATLAB 和处于后台的 DSP-Builder、Modelsim 和 Quartus II 等工具软件相互间有良好的接口,几乎所有设计、测试、仿真、硬件实现和硬件测试都可以在 MATLAB/Simulink 界面上控制完成;这包括算法模型的建模(如 FIR 模型)、系统仿真、功能仿真(后台调用 Modelsim)、VHDL 转换(后台调用 DSP-Builder)、逻辑综合(后台调用 HDL 综合器)、时序仿真、编程下载、嵌入式逻辑分析仪硬件测试等;如果涉及嵌入式系统的指令设计或接口模块设计,还要利用 SOPC-Builder 对 C 编译器和 CPU 进行重构。于是,各学科在更大的平台上得到了融合。我们将这其中的部分内容并入了清华大学出版社出版的《EDA 技术与 Verilog HDL(第 2 版)》一书中。

传统的单片机技术和应用曾一直是独立于其他学科领域的。单片机的应用开发只是围绕既定的单片机芯片来展开的,单片机系统的功能和适用领域仅取决于所用的单片机器件本身的结构特点和资源配置,而涉及的知识也不会超越单片机及其接口本身。然而随着 EDA 技术的发展和 SOC 技术的应用,以及许多有待开发的项目,在其高可靠、低成本、高速度、易升级等性能的要求下,一片传统意义下的单片机已难以适应,于是有了基于 SOC 片上系统的单片机系统开发需求。以 51 单片机为例,基于 SOC 的 51 单片机系统开发过程是,将几乎整个系统编辑构建于一个单片 FPGA 芯片中,其中包括 51 单片机 CPU 软核、嵌入其中的数据 RAM 和程序 ROM、接口控制模块、键盘和显示控制模块、A/D 和 D/A 采样和信号输出控制模块、各种通信模块等。当整个硬件系统构建、测试与仿真完成后,将进行软件设计与调试。于是单片机开发工程师必须具备 EDA 技术、硬件描述语言、SOC 开发技术等多个领域的知识。我们曾将这些相关的知识编写进了清华大学出版社出版的《单片机原理与应用技术》一书中。

新技术在教学领域的渗透是全方位的。当我国高校计算机专业大多数师生还围绕着庞杂而无实用意义的,由分离器件构建的 8 位模型机进行“计算机组成原理”课程的教学和实验并仍停留在原理认知和模型验证时,美国麻省理工学院已开设了“计算机系统设”课程,其本科学生在实验中基于硬件描述语言自主完成 ALU、不同指令周期的 CPU,乃至实现流水线 32 位 MIPS CPU 和 Cache 的设计;而斯坦福大学计算机系本科生对应的课程是“计算机组成与设计”。实验要求学生以各自独立的形式,用 VHDL 自主实现 CPU、VGA 显示控制模块等接口设计,最后实现于 FPGA 中,并完成软硬件调试。这些学校还将实用 CPU 设计



作为教学实践项目引入电子信息工程等专业中。在参考了这些学校的教学实践内容后,将传统的计算机组成原理课与 EDA 技术、SOC 技术、不同位宽不同体系结构 CPU 的设计技术融为一体,推出了《现代计算机组成原理》一书(科学出版社出版,第 2 版),被评为国家级精品教材。其中包含了基于 FPGA 单片硬件平台的 8 位、16 位和 32 位 CPU 的设计,以及基于微程序指令、状态机指令、RSIC 精简指令等不同指令形式的微处理器设计的教学与实验内容。

毫无疑问,对“微机原理与接口技术”的教学内容、实验方法和培养目标等方面都到了需要进行革新的时候,本教材正是在这样一个背景下推出的。

本教材将基于 FPGA 应用开发技术、EDA 技术、基于 SOC 的微机系统构建和测试技术等新内容引入“微机原理与接口技术”传统教学内容之中,为这一课程注入了全新的生命元素。所谓传统内容,即以 Intel 8086/8088 为基础,兼顾高性能微处理器 80x86 以及 Pentium 系列的介绍,包括微处理器结构、指令系统、汇编语言程序设计、存储器系统、总线技术、中断技术、定时/计数接口技术和 DMA 技术,以及并行接口技术、串行接口技术、模拟接口技术和其他实用的接口技术。读者将通过新融入的内容在更广阔的视野和更大的深度上提升对“微机原理与接口技术”课程的学习效果,同时将这些内容与 EDA 技术、FPGA 开发技术和 SOC 片上系统技术有机地融合起来,全方位强化和拓展了这一传统教学领域中的知识与现代技术技能的掌握。本教材的特色主要有如下 4 点:

