**CAD/VLSI Circuit Design**

**期末報告**

在FPGA板上使用CORDIC演算法

搭配脈衝陣列的定點數QR分解

**Fixed point QR decomposition using CORDIC Algorithms on FPGA with systolic array**

**學號：7111064109**

**學生：林軒宇**

**指導教授：范志鵬**

**日期：2022/11/28**

目錄

[一、 簡介 - 1 -](#_Toc120530343)

[二、 理論 - 1 -](#_Toc120530344)

[ given rotation說明 - 1 -](#_Toc120530345)

[ 使用CORDIC達成given rotation - 2 -](#_Toc120530346)

[三、 架構改良 - 2 -](#_Toc120530347)

[ 改良一：PE Element改良 - 2 -](#_Toc120530348)

[ 改良二：input size改良 - 2 -](#_Toc120530349)

[ 改良三：systolic array 改良 - 4 -](#_Toc120530350)

[ 改良四：重新定義K - 4 -](#_Toc120530351)

[四、 設計規格 - 5 -](#_Toc120530352)

[ 系統方塊圖 - 5 -](#_Toc120530353)

[ QR\_CORDIC輸入輸出介面 - 5 -](#_Toc120530354)

[ GG輸入輸出介面 - 5 -](#_Toc120530355)

[ GR輸入輸出介面 - 6 -](#_Toc120530356)

[五、 模擬結果 - 6 -](#_Toc120530357)

[ Matlab模擬結果 - 6 -](#_Toc120530358)

[ Vivado模擬結果 - 8 -](#_Toc120530359)

[ Cell based simulation - 10 -](#_Toc120530360)

[六、 FPGA驗證 - 14 -](#_Toc120530361)

[ 系統架構圖 - 14 -](#_Toc120530362)

[ Top module輸入輸出介面(AXI-Stream) - 14 -](#_Toc120530363)

[ Block Design - 15 -](#_Toc120530364)

[ Result - 15 -](#_Toc120530365)

[七、 結果與討論 - 16 -](#_Toc120530366)

[ 對照組 - 16 -](#_Toc120530367)

[ 實驗組(一顆GG+一顆GR) - 17 -](#_Toc120530368)

[**八、** 參考文獻 - 18 -](#_Toc120530369)

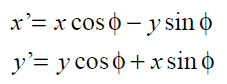
# 簡介

QR 分解是數值線性代數中具備多種用途的計算工具，主要應用於線性方程、最小平方法和特徵值問題。常見的 QR 分解的計算方法包括 Householder 變換、Givens rotation以及 Gram-Schmidt 正交法。本文使用given rotation搭配CORDIC Alogorithms。

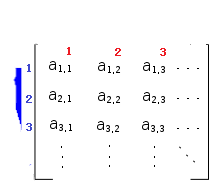
本次實作採用8\*4矩陣，每個數字大小定義在，預期得到一組8\*4的上三角矩陣R。實驗流程為先使用MATLAB估算預期使用定點數(fixed point)的長度(浮點數與定點數的誤差需足夠小)以及iteration的次數，再將MATLAB生成的隨機8\*4矩陣以定點數格式匯入verilog，並將verilog算出答案與matlab算出答案做比較，最後使用FPGA做驗證。

# 理論

## given rotation說明

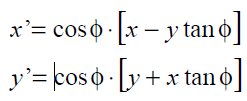
****

找到角度Φ使得y’= 0，以圖一為例，**a1,1**為x’、**a2,1**為y’，找出角度Φ後，需將右側同一列的數值皆經過同樣的旋轉矩陣運算(原理同基本矩陣第二定理)，之後依序將**a3,1**、**a4,1**…變成0，直到第一行除了**a1,1**外均變成0。同理第二行，將**a2,2**為x’、 **a3,2**為y’進行相同動作，直到第二行除了**a2,2**外皆變0。持續到第n行(最後一行)，即可得到上三角矩陣R。

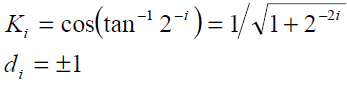
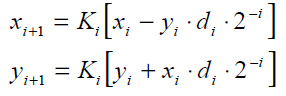


(圖一)given rotation範例圖

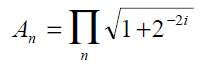
## 使用CORDIC達成given rotation



先將提出，接著限制，即可將算式簡化如下：

****，

為了簡化運算量，將每次的運量係數省略，最後再乘上所有系數的乘積和(AN)

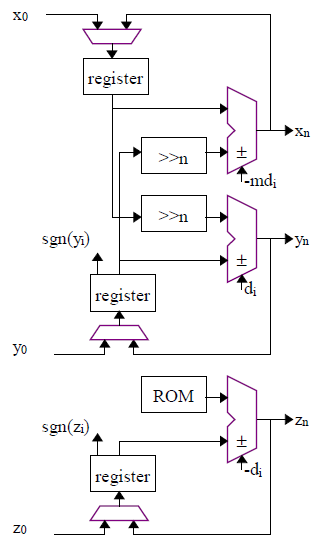
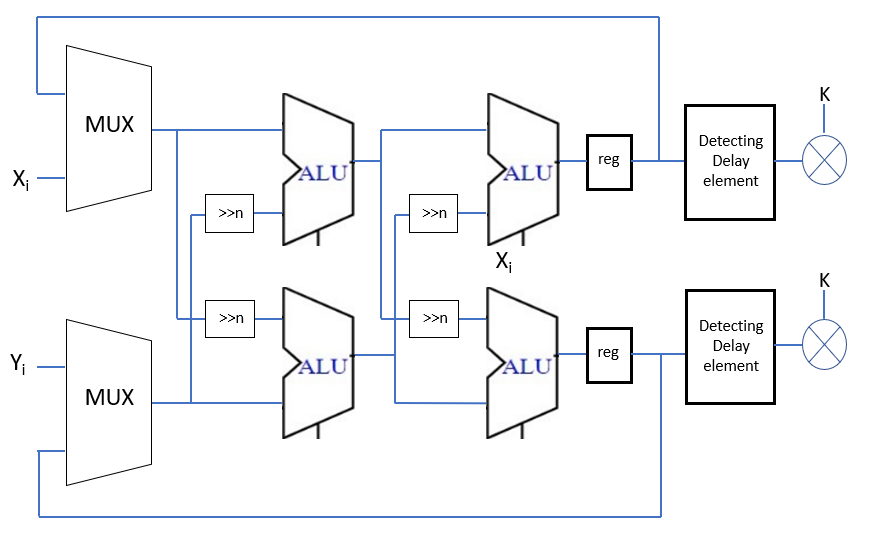


