

21
Century

21世纪高等院校电气信息类系列教材

Electrical Information •
Science and Technology

单片机原理及应用

Artificial Intelligence •

张仁彦 高正中 黄鹤松 等编著



附赠电子教案

[http:// www.cmpedu.com](http://www.cmpedu.com)



机械工业出版社
CHINA MACHINE PRESS

21 世纪高等院校电气信息类系列教材

单片机原理及应用

张仁彦 高正中 黄鹤松 等编著

机械工业出版社

本书全面系统地介绍了 MCS-51 系列单片机的基本结构、工作原理及应用技术。主要内容包括：微型计算机的基础知识、MCS-51 单片机的基本结构、汇编语言和 C51 语言程序设计方法、中断系统和定时器等内部功能单元的应用技术、并行接口和串行接口的扩展方法，以及 Keil 软件和 Proteus 软件联合调试的方法等。书中大部分例题具有汇编语言和 C51 语言的双语对照程序，便于读者选择学习。

本书内容丰富、实用性强，讲解深入浅出、全面细致，既可以作为高等院校电气信息类相关专业学生的教材，也可以作为从事单片机应用系统研发工作的工程技术人员的参考书。

图书在版编目 (CIP) 数据

单片机原理及应用/张仁彦等编著. —北京: 机械工业出版社, 2016.2
21 世纪高等院校电气信息类系列教材

ISBN 978-7-111-52840-1

I. ①单… II. ①张… III. ①单片微型计算机-高等学校-教材
IV. ①TP368.1

中国版本图书馆 CIP 数据核字 (2016) 第 008004 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

策划编辑: 时 静

责任编辑: 时 静 王 荣 责任校对: 张艳霞

责任印制: 李 洋

北京振兴源印务有限公司印刷

2016 年 3 月第 1 版·第 1 次印刷

184mm×260mm·21.75 印张·538 千字

0001—3000 册

标准书号: ISBN 978-7-111-52840-1

定价: 49.90 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

电话服务

网络服务

服务咨询热线: (010) 88379833

机工官网: www.cmpbook.com

读者购书热线: (010) 88379649

机工官博: weibo.com/cmp1952

教育服务网: www.cmpedu.com

封面无防伪标均为盗版

金 书 网: www.golden-book.com

前 言

单片机是将运算器、控制器、存储器和输入、输出接口集成在一块芯片上的单片型微型计算机，广泛应用于家用电器、工业控制和仪器仪表中。MCS-51 单片机因其简单、易学、价格低廉等优点，一经推出就受到广泛欢迎，是目前市场占有率较高的一系列单片机。

作者在多年的 MCS-51 单片机教学实践中发现，传统单片机教学以汇编语言为主，而汇编语言难度较大、不易掌握，使得部分学生无法有效编写程序来控制单片机硬件，从而影响硬件知识学习的效果。另外，单片机学习离不开硬件操作，而对于单片机初学者而言，搭建一个实际的硬件系统难度和成本均比较高。为解决上述两个问题，本书将 C51 语言程序设计方法和 Proteus 虚拟仿真软件作为重要内容加以介绍。C51 语言语法灵活，便于快速掌握，与汇编语言对照学习可以降低理解单片机程序功能的难度。Proteus 软件可以利用软件模拟单片机及其他元器件，并支持电路原理图级别的单片机系统仿真和调试，能够降低硬件学习的成本和难度。

本书共分如下 9 章：

第 1 章为绪论，介绍微型计算机的基础知识，包括微型计算机的组成、层次关系、体系结构、数制、编码以及单片机的发展概况等。

第 2 章为 MCS-51 单片机的基本结构，介绍 MCS-51 单片机的体系结构、内部资源、引脚功能和特性，以及单片机的时钟电路、复位电路等。

第 3 章为 MCS-51 单片机汇编语言程序设计，介绍汇编语言伪指令和指令的使用方法，通过实例讲解顺序程序、分支程序、循环程序和子程序的设计方法等。

第 4 章为 MCS-51 单片机的内部功能单元，讲解中断系统、外部中断、定时/计数器和串行通信接口的工作原理及其汇编语言程序设计方法。

第 5 章为 MCS-51 单片机的并行扩展，讲解利用单片机 I/O 引脚扩展片外存储器和并行 I/O 接口的方法，包括程序存储器、数据存储器、简单 I/O 接口、可编程并行接口 8255A、键盘、显示器、D-A 转换器 DAC0832 和 A-D 转换器 ADC0809 等的扩展。

第 6 章为常用串行总线接口技术，介绍 SPI 总线、I²C 总线和单总线的工作原理和接口扩展方法，并分别给出 A-D 转换器 TLC2543（SPI 总线）、日历时钟芯片 PCF8563（I²C 总线）和温度传感器 DS18B20（单总线）的扩展实例。

