

FPGA Homework0 第八組

組員一：F44071128 李其祐 組員二：C14074021 張柏彥

組員三：F14051041 陳祺

一、設計規格：

題目給定的訊號：

Signal Name	Direction	Width(bit)	Description
rst	input	1	非同步系統重置訊號，當此訊號為 1 時表示系統重置。
clk	Input	1	系統時脈訊號。
en	Input	1	資料有效信號。當此訊號為 1 時表示輸入資料為有效。
central	Input	24	集合座標資料。其組成為{x1, y1, x2, y2, x3, y3}，其中 central[23:20]:集合 A 中的 X 軸座標(x1) central[19:16]:集合 A 中的 Y 軸座標(y1) central[15:12]:集合 B 中的 X 軸座標(x2) central[11:8]:集合 B 中的 Y 軸座標(y2) central[7:4]:集合 C 中的 X 軸座標(x3) central[3:0]:集合 C 中的 Y 軸座標(y3)
radius	Input	12	集合半徑資料。其組成為{r1, r2, r3}，其中 radius[11:8]:為集合 A 的半徑值 r1 radius[7:4]:為集合 A 的半徑值 r2 radius[3:0]:為集合 A 的半徑值 r3
mode	input	2	mode 為指示進行運算模式信號。
busy	output reg	1	系統忙碌指示訊號。當此訊號為 1 時表示系統忙碌中。
valid	output reg	1	Valid 為有效的資料輸出指示信號。當信號 valid 為高準位時，於 candidate 輸出埠所輸出之資料為有效資料。
candidate	output reg	8	輸出題目所指定的元素個數。

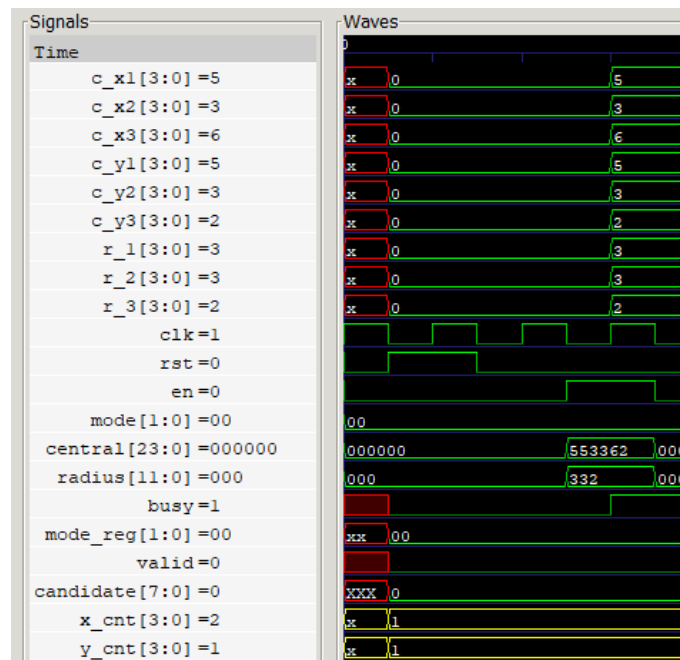
我們自定義的訊號：

Signal Name	Type	Width(bit)	Description
c_x1, c_y1	reg	4	儲存集合 A 的 X, Y 軸座標
c_x2, c_y2	reg	4	儲存集合 B 的 X, Y 軸座標
c_x3, c_y3	reg	4	儲存集合 C 的 X, Y 軸座標
r_1, r_2, r_3	reg	4	儲存集合 A, B, C 的半徑長
x_cnt, y_cnt	reg	4	儲存現在位於 8x8 座標系統空間的座標
mode_reg	reg	2	儲存 mode 的值

