

第三章 相關研究與改進方法

3.1 相關研究

本章節主要重點，搜集了幾種整數型態 FFT 方法並加以說明。分別用 FFT2、FFT3 和 FFT4 來區分各種不同整數型態 FFT。

3.1.1 整數型態 FFT2 [2]

本篇稱[2]的方法為整數型態 FFT2，以下簡稱 FFT2。由於 FFT1 在複數乘法運算上要用到 4 個暫存器。而 FFT2 的方法，其優點是複數乘法運算只要 2 個暫存器，所付出的代價就是需要”Three Step Lifting”[5] (運算過程中，會對分支 A 作運算(如下圖虛線處)，稱為 Lifting)，會降低運算速度。

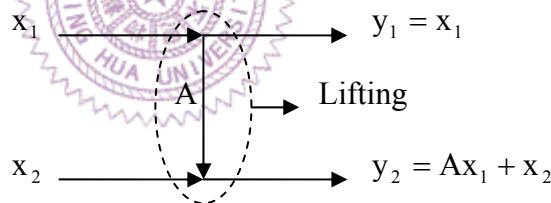


圖 3- 1 Lifting 示意圖

首先說明 FFT1 的複數乘法需要的暫存器數量，假設 $x = x_r + jx_i$ ，其中 x_r 和 x_i 分別為實數部和虛數部的值。另假設 $a = c + js$ 其中 c 和 s 分別代表 $\cos \theta$ 和 $\sin \theta$ 。

將兩個複數相乘得到下式：

$$y = x \times a = (cx_r - sx_i) + j(cx_i + sx_r)$$

由上式可知 FFT1 的複數乘法要有 4 個暫存器，分別存放 cx_r 、 sx_i 、 cx_i 和 sx_r 的值。

接下來介紹 FFT2 的方法，首先將 y 轉換成下面的矩陣形式[2]：

$$y = \begin{bmatrix} 1 & j \end{bmatrix} \begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} x_r \\ x_i \end{bmatrix}$$

再將上面矩陣形式用下圖[2]，表示其複數乘法運算過程的圖形。

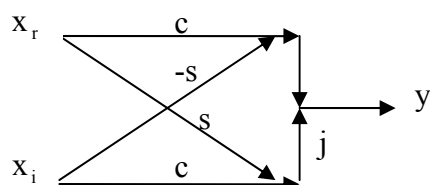


圖 3-2 複數乘法圖形

將之前的單一矩陣化分成三個矩陣相乘如下[2]：

$$R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} 1 & \frac{c-1}{s} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{c-1}{s} \\ s & c \end{bmatrix} = \begin{bmatrix} 1 & \frac{c-1}{s} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ s & 1 \end{bmatrix} \begin{bmatrix} 1 & \frac{c-1}{s} \\ 0 & 1 \end{bmatrix}$$

其中 1 可表示為 $s^2 + c^2$ (因為 $\sin^2 \theta + \cos^2 \theta = 1$)，本篇稱此矩陣為第一類型矩陣。將第一類型矩陣化成圖 3-3(a)。

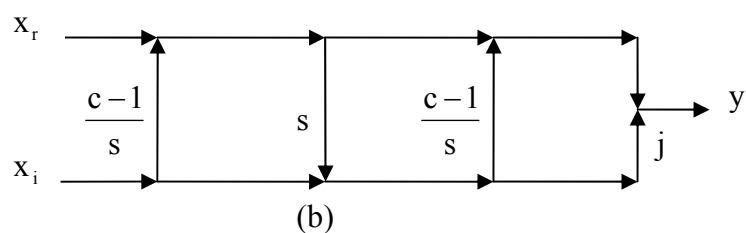
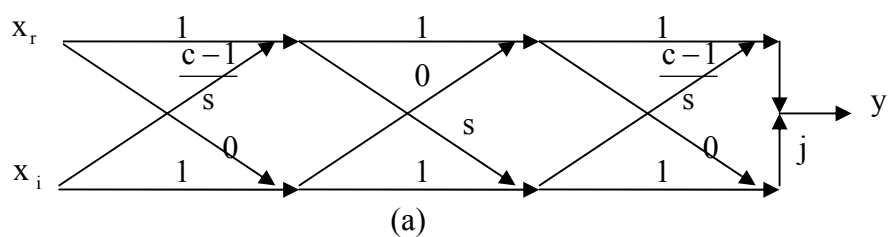


