

算數運算（乘法一）

終於要進入乘法的運算了，這真可算是一個漫長的路程，不曉得您對前幾篇算術運算的原理是否真的明白了呢？因為我們即將更進一步利用加法的原理，進行乘法的運算，您準備好了嗎？我們要開始囉！

首先還是要先介紹在 MCS-51 指令集裡專為乘法運算所安排的指令，聰明的您想到了嗎？沒錯，就是 MUL AB。這個指令的運作方式是將 Acc 累加器裡的 data 和暫存器 B 的 data 相乘，產生 16bits 的乘積，並將高位元組的結果存在暫存器 B 中，低位元組的結果存在 Acc 累加器中，這個指令只需要一個 Byte 的程式空間，卻需要四個機器週期才能運算完成。這裡還有一個很重要的動作，就是 OV 旗標與 CY 旗標的值會被改變，當結果大於 FFH 時，OV 旗標會被設為 1，而 CY 則固定設為 0。

上面就是 8 bits 無正負號的乘法運算，很簡單對不對？這是因為指令集裡剛好提供了這樣的指令，問題來了，如果是 16 bits 的乘法運算呢？或是 32 bits 的乘法運算呢？你會運用 MUL 來計算嗎？或是你有更好的方法？

回歸到乘法運算的原點，我們來探討乘法運算是如何動作的：假設今天有兩個要計算的 8 bits 數值分別為 10H 與 03H 要相乘，我們都知道結果會是 30H，重點是你會怎麼算？應該會像下面這樣的算法吧！

$$\begin{array}{r}
 10 \\
 \times 03 \\
 \hline
 30 \\
 + 00 \\
 \hline
 30H
 \end{array}
 \longrightarrow
 \begin{array}{r}
 0001\ 0000 \\
 \times 0000\ 0011 \\
 \hline
 0001\ 0000 \\
 0\ 0010\ 000 \\
 00\ 0000\ 00 \\
 000\ 0000\ 0 \\
 0000\ 0000 \\
 0\ 0000\ 000 \\
 00\ 0000\ 00 \\
 + 000\ 0000\ 0 \\
 \hline
 0011\ 0000
 \end{array}$$

我們看看右手邊的算式，它的運算原理其實是不斷地相加，由於只有 0 與 1，所以當乘數的某個 bit 為 1，就在該 bit 上加上一個被乘數，若為 0 則不理它，如果由低位元往高位元加，那麼程式可以改寫成這樣：

```
MOV    R4,#03H ;乘數的存放區
MOV    R5,#00H ;運算的暫存區
MOV    R6,#10H ;被乘數的存放區
;
MOV    R3,#9   ;旋轉 9 次,含 CY 旗標
CLR    C       ;將 CY 旗標設為 0
$1 MOV    R1,#05H ;設定運算暫存區的啟始位址
MOV    R2,#02H ;設定運算範圍(Bytes)
$2 MOV    A,@R1 ;由最低位元開始運算
RRC    A
MOV    @R1,A
DEC    R1
DJNZ   R2,$2
JNC    $3
CLRC
MOV    A,R6
ADDC  A,R5
MOV    R5,A
$3 DJNZ   R3,$1 ;往下一位元移動
;
```

想知道這個程式是怎麼運作的嗎？別急別急，先自行推算看看，下一篇文章再告訴您，很精彩唷！