Ain Shams University Faculty of Engineering CHEP- CESS

Mohammed Ehab Elsaeed 16P8160 m.elsaeed1998@gmail.com

Electronic Design Automation
CSE 215
Digital Access Control Finite State Machine
Project 1

## Introduction

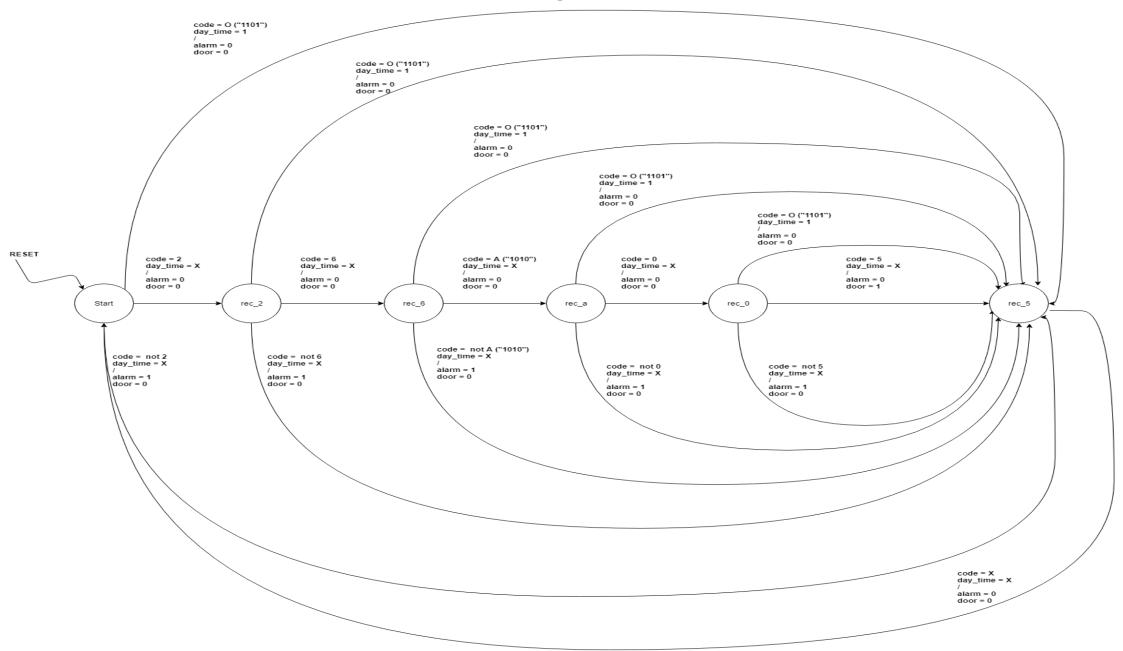
The purpose of this document is to illustrate the work done to create a digital access control finite state machine in accordance to the specifications provided.

The finite state machine was implemented as a **Mealy** machine in an attempt to reduce the number of states due to the fact that outputs depend on transitions instead of states. As a sacrifice, the finite state machine is supposedly less stable than a Moore machine.

## **Table of Contents**

- 1- Finite State Machine Diagram
- 2- Testbench Strategy
- 3- Simulation of The Finite State Machine
- 4- Finite State Machine VHDL Code
- 5- Finite State Machine VHDL Testbench Code

