

COMPUTER

主教材《汇编语言、微机原理及接口技术（第2版）》为国家十一五规划教材

《汇编语言、微机原理 及接口技术（第2版）》

习题解答与实验指导

郑初华 等编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

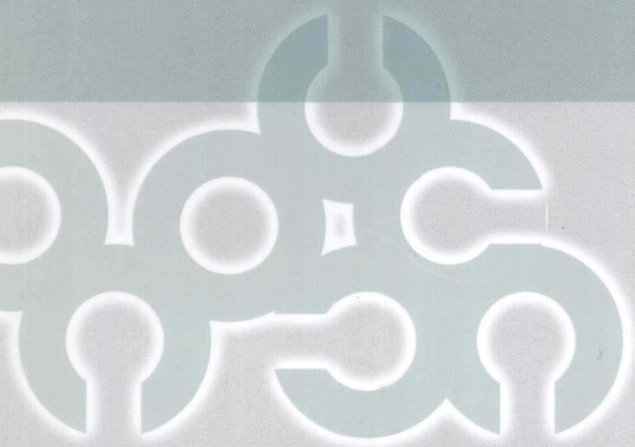
<http://www.phei.com.cn>

欢迎登录 **免费** 获取优质教学资源



www.huaxin.edu.cn
www.hxedu.com.cn

COMPUTER



责任编辑：秦 梅

责任美编：喻 晓



本书贴有激光防伪标志，凡没有防伪标志者，属盗版图书。

ISBN 978-7-121-05810-3



9 787121 058103 >

定价：25.00 元

主教材《汇编语言、微机原理及接口技术（第2版）》

为国家十一五规划教材

《汇编语言、微机原理及接口技术（第2版）》

习题解答与实验指导

郑初华 等编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是《汇编语言、微机原理及接口技术（第2版）》教材的配套习题解答与实验指导，共分五大部分：习题解答、汇编语言实验、微机原理及接口实验、模拟试卷、附录。本书可与主教材配套使用，也可独立使用，还可以作为考研参考书。

本书的习题解答详细，更强调实用性及易懂性，实验内容丰富，可操作性强。配套教材《汇编语言、微机原理及接口技术（第2版）》为“国家十一五规划教材”，2004年荣获江西省首届高校优秀教材一等奖（计算机类第一名）。

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

汇编语言、微机原理及接口技术（第2版）习题解答与实验指导 / 郑初华等编. —北京：电子工业出版社，2008.1

ISBN 978-7-121-05810-3

I. 汇… II. 郑… III. ①汇编语言—程序设计—高等学校—教学参考资料 ②微型计算机—基础理论—高等学校—教学参考资料③微型计算机—接口—高等学校—教学参考资料
IV. TP313 TP36

中国版本图书馆CIP数据核字（2008）第006776号

责任编辑：秦 梅

印 刷：北京民族印刷厂

装 订：北京鼎盛东极装订有限公司

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 印张：18.75 字数：480千字

印 次：2008年1月第1次印刷

印 数：3000册 定价：25.00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 zlt@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

《汇编语言、微机原理及接口技术（第2版）》不但修正了原书中的一些错误，更是吸取众多同行师生的意见，增删了一些内容，特别是增加了许多例子，更强调实用性及易懂性。

该教材由汇编语言、微机原理、接口技术、微机技术新发展及附录五个部分组成。该教材融入多位老师的教学经验，重点突出，详略有序，分类讲解，图表丰富，有一些讲法在其他同类教材中未曾涉及，如快速进制转换、真值与补码直接转换、指令的6个要点等。该教材适合作为理工类本科汇编语言、微机原理、接口技术及其组合课程的教材，也可作为理工类高职高专教材或参考书。

本书是《汇编语言、微机原理及接口技术（第2版）》的配套习题解答与实验指导，共分五大部分：习题解答、汇编实验、微机原理及接口实验、模拟试卷、附录。

本书由郑初华、袁坤主审，赵文龙、韩旭，海霞、周淇等参加编写。由于时间紧，错误在所难免，欢迎各位老师和同学指正。如果读者对本书内容及配套教材有什么问题，欢迎通过电子邮箱 zhengchuhua@126.com、zchzch@sina.com 或 QQ: 85207966 与作者联系。