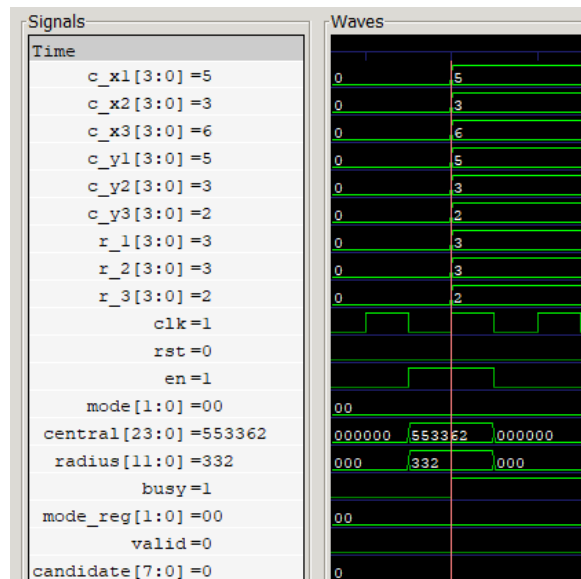
Signal Name	Type	Width(bit)	Description
sub_x1, sub_y1	wire	4	紀錄現在座標與集合 A 的距離
sub_x2, sub_y2	wire	4	紀錄現在座標與集合 B 的距離
sub_x3, sub_y3	wire	4	紀錄現在座標與集合 C 的距離
sqr_x1, sqr_y1	wire	8	紀錄現在座標與集合 A 的距離平方
sqr_x2, sqr_y2	wire	8	紀錄現在座標與集合 B 的距離平方
sqr_x3, sqr_y3	wire	8	紀錄現在座標與集合 C 的距離平方
sqr_r1, sqr_r2, sqr_r3	wire	8	紀錄各圓心半徑長的平方
sum_1, sum_2, sum_3	wire	8	紀錄現在座標與各圓心距離平方和

二、電路設計說明與模擬結果：

當 rst 訊號為 high 時，將 c_x1, c_y1, c_x2, c_y2, c_x3, c_y3, r_1, r_2, r_3, busy, valid, candidate 和 mode_reg 初始化為 0(如下圖)。此外由題目可知座標範圍為 1-8，因此將 x_cnt, y_cnt 初始化為 1。

