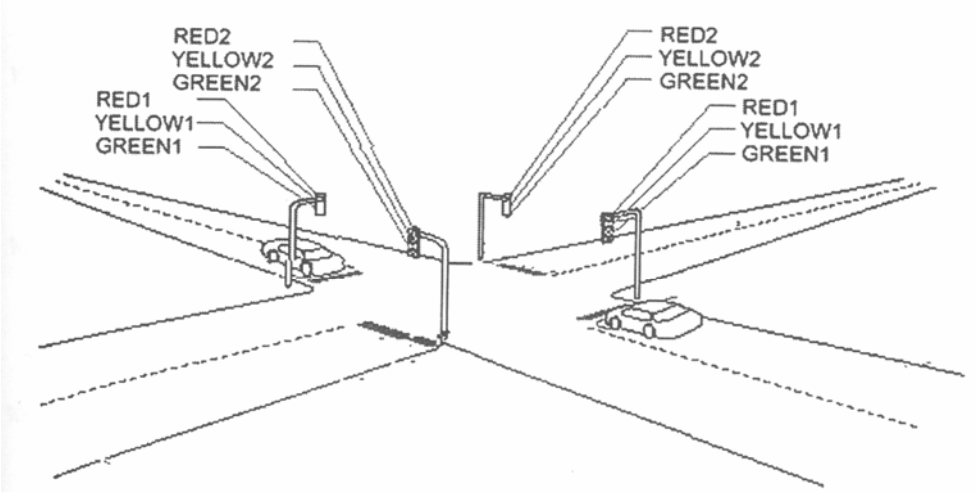
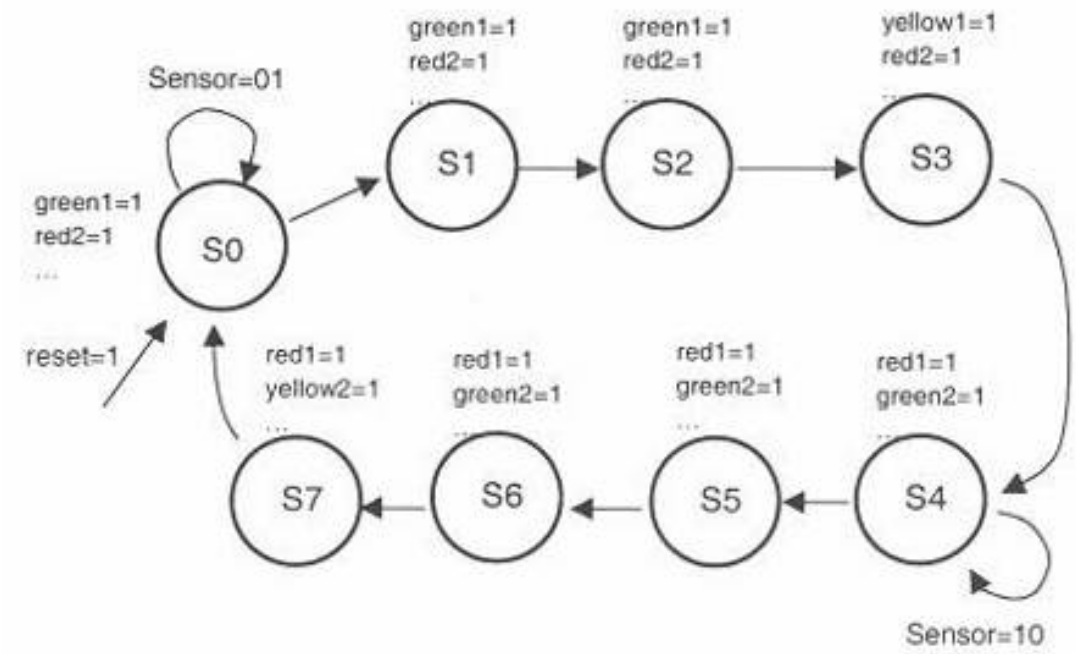
紅綠燈(用狀態機設計)



