
CS5120 Homework Assignment 02

Student ID: 107062612

Name: 熊祖玲

1. Design Concept

如下圖一的 finite state machine 所示。

I. 當 RST_N 訊號由 1 變 0 時，將所有訊號做初始化，並且進入 IDLE 狀態

II. IDLE 狀態

i. START 為 0: 停留在 IDLE 階段，並將所有訊號設為初始值。

ii. START 為 1: 將輸入訊號 (A、B 的前 8 個 bits) 放到暫存器 (A_hi、B_hi) 中，並且進入 IDLE2 狀態。

III. IDLE2 狀態

i. START 為 0: 若 A_all、B_all 其中一個為 0 時，將 next_error 設為 1，並且進入下一個狀態 (CALC)。

ii. START 為 1: 將輸入訊號 (A、B 的後 8 個 bits) 與暫存器 (A_hi、B_hi) 串接成 A_all、B_all，並停留在 IDLE2 狀態。

IV. CALC 狀態

i. ERROR 為 1 (當 clock 由 0 變 1 時，會將 next_error 的值送到 ERROR): 將 next_diff 設為 0，進入 FINISH 階段。

ii. A_all、B_all 相等: 將 next_diff 設為 A_all，進入 FINISH 階段。

iii. $A_{all} > B_{all}$: 進行 GCD 運算 ($next_A_{all} = A_{all} - B_{all}$, $next_B_{all} = B_{all}$, $next_diff = A_{all} - B_{all}$)。

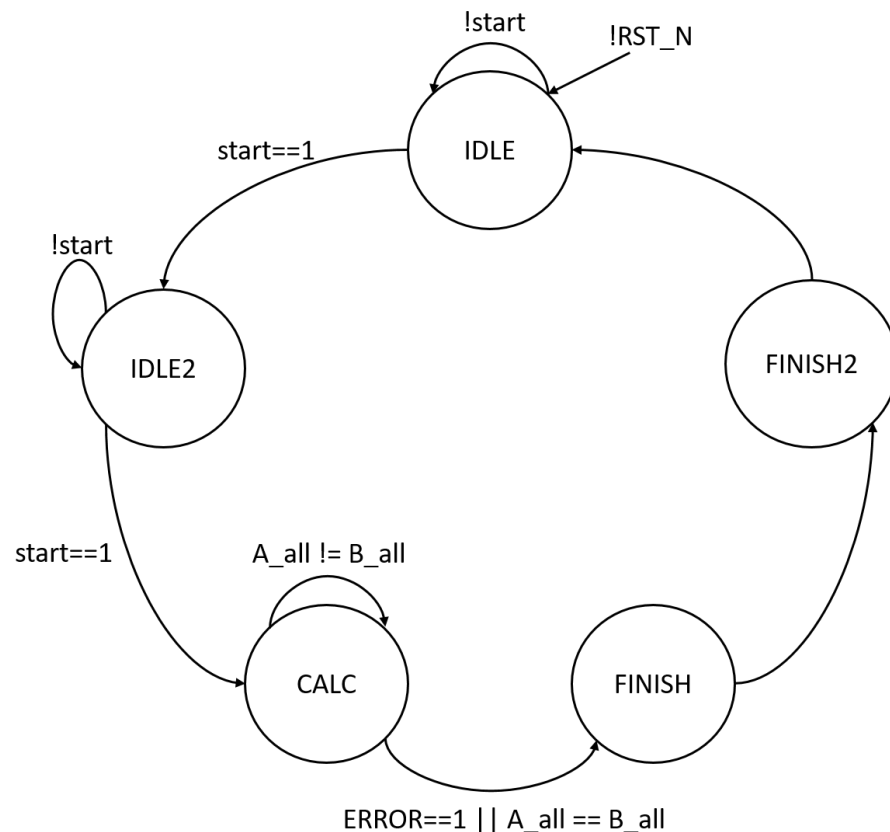
iv. $A_{all} < B_{all}$: 進行 GCD 運算 ($next_A_{all} = A_{all}$, $next_B_{all} = B_{all} - A_{all}$, $next_diff = B_{all} - A_{all}$)。

