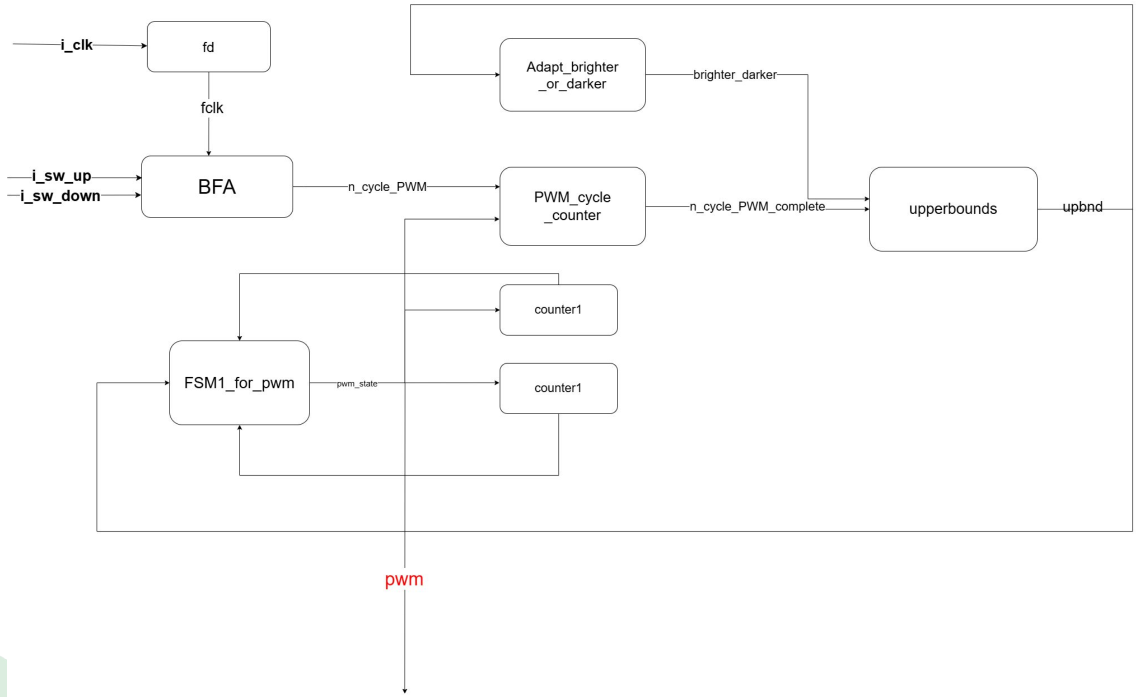
pwm_breath

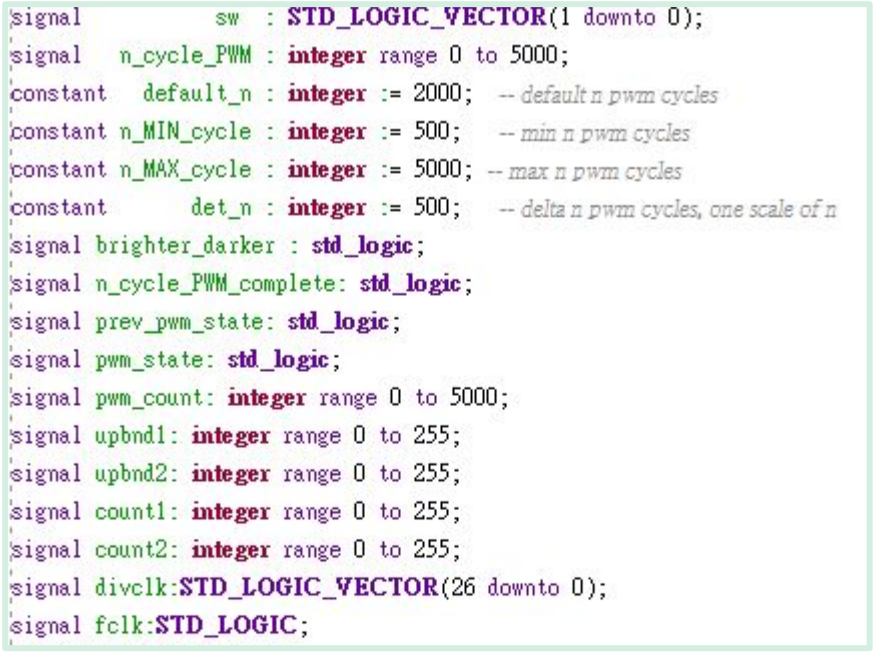
C111112132 蕭詠劍

架構圖

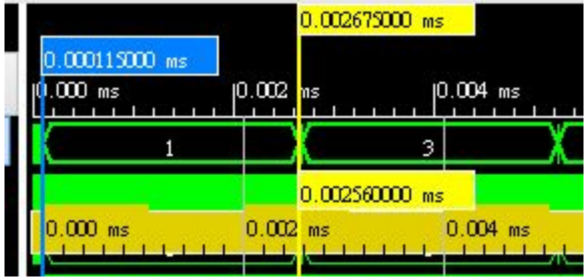


- ❑ **調節 :** 呼吸燈速度的快慢取決於n_cycle_PWM得上限，上限越大時狀態維持得越久呼吸的頻率就會越慢
- ❑ **調整上下限 :** 透過sw來控制上下
- ❑ **除頻 :** i_clk速度太快即使只按了一次要會將上下限調到最高跟最低

說明



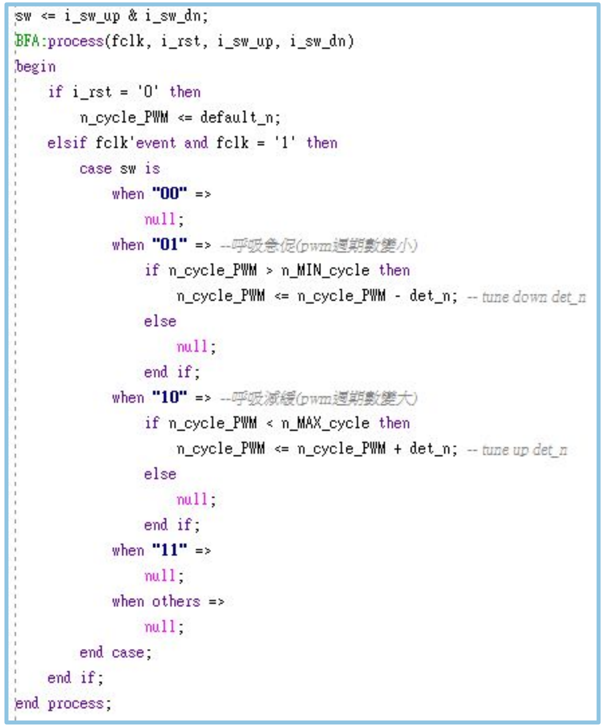
```
signal           sw  : STD_LOGIC_VECTOR(1 downto 0);
signal   n_cycle_PWM : integer range 0 to 5000;
constant   default_n : integer := 2000;   -- default n pwm cycles
constant n_MIN_cycle : integer := 500;    -- min n pwm cycles
constant n_MAX_cycle : integer := 5000; -- max n pwm cycles
constant       det_n : integer := 500;    -- delta n pwm cycles, one scale of n
signal brighter_darker : std_logic;
signal n_cycle_PWM_complete: std_logic;
signal prev_pwm_state: std_logic;
signal pwm_state: std_logic;
signal pwm_count: integer range 0 to 5000;
signal upbnd1: integer range 0 to 255;
signal upbnd2: integer range 0 to 255;
signal count1: integer range 0 to 255;
signal count2: integer range 0 to 255;
signal divclk:STD_LOGIC_VECTOR(26 downto 0);
signal fclk:STD_LOGIC;
```



count數到255是0.0026ms
default_n初始圈數設計亮到暗
1.364s，upbund有256因此一個數到256的
count要維持5.3ms
5.3m除0.0026m 大約是2000

