DOS功能调用 （教材第319页）

常用的DOS功能有5个：

键盘输入1个字符：　　01号DOS功能调用

显示器输出1个字符：　02号DOS功能调用

键盘输入缓冲区：　　0AH号DOS功能调用

显示字符串：　　09号DOS功能调用

返回DOS控制：　　4CH号DOS功能调用

1．单字符的输入输出（教材第320页）

（1）1号功能键盘输入

格式：**AH=1**

INT 21H

功能：从键盘输入一个字符并将该字符的**ASC**Ⅱ码送入**AL**中。

（2）2号功能显示器输出（教材第335页）

格式：**AH=2**

DL=字符

INT 21H

功能：输出**DL**中的一个字符到显示器的光标处。

例1 从键盘输入一个字符后，接着再显示出来

```
MOV  AH，1
INT   21H
MOV  DL，AL
MOV  AH ，2
INT   21H
```

- 键盘输入的大写字母换成小写字母显示
- 问题：如何转换
- 提示：**ASCII码**

ASCII表

十六进制	字符	十六进制	字符	十六进制	字符
30	0	49	I	61	a
31	1	4A	J	62	b
32	2	4B	K	63	c
33	3	4C	L	64	d
34	4	4D	M	65	e
35	5	4E	N	66	f
36	6	4F	O	67	g
37	7	50	P	68	h
38	8	51	Q	69	i
39	9	52	R	6A	j
41	A	53	S	6B	k
42	B	54	T	6C	l
43	C	55	U	6D	m
44	D	56	V	6E	n
45	E	57	W	6F	o
46	F	58	X	70	p
47	G	59	Y	71	q
48	H	5A	Z	72	r

大小写转换问题

- **ASCII码：61H→a　　62H →b**
- 改变一个字母的大小写，就是改变它所对应的ASCII 码

A → 41H　a → 61H

大写	二进制	小写	二进制
A	01000001	a	01100001
B	01000010	b	01100010
C	01000011	c	01100011
D	01000100	d	01100100

大小写转换的问题

- 对比：小写字母的ASCII码值比大写字母的ASCII码值大20H。

  因此，如果将 “a” 的ASCII码值减去20H，就可以得到“A”；如果将“A”的ASCII码值加上20H 就可以得到“a”

- 例2 键盘输入的大写字母换成小写字母显示

```
MOV  AH , 1
INT  21H
ADD  AL，20H；大写转换为小写
MOV  DL ，AL
MOV  AH，2
INT  21H
```

2．键盘输入字符串（教材第321页）

格式：**AH=10**

DS:DX=字节缓冲区首址

INT 21H

说明：定义缓冲区的第1个字节单元为允许输入的最大字符数，第2个单元为实际键入个数（由系统自动填入），从第3个单元开始存放键入字符。

功能：从键盘输入一串**ASCII**字符到缓冲区，用“回车”结束输入。若输入字符超过缓冲区能容纳的个数，则系统忽略此字符并响铃警告。

- 例　设置缓冲区，允许从键盘输入10个字符。

```
BUFFER  DB  10，？，10 DUP(?)
……
MOV  AX，SEG BUFFER
MOV  DS，AX
MOV  DX，OFFSET BUFFER
MOV  AH，10
INT   21H
```

执行结果：例如从键盘输入 **Hello↓**（回车）

缓冲区存储情况：

10	5	48	65	6c	6c	6f	0d	…

3．显示字符串（教材第335页）

格式：**AH=9**

DS:DX=字符串地址

INT 21H

功能：显示一个以“$”结尾的ASCⅡ码字符串

- 例　**DISPLAY DB ‘Very Good!’ , ‘$’**

```
        .....
        MOV  AX，SEG DISPLAY
        MOV  DS，AX
        LEA  DX，DISPLAY
        MOV  AH，9
        INT  21H
```

- 屏幕上显示出：**Very Good!**

十进制数加减运算

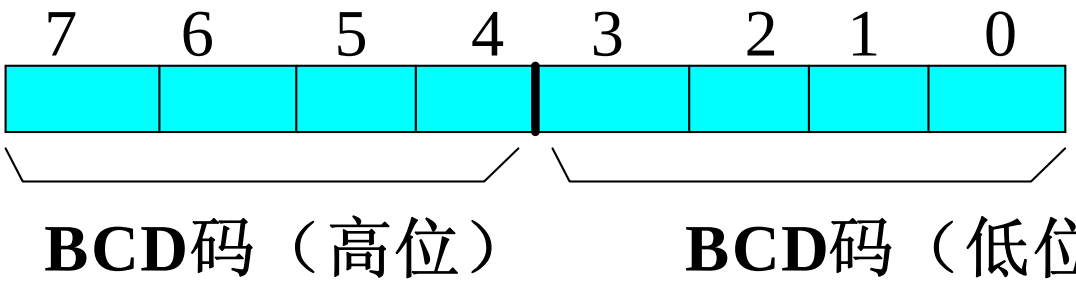
在计算机中采用**BCD**码来表示十进制数。**BCD**码就是使用四位二进制数表示一位十进制数。