在此，对曾给本书及配套教材的编写提出意见及参加校稿的张春华、王玉、胡锦涛、柴明钢、石永革、程从从、黎明、何兴道、向瑛、万光逵、黄忠良、黄华、代冀阳、杨小芹、邬冠华、陈敏、杨谊华、陈兵芽、周波、戴仕明、苑鸿冀、王青松、陈强、王勇、黄雷鸣、段玲、张国民、彭玉铃、黄竹生、任吉林、彭应秋、宋凯、杨宇华、温昌举等同志一并表示感谢！

编者
2007年12月

目 录

习题解答	1
第 1 章 进制及码元	1
第 2 章 微机硬件基础	3
第 3 章 寻址方式及指令系统	9
第 4 章 MASM 汇编语言	19
第 5 章 汇编程序设计	23
第 6 章 Intel 8086 / 8088 微处理器	33
第 7 章 内存组成、原理与接口	39
第 8 章 输入输出 (I/O) 系统	43
第 9 章 中断技术	49
第 10 章 可编程接口芯片及其应用	53
第 11 章 总线技术	59
第 12 章 键盘接口	61
第 13 章 显示接口	65
第 14 章 打印机及并口通信技术	75
第 15 章 串口通信	81
第 16 章 微型计算机应用系统的设计	93
汇编语言部分 实验	95
实验一 Turbo Debug 使用简介	95
实验二 DOS 环境下常用命令及 DEBUG 使用方法	109
实验三 DEBUG 或 TD 环境下学习指令的方法	117
实验四 熟悉磁盘结构	121
实验五 基于 TASM 汇编语言完整程序上机过程	125
实验六 基于 MASM 的汇编语言程序上机过程	133
实验七 分支程序设计与调试	137
实验八 菜单程序设计与调试	143
实验九 循环程序设计与调试	149
实验十 子程序设计与调试	153
实验十一 字符串的比较	159
实验十二 综合程序设计与调试	163
实验十三 设计并调试加密程序	167
实验十四 取、置系统日期及时间	171
实验十五 汇编多模块编程	175
实验十六 多语言混合编程	177

微机原理及接口技术部分 实验.....	179
实验十七 利用时钟中断实现倒计时.....	179
实验十八 8259 中断控制器实验.....	181
实验十九 8255 A 并行接口实验 (一)	187
实验二十 8255A 并行口实验 (二)	189
实验二十一 定时器/计数器实验.....	193
实验二十二 模拟钢琴程序.....	195
实验二十三 8255 与 8253 结合实验.....	199
实验二十四 D/A 转换实验.....	201
实验二十五 A/D 转换实验.....	203
实验二十六 8251A 及串行通信发送实验.....	207
实验二十七 串行通信接收实验.....	215
实验二十八 带缓冲区的 PC 查询式串行通信实验.....	219
模拟试卷及答案.....	221
模拟试卷 1.....	221
模拟试卷 1 答案.....	225
模拟试卷 2.....	231
模拟试卷 2 答案.....	233
模拟试卷 3.....	237
模拟试卷 3 答案.....	243
模拟试卷 4.....	247
模拟试卷 4 答案.....	251
模拟试卷 5.....	257
模拟试卷 5 答案.....	263
模拟试卷 6.....	267
模拟试卷 6 答案.....	271
模拟试卷 7.....	275
模拟试卷 7 答案.....	277
附 录.....	281
附录 A 基本 ASCII 码表.....	281
附录 B COM 文件与 EXE 文件.....	283
附录 C DEBUG 命令格式.....	287
附录 D 汇编错误信息中英文对照表.....	289

习题解答

第1章 进制及码元

1. 进制转换

$$129 = \underline{81} \text{ H} = \underline{10000001} \text{ B} = \underline{201} \text{ Q}$$

$$298 = \underline{12A} \text{ H} = \underline{100101010} \text{ B} = \underline{452} \text{ Q}$$

$$1000 = \underline{3E8} \text{ H} = \underline{1111101000} \text{ B} = \underline{1750} \text{ Q}$$

$$5DH = \underline{1011101} \text{ B} = \underline{135} \text{ Q} = \underline{93} \text{ D}$$

$$3E8H = \underline{1111101000} \text{ B} = \underline{1750} \text{ Q} = \underline{1000} \text{ D}$$

$$357Q = \underline{11101111} \text{ B} = \underline{EF} \text{ H} = \underline{239} \text{ D}$$

