FIFO

SVA AND 3 PACKAGE

By:

Mohammed Anwar

Abd allatif

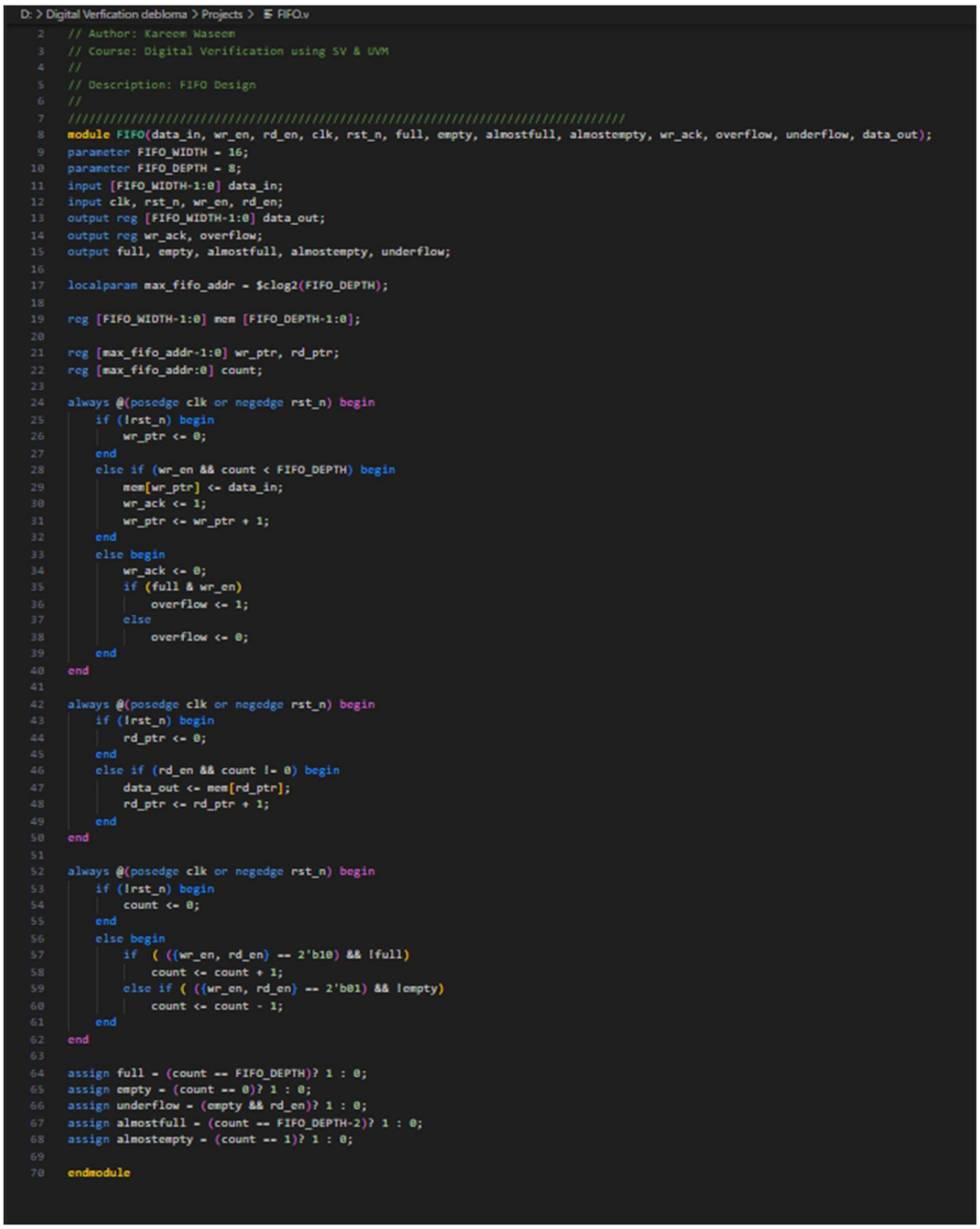
Under supervision of Eng. Kareem Waseem

Design has 6 bugs:

1. Reset signals overflow, wr_ack and underflow (data_out not to be included)
2. Unhandled 2 cases:
    - If a read and write enables were high and the FIFO was empty, only writing will take place.
    - If a read and write enables were high and the FIFO was full, only reading will take place.
3. underflow is sequential not combinational.
4. almostfull flag: FIFO_DEPTH-2 corrected to FIFO_DEPTH-1.
5. When successful write operation has occured, overflow can't be high.

6. When successful read operation has occured, underflow can't be high.

RTL DESIGN BEFORE BUGS :



```
D: > Digital Verfication debloma > Projects > FIFO.v
// Author: Kareem Waseem
// Course: Digital Verification using SV & UVM
//
// Description: FIFO Design
//
////////////////////////////////////////////////////////////////////////////////
module FIFO(data_in, wr_en, rd_en, clk, rst_n, full, empty, almostfull, almostempty, wr_ack, overflow, underflow, data_out);
parameter FIFO_WIDTH = 16;
parameter FIFO_DEPTH = 8;
input [FIFO_WIDTH-1:0] data_in;
input clk, rst_n, wr_en, rd_en;
output reg [FIFO_WIDTH-1:0] data_out;
output reg wr_ack, overflow;
output full, empty, almostfull, almostempty, underflow;

localparam max_fifo_addr = $clog2(FIFO_DEPTH);

reg [FIFO_WIDTH-1:0] mem [FIFO_DEPTH-1:0];

reg [max_fifo_addr-1:0] wr_ptr, rd_ptr;
reg [max_fifo_addr:0] count;

always @(posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        wr_ptr <= 0;
    end
    else if (wr_en && count < FIFO_DEPTH) begin
        mem[wr_ptr] <= data_in;
        wr_ack <= 1;
        wr_ptr <= wr_ptr + 1;
    end
    else begin
        wr_ack <= 0;
        if (full & wr_en)
            overflow <= 1;
        else
            overflow <= 0;
    end
end

always @(posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        rd_ptr <= 0;
    end
    else if (rd_en && count != 0) begin
        data_out <= mem[rd_ptr];
        rd_ptr <= rd_ptr + 1;
    end
end

always @(posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        count <= 0;
    end
    else begin
        if  ( ((wr_en, rd_en) == 2'b10) && !full)
            count <= count + 1;
        else if ( ((wr_en, rd_en) == 2'b01) && !empty)
            count <= count - 1;
    end
end

assign full = (count == FIFO_DEPTH)? 1 : 0;
assign empty = (count == 0)? 1 : 0;
assign underflow = (empty && rd_en)? 1 : 0;
assign almostfull = (count == FIFO_DEPTH-2)? 1 : 0;
assign almostempty = (count == 1)? 1 : 0;

endmodule
```

RTL DESIGN after bugs and interface :

