



普通高等教育“十二五”规划教材

◎ 电子信息科学与工程类专业 规划教材



微机原理 与接口技术

◎ 张颖超 叶彦斐 等编著



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

普通高等教育“十二五”规划教材
电子信息科学与工程类专业规划教材

微机原理与接口技术

张颖超 叶彦斐 陈逸菲 编著
杨春霞 侯健敏 周 媛

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是根据电气与电子信息类本科专业的共同要求而编写的,共13章,分为原理篇和接口篇。

原理篇包括微型计算机基础、8088/8086 CPU 及其指令系统、汇编语言程序设计、存储器技术、输入/输出技术、中断技术、总线技术等。指令系统和汇编语言部分均引入了与 C 语言的比较内容和 DEBUG 调试结果截图的比较,同时保证了比较内容的相对独立性。

接口篇以自动气象站设计为例,导入了 8259A、8253、8255A、8251A 等可编程接口芯片和 A/D 转换器。

本书内容深入浅出,通过案例导入、问题牵引、比较学习、点线面循序渐进等方式,实现了理论与应用的有机融合,使本书具有较强的可读性。本书适合作为高等学校电气与电子信息类各专业的教材或参考书,还可供从事微机系统设计与应用的工程技术人员阅读和参考。

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

微机原理与接口技术 / 张颖超等编著. —北京: 电子工业出版社, 2011.2

电子信息科学与工程类专业规划教材

ISBN 978-7-121-12229-3

I. ①微… II. ①张… III. ①微型计算机—理论—高等学校—教材②微型计算机—接口—高等学校—教材
IV. ①TP36

中国版本图书馆 CIP 数据核字 (2010) 第 217097 号

责任编辑: 韩同平 特约编辑: 李佩乾

印 刷: 北京季蜂印刷有限公司

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: 20.5 字数: 565 千字

印 次: 2011 年 2 月第 1 次印刷

印 数: 4 000 册 定价: 38.00 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。

前 言

一、关于微机原理与接口技术

“微机原理与接口技术”是高等学校电气与电子信息类专业的核心课程，强调学生在学习微型计算机基本原理的基础上建立软硬件结合的思维方法和培养解决实际问题的能力，侧重微机应用系统的设计。由于本课程各部分前后交叉、联系密切，不易自学，因此宜采用整体到局部再到整体，循序渐进、适当循环提高的知识组织体系和教学实践方法。

虽然微机技术的发展非常迅速，但是其基本体系结构、工作原理并没有改变，Intel x86 系列 CPU 保持了向下兼容的特性，而 8088/8086 CPU 是它们的基础。因此，在深入学习和掌握 8088/8086 CPU 基础上根据需要进行进一步学习和掌握现代微机的新技术是当前学习微机原理的合适方法。

从微机系统的组成来看，虽然存储器的容量迅速增大，外部设备越来越丰富，但是单元存储电路的工作过程、存储系统构建，以及相关接口的原理、方法均没有改变；CPU 与外设进行信息传送的方式没有改变，代表微机重要技术之一的中断技术的工作原理和管理方式也没有改变。因此以 8088/8086 CPU 构成的微型计算机可以说是现代微机技术的浓缩。

随着集成电路集成度的迅速提高，现代微机主板上已经看不到定时/计数器 8253（8254）、并行接口 8255A、串行接口 8250（8251）、中断控制器 8259，以及 DMA 控制器 8237 等可编程接口芯片和 8286 等中小规模集成电路，但是它们的基本功能仍然存在，只是集成到了更大规模的集成电路中去了，而且这些芯片在今天构成微机应用系统时仍然在使用。所以深入学习和掌握这些可编程芯片的基本原理和应用方法仍然是学习现代微机的基础，8088/8086 CPU 微机系统不仅是学习这些芯片原理和应用的合适平台，也是学习微机原理与接口技术的合理切入点。

二、本书的结构与内容

本书共 13 章，分为原理篇和接口篇。

原理篇由第 1~8 章组成。第 1 章介绍了微机的发展、结构及特点。以一个模型微型计算机执行指令的过程为切入点，建立了对微型计算机工作原理的初步认识。第 2 章以 8088/8086 CPU 为实例，讨论了 CPU 的构成和当今 CPU 的特点，并以此为基础讨论了总线操作与时序的概念。第 3 章 8088/8086 CPU 的指令系统和第 4 章汇编语言程序设计这两章均引入了与 C 语言的比较内容，实现了两种语言的互补学习；同时比较部分又有相对独立性，以满足仅进行 80X86 指令系统和汇编语言的学习需要。第 5 章存储系统，从存储器分类和单元存储电路入手，把存储原理和存储器与 CPU 的连接结合起来进行讨论，并引入了现代存储技术的基本内容。第 6 章输入/输出技术以接口、端口、输入/输出方式为主线，讨论了接口的功能、不同输入/输出方式的基本原理和特点。第 7 章中断技术，借中断式输入/输出的基础讨论了中断的概念和中断管理的方法，在此基础上阐明了 8088/8086 CPU 的中断系统。第 8 章总线技术，介绍了总线分类、功能、特点及常用的总线标准。本篇是学习接口篇的基础。

