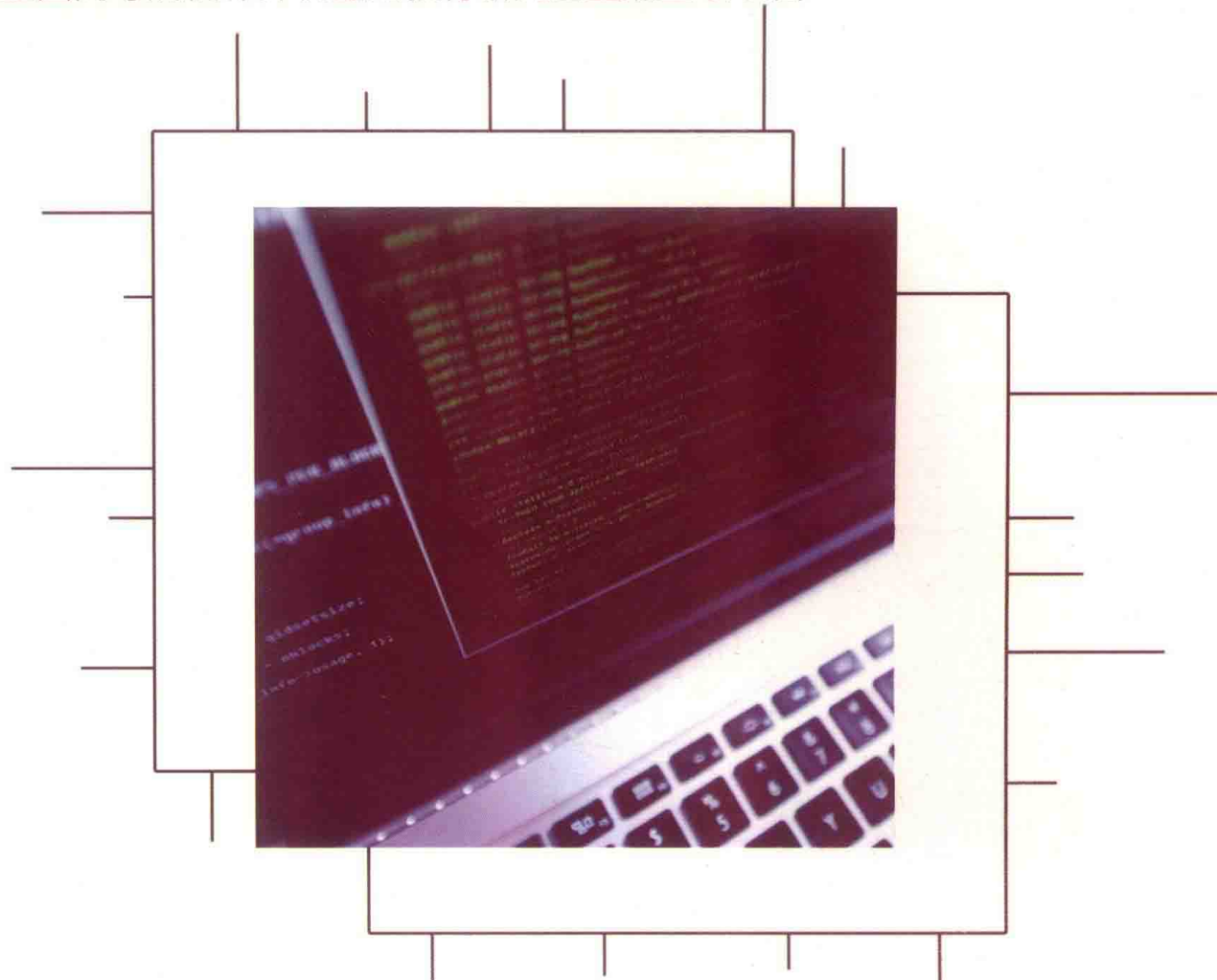




国家新闻出版改革发展项目库入库项目
高等院校计算机类规划教材
全国高等院校计算机基础教育研究会重点立项项目



微机原理与接口技术 辅导与实验

秦金磊 王桂兰 朱有产 编著

- 配套教材习题同步，典型案例项目精讲
- 视频资源内容丰富，全面覆盖实践操作
- 所有步骤翔实可靠，准确实现项目仿真



北京邮电大学出版社
www.buptpress.com



国家新闻出版改革发展项目库入库项目
高等院校计算机类规划教材
全国高等院校计算机基础教育研究会重点立项项目

微机原理与接口技术 辅导与实验

秦金磊 王桂兰 朱有产 编著



北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书作为一本辅导与实验教材,主要内容包括章节学习辅导、实验工具、精选案例与实验指导。章节学习辅导部分包括知识点梳理、重点难点剖析、例题精讲、习题解答、拓展学习等。实验工具部分包括汇编实验环境 MASM 与仿真实验工具 Proteus 的安装、使用与调试,典型例题精讲及视频学习。精选案例与实验指导部分包括汇编程序设计,存储器及其扩展,输入/输出技术,可编程控制芯片 8255A、8253A、8259A 等相关的重点案例及综合设计实验等。本书内容全面、实用性强,重要原理、技术与应用讲述清晰,通过视频提供详细内容,讲述有特点和新意,使读者能够快速理解和掌握微机与接口技术的重要原理和方法。本书中提供的案例全部通过实验环境调试。