n為疊代的次數

# 架構改良

## 改良一：PE Element改良

(圖二)為論文中一顆PE元件的架構，(圖三)為改良後一顆PE元件的架構，最大的改進在於完成一次計算所需花的CLK數變原先的一半(unfolding factor(J)=2)。缺點為會增加Critical path，但此架構的Critical path為乘法器，因此可以近乎無代價的提高Performance

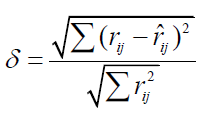
 

(圖二)論文PE元件 (圖三)改良後PE元件

## 改良二：input size改良

原論文疊代次數n為9，為了使用上文圖三架構，我們須選擇偶數次疊代次數，因此此文選用疊代次數(n)為8。此外原論文每個input為16bits，此文將每個input改為13bits(1 sign bit, 2 decimal bits, 10 fraction bits)，找法如下文。

首先定義delta凾式如下：



接著使用matlab畫圖，做法如下：

先將Fraction bit設定為5 bit，皆著連續測試10組δ值，如果不滿足δ<0.01，就將Fraction bit加１，結果如下圖：

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

橫軸為delta值，縱軸為index(從1~10)，由上圖可以看出唯有Fraction bit=10時，delta值會小於0.01，最終測出Fraction bit最小需要10 bit。

## 改良三：systolic array 改良

(圖四)為原論文架構，(圖五)為改良版架構，改良版為pipeline版本，在高頻下也可以成功運作。由於R22與R12間有資料相依，中間需要加delay，delay數由疊代次數(n)與J(unfolding factor)有關，算式如下：

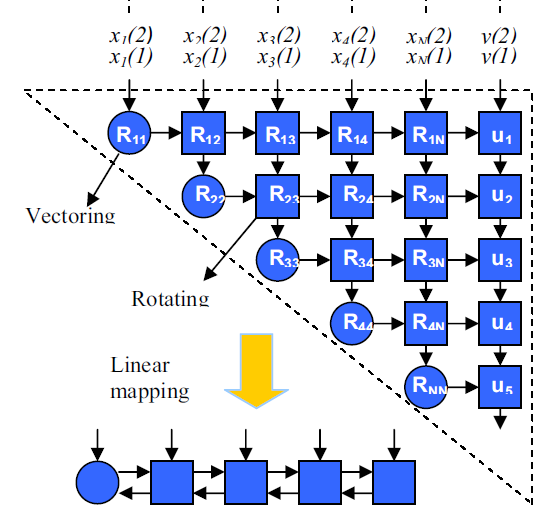
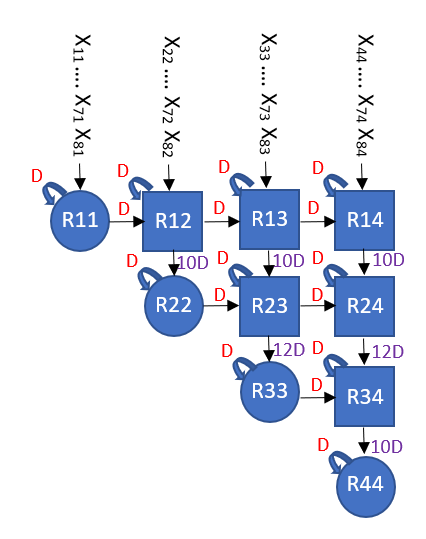
Delay >= (n/J+1)\*2

n/J為Xij執行Rotation mode的次數，加1為乘法器，乘2為有兩級資料相依。

舉例：以X72、X82為例，R22執行需等到R12執行完X72、X82才能計算。

同理(R13,R23)、(R14,R24)、(R34,R44)。

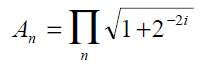
(R23,R33)、(R24,R34)多delay兩級目的為使用pipeline技巧，讓原先需要八顆乘法器降成4顆乘法器。最後結果如(圖五)。

(圖四)論文systolic array (圖五)改良後systolic array

## 改良四：重新定義K

從前文理論中，我們可以得知An如下：



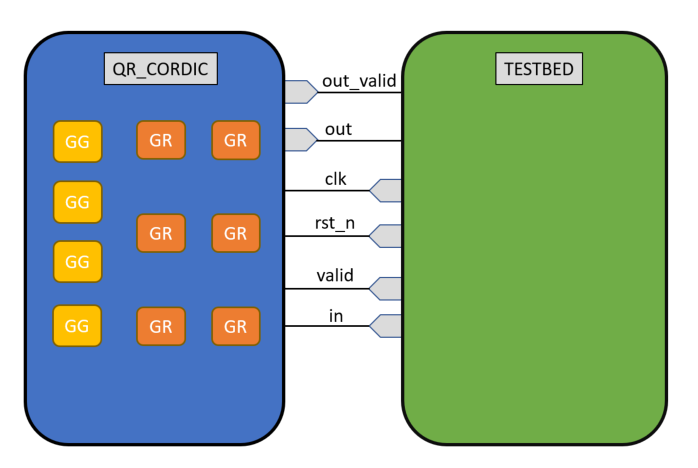
其中n為疊代次數，將n=8帶入，並將其重新定義成K，可得結果如下：

K=0.6074

最後將其轉為FIXED POINT(共11 bits, sign bit：1, fraction bit：10)

