Synchronous FIFO(First-In-First-Out) Memory in System Verilog

Mark Amgad Saleh

Content
1. **Design module with Assertions**
2. **Verification Plan**
3. **Top & Interface modules**
4. **Packges**
5. **Monitor & testbench modules**
6. **Source Files List & Do_File**
7. **Questa_Sim Snippets & Reports**

Design module with Assertions:

```
module FIFO(FIFO_interface.DUT fifo_if);

    logic [fifo_if.FIFO_WIDTH-1:0] data_in;
    logic clk, rst_n, wr_en, rd_en;
    logic [fifo_if.FIFO_WIDTH-1:0] data_out;
    logic wr_ack, overflow, full, empty, almostfull, almostempty, underflow;

    assign clk         = fifo_if.clk;
    assign rst_n       = fifo_if.rst_n;
    assign wr_en       = fifo_if.wr_en;
    assign rd_en       = fifo_if.rd_en;
    assign data_in     = fifo_if.data_in;
    assign data_out    = fifo_if.data_out;
    assign wr_ack      = fifo_if.wr_ack;
    assign overflow    = fifo_if.overflow;
    assign full        = fifo_if.full;
    assign empty       = fifo_if.empty;
    assign almostfull  = fifo_if.almostfull;
    assign almostempty = fifo_if.almostempty;
    assign underflow   = fifo_if.underflow;

    reg [fifo_if.FIFO_WIDTH-1    :0] mem [fifo_if.FIFO_DEPTH-1:0];
    reg [fifo_if.max_fifo_addr-1 :0] wr_ptr, rd_ptr;
    reg [fifo_if.max_fifo_addr   :0] count;

    always @(posedge fifo_if.clk or negedge fifo_if.rst_n) begin //////write operation////////
        if (!fifo_if.rst_n) begin
            wr_ptr          <= 0;
            fifo_if.wr_ack  <= 0;
            fifo_if.overflow <= 0;
        end

        else if (fifo_if.wr_en && count < fifo_if.FIFO_DEPTH) begin
            mem[wr_ptr]     <= fifo_if.data_in;
            fifo_if.wr_ack  <= 1;
            wr_ptr          <= wr_ptr + 1;
            fifo_if.overflow <= 0;
        end

        else begin
            fifo_if.wr_ack <= 0;

            if (fifo_if.wr_en && fifo_if.full)
                fifo_if.overflow <= 1;
            else
                fifo_if.overflow <= 0;
            end
    end

    always @(posedge fifo_if.clk or negedge fifo_if.rst_n) begin ///////read operation///////
        if (!fifo_if.rst_n) begin
            rd_ptr           <= 0;
            fifo_if.underflow <= 0;
            //fifo_if.data_out  <= 0;
        end
```

```
always @(posedge fifo_if.clk or negedge fifo_if.rst_n) begin ///////read operation///////
    if (!fifo_if.rst_n) begin
        rd_ptr            <= 0;
        fifo_if.underflow <= 0;
        //fifo_if.data_out  <= 0;
    end

    else if (fifo_if.rd_en && count != 0) begin
        fifo_if.data_out  <= mem[rd_ptr];
        rd_ptr            <= rd_ptr + 1;
        fifo_if.underflow <= 0;
    end

    else begin

        if (fifo_if.rd_en && fifo_if.empty)
            fifo_if.underflow <= 1;
        else
            fifo_if.underflow <= 0;

    end
end

always @(posedge fifo_if.clk or negedge fifo_if.rst_n) begin
    if (!fifo_if.rst_n) begin
        count <= 0;
    end
    else begin
        if      ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b10) && !fifo_if.full)
                count <= count + 1;

        else if ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b01) && !fifo_if.empty)
                count <= count - 1;

        else if   ({fifo_if.wr_en, fifo_if.rd_en} == 2'b11) begin

            if (fifo_if.full)
                count <= count - 1;
            else if (fifo_if.empty)
                count <= count + 1;
            //else
                //count <= count;

        end
    end
end

assign fifo_if.full        = (count == fifo_if.FIFO_DEPTH)   ? 1 : 0;
assign fifo_if.empty       = (count == 0)                    ? 1 : 0;
assign fifo_if.almostfull  = (count == fifo_if.FIFO_DEPTH-1) ? 1 : 0;
assign fifo_if.almostempty = (count == 1)                    ? 1 : 0;



`ifdef SIM
```

```
    `ifdef SIM

    //assertions and cover of sequential outputs
    overflow_assert_1    :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count==fifo_if.FIFO_DEPTH)&&fifo_if.wr_en)   |=>(fifo_if.overflow));
    underflow_assert_1   :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count==0)&&fifo_if.rd_en)                  |=>(fifo_if.underflow));
    wr_ack_assert_1      :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count!=fifo_if.FIFO_DEPTH)&&fifo_if.wr_en)   |=>(fifo_if.wr_ack));
    overflow_cover_1     :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count==fifo_if.FIFO_DEPTH)&&fifo_if.wr_en)   |=>(fifo_if.overflow));
    underflow_cover_1    :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count==0)&&fifo_if.rd_en)                  |=>(fifo_if.underflow));
    wr_ack_cover_1       :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count!=fifo_if.FIFO_DEPTH)&&fifo_if.wr_en)   |=>(fifo_if.wr_ack));

    overflow_assert_2    :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count!=fifo_if.FIFO_DEPTH) && fifo_if.wr_en) |=>(fifo_if.overflow   == 0));
    underflow_assert_2   :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count!=0) && fifo_if.rd_en)                 |=>(fifo_if.underflow) == 0);
    wr_ack_assert_2      :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count==fifo_if.FIFO_DEPTH)&&fifo_if.wr_en)   |=>(fifo_if.wr_ack)    == 0);
    overflow_cover_2     :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count!=fifo_if.FIFO_DEPTH)&&fifo_if.wr_en)   |=>(fifo_if.overflow)  == 0);
    underflow_cover_2    :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count!=0) && fifo_if.rd_en)                 |=>(fifo_if.underflow  == 0));
    wr_ack_cover_2       :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count==fifo_if.FIFO_DEPTH)&&fifo_if.wr_en)   |=>(fifo_if.wr_ack)    == 0);

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
    //assertions and cover of combinational outputs
    full_assert_1        :assert property (@(posedge fifo_if.clk)  (count==fifo_if.FIFO_DEPTH)   |-> fifo_if.full);
    empty_assert_1       :assert property (@(posedge fifo_if.clk)  (count==0)                    |-> fifo_if.empty);
    almostfull_assert_1  :assert property (@(posedge fifo_if.clk)  (count==fifo_if.FIFO_DEPTH-1) |-> fifo_if.almostfull);
    almostempty_assert_1 :assert property (@(posedge fifo_if.clk)  (count==1)                    |-> fifo_if.almostempty);
    full_cover_1         :cover  property (@(posedge fifo_if.clk)  (count==fifo_if.FIFO_DEPTH)   |-> fifo_if.full);
    empty_cover_1        :cover  property (@(posedge fifo_if.clk)  (count==0)                    |-> fifo_if.empty);
    almostfull_cover_1   :cover  property (@(posedge fifo_if.clk)  (count==fifo_if.FIFO_DEPTH-1) |-> fifo_if.almostfull);
    almostempty_cover_1  :cover  property (@(posedge fifo_if.clk)  (count==1)                    |-> fifo_if.almostempty);

    full_assert_2        :assert property (@(posedge fifo_if.clk)  (count!=fifo_if.FIFO_DEPTH)   |-> fifo_if.full        == 0);
    empty_assert_2       :assert property (@(posedge fifo_if.clk)  (count!=0)                    |-> fifo_if.empty       == 0);
    almostfull_assert_2  :assert property (@(posedge fifo_if.clk)  (count!=fifo_if.FIFO_DEPTH-1) |-> fifo_if.almostfull  == 0);
    almostempty_assert_2 :assert property (@(posedge fifo_if.clk)  (count!=1)                    |-> fifo_if.almostempty == 0);
    full_cover_2         :cover  property (@(posedge fifo_if.clk)  (count!=fifo_if.FIFO_DEPTH)   |-> fifo_if.full        == 0);
    empty_cover_2        :cover  property (@(posedge fifo_if.clk)  (count!=0)                    |-> fifo_if.empty       == 0);
    almostfull_cover_2   :cover  property (@(posedge fifo_if.clk)  (count!=fifo_if.FIFO_DEPTH-1) |-> fifo_if.almostfull  == 0);
    almostempty_cover_2  :cover  property (@(posedge fifo_if.clk)  (count!=1)                    |-> fifo_if.almostempty == 0);

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

    //assertions on counter
    rd_count_assert     :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  && !fifo_if.wr_en && count !=0)                          |=>($past(count)-1'b1 == count));
    wr_count_assert     :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (!fifo_if.rd_en && fifo_if.wr_en  && count !=fifo_if.FIFO_DEPTH)          |=>($past(count)+1'b1 == count));
    rd_wr_count_assert :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  && fifo_if.wr_en  && count !=0 && count!=fifo_if.FIFO_DEPTH) |=>($past(count)      == count));
    rd_count_cover      :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  &&!fifo_if.wr_en  && count !=0)                          |=>($past(count)-1'b1 == count));
    wr_count_cover      :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (!fifo_if.rd_en &&fifo_if.wr_en   && count !=fifo_if.FIFO_DEPTH)          |=>($past(count)+1'b1 == count));
    rd_wr_count_cover  :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  &&fifo_if.wr_en   &&count  !=0 && count!=fifo_if.FIFO_DEPTH) |=>($past(count)      == count));

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

    //assertions on pointers
    rd_ptr_assert       :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  && count!=0)                 |=>($past(rd_ptr)+1'b1==rd_ptr));
    wr_ptr_assert       :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.wr_en  && count!=fifo_if.FIFO_DEPTH)  |=>($past(wr_ptr)+1'b1==wr_ptr));
    rd_ptr_cover        :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  && count!=0)                 |=>($past(rd_ptr)+1'b1==rd_ptr));
    wr_ptr_cover        :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.wr_en  && count!=fifo_if.FIFO_DEPTH)  |=>($past(wr_ptr)+1'b1==wr_ptr));
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

    //more assertions
    // if almostfull is HIGH and wr_en only is HIGH then the next cycle full will be HIGH
    almostfull_full_assert   :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.almostfull  && fifo_if.wr_en && !fifo_if.rd_en) |=> fifo_if.full);

    // if almostempty is HIGH and rd_en only is HIGH then the next cycle empty will be HIGH
    almostempty_empty_assert :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.almostempty && fifo_if.rd_en && !fifo_if.wr_en) |=> fifo_if.empty);

    almostfull_full_cover   :cover property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.almostfull  && fifo_if.wr_en && !fifo_if.rd_en) |=> fifo_if.full);
    almostempty_empty_cover :cover property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.almostempty && fifo_if.rd_en && !fifo_if.wr_en) |=> fifo_if.empty);


    always_comb begin
        if(!rst_n)begin
            rst_count_assert       :assert final (count       == 0);
            rst_rd_ptr_assert      :assert final (rd_ptr      == 0);
            rst_wr_ptr_assert      :assert final (wr_ptr      == 0);
            rst_full_assert        :assert final (full        == 0);
            rst_empty_assert       :assert final (empty       == 1);
            rst_almostfull_assert  :assert final (almostfull  == 0);
            rst_almostempty_assert :assert final (almostempty == 0);
            //rst_data_out_assert    :assert final (data_out    == 0);
            rst_overflow_assert    :assert final (overflow    == 0);
            rst_underflow_assert   :assert final (underflow   == 0);
            rst_wr_ack_assert      :assert final (wr_ack      == 0);
            rst_count_cover        :cover final  (count       == 0);
            rst_rd_ptr_cover       :cover final  (rd_ptr      == 0);
            rst_wr_ptr_cover       :cover final  (wr_ptr      == 0);
            rst_full_cover         :cover final  (full        == 0);
            rst_empty_cover        :cover final  (empty       == 1);
            rst_almostfull_cover   :cover final  (almostfull  == 0);
            rst_almostempty_cover  :cover final  (almostempty == 0);
            //rst_data_out_cover     :cover final  (data_out    == 0);
            rst_overflow_cover     :cover final  (overflow    == 0);
            rst_underflow_cover    :cover final  (underflow   == 0);
            rst_wr_ack_cover       :cover final  (wr_ack      == 0);
        end
    end
`endif


endmodule
```

Verification Plan:

Label	Description	Stimulus Generation	Functional Coverage	Functionality Check
FIFO_1	Empty should not be HIGH if write enable is HIGH	Randomization on wr_en under constraint that write occurs more than read	Included as cross cover for wr_en and empty	Output checked with assertion
FIFO_2	Full should not be HIGH if read enable is HIGH	Randomization on rd_en under constraint that write occurs more than read	Included as cross cover for rd_en and full	Output checked with assertion
FIFO_3	Overflow should not be HIGH if write enable is LOW	Randomization on wr_en under constraint that write occurs more than read	Included as cross cover for wr_en and overflow	Output checked with assertion
FIFO_4	Underflow should not be HIGH if read enable is LOW	Randomization on rd_en under constraint that write occurs more than read	Included as cross cover for rd_en and underflow	Output checked with assertion
FIFO_5	Write ack should not be HIGH if write enable is LOW	Randomization on wr_en under constraint that write occurs more than read	Included as cross cover for wr_en and wr_ack	Output checked with assertion
FIFO_6	In case of both read enable and write enable are HIGH and the FIFO is full, then only read occurs	Randomization on wr_en and rd_en under constraint that write occurs more than read	Included as cross cover for wr_en and rd_en and all the output flags	Output checked with assertion

FIFO_6	In case of both read enable and write enable are HIGH and the FIFO is full, then only read occurs	Randomization on wr_en and rd_en under constraint that write occurs more than read	Included as cross cover for wr_en and rd_en and all the output flags	Output checked with assertion
FIFO_7	In case of both read enable and write enable are HIGH and the FIFO is empty, then only write occurs	Randomization on wr_en and rd_en under constraint that write occurs more than read	Included as cross cover for wr_en and rd_en and all the output flags	Output checked with assertion
FIFO_8	In case of both read enable and write enable are HIGH and the FIFO is not full or empty, then both write and read occur	Randomization on wr_en and rd_en under constraint that write occurs more than read	Included as cross cover for wr_en and rd_en and all the output flags	Output checked with assertion
FIFO_9	data_out, at start is invalid due to initial state	Randomization on rst_n under constraint that reset is deasserted most of the time	Not included	Output checked against golden model
FIFO_10	If the reset is asserted, we will be at the initial state	Randomization on rst_n under constraint that reset is deasserted most of the time	Not included	Output checked with assertion

