**[1] 設計者姓名及連絡電話**

**學生姓名：潘世軒、黃元培、吳隆暉**

**連絡電話：0978186136**

**[2] 專題名稱**

中文專題名稱： 32點快速傅立葉轉換處理器

# 英文專題名稱：32-point Fast Fourier Transform Processor

**[3] 全新設計或改版說明**

此案件為設計者全新設計

**[4] 原理及架構說明**

離散傅立葉變換(DFT)是一個在訊號處理的領域中相當常用的工具，他可以幫助我們從頻域的角度去分析或設計一個系統，而我們研究的主題：快速傅立葉變換(FFT)，是其中一類計算離散傅立葉變換的快速演算法，相較於直接計算DFT，FFT有效減少運算量，降低了計算複雜度。而本設計的研究目的，即是設計出能夠計算32點快速傅立葉變換的晶片。

我們選擇實作FFT的方法，是Radix-II DIF FFT 演算法，此演算法利用離散傅立葉變換在複數平面上的對稱性質，將具有對稱性質的多個乘法合併，以大幅減少數學運算量，在相同的數學模型架構下，較有效率的算出結果。此演算法可大致分為兩種不同架構，時間點分組架構(DIT, Decimation in time)以及頻率點分組架構(DIF, Decimation in frequency)，本設計選擇DIF架構。以下分別介紹兩種分組架構。

DIT架構之Radix-II FFT數學推導如下：

N點離散傅立葉變換(DFT)的數學公式（假設N為偶數）為

將奇數點與偶數點分開運算，可得

令偶數點的，奇數點的，則

其中，

化簡後G[k]和H[k]又可視為對奇數點與偶數點分別做N/2點的離散傅立葉轉換，再遞迴地對兩式繼續操作下去，即可計算完畢。

考慮8個點（N=8）的FFT，其數據流程圖為

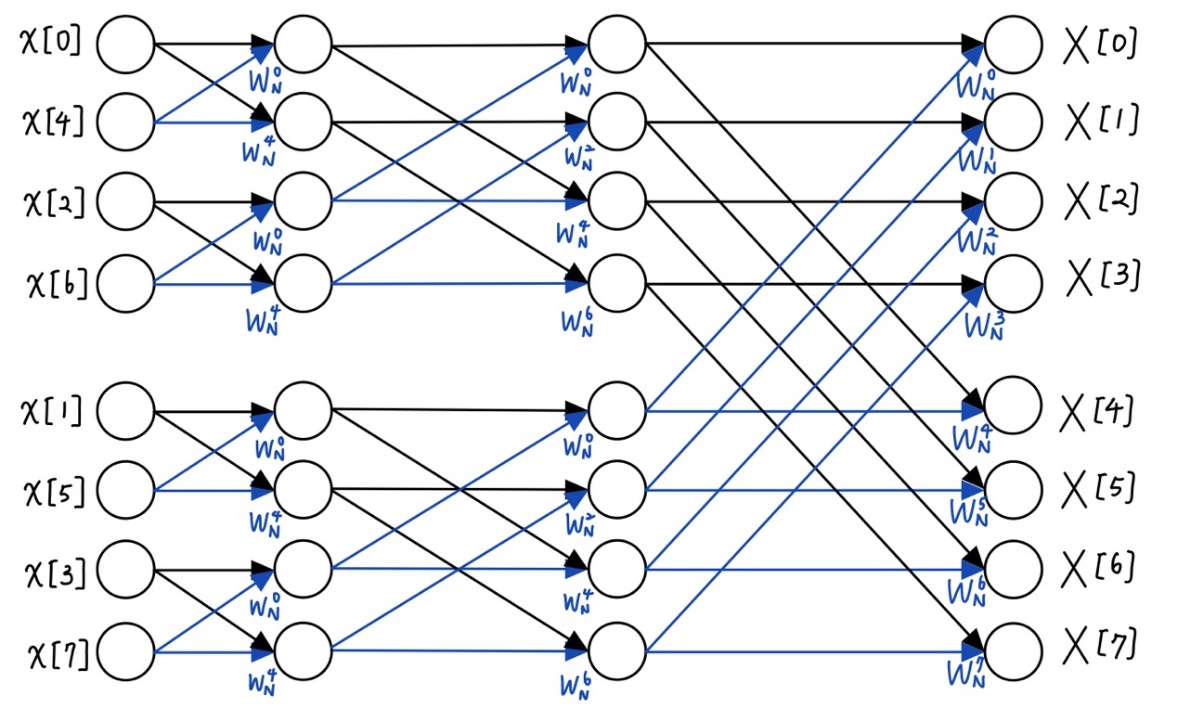


Fig.(1) N=8 Radix-II DIT FFT 數據流程圖

可觀察到上圖中有許多形似蝴蝶的運算架構，在此架構中被稱為蝴蝶圖（DIT butterfly），觀察每一個蝴蝶圖，可以發現運算過程中乘上的兩個twiddle factor均有相位差180度，亦即正負號相反，因此我們可以將蝴蝶圖的結構化簡如下圖

