高等学校计算机基础教育规划教材

微型计算机原理与接口技术习题及实验指导

侯彦利 主编

清华大学出版社

高等学校计算机基础教育规划教材

微型计算机原理与接口技术习题及实验指导

侯彦利　主编
赵永华　郭 威　马爱民　刘 通　编著

清华大学出版社
北京

内 容 简 介

本书作为《微型计算机原理与接口技术》的配套教材，包含了《微型计算机原理与接口技术》各章节的知识要点、习题解答和汇编语言程序设计实验及微型计算机接口基本实验。本书以“伟福 Lab 8000 系列单片机仿真实验系统”为基础，介绍“微型计算机原理与接口技术”课程所需的各类实验，包括实验要求、实验目的、实验电路图、实验步骤和实验例程。对每个实验都给出了较为详细的硬件原理图，对实验需要的一些基础知识也进行了必要的补充。

本书可作为普通高等院校工科非计算机类本、专科学生“微型计算机原理与接口技术”课程的实验教材，也可作为工程技术人员的学习参考书。

图书在版编目(CIP)数据

微型计算机原理与接口技术习题及实验指导/侯彦利主编. —北京：清华大学出版社，2017
（高等学校计算机基础教育规划教材）
ISBN 978-7-302-46624-6

Ⅰ. ①微… Ⅱ. ①侯… Ⅲ. ①微型计算机－理论－高等学校－教学参考资料 ②微型计算机－接口技术－高等学校－教学参考资料 Ⅳ. ①TP36

中国版本图书馆 CIP 数据核字(2017)第 031268 号

责任编辑：袁勤勇
封面设计：常雪影
责任校对：焦丽丽
责任印制：刘海龙

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦 A 座　　邮　　编：100084
社 总 机：010-62770175　　邮　　购：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：保定市中画美凯印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：11.75　　**字　　数**：269 千字
版　　次：2017 年 5 月第 1 版　　**印　　次**：2017 年 5 月第1 次印刷
印　　数：1～2000
定　　价：29.00 元

产品编号：074215-01

前言

“微型计算机原理与接口技术”是高等院校非计算机专业学生，特别是工学各相关专业学生学习微型计算机基本知识和应用技能的课程。课程帮助学生掌握微型计算机的硬件组成原理，学会运用指令系统和汇编语言进行程序设计，了解 C/C++ 与汇编语言混合编程的方法，掌握微型计算机接口的基本技术，为后续智能控制系统类课程的学习打下基础。实验教学是本课程的重要组成部分，是学生掌握程序设计方法和计算机控制系统电路设计的关键环节。

本课程实验采用的主要设备为伟福 Lab 8000 系列单片机仿真实验系统。所有实验内容均以此实验箱为基础。本书中的第 3、4、6、7、8 章安排了实验内容，每个实验都紧扣理论知识要点，采用由简到繁、步步深入的方法引导学生做实验。实验与实际应用相结合，充满趣味性，可充分调动学生的积极性。

本书第 1、2 章由郭威编写，第 3、4、7 章由侯彦利编写，第 5 章由赵永华编写，第 6 章由刘通编写，第 8、9 章由马爱民编写，全书由侯彦利统稿。

本书内容包括《微型计算机原理与接口技术》一书中各章节后的习题解答，仅供参考。

由于作者水平有限，书中难免存在不足之处，敬请读者批评指正。

编　者

2016 年 8 月于吉林大学

目录

第1章

微型计算机基础知识

1.1 知识要点

1. 数制转换方法

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

十进制数转换为十六进制数：方法一为整数部分采用“除16取余”，小数部分采用“乘16取整”。方法二为先把十进制数转换为二进制数，之后再转换为十六进制数。

2. 数据格式

微型计算机中使用的数据通常以无符号整数、有符号整数、浮点数(实数)、ASCII码、Unicode码、BCD码的形式出现。无符号整数和有符号整数以字节、字、双字的形式存储。

