

凌志浩 © 主编

微机原理

YU JIEKOU JISHU

与

接口技术



华东理工大学出版社

EAST CHINA UNIVERSITY OF SCIENCE AND TECHNOLOGY PRESS

微机原理与接口技术

凌志浩 主编

 华东理工大学出版社

图书在版编目(CIP)数据

微机原理与接口技术/凌志浩主编. —上海:华东理工大学出版社,2006.8
ISBN 7-5628-1927-0

I. 微... II. 凌... III. ①微型计算机-理论②微型计算机-接口 IV. TP36

中国版本图书馆 CIP 数据核字(2006)第 072258 号

微机原理与接口技术

.....

主 编 / 凌志浩

责任编辑 / 徐惠娟

封面设计 / 赵 军

责任校对 / 张 波

出版发行 / 华东理工大学出版社

地 址:上海市梅陇路 130 号,200237

电 话:(021)64250306(营销部)

传 真:(021)64252707

网 址:www.hdlgpress.com.cn

印 刷 / 上海崇明裕安印刷厂

开 本 / 787×1092 1/16

印 张 / 18

字 数 / 438 千字

版 次 / 2006 年 8 月第 1 版

印 次 / 2006 年 8 月第 1 次

印 数 / 1—4050 册

书 号 / ISBN 7-5628-1927-0/TP·147

定 价 / 28.00 元

内 容 提 要

本书以 Intel 8086 微处理器为主体,阐述 16 位微型计算机的基本原理、汇编语言和接口技术等内容。主要内容涵盖计算机运算基础、微型计算机的基本系统、微处理器的内部结构和外部特性、基本配置和总线时序、指令系统和汇编语言程序设计、存储器系统、输入/输出方式和中断、总线与总线接口、典型接口芯片及其相关控制接口技术(包括中断控制、DMA 控制、定时计数控制、并行接口、串行通信接口以及模拟接口等),并结合系统组成原理和应用实例讨论各种接口芯片的应用方法。全书内容丰富,图文并茂,通俗易懂,并附有习题和思考题,既可用作教材,也适合于自学。

本书可作为高等学校微机原理与接口技术等相关课程的教材或参考用书,适合于计算机、电子工程、自动化、仪器仪表、机械等相关学科的本、专科学生、高职学生、成教学生、网络学院学生阅读;也可作为计算机应用开发人员和希望深入学习微机技术的工程技术人员的参考书。

前 言

当今,微电子技术和计算机技术飞速发展,微机应用已深入到各个领域,促进了各个专业领域的技术进步和发展。作为计算机应用的基础——微机原理与接口技术已成为工科学生共同的基础知识。为适应高等工程教育和面向大众化教育的需要,编者根据多年来的教学经验,结合微机发展和应用需求,编写了本教材。

本书的编写贯彻“软硬结合、面向应用”的方针,力求将微型计算机的硬件组成原理、软件设计方法以及接口技术与实际应用相结合。全书共分11章,内容安排上力求体现“以基本原理出发,注重指导实际应用”的原则,在结构上考虑了各章节的衔接性、系统性和完整性,选材注重先进性、代表性和实用性,并尽可能做到深入浅出、循序渐进、便于自学。1~5章是基础部分,不仅对微机发展、微机系统的基本结构、影响微机发展的若干新技术、计算机运算基础等方面的内容作了介绍,而且重点介绍了Intel 8086的特性、系统组成模式、指令系统和汇编语言程序设计方法、存储器系统等内容。6~10章接口技术和应用技术部分,着重介绍目前常用的接口芯片,讨论接口技术和应用方法,其中包括数据的传输方式、中断技术、DMA技术、定时计数器、并行接口和串行接口、数/模和模/数转换技术等,并结合实例分析接口技术的应用方法。第11章介绍了微型计算机的总线和总线接口,分析了常用总线的特性和所能支持的应用。

本书由凌志浩主编,张建正编写了第7,8,10章,庞毅林编写了第3,4,9章,朱尚明编写了第5,11章,蒋翠玲编写了第6章,凌志浩编写了第1,2章。全书由凌志浩统稿。在此,编者感谢书末所列参考文献中的所有作者。

