

# 正負數表示法

- 之前所介紹的二進位數值表示法，可以表達正整數，但對於負整數卻無法表達。為了讓電腦也可以表達負數，數學家發明了許多的數值表示法，並且很多都可以透過邏輯電路加以實作，三種最常見的正負數表示法分別如下

- 『帶符號大小』

- 『1's補數』

- 『2's補數』

- 這三種表示法都必須事先固定資料的位元長度
- 現代電腦使用的是『2's補數』表示法。
- 三種表示法的對照表如下：
  - (其中2's補數的負數表示法為1's補數負數+1)

# 正負數表示法

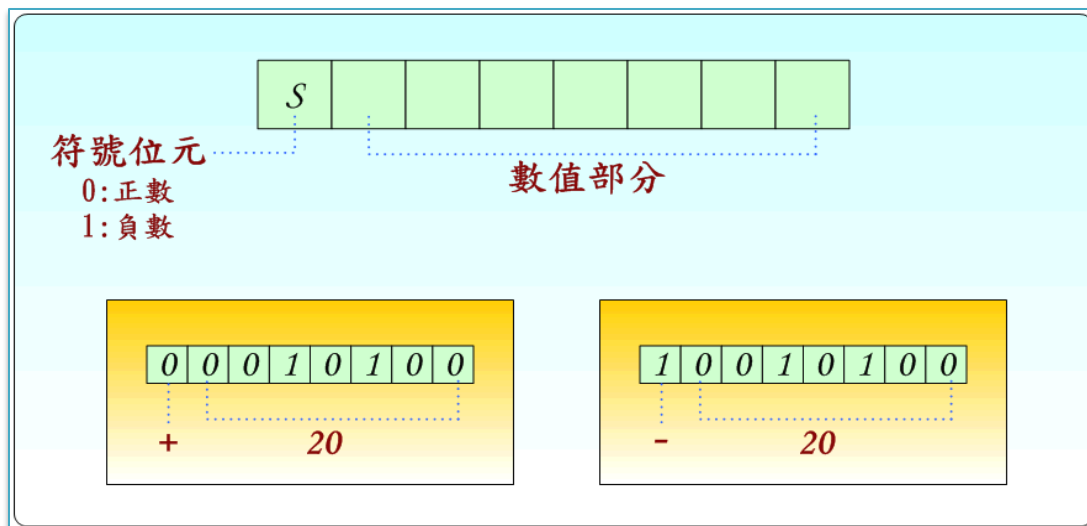
十進位	帶符號大小	1's補數	2's補數
+7	0111	0111	0111
+6	0110	0110	0110
+5	0101	0101	0101
+4	0100	0100	0100
+3	0011	0011	0011
+2	0010	0010	0010
+1	0001	0001	0001
0	+0	0000	0000
	-0	1000	
-1	1001	1110	1111
-2	1010	1101	1110
-3	1011	1100	1101
-4	1100	1011	1100
-5	1101	1010	1011
-6	1110	1001	1010
-7	1111	1000	1001
-8	無法表示	無法表示	1000

以4位元表示正負整數值的三種表示法對照表

最左位為1一定是負整數

# 正負數表示法--帶符號大小

- 帶符號大小的正負數表示法，顧名思義，就是有一個位元用來表示該數值為正數還是負數。這個位元通常位於最左邊
  - 使用 $n$ 個位元來表達正負整數時，數值的表達範圍就只剩 $n-1$ 個位元可以使用，所以正整數的表達範圍是 $+0 \sim +(2^{n-1}-1)$
  - 負整數的表達範圍是 $-(2^{n-1}-1) \sim -0$
  - 明顯地，對於 $0$ 而言，使用帶符號大小表示 $0$ 的時候， $+0$ 與 $-0$ 是不一樣的。
- 以 $8$ 位元來表示正負整數時，若採用帶符號大小來表示，方法則如下圖所列：