det_n為500肉眼能看出前後的差異

說明

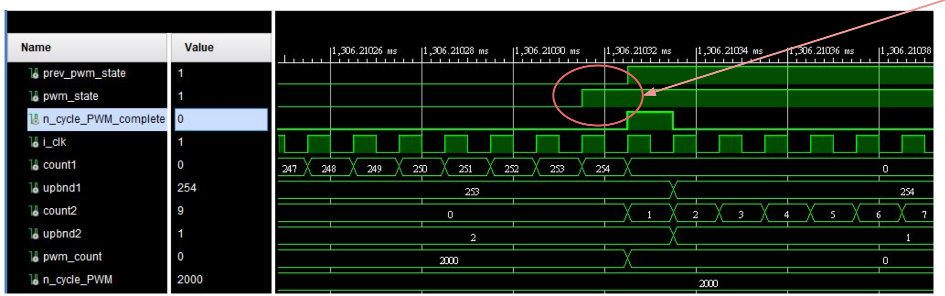


```
sw <= i_sw_up & i_sw_dn;
BFA:process(fclk, i_rst, i_sw_up, i_sw_dn)
begin
    if i_rst = '0' then
        n_cycle_PWM <= default_n;
    elsif fclk'event and fclk = '1' then
        case sw is
            when "00" =>
                null;
            when "01" => --呼吸急促(pwm週期數變小)
                if n_cycle_PWM > n_MIN_cycle then
                    n_cycle_PWM <= n_cycle_PWM - det_n; -- tune down det_n
                else
                    null;
                end if;
            when "10" => --呼吸減緩(pwm週期數變大)
                if n_cycle_PWM < n_MAX_cycle then
                    n_cycle_PWM <= n_cycle_PWM + det_n; -- tune up det_n
                else
                    null;
                end if;
            when "11" =>
                null;
            when others =>
                null;
        end case;
    end if;
end process;
```

根據按鈕 sw_up 和 sw_dn 的狀態, 調整 n_cycle_PWM控制 PWM 的週期數

說明

```
PWM_cycle_counter:process(i_clk, i_rst, n_cycle_PWM, pwm_state)
begin
  if i_rst = '0' then
    n_cycle_PWM_complete <= '0';
    pwm_count <= 0;
    prev_pwm_state <= '0';
  elsif i_clk'event and i_clk = '1' then
    prev_pwm_state <= pwm_state; --Mealey Machine
    if prev_pwm_state = '0' and pwm_state = '1' then
      if pwm_count < n_cycle_PWM then
        pwm_count <= pwm_count + 1;
        n_cycle_PWM_complete <= '0'; -- not yet
      else
        n_cycle_PWM_complete <= '1';
        pwm_count <= 0;
      end if;
    else
      n_cycle_PWM_complete <= '0'; --null;
    end if;
  end if;
end process;
```

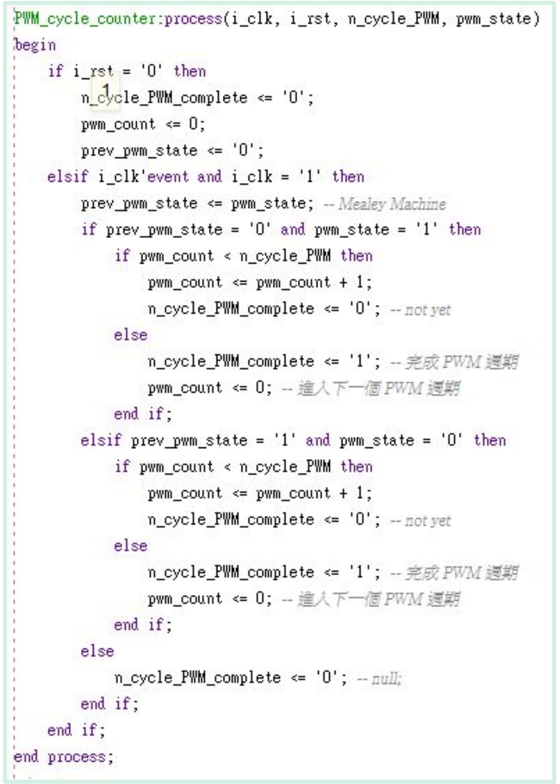


Name	Value
prev_pwm_state	1
pwm_state	1
n_cycle_PWM_complete	0
i_clk	1
count1	0
upbnd1	254
count2	9
upbnd2	1
pwm_count	0
n_cycle_PWM	2000