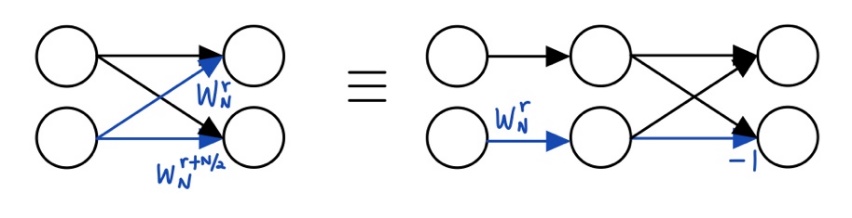


Fig.(2) DIT蝴蝶圖化簡

並將數據流程圖化簡為

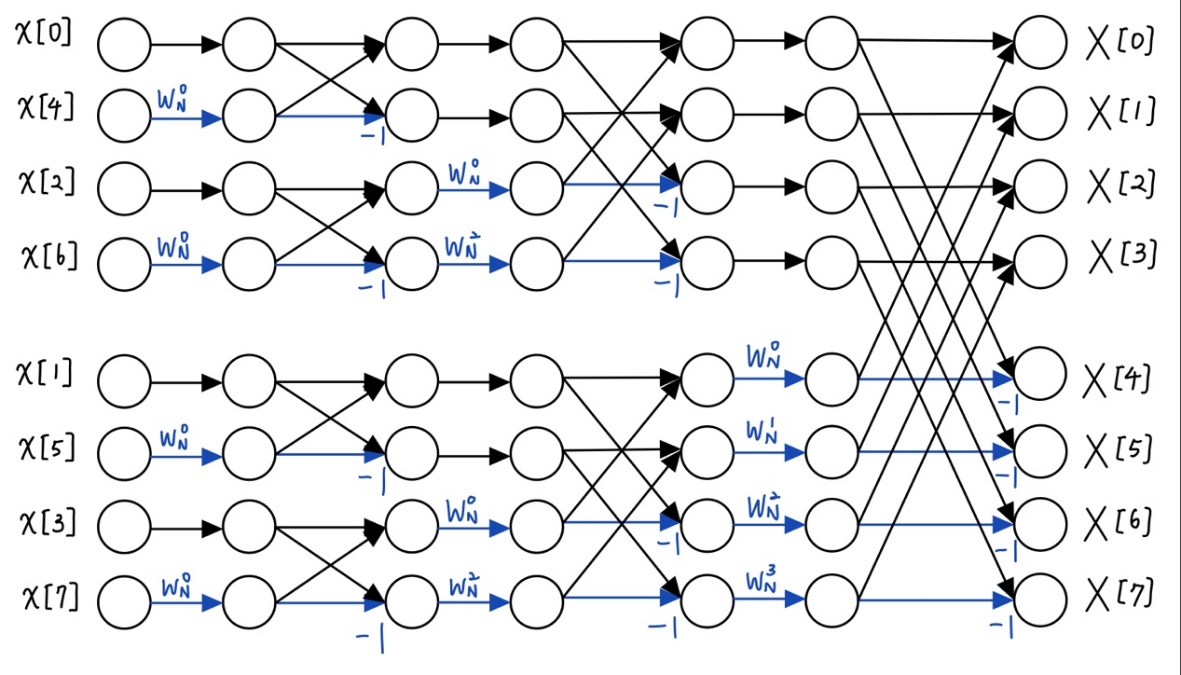


Fig.(3) 化簡過後的DIT數據流程圖

DIF 架構以頻率點切割，數學推導如下：

N點離散傅立葉變換(DFT)的數學公式（假設N為偶數）為

令偶數點的，其中r = 0, 1, 2, …, (N/2)-1，則

此即將上半平面與下半平面的資料相加後，做N/2點的離散傅立葉變換。

令，奇數點的，其中r = 0, 1, 2, …, (N/2)-1，則

此即將上半平面與下半平面的資料相減，乘上twiddle factor後，做N/2點的離散傅立葉變換。

考慮8個點（N=8）的FFT，將分組後的訊號遞迴地往下分組運算，蝴蝶圖的化簡方式與化簡過後數據流程圖為

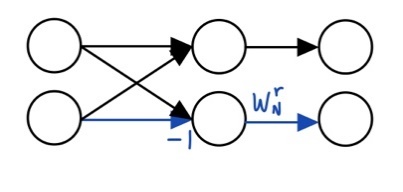


Fig.(4) DIF蝴蝶圖化簡

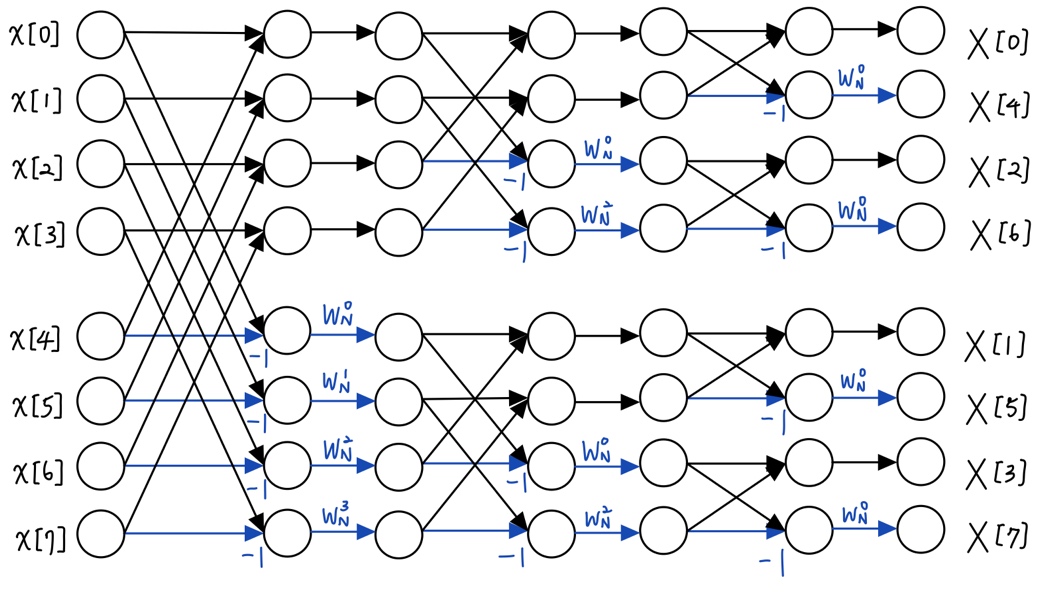


Fig.(5) 化簡後的DIF數據流程圖

此架構即為我們最終採用的演算法架構。[2]

考慮原本的離散傅立葉變換，假設輸入為N個點的訊號，則直接用公式計算約需要使用N次複數乘法以及N-1次的複數加法，每次複數乘法需要4次實數乘法以及2次實數加法，每次複數加法需要2次實數加法，因此合併來看N個輸出點共需要次實數乘法以及次實數加法，運算次數複雜度為。而Radix-II DIT FFT演算法將計算切為個stage，每個stage包含N次複數加法、N/2次複數乘法以及N/2次變號，總共需要次實數乘法、次實數加法以及次變號，運算次數複雜度為，相較於原本的計算量降低了許多。

在採用DIF架構的Fast Fourier Transform時，我們也要考慮如何表示小數，考量到floating point在做加法或者乘法的時候的運算單元複雜度高，以及面積非常大，因此我們選擇採用二補數的fixed-point表示法，此表示法的基本使用方法如下：

舉例來說，考慮一binary的數字x：001100 01010，其整數部分占六位、小數部分占五位，這樣的話，x在我們二補數fixed point 表示法中的值就是值394除以32，因此在十進位的表示法為12.3125。其中394為00110001010的值，而因為小數點占五位，因此實際上他表示的值為394除以32。

透過此種fixed-point的表示法有幾項好處，首先就是在做這些數字相加的時候，我們只要透過signed extension把兩個數字的小數點對齊之後就可以直接透過加法器將兩數相加，乘法亦然，我們可以使用乘法器直接將兩數相乘。再來就是fixed-point表示法對我們來說比較容易理解，我們便可以將心思放在電路的設計而非那些運算單元的設計。因此最後我們採用fixed-point表示法來表示運算過程中的數值以及twiddle factor的值。

