Computer-Aided Design for VLSI Design

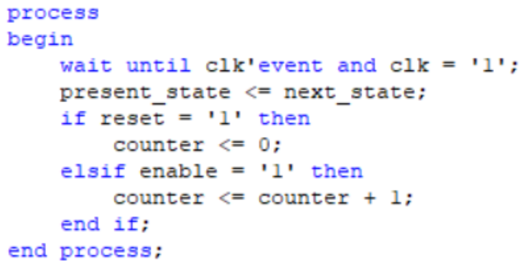
Homework3（Student ID: ntust_B10832019｜Name: 林琛琛）

1. Provide a simple explanation of your code.

```
process(present_state, complete)
    begin
        case present_state is
            when green1 =>
                g_light <= '1';
                y_light <= '0';
                r_light <= '0';
                reset <= '1';
                next_state <= green2;
            when green2 =>
                g_light <= '1';
                y_light <= '0';
                r_light <= '0';
                reset <= '0';
                enable <= '1';
                if complete = '1' then
                    next_state <= yellow1;
                else
                    next_state <= green2;
                end if;
            when red1 =>
                g_light <= '0';
                y_light <= '0';
                r_light <= '1';
                reset <= '1';
                next_state <= red2;
            when red2 =>
                g_light <= '0';
                y_light <= '0';
                r_light <= '1';
                reset <= '0';
                enable <= '1';
                if complete = '1' then
                    next_state <= yellow2;
                else
                    next_state <= red2;
                end if;
            when yellow1 =>
                g_light <= '0';
                y_light <= '1';
                r_light <= '0';
                enable <= '0';
                next_state <= red1;
            when yellow2 =>
                g_light <= '0';
                y_light <= '1';
                r_light <= '0';
                enable <= '0';
```

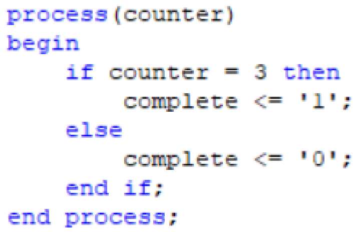
```
                                        next_state <= green1;
                        end case;
                end process;
```

這個 process 用來處理 state 之間的轉變，以及依照題目要求設定哪個燈亮、reset 與 enable 的設定。



```
process
begin
    wait until clk'event and clk = '1';
    present_state <= next_state;
    if reset = '1' then
        counter <= 0;
    elsif enable = '1' then
        counter <= counter + 1;
    end if;
end process;
```

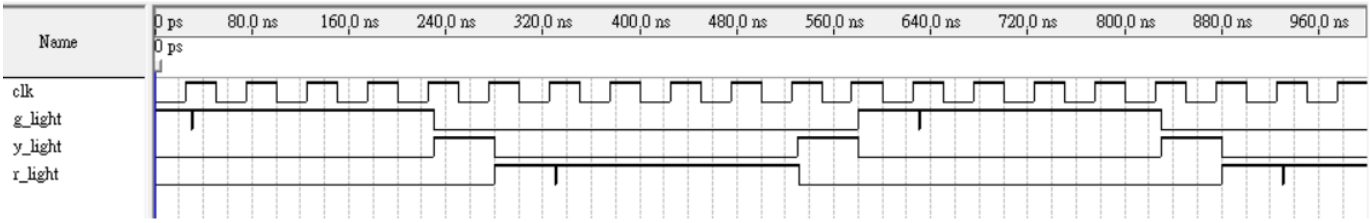
這個 process 用來更新 state、如果在 rising edge 時有需要 reset 或 enable 的話，就做相對應的處理。



```
process(counter)
begin
    if counter = 3 then
        complete <= '1';
    else
        complete <= '0';
    end if;
end process;
```

complete 的部分，每當 counter 數值有變化時，都會檢查當下是否 complete 並變更 complete 的值。我的程式設定 counter 的最大值是 3。