圖 3-3 第一類型矩陣圖形

再進行減化可得圖 3- 3(b)，運算過程即先前所稱的 Three Step Lifting，而 Three Step Lifting 的複數運算步驟如下：

$$\text{第一個 Lifting} = x_r + \frac{c-1}{s} \times x_i。$$

$$\text{第二個 Lifting} = \text{第一個 Lifting} \times s + x_i = \left(x_r + \frac{c-1}{s} \times x_i \right) \times s + x_i。$$

$$\begin{aligned} \text{第三個 Lifting} &= \text{第一個 Lifting} + \text{第二個 Lifting} \times \frac{c-1}{s} \\ &= \left(x_r + \frac{c-1}{s} \times x_i \right) + \left(\left(x_r + \frac{c-1}{s} \times x_i \right) \times s + x_i \right) \times \frac{c-1}{s}。 \end{aligned}$$

由上可知其複數運算，只需用 2 個暫存器分別存放 x_r 和 x_i ，再利用反覆疊代，就可完成複數運算。將圖 3- 3(b)中兩個主要參數 s 和 $\frac{c-1}{s}$ 各乘上 SF，並建入 Table 表內。另外因為 x_r 和 x_i 已內含 SF 倍，所以在運算 Lifting 時，都先乘上 $\frac{1}{SF}$ 將其倍數還原成 1 後再作運算，如下圖(a)所示。再將(a)簡化成(b)的圖形。

