

21 世纪重点大学规划教材

# 80 × 86 微机原理及接口技术 ——习题解答与实验指导

第 2 版

左国玉 余春暄 韩德强 等编著



机械工业出版社

本书是《80×86/Pentium 微机原理及接口技术》的配套教材。全书共分两部分：第一部分为学习指导与习题解答，为配合读者学习或复习微机原理及接口技术课程，首先给出了各章主要内容、重点及难点，通过不同形式的习题与解答，强调基本原理、基本概念，给出其应用的基本方法；第二部分为实验指导，介绍了软件和硬件实验平台及使用方法，并设计了相应的软、硬件实验题目，引导读者通过实验加深对课程内容的理解，掌握应用方法。本书的附录给出了 7 位 ASCII 码编码表和逻辑符号对照表，供读者查阅。

本书概念清楚、结构紧凑、详略得当、面向应用，可作为计算机相关专业本、专科学生的参考用书，也可作为学生考研复习的参考资料及教师教学参考用书。

## 图书在版编目 (CIP) 数据

80×86 微机原理及接口技术：习题解答与实验指导/左国玉等编著.  
—2 版. —北京：机械工业出版社，2017.9  
21 世纪重点大学规划教材  
ISBN 978-7-111-58112-3

I. ①8… II. ①左… III. ①微型计算机-理论-高等学校-教材  
②微型计算机-接口技术-高等学校-教材 IV. ①TP36

中国版本图书馆 CIP 数据核字 (2017) 第 237302 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)  
责任编辑：郝建伟 范成欣 责任校对：任秀丽 李锦莉  
责任印制：常天培  
北京京丰印刷厂印刷  
2018 年 1 月第 2 版·第 1 次印刷  
184mm×260mm·20 印张·1 插页·487 千字  
0 001—2 500 册  
标准书号：ISBN 978-7-111-58112-3  
定价：56.00 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](http://www.cmpbook.com)

读者购书热线：010-88379649

机工官博：[weibo.com/cmp1952](http://weibo.com/cmp1952)

教育服务网：[www.cmpedu.com](http://www.cmpedu.com)

封面无防伪标均为盗版

金书网：[www.golden-book.com](http://www.golden-book.com)

## 出版说明

“211 工程”是“重点大学和重点学科建设项目”的简称。进入“211 工程”的高校拥有全国 32% 的在校本科生、69% 的硕士生、84% 的博士生，以及 87% 的有博士学位的教师；覆盖了全国 96% 的国家重点实验室和 85% 的国家重点学科。相对而言，这批高校中的教授、教师有着深厚的专业知识和丰富的教学经验，其中不少教师对我国高等院校的教材建设做过很多重要的工作。为了有效地利用“211 工程”这一丰富资源，实现以重点建设推动整体发展的战略构想，机械工业出版社推出了“21 世纪重点大学规划教材”。

本套教材以重点大学、重点学科的精品教材建设为主要任务，组织知名教授、教师进行编写。教材适用于高等院校计算机及其相关专业，选题涉及公共基础课、硬件、软件、网络技术等，内容紧密贴合高等院校相关学科的课程设备和培养目标，注重教材的科学性、实用性、通用性，在同类教材中具有一定的先进性和权威性。

为了体现建设“立体化”精品教材的宗旨，本套教材为主干课程配备了电子教案、学习指导、习题解答、课程设计、毕业设计指导等内容。

机械工业出版社

# 前 言

“微机原理及接口技术”这门课程是掌握计算机软、硬件技术的基础课程，是各大专院校大部分信息类、机电类、生物工程等专业的一门计算机技术基础必修课程。该课程知识点多，初学者常感到课程难理解、作业难下手、应用难入门。本书通过对大量典型习题和实验的详尽分析和讲解，循序渐进地加深对计算机基本原理和知识的理解。

全书共分两部分：第一部分为学习指导与习题解答，每章首先给出了本章的主要内容、重点及难点，通过单项选择题、判断题、填空题、简答题、分析程序题及编程题等，强调基本原理、基本概念，给出其应用的基本方法；第二部分为实验指导，首先介绍了汇编语言程序的建立方法，并设计了 11 个软件实验，然后基于 GX-8000 微机原理创新实验系统，设计了基础和综合性硬件接口的 6 个实验，引导读者通过实验加深对课程内容的理解，掌握计算机及接口的软、硬件调试方法和设计方法。在本书的附录中给出了 7 位 ASCII 码编码表和逻辑符号对照表，以方便读者在学习和实验过程中查阅。

本书在上一版的基础上做了如下变动：新增大量习题，使题目涉及的知识范围更广，对知识的考查也更全面。硬件接口实验部分采用了新的实验平台 GX-8000 微机原理创新实验系统，上一版书中的第 2 章和第 4 章 TPC-H 型通用微机接口实验系统相关内容均调整到机械工业出版社的百度网盘（链接见封底的二维码）上，同时也将习题部分的自检试卷放在网盘上，读者均可免费下载使用。另外，本书更正了上一版书中出现的个别错误。

本书主要由左国玉和余春暄编写和统稿，另外，施远征参与了第一部分的编写，韩德强和王宗侠负责 GX-8000 系统实验部分素材的编写和整理。左国玉、余春暄完成本书全部内容的审校，施远征完成部分内容的审校。李锋、李展鹏、彭靖漩、张明杰等同学也在本书的编写过程中做了大量的工作，乔俊飞和邓军老师对本书的编写提出了宝贵的意见和建议。本书在编写与出版过程中，得到了机械工业出版社有关人员的指导与帮助，在此一并表示衷心感谢。

本书部分图片中的软件固有元器件符号可能与国家标准不一致，读者可自行查阅相关资料。

由于编者的水平有限，书中难免有错误和不妥之处，恳请广大读者提出宝贵意见。

编 者

# 目 录

出版说明

前言

第一部分 学习指导与习题解答 ..... 1

第1章 计算机基础..... 1

1.1 学习指导..... 1

1.2 单项选择题..... 2

1.3 判断题..... 5

1.4 填空题..... 6

1.5 简答题..... 7

第2章 微处理器 ..... 11

2.1 学习指导 ..... 11

2.2 单项选择题 ..... 13

2.3 判断题 ..... 19

2.4 填空题 ..... 21

2.5 简答题 ..... 24

第3章 80×86 指令系统 ..... 30

3.1 学习指导 ..... 30

3.2 单项选择题 ..... 31

3.3 判断题 ..... 38

3.4 填空题 ..... 41

3.5 简答题 ..... 47

3.6 分析程序题 ..... 57

3.7 编程题 ..... 61

第4章 汇编语言程序设计..... 73

4.1 学习指导 ..... 73

4.2 单项选择题 ..... 73

4.3 判断题 ..... 81

4.4 填空题 ..... 83

4.5 简答题 ..... 95

4.6 分析程序题..... 108

4.7 编程题..... 119

第5章 半导体存储器及其接口技术 ..... 150

5.1 学习指导..... 150

5.2 单项选择题..... 150

5.3 判断题	154
5.4 填空题	155
5.5 简答题	157
5.6 应用题	160
<b>第6章 微型计算机接口技术</b>	<b>168</b>
6.1 学习指导	168
6.2 单项选择题	168
6.3 判断题	173
6.4 填空题	174
6.5 简答题	176
<b>第7章 简单接口电路设计</b>	<b>185</b>
7.1 学习指导	185
7.2 单项选择题	185
7.3 判断题	187
7.4 填空题	188
7.5 应用题	189
<b>第8章 可编程接口技术</b>	<b>195</b>
8.1 可编程计数器 8253/8254	195
8.1.1 学习指导	195
8.1.2 单项选择题	196
8.1.3 判断题	198
8.1.4 填空题	198
8.1.5 简答题	199
8.1.6 应用题	200
8.2 并行通信接口 8255A	204
8.2.1 学习指导	204
8.2.2 单项选择题	205
8.2.3 判断题	207
8.2.4 填空题	207
8.2.5 应用题	208
8.3 串行通信接口 8250/8251	217
8.3.1 学习指导	217
8.3.2 单项选择题	222
8.3.3 判断题	224
8.3.4 填空题	224
8.3.5 简答题	225
8.3.6 应用题	228
8.4 可编程中断控制器 8259A	233
8.4.1 学习指导	233
8.4.2 单项选择题	235

8.4.3 判断题	238
8.4.4 填空题	239
8.4.5 简答题	240
8.4.6 应用题	243
<b>第二部分 实验指导</b>	<b>249</b>
<b>第9章 汇编语言程序的建立方法</b>	<b>249</b>
9.1 汇编语言程序的编程环境	249
9.2 汇编语言程序的建立过程	249
9.2.1 编辑源程序（建立 ASM 源程序文件）	249
9.2.2 汇编程序（用 MASM 命令产生 OBJ 目标文件）	250
9.2.3 连接程序（用 LINK 命令产生 EXE 可执行文件）	251
9.2.4 执行程序	252
9.3 调试程序 DEBUG 的使用	252
9.3.1 DEBUG 程序调用	252
9.3.2 DEBUG 命令的有关规定	252
9.3.3 DEBUG 的主要命令	253
9.3.4 MASM 6.11 的安装	256
9.4 集成开发系统未来汇编的使用说明	256
9.4.1 系统设置	256
9.4.2 汇编程序开发的基本步骤	257
9.4.3 未来汇编的安装	259
<b>第10章 微机硬件接口实验系统介绍</b>	<b>260</b>
10.1 系统简介	260
10.2 实验台结构	261
10.3 软件开发环境	271
<b>第11章 汇编语言程序设计实验</b>	<b>275</b>
实验一 熟悉 8086 指令编程方法及用 DEBUG 调试程序的方法	275
实验二 熟悉汇编程序的建立及其调试方法	276
实验三 多项式求值（顺序结构练习）	278
实验四 有符号数的表示（分支结构练习）	279
实验五 多位数加法（循环结构练习）	280
实验六 均值滤波（子程序结构练习）	281
实验七 字符串查询（DOS 功能调用练习）	282
实验八 建立窗口并设置光标初始位置实验	284
实验九 数码转换	285
实验十 数据排序	288
实验十一 数据分类统计	289
<b>第12章 微机接口设计实验</b>	<b>291</b>
实验一 预备实验	291
实验二 8255A 基础实验	297

实验三 8253 基础实验 .....	300
实验四 8259A 基础实验 .....	302
实验五 8251A 串行通信实验 .....	304
实验六 综合设计 .....	307
附录 .....	309
附录 A 7 位 ASCII 码编码表 .....	309
附录 B 逻辑符号对照表 .....	310
参考文献 .....	312

机械工业出版社



# 第一部分 学习指导与习题解答

## 第 1 章 计算机基础

### 1.1 学习指导

本章主要内容如下：

#### 1. 计算机的发展

简要介绍计算机冯·诺依曼体系结构的特点和计算机的发展史。

#### 2. 整机概念

计算机由微处理器、存储器、输入/输出接口及三总线（数据总线 DB、地址总线 AB、控制总线 CB）组成，如图 1-1 所示。其中，数据总线为双向三态，地址总线为单向三态，控制总线的各信号线特点各异。

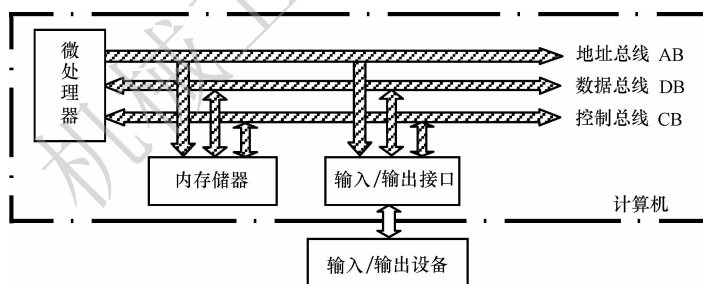


图 1-1 计算机硬件系统组成

#### 3. 计算机中数和编码的表示

1) 进制表示及相互之间的转换。计算机中常用的数的进制主要有二进制数、十进制数、十六进制数。

2) 有符号数的表示（包括真值、原码、反码、补码）及相互之间的转换。需要注意的内容如下：

- 正数的原码、反码和补码相等。
- 负数的反码等于其原码的符号位不变，其他位求反。
- 负数的补码等于其原码的符号位不变，其他位求反后加一。
- 常用的补码运算规则：

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

$$[X]_{\text{原}} = [[X]_{\text{反}}]_{\text{反}}$$

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

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

3) 编码的表示方法包括非压缩型 BCD 码 (用 8 位二进制数表示 1 位十进制数, 其中高 4 位为 0)、压缩型 BCD 码 (用 8 位二进制数表示 2 位十进制数)、ASCII 码 (美国信息交换标准代码, 参见附录 A)。

#### 4. 微机系统中采用的先进技术

微机系统中采用的先进技术包括流水线技术、高速缓冲存储技术、虚拟存储技术、CISC 和 RISC 技术、多核心技术等。

## 1.2 单项选择题

1. 从第一代电子计算机到第四代计算机的体系结构都是相同的, 都是由运算器、控制器、存储器以及输入/输出设备组成的, 称为( )体系结构。

- A. 艾伦·图灵                      B. 罗伯特·诺依斯                      C. 比尔·盖茨                      D. 冯·诺依曼

【解】 D

2. 电子计算机从问世到现在都遵循“存储程序”的概念, 最早提出它的是( )。

- A. 巴贝奇                      B. 冯·诺依曼                      C. 帕斯卡                      D. 贝尔

【解】 B

3. 目前制造计算机所采用的电子元器件是( )。

- A. 晶体管                      B. 电子管  
C. 中小规模集成电路                      D. 超大规模集成电路

【解】 D

4. 计算机之所以能自动连续地进行数据处理, 其主要原因是( )。

- A. 采用了开关电路                      B. 采用了半导体器件  
C. 具有存储程序的功能                      D. 采用了二进制

【解】 C

5. 计算机中存储数据的最小单位是二进制的( )。

- A. 位 (比特)                      B. 字节                      C. 字长                      D. 千字节

【解】 A

6. 1 字节包含( )个二进制位。

- A. 8                      B. 16                      C. 32                      D. 64

【解】 A

7. 二进制数 011001011110B 的十六进制表示为( )。

- A. 44EH                      B. 75FH                      C. 54FH                      D. 65EH

【解】 D

8. 二进制数 011001011110B 的八进制表示为( )。

- A. 4156Q                      B. 3136Q                      C. 4276Q                      D. 3176Q

【解】 B

9. 设  $(123)_{16} = (X)_8 = (Y)_2$ , 其中下标分别表示十六进制、八进制、二进制, 则 X 和 Y 应为 ( )。

A.  $X = 246, Y = 010101110$

B.  $X = 443, Y = 100100011$

C.  $X = 173, Y = 01111011$

D.  $X = 315, Y = 11001101$

【解】 B

10. 下面 4 个无符号数的大小顺序正确的是 ( )。

A.  $0FEH > 250D > 371Q > 01111111B$

B.  $250D > 0FEH > 371Q > 01111111B$

C.  $371Q > 0FEH > 250D > 01111111B$

D.  $01111111B > 0FEH > 250D > 371Q$

【解】 A

11. 带符号的八位二进制补码的表示范围是 ( )。

A.  $-127 \sim +127$

B.  $-32768 \sim +32768$

C.  $-128 \sim +127$

D.  $-32768 \sim +32767$

【解】 C

12. 十进制数  $-61$  的八位二进制原码是 ( )。

A. 00101111B

B. 00111101B

C. 10101111B

D. 10111101B

【解】 D

13. 十进制数  $+121$  的八位二进制反码是 ( )。

A. 00000110B

B. 01001111B

C. 01111001B

【解】 C

14. 十进制数  $-89$  的八位二进制补码为 ( )。

A. B9H

B. 89H

C. 10100111B

D. 00100111B

【解】 C

15. 无符号二进制数 00001101.01B 的真值为 ( )。

A. 13.25

B. 0B.1H

C. 0B.4H

D. 13.01

【解】 A

16. 有符号二进制原码数 10000001B 的真值为 ( )。

A. 01H

B.  $-1$

C. 128

D.  $-127$

【解】 B

17. 数 D8H 被看作用补码表示的有符号数时, 该数的真值为 ( )。

A.  $-58H$

B.  $-28H$

C.  $-40$

D. 216

【解】 C

18. 数 4FH 被看作用反码表示的有符号数时, 该数的真值为 ( )。

A.  $+30H$

B.  $+79$

C.  $+4FH$

【解】 B

19. 计算机内的溢出是指其运算结果 ( )。

A. 无穷大

B. 超出了计算机内存储单元所能存储的数值范围

C. 超出了该指令所指定的结果单元所能存储的数值范围

D. 超出了运算器的取值范围

【解】 C

20. 两个十六进制补码数进行运算  $3AH + B7H$ ，其运算结果（ ）溢出。

- A. 有                                      B. 无

【解】 B

21. 二进制数  $11101110B$  转换为压缩 BCD 码为（ ）。

- A. 00000010 00110011B                                      B. 00000010 01010010B  
C. 00000010 00111000B                                      D. 00000010 00110010B

【解】 C

22. 键盘输入 1999 时，实际运行的 ASCII 码是（ ）。

- A. 41H49H47H46H                                      B. 61H69H67H66H  
C. 31H39H37H36H                                      D. 51H59H57H56H

【解】 C

23. 一个完整的计算机系统通常应包括（ ）。

- A. 系统软件和应用软件                                      B. 计算机及其外围设备  
C. 硬件系统和软件系统                                      D. 系统硬件和系统软件

【解】 C

24. 微型计算机的性能主要由（ ）来决定。

- A. 价钱                                      B. CPU                                      C. 控制器                                      D. 其他

【解】 B

25. 通常所说的“裸机”是指（ ）。

- A. 只装备有操作系统的计算机                                      B. 不带输入/输出设备的计算机  
C. 未装备任何软件的计算机                                      D. 计算机主机暴露在外

【解】 C

26. 计算机运算速度的单位是  $MI/S$ （即 MIPS），其含义是（ ）。

- A. 每秒钟处理百万个字符                                      B. 每分钟处理百万个字符  
C. 每秒钟执行百万条指令                                      D. 每分钟执行百万条指令

【解】 C

27. 通常所说的 32 位机是指这种计算机的 CPU（ ）。

- A. 是由 32 个运算器组成的                                      B. 能够同时处理 32 位二进制数据  
C. 包含 32 个寄存器                                      D. 一共有 32 个运算器和控制器

【解】 B

28. 运算器的主要功能是进行（ ）。

- A. 算术运算                                      B. 逻辑运算                                      C. 算术和逻辑运算                                      D. 函数运算

【解】 C

29. 在一般微处理器中包含（ ）。

- A. 算术逻辑单元                                      B. 主内存                                      C. I/O 单元                                      D. 数据总线

【解】 A

30. 一台计算机实际上是执行（ ）。

- A. 用户编制的高级语言程序                                      B. 用户编制的汇编语言程序

C. 系统程序

D. 由二进制码组成的机器指令

【解】 D

31. 构成微机的主要部件除 CPU、系统总线、I/O 接口外, 还有 ( )。

A. CRT

B. 键盘

C. 磁盘

D. 内存 (ROM 和 RAM)

【解】 D

32. 影响 CPU 处理速度的主要因素是字长、主频、ALU 结构以及 ( )。

A. 有无中断功能

B. 有无采用微程序控制

C. 有无 DMA 功能

D. 有无 Cache

【解】 D

33. 计算机的字长是指 ( )。

A. 32 位长的数据

B. CPU 数据总线的宽度

C. 计算机内部一次可以处理的二进制数码的位数

D. CPU 地址总线的宽度

【解】 C

### 1.3 判断题

1. 汇编语言就是机器语言。( )

2. 对于种类不同的计算机, 其机器指令系统都是相同的。( )

3. 三总线就是数据总线、控制总线、地址总线。( )

4. 计算机中所有数据都是以二进制形式存放的。( )

5. 在计算机中, 数据单位 bit 的意思是字节。( )

6. 若  $[X]_{\text{原}} = [X]_{\text{反}} = [X]_{\text{补}}$ , 则该数为正数。( )

7. 补码的求法是: 正数的补码等于原码, 负数的补码是原码连同符号位一起求反加 1。  
( )

8. 无论何种微机, 其 CPU 都具有相同的机器指令。( )

9. 与二进制数 11001011B 等值的压缩型 BCD 码是 11001011B。( )

10. 十进制数 378 转换成十六进制数是 1710H。( )

11. 与十进制小数 0.5625 等值的二进制小数是 1.0011B。( )

12. 八进制数的基数为 8, 因此在八进制数中可以使用的数字符号是 0、1、2、3、4、5、6、7、8。( )

13. 二进制数 00000101 11101111B 转换成十六进制数是 0FE5H。( )

14. 如果二进制数 11111B ~ 01111B 的最高位为符号位, 则其能表示 31 个十进制数。( )

15. 在汉字国标码 GB 2312 - 80 的字符集中, 共收集了 6763 个常用汉字。( )

【答案】

1. ×    2. ×    3. ✓    4. ✓    5. ×    6. ✓    7. ×    8. ×    9. ×

10. ×    11. ×    12. ×    13. ×    14. ✓    15. ✓

## 1.4 填空题

1. 冯·诺依曼原理的基本思想是\_\_\_\_(1)\_\_\_\_和\_\_\_\_(2)\_\_\_\_。

【解】 (1) 程序存储 (2) 程序控制

2. 第一代计算机采用的电子元器件是\_\_\_\_(1)\_\_\_\_。

【解】 (1) 电子管

3. 一个完整的计算机系统应包括\_\_\_\_(1)\_\_\_\_和\_\_\_\_(2)\_\_\_\_。

【解】 (1) 硬件系统 (2) 软件系统

4. 计算机中的三总线包括\_\_\_\_(1)\_\_\_\_、\_\_\_\_(2)\_\_\_\_和\_\_\_\_(3)\_\_\_\_。

【解】 (1) 数据总线 (2) 地址总线 (3) 控制总线

5. 计算机系统中数据总线用于传输\_\_\_\_(1)\_\_\_\_信息，其特点是\_\_\_\_(2)\_\_\_\_。地址总线用于传输\_\_\_\_(3)\_\_\_\_信息，其特点是\_\_\_\_(4)\_\_\_\_。如果 CPU 的数据总线与地址总线采用同一组信号线，那么系统中需要采用\_\_\_\_(5)\_\_\_\_分离出地址总线。

【解】 (1) 数据 (2) 双向三态 (3) 地址  
(4) 单向三态 (5) 锁存器

6. 计算机的软件可以分成两大类，即\_\_\_\_(1)\_\_\_\_和\_\_\_\_(2)\_\_\_\_。

【解】 (1) 系统软件 (2) 应用软件

7. 在计算机中的负数以\_\_\_\_(1)\_\_\_\_方式表示，这样可以把减法转换为加法。

【解】 (1) 补码

8. 在计算机内部，所有信息的存取、处理、传送都是以\_\_\_\_(1)\_\_\_\_形式进行的。

【解】 (1) 二进制编码

9. 一个字节的带符号数可表示的最大正数为\_\_\_\_(1)\_\_\_\_、最小负数为\_\_\_\_(2)\_\_\_\_。

【解】 (1) +127 (2) -128

10. 一个 8 位二进制补码数 10010011B 等值扩展为 16 位二进制数后，其机器数为\_\_\_\_(1)\_\_\_\_。

【解】 (1) 11111111 10010011B

11. 用补码表示的二进制数 10001000B 转换为对应的十进制数真值为\_\_\_\_(1)\_\_\_\_。

【解】 (1) -120

12. 设机器字长为 8 位，已知  $X = -1$ ，则  $[X]_{\text{原}} = \text{____(1)____}$ ， $[X]_{\text{反}} = \text{____(2)____}$ ， $[X]_{\text{补}} = \text{____(3)____}$ 。

【解】 (1) 81H (2) FEH (3) FFH

13. 字长为 8 位的二进制数 10010100B，若它表示无符号数、原码数或补码数，则该数的真值应分别为\_\_\_\_(1)\_\_\_\_D、\_\_\_\_(2)\_\_\_\_D 或\_\_\_\_(3)\_\_\_\_D。

【解】 (1) 148 (2) -20 (3) -108

14. 将十进制整数 4120 分别转换为对应的二进制数、八进制数和十六进制数，则其转换结果分别为\_\_\_\_(1)\_\_\_\_、\_\_\_\_(2)\_\_\_\_、\_\_\_\_(3)\_\_\_\_。

【解】 (1) 0001 0000 0001 1000B (2) 10030Q (3) 1018H

15. 若  $X = -107$ ， $Y = +74$ ，则按 8 位二进制可写出： $[X]_{\text{补}} = \text{____(1)____}$ ， $[Y]_{\text{补}} = \text{____(2)____}$ ， $[X + Y]_{\text{补}} = \text{____(3)____}$ 。

【解】 (1) 10010101B (2) 01001010B (3) 11011111B

16. 若  $X = -128$ ,  $Y = -1$ , 机器字长为 16 位, 则  $[X]_{\text{补}} = \underline{(1)}$ ,  $[Y]_{\text{补}} = \underline{(2)}$ ,  
 $[X + Y]_{\text{补}} = \underline{(3)}$ 。

【解】 (1) FF80H (2) FFFFH (3) FF7FH

17. 将十进制小数 0.65625 转换为对应的二进制数、八进制数和十六进制数, 其转换结果分别为  $\underline{(1)}$ 、 $\underline{(2)}$ 、 $\underline{(3)}$ 。

【解】 (1) 0.10101B (2) 0.52Q (3) 0.A8H

18.  $(1234)_{10} = \underline{(1)}_{16}$ ;  $571.375D = \underline{(2)}_H$ ;  $1011101.01B = \underline{(3)}_H$

【解】 (1) 4D2 (2) 23B.6 (3) 5D.4

19. 将二进制数 1001.101B、八进制数 35.54Q、十六进制数 FF.1H 转换为十进制数, 结果分别为  $\underline{(1)}$ 、 $\underline{(2)}$ 、 $\underline{(3)}$ 。

【解】 (1) 9.625D (2) 29.6875D (3) 255.0625D

20. 数制转换:  $247.86 = \underline{(1)}_H = \underline{(2)}_{\text{BCD}}$ ;

【解】 (1) F7.DC (2) 001001000111.10000110

21. 二进制数 11111010B 转换成压缩的 BCD 码的形式为  $\underline{(1)}$ 。

【解】 (1) 250H

22. 16 位的二进制数 0100 0001 0110 0011B, 与它等值的十进制数是  $\underline{(1)}$ ; 如果是压缩 BCD 码, 则表示的数是  $\underline{(2)}$ 。

【解】 (1) 16739 (2) 4163

23. 十进制数 255 的 ASCII 码可以表示为  $\underline{(1)}$ ; 用压缩型 BCD 码表示为  $\underline{(2)}$ ; 其十六进制数表示为  $\underline{(3)}$ 。

【解】 (1) 32H 35H 35H (2) 00000010 01010101B (3) 0FFH

24. 可将 36.25 用 IEEE 754 的单精度浮点格式表示成  $\underline{(1)}$ 。

【解】 (1) C2110000H

## 1.5 简答题

1. 冯·诺依曼体系结构有什么特点?

