

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

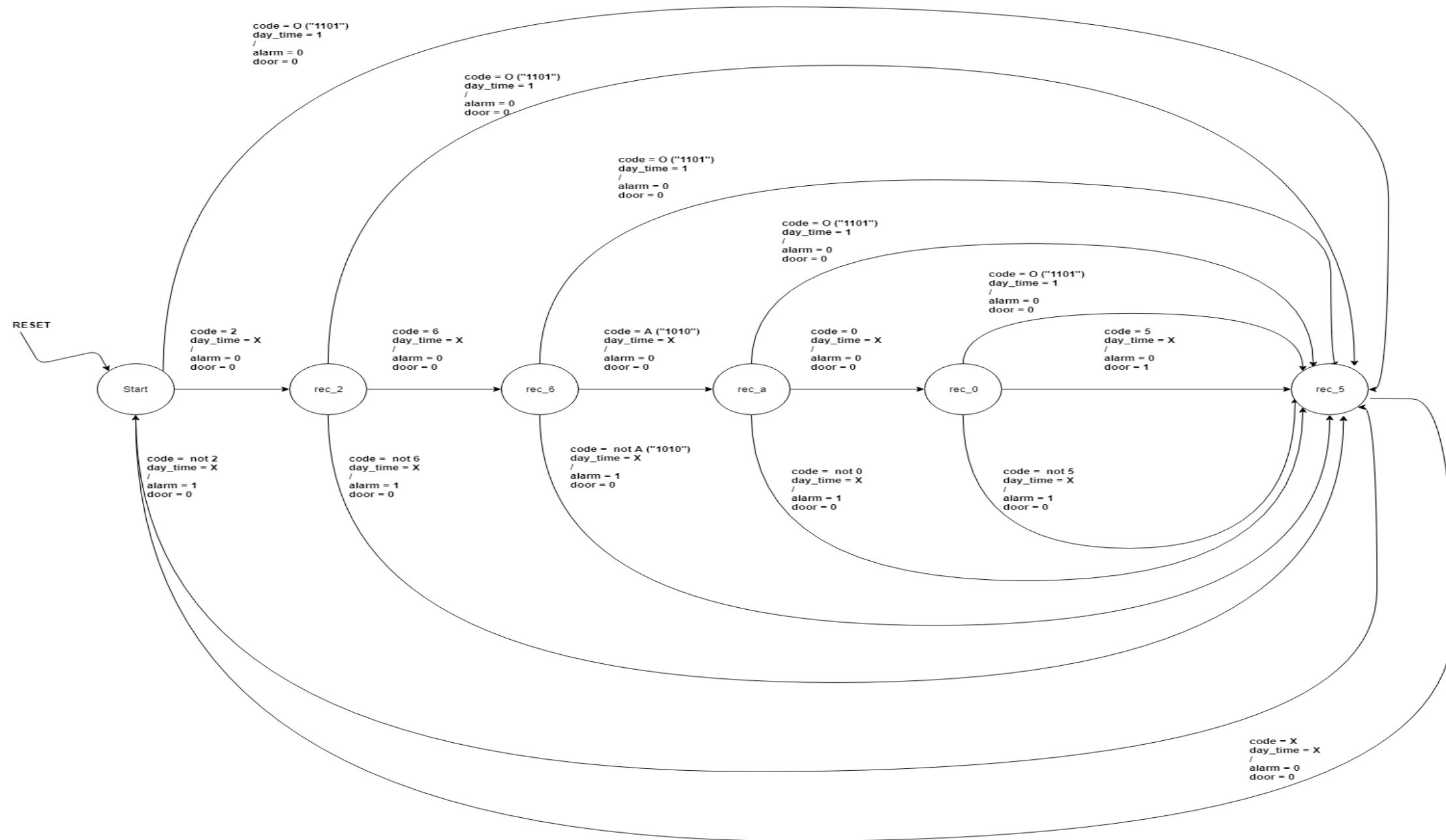
The purpose of this document is to illustrate the work done to create a digital access control finite state machine in accordance to the specifications provided.