《接口篇》由第 9~13 章组成。本篇开始就以自动气象站的设计为例，导入了中断控制器 8259A（第 9 章）、定时/计数器 8253（第 10 章）、并行接口 8255A（第 11 章）、串行接口 8251（第 12 章）等可编程接口芯片和 A/D 转换器（第 13 章）。每一章都讨论了各个可编程接口芯片的功能、结构、初始化方法和应用设计。第 13 章对 A/D、D/A 转换器的转换原理、接口方法进行了系统讨论。在分步学习与设计的基础上最后通过集成设计完成一个基本的自动气象站系

统的设计工作。本篇在讨论各类常用接口芯片的基本功能和特点的基础上侧重应用。

三、本书特色

本书作者多年来一直致力于《微机原理与接口技术》课程的教学研究与改革,该课程获得了包括江苏省高校优秀课程和精品课程在内的多项教育教学成果奖,形成了鲜明的特色。

(1) 案例导入和问题教学相融合,抓住微机课程特点,注重理论联系实际

微机系统运行时各个部件同时工作,内容前后交叉联系密切,不可分离,但是教学中必须分开讲解,这是微机教学的特点也是难点,本书通过案例(自动气象站)导入和问题牵引相融合来解决这一难题,采用整体到局部再到整体,循序渐进、适当循环提高的方法来组织内容。

本书引入的自动气象站案例,把 8253、8255、8251、8259 等可编程接口芯片以及 A/D 转换器等全部合理导入;案例虽然具有一定的行业特色,由于在内容处理上合理回避了气象知识对学习的影响,使得任何人使用本书时只需关注微机原理和接口技术本身的内容。

(2) 通过比较教学法,借 C 语言基础,促汇编语言学习

汇编语言和 C 语言思维方式不同、语言间既有区别又有联系,而大部分学习微机原理的学生都有良好的 C 语言基础,通过比较教学实现互补,既能促进汇编学习,又能深化对 C 语言的理解和找准 C 语言概念与汇编语言的异同,对嵌入式系统中常用的混合编程能力提高大有裨益。

同时,为了使没有 C 语言基础或不习惯采用与 C 语言比较教学的读者在使用本书时不受影响,教材在该部分内容的处理上,保证了与 C 语言比较内容的相对独立性,可以像没有 C 语言存在一样使用。

(3) 巧借 DEBUG 调试结果辅助汇编学习,强化与实践的结合

在指令系统和汇编语言编程的讨论中巧妙借用 DEBUG 环境下的调试结果截图来说明指令的内涵、程序执行过程和 DEBUG 软件的使用,特别是对指令与标志位的关系和转移类指令目标地址与目标程序的关系的比较论述等问题的准确理解大有益处,同时也拉近了课堂教学与上机实习的距离。

(4) 本书配套的课件注重用多媒体手段刻画抽象问题,提高教学效果

计算机既具体又抽象,机器看得见摸得着,但指令运行过程和内部原理很抽象,我们借助动画等多媒体技术着重对其刻画,如指令的读取、存储、执行和指令间的关联、时序及输入/输出过程等。这有助于学习效率的提高。

本课程有很强的工程实践性,读者通过理论学习与以案例为主线的分层分级的实践动手训练,使其在分析问题和解决问题等方面受到系统科学的训练,以有效提高逻辑思维能力、理论联系实际能力、自主动手能力和创新能力。

(5) 本书适用面宽

本书是根据电气与电子信息类本科各专业的共同要求组织编写的,既适合自动化、信息工程、电气工程、电子信息工程、通信工程、物联网工程等电气信息类专业,也适用于测控技术及仪器等其他相关专业作为教材或参考书,还可供从事微机系统设计应用的工程技术书人员阅读和参考。计算机科学与技术专业选用本书时重点学习微机硬件原理和接口技术,注重强化软件和硬件的融合。

本书由张颖超教授等编著,河海大学叶彦斐编写第 8、12、13 章;南京信息工程大学张颖超编写第 6、7、9、10 章,陈逸菲编写第 3、4 章,杨春霞编写第 2、11 章,侯健敏编写第 5 章,周媛编写第 1 章。张颖超负责全书的统编定稿。

书中不足之处,敬请读者及时指正。衷心感谢所有阅读和关心本书的朋友!

编著者

目 录

原 理 篇

第 1 章 微型计算机基础	(2)
1.1 微型计算机发展概述	(2)
1.1.1 微型计算机的发展	(2)
1.1.2 微型计算机的特点	(3)
1.2 微型计算机系统的组成、结构与工作过程	(4)
1.2.1 微型计算机系统的组成与结构	(4)
1.2.2 微机系统的工作过程	(6)
1.3 常用数制与编码表示方法	(9)
1.3.1 计算机中常用的数制	(9)
1.3.2 计算机中信息的编码表示	(11)
1.4 微型计算机中的数据表示方法	(12)
思考题与习题	(13)
第 2 章 8086/8088 微处理器	(14)
2.1 8086/8088 微处理器特点	(14)
2.2 8086/8088 微处理器结构	(14)
2.2.1 8086/8088 功能结构	(14)
2.2.2 8086/8088 寄存器结构	(16)
2.2.3 8086/8088 存储器寻址	(18)
2.2.4 8086/8088 外部引脚	(19)
2.3 8086/8088 CPU 工作模式和引脚	(21)
2.3.1 8086/8088 最小工作模式	(21)
2.3.2 8088 最大工作模式	(23)
2.4 8088 的总线时序	(26)
2.4.1 最小工作模式典型时序	(27)
2.4.2 最大工作模式典型时序	(30)
2.5 微型计算机系统组成	(32)
2.6 高性能微机技术简介	(33)
2.6.1 流水线技术	(33)
2.6.2 RISC、SIMD 简介	(34)
2.6.3 MMX、SSE、SSE2 技术	(35)
思考题与习题	(35)
第 3 章 80X86 微处理器指令系统	(37)
3.1 寻址方式	(37)
3.2 指令系统	(41)
3.2.1 数据传送指令	(41)

