

CH5_7SLED_1.vhd

```
--4 位數掃描式共陽極七段顯示器
--計數器:手動計量器
--106.12.30 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 CH5_7SLED_1 is
    port (gckp31,rstP99:in std_logic;--系統頻率,系統 reset(歸零)
          S1,S8:in std_logic;          --遞增按鈕(131),歸零(117)
          --4 位數掃描式顯示器
          SCANo:buffer std_logic_vector(3 downto 0);--掃描器輸出
          Disp7S:buffer std_logic_vector(7 downto 0)--計數位數解碼輸出
          );
end entity CH5_7SLED_1;

architecture Albert of CH5_7SLED_1 is
    signal FD:std_logic_vector(26 downto 0);  --系統除頻器
    type Disp7DataT is array(0 to 3) of integer range 0 to 9;--計數器格式
    signal Disp7Data:Disp7DataT;              --計數器
    signal scanP:integer range 0 to 3;        --掃描器指標
    signal S1S,S8S:std_logic_vector(2 downto 0);--防彈跳計數器
begin

    --計數器-----
    counter_P:process(FD(18))  --FD(18)約 95Hz
    begin
        if rstP99='0' or S8S(2)='1' then  --系統重置,歸零
            Disp7Data(3)<=0;  --計數器:千位歸零
            Disp7Data(2)<=0;  --計數器:百位歸零
            Disp7Data(1)<=0;  --計數器:十位歸零
            Disp7Data(0)<=0;  --計數器:個位歸零
        elsif rising_edge(FD(18)) then
            if S1S(1)='1' then --BCD 碼遞增
                if Disp7Data(0)/=9 then
                    Disp7Data(0)<=Disp7Data(0)+1;
                else Disp7Data(0)<=0;--調整個位數
                    if Disp7Data(1)/=9 then
                        Disp7Data(1)<=Disp7Data(1)+1;
                    else Disp7Data(1)<=0;--調整十位數
                        if Disp7Data(2)/=9 then
                            Disp7Data(2)<=Disp7Data(2)+1;
                        else Disp7Data(2)<=0;--調整百位數
                            if Disp7Data(3)/=9 then
                                Disp7Data(3)<=Disp7Data(3)+1;
                            else Disp7Data(3)<=0;--調整千位數
                                end if;
                            end if;
                        end if;
                    end if;
                end if;
            end if;
        end if;
    end process;
end architecture;
```

```

        end if;
    end if;
end if;
end if;
end process counter_P;

--4 位數掃描器--
scan_P:process(FD(17),rstP99) --FD(17)約 191Hz
begin
    if rstP99='0' then
        scanP<=0;          --位數取值指標
        SCANo<="1111";    --掃描信號 all off
    elsif rising_edge(FD(17)) then
        scanP<=scanP+1;    --位數取值指標遞增
        SCANo<=SCANo(2 downto 0) & SCANo(3);
        if scanP=3 then    --最後一位數了
            scanP<=0;      --位數取值指標重設
            SCANo<="1110"; --掃描信號重設
        end if;
    end if;
end process scan_P;

--BCD 碼解:共陽極七段顯示碼 pgfedcba
with Disp7Data(scanP) select --取出顯示值
    Disp7S<=
        "11000000" when 0, --0xC0
        "11111001" when 1, --0xF9
        "10100100" when 2, --0xA4
        "10110000" when 3, --0xB0
        "10011001" when 4, --0x99
        "10010010" when 5, --0x92
        "10000010" when 6, --0x82
        "11111000" when 7, --0xF8
        "10000000" when 8, --0x80
        "10010000" when 9, --0x90
        "11111111" when others; --不顯示

--防彈跳--
debouncer:process(FD(17)) --FD(17)約 191Hz
begin
    --S1 防彈跳
    if S1='1' then
        S1S<="000";
    elsif rising_edge(FD(17)) then
        S1S<=S1S+ not S1S(2);
    end if;
    --S8 防彈跳
    if S8='1' then
        S8S<="000";
    elsif rising_edge(FD(17)) then
        S8S<=S8S+ not S8S(2);
    end if;
end process;

