File: C:/my_designs/register_128b/register_128b/src/reg128b_32_tb.vhd

```
1    ------------------------------------------------------------------------
     -----
2    --
3    -- Title       : reg128b_32_tb
4    -- Design      : register_128b
5    -- Author      : Aaron Lin and Hang Chen
6    -- Company     : Stony Brook University
7    --
8    ------------------------------------------------------------------------
     -----
9    --
10   -- File        :
     C:\my_designs\register_128b\register_128b\src\reg128b_32_tb.vhd
11   -- Generated   : Mon Apr 20 12:29:21 2020
12   -- From        : interface description file
13   -- By          : Itf2Vhdl ver. 1.22
14   --
15   ------------------------------------------------------------------------
     -----
16   --
17   -- Description : Testbench for the 32 128 bit registers. Tests if
     registers are written to when W_EN is low, if registers are written to
     when
18   --  W_EN is high, repeated again to see if multiple writes work, and if
     reads on the same register at the same time work.
19   --
20   ------------------------------------------------------------------------
     -----
21   library work;
22   use work.all;
23   library IEEE;
24   use IEEE.std_logic_1164.all;
25   library work;
26   use work.all;
27   use IEEE.NUMERIC_STD.all;
28
29   entity reg128b_32_tb is
30   end reg128b_32_tb;
31
32   --}} End of automatically maintained section
33
34   architecture reg128b_32_tb of reg128b_32_tb is
35       signal CLK, W_EN : STD_LOGIC;
36       signal I_regA, I_regB, I_regC, I_regD : STD_LOGIC_VECTOR(4 downto 0);
37       signal I_DataIn, O_regA, O_regB, O_regC : STD_LOGIC_VECTOR(127 downto
     0);
38       constant period : time := 20ns;
39   begin
40
41       uut : entity reg128b_32
42           port map (CLK => CLK, W_EN => W_EN,
43           I_regA => I_regA, I_regB => I_regB, I_regC => I_regC, I_regD =>
     I_regD, I_DataIn => I_DataIn, O_regA => O_regA, O_regB => O_regB, O_regC
     => O_regC);
44
45
46       clk_process : process
```

```
47        begin
48            CLK <= '0';
49            wait for period/2;
50            CLK <= '1';
51            wait for period/2;
52        end process;
53
54        tb : process
55        begin
56            wait for period/2;
57
58
59            --Checks if registers are written to even though W_EN is low
60            W_EN <= '0';
61            I_regA <= std_logic_vector(to_unsigned(0, I_regA'length));
62            I_regB <= std_logic_vector(to_unsigned(1, I_regB'length));
63            I_regC <= std_logic_vector(to_unsigned(2, I_regC'length));
64            I_regD <= std_logic_vector(to_unsigned(0, I_regD'length));
65            I_DataIn <= std_logic_vector(to_unsigned(100, I_DataIn'length));
66            wait for period;
67
68            I_regD <= std_logic_vector(to_unsigned(1, I_regD'length));
69            wait for period;
70
71            I_regD <= std_logic_vector(to_unsigned(2, I_regD'length));
72            wait for period;
73
74            --Checks if registerA is written to when W_EN is high
75            W_EN <= '1';
76            I_regD <= std_logic_vector(to_unsigned(0, I_regD'length));
77            wait for period;
78
79            --Checks if registerB is written to when W_EN is high
80            I_regD <= std_logic_vector(to_unsigned(1, I_regD'length));
81            wait for period;
82
83            --Checks if registerC is written to when W_EN is high
84            I_regD <= std_logic_vector(to_unsigned(2, I_regD'length));
85            wait for period;
86
87            --Checks if registers are written again even when W_EN is low
88            W_EN <= '0';
89            I_regD <= std_logic_vector(to_unsigned(0, I_regD'length));
90            I_DataIn <= std_logic_vector(to_unsigned(1, I_DataIn'length));
91            wait for period;
92
93            I_regD <= std_logic_vector(to_unsigned(1, I_regD'length));
94            I_DataIn <= std_logic_vector(to_unsigned(2, I_DataIn'length));
95            wait for period;
96
97            I_regD <= std_logic_vector(to_unsigned(2, I_regD'length));
98            I_DataIn <= std_logic_vector(to_unsigned(3, I_DataIn'length));
99            wait for period;
100
101           --Checks if registers are written to again when W_EN is high
102           W_EN <= '1';
103           I_DataIn <= std_logic_vector(to_unsigned(1, I_DataIn'length));
```

```
104          wait for period;
105
106          I_regD <= std_logic_vector(to_unsigned(1, I_regD'length));
107          I_DataIn <= std_logic_vector(to_unsigned(3, I_DataIn'length));
108          wait for period;
109
110          I_regD <= std_logic_vector(to_unsigned(0, I_regD'length));
111          I_DataIn <= std_logic_vector(to_unsigned(2, I_DataIn'length));
112          wait for period;
113
114          I_regD <= std_logic_vector(to_unsigned(100, I_regD'length));
115          I_DataIn <= std_logic_vector(to_unsigned(16, I_DataIn'length));
116          wait for period;
117
118          --Read the same registers at the same time
119          I_regA <= std_logic_vector(to_unsigned(100, I_regA'length));
120          I_regB <= std_logic_vector(to_unsigned(100, I_regB'length));
121          I_regC <= std_logic_vector(to_unsigned(100, I_regC'length));
122
123          wait;
124          end process;
125  end reg128b_32_tb;
126
```