本設計採用32點輸入，每個輸入點之實部有11bit，小數點前有五位，小數點後有六位，我們假設時域訊號不會有複數，因此輸入點虛部為零，電路架構如下圖

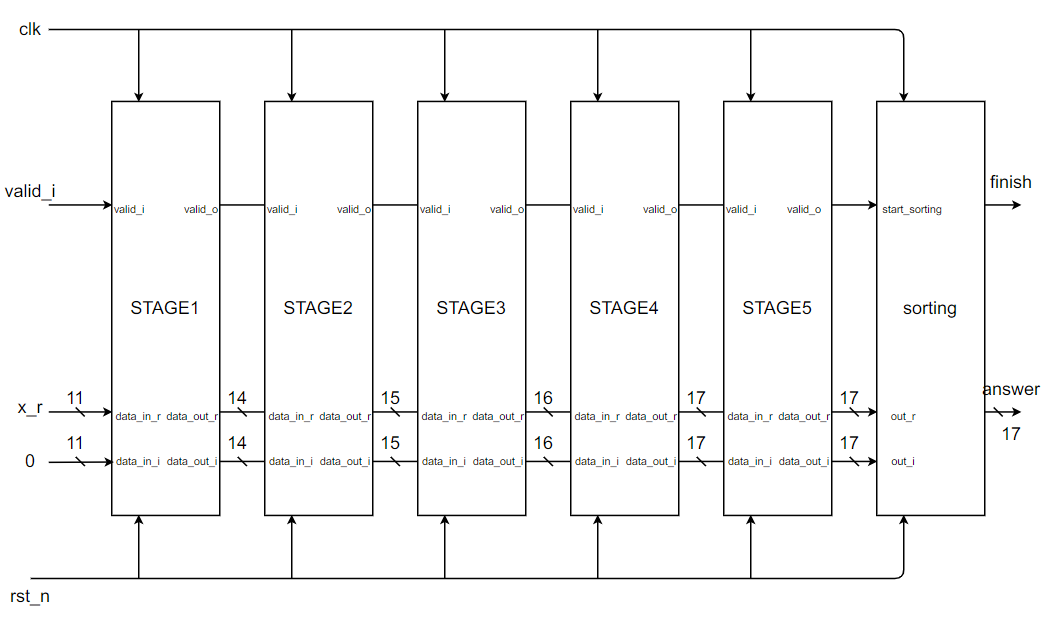
****

Fig.(6) 整體架構

其中每個stage皆為Serial-in-serial-out，使用shift register來暫存數據，並使用counter-based control unit來控制twiddle factor、state以及output，stage的工作流程可以劃分成4個state：IDLE, WAITING, FIRST, SECOND，以下講解各個state工作內容：

1. IDLE：等待輸入。
2. WAITING：從吃進第一個輸入開始，會依序將輸入值填進shift register，此state恰好填滿時結束，即進入下一個state FIRST。
3. FIRST：此state將shift register暫存的值讀出，與當下的輸入值相加輸出（輸出值為推導中的G[k]），並將新輸入的值減去shift register吐出的值再填入shift register，當輸出一半的值後，即進入下一個state SECOND。
4. SECOND：將shift register吐出的值與control unit給的twiddle factor相乘並輸出，輸出完畢回到IDLE。相乘之後的結果我們一律保留小數點後7位，避免運算到後方的stage時計算量太大。

除此之外，我們在每個stage之中傳遞資料的時候每次用來表示整數的bit數會增加1，因為考量到在運算的過程中會經過加法和乘法，其中乘法皆是乘以不大於1的數字，因此不必考慮overflow，但加法可能會導致overflow的情形發生，因此我們在每一個stage之間將data的總bit數增加1來避免overflow。

在經過五個stage之後，理論上應該已經將完成快速傅立葉變換的數據輸出，但實際上出來的順序並不是按照X[0],X[1]…X[31]的數據排列的(可以參考上方Fig.(5))，因此我們需要做bit reversal的排列使得最後的輸出可以按照X[0] ,X[1]…X[31]的順序排列，Fig.(6)中的sorting即是用來實現此bit reversal排列的電路。

此外，輸出腳位為answer，共17bit，其中小數部分占6bit，整數部分占11bit，考量到晶片以及量測用的儀器腳位限制，我們必須將此晶片的總腳位數量控制在40以內，因此我們最後的輸出按照順序會是X[0]到X[31]的實部，再來是X[0]到X[31]的虛部，最後總共會花64個cycle來將結果順序以及實部虛部輸出。Fig.(6)中的sorting除了解決上一段中的bit reversal問題，也用來控制將實部和虛部依序輸出。

具體作法如下：

首先，我們觀察Fig.(5)當中的簡化模型，我們可以發現，輸出的X[n]並不是按照順序的結果，不過，我們可以經由完整的DIF FFT流程圖，整理出實際上的output順序，實際的作法為，一步一步照著演算法的設計，觀察每個stage的output順序，得到最後的輸出順序如下圖