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

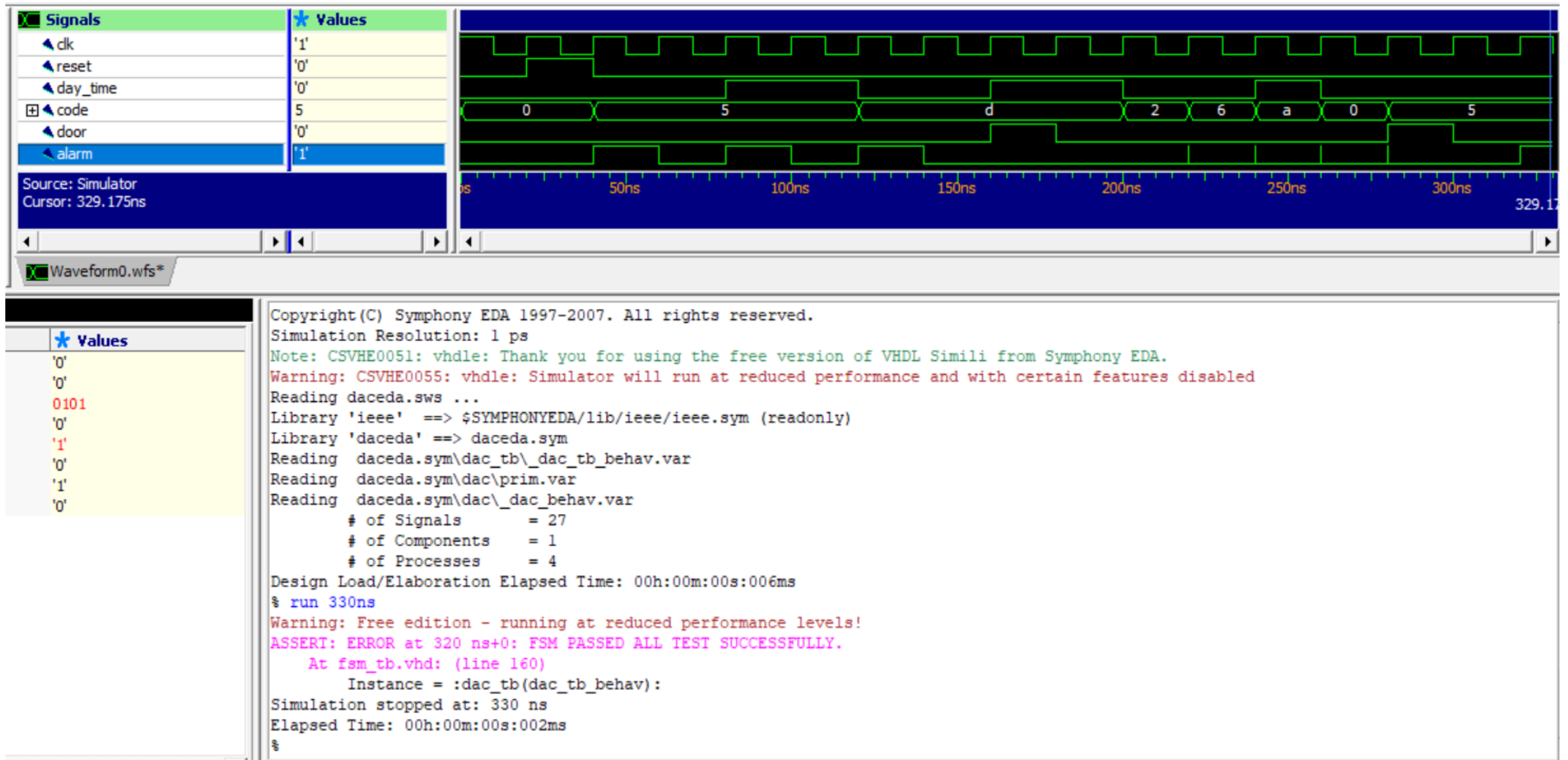
## 1- Finite State Machine State Diagram



