

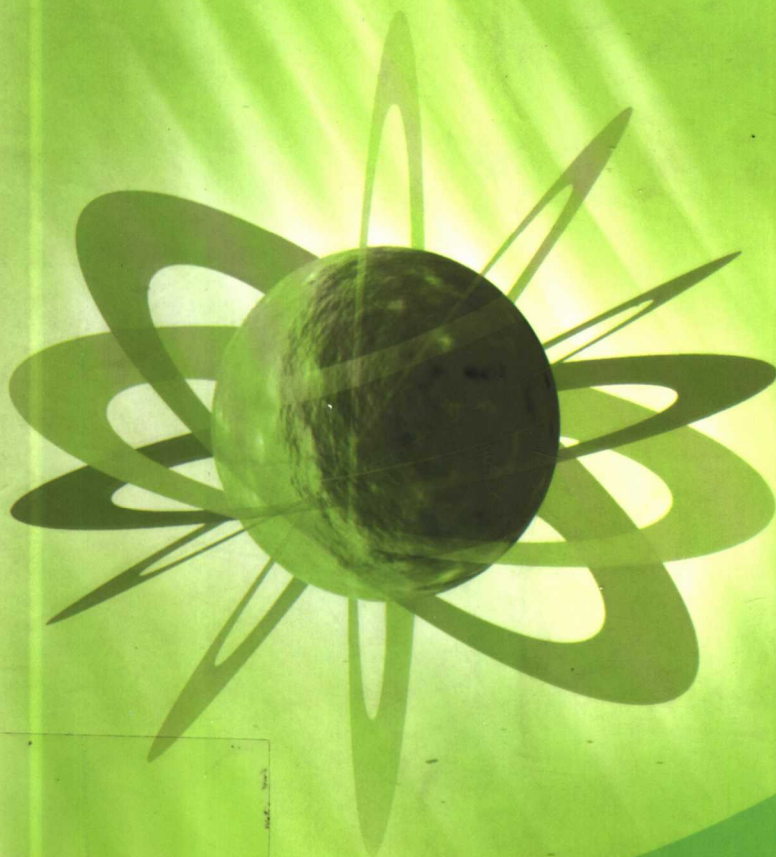


21世纪普通高等教育规划教材

微型计算机原理及应用

——习题解答、实验指导及课程设计

许立梓 主 编
何小敏 副主编



机械工业出版社
CHINA MACHINE PRESS

86-42
86-42
4 4

21 世纪普通高等教育规划教材

微型计算机原理及应用

——习题解答、实验指导及课程设计

主 编 许立梓

副主编 何小敏

主 审 黄元梅

机 械 工 业 出 版 社

本书是与机械工业出版社出版的由许立梓等编写的《微型计算机原理及应用》一书配套使用的辅助教材,由习题解答、实验指导和课程设计三部分组成。习题解答部分是《微型计算机原理及应用》教材中九章习题的全面解答,为教师和学生提供一种习题解答的参考方案;实验指导部分是微机原理及应用课程的实验指导书,分基本实验及自主实验两部分,共8个实验需16个学时;实验设备为西安唐都科教仪器公司开发的TDN(86/88)系列教学实验系统。课程设计部分是微机原理及应用课程的课程设计指导书,课程设计包括理论设计及综合实验两部分,可供1~2周设计使用。

本书与《微型计算机原理及应用》一书结合,为微机原理及应用课程的面授教学、作业、实验和课程设计等教学必需环节提供了教科书及学习指导书。本书可作为教师及电类非计算机专业学生学习的参考书。

图书在版编目(CIP)数据

微型计算机原理及应用:习题解答、实验指导及课程设计/许立梓主编. —北京:机械工业出版社,2004.4

21世纪普通高等教育规划教材

ISBN 7-111-14123-7

I. 微... II. 许... III. 微型计算机—高等学校—教学参考资料
IV. TP36

中国版本图书馆CIP数据核字(2004)第017252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:贡克勤 版式设计:霍永明 责任校对:李秋荣

封面设计:张静 责任印制:李妍

北京机工印刷厂印刷·新华书店北京发行所发行

2004年4月第1版·第1次印刷

787mm×1092mm $\frac{1}{16}$ ·9.25印张·223千字

定价:14.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

本社购书热线电话(010)68993821、88379646

封面无防伪标均为盗版

前 言

应广大读者要求，为配合机械工业出版社出版的《微型计算机原理及应用》一书，我们编写了这本习题解答、实验及课程设计指导。其中习题解答部分是《微型计算机原理及应用》一书中各章后习题的参考答案，目的是帮助在校学生和自学者学习时进行自我检查，拓宽思路、掌握解题方法，从而加深理解灵活运用。

本书第2部分为实验指导书，其中包括8个实验，基本涵盖了《微型计算机原理及应用》一书的主要内容。每个实验基本分两类，一类为基本实验，即验证实验，学生按实验指导做实验，从而了解一个实验的基本过程；另一类为自主实验（编程练习），实验指导书只给出题目，学生必须自主完成，实验教师只起监督作用，这部分实验的目的是给学生一定的空间，培养学生独立工作能力。

本书的第3部分为课程设计指导书，分理论设计和实践两部分。理论设计要求学生设计一个基于8088CPU的简单微机应用系统，目的是使学生建立一个完整的系统概念，使分散在各章的概念来一个综合，当然也掌握了一个简单微机系统的设计方法；实践部分是通过一个较大程序的编制，着重提高学生汇编语言的程序设计、调试能力。我们通过近十年的实践证明，课程设计对学生微机知识的掌握和微机应用能力的提高是十分重要的。

