

SSD1306_I2C2Wdriver4.vhd

```
--SSD1306_I2C_driver4:I2C 全功能版
--SSD1306_I2C 串列模式只能做寫入作業 Write mode
--Co:--1=word or 0=byte mode,byte mode 後不能再設回 word mode
--107.01.01 版

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 SSD1306_I2C2Wdriver4 is
    port( I2CCLK,RESET:in std_logic;           --系統時脈,系統重置
          SA0:in std_logic;                   --裝置碼位址
          CoDc:in std_logic_vector(1 downto 0);--Co & D/C
          Data_B:in std_logic_vector(7 downto 0);--資料輸入
          reLOAD:out std_logic;               --載入旗標:0 可載入 Data Byte
          LoadCK:in std_logic;               --載入時脈
          RWN:in integer range 0 to 15;      --嘗試讀寫次數
          I2Cok,I2CS:buffer std_logic;       --I2Cok,CS 狀態
          SCL:out std_logic; --介面 IO:SCL,如有接提升電阻時可設成 inout
          SDA:inout std_logic--SDA 輸入輸出
    );
end SSD1306_I2C2Wdriver4;

architecture Albert of SSD1306_I2C2Wdriver4 is
    signal Wdata:std_logic_vector(29 downto 0);--寫命令表
    signal Data_B_Bf:std_logic_vector(7 downto 0);--Data_B
    signal CoDc_Bf:std_logic_vector(1 downto 0);--CoDo
    signal Co,Buffer_Clr,Buffer_Empty:std_logic;
    signal I2Creset,SCLs,SDAs:std_logic;
    --失敗重來,SCL,SDAs->SDAout,SDAin-->SDA
    signal I:integer range 0 to 2;           --相位指標
    signal WN:integer range 0 to 29;         --寫入命令指標
    signal PN:integer range 0 to 29;         --錯誤暫停時間
    signal RWNS:integer range 0 to 15;       --嘗試讀寫次數計數器

begin

    SDA<='0' when SDAs='0' else 'Z';--SDA bus 控制

    SCL<='0' when SCLs='0' else '1';
    --介面 IO:SCL,如有接提升電阻時可設成 inout
    --SCL<='0' when SCLs='0' else 'Z';

    reLOAD<=Buffer_Empty or Buffer_Clr;
    Data_in:process(LoadCK,Reset)
    Begin
        if reset='0' or Buffer_Clr='1' then
            Buffer_Empty<='0';
        elsif rising_edge(LoadCK) then
            Data_B_Bf<=Data_B;
        end if;
    end process;
end architecture;
```

```

        CoDc_Bf<=CoDc;
        Buffer_Empty<='1';
        --Buffer_Empty='1'表示已有資料寫入(尚未傳出)
    end if;
end process Data_in;

main:process(I2CCLK,RESET)
begin
    if RESET='0' then
        Wdata<='0'&"011110"&SA0&'0'&'1'&CoDc&"000000"&'1'&Data_B&'1'&"00";
        --起始 裝置碼 SA0 寫入 ack Control 位元組 ack 寫入資料 ack P
        --(0)沒用到,結束碼
        --若 Co=1,則為 word mode(16bit):
        --(Control byte +Data byte)+(Control byte +Data byte),
        --下一筆放入 Wdata(10 downto 3)<=Data_B,WN 再從 19 起
        --若 Co=0,則為 byte mode(8bit):
        --Control byte(只有1次)+ Data byte.....,
        --下一筆放入 Wdata(10 downto 3)<=Data_B,WN 再從 10 起
        Co<=CoDc(1);--1=word or 0=byte mode

        I<=0;                --設 0 相位
        WN<=29;             --設寫入執行點

        SCLs<='1';         --設 I2C 為閒置
        SDAs<='1';         --設 I2C 為閒置
        I2CS<='0';         --設無狀態
        I2CoK<='0';        --設未完成旗標

        RWNS<=RWNS;        --嘗試讀寫次數
        PN<=29;            --錯誤暫停時間
        I2Creset<='0';     --清除重新執行旗標
        Buffer_Clr<='0';

    elsif rising_edge(I2CCLK) then
        Buffer_Clr<='0';
        if I2CoK='0' Then --尚未完成
            --失敗再嘗試
            if I2Creset='1' then --重新起始
                SCLs<='1';      --bus 暫停
                SDAs<='1';      --bus 暫停
                I<=0;WN<=29;    --錯誤回復執行點
                if PN=0 then    --暫停時間
                    PN<=29;     --重設錯誤暫停時間
                    I2Creset<='0'; --取消重新執行旗標
                    RWNS<=RWNS-1; --嘗試次數
                    if RWNS<=1 then --嘗試次數已用完
                        I2CoK<='1'; --完成
                        I2CS<='1'; --失敗
                    end if;
                else
                    PN<=PN-1;    --暫停時間倒數
                end if;
            else -- RW='0' --OLED 串列模式只能做寫入作業
                if WN=0 then    --結束點