由于编者的水平和教学经验所限,教材中难免会有错漏和不当之处,敬请读者批评指正。

编 者

2006年5月于华东理工大学

目 录

1 绪论	(1)
1.1 微型计算机概述	(1)
1.2 微型计算机的组成和指令流程	(3)
1.2.1 微型计算机的硬件组成	(3)
1.2.2 微型计算机的指令系统	(5)
1.2.3 微型计算机的指令流程	(6)
1.3 计算机中数的表示方法	(7)
1.3.1 计算机中数的表示方法	(7)
1.3.2 计算机中的编码	(11)
1.3.3 计算机中的运算	(13)
习题与思考题一	(22)
2 8086 系统结构	(24)
2.1 8086 CPU 结构	(24)
2.1.1 8086 CPU 的内部结构	(25)
2.1.2 寄存器结构	(27)
2.2 8086 CPU 的引脚功能	(30)
2.2.1 8086 CPU 在最小模式中的引脚定义	(31)
2.2.2 8086 CPU 在最大模式中的引脚定义	(34)
2.3 8086 存储器组织	(35)
2.3.1 存储器地址的分段	(35)
2.3.2 8086 存储器的分体结构	(36)
2.3.3 堆栈的概念	(38)
2.4 8086 系统配置	(40)
2.4.1 最小模式系统	(40)
2.4.2 最大模式系统	(43)
2.5 8086 CPU 时序	(48)
2.5.1 系统的复位和启动	(48)
2.5.2 最小模式下的总线操作	(49)
2.5.3 最小模式下的总线保持	(52)
2.6 Intel 80386 微处理器	(53)

2.7 Pentium 系列微处理器	(54)
2.7.1 Pentium 微处理器	(54)
2.7.2 Pentium 4 微处理器	(55)
2.7.3 若干新技术	(55)
习题与思考题二	(59)
3 寻址方式和指令系统	(60)
3.1 Intel x86 的寻址方式	(60)
3.1.1 立即寻址	(62)
3.1.2 寄存器寻址	(62)
3.1.3 直接寻址	(63)
3.1.4 寄存器间接寻址	(64)
3.1.5 变址寻址	(64)
3.1.6 基址加变址寻址	(65)
3.2 Intel x86 指令系统	(65)
3.2.1 数据传送指令	(66)
3.2.2 算术运算指令	(72)
3.2.3 逻辑运算指令	(77)
3.2.4 串操作指令	(80)
3.2.5 控制转移类指令	(84)
3.2.6 移位指令	(86)
3.2.7 标志处理与处理器控制指令	(87)
习题与思考题三	(87)
4 汇编语言程序设计	(91)
4.1 汇编语言的基本元素	(91)
4.1.1 汇编语言的语句格式	(91)
4.1.2 程序结构	(92)
4.1.3 汇编语言的运算符	(93)
4.1.4 汇编语言程序汇编步骤	(96)
4.2 伪指令	(96)
4.2.1 数据定义语句	(96)
4.2.2 符号定义 EQU、= 及 PURGE	(99)
4.2.3 段定义语句	(99)
4.2.4 过程定义伪指令 PROC 和 ENDP	(102)
4.2.5 宏指令	(103)
4.3 汇编程序设计	(105)
4.3.1 顺序程序设计	(105)
4.3.2 分支程序设计	(105)

