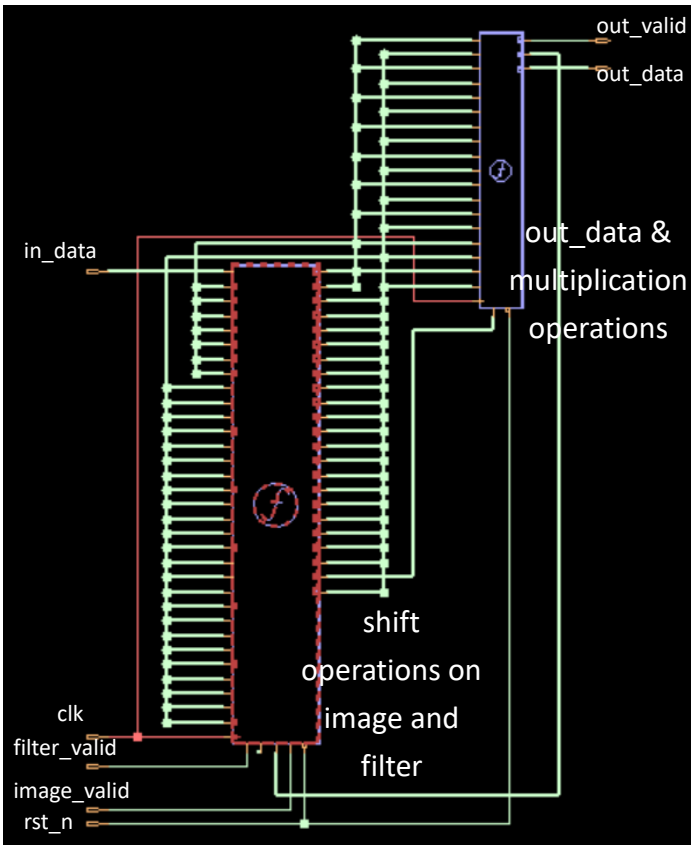
Block Diagram



02 乘法器

Hierarchical cell	Global cell area: Absolute Total	Percent Total	Local cell area: Combi-national	Noncombi-national	Black-boxes	Design
conv	38662.7473	100.0	10275.2496	16269.4222	0.0000	conv
add_0_root_add_0_root_add_111_8	811.6416	2.1	811.6416	0.0000	0.0000	conv_DW01_add_0
add_1_root_add_0_root_add_111_8	811.6416	2.1	811.6416	0.0000	0.0000	conv_DW01_add_1
add_2_root_add_0_root_add_111_8	788.3568	2.0	788.3568	0.0000	0.0000	conv_DW01_add_4
add_7_root_add_111_8	635.3424	1.6	635.3424	0.0000	0.0000	conv_DW01_add_7
mult_111	1007.8992	2.6	1007.8992	0.0000	0.0000	conv_DW_mult_tc_8
mult_111_2	1007.8992	2.6	1007.8992	0.0000	0.0000	conv_DW_mult_tc_7
mult_111_3	1007.8992	2.6	1007.8992	0.0000	0.0000	conv_DW_mult_tc_6
mult_111_4	1007.8992	2.6	1007.8992	0.0000	0.0000	conv_DW_mult_tc_5
mult_111_5	1007.8992	2.6	1007.8992	0.0000	0.0000	conv_DW_mult_tc_4
mult_111_6	1007.8992	2.6	1007.8992	0.0000	0.0000	conv_DW_mult_tc_3
mult_111_7	1007.8992	2.6	1007.8992	0.0000	0.0000	conv_DW_mult_tc_2
mult_111_8	1007.8992	2.6	1007.8992	0.0000	0.0000	conv_DW_mult_tc_1
mult_111_9	1007.8992	2.6	1007.8992	0.0000	0.0000	conv_DW_mult_tc_0
Total			22393.3251	16269.4222	0.0000	

最終的設計我用到了 9 個乘法器，和 filter 的格子數一樣 (對 filter 中每個數字做乘法)，而可從這個 synthesis report 中看到，一個乘法器的面積約為 1000，占了 combinational area 很大的一部分。