pwm_state亮到滅後又會是亮到滅

只在上升邊時才開始計數。這樣 PWM 信號的周期只能由上升邊觸發

GPT修改



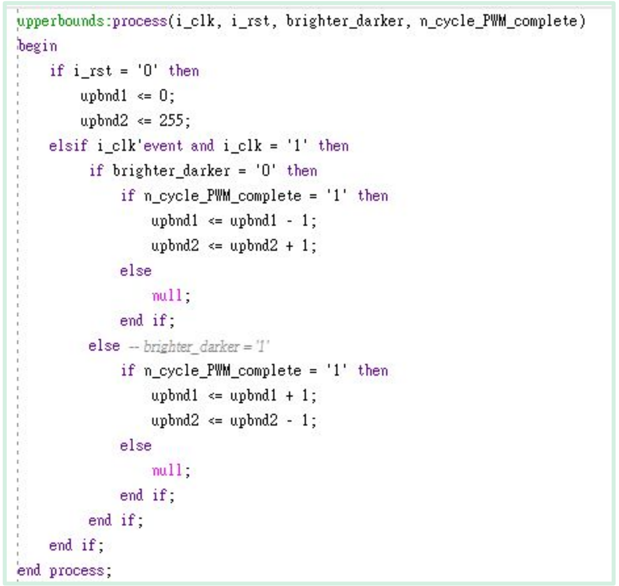
```
PWM_cycle_counter:process(i_clk, i_rst, n_cycle_PWM, pwm_state)
begin
    if i_rst = '0' then
        n_cycle_PWM_complete <= '0';
        pwm_count <= 0;
        prev_pwm_state <= '0';
    elsif i_clk'event and i_clk = '1' then
        prev_pwm_state <= pwm_state; -- Mealey Machine
        if prev_pwm_state = '0' and pwm_state = '1' then
            if pwm_count < n_cycle_PWM then
                pwm_count <= pwm_count + 1;
                n_cycle_PWM_complete <= '0'; -- not yet
            else
                n_cycle_PWM_complete <= '1'; -- 完成 PWM 週期
                pwm_count <= 0; -- 進入下一個 PWM 週期
            end if;
        elsif prev_pwm_state = '1' and pwm_state = '0' then
            if pwm_count < n_cycle_PWM then
                pwm_count <= pwm_count + 1;
                n_cycle_PWM_complete <= '0'; -- not yet
            else
                n_cycle_PWM_complete <= '1'; -- 完成 PWM 週期
                pwm_count <= 0; -- 進入下一個 PWM 週期
            end if;
        else
            n_cycle_PWM_complete <= '0'; -- null;
        end if;
    end if;
end process;
```

問

pwm_state從亮變暗後又會由亮到暗,要如何亮到滅後再到亮,只修改pwm_cycle_counter這個區塊

增加對 上升沿和下降沿 的檢測會使計數更加精確和穩定。
當 pwm_state 由 '0' 變成 '1' 或由 '1' 變成 '0' 時，不論是上升沿還是下降沿，PWM 週期的兩個邊沿都會觸發計數

