第三章　封包取樣點的同步

本章節將接續著前一章，繼續解說硬體的設計與實現。在前一章中，我們介紹的是封包的偵測和頻率偏移的估計與補償；而在本章節中，我們將會介紹如何做封包取樣點的同步，其中包含了短調整符元與短調整符元間界線的同步，以及循環字首長度的估計這兩個部分；同樣的，我們會從問題的成因開始說明，然後是演算法，最後是硬體設計的概念和實現。

3.1 取樣點的同步

在第二章的一開始，我們已經介紹過如何偵測是否有封包在介質中傳輸，但是，其機制只能用於偵測封包的存在與否，卻無法對於封包取樣點位置的判斷有任何的助益，其原因在於之前封包的偵測採用的是正相關函數的演算法，不論目前接收機所收到訊號的位置如何，只要與其相對應的取樣點具有週期的特性，我們便可以利用在第二章的演算法求得封包存在的依據；本來這對封包的偵測來說，是個好消息，但是，我們也因此無法得知目前取樣點的位置；所以，本節的重點便在於如何同步封包的取樣點；若是同步作得好，那便可以正確的判斷出正交分頻多工符元區塊的位置，因此在作快速傅立葉轉換時，不會因為取錯區間而造成不同區塊間的干擾(Inter-Block-Interference，IBI)。

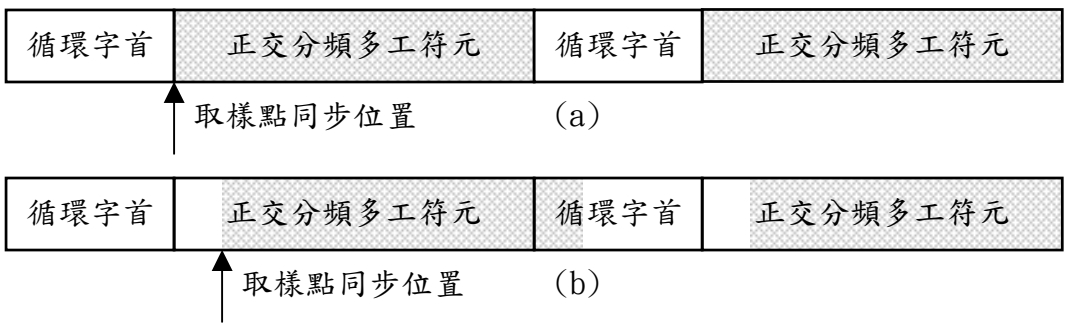


圖 3.1.1 取樣點同步位置 (a)正確、(b)錯誤

上圖是取樣點同步位置的示意圖，在圖(a)中，取樣點同步位置完美的和正交分頻多工符元的起始位置匹配，所以我們可以取得完整的符元作反傅立葉轉換，以獲得正確的資料而不會有不同區塊間的干擾產生；但是在圖(b)中，因為取樣點同步位置的錯誤，所以整個選取到的正交分頻多工符元區塊，橫跨了兩個不同的正交分頻多工符元，因此在經過反傅立葉轉換後，會有不同區塊間的干擾產生，而大幅提高了解調後資料的錯誤率；所以封包的取樣點位置必須準確，這正是需要作取樣點同步的原因。

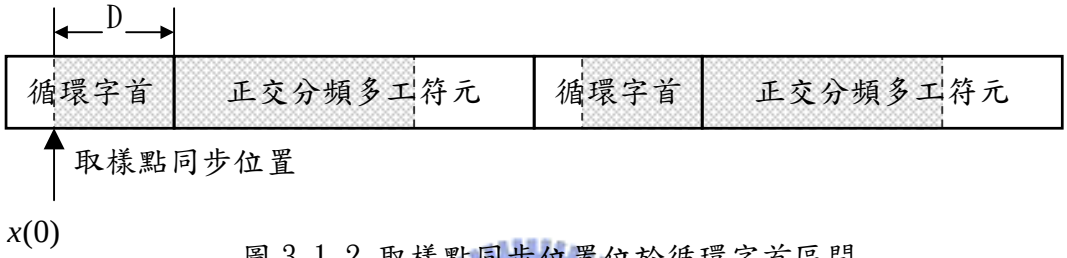


圖 3.1.2 取樣點同步位置位於循環字首區間

接著我們來看第三種取樣點同步位置的示意圖；上圖(3.1.2)中的取樣點位置位於循環字首內，而且和正確的正交分頻多工符元的起始位置有 D 個取樣點的差距，這樣的取樣點位置雖然不是正確的，但是因為沒有橫跨兩個不同的正交分頻多工符元，所以不會有不同區塊間的干擾；下式(3.1.1)所表示的是如圖(3.1.2)的起始取樣位置，經過反快速傅立葉轉換後所得到的資料，其中 $x(0)$ 是取樣點同步位置開始的第一個取樣點、N 是整個正交分頻多工符元的取樣點個數：

$$X[k]=\sum_{n=0}^{N-1}x(n)\cdot e^{j\frac{2\pi}{N}kn}$$
$$=\sum_{n=0}^{D-1}x(n)\cdot e^{j\frac{2\pi}{N}kn}+\sum_{n=D}^{N-1}x(n)\cdot e^{j\frac{2\pi}{N}kn} \qquad \text{式 3.1.1}$$

在第一章中，我們說過循環字首是該對應正交分頻多工符元的結尾的複製，所以 $x(0)$ 和 $x(N)$ 應該是相等的，也就是：

$$x(k)=x(N+k) \qquad k=0,\quad 1,\quad .....D-1 \qquad \text{式 3.1.2}$$

所以我們可以將數學式(3.1.1)改寫為下式：

$$
\begin{aligned}
X[k] &= \sum_{n=0}^{N-1} x(n)\cdot e^{j\frac{2\pi}{N}kn} \\
&= \sum_{n=0}^{D-1} x(n)\cdot e^{j\frac{2\pi}{N}kn} + \sum_{n=D}^{N-1} x(n)\cdot e^{j\frac{2\pi}{N}kn} \\
&= \sum_{n=D}^{N-1} x(n)\cdot e^{j\frac{2\pi}{N}kn} + \sum_{n=0}^{D-1} x(N+n)\cdot e^{j\frac{2\pi}{N}kn} \\
&= \sum_{n=0}^{N-1-D} x(n+D)\cdot e^{j\frac{2\pi}{N}k(n+D)} + \sum_{n=N-D}^{N-1} x(n+D)\cdot e^{j\frac{2\pi}{N}k(n+D)} \\
&= \sum_{n=0}^{N-1} x(n+D)\cdot e^{j\frac{2\pi}{N}k(n+D)} \\
&= \left(\sum_{n=0}^{N-1} x(n+D)\cdot e^{j\frac{2\pi}{N}kn}\right)\cdot e^{j\frac{2\pi}{N}kD} \\
&= X_0[k]\cdot e^{j\frac{2\pi}{N}kD} \qquad \text{式 3.1.3}
\end{aligned}
$$

