Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 21, 2020
ECE 420

Fall 2020
California State University, Northridge
Department of Electrical and Computer Engineering
Computer Assignment 5:
FSM Design - Digital Lock

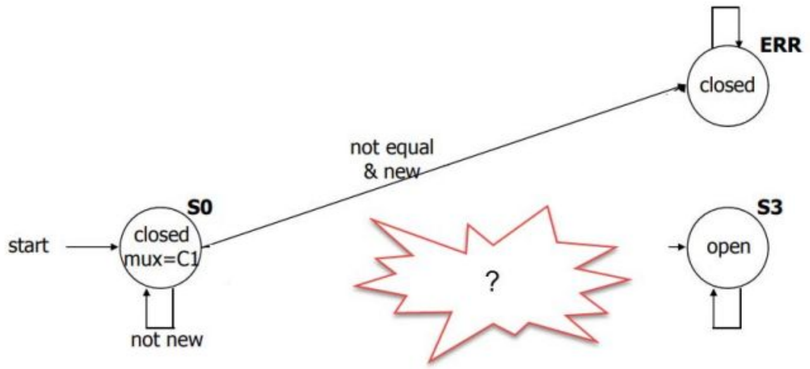
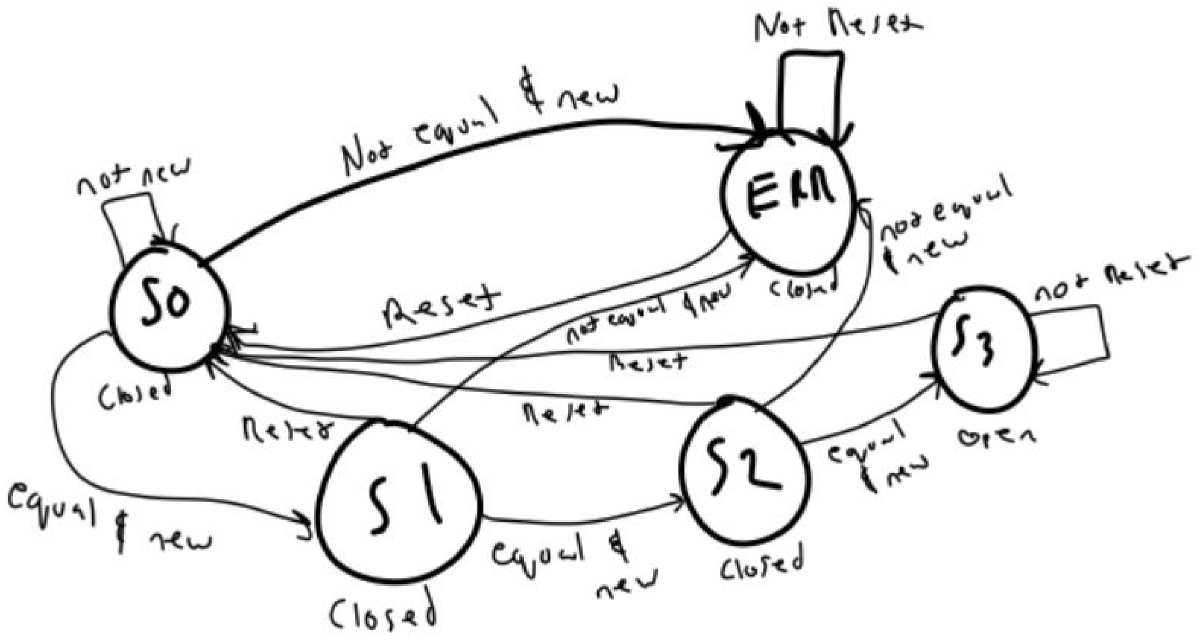


