fifo.test_top.vhd

File location:

$VCS_HOME/doc/examples/nativetestbench/systemverilog/vcs_quickstart/fifo.test_top.vhd

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
library STD;
use STD.textio.all;
use IEEE.std_logic_textio.all;
library WORK;
use WORK.all;
entity TB is
end;
architecture TEST of TB is
    signal TOP_CLOCK,rst_n,push_req_n,pop_req_n,diag_n,empty,almost_empty : std_logic;
    signal almost_full,half_full,full,error,pop_req,push_req : std_logic;
    signal data_out,data_in : std_logic_vector(15 downto 0);

  component DW_fifo_s1_sf
        generic (
                    width : INTEGER range 1 to 256 := 8;
                    depth : INTEGER range 2 to 256 := 4;
                    ae_level : INTEGER range 1 to 255 := 1;
                    af_level : INTEGER range 1 to 255 := 1;
                    err_mode : INTEGER range 0 to 2 := 0;
                    rst_mode : INTEGER range 0 to 3 := 0
                );
        port    (
                    clk : in std_logic;
                    rst_n : in std_logic;
                    push_req_n : in std_logic;
                    pop_req_n : in std_logic;
                    diag_n : in std_logic;
                    data_in : in std_logic_vector( width-1 downto 0 );
                    empty : out std_logic;
                    almost_empty : out std_logic;
                    half_full : out std_logic;
```

```
                  almost_full : out std_logic;
                  full : out std_logic;
                  error : out std_logic;
                  data_out : out std_logic_vector( width-1 downto 0 )
              );
end component;

component fifo_test
      port    (
                  clk : in std_logic;
                  rst_n : out std_logic;
                  push_req_n : out std_logic;
                  pop_req_n : out std_logic;
                  diag_n : out std_logic;
                  data_in : out std_logic_vector( 15 downto 0 );
                  empty : in std_logic;
                  almost_empty : in std_logic;
                  half_full : in std_logic;
                  almost_full : in std_logic;
                  full : in std_logic;
                  error : in std_logic;
                  data_out : in std_logic_vector( 15 downto 0 )
              );
end component;
component assert_fifo
    generic (severity_level : integer := 0;
             depth          : integer := 2;
             elem_sz        : integer := 1;
             hi_water_mark  : integer := 0;
             enq_lat        : integer := 0;
             deq_lat        : integer := 0;
             oflow_chk      : integer := 1;
             uflow_chk      : integer := 1;
             value_chk      : integer := 1;
             pass_thru      : integer := 0;
             edge_expr      : integer := 0;
             msg            : string := "VIOLATION";
             category       : integer := 0;
             coverage_level_1 : integer := 15;
             coverage_level_2 : integer := 0;
             coverage_level_3 : integer := 0);
    port (clk, reset_n, enq : in std_ulogic;
            enq_data : in unsigned ((elem_sz-1) downto 0);
            deq : in std_ulogic;
            deq_data : in unsigned ((elem_sz-1) downto 0));
  end component;


begin
  process( TOP_CLOCK)
  variable TCLK : std_logic := '0';
    begin
    TOP_CLOCK <= TCLK after 50 ns;
    TCLK := not TCLK;

  end process;
  process(push_req_n)
    begin
    push_req <= not push_req_n;
    end process ;
  process(pop_req_n)
    begin
    pop_req <= not pop_req_n;
    end process ;
  dut : DW_fifo_s1_sf
    generic map (width => 16,   depth => 128,
                 err_mode => 0,   rst_mode => 1 , ae_level => 3,
                 af_level => 120)
    port map (clk => TOP_CLOCK,   rst_n => rst_n,
              push_req_n => push_req_n,   pop_req_n => pop_req_n,
              diag_n => diag_n,
              data_in => data_in,
              empty => empty,   almost_empty => almost_empty,
              half_full => half_full,   almost_full => almost_full,
              full => full,   error => error,
              data_out => data_out);
 test : fifo_test
    port map (clk => TOP_CLOCK,   rst_n => rst_n,
              push_req_n => push_req_n,   pop_req_n => pop_req_n,
              diag_n => diag_n,
              data_in => data_in,
              empty => empty,   almost_empty => almost_empty,
              half_full => half_full,   almost_full => almost_full,
              full => full,   error => error,
              data_out => data_out);
  SVA_FIFO_inst   : assert_fifo
      generic map (depth => 128, elem_sz => 16,
                   coverage_level_1 => 31,
                   coverage_level_2 => 0,
                   coverage_level_3 => 31)
     port map(clk => TOP_CLOCK, reset_n => rst_n, enq => push_req,
              enq_data => data_in , deq => pop_req , deq_data => data_out);
end TEST;
configuration CFG_TG of TB is
      for TEST
      end for;
end;
```