【解】 (1) 计算机由运算器、控制器、存储器、输入设备和输出设备五大部分组成。  
(2) 数据和程序以二进制编码形式存放。(3) 控制器采用根据存放在存储器中的程序串行顺序处理机制来工作。

2. 简述数据总线和地址总线各自具有的特点。如果某 CPU 的数据总线与地址总线采用同一组信号线, 则可以采用什么方法将地址总线分离出来?

【解】 数据总线的特点为双向三态, 其总线位数决定 CPU 与外部一次传输的位数。地址总线的特点为单向三态, 其总线位数决定 CPU 对外部寻址的范围。如果某 CPU 的数据总线与地址总线采用同一组信号线, 则可以采用锁存器将地址总线分离出来。

3. 试举例说明什么是压缩型 (或称组合型) BCD 码? 什么是非压缩型 (或称非组合型) BCD 码?

【解】 压缩型 BCD 码为 1 字节表示两位十进制数, 如 36H 表示 36。非压缩型 BCD 码

为 1 字节表示 1 位十进制数，其中高 4 位为 0，如 0306H 表示 36。

4. 在计算机中常采用哪几种数制？如何用符号表示？

【解】 在计算机中常采用二进制数、八进制数、十进制数、十六进制数等。为了明确所采用的数值，在相应数的末尾都采用对应的符号说明。其中十进制用 D (Decimal) 表示可以默认不写；八进制原为 Octonary，为避免与数字 0 混淆，用字母 Q 表示八进制；用 H (Hexadecimal) 表示十六进制。

5. 根据 ASCII 码的表示，试写出 0、9、F、f、A、a、CR、LF、\$ 等字符的 ASCII 码。

【解】 字符      0      9      F      f      A      a      CR      LF      \$  
ASCII 码   30H   39H   46H   66H   41H   61H   0DH   0AH   24H

6. 把下列英文单词转换成 ASCII 编码的字符串。

① How              ② Great              ③ Wter              ④ Good

【解】 ① 486F77H, ② 4772656174H, ③ 5761746572H, ④ 476F6664GH

7. 从键盘输入一个大写字母，如何转换为与其相对应的小写字母？从键盘输入十六进制数字符 0~F，如何转换为其相对应的二进制数 (00000000~00001111)？

【解】 从键盘输入一大写字母后，将其 ASCII 码加上 20H，就转换成了与其相对应的小写字母。从键盘输入十六进制数字符 0~9 后，将其 ASCII 码值减去 30H，就转换成了与其相对应的二进制数；从键盘输入十六进制数字符 A~F 后，将其 ASCII 码值减去 37H，就转换成了与其相对应的二进制数。

8. 将下列十进制数分别转换成二进制数、八进制数、十六进制数。

① 39              ② 54              ③ 127              ④ 119

【解】 ① 100111B, 47Q, 27H  
② 110110B, 66Q, 36H  
③ 1111111B, 177Q, 7FH  
④ 1110111B, 167Q, 77H

9. 8 位、16 位二进制数所表示的无符号数及补码的范围是多少？

【解】 8 位二进制无符号数表示的范围为 0~255，8 位二进制补码表示的范围为 -128~+127；16 位无符号二进制数表示的范围为 0~65535，16 位二进制补码表示的范围为 -32768~+32767。

10. 8 位、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？

【解】 原码：(-127~+127)，(-32767~+32767)；  
补码：(-128~+127)，(-32768~+32767)；  
反码：(-127~+127)，(-32767~+32767)

11. 将十进制数 146.25 转换为二进制，小数保留 4 位。

【解】 10010010.0100B

12. 将下列二进制数转换为十进制数，小数保留 4 位。

① 00001011.1101B      ② 1000110011.0101B      ③ 101010110011.1011B

【解】 ① 11.8125      ② 563.3125      ③ 2739.6875

13. 写出二进制数 1101.101B、十六进制数 2AE.4H、八进制数 42.54Q 的十进制数。

【解】 1101.101B = 13.625D, 2AE.4H = 686.25D, 42.57Q = 34.6875D



14. 试判断下列各组数据中哪个数据最大？哪个数据最小？

①  $A = 0.1001B$ ,  $B = 0.1001D$ ,  $C = 0.100H$       ②  $A = 10111101B$ ,  $B = 1001D$ ,  $C = 111H$

【解】 ①  $A$  最大,  $C$  最小, ②  $B$  最大,  $A$  最小

15. 简述求原码、反码、补码的规则。

【解】

1) 求原码的规则：正数的符号位为 0，负数的符号位为 1，其他位表示数的绝对值。

2) 求反码的规则：正数的反码与其原码相同，负数的反码为原码除符号位以外的各位取反。

3) 求补码的规则：正数的补码与其原码相同，负数的补码为反码在最低位上加 1。

16. 用补码计算  $(-56) - (-17)$ 。

【解】 此处运用补码加减运算公式  $[X \pm Y]_{\text{补}} = [X]_{\text{补}} + [\pm Y]_{\text{补}}$ ，令  $X = -56$ ,  $Y = -17$ ，且有

$[X]_{\text{原}} = 10111000B$ 、 $[X]_{\text{反}} = 11000111B$ 、 $[X]_{\text{补}} = 11001000B$

$[Y]_{\text{原}} = 10010001B$ 、 $[-Y]_{\text{补}} = 00010001B$

则

$$\begin{array}{r} [X]_{\text{补}} = 11001000B \\ + [-Y]_{\text{补}} = 00010001B \\ \hline [X - Y]_{\text{补}} = 11011001B \end{array}$$

得  $[X - Y]_{\text{原}} = 10100111B = -39$

17. 简述计算机在进行有符号补码运算中进位与溢出的区别。

【解】 进位为数据运算时的正常情况，其进位状态通过 CPU 中进位状态位的状态体现。溢出为运算结果超出了所能表示的数据范围，数据侵占了符号位。

18. 简述进行有符号补码运算判断是否产生溢出的方法。

【解】 判断溢出的方法有以下两种。

1) 双进位法：两个进位位分别为次高位向最高位的进位和最高位向进位位的进位。如果两个进位均有或均无，则无溢出。如果两个进位中 1 个有进位而另 1 个无进位，则一定有溢出。

2) 符号法：同号相减无溢出，同号相加时结果符号与加数符号相反有溢出，相同则无溢出。异号相加无溢出，异号相减时结果符号与减数符号相同有溢出，相反则无溢出。

19. 用 8 位二进制补码计算  $(-56) + (-117)$ ，并判断运算结果是否有溢出。

【解】 令  $X = -56$ ,  $Y = -117$

$[X]_{\text{原}} = 10111000B$ 、 $[X]_{\text{反}} = 11001000B$ 、 $[X]_{\text{补}} = 11001001B$

$[Y]_{\text{原}} = 11110101B$ 、 $[Y]_{\text{反}} = 10001010B$ 、 $[Y]_{\text{补}} = 10001011B$

则

$$\begin{array}{r} [X]_{\text{补}} = 11001001B \\ + [Y]_{\text{补}} = 10001011B \\ \hline [X]_{\text{补}} + [Y]_{\text{补}} = 01010011B \end{array}$$

得  $[X]_{\text{补}} + [Y]_{\text{补}} = [X + Y]_{\text{补}} = 01010011B$ ,  $X + Y = [[X + Y]_{\text{补}}]_{\text{补}} = 01010011B = +83$

从上面的运算式中可以看到，次高位向最高位无进位，而最高位向进位位有进位，所以

运算结果有溢出。从另一个角度来看，两个负数相加，结果为正数，其符号与减数的符号相反，所以运算结果有溢出。 $(-56) + (-117) = -173 \neq +83$ ，运算结果不正确，这是因为运算结果有溢出，也就是运算结果的数据位超出了所能表示的范围，侵占了符号位。

20. 试将 0.0875 用 IEEE-754 的单精度浮点格式表示。

【解】

$$0.0875 = 0.0001011001100110011001100110011B = 1.01100110011001100110011B \times 2^{-100B}$$

$$\text{尾数 } L = 01100110011001100110011B$$

$$\text{阶码} = 01111111B - 100B = 01111011B$$

$$0.0875 \text{ 的浮点表示} = 0 \ 01111011 \ 01100110011001100110011 \ B = 3DB33333H$$

机械工业出版社

# 第 2 章 微 处 理 器

## 2.1 学习指导

本章主要内容如下：

### 1. 8086/8088 微处理器的内部结构与工作原理

8086/8088 微处理器的内部分为执行单元（EU）和总线接口单元（BIU）两部分。执行单元负责完成指令的执行工作，总线接口单元负责完成预取指令和数据传输的工作。这两部分既相互独立工作，又相互配合。这种结构的优点是可实现流水作业，在执行指令的同时取下一条指令，提高了微处理器的工作效率。

### 2. 8086/8088 的内部寄存器

8086/8088 具有 14 个 16 位寄存器，包括 8 个通用寄存器（4 个数据寄存器 AX、BX、CX、DX，4 个指针寄存器 SP、BP、SI、DI），4 个段寄存器（CS、SS、DS、ES），1 个指令指针寄存器 IP 和 1 个状态控制寄存器 FR。每个寄存器都具有各自的特点。

1) 4 个数据寄存器具有双重性，可以存储 16 位数据（AX、BX、CX、DX），也可以拆成 8 个 8 位寄存器（AL、AH、BL、BH、CL、CH、DL、DH）进行 8 位运算。

2) 16 位的状态控制寄存器 FR 中有 9 位有效位，其中 6 位是状态标志（进位状态位 CF、辅助进位状态位 AF、符号状态位 SF、零状态位 ZF、奇偶状态位 PF 和溢出状态位 OF），3 位是控制标志（中断允许、方向、单步）。

3) 16 位的指令指针寄存器 IP 专为微处理器使用，不能通过指令进行访问。代码段寄存器 CS 只能读出，不能通过指令赋值。

4) 只有 BX、BP、SI、DI 可以作访问内存单元的地址指针。

### 3. 工作模式与引脚

1) 8086 CPU 和 8088 CPU 的主要特点见表 2-1。

表 2-1 8086 CPU 和 8088 CPU 的主要特点

		8088	8086
相同点		内部数据总线为 16 位，寄存器和指令系统完全兼容	
不同点	1	外部数据总线为 8 位	外部数据总线为 16 位
	2	指令队列缓冲器为 4 字节	指令队列缓冲器为 6 字节
	3	无 $\overline{\text{BHE}}$ 控制线 存储器和 I/O 选择控制线为 $\text{IO}/\overline{\text{M}}$	有高 8 位数据线传输控制线 $\overline{\text{BHE}}$ 存储器和 I/O 选择控制线为 $\text{M}/\overline{\text{IO}}$

2) 8086/8088 微处理器具有以下两种工作模式：最小工作模式和最大工作模式。

最小工作模式：即为构成单微处理器的简单系统，全部信号线均由 8086/8088 CPU 提

供，引脚  $\overline{MN}/\overline{MX}$  接 +5V 电压。

最大工作模式：即为构成多处理器的复杂系统。一般所构成的系统中除有一个主微处理器 8086/8088 外，还有两个协处理器：用于数值运算处理的 8087 和用于输入/输出设备服务的 8089，从而大大提高了主处理器的运行效率。在这种工作模式下，部分控制信号线是由 8288 总线控制器提供的，而不是由 8086/8088 CPU 直接提供全部信号线。引脚  $\overline{MN}/\overline{MX}$  接地。

3) 8086/8088 为 40 引脚 DIP 芯片，其中部分引脚采用了复用技术，包括分时复用（即在一些时刻提供一种信息，而在另一些时刻提供另一种信息）和分状态复用（即输入与输出定义不同、高电平与低电平定义不同、不同模式下定义不同）等。分时复用的信号可以通过锁存器实现分离。另外，8086 与 8088 中有部分引脚定义不同，主要引脚包括（以 8086 为例）AD0 ~ AD15、A16/S3 ~ A19/S6、 $\overline{MN}/\overline{MX}$ 、ALE、 $\overline{BHE}$ 、RESET、 $\overline{RD}$ 、 $\overline{WR}$ 、 $\overline{M}/\overline{IO}$  等。8086/8088 对存储器或 IO 操作信号如图 2-1 所示。

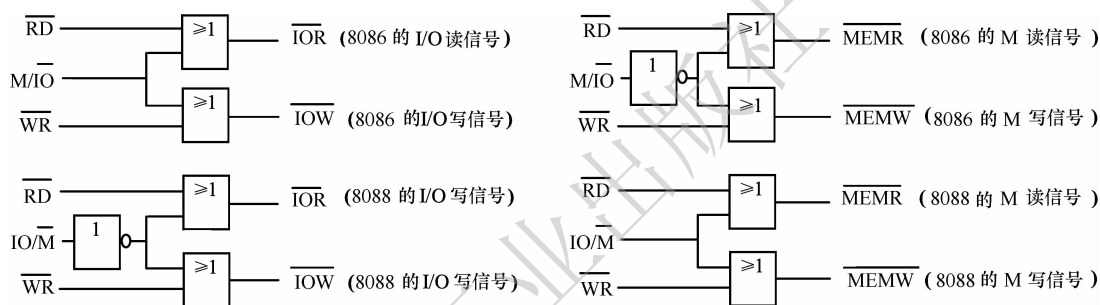


图 2-1 8086/8088 对存储器或 IO 操作信号

#### 4. 存储器组织

8086/8088 采用分段管理的概念。

1) 所有的数据以字节形式存放在存储单元中，每一个单元均占一个地址，任何两个相邻字节都可以构成一个字（word），占用两个地址。用地址值较小的那个字节单元地址作为这个字单元的地址。一个 16 位字中低 8 位数据存在较小地址的单元中，而高 8 位数据存在地址较高的单元中。

2) 每一个单元均有一个唯一的 20 位地址，称为物理地址。

3) 在软件中 8086/8088 对存储器采用分段描述的方法，即将整个存储区划分很多的段，每一个段的大小可各不相同，但均  $\leq 64\text{KB}$ ，各段之间可以是紧密连接、可以是部分或完全重叠、也可以是不相关，每个存储单元可用不唯一的逻辑地址描述（段基值：偏移量）。

4) 物理地址与逻辑地址的关系为：物理地址(20 位) = 段基值(16 位)  $\times 16$  + 偏移量(16 位)。

5) 8086 的存储器为了满足既可以 16 位操作又可以 8 位操作，采用奇、偶两个存储体的结构。

#### 5. 堆栈组织

堆栈是一个特定的存储区，它的特点是：一端是固定的，另一端是活动的，而所有的信息存取都在活动的一端进行。堆栈操作的原则是后进先出。栈操作包括的内容如下：

- 1) 设置栈指针 (设置 SS、SP) 和栈容量 (栈长度)。
- 2) 数据进栈操作 (PUSH 指令)。
- 3) 数据出栈操作 (POP 指令)。

8086/8088 的栈区操作为 16 位数据操作, 且默认通过 SS: SP 访问, 进栈操作时栈指针 SP 减 2, 出栈操作时栈指针 SP 加 2。

## 6. 时序

处理器的周期状态可以分为 3 种: 时钟周期 (处理器完成一个微操作所需的时间, 也就是处理器的基本时间计量单位)、总线周期 (处理器完成一个基本操作所用的时间) 和指令周期 (处理器执行一条指令所需的时间)。最基本的读/写总线周期由 4 个时钟周期组成。

## 2.2 单项选择题

1. 关于 8088 CPU 和 8086 CPU 的对比, 错误的叙述是 ( )。
  - A. 8088 CPU 和 8086 CPU 的地址线数量相同
  - B. 8088 CPU 和 8086 CPU 的片内数据线数量相同
  - C. 8088 CPU 和 8086 CPU 的片外数据线数量相同
  - D. 8088 CPU 和 8086 CPU 的寄存器数量相同

【解】 C

2. 关于 8088 CPU 和 8086 CPU 的对比, 正确的叙述是 ( )。
  - A. 8088 CPU 和 8086 CPU 的地址线位数不相同
  - B. 8088 CPU 和 8086 CPU 的片内数据线位数不相同
  - C. 8088 CPU 和 8086 CPU 的片外数据线位数不相同
  - D. 8088 CPU 和 8086 CPU 的寄存器数量不相同

【解】 C

3. 8086 为 16 位的 CPU, 说明 ( )。
  - A. 8086 CPU 内有 16 条数据线
  - B. 8086 CPU 内有 16 个寄存器
  - C. 8086 CPU 外有 16 条地址线
  - D. 8086 CPU 外有 16 条控制线

【解】 A

4. 80386 微型计算机是 32 位机, 根据是它的 ( )。
  - A. 地址线是 32 位的
  - B. 数据线为 32 位的
  - C. 寄存器是 32 位的
  - D. 地址线和数据线都是 32 位的

【解】 D

5. 下列不是 8086/8088 CPU 数据总线作用的为 ( )。
  - A. 用于传送指令机器码
  - B. 用于传送立即数
  - C. 用于传送偏移地址量
  - D. 用于传送控制信号

【解】 D

6. 关于 8088 CPU 叙述不正确的是 ( )。
  - A. 片内有 14 个 16 位寄存器
  - B. 片内有 1MB 的存储器
  - C. 片内有 4 字节队列缓冲器
  - D. 片外有 8 位数据总线

【解】 B

7. 8086/8088 CPU 的地址总线宽度为 20 位, 它对存储器的寻址范围为 ( )。

- A. 20KB                      B. 64KB                      C. 1MB                      D. 20MB

【解】 C

8. 8086/8088 CPU 的地址总线宽度为 20 位, 它对 I/O 接口的寻址范围为 ( )。

- A. 20KB                      B. 64KB                      C. 1MB                      D. 20MB

【解】 B

9. 8086/8088 CPU 从功能结构上看是由 ( ) 组成的。

- A. 控制器和运算器                      B. 控制器、运算器和寄存器  
C. 控制器和 20 位物理地址加法器                      D. 执行单元和总线接口单元

【解】 D

10. 8086/8088 CPU 内部具有 ( ) 个 16 位寄存器。

- A. 4                      B. 8                      C. 14                      D. 20

【解】 C

11. 8086/8088 CPU 内部具有 ( ) 个 8 位寄存器。

- A. 4                      B. 8                      C. 14                      D. 20

【解】 B

12. 8086/8088 CPU 的标志寄存器 FR 中有 ( ) 个有效位。

- A. 1                      B. 3                      C. 6                      D. 9

【解】 D

13. 8086/8088 CPU 的标志寄存器 FR 中状态标志位有 ( ) 位。

- A. 1                      B. 3                      C. 6                      D. 9

【解】 C

14. 8086/8088 CPU 的标志寄存器 FR 中控制标志位有 ( ) 位。

- A. 1                      B. 3                      C. 6                      D. 9

【解】 B

15. 有符号数溢出时, 判断依据是 ( )。

- A. CF = 1                      B. OF = 1                      C. AF = 1                      D. SF = 1

【解】 B

16. 8086/8088 CPU 有 ( ) 个 16 位的段寄存器。

- A. 2                      B. 4                      C. 8                      D. 16

【解】 B

17. 指令指针寄存器 IP 的作用是 ( )。

- A. 保存将要执行的下一条指令所在的位置  
B. 保存 CPU 要访问的内存单元地址  
C. 保存运算器的运算结果  
D. 保存正在执行的一条指令所在的位置

【解】 A

18. 8088 CPU 的指令队列缓冲器由 ( ) 组成。

- A. 1 字节移位寄存器  
B. 4 字节移位寄存器  
C. 6 字节移位寄存器  
D. 8 字节移位寄存器

【解】 B

19. 8086 CPU 的指令队列缓冲器由 ( ) 组成。

- A. 1 字节移位寄存器  
B. 4 字节移位寄存器  
C. 6 字节移位寄存器  
D. 8 字节移位寄存器

【解】 C

20. 指令队列具有 ( ) 的作用。

- A. 暂存操作数地址  
B. 暂存操作数  
C. 暂存指令地址  
D. 暂存预取指令

【解】 D

21. 8086/8088 CPU 对存储器采用分段管理的方法, 每个存储单元均拥有 ( ) 两种地址。

- A. 实地址和虚拟地址  
B. 20 位地址和 16 位地址  
C. 逻辑地址和物理地址  
D. 段基址和偏移地址

【解】 C

22. 在 8086 系统中, 每个逻辑段的存储单元数最多为 ( )。

- A. 1MB  
B. 256B  
C. 64KB  
D. 根据需要而定

【解】 C

23. 在 8086/8088 CPU 中, 由逻辑地址形成存储器物理地址的方法是 ( )。

- A. 段基值 + 偏移地址  
B. 段基值  $\times 16$  + 偏移地址  
C. 段基值  $\times 16H$  + 偏移地址  
D. 段基值  $\times 10$  + 偏移地址

【解】 B

24. 在 8086/8088 CPU 中, 确定下一条指令的物理地址应为 ( )。

- A.  $CS \times 16 + IP$   
B.  $DS \times 16 + SI$   
C.  $SS \times 16 + SP$   
D.  $ES \times 16 + DI$

【解】 A

25. 8086/8088 CPU 上电和复位后, 下列寄存器的值正确的是 ( )。

- A.  $CS = 0000H, IP = 0000H$   
B.  $CS = 0000H, IP = FFFFH$   
C.  $CS = FFFFH, IP = 0000H$   
D.  $CS = FFFFH, IP = FFFFH$