第 7 章为 Keil μ Vision4 集成开发环境使用，介绍利用 Keil μ Vision4 软件进行单片机程序设计和调试的方法。

第 8 章为 C51 语言程序设计基础，介绍 C51 语言的基础知识，并给出了第 4 章和第 5 章中部分汇编语言程序的 C51 语言对照程序。

第 9 章为 Proteus 虚拟仿真，介绍利用 Proteus 软件集成环境 ISIS 绘制单片机系统电路原理图（仿真模型）的方法，以及进行 ISIS 和 Keil 联机仿真调试的方法，并给出了简易音乐演奏系统和电动机 PWM 调速系统的设计实例。

本书由张仁彦、高正中、黄鹤松、薛琳和孙秀娟共同编写。张仁彦编写了第1、3、4、5、6、8、9章，并负责全书的统稿和定稿。高正中和黄鹤松编写了第2章。薛琳和孙秀娟编写了第7章。

感谢赵洪亮教授和郑丰隆教授在本书编写过程中给予的帮助。本书得到了山东科技大学自动化学院名校工程建设项目（项目编号：MX-JCJS-8）的资助，在此对学院给予的支持表示衷心的感谢。

由于作者水平有限，且编写时间仓促，书中难免有错误和不妥之处，恳请各位读者批评指正。

作者

目 录

前言

第 1 章 绪论	1
1.1 微型计算机的发展历史	1
1.2 微型计算机的组成	2
1.2.1 微型计算机系统的层次关系和体系结构	2
1.2.2 微处理器	5
1.2.3 存储器	6
1.2.4 I/O 接口	7
1.3 微型计算机的数制和编码	7
1.3.1 数制	8
1.3.2 数据在计算机中的表示	10
1.3.3 常用编码	11
1.4 单片机简介	13
1.4.1 单片机的发展和应用	13
1.4.2 MCS-51 系列单片机概况	13
1.5 小结	13
1.6 习题	14
第 2 章 MCS-51 单片机的基本结构	15
2.1 MCS-51 单片机的体系结构	15
2.2 MCS-51 单片机的内部资源	16
2.2.1 中央处理器 (CPU)	17
2.2.2 存储器和寄存器	18
2.2.3 常用特殊功能寄存器的功能	23
2.3 MCS-51 单片机的引脚功能	24
2.4 并行 I/O 端口的引脚特性	27
2.4.1 P0 口的引脚特性	27
2.4.2 P1 口的引脚特性	29
2.4.3 P2 口的引脚特性	29
2.4.4 P3 口的引脚特性	30
2.5 MCS-51/52 单片机的最小系统电路	31
2.5.1 时钟电路和时钟信号	31
2.5.2 复位电路	31
2.6 小结	33
2.7 习题	33

第3章 MCS-51 单片机汇编语言程序设计	35
3.1 汇编语言的伪指令.....	35
3.2 指令格式和相关符号.....	39
3.2.1 指令格式.....	39
3.2.2 指令的分类和指令描述符号.....	40
3.3 指令的寻址方式.....	41
3.3.1 立即数寻址.....	41
3.3.2 寄存器寻址.....	41
3.3.3 直接寻址.....	42
3.3.4 寄存器间接寻址.....	42
3.3.5 位寻址.....	42
3.3.6 变址寻址.....	43
3.3.7 相对寻址.....	43
3.4 MCS-51 指令集.....	44
3.4.1 数据传送指令.....	44
3.4.2 算术运算指令.....	51
3.4.3 逻辑运算指令.....	55
3.4.4 移位指令.....	58
3.4.5 控制转移指令.....	58
3.5 汇编语言程序设计.....	65
3.5.1 顺序程序设计.....	65
3.5.2 分支程序设计.....	67
3.5.3 循环程序设计.....	72
3.5.4 子程序设计.....	78
3.6 小结.....	86
3.7 习题.....	86
第4章 MCS-51 单片机的内部功能单元	88
4.1 中断系统.....	88
4.1.1 单片机与外部设备的数据传送方式.....	88
4.1.2 MCS-51 单片机中断系统的功能和结构.....	89
4.2 外部中断.....	95
4.2.1 外部中断的初始化设置.....	95
4.2.2 中断程序设计方法.....	96
4.2.3 外部中断请求的撤除.....	98
4.2.4 外部中断源的扩展.....	98
4.3 定时/计数器.....	100
4.3.1 定时/计数器的基本工作原理.....	100
4.3.2 与定时/计数器相关的寄存器.....	101

