

## R06943087 葉玲瑜

Area( $\mu\text{m}^2$ )	37402.20851
Clock cycle(ns)	4.4
Total simulation time(ns)	10908.716
Cost( $\mu\text{m}^2 \cdot \text{ns}$ )	$4.45 \times 10^{12}$

Area (total cell area) in Design compiler

```

Number of ports: 45
Number of nets: 2696
Number of cells: 2310
Number of combinational cells: 1886
Number of sequential cells: 423
Number of macros/black boxes: 1
Number of buf/inv: 284
Number of references: 152

Combinational area: 20939.126308
Buf/Inv area: 2819.381398
Noncombinational area: 13643.700802
Macro/Black Box area: 69557.296875
Net Interconnect area: 284294.781555

Total cell area: 104140.123985
Total area: 388434.905540

```

## RTL 以及 gate-level ncverilog simulation 截圖

[illegible][illegible]

### 3. 架構實現方法

我分成兩個部分做處理，第一部分負責從rom讀data，判斷是否有相連，並且寫到sram裡；第二部分則將第一部分寫到sram的data再次從sram中讀出來，將可以合併的部分重新把新的值寫進sram裡。

第一部分：

我主要是利用兩個32bit \* 16bit的register(color1, color2)來完成這部分，首先當 (read == 1) 時將從rom裡讀取到的data存到color2裡，經過4個cycle後會得到32筆data，這時候 (read == 0) 然後進行是否相連的判斷：

當遇到下圖這四種情形時，判斷為相連；因為具有對稱性，所以可以判斷上半部即可。

1	0	0
0	1	0
0	0	0

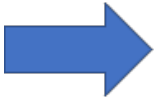
0	1	0
0	1	0
0	0	0

0	0	1
0	1	0
0	0	0

0	0	0
1	1	0
0	0	0

而當判斷為相連的時候，會根據「左—上—右上—左上」的優先順序填色，例如：若遇到下圖情形，會優先判斷將目標（紅字1）填入與左邊相同的數值（填入4），並且記錄  $3 = 4$  在color\_table的register裡，表示兩者其實是相同的前景物件，以利於第二部分的判斷。做完32筆data的判斷後，將read訊號改為1，進行新的一排rom data的讀取，然後把color2的data shift到color1 register裡，當做上排資料的判斷依據。

0	3	0
4	1	0
0	0	0



0	3	0
4	4	0
0	0	0

第二部分：

重新將sram裡的data讀出來，當讀到需要合併的值（例如在color\_table裡有紀錄  $3 = 4$ ，所以讀到3的時候就要重新將pixel的值改寫成4），就重新將新的值寫進sram裡，經過  $1024 + n$  個cycle後（ $n =$  需要重新寫值的data數量），完成第二部分。