2. 进制计算

$$101101B + 1101001B = \underline{10010110} \text{ B}$$

$$3FC9H - 0FE6H = \underline{2FE3} \text{ H}$$

一字节的 NOT 8 = F7 H = -9 (有符号数) 位操作, 展成二进制后相操作

一个字的 NOT 8 = FFF7 H = -9 (有符号数)

$$5 \text{ AND } 6 = \underline{4} \text{ D}$$

$$5 \text{ OR } 6 = \underline{7} \text{ D}$$

3. 数据表示范围:

一字节的无符号数表示范围为 0~255, 有符号数 (补码) 表示范围为 -128~+127。

一个字的无符号数表示范围为 0~65 535, 有符号数 (补码) 表示范围为 -32 768~+32 767。

N 位二进制数的无符号数表示范围为 0~(2^N-1), 有符号数 (补码) 表示范围为 -2^{N-1}~+ (2^{N-1}-1)。

4. 35H 代表的 ASCII 字符为 '5', 代表十六进制数时等价的十进制值为 53, 代表压缩 8421BCD 码等价的十进制值为 35, 代表非压缩 8421BCD 码等价的十进制值为 5。

5. FFH 代表无符号数时等价的十进制值为 255, 代表补码有符号数时等价的十进制值为 -1, 代表反码有符号数时等价的十进制值为 -0, 代表原码有符号数时等价的十进制值为 -127。

6. -20 的 8 位二进制补码为 ECH, 原码为 94H, 反码为 EBH。

158 的 16 位二进制补码为 009EH, 原码为 009EH, 反码为 009EH。

7. 英文字符一般在计算机内占用 1 字节, 每个字节的最高位一定为 0。全角英文字符在计算机内占用 2 字节, 一个汉字在计算机内占用 2 字节, 每字节最高位一定为 1。

8. 设阶码用 8 位补码表示, 尾数用 16 位补码表示: 则 $-(1/32 + 1/128 + 1/512)$ 的尾数部分及阶码分别为多少。

答: 尾数为 $-0.000010101B$, 修正为纯小数, 即 $-0.10101B$, 则指数为 -4 。

所以尾数的补码为 $1010110000000000B$, 指数的补码为 $11111100B$ 。

第 2 章 微机硬件基础

1. 计算机系统硬件的五大功能部件及其主要功能？

答：硬件系统按功能模块分为运算器、控制器、存储器、输入设备、输出设备。

运算器主要完成算术运算、逻辑运算及移位运算，其中主要包括算术逻辑运算单元 (Arithmetic Logic Unit, ALU) 和暂存数据 (原始、中间、结果) 的寄存器 (Register, R)。

控制器实现对计算机部件的协调调度完成指令的分析执行，其中主要有程序计数器 (Program counter, PC) 或叫指令指针 (Instruction Pointer, IP)、指令寄存器 (Instruction register, IR)、指令译码器 (Instruction Decoder, ID)、微操作命令发生器、时序发生器等。

