

# 國立成功大學 電機工程學系

## 個人專題實作成果報告書

Hardware Design of QRD-MGS based OMP for  
Compressive Sensing Image Reconstruction

基於改良式格拉姆-施密特 QR 分解的正交匹配  
追蹤演算法之硬體設計應用於壓縮感知影像重建

專題組別： ☒ 電路與系統 ☐ 電子與材料 ☐ 電腦與通訊

組員姓名&學號：C14074021 張柏彥

研究期間：111 年 7 月至 111 年 10 月底止

## 摘要

本專題提出一個基於改良式格拉姆-施密特 QR 分解的正交匹配追蹤演算法硬體架構設計，目的是減少傳統正交匹配追蹤演算法的硬體運算複雜度。在向量運算上使用了高度平行化的架構以已達到高速且低功率的成效，除法以及根號運算則以 CORDIC 運算單元取代建構 LUT (Lookup Table) 避免使用大量的暫存器。

最後利用 Synopsys Design Compiler 進行電路合成，且 Gate-level Netlist Simulation 與 Fixed-point Simulation 比較後的結果一致。此專題之感知壓縮系統使用高斯隨機矩陣作為測量矩陣、離散餘弦轉換矩陣作為稀疏矩陣。最後電路使用 TSMC 90 nm 實現，其合成後面積為約為  $894907\mu\text{m}^2$ ，功耗約為 20.8530mW，在工作頻率為 166.67MHz 下，重建一長度為 256，訊號壓縮率 (Compression Ratio) 為 25%，稀疏度為 8 的訊號所需的時間約為  $31.57\mu\text{s}$ 。

**關鍵字：**壓縮感知、正交匹配追蹤、QR 分解、影像重建、硬體設計

## Abstract

This paper proposes a hardware architecture design of QRD-MGS based Orthogonal Matching Pursuit algorithm, which aims to reduce the computational complexity of conventional Orthogonal Matching Pursuit algorithm. In terms of vector computations, a highly parallel architecture has been applied to achieve high speed and low consumption specifications. In addition, to avoid redundant register usage, the division and square root operations are completed by CORDIC arithmetic unit instead of building a LUT (Lookup Tables).

The circuit synthesis is done by Synopsys Design Compiler, and the result of Gate-level Netlist Simulation is consistent with that of Fixed-point Simulation. In the proposed compressive sensing image reconstruction system, the measurement matrix implements a gaussian random matrix and the sparse basis implements a discrete cosine transform matrix. The proposed architecture is implemented by 90nm TSMC CMOS technology. According to pre-layout results, the total cell area is about  $894907\mu\text{m}^2$  and the total consumption is about 20.8530mW. The reconstruction time of signal  $N=256$  with 25% compression ratio and sparsity  $K=8$  is about  $31.57\mu\text{s}$  at the operation frequency of 166.67MHz.

**Keyword :** Compressive Sensing (CS), Orthogonal Match Pursuit (OMP), QR Decomposition, Image Reconstruction, Hardware Design

# 目錄

摘要.....	1
Abstract .....	2
目錄.....	3
一、前言 .....	4
二、原理分析、演算法與系統設計.....	4
2.1 原理分析.....	4
2.1.1 有限等距性質 (Restrict Isometry Property).....	6
2.1.2 低相干性 (Low Coherence).....	6
2.1.3 稀疏度 (Sparsity).....	6
2.2 演算法.....	7
2.2.1 正交匹配追蹤 (Orthogonal Matching Pursuit).....	7
2.2.2 改良式格拉姆-施密特 QR 分解 (QRD-MGS).....	7
2.2.3 反向帶回法 (Back Substitution).....	8
2.2.4 基於 QRD-MGS 之正交匹配追蹤 (QRD-MGS based OMP).....	8
2.2.5 CORDIC 開平方根與除法運算.....	8
2.3 系統設計.....	10
2.3.1 VMU 設計.....	11
2.3.2 BS_Mult 設計.....	12
三、研究方法與實驗結果.....	12
3.1 研究方法.....	12
3.1.1 系統規格 (System Specification).....	12
3.1.2 浮點數運算模擬 (Floating-point Simulation).....	12
3.1.3 定點數運算模擬 (Fixed-point Simulation).....	13
3.1.4 暫存器傳輸級設計 (RTL design).....	15
3.2 實驗結果.....	15
四、結論.....	16
五、參考文獻.....	16