4.3.3	定时/计数器的工作方式	102
4.3.4	定时/计数器的初始化	104
4.3.5	定时功能应用举例	106
4.3.6	计数功能应用举例	109
4.3.7	测高电平时长举例	111
4.4	串行通信接口	112
4.4.1	串行通信的基础知识	113
4.4.2	串行通信的类型	114
4.4.3	MCS-51 单片机串口的的基本结构	115
4.4.4	MCS-51 单片机串口的相关特殊功能寄存器	115
4.4.5	串口的工作方式	118
4.4.6	RS-232C 串行通信接口	125
4.4.7	RS-485 串行通信接口	127
4.5	小结	128
4.6	习题	128
第 5 章	MCS-51 单片机的并行扩展	130
5.1	MCS-51 单片机的片外并行总线	130
5.1.1	总线的作用	130
5.1.2	MCS-51 单片机片外总线的构成	130
5.2	片外存储器的扩展	132
5.2.1	片外程序存储器的连接与访问	132
5.2.2	片外数据存储器的连接与访问	135
5.2.3	单片机访问片外存储空间的时序	144
5.3	并行 I/O 接口的扩展	147
5.3.1	I/O 接口概述	147
5.3.2	MCS-51 单片机并行 I/O 口的使用	149
5.3.3	简单并行 I/O 接口的扩展	149
5.4	并行接口芯片 8255A 的扩展	153
5.4.1	8255A 的内部结构和引脚	153
5.4.2	8255A 的控制字	155
5.4.3	8255A 的工作方式	157
5.4.4	应用举例	161
5.5	显示器与键盘接口的扩展	163
5.5.1	显示器的扩展	163
5.5.2	键盘的扩展	174
5.6	D-A 和 A-D 接口的扩展	185
5.6.1	D-A 转换器的工作原理和性能指标	186
5.6.2	DAC0832 芯片的接口扩展	188

5.6.3	A-D 转换器的工作原理和性能指标	198
5.6.4	ADC0809 芯片的接口扩展	201
5.7	小结	207
5.8	习题	207
第 6 章	常用串行总线接口技术	209
6.1	SPI 串行总线	209
6.1.1	SPI 总线的引脚功能和时序	209
6.1.2	SPI 总线 A-D 转换芯片 TLC2543	210
6.1.3	TLC2543 的应用实例	213
6.2	I ² C 总线	216
6.2.1	I ² C 总线的引脚功能和时序	217
6.2.2	I ² C 总线日历时钟芯片 PCF8563	219
6.2.3	PCF8563 的应用实例	221
6.3	单总线	230
6.3.1	单总线的引脚功能和时序	230
6.3.2	DS18B20 的使用方法	232
6.3.3	DS18B20 的应用实例	234
6.4	小结	239
6.5	习题	239
第 7 章	Keil μVision4 集成开发环境使用	240
7.1	建立 Keil 工程	240
7.2	生成源程序文件	242
7.3	工程的基本设置	243
7.4	程序的运行和调试	246
7.4.1	程序的编译和链接	247
7.4.2	程序的调试	248
7.5	小结	255
7.6	习题	255
第 8 章	C51 语言程序设计基础	256
8.1	计算机程序设计语言概述	256
8.2	C51 语言的变量	257
8.2.1	变量的数据类型	257
8.2.2	变量的存储	259
8.3	C51 语言的常量	260
8.4	运算符	261
8.5	数组	263
8.6	C51 语言的基本语句	263
8.6.1	条件语句	263

8.6.2	开关语句	264
8.6.3	循环语句	264
8.7	函数	265
8.7.1	函数的定义	265
8.7.2	函数的调用	266
8.8	C51 程序设计实例	266
8.8.1	外部中断程序设计	266
8.8.2	定时/计数器程序设计	267
8.8.3	串行接口程序设计	273
8.8.4	并行接口程序设计	279
8.8.5	键盘显示器接口程序设计	281
8.9	小结	299
8.10	习题	299
第 9 章	Proteus 虚拟仿真	300
9.1	集成环境 ISIS 的使用	300
9.1.1	原理图的绘制	301
9.1.2	虚拟模型的仿真运行	309
9.1.3	基于 ISIS 的汇编语言程序设计	313
9.1.4	ISIS 和 Keil 的联机调试	315
9.2	虚拟仿真实例	316
9.2.1	简易音乐演奏系统设计	316
9.2.2	直流电动机转速 PWM 控制	323
9.3	小结	337
9.4	习题	337
	参考文献	338

第1章 绪 论

微型计算机 (Microcomputer) 是现代电子技术和信息技术发展的产物, 在生产和生活中应用广泛, 其中, 最为人所熟悉的是个人计算机 (Personal Computer, PC)。本书所讲的单片机是一种将计算机各组成部分集成在一片芯片上的微型计算机, 虽然不被普通用户所认识, 但同样广泛应用于人们的日常生活中, 如电视机、电冰箱、打印机和扫描仪等家用电器和办公设备中。本章将主要介绍一些与单片机相关的微型计算机的基础知识, 为读者后续章节的学习奠定良好的基础。