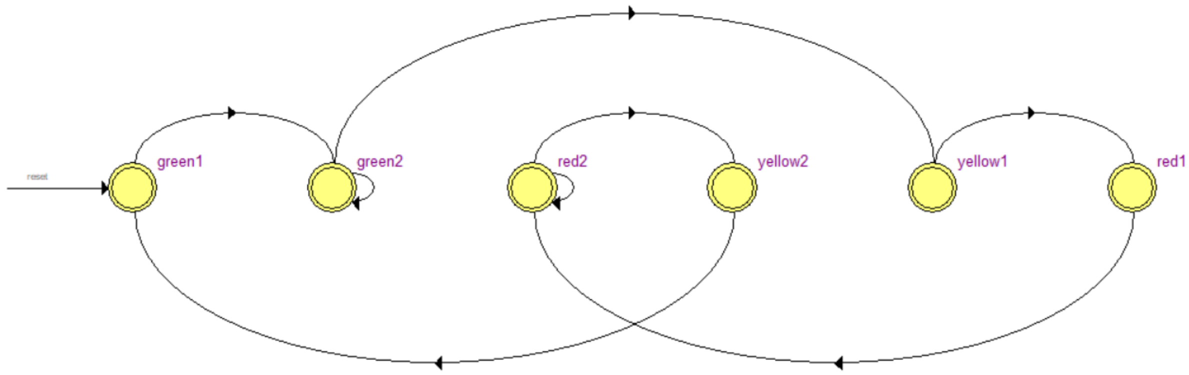
2. Waveform diagram here (Simulation Results)



3. Reflections and discussions

心得：
這次的作業類似於講義中 counter 加上紅綠燈複雜版的結合。撰寫程式的時候，卡最久的部分是要怎麼切割一個 process 中的內容。譬如檢查 complete 的部分，一開始有想過寫進更新 state 的那個 process。

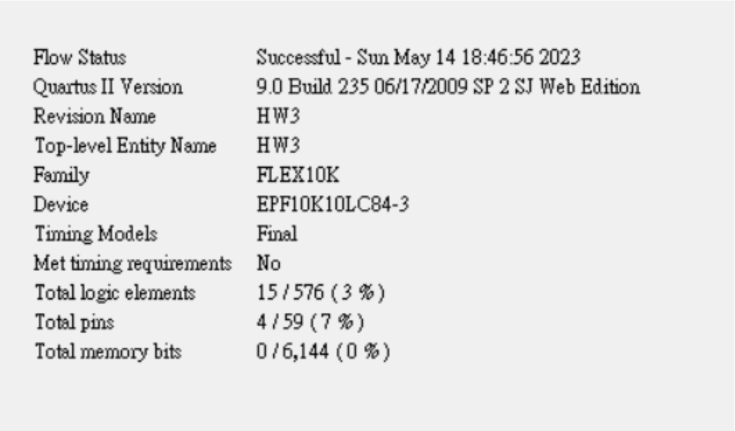
另外是寫的過程中，發現關於 reset 與 enable，作業說明只有說哪個 state 的時候要把它們設 1，但沒說什麼時候設回 0，因此也有思考說設回 0 的時機點，應該是要在下一個 state，還是要在 rising edge 判斷的那個 process 直接改。