4.3.3 循环程序设计	(108)
4.3.4 子程序设计	(114)
4.3.5 MASM 与高级语言的接口	(118)
4.3.6 DOS 功能调用	(119)
习题与思考题四	(121)
5 存储器系统	(127)
5.1 存储器概述	(127)
5.1.1 计算机的多级存储管理	(127)
5.1.2 半导体存储器	(128)
5.2 随机存取存储器(RAM)	(130)
5.2.1 静态随机存取存储器(SRAM)	(130)
5.2.2 动态随机存取存储器(DRAM)	(134)
5.2.3 新型 DRAM 存储器	(138)
5.3 只读存储器(ROM)	(139)
5.3.1 掩膜型只读存储器(MROM)	(139)
5.3.2 可编程只读存储器(PROM)	(140)
5.3.3 可擦除可编程只读存储器(EPROM)	(140)
5.3.4 电擦除可编程只读存储器(EEPROM)	(142)
5.3.5 闪速存储器	(144)
5.4 存储器芯片与 CPU 的连接	(146)
5.4.1 位扩展	(147)
5.4.2 字扩展	(149)
5.4.3 字、位同时扩展	(150)
习题与思考题五	(153)
6 输入/输出接口与中断技术	(154)
6.1 输入/输出接口概述	(154)
6.1.1 I/O 接口的基本结构	(154)
6.1.2 I/O 接口的基本功能	(155)
6.1.3 I/O 接口的编址方式	(155)
6.2 输入/输出数据的传输控制方式	(156)
6.2.1 无条件传送方式	(156)
6.2.2 程序查询传送方式	(158)
6.2.3 中断传送方式	(161)
6.2.4 DMA 传送方式	(162)
6.3 Intel x86 微处理器实模式下的中断操作	(163)
6.3.1 中断的基本概念	(163)
6.3.2 中断过程	(163)

6.3.3	中断分类与中断类型	(166)
6.3.4	中断向量与中断向量表	(168)
6.3.5	中断响应过程与时序	(170)
6.4	中断控制器 8259A	(170)
6.4.1	8259A 的结构及主要功能	(170)
6.4.2	8259A 的工作方式	(173)
6.4.3	8259A 的工作过程	(177)
6.4.4	8259A 的控制命令字及编程	(178)
6.4.5	8259A 在系统中的应用举例	(183)
	习题与思考题六	(185)
7	DMA 控制器和定时器/计数器	(187)
7.1	DMA 控制器 8237	(187)
7.1.1	概述	(187)
7.1.2	8237 的结构	(188)
7.1.3	8237 的内部寄存器	(190)
7.1.4	8237 的软命令	(193)
7.1.5	8237 的工作时序	(194)
7.1.6	8237 的编程和举例	(195)
7.2	可编程定时器/计数器 8253	(196)
7.2.1	8253 的结构	(197)
7.2.2	8253 的工作方式	(199)
7.2.3	8253 的编程	(202)
7.2.4	8253 的应用	(205)
	习题与思考题七	(207)
8	并行通信	(209)
8.1	并行接口概述	(209)
8.2	可编程并行接口芯片 8255A	(209)
8.2.1	8255A 的结构	(210)
8.2.2	8255A 的工作方式	(211)
8.2.3	8255A 编程	(215)
8.2.4	8255A 应用	(217)
	习题与思考题八	(219)
9	串行接口与通信	(221)
9.1	串行通信概述	(221)
9.2	串行通信传输方式	(222)
9.2.1	串行通信的传输方向	(222)

9.2.2 信号的调制与解调	(222)
9.2.3 串行通信的错误校验	(224)
9.3 传输速率与波特率	(225)
9.4 串行通信协议	(225)
9.4.1 异步通信	(225)
9.4.2 同步通信	(227)
9.5 RS-232 标准	(229)
9.5.1 机械特性	(229)
9.5.2 功能特性	(229)
9.5.3 常用连接方式	(230)
9.5.4 电平特性	(231)
9.6 可编程串行异步通信接口 8250	(231)
9.6.1 8250 芯片功能	(231)
9.6.2 8250 芯片引脚	(232)
9.6.3 8250 内部结构及寄存器	(233)
9.6.4 8250 初始化编程	(237)
9.6.5 串行通信程序设计	(238)
9.7 通用串行接口标准	(241)
9.7.1 通用串行接口 USB	(241)
9.7.2 USB 总线拓扑结构	(241)
9.7.3 USB 连接接口	(242)
9.7.4 USB 设备逻辑结构	(243)
9.7.5 USB 数据流模型	(243)
9.7.6 USB 传输类型	(244)
9.7.7 USB 低层通信协议	(244)
9.7.8 USB 设备的枚举过程	(245)
习题与思考题九	(246)
10 D/A 与 A/D 转换及接口	(247)
10.1 概述	(247)
10.2 D/A 转换器接口	(247)
10.2.1 D/A 转换器的工作原理	(247)
10.2.2 D/A 转换器的主要技术指标	(249)
10.2.3 D/A 转换器的芯片	(249)
10.2.4 D/A 转换器与微处理器接口要考虑的问题	(251)
10.2.5 D/A 转换器与微处理器的接口	(252)
10.3 A/D 转换器接口	(253)
10.3.1 采样与量化	(253)
10.3.2 A/D 转换器的工作原理	(254)

