



普通高等教育“十二五”规划教材 计算机系列  
中国科学院教材建设委员会“十二五”规划教材

# 微机原理与接口技术习题与实验指导

WEIJI YUANLI YU JIEKOU JISHU XITI YU SHIYANZHIDAO

侯彦利 主 编

郭 威 刘 通 马爱民 赵永华 副主编



科学出版社

普通高等教育“十二五”规划教材 计算机系列  
中国科学院教材建设委员会“十二五”规划教材

# 微机原理与接口技术 习题与实验指导

侯彦利 主 编

郭 威 刘 通  
马爱民 赵永华

副主编

科 学 出 版 社

北 京

## 内 容 简 介

本书作为《微机原理及接口技术》的配套教材,包含了《微机原理及接口技术》各章节后的习题解答,并着重于 8088/8086 系列微机的基本实验的介绍。本书以“LAB8000 改进型伟福通用微控制器仿真实验系统”为基础,介绍了微机原理及接口技术课程所需的各类实验,包括实验目的、实验内容、实验步骤等。对每一个实验都给出了较为详细的硬件原理图,对实验需要的一些基础知识也进行了必要的补充。

本书是一本教学辅导用书,可作为普通高等院校工科非计算机类的本、专科学生的微机原理及接口技术课程的实验教材,也可作为工程技术人员学习的参考书。

### 图书在版编目(CIP)数据

微机原理与接口技术习题与实验指导 / 侯彦利主编. —北京:科学出版社, 2012

(普通高等教育“十二五”规划教材·计算机系列)

ISBN 978-7-03-033556-2

I. ①微… II. ①侯… 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(2012)第 023735 号

策划编辑: 戴 薇

责任编辑: 雋青龙 / 责任校对: 马英菊

责任印制: 吕春珉 / 封面设计: 东方人华平面设计部

科学出版社出版

北京东黄城根北街 16 号

邮政编码: 100717

<http://www.sciencep.com>

骏主印刷厂印刷

科学出版社发行 各地新华书店经销

\*

2012 年 3 月第 一 版 开本: 787×1092 1/16

2012 年 3 月第一次印刷 印张: 8 3/4

字数: 196 000

定价: 19.00 元

(如有印装质量问题, 我社负责调换(骏杰))

销售部电话 010-62134988 编辑部电话 010-62135517-2037

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

## 前言

“微机原理与接口技术”是高等院校非计算机专业学生，特别是工学院各相关专业学生学习微型计算机基本知识和应用技能的重要课程。本课程帮助学生掌握微型计算机的硬件组成及应用，学会运用指令系统和汇编语言进行程序设计，掌握微型计算机接口的基本技术和应用，为后继的智能控制系统类课程的学习打好基础。实验教学是本课程的重要组成部分，对学生掌握程序设计方法和自动控制电路设计技能至关重要。

本书实验采用的主要设备为 LAB8000 改进型伟福通用微控制器仿真实验系统。所有实验内容均以此实验系统为基础。本书中的第 3、4、6、7、8 章后均安排了实验内容，每一个实验都紧扣理论知识要点，采用由简入繁、步步深入的方法引导学生做实验。实验与实际应用相结合，充满趣味性，充分调动学生学习的积极性。

本书包含《微机原理与接口技术》一书中各章节后的习题解答，仅供参考。

由于编者水平有限，书中难免存在不足之处，敬请读者批评指正。

编 者

2012 年 1 月  
于吉林大学

## 前言

|                             |    |
|-----------------------------|----|
| 第 1 章 微型计算机基础知识 .....       | 1  |
| 1.1 知识要点 .....              | 1  |
| 1.2 习题解答 .....              | 3  |
| 第 2 章 8088/8086 微处理器 .....  | 5  |
| 2.1 知识要点 .....              | 5  |
| 2.2 习题解答 .....              | 7  |
| 第 3 章 8088/8086 的指令系统 ..... | 10 |
| 3.1 知识要点 .....              | 10 |
| 3.2 习题解答 .....              | 11 |
| 3.3 debug 使用实验 .....        | 19 |
| 实验 1 debug 的使用 .....        | 23 |
| 实验 2 算术运算指令的应用 .....        | 25 |
| 实验 3 串操作指令的应用 .....         | 26 |
| 实验 4 转移指令的应用 .....          | 27 |
| 第 4 章 汇编语言及其程序设计 .....      | 28 |
| 4.1 知识要点 .....              | 28 |
| 4.2 习题解答 .....              | 30 |
| 4.3 汇编语言程序设计实验 .....        | 41 |
| 实验 1 显示字符实验 .....           | 41 |
| 实验 2 BCD 到 ASCII 码转换 .....  | 47 |
| 实验 3 计算 N! .....            | 48 |
| 实验 4 两个多位十进制数相减 .....       | 54 |
| 实验 5 从键盘输入数据并显示 .....       | 56 |
| 实验 6 响铃程序 .....             | 59 |
| 实验 7 接收月/日/年信息并显示 .....     | 61 |
| 实验 8 排序 .....               | 64 |
| 实验 9 学生成绩名次表 .....          | 69 |

