

## 第13章 十進制算術

計算機的使用者(人)使用十進制處理數值。計算機必須能接受十進制輸入;處理完畢後,以十進制顯示出來。

### 本章介紹的指令

AAA: 加法後的ASCII調整  
AAD: 除法前的ASCII調整  
AAM: 乘法後的ASCII調整  
AAS: 減法後的ASCII調整  
DAA: 加法後的十進制調整  
DAS: 減法後的十進制調整

## 十進資料的表示法

- 二進制加碼十進制 (Binary coded decimal, BCD)

- 使用四個位元來表示一位數十進數值

$0_{10}$ :  $0000_2$ ,       $1_{10}$ :  $0001_2$ ,       $2_{10}$ :  $0010_2$

$3_{10}$ :  $0011_2$ ,       $4_{10}$ :  $0100_2$ ,       $5_{10}$ :  $0101_2$

$6_{10}$ :  $0110_2$ ,       $7_{10}$ :  $0111_2$ ,       $8_{10}$ :  $1000_2$

$9_{10}$ :  $1001_2$

- 四個位元數值還有 $1010_2$ ,  $1011_2$ ,  $1100_2$ ,  $1101_2$ ,  $1110_2$ ,  $1111_2$ 。在BCD碼中, 它們都是不正確的值

## BCD表示法

- 未打包 (Unpacked): 一個位元組存放一位數。放在低階四位元。高階四位元放0
- 打包 (Packed): 一個位元組存放兩位數。高階四位元和低階四位元各放一位數
- ASCII: 30H~39H, 一個位元組存放一位數。類似未打包表示法

例子: 1527

ASCII	31 35 32 37	四個位元組
未打包	01 05 02 07	四個位元組
打包	15 27	二個位元組

## AAA 和 AAS

- AAA指令: 檢查AL的右邊四位元。若在A到F之間, 或輔助進位AF=1, 將加6到AL, 並加1到AH, 並設CF和AF為1。AL的高階四位元被清為0。
- AAS指令: 檢查AL的右邊四位元。若在A到F之間, 或輔助進位AF=1, 將由AL減去6, 並由AH減去1, 並設CF和AF為1。AL的高階四位元被清為0。

例子: 5+2。暫存器AL放5的ASCII碼35H, 暫存器BL放2的ASCII碼32H

```
ADD    AL,BL    35H
AAA                                + 32H
                                           67H
```

調整後 AL 成爲 07, 進位 0 加到 AH, CF=0

例子: 8+4。暫存器AL放 8 的 ASCII 碼 38H,  
暫存器 BL放 4 的 ASCII 碼 34H

```
ADD    AL,BL    38H
AAA                    + 34H
                        6CH
```

調整後 AL 成爲 02, 進位 1 加到 AH, CF=1

例子: 9+9。暫存器AL放 9 的 ASCII 碼 39H,  
暫存器 BL放 9 的 ASCII 碼 39H

```
ADD    AL,BL    39H
AAA                    + 39H
                        72H
```

調整後 AL 成爲 08, 進位 1 加到 AH, CF=1

## 程式例子: 十進制加法

```
ASCVALUE1 DB '548' ;ASCII items
ASCVALUE2 DB '724'
ASCTOTAL  DB '0000'
.386 ; ...
CLC
LEA SI,ASCVALUE1+2 ;Add ASCII values:
LEA DI,ASCVALUE2+2 ;initialize addresses
LEA BX,ASCTOTAL+3 ; of ASCII numbers
MOV CX,3 ;Initialize 3 loops
A20:
MOVZX AX,[SI] ;Load ASCII byte in AX
ADC AL,[DI] ;Add (with carry)
AAA ;Adjust for ASCII
MOV [BX],AL ;Store sum
DEC SI
DEC DI
DEC BX
LOOP A20 ;Loop 3 times
MOV [BX],AH ;At end, store carry
```

## 轉成 ASCII 碼

```
LEA BX,ASCTOTAL+3 ;Convert ASCTotal
MOV CX,4 ;to ASCII format
A30:
OR BYTE PTR [BX],30H
DEC BX
LOOP A30 ;Loop 4 times
MOV AX,1300H ;Request display
MOV BX,0031H ;Page and attribute
LEA BP,ASCTOTAL ;ASCII line
MOV CX,4 ;Length of line
MOV DX,0824H ;Row and column
INT 10H
...
```

## 一邊計算, 一邊轉成 ASCII 碼

```
ADC  AL, [DI]           ;Add (with carry)
AAA                               ;Adjust for ASCII
PUSHF
OR   AL, 30H
POPF
MOV  [BX], AL           ;Store sum
```

