EL-203 : Embedded Hardware Design

Project Report

Group: 2
Topic: Push button Door Lock
Question : 1
Assigned By: Prof. Biswajit Mishra

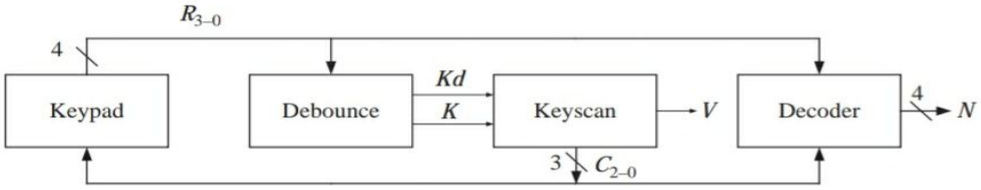
Student_ID	Name	Contribution
201701076	Vedant Parikh	VHDL Code and Report
201701077	Dhruvi Shah	ASM charts and Report
201701078	Samyak Bakliwal	State Diagram
201701082	Bhavna Kohli	Absent

Keypad Implementation

In design of push-button door lock , there are mainly three modes:
(1) Enter the password to open the door here the right sequence is to be followed by # and as long as # is being pressed the door remains open.
(2)Reset the password for this enter the right sequence followed by * and then enter new password two times with #.
(3)A RESET button which is not part of keypad and reset with it.

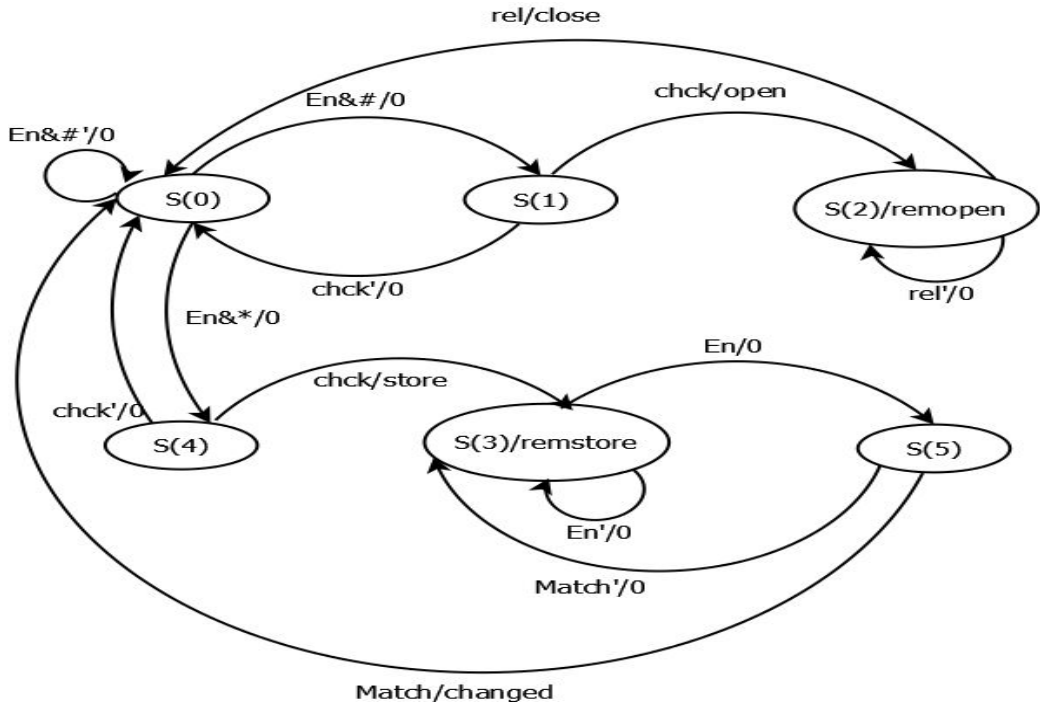
A Keypad Scanner:

Consisting of three parts : Debouncer, KEY Scanner, Decoder.



Keypad's Dial consists of 0 to 9 digits and * and # sign. Keypad take 4 inputs from the user it send it To Debounce state. Debounce State will Delay the input by Certain Clock pulse cycle(**Kd**).Then it will send the signals K and Kd to the Keyscan which decides which row and Column is pressed.
Then Decoder will produce 4 outputs. They are.

Logic Equations for Decoder:
N3 = R2C0'+R3C1'
N2 = R1 + R2C0
N1 = R0C0'+R2'C2 + R1'R0'C0
N0 = R1C1 + R1'C2 + R3'R1'C1'



Inputs are as followed:

Rel:1 if # is released,

En&# :1 if Entered the password and pressed #, **En&*** :1 if Entered the password and pressed*,

Chck: 1 if same password the entered,En: 1 if Entered the new password followed by #,

Match:1 if Entered the password 2nd time followed by # & matched with first one.