Top & Interface modules:

```
module FIFO_top();
    bit clk;

    //clock generation
    initial begin
        clk = 0;
        forever
            #5 clk = ~clk;
    end

    FIFO_interface fifo_if (clk);
    FIFO           dut     (fifo_if);
    FIFO_tb        tb      (fifo_if);
    FIFO_monitor   mon     (fifo_if);

endmodule
```

```
interface FIFO_interface (clk);
    parameter FIFO_WIDTH = 16;
    parameter FIFO_DEPTH = 8;
    parameter max_fifo_addr = $clog2(FIFO_DEPTH);

    input clk;
    logic [FIFO_WIDTH-1:0] data_in;
    logic rst_n, wr_en, rd_en;
    logic [FIFO_WIDTH-1:0] data_out;
    logic wr_ack, overflow;
    logic full, empty, almostfull, almostempty, underflow;


    // DUT modport
    modport DUT(
        input clk, data_in, rst_n, wr_en, rd_en,
        output data_out, wr_ack, overflow, full, empty, almostfull, almostempty, underflow
    );

    modport TEST(
        output data_in, rst_n, wr_en, rd_en,
        input clk, data_out, wr_ack, overflow, full, empty, almostfull, almostempty, underflow
    );


    modport MONITOR(
        input data_in, rst_n, wr_en, rd_en, clk,
        input data_out, wr_ack, overflow, full, empty, almostfull, almostempty, underflow
    );

endinterface
```

Packages:

```
package FIFO_transaction_pkg;
    class FIFO_transaction;
    parameter FIFO_WIDTH = 16;
    parameter FIFO_DEPTH = 8;
    logic clk;
    rand logic [FIFO_WIDTH-1:0] data_in;
    rand logic rst_n;
    rand logic wr_en;
    rand logic rd_en;
    logic [FIFO_WIDTH-1:0] data_out;
    logic wr_ack, overflow, full, empty, almostfull, almostempty, underflow;

    //Inside of this class add the FIFO inputs and outputs as class variables of the class as well as adding 2 integers (RD_EN_ON_DIST & WR_EN_ON_DIST)
    int RD_EN_ON_DIST , WR_EN_ON_DIST ;

    //Add a constructor that takes 2 inputs and override the values of RD_EN_ON_DIST and WR_EN_ON_DIST, let the default of RD_EN_ON_DIST be 30 and WR_EN_ON_DIST be 70
    function new(int RD_EN_ON_DIST = 30 ,int WR_EN_ON_DIST = 70);
        this.RD_EN_ON_DIST = RD_EN_ON_DIST;
        this.WR_EN_ON_DIST = WR_EN_ON_DIST;
    endfunction

    //Assert reset less often
    constraint reset_c {rst_n dist {1:/98 , 0:/2};}

    // Constraint the write enable to be high with distribution of the value WR_EN_ON_DIST & to be low with 100-WR_EN_ON_DIST
    constraint wr_en_c { wr_en dist {1 := WR_EN_ON_DIST, 0 := 100-WR_EN_ON_DIST}; }

    // Constraint the read enable the same as write enable but using RD_EN_ON_DIST
    constraint rd_en_c { rd_en dist {1 := RD_EN_ON_DIST, 0 := 100-RD_EN_ON_DIST}; }

    endclass
endpackage
```

```
package FIFO_coverage_pkg;

    import FIFO_transaction_pkg::*;

    class FIFO_coverage;

        //The class will have an object of the class FIFO_transaction named F_cvg_txn.

        FIFO_transaction F_cvg_txn;

        /* Create a covergroup. The coverage needed is cross coverage between 3 signals which are
           write enable, read enable and each output control signals (outputs except data_out) to
           make sure that all combinations of write and read enable took place in all state of the FIFO */

        covergroup read_write_cg;
            wr_en_cp:          coverpoint F_cvg_txn.wr_en       {option.weight         = 0;}
            rd_en_cp:          coverpoint F_cvg_txn.rd_en       {option.weight         = 0;}
            full_cp:           coverpoint F_cvg_txn.full        {bins full_HIGH        = {1}; option.weight = 0;}
            empty_cp:          coverpoint F_cvg_txn.empty       {bins empty_HIGH       = {1}; option.weight = 0;}
            overflow_cp:       coverpoint F_cvg_txn.overflow    {bins overflow_HIGH    = {1}; option.weight = 0;}
            underflow_cp:      coverpoint F_cvg_txn.underflow   {bins underflow_HIGH   = {1}; option.weight = 0;}
            wr_ack_cp:         coverpoint F_cvg_txn.wr_ack      {bins wr_ack_HIGH      = {1}; option.weight = 0;}
            almostfull_cp:     coverpoint F_cvg_txn.almostfull  {bins almostfull_HIGH  = {1}; option.weight = 0;}
            almostempty_cp:    coverpoint F_cvg_txn.almostempty {bins almostempty_HIGH = {1}; option.weight = 0;}



            almostfull_cross:   cross wr_en_cp, rd_en_cp, almostfull_cp;
            almostempty_cross:  cross wr_en_cp, rd_en_cp, almostempty_cp;


            // empty shouldn't be HIGH if write enable is 1 whatever read
            empty_cross:cross wr_en_cp,rd_en_cp,empty_cp iff(F_cvg_txn.rst_n)       {
                illegal_bins empty_and_wr     =  binsof(wr_en_cp)intersect {1} && (binsof(rd_en_cp) intersect{0}  || binsof(rd_en_cp)     intersect{1}) && binsof(empty_cp) intersect{1};
            }
            // full shouldn't be HIGH if read enable is 1 whatever write
            full_cross:cross wr_en_cp , rd_en_cp , full_cp       {
                illegal_bins full_wr_rd       = (binsof(wr_en_cp)intersect {0} || binsof(wr_en_cp) intersect {1}) && binsof(rd_en_cp)     intersect{1} &&  binsof (full_cp) intersect{1};
            }
            // overflow shouldn't be HIGH if write enable is 0
            overflow_cross:cross wr_en_cp ,rd_en_cp, overflow_cp {
                illegal_bins wr_and_over      = binsof(wr_en_cp)intersect {0} && binsof(overflow_cp)  intersect{1};
            }
            // underflow shouldn't be HIGH if read enable is 0
            underflow_cross:cross wr_en_cp,rd_en_cp,underflow_cp {
                illegal_bins underflow_and_rd = binsof(rd_en_cp)intersect {0} && binsof(underflow_cp) intersect{1};
            }
            // write ack shouldn't be HIGH if write enable is 0
             wr_ack_cross:cross wr_en_cp , rd_en_cp , wr_ack_cp  {
                illegal_bins wr_and_wr_ack    = binsof(wr_en_cp)intersect {0} && binsof(wr_ack_cp)    intersect{1};
            }
        endgroup

        /*Create a void function inside it named sample_data that takes one input named F_txn.
        This input is an object of class FIFO_transaction. This function will do the following
```

```
        /*Create a void function inside it named sample_data that takes one input named F_txn.
        This input is an object of class FIFO_transaction. This function will do the following
        1. Assign F_txn to F_cvg_txn
        2. Trigger the sampling of the covergroup using the .sample method*/

        function new();
            // Create an instance of the covergroup
            read_write_cg = new();
        endfunction

        function void sample_data(FIFO_transaction F_txn);
            F_cvg_txn = F_txn;
            read_write_cg.sample();
        endfunction

    endclass

endpackage
```

```
package FIFO_scoreboard_pkg;
    import FIFO_transaction_pkg::*;
    import shared_pkg::*;

    class FIFO_scoreboard;

        parameter FIFO_WIDTH = 16;
        parameter FIFO_DEPTH = 8;

        //Add variables for the data_out_ref to be used in the reference_model function
        logic [FIFO_WIDTH-1:0] data_out_ref;

        // declare queue to use for golden model
        logic [FIFO_WIDTH-1:0] FIFO_queue[$];


        /*---Create a function named check_data that takes one input which of type FIFO_transaction
        ---Inside this function, call another function named reference_model that you will
        create and pass to it the same object that you have received.*/

        function void check_data(FIFO_transaction FIFO_trans);
            reference_model(FIFO_trans);

        /*---After the reference_model function returns, you will compare the reference
        outputs calculated with the outputs of the object received. Increment the
        error_count or correct_count. Also, display a message if error occurs. */

            if( data_out_ref != FIFO_trans.data_out) begin
                shared::error_count++;
                $display("Error occurs");
            end
            else begin
                shared::correct_count++;
            end
        endfunction

        /*---Reference_model--- */

        function void reference_model(FIFO_transaction FIFO_ref);
```

```
        /*---Reference_model--- */

        function void reference_model(FIFO_transaction FIFO_ref);

            if(!FIFO_ref.rst_n) begin
                FIFO_queue.delete();
                //data_out_ref = 0;
            end

            else begin

                if ( (!FIFO_ref.wr_en) && (FIFO_ref.rd_en) && (FIFO_queue.size() != 0) ) begin
                    data_out_ref = FIFO_queue.pop_front();
                end

                else if ( (FIFO_ref.wr_en) && (!FIFO_ref.rd_en) && (FIFO_queue.size() != FIFO_DEPTH) ) begin
                    FIFO_queue.push_back(FIFO_ref.data_in);
                end

                else if ( (FIFO_ref.wr_en) && (FIFO_ref.rd_en) )begin

                    if(FIFO_queue.size() == 0)begin
                        FIFO_queue.push_back(FIFO_ref.data_in);
                    end

                    else if(FIFO_queue.size() == FIFO_DEPTH) begin
                        data_out_ref = FIFO_queue.pop_front();
                    end

                    else begin
                        // pop the front and push the back
                        data_out_ref = FIFO_queue.pop_front();
                        FIFO_queue.push_back(FIFO_ref.data_in);
                    end
                end
                else begin
                end
            end
        endfunction

    endclass
endpackage
```

+

Monitor module:

```
import FIFO_transaction_pkg::*;
import FIFO_scoreboard_pkg::*;
import FIFO_coverage_pkg::*;
import shared_pkg::*;
module FIFO_monitor(FIFO_interface.MONITOR fifo_if);
    // Declare objects for the classes
    FIFO_transaction fifo_trans;
    FIFO_scoreboard fifo_scoreboard = new();
    FIFO_coverage fifo_coverage = new();

    initial begin
        fifo_trans      = new();

        /*It will have an initial block and inside it a forever loop that waits for negedge clock at the start of
        the loop and then sample the data of the interface and assign it to the data variables of the
        object of class FIFO_transaction.*/

        forever begin
            fifo_trans.data_in     = fifo_if.data_in;
            fifo_trans.wr_en       = fifo_if.wr_en;
            fifo_trans.rd_en       = fifo_if.rd_en;
            fifo_trans.rst_n       = fifo_if.rst_n;
            @(negedge fifo_if.clk);
            fifo_trans.data_out    = fifo_if.data_out;
            fifo_trans.wr_ack      = fifo_if.wr_ack;
            fifo_trans.overflow    = fifo_if.overflow;
            fifo_trans.full        = fifo_if.full;
            fifo_trans.empty       = fifo_if.empty;
            fifo_trans.almostfull  = fifo_if.almostfull;
            fifo_trans.almostempty = fifo_if.almostempty;
            fifo_trans.underflow   = fifo_if.underflow;
            @(posedge fifo_if.clk);

            /* And then after that there will be fork join, where 2 processes
            will run, the first one is calling a method named sample_data of the object of class
            FIFO_coverage and the second process is calling a method named check_data of the object of
            class FIFO_scoreboard.*/

            fork begin
                fifo_coverage.sample_data(fifo_trans); // Call the coverage sampling method
            end
            begin
                fifo_scoreboard.check_data(fifo_trans); // Call the scoreboard checking method
            end
            /*After the fork join ends, you will check for the signal test_finished if it is high or not. If it high,
            then stop the simulation and display a message with summary of correct and error counts. */
            join
                if (shared::test_finished) begin
                    $display("Simulation finished.");
                    $display("Test Finished ! error_count is %d, correct_count is %d", shared::error_count , shared::correct_count);
                    $stop;
                end
        end
    end
endmodule
```

TestBench module:

```
import shared_pkg::*;
import FIFO_transaction_pkg::*;

module FIFO_tb(FIFO_interface.TEST fifo_if);

    /*The tb will reset the DUT and then randomize the inputs. At the end of
    the test, the tb will assert a signal named test_finished.*/
    FIFO_transaction trans ;
    integer i;
    logic clk;
    initial begin
        trans = new();

        assert_rst();


        for (i = 0; i < 20000; i = i + 1) begin
            assert(trans.randomize());

            fifo_if.rst_n   = trans.rst_n;
            fifo_if.wr_en   = trans.wr_en;
            fifo_if.rd_en   = trans.rd_en;
            fifo_if.data_in = trans.data_in;
            @(negedge fifo_if.clk);
        end

        assert_rst();

        shared::test_finished = 1;

    end

    // Reset task
    task assert_rst;
        fifo_if.rst_n   = 0;
        fifo_if.wr_en   = 0;
        fifo_if.rd_en   = 0;
        fifo_if.data_in = 0;
        @(negedge fifo_if.clk);
        @(negedge fifo_if.clk);
        fifo_if.rst_n   = 1;
    endtask
endmodule
```

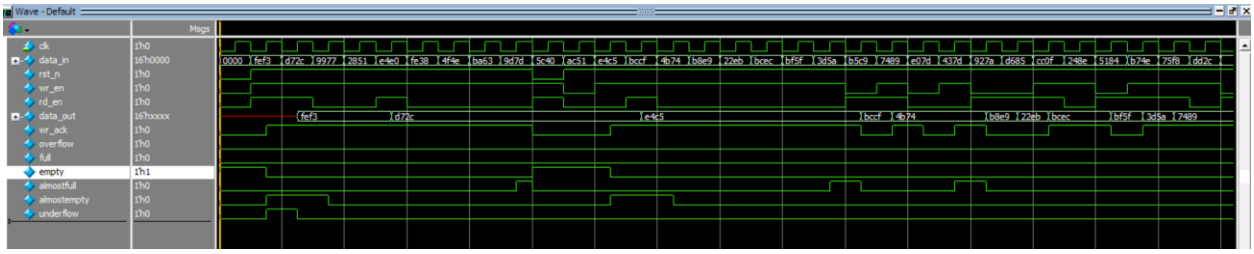
Src Files List & Do File:

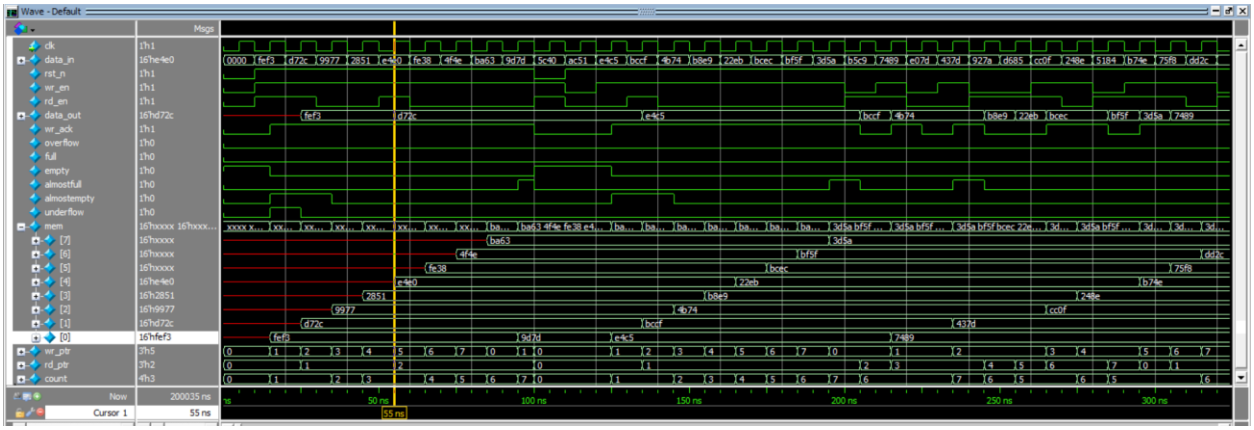
```
FIFO_interface.sv
shared_pkg.sv
FIFO_transaction_pkg.sv
FIFO_coverage_pkg.sv
FIFO_scoreboard_pkg.sv
FIFO_monitor.sv
FIFO.sv
FIFO_tb.sv
FIFO_top.sv
```

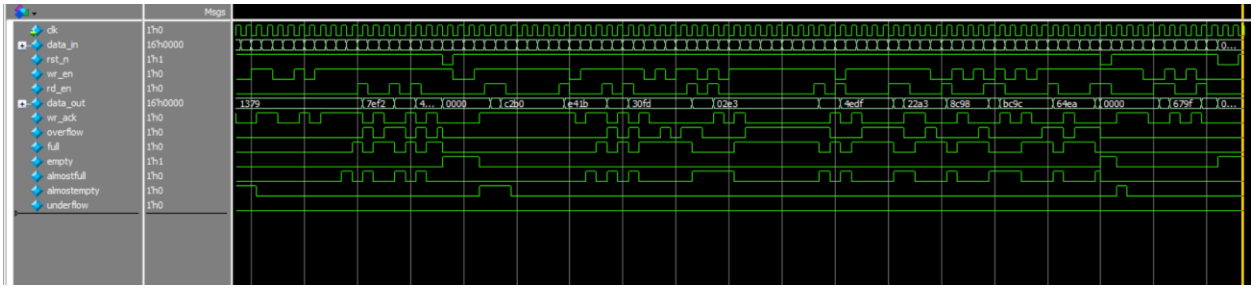
```
vlib work
vlog -f src_files.list -mfcu +define+SIM +cover
vsim -voptargs=+acc work.FIFO_top -coverage
add wave -r /FIFO_top/fifo_if/*
coverage save -onexit FIFO_top.ucdb
run -all
```

Transcript & Wave Form :

```
# Simulation finished.
# Test Finished ! error_count is          0, correct_count is       20004
# ** Note: $stop    : FIFO_monitor.sv(51)
#    Time: 200035 ns  Iteration: 1  Instance: /FIFO_top/mon
# Break in Module FIFO_monitor at FIFO_monitor.sv line 51
```







Assertions with covers:

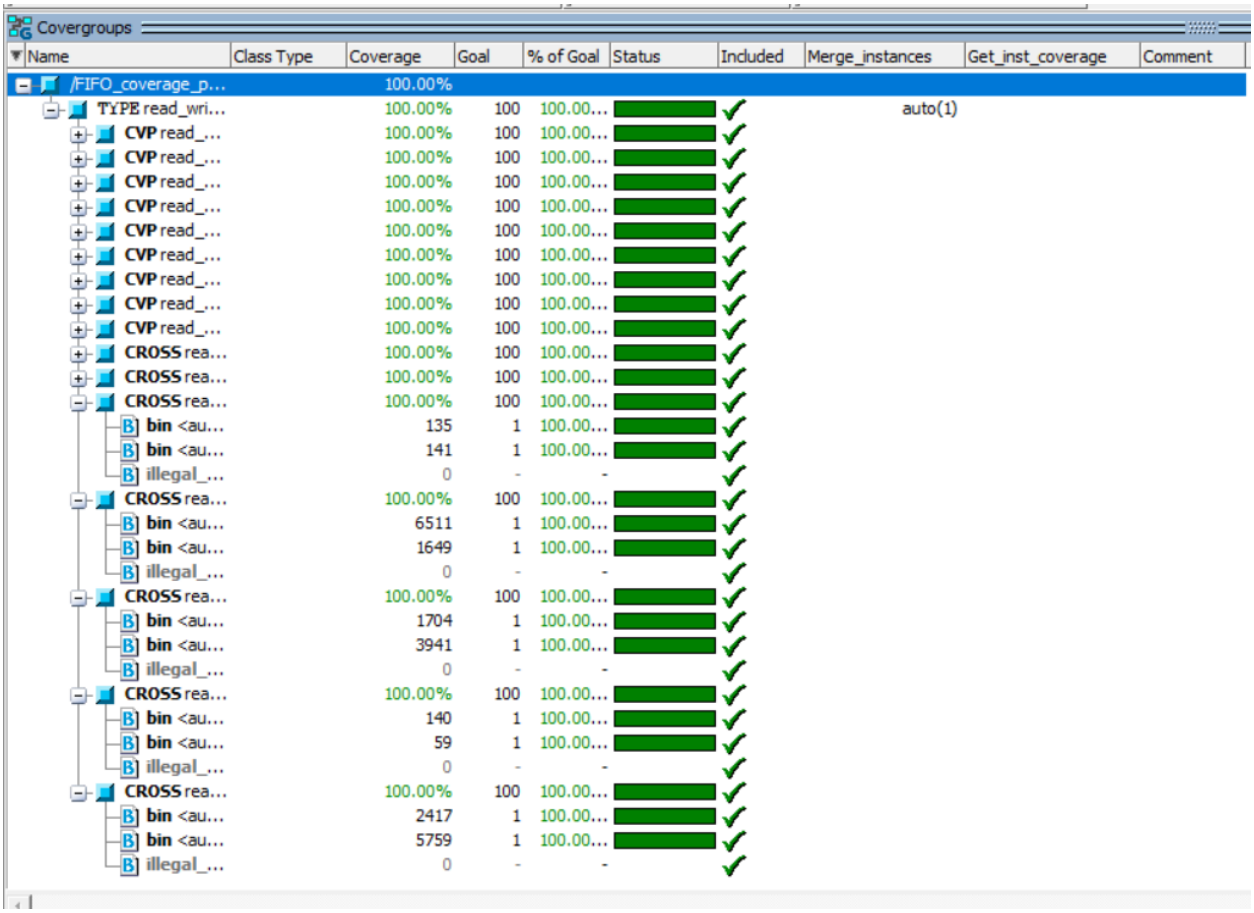
Assertions

Name	Assertion Type	Language	Enable	Failure Count	Pass Count	Active Count	Memory	Peak Memory	Peak Memory Time	Cumulative Threads	ATV	Assertion Expression	Included
/FIFO_top/dut/ove…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/und…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/wr_…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/ove…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/und…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/wr_…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/full…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) (cou…	✓
/FIFO_top/dut/em…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) (cou…	✓
/FIFO_top/dut/alm…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) (cou…	✓
/FIFO_top/dut/alm…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) (cou…	✓
/FIFO_top/dut/full…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) (cou…	✓
/FIFO_top/dut/em…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) (cou…	✓
/FIFO_top/dut/alm…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) (cou…	✓
/FIFO_top/dut/alm…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) (cou…	✓
/FIFO_top/dut/rd_…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/wr_…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/rd_…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/rd_…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/wr_…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/alm…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/alm…	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifo_if.clk) disa…	✓
/FIFO_top/dut/rst_…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (count==0)	✓
/FIFO_top/dut/rst_…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (rd_ptr==0)	✓
/FIFO_top/dut/rst_…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (wr_ptr==0)	✓
/FIFO_top/dut/rst_…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (~full)	✓
/FIFO_top/dut/rst_…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (.empty)	✓
/FIFO_top/dut/rst_…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (~almostfull)	✓
/FIFO_top/dut/rst_…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (~almostempty)	✓
/FIFO_top/dut/rst_…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (~overflow)	✓
/FIFO_top/dut/rst_…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (~underflow)	✓
/FIFO_top/dut/rst_…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (~wr_ack)	✓
/FIFO_top/tb/#ubl…	Immediate	SVA	on	0	1	-	-	-	-		off	assert (randomize(…))	✓

Cover Directives

Name	Language	Enabled	Log	Count	AtLeast	Limit	Weight	Cmplt %	Cmplt graph	Included	Memory	Peak Memory	Peak Memory Time	Cumulative Threads
/FIFO_top/dut/ove…	SVA	✓	Off	5527	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/und…	SVA	✓	Off	195	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/wr_…	SVA	✓	Off	8036	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/ove…	SVA	✓	Off	8036	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/und…	SVA	✓	Off	5561	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/wr_…	SVA	✓	Off	5527	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/full…	SVA	✓	Off	7998	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/em…	SVA	✓	Off	1050	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/alm…	SVA	✓	Off	5182	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/alm…	SVA	✓	Off	896	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/full…	SVA	✓	Off	12005	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/em…	SVA	✓	Off	18953	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/alm…	SVA	✓	Off	14821	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/alm…	SVA	✓	Off	19107	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rd_…	SVA	✓	Off	1650	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/wr_…	SVA	✓	Off	5662	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rd_…	SVA	✓	Off	2237	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rd_…	SVA	✓	Off	5561	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/wr_…	SVA	✓	Off	8036	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/alm…	SVA	✓	Off	2525	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/alm…	SVA	✓	Off	75	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rst_…	SVA	✓	Off	385	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rst_…	SVA	✓	Off	385	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rst_…	SVA	✓	Off	385	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rst_…	SVA	✓	Off	385	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rst_…	SVA	✓	Off	385	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rst_…	SVA	✓	Off	385	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rst_…	SVA	✓	Off	385	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rst_…	SVA	✓	Off	385	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rst_…	SVA	✓	Off	385	1	Unli…	1	100%		✓	0	0	0 ns	0
/FIFO_top/dut/rst_…	SVA	✓	Off	385	1	Unli…	1	100%		✓	0	0	0 ns	0

Cover Groups:



Assertion Text Report:

assertion_report.txt

```
Coverage Report by instance with details

================================================================================
=== Instance: /FIFO_top/dut
=== Design Unit: work.FIFO
================================================================================

Assertion Coverage:
    Assertions                      31        31         0   100.00%
--------------------------------------------------------------------
Name                  File(Line)                    Failure      Pass
                                                    Count        Count
--------------------------------------------------------------------
/FIFO_top/dut/overflow_assert_1
                      FIFO.sv(107)                        0          1
/FIFO_top/dut/underflow_assert_1
                      FIFO.sv(108)                        0          1
/FIFO_top/dut/wr_ack_assert_1
                      FIFO.sv(109)                        0          1
/FIFO_top/dut/overflow_assert_2
                      FIFO.sv(114)                        0          1
/FIFO_top/dut/underflow_assert_2
                      FIFO.sv(115)                        0          1
/FIFO_top/dut/wr_ack_assert_2
                      FIFO.sv(116)                        0          1
/FIFO_top/dut/full_assert_1
                      FIFO.sv(123)                        0          1
/FIFO_top/dut/empty_assert_1
                      FIFO.sv(124)                        0          1
/FIFO_top/dut/almostfull_assert_1
                      FIFO.sv(125)                        0          1
/FIFO_top/dut/almostempty_assert_1
                      FIFO.sv(126)                        0          1
/FIFO_top/dut/full_assert_2
                      FIFO.sv(132)                        0          1
/FIFO_top/dut/empty_assert_2
                      FIFO.sv(133)                        0          1
/FIFO_top/dut/almostfull_assert_2
                      FIFO.sv(134)                        0          1
/FIFO_top/dut/almostempty_assert_2
                      FIFO.sv(135)                        0          1
/FIFO_top/dut/rd_count_assert
                      FIFO.sv(144)                        0          1
/FIFO_top/dut/wr_count_assert
                      FIFO.sv(145)                        0          1
/FIFO_top/dut/rd_wr_count_assert
                      FIFO.sv(146)                        0          1
/FIFO_top/dut/rd_ptr_assert
```

