

# 深入 DOS/BIOS 编程与应用

● 毛明 何建波 王贵和 等 编著



电子工业出版社

# 深入 DOS/BIOS 编程与应用

毛明 何建波 王贵和等 编著

电子工业出版社

(京)新登字 055 号

## 内 容 提 要

本书从应用的角度出发,汇集了用 Intel 8086 汇编语言和 Turbo C 语言编写的 118 道程序设计实例,内容涉及到数据处理与代码转换、文件处理、微机系统检测与故障诊断、提高系统运行效率、发声与演奏音乐、图形绘制与存取、系统及磁盘数据保护、汉字系统应用、打印机及其它等诸多方面的问题。这些程序设计实例集知识性、技巧性、实用性为一体,不仅能使读者学到许多程序设计的方法和技巧,还能使读者深入地了解 DOS/BIOS 系统中许多重要的内容。

本书适合各行各业从事计算机系统开发和应用的广大技术人员、操作人员学习和使用,也可以作为大专院校计算机专业学生的学习参考。

### 深入 DOS/BIOS 编程与应用

毛明 何建波 王贵和等 编著

责任编辑:吕 迈

\*

电子工业出版社出版

北京市海淀区万寿路 173 信箱(100036)

电子工业出版社发行 各地新华书店经销

电子工业出版社排版室排版

冶金印刷厂印刷

\*

开本:787×1092 毫米 1/16 印张:22.5 字数:573 千字

1995 年 6 月第一版 1995 年 6 月北京第一次印刷

印数: 1—3,000 册 定价:26.00 元

ISBN 7-5053-2925-1/TP. 991

# 前 言

汇编语言是面向机器的语言,它与计算机软、硬件系统的结构有着密切的联系,正因为如此,汇编语言程序设计在系统开发和应用中占据着主导地位,它能解决许多高级语言无法解决的问题,能充分地发挥微型计算机系统的所有功能。

C语言是当今最为流行的程序设计语言,它既有高级语言的优点,又兼有汇编语言的功能,是开发系统软件和应用软件的强有力工具。

由于汇编语言和C语言的广泛应用,人们在进行系统开发和应用时经常需要使用一些工具性的程序或子程序,如:如何将ASCII码转换为二进制数;如何识别显示卡的类型;如何查找汉字的内码;如何在应用程序环境中运行DOS命令等等。如果能将这些实用的工具性程序或子程序整理出来,提供给广大的系统开发和应用人员使用或参考,即可以使他们在解决类似的问题时,避免查阅资料和重复编写程序而浪费宝贵的时间和精力,高效率地完成复杂的程序设计任务。正是出于这一目的,作者根据自己在教学和科研工作中的体会和经验,参考了一些有关文献,整理、编写了汇编语言和C语言在微型计算机系统开发和应用方面的100多道实用的工具性程序,奉献给广大读者。这些工具性的源程序,内容涉及到微型计算机系统的许多方面,为了便于阅读,我们将功能上具有共同点的程序划分为一个类型,共九大类型。每一类型有一个标题,在每一个标题下面将属于同一类的程序以问答的形式例出。所有的程序都分为“程序说明”和“程序清单”两个部分来编写,其中“程序说明”包括对本程序的功能描述、程序设计思想的描述以及必要的微型计算机系统的基本知识的介绍等,而“程序清单”则是具体的源程序清单。在源程序清单中,部分程序以子程序(过程)的形式编写,而大多数的程序都以完整的汇编语言源程序或C语言源程序的形式编写,并且为了便于读者对程序设计中具体环节的理解,程序清单中也有详细的注释和说明。

本书中所有的汇编语言源程序和C语言源程序分别在MASM 5.0和Turbo C 2.0下编译通过。这些实例集知识性、技巧性与实用性为一体,学习完这些源程序,不仅能了解DOS/BIOS系统中许多重要的内容,而且许多具体的问题也会迎刃而解,大大地提高工作和学习效率,这就是作者编写此书的目的和希望。

