

多輸入多輸出無線通訊系統偵測器之 FPGA 設計與實作

FPGA Design and Implementation of MIMO Detectors for

Wireless Communications

組員：陳禹嘉, 黃齊緯

指導教授：黃元豪 教授

Abstract

與過去傳送端與接收端都只有一個天線的系統不同，MIMO 系統的兩端有多個天線，可以使通訊資料傳輸量提升，因而被應用於現今許多無線通訊系統的架構中。本專題目標是在 FPGA 上實作出 MIMO 偵測器，我們選擇 K-Best 及 FSD 兩種演算法，並採用 4×4 16QAM 的規格。開發流程為先透過 C++ 模擬出 MIMO 的功能，之後進行定點數模擬決定各訊號位元數，之後設計硬體架構並將之轉換為 Verilog code，最後在燒入到 FPGA 進行結果驗證。

Introduction

MIMO 系統可以表達為下列數學式子：

$$y = Hs + n$$

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{bmatrix} = \begin{bmatrix} h_{11} & \cdots & h_{1N} \\ \vdots & \ddots & \vdots \\ h_{M1} & \cdots & h_{MN} \end{bmatrix} \begin{bmatrix} s_1 \\ \vdots \\ s_N \end{bmatrix} + \begin{bmatrix} n_1 \\ \vdots \\ n_M \end{bmatrix}$$

N 維向量 s 代表傳輸端輸出的訊號；M 維向量 y 代表接收端收到的訊號；向量 n 代表白色高斯雜訊。H 矩陣代表的是訊號經過通道後獲得的增益 (gain)。在本次專題中，我們採用 $N = M = 4$ 的規格。

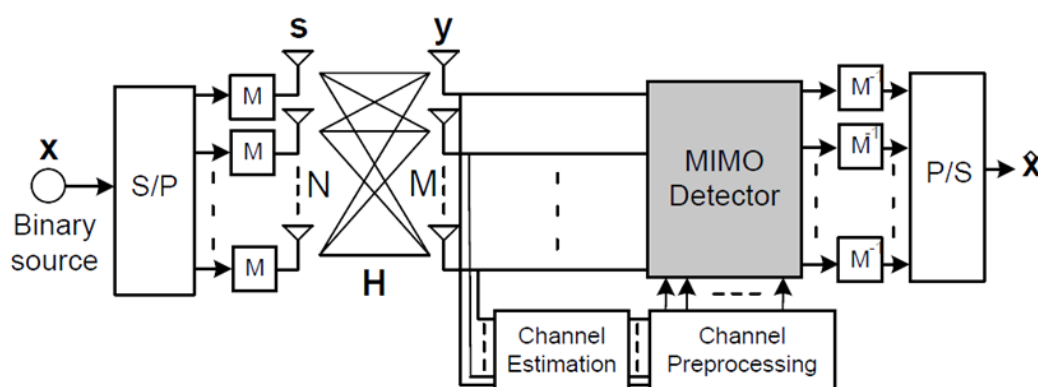


Fig 1. MIMO 系統示意圖。來源:[1]

1. Sorted QR Decomposition

在許多 MIMO 偵測器演算法中，我們都會需要使用到 QR 分解 $H = QR$ ， Q 為 orthonormal matrix， R 為上三角矩陣，而我們真正需要的矩陣為上三角矩陣 R 。在本專題中，我們使用 Givens rotation 搭配 CORDIC 的方法來實現 QR Decomposition，Givens rotation 是藉由乘上旋轉矩陣來消掉對角下方的元素。下面例子為利用旋轉矩陣削掉 a_{21} ，不過要注意的是，當作用在 MIMO 的通道矩陣時，旋轉矩陣要同時作用在與 a_{11} 、 a_{21} 同一個 row 上的所有元素，同一個 row 代表是同一根天線所造成的增益。

$$\begin{bmatrix} a_{11} & \cdots & a_{1n} \\ a_{21} & \cdots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{bmatrix}$$

$$\begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} a_{11} \\ a_{21} \end{bmatrix} = \begin{bmatrix} \sqrt{a_{11}^2 + a_{21}^2} \\ 0 \end{bmatrix}, \cos \theta = \frac{a_{11}}{\sqrt{a_{11}^2 + a_{21}^2}}, \sin \theta = \frac{a_{21}}{\sqrt{a_{11}^2 + a_{21}^2}}$$

$$\begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} a_{1k} \\ a_{2k} \end{bmatrix} = \begin{bmatrix} a'_{1k} \\ a'_{2k} \end{bmatrix}, k=1,2,\dots,n$$

