



21 世纪高等院校规划教材

微机原理与接口技术 习题解答及实验指导

THE PRACTICAL MANUAL OF MICROCOMPUTER PRINCIPLE AND INTERFACE TECHNOLOGY

于天河 蒋少禹 编著
辛士光 主审

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE



21 世纪高等院校规划教材

微机原理与接口技术 习题解答及实验指导

编著 于天河 蒋少禹

主审 辛士光

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书是《微机原理与接口技术》(于天河、高爽主编,以下简称主教材)的配套教材。本书共分5篇,主要内容包括教学内容指导(包括知识点回顾与主教材习题答案)、综合模拟试题及参考答案、汇编语言实验环境及上机过程、汇编语言程序设计实验指导、硬件接口电路实验指导。

本书习题丰富,便于学生复习和自学;实验部分脱离了固定的实验箱限制,可操作性强。

本书适合作为高等学校微机原理与接口技术等相关课程的配套教材和参考用书。

图书在版编目(CIP)数据

微机原理与接口技术习题解答及实验指导/于天河,
蒋少禹编著. —北京:中国铁道出版社,2011.8

21世纪高等院校规划教材

ISBN 978-7-113-13352-8

I. ①微… II. ①于… ②蒋… 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—高
等学校—教学参考资料 IV. ①TP36

中国版本图书馆CIP数据核字(2011)第159238号

书 名:微机原理与接口技术习题解答及实验指导

作 者:于天河 蒋少禹 编著

策划编辑:吴宏伟 孟 欣

读者热线:400-668-0820

责任编辑:徐盼欣

封面设计:付 巍

编辑助理:路 迪

封面制作:白 雪

责任印制:李 佳

出版发行:中国铁道出版社(北京市宣武区右安门西街8号 邮政编码:100054)

印 刷:三河市华丰印刷厂

版 次:2011年8月第1版 2011年8月第1次印刷

开 本:787mm×1092mm 1/16 印张:10.5 字数:250千

印 数:1~3 000册

书 号:ISBN 978-7-113-13352-8

定 价:18.00元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材研究开发中心批销部联系调换。

随着微型计算机应用的日益广泛和深入,接口技术有了迅速的发展,并已成为直接影响微机系统功能和推广应用的关键。从硬件的角度来看,微机的开发与应用,在很大程度上就是微机接口电路的开发与应用。

实验室及实验仪器的综合和开放是当前实验室建设的方向,培养学生的研究和创新能力是实验室的重要任务。微型计算机接口技术是深入学习和利用计算机、设计和开发各种微机应用系统的基础,是微机应用的关键,也是微机应用中最困难的地方。它不仅要求设计者具有微机关、硬件方面的基础,而且要求设计者具有比较强的接口技术能力。本书为微机的应用提供接口技术的基本方法和使用技巧,尤其是为工程设计应用系统打好基础,使读者能够比较容易地掌握接口技术的基本内容和设计方法。

本书是《微机原理与接口技术》(于天河、高爽主编)的配套教材,本书共分5篇,主要内容包括教学内容指导(包括知识点回顾与主教材习题答案)、综合模拟试题及参考答案、汇编语言实验环境及上机过程、汇编语言程序设计实验指导、硬件接口电路实验指导等。

本课程的教学主旨在于使学生掌握一种方法,起到抛砖引玉、触类旁通的作用。本书配合接口技术课程同步教学,学生可以自己编程动手练习,进行一些开发性实验,这样效果更佳。微机原理与接口技术实验是数字电路的后续课程,建议学生在掌握数字电路和微机课程后再修读此课程。

