

```
1  -----
2  -- Company:
3  -- Engineer:
4  --
5  -- Create Date:    21:49:37 05/14/2021
6  -- Design Name:
7  -- Module Name:    MotorController - Behavioral
8  -- Project Name:
9  -- Target Devices:
10 -- Tool versions:
11 -- Description:
12 --
13 -- Dependencies:
14 --
15 -- Revision:
16 -- Revision 0.01 - File Created
17 -- Additional Comments:
18 --
19 -----
20 library IEEE;
21 use IEEE.STD_LOGIC_1164.ALL;
22
23 -- Uncomment the following library declaration if using
24 -- arithmetic functions with Signed or Unsigned values
25 --use IEEE.NUMERIC_STD.ALL;
26
27 -- Uncomment the following library declaration if instantiating
28 -- any Xilinx primitives in this code.
29 --library UNISIM;
30 --use UNISIM.VComponents.all;
31
32 entity MotorController is
33     Port ( reset : in  STD_LOGIC;
34           MotorPulseInput : in  STD_LOGIC;
35           Enable : in  STD_LOGIC; -- To Enable The Motor To Turn --
36           Clockwise : in  STD_LOGIC; -- To Choose Direction To Turn The Stepper
37           Motor --
38           MotorEnablerInput : in  STD_LOGIC; -- To Enable And Disable The Stepper
39           Motor --
40           MotorOutput : out  STD_LOGIC_VECTOR (3 downto 0) -- Output for Motor --
41           );
42 end MotorController;
43
44 architecture Behavioral of MotorController is
45
46     type StateType is (S0, S1, S2, S3);
47     -- S0 = '1000' as Motor Output --
48     -- S1 = '0010' as Motor Output --
49     -- S2 = '0100' as Motor Output --
50     -- S3 = '0001' as Motor Output --
51
52     signal State, NextState : StateType; -- Moore Machine
53
54 begin
55     SyncProcess:
56         process (reset, MotorPulseInput)
```

```
56
57     begin
58         if (reset = '1') then -- If reset Button Is Pressed --
59             State <= S0; -- Change The State Back To S0 --
60
61             elsif (rising_edge(MotorPulseInput)) then -- If Positive Clock Edge of
MotorPulse --
62                 State <= NextState; -- Change The State To The Next State --
63
64             end if;
65
66         end process;
67
68     MotorCombinationProcess:
69         process (State, MotorEnablerInput, Enable, Clockwise)
70
71         begin
72
73             MotorOutput <= "1000"; -- Set Motor Output To '1000' --
74             NextState <= S0; -- Set The State To S0 --
75
76             case State is
77
78             when S0 => -- When State Is At S0 --
79                 MotorOutput <= "1000"; -- Set Motor Output To '1000' --
80
81                 if (MotorEnablerInput = '1') then -- Is MotorEnablerInput Is 1 --
82
83                     if (Enable = '1') then -- If Enable Is 1 --
84
85                         if (Clockwise = '1') then -- If Clockwise Is 1 --
86                             NextState <= S1; -- Change The State To The S1 --
87
88                         else -- If Clockwise Is 0 --
89                             NextState <= S3; -- Change The State To The S3 --
90
91                         end if;
92
93                     else -- If Enable Output Is 0 --
94                         NextState <= S0; -- Hold The Current State --
95
96                     end if;
97
98                     else -- Is MotorEnablerInput Is 0 --
99                         NextState <= S0; -- Hold The Current State --
100
101                     end if;
102
103             when S1 => -- When State Is At S1 --
104                 MotorOutput <= "0010"; -- Set Motor Output To '0010' --
105
106                 if (MotorEnablerInput = '1') then -- Is MotorEnablerInput Is 1 --
107
108                     if (Enable = '1') then -- If Enable Is 1 --
109
110                         if (Clockwise = '1') then -- If Clockwise Is 1 --
111                             NextState <= S2; -- Change The State To The S2 --
```

```
112
113         else -- If Clockwise Is 0 --
114             nextState <= S0; -- Change The State To The S0 --
115
116         end if;
117
118         else -- If Enable Is 0 --
119             nextState <= S1; -- Hold The Current State --
120
121         end if;
122
123         else -- Is MotorEnablerInput Is 0 --
124             nextState <= S1; -- Hold The Current State --
125
126         end if;
127
128     when S2 => -- When State Is At S2 --
129         MotorOutput <= "0100"; -- Set Motor Output To '0100' --
130
131         if (MotorEnablerInput = '1') then -- Is MotorEnablerInput Is 1 --
132
133             if (Enable = '1') then -- If Enable Is 1 --
134
135                 if (Clockwise = '1') then -- If Clockwise Is 1 --
136                     nextState <= S3; -- Change The State To The S3 --
137
138                 else -- If Clockwise Is 0 --
139                     nextState <= S1; -- Change The State To The S1 --
140
141                 end if;
142
143             else -- If Enable Is 0 --
144                 nextState <= S2; -- Hold The Current State --
145
146             end if;
147
148         else -- Is MotorEnablerInput Is 0 --
149             nextState <= S2; -- Hold The Current State --
150
151         end if;
152
153     when S3 => -- When State Is At S3 --
154         MotorOutput <= "0001"; -- Set Motor Output To '0001' --
155
156         if (MotorEnablerInput = '1') then -- Is MotorEnablerInput Is 1 --
157
158             if (Enable = '1') then -- If Enable Is 1 --
159
160                 if (Clockwise = '1') then -- If Clockwise Is 1 --
161                     nextState <= S0; -- Change The State To The S0 --
162
163                 else -- If Clockwise Is 0 --
164                     nextState <= S2; -- Change The State To The S2 --
165
166                 end if;
167
168             else -- If Enable Is 0 --
```

```
169             nextState <= S3; -- Hold The Current State --
170
171             end if;
172
173             else -- Is MotorEnablerInput Is 0 --
174                 nextState <= S3; -- Hold The Current State --
175
176             end if;
177
178         end case;
179
180     end process;
181
182 end Behavioral;
```