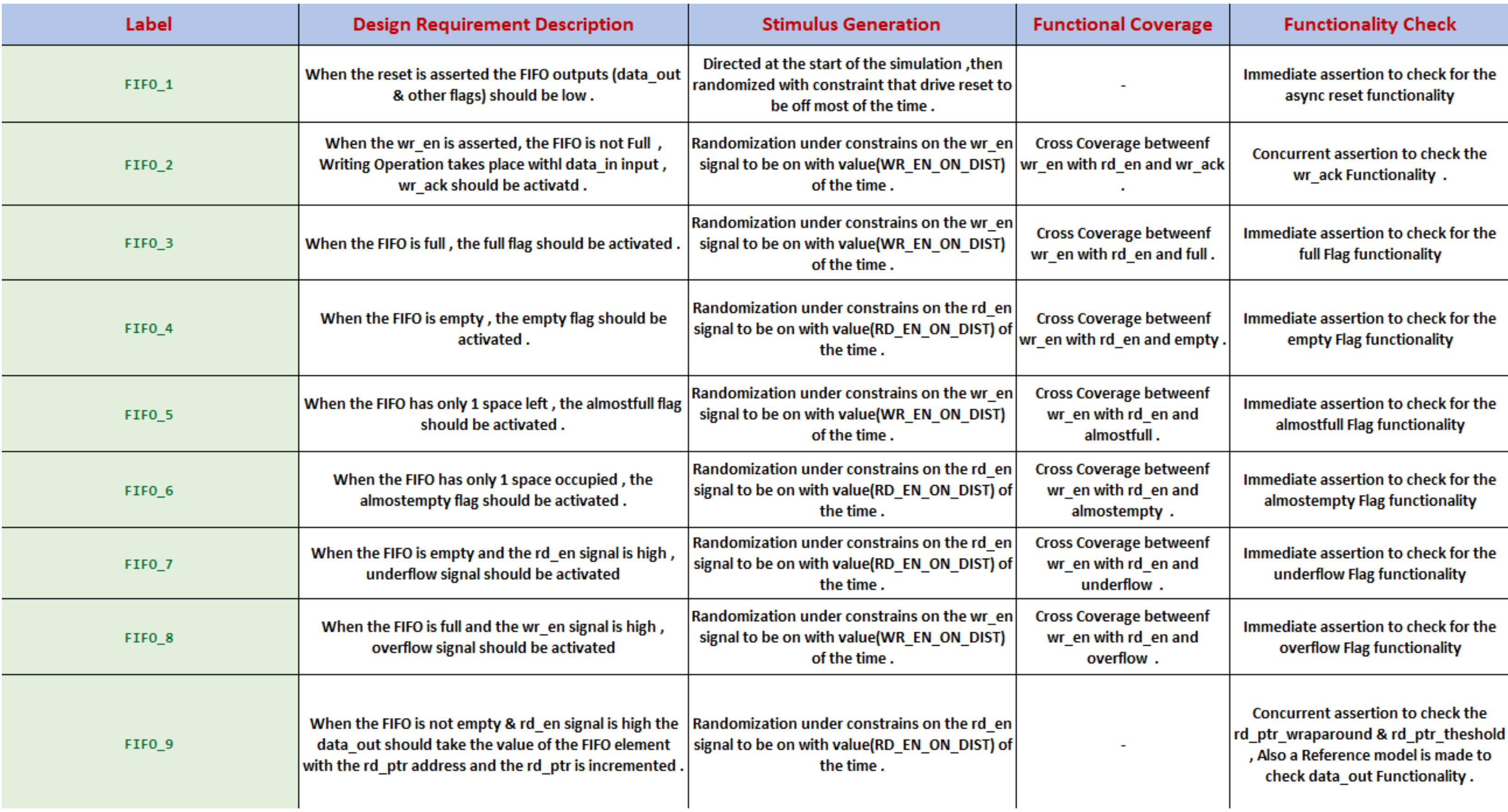
Mohamed Emadeldin Ali
SV7 – Digital Verification diploma
No.109

SV Project - Synchronous FIFO

1) Detected Bugs :

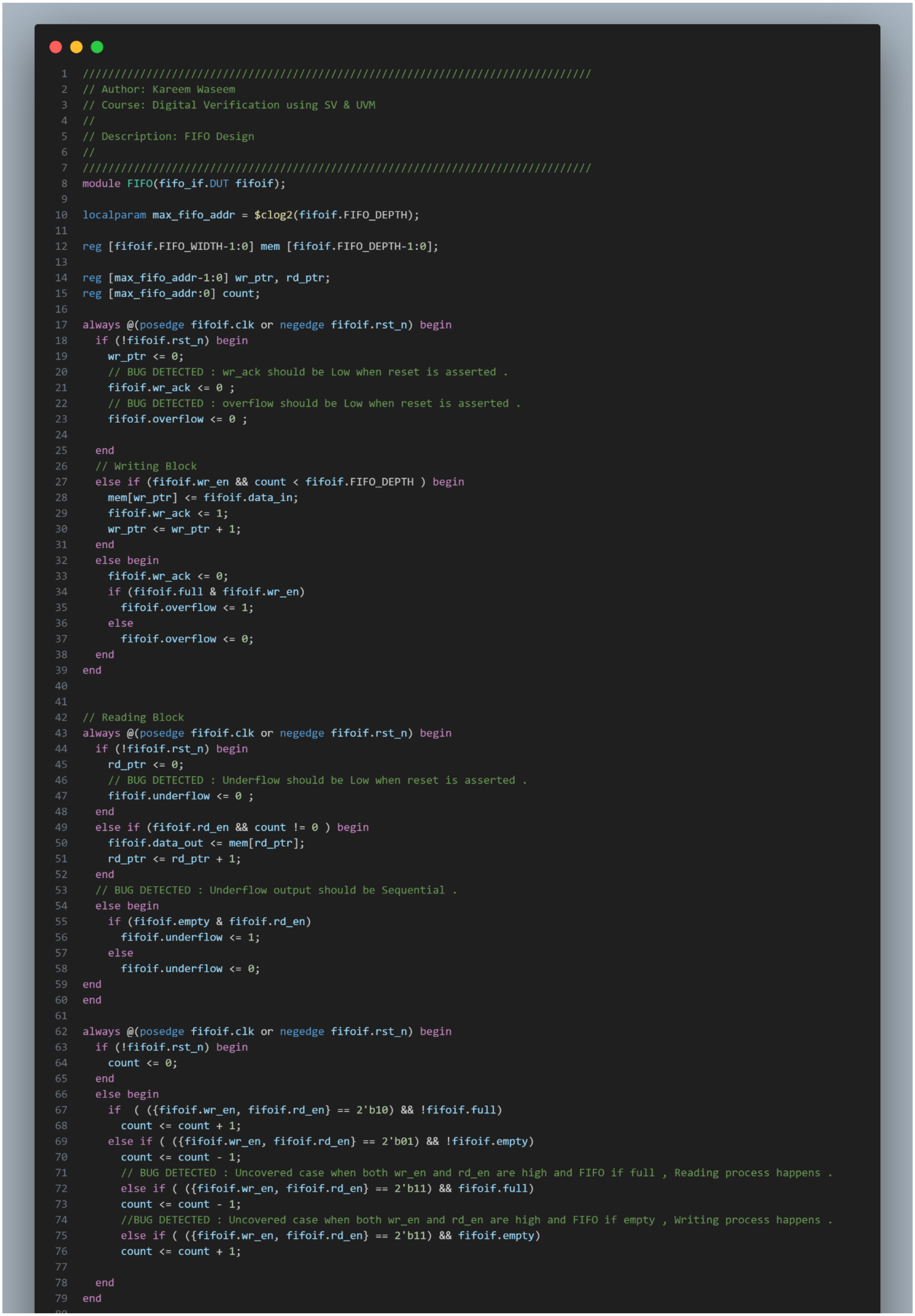
- wr_ack should be Low when reset is asserted .
- overflow should be Low when reset is asserted .
- underflow should be Low when reset is asserted .
- underflow output should be Sequential .
- Uncovered case when both wr_en and rd_en are high and FIFO if full , Reading process happens .
- Uncovered case when both wr_en and rd_en are high and FIFO if empty , Writing process happens .
- almostfull is high when there is two spots empty , while it should be when only one spot is empty .