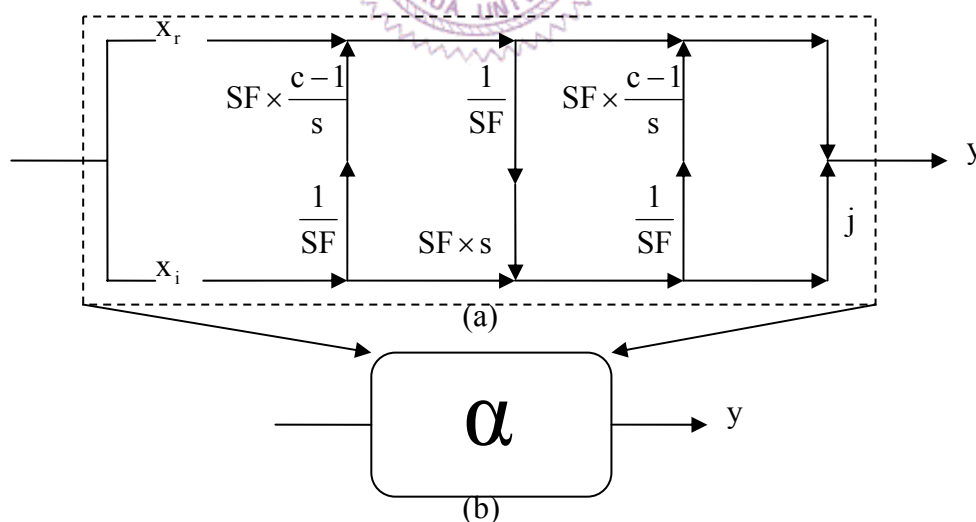


圖 3- 4 整數型態 FFT2 複數運算圖形

最後將圖 2-7 內的 W 置換成圖 3-4(b)，結果如下圖所示。

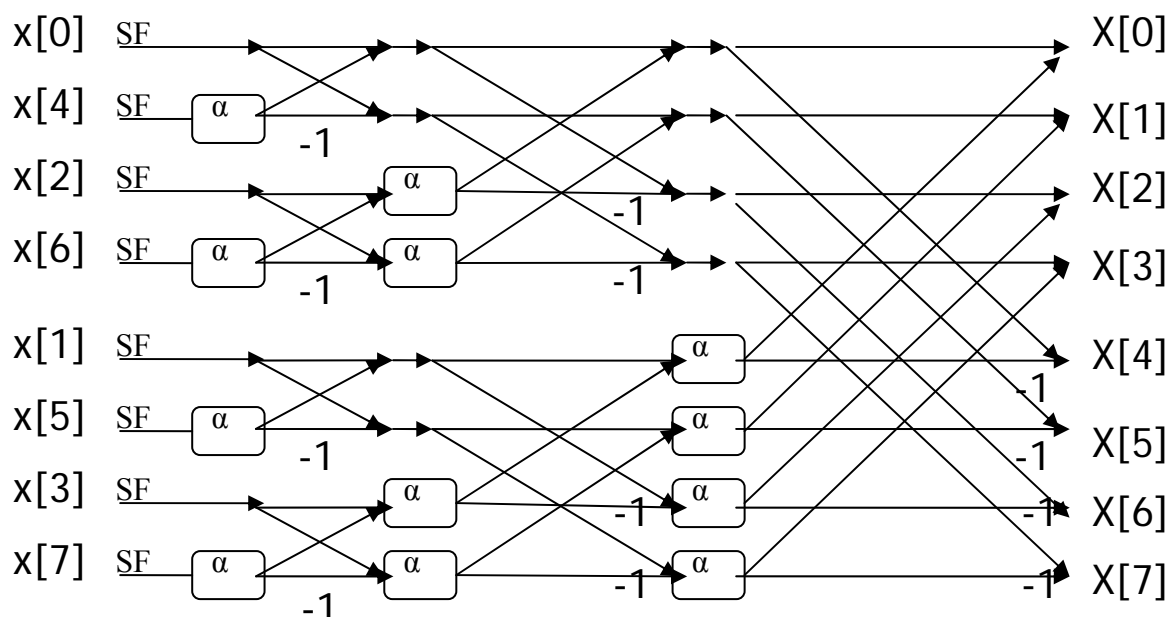


圖 3-5 整數型態 FFT2 蝴蝶圖

3.1.2 整數型態 FFT3 [5]

FFT2 改進了 FFT1 的缺點，但 FFT2 所付出的代價就是要 Three Step Lifting 的複數運算步驟，增加運算時間。因此[5]提出了 Two Step Lifting 改進方法，本篇將此方法稱為整數型態 FFT3。

首先將之前所提到的單一矩陣化成三個矩陣相乘如下所示，本篇稱此矩陣為第二類型矩陣。

$$R = \begin{bmatrix} c & -s \\ s & c \end{bmatrix} = \begin{bmatrix} c & 0 \\ 0 & \frac{1}{c} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ sc & 1 \end{bmatrix} \begin{bmatrix} 1 & -\frac{s}{c} \\ 0 & 1 \end{bmatrix}$$

