112061621 貢暐家

COM519000: Digital Communication Integrated Circuits

Final Project

(一)Final project introduction

本次期末專題我主要實作兩個方向：

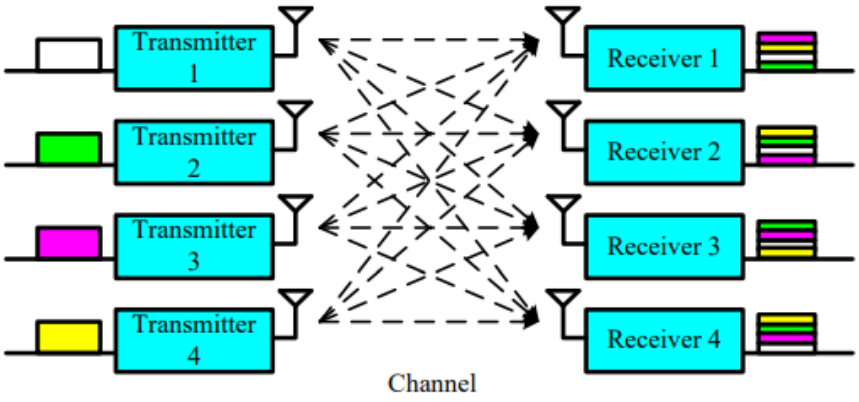
(1) 使用 MIMO 提升 OFDM 系統的傳輸速率

(2) 在 Synchronization 中加入 short preamble 及 long preamble 實作 coarse correlation 和 fine correlation

1. MIMO

優點:

MIMO（Multiple Input Multiple Output，多輸入多輸出）是一種廣泛應用於無線通信的技術。透過在發射端和接收端使用多根天線，MIMO 系統可以同時傳輸和接收多條信號路徑，從而顯著提升通信效能。此外，由於每單位頻寬能傳輸更多數據，因此也能同時提高頻譜效率。

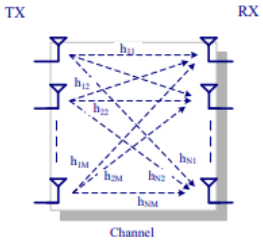


缺點:

理想上 MIMO 應該能達到 N 倍的 Throughput 提升，然而事實卻並非如，因為考慮到天線之間的訊號會互相干擾，如下圖。因此需要在 receiver 端進行複雜的 Detect 運算來將互相干擾的部分補償掉，這會大大增加硬體設計的複雜性與成本，且對於 Synchronization 的準確率要求較高，將在下一點中詳述。

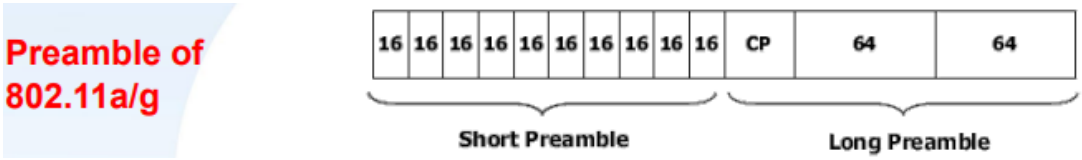
$$y = Hs + n$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1M} \\ h_{21} & h_{22} & \cdots & h_{2M} \\ \vdots & & \ddots & \vdots \\ h_{N1} & \cdots & \cdots & h_{NM} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_M \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_N \end{bmatrix}$$



2. Coarse & Fine correlation

由於 MIMO 對於是否能同步到正確信號的要求較高，接收端需要精確地對齊時間和頻率，以解碼數據，因此需要在輸出端先加入 short preamble 及 long preamble，我採用的是下圖的 preamble 規格。以方便之後的操作



首先，Coarse Correlation，主要目的是在較大的範圍內使用 auto correlation 快速檢測信號的存在，並初步估算時間和頻率偏移，不需要過於精準，只需超過一定的 Threshold 即認定為有效訊號，公式如下

$$\Phi_{\mathrm{DC}}(m) = \left| \sum_{r=0}^{R-1} z_{m-r} z_{m-r-L}^{*} \right|,$$

if (P_received > P_threshold)

Signal Detected = 1

else

Signal Detected = 0

此方法可以大致估算出 short preamble 的起點，再往後加 160，便能大致得出 CP 的起點，需要注意的是由於這步驟的結果只需超過一定的 Threshold 就算完成，因此得出的 CP 起點 不一定準確，故需要再做下一步的 Fine correlation

Fine correlation 是在 Coarse correlation 的基礎上進一步校正，將前一步去除 short preamble 的訊號與已知的 long preamble 訊號進行 cross correlation 以精準定位 long preamble 的起點，確保符號和頻率對齊，提升整體通信性能，公式如下。

