```
-- Loss_Lock_counter - frontend
library IEEE;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
use ieee.std_logic_arith.all;
library synplify;
use synplify.attributes.all;


entity Loss_Lock_counter is port

(clock              : in std_logic;
resetn              : in std_logic;
Number_lost         : out std_logic_vector(7 downto 0);
QPLL_Lost_sync      : in std_logic);

end Loss_Lock_counter;

architecture rtl of Loss_Lock_counter is
attribute syn_radhardlevel of rtl : architecture is "tmr";


type state_values is (st0, st1, st2); --st2, st3, st4);
signal pres_state : state_values;
signal Num_lost   : std_logic_vector(7 downto 0);

begin
    -- fsm register
    state_reg: process (clock, resetn)

    begin

         if (resetn = '0') then
                pres_state <= st0;
                Number_lost <= "00000000";
                Num_lost <= "00000000";

         elsif clock'event AND clock = '1' then

         case pres_state is
            when st0 =>
                if QPLL_Lost_sync = '0' then -- QPLL lost sync
                   Num_lost <= Num_lost + 1; -- increment counter
                   pres_state  <= st1;
                else -- QPLL synced
                   pres_state  <= st0;
                   Num_lost <= Num_lost;
                end if;

            when st1 =>
                if QPLL_Lost_sync = '1' then -- QPLL regained sync
                   pres_state <= st2;
                else -- QPLL still lost
                   pres_state <= st1;
                end if;

            when st2 =>
                Number_lost <= Num_lost;
                pres_state <= st0;

        end case;

        end if;
        end process;
 end rtl;
```