



普通高等教育“十一五”国家级规划教材

微型计算机原理及应用

第二版

侯晓霞 王建宇 戴跃伟 编著



化学工业出版社

普通高等教育“十一五”国家级规划教材

微型计算机原理及应用

第二版

侯晓霞 王建宇 戴跃伟 编著



化学工业出版社

· 北京 ·

本书紧紧围绕微型计算机原理和应用主题,以 8086/8088 为主线,系统地介绍了 16 位微型计算机的基本知识、基本组成和体系结构,8086/8088 系统中的指令系统、汇编语言及程序设计方法和技巧,存储器的组成和构成方法,常见的可编程接口芯片 Intel 8251、Intel 8253、Intel 8237、Intel 8259 和 Intel 8255 基本结构和应用,A/D、D/A 转换原理及典型芯片,并对现代微机系统中涉及的总线技术、高速缓存技术、数据传输方法、高性能计算机的体系结构和主要技术作了简要分析。

本书注重理论联系实际、突出实用技术,内容简明扼要、融入作者多年的经验和体会,可作为高等院校非计算机专业本、专科生微机原理或接口技术教材,也可作为工程技术人员学习和应用相关内容的参考材料。

图书在版编目 (CIP) 数据

微型计算机原理及应用/侯晓霞,王建宇,戴跃伟编著.

—2 版. —北京:化学工业出版社,2007.1

普通高等教育“十一五”国家级规划教材

ISBN 978-7-5025-9862-4

I. 微… II. ①侯…②王…③戴… III. 微型计算机-高等学校-教材 IV. TP36

中国版本图书馆 CIP 数据核字 (2007) 第 011650 号

责任编辑:唐旭华
责任校对:战河红

文字编辑:郝英华
装帧设计:潘 峰

出版发行:化学工业出版社(北京市东城区青年湖南街 13 号 邮政编码 100011)

印 刷:北京市振南印刷有限责任公司

装 订:三河市宇新装订厂

787mm×1092mm 1/16 印张 18½ 字数 490 千字 2007 年 1 月北京第 2 版第 1 次印刷

购书咨询:010-64518888 (传真:010-64519686) 售后服务:010-64518899

网 址: <http://www.cip.com.cn>

凡购买本书,如有缺损质量问题,本社销售中心负责调换。

定 价:29.00 元

版权所有 违者必究

第二版前言

本书自 2001 年 8 月出版以来,经过了 5 年的教学应用实践,先后被多所学校选作本科生同类课程的教材使用,2006 年被评为兵工高校优秀教材二等奖,本书所依托的课程“微机原理及应用”被评为江苏省精品课程。

随着课程教学实践的深入进行,以及大学本科(非计算机专业)培养方案的不断调整,我们对本书的内容编排、重点难点、学时安排等又有了新的认识,为此申报了国家“十一五”规划教材建设并获立项批准。本书第二版的修订思路是:以微型计算机的四大组成部分为主线来安排教材的章节,为此将原书的第 6 章进行了划分,形成目前的第 5 章定时与计数和第 6 章输入输出控制,这样,从微处理器、存储器、输入输出控制、输入输出接口、总线到软硬件应用(汇编程序设计和 A/D、D/A 转换),每一章均相对独立又互相配合,内容与难度上则循序渐进,有利于学生形成一个完整的微型计算机的概念,并了解其中的工作原理和处理流程。第 10 章高性能微机技术简介则给出了当前流行的且广泛应用于微处理器中的各种新技术,以开阔学生的视野。

本书第二版的修订注重基本概念和基本原理,有意识地减少了一些芯片内部较繁琐的原理说明,立足应用,尽量用较简洁、通俗的语言讲清与微机组成相关的基本概念和工作流程。对原书中的例题进行了大规模的改进并增加设计了大量新的例题,同时,对每一例题都给出了详细的设计分析思路,并尽可能地给出完整的硬件设计图和相应程序代码,使学生能通过这些例题,加深对基本概念及工作流程的理解,与文字讲述相得益彰。为达到举一反三的目的,有些例题还给出了进一步思考的问题,以引导、开阔学生的思路,其中不少例题的软硬件设计均可直接拿来应用在小型系统中。

本书第二版的另一特点是:针对学生反映本课程抽象难学的特征,我们编写了各章的学习指导,从学生学的角度出发,简明扼要、重点突出地指明本章的目的要求以及如何去学的方法,有利于学生掌握要点、明确方向、少走弯路,同时也有利于教师统一把握教学尺度。

本书的最佳参考学时为 64 学时,外加不少于 16 学时的实验与上机。我们有配套的实验指导书和多媒体教学课件供同行参考使用,如需要可联系:txh@cip.com.cn。

本书第二版的修订由侯晓霞任主编,王建宇、戴跃伟任副主编,在集体讨论之后,其中第 10 章由王建宇执笔修订,其余各章均由侯晓霞执笔修订。感谢林嵘老师和殷代红老师以及所有使用本书的学生对教材所提出的意见和建议。