```

```

SDAs<='1';      --Stop
I2CoK<='1';    --結束寫入(成功)
else
    I<=I+1;      --下一相位
    case I is
        when 0 =>  --0 相位
            SDAs<=Wdata(WN); --位元輸出
        when 1 =>  --1 相位
            SCLs<='1';  --SCK 拉高
            WN<=WN-1;    --下一 bit
            if WN=20 or WN=11 or WN=2 then --測 ACK 點
                if WN=20 then
                    --ACK 載入--第一次發現 ACK 錯誤時才重新執行
                    I2Creset<=SDA;
                    --SSD1306 的 ACK(低態:正常,高態:錯誤)
                elsif SDA='1' then --讀 SSD1306 的 ACK
                    I2CoK<='1'; --結束寫入(失敗)
                    I2CS<='1';  --失敗
                end if;
            end if;
        when others => --2 相位
            SCLs<='0';  --SCK 下拉
            I<=0;        --回 0 相位
            if WN=1 then
                if Buffer_Empty='1' then
                    --下一筆已經進來
                    Wdata(10 downto 3)<=Data_B_Bf;
                    --下一筆載入
                    Wdata(19 downto 18)<=CoDc_Bf;
                    if Co='1' then --word mode
                        Co<=CoDc_Bf(1);
                        WN<=19; --新執行點
                    else --byte mode
                        WN<=10; --新執行點
                    end if;
                    Buffer_Clr<='1'; --清除 buffer
                end if;
            end if;
        end case;
    end if;
end if;
end if;
end process;

end Albert;

```

CH7_OLED_1.vhd

```
--oLED 測試 (SCL=50、SDA=52)
--107.01.01 版
--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 CH7_OLED_1 is
port (gckp31,rstP99:in std_logic;             --系統頻率,系統 reset
      --oLED SSD1306 128x64
      oLED_SCL:out std_logic;                 --介面 IO:SCL(50)
      oLED_SDA:inout std_logic                --介面 IO:SDA,有接提升電阻(52)
    );
end entity CH7_OLED_1;

architecture Albert of CH7_OLED_1 is
  --oLED SSD1306 Driver --107.01.01 版
  component SSD1306_I2C2Wdriver4 is
    port (I2CCLK,RESET:in std_logic;           --系統時脈,系統重置
          SA0:in std_logic;                   --裝置碼位址
          CoDc:in std_logic_vector(1 downto 0); --Co & D/C
          Data_B:in std_logic_vector(7 downto 0); --資料輸入
          reLOAD:out std_logic;               --載入旗標:0 可載入 Data Byte
          LoadCK:in std_logic;               --載入時脈
          RWN:in integer range 0 to 15;       --嘗試讀寫次數
          I2Cok,I2CS:buffer std_logic;        --I2Cok,CS 狀態
          SCL:out std_logic;                  --介面 IO:SCL,如有接提升電阻時可設成 inout
          SDA:inout std_logic;                --SDA 輸入輸出
    );
  end component SSD1306_I2C2Wdriver4;
  signal oLED_I2CCLK,oLED_RESET:std_logic;    --系統時脈,系統重置
  signal oLED_SA0:std_logic:='0';            --裝置碼位址
  signal oLED_CoDc:std_logic_vector(1 downto 0); --Co & D/C
  signal oLED_Data_B:std_logic_vector(7 downto 0); --資料輸入
  signal oLED_reLOAD:std_logic;               --載入旗標:0 可載入 Data Byte
  signal oLED_LoadCK:std_logic;               --載入時脈
  signal oLED_RWN:integer range 0 to 15;      --嘗試讀寫次數
  signal oLED_I2Cok,oLED_I2CS:std_logic;      --I2Cok,CS 狀態

  --oLED 指令&資料表格式:
  type oLED_T is array (0 to 38) of std_logic_vector(7 downto 0);
  signal oLED_RUNT:oLED_T;
  --oLED=0:oLED 初始化 128x64
  constant oLED_IT:oLED_T:= (
    X"26",--0 長度
    X"AE",--1 display off
    X"D5",--2 設定除頻比及振盪頻率
    X"80",--3 [7:4]振盪頻率,[3:0]除頻比
    X"A8",--4 設 COM N 數
  )
end architecture Albert;
```

```

X"3F",
--5 1F:32COM(COM0~COM31 N=32),3F:64COM(COM0~COM31 N=64)
X"40",--6 設開始顯示行:0(SEG0)
X"E3",
--X"A1",--7 non Remap(column 0=>SEG0),A1 Remap(column 127=>SEG0)
X"C8",
--8 掃描方向:COM0->COM(N-1) COM31,C8:COM(N-1) COM31->COM0
X"DA",--9 設 COM Pins 配置
X"12",
--10 02:順配置(Disable COM L/R remap)
--12:交錯配置(Disable COM L/R remap)
--22:順配置(Enable COM L/R remap)
--32:交錯配置(Enable COM L/R remap)
X"81",--11 設對比
X"EF",--12 越大越亮
X"D9",--13 設預充電週期
X"F1",--14 [7:4]PHASE2,[3:0]PHASE1
X"DB",--15 設 Vcomh 值
X"30",
--16 00:0.65xVcc,20:0.77xVcc,30:0.83xVcc
X"A4",
--17 A4:由 GDDRAM 決定顯示內容,A5:全部亮(測試用)
X"A6",
--18 A6:正常顯示(1 亮 0 不亮),A7 反相顯示(0 亮 1 不亮)
X"D3",--19 設顯示偏移量 Offset
X"00",--20 00
X"E3",
--X"20",--21 設 GDDRAM pointer 模式
X"E3",
--X"02",--22 00:水平模式, 01:垂直模式,02:頁模式
--頁模式 column start address=[higher nibble,lower nibble] [00]
X"E3",
--X"00",--23 頁模式下設 column start address(lower nibble):0
X"E3",
--X"10",--24 頁模式下設 column start address(higher nibble):0
X"E3",
--X"B0",--25 頁模式下設 Page start address
X"20",--26 設 GDDRAM pointer 模式
X"00",
--27 00:水平模式, 01:垂直模式,02:頁模式
X"21",--28 水平模式下設行範圍:
X"00",
--29 行開始位置 0(Column start address)
X"7F",
--30 行結束位置 127(Column end address)
X"22",--31 水平模式下設頁範圍:
X"00",
--32 頁開始位置 0(Page start address)
X"07",
--33 頁結束位置 3(Page end address)
X"A1",
--34 non Remap(column 0=>SEG0),A1 Remap(column 127=>SEG0)
X"8D",--35 設充電 Pump