十字路口紅綠燈



十字路口紅綠燈狀態機設計

紅綠燈 RYG.vhd

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;
ENTITY RYG IS
PORT (clock,reset : IN std_logic;
  sensor: IN std_logic_vector(1 downto 0);
  red1, yellow1, green1, red2, yellow2, green2 : OUT std_logic);
END RYG;
ARCHITECTURE a OF RYG IS
      TYPE state IS ( S0, S1, S2, S3, S4, S5, S6, S7 );    --定義 state 型態為 S0~S7
      SIGNAL present_state, next_state : state;    --將 present_state, next_state 定義為 stste 狀態
BEGIN
      state_comp:PROCESS (present_state, sensor)
      BEGIN
            red1 <= '0'; yellow1 <= '0'; green1 <= '0';
            red2 <= '0'; yellow2 <= '0'; green2 <= '0';
      CASE present_state IS
            WHEN S0 =>
                  green1 <= '1';
                  red2 <= '1';
                  IF sensor= "01" THEN
                  next_state <= S0;
                  ELSE
                  next_state <= S1;
                  END IF;
            WHEN S1 =>
                  green1 <= '1';
                  red2 <= '1';
                  next_state <= S2;
            WHEN S2 =>
                  green1 <= '1';
                  red2 <= '1';
                  next_state <= S3;
            WHEN S3 =>
                  yellow1 <= '1';
                  red2 <= '1';
                  next_state <= S4;
            WHEN S4 =>
                  red1 <= '1';
                  green2 <= '1';
                  IF sensor= "10" THEN
                  next_state <= S4;
                  ELSE
                  next_state <= S5;
                  END IF;
            WHEN S5 =>
                  red1 <= '1';
                  green2 <= '1';
                  next_state <= S6;
            WHEN S6 =>
                  red1 <= '1';
                  green2 <= '1';
                  next_state <= S7;
            WHEN S7 =>
                  red1 <= '1';
                  yellow2 <= '1';
                  next_state <= S0;
      END CASE;
      END PROCESS state_comp;

      state_clocking :PROCESS (reset, clock)
      BEGIN
            IF (reset='1') THEN
            present_state <= S0 ;
            ELSIF clock'event and clock='1' THEN
            present_state <= next_state ;
      END IF ;
      END PROCESS state_clocking;


END a;
```

電子時鐘

```
library ieee ;
use ieee.std_logic_1164.all ;
use ieee.std_logic_unsigned.all ;
use ieee.std_logic_arith.all ;
-----------------------------------------------------------------------------------
entity clock is
   port ( Scan     : out std_logic_vector(5 downto 0);--循序水平掃瞄接點
          Seven    : out std_logic_vector(7 downto 0);--七節顯示器接點
          Set,Up : in std_logic ;                    --模式切換按鈕,數值遞增按鈕
          Reset    : in std_logic ;                  --數值歸零按鈕
          Clk      : in std_logic );                 --石英振盪器接點
end clock ;
-----------------------------------------------------------------------------------
architecture Arch of clock is
  signal   tempa     :STD_LOGIC_vector(12 downto  0);      --20MHz 降頻至 2KHz
  signal   tempb     :STD_LOGIC;                           --準位轉換
  signal   ch        :STD_LOGIC;                           --循序水平掃瞄頻率
  signal Count     : std_logic_vector(21 downto 0);     --數值暫存器
  signal Mode      : std_logic_vector(1 downto 0);      --模式切換編碼
  signal Bcd       : std_logic_vector(3 downto 0);      --七節顯示器編碼
  signal Hrb       : std_logic_vector(3 downto 0);      --調時閃爍信號
  signal Mrb       : std_logic_vector(3 downto 0);      --調分閃爍信號
  signal Srb       : std_logic_vector(3 downto 0);      --調秒閃爍信號
  signal Fre_1hz : std_logic ;                          --閃爍頻率
  signal Dot_buf : std_logic ;                          --小數點閃爍信號
  signal Up_buf    : std_logic ;                        --數值遞增信號
  signal Set_buf : std_logic ;                          --模式切換信號
  signal Time      : std_logic ;                        --數值遞增頻率
  signal Pulse     : std_logic ;                        --防按鈕彈跳頻率
BEGIN
--20MHz 降頻至 2KHz-----------------------------------------------------------------
process   (clk)
begin
  if   clk'event and   clk='1'    then
    if tempa     /=4999    then tempa<=tempa+1;
    else
    tempa<="0000000000000";
    tempb<=not tempb;    --反向除 2
    ch<=tempb;
    end     if;
  endif;
end  process;
```

$20M \div 5000 \div 2 = 2KHZ$