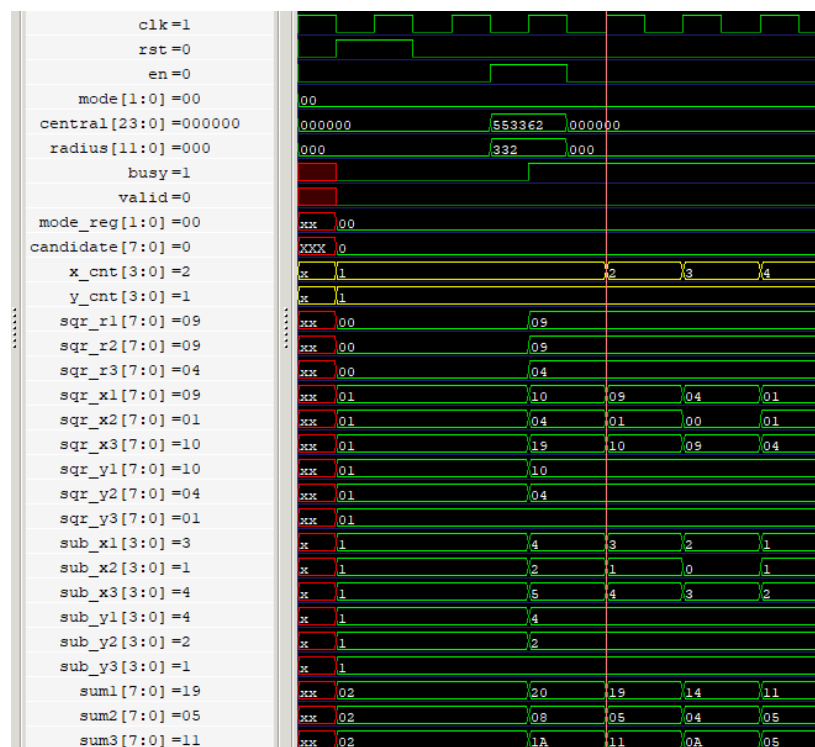


當 busy=0 且 en=1 時(如下圖紅線時刻)，central, radius 訊號將會在下一個 cycle 寫入暫存器，從題目可得知 central[23:20] 為第一個圓的圓心 x 座標，central[19:16] 為第一個圓的圓心 y 座標，central[15:12] 為第二個圓的圓心 x 座標，central[11:8] 為第二個圓的圓心 y 座標，central[7:4] 第三個圓的圓心 x 座標，central[3:0] 第三個圓的圓心 y 座標，且 radius[11:8] 為第一個圓的半徑值，radius[7:4] 為第二個圓的半徑值，radius[3:0] 為第三個圓的半徑值，並分別寫入 c_x1, c_y1, c_x2, c_y2, c_x3, c_y3 及 r_1, r_2, r_3。最後將 busy 訊號在下個 cycle 拉為 high。

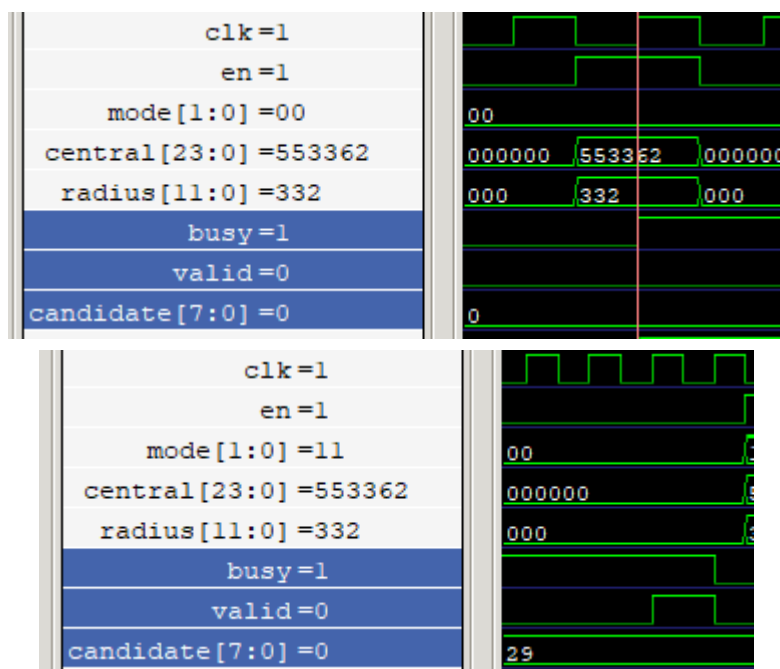


*註: 已經有經過 vivado 模擬(vivado 模擬圖放在本文最後面)，但因為使用 gtkwave 較為方便、快速，所以以下波型圖都擷取自 gtkwave

當 busy=1 且 valid=0 時(如下圖紅線時刻)，我們利用 x_cnt 及 y_cnt 控制座標點(如下圖黃色訊號)，x_cnt 會在每個 cycle 後加 1，直到 x_cnt = 8 時，會在下一個 cycle 變回到 1，並將 y_cnt 也加 1。此時 sub_x1, sub_y1, sub_x2, sub_y2, sub_x3, sub_y3 會記錄 x_cnt, y_cnt 與各個圓心 x, y 座標的距離，接著 sqr_x1, sqr_y1, sqr_x2, sqr_y2, sqr_x3, sqr_y3 會記錄 x_cnt, y_cnt 與各個圓心 x, y 座標距離平方，srq_r1, sqr_r2, sqr_r3 則為各個圓的半徑平方。最後 sum1, sum2, sum3 則為 x_cnt, y_cnt 與各個圓心 x, y 座標距離平方和。在獲得所有和集合的距離後，就會根據 mode 決定此座標是否是 candidate 的變化條件。



