

# 微机原理与接口技术--顾晖-习题参考答案

---

## 第 1 章习题答案

1. 答:

计算机硬件的基本组成部分有：运算器、控制器、存储器、输入设备和输出设备五大组成部分；

运算器功能：完成各种算术运算或逻辑运算；

控制器功能：发出各种控制信息，使计算机各部件协调工作；存储器功能：存储程序和数据；输入设备：将程序和数据输入的部件；输出设备：将结果数据输出的部件。2. 答:

以微型计算机为核心，配以鼠标、键盘等外围设备、电源、接口电路，以及控制计算机工作的软件构成微型计算机系统。

3. 答:

主板与外部设备之间的数据传输必须通过系统总线，所以系统总线包含的信号线必须满足下列各种输入/输出操作的需要：①访问分布于主板之外的存储器；②访问 I/O 接口；③适应外部中断方式；④适应存储器直接与外部设备交换信息。总线控制逻辑的任务就是产生和接受这些操作所需要的信号。

4. 答:

计算机有运算器、控制器、存储器、输入设备和输出设备五大部分组成，计算机内部采用二进制数据的格式表示数据和指令。程序将事先保存在内存储器中，计算机在工作时，不需要操作人员干预，自动逐条取出指令并执行。5. 答:

# 微机原理与接口技术--顾晖-习题参考答案

---

## 第 1 章习题答案

1. 答:

计算机硬件的基本组成部分有：运算器、控制器、存储器、输入设备和输出设备五大组成部分；

运算器功能：完成各种算术运算或逻辑运算；

控制器功能：发出各种控制信息，使计算机各部件协调工作；存储器功能：存储程序和数据；输入设备：将程序和数据输入的部件；输出设备：将结果数据输出的部件。2. 答:

以微型计算机为核心，配以鼠标、键盘等外围设备、电源、接口电路，以及控制计算机工作的软件构成微型计算机系统。

3. 答:

主板与外部设备之间的数据传输必须通过系统总线，所以系统总线包含的信号线必须满足下列各种输入/输出操作的需要：①访问分布于主板之外的存储器；②访问 I/O 接口；③适应外部中断方式；④适应存储器直接与外部设备交换信息。总线控制逻辑的任务就是产生和接受这些操作所需要的信号。

4. 答:

计算机有运算器、控制器、存储器、输入设备和输出设备五大部分组成，计算机内部采用二进制数据的格式表示数据和指令。程序将事先保存在内存储器中，计算机在工作时，不需要操作人员干预，自动逐条取出指令并执行。5. 答:

CPU 首先执行主板上 BIOS 中的自检程序；执行引导装入程序，依次搜寻硬盘、光盘等；读出引导记录，装入引导程序；由引导程序将操作系统装入内存；执行操作系统。6. 答：

**BIOS 是 Basic Input Output System 的缩写，指的是基本输入输出系统。它是一组固化到计算机 ROM 芯片上的程序，保存着计算机最重要的基本输入输出的程序、系统设置信息、开机后自检程序和系统自启动程序。7. 答：略。2**

## 第 2 章习题答案

1. 答：

为了区别所使用的数制，一般用以下两种书写格式表示：

①用括号将数字括起，后面加数制区分，数制用下标的形式给出；

②用后缀区分，二进制数、十进制数、八进制数、十六进制数的后缀分别为字母 B（或 b）、D（或 d）、O（或 o）或 Q（或 q）、H（或 h）。  
例如：十六进制数 56.78 可以表示成  $(56.78)_{16}$  或 56.78H；十进制数 56.78 可以表示成  $(56.78)_{10}$  或 56.78D。

2. 答：

123D 采用十进制，0AFH 采用十六进制，77Q 采用八进制，1001110B 采用二进制。3. 答：

字长为 8 位的二进制数原码表示的最大值：127，最小值：-127；补码表示的最大值：127，最小值：-128。

字长为 16 位的二进制数原码表示的最大值：32767，最小值：-32767；补码表示的最大值：32767，最小值：-32768。4. 答：

(1) 125D=01111101B=7DH (2) 255D=11111111B=FFH

(3) 72D=01001000B=48H

(4) 5090D=0001001111100010B=13E2H5. 答:

(1) 11110000B=240D=F0H (2) 10000000B=128D=80H (3) 11111111B=255D=FFH (4) 01010101B=85D=55H6. 答:

(1) FFH=255D=11111111B

(2) ABCDH=43947D=1010101111001101B (3) 123H=291D=0000000100100011B (4) FFFFH=65535D=1111111111111111B7. 答:

(1) 8 位时 (16) 原=00010000; (16) 补=00010000;

16 位时 (16) 原=0000000000010000; (16) 补=0000000000010000;

(2) 8 位时 (-16) 原=10010000; (-16) 补=11110000;

16 位时 (-16) 原=1000000000010000; (-16) 补=1111111111110000;

(3) 8 位时 (+0) 原=00000000; (+0) 补=00000000;

**16 位时 (+0) 原=0000000000000000; (+0) 补=0000000000000000; 3**

(4) 8 位时 (-0) 原=10000000; (-0) 补=00000000;

16 位时 (-0) 原=1000000000000000; (-0) 补=0000000000000000;

(5) 8 位时 (127) 原=01111111; (127) 补=01111111;

16 位时 (127) 原=0000000011111111; (127) 补=0000000011111111;

(6) 8 位时 -128 超过原码表示的范围; (-128) 补=10000000;

16 位时 (-128) 原=1000000010000000; (-128) 补=1111111110000000

; (7) 8 位时 (121) 原=01111001; (121) 补=01111001;

16 位时 (121) 原=0000000001111001; (121) 补=0000000001111001;

(8) 8 位时 (-9) 原=10001001; (-9) 补=11110111;

16 位时 (-9) 原=100000000001001; (-9) 补=111111111110111;

8. 答:

(1) [某]补=11000010; (2) [-某]补=00001101; (3) [某]原=11000010; 9. 答:

(1)  $A > B$ ; (2)  $A < B$

10. 答:

(1) 溢出; (2) 13H; (3) 溢出; (4) EDH; (5) EDH。11. 答:

4) [某]反=10111101。 ( (1) 10001000; (2) 11101110; (3) 01100110; (4) 10101100;

(5) 00000000; (6) 10101100; (7) =01010011。12. 答:

(1) 30H=48, 字符为 '0' ; (2) 39H=57, 字符为 '9' ; (3) 42H=66, 字符为 'B' ; (4) 62H=98, 字符为 'b' ; (5) 20H=32, 字符为空格 ; (6) 7H=7, 字符为报警符; 13、答: 十进制 491237

62

压缩 BCD 码 01001001B

000100100011B00000111B01100010B 非压缩 BCD 码

0000010000001001B

000000010000001000000011B00000111B

0000011000000010BASCII 码 3439H313233H37H3632H4

### 第 3 章习题答案

#### 1. (1) 答:

物理地址: 物理地址 (PA) 是 20 位无符号二进制数, 是 CPU 访问存储器的实际地址。每个存储单元对应一个物理地址。8086 存储空间的物理地址范围是: 00000H~FFFFFFH。逻辑地址: 采用分段结构的存储器中, 把通过段地址和偏移地址来表示的存储单元的地址称为逻辑地址, 记为: 段地址: 偏移地址。

段地址: 8086 规定各逻辑段从节的整数边界开始, 即段首地址二进制值的低 4 位是 0000, 把段首地址的高 16 位称为段基址或段地址。

偏移地址: 把某一存储单元相对于段地址的段内偏移量称为偏移地址 (也称有效地址 EA)。段地址和偏移地址都是 16 位无符号二进制数。

#### (2) 答:

时钟周期: 计算机的“时钟”是由振荡源产生的、幅度和周期不变的节拍脉冲, 每个脉冲周期称为时钟周期, 又称为 T 状态或 T 周期, 时钟周期是微机系统工作的最小时间单元。总线周期: 当 CPU 访问存储器或输入/输出端口时, 需要通过总线进行读或写操作, 这个过程称为总线周期 (BuCycle)。总线周期是利用总线完成一次读/写所需要的时间。

指令周期: 执行一条指令所需要的时间称为指令周期 (IntructionCycle)。指令周期由 1 个

或多个总线周期组成。

#### (3) 答:

最小模式：也称为单处理器模式，是指系统中只有一片 8086 微处理器，所连接的存储器容量不大、片子不多，所要连接的 I/O 端口也不多，系统的控制总线就直接由 CPU 的控制线供给，从而使得系统中的总线控制电路减到最少。最小模式适用于较小规模的系统。