```

```

--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;

```

CH5_7SLED_2.vhd

```

--4 位數掃描式共陽極七段顯示器
--倒時計時器 59 分 59 秒
--106.12.30 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 CH5_7SLED_2 is
    port(gckp31,rstP99:in std_logic;--系統頻率,系統 reset(歸零)
          S1,S2,S0:in std_logic;
          --分(131),秒(128)遞增按鈕,設定/暫停/倒時(117)按鈕
          --4 位數掃描式顯示器
          SCANo:buffer std_logic_vector(3 downto 0); --掃描器輸出
          Disp7S:buffer std_logic_vector(7 downto 0);--計數位數解碼輸出
          Dd:buffer std_logic
          --狀態顯示
          );
end entity CH5_7SLED_2;

architecture Albert of CH5_7SLED_2 is
    signal FD:std_logic_vector(26 downto 0); --系統除頻器
    signal Scounter:integer range 0 to 390625; --半秒計時器
    type Disp7DataT is array(0 to 3) of integer range 0 to 9;--顯示格式
    signal Disp7Data:Disp7DataT;              --顯示區
    signal scanP:integer range 0 to 3;        --掃描器指標
    signal S2S,S1S,S0S:std_logic_vector(2 downto 0);--防彈跳計數器
    signal M,S:integer range 0 to 59;         --分,秒
    signal Ss,M_S_P_clk:std_logic;           --1 秒,M_S_P 時脈操作
    signal MSs:std_logic_vector(1 downto 0); --設定 on/off
begin
    Disp7Data(3)<=M/10;                      --分十位
    Disp7Data(2)<=M mod 10;                  --分個位

```

```

Disp7Data(1)<=S/10;          --秒十位
Disp7Data(0)<=S mod 10;      --秒個位
Dd<='0' when MSs>0 else S;  --MSs=0 暫停, 恆亮; MSs=1 倒數, 閃秒

--倒數計時器--
M_S_P_clk<=FD(23) when MSs>0 else S;  --M_S_P 時脈選擇
M_S_P:process(M_S_P_clk)
begin
    if rstP99='0' then --系統重置, 歸零
        M<=0;          --分歸零
        S<=0;          --秒歸零
        MSs<="00";     --狀態切換控制
    elsif rising_edge(M_S_P_clk) then
        if S0S(2)='1' then --狀態進行切換
            if MSs=0 or MSs=2 then --計時轉設定 or 設定轉計時
                MSs<=MSs+1;      --切換
            end if;
        else --狀態轉換
            if MSs=1 or MSs=3 then --計時轉設定 or 設定轉計時
                MSs<=MSs+1;      --轉換: 轉可執行穩定狀態
            end if;
        end if;
        if MSs>0 then --狀態中
            if MSs=2 then --可設定
                if S1S(2)='1' then --調整分
                    if M=59 then
                        M<=0;
                    else
                        M<=M+1;
                    end if;
                end if;
                if S2S(2)='1' then --調整秒
                    if S=59 then
                        S<=0;
                    else
                        S<=S+1;
                    end if;
                end if;
            end if;
        elsif M/=0 or S/=0 then --時未到
            if S/=0 then --倒時計時
                S<=S-1;
            else
                S<=59;
                M<=M-1;
            end if;
        end if;
    end if;
end process M_S_P;

--秒信號產生器
S_G_P:process(FD(5))
begin

```

```

    if rstP99='0' or MSs>0 then--系統重置 or 重新計時
        Ss<='1';
        Scounter<=390625;      --半秒計時器預設
    elsif rising_edge(FD(5)) then--781250Hz
        Scounter<=Scounter-1;  --半秒計時器遞減
        if Scounter=1 then      --半秒到
            Scounter<=390625;  --半秒計時器重設
            Ss<=not Ss;        --1 秒狀態
        end if;
    end if;
end process S_G_P;