本书可作为高等院校理工科自动化、电气与电子类等相关专业的本科、成人高等教育或大专层次的配套实验教材,对研究生和从事微机测控及接口技术应用的工程技术人员具有很好的参考价值。

图书在版编目(CIP)数据

微机原理与接口技术辅导与实验 / 秦金磊, 王桂兰, 朱有产编著. -- 北京: 北京邮电大学出版社, 2021. 3
ISBN 978-7-5635-6346-3

I. ①微… II. ①秦… ②王… ③朱… III. ①微型计算机—理论—高等学校—教学参考资料②微型计算机—接口技术—高等学校—教学参考资料 IV. ①TP36

中国版本图书馆 CIP 数据核字(2021)第 052901 号

策划编辑: 马晓仟 责任编辑: 王晓丹 米文秋 封面设计: 七星博纳

出版发行: 北京邮电大学出版社

社 址: 北京市海淀区西土城路 10 号

邮政编码: 100876

发 行 部: 电话: 010-62282185 传真: 010-62283578

E-mail: publish@bupt.edu.cn

经 销: 各地新华书店

印 刷: 北京玺诚印务有限公司

开 本: 787 mm×1 092 mm 1/16

印 张: 15

字 数: 402 千字

版 次: 2021 年 3 月第 1 版

印 次: 2021 年 3 月第 1 次印刷

ISBN 978-7-5635-6346-3

定价: 39.00 元

· 如有印装质量问题,请与北京邮电大学出版社发行部联系 ·

前 言

本书是高等院校理工科相关专业基础核心课程“微机原理与接口技术”的配套教材,其目标是使学生从实践的角度出发,掌握微机系统的基本组成、工作原理、接口技术及应用方法,具备微机系统的初步开发能力。作者在总结多年教学科研及实践经验的基础上,结合计算机仿真技术的发展对课程相关资料进行综合分析提炼,编写了本书。

本书内容在选取与组织方面有所突破,以 8086/8088 微处理器和 IBM PC 系列微机为对象,从微机系统应用实践出发,精选案例设计,利用图文与视频同步的形式对重点、难点进行讲解,辅导学生快速掌握微机系统的基本组成、工作原理、接口技术及应用。全书紧扣教材《微机原理与接口技术——基于 Proteus 仿真》,共分 3 章,包括:章节学习辅导、实验工具、精选案例与实验指导。本书以解决学生在实践环节遇到的重点、难点为主线,从知识点梳理、习题解答、拓展学习、案例精讲、实践入门与提高等方面进行学习辅导。本书内容全面、实用性强,重要原理、技术与应用讲述清晰,图文并茂,网络视频同步更新,讲述有特点和新意。书中提供的案例全部调试通过。

本书具有如下特色:

- ① 紧扣重点难点。以应用实践过程中的重点、难点为出发点,通过知识点梳理、习题解析、精选案例设计等手段,把重点讲透,把难点讲清。
- ② 视频同步更新。在进行图文说明的同时配备视频,以全方位展现系统设计过程,使学生更易掌握原理与方法。视频适时在线更新,学生可随时获取最新资料。
- ③ 任务与兴趣并重。在完成精选案例的同时,在拓展学习环节增加相关知识,以引发学生更大的兴趣,进一步促进学生对相关原理与方法的掌握。

本书的第 1 章由秦金磊、王桂兰、朱有产共同编写;第 2、3 章由秦金磊编写;视频录制及附录由秦金磊完成。全书由秦金磊统稿并最后定稿。

本书的编写得到了华北电力大学专业建设平台领导的大力支持;得到了华北电力大学

微机原理教学团队全体教师的大力支持;得到了广州风标电子技术公司的大力支持,该公司技术人员指导了部分 Proteus 仿真实例的设计;得到了全国高等院校计算机基础教育研究会和北京邮电大学出版社的大力支持。在此,作者向所有对本书的编写、出版等工作给予大力支持的单位和领导表示真诚的感谢!

由于作者水平有限,书中难免有错漏之处,敬请广大读者提出宝贵意见。

作 者

2020 年 9 月于华北电力大学

目 录

第 1 章 章节学习辅导	1
1.1 微型计算机基础知识概述(教材第 1 章)学习辅导	1
1.1.1 知识点梳理	1
1.1.2 习题解答	2
1.1.3 拓展学习:初级计算机	3
1.2 微处理器(教材第 2 章)学习辅导	11
1.2.1 知识点梳理	11
1.2.2 习题解答	11
1.2.3 拓展学习:微机总线	17
1.3 指令系统及汇编语言程序设计(教材第 3 章)学习辅导	29
1.3.1 知识点梳理	29
1.3.2 典型案例	30
1.3.3 习题解答	35
1.3.4 拓展学习:指令、宏、嵌套、子程序及模块化	51
1.4 存储器系统(教材第 4 章)学习辅导	67
1.4.1 知识点梳理	67
1.4.2 习题解答	68
1.4.3 拓展学习:E ² PROM	73
1.5 输入/输出技术(教材第 5 章)学习辅导	76
1.5.1 知识点梳理	76
1.5.2 习题解答	77
1.5.3 拓展学习:LED 数码管	81
1.5.4 典型案例	85
1.6 可编程并行 I/O 接口芯片 8255A(教材第 6 章)学习辅导	87
1.6.1 知识点梳理	87
1.6.2 习题解答	87