(1) 将 EDA 的时序仿真技术全方位融入其中,利用时序仿真技术和在系统测试技术从多角度向读者展示接口器件及其电路系统的工作细节和时序特性。

(2) 所有涉及接口技术的章节都安排了针对本章内容的 SOC 技术和相应的实验项目,使读者从这些新颖的内容和极具启发性的实验中高效率地掌握课程给出的实用的接口技术及其基本原理。

(3) 各章的实践示例部分显示了明显的阶梯承接性关系,从最简 8088 SOC 微机模块的构建、仿真和硬件测试,到随章节的深入逐步加入的不同接口器件(包括 8255、8254、8259、16550、8237 等 IP 软核)的 SOC 微机系统的创建、软硬件调试,直至将整个完整的可运行不同操作系统的类 IBM PC/AT 的 8086 SOC 系统在一个单片 FPGA 中呈现给读者,使读者在基本原理方面、接口技术方面、完整系统构建方面以及现代技术的应用方面有了更好的理解和掌握。

(4) 全书奉献给读者的不仅是传统观念下的“微机原理与接口技术”,而是包括了基于 FPGA 开发技术和 SOC 技术的接口电路设计技术以及 EDA 的实用技术等,使传统的“微机原理与接口技术”知识在一个全新的平台上与大量的新知识新方法新技术新工具完成有机整合,从而能大步超越此课程原有的知识定位和学习目标。

本书内容具体安排如下:

第 1~4 章基本属于此课程的传统编排方式:第 1 章概述微机系统;第 2 章介绍 80x86 微处理器的功能和结构;第 3 章介绍 8086/8088 CPU 的指令系统;第 4 章介绍汇编语言程序设计。与传统所不同的是,为了适应此后加入的基于 FPGA 的 IBM PC 片上系统的构建与设计实践方面的内容,在第 2 章中增加了 8088 IP 软核接口和用法的内容,在第 4 章中增加了简化段定义格式的汇编程序结构。

第 5 章介绍存储器。本章最后部分启动了 SOC 的内容,其中包括在 Quartus II 平台调



用嵌入式存储器的方法、8088 微机最小 SOC 系统构建、仿真和软件测试。在时序仿真工具及其他在系统测试工具（如 In-System Memory Content Editor）的帮助下，读者可以清晰地观察和精确地定位 CPU 在指令代码运行过程中所有的时序以及存储器内容的实时变化情况，包括 8086/8088 流水线方式读取指令代码的时序细节。这对深入学习微机原理和接口控制时序十分有帮助。这些信息在传统方式下是完全无从获得的。

第 6 章介绍微机系统的输入/输出接口基本知识。最后，在第 5 章原有的 SOC 基础上，介绍了构建对应于第 6 章内容的不同接口电路的步骤、测试方法、硬件实现方法以及各种先进的测试工具的用法，如 In-System Memory Content Editor、In-System Sources and Probes Editor 或 SignalTap 嵌入式逻辑分析仪。使读者能进一步熟悉微机工作原理和熟悉微机系统构建及调试方法，将微机接口技术、EDA 技术与 SOC 技术更好地融合。

第 7~12 章分别介绍了微机系统中常用的可编程接口器件 8255A、8254、8259A、16550、8237A 和 A/D 与 D/A 控制模块的结构原理、基本用法和对应的接口技术。在各章的最后，针对该章的基本内容都安排了包含对应接口器件 IP 核的 SOC 系统构建、硬件仿真、软件调试和 FPGA 平台系统实验方面的内容，而且相关的知识点和软硬件设计内容是逐章累加的，使得读者能通过各章的理论学习 and 对应的软硬件实验，逐步建立起对微机及其接口系统的完整概念。

例如，在第 9 章（介绍中断接口技术和中断控制器 8259A 的用法）的一个实验示例中，为了能使系统自动演奏出《我的祖国》的乐曲，需要构建起一个更完整的系统；于是介绍了基于 SOC 的微机系统的更完整的构建、软硬件实现和测试过程。这个系统包括 8088 CPU 软核、嵌入式锁相环（为不同器件提供时钟）硬核、8254 软核、8255 软核、8259 软核、嵌入式存储器、总线控制模块、片选译码模块和对接口的控制模块等。其中许多内容是在前几章中介绍过和实验中用到过的模块和 IP 核。