Outputs are as followed:

Open : the lock is now opening , **remopen** : the lock is remaining opened now,

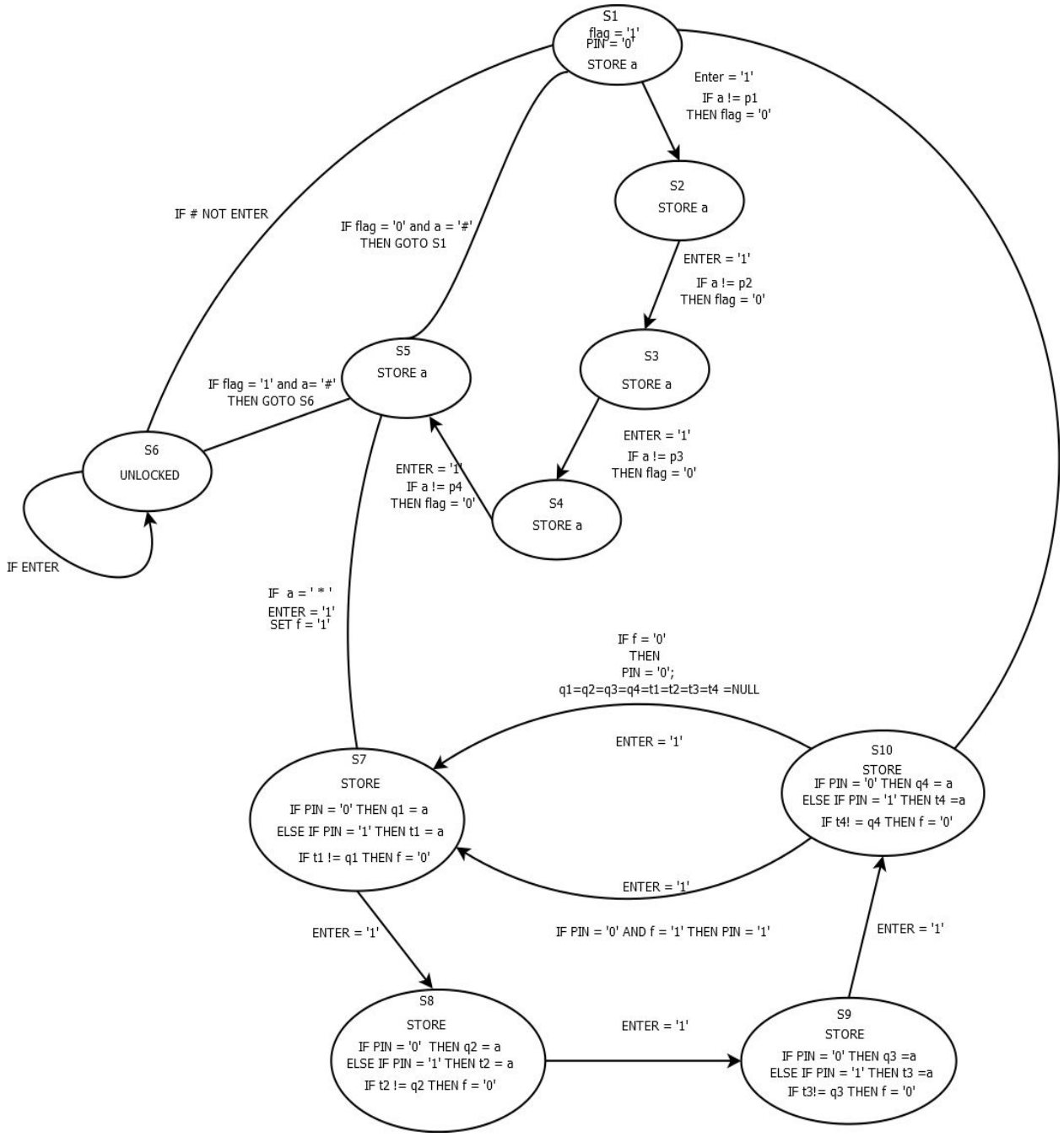
Close:1 closes the lock , **store**: 1 enables the store light on,

