
第六章 FPGA 的設計與模擬

在第二、三、四章中，我們已經介紹過各個硬體單元的演算法、架構和運行機制；接著我們在第五章中，介紹了各個硬體間的配合，以及整個系統硬體的時序圖；因此本章的目的就在於介紹如何將上述的理論，在我們的 FPGA 板上實現，並且提供其模擬的結果，用以和之前幾章的介紹作比對之用。

在本論文所採用的 FPGA 板為 Altera 公司、Stratix 系列中，型號為 EP1S25F780 的可程式化邏輯陣列；其中含有 25,660 個邏輯單元和 1,944,576 大小的記憶體以及 597 個腳位，這些都可以滿足我們整合後硬體的 demand。

6.1 封包偵測 Packet Detection

下圖(6.1.1)中所示的，是封包偵測硬體的模擬結果圖；由於為了簡化比對結果正確與否的關係，所以在圖中的輸入取樣點資料，我們輸入資料是以六十四點取樣點為週期的、一到六十四遞增數列的實數訊號，也就是輸入訊號的實部將會從一開始累進，直到六十四後再回到一，如此週而復始下去；而在虛部方面的話，則一直維持著零電壓的狀況，如此以利我們在比對資料時的方便。

在下圖中的欄位 Rx_Re 和 Rx_Im 便是代表著輸入的訊號；由於必須要模擬滑動框架的架構，因此不同於第二章的圖(2.1.3)中的架構，除了原本的架構外，我們還把記憶體的運作，至於此模擬當中。在第二章中我們提過：滑動框架的實踐，需要使用三塊記憶體，而這三塊記憶體將會由下圖中的 Mem1_we 和 Mem23_we 訊號線，來控制這三塊記憶體的寫入與讀取；而 Mem1A 和 Mem23 則控制著這三塊記憶體，寫入或是讀取的位置。

由於整個模擬的時間很長，我們無法在此全部的展現出來，所以我們只擷選最精華的片段，也就是 S_cond 訊號由低電位轉為高電位的時候。在下圖(6.1.1)中，我們可以看到當第 $64+57=121$ 個資料輸入時，由於此時資料的輸入，再經

過中間硬體的運作，使得此時的 $|Cn|^2$ 大於 $|Pn|^2 \cdot Threshold$ ，也就是訊號線 ABSout1

大於 ABSout2；因此 S_cond 的訊號將會由低電位轉為高電位，如同下圖所示。

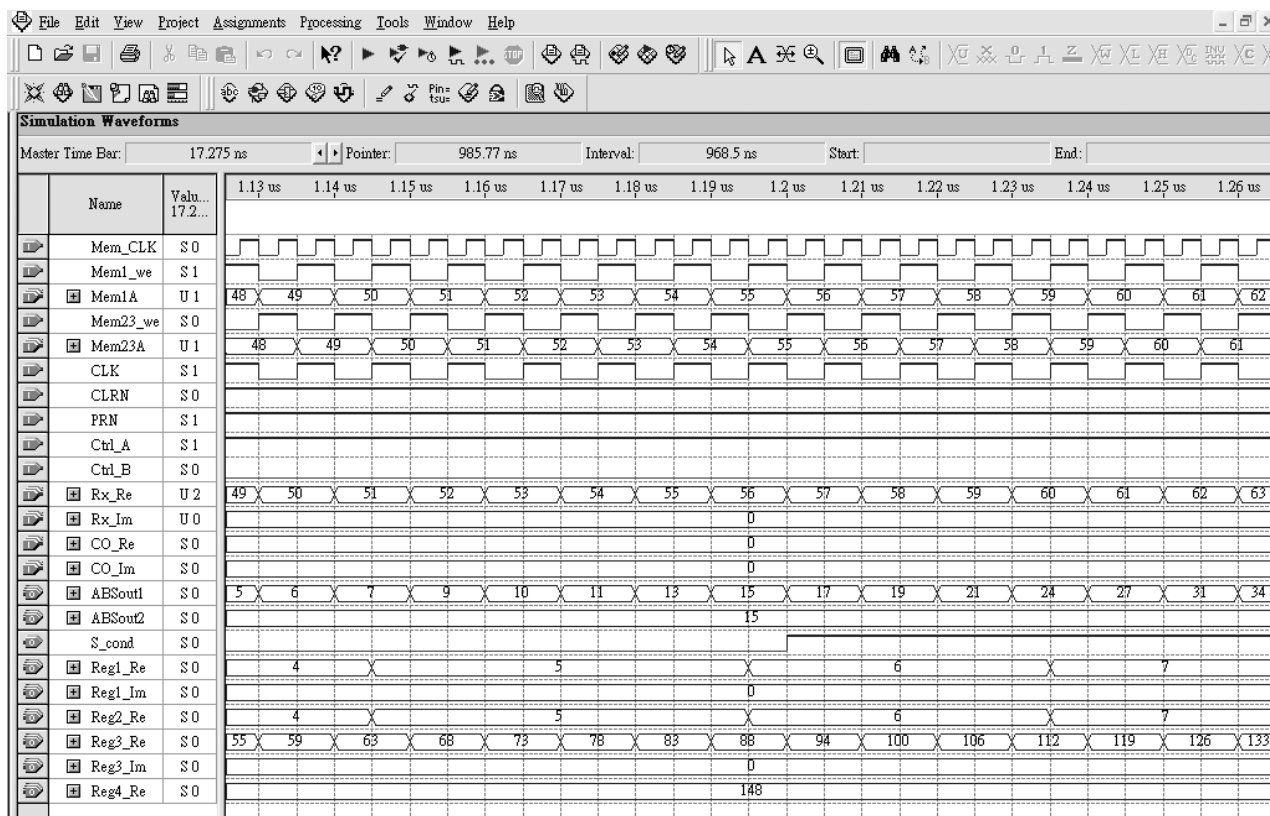


圖 6.1.1 封包偵測硬體的模擬結果圖

6.2 數位座標旋轉 CORDIC

接著我們來看數位旋轉座標演算法作求主幅角度用的硬體架構所得到的模擬結果。下圖(6.2.1)就是該硬體架構作模擬後所得到的結果，輸入的資料分別是隨機產生的十六個字元所能表示出來帶有正負號的整數。由於本身在設計此硬體時，便是在系統時脈頻率為資料時脈頻率兩倍的前提之下，因此，理論上系統時脈取樣到資料後，將需要經過包括本次系統時脈在內的兩個系統時脈後，才可以求得正確的結果輸出，但在實際情形，由於硬體本身設計的關係，所以還需要在延遲一個系統時脈的偏移(Offset)；因此，在實際的情形下，當系統時脈取樣到資料後，將會在包括此系統時脈後的第三個系統時脈，得到該輸入取樣點所對