而 CORDIC 則是規定第 i 次旋轉的角度 θ_i 的 $\tan \theta_i = 2^{-i}$ ，並利用迭代的方式來累積成我們所要的目標角度，這樣的好處是我們可以利用 shift 來處理過程中的運算。

第 i 次迭代可以表示為：

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = K_i \begin{bmatrix} 1 & -d_i 2^{-i} \\ d_i 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix}$$

$$K_i = \cos \theta_i, d_i = -\text{sign}(y_i)$$

由於 CORDIC 演算法先天上的限制，最終迭代出的角度範圍是落在 $\pm\pi/2$ ，所以如果原始夾角超過這個範圍，我們就需要先做預處理，將它旋轉 π ，好讓他的夾角落到 $\pm\pi/2$ 的範圍內。

Sorted 的意思則是我們每次再進行 QR 分解前，會先對矩陣 H 作排序，這樣可以讓後面的演算法偵測成功率變高。以 K-Best 為例子，當 K-Best 要對第一行進行 QR 分解前，會先將矩陣 H 中，column norm 最小的那一行移到第一行，並進行 QR 分解，之後更新每一行的 column norm，進行第 i 次 QR 分解時，就會將剩餘行數中 column norm 最小的一行移到第 i 行。而 FSD 的排序則較為特別，以本專題為例，我們總共有八行需要進行 QR 分解，前面六行在排序時，要找出的目標為剩餘行數中，column norm 第三小的那一行，而剩下的兩行則是 column norm 最小的兩行。雖然兩種演算法排序方式不太一樣，但其目的都是為了讓後續偵測時，能夠先偵測出獲得通道增益較大的訊號，換言之，這個訊號解出來是正確的可能性較高，因此先解出來之後，比較不會因為前一個訊號解錯，而去影響到後面偵測其他訊號的結果。

2. Detection Algorithm

首先要介紹一下 QR-based Successive Interference Cancellation (QR-SIC) 以及 PED 的概念。當我們完成 QR 分解後，可以得到 $\hat{y} = Rs + \hat{n}$ ，接著我們會從 s_N 開始往上逐個解出

s 中的元素，即

$$\hat{s}_i = \text{Quantize}\left(\frac{\hat{y} - r_{i+1,i+1:N}\hat{s}_{i+1:N}}{r_{i,i}}\right)$$

如果 $\hat{s}_i$ 解出來與實際 s_i 不同，那它所造成的錯誤將會累積，導致越後面解出的訊號越有可能出現錯誤。PED 某方面象徵著我們目前所累積的誤差大小，PED 計算方式為

$$PED_i = PED_{i+1} + |e_i(\hat{s}^{(i)})|^2$$

$$|e_i(\hat{s}^{(i)})|^2 = |\hat{y}_i - r_{i,i}\hat{s}_i - \sum_{j=i+1}^N r_{i,j}\hat{s}_j|^2$$

我們所使用的 K-Best 及 FSD 主要就是運用到這兩個概念。

K-Best 是以廣度優先搜尋，每次要解出 s_i 時，會找出 K 條可能的路徑，本專題的 $K=4$ ，因此它會再延伸出的 4×4 條路徑中，保留四條路徑，而這四條路徑就是 PED 較小的四條，接著再利用這保留下來的四條路徑繼續解出下一個訊號，到最後最後會選擇四條路徑中 PED 最小的當作解。

