```
`timescale 1ns/100ps
`define AssertionFlag 1'b 1
`define esterelSTRING [10*8:1]
`define esterelINTEGER [31:0]
`define esterelBOOLEAN
`define esterelLOGIC
`include "ABRO_data_type_pkg.v"

module ABRO_abro_tb;
    reg stop;
    reg `esterelSTRING Assertion;
    reg [31:0] temporaryOutputFile;
    parameter Delay = 10;


    `include "ABRO_data_pkg.v"
    `include "ABRO_data_sim_pkg.v"

    reg `esterelLOGIC  sig_clk;
    reg `esterelLOGIC  sig_rst;
    reg `esterelLOGIC  sig_A;
    reg `esterelSTRING  sig_Adata;
    reg `esterelLOGIC  sig_B;
    reg `data  sig_Bdata;
    reg `esterelLOGIC  sig_R;
    wire `esterelLOGIC  sig_O;
    wire `data  sig_Odata;

    task WriteLogOutputs;
        input `esterelLOGIC sig_O;
        input `data sig_Odata;

        begin
            if (sig_O) begin
                $fdisplay( temporaryOutputFile, "%% Output: =\"%s\"", data_to_text(sig_
Odata));
            end
        end
    endtask


    initial sig_clk = 1'b 0;
    always #(Delay / 2) begin
        if (stop) begin
            sig_clk <= 1'b 0;
            disable SCENARIO;
        end else begin
            sig_clk <= ~sig_clk;
        end
    end

    ABRO COMP (
        .clk(sig_clk),
        .rst(sig_rst),
        .A(sig_A),
        .Adata(sig_Adata),
        .B(sig_B),
        .Bdata(sig_Bdata),
        .R(sig_R),
```

```
        .O(sig_O),
        .Odata(sig_Odata)

    );

    initial
        begin : SCENARIO
            temporaryOutputFile = $fopen("record_abro.eso");

            stop = 1'b 0;
            Assertion = "NOTE";


            // -----
            $fdisplay(temporaryOutputFile, "%% -----");
            // File record_abro.eso generated from Esterel module ABRO.
            $fdisplay(temporaryOutputFile, "%% File record_abro.eso generated from Esterel module
ABRO.");
            // -----
            $fdisplay(temporaryOutputFile, "%% -----");
// RST //
            sig_rst <= 1'b 1;
            @(posedge sig_clk);
            $fdisplay( temporaryOutputFile, "!reset;");

            sig_A <= 1'b 1;
            sig_Adata <= "1";
            $fdisplay( temporaryOutputFile, "A =\"1\"");
            sig_rst <= 1'b 0;
            sig_B <= 1'b 0;
            sig_R <= 1'b 0;
            // Sync on CLK rising edge //
            @(posedge sig_clk);
            $fdisplay( temporaryOutputFile, "%% Cycle 1");
            // Log outputs to file //
            WriteLogOutputs( sig_O, sig_Odata);
            $fdisplay( temporaryOutputFile, ";");
            // Outputs checking //


//////////////////////////////////
// CLK cycle number: 2 //
//////////////////////////////////
            // Inputs initialization //
            sig_B <= 1'b 1;
            if (!check_data("premier tirage")) begin
                stop <= 1'b 1;
                $fdisplay( temporaryOutputFile, "bad user type value checked, in file abro.esi, line 2"
);
                $fclose(temporaryOutputFile);
                Assertion <= "FAILURE";
                $stop;
            end else begin
                sig_Bdata <= text_to_data("premier tirage");
            end
            $fdisplay( temporaryOutputFile, "B =\"premier tirage\"");
            sig_rst <= 1'b 0;
            sig_A <= 1'b 0;
            sig_R <= 1'b 0;
```

```
            // Sync on CLK rising edge //
            @(posedge sig_clk);
            $fdisplay( temporaryOutputFile, "%% Cycle 2");
            // Log outputs to file //
            WriteLogOutputs( sig_O, sig_Odata);
            $fdisplay( temporaryOutputFile, ";");
            // Outputs checking //


//////////////////////////////////
// CLK cycle number: 3 //
//////////////////////////////////
            // Inputs initialization //
            sig_A <= 1'b 1;
            sig_Adata <= "2";
            $fdisplay( temporaryOutputFile, "A =\"2\"");
            sig_rst <= 1'b 0;
            sig_B <= 1'b 0;
            sig_R <= 1'b 0;
            // Sync on CLK rising edge //
            @(posedge sig_clk);
            $fdisplay( temporaryOutputFile, "%% Cycle 3");
            // Log outputs to file //
            WriteLogOutputs( sig_O, sig_Odata);
            $fdisplay( temporaryOutputFile, ";");
            // Outputs checking //


//////////////////////////////////
// CLK cycle number: 4 //
//////////////////////////////////
            // Inputs initialization //
            sig_B <= 1'b 1;
            if (!check_data("second tirage")) begin
                stop <= 1'b 1;
                $fdisplay( temporaryOutputFile,  "bad user type value checked, in file abro.esi, line 8"
);
                $fclose(temporaryOutputFile);
                Assertion <= "FAILURE";
                $stop;
            end else begin
                sig_Bdata <= text_to_data("second tirage");
            end
            $fdisplay( temporaryOutputFile, "B =\"second tirage\"");
            sig_rst <= 1'b 0;
            sig_A <= 1'b 0;
            sig_R <= 1'b 0;
            // Sync on CLK rising edge //
            @(posedge sig_clk);
            $fdisplay( temporaryOutputFile, "%% Cycle 4");
            // Log outputs to file //
            WriteLogOutputs( sig_O, sig_Odata);
            $fdisplay( temporaryOutputFile, ";");
            // Outputs checking //


//////////////////////////////////
// CLK cycle number: 5 //
//////////////////////////////////
```

```
            // Inputs initialization //
            if (`AssertionFlag) begin
                if (!(((sig_Adata == "1") & sig_B == 1'b 1))) begin
                    $fdisplay(temporaryOutputFile, "%% %%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%\n%% NOTE: Break point reached\n
%%  file 'abro.esi'\n%%  line '8'\n%% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%\n" );
                    Assertion = "NOTE";
                end
            end
            sig_R <= 1'b 1;
            $fdisplay( temporaryOutputFile, "R");
            sig_rst <= 1'b 0;
            sig_A <= 1'b 0;
            sig_B <= 1'b 0;
            // Sync on CLK rising edge //
            @(posedge sig_clk);
            $fdisplay( temporaryOutputFile, "%% Cycle 5");
            // Log outputs to file //
            WriteLogOutputs( sig_O, sig_Odata);
            $fdisplay( temporaryOutputFile, ";");
            // Outputs checking //


//////////////////////////////////
// CLK cycle number: 6 //
//////////////////////////////////
            // Inputs initialization //
            // A (\"3\");
            $fdisplay(temporaryOutputFile, "%% A (\"3\");");
            // B (\"troisieme tirage\");
            $fdisplay(temporaryOutputFile, "%% B (\"troisieme tirage\");");

            stop = 1'b 1;
            $fclose(temporaryOutputFile);
            $stop;
        end
endmodule
```