3.2.2	算术运算指令	(46)
3.2.3	逻辑运算指令	(52)
3.2.4	移位指令和循环移位指令	(53)
3.2.5	处理器控制指令及标志位处理指令	(55)
3.3	32 位微处理器指令系统	(55)
3.3.1	32 位微处理器寄存器结构	(55)
3.3.2	32 位微处理器操作模式和指令特点	(56)
	思考题与习题	(56)
第 4 章	汇编语言程序设计	(59)
4.1	汇编语言基础	(59)
4.1.1	汇编语言的语句组成	(59)
4.1.2	常数与表达式	(60)
4.1.3	标识符	(61)
4.1.4	定义数据	(61)
4.1.5	分析操作符与合成操作符	(63)
4.1.6	符号定义	(64)
4.2	顺序程序设计	(65)
4.2.1	最简单的汇编程序	(65)
4.2.2	包含多个段的汇编程序	(66)
4.3	分支程序设计	(68)
4.3.1	转移指令	(68)
4.3.2	分支程序举例	(73)
4.4	循环程序设计	(77)
4.4.1	循环程序的基本结构	(77)
4.4.2	循环指令	(78)
4.4.3	循环程序举例	(80)
4.5	DOS 系统功能调用	(82)
4.6	子程序设计	(85)
4.6.1	调用与返回指令	(85)
4.6.2	过程定义	(88)
4.6.3	参数和结果传递的问题	(89)
4.6.4	子程序设计举例	(96)
4.7	字符串处理	(97)
4.7.1	字符串操作指令	(97)
4.7.2	重复指令前缀	(102)
4.7.3	程序举例	(103)
4.8	码转换程序设计	(104)
4.8.1	十六进制数到 ASCII 码的转换	(104)
4.8.2	二进制位串到 ASCII 串的转换	(106)
4.8.3	ASCII 码到十六进制数的转换	(107)
	思考题与习题	(108)

第 5 章 存储器技术	(111)
5.1 存储器概述	(111)
5.1.1 存储器分类	(111)
5.1.2 存储器性能指标	(112)
5.1.3 存储器系统结构	(113)
5.2 随机读写存储器	(115)
5.2.1 静态 RAM	(115)
5.2.2 动态 RAM	(117)
5.3 CPU 与存储器的连接	(121)
5.3.1 存储器扩展	(121)
5.3.2 存储器译码	(123)
5.4 现代微机的存储体系	(125)
5.4.1 Cache-主存存储层次	(125)
5.4.2 主辅存存储层次	(128)
5.4.3 并行主存系统及新型 RAM	(129)
思考题与习题	(130)
第 6 章 输入/输出接口技术	(132)
6.1 I/O 接口概述	(133)
6.2 I/O 端口及其编址方法	(135)
6.3 输入/输出方式及其接口	(138)
6.3.1 程序控制传输方式	(138)
6.3.2 中断传输方式	(143)
6.3.3 DMA 方式	(145)
6.3.5 4 种 I/O 方式的比较	(149)
思考题与习题	(149)
第 7 章 中断技术	(150)
7.1 中断的基本概念及其应用	(150)
7.2 中断系统的功能和中断优先级	(151)
7.3 8086/8088 的中断操作	(155)
7.4 中断响应过程与时序	(159)
7.4.1 硬件中断的响应过程和中断服务	(159)
7.4.2 软件中断的响应过程	(162)
思考题与习题	(164)
第 8 章 总线技术	(165)
8.1 总线概述	(165)
8.2 总线控制	(166)
8.2.1 总线仲裁	(167)
8.2.2 总线握手	(168)
8.3 常用总线标准	(170)
8.3.1 ISA 总线	(170)
8.3.2 PCI 总线	(172)
8.3.3 USB 总线	(175)