# 設計規格

## 系統方塊圖



## QR\_CORDIC輸入輸出介面

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | I/O | Width | Simple Description |
| Clk | I | 1 | Clock Signal(posedge trigger) |
| Rst\_n | I | 1 | Negedge reset |
| valid | I | 1 | valid為high，in資料開始輸入 |
| out\_valid | 0 | 1 | out\_valid為high，out資料開始輸出 |
| in | I | 52 | 為第i列資料，4筆13bits資料組成輸入 |
| out | 0 | 52 | 為第i列資料，4筆13bits資料組成輸出 |

## GG輸入輸出介面

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | I/O | Width | Simple Description |
| out\_X | 0 | 13 | Xij資料輸出 |
| out\_Y | 0 | 13 | Xi(j-1)資料輸出 |
| Sign\_d | 0 | 2 | 決定第i與(i+1)次疊代旋轉方向 |
| Iter\_num | I | 3 | 為第i次疊代，i為0、2、4、6 |
| In\_X | I | 13 | Xij資料輸入 |
| In\_Y | I | 13 | Xi(j-1)資料輸入 |

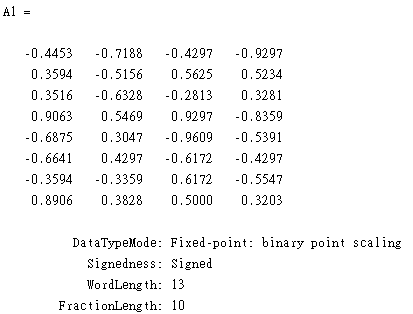
## GR輸入輸出介面

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | I/O | Width | Simple Description |
| out\_X | 0 | 13 | Xij資料輸出 |
| out\_Y | 0 | 13 | Xi(j-1)資料輸出 |
| Sign\_d | I | 2 | 決定第i與(i+1)次疊代旋轉方向 |
| Iter\_num | I | 3 | 為第i次疊代，i為0、2、4、6 |
| In\_X | I | 13 | Xij資料輸入 |
| In\_Y | I | 13 | Xi(j-1)資料輸入 |