第 13 章介绍了置于单片 FPGA 中的基于 8086 IP 的完整的 SOC 微机系统的结构、构建及运行等情况。这是一个经典 IBM PC 的单片微机系统，内部系统通过 FPGA 外围连接了键盘、鼠标、VGA 彩色显示器、硬盘（用 SD 卡代替）等接口设备（BIOS 设在 FPGA 内部），整个系统能运行 MS-DOS 和 Windows 3.0，并在这些操作系统上能运行许多传统软件。这部分内容无疑将进一步激发读者对微机原理与接口技术、SOC 技术和 EDA 技术的学习、掌握和应用的兴趣，并有利于培养读者的自主创新能力。

为了降低学习的门槛，本书中相关的实验尽量用原理图表述，尽量不涉及 HDL 编程。只要初步学习过 EDA 技术课程，就能直接学习此教材了。

为了利于读者的学习和实践，书中各章都安排了对应的设计示例、实验任务和自主设计项目，且大多数都能提供相关的工程设计源文件。

与此书相关的资料，包括本书的配套课件、实验示例源程序资料等文件资料都可免费索取，可浏览 www.kx-soc.com，或直接与清华大学出版社联系，欢迎读者对本教材相关内容进行交流和探讨（作者电子邮箱：pml123@sina.com.cn）。

本书作者潘明是桂林电子科技大学教师，潘松和黄继业均为杭州电子科技大学教师。

作 者

2015.1

目 录

第 1 章 微型计算机概述.....	1	2.6 80x86 系列高性能微处理器.....	28
1.1 微型计算机发展概况.....	1	2.6.1 80186/80188 微处理器.....	28
1.2 微型计算机系统的组成.....	2	2.6.2 80286 微处理器.....	28
1.2.1 微处理器、微型计算机与微型计 算机系统间的关系.....	2	2.6.3 80386 微处理器.....	30
1.2.2 微机系统基本结构.....	3	2.6.4 80486 微处理器.....	33
1.2.3 微机硬件系统的典型配置.....	4	2.7 奔腾系列微处理器简介.....	34
1.2.4 微机软件系统.....	5	2.7.1 Pentium 微处理器的特点.....	34
1.2.5 微型计算机的性能指标.....	6	2.7.2 Pentium 系列微处理器.....	36
1.3 微型计算机的特点及应用领域.....	7	2.7.3 Pentium 的工作模式.....	37
习题.....	8	2.7.4 酷睿 (Core) 系列微处理器.....	39
第 2 章 80x86 微处理器.....	9	2.8 IBM PC/XT 微型计算机系统 结构.....	40
2.1 8086/8088 微处理器.....	9	2.9 8088 IP 软核.....	41
2.1.1 8086/8088 的内部结构和工作 特性.....	9	2.9.1 引言.....	42
2.1.2 8086/8088 的内部寄存器.....	12	2.9.2 基于 8088 IP 软核的系统设计 特点.....	43
2.2 8086/8088 CPU 的引脚信号.....	15	2.9.3 8088 IP 的引脚信号.....	44
2.2.1 8086/8088 CPU 的总线周期基 本概念.....	16	习题.....	45
2.2.2 8086/8088 的地址总线和数据 总线.....	17	第 3 章 8086/8088 CPU 的指令系统...47	
2.2.3 8086/8088 的控制总线和基本 信号线.....	17	3.1 8086/8088 CPU 指令格式.....	47
2.3 8086/8088 总线接口器件和基本配 置电路.....	20	3.2 8086/8088 CPU 指令的寻址方式..	49
2.3.1 地址锁存器与双向三态驱 动器.....	21	3.2.1 立即寻址.....	49
2.3.2 总线控制器 8288.....	22	3.2.2 寄存器寻址.....	50
2.4 8086/8088 CPU 的总线工作 时序.....	23	3.2.3 存储器寻址.....	50
2.5 8086/8088 系统存储器和 I/O 组织.....	25	3.2.4 端口寻址.....	54
		3.3 DEBUG 的使用方法及其示例.....	54
		3.3.1 DEBUG 使用方法.....	54
		3.3.2 DEBUG 使用示例.....	57
		3.4 指令系统.....	58
		3.4.1 数据传送指令.....	59
		3.4.2 算术运算指令.....	63
		3.4.3 逻辑运算指令.....	72



