WS2812B_Driver.vhd

```
--WS2812BCLK .=. 0.4us
--WS2812B 驅動器

Library IEEE;                          --連結零件庫
Use IEEE.std_logic_1164.all;           --引用套件
Use IEEE.std_logic_unsigned.all;       --引用套件

entity WS2812B_Driver is
    port(   WS2812BCLK,WS2812BRESET,loadck:in std_logic;
            --操作頻率,重置,載入 ck
            LEDGRBdata:in std_logic_vector(23 downto 0);
            --色彩資料
            reload,emitter,WS2812Bout:out std_logic
            --要求載入,發射狀態,發射輸出
        );
end entity WS2812B_Driver;

architecture Albert of WS2812B_Driver is
    signal load_clr,reload1:std_logic:='0';--載入信號操作
    signal LEDGRBdata0,LEDGRBdata1:std_logic_vector(23 downto 0);
    --色彩資料載入
    signal DATA01:std_logic_vector(2 downto 0):="000";
    --編碼位元:bit out=>0:100,1:110
    signal DATAn:integer range 0 to 31:=0; --色彩資料位元指標
    signal bitn:integer range 0 to 3:=0;   --編碼位元發射數

begin
WS2812Bout<=DATA01(2);--LED 色彩資料位元輸出
reload<=not (reload1 or load_clr) and WS2812BRESET;
--緩衝器要求載入資料脈衝

--預載緩衝器--
LEDdata_load:process(loadck,WS2812BRESET)
begin
    if WS2812BRESET='0' or (load_clr='1' and reload1='1') then
        reload1<='0';   --緩衝器空
    elsif rising_edge(loadck) then --色彩資料載入 ck
        LEDGRBdata1<=LEDGRBdata;    --色彩資料載入緩衝器
        reload1<='1';   --緩衝器滿
    end if;
end process LEDdata_load;

--串列傳輸器--
WS2812B_Send:process(WS2812BCLK,WS2812BRESET)
begin
    if WS2812BRESET='0' then
        DATA01<="000";  --輸出停止位元
        load_clr<='0';  --允許緩衝器動作
        emitter<='0';   --停止發射
        DATAn<=0;       --等待發射位元數
```

```
        bitn<=0;          --編碼位元發射剩 0 位元
    elsif rising_edge(WS2812BCLK) then
        load_clr<='0';                  --允許緩衝器動作
        if bitn/=0 then                 --尚有編碼位元未發射
            DATA01<=DATA01(1 downto 0) & "0";--發射位元
            bitn<=bitn-1;               --編碼位元發射位元減 1
        elsif DATAn/=0 then             --尚有資料位元未編碼
            DATA01<='1' & LEDGRBdata0(DATAn-1) & '0';
            --發射位元編碼(等待發射位元編碼成 3 位元)
            DATAn<=DATAn-1;             --等待發射位元數減 1
            bitn<=2;                    --編碼位元發射剩 2 位元
        elsif reload1='1' then          --緩衝器已有色彩資料進來
            LEDGRBdata0<=LEDGRBdata1;   --色彩資料載入
            DATAn<=23;                  --等待發射位元數
            DATA01<='1' & LEDGRBdata1(23) & '0';
            --發射位元編碼(等待發射位元編碼成 3 位元)
            bitn<=2;                    --編碼位元發射剩 2 位元
            load_clr<='1';              --已載入發射中,清除緩衝器
            emitter<='1';               --發射中
        else                            --緩衝器無色彩資料
            emitter<='0';               --停止發射
        end if;
    end if;
end process WS2812B_Send;

end Albert;
```

CH3_WS2812B_1.vhd

```
--Ws2812B RGB_LED 霹靂燈 1
--EP3C16Q240C8 50MHz LEs:15,408 PINs:161 ,gckP31 ,rstP99

Library IEEE;                           --連結零件庫
Use IEEE.std_logic_1164.all;            --引用套件
Use IEEE.std_logic_unsigned.all;        --引用套件

entity CH3_WS2812B_1 is
    port(gckP31,rstP99:in std_logic;--系統時脈,系統重置
         WS2812Bout:out std_logic);--WS2812B_Di 信號輸出(184)
end entity CH3_WS2812B_1;

architecture Albert of CH3_WS2812B_1 is
    --WS2812B 驅動器--
    component WS2812B_Driver is
        port(   WS2812BCLK,WS2812BRESET,loadck:in std_logic;
                --操作頻率,重置,載入 ck
                LEDGRBdata:in std_logic_vector(23 downto 0);
                --色彩資料
                reload,emitter,WS2812Bout:out std_logic
```

```
                --要求載入,發射狀態,發射輸出
            );
    end component;
    signal WS2812BCLK,WS2812BRESET:std_logic;
    --操作頻率,重置
    signal loadck,reload,emitter:std_logic;
    --載入 ck,要求載入,發射狀態
    signal LEDGRBdata:std_logic_vector(23 downto 0);
    --色彩資料

    signal FD:std_logic_vector(24 downto 0);
    --系統除頻器
    signal FD2:std_logic_vector(3 downto 0);
    --WS2812B_Driver 除頻器
    signal SpeedS,WS2812BPCK:std_logic;
    --WS2812BP 操作頻率選擇,WS2812BP 操作頻率
    signal delay:integer range 0 to 127;--停止時間
    signal LED_WS2812B_N:integer range 0 to 127;
    --WS2812B 個數指標
    constant NLED:integer range 0 to 127:=29;
    --WS2812B 個數:61 個(0~60)
    signal RC,GC,BC:std_logic_vector(7 downto 0);
    --紅,綠,藍色

begin

--WS2812B 驅動器--