本书由毛明、何建波、王贵和主编,郑秀林、周长春、任秀英、武一平、林改莲、毛梅英、薛仁青、宁殿中、刘萍、李杰等同志编写并调试了部分程序。全书由朱继生同志审阅。

由于水平所限,经验不足,书中难免存在错误和不妥之处,敬请各位读者批评指正。

毛明

1994年12月

于北京电子科技学院

# 目 录

一、数据处理与代码转换 .....	(1)
1 如何实现多位二进制加法 .....	(1)
2 如何实现多位二进制减法 .....	(2)
3 如何实现多位二进制乘法 .....	(4)
4 如何实现多位二进制除法 .....	(6)
5 如何将 BCD 码转换为二进制数 .....	(8)
6 如何将二进制数转换为 BCD 码 .....	(9)
7 如何将 ASCII 码转换为 BCD 码 .....	(10)
8 如何将 BCD 码转换为 ASCII 码 .....	(11)
9 如何将二进制数转换为 ASCII 码 .....	(12)
10 如何将 ASCII 码转换为二进制数 .....	(13)
11 如何将十六进制数转换为十进制数 .....	(14)
12 如何将十进制数转换为十六进制数 .....	(17)
13 如何实现冒泡排序 .....	(20)
14 如何实现快速排序 .....	(22)
二、文件处理 .....	(27)
15 如何将文件中的小写字母变为大写字母 .....	(27)
16 如何清除文本文件行尾的空格 .....	(30)
17 如何将十六进制代码格式文件转换为 ASCII 码格式文件 .....	(33)
18 如何分割十六进制代码文件 .....	(38)
19 如何连接十六进制代码文件 .....	(40)
20 如何给 EXE 文件添加程序 .....	(45)
21 如何将 COM 文件转换为 EXE 文件 .....	(51)
22 如何将 BAT 文件转换为 EXE 文件 .....	(56)
23 如何用 C 语言修改数据库记录 .....	(57)
24 如何查找文件所在的目录 .....	(61)
25 如何方便地进行文件的阅读和打印 .....	(63)
26 如何实现同一驱动器下文件的直接拷贝 .....	(67)
27 如何方便地修改文件属性 .....	(70)
28 如何增强 COPY 命令的功能 .....	(74)
29 如何增强 TYPE 命令的功能 .....	(80)
三、微机系统检测与故障诊断 .....	(82)
30 如何判别微型计算机的类型 .....	(82)
31 如何判别 CPU 的类型 .....	(85)

32	如何判别显示卡的类型 .....	(87)
33	如何检测键盘的类型 .....	(90)
34	如何检测鼠标是否安装 .....	(91)
35	如何确定 DOS 及 BIOS 的版本 .....	(92)
36	如何检测内存容量 .....	(92)
37	如何检测软盘驱动器的类型 .....	(94)
38	如何检测硬盘驱动器的类型 .....	(95)
39	如何测定高档微机硬盘的物理参数 .....	(96)
40	如何测定并行口和串行口的个数 .....	(100)
41	如何判断 PC 机串行口是否正常 .....	(101)
42	如何检测针式打印机的断针位置 .....	(104)
43	如何检测单色显示器的各种字符属性 .....	(110)
44	如何检测彩色显示器的各种字符属性 .....	(112)
45	如何检测 VGA 显示卡的 256 种颜色 .....	(114)
46	如何检测 RAM 存储器的故障位置 .....	(117)
47	如何检测软盘驱动器 .....	(122)
<b>四、提高系统运行效率 .....</b>		<b>(126)</b>
48	如何在字符方式下快速显示字符 .....	(126)
49	如何在长城 0520C-H 机上快速显示汉字 .....	(128)
50	如何控制光标的移动速度 .....	(131)
51	如何清除内存驻留程序 .....	(133)
52	如何扩充键盘缓冲区的容量 .....	(136)
53	如何对键盘功能键进行重定义 .....	(139)
54	如何使 F11、F12 功能键发挥作用 .....	(141)
55	如何在当前运行的程序中加载另一个程序 .....	(143)
56	如何在应用程序的环境中运行所有 DOS 命令 .....	(146)
<b>五、发声与演奏音乐 .....</b>		<b>(149)</b>
57	如何在程序运行中用发声进行报警 .....	(149)
58	如何模拟枪声 .....	(151)
59	如何模拟整点报时声音 .....	(152)
60	如何演奏一首乐曲 .....	(154)
61	如何用键盘弹奏乐曲 .....	(158)
62	如何在其它操作进行的同行演奏音乐 .....	(160)
63	如何发出电话振铃声 .....	(164)
64	如何用 C 语言编写奏乐语句 .....	(167)
<b>六、图形绘制与存取 .....</b>		<b>(171)</b>
65	如何利用键盘绘图 .....	(171)
66	如何画斜线 .....	(173)
67	如何显示和移动字幕 .....	(177)
68	如何保存和恢复 CGA 方式下的屏幕图像 .....	(181)

