

新编考研辅导丛书

Principle of Microprocessor & Programming

微机原理与程序设计辅导

楼顺天 王丽娟 编著

重点·难点·考点

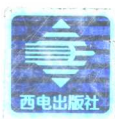
典型题例解析

精选习题与解答

模拟试题与答案

西安电子科技大学出版社

<http://www.xduph.com>



新编考研辅导丛书

微机原理与程序设计辅导

楼顺天 王丽娟 编著

西安电子科技大学出版社

2001

内 容 简 介

本书是根据“微机原理与程序设计”课程硕士研究生入学考试大纲编写而成的，旨在帮助考生在该门课程的考试中获得高分。全书根据大纲要求，详细介绍了该门课程的考点、重点与难点，并以大量例题、习题帮助考生熟悉并掌握各种题型。

本书可作为报考西安电子科技大学硕士研究生的考生的复习辅导书，也可作为“微机原理与系统设计”和相关课程的学习指导书，还可作为计算机等级(三级)考试(偏硬)的备考书。

图书在版编目(CIP)数据

微机原理与程序设计辅导/楼顺天,王丽娟编著.

—西安:西安电子科技大学出版社,2001.7

(新编考研辅导丛书)

ISBN 7-5606-1041-2

I. 微… II. ①楼…②王… III. ①微型计算机—理论—研究生—入学考试—自学参考资料
②程序设计—研究生—入学考试—自学参考资料 N. TP36

中国版本图书馆 CIP 数据核字(2001)第 043305 号

责任编辑 李惠萍 钟宏萍

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)8227828 邮编 710071

http://www.xduph.com E-mail: xdupfxb@pub.xacnline.com

经 销 新华书店

印 刷 西安兰翔印刷厂

版 次 2001年7月第1版 2001年7月第1次印刷

开 本 787毫米×960毫米 1/16 印张 7.5

字 数 143千字

印 数 1~6 000册

定 价 10.00元

ISBN 7-5606-1041-2/TN·0514

* * * 如有印装问题可调换 * * *

本书封面贴有西安电子科技大学出版社的激光防伪标志,无标志者不得销售。

《新编考研辅导丛书》编审委员会

主任委员：	傅丰林	副校长	教授
副主任委员：	焦李成	博士生导师	教授
委员：	刘三阳	博士生导师	教授
	曾兴雯		教授
	孙肖子		教授
	李伯成		教授
	张永瑞		教授

序

人类走过了又一个千年之交。世界正在发生深刻变化。这一变化是20世纪以来科学技术革命不断深入的必然结果，她已经成为推动社会发展与文明进步的革命性力量。人类走过了农业经济时代、工业经济时代，正在进入知识经济时代。

自1978年国家恢复招收研究生和1980年建立学位制度至今，研究生教育已经走过了20多年的历程，她是我国教育结构中最高层次的教育，肩负着为国家现代化培养高素质、高层次创造性人才的重任，是我国增强综合国力、增强国际竞争力的重要支撑力量。研究生教育的改革和发展，直接关系到21世纪我国第三步战略目标的实现。

西安电子科技大学是一所有70年历史的教育部直属的重点高等学校，也是国家“211”重点建设高校，同时又是国家首批具有硕士、博士授予权的单位之一。现有在校生15 000多人，其中研究生2000余人。学校建有研究生院等10个学院，有3个国家重点学科和27个省部级重点学科；同时建有3个国家重点实验室和16个省部级重点实验室，在“通信与信息系统”、“信号与信息处理”、“电路与系统”、“微电子与固体电子学”、“电磁场与微波技术”和“密码学”等领域设有“长江计划”特聘教授岗位。近年来，西安电子科技大学研究生教育得到了迅速的发展，年招生已超过1000人，招生质量和培养质量在省内名列前茅。毕业生遍布国内外，受到了广泛赞誉。

当前，研究生教育面临新的挑战，同时给研究生教育的发展带来了新的机遇。如何选拔优秀人才是一项长期的研究课题。西安电子科技大学出版社组织我校长期在教学科研第一线、在国内有一定知名度的教授编写了这套考研辅导丛书，并从重点、难点、考点、典型例题分析及自测题等方面进行有剖析、对比总结性的阐述，有助于考生在有限的时间内复习所学内容，并有新的提高和启发。