本书由许立梓、何小敏、李传芳、陈玮、高明琴编写，全书由许立梓任主编，并统编全稿。华南师范大学黄元梅副教授主审本书，并对本书提供了不少宝贵建议在此表示感谢。

由于编者水平和经验所限，书中不足和错误之处在所难免，恳请广大读者批评指正。

编 者

目 录

前言

第 1 部分 习题解答	1
(一) 习题一解答	1
(二) 习题二解答	3
(三) 习题三解答	9
(四) 习题四解答	17
(五) 习题五解答	25
(六) 习题六解答	30
(七) 习题七解答	34
(八) 习题八解答	39
(九) 习题九解答	54
第 2 部分 实验指导书	59
实验一 操作训练及数码转换编程实验	59
实验二 运算类编程实验	74
实验三 分支、循环及子程序设计实验	80
实验四 显示程序设计实验	86
实验五 中断特性及 8259 应用编程实验	87
实验六 8255 并行接口应用实验	95
实验七 8253 定时/计数器应用实验	99
实验八 存储器扩展实验	105
第 3 部分 微机原理及应用课程设计指导书	111
一、理论设计部分	111
(一) 基于 8088CPU 的简单微型计算机应用系统的设计	111
(二) 设计任务书	114
二、实践部分	114
(一) 课程设计举例 (I) 可调计时器设计	114
(二) 课程设计举例 (II) 模拟计算器设计	125
参考文献	141

第1部分 习题解答

习题解答部分与《微型计算机原理及应用》一书相配套,为书中各章习题的一种解答方案。学生进行习题练习,也是一种实践,是教学环节中的一环。习题练习可分为每次面授后的课外作业及课堂练习。课外作业是教师为巩固本次面授知识而布置的。教师可根据循序渐进及与教学同步的原则布置作业,学生应在面授后抓紧复习。学生在面授时接受知识的百分比因人而异,而且距离面授时间间隔愈长衰减愈快。课堂练习一般在完成一个教学阶段后用一或两个学时完成。教师利用这个时间示范典型习题,以便学生模仿,同时也借此贴近学生,了解学生课堂接受知识情况,以便因人而异解决一些课堂上无法解决的特殊问题。

(一) 习题一解答

1. 解释下列术语:

- (1) 微处理器、微计算机和微计算机系统。
- (2) 单片机、单板机和多板机。
- (3) 微计算机系统的硬件和软件。
- (4) 总线。
- (5) 机器语言、汇编语言和高级语言。

答:(1) 微处理器由算术逻辑单元、控制器、寄存器阵列等组成;微计算机由微处理器、内存储器、I/O 接口电路及系统总线等组成;微计算机系统由硬件(包括微计算机、外围设备等)和软件(包括系统软件、程序设计语言、应用软件等)等组成。

(2) 如果把微处理器、内存储器、I/O 接口电路等这些部件都集成在一个芯片上,由一个芯片构成一台计算机,称之为单片计算机;如果把上述各部件安装在一块印制电路板上而组成微型计算机,称之为单板计算机;如果将处理器、存储器以及 I/O 接口电路安装在不同的印制电路板上,由若干块这样的电路板组合而成的计算机,称之为多板计算机。

(3) 微计算机系统的硬件包括微计算机、外围设备等,微计算机系统的软件包括系统软件、程序设计语言、应用软件等。

(4) 总线是微机中用来进行信息传递的一族公共信号线,按传送信息种类的不同可分为地址总线、数据总线和控制总线。

2. 画出典型的 8 位微机的组成框图,说明各组成部分的作用。

解:8 位微机的组成框图如图 1-1-2 所示。

各组成部分的作用为:

CPU:完成算术和逻辑运算以及控制和协调各部件工作;

存储器:存储器用来存放数据和程序;

外围设备:包括输入和输出设备,完成信息的输入和输出;

接口电路:外围设备与系统总线之间的缓冲区。

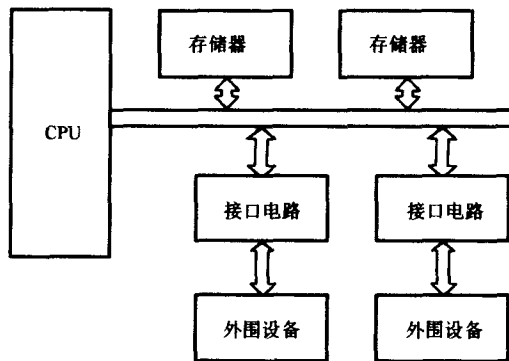


图 1-1-2 8 位微机的组成框图

3. 求出下列各补码表示的二进制数的真值:

- (1) 00000000 (2) 01111110 (3) 11111111 (4) 10000000

答: (1) 0 (2) +126 (3) -1 (4) -128

4. 下列二进制数若为无符号数,它们的值是多少? 若为带符号数,它们的值是多少? 用十进制表示。

- (1) 01101110 (2) 01011001 (3) 10001101 (4) 11111001

答: (1) 01101110 若为无符号数,其值为 110;若为带符号数,其值为 110。

(2) 01011001 若为无符号数,其值为 89;若为带符号数,其值为 89。

(3) 10001101 若为无符号数,其值为 141;若为带符号数,其值为 -115。

(4) 11111001 若为无符号数,其值为 249;若为带符号数,其值为 -7。

5. 已知下列二进制数码,写出它们的原码、反码和补码。

- (1) 00000000 (2) 01111100 (3) 10000010 (4) 11111111

答: (1) 0000 0000 的原码、反码和补码都为 0000 0000

(2) 0111 1100 的原码、反码和补码都为 0111 1100

(3) 1000 0010 的原码、反码和补码分别为 1000 0010、1111 1101、1111 1110

(4) 1111 1111 的原码、反码和补码分别为 1111 1111、1000 0000、1000 0001

6. 下列各数均为用十六进制表示的 8 位二进制数,试说明当它们分别被看作是补码表示的数或字符的 ASCII 时,它们所表示的十进制数及字符是什么?

