2021 Digital IC Design Homework 3

NAME	黃柏瑄
Student ID	P78081528

Simulation Result

Functional simulation	All Pass	Gate-level simulation	All Pass	Gate-level simulation time	simulation time (ns) 49750

```
# Object45: PASS
#
# Object46: PASS
#
# Object47: PASS
#
# Object48: PASS
#
# Object49: PASS
#
# Object50: PASS
#
# ------------------------------------------------
# --    Simulation finish,  ALL PASS            --
# ------------------------------------------------
# ** Note: $finish    : D:/hsuan/PhD/109_2/DIC/DIC_hw
#    Time: 49750 ns  Iteration: 1  Instance: /testfix
# 1
# Break in Module testfixture at D:/hsuan/PhD/109_2/D
```

(your pre-sim result)

```
#
# Object45: PASS
#
# Object46: PASS
#
# Object47: PASS
#
# Object48: PASS
#
# Object49: PASS
#
# Object50: PASS
#
# ------------------------------------------------
# --    Simulation finish,  ALL PASS            --
# ------------------------------------------------
# ** Note: $finish    : D:/hsuan/PhD/109_2/DIC/DIC_hw
#    Time: 49750 ns  Iteration: 1  Instance: /testfix
# 1
# Break in Module testfixture at D:/hsuan/PhD/109_2/D
```

(your post-sim result)

Synthesis Result

Total logic elements	532
Total memory bit	0
Embedded multiplier 9-bit element	4
Clock width (Cycle)	50.0

Flow Summary

Flow Status	Successful - Fri May 14 11:17:22 2021
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition
Revision Name	PSE
Top-level Entity Name	PSE
Family	Cyclone II
Device	EP2C70F896C8
Timing Models	Final
Total logic elements	532 / 68,416 (< 1 %)
Total combinational functions	531 / 68,416 (< 1 %)
Dedicated logic registers	139 / 68,416 (< 1 %)
Total registers	139
Total pins	46 / 622 (7 %)
Total virtual pins	0
Total memory bits	0 / 1,152,000 (0 %)
Embedded Multiplier 9-bit elements	4 / 300 (1 %)
Total PLLs	0 / 4 (0 %)

Description of your design

為了達到速度與硬體大小的平衡，本作業採用兩階段氣泡排序，靈感來源來自：https://ithelp.ithome.com.tw/articles/10195078，不過本作業較為複雜因為要排序的個數是動態的。我首先觀察輸入點個數及對應的排序次數，當大於 3 個點時，需要排 n-1 次 (因為第一點 (0) 不需要排)，當 n 恰為 3 時只要排一次即可所以要特殊處理。兩階段排序代表一次從奇數一次從偶數，例如 n 為 5 時，第一次排序比較兩組數 (1,2)(3,4)，第二次比較 (2,3)，第三次 (1,2)(3,4)，最後 (2,3) 即可完成。原先有考慮將向量存在另外的區塊，後來發現不需要且浪費空間，在比較時即時計算即可。

本作業中用了許多 counter 使得每一個 clock 都有做事，雖然高效率，不過不好寫。雖然作業限制不可修改 Clock width，但是我自行實測可支援到 37.9 (Gate-level simulation time: 37710.5 ns)。

另外有發現使用 B 的 Total logic elements 數量會比 A 少 200 多，因此我採用 B。

A:

```
if (cross_compare_result) begin
    px[i  ] <= px[i+1];
    px[i+1] <= px[i  ];
    py[i  ] <= py[i+1];
    py[i+1] <= py[i  ];
end
else begin
    px[i  ] <= px[i  ];
    px[i+1] <= px[i+1];
    py[i  ] <= py[i  ];
    py[i+1] <= py[i+1];
end
```

B:

```
px[i  ] <= (cross_compare_result) ? px[i+1] : px[i  ];
px[i+1] <= (cross_compare_result) ? px[i  ] : px[i+1];
py[i  ] <= (cross_compare_result) ? py[i+1] : py[i  ];
py[i+1] <= (cross_compare_result) ? py[i  ] : py[i+1];
```

Scoring = Total logic elements