應到的主幅角度，即如下圖所示。

在下圖(6.2.1)中，OneBC 的控制訊號線是用來控制此時本硬體該擷取新的取樣點，用以求得新取樣點的主幅角度，亦或是選取回饋的資料，將整個運行的機制完成；此外，此 OneBC 的控制訊號亦可以控制此時選取到的階段角度(Step Angle)。所以此 OneBC 的控制訊號其實就是相同於圖(2.2.3.1)中的一位元加法器，相關的硬體設計與演算法，可以參照第二章的 2.2.2 節。

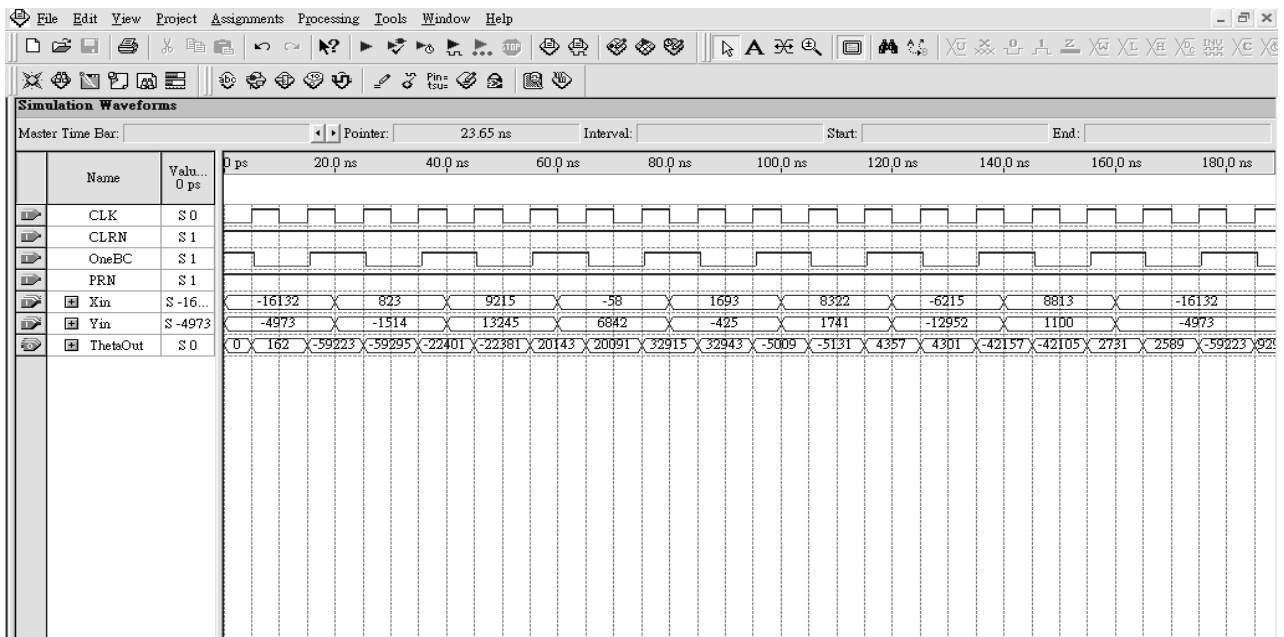


圖 6.2.1 數位旋轉座標演算法作求主幅角度用的模擬圖

下表(6.2.1)則是將圖中輸入的資料，去計算其實際相對應的主幅角度，接著對求得的結果作轉換的動作，用以與利用數位座標旋轉演算法所設計出來的硬體的輸出，作相互的比較，證明本硬體的運作正確。

實部輸入	虛部輸入	實際值(徑度)	實際值轉換	硬體結果
-16132	-4973	-2.8426	-59298	-59295
823	-1514	-1.0726	-22381	-22381
9215	13245	0.9629	20088	20091
-58	6842	1.5793	32945	32943

1693	-425	-0.2460	-5130	-5131
8322	1741	0.2057	4292	4301
-6215	-12952	-2.0182	-42101	-42105
8813	1100	0.1242	2590	2589

表 6.2.1 主幅角硬體輸出與理論值的比較

接著我們來看數位旋轉座標作旋轉角度用之硬體的模擬結果。下圖(6.2.2)就是其模擬的結果圖，其輸入的資料是隨機產生的十個字元所能表示出來帶有正負號的整數。相同於數位座標演算法作求主幅角一般，控制訊號 OneBC 代表著是圖(2.2.4.2)中，一位元加法器的輸出值，可以用來控制整個硬體的運作是接收新的取樣點，用以作旋轉角度之用，還是選取回饋的資料，用以將未完成的硬體運作機制完成；同時該控制訊號亦可以用於選取正確的階段角度，如此才可以得到正確的結果。不同於求取主幅角的硬體運作，旋轉硬體的時脈延遲雖然同樣只有兩個系統時脈，但是結果的輸出並沒有任何的時脈偏移，所以當此次系統時脈取樣到輸入資料後，我們將可以在包括此次時脈訊號在內的兩個時脈訊號後，得到正確結果的輸出，即如下圖所示。