存储器用来记忆程序与数据，它分为内存和外存，内存主要用于存放正在或正待运行的程序和数据，外存用于存放暂时不运行的程序和数据。

输入设备是实现将外界的信息输入计算机内部的部件，主要有键盘、鼠标等。

输出设备是实现将计算机内部的信息输出的部件，主要有显示器、打印机等。

2. 请画出计算机系统硬件框图。

答：如图 2.1 所示。

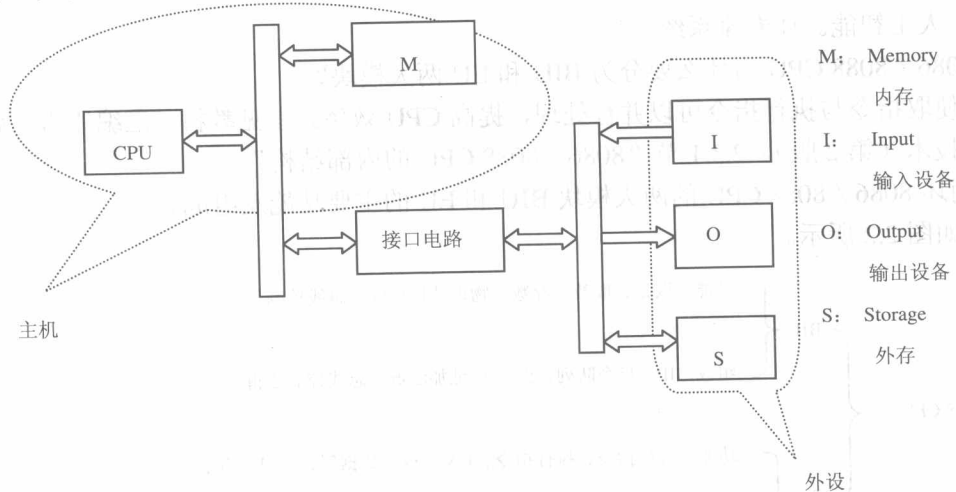


图 2.1 计算机硬件结构图

3. 请简述计算机系统的构成，并说明软硬件之间的关系。

答：计算机系统由硬件系统和软件系统构成，硬件是我们可以看得见摸得着的物理电子设备，软件是运行于计算机硬件基础上的程序和数据的集合。它们之间的关系是：硬件是基础，软件是发展；硬件和软件相辅相成缺一不可；硬件和软件互相促进发展。

4. 1946年，第一台电子计算机 ENIAC 诞生；1969年，Internet 的雏形 APARNET 诞生；1971年，第一块微处理器芯片 Intel 4004 诞生；1981年，第一台 IBM PC 诞生。

5. 微机的主要性能指标有哪些？

答：

(1) 字长。字长是指 CPU 可以同时运算或传输的二进制的数据位数。字长越长，速度越快，精度越高，表示范围越大。

(2) 速度。在微机的表示速度用的最多的是 CPU 的主频，单位为 MHz 或 GHz，但实质上并不一定是主频越高速度越快，如 AMD 提出的 PR 表示法声称其 1.3GHz 的速龙 XP 可以达到 1.6GHz 的 P4 的速度。表示速度还有另外两个单位：次/秒（每秒运算次数）、MIPS（百万条指令每秒）。

(3) 容量。内存容量从原来的 KB 已上升到 MB，2002 年上半年以 256MB 为主流配置。硬盘容量从原来的 MB 已上升到 GB，目前主要配置 40GB~100GB 的硬盘。

(4) 外设配置。例如显示器、光驱、声卡、音箱、打印机等。

(5) 软件配置。操作系统及其他系统软件和应用软件。

6. 计算机主要有哪些应用领域？并举例说明。

答：因为微机具有体积小、价格低、功耗小、可靠性高等优点，所以它的应用几乎深入到生产与生活的各个领域。主要应用有如下几个方面：

(1) 科学与工程计算。

(2) 数据与信息处理。

(3) 工业过程控制及智能化仪器仪表电器等自动化控制。

(4) 计算机辅助。如 CAD、CAM、CAT、CAI 等。

(5) 网络通信技术。如 Internet 及网上办公系统。

(6) 游戏及计算机仿真。

(7) 人工智能。如专家系统。

7. 8086 / 8088 CPU 为什么要分为 BIU 和 EU 两大模块？

答：使取指令与执行指令可以并行处理，提高 CPU 效率。参见教材《汇编语言、微机原理及接口技术（第 2 版）》2.3.1 节“8086 / 8088 CPU 的内部结构”。

8. 简述 8086 / 8088 CPU 的两大模块 BIU 和 EU 的主要功能及组成。

答：如图 2.2 所示。

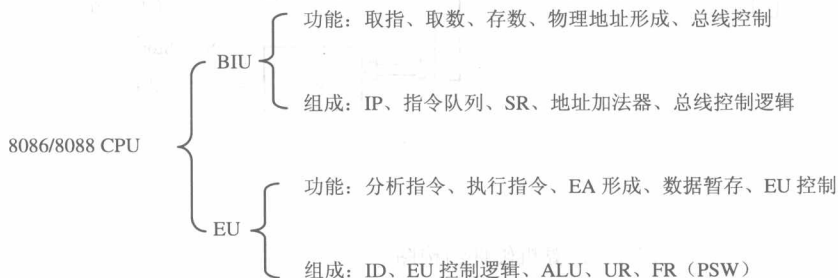


图 2.2 8086/8088 CPU 的两大模块

9. 简述 8086 / 8088 CPU 的 14 个寄存器的英文名、中文名及主要作用。

答：

AX (Accumulator) (AH、AL) 累加器，它是汇编编程中最常用的一个寄存器，主要用于乘除运算、BCD 运算、换码、I/O 操作、串操作、中断调用等。