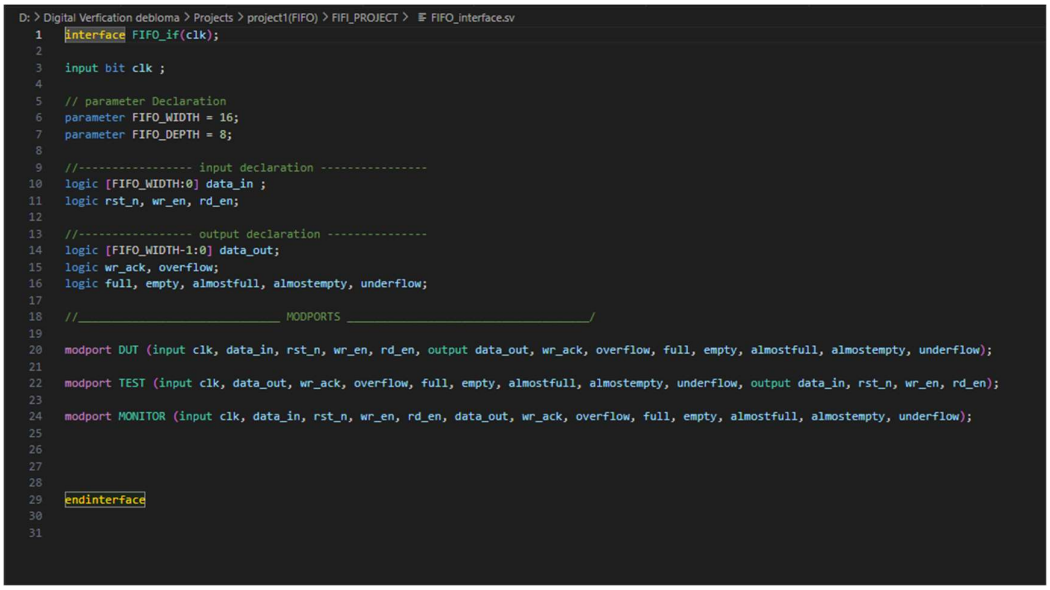
```
module FIFO(FIFO_if.DUT if_obj);

// declaration of max. FIFO address
localparam max_fifo_addr = $clog2(if_obj.FIFO_DEPTH);   // max_fifo_addr = 3

// declaration of Memory (FIFO)
reg [if_obj.FIFO_WIDTH-1:0] mem [if_obj.FIFO_DEPTH-1:0];

// Declaration of read & write pointers
reg [max_fifo_addr-1:0] wr_ptr, rd_ptr;
reg [max_fifo_addr:0] count;              // extra bit to distinguish between full & empty flags & it represents the fill level of the FIFO

// always block specialized for writing operation
always @(posedge if_obj.clk or negedge if_obj.rst_n) begin
    if (!if_obj.rst_n) begin
        wr_ptr <= 0;
        if_obj.overflow <= 0;
        if_obj.wr_ack <= 0;             // reset the sequential outputs as wr_ack , overflow
    end
    else if (if_obj.wr_en && count < if_obj.FIFO_DEPTH) begin
        mem[wr_ptr] <= if_obj.data_in;
        if_obj.wr_ack <= 1;
        wr_ptr <= wr_ptr + 1;
        if_obj.overflow <=0 ;
    end
    else begin
        if_obj.wr_ack <= 0;
        if (if_obj.full && if_obj.wr_en)
            if_obj.overflow <= 1;
        else
            if_obj.overflow <= 0;
    end
end

// always block specialized for reading operation
always @(posedge if_obj.clk or negedge if_obj.rst_n) begin
    if (!if_obj.rst_n) begin
        rd_ptr <= 0;
        if_obj.underflow <= 0;  // reset the sequential outputs as data_out , underflow

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

// always block specialized for counter signal
always @(posedge if_obj.clk or negedge if_obj.rst_n) begin
    if (!if_obj.rst_n) begin
        count <= 0;
    end
    else begin
        if  ( ({if_obj.wr_en, if_obj.rd_en} == 2'b10) && !if_obj.full)
            count <= count + 1;
        else if ( ({if_obj.wr_en, if_obj.rd_en} == 2'b01) && !if_obj.empty)
            count <= count - 1;
        else if (({if_obj.wr_en, if_obj.rd_en} == 2'b11) && if_obj.full)      // priority for write operation
            count <= count - 1;
        else if (({if_obj.wr_en, if_obj.rd_en} == 2'b11) && if_obj.empty)      // priority for read operation
            count <= count + 1;
    end
end
```

```
        else if ({if_obj.wr_en, if_obj.rd_en} == 2'b11) && if_obj.empty)      // priority for read operation
            count <= count + 1;
    end
end

// continous assignment for the combinational outputs
assign if_obj.full = (count == if_obj.FIFO_DEPTH)? 1 : 0;
assign if_obj.empty = (count == 0)? 1 : 0;
assign if_obj.almostfull = (count == if_obj.FIFO_DEPTH-1)? 1 : 0;
assign if_obj.almostempty = (count == 1)? 1 : 0;

//=============================================================
// Assertions + Cover Properties
//=============================================================
//`ifdef ASSERT_ON

// SVA 1 - Reset behavior
property p_reset;
  @(posedge if_obj.clk) !if_obj.rst_n |=> (wr_ptr == 0 && rd_ptr == 0 && count == 0);
endproperty
a1: assert property(p_reset);
c1: cover property(p_reset);

// SVA 2 - Write ACK
property p_wr_ack;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (if_obj.wr_en && !if_obj.full) |=> if_obj.wr_ack;
endproperty
a2: assert property(p_wr_ack);
c2: cover property(p_wr_ack);

// SVA 3 - Overflow
property p_overflow;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (if_obj.wr_en && if_obj.full) |=> if_obj.overflow;
endproperty
a3: assert property(p_overflow);
c3: cover property(p_overflow);

// SVA 4 - Underflow
property p_underflow;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (if_obj.rd_en && if_obj.empty) |=> if_obj.underflow;
endproperty
a4: assert property(p_underflow);
c4: cover property(p_underflow);

// SVA 5 - Empty flag
property p_empty;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (count == 0) |-> if_obj.empty;
endproperty
a5: assert property(p_empty);
c5: cover property(p_empty);
```

```
120
121   // SVA 6 - Full flag
122   property p_full;
123     @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
124       (count == if_obj.FIFO_DEPTH) |-> if_obj.full;
125   endproperty
126   a6: assert property(p_full);
127   c6: cover property(p_full);
128
129   // SVA 7 - Almost empty
130   property p_almostempty;
131     @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
132       (count == if_obj.FIFO_DEPTH - 1) |-> if_obj.almostfull;
133   endproperty
134   a7: assert property(p_almostempty);
135   c7: cover property(p_almostempty);
136
137   // SVA 8 - Almost full
138   property p_almostfull;
139     @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
140       (count == 1) |-> if_obj.almostempty;
141   endproperty
142   a8: assert property(p_almostfull);
143   c8: cover property(p_almostfull);
144
145   // SVA 9 - Counter increases when both en = 1 and not full
146   property p_count_inc;
147     @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
148       (!if_obj.full && if_obj.wr_en && !if_obj.rd_en) |=> (count == $past(count) + 1);
149   endproperty
150   a9: assert property(p_count_inc);
151   c9: cover property(p_count_inc);
152
153   // SVA 10 - Counter decreases when both en = 1 and not empty
154   property p_count_dec;
155     @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
156       (!if_obj.empty && !if_obj.wr_en && if_obj.rd_en) |=> (count == $past(count) - 1);
157   endproperty
158   a10: assert property(p_count_dec);
159   c10: cover property(p_count_dec);
160
161   // SVA 11 - Pointer and counter limits
162   property p_limit;
163     @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
164       (wr_ptr < if_obj.FIFO_DEPTH && rd_ptr < if_obj.FIFO_DEPTH && count <= if_obj.FIFO_DEPTH);
165   endproperty
166   a11: assert property(p_limit);
167   c11: cover property(p_limit);
168
169   //`endif
170
171   endmodule
```

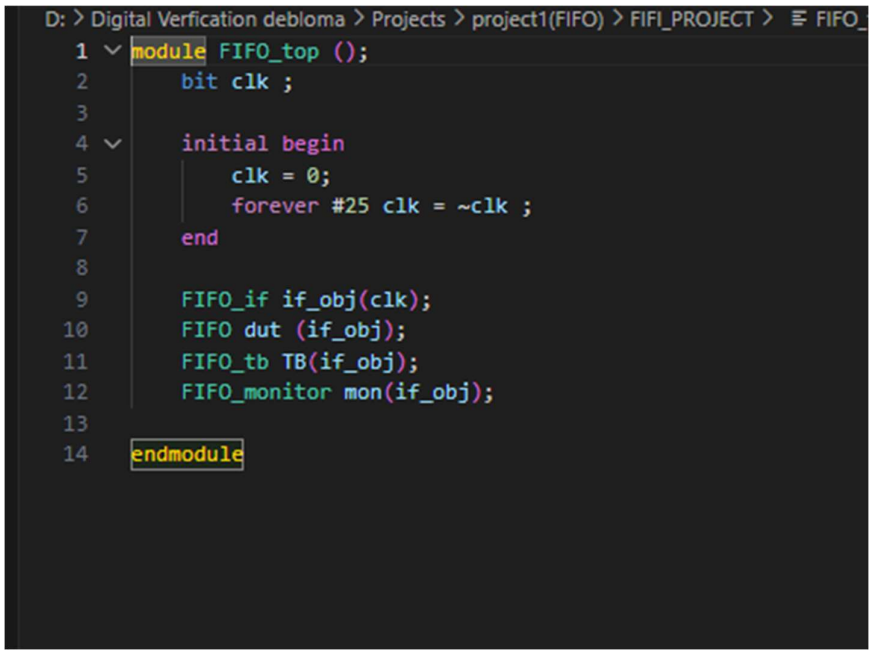
⊗ 0 ⚠ 0

Interface :



```
D: > Digital Verfication debloma > Projects > project1(FIFO) > FIFI_PROJECT > FIFO_interface.sv
1   interface FIFO_if(clk);
2
3   input bit clk ;
4
5   // parameter Declaration
6   parameter FIFO_WIDTH = 16;
7   parameter FIFO_DEPTH = 8;
8
9   //---------------- input declaration ---------------
10  logic [FIFO_WIDTH:0] data_in ;
11  logic rst_n, wr_en, rd_en;
12
13  //---------------- output declaration --------------
14  logic [FIFO_WIDTH-1:0] data_out;
15  logic wr_ack, overflow;
16  logic full, empty, almostfull, almostempty, underflow;
17
18  //_____________________________ MODPORTS ____________________________________/
19
20  modport DUT (input clk, data_in, rst_n, wr_en, rd_en, output data_out, wr_ack, overflow, full, empty, almostfull, almostempty, underflow);
21
22  modport TEST (input clk, data_out, wr_ack, overflow, full, empty, almostfull, almostempty, underflow, output data_in, rst_n, wr_en, rd_en);
23
24  modport MONITOR (input clk, data_in, rst_n, wr_en, rd_en, data_out, wr_ack, overflow, full, empty, almostfull, almostempty, underflow);
25
26
27
28
29  endinterface
30
31
```

Top module :



```
D: > Digital Verfication debloma > Projects > project1(FIFO) > FIFI_PROJECT > FIFO_
1   module FIFO_top ();
2       bit clk ;
3
4       initial begin
5           clk = 0;
6           forever #25 clk = ~clk ;
7       end
8
9       FIFO_if if_obj(clk);
10      FIFO dut (if_obj);
11      FIFO_tb TB(if_obj);
12      FIFO_monitor mon(if_obj);
13
14  endmodule
```

FIFO_TRANSACTION_pkg:

```
package FIFO_transaction_pkg;

class FIFO_transaction;
  parameter FIFO_WIDTH = 16;
  parameter FIFO_DEPTH = 8;

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

  // Control read/write enable bias (sum should not exceed 100)
  int RD_EN_ON_DIST = 30;
  int WR_EN_ON_DIST = 70;

  /********** Constraints **********/

  // Reset active-low, mostly deasserted
  constraint c_reset {
    rst_n dist {0:/2, 1:/98};
  }

  // Write enable probability
  constraint c_write {
    wr_en dist {0 := (100 - WR_EN_ON_DIST), 1 := WR_EN_ON_DIST};
  }

  // Read enable — same structure as wr_en, but using RD_EN_ON_DIST
  constraint c_read {
    rd_en dist {0 := (100 - RD_EN_ON_DIST), 1 := RD_EN_ON_DIST};
  }

 // Additional constraint block for write operation only
constraint write_only {rst_n == 1 ; wr_en == 1 ; rd_en == 0; }

// Additional constraint block for read operation only

constraint read_only {rst_n == 1 ; wr_en == 0 ; rd_en == 1; }




endclass

endpackage

```

FIFO_COVERAGE_PKG :