8.3.4 IEEE 1394 总线	(179)
8.3.5 SCSI 总线	(179)
8.3.6 AGP 总线	(180)
思考题与习题	(181)

接 口 篇

第 9 章 可编程中断控制器 8259A	(184)
9.1 8259A 的主要功能及结构	(184)
9.2 8259A 的工作过程	(187)
9.3 8259A 的中断管理	(189)
9.4 8259A 的编程	(194)
9.4.1 初始化命令字	(194)
9.4.2 初始化的程序流程	(196)
9.4.3 操作命令字	(197)
9.5 8259A 的应用举例	(200)
思考题与习题	(203)
第 10 章 可编程定时/计数器	(204)
10.1 8253 的引脚功能和编程结构	(205)
10.2 8253 的编程	(207)
10.3 8253 的工作方式	(210)
10.3.1 方式 0	(210)
10.3.2 方式 1	(211)
10.3.3 方式 2	(213)
10.3.4 方式 3	(214)
10.3.5 方式 4	(216)
10.3.6 方式 5	(217)
10.3.7 6 种工作方式的比较	(219)
10.4 8253 的综合应用举例——自动气象站瞬时风速和降雨量测量	(220)
10.5 8254 芯片	(222)
思考题与习题	(224)
第 11 章 可编程并行接口芯片 8255A	(226)
11.1 并行接口概述	(226)
11.2 8255A 概述	(227)
11.3 8255 的工作过程分析	(230)
11.4 并行接口应用举例	(234)
11.4.1 8255 作为打印机接口	(235)
11.4.2 8255 在自动气象站中的应用——风向信息采集、键盘和显示接口	(236)
思考题与习题	(243)
第 12 章 串行通信与可编程接口芯片 8251	(245)
12.1 串行通信基础	(245)
12.1.1 串行通信方式	(245)
12.1.2 数据传送方式	(247)

12.1.3 信号传输方式	(247)
12.1.4 串行接口标准	(248)
12.2 可编程串行接口芯片 8251A	(251)
12.2.1 8251A 概述	(251)
12.2.2 8251A 初始化编程	(254)
12.2.3 8251A 应用举例	(257)
思考题与习题	(260)
第 13 章 A/D、D/A 转换技术及其接口设计	(262)
13.1 D/A 转换器及其与主机的接口	(263)
13.1.1 D/A 转换器的基本组成	(263)
13.1.2 D/A 转换原理	(264)
13.1.3 D/A 转换器的的主要技术指标及选用要点	(266)
13.1.4 DAC0832	(268)
13.1.5 使用 D/A 转换器时应注意的几个问题	(270)
13.1.6 串行接口 D/A 转换器	(272)
13.1.7 D/A 转换器应用举例——D/A 转换器用于波形发生	(273)
13.2 A/D 转换器及其与主机的接口	(274)
13.2.1 A/D 转换接口	(274)
13.2.2 A/D 转换原理	(279)
13.2.3 借助计算机的软硬件资源实现 A/D 转换	(281)
13.2.4 A/D 转换器的主要技术指标、选择原则	(282)
13.2.5 模数(A/D)转换器 ADC0809	(284)
13.2.6 12 位 A/D 转换器 AD574	(288)
13.2.8 自动气象站集成设计	(291)
思考题与习题	(292)
附录 A DEBUG 的使用及汇编程序的编写与调试	(293)
A.1 查看 CPU 和内存, 寻址方式验证	(293)
A.2 编程、编译、连接、跟踪汇编程序	(296)
附录 B ASCII 编码表	(299)
附录 C 8086 指令集	(301)
附录 D IBM PC 中断号列表	(306)
附录 E 常用 DOS 功能调用 (INT 21H)	(308)
附录 F 常用 BIOS 功能调用	(311)
附录 G 基本逻辑门电路图形符号对照表	(313)
参考文献	(314)

原理篇

微型计算机发展迅速，结构特点鲜明。本篇首先以一个模型微型计算机执行指令的过程为切入点，建立对微型计算机工作原理的初步认识。接着以 8088/8086 CPU 为实例，讨论 CPU 的构成和当今 CPU 的特点，并以此为基础讨论总线操作与时序的概念，这有助于正确理解微机的工作原理和指令。指令系统和汇编语言部分均引入了与 C 语言的比较内容，实现了两种语言的互补学习；同时比较部分又相对独立，以满足仅进行汇编指令和汇编语言的学习需要。存储系统从存储器分类和单元存储电路入手，把存储原理和存储器与 CPU 的连接结合起来讨论，并引入了现代存储技术的基本内容。输入/输出技术以接口、端口、输入/输出方式为主线，讨论了接口的功能、不同输入/输出方式的基本原理和特点，实现了人与计算机的交流。中断技术借中断式输入/输出的基础讨论了中断的概念和中断管理的方法，在此基础上阐明了 8088/8086 CPU 的中断系统。总线是构成微型计算机应用系统的重要技术，介绍了总线分类、功能、特点及常用的总线标准。

本篇是接口篇学习的基础。

第 1 章 微型计算机基础

从 16 世纪初苏格兰人 John Napier 发表的一篇论文(其中提到一种可以计算四则运算和方根运算的精巧装置,这是最早的“类计算机”),到现今各个领域广泛使用的计算机,其中,微型计算机(简称微型机)作为重要的分支正在融入人们的日常工作、生活。

1.1 微型计算机发展概述

1.1.1 微型计算机的发展

1971 年,第一个 4 位的微处理器 Intel 4004 由英特尔(Intel)公司研制成功,利用该处理器组成的 MCS—4 作为世界上第一台微型机掀起了计算机大普及的浪潮。微处理器是微型机的核心部件,微处理器是将中央处理器(CPU)和一组称为寄存器(Registers)的特殊存储器集成在一片大规模集成电路或超大规模集成电路封装起来的器件。狭义地说,我们所说的微处理器就是 CPU。微型机的发展是以 CPU 的发展来表征的。