3. 无符号数的表示范围及运算溢出判断

一个 n 位无符号二进制数 X，其表示数的范围为 $0 \leqslant X \leqslant 2^n-1$。

微处理器的算术运算单元只能进行有限位的二进制数运算。2个8位的二进制数进行加减运算，运算结果只保留8位，超出的部分被丢弃。16位、32位的二进制数也是如此。

运算结果如果超出了8位二进制数的取值范围(0～255)，则最高位被丢弃，导致运算结果错误，计算机将这种情况称为溢出。2个16位二进制数相加，结果有可能超出16位二进制数的取值范围，导致最高位被丢弃，运算结果溢出。32位、64位的二进制数加法运算都有可能溢出。

由此可见，在无符号数的加减运算中，如果最高位向前有进位(加法)或借位(减法)，则运算结果产生溢出。

微型计算机中，如果加减运算中最高位向前有进位或借位，将使微处理器标志寄存器中的CF位置1。利用CF位，通过编程可以矫正运算结果，还可以在8位的微处理器上实现16位、32位或者64位甚至更多位的二进制数加减运算。

无符号二进制数的乘法运算一般不会产生溢出，因为 2 个 8 位数相乘，乘积不会超出 16 位二进制数；2 个 16 位数相乘，乘积不会超出 32 位。微处理器为乘积准备了足够的存储空间。

无符号二进制数的除法运算有可能产生溢出，当除数较小时，运算结果可能超出微处理器为除法运算结果准备的存储空间，从而产生溢出。除法溢出时微处理器会产生溢出中断，提醒程序员程序出错。

4. 补码

带符号数在计算机中有三种表示方法：原码、反码和补码。在计算机中，带符号整数一般都用补码表示。带符号数的运算都是补码运算。

补码加法规则：　$[X+Y]_{补}=[X]_{补}+[Y]_{补}$

补码减法规则：　$[X-Y]_{补}=[X]_{补}-[Y]_{补}=[X]_{补}+[-Y]_{补}$

$[-Y]_{补}$ 是通过对 $[Y]_{补}$ 求变补得来的，即对 $[Y]_{补}$ 包括符号位在内的每一位，按位取反并加 1。

5. 带符号数的溢出判断

补码运算判断溢出的方法：对于一个 n 位的补码，如果运算过程中 $C_{n-1}\oplus C_{n-2}=1$，则运算结果产生溢出；如果 $C_{n-1}\oplus C_{n-2}=0$，则运算结果没有溢出。其中，$\oplus$ 表示异或运算。微型计算机中，如果运算结果溢出，将使微处理器标志寄存器中的 OF 位置 1。

6. 浮点数

日常工作中经常遇到实数，计算机中的实数也称为浮点数。浮点数的表示方法采用科学计数法。浮点数有 4 字节和 8 字节两种类型。4 字节浮点数称为单精度实数，8 字节浮点数称为双精度实数。单精度格式包括 1 个符号位、8 位阶码和 23 位尾数，其中第 31 位是符号位，表示实数的符号，正数用 0 表示，负数用 1 表示，与有符号数的表示方法一致。第 30～23 位存储阶码，第 22～0 位存储尾数。

双精度格式包括 1 个符号位、11 位阶码和 52 位尾数。其中第 63 位是符号位，表示实数的符号。第 62～52 位存储阶码，第 51～0 位存储尾数。

阶码以移码的形式表示。单精度格式中偏移量为 127(7FH)，双精度格式中偏移量为 1023(3FFH)。存储阶码之前，阶码要加上偏移量，所以阶码也称为移码阶。

7. 基本逻辑运算及常用逻辑部件

计算机中的“逻辑”指的是输入与输出之间的一种因果关系，用 0 和 1 表示，并依此进行推理运算，就是常说的逻辑代数或布尔代数。逻辑代数可以用 $Y=F(a,b,c,d)$ 这样的逻辑函数表示。变量可以有一个、两个或多个。变量的取值只有两个：0 或 1，它不代表大小，只代表事物的两个对立性质，如真假、有无、对错等。函数值也只有这两个取值。在逻辑代数中有与、或、非三种基本的逻辑运算。