3.4.4 移位指令	73	习题	124
3.4.5 串操作指令	75	第 5 章 存储器	127
3.4.6 控制转移指令	80	5.1 存储器的分类和性能指标	127
3.4.7 处理器控制指令	88	5.1.1 存储器分类方法	127
习题	89	5.1.2 存储器的性能指标	129
第 4 章 汇编语言程序设计	92	5.2 半导体存储器	130
4.1 汇编语言类型与格式	92	5.2.1 只读存储器 ROM	130
4.1.1 程序设计语言类型	92	5.2.2 随机存取存储器 RAM	132
4.1.2 汇编语句	93	5.2.3 新型的 DRAM	137
4.1.3 汇编语句基本格式	94	5.3 存储器容量扩展	139
4.2 汇编语言的基本语法	95	5.3.1 存储器片选控制方法	139
4.2.1 常数	95	5.3.2 存储芯片的扩展	141
4.2.2 数字与文字字符	95	5.3.3 存储器系统建立举例	143
4.2.3 标识符	96	5.3.4 IBM PC/XT 的存储器子系统 ..	143
4.2.4 变量	96	5.3.5 微机存储器层次结构	145
4.2.5 变量和复制操作符定义语句 ..	96	5.3.6 Cache 工作原理	146
4.2.6 表达式	97	5.4 FPGA 中的嵌入式存储器	148
4.2.7 一般运算符	98	5.5 基于 8088 IP 最简 SOC 构建与	
4.2.8 数值返回运算符	100	测试	149
4.2.9 合成运算符	101	5.5.1 建立工作库文件夹和存盘原理	
4.2.10 SHORT 运算符	102	图空文件	149
4.3 汇编程序结构形式	102	5.5.2 创建工程	150
4.3.1 简化段定义格式的汇编程序		5.5.3 全程编译	156
结构	103	5.5.4 时序仿真	156
4.3.2 简化段定义伪指令	103	5.5.5 分析时序仿真结果	158
4.3.3 完整段定义格式的汇编程序		5.5.6 引脚锁定	159
结构	106	5.5.7 编译文件下载	160
4.3.4 完整段定义伪指令	107	5.5.8 USB-Blaster 驱动程序安装	
4.4 DOS 和 BIOS 功能调用	109	方法	161
4.4.1 DOS 系统功能调用	110	5.5.9 8088 系统主存在线读写与软件	
4.4.2 BIOS 功能调用	111	调试	161
4.5 汇编语言程序设计	112	习题	163
4.5.1 顺序结构程序设计	112	实验 基于 8088 软核的 SOC 最简电路	
4.5.2 分支结构程序设计	113	系统创建和测试	164
4.5.3 循环结构程序设计	115	第 6 章 输入/输出接口技术	165
4.5.4 子程序设计	118	6.1 概述	165
4.6 宏指令	123		