## 一、前言

壓縮感知(Compressed Sensing)是近年來提出的訊號處理理論，是一種在取樣過程中完成了數據壓縮的過程，用於取得和重構稀疏訊號，現已被應用於核磁共振、通道估測、物聯網或是加密系統。有別於傳統奈奎斯特取樣理論(Nyquist Sampling Theorem)中，取樣頻率必須大於等於兩倍訊號最高頻率的限制。壓縮感知理論提出，若信號本身或是該信號在某個正交空間中具有稀疏性，就可以以遠低於奈奎斯特取樣理論所要求的取樣頻率進行隨機取樣，並高機率地能夠重建該訊號。

由於大部分的壓縮感知演算法都具有很高的運算複雜度，以正交匹配追蹤(以下皆以 OMP 簡稱)演算法為例，OMP 演算法主要分為 Find best fitting column, LS optimization 及 Residual update 三個部分。在傳統 OMP 演算法中，每次迭代計算 LS optimization 部分時皆須計算傳感矩陣  $\Phi$  的偽逆矩陣以更新殘差(residual)及計算重建信號，不利於硬體實現。因此許多學者們提出針對演算法層面或架構層面的改良，如：Cholesky based、QRD based 及 MIB based Implementation，以降低硬體實現的複雜度。本專題主要實現 QRD based 的 OMP 演算法硬體設計，並應用在影像重建。

## 二、原理分析、演算法與系統設計

### 2.1 原理分析

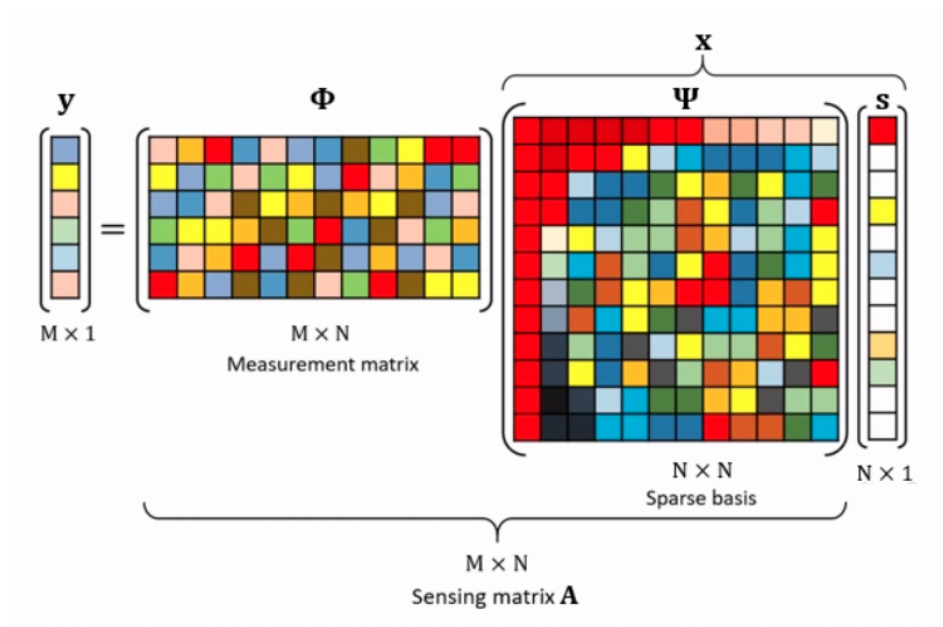
壓縮感知是一種非線性壓縮信號並重建的方法，圖(一)為感知壓縮系統：

$x$  為原始訊號， $\Phi$  為測量矩陣(Measurement Matrix)，通常使用伯努利矩陣(Bernoulli Matrix)或高斯隨機矩陣(Gaussian Random Matrix)， $y$  為隨機觀測的訊號。由於自然界中的訊號大部分並非稀疏訊號(如：聲音、光)，因此選擇  $N$  組稀疏基底(sparse basis)組成稀疏矩陣  $\Psi$ (Sparse Matrix)，將原始訊號投影到某個變換域上得到稀疏訊號，稀疏矩陣為標準化正交矩陣(Orthonormal Matrix)，通常選擇離散餘弦轉換矩陣

(Discrete Cosine Transform Matrix)、離散小波轉換矩陣(Discrete Wavelet Transform Matrix)或離散傅立葉轉換矩陣(Discrete Fourier Transform Matrix)。而傳感矩陣  $\mathbf{A}$  (Sensing Matrix)則為測量矩陣與稀疏矩陣相乘，大小為  $M \times N$  的矩陣，其中  $M \ll N$ 。

由此系統可得，此問題本質上為找到  $\min \|x\|_0$  使得  $y = \Phi * x$ 。由於  $y = \Phi * x$  存在無限多組解，所以此唯一 NP-hard 問題。因此需要透過轉換成 L1 范數，也就是求解  $\min \|x\|_1$  使得  $y = \Phi * x$  的問題，此為訊號重建演算法研究的重要方向之一，目前主要以基追蹤算法(Basis Pursuit)、匹配追蹤算法(Match Pursuit)以及正交匹配追蹤算法(Orthogonal Match Pursuit)為典型的訊號重建演算法代表。