D: > Digital Verfication debloma > Projects > project1(FIFO) > FIFI_PROJECT > FIFO_coverage_pkg.sv

```
// FIFO Package 2
package FIFO_coverage_pkg ;

import FIFO_transaction_pkg ::*;

class FIFO_coverage;

FIFO_transaction F_cvg_txn ;

function void sample_data (input FIFO_transaction F_txn );

F_cvg_txn = F_txn ;

cg.sample();

endfunction

covergroup cg ;

// Here we create cover point for each signal to be used in the cross coverage
wr_en_cp : coverpoint F_cvg_txn.wr_en;
rd_en_cp : coverpoint F_cvg_txn.rd_en;
wr_ack_cp : coverpoint F_cvg_txn.wr_ack;
full_cp : coverpoint F_cvg_txn.full;
empty_cp : coverpoint F_cvg_txn.empty;
almostfull_cp : coverpoint F_cvg_txn.almostfull;
almostempty_cp : coverpoint F_cvg_txn.almostempty;
overflow_cp : coverpoint F_cvg_txn.overflow;
underflow_cp : coverpoint F_cvg_txn.underflow;

// Here we create cross coverage for each output with read enable & write enable except(dout)
wr_rd_wr_ack_cross : cross wr_en_cp , rd_en_cp , wr_ack_cp
        {
            ignore_bins write_active_with_wr_ack = ! binsof(wr_en_cp) intersect {1} && binsof(wr_ack_cp) intersect {1};
            ignore_bins read_write_active_with_wr_ack = ! binsof(wr_en_cp) intersect {1} && binsof(rd_en_cp) intersect {1} && binsof(wr_ack_cp) intersect {1};
        }

wr_rd_full_cross : cross wr_en_cp , rd_en_cp , full_cp
        {
            ignore_bins write_active_with_full = ! binsof(wr_en_cp) intersect {1} && binsof(full_cp) intersect {1};
            ignore_bins read_active_with_full = binsof(rd_en_cp) intersect {1} && binsof(full_cp) intersect {1};
        }

wr_rd_empty_cross : cross wr_en_cp , rd_en_cp , empty_cp
        {
            ignore_bins read_active_with_empty = ! binsof(rd_en_cp) intersect {1} && binsof(empty_cp) intersect {1};
        }

wr_rd_overflow_cross : cross wr_en_cp , rd_en_cp , overflow_cp
        {
            ignore_bins write_active_with_overflow = ! binsof(wr_en_cp) intersect {1} && binsof(overflow_cp) intersect {1};
        }

wr_rd_underflow_cross : cross wr_en_cp , rd_en_cp , underflow_cp
        {
            ignore_bins read_active_with_underflow = ! binsof(rd_en_cp) intersect {1} && binsof(underflow_cp) intersect {1};
        }

wr_rd_almostfull_cross : cross wr_en_cp , rd_en_cp , almostfull_cp
        {
            ignore_bins write_active_with_almostfull = ! binsof(wr_en_cp) intersect {1} && binsof(almostfull_cp) intersect {1};
            ignore_bins read_write_active_with_almostfull = ! binsof(wr_en_cp) intersect {1} && binsof(rd_en_cp) intersect {1} && binsof(almostfull_cp) intersect {1};
        }

wr_rd_almostempty_cross : cross wr_en_cp , rd_en_cp , almostempty_cp
        {
            ignore_bins read_active_with_almostempty = ! binsof(rd_en_cp) intersect {1} && binsof(almostempty_cp) intersect {1};
            ignore_bins read_write_active_with_almostempty = ! binsof(wr_en_cp) intersect {1} && binsof(rd_en_cp) intersect {1} && binsof(almostempty_cp) intersect {1};
        }

endgroup

function new();
    cg = new ;
    F_cvg_txn = new;
endfunction

endclass

endpackage
```

FIFO_SCOREBOARED_PKG:

D: > Digital Verfication debloma > Projects > project1(FIFO) > FIFI_PROJECT > FIFO_scoreboard_pkg.sv

```
package FIFO_scoreboard_pkg;

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

class FIFO_scoreboard;

  parameter FIFO_WIDTH = 16;
  parameter FIFO_DEPTH = 8;

  bit [FIFO_WIDTH-1:0] data_out_ref;
  bit wr_ack_ref, overflow_ref;
  bit full_ref, empty_ref, almostfull_ref, almostempty_ref, underflow_ref;

  int counter;                          // Tracks number of elements in the queue
  bit [FIFO_WIDTH-1:0] queue[$];        // Reference queue to model FIFO behavior

  FIFO_transaction obj = new();         // Transaction object

  /********************** comb_flags ******************/
  function void comb_flags();
    full_ref       = (counter == FIFO_DEPTH)?1:0;
    empty_ref      = (counter == 0)?1:0;
    almostfull_ref  = (counter == FIFO_DEPTH - 1)?1:0;
    almostempty_ref = (counter == 1)?1:0;
  endfunction

  /********************** check_data ******************/
  function void check_data(input FIFO_transaction obj);
    logic [6:0] flags_ref, flags_dut;

    reference_model(obj);  // Run golden model

    flags_ref = {wr_ack_ref, overflow_ref, full_ref, empty_ref, almostempty_ref, almostfull_ref, underflow_ref};
    flags_dut = {obj.wr_ack, obj.overflow, obj.full, obj.empty, obj.almostempty, obj.almostfull, obj.underflow};

    if ((obj.data_out !== data_out_ref) || (flags_dut !== flags_ref)) begin
      $display("❌ MISMATCH at time = %0t: DUT outputs don't match reference model.", $time);
      error_count++;
    end else begin
      correct_count++;
      $display("✅ MATCH at time = %0t: Current queue = %p", $time, queue);
    end
  endfunction
```

```
  /********************** reference_model ******************/
  function void reference_model(input FIFO_transaction obj_gold);

    fork
      // Write thread
      begin
        if (!obj_gold.rst_n) begin
          wr_ack_ref      = 0;
          overflow_ref    = 0;
          queue.delete();
        end else if (obj_gold.wr_en && counter < FIFO_DEPTH) begin
          queue.push_back(obj_gold.data_in);
          wr_ack_ref = 1;
        end else begin
          wr_ack_ref = 0;
          if (full_ref && obj_gold.wr_en)
            overflow_ref = 1;
          else
            overflow_ref = 0;
        end
      end

      // Read thread
      begin
        if (!obj_gold.rst_n) begin
          data_out_ref      = 0;
          underflow_ref     = 0;
          queue.delete();
        end else if (obj_gold.rd_en && counter > 0) begin
          data_out_ref = queue.pop_front();
          underflow_ref = 0;
        end else begin
          if (empty_ref && obj_gold.rd_en)
            underflow_ref = 1;
          else
            underflow_ref = 0;
        end
      end
    join

    // Update counter
    if (!obj_gold.rst_n) begin
      counter = 0;
    end else begin
      case ({obj_gold.wr_en, obj_gold.rd_en})
        2'b10: if (!full_ref)  counter++;
        2'b01: if (!empty_ref) counter--;
        2'b11: begin
          if (full_ref && !empty_ref) counter--; // writing when full, read takes priority
          else if (empty_ref && !full_ref) counter++;
          // else no change when full & empty are both 0 (balanced ops)
        end
```

```
    // Update counter
    if (!obj_gold.rst_n) begin
      counter = 0;
    end else begin
      case ({obj_gold.wr_en, obj_gold.rd_en})
        2'b10: if (!full_ref)  counter++;
        2'b01: if (!empty_ref) counter--;
        2'b11: begin
          if (full_ref && !empty_ref) counter--; // writing when full, read takes priority
          else if (empty_ref && !full_ref) counter++;
          // else no change when full & empty are both 0 (balanced ops)
        end
      endcase
    end

    comb_flags();  // Update status flags after counter change

  endfunction

endclass

endpackage

```

SHARED_pkg:

D: > Digital Verfication debloma > Projects > project1(FIFO) > FIFI_PROJECT > SHARED_pkg.sv

```
1  // FIFO Package 4
2  package shared_pkg;
3
4  bit test_finished;  // signal refer to the end of the test bench
5
6  // counters declaration
7  int error_count, correct_count;
8
9  endpackage
```

FIFO_Test:

```
import shared_pkg ::*;

import FIFO_transaction_pkg ::*;

module FIFO_tb(FIFO_if.TEST if_obj);

FIFO_transaction obj_test ;       // create obj from the class to randomize it

// we choose these certain values to make the overall iteration = 10,000
localparam MIXED_TESTS = 9933;
localparam WRITE_TESTS = 40;
localparam READ_TESTS = 25;

initial begin
    obj_test = new();

//-------initially reset the FIFO for 2 clk cycles--------
if_obj.rst_n = 0;
repeat(2) @(negedge if_obj.clk);
if_obj.rst_n = 1;

//--------first loop to write only inside the FIFO--------
obj_test.constraint_mode(0);
obj_test.write_only.constraint_mode(1);
repeat(WRITE_TESTS) begin
    assert(obj_test.randomize());
    if_obj.rst_n = obj_test.rst_n ;
    if_obj.rd_en = obj_test.rd_en ;
    if_obj.wr_en = obj_test.wr_en ;
    if_obj.data_in = obj_test.data_in ;
    @(negedge if_obj.clk);
end

//---------Second loop to read only from the FIFO---------
obj_test.constraint_mode(0);
obj_test.read_only.constraint_mode(1);
obj_test.data_in.rand_mode(0);   // we disable the randomization for the data_in as in the read mode we don't need it
repeat(READ_TESTS) begin
    assert(obj_test.randomize());
    if_obj.rst_n = obj_test.rst_n ;
    if_obj.rd_en = obj_test.rd_en ;
    if_obj.wr_en = obj_test.wr_en ;
    if_obj.data_in = obj_test.data_in ;
    @(negedge if_obj.clk);
end

//----------Third loop for mixed operations(read & write) inside FIFO--------
obj_test.constraint_mode(1);
obj_test.data_in.rand_mode(1);   // Again , we enable the data_in for write operation
obj_test.read_only.constraint_mode(0);
obj_test.write_only.constraint_mode(0);
repeat(MIXED_TESTS) begin
    assert(obj_test.randomize());
    if_obj.rst_n = obj_test.rst_n ;
    if_obj.rd_en = obj_test.rd_en ;
    if_obj.wr_en = obj_test.wr_en ;
    if_obj.data_in = obj_test.data_in ;
    @(negedge if_obj.clk);
end

test_finished = 1 ;    // End of the test stimulus

end   // end of the initial block

endmodule
```

