Erich Viebrock

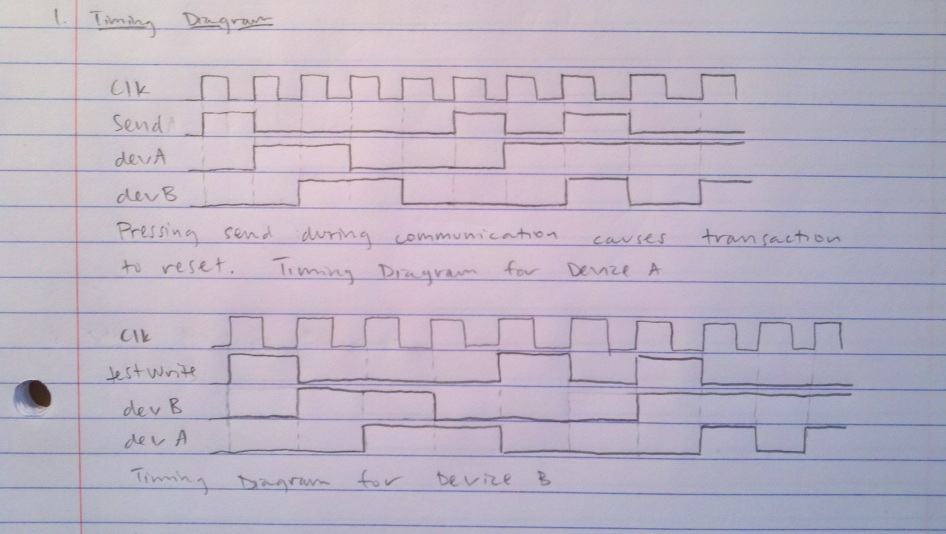
ECPE 174

**Pre-Lab #5**

Purpose:

This lab is intended to familiarize students with machine communication. In this instance, we are getting two FPGA boards to talk to one another. The user initiates the transaction of information, however the FPGA boards carry out the entire transaction.

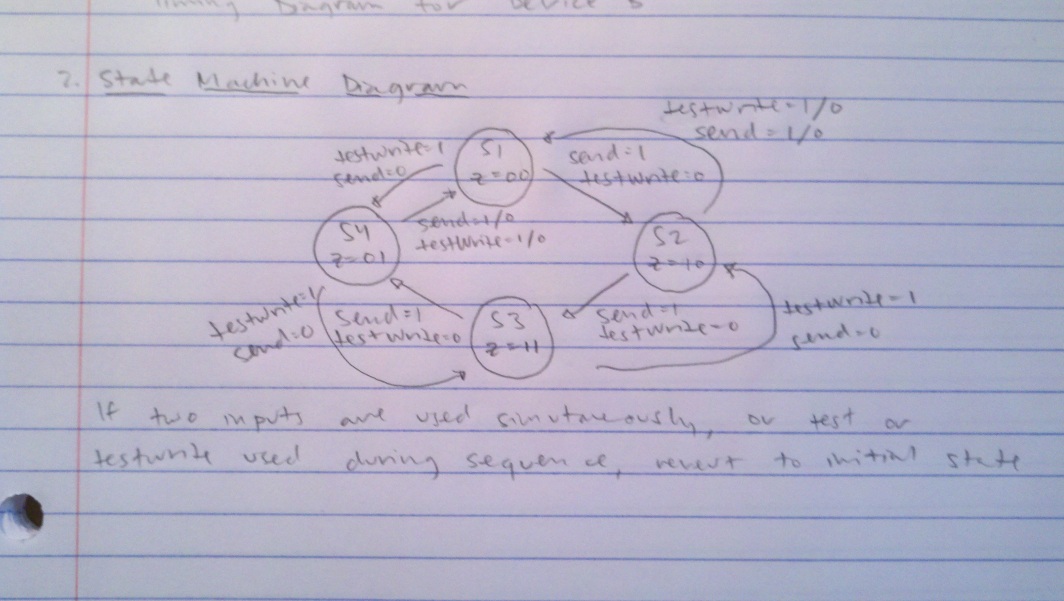
Timing Diagram:



The notes on the bottom of the diagrams say:

Pressing send during communication causes transaction to reset. However, please disregard this, as communication is set up to operate while one key is used constantly. Communication between device A and device B will continue as long as send is HIGH. Communication between device B and device A will continue as long as testWrite is HIGH. If both inputs are HIGH simultaneously, the communication will end, and result back to S1.

State Machine Diagram:



The notes on the bottom of the diagram says:

If two inputs are used simultaneously, or test or testWrite is used during improper sequence, FSM will revert to S1.

VHDL:

library ieee;

use ieee.std\_logic\_1164.all;

ENTITY lab\_5 IS

PORT(clk,send,testWrite : IN STD\_LOGIC;

riseSend,devAView,devBView : OUT STD\_LOGIC);

END lab\_5;

ARCHITECTURE behavior OF lab\_5 IS

SIGNAL rise\_send:std\_logic;

SIGNAL rise\_testWrite:std\_logic;

SIGNAL SEND1:std\_logic;

SIGNAL SEND2:std\_logic;

SIGNAL SEND3:std\_logic;

SIGNAL TESTWRITE1:std\_logic;

SIGNAL TESTWRITE2:std\_logic;

SIGNAL TESTWRITE3:std\_logic;

COMPONENT lab5

PORT(clkin, send, devB : IN STD\_LOGIC;

devA : BUFFER STD\_LOGIC;

riseSend, devAView, devBView : BUFFER STD\_LOGIC);

END COMPONENT;

Type state\_type is (S1, S2, S3, S4);

signal y:state\_type:=S1 ;

BEGIN

PROCESS(clk)

BEGIN

IF(rising\_edge(clk)) THEN

SEND1 <= send;

SEND2 <= SEND1;

SEND3 <= SEND2;

rise\_send <= (NOT SEND3) AND SEND2;

TESTWRITE1 <= testWrite;

TESTWRITE2 <= TESTWRITE1;

TESTWRITE3 <= TESTWRITE2;

rise\_testWrite <= (NOT TESTWRITE3) AND TESTWRITE2;

case y is

when S1=>

if rise\_send='1' then y<=S2;

elsif rise\_testWrite='1' then y<=S4;

else y<=S1;

end if;

when S2=>

if rise\_send='1' then y<=S3;

elsif rise\_send='0' AND rise\_testWrite='0' then y<=S2;

elsif rise\_testWrite='1' then y<=S1;

else y<=S1;

end if;

when S3=>

if rise\_send='1' then y<=S4;

elsif rise\_send='0' AND rise\_testWrite='0' then y<=S3;

elsif rise\_testWrite='1' then y<=S2;

else y<=S1;

end if;

when S4=>

if rise\_send='1' then y<=S1;

elsif rise\_send='0' AND rise\_testWrite='0' then y<=S4;

elsif rise\_testWrite='1' then y<=S3;

else y<=S1;

end if;

end case;

END IF;

END PROCESS;

devAView<='1' when y=S4 AND devBView<='1' when y=S2 else '0';

END behavior;