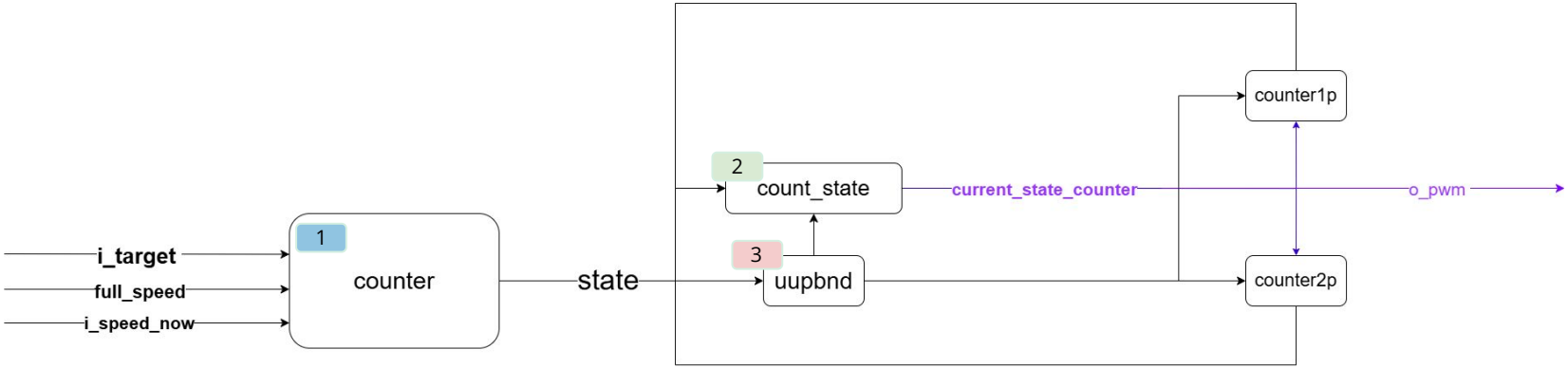
pwm

C111112132 蕭詠劍

架構圖



1 判斷狀態：透過counter來決定目前要進行哪一種count_state

2 調整：在相應的state給予對應的upbund

3 切換狀態：當counter數完時進行狀態的切換

說明

```
begin
    if i_rst = '0' then
        state <= full_speed;
    elsif i_clk' event and i_clk='1' then
        case state is
            when full_speed =>
                if i_th_speed>i_speed_now then
                    state <= full_speed;
                elsif i_speed_now >= i_th_speed and i_speed_now < i_target then
                    state <= speeding_up;
                elsif i_speed_now > i_target then
                    state <= speeding_dn;
                elsif i_speed_now = i_target then
                    state <= final;
                end if;
```

不管是 full_speed 、final speeding_up、speeding_dn ,哪一種狀態都要重新判斷 state的狀態

當目前速度低於標準時進入 full_speed

如果介於目標和標準時狀態為 speeding_up

超過目標時 speeding_down

達到目標速度時狀態為 final

說明

```
count_state:process(i_clk,i_rst)
begin
    if i_rst='0' then
        current_state_counter <= counter1_is_counting ;
    elsif i_clk' event and i_clk='1' then
        case current_state_counter is
            when counter1_is_counting =>
                if counter1= upbnd1 then
                    current_state_counter <= counter2_is_counting ;
                end if;
            when counter2_is_counting =>
                if counter2= upbnd2 then
                    current_state_counter <= counter1_is_counting ;
                end if;
            when others =>
                null;
        end case;
     end if;
end process;
```

當 counter1_is_counting狀態下
counter1 數到 upbnd1 時
, current_state_counter 會切換為
counter2_is_counting

當 counter2_is_counting狀態下
counter2 數到 upbnd2 時
, current_state_counter 會切換為
counter1_is_counting

其他狀態不動作

說明

```
upbnd:process(i_clk,i_rst)
begin
    if i_rst='0' then
        upbnd1 <= "11111111";
        upbnd2 <= "00000000";
    elsif i_clk'event and i_clk='1' then
        case state is
            when full_speed =>
                upbnd1 <= "11111111";
                upbnd2 <= "00000000";
            when speeding_up =>
                upbnd1 <= "10111111"; --191
                upbnd2 <= "00111111"; --63
            when speeding_dn =>
                upbnd1 <= "00111111"; --63
                upbnd2 <= "10111111"; --191
            when final       =>
                upbnd1 <= "01111111"; --127
                upbnd2 <= "01111111"; --127
            when others =>
                null;
        end case;
    end if;
end process;
```

將四種狀態各自分配相應的週期

工作週期:

100% ➡ full_speed

75% ➡ speeding_up

25% ➡ speeding_down

0% ➡ final

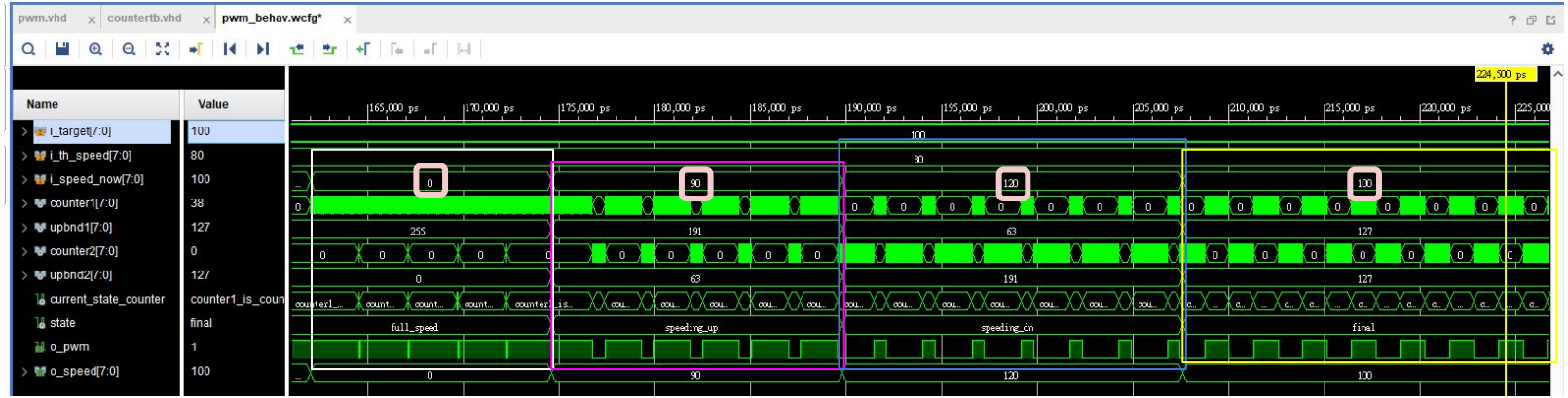
說明

```
pwm:process(i_clk,i_rst)
begin
    if current_state_counter = counter1_is_counting then
        o_pwm <= '1';
    elsif current_state_counter = counter2_is_counting then
        o_pwm <= '0';
    end if;
end process;
```

輸出波形為在

counter1計數時輸出1
counter2計數時輸出0

說明



假設的**四種速度**都會進入對應的state

Thank