$y$ : Measurement,  $y \in \mathbb{R}^{M \times 1}$   
 $x$ : Original data,  $x \in \mathbb{R}^{N \times 1}$   
 $s$ : Signal in sparse domain,  $s \in \mathbb{R}^{N \times 1}$   
 $\Phi$ : Measurement matrix,  $\Phi \in \mathbb{R}^{M \times N}$   
 $\Psi$ : Sparse basis,  $\Psi \in \mathbb{R}^{N \times N}$   
 $\mathbf{A}$ : Sensing matrix  $\mathbf{A} \in \mathbb{R}^{M \times N}$   
 $M$ : Number of measurement  
 $N$ : Data length,  $M < N$



圖(一) Compressive Sensing Model

### 2.1.1 有限等距性質(Restricted Isometry Property)

滿足有限等距性質代表矩陣的 null space 對於 sparse vector 而言足夠小，即保證該矩陣不會將兩個不同的稀疏信號映射到同一個集合，主要用來分析還原演算法的表現好壞。

$$(1-\delta)\|s\|_2^2 \leq \|As\|_2^2 \leq (1+\delta)\|s\|_2^2$$

若  $0 < \delta < 1$ ，則代表該感知矩陣滿足 RIP 性質。

### 2.1.2 低相干性(Low Coherence)

相干性的定義為任兩相異的行向量做內積後，取絕對值所產生的最大值，以衡量訊號重建的準確度。

$$\mu(\Phi) = \max_{i \neq j} |\phi_i^T \phi_j|$$

而為了確保訊號重建的準確度，所採用的測量矩陣各行列之間的相干性需盡量的低。

### 2.1.3 稀疏度(Sparsity)

若使用之感知矩陣滿足 RIP 性質且測量矩陣各行列之間具有低相干性，當訊號  $x$  的非零數目滿足下列關係：

$$M \geq C * K \log\left(\frac{N}{K}\right)$$

其中  $C$  為一個根據情況不同而改變的常數，則訊號有高機率被能夠被完美還原。

## 2.2 演算法

### 2.2.1 正交匹配追蹤 (Orthogonal Matching Pursuit)

正交匹配追蹤演算法為一貪婪重構演算法，即給定傳感矩陣  $\mathbf{A}$ 、取樣訊號  $y$  及稀疏度  $K$  後，最後得到重建稀疏訊號  $s$ 。

---

#### Algorithm 1 Orthogonal Matching Pursuit (OMP)

---

**Input:** Sensing matrix  $\mathbf{A}$  ; Measurement  $y$  ; Sparsity  $K$   
**Output:** Reconstruction sparse signal  $s$

- 1:  $r_0 = y, \Gamma_0 = \emptyset, k = 1$
- 2: **repeat**
- 3:    $\lambda_k = \operatorname{argmax}_j |\langle a_j, r_{k-1} \rangle|$
- 4:    $\Gamma_k = \Gamma_{k-1} \cup \lambda_k$
- 5:    $s_k = \operatorname{argmin}_s \|\mathbf{A}_{\Gamma_k} s - y\|_2^2$
- 6:    $r_k = y - \mathbf{A}_{\Gamma_k} s_k$
- 7: **until** ( $k > K$  or  $\operatorname{norm}(r_k) < \epsilon$ )

---

### 2.2.2 改良式格拉姆-施密特 QR 分解 (QRD-MGS)

由於在計算最小平方時，每次迭代需計算  $\mathbf{A}_{\Gamma_k}$  的偽逆矩陣(Alg.1 step 5)，硬體複雜度相當高，因此使用Modified Gram-Schmidt algorithm對  $\mathbf{A}_{\Gamma_k}$  進行QR分解，其中  $\mathbf{R}$  為上三角矩陣， $\mathbf{Q}$  與  $\mathbf{R}$  矩陣的大小會隨著迭代次數逐漸增加且與上一次迭代的結果有關，因此降低了運算的複雜度，然而需要足夠的記憶體存取  $\mathbf{Q}$  與  $\mathbf{R}$  矩陣的值。

---

#### Algorithm 2 Incremental QRD by Modified Gram-Schmidt

---

**Input:** New column  $\mathbf{A}_{\lambda_k}$  ; Last iteration's  $\mathbf{Q}^{k-1}, \mathbf{R}^{k-1}$  ; Iteration counter  $k$   
**Output:**  $\mathbf{Q}^k, \mathbf{R}^k$