2) Verification Plan :

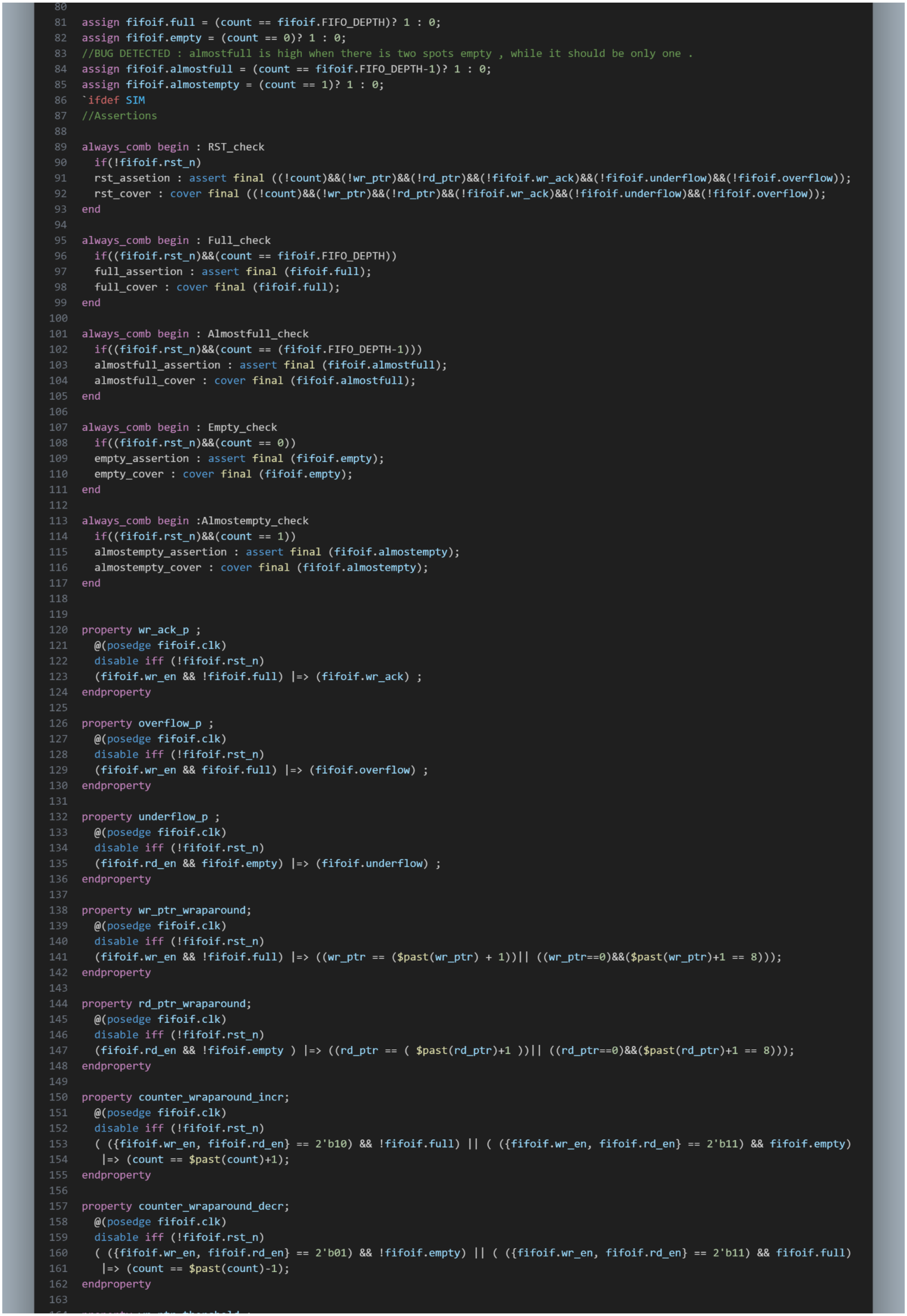


Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
FIFO_1	When the reset is asserted the FIFO outputs (data_out & other flags) should be low .	Directed at the start of the simulation ,then randomized with constraint that drive reset to be off most of the time .	-	Immediate assertion to check for the async reset functionality
FIFO_2	When the wr_en is asserted, the FIFO is not Full , Writing Operation takes place withl data_in input , wr_ack should be activatd .	Randomization under constrains on the wr_en signal to be on with value(WR_EN_ON_DIST) of the time .	Cross Coverage betweenf wr_en with rd_en and wr_ack .	Concurrent assertion to check the wr_ack Functionality .
FIFO_3	When the FIFO is full , the full flag should be activated .	Randomization under constrains on the wr_en signal to be on with value(WR_EN_ON_DIST) of the time .	Cross Coverage betweenf wr_en with rd_en and full .	Immediate assertion to check for the full Flag functionality
FIFO_4	When the FIFO is empty , the empty flag should be activated .	Randomization under constrains on the rd_en signal to be on with value(RD_EN_ON_DIST) of the time .	Cross Coverage betweenf wr_en with rd_en and empty .	Immediate assertion to check for the empty Flag functionality
FIFO_5	When the FIFO has only 1 space left , the almostfull flag should be activated .	Randomization under constrains on the wr_en signal to be on with value(WR_EN_ON_DIST) of the time .	Cross Coverage betweenf wr_en with rd_en and almostfull .	Immediate assertion to check for the almostfull Flag functionality
FIFO_6	When the FIFO has only 1 space occupied , the almostempty flag should be activated .	Randomization under constrains on the rd_en signal to be on with value(RD_EN_ON_DIST) of the time .	Cross Coverage betweenf wr_en with rd_en and almostempty .	Immediate assertion to check for the almostempty Flag functionality
FIFO_7	When the FIFO is empty and the rd_en signal is high , underflow signal should be activated	Randomization under constrains on the rd_en signal to be on with value(RD_EN_ON_DIST) of the time .	Cross Coverage betweenf wr_en with rd_en and underflow .	Immediate assertion to check for the underflow Flag functionality
FIFO_8	When the FIFO is full and the wr_en signal is high , overflow signal should be activated	Randomization under constrains on the wr_en signal to be on with value(WR_EN_ON_DIST) of the time .	Cross Coverage betweenf wr_en with rd_en and overflow .	Immediate assertion to check for the overflow Flag functionality
FIFO_9	When the FIFO is not empty & rd_en signal is high the data_out should take the value of the FIFO element with the rd_ptr address and the rd_ptr is incremented .	Randomization under constrains on the rd_en signal to be on with value(RD_EN_ON_DIST) of the time .	-	Concurrent assertion to check the rd_ptr_wraparound & rd_ptr_theshold , Also a Reference model is made to check data_out Functionality .