本书由于天河、蒋少禹编著,辛士光主审。书中习题部分由于天河编写,实验部分由蒋少禹编写。在编写本书的过程中参考了相关文献,在此向这些文献的作者深表感谢。

书中难免有错误与不足之处,恳请专家和广大读者批评指正。

编 者

2011年6月

第一篇 教学内容指导

第 1 章	概述	1
1.1	知识点回顾	1
1.2	主教材习题答案	5
第 2 章	微处理器系统	8
2.1	知识点回顾	8
2.2	主教材习题答案	10
第 3 章	寻址方式和指令系统	13
3.1	知识点回顾	13
3.2	主教材习题答案	19
第 4 章	汇编语言程序设计	23
4.1	知识点回顾	23
4.2	主教材习题答案	25
第 5 章	存储器系统	34
5.1	知识点回顾	34
5.2	主教材习题答案	35
第 6 章	输入/输出和总线技术	40
6.1	知识点回顾	40
6.2	主教材习题答案	42
第 7 章	中断	45
7.1	知识点回顾	45
7.2	主教材习题答案	47
第 8 章	常用数字接口芯片	51
8.1	知识点回顾	51
8.2	主教材习题答案	53
第 9 章	A/D 和 D/A 转换	60
9.1	知识点回顾	60
9.2	主教材习题答案	62

第二篇 综合试题及参考答案

综合试题 1	65
综合试题 2	70
综合试题 3	76
综合试题 4	79
综合试题 1 参考答案	84
综合试题 2 参考答案	87
综合试题 3 参考答案	91
综合试题 4 参考答案	93

第三篇 汇编语言实验环境及上机过程

第 1 章 汇编语言程序设计的实验环境和上机过程简介	96
第 2 章 应用案例	97
第 3 章 开发环境所需文件功能详解	99

第四篇 汇编语言程序设计实验指导

实验 1 TD 调试程序的使用及寻址方式实验	106
实验 2 算术运算、逻辑运算和移位指令实验	108
实验 3 串操作指令实验	110
实验 4 顺序程序结构设计实验	112
实验 5 两个数相乘实验	113
实验 6 分支及循环程序设计实验	115
实验 7 排序实验	117
实验 8 综合设计实验	119

第五篇 硬件接口电路实验指导

实验 1 8255 可编程并行接口实验	124
实验 2 8253 可编程定时器/计数器实验	127

实验 3	8251 可编程串行接口实验	130
实验 4	8259 可编程中断控制器实验	135
实验 5	存储器扩展实验	140
实验 6	D/A 转换接口实验	143
实验 7	A/D 转换接口实验	146
实验 8	步进电机控制实验	150
附录 A	TD 使用说明	155
参考文献		160

第一篇 教学内容指导

第 1 章 | 概 述

1.1 知识点回顾

1. 微处理器的发展历史

微机采用了集成度相当高的器件和部件，它的核心部分是微处理器。微处理器是指一片或几片大规模集成电路组成的，具有运算器和控制器功能的中央处理器（CPU）。

- （1）第一代 4 位或低档 8 位微处理器；
- （2）第二代中高档 8 位微处理器；
- （3）第三代 16 位微处理器；
- （4）第四代 32 位微处理器；
- （5）第五代高档 32 位微处理器；
- （6）第六代微处理器。

2. 计算机的常用术语

（1）字长

字（word）是计算机一次处理的指令和数据的基本单位，一般由若干个字节组成。

（2）主频

CPU 的主频（main frequency）即 CPU 内核工作的时钟频率。

（3）运算速度

单字长定点指令平均执行速度（MIPS），即每秒处理的百万条的机器语言指令数。

（4）存储容量

主存容量（memory）是指主存储器能够存储信息的总字节数。

（5）位

位（bit）是计算机能处理的最小数据单位，表示二进制中的 1 位。

（6）字节

计算机的存储器存储容量很大，里面有着数量庞大的存储单元。8 位二进制数表示为一个

字节 (byte, 简称为 B)。常用的存储容量单位之间的换算公式:

$$1 \text{ B}=8 \text{ bit} \quad 1 \text{ KB}=2^{10} \text{ B}=1\,024 \text{ B} \quad 1 \text{ MB}=2^{10} \text{ KB}=1\,024 \text{ KB}$$

$$1 \text{ GB}=2^{10} \text{ MB}=1\,024 \text{ MB} \quad 1 \text{ TB}=2^{10} \text{ GB}=1\,024 \text{ GB}$$