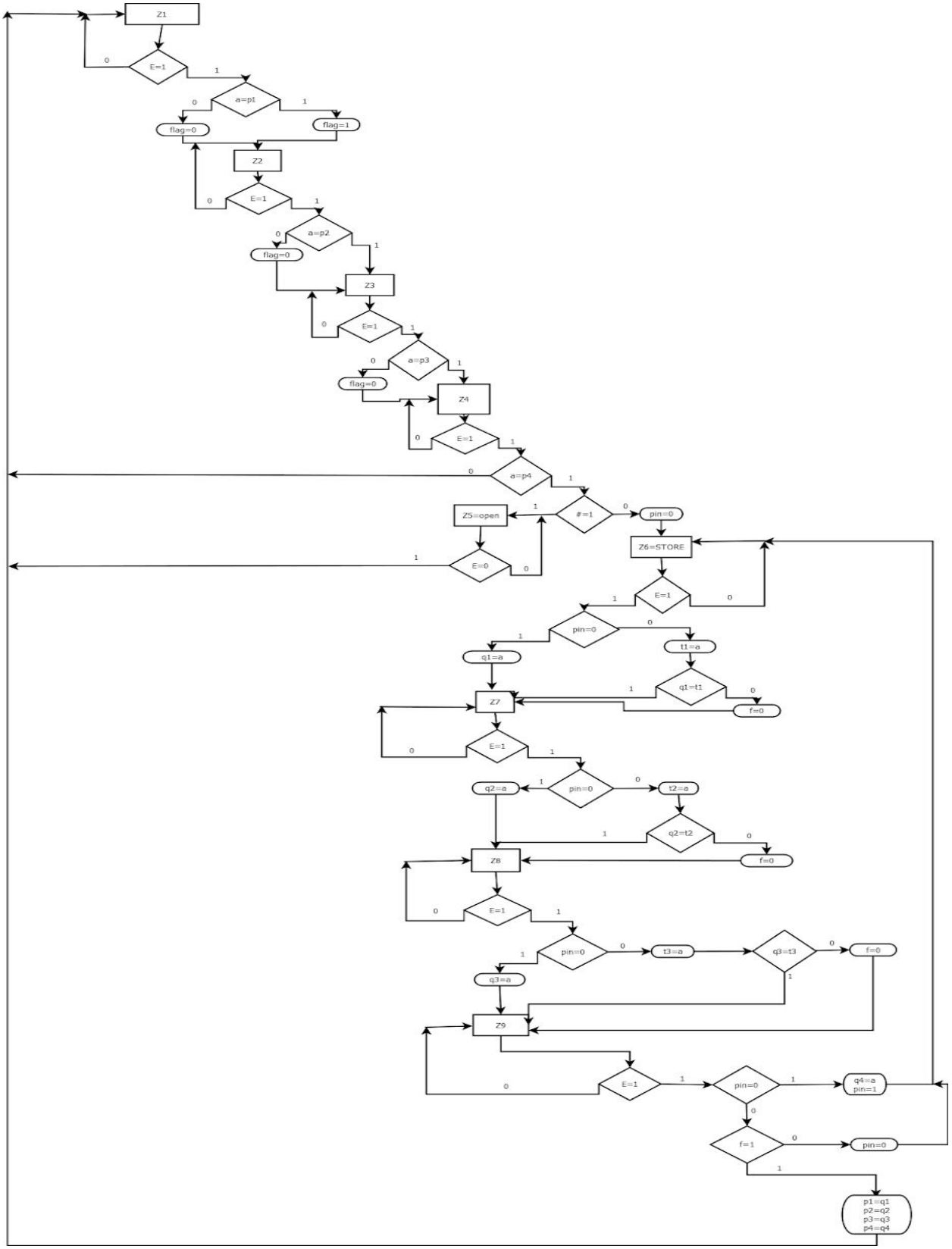
remstore :1 enables lock remaining in stored mode,

changed: 1 enables the new password of lock.

The logic for checking the password:

STATE Diagram




Z1
E=1
a=p1
flag=0
flag=1
Z2
E=1
a=p2
flag=0
Z3
E=1
a=p3
flag=0
Z4
E=1
a=p4
#=1
Z5=open
E=0
pin=0
Z6=STORE
E=1
pin=0
q1=a
t1=a
q1=t1
f=0
Z7
E=1
pin=0
q2=a
t2=a
q2=t2
f=0
Z8
E=1
pin=0
t3=a
q3=t3
f=0
q3=a
Z9
E=1
pin=0
q4=a
pin=1
f=1
pin=0
p1=q1
p2=q2
p3=q3
p4=q4

