```
////////////////////////////////////////////////////////////////////////
//                                                                    //
//  File name : testbench.sv                                          //
//  Author    : G. Andres Mancera                                     //
//  License   : GNU Lesser General Public License                     //
//  Course    : Advanced Verification with SystemVerilog OOP          //
//              Testbench - UCSC Silicon Valley Extension             //
//                                                                    //
////////////////////////////////////////////////////////////////////////

module testbench();

  logic         clk_156m25, clk_xgmii_rx, clk_xgmii_tx;
  logic         reset_156m25_n, reset_xgmii_rx_n, reset_xgmii_tx_n;
  logic         pkt_rx_ren, pkt_tx_eop, pkt_tx_sop, pkt_tx_val;
  logic         wb_clk_i, wb_cyc_i, wb_rst_i, wb_stb_i, wb_we_i;
  logic [63:0]  pkt_tx_data, xgmii_rxd;
  logic [2:0]   pkt_tx_mod;
  logic [7:0]   wb_adr_i, xgmii_rxc;
  logic [31:0]  wb_dat_i;
  logic         pkt_rx_avail, pkt_rx_eop, pkt_rx_err, pkt_rx_sop, pkt_rx_val,
pkt_tx_full;
  logic         wb_ack_o, wb_int_o;
  logic [63:0]  pkt_rx_data, xgmii_txd;
  logic [2:0]   pkt_rx_mod;
  logic [31:0]  wb_dat_o;
  logic [7:0]   xgmii_txc;

  //-----------------------------------------------------------------
  // In order to enable waveform dumping, either uncomment the system
  // call below or use the +vcs+vcdpluson vcs command line option.
  //initial begin
  //  $vcdpluson;     // Enable waveform dumping
  //end

  // Generate free running clocks
  initial begin
    clk_156m25      = 1'b0;
    clk_xgmii_rx    = 1'b0;
    clk_xgmii_tx    = 1'b0;
    wb_clk_i        = 1'b0;
    forever begin
      #3200;
      clk_156m25    = ~clk_156m25;
      clk_xgmii_rx  = ~clk_xgmii_rx;
      clk_xgmii_tx  = ~clk_xgmii_tx;
      wb_clk_i      = ~wb_clk_i;
    end
  end

  // Generate the reset signals
  initial begin
    reset_156m25_n      = 1'b0;
    reset_xgmii_rx_n    = 1'b0;
    reset_xgmii_tx_n    = 1'b0;
    wb_rst_i            = 1'b1;
    #20000;
    reset_156m25_n      <= 1'b1;
    reset_xgmii_rx_n    <= 1'b1;
    reset_xgmii_tx_n    <= 1'b1;
```

```
60      wb_rst_i            <= 1'b0;
61    end
62
63
64    // xge_mac_interface intantiated here
65    xge_mac_interface     xge_mac_if  (
66                                          .clk_156m25         (clk_156m25),
67                                          .clk_xgmii_rx       (clk_xgmii_rx),
68                                          .clk_xgmii_tx       (clk_xgmii_tx),
69                                          .wb_clk_i           (wb_clk_i),
70                                          .reset_156m25_n     (reset_156m25_n),
71                                          .reset_xgmii_rx_n   (reset_xgmii_rx_n),
72                                          .reset_xgmii_tx_n   (reset_xgmii_tx_n),
73                                          .wb_rst_i           (wb_rst_i)
74                                       );
75
76    // DUT instantiated here
77    xge_mac    mac_core_dut  ( // Outputs
78                               .pkt_rx_avail       (xge_mac_if.pkt_rx_avail),
79                               .pkt_rx_data        (xge_mac_if.pkt_rx_data),
80                               .pkt_rx_eop         (xge_mac_if.pkt_rx_eop),
81                               .pkt_rx_err         (xge_mac_if.pkt_rx_err),
82                               .pkt_rx_mod         (xge_mac_if.pkt_rx_mod),
83                               .pkt_rx_sop         (xge_mac_if.pkt_rx_sop),
84                               .pkt_rx_val         (xge_mac_if.pkt_rx_val),
85                               .pkt_tx_full        (xge_mac_if.pkt_tx_full),
86                               .wb_ack_o           (xge_mac_if.wb_ack_o),
87                               .wb_dat_o           (xge_mac_if.wb_dat_o),
88                               .wb_int_o           (xge_mac_if.wb_int_o),
89                               .xgmii_txc          (xge_mac_if.xgmii_txc),
90                               .xgmii_txd          (xge_mac_if.xgmii_txd),
91                               // Inputs
92                               .clk_156m25         (clk_156m25),
93                               .clk_xgmii_rx       (clk_xgmii_rx),
94                               .clk_xgmii_tx       (clk_xgmii_tx),
95                               .pkt_rx_ren         (xge_mac_if.pkt_rx_ren),
96                               .pkt_tx_data        (xge_mac_if.pkt_tx_data),
97                               .pkt_tx_eop         (xge_mac_if.pkt_tx_eop),
98                               .pkt_tx_mod         (xge_mac_if.pkt_tx_mod),
99                               .pkt_tx_sop         (xge_mac_if.pkt_tx_sop),
100                              .pkt_tx_val         (xge_mac_if.pkt_tx_val),
101                              .reset_156m25_n     (reset_156m25_n),
102                              .reset_xgmii_rx_n   (reset_xgmii_rx_n),
103                              .reset_xgmii_tx_n   (reset_xgmii_tx_n),
104                              .wb_adr_i           (xge_mac_if.wb_adr_i),
105                              .wb_clk_i           (wb_clk_i),
106                              .wb_cyc_i           (xge_mac_if.wb_cyc_i),
107                              .wb_dat_i           (xge_mac_if.wb_dat_i),
108                              .wb_rst_i           (wb_rst_i),
109                              .wb_stb_i           (xge_mac_if.wb_stb_i),
110                              .wb_we_i            (xge_mac_if.wb_we_i),
111                              .xgmii_rxc          (xge_mac_if.xgmii_rxc),
112                              .xgmii_rxd          (xge_mac_if.xgmii_rxd)
113                            );
114
115   // Testcase instantiated here
116   testcase  itestcase (     xge_mac_if.testcase_port,
117                             xge_mac_if.testcase_port    );
118
119 endmodule
```

120