在**8086/8088**系统中，将**BCD**码分为两种格式：

- 压缩型（组合型、装配型、**PACKED**）
- 非压缩型（非组合型、拆散型、**UNPACKED**）

压缩型：

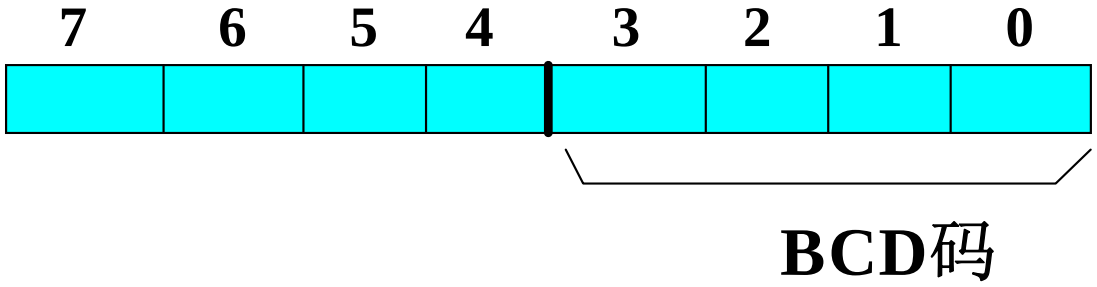
一个字节表示两个BCD码，即两位十进制数



例如：**0010 0011** 表示十进制数的**23**

非压缩型：一个字节的低四位表示一个BCD码，而高四位对所表示的十进制数没有影响。

常为0000B或0011B。



例如：0000 1001与0011 1001都是十进制数9的非组合型的BCD码

BCD码

BCD码：

用4位二进制数表示十进制数的编码方法

数码	0	1	2	3	4
BCD码	0000	0001	0010	0011	0100
数码	5	6	7	8	9
BCD码	0101	0110	0111	1000	1001

- 比如：
  数值26，用BCD码表示为：

  0010 0110

- 可见，一个字节可表示两个BCD码，存储两个 BCD码表示两位十进制数，高 4 位的BCD码表示十位，低4 位的BCD 码表示个位。

  比如：0001 0100b表示14。

- BCD 码值＝十进制数码值，则BCD码值＋30h＝十进制数对应的ASCII码。

在计算机中直接实现十进制数的运算有两种方法:

1. 数制转换: 先把十进制数转换为二进制数,然后用计算机中的二进制运算指令进行运算。最后将结果由二进制数转换为十进制数。

2. 直接进行十进制数运算：使用计算机中的**BCD**码指令进行运算。

修正调整

- 例如：5+7=12
  - BCD码表示：0101+0111
- 结果是：1100（BCD码？）
- BCD码的12应该是：0001 0010
- 1100→0001 0010（加6）
- 原因：逢十进一与逢五进一的差别
- 解决：只要结果大于9，就应该调整

1、压缩的BCD码加法调整

格式：**DAA**

功能：

如果**AL**的低**4**位大于**9**，则将**AL**加**6**，并将辅助进位标志**AF**置**1**

如果**AL**的高**4**位大于**9**，将**AL**加**60H**，并将进位标志**CF**置**1**

在执行**DAA**指令前，必须是用**ADD**或**ADC**完成了加法操作，且加的结果放在**AL**中。

例 十进制计算5＋7=12，用BCD码表示做计算。

```
        X  DB  05H
        Y  DB  07H
        ……
        MOV  AL，X
        ADD  AL，Y
；相加后，(AL)=00001100=0CH
        DAA
；加6调整后，(AL)=00010010=12H（压缩的BCD码）
```

2．压缩的BCD码减法调整

格式：**DAS**

功能：

如果**AL**的低**4**位大于**9**，则将**AL**减**6**，并将**AF**置**1**

如果**AL**的高**4**位大于**9**，将**AL**减**60H**，并将**CF**置**1**

例　十进制计算62－38=24

```
W1  DB  62H        ；BCD码表示的十进制62
W2  DB  38H
……
MOV  AL，W1
SUB  AL，W2        ；相减后，(AL)=2AH
DAS                ；减6调整后，(AL)=24H
```