3. 计算机系统的组成

冯·诺依曼提出的计算机设计思想, 可以简要地概括为以下 3 点:

- ① 计算机应包括运算器、存储器、控制器、输入设备和输出设备五大基本部件。
- ② 计算机内部应采用二进制来表示指令和数据。每条指令一般具有一个操作码和一个地址码。
- ③ 采用程序存储的方式, 指令自动执行。计算机无须操作人员干预, 能自动逐条取出指令和执行指令。

计算机系统的组成如图 1-1 所示, 其中软件系统由系统软件和应用软件组成。



图 1-1 计算机系统组成

下面对计算机硬件系统各部件进行具体介绍。

(1) 运算器

运算器主要由算术逻辑单元 (Arithmetic Logic Unit, ALU) 和一些寄存器组构成, 它的功能就是进行算术运算和逻辑运算。

(2) 控制器

控制器是计算机的指挥中心。一般由指令寄存器、指令译码器、时序部件和控制电路等组成。控制器和运算器合在一起被称为中央处理器单元, 通常称做 CPU (Central Processing Unit)。CPU 是计算机的核心部件。

(3) 存储器

存储器是计算机用来存储程序 and 数据的设备, 由一系列的存储单元组成。每个存储单元按顺序进行编号, 这种编号称为存储单元的地址。

(4) 输入设备

输入设备用来将外部数据 (如文字、数值、声音、图像等) 转变为计算机可识别的形式 (二进制代码), 输入到计算机中, 以便加工、处理。最常用的输入设备是键盘。

(5) 输出设备

输出设备的作用是将计算机处理的结果用人们能接受的形式（如字符、图像、语音、视频等）表示出来。

4. 计算机的工作过程

CPU 的主要工作原理是执行程序里的一系列指令，在此讨论的是普遍意义上的原理。程序以一系列二进制数在存储器中。CPU 的工作原理可归纳为 4 个阶段：取指令、分析指令、执行指令和写回结果。

① 提取指令，从存储器中检索指令，即一系列二进制数。由程序计数器指定程序存储器的位置，程序计数器保存供识别目前程序位置的数值。程序计数器记录了 CPU 在目前程序里的踪迹。

② 分析指令，指令被拆解为有意义的片断。根据 CPU 的指令集定义将信息解译为指令。一部分的指令数值为运算码，其指示要进行哪些运算。

③ 执行指令，如要做一个加法运算，算数逻辑单元将会连接到一组输入和一组输出。输入提供了要相加的数值，而且在输出将含有总和结果。

④ 写回结果，以一定格式将执行阶段的结果简单的写回。

指令执行过程如下：

① 程序启动之后，程序计数器 PC 把指令地址发给地址寄存器，将指令从内存中取出并送到指令寄存器（Instruction Register, IR）中。

② 将取出的指令送到指令译码器，以确定要进行的操作。

③ 读取相应的操作数，即操作对象。

④ 执行指令。

⑤ 存放执行结果。

⑥ 一条指令执行完成后，程序计数器 PC 自身加 1，以转入下一条指令的读取阶段。这样的循环直到程序遇到了停止指令才结束。

5. 计算机的信息表示

计算机使用二进制进行编码，而不是人们熟悉的十进制，最重要的原因是二进制在物理上更容易实现，因为电子器件大多具有两种稳定状态。比如晶体管的导通和截止、电压的高和低、磁性的有和无等。而找到一个具有 10 个稳定状态的电子器件是很困难的。使用二进制还有运算简单的优点。