assertion_report.txt

```
/FIFO_top/dut/rd_ptr_assert
                     FIFO.sv(154)                          0          1
/FIFO_top/dut/wr_ptr_assert
                     FIFO.sv(155)                          0          1
/FIFO_top/dut/almostfull_full_assert
                     FIFO.sv(164)                          0          1
/FIFO_top/dut/almostempty_empty_assert
                     FIFO.sv(167)                          0          1
/FIFO_top/dut/rst_count_assert
                     FIFO.sv(175)                          0          1
/FIFO_top/dut/rst_rd_ptr_assert
                     FIFO.sv(176)                          0          1
/FIFO_top/dut/rst_wr_ptr_assert
                     FIFO.sv(177)                          0          1
/FIFO_top/dut/rst_full_assert
                     FIFO.sv(178)                          0          1
/FIFO_top/dut/rst_empty_assert
                     FIFO.sv(179)                          0          1
/FIFO_top/dut/rst_almostfull_assert
                     FIFO.sv(180)                          0          1
/FIFO_top/dut/rst_almostempty_assert
                     FIFO.sv(181)                          0          1
/FIFO_top/dut/rst_overflow_assert
                     FIFO.sv(183)                          0          1
/FIFO_top/dut/rst_underflow_assert
                     FIFO.sv(184)                          0          1
/FIFO_top/dut/rst_wr_ack_assert
                     FIFO.sv(185)                          0          1

============================================================================
=== Instance: /FIFO_top/tb
=== Design Unit: work.FIFO_tb
============================================================================

Assertion Coverage:
    Assertions                       1         1         0   100.00%
----------------------------------------------------------------
Name                 File(Line)                    Failure      Pass
                                                   Count        Count
----------------------------------------------------------------
/FIFO_top/tb/#ublk#182146786#17/immed__18
                     FIFO_tb.sv(18)                        0          1


ASSERTION RESULTS:
----------------------------------------------------------------
Name                 File(Line)                    Failure      Pass
                                                   Count        Count
```

```
ASSERTION RESULTS:
--------------------------------------------------------------------
Name                    File(Line)                      Failure      Pass
                                                        Count        Count
--------------------------------------------------------------------
/FIFO_top/dut/overflow_assert_1
                        FIFO.sv(107)                        0          1
/FIFO_top/dut/underflow_assert_1
                        FIFO.sv(108)                        0          1
/FIFO_top/dut/wr_ack_assert_1
                        FIFO.sv(109)                        0          1
/FIFO_top/dut/overflow_assert_2
                        FIFO.sv(114)                        0          1
/FIFO_top/dut/underflow_assert_2
                        FIFO.sv(115)                        0          1
/FIFO_top/dut/wr_ack_assert_2
                        FIFO.sv(116)                        0          1
/FIFO_top/dut/full_assert_1
                        FIFO.sv(123)                        0          1
/FIFO_top/dut/empty_assert_1
                        FIFO.sv(124)                        0          1
/FIFO_top/dut/almostfull_assert_1
                        FIFO.sv(125)                        0          1
/FIFO_top/dut/almostempty_assert_1
                        FIFO.sv(126)                        0          1
/FIFO_top/dut/full_assert_2
                        FIFO.sv(132)                        0          1
/FIFO_top/dut/empty_assert_2
                        FIFO.sv(133)                        0          1
/FIFO_top/dut/almostfull_assert_2
                        FIFO.sv(134)                        0          1
/FIFO_top/dut/almostempty_assert_2
                        FIFO.sv(135)                        0          1
/FIFO_top/dut/rd_count_assert
                        FIFO.sv(144)                        0          1
/FIFO_top/dut/wr_count_assert
                        FIFO.sv(145)                        0          1
/FIFO_top/dut/rd_wr_count_assert
                        FIFO.sv(146)                        0          1
/FIFO_top/dut/rd_ptr_assert
                        FIFO.sv(154)                        0          1
/FIFO_top/dut/wr_ptr_assert
                        FIFO.sv(155)                        0          1
/FIFO_top/dut/almostfull_full_assert
```

```
/FIFO_top/dut/full_assert_2
                     FIFO.sv(132)                          0          1
/FIFO_top/dut/empty_assert_2
                     FIFO.sv(133)                          0          1
/FIFO_top/dut/almostfull_assert_2
                     FIFO.sv(134)                          0          1
/FIFO_top/dut/almostempty_assert_2
                     FIFO.sv(135)                          0          1
/FIFO_top/dut/rd_count_assert
                     FIFO.sv(144)                          0          1
/FIFO_top/dut/wr_count_assert
                     FIFO.sv(145)                          0          1
/FIFO_top/dut/rd_wr_count_assert
                     FIFO.sv(146)                          0          1
/FIFO_top/dut/rd_ptr_assert
                     FIFO.sv(154)                          0          1
/FIFO_top/dut/wr_ptr_assert
                     FIFO.sv(155)                          0          1
/FIFO_top/dut/almostfull_full_assert
                     FIFO.sv(164)                          0          1
/FIFO_top/dut/almostempty_empty_assert
                     FIFO.sv(167)                          0          1
/FIFO_top/dut/rst_count_assert
                     FIFO.sv(175)                          0          1
/FIFO_top/dut/rst_rd_ptr_assert
                     FIFO.sv(176)                          0          1
/FIFO_top/dut/rst_wr_ptr_assert
                     FIFO.sv(177)                          0          1
/FIFO_top/dut/rst_full_assert
                     FIFO.sv(178)                          0          1
/FIFO_top/dut/rst_empty_assert
                     FIFO.sv(179)                          0          1
/FIFO_top/dut/rst_almostfull_assert
                     FIFO.sv(180)                          0          1
/FIFO_top/dut/rst_almostempty_assert
                     FIFO.sv(181)                          0          1
/FIFO_top/dut/rst_overflow_assert
                     FIFO.sv(183)                          0          1
/FIFO_top/dut/rst_underflow_assert
                     FIFO.sv(184)                          0          1
/FIFO_top/dut/rst_wr_ack_assert
                     FIFO.sv(185)                          0          1
/FIFO_top/tb/#ublk#182146786#17/immed__18
                     FIFO_tb.sv(18)                        0          1

Total Coverage By Instance (filtered view): 100.00%
```

Function Coverage Text Report:

functional_cover_report.txt

```
Coverage Report by instance with details

================================================================================
=== Instance: /FIFO_top/dut
=== Design Unit: work.FIFO
================================================================================

Directive Coverage:
    Directives                      32        32         0   100.00%

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                    Design Design   Lang File(Line)      Hits Status
                                        Unit   UnitType
--------------------------------------------------------------------------------------------
/FIFO_top/dut/overflow_cover_1          FIFO   Verilog  SVA  FIFO.sv(110)    5527 Covered
/FIFO_top/dut/underflow_cover_1         FIFO   Verilog  SVA  FIFO.sv(111)     195 Covered
/FIFO_top/dut/wr_ack_cover_1            FIFO   Verilog  SVA  FIFO.sv(112)    8036 Covered
/FIFO_top/dut/overflow_cover_2          FIFO   Verilog  SVA  FIFO.sv(117)    8036 Covered
/FIFO_top/dut/underflow_cover_2         FIFO   Verilog  SVA  FIFO.sv(118)    5561 Covered
/FIFO_top/dut/wr_ack_cover_2            FIFO   Verilog  SVA  FIFO.sv(119)    5527 Covered
/FIFO_top/dut/full_cover_1              FIFO   Verilog  SVA  FIFO.sv(127)    7998 Covered
/FIFO_top/dut/empty_cover_1             FIFO   Verilog  SVA  FIFO.sv(128)    1050 Covered
/FIFO_top/dut/almostfull_cover_1        FIFO   Verilog  SVA  FIFO.sv(129)    5182 Covered
/FIFO_top/dut/almostempty_cover_1       FIFO   Verilog  SVA  FIFO.sv(130)     896 Covered
/FIFO_top/dut/full_cover_2              FIFO   Verilog  SVA  FIFO.sv(136)   12005 Covered
/FIFO_top/dut/empty_cover_2             FIFO   Verilog  SVA  FIFO.sv(137)   18953 Covered
/FIFO_top/dut/almostfull_cover_2        FIFO   Verilog  SVA  FIFO.sv(138)   14821 Covered
/FIFO_top/dut/almostempty_cover_2       FIFO   Verilog  SVA  FIFO.sv(139)   19107 Covered
/FIFO_top/dut/rd_count_cover            FIFO   Verilog  SVA  FIFO.sv(147)    1650 Covered
/FIFO_top/dut/wr_count_cover            FIFO   Verilog  SVA  FIFO.sv(148)    5662 Covered
/FIFO_top/dut/rd_wr_count_cover         FIFO   Verilog  SVA  FIFO.sv(149)    2237 Covered
/FIFO_top/dut/rd_ptr_cover              FIFO   Verilog  SVA  FIFO.sv(156)    5561 Covered
/FIFO_top/dut/wr_ptr_cover              FIFO   Verilog  SVA  FIFO.sv(157)    8036 Covered
/FIFO_top/dut/almostfull_full_cover     FIFO   Verilog  SVA  FIFO.sv(169)    2525 Covered
/FIFO_top/dut/almostempty_empty_cover   FIFO   Verilog  SVA  FIFO.sv(170)      75 Covered
/FIFO_top/dut/rst_count_cover           FIFO   Verilog  SVA  FIFO.sv(186)     385 Covered
/FIFO_top/dut/rst_rd_ptr_cover          FIFO   Verilog  SVA  FIFO.sv(187)     385 Covered
/FIFO_top/dut/rst_wr_ptr_cover          FIFO   Verilog  SVA  FIFO.sv(188)     385 Covered
/FIFO_top/dut/rst_full_cover            FIFO   Verilog  SVA  FIFO.sv(189)     385 Covered
/FIFO_top/dut/rst_empty_cover           FIFO   Verilog  SVA  FIFO.sv(190)     385 Covered
/FIFO_top/dut/rst_almostfull_cover      FIFO   Verilog  SVA  FIFO.sv(191)     385 Covered
/FIFO_top/dut/rst_almostempty_cover     FIFO   Verilog  SVA  FIFO.sv(192)     385 Covered
/FIFO_top/dut/rst_data_out_cover        FIFO   Verilog  SVA  FIFO.sv(193)     385 Covered
/FIFO_top/dut/rst_overflow_cover        FIFO   Verilog  SVA  FIFO.sv(194)     385 Covered
/FIFO_top/dut/rst_underflow_cover       FIFO   Verilog  SVA  FIFO.sv(195)     385 Covered
/FIFO_top/dut/rst_wr_ack_cover          FIFO   Verilog  SVA  FIFO.sv(196)     385 Covered
```

functional_cover_report.txt

```
================================================================================
=== Instance: /FIFO_coverage_pkg
=== Design Unit: work.FIFO_coverage_pkg
================================================================================

Covergroup Coverage:
    Covergroups                     1        na        na   100.00%
        Coverpoints/Crosses        16        na        na        na
            Covergroup Bins        29        29         0   100.00%
--------------------------------------------------------------------------------------------------------------------
Covergroup                                             Metric       Goal       Bins    Status

--------------------------------------------------------------------------------------------------------------------
 TYPE /FIFO_coverage_pkg/FIFO_coverage/read_write_cg
                                                      100.00%        100          -    Covered
    covered/total bins:                                    29         29          -
    missing/total bins:                                     0         29          -
    % Hit:                                            100.00%        100          -
    Coverpoint wr_en_cp                               100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      5916          1          -    Covered
        bin auto[1]                                     14088          1          -    Covered
    Coverpoint rd_en_cp                               100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                     14004          1          -    Covered
        bin auto[1]                                      6000          1          -    Covered
    Coverpoint full_cp                                100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
        missing/total bins:                                 0          1          -
        % Hit:                                        100.00%        100          -
        bin full_HIGH                                    8160          1          -    Covered
    Coverpoint empty_cp                               100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
        missing/total bins:                                 0          1          -
        % Hit:                                        100.00%        100          -
        bin empty_HIGH                                    669          1          -    Covered
    Coverpoint overflow_cp                            100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
        missing/total bins:                                 0          1          -
        % Hit:                                        100.00%        100          -
        bin overflow_HIGH                                5645          1          -    Covered
    Coverpoint underflow_cp                           100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
```

```
        missing/total bins:                          0         2         -
        % Hit:                                 100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[0],auto[1],empty_HIGH>         135         1         -    Covered
            bin <auto[0],auto[0],empty_HIGH>         141         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin empty_and_wr                   0                   -    ZERO
    Cross full_cross                               100.00%       100         -    Covered
        covered/total bins:                            2         2         -
        missing/total bins:                            0         2         -
        % Hit:                                 100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[0],full_HIGH>         6511         1         -    Covered
            bin <auto[0],auto[0],full_HIGH>         1649         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin full_wr_rd                     0                   -    ZERO
    Cross overflow_cross                           100.00%       100         -    Covered
        covered/total bins:                            2         2         -
        missing/total bins:                            0         2         -
        % Hit:                                 100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],overflow_HIGH>         1704         1         -    Covered
            bin <auto[1],auto[0],overflow_HIGH>         3941         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin wr_and_over                    0                   -    ZERO
    Cross underflow_cross                          100.00%       100         -    Covered
        covered/total bins:                            2         2         -
        missing/total bins:                            0         2         -
        % Hit:                                 100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],underflow_HIGH>          140         1         -    Covered
            bin <auto[0],auto[1],underflow_HIGH>           59         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin underflow_and_rd               0                   -    ZERO
    Cross wr_ack_cross                             100.00%       100         -    Covered
        covered/total bins:                            2         2         -
        missing/total bins:                            0         2         -
        % Hit:                                 100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],wr_ack_HIGH>         2417         1         -    Covered
            bin <auto[1],auto[0],wr_ack_HIGH>         5759         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin wr_and_wr_ack                  0                   -    ZERO

COVERGROUP COVERAGE:
--------------------------------------------------------------------------------------------------------------
```

```
        missing/total bins:                          0         2         -
        % Hit:                                 100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[0],auto[1],empty_HIGH>         135         1         -    Covered
            bin <auto[0],auto[0],empty_HIGH>         141         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin empty_and_wr                   0                   -    ZERO
    Cross full_cross                               100.00%       100         -    Covered
        covered/total bins:                            2         2         -
        missing/total bins:                            0         2         -
        % Hit:                                 100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[0],full_HIGH>         6511         1         -    Covered
            bin <auto[0],auto[0],full_HIGH>         1649         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin full_wr_rd                     0                   -    ZERO
    Cross overflow_cross                           100.00%       100         -    Covered
        covered/total bins:                            2         2         -
        missing/total bins:                            0         2         -
        % Hit:                                 100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],overflow_HIGH>         1704         1         -    Covered
            bin <auto[1],auto[0],overflow_HIGH>         3941         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin wr_and_over                    0                   -    ZERO
    Cross underflow_cross                          100.00%       100         -    Covered
        covered/total bins:                            2         2         -
        missing/total bins:                            0         2         -
        % Hit:                                 100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],underflow_HIGH>          140         1         -    Covered
            bin <auto[0],auto[1],underflow_HIGH>           59         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin underflow_and_rd               0                   -    ZERO
    Cross wr_ack_cross                             100.00%       100         -    Covered
        covered/total bins:                            2         2         -
        missing/total bins:                            0         2         -
        % Hit:                                 100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],wr_ack_HIGH>         2417         1         -    Covered
            bin <auto[1],auto[0],wr_ack_HIGH>         5759         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin wr_and_wr_ack                  0                   -    ZERO

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1
```

```
TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                     Design Design   Lang File(Line)      Hits Status
                                         Unit   UnitType
--------------------------------------------------------------------------------------------
/FIFO_top/dut/overflow_cover_1           FIFO   Verilog  SVA  FIFO.sv(110)    5527 Covered
/FIFO_top/dut/underflow_cover_1          FIFO   Verilog  SVA  FIFO.sv(111)     195 Covered
/FIFO_top/dut/wr_ack_cover_1             FIFO   Verilog  SVA  FIFO.sv(112)    8036 Covered
/FIFO_top/dut/overflow_cover_2           FIFO   Verilog  SVA  FIFO.sv(117)    8036 Covered
/FIFO_top/dut/underflow_cover_2          FIFO   Verilog  SVA  FIFO.sv(118)    5561 Covered
/FIFO_top/dut/wr_ack_cover_2             FIFO   Verilog  SVA  FIFO.sv(119)    5527 Covered
/FIFO_top/dut/full_cover_1               FIFO   Verilog  SVA  FIFO.sv(127)    7998 Covered
/FIFO_top/dut/empty_cover_1              FIFO   Verilog  SVA  FIFO.sv(128)    1050 Covered
/FIFO_top/dut/almostfull_cover_1         FIFO   Verilog  SVA  FIFO.sv(129)    5182 Covered
/FIFO_top/dut/almostempty_cover_1        FIFO   Verilog  SVA  FIFO.sv(130)     896 Covered
/FIFO_top/dut/full_cover_2               FIFO   Verilog  SVA  FIFO.sv(136)   12005 Covered
/FIFO_top/dut/empty_cover_2              FIFO   Verilog  SVA  FIFO.sv(137)   18953 Covered
/FIFO_top/dut/almostfull_cover_2         FIFO   Verilog  SVA  FIFO.sv(138)   14821 Covered
/FIFO_top/dut/almostempty_cover_2        FIFO   Verilog  SVA  FIFO.sv(139)   19107 Covered
/FIFO_top/dut/rd_count_cover             FIFO   Verilog  SVA  FIFO.sv(147)    1650 Covered
/FIFO_top/dut/wr_count_cover             FIFO   Verilog  SVA  FIFO.sv(148)    5662 Covered
/FIFO_top/dut/rd_wr_count_cover          FIFO   Verilog  SVA  FIFO.sv(149)    2237 Covered
/FIFO_top/dut/rd_ptr_cover               FIFO   Verilog  SVA  FIFO.sv(156)    5561 Covered
/FIFO_top/dut/wr_ptr_cover               FIFO   Verilog  SVA  FIFO.sv(157)    8036 Covered
/FIFO_top/dut/almostfull_full_cover      FIFO   Verilog  SVA  FIFO.sv(169)    2525 Covered
/FIFO_top/dut/almostempty_empty_cover    FIFO   Verilog  SVA  FIFO.sv(170)      75 Covered
/FIFO_top/dut/rst_count_cover            FIFO   Verilog  SVA  FIFO.sv(186)     385 Covered
/FIFO_top/dut/rst_rd_ptr_cover           FIFO   Verilog  SVA  FIFO.sv(187)     385 Covered
/FIFO_top/dut/rst_wr_ptr_cover           FIFO   Verilog  SVA  FIFO.sv(188)     385 Covered
/FIFO_top/dut/rst_full_cover             FIFO   Verilog  SVA  FIFO.sv(189)     385 Covered
/FIFO_top/dut/rst_empty_cover            FIFO   Verilog  SVA  FIFO.sv(190)     385 Covered
/FIFO_top/dut/rst_almostfull_cover       FIFO   Verilog  SVA  FIFO.sv(191)     385 Covered
/FIFO_top/dut/rst_almostempty_cover      FIFO   Verilog  SVA  FIFO.sv(192)     385 Covered
/FIFO_top/dut/rst_data_out_cover         FIFO   Verilog  SVA  FIFO.sv(193)     385 Covered
/FIFO_top/dut/rst_overflow_cover         FIFO   Verilog  SVA  FIFO.sv(194)     385 Covered
/FIFO_top/dut/rst_underflow_cover        FIFO   Verilog  SVA  FIFO.sv(195)     385 Covered
/FIFO_top/dut/rst_wr_ack_cover           FIFO   Verilog  SVA  FIFO.sv(196)     385 Covered

TOTAL DIRECTIVE COVERAGE: 100.00%  COVERS: 32

Total Coverage By Instance (filtered view): 100.00%
```