虽执教多年但深知编写教材难度很大,尽管反复斟酌、修改仍难免不出纰漏,恳请各位同行在使用中多提宝贵意见,使本书能越写越好。

编者

2006 年 10 月于南京

第一版前言

近年来,随着科学技术的发展和工艺水平的提高,微处理器芯片在不断地进行更新换代,其字长、集成度、功能、结构等均有了长足的发展。目前,一台普通的微型计算机的功能已超过了20世纪70年代小型机甚至中型机的功能;由多个微处理器构成的系统几乎可以达到大型机的计算能力;而由高档微机组成的图形工作站,使得实时图像处理和网络化大型计算得以实现,从而使科学计算可视化技术走向大众;在工业生产上,由微机控制的自动化生产线,为提高生产能力和产品质量提供了保证;而大量由微处理器控制的仪器、仪表、家用电器、医疗设备等,已成为当今生活中不可缺少的一部分。总之,微型计算机及其应用技术随处可见,充满了生产和生活的各个方面,学习并掌握与之相关的内容,已成为高等院校的学生及社会各界人士的需求和迫切愿望。

为了使学生更好地掌握微机原理与接口技术的核心内容,根据我们多年教学的体会和经验,以及当前各大专院校普遍使用的16位微机实验设备的现状,同时也为满足21世纪培养高素质人才和教学改革的需要,在作者主持完成的“微型计算机原理及应用”课程被评为江苏省优秀课程、相应的教学体系建设成果被评为江苏省优秀教学成果的基础上,我们编写了《微型计算机原理及应用》一书。

本书以8086/8088为主线,讲述了8086/8088微处理器的组成原理、体系结构、汇编语言及程序设计技术、接口技术及应用的有关内容。考虑到学生对计算机知识学习的系统性和完整性,我们将当前高性能微机系统采用的新技术融合到各相关章节中进行了介绍,如高速缓存Cache,PCI、AGP总线技术,USB、1394通讯接口技术等,并在第10章重点分析了高性能微机的体系结构和采用的新技术,这样使学生对微机系统整体结构有一个完整的了解。在第2、3章中,详细介绍了指令系统和汇编程序设计方法,并加入了高级语言与汇编语言的交叉调用内容,同时,增添了同类教材涉及不多的,而且是计算机应用和使用所必须具备的总线技术和A/D、D/A内容。本书作者开发了与本书配套的多媒体教学软件,对购买本书量大的单位,我们将免费赠送该软件。

全书共分十章。第1章微型计算机概述,简要介绍微型计算机的发展、基本结构和工作过程,8086/8088CPU结构、存储器组织及典型时序分析。第2章8086/8088指令系统,主要介绍8086/8088寻址方式和指令系统,对常用指令的寻址方式及操作做了较详细的阐述,同时通过程序举例,帮助读者深入理解指令的功能。第3章汇编语言程序设计,讲述了汇编语言源程序的设计方法,伪指令格式,以及汇编语言程序和高级语言程序的相互调用。第4章存储器系统,论述了存储器的组成及工作原理、存储器系统扩展方法,并介绍了Cache的结构和工作原理等。第5章中断系统,描述了中断的概念及典型中断芯片8259的结构和应用。第6章DMA控制器和定时/计数器,主要介绍了DMA和定时/计数器的工作原理和典型芯片8237、8253的结构和应用。第7章接口与串并行通信,介绍常用并行和串行接口芯片的结构及其与CPU接口方式和编程,同时增加了通用串行接口规范USB及1394技术。第8章总线技术,主要描述总线的有关概念,总线的类别及功能,常用总线的有关规范等。

第9章 D/A、A/D 转换器及其与 CPU 的接口，讲述了数模转换器和模数转换器的一般工作原理，重点介绍与 CPU 的接口技术及其编程。第10章高性能微机系统新技术简介，讲述了 PⅡ、PⅢ系列微机中采用的新技术，重点分析了 MMX、SSE 技术和体系结构涉及的寄存器结构、工作方式、存储管理及存储保护技术。每章都有习题与思考题，以便帮助读者理解和掌握有关内容。

本书第6、8、9章由戴跃伟编写，第4、5、7章由侯晓霞编写，其余各章由王建宇编写，全书由王建宇统稿。杨洋、钟晓霞、袁秋林、项文波、施友松、王翔、陈果、秦华旺等参加了本书插图的绘制，在此表示感谢。

由于水平所限，书中难免有谬误之处，恳请广大读者批评指正。

编者

2001年3月于南京

目 录

1 微型计算机概述	1
1.1 微机的发展与特点	1
1.1.1 微机的发展历史	1
1.1.2 微机的特点	2
1.2 微机的组成结构与工作过程	2
1.2.1 微机的组成结构	2
1.2.2 微机的工作过程	4
1.3 8086/8088 微处理器	7
1.3.1 8086/8088CPU 的编程结构	7
1.3.2 存储器组织	10
1.3.3 8086/8088CPU 的工作模式与引脚 功能	11
1.3.4 系统典型配置	15
1.4 典型时序分析	16
1.4.1 基本概念	16
1.4.2 8086/8088 微机系统的基本 操作	17
1.4.3 最小模式下的典型时序	17
1.4.4 最大模式下的典型时序	19
习题与思考题	21
本章学习指导	21
2 8086/8088 指令系统	23
2.1 8086/8088 寻址方式	23
2.2 8086/8088 指令系统	26
2.2.1 数据传送指令	27
2.2.2 算术运算指令	31
2.2.3 逻辑运算和移位指令	36
2.2.4 程序控制指令	39
2.2.5 串操作类指令	43
2.2.6 标志处理和 CPU 控制类指令	46
习题与思考题	46
本章学习指导	48
3 汇编语言程序设计	50
3.1 汇编语言的基本元素	50
3.1.1 汇编语言的语句格式	50
3.1.2 汇编语言的运算符	51
3.1.3 表达式	53
3.1.4 汇编语言程序汇编步骤	54
3.2 伪指令	55
3.2.1 定义数据伪指令	55
3.2.2 符号定义伪指令 EQU、 PURGE 及 =	56
3.2.3 段定义伪指令 SEGMENT 和 ENDS	56
3.2.4 段寄存器定义伪指令 ASSUME	57
3.2.5 过程定义伪指令 PROC 和 ENDP	57
3.2.6 宏指令	59
3.2.7 定位伪指令 ORG	59
3.2.8 汇编结束伪指令 END	60
3.3 汇编程序设计	60
3.3.1 顺序程序设计	60
3.3.2 分支程序设计	62
3.3.3 循环程序设计	63
3.3.4 子程序设计	72
3.3.5 MASM 与高级语言的接口	75
3.3.6 DOS 功能调用	79
习题与思考题	81
本章学习指导	82
4 存储器系统	84
4.1 概述	84
4.1.1 存储器分类	84
4.1.2 存储器系统结构	85
4.2 读写存储器 RAM	86
4.2.1 静态 RAM	86
4.2.2 动态 RAM	88
4.3 只读存储器 ROM	91
4.3.1 掩模 ROM	91
4.3.2 可编程的 ROM	91
4.3.3 可擦除可编程的 ROM	92
4.3.4 电可擦除可编程的 ROM	94
4.3.5 快擦型存储器	95
4.4 存储器芯片扩展及其与 CPU 的	