10.3.3	A/D 转换器的主要技术指标	(256)
10.3.4	A/D 转换器的芯片	(256)
10.3.5	A/D 转换器与微处理器接口要考虑的问题	(258)
10.3.6	A/D 转换器与微处理器的接口	(259)
习题与思考题十		(261)
11	微型计算机总线及总线接口	(262)
11.1	总线技术概述	(262)
11.1.1	总线的特性	(262)
11.1.2	总线的分类	(262)
11.1.3	总线的指标	(263)
11.1.4	总线的数据传输	(264)
11.2	PC 总线和 ISA 总线	(266)
11.2.1	PC 总线	(266)
11.2.2	ISA 总线	(266)
11.3	PCI 总线	(269)
11.3.1	PCI 的特点	(269)
11.3.2	PCI 总线信号	(270)
11.3.3	PCI 总线系统结构	(272)
11.3.4	PCI 总线周期和地址空间	(273)
11.4	USB 总线	(274)
11.4.1	USB 概述	(274)
11.4.2	USB 总线的构成	(275)
11.4.3	USB 数据传输	(276)
11.4.4	USB 设备接入和开发	(277)
参考文献		(278)



1

绪 论

1.1 微型计算机概述

从 1946 年诞生第一台电子计算机以来,计算机技术得到了突飞猛进的发展,经历了电子管计算机、晶体管计算机、集成电路计算机、大规模及超大规模集成电路计算机等四代。在此发展过程中,计算机的性能不断提高,性能价格比越来越高,应用领域越来越广。

自 1971 年美国 Intel 公司研制成功 4004 微处理器以来,随着大规模及超大规模集成电路的不断发展,微处理器的发展是日新月异,基于微处理器的微型计算机以其惊人的速度发展,几乎每两年微处理器的集成度和性能提高一倍,每 3~4 年微机就要更新换代一次。

微型计算机的核心是微处理器(Microprocessor),又称中央处理器或中央处理单元(Central Processing Unit, CPU)。

下面对微处理器的发展情况作一简要回顾和介绍。

第一代(1971~1973 年) 4 位和低档 8 位微处理器。

1971 年,Intel 的 4 位微处理器开创了微型计算机时代,1972 年 Intel 又推出了 8 位微处理器 8008。它们的特点是采用 PMOS 工艺,速度低,时钟频率 1 MHz,指令系统简单,运算功能较差。

第二代(1973~1977 年) 中高档 8 位微处理器。

代表产品有 Intel 公司的 8080/8085、Motorola 公司的 MC6800/6802、Zilog 公司的 Z80 和 Rockwell 公司的 6502 等。其特点是采用 NMOS 工艺,运算速度是第一代的 10~15 倍,指令系统比较完善,寻址能力有所增强,已有典型的计算机体系结构以及中断和 DMA 功能,支持语言有汇编、BASIC、FORTRAN 和 PL/M 等,后期开始配备 CP/M 操作系统。

第三代(1978~1981 年) 16 位微处理器。

随着超大规模集成电路(VLSI)的发展,1978 年 Intel 公司推出了 16 位微处理器 8086,片内集成了 29 000 个晶体管。Zilog 公司和 Motorola 公司也分别先后推出 16 位的 Z8000 和 MC68000。1979 年 Intel 公司推出了内部 16 位结构、外部数据总线 8 位的微处理器 8088。特点是采用高密度的 HMOS 工艺,时钟频率 4~8 MHz,平均指令执行时间 0.5 μ s,集成度达 2~6 万晶体管/片;1982 年 Intel 公司推出的 80286,其集成度已达 13 万晶体

