File: C:/My_Designs/lab09/lab09/src/spi_rx_shifter.vhd

```
1   ----------------------------------------------------------------------------
    ----
2   --
3   -- Title       : spi_rx_shifter
4   -- Design      : task1_2
5   -- Author      : Ishabul Haque and Ken Ejinkonye
6   -- Company     : Stony Brook
7   --
8   -- Description : The receiver shifter spi_rx_shifter converts the serial
    data
9   -- at its rxd input to parallel and and provides this parallel result as
    output
10  -- data_out. This data may arrive most significant bit first or least
    significant
11  -- bit first, as determined by the dord input.The final parallel value at
    data_out
12  -- must always have its most significant bit in the leftmost position.
13
14  -- List of Circuit Features To Be Verified:
15  -- Inputs: rxd, rst_bar, clk,spi_rxen,dord
16
17  -- Outputs: data_out
18
19  ----------------------------------------------------------------------------
    ----
20
21  library ieee;
22  use ieee.std_logic_1164.all;
23  use ieee.numeric_std.all;
24
25
26  entity spi_rx_shifter is
27      port(
28      rxd : in std_logic;          -- data received from slave
29      rst_bar : in std_logic;      -- asynchronous reset
30      clk : in std_logic;          -- system clock
31      spi_rxen : in std_logic;     -- signal to enable shift
32      dord : in std_logic;         -- data order bit
33      data_out : out std_logic_vector(7 downto 0) -- received data
34      );
35  end spi_rx_shifter;
36
37
38  --}} End of automatically maintained section
39
40  architecture rx_shifter of spi_rx_shifter is
41  -- Signal g is a temporary vector to manipulate value going to data_out
42  signal g : std_logic_vector(7 downto 0):= "00000000";
43
44
45
46
47  begin
48
49      dord_change: process(dord, rxd, spi_rxen,g)
50      begin
51          -- Data can only be shifted when spi_rxen is asserted
```

```
52          if spi_rxen = '1' then
53              -- If dord = 0, data is shifted to most significant bit
54              if (dord='0') then
55                  -- Data is only shifted at the rising edge of clock
56                  if rising_edge(clk) then
57                      -- Data being shifted in is the value of rxd
58                      -- Initial bit position will always equal rxd
59                      data_out(0) <= rxd;
60                      g(0) <= rxd;
61                      -- For loop to implement shifting method using g
62                      -- as a test vector to manipulate data
63                      for i in 7 downto 0 loop
64                          if i>0 then
65                              g(i)<=g(i-1);
66                              data_out<=g;
67                          end if;
68
69
70                      end loop;
71                  end if ;
72              -- If dord = 1, data is shifted to least significant bit
73              elsif (dord='1') then
74                  if rising_edge(clk) then
75
76                      data_out(7)<=rxd;
77                      g(0) <= rxd;
78                      data_out <= g;
79
80                      for i in 7 downto 0 loop
81
82                          if i>0 then
83                              g(i-1)<=g(i);
84                              data_out<=g;
85                          end if;
86
87                      end loop;
88                  end if;
89
90
91              end if;
92          end if;
93
94      end process;
95
96  end rx_shifter;
97
98
```