连接	95	6.3.4 8259A 的工作方式小结	151
4.4.1 存储器芯片与 CPU 的连接	95	6.3.5 8259A 的应用举例	152
4.4.2 存储器芯片的扩展	96	6.4 可编程 DMA 控制器 Intel 8237A	154
4.5 高速缓冲存储器 Cache	101	6.4.1 8237A 的编程结构与主要 功能	154
4.5.1 主存-Cache 层次结构	101	6.4.2 8237A 的编程	158
4.5.2 Cache 的基本工作原理	102	6.4.3 8237A 的操作时序	163
4.5.3 地址映像	103	6.4.4 DMA33/66/100 简介	163
4.5.4 替换策略	105	习题与思考题	164
4.5.5 PⅢ中采用的 Cache 技术	105	本章学习指导	165
4.6 虚拟存储器	106	7 串并行通信及其接口技术	168
4.6.1 主存-辅存层次结构	106	7.1 CPU 与外设之间的数据传输	168
4.6.2 虚拟存储器的基本概念	106	7.1.1 CPU 与 I/O 接口	168
4.6.3 页式虚拟存储器	108	7.1.2 I/O 接口与系统的连接	169
4.6.4 段式虚拟存储器	109	7.2 可编程并行接口芯片 Intel 8255A	171
4.6.5 段页式虚拟存储器	110	7.2.1 并行通信与接口	171
习题与思考题	110	7.2.2 8255A 的编程结构	171
本章学习指导	111	7.2.3 8255A 的引脚功能	172
5 定时与计数	113	7.2.4 8255A 的工作方式	173
5.1 概述	113	7.2.5 8255A 的初始化编程	177
5.1.1 定时与计数问题的提出	113	7.2.6 8255A 的应用	179
5.1.2 端口的概念	113	7.3 可编程串行接口芯片 Intel 8251A	185
5.2 可编程定时器/计数器芯片 Intel 8253	114	7.3.1 串行通信基础	185
5.2.1 8253 的功能与结构	114	7.3.2 8251A 的基本功能	188
5.2.2 8253 的初始化编程	116	7.3.3 8251A 的内部结构	189
5.2.3 8253 的工作方式	119	7.3.4 8251A 的引脚功能	190
5.2.4 8253 的应用	122	7.3.5 8251A 的编程	192
5.2.5 其它定时/计数芯片	124	7.3.6 8251A 应用举例	195
习题与思考题	125	7.4 通用串行接口标准	197
本章学习指导	126	7.4.1 通用串行接口 USB	197
6 输入输出控制	128	7.4.2 1394 接口	199
6.1 输入输出数据的传输控制方式	128	习题与思考题	200
6.1.1 程序方式	128	本章学习指导	201
6.1.2 中断方式	132	8 总线技术	203
6.1.3 DMA (Direct Memory Access) 方式	135	8.1 总线标准与总线传输	203
6.2 8086/8088 的中断操作	137	8.1.1 总线标准与分类	203
6.2.1 中断分类与中断类型码	137	8.1.2 总线传输	203
6.2.2 中断向量与中断向量表	138	8.2 PC 总线	206
6.2.3 中断响应过程与时序	139	8.2.1 ISA 工业标准总线	206
6.3 可编程中断控制器 Intel 8259A	141	8.2.2 EISA 扩展的工业标准结构 总线	209
6.3.1 8259A 的结构及主要功能	141	8.2.3 VESA 总线	210
6.3.2 8259A 的编程	144	8.2.4 PCI 总线	210
6.3.3 8259A 的级联	149	8.2.5 加速图形端口 (AGP)	212