FIFO_MONITOR :

D: > Digital Verfication debloma > Projects > project1(FIFO) > FIFI_PROJECT > Monitor.sv

```
import shared_pkg ::*;

import FIFO_transaction_pkg ::*;

import FIFO_coverage_pkg ::*;

import FIFO_scoreboard_pkg ::*;

module FIFO_monitor (FIFO_if.MONITOR if_obj);

// create obj for each class
FIFO_transaction obj_trans ;
FIFO_scoreboard obj_score ;
FIFO_coverage obj_cov ;

initial begin

    obj_trans = new();
    obj_score = new();
    obj_cov = new();

forever begin

    @(negedge if_obj.clk);

    // sample input data
    obj_trans.data_in = if_obj.data_in ;
    obj_trans.rst_n = if_obj.rst_n ;
    obj_trans.wr_en = if_obj.wr_en ;
    obj_trans.rd_en = if_obj.rd_en ;

    // sample output data
    obj_trans.data_out = if_obj.data_out ;
    obj_trans.wr_ack = if_obj.wr_ack ;
    obj_trans.overflow = if_obj.overflow ;
    obj_trans.full = if_obj.full ;
    obj_trans.empty = if_obj.empty ;
    obj_trans.almostfull = if_obj.almostfull ;
    obj_trans.almostempty = if_obj.almostempty ;
    obj_trans.underflow = if_obj.underflow ;

fork

    begin   // 1st thread
        obj_cov.sample_data(obj_trans);
    end

    begin  // 2nd thread
        @(posedge if_obj.clk);
        #10;                    // small delay to check the data after the output change at the posedge of the clock
        obj_score.check_data(obj_trans);
    end

join

if(test_finished == 1) begin
    $display("Final values stored in the queue = %p ",obj_score.queue);
    $display("no.of error_count :%0d ,no.of correct_count :%0d", error_count , correct_count);
    $stop;
end

end // end of the forever loop

end   // end of the initial block


endmodule
```

SVA_IN_DESIGN :

```
// SVA 1 - Reset behavior
property p_reset;
  @(posedge if_obj.clk) !if_obj.rst_n |=> (wr_ptr == 0 && rd_ptr == 0 && count == 0);
endproperty
a1: assert property(p_reset);
c1: cover property(p_reset);

// SVA 2 - Write ACK
property p_wr_ack;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (if_obj.wr_en && !if_obj.full) |=> if_obj.wr_ack;
endproperty
a2: assert property(p_wr_ack);
c2: cover property(p_wr_ack);

// SVA 3 - Overflow
property p_overflow;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (if_obj.wr_en && if_obj.full) |=> if_obj.overflow;
endproperty
a3: assert property(p_overflow);
c3: cover property(p_overflow);

// SVA 4 - Underflow
property p_underflow;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (if_obj.rd_en && if_obj.empty) |=> if_obj.underflow;
endproperty
a4: assert property(p_underflow);
c4: cover property(p_underflow);

// SVA 5 - Empty flag
property p_empty;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (count == 0) |-> if_obj.empty;
endproperty
a5: assert property(p_empty);
c5: cover property(p_empty);

// SVA 6 - Full flag
property p_full;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (count == if_obj.FIFO_DEPTH) |-> if_obj.full;
endproperty
a6: assert property(p_full);
c6: cover property(p_full);

// SVA 7 - Almost empty
property p_almostempty;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (count == if_obj.FIFO_DEPTH - 1) |-> if_obj.almostfull;
endproperty
a7: assert property(p_almostempty);
c7: cover property(p_almostempty);

// SVA 8 - Almost full
property p_almostfull;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (count == 1) |-> if_obj.almostempty;
endproperty
a8: assert property(p_almostfull);
c8: cover property(p_almostfull);

// SVA 9 - Counter increases when both en = 1 and not full
property p_count_inc;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (!if_obj.full && if_obj.wr_en && !if_obj.rd_en) |=> (count == $past(count) + 1);
endproperty
a9: assert property(p_count_inc);
c9: cover property(p_count_inc);

// SVA 10 - Counter decreases when both en = 1 and not empty
property p_count_dec;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (!if_obj.empty && !if_obj.wr_en && if_obj.rd_en) |=> (count == $past(count) - 1);
endproperty
a10: assert property(p_count_dec);
c10: cover property(p_count_dec);

// SVA 11 - Pointer and counter limits
property p_limit;
  @(posedge if_obj.clk) disable iff(!if_obj.rst_n)
    (wr_ptr < if_obj.FIFO_DEPTH && rd_ptr < if_obj.FIFO_DEPTH && count <= if_obj.FIFO_DEPTH);
endproperty
a11: assert property(p_limit);
c11: cover property(p_limit);

//`endif

endmodule
```

DO_FILE :

D: > Digital Verfication debloma > Projects > project1(FIFO) > FIFI_PROJECT > run.do

```
vlib work
vlog FIFO_after.sv FIFO_interface.sv FIFO_top.sv FIFO_transaction_pkg.sv FIFO_coverage_pkg.sv FIFO_scoreboard_pkg.sv SHARED_pkg.sv FIFO_tb.sv Monitor.sv +cover
vsim -voptargs=+acc work.FIFO_top -cover
add wave *
add wave -position insertpoint  \
 sim:/FIFO_top/if_obj/data_in \
 sim:/FIFO_top/if_obj/rst_n \
 sim:/FIFO_top/if_obj/wr_en \
 sim:/FIFO_top/if_obj/rd_en \
 sim:/FIFO_top/if_obj/data_out \
 sim:/FIFO_top/if_obj/wr_ack \
 sim:/FIFO_top/if_obj/overflow \
 sim:/FIFO_top/if_obj/full \
 sim:/FIFO_top/if_obj/empty \
 sim:/FIFO_top/if_obj/almostfull \
 sim:/FIFO_top/if_obj/almostempty \
 sim:/FIFO_top/if_obj/underflow \
 sim:/FIFO_top/monitor/obj_score
add wave /FIFO_top/dut/a1 /FIFO_top/dut/a2 /FIFO_top/dut/a3 /FIFO_top/dut/a4 /FIFO_top/dut/a5 /FIFO_top/dut/a6 /FIFO_top/dut/a7 /FIFO_top/dut/a8 /FIFO_top/dut/a9 /FIFO_top/dut/a10 /FIFO_top/dut/a11
coverage save FIFO_after.ucdb -onexit -du work.FIFO
run -all
```

BUGS DETECTED:

1: at always block of writing operation : reset the sequential as wr_ack , overflow

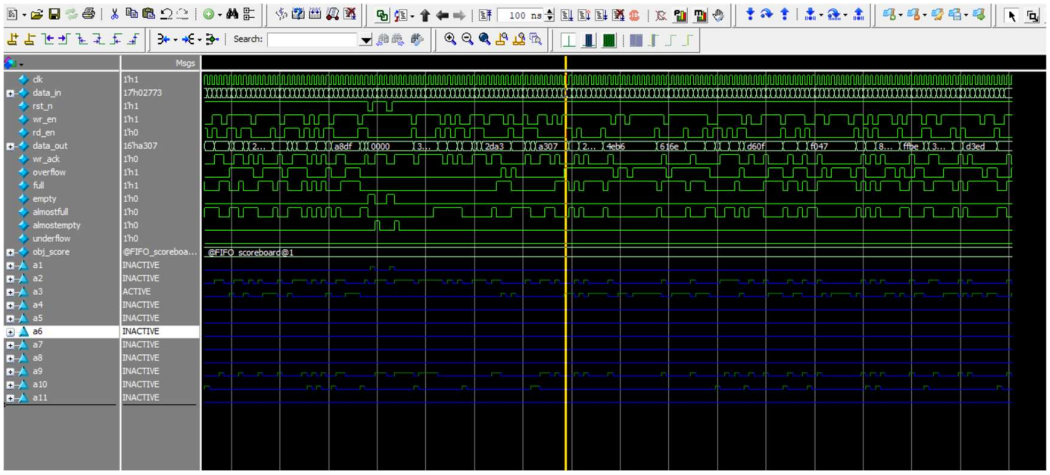
2 : at always block of reading operation : reset the sequential as underflow, data_out

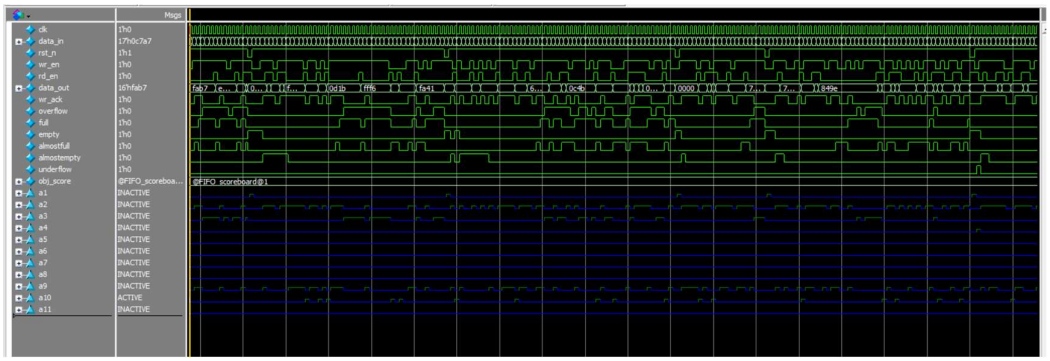
3: at always block of count :