1.1 微型计算机的发展历史

从结绳计算、算筹到计算尺, 人类从远古时期就已开始探索提高计算速度和效率的方法。

1642 年, 法国数学家使用齿轮等配件制造了世界上第一台机械式计算机——帕斯卡加法器, 这是人类从手动计算时代进入机械式计算时代的里程碑。

1801 年, 法国机械师将穿孔纸带上的小孔用于自动提花机工作流程和步骤的控制, 这是现代计算机程序设计思想的萌芽。而纸带上的“有孔”和“无孔”分别类似于二进制数的 0 和 1, 是二进制数在机械控制中的早期应用。1843 年, 英国数学家查尔斯·巴贝奇受这种“穿孔纸带”控制思想的启发, 设计了一种通用的自动计算机器——分析机。分析机以齿轮为主要部件, 由蒸汽机提供动力, 齿轮存放数据, 通过齿轮间的啮合完成计算, 穿孔纸带控制运算过程。虽然由于设计理念超越时代, 巴贝奇并没有成功地制造出一台实际可用的分析机, 但是分析机已经具备了现代计算机的某些基本特征, 如: 存放数据的齿轮相当于存储器, 齿轮啮合完成了运算器的工作, 而穿孔纸带则是控制机器工作流程的程序。

1854 年, 英国数学家布尔创立了布尔代数, 这是现代计算机工作的重要理论基础之一。1936 年, “人工智能之父”艾伦·麦席森·图灵在其论文《论可计算数及其在判定问题上的应用》中提出了算法 (Algorithm) 的概念和一种抽象计算机 (Computing machine) 模型——“图灵机”。图灵机的基本思想是用机器模拟人用纸笔进行计算的过程, 是现代计算机和人工智能领域的开端。

与图灵同时代, 被称为“计算机之父”的美国数学家冯·诺依曼研究了离散变量自动电子计算机 (Electronic Discrete Variable Automatic Computer, EDVAC), 并和他的研究小组发表了“存储程序的通用计算机方案”。该方案解决了计算机设计中的许多关键问题, 其中三个主要设计思想需要本书读者掌握:

1) 计算机采用的数制为二进制。采用二进制设计可降低计算机的结构复杂度。

2) 计算机由五部分组成, 包括运算器、控制器、存储器、输入设备和输出设备。其中, 运算器可以完成各种算术和逻辑运算; 控制器能够控制计算机的各部件协调工作; 存储器用于存放程序指令和数据; 输入和输出设备用于实现人与计算机之间的交互。

3) 计算机的工作原理是“存储程序的原理”，即计算机工作之前，程序与数据预先存放在存储器的存储单元中；计算机工作时，控制器按照指令的存放顺序（存储单元的地址顺序）从存储单元中读取指令，然后分析并执行指令；若被执行的指令具有判断或转移的功能，则根据判断结果或转移要求确定后续指令读取的顺序，从而控制指令的执行顺序；上述过程将重复进行，直到遇到停机指令。

“存储程序的通用计算机方案”的提出标志着人类进入了电子计算机时代，是计算机科学发展的又一座里程碑。而按照该方案设计的计算机被称为“冯·诺依曼机”，世界上的第一台通用计算机“埃尼阿克”（Electronic Numerical Integrator And Calculator, ENIAC）就是按照该方案设计的。

从埃尼阿克起，微型计算机的发展经历了电子管计算机、晶体管计算机、集成电路计算机和大规模集成电路计算机四个阶段。电子管计算机以电子管为主要逻辑器件，使用磁鼓存储数据，体积大、运算速度慢，编程语言为机器语言；晶体管计算机以比电子管体积更小的晶体管为主要器件，采用磁芯存储器，速度快、价格昂贵，可以使用高级语言（如 FORTRAN 语言）进行程序设计；集成电路将多个元器件集成在一片半导体芯片上，以集成电路为主要逻辑器件的计算机体积更小、速度更快、功耗更低；从 20 世纪 70 年代初开始至今，计算机进入了大规模集成电路时代，一片半导体芯片上可以集成几十万甚至几百万个元器件，使得计算机的体积更小、价格更低、性能和可靠性更高。

1.2 微型计算机的组成

在微型计算机的五个组成部分（运算器、控制器、存储器和输入、输出设备）中，运算器和控制器是核心部分，由它们所构成的运算和控制中心被称为微处理器（Microprocessor）或中央处理单元（Central Processing Unit, CPU）。存储器用于存放程序指令和数据，可分为只读存储器（Read-Only Memory, ROM）和随机存取存储器（Random-Access Memory, RAM）两大类。输入/输出（I/O）设备因其电压、电流和数据传输速度等与微处理器不匹配，而必须通过输入/输出接口（I/O 接口）才能与微处理器相连。本节将介绍微型计算机系统的层次关系和体系结构及微型计算机各组成部分的功能和相关基础知识。

