算數運算（除法一）

相信有很多人看完乘法運算的原理之後，都很期待除法運算快一點公布，然而除法運算的動作原理雖然很單純，但運算的原理卻不簡單，我們花了一些時間試著將這個部分以較淺顯的方法來詮釋，讓大家更容易理解，另一方面也表達出我們對介紹算術運算的謹慎與重視，接下來我們要正式進入除法的領域了，您準備好了嗎？

在 8051 的指令集裡，有提供 8 bits 相除的指令，用心的你一定馬上猜到這個指令是 DIV AB 了，只要將被除數放在 Acc 累加器，將除數放在 B 暫存器，那麼 CPU 就會自動幫你把運算後的商數放在 Acc 累加器中，把餘數放在 B 暫存器中。這個部分就算不用解釋，相信你一定也知道，不過，這個指令只能處理 8 bits 的相除，我們真正好奇的是：16 bits 或是 32 bits 的除法應該如何運算？因此，我們必須同乘法一般，探討一下除法運算的運算法則。

看過乘法運算的人應該都知道：乘法是由連續的相加所組成的；同樣的道理，除法是由連續的相減所組成的，我們先來回想一下我們所學過的除法是怎麼運算的：

假設被除數是 317，除數是 3，聰明的你一定馬上算出：商數是 105，餘數是 2。OK，那麼換我問你，你是怎麼算出來的呢？應該是利用下面這樣的方法吧？

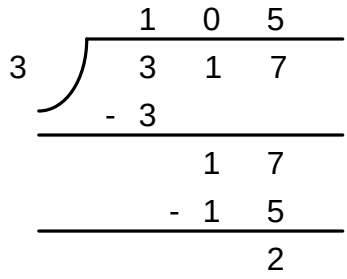
```
        1
   ┌──────
3  │  3 1 7
   - 3
   ──────
     1
```

一開始的時候，我們會先從**最高的位數**先算，判斷最高位數的值**夠不夠除數來減**，而且**可以減幾次**。經過判斷後，我們會發現百位數的值剛好等於被除數，可以讓被除數減掉 1 次，所以我們在被除數的**百位數上面做個被減掉 1 次的記號**，並**將下一個位數放進來**。

```
        1 0
   ┌──────
3  │  3 1 7
   - 3
   ──────
       1 7
```

接下來，我們把十位數擺進來，經判斷後發現被除數減去除數後，所剩下在十位數的值並**不夠讓除數減**，因此我們在被除數的十位數上面**做個減掉 0 次的記號**，再將下一個位數放進來。

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



最後，我們發現被除數減去除數後，在個位數所剩下的值可以讓被除數減去 5 次，因此我們在被除數個位數的上面做個減去 5 次的記號，因為我們做的是整數運算，因此**剩下的數值就是餘數了**。

看完了上面的介紹，相信你一定覺得我很囉嗦，把一個簡單的除法拆成那麼多個部分來介紹，好像在教小學生一樣。不過對一個 CPU 來說，它並沒有思考的能力，程度比一個小學生還差，如果不能夠把除法運算的每一個小細節交待清楚，那麼 CPU 就一定會算錯，而且錯得很離譜，錯得很無厘頭，這是題外話，我們回到主題：除法的運算法則。

到此為止，我們可以體會除法的運算，就是先把被除數最高位數的值拿出來跟除數比較，如果大於等於除數的話，就減掉除數，並記錄減掉的次數，再把下一個位數搬進來，直到所有的位數都搬完為止。不過，在十進位的運算裡，也許每次減的次數可能會超過一次，就像上面的例子一樣，在個位數時可以減掉五次，然而在二進位的領域裡，只有 1 跟 0，最多只要減掉一次，被除數在該位數就一定會比除數小，因此我們只要把該位數有發生減法的動作記錄下來就 OK 了。

講解完了除法的運算法則，聰明的你是否已經想到要怎麼撰寫 16 bits 或是 32 bits 的運算方式了呢？讓我們先賣個關子，下一篇文章我們要用實際的程式，推導一下除法運算的動作，很精彩唷！

提示：文章裡標示為**紅色的文字**，為除法運算的動作精華。

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