Figure 2: Digital lock controller state diagram

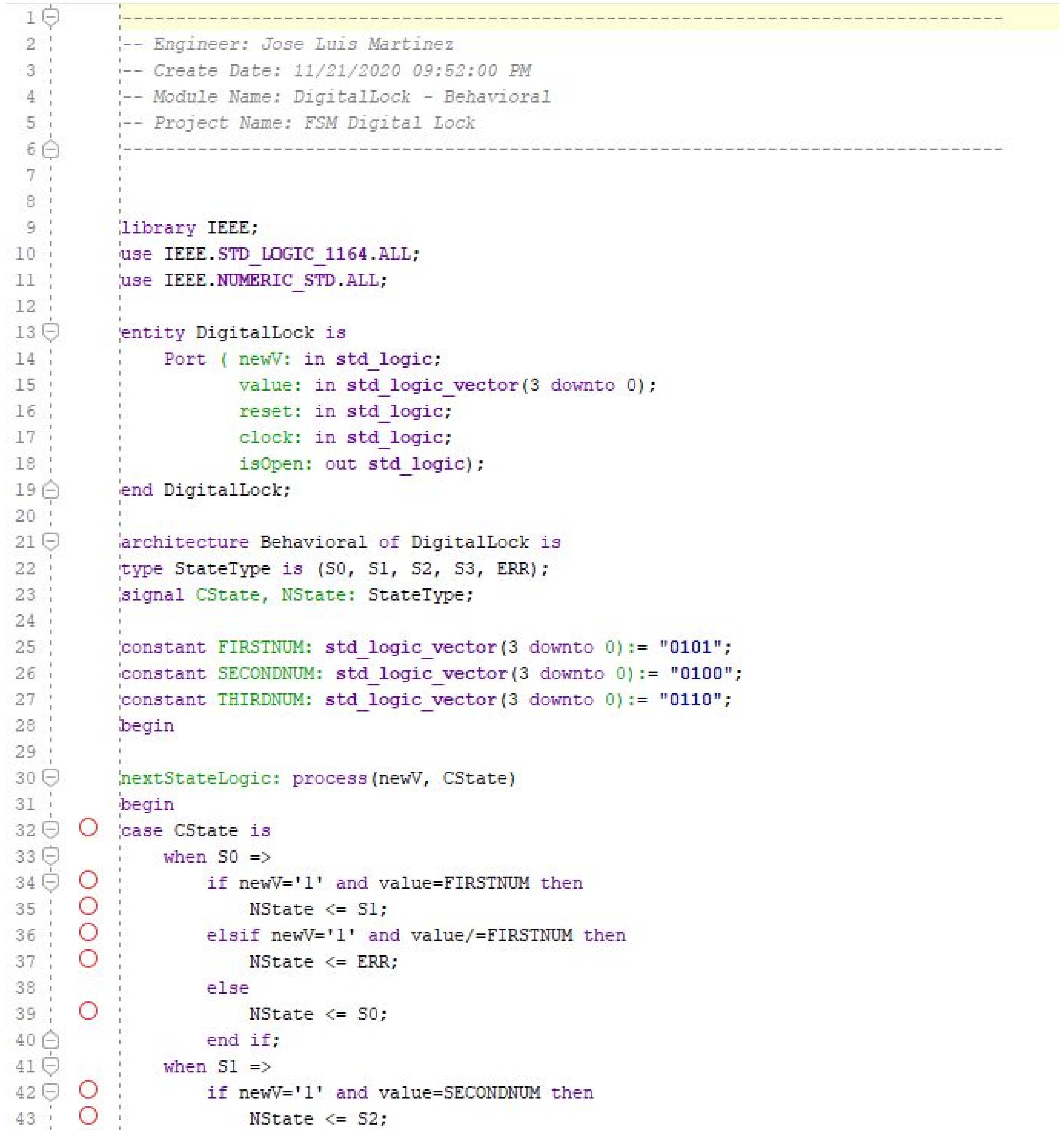
1. Complete the state diagram that is given to you in Figure 2 and specify all the transitions and conditions on your diagram. **(15 points)**



Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 21, 2020
ECE 420

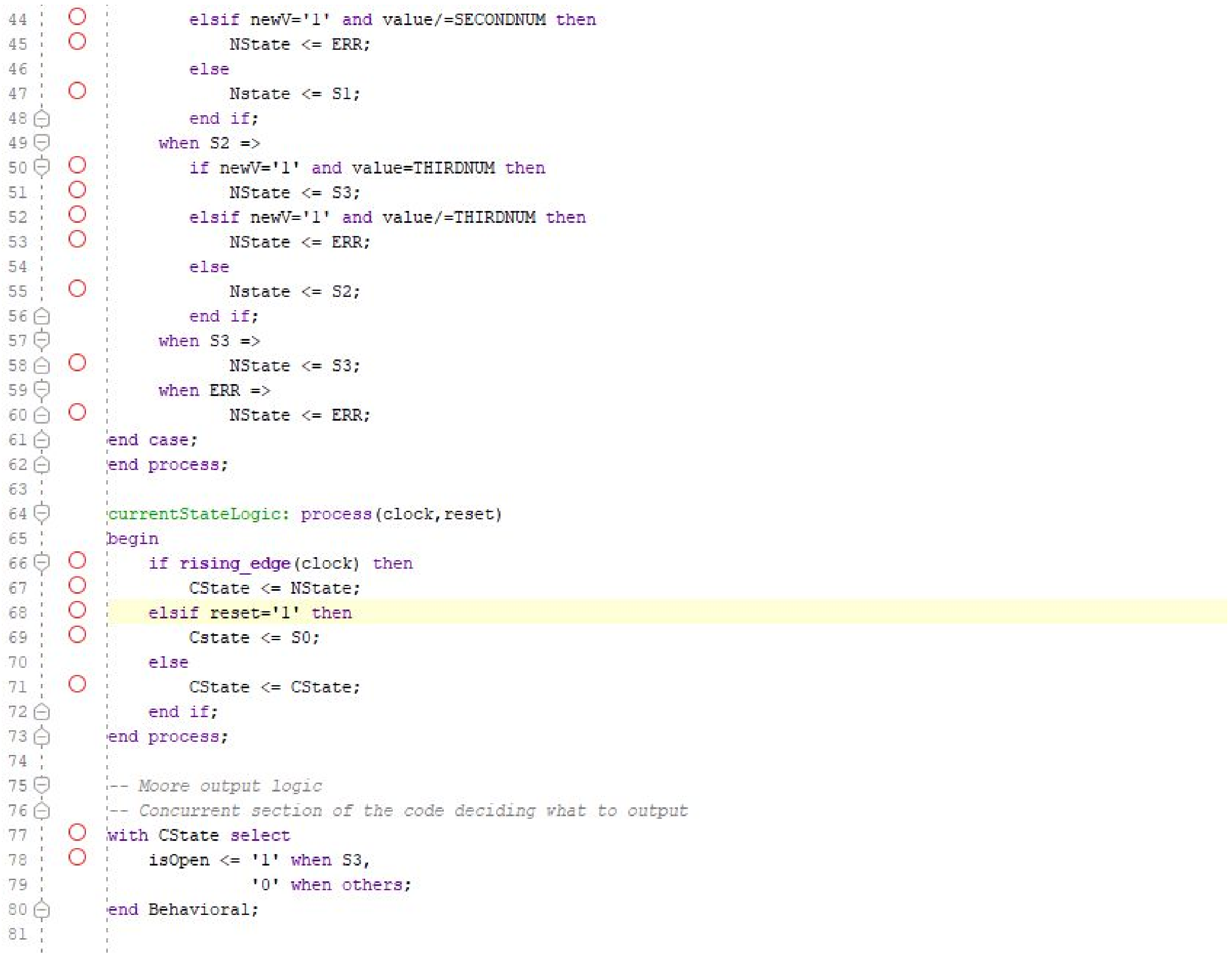
2. Provide the complete source code for your controller design. **(40 points)**