(1) 采取二进制的原因

在计算机内部，信息的表示依赖于机器硬件电路的状态，信息采用什么表示形式，直接影响到计算机的结构与性能。采用二进制表示信息，有以下几个优点：

① 易于物理实现。因为具有两种稳定状态的物理器件是很多的，如门电路的导通与截止，电压的高与低，而它们恰好对应表示 1 和 0 两个符号。

② 运算简单。数学推导证明，进行算术求和或求积运算，对于二进制运算规则仅有 3 种，因而简化了运算器等物理器件的设计。

③ 机器可靠性高。由于电压的高低、电流的有无等都是两种分明的状态，因此二进制的

传递抗干扰能力强, 鉴别信息的可靠性高。

④ 通用性强。二进制适用于各种非数值信息的数字化编码。特别是 0 和 1 两个符号正好与逻辑命题的“真”与“假”两个值相对应, 从而为计算机实现逻辑运算和逻辑判断提供了方便。

(2) 各种数制之间的转换

① 非十进制数到十进制数的转换。按相应进位计数制的权表达式展开, 再按十进制求和。

② 十进制数转换为二进制:

- 十进制转换成二进制:

整数部分: 除 2 取余; 小数部分: 乘 2 取整。

- 十进制转换成十六进制:

整数部分: 除 16 取余; 小数部分: 乘 16 取整。

以小数点为起点求得整数和小数的各个位。

③ 二进制与八进制数之间的转换。

用 3 位二进制数表示 1 位八进制数。

④ 二进制与十六进制之间的转换。

十六进制的两个基本特点是: 有 16 个基本数字, 逢十六进一。

十六进制的前 10 位基本数字是从十进制借过来的, 即 0~9。其余 6 个基本数字, 则是用字母来表示, 用 A~F 分别来表示 10~15。

由于 $2^4=16$, 所以二进制与十六进制之间的转换也和八进制类似, 只不过 1 位十六进制数字恰好和 4 位二进制数字相对应, 如十六进制数 A 对应的是二进制数 1010。

6. 计算机的数字编码

(1) 原码

在表示带符号数 0 时, 正数的符号位为 0, 负数的符号位为 1, 数值位表示数的绝对值, 这样就得到了数的原码表示形式。原码的 0 有两种不同的表示形式, 即 +0 和 -0。 $[+0]_{\text{原}}=00000000$, $[-0]_{\text{原}}=10000000$ 。原码表示简单易懂, 与真值之间的转换较为方便。在 8 位微型计算机中, 原码可表示数的范围为 $-127 \sim +127$ 。

(2) 反码

正数的反码表示与原码相同; 负数的反码, 可将负数原码的符号位保持不变、数值位按位取反得到, 或者将负数看作正数求原码, 再将所有位按位取反得到。8 位反码表示数的范围与原码相同, 8 位二进制反码表示的范围仍是 $-127 \sim +127$ 。

(3) 补码

引进补码就是为了把减法运算转换成容易实现的加法运算。一个负数 x 的补码等于其原码中除符号值保持不变外, 其余各位按位求反, 再在最低位加 1。 $[+0]_{\text{补}}=[-0]_{\text{补}}=00000000$, 8 位二进制补码表示的范围是 $-128 \sim +127$, 在微型计算机中所有有符号数都是由补码表示的, 所以结果由补码表示。

7. 二进制数的计算及编码

二进制数表示分为无符号数和有符号数两种。无符号数不考虑数据的符号, 数中的每一位 0 和 1 都是有效数据。有符号数则不同, 在最高位的 0 或 1 表示的是数据的正负。

(1) 数的计算

在计算机系统中, 数值一律用补码来表示(存储)。其主要原因是, 使用补码, 可以将符号位和其他位统一处理; 同时, 减法也可按加法来处理。另外, 两个用补码表示的数相加时, 如果最高位(符号位)有进位, 则进位被舍弃。补码与原码的转换过程几乎是相同的。

补码的运算规则如下。

补码加法的运算规则: $[X+Y]_{\text{补}} = [X]_{\text{补}} + [Y]_{\text{补}}$

补码减法的运算规则: $[X-Y]_{\text{补}} = [X]_{\text{补}} - [Y]_{\text{补}} = [X]_{\text{补}} + [-Y]_{\text{补}}$

有符号数的范围:

对于 8 位二进制数, 原码、反码和补码能表示的范围分别如下:

原码 1111111B~01111111B (-127~+127);

反码 10000000B~01111111B (-127~+127);

补码 10000000B~01111111B (-128~+127)。

对于 16 位二进制数, 原码、反码和补码所能表示的范围分别如下:

原码 FFFFH~7FFFH (-32 767~+32 767);

反码 8000H~7FFFH (-32 767~+32767);

补码 8000H~7FFFH (-32 768~+32 767);

无符号数的范围:

对于 8 位二进制数其范围为 00000000B~11111111B (0~255); 对于 16 位二进制数其范围为 0000H~FFFFH (0~65535)。

(2) 二进制数计算的溢出问题

对于无符号数的加减运算, 若最高有效位向更高位有进位或有借位 ($CF=1$), 则产生溢出。

两个有符号数相加减时, 如果运算结果超出有符号数表示的范围时, 就会发生溢出错误。溢出出现在同符号数相加或异符号数相减的情况下, 判断有符号数运算溢出的情况如下:

①两个有符号数运算时, 结果的次高位向最高位有进位或借位, 而最高位向上无进位, 则结果产生溢出。

②如果次高位向最高位无进位或借位, 而最高位向上有进位, 则结果也产生溢出。可以归纳为, 最高位和次高位如果一个有进位一个无进位, 结果即产生溢出 ($OF=1$)。

(3) ASCII 码

ASCII 码为美国标准信息交换码。它是目前最普遍使用的字符编码, 包含了日常应用的大多数常用字符。其中除了数字、字母这样的可打印字符外, 还有 33 种控制字符, 如回车符、换行符等。

1.2 主教材习题答案

一、选择题

1. 最早的计算机是用来进行 A 的。

A. 科学计算

B. 系统仿真

C. 自动控制

D. 信息处理

2. 在计算机中采用二进制是因为 D。
 - A. 电子元件只有两个状态
 - B. 二进制的运算简单
 - C. 系统部件少, 可以增加稳定性
 - D. 以上三个原因
3. 存储器的存储单位是 B。
 - A. 比特 (bit)
 - B. 字节 (byte)
 - C. 字 (word)
 - D. 字符 (character)
4. 内存存储器中的每个存储单元都被赋予一个唯一的序号, 称为 D。
 - A. 单元号
 - B. 下标
 - C. 编号
 - D. 地址
5. 下列数中最小的一个是 A。
 - A. $(100)_2$
 - B. $(8)_{10}$
 - C. $(12)_{16}$
 - D. $(11)_8$
6. 已知小写字母的 ASCII 码值比大写字母大 32, 而大写字母 A 的 ASCII 码为十进制数 65, 则小写字母 d 的 ASCII 码是二进制数 A。
 - A. 1100100
 - B. 1000100
 - C. 1000111
 - D. 1110111
7. 计算机能直接执行的程序设计语言是 D。
 - A. C
 - B. BASIC
 - C. 汇编语言
 - D. 机器语言

二、填空题

1. $11001010B = \underline{202} \quad D = \underline{312} \quad O = \underline{0CA} \quad H$ 。
2. $10111.1101B = \underline{23.8125} \quad D = \underline{27.64} \quad O = \underline{17.D} \quad H$ 。
3. $111001.0101B = \underline{57.3125} \quad D = \underline{71.24} \quad O = \underline{39.5} \quad H$ 。
4. $235.25D = \underline{11101011.01} \quad B = \underline{EB.4} \quad H$ 。
5. $-1110011B$ 的反码为: 10001100 B, 补码为: 10001101 B。
6. $-71D$ 的原码为: 11000111 B, 补码为: 10111001 B。
7. $+1001001B$ 的原码为: 01001001 B, 补码为: 01001001 B。
8. 带符号数 $10110101B$ 的反码为: 11001010 B, 补码为: 11001011 B。

三、简答题

1. 计算机内部为什么要采用二进制编码表示?

