

2023 Digital IC Design Homework 4

NAME	蘇勇達		
Student ID	N96114603		
Simulation Result			
Functional simulation	60	Gate-level simulation	20
<pre>VSIM 21> run -all # ----- # # START!!! Simulation Start # # ----- # # Layer 0 output is correct ! # Layer 1 output is correct! # # ----- # # ----- S U M M A R Y ----- # # Congratulations! Layer 0 data have been generated successfully! # # Congratulations! Layer 1 data have been generated successfully! # # terminate at 64519 cycle # ----- # # ** Note: \$finish : D:/DICdesign/HW4/file/testfixture.v(178) # Time: 3225950 ns Iteration: 0 Instance: /testfixture # 1 # Break in Module testfixture at D:/DICdesign/HW4/file/testfixture</pre>		<pre>VSIM 4> run -all # ----- # # START!!! Simulation Start # # ----- # # Layer 0 output is correct ! # Layer 1 output is correct! # # ----- # # ----- S U M M A R Y ----- # # Congratulations! Layer 0 data have been generated successfully! # # Congratulations! Layer 1 data have been generated successfully! # # terminate at 64519 cycle # ----- # # ** Note: \$finish : D:/DICdesign/HW4/file/testfixture.v(178) # Time: 3225959351 ps Iteration: 0 Instance: /testfixture #</pre>	
Synthesis Result			
Total logic elements	836		
Total memory bits	0		
Embedded multiplier 9-bit elements	0		
Total cycle used	64519		
Compilation Report - ATCONV			
Flow Summary			
<<Filter>>			
Flow Status		Successful - Sun May 21 19:09:05 2023	
Quartus Prime Version		20.1.1 Build 720 11/11/2020 SJ Lite Edition	
Revision Name		ATCONV	
Top-level Entity Name		ATCONV	
Family		Cyclone IV E	
Device		EP4CE55F23A7	
Timing Models		Final	
Total logic elements		836 / 55,856 (1 %)	
Total registers		208	
Total pins		82 / 325 (25 %)	
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 %)	

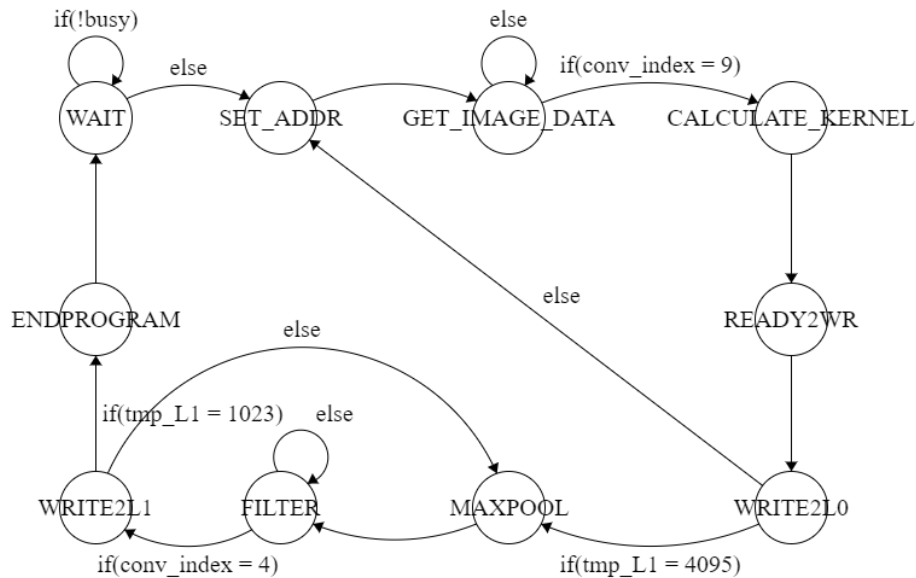
Description of your design

Notice

我的 testbench end cycle 有修改:

``define End_CYCLE 100000000`

Here is my FSM graph:



我總共用了 10 個 states 來完成我的電路，總結來說，我在第二個狀態把我要取的 9 個 address 放到一個 array，然後到另一個 state，花費 9 個 cycle 把所有的 idata 取出。之後再做其餘的運算。

至於 address 判斷的部分，我使用行和列來做實際 address 的運算，一般情況時，每個 address 都是(列*64+行)，所以在取得 address 方面非常方便。

在 L0 前，我計算完 kernel 的卷積之後，判斷第 12 個 bit 是不是 1 來決定是否需要 ReLU。

而

這次比較困難的地方是處理圖像邊界條件的方式，最一開始我使用了每個特例分開處理，造成我程式的 logic elements 無法降低。

$$\text{Scoring} = (836 + 0 + 0) * 64519 = 53937884$$

$$\text{Scoring} = (\text{Total logic elements} + \text{Total memory bits} + 9 * \text{Embedded multipliers 9-bit elements}) * \text{Total cycle used}$$

*** Total logic elements must not exceed 1000.**