8.3 系统总线	213	10.1.1 标量流水工作原理	241
8.4 通信总线	213	10.1.2 超流水线超标量方法	242
8.4.1 IEEE 488 总线	214	10.1.3 超长指令字 (VLIW) 技术	242
8.4.2 RS-232C 总线	217	10.1.4 其它相关技术	242
8.4.3 RS-423A/422A/485 总线	217	10.2 RISC、SIMD 简介	243
习题与思考题	219	10.2.1 RISC 简介	243
本章学习指导	219	10.2.2 SIMD 技术简介	244
9 D/A、A/D 转换与接口技术	220	10.3 MMX、SSE、SSE2 技术	244
9.1 D/A 转换器的工作原理	220	10.3.1 MMX 技术	244
9.1.1 权电阻网络 D/A 转换器	220	10.3.2 SSE 技术	246
9.1.2 R-2R T 型电阻网络 D/A 转换器	221	10.3.3 SSE2 技术	249
9.1.3 $2^n R$ 电阻分压式 D/A 转换器	221	10.4 操作方式和寄存器	249
9.1.4 集成化 D/A 转换器	222	10.4.1 操作方式	249
9.2 数/模转换器芯片 (DAC) 及其接口 技术	222	10.4.2 基本执行环境	250
9.2.1 D/A 转换器的主要性能参数	222	10.4.3 用户级数据结构与寄存器组	252
9.2.2 D/A 转换器芯片 DAC0832	223	10.4.4 系统级数据结构与寄存器组	254
9.2.3 数/模转换器芯片与微处理器接口 时需注意的问题	227	10.5 存储管理	264
9.3 模/数转换芯片 (ADC) 及其接口 技术	229	10.5.1 存储器管理概述	264
9.3.1 从物理信号到电信号的转换	229	10.5.2 物理地址、线性地址与逻辑 地址	265
9.3.2 采样、量化与编码	229	10.5.3 分段技术	266
9.3.3 A/D 转换器的工作原理	230	10.5.4 分页技术	268
9.3.4 A/D 转换器的性能参数和 术语	231	10.5.5 物理地址扩展	273
9.3.5 A/D 转换器芯片 ADC0809	231	10.6 存储保护	277
9.3.6 模/数转换器芯片与微处理器 接口需注意的问题	236	10.6.1 段页保护机制	277
习题与思考题	239	10.6.2 段限与类型的保护校验	278
本章学习指导	239	10.6.3 特权级	280
10 高性能微机技术简介	241	10.6.4 指针验证	281
10.1 流水线技术	241	10.6.5 校验对界	281
		10.6.6 页面级保护	282
		习题与思考题	283
		附录 ASCII (美国标准信息交换 码) 表	284
		参考文献	285

1 微型计算机概述

1.1 微机的发展与特点

1.1.1 微机的发展历史

微型计算机作为计算机大家族中的一员，它具有一般计算机的所有特性。我们知道，计算机的核心部件是 CPU，它是整个计算机的心脏，控制着计算机的全部工作。而微型计算机的核心部件是微处理器，微处理器是微型计算机中的 CPU。因此，我们讲微型计算机的发展历史就是讲微处理器的发展历史。

1971 年，Intel 公司推出了第一个微处理器芯片 Intel 4004。这是一个 4 位的微处理器，原本是为高级袖珍计算器设计的，推出后却取得了意想不到的成功。Intel 公司立刻对其进行改进，正式生产出第一片通用微处理器芯片 Intel 4040。此后，Intel 公司又推出了 8 位的通用微处理器芯片 Intel 8080。这些微处理器一般被称为第一代微处理器，它们的字长为 4~8 位，时钟频率为 1MHz。据资料报道，它们在许多低端应用领域（如智能玩具等）仍有较大市场。

很快，不少厂商投入到微处理器的设计生产领域。1973~1977 年之间，Intel 公司又在原 8080 的基础上提高集成度，设计生产了 Intel 8085。同时，Zilog 公司推出了著名的 Z80，该芯片曾长期占据单板机市场，成为组成单板机的主流处理器。另外，Rockwell 公司设计的 8 位微处理器 6502，也曾作为著名的 Apple 机的 CPU，对微型计算机的发展起到了极大的推动作用。这些微处理器一般被称为第二代微处理器。它们的字长为 8 位，时钟频率为 2~4MHz。

1978 年，Intel 首次推出 16 位微处理器 8086，这是 80x86 系列 CPU 的鼻祖。8086 的内部和外部数据总线都是 16 位，地址总线为 20 位，可直接访问 1MB 内存。1979 年，Intel 又推出 8086 的姊妹芯片 8088，它与 8086 不同的是外部数据总线为 8 位，以适应当时已广泛使用的 8 位接口芯片。很快 Intel 就在 8086/8088 上取得了巨大成功，IBM 选用它来制造著名的 IBM PC，开辟了一个全新的个人计算机时代，而 8086/8088 的许多设计思想则一直影响着 Intel 的后续芯片。此外还有 Zilog 公司推出的 Z8000 和 Motorola 公司的 68000，这些微处理器一般被称为第三代微处理器。

20 世纪 80 年代以后，随着集成电路设计生产技术的提高，微处理器进入快速发展阶段，生产厂商在提高集成度、速度和功能方面取得了很大进展。Intel 公司相继推出了 80286、80386 和 80486，这些高性能的微处理器的字长已达到 32 位，时钟频率可高达 50MHz，同时，不仅支持片内高速缓存，即一级（L1）缓存，486 还支持主板上的二级（L2）缓存。尤其是 486 DX2 还首次引入了倍频的概念，有效地缓解了外部设备的制造工艺跟不上 CPU 主频发展速度的矛盾。AMD、Cyrix、TI、UMC 等厂商开始生产兼容 80386/80486 的芯片，并用低价手段来抢占市场，推动了技术的快速进步。