【答】 ①易于物理实现。因为物理器件有两种稳定状态, 如电压的高与低能表示 1 和 0 两个符号。②运算简单。数学推导证明, 进行算术求和或求积运算, 对于二进制运算规则仅有 3 种, 因而简化了运算器等物理器件的设计。③机器可靠性高。由于电压的高低、电流的有无等都是两种分明的状态, 因此二进制的传递抗干扰能力强, 鉴别信息的可靠性高。④通用性强。二进制适用于各种非数值信息的数字化编码。特别是 0 和 1 两个符号正好与逻辑命题的“真”与“假”两个值相对应, 从而为计算机实现逻辑运算和逻辑判断提供了方便。

2. 简述冯·诺依曼型计算机的特点。

【答】 冯·诺依曼型计算机的设计方案是“存储程序”和“程序控制”, 其有以下 5 方面特点: ①用二进制数表示数据和指令。②指令和数据存储在内部存储器中, 按顺序自动依次执行指令。③由运算器、控制器、存储器、输入设备和输出设备组成基本硬件系统。④由控制器来控制程序和数据的存取及程序的执行。⑤以运算器为核心。

3. 计算机中有哪些常用的数制和码制? 如何进行数制之间的转换?

【答】数值数据经常使用二进制、八进制、十进制和十六进制; 字符数据使用 ASCII 码; 表示十进制数字用 BCD 码。

十进制到二进制: 整数部分连续除以 2 后“倒取余”, 小数部分连续乘以 2 后“正取整”; 二进制到十进制: 将二进制数按权展开即可。二进制到八进制: 将 3 位二进制一组对应 1 位八进制数码。八进制到二进制: 将 1 位八进制数码对应 3 位二进制数码。十六进制与二进制间转换与八进制与二进制间转换类似, 只是比例关系为 1 位十六进制数码对应 4 位二进制数码。

4. 为什么要采用补码运算有符号数?

使用补码, 可以将符号位和其他位统一处理; 同时, 减法也可按加法来处理。另外, 两个用补码表示的数相加时, 如果最高位(符号位)有进位, 则进位被舍弃。

5. 有符号数运算的溢出如何判断?

【答】溢出出现在同符号数相加或异符号数相减的情况下, 判断有符号数运算溢出的情况如下: ①两个有符号数运算时, 结果的次高位向最高位有进位或借位, 而最高位向上无进位, 则结果产生溢出。②如果次高位向最高位无进位或借位, 而最高位向上有进位, 则结果也产生溢出。可以归纳为, 最高位和次高位如果一个有进位一个无进位, 结果即产生溢出 (OF=1)。

6. $X=-1110111\text{B}$, $Y=+1011010\text{B}$, 求 $[X+Y]_{\text{补}}$ 。

【解】 $[X]_{\text{原}}=11110111\text{B}$ $[X]_{\text{补}}=10001001\text{B}$

$[Y]_{\text{原}}=[Y]_{\text{补}}=01011010\text{B}$

$[X+Y]_{\text{补}}=[X]_{\text{补}}+[Y]_{\text{补}}=11100011\text{B}$

7. $X=56\text{D}$, $Y=-21\text{D}$, 求 $[X+Y]_{\text{补}}$ 。

【解】 $[X]_{\text{原}}=[X]_{\text{补}}=00111000\text{B}$

$[Y]_{\text{原}}=10010101\text{B}$ $[Y]_{\text{补}}=11101011\text{B}$

$[X+Y]_{\text{补}}=[X]_{\text{补}}+[Y]_{\text{补}}=00100011\text{B}$

8. $X=-1101001\text{B}$, $Y=-1010110\text{B}$, 求 $[X-Y]_{\text{补}}$ 。

【解】 $[X-Y]_{\text{补}}=[X+(-Y)]_{\text{补}}=[X]_{\text{补}}+[-Y]_{\text{补}}$

$[X]_{\text{原}}=11101001\text{B}$ $[X]_{\text{补}}=10010111\text{B}$

$[-Y]_{\text{原}}=01010110\text{B}$ $[-Y]_{\text{补}}=01010110\text{B}$

$[X-Y]_{\text{补}}=[X]_{\text{补}}+[-Y]_{\text{补}}=11101101\text{B}$

第 2 章 | 微处理器系统

2.1 知识点回顾

1. 8086/8088 CPU 的功能与结构

(1) 8086 概述

8086 是一种单片微处理芯片，其内部数据总线的宽度是 16 位，外部数据总线宽度也是 16 位，片内包含有控制计算机所有功能的各种电路。8086 地址总线的宽度为 20 位，有 1 MB (2^{20}) 寻址空间。