【解】 C

26. RESET 信号有效后, 8086 CPU 将执行的第一条指令地址为 ( )。

- A. 00000H  
B. 0FFFFH  
C. 0FFFF0H  
D. 0FFFFFFH

【解】 C

27. 当 RESET 信号为高电平时, 寄存器初值为 FFFFH 的是 ( )。

- A. CS  
B. ES  
C. IP  
D. BP

【解】 A

28. 在 8086/8088 系统中, 某存储单元的物理地址为 24680H, 与其不对应的逻辑地址为 ( )。

- A. 4680H: 2000H  
B. 2468H: 0000H  
C. 2460H: 0080H

D. 2400H; 0680H

E. 2000H; 4680H

【解】 A

29. 若某指令存放在代码段为 CS = 789A H、指令指针为 IP = 2345H 处, 则该指令存放单元的物理地址是 ( )。

A. 0H

B. 7ACE5H

C. 2ACEAH

D. 9BDF01H

【解】 B

30. 若某 8 位数据存放在 2300H: 2300H 处, 则该 8 位数据存放单元的物理地址是 ( )。

A. 23000H

B. 23230H

C. 23023H

D. 25300H

【解】 D

31. 若某存储单元的物理地址为 ABCDEH, 则 ( ) 不是其相应的逻辑地址。

A. ABCDH; 000EH

B. ABC0H; 00DEH

C. AB00H; 0CDEH

D. A000H; 0CDEH

【解】 D

32. 下列逻辑地址中对应不同的物理地址的是 ( )。

A. 0400H; 0340H

B. 0420H; 0140H

C. 03E0H; 0740H

D. 03C0H; 0740H

【解】 C

33. 8086/8088 CPU 存放当前指令的存储单元的逻辑地址为 ( )。

A. DS; BX

B. SS; SP

C. CS; PC

D. CS; IP

【解】 D

34. 对微处理器而言, 它的每条指令都有一定的时序, 其时序关系是 ( )。

A. 一个时钟周期包括几个机器周期, 一个机器周期包括几个指令周期

B. 一个机器周期包括几个指令周期, 一个指令周期包括几个时钟周期

C. 一个指令周期包括几个机器周期, 一个机器周期包括几个时钟周期

D. 一个指令周期包括几个时钟周期, 一个时钟周期包括几个机器周期

【解】 C

35. 在 8086/8088 CPU 中, 时钟周期、指令周期和总线周期按费时长短的排列是 ( )。

A. 时钟周期 > 指令周期 > 总线周期

B. 时钟周期 > 总线周期 > 指令周期

C. 指令周期 > 总线周期 > 时钟周期

D. 总线周期 > 指令周期 > 时钟周期

【解】 C

36. 8086/8088 CPU 的地址有效发生在总线周期的 ( ) 时刻。

A. T1

B. T2

C. T3

D. T4

【解】 A

37. 8086/8088 CPU 的读数据操作发生在总线周期的 ( ) 时刻。

A. T1

B. T1, T2

C. T2, T3

D. T3, T4

【解】 D

38. 8086/8088 CPU 的写数据操作发生在总线周期的 ( ) 时刻。

A. T1

B. T2

C. T2, T3

D. T2, T3, T4

【解】 D



39. 当控制线  $\text{READY} = 0$  时, 应在 ( ) 插入等待周期  $T_w$ 。

- A.  $T_1$  和  $T_2$  间      B.  $T_2$  和  $T_3$  间      C.  $T_3$  和  $T_4$  间      D. 任何时候

【解】 C

40. CPU 对  $\text{INTR}$  中断请求响应过程是执行 ( )  $\text{INTA}$  总线周期。

- A. 1 个      B. 2 个      C. 3 个      D. 4 个

【解】 B

41. 下列说法中属于最小工作模式特点的是 ( )。

- A. CPU 提供全部的控制信号      B. 由编程进行模式设定  
C. 不需要 8286 收发器      D. 需要总线控制器 8288

【解】 A

42. 下列器件中在 8086 CPU 最小模式下不需要的是 ( )。

- A. 时钟发生器      B. 地址锁存器      C. 总线控制器      D. 总线驱动器

【解】 C

43. 下列说法中属于最大工作模式特点的是 ( )。

- A. CPU 提供全部的控制信号      B. 由编程进行模式设定  
C. 需要 8286 收发器      D. 需要总线控制器 8288

【解】 D

44. 在 8088/8086 系统中, 读/写一个字时,  $\text{A0}$ 、 $\overline{\text{BHE}}$  应该分别为 ( )。

- A. 0、0      B. 0、1      C. 1、0      D. 1、1

【解】 A

45. 当 8086 CPU 的控制线  $\overline{\text{BHE}} = 0$ 、地址线  $\text{A0} = 0$  时, 将实现 ( )。

- A. 传送地址为偶地址的 8 位内存数据      B. 传送地址为偶地址的 16 位内存数据  
C. 传送地址为奇地址的 8 位内存数据      D. 传送地址为奇地址的 16 位内存数据

【解】 B

46. 当 8086 CPU 的控制线  $\overline{\text{BHE}} = 0$ 、地址线  $\text{A0} = 1$  时, 将实现 ( )。

- A. 传送地址为偶地址的 8 位内存数据      B. 传送地址为偶地址的 16 位内存数据  
C. 传送地址为奇地址的 8 位内存数据      D. 传送地址为奇地址的 16 位内存数据

【解】 C

47. 数据总线驱动电路采用的基本逻辑单元是 ( )。

- A. 反相器      B. 触发器      C. 三态门      D. 译码器

【解】 C

48. 8086/8088 CPU 数据总线和部分地址总线采用分时复用技术, 系统中可通过基本逻辑单元 ( ) 获得稳定的地址信息。

- A. 译码器      B. 触发器      C. 锁存器      D. 三态门

【解】 C

49. 在 8088 CPU 构成的系统中, 需要 ( ) 片 8286 数据总线收发器。

- A. 1      B. 2      C. 8      D. 16

【解】 A

50. 在 8086 CPU 构成的系统中, 需要 ( ) 片 8286 数据总线收发器。

- A. 1                      B. 2                      C. 8                      D. 16

【解】 B

51. 在 8086/8088 CPU 中, 控制线  $\overline{RD}$  和  $\overline{WR}$  的作用是 ( )。

- A. CPU 控制数据传输的方向                      B. CPU 实现存储器存取操作控制  
C. CPU 实现读或写操作时的控制线                      D. CPU 实现读地址/数据线分离控制

【解】 C

52. 在 8086/8088 CPU 中, 控制线  $DT/\overline{R}$  的作用是 ( )。

- A. 数据传输方向的控制                      B. 存储器存取操作控制  
C. 数据传输有效控制                      D. 地址/数据线分离控制

【解】 A

53. 在 8086/8088 CPU 中, 控制线 ALE 的作用是 ( )。

- A. CPU 发出的数据传输方向控制信号  
B. CPU 发出的数据传输有效控制信号  
C. CPU 发出的存储器存取操作控制信号  
D. CPU 发出的地址有效信号

【解】 D

54. 在 8086/8088 CPU 中, 控制线  $\overline{DEN}$  的作用是 ( )。

- A. CPU 发出的数据传输方向控制信号  
B. CPU 发出的数据传输有效控制信号  
C. CPU 发出的存储器存取操作控制信号  
D. CPU 发出的地址有效信号

【解】 B

55. 在 8086/8088 CPU 中, 可屏蔽中断请求的控制线是 ( )。

- A. NMI                      B. HOLD                      C. INTR                      D.  $\overline{INTA}$

【解】 C

56. 在 8086/8088 CPU 中, 可屏蔽中断响应的控制线是 ( )。

- A. NMI                      B. HOLD                      C. INTR                      D.  $\overline{INTA}$

【解】 D

57. 在 8086/8088 CPU 中, 非屏蔽中断请求的控制线是 ( )。

- A. NMI                      B. HOLD                      C. INTR                      D.  $\overline{INTA}$

【解】 A

58. 在 8086/8088 CPU 中, 与 DMA 操作有关的控制线是 ( )。

- A. NMI                      B. HOLD                      C. INTR                      D.  $\overline{INTA}$

【解】 B

59. 当 8086/8088 CPU 为最小工作方式时, 控制线  $MN/\overline{MX}$  应接 ( )。

- A. 低电平                      B. 高电平                      C. 下降沿脉冲                      D. 上升沿脉冲

【解】 B

60. 若 8086 CPU 访问 I/O 端口时, 控制线  $M/\overline{IO}$  应输出 ( )。

- A. 低电平                      B. 高电平                      C. 下降沿脉冲                      D. 上升沿脉冲

【解】 A

61. 8086 CPU 可访问 ( ) 个字节 I/O 端口。

- A. 1KB                      B. 32KB                      C. 64KB                      D. 1MB

【解】 C

62. 8086 CPU 可访问 ( ) 个字 I/O 端口。

- A. 1K                      B. 32K                      C. 64K                      D. 1M

【解】 B

63. 当 8088 CPU 从存储器单元读数据时, 有 ( )。

- A.  $\overline{RD}=0$ 、 $IO/\overline{M}=0$                       B.  $\overline{RD}=0$ 、 $IO/\overline{M}=1$   
C.  $\overline{RD}=1$ 、 $IO/\overline{M}=0$                       D.  $\overline{RD}=1$ 、 $IO/\overline{M}=1$

【解】 A

64. 当 8088 CPU 向 I/O 端口输出数据时, 有 ( )。

- A.  $\overline{WR}=0$ 、 $IO/\overline{M}=0$                       B.  $\overline{WR}=1$ 、 $IO/\overline{M}=0$   
C.  $\overline{WR}=0$ 、 $IO/\overline{M}=1$                       D.  $\overline{WR}=1$ 、 $IO/\overline{M}=1$

【解】 C

65. 对堆栈进行数据存取的原则是 ( )。

- A. 先进先出                      B. 后进先出                      C. 随机存取                      D. 都可以

【解】 B

66. 8086/8088 CPU 将数据压入堆栈时, 栈区指针的变化为 ( )。

- A. SS 内容改变、SP 内容不变                      B. SS 内容不变、SP 内容加 2  
C. SS 内容不变、SP 内容减 2                      D. SS 和 SP 内容都改变

【解】 C

67. 8086/8088 CPU 将数据从堆栈中弹出时, 栈区指针的变化为 ( )。

- A. SS 内容改变、SP 内容不变                      B. SS 内容不变、SP 内容减 2  
C. SS 内容不变、SP 内容加 2                      D. SS 和 SP 内容都改变

【解】 C

## 2.3 判断题

1. 8086 CPU 中包含了寄存器和存储器。( )
2. 寄存器寻址比存储器寻址的运算速度快。( )
3. 存储器是计算机系统中不可缺少的部分。( )
4. 8086/8088 CPU 的片内数据线和片外数据线的宽度均为 16 位。( )
5. 8086/8088 CPU 为 16 位处理器, 一次可并行传送 8 位或 16 位二进制信息。( )
6. 8086 CPU 的数据总线和地址总线都是 20 位。( )
7. 80486 CPU 的数据总线和地址总线都是 32 位。( )
8. 8086/8088 CPU 对外部存储器和 I/O 端口的寻址范围为 1MB。( )
9. 8086/8088 CPU 内部分为两个功能模块: 执行单元 EU 和总线接口单元 BIU。( )
10. 8086/8088 CPU 的 EU 直接通过外部总线读取指令后执行。( )

11. 8086 CPU 的 BIU 直接经外部总线读取数据。( )
12. 8086/8088 CPU 的 BIU 中包含一个 6 字节指令队列。( )
13. 8086/8088 CPU 在执行转移指令时, 指令队列中的原内容不变。( )
14. 8086/8088 CPU 指令队列满足先进后出的原则。( )
15. 8086/8088 CPU 的 BIU 中包含一个 16 位的地址加法器。( )
16. 因为 8086 存储单元的段基值和偏移地址均为 16 位, 所以 8086 存储单元的地址线为 32 位。( )
17. 8086/8088 CPU 中为用户提供了 14 个 16 位的可读/写的寄存器。( )
18. 8086/8088 CPU 可以通过改变指令指针 IP 的内容来改变指令执行顺序。( )
19. 在 8086/8088 系统中, 用户可以通过指令改变指令指针 IP 的内容。( )
20. 8086/8088 CPU 的 16 位标志寄存器 FR 中的每位均有确定含义。( )
21. 在 8086/8088 CPU 中, 当两个数的运算结果为零时, 状态标志位  $ZF=0$ 。( )
22. 在 8086/8088 CPU 中, 当两个数做加减运算, 结果有进位时, 状态标志位  $CF=1$ 。( )
23. 在 8086/8088 CPU 中, 当两个符号数的运算结果产生溢出时, 状态标志位  $OF=1$ 。( )
24. 在 8086/8088 CPU 中, 当两个数做加减运算后, 结果最高位为 1 时, 状态标志位  $SF=1$ 。( )
25. 8086/8088 CPU 的基本读/写总线周期由 4 个时钟周期组成。( )
26. 8086/8088 CPU 在总线周期的  $T_1$  时刻, 从地址/数据提供数据信息。( )
27. 在总线周期中, 等待状态周期  $T_w$  仅能出现在  $T_3$  状态和  $T_4$  状态之间。( )
28. 在总线周期中, 空闲状态周期  $T_i$  仅能出现在  $T_3$  状态之后。( )
29. 当控制线  $READY$  输出低电平时, 等待状态周期  $T_w$  才会出现。( )
30. 8086/8088 CPU 在一个存储单元中, 可存入 8 位数据或者 16 位数据。( )
31. 在 8086/8088 系统中, 每个存储单元均具有唯一的物理地址和逻辑地址。( )
32. 8086/8088 CPU 允许多个逻辑段重叠或交叉。( )
33. 8088 CPU 将 1MB 的存储空间分为奇地址存储体和偶地址存储体。( )
34. 8086 CPU 将 1MB 的存储空间分为两个 512KB 的存储体。( )
35. 在 8086 系统中, 若地址线  $A_0=0$ , 则 512KB 的偶存储体操作有效。( )
36. 在 8086 系统中, 字数据的低 8 位存放在偶存储体, 高 8 位存放在奇存储体。( )
37. 在 8086/8088 系统中, 存储器奇地址存储体的片选有效控制信号由控制线  $BHE$  提供。( )
38. 在 8086 系统中, 若  $\overline{BHE}=0$ 、 $A_0=0$ , 则一个总线周期可完成 16 位数据的操作。( )
39. 在 8086/8088 系统的字存储中, 低地址存字的高 8 位, 高地址存字的低 8 位。( )
40. 8086/8088 CPU 有 16 根地址/数据分时复用引脚。( )
41. 8086/8088 CPU 的地址/数据复用线可通过缓冲器分离出地址信息。( )
42. 在 8086/8088 CPU 中, 可利用地址有效控制线  $ALE$  对地址/数据复用线进行锁存, 获取地址信息。( )
43. 8086/8088 CPU 的控制线  $\overline{DEN}$  提供数据传输有效信号。( )

44. 8086/8088 CPU 的控制线  $\overline{RD}$  和  $\overline{WR}$  提供对芯片外部实现读/写操作信号。( )
45. 8086/8088 CPU 响应可屏蔽中断 INTR 的条件是标志位 IF 置 1。( )
46. 8086/8088 CPU 响应不可屏蔽中断 NMI 请求的条件是标志位 IF 置 0。( )
47. 8086/8088 CPU 可屏蔽中断 INTR 的中断请求信号为高电平有效。( )
48. 8086/8088 CPU 在上电或 RESET 有效时, 所有寄存器为 0000H。( )
49. 堆栈操作的原则是后进先出。( )
50. 堆栈指针 SP 总是指向堆栈的栈顶。( )
51. 8086/8088 系统中的进栈操作时栈指针 SP 加 2, 出栈操作时栈指针 SP 减 2。( )
52. 在 8086/8088 系统中, 在执行调用指令或中断响应时, 断点会自动进栈加以保护。( )
53. 存储器和 I/O 统一编址时, 不需要单独的 I/O 操作指令。( )

### 【答案】

- |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1. ×  | 2. ✓  | 3. ✓  | 4. ×  | 5. ×  | 6. ×  | 7. ✓  | 8. ×  | 9. ✓  |
| 10. × | 11. ✓ | 12. × | 13. × | 14. × | 15. × | 16. × | 17. × | 18. ✓ |
| 19. × | 20. × | 21. × | 22. ✓ | 23. ✓ | 24. ✓ | 25. ✓ | 26. × | 27. ✓ |
| 28. × | 29. ✓ | 30. × | 31. × | 32. ✓ | 33. × | 34. ✓ | 35. ✓ | 36. × |
| 37. × | 38. ✓ | 39. × | 40. × | 41. × | 42. ✓ | 43. ✓ | 44. ✓ | 45. ✓ |
| 46. × | 47. ✓ | 48. × | 49. ✓ | 50. ✓ | 51. × | 52. ✓ | 53. ✓ |       |

## 2.4 填空题

1. 8086/8088 CPU 的内部由两个功能单元组成, 即 (1) 和 (2)。

【解】(1) 执行单元 EU (2) 总线接口单元 BIU

2. 在 8086 CPU 中, 由于 BIU 和 EU 分开, 因此 (1) 和 (2) 可以重叠操作, 提高了 CPU 的利用率。

【解】(1) 取指令 (2) 执行指令

3. 在 8086 中, BIU 部件完成 (1) 功能, EU 部件完成 (2) 功能。

【解】(1) 总线接口 (2) 指令的译码及执行

4. 8086 CPU 的指令队列由 (1) 移位寄存器组成, 8088 CPU 的指令队列由 (2) 移位寄存器组成。指令队列的作用是 (3)。

【解】(1) 6 字节 (2) 4 字节 (3) 存放预取的指令

5. 8086 CPU 的内部数据总线宽度为 (1) 位、外部数据总线宽度为 (2) 位。8088 CPU 的内部数据总线宽度为 (3) 位、外部数据总线宽度为 (4) 位。

【解】(1) 16 (2) 16 (3) 16 (4) 8

6. 在 8086/8088 CPU 中, 执行单元 EU 中的运算单元 ALU 完成的工作是 (1) 运算、(2) 运算和 (3) 运算。

【解】(1) 算术 (2) 逻辑 (3) 16 位段内偏移地址

7. 8086/8088 CPU 中有 (1) 个 (2) 位的寄存器。其中, 称 AX、BX、CX 和 DX 为 (3) 寄存器, 称 SP、BP、SI 和 DI 为 (4) 寄存器, 称 CS、DS、SS 和 ES 为 (5) 寄存器, 称 IP 为 (6) 寄存器, 称 FR 为 (7) 寄存器。

【解】 (1) 14 (2) 16 (3) 通用数据 (4) 通用地址  
(5) 段 (6) 指令指针 (7) 标志

8. 8086/8088 CPU 中有 8 个用于 8 位运算的通用寄存器, 它们是 (1)、(2)、(3)、(4)、(5)、(6)、(7)、(8)。

【解】 (1) AH (2) AL (3) BH (4) BL (5) CH (6) CL (7) DH (8) DL

9. 8086/8088 CPU 常用于定义堆栈段的寄存器为 (1), 用 (2) 寄存器作堆栈指针。

【解】 (1) SS (2) SP

10. 8086/8088 CPU 可用于存放存储单元偏移量的寄存器为 (1)、(2)、(3)、(4)。

【解】 (1) SI (2) DI (3) BX (4) BP

11. 8086/8088 CPU 在串操作指令中时, 规定 (1) 寄存器存放源操作数的段基值、(2) 寄存器存放目标操作数的段基值、(3) 寄存器作为源操作数的指针、(4) 寄存器作为目的操作数的指针。

【解】 (1) DS (2) ES (3) SI (4) DI

12. 8086/8088 CPU 对存储单元地址的描述有以下两种: 一种是在总线上唯一的 (1), 用 (2) 位二进制或 (3) 位十六进制数表示; 另一种是在程序中多样化的 (4), 用 (5) 和 (6) 表示。

【解】 (1) 物理地址 (2) 20 (3) 5  
(4) 逻辑地址 (5) 16 位二进制数的段基值 (6) 16 位二进制数的偏移量

13. 在 8086/8088 系统中, 若某存储器单元的物理地址为 2ABCDH, 且该存储器单元所在的段基值为 2A12H, 则该存储器单元的偏移地址应为 (1)。

【解】 (1) 0AADH

14. 在 8086/8088 系统中, 字数据存放在 (1) 个存储单元中, 以 (2) 地址作为该字数据的地址。

【解】 (1) 2 (2) 较小

15. 在 8086 系统中, 如果字数据 8BF0H 存放在偶地址开始的 2 个存储单元内, 则可以通过 (1) 个总线周期实现 16 位字数据访问。如果字数据 8BF0H 存放在奇地址开始的存储单元内, 则需要 (2) 个总线周期数完成 16 位字数据读取操作。

【解】 (1) 1 (2) 2

16. 8086/8088 CPU 在上电或复位后, 寄存器中的值处于初始态, 此时 CS = (1), IP = (2), DS = (3)。

【解】 (1) FFFFH (2) 0000H (3) 0000H

17. 8086/8088 构成的系统, 在开机或复位时, 第一条执行的指令所在存储单元的物理地址为 (1)。

【解】 (1) FFFF0H

18. 8086/8088 CPU 的标志寄存器 FR 中有 3 个控制标志位, 分别是 (1)、(2)、(3); 有 6 个状态标志位, 分别是 (4)、(5)、(6)、(7)、

\_\_\_\_(8)\_\_\_\_、\_\_\_\_(9)\_\_\_\_、

【解】 (1) IF (2) DF (3) TF (4) AF (5) CF  
(6) OF (7) SF (8) ZF (9) PF

19. 8086/8088 CPU 中与中断操作有关的操作标志位是\_\_\_\_(1)\_\_\_\_，与串操作有关的操作标志位是\_\_\_\_(2)\_\_\_\_，与单步调试操作有关的操作标志位是\_\_\_\_(3)\_\_\_\_。

【解】 (1) IF (2) DF (3) TF

20. 若 AL = 4AH、BL = 86H，试问在执行 ADD AL、BL 指令后 CF = \_\_\_\_ (1) \_\_\_\_、AF = \_\_\_\_ (2) \_\_\_\_、SF = \_\_\_\_ (3) \_\_\_\_、ZF = \_\_\_\_ (4) \_\_\_\_、OF = \_\_\_\_ (5) \_\_\_\_、PF = \_\_\_\_ (6) \_\_\_\_。

【解】 (1) 0 (2) 1 (3) 1 (4) 0 (5) 0 (6) 0

21. 8086/8088 CPU 将所能寻址的 1MB 存储空间分为\_\_\_\_(1)\_\_\_\_段，每段存储容量最多为\_\_\_\_(2)\_\_\_\_。

【解】 (1) 若干个 (2) 64KB

22. 8086CPU 有\_\_\_\_(1)\_\_\_\_根地址线，可直接寻址的存储器容量为\_\_\_\_(2)\_\_\_\_字节，在访问 I/O 端口时，使用地址线\_\_\_\_(3)\_\_\_\_，最多可寻址\_\_\_\_(4)\_\_\_\_个 I/O 端口。

【解】 (1) 20 (2) 1M (3) 16 条 (4) 64K

23. 8086 CPU 执行存储器读/写指令时，控制线  $\overline{M}/IO$  输出\_\_\_\_(1)\_\_\_\_电平，执行输入/输出指令时，控制线  $\overline{M}/IO$  输出\_\_\_\_(2)\_\_\_\_电平。

【解】 (1) 高 (2) 低

24. 32 位的逻辑地址 589AH: 3210H 表示的 20 位物理地址为\_\_\_\_(1)\_\_\_\_。

【解】 (1) 5BBB0H

25. 8086 CPU 的堆栈操作原则为\_\_\_\_(1)\_\_\_\_，指令队列操作原则为\_\_\_\_(2)\_\_\_\_。

【解】 (1) 后进先出 (2) 先进先出

26. 若堆栈栈顶指针 SP = 2000H，则执行 5 条入栈指令和 2 条出栈指令后，SP = \_\_\_\_ (1) \_\_\_\_。

【解】 (1) 1FFAH

27. 在计算机系统中，对 I/O 端口地址的编码方式有\_\_\_\_(1)\_\_\_\_和\_\_\_\_(2)\_\_\_\_两种方式。

【解】 (1) 统一编址 (2) 独立编址

28. 8086 CPU 地址/数据复用线可通过\_\_\_\_(1)\_\_\_\_分离出地址信息，此时控制线 ALE 应输出\_\_\_\_(2)\_\_\_\_电平。

【解】 (1) 锁存器 (2) 高

29. 若 8086/8088 CPU 工作于最小工作方式，则控制线  $\overline{MN}/\overline{MX}$  应接\_\_\_\_(1)\_\_\_\_电平；若 8086/8088 CPU 工作于最大工作方式，则控制线  $\overline{MN}/\overline{MX}$  应接\_\_\_\_(2)\_\_\_\_电平。

【解】 (1) 高 (2) 低

30. 8086 中引脚 BHE 信号有效的含义表示\_\_\_\_(1)\_\_\_\_。

【解】 (1) 高 8 位数据线 D15 ~ D8 有效

31. 当 8086/8088 CPU 访问外部数据时，控制线  $\overline{DEN}$  应输出\_\_\_\_(1)\_\_\_\_电平；在从外部读入数据时，控制线  $\overline{DT}/R$  应输出\_\_\_\_(2)\_\_\_\_电平；再将数据输出到外部时，控制线  $\overline{DT}/R$  应输出\_\_\_\_(3)\_\_\_\_电平。

【解】 (1) 低 (2) 低 (3) 高

32. 当 8086/8088 CPU 在进行写数据操作时，控制线  $\overline{RD}$  应输出\_\_\_\_(1)\_\_\_\_电平，控制线  $\overline{WR}$

应输出\_\_ (2) \_\_ 电平。

【解】 (1) 高 (2) 低

33. 8086 微处理器在最小模式下, 用\_\_ (1) \_\_ 来控制输出地址是访问内存还是访问 I/O。

【解】 (1) M/IO

34. 8086 CPU 可访问的存储器空间为\_\_ (1) \_\_, 可访问字节 I/O 空间为\_\_ (2) \_\_, 可访问字 I/O 空间为\_\_ (3) \_\_。

【解】 (1) 1MB (2) 64KB (3) 32KB

35. 8086 CPU 的 MN/MX 引脚的作用是\_\_ (1) \_\_。

【解】 (1) 决定 CPU 工作在什么模式 (最小/最大)

36. 8086 CPU 中典型总线周期由\_\_ (1) \_\_ 个时钟周期组成, 其中 T1 期间, CPU 输出\_\_ (2) \_\_ 信息; 如有必要时, 可以在\_\_ (3) \_\_ 两个时钟周期之间插入 1 个或多个  $T_w$  等待周期。

【解】 (1) 4 (2) 地址 (3) T3 和 T4

37. 当存储器的读取时间大于 CPU 的读出时间时, 8086 CPU 根据控制线 READY 的状态, 应在周期\_\_ (1) \_\_ 状态间插入\_\_ (2) \_\_ 周期。若 8086 CPU 不执行总线操作时, 则应在周期\_\_ (3) \_\_ 后插入\_\_ (4) \_\_ 周期。

【解】 (1) T3 与 T4 (2) 等待 ( $T_w$ ) (3) T4 (4) 空闲  $T_i$

38. 80386 以上的微处理器所支持的 3 种工作模式是: \_\_ (1) \_\_、\_\_ (2) \_\_ 和\_\_ (3) \_\_。

【解】 (1) 实模式 (2) 保护模式 (3) 虚拟 86 模式

## 2.5 简答题

1. 简述 CPU 执行程序的过程。

【解】 当程序的第一条指令所在的地址送入程序计数器后, CPU 就进入取指阶段准备取第一条指令。在取指阶段, CPU 从内存中读出指令, 并把指令送至指令寄存器 IR 暂存。在取指阶段结束后, 机器就进入执行阶段, 这时由指令译码器对指令译码, 再经控制器发出相应的控制信号, 控制各部件执行指令所规定的具体操作。当一条指令执行完毕以后, 就转入了下一条指令的取指阶段。以上步骤周而复始地循环, 直到遇到停机指令。

2. 在计算机中, CPU 地址线的位数与访问存储器单元范围的关系是什么?