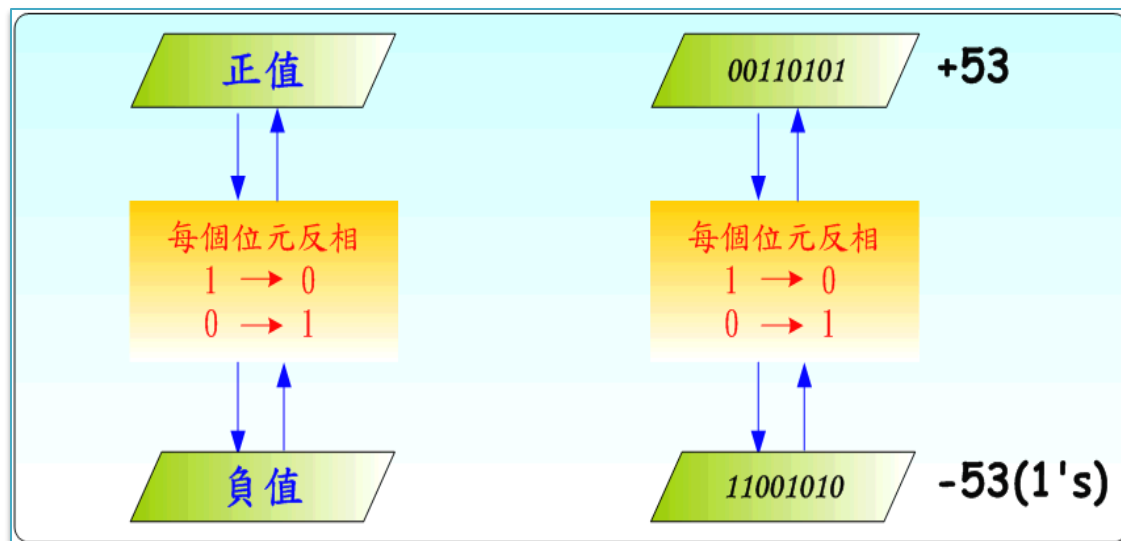


8位元的帶符號表示法

# 正負數表示法

# 1's補數

- 1's補數 (1's complement) 和帶符號大小表示法的原理不太相同，在1's補數中，如果要表達負數，則必須先求得正整數，然後再將每個位元加以反相 (inverse)，就可以得到負整數了
- 所謂反相，其實就是將該位元由1變0或由0變1，如圖範例：