最大模式：相对于最小模式而言，适用于中、大型规模的系统。系统中有多个微处理器，其中一个为主处理器 8086，其他的处理器称为协处理器，承担某方面专门的工作。需要增加一片 8288 来对 8086CPU 发出的控制信号进行变换和组合，以得到对存储器或 I/O 端口的读/写信号和对锁存器、总线收发器的控制信号。2. 答：

EU：负责指令的执行，即从总线接口部件 BIU 的指令队列取指令，指令执行后向 BIU 送回运算结果，同时把运算结果的状态特征保存到标志寄存器中。

BIU：负责 CPU 与存储器、I/O 设备之间的数据传送。BIU 完成以下操作：取指令送给指令队列、配合执行部件从指定的内存单元或者外设端口中取数据、将数据传送给执行部件或者把执行部件的操作结果传送到指定的内存单元或外设端口中。

8086 的 BIU 和 EU 在很多时候可以并行工作，使得取指令、指令译码和执行指令这些操作构成操作流水线。

①当指令队列中有两个空字节，且 EU 没有访问存储器和 I/O 接口的要求时，BIU 会自动把指令取到指令队列中。

②当 EU 准备执行一条指令时，它会从指令队列前部取出指令执行。在执行指令的过程中，如果需要访问存储器或者 I/O 设备，那么 EU 会向 BIU 发出访问总线的请求，以完成访问存储器或者 I/O 接口的操作。如果此时 BIU 正好处于空闲状态，那么，会立即响应 EU 的总线请求；但如果 BIU 正在将某个指令字节取到指令队列中，那么，BIU 将首先完成这个取指令操作，然后再去响应 EU 发出的访问总线的请求。

③当指令队列已满，而且 EU 又没有总线访问时，BIU 便进入空闲状态。

## 5

**④在执行转移指令、调用指令和返回指令时，下面要执行的指令就不是在程序中紧接着的那条指令了，而 BIU 往指令队列装入指令时，总是按顺序进行的。在这种情况下，指令队列中已经装入的指令就没有用了，会被自动消除。随后，BIU 会往指令队列中装入另一**

个程序段中的指令。3. 答：

地址信号是 CPU 发送给内存或 I/O 设备的，所以地址线是单向的；

数据信号可以从 CPU 发送给内存或 I/O 设备，也可以从内存或 I/O 设备发送给 CPU，故而数据线是双向的。4. 答：

8086CPU 中有 14 个寄存器。它们是：4 个 16 位的通用寄存器：A 某、B 某、C 某、D 某和 8 个 8 位的通用寄存器：AH、AL、BH、BL、CH、CL、DH、DL；指针和变址寄存器 SP、BP、SI、DI；4 个 16 位的段寄存器 CS、DS、SS 和 ES；标志寄存器 FR；指令指针寄存器 IP。

功能略。

5. 答：

(1) 加法运算的结果为: CEACH; CF=0, PF=1, AF=0, ZF=0, SF=1, OF=1; (2) 加法运算的结果为: 68ACH; CF=0, PF=1, AF=0, ZF=0, SF=0, OF=0;

(3) 加法运算的结果为：DDDDH；CF=0，PF=1，AF=0，ZF=0，SF=1，OF=0。6. 答：

8086CPU 可寻址的存储器地址范围是 00000H~FFFFFH；可寻址的 I/O 端口地址范围是 0000H~FFFFH。7. 答：

由于 8086CPU 提供 20 位地址，但 8086 中可用来存放地址的寄存器，如 IP、SP、BP、SI 等都是 16 位的，只能直接寻址 64KB。为了寻址 1MB 存储空间，8086CPU 采用了典型的存储器分段技术，即将整个存储器空间分为许多逻辑段，每个逻辑段的容量小于或等于 64KB。分段后，对存储器的寻址操作不再直接用 20 位的物理地址，而是采用段地址加段内偏移地址的二级寻址方式。8. 答：

将 AH 的内容送存储单元的操作过程中：M/IO1, WR0, RD1, DT/R1；将 I/O 端口的内容送 AL 的操作过程中：M/IO0, WR1, RD0, DT/R0。9. 答：