再用之前 FFT2 所介紹過的簡化方法，簡化成圖 3-6(b)。

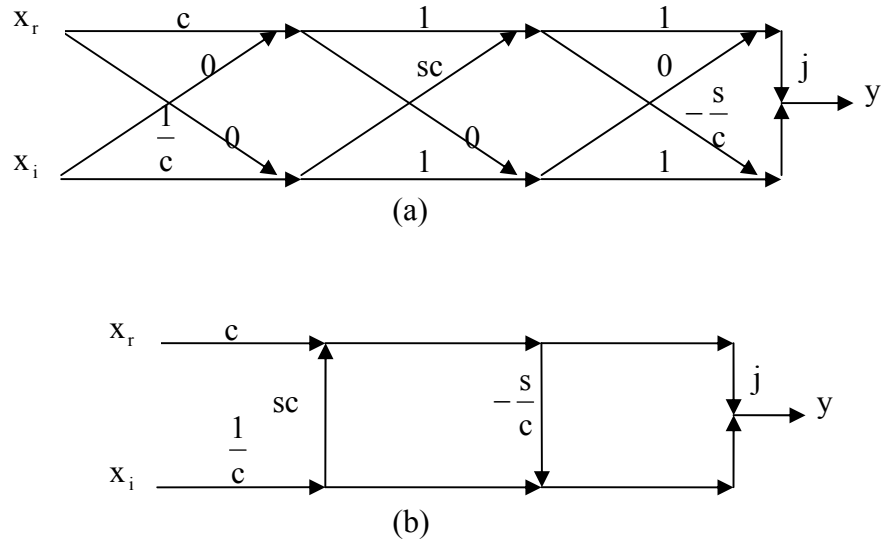


圖 3-6 第二類型矩陣圖形

由上圖可知 FFT3 複數乘法只需 Two Step Lifting，加快了運算速度。將上圖 (b) 中的四個主要參數 c 、 $-\frac{c}{s}$ 、 $\frac{1}{c}$ 和 sc 各乘上 SF ，並建入 Table 表內，另外因為 x_r 和 x_i 已內含 SF 倍，所以在運算 Lifting 時，都先乘上 $\frac{1}{SF}$ 將其倍數還原成 1 後再作運算，如下圖(a)所示。再將(a)簡化成(b)的圖形。

再將圖 2-7 內的 W 置換成下圖(b)，結果和圖 3-5 相似。

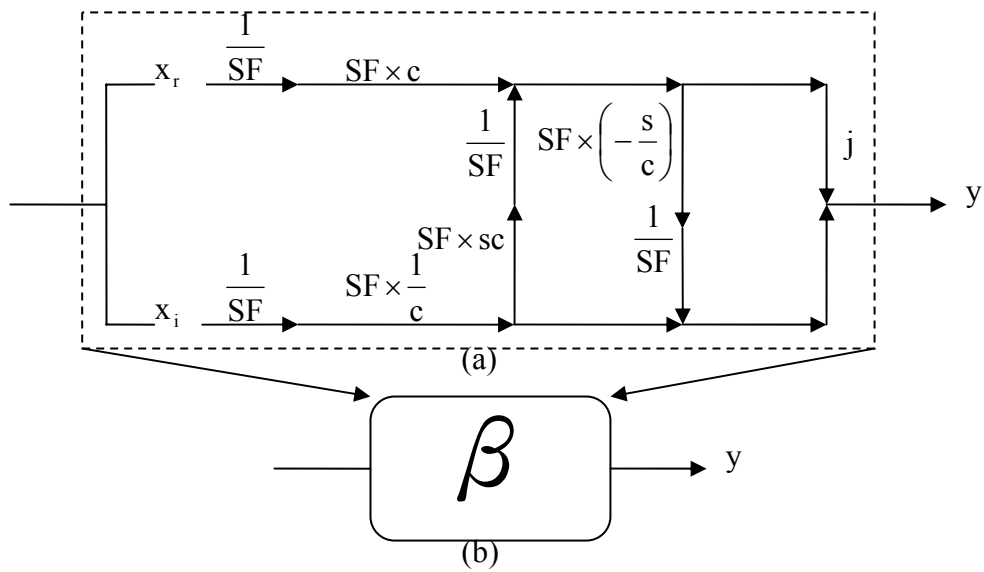


圖 3-7 整數型態 FFT3 複數運算圖形

3.1.3 整數型態 FFT4 [9]

本篇稱[9]的方法為整數型態 FFT4，其優點為 FFT4 在 $N=512$ 時，只要 18 個建表元素(靜態值)，節省許多記憶體容量。所付出的代價就是其它的 W 值(動態值)，要用運算方式算出來，會有誤差發生。