```



```

times<=200;
when "001" =>    --001 全亮 轉 垂直線
    OLEDtestM<="010";
    --0010 Vline:垂直線操作功能
    times<=0;
when "010"|"011"|"100"=>
    --0010 011 0100:Vline 垂直線操作功能
    times<=0;    --重設計時
    if Vline=0 and RL='1' then --功能該切換了
        RL<='0';--方向重設
        if OLEDtestM="010" then
            OLEDtestM<="011";--顯示資料通道選擇
        elsif OLEDtestM="011" then
            OLEDtestM<="100";--顯示資料通道選擇
        else
            OLEDtestM<="101";
            --顯示資料通道選擇及水平線操作變換
            Hline<=(Hline'range=>'0')+ '1';
        end if;
    elsif Vline=127 and RL='0' then--該變方向了
        RL<='1';        --方向變換
    else
        if RL='0' then
            Vline<=Vline+1; --L->R
        else
            Vline<=Vline-1; --R->L
        end if;
    end if;
when "101"=>    --Hline 水平線操作
    times<=5;    --重設計時
    if HN=0 and RL='1' then
        RL<='0';--方向重設
        OLEDtestM<="110";    --水平線操作變換
    elsif HN=63 and RL='0' then--該變方向了
        RL<='1';--方向變換
        HN<=64; --設 64 次
    else
        if RL='0' then
            Hline<=Hline(62 downto 0) & '0';
            --U->D
            HN<=HN+1;
        else
            Hline<='0' & Hline(63 downto 1);
            --D->U
            HN<=HN-1;
        end if;
    end if;
when others =>
    times<=5;    --重設計時
    if HN=0 and RL='1' then
        RL<='0';        --方向重設
        OLEDtestM<="000";    --功能重來
        not01<=not not01;    --反相操作

```

```

        times<=200;          --重設計時
    elsif HN=128 and RL='0' then--該變方向了
        RL<='1';            --方向變換
    else
        if RL='0' then
            Hline<=Hline(62 downto 0)&not Hline(63);
            --U->D
            HN<=HN+1;
        else
            Hline<=not Hline(0)&Hline(63 downto 1);
            --D->U
            HN<=HN-1;
        end if;
    end if;
end case;
end if;
else
    OLEDset_P_RESET<='1';  --重啟 OLEDset_P
end if;
end if;
end process oLED_test_Main;

--OLED 顯示器
--頁顯示資料解碼
--10 垂直線 解碼
GDDRAM2<="11111111" when Vline=oLED_DATA_POINTER else
    "00000000";
GDDRAM3<="11111111" when Vline>=oLED_DATA_POINTER else
    "00000000";
GDDRAM4<="11111111" when (127-Vline)<=oLED_DATA_POINTER else
    "00000000";

--11 水平線 解碼
GDDRAM5<= Hline(7 downto 0)  when GDDRAM_i=0 else
    Hline(15 downto 8)  when GDDRAM_i=1 else
    Hline(23 downto 16) when GDDRAM_i=2 else
    Hline(31 downto 24) when GDDRAM_i=3 else
    Hline(39 downto 32) when GDDRAM_i=4 else
    Hline(47 downto 40) when GDDRAM_i=5 else
    Hline(55 downto 48) when GDDRAM_i=6 else
    Hline(63 downto 56);

--顯示資料通道選擇--
with oLEDtestM select
GDDRAMo1<= "00000000" when "000",    --清除
    "11111111" when "001",    --填滿
    GDDRAM2 when "010",    --垂直線掃瞄
    GDDRAM3 when "011",    --垂直區塊掃瞄
    GDDRAM4 when "100",    --垂直區塊掃瞄
    GDDRAM5 when others;    --水平線、區塊掃瞄

GDDRAMo<=GDDRAMo1 when not01='0' else not GDDRAMo1; --反相解碼