其中，$X_0[k]$是在圖 3.1.1(a)中，以正確位置所得到的取樣點、經過反快速傅立葉運算後所得到結果。

由上式(3.1.3)中，我們可以發現：當取樣點的起始位置若是位於循環字首內時，其反快速傅立葉運算後的結果會與理想起始位置所求得的結果，僅在複數平面上相差一個旋轉角度，而這旋轉角度的效應，我們可以等同視為通道特性的影響，在補償通道特性時，便會一起消弭；關於通道特性的補償，我們將會在第四章中，作深入的介紹。因此在作封包取樣點同步的時候，只需要將我們所認為的正交分頻多工符元的起始位置，同步於實際符元的循環字首區間內，如此便達到我們同步封包取樣點的目的。

3.1.1 短調整符元同步的演算法

在前一章節中，我們已經說明了封包取樣點同步的重要性，若是同步作的不好將會產生不同區塊間的干擾，進而造成解調後資料的錯誤率大增；若要有良好的同步，便要將所取正交分頻多工符元的起始位置落於理想符元的循環字首區間內，而在本論文中，我們將利用匹配濾波器來達到這個目的。

由於短調整符元是傳送端與接收端都已經溝通好的資料，所以我們可以利用這個資料已知的特性，設計出其相對應的匹配濾波器；當有一個完整而且起始位置與匹配濾波器係數相同的短調整符元送入此濾波器後，我們將可以發現有一個峰值(Peak Value)產生。換句話說，當我們一直把接收到的取樣點送入匹配濾波器後，若在某一個取樣點造成了一個峰值，那便表示該取樣點所在的位置便是短調整符元的起始位置，而我們便可達到封包同步的目的。

理論上這樣是可行的，但是在實際的實現上，這樣有很大的難度，其中最大的問題在於一個短調整符元的長度是兩百五十六個取樣點，也就是說，我們必須要等這兩百五十六個取樣點都送入匹配濾波器後，才有可能獲得一個峰值的產生，可是在判斷是否為峰值的同時，接收機還是在持續的接收資料的取樣點；換句話說，可能會發生當接收機判斷出正確的起始位置時，已經損失數個長調整符元取樣點的情形，而不利於將來循環字首長度的估計和通道特性的評估。因此在實際的硬體設計上，我們還利用了短調整符元每六十四點會有一週期的特性，將原本需要兩百五十六個係數的匹配濾波器，化簡為六十四點係數的匹配濾波器。如此既可以簡化硬體的需求，同時又可以提早同步封包的取樣點，以利於後續的工作；但是相對的，這會造成不易判斷出長調整符元的起始位置，這問題將在本章 3.2 節中獲得解決。下圖(3.1.3)是取六十四點係數的匹配濾波器，其輸出結果的示意圖：

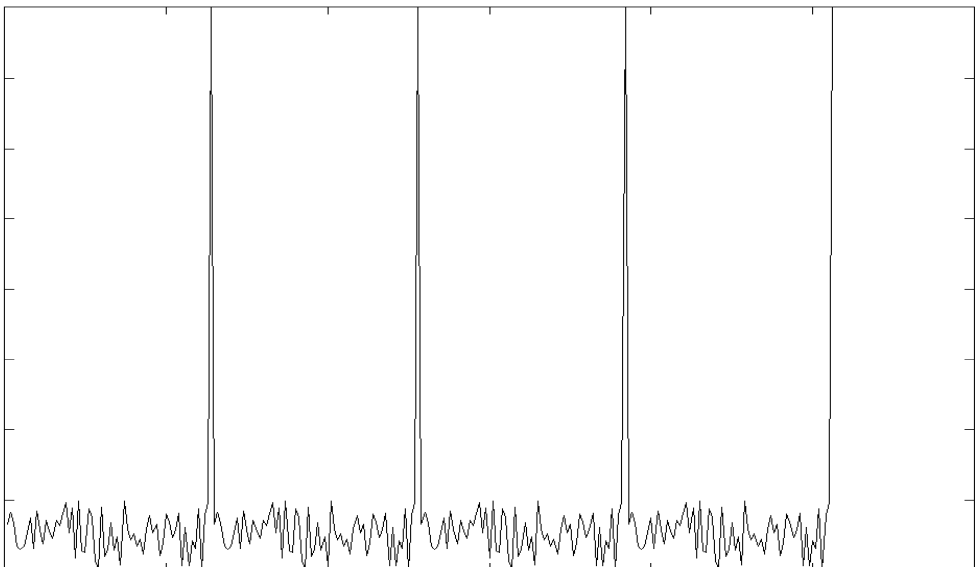
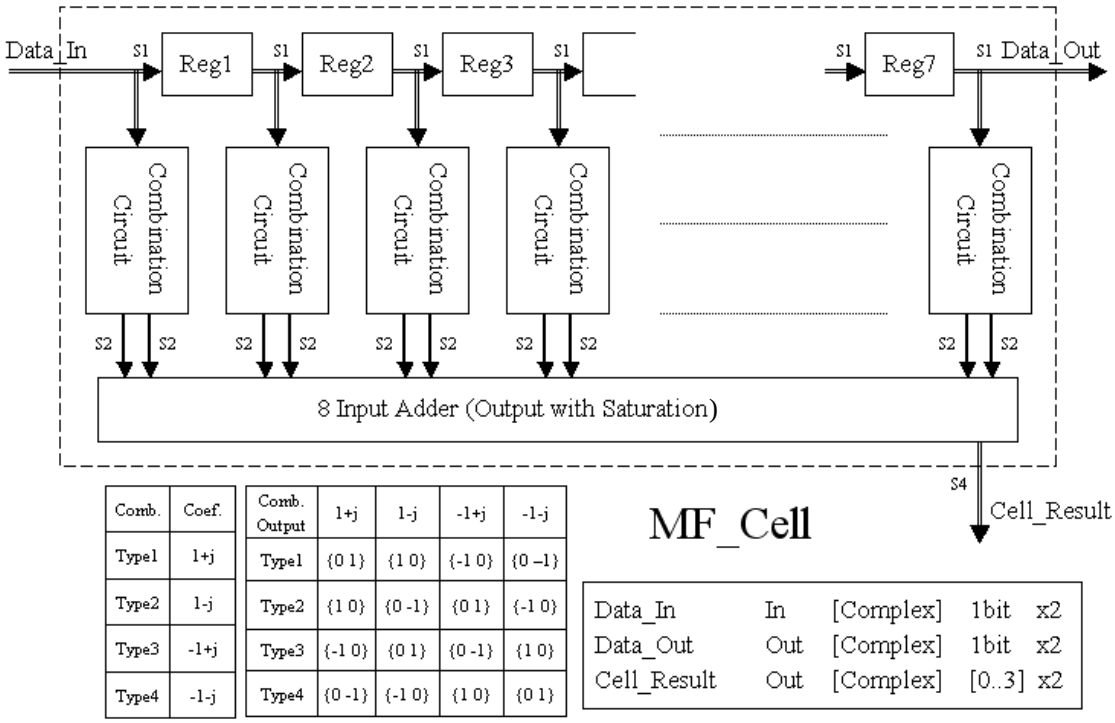