- 1:  $\mathbf{R}^k = \begin{bmatrix} \mathbf{R}^{k-1} & 0 \\ 0 & 0 \end{bmatrix}, \gamma^k = \mathbf{A}_{\lambda_k}, i = 1$
- 2: **repeat**
- 3:    $\mathbf{R}_{ik}^k = (\mathbf{Q}^{k-1})^T \gamma^k$
- 4:    $\gamma^k = \gamma^k - \mathbf{R}_{ik}^k \mathbf{Q}_i^{k-1}$
- 5: **until** ( $i > k$ )
- 6:  $\mathbf{R}_{kk}^k = \sqrt{\|\gamma^k\|_2^2}$
- 7:  $\mathbf{Q}^k = \frac{\mathbf{Q}^{k-1} \gamma^k}{\sqrt{\|\gamma^k\|_2^2}}$

---



### 2.2.3 反向代回法 (Back Substitution)

在求得重建訊號 $s$ 時需計算 $\mathbf{R}^{-1}$ ，由於 $\mathbf{R}$ 為上三角矩陣，因此可以透過Back Substitution的方式求得 $\mathbf{R}^{-1}$ 。其演算法表示如下：

---

**Algorithm 3** Back Substitution

---

**Input:** Upper triangular matrix  $\mathbf{R}$  ; Size of the matrix  $N$

**Output:** Inverse of upper triangular matrix  $\mathbf{R}^{-1}$

```
1:  $\mathbf{R}^{-1} = \mathbf{O}_N$ 
2: for  $i = N, \dots, 1$  do
3:   for  $j = N, \dots, 1$  do
4:      $\mathbf{R}_{ij}^{-1} = [\mathbf{I}_{ij} - (\mathbf{R}^T)_i \mathbf{R}_j^{-1}] / \mathbf{R}_{ii}$ 
5:   end for
6: end for
```

---

### 2.2.4 基於 QRD-MGS 之正交匹配追蹤 (QRD-MGS based OMP)

由 Alg.2 及 Alg.3，可將 OMP 演算法中更新殘差的部分(Alg.1 step 5)取代成， $\mathbf{r}_k = \mathbf{r}_{k-1} - \mathbf{Q}_k^* (\mathbf{Q}_k)^T \mathbf{r}_{k-1}$ ，而求得重建訊號  $s$  的部分(Alg.1 step 6)則取代成， $s = \mathbf{R}^{-1} * \mathbf{Q}^T * y$ 。

---

**Algorithm 4** QRD-MGS based OMP

---

**Input:** Sensing matrix  $\mathbf{A}$  ; Measurement  $y$  ; Sparsity  $K$

**Output:** Reconstruction sparse signal  $s$

```
1:  $\mathbf{r}_0 = y, \Gamma_0 = \emptyset, k = 1$ 
2: repeat
3:    $\lambda_k = \text{argmax}_j |\langle a_j, \mathbf{r}_{k-1} \rangle|$ 
4:    $\Gamma_k = \Gamma_{k-1} \cup \lambda_k$ 
5:   Apply QRD-MGS on  $\mathbf{A}_{\Gamma_k} = \mathbf{Q}_k \mathbf{R}_k$ 
6:    $\mathbf{r}_k = \mathbf{r}_{k-1} - \mathbf{Q}_k (\mathbf{Q}_k)^T \mathbf{r}_{k-1}$ 
7: until ( $k > K$  or  $\text{norm}(\mathbf{r}_k) < \epsilon$ )
8:  $s = \mathbf{R}^{-1} \mathbf{Q}^T y$ 
```

---

### 2.2.5 CORDIC 開平方根與除法運算

CORDIC 是 Coordinate Rotation Digital Computer 的簡稱，主要用於解決三角函數、反三角函數、平方根與乘除法等運算問題。CORDIC 迭代運算可使用在圖

周系統、線性系統以及雙曲系統，並分成旋轉模式與向量模式，提供了一種數學計算的逼近方法，前述所提之複雜運算分解為一系列的加減和移位運算。

開平方根運算：假設輸入為  $a$ ，在雙曲系統的向量模式下，令  $x=a+1, y=a-1$ ，經過 9 次 CORDIC 迭代運算後，再將  $x*0.60374807$  即可為逼近後的根號  $a$ ，其演算法如下：

---

**Algorithm 5** CORDIC based square root operation

---

**Input:** Input value  $a$  ; Iteration counter  $N$

**Output:** Output value  $z$

```

1:  $x_0 = a + 1, y_0 = a - 1$ 
2: for  $i = 1, \dots, N$  do
3:   if  $y < 0$  then
4:      $x_i = x_{i-1} + y_{i-1} \ll i$ 
5:      $y_i = y_{i-1} + x_{i-1} \ll i$ 
6:   else  $y \geq 0$  do
7:      $x_i = x_{i-1} - y_{i-1} \ll i$ 
8:      $y_i = y_{i-1} - x_{i-1} \ll i$ 
9:   end if
10: end for
11:  $z = x_N * 0.60374807$ 