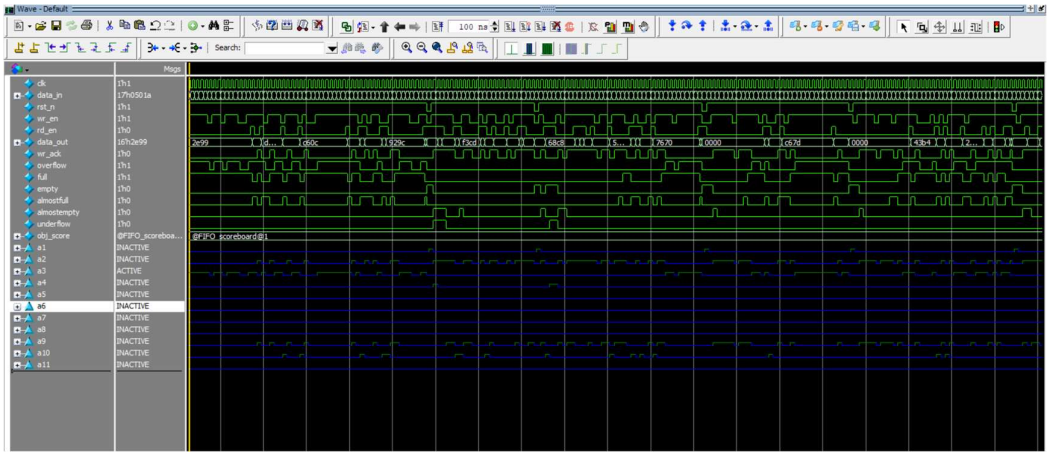
```
            count <= count - 1;
        else if (({if_obj.wr_en, if_obj.rd_en} == 2'b11) && if_obj.full)      // priority for write operation
            count <= count - 1;
        else if (({if_obj.wr_en, if_obj.rd_en} == 2'b11) && if_obj.empty)      // priority for read operation
            count <= count + 1;
    end
end
```

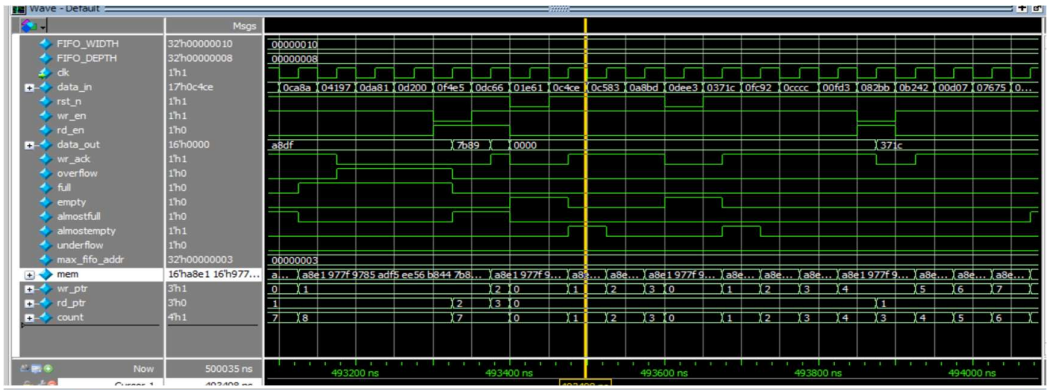
priority for write operation & priority for read operation

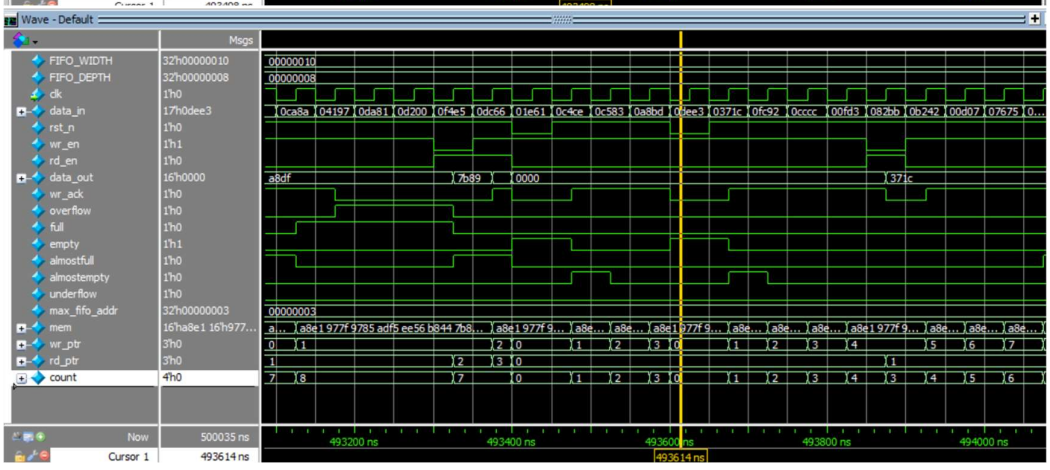
WAVE_FORM :






Wave - Default
Msgs
clk
data_in
rst_n
wr_en
rd_en
data_out
wr_ack
overflow
full
empty
almostfull
almostempty
underflow
obj_score
a1
a2
a3
a4
a5
a6
a7
a8
a9
a10
a11
@FIFO_scoreboard@1


Wave - Default
Msgs
FIFO_WIDTH
FIFO_DEPTH
clk
data_in
rst_n
wr_en
rd_en
data_out
wr_ack
overflow
full
empty
almostfull
almostempty
underflow
max_fifo_addr
mem
wr_ptr
rd_ptr
count
Now
493200 ns
493400 ns
493600 ns
493800 ns
494000 ns


Wave - Default
Msgs
FIFO_WIDTH
FIFO_DEPTH
clk
data_in
rst_n
wr_en
rd_en
data_out
wr_ack
overflow
full
empty
almostfull
almostempty
underflow
max_fifo_addr
mem
wr_ptr
rd_ptr
count
Now
Cursor 1
493200 ns
493400 ns
493600 ns
493800 ns
494000 ns
493614 ns

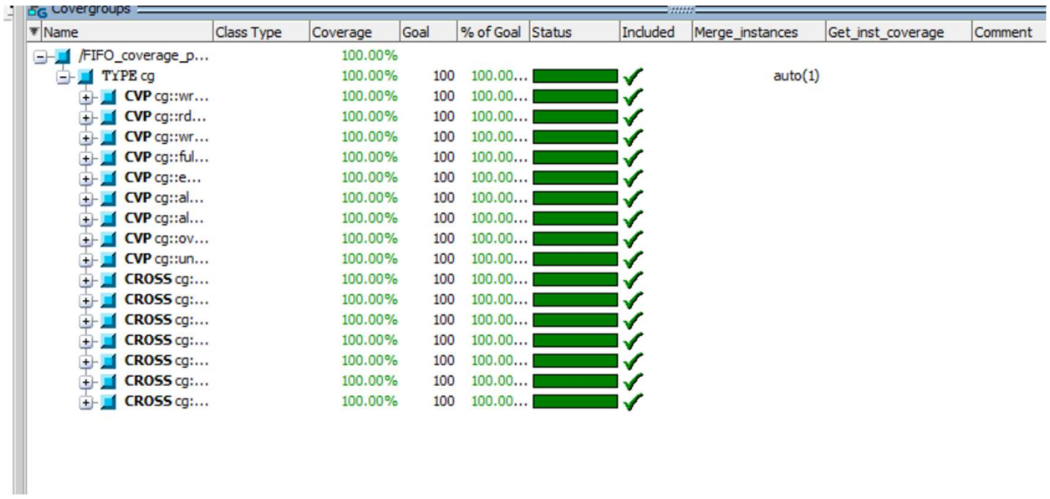
```
# ** MATCH at time = 250435: Current queue = '{61592, 43956, 31545, 7645, 17690}
# ** MATCH at time = 250485: Current queue = '{61592, 43956, 31545, 7645, 17690, 12456}
# ** MATCH at time = 250535: Current queue = '{61592, 43956, 31545, 7645, 17690, 12456, 32547}
# ** MATCH at time = 250585: Current queue = '{61592, 43956, 31545, 7645, 17690, 12456, 32547, 59849}
# ** MATCH at time = 250635: Current queue = '{43956, 31545, 7645, 17690, 12456, 32547, 59849}
# ** MATCH at time = 250685: Current queue = '{31545, 7645, 17690, 12456, 32547, 59849, 21780}
# ** MATCH at time = 250735: Current queue = '{31545, 7645, 17690, 12456, 32547, 59849, 21780}
# ** MATCH at time = 250785: Current queue = '{7645, 17690, 12456, 32547, 59849, 21780, 59978}
# ** MATCH at time = 250835: Current queue = '{7645, 17690, 12456, 32547, 59849, 21780, 59978, 55964}
# ** MATCH at time = 250885: Current queue = '{7645, 17690, 12456, 32547, 59849, 21780, 59978, 55964}
# ** MATCH at time = 250935: Current queue = '{7645, 17690, 12456, 32547, 59849, 21780, 59978, 55964}
# ** MATCH at time = 250985: Current queue = '{7645, 17690, 12456, 32547, 59849, 21780, 59978, 55964}
# ** MATCH at time = 251035: Current queue = '{7645, 17690, 12456, 32547, 59849, 21780, 59978, 55964}
# ** MATCH at time = 251085: Current queue = '{7645, 17690, 12456, 32547, 59849, 21780, 59978, 55964}
# ** MATCH at time = 251135: Current queue = '{17690, 12456, 32547, 59849, 21780, 59978, 55964}
# ** MATCH at time = 251185: Current queue = '{12456, 32547, 59849, 21780, 59978, 55964, 1196}
# ** MATCH at time = 251235: Current queue = '{12456, 32547, 59849, 21780, 59978, 55964, 1196, 22335}
# ** MATCH at time = 251285: Current queue = '{12456, 32547, 59849, 21780, 59978, 55964, 1196, 22335}
# ** MATCH at time = 251335: Current queue = '{12456, 32547, 59849, 21780, 59978, 55964, 1196, 22335}
# ** MATCH at time = 251385: Current queue = '{12456, 32547, 59849, 21780, 59978, 55964, 1196, 22335}
# ** MATCH at time = 251435: Current queue = '{12456, 32547, 59849, 21780, 59978, 55964, 1196, 22335}
# ** MATCH at time = 251485: Current queue = '{12456, 32547, 59849, 21780, 59978, 55964, 1196, 22335}
# ** MATCH at time = 251535: Current queue = '{12456, 32547, 59849, 21780, 59978, 55964, 1196, 22335}
# ** MATCH at time = 251585: Current queue = '{32547, 59849, 21780, 59978, 55964, 1196, 22335}
# ** MATCH at time = 251635: Current queue = '{32547, 59849, 21780, 59978, 55964, 1196, 22335, 28694}
# ** MATCH at time = 251685: Current queue = '{32547, 59849, 21780, 59978, 55964, 1196, 22335, 28694}
# ** MATCH at time = 251735: Current queue = '{32547, 59849, 21780, 59978, 55964, 1196, 22335, 28694}
# ** MATCH at time = 251785: Current queue = '{32547, 59849, 21780, 59978, 55964, 1196, 22335, 28694}
# ** MATCH at time = 251835: Current queue = '{32547, 59849, 21780, 59978, 55964, 1196, 22335, 28694}
# ** MATCH at time = 251885: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694}
# ** MATCH at time = 251935: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694}
# ** MATCH at time = 251985: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252035: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252085: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252135: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252185: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252235: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252285: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252335: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252385: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252435: Current queue = '{59849, 21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252485: Current queue = '{21780, 59978, 55964, 1196, 22335, 28694, 23978}
# ** MATCH at time = 252535: Current queue = '{21780, 59978, 55964, 1196, 22335, 28694, 23978, 40565}
# ** MATCH at time = 252585: Current queue = '{21780, 59978, 55964, 1196, 22335, 28694, 23978, 40565}
# ** MATCH at time = 252635: Current queue = '{21780, 59978, 55964, 1196, 22335, 28694, 23978, 40565}
# ** MATCH at time = 252685: Current queue = '{59978, 55964, 1196, 22335, 28694, 23978, 40565}
# ** MATCH at time = 252735: Current queue = '{59978, 55964, 1196, 22335, 28694, 23978, 40565}
```