6.1.1 接口电路的基本功能	165
6.1.2 典型的 I/O 接口结构	167
6.1.3 I/O 端口的寻址方法	169
6.1.4 PC/XT 微机 I/O 端口地址 分配	170
6.2 CPU 与外设之间的信息传送 方式	171
6.2.1 程序控制方式	171
6.2.2 中断控制 I/O 方式	175
6.2.3 直接存储器存取方式	179
6.3 微型计算机总线	181
6.3.1 总线基本概念	181
6.3.2 IBM PC 总线	183
6.3.3 ISA 总线	184
6.3.4 EISA 总线	184
6.3.5 VESA 总线	185
6.3.6 PCI 总线	185
6.3.7 AGP 总线	186
6.4 8088 SOC 微机系统基本输入/输 出电路设计	187
6.4.1 含简单接口电路的 8088 SOC 系 统的构建	187
6.4.2 时序仿真与硬件系统测试	188
6.5 In-System Sources and Probes Editor 使用方法	189
习题	192
实验 8088 SOC 系统的基本输入/输出 电路设计及软硬件测试	193
第 7 章 并行控制接口技术	195
7.1 8255A 的结构与工作原理	195
7.1.1 8255A 的内部结构	195
7.1.2 8255A 的外部引脚	196
7.1.3 8255A 的工作方式	197
7.1.4 控制字设置方法	201
7.2 8255A 应用举例	202
7.3 含 8255 软核的 SOC 构建与 测试	205
7.3.1 含有 8255 IP 的 8088 SOC 系统 的构建	206
7.3.2 软件设计与系统时序仿真	207
7.3.3 实用程序设计与硬件系统测试 实例	209
习题	210
实验 8088 SOC 系统的 8255 核的 基本应用	211
第 8 章 定时/计数接口技术	212
8.1 8254 的结构及其接口技术	212
8.1.1 8254 的内部结构与工作原理 ..	212
8.1.2 8254 的引脚信号	214
8.1.3 8254 芯片控制字与初始化 编程	215
8.1.4 8254 的工作方式	216
8.2 8254 初始化及其应用实例	222
8.3 SOC 微机系统中 8254 IP 的 应用	225
习题	228
实验	229
8-1 8254 核基本时序性能仿真与硬件 实测	229
8-2 多功能秒表设计	229
8-3 简易电子琴设计	229
8-4 四相步进电机控制信号发生器 设计	230
第 9 章 中断系统及其接口技术	231
9.1 中断概念与中断机制	231
9.1.1 中断原理与中断源	231
9.1.2 中断系统的功能	232
9.1.3 中断操作过程	234
9.2 8086/8088 微机中断系统	236
9.2.1 外部中断和内部中断	236
9.2.2 中断向量及其生成方法	238
9.2.3 8086/8088 CPU 响应中断的 过程	240

第1章 微型计算机概述

数字电子计算机系统俗称计算机或电脑，是一种具有高速数值运算、自动信息处理、逻辑判断和超强信息存储记忆功能的电子设备，也是一种能按照程序员事先安排的程序来工作，高速地处理各类数据信息的智能化设备。

电子计算机是人类现代科技的重大成就之一，它起始于1946年由美国宾夕法尼亚大学研制成功的第一台电子计算机ENIAC。这是一台由电子管构成的重达30吨并能按照事先安排的指令快速完成指定计算任务的现代意义上的电子计算机。自此，电子计算机及其相关技术经历了一个快速发展的过程。这期间，计算机的构成经历了电子管、晶体管、集成电路和大规模及超大规模集成电路的几个发展阶段，对应了电子计算机发展的4个时代，而微型计算机属于第4代电子计算机产品，属于超大规模集成电路计算机，是集成电路技术高度发展的产物。

电子计算机按体积、性能、用途和价格指标可分为巨型机、大型机、中型机、小型机和微型机5类。从工作原理上来讲，微型机与其他计算机并没有本质的区别，所不同的是微型计算机（简称微机）是直接面向个人用户的，其结构、功能和售价也是全面适应个人用户的。微型计算机的核心部件是微处理器，其性能与微处理器的性能直接相关。

微型计算机对现代社会的方方面面有着无可比拟的影响力。

1.1 微型计算机发展概况

微型计算机的发展是以微处理器的发展为特征的。微处理器自1971年问世以来，随着大规模集成电路技术的不断进步，在短短几十年的时间里以极快的速度发展。

1971—1973年，标志着第1代微机的4位和8位低档微处理器问世了，其代表性产品是Intel公司的4004和8008微处理器，分别是4位和8位微处理器。8008算得上是世界上第一种8位的微处理器，其集成度约4000个晶体管/片，芯片主频仅1MHz。

1974—1977年，出现了标志着第2代微机的8位中高档微处理器，其代表产品有Intel 8080、M6800和Z80等，集成度有了大幅提高，已具备了典型的计算机体系结构，包括中断、DMA等控制功能，主频约2MHz。由于有了较完善的指令系统，软件可采用BASIC、Fortran等高级语言及相应的编译程序。

1978—1984年进入了拥有16位微处理器的第3代微机时代。这个时代的微处理器代表产品是Intel 8086、Intel 8088、M68000和Z8000等，集成度和运算速度比上一代有了成倍的提高，主频时钟已超越了5MHz。处理器的指令系统也更加丰富和完善，也有了配套的操作系统和多任务处理功能。特别值得指出的是，1979年，IBM公司以8088作为个人计算机（PC）IBM PC系统的微处理器，自此开创了PC时代。1982年，Intel又推出了高



