HDL Digital Design (Graduate Level)

Spring 2024

Lab 9 Geofence

REPORT

Must do self-checking before submission:

Compress all files described in the problem into one zip file.

All files can be compiled under ModelSim & Quartus environment.

All port declarations comply with I/O port specifications.

Organize files according to File Hierarchy Requirement

No waveform files or project file in deliverables

Due Date: 2024/05/23 8:59:59 a.m.

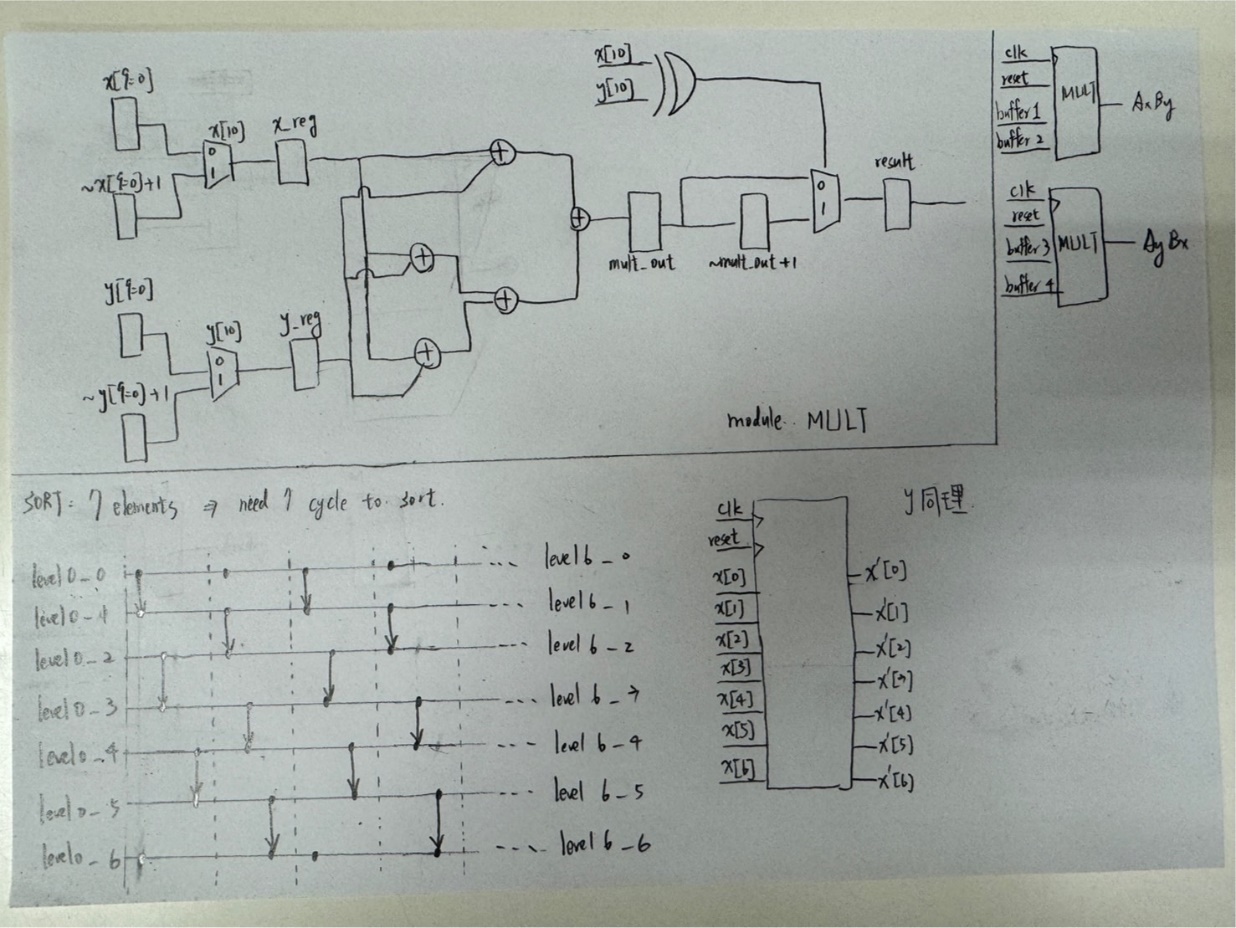
Student name: \_\_\_林柏戎\_\_\_\_

Student ID: \_\_Q36114239\_\_\_\_

1. Performance & Area table

|  |  |
| --- | --- |
| Gate level simulation all pass (Yes/No) | Yes |
| Clock period in testbench | 12.06 (ns) |
| Simulation time in Gate Level | 10781640 (ps) |
| Area  (Total logic elements + 1000 \* embedded Multiplier) | 1154 |

1. Architecture Diagram (10%)



1. Explain algorithm & how your design works? (25%)

OBJECT:

* 將pattern中的第2個的值存下來為目標座標。(1 cycle)

INPUT:

* 將pattern中的第3~9的值存在sort\_x array and sort\_y array。(7 cycles)

EDGE:

* 若 (bottom\_y > obj\_y) | (bottom\_y > obj\_x) | (obj\_y > top\_y) | (obj\_x > top\_y)，則代表目標座標肯定不會再凸包內，就會進入DONE狀態。(1 cycle)

SORT:

* 可以知道第3~9的值哪一個在”最左邊”，並記錄下來。
* 令最左邊的值為中心點，若第3~9的y值比”最左邊”的y值小，則對應的x值會作2補數，達到翻轉的設計方法。
* 得到翻轉後的 x array後，就可以從小到大排序，y值則是根據x的狀況作排序。
* 排序完後，會將剛剛有翻轉的值，再翻轉回去。
* 當翻轉前=翻轉後，則會跳離SORT，進入FIND狀態。(3-5 cycles)

(若只觀察第3~9的y值比”最左邊”的y值小，會有某些狀況會fail，fail的case會在Verification design作說明。)

FIND:

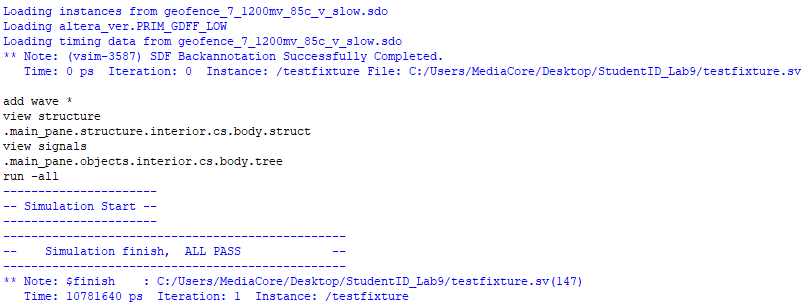
* 將目標物與排序完的sort\_x array and sort\_y array作外積，可以得到AxBy 與AyBx的數值。
* 當AxBy > AyBx && count > 1 ，代表目標物在凸包的外面，會有count > 1是因為我設計的乘法器會delay 1 cycle，所以當count > 1才會得到正確的乘法數值。
* 當count = 4'd8，代表目標物在凸包的裡面，因為七個點都與目標物計算完外積，但都沒發生AxBy > AyBx的情況，故目標物在凸包的裡面。

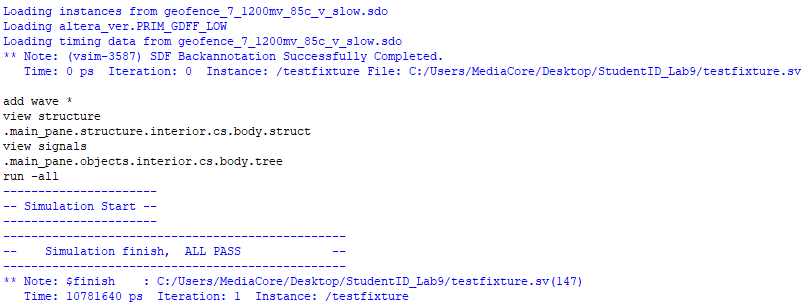
(AxBy>AyBx and count > 1) || (count = 4'd8)，會進到OUTPUT狀態。(1-8 cycles)