```

--4 位數掃描器--

```

scan_P:process(FD(17),rstP99)
begin
    if rstP99='0' then
        scanP<=0;              --位數取值指標
        SCANo<="1111";        --掃描信號 all off
    elsif rising_edge(FD(17)) then
        scanP<=scanP+1;        --位數取值指標遞增
        SCANo<=SCANo(2 downto 0)&SCANo(3);
        if scanP=3 then         --最後一位數了
            scanP<=0;          --位數取值指標重設
            SCANo<="1110";     --掃描信號重設
        end if;
    end if;
end process scan_P;

```

--BCD 碼解共陽極七段顯示碼 pgfedcba

```

with Disp7Data(scanP) select --取出顯示值
    Disp7S<=
        "11000000" when 0, --0xC0
        "11111001" when 1, --0xC0
        "10100100" when 2, --0xA4
        "10110000" when 3, --0xB0
        "10011001" when 4, --0x99
        "10010010" when 5, --0x92
        "10000010" when 6, --0x82
        "11111000" when 7, --0xF8
        "10000000" when 8, --0x80
        "10010000" when 9, --0x9
        "11111111" when others; --不顯示

```

--防彈跳--

```

debouncer:process(FD(17))
begin
    --S0 防彈跳
    if S0='1' then
        S0S<="000";
    elsif rising_edge(FD(17)) then
        S0S<=S0S+ not S0S(2);
    end if;
    --S1 防彈跳

```

```

        if S1='1' then
            S1S<="000";
        elsif rising_edge(FD(17)) then
            S1S<=S1S+ not S1S(2);
        end if;
        --S1 防彈跳
        if S2='1' then
            S2S<="000";
        elsif rising_edge(FD(17)) then
            S2S<=S2S+ not S2S(2);
        end if;
    end process;

    --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;

```

CH5_7SLED_3.vhd

```

--4 位數掃描式共陽極七段顯示器
--數位電子鐘 24 小時制
--106.12.30 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 CH5_7SLED_3 is
    port(gckp31,rstP99:in std_logic;--系統頻率,系統 reset(歸零)
          S1,S2,S8:in std_logic;
          --時(131),分(128)遞增按鈕,設定/暫停/倒時按鈕(117)
          --4 位數掃描式顯示器
          SCANo:buffer std_logic_vector(3 downto 0);--掃描器輸出
          Disp7S:buffer std_logic_vector(7 downto 0);--計數位數解碼輸出
          Dd:buffer std_logic          --狀態顯示
        );
end entity CH5_7SLED_3;

architecture Albert of CH5_7SLED_3 is
    signal FD:std_logic_vector(26 downto 0);  --系統除頻器