(2) 8086 CPU 的组成

8086 CPU 由总线接口部件 BIU 和执行部件 EU 组成。BIU 和 EU 的操作是异步的，为 8086 取指令和执行指令的并行操作做硬件支持。

(3) 寄存器结构

8086 微处理器包含有 13 个 16 位的寄存器和 9 位标志位：

- ① 4 个通用寄存器 (AX、BX、CX、DX)。
- ② 4 个段寄存器 (CS、DS、SS、ES)。
- ③ 4 个指针和变址寄存器 (SP、BP、SI、DI)。
- ④ 指令指针 (IP)。

(4) 通用寄存器

① 8086 含 4 个 16 位数据寄存器，它们又可分为 8 个 8 位寄存器，常用来存放参与运算的操作数或运算结果，即：

AX 可分为 AH、AL。

BX 可分为 BH、BL。

CX 可分为 CH、CL。

DX 可分为 DH、DL。

② 数据寄存器特有的习惯用法：

AX：累加器。多用于存放中间运算结果。所有 I/O 指令必须通过 AX 与接口传送信息。

BX：基址寄存器。在间接寻址中用于存放基地址。

CX：计数寄存器。用于在循环或串操作指令中存放循环次数或重复次数。

DX：数据寄存器。在 32 位乘除法运算时，存放高 16 位数；在间接寻址的 I/O 指令中存放 I/O 端口地址。

(5) 指针和变址寄存器

SP: 堆栈指针寄存器, 其内容为栈顶的偏移地址。

BP: 基址指针寄存器, 常用于在访问内存时存放内存单元的偏移地址。

SI: 源变址寄存器。

DI: 目标变址寄存器。

变址寄存器常用于指令的间接寻址或变址寻址。

(6) 段寄存器

CS: 代码段寄存器, 代码段用于存放指令代码。

DS: 数据段寄存器。

ES: 附加段寄存器, 数据段和附加段用来存放操作数。

SS: 堆栈段寄存器, 堆栈段用于存放返回地址, 保存寄存器内容, 传递参数。

(7) 指令指针 (IP)

16 位指令指针寄存器, 其内容为下一条要执行的指令的偏移地址。

(8) 标志寄存器 (见表 1-1)

表 1-1 标志寄存器的各位功能表

标志位	含义	作用
CF	进位标志	CF=1, 指令执行结果在最高位上产生一个进位或借位; CF=0, 则无进位或借位产生
PF	奇偶标志	PF=1, 结果低 8 位含偶数个 1; PF=0, 结果低 8 位含奇数个 1
AF	辅助进位标志	AF=1, 运算结果的低 4 位产生了一个进位或借位; AF=0, 则无此进位或借位
ZF	零标志	ZF=1, 运算结果为零; ZF=0, 则运算结果不为零
SF	符号标志	SF=1, 运算结果为负数; SF=0, 则结果为正数
OF	溢出标志	OF=1, 带符号数在进行运算时产生了溢出; OF=0, 则无溢出
TF	陷阱标志	TF=1, 8086 CPU 处于单步工作方式; TF=0, 8086 CPU 正常执行程序
IF	中断允许标志	IF=1, 允许 CPU 接受外部从 INTR 引脚上发来的可屏蔽中断请求信号; IF=0, 则禁止接受可屏蔽中断请求
DF	方向标志	DF=1, 字符串操作指令按递减的顺序对字符串进行处理; DF=0, 字符串操作指令按递增的顺序进行处理