圖 3.1.1.1 短調整符元匹配濾波器輸出示意圖

3.1.2 短調整符元同步的硬體架構

承前所述，我們將使用匹配濾波器來作封包的同步；所謂的匹配濾波器，其實跟一般的濾波器一樣，都是由一些乘加器組合而成的，只是在係數上會選取特定與短調整符元匹配。本論文在硬體的設計上，也是以此為目標，但是為了節省硬體的負擔，所以我們希望將乘法器能簡化成一些簡單的組合電路。我們使用的方法是對接收機取樣到的資料、以及匹配濾波器的係數，取其符號位元(Sign Bit)來簡化乘法器，將原本濾波器所需要的乘法器，依照所乘的係數化簡為四種組合電路中的其中一個，如下圖(3.1.2.1)所示：



Comb.	Coef.
Type1	1+j
Type2	1-j
Type3	-1+j
Type4	-1-j

Comb. Output	1+j	1-j	-1+j	-1-j
Type1	{0 1}	{1 0}	{-1 0}	{0 -1}
Type2	{1 0}	{0 -1}	{0 1}	{-1 0}
Type3	{-1 0}	{0 1}	{0 -1}	{1 0}
Type4	{0 -1}	{-1 0}	{1 0}	{0 1}

圖 3.1.2.1 匹配濾波器單元的硬體架構圖

除了匹配濾波器以外，我們亦需要有個機制來判別匹配濾波器的輸出是否為一個峰值，在硬體的實踐上，我們利用一個記數器(Counter)和暫存器(Register)來達成我們的目的；下頁圖(3.1.2.2)即是整個短調整符元同步的硬體設計，圖中的上半部分即是匹配濾波器的架構，而下半部分便是記數器與暫存器的組合，用以完成判別峰值的機制。

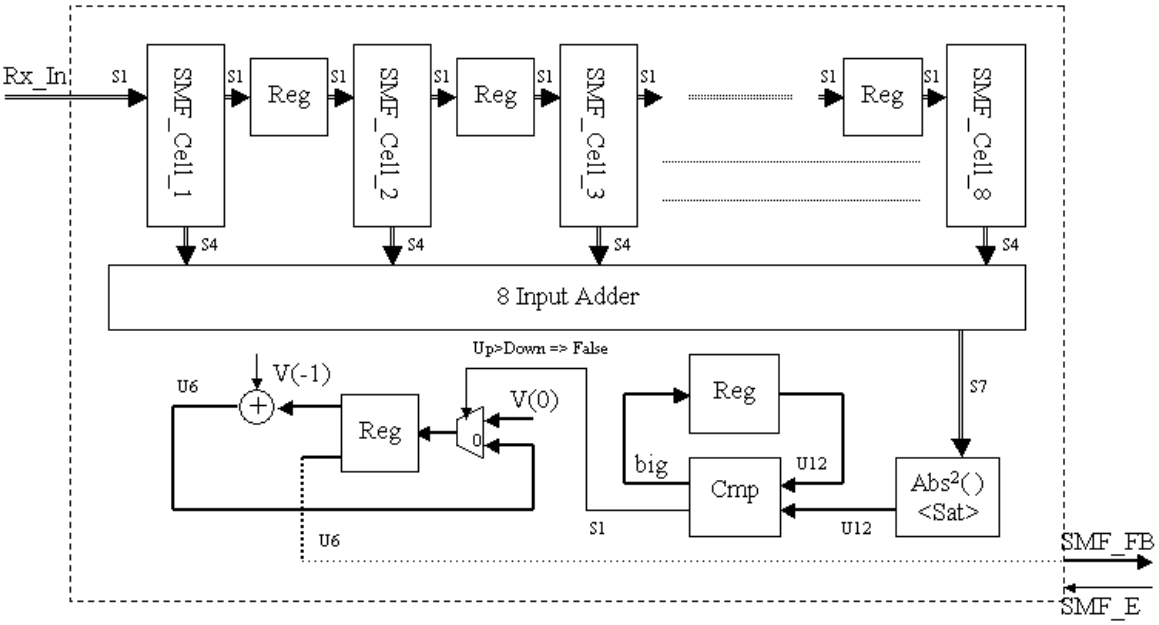


圖 3.1.2.2 短調整符元同步的硬體架構圖

3.1.2.1 輸入輸出以及控制訊號

資料輸入訊號：

■ Rx_In：沒經過頻率偏移補償的資料取樣點，有實部、虛部訊號之分

資料輸出訊號：

■ SMF_FB：傳出目前接收機取樣點的位置與下次峰值產生的位置的差距

控制訊號：

■ SMF_E：輸入控制訊號，控制短調整符元同步硬體是否運作

3.1.2.2 乘法替代組合電路

承本章 3.1.2 節所述，我們將利用組合電路來替代乘法器的使用，因為本身匹配濾波器的係數亦有實部與虛部之分，所以雖然我們只取其符號位元來作運算，但是這樣亦需要 2×2＝4 種的組合電路，來完成原本乘法器所達成的機制；下表便是六十四組匹配濾波器係數與四種組合電路之一相對應的表格：

#th Coef.	Type	#th Coef.	Type	#th Coef.	Type	#th Coef.	Type
1	IV	17	IV	33	II	49	III