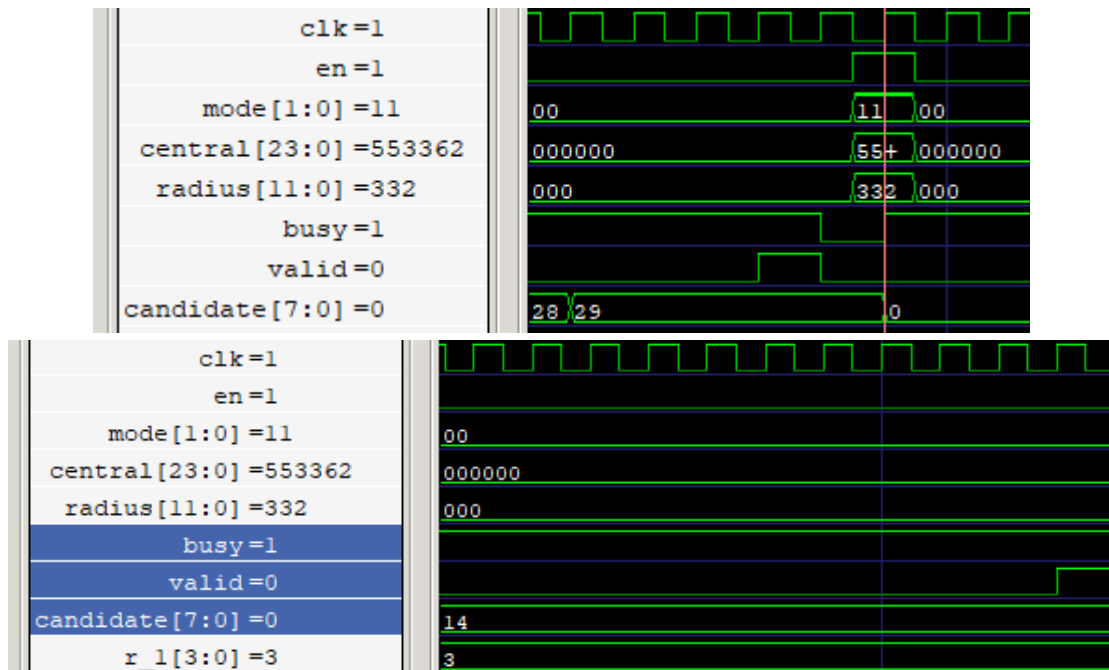
接著說明 4 種模式的設計原理，當 mode 為 0 時(如下圖紅線時刻)，計算一個 A 集合在二維 8x8 空間裡所涵蓋的座標元素個數，也就是判斷和圓心 A 的距離是否小於等於集合 A 之半徑，因此當 $\text{sum1} \leq \text{sqr_r1}$ 時，代表該座標在第一個圓的空間裡，因此 candidate 加 1。最後經過本電路運算後，輸出 candidate 為 29。



