算數運算（減法一）

介紹完加法，接下來要介紹減法了，以 8 bits 的二進位運算為例：0000 1000B - 0000 0010B = 0000 0110B，對於這個答案的正確性應該是沒有疑問的吧！重點是如何得到這個答案？第一種方式是利用借位的減法運算，這是減法最基本的運算法則，也是此刻您心中的算法，這個方式可以利用 SUBB 的指令做到，它的運算方式是先減去 CY 旗標的值，再進行減法運算，如果有借位的情形，CY 旗標會被設為 1。用上面的算式做例子，我們可以利用以下的指令來完成：

```
CLR     C
MOV     R0,#08H
MOV     A,R0
SUBB    A,#02H
MOV     R0,A
```

答案就在 Acc 累加器中。特別需要注意的是：進行 SUBB 的減法運算時，一定要先把 CY 旗標設為 0，以免影響到第一次 SUBB 的運算值，因為 SUBB 的指令就如同 ADDC 一樣，是一開始便先處理 CY 值，再做數值的運算。

第二種方法是將減數反相再加 1，把被減數與反相後的減數相加，得到的結果便是答案了，我們實際來算一下好了。

先將 0000 0010B 反相得到 1111 1101B
再將 1111 1101B 加 1 得到 1111 1110B
再將 1111 1110B 與 0000 1000B 相加得到 0000 0110B（CY=1） → 解答

很神奇嗎？這就是二進位運算的特色，如果您不相信，可以再用其他的數值算看看，得到的結果應該都是一樣的，所以在 8051 裡如果要進行減法的話，您也可以這樣寫：

```
CLR     C
MOV     R0,#08H
MOV     A,#02H
CPL     A
INC     A
ADD     A,R0
MOV     R0,A
```


旗威科技有限公司　地址：高雄市三民區昌裕街 18-1 號　網址：http://www.chipware.com.tw
技術專線：07-395-5152　技術支援傳真：07-395-5155　E-mail：chipware@chipware.com.tw

這裡有一個重點要特別強調！如果您所用的是第一種方法來運算，那麼運算結束時 CY=1 是產生借位，CY=0 是沒有借位；可是以第二種方式運算，CY=1 是沒有借位，CY=0 是產生借位，兩種情況剛好相反，這就要端看您程式上的應用來做選擇了。

除了 SUBB 之外，還有一個叫做 DEC 的助憶碼，它和 INC 的功能相仿，是把指定位址的值減 1，不過要留意的是：DEC DPTR 這個指令是不存在的，DEC 僅能做 8 bits 的運算，不能做 16 bits 的運算，如果要執行 DEC DPTR，那麼您可以這樣改寫：

```
DPTR_DEC
    PUSH    PSW
    PUSH    A
    CLR     C
    MOV     A,DPL
    SUBB    A,#01H
    MOV     DPL,A
    MOV     A,DPH
    SUBB    A,#00H
    MOV     DPH,A
    POP     A
    POP     PSW
```

這一段範例中，用到 PUSH 與 POP 兩個指令，這是為了儲存 PSW 與 A 的狀態，因為 INC 並不會更改任何的旗標值，如果要將 DEC 設定成與 INC 具有同樣的特性，那麼 PSW 與 Acc 累加器的內容是不能被更動的，而 SUBB A,#00H 這一行是為了將借位的值減去，這與 INC DPTR 有相同的作用。到此 8 bits 的減法算是完整介紹完了，同樣地，下一篇文章我們要進入 32 bits 的減法運算，您不妨先著手試試，再比較文章中的寫法與您所撰寫的程式有什麼不同的地方。

旗威科技有限公司　地址：高雄市三民區昌裕街 18-1 號　網址：http://www.chipware.com.tw
技術專線：07-395-5152　技術支援傳真：07-395-5155　E-mail：chipware@chipware.com.tw