DONE:

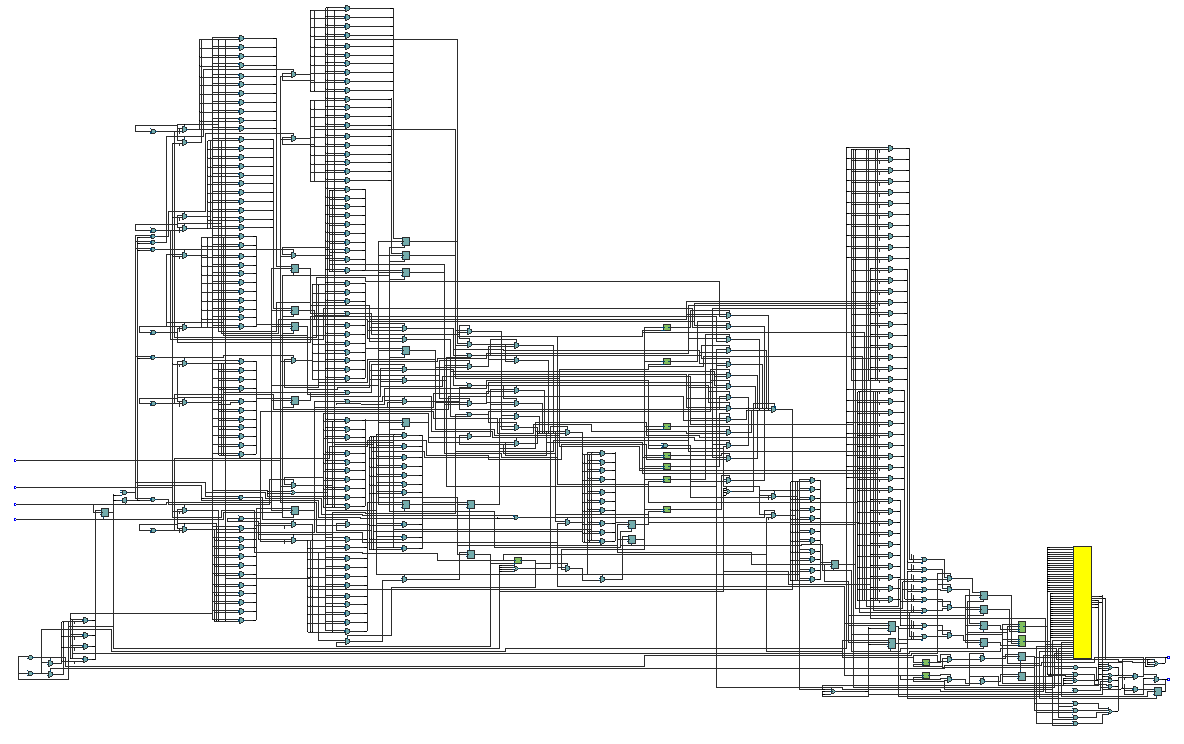
* 將valid拉高，結束目前的pattern。

1. RTL & Gate-level simulation result on the terminal. (5%)

RTL : 

Gate-level: 