2	III	18	II	34	I	50	I
3	II	19	I	35	IV	51	IV
4	III	20	I	36	IV	52	III
5	III	21	IV	37	III	53	III
6	III	22	II	38	I	54	IV
7	I	23	I	39	II	55	III
8	I	24	IV	40	IV	56	I
9	III	25	IV	41	II	57	I
10	II	26	I	42	II	58	I
11	II	27	II	43	I	59	II
12	III	28	II	44	III	60	I
13	IV	29	I	45	IV	61	IV
14	III	30	I	46	IV	62	IV
15	III	31	I	47	III	63	I
16	IV	32	II	48	III	64	II

表 3.1.2.2 乘法替代組合電路與短調整符元在時序上的關係

3.1.2.3 硬體運作的機制

在本章 3.1.2 節中，我們說過要同步封包的取樣點必須要有兩個機制：第一、需要一個匹配濾波器，用來知道在此之前的輸入取樣點是否與短調整符元匹配；第二、需要一個硬體來判斷匹配濾波器的輸出結果是否為一個峰值，若是峰值的話，表示此次輸入的取樣點為一個短調整符元內、週期區間的起始位置；若否，則此硬體亦必須要傳出其預估到達下次起始位置的取樣點個數。

圖(3.1.2.2)便可完成上述的兩個機制；在圖中的上半部分即是一個匹配濾波器，當一個取樣點輸入後，便會經過此匹配濾波器而得到包括此次取樣點在內

的前六十四個取樣點與短調整符元匹配係數的迴旋積(Convolution)結果；接著對此複數結果取絕對值平方，以獲得其大小；然後經過一個比較器，此比較器用於判斷剛獲得迴旋積結果的大小是否為一個峰值，若迴旋積結果的大小比暫存器內所存放的值來得大時，表示此次迴旋積結果的大小為一個峰值，並更新暫存器：將新得到的峰值存入其中，同時將接在比較器後方的記數器歸零，以表示此次輸入的取樣點和起始位置的差距為零。相反的，當比較的結果比暫存器內的值來得小時，表示此次迴旋積結果的大小非一個峰值，所以暫存器保持不變，但是後接於比較器的記數器將會減一，用以表示向下一個起始位置前進了一個取樣點；所以記數器所輸出的值可以看成是距離下一個短調整符元內、週期區間的起始位置的差距，而知道了這個差距，我們便可以很容易的達到短調整符元同步的目的。

3.1.2.4 字元長度的選擇

一開始我們先從乘法替代的組合電路看起；如同之前所述：我們只取資料以及係數的符號位元作乘法，而又資料以及係數皆為複數，所以取符號位元後，資料以及係數都可能會有四種組合：$(1+j)$、$(1-j)$、$(-1+j)$、$(-1-j)$；因此理論上會有$4\times4=16$種的組合，可是由於資料特性的影響，原本十六種的組合有十二組是重複的，實際上只有四種結果：$(2+0j)$、$(-2+0j)$、$(0+2j)$、$(0-2j)$；接著，因為這四種結果都有二的公因數，所以同除以二亦不會對結果造成問題，因此當一個資料取符號字元經過乘法替代的組合電路後，其實部或是虛部只可能有三種可能{1,0,-1}，因此在這個乘法替代的組合電路，我們用兩位元來表示其實部或是虛部的輸出結果，如圖(3.1.2.1)。

接著我們來看 MF_Cell 的輸出，其輸出結果其實是八個乘法替代組合電路輸出的和，所以理論上我們需要$2+\log_2(8)=5$個字元來表示其結果，但由於我們考量到實際上這八個乘法替代組合電路輸出的和只可能落在區間[8,-8]裡，用五個字元來表示會過於浪費，因此我們加入一個飽和(Saturation)的機制，若 MF_Cell 輸出的結果超過四個字元所能表示的大小，便以四個位元能表示的最大正值或是負值來表示，藉此簡化硬體的需求。

然後我們來看經過運算元—絕對值平方後的結果所需要的位元數；此運算元的輸入為八個 MF_Cell 的和，所以輸入的實部或是虛部皆需要 $4+\log_2(8)=7$ 個位元表示，所以理論上經過絕對值平方後會需要十三個字元，方可把所有的結果儲存於其中，這跟在封包偵測裡的推導是一樣的；但是，實際上我們不需要使用那麼多的字元就可以儲存所有經過絕對值平方運算後的結果，其中最大的原因在於當初乘法替代的組合電路輸出只有 $(1+0j)$、$(-1+0j)$、$(0+j)$、$(0-j)$ 這四種可能，使得此絕對值平方運算元的輸入若為 $(A+B\cdot j)$ 的話，需滿足下式：

$$-64 \le A, B \le 56$$
$$|A|+|B| \le 64 \qquad \text{式 3.1.2.4}$$

因此利用線性規劃的方法，我們可以發現當實部或是虛部為負六十四的時候會有最大值，因此我們可以用十二個字元即可將所有可能的輸出表示，如圖(3.1.2.2)所示。

3.1.3 長調整符元循環字首的同步

所謂的長調整符元循環字首的同步，便是要找到短調整符元和長調整符元循環字首間的邊界；承前，因為我們在作短調整符元的同步時，採用的是六十四點、短調整符元內、週期區間的係數，所以我們只可以找到此週期區間的起始位置，卻無法找到短調整符元和長調整符元循環字首間的邊界，因此我們需要一個方法來界定邊界的所在；使用的方法是利用短調整符元和長調整符元在時域上相關性薄落的特性，如下圖所示：

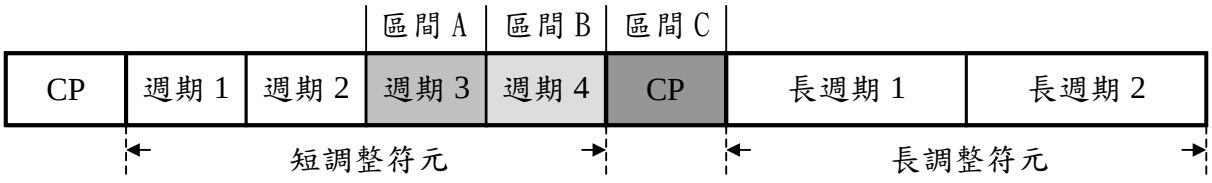


圖 3.1.3.1 短調整符元與長調整符元示意圖

圖中區間 A 和區間 B 為短調整符元內、週期六十四的區塊；而區間 C 為長調整符元循環字首開始算起的前六十四個取樣點；由於短調整符元和長調整符元

取樣點的相關性薄落，所以區間 B 與區間 C 的正相關函數值(Autocorrelation Value)，一定會小於區間 A 與區間 B 的正相關函數值。因此我們可以設定一個 Threshold，若區間 B 與區間 C 的正相關函數值 mp_{BC} ，小於區間 A 與區間 B 的正相關函數值 mp_{AB} 再乘以 Threshold 的話，即表示找到了短調整符元與長調整符元的邊界；如下式(3.1.3.1)所示：