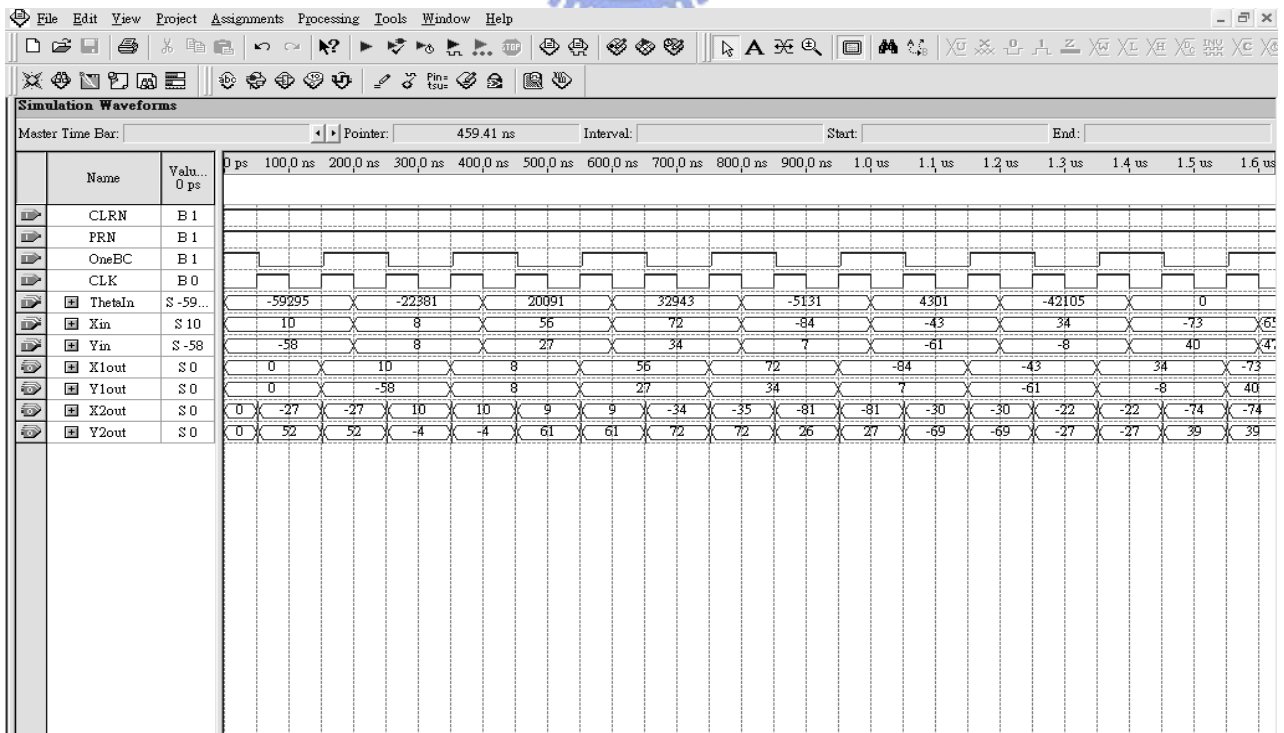


圖 6.2.2 數位旋轉座標演算法作旋轉角度用的模擬圖

同樣的，為了驗證模擬結果的正確性，所以我們將硬體模擬的結果與實際值作比較，得到了下表(6.2.2)的結果。

輸入實部	輸入虛部	旋轉角度	理論實部	理論虛部	輸出實部	輸出虛部
10	-58	-59295	-26.65	52.47	-27	52
8	8	-22381	10.84	-3.20	10	-4
56	27	20091	9.80	61.39	9	61
72	34	32943	-34.60	71.71	-35	72
-84	7	-5131	-79.76	27.24	-81	27
-43	-61	4301	-29.60	-68.51	-30	-69
34	-8	-42105	-21.92	-27.18	-22	-27
-73	40	0	-73	40	-74	39

表 6.2.2 旋轉角度硬體輸出與理論值的比較

6.3 短調整符元匹配濾波器 Short Match Filter

接著來看短調整符元匹配濾波器架構的模擬圖。下圖(6.3.1)即是短調整符元濾波器架構的模擬圖，在圖中我們輸入的資料，是將輸入取樣點經過取符號字元後的結果，因此，在圖中的輸入訊號 R_{in} 和 I_{in} 都是以一個位元來表示，當此一位元輸入訊號為低電壓準位時，表示當初的輸入訊號為正；相反的，當為高電壓準位時，即表示當初輸入的訊號為負。

在此硬體的模擬當中，我們採用的輸入訊號是原始不加入任何雜訊，而且也沒有任何通道效應的短調整符元，如此的目的便是為了萬一發生錯誤，我們也易於除錯，同時也可以測試演算法的可行與否。由於短調整符元內有四個循環週期區間，因此模擬的結果，我們將可以看到四次由六十三倒數到零的結果，但由於篇幅的關係，所以圖中只顯示：當短調整符元找到一個新峰值時，將此硬體內部的累加器重設(Reset)，而且從下一個資料時脈開始，累加器又接著開始倒數的情形。