|                    |     |
|--------------------|-----|
| 第5章 存储器            | 74  |
| 5.1 知识要点           | 74  |
| 5.2 习题解答           | 75  |
| 第6章 输入/输出与中断技术     | 80  |
| 6.1 知识要点           | 80  |
| 6.2 习题解答           | 82  |
| 6.3 简单接口的数据输入/输出实验 | 87  |
| 实验1 用74HC245读入数据   | 87  |
| 实验2 用74HC273输出数据   | 89  |
| 实验3 8259外部中断实验     | 91  |
| 第7章 可编程接口芯片        | 95  |
| 7.1 知识要点           | 95  |
| 7.2 习题解答           | 97  |
| 7.3 8253/8255应用实验  | 102 |
| 实验1 8253计数器实验      | 102 |
| 实验2 8253定时器实验      | 103 |
| 实验3 8255输入/输出实验    | 105 |
| 实验4 八段数码管显示        | 108 |
| 实验5 键盘扫描显示实验       | 112 |
| 第8章 数/模转换及模/数转换技术  | 118 |
| 8.1 知识要点           | 118 |
| 8.2 习题解答           | 120 |
| 8.3 数/模转换和模/数转换实验  | 122 |
| 实验1 数/模转换实验        | 122 |
| 实验2 模/数转换实验        | 124 |
| 实验3 压力传感器实验        | 126 |
| 第9章 总线技术           | 128 |
| 9.1 知识要点           | 128 |
| 9.2 习题解答           | 129 |
| 参考文献               | 132 |

# 第 1 章 微型计算机基础知识

## 1.1 知识要点

(1) 根据组成计算机基本电路的元器件的发展, 计算机的发展过程大致分成四个阶段。

第一代: 电子管计算机。

第二代: 晶体管计算机。

第三代: 集成电路计算机。

第四代: 大规模集成电路计算机。

(2) 冯·诺依曼计算机体系结构包含运算器、控制器、存储器、输入设备、输出设备五大部件。

(3) 计算机按其性能、体积和价格等的不同, 可分为巨型机、大型机、中型机、小型机、微型机和单片机六大类。

(4) 中央处理器 (Central Processing Unit, CPU) 的内部结构一般包含运算器、控制器和寄存器组三大部分。

(5) 按工作方式不同, 存储器 (简称内存) 可分为两大类, 即随机存储器 (Random Access Memory, RAM) 和只读存储器 (Read Only Memory, ROM)。RAM 可以被 CPU 随机地读和写, 当机器断电后, 所存信息消失。ROM 中的信息只能被 CPU 随机读取, 而不能由 CPU 写入。机器断电后, 信息并不丢失。

(6) 总线由一组导线和相关控制电路组成, 是各种公共信号线的集合, 用于微机系统各大部件之间的信息传送。在 CPU、存储器、I/O 接口之间传送信息的总线称为系统总线。根据所传送信息的类别的不同, 总线又分为三种类型 (俗称系统三总线)。它们分别是地址总线、数据总线和控制总线。

(7) 各种数制之间的转换方法。

1) 非十进制数到十进制数的转换。任何一个非十进制数, 按其权表达式展开后计算所得结果即为十进制数。

2) 十进制数转换为二进制数。整数部分的转换方法是“除 2 取余”, 小数部分采用“乘 2 取整”的方法

3) 十进制数转换为十六进制数。其有以下两种方法。

方法 1: 整数部分用“除 16 取余”, 小数部分则“乘 16 取整”。

方法 2: 先把十进制数转换为二进制数, 之后再转换为十六进制数。

#### 4) 二进制数与十六进制数之间的转换。

将二进制数转换为十六进制数的方法是从小数点开始分别向左和向右把整数部分和小数部分每 4 位分成一组。若整数部分的最高的一组不够 4 位, 则在其左边补零来构成 4 位; 同样若小数部分最后一组不够 4 位, 则在其右补零来构成 4 位。之后将每组二进制数用相应的十六进制数替换, 则得到转换结果。