OR 指令會改變旗標 (邏輯指令會使 CF,OF=0)  
但 CF 在下一次iteration 中須使用到, 因此以  
PUSHF和 POPF來保存

## ASCII 數字的減法

例子: 9-5。ASCVAL1放 9 的 ASCII 碼 39H,  
ASCVAL2放 5 的 ASCII 碼 35H

```
MOV    AL,ASCVAL1      39H
SUB    AL,ASCVAL2      - 35H
AAS                                04H
OR     AL,30H
```

調整後 AL 仍為 04

例子: 5-9。ASCVAL1放9的ASCII碼39H,  
ASCVAL2放5的ASCII碼35H

```
MOV    AL,ASCVAL2    35H
SUB    AL,ASCVAL1    - 39H
AAS
OR     AL,30H
```

調整後 AL 成爲 06, AH減1, CF=1

## AAM 指令

將AL除以10, 商放在AH, 餘數放在AL

例子: 5\*9。暫存器AL放5的ASCII碼  
35H, 暫存器CL放9的ASCII碼39H

```
AND    CL, 0FH
AND    AL, 0FH
MUL   CL
AAM
OR     AX, 3030H
```

不能直接將ASCII  
碼相乘

## 程式例子

```
multcand db '3783' ;ascii items
multpler db '5'
ascprod db 5 dup (0)
;
; ...
mov cx,4 ;initialize 4 loops
lea si,multcand+3
lea di,ascprod+4
and multpler,0fh ;Clear ASCII 3
a20:
mov al,[si] ;Load ASCII character
and al,0fh ;Clear ASCII 3
mul multpler ;Multiply
aam ;Adjust for ASCII
add al,[di] ;Add to
aaa ; stored
mov [di],al ; product
dec di
mov [di],ah
dec si
loop a20 ;Loop 4 times
;
; ...
```

```
lea bx,ascprod+4 ;Convert product to ASCII
mov cx,5 ;right to left, 5 bytes
a30:
or byte ptr [bx],30h
dec bx
loop a30 ;Loop 5 times
mov ax,1300h ;request display
mov bx,0031h ;Page and attribute
lea bp,ascprod;ASCII line
mov cx,5 ;Length of line
mov dx,0824h ;Row
int 10h
```



## AAD 指令

- 在除法運算之前作。
- 將AH乘以10, 加到AL。AH 清為 0。

例子:  $28 \div 7$ 。暫存器AX放 28 的 ASCII 碼 3238H, 暫存器 CL放 7 的 ASCII 碼 37H。

```
AND    CL, 0FH    ;CL=7
AND    AX, 0F0FH  ;AX=0208
AAD                    ;AX=1C
DIV    CL          ;AX=0004
```

## 程式例子

```
dividend db '3698' ;ASCII items
divisor  db '4'
ascquot  db 4 dup (0), '$'
;
; ...
mov  cx, 4 ;Initialize 4 loops
sub  ah, ah ;clear left byte of dividend
and  divisor, 0fh ;Clear divisor of ASCII 3
lea  si, dividend
lea  di, ascquot
a20:
  lodsb ;load ASCII byte
  and  al, 0fh ;Clear ASCII 3
  aad ;Adjust for divide
  div  divisor ;divide
  stosb ;store quotient
  loop a20 ;Loop 4 times
```

```

;Convert product to ASCII
    lea  bx,ascquot    ;left to right
    mov  cx,4          ; 4 bytes
a30:
    or   byte ptr [bx],30h
    inc  bx
    loop a30           ;Loop 4 times
    mov  ah,9          ;display
    lea  dx,ascquot    ; quotient
    int  21h

```

## 打包BCD資料的處理

- DAA: 若AL的低階四位元大於9, 或AF=1, 加6到AL, 設AF=1。接著, 若AL大於99, 或CF=1,加60H到AL, 設CF=1。
- DAS: 若AL的低階四位元大於9, 或AF=1, 減6到AL, 設AF=1。接著, 若AL大於99, 或CF=1,減60H到AL, 設CF=1。

## 程式例子

- 圖13-4, 將ASCII數值轉成Packed BCD, 而後相加

```
ascvalue1    db    '057836'    ;ASCII data items
ascvalue2    db    '069427'
bcdvalue1    db    '000'      ;BCD data items
bcdvalue2    db    '000'
bcdsum       db    4 dup (0)
            .386    ;
            ...
            lea    si,ascvalue1+4 ;Initialize ASCII
            lea    di,bcdvalue1+2 ; and BCD values
            call   b10convrt      ;call convert routine
            lea    si,ascvalue2+4 ;Initialize ASCII
            lea    di,bcdvalue2+2 ; and BCD values
            call   b10convrt      ;call convert routine
            ;Add BCD numbers:
            xor    ah,ah          ;clear AH
            lea    si,bcdvalue1+2 ;Initialize
            lea    di,bcdvalue2+2 ; BCD
            lea    bx,bcdsum+3    ; addresses
            mov    cx,3           ;3-byte fields
            cld

a20:         mov    al,[si]        ;get BCDVALUE1
            adc    al,[di]        ;Add BCDVALUE2
            daa                    ;Decimal adjust
            mov    [bx],al        ;Store in BCDSUM
            dec    si
            dec    di
            dec    bx
            loop  a20            ;Loop 3 times
            ;
            ...
```