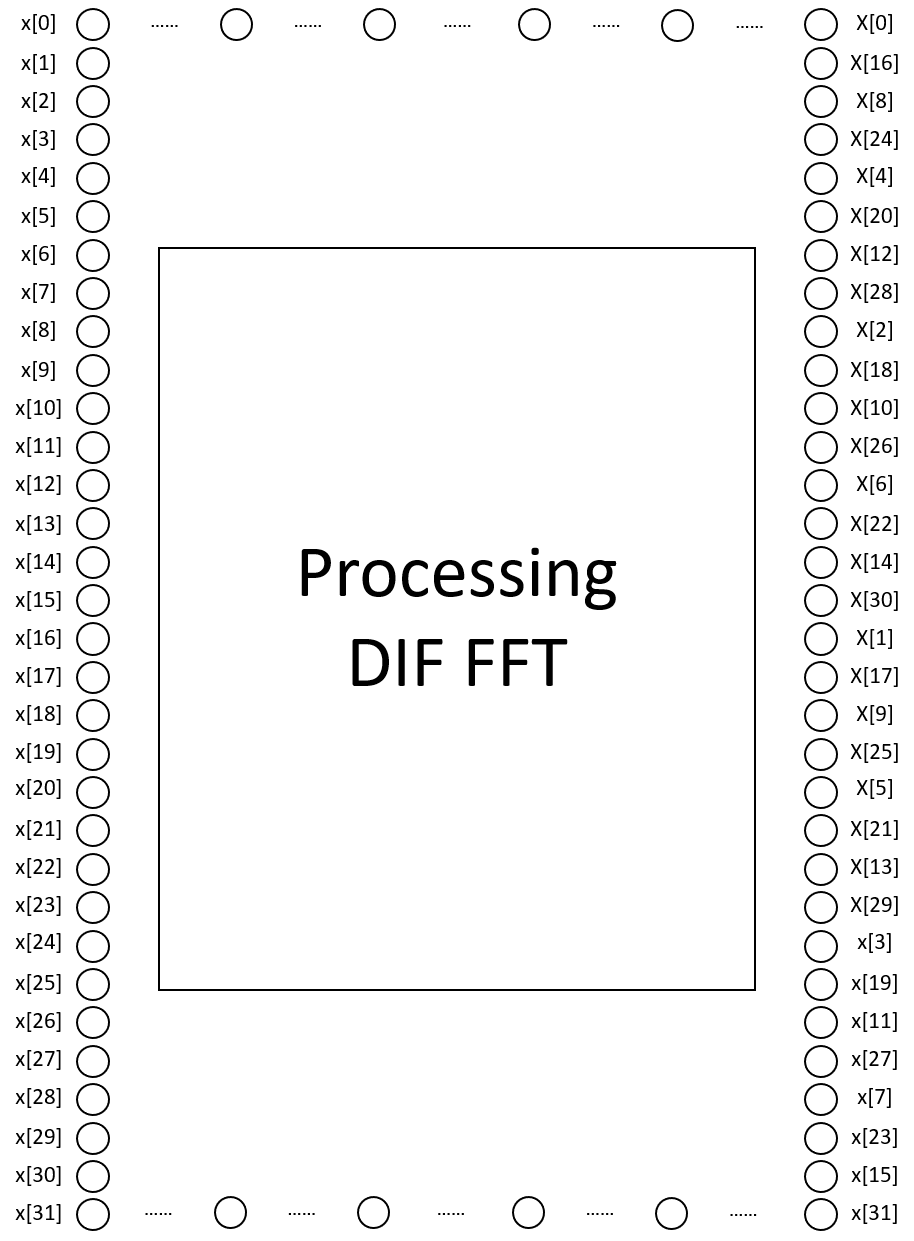


Fig.(7) 完整DIF FFT輸入順序及輸出順序示意圖

其中，左邊直行由上到下代表input的輸入順序，右邊直行從上到下代表output的輸出順序。因此，我們在得知output的資訊後，我們設計了一個做bit reversal的module，將所有的output暫存至一個64(個)\*17(位)的register中，再將這些output按照正確的順序一個一個輸出，先輸出X[0]~X[31]的實部，再輸出X[0]~X[31]的虛部。

最後，下表是本晶片的輸入及輸出腳位配置：

|  |  |  |  |
| --- | --- | --- | --- |
| Name | I/O | Width | Description |
| clk | I | 1 | Positive edge trigger clock |
| valid\_i | I | 1 | Active-high signal that tells if input is valid |
| rst\_n | I | 1 | Asynchronous active-low reset signal |
| x\_r | I | 11 | Serial-in time domain input signal |
| finish | O | 1 | Asynchronous active-high signal that indicates the data output is done |
| answer | O | 17 | Serial out frequency domain output signal |

表一 輸入及輸出腳位

在設計電路之外，我們另外寫了python檔來驗證電路的正確性，其具體功能包括將數值轉換為二進位fixed-point、以及按照電路順序生出每一個stage過後的output(binary)，還有將binary數據轉化為我們常看到的小數格式並劃出spectrum功能，透過python輔助產生的Expected output，我們可以在testbench中讀取這些檔案，將其與電路的output做比較來驗證電路的正確性。總共使用十組數據來驗證電路的正確性，最後的結果是十組數據皆能成功，畫出來的spectrum也和matlab內建的FFT畫出來的spectrum雷同。

**[5] 設計流程**

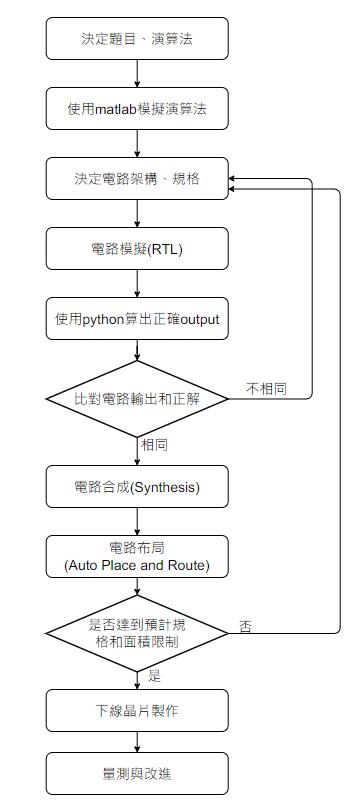
****

Fig.(8) 設計流程圖

**[6] 電路詳圖**

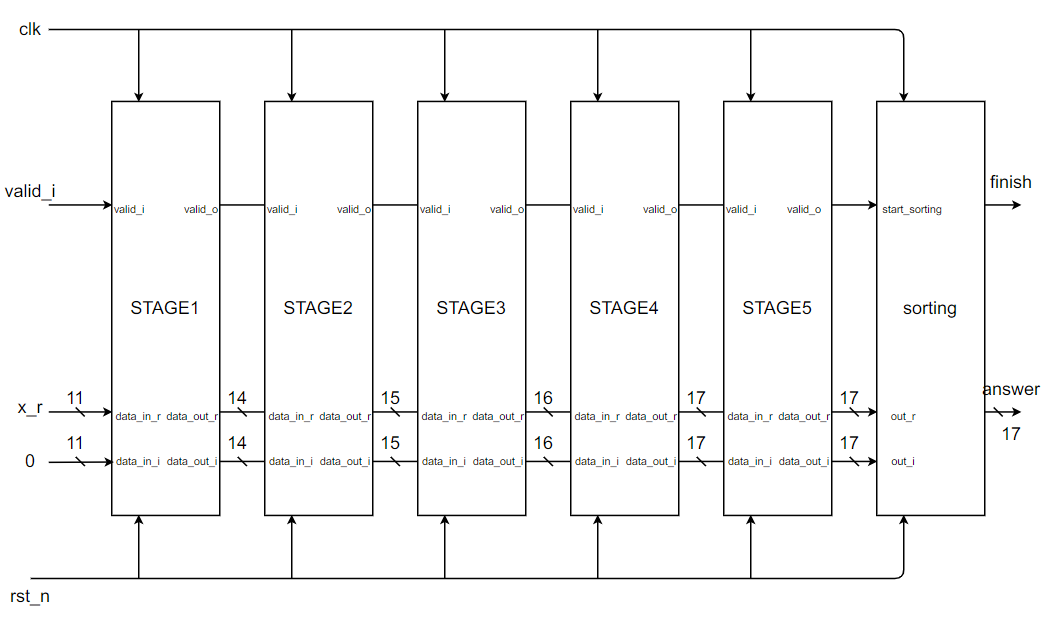
****

Fig.(9) 整體架構

其中，每一stage電路圖為butterfly架構，如下(以stage1為例)，其中SR16為長度16的shift register而下方的CTRL16則是用來控制state(state控制了送進shift register的數是從上一級的output還是此stage中的減法結果、以及此stage的output是加法結果還是減法和twiddle factor的乘積)以及送出twiddle factor給上方之乘法器