```
-----------------------------------------------------------------------------------------
   Hrb       <= "1111"   when Mode="11" and     Fre_1hz='0'   else "0000" ;--調時閃爍設定值
   Mrb       <= "1111"   when Mode="10" and   Fre_1hz='0'   else "0000" ;--調分閃爍設定值
   Srb       <= "1111"   when Mode="01" and   Fre_1hz='0'   else "0000" ;--調秒閃爍設定值
   Time      <= Fre_1hz when Mode="00" else Up_buf ;                        --數值遞增頻率設定值
   Seven(7)<= Dot_buf when Reset='0' and   Mode    ="00" else '0' ;     --小數點閃爍設定值
--2KHz 降頻至 1Hz-----------------------------------------------------------------------------
Ch_generator:
process(Ch)
   variable Delay : std_logic_vector(9   downto 0);
   begin
   if rising_edge(Ch) then
      if Delay=1000 then Delay := "0000000000" ;--2KHz 降頻至 2Hz
      Fre_1hz <= not Fre_1hz ;--除 2 程式
      else Delay := Delay+1 ;
      end if ;
   Pulse <= Delay(4);--防按鈕彈跳頻率設定值
   end if ;
end process Ch_generator ;
--防按鈕彈跳程式-----------------------------------------------------------------------------
Timer_Set:
process(Pulse)
   begin
   if Reset='1' then
   Set_buf <= '0' ;
   Up_buf <= '0' ;
   elsif rising_edge(Pulse) then
   Set_buf <= Set ;
   Up_buf <= Up ;
   end if ;
end process Timer_Set ;
--模式切換程式-------------------------------------------------------------------------------
Mode_select:
process(Set_buf)
   begin
   if Reset='1' then
   Mode <= "00" ;--回復計時功能
   elsif rising_edge(Set_buf) then
   Mode <= Mode+1 ;
   end if ;
end process Mode_select ;
--時數高位元進位程式-------------------------------------------------------------------------
Timer_Count:
process(Time,Reset)
```

$2k \div 1000 \div 2 = 1HZ$

```
begin
  if   Reset='1' then Count <= "00000000000000000000" ;
  elsif   rising_edge(Time) then
    if       (Count(21 downto 0) >= "1000110101100101011001" and Mode="00") or
              (Count(21 downto 16)>= "100011" and Mode="11") then Count(21 downto 20) <= "00" ;
    elsif (Count(19 downto 0) >= "10010101100101011001" and Mode="00") or
              (Count(19 downto 16)>= "1001" and Mode="11") then
                Count(21 downto 20) <= Count(21 downto 20)+1 ;
    end if ;
--時數低位元進位程式-----------------------------------------------------------------
    if      ((Count(19 downto 0) >= "10010101100101011001" or
                Count(21 downto 0) >= "1000110101100101011001" ) and Mode="00") or
             ((Count(19 downto 16)>= 9 or Count(21 downto 16) >= "100011")and Mode="11" ) then
                Count(19 downto 16)<= "0000" ;
    elsif (Count(15 downto 0) >= "0101100101011001" and Mode="00") or Mode="11" then
                Count(19 downto 16) <= Count(19 downto 16)+1 ;
    end if ;
--分數高位元進位程式-----------------------------------------------------------------
    if (Count(15 downto 0) >= "0101100101011001" and Mode="00") or
              (Count(15 downto 8) >= "01011001" and Mode="10" ) then
                Count(15 downto 12)<= "0000" ;
    elsif (Count(11 downto 0) >= "100101011001" and Mode="00") or
              (Count(11 downto 8) >= "1001" and Mode="10") then
                Count(15 downto 12)<= Count(15 downto 12)+1 ;
    end if ;
--分數低位元進位程式-----------------------------------------------------------------
    if (Count(11 downto 0) >= "100101011001" and Mode="00" ) or
              (Count(11 downto 8) >= 9 and Mode="10" )then
                Count(11 downto 8) <= "0000" ;
    elsif (Count(7    downto 0) >= "01011001" and Mode="00") or Mode="10" then
                Count(11 downto 8) <= Count(11 downto 8)+1 ;
    end if ;
--秒數高位元進位程式-----------------------------------------------------------------
    if (Count(7 downto 0) >= "01011001" and (Mode="00" or Mode="01")) then
                Count(7 downto 4) <= "0000" ;
    elsif (Count(3 downto 0) >= 9 and (Mode="00" or Mode="01")) then
                Count(7 downto 4) <= Count(7 downto 4 )+1 ;
    end if ;
--秒數低位元進位程式-----------------------------------------------------------------
    if (Count(3 downto 0)=9    and (Mode="00" or Mode="01"))then
                Count(3 downto 0) <= "0000" ;
    elsif (Mode="00" or Mode="01") then
                Count(3 downto 0 ) <= Count(3 downto 0)+1 ;
    end if ;
```