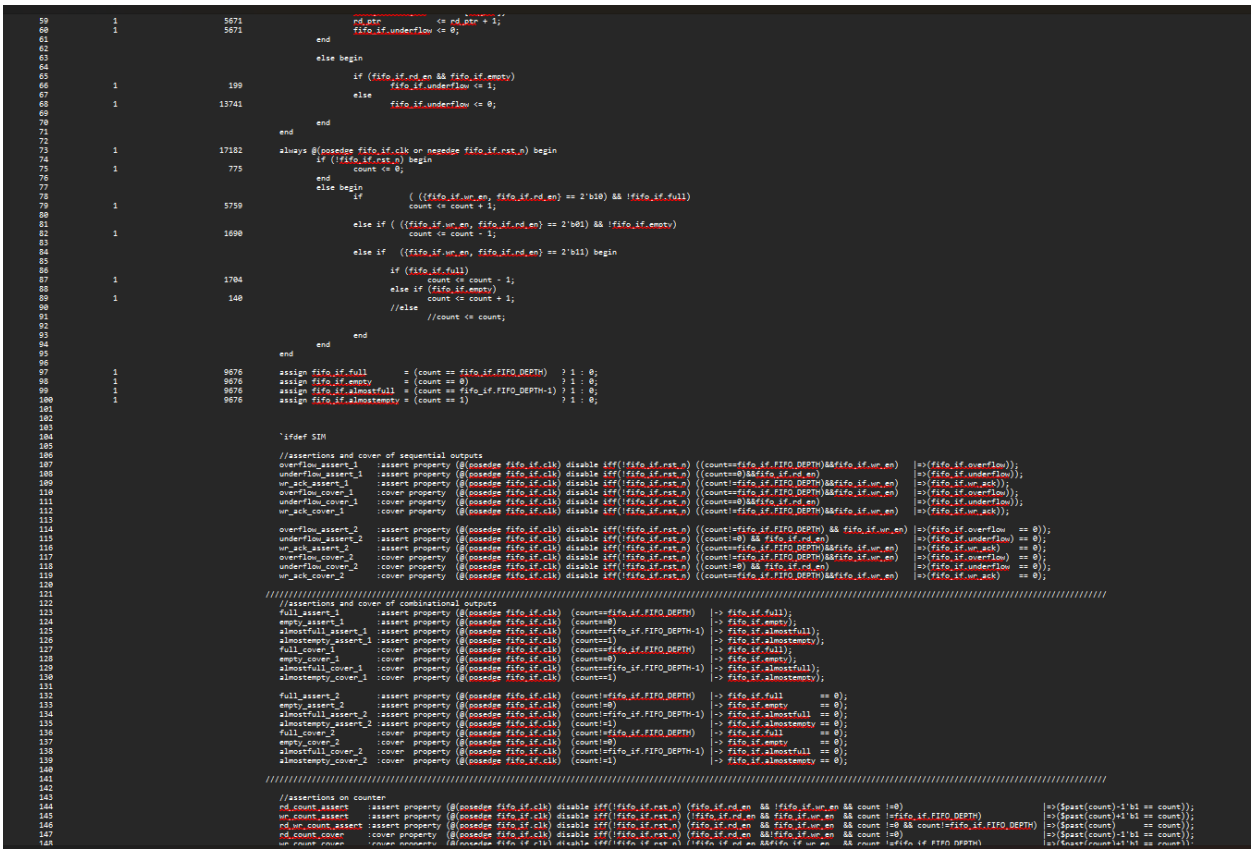
Summary Coverage Report :