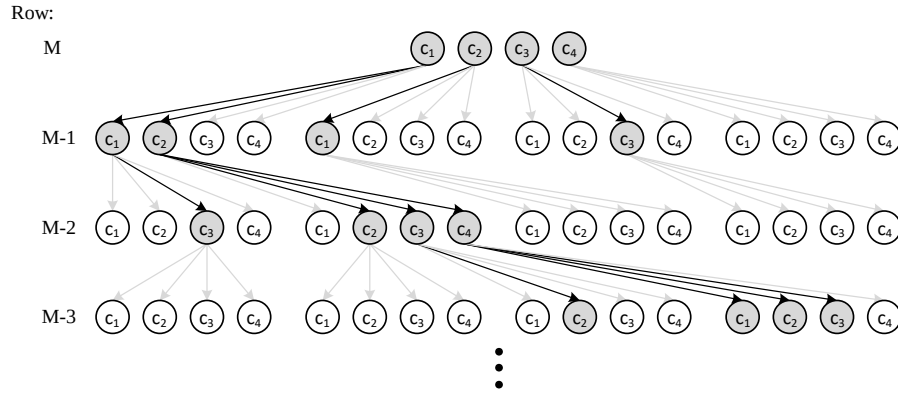


Fig 2. K-Best(K=4) 示意圖

FSD 則是一種廣義的 K-Best，FSD 每一層會有自己的 n_i ，當要解出 s_{i-1} 時，每條 s_i 的路徑都會延伸出 n_{i-1} 條 PED 較小的路徑，本專題 n_i 設定為 (4,4,1,1,1,1,1,1)，因此到最後最後總共會有 $4 \times 4 \times 1 \times 1 \times 1 \times 1 \times 1 \times 1 = 16$ 條路徑，因此最後會從 16 條路徑中，選擇 PED 最小的作為解。

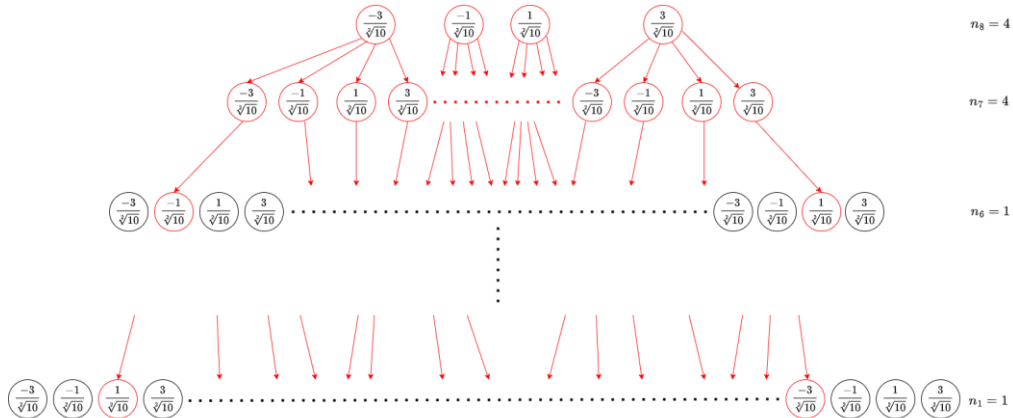


Fig 3. FSD 示意圖

3. Result

本次專題先利用 C++ 模擬，並利用 Maximum Likelihood (ML) 的成果作為本次專題的 Benchmark，完成定點數模擬、硬體架構設計後，撰寫 Verilog 燒入到 FPGA，以下是 FPGA 模擬結果。

