EXPERIMENT – 10
PROJECT: 4-DIGIT PASSWORD PROTECTED LOCK SYSTEM

Introduction:

Lock is very common device to protect so many things like main doors, lockers, suitcase etc. but there is a trouble to carry key with us in case of conventional locks. Digital password protected systems can be best solution of it. Here we have demonstrated an implementation of sequential circuit of this password protected lock system. Features of the system is as given below.

- ✓ 4-digit length of password, each digit could have 15 possibilities (0X01 to 0Xff).
- ✓ Password could be change any time.
- ✓ After three attempts of continuous wrong password, alarm would be activate automatically.
- ✓ Alarm can only be turn off by entering right password.

While implementing hardware, there will be a 15 digit number panel to enter password. Moreover, using decoder every digit will be addressed by ex-1 method of 4-bit binary code at output line, so when no any button is pressed all 4 lines will become 0 and respective ex-1 code will be generated when button is pressed. Here we have generated clock by taking advantage of this pattern to synchronize our circuitry.
As we are going to demonstrate our circuit only on software, manner of entering password can be a little bit different. For entering password, we have to follow the sequence given in table 10.1 in software demonstration only.

Random digit (for initialization)	1st Digit	0	2nd Digit	0	3rd Digit	0	4th Digit	0

Table 10.1

Block diagram of functioning of digital password protected lock system is shown in fig.10.2