【解】 在计算机中, 若 CPU 的地址线位数为  $N$  (即有  $N$  条地址线), 则访问存储器单元的数量为  $2^N$  个, 访问存储器单元的范围为  $0 \sim 2^N - 1$ 。

3. 试对 8086 CPU 和 8088 CPU 的主要特点进行比较。

【解】 8086 CPU 和 8088 CPU 的主要特点见表 2-2。

表 2-2 8086 CPU 和 8088 CPU 的主要特点

	8088	8086
相同点	内部数据总线为 16 位、寄存器和指令系统完全兼容	
不同点	1 外部数据总线为 8 位	外部数据总线为 16 位
	2 指令队列缓冲器为 4 字节	指令队列缓冲器为 6 字节
	3 无 $\overline{BHE}$ 控制线 存储器和 I/O 选择控制线为 $IO/\overline{M}$	有高 8 位数据线传输控制线 $\overline{BHE}$ 存储器和 I/O 选择控制线为 $M/\overline{IO}$



4. 8086/8088 CPU 由哪两个功模块构成？简述它们之间的关系。

**【解】** 8086/8088 CPU 为实现指令的流水线操作，将内部划分为执行单元 EU 和总线接口单元 BIU 两个模块。EU 和 BIU 的工作既相互独立，又相互配合。其中，EU 负责执行指令，BIU 负责通过外部总线读/写 CPU 的外部数据。BIU 经总线从存储器中读取指令后存入指令队列缓冲器，以便 EU 从指令队列中获取指令。当 EU 需要从外部获取数据时，便通知 BIU，BIU 经总线操作获得数据后，通过内部总线提供给 EU。

5. 简述 8086/8088 CPU 中指令队列的功能和工作原理。

**【解】** 8086/8088 CPU 中指令队列的功能是完成指令的流水线操作，其操作原则为先进先出。BIU 单元经总线从程序存储器中读取指令后存入指令队列缓冲器，EU 单元从指令队列缓冲器中获取先存入的指令并执行。在 EU 执行指令的同时 BIU 又可以继续取指令，由此实现取指令和执行指令同时操作，提高了 CPU 的效率。

6. 8086 系统中存储器的逻辑地址由哪两部分组成？物理地址由何器件生成？如何生成？每个段的逻辑地址与寄存器之间有何对应关系？

**【解】** 8086 系统中存储器的逻辑地址由段地址（段首址）和段内偏移地址（有效地址）两部分组成；存储单元的物理地址由地址加法器生成，寻址时，CPU 首先将段地址和段内偏移地址送入地址加法器，地址加法器将段地址左移 4 位并与段内偏移地址相加，得到一个 20 位的物理地址。数据段的段地址在 DS 寄存器中，段内偏移地址可能在 BX、BP、SI 或 DI 寄存器中。代码段的段地址在 CS 寄存器中，段内偏移地址在 IP 寄存器中。堆栈段的段地址在 SS 寄存器中，段内偏移地址在 SP 寄存器中。扩展段的段地址在 ES 寄存器中，段内偏移地址可能在 BX、BP、SI 或 DI 寄存器中。

7. 简述何谓物理地址？何谓逻辑地址？

**【解】** 物理地址：完成对存储器单元或 I/O 端口寻址的实际地址称为物理地址，其具有唯一性，且根据 CPU 型号不同，地址线位数不同，寻址范围不同。例如，8080 CPU 的物理地址为 16 位，寻址范围是 64KB。8086 CPU 的物理地址为 20 位，寻址范围是 1MB。80286 CPU 的物理地址为 24 位，寻址范围是 16MB。

逻辑地址为在程序中对存储器的寻址。例如，8086 CPU 中存储单元的逻辑地址由两个 16 位分量描述，即 16 位段基值和 16 位偏移量。逻辑地址不唯一。

8. 已知两个 16 位的字数据 268AH 和 357EH，它们在 8086 存储器中的地址分别为 00120H 和 00124H，试画出它们的存储示意图。

**【解】** 数据的存储示意图见图 2-2。

9. 找出字符串“Pentium”的 ASCII 码，将它们依次存入从 00510H 开始的字节单元中，画出它们存放的内存单元示意图。

**【解】** 存储示意图见图 2-3。

10. 8086/8088 CPU 具有哪些寄存器？可存放段基值的寄存器有哪些？可存放存储器单元的偏移地址分量的寄存器有哪些？存放状态和控制信息的寄存器有哪些？

**【解】** 8086/8088 CPU 具有 14 个 16 位的寄存器，包括 AX、BX、CX、DX、SI、DI、SP、BP、CS、DS、ES、SS、FR 和 IP。

可存放段基值的寄存器有 CS、DS、ES、SS。

可存放存储器单元的偏移地址分量的寄存器有 SI、DI、BX、BP。

存放状态和控制信息的寄存器为 FR。

8AH	00120H
26H	00121H
	00122H
	00123H
7EH	00124H
35H	00125H

图 2-2 数据的存储示意图

50H	00510H
65H	00511H
6EH	00512H
74H	00513H
69H	00514H
75H	00515H
6DH	00516H

图 2-3 字符的存储示意图

11. 8086/8088 CPU 的标志寄存器 FR 具有几个有效位？几个状态标志位？几个控制标志位？

【解】

8086/8088 CPU 的标志寄存器 FR 具有 9 个有效位，包括 6 个状态标志位（CF、OF、ZF、SF、AF、PF）和 3 个控制标志位（IF、DF、TF）。

12. 8086/8088 CPU 的标志寄存器 FR 中状态标志位的作用是什么？控制标志位的作用是什么？

【解】 1) 8086/8088 CPU 的标志寄存器 FR 具有 6 个状态标志位，分别为 CF、AF、ZF、SF、OF 和 PF，其作用如下：

CF 为进位或借位标志，当算术运算无进位或借位时，CF = 0；当算术运算有进位或借位时，CF = 1。

AF 为辅助进位标志（或称半进位标志），若 D<sub>3</sub> 位向 D<sub>4</sub> 位无进位（或无借位）时，则 AF = 0；若 D<sub>3</sub> 位向 D<sub>4</sub> 位有进位（或有借位）时，AF = 1。

ZF 为零标志位，若运算结果为 0，则 ZF = 1，否则 ZF = 0。

SF 为符号位。对于有符号数运算，运算结果为正时 SF = 0，运算结果为负时 SF = 1。实际上，SF 的状态反映运算结果的最高位的状态。

OF 为溢出标志，对于有符号数运算，运算结果无溢出时 OF = 0，有溢出时 OF = 1。

PF 为奇偶校验标志，若运算结果低 8 位中 1 的个数为奇数，则 PF = 0；若运算结果低 8 位中 1 的个数为偶数时，则 PF = 1。

2) 8086/8088 CPU 的标志寄存器 FR 具有 3 个控制标志位是 IF、DF 和 TF，其作用如下：

IF 为允许 CPU 响应可屏蔽中断的控制位，当 IF = 1 时，允许 CPU 响应可屏蔽中断请求，当 IF = 0 时，禁止 CPU 响应可屏蔽中断请求。

DF 为增量方向控制位，在数据串操作时，当 DF = 0 时，地址将会自动加 1 或 2；当 DF = 1 时，地址将会自动减 1 或 2。

TF 为指令单步调试陷阱控制位，当 TF = 0 时，无指令单步调试操作；当 TF = 1 时，有指令单步调试操作。

13. 简述 8086/8088 CPU 的最小和最大工作模式的主要区别。

【解】 8086/8088 CPU 的最小和最大工作模式的主要区别见表 2-3。

表 2-3 8086/8088 CPU 最小和最大工作模式的主要区别

		最小工作模式	最大工作模式
主要区别	1	构成单处理器的简单系统	构成多处理器的复杂系统
	2	全部信号由 8086/8088 CPU 提供	部分信号由 8288 总线控制器提供
	3	MN/ $\overline{\text{MX}}$ 接高电平	MN/ $\overline{\text{MX}}$ 接低电平

14. 8088CPU 工作在最小模式下时:

- (1) 当 CPU 访问存储器时, 要利用哪些信号?
- (2) 当 CPU 进行 I/O 操作时, 要利用哪些信号?
- (3) 当 HOLD 有效并得到响应时, CPU 的哪些信号置高阻态?

**【解】**

- (1) 要利用的信号线包括  $\overline{\text{WR}}$ 、 $\overline{\text{RD}}$ 、 $\overline{\text{IO/M}}$ 、ALE、AD0 ~ AD7、A8 ~ A19。
- (2) 同 (1)。
- (3) 所有三态输出的地址信号、数据信号和控制信号均置为高阻态。

15. 在总线周期中, 什么情况下要插入  $T_w$  等待周期? 插入  $T_w$  周期的个数取决于什么因素?

**【解】** 在每个总线周期的 T3 的开始处, 若 READY 为低电平, 则 CPU 在 T3 后插入一个等待周期  $T_w$ 。在  $T_w$  的开始时刻, CPU 还要检查 READY 状态, 若仍为低电平, 则再插入一个  $T_w$ 。此过程一直进行到某个  $T_w$  开始时, READY 已经变为高电平, 这时下一个时钟周期才转入 T4。可以看出, 插入  $T_w$  周期的个数取决于 READY 电平维持的时间。

16. 什么叫总线周期? 在 CPU 读/写总线周期中, 数据在哪个机器状态时出现在数据总线上?

**【解】** CPU 完成一次存储器访问或 I/O 端口操作所需要的时间称为一个总线周期, 由几个 T 状态组成。在读/写总线周期中, 数据在 T2 ~ T4 状态出现在数据总线上。

17. 什么是堆栈? 简述堆栈在微型计算机中的作用。

**【解】** 堆栈是计算机中按照先进后出原则组织的一块特殊的存储区域。堆栈的作用: 存放临时数据和地址, 保护断点和现场等。

18. 简述 8086/8088 CPU 的堆栈操作原理。

**【解】** 8086/8088 CPU 的堆栈是一段特殊定义的存储区, 用于存放 CPU 堆栈操作时的数据。在执行堆栈操作前, 需先定义堆栈段 SS、堆栈深度 (栈底) 和堆栈栈顶指针 SP。数据的入栈、出栈操作类型均为 16 位二进制数, 入栈操作时, 栈顶指针值先自动减 2 (即  $SP = SP - 2$ ), 然后 16 位数据从栈顶处入栈; 出栈操作时, 16 位数据先从栈顶处出栈, 然后栈顶指针值自动加 2 (即  $SP = SP + 2$ )。

19. 在 8086/8088 系统中, 存储器为什么要分段? 一个段最大为多少字节? 最小为多少字节?

**【解】** 分段的主要目的是便于存储器的管理, 使得可以用 16 位寄存器来寻址 20 位的内存空间。一个段最大为 64KB, 最小为 16B。

20. 设当前数据段位于存储器的 A8000H ~ B7FFFH, DS 段寄存器的内容应是什么?

**【解】** 因为 A8000H ~ B7FFFH 之间的地址范围大小为 64KB, 未超出一个段的最大范围。故要访问此地址范围的数据, 数据段的起始地址 (即段首地址) 应为 A8000H, 则 DS 段寄存器为 A800H。

21. 设  $AX = 2875H$ 、 $BX = 34DFH$ 、 $SS = 1307H$ 、 $SP = 8H$ , 依此执行 PUSH AX、PUSH BX、POP AX、POP CX 后栈顶指针变为多少?  $AX = ?$   $BX = ?$   $CX = ?$

**【解】** 当前栈顶指针  $= SS * 10H + SP = 13070H + 8H = 13078H$ , 依此执行 PUSH AX, PUSH BX, POP AX, POP CX 后栈顶指针仍为 13078H。  $AX = 34DFH$ ,  $BX = 34DFH$ ,  $CX = 2875H$ 。

22. 8086/8088 CPU 的 1MB 存储空间可分为多少个逻辑段? 每段的寻址范围是多少?

【解】 8086/8088 CPU 的 1MB 存储空间可分为任意个逻辑段，段与段之间可以连续也可以不连续，可以重叠也可以相交。但每个逻辑段的寻址范围不能大于 64KB。

23. 在 8086 系统中，存储器采用什么结构？用什么信号来选中存储体？

【解】 在 8086 系统中，存储器采用分体结构，1MB 的存储空间分成两个存储体：偶地址存储体和奇地址存储体，各为 512KB。使用  $A_0$  和  $\overline{BHE}$  来区分两个存储体。当  $A_0 = 0$  时，选中偶地址存储体，与数据总线低 8 位相连，从低 8 位数据总线读/写 1 字节。当  $\overline{BHE} = 0$  时，选中奇地址存储体，与数据总线高 8 位相连，从高 8 位数据总线读/写 1 字节。当  $A_0 = 0$ ， $\overline{BHE} = 0$  时，同时选中两个存储体，读/写一个字。

24. 8086 CPU 控制线  $\overline{BHE}$  和地址线  $A_0$  对存储器访问的控制作用是什么？

【解】 8086 CPU 对存储器进行组织时，每一存储单元地址中仅能存放 8 位二进制数据，所以 8086 在进行 16 位数据操作时需同时访问两个 8 位的存储单元。8086 为了既能传输 8 位数据也能传输 16 位数据，将存储空间分成两部分：奇存储体连接 16 位数据总线的高 8 位，即  $D_{15} \sim D_8$ ，由控制线  $\overline{BHE}$  参与选通控制；偶存储体连接 16 位数据总线的低 8 位，即  $D_7 \sim D_0$ ，由  $A_0$  参与选通控制。用  $\overline{BHE}$  和  $A_0$  不同的状态组态，实现对存储区中任意 8 位或 16 位数据的访问，见表 2-4。

表 2-4  $\overline{BHE}$  与  $A_0$  信号的作用

$\overline{BHE}$	$A_0$	操 作	所用的数据线
0	0	读/写一个偶地址字	$AD_{15} \sim AD_0$
0	1	读/写一个奇地址字节	$AD_{15} \sim AD_8$
1	0	读/写一个偶地址字节	$AD_7 \sim AD_0$
1	1	无	无

25. 在 8086/8088 CPU 中，控制线 ALE 的作用是什么？

【解】 8086/8088 CPU 在地址/数据复用线上提供地址信息时，地址有效控制线 ALE 为高电平，除此之外，地址有效控制线 ALE 为低电平无效状态。在用 8086/8088 CPU 构成系统时，常用地地址有效控制线 ALE 控制地址锁存器来获取稳定的地址信息。

26. 在 8086 CPU 中，控制线  $\overline{RD}$ 、 $\overline{WR}$  和  $M/\overline{IO}$  的作用是什么？

【解】 控制线  $\overline{RD}$ 、 $\overline{WR}$  的作用是完成存储器单元或 I/O 端口的数据读/写控制。当  $\overline{RD} = 0$ 、 $\overline{WR} = 1$  时，CPU 经数据总线从选中的存储器单元或 I/O 端口中读取数据，当  $\overline{RD} = 1$ 、 $\overline{WR} = 0$  时，CPU 经数据总线向选中的存储器单元或 I/O 端口中写入数据。

控制线  $M/\overline{IO}$  的作用是确定在某一时刻 CPU 对存储器操作，还是对 I/O 端口操作。当  $M/\overline{IO} = 0$  时，CPU 对 I/O 端口操作有效；当  $M/\overline{IO} = 1$  时，CPU 对存储器操作有效。

8086 CPU 控制线  $\overline{RD}$ 、 $\overline{WR}$  和  $M/\overline{IO}$  的不同组态，可以产生对存储器读信号  $\overline{MEMR}$ 、对存储器写信号  $\overline{MEMW}$ 、对 I/O 端口读信号  $\overline{IOR}$  和对 I/O 端口写信号  $\overline{IOW}$ ，如图 2-4 所示。

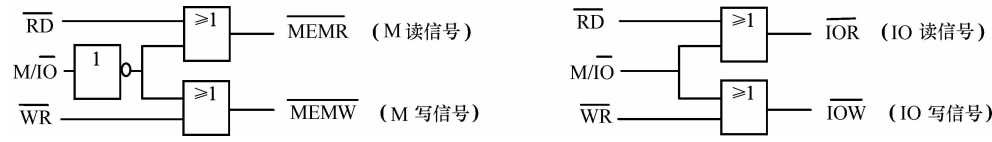


图 2-4 8086 对存储器或 IO 操作信号

27. 在 8086/8088 CPU 中, 控制线  $\overline{\text{DEN}}$ 、 $\overline{\text{DT/R}}$  的作用是什么?

【解】 8086/8088 CPU 通过控制线  $\overline{\text{DEN}}$  和  $\overline{\text{DT/R}}$  提供其数据传输及数据流方向信息。在用 8086/8088 CPU 构成系统时, 可用控制线  $\overline{\text{DEN}}$  和  $\overline{\text{DT/R}}$  完成对双向数据缓冲器芯片的控制。当控制线  $\overline{\text{DEN}}=0$  时, 数据缓冲器片选有效。控制线  $\overline{\text{DT/R}}$  的作用是数据缓冲器中数据传送方向控制, 当  $\overline{\text{DT/R}}=0$  时, 数据从数据总线上流入 CPU。当  $\overline{\text{DT/R}}=1$  时, CPU 经数据总线流出数据。

28. 什么是统一编址? 什么是独立编址? 各有何特点?

【解】 在计算机中, I/O 端口的编制方式分为统一编址和独立编址。

统一编址的存储器单元地址和 I/O 端口地址在同一个地址空间。由于 I/O 端口地址占用存储器单元地址, 减少了存储器的寻址空间, 因此访问存储器单元和 I/O 端口可用相同的指令。

独立编址的存储器单元地址和 I/O 端口地址具有不同的地址空间。存储器和 I/O 端口都具有独立的且较大的寻址空间, CPU 用不同的控制线来区别, 采用不同的指令访问存储器单元或 I/O 端口。

29. 直接端口寻址、间接端口寻址的特点是什么?

【解】 8086/8088 CPU 在对 I/O 端口访问时有以下两种寻址方式: 直接寻址和寄存器间接寻址。当 I/O 端口地址为 8 位地址 ( $A7 \sim A0$ ) 表示时, 可以采用直接寻址, 即在 IN/OUT 指令中, 直接给出 8 位端口地址。当 I/O 端口地址为 16 位地址 ( $A15 \sim A0$ ) 表示时, 必须采用寄存器间接寻址方式, 即 16 位端口地址应先赋给寄存器 DX, 在 IN/OUT 指令中仅出现寄存器 DX。

30. 什么是规则字? 什么是非规则字?

【解】 在 8086 CPU 的存储系统中, 规定每个存储单元仅存放 8 位二进制信息。而 8086 CPU 的数据总线宽度为 16 位, 即可以实现同时将两个存储单元中的数据经数据总线传输。当 16 位字数据按规则字存放在存储器中时 (即偶地址存放 16 位字数据的低 8 位、奇地址存放 16 位字数据的高 8 位), 则用一个总线周期可以完成 16 位数据的传送。若 16 位字数据按非规则字存放在存储器中时 (即偶地址存放 16 位字数据的高 8 位、奇地址存放 16 位字数据的低 8 位), 则需两个总线周期才可以完成 16 位数据的传送。

31. 写出当 8088 CPU 执行下列指令时, CPU 控制总线上的  $\overline{\text{IO/M}}$ 、 $\overline{\text{RD}}$ 、 $\overline{\text{WR}}$  信号线的状态:

MOV AL, BH

MOV [BX], CL

【解】 MOV AL, BH; 由于该指令是在 CPU 内工作, 因此无任何信号线有效。

MOV [BX], CL; 执行该指令时存储器写信后有效, 即  $\overline{\text{IO/M}}=0$ ,  $\overline{\text{WR}}=0$ 。

32. 8086CPU 的最小模式系统配置包括哪几部分?