V. FINISH 狀態

將 next_DONE 設為 1，並將 output 訊號 (next_Y) 設為 diff (當 clock 由 0 變 1 時，會將 next_diff 的值送到 diff) 的前 8 個 bits，進入 FINISH2 階段。

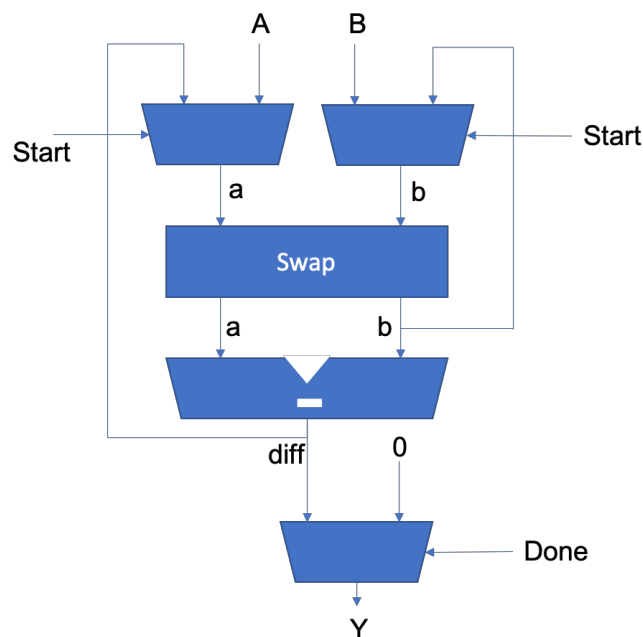
VI. FINISH2 狀態

將 output 訊號 (next_Y) 設為 diff 的後 8 個 bits，進入 IDLE 階段。



圖一、Finite State Machine 圖

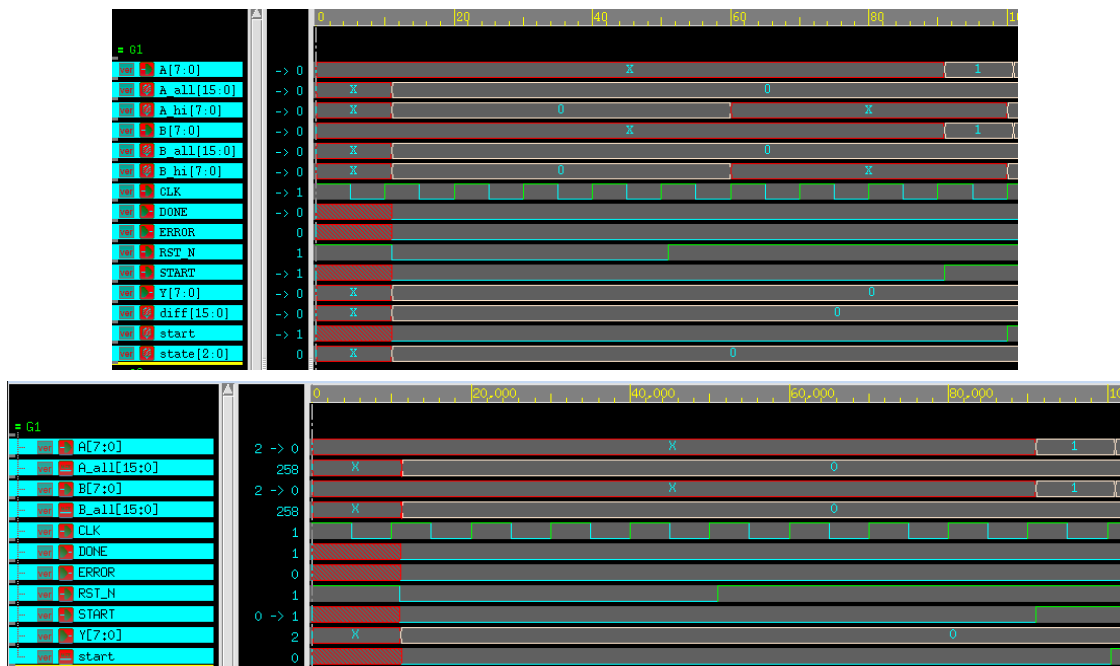
以下介紹 GCD 的計算方式 (如圖二所示): 首先, 當 START 訊號變成 1 時, 將輸入訊號 A、B 放到暫存器 a、b 中, 接著比較 a、b 的大小, 較大者於下一個 clock cycle 設為 a, 較小者則設為 b, 之後計算出兩數的差值 diff (a-b), 並於下一個 clock cycle 將 a 設為 diff, 再重複執行比大小、計算差值, 當 a、b 相等時, 表示計算已完成, 則 $\text{gcd}(A, B) = a = b$ 。