```

---

除法運算：假設輸入為  $x, y$ ，在線性系統的向量模式下，經過 15 次 CORDIC 迭代運算後，即可獲得逼近後之  $y/x$ ，其演算法如下：

---

**Algorithm 6** CORDIC based division

---

**Input:** Denominator  $x$  ; Numerator  $y$  ; Iteration counter  $N$

**Output:** Output value  $z$

```

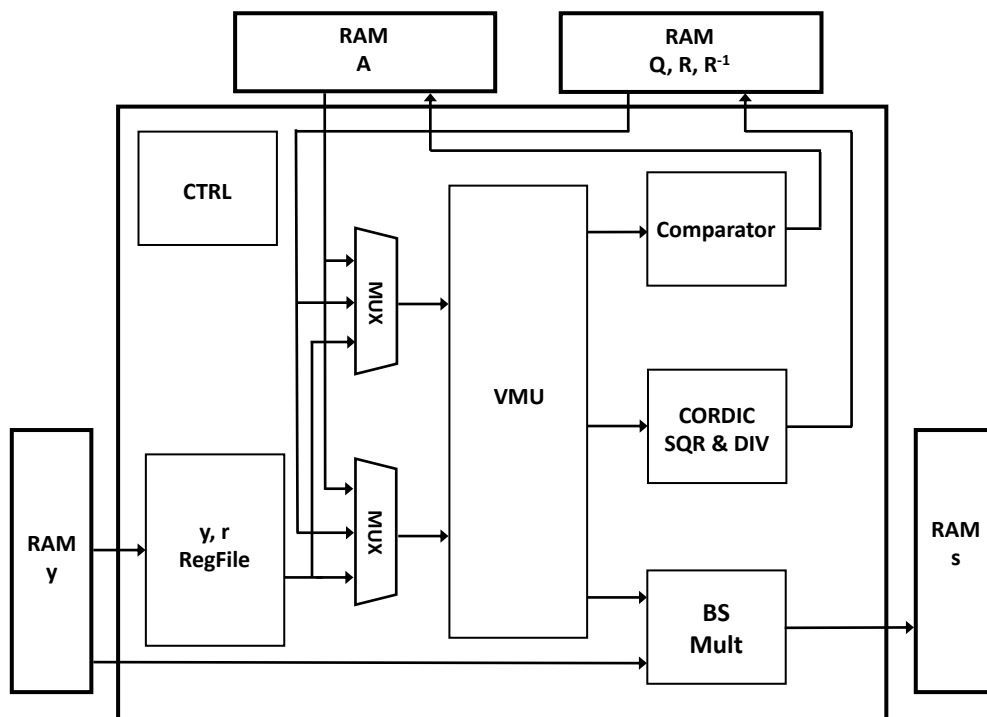
1:  $x_0 = x, y_0 = y, z_0 = 1$ 
2: for  $i = 1, \dots, N$  do
3:   if  $y < 0$  then
4:      $x_i = x_{i-1}$ 
5:      $y_i = y_{i-1} + x_{i-1} \ll i$ 
6:      $z_i = z_{i-1} - 1 \ll i$ 
7:   else  $y \geq 0$  do
8:      $x_i = x_{i-1}$ 
9:      $y_i = y_{i-1} - x_{i-1} \ll i$ 
10:     $z_i = z_{i-1} + 1 \ll i$ 
11:   end if
12: end for