【解】 8086 最小模式系统配置包括 8086CPU、存储器、I/O 接口芯片、1 片 8284 时钟发生器、3 片 8282 地址锁存器、2 片 8286 双向数据总线收发器。

## 第 3 章 80 × 86 指令系统

### 3.1 学习指导

本章主要内容和要求如下：

#### 1. 指令中包含的信息

指令中一般包含的信息有操作码和操作数。其中操作数可以处在计算机的指令中、CPU 内部的寄存器中、存储器中和 I/O 端口中。

#### 2. 寻址方式

寻址方式即寻找操作数或操作数所在位置的方式。操作数在计算机中所处的位置不同，寻址方式也不同。8086/8088 CPU 的寻址方式如下：

1) 隐含寻址方式，即指令中隐含规定了操作数所在位置。

2) 数据型操作数寻址方式。

● 立即数寻址：操作数在指令中。

● 寄存器寻址：操作数在寄存器中。

● 直接寻址

● 寄存器间接寻址

● 基址寻址和变址寻址

● 基址变址寻址

操作数在存储器中，其逻辑地址 = 段寄存器：[EA]，其中：

$$EA = \begin{cases} BX \\ BP \end{cases} + \begin{cases} SI \\ DI \end{cases} + \begin{cases} 0 \\ n_8 \\ n_{16} \end{cases}$$

● 串操作寻址：操作数在存储器中，源操作数为 DS：[SI]，目标操作数为 ES：[DI]。

● I/O 端口寻址：操作数在 I/O 端口中，分为 8 位地址直接寻址和 16 位地址 DX 间接寻址。

80386 和更高档的微处理器还包含比例变址方式的数据寻址。

3) 目标地址寻址方式，包括相对程序寻址、直接程序寻址、间接程序寻址。

#### 3. 8086/8088 指令系统

要求了解指令的格式、特点、用法、对状态控制寄存器 FR 的影响。常用指令要熟练掌握。注意隐含规定的各种寻址规则。8086/8088 指令系统包括的主要指令如下。

1) 数据传送指令：

MOV	dest, src	LEA	dest, src	XCHG	dest, src	XLAT
PUSH	src	POP	dest	PUSHF		POPF
LDS	dest, src	LES	dest, src	LAHF		SAHF
IN	dest, src	OUT	dest, src			

2) 算术运算指令：

ADD	dest, src ;	ADC	dest, src ;	AAA	;	DAA
SUB	dest, src ;	SBB	dest, src ;	AAS	;	DAS

MUL	src	;	IMUL	src	;	AAM		;	CBW
DIV	src	;	IDIV	src	;	AAD		;	CWD
CMP	dest, src	;	INC	dest	;	DEC	dest	;	NEG dest

3) 逻辑运算指令:

AND	dest, src	;	OR	dest, src	;	NOT	dest	;	XOR dest, src
TEST	dest, src								

4) 位移指令:

SHL	dest, 1	;	SHR	dest, 1	;	SAL	dest, 1	;	SAR dest, 1
ROL	dest, 1	;	ROR	dest, 1	;	RCL	dest, 1	;	RCR dest, 1
SHL	dest, CL	;	SHR	dest, CL	;	SAL	dest, CL	;	SAR dest, CL
ROL	dest, CL	;	ROR	dest, CL	;	RCL	dest, CL	;	RCR dest, CL

5) 串操作指令:

MOVS	dest, src	;	CMPS	dest, src	;	SCAS	dest	;	LODS src
STOS	dest								

重复前缀——REP、REPE、REPNE

6) 控制转移指令:

JMP	dest	;	Jxx	dest	;	CALL	dest		RET (n)
INT	n		IRET						

7) 处理器控制指令:

CLC		;	STC		;	CLD		;	STD
CLI		;	STI		;	CMC		;	NOP
HLT		;							

## 3.2 单项选择题

1. 8086/8088 CPU 内部具有 ( ) 个寄存器可以装载内存操作数的偏移地址信息。  
A. 4                      B. 8                      C. 14                      D. 20

【解】 A

2. 8086/8088 CPU 内部 ( ) 寄存器可以装载内存操作数的偏移地址信息。  
A. AX, BX, CX, DX                      B. SI, DI, SP, BP  
C. BX, BP, SI, DI                      D. AX, BX, CX, DX, SP, BP, SI, DI

【解】 C

3. 确定一个内存单元有效地址 EA 是由几个地址分量组合而成的, 这些分量不包括 ( )。  
A. 位移量                      B. 基地址                      C. 逻辑地址                      D. 变址地址

【解】 C

4. 在寄存器间接寻址方式中, 操作数的有效地址 EA 可以通过寄存器 ( ) 间接得到。  
A. AX                      B. BP                      C. CX                      D. SP

【解】 B

5. 常用来获取内存单元偏移量的指令是 ( )。

- A. LAHF                      B. LEA                      C. LES                      D. LDS

【解】 B

6. 在寄存器间接寻址方式下, 在 EA 中使用寄存器 ( ) 时默认段寄存器为 SS。

- A. BX                      B. BP                      C. SI                      D. DI

【解】 B

7. 当基地址变址寻址时, 可以与基址寄存器 BP 作变址寄存器的是 ( )。

- A. BX                      B. SS                      C. SI                      D. DS

【解】 C

8. 在程序运行过程中, 下一条指令的物理地址的计算表达式是 ( )。

- A.  $CS \times 10H + IP$     B.  $DS \times 10H + BX$     C.  $SS \times 10H + SP$     D.  $SS \times 10H + BP$

【解】 A

9. MOV AX, [BP] [SI] 的源操作数的物理地址是 ( )。

- A.  $10H \times DS + BP + SI$                       B.  $10H \times ES + BP + SI$   
C.  $10H \times SS + BP + SI$                       D.  $10H \times CS + BP + SI$

【解】 C

10. 指令 MOV CX, 1245H 中的源操作数存放在 ( )。

- A. DS: 1245H 所指定的内存中                      B. 该指令中  
C. 某个寄存器中                      D. 都不是

【解】 B

11. 8088/8086 的字乘法指令的乘积在 ( ) 寄存器对中。

- A. BX: AX                      B. AX: BX                      C. AX: DX                      D. DX: AX

【解】 D

12. 在 8086/8088 乘法指令中的两个操作数, 其中有一个操作数一定存放在 ( ) 中。

- A. AL 或 AX                      B. BL 或 BX                      C. CL 或 CX                      D. DL 或 DX

【解】 A

13. 设 AL 的内容为 4FH, 执行指令 “TEST AL, 05H” 后, AL 的内容为 ( )。

- A. 01H                      B. 05H                      C. 45H                      D. 4FH

【解】 D

14. 对于算术左移指令 SAL AL, 1, 若 AL 中的带符号数在指令执行后符号有变, 则可以通过 ( ) 来确认。

- A.  $OF = 1$                       B.  $OF = 0$                       C.  $CF = 1$                       D.  $CF = 0$

【解】 A

15. 8086/8088 的移位类指令若需移动多位时, 则应该先将移动位数置于 ( )。

- A. AL                      B. AH                      C. CL                      D. CH

【解】 C

16. 如果要想实现正确返回, 则 CALL 指令和 ( ) 指令两者必须成对出现, 且属性相同。

- A. MACRO                      B. JMP                      C. RET                      D. END

【解】 C

17. 条件转移指令 JNZ 的转移条件是 ( )。



- A. CF = 1                      B. ZF = 0                      C. OF = 0                      D. ZF = 1

【解】 B

18. JMP WORD PTR [DI] 是 ( )。

- A. 段内间接转移    B. 段间间接转移    C. 段内直接转移    D. 段间直接转移

【解】 A

19. 指令 LOOPNE/LOOPNZ 循环的条件是 ( )。

- A. ZF = 1 且 CX = 0                      B. ZF = 0 且 CX ≠ 0  
C. ZF = 0 且 CX = 0                      D. ZF = 1 且 CX ≠ 0

【解】 B

20. 指令 REPNE SCASB 执行以后, 如果 ZF = 1, 则表示 ( )。

- A. 在此字符串中, 没有找到指定字符    B. 已经找到要查找的字符  
C. 两个字符串相等                      D. 此字符串是由同一字符组成的

【解】 B

21. 不能实现  $AX = BX - CX$  功能的指令是 ( )。

- A. SUB BX, CX                      B. SUB AX, BX  
   MOV AX, BX                      SUB AX, CX  
C. XCHG AX, BX                      D. MOV AX, BX  
   SUB AX, CX                      SUB AX, CX

【解】 B

22. 在 8086/8088 指令中, 下述寻址方式不正确的是 ( )。

- A. [BX] [SI]    B. [BP + DI + 25]    C. [BX + BP]    D. [BX + DI]

【解】 C

23. AND、OR、XOR、NOT 为 4 条逻辑运算指令, 下面 ( ) 解释有误。

- A. 它们都是按位操作的  
B. 指令 XOR AX, AX 执行后, 结果不变, 但是影响标志位  
C. 指令 AND AL, 0FH 执行后, 使 AL 的高 4 位清零, 低 4 位不变  
D. 若 DL = 09H, CH = 30H, 则执行 OR DL, CH 后, 结果为 DL = 39H

【解】 B

24. 下列语句中有语法错误的是 ( )。

- A. MOV AX, [BX] [BP]                      B. ADD AX, [BP]  
C. CMP [BX + DI], 0FH                      D. LEA SI, SS: 20H [BX]

【解】 A

25. 下列指令中有语法错误的是 ( )。

- A. MOV [SI], [DI]                      B. IN AL, DX  
C. JMP WORD PTR [BX + 8]                      D. PUSH [BX + DI - 10H]

【解】 A

26. 下列指令出现语法错误的指令有 ( )。

- A. MOV [BX + SI], AL                      B. MOV AX, [BP + DI]  
C. MOV DS, AX                      D. MOV CS, AX

【解】 D

27. 下面的数据交换指令中，错误的操作是（ ）。

- A. XCHG AX, DI
- B. XCHG BX, [BP + DAT]
- C. XCHG DS, SS
- D. XCHG BUF, DX

【解】 C

28. 在 8086/8088 微机系统中，将 AL 内容送到 I/O 接口中，使用的指令是（ ）。

- A. IN AL, 端口地址
- B. MOV AL, 端口地址
- C. OUT AL, 端口地址
- D. OUT 端口地址, AL

【解】 D

29. 下列语句中，语法有错误的是（ ）。

- A. IN AL, DX
- B. OUT AX, DX
- C. IN AX, DX
- D. OUT DX, AL

【解】 B

30. 两个非压缩型 BCD 码数据相减后，执行减法调整指令 AAS 时，将自动测试是否满足（ ），从而决定是否需要校正。

- A. AL 中的数值 >9, 且 AF = 1
- B. AL 中低 4 位数 >9, 且 AF = 1
- C. AL 中的数值 >9, 或 AF = 1
- D. AL 中低 4 位数 >9, 或 AF = 1

【解】 D

31. 用 REPNE CMPSB 指令实现两个字符串比较，如果在指令完成后 CX = 0，其原因是（ ）。

- A. 出现两个相同位置字符相等的情况
- B. 出现两个相同位置字符不等的情况
- C. 两个字符串长度不等
- D. 字符串大小不同

【解】 B

32. 在执行 STD 和 MOVSW 指令后，SI 和 DI 的变化是（ ）。

- A. 加 1
- B. 减 1
- C. 加 2
- D. 减 2

【解】 D

33. 将 DX 的内容除以 2，正确的指令是（ ）。

- A. DIV 2
- B. DIV DX, 2
- C. SAR DX, 1
- D. SHL DX, 1

【解】 C

34. AL 的内容实现算术右移 4 位的正确指令是（ ）。

- A. SHR AL, 4
- B. MOV CL, 4
- C. SAR AL, 4
- D. MOV CL, 4
- SHR AL, CL
- SAR AL, CL

【解】 D

35. 指令 RET 8 是 NEAR 过程的返回语句，执行之后，SP 的值增加（ ）。

- A. 6
- B. 8
- C. 10
- D. 12

【解】 C

36. 执行 PUSH AX 指令时将自动完成（ ）。

- A. ①  $SP \leftarrow SP - 1$ ,  $SS: [SP] \leftarrow AL$       B. ①  $SP \leftarrow SP - 1$ ,  $SS: [SP] \leftarrow AH$   
 ②  $SP \leftarrow SP - 1$ ,  $SS: [SP] \leftarrow AH$       ②  $SP \leftarrow SP - 1$ ,  $SS: [SP] \leftarrow AL$   
 C. ①  $SP \leftarrow SP + 1$ ,  $SS: [SP] \leftarrow AL$       D. ①  $SP \leftarrow SP + 1$ ,  $SS: [SP] \leftarrow AH$   
 ②  $SP \leftarrow SP + 1$ ,  $SS: [SP] \leftarrow AH$       ②  $SP \leftarrow SP + 1$ ,  $SS: [SP] \leftarrow AL$

【解】 B

37. 执行 POP AX 指令时将自动完成 ( )。

- A. ①  $AH \leftarrow SS: [SP]$ ,  $SP \leftarrow SP + 1$       B. ①  $SP \leftarrow SP + 1$ ,  $AH \leftarrow SS: [SP]$   
 ②  $AL \leftarrow SS: [SP]$ ,  $SP \leftarrow SP + 1$       ②  $SP \leftarrow SP + 1$ ,  $AL \leftarrow SS: [SP]$   
 C. ①  $AL \leftarrow SS: [SP]$ ,  $SP \leftarrow SP + 1$       D. ①  $SP \leftarrow SP + 1$ ,  $AL \leftarrow SS: [SP]$   
 ②  $AH \leftarrow SS: [SP]$ ,  $SP \leftarrow SP + 1$       ②  $SP \leftarrow SP + 1$ ,  $AH \leftarrow SS: [SP]$

【解】 C

38. 执行以下指令后, SP 寄存器的值应是 ( )。

```
MOV SP, 100H
PUSH AX
```

- A. 00FFH      B. 00FEH      C. 0101H      D. 0102H

【解】 B

39. 假定  $SS = 1000H$ ,  $SP = 0100H$ ,  $AX = 2107H$ , 执行指令 PUSH AX 后, 存放数据 07H 的内存物理地址是 ( )。

- A. 10102H      B. 10101H      C. 100FEH      D. 100FFH

【解】 C

40. 若  $AL = -79$ ,  $BL = -102$ , 当执行 ADD AL, BL 后, 进位 CF 和溢出位 OF 的状态为 ( )。

- A.  $CF = 0$ ,  $OF = 1$       B.  $CF = 1$ ,  $OF = 1$       C.  $CF = 0$ ,  $OF = 0$       D.  $CF = 1$ ,  $OF = 0$

【解】 B

41. INC 和 DEC 指令不影响标志位 ( ) 的状态。

- A. OF      B. CF      C. SF      D. ZF

【解】 B

42. 完成下列程序段操作后, 各状态位的状态为 ( )。

```
MOV AL, 1AH
MOV BL, 97H
ADD AL, BL
```

- A.  $ZF = 0$ ,  $SF = 1$ ,  $CF = 0$ ,  $AF = 0$ ,  $PF = 1$ ,  $OF = 0$   
 B.  $ZF = 0$ ,  $SF = 1$ ,  $CF = 0$ ,  $AF = 1$ ,  $PF = 1$ ,  $OF = 0$   
 C.  $ZF = 0$ ,  $SF = 0$ ,  $CF = 1$ ,  $AF = 0$ ,  $PF = 1$ ,  $OF = 1$   
 D.  $ZF = 0$ ,  $SF = 0$ ,  $CF = 1$ ,  $AF = 1$ ,  $PF = 0$ ,  $OF = 1$

【解】 B

43. 完成将累加器 AX 清零, 但不影响进位标志位 CF 状态的指令是 ( )。

- A. SUB AX, AX      B. XOR AX, AX  
 C. MOV AX, 00H      D. AND AX, 00H

【解】 C

44. 完成将累加器 AL 清零, 并将进位标志 CF 清零, 下面错误的指令是 ( )。

- A. MOV AL, 00H
- B. AND AL, 00H
- C. XOR AL, AL
- D. SUB AL, AL

【解】 A

45. 下列指令分别执行后, 将总是使 CF = 0 和 OF = 0 的指令为 ( )。

- A. MOV
- B. OR
- C. NEG
- D. INC

【解】 B

46. 对状态标志位 CF 产生影响的指令是 ( )。

- A. INC AX
- B. NOT AX
- C. NEG AX
- D. DEC AX

【解】 C

47. 下列指令助记符中影响标志寄存器中进位位 CF 的指令有 ( )。

- A. MOV
- B. ADD
- C. DEC
- D. INC

【解】 B

48. 使状态标志位 CF 置零的不正确指令是 ( )。

- A. SUB AX, AX
- B. CLC
- C. NEG AX
- D. XOR AX, AX

【解】 C

49. 执行中断服务程序返回指令 RETI 时, 返回地址来自于 ( )。

- A. ROM 区
- B. 程序计数器
- C. 堆栈区
- D. 中断向量表

【解】 C

50. 将 BUF 字节单元内容算术左移一位, 以下指令不正确的是 ( )。

- A. MOV BX, OFFSET BUF
- B. MOV BL, BUF
- SAL BX, 1
- SAL BL, 1
- C. SAL BUF, 1
- D. LEA BX, BUF
- SAL [BX], 1

【解】 A

51. 完成下列操作以后, 传送到寄存器 AL, BL, CL, DL 中的数, 正确的是 ( )。

MOV AL, 41H

MOV BL, 92

MOV CL, 'B'

MOV DL, 01111111B

- A. AL = 41H
- B. AL = 41H
- C. AL = 65
- D. AL = 01000001B
- BL = 5CH
- BL = 92
- BL = 134
- BL = 1011100B
- CL = 42H
- CL = B
- CL = 66
- CL = 00001011B
- DL = 7FH
- DL = 3FH
- DL = 127
- DL = 01111111B

【解】 A

52. 将寄存器 BX 的内容求反, 不正确的操作是 ( )。

- A. NOT BX
- B. XOR BX, 0FFFFH
- C. AND BX, 0FFFFH

【解】 C

53. 下面指令组完成将字单元 BUF1 和 BUF2 的内容互换, 错误的操作为 ( )。

- |                 |                    |
|-----------------|--------------------|
| A. MOV AX, BUF1 | B. MOV AX, BUF1    |
| MOV BX, BUF2    | MOV BX, BUF2       |
| XCHG AX, BX     | MOV BUF2, AX       |
| MOV BUF1, AX    | MOV BUF1, BX       |
| MOV BUF2, BX    |                    |
| C. MOV AX, BUF1 | D. XCHG BUF1, BUF2 |
| XCHG AX, BUF2   |                    |
| MOV BUF, AX     |                    |

【解】 D

54. 当前 BX = 0003H, AL = 03H, DS = 2000H, (20003H) = 0ABH, (20004H) = 0CDH, (20005H) = 0ACH, (20006H) = 0BDH, 则执行了 XLAT 指令后, AL 中的内容是 ( )。

- A. 0ABH                      B. 0ACH                      C. 0CDH                      D. 0BDH

【解】 D

55. 将字变量 BUF 的偏移地址存入寄存器 BX, 正确的操作是 ( )。

- A. LEA BX, BUF    B. MOV BX, BUF    C. LDS BX, BUF    D. LES BX, BUF

【解】 A

56. 下列串操作指令中, 一般不加重复前缀 (如 REP) 的指令是 ( )。

- A. STOSW                      B. CMPSW                      C. LODSW                      D. SCASW

【解】 C

57. SAR 和 SHR 两条指令执行后, 结果完全相同的情况是 ( )。

- |                |                |
|----------------|----------------|
| A. 目的操作数最高位为 0 | B. 目的操作数最高位为 1 |
| C. 目的操作数为任意的情况 | D. 任何情况下都不可能相同 |

【解】 A

58. 在 POP [BX] 指令中, 目的操作数的段地址和偏移地址分别在 ( )。

- |               |              |
|---------------|--------------|
| A. 没有段地址和偏移地址 | B. DS 和 BX 中 |
| C. ES 和 BX 中  | D. SS 和 SP 中 |

【解】 B

59. 对寄存器 BX 内容求补运算, 下面错误的指令是 ( )。

- |                   |              |
|-------------------|--------------|
| A. NEG BX         | B. NOT BX    |
|                   | INC BX       |
| C. XOR BX, 0FFFFH | D. MOV AX, 0 |
| INC BX            | SUB AX, BX   |

【解】 D

60. 指令 LOOPZ 的循环执行条件是 ( )。

- |                      |                     |
|----------------------|---------------------|
| A. CX ≠ 0, 并且 ZF = 0 | B. CX ≠ 0, 或 ZF = 0 |
| C. CX ≠ 0, 并且 ZF = 1 | D. CX ≠ 0, 或 ZF = 1 |

【解】 C

61. LDS SI, ES: [1000H] 指令的功能是 ( )。
- A. 把地址 1000H 送 SI
  - B. 把地址 ES: [1000H] 字单元内容送 SI
  - C. 把地址 ES: [1000H] 字单元内容送 SI, 把地址 ES: [1002H] 字单元内容送 DS
  - D. 把地址 ES: [1000H] 字单元内容送 DS, 把地址 ES: [1002H] 字单元内容送 SI

【解】 C

62. 当下列程序执行时, 屏幕上显示的内容是 ( )。

```
MOVDI, 30H
MOVAH, 2
INT 21H
```

- A. 30
- B. 0
- C. 30H
- D. 无任何显示

【解】 B

63. DOS 系统功能调用中的 1 号功能是: 从键盘输入的字符并存放在 ( )。

- A. AL
- B. BL
- C. CL
- D. DL

【解】 A

64. DOS 系统功能调用中的 2 号功能是: 在屏幕上显示 ( ) 字符。

- A. AL 中 ASCII 码所表示的
- B. DL 中 ASCII 码所表示的
- C. DS 和 DX 所指明的内存中一串
- D. DS 和 BX 所指明的内存中一串

【解】 B

### 3.3 判断题

- 1. 立即数寻址方式只能用于源操作数。( )
- 2. 立即数不允许用作目的操作数。( )
- 3. 对于所有的存储器寻址方式, 都可以采用段超越前缀。( )
- 4. 指令指针寄存器 IP 是不能通过指令访问的。( )
- 5. 代码段寄存器 CS 的内容可以被压入栈区, 也可以将堆栈中的数据弹出至 CS 中。( )
- 6. 数据段寄存器 DS 只能读出信息, 不能写入信息。( )
- 7. CS 和 IP 中的内容是不能通过指令随意改变的, 也就是 CS 和 IP 都不能用作目的操作数。( )
- 8. INC 和 DEC 指令不影响 CF 的状态。( )
- 9. AND、OR、XOR 指令执行后, 会使 CF 状态为 0。( )
- 10. 压缩型 BCD 码和非压缩型 BCD 码均有加减运算调整指令。( )
- 11. 压缩型 BCD 码和非压缩型 BCD 码均有乘除运算调整指令。( )
- 12. NOT 指令的操作数不能是立即数。( )
- 13. 在条件转移指令中, 只能用 8 位的位移量 -128 ~ +127。( )
- 14. CALL 指令和 JMP 指令的区别在于前者转移时需要保存返回地址, 而后者不需要。( )
- 15. 中断指令与 CALL 指令的不同之处在于, 中断指令还要将标志寄存器 FR 压入堆栈。( )
- 16. DOS 所有的功能子程序调用是利用 INT 21H 中断指令。( )

17. MOV AX, [BP] 的源操作数的物理地址 = SS \* 16 + BP。( )
18. 段内转移要改变 IP、CS 的值。( )
19. 条件转移指令只能用于段内直接短转移。( )
20. 立即寻址方式不能用于目的操作数字段。( )
21. 不能给段寄存器直接赋立即数。( )
22. MOV 指令执行时会影响标志位状态。( )
23. CF 位可以用来表示有符号数的溢出。( )
24. DIV 指令在执行字节除法时, 运算后的商存放在 AH 中, 余数存放在 AL 中。( )
25. 堆栈存取操作是以字节为单位的。当堆栈存入数据时, SP 减 1; 当从堆栈取出数据时, SP 加 1。( )
26. 判断下列指令是否正确, 若错误, 请指出指令的错误之处。
- 1) MOV AH, CX
  - 2) MOV 33H, AL
  - 3) MOV AX, [SI] [DI]
  - 4) MOV [BX], [SI]
  - 5) ADD BYTE PTR [BP], 256
  - 6) MOV DATA [SI], ES: AX
  - 7) JMP BYTE PTR [BX]
  - 8) OUT 230H, AX
  - 9) MOV DS, BP
  - 10) MUL 39H
27. 判断下列指令是否正确, 若错误, 请指出指令的错误之处。
- 1) POP CS
  - 2) MOV DS, 2000H
  - 3) PUSH FLAG
  - 4) MOV BP, AL
  - 5) LEA BX, 2000H
  - 6) ADD AL, [BX + DX + 10]
  - 7) AND [BX] [BP], AX
  - 8) SAR AX, 5
  - 9) CMP [DI], [SI]
  - 10) IN AL, 180H
  - 11) MUL 25
  - 12) INC IP
28. 指出下列指令的正误, 说明原因并改正。
- 1) LEA BX, AX
  - 2) XCHG BL, 100
  - 3) IN AL, 300H
  - 4) TEST AL, 100H
  - 5) MOVS [BX], [SI]

29. 指出下列传送指令的正误, 并说明原因。

- 1) POP AL
- 2) MOV CS, AX
- 3) OUT 310, AL
- 4) MOV [BX + CX], 2130H
- 5) ADD [BX], [SI]

30. 设 VAR1、VAR2 为字变量, LAB 为标号, 判断下列指令的正误, 若指令有误, 请加以改正。

- 1) ADD AR1, VAR2
- 2) MOV AL, VAR2
- 3) SUB AL, VAR1
- 4) JMP LAB [SI]
- 5) JNZ VAR1
- 6) JMP NEAR LAB

【答案】

- |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| 1. ✓  | 2. ✓  | 3. ×  | 4. ✓  | 5. ×  | 6. ×  | 7. ✓  | 8. ✓  | 9. ✓  |
| 10. ✓ | 11. × | 12. ✓ | 13. ✓ | 14. ✓ | 15. × | 16. ✓ | 17. ✓ | 18. × |
| 19. ✓ | 20. ✓ | 21. ✓ | 22. × | 23. × | 24. × | 25. × |       |       |

26. 1) ×。两操作数长度不相等。

2) ×。MOV 指令不允许目标操作数为立即数。

3) ×。在间接寻址中, 不允许两个间址寄存器同时为变址寄存器。

4) ×。MOV 指令不允许两个操作数同时为存储器操作数。

5) ×。ADD 指令要求两操作数等字长。

6) ×。源操作数形式错, 寄存器操作数不加段重设符。

7) ×。转移地址的字长至少应是 16 位的。

8) ×。对输入/输出指令, 当端口地址超出 8 位二进制数的表达范围 (即寻址的端口超出 256 个) 时, 必须采用间接寻址。

9) ✓。

10) ×。MUL 指令不允许操作数为立即数。

27. 1) ×。禁止对 CS 寄存器赋值。

2) ×。段寄存器不能直接赋值。

3) ×。无此指令, 可改用 PUSHF。

4) ×。源操作数和目标操作数的尺寸不一致。

5) ×。LEA 指令的源操作数必须是内存操作数。

6) ×。存储器寻址的有效地址描述只能使用 BX、BP、SI、DI, 不能使用 DX。

7) ×。BX 和 BP 不能同时使用, BX 只能与 SI 或 DI 相搭配。

8) ×。8086/8088 的移位指令, 当移位超过 1 位时, 就必须将移位位数赋给 CL。

9) ×。不允许在存储单元之间比较。

10) ×。端口地址超过 255 必须要放入 DX, 采用寄存器 DX 间接寻址。



- 11) ×。8086/8088 的 MUL 指令源操作不允许为立即数。
- 12) ×。不能对指令指针 IP 进行任何操作。
28. 1) ×。本条指令取存储单元有效地址，源操作数必须是存储单元而不能是寄存器 AX。  
目标操作数必须是通用寄存器之一。 改：LEA BX, [SI]
- 2) ×。不能与立即数进行交换。 改：XCHG BL, [100]
- 3) ×。300H > 255，I/O 地址由 DX 给出。 改：MOV DX, 300H  
IN AL, DX
- 4) ×。操作数尺寸不匹配，AL 是 8 位寄存器，100H 不是 8 位数据。  
改：TEST AX, 100H
- 5) ×。串操作指令中的目标操作数只能是 ES: [DI]。改：MOV ES: [DI], DS: [SI]
29. 1) ×。出栈指令为字操作。
- 2) ×。CS 不能作目标操作数。
- 3) ×。因为 310 > 255，所以端口地址应放入 DX。
- 4) ×。CX 不能用作偏移地址寄存器。
- 5) ×。内存单元间不能直接运算。
30. 1) ×。两个操作数不能都为存储单元，可改为  
MOV AX, VAR2  
ADD VAR1, AX
- 2) ×。数据类型不匹配，可改为 MOV AX, VAR2
- 3) ×。数据类型不匹配，可改为 SUB AX, VAR1
- 4) ×。寄存器相对寻址形式中不能用标号做位移量，可改为 JMP VAR1 [SI]
- 5) ×。条件跳转指令只能进行段内短跳转，所以后面只能跟短标号，可改为 JNZ LAB。
- 6) ×。缺少运算符 PTR，可改为 JMP NEAR PTR LAB。