管/片。

1981年,IBM公司采用Intel 8088微处理器,生产了第一台通用微型计算机IBM PC,从此IBM PC系列微机成为个人计算机的主流机之一。

第四代(1982~1992年) 32位微处理器。

1985年,Intel公司推出了第一个32位微处理器80386,片内集成了27.5万个晶体管。它与8086向上兼容,具有32位数据线和32位地址线,内部寄存器均为32位,最大地址空间为4GB,支持64TB的虚拟存储空间。它提供有三种工作模式:实地址模式、虚地址模式和虚拟8086模式。80386也是第一个支持片外Cache的CPU。时钟频率为32MHz。

1989年,Intel公司推出了更高性能的32位微处理器80486,片内集成了120万个晶体管,集成了一个80386微处理器、80387浮点运算处理器、8KB的高速缓冲存储器,极大地提高了存储器访问速度。80486的整数处理部件采用了RISC技术,减少了指令的执行时间,其他部分仍然保持CISC结构。

同期的32位微处理器还有Motorola公司的MC68030和MC68040。

第五代(1993~1995年) 奔腾微处理器。

1993年,Intel公司推出了全新一代的微处理器Pentium(奔腾),片内集成了310万个晶体管。Pentium具有64条数据线,32条地址线,时钟频率达120MHz以上。内部集成了8KB的代码Cache和8KB的数据Cache。采用了超标量流水线和指令分支预测技术,集成了高性能的浮点处理单元。它提供有四种工作模式:实地址模式、虚地址模式、虚拟8086模式和系统管理模式。

1996年,Intel公司推出了Pentium MMX(MultiMedia eXecution),在Pentium的基础上增加了57条MMX(多媒体扩展指令集)指令,采用了SIMD(单指令流多数据流)技术,用于音频、视频、图形/图像数据处理,使多媒体和通信处理能力得到了很大提高。

同期推出的微处理器还有IBM、Apple和Motorola三家联盟的PowerPC(这是一款RISC微处理器)以及AMD公司的K5和Cyrix公司的M1。

第六代(1995~1999年) 加强型Pentium微处理器。

1995年,Intel公司推出了Pentium Pro(高能奔腾),片内集成了550万个晶体管。具有64条数据线、36条地址线,时钟频率可达150MHz、166MHz和200MHz。内部集成了16KB的L1 Cache和256/512KB的L2 Cache。Pentium Pro具有3个整数执行单元和1个浮点单元,可以同时执行3条整数型指令。并通过指令乱序执行以及重排序缓冲器,进一步提高指令级的并行性。

1997年5月和1999年2月,Intel公司先后发布了Pentium II(PⅡ)和Pentium III(PⅢ)微处理器,Pentium II集成了750万个晶体管,内部L1级指令Cache和数据Cache分别增加到16MB,内部L2级Cache仍为256/512KB,增加了MMX技术。开始采用了双独立总线结构,一条用于连接处理器内核和L2 Cache,一条连接系统总线。Pentium III在Pentium II的基础上进一步提高了性能,集成了950万个晶体管,时钟频率为500MHz,而2000年推出的PⅢ其时钟频率高达1GHz,并增加了128位的SIMD(单指令多数据)寄存器和72条指令,用于互联网流式SIMD扩展SSE(internet streaming SIMD extensions),内部L1级Cache达到32MB,L2级Cache为512KB。

类似的微处理器还有AMD公司的K6。

第六代后(2000 年至今) Pentium 4 微处理器。

2000 年末, Intel 公司推出了 Pentium 4 微处理器, 片内集成了 4 200 万个晶体管, 采用了超级流水线技术和快速执行引擎, 增强了浮点和多媒体单元。L1 级 Cache 还增加了 12 K 的执行跟踪 Cache, 系统总线速度达到 400 MHz。Pentium 4 增加了由 144 条新指令组成的 SSE2, 提供 128 位 SIMD 整数算法操作和双精度浮点操作, 时钟频率达 1.3 GHz, 现在的 Pentium 4 其主频已高达 3.8 GHz。