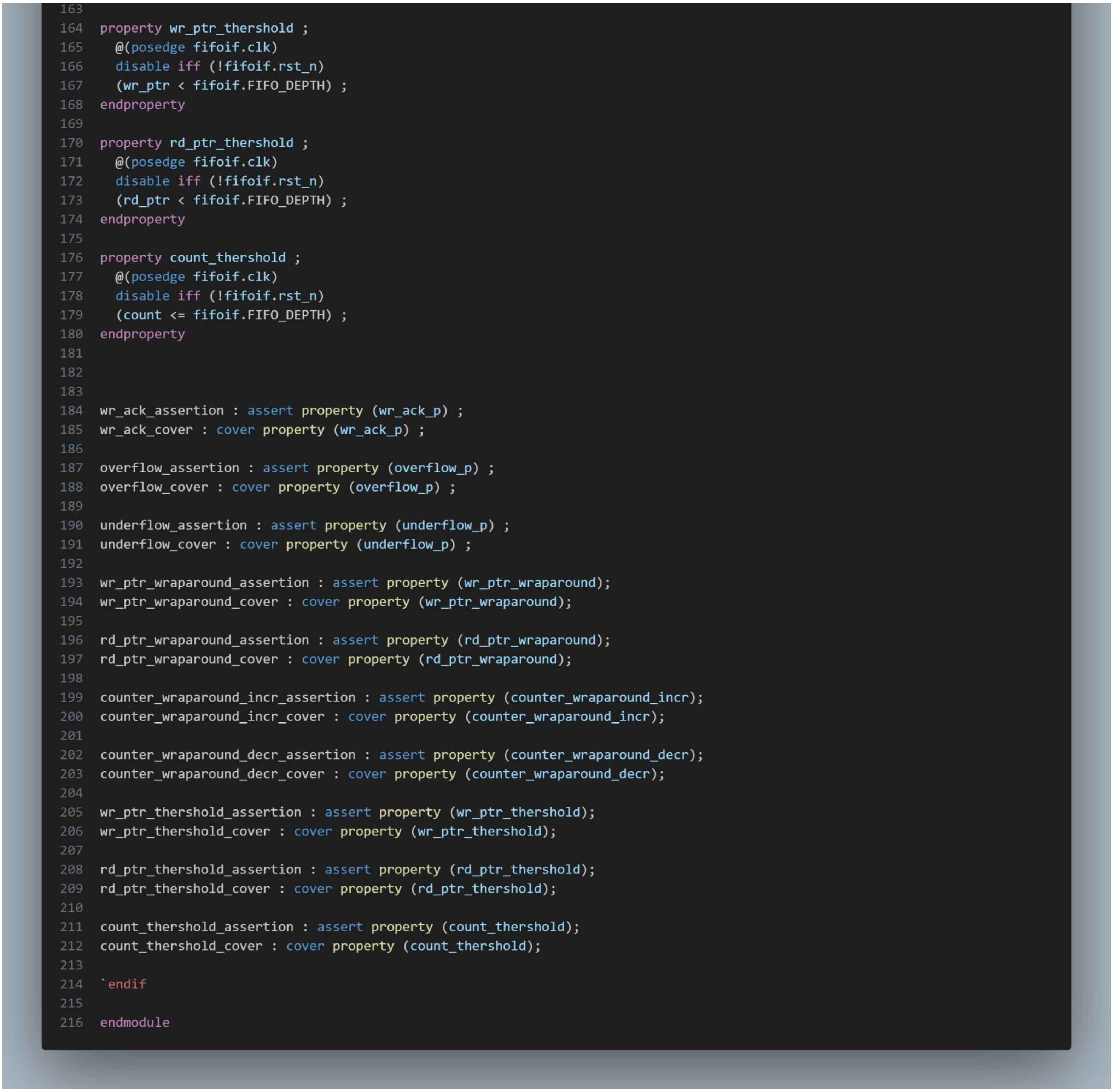
3) Design (RTL) Code [with assertions added in the same file]:



```
////////////////////////////////////////////////////////////////////////////////
// Author: Kareem Waseem
// Course: Digital Verification using SV & UVM
//
// Description: FIFO Design
//
////////////////////////////////////////////////////////////////////////////////
module FIFO(fifo_if.DUT fifoif);

localparam max_fifo_addr = $clog2(fifoif.FIFO_DEPTH);

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

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

always @(posedge fifoif.clk or negedge fifoif.rst_n) begin
  if (!fifoif.rst_n) begin
    wr_ptr <= 0;
    // BUG DETECTED : wr_ack should be Low when reset is asserted .
    fifoif.wr_ack <= 0 ;
    // BUG DETECTED : overflow should be Low when reset is asserted .
    fifoif.overflow <= 0 ;

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


// Reading Block
always @(posedge fifoif.clk or negedge fifoif.rst_n) begin
  if (!fifoif.rst_n) begin
    rd_ptr <= 0;
    // BUG DETECTED : Underflow should be Low when reset is asserted .
    fifoif.underflow <= 0 ;
  end
  else if (fifoif.rd_en && count != 0 ) begin
    fifoif.data_out <= mem[rd_ptr];
    rd_ptr <= rd_ptr + 1;
  end
  // BUG DETECTED : Underflow output should be Sequential .
  else begin
    if (fifoif.empty & fifoif.rd_en)
      fifoif.underflow <= 1;
    else
      fifoif.underflow <= 0;
end
end

always @(posedge fifoif.clk or negedge fifoif.rst_n) begin
  if (!fifoif.rst_n) begin
    count <= 0;
  end
  else begin
    if  ( ({fifoif.wr_en, fifoif.rd_en} == 2'b10) && !fifoif.full)
      count <= count + 1;
    else if ( ({fifoif.wr_en, fifoif.rd_en} == 2'b01) && !fifoif.empty)
      count <= count - 1;
      // BUG DETECTED : Uncovered case when both wr_en and rd_en are high and FIFO if full , Reading process happens .
      else if ( ({fifoif.wr_en, fifoif.rd_en} == 2'b11) && fifoif.full)
      count <= count - 1;
      //BUG DETECTED : Uncovered case when both wr_en and rd_en are high and FIFO if empty , Writing process happens .
      else if ( ({fifoif.wr_en, fifoif.rd_en} == 2'b11) && fifoif.empty)
      count <= count + 1;

  end
end
```

