**VLSI\_DSP HW\_2**

1. **Patten 產生**

**QR分解需要滿足滿秩的條件，8\*4矩陣做QR分解rank必須等於4。**

**利用matlab提供的函式rank(A)，驗證輸入的矩陣是否可以被QR分解。**

**本次作業所使用的輸入矩陣為:**

**Input matrix**

1. **matlab模擬**

**硬體規格預計使用八位小數，因此做matlab模擬時會先將input matrix\*2^8，做小數後八位的保留**

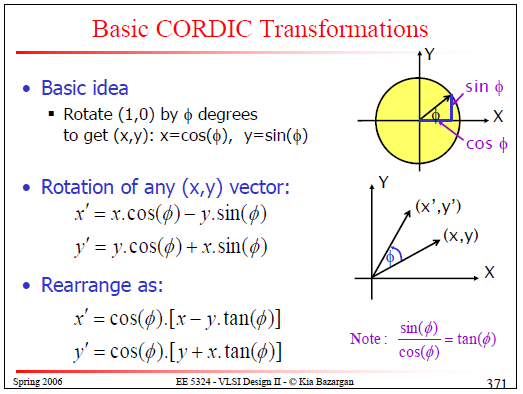
**Input matrix\*2^8 output matrix**

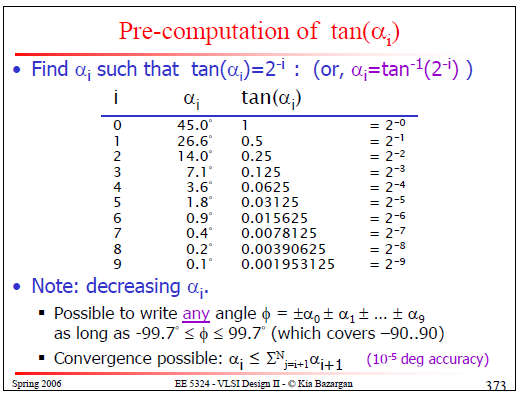
1. **硬體架構**

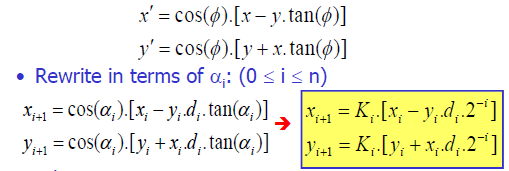
**利用cordic演算法中的Given Generation mode 與Given Rotation mode來解QR分解的問題，cordic演算法利用迭代來求出旋轉角度的近似值，求得QR分解的近似解。**

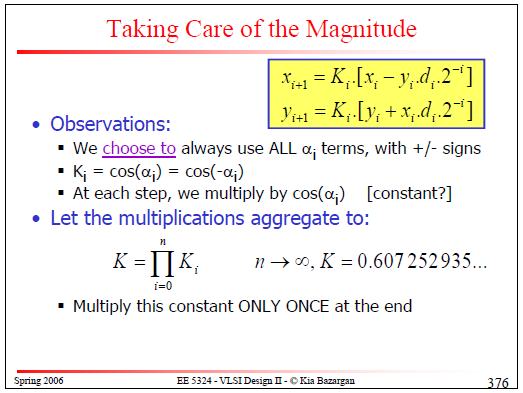
**Cordic演算法簡述**

**為了要找到上三角的R矩陣，需要將上三角矩陣下方的元素消為0，而其中一種方式是將上下相鄰的元素視為一個向量，**

**將這個向量旋轉一個角度後，使得該向量與X軸重疊，即可將下方的元素(y)消為0。**

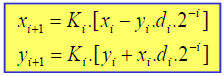
**其中利用迭代法來尋找旋轉角度ψ，每一次迭代都有固定旋轉的角度，而為了方便硬體設計，將tan()設計成的形式，便可以利用shifter來等效乘上tan()的效果。**

**而因為每次迭代的旋轉角度都是已知的，因此cos()的數值在每一次迭代中都是確定的，因此可以將其視為一個定值。**

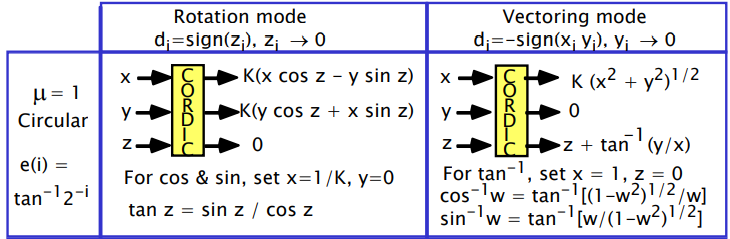
 **cos()= cos()，因此直接將連加到n→∞，並且在最後一次迭代時乘上K就可以完整的表示一次旋轉。**

