**Team03 Final Project Report – QR Engine**

1. **Algorithm:**

* **QR decomposition algorithm**

**主要使用講義上Modified Gram-Schmidt演算法**

1. Calculate Euclidean distance: 首先，算出一個四維向量的8項系數的平方和，接著將平方和放進design ware的sqrt\_pipe，利用4個cycle去計算並得出開根號的結果。
2. Calculate normalized orthogonal vector: 將上一步得出的結果丟進Div\_LUT，利用look up table查出近似的倒數結果以及算出要shift的位數。

倒數設置:

首先根據傳進來的除數，找出除數的leading 1，leading 1的位數即為要額外shift的位數。

接下來我們根據leading 1 後面的七位數字，找出倒數並乘上，此處我們設置N = 22，讓得出的 在16 bit 內

範圍:

.

.

.

得出倒數後，將x乘上 再除上(即為shift) 並往左shift剛剛數出來leading 1前的bit數即可得到正確的 。

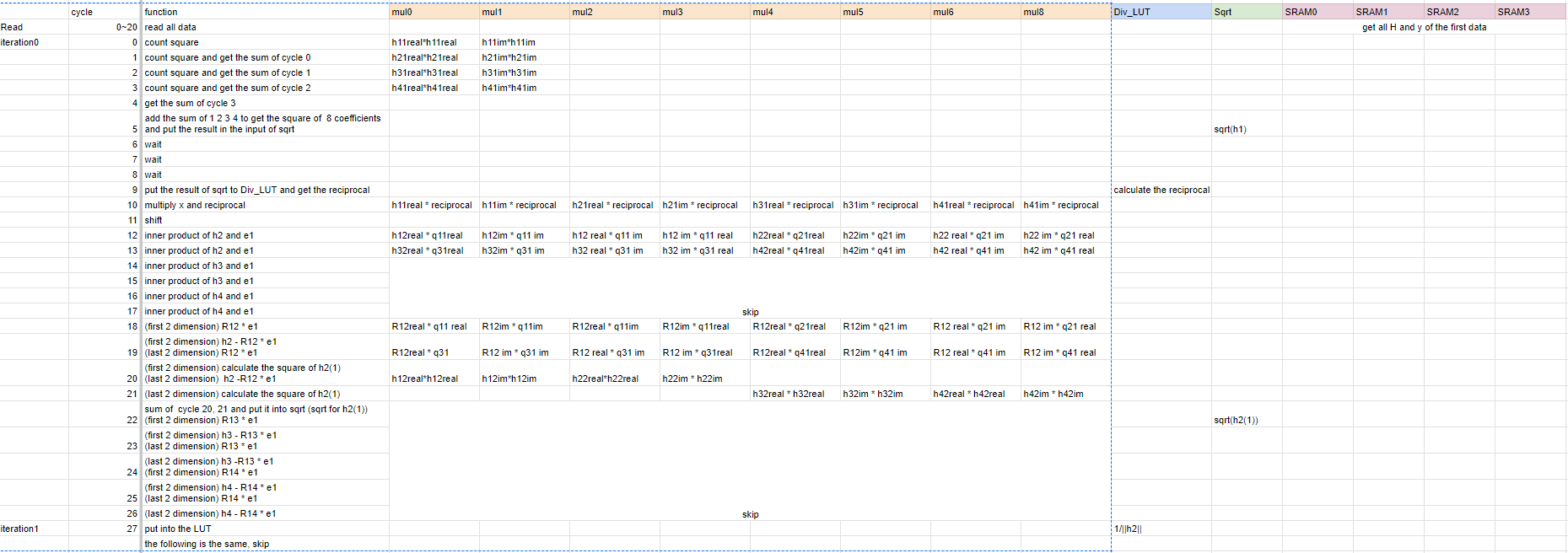
1. Calculate inner products:我們利用4個乘法器算出ac, bd, ad, bc並加起來 ()
2. Calculate orthogonal vectors by removing projection: 同樣的利用4個乘法器算出ac, bd, ad, bc )，並將原本的向量減掉所算出的實部及虛部
3. 對下一個向量重複上述1~4步驟，針對每一組資料的第四個向量不需要做第三步與第四步。

* **FXP setting**
  + 1. 輸入的data只留下32 bit ，實部虛部各16 bit。
    2. 每個乘法器為16 bit 乘上 16 bit (S1.14)，得出來的結果為32 bit (S3.28)
    3. 輸入Div\_LUT的除數為2.14 (無sign bit)，所得出的倒數為16 bit