- (1) 4FH (2) 73H (3) 2BH (4) 6CH

答: (1) 79 'O' (2) 115 's' (3) 43 '+' (4) 108 'l'

7. 试写出下列字符串的 ASCII 码值:

For example

This is a number 1997

答: For example 的 ASCII 码值: 46H 6FH 72H 20H 65H 78H 61H 6DH 70H 6CH 65H

This is a number 1997 的 ASCII 码值: 54H 68H 69H 73H 20H 69H 73H 20H 61H 20H

6EH 75H 6DH 62H 65H 72H 20H 31H 39H 39H 37H。

8. 下列各数均为十进制数,试用 8 位二进制补码计算下列各题,将运算结果用两位十六进制数表示并说明运算结果是否溢出。

$$(1) (-85) + 76 \quad (2) 85 + (-76) \quad (3) (-85) - 76 \quad (4) 85 - (-76)$$

答:(1) 1010 1011

0100 1100

1111 0111

运算结果 0F7H, 即 $(-9)_{16}$, 无溢出。

$$(2) 0101 0101$$

1011 0100

0000 1001

运算结果 09H, 无溢出。

$$(3) 1010 1011$$

1011 0100

0101 1111

$(-85) - 76$ 的十进制运算结果为 -161 , 超出了 -128 , 发生溢出。二进制运算结果为 5FH, 为一错误结果。

$$(4) 0101 0101$$

0100 1100

1010 0001

$85 - (-76)$ 的十进制运算结果为 161 , 超出了 127 , 发生溢出。二进制运算结果为 0A1H, 为一错误结果。

9. 完成下列二进制数的运算,并转换为十进制数进行校核:

$$(1) 10011010 + 00101101 \quad (2) 1100000 - 00001101$$

$$(3) 1101 \times 1010 \quad (4) 11000011 \div 1001$$

答:(1) $10011010 + 00101101 = 1100 0111$ 或 $154 + 45 = 199$

$$(2) 11000000 - 00001101 = 1011 0011 \text{ 或 } 192 - 13 = 179$$

$$(3) 1101 \times 1010 = 1000 0010 \text{ 或 } 13 \times 10 = 130$$

$$(4) 0101 1010 \div 1001 = 1010 \text{ 或 } 90 \div 9 = 10$$

(二) 习题二解答

1. 在下列各项中,选出 8086/8088 的 EU 和 BIU 的组成部件,将所选部件的编号填于后:

EU _____

BIU _____

(1) 标志寄存器

(2) 段寄存器组

(3) ALU

(4) 指令指针

(5) 指令队列

(6) 20 位地址加法器

(7) 通用寄存器组

(8) EU 控制器

(9) 暂存器

(10) 总线控制电路

答:EU (1) (3) (7) (8) (9) _____

BIU (2) (4) (5) (6) (10)

2.8086/8088CPU 内部有哪些寄存器? 它们的主要作用各是什么?

答:8086/8088CPU 内部有以下寄存器:

(1)数据寄存器 包括 AX、BX、CX 和 DX。通常用来存放 16 位的数据和地址。它们中的每一个又可以分为高字节 H 和低字节 L 寄存器,即 AH、BH、CH、DH 及 AL、BL、CL、DL 两组,用来存放 8 位数据,它们均可独立寻址,独立使用。

(2)指针寄存器和变址寄存器 包括 SP、BP、SI 和 DI。它们都是十六位寄存器,可以存放数据,但通常用来存放逻辑地址的偏移量,是形成 20 位物理地址码的其中一部分,在任何情况下,它们都不能独立地形成访问内存的地址码。

(3)段寄存器 8086/8088CPU 的 BIU 中有 4 个 16 位段寄存器,用来存放段起始地址(段基址)的高 16 位。8086/8088 的指令能直接访问这 4 个段。4 个段寄存器是:代码段寄存器 CS(Code Segment)存放当前执行程序所在段的段基址;数据段寄存器 DS(Data Segment)存放当前使用的数据段的段基址,通常数据段用来存放数据和变量;堆栈段寄存器 SS(Stack Segment)存放当前堆栈段的段基址;附加段寄存器 ES(Extra Segment)存放附加数据段的段基址。

(4)控制寄存器组 包括一个 16 位的指令指针 IP 和一个 16 位标志寄存器。指令指针 IP (Instruction Pointer)总是存放着下一次要取出指令的偏移地址;标志寄存器只用了 9 位,其中有 6 位是反映前一次涉及 ALU 操作结果的状态标志,其余 3 位是控制 CPU 操作特征的控制标志。

3.8086/8088CPU 中的标志寄存器有几个标志位? 各标志位在什么情况下置位?

答:8086/8088CPU 设立了一个 16 位寄存器,用了其中 9 位作标志位。有 6 个是反映前一次涉及 ALU 操作结果的状态标志,其余 3 个是控制 CPU 操作特征的控制标志。

6 个状态标志是:

CF(Carry Flag)进位标志:加法时的最高位(字节操作是 D_7 位,字操作是 D_{15} 位)产生进位或减法时最高位产生借位,则 $CF=1$,否则 $CF=0$ 。

AF(Auxiliary Carry Flag)辅助进位标志:加法时 D_3 位有进位或减法时 D_3 位有借位,则 $AF=1$,否则 $AF=0$ 。这个标志只供 BCD 码算术运算时使用。

OF(Overflow Flag)溢出标志:带符号数在进行算术运算时,其结果超出 8 位或 16 位带符号数所能表示的数值范围,产生溢出,则 $OF=1$,否则 $OF=0$ 。

ZF(Zero Flag)零标志:运算结果各位都为零,则 $ZF=1$,否则 $ZF=0$ 。

SF(Sign Flag)符号标志:运算结果为负数即结果的最高位为 1,则 $SF=1$,否则 $SF=0$ 。

PF(Parity Flag)奇偶标志:操作结果的低 8 位中含偶数个 1,则 $PF=1$,否则 $PF=0$ 。

控制标志位的状态由程序设置或由程序清除。

3 个控制标志是:

DF(Direction Flag)方向标志:用来控制数据串操作指令的步进方向。 $DF=1$ 时,数据串指令将以地址的递减顺序对数据串中的数据进行处理; $DF=0$ 时,数据串指令则从低位地址到高位地址作自动递增处理。

IF(Interrupt-Enable Flag)中断允许标志:IF 为 1 时为开中断,CPU 可响应可屏蔽中断请求;IF 为 0 时为关中断,CPU 不响应可屏蔽中断请求。

TF(Trap Flag)陷阱标志:它是为方便程序调试而设的。当 TF 为 1 时,CPU 处于单步执行指令的方式,每执行一条指令就自动产生一个内部中断,转去执行一个中断服务程序,使操作者能够逐条指令地检查一个程序的执行情况。当 TF 为 0 时,CPU 正常执行程序。

4. 对于 8086CPU,已知(DS)=0150H,(CS)=0640H,(SS)=1200H,问:

(1) 在数据段中可存放的数据最多为多少字节? 首末地址各是什么?