BX (Base) (BH、BL) 基址寄存器，主要用于存放地址、基址（默认相对于 DS 段）等。

CX (Counter) (CH、CL) 计数器，主要用于循环计数、串操作计数、移位计数 (CL) 等。

DX (Data) (DH、DL) 数据寄存器，主要用于 16 位乘除、间接 I/O、中断调用等。

BP (Base Pointer) 基址指针，主要用于存放地址、基址（默认相对于 SS 段）等。

SP (Stack Pointer) 堆栈指针（栈顶指针），主要用于存放栈顶地址。

SI (Source Index) 源变址寄存器，用于存放地址、变址、串操作源变址。

DI (Destination Index) 目的变址寄存器，用于存放地址、变址、串操作目的变址。

CS (Code Segment) 代码段寄存器（代码段），用于存放正在或正待执行的程序段的起始地址的高 16 位二进制数据，即程序段的段地址。

DS (Data Segment) 数据段寄存器（数据段），用于存放正在或正待处理的一般数据段的起始地址的高 16 位二进制数据，即一般数据段的段地址。

ES (Extra Segment) 附加数据段寄存器（附加段），用于存放正在或正待处理的附加数据段的起始地址的高 16 位二进制数据，即附加数据段的段地址。

SS (Stack Segment) 堆栈数据段寄存器（堆栈段），用于存放正在或正待处理的堆栈数据段的起始地址的高 16 位二进制数据，即堆栈数据段的段地址。

IP (Instruction pointer) 指令指针，它的内容始终是下一条待执行指令的起始偏移地址，与 CS 一起形成下一条待执行指令的起始物理地址。CS:IP 的作用是控制程序的执行流程。IP 一般会自动加 1（逻辑加 1、实际随指令长度变化）移向下一条指令实现顺序执行；若通过指令修改 CS 或 IP 的值，则可实现程序的转移执行。

PSW (Program Status Word) 程序状态字，它其中有三个控制标志（IF、DF、TF）和 6 个状态标志（SF、PF、ZF、OF、CF、AF）。控制标志是用于控制 CPU 某方面操作的标志，状态标志是部分指令执行结果的标志。