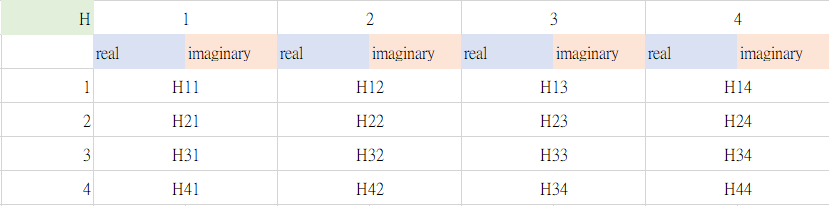
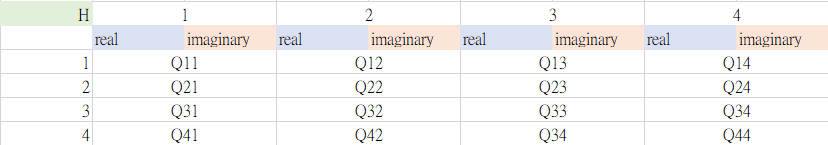
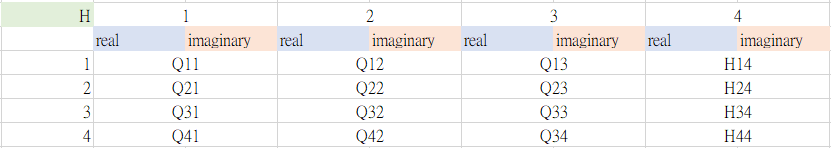
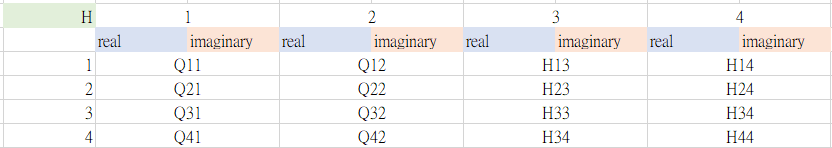
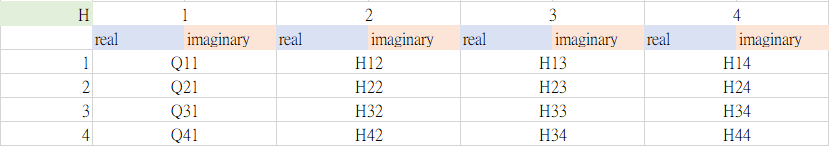
1. **HW implementation:**

* **HW scheduling**

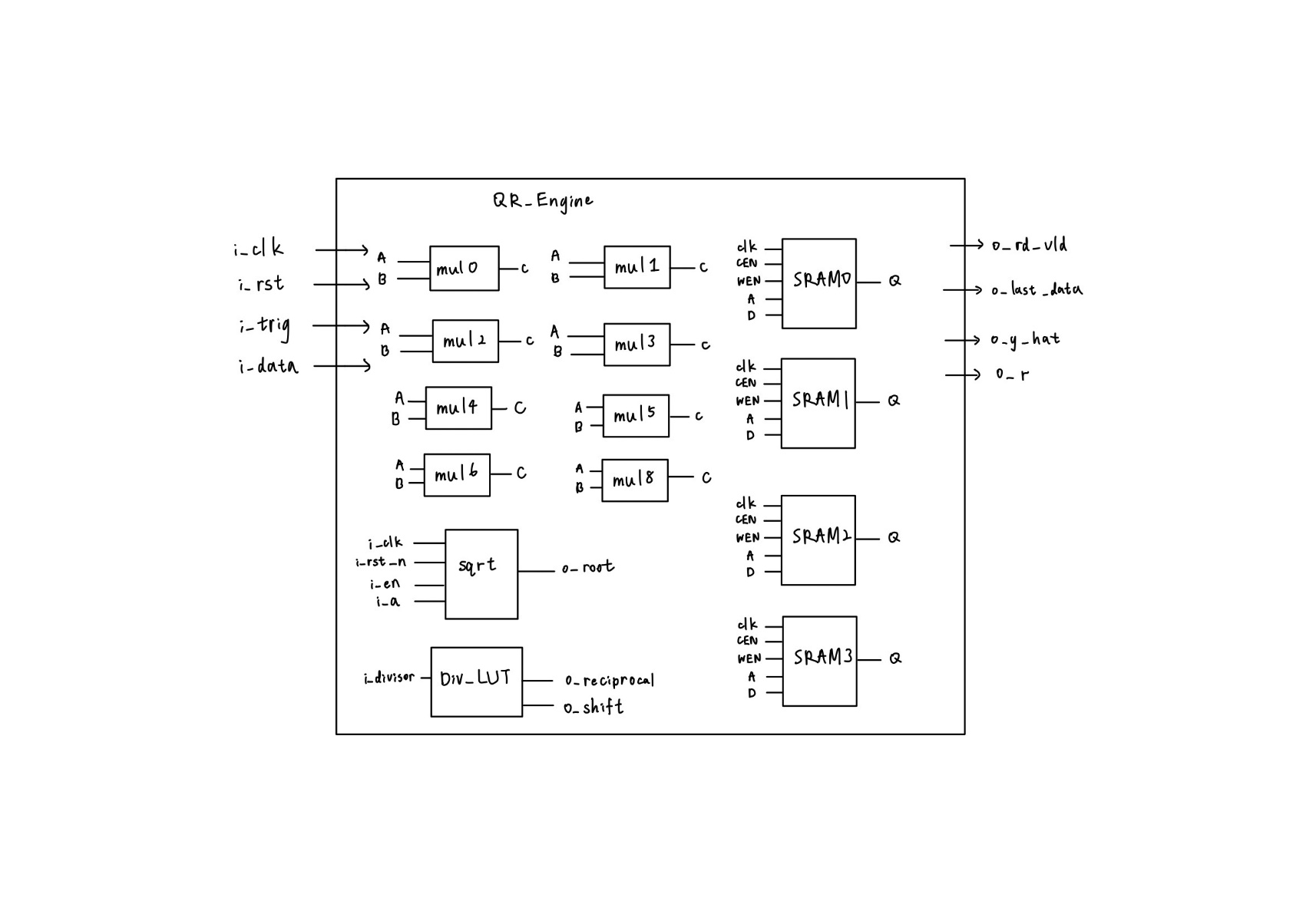
以下圖示表示iteration1中每個cycle要做的事，iteration2、3、4也是相同的概念去做scheduling的。



* **Register sharing**

****

* **HW block diagram**



* **Area / Power / Latency report**
  + Technique sharing for HW improvement

Area: 共用乘法器和除法用LUT做

Power: clock gating

Latency: 平行2組複數乘法