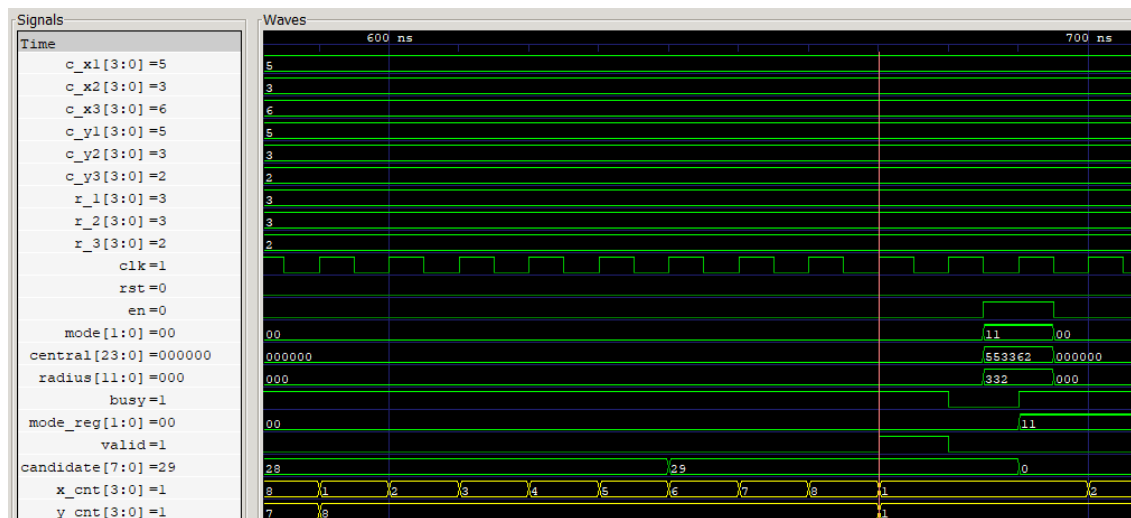
當 mode 為 1 時，計算 A、B 兩個集合，在 2 維 8x8 座標空間裡，交集所涵蓋的元素座標點個數，也就是判斷某點座標和圓心 A、B 的距離是否都小於等於 A、B 集合的半徑，因此當 $(\text{sum1} \leq \text{sqr_r1}) \text{ 且 } (\text{sum2} \leq \text{sqr_r2})$ 時，candidate 加 1。

當 mode 為 2 時，計算 A、B 兩個集合在 2 維 8x8 座標空間裡，差集所涵蓋的元素座標點個數，也就是判斷此座標是否在 A 集合內且 B 集合外，或是在 B 集合內，A 集合外，因此當 $(\text{sum1} \leq \text{sqr_r1}) \text{ 且 } (\text{sum2} > \text{sqr_r2})$ 或是 $(\text{sum2} \leq \text{sqr_r2}) \text{ 且 } (\text{sum1} > \text{sqr_r1})$ 時，candidate 加 1。

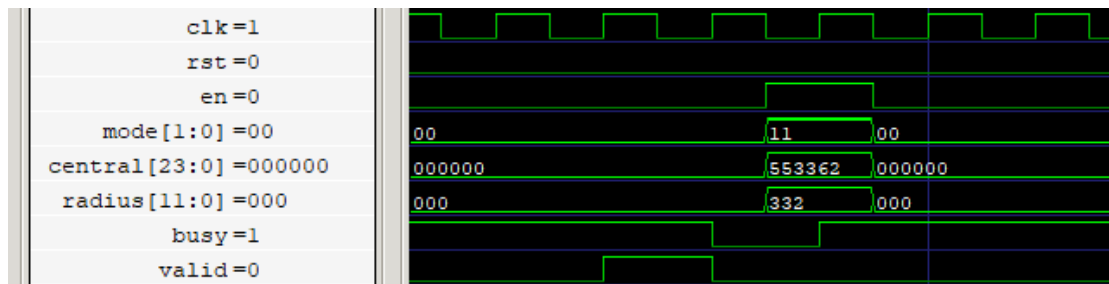
當 mode 為 3 時(如下圖紅線時刻)，計算 A、B、C 三個集合在 2 維 8x8 座標空間裡，每個兩個集合作交集元素座標點個數，並排除三個集合所交集之元素座標點個數，也就是判斷以下三種可能，即此座標是否在 A、B 集合內，C 集合外，或是 B、C 集合內，A 集合外，或是 C、A 集合內，B 集合外，因此當 $((\text{sum1} \leq \text{sqr_r1}) \text{ 且 } (\text{sum2} \leq \text{sqr_r2}) \text{ 且 } (\text{sum3} > \text{sqr_r3}))$ 或是 $((\text{sum2} \leq \text{sqr_r2}) \text{ 且 } (\text{sum3} \leq \text{sqr_r3}) \text{ 且 } (\text{sum1} > \text{sqr_r1}))$ 或是 $((\text{sum3} \leq \text{sqr_r3}) \text{ 且 } (\text{sum1} \leq \text{sqr_r1}) \text{ 且 } (\text{sum2} > \text{sqr_r2}))$ 時，candidate 加 1。最後經過本電路運算後，輸出 candidate 為 14。