(2) 堆栈段中可存放多少个 16 位的字? 首末地址各是什么?

(3) 代码段最大可存放多少个字节的程序? 首末地址各是什么?

答:(1) 在数据段中可存放的数据最多为 64KB,首地址为 01500H,末地址为 114FFH。

(2) 在堆栈段中可存放 32K 个 16 位的字,首地址为 12000H,末地址为 21FFFH。

(3) 代码段中最大可存放 64KB 的程序,首地址为 06400H,末地址为 163FFH。

5. 有一个由 27 个字节组成的数据区,其起始地址为 BA00H:1BA0H,试写出该数据区的首末单元的实际地址。

答:数据区的首地址为 $BA00H \times 10H + 1BA0H = BBBA0H$

数据区的末地址为 $BBBA0H + 1BH - 1 = BBBBAH$

6. 若代码段寄存器(CS)=2000H,指令指针(IP)=2000H,试问指令的实际地址是什么?

答:指令的实际地址为 $2000H \times 10H + 2000H = 22000H$ 。

7. 已知当前数据段位于存储器的 A1000H~B0FFFH 范围内,问 DS 等于多少?

答:因为数据段的范围为 A1000H~B0FFFH,数据段具有 64KB,且其末地址为 B0FFFH,所以 DS 等于 A100H。

8. 有两个 16 位的字 ABCDH,1234H。它们在 8086 系统存储器中的地址分别是 00100H 及 00105H,试画出它们在存储器中存放示意图。

答:这两个字在存储器中的存放示意图如下:

00100H	CD
00101H	AB
00105H	34
00106H	12

9. 8086/8088 最小工作方式与最大工作方式有什么不同? 用什么方法将 8086/8088 置为上述两种工作方式?

答:当把 8086/8088 的 $\overline{MN}/\overline{MX}$ 引脚接到 +5V 时,CPU 就处于最小工作方式。所谓最小工作方式,就是系统中只有一个微处理器 8086 或 8088。在这种系统中,所有的总线控制信号都直接由 8086 或 8088 产生,系统中总线控制逻辑电路被减小到最小,这种方式适合于较小规模系统的应用。

当把 8086/8088CPU 的 $\overline{MN}/\overline{MX}$ 引脚接地时,CPU 就处于最大工作方式。最大工作方式用在需要利用 8086/8088CPU 构成中等或较大(相对于最小方式)系统时。在最大方式系统中,可以只有一个微处理器,也可以有两个或两个以上的微处理器,其中 8086 或 8088 为主处理器,其他处理器称为后援处理器,用来协助主处理器处理某方面的工作。多个微处理器构成的系统称多处理器系统。

10. 试指出 8086/8088 系统总线结构中,8284A 时钟产生器、8282 地址锁存器、8286 收发器及 8288 总线控制器的作用。

答:时钟发生器 8284A 提供频率恒定的单相时钟脉冲 CLK 给 CPU,除此之外,还具有系统复位信号 RESET 产生电路和准备好信号 READY 控制电路。

8282 用来锁存 8086/8088 访问存储器或 I/O 端口时,于总线周期 T_1 状态下发出的地址信号。经锁存后的地址信号可以在整个周期内保持稳定不变。系统中需要用几片 8282,要根据地址的位数来决定。

8286 是 8 位同相收发器,它有 8 路双向缓冲电路,两组数据引脚是对称的。它用来加强数据总线的负载能力。

8288 是 8086/8088CPU 工作于最大方式时,用来代替 CPU 提供总线控制和命令信号的总线控制器。

11. 试叙述 8086 最小工作方式下总线读写操作过程。

答:8086 在最小方式下总线读操作的时序如图 1-2-11(1)所示。

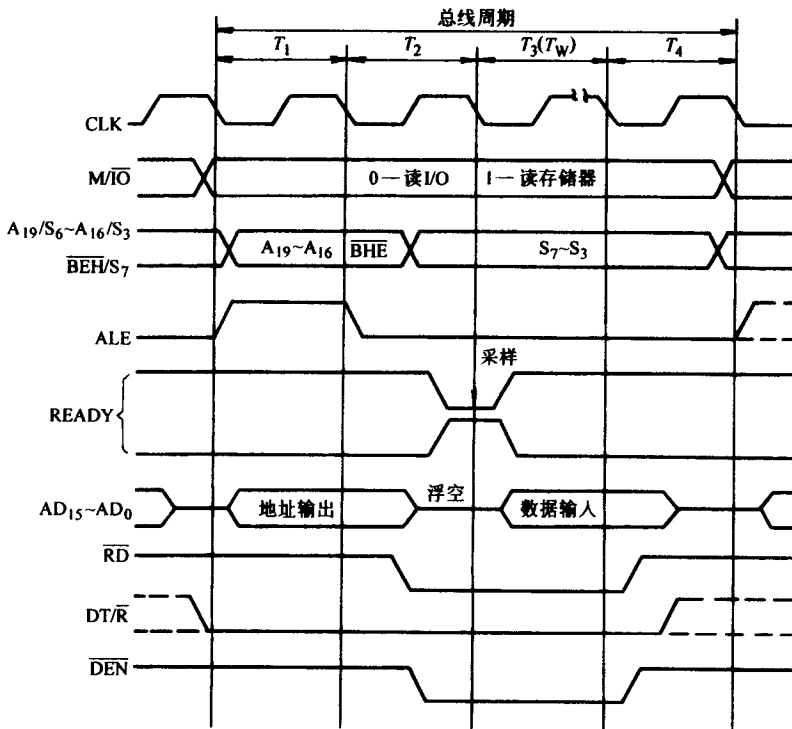


图 1-2-11 (1) 8086 在最小方式下总线读操作的时序

操作过程如下:

(1) T_1 状态

1) CPU 根据执行的是访问存储器还是访问 I/O 端口指令,在 $M/\overline{IO}$ 线上发有效电平。若为读存储器,发高电平;若为读 I/O 端,则为低电平。此信号将持续整个周期。

2) 读取的存储单元或 I/O 端口的 20 位地址信号通过多路复用总线输出,其中高 4 位地址通过地址/状态线 $A_{19}/S_6 \sim A_{16}/S_3$ 送出,低 16 位地址通过地址/数据线 $AD_{15} \sim AD_0$ 送出。这类

信号在 T_1 状态开始送出但只持续一个状态,因此必须及时锁存,以供整个总线周期使用。

3)为了锁存地址信号,CPU 便在 T_1 状态从 ALE 引脚上输出一个正脉冲作地址锁存器 8282 的地址锁存信号。在 ALE 的下降沿到来之前, $\overline{\text{BHE}}$ 和地址信号均已有效。8282 正是用 ALE 的下降沿对地址进行锁存。

4) $\overline{\text{BHE}}$ 信号也在 T_1 状态通过 $\overline{\text{BHE}}/\text{S}_7$ 引脚送出, $\overline{\text{BHE}}$ 和地址 A_0 分别用来对奇、偶地址库进行寻址。

5)使 $\text{DT}/\overline{\text{R}}$ 变为低电平,控制数据总线收发器 8286 为接收数据状态。

(2) T_2 状态

1) $A_{19}/\text{S}_6 \sim A_{16}/\text{S}_3$ 线上由地址信息变成状态信息 $\text{S}_6 \sim \text{S}_3$, $\overline{\text{BHE}}/\text{S}_7$ 由 $\overline{\text{BHE}}$ 变为状态信息 S_7 ,一直持续到 T_4 。

2) $\text{AD}_{15} \sim \text{AD}_0$ 线上的地址信号消失,进入高阻状态,为读入数据作准备。

3) $\overline{\text{DEN}}$ 信号开始变为有效(低电平),开放 8286 总线收发器, $\overline{\text{DEN}}$ 维持有效到 T_4 的中期才结束。

4) $\overline{\text{RD}}$ 信号开始变为有效的低电平,控制被地址信号选中的存储单元或 I/O 端口打开数据输出缓冲器,以便将数据送上数据总线。

5) $\text{DT}/\overline{\text{R}}$ 继续保持低电平,使 8286 处于接收状态。

(3) T_3 状态 存储单元或 I/O 端口将数据送到数据总线 $\text{AD}_{15} \sim \text{AD}_0$,供 8286 缓冲后向 CPU 输入。

(4) T_w 状态 当系统所用的存储器或外设的工作速度较慢,不能在基本总线周期规定的四个状态完成读操作时,它们将通过 8284A 时钟发生器送出 READY 信号给 CPU。CPU 在 T_3 的前沿(下降沿)采样 READY。当采到 READY 为低电平(数据准备就绪)时,CPU 就会在 T_3 和 T_4 之间自动插入等待状态 T_w 。 T_w 可以是一个或多个。CPU 在每个 T_w 的前下降沿又去采样 READY,直到 READY 为高电平(数据准备就绪)时,在本 T_w 状态结束后进入 T_4 状态。在最后一个 T_w 状态,数据肯定已经出现在数据总线上,因此,这时的总线操作和基本总线周期中的 T_3 状态中完全一样。

(5) T_4 状态 在 T_4 状态和前一个状态交界的下降沿处,CPU 读取数据总线上的数据。

8086 最小方式下的总线写操作时序如图 1-2-11(2)所示。基本写操作周期也包括 4 个 T 状态 T_1 、 T_2 、 T_3 和 T_4 ,当存储器或外设速度较慢时,在 T_3 和 T_4 之间插入 1 个或多个 T_w 。各状态下的操作如下:

(1) T_1 状态 与读操作一样,在 $\text{M}/\overline{\text{IO}}$ 线上发有效电平,确定是访问存储器还是访问 I/O 端口,同时将 20 位地址和 $\overline{\text{BHE}}$ 输出,ALE 引脚上的正脉冲的下降沿把 20 位地址及 $\overline{\text{BHE}}$ 状态锁存到 8282 锁存器中, $\text{DT}/\overline{\text{R}}$ 为高电平。

(2) T_2 状态 $\overline{\text{WR}}$ 为低电平,并在 AD 线上撤去地址信号后,立即将数据输出到 $\text{AD}_{15} \sim \text{AD}_0$ 上,不需像读周期一样要维持一个状态的浮空以作缓冲。 $\text{DT}/\overline{\text{R}}$ 维持高电平, $\overline{\text{DEN}}$ 变为低电平,8286 处于发送状态, $A_{19}/\text{S}_6 \sim A_{16}/\text{S}_3$ 及 $\overline{\text{BHE}}/\text{S}_7$ 变为状态输出。

(3) T_3 状态 各信号维持 T_2 时的状态不变,与读周期一样,在 T_3 的前下降沿采样 READY,若需插入 T_w 状态则产生 T_w 从而进入等待状态,否则转入 T_4 状态。 T_w 状态,各信

