

Homework 4: Max-Priority Queue

NAME		盧尚毅					
Student ID		N26124264					
Simulation Result							
Functional simulation	100	Gate-level simulation	100	Clock width	20	Gate-level simulation time	10360
<pre>VSIM2> run -all # ***** # ** # ** Congratulations !! ** # ** Simulation PASS !! ** # ** Your score =100 ** # ** # ***** # ** Note: sfinish : D:/intelFPGA20.1.1/diclab/DIC2024_hw4/testfixture.v(157) # ** Time: 3080 ns Iteration: 0 Instance: /test</pre>				<pre>VSIM6> run -all # ***** # ** # ** Congratulations !! ** # ** Simulation PASS !! ** # ** Your score =100 ** # ** # ***** # ** Note: sfinish : D:/intelFPGA20.1.1/diclab/DIC2024_hw4/testfixture.v(157) # ** Time: 3087626 ps Iteration: 0 Instance: /test # 1 # Break in Module test at D:/intelFPGA20.1.1/diclab/DIC2024_hw4/testfixture.v line 157</pre>			
Synthesis Result							
Total logic elements				1125			
Total memory bit				0			
Embedded multiplier 9-bit element				0			
Flow Status		Successful - Sun May 26 12:28:57 2024					
Quartus Prime Version		20.1.1 Build 720 11/11/2020 SJ Lite Edition					
Revision Name		MPQ					
Top-level Entity Name		MPQ					
Family		Cyclone IV E					
Device		EP4CE55F23A7					
Timing Models		Final					
Total logic elements		1,125 / 55,856 (2 %)					
Total registers		172					
Total pins		50 / 325 (15 %)					
Total virtual pins		0					
Total memory bits		0 / 2,396,160 (0 %)					
Embedded Multiplier 9-bit elements		0 / 308 (0 %)					
Total PLLs		0 / 4 (0 %)					
Description of your design							

在我的設計中共有 12 個狀態，首先是 idle，在 rst 後會進入 idle 狀態，此狀態用來判斷 cmd 決定下一個要進入的狀態，在 data_valid=1 時，idle 會進入 receive 狀態，將輸入的資料儲存進 queue 中，並記錄共有幾個數字被存入，接著有 build_queue 狀態，此狀態為建立 queue 的初始狀態，將 queue 中有幾個數字送入 heapify 狀態，heapify 的狀態會被分為三個部分，分別為檢察 largest 是否為左，檢察 largest 是否為右，最後交換與處理繼續 heapify 的 index，如果目前的 index 已經為 largest，則將 busy 變 0，並接收 cmd 跳至下一狀態，extract_max 狀態就是將 queue 中的最後一個數字與第一個交換，將 queue number-1 後再跳至 heapify 狀態，increase 狀態會先判斷 index 與 value，因為在此次作業中，可以將 insert_value 的數值直接接上 increase_value 的狀態對 queue 進行操作，接著判斷是否需要進入 swap 決定下一狀態，若不用更換數值，則回到 idle 接著讀取 cmd 繼續執行其他動作，swap 狀態則是將數值交換並將 index/2，最後是 write 狀態，會將 index 與在 queue 中的數值輸出，當 queue 總數為 0，進入 write_done 狀態，並拉起 done 訊號，表示執行完成。

*Scoring = (Total logic elements + total memory bit + 9*embedded multiplier 9-bit element) × (Total cycle used*clock width)*