$$\Phi_{zp}(m) = \sum_{q=0}^{Q-1} z_{m+q} \cdot p_q^*$$

$$\hat{m}_{\mathrm{MAX}} = \arg\max_m |\Phi_{zp}(m)|$$

找到 long preamble 的起點後加 128 便可得到真正訊號的起點。

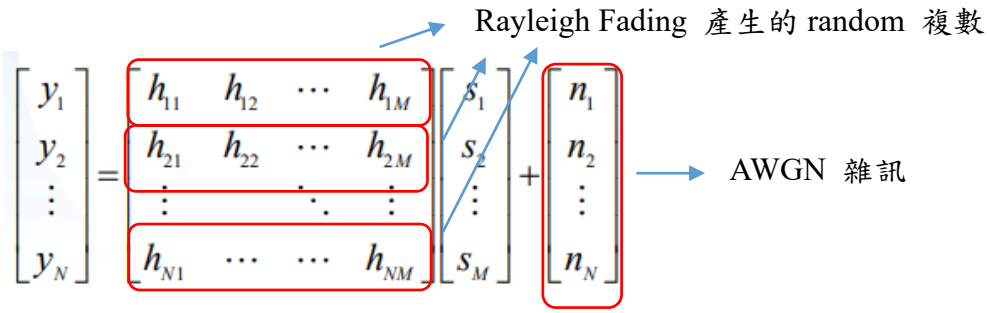
(二)MIMO channel model

Multipath Fading 指無線通信中，信號從發射端到接收端的傳輸路徑不止一條。信號在遇到建築物、地面、山體等障礙物時，可能會發生反射、折射、散射和繞射，形成多條傳輸路徑，導致信號強度的隨機波動。而其中 Rayleigh Fading 是一種特定的多徑衰落模型，屬於 Non-Line-of-Sight, NLOS 的無線環境，公式如下

- Mean, $E\{g_r\}=0$
- $|g_r|$, Rayleigh distribution
- $\angle g_r$, uniform distribution

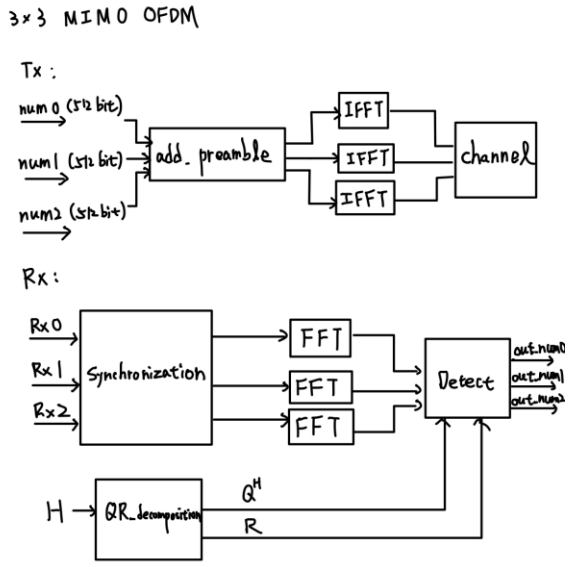
$$f(x;\sigma) = \frac{x}{\sigma^2} e^{-x^2/2\sigma^2}, \quad x \geq 0,$$

本次實驗的 matrix H 便是要模擬 3x3 MIMO 並額外加上 Rayleigh Fading 作為 channel model。



(三)C code simulation 3x3 MIMO OFDM

以 C code 模擬整個 3x3 MIMO OFDM 流程大致如下圖



首先，產生三組 512bit 的 random bit 將他們經過 16-QAM mapping 後變成三組 128 個的複數，加入 preamble 後進行 IFFT，再經過 channel 並乘上 channel matrix H 及加上 Gaussian noise，由於我想要檢測 synchronization 是否能準確運作，因此會在原本的訊號前方加入 20 組雜訊，故 Tx 輸出的訊號有三組各 500 筆複數

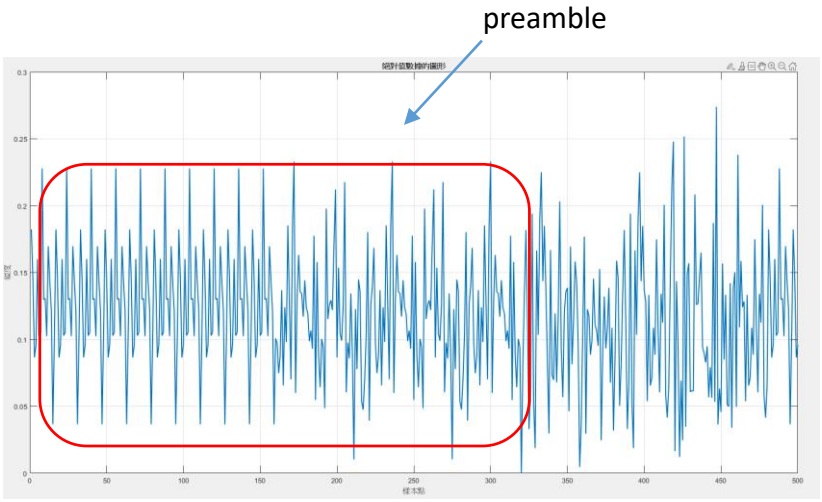
由於需要進行複數運算因此原本的 3x3 複數矩陣 H，可以變成旋轉矩陣的形式，這樣就只要做矩陣乘法，較為簡單。