1.6.3	拓展学习:键盘	92
1.6.4	典型案例	95
1.7	可编程计数器/定时器 8253A(教材第 7 章)学习辅导	96
1.7.1	知识点梳理	96
1.7.2	习题解答	96
1.7.3	典型案例	111
1.8	中断技术及 8259A(教材第 8 章)学习辅导	112
1.8.1	知识点梳理	112
1.8.2	习题解答	113
1.8.3	拓展学习:8259A 的级联方式	117
1.8.4	典型案例	117
1.9	微机系统串行通信及接口(教材第 9 章)学习辅导	121
1.9.1	知识点梳理	121
1.9.2	习题解答	121
1.10	D/A 和 A/D 转换接口(教材第 10 章)学习辅导	122
1.10.1	知识点梳理	122
1.10.2	拓展学习:12 位 D/A 转换器 DAC1210	122
1.10.3	典型应用	123
第 2 章	实验工具	126
2.1	汇编实验环境 MASM	126
2.1.1	软件安装	126
2.1.2	入门案例	128
2.1.3	程序调试与运行	130
2.1.4	寄存器与存储器单元数值的修改	136
2.2	接口仿真实验环境 Proteus 8.4	137
2.2.1	介绍与安装	137
2.2.2	Proteus 8.4 工程的创建	143
2.2.3	可视化界面	145
2.2.4	元件的查找与选择	148
2.2.5	元件的操作	153
2.2.6	连线	155

2.2.7 元件标签	156
2.2.8 器件的标注	157
2.2.9 入门案例	159
第3章 精选案例与实验指导	165
3.1 精选案例	165
3.1.1 微处理器	165
3.1.2 存储器系统	168
3.1.3 输入/输出技术	179
3.1.4 可编程并行 I/O 接口 8255A	184
3.1.5 可编程计数器/定时器 8253A	193
3.1.6 可编程中断控制器 8259A	202
3.1.7 微机系统串行通信及接口	214
3.1.8 D/A 和 A/D 转换接口	218
3.2 基础与综合实验	221
3.2.1 实验 1:汇编基础	221
3.2.2 实验 2:存储器实验	222
3.2.3 实验 3:输入与输出实验	223
3.2.4 实验 4:可编程并行 I/O 接口 8255A	223
3.2.5 实验 5:可编程计数器/定时器 8253A	224
3.2.6 实验 6:可编程中断控制器 8259A	224
3.2.7 实验 7:可编程串行通信接口 8251A	225
3.2.8 实验 8:D/A 和 A/D 转换接口	225
3.2.9 实验 9:综合实验	225
附录	227

