

DoReMi_S.vhd

```
--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 DoReMi_S is
    port (CLK, Sound_RESET, S_P_N:in std_logic;
           --系統時脈,系統重置,輸出預設
           ToneS:in integer range 0 to 37;      --音階
           BeatS:in std_logic_vector(9 downto 0); --節拍
           Soundend, Do_Re_Mio:out std_logic);   --音完成,音輸出
end DoReMi_S;

architecture Albert of DoReMi_S is
    signal Ftime:integer range 0 to 500000;      --節拍基準時間
    constant FtimeS:integer range 0 to 500000:=500000; --節拍預設值
    signal BeatN:std_logic_vector(9 downto 0); --節拍

    --12bit
    signal Ftone:integer range 0 to 31;           --音階基準時間
    constant FtoneS:integer range 0 to 31:=25; --音階預設值
    signal DoReMi:integer range 0 to 4095;        --音階計時
    signal Do_Re_Mi:std_logic;                    --音輸出
    --音階轉換表 12Bit
    type SFT is array (0 to 37) of integer range 0 to 4095; --12Bit
    constant Tone:SFT:=(
        0,--SOUND OFF
        3817,3610,3401,3215,3030,2865,2703,2551,2410,2273,2155,2024,
        1912,1805,1704,1608,1517,1433,1351,1276,1203,1136,1073,1012,
        956, 902, 851, 803, 759, 716, 676, 638, 602, 568, 536, 506,
        0);

begin

    --音階輸出--
    Do_Re_Mio<=Do_Re_Mi;

    DoReMi_Timer:process (CLK, Sound_RESET)
    begin
        if Sound_RESET='0' then
            Ftime<=FtimeS;          --節拍基準時間設定
            BeatN<=BeatS;           --節拍設定
            Ftone<=FtoneS;          --音階預設值
            DoReMi<=Tone(ToneS);    --音階轉換
            Do_Re_Mi<='1' xor S_P_N; --輸出預設
            Soundend<='0';          --音完成旗標預設未完成
        elsif rising_edge(CLK) then
            if BeatN/=0 then        --節拍未完成
```

```

--節拍產生器
if Ftime=1 then --Timer:節拍計時
    Ftime<=FtimeS; --節拍基準時間重設定
    if BeatN=1 then --節拍完成
        Soundend<='1'; --音完成旗標預設完成
    end if;
    BeatN<=BeatN-1; --數節拍
else
    Ftime<=Ftime-1; --節拍計時
end if;

--音階產生器
if Ftone=1 then --音階預除到了
    Ftone<=FtoneS; --音階預除重設
    if DoReMi/=0 then --非靜音
        if DoReMi=1 then --音階計時到了
            DoReMi<=Tone(ToneS); --音階轉換重設
            Do_Re_Mi<=not Do_Re_Mi; --輸出反相
        else
            DoReMi<=DoReMi-1; --數音階
        end if;
    end if;
else
    Ftone<=Ftone-1; --音階預除
end if;

else
    Soundend<='1'; --音完成旗標預設完成
    Do_Re_Mi<='1' xor S_P_N; --輸出預設
end if;
end if;
end Process DoReMi_Timer;

end Albert;

```

CH8_SOUND_1.vhd

```

--DoReMi、音樂 IC 測試
--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 CH8_SOUND_1 is
    port (gckp31,rstP99:in std_logic;--系統頻率,系統 reset
          S1,S2:in std_logic; --音樂 IC 按鈕、脈波式蜂鳴器按鈕
          --音樂 IC、外激式蜂鳴器輸出