最後檢查 state machine 的狀態圖，也與作業說明相同。

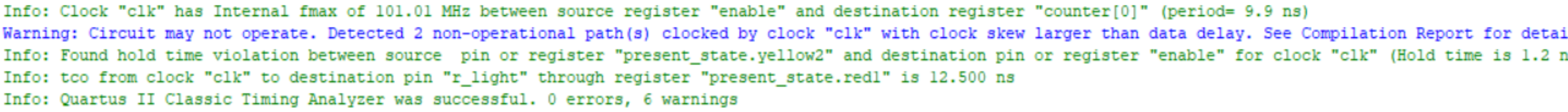
討論：
討論的部分主要是嘗試講義第七章的內容，以作業的程式來做比較。

- Family = FLEX10K



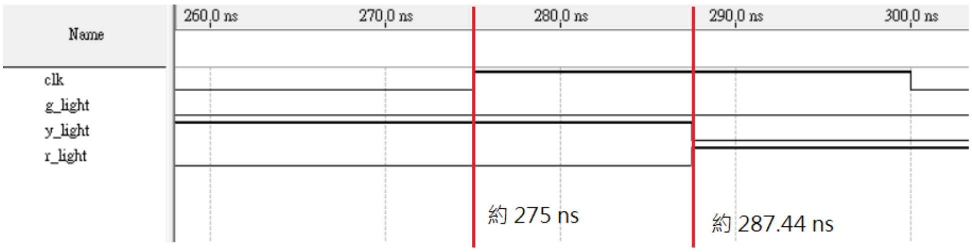
Flow Status	Successful - Sun May 14 18:46:56 2023
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Web Edition
Revision Name	HW3
Top-level Entity Name	HW3
Family	FLEX10K
Device	EPF10K10LC84-3
Timing Models	Final
Met timing requirements	No
Total logic elements	15 / 576 (3 %)
Total pins	4 / 59 (7 %)
Total memory bits	0 / 6,144 (0 %)

將 family 設為 FLEX10K、device 設為 EPF10K10LC，得到上面的圖。在 Area Performance 上，我的程式使用了 15 個 logic element。



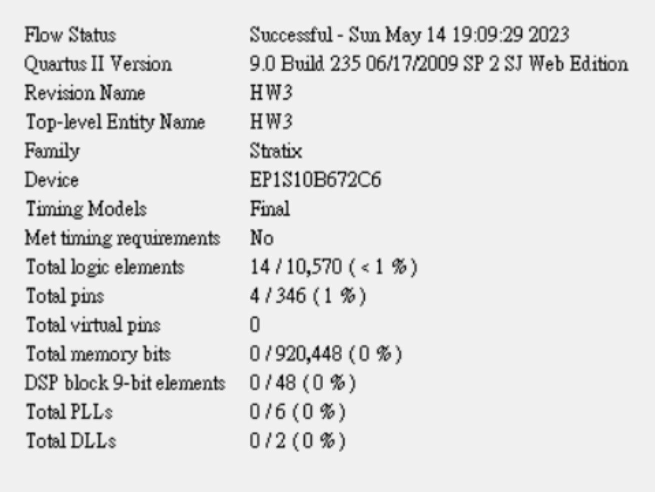
```
Info: Clock "clk" has Internal fmax of 101.01 MHz between source register "enable" and destination register "counter[0]" (period= 9.9 ns)
Warning: Circuit may not operate. Detected 2 non-operational path(s) clocked by clock "clk" with clock skew larger than data delay. See Compilation Report for detai
Info: Found hold time violation between source  pin or register "present_state.yellow2" and destination pin or register "enable" for clock "clk" (Hold time is 1.2 n
Info: tco from clock "clk" to destination pin "r_light" through register "present_state.red1" is 12.500 ns
Info: Quartus II Classic Timing Analyzer was successful. 0 errors, 6 warnings
```

其中 fmax = 101.01 MHz、worst-case tco = 12.5 ns。



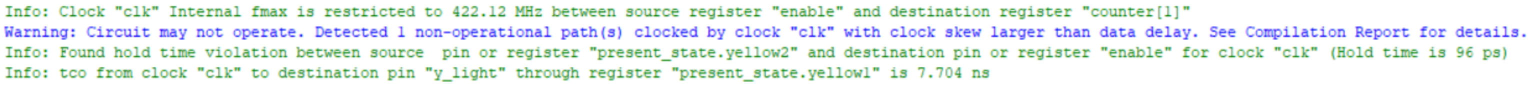
從 clock 敲下去到紅綠燈值變化的時間差，確實在 worst-case tco 的範圍內。