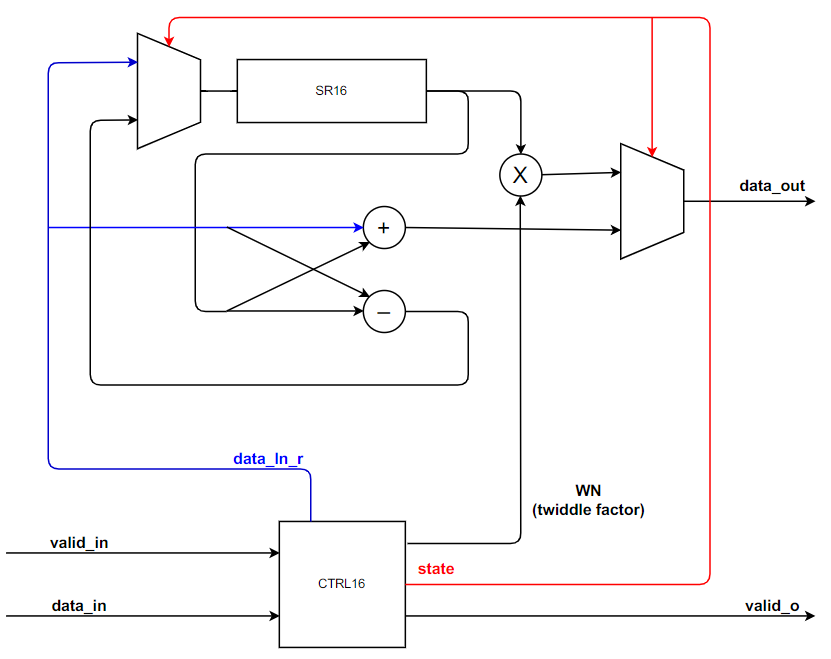
****

Fig.(10) stage1架構圖

**[7] 模擬結果**

波形圖：(時間單位ps，週期為10ns)

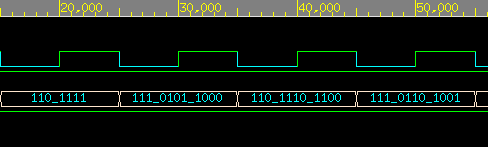


Fig.(11) 輸入波形圖(節錄)

上圖為Input波形圖，每一個cycle會輸入一組以二補數表示，小數部分占6bit，整數部分占5bit的數值(第三個波形)，連續輸入32個cycle，分別代表x[0], x[1]…x[31]，當32個cycle結束後，會將valid訊號拉為0，代表輸入結束。

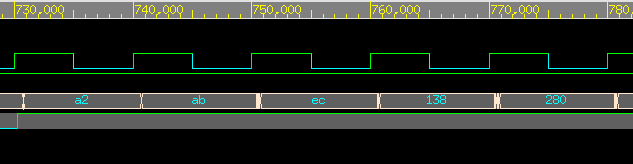


Fig.(12) 輸出波形圖(節錄)

上圖為output波形圖，當計算完成後，會依據傅立葉變換之後的結果，將X[0], X[1]…X[32]的實部輸出，接著再將X[0], X[1]…X[32]的虛部輸出，其數值以以二補數表示，小數部分占6bit，整數部分占11bit(第三個波形)，在輸出的同時將finish拉至1，代表有效的輸出。完成整個快速傅立葉變換(包含排列輸出次序)需時約72個cycle。

在完成快速傅立葉變換之後我們可以將晶片的輸出透過python來畫出spectrum，並和matlab模擬出來的結果相比較，我們共準備了十組數據，對每一組數據，晶片都能確實輸出正確答案，以下提供了其中兩組數據，可以看見圖形幾乎完全雷同。

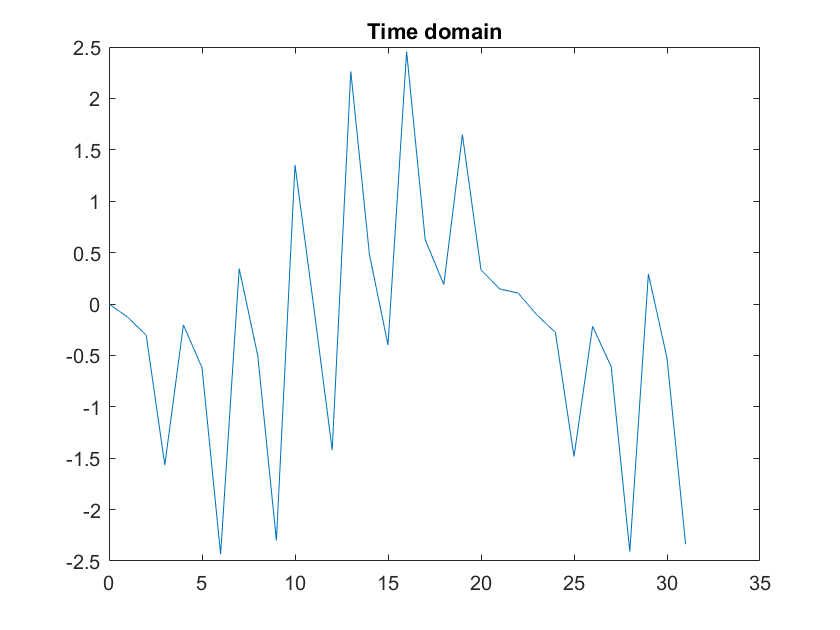


Fig.(13) x = sin(2t)+sin(4t)+sin(6t)