My combination for this design is 546.
Digital Lock FSM Design (VHDL Code)



```vhdl
----------------------------------------------------------------------------------
-- Engineer: Jose Luis Martinez
-- Create Date: 11/21/2020 09:52:00 PM
-- Module Name: DigitalLock - Behavioral
-- Project Name: FSM Digital Lock
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity DigitalLock is
    Port ( newV: in std_logic;
           value: in std_logic_vector(3 downto 0);
           reset: in std_logic;
           clock: in std_logic;
           isOpen: out std_logic);
end DigitalLock;

architecture Behavioral of DigitalLock is
type StateType is (S0, S1, S2, S3, ERR);
signal CState, NState: StateType;

constant FIRSTNUM: std_logic_vector(3 downto 0):= "0101";
constant SECONDNUM: std_logic_vector(3 downto 0):= "0100";
constant THIRDNUM: std_logic_vector(3 downto 0):= "0110";
begin

nextStateLogic: process(newV, CState)
begin
case CState is
    when S0 =>
        if newV='1' and value=FIRSTNUM then
            NState <= S1;
        elsif newV='1' and value/=FIRSTNUM then
            NState <= ERR;
        else
            NState <= S0;
        end if;
    when S1 =>
        if newV='1' and value=SECONDNUM then
            NState <= S2;
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 21, 2020
ECE 420

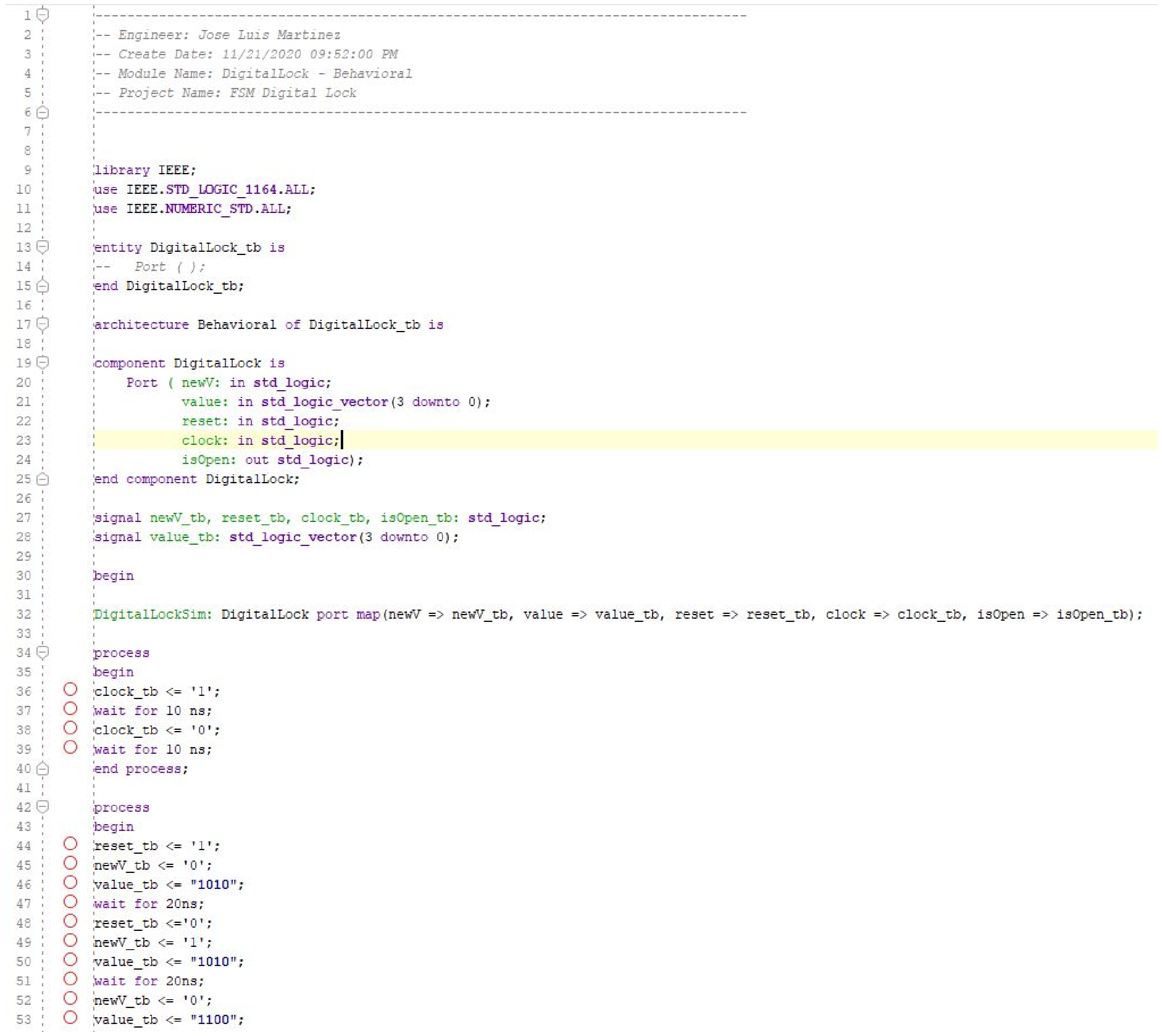