计算机的发展一般可划分为以下几个阶段。

第一代, 1971 年开始, 4 位和低档 8 位微处理器时代。典型产品有 Intel 4004(1971 年, 仅用于计算器样品研制, 改进型为 Intel 4040)和 Intel 8008(1972 年)。

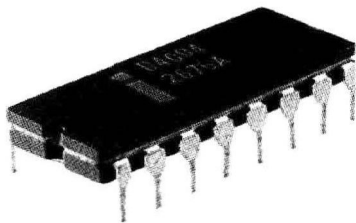


图 1-1 Intel4004

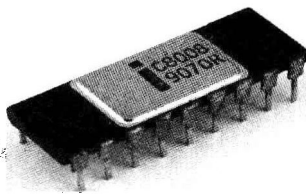


图 1-2 Intel 8008

Intel 4004 有 16 个引脚, 集成了 2300 个晶体管, 能执行 4 位运算, 最高频率为 740kHz, 平均执行指令时间为 $10\mu\text{s}\sim 15\mu\text{s}$, 支持 8 位指令集及 12 位地址。Intel 8008 有 0.5MHz 和 0.7MHz 两种频率, 能执行 8 位运算, 整体效能要比 Intel 4004 好很多。

第二代, 1973 年开始, 8 位微处理器时代。典型产品有 Intel 8080(1973 年), Motorola MC6800、MC6502(1974 年), Zilog Z80(1976 年), Intel 8085(1977 年)。

Intel 8080 集成了 6000 只晶体管, 主频为 2MHz, 平均执行指令时间为 $1\mu\text{s}\sim 2\mu\text{s}$, 有 16 位地址总线和 8 位数据总线, 包含 7 个 8 位寄存器, 支持 16 位内存, 同时也包含一些输入/输出端口, 这也是一个相当成功的设计, 有效地解决了外部设备内存寻址能力不足的问题。

第三代, 1978 年开始, 16 位微处理器时代。典型产品有 Intel 8086(1978 年), Intel 8088(1979 年), Zilog Z8000(1979 年), Motorola MC68000(1979 年), Intel 80286(1982 年)。

Intel 8086 集成了 29 000 只晶体管, 主频为 4.77MHz, 采用 16 位寄存器、16 位数据总线, 标志着第三代微处理器问世; Intel 8088 是一个以 Intel 8086 为基础的微处理器, 采用 16

位寄存器和 8 位外部数据总线。Intel 80286 处理器集成了大约 13 万个晶体管，最大主频为 20MHz，采用 16 位数据总线和 24 位地址总线。80286 有两种工作模式：真实模式和保护模式。在真实模式下，80286 直接访问内存的空间被限制在 1MB 内；而在保护模式下，80286 可以直接访问 16MB 的内存，并具有异常处理机制，这为后来的微软的多任务操作系统 Windows 准备了条件。

第四代，1983 年开始，32 位微处理器时代。典型产品有 Zilog Z80000(1983 年)，Motorola MC68020(1984 年)，Intel 80386(1985 年)，Intel 80486(1989 年)，Motorola MC68030、MC68040(1989 年)。

Intel 80386 集成了约 27.5 万只晶体管，主频 16~40MHz，平均指令执行时间小于 1 μ s，数据总线和地址总线都是 32 位，首次采用高速缓存(外置)解决内存速度瓶颈问题。80386 除了有真实模式和保护模式外，还增加了“虚拟 86”工作模式。在真实模式下，80386 直接访问内存空间被限制在 1MB；在保护模式下，80386-DX 可以直接访问 4GB 内存，并具有异常处理机制；在虚拟 86 模式下，可以同时模拟多个 8086 处理器来加强多处理能力。

Intel 80486 的前身是 Intel 80386，在结构上有很大的突破，它有内置资料快取芯片，一个浮点运算处理器和多重管线，在最佳的条件下，80486 的核心可以在一个时钟周期内处理一个指令。

第五代，1993 年开始，X86 架构处理器时代。典型产品有 Intel Pentium(1993 年)，Intel Pentium Pro(1995 年)，Pentium MMX(1996 年)，Pentium II(1997 年)，Pentium III(1999 年)，Pentium 4(2000 年)等。第五代微处理器是适应人们对多媒体信息和大流量数据处理要求产生的。

Intel Pentium 系列主频从 Pentium 的 66MHz 到 Pentium 4 的最高 3.6GHz，采用了全新的超标量架构，内部有两条管线并行工作，每个时钟周期执行两条运算指令，片内有两片 8KB Cache，节省了 CPU 的存取时间；Pentium 系列后期型号增加了 MMX 多媒体指令，从而拥有更强大的数据处理能力。

第六代，至今，64 位和双核微处理器时代。典型产品有 Intel Pentium D 系列，Core Duo 系列，Intel Itanium 系列，AMD Athlon(速龙)等。