(1) 与门。

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

$$F = A \wedge B$$

(2) 或门。

如图 1-2 所示，或门是实现“或”运算的电路。若输入的逻辑变量为 A 和 B，则通过或门输出的结果 F 可表示如下：

$$F = A \vee B$$

(3) 非门。

如图 1-3 所示，非门是实现“非”运算的电路，又称反相器。它只有一个输入端和一个输出端。若输入的逻辑变量为 A，则通过非门输出的结果 F 可表示如下：

$$F = \overline{A}$$

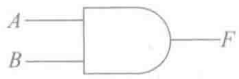

图 1-1 与门逻辑符号

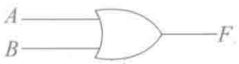

图 1-2 或门逻辑符号

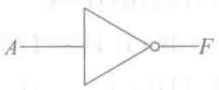

图 1-3 非门逻辑符号

(4) 异或门。

如图 1-4 所示，实现异或运算的电路称为异或电路。

(5) 与非门。

如图 1-5 所示，与非门是实现先“与”运算再“非”运算的电路。若输入的逻辑变量为 A 和 B，则通过与非门输出的结果 F 可表示如下：

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

(6) 或非门。

如图 1-6 所示，或非门是先实现“或”运算再实现“非”运算的电路。

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

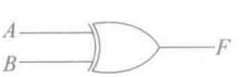

图 1-4 异或门逻辑符号

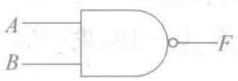

图 1-5 与非门逻辑符号

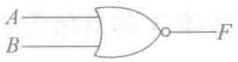

图 1-6 或非门逻辑符号

(7) 三态门。

如图 1-7 所示，三态门常用于微处理器的总线传输。三态门除了具有一般门电路的输出高、低电平外，还具有高输出阻抗的第三种状态，称为高阻态，又称禁止态。图 1-7 是低电平使能的三态门，其中 A 是输入端，F 是输出端，EN 是使能端。

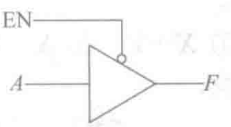

图 1-7 三态门逻辑符号

8. 编码

将人们常用的每个字母或符号统一分配一个二进制数，称为字符编码。ASCII 码、Unicode 码都是字符编码。

BCD(Binary Coded Decimal)码也是一种编码，但它不是用来显示或打印的编码。它是对十进制数进行编码，是用二进制数表示十进制数。例如用4位二进制数表示1位十进制数。BCD码有两种格式：压缩格式和非压缩格式。压缩BCD码用1个字节存储2位十进制数。非压缩BCD码用1个字节存储1位十进制数，这种方式只使用低4位存储十进制数，高4位为0。

1.2 习题解答

1. 微型计算机包括几部分？

解：微处理器、存储器、输入/输出、总线。

2. 完成下列数制的转换。

① 10101101B=(　　)D=(　　)H。

解：10101101B=173D=ADH。

② 0.11B=(　　)D。

解：0.11B=0.75D。

③ 211.25=(　　)B =(　　)H。

解：211.25=11010011.01B=D3.4H。

④ 10111.0101B=(　　)H=(　　)BCD。

解：10111.0101B=17.5H=(0010 0011.0011 0001 0010 0101)BCD。

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

解：

$$[X]_{原}=01011010B \quad [Y]_{原}=10011011B$$
$$[X]_{反}=01011010B \quad [Y]_{反}=11100100B$$
$$[X]_{补}=01011010B \quad [Y]_{补}=11100101B$$

4. 已知X的真值为32，Y的真值为-19，求$[X+Y]_{补}=?$

解：

$$[X]_{补}=[32]_{补}=00100000,\quad [Y]_{补}=[-19]_{补}=11101101$$
$$[X+Y]_{补}=00100000+11101101=00001101,\quad X+Y=13$$

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

① $X+Y$　② $X-Y$　③ $-X+Y$　④ $-X-Y$

解：

$$[X]_{补}=[51]_{补}=00110011,\quad [Y]_{补}=[-86]_{补}=10101010$$
$$[-X]_{补}=11001101,\quad [-Y]_{补}=01010110$$
$$[X+Y]_{补}=00110011+10101010=11011101,\quad X+Y=-35$$