- Family = Straix



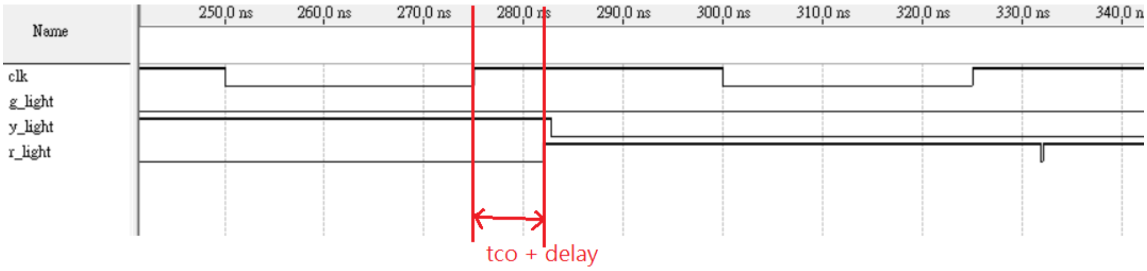
Flow Status	Successful - Sun May 14 19:09:29 2023
Quartus II Version	9.0 Build 235 06/17/2009 SP 2 SJ Web Edition
Revision Name	HW3
Top-level Entity Name	HW3
Family	Stratix
Device	EP1S10B672C6
Timing Models	Final
Met timing requirements	No
Total logic elements	14 / 10,570 (< 1 %)
Total pins	4 / 346 (1 %)
Total virtual pins	0
Total memory bits	0 / 920,448 (0 %)
DSP block 9-bit elements	0 / 48 (0 %)
Total PLLs	0 / 6 (0 %)
Total DLLs	0 / 2 (0 %)

將 family 設為 Straix，得到上面的圖，使用了 14 個 logic element。



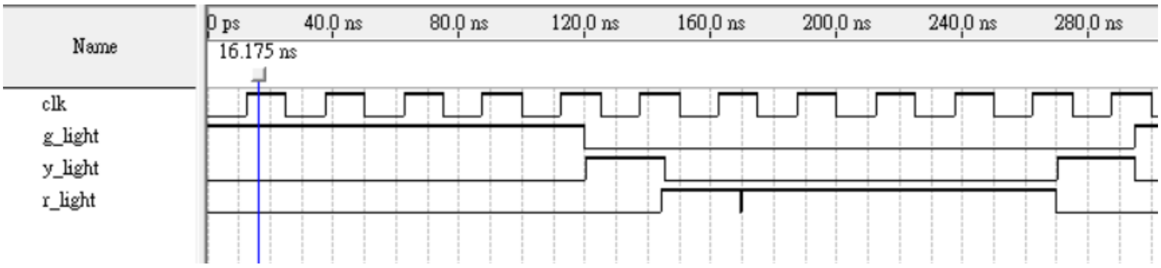
```
Info: Clock "clk" Internal fmax is restricted to 422.12 MHz between source register "enable" and destination register "counter[1]"
Warning: Circuit may not operate. Detected 1 non-operational path(s) clocked by clock "clk" with clock skew larger than data delay. See Compilation Report for details.
Info: Found hold time violation between source  pin or register "present_state.yellow2" and destination pin or register "enable" for clock "clk" (Hold time is 96 ps)
Info: tco from clock "clk" to destination pin "y_light" through register "present_state.yellow1" is 7.704 ns
```

其中 fmax = 422.12 MHz、tco = 7.704ns

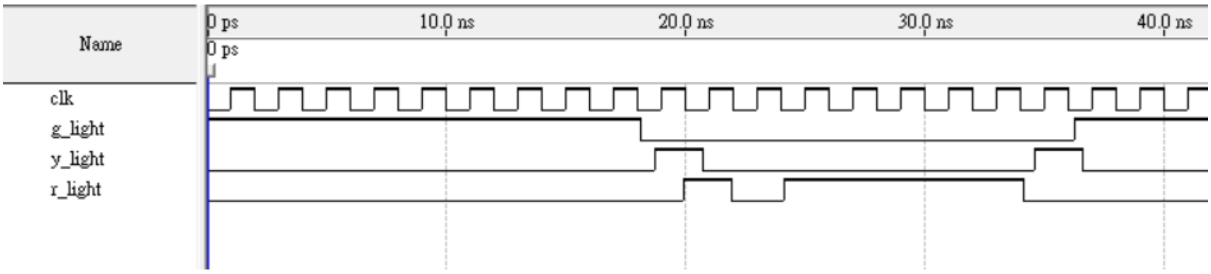


相比之下 worst-case tco 比較 family 為 FLEX10K 的時候小，從波型圖看來也是如此。

再來討論 fmax 與 clock rate 的關係。clock rate 必須小於 fmax，不然模擬出來的波型結果會有錯誤。原本的波型週期是 50 ns，相當於 clock rate = 20 MHz，模擬的結果正確。