第 1 章 章节学习辅导

1.1 微型计算机基础知识概述(教材第 1 章)学习辅导

1.1.1 知识点梳理

微型计算机基础知识结构如图 1.1.1 所示。

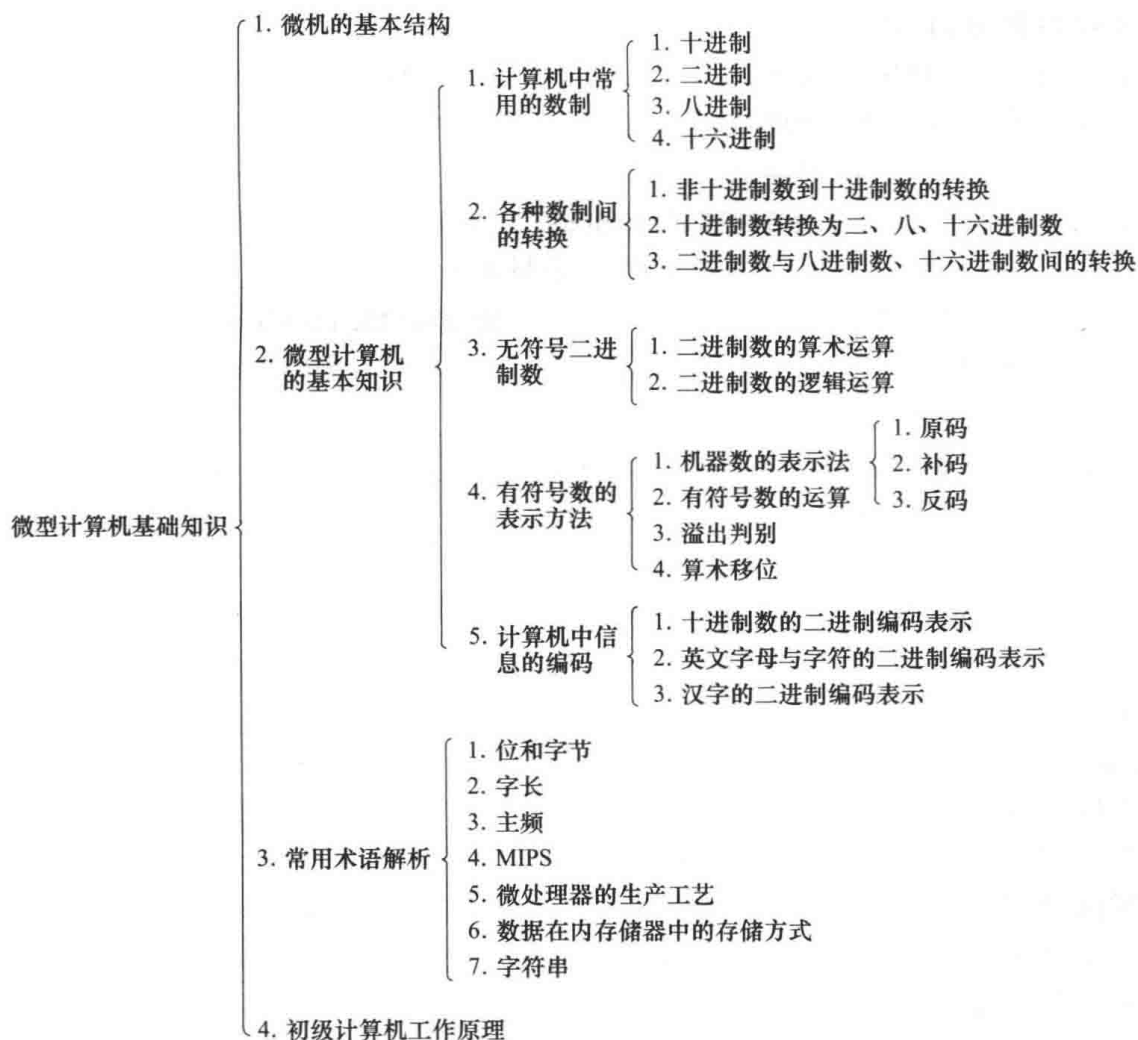


图 1.1.1 微型计算机基础知识结构

重点:补码的计算方法。

难点:溢出判断方法。

1.1.2 习题解答

1. 计算机中常用的计数制有哪些?

答:计算机中常用的计数制有二进制、八进制和十六进制。

2. 请简述机器数和真值的概念。

答:计算机只能识别由 0 和 1 组成的数或代码,有符号数的符号也只能用 0 和 1 来表示,一般用“0”表示正,用“1”表示负,这种将符号数码化,连同同一个符号位在一起的一个数称为机器数。直接用“+”号和“-”号来表示其正负的数为有符号数(该机器数)的真值。

3. 将下列十进制数分别转换为二进制数(保留 4 位小数)、八进制数、十六进制数和 BCD 数。

- ① 125.74 ② 513.85 ③ 742.24 ④ 69.357

答:① $125.74 = 01111101.1011B = 175.54Q = 7D.BH$ 。

压缩 BCD 数为 000100100101.01110100B(125.74H)。

② $513.85 = 100000001.1101B = 1001.64Q = 201.DH$ 。

压缩 BCD 数为 010100010011.10000101B(513.85H)。

③ $742.24 = 001011100110.0011B = 1346.14Q = 2E6.3H$ 。

压缩 BCD 数为 011101000010.00100100B(742.24H)。

④ $69.357 = 1000101.0101B = 105.24Q = 45.5H$ 。

压缩 BCD 数为 01101001.001101010111B(69.357H)。

4. 将下列二进制数分别转换为十进制数、八进制数和十六进制数。

- ① 101011.101B ② 110110.1101B
③ 1001.11001B ④ 100111.0101B

答:① $101011.101B = 43.625D = 53.5Q = 2B.AH$ 。

② $110110.1101B = 54.8125D = 66.64Q = 36.DH$ 。

③ $1001.11001B = 9.78125D = 11.62Q = 9.C8H$ 。

④ $100111.0101B = 39.3125D = 47.24Q = 27.5H$ 。

5. 将下列十六进制数分别转换为二进制数、八进制数、十进制数和 BCD 数。

- ① 5A.26H ② 143.B5H
③ 6CB.24H ④ E2F3.2CH

答:① $5A.26H = 01011010.00100110B = 132.114Q = 90.148D$ 。

压缩 BCD 数为 10010000.000101001000B(90.148H)。

② $143.B5H = 000101000011.10110101B = 503.552Q = 323.6875D$ 。

压缩 BCD 数为 001100100011.0110100001110101B(323.6875H)。

③ $6CB.24H = 011011001011.00100100B = 3313.11Q = 1739.141D$ 。

压缩 BCD 数为 0001011100111001.000101000001B(1739.141H)。

④ $E2F3.2CH = 1110001011110011.00101100B = 161363.13Q = 58099.125D$ 。

压缩 BCD 数为 01011000000010011001.000100100101B(58099.125H)。

6. 8 位和 16 位二进制数的原码、补码和反码可表示的数的范围分别是多少?