3.非压缩BCD码加法调整

格式：AAA

功能：

如果AL的低4位大于9，将AL加6、AH加1，AL的高4位清零、CF、AF置1

由于非压缩的BCD码用1个字节表示1个十进制数，所以调整后若加上30H就是该数值的ASCII码。

在AAA指令执行前，必须是使用ADD或ADC指令完成了加法，且结果是在AL中。AAA指令对AL中内容进行校正。

- 思考：十进制计算6+8=14，用非压缩的BCD码表示并显示在屏幕上。

例　十进制计算6+8=14，用非压缩的BCD码表示并显示在屏幕上。

```
T1  DB  06H
T2  DB  08H
……
MOV    AL，T1                    ；(AL)=00000110=06H
ADD    AL，T2                    ；(AL)=00001110=0EH
AAA                        ；调整后(AH)=01H，(AL)=04H
ADD    AX，3030H          ；AH、AL分别加上30H，变成ASCII码
MOV    BX，AX                    ；用BX保存
MOV DL，BH                 ；显示“1”
MOV  AH，2                       ；2号显示功能
INT      21H                     ；DOS中断调用
MOV  DL，BL                ；显示“4”
INT      21H
```

4．非压缩的BCD码减法调整

格式：AAS

功能：

如果AL的低4位大于9，将AL减6、AH减1，AL的高4位清零、CF、AF置1

例　十进制计算57－18=39，用非压缩的BCD码表示。

```
MOV AX,0507H
MOV BX,0108H
SUB AL,BL
SUB AH,BH          ；高位不用带借位减
  AAS              ；减法调整后
                   (AX)=0309H
```

5．非压缩的BCD码乘法调整

格式：**AAM　(ASCII Adjust Multiply)**

功能：

将乘积**AX**中的**2**个非压缩的**BCD**码调整。

AL除以**0AH**，得到的商送**AH**，余数送入**AL**。即乘积的高位数在**AH**、低位数在**AL**中。

- 思考：十进制乘法6×8=48，用非压缩的BCD码表示，并显示

例 十进制乘法6×8=48，用非压缩的BCD码表示，并显示。

```
P1  DB  06H
P2  DB  08H
……
MOV AL，P1    ；(AL)=00000110=06H
IMUL  P2      ；(AL)=00110000=30H
AAM           ；调整后(AH)=04H，(AL)=08H
ADD  AX，3030H        ；AH、AL分别加上30H
MOV BX，AX    ；用BX保存
MOV DL，BH    ；显示“4”
MOV  AH，2
INT 21H
MOV DL，BL    ；显示“8”
INT 21H
```

6．非压缩的BCD码除法调整

格式：**AAD　(ASCII Adjust Division)**

功能：在做除法之前，将被除数**AX**中的2个非压缩的**BCD**码调整。

(AL)=(AL)+(AH)*10，**AH**清零。除法之后，商在**AL**、余数在**AH**中。

实例：带显示的算术程序

简化的程序结构

简化的段定义结构用于小规模的程序设计中。程序有一个代码段、一个数据段，每段不大于64KB。堆栈段、附加段和数据段共用。因此，小规模的程序最大不能超过128KB。

使用简化段结构便于汇编语言模块与高级语言模块的连接。

实验示例

示例4-9 从键盘输入两个一位的十进制数，做乘法运算。相乘的结果保存在存储单元x中，算式显示在屏幕上。用简化的程序格式。

设计思路：

（1）用DOS中断调用的1号功能输入数据，用2号功能显示结果，9号功能显示提示信息；

（2）做乘法时必须将输入数字的ASCII码去掉，转换成数值；

（3）乘法之后用十进制调整指令AAM。

运行结果：

```
D:\MASM>prog4
input:8*4=32
D:\MASM>prog4
input:6**=F0
D:\MASM>prog4
input:6*0=00
D:\MASM>_
```

参考程序：

```
;prog4.asm
.model small
.data
    x db ?,?
    infor db 'input:','$'
.stack 100h
.code
start:
   mov ax, @data
   mov ds, ax
   mov dx, offset infor
   mov ah, 9       ;显示提示信息“input:”
   int 21h
   mov ah, 1       ;键盘输入
   int 21h
   sub al, 30h     ;去掉ASCII码
   mov bl, al
```

```
mov dl, 2ah    ;显示乘号
mov ah, 2
int 21h
mov ah, 1
int 21h        ;输入第2个数
sub al, 30h
mov ah, 0
mul bl         ;相乘
aam              ;十进制乘法调整
  ;乘积高位数在AH,低位数在AL
mov x, al      ;保存结果
mov x+1, ah
add ax, 3030h
mov bx, ax
```

```
    mov ah, 2
    mov dl, 3dh   ;显示‘=’
    int 21h
    mov dl, bh    ;显示结果
    int 21h
    mov dl, bl
    int 21h
    mov ah, 4ch
    int 21h
end start
```

实验任务

- 实验：

  完成下列实验内容：

  设计程序。实现**Y=2X+3**，**X**是一位十进制数。要求**X**从键盘输入，在下一行上显示**’y=2X+3=’**以及十进制计算结果。