#### (8) BCD 码与十进制数、二进制数的转换。

按 BCD 码与十进制数对应关系把 4 位 BCD 码与 1 位十进制数相互转换。

#### (9) 无符号数的表示范围及运算溢出判断。

一个  $n$  位无符号二进制数  $X$ , 其表示数的范围为:

$$0 \leq X \leq 2^n - 1$$

无符号数加减运算的溢出判断的准则是如果运算结果的最高位向更高位产生进位(加法)或借位(减法), 则运算结果产生溢出。

#### (10) 带符号数在计算机中的原码、反码和补码的表示方法。

#### (11) 补码运算规则。

##### 1) 补码的加法规则为

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

##### 2) 补码的减法规则为

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

#### (12) 带符号数的溢出判断。

对于一个  $n$  位的带符号二进制数, 如果运算过程中  $C_{n-1} \oplus C_{n-2} = 1$ , 则运算结果产生溢出; 如果  $C_{n-1} \oplus C_{n-2} = 0$ , 则运算结果没有产生溢出。

#### (13) 基本逻辑门及常用逻辑部件。

1) 与门。与门是实现“与”运算的电路。若输入的逻辑变量为  $A$  和  $B$ , 则通过与门输出的结果  $F$  可表示为:

$$F = A \wedge B$$

2) 或门。或门是实现“或”运算的电路。若输入的逻辑变量为  $A$  和  $B$ , 则通过或门输出的结果  $F$  可表示为:

$$F = A \vee B$$

3) 非门。非门是实现“非”运算的电路, 又称反相器。它只有一个输入端和一个输出端。若输入的逻辑变量为  $A$ , 则通过非门输出的结果  $F$  可表示为:

$$F = \overline{A}$$

4) 与非门。与非门是实现先“与”运算再“非”运算的电路。若输入的逻辑变量为  $A$  和  $B$ , 则通过与非门输出的结果  $F$  可表示为:

$$F = \overline{A \wedge B}$$

5) 或非门。或非门是实现先“或”运算再“非”运算的电路。若输入的逻辑变量为  $A$  和  $B$ , 则通过或非门输出的结果  $F$  可表示为:



$$F = \overline{A \vee B}$$

6) 74LS138 的逻辑关系。

## 1.2 习题解答

(1) CPU 内部包含哪三大部分?

答: 运算器、控制器和寄存器组。

(2) 完成下列数制的转换。

①  $10101101\text{B} = (\quad) \text{D} = (\quad) \text{H}$ 。

②  $0.11\text{B} = (\quad) \text{D}$ 。

③  $211.25 = (\quad) \text{B} = (\quad) \text{H}$ 。

④  $10111.0101\text{B} = (\quad) \text{H} = (\quad) \text{BCD}$ 。

答: ①  $10101101\text{B} = 173\text{D} = \text{ADH}$ 。

②  $0.11\text{B} = 0.75\text{D}$ 。

③  $211.25 = 11010011.01\text{B} = \text{D3.4H}$ 。

④  $10111.0101\text{B} = 17.5\text{H} = (0001\ 0111.0101) \text{BCD}$ 。

(3) 已知  $X = +1011010\text{B}$ ,  $Y = -0011011\text{B}$ , 设机器数为 8 位, 分别写出  $X$ 、 $Y$  的原码、反码和补码。

答:

$$[X]_{\text{原}} = 01011010\text{B}; \quad [Y]_{\text{原}} = 10011011\text{B}.$$

$$[X]_{\text{反}} = 01011010\text{B}; \quad [Y]_{\text{反}} = 11100100\text{B}.$$

$$[X]_{\text{补}} = 01011010\text{B}; \quad [Y]_{\text{补}} = 11100101\text{B}.$$

(4) 已知  $X$  的真值为 32,  $Y$  的真值为 -19, 求  $[X+Y]_{\text{补}}$  的值。

答:  $[X+Y]_{\text{补}} = 00001101\text{B}$ 。

(5) 已知  $X = 51$ ,  $Y = -86$ , 用补码完成下列运算, 并判断是否产生溢出 (设字长为 8 位)。

①  $X+Y$ ;