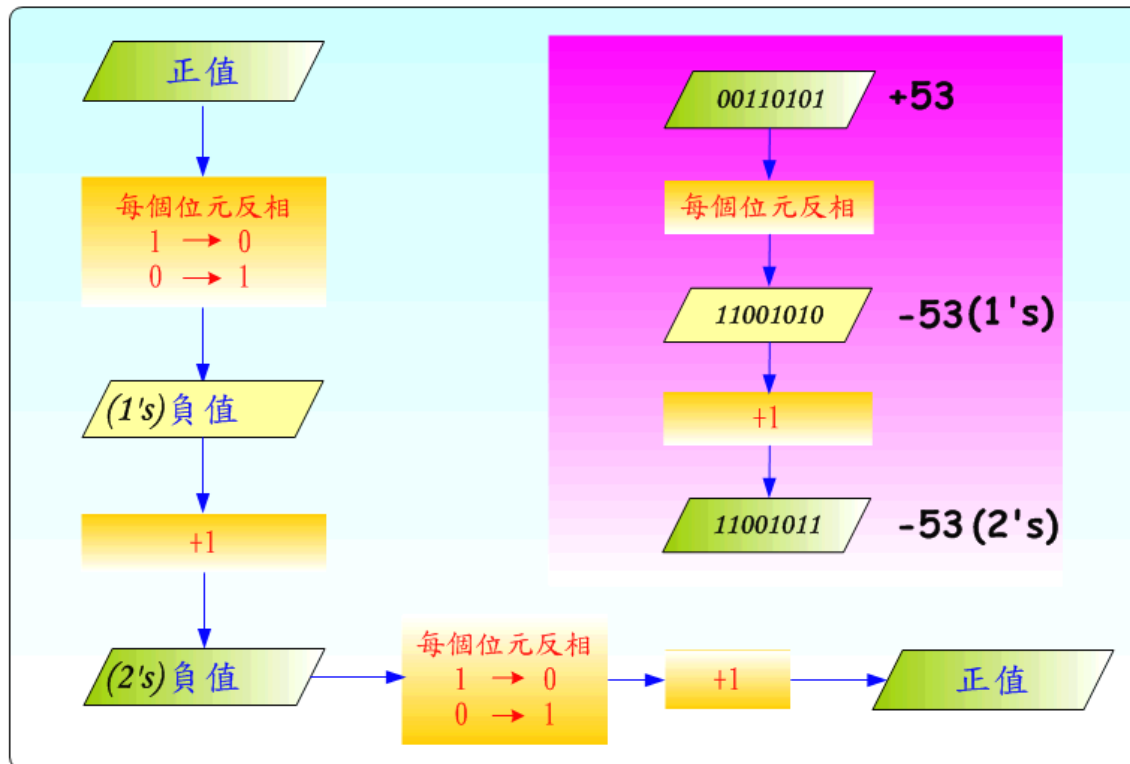
1's補數的正負轉換

- 在圖中，我們可以很明顯看到，1's補數的轉換機制是可逆的
- 換句話說，當您看到一個負的1's補數二進位數字時（最左位元必為1），如果想要知道該數值代表多少，同樣可以透過反相求出該負整數的絕對值。

# 正負數表示法

# 2's補數

- 2's補數 (2's complement) 是一個最完美的正負二進位整數的表示法，它是基於1's補數演變而來（同樣必須限制表示的位元寬度），也就是當我們要將一個二進位整數變號時，只需要先將其1's補數求出，然後再加1，就可以得到2's補數了，如圖所示：



2's補數的正負轉換

- 2's補數的轉換機制對於『正整數轉負整數』或『負整數轉正整數』所使用的方法都是一樣的
  - 例如，當您看到2's補數二進位負整數 $11001011_2$ 時，也可以先將其反相（使用1's補數的方法），然後再加1，就可以得到該負整數的絕對值了。

# 3 2's補數

	帶符號大小	1's補數	2's補數
採用	非個人電腦採用	非個人電腦採用	個人電腦採用
優點	<ol style="list-style-type: none"> <li>1.最左邊位元可判定正負數。</li> <li>2.對人來說非常簡單，容易理解。</li> </ol>	<ol style="list-style-type: none"> <li>1.最左邊位元可判定正負數。</li> <li>2.轉換機制是可逆的。</li> <li>3.可以透過邏輯電路輕鬆設計完成反相。</li> </ol>	<ol style="list-style-type: none"> <li>1.最左邊位元可判定正負數。</li> <li>2.+0與-0表示法相同。</li> <li>3.表達範圍是 <math>+(2^{n-1}-1) \sim -2^{n-1}</math>。</li> <li>4.轉換機制是可逆的。</li> <li>5.可以輕易地使用邏輯電路實作相關應用電路。</li> </ol>
缺點	<ol style="list-style-type: none"> <li>1.不容易使用邏輯電路實作相關應用電路。</li> <li>2.表達範圍是 <math>+0 \sim +(2^{n-1}-1)</math>、<math>-(2^{n-1}-1) \sim -0</math>。</li> <li>3.+0與-0表示方式不同。</li> </ol>	<ol style="list-style-type: none"> <li>1.必須限制表示的位元數。</li> <li>2.表達範圍是 <math>+0 \sim +(2^{n-1}-1)</math>、<math>-(2^{n-1}-1) \sim -0</math>。</li> <li>3.+0與-0表示方式不同。</li> </ol>	必須限制表示的位元數。

正負整數表示法比較表