69	如何保存和恢复长城 0520C-H 高分辨率下的屏幕图像	(183)
70	如何保存和恢复 EGA/VGA 高分辨率下的屏幕图像	(186)
71	如何将屏幕内容进行文件转储	(193)
72	如何使 HGC 卡仿真 CGA 卡	(196)
73	如何设计弹出式菜单	(197)
74	如何设计下拉式菜单	(206)
75	如何设计立体投影窗口	(216)
76	如何实现推镜头的显示效果	(219)
77	如何巧妙地实现屏幕图像的动画显示	(221)
<b>七、系统及磁盘数据保护</b>		<b>(224)</b>
78	如何保存与恢复 CMOS 数据	(224)
79	如何自动记录微机的运行过程	(227)
80	如何备份和恢复硬盘重要的数据区	(230)
81	如何加/解密数据文件	(234)
82	如何实现对文件首簇号的加密	(238)
83	如何加密 dBASE III 的源程序文件	(241)
84	如何加密 dBASE III 数据库文件	(244)
85	如何解 WPS 密码文件	(245)
86	如何解用 P 格式加密的 BASIC 程序	(250)
87	如何加密批处理文件	(253)
88	如何对驱动器和打印机进行软加锁	(255)
89	如何给硬盘加锁	(258)
90	如何给键盘加锁	(259)
91	如何简单地检测计算机病毒	(262)
92	如何消除硬盘主引导扇区的病毒	(265)
93	如何消除“小球”病毒	(268)
94	如何消除“雨点”病毒	(274)
<b>八、汉字系统应用</b>		<b>(279)</b>
95	如何使 CCDOS 下的光标闪烁	(279)
96	如何快速输入汉字制表符	(281)
97	如何获取汉字内码、国标码和区位码	(284)
98	如何获取 24×24 汉字点阵打印字模	(288)
99	如何将汉字显示字库装入扩展内存	(292)
100	如何巧用长城系列微机键盘上特定的功能键	(297)
101	如何由汇编语言向批处理文件传递参数	(299)
102	如何自动识别长城 0520C-H 机的显示状态	(301)
103	如何用 C 语言处理彩色汉字的输入/输出问题	(303)
104	如何在纯西文方式下显示汉字	(307)
105	如何放大显示汉字	(310)
106	如何将 XE 制表符转换成 WPS 制表符	(314)

九、打印机控制与其它 .....	(317)
107 如何将 LQ-1600K 的“¥”改为“\$” .....	(317)
108 如何驱动 LQ-1600K 打印各种字体的汉字 .....	(318)
109 如何灵活方便地向打印机发送控制命令 .....	(321)
110 如何使打印机以中断方式工作 .....	(323)
111 如何驱动打印机进行图像打印 .....	(324)
112 如何显示实时时钟 .....	(330)
113 如何用 C 语言实现中断驻留和热键激活 .....	(333)
114 如何实现 INT 25H 两种调用格式的自动转换 .....	(335)
115 如何用 C 语言直接执行二进制代码 .....	(338)
116 如何打印任一年的年历 .....	(339)
117 如何用软件实现 RESET 启动和热启动 .....	(346)
118 如何用程序模拟按键 .....	(348)



# 一、数据处理与代码转换

程序设计的基本任务是进行数据处理,数据处理的具体内容包括数据运算、存储、检索、排序、输入和输出等操作。在数据处理过程中,往往由于不同的应用目的,需要将数据按不同的编码格式进行编码和存储,在微型计算机中常用的数据编码有二进制编码、BCD 码和 ASCII 码。二进制编码是可以直接进行算术运算的编码,汇编语言中的四种基本运算指令 ADD、SUB、MUL 和 DIV 都可以单独地直接对二进制数进行运算,其结果还是二进制数。BCD 码是用 4 位二进制数表示一位十进制数的编码,其编码范围为 0000B~1001B,分别表示十进制数的 0~9。BCD 码是为了便于在某些场合中,将数据按十进制数进行运算、存储和输出所设定的一种数据格式编码。ASCII 码是用于数据的输入和输出的一种数据编码格式,即主机与计算机外部设备进行信息交换的一种数据编码格式。由于计算机在对某一具体的数据处理过程中往往包括数据的输入、存储、运算、输出等操作,在这些操作中,每一种操作所需要的数据格式往往不同于其它操作所需要的数据格式。例如输入的数据是 ASCII 码,而存储时可能要转换为二进制编码格式以便于运算,而运算的最后结果要输出,则又需要按 ASCII 码格式来表示。上述的过程表明,在数据处理过程中,往往要进行数据格式之间的转换,以适应于不同的数据处理操作。我们将两种数据格式之间的转换问题称之为代码转换。常用的代码转换有 BCD 码转二进制,二进制转 BCD 码;ASCII 码转 BCD 码,BCD 码转 ASCII 码;二进制转 ASCII 码,ASCII 码转二进制等。

