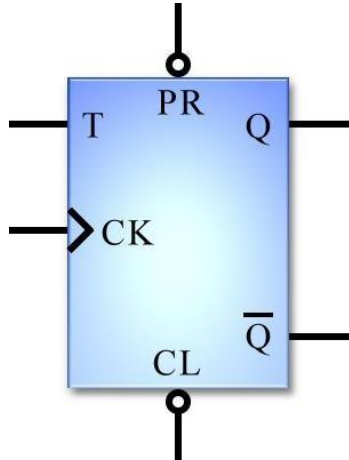


# Shift Registers

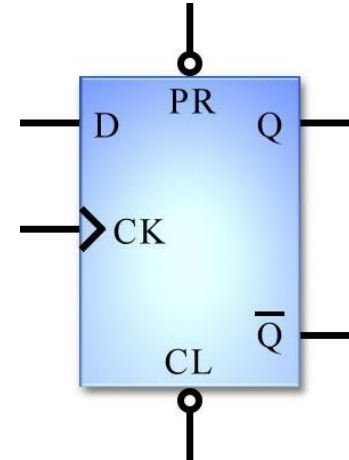
# Flip-Flop 正反器設計

- T型正反器

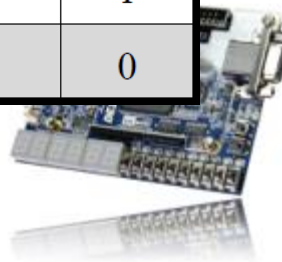


PR	CL	CK	T	Q	$\bar{Q}$
0	1	-	-	1	0
1	0	-	-	0	1
1	1	$\uparrow$	0	Q	$\bar{Q}$
1	1	$\uparrow$	1	$\bar{Q}$	Q

- D型正反器



PR	CL	CK	D	Q	$\bar{Q}$
0	1	-	-	1	0
1	0	-	-	0	1
1	1	$\uparrow$	0	0	1
1	1	$\uparrow$	1	1	0



# Flip-Flop 正反器設計

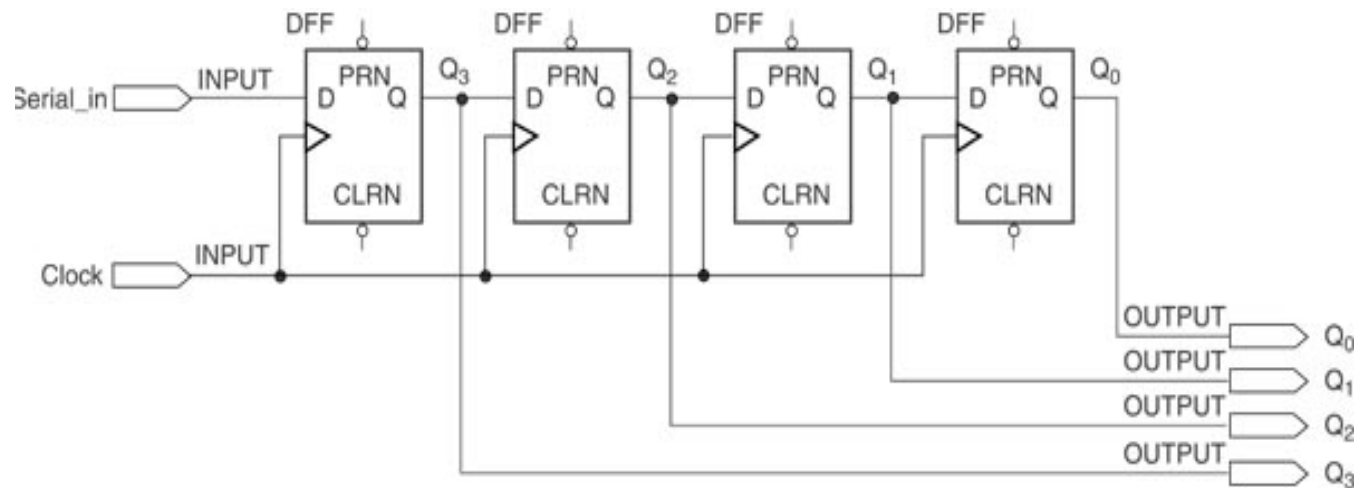
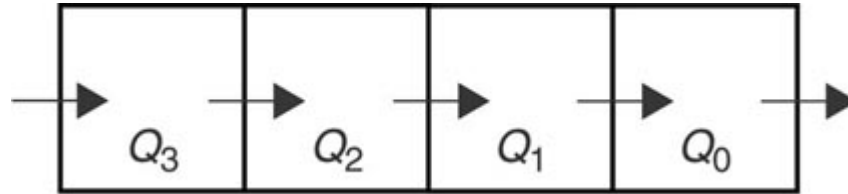
- VHDL for T 型正反器

```
Library IEEE;
Use IEEE.std_logic_1164.all;
Entity TFF1 is
    Port( PR,CL,CK,T:in std_logic;
          Q,Qbar:out std_logic);
End TFF1;
Architecture ARCH of TFF1 is
Begin
    Process (PR,CL,CK)
        Variable TMP:std_logic;
    Begin
        If PR='0' Then TMP := '1';
        Elsif CL='0' Then TMP := '0';
        Elsif Rising_edge(CK) Then
            If T='1' Then TMP := not TMP;
            Else null;
            End If;
        End If;
        Q <= TMP;
        Qbar <= not TMP;
    End Process;
End ARCH;
```

