

2024 Digital IC Design

Homework 4: Max-Priority Queue

NAME		洪裕翔					
Student ID		P76124215					
Simulation Result							
Functional simulation	P0: 100	Gate-level simulation	P0: 100	Clock width	18 (ns)	Gate-level simulation time	P0: 1,124,828
	P1: 100		P1: 100				P1: 2,096,828
	P2: 100		P2: 100				P2: 2,906,828
	P3: 100		P3: 100				P3: 4,454,828
							(ns)
<p>P0:</p> <pre># ***** # ** # ** Congratulations !! # ** # ** Simulation PASS !! # ** # ** Your score =100 # ** # ***** # ** Note: \$finish : testfixture.v(157) # Time: 1117 ns Iteration: 0 Instance: /test</pre>				<p>P0:</p> <pre># ***** # ** # ** Congratulations !! # ** # ** Simulation PASS !! # ** # ** Your score =100 # ** # ***** # ** Note: \$finish : testfixture.v(157) # Time: 1124828 ps Iteration: 0 Instance: /test</pre>			
<p>P1:</p> <pre># ***** # ** # ** Congratulations !! # ** # ** Simulation PASS !! # ** # ** Your score =100 # ** # ***** # ** Note: \$finish : testfixture.v(157) # Time: 2089 ns Iteration: 0 Instance: /test</pre>				<p>P1:</p> <pre># ***** # ** # ** Congratulations !! # ** # ** Simulation PASS !! # ** # ** Your score =100 # ** # ***** # ** Note: \$finish : testfixture.v(157) # Time: 2096828 ps Iteration: 0 Instance: /test</pre>			
<p>P2:</p> <pre># ***** # ** # ** Congratulations !! # ** # ** Simulation PASS !! # ** # ** Your score =100 # ** # ***** # ** Note: \$finish : testfixture.v(157) # Time: 2899 ns Iteration: 0 Instance: /test</pre>				<p>P2:</p> <pre># ***** # ** # ** Congratulations !! # ** # ** Simulation PASS !! # ** # ** Your score =100 # ** # ***** # ** Note: \$finish : testfixture.v(157) # Time: 2906828 ps Iteration: 0 Instance: /test</pre>			
<p>P3:</p> <pre># ***** # ** # ** Congratulations !! # ** # ** Simulation PASS !! # ** # ** Your score =100 # ** # ***** # ** Note: \$finish : testfixture.v(157) # Time: 4447 ns Iteration: 0 Instance: /test</pre>				<p>P3:</p> <pre># ***** # ** # ** Congratulations !! # ** # ** Simulation PASS !! # ** # ** Your score =100 # ** # ***** # ** Note: \$finish : testfixture.v(157) # Time: 4454828 ps Iteration: 0 Instance: /test</pre>			
Synthesis Result							
Total logic elements				1,969 / 55,856 (4%)			
Total memory bit				0 / 2,396,160 (0%)			
Embedded multiplier 9-bit element				0 / 308 (0%)			

Flow Status	Successful - Sun May 26 15:25:16 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	1,969 / 55,856 (4 %)
Total registers	317
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

在本次作業中，我以 9 個狀態實作有限狀態機。其中，IDLE、READ、WRITE 的功能和過去幾次作業差不多，分別為初始化暫存器、依照 testbench 之設計讀取、輸出資料，此處略過詳細說明。

較為特別的是，本次作業要求實作 BUILD_QUEUE、MAX_HEAPIFY、EXTRACT_MAX、INCREASE_VALUE 以及 INSERT_DATA 等功能，除了 MAX_HEAPIFY 我以 2 個狀態（MH_COMPARE、MH_RECURSE）分開實作之外，其餘功能皆以同名狀態實作。值得注意的是，助教提供的演算法虛擬碼是以範圍 [1, 資料數量] 實作的，而我自己則是以 [0, 資料數量-1] 範圍實作。

在 BUILD_QUEUE 功能中，我要讓系統運作「(資料數量-1)/2」次 MAX_HEAPIFY，以確認所有資料皆滿足 MAX HEAP 的規則。因此，這個功能的狀態會不斷在 BUILD_QUEUE 與 MAX_HEAPIFY 之間轉移，直到計算了「(資料數量-1)/2」次 MAX_HEAPIFY。

在 MAX_HEAPIFY 功能中，基本上就是根據建立 MAX HEAP 的演算法實作而已，此處略過詳細說明。這個功能的狀態會不斷在 MH_COMPARE 與 MH_RECURSE 之間轉移，直到當前檢查的資料已是子樹的最大值才結束。

在 EXTRACT_MAX 功能中，會把樹的根節點以樹的最後一個葉節點取代，再對整棵樹調整，使其符合 MAX HEAP 之規則。這個功能的狀態轉移會是 EXTRACT_MAX -> BUILD_QUEUE，而 BUILD_QUEUE 則會再呼叫多次 MAX_HEAPIFY（如前所述）。

在 INCREASE_VALUE 與 INSERT_DATA 功能中，兩功能分別會「把指定節點的值以給定新數值取代」、「在樹的最後一個葉節點加入一個新節點」，再調整整棵樹，使其符合 MAX HEAP 之規則。此二功能的狀態轉移和 EXTRACT_MAX 類似，僅差在要將 EXTRACT_MAX 換為「INCREASE_VALUE」或「INSERT_DATA」而已。

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