圖二、block diagram

2. Simulation and Discussion

當 RST_N 訊號為 0 時，將訊號初始化。在 RTL 和 gate-level 的模擬中波形是一致的。

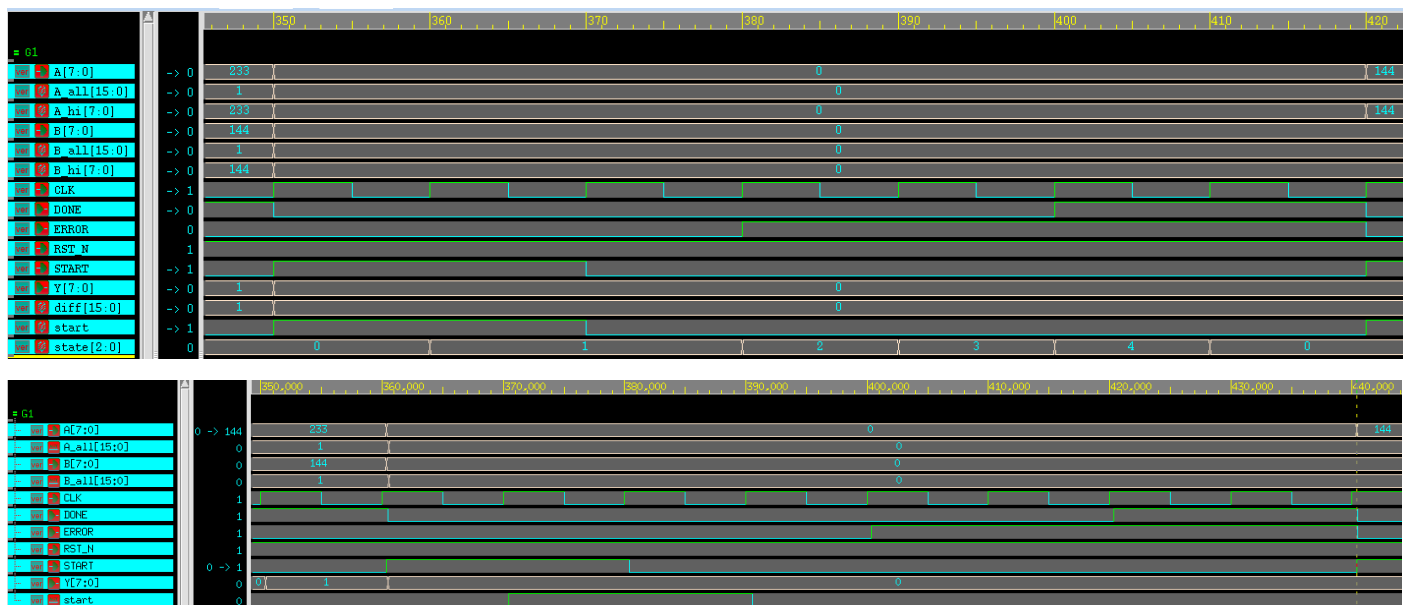


圖三、初始化之波形圖(上圖為 RTL，下圖為 gate-level)

[Testcase1] A=0, B=0 (波形圖如圖四所示)