舉一例說明靜態值和動態值的差別：

假設 $SF=1024$ ，只將 $\sin\left(\frac{\pi}{8}\right)$ 和 $\cos\left(\frac{\pi}{8}\right)$ 建入 Table 表中，分別為 392(靜態值)和 946(靜態值)兩個值。若原本在運算時需要第三個值為 $\sin\left(\frac{\pi}{4}\right)$ ，可是並沒有在 Table 表中，就利用三角函數性質 $\sin(2\theta) = 2\sin\theta\cos\theta$ 求出其動態值為 $\frac{2 \times 392 \times 946}{SF} = 724$ (其中因為 392 和 946 分別內含 SF 倍，兩者相乘後則為 SF^2 倍，所以要乘上 $\frac{1}{SF}$ 才可使求出的值內含相同的 SF 倍)。

而 FFT4 就是用類似的方式，每級只需 2 個建表元素(靜態值)，就可求得同一級內所需的 W 值(動態值)，又[9]取 $N=512=2^9$ 共有九級，所以共需 18 個建表元素(靜態值)，如下表所示(其中 Table_c[m]和 Table_s[m]，分別為第 m 級 \cos 和 \sin 的建表值)：

	第 1 級	第 2 級	第 3 級	第 4 級	第 5 級
Table_c[m]	$\cos(0)$	$\cos\left(\frac{\pi}{4}\right)$	$\cos\left(\frac{\pi}{8}\right)$	$\cos\left(\frac{\pi}{16}\right)$	$\cos\left(\frac{\pi}{32}\right)$
Table_s[m]	$\sin(0)$	$\sin\left(\frac{\pi}{2}\right)$	$\sin\left(\frac{\pi}{4}\right)$	$\sin\left(\frac{\pi}{8}\right)$	$\sin\left(\frac{\pi}{16}\right)$
	第 6 級	第 7 級	第 8 級	第 9 級	
Table_c[m]	$\cos\left(\frac{\pi}{64}\right)$	$\cos\left(\frac{\pi}{128}\right)$	$\cos\left(\frac{\pi}{256}\right)$	$\cos\left(\frac{\pi}{512}\right)$	
Table_s[m]	$\sin\left(\frac{\pi}{32}\right)$	$\sin\left(\frac{\pi}{64}\right)$	$\sin\left(\frac{\pi}{128}\right)$	$\sin\left(\frac{\pi}{256}\right)$	

表格 3- 1 FFT4($N=512$)建表元素值

例如要求第3級的cos和sin的建表值，則由表格3-1得知 $\text{Table_c}[3]=\cos\left(\frac{\pi}{8}\right)$ 和 $\text{Table_s}[3]=\sin\left(\frac{\pi}{4}\right)$ 。

接下來說明 W 值(動態值)運算方法，我們將圖2-7中，每級所需的 W 值列出，如下表所示：

	所需 W 值
第1級	$W_N^0 = \cos\left(-\frac{0\pi}{8}\right) + j\sin\left(-\frac{0\pi}{8}\right) = 1 + j0$
第2級	$W_N^0, W_N^2 = \cos\left(-\frac{4\pi}{8}\right) + j\sin\left(-\frac{4\pi}{8}\right) = 0 - j$
第3級	$W_N^0, W_N^1 = \cos\left(-\frac{2\pi}{8}\right) + j\sin\left(-\frac{2\pi}{8}\right) = 0.707 - j0.707$ $W_N^2, W_N^3 = \cos\left(-\frac{6\pi}{8}\right) + j\sin\left(-\frac{6\pi}{8}\right) = -0.707 - j0.707$

表格3-2 FFT (N=8)所需 W 值列表

在上表第1級中，所需 W 值只有 $(1+j0)$ ，可以直接由表格3-1中得知，實數部= $\text{Table_c}[1]=1$ 和虛數部= $\text{Table_s}[1]=0$ 。第2級所需的值為 $(1+j0)$ 和 $(0-j)$ ，因為 $(1+j0)$ 在第1級時已求出，所以在第2級只要求出 $(0-j)$ 即可；實數部只要取表格3-1中，第2級 $\text{Table_c}[2]=\cos\left(\frac{\pi}{4}\right)$ ，並利用三角函數的倍角公式求出 $\cos\left(\frac{\pi}{2}\right)=0$ ，而虛數部則直接由表格3-1的 $\text{Table_s}[2]=1$ ，再取負號。用上述方式可求得 FFT4 第1級到第9級所有的 W 值(動態值)。