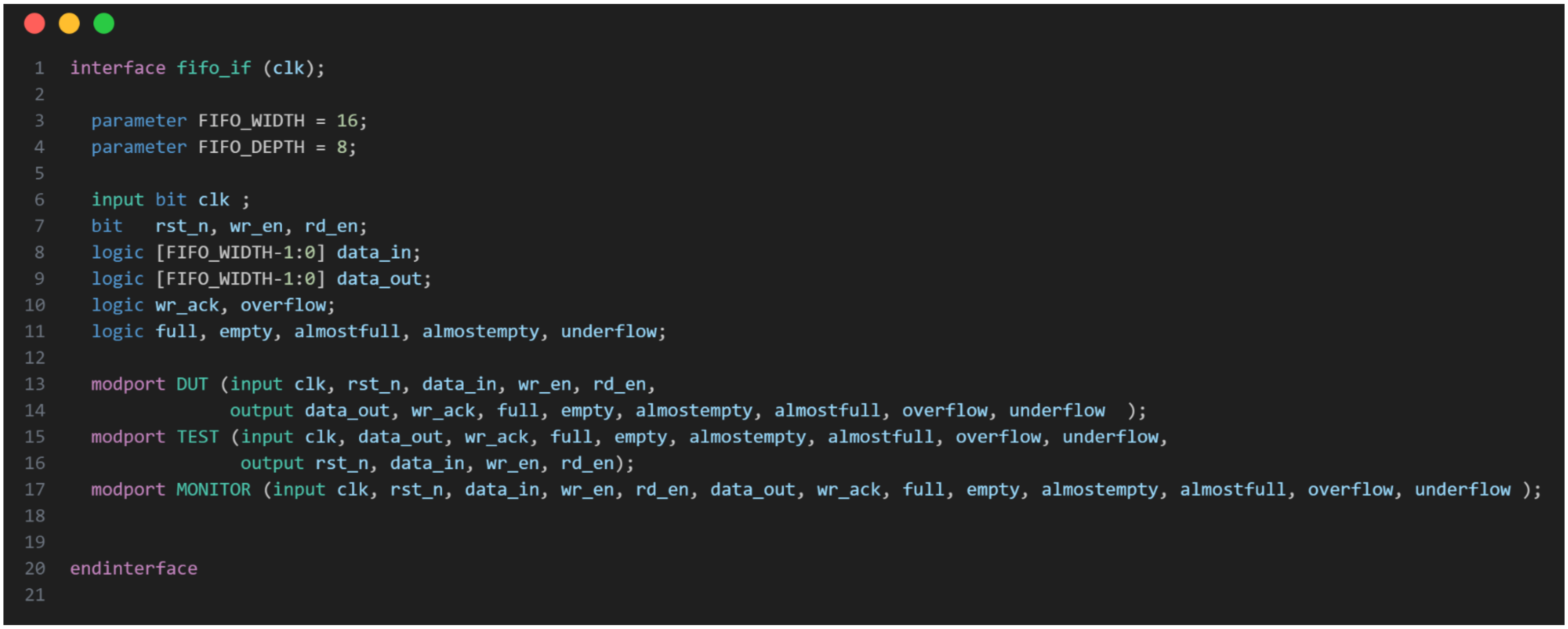


```
assign fifoif.full = (count == fifoif.FIFO_DEPTH)? 1 : 0;
assign fifoif.empty = (count == 0)? 1 : 0;
//BUG DETECTED : almostfull is high when there is two spots empty , while it should be only one .
assign fifoif.almostfull = (count == fifoif.FIFO_DEPTH-1)? 1 : 0;
assign fifoif.almostempty = (count == 1)? 1 : 0;
`ifdef SIM
//Assertions

always_comb begin : RST_check
  if(!fifoif.rst_n)
  rst_assetion : assert final ((!count)&&(!wr_ptr)&&(!rd_ptr)&&(!fifoif.wr_ack)&&(!fifoif.underflow)&&(!fifoif.overflow));
  rst_cover : cover final ((!count)&&(!wr_ptr)&&(!rd_ptr)&&(!fifoif.wr_ack)&&(!fifoif.underflow)&&(!fifoif.overflow));
end

always_comb begin : Full_check
  if((fifoif.rst_n)&&(count == fifoif.FIFO_DEPTH))
  full_assertion : assert final (fifoif.full);
  full_cover : cover final (fifoif.full);
end

always_comb begin : Almostfull_check
  if((fifoif.rst_n)&&(count == (fifoif.FIFO_DEPTH-1)))
  almostfull_assertion : assert final (fifoif.almostfull);
  almostfull_cover : cover final (fifoif.almostfull);
end

always_comb begin : Empty_check
  if((fifoif.rst_n)&&(count == 0))
  empty_assertion : assert final (fifoif.empty);
  empty_cover : cover final (fifoif.empty);
end

always_comb begin :Almostempty_check
  if((fifoif.rst_n)&&(count == 1))
  almostempty_assertion : assert final (fifoif.almostempty);
  almostempty_cover : cover final (fifoif.almostempty);
end


property wr_ack_p ;
  @(posedge fifoif.clk)
  disable iff (!fifoif.rst_n)
  (fifoif.wr_en && !fifoif.full) |=> (fifoif.wr_ack) ;
endproperty

property overflow_p ;
  @(posedge fifoif.clk)
  disable iff (!fifoif.rst_n)
  (fifoif.wr_en && fifoif.full) |=> (fifoif.overflow) ;
endproperty

property underflow_p ;
  @(posedge fifoif.clk)
  disable iff (!fifoif.rst_n)
  (fifoif.rd_en && fifoif.empty) |=> (fifoif.underflow) ;
endproperty

property wr_ptr_wraparound;
  @(posedge fifoif.clk)
  disable iff (!fifoif.rst_n)
  (fifoif.wr_en && !fifoif.full) |=> ((wr_ptr == ($past(wr_ptr) + 1))|| ((wr_ptr==0)&&($past(wr_ptr)+1 == 8)));
endproperty

property rd_ptr_wraparound;
  @(posedge fifoif.clk)
  disable iff (!fifoif.rst_n)
  (fifoif.rd_en && !fifoif.empty ) |=> ((rd_ptr == ( $past(rd_ptr)+1 ))|| ((rd_ptr==0)&&($past(rd_ptr)+1 == 8)));
endproperty

property counter_wraparound_incr;
  @(posedge fifoif.clk)
  disable iff (!fifoif.rst_n)
  ( ({fifoif.wr_en, fifoif.rd_en} == 2'b10) && !fifoif.full) || ( ({fifoif.wr_en, fifoif.rd_en} == 2'b11) && fifoif.empty)
   |=> (count == $past(count)+1);
endproperty

property counter_wraparound_decr;
  @(posedge fifoif.clk)
  disable iff (!fifoif.rst_n)
  ( ({fifoif.wr_en, fifoif.rd_en} == 2'b01) && !fifoif.empty) || ( ({fifoif.wr_en, fifoif.rd_en} == 2'b11) && fifoif.full)
   |=> (count == $past(count)-1);
endproperty

```



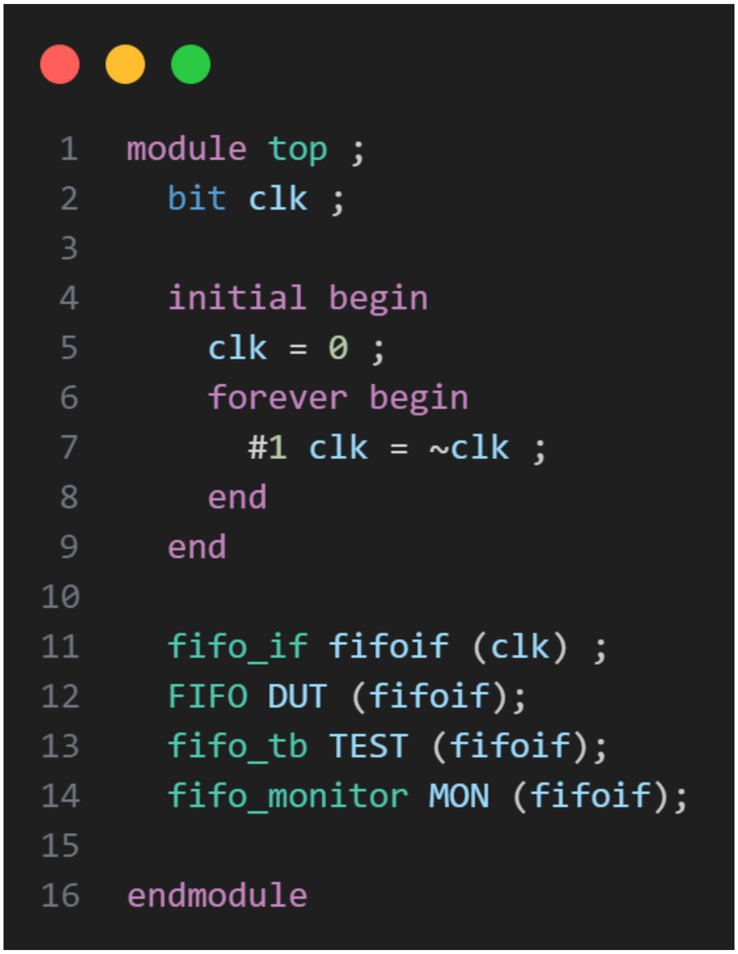
```
property wr_ptr_thershold ;
  @(posedge fifoif.clk)
  disable iff (!fifoif.rst_n)
  (wr_ptr < fifoif.FIFO_DEPTH) ;
endproperty

property rd_ptr_thershold ;
  @(posedge fifoif.clk)
  disable iff (!fifoif.rst_n)
  (rd_ptr < fifoif.FIFO_DEPTH) ;
endproperty

property count_thershold ;
  @(posedge fifoif.clk)
  disable iff (!fifoif.rst_n)
  (count <= fifoif.FIFO_DEPTH) ;
endproperty



wr_ack_assertion : assert property (wr_ack_p) ;
wr_ack_cover : cover property (wr_ack_p) ;

overflow_assertion : assert property (overflow_p) ;
overflow_cover : cover property (overflow_p) ;

underflow_assertion : assert property (underflow_p) ;
underflow_cover : cover property (underflow_p) ;

wr_ptr_wraparound_assertion : assert property (wr_ptr_wraparound);
wr_ptr_wraparound_cover : cover property (wr_ptr_wraparound);

rd_ptr_wraparound_assertion : assert property (rd_ptr_wraparound);
rd_ptr_wraparound_cover : cover property (rd_ptr_wraparound);

counter_wraparound_incr_assertion : assert property (counter_wraparound_incr);
counter_wraparound_incr_cover : cover property (counter_wraparound_incr);

counter_wraparound_decr_assertion : assert property (counter_wraparound_decr);
counter_wraparound_decr_cover : cover property (counter_wraparound_decr);

wr_ptr_thershold_assertion : assert property (wr_ptr_thershold);
wr_ptr_thershold_cover : cover property (wr_ptr_thershold);

rd_ptr_thershold_assertion : assert property (rd_ptr_thershold);
rd_ptr_thershold_cover : cover property (rd_ptr_thershold);

count_thershold_assertion : assert property (count_thershold);
count_thershold_cover : cover property (count_thershold);

`endif

endmodule
```