答:8 位二进制数的原码可表示的数的范围为 $-127 \sim +127$,反码可表示的数的范围为 $-127 \sim +127$,补码可表示的数的范围为 $-128 \sim +127$;16 位二进制数的原码可表示的数的范

围为 $-32767 \sim +32767$,反码可表示的数的范围为 $-32767 \sim +32767$,补码可表示的数的范围为 $-32768 \sim +32767$ 。

7. 写出下列十进制数的原码、反码、补码表示(采用8位二进制,最高位为符号位)。

- ① 120 ② 62 ③ -26 ④ -127

答: ① $[X]_{\text{原}} = 01111000\text{B}$, $[X]_{\text{反}} = 01111000\text{B}$, $[X]_{\text{补}} = 01111000\text{B}$ 。

② $[X]_{\text{原}} = 00111110\text{B}$, $[X]_{\text{反}} = 00111110\text{B}$, $[X]_{\text{补}} = 00111110\text{B}$ 。

③ $[X]_{\text{原}} = 10011010\text{B}$, $[X]_{\text{反}} = 11100101\text{B}$, $[X]_{\text{补}} = 11100110\text{B}$ 。

④ $[X]_{\text{原}} = 11111111\text{B}$, $[X]_{\text{反}} = 10000000\text{B}$, $[X]_{\text{补}} = 10000001\text{B}$ 。

8. 已知补码表示的机器数,分别求出其真值。

- ① 46H ② 9EH ③ B6H ④ 6C20H

答: ① $[X]_{\text{补}} = 46\text{H}$, 真值: +70。

② $[X]_{\text{补}} = 9\text{EH}$, 真值: -98。

③ $[X]_{\text{补}} = \text{B6H}$, 真值: -74。

④ $[X]_{\text{补}} = 6\text{C20H}$, 真值: +27680。

9. 已知某个8位的机器数65H,在其作为无符号数、补码带符号数、BCD码以及ASCII码时分别表示什么真值和含义?

答:65H作为无符号数,其真值为+101;作为补码带符号数,其真值为+101;作为BCD码,其真值为65;作为ASCII码,代表字母e。

10. 用现有的知识完成一个交通信号灯系统的初步设计(可以用框图、文字、流程图等方式来描述)。

答:交通信号灯系统的主要功能应包括:①正常通行时交通信号灯的状态;②有特殊情况发生时,交通信号灯的状态;③附加行人通道、左转弯、右转弯、声音示警、LED显示屏等扩展功能。

1.1.3 拓展学习:初级计算机

1. 初级计算机的基本结构

(1) 概述

初级计算机的简化结构如图1.1.2所示。图中虚线以上部分为运算器和控制器,构成计算机的中央处理单元(Central Processing Unit, CPU),虚线以下部分为RAM和I/O接口电路。RAM的容量为 256×8 ,即256个存储单元,每个存储单元存8位二进制数。I/O接口电路是输入设备键盘、输出设备显示器(CRT)与CPU之间的缓冲和连接部件,键盘用来输入原始操作数及解题程序,CRT用来显示运算结果。

通常,在微型计算机中,CPU被做成一个独立的芯片,称为微处理器或微处理机。存储器是位于CPU之外的另一种芯片,称为内存储器或主存储器,它是计算机的一个记忆装置,用来存放以二进制编码形式表示的程序、原始操作数、运算和处理的中间结果及最后结果,需要时可以把它们读出来。

一个存储器包含很多存储单元,被存储的二进制信息分别存放在这些存储单元中。每个存储单元都有自己的编号,这个编号叫作地址。计算机中存储单元的地址也采用二进制编码方式表示。例如,图1.1.2所示的模型机中存储器有256个存储单元,地址编号可以是0~255。256个存储单元若要用二进制编码表示地址号,需8位二进制数,即二进制和十六进制表示的地址编号分别为 $00000000\text{B} \sim 11111111\text{B}$ 和 $00\text{H} \sim \text{FFH}$ 。显然,存储器的存储单元数越多,所需要的

(2) 运算器的结构和功能

运算器主要包括算术逻辑部件(Arithmetic Logic Unit, ALU)、累加器(Accumulator, AL)、操作数暂存器(Operand Temporary Register, OTR)及标志寄存器(Flags)。它们均是8位的。ALU一般是直接执行各种操作和运算的部件,它在控制器的控制下完成各种算术运算(加、减、乘、除等)、逻辑运算(与、或、非、异或等)以及其他操作(取数、送数、移位等)。在初级计算机中,假设ALU仅为一个8位全加器,则可以实现两个8位补码数的加、减运算。累加器用来存放两个操作数中的一个,如被加数或被减数,且存放运算结果。运算的最终结果通过CRT显示出来。操作数暂存器用来存放两个操作数中的另一个,如加数或减数。累加器和操作数暂存器都属于CPU内部的通用寄存器,用来存放操作数和运算的中间结果。微型计算机中有多个通用寄存器,组成一个通用寄存器组。而且,通用寄存器的数目越多,CPU运行起来越方便,也越快。原始操作数通常通过键盘送入内存,CPU执行程序时再从内存中取出。标志寄存器用来存放运算结果的标志信息。例如,运算结果的正负情况用符号标志(Sign Flag, SF)来表示。在不溢出的情况下,若运算结果最高位为1,则 $SF=1$,表明结果为负;否则 $SF=0$,表明结果为正。又如,运算结果最高位的进位(对加法)或借位(对减法)情况用进位/借位标志(Cary Flag, CF)来表示。对于加法运算,若最高位有进位,则 $CF=1$;否则 $CF=0$ 。对于减法运算,若最高位有借位,则 $CF=1$;否则 $CF=0$ 。再如,运算结果的溢出情况可用溢出标志(Overflow Flag, OF)来表示。若运算结果有溢出,则 $OF=1$;否则 $OF=0$ 。