```vhdl
        elsif newV='1' and value/=SECONDNUM then
            NState <= ERR;
        else
            Nstate <= S1;
        end if;
     when S2 =>
        if newV='1' and value=THIRDNUM then
            NState <= S3;
        elsif newV='1' and value/=THIRDNUM then
            NState <= ERR;
        else
            Nstate <= S2;
        end if;
     when S3 =>
            NState <= S3;
     when ERR =>
            NState <= ERR;
end case;
end process;

currentStateLogic: process(clock,reset)
begin
    if rising_edge(clock) then
        CState <= NState;
    elsif reset='1' then
        Cstate <= S0;
    else
        CState <= CState;
    end if;
end process;

-- Moore output logic
-- Concurrent section of the code deciding what to output
with CState select
    isOpen <= '1' when S3,
              '0' when others;
end Behavioral;
```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 21, 2020
ECE 420

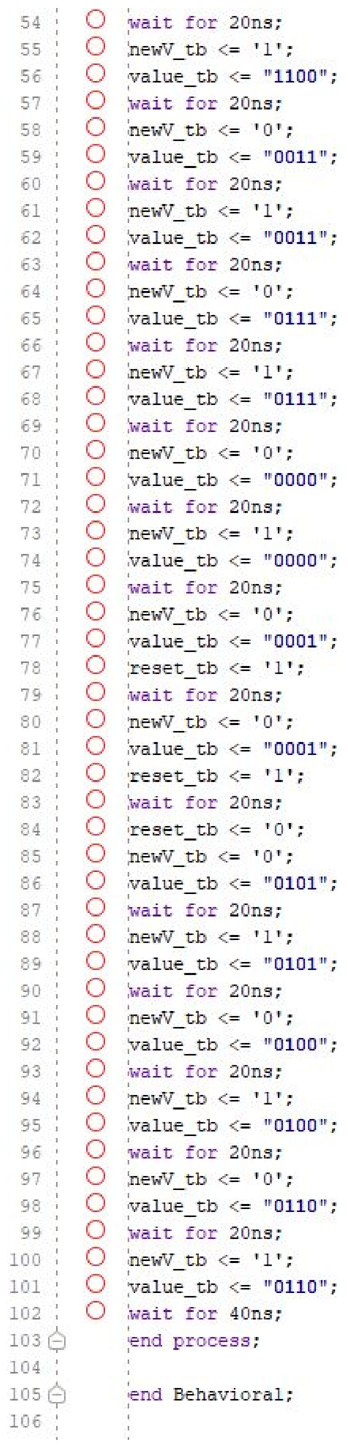
3. Write a VHDL testbench to simulate your design and verify its functionality. Show different combinations on your simulation and critical input combinations to prove your design works. **This is very critical point in your report. (45 points)**

Digital Lock FSM Design (VHDL Testbench Code)



```vhdl
----------------------------------------------------------------------------------
-- Engineer: Jose Luis Martinez
-- Create Date: 11/21/2020 09:52:00 PM
-- Module Name: DigitalLock - Behavioral
-- Project Name: FSM Digital Lock
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity DigitalLock_tb is
--   Port ( );
end DigitalLock_tb;

architecture Behavioral of DigitalLock_tb is

component DigitalLock is
    Port ( newV: in std_logic;
           value: in std_logic_vector(3 downto 0);
           reset: in std_logic;
           clock: in std_logic;
           isOpen: out std_logic);
end component DigitalLock;