PR	CL	CK	T	Q	$\bar{Q}$
0	1	-	-	1	0
1	0	-	-	0	1
1	1	↑	0	Q	$\bar{Q}$
1	1	↑	1	$\bar{Q}$	Q



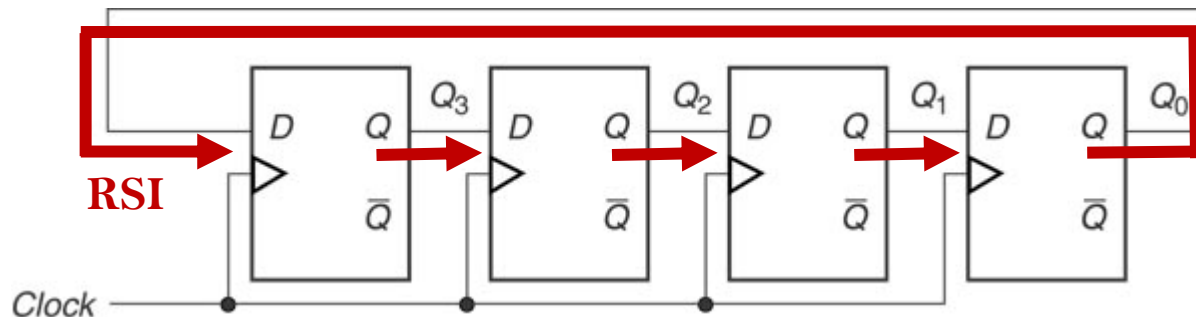
# 序列移位暫存器



# 環形計數器 (Ring Counter)

- 環形計數器是一種特殊的序列移位暫存器，在每個clock下將數據移動一位，並且輸出從最後一個元件重新連接到第一個元件，從而形成一個閉環。

$S_1$	$S_0$	Function	$D_3$	$D_2$	$D_1$	$D_0$
1	1	Hold	$Q_3$	$Q_2$	$Q_1$	$Q_0$
0	1	Shift Right	RSI	$Q_3$	$Q_2$	$Q_1$
1	0	Shift Left	$Q_2$	$Q_1$	$Q_0$	LSI
0	0	Load	$P_3$	$P_2$	$P_1$	$P_0$



RSI: Right-Shift Input  
LSI: Left-Shift Input



# 作業

- 設計一個跑馬燈的電路，讓LED週期性的向左右位移，電路運作如下：
  - SW[0:4]**：控制 **LED[0:4]** 的亮暗，作為初始狀態。
  - SW[8]**：觸發時，將初始為亮的LED每次向右位移1格。
  - SW[9]**：觸發時，將初始為亮的LED每次向左位移1格。
- 同組一起完成DEMO，且由一位代表上傳專案壓縮檔，檔名：lab12\_Gxx.zip (xx為組別)。
- 註：**有DEMO才算分**，若當週未完成DEMO，統一於日後公布之日期補DEMO。
- 註：可參考 Lab5