$$mp_{BC} \le mp_{AB} \cdot Threshold \qquad \text{式 3.1.3.1}$$

在第二章的封包偵測裡，我們已經說明過如何找出兩連續區間的正相關函數值，但是，在實際的硬體設計上，其實我們並未求得 mp 值的大小，而是利用一個乘法器來替代求 mp 值所需要的除法器；因此為了比較 mp_{AB} 和 mp_{BC} 兩者大小的差異，我們勢必要將上述的演算法變形，如此才能符合我們的硬體設計，下式(3.1.3.2)是長調整符元不等式(3.1.3.1)的變形：

$$\begin{aligned} & mp_{BC} \le mp_{AB} \cdot Threshold \\ \Rightarrow & \frac{\left|Cn_{BC}\right|^2}{\left|Pn_{BC}\right|^2} \le \frac{\left|Cn_{AB}\right|^2}{\left|Pn_{AB}\right|^2} \cdot Threshold \\ \Rightarrow & \left|Cn_{BC}\right|^2 \cdot \left|Pn_{AB}\right| \le \left|Cn_{AB}\right|^2 \cdot \left|Pn_{BC}\right|^2 \cdot Threshold \end{aligned} \qquad \text{式 3.1.3.2}$$

因此，只需要利用交叉相成的概念，便可以使用原本封包偵測的硬體架構，判斷出長調整符元不等式的成立與否，進而決定出短調整符元與長調整符元間的邊界。在實際的硬體實現當中，Threshold 的設定為 0.5；一方面滿足一般的環境特性，另一方面在硬體的實現上，只需將字元右移一位元，便可以替代原本所需的常數乘法器，可說是一舉兩得。

3.2 循環字首長度的估計

在本論文的第一章中，我們曾經提過：循環字首的長度並非是一個固定的常數，其長度可以為八點、十六點、三十二點或是六十四點，端看當時媒體接取層(MAC)的設定。這對上層的設計者，是提供了多種的選擇以面臨多變化的環境；

但是對於硬體的設計上來說，卻帶來了同步上的麻煩。最大的原因是因為在尚未解調出任何資料正交分頻多工符元的我們，是無法得知當初傳送端所設定的循環字首大小，在處裡的時序上這是無法達成的。因此我們必須要有一個機制，在不用解調出資料符元的情況下，利用接收到的取樣點，決定出循環字首長度的大小；這亦屬於封包取樣點同步的一環，因為若循環字首長度的大小估計錯誤，便將無法選取到正確正交分頻多工符元的區間，以作反快速傅立葉運算，同時不同區塊間干擾(Inter-Block Interference)的問題，依會接踵而來。

3.2.1 估計循環字首長度的演算法

前一節中，我們已經概述了問題的成因，接著我們來探討要如何解決。解決的方法，我們同樣是採用匹配濾波器的概念，用以找尋長調整符元和與該長調整符元對應的循環字首的邊界，即找尋下圖(3.2.1.1)中的邊界 L：

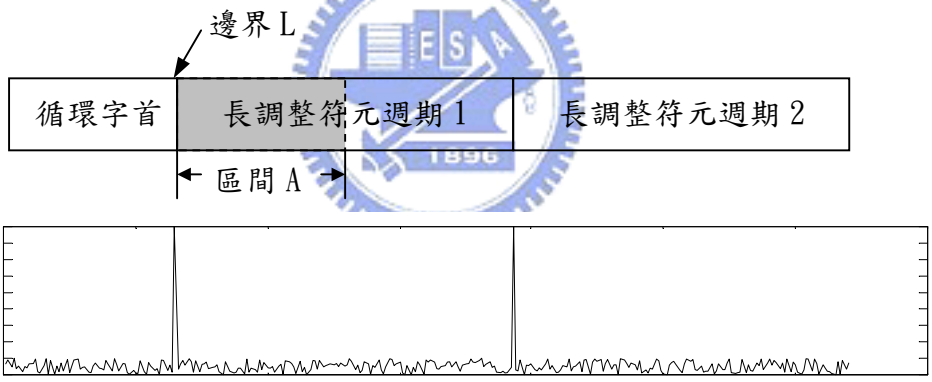


圖 3.2.1.1 長調整符元示意圖與匹配濾波器的輸出結果

接著，在上圖中若我們選取區間 A，這個事前已知的時域訊號作匹配濾波器的係數，並將此時接收機所取樣到的長調整符元或是其循環字首的取樣點，經過該匹配濾波器，便可以得到圖(3.2.1.1)中下半部分的結果。由於我們匹配濾波器是選取與區間 A 匹配作為其係數的依據，所以我們可以發現，經由匹配濾波器輸出後的結果，將會在邊界 L 上產生峰值；此時再配合著前一節長調整符元循環字首的同步，我們很容易可以找到長調整符元循環字首的起始位置，和邊界 L 的所在位置，而這兩者間距離的差距，便是我們想要求得的循環字首的長度。

上面所述，是在理想的環境下才有可能發生的事情；但是在實際的過程中，由於多重路徑(Multiple Path)的影響，造成循環字首長度的判斷不易，下圖(3.2.1.2)是在考慮到多重路徑下，匹配濾波器輸出的結果：