**其中代表旋轉的方向，若向量在第一象限則順時針旋轉，在第四象限則逆時針旋轉。**

**因為矩陣運算中第一行中的元素發生改變，與變動元素同一列的其他元素也要進行相同的運算，而在cordic演算法中，第一行的運算稱為GG(Given Generation mode)，要利用向量的位置判斷需要正轉或反轉，其他行的運算則稱為GR(Given Rotation mode)，接收GG所計算出來的旋轉方向，並與GG做一樣的旋轉運算**

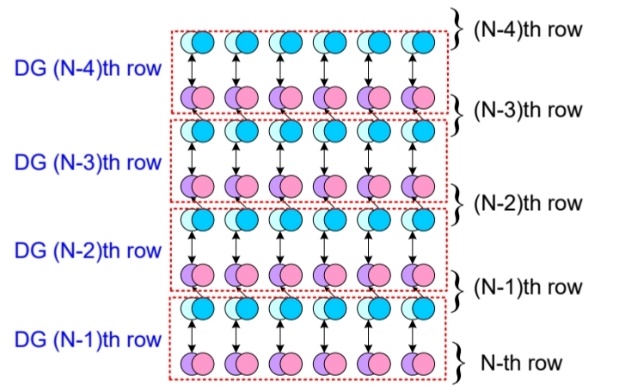
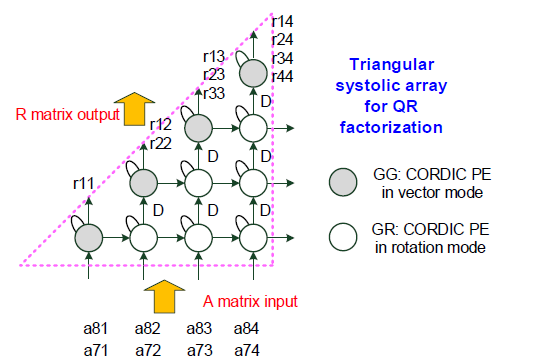