以下為 FFT4 在第3級時，所需 W 值(動態值)的實際運算步驟：

圖3-8中初值 $c'=c=1, s'=s=0, \varepsilon=\text{Table_c}[m], \eta=\text{Table_s}[m], W=c+js$ ，虛線箭頭代表同一級作循環運算時，設定 $c'=c, s'=s$ (也就是將輸出的結果，放到輸入端去作循環運算)。

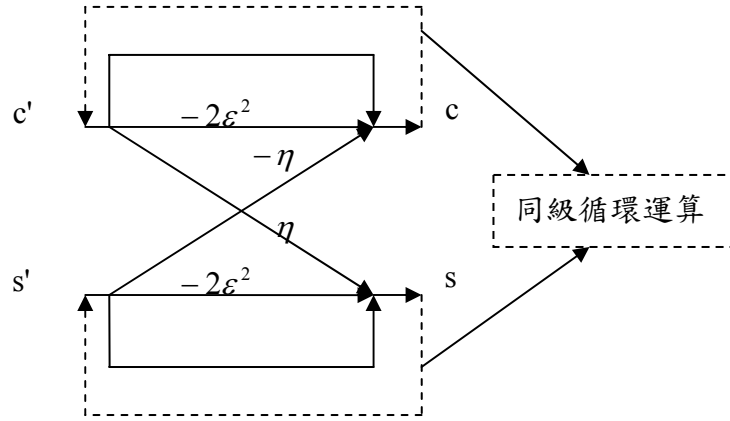


圖 3- 8 FFT4 W 值(動態值)運算圖形

首先設定初值 $c' = c = 1$ 和 $s' = s = 0$ ，接下來設定 ε, η ，由表格 3- 1 得知第 3 級的 $\varepsilon = \cos\left(\frac{\pi}{8}\right) = 0.9239$ 和 $\eta = \sin\left(\frac{\pi}{4}\right) = 0.707$ 。完成設定後利用上圖的運算方式，開始進行運算，其過程如下。

第 3 級第一次運算：

$$\begin{aligned}
 c &= c' \times (-2\varepsilon^2) + s' \times (-\eta) + c' \\
 &= 1 \times (-2 \times 0.9239^2) + 0 \times (-0.707) + 1 = -0.707 \\
 s &= s' \times (-2\varepsilon^2) + c' \times (\eta) + s' \\
 &= 0 \times (-2 \times 0.9239^2) + 1 \times (0.707) + 0 = 0.707
 \end{aligned}$$

第 3 級第二次運算：

運算方式同上，不同點為進入循環運算。由圖 3- 8 中虛線得知

$$c' = c = -0.707, s' = s = 0.707。$$

$$\begin{aligned}
 c &= c' \times (-2\varepsilon^2) + s' \times (-\eta) + c' \\
 &= -0.707 \times (-2 \times 0.9239^2) + 0.707 \times (-0.707) - 0.707 = 0 \\
 s &= s' \times (-2\varepsilon^2) + c' \times (\eta) + s' \\
 &= 0.707 \times (-2 \times 0.9239^2) - 0.707 \times (0.707) + 0.707 = -1
 \end{aligned}$$

第 3 級第三次運算：

$$c' = c = 0, s' = s = -1$$

$$c = 0 \times (-2 \times 0.9239^2) - 1 \times (-0.707) + 0 = 0.707$$

$$s = -1 \times (-2 \times 0.9239^2) + 0 \times (0.707) - 1 = 0.707$$

由一開始的初值設定 $c=1, s=0$ 和三次運算結果，可得到 4 個 W 值(動態值)分別為 $(1+j0)$ 、 $(-0.707+j0.707)$ 、 $(0+j1)$ 和 $(0.707+0.707)$ ，將 W 值(動態值)和圖 2-7 中第 3 級所需的 W 值(表格 3-2)作比對後， W 值(動態值)和表中的 W 值一樣。對於有正負號差異的 W 值(動態值)，只要用程式加負號即可。