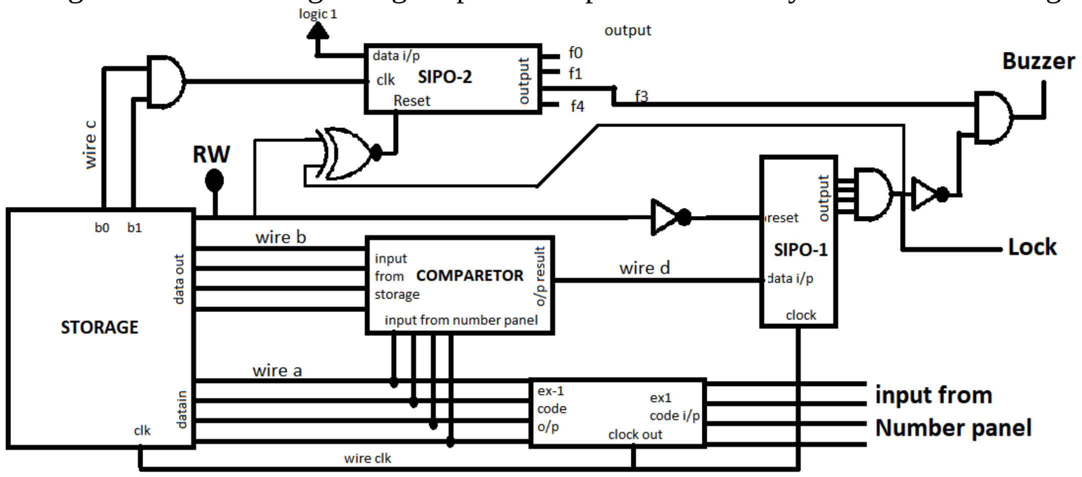


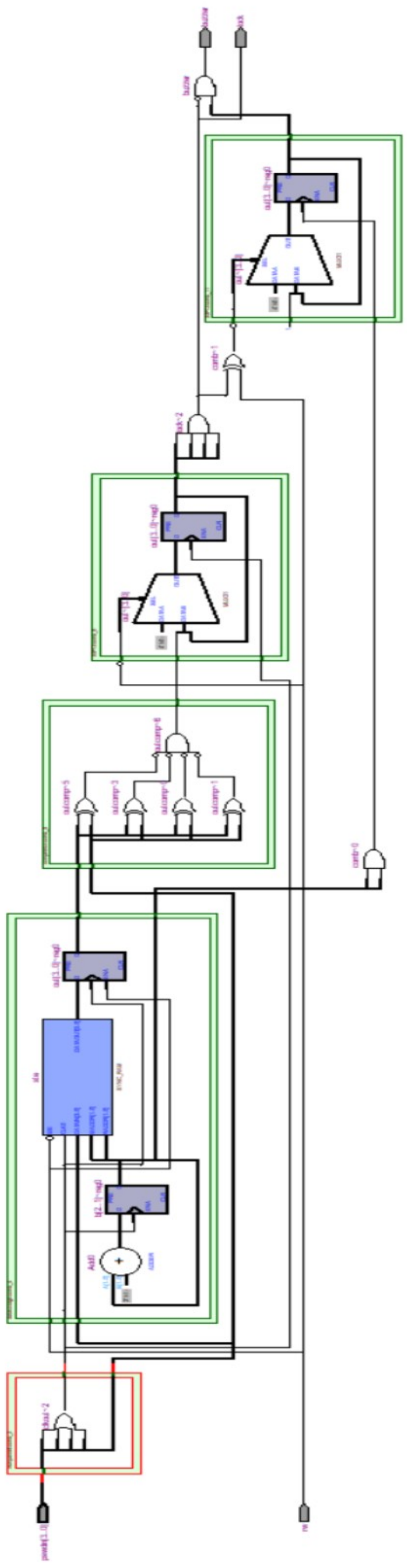
Fig 10.2

Verilog HDL code:

```
//////////////////////main module/////////////////////////////
module finallock (pswdin,rw,lock ,buzzer);
input [3:0]pswdin;
input rw;
output lock,buzzer;
wire [3:0]a;            //connecting num penal to storage & comperetor
wire [3:0]b;            //connecting storage o/p to comparetor
wire [2:1]c;            //connecting addres counter to sipo2
wire [3:0]e;            //connecting sipo 1 o/p to lock by anding all bits
wire [3:0]f;            //connecting sipo 1 o/p to buzzer
wire d;                         //connecting comparetor o/p to sipo1
wire clk;                       //connecting num penal clock o/p to various modules

numpannel(pswdin,clk,a);
datastorage(b,a,clk,c,rw);
comperetor (b,a,d);
SIPO (d,e,clk,~rw);                     //sipo 1 (that counts number of wrong attempts)
assign lock = e[0] & e[1] & e[2] & e[3];
SIPO (1,f ,c[2] & c[1],rw ~^ lock);    //sipo2(stores 4 digit result of comparetor)
assign buzzer = f[2] & (~lock);
endmodule

////////////////module for password storage///////////////////
module datastorage(out,in , clk  ,b, rw);
input [3:0]in;
output reg [2:1]b;                  //address counting reg.
input clk,rw;
reg [3:0]ste[3:0];     //4*4 reg. matrix for storing 4 digit, each of 4 bit
output reg [3:0]out;   //output port reg

always @(posedge clk )
begin
        if (rw == 1)       //rw = 1 for write mode
        begin
                out = ste[b];  // gatting data on o/p reg. from storage reg
                b = b+1;
        end

        else if(rw == 0)   //rw=0 for read mode
        begin
                 ste[b] = in;  //reseting storeg reg data
                 b = b+1;
        end
end

endmodule

//////module for compiring entaring password with stored one/////
```

```
module comperetor(instr,inuser,outcomp);
input [3:0]instr;                    //  input form storage
input [3:0]inuser;                   // input from number panel
output outcomp;                                //  o/p result of comparision

assign outcomp = ((instr[0]) ~^ inuser[0]) & (instr[1] ~^ inuser[1]) & (instr[2] ~^ inuser[2])
& (instr[3] ~^ inuser[3]);

endmodule

////////modul representing number penal for genarating clock/////
module numpannel (passin , clkout , passout);
input [3:0]passin ;
output  clkout ;
output [3:0]passout;

assign clkout = passin[0] | passin[1] | passin[2] | passin[3] ;
ssign passout = passin;

endmodule

///////////////serial in perellal o/p module///////////////////
module SIPO (in,out ,clk,reset);
input clk,in,reset;
output reg [3:0]out;

always @(posedge clk )
        begin
                if (reset == 0)                                        //sipo function on reset = 0
                begin
                        out[3]=out[2];
                        out[2]=out[1];
                        out[1]=out[0];
                        out[0]=in;
                end
                else                    //resets on reset = 1
                begin
                        out = 4'b0000;
                end
                end
endmodule
```

RTL view (4-digit password protected lock):



Simulation report (4-digit password protected lock system):

Here fig 10.3 represents simulation of condition while user enters right password in 1st attempt and we can observe that door gates unlocked. Fig 10.4 represents simulation condition while user attempts continuous 3 wrong password and we can observe that buzzer gets activated at third wrong attempt. Fig. 10.5 represents simulation condition while user enters right password after three attempts of wrong password and we can observe that buzzer could only be turn off when user enters right password and door gets unlocked at last right attempt.