### 3.4 填空题

1. 一条指令中一般包含有 (1) 和 (2) 两部分信息。  
【解】 (1) 操作码 (2) 操作数
2. 计算机中操作数可以在 (1) 中、(2) 中、(3) 中和 (4) 中。  
【解】 (1) 指令 (2) 寄存器 (3) 存储器单元 (4) I/O 端口
3. 对内存操作数寻址其有效地址 EA 由 (1)、(2) 和 (3) 3 部分分量之和来表示。  
【解】 (1) 基址寄存器 (2) 变址寄存器 (3) 位移量
4. BX、BP 被称为 (1) 寄存器，用它们寻址称为 (2)；将 SI、DI 称为 (3) 寄存器，用其寻址称为 (4)。  
【解】 (1) 基址 (2) 基址寻址 (3) 变址 (4) 变址寻址
5. 可以用于间接寻址的寄存器有 (1)、(2)、(3) 和 (4)。  
【解】 (1) BX (2) BP (3) SI (4) DI
6. 当采用寄存器间接寻址时，使用通用寄存器 BX, DI, SI 时，可以默认不写的段寄存器是

\_\_\_\_(1)\_\_\_\_；当使用通用寄存器 BP 时，可以默认不写的段寄存器是\_\_\_\_(2)\_\_\_\_。

【解】 (1) DS (2) SS

7. 指令 MOV [BX + SI + 10H], AX 中存储器操作数所使用的寻址方式为\_\_\_\_(1)\_\_\_\_寻址。

【解】 (1) (相对的) 基址加变址

8. 指令 MOV AX, [BX][SI] 中，目标操作数在\_\_\_\_(1)\_\_\_\_中，源操作数在\_\_\_\_(2)\_\_\_\_中，此时源操作数隐含使用的段寄存器为\_\_\_\_(3)\_\_\_\_。

【解】 (1) 寄存器 AX (2) 内存单元 (3) DS

9. 指令 ADD [BP], AL 执行时，操作的结果在\_\_\_\_(1)\_\_\_\_段中，此时计算目的操作数物理地址的表达式是\_\_\_\_(2)\_\_\_\_。

【解】 (1) 堆栈 (2)  $SS * 16 + BP$

10. 在串寻址中，使用了一种隐含的变址寄存器寻址，分别使\_\_\_\_(1)\_\_\_\_和\_\_\_\_(2)\_\_\_\_指向源串和目的串，实现字符串操作。

【解】 (1) SI (2) DI

11. 串处理指令规定源串指针寄存器必须使用\_\_\_\_(1)\_\_\_\_，源串默认为在\_\_\_\_(2)\_\_\_\_段中，也可以在其他段，但必须指明；目的串指针寄存器必须使用\_\_\_\_(3)\_\_\_\_，目的串只能在\_\_\_\_(4)\_\_\_\_段中。

【解】 (1) SI (2) DS (3) DI (4) ES

12. 操作数在 I/O 端口时，当端口地址\_\_\_\_(1)\_\_\_\_时，必须先把端口地址放在\_\_\_\_(2)\_\_\_\_中，类似于存储器寻址中的寄存器间接寻址。

【解】 (1)  $\geq 256$  (2) DX

13. 对于乘法、除法指令，其目的操作数一定在\_\_\_\_(1)\_\_\_\_或\_\_\_\_(2)\_\_\_\_中，而其源操作数可以在\_\_\_\_(3)\_\_\_\_中。

【解】 (1) AX (2) DX 和 AX (3) 寄存器或存储单元

14. 当  $AL < 80H$  时，执行 CBW 后，AH = \_\_\_\_ (1) \_\_\_\_；当  $AL \geq 80H$  时，执行 CBW 后，AH = \_\_\_\_ (2) \_\_\_\_。

【解】 (1) 00H (2) 0FFH

15. 算术右移指令 SAR 可用于实现\_\_\_\_(1)\_\_\_\_数除 2，而逻辑右移指令 SHR 则可以用来实现\_\_\_\_(2)\_\_\_\_数除 2。

【解】 (1) 对带符号 (2) 对无符号

16. 如果 TABLE 为数据段中 0032H 单元的符号名，其中存放的内容为 1234H，当执行指令 MOV AX, TABLE 后，AX = \_\_\_\_ (1) \_\_\_\_；当执行指令 LEA AX, TABLE 后，AX = \_\_\_\_ (2) \_\_\_\_。

【解】 (1) 1234H (2) 0032H

17. 已知  $BX = 7830H$ ， $CF = 1$ ，执行指令 ADC BX, 87CFH 之后， $BX =$  \_\_\_\_ (1) \_\_\_\_，标志位的状态分别为  $CF =$  \_\_\_\_ (2) \_\_\_\_， $ZF =$  \_\_\_\_ (3) \_\_\_\_， $OF =$  \_\_\_\_ (4) \_\_\_\_， $SF =$  \_\_\_\_ (5) \_\_\_\_。

【解】 (1) 0000H (2) 1 (3) 1 (4) 0 (5) 0

18. 设当前的  $SI = 1000H$ ， $DS = 5000H$ ，内存字单元 (51000H) = 1234H，则执行指令 MOV BX, [SI] 后， $BX =$  \_\_\_\_ (1) \_\_\_\_，执行指令 LEA BX, [SI] 后， $BX =$  \_\_\_\_ (2) \_\_\_\_。

【解】 (1) 1234H (2) 1000H

19. 假设  $DS = B000H$ ,  $ES = A000H$ ,  $BX = 080AH$ ,  $DI = 1200H$ ,  $(0B080AH) = 05AEH$ ,  $(0B080CH) = 4000H$ , 当执行指令  $LES\ DI, [BX]$  后,  $DS =$  (1),  $ES =$  (2),  $DI =$  (3)。

【解】 (1)  $B000H$  (2)  $4000H$  (3)  $05AEH$

20. 使用查表指令  $XLAT$  之前, 要求 (1) 寄存器指向表所在的段, (2) 寄存器指向表的首地址, (3) 寄存器中存放待查项在表中的位置与表首址的距离。

【解】 (1)  $DS$  (2)  $BX$  (3)  $AL$

21. 读取标志位指令  $LAHF$  和设置标志位指令  $SAHF$  均只对标志寄存器  $FR$  中的 (1) 标志操作。

【解】 (1) 低 8 位

22. 条件转移指令是一种短转移, 其转移范围为 (1) 字节。

【解】 (1)  $-128 \sim +127$

23. 条件转移指令的目标地址应在本条件转移指令的下一条指令地址的 (1) 字节内。

【解】 (1)  $-128 \sim +127$

24. 子程序的调用与返回分段内和段间两种情况, 对于段内调用与返回仅需修改 (1) 的值, 对于段间调用与返回需要同时修改 (2) 和 (3) 的值。执行指令  $CALL$  时, 这些值均自动保存在 (4) 中。

【解】 (1)  $IP$  (2)  $CS$  (3)  $IP$  (4) 堆栈

25. 近过程 (NEAR) 的返回指令  $RET$  把当前栈顶的一个字弹出到 (1); 远过程 (FAR) 的返回指令  $RET$  将先弹出一个字到 (2) 后, 又弹出一个字到 (3);  $IRET$  是 (4) 指令, 它从堆栈栈顶顺序弹出 3 个字分别送到 (5)、(6) 和 (7) 中。

【解】 (1)  $IP$  (2)  $IP$  (3)  $CS$  (4) 中断返回 (5)  $IP$  (6)  $CS$  (7) 标志寄存器  $FR$

26. 段内和段间的转移指令寻址方式有 (1) 和 (2) 两种。

【解】 (1) 直接寻址 (2) 间接寻址

27. 用  $CMP$  指令对无符号数比较 ( $A - B$ ), 当  $A < B$  时, 可判断标志  $CF =$  (1)。用  $CMP$  指令对带符号数比较 ( $A - B$ ), 当  $A < B$  时, 可判断标志  $SF$  (2) 0。

【解】 (1) 1 (2)  $\neq$

28. 在 4 条逻辑运算指令  $AND$ 、 $OR$ 、 $XOR$ 、 $NOT$  中, (1) 指令对标志位均无影响, 而其他 3 条指令除对标志位  $SF$ 、 $ZF$ 、 $PF$  有影响外, 还使 (2) 和 (3) 总是置“0”,  $AF$  不确定。

【解】 (1)  $NOT$  (2)  $CF$  (3)  $OF$

29. 如果要对 1 字节或一个字的数求反, 则可用指令 (1); 要对寄存器或存储单元内容中的指定位求反, 则可运用 (2) 指令。

【解】 (1)  $NOT$  (2)  $XOR$

30. 清除  $CF$  标志的指令为 (1), 设置  $DF = 1$  的指令为 (2)。

【解】 (1)  $CLC$  (2)  $STD$

31. 根据要求写出相应的指令。

1) 将附加段  $200H$  偏移地址中的数据送到  $BX$  中。 (1)

2) 将 DH 中的高 4 位求反, 低 4 位保持不变。\_(2)\_

3) 将 CL 的符号位 (D7 位) 置 1, 保持其他位不变。\_(3)\_

【解】 (1) MOV BX, ES: [200H]

(2) XOR DH, 0F0H

(3) OR CL, 80H

32. 压缩型 BCD 码加法调整指令为\_(1)\_, 非压缩型 BCD 码加法调整指令为\_(2)\_, 压缩型 BCD 码减法调整指令为\_(3)\_, 非压缩型 BCD 码减法调整指令为\_(4)\_。

【解】 (1) DAA (2) AAA (3) DAS (4) AAS

33. 下面一段程序:

MOV AX, 0

MOV AL, 09H

ADD AL, 04H

1) 若要获得 AX = 13H, 则在 ADD 指令后面加一条指令\_(1)\_。

2) 若要获得 AX = 0103H, 则在 ADD 指令后面加一条指令\_(2)\_。

【解】 (1) DAA (2) AAA

34. 试填空完善下面一段程序, 使之完成对 100 个字单元的缓冲区清零, 设缓冲区为 2000H: 0800H。

MOV AX, 2000H

MOV ES, AX

MOV DI, 0800H

\_(1)\_

MOV AL, 00H

CLD

REP STOSB

【解】 (1) MOV CX, 100

35. 试完成下面的子程序, 使其实现利用 DOS 功能调用 INT 21H, 将一个 DL 中的字节数据的低 4 位以 ASCII 码的形式显示出来。

DISPL PROC NEAR

\_(1)\_

CMP DL, 9

JBE NEXT

ADD DL, 7

NEXT: ADD DL, 30H

\_(2)\_

\_(3)\_

RET

DISPL \_(4)\_

【解】 (1) AND DL, 0FH (2) MOV AH, 2 (3) INT 21H

(4) ENDP

36. 试完成下面的程序段, 使其完成将存储单元 DA1 中压缩型 BCD 码拆成两个非压缩型 BCD 码, 低位放入 DA2 单元, 高位放入 DA3 单元, 并分别转换为 ASCII 码。

```

STRT:  MOV  AL, DA1
        MOV  CL, 4
        (1)
        OR   AL, 30H
        MOV  DA3, AL
        MOV  AL, DA1
        (2)
        OR   AL, 30H
        MOV  DA2, AL

```

【解】 (1) SHR AL, CL (2) AND AL, 0FH

37. 分析下列程序段，程序段执行后 AX = (1) , CF = (2) 。

```

MOV  AX, 0099H
MOV  BL, 88H
ADD  AL, BL
DAA
ADC  AH, 0

```

【解】 (1) 0187H (2) 0

38. 分析下列程序段，程序段执行后 AX = (1) , BX = (2) , CF = (3) 。

```

MOV  AX, 5C8FH
MOV  BX, 0AB8FH
XOR  AX, BX
XOR  AX, BX

```

【解】 (1) 5C8FH (2) 0AB8FH (3) 0

39. 分析下列程序段，程序段执行后 AX = (1) , BX = (2) , CF = (3) 。

```

XOR  AX, AX
INC  AX
NEG  AX
MOV  BX, 3FFFH
ADC  AX, BX

```

【解】 (1) 3FFFH (2) 3FFFH (3) 1

40. 源程序如下：

```

MOV  AH, 0
MOV  AL, 9
MOV  BL, 8
ADD  AL, BL
AAA
AAD
DIV  AL

```

结果 AL = (1) , AH = (2) , BL = (3) 。

【解】 (1) 01H (2) 00H (3) 08H

41. 分析下列程序段，在横线上填上适当的内容。

1) MOV AL, 0FH

MOV BL, 0C3H

XOR AL, BL

则有 AL = (1), BL = (2), CF = (3)

2) MOV BL, 93H

MOV AL, 16H

ADD AL, BL

DAA

则有 AL = (4), CF = (5), AF = (6)

3) MOV AX, BX

NOT AX

ADD AX, BX

则有 AX = (7), CF = (8)

4) MOV AL, 88H

MOV BL, 5AH

XOR AL, BL

XOR AL, BL

则有 AL = (9), CF = (10)

【解】 (1) 0CCH (2) 0C3H (3) 0 (4) 09H (5) 1  
(6) 0 (7) 0FFFFH (8) 0 (9) 88H (10) 0

42. 源程序如下:

MOV CL, 4

MOV AX, [2000H]

SHL AL, CL

SHR AX, CL

MOV [2000H], AX

1) 若程序执行前, 数据段内 [2000H] = 09H, [2001H] = 03H, 则执行后有  
[2000H] = (1), [2001H] = (2)。

2) 本程序段的功能 (3)。

【解】 (1) 39H (2) 00H (3) 将 (2000H), (2001H) 两相邻单元中存放的未组合型 BCD 码压缩成组合型 BCD 码, 并存入 (2000H) 单元, 0 → (2001H)。

43. 源程序如下:

MOV AL, 0B7H

AND AL, 0DDH

XOR AL, 81H

OR AL, 33H

JPL AB1

JMP LAB2

1) 执行程序后, AL = (1)。

2) 程序将转到 (2) 执行。

【解】 (1) 37H (2) LAB2

44. 源程序如下:

```

MOV    CX, 9
MOV    AL, 01H
MOV    SI, 1000H
LP: MOV    [SI], AL
INC    SI
SHL    AL, 1
LOOP   LP

```

- 1) 执行本程序后,  $AL = \underline{\quad(1)\quad}$ ,  $SI = \underline{\quad(2)\quad}$ ,  $CX = \underline{\quad(3)\quad}$ 。  
 2) 本程序的功能是  $\underline{\quad(4)\quad}$ 。

【解】 (1) 0 (2) 1009H (3) 0 (4) 对数据段内 1000H ~ 1008H 单元置数, 依次送入 1, 2, 4, 8, 16, 32, 64, 128, 0 共 9 个。

45. 执行完下列程序后, 回答指定的问题。

```

MOV    AX, 0
MOV    BX, 2
MOV    CX, 50
LP: ADD    AX, BX
ADD    BX, 2
LOOP   LP

```

- 问: 1) 该程序的功能是  $\underline{\quad(1)\quad}$ 。  
 2) 程序执行完成后,  $[AX] = \underline{\quad(2)\quad}$ 。

【解】 (1) 完成 0 ~ 100 间所有偶数求和的功能 (2) 2550

### 3.5 简答题

1. 简述一条指令中一般包含的信息。

【解】 一般一条指令中包含操作码和操作数两部分信息。

2. 简述计算机中操作数可能存放的位置。

【解】 计算机中操作数可能存放在指令中、寄存器中、存储单元中和 I/O 接口中。

3. 名词解释: 操作码、操作数、立即数、寄存器操作数、存储器操作数。

【解】

- 1) 操作码: 给出指令要完成的操作。
- 2) 操作数: 给出参与操作的对象。
- 3) 立即数: 要参与操作的数据在指令中。
- 4) 寄存器操作数: 要参与操作的数据, 存放在指定的寄存器中。
- 5) 存储器操作数: 要参与操作的数据, 存放在指定的存储单元中。

4. 什么是寻址方式?

【解】 指令中用以描述操作数所在位置的方法称为寻址方式。

5. 内存操作数的逻辑地址表达式为——段基值: 偏移量, 试写出偏移量的有效地址 EA 的计算通式。

【解】 偏移量的有效地址 EA 的计算通式:  $EA = \text{基址值} + \text{变址值} + \text{位移量}$ 。

其中，基址值为 BX 或 BP 的内容，变址值为 SI 或 DI 的内容，位移量为 8 位 disp8 或 16 位 disp16 数据。

6. 指出下列各指令目的操作数所在的位置，并写出相应的地址。

- 1) ADD [SI], AX
- 2) MOV CL, BUF
- 3) DEC [BP + 50H]
- 4) OUT 20H, AL
- 5) JMP 2000H: 0100H
- 6) JMP [BX]
- 7) JMP BX

**【解】**

- 1) 目的操作数在内存单元中，其逻辑地址为 DS: [SI]，其物理地址 =  $DS * 16 + SI$ 。
  - 2) 目的操作数在寄存器 CL 中。
  - 3) 目的操作数在内存单元中，其逻辑地址为 SS: [BP + 50H]，其物理地址 =  $SS * 16 + BP + 50H$ 。
  - 4) 目的操作数在端口中，端口地址为 20H。
  - 5) 目的操作数在指令中。
  - 6) 目的操作数在内存单元中，其逻辑地址为 DS: [BX]，其物理地址 =  $DS * 16 + BX$ 。
  - 7) 目的操作数在寄存器 BX 中。
7. 两个逻辑段地址分别为 2345H: 0000H 和 2000H: 3450H，它们对应的物理地址是多少？说明了什么？

**【解】** 这两个逻辑段对应的物理地址均为 23450H，说明对应同一个物理地址可以有不同的逻辑地址，即物理地址是唯一的，逻辑地址不唯一。

8. 设 DS = 3000H, BX = 2000H, [SI] = 1000H, MAX = 1230H, 则指令 MOV AX, MAX [BX][SI] 的源操作数物理地址为多少？

**【解】** 源操作数物理地址 =  $DS * 16 + BX + SI + MAX = 30000H + (2000H + 1000H + 1230H) = 34230H$ 。

9. 设 [DS] = 6000H, [ES] = 2000H, [SS] = 1500H, [SI] = 00A0H, [BX] = 0800H, [BP] = 1200H, 数据变量 VAR 为 0050H, 请分别指出下列各条指令源操作数的寻址方式？它的物理地址是多少？

- 1) MOV AX, BX
- 2) MOV DL, 80H
- 3) MOV AX, VAR
- 4) MOV AX, VAR [BX] [SI]
- 5) MOV AL, 'B'
- 6) MOV DI, ES: [BX]
- 7) MOV DX, [BP]
- 8) MOV BX, 20H [BX]

**【解】**

- 1) 寄存器寻址。因为源操作数是寄存器，所以寄存器 BX 就是操作数的地址。



- 2) 立即寻址。操作数 80H 存放于代码段中指令码 MOV 之后。
- 3) 直接寻址。
- 4) 基址变址相对寻址。操作数的物理地址 =  $DS \times 16 + SI + BX + VAR = 60000H + 00A0H + 0800H + 0050H = 608F0H$ 。
- 5) 立即寻址。
- 6) 寄存器间接寻址。操作数的物理地址 =  $ES \times 16 + BX = 20000H + 0800H = 20800H$ 。
- 7) 寄存器间接寻址。操作数的物理地址 =  $SS \times 16 + BP = 15000H + 1200H = 16200H$ 。
- 8) 寄存器相对寻址。操作数的物理地址 =  $DS \times 16 + BX + 20H = 60000H + 0800H + 20H = 60820H$ 。

10. 试说明指令 MOV BX, 5[BX] 与指令 LEA BX, 5[BX] 的区别。

【解】前者是数据传送类指令，表示将数据段中以 (BX + 5) 为偏移地址的 16 位数据送寄存器 BX；后者是取偏移地址指令，执行的结果是  $BX = (BX + 5)$ 。

11. 设当前  $BX = 0158H$ ,  $DI = 10A5H$ , 位移量 =  $1B57H$ ,  $DS = 2100H$ ,  $SS = 1100H$ ,  $BP = 0100H$ , 段寄存器默认，写出以下各寻址方式的物理地址。

- 1) 直接寻址。
- 2) 寄存器间接寻址 (设采用 BX)。
- 3) 寄存器相对寻址 (设采用 BP)。
- 4) 基址变址寻址 (设采用 BX 和 DI)。
- 5) 相对基址变址寻址 (设采用 BP、DI 和位移量)。

【解】

- 1) 物理地址 =  $DS \times 16 + \text{位移量} = 21000H + 1B57H = 22857H$ 。
  - 2) 物理地址 =  $DS \times 16 + BX = 21000H + 0158H = 21158H$ 。
  - 3) 物理地址 =  $SS \times 16 + BP = 11000H + 0100H = 11100H$ 。
  - 4) 物理地址 =  $DS \times 16 + BX + DI = 21000H + 0158H + 10A5H = 221FDH$ 。
  - 5) 物理地址 =  $SS \times 16 + BP + DI = 11000H + 0100H + 10A5H + 1B57H = 13CFCH$ 。
12. 在转移类指令中，对转移的目标地址的寻址方式有几种？段内转移的范围是多大？段间转移的范围是多大？条件转移的范围是多大？

【解】在转移类指令中，对转移的目标地址的寻址方式有段内直接转移、段内间接转移、段间直接转移和段间间接转移。段内直接转移的范围为  $-32768 \sim +32767$ ，段内短转移是在当前 IP 偏移值的  $-128 \sim +127$  字节的范围内 (又称为相对寻址)。段间转移的范围可以在 1MB 范围内。条件转移指令均为段内短转移，其转移的范围为 IP 当前值的  $-128 \sim +127$  字节的偏移量范围内。

13. 条件转移指令均为相对近转移指令，请解释“相对近转移”的含义。若需往较远处进行条件转移，则应怎么做？

【解】“相对”是指相对于指令指针寄存器 IP 的当前值进行的转移。相对近转移范围为  $-128 \sim +127$ 。若转移的目标地址较远，超出指令要求的范围，则可在相对转移的目标地址处放置一条无条件转移指令，从而实现较远距离的转移。

14. 试比较无条件转移指令、条件转移指令、调用指令和中断指令有什么异同？

**【解】**

1) 无条件转移指令的操作是无条件地使程序转移到指定的目标地址, 并从该地址开始执行新的程序段, 其转移的目标地址既可以是在当前逻辑段, 也可以是在不同的逻辑段。

2) 条件转移指令是在满足一定条件下使程序转移到指定的目标地址, 其转移范围很小, 在当前逻辑段的  $-128 \sim +127$  地址范围内。

3) 调用指令用于调用程序中常用到的功能子程序, 是在程序设计中就设计好的。根据所调用过程入口地址的位置, 可将调用指令分为段内调用 (入口地址在当前逻辑段内) 和段间调用。在执行调用指令后, CPU 要保护断点。对段内调用是将其下一条指令的偏移地址压入堆栈, 对段间调用则要保护其下一条指令的偏移地址和段基地址, 然后将子程序入口地址赋给 IP (或 CS 和 IP)。

4) 中断指令是因一些突发事件而使 CPU 暂时中止它正在运行的程序, 转去执行一组专门的中断服务程序, 并在执行完后返回原被中止处继续执行原程序。它是随机的。在响应中断后 CPU 不仅要保护断点 (即 INT 指令下一条指令的段地址和偏移地址), 还要将标志寄存器 FLAGS 压入堆栈保存。

15. 已知  $DS = 5000H$ ,  $CS = 6000H$ ,  $BX = 1278H$ ,  $SI = 345FH$ ,  $(546D7H) = 0$ ,  $(546D8H) = 80H$ 。在分别执行下面两条段内转移指令后, 实际转移的目标物理地址是多少?