1.2.1 微型计算机系统的层次关系和体系结构

微处理器、存储器和 I/O 接口需要通过总线连接在一起，总线按功能可以分为三类：①地址总线（Address Bus, AB），负责传输存储单元的地址信息，微处理器通过地址信息才能找到存储单元或 I/O 接口；②数据总线（Data Bus, DB），负责在 CPU 和存储器（或 I/O 接口）之间传输数据；③控制总线（Control Bus, CB），用于传输微处理器的控制信号，如确定数据总线上的数据流向（数据由微处理器流向存储器或 I/O 接口时，被视为输出数据，即 CPU 执行“写”操作；反之，被视为输入数据，即 CPU 执行“读”操作）。

1. 微型计算机系统的层次关系

图 1-1 给出了微型计算机的组成结构图，图 1-2 给出了微型计算机系统的层次关系，由这两个图可知，仅有微处理器无法构成微型计算机，而没有软件支持的微型计算机硬件也无

法工作，只有软件和硬件配合构成的微型计算机系统才能为人所用。

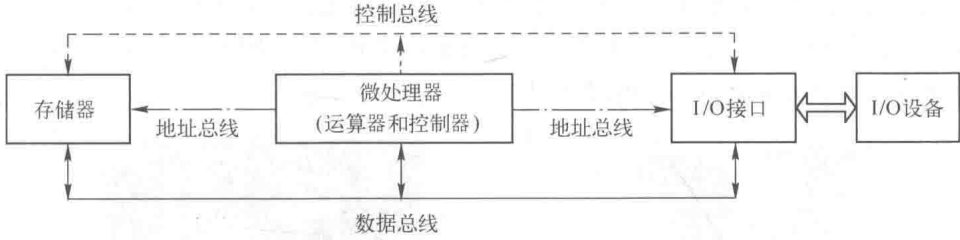


图 1-1 微型计算机的组成结构

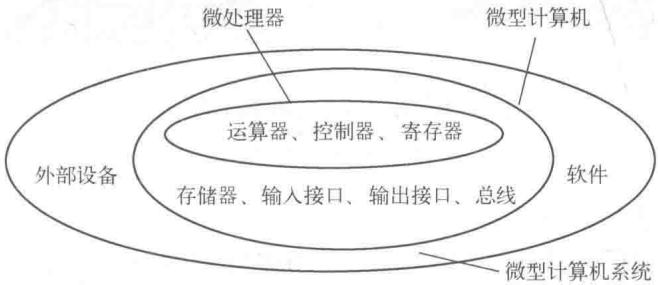


图 1-2 微型计算机系统的层次关系

在 PC 系统中，运算器和控制器集成在一片芯片上，被称为微处理器芯片，其外形如图 1-3 所示。而单片机是将微处理器、存储器和 I/O 接口集成在一片芯片上的单片型微型计算机，简称为单片机 (Single-chip Computer)，其外形如图 1-4 所示。

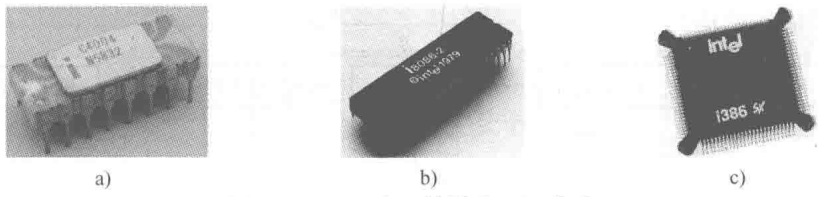


图 1-3 Intel 公司的微处理器芯片
a) Intel 4004 b) Intel 80286 c) Intel 80386

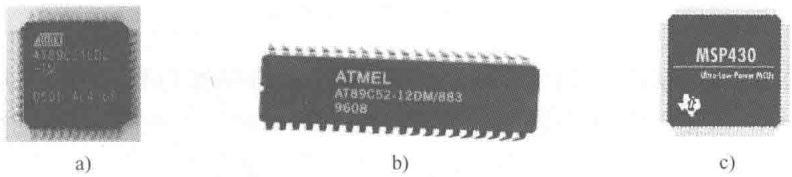


图 1-4 单片机芯片
a) AT89C51 b) AT89C52 c) MSP430

作为半导体芯片，微处理器芯片和单片机芯片均利用引脚与其他电路或芯片相连，其引脚按功能可以分为供电引脚、传输数据的引脚、传输地址的引脚、传输控制信号的引脚和其他辅助功能引脚，其中传输数据 (Data)、地址 (Address) 和控制 (Control) 信号的引脚被称为总线 (BUS)。