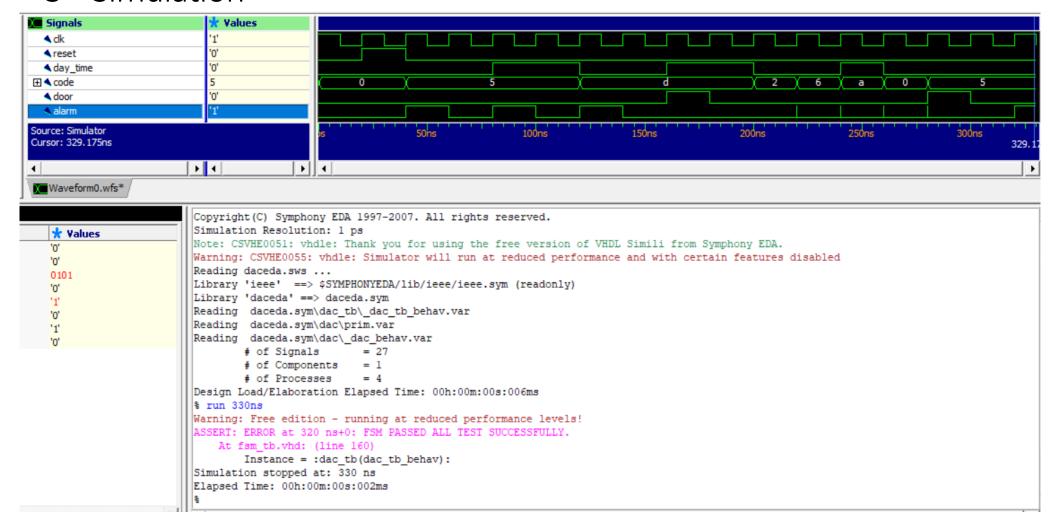
# 1- Finite State Machine State Diagram



# 2- Testbench Strategy

reset	day_time	code	door	alarm	Justification
1	X	Χ	0	0	Setting reset = 1 should reset the circuit to its initial state
0	0	5	0	1	Entering a wrong code, should trigger the alarm
0	X	X	0	0	Waiting for a clock cycle, should reset both door and alarm, to zero
0	1	5	0	1	Entering a wrong code, even during daytime should trigger the alarm
0	Х	Х	0	0	Waiting for a clock cycle, should reset both door and alarm, to zero
0	0	"O"	0	1	Entering "O" during night time should trigger the alarm
0	Х	Х	0	0	Waiting for a clock cycle, should reset both door and alarm, to zero
0	1	"O"	1	0	Entering "O" during daytime should open the door
0	Х	X	0	0	Waiting for a clock cycle, should reset both door and alarm, to zero
0	0	2	0	0	Entering 2 neither opens the door, nor triggers the alarm
0	0	6	0	0	Entering 2, 6 neither opens the door, nor triggers the alarm
0	1	"A"	0	0	Entering 2, 6, A neither opens the door, nor triggers the alarm, switching daytime to 1 and not pressing "O", will not open the door nor trigger the alarm
0	0	0	0	0	Entering 2, 6, A, 0 neither opens the door, nor triggers the alarm
0	0	5	1	0	Entering 2, 6, A, 0, 5 opens the door, but doesn't trigger the alarm
0	Х	X	0	0	Waiting for a clock cycle, should reset both door and alarm, to zero

### 3- Simulation



Note: "d" is the hexadecimal code for "1101" (Which is the code of input O)

Glitches in the alarm signal are due to the fact of using a Mealy machine.

The system **passes** all the assertions.

