گزارش کار آزمایش 10 مدارهای منطقی

آبتین زندی 9931071

سروین نامی 9931013

داخل سیستم fsm توسط ماژول xnor comparator برابری passed data و پسورد چک شده و داخل وایر p1 ذخیره میشود وایر p2 هم p1~ است.

به عنوان رجیستر از سه تا دی فلیپ‌فلاپ استفاده شده که طبق کد نوشته شده از خط ۳۱ تا ۵۲ ورودی‌ها و خروجی‌های دی فلیپ فلاپ ها مقداردهی شدند.

خروجی های دی فلیپ فلاپ ها که q0 و q1 و q2 هستند داخل دو جعبه ی odd و even که دوتا and هستند استفاده میشود و خروجی even enable که نشان دهنده این هست که اطلاعات داخل جعبه زوج ذخیره شده باشند یک and هست که مقدار q2 و~q1 و q3 بعلاوه ریکوئست کانفیرم و مقدار نات ورودی d تعیین میشود. جعبه odd خروجی فرد است توسط and مقادیر q2 و ~q1 و q3 آتش و ریکوئست و کانفریم و خود مقدار دی تعیین میشود.

داخل محیط شبیه سازی به مقادیر ورودی یک حالت پایه داده شده که passed data ۱۰۱۰ هست و پسورد اولیه غلط و ۱۰۰۰ است و مقدار دی و ۰۰۰۱ و مقدار کلاک کانفیرم و ریکوئست هم صفر است.

کلاک به تعداد ۳۰ بار در دوره های ۱۰ نانو ثانیه تغییر میکند و مقادیر هم پس از ۲۰ نانو ثانیه تغییر میکنند تا حالات مختلف را ببینیم.

در نهایت هم در جدول زمانی یک حالت نمایش داده شده است.

```
21  module DFF(input clk,input q,output q2);
22      //a DFlipFlop by data flow design
23  reg q2;
24  always @ ( posedge clk ) begin
25          q2 <= q;
26     end
27  endmodule
```

```
21  module xnorComparator4bit(input [3:0] first,input [3:0] second,output out);
22   wire [3:0] result;
23      xnor
24        b0 (result[0], first[0], second[0]),
25        b1 (result[1], first[1], second[1]),
26        b2 (result[2], first[2], second[2]),
27        b3 (result[3], first[3], second[3]);
28
29      and finalResult (out, result[3], result[2], result[1], result[0]);
30
31  endmodule
32
```

```
module FSM(input [3:0] passedData,input clk,input confirm,input request,input [3:0] password,input d
,output evenEnable,output oddEnable);
//The FSM circuit which will manage the user inputs,and change the outputs if needed.Using 3 D-FFs and 1 comparator.

 // The password comparator
    wire p1, p2;
    xnorComparator4bit checkingPassword (passedData, password, p1);
    not pInventor (p2, p1);


    // FlipFlops inputs and outputs
    wire q0, q02, q1, q12, q2, q22, in0, in1, in2;
    not
      Inventor0 (q02, q0),
      Inventor1 (q12, q1),
      Inventor2 (q22, q2);

    // N2 Flip Flop input
    wire n2MiddleWire;
    or n2Middle (n2MiddleWire, q2, confirm);
    and n2Input (in2, q0, q12, request, n2MiddleWire);

    // N1 Flip Flop input
    wire n1MiddleWire;
    or n1Middle (n1MiddleWire, q2, p2);

     // N0 Flip Flop input
     wire n0MiddleWire1, n0MiddleWire2, CP;
     not cInventor (CP, confirm);
     and n0Middle1 (n0MiddleWire1, q0, CP);
     or n0Middle2 (n0MiddleWire2, q22, n0MiddleWire1);
     and n0Input (in0, q12, request, n0MiddleWire2);


     // Flip Flops instantiates
     DFF
       n0 (clk, in0, q0),
       n1 (clk, in1, q1),
       n3 (clk, in2, q2);


     // Output functions
     wire dp;
     not dInventor (dp, d);
     and
       evenOut (evenEnable, q2, q12, q0, request, confirm, dp),
       oddOut (oddEnable, q2, q12, q0, request, confirm, d);
endmodule
```

```
50
51      initial begin
52         // Clock intialize
53         clk = 0;
54         repeat (30)
55            #10 clk = ~ clk;
56      end
57
58      initial begin
59         // Initialize Inputs
60         passedData = 4'b1010;
61         clk = 0;
62         confirm = 0;
63         request = 0;
64         password = 4'b1000;
65         d = 4'b0001;
66
67         // Wait 100 ns for global reset to finish
68         #100;
69
70         // Add stimulus here
71         // Defining a process
72         # 10
73         request = 1;
74         # 20
75         confirm = 1;
76         # 20
77         confirm = 1;
78         # 20
79         request = 0;
80         confirm = 0;
81         # 20
82         password = 4'b1010;
83         request = 1;
84         confirm = 1;
85         # 20
86         d = 4'b0000;
87         confirm = 1;
88         # 20
89         request = 0;
90         confirm = 0;
91         # 20
92         password = 4'b1010;
93         request = 1;
94         confirm = 1;
95         # 20
96         d = 4'b0011;
97         confirm = 0;
98         # 20
99         request = 0;
100
101        confirm = 0;
102        # 20
103        password = 4'b1010;
104        request = 1;
105        confirm = 1;
106        # 20
107        d = 4'b0011;
108        confirm = 1;
109        # 20
110        request = 0;
111        $finish;
```