②  $X-Y$ ;

③  $-X+Y$ ;

④  $-X-Y$ ;

答:

①  $10100011\text{B}$ , 因为  $C_6 \oplus C_7 = 0$ , 所以未产生溢出。

②  $11110111\text{B}$ , 因为  $C_6 \oplus C_7 = 1$ , 所以产生溢出。

③  $01110111\text{B}$ , 因为  $C_6 \oplus C_7 = 1$ , 所以产生溢出。

④  $00100011\text{B}$ , 因为  $C_6 \oplus C_7 = 0$ , 所以未产生溢出。

(6) 若使与门的输出端输出高电平, 则各输入端的状态是什么?

答: 各输入端为高电平。

(7) 若使与非门的输出端输出低电平, 则各输入端的状态是什么?

答：各输入端为高电平。

(8) 如果 74LS138 译码器的  $\bar{Y}_4$  端输出低电平，则 C、B、A 三个输入端的状态分别是什么？

答：C、B、A 三个输入端的状态分别是 ‘1’、‘0’、‘0’。

## 第 2 章 8088/8086 微处理器

### 2.1 知识要点

(1) 8088 微处理器 (CPU) 与 8086 在硬件结构上的主要区别。

8086 CPU 的外部数据总线宽度为 16 位, 而 8088 CPU 的外部数据总线宽度为 8 位; 另外 8086 CPU 的指令预取队列长度为 6 字节, 而 8088 CPU 的指令预取队列长度为 4 字节。

(2) 8088 CPU 包含两大功能部件, 即执行单元 (Execution Unit, EU) 和总线接口单元 (Bus Interface Unit, BIU)。

1) 执行单元主要完成指令的译码和执行。执行单元通过 EU 控制电路从 BIU 中取出指令, 经过指令译码形成各种定时控制信号, 向 EU 内各功能部件发出相应的控制命令, 以完成指令所规定的操作。

2) 总线接口单元是 8086/8088 CPU 同外部联系的接口。它负责所有涉及外部总线的操作, 包括取指令、读操作数、写操作数、地址转换和总线控制等。BIU 内部设置指令预取队列, 每当指令队列有两个或两个以上的字节空间, 且执行单元未向 BIU 申请读/写存储器操作数时, BIU 顺序地预取后续指令的代码, 并填入指令队列中。当 EU 执行的是转移指令时, 则 BIU 清除当前的指令预取队列的数据, 从新的地址取回指令, 并立即送到 EU 去执行, 然后, 从后续的指令序列中取回指令填满队列。

(3) 8088 CPU 地址线的宽度为 20 位, 可寻址的最大内存空间为  $2^{20}=1\text{MB}$ 。

(4) 8088 CPU 对存储器的管理。8088 CPU 内部的寄存器的宽度只有 16 位, 如何把 16 位的二进制数转换成 20 位物理地址? 8088 CPU 采用将内存存储器分段的方法来解决这一问题, 即将 1MB 的地址空间分为 16 个 64KB 的段, 然后用段基址加上段内偏移地址来访问存储器。

分段的具体方法是在分段时, 要求段的起始单元的物理地址是 16 的整数倍, 即 XXXX0H。其中, 把前 16 位 (XXXXH) 称为段基址, 段内各存储单元地址相对于该段起始存储单元地址的位移增量称为段内偏移量。

段基址和段内偏移量称为逻辑地址, 逻辑地址通常写成 XXXXH:YYYYH 的形式, 其中 XXXXH 为段基址, YYYYYH 为段内偏移地址。物理地址与逻辑地址的关系如下:

$$\text{物理地址} = \text{段基址} \times 16 + \text{段内偏移}$$

例如, 逻辑地址 A562H:9236H 对应的物理地址是 AE856H。

物理地址是机器硬件操作时所使用的地址; 逻辑地址是应用人员在编程时所使用的

地址。由逻辑地址形成物理地址是由总线接口部件中的电路实现的。

(5) 8086/8088 CPU 内部的各寄存器的作用。

AX、BX、CX 和 DX 通用寄存器一般用于存放参与运算的数据或运算的结果。

SP 用于存放堆栈栈顶的段内偏移量。

BP 用于存放访问存储器时的偏移地址。

SP 和 BP 也可以用来存放数据，它们的默认段寄存器都是 SS。

SI 和 DI 通常在间接寻址方式中存放操作数的偏移地址。在串操作指令中，DI 的默认段寄存器是 ES。SI 和 DI 也可以用来存放数据。

CS 为代码段寄存器，用于存放代码段的段基址。

DS 为数据段寄存器，用于存放数据段的段基址。

SS 为堆栈段寄存器，用于存放堆栈段的段基址。

ES 为附加段寄存器，用于存放附加段的段基址。

IP 为指令指针寄存器，用于存放 CPU 即将执行的下一条指令在代码段中的段内偏移地址。

FLAGS 为标志寄存器，用于存放指令执行结果的特征。

(6) 8088 CPU 管脚在两种工作模式下的功能。

(7) 8088 CPU 的读、写时序。

(8) 8088 CPU 有两种工作模式，即最小模式和最大模式，如图 2-1 和图 2-2 所示。

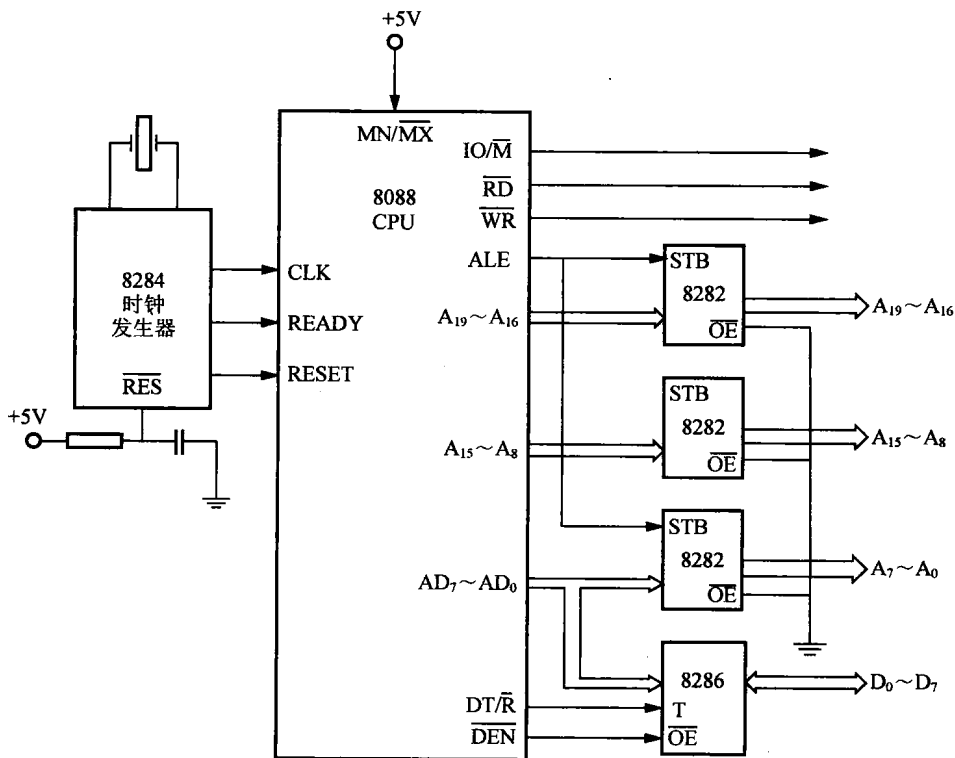


图 2-1 8088 最小模式总线

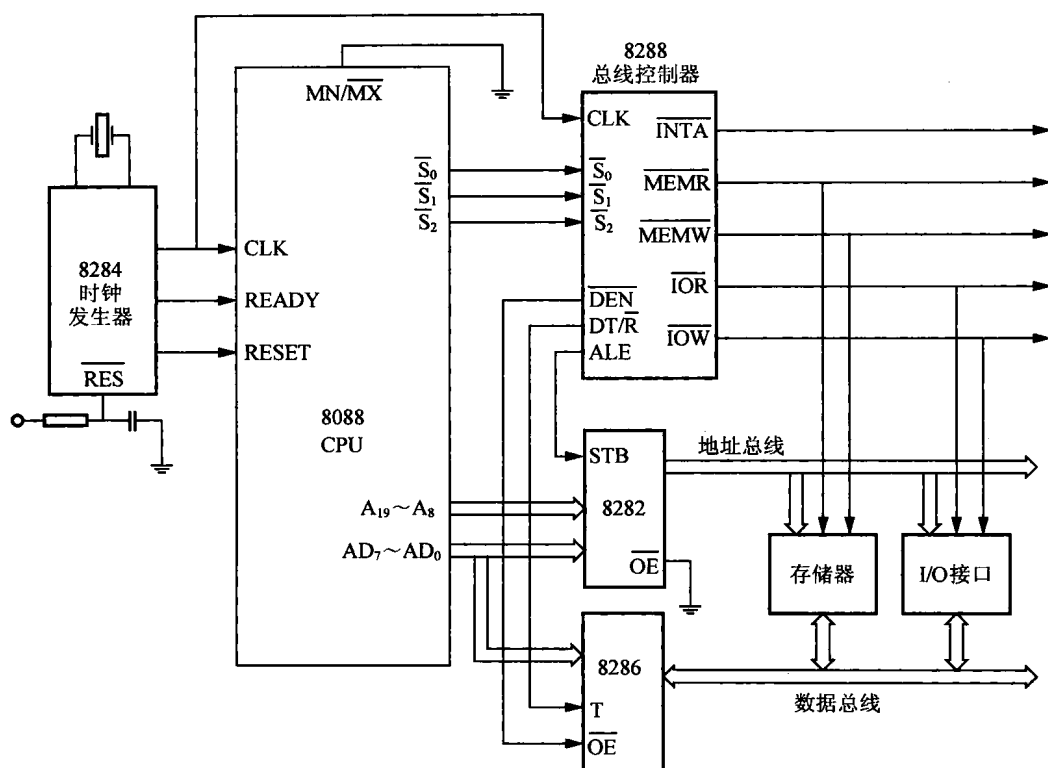


图 2-2 8088 最大模式总线