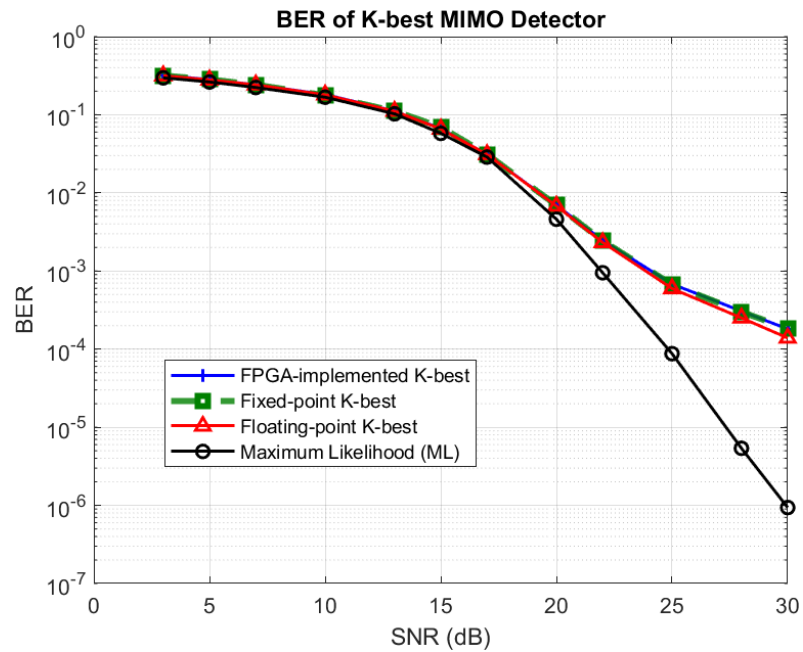


Fig 4(a). K-Best 模擬結果

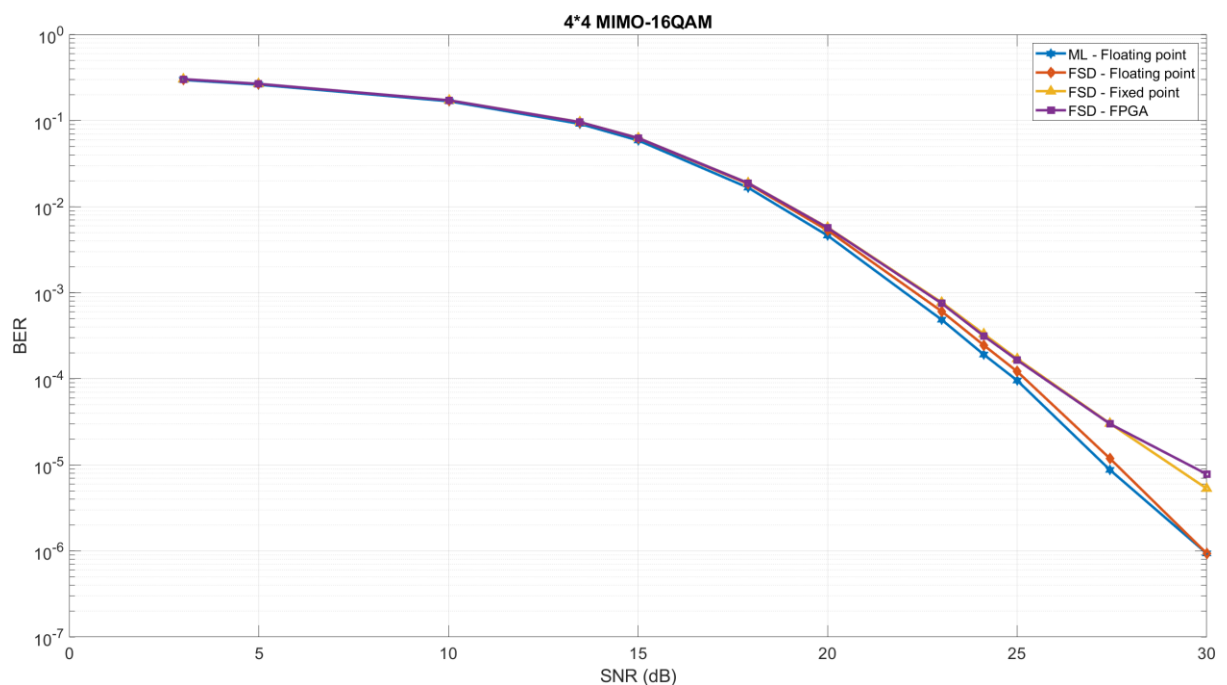


Fig 4(a). FSD 模擬結果

Reference

- [1] 《數位通訊積體電路設計》課程講義。黃元豪。
- [2] Xuezheng Chu, John McAllister and Roger Woods eStreams, "A LOW COMPLEXITY REAL-TIME MIMO-PREPROCESSING FOR FIXED-COMPLEXITY SPHERE DECODER," in Wireless Innovation Forum, 2011.
- [3] R. -H. Lai, C. -M. Chen, P. -A. Ting and Y. -H. Huang, "A modified sorted-QR decomposition algorithm for parallel processing in MIMO detection," 2009 IEEE International Symposium on Circuits and Systems, 2009, pp. 1405-1408, doi: 10.1109/ISCAS.2009.5118028.

心得感想

陳禹嘉:

從二月開始研讀資料與講義，到八月底完成所有的設計，在這幾個月的時間，我學到了許多通訊相關，以及數位電路設計相關的知識，也培養了自己獨立研究、搜尋資料與學術報告的能力。