因为$C_6 \oplus C_7=0$，所以未产生溢出

$[X-Y]_{补}=[X]_{补}+[-Y]_{补}=00110011+01010110=10001001$，　$X-Y=-119$

因为 $C_6 \oplus C_7=1$，所以产生溢出

$[-X+Y]_{补}=[-X]_{补}+[Y]_{补}=11001101+10101010=01110111$，　$-X+Y=119$

因为 $C_6 \oplus C_7=1$，所以产生溢出

$[-X-Y]_{补}=[-X]_{补}+[-Y]_{补}=11001101+01010110=00100011$，　$X-Y=35$

因为 $C_6 \oplus C_7=0$，所以未产生溢出。

6. 若使与门的输出端输出高电平，则各输入端的状态是什么？

解：各输入端为高电平。

7. 若使与非门的输出端输出低电平，则各输入端的状态是什么？

解：各输入端为高电平。

第2章

8086/8088 微处理器

2.1 知识要点

1. 8086 与 8088 的主要区别

8086 与 8088 同属于第三代 16 位的微处理器。这两款 CPU 的硬件结构没有太大的区别，二者的主要区别是数据总线宽度，8086 的数据总线宽度为 16 位，而 8088 的数据总线宽度为 8 位。在内部结构上，两款微处理器都有指令队列，8086 的指令队列长度为 6B，而 8088 的指令队列长度为 4B。

2. 8088 微处理器的功能结构

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

执行单元的主要功能是译码分析指令，执行指令，暂存中间运算结果并保留结果特征。执行单元包括 EU 控制器，算术逻辑运算单元 ALU，通用寄存器组 AX、BX、CX、DX、SP、BP、DI、SI，专用寄存器，状态标志寄存器等部件，这些部件的宽度都是 16 位。执行单元通过 EU 控制电路从指令队列中取出指令代码，并对指令进行译码形成各种操作控制信号，控制 ALU 完成算术或逻辑运算，并将运算结果的特征保存在标志寄存器 FLAGS 中；控制其他各部件完成指令所规定的操作。如果指令队列为空，EU 就等待。

总线接口单元包括指令队列、地址加法器、段寄存器、指令指针寄存器和总线控制逻辑。总线接口单元负责 CPU 与内存或输入/输出接口的信息传送，包括取指令、取操作数、保存运算结果。当 EU 从指令队列中取走指令，指令队列出现两个或两个以上的字节空间，且 EU 未向 BIU 申请读/写存储器操作数时，BIU 就顺序地预取后续指令的代码，并填入指令队列中。在 EU 执行指令过程中，BIU 负责从指定的内存单元或外部设备读取 EU 需要的数据，并负责将 EU 运算结果存储到存储器。当 EU 执行跳转指令时，BIU 使指令队列复位，并立即从新地址取出指令传送给 EU 执行，然后再读取后续的指令序列填满指令队列。

3. 指令流水线

在8086/8088 CPU中，EU和BIU两部分按流水线方式工作。EU从BIU的指令队列中取指令并执行指令。在EU执行指令期间，BIU可以取指令并放在指令队列中。EU执行指令和BIU取指令同时进行，节省了CPU访问内存的时间，加快了程序的运行速度。

8086/8088 CPU通过采用指令流水线技术，执行单元和总线接口单元与指令队列协同工作，实现指令的并行执行，提高了CPU的利用率，同时也降低了CPU对存储器存取速度的要求。

4. 8088微处理器的引脚定义

$AD_7 \sim AD_0$：8088 CPU地址/数据分时复用总线(Address/Data bus)，双向，三态。

$A_{15} \sim A_8$：8位地址信号，输出，三态。在整个总线周期内提供存储器高8位地址。

$A_{19}/S_6 \sim A_{16}/S_3$：分时复用地址/状态总线(Address/Status bus)，输出，三态。提供地址信号$A_{19} \sim A_{16}$及状态位$S_6 \sim S_3$。