```



```

--OLEDset_P--
--OLED 掃瞄管控
OLEDset_P:process(gckP31)
begin
    if OLEDset_P_RESET='0' then          --OLED 掃瞄管控重置
        OLED_P_RESET<='0';                --OLED_P 重置
        OLEDset_P_ok<='0';                --OLED 掃瞄管控尚未完成
    elsif rising_edge(gckP31) then
        if OLEDset_P_ok='0' then          --OLED 掃瞄管控尚未完成
            if OLED_P_RESET='1' then      --OLED_P 已啟動
                if OLED_P_ok='1' then     --OLED_P 已完成
                    OLEDset_P_ok<='1';    --OLED 掃瞄管控已完成
                end if;
            else
                OLED_P_RESET<='1';        --啟動 OLED_P
            end if;
        end if;
    end if;
end process OLEDset_P;

--OLED_P--
--          命令          顯示資料
OLED_Data_B<=OLED_RUNT(OLED_COM_POINTER) when OLED_CoDc="10"
                                         else GDDRAMo;

OLED_I2CCLK<=FD(3); --OLED 操作速率

OLED_P:process(gckP31,OLED_P_RESET)
    variable SW:Boolean;                --狀態控制旗標
begin
    if OLED_P_RESET='0' then
        OLED_RESET<='0';                --SSD1306_I2C2Wdriver2 重置
        OLED_RUNT<=OLED_IT;              --OLED 初始化設定表
        OLED_COM_POINTER<=OLED_COM_POINTERS;--命令起點
        OLED_DATA_POINTER<=0;
        GDDRAM_i<=0;                     --GDDRAM 指標 i
        OLED_P_ok<='0';                  --OLED_P 完成指標
        SW:=True;                         --載入狀態旗標
        OLED_CoDc<="10";                 --word mode ,command
    elsif rising_edge(gckP31) then
        OLED_LoadCK<='0';
        if OLED_RUNT(0)>=OLED_COM_POINTER then --傳送命令
            if OLED_RESET='0' then
                OLED_RESET<='1';          --啟動 SSD1306_I2C2Wdriver2
            elsif SW=True then
                OLED_COM_POINTER<=OLED_COM_POINTER+1;
                SW:=False;
            elsif OLED_reLOAD='0' then --載入
                OLED_LoadCK<='1';
                SW:=True;
            end if;
        elsif OLED_CoDc="10" then --切換成 byte 模式,連續傳送顯示資料
            OLED_CoDc<="01";              --byte mode,display data
            SW:=True;
        end if;
    end if;
end process OLED_P;

```

```

        elsif GDDRAM_i<8 then      --傳送顯示資料(畫面更新)
            if OLED_RESET='0' then --尚未啟動 SSD1306_I2C2Wdriver2
                OLED_RESET<='1';  --啟動 SSD1306_I2C2Wdriver2
                SW:=False;
            else
                if OLED_reLOAD='0' then --等待載入
                    if SW then        --載入
                        OLED_LoadCK<='1';
                        SW:=False;
                    else
                        OLED_DATA_POINTER<=OLED_DATA_POINTER+1;
                        --下一行
                        if OLED_DATA_POINTER=127 then --資料換頁
                            GDDRAM_i<=GDDRAM_i+1;    --下一頁
                        end if;
                        SW:=True;    --準備載入
                    end if;
                end if;
            end if;
        else
            OLED_P_ok<=OLED_I2Cok; --等待 SSD1306_I2C2Wdriver4 結束
        end if;
    end if;
end process OLED_P;

--除頻器--
Freq_Div:process(gckP31)          --系統頻率 gckP31:50MHz
begin
    if rstP99='0' then            --系統重置
        FD<=(others=>'0');        --除頻器:歸零
    elsif rising_edge(gckP31) then --50MHz
        FD<=FD+1;                --除頻器:2 進制上數(+1)計數器
    end if;
end process Freq_Div;

end Albert;