3.2 改進方法(整數型態 FFT5)

首先由 FFT4 可知其優點為建表元素個數只需 18 個(靜態值)，但是利用原先在建表時，已有誤差的 W 值(靜態值)，去求其它 W 值(動態值)則誤差會更大。所以本篇提出的 FFT5 優點，就是不用有誤差的 W 值(動態值)來作運算，全部用 Table 表內的 W 值(靜態值)，來增加運算時的精確度。而所付出的代價就是建表元素在 $N=512$ 時，比 FFT4 多 110 個。接下來介紹 FFT5 的方法：

首先求出 FFT4 中所有 W 動態值和 W 靜態值共 512 個，其範圍為 $\sin\left(\frac{n}{256} \times \pi\right)$ 和 $\cos\left(\frac{n}{256} \times \pi\right)$ 其中 $n=0,1,2,\dots,255$ 。若我們直接建表則需要建兩個表，分別為 \sin Table 和 \cos Table 各 256 個共 512 個建表元素，對於記憶體要求會大了一些，所以要進行簡化。

由三角函數性質得知 $\cos(\theta) = \sin\left(\frac{\pi}{2} - \theta\right)$ ，所以 $\cos(\theta)$ 的值全部可由 $\sin(\theta)$ 來代替，如此一來就只要建一個 \sin Table 表共 256 個元素。又 $\sin\left(\frac{n}{256} \times \pi\right)$ 的角度

範圍為 $0 \sim \frac{255\pi}{256}$ ，再利用 $\sin(\theta) = \sin(\pi - \theta)$ 的性質，其中 $\left(0 \leq \theta \leq \frac{\pi}{2}\right)$ ，使角度範圍縮小到 $\left(0 \sim \frac{\pi}{2}\right)$ 之間。最後只需用到 $\sin\left(\frac{n}{256} \times \pi\right)$ ，其中 $n = 0, 1, 2, \dots, 127$ ，共 128 個建表元素，和未作簡化前的 512 個建表元素的數量差了四倍。

接下來我們將圖 3-2 中的參數 s 乘上 SF ，建成 Table 表。因為 x_r 和 x_i 已內含 SF 倍，所以在運算時，都先乘上 $\frac{1}{SF}$ 將其倍數還原成 1 後再作運算，如下圖(a)所示。再將(a)簡化成(b)的圖形。因為只用一個 Table 表， c 值只要在 \sinTable 表中取相對位置的值即可，因此(a)圖中用 s' 來代替 c 。

再將圖 2-7 內的 W 置換成下圖(b)，結果和圖 3-5 相似。

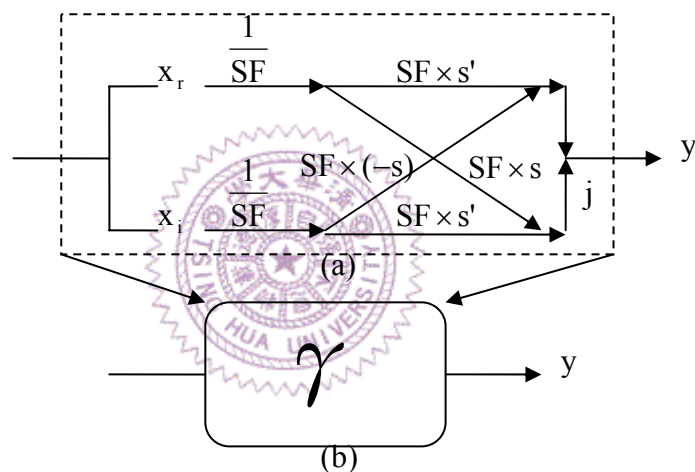


圖 3-9 複數乘法簡化圖形