$$H_{r1} = \begin{bmatrix} \Re(H) & -\Im(H) \\ \Im(H) & \Re(H) \end{bmatrix}$$

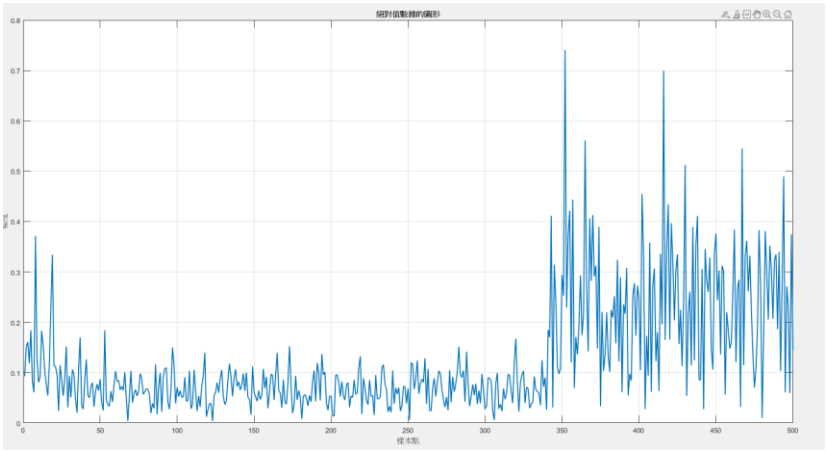
轉變為 6x6 channel matrix:

```
H matrix:
-0.2966 0.1470 -0.9009 0.6621 -0.2344 0.7204
-0.1470 -0.2966 -0.6621 -0.9009 -0.7204 -0.2344
-0.9320 -0.9972 0.1579 1.2517 0.6547 -0.7098
0.9972 -0.9320 -1.2517 0.1579 0.7098 0.6547
-0.6359 0.2305 -0.3620 -0.4933 0.2254 -0.1620
-0.2305 -0.6359 0.4933 -0.3620 0.1620 0.2254
```

Tx1 無 channel model:



Tx1 有 channel model:



由上面兩張圖，可看出插入 preamble 及乘上 channel matrix 的功能正確，之後便可以進行輸出到 Rx

Rx 端首先進行之前提到的 Coarse & Fine correlation 得到結果如下

```
coarse_result:17
fine_result:340
```

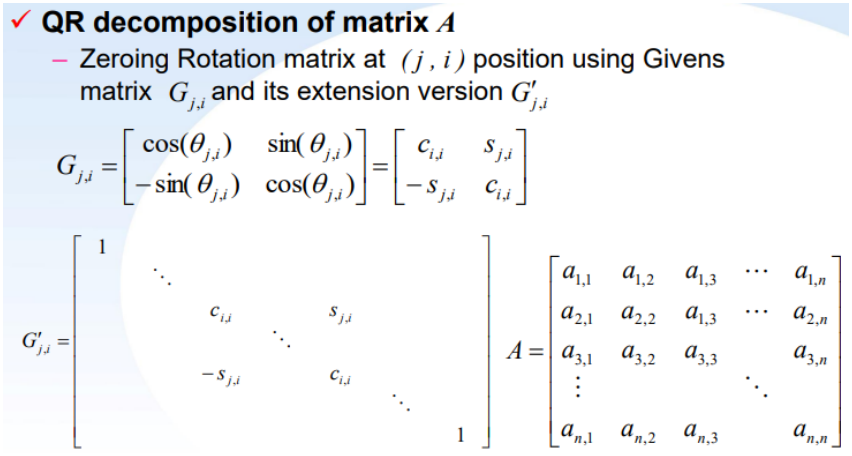
Coarse result 為 17 是因為我們在前端加入了 20 個雜訊，因此他大約算出 short preamble 起始為 17，故功能正確，而 Coarse 中的 threshold 經過觀察後，我訂為 1 方便在硬體上比較。

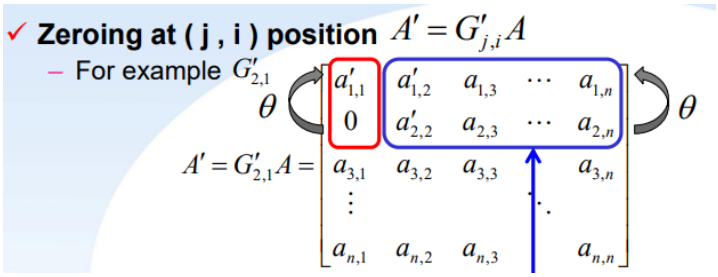
Fine result 為 340 是以 Coarse result+160 為起點計算往後 64 個訊號中的最大值，最終找到 35，因此為 17+160+35+128=350，符合我們在原始數據前加入 20 個雜訊和 320 個 preamble 的結果，故功能正確。

```
14 0.06445
15 0.05664
16 0.16992
17 0.10742
18 0.11719
19 0.13477
20 0.17383
21 0.08203
22 0.12891
23 0.18750
24 0.05469
25 0.24805
26 0.08008
27 0.17383
28 0.14453
29 0.06641
30 0.10352
31 0.23828
32 0.19531
33 0.08594
34 0.25391
35 1.95313
36 0.25195
37 0.11719
38 0.22070
39 0.19336
40 0.13477
41 0.08008
42 0.12500
43 0.16602
44 0.09766
```

且與之前作業的 Synchronization 正確率相比，此次 Coarse & Fine correlation 的正確率為 100%，即使 SNR:0dB 也是如此

接下來進行 FFT 、QR_decomposition 及 Detect，由於 FFT 作業已經做過，故不在此贅述，首先介紹 QR 分解是將 H 矩陣分解為正交矩陣 Q 和上三角矩陣 R，至於目的之後會提到，做法是不斷乘上一個矩陣 G 使其漸漸將 H 矩陣轉變為上三角矩陣 R，而矩陣 G 的不斷乘積就是我們之後會用到的 Q_H。





演算法:

✓ **Givens rotation algorithms for QR decomposition**

For i=1 to N-1

For j=i+1 to N-1

$$c_{i,i}=\frac{a_{i,i}}{\sqrt{a^2_{i,i}+a^2_{j,i}}} \quad s_{j,i}=\frac{a_{j,i}}{\sqrt{a^2_{i,i}+a^2_{j,i}}}$$

$$G_{j,i}=\begin{bmatrix}\cos(\theta_{j,i}) & \sin(\theta_{j,i})\\ -\sin(\theta_{j,i}) & \cos(\theta_{j,i})\end{bmatrix}=\begin{bmatrix}c_{i,i} & s_{j,i}\\ -s_{j,i} & c_{i,i}\end{bmatrix}$$

$A'=G'_{j,i}A$ $\quad G'_{j,i}$ is extension of $G_{j,i}$

For k=i+1 to N

$$G_{j,i}\begin{bmatrix}a_{i,k}\\ a_{j,k}\end{bmatrix}=\begin{bmatrix}a'_{i,k}\\ a'_{j,k}\end{bmatrix} \quad k=i+1,\ldots,N$$

End k

End j

End i

實作演算法後，我們能得到

```
Q_real matrix:
-0.1914 -0.0948 -0.5985 0.6388 -0.4077 -0.1479
0.0952 -0.1903 -0.6391 -0.5985 0.1479 -0.4077
-0.6438 -0.3878 0.0526 -0.3772 -0.4047 0.3553
0.3880 -0.6443 0.3762 0.0527 -0.3553 -0.4047
-0.2881 -0.5536 0.0026 0.2977 0.7223 0.0121
0.5534 -0.2878 -0.2984 0.0031 -0.0121 0.7223
R matrix:
1.5589 0.0015 -0.5842 -0.4350 0.0589 0.7600
0.0000 1.5589 0.4339 -0.5854 -0.7611 0.0591
0.0000 0.0000 1.6390 0.0005 0.1633 -0.5115
0.0000 0.0000 0.0000 1.6384 0.5112 0.1643
0.0000 0.0000 0.0000 0.0000 0.8442 0.0009
0.0000 0.0000 0.0000 0.0000 0.0000 0.8447
```

證明此功能正確

而之所以要做 QR 分解是為了要將 channel matrix 消除，以補償 MIMO 及 Rayleigh Fading 後的失真，將 H 矩陣分解為正交矩陣 Q 和上三角矩陣 R 最後在輸出端乘上 Q_H，讓他變成原本進入通道前的訊號與上三角矩陣 R 的乘積，便可依次解出進入通道前的訊號，以此達到補償，這便是 Detect 的功能。

Detect 部分演算法，我則是使用 QR_decomposition base 的 SIC detector

✓ **QR decomposition of channel matrix $\mathbf{H}$ to reduce the computation complexity of MMSE-SIC or ZF-SIC detector.** $\mathbf{Q}^H\mathbf{Q}=\mathbf{I}$

✓ **$\mathbf{Q}$ is unitary matrix such that**

$$y=\mathbf{H}x+n$$
$$y=\mathbf{QR}x+n$$
$$\Rightarrow \mathbf{Q}^H y=\mathbf{Q}^H\mathbf{QR}x+\mathbf{Q}^H n$$
$$\Rightarrow \hat{y}=\mathbf{R}x+\hat{n}$$

$$\begin{bmatrix}\hat{y}_0\\ \hat{y}_1\\ \hat{y}_2\\ \hat{y}_3\end{bmatrix}=\begin{bmatrix}R_{00} & R_{01} & R_{02} & R_{03}\\ 0 & R_{11} & R_{12} & R_{13}\\ 0 & 0 & R_{22} & R_{23}\\ 0 & 0 & 0 & R_{33}\end{bmatrix}\begin{bmatrix}x_0\\ x_1\\ x_2\\ x_3\end{bmatrix}+\begin{bmatrix}n'_0\\ n'_1\\ n'_2\\ n'_3\end{bmatrix}$$

QRSIC detection

(1) $\hat{\mathbf{y}}=\mathbf{Q}^{\mathrm{H}}\mathbf{y}$

(2) for $i=n_t,\ldots,1$

(3) $\hat{\mathbf{x}}_i=Q\left(\frac{\hat{\mathbf{y}}_i-\mathbf{R}_{i+1,i+1:nt}\times\hat{\mathbf{x}}_{i+1:nt}}{\mathbf{R}_{i,i}}\right)$

(4) end

$$\hat{x}_3=Q\left(\frac{\hat{y}_3}{R_{33}}\right)\qquad x_3\leftarrow\hat{x}_3$$
$$\hat{x}_2=Q\left(\frac{\hat{y}_2-R_{23}x_3}{R_{22}}\right)\qquad x_2\leftarrow\hat{x}_2$$
$$\hat{x}_1=Q\left(\frac{\hat{y}_1-R_{12}x_2-R_{13}x_3}{R_{11}}\right)\qquad x_1\leftarrow\hat{x}_1$$
$$\hat{x}_0=Q\left(\frac{\hat{y}_0-R_{01}x_1-R_{02}x_2-R_{03}x_3}{R_{00}}\right)\qquad x_0\leftarrow\hat{x}_0$$

另外補充，由於在 QR 分解實需要將矩陣不斷乘以 G 矩陣，而乘以 G 矩陣又可視為旋轉一個-degree 角度，且我們原本的 H 矩陣就是由旋轉矩陣所組成的，因此這邊可大量運用 Cordic 演算法，此演算法是為了硬體實作方便且節省資源所開發，詳細如下圖