目的：驗證當其中一個訊號為 0 時是否正確

當 START 為 1 時，A_hi、B_hi 有正確儲存 A、B 訊號，並且進入 IDLE2 狀態，A_all、B_all 也有得到將新的 A、B 訊號做串接，於下一個 cycle 得到 ERROR（因為 A、B 其中一個為 0，ERROR 也有正確的變為 1）再進入 CALC 狀態，偵測到 ERROR 為 1 後，亦正確進入 FINISH 狀態，先輸出答案的前 8 個 bits，同時把 DONE 訊號設為 1，進入 FINISH2 狀態後再輸出答案的後 8 個 bits，同時 ERROR、DONE 維持 1，最後回到 IDLE 狀態等待下次被啟動。在 RTL 和 gate-level 的模擬中波形是一致的。

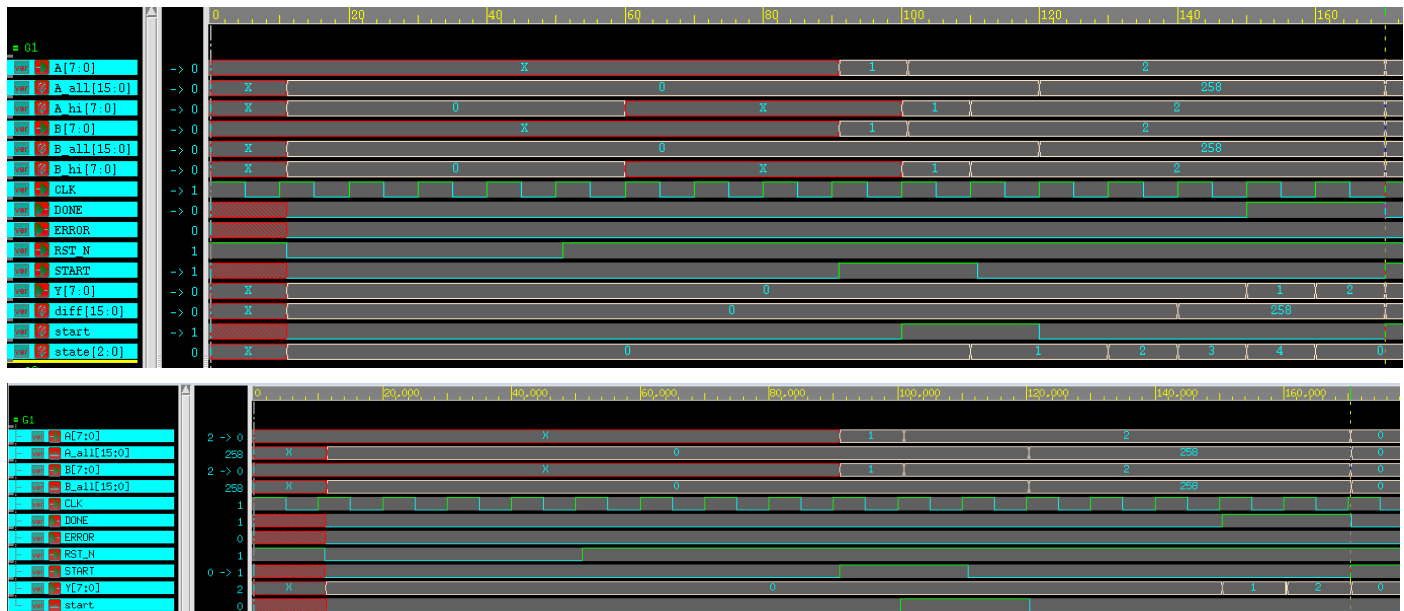


圖四、testcase 1 之波形圖(上圖為 RTL，下圖為 gate-level)

[Testcase2] A=258, B=258 (波形圖如圖五所示)

目的：驗證當兩個訊號相同時是否正確

當 START 為 1 時，A_hi、B_hi 有正確儲存 A、B 訊號，並且進入 IDLE2 狀態，A_all、B_all 也有得到將新的 A、B 訊號做串接，於下一個 cycle 得到 ERROR=0（因為 A、B 皆不為 0）再進入 CALC 狀態，偵測到 A==B 後，亦正確進入 FINISH 狀態，先輸出答案的前 8 個 bits，同時把 DONE 訊號設為 1，進入 FINISH2 狀態後再輸出答案的後 8 個 bits，同時 DONE 維持 1，最後回到 IDLE 狀態等待下次被啟動。在 RTL 和 gate-level 的模擬中波形是一致的。