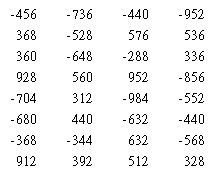
# 模擬結果

## Matlab模擬結果

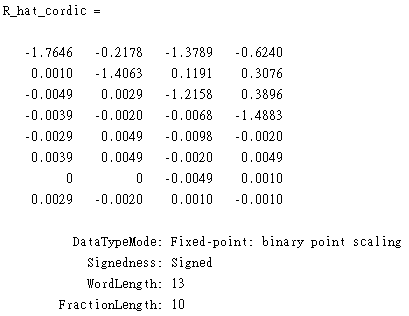
1. 8\*4 input matrix in floating



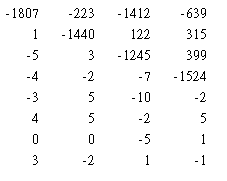
1. 8\*4 input matrix in sign decimal



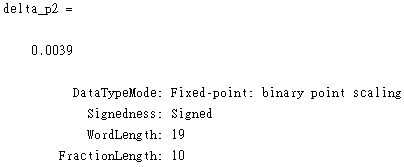
1. 8\*4 output matrix in floating



1. 8\*4 output matrix in sign decimal

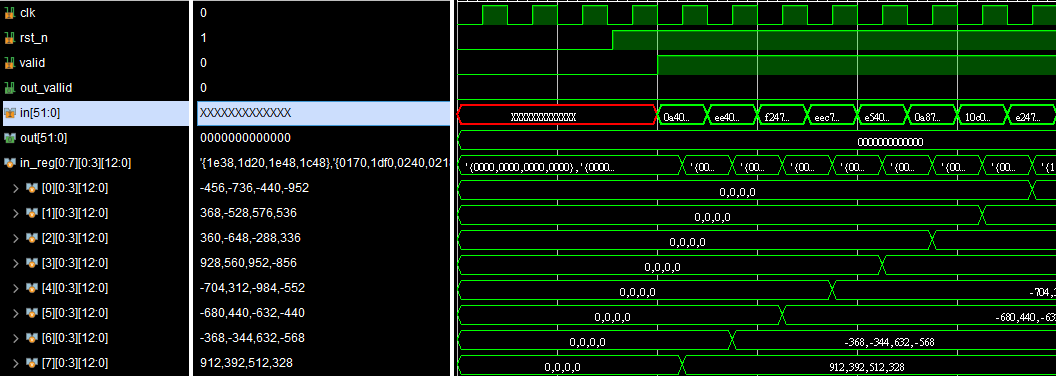


1. Delta



## Vivado模擬結果

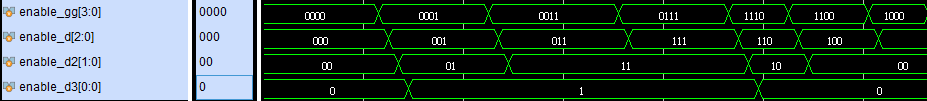
1. valid拉起時