号也维持不变。

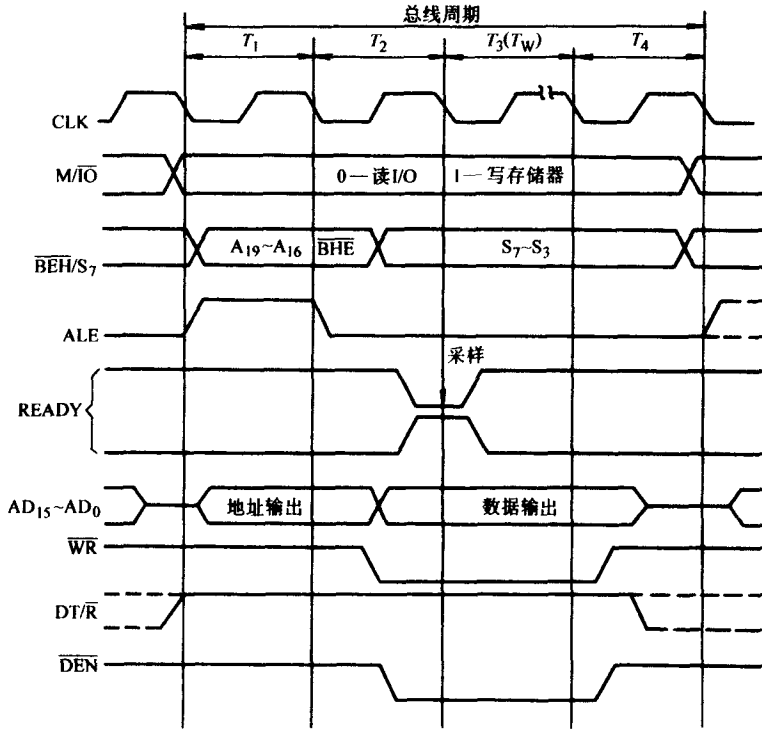


图 1-2-11 (2) 8086 最小方式下的总线写操作时序

(4) T_4 状态 在 T_4 状态前期, $\overline{WR}$ 变无效并撤除数据总线上的数据; 在 T_4 后期, $M/\overline{IO}$ 也变无效。此时 $\overline{DEN}$ 也变为高电平, 从而关闭 8286 收发器。

12. 试叙述 8086 最大工作模式下, $\overline{RQ}/\overline{GT}_0$ 与 $\overline{RQ}/\overline{GT}_1$ 引脚的作用。

答: 8086/8088 在最大方式下, 引脚 $\overline{RQ}/\overline{GT}_0$ 和 $\overline{RQ}/\overline{GT}_1$ 提供的信号称为总线请求/总线允许/总线释放信号, 它们可分别连接到两个其他的总线主模块 (例如数值协处理器和 I/O 处理器)。 $\overline{RQ}/\overline{GT}_0$ 和 $\overline{RQ}/\overline{GT}_1$ 均为双向低电平有效, 前者优先级高于后者。8086/8088 最大方式下的总线请求/允许/释放的操作时序如图 1-2-12 所示。

由图可见, CPU 在每个时钟周期的上升沿对 $\overline{RQ}/\overline{GT}$ 引脚进行检测, 当检测到外部向 CPU 送来一个“请求”负脉冲时 (宽度为一个时钟周期), 则在下一个 T_4 状态或 T_1 状态从同一引脚上由 CPU 向请求总线使用权的主模块发回一个“允许”负脉冲 (宽度仍为一个时钟周期), 并使具有三态的输出线, $AD_{15} \sim AD_0$ 、 $A_{19}/S_6 \sim A_{16}/S_3$ 、 $\overline{RD}$ 、 $\overline{LOCK}$ 、 S_2 、 S_1 、 S_0 和 $\overline{BHE}/S_7$ 都处于浮空状态, CPU 暂时与总线断开。

请求总线使用权的主模块收到这个“允许”脉冲时, 便获得了总线的控制权, 可以对总线占用一个或几个总线周期。当它要释放总线时, 便又从 $\overline{RQ}$ 线上发一个“释放”负脉冲 (宽度仍为一个时钟周期), CPU 检测到释放脉冲后, 经过两个时钟周期的延迟, 才能重新获得总线的控制权。需要注意的是, 由于 $\overline{RQ}/\overline{GT}_0$ 优先级高于 $\overline{RQ}/\overline{GT}_1$, 当两个引脚都同时向 CPU 发出总线请求时, CPU 会先在 $\overline{RQ}/\overline{GT}_0$ 上发允许信号, 等到 CPU 再次得到总线控制权时, 才去响应 $\overline{RQ}/\overline{GT}_1$ 引脚上的请求。不过, 一旦接于 $\overline{RQ}/\overline{GT}_1$ 上的总线主模块得到了总线控制权, 就要等

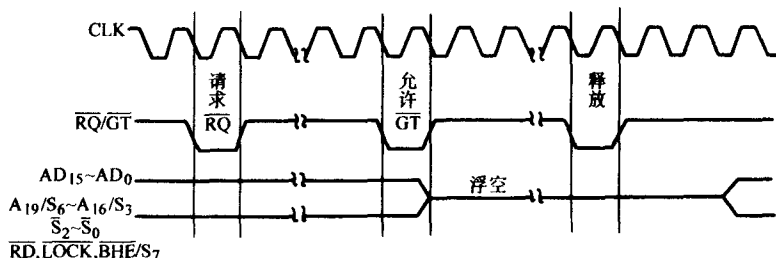


图 1-2-12 8086/8088 最大方式下的总线请求/允许/释放的操作时序

到该主模块释放了总线, CPU 又收回了总线控制权后, 才会去响应 $\overline{RQ}/\overline{GT}_0$ 引脚上的总线请求。另外, 与最小方式下执行总线保持请求/保持响应操作一样, 在 CPU 让出了总线的控制权后, CPU 内部的执行部件 EU 仍可以继续执行指令队列中的指令, 直到遇到一条需执行总线操作周期的指令或指令队列已空为止。

(三) 习题三解答

1. 8086 语言指令的寻址方式有哪几类? 用哪一种寻址方式的指令执行速度最快?