图 1-5 给出了 PC 系统和单片机系统的外观图。PC 系统的特点是功能丰富、用途广、价

格高，属于通用型微型计算机，其核心是集成了运算器和控制器的微处理器芯片，而存储器和 I/O 接口被放置在多块不同的印制电路板上。与通用型微型计算机不同，单片机功能简单、用途单一、价格便宜，属于专用型微型计算机，常被用作控制系统的控制器，因此也被称为微控制器（Microcontroller Unit, MCU）。

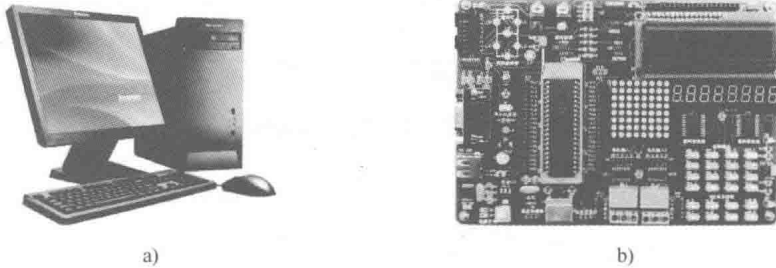


图 1-5 PC 系统与单片机系统的外观图

a) PC 系统 b) 单片机系统

2. 微型计算机的体系结构

1964 年，IBM 公司的阿姆达尔将计算机体系结构（Computer Architecture）定义为“程序员所看到的计算机属性，即概念性结构与功能特性”。目前，主要的计算机体系结构有冯·诺依曼结构和哈佛结构。

(1) 冯·诺依曼结构

按照冯·诺依曼的“存储程序的原理”所设计的计算机的体系结构为冯·诺依曼结构（也被称为普林斯顿结构），其系统结构如图 1-6 所示。使用 Intel 公司 x86 系列微处理器的 PC 均为冯·诺依曼结构。

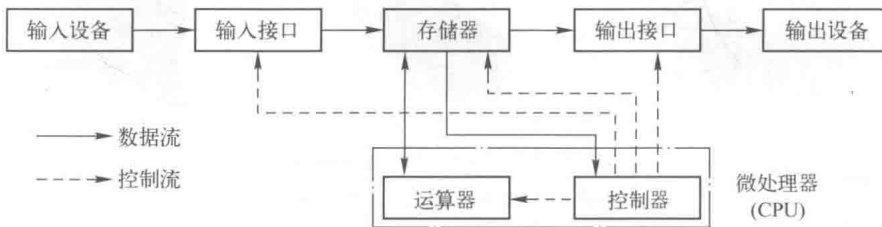


图 1-6 冯·诺依曼结构

冯·诺依曼结构的特点是，指令和数据存放在同一个存储器的不同存储单元中，使用同一套总线（地址总线、数据总线和控制总线）进行读或写的访问。这种体系结构的缺点是：

1) 因为使用同一套总线访问指令和数据，所以数据和指令的宽度（即所含二进制数的位数）是相同的，而且不能同时访问指令和数据。

2) 因为指令和数据在存储器中混合存放，为了避免混淆，必须在程序中进行存储器空间的逻辑划分，将指令和数据划分入不同的逻辑空间，例如：Intel 公司的 16 位 CPU 8086 将存储器划分成不同的逻辑段，包括：存放数据的数据段和存放指令的代码段等，这使得计算机程序的结构相对复杂。

(2) 哈佛结构

计算机的哈佛结构如图 1-7 所示，与冯·诺依曼结构相比，其最大特点是指令和数据分

别存放在不同的物理存储器中，并通过两套总线进行访问。这种结构的优点是：

1) 指令和数据的宽度可以不同，可以实现指令和数据的同时访问。

2) 因为指令和数据的存储空间在物理上是独立的，因此不需要在程序中进行存储器空间的逻辑划分，程序结构相对简单。哈佛结构因其能够有效提高计算机的数据吞吐量，而被广泛应用于嵌入式微型计算机，如以 MCS-51 单片机为代表的各种微控制器。

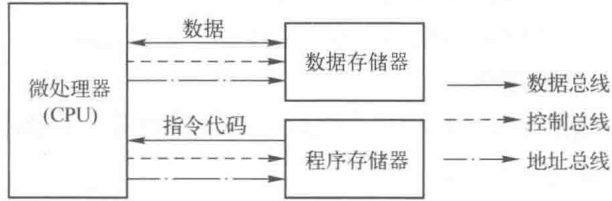


图 1-7 哈佛结构

1.2.2 微处理器

微处理器 (CPU) 是计算机的核心部件，其中除了运算器和控制器外，还包括用于暂存数据的寄存器和传输信息用的内部总线。图 1-8 为一个简化的 CPU 模型，CPU 需要通过三总线 (数据总线、地址总线和控制总线) 与存储器和 I/O 接口进行通信和联络。本小节将介绍微处理器各组成部件的功能以及微处理器的主要性能指标。