我们相信这套丛书的出版对我国工科电子信息类研究生教育的发展会起到积极的促进作用。

西安电子科技大学研究生院

博士生导师 焦李成

2000年7月

前 言

本书是以西安电子科技大学 2000 年制订的硕士研究生入学考试大纲为依据,针对近年考研命题趋势,在结合编者多年来从事考研辅导所积累的经验 and 资料并参考多种有关书籍的基础上编写而成的。此书旨在帮助考生在较短时间内复习和掌握考试大纲所规定的考试内容,熟悉各种题型,掌握答题方法及解答技巧,取得好的成绩。

全书分六章,每一章以重点难点考点方式给考生以实际指导,复习本章所要求掌握的内容,巩固重点内容,突破难点。在典型例题分析中,以大量例题给考生在答题方面的辅导,通过解题,使考生掌握常见考题的解答方法。在练习题中,给出相应的习题,供考生考前练习。

由于 8255A 是每年必考的重点,而且主要教学参考书《微型计算机原理与应用》(王永山等编著)中略去了这一节,因此我们在附录 A 中详细给出了 8255A 的内容。在附录 B 中给出了近五年的历届硕士研究生入学考试试题。在附录 C 中给出了 2000 年版的研究生入学考试大纲。

由于时间紧迫和作者水平有限,书中可能还有疏忽和错误之处,恳请读者和同行批评指正。

编 者

2001. 6. 5

目 录

第 1 章 计算机中的数制和码制	1
1.1 重点难点考点	1
1.2 典型例题分析	3
1.3 练习题	4
第 2 章 微机系统中的微处理器	7
2.1 重点难点考点	7
2.2 典型例题分析	11
2.3 练习题	13
第 3 章 汇编语言程序设计基本方法	15
3.1 重点难点考点	15
3.2 典型例题分析	25
3.3 练习题	42
第 4 章 总线结构和时序	49
4.1 重点难点考点	49
4.2 典型例题分析	51
4.3 练习题	51
第 5 章 存储器系统	53
5.1 重点难点考点	53
5.2 典型例题分析	55
5.3 练习题	56
第 6 章 输入输出接口(1)	58
6.1 重点难点考点	58
6.2 典型例题分析	66
6.3 练习题	71
附录 A 常用并行接口芯片 8255A	74
A.1 并行接口的基本原理及结构	74
A.2 常用并行接口芯片 8255A 基础	75
A.3 8255A 工作方式选择	77
A.4 8255A 工作方式	78
A.5 读取端口 C 状态	83
A.6 练习题	83

附录 B 历届硕士研究生入学考试试题	85
试题 1: 1997 年硕士研究生入学考试试题	85
试题 2: 1998 年硕士研究生入学考试试题	88
试题 3: 1999 年硕士研究生入学考试试题	92
试题 4: 2000 年硕士研究生入学考试试题	99
试题 5: 2001 年硕士研究生入学考试试题	105
附录 C 研究生入学考试大纲	110

第 1 章 计算机中的数制和码制

本章介绍计算机中数制和码制的基础知识, 主要包括各种进制数的表示法及其相互转换、二进制数的运算、有符号二进制数的表示方法及运算时的溢出问题、实数的二进制表示法、BCD 编码和 ASCII 字符代码等内容。要求重点掌握各种进制数的表示及相互转换、有符号数的补码表示及补码运算。

1.1 重点难点考点

1. 任意进制数的表示

任意一个数 N 可表示成 p 进制数:

$$(N)_p = \sum_{i=-m}^{n-1} k_i p^i$$

其中 $k_i = 0, 1, \dots, p-1$, 数 N 表示成 m 位小数和 n 位整数。

2. 数制之间的变换

十进制到任意进制(设为 p 进制)的变换规则: (1) 整数部分: N 除以 p 取余数; (2) 纯小数部分: N 乘以 p 取整数。

任意进制(设为 p 进制)到十进制的变换规则: 按权展开。

3. 二进制数的算术运算

二进制数的算术运算与十进制数的运算类似, 只是运算规则为: 逢二进一, 借一当二。

4. 二进制数的逻辑运算

二进制数的逻辑运算有四种: AND(与)、OR(或)、NOT(非)、XOR(异或), 其特点是按位运算, 即本位的运算结果不会对其它位产生任何影响。这四种逻辑运算的规则可以表示成:

1 AND 1 = 1	1 AND 0 = 0	0 AND 0 = 0	
1 OR 1 = 1	1 OR 0 = 1	0 OR 0 = 0	
1 XOR 1 = 0	1 XOR 0 = 1	0 XOR 0 = 0	0 XOR 1 = 1
NOT 1 = 0	NOT 0 = 1		

