```
----------------------------------------------------------------------------------
-- Company:
-- Engineer: Brandon Frazier
--
-- Create Date: 10/23/2024 10:05:29 AM
-- Design Name:
-- Module Name: ica1SM - Behavioral
-- Project Name:
-- Target Devices:
-- Tool Versions:
-- Description:
--
-- Dependencies:
--
-- Revision:
-- Revision 0.01 - File Created
-- Additional Comments:
--
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity ica1SM is
  Port (x, clk, rst: in std_logic;
        z: out std_logic);
end ica1SM;

architecture Behavioral of ica1SM is
    type stateType is (S0, S1, S2, S3, S4);
    signal presentState, nextState : stateType;
begin

clockProc : process(clk, rst)
begin
    if(rst = '1') then
        presentState <= S0;
        --nextState <= S0;
    elsif(rising_edge(clk)) then
        presentState <= nextState;
    end if;
end process clockProc;

nextStateProc : process(presentState, x)
begin
    case presentState is
        when S0 =>
            if(x = '0') then
                nextState <= S0;
            else
                nextState <= S1;
            end if;
        when S1 =>
            if(x = '0') then
                nextState <= S0;
            else
                nextState <= S2;
            end if;
        when S2 =>
            if(x = '0') then
                nextState <= S3;
            else
                nextState <= S2;
            end if;
        when S3 =>
            if(x = '0') then
                nextState <= S0;
            else
```

```
                nextState <= S4;
            end if;
        when S4 =>
            if(x = '0') then
                nextState <= S0;
            else
                nextState <= S2;
            end if;
        end case;
end process nextStateProc;

outputProc: process(presentState)
begin
    if(presentState = S4) then
        Z <= '1';
    else
        Z <= '0';
    end if;
end process outputProc;

end Behavioral;

```