WS2812BN:WS2812B_Driver port map(
        WS2812BCLK,WS2812BRESET,loadck,LEDGRBdata,
        reload,emitter,WS2812Bout);
WS2812BRESET<=rstP99;   --系統重置

--色彩資料--
LEDGRBdata<=GC & RC & BC;

--WS2812BP 操作頻率選擇
WS2812BPCK<=FD(8) when SpeedS='0' else FD(16);--最慢速率

--WS2812BP 主控器--
WS2812BP:process(WS2812BPCK)
variable cc:integer range 0 to 15;              --色階
variable RGBcase:std_logic_vector(3 downto 0); --色盤種類
variable RA,GA,BA:std_logic_vector(1 downto 0);
--紅,綠,藍調色狀態
begin
    if rstP99='0' then
        LED_WS2812B_N<=NLED;        --從頭開始
        RGBcase:=(others=>'0');     --色盤種類預設
        cc:=0;                      --色階預設
        loadck<='0';                --等待載入
        SpeedS<='1';                --加快操作速率
    elsif rising_edge(WS2812BPCK) then
        if loadck='0' then          --等待載入
```

```
            loadck<=reload;          --是否載入
        elsif LED_WS2812B_N=NLED then--輸出個數完成
            SpeedS<='1';             --放慢操作速率
            if emitter='0' then      --已停止發射
                if delay/=0 then     --點亮時間&變化速率
                    delay<=delay-1; --時間遞減
                else
                    loadck<='0';     --reemitter
                    LED_WS2812B_N<=0;--從頭開始
                    SpeedS<='0';     --加快操作速率
                    if cc=0 then
                        cc:=8;       --8 色階數
                        case RGBcase is
                            when "0000"=>
                                RC<=(others=>'0');  --紅全暗
                                GC<=(others=>'0');  --綠全暗
                                BC<=(others=>'0');  --藍全暗
                                RA:="10";    --8 段遞增
                                GA:="00";    --不變
                                BA:="00";    --不變
                            when "0001"=>
                                RC<=(others=>'0');
                                GC<=(others=>'0');
                                BC<=(others=>'0');
                                RA:="00";    --不變
                                GA:="10";    --8 段遞增
                                BA:="00";    --不變
                            when "0010"=>
                                RC<=(others=>'0');
                                GC<=(others=>'0');
                                BC<=(others=>'0');
                                RA:="00";    --不變
                                GA:="00";    --不變
                                BA:="10";    --8 段遞增
                            when "0011"=>
                                RC<=(others=>'0');
                                GC<=(others=>'0');
                                BC<=(others=>'0');
                                RA:="10";    --8 段遞增
                                GA:="10";    --8 段遞增
                                BA:="00";    --不變
                            when "0100"=>
                                RC<=(others=>'0');
                                GC<=(others=>'0');
                                BC<=(others=>'0');
                                RA:="10";    --8 段遞增
                                GA:="00";    --不變
                                BA:="10";    --8 段遞增
                            when "0101"=>
                                RC<=(others=>'0');
                                GC<=(others=>'0');
                                BC<=(others=>'0');
                                RA:="00";    --不變
```

```
                GA:="10";   --8段遞增
                BA:="10";   --8段遞增
            when "0110"=>
                RC<=(others=>'0');
                GC<=(others=>'0');
                BC<=(others=>'0');
                RA:="10";   --8段遞增
                GA:="10";   --8段遞增
                BA:="10";   --8段遞增
            when "0111"=>
                RC<=(others=>'1'); --紅全亮
                GC<=(others=>'1'); --綠全亮
                BC<=(others=>'1'); --藍全亮
                RA:="01";   --8段遞減
                GA:="00";   --不變
                BA:="00";   --不變
            when "1000"=>
                RC<=(others=>'1');
                GC<=(others=>'1');
                BC<=(others=>'1');
                RA:="00";   --不變
                GA:="01";   --8段遞減