答: 8086/8088 提供了 8 种寻址方式对操作数寻址, 它们是: 隐含寻址、立即数寻址、寄存器寻址、直接寻址、寄存器间接寻址、寄存器相对寻址、基址变址寻址和相对基址变址寻址。寄存器寻址的指令执行速度最快。

2. 用寄存器进行间接寻址方式时, BX、BP、SI、DI 分别针对什么情况来使用? 用这 4 个寄存器分别组合进行间接寻址时, 物理地址是怎样计算的? 举例说明。

答: 当指令中指定用 BX、SI 或 DI 为间址寄存器, 则操作数在数据段中, 这时以段寄存器 DS 的内容为段基址, 操作数的物理地址为:

$$PA = (DS) \times 16 + \begin{cases} (BX) \\ (SI) \\ (DI) \end{cases}$$

源操作数的偏移地址存放于 SI 中, 所以称 SI 为源变址寄存器; 目的操作数的偏移地址存放于 DI 中, 所以称 DI 为目的变址寄存器。

例如: 若已知 $(DS) = 2000H$, $(BX) = 4000H$, 则源操作数的物理地址 $PA = (DS) \times 16 + (BX) = 24000H$ 。

若已知 $(DS) = 2000H$, $(SI) = 5000H$, 则源操作数的物理地址 $PA = (DS) \times 16 + (SI) = 25000H$ 。

若已知 $(DS) = 2000H$, $(DI) = 6000H$, 则源操作数的物理地址 $PA = (DS) \times 16 + (DI) = 26000H$ 。

BP 称基址指针, 用于存放位于堆栈段中的一个数据区基址的偏移地址。若指令中指定 BP 为间址寄存器, 则操作数放在堆栈段中。这时, 以段寄存器 SS 中的内容为段基址, 操作数的物理地址为: $PA = (SS) \times 16 + (BP)$ 。

例如: 若已知 $(SS) = 3000H$, $(BP) = 4000H$, 则源操作数的物理地址 $PA = (SS) \times 16 +$

(BP) = 3400H。

3. 设 (DS) = 2000H, (SS) = 1500H, (SI) = 00A0H, (BX) = 0100H, 数据变量 VAL 的偏移地址为 0050H, 试指出以下指令的源操作数是什么寻址方式? 若为存储器操作数, 其物理地址是多少?

- (1) MOV AX, 0ABH
- (2) MOV AX, [100H]
- (3) MOV AX, VAL
- (4) MOV BX, [SI]
- (5) MOV AL, VAL[BX]
- (6) MOV CL, [BX][SI]

答: (1) 源操作数是立即数寻址。

(2) 源操作数是直接寻址, 有效地址 EA 是一个立即数, 其物理地址是:

$$PA = (DS \times 10H) + EA = (2000H \times 10H) + 100H = 20100H$$

(3) 源操作数是直接寻址, 有效地址 EA 是个符号地址, 其物理地址是:

$$PA = (DS \times 10H) + EA = (2000H \times 10H) + 0050H = 20050H$$

(4) 操作数是寄存器间接寻址, 其物理地址是:

$$PA = (DS) \times 10H + (SI) = (2000H \times 10H) + 00A0H = 200A0H$$

(5) 源操作数是寄存器相对寻址, 其物理地址是:

$$PA = (DS) \times 10H + (BX) + VAL = (2000H \times 10H) + 0100H + 0050H = 20150H$$

(6) 源操作数是基址变址寻址, 其物理地址是:

$$PA = (DS) \times 10H + (SI) + (BX) = (2000H \times 10H) + 00A0H + 0100H = 201A0H$$

4. 指出下列指令中哪些是正确的? 哪些是错误的? 若有错误的, 请说明原因。

- (1) MOV AH, BX
- (2) MOV [BX], [SI]
- (3) MOV AX, [SI][DI]
- (4) MOV [BX], [1000H]
- (5) XCHG CS, AX
- (6) XCHG BX, IP
- (7) PUSH CS
- (8) POP CS

答: (1) 错误, 源和目的操作数的类型(长度)不一致。

(2) 错误, 源和目的操作数不能同为存储器操作数。

(3) 错误, SI、DI 不能同时出现在源操作数中。

(4) 错误, MOV 指令中源和目的操作数不能同为存储器操作数。

(5) 错误, 交换指令不能使用段寄存器。

(6) 错误, 交换指令不能使用 IP, CS 和 IP 的值一般只能在执行转移指令时改变。

(7) 正确, CS 可作源操作数。

(8) 错误, CS 不能作目的操作数。

5. 已知 (DS) = 2000H, (BX) = 0300H, (SI) = 0002H, (20300H) = 12H, (20301H) = 34H,

(20302H) = 56H, (20303H) = 78H, (21400H) = 2AH, (21401H) = 4CH, (21402H) = B7H, (21403H) = 65H, 试说明下列各条指令执行完后 AX 寄存器的内容。

- (1) MOV AX, 1400H
- (2) MOV AX, BX
- (3) MOV AX, [1400H]
- (4) MOV AX, [BX]
- (5) MOV AX, 1100H[BX]
- (6) MOV AX, [BX][SI]
- (7) MOV AX, 1100H[BX][SI]