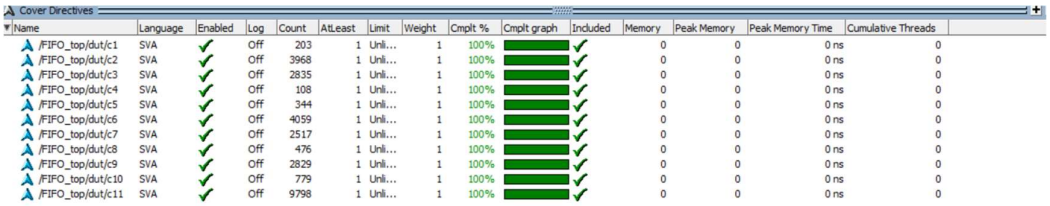
```
# ** MATCH at time = 498035: Current queue = '{13700, 64119, 33989, 65470, 8035, 12984, 30909, 54253}
# ** MATCH at time = 498135: Current queue = '{13700, 64119, 33989, 65470, 8035, 12984, 30909, 54253}
# ** MATCH at time = 498185: Current queue = '{13700, 64119, 33989, 65470, 8035, 12984, 30909, 54253}
# ** MATCH at time = 498235: Current queue = '{13700, 64119, 33989, 65470, 8035, 12984, 30909, 54253}
# ** MATCH at time = 498285: Current queue = '{13700, 64119, 33989, 65470, 8035, 12984, 30909, 54253}
# ** MATCH at time = 498335: Current queue = '{13700, 64119, 33989, 65470, 8035, 12984, 30909, 54253}
# ** MATCH at time = 498385: Current queue = '{13700, 64119, 33989, 65470, 8035, 12984, 30909, 54253}
# ** MATCH at time = 498435: Current queue = '{13700, 64119, 33989, 65470, 8035, 12984, 30909, 54253}
# ** MATCH at time = 498485: Current queue = '{64119, 33989, 65470, 8035, 12984, 30909, 54253}
# ** MATCH at time = 498535: Current queue = '{64119, 33989, 65470, 8035, 12984, 30909, 54253, 38139}
# ** MATCH at time = 498585: Current queue = '{33989, 65470, 8035, 12984, 30909, 54253, 38139}
# ** MATCH at time = 498635: Current queue = '{33989, 65470, 8035, 12984, 30909, 54253, 38139, 40130}
# ** MATCH at time = 498685: Current queue = '{65470, 8035, 12984, 30909, 54253, 38139, 40130}
# ** MATCH at time = 498735: Current queue = '{65470, 8035, 12984, 30909, 54253, 38139, 40130, 10616}
# ** MATCH at time = 498835: Current queue = '{65470, 8035, 12984, 30909, 54253, 38139, 40130, 10616}
# ** MATCH at time = 498885: Current queue = '{65470, 8035, 12984, 30909, 54253, 38139, 40130, 10616}
# ** MATCH at time = 498935: Current queue = '{8035, 12984, 30909, 54253, 38139, 40130, 10616}
# ** MATCH at time = 498985: Current queue = '{8035, 12984, 30909, 54253, 38139, 40130, 10616}
# ** MATCH at time = 499035: Current queue = '{8035, 12984, 30909, 54253, 38139, 40130, 10616, 52816}
# ** MATCH at time = 499085: Current queue = '{8035, 12984, 30909, 54253, 38139, 40130, 10616, 52816}
# ** MATCH at time = 499135: Current queue = '{8035, 12984, 30909, 54253, 38139, 40130, 10616, 52816}
# ** MATCH at time = 499185: Current queue = '{12984, 30909, 54253, 38139, 40130, 10616, 52816}
# ** MATCH at time = 499235: Current queue = '{30909, 54253, 38139, 40130, 10616, 52816, 28848}
# ** MATCH at time = 499285: Current queue = '{30909, 54253, 38139, 40130, 10616, 52816, 28848, 8731}
# ** MATCH at time = 499335: Current queue = '{30909, 54253, 38139, 40130, 10616, 52816, 28848, 8731}
# ** MATCH at time = 499385: Current queue = '{30909, 54253, 38139, 40130, 10616, 52816, 28848, 8731}
# ** MATCH at time = 499435: Current queue = '{30909, 54253, 38139, 40130, 10616, 52816, 28848, 8731}
# ** MATCH at time = 499485: Current queue = '{54253, 38139, 40130, 10616, 52816, 28848, 8731}
# ** MATCH at time = 499535: Current queue = '{54253, 38139, 40130, 10616, 52816, 28848, 8731, 11042}
# ** MATCH at time = 499585: Current queue = '{38139, 40130, 10616, 52816, 28848, 8731, 11042}
# ** MATCH at time = 499635: Current queue = '{38139, 40130, 10616, 52816, 28848, 8731, 11042, 42047}
# ** MATCH at time = 499685: Current queue = '{38139, 40130, 10616, 52816, 28848, 8731, 11042, 42047}
# ** MATCH at time = 499735: Current queue = '{38139, 40130, 10616, 52816, 28848, 8731, 11042, 42047}
# ** MATCH at time = 499785: Current queue = '{38139, 40130, 10616, 52816, 28848, 8731, 11042, 42047}
# ** MATCH at time = 499835: Current queue = '{38139, 40130, 10616, 52816, 28848, 8731, 11042, 42047}
# ** MATCH at time = 499885: Current queue = '{38139, 40130, 10616, 52816, 28848, 8731, 11042, 42047}
# ** MATCH at time = 499935: Current queue = '{40130, 10616, 52816, 28848, 8731, 11042, 42047}
# ** MATCH at time = 499985: Current queue = '{40130, 10616, 52816, 28848, 8731, 11042, 42047}
# ** MATCH at time = 500035: Current queue = '{40130, 10616, 52816, 28848, 8731, 11042, 42047, 41024}
Final values stored in the queue = '{40130, 10616, 52816, 28848, 8731, 11042, 42047, 41024}
no.of error_count :26 ,no.of correct_count :9974
** Note: $stop    : Monitor.sv(59)
   Time: 500035 ns  Iteration: 0  Instance: /FIFO_top/monitor
Break in Module FIFO_monitor at Monitor.sv line 59
```

Assertion :

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/a1	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) (~if_...	✓
/FIFO_top/dut/a2	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) disa...	✓
/FIFO_top/dut/a3	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) disa...	✓
/FIFO_top/dut/a4	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) disa...	✓
/FIFO_top/dut/a5	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) disa...	✓
/FIFO_top/dut/a6	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) disa...	✓
/FIFO_top/dut/a7	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) disa...	✓
/FIFO_top/dut/a8	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) disa...	✓
/FIFO_top/dut/a9	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) disa...	✓
/FIFO_top/dut/a10	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) disa...	✓
/FIFO_top/dut/a11	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge if_obj.clk) disa...	✓
/FIFO_top/TB/#ubl...	Immediate	SVA	on	0	1	-	-	-	-		off	assert (randomize(...))	✓
/FIFO_top/TB/#ubl...	Immediate	SVA	on	0	1	-	-	-	-		off	assert (immed__53)	✓
/FIFO_top/TB/#ubl...	Immediate	SVA	on	0	1	-	-	-	-		off	assert ()	✓

Function coverage :