内核是 CPU 的最主要部分(CPU 中间隆起的部分就是内核)，CPU 所有的计算、接收/存储命令、处理数据都由核心执行。

Core(酷睿)系列是 Intel 公司为结束使用长达 12 年之久的 Pentium 处理器而推出的。它采用先进的双核架构，将两个处理器内核封装在一起，支持新的 SSSE3 指令集，全面支持 64 位内存扩展技术。

AMD 和 Intel 的双核技术在物理结构上是不一样的。AMD 将两个内核做在一个晶元上，通过直连架构连接起来，集成度更高；Intel 则是将放在不同晶元上的两个内核封装在一起。

1.1.2 微型计算机的特点

微型计算机与之前的巨型机、大型机、中型机、小型机相比并没有本质上的区别，只是在结构上微型机的中央处理器(CPU)集成在了一块小硅片上，而大、中型机的 CPU 可由相当多的电路或集成电路组成。

微型计算机的特点可以概括为以下几点：

- (1) 标准的工业化装配结构，体积小、重量轻，系统扩展及性能升级容易；
- (2) 微型计算机的芯片集成度高，基本不需要人工焊点，降低了故障发生的概率，提高了

可靠性;

(3) 开放的标准体系结构和多元化的大规模工业生产, 使微型计算机的价格变得低廉;

(4) 微型计算机的应用范围广泛, 标准化的体系结构、超大规模集成电路的使用、规模化的生产, 使得微型计算机的性能价格比越来越高, 它的应用也越来越广泛。

1.2 微型计算机系统的组成、结构与工作过程

微型计算机和大、中、小型机一样, 基本结构都是基于冯·诺依曼结构的, 如图 1-3 所示, 由运算器、控制器、存储器、输入设备和输出设备 5 部分组成。

1.2.1 微型计算机系统的组成与结构

1. 微型计算机的组成

微型计算机包括微处理器、存储器、输入/输出(I/O)接口电路和系统总线, 如图 1-4 所示。它最大的特点就是将控制器和运算器集成在一起, 也就是我们常说的中央处理器(CPU)。随着电路集成度的提高, 整个 CPU 集成在一个集成电路芯片上, 统称为微处理器(Microprocessor): 一般在不引起混淆的情况下, CPU 就是微处理器。

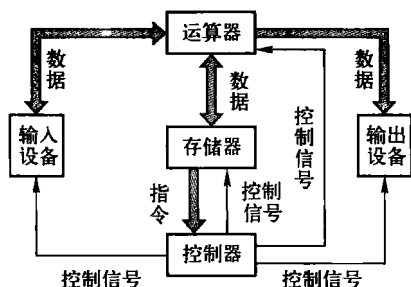


图 1-3 冯·诺依曼结构

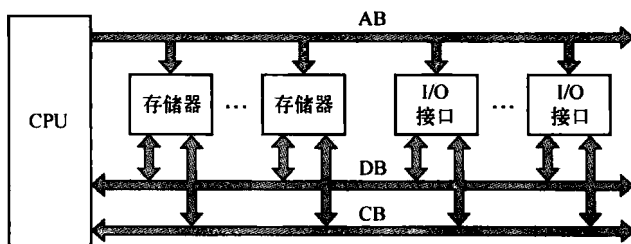


图 1-4 微型计算机的组成

(1) CPU

CPU(微处理器)是微型计算机的核心, 包括运算器、控制器和寄存器三个主要部分, 主要进行指令译码、算术逻辑运算, 通过外部系统总线和存储器或输入/输出接口电路来进行信息交换等。运算器又叫做算术逻辑运算单元(ALU), 主要负责算术逻辑运算; 控制器一般由指令寄存器、指令译码器和控制电路组成, 主要负责指令译码并根据指令要求, 发出相应控制信息; 寄存器主要负责存放经常使用的数据。CPU 的一个重要指标是字长, 即它在单位时间内一次可处理的二进制码的位数, 如 16 位机、32 位机、64 位机即指 CPU 的字长。

(2) 存储器

这里的存储器主要指主存储器(又叫内存储器), 是具有记忆功能的部件, 用来存放数据和程序。存储器中存放信息的最小单位叫存储单元, 每个存储单元都用一个地址来标识, 也就是说通过地址即可找到某个存储单元, 也就可以访问存储单元里存放的信息了。微机中常用的存储容量单位有“位(bit)”、“字节(Byte)”、“字(Word)”等, 1bit 即二进制中 0 或 1 所占大小, 1Byte = 8bit, 1Word = 2Byte = 16bit, 通常一个存储单元, 存放一个字节信息。若有 1 个字, 则需要 2 个存储单元存放, 根据高字节放高地址、低字节放低地址原则存储。图 1-5 示出的是存储器结构。