1993 年 Pentium 处理器面世。由于采用了一系列新的设计，尤其是将一些原本用于大型机的技术逐步引入微处理器中（如流水线技术、动态预测技术等），使得它的速度比 80486 快了几倍。此后，直到 2000 年末，Intel 又陆续推出 PentiumPro，Pentium MMX、PⅡ~PⅣ，不仅处理器的运算速度大幅提高，主频高达数 G 赫兹，而且增强了对浮点运算

和多媒体技术的支持,改进了插槽技术和生产工艺,使得这样的高档微处理器的功能已达到原中型机的功能。与此同时,AMD公司也推出了它的K系列微处理器,以较好的性价比占据中低端市场,取得了较好的业绩。

目前,Intel、AMD、Cyrix、IDT等x86系列CPU厂商之间的竞争还将长期持续下去,并为我们带来更多可供选择的微处理器产品,进一步推动微型计算机的快速发展。

1.1.2 微机的特点

微型计算机本质上与其它计算机并无太多的区别,所不同的是微型计算机广泛采用了集成度相当高的器件和部件,因此带来以下一系列特点。

① 体积小,重量轻,功能强 由于采用大规模集成电路(LSI)和超大规模集成电路(VLSI),微型机所含的器件数目大为减少,体积也大为缩小,20世纪70年代中小型计算机所能实现的功能,在当今内部只含几十片集成电路的微型机就能实现。

② 价格低 当前微型机的价格依据摩尔定律变化,即计算机芯片的价格每18个月降低一半。速度每18个月则提高一倍。很好的性价比使得微型机具有极高的市场占有率。

③ 可靠性高,结构灵活 由于采用了超大规模集成电路,使得微机内部元器件数目少;采用了总线结构、积木式设计,使得微型机的组合非常灵活,整机系统可靠性高。

④ 应用面广 现在微型机已完全取代了原来的中小型机,不仅应用于科学研究,而且广泛应用于工业生产、过程控制、公共信息、商业服务等各行各业,可以说,微型机已经渗透到社会生活的每一个角落,尤其是随着Internet的发展,计算机已成为现代社会的重要标志。

1.2 微机的组成结构与工作过程

1.2.1 微机的组成结构

(1) 微机的组成

微型计算机的基本组成结构如图1-1所示。从该结构图中可以看出,一台微型计算机由四部分组成:微处理器、存储器、输入输出接口和总线。它们以微处理器为核心,把存储器(ROM、RAM)、I/O接口电路通过总线有机地结合在一起形成微型计算机。在这里,CPU如同微型机的“心脏”,它的性能决定了整个微型机的各项关键指标。存储器包括随机存取存储器(RAM)和只读存储器(ROM)。输入输出接口电路用来使外部设备与微型机相连。总线为CPU和其它部件之间提供数据、地址和控制信息的传输通道。

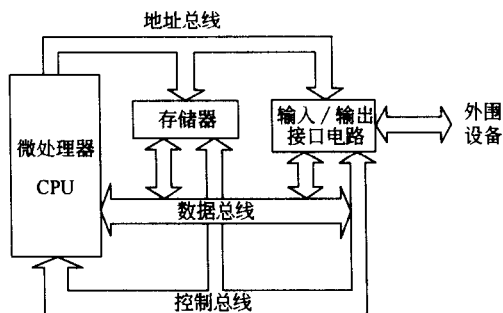


图 1-1 微型机的基本结构

(2) 微型机的总线结构

现代微型计算机都采用总线结构。在微型计算机系统中,无论是各部件之间的信息传送,还是处理器内部信息的传送,都是通过总线进行的。总线是连接多个功能部件或多个装置的一组公共信号线,它提供各功能部件或装置之间的信息传输通道。根据总线所处的位置不同,总线有不同的分类,如片内总线和片间总线等。片内总线是集成电路芯片内部各功能部件和各寄存器之间的连线;片间总线是连接各芯片的总线,即连接CPU、存储器和I/O接口的总线,又称为局部总线。微型计算机有了总线结构后,系统中各功能部件之间的相互

关系变为各个部件面向总线的单一关系。一个部件只要符合相同的总线标准,就可以直接连接到采用这种总线标准的系统中,使系统的功能得以发挥。总线结构为微机的结构扩展提供了极大的灵活性,因此,它也是现代计算机结构的重要特征。有关总线的详细内容见本书的第8章。以后提到的总线如不加说明,均指局部总线。

尽管各种微型机的总线类型和标准有所不同,但从总线所承担的任务来看,一般分为三种不同功能的总线:地址总线(Address Bus, AB)、数据总线(Data Bus, DB)和控制总线(Control Bus, CB)。

1) 地址总线

地址总线是微型计算机用来传送地址的信号线。地址总线的位数决定了CPU可以直接寻址的内存范围。微型机根据直接寻址能力决定地址线的根数,通常8位机的寻址范围为32K或64K($1K=1024$)字节,16位机能寻址的范围为 $1M(2^{20})\sim 16M(2^{24})$ 字节的能力。因为地址信号总是由CPU提供的,所以地址总线是单向三态总线。单向指信息只能向一个方向传送,三态指除了输出高电平和低电平外,还可以处于高阻状态(浮空状态)。

2) 数据总线

数据总线是CPU用来传送数据和代码的信号线。从结构上看,数据总线总是双向的,即数据既可以从CPU送到其它部件,也可以从其它部件传送给CPU。数据总线的位数(宽度)是微型计算机的一个很重要指标,它通常和处理器的位数相对应。例如16位微处理器,有16根数据线,32位微处理器有32根数据线。和其它计算机一样,在微型机中,数据的含义也是广义的,通过数据总线传送的除了数据以外,还可能有代码、状态量,有时还可能是控制量。数据总线也采用三态逻辑。

