2024 Digital IC Design

Homework 4: Max-Priority Queue

NAME	陳了	育政							
Student II	1094198								
			Simulat	tion Resi	ult				
Functional	100	Gate-level	100	Clock	30 ns	Gate-level	3543 ns		
simulation	100	simulation		width		simulation time			
VSIM 3> run -all * ******************************* * ** Congratulations !! ** * ** Simulation PASS !! ** / 0.0 * **				VSIM 38> run -all *************************** ***					
			Synthe	sis Resu	lt				
Total logic elements				15848					
Total memory bit				0					
	Embedded multiplier 9-bit element					0			
	d multiplie	r 9-bit eleme	/11t	U					
Embedded	d multiplie vision Name		.IIt	MPQ		.,,=			
Embedded			ont			.,,			
Embedded	vision Name		ant	MPQ	/ E	.,,			
Embedded Re To Fa	vision Name		ont.	MPQ MPQ		.,,			
Embedded Ree To Fa De	vision Name p-level Entit mily	y Name	int.	MPQ MPQ Cyclone IV		.,,			
Embedded Re To Fa De Tir	vision Name p-level Entit mily vice	ty Name	in	MPQ MPQ Cyclone IV EP4CE55F	-23A7	28 %)			
Embedded Ree To Fa Dee Tir To	vision Name p-level Entit mily evice ming Models	y Name	int	MPQ MPQ Cyclone I\ EP4CE55F	-23A7	28 %)			
Embedded Ree To Fa Dee Tir To	vision Name op-level Entit mily evice ming Models otal logic eler	y Name	int	MPQ MPQ Cyclone IV EP4CE55F Final 15,848 / 5	- 55,856 (2	28 %)			
Re To Fa De Tir To To To	evision Name op-level Entit mily evice ming Models otal logic eler	ty Name	int	MPQ MPQ Cyclone IV EP4CE55F Final 15,848 / 5	- 55,856 (2	28 %)			
Embedded Re To Fa De Tir To To To	evision Name op-level Entit mily evice ming Models otal logic eler otal pins	y Name ments		MPQ MPQ Cyclone IV EP4CE55F Final 15,848 / 5 2106 50 / 325 (F23A7 55,856 (?				
Embedded Re To Fa De Tir To To To To	evision Name op-level Entit mily evice ming Models otal logic eler otal registers otal pins otal virtual pir	y Name ments		MPQ MPQ Cyclone IV EP4CE55F Final 15,848 / 5 2106 50 / 325 (55,856 (3 15 %)				

Description of your design

這次的作業一開始我是直接把 BUILD QUEUE 寫在同一個 state,但是在 合成的時候發現似乎是 BUILD QUEUE 的邏輯太複雜,會讓合成的時間拉長 很久,而且出來的 critical path 也會變長,因此後來我把 BUILD QUEUE 分 成 BUILD QUEUE 1 和 BUILD QUEUE 2 兩部分實作。BUILD QUEUE 1 是決定 largest 的 index,而 BUILD QUEUE 2 則更新 QUEUE 的排序,這樣 也的確大幅降低合成所需的時間。

至於最初 register 的宣告,我是在各個 state 各自宣告 register,但後來我 發現整個 module 中的單一個 state 最多只需要 3 個,因此便使用教授上課教

過的 resource sharing 技巧把 register 共用,而我需要做的就只是在進入 state 之前,根據 state 所需先更新 register 的值。

在 Post-Simulation 的時候,我最初也沒有通過,而 RAM_D 的輸出結果都是 0,但是在 Pre-Simulation 又可以正常通過。後來我才發現不知為何我沒在 reset 階段初始化 RAM_A 和 RAM_D 會致使 Quartus 在合成的時候有一個警告,就是 output stuck at VCC and GND,後果是合成的電路會讓 RAM_A 和 RAM_D 永遠卡在 0 或 1,於是我把 RAM_A 和 RAM_D 初始化後,Post-Simulation 的結果便成功通過了。

在實作 Max-Priority Queue 的演算法,我都是以助教提供的 pseudo-code 為主,而 EXTRACT_MAX 和 INSERT_DATA 因為在 pseudo-code 分別是直接呼叫 BUILD_QUEUE 和 INCREASE_VALUE,於是在 Verilog 實作中,我是直接讓 currState 跳轉到對應的 state,而非重複寫一次同樣的 code。

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