```vhdl
 1    library ieee;
 2    use ieee.std_logic_1164.all;
 3
 4    entity code_lock is
 5    port( clk, reset, a, b, c  : in std_logic;
 6          z                    : out std_logic;
 7          state                : out std_logic_vector(3 downto 0)
 8          );
 9    end code_lock;
10
11    architecture twoprocess of code_lock is
12       type statecheck is (s1, s2, s3, s4, s5, s6, s7, s8, s9);
13       signal present_state, next_state : statecheck;
14       signal countattempt : integer;
15
16    begin
17       state_reg : process ( clk, reset, countattempt )
18       begin
19          if reset = '0' then
20             present_state <= s1;
21             countattempt <= 0;
22          elsif rising_edge(clk) then
23             present_state <= next_state;
24          if (present_state = s9) then
25             countattempt <= countattempt +1;
26          end if;
27       end if;
28       end process;
29
30       outputs : process (countattempt, present_state, a, b, c)
31       begin
32          case present_state is
33
34             when s1 =>
35                   state <= "0001";
36                   z <= '0';
37                if (a= '1' and b= '1' and c= '1') then
38                   next_state <= present_state;
39                elsif ( a= '1' and b= '0' and c= '1') then
40                   next_state <= s2;
41                else
42                   next_state <= s9;
43                end if;
44
45             when s2 =>
46                   state <= "0010";
47                   z <= '0';
48                if (a= '1' and b= '1' and c= '1') then
49                   next_state <= s3;
50                elsif (a= '1' and b= '0' and c= '1') then
51                   next_state <= present_state;
52                else
53                   next_state <= s9;
54                end if;
55
56             when s3 =>
57                   state <= "0011";
58                   z <= '0';
59                if (a= '1' and b= '1' and c= '1') then
60                   next_state <= present_state;
61                elsif (a= '0' and b= '1' and c= '1') then
62                   next_state <= s4;
63                else
64                   next_state <= s9;
65                end if;
66
```

```vhdl
 67             when s4 =>
 68                     state <= "0100";
 69                     z <= '0';
 70                 if (a= '1' and b= '1' and c= '1') then
 71                     next_state <= s5;
 72                 elsif (a= '0' and b= '1' and c= '1') then
 73                     next_state <= present_state ;
 74                 else
 75                     next_state <= s9;
 76                 end if;
 77
 78             when s5 =>
 79                     state <= "0101";
 80                     z <= '0';
 81                 if (a= '1' and b= '1' and c= '1') then
 82                     next_state <= present_state ;
 83                 elsif (a= '0' and b= '1' and c= '1') then
 84                     next_state <= s6;
 85                 else
 86                     next_state <= s9;
 87                 end if;
 88
 89             when s6 =>
 90                     state <= "0110";
 91                     z <= '0';
 92                 if ( a= '1' and b= '1' and c= '1') then
 93                     next_state <= s7;
 94                 elsif (a= '0' and b= '1' and c= '1') then
 95                     next_state <= present_state ;
 96                 else
 97                     next_state <= s9;
 98                 end if;
 99
100             when s7 =>
101                     state <= "0111";
102                     z <= '1';
103                 if (a= '1' and b= '1' and c= '1') then
104                     next_state <= present_state ;
105                 else
106                     next_state <= s8;
107                 end if;
108
109             when s8 =>
110                     z <= '0';
111                     state <= "1000";
112                 if ((a= '1' and b= '1' and c= '1') and countattempt < 3) then
113                     next_state <= s1;
114                 else
115                     next_state <= present_state ;
116                 end if;
117
118             when s9 =>
119                     state <= "1001";
120                     next_state <= s8;
121         end case;
122       end process;
123   end twoprocess;
```