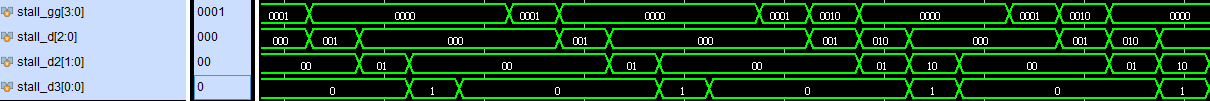
13bits 8\*4 in\_reg，用來存放8\*4矩陣的值

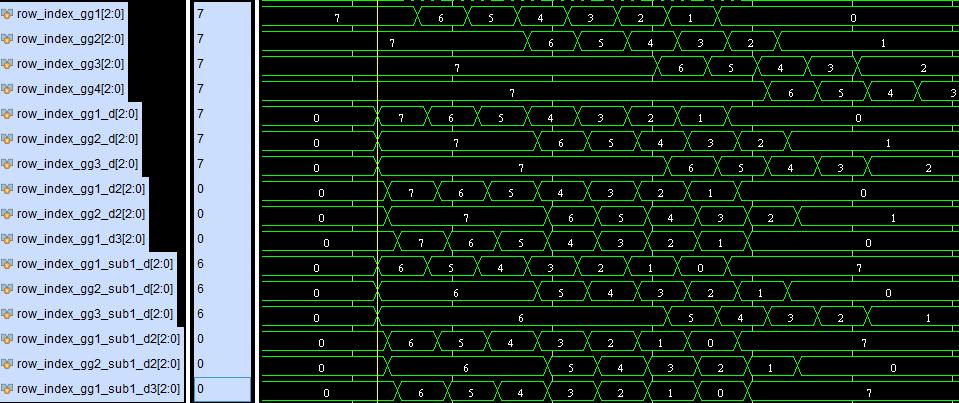
1. 中間計算的control訊號

控制GG、GR是否運作



控制MUL是否運作

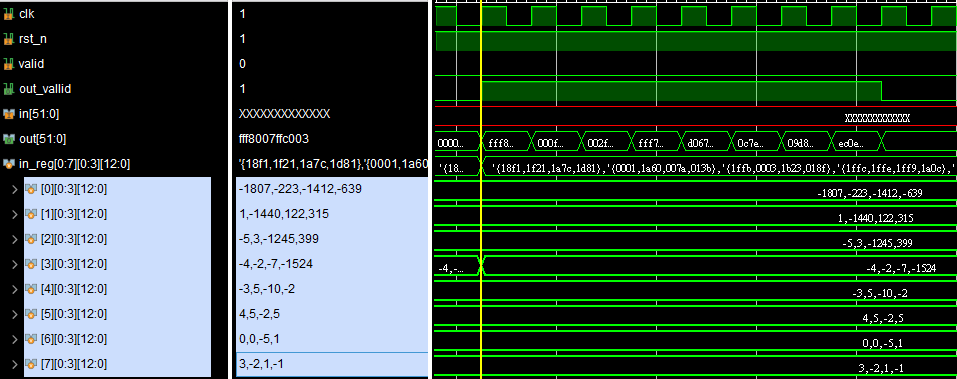




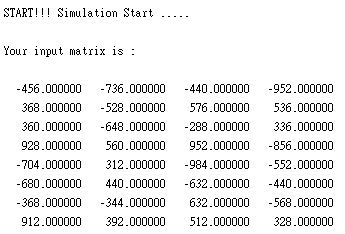
d、d2、d3分別代表同一列delay 1、2、3個clk

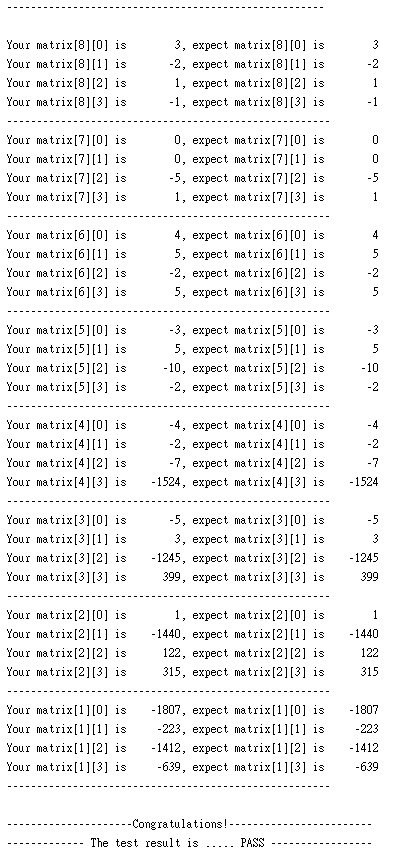
控制GG、GR存入in\_reg的指標

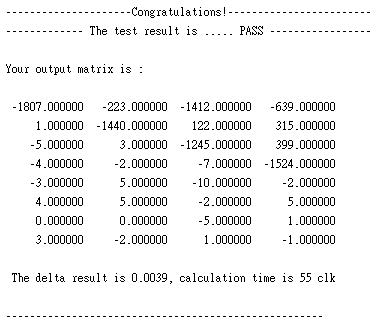
1. outvalid拉起時



1. TESTBED最終測試結果



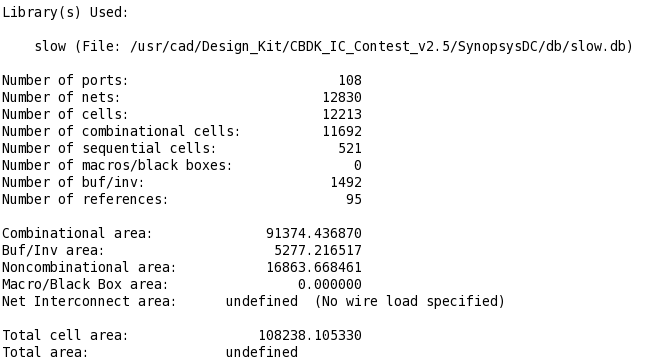




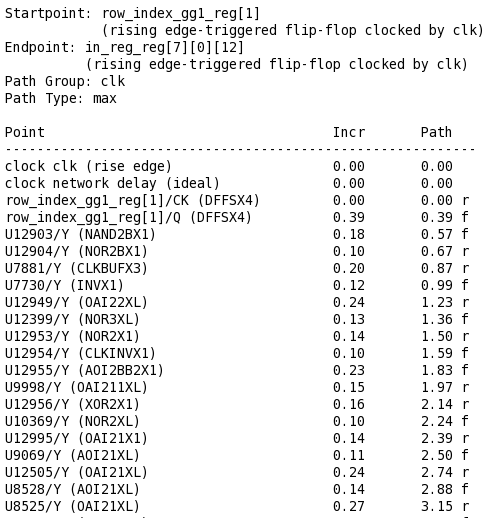