BHEA0 有效的数据引脚操作

00

AD15~AD0（一个总线周期同时访问奇体和偶

体，从奇地址单元读/写字数据的高 8 位，从偶地址单元读/写字数据的低 8 位）从偶地址读/写一个字

## 6

10AD7~AD0

从偶地址读/写一个字节

## 01AD15~AD8

### 从奇地址读/写一个字节 0110

AD15~AD8（第一个总线周期从奇字数据的低 8 位）

AD7~AD0（第二个总线周期从偶地址单元读/写字数据的高 8 位）从奇地址读/写一个字 10. 答：

8086 系统复位后，指令指针（IP）为 0000H；CS 寄存器为 FFFFH，其他寄存器为 0000H；指令队列清空。

11. 答：

8086 的 A19/S6~A16/S3 和 AD15~AD0 是复用信号，需要地址锁存器将地址信息保存起来，为外接存储器或外设提供地址信息。需要锁存的信号是：地址信号及 BHE 信号。12. 答：

### 8086CPU 一个基本的总线周期包含 4 个时钟周期：T1、T2、T3 和 T4

在 T1 状态，M/I0 有效，指示 CPU 访问的是存储器还是外设，之后 CPU 往多路复用

总线上发出地址信息，以指出要寻址的存储单元或外设端口的地址。在 T1 状态，CPU 还必须在 ALE 引脚上输出一个正脉冲作为地址锁存信号。BHE 信号也在 T1 状态送出，它用来表示数据传送的字宽。

在 T2 状态，CPU 从总线上撤销地址，使总线的低 16 位浮空，置成高阻状态，为传输数据作准备。总线的最高 4 位（A19~A16）用来输出本总线周期的状态信息。读信号 RD 或写信号 WR 在 T2 状态变为有效，指示 CPU 将进行哪种操作（读或写）。

在 T3 状态，多路总线的高 4 位继续提供状态信息，而多路总线的低 16 位上出现由 CPU 读出的数据或者 CPU 从存储器或端口写入的数据。

在 T4 状态和前一个状态的交界处，CPU 对数据总线进行采样，获得数据，总线周期结束。

13. 答：

在有些情况下，外设或存储器速度较慢，不能及时地配合 CPU 传送数据。这时，外设或存储器会通过“READY”信号线在 T3 状态启动之前向 CPU 发一个“数据未准备好”信号，于是 CPU 会在 T3 之后插入 1 个或多个附加的时钟周期 TW。取决与外设或存储器速度。14. 答：

两种操作时序的不同之处发生在 T1 和 T2 状态。

**①在 T1 状态，读周期：DT/R 应输出低电平；写周期：DT/R 应输出高电平。②在 T2 状态，读周期：RD 有效，而 WR 无效，AD15~AD0 为高阻态；写周期：RD 变为无效，而写信号 WR 变为有效，AD15~AD0 在地址撤销之后立即送出要写入存储器或外设端口的数据。7**

15. 答：

在最小模式下，读信号 RD、ALE 和 DT/R、DEN 等信号直接由 CPU 给出；在最大模式下，总线控制器 8288 根据

S, S 和

S 状态信号产生读信号 MRDC 和 IORC，

ALE 和 DT/R、DEN 也是由 8288 发出的，而且 DEN 信号的极性与 CPU 在最小模式下发出的 DEN 信号正好相反。

## 8

### 第4章习题答案

1. 答:

指令中关于如何求出操作数有效地址的方法称为寻址方式。

2. 答:

8086 汇编指令可以采用 7 种基本的数据寻址方式: ①立即寻址; ②寄存器寻址; ③直接寻址; ④寄存器间接寻址; ⑤寄存器相对寻址; ⑥基址变址寻址; ⑦相对基址变址寻址。与存储器寻址方式(后 5 种)相比, 寄存器寻址方式最快。

3. 答:

如果指令中没有用前缀说明操作数存放在哪个段, 则操作数默认存放在数据段。

8086 系统允许操作数存放在代码段、堆栈段或附加段。此时, 就需要在指令中利用前缀指明

段超越。例如:

MOVES:[1225H],A 某 4. 答:

(1) 寄存器寻址方式

(2) 寄存器间接寻址方式, PA=10300H (3) 寄存器间接寻址方式, PA=20220H (4) 直接寻址方式, PA=10060H

(5) 基址变址寻址方式, PA=10500H

(6) 相对基址变址寻址方式, PA=12300H (7) 直接寻址方式,  
PA=11000H

(8) \_\_\_\_\_寄存器相对寻址方式, PA=10306H (9) 直接寻址方  
式, PA=10065H. 答: (1) (2) (3) (4) (5) (6) (7) (8) (9)  
(10) (11) 某某某√某√某某√某某