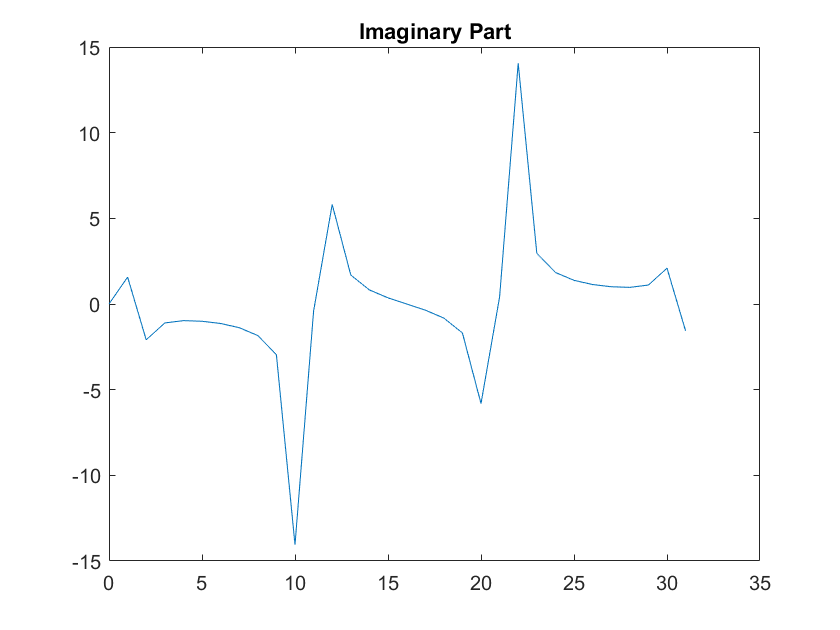
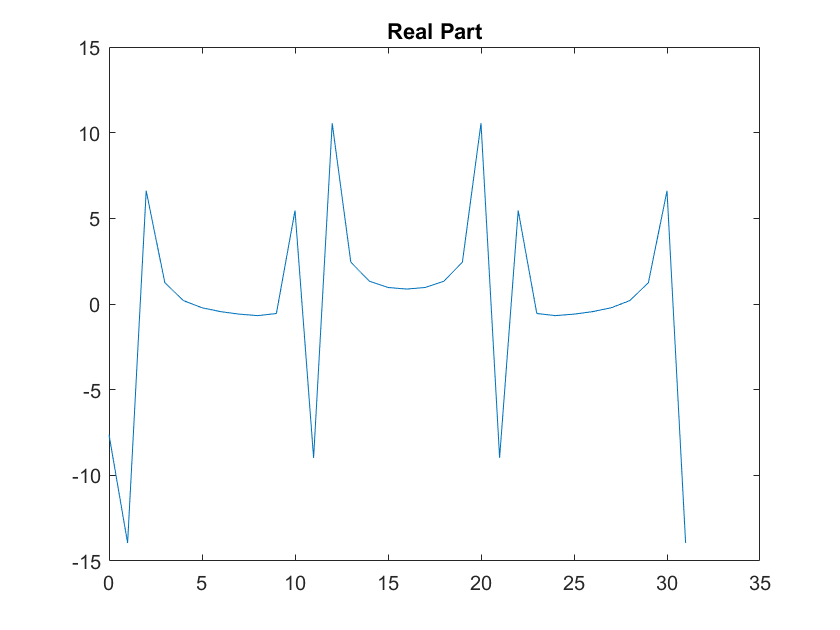


Fig.(14) matlab模擬出之頻譜

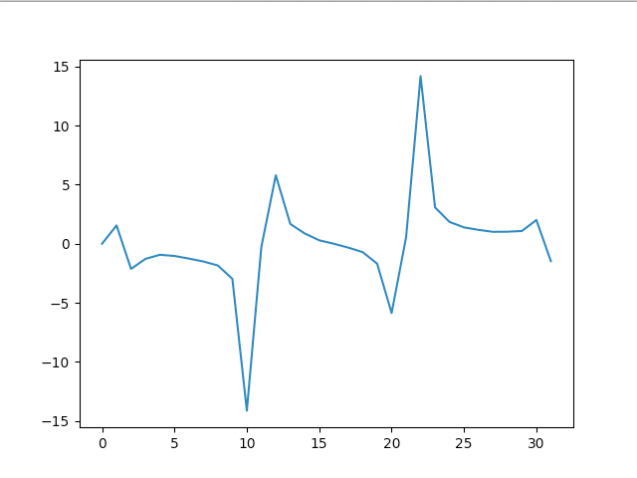
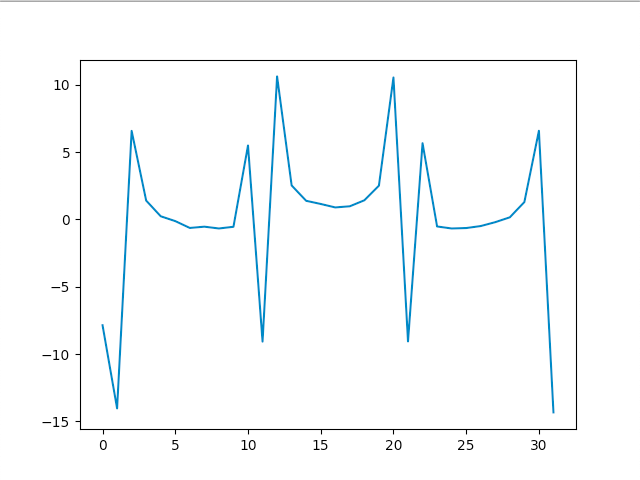


Fig.(15) 晶片輸出之頻譜

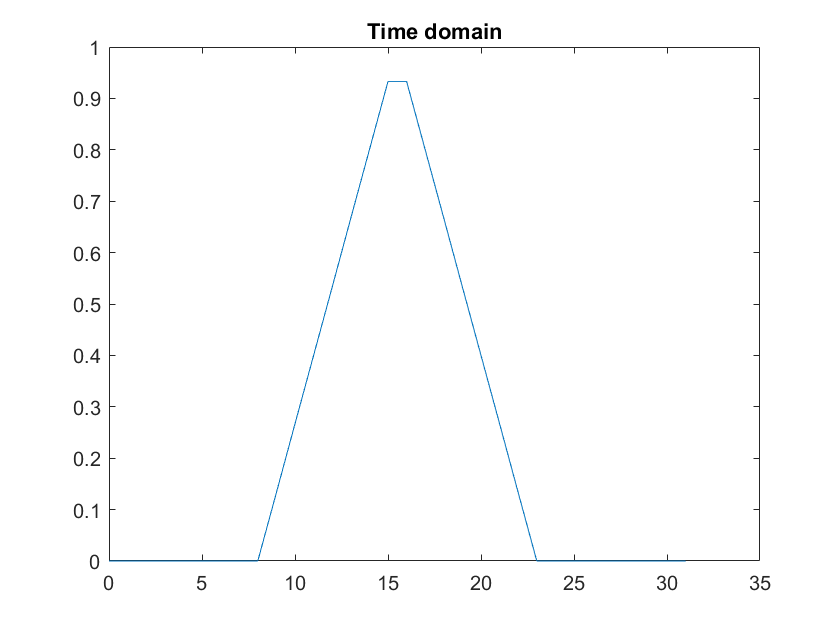


Fig.(16) x = traingle(8,23,t)