遇到困難與如何解決/設計方法

我的設計共做了三次，而以下會大致介紹每一次的概念、方法，以及針對一、二的改良:

- 設計一:

我用了最笨也是最圖法煉鋼的方式，就是直接在 case 判斷裏頭做各項的乘法，出來的面積大得驚人，這是我第一次寫出六位數子的設計！而之所以會有這樣的情況應該是因為利用太多的乘法器，如上可知，一個乘法器面積約為 1000，若以這樣的寫法，做乘法的面積就大約 1000*25*9 = 225000，可想而知為甚麼會有那麼大的電路。

```
case(out_count)
    0: begin
    out_data <= filter[0][0] * image[0][0]+ filter[0][1] * image[0][1] + filter[0][2] * image[0][2]+ filter[1][0] * image[1][
    filter[1][1] * image[1][1]+ filter[1][2] * image[1][2]+ filter[2][0] * image[2][0]+ filter[2][1] * image[2][1]+ filter[2]
    image[2][2];
    end
    1: begin
    out_data <= filter[0][0] * image[0][1]+ filter[0][1] * image[0][2] + filter[0][2] * image[0][3]+ filter[1][0] * image[1][
    filter[1][1] * image[1][2]+ filter[1][2] * image[1][3]+ filter[2][0] * image[2][1]+ filter[2][1] * image[2][2]+ filter[2]
    image[2][3];
    end
    2: begin
    out_data <= filter[0][0] * image[0][2]+ filter[0][1] * image[0][3] + filter[0][2] * image[0][4]+ filter[1][0] * image[1][
    filter[1][1] * image[1][3]+ filter[1][2] * image[1][4]+ filter[2][0] * image[2][2]+ filter[2][1] * image[2][3]+ filter[2]
    image[2][4];
    end
    3: begin
    out_data <= filter[0][0] * image[0][3]+ filter[0][1] * image[0][4] + filter[0][2] * image[0][5]+ filter[1][0] * image[1][
    filter[1][1] * image[1][4]+ filter[1][2] * image[1][5]+ filter[2][0] * image[2][3]+ filter[2][1] * image[2][4]+ filter[2]
    image[2][5];
    end
    4: begin
    out_data <= filter[0][0] * image[0][4]+ filter[0][1] * image[0][5] + filter[0][2] * image[0][6]+ filter[1][0] * image[1][
```

```
Combinational area:            424694.800130
Buf/Inv area:                   34847.367605
Noncombinational area:          19851.955589
Macro/Black Box area:               0.000000
Net Interconnect area:      undefined  (No wire load specified)

Total cell area:               444546.755719
Total area:                 undefined
```

- 設計二:

我首先對乘法的面積進行改良，把乘法拉出 case 外面來做，另外設 9 個 reg，而用 case 判斷甚麼時候把哪 9 個數字存進這些 reg 和 filter 進行乘法加總，這次的面積大概 6 萬多，而面積都主要為 combinational。

- 設計三(最終設計):

這次的改良我針對 case 的條件判斷進行修正，因為 combinational 的面積依然很大，所以我直接對 image 做 shift 來代替 25 個的條件判斷，而當下被 shift 到乘法位置的那 9 個數字就和 filter 做乘法，一樣是使用 9 個乘法器，能夠再減了一些面積。結果如下:

```
296
297   Combinational area:              22393.325099
298   Buf/Inv area:                     1187.524843
299   Noncombinational area:           16269.422226
300   Macro/Black Box area:                0.000000
301   Net Interconnect area:      undefined  (No wire load specified)
302
303   Total cell area:                 38662.747325
304   Total area:                 undefined
```

心得

這次的作業讓我真正體會到乘法器面積的龐大，將來在做設計的時候，若能降低運算面積改為較簡單的 operation 像是這次的 shift 就要盡量使用，避免像我這次第一次設計出這麼恐怖的電路。所以從這次的作業我學到了: 要做出良好的設計，一定要對實質電路有相當的概念和認知，像是常用到的 blocks (multiplier, adder, shift, mux 等) 大概都會合成出多大的面積。