                BA:="00";   --不變
            when "1001"=>
                RC<=(others=>'1');
                GC<=(others=>'1');
                BC<=(others=>'1');
                RA:="00";   --不變
                GA:="00";   --不變
                BA:="01";   --8段遞減
            when "1010"=>
                RC<=(others=>'1');
                GC<=(others=>'1');
                BC<=(others=>'1');
                RA:="01";   --8段遞減
                GA:="01";   --8段遞減
                BA:="00";   --不變
            when "1011"=>
                RC<=(others=>'1');
                GC<=(others=>'1');
                BC<=(others=>'1');
                RA:="01";   --8段遞減
                GA:="00";   --不變
                BA:="01";   --8段遞減
            when "1100"=>
                RC<=(others=>'1');
                GC<=(others=>'1');
                BC<=(others=>'1');
                RA:="00";   --不變
                GA:="01";   --8段遞減
                BA:="01";   --8段遞減
            when "1101"=>
                RC<=(others=>'1');
```

```
                                GC<=(others=>'1');
                                BC<=(others=>'1');
                                RA:="01";   --8段遞減
                                GA:="01";   --8段遞減
                                BA:="01";   --8段遞減
                            when "1110"=>
                                RC<=(others=>'0'); --紅全暗
                                GC<=(others=>'1'); --綠全亮
                                BC<=(others=>'1'); --藍全亮
                                RA:="10";   --8段遞增
                                GA:="01";   --8段遞減
                                BA:="01";   --8段遞減
                            when others=>
                                RC<=(others=>'1'); --紅全亮
                                GC<=(others=>'0'); --綠全暗
                                BC<=(others=>'1'); --藍全亮
                                RA:="01";   --8段遞減
                                GA:="10";   --8段遞增
                                BA:="01";   --8段遞減
                        end case;
                        RGBcase:=RGBcase+1;
                    else
                        if RA="10" then
                            RC<=RC(6 downto 0) & '1';   --遞增
                        elsif RA="01" then
                            RC<='0' & RC(7 downto 1);   --遞減
                        end if;
                        if GA="10" then
                            GC<=GC(6 downto 0) & '1';   --遞增
                        elsif GA="01" then
                            GC<='0' & GC(7 downto 1);   --遞減
                        end if;
                        if BA="10" then
                            BC<=BC(6 downto 0) & '1';   --遞增
                        elsif BA="01" then
                            BC<='0' & BC(7 downto 1);   --遞減
                        end if;
                        cc:=cc-1;           --色階數 遞減
                    end if;
                end if;
            end if;
        else
            loadck<='0';
            LED_WS2812B_N<=LED_WS2812B_N+1;--輸出個數遞增
            delay<=80;
        end if;
    end if;
end process WS2812BP;

--除頻器--
Freq_Div:process(gckP31)
begin
    if rstP99='0' then              --系統重置
```

```
        FD<=(others=>'0');
        FD2<=(others=>'0');
        WS2812BCLK<='0';              --WS2812BN 驅動頻率
    elsif rising_edge(gckP31) then --50MHz
        FD<=FD+1;                     --除頻器:2 進制上數(+1)計數器
        if FD2=9 then                 --7~12
            FD2<=(others=>'0');
            WS2812BCLK<=not WS2812BCLK;
            --50MHz/20=2.5MHz T.=. 0.4us
        else
            FD2<=FD2+1;               --除頻器 2:2 進制上數(+1)計數器
        end if;
    end if;
end process Freq_Div;

end Albert;
```

CH3_WS2812B_2.vhd

```
--Ws2812B RGB_LED 霹靂燈 2
--EP3C16Q240C8 50MHz LEs:15,408 PINs:161 ,gckP31 ,rstP99

Library IEEE;                              --連結零件庫
Use IEEE.std_logic_1164.all;               --引用套件
Use IEEE.std_logic_unsigned.all;           --引用套件

entity CH3_WS2812B_2 is
    port(gckP31,rstP99:in std_logic;       --系統重置、系統時脈
         WS2812Bout:out std_logic);        --WS2812B_Di 信號輸出(184)