```

---

## 2.3 系統設計

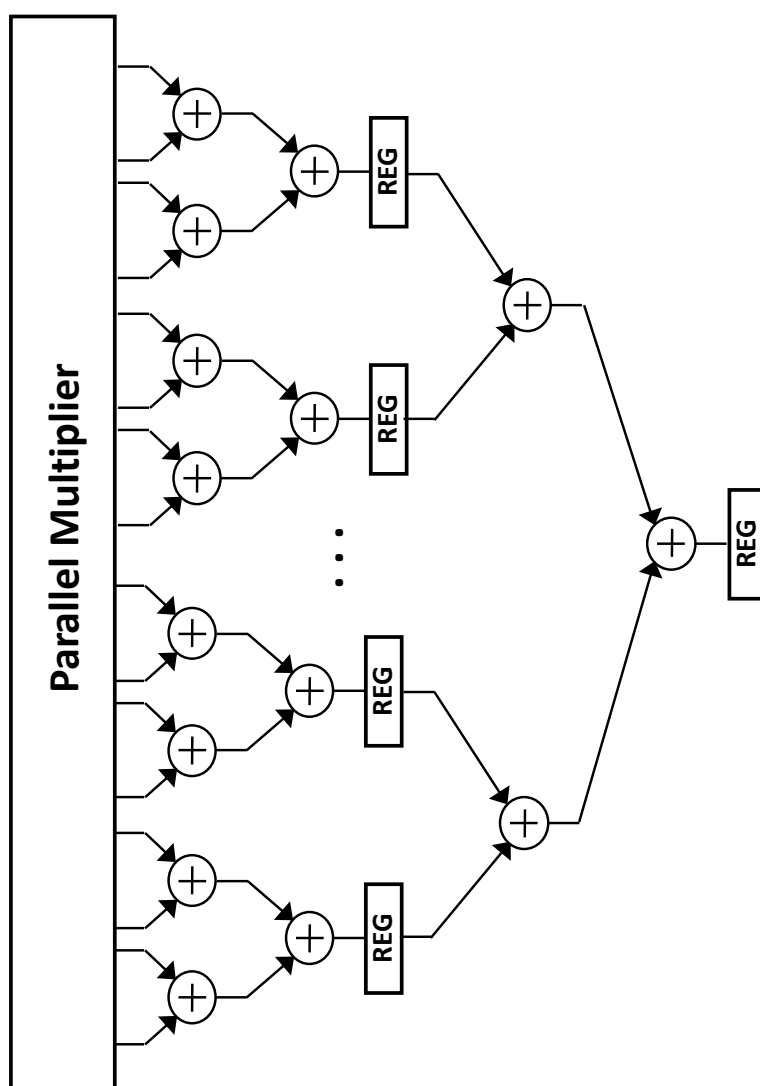
QRD-MGS based OMP 的硬體架構如圖(二)，y,r Regfile 暫存從 RAM y 裡讀取的資料以及暫存更新後的殘差向量，並且能平行地傳遞資料到 VMU 進行運算。VMU 主要處理向量運算，分成三種模式以完成演算法中所需的計算，分別為向量內積、純量乘向量以及向量減法，其中向量內積主要用於計算感知矩陣的行向量與殘差向量的內積、用於 QRD\_MGS 演算法中求得  $\mathbf{Q}$ ,  $\mathbf{R}$ 、Back Substitution 中求得  $\mathbf{R}^{-1}$  的計算以及 BS\_Mult 計算重構訊號  $\mathbf{s}$ ；純量乘向量運算主要用於更新殘差向量的計算以及用於 QRD\_MGS 演算法中求得  $\mathbf{Q}$ ,  $\mathbf{R}$ 。向量的減法運算主要用於計算殘差向量的更新。CORDIC SQR & DIV 主要用於開平方根及除法運算，目的是避免使用 LUT 或是 DesignWare Library 中的 IP，以此減少硬體成本。Comparator 主要用於求得感知矩陣的行向量與殘差向量內積最大絕對值的索引，並且將索引傳遞給 RAM A 以輸出對應索引的行向量。RAM  $\mathbf{Q}$ ,  $\mathbf{R}$ ,  $\mathbf{R}^{-1}$  分別暫存計算得到的  $\mathbf{Q}$ ,  $\mathbf{R}$  以及  $\mathbf{R}^{-1}$  矩陣，且三者皆平行地輸出資料。CTRL 主要利用 Finite State Machine 控制各 Module 的運作以及 RAM、Regfile 讀取與寫入的位址及資料。



圖(二) Architecture of QRD-MGS based OMP

### 2.3.1 VMU 設計

此 VMU 能夠進行向量內積、純量乘向量以及向量減法三種運算，並使用 result\_sel 訊號控制輸出訊號的輸出運算結果或是 0。在 VMU 內部設計的部分，因為輸入向量長度為 64，使用 64 個 20bits\*20bits 平行的乘法器，再使用 two-stage pipeline adder tree 將 64 個乘績相加。其中 adder tree 第一級的 32 個 20bits 加法器與額外使用的 32 個加法器則執行向量減法運算。此 VMU 一共使用 64 個乘法器，63 個加法器以及 32 個減法器。VMU 中向量內積運算之硬體架構如圖(三)所示。



圖(三) Architecture of VMU for inner product operation

### 2.3.2 BS\_Mult 設計

在 BS\_MULT 內部設計的部分，由一個乘法器與一個累加器組成，並使用 bs\_clear\_en 訊號控制累加器是否歸零。主要用於 QRD-MGS based OMP 演算法中，將 VMU 計算  $\mathbf{R}^{-1}$  與  $\mathbf{Q}^T$  向量相乘的結果與  $y$  進行運算，經過累加後得到重建訊號  $s$  並輸出至 RAM  $s$ 。