# 作業參考

- 跑馬燈設計可參考環型計數器
- 請注意，當 **SW[8:9]** 狀態皆未開啟時，**SW[0:4]** 的設定值才能順利導入 **LED[0:4]**

PD <sub>2</sub>	PD <sub>1</sub>	Function	LED <sub>4</sub>	LED <sub>3</sub>	LED <sub>2</sub>	LED <sub>1</sub>	LED <sub>0</sub>
1	1	Hold	LED <sub>4</sub>	LED <sub>3</sub>	LED <sub>2</sub>	LED <sub>1</sub>	LED <sub>0</sub>
1	0	Shift Right	LED <sub>0</sub>	LED <sub>4</sub>	LED <sub>3</sub>	LED <sub>2</sub>	LED <sub>1</sub>
0	1	Shift Left	LED <sub>3</sub>	LED <sub>2</sub>	LED <sub>1</sub>	LED <sub>0</sub>	LED <sub>4</sub>
0	0	Load	SW <sub>4</sub>	SW <sub>3</sub>	SW <sub>2</sub>	SW <sub>1</sub>	SW <sub>0</sub>



# 加分

- 增加新功能，按下按鈕來改變位移的頻率。
  - BUTTON[0]**：降低位移速率。
  - BUTTON[1]**：增加位移速率。
- 同組一起完成DEMO，且由一位代表上傳專案壓縮檔，檔名：lab12\_bonus\_Gxx.zip (xx為組別)。
- 註：**有DEMO才算分**，若當週未完成DEMO，統一於日後公布之日期補DEMO。





# 作業參考

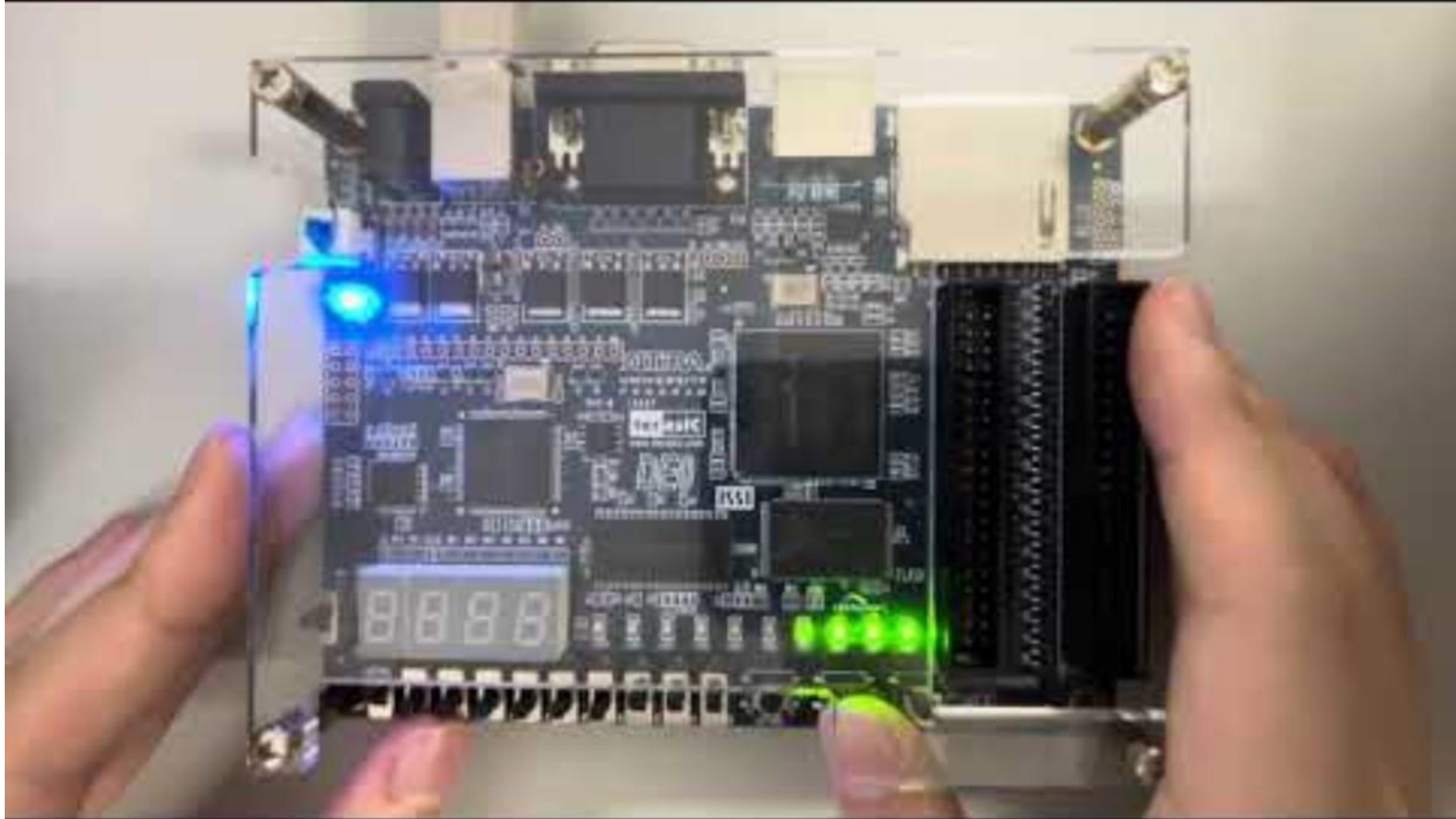
- **clk, pb1, pb2**: in std\_logic
- **btn1, btn2**: in std\_logic
- **LED\_data**: in std\_logic\_vector(4 downto 0)
- **LEDs**: out std\_logic\_vector(9 downto 0)

