**113學年度**

**國立中山大學**

**課程名稱:硬體描述語言**

**作業/成果報告/專題**

**授課教師: 蕭勝夫教授**

**學生學號/班級/姓名: B103040009 資工系大四 尹信淳**

**題目 HW4 Sobel Edge Detector**

**架構圖**

**一張含有 文字, 圖表, 螢幕擷取畫面, 地圖 的圖片

自動產生的描述**

* 架構說明

基本上是照著上方的架構圖實作的，但有幾個稍微不同之處:

1. Intensity Converter的部分，改成:

assign Y = (306 \* R + 601 \* G + 117 \* B) >> 10;

306 約等於 0.299 x 210

601 約等於 0.587 x 210

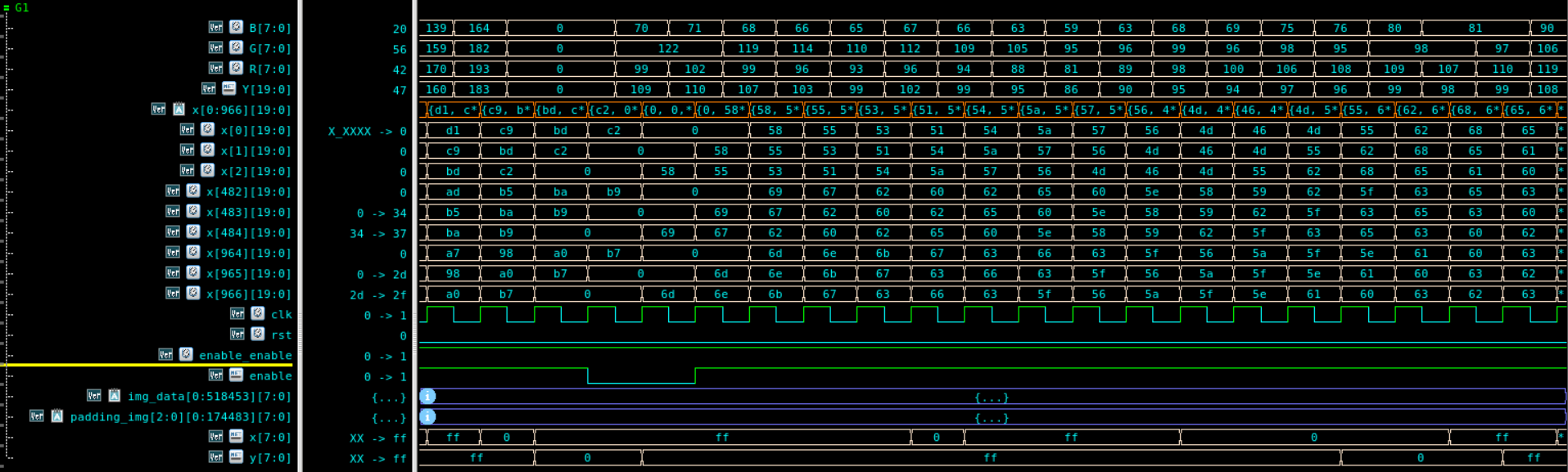
117 約等於 0.114 x 210

相加後結果再右移10位回來即可

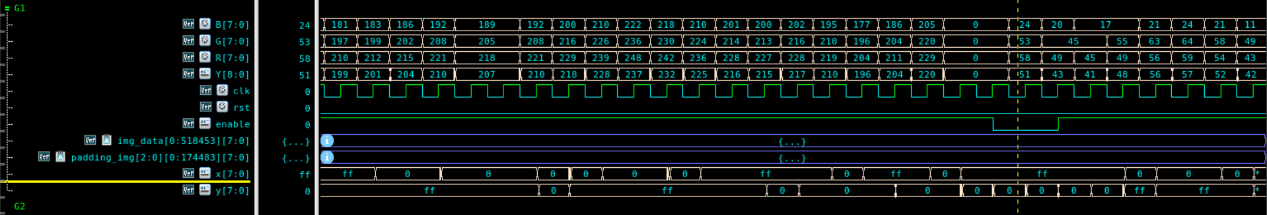
1. Line Buffer下方處理sliding window切割的部分，我加了一個訊號來開關enable，遇到sliding window被切割時，enable會被忽略。同時也藉這個機制在前966個shift register的平移，讓enable被忽略。

**模擬波型圖**

* RTL level



* Gate level (area optimized)



* 波型解釋

事先做了zero padding，因此sliding window被換行切割時，window的內容會是這兩種情況:

1. column1有data，column2 & 3 會全是padded zero
2. column1 & 2 全是padded zero，column3有data

(sliding window: x[0] x[1] x[2]

x[482] x[483] x[484]

x[964] x[965] x[966] )

上方RTL-level波型可以看出，這兩種情況時，enable訊號是0，代表這時的x, y不會被寫入輸出的bmp檔。

下方Gate-level找不到sliding window，大概是被design compiler置換成其他東西了，但enable還是有變0的地方，所以應該沒錯。