存储器读操作: CPU 发地址信息, 经过地址总线送到译码电路译码, 选中相应的存储单

元, CPU 发读控制命令, 相应存储单元中的数据通过数据总线被读出。注意, 被读存储单元中内容不变, 除非重新写入新的信息。

存储器写操作: CPU 发地址信息, 经过地址总线送到译码电路译码, 选中相应的存储单元, CPU 发写控制命令, 通过数据总线将数据写入相应存储单元。

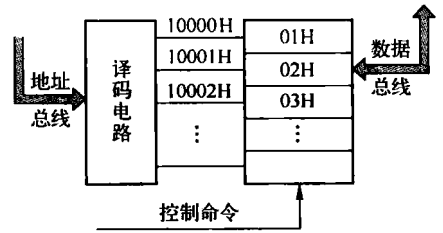


图 1-5 存储器结构

(3) 输入/输出接口

输入/输出接口电路是外围设备与 CPU 的连接电路, 在两者之间进行信息交换的过程中, 起暂存、缓冲、类型变换及时序匹配的作用。外部设备是指微型计算机上配备的输入/输出设备, 其功能是为微型计算机提供具体的输入/输出手段。常用的输入设备有键盘、鼠标器和扫描仪等, 常用的输出设备有显示器、打印机和绘图仪等, 磁盘、光盘既是输入设备, 又是输出设备。由于各种外部设备的工作速度、驱动方法差别很大, 无法与 CPU 直接匹配, 所以不能将它们简单地连接到系统总线上, 需要有一个接口电路来充当它们和 CPU 间的桥梁, 通过接口电路来完成信号变换、数据缓冲、与 CPU 联络等工作, 这种接口电路叫做 I/O 接口。

(4) 系统总线

系统总线负责在 CPU 和存储器、输入/输出接口电路间传送地址、数据、控制信息, 即地址总线(AB)、数据总线(DB)和控制总线(CB)。

除了 CPU 外, 还有 DMA 控制器和协处理器都具有控制系统总线的能力, 它们又叫做总线控制器; 在某一个时刻, 只能由一个总线控制器来控制系统总线; 在连接系统总线的各个设备中, 某一时刻只能有一个向总线发送信号, 但可以有多个设备从总线上同时获得信号。

地址总线输出将要访问的内存单元或 I/O 端口的地址, CPU 通过地址总线对存储器的存储单元或 I/O 端口寻址。地址总线宽度决定了 CPU 的寻址能力。一般地, n 根地址线可寻址 2^n 个存储单元。假设一个存储单元容量为一个字节, 则有 20 根地址总线的系统, 其存储器寻址范围为 2^{20} 字节即 1MB。

CPU 进行读操作时, 外部数据通过数据总线 DB 送往 CPU; 写操作时, CPU 中的数据通过数据总线送往外部。数据线的多少决定了一次能够传送数据的位数。

控制总线 CB 协调系统中各部件的操作, 有输出控制、输入状态等信号, 控制总线决定了系统总线的特点, 如功能、适应性等。

2. 微处理器的内部结构与基本功能

微处理器的内部主要包括算术逻辑单元(运算器, ALU)、累加器、标志寄存器、寄存器阵列、指令寄存器等, 以及定时、产生控制信号的电路。一个典型的 8 位微处理器结构如图 1-6 所示。

(1) 累加器(Accumulator, A)和算术逻辑单元(Arithmetic Logic Unit, ALU)

累加器是一种特殊的寄存器, 不做加法运算而是用于运算和传输过程中临时存储数据的; 算术逻辑单元主要完成算术和逻辑运算。ALU 有两个输入端和两个输出端, 其中一端接至累加器, 接收由累加器送来的一个操作数; 另一端通过数据总线接到寄存器阵列, 以接收第二个操作数。参加运算的操作数在 ALU 中进行规定的操作运算, 运算结束后, 一方面将结果送至累加器, 同时将操作结果的特征状态送入标志寄存器。

(2) 寄存器组

寄存器组包括通用寄存器、段寄存器、标志寄存器(FLAGS)和指令指针寄存器(IP)。通用寄存器又分为通用数据寄存器和通用地址寄存器, 可由用户灵活支配, 用来寄存参与运算的数

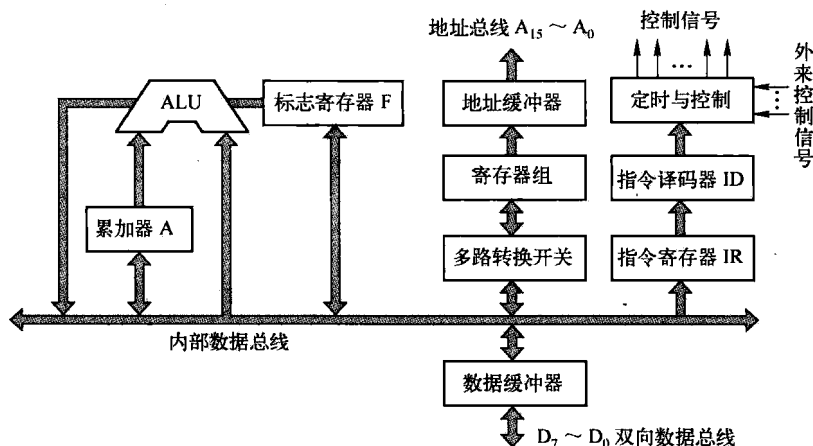


图 1-6 典型 8 位微处理器结构

据或地址信息；段寄存器用来存放存储器段基址；标志寄存器存放正在执行的程序的状态以及相应的控制信息；指令指针寄存器 (Instruction Pointer, IP) 的作用是指明下一条指令在存储器中的地址，每取一个指令字节，IP 自动加 1，如果程序需要转移或分支，只要把转移地址放入 IP 即可。