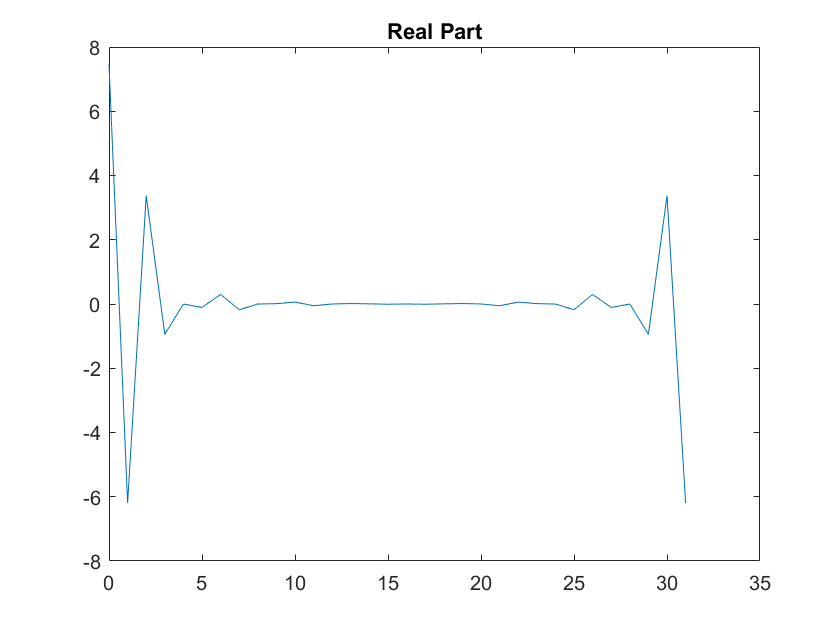
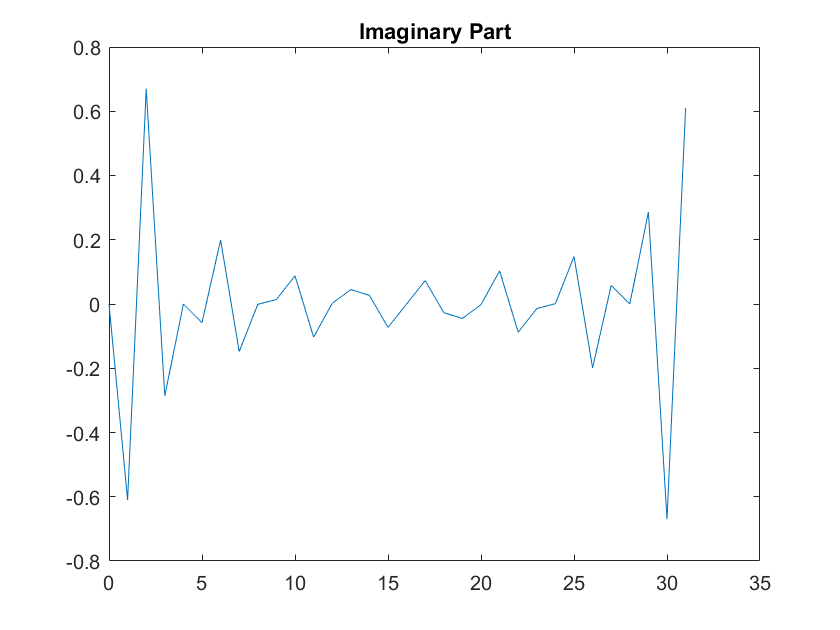
 

Fig.(17) matlab模擬出之頻譜

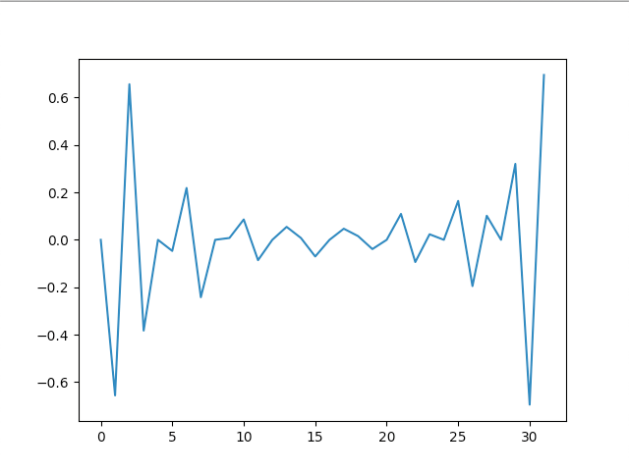
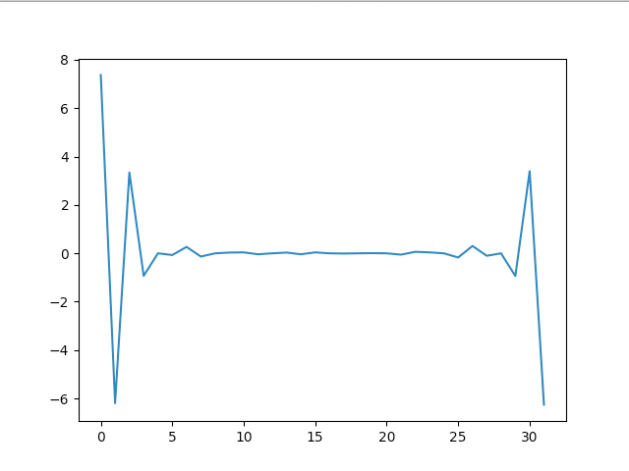