2. 8086 的引脚及其功能 (重点掌握以下引脚)

AD₁₅~AD₀: 双向三态的地址总线, 输入/输出信号。

INTR: 可屏蔽中断请求输入信号, 高电平有效。可通过设置 IF 的值来控制。

NMI: 非屏蔽中断输入信号。不能用软件进行屏蔽。

RESET: 复位输入信号, 高电平有效。

MN/MX: 最小/最大模式输入控制信号。

3. 8086 的最大工作模式和最小工作模式的主要区别

8086 工作在最小模式时, 系统只有一个微处理器, 且系统所有的控制信号全部由 8086 CPU 提供; 在最大模式时, 系统是由多个微处理器/协处理器构成的多机系统, 控制信号通过总线控制器产生, 且系统资源由各处理器共享。

8086 CPU 工作在何种模式下通过 CPU 的第 33 条引脚 $\overline{MN}/\overline{MX}$ 来控制: $\overline{MN}/\overline{MX}=1$, 系统处于最小工作模式; $\overline{MN}/\overline{MX}=0$, 系统处于最大工作模式。

4. 逻辑地址、偏移地址、物理地址的含义, 物理地址的形成

逻辑地址: 表示为“段地址: 偏移地址”, 书写程序时用到, 一个存储单元可对应出多个逻辑地址。

偏移地址: 是某一存储单元距离所在逻辑段的开始地址的字节个数。

物理地址: 是 CPU 访问存储器时用到的 20 位地址, 是存储单元唯一的编号。

物理地址计算公式: 物理地址=段地址 $\times 10H$ +有效地址(或偏移地址)。

5. 堆栈的使用

堆栈的工作方式是“先进后出”方式, 用入栈指令 PUSH 和出栈指令 POP 可将数据压入堆栈或从堆栈中弹出数据, 栈顶指针 SP 的变化由 CPU 自动管理。堆栈以字为单位进行操作, 堆栈中的数据项以低字节在偶地址、高字节在奇地址的次序存放, 这样可保证每访问一次堆栈就能压入/弹出一个字。当执行 PUSH 指令时, CPU 自动修改指针 $SP-2 \rightarrow SP$ 。使 SP 指向新栈顶, 然后将低位数据压入 (SP) 单元, 高位数据压入 (SP+1) 单元。当执行 POP 指令时, CPU 先将当前栈顶 SP (低位数据) 和 SP+1 (高位数据) 中的内容弹出, 然后再自动修改指针, 使 $SP+2 \rightarrow SP$, SP 指向新栈顶。

2.2 主教材习题答案

一、选择题

- 8086 微处理器可以分为两个部件, 它们是 A。
 - 运算器和控制器
 - 运算器和寄存器
 - 寄存器组和标志寄存器
 - 分析指令和执行指令部件
- 8086 CPU 中 IP 的作用是 B。
 - 保存放置在数据总线上的数据
 - 装有将要执行的下一条指令的段内地址
 - 保存被译码的指令操作码
 - 装有当前正在执行指令的段地址
- 在 8086 最小方式的读操作时序中, 一个基本的总线读周期包含 4 个状态, T_1 、 T_2 、 T_3 和 T_4 。在存储器和外设的速度较慢时, 要在 之后插入一个或者几个等待周期 T_w 。
 - T_1
 - T_2
 - T_3
 - T_4
- 假设某处理器具有 2 MB 的寻址能力, 则该处理器具有的地址线条数是 D。
 - 16 条
 - 8 条
 - 20 条
 - 21 条
- 8086 CPU 中, 控制线 $\overline{RD}$ 和 $\overline{WR}$ 的作用是 A。
 - 数据收发器方向控制
 - 存储器存取操作控制
 - 存储器片选控制
 - 地址/数据线分离控制
- 8086 CPU 中, 确定下一条指令的物理地址的算术表达式为 A。
 - $CS \times 16 + IP$
 - $DS \times 16 + SI$
 - $SS \times 16 + SP$
 - $ES \times 16 + DI$