(3) 控制器的结构和功能

1) 控制器的结构

初级计算机中的控制器主要包括程序计数器(Program Counter, PC)或指令指针(Instruction Pointer, IP)、指令寄存器(Instruction Register, IR)、指令译码器(Instruction Decode, ID)、内存地址寄存器(Memory Address Register, MAR)及定时与控制部件(Timing and Control, TC)等。计算机的整个工作过程就是执行程序的过程。程序就是一系列按一定顺序排列的指令。控制器利用指令指挥计算机工作,用户则用指令表达自己的意图并交给控制器指挥机器执行。

2) 控制器的功能

我们知道,为使计算机能自动执行一个解题程序,必须将程序中的指令按解题顺序预先存放在内存中。同时,计算机在执行程序时应能自动把这些指令按顺序逐条取出并执行,为此,要有一个追踪指令地址的PC(或IP)。程序开始执行时,由计算机的操作系统给PC(或IP)赋一个初值,这个初值就是内存中要执行的解题程序的起始地址。然后,每取出一个指令字节,PC(或IP)的内容便自动加1,指向下一个指令字节地址,从而保证指令的顺序执行。只有当程序需要转移时,PC才置入转移到所需要的指令处的新值。

为了完成一条指令所规定的操作,计算机的运算器、内存储器等部件必须在控制器的控制下,相应地完成一系列基本动作,而这些基本动作又必须按时间先后次序、互相配合。为此,首先必须由IR根据PC所指指令地址接收要被执行的指令操作码,直接送ID进行译码。然后,该操作码译码信号便作为一条指令的特征信号送TC,变成一系列按时间顺序排列的控制信号,发向运算器、控制器、存储器及I/O接口电路等,从而控制它们完成该指令所规定的操作。存储器和I/O接口电路由于位于CPU之外,因此TC向它们发送的控制信号需经外部控制总线(Control Bus, CB)传送。

如果指令执行中需要从内存或I/O接口中取操作数,该操作数的地址由指令的操作数地址

部分给出。MAR 具有三态控制功能,它接收 PC 发来的指令地址或来自指令的操作数地址,经三态控制并驱动后,至外部地址总线(Address Bus, AB)送存储器或 I/O 接口电路。AB 是单向的,即存储器或 I/O 接口电路的地址信号均由 CPU 发出。

由于 MAR 及 DR 具有三态控制功能,只有当 CPU 需要通过 AB 及 DB 访问存储器或 I/O 接口时,CPU 内部的地址信号、数据信号才能与 AB 及 DB 连通;否则,它们之间会呈现高阻(即断开)状态。

(4) 内存储器的结构及工作原理

设初级计算机中的内存储器由 256 个存储单元组成,每个存储单元存放 8 位二进制信息,该内存储器的结构如图 1.1.3 所示。由图 1.1.3 可见,内存储器由存储阵列、地址译码器、三态数据缓冲器及控制电路组成。在本例中,256 个存储单元的地址号为 00H,01H,⋯,FFH。从 CPU 发来的 8 位指令地址或操作数地址经 AB 送入存储器的地址译码器进行译码,从而选中所需要的存储单元,进行读/写操作。

三态数据缓冲器对 RAM 来说是双向的,它将存储单元的数据进行三态缓冲控制后,与 CPU 的外部 DB 相连。只有当存储器中的存储单元被选中时,存储器内部的数据信号线才能与外部 DB 连通;否则,将呈现高阻(即断开)状态。存储器中的控制电路则接收来自 CPU 的控制信号(如“访问存储器信号”“读信号”或“写信号”等),经组合变换后对存储器的地址译码、数据存储操作等进行控制。