性能的 16 位处理器 80286, 主频达 20MHz, 各方面性能指标大幅提高。IBM PC/AT 微机就是基于这款处理器的代表机型。

1985—1992 年是 32 位微处理器时代, 对应第 4 代微机。其代表产品是 Intel 80386/80486, Motorola 的 M69030/68040 等, 最为经典的 80386 DX 集成度已高达每片 27.5 万个晶体管, 时钟频率达 33MHz, 而 Intel 于 1989 年推出的 80486 微处理器的集成度和速度要高于 80386 DX 4 倍, 特别是在其指令系统中首次采用了精简指令技术, 可以在一个时钟周期内执行一条指令, 大大提高了与内存的数据交换速度。

1993—2005 年是奔腾 (Pentium) 系列微处理器时代, 对应第 5 代微机。这个时代的微处理器的代表产品是 Intel 的奔腾系列芯片。这个系列芯片的集成度、运算速度和性能均大大优于上一代的 32 位微处理器, 使微机的发展在网络化、多媒体化和智能化等方面跨上了更高的台阶。

2006 年至今应该属于单片多核或 64 位微处理器时代。通常认为 Intel 多核的酷睿 (Core) 系列微处理器是第 6 代处理器。另一方面, AMD 公司于 2003 年推出了面向个人机的 64 位微处理器 Athlon 64 和 Athlon 64 FX 被认为开创了又一个新的微机时代。

随着大规模集成电路技术的不断发展, 微处理器的不断更新换代, 微型计算机正向着更微型、更高速、更廉价、更低能耗、更强功能和更多用途方向发展。

1.2 微型计算机系统的组成

微型计算机及其完整系统的结构和组成是与整个电子技术、集成电路技术以及计算机技术的发展紧密联系的, 而这一切又决定了微机系统的基本性能和功能。但就现代一个不算长的历史阶段而言, 典型的微机系统的组成有其一致性。

1.2.1 微处理器、微型计算机与微型计算机系统间的关系

1. 微处理器 (Microprocessor)

微处理器是整个微型计算机的核心部件和总控制单元, 决定了微机的基本性能; 微处理器就是把运算器、寄存器组和控制器集成在一个超大规模集成电路芯片上的功能部件, 是具有计算机系统运算和控制功能的中央处理单元 (Central Processing Unit, CPU)。

2. 微型计算机 (MicroComputer, MC)

微型计算机是以微处理器芯片为核心, 配以一定规模的半导体存储器、系统总线 (包括数据总线、地址总线和控制总线) 及其控制单元、硬盘、输入/输出 (Input/Output, I/O) 接口电路和辅助电路构成的装置, 构成了微型计算机的基本硬件结构, 即裸机。

3. 微型计算机系统 (MicroComputer System, MCS)

微型计算机系统是以微型计算机为主体, 配上输入设备、输出设备、外部存储器设备、



电源、机箱以及系统软件和应用软件组成的系统，简称“微机系统”。

单纯的微型计算机总体上只是一个硬件模块的组织结构，本身无法运行；如果在微机中建立起必要的软件，就能独立执行程序，完成运算和逻辑判断功能。但是，微机所运行的程序和所需的数据需要通过外部设备输入，而运算完成后的结果也需要显示器或打印机之类外部设备输出帮助显示或打印出来。

显然，微型计算机除了需要软件外，还要配备必要的输入和输出设备，所有这一切就构成了能投入实际应用的微型计算机系统。微处理器、微型计算机与微型计算机系统是 3 个含义不同但又密切相关的概念，对应着互为依存的各类软硬件部件，三者的关系可以用如图 1-1 所示的结构来描述。

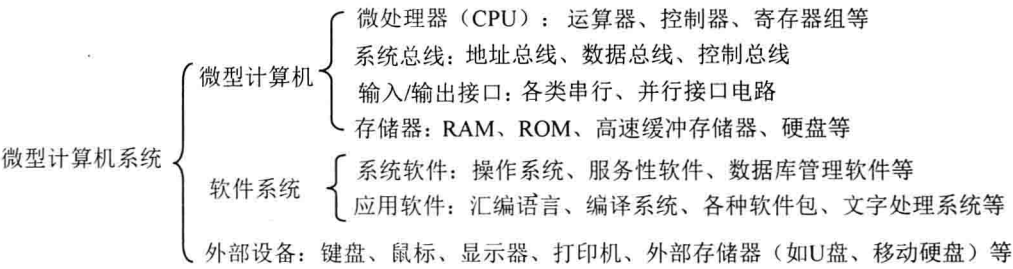


图 1-1 微型计算机系统的组成

1.2.2 微机系统基本结构

图 1-2 给出了微机系统的硬件组成基本电路结构示意图。微机硬件系统由微处理器、存储器、译码控制模块、系统总线控制模块、系统总线、输入/输出接口电路和各类外部设备等部件组成。在如图 1-2 所示的微机结构中，CPU 是核心部件，通过系统总线控制模块和译码控制模块，来控制地址总线、数据总线和系统总线，与其他部件进行数据和信息交换，完成指定的计算和控制任务。