4) Interface Code :



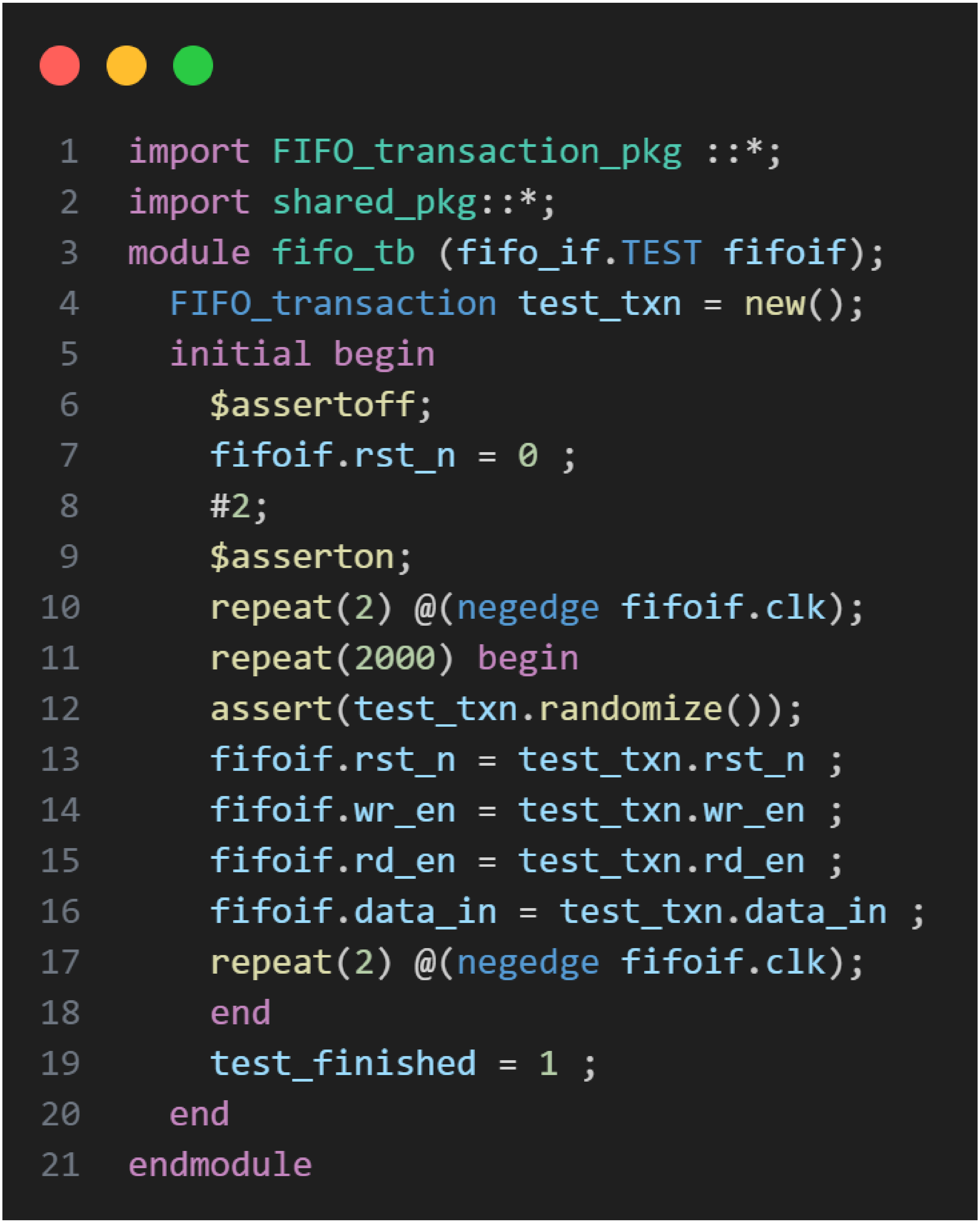
```
interface fifo_if (clk);

  parameter FIFO_WIDTH = 16;
  parameter FIFO_DEPTH = 8;

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

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


endinterface
```

5) Top module Code :



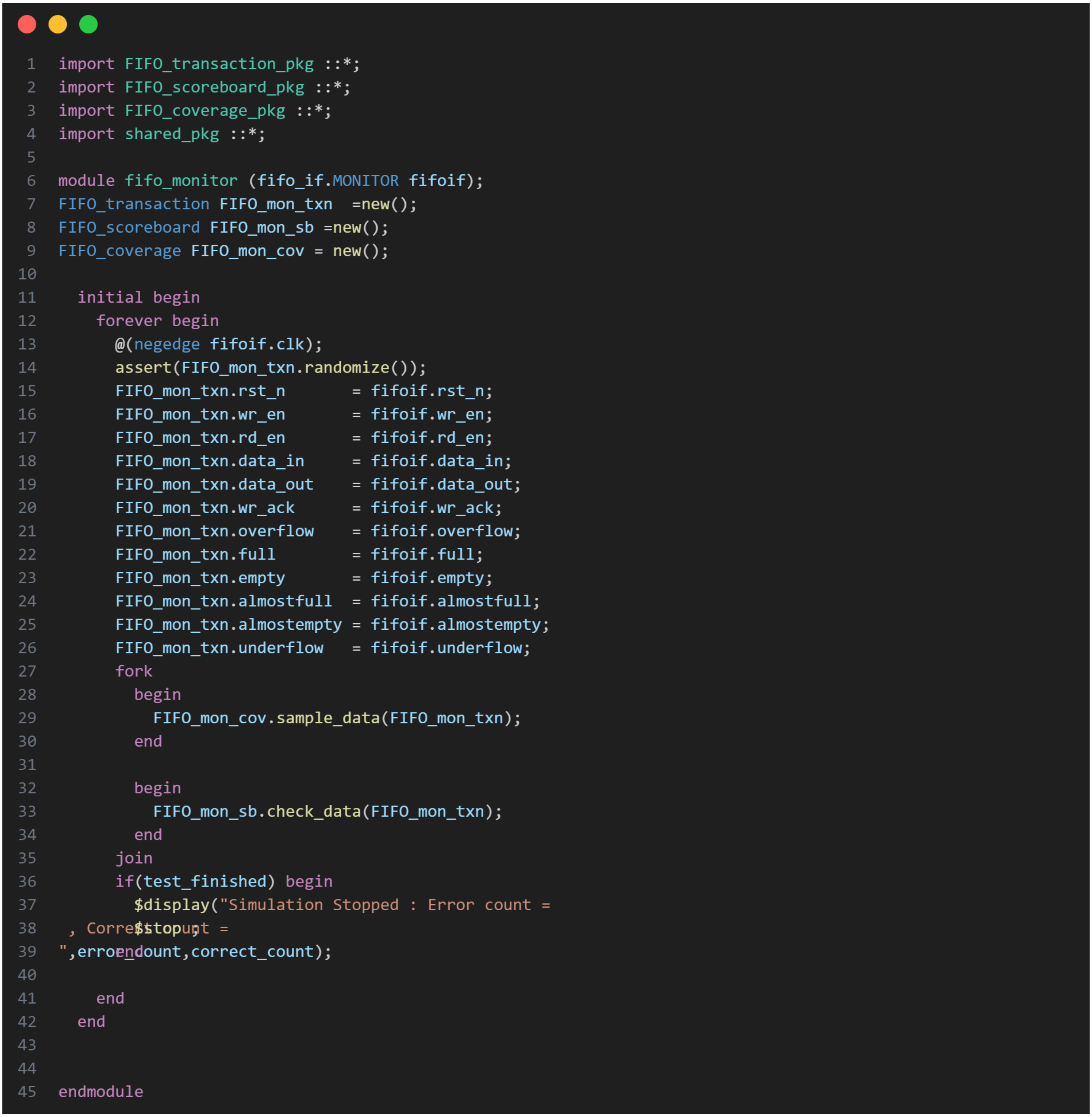
```
module top ;
  bit clk ;

  initial begin
    clk = 0 ;
    forever begin
      #1 clk = ~clk ;
    end
  end

  fifo_if fifoif (clk) ;
  FIFO DUT (fifoif);
  fifo_tb TEST (fifoif);
  fifo_monitor MON (fifoif);

endmodule
```

