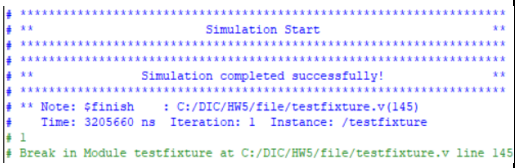
# 2023 Digital IC Design Homework 5

| NAME | 林珮玉 |
|---|---|
| Student ID | E24084096 |

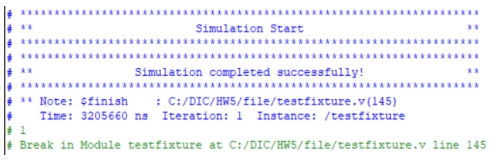
**Simulation Result**

| Functional simulation | Completed | Gate-level simulation | Completed |
|---|---|---|---|

(your functional sim result)




(your gate-level sim result)




**Evaluation Results**

| | | | |
|---|---|---|---|
| test1.png | 25.32 | test2.png | 24.82 |
| test3.png | 29.12 | test4.png | 20.95 |
| test5.png | 21.94 | test6.png | 25.21 |

**Description of your design**

這次我只有用三個 state: DATA_IN, BILINEAR, RESULT，以及 五個 reg: len (用來記錄目前所在的 pixel), count (用來計算需做 Bilinear 的 pixel 處理數量), red, green, blue (皆為用來儲存 Bilinear 的運算結果)。

DATA_IN 用來寫入進來的資料，當 in_en 時，開始從 128*128 pixel 的第一筆到最後一筆。讀進前會先判斷這個 len 是在哪一排 row 跟 column，如果 len[7]==1 且 len[0]==1 代表位於奇數的 row 跟奇數的 column，如果 len[7]==1 且 len[0]==0 代表所在的是奇數的 row 和偶數的 column，用此可判斷資料要讀進 r,g,b 哪一個 memory。當 len==16383 時，表示已讀到最後一筆，即可把 nextState 設為下個狀態 Bilinear。

因為 len 從 0 到 127 皆為邊界，而 128%與 128%127 也是邊界，不需執行 Bilinear，故直接從 len=129 開始執行。由於 Bilinear 如講義所說的，有四種 case，所以我們必須先用奇偶數判斷 len 在哪個位置，以決定要進入哪個 case。

Case1: 必在奇數 row 奇數 column，因為 missing B, R on G，所以先把 B,R 用 Bilinear 的方式寫到該 address。此時要注意的地方是，因 data 是負緣才 read 進來，要等得到下個 clock 才能讀值，所以在得到欲讀取的 B 和 R 之記憶體

位置之後，要等到下個 clock 才能把 rdata_r 和 rdata_b 讀進來。接著直接加到 red 和 blue，然後再讀一次值。當把兩個要做 Bilinear 的 rdata_r 和 rdata_b 的值都讀進來之後，直接在同個 clock 除 2，然後把 wr_r 和 wr_b 拉起來，存 red 和 blue。也就是說 red 與 rdata_r 相加後後馬上 >> 1，並直接存入 wdata_r 和 wdata_b。

隨後 len+1 進入下個 pixel，這個位置剛好就是 Case2。Case2 需要判讀左右上下的 green 以及左上右上左下右下的 red，讀完後個別除 4 再存進位置在 len 的 wdata_g, wdata_r。然後 len+1，進入 Case3。Case3 跟 Case1 一樣，只需做上下和左右各兩個顏色再除 1，做完後存入數值並進入 Case4。Case4 與 Case2 作法雷同。需讀取兩中顏色的 4 個所在位置的資料，讀完並累加再除 4 後，len+1。四個 Case 一直輪流做下去，等到 len 抵達倒數到二行倒

數第二個時，因後面都不需要做 Bilinear，故跳掉最後一個 State: RESULT。在這個 State 只需做一件事，就是把 done 拉為 1。這樣一來就完成整張圖的 Demosaic 處理啦。

*Scoring = average PSNR of the six test images*

*** PSNR of all interpolation results should meet at least the baseline.**