## 功能解釋

- **pd1, pd2** : SW[8:9] , 控制資料讀入及傳遞方向。
- **btn1, btn2** : DE0 Pushbutton , 控制資料傳送速度。
- **LED\_data** : SW[0:4] , 導入 LEDs 的初始狀態。



# 作業及加分範例

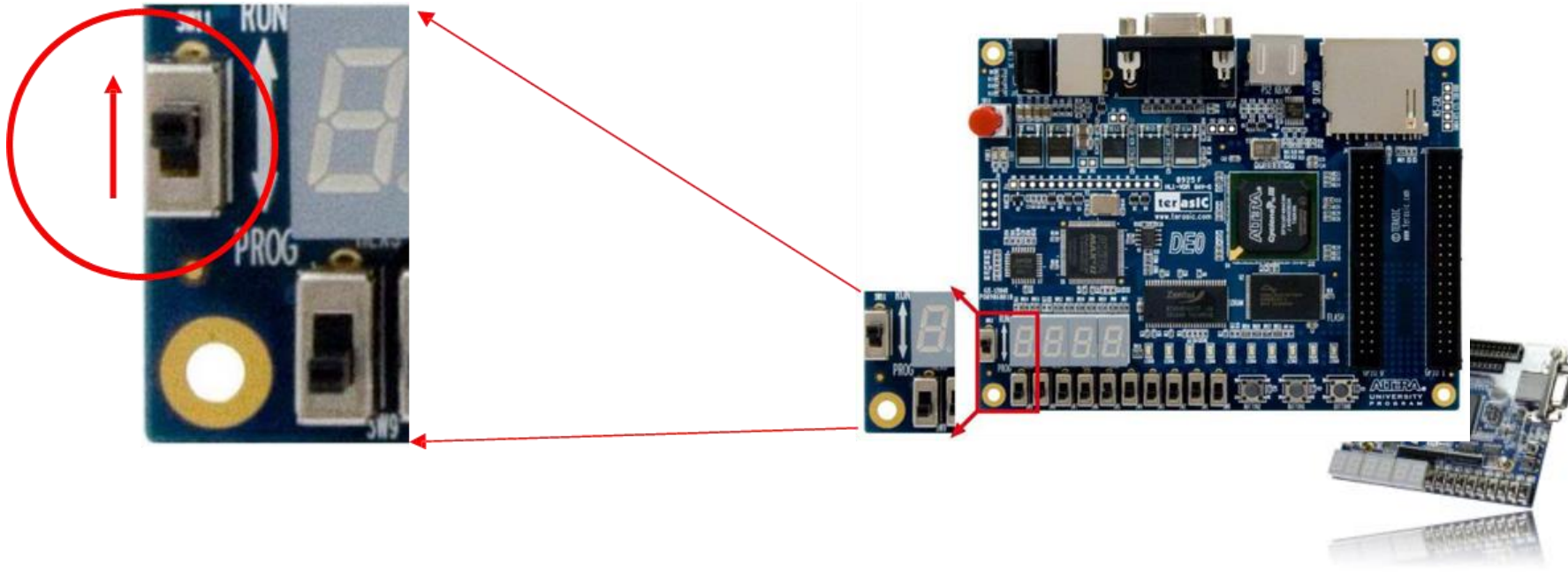
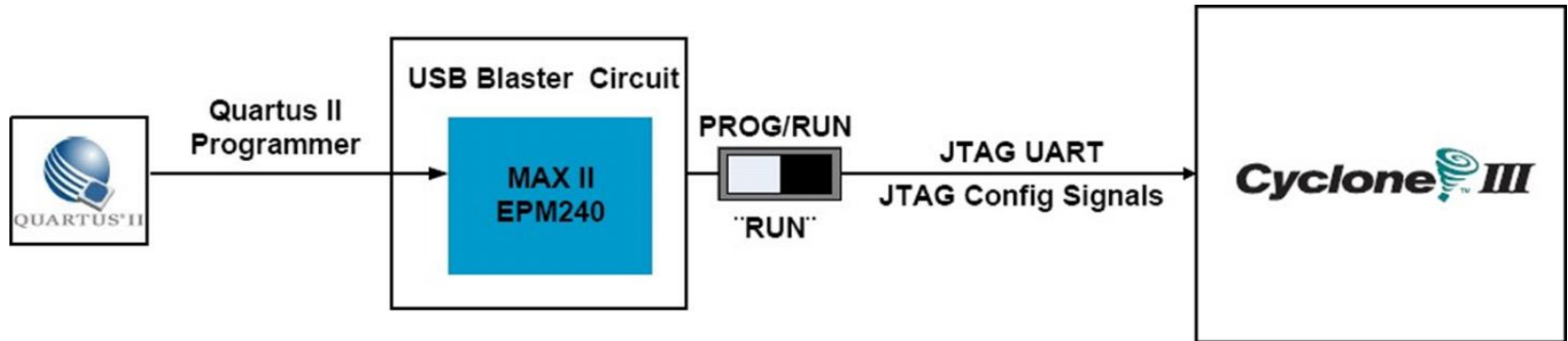