```

```

        MusicIC,Do_Re_Mio:out std_logic
    );
end entity CH8_SOUND_1;

architecture Albert of CH8_SOUND_1 is
    --DoReMi_S Driver--
    component DoReMi_S is
        Port (CLK,Sound_RESET,S_P_N:in std_logic;    --系統時脈,系統重置
              ToneS:in integer range 0 to 37;        --音階
              BeatS:in std_logic_vector(9 downto 0); --節拍
              Soundend,Do_Re_Mio:out std_logic);      --音完成,音輸出
    end component;

    signal Sound_RESET,Soundend:std_logic;    --重置,音完成
    signal S_WAIT:integer range 0 to 3;        --等待音結束
    signal BeatS:std_logic_vector(9 downto 0); --節拍
    signal ToneS:integer range 0 to 37;        --音階

    signal FD:std_logic_vector(24 downto 0);    --除頻器
    signal UpDn,S_L_UD:std_logic;              --升降音階、節拍
    signal S_L:std_logic_vector(4 downto 0);    --節拍調整倍率
    signal S1S,S2S:std_logic_vector(2 downto 0); --防彈跳計數器
    signal MusicIC_on,Sound_main_reset,S2_off:std_logic:='0';
    --音樂 IC 輸出控制,Sound_main 重置,S2 控制

begin

    --DoReMi_S Driver--
    U1: DoReMi_S port map(
        gckp31,Sound_RESET,'1',ToneS,BeatS,Soundend,Do_Re_Mio);

    MusicIC<=MusicIC_on;--S1S(2);                --音樂 IC 輸出

    --節拍--
    BeatS<="0000001010" + S_L * "00101";    --0.1s~1.65s 節拍調整

    --按鈕操作--
    PB:process(FD(18))
    begin
        --音樂 IC 按鈕
        if rising_edge(FD(18)) then
            if S1S(2)='1' then
                --音樂 IC 輸出控制
                MusicIC_on<=not MusicIC_on;
            end if;
        end if;

        --外激式蜂鳴器按鈕
        if S2S=0 then                --S2 按鈕放開
            Sound_main_reset<=S2_off; --Sound_main on_off 控制
        elsif rising_edge(FD(18)) then
            if S2S(2)='1' then
                --Sound_main on_off 控制
                Sound_main_reset<=not Sound_main_reset;
            end if;
        end if;
    end process;
end architecture Albert;

```

```

        end if;
    end if;
end process;

--Sound_main--
Sound_main:process(FD(0))
begin
    if Sound_main_reset='0' then
        ToneS<=0; --音階預設:0
        S_L<="00000"; --節拍調整倍率預設:0
        UpDn<='1'; --升降音階預設:升
        S_L_UD<='1'; --升降節拍預設:升
        Sound_RESET<='0'; --DoReMi_S 重置
        S2_off<='0'; --控制 Sound_main_reset
    elsif rising_edge(FD(0)) then
        if S_L/="10000" then
            S2_off<='1'; --維持啟動 Sound_main_reset
            if Soundend='1' then --DoReMi_S 音結束了
                Sound_RESET<='0'; --DoReMi_S 重置
                if UpDn='1' then --升音階
                    ToneS<=ToneS+1; --升音階
                    if ToneS=36 then --最高一個了
                        UpDn<='0'; --改降音階
                    end if;
                else --降音階
                    ToneS<=ToneS-1; --降音階
                    if ToneS=1 then --最低一個了
                        UpDn<='1'; --改升音階
                        if S_L_UD='1' then --升節拍(加長)
                            S_L<=S_L+1; --加長
                            if S_L=3 then --最長了
                                S_L_UD<='0'; --改降節拍(變短)
                            end if;
                        else --降節拍
                            S_L<=S_L-1; --變短
                            if S_L=1 then --最短了
                                --Sound_main 結束了
                                S_L<="10000"; --防 S2 按鈕未放開
                                --回控 Sound_main_reset
                                S2_off<='0';
                            end if;
                        end if;
                    end if;
                end if;
            end if;
        else
            Sound_RESET<='1'; --啟動 DoReMi_S
        end if;
    end if;
end process Sound_main;

--防彈跳--
debouncer:process(FD(17))

```

```

begin
    --S1 防彈跳--音樂 IC 按鈕
    if S1='1' then
        S1S<="000";
    elsif rising_edge(FD(17)) then
        S1S<=S1S+ not S1S(2);
    end if;
    --S2 防彈跳--外激式蜂鳴器按鈕
    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;

```