本章选取了数据处理和代码转换方面最常用的一些实例,为便于用户直接在自己的源程序中引用这些实例,大多数的实例都以过程(子程序)的形式给出,少数实例以完整的源程序形式给出。

## 1 如何实现多位二进制加法

程序名	MBINADD. ASM	程序类别	汇编语言子程序
功能	实现以字为单位的多位二进制数的加法		
使用说明	入口参数:DS,SI→加数地址,DS,DI→被加数地址 出口参数:DS,BX→和的存放地址		

### (1) 程序说明

本程序具有将两个多位二进制数进行加法运算的功能。程序在进行加法运算时,每一次将加数与被加数对应的一个字进行加运算。多位数的长度由常数 COUNT(以字为单位)控制,所有三个数都含有  $16 \times \text{COUNT}$  个位,存放在 COUNT 个字长度的内存单元中。

(2) 程序清单

```

MBINADD    PROC    NEAR
            PUSH    SI
            PUSH    DI
            PUSH    BX
            PUSH    CX
            PUSH    AX
            MOV     CX,COUNT    ;加法运算的字计数值→CX
            CLC                ;清进位标志
MBINADD1:
            MOV     AX,[SI]     ;第一个数
            INC     SI
            INC     SI
            ADC     AX,[DI]     ;加上第二个数
            INC     DI
            INC     DI
            MOV     [BX],AX     ;存储中间结果
            INC     BX
            INC     BX
            LOOP   MBINADD1    ;循环
            POP     AX
            POP     CX
            POP     BX
            POP     DI
            POP     SI
            RET
MBINADD    ENDP
    
```

## 2 如何实现多位二进制减法

程序名	MBINSUB.ASM	程序类别	汇编语言子程序
功能	实现以字为单位的多位二进制数的减法		
使用说明	入口参数:DS,SI→被减数,DS,DI→减数,DS,BX→差的存放地址 出口参数:DS,BX→差的存放地址		

### (1) 程序说明

本程序具有将两个多位二进制数进行减法运算的功能。程序在进行减法运算时,每一次将被减数与减数对应的一个字进行减运算。多位数的长度由常数 COUNT(以字为单位)控制,所有三个数都含有  $16 \times \text{COUNT}$  个位,存放在 COUNT 个字长度的内存单元中。

### (2) 程序清单

```
MBINSUB    PROC    NEAR
            PUSH    SI
            PUSH    DI
            PUSH    BX
            PUSH    CX
            PUSH    AX
            MOV     CX,COUNT    ;减法运算的字长度计数→CX
            CLC                ;清进位标志
MBINSUB1:
            MOV     AX,[SI]    ;取被减数的一个字
            INC     SI
            INC     SI
            SBB    AX,[DI]    ;减去减数的对应字
            INC     DI
            INC     DI
            MOV     [BX],AX    ;存差
            INC     BX
            INC     BX
            LOOP   MBINSUB1    ;循环
            POP     AX
            POP     CX
            POP     BX
            POP     DI
            POP     SI
            RET
MBINSUB    ENDP
```

### 3 如何实现多位二进制乘法

程序名	MBINMUL. ASM	程序类别	汇编语言子程序
功能	实现以字为单位的多位二进制数的乘法		
使用说明	入口参数: DS: SI→乘数, DS: DI→被乘数, DS: BX→积的存放地址 出口参数: DS: BX→积的存放地址		

#### (1) 程序说明

本程序能够完成两个多位的二进制数的乘法运算。程序的设计思想完全模仿手工计算的过程,即取乘法的一个字(从低字开始取),乘以被乘数的各个字,保存部分积;再取乘数中较高的一个字,乘以被乘数的各个字,将本次的部分积向高地址方向移动一个字,加上前面运算结果的部分积;……,如此过程,直到将乘数中最高的一字乘完被乘数的各个字,并将最后一次的部分积向高地址方向移动一个字,加上前面所有积的结果,形成多位二进制数的积。