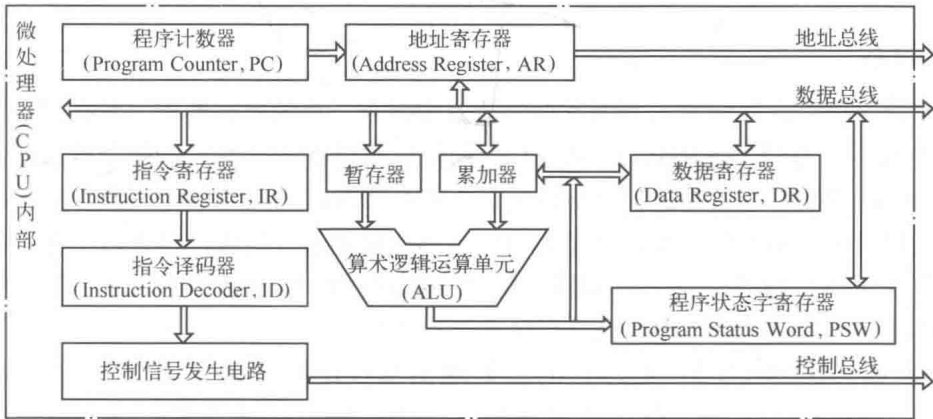


图 1-8 简化的 CPU 模型

1. 微处理器各部件的功能

(1) 运算器

运算器由算术逻辑运算单元 (Arithmetic and Logical Unit, ALU)、累加器和暂存器等部件构成。ALU 是运算器的核心部件，可以完成两个数的加法、减法、比较以及与、或、非等运算，参与运算的两个数分别由累加器和暂存器提供。ALU 的运算结果被送回累加器，并且运算结果的状态将被记录在程序状态字 (Program Status Word, PSW) 寄存器中。这里所谓的运算结果状态是指运算是否产生了进位、借位，运算结果是否为零，是否为负数等，每种状态均以 1 位二进制数来表示。

(2) 寄存器

寄存器是 CPU 内部用于存储信息的物理器件。所谓的信息可以是数据、地址或指令。比如：累加器是用于存放数据的寄存器；PSW 是用于存放 ALU 运算结果状态的寄存器；而指令寄存器（IR）存放从存储器中读取的指令代码。

(3) 控制器

控制器是控制和协调计算机各部件协同工作的机构，主要包括程序计数器（PC）、指令寄存器（IR）、指令译码器（ID）和控制信号发生电路。

2. 微处理器的主要性能指标

微处理器的主要性能指标有字长和指令执行时间，分别用于衡量微处理器的运算能力和运算速度。

(1) 字长

字长是微处理器一次可以处理的二进制数的位数。字长越长，CPU 的计算能力越强、计算速度越快。比如，Intel 公司 1971 年推出的第一代微处理器 Intel 4004（见图 1-3a）的字长是 4 位，每次只能进行 4 位二进制数计算，4 位二进制无符号数的数值范围是 0~15；而该公司生产的微处理器 Intel 80386（如图 1-3c 所示）的字长是 32 位，每次可以完成 32 位二进制数的计算，32 位二进制无符号数的数值范围是 0~4294967295。

(2) 指令执行时间

指令执行时间越短，速度越快。指令执行的时间与微型计算机的时钟频率有关，每条指令执行所消耗的时钟周期个数是固定的，因此时钟频率越高，指令执行速度越快。

1.2.3 存储器

在微型计算机中，存储器主要用于存放数据和指令。存储器有两类，包括随机存取存储器（RAM）和只读存储器（ROM）。RAM 中的信息可以被读、写，既能存放数据，也能存放指令代码。而 ROM 中的信息只能被读取，不能被修改，因此 ROM 只能存放指令代码或程序执行过程中保持不变的数据。存储器由半导体存储器芯片构成，包含若干个存储单元，每个存储单元可以存放若干位二进制数，每个存储单元都被分配一个地址，即存储单元地址。微处理器读、写存储器时必须提供存储单元的地址。

图 1-9 给出了 MCS-51 单片机的微处理器从程序存储器中读取一条指令“MOV A,#12H”（该指令中“#12H”代表十六进制数 12H，A 代表累加器，指令功能是将数字 12H 送入累加器）的过程示意图，可以帮助读者更好地理解计算机的工作原理，即“存储程序的原理”，另外该图中的①~⑥是指令执行步骤的序号。如图 1-9 所示，访问存储器时必须提供被访问存储单元的地址，而被取指令在程序存储器中的存放地址由程序计数器（PC）提供。读指令的过程中，PC 的值会自动增加（当程序出现分支或循环时可能是减小）指向下一个存储单元，为取下一个指令做准备。需要注意的是，指令操作码用于指明指令要完成的操作，需要经指令译码器翻译后才能被 CPU “理解”，而指令操作数是被指令处理的数据，不需要指令译码器翻译。另外，单片机进行数据存储器读、写的过程与读取指令操作数的过程类似，主要差别是数据存放在数据存储器中，并且其存储单元的地址不由 PC 提供。