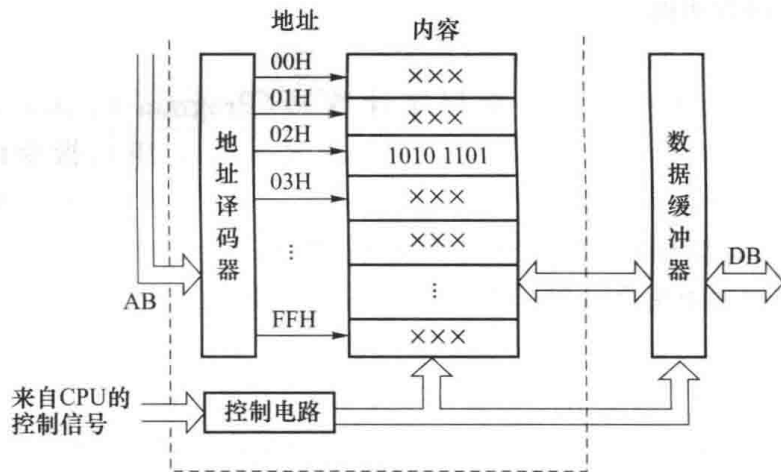


图 1.1.3 内存储器的结构

1) 读操作

图 1.1.4 是存储器读操作示意图。若设图中 04H 存储单元的内容为 00100110B(即 26H),则读出时,首先由 CPU 的 MAR 发出 8 位二进制地址信号 00000100B(即 04H),经 AB 送存储器的地址译码器。同时,由 CPU 的控制部件发出“访问存储器信号”和“读信号”等,这些控制信号经存储器的控制电路组合变换后送往存储器的相应部件,从而选中 04H 存储单元,并将其内容 26H 经三态数据缓冲器及 DB 送 CPU 中某寄存器(如 AL)。数据信息 26H 从存储单元 04H 中读出至 CPU 后,04H 中的内容保持不变。

2) 写操作

图 1.1.5 是存储器写操作示意图。若要把 CPU 中某寄存器(如 AL)的内容写入存储器的某存储单元中,则首先由 CPU 的 MAR 发出地址信号(如 10H),经 AB 送存储器的地址译码器,并由 CPU 的控制部件发出“访问存储器信号”和“写信号”等,经 CB 送往存储器相应部件,从而

选中 10H 单元。然后, CPU 将要写入的数据经 DB 写入 10H 单元。数据写入 10H 单元后, 该单元原存数据将被新写入的数据信息代替。若 10H 单元原内容为 0AH, AL 中的内容为 26H, 则将 AL 中的内容写入 10H 单元后, 10H 单元的内容变为 26H, AL 中的内容不变。

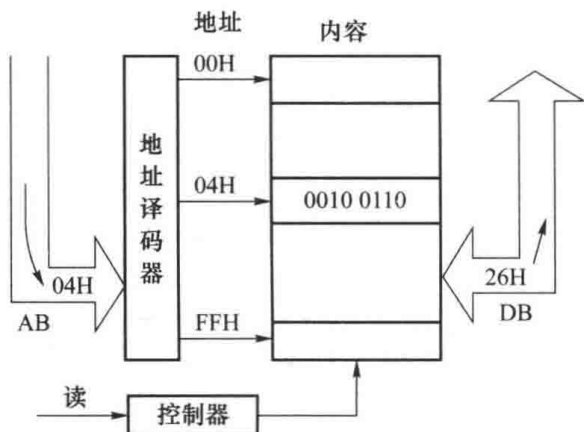


图 1.1.4 存储器读操作示意图

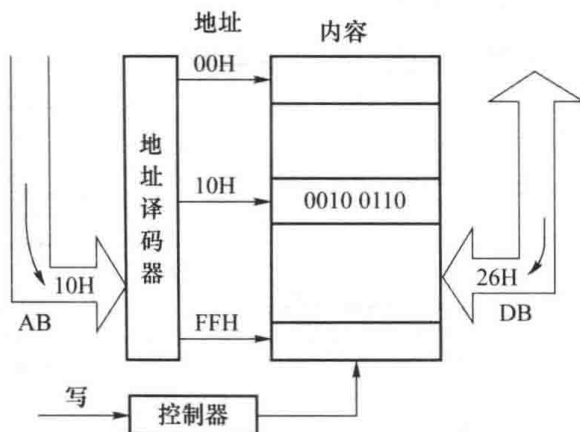


图 1.1.5 存储器写操作示意图

2. 简单程序举例

下面以一个简单的例子来说明程序执行的过程。例如, 求 7 和 10 的和。

第一步, 要为这样的操作编制一个程序。用助记符形式表示的程序如下:

```
MOV    AL,7      ;数 7 送 AL 寄存器
ADD    AL,10     ;AL 寄存器内容与数 10 做加法, 和值保存在 AL 寄存器中
HLT    ;CPU 暂停
```

但是, 模型机并不认得助记符, 指令必须用机器码来表示, 操作数也只能用二进制或十六进制表示。对应的机器码程序如下:

```
10110000      ;第一条指令 MOV AL,n
00000111      ;n = 7
00000100      ;第二条指令 ADD AL,n
00001010      ;n = 10
11110100      ;第三条指令 HLT
```

总共是 3 条指令, 5 字节。

第二步, 程序应该放入存储器中, 若程序放在以 00H (2 位十六进制数) 开始的存储单元内, 则需要 5 个存储单元, 如图 1.1.6 所示。

第三步, 执行程序。在执行程序时, 对 PC 赋以第一条指令的地址 00H, 然后就进入第一条指令的取指阶段, 具体操作过程如下:

- ① 把 PC 的内容 (00H) 送至地址寄存器;
- ② 待 PC 的内容可靠地送至地址寄存器后, PC 的内容自动加 1, 变为 01H;
- ③ 地址寄存器通过地址总线把地址号 00H 送至存储器, 经过地址译码器译码, 选中 00 号存储单元;

④ CPU 发出读命令;

⑤ 所选中的 00 号存储单元的内容 B0H

地址	内容		
十六进制	二进制		
00	0000 0000	1011 0000	MOV AL,n
01	0000 0001	0000 0111	n=7
02	0000 0010	0000 0100	ADD AL,n
03	0000 0011	0000 1010	n=10
04	0000 0100	1111 0100	HLT