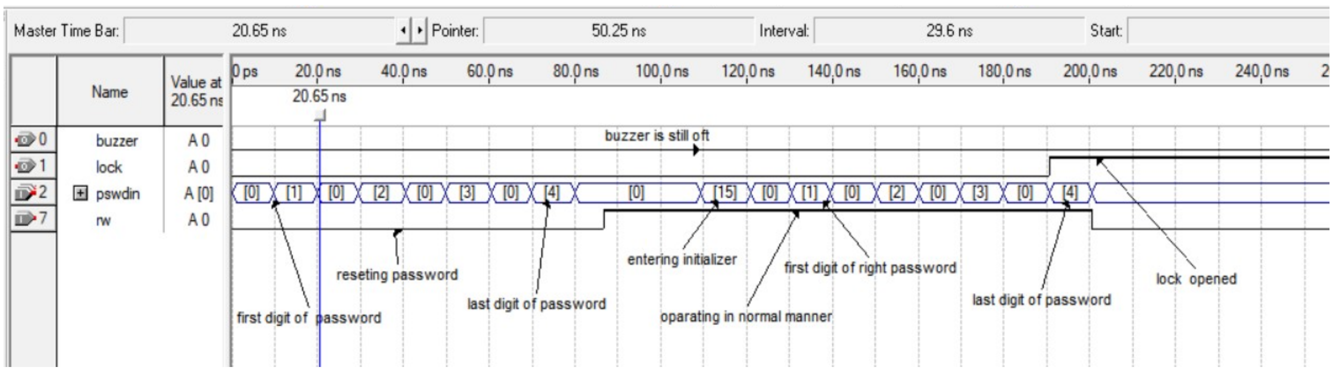


Fig 10.3

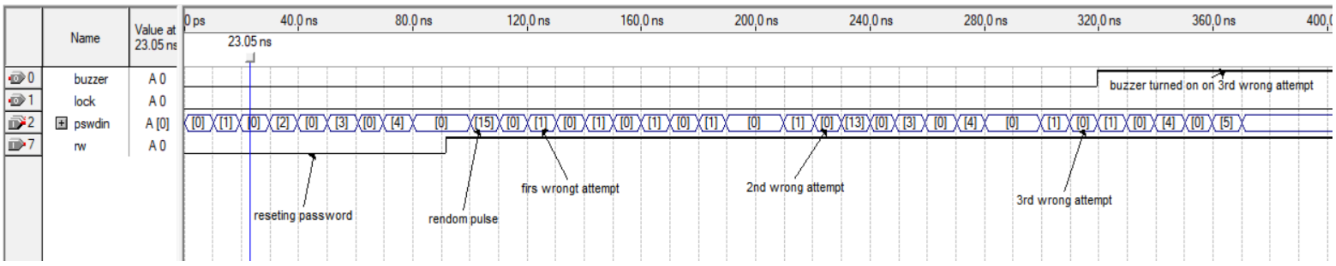


Fig. 10.4

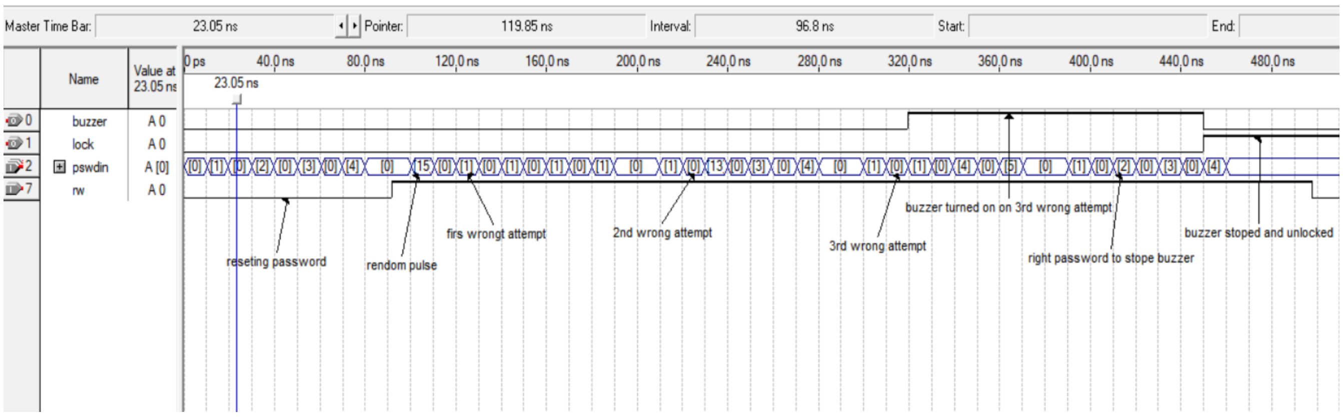


Fig.10.5