

**Department of Electrical Engineering**

**Optional 5: Four Digits Passcodes Digital Lock**

Name: Socheath Sok, ID: 014470701

Professor: Dr. Ray Wang

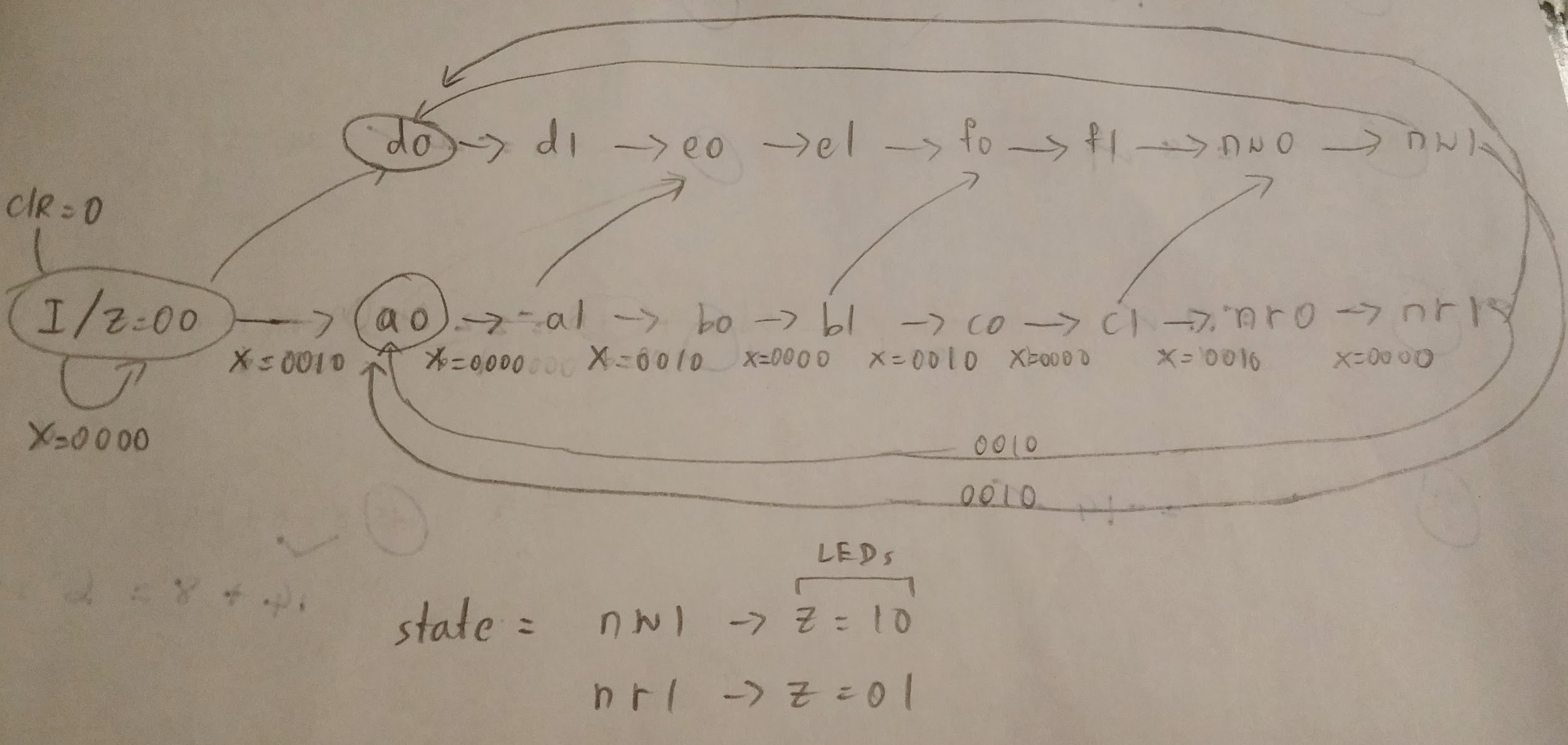
Class: EE 301

Date Due: December 11, 2017

**Explanation**

The goal of this lab is to expand the design from Lab 7 to include another digit of passcode. An new digit will requires four additional states: 2 for the right states and 2 for the wrong state. When all four inputs are correct, output ‘z’ will be “01” and when they are wrong, z will be “10”. The LEDs will not light up for any states in between. The correct passcode for this design is “1111” which can be input using the push buttons on the board. In addition to the lock code, the testbench also required some changes to include the new states. All results for this optional are shown below.