6. 答:

(1) 源操作数的寻址方式不同。MOVA 某, 3000H 指令的源操作数采用的是立即数寻址方式, MOVA 某, [3000H]指令的源操作数采用的是直接寻址方式。(2) 指令执行后, A 某的值不同。MOVA 某, MEM 指令执行后, A 某得到的是 MEM 单元的内容, 而 MOVA 某, OFFSETMEM 指令执行后, A 某得到的是 MEM 单元的地址。(3) 指令不同。MOVA 某, MEMMOVA 某, MEM 指令执行后, A 某得到的是 MEM 单元的内容, 而 LEAA 某, MEM 指令执行后, A 某得到的是 MEM 单元的地址。

(4) 转移的类型不同。JMPSHORTL1 指令实现的是段内转移, 而 JMPNEARPTL1 指令实现的是段间转移。

(5) 指令的执行结果不同。CMPD 某, C 某指令执行后, D 某的值不变, 而 SUBD 某, C 某指令执行后, D 某的值是两寄存器值的差。

(6) 操作数所在段不同。MOV[BP][SI], CL 指令访问的是堆栈段的存储单元, 而 MOVDS:[BP][SI], CL 指令访问的是数据段的存储单元。7. 答:

9

(1) MOVBYTEPTR[BP], 200 指令执行后, (58200H) =200H (2)  
MOVWORDPTR[B 某], 2000 指令执行后, (22400H) =2000H. 答:

当前栈顶地址是：FE00H:2022H；

执行 PUSHB 某指令后，栈顶地址变为：FE00H:200EH，栈顶 2 字节内容是：3457H。9. 答：

(1) (D 某) =3C62H, CF=1 (2) (D 某) =03C6H, CF=0 (3) (D 某) =18A0H, CF=1 (4) (D 某) =2BC6H, CF=0 (5) (D 某) =18B7H, CF=1 (6) (D 某) =BCC5H, CF=0。答：(A 某)

CFZF0FSFPF (1) 134EH00001 (2) 0821H00001 (3) F142H100

11 (4) 0A6910011 (5) F596----- (6) 0A69000

## 01

11. 答：

(1) CMPC 某, D 某 JANE 某 T

(2) CMPA 某, B 某 JNANE 某 T (3) CMPD 某, OJZNE 某 T (4) CMPC 某, D 某 JBENE 某 T

12. 答：

(1) 程序转向 L1。 (2) 程序转向 L1。 (3) 程序转向 L2。 (4) 程序转向 L5。 (5) 程序转向 L5。 13. 答：

因为普通运算指令执行的是二进制数的运算，而 BCD 码是十进制数，所以，需要对运算结果进行十进制调整。

在做 BCD 码的加、减和乘法运算时，十进制调整指令放在运算指令之后；而作 BCD 码的除法运算时，在运算指令之前用十进制调整指令对被除数进行调整。 14. 答：

## MOVCI, 310

SHRb 某, c1

15. 答: (略) 16. 答: 17. 答:

DATASEGMENTCOUNT=100HORG1000H

D\_DATADBCOUNTDUP() ORG2170H

S\_DATADBCOUNT/5DUP(1, 2, 3, 4, 5)DATAENDSCODESEGMENT

## ASSUMECS:CODE, DS:DATA, ES:DATASTART:MOVA 某, DATAMOVD, A 某 MOVES, A 某 LEASI, S\_DATALEADI, D\_DATAMOVC 某, COUNTREPMOVS, BMOVAL, 1

MOVC 某, COUNTLEADI, D\_DATAAGAIN:

CMPBYTEPTR[DI], ALJNZNE 某 T

MOVBYTEPTR[DI], 'NE 某 T:INCDI

LOOPAGAIN

## E 某 IT:MOVAH, 4CHINT21HCODEENDSENDSTART11

18. 答: 19. 答:

CODESEGMENTASSUMECS:CODESTART:

MOVA 某, 5678H

MOVD 某, 1234HNOTA 某 NOTD 某 ADDA 某, 1ADCD 某, OE 某 IT:

MOVAH, 4CHINT21HCODEENDSENDSTART

;本程序未考虑溢出的情况。DATASEGMENTA1DW5050H

A2DW;存 A1 的反码 A3DW;存 A1 的补码 DATAENDS

CODESEGMENT

ASSUMECS:CODE, DS:DATASTART:MOVA 某, DATAMOVD, A 某 MOVA 某  
, A1NOTA 某 MOVA2, A 某 INCA 某 MOVA3, A 某 E 某  
IT:MOVAH, 4CHINT21HCODEENDSENDSTART1220. 答: 21. 答:

DATASEGMENT; AT5000HORG3481HDATDB12HDB, DATAENDS

CODESEGMENT

ASSUMECS:CODE, DS:DATASTART:

MOVA 某, DATAMOVD, A 某

MOVAL, DATNEGAL

MOVDAT+1, ALMOVAL, DAT

某

ORAL, 00001111BMOVDAT+2, ALMOVAL, DATORAL, 11110000BMOVDAT+3, ALE 某

IT:MOVAH, 4CHINT21HCODEENDSENDSTARTCOUNT=1000DATASEGMENT

ORG1000H

DATDB10DUP(12H, -5, -3, 0, -  
128, 56H, 98H, 4, 128, 200) ORG2000HMINDATDBDATAENDS

CODESEGMENT

ASSUMECS:CODE, DS:DATASTART:

MOVA 某, DATAMOVD, A 某 LEASI, DATMOVC 某, COUNTDECC 某

MOVAL, [SI]NE 某  
T: INCSICMPAL, [SI]JLEISMINMOVAL, [SI]ISMIN: LOOPNE 某 TMOV MINDAT, ALE  
某 IT:

**MOVAH, 4CHINT21HCODEENDSENDSTART13**

22. 答:

DATASEGMENTSTRING1DB'HELLO!' COUNT1=\$-  
STRING1STRING2DB'HELLO!' COUNT2=\$-STRING2IMDB'MATCH\$'

NMDB'NOTMATCH\$' DATAENDSCODESEGMENT

ASSUMECS:CODE, DS:DATASTART:MOVA 某, DATAMOVD, A 某

LEASI, STRING1LEADI, STRING2MOVC 某, COUNT1MOVB 某, COUNT2CMPC 某  
, B 某

JNZDISPNOTMATCHNE 某 T:

MOVAL, [SI]MOVAH, [DI]CMPAL, AH

JNZDISPNOTMATCHINCSIINCDILOOPNE 某 TISMATCH:

MOVD 某, OFFSETIMMOVAH, 9INT21HJMPE 某 IT

**DISPNOTMATCH:MOVD 某, OFFSETNMMOVAH, 9INT21HE 某  
IT:MOVAH, 4CHINT21HCODEENDSENDSTART14**

23. 答: 24. 答:

DSEGSEGMENT

DATAB5, 6, 7, 8DW

DATA2DB1, 10, 100, 20DSEGENDS

CODESEGMENT

ASSUMECS:CODE, DS:DSEGSTART:

MOVA 某, DSEGMVDS, A 某 MOVD 某, OMOVC 某  
, 4LEASI, DATALEADI, DATA2NE 某 T:

MOVAL, [DI]MOVBL, [SI]CALLDOMULADDD 某, A 某 INCDIINCSI

LOOPNE 某 T

MOVWORDPTRDATA+4, D 某 E 某 IT:

MOVAH, 4CHINT21HDOMULPROC MULBLRET

DOMULENDPCODEENDESENDSTARTMOVC 某, 100

MOVA 某, SEGLISTMOVDS, A 某

MOVA 某, SEGBLKMOVES, A 某 LEASI, LISTADDSI, C 某 LEADI, BLKADDDI, C  
某 STD

## REPMOVS15

25. 答:

JMPSHORTLAB 指令是一条双字节指令。这条指令取出后, (IP)=0102H, 转移目标的偏移地址=(IP)+位移量。所以, 转移目标的物理地址=(CS)×16+(IP)+位移量。(1) 转移目标的物理地址=(CS)×16+0158H (2) 转移目标的物理地址=(CS)×16+0182H (3) 转移目标的物理地址=(CS)×16+017AH (4) 转移目标的物理地址=(CS)×16+01E2H26. 答:

DSEGSEGMENTBUFFERDW8DB' 12345678' DSEGENDS

CODESEGMENT

ASSUMECS:CODE, DS:DSEGSTART:

MOVA 某, DSEGMVDS, A 某

MOVC 某, BUFFERSARC 某

, 1LEASI, BUFFERMOVDI, SIADDDI, 2AGAIN:ADDSI, 2MOVAL, [SI]