(3) 指令寄存器、指令译码器和定时及产生控制信号的电路

指令寄存器 (Instruction Register, IR) 用来存放当前正在执行的指令代码；指令译码器 (Instruction Decoder, ID) 用来对指令代码进行分析、译码，根据指令译码的结果，输出相应的控制信号；时序逻辑产生出各种操作电位、不同节拍的信号、时序脉冲等执行此条命令所需的全部控制信号。

(4) 内部总线和总线缓冲器

内部总线把 CPU 内各寄存器和 ALU 连接起来，以实现各单元之间的信息传送。内部总线分为内部数据总线和地址总线，它们分别通过数据缓冲器和地址缓冲器与芯片外的系统总线相连。缓冲器用来暂时存放信息 (数据或地址)，它具有驱动放大能力。

2. 微机系统

微机系统是由硬件系统和软件系统组成的。

硬件系统包括微处理器、系统总线、内存储器、I/O 接口和外部设备等。

软件系统是为了运行、管理和维护微机而编制的各种程序的总和，它包括系统软件和应用软件。系统软件通常包括操作系统、语言处理程序、诊断调试程序、设备驱动程序以及为提高机器效率而设计的各种程序。应用软件是指用于特定应用领域的专用软件，它又分为两类：一类是为解决某一具体应用、按用户的特定需要而编制的应用程序；另一类是可以适合多种不同领域的通用性的应用软件，如文字处理软件、绘图软件、财务管理软件等。

1.2.2 微机系统的工作过程

1. 相关概念

微机按照人的思维编写的各种工作程序自动地完成各种信息处理任务，它的工作过程实质就是执行程序的过程；微机在时钟脉冲作用下有条不紊地执行指令，即按照时序进行工作。时钟周期是计算机中最小的时间单位；利用总线进行数据传递所花的时间叫总线周期，一个基本

的总线周期需要 4 个时钟周期；完成一条指令所需的时间称为指令周期，一个指令周期往往包括多个总线周期。

2. 微机工作过程

程序是由一条条指令组合而成的，指令通常由操作码和操作数构成，操作码表示要进行什么样的操作，操作数则是参与操作的对象，机器内部指令是以二进制代码的形式出现的；执行一项信息处理任务的程序代码，以字节为单位按顺序存放在存储器的一段连续的存储区域内（程序存储）；微机工作时，CPU 中的控制器部分，按照程序指定的顺序（由代码段寄存器 CS 及指令指针寄存器 IP 指引），到存放程序代码的内存区域中去取指令代码，在 CPU 中完成对代码的分析，然后，由 CPU 的控制器部分依据对指令代码的分析结果，适时地向各个部件发出完成该指令功能的所有控制信号（程序控制）。

根据冯·诺依曼的设计，执行一条指令又可分为以下 5 个基本操作：取指令、分析指令、取操作数、执行指令、保存结果。

取指令是从存储器某个地址单元中取出要执行的指令送到 CPU 内部的指令寄存器暂存；分析指令又称为指令译码，把保存在指令寄存器中的指令送到指令译码器，译出该指令对应的操作信号，控制各个部件的操作；如果需要操作数，取操作数过程就是发出取数据命令，到存储器取出所需的操作数；执行指令即根据指令译码，向各个部件发出相应控制信号，完成指令规定的各种操作；如果需要保存计算结果，则把结果保存到指定的存储器单元中。

3. 一个程序工作的例子

以一个模型机为例来说明微机的工作过程。假设计算“12H + 34H”，我们知道需要编写程序，微机才能自动执行该操作，程序如下：

```
MOV  AL, 12H      ; 将 12H 送到累加器中
ADD  AL, 34H      ; 计算 12H + 34H, 结果送回累加器
```

编译后，两条指令对应的机器指令（二进制码）为：

```
10110000 00010010 ; “MOV AL, 12H” 的机器指令
                ; 10110000 是操作码, 00010010 是操作数
00000100 00110100 ; “ADD AL, 34H” 的机器指令
                ; 00000100 是操作码, 00110100 是操作数
```

两条指令译码后共有 4 个字节，存储器 1 个存储单元存放 1 字节信息，所以，这段程序在存储器中占 4 个存储单元（假设它存放在存储器的 10000H 开始的 4 个单元里）。

CPU 取第一条指令过程如图 1-7 所示，图中程序计数器 PC (Program Counter) 用以存放并生成指令地址，如程序载入存储器开始执行时，赋给 PC 第一条指令的首地址，本例中即为 10000H，将 10000H 单元中的内容送入数据缓冲器 (DR)，通过 PC 加 1 操作，可以依次取指令执行。

CPU 取第一条指令过程：

- ① PC 的内容 10000H 送到地址缓冲器。
- ② PC 的内容送入地址缓冲器后，PC 自动加 1，即由 10000H 变为 10001H。
- ③ 把地址缓冲器中内容 10000H 送到外部地址总线上至存储器，经地址译码器译码后，选中 10000H 单元。
- ④ CPU 发读控制命令。
- ⑤ 在读控制命令控制下，存储器把 10000H 单元中的内容（即第一条指令的操作码 10110000）