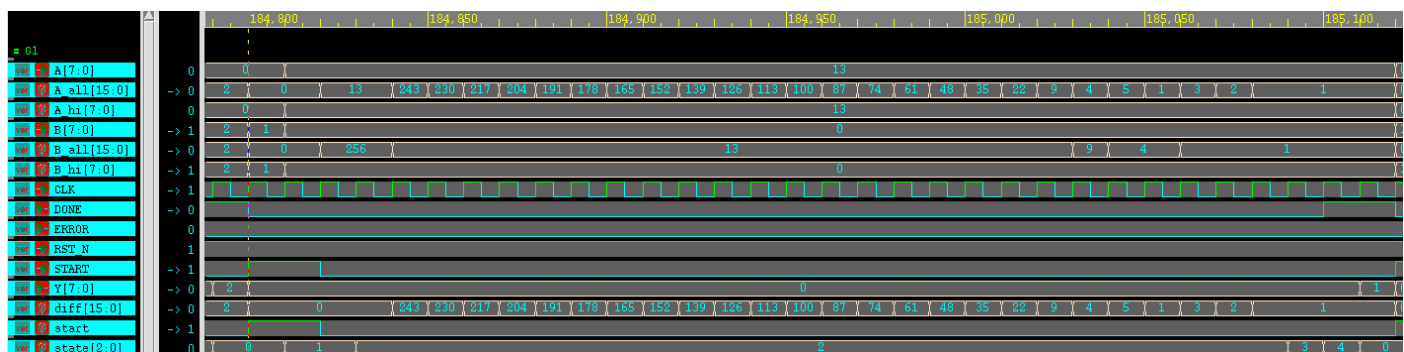


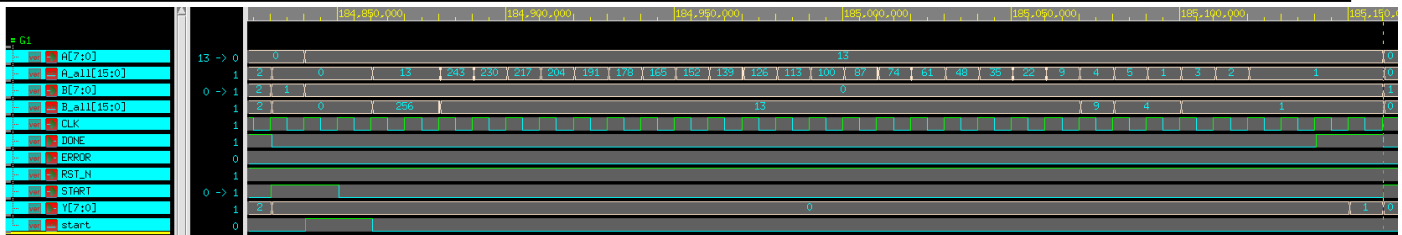
圖五、testcase 2 之波形圖(上圖為 RTL，下圖為 gate-level)

[Testcase3] A=256, B=13 (波形圖如圖六所示)

目的：驗證當兩數互質相同時是否正確

當 START 為 1 時，A_hi、B_hi 有正確儲存 A、B 訊號，並且進入 IDLE2 狀態，A_all、B_all 也有得到將新的 A、B 訊號做串接，於下一個 cycle 得到 ERROR=0（因為 A、B 皆不為 0）再進入 CALC 狀態，偵測到 A>B 後，進行 GCD 運算直到 A==B 後，正確進入 FINISH 狀態，先輸出答案的前 8 個 bits，同時把 DONE 訊號設為 1，進入 FINISH2 狀態後再輸出答案的後 8 個 bits，同時 DONE 維持 1，最後回到 IDLE 狀態等待下次被啟動。在 RTL 和 gate-level 的模擬中波形是一致的。