## Cell based simulation

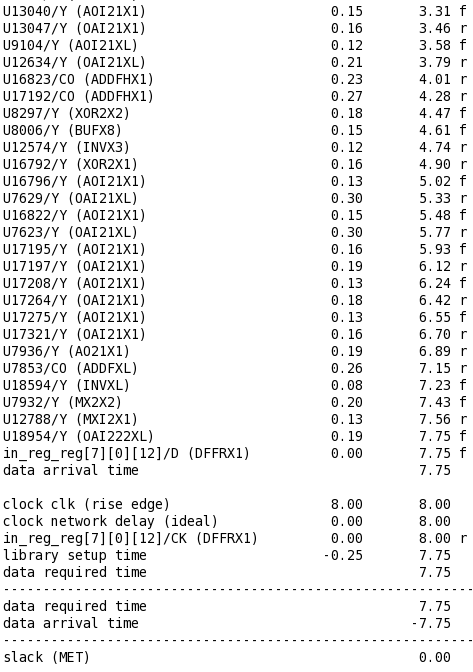
使用CIC 0.13um製程

1. Area report (area.log)



1. Timing report(timing.log)

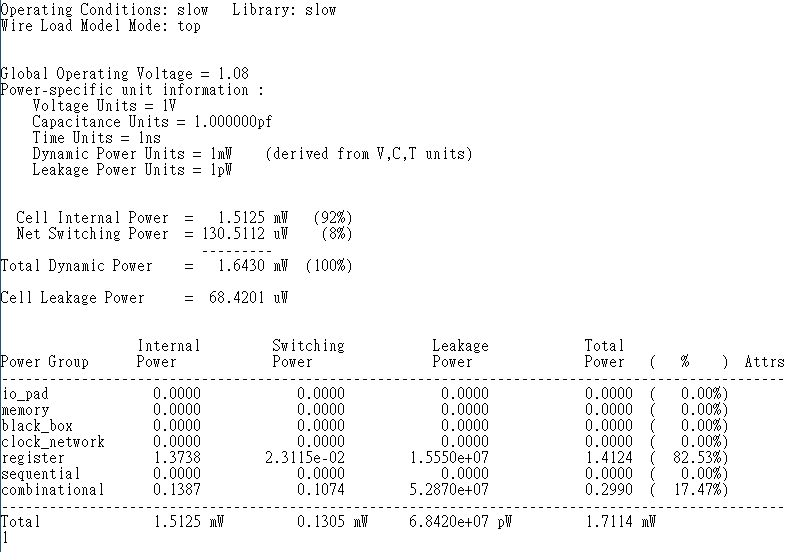




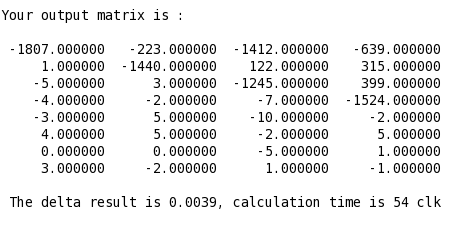
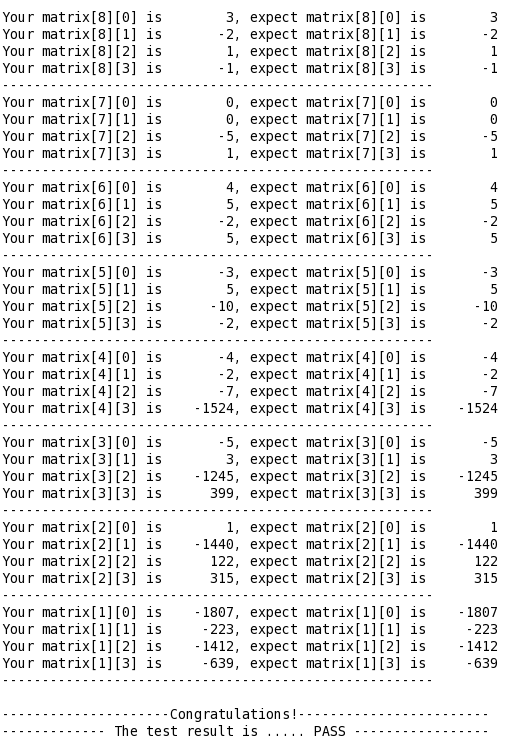
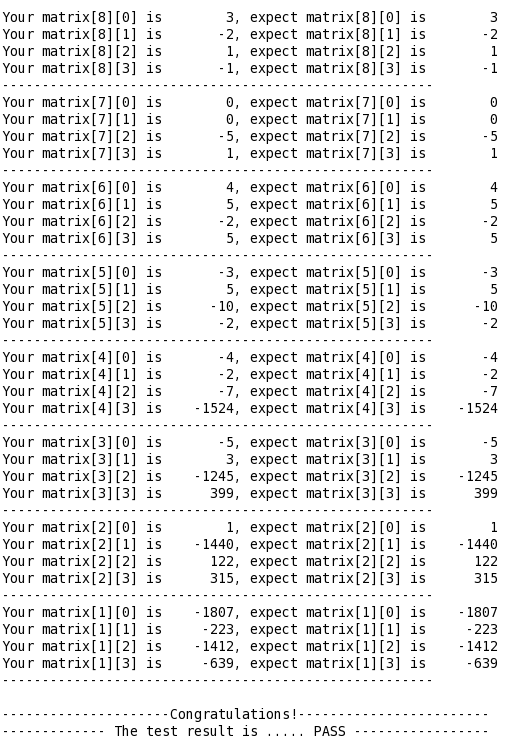
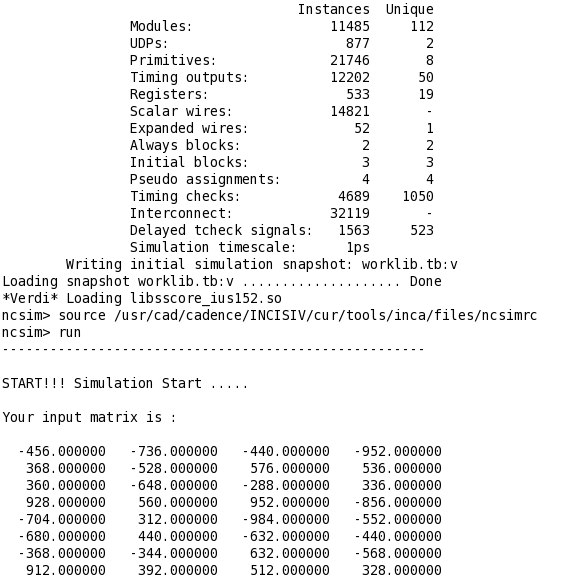
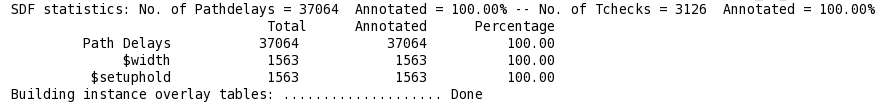
Clock period設定為8

Slack不能為負

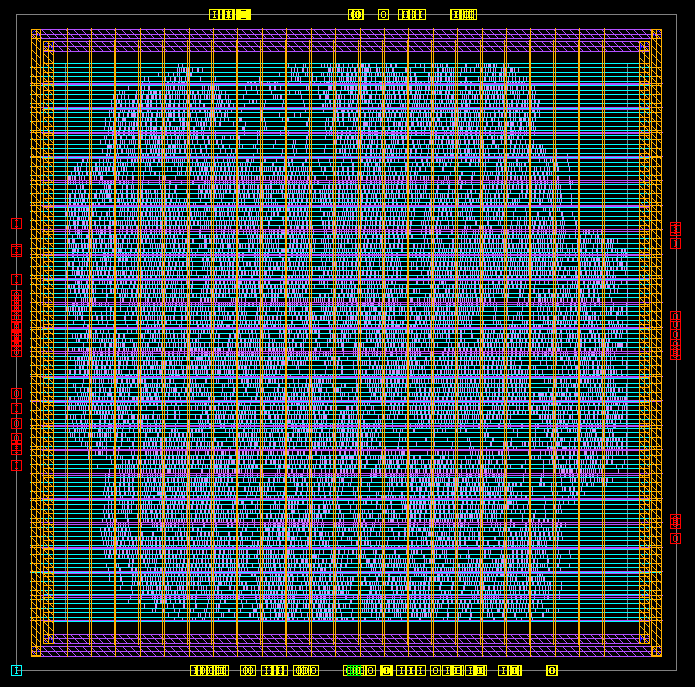
1. Power(power.log)