```
end if ;
end process Timer_Count ;
--七節顯示器解碼程式-----------------------------------------------------------------
Signal_Scan:
process(Ch)
variable Scan1 : std_logic_vector(2 downto 0);--循序水平掃瞄信號編碼
begin
  if (Ch'event and Ch='1')then
    if (Scan1="000")then Scan <= "100000" ;--循序水平掃瞄信號解碼
    Bcd          <=("00" & (Count(21 downto 20)) or Hrb(3 downto 0));--七節顯示器數值編碼
      if        Bcd="0000"then Seven(6 downto 0) <= "0111111";--0
      elsif Bcd="0001"then Seven(6 downto 0) <= "0000110";--1
      elsif Bcd="0010"then Seven(6 downto 0) <= "1011011";--2
      elsif Bcd="0011"then Seven(6 downto 0) <= "1001111";--3
      elsif Bcd="0100"then Seven(6 downto 0) <= "1100110";--4
      elsif Bcd="0101"then Seven(6 downto 0) <= "1101101";--5
      elsif Bcd="0110"then Seven(6 downto 0) <= "1111101";--6
      elsif Bcd="0111"then Seven(6 downto 0) <= "0100111";--7
      elsif Bcd="1000"then Seven(6 downto 0) <= "1111111";--8
      elsif Bcd="1001"then Seven(6 downto 0) <= "1101111";--9
      else                              Seven(6 downto 0) <= "0000000";--遮沒
      end if;
    Dot_buf <= '0' ;--小數點遮沒
      elsif   (Scan1="001")then Scan <= "010000" ;
      Bcd         <=Count(19      downto 16)or Hrb(3 downto 0);
      if        Bcd="0000"then Seven(6 downto 0) <= "0111111";
      elsif Bcd="0001"then Seven(6 downto 0) <= "0000110";
      elsif Bcd="0010"then Seven(6 downto 0) <= "1011011";
      elsif Bcd="0011"then Seven(6 downto 0) <= "1001111";
      elsif Bcd="0100"then Seven(6 downto 0) <= "1100110";
      elsif Bcd="0101"then Seven(6 downto 0) <= "1101101";
      elsif Bcd="0110"then Seven(6 downto 0) <= "1111101";
      elsif Bcd="0111"then Seven(6 downto 0) <= "0100111";
      elsif Bcd="1000"then Seven(6 downto 0) <= "1111111";
      elsif Bcd="1001"then Seven(6 downto 0) <= "1101111";
      else                              Seven(6 downto 0) <= "0000000";
      end if;
  Dot_buf <= '0' ;
        elsif        (Scan1="010")then Scan <= "001000" ;
        Bcd       <= Count(15 downto 12)or Mrb(3 downto 0);
      if        Bcd="0000"then Seven(6 downto 0) <= "0111111";
      elsif Bcd="0001"then Seven(6 downto 0) <= "0000110";
      elsif Bcd="0010"then Seven(6 downto 0) <= "1011011";
      elsif Bcd="0011"then Seven(6 downto 0) <= "1001111";
```

