LMFE Report

一、數據表

Area(um^2)	211,122.153271
Clock cycle(ns)	10
Total simulation time(ns)	1,590,541
Cost(um^2*ns)	335,798,440,786

二、合成結果

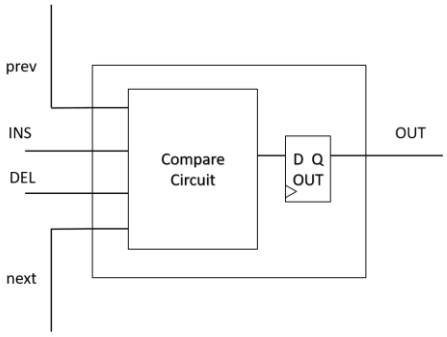
```
****************************************
Report : area
Design : LMFE
Version: G-2012.06
Date   : Sat Apr 21 10:39:51 2018
****************************************

Library(s) Used:

    slow (File: /home/raid7_2/course/cvsd/CBDK_IC_Contest/CIC/SynopsysDC/db/slow.db)
    sram_1024x8_t13 (File: /home/raid7_2/userb03/b03095/CVSD/HW3_ST/Memory/sram_1024x8_t13/sram_1024x8_t13_slow_syn.db)

Number of ports:                         21
Number of nets:                        5875
Number of cells:                       5201
Number of combinational cells:         4625
Number of sequential cells:             562
Number of macros:                         0
Number of buf/inv:                      686
Number of references:                   208

Combinational area:       109804.805363
Noncombinational area:    101317.347908
Net Interconnect area:    1907929.576508

Total cell area:          211122.153271
Total area:               2119051.729779
1
design_vision>
```

```
Output pixel: 0 ~ 15000 are correct!

Output pixel: 0 ~ 16000 are correct!

--------------------------------------------------

Congratulations! All data have been generated successfully!

-------------------------PASS-------------------------

Simulation complete via $finish(1) at time 1590541 NS + 0
./testfixture1.v:135       #(`CYCLE/2); $finish;
ncsim> exit
[b03095@cad29 HW3_ST]$
```

```
clock clk (rise edge)                    10.00      10.00
clock network delay (ideal)               0.50      10.50
clock uncertainty                        -0.10      10.40
R1_addr_reg_8_/CK (DFFRX1)                0.00      10.40 r
library setup time                       -0.23      10.17
data required time                                  10.17
-----------------------------------------------------------
data required time                                  10.17
data arrival time                                  -10.17
-----------------------------------------------------------
slack (MET)                                         0.00
```

三、架構實現方法

A. Median Filter



設計 Comparator，有四個 input，prev 為前一個 comparator 的 output；next 為下一個 comparator 的 output；INS 為現在想要加入 filter 的值；DEL 為現在想要從 filter 刪去的值，所有的初始值皆為 ff。

依 INS、DEL 的大小分成兩種情況討論：

1. INS<DEL，此時要變動的是 out value 介於 INS 和 DEL 之間的 comparators (INS < out < DEL)，若 prev > INS，則 out_write 的值為 prev，等同於此時的值往後面的 comparator 移動；若 prev <= INS，則 out_write 的值為 INS，代表在這個點插入想要的值。想要 DEL 的值會因為最後一個變動的 comparator 繼承了 prev 而被刪去。
2. INS>DEL，此時要變動 out 介於 DEL 和 INS 之間的 comparators (DEL< out < INS)，若 next < INS，等同於此時的值往前面的 comparator 移動；若 next >= INS，則 out_write 的值為 INS，代表在這個點插入想要的值。想要 DEL 的值會因為變動的第一個 comparator 繼承了 next 而被刪去。

因此，利用 49 個 comparator 串接，輸出第 24 個 comparator 的 out，即可達到 median filter 的效果

B. SRAM 使用

使用了一個 1024*8bit 的 SRAM，共紀錄 7x128x8bit 的值。

C. 控制電路

主要的實作方式為從最左上角的 pixel 開始，紀錄一組(x,y)，代表現在正在計算哪一點的 median；另外，使用一個 7x7 的 mask 紀錄現在 median filter 中的值，這樣在下一個點就不用全部重新讀取 49 個值計算，只要更新左右的 column 就可(INS 右邊的 column，DEL 左邊的 column)；最後，利用不同的 state 控制邊界條件。

以下是各個 state 的說明：

1、 INITIAL: 讀進前 7x128 格的值

2、 COMPUTE_B_*: U=upper, B=bottom, L=left, R=right，handle 不同的邊界條件，例如 COMPUTE_B_BL 即是負責計算左下角的 median values。

3、 COMPUTE: 負責 general case 的計算。

4、 UPDATE_M: 每次在計算 median 時，都將此時的 49 個 values 存到 7x7 的 mask 中，如此在計算右邊一格的點時，只需要讀入 7 個新的值，再刪去最左邊 column 的 7 個值，就可以得到新的 median，不需要全部重讀一次。

5、 UPDATE_R: 每計算完一個 row，先讀進下一個 row 共 128 個值，再進行下一排的計算。

6、 CLEARMASK: 每計算一個 row，必須先清掉此時 mask 中所有的值，等待計算下一個 row 時，再重新讀進 49 個值。

D. FSM