同期, AMD 公司推出了微处理器 Athlon。

1.2 微型计算机的组成和指令流程

本节将首先介绍传统计算机和微型计算机的基本组成, 然后简述微型计算机的工作过程。

1.2.1 微型计算机的硬件组成

计算机硬件一般由运算器、控制器、存储器、输入设备和输出设备等 5 个部分组成, 如图 1-1 所示。

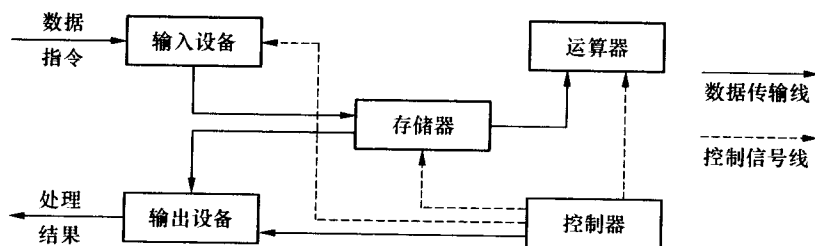


图 1-1 传统的冯·诺伊曼计算机硬件结构

在采用大规模集成电路的微型机中, 运算器通常与控制器合并为中央处理单元(CPU), 制作在一块微处理器芯片上。因此, 微型计算机硬件一般可划分为中央处理单元、存储器、输入输出设备、输入输出接口和总线等部分, 如图 1-2 所示。

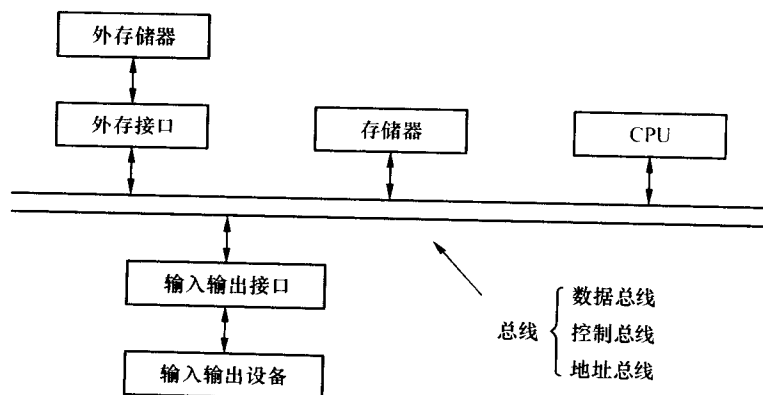


图 1-2 微机系统的硬件结构

1. 中央处理单元

中央处理单元(Central Processing Unit)简称 CPU。它是微型计算机的核心部分,主要包括运算器和控制器。

运算器(Arithmetic Logic Unit) 运算器是计算机中进行算术运算、逻辑运算的部件,故有时也称为算术逻辑运算单元(简称 ALU)。其核心是一个全加器。典型的运算器能够实现以下几种运算功能:两数相加,两数相减,把一个数左移或右移一位,比较两个数的大小,将两数进行逻辑“与”、“或”、“异或”运算,逻辑“非”运算等。必须指出,在早期的微处理器中并没有进行乘、除运算和浮点运算的硬件电路,运算器只能完成定点加、减运算,由于减法运算可通过二进制补码的加法运算来实现,因此准确地说它只能完成加法的运算,而复杂的算术运算(如乘、除运算)则由程序来完成。

控制器(Control Unit) 控制器是用来控制计算机进行运算及指挥各个部件协调工作的部件,主要由指令部件(包括指令寄存器和指令译码器)、时序部件和操作控制部件等构成。它根据指令的内容产生和发出控制计算机操作的信号,从而把微型计算机的各个部分组成一体,执行指令所规定的一系列有序的操作。

2. 存储器

微型计算机通常把半导体存储器用作内存储器或主存储器,磁盘、磁带、光盘等用作外存储器或辅助存储器。存储器好像一座大楼,大楼的每间房间称为存储单元,每个存储单元有一个唯一的地址(好比房间号),存储单元中的内容可以为数据或指令。在微机中,通常让每个存储单元存放一个字节,以保证随时可对任意一个字节进行访问。