## 三、研究方法與實驗結果

### 3.1 研究方法

#### 3.1.1 系統規格 (System Specification)

本專題使用 Matlab 實作一壓縮感知影像重建系統，設定壓縮率為 25%，輸入為 256\*256 的圖像，將圖像分成 256 個 16\*16 的 block。建構一 16\*16 的離散餘弦轉換矩陣作為稀疏矩陣，接著將 16\*16 的 block 經由稀疏矩陣投影到稀疏空間，即對 image block 進行二維離散餘弦轉換。接著將轉換後的 16\*16 image block 拉長成長度為 256 的原始訊號向量  $x$ ，並建構一大小為 64\*64，變異數為 1 的高斯隨機矩陣作為測量矩陣，最後將  $x$  進行取樣得到  $y$ ，即  $y = \Phi * x$ 。

將取樣訊號經過重建演算法將訊號重建後，將重構訊號向量還原成 16\*16 的 image block，重複執行 256 次將所有 image block 還原後，比對重建影像與原始影像並計算 PSNR。


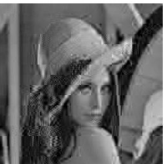








#### 3.1.2 浮點數運算模擬 (Floating-point Simulation)

首先用 Matlab 實作 OMP 演算法，應用於上述實作的壓縮感知影像重建系統，並以 PSNR 作為影像還原的指標，驗證重建訊號的準確率。其中 PSNR 的計算如下：

$$MSE = \frac{1}{mn} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [x(i, j) - \hat{x}(i, j)]^2$$

$$PSNR = 20 * \log\left(\frac{255}{\sqrt{MSE}}\right)$$

接著實作 QRD-MGS 以及 Back Substitution 演算法以實作 QRD-MGS based OMP 演算法，並同樣以 PSNR 作為指標，計算在不同壓縮率的情況下 PSNR 的變化情形。本專題使用的測試圖片分別為 Lena、Peppers 及 Camera Man 且設定稀疏度為 8。

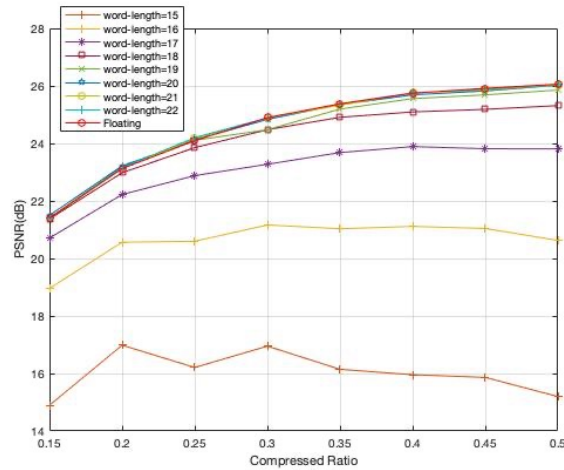
	Original Image	Compression Ratio = 0.15	Compression Ratio = 0.25	Compression Ratio = 0.35	Compression Ratio = 0.45
Lena		 PSNR = 21.2773 dB	 PSNR = 24.0803 dB	 PSNR = 25.4108 dB	 PSNR = 25.9984 dB
Peppers		 PSNR = 20.0322 dB	 PSNR = 22.8057 dB	 PSNR = 24.2396 dB	 PSNR = 24.6934 dB
Camera Man		 PSNR = 17.7291 dB	 PSNR = 19.6544 dB	 PSNR = 20.7444 dB	 PSNR = 21.5254 dB

圖(四) Comparison of PSNR of reconstruct images with different compression ratio and sparsity K=8

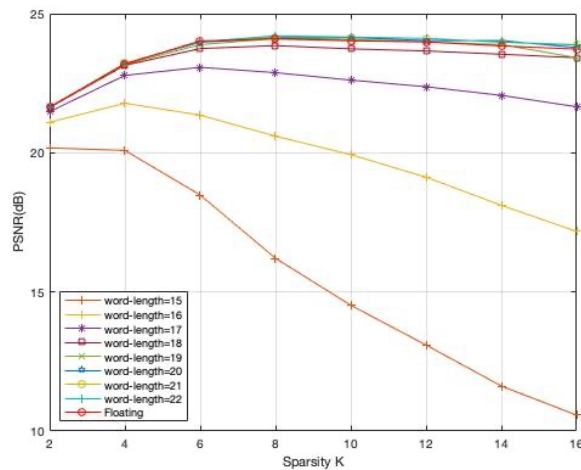
### 3.1.3 定點數運算模擬 (Fixed-point Simulation)