3) 控制总线

控制总线是用来传送控制信号的。这组信号线比较复杂,由它来实现CPU对外部部件(包括存储器和I/O接口)的控制。不同的微处理器采用不同的控制信号,通常来讲,控制信号用来实现CPU对外部部件的控制(如读写命令)、状态的传送(如应答联络信号)、中断、直接存储器存取(DMA)的控制,提供系统使用的时钟和复位信号等。

控制总线的信号线,根据使用条件不同,有的为单向,有的为双向或三态,还有的为非三态的信号线。控制总线是一组很重要的信号线,它决定了总线功能的强弱和适应性的好坏。

(3) 微处理器的结构与功能

微处理器的结构严格地受到大规模集成电路制造工艺的约束,因此芯片的面积不能过大,引出端的数量也受到了约束。上述条件就严格规定了通用微处理器的内部结构及其同外部设备的连接方式:外部一般采用上述的三总线结构,内部采用单总线,即内部所有单元电路都挂在内部总线上,分共享用。一个典型的8位微处理器的结构如图1-2所示。

它包括以下几个重要部分:累加器、算术逻辑运算单元(ALU)、状态标志寄存器、寄存器阵列、指令寄存器、指令译码器和定时及各种控制信号的产生电路。

1) 累加器和算术逻辑运算单元

累加器和算术逻辑运算单元主要用来完成数据的算术和逻辑运算。ALU有两个输入端和两个输出端。两个输入端一端接至累加器,接收由累加器送来的第一个操作数;另一端通过内部数据总线接到寄存器阵列,以接收第二个操作数。两个输出端一端接至数据总线,通过总线与累加器和寄存器阵列相联系;另一端接至标志寄存器。

参加运算的操作数先送到累加器和寄存器,然后在控制信号的控制下,在ALU中进行规定的运算操作,运算结束后,将结果送至累加器或寄存器,同时将操作结果的特征状态送标志寄存器。

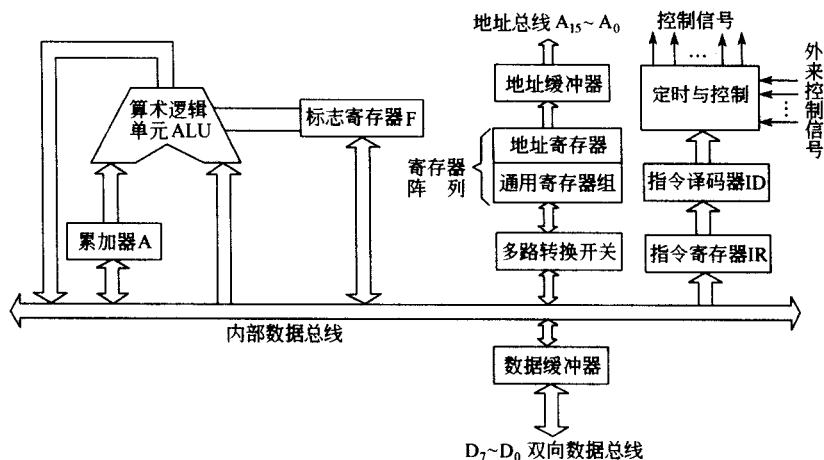


图 1-2 典型 8 位微处理器结构

累加器是一个特殊的寄存器，它的字长和微处理器的字长一样，例如：16 位微处理器的累加器字长为 16 位。累加器一般具有输入输出和移位功能，微处理器采用累加器结构可以简化某些逻辑运算。由于所有参加运算的数据都要通过累加器，故累加器在微处理器中占有重要的位置。

2) 寄存器阵列

通用寄存器组：用来寄存参与运算的数据（8 位），它们也可以连成 16 位的寄存器对，用以存放操作数的地址。

地址寄存器：用来存放地址。常用的地址寄存器有三种：指令指针 IP（有时也称为程序计数器 PC），变址寄存器 SI、DI，堆栈指针 SP。

指令指针 IP：它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。

变址寄存器 SI、DI：程序设计中往往要修改地址，变址寄存器的作用是用来存放要修改的地址，它也可以用来暂存数据。

堆栈指针 SP：用来指示 RAM 中堆栈栈顶的地址。堆栈中每压入或弹出 1 个数据，SP 的内容就自动减 1 或加 1，以指示新的栈顶地址。SP 的值始终指向栈的顶部。

3) 指令寄存器、指令译码器和定时及各种控制信号的产生电路

指令寄存器（Instruction Register, IR）用来存放当前正在执行的指令。当指令执行完毕后下一条指令才存入。指令译码器（Instruction Decoder, ID）用来对指令进行分析译码，根据指令译码器的输出信号，定时及控制信号产生电路产生出执行此条指令所需的全部控制信号，以控制各部件协调工作。

4) 内部总线和总线缓冲器

内部总线把 CPU 内各部件和 ALU 连接起来，以实现各部件之间的信息传送。内部总线分为内部数据总线和内部地址总线，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），具有驱动放大和隔离功能。

1.2.2 微机的工作过程

计算机之所以能在没有人干预的情况下自动地完成各种工作任务，是因为人们事先为它编制了完成这些任务所需的工作程序，并把程序存放到存储器中，这就是程序存储。计算机的工作过程就是执行程序的过程，控制器按照预先规定好的顺序，从存储器中一条一条地取

