```
1   --------------------------------------------------------------------------------
2   -- Company:
3   -- Engineer:
4   --
5   -- Create Date:   22:05:53 05/14/2021
6   -- Design Name:
7   -- Module Name:   C:/Xilinx/14.7/VHDL_Design_Project/TB_MotorController.vhd
8   -- Project Name:  VHDL_Design_Project
9   -- Target Device:
10  -- Tool versions:
11  -- Description:
12  --
13  -- VHDL Test Bench Created by ISE for module: MotorController
14  --
15  -- Dependencies:
16  --
17  -- Revision:
18  -- Revision 0.01 - File Created
19  -- Additional Comments:
20  --
21  -- Notes:
22  -- This testbench has been automatically generated using types std_logic and
23  -- std_logic_vector for the ports of the unit under test.  Xilinx recommends
24  -- that these types always be used for the top-level I/O of a design in order
25  -- to guarantee that the testbench will bind correctly to the post-implementation
26  -- simulation model.
27  --------------------------------------------------------------------------------
28  LIBRARY ieee;
29  USE ieee.std_logic_1164.ALL;
30
31  -- Uncomment the following library declaration if using
32  -- arithmetic functions with Signed or Unsigned values
33  --USE ieee.numeric_std.ALL;
34
35  ENTITY TB_MotorController IS
36  END TB_MotorController;
37
38  ARCHITECTURE behavior OF TB_MotorController IS
39
40      -- Component Declaration for the Unit Under Test (UUT)
41
42      COMPONENT MotorController
43      PORT(
44           reset : IN  std_logic;
45           MotorPulseInput : IN  std_logic;
46           Enable : IN  std_logic; -- To Enable The Motor To Turn --
47           Clockwise : IN  std_logic; -- To Choose Direction To Turn The Stepper Motor --
48           MotorEnablerInput : IN std_logic; -- To Enable And Disable The Stepper Motor
    --
49           MotorOutput : OUT  std_logic_vector(3 downto 0) -- Output for Motor --
50          );
51      END COMPONENT;
52
53
54     --Inputs
55     signal reset : std_logic := '0';
56     signal MotorPulseInput : std_logic := '0';
```

```
57      signal Enable : std_logic := '0';
58      signal Clockwise : std_logic := '0';
59      signal MotorEnablerInput : std_logic := '0';
60
61      --Outputs
62      signal MotorOutput : std_logic_vector(3 downto 0);
63
64      -- Clock period definitions
65      constant MotorPulseInput_period : time := 10 ns;
66
67   BEGIN
68
69      -- Instantiate the Unit Under Test (UUT)
70      uut: MotorController PORT MAP (
71             reset => reset,
72             MotorPulseInput => MotorPulseInput,
73             Enable => Enable,
74             Clockwise => Clockwise,
75             MotorEnablerInput => MotorEnablerInput,
76             MotorOutput => MotorOutput
77           );
78
79      -- Clock process definitions
80      MotorPulseInput_process :process -- Follows The Motor Pulse --
81      begin
82         MotorPulseInput <= '0';
83         wait for MotorPulseInput_period/2;
84         MotorPulseInput <= '1';
85         wait for MotorPulseInput_period/2;
86      end process;
87
88
89      -- Stimulus process
90      stim_proc: process
91      begin
92         -- hold reset state for 100 ns.
93         reset <= '1';
94
95         wait for 100 ns;
96         reset <= '0';
97
98         -- insert stimulus here --
99
100        -- When MotorEnablerInput Is 0, Meaning Stepper Motor Will Not Be Enabled --
101        wait for MotorPulseInput_period*10;
102        MotorEnablerInput <= '0';
103        Enable <= '0';
104        Clockwise <= '0';
105
106        wait for MotorPulseInput_period*10;
107        MotorEnablerInput <= '0';
108        Enable <= '1';
109        Clockwise <= '0';
110
111        wait for MotorPulseInput_period*10;
112        MotorEnablerInput <= '0';
113        Enable <= '0';
```

```
114         Clockwise <= '1';
115 
116         wait for MotorPulseInput_period*10;
117         MotorEnablerInput <= '0';
118         Enable <= '1';
119         Clockwise <= '1';
120 
121         -- When MotorEnablerInput Is 1, Meaning Stepper Motor Will Be Enabled --
122         wait for MotorPulseInput_period*10;
123         MotorEnablerInput <= '1';
124         Enable <= '0';
125         Clockwise <= '0';
126 
127         wait for MotorPulseInput_period*10;
128         MotorEnablerInput <= '1';
129         Enable <= '1';
130         Clockwise <= '0';
131 
132         wait for MotorPulseInput_period*10;
133         MotorEnablerInput <= '1';
134         Enable <= '0';
135         Clockwise <= '1';
136 
137         wait for MotorPulseInput_period*10;
138         MotorEnablerInput <= '1';
139         Enable <= '1';
140         Clockwise <= '1';
141 
142         wait;
143      end process;
144 
145   END;
146 
```