

## Homework 4: Edge-Based Line Average interpolation

NAME	何立洋						
Student ID	Q36101066						
Simulation Result							
Functional simulation	Pass	Gate-level simulation	Pass	Clock width	64475(ns)	Gate-level simulation time	64239(ns)
<pre># # #  START!!! Simulation Start ..... # #-----S U M M A R Y----- # # #  Congratulations! # #  Result image data are generated successfully! # #  The result is PASS!!! # #-----S U M M A R Y----- # #  Note: \$finish      : C:/Users/HLY/Desktop/Dic/HW4/file/testfixture.v(176) #  Time: 64475 ns    Iteration: 0   Instance: /TB_ELA # #  Break in Module TB_ELA at C:/Users/HLY/Desktop/Dic/HW4/file/testfixture.v line 176</pre>				<pre>-----S U M M A R Y----- # # #  Congratulations! # #  Result image data are generated successfully! # #  The result is PASS!!! # #-----S U M M A R Y----- # #  Note: \$finish      : C:/Users/HLY/Desktop/Dic/HW4/file/testfixture.v(176) #  Time: 64239999 ps  Iteration: 0   Instance: /TB_ELA</pre>			
Synthesis Result							
Total logic elements				2210/68416(3%)			
Total memory bit				0/1152000(0%)			
Embedded multiplier 9-bit element				0/300(0%)			
Flow Summary							
Flow Status		Successful - Sun May 22 12:10:28 2022					
Quartus II 64-Bit Version		13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition					
Revision Name		ELA					
Top-level Entity Name		ELA					
Family		Cyclone II					
Device		EP2C70F896C8					
Timing Models		Final					
Total logic elements		2,210 / 68,416 ( 3 % )					
Total combinational functions		1,925 / 68,416 ( 3 % )					
Dedicated logic registers		832 / 68,416 ( 1 % )					
Total registers		832					
Total pins		39 / 622 ( 6 % )					
Total virtual pins		0					
Total memory bits		0 / 1,152,000 ( 0 % )					
Embedded Multiplier 9-bit elements		0 / 300 ( 0 % )					
Total PLLs		0 / 4 ( 0 % )					
Description of your design							

這次作業我的方法是將 `in_data` 依序在電路處理後，再存進 `result image memory`，因此，沒有使用到 `addr`。

#### 程式描述:

基本上還是使用上兩次作業的骨幹，循序電路設四個 `state`。分別對應:

`State0`:reset 時，不做存和算的動作，但預先準備計數值。

`State1`:讀取 `in_data`。我將其分為上行和下行(第一筆對應第一行和第三行)，

`State2`:計算中行。除了計數值，其他部分寫在組合電路，包括資料比較後 `D1,D2,D3` 給值。

`State3`:存入 `data_wr` 和準備新的行。將上行和中行存入 `data_wr` 後，將下行存入上行，之後回到 `State1`，此時，讀取 `in_data` 僅存入下行。

為了方便，我設置需多的計數，作業遇到比較大的問題是在 `State2` 比較數值時，經常會延一個 `clk`。

#### 感想:

這次作業沒有上次那麼複雜，有可能是因為我的做法是屬於比較簡單的那種，也因此我的 `performance` 比較普通，但這是目前的我想到的最直接、最直觀的方法。這段時間我對 `verilog` 漸漸熟悉了，但我在寫程式時，還是比較容易從撰寫軟體的方式去思考。

#### 改進:

我發現我用 `datamem_middle[caculate-1]`，似乎會花費相當多時間，我在撰寫時都沒有注意到這個問題。之後做類似的儲存動作時，我應該會改用 `shfit`。

*Scoring = (Total logic elements + total memory bit + 9\*embedded multiplier 9-bit element) × (longest gate-level simulation time in ns)*