10. 请画出 8086 / 8088 CPU 的内部结构图。

答：如图 2.3 所示。

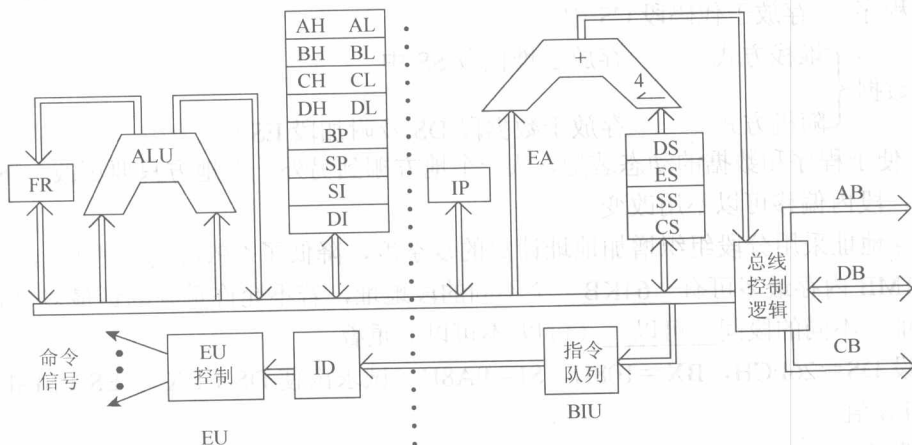


图 2.3 8086 / 8088 CPU 的内部结构图

11. 请说明 8086/8088 CPU 的标志位的英文名、中文名及通用填充方法。

答：

IF (Interrupt enable Flag) 中断允许标志，用于控制 CPU 能否响应可屏蔽中断请求，IF=1 能够响应，IF=0 不能响应。

DF (Direction Flag) 方向标志，用于指示串操作时源串的源变址和目的串的变址的变化方向，DF=1 向减的方向变化，DF=0 向加的方向变化。

TF (Trap Flag) 陷阱标志 (单步中断标志), TF=1 程序执行当前指令后暂停, TF=0 程序执行当前指令后不会暂停。

SF (Sign Flag) 符号标志, 指令执行结果的最高二进制位是 0 还是 1, 为 0, 则 SF=0, 代表正数; 为 1, 则 SF=1, 代表负数。我们一般是用十六进制数表示, 则可以根据十六进制的最高位是落在 0~7 还是落在 8~F 之间, 若落在 0~7 之间则 SF=0, 否则 SF=1。

PF (Parity check Flag) 奇偶校验标志, 指令执行结果的低 8 位中 1 的个数是奇数个还是偶数个, 若为奇数个则 PF=0, 若为偶数个则 PF=1。

ZF (Zero Flag) 零标志, 指令执行结果是不是为 0, 若为 0 则 ZF=1, 否则 ZF=0。

OF (Overflow Flag) 有符号数的溢出标志, 指令执行结果是否超出有符号数的表示范围, 若超过则 OF=1, 否则 OF=0。我们可以通过判断是否出现以下四种情况之一: 正加正得负, 正减负得负, 负加负得正, 负减正得正。若出现则 OF=1, 否则 OF=0。

CF (Carry Flag) 进位/借位标志 (无符号数的溢出标志), 指令执行结果的最高位是否有向更高位进位或借位, 若有则 CF=1, 同时也代表无符号数溢出; 若无则 CF=0, 也代表无符号数未溢出。

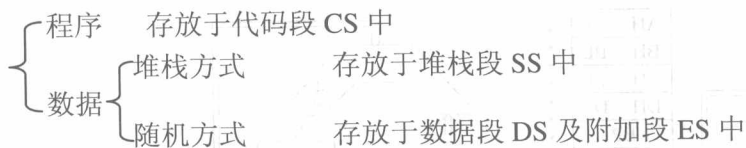
AF (Auxiliary carry Flag) 辅助进位/借位标志, 低 4 位二进制是不是有向高位进位或借位, 若有则 AF=1, 否则 AF=0, 其主要用于 BCD 修正运算。

12. 内存为什么要分段组织?

答:

(1) 8086/8088 CPU 中的寄存器只有 16 位, 如果采用它们直接寻址, 则寻址能力势必限制在 64 KB 范围内, 而采用分段组织可以较好地实现扩展 CPU 的寻址能力。每段的大小可达 64 KB, 不同段的组合则可寻址更大的范围。

(2) 使程序与数据相对独立, 不同存取方式的数据也相对独立。



(3) 便于程序和数据的动态装配, 从一个地方挪到另外一个地方只要更改一下段寄存器的值即可, 段内偏移可以不用改变。

但内存地址采用分段组织增加地址计算的复杂度, 降低了 CPU 的执行效率。

13. 1MB 内存最多可有 64KB 个不同的段地址, 若不允许重叠的话最多可有 16 个不同的段地址。不同的段间 可以 (可以/不可以) 重叠。

14. 设 DS=26FCH, BX=108H, SI=9A8H, 试求出使 DS : BX 与 ES : SI 指向同一物理地址的 ES 值。

答: 即 DS : BX = ES : SI

$$DS : BX = DS \times 16 + BX = 26FCH \times 16 + 108H = 270C8H$$

$$ES : SI = ES \times 16 + SI = ES \times 16 + 9A8H$$

$$ES = (270C8H - 9A8H) \div 16 = 2672H$$

15. 接口、端口、端口地址之间的对应关系如何?

答: 一个接口内一般有多个端口, 一个端口可以有多个端口地址与其对应 (这叫重叠地址), 一个端口地址可以对应多个端口 (但需要继续区分才能访问, 如读写顺序、读写信号、

数据位、索引区分)。

16. 访问端口有哪些方法?