6) Testbench Code :



```
import FIFO_transaction_pkg ::*;
import shared_pkg::*;
module fifo_tb (fifo_if.TEST fifoif);
  FIFO_transaction test_txn = new();
  initial begin
    $assertoff;
    fifoif.rst_n = 0 ;
    #2;
    $asserton;
    repeat(2) @(negedge fifoif.clk);
    repeat(2000) begin
    assert(test_txn.randomize());
    fifoif.rst_n = test_txn.rst_n ;
    fifoif.wr_en = test_txn.wr_en ;
    fifoif.rd_en = test_txn.rd_en ;
    fifoif.data_in = test_txn.data_in ;
    repeat(2) @(negedge fifoif.clk);
    end
    test_finished = 1 ;
  end
endmodule
```

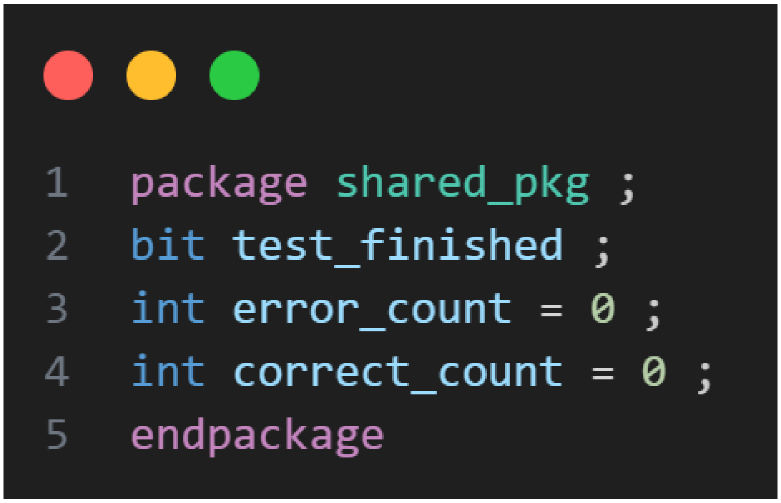
7) Monitor module Code :



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

module fifo_monitor (fifo_if.MONITOR fifoif);
FIFO_transaction FIFO_mon_txn  =new();
FIFO_scoreboard FIFO_mon_sb =new();
FIFO_coverage FIFO_mon_cov = new();

  initial begin
    forever begin
      @(negedge fifoif.clk);
      assert(FIFO_mon_txn.randomize());
      FIFO_mon_txn.rst_n       = fifoif.rst_n;
      FIFO_mon_txn.wr_en       = fifoif.wr_en;
      FIFO_mon_txn.rd_en       = fifoif.rd_en;
      FIFO_mon_txn.data_in     = fifoif.data_in;
      FIFO_mon_txn.data_out    = fifoif.data_out;
      FIFO_mon_txn.wr_ack      = fifoif.wr_ack;
      FIFO_mon_txn.overflow    = fifoif.overflow;
      FIFO_mon_txn.full        = fifoif.full;
      FIFO_mon_txn.empty       = fifoif.empty;
      FIFO_mon_txn.almostfull  = fifoif.almostfull;
      FIFO_mon_txn.almostempty = fifoif.almostempty;
      FIFO_mon_txn.underflow   = fifoif.underflow;
      fork
        begin
          FIFO_mon_cov.sample_data(FIFO_mon_txn);
        end

        begin
          FIFO_mon_sb.check_data(FIFO_mon_txn);
        end
      join
      if(test_finished) begin
        $display("Simulation Stopped : Error count =
 , Corre$stopunt =
",erroendount,correct_count);