Rotation mode:

$$x_{i+1}=x_i-y_i\cdot d_i\cdot 2^{-i}$$
$$y_{i+1}=y_i+x_i\cdot d_i\cdot 2^{-i}$$
$$z_{i+1}=z_i-d_i\cdot\tan^{-1}\left(2^{-i}\right)$$
where
$$d_i=-1 \text{ if } z_i<0,\ d_i=+1 \text{ if } z_i>0$$

➔

$$x_n=A_n\left(x_0\cos(z_0)-y_0\sin(z_0)\right)$$
$$y_n=A_n\left(x_0 sin(z_0)+y_0\cos(z_0)\right)$$
$$z_n=0$$
$$A_n=\prod_n\sqrt{1+2^{-2i}}$$

Vector mode:

$$x_{i+1}=x_i-y_i\cdot d_i\cdot 2^{-i}$$
$$y_{i+1}=y_i+x_i\cdot d_i\cdot 2^{-i}$$
$$z_{i+1}=z_i-d_i\cdot\tan^{-1}\left(2^{-i}\right)$$
where
$$d_i=+1 \text{ if } y_i<0,\ d_i=-1 \text{ if } y_i>0$$

➔

$$x_n=A_n\sqrt{x_0^2+y_0^2}$$
$$y_n=0$$
$$z_n=z_0+\tan^{-1}\left(y_0/x_0\right)$$
$$A_n=\prod_n\sqrt{1+2^{-2i}}$$

在計算 QR 分解時，需要先得到一個角度 degree，以形成 G 矩陣，於是我們便可用 Vector mode，將 H[i][i]當作 x，H[j][i]當作 y，初始角度為 0，如此一來，便可以得到角度 degree

$$G_{j,i}=\begin{bmatrix}\cos(\theta_{j,i}) & \sin(\theta_{j,i})\\ -\sin(\theta_{j,i}) & \cos(\theta_{j,i})\end{bmatrix}=\begin{bmatrix}c_{i,i} & s_{j,i}\\ -s_{j,i} & c_{i,i}\end{bmatrix}$$

而乘上 G 矩陣的部分則可視為旋轉一個-degree 角度，，於是我們便可用 Rotation mode 將 H[i][k]當作 x，H[j][k]當作 y，初始角度為-degree，如此一來，便可以得到最後要的乘積 X、Y

$$G_{j,i}\begin{bmatrix} a_{i,k} \\ a_{j,k} \end{bmatrix} = \begin{bmatrix} a'_{i,k} \\ a'_{j,k} \end{bmatrix} \quad k = i+1,\ldots,N$$

將 Cordic 運用在 QR 分解後，並計算在 SNR:15dB 下的 Bit Error rate，結果如下圖，這邊的結果是 Cordic iteration 次數設為 10，原因之後會詳述:

```
Float32:BER:0.01171875
```

由前面的作業結果得知，在沒有 MIMO 及 Rayleigh Fading 情況下的單通道 OFDM，在 SNR:15dB 下的 Bit Error rate 為 0.007856

```
BER:0.007856
```

因此可以看出我們確實有模擬出 MIMO 及 Rayleigh Fading 效應並且能透過 QR_decomposition base 的 SIC detector 補償回去，讓 receiver 端仍能正確 decode

Fixed point:

經過計算及觀察後，我的整個系統最大整數為 4，因此 ineteger bit 數設為 4。

```
max integer:4
```

而 fraction bit 則需要考量 BER，下面會介紹如何選擇，

首先，觀察 fraction bit 至少需要幾 bit

6 bit:

```
Float32:BER:0.01302083
Fixed:BER:0.05533854
```

7 bit:

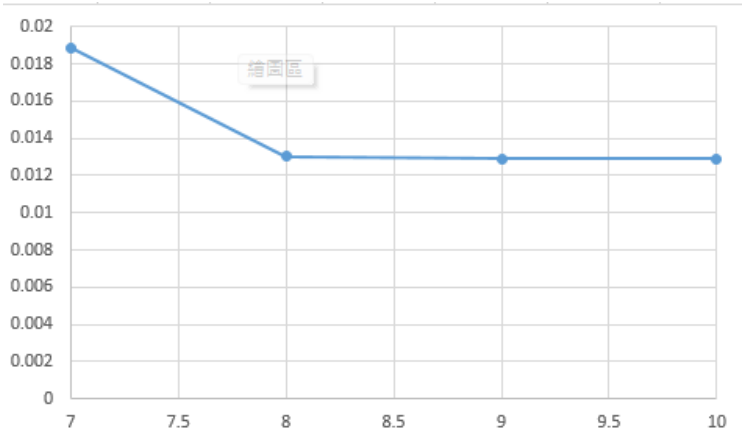
```
Float32:BER:0.01171875
Fixed:BER:0.01888021
```

可以看到 fraction bit 至少需要 7bit 功能才能正確，而由下圖 7bit BER 我們也可知，iteration 次數越大，BER 越小，而由上面公式可得知 iteration 次數最大是 fraction bit 數+1，因此接下來的圖表皆以 iteration 次數=fraction bit 數+1 表示

7bit_iterartion		
	float	fixed
1	0.204427	0.211589
2	0.140625	0.136719
3	0.079427	0.076172
4	0.02474	0.027995
5	0.026693	0.015625
6	0.01888	0.013021
7	0.018229	0.013021
8	0.01888	0.011719

由以下圖表，Fraction bit 數對應 BER 表現，我們能發現當 fraction bit 為 9 以後，BER 變化已經不大，因此我決定 fraction bit 取 9 Bit ，iteration 次數 10 次