```

CH7_OLED_2.vhd

```

--OLED 測試:開機畫面(SCL=50,SDA=52)
--107.01.01 版
--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 CH7_OLED_2 is
    port (gckp31,rstP99:in std_logic; --系統頻率,系統 reset

```

```

--oLED SSD1306 128x64
oLED_SCL:out std_logic;      --介面 IO:SCL
oLED_SDA:inout std_logic     --介面 IO:SDA,有接提升電阻
);
end entity CH7_OLED_2;

architecture Albert of CH7_OLED_2 is
--oLED SSD1306 Driver --107.01.01 版
component SSD1306_I2C2Wdriver4 is
    port( I2CCLK,RESET:in std_logic; --系統時脈,系統重置
          SA0:in std_logic;          --裝置碼位址
          CoDc:in std_logic_vector(1 downto 0); --Co & D/C
          Data_B:in std_logic_vector(7 downto 0);--資料輸入
          reLOAD:out std_logic;      --載入旗標:0 可載入 Data Byte
          LoadCK:in std_logic;      --載入時脈
          RWN:in integer range 0 to 15;--嘗試讀寫次數
          I2Cok,I2CS:buffer std_logic;--I2Cok,CS 狀態
          SCL:out std_logic;        --介面 IO:SCL,如有接提升電阻時可設成 inout
          SDA:inout std_logic      --SDA 輸入輸出
    );
end component SSD1306_I2C2Wdriver4;
signal oLED_I2CCLK,oLED_RESET:std_logic; --系統時脈,系統重置
signal oLED_SA0:std_logic:='0';          --裝置碼位址
signal oLED_CoDc:std_logic_vector(1 downto 0); --Co & D/C
signal oLED_Data_B:std_logic_vector(7 downto 0);--資料輸入
signal oLED_reLOAD:std_logic; --載入旗標:0 可載入 Data Byte
signal oLED_LoadCK:std_logic; --載入時脈
signal oLED_RWN:integer range 0 to 15; --嘗試讀寫次數
signal oLED_I2Cok,oLED_I2CS:std_logic; --I2Cok,CS 狀態

--oLED 指令&資料表格式:
type oLED_T is array (0 to 38) of std_logic_vector(7 downto 0);
signal oLED_RUNT:oLED_T;
--oLED=0:oLED 初始化 128x64
constant oLED_IT:oLED_T:= (
    X"26",--0 長度
    X"AE",--1 display off
    X"D5",--2 設定除頻比及振盪頻率
    X"80",--3 [7:4]振盪頻率,[3:0]除頻比
    X"A8",--4 設 COM N 數
    X"3F",
    --5 1F:32COM(COM0~COM31 N=32),3F:64COM(COM0~COM31 N=64)
    X"40",--6 設開始顯示行:0(SEG0)
    X"E3",
    --X"A1",--7 non Remap(column 0=>SEG0),A1 Remap(column 127=>SEG0)
    X"C8",
    --8 掃描方向:COM0->COM(N-1) COM31,C8:COM(N-1) COM31->COM0
    X"DA",--9 設 COM Pins 配置
    X"12",
    --10 02:順配置(Disable COM L/R remap)
    --12:交錯配置(Disable COM L/R remap)
    --10 22:順配置(Enable COM L/R remap)
    --10 32:交錯配置(Enable COM L/R remap)

```

```

X"81",--11 設對比
X"EF",--12 越大越亮
X"D9",--13 設預充電週期
X"F1",--14 [7:4] PHASE2,[3:0] PHASE1
X"DB",--15 設 Vcomh 值
X"30",
--16 00:0.65xVcc,20:0.77xVcc,30:0.83xVcc
X"A4",
--17 A4:由 GDDRAM 決定顯示內容,A5:全部亮(測試用)
X"A6",
--18 A6:正常顯示(1 亮 0 不亮),A7 反相顯示(0 亮 1 不亮)
X"D3",--19 設顯示偏移量 Offset
X"00",--20 00
X"E3",
--X"20",--21 設 GDDRAM pointer 模式
X"E3",
--X"02",--22 00:水平模式, 01:垂直模式,02:頁模式
--頁模式 column start address=[higher nibble,lower nibble] [00]
X"E3",
--X"00",--23 頁模式下設 column start address(lower nibble):0
X"E3",
--X"10",--24 頁模式下設 column start address(higher nibble):0
X"E3",--X"B0",
--25 頁模式下設 Page start address
X"20",--26 設 GDDRAM pointer 模式
X"00",
--27 00:水平模式, 01:垂直模式,02:頁模式
X"21",--28 水平模式下設行範圍:
X"00",
--29 行開始位置 0(Column start address)
X"7F",
--30 行結束位置 127(Column end address)
X"22",--31 水平模式下設頁範圍:
X"00",
--32 頁開始位置 0(Page start address)
X"07",
--33 頁結束位置 3(Page end address)
X"A1",
--34 non Remap(column 0=>SEG0),A1 Remap(column 127=>SEG0)
X"8D",--35 設充電 Pump
X"14",--36 14:開啟,10:關閉
X"AF",--37 display on
X"E3" );--38 nop