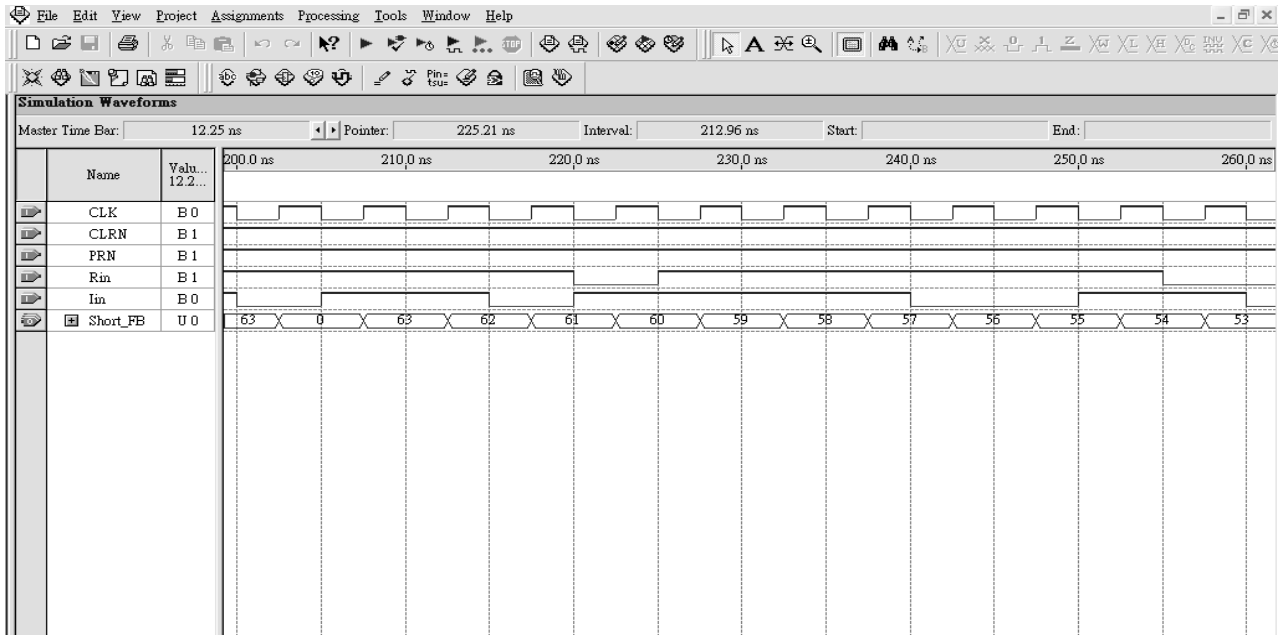


圖 6.3.1 短調整符元匹配濾波器的模擬結果

6.4 系統的整合

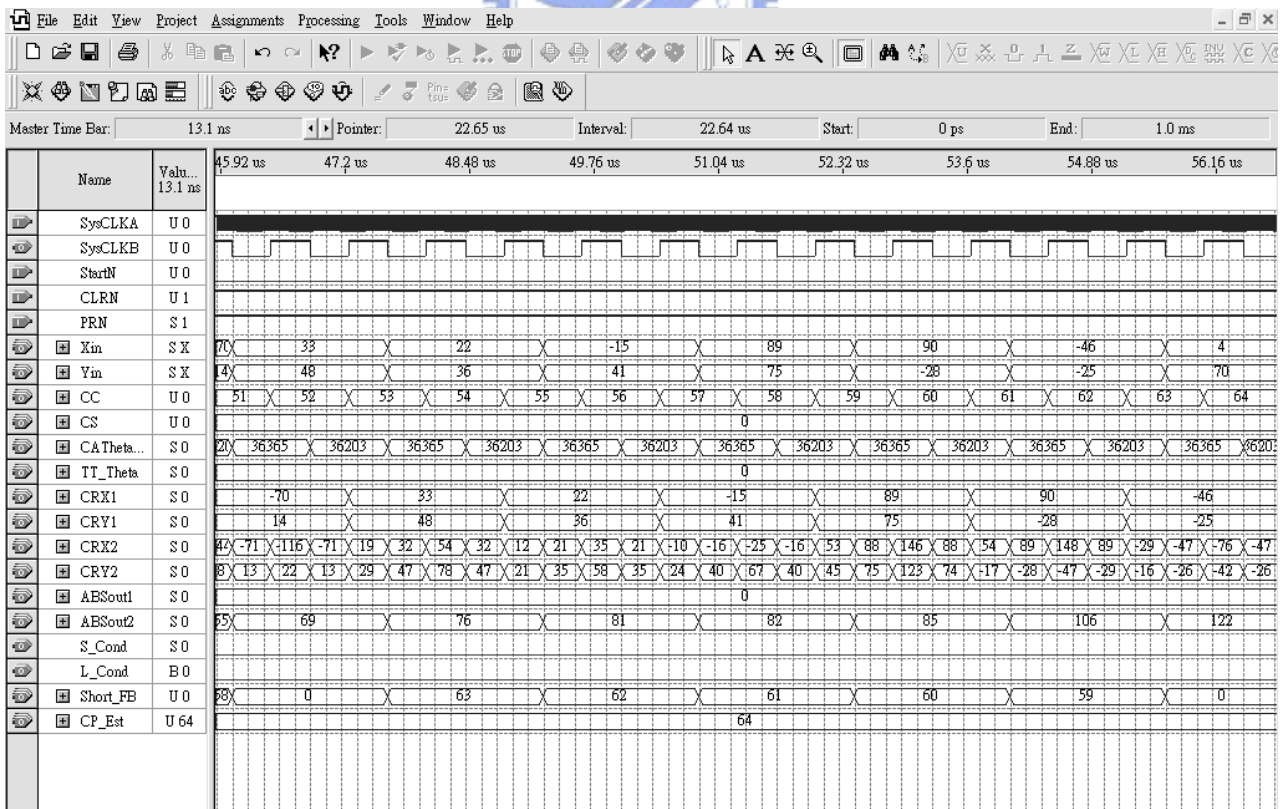


圖 6.4.1 系統整合的模擬結果 (一)

上圖(6.4.1)中所顯示的是當系統開始啟動時，而尚未偵測到有封包存在的情形；在圖中的訊號線 SysCLKA 是採用 FPGA 板內建之 80MHz 的時脈訊號，而 SysCLKB 才是我們硬體系統真正使用的時脈訊號；Xin 和 Yin 訊號分別表示著輸入取樣點的實部和虛部，而 CC 和 CS 分別表示著控制單元(Control Block)的計數器和目前此控制單元所運行的狀態(State)；CA_Theta 則表示著此時利用數位座標旋轉演算法所求得的主幅角度，配合著累加器便成為 TT_Theta 訊號線，表示著此時輸入資料點所補償的頻率偏移角度；而 CRX1、CRY1、CRX2 和 CRY2 則是 CORDIC A 硬體單元的輸出；其餘的訊號線則如同之前所述，我們就不再介紹。