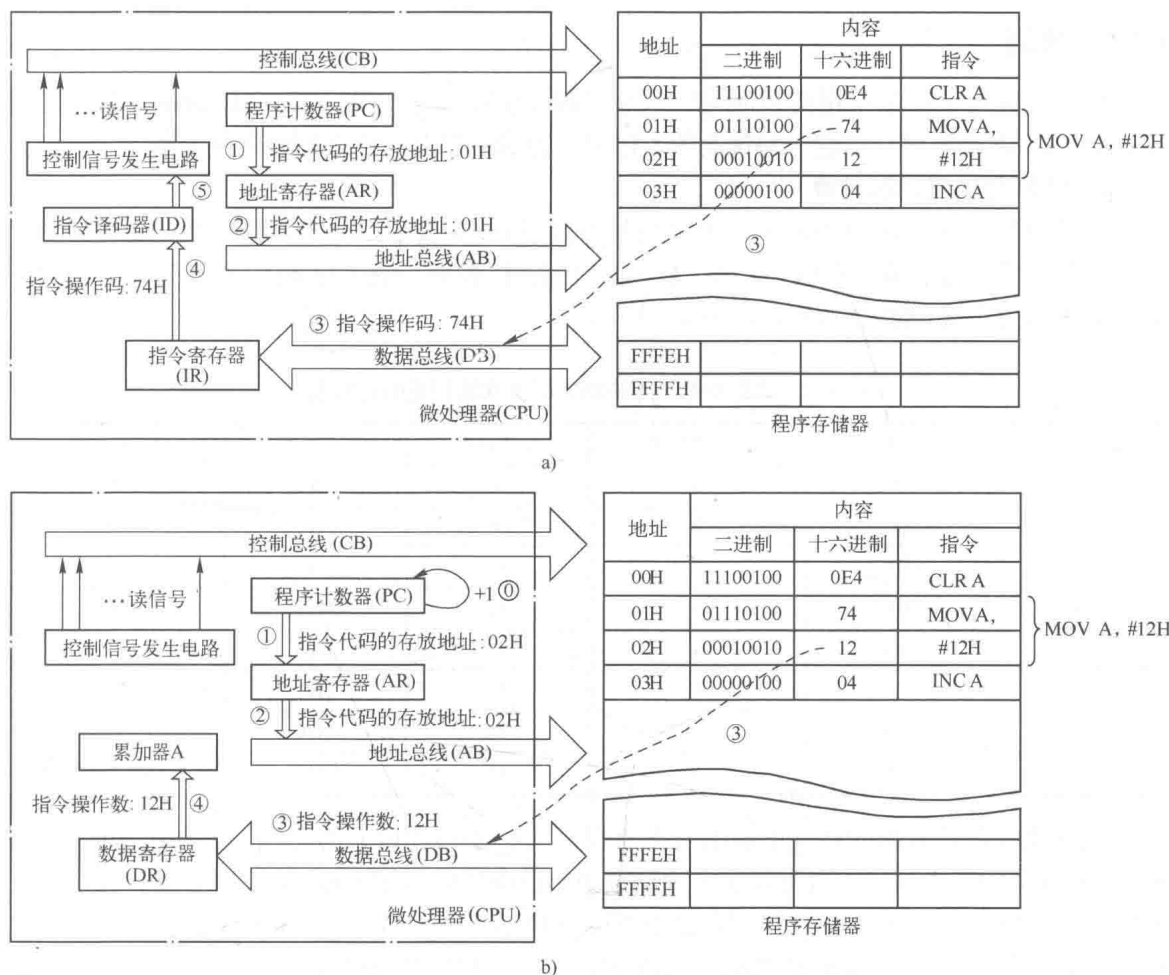


图 1-9 指令“MOV A,#12H”读取过程示意图
a) 读取操作码 b) 读取操作数

1.2.4 I/O 接口

I/O 接口是微处理器和 I/O 设备之间的桥梁，常用的 I/O 接口包括串行通信接口、定时器接口和模拟通道接口等。I/O 接口通过三总线与微处理器相连，为了区分各 I/O 接口，CPU 给每个 I/O 接口分配固定且不同的地址。与访问存储器类似，CPU 访问 I/O 接口时也需要先将 I/O 接口的地址送入地址总线，然后再通过数据总线传输数据。

1.3 微型计算机的数制和编码

在微型计算机中，所有信息（如数值、符号和图像等）均以二进制形式存储、传输和计算。由于二进制数冗长、不方便读写和辨认，因此，现代微型计算机也支持编程时使用书写长度更短的十六进制数和十进制数，同时也为各种非数值信息提供了相应的数值编码（即用