當 clock period = 25 ns，也就是 clock rate = 40 MHz 時，模擬的結果也是正確的。

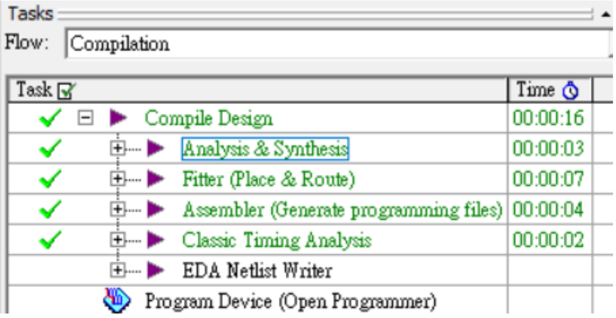


當 clock period = 2 ns，也就是 clock rate = 500 MHz 時，很明顯看到紅綠燈的變化情形有誤，clk 經過太多次 rising edge 之後才開始變燈，跟程式寫的 counter max value = 3 不符。

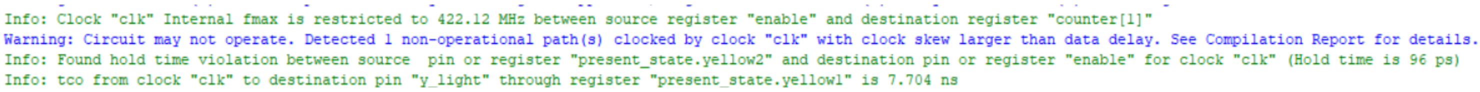
最後是比較選擇 fitter effort 中不同方法的影響。

- Standard Fit

fmax = 422 MHz, tco = .12 7.704 ns



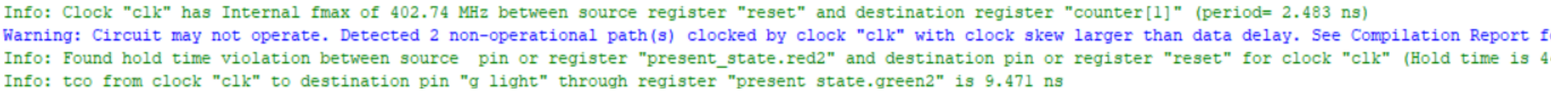
Task	Time
✓ ⊟ ▶ Compile Design	00:00:16
✓ ⊞ ▶ Analysis & Synthesis	00:00:03
✓ ⊞ ▶ Fitter (Place & Route)	00:00:07
✓ ⊞ ▶ Assembler (Generate programming files)	00:00:04
✓ ⊞ ▶ Classic Timing Analysis	00:00:02
⊞ ▶ EDA Netlist Writer	
Program Device (Open Programmer)	



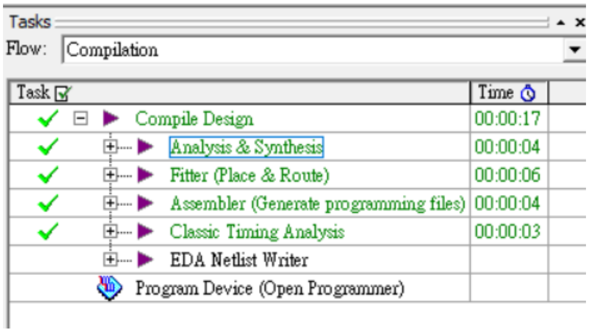
```
Info: Clock "clk" Internal fmax is restricted to 422.12 MHz between source register "enable" and destination register "counter[1]"
Warning: Circuit may not operate. Detected 1 non-operational path(s) clocked by clock "clk" with clock skew larger than data delay. See Compilation Report for details.
Info: Found hold time violation between source  pin or register "present_state.yellow2" and destination pin or register "enable" for clock "clk" (Hold time is 96 ps)
Info: tco from clock "clk" to destination pin "y_light" through register "present_state.yellow1" is 7.704 ns
```

