```
-- regulator_reset - 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 regulator_reset is port

(clock              : in std_logic;
resetn              : in std_logic;
--state_out           : out std_logic;--_vector(1 downto 0);
reset_regulator_in  : in std_logic;
--FE_reset_OR         : out std_logic;
FE_reset_regulator  : out std_logic);

end regulator_reset;

architecture rtl of regulator_reset 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 load_timer : std_logic;
signal timer_en : std_logic;
--signal half_way : std_logic;
--signal timed_out : std_logic;
signal pulse_finish : std_logic;
signal timer : std_logic_vector(15 downto 0);

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

    begin

         if (resetn = '0') then
                pres_state <= st0;
                --state_out <= '0';
                FE_reset_regulator <= '1'; --reset the regulators during power on reset
                --FE_reset_OR <= '0';       -- as of 9/8/03
                load_timer <= '0';
                pulse_finish <= '0';
  --              timed_out <= '0';
                timer_en <= '0';
                timer <= "0000000000000000";

         elsif clock'event AND clock = '1' then

-------------------------------------------------
            --FE_reset_regulator <= '0';
            if load_timer = '1' then
                timer <= "0000000000000000";
            elsif timer_en = '1' then
                timer <= timer + 1;
            else
                timer <= timer;
            end if;

            if timer(15) = '1' then
                pulse_finish <= '1';
            else
                pulse_finish <= '0';
            end if;

--          if timer(14) = '1' then
```

```
--              half_way <= '1';
--          else
--              half_way <= '0';
--          end if;


--          if timer(15 downto 13) = "101" then
--              timed_out <= '1';
--          else
--              timed_out <= '0';
--          end if;
-------------------------------------------------

         case pres_state is
            when st0 =>
                --state_out <= '0';
                FE_reset_regulator <= '0';
--                FE_reset_OR <= '0';
                timer_en <= '0';
                load_timer <= '1';
                pulse_finish <= '0';
--                timed_out <= '0';
                if reset_regulator_in = '1' then
                   pres_state <= st1;
                else
                   pres_state <= st0;
                end if;

            when st1 =>
                --state_out <= '1';
                timer_en <= '1';
                load_timer <= '0';
                FE_reset_regulator <= '1';
                if pulse_finish = '1' then
                   pres_state <= st2;
                else
                   pres_state <= st1;
                end if;

            when st2 =>
                FE_reset_regulator <= '0';
                pres_state <= st0;

  --              timer_en <= '1';
  --              if half_way = '1' then
  --                 pres_state <= st3;
  --              else
  --                 pres_state <= st2;
  --              end if;

  --          when st3 =>
  --              state_out <= ("011");
 --               FE_reset_OR <= '1';
  --              timer_en <= '1';
 --                if pulse_finish = '1' then
 --                   pres_state <= st4;
 --                else
 --                   pres_state <= st3;
 --                end if;

 --           when st4 =>
 --               state_out <= ("100");
 --               FE_reset_OR <= '0';
 --               timer_en <= '1';
 --               if timed_out = '1' then
 --                  pres_state <= st0;
 --               else
 --                  pres_state <= st4;
```

```
 --                end if;

        end case;
        end if;

        end process;
 end rtl;
```