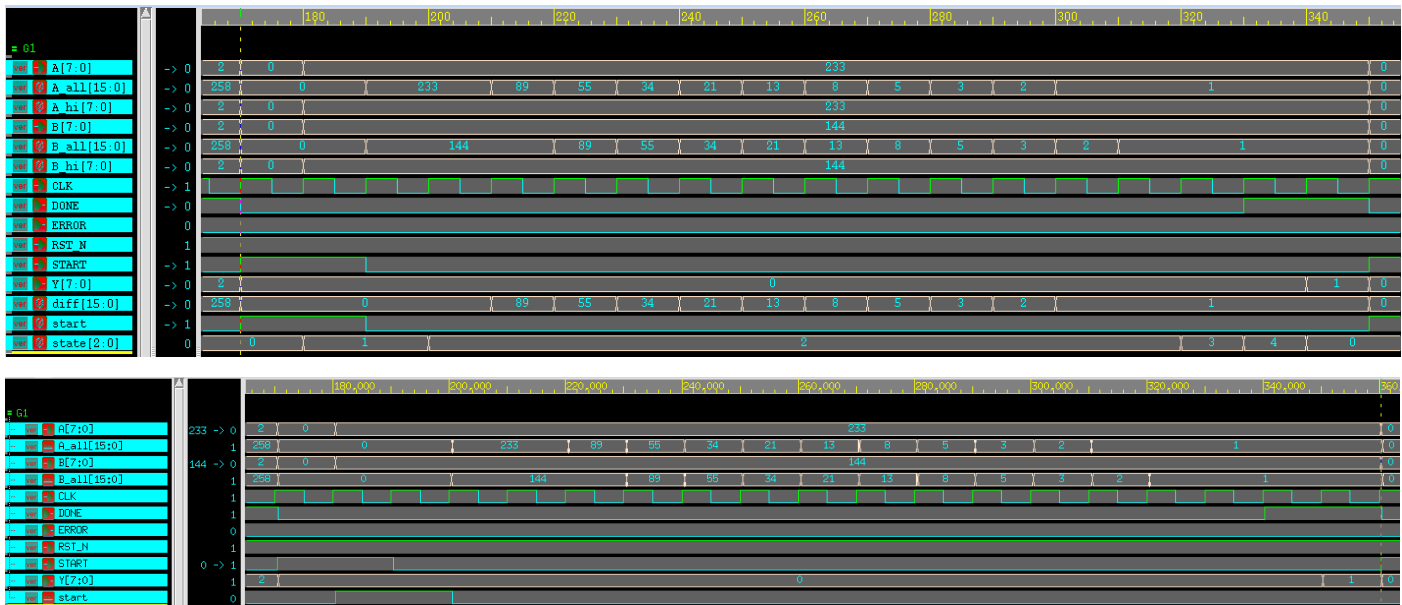


圖六、testcase 3 之波形圖(上圖為 RTL，下圖為 gate-level)

[Testcase4] A=233, B=144 (波形圖如圖七所示)

目的：驗證當輸入為兩個 fibonacci 數是否正確

當 START 為 1 時，A_hi、B_hi 有正確儲存 A、B 訊號，並且進入 IDLE2 狀態，A_all、B_all 也有得到將新的 A、B 訊號做串接，於下一個 cycle 得到 ERROR=0 (因為 A、B 皆不為 0) 再進入 CALC 狀態，偵測到 A>B 後，進行 GCD 運算直到 A==B 後，正確進入 FINISH 狀態，先輸出答案的前 8 個 bits，同時把 DONE 訊號設為 1，進入 FINISH2 狀態後再輸出答案的後 8 個 bits，同時 DONE 維持 1，最後回到 IDLE 狀態等待下次被啟動。在 RTL 和 gate-level 的模擬中波形是一致的。



圖七、testcase 4 之波形圖(上圖為 RTL，下圖為 gate-level)

因為兩個數相差為 1 時必定互質，所以在這個 design 中，可以透過判斷兩數的差值是否為 1，來提前終止 GCD 的運算，縮減運算的時間。

[Timing report]

如圖八所示，gate-level design 的 slack 大於零，代表我的 design 沒有太慢，在 clock 轉換前可以計算完畢，並且 slack 不是一個很大的值，代表我的 design 也沒有跑得太快的問題，不會在訊號被送出去前就換成下一個訊號。