```
    elsif Bcd="0100"then Seven(6 downto 0) <= "1100110";
    elsif Bcd="0101"then Seven(6 downto 0) <= "1101101";
    elsif Bcd="0110"then Seven(6 downto 0) <= "1111101";
    elsif Bcd="0111"then Seven(6 downto 0) <= "0100111";
    elsif Bcd="1000"then Seven(6 downto 0) <= "1111111";
    elsif Bcd="1001"then Seven(6 downto 0) <= "1101111";
    else                                Seven(6 downto 0) <= "0000000";
    end if;
  Dot_buf <= '0' ;
    elsif          (Scan1="011")        then Scan <= "000100" ;
    Bcd         <= Count(11 downto 8)or Mrb(3 downto 0);
    if         Bcd="0000"then Seven(6 downto 0) <= "0111111";
    elsif Bcd="0001"then Seven(6 downto 0) <= "0000110";
    elsif Bcd="0010"then Seven(6 downto 0) <= "1011011";
    elsif Bcd="0011"then Seven(6 downto 0) <= "1001111";
    elsif Bcd="0100"then Seven(6 downto 0) <= "1100110";
    elsif Bcd="0101"then Seven(6 downto 0) <= "1101101";
    elsif Bcd="0110"then Seven(6 downto 0) <= "1111101";
    elsif Bcd="0111"then Seven(6 downto 0) <= "0100111";
    elsif Bcd="1000"then Seven(6 downto 0) <= "1111111";
    elsif Bcd="1001"then Seven(6 downto 0) <= "1101111";
    else                                Seven(6 downto 0) <= "0000000";
    end if;
  Dot_buf <= '0' ;
    elsif   (Scan1="100")        then Scan <= "000010" ;
    Bcd         <= Count(7 downto 4)or Srb(3 downto 0);
    if         Bcd="0000"then Seven(6 downto 0) <= "0111111";
    elsif Bcd="0001"then Seven(6 downto 0) <= "0000110";
    elsif Bcd="0010"then Seven(6 downto 0) <= "1011011";
    elsif Bcd="0011"then Seven(6 downto 0) <= "1001111";
    elsif Bcd="0100"then Seven(6 downto 0) <= "1100110";
    elsif Bcd="0101"then Seven(6 downto 0) <= "1101101";
    elsif Bcd="0110"then Seven(6 downto 0) <= "1111101";
    elsif Bcd="0111"then Seven(6 downto 0) <= "0100111";
    elsif Bcd="1000"then Seven(6 downto 0) <= "1111111";
    elsif Bcd="1001"then Seven(6 downto 0) <= "1101111";
    else                                Seven(6 downto 0) <= "0000000";
    end if;
Dot_buf <= Fre_1hz ;--小數點顯現
    elsif   (Scan1="101")        then Scan <= "000001" ;
    Bcd         <= Count(3 downto 0)or Srb(3 downto 0);
    if         Bcd="0000"then Seven(6 downto 0) <= "0111111";
    elsif Bcd="0001"then Seven(6 downto 0) <= "0000110";
    elsif Bcd="0010"then Seven(6 downto 0) <= "1011011";
```

```
      elsif Bcd="0011"then Seven(6 downto 0) <= "1001111";
      elsif Bcd="0100"then Seven(6 downto 0) <= "1100110";
      elsif Bcd="0101"then Seven(6 downto 0) <= "1101101";
      elsif Bcd="0110"then Seven(6 downto 0) <= "1111101";
      elsif Bcd="0111"then Seven(6 downto 0) <= "0100111";
      elsif Bcd="1000"then Seven(6 downto 0) <= "1111111";
      elsif Bcd="1001"then Seven(6 downto 0) <= "1101111";
      else                                Seven(6 downto 0) <= "0000000";
      end if;
Dot_buf <= '0' ;
    end if ;
        if Scan1 >= "101" then Scan1 := "000" ;
        else Scan1 := Scan1 + 1 ;
        end if ;
end if ;
end process Signal_Scan ;
-----------------------------------------------------------------------------------------
end Arch ;
```