多位数的长度由常数 COUNT(以字为单位)控制,输入的两个数各有  $16 \times \text{COUNT}$  位,输出数精度加倍。输入存放在 COUNT 个字长度的内存单元中,输出存放在  $2 \times \text{COUNT}$  个字长度的内存单元中。

#### (2) 程序清单

```

MBINMUL    PROC    NEAR
            PUSH    SI
            PUSH    DI
            PUSH    BX
            PUSH    CX
            PUSH    AX
            PUSH    BX                ;保存结果单元首地址
            MOV     AX,0
            MOV     CX,2 * COUNT    ;乘积的字长度→CX
            CLD

MBINMUL1:
            MOV     [BX],AX          ;初始化乘积单元
            INC     BX
            INC     BX
            LOOP   MBINMUL1
            POP     BX                ;恢复结果单元首地址
            MOV     CX,COUNT         ;被乘数的字长度→CX

MBINMUL2:

```

```

                PUSH    CX
                MOV     DX,[SI]          ;取乘数的一个字
                INC     SI
                INC     SI
                PUSH   BX
                PUSH   DI
                MOV     CX,COUNT
MBINMUL3:
                PUSH   CX
                PUSH   DX
                MOV     AX,[DI]          ;取被乘数的一个字
                INC     DI
                INC     DI
                MUL     DX              ;进行乘法运算
                ADD     [BX],AX         ;处理本次积的低字
                INC     BX
                INC     BX
                ADC     [BX],DX         ;处理本次积的高字
                POP     DX
                POP     CX
                LOOP   MBINMUL3        ;循环,用被乘数的每一字乘以乘数的当前字
                POP     DI
                POP     BX
                INC     BX              ;修改乘数指针
                INC     BX
                POP     CX
                LOOP   MBINMUL2        ;循环,准备用乘数的下一字乘以被乘数的每一字
                POP     AX
                POP     CX
                POP     BX
                POP     DI
                POP     SI
                RET
MBINMUL        ENDP

```

## 4 如何实现多位二进制除法

程序名	MBINDIV. ASM	程序类别	完整的汇编语言程序
功能	实现以字为单位的多位二进制数的除法		
使用说明	入口参数: DIVIDED→被除数, DIVISOR→除数, REMAIND→商 出口参数: DIVIDED→余数, REMAIND→商		

### (1) 程序说明

本程序能够完成两个多位二进制数的除法运算。程序设计的基本思想是,用被除数减去除数的方法实现除法运算。运算开始时,首先比较被除数与除数的大小,如果被除数小于除数,则返回 DOS,错误号为 01。如果被除数大于除数,则从被除数中减去除数,并将商加 1,然后循环,直到被除数小于除数时结束运算。这样,原来存放被除数的单元中得到的是余数,而商则存放在 REMAIND 开始的内存单元中,所有的数都按低字在低地址,高字在高地址方式存放。为便于用户直接应用,本例以源程序的形式给出。

多位数的长度由常数 COUNT(以字为单位)控制。被除数存放在 DIVIDED 开始的内存单元中,除数存放在 DIVISOR 开始的内存单元中,商存放在 REMAIND 开始的内存单元中。被除数、除数、余数及商的长度都是 COUNT 个字。

### (2) 程序清单

```

CODE          SEGMENT
               ASSUME  CS:CODE,DS:CODE,ES:CODE

COUNT        EQU     2                ;被除数、除数的长度(以字为单位)
DIVIDED        DW     60000,1          ;被除数
DIVISOR        DW     50000,0          ;除数
REMAIND        DW     0,0              ;商
START:        PUSH    CS
               PUSH    CS
               POP     DS
               POP     ES
               MOV     SI,OFFSET DIVIDED
               MOV     DI,OFFSET DIVISOR
               MOV     BX,OFFSET REMAIND
               CALL    DIVCMP           ;比较被除数和除数
               JAE     NEXT            ;被除数大于等于除数时,转
               MOV     AX,4C01H        ;错误返回
               INT     21H
    
```

```

NEXT:    CALL    DIVSUB          ;从被除数中减去除数
         PUSH    BX
         MOV     CX,COUNT
         STC                                ;置进位标志,表示当前商要加 1
CACULA:  ADC     WORD PTR [BX],0
         INC     BX
         INC     BX
         LOOP   CACULA          ;循环处理商的进位
         POP     BX
         CALL   DIVCMP          ;比较被除数与除数的大小
         JAE    NEXT           ;被除数大于等于除数时转
         MOV    AX,4C00H       ;正确返回 DOS
         INT    21H
         ;比较字符串子程序
DIVCMP   PROC    NEAR
         PUSH   SI
         PUSH   DI
         PUSH   CX
         ADD    SI,2 * COUNT-2 ;SI 指向被除数的最后一个字
         ADD    DI,2 * COUNT-2 ;DI 指向被除的最后一个字
         MOV    CX,COUNT       ;比较的字长度
         STD    ;反向比较(从高地址到低地址方向)
         REPZ   CMPSW
         POP    CX
         POP    DI
         POP    SI
         RET
DIVCMP   ENDP
         ;字符串相减子程序
DIVSUB   PROC    NEAR
         PUSH   SI
         PUSH   DI
         PUSH   CX
         CLC
         MOV    CX,COUNT       ;相减的字计数值
DIVSUB1: MOV    AX,[DI]         ;取除数的一个字→AX
         INC    DI
         INC    DI
         SBB   [SI],AX        ;减去被除数中的对应字(考虑借位)
         INC    SI