**State Diagram**

****

**Lock Code**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity Lock is

Port ( clk : in STD\_LOGIC;

clr : in STD\_LOGIC;

x : in STD\_LOGIC\_VECTOR (3 downto 0);

z : out STD\_LOGIC\_VECTOR (1 downto 0) := "00");

end Lock;

architecture Behavioral of Lock is

TYPE State\_type is(I,A0,A1,B0,B1,C0,C1,d0,d1,e0,e1,f0,f1,nW0,nW1,nR0,nR1) ;

signal y : state\_type;

begin

process (clk,clr)

begin

if clr='0' then y <= I;

elsif (clk' event and clk = '1') then

--gg: for u in 1 to 3 loop

case y is

when I =>

if x = "0000" then y <= I;

elsif x = "0010" then y <= A0;

else y <= d0;

end if;

when A0 =>

if x = "0000" then y <= A1;

else y <= A0;

end if;

when A1 =>

if x = "0000" then y <= A1;

elsif x = "0010" then y <= B0;

else y <= e0;

end if;

when B0 =>

if x = "0000" then y <= B1;

else y <= B0;

end if;

when B1 =>

if x = "0000" then y <= B1;

elsif x = "0010" then y <= C0;

else y <= f0;

end if;

when C0 =>

if x = "0000" then y <= C1;

else y <= C0;

end if;

when C1 =>

if x = "0000" then y <= C1;

elsif x = "0010" then

y <= nR0;

else

y <= nW0;

end if;

when nR0 =>

if x = "0000" then y <= nR1;

else y <= nR0;

end if;

when nR1 =>

if x = "0000" then y <= nR1;

elsif x = "0010" then y <= A0;

else y <= d0;

end if;

when d0 =>

if x = "0000" then y <= d1;

else y <= d0;

end if;

when d1 =>

if x = "0000" then y <= d1;

else y <= e0;

end if;

when e0 =>

if x = "0000" then y <= e1;

else y <= e0;

end if;

when e1 =>

if x = "0000" then y <= e1;

else y <= f0;

end if;

when f0 =>

if x = "0000" then y <= f1;

else y <= f0;

end if;

when f1 =>

if x = "0000" then y <= f1;

else y <= nW0;

end if;

when nW0 =>

if x = "0000" then y <= nW1;

else y <= nW0;

end if;

when nW1 =>

if x = "0000" then y <= nW1;

elsif x = "0010" then y <= A0;

else y <= d0;

end if;

end case;

end if;

end process;

z <= "01" when y = nR1 else

"10" when y = nW1 else

"00";

end Behavioral;

**Debounce Code**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

use IEEE.STD\_LOGIC\_ARITH.ALL;

use IEEE.STD\_LOGIC\_UNSIGNED.ALL;

entity debounce is

Port ( clock, clr: in STD\_LOGIC;

inp : in std\_logic\_vector(3 downto 0);

outp : out STD\_LOGIC\_vector(3 downto 0) );

end debounce;

architecture Behavioral of debounce is

signal count : STD\_LOGIC\_VECTOR (17 downto 0);

signal clk190 : std\_logic;

signal delay1, delay2, delay3: std\_logic\_vector(3 downto 0);

begin

PROCESS ( clr, clock )

BEGIN

IF clr = '0' THEN

count <= (others => '0') ;

ELSIF Clock'EVENT AND Clock = '1' THEN

count <= count+1 ;

END IF ;

END PROCESS ;

clk190<=count(17);

process (clr, clk190)

begin

if clr='0' then

delay1<="0000";

delay2<="0000";

delay3<="0000";

elsif clk190'event and clk190='1' then

delay1 <=inp;

delay2 <=delay1;

delay3<=delay2;

end if;

end process;

outp<=delay1 and delay2 and delay3;

end Behavioral;

**Structural Code**

library IEEE;

use IEEE.STD\_LOGIC\_1164.ALL;

entity Digital\_Lock is

Port ( btn : in STD\_LOGIC\_VECTOR (3 downto 0);

clk,clr : in STD\_LOGIC;

z : out STD\_LOGIC\_VECTOR (1 downto 0));

end Digital\_Lock;

architecture Behavioral of Digital\_Lock is

component debounce

Port ( clock, clr: in STD\_LOGIC;

inp : in std\_logic\_vector(3 downto 0);

outp : out STD\_LOGIC\_vector(3 downto 0) );

end component;

component Lock

Port ( clk : in STD\_LOGIC;

clr : in STD\_LOGIC;

x : in STD\_LOGIC\_VECTOR (3 downto 0);

z : out STD\_LOGIC\_VECTOR (1 downto 0));

end component;

signal wire : std\_logic\_vector (0 to 3);

begin

Q1: debounce port map(clk,clr,btn,wire);