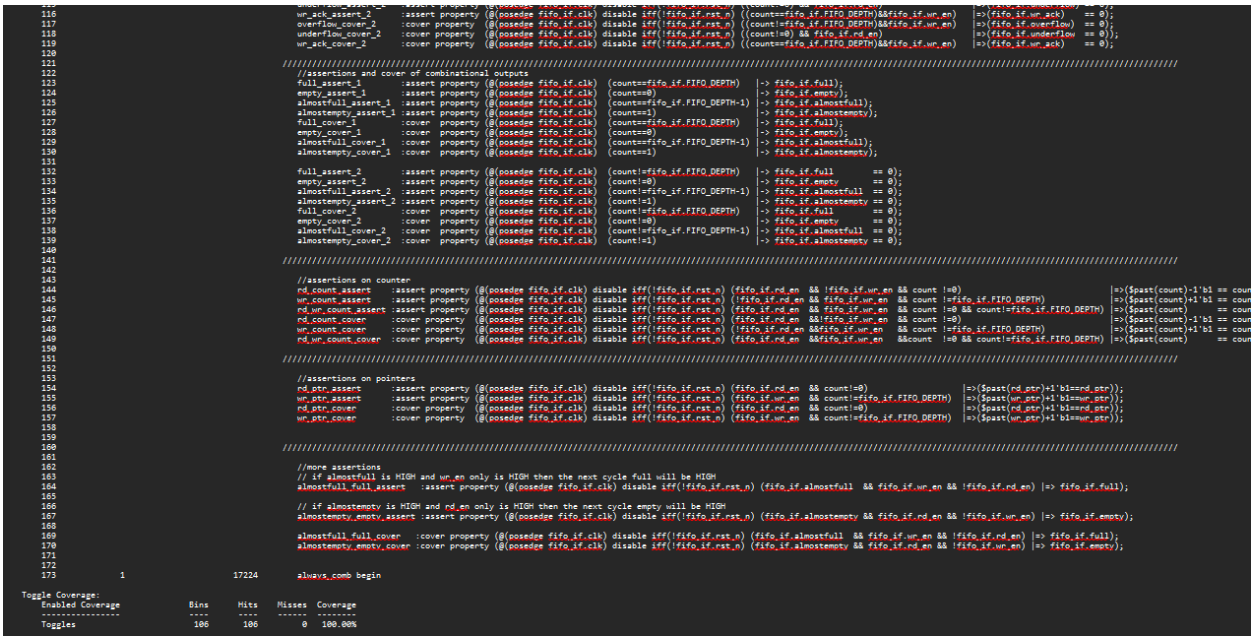
```
Coverage Report by instance with details

================================================================================
=== Instance: /FIFO_top/fifo_if
=== Design Unit: work.FIFO_interface
================================================================================
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         86        86         0   100.00%

================================Toggle Details================================

Toggle Coverage for instance /FIFO_top/fifo_if --

                                          Node      1H->0L      0L->1H  "Coverage"
                                          ----------------------------------------
                                   almostempty           1           1      100.00
                                    almostfull           1           1      100.00
                                           clk           1           1      100.00
                                  data_in[15-0]          1           1      100.00
                                 data_out[15-0]          1           1      100.00
                                         empty           1           1      100.00
                                          full           1           1      100.00
                                      overflow           1           1      100.00
                                         rd_en           1           1      100.00
                                         rst_n           1           1      100.00
                                     underflow           1           1      100.00
                                        wr_ack           1           1      100.00
                                         wr_en           1           1      100.00

Total Node Count     =         43
Toggled Node Count   =         43
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (86 of 86 bins)

================================================================================
=== Instance: /FIFO_top/dut
=== Design Unit: work.FIFO
================================================================================
```

```
================================================================================
=== Instance: /FIFO_top/dut
=== Design Unit: work.FIFO
================================================================================

Assertion Coverage:
    Assertions                      32        32         0   100.00%
--------------------------------------------------------------------
Name                 File(Line)                    Failure      Pass
                                                   Count        Count
--------------------------------------------------------------------
/FIFO_top/dut/overflow_assert_1
                     FIFO.sv(107)                         0          1
/FIFO_top/dut/underflow_assert_1
                     FIFO.sv(108)                         0          1
/FIFO_top/dut/wr_ack_assert_1
                     FIFO.sv(109)                         0          1
/FIFO_top/dut/overflow_assert_2
                     FIFO.sv(114)                         0          1
/FIFO_top/dut/underflow_assert_2
                     FIFO.sv(115)                         0          1
/FIFO_top/dut/wr_ack_assert_2
                     FIFO.sv(116)                         0          1
/FIFO_top/dut/full_assert_1
                     FIFO.sv(123)                         0          1
/FIFO_top/dut/empty_assert_1
                     FIFO.sv(124)                         0          1
/FIFO_top/dut/almostfull_assert_1
                     FIFO.sv(125)                         0          1
/FIFO_top/dut/almostempty_assert_1
                     FIFO.sv(126)                         0          1
/FIFO_top/dut/full_assert_2
                     FIFO.sv(132)                         0          1
/FIFO_top/dut/empty_assert_2
                     FIFO.sv(133)                         0          1
/FIFO_top/dut/almostfull_assert_2
                     FIFO.sv(134)                         0          1
/FIFO_top/dut/almostempty_assert_2
                     FIFO.sv(135)                         0          1
/FIFO_top/dut/rd_count_assert
                     FIFO.sv(144)                         0          1
/FIFO_top/dut/wr_count_assert
                     FIFO.sv(145)                         0          1
/FIFO_top/dut/rd_wr_count_assert
                     FIFO.sv(146)                         0          1
/FIFO_top/dut/rd_ptr_assert
                     FIFO.sv(154)                         0          1
/FIFO_top/dut/wr_ptr_assert
                     FIFO.sv(155)                         0          1
/FIFO_top/dut/almostfull_full_assert
                     FIFO.sv(164)                         0          1
/FIFO_top/dut/almostempty_empty_assert
                     FIFO.sv(167)                         0          1
/FIFO_top/dut/rst_count_assert
                     FIFO.sv(175)                         0          1
/FIFO_top/dut/rst_rd_ptr_assert
                     FIFO.sv(176)                         0          1
/FIFO_top/dut/rst_wr_ptr_assert
                     FIFO.sv(177)                         0          1
/FIFO_top/dut/rst_full_assert
                     FIFO.sv(178)                         0          1
/FIFO_top/dut/rst_empty_assert
                     FIFO.sv(179)                         0          1
/FIFO_top/dut/rst_almostfull_assert
                     FIFO.sv(180)                         0          1
/FIFO_top/dut/rst_almostempty_assert
                     FIFO.sv(181)                         0          1
/FIFO_top/dut/rst_data_out_assert
                     FIFO.sv(182)                         0          1
/FIFO_top/dut/rst_overflow_assert
                     FIFO.sv(183)                         0          1
/FIFO_top/dut/rst_underflow_assert
                     FIFO.sv(184)                         0          1
/FIFO_top/dut/rst_wr_ack_assert
                     FIFO.sv(185)                         0          1
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        29        29         0   100.00%

================================Branch Details================================
```

```
DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                    Design Design   Lang File(Line)      Hits Status
                                        Unit   UnitType
--------------------------------------------------------------------------------------------
/FIFO_top/dut/overflow_cover_1          FIFO   Verilog  SVA  FIFO.sv(110)    5527 Covered
/FIFO_top/dut/underflow_cover_1         FIFO   Verilog  SVA  FIFO.sv(111)     195 Covered
/FIFO_top/dut/wr_ack_cover_1            FIFO   Verilog  SVA  FIFO.sv(112)    8036 Covered
/FIFO_top/dut/overflow_cover_2          FIFO   Verilog  SVA  FIFO.sv(117)    8036 Covered
/FIFO_top/dut/underflow_cover_2         FIFO   Verilog  SVA  FIFO.sv(118)    5561 Covered
/FIFO_top/dut/wr_ack_cover_2            FIFO   Verilog  SVA  FIFO.sv(119)    5527 Covered
/FIFO_top/dut/full_cover_1              FIFO   Verilog  SVA  FIFO.sv(127)    7998 Covered
/FIFO_top/dut/empty_cover_1             FIFO   Verilog  SVA  FIFO.sv(128)    1050 Covered
/FIFO_top/dut/almostfull_cover_1        FIFO   Verilog  SVA  FIFO.sv(129)    5182 Covered
/FIFO_top/dut/almostempty_cover_1       FIFO   Verilog  SVA  FIFO.sv(130)     896 Covered
/FIFO_top/dut/full_cover_2              FIFO   Verilog  SVA  FIFO.sv(136)   12005 Covered
/FIFO_top/dut/empty_cover_2             FIFO   Verilog  SVA  FIFO.sv(137)   18953 Covered
/FIFO_top/dut/almostfull_cover_2        FIFO   Verilog  SVA  FIFO.sv(138)   14821 Covered
/FIFO_top/dut/almostempty_cover_2       FIFO   Verilog  SVA  FIFO.sv(139)   19107 Covered
/FIFO_top/dut/rd_count_cover            FIFO   Verilog  SVA  FIFO.sv(147)    1650 Covered
/FIFO_top/dut/wr_count_cover            FIFO   Verilog  SVA  FIFO.sv(148)    5662 Covered
/FIFO_top/dut/rd_wr_count_cover         FIFO   Verilog  SVA  FIFO.sv(149)    2237 Covered
/FIFO_top/dut/rd_ptr_cover              FIFO   Verilog  SVA  FIFO.sv(156)    5561 Covered
/FIFO_top/dut/wr_ptr_cover              FIFO   Verilog  SVA  FIFO.sv(157)    8036 Covered
/FIFO_top/dut/almostfull_full_cover     FIFO   Verilog  SVA  FIFO.sv(169)    2525 Covered
/FIFO_top/dut/almostempty_empty_cover   FIFO   Verilog  SVA  FIFO.sv(170)      75 Covered
/FIFO_top/dut/rst_count_cover           FIFO   Verilog  SVA  FIFO.sv(186)     385 Covered
/FIFO_top/dut/rst_rd_ptr_cover          FIFO   Verilog  SVA  FIFO.sv(187)     385 Covered
/FIFO_top/dut/rst_wr_ptr_cover          FIFO   Verilog  SVA  FIFO.sv(188)     385 Covered
/FIFO_top/dut/rst_full_cover            FIFO   Verilog  SVA  FIFO.sv(189)     385 Covered
/FIFO_top/dut/rst_empty_cover           FIFO   Verilog  SVA  FIFO.sv(190)     385 Covered
/FIFO_top/dut/rst_almostfull_cover      FIFO   Verilog  SVA  FIFO.sv(191)     385 Covered
/FIFO_top/dut/rst_almostempty_cover     FIFO   Verilog  SVA  FIFO.sv(192)     385 Covered
/FIFO_top/dut/rst_overflow_cover        FIFO   Verilog  SVA  FIFO.sv(194)     385 Covered
/FIFO_top/dut/rst_underflow_cover       FIFO   Verilog  SVA  FIFO.sv(195)     385 Covered
/FIFO_top/dut/rst_wr_ack_cover          FIFO   Verilog  SVA  FIFO.sv(196)     385 Covered

TOTAL DIRECTIVE COVERAGE: 100.00%  COVERS: 31
```

```
================================Statement Details================================

Statement Coverage for instance /FIFO_top/dut --
NOTE: The modification timestamp for source file 'FIFO.sv' has been altered since compilation.

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File FIFO.sv
    1                                                module FIFO(FIFO_interface.DUT fifo_if);
    2
    3                                                   logic [fifo_if.FIFO_WIDTH-1:0] data_in;
    4                                                   logic clk, rst_n, wr_en, rd_en;
    5                                                   logic [fifo_if.FIFO_WIDTH-1:0] data_out;
    6                                                   logic wr_ack, overflow, full, empty, almostfull, almostempty, underflow;
    7
    8               1                         40009     assign clk         = fifo_if.clk;
    9               1                           771     assign rst_n       = fifo_if.rst_n;
    10              1                          8314     assign wr_en       = fifo_if.wr_en;
    11              1                          8398     assign rd_en       = fifo_if.rd_en;
    12              1                         20002     assign data_in     = fifo_if.data_in;
    13              1                          6042     assign data_out    = fifo_if.data_out;
    14              1                          8634     assign wr_ack      = fifo_if.wr_ack;
    15              1                          5902     assign overflow    = fifo_if.overflow;
    16              1                          5142     assign full        = fifo_if.full;
    17              1                           916     assign empty       = fifo_if.empty;
    18              1                          6254     assign almostfull  = fifo_if.almostfull;
    19              1                          1086     assign almostempty = fifo_if.almostempty;
    20              1                           356     assign underflow   = fifo_if.underflow;
    21
    22                                                  reg [fifo_if.FIFO_WIDTH-1    :0] mem [fifo_if.FIFO_DEPTH-1:0];
    23                                                  reg [fifo_if.max_fifo_addr-1 :0] wr_ptr, rd_ptr;
    24                                                  reg [fifo_if.max_fifo_addr   :0] count;
    25
    26              1                         20389     always @(posedge fifo_if.clk or negedge fifo_if.rst_n) begin //////write operation////////
    27                                                          if (!fifo_if.rst_n) begin
    28              1                           778                     wr_ptr          <= 0;
    29              1                           778                     fifo_if.wr_ack  <= 0;
    30              1                           778                     fifo_if.overflow <= 0;
    31                                                          end
    32
    33                                                          else if (fifo_if.wr_en && count < fifo_if.FIFO_DEPTH) begin
    34              1                          8176                     mem[wr_ptr]     <= fifo_if.data_in;
    35              1                          8176                     fifo_if.wr_ack  <= 1;
    36              1                          8176                     wr_ptr          <= wr_ptr + 1;
    37              1                          8176                     fifo_if.overflow <= 0;
    38                                                          end
    39
    40                                                          else begin
    41              1                         11435                     fifo_if.wr_ack <= 0;
    42
    43                                                                  if (fifo_if.wr_en && fifo_if.full)
    44              1                          5645                             fifo_if.overflow <= 1;
    45                                                                  else
    46              1                          5790                             fifo_if.overflow <= 0;
    47                                                                  end
    48                                                  end
    49
    50              1                         20389     always @(posedge fifo_if.clk or negedge fifo_if.rst_n) begin ///////read operation///////
    51                                                          if (!fifo_if.rst_n) begin
    52              1                           778                     rd_ptr          <= 0;
    53              1                           778                     fifo_if.underflow <= 0;
    54              1                           778                     fifo_if.data_out  <= 0;
    55                                                          end
    56
    57                                                          else if (fifo_if.rd_en && count != 0) begin
    58              1                          5671                     fifo_if.data_out  <= mem[rd_ptr];
    59              1                          5671                     rd_ptr          <= rd_ptr + 1;
    60              1                          5671                     fifo_if.underflow <= 0;
    61                                                          end
    62
    63                                                          else begin
    64
    65                                                                  if (fifo_if.rd_en && fifo_if.empty)
    66              1                           199                             fifo_if.underflow <= 1;
    67                                                                  else
    68              1                         13741                             fifo_if.underflow <= 0;
    69
    70                                                                  end
    71                                                  end
    72
    73              1                         17182     always @(posedge fifo_if.clk or negedge fifo_if.rst_n) begin
    74                                                          if (!fifo_if.rst_n) begin
    75              1                           775                     count <= 0;
    76                                                          end
    77                                                          else begin
    78                                                                  if          ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b10) && !fifo_if.full)
    79              1                          5759                             count <= count + 1;
    80
    81                                                                  else if ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b01) && !fifo_if.empty)
```





```
116                                     wr_ack_assert_2     :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count==fifo_if.FIFO_DEPTH)&&fifo_if.wr_en)   |=>(fifo_if.wr_ack)   == 0);
117                                     overflow_cover_2    :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count!=fifo_if.FIFO_DEPTH)&&fifo_if.wr_en)   |=>(fifo_if.overflow)  == 0);
118                                     underflow_cover_2   :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count!=0) && fifo_if.rd_en)                 |=>(fifo_if.underflow) == 0);
119                                     wr_ack_cover_2      :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) ((count==fifo_if.FIFO_DEPTH)&&fifo_if.wr_en)   |=>(fifo_if.wr_ack)   == 0);
120
121                                   ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
122                                     //assertions and cover of combinational outputs
123                                     full_assert_1       :assert property (@(posedge fifo_if.clk)  (count==fifo_if.FIFO_DEPTH)   |-> fifo_if.full);
124                                     empty_assert_1      :assert property (@(posedge fifo_if.clk)  (count==0)                    |-> fifo_if.empty);
125                                     almostfull_assert_1 :assert property (@(posedge fifo_if.clk)  (count==fifo_if.FIFO_DEPTH-1) |-> fifo_if.almostfull);
126                                     almostempty_assert_1:assert property (@(posedge fifo_if.clk)  (count==1)                    |-> fifo_if.almostempty);
127                                     full_cover_1        :cover  property (@(posedge fifo_if.clk)  (count==fifo_if.FIFO_DEPTH)   |-> fifo_if.full);
128                                     empty_cover_1       :cover  property (@(posedge fifo_if.clk)  (count==0)                    |-> fifo_if.empty);
129                                     almostfull_cover_1  :cover  property (@(posedge fifo_if.clk)  (count==fifo_if.FIFO_DEPTH-1) |-> fifo_if.almostfull);
130                                     almostempty_cover_1 :cover  property (@(posedge fifo_if.clk)  (count==1)                    |-> fifo_if.almostempty);
131
132                                     full_assert_2       :assert property (@(posedge fifo_if.clk)  (count!=fifo_if.FIFO_DEPTH)   |-> fifo_if.full       == 0);
133                                     empty_assert_2      :assert property (@(posedge fifo_if.clk)  (count!=0)                    |-> fifo_if.empty      == 0);
134                                     almostfull_assert_2 :assert property (@(posedge fifo_if.clk)  (count!=fifo_if.FIFO_DEPTH-1) |-> fifo_if.almostfull  == 0);
135                                     almostempty_assert_2:assert property (@(posedge fifo_if.clk)  (count!=1)                    |-> fifo_if.almostempty == 0);
136                                     full_cover_2        :cover  property (@(posedge fifo_if.clk)  (count!=fifo_if.FIFO_DEPTH)   |-> fifo_if.full       == 0);
137                                     empty_cover_2       :cover  property (@(posedge fifo_if.clk)  (count!=0)                    |-> fifo_if.empty      == 0);
138                                     almostfull_cover_2  :cover  property (@(posedge fifo_if.clk)  (count!=fifo_if.FIFO_DEPTH-1) |-> fifo_if.almostfull  == 0);
139                                     almostempty_cover_2 :cover  property (@(posedge fifo_if.clk)  (count!=1)                    |-> fifo_if.almostempty == 0);
140
141                                   ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
142
143                                     //assertions on counter
144                                     rd_count_assert     :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  && !fifo_if.wr_en && count !=0)                      |=>($past(count)-1'b1 == coun
145                                     wr_count_assert     :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (!fifo_if.rd_en && fifo_if.wr_en  && count !=fifo_if.FIFO_DEPTH)            |=>($past(count)+1'b1 == coun
146                                     rd_wr_count_assert :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  && fifo_if.wr_en  && count !=0 && count!=fifo_if.FIFO_DEPTH) |=>($past(count)      == coun
147                                     rd_count_cover      :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  &&!fifo_if.wr_en  && count !=0)                      |=>($past(count)-1'b1 == coun
148                                     wr_count_cover      :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (!fifo_if.rd_en &&fifo_if.wr_en   && count !=fifo_if.FIFO_DEPTH)            |=>($past(count)+1'b1 == coun