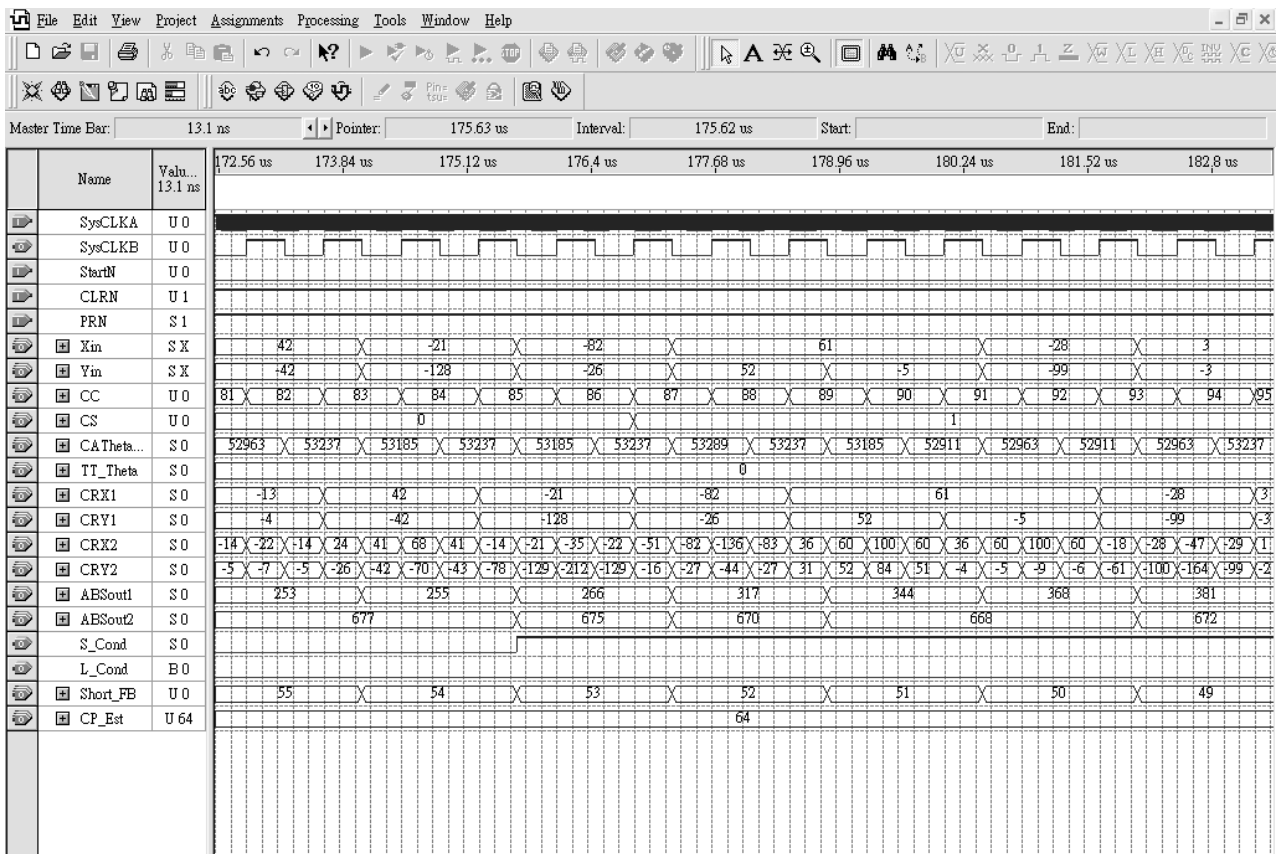


圖 6.4.2 系統整合的模擬結果 (二)

接著我們來看上圖(6.4.2)，在上圖中，顯示的是當 S_Cond 訊號由低電位轉高電位的情形，表示著整個系統硬體偵測到一個封包的存在，所以此時的控制狀態(CS)訊號將會由 0 轉 1，做下一個硬體運作的機制。

下圖(6.4.3)是當 S_Cond 由低電位轉高電位，再經過六十四個取樣點後的情形，此時的目的是為了確保為求得頻率偏移所需要的 Cn 值正確；因此當控制狀態由 1 轉 2 時，便代表著該時候的 Cn 值可以求得粗調頻率偏移，而接下來的控制狀態 2、3 和 4，其存在的目的便是為了估出頻率偏移。