R1: Lock port map(clk,clr,wire,z);

end Behavioral;

**TestBench**

LIBRARY ieee;

USE ieee.std\_logic\_1164.ALL;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

ENTITY lock\_only\_test IS

END lock\_only\_test;

ARCHITECTURE behavior OF lock\_only\_test IS

COMPONENT Lock

PORT(

clk : IN std\_logic;

clr : IN std\_logic;

x : IN std\_logic\_vector(3 downto 0);

z : OUT std\_logic\_vector(1 downto 0) );

END COMPONENT;

signal clk : std\_logic := '0';

signal clr : std\_logic := '0';

signal x : std\_logic\_vector(3 downto 0) := (others => '0');

signal z : std\_logic\_vector(1 downto 0);

constant clk\_period : time := 20 ns;

BEGIN

-- Instantiate the Unit Under Test (UUT)

uut: Lock PORT MAP (

clk => clk,

clr => clr,

x => x,

z => z );

clk\_process :process

begin

clk <= '0';

wait for clk\_period/2;

clk <= '1';

wait for clk\_period/2;

end process;

stim\_proc: process

begin

-- test clr

clr<='0';

wait for 30ns;

clr<='1';

-- test using correct secret code

x <= "0010";

wait for 20ns;

x <= "0000";

wait for 20ns;

x <= "0001";

wait for 20ns;

x <= "0000";

wait for 20ns;

x <= "0100";

wait for 20ns;

x <= "0000";

wait for 20ns;

x <= "0010";

wait for 20ns;

x <= "0000";

wait for 50ns;

-- test using incorrect secret code

x <= "0010";

wait for 20ns;

x <= "0000";

wait for 20ns;

x <= "0010";

wait for 20ns;

x <= "0000";

wait for 20ns;

x <= "0010";

wait for 20ns;

x <= "0000";

wait for 20ns;

x <= "0010";

wait for 20ns;

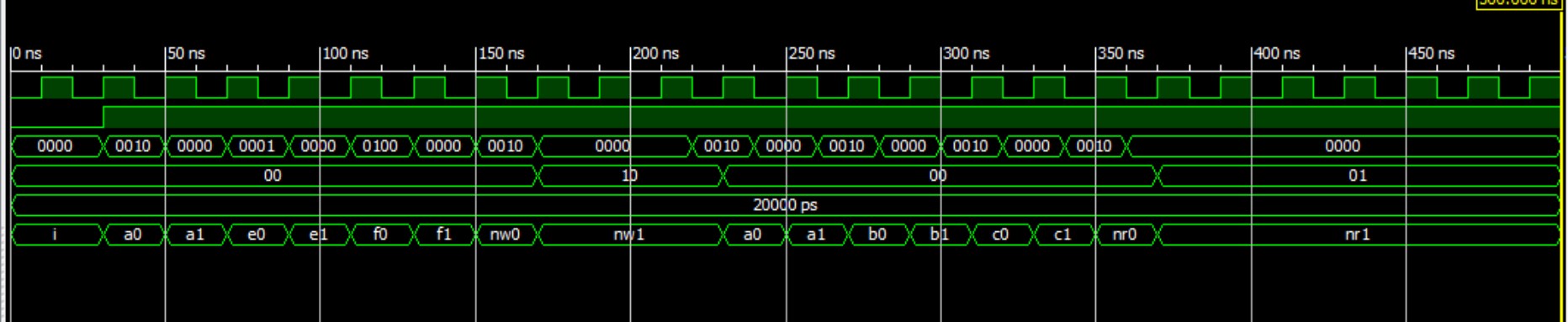
x <= "0000";

wait;

end process;

END;

**Waveform**



**Constraint File**

NET "clk" LOC = "B8" ;

NET "clr" LOC = "P11" ;

NET "btn(0)" LOC = "G12" ;

NET "btn(1)" LOC = "C11" ;

NET "btn(2)" LOC = "M4" ;

NET "btn(3)" LOC = "A7" ;

NET "z(0)" LOC = "M5" ;

NET "z(1)" LOC = "G1" ;