
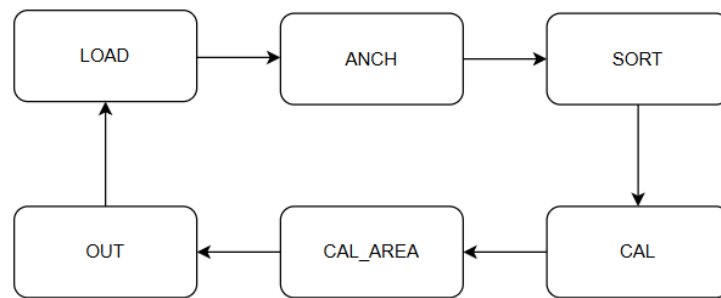


2025 Digital IC Design Homework 5

NAME	王華昀		
Student ID	N2614308		
Simulation Result			
Functional simulation	Pass	Pre-Layout simulation	Pass
<pre># ===== Running pattern 8 ===== # [PASS] Pattern 8: area = 01066 # # ===== Running pattern 9 ===== # [PASS] Pattern 9: area = 15e7a # # ===== RESULT ===== # # All 10 patterns passed! # Cycle: 2321 # # ** Note: \$finish : C:/DIC_2025/HW5/syn18_22pass/file/testfixture.sv(83) # Time: 49205200 ps Iteration: 0 Instance: /testfixture # .</pre>		<pre># ===== Running pattern 8 ===== # [PASS] Pattern 8: area = 01066 # # ===== Running pattern 9 ===== # [PASS] Pattern 9: area = 15e7a # # ===== RESULT ===== # # All 10 patterns passed! # Cycle: 2321 # # ** Note: \$finish : C:/DIC_2025/HW5/syn18_22pass/file/testfixture.sv(83) # Time: 49205200 ps Iteration: 0 Instance: /testfixture</pre>	
Synthesis Result			
Total logic elements	1927		
Total memory bits	0		
Total registers	363		
Embedded multiplier 9-bit elements	4		
Clock period (ns)	21.2		
Total Cycle used	2321		
Flow Summary			
 <<Filter>>			
Flow Status		Successful - Wed Jun 18 13:50:20 2025	
Quartus Prime Version		20.1.1 Build 720 11/11/2020 SJ Lite Edition	
Revision Name		MCH	
Top-level Entity Name		MCH	
Family		Cyclone IV E	
Device		EP4CE55F23A7	
Timing Models		Final	
Total logic elements		1,927 / 55,856 (3 %)	
Total registers		363	
Total pins		36 / 325 (11 %)	
Total virtual pins		0	
Total memory bits		0 / 2,396,160 (0 %)	
Embedded Multiplier 9-bit elements		4 / 308 (1 %)	
Total PLLs		0 / 4 (0 %)	
Description of your design			



本次作業主要分為六個狀態去進行實作:

- **LOAD:**

讀取 X、Y 數值，並將數值存到 buffer 中。Buffer 深度是 20 寬度是 16bit。

- **ANCH:**

LOAD 完 20 個 cycle 後，會進到 ANCH 狀態，在這個 cycle 會確定 ANCH 位置，並將 ANCH 位置記錄到 DFF 中。

- **SORT:**

SORT 狀態主要是將 buffer 中的 20 個點根據極角排列，使用的排列演算法是 bubble sort，首先會先將 ANCH 擺到 0 的位置，接著就隨著極角由小排到大，由於是在同一塊 buffer 中進行排列，因此使用的是交換策略，也就是倆倆對比比較小的會交換到前面，比較大的則會交換到後面。

- **CAL:**

CAL 狀態主要就是在計算 stack 的 push 跟 pop，我同樣是使用同一塊 buffer 完成這個操作，操作的方法是使用 stack 加上 in-place 的概念，透過兩個 pointer 去進行 push 跟 pop，第一個 pointer 是紀錄現在 stack 的位置，第二個 pointer 是紀錄下一個點的位置，根據兩極角的關係決定 pointer 移動的方式，並記錄最後 stack 的位置，而 stack 後面的位置則不管他。

- **CAL_AREA:**

CAL_AREA 這個狀態是在計算面積，根據 CAL 狀態紀錄的 stack 位置，需要將 stack pointer 前的位置都要計算面積，而 stack pointer 後的位置則就不計算。

- **OUT:**

最後花一個 cycle 將計算好的面積吐出，並回到 LOAD 繼續接收下一個 pattern 資料。

這次的作業實作起來其實蠻有趣的，因為有很多演算法的架構可以進行優化，每個人也都有不同的做法，因此在比 PA 時就可以很明確看出自己的優勢在哪裡，其實我的設計中應該也還是有很多優化的空間，像是排序法可以使用較少 cycle 的排序法，不用像 bubble sort 一樣每次都重新比，但我這次作業重頭到尾都使用一個 buffer 操作以及共用所有的 pointer 我認為是不錯的做法。