5. 有符号数的原码表示

8 位有符号二进制数可用原码直接表示, 如图 1.1 所示。



图 1.1 8 位有符号二进制数的原码表示

D7=0 表示正数; D7=1 表示负数。D6~D0 表示该数的绝对值。例如 +65 表示成 0100 0001B, 而 -65 表示成 1100 0001B。

6. 有符号数的补码表示

一个数 x 的补码定义为(设机器字长为 n 位):

$$[x]_c = \begin{cases} x & 0 \leq x < 2^{n-1} \\ x + 2^n & -2^{n-1} \leq x < 0 \end{cases} \pmod{2^n}$$

对于任意一个在机器字长能表示的范围内的有符号数, 可分两步得到其补码表示:

(1) 取 x 的绝对值 $x_1 = |x|$, 并将 x_1 表示成补码形式 $[x_1]_c$ 。(2) 如果 x 为正数, 则 $[x]_c = [x_1]_c$; 如果 x 为负数, 则对 $[x_1]_c$ 中的每一位(包括符号位)取反, 再在最低位加 1, 即 $[x]_c = \overline{[x_1]_c} + 1$ 。

7. 补码运算规则

- ① $[x+y]_c = [x]_c + [y]_c \pmod{2^n}$
- ② $[x-y]_c = [x]_c - [y]_c \pmod{2^n}$
- ③ $[x-y]_c = [x]_c + [-y]_c \pmod{2^n}$
- ④ $[x]_c - [y]_c = [x]_c + [-y]_c \pmod{2^n}$

已知 $[y]_c$ 而求 $[-y]_c$ 的过程, 称为变补或求负。其规则为: $[-y]_c = \overline{[y]_c} + 1$, 这可以直接采用 NEG 指令完成。

8. BCD 编码

用 4 位二进制数表示 1 位十进制数的表示方法称为 BCD(编)码。最常用的编码方法是采用 4 位二进制数的前十种组合来表示 0~9, 这种编码方案也称为 8421BCD 码。

BCD 码的引入完全是为了迎合我们日常的十进制计数习惯。但计算机内部只能进行二进制数的运算, 因此, 当让计算机处理 BCD 码时, 应对计算结果进行适当的修正。对加法运算应采用“加 6 修正”, 对减法运算应采用“减 6 修正”, 其规则可总结如下:

① 当两个 BCD 码位相加(相减)无进(借)位时,如果结果小于或等于 9,则该位不需要修正;如果结果大于 9,则该位进行加 6(减 6)修正;

② 如果两个 BCD 码位相加有进(借)位,则该位进行加 6(减 6)修正;

③ 如果低位修正结果使得高位大于 9,则高位进行加 6(减 6)修正。

9. 常用字符的 ASCII 码

应记住的常用字符的 ASCII 码如下:

数字 0~9: 30H~39H; 大写字母 A~Z: 41H~5AH; 小写字母 a~z: 61H~7AH; 空格: 20H; 回车(CR): 0DH; 换行(LF): 0AH; 换码(ESC): 1BH。

1.2 典型例题分析

每年本章的考题约 5 分。题目虽简单,但不可以轻视。

例 1 填空题:

(1) $115.1875 = \underline{\hspace{2cm}} \text{H};$

(2) $4\text{D.AH} = \underline{\hspace{2cm}} \text{D};$

(3) $21.3 = \underline{\hspace{2cm}} \text{B};$

(4) $0110\ 1011\text{B} = \underline{\hspace{2cm}} \text{D};$

(5) $4095 = \underline{\hspace{2cm}} \text{H};$

(6) $12\text{D}4\text{H} = \underline{\hspace{2cm}} \text{D}.$

解: 根据数制之间的变换规则,很容易计算出结果:

(1) $115.1875 = 73.3\text{H};$

(2) $4\text{D.AH} = 77.625\text{D};$

(3) $21.3 = 0001\ 0101.0011\text{B};$

(4) $0110\ 1011\text{B} = 107\text{D};$

(5) $4095 = 0\text{FFFH};$

(6) $12\text{D}4\text{H} = 4820\text{D}.$

例 2 填空题:

(1) 若 $x=76$, $[x]_{\text{原}} = \underline{\hspace{2cm}} \text{H}$, $[x]_{\text{补}} = \underline{\hspace{2cm}} \text{H}$;

(2) 若 $x=-99$, $[x]_{\text{原}} = \underline{\hspace{2cm}} \text{H}$, $[x]_{\text{补}} = \underline{\hspace{2cm}} \text{H}$;