INTR：中断请求(Interrupt Request)信号，输入，用于申请一个硬件中断。当IF=1时，若INTR保持高电平，则8088 CPU在当前指令执行完毕后就进入中断响应周期($\overline{\text{INTA}}$变为有效)。

NMI：非屏蔽中断(Non-Maskable Interrupt)输入信号。与INTR信号类似，但NMI中断不必检查IF标志位是否为1。

I/O：输出，三态。该引脚选择存储器或I/O端口，即微处理器地址总线是存储器地址还是I/O端口地址。

$\overline{\text{RD}}$：读信号，输出，三态。当它为低电平时，CPU通过数据总线接收来自存储器或I/O设备的数据。

$\overline{\text{WR}}$：写选通信号，输出，三态。指示8086/8088 CPU正在输出数据给存储器或I/O设备。在$\overline{\text{WR}}$为低电平期间，数据总线包含给存储器或I/O设备的有效数据。

$\overline{\text{INTA}}$：中断响应(Interrupt Acknowledge)信号，输出。响应INTR输入。该引脚常用于选通中断向量码以响应中断请求。

5. 最小模式下系统总线的形成

8088微处理器工作在最小模式时，系统所有控制信号由CPU自身产生。最小模式下系统总线包括数据总线$D_0 \sim D_7$、地址总线$A_0 \sim A_{19}$，主要的控制信号有$\overline{\text{RD}}$、$\overline{\text{WR}}$、IO/$\overline{\text{M}}$和$\overline{\text{INTA}}$。

6. 最大模式下系统总线的形成

8088 CPU工作在最大模式时，系统的一些控制信号由总线控制器8288产生。地址总线和数据总线的生成与最小模式一样。最大模式仅仅用作系统包含8087算术协处理器的情况。

最大模式下系统总线包括数据总线 $D_0 \sim D_7$、地址总线 $A_0 \sim A_{19}$，主要的控制信号有 $\overline{\text{MEMR}}$、$\overline{\text{MEMW}}$、$\overline{\text{IOR}}$、$\overline{\text{IOW}}$、$\overline{\text{INTA}}$。

7. 8088 CPU 的存储器组织

8088 CPU 有 20 根地址线，可寻址的最大内存空间为 $2^{20}=1\text{MB}$，地址范围为 00000H～FFFFFH。每个存储单元对应一个 20 位的地址，这个地址称为存储单元的物理地址。每个存储单元都有唯一的一个物理地址。

8088 CPU 将可直接寻址的 1MB 的内存空间划分成一些连续的区域，称为段。每段的长度最大为 64KB，并要求段的起始地址必须能被 16 整除，形式如 XXXX0H。8088 将 XXXXH 称为段基址，存储在段寄存器 CS、DS、SS、ES 中。段基址决定了该段在 1MB 内存空间中的位置。段内各存储单元地址相对于该段起始单元地址的位移量称为段内偏移量。段内偏移量从 0 开始，取值范围为 0000H～FFFFH。

分段管理要求每个段都由连续的存储单元构成，并且能够独立寻址，而且段和段之间允许重叠。根据 8088 CPU 分段的原则，1MB 的存储空间中有 $2^{16}=64\text{K}$ 个地址符合要求，这使得理论上程序可以位于存储空间的任何位置。

程序中使用的存储器地址由段基址和段内偏移地址组成，这种在程序中使用的地址称为逻辑地址。逻辑地址通常写成 XXXXH:YYYYH 的形式，其中 XXXXH 为段基址，YYYYH 为段内偏移地址。段基址和偏移地址与物理地址之间的关系如下：

$$\text{物理地址}=\text{段基址}\times 10\text{H}+\text{段内偏移}$$

段基址乘以 10H 相当于把 16 位的段基址左移 4 位，然后再与段内偏移地址相加就得到物理地址。例如，逻辑地址 A562H:9236H 对应的物理地址是 AE856H。

$$\text{A562H}\times 10\text{H}=\text{A5620H}$$

$$\text{A5620H}+9236\text{H}=\text{AE856H}$$

8. 8088 CPU 的编程结构

8088 CPU 含有 14 个 16 位寄存器，按功能可以分为三类：通用寄存器、段寄存器和控制寄存器。

AX、BX、CX、DX、SP、BP、SI 和 DI 是 8 个 16 位的通用寄存器，这些寄存器都可以用来存放 16 位的二进制数。每一个 AX、BX、CX 和 DX 寄存器又可以分为 2 个 8 位的寄存器，它们可以单独使用以处理字节类型的数据。通用寄存器一般用于存放参与运算的数据或保存运算结果。

