

## 常用指令排行榜(十)

這是常用指令排行榜的最後一篇文章，我們要探討的主題有兩個：一個是 MOV 指令的特殊用法，另一個是記憶空間與 MOV 指令的關係。在所有 MOV 的指令中，有幾個指令是比較特別的，第一類是布林位元處理的指令，第二類是外部資料傳輸的指令，我們來看看這些指令有什麼與眾不同的地方。

首先介紹布林位元處理的指令。讓我們回憶一下什麼是『布林位元處理』？所謂的『布林位元處理』，就是可以針對單一 bit 做處理及運算的功能，這樣的功能簡化了程式撰寫的複雜度，讓 8051 的運用更顯得靈活自如。在 MOV 的指令中，有兩個指令是屬於這一類的指令：一個是 MOV C,bit，另一個是 MOV bit,C。就定址的分類來看，它們是屬於直接定址模式的運作方法，較特別的地方是：這兩個不同的 bit 裡，其中一個必須是 CY 進位旗標。在算術運算裡，這是非常重要的指令，因為它們可以直接影響到進位旗標的值，提供觸發條件的設定，如果您的程式裡有必要記錄 CY 旗標的借溢位狀況的話，那麼這兩個指令必定是不可或缺的。

此外，您有注意到這兩個指令的機械週期不太一樣嗎？您沒有看錯，在 intel 原廠所公布的資料裡，這兩個指令的運算週期的確是不同的！然而某些改良過的相容晶片，會將這兩個指令的機器週期變得一樣，這會造成怎樣的結果呢？最明顯的就是運算時序上的改變，一樣的程式從舊系統移轉到改良過的晶片時，整個動作完全亂掉了！只差一個機器週期，就會差很多，因此撰寫程式與選用晶片時，這也是必須考量的重點。

其次我們要談到的是外部資料的傳輸。在 MOV 指令中，使用 MOVX 的指令便是專門用在外部資料傳輸的指令，其中包含了 MOVX A,@DPTR、MOVX @DPTR,A、MOVX A,@Ri、MOVX @Ri,A。由於現在的外部記憶體空間越來越大，相形之下使用 MOVX A,@Ri、MOVX @Ri,A 這兩個指令的人應該是稀有動物了。在早期有較小空間（如 256 Bytes 或是 128 Bytes）的記憶體時，這些指令可用來儲存一些系統的設定值並有效節省程式空間，然而現在記憶體的容量動輒 32K Bytes（256K bits）以上，甚至已經出現擁有 1M Bytes 的資料記憶體，這樣的指令便顯得有些不合時宜了。

談到這裡，筆者要引入另一個 8051 的重頭戲，也就是記憶空間的分配與 MOV 的關係。還記得上一篇文章中我們還有一個問題尚未探討嗎？就是 MOV R1,80H 為什麼不能用呢？回想一下，80H 在 8051 的記憶體中所佔用的位址，已經是 SFR 的起點了！在『8051 單晶片徹底研究--基礎篇』的附錄 F 裡，有一個 SFR（Special Function Register）表，80H 所代表的位址正是 P0 埠，因此在使用上可能會受到一些限制。如果利用 MOV R1,P0 的方式來寫，那就 OK 囉。不過在多數的組譯器裡還是會提供 MOV R1,80H 這樣的用法，若換成是 MOV R1,86H 便真的無法使用了，因為 86H 在 SFR 裡並未定義其功能！身為程式設計師與系統開發人員的您，可要特別注意這些小細節，因為某些晶片所提供專用的組譯器或模擬器裡會有不少類似的限制。

附帶一提的是：各家 MCS-51 相容晶片的規格不盡相同，還必須依晶片特性與改良部分加以修正您的程式，諸如時間延遲、動作時序，這些都必須跟著有所修正，因此在選用晶片之前，先花點時間了解一下您所選用的晶片特性吧！常用指令排行榜到此便告一段落，有了這些基本概念，要寫簡單的 Assembler 程式應該是輕而易舉的了。下一個單元，我們要進入算術運算的領域，又是一個引人入勝的領域，敬請期待。