- 1) `JMP BX`      2) `JMP [BX + SI]`

**【解】**

1) 段内转移的目标地址存放在寄存器 BX 中, 故实际的转移地址是  $6000H: 1278H$ , 其物理地址  $= CS * 16 + BX = 60000H + 1278H = 61278H$ 。

2) 段内转移的目标地址存放在内存单元中, 该单元的物理地址  $= DS * 16 + BX + SI = 50000H + 1278H + 345FH = 546D7H$ , 故实际的转移地址是  $6000H: 8000H$ , 其物理地址  $= CS * 16 + 8000H = 60000H + 8000H = 68000H$ 。

16. 若一个堆栈段的起始地址为  $3520H: 0000H$ , 栈区的长度为  $0100H$ , 当前 SP 的内容为  $0020H$ , 试问:

- 1) 栈顶的物理地址是什么?
- 2) 栈底的物理地址是什么?
- 3) 栈区中已有字节数为多少?
- 4) 存入数据  $1234H$  和  $5678H$  后 SP 的内容是多少?

**【解】** 堆栈的最高地址叫栈底, 堆栈指针 SP 总是指向栈顶。

1) 栈顶的物理地址  $= SS * 10H + SP = 3520H * 10H + 0020H = 35220H$ 。

2) 栈底的物理地址  $= SS * 10H + 0120H = 35320H$ 。

3) 栈区中已有字节数  $= 100H$ 。

4) 8086/8088 的堆栈操作为字操作。在压入两个字数据  $1234H$  和  $5678H$  后, SP 减 4, 所以 SP 的内容为  $SP - (字数 * 2) = 0020H - 2 * 2 = 001CH$ , SS 的内容不变。

17. 简述 RET 指令与 IRET 指令的异同点。

**【解】** RET 指令与 IRET 指令的异同见表 3-1。

表 3-1 RET 指令与 IRET 指令的异同

	RET	IRET
返回指令	为子程序返回指令	为中断返回指令
放置位置	放置在子程序的末尾	放置在中断服务程序的末尾
功能	从栈顶处弹出断点地址，使 CPU 返回到主程序的断点处继续执行	控制从栈顶处顺序弹出程序断点送回 CS 和 IP 中，弹出保存的断点时的标志寄存器内容送回 FR 中，使 CPU 返回到中断时的断点处继续执行后续程序
返回情况	分段内返回（断点地址只有 IP 值）和段间返回（断点地址为 CS 和 IP）两种情况	—
参数值	还可以带参数值，即 RET n 形式	—

18. 按下列要求写出相应的指令或程序段。

- 1) 写出两条使 AX 内容为 0 的指令。
- 2) 使 BL 寄存器中的高 4 位和低 4 位互换。
- 3) 屏蔽 CX 寄存器的 b11、b7 和 b3 位。
- 4) 测试 DX 中的 b0 和 b8 位是否为 1。

【解】

- 1) MOV AX, 0  
XOR AX, AX ; AX 寄存器自身相异或，可使其内容清 0
- 2) MOV CL, 4  
ROL BL, CL ; 将 BL 内容循环左移 4 位，可实现其高 4 位和低 4 位的互换
- 3) AND CX, 0F777H ; 将 CX 寄存器中需屏蔽的位“与”0。也可用“或”指令实现
- 4) AND DX, 0101H ; 将需测试的位“与”1，其余“与”0 屏蔽掉  
CMP DX, 0101H ; 与 0101H 比较  
JZ ONE ; 若相等，则表示 b0 和 b8 位同时为 1

19. 采用最少的指令，实现下述要求的功能。

- 1) AH 的高 4 位清 0。
- 2) 将 AH 中的非压缩型 BCD 码转化成 ASCII 码。
- 3) AL 的高 4 位取反。
- 4) AL 的高 4 位移到低 4 位，高 4 位清 0。

【解】

- 1) AND AH, 0FH
- 2) OR AH, 30H
- 3) XOR AL, 0F0H
- 4) MOV CL, 4  
SHR AL, CL

20. 试比较 SUB AL, 09H 与 CMP AL, 09H 这两条指令的异同，若 AL = 08H，写出执行两条指令后，对 6 个状态标志位的影响。

【解】 SUB AL, 09H 与 CMP AL, 09H 指令功能的对比见表 3-2。

表 3-2 SUB AL, 09H 与 CMP AL, 09H 指令功能的对比

SUB AL, 09H	CMP AL, 09H
dest ← dest - src	dest - src 不回送结果，只影响状态标志位
双字节指令	双字节指令
AL = 08H - 09H = -01H = FFH	AL = 08H
状态标志位影响相同	对状态标志位影响相同
OF SF ZF AF PF CF	OF SF ZF AF PF CF
0 1 0 1 1 1	0 1 0 1 1 1

21. 阅读下面 8086 程序段，指出该程序段的功能。

```

AGN1: MOV    AL, [DI]
      INC    DI
      TEST   AL, 04H
      JE     AGN1
      ...
AGN2: ...

```

【解】 当 AL 的 D2 位为 0 时，程序转至 AGN2。

22. 分析下面的程序段，分析 AL 满足什么条件程序将转到 LOP 标号执行？

```

CMP    AL, 0FFH
JNL    LOP
      ...
LOP: ...

```

【解】 指令 JNL NEXT 为对两个带符号数比较后的状态进行判断，实现转移的条件转移指令。因此，只有当 AL 中的带符号数不小于（即大于等于）0FFH（即为 -1）时，满足条件才能转至标号 NEXT 处执行。故  $AL \geq -1$ 。

23. 把首地址为 BLOCK 的字数组的第 6 个字传送到 CX 寄存器中。试写出该指令序列，要求使用两种寻址方式。

【解】 1) 以 BX 的寄存器间接寻址：

```

MOV    BX, BLOCK + 10
MOV    CX, [BX]          ; 将 BLOCK 开始的第 6 个字的地址存入 CX

```

2) 用 BX、SI 的基址变址寻址：

```

LEA    BX, BLOCK
MOV    SI, 10
MOV    CX, [BX + SI]     ; 将 BLOCK 开始的第 6 个字的地址存入 CX

```

24. 下列指令完成了什么功能？

- 1) ADD AL, DH
- 2) ADC BX, CX
- 3) SUB AX, 2710H
- 4) DEC BX
- 5) NEG CX
- 6) INC BL

7) MUL BX

8) DIV CL

【解】

1)  $AL + DH \rightarrow AL$

2)  $BX + CX + CF \rightarrow BX$

3)  $AX - 2710H \rightarrow AX$

4)  $BX - 1 \rightarrow BX$

5)  $0 - CX \rightarrow CX$

6)  $BL + 1 \rightarrow BL$

7)  $AX * BX \rightarrow DX, AX$

8)  $AX/CL$  商  $\rightarrow AL$ , 余数  $\rightarrow AH$

25. 已知  $AX = 789AH$ ,  $BP = 0F4A2H$ ,  $CF = 1$ , 写出下列指令单独执行后的结果及  $CF$  状态。

1) ADD AX, 0800H

2) ADC AX, 5

3) DEC AX

4) SUB AX, 5678H

5) AND AX, 00FFH

6) OR AX, 8888H

7) SAR AX, 1

8) RCR AX, 1

9) SBB BP, 0A034H

【解】

1)  $AX = 809AH$ ,  $CF = 0$

2)  $AX = 78A0H$ ,  $CF = 0$

3)  $AX = 7899H$ ,  $CF = 1$

4)  $AX = 2222H$ ,  $CF = 0$

5)  $AX = 009AH$ ,  $CF = 0$

6)  $AX = 0F89AH$ ,  $CF = 0$

7)  $AX = 3C4DH$ ,  $CF = 0$

8)  $AX = 0BC4DH$ ,  $CF = 0$

9)  $BP = 546DH$ ,  $CF = 0$

26. 写出满足下列各要求的指令：

1) 将有效地址为  $1000H$  的内存单元内容送到  $BX$  寄存器中。

2) 将偏移地址为  $1000H$  的内存单元的有效地址送到  $BX$  寄存器中。

3) 将源操作数为  $SI$  间接寻址方式中的数据送到  $SI$  寄存器中。

4) 从偏移地址为  $1000H$  开始的内存中取出双字送入  $BX$  和  $DS$  中。

【解】

1) MOV BX, [1000H]

2) LEA BX, [1000H]

3) MOV SI, [SI]

4) LDS BX, [1000H]

27. 写出把首地址为  $BLOCK$  的数组的第 6 个字送到  $DX$  寄存器的指令。要求使用以下几种寻址方式：

1) 寄存器间接寻址。

2) 寄存器相对寻址。

3) 基址变址寻址。

【解】

1) 使用寄存器间接寻址，把首地址为  $BLOCK$  的数组的第 6 个字送到  $DX$  寄存器的指令为

```
MOV    BX, BLOCK
ADD    BX, 12
MOV    DX, [BX]
```

2) 使用寄存器相对寻址，把首地址为 BLOCK 的字数组的第 6 个字送到 DX 寄存器的指令为

```
MOV    BX, BLOCK
MOV    DX, [BX + 12]
```

3) 使用基址变址寻址，把首地址为 BLOCK 的字数组的第 6 个字送到 DX 寄存器的指令为

```
MOV    BX, BLOCK
MOV    SI, 12
MOV    DX, [BX + SI]
```

28. 设 DS = 2000H, BX = 0100H, SI = 0002H, (21200H) = 4C2AH, (21202H) = 8765H。试求执行以下各条指令以后, AX 寄存器的内容是什么?

- 1) MOV AX, 1200H
- 2) MOV AX, [1200H]
- 3) MOV AX, 1100[BX]
- 4) MOV AX, 1100[BX][SI]

【解】

- 1) AX = 1200H
- 2) AX = 4C2AH
- 3) AX = 4C2AH
- 4) AX = 8765H

29. 已知 DS = 1000H, BX = 0200H, SI = 02H, 内存 10200H ~ 10205H 单元的内容分别为 10H, 2AH, 3CH, 46H, 59H, 6BH。下列每条指令执行完后 AX 寄存器的内容各是什么?

- 1) MOV AX, 0200H
- 2) MOV AX, [200H]
- 3) MOV AX, BX
- 4) MOV AX, 3 [BX]
- 5) MOV AX, [BX + SI]
- 6) MOV AX, 2 [BX + SI]

【解】

- 1) 0200H
- 2) 2A10H
- 3) 0200H
- 4) 5946H
- 5) 463CH
- 6) 6B59H

30. 试写出下面程序段中每条指令执行后 AL 的值及 OF、SF、ZF、CF、AF 和 PF 等状态标志的变化。

```

XOR    AL, AL
MOV    AL, 7FH
ADD    AL, 1
ADD    AL, 80H
MOV    AH, 2
SUB    AL, AH

```

【解】

```

XOR    AL, AL      ; AL=00H、OF=0、SF=1、ZF=1、CF=0、AF=0、PF=1
MOV    AL, 7FH     ; AL=7FH、OF=0、SF=1、ZF=1、CF=0、AF=0、PF=1
ADD    AL, 1       ; AL=80H、OF=1、SF=1、ZF=0、CF=0、AF=1、PF=0
ADD    AL, 80H     ; AL=00H、OF=1、SF=1、ZF=1、CF=1、AF=0、PF=1
MOV    AH, 2       ; AL=00H、OF=1、SF=1、ZF=1、CF=1、AF=0、PF=1
SUB    AL, AH      ; AL=0FEH、OF=0、SF=1、ZF=0、CF=1、AF=1、PF=0

```

31. 设 AX = AA55H, 写出下面程序段中每条指令执行后 AX = ? CF = ?

```

SHR    AL, 1
RCL    AX, 1
MOV    CL, 3
SAL    AX, CL
ROL    AH, 1
SAR    AL, 1
ROR    AL, CL
SHL    AX, 1
RCR    AH, CL

```

【解】

```

SHR    AL, 1      ; AX = AA2AH, CF = 1
RCL    AX, 1      ; AX = 5455H, CF = 1
MOV    CL, 3      ; AX = 5455H, CF = 1
SAL    AX, CL     ; AX = A2A8H, CF = 0
ROL    AH, 1      ; AX = 45A8H, CF = 1
SAR    AL, 1      ; AX = 45D4H, CF = 0
ROR    AL, CL     ; AX = 459AH, CF = 1
SHL    AX, 1      ; AX = 8B34H, CF = 0
RCR    AH, CL     ; AX = D134H, CF = 0

```

32. 已知 AX = AA55H, BX = 55AAH, 在执行下列各指令后, AX = ?

- |                 |                 |
|-----------------|-----------------|
| 1) NOT AX       | 2) NEG AX       |
| 3) AND AX, BX   | 4) OR AX, BX    |
| 5) XOR AX, BX   | 6) CMP AX, BX   |
| 7) TEST AX, 80H | 8) OR AX, 80H   |
| 9) AND AX, 0FH  | 10) XOR AX, 0FH |

【解】

- |          |           |          |          |
|----------|-----------|----------|----------|
| 1) 55AAH | 2) 55ABH  | 3) 0000H | 4) FFFFH |
| 5) FFFFH | 6) AA55H  | 7) AA55H | 8) AAD5H |
| 9) 0005H | 10) AA5AH |          |          |

33. 已知 AX = 8060H, DX = 03F8H, 端口 PORT1 的地址是 48H, 内容为 40H; PORT2 的地址是 84H, 内容为 85H。请指出下列指令执行后的结果。

- 1) OUT DX, AL
- 2) IN AL, PORT1
- 3) OUT DX, AX
- 4) IN AX, 48H
- 5) OUT PORT2, AX

【解】

- 1) 将 60H 输出到地址为 03F8H 的端口中。
- 2) 从 PORT1 读入一个字节数据, 执行结果: [AL] = 40H。
- 3) 将 AX = 8060H 输出到地址为 03F8H 的端口中。
- 4) 由 48H 端口读入 16 位二进制数。
- 5) 将 8060H 输出到地址为 84H 的端口中。

34. 在下列程序段的括号中分别填入以下指令

- 1) LOOP NEXT
- 2) LOOPE NEXT
- 3) LOOPNE NEXT

试说明在这 3 种情况下, 程序段执行完后, AX、BX、CX、DX 的内容分别是什么。

```
START: MOV AX, 01H
        MOV BX, 02H
        MOV DX, 03H
        MOV CX, 04H
NEXT:   INC AX
        ADD BX, AX
        SHR DX, 1
        ( )
```

【解】

- 1) AX = 05H, BX = 10H, CX = 0, DX = 0
- 2) AX = 02H, BX = 04H, CX = 03H, DX = 01H
- 3) AX = 03H, BX = 07H, CX = 02H, DX = 0

35. 试写出能完成下列各操作功能的指令或程序段。

- 1) 将累加器 AX 清零, 同时进位标志 CF 清零。
- 2) 将 DS: [1000H] 字节单元内容与寄存器 CL 内容相加后存入 DS: [2000H] 字节单元。
- 3) 不用立即数, 也不用存储单元, 完成  $AL \leftarrow 0FFH$ ,  $BL \leftarrow 0$ ,  $CL \leftarrow 1$ 。
- 4) 取 AX 的低 4 位。
- 5) 将 BX 的高 8 位置 1, 其他位不变。
- 6) 把标志寄存器中溢出位 OF 变反。

【解】

- 1) XOR AX, AX 或者 AND AX, 0
- 2) ADD CL, DS: [1000H]



```

MOV    DS: [2000H], CL
3) XOR   CL, CL                ; CL = 0, CF = 0
MOV     BL, CL                 ; BL = 0
MOV     AL, CL                 ; AL = 0
NOT      AL                    ; AF = FFH
INC      CL                    ; CF = D1H
4) AND   AX, 000FH
5) OR    BX, 0FF00H
6) PUSHF
POP      AX
XOR      AX, 800H
PUSH     AX
POPF

```

36. 根据下列各要求，写出程序段。

- 1) 判断 AL 为负，则转至 NEXT。
- 2) 判断字节变量 DA1 为 0，则转至 NEXT。
- 3) 判断 AL 的 D1 位为 1，则转至 NEXT。
- 4) 判断字变量 DA1 的 D1 位为 0，则转至 NEXT。
- 5) 比较 AH 与字节变量 DA1 中的数，若 AH 不等于 DA1，则转至 NEXT。
- 6) 比较字节变量 DA1 与字符“A”，若 DA1 ≥ “A”，则转至 NEXT。

【解】

1) OR    AL, AL	2) CMP    DA1, 0
JS      NEXT	JZ      NEXT
3) TEST  AL, 2	4) TEST  DA1, 0002H
JNZ     NEXT	JZ      NEXT
5) CMP   AH, DA1	6) CMP   DA1, 'A'
JNE     NEXT	JAE     NEXT

### 3.6 分析程序题

1. 试判断下列程序执行后，BX 中的内容。

```

MOV     CL, 3
MOV     BX, 0B7H
ROL     BX, 1
ROR     BX, CL

```

【解】 该程序段是首先将 BX 内容不带进位循环左移 1 位，然后循环右移 3 位。即相当于将原 BX 内容不带进位循环右移 2 位，故结果为 BX = 0C02DH

2. 写出下面每条指令执行以后，有关寄存器的内容。

```

XOR     AX, AX
MOV     AX, 0ABCDH

```

```

INC    AX
MOV    CL, 4
SHR    AL, CL
XOR    AX, 0FFFH
OR     AX, 6000H
MOV    SP, 0100H
PUSH   AX
POP    BX

```

【解】

```

AX = 0000H
AX = 0ABCDH
AX = 0ABCEH
CL = 04H
AX = 0ABCH
AX = 0543H
AX = 6543H
SP = 0100H
SP = 00FEH
BX = 6543H

```

3. 设初值 AX = 1234H, CX = 0004H, 在执行下列程序段后, AX = ?

```

AND    AX, AX
JZ     DONE
SHL    CX, 1
      ROR    AX, CL
      JMP    LP
DONE:  ROL    AX, CL
LP:    NOP

```

【解】

```

AND    AX, AX      ; AX = 1234H, OF = 0, PF = 0, ZF = 0, CF = 0, AF = 0, SF = 0
JZ     DONE        ; 为零, 转 DONE; 非零, 则顺序执行
SHL    CX, 1        ; CX = 0008H
ROR    AX, CL        ; AX = 3412H
JMP    LP
DONE:  ROL    AX, CL  ; AX = 2341H
LP:    NOP

```

程序段运行后, AX = 3412H

4. 阅读下面程序后, 回答问题。

```

LEA    SI, BUF
MOV    CX, 100
CLD
LP1:   LODSB
      CMP    AL, 39H

```

```

JE      LP2
LOOP    LP1
JMP     LP3

```

LP2: ...

问：1) 该程序完成的功能是什么？

2) 程序执行完后，若  $CX \neq 0$ ，则程序转移到哪里？

3) 程序执行完后，如果  $CX = 0$ ，则程序转移到哪里？

【解】

1) 查找 BUF 起始的 100 字节数据中是否有 39H，有则转到 LP2，无则转到 LP3。

2) 若  $CX \neq 0$ ，则程序转移到 LP2 去执行。

3) 如果  $CX = 0$ ，则程序转移到 LP3 去执行。

5. 阅读下面子程序后，回答指定问题。

```

SUB1    PROC    FAR
        TEST    AL, 80H
        JE      LP1
        TEST    BL, 80H
        JNE     LP2
        JMP     LP3
LP1:     TEST    BL, 80H
        JE      LP2
LP3:     XCHG    AL, BL
LP2:     RET
SUB1    ENDP

```

问：

1) 子程序的功能是什么？

2) 如调用前  $AL = 88H$ ， $BL = 77H$ ，则在返回时， $AL = ?$   $BL = ?$

【解】

1) 该子程序的功能是测试 AL 和 BL 内容是否为异号数。若是，则互相交换；若不是，则不改变。

2) 若在调用前  $AL = 88H$ ， $BL = 77H$ ，则在返回时， $AL = 77H$ ， $BL = 88H$ 。

6. 源程序如下：

```

MOV     AX, SEG TABLE    ; TABLE 为表头
MOV     ES, AX
MOV     DI, OFFSET TABLE
MOV     AL, '0'
MOV     CX, 100
CLD
REPNE   SCASB

```

问：1) 该段程序完成什么功能？

2) 该段程序执行完毕之后，ZF 和 CX 有几种可能的数值？各代表什么含义？

【解】

1) 从目的串中查找是否包含字符“0”，若找到则停止，否则继续重复搜索。

2)  $ZF = 1$ ，说明已找到字符； $ZF = 0$ ，说明未找到字符； $CX \neq 0$ ，说明中途已找到字符退出； $CX = 0$ ，且  $ZF = 0$ ，说明串中无字符“0”。

7. 源程序如下：

```
CMP    AX, BX
JNC    L1
JZ     L2
JNS    L3
JNO    L4
JMP    L5
```

设  $AX = 74C3H$ ， $BX = 95C3H$ ，则程序最后将转到哪个标号处执行？试说明理由。

【解】 因为  $74C3H - 95C3H = DF00H$ ，且  $CF = 1$ ， $ZF = 0$ ， $SF = 1$ ， $OF = 1$ ，所以程序将转到 L5 标号处执行。

8. 阅读程序段，说明程序段执行后转移到哪里？

```
MOV    AX, 1379H
MOV    BX, 8ACEH
ADD    AX, BX
JNO    L1
JNC    L2
JMP    L3
```

【解】 本题意为  $AX + BX$  无溢出，则程序转到 L1，有溢出但没进位程序将转移到 L2，否则程序转移到 L3。由于  $1379H + 8ACEH$  无溢出，则程序转到 L1。

9. 阅读程序段，说明程序段执行后转移到哪里？

```
MOV    AX, 8765H
MOV    BX, 1234H
SUB    AX, BX
JNO    L1
JNC    L2
JMP    L3
```

【解】 本题意为  $AX - BX$  无溢出，则程序转到 L1，有溢出但没进位程序将转移到 L2，否则程序转移到 L3。由于  $AX - BX = 8765H - 1234H = 7531H$ ， $OF = 1$ （有溢出）， $CF = 0$ （无借位），因此程序段执行后转 L2。

10. 阅读程序段，说明程序段的功能。

```
START: LEA    BX, CHAR
      MOV    AL, 'A'
      MOV    CX, 26
LOP1: MOV    [BX], AL
      INC    AL
      INC    BX
      LOOP   LOP1
```

HLT

【解】 程序段的功能是在 CHAR 开始的 26 个字节单元中依次存放 A ~ Z。

11. 阅读下面程序段，请指出各程序段的功能。

```
1)      MOV    CX, 10
        CLD
        LEA    SI, First
        LEA    DI, Second
        REP    MOVSB

2)      CLD
        LEA    DI, [0404H]
        MOV    CX, 0080H
        XOR    AX, AX
        REP    STOSW

3)      MOV    CX, 10
        MOV    AL, 0
        MOV    BL, AL
LP:      ADD    AL, BL
        INC    BL
        LOOP   LP
```

【解】

- 1) 将 First 开始的 10 字节传送到 Second 处。
- 2) 将偏移量为 0404H 开始 80H 个字单元清 0。
- 3) 计算 0 ~ 9 之和，结果为 55。

### 3.7 编程题

1. 有 3 个无符号数分别在 AL、BL、CL 中，其中有两个相同，编写一程序找出不相同的数并送入 DL 中。

【解】

```
      CMP     AL, BL      ; AL 与 BL 比较
      JNZ     L1          ; AL ≠ BL, 转去 L1
      MOV     DL, CL      ; AL = BL, 则 CL 一定是不同数, 将 CL 存入 DL
      JMP     L3          ; 转结束
L1:    CMP     AL, CL      ; AL 与 CL 比较
      JNZ     L2          ; AL ≠ BL, 且 AL ≠ CL, 则 AL 一定是不同数
      MOV     DL, BL      ; AL ≠ BL, 但 AL = CL, 则 BL 一定是不同数, 将 BL 存入 DL
      JMP     L3          ; 转结束
L2:    MOV     DL, AL      ; 将 AL 存入 DL
L3:    HLT
```

2. 设在 DA 开始的单元中存放有一个 4 字节的有符号补码数，高字节存在较高地址单元。试编写一个程序段，完成求出此 4 字节的有符号补码数的绝对值，并存入 ABS 开始的单元。

【解】 分析：将 4 字节的有符号补码数从内存取出后，首先判符号位，正数的绝对值就是其本身，负数的绝对值需要求出相应的原码，去掉负号即为绝对值。

```

START:  LEA    BX, DA
        MOV    AX, [BX]
        MOV    DX, [BX+2]
        OR     DX, DX
        JNS    LP1
        NOT    AX
        NOT    DX
        ADD    AX, 1
        ADC    DX, 0
        AND    DX, 7FFFH
LP1:    LEA    BX, ABS
        MOV    [BX], AX
        MOV    [BX+2], DX
        HLT

```

3. 编写程序实现将首地址为 STR、长度为 N 的字符串顺序颠倒（见图 3-1）。

【解】

```

R-STR:  LEA    SI, STR
        MOV    CX, N
        MOV    DI, SI
        ADD    DI, CX
        DEC    DI
        SHR    CX, 1      ; [CX] = N/2
NEXT:   MOV    AL, [SI]
        XCHG   AL, [DI]
        MOV    [SI], AL
        INC    SI
        DEC    DI
        LOOP   NEXT
        HLT

