```
l_phi=> phi_s,enable_out=>en4) ;
OBS_4: observer GENERIC MAP(observernumber => x"000A")
   PORT MAP ( output=>add(4),clk=>clk_s,reset =>reset_s, enable_in =>en4,invariance_tau => tau_s,signa
l_phi=> phi_s,enable_out=>en5) ;
OBS_5: observer GENERIC MAP(observernumber => x"000A")
   PORT MAP ( output=>add(5),clk=>clk_s,reset =>reset_s, enable_in =>en5,invariance_tau => tau_s,signa
l_phi=> phi_s,enable_out=>en6) ;
OBS_6: observer GENERIC MAP(observernumber => x"000A")
   PORT MAP ( output=>add(6),clk=>clk_s,reset =>reset_s, enable_in =>en6,invariance_tau => tau_s,signa
l_phi=> phi_s,enable_out=>en7) ;
OBS_7: observer GENERIC MAP(observernumber => x"000A")
   PORT MAP ( output=>add(7),clk=>clk_s,reset =>reset_s, enable_in =>en7,invariance_tau => tau_s,signa
l_phi=> phi_s,enable_out=>en8) ;
OBS_8: observer GENERIC MAP(observernumber => x"000A")
   PORT MAP ( output=>add(8),clk=>clk_s,reset =>reset_s, enable_in =>en8,invariance_tau => tau_s,signa
l_phi=> phi_s,enable_out=>en9) ;
OBS_9: observer GENERIC MAP(observernumber => x"000A")
   PORT MAP ( output=>add(9),clk=>clk_s,reset =>reset_s, enable_in =>en9,invariance_tau => tau_s,signa
l_phi=> phi_s,enable_out=> next_obs_s) ;

-- <END_2>
--------------------------------------------------------------------------------


 --------------------------------------------------------------------------------
-- <BEGIN_3>
output_s <= and_reduce(add);
-- <END_3>
--------------------------------------------------------------------------------


--------------------------------FPGA OUTPUTS-----------------------------------
----------


 reset_s <= not KEY(0);

 GPIO(0) <= reset_s;
 GPIO(1) <= enable_s;
 GPIO(2) <= en1;
 GPIO(3) <= en2;
 GPIO(4) <= en3;
 GPIO(5) <= en4;
 GPIO(6) <= en5;
 GPIO(7) <= en6;
 GPIO(8) <= en7;
 GPIO(9) <= en8;
 GPIO(10) <= en9;
 GPIO(11) <= next_obs_s;
 GPIO(12) <= clk_g;
 GPIO(13) <= phi_s;
 GPIO(14)<= clk_s;

 GPIO(15) <= add(0);
 GPIO(16) <= add(1);
 GPIO(17) <= add(2);
 GPIO(18) <= add(3);
 GPIO(19) <= add(4);
 GPIO(20) <= add(5);
 GPIO(21) <= add(6);
 GPIO(22) <= add(7);
 GPIO(23) <= add(8);
 GPIO(24) <= add(9);
 GPIO(25)<= output_s;

 tau_s             <= std_logic_vector(to_unsigned(tau_range,8));

 --------------------------------SYNCHRONIZED-----------------------------------
------------
 sync:process(clk_s)
 begin
   if(clk_s'event and clk_s='1') then
     if reset_s ='0' then
       enable_s <= '1';
     else
       enable_s <= '0';
     end if;
   end if;
 end process;

end architecture;
```