    end
  end


endmodule
```

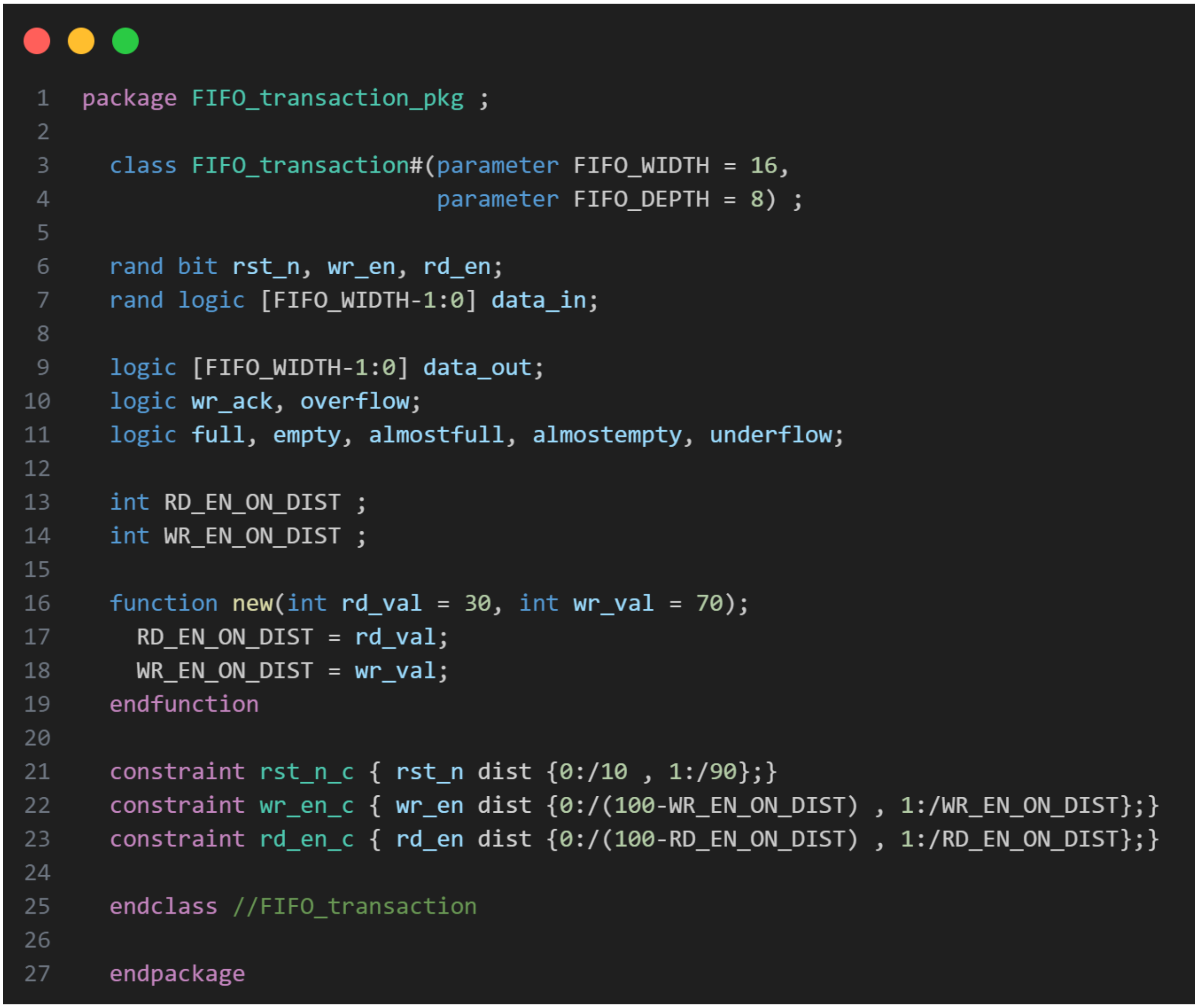
8) Packages :

- **Shared package :**



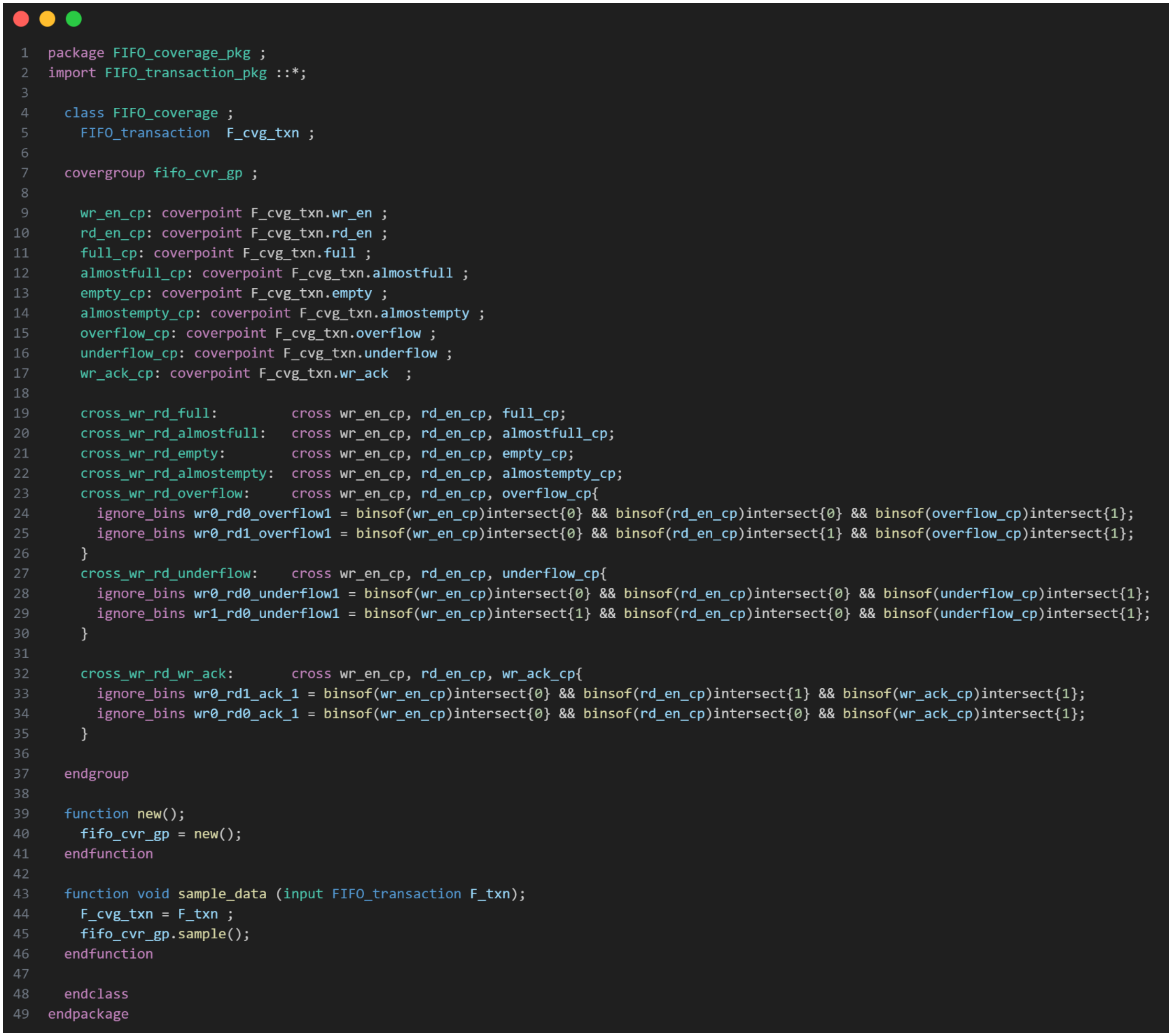
```
package shared_pkg ;
bit test_finished ;
int error_count = 0 ;
int correct_count = 0 ;
endpackage
```

- **Transaction package :**



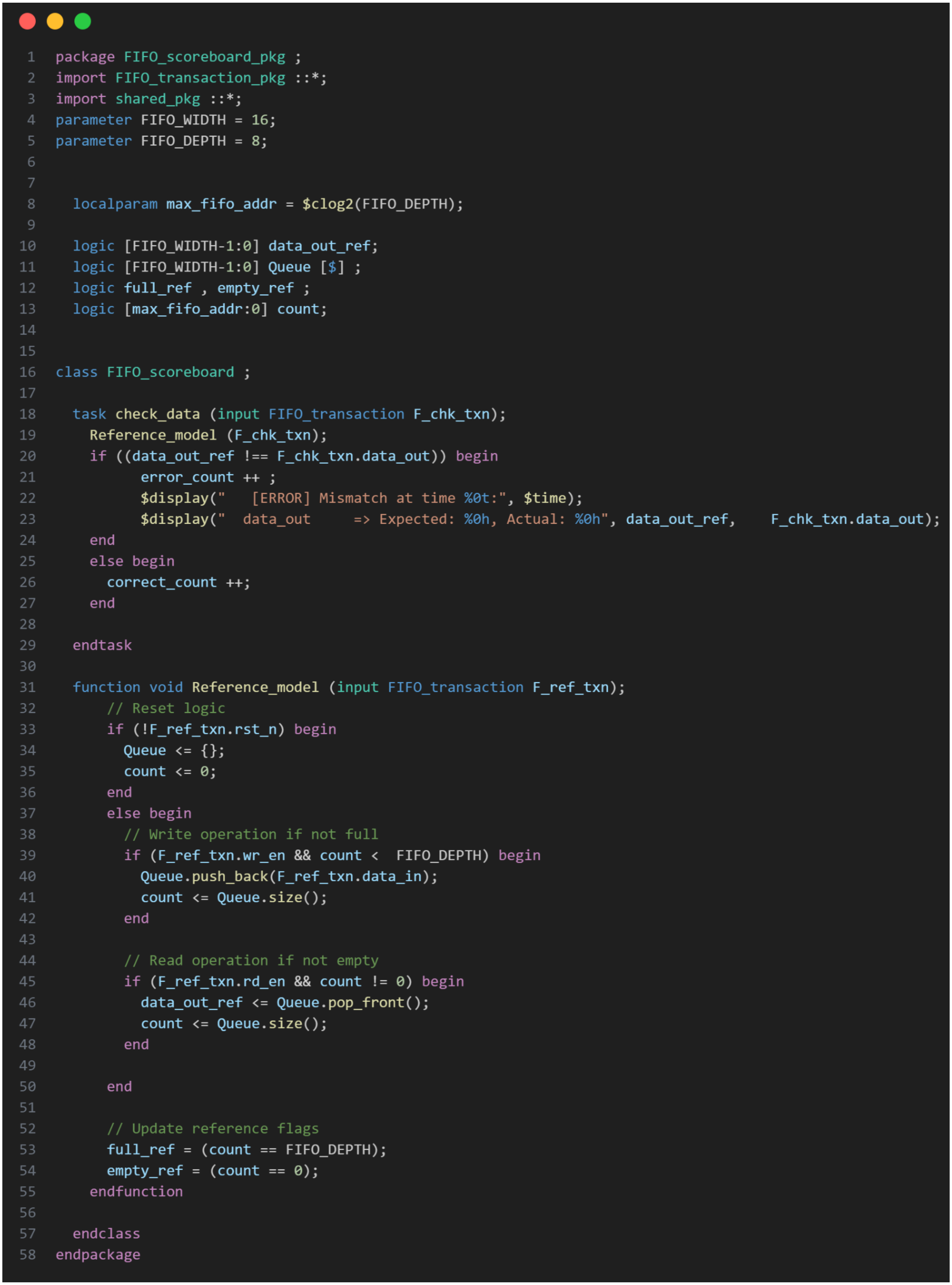
```
package FIFO_transaction_pkg ;

  class FIFO_transaction#(parameter FIFO_WIDTH = 16,
                          parameter FIFO_DEPTH = 8) ;

  rand bit rst_n, wr_en, rd_en;
  rand logic [FIFO_WIDTH-1:0] data_in;

  logic [FIFO_WIDTH-1:0] data_out;
  logic wr_ack, overflow;
  logic full, empty, almostfull, almostempty, underflow;

  int RD_EN_ON_DIST ;
  int WR_EN_ON_DIST ;