1. Gatelevel simulation(neverilog.log)

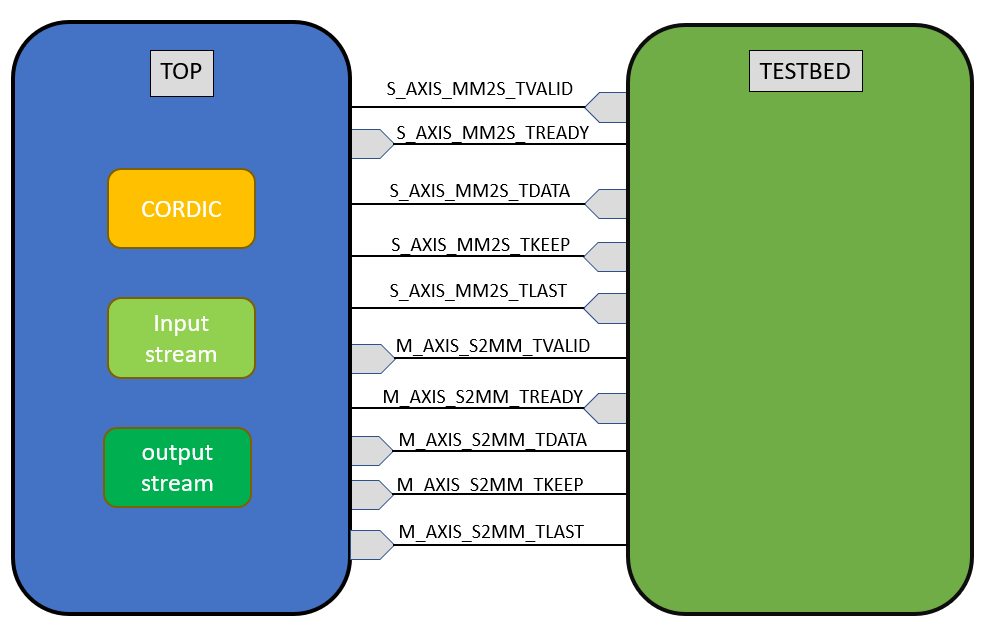


1. APR



# FPGA驗證

## 系統架構圖

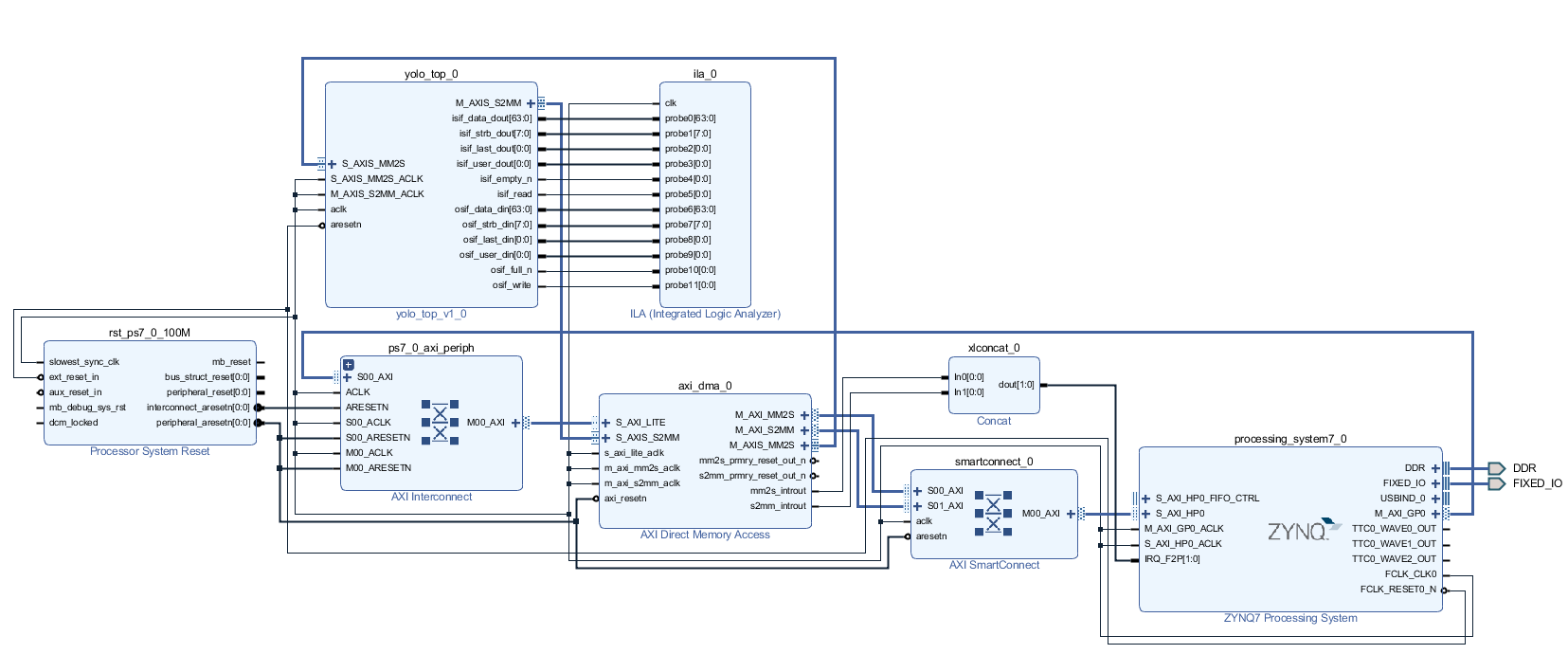


Top module輸入輸出介面(AXI-Stream)

|  |  |  |  |
| --- | --- | --- | --- |
| Signal Name | I/O | Width | Simple Description |
| S\_AXIS\_MM2S\_TVALID | I | 1 | TVALID表示主設備正在驅動一個有效的傳輸。當TVALID和TREADY都置位時，發生一個傳輸。 |
| S\_AXIS\_MM2S\_TREADY | O | 1 | TREADY表示從設備在當前週期能夠接收一次傳輸。 |
| S\_AXIS\_MM2S\_TDATA | I | 64 | TDATA是基本的有效載荷，用來提供跨越接口的數據。數據為整數個字節。 |
| S\_AXIS\_MM2S\_TKEEP | I | 8 | TKEEP是字節修飾符。用來表明TDATA相關字節的內容是否作為數據流的一部分被處理。TKEEP字節修飾符未被確認的那些相關字節是空字節，可以從數據流中去除。 |
| S\_AXIS\_MM2S\_TLAST | I | 1 | TLAST表明了包的邊界。 |
| aclk | I | 1 | Clock Signal(posedge trigger) |
| aresetn | I | 1 | negedge reset |
| M\_AXIS\_S2MM\_TVALID | O | 1 | 同上 |
| M\_AXIS\_S2MM\_TREADY | I | 1 | 同上 |
| M\_AXIS\_S2MM\_TDATA | O | 64 | 同上 |
| M\_AXIS\_S2MM\_TKEEP | O | 8 | 同上 |
| M\_AXIS\_S2MM\_TLAST | O | 1 | 同上 |