### 4- Finite State Machine VHDL Code

```
-- 0 to 9 binary encoded
6-
    -- A => 1010
7-
    -- B => 1011
8-
    -- 0 => 1101
9-
     entity dac is
10-
         port (
             reset : in bit;
11-
12-
             day time : in bit;
13-
                      : in bit vector(3 downto 0);
             code
14-
             door
                      : out bit;
15-
             alarm
                      : out bit;
16-
             c1k
                      : in bit;
17-
             vdd
                      : in bit;
18-
                     : in bit
             VSS
19-
         );
20-
    end dac;
21- architecture dac_behav of dac is
22-
         type state is (start, rec 2, rec 6, rec a, rec 0, rec 5);
23-
         signal current state : state;
24-
         signal next state : state;
25-
         constant a
                               : bit vector(3 downto 0) := "1010";
26-
                              : bit_vector(3 downto 0) := "1011";
         constant b
27-
                               : bit vector(3 downto 0) := "1101";
         constant o
28-
         --pragma current state current state
29-
         --pragma next_state next_state
30-
         --pragma clock clk
31-
    begin
32-
         process (clk)
33-
         begin
34-
             if clk = '1' and not clk'stable then
35-
                 current_state <= next_state;</pre>
36-
             end if;
37-
         end process;
38-
         process (current_state, reset, day_time, code)
39-
         begin
40-
             if reset = '1' then
41-
                 next state <= start;</pre>
42-
             else
43-
                 case current_state is
44-
                     when start =>
```

```
45-
                         if day_time = '1' and code = o then
46-
                             door <= '1';</pre>
47-
                             alarm <= '0';
48-
                             next state <= rec 5;</pre>
49-
                         elsif code = x"2" then
                                   <= '0';
50-
                             door
51-
                             alarm <= '0';
52-
                             next_state <= rec_2;</pre>
53-
                         else
                             door <= '0';
54-
55-
                             alarm <= '1';
                             next_state <= rec_5;</pre>
56-
57-
                         end if;
58-
                     when rec 2 =>
59-
                         if day_time = '1' and code = o then
                             door <= '1';
60-
61-
                             alarm <= '0';
62-
                             next_state <= rec_5;</pre>
63-
                         elsif code = x"6" then
64-
                             door <= '0':
                             alarm <= '0';
65-
66-
                             next_state <= rec_6;</pre>
67-
                         else
68-
                             door <= '0';</pre>
                             alarm <= '1';
69-
70-
                             next state <= rec 5;</pre>
71-
                         end if;
72-
                     when rec 6 =>
73-
                         if day time = '1' and code = o then
                             door <= '1';
74-
                             alarm <= '0';
75-
76-
                             next state <= rec 5;</pre>
77-
                         elsif code = a then
78-
                             door
                                       <= '0';
79-
                             alarm <= '0';
80-
                             next_state <= rec_a;</pre>
81-
                         else
                             door <= '0';
82-
83-
                             alarm <= '1';
84-
                             next_state <= rec_5;</pre>
85-
                         end if;
86-
                     when rec a =>
```

```
87-
                        if day_time = '1' and code = o then
88-
                            door <= '1';</pre>
89-
                            alarm <= '0';
90-
                            next_state <= rec_5;</pre>
91-
                        elsif code = x"0" then
                            door <= '0';
92-
93-
                            alarm <= '0';
94-
                            next_state <= rec_0;</pre>
95-
                        else
                            door <= '0';
96-
97-
                            alarm <= '1';
98-
                            next_state <= rec_5;</pre>
99-
                        end if;
100-
                    when rec 0 =>
                        if day_time = '1' and code = o then
101-
                            door <= '1';
102-
103-
                            alarm <= '0';
104-
                            next_state <= rec_5;</pre>
                        elsif code = x"5" then
105-
106-
                            door <= '1':
107-
                            alarm <= '0';
108-
                            next_state <= rec_5;</pre>
109-
                        else
110-
                            door <= '0';</pre>
                            alarm <= '1';
111-
112-
                            next state <= rec 5;</pre>
113-
                        end if;
                    when rec_5 =>
114-
                        115-
116-
117-
                        next state <= start;</pre>
118-
                    when others =>
119-
                        assert false
120-
                        report "Invalid state"
121-
                            severity failure;
122-
          end case;
            end if;
123-
124- end process;
125- end dac behav;
```