(3) 若 $x=114$, $[x]_{\text{补}} = \underline{\hspace{2cm}} \text{H}$;

(4) 若 $x=-126$, $[x]_{\text{补}} = \underline{\hspace{2cm}} \text{H}$;

(5) 若补 $[x]_{\text{补}} = 8\text{FH}$, $[-x]_{\text{补}} = \underline{\hspace{2cm}} \text{H}$;

(6) 若 $x=-128$, $[x]_{\text{补}} = \underline{\hspace{2cm}} \text{H}$ 。

解: 根据计算补码的规则,很容易计算出结果:

(1) 若 $x=76$, $[x]_{\text{原}} = 4\text{CH}$, $[x]_{\text{补}} = 4\text{CH}$;

(2) 若 $x=-99$, $[x]_{\text{原}} = 0\text{E}3\text{H}$, $[x]_{\text{补}} = 9\text{DH}$;

(3) 若 $x=114$, $[x]_{\text{补}} = 72\text{H}$;

(4) 若 $x=-126$, $[x]_{\text{补}} = 82\text{H}$;

(5) 若 $[x]_{\text{补}} = 8\text{FH}$, $[-x]_{\text{补}} = 71\text{H}$;

(6) 若 $x = -128$, $[x]_{\text{补}} = 80\text{H}$ 。

例 3 下列指令执行后, $(\text{AX}) =$ _____

MOV AL, 07

MOV BL, 08

ADD AL, BL

DAA

解: 根据相加运算的 BCD 码修正规则, 可以得到结果 $(\text{AX}) = 0105\text{H}$

例 4 写出下列数字或字符的 ASCII 码:

(1) Macro Assemble; (2) My number is 318;

(3) $12+78=90$; (4) 0110 1101B。

解: 参照字符的 ASCII 码表, 可以得到结果, 但要求大家能记住常用字符的 ASCII 码, 应该在不查表的情况下完成。

(1) 4DH 61H 63H 72H 6FH 20H 41H 73H 73H 65H 6DH 62H 6CH 65H;

(2) 4DH 79H 20H 6EH 75H 6DH 62H 65H 72H 20H 69H 73H 20H 33H 31H 38H;

(3) 31H 32H 2BH 37H 38H 3DH 39H 30H;

(4) 30H 31H 31H 30H 31H 31H 30H 31H 42H。

1.3 练 习 题

1. 将下列十进制数转换成二进制数:

(1) 49; (2) 73.8125; (3) 79.75。

2. 将二进制数变换成十六进制数:

(1) 101101B; (2) 1101001011B; (3) 111111111111101B;

(4) 100000010101B; (5) 1111111B; (6) 10000000001B。

3. 将十六进制数变换成二进制数和十进制数:

(1) FAH; (2) 5BH; (3) 78A1H; (4) FFFFH。

4. 将下列十进制数转换成十六进制数:

(1) 39; (2) 299.34375; (3) 54.5625。

5. 将下列二进制数转换成十进制数:

(1) 10110.101B; (2) 10010010.001B; (3) 11010.1101B。

6. 计算(按原进制运算):

(1) $10001101\text{B} + 11010\text{B}$; (2) $10111\text{B} + 11100101\text{B}$; (3) $1011110\text{B} - 1110\text{B}$;

(4) $124AH+78FH$; (5) $5673H+123H$; (6) $1000H-F5CH$ 。

7. 已知 $a=1011B$, $b=11001B$, $c=100110B$, 按二进制完成下列运算, 并用十进制运算检查计算结果:

(1) $a+b$; (2) $c-a-b$; (3) $a \cdot b$; (4) c/b 。

8. 已知 $a=00111000B$, $b=11000111B$, 计算下列逻辑运算:

(1) $a \text{ AND } b$; (2) $a \text{ OR } b$; (3) $a \text{ XOR } b$; (4) $\text{NOT } a$ 。

9. 设机器字长为 8 位, 写出下列各数的原码和补码:

(1) $+1010101B$; (2) $-1010101B$; (3) $+11111111B$;

(4) $-1111111B$; (5) $+1000000B$; (6) $-1000000B$ 。

10. 写出下列十进制数的二进制补码表示(设机器字长为 8 位):

(1) 15; (2) -1; (3) 117; (4) 0;

(4) -15; (5) 127; (6) -128; (7) 80。

11. 设机器字长为 8 位, 先将下列各数表示成二进制补码, 然后按补码进行运算, 并用十进制数运算进行检验:

(1) $87-73$; (2) $87+(-73)$; (3) $87-(-73)$;

(4) $(-87)+73$; (5) $(-87)-73$; (6) $(-87)-(-73)$ 。

12. 已知 a, b, c, d 为二进制补码: $a=00110010B$, $b=01001010B$, $c=11101001B$, $d=10111010B$, 计算:

(1) $a+b$; (2) $a+c$; (3) $c+b$; (4) $c+d$;

(5) $a-b$; (6) $c-a$; (7) $d-c$; (8) $a+d-c$ 。

13. 设下列四组为 8 位二进制补码表示的十六进制数, 计算 $a+b$ 和 $a-b$, 并判断其结果是否溢出:

(1) $a=37H, b=57H$; (2) $a=0B7H, b=0D7H$;

(3) $a=0F7H, b=0D7H$; (4) $a=37H, b=0C7H$ 。

14. 求下列组合 BCD 数的二进制和十六进制表示形式:

(1) 3251; (2) 12907; (3) ABCD; (4) abcd。

15. 将下列算式中的十进制数表示成组合 BCD 码进行运算, 并用加 6/减 6 修正其结果:

(1) $38+42$; (2) $56+77$; (3) $99+88$; (4) $34+69$;

(5) $38-42$; (6) $77-56$; (7) $15-76$; (8) $89-23$ 。

16. 将下列字符串表示成相应的 ASCII 码(用十六进制数表示):

(1) Example 1; (2) XiDian University; (3) -108.652;

(4) How are you?; (5) Computer; (6) Internet Web。

17. 将下列字符串表示成相应的 ASCII 码(用十六进制数表示) :

(1) Hello; (2) 123<CR>456; (注: <CR>表示回车) (3) ASCII;

(4) The number is 2315。

第 2 章 微机系统中的微处理器

本章介绍微型计算机系统中的核心部件微处理器(CPU)。通过了解 CPU 的内部和外部结构,理解微处理器级总线(地址总线、数据总线和控制总线)的概念;通过学习 CPU 的功能结构,掌握 CPU 中两个独立单元(执行单元 EU 和总线接口单元 BIU)的并行执行过程;通过介绍 8086 的寄存器结构,学习汇编语言程序设计所需要的 14 个寄存器,掌握这些寄存器的正确使用;通过介绍 8086 的存储器组织与分段、I/O 端口地址空间等基本知识,了解 8086CPU 与外围电路的关系;重点掌握数据的八种基本寻址方式和转移地址的四种寻址方式。

2.1 重点难点考点

1. 微处理器的内部结构

从微处理器(也称中央处理单元,即 CPU)的内部结构,可以理解 CPU 的工作过程,这对掌握汇编语言的编程是很有好处的。

典型的微处理器内部结构可用图 2.1 表示。CPU 主要由四部分组成:(1) 算术逻辑运算单元(ALU): CPU 的核心,完成所有的算术和逻辑运算操作;(2) 工作寄存器:用于暂存寻址信息和中间计算结果;(3) 控制器: CPU 的“指挥中心”,在它的控制下, CPU 才能完成指令的读入、寄存、译码和执行;(4) I/O 控制逻辑:处理 CPU 的 I/O 操作。

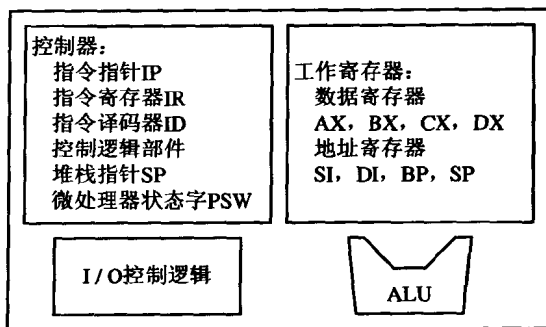


图 2.1 微处理器内部结构图

名词解析：

(1) 程序计数器(PC; Program Counter)：也称作指令指针(IP; Instruction Pointer)，用于保存下一条将要执行的指令的(段内偏移)地址。