說明

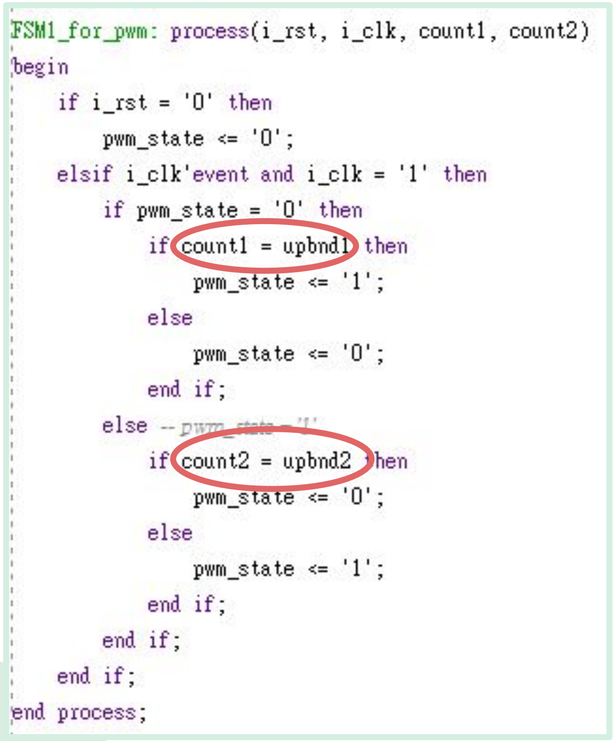


```
upperbounds:process(i_clk, i_rst, brighter_darker, n_cycle_PWM_complete)
begin
    if i_rst = '0' then
        upbnd1 <= 0;
        upbnd2 <= 255;
    elsif i_clk'event and i_clk = '1' then
        if brighter_darker = '0' then
            if n_cycle_PWM_complete = '1' then
                upbnd1 <= upbnd1 - 1;
                upbnd2 <= upbnd2 + 1;
            else
                null;
            end if;
        else -- brighter_darker = '1'
            if n_cycle_PWM_complete = '1' then
                upbnd1 <= upbnd1 + 1;
                upbnd2 <= upbnd2 - 1;
            else
                null;
            end if;
        end if;
    end if;
end process;
```

brighter_darker = 1 時
upbnd1 + 1、upbnd2 - 1　　變暗

brighter_darker = 0 時
upbnd1 - 1、upbnd2 + 1　　變亮

說明

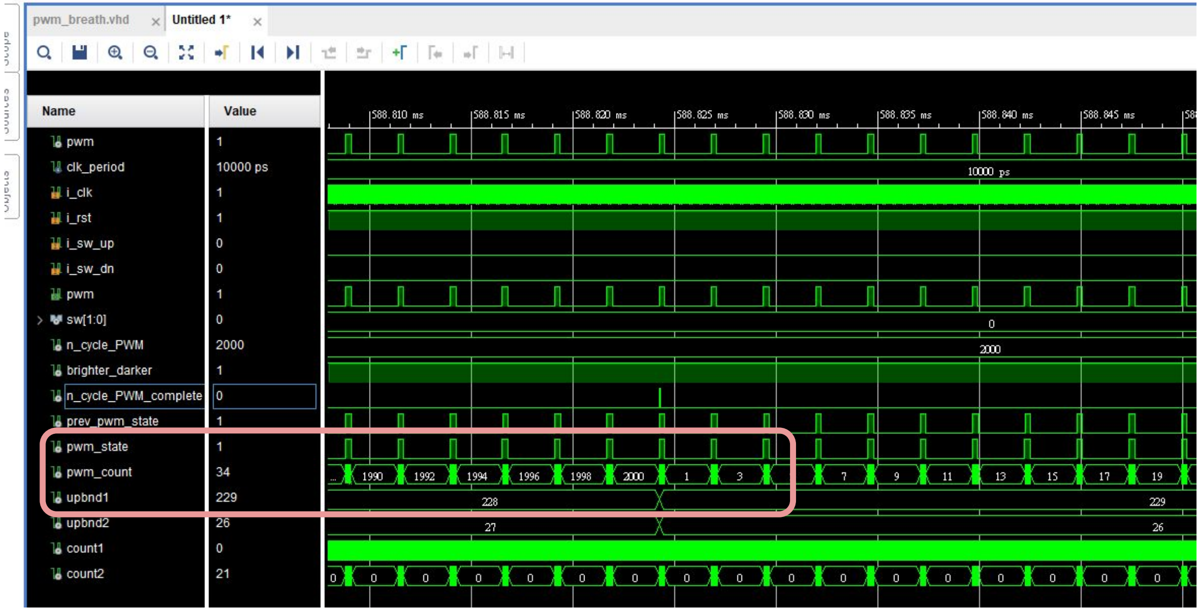


```
FSM1_for_pwm: process(i_rst, i_clk, count1, count2)
begin
    if i_rst = '0' then
        pwm_state <= '0';
    elsif i_clk'event and i_clk = '1' then
        if pwm_state = '0' then
            if count1 = upbnd1 then
                pwm_state <= '1';
            else
                pwm_state <= '0';
            end if;
        else -- pwm_state = '1'
            if count2 = upbnd2 then
                pwm_state <= '0';
            else
                pwm_state <= '1';
            end if;
        end if;
    end if;
end process;
```

