

library ieee;

use ieee.std\_logic\_1164.all;

entity fsm is

port(

reset, input, clock: in std\_logic;

output: out std\_logic

);

end fsm;

architecture fsm\_logic of fsm is

type state\_available is (A,B,C,D);

signal present\_state, next\_state : state\_available;

begin

process(clock, reset) begin

if(reset='1') then

present\_state <= A;

elsif(rising\_edge(clock)) then

present\_state <= next\_state;

end if;

end process;

process(present\_state, input) begin

case present\_state is

when A =>

if(input='0') then

output <= '1';

next\_state <= C;

else

output <= '0';

next\_state <= B;

end if;

when B =>

if(input='0') then

output <= '0';

next\_state <= D;

else

output <= '1';

next\_state <= B;

end if;

when C =>

if(input='0') then

output <= '1';

next\_state <= D;

else

output <= '1';

next\_state <= C;

end if;

when D =>

if(input='0') then

output <= '0';

next\_state <= D;

else

output <= '1';

next\_state <= A;

end if;

end case;

end process;

end architecture;