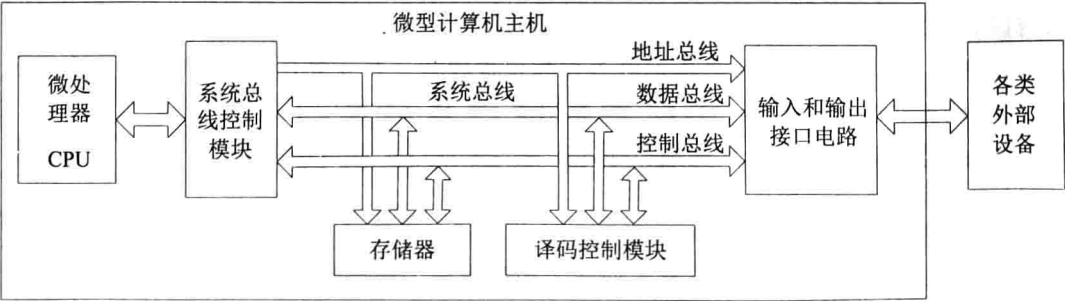


图 1-2 微机系统的硬件基本结构

具体而言，CPU 指挥控制模块，经由系统总线把需要执行的指令代码从存储器中取出，进入 CPU 后实行执行指令的操作，完成各类运算、逻辑判断，或对外设进行输入或输出的控制操作，从而完成程序员预设的各类任务。



在这个过程中,存储器是微机存放和记忆程序、数据的重要部件。通常把位于主机内部用于暂时存放程序和数据的存储器称为内存,也称为主存,由只读存储器(ROM)和随机存储器(RAM)两部分组成,属于主板(焊有所有微机芯片的印刷电路板,即PCB)内的存储器,而主板外或主机外部的用于存放大量信息的存储器则称为外存。

外存分为两大部分,一部分是硬盘存储器,用于存储微机运行必需的系统软件和应用软件。对于现代微机来说,微机的独立运算和控制已完全无法实质性地离开硬盘,因此硬盘存储器应该属于微机主机内部的部件,但属于主板外存储器。另一部分属于主机外部的,可以暂时不介入微机运行和控制的辅助型存储器,包括U盘、光盘、移动硬盘等大存储量存储器。

输入/输出接口电路是设置在外部设备(简称外设)与系统总线之间的专门电路,又称I/O接口,用于协调CPU与外设之间的信息和数据的交换。

系统总线包括数据总线、地址总线和控制总线,简称三总线。系统总线把CPU、存储器和输入/输出接口电路有机地连接起来,用来传送微机内部各部件间的信息;数据总线传送各类数据,其中包括指令代码、运算数据、中间数据和结果数据;地址总线上的信息主要是针对存储器具体存储单元或具体外设对象的地址信息,这些地址信息指出了数据的来源或传送的目的地;控制总线传送CPU对存储器或I/O设备的控制命令和I/O设备对CPU的请求服务的信号。整个系统总线的工作由系统总线控制模块统一安排和指挥。

外部设备,即I/O设备,可以分成两个部分,一部分是指直接为计算机本身的顺利运行而服务的,必须与外界及时进行信息与数据交换的输入和输出设备,如键盘、鼠标器、扫描仪、图形或文字识别仪、条码读入器、模/数和数/模转换器等;另一部分是暂时不会影响计算机运行的外设,如打印机、外部存储器(如U盘、光盘等)、开环控制对象(如步进电机)、大屏幕LED显示器、VGA显示器等。

1.2.3 微机硬件系统的典型配置

如图1-2所示,微机硬件系统由主机和外设组成。主机包括PCB主板、I/O接口卡(又称适配器)以及电源、机箱等部件;微机的外设很丰富,典型的外设有键盘、鼠标、显示器、打印机、光盘驱动器、网卡、摄像头等设备。