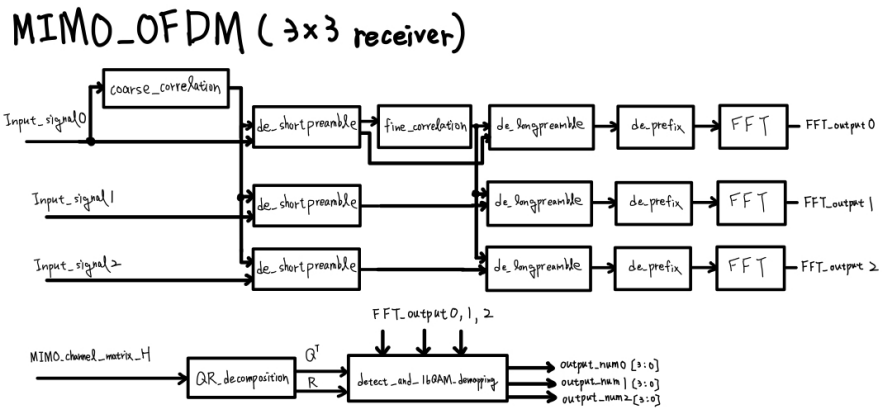
	float	fixed
7	0.011719	0.01888
8	0.011719	0.013021
9	0.011719	0.0129
10	0.011719	0.0129



Word length : ineteger 4 bit + fraction 9 bit，iteration 次數 10 次
BER : 1.29%

(四)Hardware simulation 3x3 MIMO OFDM

首先，整個 MIMO OFDM 的架構圖如下:



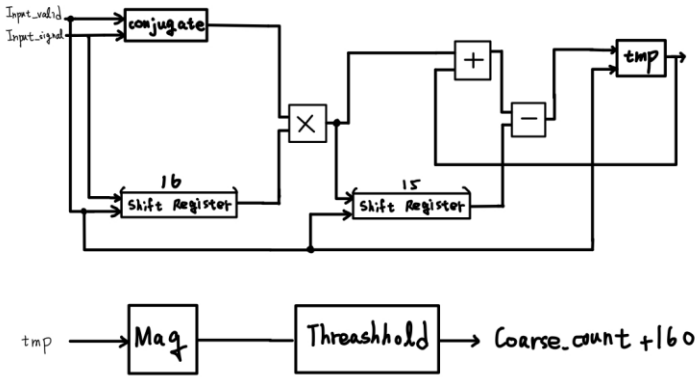
其流程大致與上面的軟體一致，在此不多贅述

Coarse correlation:

首先介紹 Coarse correlation，這部分需要接收 input 訊號並進行 auto correlation 計算，為了追求高 Throughput 我採用 two stage pipeline 形式計算，第一個 stage 是計算內層乘積加總，第二個 stage 則是再將 stage1 跟 Threshold 比較，只要大於 Threshold，便可輸出當時的 count+160 且讓 output_valid = 1 代表完成 Coarse correlation，且我在這之中設計兩個 shift register 以達到 delay 的效果，如此一來輸入便可一直連續進來不用中斷。

$$\Phi_{\mathrm{DC}}(m) = \left| \sum_{r=0}^{R-1} z_{m-r} z_{m-r-L}^{*} \right|,$$

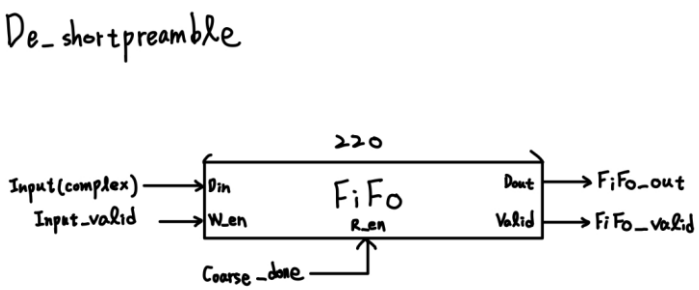
Coarse_correlation



De_shortpreamble:

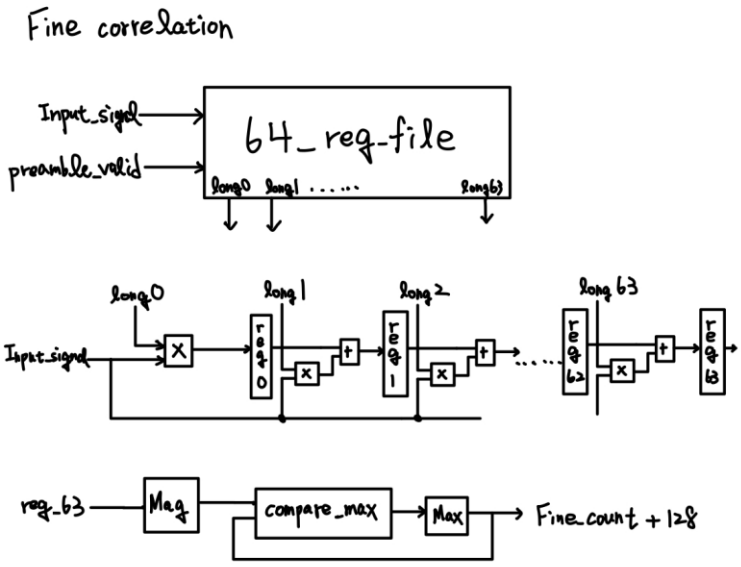
這部分電路是將輸入訊號利用 FIFO 先存起來，等到前一級的 Coarse

correlation 完成後，再根據 Coarse correlation 輸出的 count 決定前面有幾筆資料要丟棄



Fine correlation:

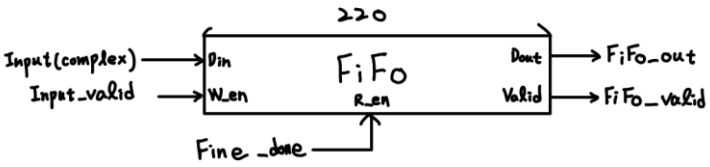
再來介紹 Fine correlation，這部分需要接收 De_shortpreamblegj 輸出的訊號並進行 cross correlation 計算，為了追求高 Throughput 我採用 64 stage pipeline 形式計算，每一個 stage 都是計算前一個 stage 的結果與當下輸入和已知訊號(long preamble)乘積加總，最後會與 Max_register 中存的值比較，如果大於 Max 就取代，並將 count 更新為當下的 count，等到 64 stage pipeline 完成了 64 次以後，輸出 Max_count+128，且讓 output_valid = 1 代表完成 Fine correlation



De_longpreamble:

這部分電路是將輸入訊號利用 FIFO 先存起來，等到前一級的 Fine correlation 完成後，再根據 Fine correlation 輸出的 count 決定前面有幾筆資料要丟棄

De_longpreamble



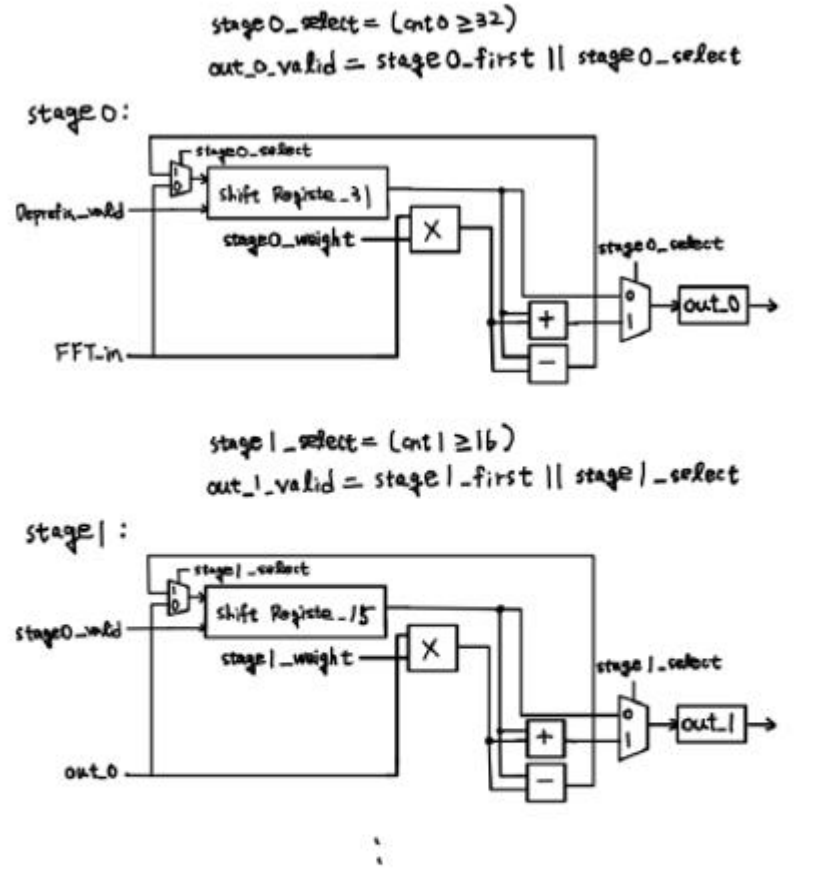
FFT:

為了追求高Throughput我採用sixstage pipeline形式計算FFT，每個stage各自代表64point-FFT中的每一層Butterfly 架構，以stage0為例，當Deprfix_valid為1時，cnt0+1，以此計算input個數。

Stage0_first代表是否為第一次運算，0為是，1為否。當第一次運算時，因為需要經過delay，所以還沒有結果，故out_0_valid為0，但等到cnt0>=32以後，便可以開始輸出0~31的結果了，之後再輸出pair result。

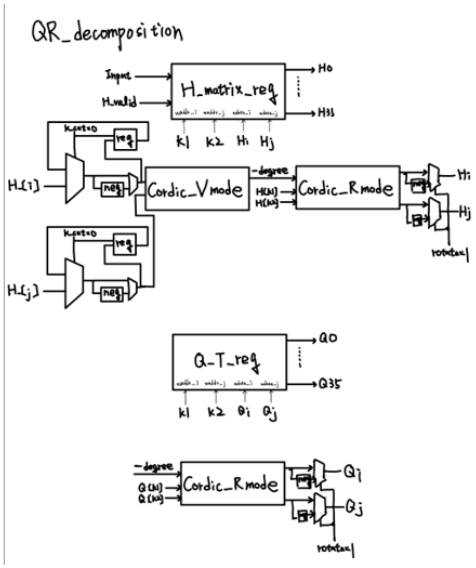
需要注意的是，由於輸出 pair result 時，cnt0 已循環為 0，Stage0_first= 1，但這時剛好是下一組 symbol 的 delay 時間，故可以拿來輸出 pair result，實現 100%的時間利用效率而不需等待。