出指令、分析指令，根据不同的指令向各个部件发出完成该指令所规定操作的控制信号，这就是程序控制。程序存储和程序控制的概念是美籍匈牙利人约翰·冯·诺依曼提出来的，因此又称为冯·诺依曼概念。当代的计算机，不管是微型机还是大型机，甚至像 CRAY 等巨型机都是按冯·诺依曼模型工作的，故统称为冯·诺依曼计算机。

下面我们以图 1-3 所示的简化模型来分析微机的工作过程。假设要完成 $Y=10+20$ ，结果送 30 单元的操作。

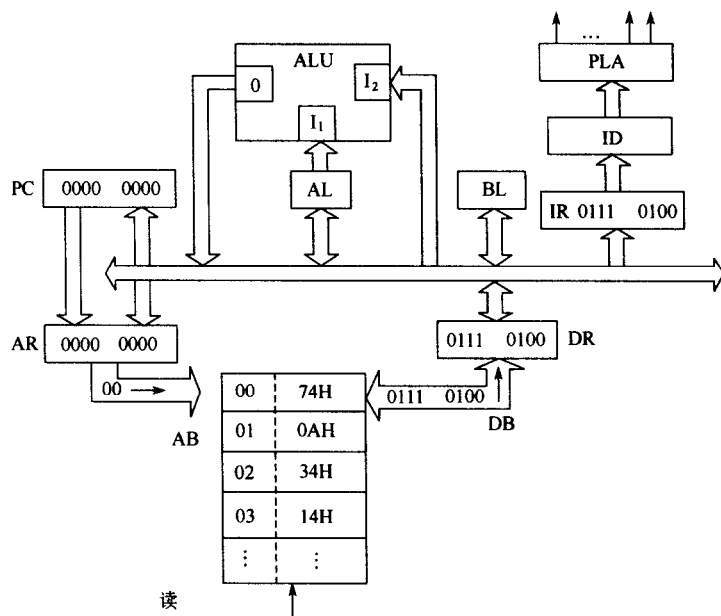


图 1-3 取指阶段执行示意

要完成上述功能，首先要查找指令表（如表 1-1 所示）找到相关指令，编写程序如下。

```
MOV    AL, 10    ; AL=10
ADD     AL, 20    ; AL=AL+20=10+20=30
MOV     [30], AL  ; AL 送 30 单元
HLT     ; 系统暂停
```

然后对上述程序进行汇编，翻译成机器码。翻译过程一般通过汇编程序 MASM 和 LINK 自动完成。另一种为手工汇编，在这里通过手工方法实现，查表 1-1 可将上述程序翻译为如下机器代码：

```
MOV     AL, 10    ; 01110100    00001010    740AH
ADD     AL, 20    ; 00110100    00010100    3414H
MOV     [30], AL  ; 01010011    00011110    531EH
HLT     ; 01000011    43H
```

表 1-1 指令表

功 能	助 记 符	机 器 码	说 明
立即数 n 送 AL	MOV AL,n	01110100 n	两字节指令
AL 内容加立即数 n	ADD AL,n	00110100 n	结果在 AL 中
AL 内容送 M 为地址的单元	MOV [M],AL	01010011 M	两字节指令
停止操作	HLT	01000011	一字节指令

将机器码从 00 单元开始存入内存中，如图 1-3 所示。最后按如下步骤执行程序。

数据寄存器 DR, CPU 知道这是取指阶段, 故把它送到指令译码器, 经译码后, CPU 识别出这是一条加法指令, 一个加数在累加器 AL, 一个加数是紧跟在操作码后面的操作数, 故发出执行这条指令的各种控制信号, 过程如下。

把 PC 的内容 03H 送 AR, PC 的内容自动加 1 变为 04H。AR 把地址码 03H 通过地址总线送至存储器, 选中 03 单元, 然后, CPU 内的控制电路发出存储器读命令, 通过数据总线把 03 单元的内容 14H 送至 DR, CPU 已经知道这是与累加器 AL 的内容相加的一个操作数, 故把它和累加器 AL 的内容 0AH 同时送运算单元 ALU, 由 ALU 完成 0AH+14H 的操作, 结果送到累加器 AL。至此, 第二条指令执行完毕, CPU 紧接着进行第三条指令的取指与译码, 过程如下。

把 PC 的内容 04H 送 AR, PC 的内容自动加 1 变为 05H。AR 把地址码 04H 通过地址总线送至存储器, 选中 04 单元, 然后, CPU 内的控制电路发出存储器读命令, 通过数据总线把 04 单元的内容 53H 送至 DR, 经译码后, CPU 识别出这是一条把 AL 中的内容写到存储器中的操作, 这个存储单元的地址就是紧跟在操作码后面的操作数, 故执行该指令的过程如下。

把 PC 的内容 05H 送 AR, PC 的内容变为 06H。AR 把地址码 05H 通过地址总线送至存储器, 选中 05 单元, 然后, CPU 内的控制电路发出存储器读命令, 通过数据总线把 05 单元的内容 1EH 送至 DR, CPU 已经知道这是存储单元的地址, 故把它送到 AR, AR 把地址码 1EH 通过地址总线送至存储器, 经存储器内部的地址译码器译码后, 选中 1E 单元, 然后, CPU 发出存储器写命令, 通过数据总线把 AL 中的内容写入 1E 单元。CPU 紧接着取最后一条指令, 译码后停止操作。至此程序执行完毕。