<https://youtu.be/lwXO2rVgEnM?si=yi5f4zcxwFmhLLiaw>



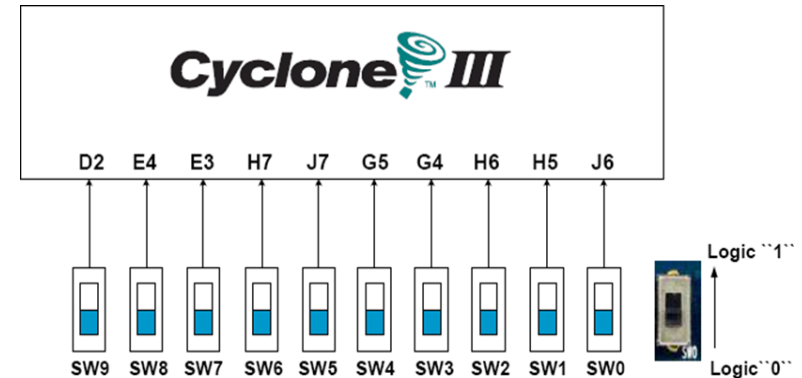
# 模式選取

- Switch RUN 設定為JTAG模式



# Switch 腳位

Signal Name	FPGA Pin No.	Description
SW[0]	PIN_J6	Slide Switch[0]
SW[1]	PIN_H5	Slide Switch[1]
SW[2]	PIN_H6	Slide Switch[2]
SW[3]	PIN_G4	Slide Switch[3]
SW[4]	PIN_G5	Slide Switch[4]
SW[5]	PIN_J7	Slide Switch[5]
SW[6]	PIN_H7	Slide Switch[6]
SW[7]	PIN_E3	Slide Switch[7]
SW[8]	PIN_E4	Slide Switch[8]
SW[9]	PIN_D2	Slide Switch[9]

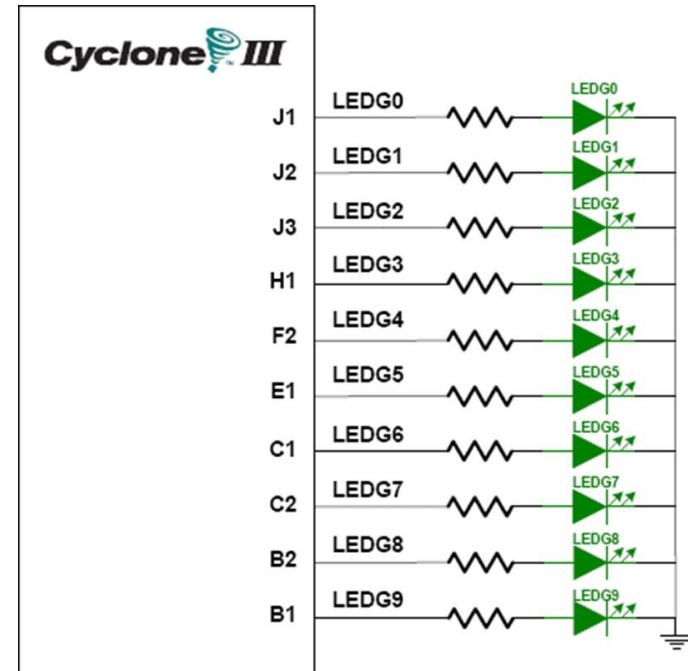


10 Slide switches (Sliders):  
 Up → Logic High  
 Down → Logic Low



# LED 腳位

Signal Name	FPGA Pin No.	Description
LEDG[0]	PIN_J1	LED Green[0]
LEDG[1]	PIN_J2	LED Green[1]
LEDG[2]	PIN_J3	LED Green[2]
LEDG[3]	PIN_H1	LED Green[3]
LEDG[4]	PIN_F2	LED Green[4]
LEDG[5]	PIN_E1	LED Green[5]
LEDG[6]	PIN_C1	LED Green[6]
LEDG[7]	PIN_C2	LED Green[7]
LEDG[8]	PIN_B2	LED Green[8]
LEDG[9]	PIN_B1	LED Green[9]