當pwm_count = upbund1
 pwm_state 為1

當pwm_count 不等於 upbund1
 pwm_state 為0

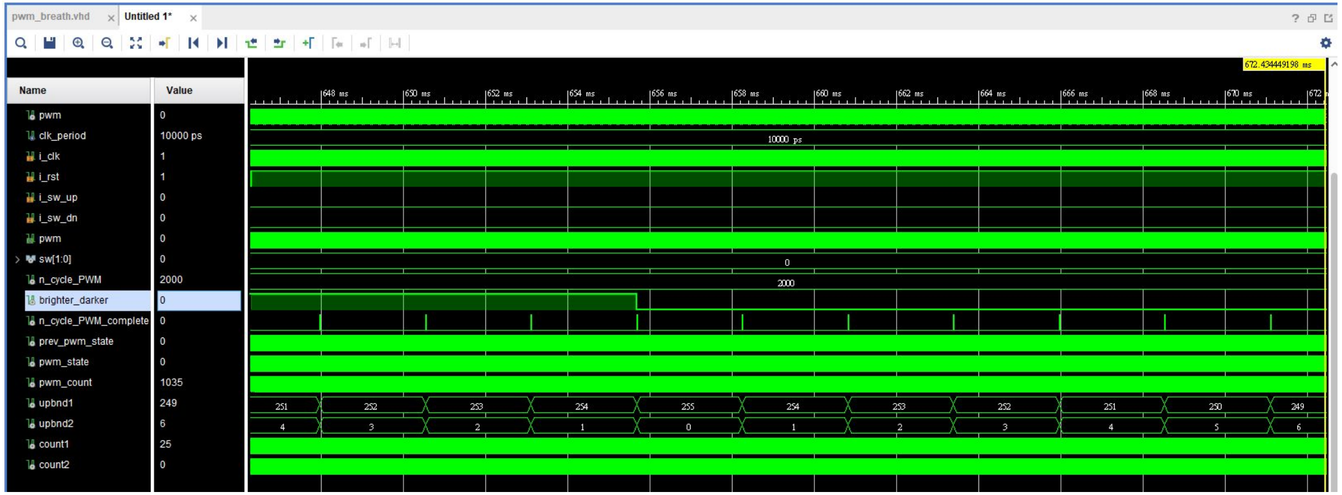
說明



Name	Value
pwm	1
clk_period	10000 ps
i_clk	1
i_rst	1
i_sw_up	0
i_sw_dn	0
pwm	1
sw[1:0]	0
n_cycle_PWM	2000
brighter_darker	1
n_cycle_PWM_complete	0
prev_pwm_state	1
pwm_state	1
pwm_count	34
upbnd1	229
upbnd2	26
count1	0
count2	21

當pwm_count跑完後upbund就會進行調整,使_pwm_state變得更小

說明



Name	Value
pwm	0
clk_period	10000 ps
i_clk	1
i_rst	1
i_sw_up	0
i_sw_dn	0
pwm	0
sw[1:0]	0
n_cycle_PWM	2000
brighter_darker	0
n_cycle_PWM_complete	0
prev_pwm_state	0
pwm_state	0
pwm_count	1035
upbnd1	249
upbnd2	6
count1	25
count2	0

bright_darker變為0讓upbund1數到255後會在
從255往下數再由暗變亮呈現一個呼吸燈的狀態

Thanks