149                                     rd_wr_count_cover  :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  &&fifo_if.wr_en   &&count  !=0 && count!=fifo_if.FIFO_DEPTH) |=>($past(count)      == coun
150
151                                   ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
152
153                                     //assertions on pointers
154                                     rd_ptr_assert       :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  && count!=0)                    |=>($past(rd_ptr)+1'b1==rd_ptr));
155                                     wr_ptr_assert       :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.wr_en  && count!=fifo_if.FIFO_DEPTH)  |=>($past(wr_ptr)+1'b1==wr_ptr));
156                                     rd_ptr_cover        :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.rd_en  && count!=0)                    |=>($past(rd_ptr)+1'b1==rd_ptr));
157                                     wr_ptr_cover        :cover property  (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.wr_en  && count!=fifo_if.FIFO_DEPTH)  |=>($past(wr_ptr)+1'b1==wr_ptr));
158
159
160                                   ////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
161
162                                     //more assertions
163                                     // if almostfull is HIGH and wr_en only is HIGH then the next cycle full will be HIGH
164                                     almostfull_full_assert   :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.almostfull  && fifo_if.wr_en && !fifo_if.rd_en) |=> fifo_if.full);
165
166                                     // if almostempty is HIGH and rd_en only is HIGH then the next cycle empty will be HIGH
167                                     almostempty_empty_assert :assert property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.almostempty && fifo_if.rd_en && !fifo_if.wr_en) |=> fifo_if.empty);
168
169                                     almostfull_full_cover   :cover property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.almostfull  && fifo_if.wr_en && !fifo_if.rd_en) |=> fifo_if.full);
170                                     almostempty_empty_cover :cover property (@(posedge fifo_if.clk) disable iff(!fifo_if.rst_n) (fifo_if.almostempty && fifo_if.rd_en && !fifo_if.wr_en) |=> fifo_if.empty);
171
172
173             1                17224       always_comb begin

Toggle Coverage:
    Enabled Coverage             Bins      Hits    Misses  Coverage
    ----------------             ----      ----    ------  --------
    Toggles                       106       106         0  100.00%
```

```
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                        106       106         0   100.00%

================================Toggle Details================================

Toggle Coverage for instance /FIFO_top/dut --

                                          Node      1H->0L      0L->1H  "Coverage"
                                          ----------------------------------------
                                    almostempty           1           1      100.00
                                     almostfull           1           1      100.00
                                            clk           1           1      100.00
                                       count[3-0]         1           1      100.00
                                     data_in[15-0]        1           1      100.00
                                    data_out[15-0]        1           1      100.00
                                          empty           1           1      100.00
                                           full           1           1      100.00
                                       overflow           1           1      100.00
                                          rd_en           1           1      100.00
                                     rd_ptr[2-0]          1           1      100.00
                                          rst_n           1           1      100.00
                                      underflow           1           1      100.00
                                         wr_ack           1           1      100.00
                                          wr_en           1           1      100.00
                                     wr_ptr[2-0]          1           1      100.00

Total Node Count     =         53
Toggled Node Count   =         53
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (106 of 106 bins)

=============================================================================
=== Instance: /FIFO_top/tb
=== Design Unit: work.FIFO_tb
=============================================================================

Assertion Coverage:
    Assertions                       1         1         0   100.00%
-----------------------------------------------------------------
Name                 File(Line)                   Failure      Pass
                                                  Count        Count
-----------------------------------------------------------------
/FIFO_top/tb/#ublk#182146786#17/immed__18
                     FIFO_tb.sv(18)                     0          1
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                      18        18         0   100.00%

================================Statement Details================================
```

```
=== Instance: /FIFO_coverage_pkg
=== Design Unit: work.FIFO_coverage_pkg
================================================================================

Covergroup Coverage:
    Covergroups                    1        na        na   100.00%
        Coverpoints/Crosses       16        na        na        na
            Covergroup Bins       29        29         0   100.00%
----------------------------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status

----------------------------------------------------------------------------------------------------
 TYPE /FIFO_coverage_pkg/FIFO_coverage/read_write_cg
                                                   100.00%        100          -    Covered
    covered/total bins:                                 29         29          -
    missing/total bins:                                  0         29          -
    % Hit:                                         100.00%        100          -
    Coverpoint wr_en_cp                            100.00%        100          -    Covered
        covered/total bins:                              2          2          -
        missing/total bins:                              0          2          -
        % Hit:                                     100.00%        100          -
        bin auto[0]                                   5916          1          -    Covered
        bin auto[1]                                  14088          1          -    Covered
    Coverpoint rd_en_cp                            100.00%        100          -    Covered
        covered/total bins:                              2          2          -
        missing/total bins:                              0          2          -
        % Hit:                                     100.00%        100          -
        bin auto[0]                                  14004          1          -    Covered
        bin auto[1]                                   6000          1          -    Covered
    Coverpoint full_cp                             100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                     100.00%        100          -
        bin full_HIGH                                 8160          1          -    Covered
    Coverpoint empty_cp                            100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                     100.00%        100          -
        bin empty_HIGH                                 669          1          -    Covered
    Coverpoint overflow_cp                         100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                     100.00%        100          -
        bin overflow_HIGH                             5645          1          -    Covered
    Coverpoint underflow_cp                        100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                     100.00%        100          -
        bin underflow_HIGH                             199          1          -    Covered
    Coverpoint wr_ack_cp                           100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                     100.00%        100          -
        bin wr_ack_HIGH                               8176          1          -    Covered
    Coverpoint almostfull_cp                       100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                     100.00%        100          -
        bin almostfull_HIGH                           5288          1          -    Covered
    Coverpoint almostempty_cp                      100.00%        100          -    Covered
        covered/total bins:                              1          1          -
        missing/total bins:                              0          1          -
        % Hit:                                     100.00%        100          -
        bin almostempty_HIGH                           907          1          -    Covered
    Cross almostfull_cross                         100.00%        100          -    Covered
        covered/total bins:                              4          4          -
        missing/total bins:                              0          4          -
        % Hit:                                     100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],almostfull_HIGH>     2789          1          -    Covered
            bin <auto[0],auto[1],almostfull_HIGH>      704          1          -    Covered
            bin <auto[1],auto[0],almostfull_HIGH>      718          1          -    Covered
            bin <auto[0],auto[0],almostfull_HIGH>     1077          1          -    Covered
    Cross almostempty_cross                        100.00%        100          -    Covered
        covered/total bins:                              4          4          -
        missing/total bins:                              0          4          -
        % Hit:                                     100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],almostempty_HIGH>     329          1          -    Covered
            bin <auto[0],auto[1],almostempty_HIGH>      85          1          -    Covered
            bin <auto[1],auto[0],almostempty_HIGH>     317          1          -    Covered
            bin <auto[0],auto[0],almostempty_HIGH>     176          1          -    Covered
    Cross empty_cross                              100.00%        100          -    Covered
        covered/total bins:                              2          2          -
        missing/total bins:                              0          2          -
        % Hit:                                     100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[0],auto[1],empty_HIGH>           172          1          -    Covered
            bin <auto[0],auto[0],empty_HIGH>           230          1          -    Covered
        Illegal and Ignore Bins:
```

```
        % Hit:                                  100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[0],auto[1],empty_HIGH>       172         1         -    Covered
            bin <auto[0],auto[0],empty_HIGH>       230         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin empty_and_wr               267                   -    Occurred
    Cross full_cross                            100.00%       100         -    Covered
        covered/total bins:                           2         2         -
        missing/total bins:                           0         2         -
        % Hit:                                  100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[0],full_HIGH>       6511         1         -    Covered
            bin <auto[0],auto[0],full_HIGH>       1649         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin full_wr_rd                   0                   -    ZERO
    Cross overflow_cross                        100.00%       100         -    Covered
        covered/total bins:                           2         2         -
        missing/total bins:                           0         2         -
        % Hit:                                  100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],overflow_HIGH>       1704         1         -    Covered
            bin <auto[1],auto[0],overflow_HIGH>       3941         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin wr_and_over                  0                   -    ZERO
    Cross underflow_cross                       100.00%       100         -    Covered
        covered/total bins:                           2         2         -
        missing/total bins:                           0         2         -
        % Hit:                                  100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],underflow_HIGH>        140         1         -    Covered
            bin <auto[0],auto[1],underflow_HIGH>         59         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin underflow_and_rd             0                   -    ZERO
    Cross wr_ack_cross                          100.00%       100         -    Covered
        covered/total bins:                           2         2         -
        missing/total bins:                           0         2         -
        % Hit:                                  100.00%       100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],wr_ack_HIGH>       2417         1         -    Covered
            bin <auto[1],auto[0],wr_ack_HIGH>       5759         1         -    Covered
        Illegal and Ignore Bins:
            illegal_bin wr_and_wr_ack                0                   -    ZERO
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                       3         3         0   100.00%

================================Statement Details================================

Statement Coverage for instance //FIFO_coverage_pkg --
NOTE: The modification timestamp for source file 'FIFO_coverage_pkg.sv' has been altered since compilation.

    Line        Item                      Count    Source
    ----        ----                      -----    ------
  File FIFO_coverage_pkg.sv
    1                                              package FIFO_coverage_pkg;
    2
    3                                                 import FIFO_transaction_pkg::*;
    4
    5                                                 class FIFO_coverage;
    6
    7                                                       //The class will have an object of the class FIFO_transaction named F_cvg_txn.
    8
    9                                                       FIFO_transaction F_cvg_txn;
    10
    11                                                      /* Create a covergroup. The coverage needed is cross coverage between 3 signals which are
    12                                                         write enable, read enable and each output control signals (outputs except data_out) to
    13                                                         make sure that all combinations of write and read enable took place in all state of the FIFO */
    14
    15                                                      covergroup read_write_cg;
    16                                                            wr_en_cp:         coverpoint F_cvg_txn.wr_en      {option.weight        = 0;}
    17                                                            rd_en_cp:         coverpoint F_cvg_txn.rd_en      {option.weight        = 0;}
    18                                                            full_cp:          coverpoint F_cvg_txn.full       {bins full_HIGH       = {1}; option.weight = 0;}
    19                                                            empty_cp:         coverpoint F_cvg_txn.empty      {bins empty_HIGH      = {1}; option.weight = 0;}
    20                                                            overflow_cp:      coverpoint F_cvg_txn.overflow   {bins overflow_HIGH   = {1}; option.weight = 0;}
    21                                                            underflow_cp:     coverpoint F_cvg_txn.underflow  {bins underflow_HIGH  = {1}; option.weight = 0;}
    22                                                            wr_ack_cp:        coverpoint F_cvg_txn.wr_ack     {bins wr_ack_HIGH     = {1}; option.weight = 0;}
    23                                                            almostfull_cp:    coverpoint F_cvg_txn.almostfull {bins almostfull_HIGH = {1}; option.weight = 0;}
    24                                                            almostempty_cp:   coverpoint F_cvg_txn.almostempty{bins almostempty_HIGH= {1}; option.weight = 0;}
    25
    26
    27
    28                                                            almostfull_cross:       cross wr_en_cp, rd_en_cp, almostfull_cp;
    29                                                            almostempty_cross:  cross wr_en_cp, rd_en_cp, almostempty_cp;
    30
    31
    32                                                            // empty shouldn't be HIGH if write enable is 1 whatever read
    33                                                            empty_cross:cross wr_en_cp,rd_en_cp,empty_cp iff(F_cvg_txn.rst_n)       {
    34                                    |                     illegal_bins empty_and_wr    =  binsof(wr_en_cp)intersect {1} && (binsof(rd_en_cp) intersect{0}  || binsof(rd_en_cp)    intersect{1}) && binsof(empty_cp) intersect{1};
```

```
                                                  //The class will have an object of the class FIFO_transaction named f_cvg_txn.

                                                  FIFO_transaction f_cvg_txn;

                                                  /* Create a covergroup. The coverage needed is cross coverage between 3 signals which are
                                                    write enable, read enable and each output control signals (outputs except data_out) to
                                                    make sure that all combinations of write and read enable took place in all state of the FIFO */

                                                  covergroup read_write_cg;
                                                        wr_en_cp:          coverpoint f_cvg_txn.wr_en       {option.weight        = 0;}
                                                        rd_en_cp:          coverpoint f_cvg_txn.rd_en       {option.weight        = 0;}
                                                        full_cp:           coverpoint f_cvg_txn.full        {bins full_HIGH       = {1}; option.weight = 0;}
                                                        empty_cp:          coverpoint f_cvg_txn.empty       {bins empty_HIGH      = {1}; option.weight = 0;}
                                                        overflow_cp:       coverpoint f_cvg_txn.overflow    {bins overflow_HIGH   = {1}; option.weight = 0;}
                                                        underflow_cp:      coverpoint f_cvg_txn.underflow   {bins underflow_HIGH  = {1}; option.weight = 0;}
                                                        wr_ack_cp:         coverpoint f_cvg_txn.wr_ack      {bins wr_ack_HIGH     = {1}; option.weight = 0;}
                                                        almostfull_cp:     coverpoint f_cvg_txn.almostfull  {bins almostfull_HIGH = {1}; option.weight = 0;}
                                                        almostempty_cp:    coverpoint f_cvg_txn.almostempty {bins almostempty_HIGH = {1}; option.weight = 0;}


                                                        almostfull_cross:       cross wr_en_cp, rd_en_cp, almostfull_cp;
                                                        almostempty_cross:  cross wr_en_cp, rd_en_cp, almostempty_cp;


                                                        // empty shouldn't be HIGH if write enable is 1 whatever read
                                                        empty_cross:cross wr_en_cp,rd_en_cp,empty_cp iff(f_cvg_txn.rst_n)       {
                                                    illegal_bins empty_and_wr      =  binsof(wr_en_cp)intersect {1} && (binsof(rd_en_cp) intersect{0}  || binsof(rd_en_cp)     intersect{1}) && binsof(empty_cp) intersect{1};
                                                  }
                                                        // full shouldn't be HIGH if read enable is 1 whatever write
                                                        full_cross:cross wr_en_cp , rd_en_cp , full_cp        {
