----------------------------------------------------------------------------------

-- Company:

-- Engineer:

--

-- Create Date: 17:43:02 04/11/2017

-- Design Name:

-- Module Name: fsm\_demux\_impl - 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;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

---- Uncomment the following library declaration if instantiating

---- any Xilinx primitives in this code.

--library UNISIM;

--use UNISIM.VComponents.all;

entity fsm\_demux\_impl is

Port ( reset : in STD\_LOGIC;

x : in STD\_LOGIC\_VECTOR(1 downto 0);

clk : in STD\_LOGIC;

z : out STD\_LOGIC);

end fsm\_demux\_impl;

architecture Behavioral of fsm\_demux\_impl is

type state\_type is (S0, S1, S2, S3);

signal state, next\_state : state\_type;

begin

SYNC\_PROC : process (clk)

begin

if rising\_edge(clk) then

if (reset = '1') then

state <= S0;

else

state <= next\_state;

end if;

end if;

end process;

NEXT\_STATE\_DECODE : process (state, x)

begin

z <= '0';

case (state) is

when S0 =>

if (x = "00") then

z <= '1';

next\_state <= S0;

elsif (x = "01") then

z <= '0';

next\_state <= S1;

elsif (x = "10") then

z <= '0';

next\_state <= S2;

else

z <= '0';

next\_state <= S3;

end if;

when S1 =>

if (x = "00") then

z <= '0';

next\_state <= S0;

elsif (x = "01") then

z <= '1';

next\_state <= S1;

elsif (x = "10") then

z <= '0';

next\_state <= S2;

else

z <= '0';

next\_state <= S3;

end if;

when S2 =>

if (x = "00") then

z <= '0';

next\_state <= S0;

elsif (x = "01") then

z <= '0';

next\_state <= S1;

elsif (x = "10") then

z <= '1';

next\_state <= S2;

else

z <= '0';

next\_state <= S3;

end if;

when S3 =>

if (x = "00") then

z <= '0';

next\_state <= S0;

elsif (x = "01") then

z <= '0';

next\_state <= S1;

elsif (x = "10") then

z <= '0';

next\_state <= S2;

else

z <= '1';

next\_state <= S3;

end if;

when others =>

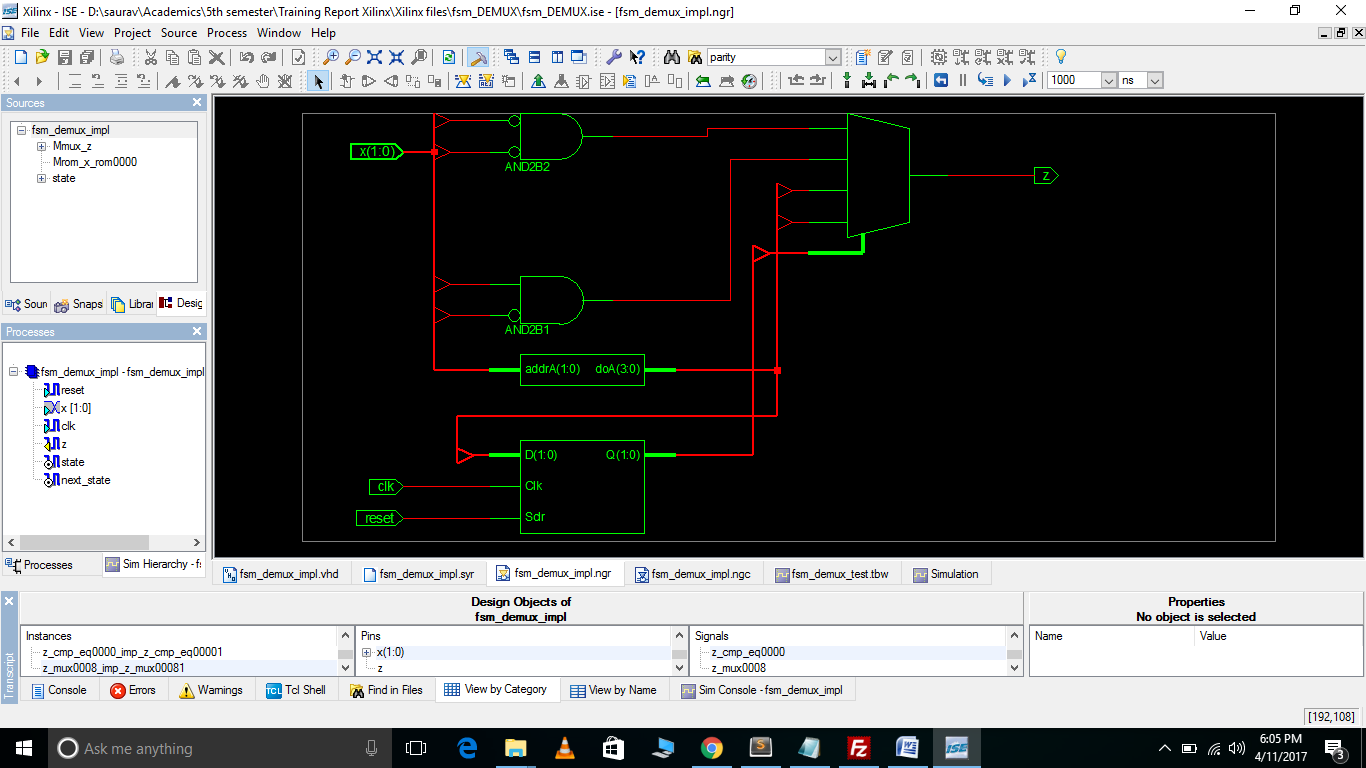
next\_state <= S0;