```

```

signal OLED_COM_POINTER:integer range 0 to 63;--命令操作指標
signal OLED_COM_POINTERs:integer range 0 to 63;--命令操作指標
signal OLED_DATA_POINTER:integer range 0 to 127;--行碼
signal GDDRAM_i:integer range 0 to 15;          --oled 頁碼
signal GDDRAMo,GDDRAM6:std_logic_vector(7 downto 0);--通道圖資

signal FD:std_logic_vector(24 downto 0);        --除頻器
signal oLED_P_RESET,oLED_P_ok:std_logic;        --oLED_P 重置、完成
signal N:integer range 0 to 15;                  --操作次數

```

```

signal show_N:integer range 0 to 15;          --功能
signal OLEDtestM:std_logic_vector(1 downto 0);
--OLED 圖資通道選擇
signal OLEDset_P_RESET,OLEDset_P_ok,sw:std_logic;
--OLEDset_P 重置、完成 , 切換旗標
signal times:integer range 0 to 2047;        --停止時間

--OLED 128*64 開機畫面
type OLED_T1 is array (0 to 127) of std_logic_vector(63 downto
0);

constant OLED_screenShow:OLED_T1:= (
X"0000000000000000",X"0000000000000000",X"0000000000000000",
X"0000000000000000",X"0000000000000000",X"0020400000000000",
X"0020400000000000",X"0020401FFE000000",
X"0020401FFE000000",X"0000407FE0000000",X"0000407FE0000000",
X"000441FF80000000",X"000441FF80000000",X"000407FE00000000",
X"000407FE00000000",X"00001FFE00000000",
X"00001FFE00000000",X"001FFFFFF800000000",X"001FFFFFF800000000",
X"0019FFFFE0000000",X"0019FFFFE0000000",X"00007FFFFE00000000",
X"00007FFFF8000000",X"00001FFFFF80000000",
X"00001FFFFF80000000",X"00007FFFFE00000000",X"00007FFFFE00000000",
X"001FFFFFFFFF000",X"001FFFFFFFFF000",X"001807FFFFFFFFC00",
X"001807FFFFFFFFC00",X"000001FFFFFFFF9C00",
X"000001FFFFFFFF9C00",X"0000403FFFFFFFFC00",X"0000403FFFFFFFFC00",
X"0000400061FFFC00",X"0000400061FFFC00",X"00004001E19FFC00",
X"00004001E19FFC00",X"00004000019FFC00",
X"00004000019FFC00",X"00004000019FFC00",X"00004000019FFC00",
X"00004000001FFC00",X"00004000001FFC00",X"00004000001FF000",
X"00104000001FF000",X"0000400000000000",
X"0000400000000000",X"0000000000000000",X"0000000000000000",
X"0000000000000000",X"0000000000000000",X"0000000000000000",
X"0000000000000000",X"0000000000000000",
X"0000000000000000",X"0000000000000000",X"0000000000000000",
X"0000000000000000",X"0000000000000000",X"0100000000000000",
X"0080000000000000",X"0060000000000000",
X"0018004000000000",X"0080800400000000",X"1000104000001FF0",
X"3000004000001FF0",X"1FFF804000001FFC",X"0000804000001FFC",
X"00000040000019FFC",X"00080040000019FFC",
X"00100040000019FFC",X"00600040000019FFC",X"00C0004001E19FFC",
X"0180004001E19FFC",X"000000400061FFFC",X"186100400061FFFC",
X"0E2100403FFFFFFFFC",X"003F00403FFFFFFFFC",
X"1F118001FFFFFFFF9C",X"21110001FFFFFFFF9C",X"20001807FFFFFFFFC",
X"21451807FFFFFFFFC",X"22291FFFFFFFFFFFF0",X"261F1FFFFFFFFFFFF0",
X"2031007FFFFE0000",X"3C01007FFFFE0000",
X"103F801FFFF80000",X"0141001FFFF80000",X"0670007FFFF80000",
X"0C20007FFFFE0000",X"000019FFFFE00000",X"201219FFFFE00000",
X"1FD61FFFFFFFF80000",X"055A9FFFFFFFF80000",
X"1553001FFE000000",X"325A001FFE000000",X"1FD60407FE000000",
X"00120407FE000000",X"00000441FF800000",X"1FEF8441FF800000",
X"2AAA80407FE00000",X"2AAA00407FE00000",
X"2AAA20401FFE0000",X"2AAB20401FFE0000",X"203A204000000000",
X"3800204000000000",X"007C000000000000",X"0183000000000000",