**其中GG、GR的迭代如下圖所示，x、y均為輸入並且得到相對應的輸出，z則為旋轉量，值得注意的是GG透過初值sign的相乘來輸出旋轉向量d。**

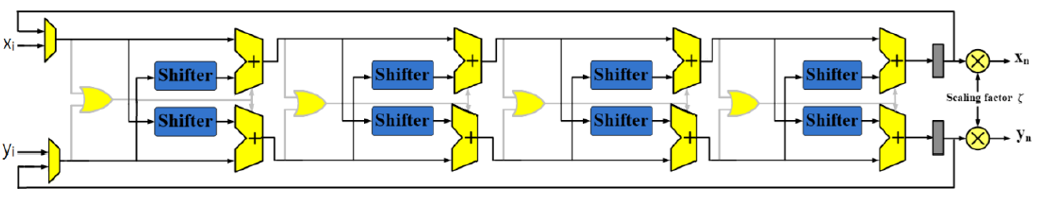


**設計出Given Generation mode與Given Rotation mode的硬體，作為DG map上的兩種PE。**

**DG mapping DFG mapping**

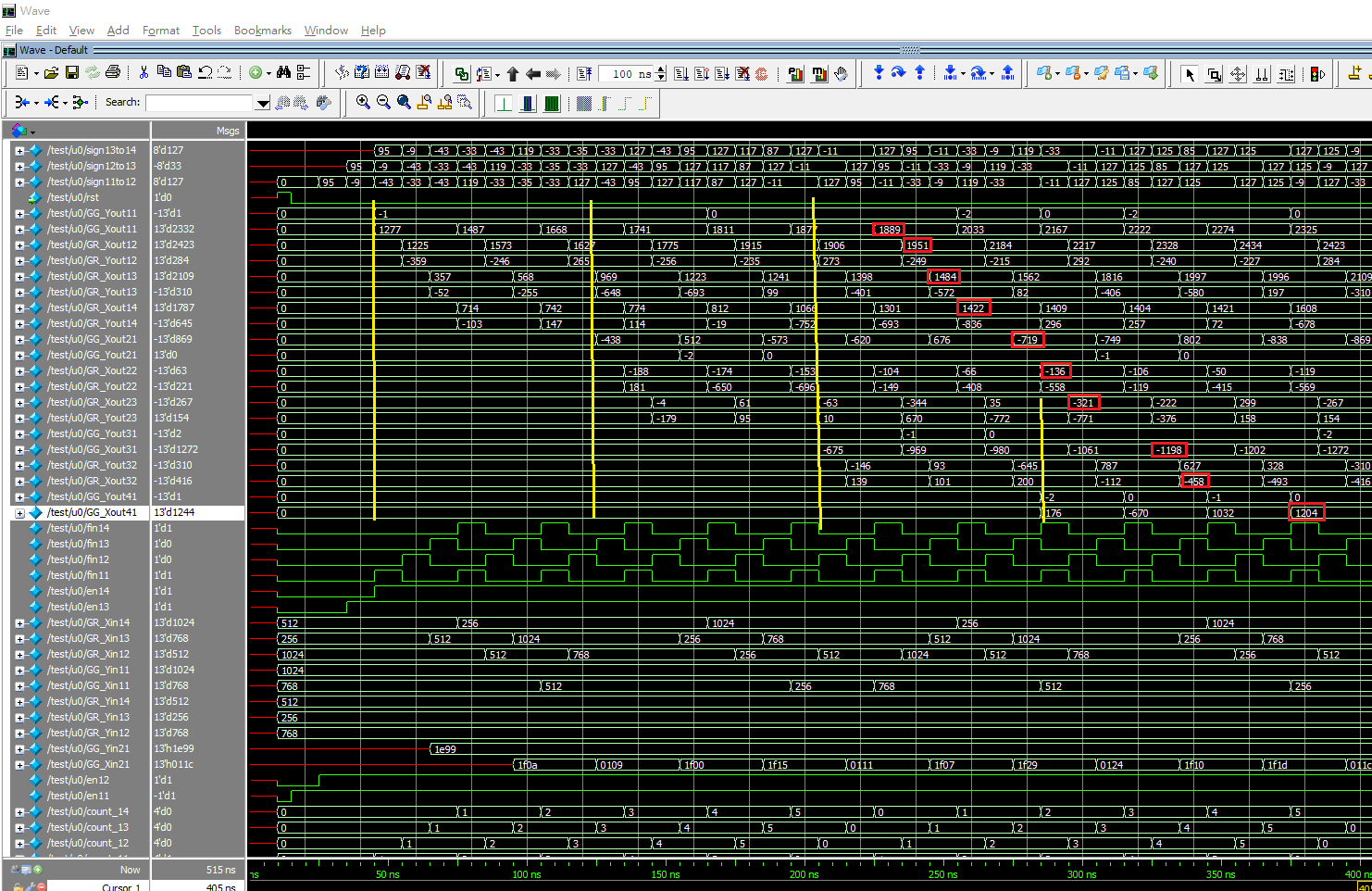
****

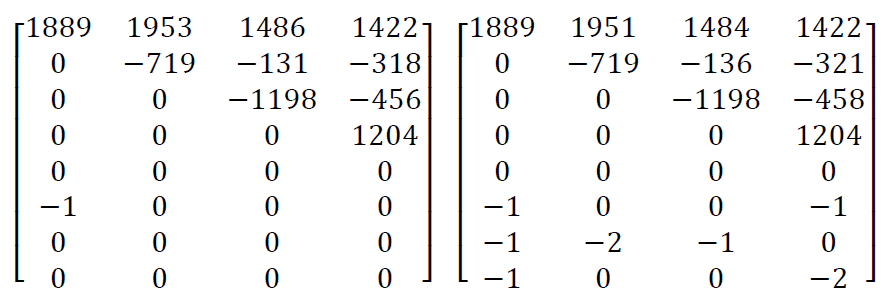
**將DG map投影到DFG上可以得到右邊的圖，作為硬體的主要架構。**

**此次作業中，每個PE的迭代次數設定為12次，並且將PE內部的運算硬體folded成4組運算單元，因此可以在3個clock完成一個PE的運算。**

1. **模擬結果**

**電路設計完成的modelsim模擬結果如下圖**

**紅色框選部分為8\*4上三角矩陣的輸出結果，輸出的時序同一列PE都間隔一個clock，而下一列的GG則是在上一列第一個GR輸出兩個值之後開始動作，輸出的值會如上圖所示呈現階梯式排列，做成systolic array可以將critical path縮短。**