起初，我和隊友對於通訊相關的知識所知甚少，於是我們透過許多的資源了解現代無線通訊背後的原理，以及訊號傳輸時所用的調變方式，充實我們的背景知識。在研讀教授給的數位通訊晶片設計課程的講義時，有許多不明白的點，我藉由和隊友討論，以及搜尋相關文獻，一一克服，在這過程中，甚至獲得了更多的知識，像是不同的研究者改善解碼正確率的方式，或是下一代的通訊中會用到的技術。不過教授也提醒我們期刊上的研究成果並不一定能實用於現實生活中，因為很多研究人員是不具有工程師背景的，他們考慮的環境可能會過於理想化，而與實際情況不合。這也令我學到之後在做研究時，在做各種前提假設前，應該多考慮是否合乎現實，以免讓研究與現實脫軌。

另外，在與教授 meeting 時，我也發現自己在表達能力上的不足。在做學術研究的報告時，必須內容完整、用詞精確，才能讓聽者了解自己腦中所想表達的內容。我經常講了半天卻難以完整傳達給聽者，後來，我了解到在進行報告前，我應該先以聽者的角度思考，我所報告的內容有哪些是聽者所不清楚的，避免在報告時，跳過太多讓聽者存疑、困惑的部分。另外，也要掌握各個專有名詞的定義，才能更精準以及精簡地描述我們所講的事物，這些都是在與教授報告的過程中，經過教授的提點我才慢慢學會掌握的，目前依然還在培養自己進行專業報告的能力。

最後，經過這幾個月的訓練與努力，我們完成了各自的演算法電路設計，也培養自己一些研究的能力，過程中，感謝隊友和我互相討論，我們一起解決了各種困難，也感謝教授每週親自與我們進行 meeting，提點了許多我們沒做好或是可以做更好的地方，有時，我有些地方做錯方向或是想法錯了，多虧教授細心的講解我才能及時更正，也對於研究內容更加清楚。另外，也要感謝實驗室的學長姊，協助我們使用硬體設備，並順利完成最後的驗證。

黃齊緯:

今年二月開學後，我開始讀教授在碩班所開設課程的課程講義，由於我在修這門專題以前，除了訊號與系統外，沒有修過任何通訊相關的課程，因此一開始在研讀講義時，其實有許多地方我是看不太懂的，常常需要花許多時間上網搜尋相關資料，或是與隊友討論。到了四月初我們讀完這些文獻後，教授要我們決定各自負責的演算法，可以開始進行 C++ 的模擬了，而我所選擇的是 FSD 的部分。C++ 首先要模擬的是 ML 的部分，因為 ML 所得到的結果會是最好的，但它不適合實作在硬體上，所以我們主要是要拿它來當作一個標準，在這個過程中，我們遇到最大的問題是對於矩陣 H 的增益的設定，我們想要將增益設為 1，但一開始一直想不通怎麼設才會是 1，後來經過與隊友討論後，終於克服了這個難關。在五月，我初步完成了 ML 及 FSD 的模擬，但由於 FSD 的結果不如我預期，因此我開始查找論文，發現原因是我沒有使用 Sorted QR Decomposition 的關係，將 SQRD 加入設計後，模擬結果就變得非常漂亮了。

五月底我們開始進行定點數的模擬，這部分要決定小數點的精度，是之前完全沒有接觸過的開發流程，因此一開始有點不知道怎麼下手，好在與隊友討論後，這部分的模擬也順利完成了。到了六月我們開始設計硬體架構，在這個過程中，非常感謝教授點出我們 block diagram 畫得不好的地方，讓我們注意到許多以前沒注意過細節，這部分我們主要是設計大致的架構，而 pipeline 的部分由於還沒有實際測試，因此非有切的非常細。接下來到了暑假，七月的時候我們開始進行 Verilog 的模擬，針對各個 Block 簡單丟幾個測資確認一下功能正確，並且開始測試 Timing，決定真正的 pipeline 應該如何切，之後確認大致上的功能正確後，就可以開始燒入到 FPGA 測試。八月是我們開始實際到實驗室進行 FPGA 的測試，一開始燒入後果然與 C++ 的模擬結果不符，不過將錯誤的側資抓出來檢查後，發現是我們的整數位元數不夠，還好不是邏輯上的錯誤，因此這部分到了八月中也順利解決了，最後我們終於完成 FPGA 的測試，也代表我們專題正式結束了。

非常感謝教授在每個禮拜四的早上抽出時間與我們 meeting，在 meeting 過程中，也學習到了一些報告的技巧，盡量避免在報告時，只是單純將論文或講義進行英文翻譯，而是要理解後加入自己的見解並說出來。也非常感謝暑假期間實驗室學長的幫忙，讓我們能夠順利完成 FPGA 的測試。