3. 输入/输出设备

输入设备的作用是从外界将数据、指令等输入到微机的内存;输出设备的作用是将微机处理后的结果信息转换为外界能够使用的数字、文字、图形、声音等。微机外部设备的种类和形式很多,常见的输入设备有键盘、鼠标、模/数转换器、软、硬盘驱动器、光盘驱动器等。近年来语音、图像等输入设备已正式进入实用阶段。常见的输出设备有打印机、绘图仪、数/模转换器、显示终端、音响设备等。

4. 输入/输出接口

外部设备由于结构不同,各有不同的特性,而且它们的工作速度比微型机的运算速度低得多。为使微型机与外部设备能够协调工作,必须由适当的接口来完成协调工作。目前很多接口逻辑电路也采用大规模集成电路,并且已系列化、标准化。很多接口芯片具有可编程能力,并有很好的灵活性。这些接口芯片又可分为通用接口和专用接口。它们的主要任务和功能是:完成外部设备与计算机的连接、转换数据传送速率、转换电平、转换数据格式等。

5. 总线

将微处理器、存储器和输入输出接口等装置或功能部件连接起来,并传送信息(信号)的公共通道称为总线(Bus)。总线实际上是一组传输信息的导线,其中又包括:

数据总线(Data Bus) 数据总线是双向的通信总线。通过它可以实现微处理器、存储器和输入/输出接口三者之间的数据交换。例如,它可以将微处理器输出的数据传送到存储器或输入/输出接口,又可以把从存储器中取出的信息或从外设接口取来的信息传送到微处理器内部去。

地址总线(Address Bus) 地址总线是单向总线,用来从 CPU 单向地向存储器或 I/O 接

口传送地址信息。

控制总线(Control Bus) 控制总线传输的信号可以控制微型计算机各个部件有条不紊地动作,其中包括由微处理器向其他部件发出的读、写等控制信号,也包括由其他部件输入到微处理器中的信号。控制总线的多少因不同性能的微处理器而异。

按照总线的所在位置,又可区分为片内总线和系统总线。前者制作在 CPU 芯片中,是运算器与各种通用寄存器的连接通道;后者则制作在微机主板上,承担 CPU 与主存储器及外部设备接口的连接。

1.2.2 微型计算机的指令系统

微型机严格按照人们下达的命令去完成指定的任务,这些命令就是机器指令。机器指令随微型机所使用的微处理器而不同。某种微型机所能识别和执行的全部指令即称为该机的指令系统。

由于微机的硬件仅可识别二进制信息,因此机器指令也要用二进制数来表示。每一条指令执行一种简单的特定操作,如取数、相加、比较、判断、转移等。大多数需要微型机完成的任务可分解成一组步骤,用一连串指令去实现。这类为特定目的而组织起来的指令序列称作程序,而编制程序的工作则称为程序设计。

机器指令又称为机器语言。它虽然为计算机所“乐意”接受,但对用户却十分不便。例如进行一个“8+5”的加法运算,用 Intel 8086 的机器语言需写成:

```
10110000    00001000
10110011    00000101
00000000    11011000
```

若改用简便的十六进制数表示,也可以写为:

```
B0 08 B3 05 00 D8
```

从这一例子可以看到,用机器语言编写的程序难读、难记、难检查、难修改。为了更方便地使用微型机,人们创造了用缩写的英语单词来表示指令操作的方法,这些单词称作“助记符”。采用助记符表示的指令称作汇编指令。在采用汇编指令时,还为之规定了严格的语法规则,构成了汇编语言。上例用汇编语言可写成:

```
MOV AL, 08H    ; 将数 08 取至低 8 位累加寄存器 AL 中
MOV BL, 05H    ; 将数 05 取至低 8 位通用寄存器 BL 中
ADD AL, BL     ; 把 AL 和 BL 中的内容相加
```