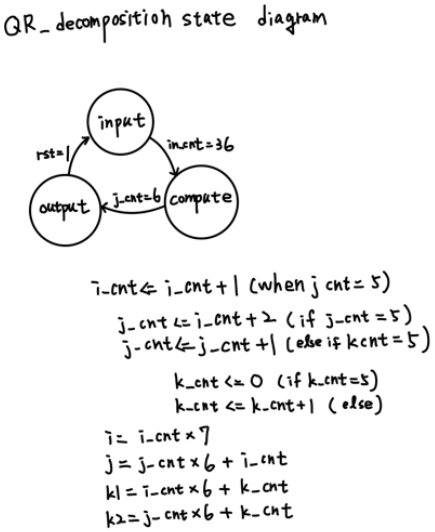
FFT



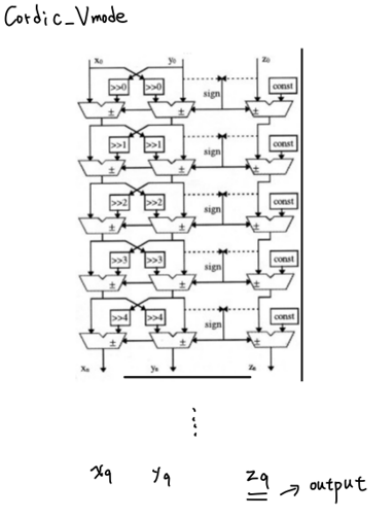
QR_decomposition:

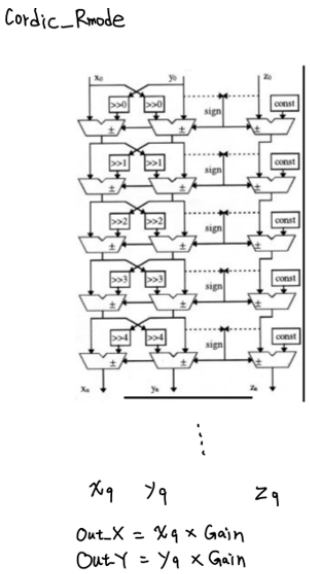
採用 pipeline 架構以增加 throughput，並使用 FSM 區別何時完成 QR_分解



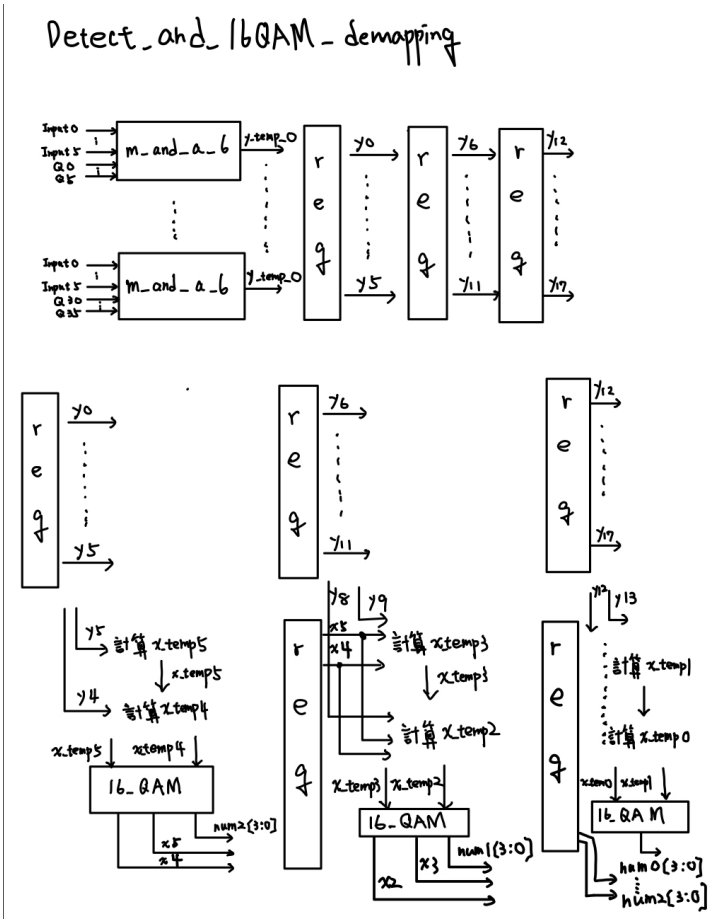


Cordic:





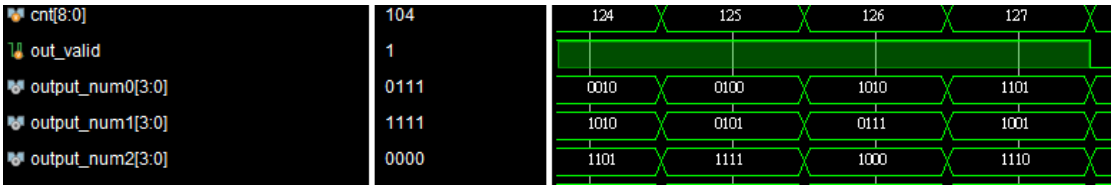
QR_decomposition base SIC detector:
採用 pipeline 架構以增加 throughput，並且讓最後 output_num0 output_num1 output_num2 可以同時輸出



Hardware result:

波型圖:

從圖中可以看出 ouput 數量正確為 128*4*3bit



BER:

從下圖與前面的(fixed)軟體結果(1.29%)比較，可以發現差距並不大，因此可以證明功能正確

```
Total Errors: 38 out of 1536
Error Rate: 2.473958%
```