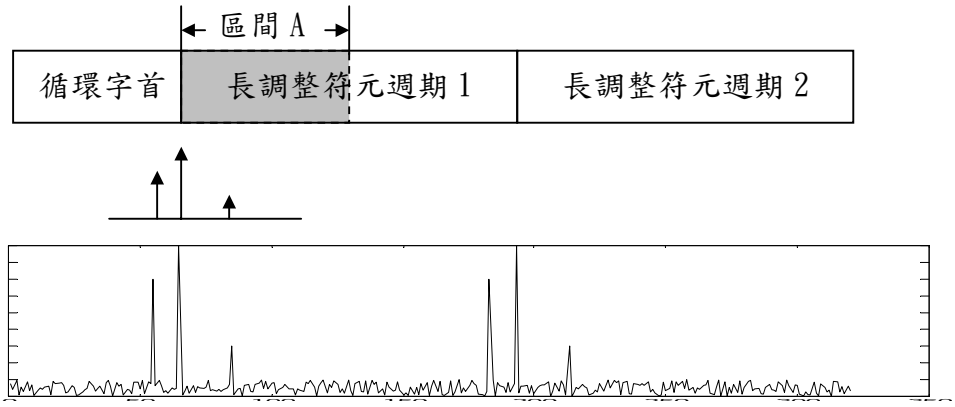


圖 3.2.1.2 多重路徑下匹配濾波器的輸出結果

在圖的中間部分，就是多重路徑所造成影響，而這影響主要來自於訊號的散射或是繞射，使得原本該接收到的訊號會有早到或是晚到的現象；因此，匹配濾波器的輸出將不會像之前一樣只有一個峰值，而會有多根峰值的產生，這會使得循環字首的長度無法估計出來，因為我們無法得知哪一個峰值距離理想的位置最為接近；但是所幸，循環字首的長度雖然不是固定的，可是其也並非是任意的數值，循環字首的長度只有八點、十六點、三十二點或是六十四點這四種可能，因此我們可以對匹配濾波器的每一根峰值，計算其與上述四種可能的距離差異，然後選取最小差異的可能，作為此峰值所求得的可能循環字首長度，最後統計同類可能循環字首長度的個數，選取最多可能的結果，作為估計出來的循環字首長度。除了上述的機制外，我們還可以利用匹配濾波器的輸出結果，對該峰值所對應到的可能循環字首長度作加權的動作，如此一來便可以大幅降低字首長度誤判的機會。

3.2.2 估計循環字首長度的硬體架構

在前一章節中，已經說明硬體設計所採用的演算法，接著我們來看如何在硬體上實現。下圖(3.2.2.1)便是整個估計循環字首長度的硬體架構；在整個硬體架構中的上半部分，就如同短調整符元同步一般，是一個匹配濾波器，其架構的雛型跟圖(3.1.2.2)裡的上半部、短調整符元匹配濾波器是同樣的，差別只在於濾波器的係數不同；也就是說 MF_Cell 裡面使用的乘法替代組合電路，會因為短調整符元和長調整符元本身在時域上的差異，而選取不同的型態，但其他的部分其實是一樣的，因此我們就不再說明，詳細的情形可以參照本章 3.1.2 節。

接著在圖中的下半部分，其實和之前短調整符元同步的架構雷同，基本上就是一個暫存器和一個記數器的組合；暫存器是用來儲存峰值的大小，而記數器用來計算此峰值所估計出來的循環字首長度，跟之前不同之處，僅在於暫存器和記數器使用個數的不同；在估計循環字首長度的硬體裡，因為我們必須要考量到多重路徑所帶來的問題，因此我們必須要儲存多個峰值以減小誤判長度的機會，而每一個峰值便需要一個暫存器儲存、一個記數器計算字首長度，這就是造成需要數個暫存器和記數器的原因；在實際的硬體設計上，我們總共採用七組暫存器和記數器的組合，如此一來便可抵抗七根以下的多重路徑所造成的干擾。