从以上分析可知, 微机工作的过程, 就是一个不断的取指令、指令译码、取操作数、执行运算、送运算结果的循环过程。

1.3 8086/8088 微处理器

8086/8088 微处理器是 Intel 公司推出的第三代微处理器芯片, 它们的内部结构基本相同, 但外部特性有所不同。8086 对外是 16 位数据线, 而 8088 对外是 8 位数据线, 因此, 在处理 16 位数时, 8088 需要两步操作, 而 8086 只需一步。8086/8088CPU 内部都采用 16 位结构进行操作, 它们都是 40 脚双列直插式封装, 对外有 20 根地址线, 可直接寻址的地址范围为 2^{20} , 即 1M 字节。

1.3.1 8086/8088CPU 的编程结构

(1) 8086/8088CPU 的编程结构

编程结构也称为功能结构, 是从程序员的角度来看的处理器结构。8086/8088CPU 的编程结构分为两部分: 总线接口部件 BIU (Bus Interface Unit) 和执行部件 EU (Execution Unit)。基本结构如图 1-5 所示。

1) 执行部件

EU 单元负责指令的执行。它包括 ALU (运算器)、通用寄存器和状态寄存器等, 主要进行 16 位的各种算术运算及逻辑运算。

2) 总线接口部件

BIU 单元负责与存储器和 I/O 接口之间传送数据。它由段寄存器、指令指针、地址加法器和指令队列缓冲器组成。地址加法器将段地址和偏移地址相加, 生成 20 位的物理地址。

8086/8088 的 BIU 有如下特点。

① 8086 的指令队列为 6 个字节, 8088 的指令队列为 4 个字节。这样, 在 EU 执行指令的同时, BIU 可从内存中取下一条指令或下几条指令放在指令队列中。而 EU 执行完一条指

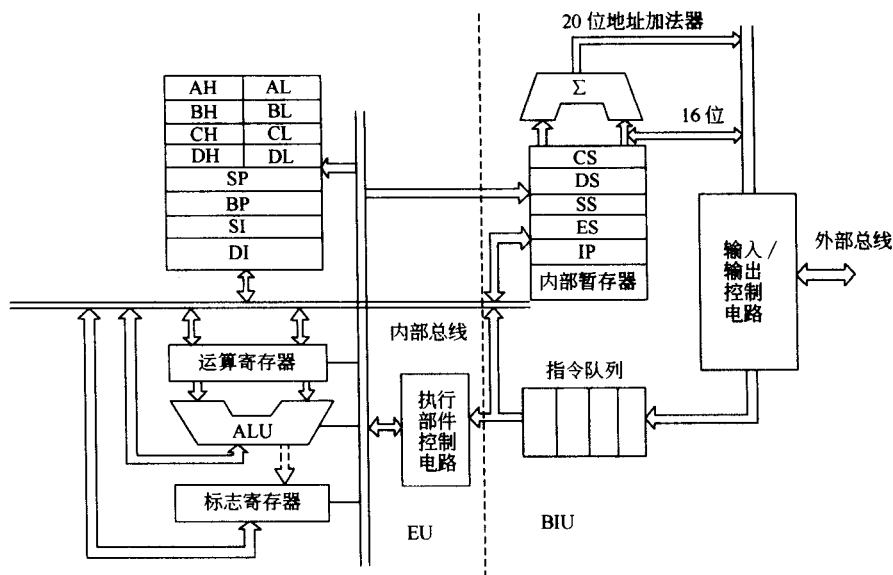


图 1-5 8086/8088CPU 内部功能结构图

令后就可以直接从指令队列中取出下一条指令执行，从而提高了 CPU 的效率。

② 地址加法器用来产生 20 位地址。上面已经提到，8086/8088 可用 20 位地址寻址 1M 字节的内存空间，但 8086/8088 内部所有的寄存器都是 16 位的，所以需要由一个附加的机构来根据 16 位寄存器提供的信息计算出 20 位的物理地址，这个机构就是 20 位的地址加法器。

总线接口部件和执行部件并不是同步工作的，它们按以下流水线技术原则来协调管理。

① 每当 8086 的指令队列中有两个空字节，或者 8088 的指令队列中有一个空字节时，总线接口部件就会自动把指令取到指令队列中。

② 每当执行部件准备执行一条指令时，它会从总线接口部件的指令队列前部取出指令的代码，然后用几个时钟周期去执行指令。在执行指令的过程中，如果必须访问存储器或者输入/输出设备，那么，执行部件就会请求总线接口部件进入总线周期，完成访问内存或者输入/输出端口的操作；如果此时总线接口部件正好处于空闲状态，那么，会立即响应执行部件的总线请求。但有时会遇到这样的情况，执行部件请求总线接口部件访问总线时，总线接口部件正在将某个指令字节取到指令队列中，此时总线接口部件将首先完成这个取指令的操作，然后再去响应执行部件发出的访问总线的请求。

③ 当指令队列已满，而且执行部件又没有总线访问请求时，总线接口部件便进入空闲状态。

④ 在执行转移指令、调用指令和返回指令时，由于程序执行的顺序发生了改变，不再是顺序执行下面一条指令，这时，指令队列中已经按顺序装入的字节就没用了。遇到这种情况，指令队列中的原有内容将被自动消除，总线接口部件会按转移位置往指令队列装入另一个程序段中的指令。

(2) 8086/8088CPU 的内部寄存器

在 8086/8088 微处理器中，有许多不同用途的内部寄存器，这些寄存器在微处理器的工作过程中起着非常重要的作用，也是用汇编语言编程必须用到的。8086/8088 微处理器的内部寄存器结构如图 1-5 所示。