```

```

X"0300800000000000",X"0501400000000000",
X"05F9400000000000",X"0901200000000000",X"0901200000000000",
X"09FF200000000000",X"0911200000000000",X"0911200000000000",
X"0511400000000000",X"0501400000000000",
X"0300800000000000",X"0183000000000000",X"007C000000000000",
X"0000000000000000",X"0000000000000000",X"0000000000000000",
X"0000000000000000",X"0000000000000000");

begin
--oLED--
U1: SSD1306_I2C2Wdriver4 port map(
    oLED_I2CCLK,oLED_RESET,'0',oLED_CoDc,
    oLED_Data_B,oLED_reLOAD,oLED_LoadCK,
    3,oLED_I2Cok,oLED_I2CS,oLED_SCL,oLED_SDA);

oLED_test_Main:process(FD(17)) --oLED_test_Main 主控器操作速率
begin
    if rstP99='0' then                --系統重置
        OLEDtestM<="00";              --oLED 圖資通道及功能選擇
        OLEDset_P_RESET<='0';         --OLEDset_P 控制旗標:重置
        oLED_COM_POINTERS<=1;         --oLED 命令指標:重下命令
        N<=0;                          --操作次數預設
        show_N<=0;                    --功能 0 預設:全暗
        sw<='0';                      --切換 0 預設
        times<=200;                   --停止時間預設
    elsif rising_edge(FD(17)) then
        if OLEDset_P_ok='1' then      --等待 OLEDset_P 完成
            OLED_COM_POINTERS<=conv_integer(OLED_RUNT(0))+1;
            --oLED 命令指標:不再下命令
            times<=times-1;            --停止計時
            if times=0 then            --計時到
                OLEDset_P_RESET<='0';--oLED_P 控制旗標:重置
                case show_N is         --選功能
                    when 0 =>
                        OLEDtestM<="01";    --全亮
                        show_N<=1;
                        times<=200;          --重設計時
                    when 1 =>
                        OLEDtestM<="10";    --正常
                        show_N<=2;
                        times<=200;          --重設計時
                    when 2=>
                        OLEDtestM<="11";    --反白
                        show_N<=3;
                        times<=200;          --重設計時
                    when 3=>
                        --全亮<-->全暗
                        times<=25;           --重設計時
                        if N=10 then         --次數結束
                            OLEDtestM<="10"; --正常
                            show_N<=4;
                            N<=0;           --次數歸零
                        else
                            N<=N+1;         --次數遞增

```

```

sw<=not sw;          --切換
if sw='0' then
    OLEDtestM<="01";--全亮
else
    OLEDtestM<="00";--全暗
end if;
end if;
when 4=>              --正常<-->全暗
    times<=50;        --重設計時
    if N=10 then      --次數結束
        OLEDtestM<="11";  --反白
        show_N<=5;
        N<=0;         --次數歸零
    else
        N<=N+1;        --次數遞增
        sw<=not sw;    --切換
        if sw='0' then
            OLEDtestM<="10";--正常
        else
            OLEDtestM<="00";--全暗
        end if;
    end if;
when 5=>              --反白<-->全暗
    times<=50;        --重設計時
    if N=10 then      --次數結束
        OLEDtestM<="10";  --正常
        show_N<=6;
        N<=0;         --次數歸零
    else
        N<=N+1;        --次數遞增
        sw<=not sw;    --切換
        if sw='0' then
            OLEDtestM<="11";--反白
        else
            OLEDtestM<="00";--全暗
        end if;
    end if;
when 6=>              --正常<-->全亮
    times<=50;        --重設計時
    if N=10 then      --次數結束
        OLEDtestM<="11";  --反白
        show_N<=7;
        N<=0;         --次數歸零
    else
        N<=N+1;        --次數遞增
        sw<=not sw;    --切換
        if sw='0' then
            OLEDtestM<="10";--正常
        else
            OLEDtestM<="01";--全亮
        end if;
    end if;
when 7=>              --反白<-->全亮

```

```

times<=50;          --重設計時
if N=10 then        --次數結束
    OLEDtestM<="10";  --正常
    show_N<=8;
    N<=0;           --次數歸零
else
    N<=N+1;         --次數遞增
    sw<=not sw;     --切換
    if sw='0' then
        OLEDtestM<="11";--反白
    else
        OLEDtestM<="01";--全亮
    end if;
end if;
when others =>      --正常<-->反白
    times<=50;      --重設計時
    if N=10 then    --次數結束
        OLEDtestM<="00";  --全暗
        show_N<=0;
        N<=0;       --次數歸零
        times<=200;  --重設計時
    else
        N<=N+1;     --次數遞增
        sw<=not sw;  --切換
        if sw='0' then
            OLEDtestM<="10";--正常
        else
            OLEDtestM<="11";--反白
        end if;
    end if;
end case;
end if;
else
    OLEDset_P_RESET<='1';  --重啟 OLEDset_P
end if;
end if;
end process oLED_test_Main;

--oLED 顯示器
--頁顯示資料解碼
GDDRAM6<=OLED_screenShow(oLED_DATA_POINTER) (7 downto 0)
    when GDDRAM_i=0 else
        OLED_screenShow(oLED_DATA_POINTER) (15 downto 8)
    when GDDRAM_i=1 else
        OLED_screenShow(oLED_DATA_POINTER) (23 downto 16)
    when GDDRAM_i=2 else
        OLED_screenShow(oLED_DATA_POINTER) (31 downto 24)
    when GDDRAM_i=3 else
        OLED_screenShow(oLED_DATA_POINTER) (39 downto 32)
    when GDDRAM_i=4 else
        OLED_screenShow(oLED_DATA_POINTER) (47 downto 40)
    when GDDRAM_i=5 else
        OLED_screenShow(oLED_DATA_POINTER) (55 downto 48)