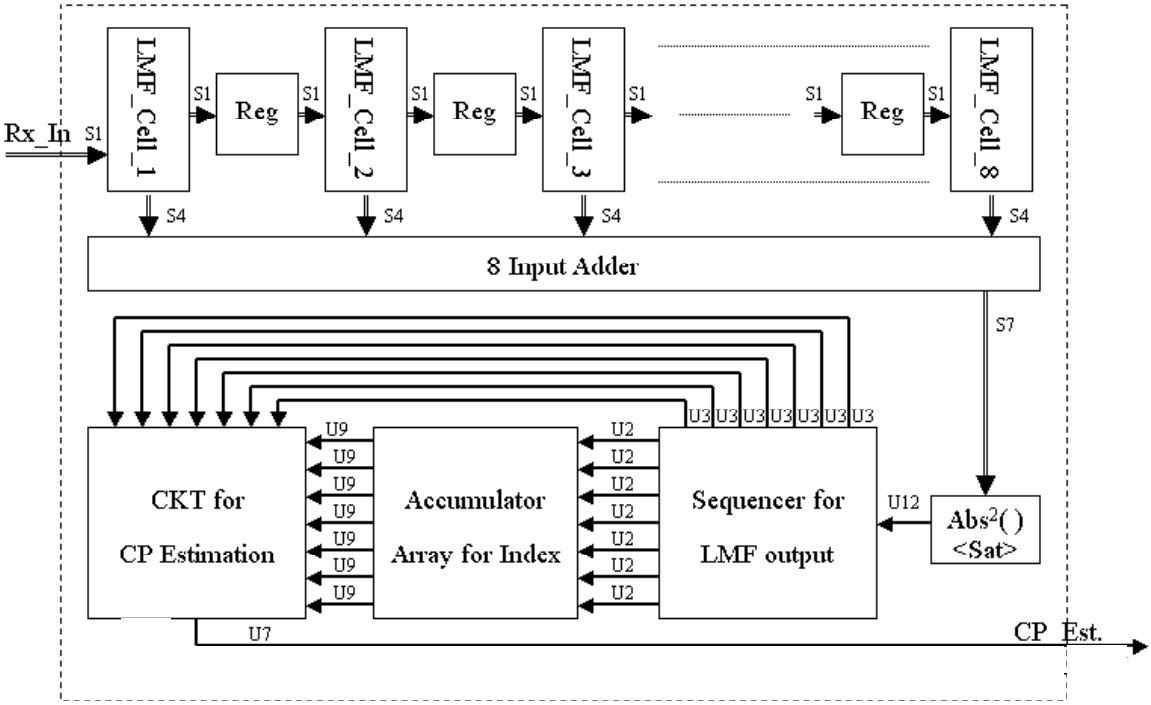


圖 3.2.2.1 估計循環字首長度的硬體架構圖

最後的一部分就是一些組合電路的合成，主要的目的在於將前一節計算可能的循環字首長度的個數和利用匹配濾波器的輸出作加權的機制至於其中，如圖(3.2.2.3)所示。

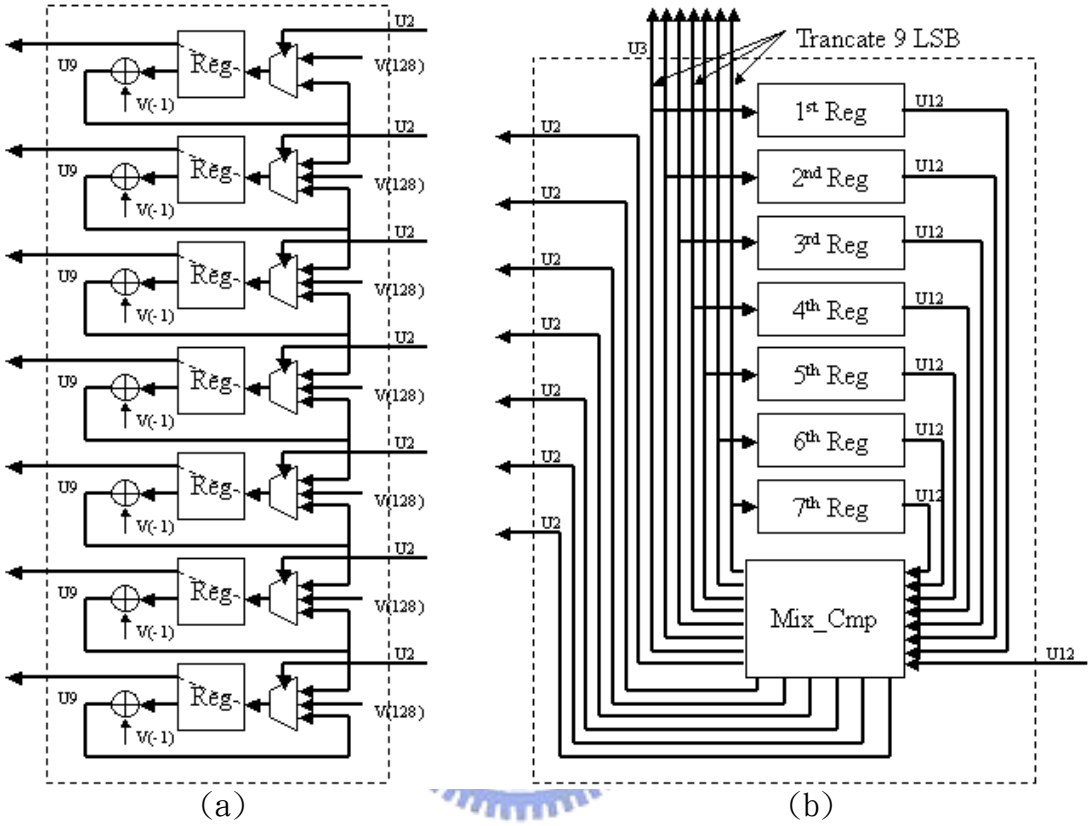


圖 3.2.2.2 (a)記數器、(b)暫存器與比較器的組合

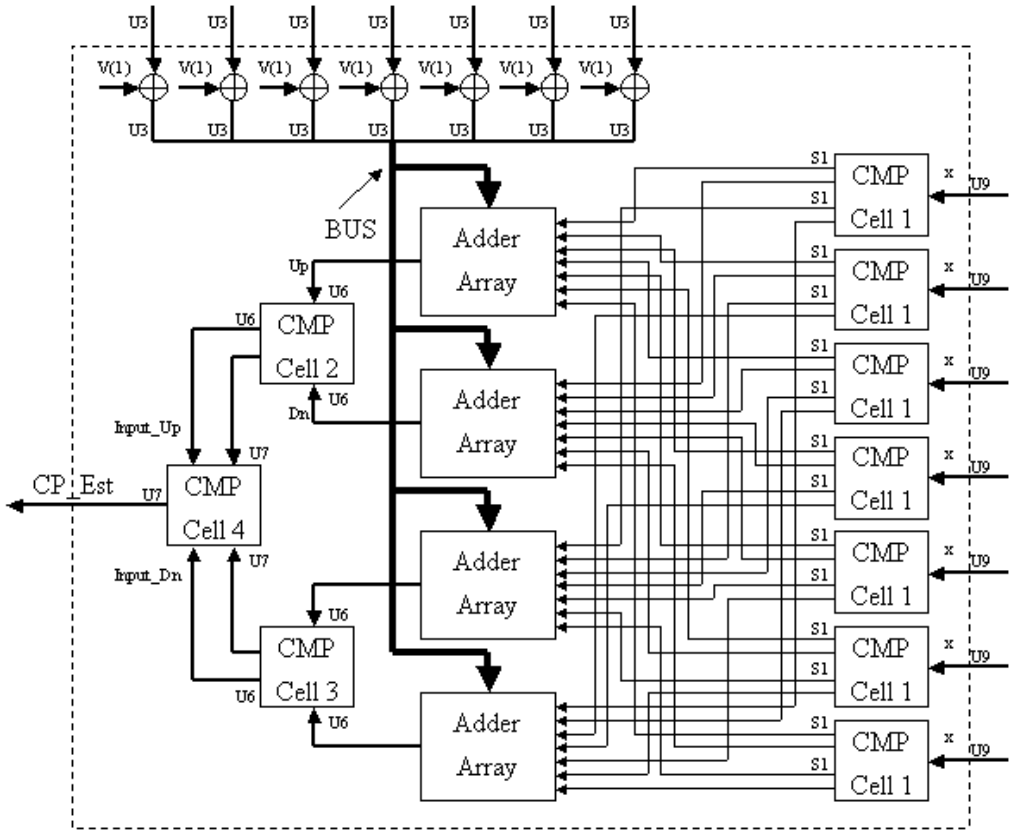
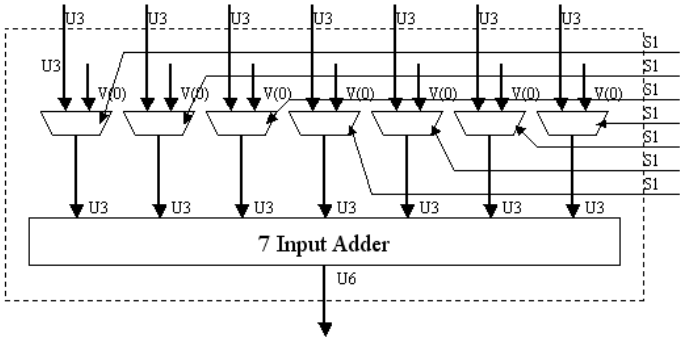


圖 3.2.2.3 估計循環字首長度的組合電路

上圖(3.2.2.3)即是圖(3.2.2.1)中，最後一部分元件的區塊展開圖，此組合電路會接收每個峰值所對應到的可能循環字首長度，以及該峰值所對應到、經過加權後的匹配濾波器輸出，利用簡單的加法陣列和比較器，估計出循環字首長度。