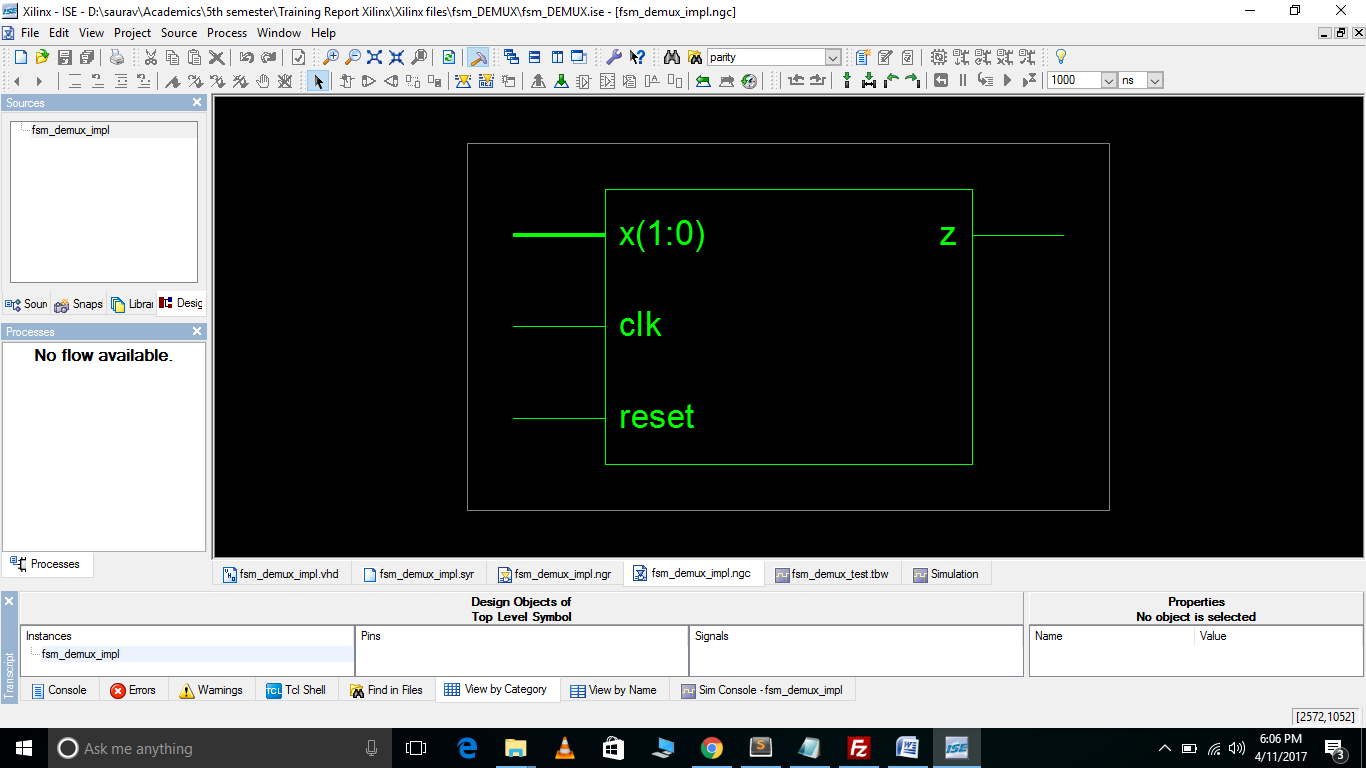
end case;