答: (1) (AX) = 1400H

(2) (AX) = (BX) = 0300H

(3) 物理地址 PA = (2000H × 10H) + 1400H = 21400H, (AX) = 4C2AH

(4) 物理地址 PA = (2000H × 10H) + 0300H = 20300H, (AX) = 3412H

(5) 物理地址 PA = (2000H × 10H) + 1100H + 0300H = 21400H, (AX) = 4C2AH

(6) 物理地址 PA = (2000H × 10H) + 0300H + 0002H = 20302H, (AX) = 7856H

(7) 物理地址 PA = (2000H × 10H) + 1100H + 0300H + 0002H = 21402H, (AX) = 65B7H

6. 根据以下要求, 试写出相应的汇编语言指令。

(1) 把 BX 寄存器和 DX 寄存器的内容相加, 结果存入 BX 寄存器中。

(2) 用寄存器 BX 和 SI 的基址变址寻址方式把存储器中的一个字节与 AL 寄存器的内容相加, 并把结果送到 AL 寄存器中。

(3) 用寄存器 BX 和位移量 0B2H 的寄存器相对寻址方式把存储器中的一个字和 CX 相加, 并把结果送回存储器中。

(4) 用位移量为 0524H 的直接寻址方式把存储器中的一个字与立即数 3C5AH 相加, 并把结果送回该存储单元中。

解: 相应的汇编语言指令如下:

- (1) ADD BX, DX
- (2) ADD AL, [BX][SI]
- (3) ADD [BX + 00B2H], CX
- (4) ADD WORD PTR [0524H], 3C5AH

7. 已知堆栈段寄存器 SS 的内容是 0FF0AH, 堆栈指针 SP 的内容是 0008H, 先执行两条把 8057H 和 0F7CH 分别进栈的 PUSH 指令, 再执行一条 POP BX 指令。试画出堆栈区和 SP 的内容过程变化示意图(标出存储单元的物理地址)。

解: 堆栈区和 SP 的内容变化过程示意如图 1-3-7 所示:

8. 字节扩展指令和字扩展指令用在什么场合? 举例说明。

答: 8086/8088 的除法要求被除数的位数为除数的 2 倍, 即字节除法时要求被除数为 16 位, 字相除时要求被除数为 32 位。如果被除数不满足此要求时, 对于无符号数, 将高位补零即可; 但对于带符号数, 不能简单地将被除数的高位部分填 0, 必须按照被除数的符号来扩展。当为正数时, 高位部分各位补 0; 当为负数时, 高位部分各位补 1。为此, 8086/8088 提供了专门的符号扩展指令 CWD 和 CBW, 方便用户扩展。

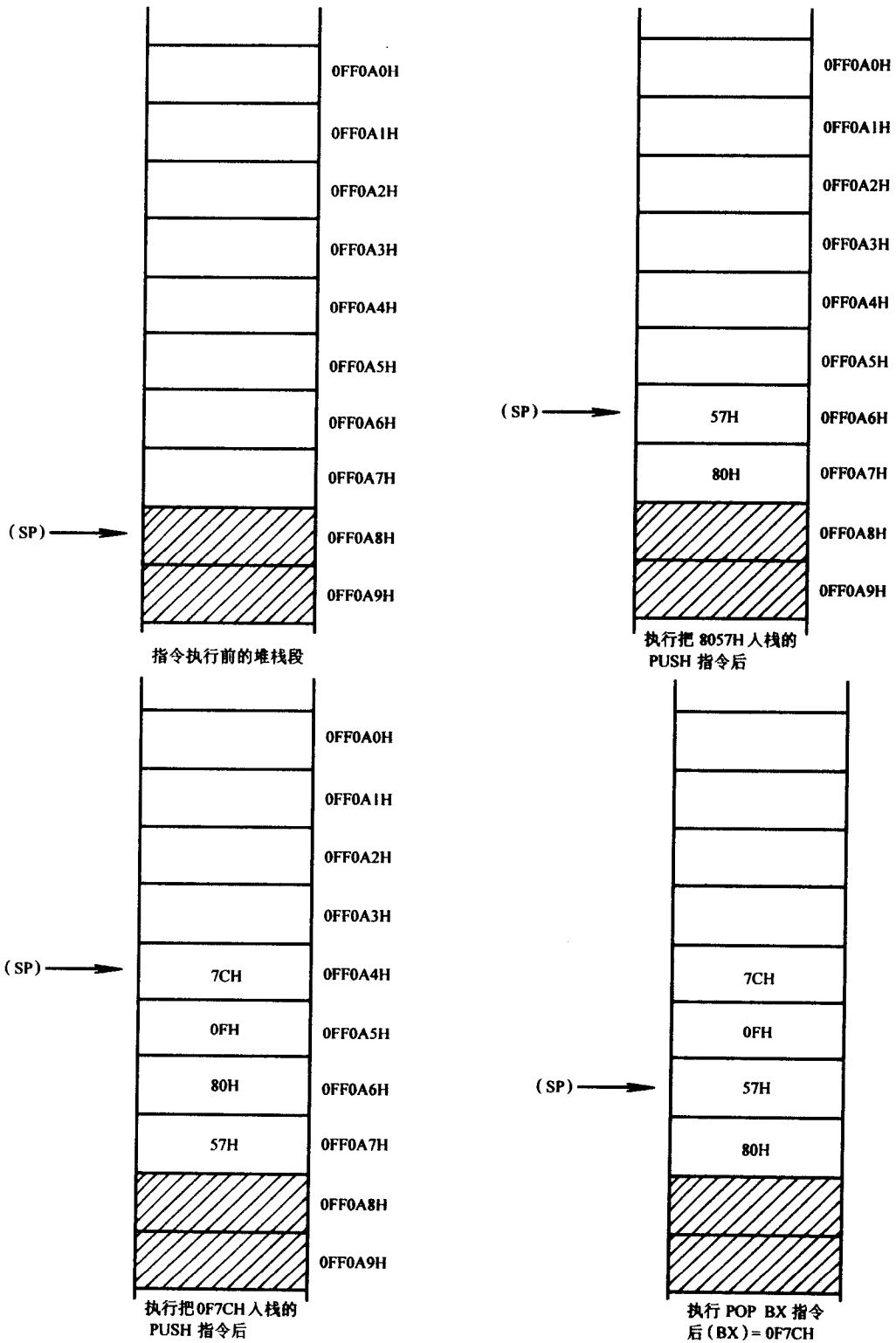


图 1-3-7 堆栈区和 SP 的内容变化过程示意