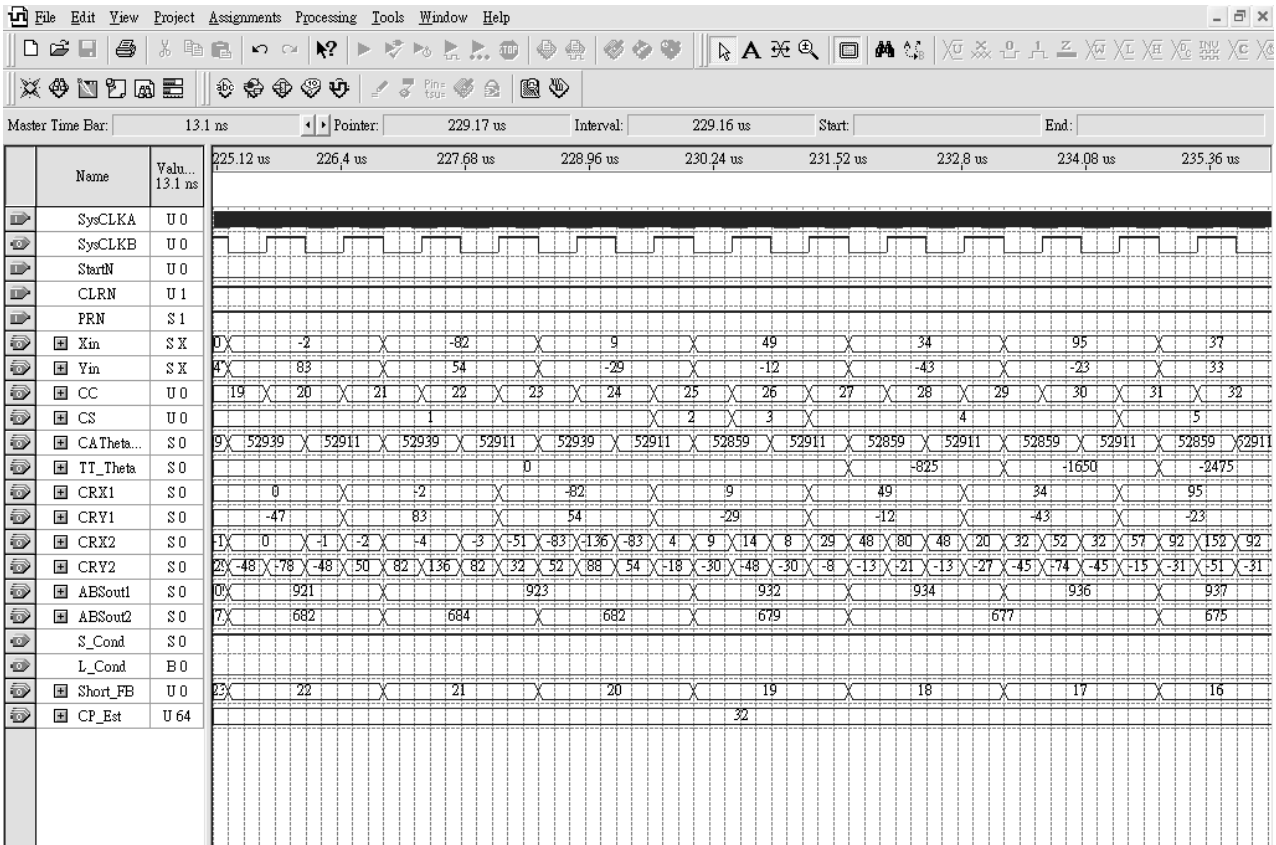


圖 6.4.3 系統整合的模擬結果 (三)

在求得頻率偏移的同時，硬體 Short Match Filter 亦可以將距離下一個調整符元邊界的取樣點個數求出，而控制狀態 7 便是為了這個目的而存在的；當控制單元得知距離下一個調整符元邊界後，控制單元的計數器便會開始倒數，直到倒數該值為零時，便表示到了下一個調整符元的開頭，此時控制單元便會轉換到下一個狀態，即如下圖(6.4.4)中所示的控制狀態 7 轉控制狀態 8 的情況。

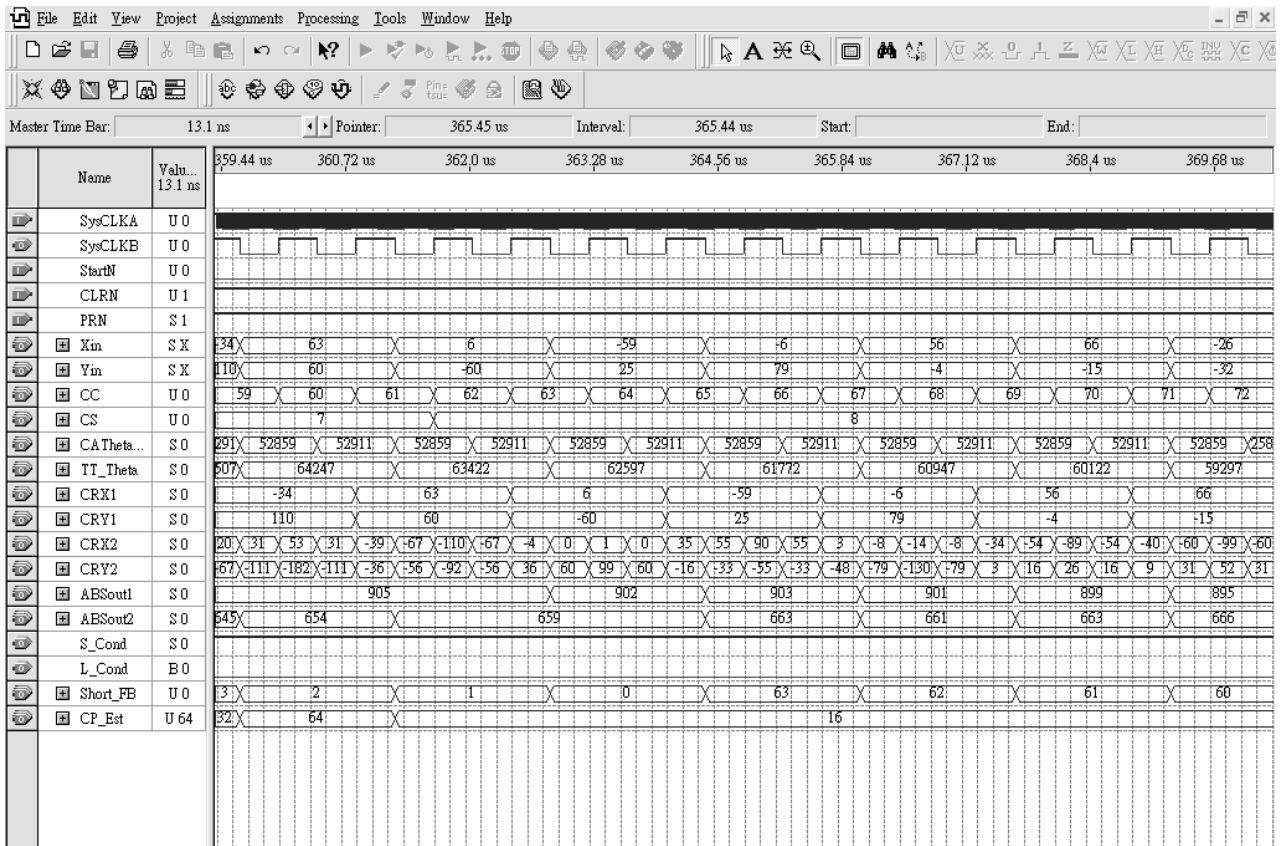


圖 6.4.4 系統整合的模擬結果 (四)