答: 在汇编语言中用专用的 I/O 指令实现信息传输, 更多的是通过 BIOS 中断调用或 DOS 中断调用实现信息传输。BIOS 中断调用是更低级的中断调用, 它是与硬件相关的, 兼容性较差, 但速度较快; DOS 中断调用是较高级的中断调用, 它与硬件无直接关系, 兼容性较好, 但速度较慢, 大多数情况下其内部会再调用 BIOS 中断调用。

17. 请根据教材《汇编语言、微机原理及接口技术(第2版)》图 2.4 说明 ADD AX, BX (将 AX+BX 的值送 AX) 指令的取指及执行过程。

答: 取指: ①由 CS:IP 形成取指的物理地址; ②CPU 将此地址送地址总线 AB; ③译码选中内存单元; ④CPU 发出取指信号; ⑤内存中指令送上数据总线 DB; ⑥CPU 将读入的指令存入指令队列; ⑦IP=IP+1, 移向下一条指令。

执行指令: ①指令队列中的指令送指令译码器译码; ②译码后执行单元发出相关命令信号完成指令的执行, 即将 AX 送 ALU 的一端、BX 送 ALU 的另一端, 并完成相加结果送 AX, 而且根据结果填充标志位。

18. 8086 / 8088 的 20 位物理地址是怎样形成的? 当 CS=2000H, IP=0100H, 下一条待执行指令的物理地址等于多少?

答: 物理地址=段的起始地址+段内偏移地址=段地址×16+EA

$$\begin{aligned} \text{下一条待执行指令的物理地址} &= \text{CS} : \text{IP} = \text{CS} \times 16 + \text{IP} \\ &= 2000\text{H} \times 10\text{H} + 0100\text{H} = 20100\text{H} \end{aligned}$$

19. 已知当前数据段位于存储器的 B1000H 到 BF0FFH 范围内, 请指出 DS 段寄存器的取值范围。

答: 段内偏移介于 0~FFFFH, 所以最大的 DS 值为 B100H, 最小的为 (BF0FFH-FFFFH)÷16=AF10H, 所以 DS 段寄存器的取值范围为 AF10H~B100H。(求法: 最小的段值为最大地址-FFFFH 再除以 10H, 最大的段值为最小地址除以 10H 即可)

20. 某程序在当前数据段中存放有两个数据 1234H 和 5678H, 若已知 DS=2010H、偏移量分别为 3400H 和 4501H, 请写出这两个数据在存储器中的物理地址。

答: 物理地址 1=DS×16+EA=2010H×16+3400H=23500H

23500H 单元存 34H, 23501H 单元存 12H

物理地址 2=DS×16+EA=2010H×16+4501H=24601H

24601H 单元存 78H, 24602H 单元存 56H

第3章 寻址方式及指令系统

1. 指令是 CPU 可以理解并执行的操作命令, 指令由 操作码 和 操作数 两部分组成, 指令有两个级别, 即 机器级 和 汇编级。

2. 请画出程序执行流程示意图。

答: 程序执行流程示意图如图 3.1 所示。

3. 指令主要有哪些分类方式及其主要类别是什么?

答: 指令分类有不同的方式。按操作数的个数将指令分为零地址指令(无操作数)、一地址指令(只有一个操作数)、二地址指令(有两个操作数)、多地址指令(多于两个操作数); 按指令级别分为机器指令和汇编指令; 按是否转移分为转移指令和顺序指令; 按功能分为七大类: 传送类、算术运算类、位操作类、I/O 类、串操作类、CPU 控制类、转移类; 按指令长度(即指令占用的字节数)分为 1 字节指令、2 字节指令等, 8086/8088 指令长度是不同的, 叫变字长, 不同的指令有不同的指令长度, 从 1~6 字节均有; 按指令执行期间取操作数是否需要与总线打交道分为内部指令(不需)和外部指令(需要), 类似功能的指令内部指令比外部指令执行速度快, 所以在编程时尽量采用内部指令即多用寄存器优化程序。

4. 操作数的寻址方式有哪些? 并举例说明其主要特点。

答: 有立即寻址、寄存器寻址、存储器寻址、端口寻址, 具体参见教材《汇编语言、微机原理及接口技术(第2版)》3.2 节“寻址方式”。

5. 内存寻址中段寄存器与段内偏移地址的对应关系如何?