```

```

    signal Scounter:integer range 0 to 390625; --半秒計時器
    type Disp7DataT is array(0 to 3) of integer range 0 to 9;--顯示區格式
    signal Disp7Data:Disp7DataT;                --顯示區
    signal scanP:integer range 0 to 3;           --掃描器指標
    signal S2S,S1S,S8S:std_logic_vector(2 downto 0);--防彈跳計數器
    signal H:integer range 0 to 23;              --時
    signal M,S:integer range 0 to 59;  --分,秒
    signal Ss,E_Clock_P_clk:std_logic; --1秒,E_Clock_P 時脈操作
    signal MSs:std_logic_vector(1 downto 0);--設定 on/off
begin

    Disp7Data(3)<=H/10;          --時十位
    Disp7Data(2)<=H mod 10;      --時個位
    Disp7Data(1)<=M/10;          --分十位
    Disp7Data(0)<=M mod 10;      --分個位
    Dd<='0' when MSs>0 else Ss;--MSs=0 暫停,恆亮;MSs=1 倒數,閃秒

    --數位電子鐘 24 小時制--
    E_Clock_P_clk<=FD(23) when MSs>0 else Ss;  --E_Clock_P 時脈選擇
    E_Clock_P:process(E_Clock_P_clk)
    begin
        if rstP99='0' then          --系統重置,歸零
            M<=0;                    --分歸零
            S<=0;                    --秒歸零
            MSs<="00";              --狀態切換控制
        elsif rising_edge(E_Clock_P_clk) then
            if S8S(2)='1' then        --狀態進行切換
                if MSs=0 or MSs=2 then --計時轉設定 or 設定轉計時
                    MSs<=MSs+1;       --切換
                end if;
            else                      --狀態轉換
                if MSs=1 or MSs=3 then --計時轉設定 or 設定轉計時
                    MSs<=MSs+1;       --轉換:轉可執行穩定狀態
                end if;
            end if;
            if MSs>0 then              --狀態中
                if MSs=2 then          --可設定
                    if S1S(2)='1' then --調整時
                        if H=23 then
                            H<=0;
                        else
                            H<=H+1;
                        end if;
                    end if;
                end if;
                if S2S(2)='1' then     --調整分
                    if M=59 then
                        M<=0;
                    else
                        M<=M+1;
                    end if;
                end if;
                S<=0;                  --秒歸零
            end if;
        end if;
    end process;

```

```

        else
            if S/=59 then          --秒計時
                S<=S+1;
            else
                S<=0;
                if M/=59 then      --分計時
                    M<=M+1;
                else
                    M<=0;
                    if H/=23 then  --時計時
                        H<=H+1;
                    else
                        H<=0;
                    end if;
                end if;
            end if;
        end if;
    end if;
end process E_Clock_P;

--秒信號產生器 --
S_G_P:process(FD(5))
begin
    if rstP99='0' or MSs>0 then--系統重置 or 重新計時
        Ss<='1';
        Scounter<=390625;      --半秒計時器預設
    elsif rising_edge(FD(5)) then--781250Hz
        Scounter<=Scounter-1;  --半秒計時器遞減
        if Scounter=1 then      --半秒到
            Scounter<=390625;  --半秒計時器重設
            Ss<=not Ss;        --1 秒狀態
        end if;
    end if;
end process S_G_P;

--4 位數掃描器--
scan_P:process(FD(17),rstP99)
begin
    if rstP99='0' then
        scanP<=0;              --位數取值指標
        SCANo<="1111";        --掃描信號 all off
    elsif rising_edge(FD(17)) then
        scanP<=scanP+1;        --位數取值指標遞增
        SCANo<=SCANo(2 downto 0) & SCANo(3);
        if scanP=3 then         --最後一位數了
            scanP<=0;          --位數取值指標重設
            SCANo<="1110";     --掃描信號重設
        end if;
    end if;
end process scan_P;

--BCD 碼解共陽極七段顯示碼 pgfedcba
with Disp7Data(scanP) select --取出顯示值

```



```

Disp7S<=  "11000000" when 0, --0xC0
           "11111001" when 1, --0xF9
           "10100100" when 2, --0xA4
           "10110000" when 3, --0xB0
           "10011001" when 4, --0x99
           "10010010" when 5, --0x92
           "10000010" when 6, --0x82
           "11111000" when 7, --0xF8
           "10000000" when 8, --0x80
           "10010000" when 9, --0x90
           "11111111" when others; --不顯示

--防彈跳--
process (FD(17))
begin
    --S8 防彈跳
    if S8='1' then
        S8S<="000";
    elsif rising_edge(FD(17)) then
        S8S<=S0S+ not S8S(2);
    end if;
    --S1 防彈跳
    if S1='1' then
        S1S<="000";
    elsif rising_edge(FD(17)) then
        S1S<=S1S+ not S1S(2);
    end if;
    --S1 防彈跳
    if S2='1' then
        S2S<="000";
    elsif rising_edge(FD(17)) then
        S2S<=S2S+ not S2S(2);
    end if;
end process;

--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;

```