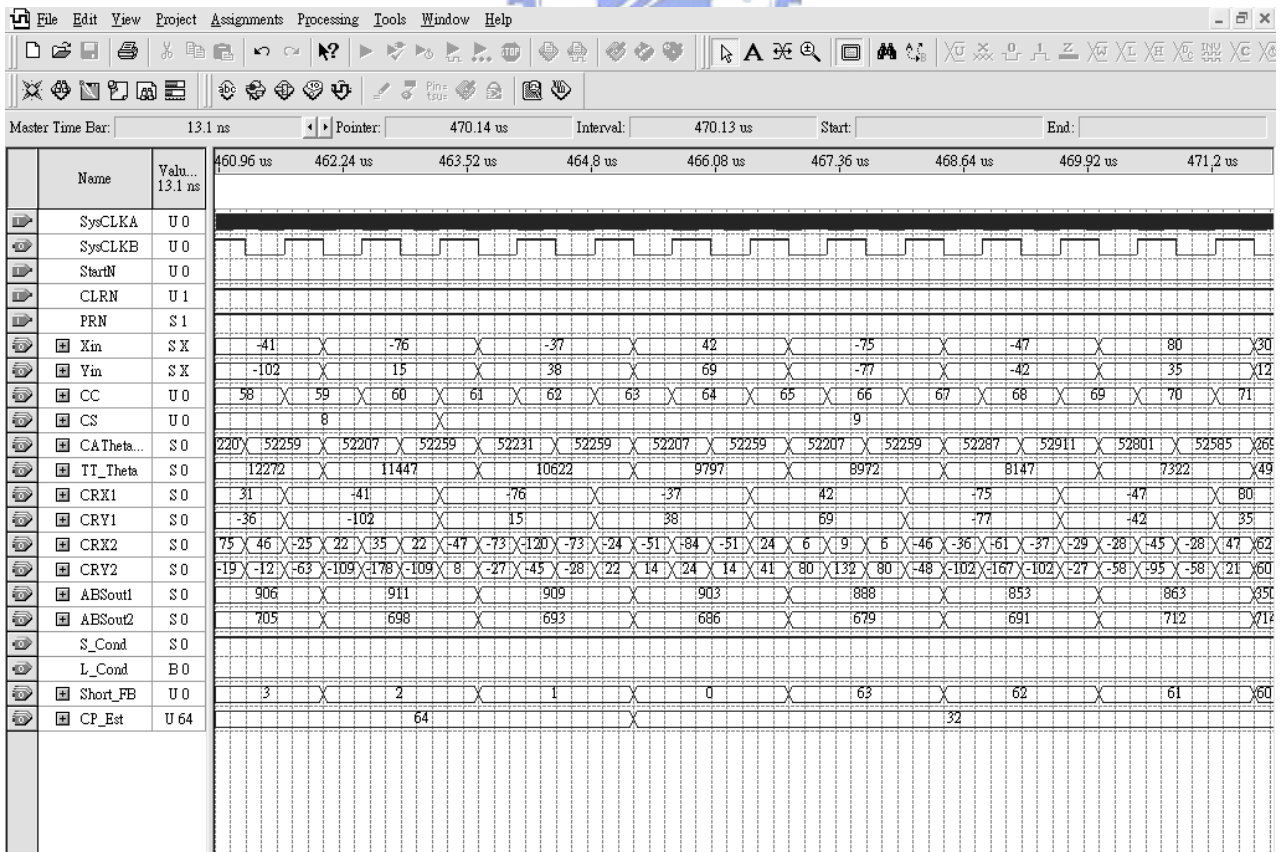


圖 6.4.5 系統整合的模擬結果 (五)

當控制狀態由 7 轉 8 的時候，表示我們已經對齊好調整符元的開頭，接下來，便是每隔六十四個取樣點便查看 L_Cond 訊號是否由低電位轉為高電位，若是，表示適才所接收的六十四個取樣點為長調整符元的前六十四個取樣點；若否，則表示該六十四個取樣點為短調整符元的其中一個循環區間，而硬體便需要檢測下一個六十四點時的 L_Cond 訊號，以便接下來循環字首估計問題的解決。上圖(6.4.5)便是展現這樣的結果，在控制狀態 8 結束的同時，表示著已經接收過六十四個取樣點，此時查探 L_Cond 訊號，發現其電壓準位尚為低電壓準位，因此控制狀態 8 轉為控制狀態 9，等待下一次接收完六十四點取樣點。

下圖(6.4.6)則是當控制狀態 9 維持了六十四個取樣點後，L_Cond 訊號線為高電壓準位的情形，由於 L_Cond 為高電壓準位，所以表示再此之前的六十四個取樣點為長調整符元的前六十四點，因此控制狀態轉為 11，準備作循環字首的估計。