## 2.2 习题解答

(1) 8086/8088 CPU 由哪两大功能部分所组成？简述它们的主要功能。

答：8086/8088 CPU 由 EU 和 BIU 两大功能部分组成。

执行单元主要完成指令的译码和执行。执行单元通过 EU 控制电路从 BIU 中取出指令，经过指令译码形成各种定时控制信号，向 EU 内各功能部件发出相应的控制命令，以完成指令所规定的操作。

总线接口单元是 8086/8088 CPU 同外部联系的接口。它负责所有涉及外部总线的操作，包括取指令、读操作数、写操作数、地址转换和总线控制等。BIU 内部设置指令预取队列，每当指令队列有两个或两个以上的字节空间，且执行单元未向 BIU 申请读/写存储器操作数时，BIU 顺序地预取后续指令的代码，并填入指令队列中。当 EU 执行的是转移指令时，则 BIU 清除当前的指令预取队列的数据，从新的地址取回指令，并立即送到 EU 去执行，然后从后续指令序列中取回指令填满队列。

(2) 什么是指令流水线？指令流水线需要哪些硬件支持？

答：指令流水线是指 8086/8088 CPU 内部的 EU 和 BIU 通过指令预取队列协同工作从而实现指令的并行执行。指令流水线最主要的硬件支持是 BIU 内部的指令预取队列。

(3) 逻辑地址如何转换成物理地址? 已知逻辑地址为 2D1EH:35B8H, 对应的物理地址是什么?

答: 逻辑地址是段基址和段内偏移地址形式的地址。物理地址与逻辑地址的关系如下:

$$\text{物理地址} = \text{段基址} \times 16 + \text{段内偏移}$$

已知逻辑地址为 2D1EH:35B8H, 则对应的物理地址是 30798H ( $2\text{D1EH} \times 16 + 35\text{B8H}$ )

(4) 8088 CPU 和 8086 CPU 的指令预取队列的长度分别是多少?