end entity CH3_WS2812B_2;

architecture Albert of CH3_WS2812B_2 is
    --WS2812B 驅動器--
    component WS2812B_Driver is
        port(
            WS2812BCLK,WS2812BRESET:in std_logic;--操作頻率,重置
            loadck:in std_logic;--載入 ck
            LEDGRBdata:in std_logic_vector(23 downto 0);--色彩資料
            reload,emitter,WS2812Bout:out std_logic
            --要求載入,發射狀態,發射輸出
            );
    end component;
    signal WS2812BCLK,WS2812BRESET:std_logic;
            --操作頻率,重置
    signal loadck,reload,emitter:std_logic;
            --載入 ck,要求載入,發射狀態
    signal LEDGRBdata:std_logic_vector(23 downto 0);--色彩資料

    signal FD:std_logic_vector(24 downto 0);--系統除頻器
```

```
    signal FD2:std_logic_vector(3 downto 0);--WS2812B_Driver除頻器
    signal SpeedS,WS2812BPCK:std_logic;
    --WS2812BP操作頻率選擇,WS2812BP操作頻率
    signal delay:integer range 0 to 127;        --停止時間
    signal LED_WS2812B_N:integer range 0 to 127;--WS2812B個數指標
    constant NLED:integer range 0 to 127:=29;  --30個RGB LED
    --WS2812B個數:61個(0~60)
    signal LED_WS2812B_shiftN:integer range 0 to 7;
    --WS2812B移位個數指標
    signal dir_LR:std_logic_vector(15 downto 0);   --方向控制
    type LED_T is array(0 to 7) of std_logic_vector(23 downto 0);
    --圖像格式

    --圖像
    signal LED_WS2812B_T8:LED_T:=(--G      R      B
                                  "000000001111111100000000",--紅
                                  "111111110000000000000000",--綠
                                  "000000000000000011111111",--藍
                                  "000000000000000000000000",--黑
                                  "111111111111111100000000",--黃
                                  "000000001111111111111111",--青
                                  "111111110000000011111111",--洋紅
                                  "111111111111111111111111" --白
                                  );

begin

--WS2812B驅動器--
WS2812BN: WS2812B_Driver port map(
            WS2812BCLK,WS2812BRESET,
            loadck,LEDGRBdata,reload,emitter,WS2812Bout);
WS2812BRESET<=rstP99;   --系統reset

--色彩資料--
LEDGRBdata<=LED_WS2812B_T8((LED_WS2812B_N+LED_WS2812B_shiftN) mod 8);

--WS2812BP操作頻率選擇
WS2812BPCK<=    FD(8) when SpeedS='0' else      --約97.7KHz
                FD(16)when dir_LR(7)='0' else   --約381Hz
                FD(18);--最慢速率(約95Hz)
WS2812BP:process(WS2812BPCK)
begin
    if rstP99='0' then
        LED_WS2812B_N<=0;           --從頭開始
        LED_WS2812B_shiftN<=0;      --移位0
        dir_LR<=(others=>'0');
        loadck<='0';
        SpeedS<='0';                --加快操作速率
    elsif rising_edge(WS2812BPCK) then
        if loadck='0' then          --等待載入
            loadck<=reload;
        elsif LED_WS2812B_N=NLED then
            SpeedS<='1';            --放慢操作速率
```

```
            if emitter='0' then     --已停止發射
                if delay/=0 then    --點亮時間&變化速率
                    delay<=delay-1; --時間遞減
                else
                    loadck<='0';    --reemitter
                    LED_WS2812B_N<=0;--從頭開始
                    dir_LR<=dir_LR+1;--方向控制
                    if dir_LR(4)='1' then
                        LED_WS2812B_shiftN<=LED_WS2812B_shiftN+1;
                        --移位遞增
                    else
                        LED_WS2812B_shiftN<=LED_WS2812B_shiftN-1;
                        --移位遞減
                    end if;
                    SpeedS<='0';    --加快操作速率
                end if;
            end if;
        else
            loadck<='0';
            LED_WS2812B_N<=LED_WS2812B_N+1;     --調整輸出色彩
            delay<=20;
        end if;
    end if;
end process WS2812BP;

--除頻器--
Freq_Div:process(gckP31)
begin
    if rstP99='0' then              --系統重置
        FD<=(others=>'0');
        FD2<=(others=>'0');
        WS2812BCLK<='0';            --WS2812BN驅動頻率
    elsif rising_edge(gckP31) then --50MHz
        FD<=FD+1;                   --除頻器:2進制上數(+1)計數器
        if FD2=9 then               --7~12
            FD2<=(others=>'0');
            WS2812BCLK<=not WS2812BCLK;--50MHz/20=2.5MHz T.=. 0.4us
        else
            FD2<=FD2+1;             --除頻器2:2進制上數(+1)計數器
        end if;
    end if;
end process Freq_Div;

end Albert;
```