**Matlab模擬與modelsim模擬的差異，最大誤差為0.0195**

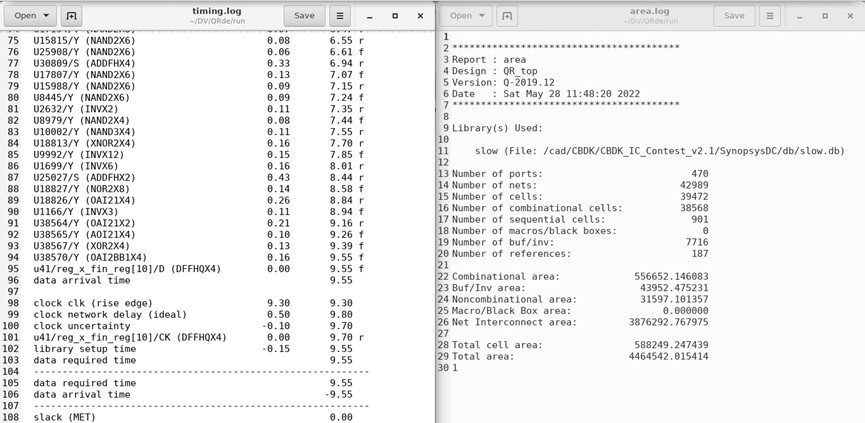
**電路合成**

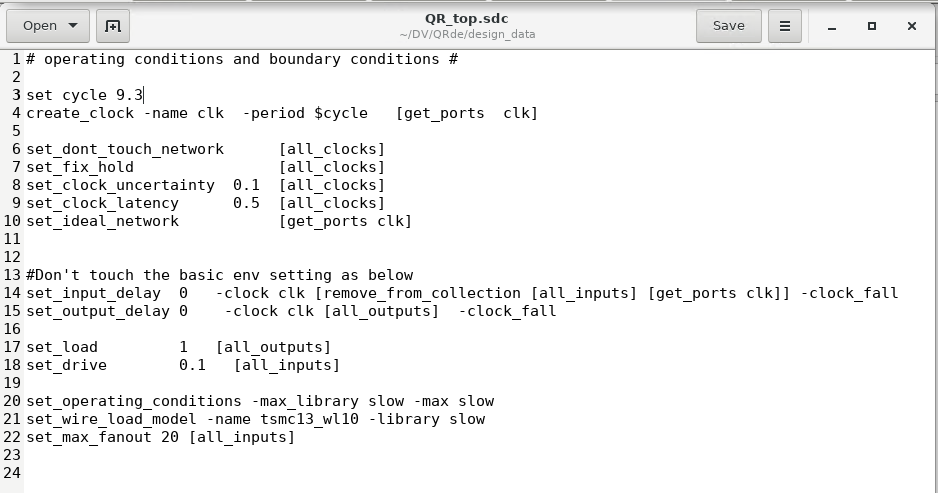
**使用DC進行電路合成，所得到的面積與時間如下圖所示**

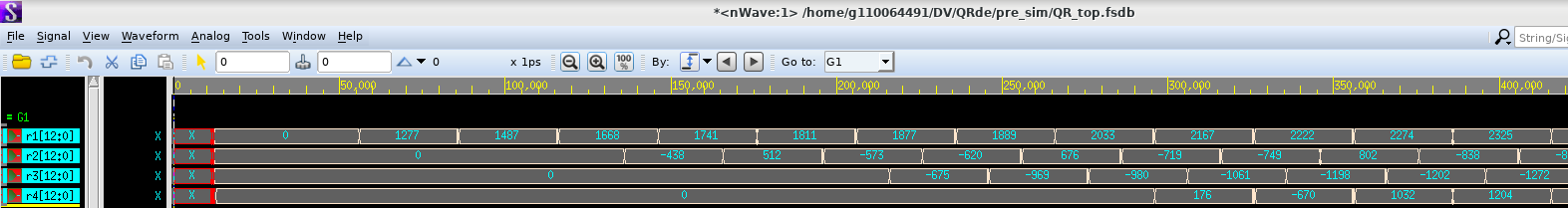
**可以看到 total cell area 為588249 (um)**

**而時間部分slack 為0**

**其中在sdc檔中設定clock cycle 為9.3(ns)，頻**

**率為107Mhz(如下圖所示)**



**Pre-sim**

**利用合成檔進行pre-sim 可以看到輸出值與modelsim的**

**RTL-simulation結果相同。**