### 5- Testbench VHDL Code

```
6-
    entity dac tb is
    end dac_tb;
7-
8-
9 -
    architecture dac tb behav of dac tb is
10-
         component dac is
             port (
11-
12-
                 reset : in bit;
13-
                 day time : in bit;
14-
                         : in bit vector(3 downto 0);
                 code
15-
                 door
                         : out bit;
16-
                 alarm : out bit;
17-
                 clk
                         : in bit;
18-
                 vdd
                         : in bit;
19-
                       : in bit
                 VSS
20-
             );
21-
        end component dac;
22-
        signal reset
                             : bit;
23-
        signal day time
                            : bit;
24-
        signal code
                            : bit vector(3 downto 0);
25-
        signal door
                            : bit;
26-
        signal alarm
                             : bit;
27-
        signal clk
                            : bit;
28-
        signal vdd
                            : bit := '1';
                             : bit := '0';
29-
        signal vss
30-
31-
        for all
                             : dac use entity work.dac(dac_behav);
32-
33-
        constant clk_period : time
                                                            := 20
    ns;
                             : bit vector(3 downto 0) := "1010";
34-
        constant a
                            : bit_vector(3 downto 0) := "1011";
35-
        constant b
                           : bit vector(3 downto 0) := "1101";
36-
        constant o
37-
    begin
38-
39-
        dut : dac port map(reset, day_time, code, door, alarm, clk,
     vdd, vss);
40-
        process begin
41-
42-
            wait for clk_period;
43-
```

```
44-
             reset <= '1';
45-
             -- day time<='0';</pre>
46-
47-
             -- code <= x"";</pre>
48-
             wait for clk period;
49-
             assert door = '0' and alarm = '0'
50-
             report "Setting reset = 1 should reset the circuit to
     its initial state"
51-
                 severity failure;
52-
53-
54-
             reset <= '0';
             day time <= '0';</pre>
55-
56-
             code <= x"5";
57-
             wait for clk period;
             assert door = '0' and alarm = '1'
58-
             report "Entering a wrong code, should trigger the
59-
     alarm"
60-
                 severity failure;
61-
62-
             wait for clk period;
63-
             assert door = '0' and alarm = '0'
64-
65-
             report "Waiting for a clock cycle, should reset both
     door and alarm, to zero"
66-
                 severity failure;
67-
68-
69-
                     <= '0':
             reset
70-
             day time <= '1';</pre>
             code <= x"5";
71-
             wait for clk period;
72-
73-
             assert door = '0' and alarm = '1'
74-
             report "Entering a wrong code, even during daytime
     should trigger the alarm"
75-
                 severity failure;
76-
77-
78-
             wait for clk period;
             assert door = '0' and alarm = '0'
79-
             report "Waiting for a clock cycle, should reset both
80-
     door and alarm, to zero"
```

```
81-
                 severity failure;
82-
83-
84-
                     <= '0';
             reset
85-
             day time <= '0';</pre>
86-
                      <= o;
             code
87-
             wait for clk_period;
             assert door = '0' and alarm = '1'
88-
89-
             report "Entering 'O' during night time should trigger
     the alarm"
90-
                 severity failure;
91-
92-
93-
             wait for clk period;
             assert door = '0' and alarm = '0'
94-
95-
             report "Waiting for a clock cycle, should reset both
     door and alarm, to zero"
                 severity failure;
96-
97-
98-
99-
             reset <= '0';
             day time <= '1';</pre>
100-
101-
             code
                       <= o:
102-
             wait for clk period;
             assert door = '1' and alarm = '0'
103-
104-
             report "Entering 'O' during daytime should open the
     door"
105-
                 severity failure;
106-
107-
108-
             wait for clk period;
             assert door = '0' and alarm = '0'
109-
             report "Waiting for a clock cycle, should reset both
110-
    door and alarm, to zero"
111-
                 severity failure;
112-
113-
             -- 10
114-
             reset <= '0';
115-
             day time <= '0';</pre>
116-
             code <= x"2";
117-
             wait for clk_period;
             assert door = '0' and alarm = '0'
118-
```

```
119-
             report "Entering 2 neither opens the door, nor triggers
  the alarm"
                 severity failure;
120-
121-
122-
             -- 11
123-
             reset <= '0';
             day time <= '0';</pre>
124-
                   <= x"6";
             code
125-
126-
             wait for clk period;
             assert door = '0' and alarm = '0'
127-
             report "Entering 2, 6 neither opens the door, nor
128-
    triggers the alarm"
129-
                 severity failure;
130-
131-
             -- 12
132-
             reset <= '0';
             day time <= '1';</pre>
133-
134-
             code
                      <= a;
135-
             wait for clk period;
             assert door = '0' and alarm = '0'
136-
             report "Entering 2, 6, A neither opens the door, nor
137-
    triggers the alarm, switching daytime to 1 and not pressing '0'
     , will not open the door nor trigger the alarm"
                 severity failure;
138-
139-
140-
             -- 13
             reset <= '0';
141-
             day time <= '0';</pre>
142-
143-
             code
                     <= x"0";
144-
             wait for clk period;
             assert door = '0' and alarm = '0'
145-
146-
             report "Entering 2, 6, A, 0 neither opens the door, nor
    triggers the alarm"
147-
                 severity failure;
148-
             -- 14
149-
150-
             reset <= '0';
             day time <= '0';</pre>
151-
152-
             code
                      <= x"5";
153-
             wait for clk period;
             assert door = '1' and alarm = '0'
154-
```

```
report "Entering 2, 6, A, 0, 5 opens the door, but
155-
 doesn't trigger the alarm"
156-
             severity failure;
157-
        -- 15
158-
          wait for clk_period;
159-
           assert door = '0' and alarm = '0'
160-
            report "Waiting for a clock cycle, should reset both
161-
   door and alarm, to zero"
               severity failure;
162-
163-
164-
            assert false
165-
        report "FSM PASSED ALL TEST SUCCESSFULLY."
166-
                severity error;
167-
            wait;
168-
     end process;
169-
170- process begin
            clk <= '1', '0' after (clk_period/2);</pre>
171-
172-
            wait for clk period;
173-
        end process;
174-
175- end architecture dac tb behav;
```