图 1.1.6 指令的存放

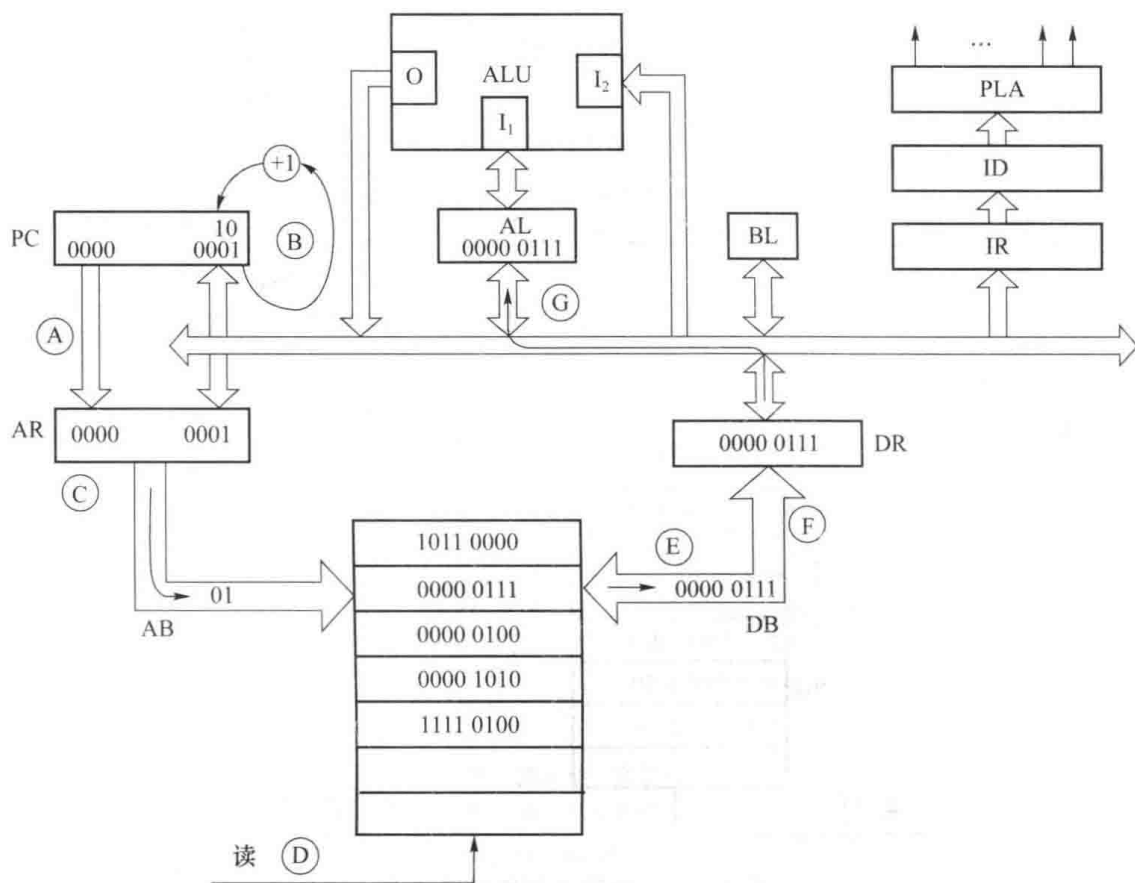


图 1.1.8 取第一条指令的操作数的过程

- ② 待 PC 的内容可靠地送至地址寄存器后,PC 自动加 1,变为 03H;
 - ③ 地址寄存器通过地址总线把地址号 02H 送至存储器,经过译码,选中相应的 02 号存储单元;
 - ④ CPU 发出读命令;
 - ⑤ 选中的存储单元的内容 04H 读至数据总线上;
 - ⑥ 读出的内容通过数据总线送至 DR;
 - ⑦ 因为是取指阶段,读出的为指令,所以 DR 把它送至 IR,经过译码,发出各种控制信息。
- 取第二条指令的过程如图 1.1.9 所示。

对指令进行译码后可知,此为加法指令,以 AL 的内容为一个操作数,另一个操作数在指令的第二个字节中,执行第二条指令必须取出指令的第二个字节。

取第二条指令的第二个字节及执行指令的过程如下:

- ① 把 PC 的内容 03H 送至 AR;
- ② 待 PC 的内容可靠地送至 AR 以后,PC 自动加 1,变为 04H;
- ③ AR 通过地址总线把地址号 03H 送至存储器,经过译码,选中相应的 03 号存储单元;
- ④ CPU 发出读命令;
- ⑤ 选中的存储单元的内容 0AH 读至数据总线上;
- ⑥ 数据通过数据总线送至 DR;
- ⑦ 因为由指令译码已知读出的为操作数,而且要与 AL 中的内容相加,故数据由 DR 通过内部数据总线送至 ALU 的另一个输入端;
- ⑧ AL 中的内容送至 ALU,且执行加法操作;
- ⑨ 相加的结果由 ALU 输出至 AL 中。