Texas A&M University

Simple Digital Combination Lock

*Author:*

Eden Habtegebrial

November 26, 2016



**Objective**

This lab’s main purpose is to implement and support the ideas covered in lecture by creating a simple combinational lock in digital form on the Spartan 3E board. To accomplish this we used a state diagram where we conveniently represented the combinational lock. The Spartan 3E board has an LCD light display along with a rotating knob that we utilized in the implementation.

**Design**

*Combination Lock*

`timescale 1ns / 1ps

`default\_nettype none

/\*This module describes the combination-lock FSM discussed in the pre-lab

using behavioral Verilog \*/

module combination\_lock\_fsm(

output reg[2:0] state,

output wire Locked, //asserted when locked

input wire Right, Left, //indicate direction

input wire [4:0] Count, // indicate position

input wire Center, //the unlock button

input wire Clk, South //clock and reset

);

//intermediate nets

parameter S0 = 3'b000,

S1 = 3'b001,

S2 = 3'b010,

S3 = 3'b011,

S4 = 3'b100,

S5 = 3'b101;

reg [2:0] nextState;

always@(\*)

case(state)

S0: begin

if(Right)

nextState = S1;

else

nextState = S0;

end

S1: begin

if(Left)

if(Count == 5'b01101)

nextState = S2;

else

nextState = S0;

else

nextState = S1;

end

S2: begin

if(Right)

if(Count == 5'b00111)

nextState = S3;

else

nextState = S0;

else

nextState = S2;

end

S3: begin

if(Left)

if(Count == 5'b10001)

nextState = S4;

else

nextState = S0;

else

nextState = S3;

end

S4: begin

if (Center)

if(Count == 5'b00001)

nextState = S5;

else

nextState = S0;

else

nextState = S4;

end

S5: begin

nextState = S5;

end

endcase

// The following describes the synchronous logic to hold our state

always@(posedge Clk)

if(South) //reset state

state <=S0;

else

state <= nextState;

//describes the output logic

assign Locked = ~(state == S5);

endmodule

*Up-Down Counter*

`timescale 1ns / 1ps

`default\_nettype none

/\*This behavioral Verilog description models

an up down counter that counts between 0-19 \*/

module up\_down\_counter(

//output will be driven in an always block

output reg[4:0] Count,

input wire Up, Down,

input wire Clk, South

);

//Positive edge triggered synchronous logic with a synchronous reset

always@(posedge Clk)

if(South)

Count<=0;

else if(Up)

begin

if(Count == 19) //if at top end

Count <= 0; //roll over

else //count up

Count <= Count+1;

end

else if(Down)

begin

if(Count == 0) //if at bottom end

Count <= 19; //roll over

else //count down

Count<= Count-1;

end

endmodule

*Rotary Combination Lock*

/\*This is the top-level module for our digital \*

\*rotary combination-lock based on the diagram \*

\*provide in the lab manual                    \*/

module rotary\_combination\_lock(

   /\*LCD interface wires make up our output!\*/

   output wire LCD\_E, LCD\_RW, LCD\_RS,

   output wire [3:0] SF\_D,

   /\*Let's output state for debugging!\*/

   output wire [2:0] J1,

   input Clk,

   /\*the buttons and rotary encoder outputs\*

    \*provide input to our top-level circuit\*/

   input Center,

   input South,

   input wire rotA, rotB

);

   /\*intermediate nets\*/

   wire CenterSync, SouthSync;

   wire Right, Left;

   wire Locked;

   wire [4:0] Count;

   /\*synchronize button inputs\*/

   synchronizer syncA(CenterSync, Center, Clk);

   synchronizer syncB(SouthSync, South, Clk);

   /\*wire up rotary encoder module\*/

   rotary\_encoder\_module U0(

        .Left(Left),

        .Right(Right),

        .Clk(Clk),

        .rotA(rotA),

        .rotB(rotB)

   );

   /\*wire up combination lock FSM\*/

   combination\_lock\_fsm U1(

        .Locked(Locked),

        .Right(Right),

        .state(J1),

        .Left(Left),

        .Center(CenterSync),

        .Clk(Clk),

        .South(SouthSync),

        .Count(Count)

   );