## 二進制數值轉成ASCII

```
ASCVALUE DB 4 DUP (?)
BINVALUE DW 0DF1H
...
MOV CX, 10
LEA SI, ASCVALUE+3
MOV AX, BINVALUE
L20:
CMP AX, CX
JB L30
XOR DX, DX
DIV CX
OR DL, 30H
MOV [SI], DL
DEC SI
JMP L20
L30:
OR AL, 30H
MOV [SI], AL
```

## 四捨五入

- 運算時, 先移去小數點(小數部分位數另外存放), 十進數值轉成二進制
- 若想將運算結果的小數部分捨去N位, 應將結果右移N位(除以 $10^N$ )
- 要作四捨五入, 應在右移前先加 $10^N/2$

例如: 1.2345 (數值是12345, 小數部分位數4)  
要捨去兩位, 應先加上50(得12395)後, 再除以100 (得123, 小數部分位數剩2位)

## 程式例子

- 圖13-5。
- 鍵盤輸入兩個有小數點的十進數字
- 數值部分轉成二進制, 相乘
- 結果只保留2位小數點, 顯示出來

## 資料定義

```
.MODEL SMALL
.STACK 64
.DATA
LEFTCOL EQU 28
RIGHTCOL EQU 52
TOPROW EQU 10
BOTROW EQU 14
QTYPARAM LABEL BYTE
MAXQLEN DB 6
ACTQLEN DB ?
QTYFLD DB 6 DUP (?)
RATEPAR LABEL BYTE
MAXRLEN DB 6
ACTRLEN DB ?
RATEFLD DB 6 DUP (?)
;
PROMPT1 DB 'Quantity?      '
PROMPT2 DB 'Rate?                '
PROMPT3 DB 'Product = '
ASCPROD DB 10 DUP (30H)
PROMPT4 DB 'Press any key to continue or ESC to quit'
ADJUST DW ?
BINPROD DW 0
BINQTY DW 0
BINRATE DW 0
COL DB 0
DECIND DB 0
MULT10 DW 1
NODECIMS DW 0
ROW DB 0
SHIFT DW ?
TENWD DW 10
```

## 主程式

```
.386 ;-----  
      .CODE  
A10MAIN PROC FAR  
      MOV     AX,@DATA  
      MOV     DS,AX  
      MOV     ES,AX  
      MOV     AX,3 ;SET VIDEO MODE  
      INT     10H  
  
A20:  
      CALL    Q10WINDOW  
      CALL    B10INPUT  
      CALL    C10QTY  
      CALL    D10RATE  
      CALL    E10MULT  
      CALL    F10PROD  
      CALL    G10FORMAT  
      CALL    H10PAUSE  
      CMP     AL,1BH  
      JNE     A20  
      MOV     AX,4C00H  
      INT     21H  
A10MAIN ENDP
```

## 輸入數量

```
      ; ACCEPT QUANTITY & RATE FROM KEYBOARD  
      ;  
B10INPUT PROC NEAR  
      MOV     ROW,TOPROW+1  
      MOV     COL,LEFTCOL+3  
      LEA     BP,PROMPT1  
      MOV     CX,15  
      CALL    K10DISPLAY  
      MOV     AH,0AH  
      LEA     DX,QTYPARAM  
      INT     21H  
      MOV     COL,LEFTCOL+3  
      INC     ROW  
      LEA     BP,PROMPT2  
      MOV     CX,15  
      CALL    K10DISPLAY  
      MOV     AH,0AH  
      LEA     DX,RATEPAR  
      INT     21H  
      INC     ROW  
      RET  
B10INPUT ENDP
```

## 轉換成二進制

```

;          CONVERT QUANTITY TO BINARY
;
C10QTY PROC    NEAR
            MOV     NODECIMS,0
            MOUZX  CX,ACTQLEN
            LEA    SI,QTYFLD-1
            ADD    SI,CX
            CALL   J10ASCBIN
            MOV    AX,BINPROD
            MOV    BINQTY,AX
            RET
C10QTY ENDP
;          CONVERT RATE TO BINARY
;
D10RATE PROC    NEAR
            MOUZX  CX,ACTRLEN
            LEA    SI,RATEFLD-1
            ADD    SI,CX
            CALL   J10ASCBIN
            MOV    AX,BINPROD
            MOV    BINRATE,AX
            RET
C10RATE ENDP