                                                    illegal_bins full_wr_rd       = (binsof(wr_en_cp)intersect {0} || binsof(wr_en_cp) intersect {1}) && binsof(rd_en_cp)     intersect{1} &&  binsof (full_cp) intersect{1};
                                                  }
                                                        // overflow shouldn't be HIGH if write enable is 0
                                                        overflow_cross:cross wr_en_cp ,rd_en_cp, overflow_cp {
                                                    illegal_bins wr_and_over       = binsof(wr_en_cp)intersect {0} && binsof(overflow_cp)  intersect{1};
                                                  }
                                                        // underflow shouldn't be HIGH if read enable is 0
                                                        underflow_cross:cross wr_en_cp,rd_en_cp,underflow_cp {
                                                    illegal_bins underflow_and_rd = binsof(rd_en_cp)intersect {0} && binsof(underflow_cp) intersect{1};
                                                  }
                                                        // write ack shouldn't be HIGH if write enable is 0
                                                        wr_ack_cross:cross wr_en_cp , rd_en_cp , wr_ack_cp  {
                                                    illegal_bins wr_and_wr_ack     = binsof(wr_en_cp)intersect {0} && binsof(wr_ack_cp)    intersect{1};
                                                  }
                                                  endgroup

                                                  /*Create a void function inside it named sample_data that takes one input named f_txn.
                                                  This input is an object of class FIFO_transaction. This function will do the following
                                                  1. Assign f_txn to f_cvg_txn
                                                  2. Trigger the sampling of the covergroup using the .sample method*/

                                                  function new();
                                                  // Create an instance of the covergroup
          1                       1               read_write_cg = new();
                                                  endfunction

                                                  function void sample_data(FIFO_transaction f_txn);
          1                   20004                     f_cvg_txn = f_txn;
          1                   20004                     read_write_cg.sample();
```

```
COVERGROUP COVERAGE:
----------------------------------------------------------------------------------------------------------------
Covergroup                                              Metric       Goal       Bins    Status

----------------------------------------------------------------------------------------------------------------
 TYPE /FIFO_coverage_pkg/FIFO_coverage/read_write_cg
                                                      100.00%        100          -    Covered
    covered/total bins:                                    29         29          -
    missing/total bins:                                     0         29          -
    % Hit:                                            100.00%        100          -
    Coverpoint wr_en_cp                               100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      5916          1          -    Covered
        bin auto[1]                                     14088          1          -    Covered
    Coverpoint rd_en_cp                               100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                     14004          1          -    Covered
        bin auto[1]                                      6000          1          -    Covered
    Coverpoint full_cp                                100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
        missing/total bins:                                 0          1          -
        % Hit:                                        100.00%        100          -
        bin full_HIGH                                    8160          1          -    Covered
    Coverpoint empty_cp                               100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
        missing/total bins:                                 0          1          -
        % Hit:                                        100.00%        100          -
        bin empty_HIGH                                    669          1          -    Covered
    Coverpoint overflow_cp                            100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
        missing/total bins:                                 0          1          -
        % Hit:                                        100.00%        100          -
        bin overflow_HIGH                                5645          1          -    Covered
    Coverpoint underflow_cp                           100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
        missing/total bins:                                 0          1          -
        % Hit:                                        100.00%        100          -
        bin underflow_HIGH                                199          1          -    Covered
    Coverpoint wr_ack_cp                              100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
        missing/total bins:                                 0          1          -
        % Hit:                                        100.00%        100          -
        bin wr_ack_HIGH                                  8176          1          -    Covered
    Coverpoint almostfull_cp                          100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
        missing/total bins:                                 0          1          -
        % Hit:                                        100.00%        100          -
        bin almostfull_HIGH                              5288          1          -    Covered
    Coverpoint almostempty_cp                         100.00%        100          -    Covered
        covered/total bins:                                 1          1          -
        missing/total bins:                                 0          1          -
        % Hit:                                        100.00%        100          -
        bin almostempty_HIGH                              907          1          -    Covered
    Cross almostfull_cross                            100.00%        100          -    Covered
        covered/total bins:                                 4          4          -
        missing/total bins:                                 0          4          -
        % Hit:                                        100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],almostfull_HIGH>        2789          1          -    Covered
            bin <auto[0],auto[1],almostfull_HIGH>         704          1          -    Covered
            bin <auto[1],auto[0],almostfull_HIGH>         718          1          -    Covered
            bin <auto[0],auto[0],almostfull_HIGH>        1077          1          -    Covered
    Cross almostempty_cross                           100.00%        100          -    Covered
```

```
        % Hit:                                    100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],almostfull_HIGH>        2789          1          -    Covered
            bin <auto[0],auto[1],almostfull_HIGH>         704          1          -    Covered
            bin <auto[1],auto[0],almostfull_HIGH>         718          1          -    Covered
            bin <auto[0],auto[0],almostfull_HIGH>        1077          1          -    Covered
    Cross almostempty_cross                       100.00%        100          -    Covered
        covered/total bins:                             4          4          -
        missing/total bins:                             0          4          -
        % Hit:                                    100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],almostempty_HIGH>        329          1          -    Covered
            bin <auto[0],auto[1],almostempty_HIGH>         85          1          -    Covered
            bin <auto[1],auto[0],almostempty_HIGH>        317          1          -    Covered
            bin <auto[0],auto[0],almostempty_HIGH>        176          1          -    Covered
    Cross empty_cross                             100.00%        100          -    Covered
        covered/total bins:                             2          2          -
        missing/total bins:                             0          2          -
        % Hit:                                    100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[0],auto[1],empty_HIGH>          135          1          -    Covered
            bin <auto[0],auto[0],empty_HIGH>          141          1          -    Covered
        Illegal and Ignore Bins:
            illegal_bin empty_and_wr                    0                     -    ZERO
    Cross full_cross                              100.00%        100          -    Covered
        covered/total bins:                             2          2          -
        missing/total bins:                             0          2          -
        % Hit:                                    100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[0],full_HIGH>          6511          1          -    Covered
            bin <auto[0],auto[0],full_HIGH>          1649          1          -    Covered
        Illegal and Ignore Bins:
            illegal_bin full_wr_rd                      0                     -    ZERO
    Cross overflow_cross                          100.00%        100          -    Covered
        covered/total bins:                             2          2          -
        missing/total bins:                             0          2          -
        % Hit:                                    100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],overflow_HIGH>          1704          1          -    Covered
            bin <auto[1],auto[0],overflow_HIGH>          3941          1          -    Covered
        Illegal and Ignore Bins:
            illegal_bin wr_and_over                     0                     -    ZERO
    Cross underflow_cross                         100.00%        100          -    Covered
        covered/total bins:                             2          2          -
        missing/total bins:                             0          2          -
        % Hit:                                    100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],underflow_HIGH>          140          1          -    Covered
            bin <auto[0],auto[1],underflow_HIGH>           59          1          -    Covered
        Illegal and Ignore Bins:
            illegal_bin underflow_and_rd                0                     -    ZERO
    Cross wr_ack_cross                            100.00%        100          -    Covered
        covered/total bins:                             2          2          -
        missing/total bins:                             0          2          -
        % Hit:                                    100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],wr_ack_HIGH>          2417          1          -    Covered
            bin <auto[1],auto[0],wr_ack_HIGH>          5759          1          -    Covered
        Illegal and Ignore Bins:
            illegal_bin wr_and_wr_ack                   0                     -    ZERO

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1
```

```
/FIFO_top/dut/full_cover_1              FIFO   Verilog  SVA  FIFO.sv(127)    7998 Covered
/FIFO_top/dut/empty_cover_1             FIFO   Verilog  SVA  FIFO.sv(128)    1050 Covered
/FIFO_top/dut/almostfull_cover_1        FIFO   Verilog  SVA  FIFO.sv(129)    5182 Covered
/FIFO_top/dut/almostempty_cover_1       FIFO   Verilog  SVA  FIFO.sv(130)     896 Covered
/FIFO_top/dut/full_cover_2              FIFO   Verilog  SVA  FIFO.sv(136)   12005 Covered
/FIFO_top/dut/empty_cover_2             FIFO   Verilog  SVA  FIFO.sv(137)   18953 Covered
/FIFO_top/dut/almostfull_cover_2        FIFO   Verilog  SVA  FIFO.sv(138)   14821 Covered
/FIFO_top/dut/almostempty_cover_2       FIFO   Verilog  SVA  FIFO.sv(139)   19107 Covered
/FIFO_top/dut/rd_count_cover            FIFO   Verilog  SVA  FIFO.sv(147)    1650 Covered
/FIFO_top/dut/wr_count_cover            FIFO   Verilog  SVA  FIFO.sv(148)    5662 Covered
/FIFO_top/dut/rd_wr_count_cover         FIFO   Verilog  SVA  FIFO.sv(149)    2237 Covered
/FIFO_top/dut/rd_ptr_cover              FIFO   Verilog  SVA  FIFO.sv(156)    5561 Covered
/FIFO_top/dut/wr_ptr_cover              FIFO   Verilog  SVA  FIFO.sv(157)    8036 Covered
/FIFO_top/dut/almostfull_full_cover     FIFO   Verilog  SVA  FIFO.sv(169)    2525 Covered
/FIFO_top/dut/almostempty_empty_cover   FIFO   Verilog  SVA  FIFO.sv(170)      75 Covered
/FIFO_top/dut/rst_count_cover           FIFO   Verilog  SVA  FIFO.sv(186)     385 Covered
/FIFO_top/dut/rst_rd_ptr_cover          FIFO   Verilog  SVA  FIFO.sv(187)     385 Covered
/FIFO_top/dut/rst_wr_ptr_cover          FIFO   Verilog  SVA  FIFO.sv(188)     385 Covered
/FIFO_top/dut/rst_full_cover            FIFO   Verilog  SVA  FIFO.sv(189)     385 Covered
/FIFO_top/dut/rst_empty_cover           FIFO   Verilog  SVA  FIFO.sv(190)     385 Covered
/FIFO_top/dut/rst_almostfull_cover      FIFO   Verilog  SVA  FIFO.sv(191)     385 Covered
/FIFO_top/dut/rst_almostempty_cover     FIFO   Verilog  SVA  FIFO.sv(192)     385 Covered
/FIFO_top/dut/rst_data_out_cover        FIFO   Verilog  SVA  FIFO.sv(193)     385 Covered
/FIFO_top/dut/rst_overflow_cover        FIFO   Verilog  SVA  FIFO.sv(194)     385 Covered
/FIFO_top/dut/rst_underflow_cover       FIFO   Verilog  SVA  FIFO.sv(195)     385 Covered
/FIFO_top/dut/rst_wr_ack_cover          FIFO   Verilog  SVA  FIFO.sv(196)     385 Covered

TOTAL DIRECTIVE COVERAGE: 100.00%  COVERS: 32

ASSERTION RESULTS:
--------------------------------------------------------------------
Name                 File(Line)                   Failure      Pass
                                                  Count        Count
--------------------------------------------------------------------
/FIFO_top/dut/overflow_assert_1
                     FIFO.sv(107)                    0          1
/FIFO_top/dut/underflow_assert_1
                     FIFO.sv(108)                    0          1
/FIFO_top/dut/wr_ack_assert_1
                     FIFO.sv(109)                    0          1
/FIFO_top/dut/overflow_assert_2
                     FIFO.sv(114)                    0          1
/FIFO_top/dut/underflow_assert_2
                     FIFO.sv(115)                    0          1
/FIFO_top/dut/wr_ack_assert_2
                     FIFO.sv(116)                    0          1
/FIFO_top/dut/full_assert_1
                     FIFO.sv(123)                    0          1
/FIFO_top/dut/empty_assert_1
                     FIFO.sv(124)                    0          1
/FIFO_top/dut/almostfull_assert_1
                     FIFO.sv(125)                    0          1
/FIFO_top/dut/almostempty_assert_1
                     FIFO.sv(126)                    0          1
/FIFO_top/dut/full_assert_2
                     FIFO.sv(132)                    0          1
/FIFO_top/dut/empty_assert_2
                     FIFO.sv(133)                    0          1
/FIFO_top/dut/almostfull_assert_2
                     FIFO.sv(134)                    0          1
/FIFO_top/dut/almostempty_assert_2
                     FIFO.sv(135)                    0          1
/FIFO_top/dut/rd_count_assert
                     FIFO.sv(144)                    0          1
/FIFO_top/dut/wr_count_assert
                     FIFO.sv(145)                    0          1
/FIFO_top/dut/rd_wr_count_assert
                     FIFO.sv(146)                    0          1
/FIFO_top/dut/rd_ptr_assert
                     FIFO.sv(154)                    0          1
/FIFO_top/dut/wr_ptr_assert
                     FIFO.sv(155)                    0          1
/FIFO_top/dut/almostfull_full_assert
                     FIFO.sv(164)                    0          1
/FIFO_top/dut/almostempty_empty_assert
                     FIFO.sv(167)                    0          1
/FIFO_top/dut/rst_count_assert
                     FIFO.sv(175)                    0          1
/FIFO_top/dut/rst_rd_ptr_assert
                     FIFO.sv(176)                    0          1
/FIFO_top/dut/rst_wr_ptr_assert
                     FIFO.sv(177)                    0          1
/FIFO_top/dut/rst_full_assert
                     FIFO.sv(178)                    0          1
/FIFO_top/dut/rst_empty_assert
                     FIFO.sv(179)                    0          1
/FIFO_top/dut/rst_almostfull_assert
                     FIFO.sv(180)                    0          1
/FIFO_top/dut/rst_almostempty_assert
                     FIFO.sv(181)                    0          1
```

```
ASSERTION RESULTS:
--------------------------------------------------------------------
Name                File(Line)                  Failure      Pass
                                                Count        Count
--------------------------------------------------------------------
/FIFO_top/dut/overflow_assert_1
                    FIFO.sv(107)                      0          1
/FIFO_top/dut/underflow_assert_1
                    FIFO.sv(108)                      0          1
/FIFO_top/dut/wr_ack_assert_1
                    FIFO.sv(109)                      0          1
/FIFO_top/dut/overflow_assert_2
                    FIFO.sv(114)                      0          1
/FIFO_top/dut/underflow_assert_2
                    FIFO.sv(115)                      0          1
/FIFO_top/dut/wr_ack_assert_2
                    FIFO.sv(116)                      0          1
/FIFO_top/dut/full_assert_1
                    FIFO.sv(123)                      0          1
/FIFO_top/dut/empty_assert_1
                    FIFO.sv(124)                      0          1
/FIFO_top/dut/almostfull_assert_1
                    FIFO.sv(125)                      0          1
/FIFO_top/dut/almostempty_assert_1
                    FIFO.sv(126)                      0          1
/FIFO_top/dut/full_assert_2
                    FIFO.sv(132)                      0          1
/FIFO_top/dut/empty_assert_2
                    FIFO.sv(133)                      0          1
/FIFO_top/dut/almostfull_assert_2
                    FIFO.sv(134)                      0          1
/FIFO_top/dut/almostempty_assert_2
                    FIFO.sv(135)                      0          1
/FIFO_top/dut/rd_count_assert
                    FIFO.sv(144)                      0          1
/FIFO_top/dut/wr_count_assert
                    FIFO.sv(145)                      0          1
/FIFO_top/dut/rd_wr_count_assert
                    FIFO.sv(146)                      0          1
/FIFO_top/dut/rd_ptr_assert
                    FIFO.sv(154)                      0          1
/FIFO_top/dut/wr_ptr_assert
                    FIFO.sv(155)                      0          1
/FIFO_top/dut/almostfull_full_assert
                    FIFO.sv(164)                      0          1
/FIFO_top/dut/almostempty_empty_assert
                    FIFO.sv(167)                      0          1
/FIFO_top/dut/rst_count_assert
                    FIFO.sv(175)                      0          1
/FIFO_top/dut/rst_rd_ptr_assert
                    FIFO.sv(176)                      0          1
/FIFO_top/dut/rst_wr_ptr_assert
                    FIFO.sv(177)                      0          1
/FIFO_top/dut/rst_full_assert
                    FIFO.sv(178)                      0          1
/FIFO_top/dut/rst_empty_assert
                    FIFO.sv(179)                      0          1
/FIFO_top/dut/rst_almostfull_assert
                    FIFO.sv(180)                      0          1
/FIFO_top/dut/rst_almostempty_assert
                    FIFO.sv(181)                      0          1
/FIFO_top/dut/rst_data_out_assert
                    FIFO.sv(182)                      0          1
/FIFO_top/dut/rst_overflow_assert
                    FIFO.sv(183)                      0          1
/FIFO_top/dut/rst_underflow_assert
                    FIFO.sv(184)                      0          1
/FIFO_top/dut/rst_wr_ack_assert
                    FIFO.sv(185)                      0          1
/FIFO_top/tb/#ublk#182146786#17/immed__18
                    FIFO_tb.sv(18)                    0          1
```

THANK YOU