答: 8088 CPU 的指令预取队列的长度为 4 字节; 8086 CPU 的指令预取队列的长度为 6 字节。

(5) 简述 8086/8088 CPU 内部的各寄存器的作用。

答: AX、BX、CX 和 DX 通用寄存器一般用于存放参与运算的数据或运算的结果。SP 用于存放堆栈栈顶的段内偏移量。

BP 用于存放访问内存时的偏移地址。

SP 和 BP 也可以用来存放数据, 它们的默认段寄存器都是 SS。

SI 和 DI 通常在间接寻址方式中存放操作数的偏移地址。在串操作指令中, DI 的默认段寄存器是 ES。SI 和 DI 也可以用来存放数据。

CS 为代码段寄存器, 用于存放代码段的段基址。

DS 为数据段寄存器, 用于存放数据段的段基址。

SS 为堆栈段寄存器, 用于存放堆栈段的段基址。

ES 为附加段寄存器, 用于存放附加段的段基址。

IP 为指令指针寄存器, 用于存放 CPU 即将执行的下一条指令在代码段中的段内偏移地址。

FLAGS 为标志寄存器, 用于存放指令执行结果的特征。

(6) 8086/8088 CPU 内部的状态标志寄存器共有几位标志位? 各位的含义是什么?

答: 状态标志寄存器共有 9 位标志位, 其中包含 6 个状态标志位和 3 个控制标志位。各状态标志位的含义分别如下。

CF (Carry Flag): 进位标志。当算术运算结果使最高位产生进位或借位时, 则 CF=1; 否则 CF=0。

PF (Parity Flag): 奇偶标志。若运算结果中的低 8 位含有偶数个 1, 则 PF=1; 否则 PF=0。

AF (Auxiliary carry Flag): 辅助进位标志。运算过程中若 D<sub>3</sub> 位向 D<sub>4</sub> 位有进位或借位时, AF=1; 否则 AF=0。

ZF (Zero Flag): 零标志。若运算结果为 0, 则 ZF=1; 否则 ZF=0。

SF (Sign Flag): 符号标志。若运算结果为负, 则 SF=1; 否则 SF=0。

OF (Overflow Flag): 溢出标志。当带符号数的补码运算结果超出机器所能表达的范围时, 就会产生溢出, 这时 OF=1; 否则 OF=0。

各控制标志位的含义分别如下。

**DF (Direction Flag):** 方向标志。控制串操作指令的地址变化的方向。当  $DF=0$  时, 串操作指令的地址指针按增量变化; 当  $DF=1$  时, 串操作指令的地址指针按减量变化。

**IF (Interrupt Flag):** 中断允许标志。控制微处理器是否允许响应可屏蔽中断请求。若  $IF=1$ , 则允许响应; 否则禁止响应。

**TF (Trap Flag):** 单步标志。当  $TF=1$  时, CPU 工作在单步方式。

(7) 8086/8088 系统中存储器的分段原则是什么?

**答:** 8086/8088 系统中存储器的分段原则是要求段的起始单元的物理地址是 16 的整数倍数, 每段的物理长度为 64KB。

(8) 当 ALE (Address Latch Enable) 有效时, 8088 CPU 的地址/数据总线上将出现什么信息?

**答:** 当 ALE 有效时, 8088 CPU 的地址/数据总线上将出现地址信息。

(9) READY 管脚的作用是什么?

**答:** READY 用于 CPU 时序中插入等待状态。若该引脚被置为低电平, 则 CPU 进入等待状态并保持空闲; 若该引脚被置为高电平, 则它对 CPU 的操作不产生影响。

CPU 在读、写操作时序中的  $T_3$  时钟周期开始处, 通过检测 READY 管脚的状态来决定是否插入  $T_w$  等待时钟周期, 以解决 CPU 与存储器或 I/O 接口之间速度不匹配的矛盾。

(10) 为什么在基于 8086/8088 CPU 的系统中经常需要使用缓冲器?

**答:** 由于基于 8086/8088 CPU 的系统驱动能力的不足, 需要使用缓冲器。

(11) 8088 CPU 工作在最小模式下包含哪些控制信号?

**答:** 8088 CPU 工作在最小模式下包含的控制信号有  $\overline{RD}$ 、 $\overline{WR}$ 、 $IO/\overline{M}$ 、ALE、 $DT/\overline{R}$ 、 $\overline{DEN}$  和  $\overline{INTA}$  等信号。

(12) 若  $CS=4000H$ , 则当前代码段可寻址的存储空间范围是多少?