當 $y_cnt = 8$ 且 $x_cnt = 8$ 時，代表所有座標點都已經執行過一輪，因此將 $valid$ 在下一個 cycle 拉為 1(如下圖紅線時刻)，使 $candidate$ 為有效輸出。



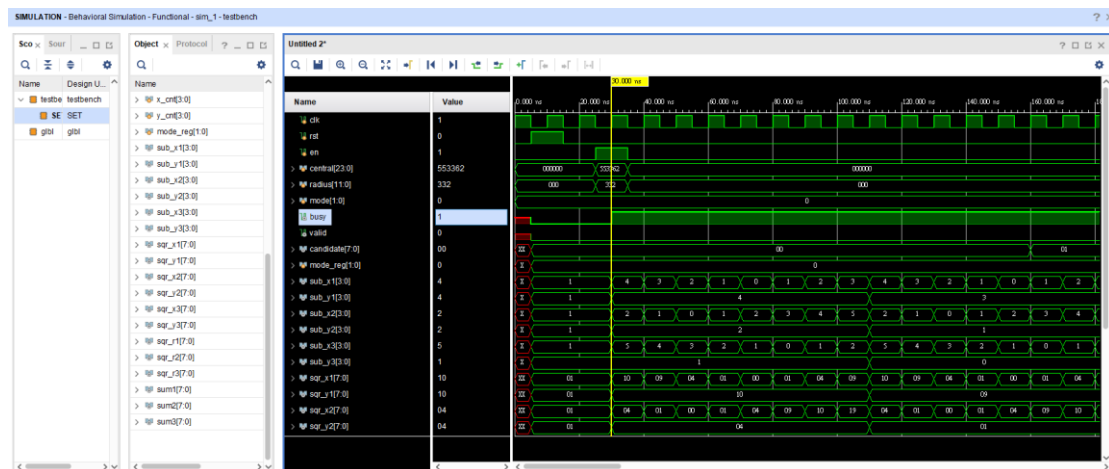
最後，當 $busy=1$ 且 $valid=1$ 時，在下個 cycle 把 $busy$ 與 $valid$ 設為 0。(如下圖)



三、設計成果：

我們除了使用助教所提供的 tb 檔(SETtb.v)以外，也去網路上搜尋此年的測試樣本 (testfixture.v)，裡面總共有 4 個測資(MD1~MD4)，經過 NC-verilog 還有 Vivado 驗證後，確認我們所寫的 RTL Code 是可以通過 pre-sim。此外我們也將 RTL code 丟入 Design compiler 合成，也有通過 gate-level simulation，從下圖可以看到我們 critical path length < critical path clk period，也就是沒有 timing violation，最後我們比對此年面積要求以及時脈要求有達到 A 等級，即面積小於 9000 且操作時脈在 10ns。

Vivado 模擬：



QoR Report:

Timing Path Group 'clk'		Area	
Levels of Logic:	26.00	Combinational Area:	6207.391809
Critical Path Length:	9.69	Noncombinational Area:	1897.693161
Critical Path Slack:	0.00	Buf/Inv Area:	733.276801
Critical Path Clk Period:	10.00	Total Buffer Area:	178.23
Total Negative Slack:	0.00	Total Inverter Area:	555.05
No. of Violating Paths:	0.00	Macro/Black Box Area:	0.000000
Worst Hold Violation:	0.00	Net Area:	71960.360229
Total Hold Violation:	0.00		
No. of Hold Violations:	0.00	Cell Area:	8105.084970
		Design Area:	80065.445200

Design WNS: 0.00 TNS: 0.00 Number of Violating Paths: 0

Design (Hold) WNS: 0.00 TNS: 0.00 Number of Violating Paths: 0