FIFO Verification Plan

Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
FIFO_1	When `rst_n` is asserted low, FIFO should reset all control and data outputs	Directed at beginning of simulation; randomized to keep `rst_n` high for most of sim time	Cover single and multiple reset events	Immediate assertion to check FIFO resets correctly
FIFO_2	When `wr_en` is high and FIFO not full, data should be written and `wr_ack` should be high	Randomize `wr_en` with constraints to avoid full condition; monitor `full` flag	Cover full range of `data_in`, and cross with `wr_en` and `full`	Concurrent assertion to check write and `wr_ack` handshake
FIFO_3	When `rd_en` is high and FIFO not empty, data should be read and `data_out` should be valid	Randomize `rd_en` avoiding empty condition; ensure `data_out`	Cover values of `data_out`, and cross with `rd_en` and `empty`	Concurrent assertion to check read and data validity

		changes appropriately		
FIFO_4	`overflow` should be high if write attempted when FIFO is full	Randomized overflow test cases; force writes when full	Cover transition to full and attempted writes thereafter	Assertion to check for overflow generation
FIFO_5	`underflow` should be high if read attempted when FIFO is empty	Randomized underflow cases; force reads when empty	Cover transition to empty and attempted reads thereafter	Assertion to check for underflow generation
FIFO_6	`almostfull` should be high when FIFO is near full (as per defined threshold)	Fill FIFO close to threshold using constrained randomization	Cover FIFO fill levels near full and trigger `almostfull`	Assertion to validate `almostfull` threshold behavior
FIFO_7	`almostempty` should be high when FIFO is near empty	Drain FIFO close to empty threshold using constrained randomization	Cover FIFO levels near empty and trigger `almostempty`	Assertion to validate `almostempty` threshold behavior
FIFO_8	`full` and `empty` flags should be exclusive; cannot be high at the same time	Random scenarios to test boundary conditions	Cover transitions between `full` ↔ `empty`	Assertion to ensure mutual exclusivity between `full` and `empty` flags
FIFO_9	FIFO should maintain write-read order (first-in first-out)	Directed test: Push known sequence of values and verify correct read order	Cover cross of `wr_en`, `rd_en`, and index tracking	Functional checking via scoreboard or model comparison

Function coverage :

```
Coverage Report by instance with details

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

Directive Coverage:
    Directives                      11        11         0   100.00%

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                    Design Design   Lang File(Line)      Hits Status
                                        Unit   UnitType
--------------------------------------------------------------------------------------------
/FIFO_top/dut/c1                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(87)
                                                                                  203 Covered
/FIFO_top/dut/c2                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(95)
                                                                                 3968 Covered
/FIFO_top/dut/c3                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(103)
                                                                                 2835 Covered
/FIFO_top/dut/c4                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(111)
                                                                                  108 Covered
/FIFO_top/dut/c5                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(119)
                                                                                  344 Covered
/FIFO_top/dut/c6                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(127)
                                                                                 4059 Covered
/FIFO_top/dut/c7                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(135)
                                                                                 2517 Covered
/FIFO_top/dut/c8                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(143)
                                                                                  476 Covered
/FIFO_top/dut/c9                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(151)
                                                                                 2829 Covered
/FIFO_top/dut/c10                       FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(159)
                                                                                  779 Covered
/FIFO_top/dut/c11                       FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(167)
                                                                                 9798 Covered

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

```
================================================================================
=== 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        58        58         0   100.00%
----------------------------------------------------------------------------------------------------------------
Covergroup                                                Metric       Goal       Bins    Status

----------------------------------------------------------------------------------------------------------------
 TYPE /FIFO_coverage_pkg/FIFO_coverage/cg                100.00%        100          -    Covered
    covered/total bins:                                       58         58          -
    missing/total bins:                                        0         58          -
    % 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]                                          2917          1          -    Covered
        bin auto[1]                                          7083          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]                                          7087          1          -    Covered
        bin auto[1]                                          2913          1          -    Covered
    Coverpoint wr_ack_cp                                  100.00%        100          -    Covered
        covered/total bins:                                    2          2          -
        missing/total bins:                                    0          2          -
        % Hit:                                            100.00%        100          -
        bin auto[0]                                          5952          1          -    Covered
        bin auto[1]                                          4048          1          -    Covered
    Coverpoint full_cp                                    100.00%        100          -    Covered
        covered/total bins:                                    2          2          -
        missing/total bins:                                    0          2          -
        % Hit:                                            100.00%        100          -
        bin auto[0]                                          5852          1          -    Covered
        bin auto[1]                                          4148          1          -    Covered
    Coverpoint empty_cp                                   100.00%        100          -    Covered
        covered/total bins:                                    2          2          -
        missing/total bins:                                    0          2          -
        % Hit:                                            100.00%        100          -
        bin auto[0]                                          9653          1          -    Covered
        bin auto[1]                                           347          1          -    Covered
    Coverpoint almostfull_cp                              100.00%        100          -    Covered
        covered/total bins:                                    2          2          -
        missing/total bins:                                    0          2          -
        % Hit:                                            100.00%        100          -
        bin auto[0]                                          7428          1          -    Covered
        bin auto[1]                                          2572          1          -    Covered
    Coverpoint almostempty_cp                             100.00%        100          -    Covered
        covered/total bins:                                    2          2          -
        missing/total bins:                                    0          2          -
        % Hit:                                            100.00%        100          -
        bin auto[0]                                          9517          1          -    Covered
```

```
Coverpoint almostempty_cp                          100.00%        100          -    Covered
    covered/total bins:                                  2          2          -
    missing/total bins:                                  0          2          -
    % Hit:                                         100.00%        100          -
    bin auto[0]                                       9517          1          -    Covered
    bin auto[1]                                        483          1          -    Covered
Coverpoint overflow_cp                             100.00%        100          -    Covered
    covered/total bins:                                  2          2          -
    missing/total bins:                                  0          2          -
    % Hit:                                         100.00%        100          -
    bin auto[0]                                       6391          1          -    Covered
    bin auto[1]                                       3609          1          -    Covered
Coverpoint underflow_cp                            100.00%        100          -    Covered
    covered/total bins:                                  2          2          -
    missing/total bins:                                  0          2          -
    % Hit:                                         100.00%        100          -
    bin auto[0]                                       9865          1          -    Covered
    bin auto[1]                                        135          1          -    Covered
Cross wr_rd_wr_ack_cross                           100.00%        100          -    Covered
    covered/total bins:                                  6          6          -
    missing/total bins:                                  0          6          -
    % Hit:                                         100.00%        100          -
    Auto, Default and User Defined Bins:
        bin <auto[1],auto[1],auto[1]>                 1165          1          -    Covered
        bin <auto[1],auto[0],auto[1]>                 2883          1          -    Covered
        bin <auto[1],auto[1],auto[0]>                  898          1          -    Covered
        bin <auto[1],auto[0],auto[0]>                 2137          1          -    Covered
        bin <auto[0],auto[1],auto[0]>                  850          1          -    Covered
        bin <auto[0],auto[0],auto[0]>                 2067          1          -    Covered
    Illegal and Ignore Bins:
        ignore_bin read_write_active_with_wr_ack
                                                         0                     -    ZERO
        ignore_bin write_active_with_wr_ack              0                     -    ZERO
Cross wr_rd_full_cross                             100.00%        100          -    Covered
    covered/total bins:                                  5          5          -
    missing/total bins:                                  0          5          -
    % Hit:                                         100.00%        100          -
    Auto, Default and User Defined Bins:
        bin <auto[1],auto[0],auto[1]>                 3305          1          -    Covered
        bin <auto[1],auto[1],auto[0]>                 2063          1          -    Covered
        bin <auto[0],auto[1],auto[0]>                  850          1          -    Covered
        bin <auto[1],auto[0],auto[0]>                 1715          1          -    Covered
        bin <auto[0],auto[0],auto[0]>                 1224          1          -    Covered
    Illegal and Ignore Bins:
        ignore_bin read_active_with_full                 0                     -    ZERO
        ignore_bin write_active_with_full              843                     -    Occurred
Cross wr_rd_empty_cross                            100.00%        100          -    Covered
    covered/total bins:                                  6          6          -
    missing/total bins:                                  0          6          -
    % Hit:                                         100.00%        100          -
    Auto, Default and User Defined Bins:
        bin <auto[1],auto[1],auto[1]>                   45          1          -    Covered
        bin <auto[1],auto[1],auto[0]>                 2018          1          -    Covered
        bin <auto[0],auto[1],auto[1]>                   98          1          -    Covered
        bin <auto[0],auto[1],auto[0]>                  752          1          -    Covered
        bin <auto[1],auto[0],auto[0]>                 4926          1          -    Covered
        bin <auto[0],auto[0],auto[0]>                 1957          1          -    Covered
    Illegal and Ignore Bins:
        ignore_bin read_active_with_empty              204                     -    Occurred
Cross wr_rd_overflow_cross                         100.00%        100          -    Covered
```