**答:**  $CS=4000H$  时, 当前代码段可寻址的存储空间范围为  $40000H \sim 4FFFFH$ 。

## 第 3 章 8088/8086 的指令系统

### 3.1 知 识 要 点

汇编语言是一种符号语言。用汇编语言编写的程序称为汇编语言源程序，计算机不能直接识别执行，必须翻译成机器语言程序。

一条指令一般由操作码和操作数两部分组成。操作码详细地说明指令要执行的操作，操作数是指令执行时需要的数据。8086 CPU 指令的操作数有三种类型，即立即数、寄存器操作数和存储器操作数。

#### 1. 8086 CPU 的寻址方式

寻址方式，即获得地址的方法，主要指获得指令中操作数地址的方法，即段内偏移地址的方法。

- (1) 立即寻址：操作数是立即数。
- (2) 直接寻址：操作数在内存中，指令中直接给出操作数所在的内存单元的偏移地址。
- (3) 寄存器寻址：操作数在 CPU 内部的寄存器中。
- (4) 寄存器间接寻址：操作数在内存中，内存单元的偏移地址存放在寄存器中。
- (5) 寄存器相对寻址：操作数在内存中，内存单元的偏移地址一部分由间接寻址寄存器提供，一部分是指令给定的 8 位或 16 位地址位移量，二者相加形成操作数的有效地址。
- (6) 基址变址寻址：操作数在内存中，基址寄存器和变址寄存器相加作为操作数的偏移地址。
- (7) 基址变址相对寻址：操作数在内存中，操作数的地址由基址寄存器加上变址寄存器再加上地址位移量构成。
- (8) 隐含寻址：操作码隐含地指明操作数的地址。

#### 2. 8086 指令系统

8086 指令系统的基本指令如表 3-1 所示。

表 3-1 8086 指令系统的基本指令

| 类别               | 指令类型    | 助 记 符                           |
|------------------|---------|---------------------------------|
| 数<br>据<br>传<br>送 | 一般数据传送  | MOV, PUSH, XCHG, XLAT, CBW, CWD |
|                  | 输入/输出指令 | IN, OUT                         |
|                  | 地址传送指令  | LEA, LDS, LES                   |
|                  | 标志传送指令  | LAHF, SAHF, PUSHF, POPF         |



续表

| 类别        | 指令类型    | 助记符                                                          |
|-----------|---------|--------------------------------------------------------------|
| 算术运算      | 加法指令    | ADD, ADC, INC                                                |
|           | 减法指令    | SUB, SBB, DEC, NEG, CMP                                      |
|           | 乘法指令    | MUL, IMUL                                                    |
|           | 除法指令    | DIV, IDIV                                                    |
|           | 十进制调整指令 | DAA, AAA, DAS, AAS, AAM, AAD                                 |
| 逻辑运算和移位指令 |         | AND, OR, NOT, XOR, TEST, SHL, SAL, SHR, SAR, ROL, ROR, RCL,  |
| 串操作指令     |         | MOVS, CMPS, SCAS, LODS, STOS                                 |
| 控制转移指令    |         | JMP, CALL, RET, LOOPE, INT, INTO, IRET, 各类条件转移指令             |
| 处理器控制指令   |         | CLC, STC, CMC, CLD, STD, CLI, STI, HLT, WAIT, ESC, LOOK, NOP |

## 3.2 习题解答

(1) 什么叫寻址方式？8086 指令系统中有哪几种寻址方式？

答：寻址方式，即获得地址的方法，主要指获得段内偏移地址的方法。

(2) 下列指令中 BUFF 为字节类型变量，DATA 为常量，指出下列指令中源操作数的寻址方式。

- 1) MOV AX, 1200
- 2) MOV AL, BUFF
- 3) SUB BX, [2000H]
- 4) MOV CX, [SI]
- 5) MOV DX, DATA[SI]
- 6) MOV BL, [SI][BX]
- 7) MOV [DI], AX
- 8) ADD AX, DATA[DI+BP]
- 9) PUSHF
- 10) MOV BX, ES:[SI]

- 答：
- 1) MOV AX, 1200 ;立即寻址
  - 2) MOV AL, BUFF ;直接寻址
  - 3) SUB BX, [2000H] ;直接寻址
  - 4) MOV CX, [SI] ;寄存器间接寻址
  - 5) MOV DX, DATA[SI] ;寄存器相对寻址
  - 6) MOV BL, [SI][BX] ;基址变址寻址
  - 7) MOV [DI], AX ;寄存器寻址
  - 8) ADD AX, DATA[DI+BP] ;基址变址相对寻址
  - 9) PUSHF ;隐含寻址
  - 10) MOV BX, ES:[SI] ;寄存器间接寻址