```

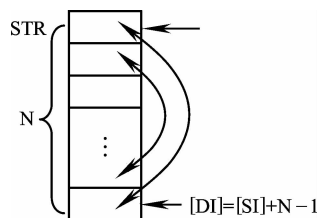


图 3-1 示意图

4. 将段地址为 1000H、偏移地址为 100H 开始的 100 个字单元清为 0。

【解】 分析：可以采用 MOV 指令实现，也可以采用串操作指令 STOSW 实现。

方法 1：

```

        MOV    AX, 1000H      ; AX = 1000H
        MOV    DS, AX         ; 给段寄存器 DS 赋值
        MOV    SI, 100H       ; 设置地址指针
        MOV    CX, 100        ; 设置循环次数
        XOR    AX, AX         ; AX = 0
LP:     MOV    [SI], AX        ; 将字单元清 0
        ADD    SI, 2          ; 字地址指针加 2
        LOOP   LP             ; CX = CX - 1, 若 CX ≠ 0 转到 LP

```

HLT

方法 2:

```
MOV    AX, 1000H
MOV    ES, AX
MOV    DI, 100H
XOR    AX, AX
MOV    CX, 100
CLD                                ; 令 DF=0, 设置串操作中指针增加
REP    STOSW                       ; ES: [DI]←AX, 循环执行 CX 次, 且每次 DI 自动加 2
HLT
```

5. 分别写出实现如下功能的程序段:

- 1) 双字减法 (被减数 7B1D2A79H, 减数 53E2345FH)。
- 2) 使用移位指令实现一个字乘 18 的运算。
- 3) 使用移位指令实现一个字除以 10 的运算。
- 4) 将 AX 中间 8 位、BX 低 4 位、DX 高 4 位拼成一个新字。
- 5) 将 BX 中的 4 位压缩 BCD 数用非压缩 BCD 数形式顺序放在 AL、BL、CL、DL 中。

【解】

1) 双字减法的程序段是:

```
MOV    AX, 2A79H                ; 被减数的低位字送 AX
SUB     AX, 345FH                ; 低位字相减, 结果送 AX
MOV     BX, 7B1DH               ; 被减数的高位字送 BX
SBB     BX, 53E2H               ; 高位字相减, 并减去低位字相减产生的借位, 结果送 BX
```

2) 使用移位指令实现一个字乘 18 的程序段是:

```
MOV     AX, 05F7H               ; 被乘数送 AX
SHL     AX, 1                   ; 被乘数乘以 2, 结果在 AX 中
MOV     BX, AX                  ; 被乘数乘以 2, 结果暂存到 BX
MOV     CL, 3                   ; 设置移位位数 3
SHL     AX, CL                  ; 被乘数再乘以 8 (共乘以 16), 结果在 AX 中
ADD     AX, BX                  ; 被乘数再乘以 18, 结果在 AX 中
```

3) 使用移位指令实现一个字除以 10 的运算, 必须将  $X/10$  拆分成多项的和, 而每一项都应是 2 的某次幂的倒数。利用等比级数的前 N 项和公式, 可求出  $A_0 = X/8$ , 公比  $Q = -1/4$ , 故  $X/10 = X/8 - X/32 + X/128 - X/512 + \dots$ , 程序段编写如下:

```
MOV     AX, FE00H               ; 被除数送 AX
MOV     CL, 3                   ; 设置移位位数 3
SHR     AX, CL                  ; 被乘数除以 8, 结果在 AX 中
MOV     BX, AX                  ; 被乘数除以 8 的结果暂存到 BX
MOV     CL, 2                   ; 设置移位位数 2
SHR     AX, CL                  ; 被乘数除以 4 (累计除 32), 结果在 AX 中
SUB     BX, AX                  ; 被除数/8 - 被除数/32, 结果在 BX 中
MOV     CL, 2                   ; 设置移位位数 2
SHR     AX, CL                  ; 被乘数除以 4 (累计除 128), 结果在 AX 中
```

ADD	BX, AX	; 被除数/8 - 被除数/32 + 被除数/128, 结果在 BX 中
MOV	CL, 2	; 设置移位位数 2
SHR	AX, CL	; 被乘数除以 4 (累计除 512), 结果在 AX 中
SUB	BX, AX	; 被除数/8 - 被除数/32 + 被除数/128 - 被除数/512, 结果在 BX 中

4) 将 AX 中间 8 位、BX 低 4 位、DX 高 4 位拼成一个新字的程序段是:

AND	DX, 0F000H	; 将 DX 的低 12 位清零、高 4 位不变
AND	AX, 0FF0H	; 将 AX 的低 4 位清零、高 4 位清零、中间 8 位不变
AND	BX, 0FH	; 将 BX 的高 12 位清零、低 4 位不变
ADD	AX, BX	
ADD	AX, DX	; 按要求组成一个新字, 结果放在 AX 中

5) 将 BX 中的 4 位压缩 BCD 数用非压缩 BCD 数形式顺序放在 AL、BL、CL、DL 中的程序段是:

MOV	DL, BL	; 4 位压缩 BCD 数的低位字节送 DL
AND	DL, 0FH	; DL 的高 4 位清零, 得 4 位非压缩 BCD 数的最低位, 放入 DL 中
MOV	CL, 4	; 设置移位位数 4
SHR	BX, CL	
MOV	CH, BL	; 将 BL 的内容暂存到 CH 中保留
AND	CH, 0FH	; CH 的高 4 位清零, 得 4 位非压缩 BCD 数的次低位, 放 CH 中
MOV	CL, 4	; 设置移位位数 4
SHR	BX, CL	
MOV	AL, BL	; 将 BL 的内容暂存到 AL 中保留
AND	BL, 0FH	; BL 的高 4 位清零, 得 4 位非压缩 BCD 数的次高位, 放 BL 中
MOV	CL, 4	; 设置移位位数 4
SHR	AL, CL	
MOV	CL, CH	; 将 4 位非压缩 BCD 数的次低位移入 CL 中

6. 编写一个程序段, 将 AL 中的各位顺序完全颠倒后存入 AH, 如 AL = 11010010B, AH = 01001011B。

【解】 采用的算法为将 AL 中各位依次从低位移出, 同时将移出位从低位向高位移入 AH 中 (见图 3-2), 编程思路如下:

- 1) 设置循环计数器 CX←8, 令 AH←0。
- 2) 利用 SHR 指令将 AL 右移一次。
- 3) 利用 RCL 指令将 AH 带进位位循环左移一次。
- 4) 重复执行步骤 2) 和步骤 3) 共 8 次。
- 5) 最后保存到 AL 中。
- 6) 程序段如下:

	MOV	AH, 0
	MOV	CX, 8
LOP:	SHR	AL, 1
	RCL	AH, 1
	LOOP	LOP

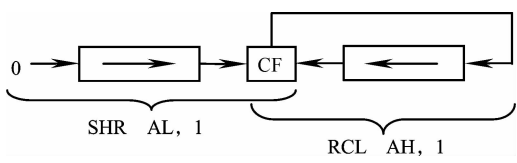


图 3-2 操作示意图



```
MOV    AL, AH
HTL
```

7. 若接口 03F8H 的第 1 位 (b1) 和第 3 位 (b3) 同时为 1, 表示接口 03FBH 有准备好的 8 位数据, 当 CPU 将数据取走后, b1 和 b3 就不再同时为 1 了。仅当又有数据准备好时才再同时为 1。试编写程序, 从上述接口读入 200 字节的数据, 并顺序放在 DATA 开始的地址中。

【解】 即当从输入接口 03F8H 读入的数据满足  $\times \times \times \times 1 \times 1 \times B$  时, 可以从接口 03FBH 输入数据。

```
LEA    SI, DATA
MOV    CX, 200
NEXT:  MOV    DX, 03F8H
WAIT:  IN     AL, DX
AND    AL, 0AH          ; 判断 b1 和 b3 位是否同时为 1
CMP    AL, 0AH
JNZ    WAIT             ; b1 和 b3 位同时为 1, 则读数据, 否则等待
MOV    DX, 03FBH
IN     AL, DX
MOV    [SI], AL
INC    SI
LOOP   NEXT
HLT
```

8. 编写一个将 STR 字符串加密的子程序, 设加密码已存在 AL 中, 字符串的长度在 CX 中。  
说明: 常利用“异或”操作具有两次操作后原数据不变的特点, 通过将数据与加密码进行“异或”操作实现加密, 在解密时与加密码再进行“异或”操作后, 即可得到原数据。

【解】 分析: 在子程序运行时, 不希望影响到主程序, 所以在进入子程序后, 常把子程序中用到的寄存器压栈保护起来, 在返回前依次恢复。注意压栈与出栈的顺序。

```
S_STR:  PUSHF
        PUSH    AX
        PUSH    SI
        LEA     SI, STR
NEXT:   MOV     AH, [SI]
        XOR     AH, AL
        MOV     [SI], AH
        INC     SI
        LOOP    NEXT
        POP     SI
        POP     AX
        POPF
        RET
```

9. 编写一段程序, 实现将 CH 寄存器的内容转换为非压缩型 BCD 码, 存入 AX 和 CH 寄存器

中，其中，百位数存放在 CL 中，十位数存放在 AH 中，个位数存放在 AL 中。

【解】 可以采用不同的方法实现。

方法 1：因为 CH 的内容为 00H ~ FFH，所以转换成十进制数为 0 ~ 255，采用除法除以 100 取商得百位数，再将余数除以 10 取商得十位数，最后的余数即为个位数。程序如下：

```
MOV     AL, CH      ; 取数存到 AL 中
XOR     AH, AH      ; 清 AH
MOV     BL, 100     ; 将除数 100 送 BL
DIV     BL           ; AX/BL, 商（百位数）在 AL 中，余数在 AH 中
MOV     CL, AL       ; 商（百位）送 CL
MOV     BL, 10      ; 除数 10 送 BL
MOV     AL, AH       ; 将余数 AH 送入 AL
XOR     AH, AH      ; 清 AH
DIV     BL           ; AX/BL, 商（十位数）在 AL 中，余数在 AH 中
XCHG    AH, AL       ; 商（十位）送 AH, 余数（个位）送 AL
HLT
```

方法 2：因为 CH 的内容为 X，其可以看成是 X 个 1 的累加值，所以可以采用非压缩型 BCD 码累加 1；因为 X 可能大于 99，所以需要做两次非压缩型 BCD 码累加。

```
LP1:    XOR     AX, AX      ; 清 AX
        ADD     AL, 1       ; 累加 1
        AAA                     ; 非压缩型 BCD 码调整
        DEC     CH         ; 累加次数减 1
        JNZ     LP1
        MOV     CL, AL      ; 暂存个位数据到 CL
        MOV     CH, AH
        XOR     AX, AX      ; 清 AX
LP2:    ADD     AL, 1       ; 累加 1
        AAA                     ; 非压缩型 BCD 码调整
        DEC     CH         ; 累加次数减 1
        JNZ     LP2
        XCHG    AH, AL      ; 将十位数据存入 AH
        XCHG    CL, AL      ; 百位数据存入 CL, 个位数据存入 AL
        HLT
```

方法 3：先利用压缩型 BCD 码累加 1 的方法，实现将 CH 内容转换成压缩型 BCD 码，存入 CL 和 AL 中，再将 AL 中的压缩型 BCD 码拆成非压缩型 BCD 码。

```
LP1:    XOR     AX, AX      ; 清 AX
        ADD     AL, 1       ; 累加 1
        DAA                     ; 压缩型 BCD 码调整
        ADC     AH, 0
        DEC     CH         ; 累加次数减 1
        JNZ     LP1
        MOV     CL, AH      ; 保存百位数据到 CL
        MOV     AH, AL
```

```

MOV     CL, 4
SHR     AH, CL      ; 获得十位数存入 AH
AND     AL, 0FH     ; 获得个位数存入 AL
HLT

```

10. 试编写程序，实现找出 M 字节的有符号补码数数组 DATA 中绝对值最大的值，并存入 MAX 单元。

【解】 正数的绝对值是其本身，负补码数的绝对值是连同符号求反后加 1。程序如下：

```

START:  LEA     BX, DATA
        MOV     AL, 0      ; 设最大绝对值单元 AL 初始值 = 0
        MOV     CX, M      ; 字节数送 CX
L1:      MOV     AH, [BX]   ; 从数组中取 1 字节
        TEST    AH, 80H    ; 测试符号位
        JNS     L2        ; 正数转到 L2
        NEG     AH         ; 负数求变补，即连同符号位求反加 1，得到其绝对值
L2:      CMP     AH, AL     ; 绝对值与最大绝对值单元 AL 比较
        JC      L3        ; AH < AL, 转到 L3
        MOV     AL, AH     ; AH ≥ AL, 则 AL ← AH
L3:      INC     BX        ; 指针增 1
        LOOP    L1        ; CX = CX - 1, CX ≠ 0 则转到 L1
        MOV     MAX, AL    ; 保存最大绝对值
        HLT

```

11. 编写计算  $AX^2 + BX + C$  类型多项式值的程序段。设系数 A, B, C 和变量 X 均为无符号字节数，计算结果存入 BUF，如图 3-3 所示。

【解】 注意， $X * X \leq 2$  字节， $A * X * X \leq 3$  字节，考虑极端情况  $AX^2 + BX + C \leq 3$  字节

```

LEA     SI, BUF1      ; SI 指向 BUF
MOV     AL, X          ; 取 X 值至 AL
XOR     AH, AH         ; AH 清 0
MUL     AL             ; AX = X^2
MOV     DL, A          ; 取系数 A
XOR     DH, DH         ; DH 清 0
MUL     DX             ; DX, AX = DX * AX = A * X^2
MOV     [SI], AX       ; 暂存 AX
MOV     [SI+2], DX     ; 暂存 DX
MOV     AL, X          ; AL = X
MOV     DL, B          ; DL = B
MUL     DL             ; AX = DL * AL = B * X
ADD     AL, C
ADC     AH, 0          ; AX = B * X + C
MOV     BX, [SI]       ; BX = A * X^2 低字
ADD     AX, BX          ; 将 A * X^2 低字 + B * X + C 存入 AX 中
MOV     [SI], AX       ; 保存

```

A	XX
B	XX
C	XX
X	XX

图 3-3 内存示意图

```

MOV     AX, [SI]    ; AX = A * X2 高字
ADC     AX, 0        ; AX = AX + 0 + CF
MOV     [SI+2], AX ; 保存
HLT

```

12. 请编写一程序段，实现从键盘输入 2 位十进制数，将其转换成 1 字节压缩 BCD 码存放在 BCD 单元中。

【解】 可以采用 INT 21H 中的 1 号功能，实现从键盘输入 1 个字符。

```

MOV     AH, 1
INT     21H          ; 从键盘输入 1 位 0~9 的字符，其对应的 ASCII 码存在 AL 中
AND     AL, 0FH      ; 将 AL 中的 ASCII 码转换成非压缩型 BCD 码
MOV     CL, 4
SAL     AL, CL       ; 非压缩型 BCD 码移至高 4 位
MOV     BL, AL       ; 保存到 BL 中
MOV     AH, 1
INT     21H          ; 再从键盘输入 1 位 0~9 的字符，其对应的 ASCII 码存在 AL 中
AND     AL, 0FH      ; 将 AL 中的 ASCII 码转换成非压缩型 BCD 码
ADD     AL, BL       ; 与暂存在 BL 中的高 4 位合成一个压缩型 BCD 码
MOV     BCD, AL      ; 将压缩型 BCD 码保存入 BCD 单元中
HLT

```

13. 编写一个程序段，实现将 BL 中的压缩型 BCD 码转换成 ASCII 码，并在屏幕上显示出来。

【解】 可以采用 INT 21H 中的 2 号功能，实现将 DL 中的 ASCII 码对应的字符显示在屏幕上。

```

MOV     DL, BL       ; 取压缩型 BCD 码
MOV     CL, 4        ; CL 赋值为 4
SHR     DL, CL       ; 将 DL 中的 BCD 码高 4 位移到低 4 位，同时高位清 0
ADD     DL, 30H      ; 转成 ASCII 码
MOV     AH, 02
INT     21H          ; 在屏幕上显示 BCD 码的高位
AND     BL, 0FH      ; BL 高 4 位清零
ADD     BL, 30H      ; BL 高 4 位位置为 3，转成 ASCII 码
MOV     DL, BL
MOV     AH, 02
INT     21H          ; 在屏幕上显示 BCD 码的低位
HLT

```

14. 请编写一程序段，实现从键盘输入 2 位十进制数，将其转换成 1 字节的十六进制数保存在 BUF 单元中。

【解】 从键盘输入 2 位十进制数 XY，转换成十六进制数应为  $X * 10 + Y$ 。

```

MOV     AH, 1
INT     21H          ; 从键盘输入 1 位 0~9 的字符，其对应的 ASCII 码存在 AL 中
AND     AL, 0FH      ; 将 AL 中的 ASCII 码转换成非压缩型 BCD 码
MOV     BH, AL       ; 保存到 BH 中

```

```

MOV    AH, 1
INT    21H      ; 再从键盘输入 1 位 0~9 的字符, 其对应的 ASCII 码存在 AL 中
AND    AL, 0FH  ; 将 AL 中的 ASCII 码转换成非压缩型 BCD 码
MOV    BL, AL   ; 保存到 BL 中
MOV    AL, 10
MUL    BH       ; AL = BH * 10
ADD    AL, BL   ; AL = BH * 10 + BL
MOV    BUF, AL  ; 将十六进制数保存入 BUF 单元中
HLT

```

15. 已知一个关于 0~9 的数字的 ASCII 码表首址是当前数据段 DS 的 0A80H 位置, 现要找出数字 5 的 ASCII 码的个数, 且将其个数存于 BL 中。设码表中第一个单元存放着码表的长度。试编写程序实现。

【解】

```

MOV    SI, 0A80H ; 设置 ASCII 码表指针
XOR    BL, BL    ; 5 的 ASCII 码的个数计数器清 0
MOV    CL, [SI]  ; 取表长度
INC    SI        ; 指针增 1
MOV    AL, 35H   ; AL = 5 的 ASCII 码 = 35H
LP0:   CMP    [SI], AL ; 表中数与 5 比较
      JNZ    LP1    ; 不是 5, 则转到 LP1
      INC    BL     ; 是 5, 则 5 的 ASCII 码的个数计数器增 1
LP1:   INC    SI    ; 指针增 1
      DEC    CL     ; 循环次数减 1
      JNZ    LP0
      HLT

```

16. 试用查表指令 (XLAT) 实现计算  $c = a^3 + b^3$ , 其中 a、b 为 0~5 的内存数据, 如图 3-4 所示。

【解】

```

MOV    AX, DATA ; 取段基值
MOV    DS, AX    ; DS 指向表所在段
LEA    BX, TAB   ; 取表的首地址偏移量
MOV    AL, a     ; 取 a
XLAT    TAB      ; AL = DS: [BX + AL]
MOV    CL, AL    ; 暂存 a^3
MOV    AL, b     ; 取 b
XLAT    TAB      ; 查表得到 AL = b^3
ADD    AL, CL    ; AL = a^3 + b^3
MOV    c, AL     ; 保存结果
HLT

```

DATA	:
TAB	0
	1
	8
	27
	64
	125
	:
a	XX
b	XX
c	XX
	:

17. 试编制程序段实现求  $|X + Y| * |X - Y|$ , 并将结果存入内存。

其中 X 和 Y 是内存数据段 DATA 中两个单字节无符号二进制数。

【解】  $|X + Y| \leq 2$  字节,  $|X - Y| = 1$  字节,  $|X + Y| * |X - Y| \leq 2$  字节, 内存示意图如图 3-5 所示。

图 3-4 内存示意图

```

MOV     AX, DATA
MOV     DS, AX
XOR     AX, AX      ; AX = 0
MOV     AL, X       ; AL = X
ADD     AL, Y
ADC     AH, 0       ; AX = X + Y
XOR     BH, BH      ; BH = 0
MOV     BL, X       ; BL = X
SUB     BL, Y       ; BL = X - Y
JNC     LP          ; 若 X ≥ Y, 则转 LP
MOV     BL, Y       ; BL = Y
SUB     BL, X       ; BL = Y - X
LP:     MUL     BX    ; AX = AX * BX = |X + Y| * |X - Y|
MOV     Z, AX
HLT

```

DATA	⋮
X	XX
Y	XX
Z	XX
	⋮

图 3-5 内存示意图

18. 请根据图 3-6 所示的流程图，编写程序段。

【解】

```

MOV     AX, DA1    ; 取数
OR      AX, AX     ; 测试符号
JS      MINUS      ; 为负，则转 MINUS
JZ      ZERO       ; 为 0，则转 ZERO
PLUS:   SAR     BX, 1 ; 为正，则 BX = BX/2
JMP     QQ
MINUS:  SAL     BX, 1 ; 为负，则 BX = BX * 2
JMP     QQ
ZERO:   XOR     BX, BX ; 为 0，则 BX = 0
QQ:     MOV     BUF, BX
HLT

```

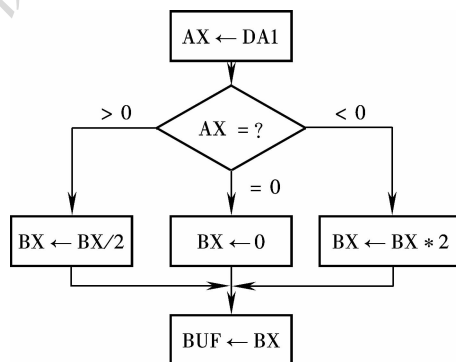


图 3-6 流程图 1

19. 请根据图 3-7 所示的流程图，编写程序段。

【解】

```

MOV     AL, DA1
CMP     AL, 09H
JBE     LP1

```

```

ADD    AL, 37H
JMP    LP2
LP1:   ADD    AL, 30H
LP2:   MOV    DL, AL
        MOV    AH, 2
        INT    21H

```

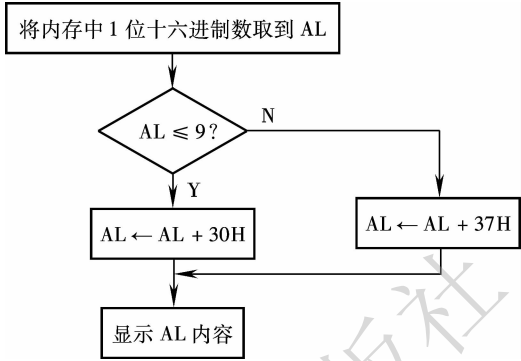


图 3-7 流程图 2

20. 请根据如图 3-8 所示的流程图，编写程序段。

【解】

```

START:  MOV    AL, DA1    ; 取数
        CMP    AL, 7FH    ; AL 与 7FH 比较
        JA     LP1        ; 若 AL > 7FH, 则转到 LP1
        CMP    AL, 3FH    ; AL 与 3FH 比较
        JB     LP2        ; 若 AL < 3FH, 则转到 LP2
        JMP    EXIT       ; 3FH ≤ AL ≤ 7FH
LP1:    NEG     AL         ; 求 AL 补码 (符号位不变, 其余位求反后加 1)
        OR     AL, 80H     ; 求 AL 补码 (符号位不变, 其余位求反后加 1)
        JMP    EXIT
LP2:    SAL     AL, 1       ; AL = AL * 2
EXIT:   HLT

```

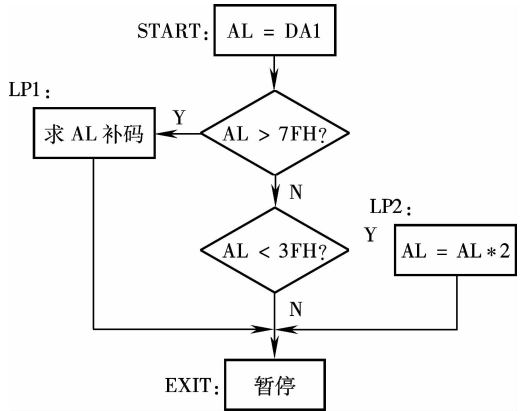


图 3-8 流程图 3

21. 请根据如图 3-9 所示的流程图，编写程序段。

【解】

```
MOV    AH, 1
INT     21H
CMP     AL, '0'
JB      THREE
CMP     AL, '9'
JNA     ONE
CMP     AL, 'Z'
JBE     TWO
```

ONE: ...

TWO: ...

THREE: ...

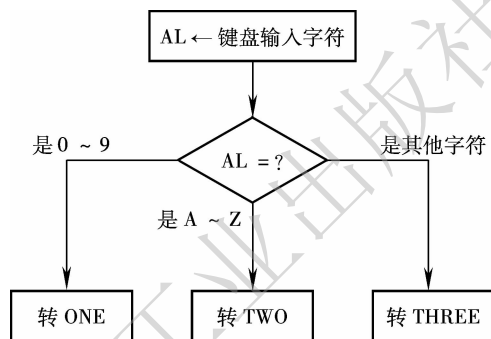


图 3-9 流程图 4