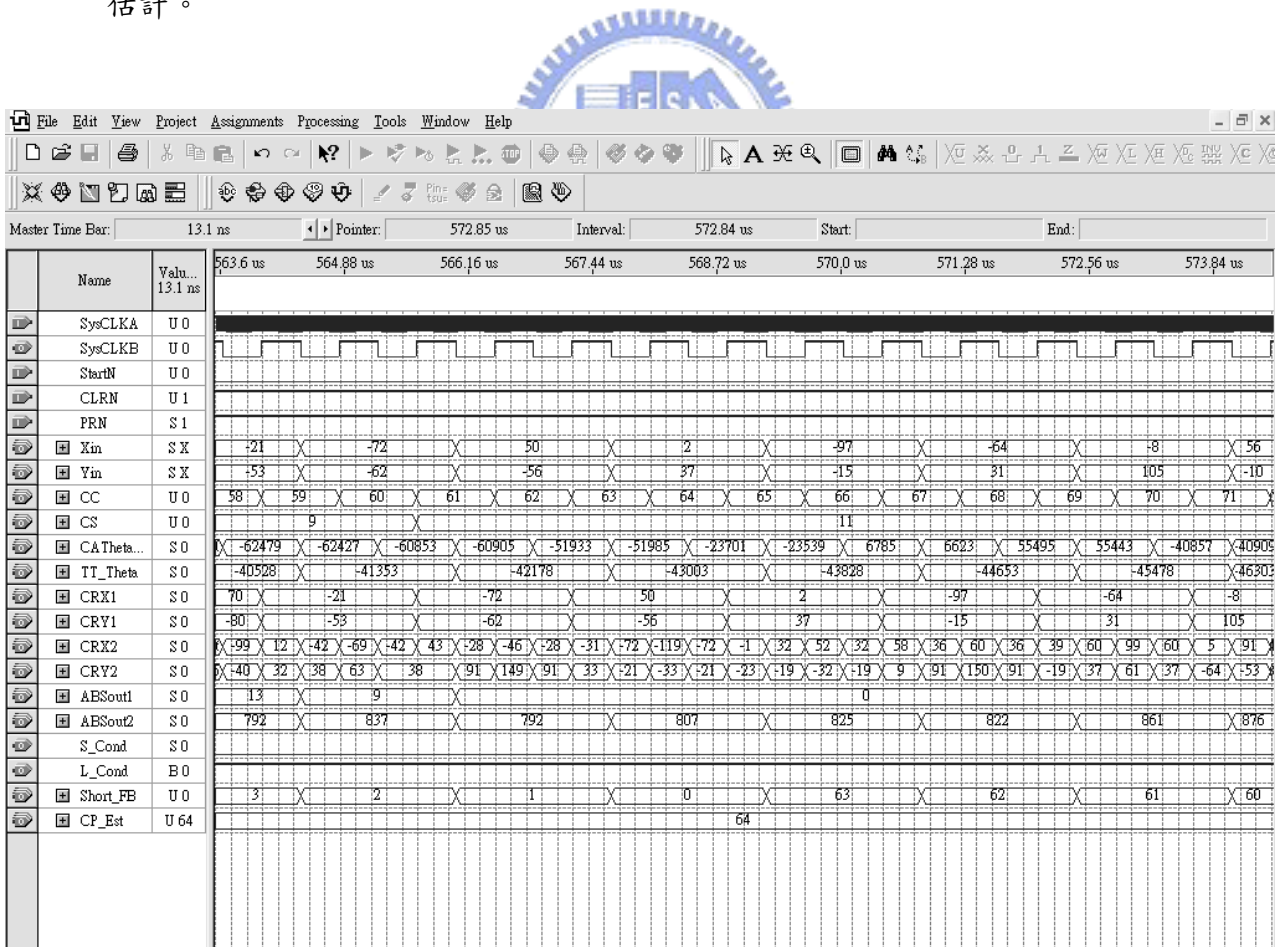


圖 6.4.6 系統整合的模擬結果 (六)

下圖(6.4.7)中，控制狀態 15 的開始位置是在當控制狀態由 9 轉為 11 後的第一百二十八個取樣點，目的是在於配合 Long Match Filter 硬體單元的運作；當控制狀態轉為 15 的同時，控制單元會儲存此時 Long Match Filter 硬體單元所估計出來的循環字首長度；如圖中所示：由 Long Match Filter 硬體單元所估計出來的循環字首長度為三十二個取樣點，與實際狀況符合。

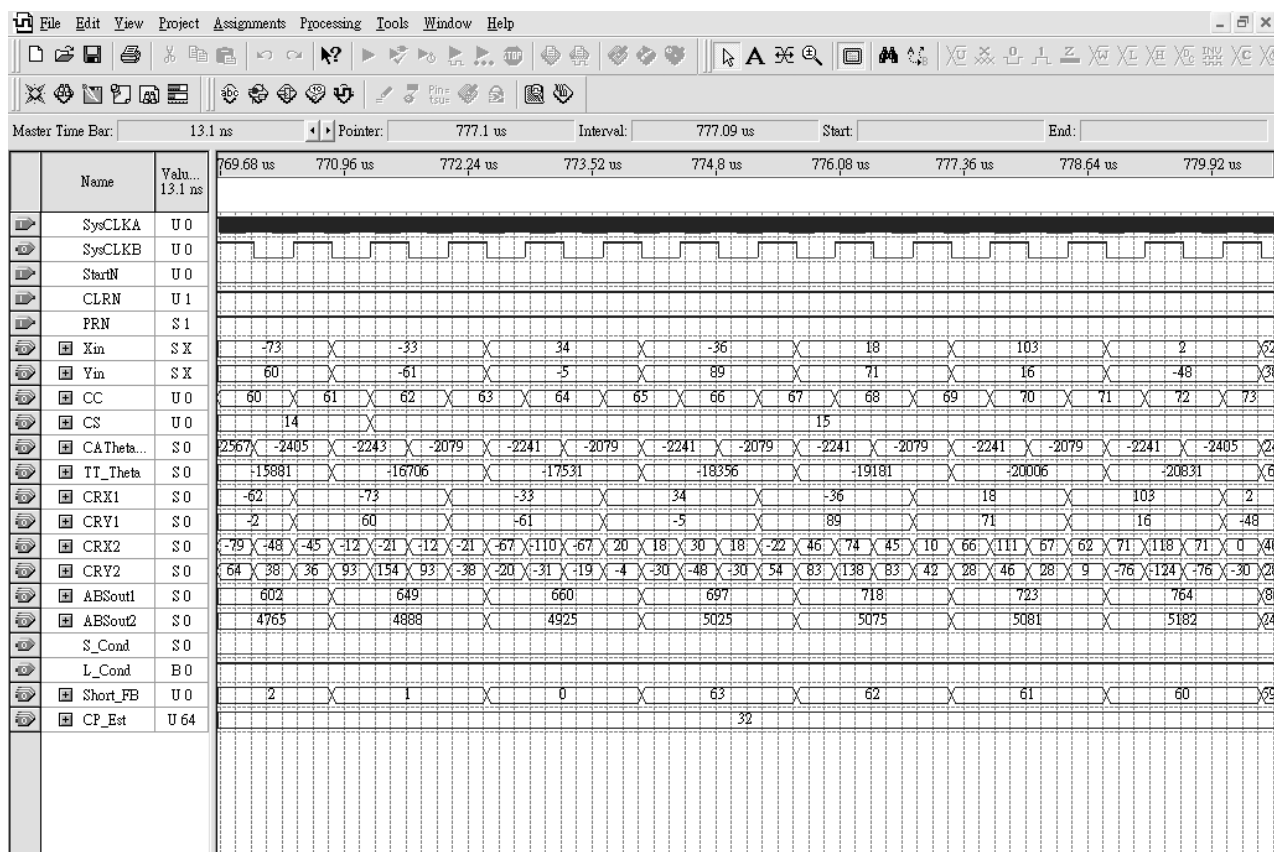


圖 6.4.7 系統整合的模擬結果 (七)

在求得循環字首長度後，我們便可以輕易的找出長調整符元的結束位置，進而達到封包取樣點同步的目的，同時也完成我們系統整合的模擬。