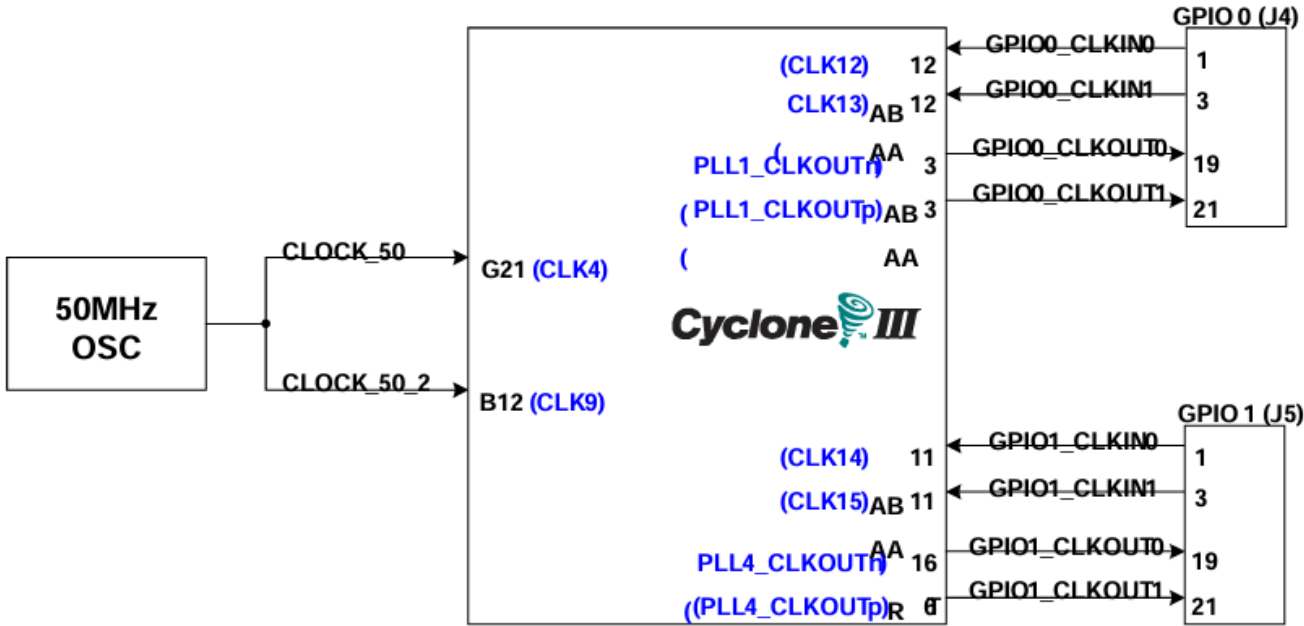


10 LEDs  
 Output High → LED ON  
 Output Low → LED OFF



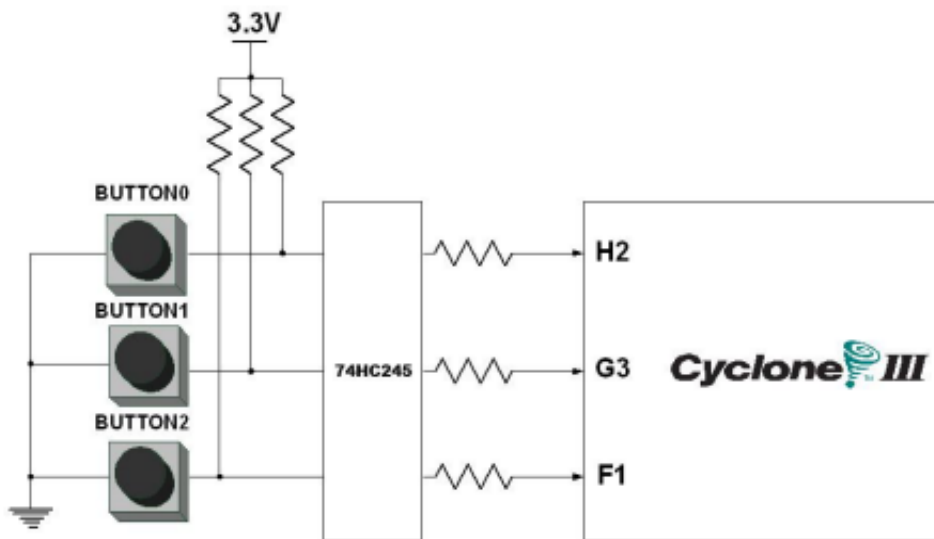
# Clock 腳位

Signal Name	FPGA Pin No.	Description
CLOCK_50	PIN_G21	50 MHz clock input
CLOCK_50_2	PIN_B12	50 MHz clock input



# Button 腳位

Signal Name	FPGA Pin No.	Description
BUTTON [0]	PIN_ H2	Pushbutton[0]
BUTTON [1]	PIN_ G3	Pushbutton[1]
BUTTON [2]	PIN_ F1	Pushbutton[2]



3 Pushbutton:  
Not pressed → Logic High  
Pressed → Logic Low