signal newV_tb, reset_tb, clock_tb, isOpen_tb: std_logic;
signal value_tb: std_logic_vector(3 downto 0);

begin

DigitalLockSim: DigitalLock port map(newV => newV_tb, value => value_tb, reset => reset_tb, clock => clock_tb, isOpen => isOpen_tb);

process
begin
clock_tb <= '1';
wait for 10 ns;
clock_tb <= '0';
wait for 10 ns;
end process;

process
begin
reset_tb <= '1';
newV_tb <= '0';
value_tb <= "1010";
wait for 20ns;
reset_tb <='0';
newV_tb <= '1';
value_tb <= "1010";
wait for 20ns;
newV_tb <= '0';
value_tb <= "1100";
```

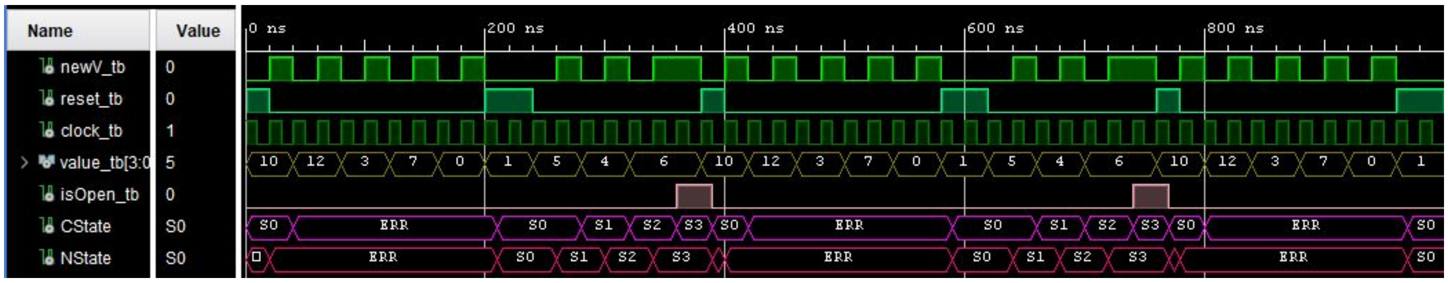
Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 21, 2020
ECE 420

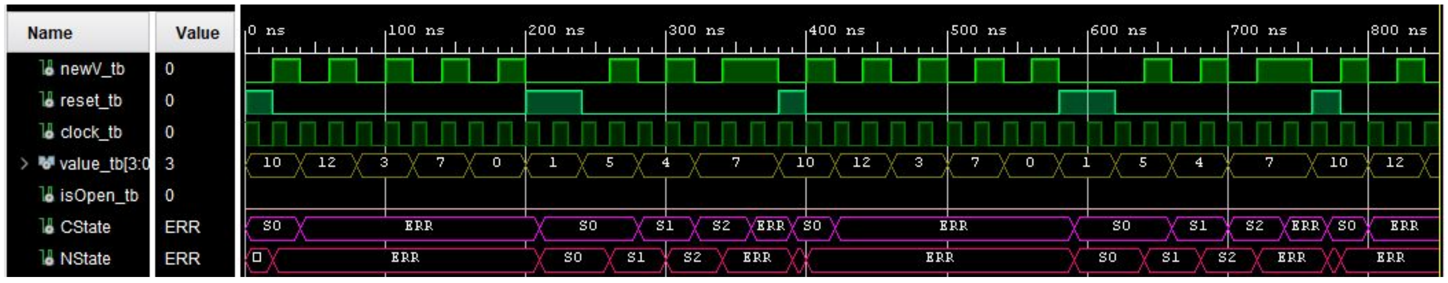


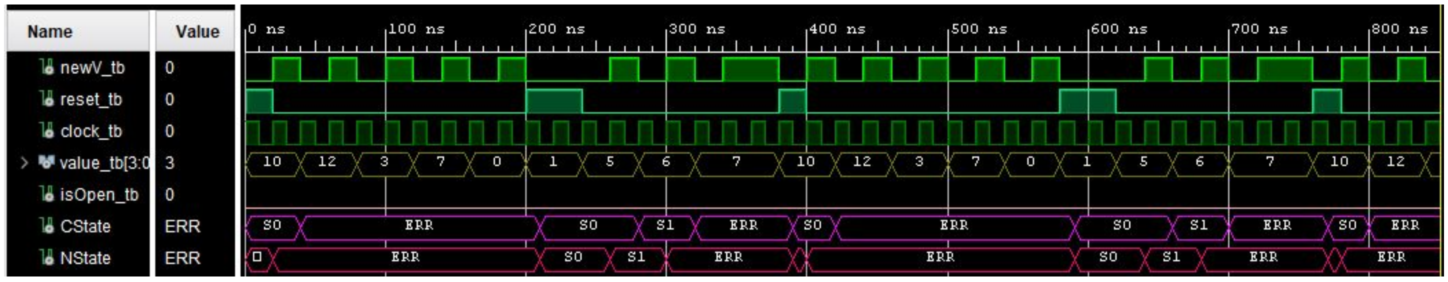
```vhdl
wait for 20ns;
newV_tb <= '1';
value_tb <= "1100";
wait for 20ns;
newV_tb <= '0';
value_tb <= "0011";
wait for 20ns;
newV_tb <= '1';
value_tb <= "0011";
wait for 20ns;
newV_tb <= '0';
value_tb <= "0111";
wait for 20ns;
newV_tb <= '1';
value_tb <= "0111";
wait for 20ns;
newV_tb <= '0';
value_tb <= "0000";
wait for 20ns;
newV_tb <= '1';
value_tb <= "0000";
wait for 20ns;
newV_tb <= '0';
value_tb <= "0001";
reset_tb <= '1';
wait for 20ns;
newV_tb <= '0';
value_tb <= "0001";
reset_tb <= '1';
wait for 20ns;
reset_tb <= '0';
newV_tb <= '0';
value_tb <= "0101";
wait for 20ns;
newV_tb <= '1';
value_tb <= "0101";
wait for 20ns;
newV_tb <= '0';
value_tb <= "0100";
wait for 20ns;
newV_tb <= '1';
value_tb <= "0100";
wait for 20ns;
newV_tb <= '0';
value_tb <= "0110";
wait for 20ns;
newV_tb <= '1';
value_tb <= "0110";
wait for 40ns;
end process;

