File: C:/My_Designs/lab09/lab09/src/SPI_test_system_II_tb.vhd *

```
1    ------------------------------------------------------------------------
     -----
2    --
3    -- Title       : SPI_test_system_II_tb
4    -- Design      : SPI_test_system_II_tb
5    -- Author      : Ishabul Haque and Ken Ejinkonye
6    -- Company     : Stony Brook
7    --
8    ------------------------------------------------------------------------
     -----
9    --
10   -- File        : c:\My_Designs\Lab_09\SPI_Transmitter_and_Reciever\src\spi_tx_shifter_t
     d
11   -- Generated   : Sun May  3 10:51:17 2020
12   -- From        : interface description file
13   -- By          : Itf2Vhdl ver. 1.22
14   --
15   ------------------------------------------------------------------------
     -----
16   --
17   -- Description : Non-self checking testbench for spi_tx_shifter design
18   --
19   ------------------------------------------------------------------------
     -----
20
21
22   library ieee;
23   use ieee.std_logic_1164.all;
24   use ieee.numeric_std.all;
25   library work;
26   use work.all;
27
28
29   entity SPI_test_system_II_tb is
30   end SPI_test_system_II_tb;
31
32
33
34   architecture SPI_test_system_II_tb of SPI_test_system_II_tb is
35      --stimulus signals
36       signal rst_bar : std_logic;
37       signal clk : std_logic;
38       signal send : std_logic;        -- positive pulse to start
     transmission
39       signal send_en: std_logic;
40       signal miso : std_logic;        -- master in slave out
41       signal cpha: std_logic;
42       signal cpol: std_logic;
43       signal dord: std_logic;
44       signal data_in: std_logic_vector(7 downto 0);
45       signal spi_rxen: std_logic;
46       signal rxd : std_logic;
47
48       --observed signals
49       signal txd: std_logic;
50       signal sck: std_logic;
51       signal ss_bar: std_logic;
```

```
52      signal data_out :  std_logic_vector(7 downto 0);    -- parallel output
    data
53      signal mosi :  std_logic;       -- master out slave in SPI serial data
54
55
56      constant period : time := 40ns;
57      signal end_sim : boolean := false;
58      --constant data_in : std_logic_vector(7 downto 0) := x;
59      signal loopback : std_logic;
60
61
62  begin
63      -- Unit Under Test port map
64      UUT: entity SPI_test_system_II
65      port map (
66          rst_bar => rst_bar,
67          clk => clk,
68          --send_en => send_en,
69          send => send,
70          miso => miso,
71          cpha => cpha,
72          spi_rxen => spi_rxen,   -- signal to enable shift
73          cpol => cpol,
74          dord => dord,
75          data_in => data_in,
76          mosi => mosi,
77          sck => sck,
78          ss_bar => ss_bar
79          --spi_rxen => spi_rxen,
80          );
81
82          loopback <= mosi;
83          miso <= loopback;
84
85          -- Duration of signal send is 200ns at a high value
86          send_gen: process
87          begin
88              send <= '1';
89              for i in 0 to 28 loop      --28 rising edges
90                  wait for 200ns;        -- Send pulse duration of 200ns
91                  send <= not send;
92              end loop;                  --end pulse
93
94              std.env.finish;
95          end process;
96
97
98
99       -- Process to generate value for input data_in
100      data_gen:process
101      begin
102          for i in 1 to 2 loop
103              wait until clk='0';
104         data_in <= "11001010";
105         end loop;
106         wait;
107      end process;
```

```
108
109       -- Proocess to generate clock
110       clock_gen:process
111       begin
112           clk<='0';
113           loop
114               wait for period/4;
115               clk<=not clk;
116               exit when end_sim = true;
117           end loop;
118           wait;
119       end process;
120
121       -- Process to generate values for reset bar
122       reset: process
123       begin
124           -- Reset bar intially 0 for 60ns
125           rst_bar<='0';
126           for i in 1 to 2 loop
127               wait until clk = '1';
128           end loop;
129           rst_bar<='1';
130           wait;
131       end process;
132
133       -- Proocess to generate value for rxd, data
134           -- being shifted in
135           rxd_gen : process
136           begin
137               rxd <= '1';
138               loop
139                   wait for period/2;
140                   rxd <= '1';
141                   wait for period/2 ;
142                   rxd <= '0';
143                   wait for period/2;
144                   exit when end_sim = true;
145               end loop;
146           end process;
147
148           dord_gen: process
149           begin
150               loop
151               dord <= '0';
152               wait for 400 ns;
153               dord <= '1';
154               exit when end_sim = true;
155               end loop;
156           end process;
157
158
159
160
161
162       -- Process to generate values for dord,cpol,cpha. The push button
163       -- is represented by design send_pos_edge and cpol and cpha are
164       -- not changed in the middle of a transmission. Value of dord
```

```
165      -- determines which bit position the data is shifted to.
166      push_button:process
167      begin
168          -- Initialized values
169          send_en <= '0';
170          cpol <= '0';
171          cpha <= '0';
172          dord <= '0';
173
174          wait for 4*period;
175          for i in 0 to 9 loop
176              -- Generates the different modes of operation
177              (dord, cpol, cpha) <= to_unsigned(i,3);
178              wait for 2*period;
179              send_en <= '1';
180              wait for 20*period;
181              send_en <= '0';
182              wait for 2*period;
183
184          end loop;
185          dord <= '1';
186          end_sim <= true;
187          wait;
188      end process;
189
190  end SPI_test_system_II_tb;
191
```