其中“MOV”是数据传送的意思,“ADD”是两数相加的意思。当然微型机的硬件也不懂得汇编语言,因此在使用较低档的微型机(如单板机、单片机)时,往往要用人工查表的方法,把汇编语言指令逐条翻译为用十六进制表示的机器语言形式才能送入机器中,然后由机器自动把它转换为二进制形式再予以执行。当然,在使用高档的微型机时,通常在机器内配有翻译软件——汇编程序,它能将汇编语言自动翻译为机器语言。

目前,绝大多数微机用户均使用高级语言,但机器语言和汇编语言也有其独特的优势。之所以在本节中提及机器指令和汇编指令,主要是因为:①汇编语言在微机应用中仍然占有一席之地,这主要是因为汇编语言程序可以在最简单的硬件和最少的软件支持下

运行,而运行高级语言则需使用至少拥有键盘和屏幕显示器的微型机并配备翻译软件和较大的存储空间;②通过机器指令和汇编指令,较易说明计算机的工作过程,从而帮助读者对计算机的内部工作获得更多的理解;③汇编语言程序的时空效率高,即执行速度快,所占存储单元少。

1.2.3 微型计算机的指令流程

如前所述,当用微型机来完成某项任务时,首先要按照此任务之要求,编写出适合于机器工作的全部操作步骤,即程序。程序是一串有序指令的集合。把编好的程序(即一条条指令)连续地由输入设备通过 I/O 接口存放到存储器中,然后启动程序运行,计算机便能按程序的逻辑顺序一条一条地执行这些指令。那么,微型机是怎样执行一条条指令的呢? 下面仅结合图 1-3 简要说明某假想微型机在执行程序中第 n 条指令时的典型工作过程。

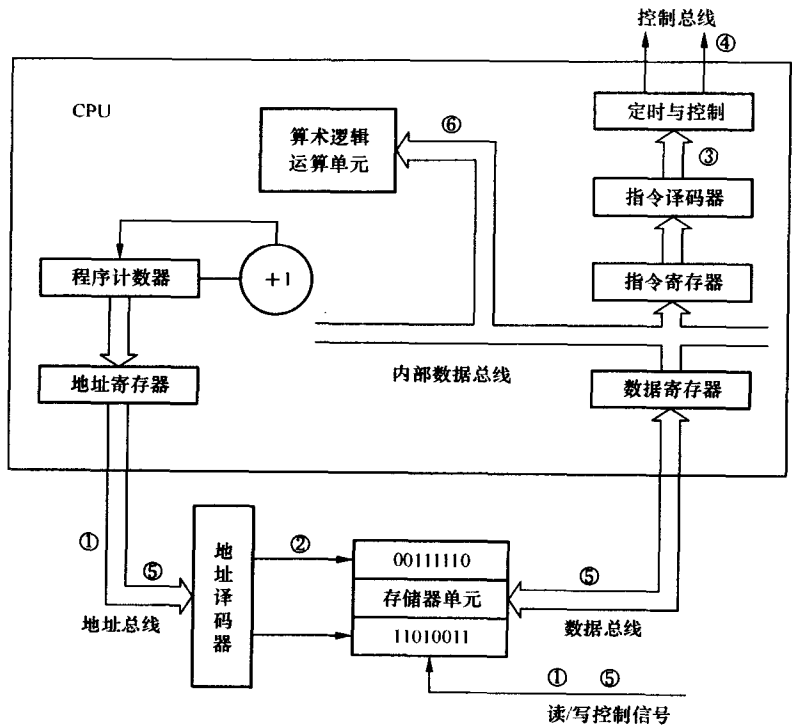


图 1-3 指令执行过程示意图

- ① CPU 通过地址总线(AB)指出指令所在内存单元的地址,同时通过控制总线(CB)向存储器发出准备读出数据的控制信号。
- ② 存储器中这一单元被地址线上的地址码选中,于是 CPU 即通过数据总线(DB)从存储器中读取存放在这一单元中的指令。
- ③ 指令是以二进制代码的形式存放在存储器中的,CPU 取出这一指令代码后在内部进行译码,判断出该指令是要进行哪一类操作以及参加这类操作的数放在什么单元地址。
- ④ CPU 根据对指令的译码结果,由控制器有序地发出为完成此指令所需要的各种控制