```
    % Hit:                                         100.00%        100          -
    Auto, Default and User Defined Bins:
        bin <auto[1],auto[1],auto[1]>                 1054          1          -    Covered
        bin <auto[1],auto[0],auto[1]>                 2555          1          -    Covered
        bin <auto[1],auto[1],auto[0]>                 1009          1          -    Covered
        bin <auto[0],auto[1],auto[0]>                  850          1          -    Covered
        bin <auto[1],auto[0],auto[0]>                 2465          1          -    Covered
        bin <auto[0],auto[0],auto[0]>                 2067          1          -    Covered
    Illegal and Ignore Bins:
        ignore_bin write_active_with_overflow            0                     -    ZERO
Cross wr_rd_underflow_cross                        100.00%        100          -    Covered
    covered/total bins:                                  6          6          -
    missing/total bins:                                  0          6          -
    % Hit:                                         100.00%        100          -
    Auto, Default and User Defined Bins:
        bin <auto[1],auto[1],auto[1]>                   77          1          -    Covered
        bin <auto[1],auto[0],auto[1]>                 1986          1          -    Covered
        bin <auto[0],auto[1],auto[1]>                   58          1          -    Covered
        bin <auto[0],auto[1],auto[0]>                  792          1          -    Covered
        bin <auto[1],auto[0],auto[0]>                 5020          1          -    Covered
        bin <auto[0],auto[0],auto[0]>                 2067          1          -    Covered
    Illegal and Ignore Bins:
        ignore_bin read_active_with_underflow            0                     -    ZERO
Cross wr_rd_almostfull_cross                       100.00%        100          -    Covered
    covered/total bins:                                  6          6          -
    missing/total bins:                                  0          6          -
    % Hit:                                         100.00%        100          -
    Auto, Default and User Defined Bins:
        bin <auto[1],auto[1],auto[1]>                 1355          1          -    Covered
        bin <auto[1],auto[0],auto[1]>                  365          1          -    Covered
        bin <auto[1],auto[1],auto[0]>                  708          1          -    Covered
        bin <auto[1],auto[0],auto[0]>                 4655          1          -    Covered
        bin <auto[0],auto[1],auto[0]>                  531          1          -    Covered
        bin <auto[0],auto[0],auto[0]>                 1534          1          -    Covered
    Illegal and Ignore Bins:
        ignore_bin read_write_active_with_almostfull
                                                       319                     -    Occurred
        ignore_bin write_active_with_almostfull
                                                       852                     -    Occurred
Cross wr_rd_almostempty_cross                      100.00%        100          -    Covered
    covered/total bins:                                  5          5          -
    missing/total bins:                                  0          5          -
    % Hit:                                         100.00%        100          -
    Auto, Default and User Defined Bins:
        bin <auto[1],auto[1],auto[1]>                  185          1          -    Covered
        bin <auto[1],auto[1],auto[0]>                 1878          1          -    Covered
        bin <auto[1],auto[0],auto[0]>                 4845          1          -    Covered
        bin <auto[0],auto[1],auto[0]>                  805          1          -    Covered
        bin <auto[0],auto[0],auto[0]>                 1989          1          -    Covered
    Illegal and Ignore Bins:
        ignore_bin read_write_active_with_almostempty
                                                        45                     -    Occurred
        ignore_bin read_active_with_almostempty
                                                       253                     -    Occurred
```

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

----------------------------------------------------------------------------------------------------------------
 TYPE /FIFO_coverage_pkg/FIFO_coverage/cg             100.00%        100          -    Covered
    covered/total bins:                                    58         58          -
    missing/total bins:                                     0         58          -
    % 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]                                      2917          1          -    Covered
        bin auto[1]                                      7083          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]                                      7087          1          -    Covered
        bin auto[1]                                      2913          1          -    Covered
    Coverpoint wr_ack_cp                              100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      5952          1          -    Covered
        bin auto[1]                                      4048          1          -    Covered
    Coverpoint full_cp                                100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      5852          1          -    Covered
        bin auto[1]                                      4148          1          -    Covered
    Coverpoint empty_cp                               100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      9653          1          -    Covered
        bin auto[1]                                       347          1          -    Covered
    Coverpoint almostfull_cp                          100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      7428          1          -    Covered
        bin auto[1]                                      2572          1          -    Covered
    Coverpoint almostempty_cp                         100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      9517          1          -    Covered
        bin auto[1]                                       483          1          -    Covered
    Coverpoint overflow_cp                            100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      6391          1          -    Covered
        bin auto[1]                                      3609          1          -    Covered
    Coverpoint underflow_cp                           100.00%        100          -    Covered
```

```
        % Hit:                                  100.00%        100          -
        bin auto[0]                               9865          1          -    Covered
        bin auto[1]                                135          1          -    Covered
    Cross wr_rd_wr_ack_cross                    100.00%        100          -    Covered
        covered/total bins:                           6          6          -
        missing/total bins:                           0          6          -
        % Hit:                                  100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>          1165          1          -    Covered
            bin <auto[1],auto[0],auto[1]>          2883          1          -    Covered
            bin <auto[1],auto[1],auto[0]>           898          1          -    Covered
            bin <auto[1],auto[0],auto[0]>          2137          1          -    Covered
            bin <auto[0],auto[1],auto[0]>           850          1          -    Covered
            bin <auto[0],auto[0],auto[0]>          2067          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin read_write_active_with_wr_ack
                                                      0                     -    ZERO
            ignore_bin write_active_with_wr_ack       0                     -    ZERO
    Cross wr_rd_full_cross                      100.00%        100          -    Covered
        covered/total bins:                           5          5          -
        missing/total bins:                           0          5          -
        % Hit:                                  100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[0],auto[1]>          3305          1          -    Covered
            bin <auto[1],auto[1],auto[0]>          2063          1          -    Covered
            bin <auto[0],auto[1],auto[0]>           850          1          -    Covered
            bin <auto[1],auto[0],auto[0]>          1715          1          -    Covered
            bin <auto[0],auto[0],auto[0]>          1224          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin read_active_with_full          0                     -    ZERO
            ignore_bin write_active_with_full       843                     -    Occurred
    Cross wr_rd_empty_cross                     100.00%        100          -    Covered
        covered/total bins:                           6          6          -
        missing/total bins:                           0          6          -
        % Hit:                                  100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>            45          1          -    Covered
            bin <auto[1],auto[1],auto[0]>          2018          1          -    Covered
            bin <auto[0],auto[1],auto[1]>            98          1          -    Covered
            bin <auto[0],auto[1],auto[0]>           752          1          -    Covered
            bin <auto[1],auto[0],auto[0]>          4926          1          -    Covered
            bin <auto[0],auto[0],auto[0]>          1957          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin read_active_with_empty       204                     -    Occurred
    Cross wr_rd_overflow_cross                  100.00%        100          -    Covered
        covered/total bins:                           6          6          -
        missing/total bins:                           0          6          -
        % Hit:                                  100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>          1054          1          -    Covered
            bin <auto[1],auto[0],auto[1]>          2555          1          -    Covered
            bin <auto[1],auto[1],auto[0]>          1009          1          -    Covered
            bin <auto[0],auto[1],auto[0]>           850          1          -    Covered
            bin <auto[1],auto[0],auto[0]>          2465          1          -    Covered
            bin <auto[0],auto[0],auto[0]>          2067          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin write_active_with_overflow     0                     -    ZERO
    Cross wr_rd_underflow_cross                 100.00%        100          -    Covered
        covered/total bins:                           6          6          -
        missing/total bins:                           0          6          -
```

```
            bin <auto[1],auto[1],auto[1]>            77          1          -    Covered
            bin <auto[1],auto[1],auto[0]>          1986          1          -    Covered
            bin <auto[0],auto[1],auto[1]>            58          1          -    Covered
            bin <auto[0],auto[1],auto[0]>           792          1          -    Covered
            bin <auto[1],auto[0],auto[0]>          5020          1          -    Covered
            bin <auto[0],auto[0],auto[0]>          2067          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin read_active_with_underflow     0                     -    ZERO
    Cross wr_rd_almostfull_cross                100.00%        100          -    Covered
        covered/total bins:                           6          6          -
        missing/total bins:                           0          6          -
        % Hit:                                  100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>          1355          1          -    Covered
            bin <auto[1],auto[0],auto[1]>           365          1          -    Covered
            bin <auto[1],auto[1],auto[0]>           708          1          -    Covered
            bin <auto[1],auto[0],auto[0]>          4655          1          -    Covered
            bin <auto[0],auto[1],auto[0]>           531          1          -    Covered
            bin <auto[0],auto[0],auto[0]>          1534          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin read_write_active_with_almostfull
                                                    319                     -    Occurred
            ignore_bin write_active_with_almostfull
                                                    852                     -    Occurred
    Cross wr_rd_almostempty_cross               100.00%        100          -    Covered
        covered/total bins:                           5          5          -
        missing/total bins:                           0          5          -
        % Hit:                                  100.00%        100          -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>           185          1          -    Covered
            bin <auto[1],auto[0],auto[1]>          1878          1          -    Covered
            bin <auto[1],auto[1],auto[0]>          4845          1          -    Covered
            bin <auto[0],auto[1],auto[0]>           805          1          -    Covered
            bin <auto[0],auto[0],auto[0]>          1989          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin read_write_active_with_almostempty
                                                     45                     -    Occurred
            ignore_bin read_active_with_almostempty
                                                    253                     -    Occurred

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------
Name                                 Design Design   Lang File(Line)      Hits Status
                                     Unit   UnitType
--------------------------------------------------------------------------------
/FIFO_top/dut/c1                     FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(87)
                                                                  203 Covered
/FIFO_top/dut/c2                     FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(95)
                                                                 3968 Covered
/FIFO_top/dut/c3                     FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(103)
                                                                 2835 Covered
/FIFO_top/dut/c4                     FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(111)
                                                                  108 Covered
/FIFO_top/dut/c5                     FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(119)
                                                                  344 Covered
/FIFO_top/dut/c6                     FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(127)
                                                                 4059 Covered
/FIFO_top/dut/c7                     FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(135)
```

```
                                                  253                    -    Occurred

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------
Name                                    Design Design   Lang File(Line)      Hits Status
                                        Unit   UnitType
--------------------------------------------------------------------------------------
/FIFO_top/dut/c1                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(87)
                                                                                  203 Covered
/FIFO_top/dut/c2                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(95)
                                                                                 3968 Covered
/FIFO_top/dut/c3                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(103)
                                                                                 2835 Covered
/FIFO_top/dut/c4                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(111)
                                                                                  108 Covered
/FIFO_top/dut/c5                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(119)
                                                                                  344 Covered
/FIFO_top/dut/c6                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(127)
                                                                                 4059 Covered
/FIFO_top/dut/c7                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(135)
                                                                                 2517 Covered
/FIFO_top/dut/c8                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(143)
                                                                                  476 Covered
/FIFO_top/dut/c9                        FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(151)
                                                                                 2829 Covered
/FIFO_top/dut/c10                       FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(159)
                                                                                  779 Covered
/FIFO_top/dut/c11                       FIFO   Verilog  SVA  D:/Digital Verfication debloma/Projects/project1(FIFO)/FIFI_PROJECT/FIFO_after.sv(167)
                                                                                 9798 Covered

TOTAL DIRECTIVE COVERAGE: 100.00%  COVERS: 11

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