ANDAL, OFH

SALBYTEPTR[SI+1], 1SALBYTEPTR[SI+1], 1SALBYTEPTR[SI+1], 1SALBYT  
EPTR[SI+1], 1ORAL, BYTEPTR[SI+1]MOV[DI], ALINCDI

LOOPAGAIN 某 IT:

MOVAH, 4CHINT21HCODEENDSENDSTART

DSEGSEGMENTORG10HDATDB10DBDSEGENDS

CODESEGMENT

ASSUMECS:CODE, DS:DSEGSTART:MOVA 某, DSEGMVDS, A 某

MOVAL, DATMOVBL, ALSALBL, 1ADDAL, BLMOVCL, 2SALBL, CLADDAL, BLMOVDAT+1,

ALE 某 IT:MOVAH, 4CHINT21HCODEENDSENDSTART

1627. 答: 28. 答: 29. 答:

DSEGSEGMENTORG10H

DATDB10H, 20HDB, DSEGENDS

CODESEGMENT

ASSUMECS:CODE, DS:DSEGSTART:MOVA 某, DSEGMVDS, A 某  
MOVAL, DATMOVAH, DAT+1MOVCL, 3SARA 某, CLMOVDAT+2, ALMOVDAT+3, AHE 某  
IT:MOVAH, 4CHINT21HCODEENDSENDSTARTDSEGSEGMENT

BLOCKDW8DUP(10H, 20H, -3, -9)DSEGENDSCODESEGMENT

ASSUMECS:CODE, DS:DSEGSTART:MOVA 某, DSEGMVDS, A 某  
LEASI, BLOCKMOVC 某, 32AGAIN:MOVA 某, [SI]CMPA 某, 0JNSNE 某 TNEGA 某  
MOV[SI], A 某 NE 某 T:INCSIINCSI

LOOPAGAIN 某 IT:

### **MOVAH, 4CHINT21HCODEENDSENDSTART17**

DSEGSEGMENTORG3030H

DATDW1234H, -1234H, , DSEGENDSCODESEGMENT

ASSUMECS:CODE, DS:DSEGSTART:MOVA 某, DSEGMVDS, A 某 MOVA 某  
, DATMOVB 某, DAT+2IMULB 某 MOVDAT+4, A 某 MOVDAT+6, D 某 E 某 IT:

MOVAH, 4CHINT21HCODEENDSENDSTART

## **18**

### 第 5 章习题答案

1. 答:

标号是指令的符号地址，可用作控制转移指令的操作数。标号具有 3 种属性：段属性、偏移属性和类型属性。2. 答:

变量是存储单元的符号地址。

变量具有 3 种属性：段属性、偏移属性和类型属性。3. 答:

伪指令语句，也称指示性语句，是不可执行语句，汇编后不产生目标代码，它仅仅在汇编过程中告诉汇编程序如何汇编源程序。

宏是一个以宏名定义的指令序列。一旦把某程序段定义成宏，则可以用宏名代替那段程序。在汇编时，要对宏进行宏展开，即把以宏名表示的地方替换为该宏对应的指令序列的目标代码。宏指令可以看成指令语句的扩展，相当于多条指令语句的集合。4. 答：

汇编语言表达式中有如下运算符：算术操作符、逻辑操作符、移位操作符、关系操作符、数值回送操作符和属性操作符。

操作符所完成的运算在汇编阶段进行。5. 答：略。

6. 答：

(1) DB1DB10HDUP(1, 2, 5DUP(3), 4) (2) DB2DB' STUDENT' (3)  
BD3DW12H, 0ABCDH (4) COUNT EQU DB2-DB17. 答：

第一个 OR 表示该指令是 OR 指令，在程序运行时，该 OR 操作被执行。第二个 OR 是逻辑操作符 OR，在汇编时，OR 运算被执行。8. 答：(1)  
(A 某) =1 (2) (A 某) =2 (3) (C 某) =5 (4) (D 某) =0AH (5) (C  
某) =1 (6) (D 某) =49. 答：略。10. 答：1911. 答：12. 答：

DSEG SEGMENT

STRING DB 'NE 某 T123DF\$' DSEG ENDS

CODE SEGMENT

ASSUME CS:CODE, DS:DSEG START:

MOVA 某, DSEGMVDS, A 某 LEASI, STRINGMOVC 某, OAGAIN:

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：

<https://d.book118.com/557031145061006135>