CMP Cell 1	0<x<12	12<x<24	24<x<48	48<x<80	otherwise
Output1	1	0	0	0	0
Output2	0	1	0	0	0
Output3	0	0	1	0	0
Output4	0	0	0	1	0

CMP Cell 2	Up>Dn	Up<Dn
Output1	8	16
Output2	Up	Dn

CMP Cell 3	Up>Dn	Up<Dn
Output1	32	64
Output2	Up	Dn

CMP Cell 4	Up>Dn	Up<Dn
Output	Input_Up	Input_Dn

圖 3.2.2.4 加法陣列電路以及比較器輸出判斷示意圖

3.2.2.1 輸入輸出以及控制訊號

資料輸入訊號：

- Rx_In：經過頻率偏移補償的資料取樣點，有實部、虛部訊號之分

資料輸出訊號：

- CP_Est：所估計出來的循環字首長度

3.2.2.2 乘法替代組合電路

就如同短調整符元同步的硬體一樣，在估計循環字首的硬體當中，我們亦採用了匹配濾波器的架構；而同樣的，為了降低硬體的負擔，所以我們使用乘法替代組合電路來減少濾波器中，使用六十四個乘法器所帶來的成本，下表(3.2.2.2)即是該六十四點匹配濾波器係數，與其相對應的組合電路型態關係。

#th Coef.	Type	#th Coef.	Type	#th Coef.	Type	#th Coef.	Type
1	II	17	IV	33	II	49	I
2	II	18	III	34	II	50	IV
3	IV	19	III	35	III	51	IV
4	III	20	II	36	III	52	I
5	II	21	II	37	IV	53	I
6	II	22	IV	38	I	54	III
7	III	23	II	39	IV	55	I
8	II	24	IV	40	III	56	II
9	II	25	IV	41	I	57	II
10	III	26	II	42	I	58	III
11	III	27	IV	43	III	59	IV
12	III	28	III	44	II	60	II
13	III	29	I	45	I	61	IV

14	IV	30	III	46	I	62	II
15	II	31	II	47	II	63	I
16	II	32	I	48	I	64	IV

表 3.2.2.2 乘法替代組合電路與長調整符元在時序上的關係

3.2.2.3 硬體運作的機制

如同之前所述，圖(3.2.2.1)的估計循環字首長度硬體，和本章一開始的短調整符元同步硬體，是非常相近的，所在的差別只是在於如何對求得峰值的位置、以及該峰值作正確的解讀以及分析，以獲得正確的結果。

在估計循環字首長度硬體的一開始，輸入的資料取樣點會先經過匹配濾波器，以求得包括在此次收到的取樣點內、前六十四個資料點與匹配濾波器係數間的迴旋積值；經過絕對值平方的運算元後，可以獲得迴旋積值的大小；然後將此得到的結果與暫存器內儲存的七個最大峰值作排序的動作，利用比較器和暫存器的組合，將此八個輸入作由到大小的排列，同時將經過排序後的前七個最大值，更新於之前七個暫存器中，作為下一次比較所需要的前七個最大峰值；再排序的同時，亦會傳送控制訊號給後端的記數器，來控制記數器內的值為原本的值減一、為上一個記數器內的值減一、或者是一個初始的常數值；這些控制的依據在於此次資料取樣點所造成的匹配濾波器輸出結果的大小，當此值的大小大於七個峰值暫存器內所存放的值的時候，該值經過排序後所位在次序的相對應記數器，將會被初始為一常數值；而峰值暫存器內存放的值比該值大的次序將不會改變，因此其相對應的記數器將會減一，用以表示遠離該峰值所對應到的長調整符元起始位置一個取樣點；但若峰值暫存器內存放的值比該值小的時候，該峰值的次序將會向後減一，因此其峰值對應到記數器內所存放的值，應更新為次序位在該峰值前一位所對應到的記數器內所存放的值減一，如此才可以滿足七個峰值暫存器會與七個記數器相互對應的關係。

舉個例子來說，當現在的資料取樣點所求得的匹配濾波器輸出大小，比七個峰值暫存器內的第四個值來的大，則比較器和暫存器所組合而成的排序器，會

將此七個峰值暫存器內所儲存的值更新，更新後的結果為：原本的第一、第二、第三個暫存器值不變，而第四個暫存器值會更新為此次資料取樣點所求得的峰值，而第五、第六、第七個暫存器將會由原先第四、第五、第六個暫存器內存放的值依序更新；同時，為了要符合暫存器與記數器能相互對應，所以會傳送控制訊號來控制記數器內的值的更新；在上面的例子中，由於第一、第二、第三個暫存器值的次序不變，所以該對應的第一、第二、第三個記數器值更新為原先的值減一；而第四個暫存器因更新為該次匹配濾波器輸出的峰值，所以其相對應的記數器便會更新為一個初始的常數，表示有一個可能的長調整符元起始位置被找到。而第五、第六、第七個記數器內所儲存的值，由於原本其相對應的第五、第六、第七個暫存器，已經由第四、第五、第六個暫存器內存放的值依序更新，所以在更新第五、第六、第七個記數器內所儲存的值的時候，應該要選取第四、第五、第六個記數器內的值減一的結果，如此才能符合暫存器與記數器相對應的機制。

最後我們來看在記數器後方的組合電路；在圖(3.2.2.1)中我們可以知道：該組合電路的輸入訊號可以分為兩個部分，一部分是七個峰值經過加權過後的結果，另一部分是這七個峰值所對應到的可能循環字首長度；接著我們來看圖(3.2.2.4)的內部展開圖：在圖中，CMP Cell 1 會先針對七個可能循環字首長度的輸入訊號作分類，看其值落於{0,12}、{12,24}、{24,48}、{48,80}或是其他的哪個區間中，用以來決定四種可能循環字首長度的比重；今天若輸入的可能循環字首長度為四十三，由於其值落於{24,48}的區間中，因此我們就將可能為三十二的循環字首長度的比重，增加{1+該對應的峰值經過加權過後的結果}；最後統計這四種可能循環字首長度的比重，看何者的比重最大，即表示該可能的循環字首長度最為可能。

3.2.2.4 字元長度的選擇

誠如之前所述，基本上估計循環字首長度的硬體設計，與短調整符元同的的硬體設計，是極為相似的；一開始皆是一個匹配濾波器的架構，然後接著為比較器、暫存器、記數器；除了匹配濾波器所採用的係數不同，以及使用的暫存器、

記數器個數也不同外，基本上沒有太大的差距；而上面所述的不同也不會影響到字元長度的選擇，所以這部分字元長度的選擇請參考本章 3.1.2.4 節。