## 2- Testbench Strategy

reset	day_time	code	door	alarm	Justification
1	X	X	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	<b>Waiting for a clock cycle, should reset both door and alarm, to zero</b>
0	1	5	0	1	Entering a wrong code, even during daytime should trigger the alarm
0	X	X	0	0	<b>Waiting for a clock cycle, should reset both door and alarm, to zero</b>
0	0	"O"	0	1	Entering "O" during night time should trigger the alarm
0	X	X	0	0	<b>Waiting for a clock cycle, should reset both door and alarm, to zero</b>
0	1	"O"	1	0	Entering "O" during daytime should open the door
0	X	X	0	0	<b>Waiting for a clock cycle, should reset both door and alarm, to zero</b>
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	X	X	0	0	<b>Waiting for a clock cycle, should reset both door and alarm, to zero</b>

# 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

```
5-  -- 0 to 9 binary encoded
6-  -- A => 1010
7-  -- B => 1011
8-  -- 0 => 1101
9-  entity dac is
10-      port (
11-          reset      : in bit;
12-          day_time   : in bit;
13-          code       : in bit_vector(3 downto 0);
14-          door       : out bit;
15-          alarm      : out bit;
16-          clk        : in bit;
17-          vdd        : in bit;
18-          vss        : in bit
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-      constant b           : bit_vector(3 downto 0) := "1011";
27-      constant o           : bit_vector(3 downto 0) := "1101";
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;
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;
42-          else
43-              case current_state is
44-                  when start =>
```

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

```

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

```



## 5- Testbench VHDL Code

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

```
44-         -- 1
45-         reset <= '1';
46-         -- day_time<='0';
47-         -- code <= x"";
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-         -- 2
54-         reset      <= '0';
55-         day_time <= '0';
56-         code       <= x"5";
57-         wait for clk_period;
58-         assert door = '0' and alarm = '1'
59-         report "Entering a wrong code, should trigger the
alarm"
60-             severity failure;
61-
62-         -- 3
63-         wait for clk_period;
64-         assert door = '0' and alarm = '0'
65-         report "Waiting for a clock cycle, should reset both
door and alarm, to zero"
66-             severity failure;
67-
68-         -- 4
69-         reset      <= '0';
70-         day_time <= '1';
71-         code       <= x"5";
72-         wait for clk_period;
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-         -- 5
78-         wait for clk_period;
79-         assert door = '0' and alarm = '0'
80-         report "Waiting for a clock cycle, should reset both
door and alarm, to zero"
```

```
81-         severity failure;
82-
83-         -- 6
84-         reset      <= '0';
85-         day_time <= '0';
86-         code       <= 0;
87-         wait for clk_period;
88-         assert door = '0' and alarm = '1'
89-         report "Entering '0' during night time should trigger
the alarm"
90-         severity failure;
91-
92-         -- 7
93-         wait for clk_period;
94-         assert door = '0' and alarm = '0'
95-         report "Waiting for a clock cycle, should reset both
door and alarm, to zero"
96-         severity failure;
97-
98-         -- 8
99-         reset      <= '0';
100-        day_time <= '1';
101-        code       <= 0;
102-        wait for clk_period;
103-        assert door = '1' and alarm = '0'
104-        report "Entering '0' during daytime should open the
door"
105-        severity failure;
106-
107-        -- 9
108-        wait for clk_period;
109-        assert door = '0' and alarm = '0'
110-        report "Waiting for a clock cycle, should reset both
door and alarm, to zero"
111-        severity failure;
112-
113-        -- 10
114-        reset      <= '0';
115-        day_time <= '0';
116-        code       <= x"2";
117-        wait for clk_period;
118-        assert door = '0' and alarm = '0'
```

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

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