(2) 指令寄存器(IR; Instruction Register)：保存从存储器中读入的、当前将要执行的指令。

(3) 指令译码器(ID; Instruction Decode)：完成对指令的译码。

(4) 控制逻辑部件：根据译码结果，产生相应的控制信号，使各部件完成指令指定的动作。

(5) 堆栈指针(SP; Stack Pointer)：用于指示堆栈操作的(段内偏移)地址。

(6) 微处理器状态字(PSW; Processor State Word)：用于寄存 CPU 指令执行后的状态，包括六个状态标志位和三个控制位。

2. 微处理器的外部结构

CPU 的引脚信号通过逻辑部件的处理和组合，构成了系统总线：(1) 数据总线(16 位，注意：对 8088 CPU 而言只有 8 位)：用于传送信息；(2) 地址总线(20 位)：用于传送地址码，可寻址 $2^{20}=1\text{ M}$ 字节空间；(3) 控制总线(16 条)：用于控制各个逻辑部件。

存储单元是指由 RAM 或 ROM 构成的、用于存储指令和数据的区域，每个存储单元为 8 位的二进制数，对应于惟一的一个地址，两个相邻的存储单元可以构成一个字。同样的叙述适用于 I/O 端口寄存器。

在以 8086/8088 CPU 构成的系统中，存储地址空间与端口地址空间分开，采用两个独立的地址空间：存储单元地址采用 A0~A19 编址，端口地址采用 A0~A15 编址。

3. 微处理器的功能结构

在功能上，8086/8088 CPU 由两个独立的逻辑单元组成：执行单元(EU)和总线接口单元(BIU)。EU 用于完成指令所要求的运算操作，而 BIU 用于完成指令地址和数据访问地址的计算，也即由逻辑地址计算出物理地址。这两个单元是独立、并行执行的。

4. 寄存器结构

8086/8088 CPU 内部有 14 个 16 位的寄存器，它们可分为三组：通用寄存器(8 个)，段寄存器(4 个)和控制寄存器(2 个)。

5. 通用寄存器

通用寄存器又可以分为数据寄存器和地址寄存器两种，分别用于存放数据和地址信息，但这种划分不是绝对的。一方面，数据寄存器并不一定存放的就是数据信息；另一方面，数据寄存器并不是惟一能够存放数据的寄存器。这就是说，数据寄存器也可以存放地

址信息，同样，地址寄存器也可以存放数据。然而，一般而言在指令中用作地址的寄存器是地址寄存器。

数据寄存器有四个 16 位的寄存器：AX(Accumulator) 累加器、BX(Base Register) 基址寄存器、CX(Count Register) 计数寄存器和 DX(Data Register) 数据寄存器。它们又可以分成高位和低位字节寄存器使用，因此又得到了八个 8 位的寄存器：AH、AL、BH、BL、CH、CL、DH 和 DL。

地址寄存器有四个 16 位的寄存器：SI(Source Index) 源变址寄存器、DI(Destination Index) 目的变址寄存器、BP(Base Pointer) 基址指针和 SP(Stack Pointer) 堆栈指针。

6. 段寄存器

段寄存器有四个(16 位)：CS(Code Segment) 代码段寄存器、DS(Data Segment) 数据段寄存器、ES(Extra Segment) 附加段寄存器和 SS(Stack Segment) 堆栈段寄存器。它们与地址寄存器一起构成物理地址，即有：物理地址 = 段地址 × 10H + 偏移地址。

7. 物理地址与逻辑地址

逻辑地址的表示形式为：

段地址：偏移地址

其相应的物理地址为：段地址 × 10H + 偏移地址。例如 0800: 01A0 的物理地址为 0800H × 10H + 01A0H = 081A0H。

8. 控制寄存器

控制寄存器有两个(16 位)：IP(Instruction Pointer) 指令指针和 PSW(Processor State Word) 微处理器状态字。

IP 相当于 PC，它用于保存下一条要执行指令的段内偏移地址，IP 的内容不能用普通的传送类指令改变，但可以通过转移、子程序调用/返回、中断调用/返回等指令改变。

PSW 中的标志位定义如图 2.2 所示。其中状态标志位 CF、AF、ZF、SF、OF 和 PF 用于表示上一次 CPU 运算操作的状态，下列情况下标志位为 1：

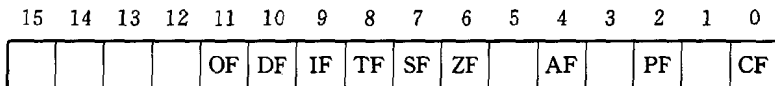


图 2.2 微处理器中定义的标志位

CF(Carry Flag) 进位标志：运算中产生了进位；

AF(Auxiliary Carry Flag) 辅助进位标志：运算中 D3 位产生了进位；

ZF(Zero Flag) 零标志：运算结果等于 0；