   /\*instantiate up down counter\*/

   up\_down\_counter U2(

       .Count(Count),

       .Up (Left),

       .Down(Right),

       .Clk(Clk),

       .South(South)

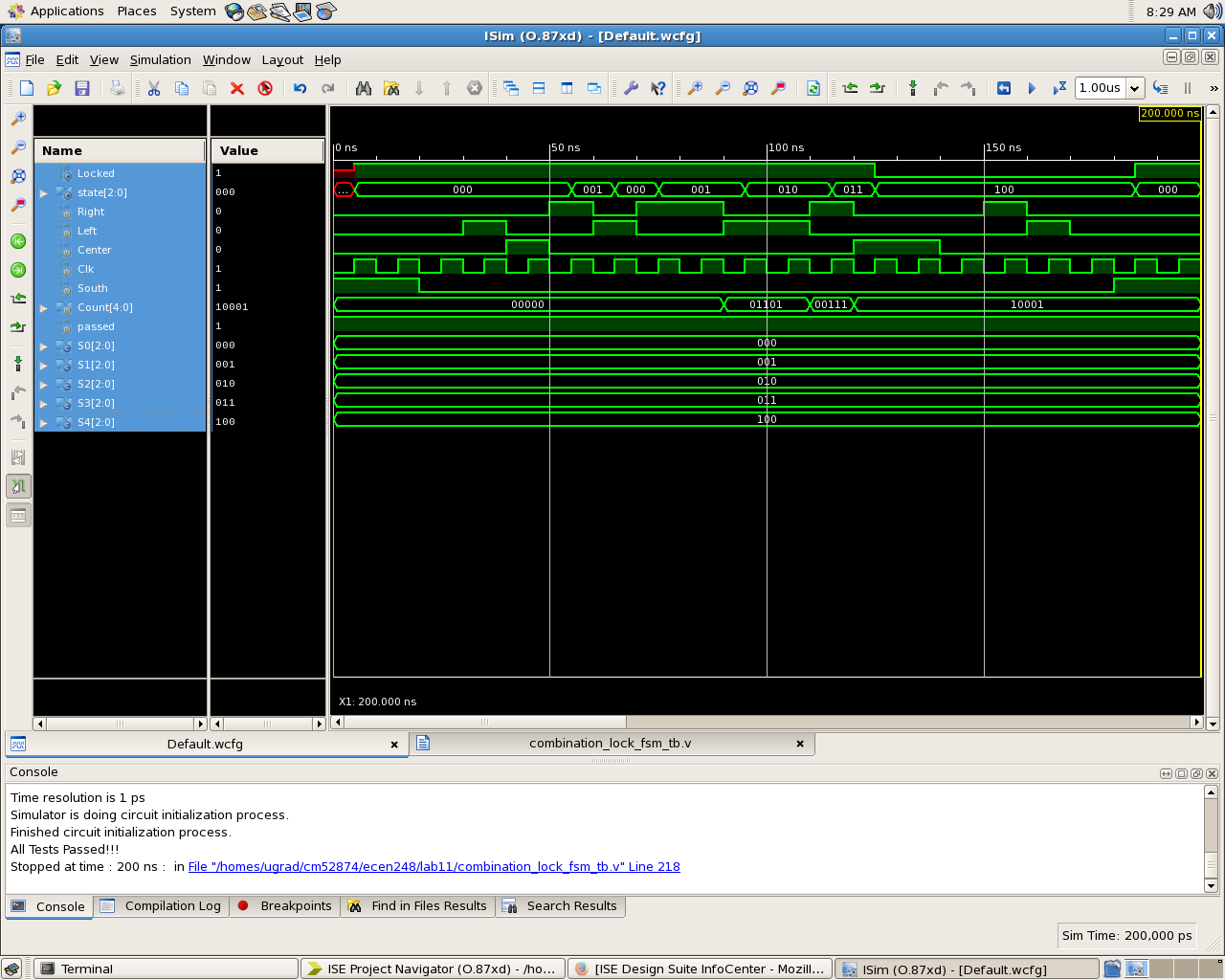
   );

   /\*hook up LCD driver\*/

   lcd\_driver U3(Clk, South, Count, Locked, SF\_D, LCD\_E, LCD\_RS, LCD\_RW);

endmodule

Combination Lock



Up-Down Counter