PC主机大部分电路包括集成电路芯片和接插模块,是安装在机箱内的PCB主机板上的,外部设备通过输入/输出接口与主机板相连。主板几乎集中了微机系统的全部功能,控制着系统各部分之间的指令流和数据流,能够根据系统的要求调度各个子系统并为实现系统的自动化管理提供完整的硬件保证。

主板上最常规的部件是CPU、系统支持芯片组、内存芯片、I/O接口、总线扩展槽、键盘或鼠标接口、可充电电池以及各种开关和跳线等,还集成了显示卡、声效卡、网卡、调制解调卡等接口部件。

主板上的存储器包括高速缓存和用作主存的大容量动态随机存取存储器。主板上还有一种只读存储器,用于上电自检程序、基本外设输入/输出控制程序和系统配置程序等。

主板上还安装有总线接口插槽,即总线插槽,可用于插接各类I/O接口卡。通过这些插槽,可将外设I/O接口卡连接到系统总线上,从而把外设与主机连在一起。



键盘、鼠标、扬声器的接口电路一般直接集成在系统主板上，由一专门处理器来控制，整个处理器负责将键盘按键产生的扫描码转换成能表示字符的 ASCII 码，而将鼠标送来的电脉冲转换成光标的移动数据，并产生相应中断后，把输入数据传送给 CPU 进行处理。

主板上的总线接口插槽通过插上面的各种功能用途的接口卡极大延伸了计算机的功能和用途，这些接口卡包括声卡、显示卡、图形采集卡、录音卡等。

现代微机中，外部设备除了能利用主机的接口插槽和对应的接口卡与系统连接外，也能直接通过主板上专设的 I/O 接口电路或者通过无线方式与主机系统连接，这些外设包括异步串口接口的外设、基于 USB 接口的外设和蓝牙设备，即无线接口设备等。

1.2.4 微机软件系统

微型计算机系统不但需要有功能齐全的硬件来支持，还必须要有专门的软件来配合，这样，计算机才能真正处理信息。软件就是能通过微机硬件系统完成特定任务的 CPU 指令代码的集合。微型计算机能正常工作的所有软件的总和称为软件系统。软件系统可分为系统软件和应用软件两大部分，其中，系统软件用来支持应用软件的开发和运行，主要包括操作系统、系统检测软件和编译软件等软件；而应用软件则是用来为用户解决某种具体问题的程序软件。

图 1-3 中列出了这两类软件的部分内容。

下面对这两类软件做简要说明。

1. 操作系统

在管理计算机中各部件协调工作的软件中，规模较大、功能较强、结构较复杂的软件称为操作系统（Operating System）软件，或称操作系统。操作系统的工作要点主要有如下两点：

（1）方便用户使用计算机。操作系统为用户提供了一整套方便使用整个计算机系统的方法，因此可以把操作系统看作是用户与计算机系统间的媒介或协调者。

（2）提高计算机的使用效率。在用户对计算机系统的具体硬件结构完全不熟悉的情况下，却可通过操作系统提供的一整套功能很强的命令调用相关软件来高效地使用计算机。

操作系统本身是由许多程序软件组成的，其中许多软件负责管理计算机底层硬件的各个部件进行协调工作。此外，各种应用程序、语言处理程序和辅助处理程序等软件都在操作系统的管理、协调和控制下运行。

2. 程序设计语言及其处理软件

计算机语言就是计算机程序的特定编写规则，依据这种规则编写出的程序将成为用户与计算机之间交流的媒介。人们需要通过编写的程序软件告诉计算机完成指定的任务。然而，计算机（或者说是 CPU）真正能“懂”的“语言”是被称为机器语言的程序编码（或指令代码）。因此，从程序员能方便地使用和辨认的计算机语言所对应的程序到 CPU 能直接辨认的指令编码集合，需要有一个类似于“翻译员”的软件来沟通程序员和 CPU 之间的

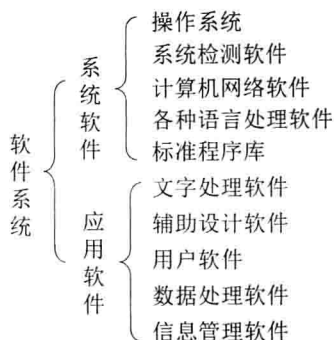


图 1-3 微机系统的软件分类