答: 内存寻址中段寄存器与段内偏移地址的对应关系如表 3.1 所示。

表 3.1 内存地址对应的 SR 及 EA

情况	段寄存器 SR	段内偏移 EA
指令	CS	IP
栈顶	SS	SP
目的串	ES	DI
源串	DS (ES、SS、CS)	SI
涉及 BP	SS (DS、ES、CS)	计算 EA
其他	DS (ES、SS、CS)	计算 EA

6. 设 CS=1000H, DS=2000H, ES=3000H, SS=4000H, IP=100H, SP=200H, BX=300H, BP=400H, SI=500H, 则①下一条待执行指令的物理地址为多少? ②当前栈顶的物理地址为多少? ③[BX]代表的存储单元的物理地址为多少? ④[BP]代表的存储单元的物理地址为多少? ⑤ES:[BX+SI]代表的存储单元的物理地址为多少?

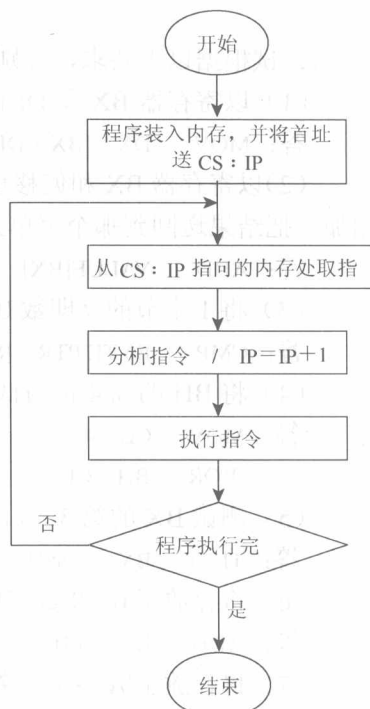


图 3.1 程序执行流程图

- 答：① 下一条待执行指令的物理地址 = $CS \times 16 + IP = 1000H \times 16 + 100H = 10100H$ 。
 ② 当前栈顶的物理地址 = $SS \times 16 + SP = 4000H \times 16 + 200H = 40200H$ 。
 ③ [BX] 代表的存储单元物理地址 = $DS \times 16 + BX = 2000H \times 16 + 300H = 20300H$ 。
 ④ [BP] 代表的存储单元物理地址 = $SS \times 16 + BP = 4000H \times 16 + 400H = 40400H$ 。
 ⑤ ES : [BX+SI] 代表的存储单元物理地址 = $ES \times 16 + BX + SI$
 $= 3000H \times 16 + 300H + 500H = 30800H$

7. 试根据以下要求，分别写出相应的汇编语言指令。

(1) 以寄存器 BX 和 DI 作为基址变址寻址方式把存储器中的一个字送到 DX 寄存器。

答：MOV DX, [BX+DI]

(2) 以寄存器 BX 和偏移量 VALUE 作为寄存器相对寻址方式把存储器中的一个字和 AX 相加，把结果送回到那个字单元。

答：ADD VALUE[BX], AX

(3) 将 1 字节的立即数 0B6H 与以 SI 作为寄存器间接寻址方式的字节单元相比较。

答：CMP BYTE PTR [SI], 0B6H

(4) 将 BH 的高 4 位与低 4 位互换。

答：MOV CL, 4

ROR BH, CL ; 或者 ROL BH, CL

(5) 测试 BX 的第 3、7、9、12、13 位是否同时为 0。

答：TEST BX, 3288H ; 0011001010001000B

(6) 将存放了 0~9 数值的 DL 寄存器中的内容转化为相应的 '0'~'9' 的字符。

答：ADD DL, 30H ; 或者 OR DL, 30H

(7) 将存放了 'A'~'F' 字符的 AL 寄存器中的内容转化为相应的数值。

答：SUB AL, 37H

8. 写出清除 AX 寄存器的多种方法并比较。(要求单指令实现)

答：MOV AX, 0 ; 不会影响标志位

SUB AX, AX

AND AX, 0

XOR AX, AX ; 使用最为频繁

9. 分别用存储器的五种寻址方式实现将以 A 为首址的第 5 个字 (注意：从第 0 个起算)，送 AX 的指令序列。

答：直接寻址：MOV AX, A+5*2

间接寻址：LEA BX, A+5*2

MOV AX, [BX]

相对寻址：MOV SI, 5*2

MOV AX, A[SI]

相对寻址：LEA BX, A

MOV AX, [BX+5*2]

基址变址：LEA BX, A

MOV SI, 5*2

MOV AX, [BX+SI]