1. Screenshot form Quartus (10%, 2% each)
   * Schematic view of the design netlist



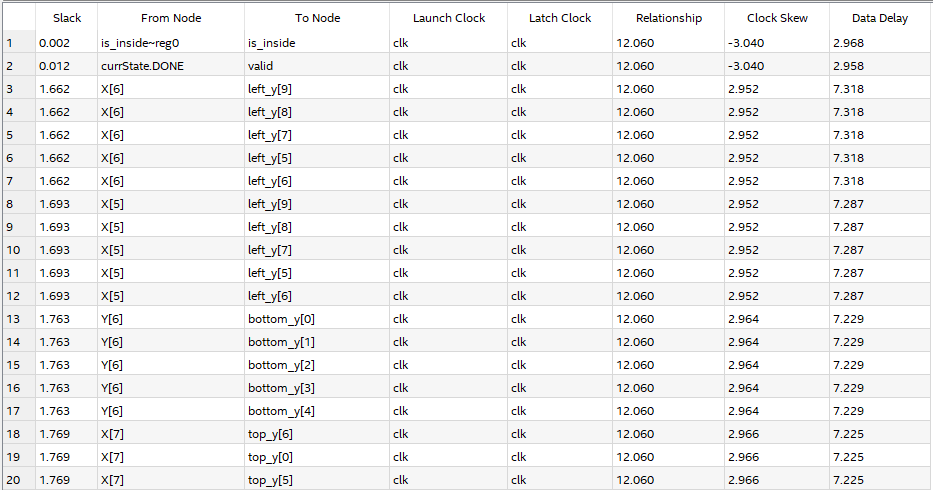
* + Finite State Machine view



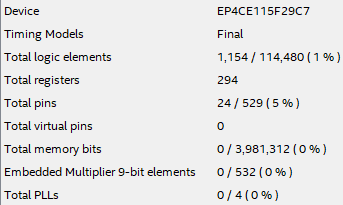
* + Fmax summery



* + worst-case timing paths



* + flow summery



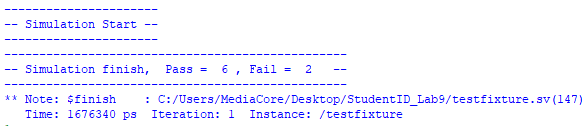
1. Verification (create new pattern in verification.txt) (40%)
   * Explain the pattern you generate to test your design

在Geofence的design中，若要100%正確，代表”sort一定要順時針或逆時針排序”以及”判斷AxBy與AyBx的關係不能有誤”。

1. 在Verification.txt的pattern 1~4中，sort有作正確的排序，以及判斷AxBy與AyBx的關係沒有問題，所以會pass。
2. 在Verification.txt的pattern 5~8中，sort沒有作正確的排序，導致計算AxBy與AyBx會是錯誤的方向，所以有些pattern會pass，有些pattern會fail。

* 當目標點座標在凸包外時，雖然是使用錯誤的sort排序作外積，但判斷AxBy與AyBx的關係剛好不會有問題，所以在pattern 5、pattern 6會是pass。
* 當目標點座標在凸包內時，因為sort不是正確的排序，所以在判斷 AxBy與AyBx 的關係就有問題，所以在pattern 7、pattern 8會有fail的狀況發生。

1. 綜合I.、II.，只要sort有正確的按照順時針或逆時針排序，在後續的判斷AxBy與AyBx的關係基本上沒有問題。
   * Simulation result



1. lesson learned (10%)

* 理解什麼是地理圍籬系統，利用接收器的座標來判斷物體是否在虛擬圍籬內。學習如何設計一個硬體架構來處理多次輸入(包括目標座標和接收器座標)並根據計算結果產生相應的輸出信號。
* 接收器排序演算法可以利用斜率以順時針或逆時針方向進行排序。我使用比大小的方式來達到排序的效果，可以減少乘法器以及除法器的使用。
* 學習如何根據座標確定物件是在定義區域內部還是外部。了解向量外積計算方法，用來判斷兩個向量的方向關係，從而確定目標物體是否在圍籬內。
* 產生有意義的pattern來測試和驗證自己的設計，可以知道自己設計的架構的coverage有多高，同時也可以知道該往甚麼方向作改進，進而提高coverage。
* 設計自己的11bits乘法器，在FPGA上就不會使用到9bits乘法器，且面積會更少，因為9bits乘法器的加權是1000，但自己設計的話，面積400左右。
* 設計自己的10bits比較器，在找left\_x, left\_y…時可以用到，這樣面積也會比直接使用大於或小於還小一點。