Pre-Lab 10: A simple digital combinational lock

ECEN 248 - 505

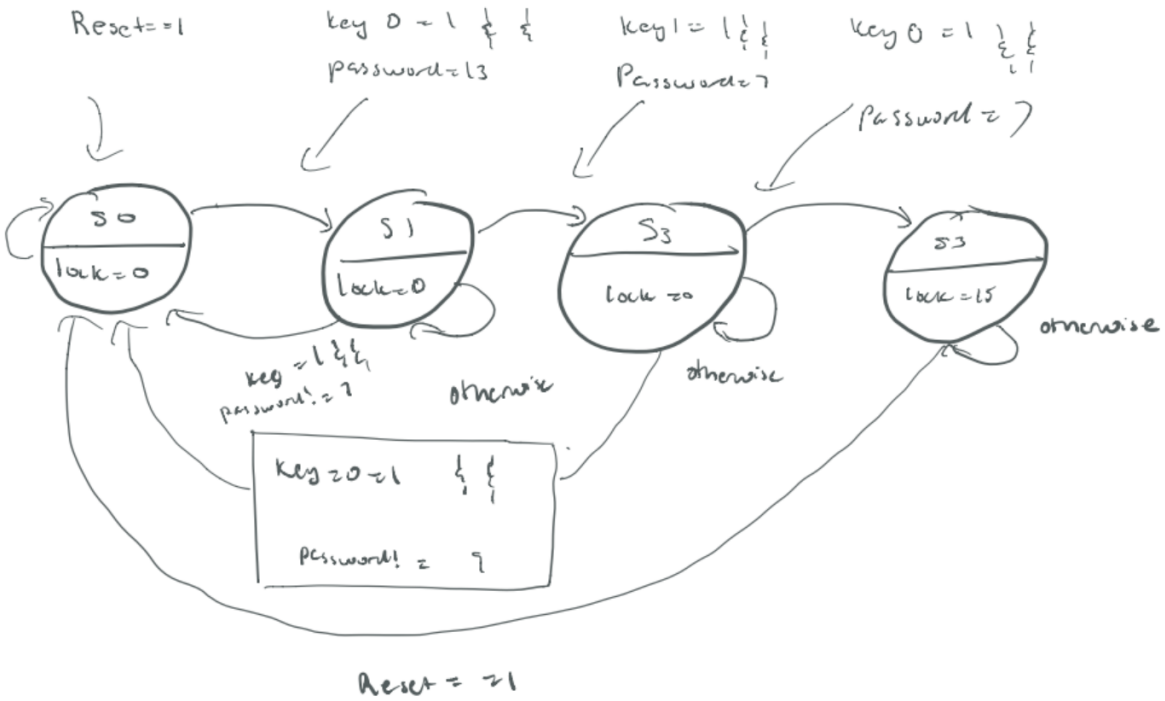
TA: Younggyun Cho

Date: November 3, 2020

1. Binary numbers for 13,7 and 9.

13 = 1101
7 = 0111
9 = 1001

2. The completed state diagram for the combination-lock FSM.



3. The combination-lock FSM Verilog module.

```
`timescale 1ns/ 1ps
module combination_lock_fsm(
output reg [1:0] state,
output wire [3:0] Lock,
input wire Key0,
input wire Key1,
input wire [3:0] Password,
input wire Reset,
input wire Clk
);

parameter S0 = 2'b00,
```

```
S1 = 2'b01,
S2 = 2'b10,
S3 = 2'b11;
reg [1:0] nextState;
reg [3:0] Lockstate;
always@(*)

case(state)
S0: begin
Lockstate = 4'b0001;
if(Key1 == 1 && Password == 4'b1101)
nextState = S1;
else
nextState = S0;
end
S1: begin
Lockstate = 4'b0010;
if(Key2 == 1 && Password == 4'b0111)
nextState = S2;
else if(Key2 == 1 && Password != 4'b0111)
nextState = S0;
else
nextState = S1;
end

S2: begin
Lockstate = 4'b0011;
if(Key1 == 1 && Password == 4'b1001)

nextState = S3;
else if(Key1 == 1 && Password !== 4'b1001)
nextState = S0;
else
nextState = S2;
end
S3: begin
Lockstate = 4'b1111;
nextState = S3;
```

```
end

endcase

always@ (posedge Clk)
if(Reset)
state <= S0;
else
state <= nextState;
assign Lock=Lockstate;

endmodule
```