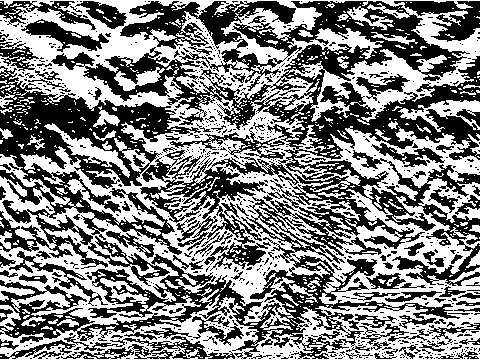
**數據表格**

|  |  |  |  |
| --- | --- | --- | --- |
|  | area | delay | Power |
| Area optimized | 23049.723697μm2 | 1.16ns | 37.6490 mW |

**原圖**

****

**水平邊緣**



**垂直邊緣**

****

**FPGA波型圖**

* Behavioral

一張含有 螢幕擷取畫面 的圖片

自動產生的描述

* Post-implementation

一張含有 螢幕擷取畫面, 行, 電路 的圖片

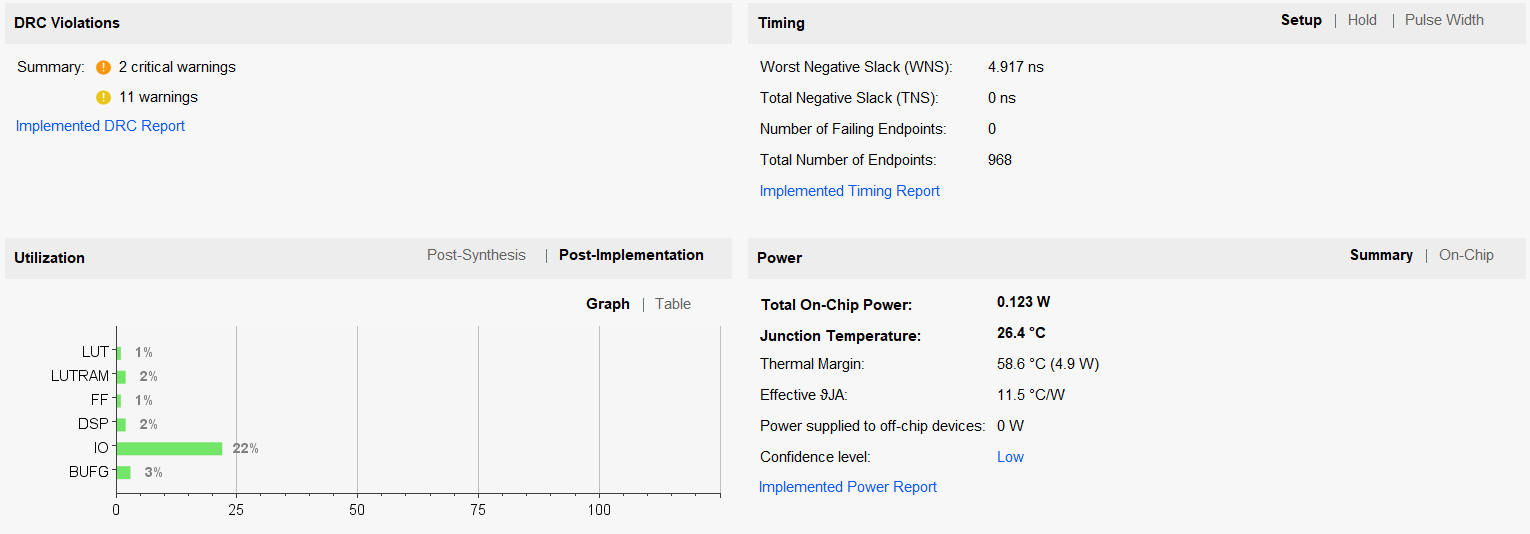
自動產生的描述

* 波型解釋

途中可以看到RGB連續出現兩個0(當時的Y也會是0)，這是我們的zero padding，第一個0時，enable是1，因為這時sliding window是正常的，但是到第二個0時，enable變0，因為這時sliding window被切割，並且enable等於0有再持續了一個posedge clk，因此兩個被切割的情況都被處理。

**FPGA project summary overview snapshot**

**一張含有 文字, 螢幕擷取畫面, 字型, 數字 的圖片

自動產生的描述**

**心得**

這次作業很有趣，算是第一次從頭開始寫一個比較大的verilog design。一開始line buffer那邊邏輯有錯，輸出到bmp檔的byte少了幾個，bmp檔就直接開不起來。最後看到輸出的兩張結果圖還是挺開心的，只是邊緣好像沒有被辨識的很犀利? 有很多曲線什麼的。我個人感覺作業檔案裡面的這張比較好看哈哈。

一張含有 貓, 黑與白 的圖片

自動產生的描述