段寄存器 CS、DS、SS 和 ES 用于存放段基址，即段起始地址的高 16 位二进制数。

控制寄存器包括指令指针寄存器 IP 和状态标志寄存器 FLAGS。

FLAGS 称为标志寄存器或程序状态字(Program Status Word，PSW)。标志寄存器是一个 16 位的寄存器，8088 CPU 只使用了其中 9 位，分为两类：一类称为状态标志，反映指令执行结果的特征，共有 6 位；另一类是控制标志，用于控制微处理器的操作，共有 3 位。

2.2 习题解答

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

解：8086/8088 CPU 由 EU 和 BIU 两大功能部分组成。

执行单元的主要功能是译码分析指令，执行指令，暂存中间运算结果并保留结果特征。

总线接口单元负责 CPU 与内存或输入/输出接口的信息传送，包括取指令、取操作数、保存运算结果。

2. 什么是指令流水线？指令流水线需要哪些硬件支持？

解：指令流水线是指 8086/8088 CPU 内部的执行单元 EU 和总线接口单元 BIU 通过指令队列协同工作，从而实现指令的并行执行。指令流水线最主要的硬件支持是 BIU 内部的指令队列。

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

解：物理地址＝段基址×16＋段内偏移。

已知逻辑地址为 2D1EH：35B8H，则对应的物理地址＝2D1EH×16＋35B8H＝30798H。

4. 8088 和 8086 CPU 的指令预取队列的长度分别是多少？

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

5. 简述 8086/8088 CPU 内部的各寄存器的作用。

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

SP：用于存放堆栈栈顶的段内偏移量。

BP：用于存放访问内存时的偏移地址。

SP 和 BP 也可以用来存放数据，它们的默认段寄存器都是 SS。

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

CS：代码段寄存器，用于存放代码段的段基址。

DS：数据段寄存器，用于存放数据段的段基址。

SS：堆栈段寄存器，用于存放堆栈段的段基址。

ES：附加段寄存器，用于存放附加段的段基址。

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

FLAGS：标志寄存器，用于存放指令执行结果的特征。

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

解：状态标志寄存器共有 9 位标志位，其中包含 6 个状态标志位和 3 个控制标志位。

状态标志位：

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

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

AF(Auxiliary carry Flag)：辅助进位标志。运算过程中若 D_3 位向 D_4 位有进位或借位时，AF=1；否则 AF=0。

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

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

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

控制标志位：

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

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

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

7. 8086/8088 系统中存储器的分段原则是什么？

解：段原则是每段的长度最大为 64KB，并要求段的起始地址必须能被 16 整除，形式如 XXXX0H。段内各存储单元地址相对于该段起始单元地址的位移量称为段内偏移量。

分段管理要求每个段都由连续的存储单元构成，并且能够独立寻址，而且段和段之间允许重叠。

8. 当 ALE 有效时，8088 CPU 的地址/数据线上将出现什么信息？

解：当 ALE 有效时，8088 CPU 的地址/数据线上将出现地址信息。

9. READY 引脚的作用是什么？

解：READY 引脚用于在微处理器时序中插入等待状态。若该引脚被置为低电平，则微处理器进入等待状态并保持空闲；若该引脚被置为高电平，则它对微处理器的操作不产生影响。

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

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

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

11. 8088 CPU 工作在最小模式下包含哪些控制信号？

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

12. 若 CS=4000H，则当前代码段可寻址的存储空间范围是多少？

解：当 CS=4000H 时，当前代码段可寻址的存储空间范围为 40000H～4FFFFH。