Fig.(18)晶片輸出之頻譜

**[8] 量測考量**

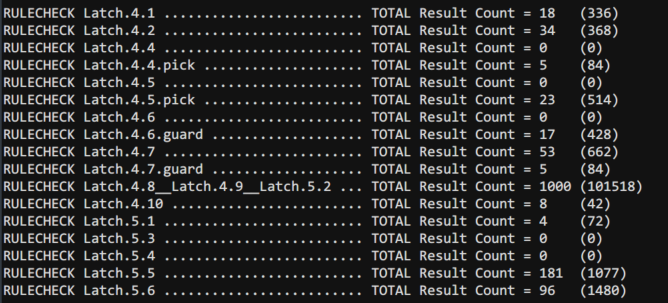
**預計量測流程**

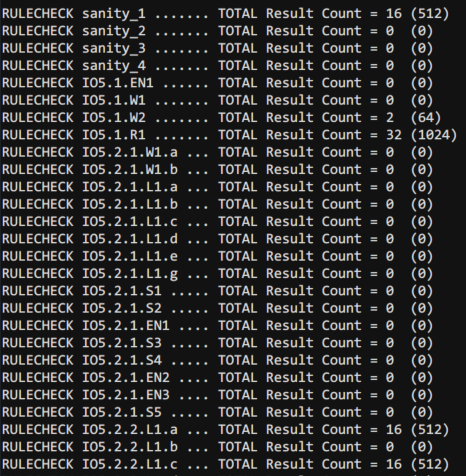
1. 將準備好之input pattern轉換為fixed point，輸入按照順序準備好，利用訊號產生器向晶片輸出，此為測試資料
2. 儀器設置：電源供應器調整為1.8V直流電，並接上晶片的電源腳位
3. 儀器設置：訊號產生器接到晶片的輸入腳位，並產生100MHz的方波，輸入clk腳位
4. 儀器設置：將邏輯分析儀接制晶片的輸出腳位
5. 使用訊號產生器像晶片輸入測試資料
6. 將晶片輸出與python模擬出來的output binary pattern做比對
7. 確認比對結果是否一致

**[9] 佈局驗證結果錯誤說明**

(A) DRC驗證結果：(到時候要用Qserver的截圖)

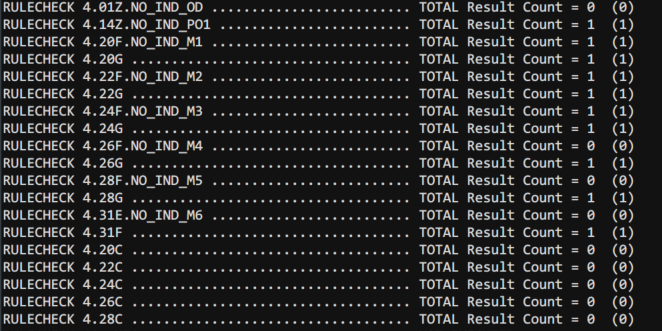
所有錯誤如下圖，皆為可容許之錯誤



(LATCH\_UP.rep)  


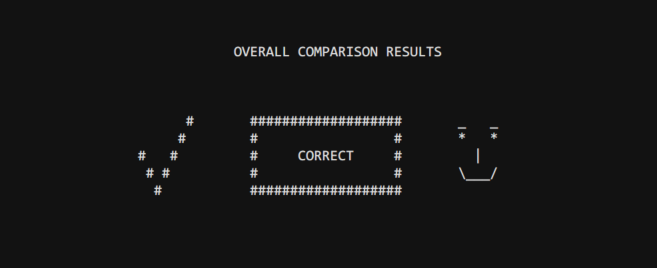
(ESD.rep)





(DRC.rep)

(B) LVS驗證結果：通過

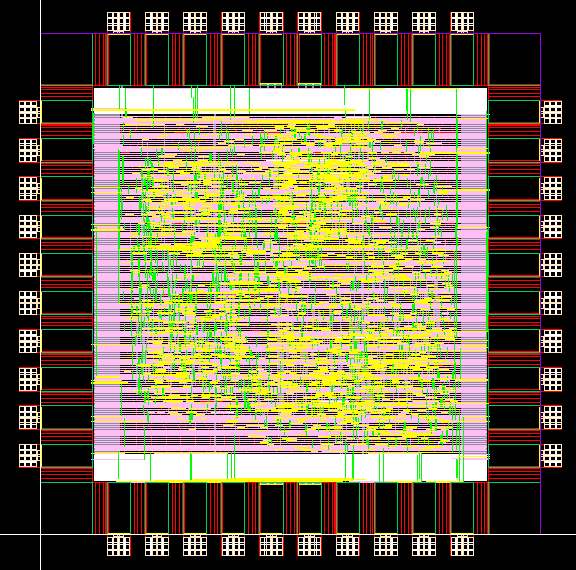


**[10] 佈局平面圖**

Chip Size: 15000x1500

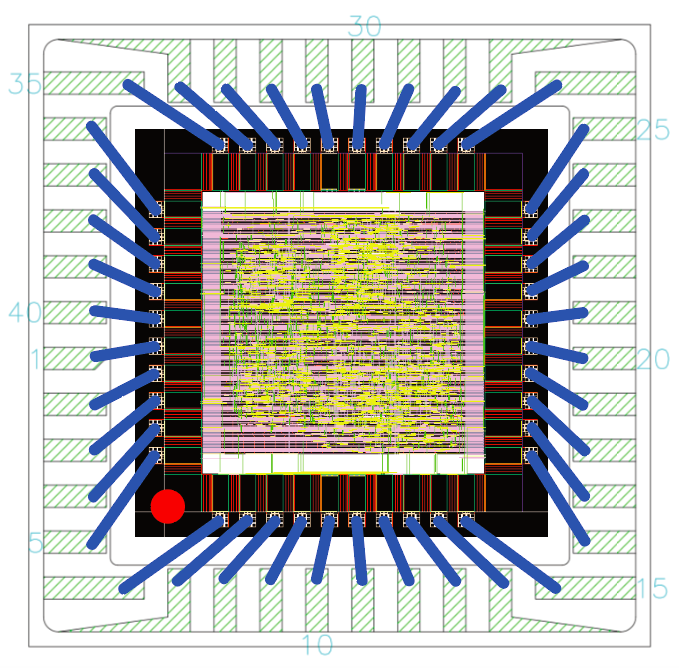
Power Dissipation: 32.8 mW

Max. Frequency: 100MHz



**[11] 打線圖**

SB40

****

**[12] 預計規格列表**

|  |  |  |  |
| --- | --- | --- | --- |
| **Specification** | **Spec.** | **Pre-sim(tt)** | **Post-sim(tt)** |
| **Supply Voltage** | **1.8V** | **1.8V** | |
| **Freqeuncy** | **100MHz** | **100MHz** | |
| **Chip size (μm2)** | **1500x1500** | **1500x1500** | |
| **Power** | **-** | **32.8 mW** | |
| **PADs** | **40** | **40** | |

**[13] 參考文獻**

[1] Ying, G. (2015). *Hardware Implementation of a 32-point Radix-2 FFT Architecture.* (Master’s Thesis, Lung University, Department of Electrical and Information Technology). Retrieved from <https://www.eit.lth.se/sprapport.php?uid=856>

[2] 陳振模 (2004)。快速傅立葉轉換之複數乘法器設計及其系統整合（碩士論文）。取自<https://ir.nctu.edu.tw/bitstream/11536/70434/4/360104.pdf>