- Fast Fit

fmax = 402.85, tco = 9.471



```
Info: Clock "clk" has Internal fmax of 402.74 MHz between source register "reset" and destination register "counter[1]" (period= 2.483 ns)
Warning: Circuit may not operate. Detected 2 non-operational path(s) clocked by clock "clk" with clock skew larger than data delay. See Compilation Report f
Info: Found hold time violation between source  pin or register "present_state.red2" and destination pin or register "reset" for clock "clk" (Hold time is 4
Info: tco from clock "clk" to destination pin "g light" through register "present state.green2" is 9.471 ns
```

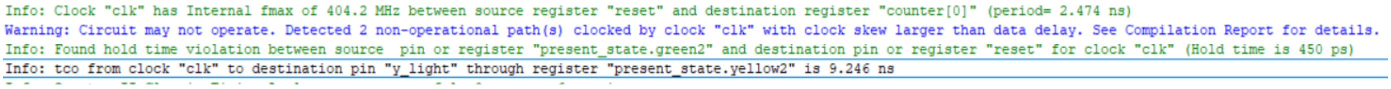


Task	Time
✓ ⊟ ▶ Compile Design	00:00:17
✓ ⊞ ▶ Analysis & Synthesis	00:00:04
✓ ⊞ ▶ Fitter (Place & Route)	00:00:06
✓ ⊞ ▶ Assembler (Generate programming files)	00:00:04
✓ ⊞ ▶ Classic Timing Analysis	00:00:03
⊞ ▶ EDA Netlist Writer	
Program Device (Open Programmer)	

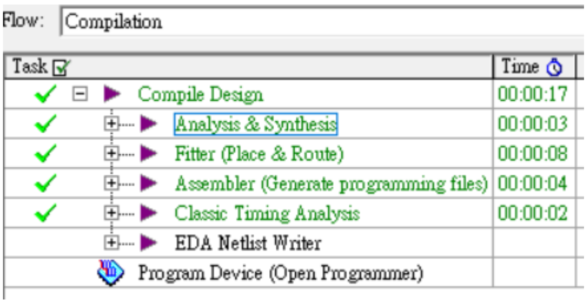
fast fit 是三種選項中，在 fitter 階段耗費時間最少的一個，並且 fmax 也因此受影響變小。

- Auto Fit

fmax = 404.2 MHz, tco = 9.246 ns



```
Info: Clock "clk" has Internal fmax of 404.2 MHz between source register "reset" and destination register "counter[0]" (period= 2.474 ns)
Warning: Circuit may not operate. Detected 2 non-operational path(s) clocked by clock "clk" with clock skew larger than data delay. See Compilation Report for details.
Info: Found hold time violation between source  pin or register "present_state.green2" and destination pin or register "reset" for clock "clk" (Hold time is 450 ps)
Info: tco from clock "clk" to destination pin "y_light" through register "present_state.yellow2" is 9.246 ns
```



Flow: Compilation

Task	Time
✓ ⊟ ▶ Compile Design	00:00:17
✓ ⊞ ▶ Analysis & Synthesis	00:00:03
✓ ⊞ ▶ Fitter (Place & Route)	00:00:08
✓ ⊞ ▶ Assembler (Generate programming files)	00:00:04
✓ ⊞ ▶ Classic Timing Analysis	00:00:02
⊞ ▶ EDA Netlist Writer	
Program Device (Open Programmer)	

auto fit 的 fitter 階段耗時是三種選項最長的，也符合 quartus 介面寫的 "reduce fitter effort after meeting time requirements”

雖然調整不同 fitter effort 的選項確實讓 compilation 各階段的秒數有所不同，但可能是城市的規模不大，或者不夠複雜，因此影響的程度有限。