45	r318/U2_15/Y (XOR3X1)	0.26	6.32	r
46	r318/DIFF[15] (GCD_DW01_sub_0)	0.00	6.32	r
47	U290/Y (CLKINVX1)	0.06	6.38	f
48	U288/Y (OAI221XL)	0.18	6.56	r
49	diff_reg[15]/D (DFFRX1)	0.00	6.56	r
50	data arrival time		6.56	
51				
52	clock CLK (rise edge)	10.00	10.00	
53	clock network delay (ideal)	0.00	10.00	
54	diff_reg[15]/CK (DFFRX1)	0.00	10.00	r
55	library setup time	-0.27	9.73	
56	data required time		9.73	
57	-----			
58	data required time		9.73	
59	data arrival time		-6.56	
60	-----			
61	slack (MET)		3.18	

圖八、Timing report 部分內容之截圖

[Power report]

整體看來，我的 design 的 power consumption 滿低的，不過沒有可以比較的對象，不確定跟其他同學比較會不會算是低的。

21	Power-specific unit information :				
22	Voltage Units = 1V				
23	Capacitance Units = 1.000000pf				
24	Time Units = 1ns				
25	Dynamic Power Units = 1mW (derived from V,C,T units)				
26	Leakage Power Units = 1pW				
27					
28					
29	Cell Internal Power = 208.1597 uW (96%)				
30	Net Switching Power = 8.3212 uW (4%)				
31	-----				
32	Total Dynamic Power = 216.4809 uW (100%)				
33					
34	Cell Leakage Power = 4.9974 uW				
35					
36					
37		Internal	Switching	Leakage	Total
38	Power Group	Power	Power	Power	Power (%) Attrs
39	-----				
40	io_pad	0.0000	0.0000	0.0000	0.0000 (0.00%)
41	memory	0.0000	0.0000	0.0000	0.0000 (0.00%)
42	black_box	0.0000	0.0000	0.0000	0.0000 (0.00%)
43	clock_network	0.0000	0.0000	0.0000	0.0000 (0.00%)
44	register	0.1976	2.8336e-03	2.7902e+06	0.2032 (91.76%)
45	sequential	0.0000	0.0000	0.0000	0.0000 (0.00%)
46	combinational	1.0564e-02	5.4876e-03	2.2071e+06	1.8259e-02 (8.24%)
47	-----				
48	Total	0.2082 mW	8.3212e-03 mW	4.9974e+06 pW	0.2215 mW
49					

圖九、Power report 部分內容之截圖

[Area report]

我的 design 的 $\text{area} = \frac{72 \times 3079.083627}{254.609996} \approx 870 \text{ units}$, 我認為可能有點多, 可以再做改善化簡整底的面積, 不過沒有可以比較的對象, 不確定跟其他同學比較會不會算是多的。

```
13  Number of ports:                168
14  Number of nets:                 618
15  Number of cells:                407
16  Number of combinational cells:  307
17  Number of sequential cells:     97
18  Number of macros/black boxes:   0
19  Number of buf/inv:              72
20  Number of references:           25
21
22  Combinational area:              3079.083627
23  Buf/Inv area:                   254.609996
24  Noncombinational area:           3031.556297
25  Macro/Black Box area:            0.000000
26  Net Interconnect area:           undefined (No wire load specified)
27
28  Total cell area:                 6110.639924
29  Total area:                     undefined
```

圖十、Area report 部分內容之截圖

3. Summary

這次作業遇到兩個問題, 第一個是 coding style 引起的, 因為不小心在程式當中寫出 latch, 在一般模擬的時候沒有出現問題, 但是在合成之後, 就會執行到一半突然出現震盪, 讀到一些不如預期的值; 第二個問題則是在 test bench 中發生的, 在 reset 時沒有將 START 訊號做初始化, 讓一開始的 state 出現 high z 或 unknown 的狀況, 在 debug 時還以為是 design 的問題, 所以一直無法解決, 幸好最後有老師的幫忙, 讓我可以如期地完成作業。