```



```

        when GDDRAM_i=6 else
            OLED_screenShow(oLED_DATA_POINTER)(63 downto 56);

--顯示資料通道選擇--
with oLEDtestM select
GDDRAMo<="00000000"    when "00",--全暗
            "11111111"    when "01",--全亮
            GDDRAM6        when "10",--正常
            not GDDRAM6    when "11";--反白

--OLEDset_P--
--OLED 掃描管控
OLEDset_P:process(gckP31)
begin
    if OLEDset_P_RESET='0' then          --OLED 掃描管控重置
        OLED_P_RESET<='0';               --OLED_P 重置
        OLEDset_P_ok<='0';               --OLED 掃描管控尚未完成
    elsif rising_edge(gckP31) then
        if OLEDset_P_ok='0' then          --OLED 掃描管控尚未完成
            if OLED_P_RESET='1' then      --OLED_P 已啟動
                if OLED_P_ok='1' then     --OLED_P 已完成
                    OLEDset_P_ok<='1';   --OLED 掃描管控已完成
                end if;
            else
                OLED_P_RESET<='1';        --啟動 OLED_P
            end if;
        end if;
    end if;
end process OLEDset_P;

--OLED_P--
--          命令          顯示資料
OLED_Data_B<=OLED_RUNT(OLED_COM_POINTER) when OLED_CoDc="10"
                                           else GDDRAMo;

OLED_I2CCLK<=FD(3); --OLED 操作速率

OLED_P:process(gckP31,OLED_P_RESET)
    variable SW:Boolean;                --狀態控制旗標
begin
    if OLED_P_RESET='0' then
        OLED_RESET<='0';                --SSD1306_I2C2Wdriver2 重置
        OLED_RUNT<=OLED_IT;              --OLED 初始化設定表
        OLED_COM_POINTER<=OLED_COM_POINTERS;--命令起點
        OLED_DATA_POINTER<=0;
        GDDRAM_i<=0;                     --GDDRAM 指標 i
        OLED_P_ok<='0';                  --OLED_P 完成指標
        SW:=True;                         --載入狀態旗標
        OLED_CoDc<="10";                 --word mode ,command
    elsif rising_edge(gckP31) then
        OLED_LoadCK<='0';
        if OLED_RUNT(0)>=OLED_COM_POINTER then --傳送命令
            if OLED_RESET='0' then
                OLED_RESET<='1';          --啟動 SSD1306_I2C2Wdriver2

```

```

        elsif SW=true then
            OLED_COM_POINTER<=OLED_COM_POINTER+1;
            SW:=false;
        elsif OLED_reLOAD='0' then --載入
            OLED_LoadCK<='1';
            SW:=True;
        end if;
        elsif OLED_CoDc="10" then --切換成 byte 模式,連續傳送顯示資料
            OLED_CoDc<="01"; --byte mode,display data
            SW:=True;
        elsif GDDRAM_i<8 then --傳送顯示資料(畫面更新)
            if OLED_RESET='0' then --尚未啟動 SSD1306_I2C2Wdriver2
                OLED_RESET<='1'; --啟動 SSD1306_I2C2Wdriver2
                SW:=False;
            else
                if OLED_reLOAD='0' then
                    if SW then --載入
                        OLED_LoadCK<='1';
                        SW:=False;
                    else
                        OLED_DATA_POINTER<=OLED_DATA_POINTER+1;
                        --下一行
                        if OLED_DATA_POINTER=127 then --資料換頁
                            GDDRAM_i<=GDDRAM_i+1;
                        end if;
                        SW:=True;
                    end if;
                end if;
            end if;
        else
            OLED_P_ok<=OLED_I2Cok;
        end if;
    end if;
end process OLED_P;

--除頻器--
Freq_Div:process(gckP31) --系統頻率 gckP31:50MHz
begin
    if rstP99='0' then --系統重置
        FD<=(others=>'0'); --除頻器:歸零
    elsif rising_edge(gckP31) then --50MHz
        FD<=FD+1; --除頻器:2 進制上數(+1)計數器
    end if;
end process Freq_Div;

end Albert;

```