```
1   library ieee;
2   use ieee.std_logic_1164.all;
3   use ieee.numeric_std.all;
4
5   Entity MOORE_SM IS Port
6   (
7           clk_input, rst_n, Extender_en, Extend_tog                : in std_logic;
8           leds                                                     : in std_logic_vector(3
    downto 0);
9           shift_reg_en, shift_reg_dir, Extender_out, Grappler_en   : out std_logic
10   );
11  END ENTITY;
12
13   Architecture SM of MOORE_SM is
14
15   TYPE STATE_NAME IS (Start,Extending,Extender_hold, Extended, Retracting, Retracter_hold,
    Retracted);   -- All the STATE_NAME values which are start, extender extending, extender
    hold, fully extended, retracting, retracter hold and fully retracted
16
17
18   SIGNAL current_state, next_state   :  STATE_NAME;       -- signals of type STATE_NAME
19
20
21    BEGIN
22
23   --------------------------------------------------------------------------------
24   --State Machine:
25   --------------------------------------------------------------------------------
26
27   -- REGISTER_LOGIC PROCESS:
28
29  Register_Section: PROCESS (clk_input, rst_n, next_state)  -- this process synchronizes the
    activity to a clock
30  BEGIN
31     IF (rst_n = '0') THEN
32        current_state <= Start;
33     ELSIF(rising_edge(clk_input)) THEN
34        current_state <= next_State;
35     END IF;
36  END PROCESS;
37
38   -- Here Extend_tog is the input from the push button pb(1)
39
40  -- TRANSITION LOGIC PROCESS
41
42  Transition_Section: PROCESS (Extender_en, Extend_tog, leds, current_state) -- Sensitivity
    list
43
44  BEGIN
45       CASE current_state IS
46           WHEN Start =>
47              next_state <= Start;
48              IF(Extender_en ='1' AND Extend_tog='0') THEN
49                 next_state <= Extending;  -- If the push button is pressed and extender
    enable is 1, then the next state should be extending
50              END IF;
51
52           WHEN Extending =>
53              next_state <= Extending;
54              IF(leds = "1111") THEN
55                 next_state <= Extended;  -- If the leds(7..4) are all lighted up, then that
    means the extender is fully extended and we need to move to the extended state
56              ELSIF (Extend_tog = '1') THEN
57                 next_state <= Extender_hold;  -- If we release the push button midway, then
    we should move to the hold state
58                 END IF;
59
60
61           WHEN Extender_hold =>
62              next_state <= Extender_hold;
63              IF(Extend_tog='0') THEN
64                 next_state <= Extending;  -- If we again press the push button, it should
    resume extending
65              END IF;
66
67           WHEN Extended =>
```

```
68              next_state <= Extended;
69              IF(Extend_tog='0') THEN
70                 next_state <= Retracting;  -- Here when the push button is presses, it would
    retract now
71              END IF;
72
73           WHEN Retracting =>
74              next_state <= Retracting;
75              IF(leds = "0000")THEN
76                 next_state <= Retracted;  -- If all the leds are down, then that means the
    extender has retracted so we move to that state
77              ELSIF (Extend_tog = '1') THEN
78                 next_state <=  Retracter_hold; -- Similarly if we release the button midway,
    it is now in hold state
79                 END IF;
80
81           WHEN Retracter_hold =>
82              next_state <= Retracter_hold;
83              IF(Extend_tog='0') THEN
84                 next_state <= Retracting;  -- If the button is pressed again, it should keep
    on retracting
85                 END IF;
86
87           WHEN Retracted =>
88              next_state <= Retracted;
89              IF(Extend_tog='0' AND Extender_en ='1') THEN
90                 next_state <= Extending;  -- After it has been retracted and we press the
    push button along with the enabler being 1, it should start extending again
91              END IF;
92        END CASE;
93    END PROCESS;
94
95    -- DECODER SECTION PROCESS
96    Decoder_Section: PROCESS (current_state)
97
98    BEGIN
99         CASE current_state IS
100          WHEN Start =>
101          shift_reg_en <= '0';  -- Everything should be 0 because we don't want to do anything
102          shift_reg_dir <= '0';
103          Extender_out <= '0';
104          Grappler_en <= '0';
105
106          WHEN Extending =>    -- Here since leds should increase to the right,
    shift_reg_dir is 1 along with enable and the extender will be out
107          shift_reg_en <= '1';
108          shift_reg_dir <= '1';
109          Extender_out <= '1';
110          Grappler_en <= '0';
111
112          WHEN Extended =>     -- when it is extended, we can use the grappler, so
    grappler_en is 1
113          shift_reg_en <= '0';
114          shift_reg_dir <= '0';
115          Extender_out <= '1';
116          Grappler_en <= '1';
117
118          WHEN Retracting =>   -- While retracting the leds will go down in the left
    direction so shhift_reg_dir is 0
119          shift_reg_en <= '1';
120          shift_reg_dir <= '0';
121          Extender_out <= '1';
122          Grappler_en <= '0';
123
124          WHEN Retracted =>    -- after the extender has been retracted, everything should
    be 0 because we cannot do anything
125          shift_reg_en <= '0';
126          shift_reg_dir <= '0';
127          Extender_out <= '0';
128          Grappler_en <= '0';
129
130          WHEN others =>      -- In all other cases, the extender will be open so its value
    is 1
131          shift_reg_en <= '0';
132          shift_reg_dir <= '0';
133          Extender_out <= '1';
```

```
134             Grappler_en <= '0';
135         END CASE;
136     END PROCESS;
137
138     END ARCHITECTURE SM;
```