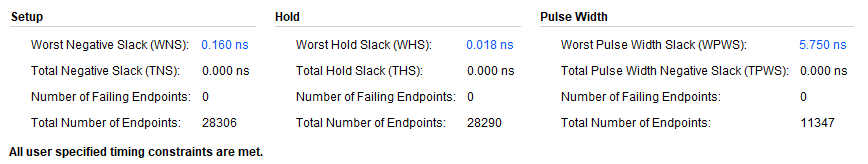
備註: (M2S:master to slave、S2M:slave to master)

## Block Design

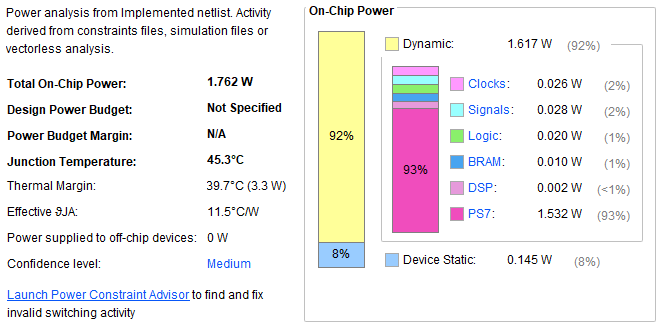


## Result

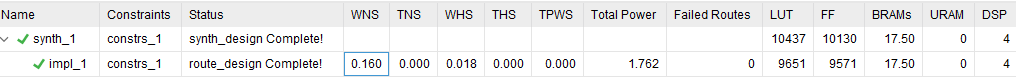
1. Timing Report



1. Power Report

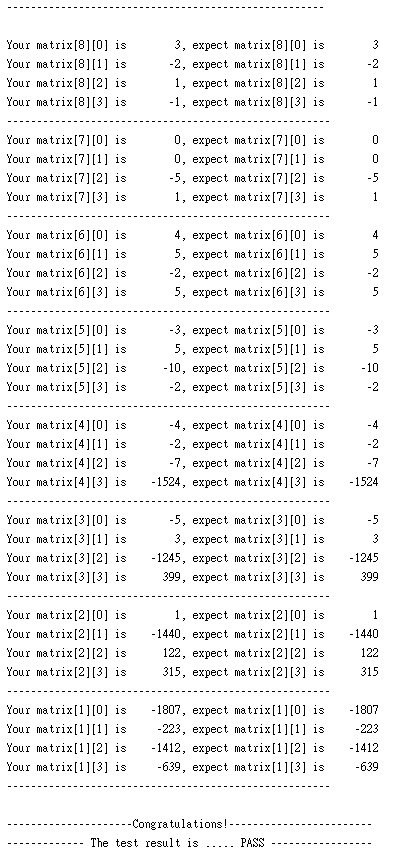


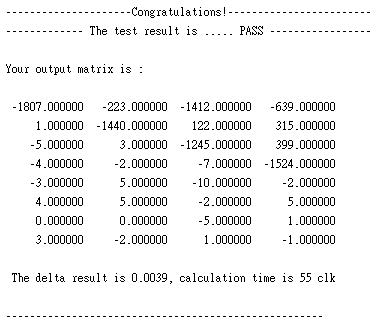
1. Total Report



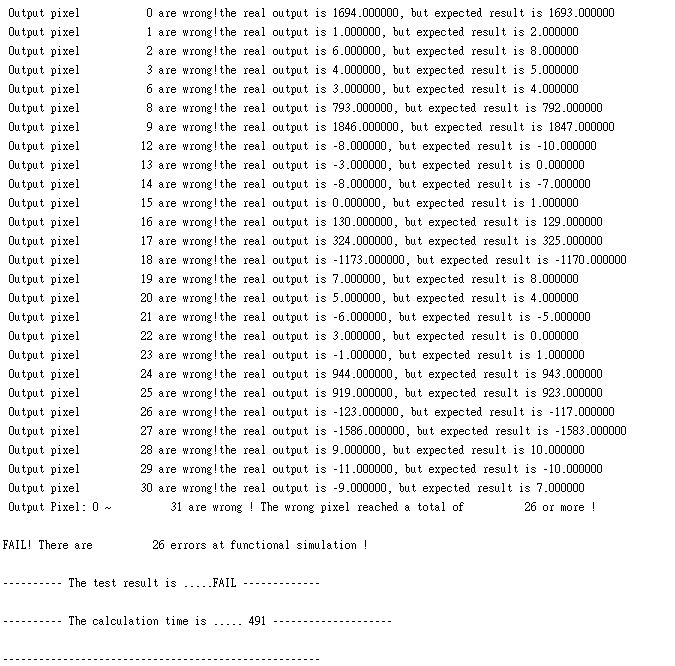
# 結果與討論

## 對照組





## 實驗組(一顆GG+一顆GR)



實驗組僅將PE元件改成一顆GG+一顆GR(即表示無systolic array)，其餘皆固定，可得結果如下：

1. 如上圖黃框所標處，有systolic array的總計算時間少了將近9倍
2. expect值為matlab計算後的值，造成實驗組紅框處expect value與output value不同的原因在於rounding的差異，需將matlab的fi函數rounding方式改為”Floor”，才可以使expect value 與output value符合預期。

# 參考文獻

### D. Boppana, K. Dhanoa and J. Kempa, "FPGA based embedded processing architecture for the QRD-RLS algorithm,"*12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines*, 2004, pp. 330-331, doi: 10.1109/FCCM.2004.34

### ANDRAKA, Ray. A survey of CORDIC algorithms for FPGA based computers. In: *Proceedings of the 1998 ACM/SIGDA sixth international symposium on Field programmable gate arrays.* 1998. p. 191-200