```

```

                INC      SI
                LOOP    DIVSUB1
                POP     CX
                POP     DI
                POP     SI
                RET
DIVSUB          ENDP
CODE           ENDS
                END     START

```

## 5 如何将 BCD 码转换为二进制数

程序名	BCDTOBIN. ASM	程序类别	汇编语言子程序
功能	将 AX 寄存器中的 4 位 BCD 码转换成二进制数		
使用说明	入口参数: AX → BCD 码, 出口参数: AX → 二进制数 注: 引用该子程序时, 数据段必须有存放权值的单元: W10		

### (1) 程序说明

本程序具有将 AX 中的四位 BCD 码转换成二进制数的功能, 转换的结果放在 AX 中。

由于一位 BCD 码用 4 位二进制数表示, 这样一个字长的寄存器中最多能存放 4 位的 BCD 码。例如, AX=9827H, 如果被看作为 BCD 码, 则其代表的数为九千八百二十七。

将 AX 中的 4 位 BCD 码转换为二进制数的过程可以用下述公式来描述:

$$((\text{千位数} \times 10 + \text{百位数}) \times 10 + \text{十位数}) \times 10 + \text{个位数}$$

引用该子程序时, 数据段中必须有下列存放权值的单元:

```
W10    DW    10    ;十进制数权值
```

### (2) 程序清单

```

AXBCD202 PROC NEAR
                PUSH    BX
                PUSH    CX
                PUSH    DX
                MOV     BX, AX        ;保存 AX 中的 BCD 码到 BX
                MOV     AX, 0        ;结果单元清零
                MOV     CX, 4        ;共处理四位 BCD 码
RETRY:         PUSH    CX
                MOV     CL, 4
                ROL     BX, CL       ;一位 BCD 码移到 BX 中的低半字节

```



```

POP      CX
MUL     W10      ;累加和乘以权值送 AX
PUSH    BX
AND     BX,0FH   ;屏蔽 BX 的高半字节
ADD     AX,BX    ;累加下一位 BCD 码
POP     BX
LOOP    RETRY
POP     DX
POP     CX
POP     BX
RET

```

AXBCD202 ENDP

## 6 如何将二进制数转换为 BCD 码

程序名	BINTOBCD.ASM	程序类别	汇编语言子程序
功能	将 AX 中的二进制数转换为 BCD 码		
使用说明	入口参数:AX→二进制数,出口参数:AX→BCD 码 注:引用该子程序时,数据段必须有存放权值的单元:W1000		

### (1) 程序说明

本程序具有将 AX 中的二进制数转换成四位 BCD 码的功能,转换的结果放在 AX 中(设 AX 中的数值小于十进制数 10000)。

程序设计的思想是,首先将 AX 中的二进制数除以 1000 得到的商即是千位上的 BCD 码,将所得余数再除以 100 得到的商即百位上的 BCD 码,然后再将所得的余数除以 10 得到的商即是十位上的 BCD 码,最后所得的余数是个位上的 BCD 码。

引用该子程序时,数据段中必须有下列存放权值的单元:

W1000    DW    1000,100,10,1    ;十进制数千,百,十,个位上的权值

### (2) 程序清单

```

AX2TOBCD PROC        NEAR
                      PUSH     SI
                      PUSH     BX
                      PUSH     CX
                      PUSH     DX
                      XOR      BX,BX                ;BCD 码暂存单元清零
                      MOV      SI,OFFSET W1000     ;权值首地址送 SI

```