end Behavioral;

```

Jose Luis Martinez
Prof: Shahnam Mirzaei, Ph.D.
November 21, 2020
ECE 420

Digital Lock FSM Design (Waveforms)


Name | Value
newV_tb 0
reset_tb 0
clock_tb 1
value_tb[3:0] 5
isOpen_tb 0
CState S0
NState S0
0 ns
200 ns
400 ns
600 ns
800 ns
value: 10 12 3 7 0 1 5 4 6 10 12 3 7 0 1 5 4 6 10 12 3 7 0 1
CState: S0 ERR S0 S1 S2 S3 S0 ERR S0 S1 S2 S3 S0 ERR S0
NState: 0 ERR S0 S1 S2 S3 ERR S0 S1 S2 S3 ERR S0


Name | Value
newV_tb 0
reset_tb 0
clock_tb 0
value_tb[3:0] 3
isOpen_tb 0
CState ERR
NState ERR
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
value: 10 12 3 7 0 1 5 4 7 10 12 3 7 0 1 5 4 7 10 12
CState: S0 ERR S0 S1 S2 ERR S0 ERR S0 S1 S2 ERR S0 ERR
NState: 0 ERR S0 S1 S2 ERR ERR S0 S1 S2 ERR ERR


Name | Value
newV_tb 0
reset_tb 0
clock_tb 0
value_tb[3:0] 3
isOpen_tb 0
CState ERR
NState ERR
0 ns
100 ns
200 ns
300 ns
400 ns
500 ns
600 ns
700 ns
800 ns
value: 10 12 3 7 0 1 5 6 7 10 12 3 7 0 1 5 6 7 10 12
CState: S0 ERR S0 S1 ERR S0 ERR S0 S1 ERR S0 ERR
NState: 0 ERR S0 S1 ERR ERR S0 S1 ERR ERR