Code Snippets

```
TYPE state_type IS (locked,z1,z2,z3,z4,z6,z7,z8,z9,z10,unlck);
 SIGNAL state : state_type := locked;
        signal flag: STD_LOGIC := '1';
        signal p: STD_LOGIC := '0';
        signal f: STD_LOGIC := '1';
        signal p1: STD_LOGIC_VECTOR (3 downto 0):="0001";
    signal p2: STD_LOGIC_VECTOR (3 downto 0):="0010";
    signal p3: STD_LOGIC_VECTOR (3 downto 0):="0011";
        signal p4: STD_LOGIC_VECTOR (3 downto 0):="1010";
        signal t1: STD_LOGIC_VECTOR (3 downto 0);
    signal t2: STD_LOGIC_VECTOR (3 downto 0);
    signal t3: STD_LOGIC_VECTOR (3 downto 0);
        signal t4: STD_LOGIC_VECTOR (3 downto 0);
        signal q1: STD_LOGIC_VECTOR (3 downto 0);
    signal q2: STD_LOGIC_VECTOR (3 downto 0);
    signal q3: STD_LOGIC_VECTOR (3 downto 0);
        signal q4: STD_LOGIC_VECTOR (3 downto 0);
```

We have intialized locked,unlck and other z1 to z10 states. Flag,p,f are used as the flags in the code.
P1,p2,p3,p4 are the variables which stores the password given by the user.
T1,t2,t3,t4,q1,q2,q3,q4 are the temporary variables which are in two loops for storing.

```
when locked =>
unlocked <='0';
  flag <= '1';
  store <='0';
  f<= '1';
  if(a/=p1) then
     flag <= '0';
     state <=z1;
  else
     state <= z1;
  end if;
```

```
when z1 =>
  flag <= '1';

  if(a/=p2) then
     flag <= '0';
     state <=z2;
  else
     state <= z2;
  end if;
```

```
when z2 =>
  flag <= '1';

  if(a/=p3) then
     flag <= '0';
     state <=z3;
  else
     state <= z3;
  end if;
```

```
when z3 =>
  flag <= '1';

  if(a/=p4) then
     flag <= '0';
     state <=z4;
  else
     state <= z4;
  end if;
```

The states locked, Z1, Z2, Z3 are used for checking whether input given by the user **‘a’** is equal to password stored in the system. If the password entered is same, then ‘**flag’** would be equal 1 at the end of Z3 state.

```
when z4=>

  if(a="1011") then
     if flag <= '0' then
               state <= locked;
     else
               unlocked <= '1';
          state <= unlock;
           end if;
  elsif(a="1010") then
             if flag <='0' then
              state <= locked;
             else
              state <= z6;
             end if;
  end if;
```

```
when unlck =>
if (a/="1011") then
   unlocked <='0';
   state <= locked;

else
   state <= unlck;
end if;
```

Then we enter '#' ,i.e, 1011 or '*' ,i.e, 1010. If the flag is '0' , that is entered password is wrong therefore it goes to 'locked' state. Else if we enter '#' and password is correct the system enters unlck state. IF the user enters '*', system moves to z6 state which is store state.

```
when z6 =>
       store <='1';
       f <= '1';
       if(p = '0') then
       q1 <= a;
       state <= z7;
       else
       t1 <=a;
       if(q1/=t1) then
              f <='0';
              state <= z7;
       else
              state <= z7;
       end if;
end if;
```

```
when z7 =>
if(p = '0') then
       q2 <= a;
       state <= z8;
else
   t2 <=a;
       if(q2/=t2) then
       f <='0';
       state <= z8;
       else
       state <= z8;
       end if;
end if;
```

```
when z8 =>
       if(p = '0') then
       q3 <= a;
       state <= z9;
else
       t3 <=a;
       if(q3/=t3) then
       f <= '0';
       state <= z9;
       else
       state <= z9;
       end if;
end if;
```

```
when z9 =>
       if(p = '0') then
       q4 <= a;
       state <= z10;
       else
       t4 <=a;
       if(q4/=t4) then
       f <='0';
       state <= z10;
       else
       state <= z10;
       end if;
end if;
```

The States Z6,Z7,Z8,Z9 states are for storing the password entered by user. Here **'f' flag** used for checking Wheather password entered the second time is same as the first. And 'p' flag for Calculating the nos. Of Nos. of time these states have executed.Z10 stores the password to p1,p2,p3,p4 variables .

Problem Faced

- Keypad Implementation
  We tried to code. It worked in the Xilinx code, but the integration of it to the Door Lock code was not correctly working.
- Seven Segment Display
  We thought we would print Asterisk(**'*'**) sign on Display. But it was left, we try do it.
- Length of Password is considered constant.

References

- https://www.youtube.com/watch?v=b-DL3LiJrOk
- Digital System Design using VHDL by Charles H Roth and Jr. Lizy Kurian.
- https://www.fpga4student.com/