end process;

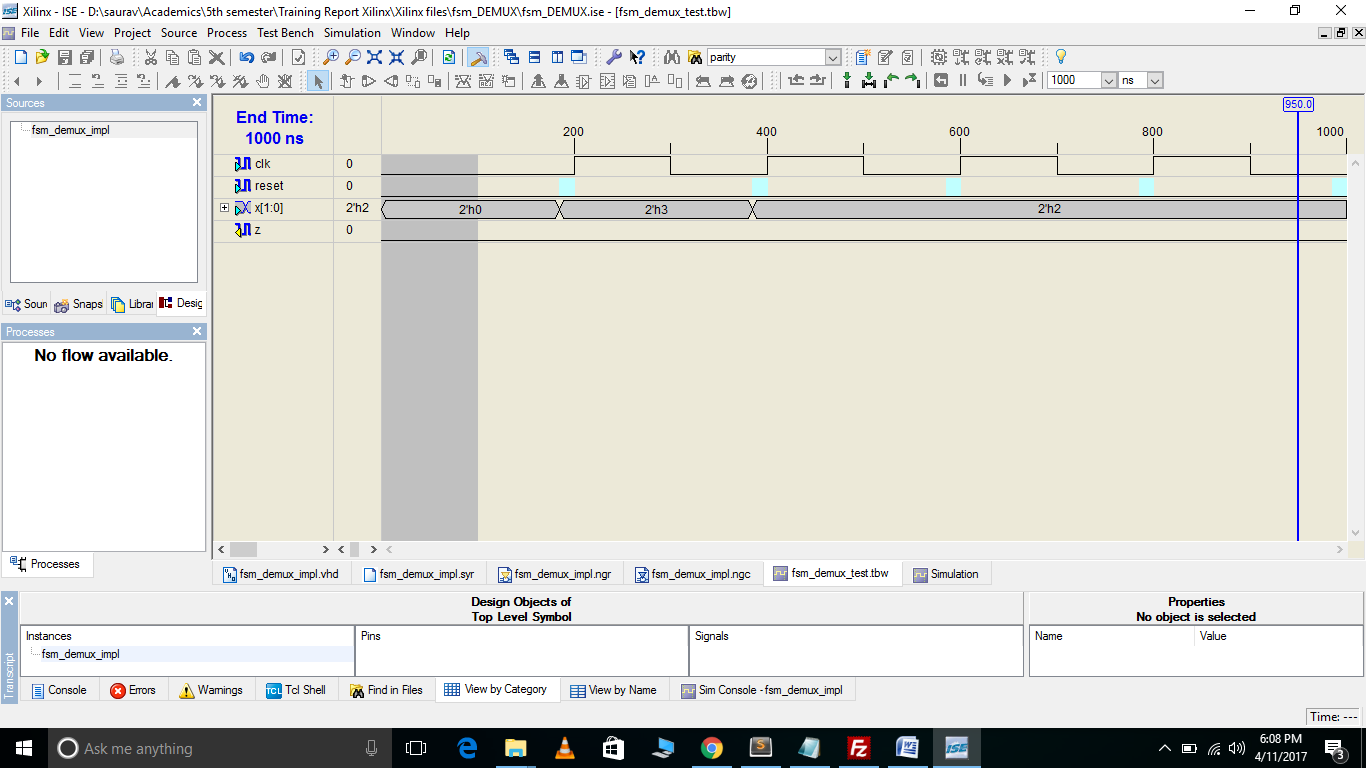
end Behavioral;

View RTL Schematic



View Technology Schematic  
  


Test Bench Waveform



Stimulation  
  