```

## 相乘

```

;          MULTIPLY QUANTITY X RATE, ROUND AND SHIFT
;          PRODUCT BASED ON NO. OF DECIMAL PLACES
;
E10MULT PROC    NEAR
            MOV     CX,10
            LEA    DI,ASCPROD
            MOV    AL,30H
            CLD
            REP STOSB
            MOV    SHIFY,10
            MOV    ADJUST,0
            MOV    CX,NODECIMS
            CMP    CL,6
            JA     E40
            SUB    CX,2
            JLE   E30
            MOV    NODECIMS,2
            MOV    AX,1
E20:       INUL   AX,10
            LOOP  E20
            MOV    SHIFT,AX
            SHR    AX,1
            MOV    ADJUST,AX
E30:       MOV    AX,BINQTY
            MUL   BINRATE
            ADD   AX,ADJUST
            ADC   DX,0
            CMP   DX,SHIFT
            JB    E50
E40:       XIR    AX,AX
            JMP   E70
E50:       CMP    ADJUST,0
            JZ    E80
            DIV   SHIFT
E70:       XOR    DX,DX
E80:       RET
E10MULT ENDP

```



## 轉成ASCII，前面的0變為空格

```

;          CONVERT PRODUCT TO ASCII
;          -----
F10PROD PROC    NEAR
            LEA    SI,ASCPROD+7
            MOV    BYTR PTR [SI],'. '
            ADD    SI,NODECIMS

F30:        CMP    BYTE PTR [SI],'. '
            JNE    F40
            DEC    SI

F40:        CMP    DX,0
            JNZ    F50
            CMP    AX,10
            JB     F60

F50:        DIV    TENWD
            OR     DL,30H
            MOV    [SI],DL
            DEC    SI
            SUB    DX,DX
            JMP    F30

F60:        OR     AL,30H
            MOV    [SI],AL
            RET

F10PROD ENDP

```

```

;          FORMAT PRODUCT TO DISPLAY
;          -----
G10FORMAT PROC NEAR
            MOV    CX,LEFTCOL+3
            MOV    CX,9
            LEA    SI,ASCPROD

G20:        CMP    BYTE PTR [SI],30H
            JNE    G30
            MOV    BYTE PTR [SI],20H
            INC    SI
            LOOP   G20

G30:        LEA    BP,PROMPT3
            MOV    CX,20
            CALL   K10DISPLAY
            RET

G10FORMAT ENDP

```

## ASCII至 二進制轉換

```

;          convert ASCII quantity and rate to binary
;          SI set with address of qty/rate on entry:
;          -----
J10ASCBIN PROC NEAR
            MOV    MULT10,0
            MOV    BINPROD,0
            MOV    DECIND,0
            XOR    BX,BX

J20:        MOV    AL,[SI]
            CMP    AL,'. '
            JNE    J30
            MOV    DECIND,1
            JMP    J40

J30:        AND    AX,0FH
            MUL    MULT10
            ADD    BINPROD,AX
            MOV    AX,MULT10
            IMUL  AX,10
            MOV    MULT10,AX
            CMP    DECIND,0
            JNZ    J40
            INC    BX

J40:        DEC    SI
            LOOP  J20
            CMP    DECIND,0
            JZ     J90
            ADD    NODECIMS,BX

J90:        RET

J10ASCBIN ENDP

```

等待按鍵、  
顯示字、  
清螢幕

```
; PAUSE FOR USER, PRESS ANY KEY TO QUIT  
;  
;-----  
H10PAUSE PROC NEAR  
    MOV     COL,20  
    MOV     ROW,22  
    LEA     BP,PROMPT4  
    MOV     CX,40  
    CALL    K10DISPLAY  
    MOV     AH,10  
    INT     16H  
    RET  
H10PAUSE ENDP  
; Display characters, set attribute  
;-----  
K10DISPLAY PROC NEAR  
    MOV     AX,1301H  
    MOV     BX,0016H  
    MOV     DH,ROW  
    MOV     DL,COL  
    INT     10H  
    RET  
K10DISPLAY ENDP  
; Scroll display window, set attribute  
;-----  
Q10WINDOW PROC NEAR  
    MOV     AX,0600H  
    MOV     BH,16H  
    MOV     CH,TOPROW  
    MOV     CL,LEFTCOL  
    MOV     DH,BOTROW  
    MOV     DL,RIGHTCOL  
    INT     10H  
    RET  
Q10WINDOW ENDP
```