使用 C++ 實作上述之壓縮感知影像重建系統及 QRD-MGS based OMP 演算法，並建構 gFix 資料型態以自行定義定點數的四則運算。首先決定整數部分的長度，若資料經過四則運算後造成溢位(overflow)，則增加 1bit 的長度，反覆此步驟直到不會產生溢位，本專題設定整數部分長度為 4。接著決定整個資料的 word-length 長度，word-length 長度越長則精確度越高，但相對需付出的硬體成本也越高，因此本專題從 word-length=15 開始模擬重建訊號的準確率，並逐次增加 1bit 的小數

部分，即增加 1bit 的精確度，直到與浮點數運算模擬接近吻合。圖(五)與圖(六)分別顯示，在稀疏度為 8 時不同壓縮率與 word-length 在 PSNR 上變化及在壓縮率為 25% 時不同稀疏度與 word-length 在 PSNR 上變化，並與浮點數運算模擬結果比較。本專題選擇兩圖中與浮點數運算模擬結果接近的最小 word-length，即選擇 word-length=19，其中整數部分為 4bits(含 sign bit)，小數部分為 15bits。



圖(五) Reconstruction accuracy in terms of compression ratio with different word-length and sparsity K=8



圖(六) Reconstruction accuracy in terms of sparsity K with different word-length and compression ratio=25%

### 3.1.4 暫存器傳輸級設計 (RTL design)

本專題使用的硬體描述語言為 Verilog，主要以硬體實現演算法中各個運算模組，並使用有限狀態機(Finite State Machine)控制時序的運作。所設計之電路除了其 behavior 需要與 C++的模擬相同之外，電路的運算功能也需要與定點數運算模擬一致。

在驗證電路功能與定點數運算模擬一致後，接著對電路優化。在時序上，透過timing report的分析發現critical path主要發生在VMU的向量內積運算，因此對整個資料流經的路徑切了3級的pipeline，以提高整體電路的最高工作頻率。在合成面積方面，除了演算法中所有的向量運算都由VMU運算之外，也將CORDIC SQT與CORDIC DIV運算合併成一個CORDIC processor運算，大幅提高整體電路的硬體資源重複使用。

## 3.2 實驗結果

利用Synopsys Design Compiler合成電路並執行Gate-level Netlist Simulation後，電路整體的性能分析如圖(七)所示。在設定原始訊號長度為256，取樣訊號長度為64，稀疏度為8的條件下，此電路使用TSMC 90nm CMOS製程技術進行合成，且電路之最高操作頻率為166.67MHz，時脈週期為6.0ns，合成面積為894907.550553  $\mu\text{m}^2$ ，整體功耗為20.8530mW，訊號重建所需時間為31.575736 $\mu\text{s}$ 。

Hardware Specification	N=256, M=64, K=8 TSMC 90nm CMOS technology
Max. frequency (MHz)	166.67
Clock period (ns)	6.0
Total cell area ( $\mu\text{m}^2$ )	894907.550553
Power (mW)	20.8530
Reconstruction time ( $\mu\text{s}$ )	31.575736

圖(七) Hardware Characteristics of proposed OMP implementation



## 四、結論

在這個專題中，主要閱讀了多篇論文，了解壓縮感知的原理與OMP演算法，且透過改良式格拉姆-施密特QR分解與反向代回法降低傳統OMP演算法的複雜度，得到基於改良式格拉姆-施密特QR分解之正交匹配追蹤演算法(QRD-MGS based OMP)，並使用Matlab及C++建構壓縮感知影像重建系統並用PSNR當作影像還原的指標以驗證QRD-MGS based OMP演算法重建訊號的準確度。接著執行Fixed-point Simulation以決定資料的word-length的以及小數部分的大小，並撰寫RTL程式碼，最後利用Synopsys Design Compiler合成電路並執行Gate-level Netlist Simulation，且模擬結果與Fixed-point Simulation的結果一致。

## 五、參考文獻

- [1] D. L. Donoho, “Compressed sensing”, IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289-1306, April 2006.
- [2] L. Bai, P. Maechler, M. Muehlberghuber, and H. Kaeslin, “High-speed compressed sensing reconstruction on fpga using omp and amp”, in 19th IEEE International Conference on Electronics, Circuits, and Systems (ICECS 2012), Dec 2012, pp.53-56.
- [3] Yoshikawa, K., Iwanaga, N., Yamawaki, A., “Development of Fixed-point Square root Operation for High-level Synthesis”, In: 2<sup>nd</sup> International Conference on Industrial Application Engineering, China, pp. 16-20(2014)