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.**