  function new(int rd_val = 30, int wr_val = 70);
    RD_EN_ON_DIST = rd_val;
    WR_EN_ON_DIST = wr_val;
  endfunction

  constraint rst_n_c { rst_n dist {0:/10 , 1:/90};}
  constraint wr_en_c { wr_en dist {0:/(100-WR_EN_ON_DIST) , 1:/WR_EN_ON_DIST};}
  constraint rd_en_c { rd_en dist {0:/(100-RD_EN_ON_DIST) , 1:/RD_EN_ON_DIST};}

  endclass //FIFO_transaction

  endpackage
```

- **Coverage package :**



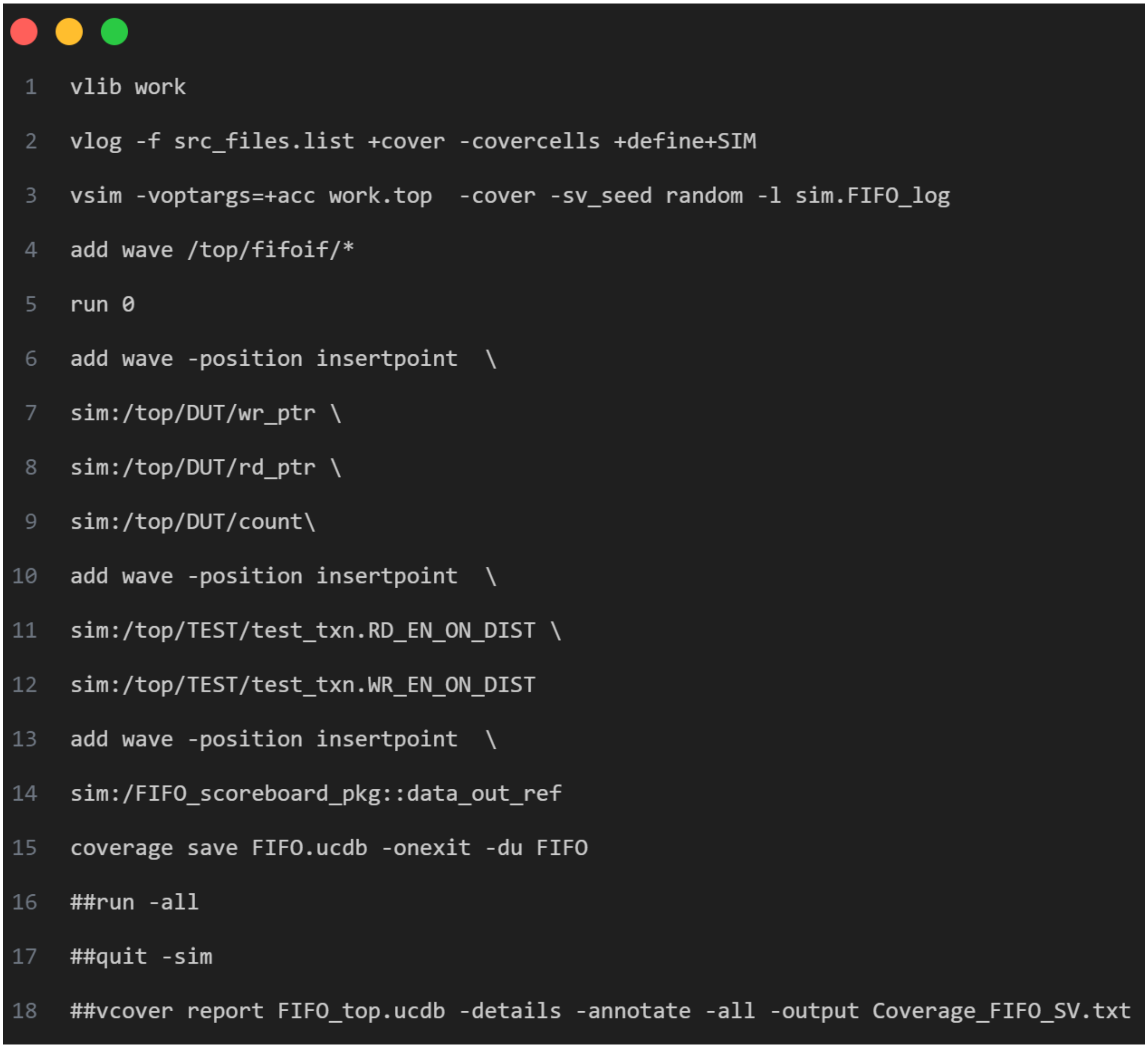
```
package FIFO_coverage_pkg ;
import FIFO_transaction_pkg ::*;

  class FIFO_coverage ;
    FIFO_transaction  F_cvg_txn ;

  covergroup fifo_cvr_gp ;

    wr_en_cp: coverpoint F_cvg_txn.wr_en ;
    rd_en_cp: coverpoint F_cvg_txn.rd_en ;
    full_cp: coverpoint F_cvg_txn.full ;
    almostfull_cp: coverpoint F_cvg_txn.almostfull ;
    empty_cp: coverpoint F_cvg_txn.empty ;
    almostempty_cp: coverpoint F_cvg_txn.almostempty ;
    overflow_cp: coverpoint F_cvg_txn.overflow ;
    underflow_cp: coverpoint F_cvg_txn.underflow ;
    wr_ack_cp: coverpoint F_cvg_txn.wr_ack  ;

    cross_wr_rd_full:         cross wr_en_cp, rd_en_cp, full_cp;
    cross_wr_rd_almostfull:   cross wr_en_cp, rd_en_cp, almostfull_cp;
    cross_wr_rd_empty:        cross wr_en_cp, rd_en_cp, empty_cp;
    cross_wr_rd_almostempty:  cross wr_en_cp, rd_en_cp, almostempty_cp;
    cross_wr_rd_overflow:     cross wr_en_cp, rd_en_cp, overflow_cp{
      ignore_bins wr0_rd0_overflow1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{0} && binsof(overflow_cp)intersect{1};
      ignore_bins wr0_rd1_overflow1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{1} && binsof(overflow_cp)intersect{1};
    }
    cross_wr_rd_underflow:    cross wr_en_cp, rd_en_cp, underflow_cp{
      ignore_bins wr0_rd0_underflow1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{0} && binsof(underflow_cp)intersect{1};
      ignore_bins wr1_rd0_underflow1 = binsof(wr_en_cp)intersect{1} && binsof(rd_en_cp)intersect{0} && binsof(underflow_cp)intersect{1};
    }

    cross_wr_rd_wr_ack:       cross wr_en_cp, rd_en_cp, wr_ack_cp{
      ignore_bins wr0_rd1_ack_1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{1} && binsof(wr_ack_cp)intersect{1};
      ignore_bins wr0_rd0_ack_1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{0} && binsof(wr_ack_cp)intersect{1};
    }

  endgroup

  function new();
    fifo_cvr_gp = new();
  endfunction

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

  endclass
endpackage
```

- **Scoreboard package :**



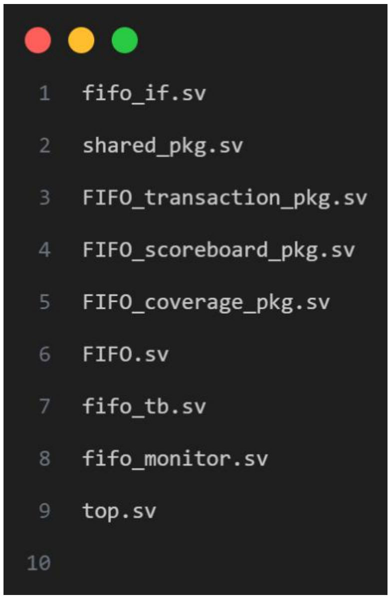
```
package FIFO_scoreboard_pkg ;
import FIFO_transaction_pkg ::*;
import shared_pkg ::*;
parameter FIFO_WIDTH = 16;
parameter FIFO_DEPTH = 8;


  localparam max_fifo_addr = $clog2(FIFO_DEPTH);

  logic [FIFO_WIDTH-1:0] data_out_ref;
  logic [FIFO_WIDTH-1:0] Queue [$] ;
  logic full_ref , empty_ref ;
  logic [max_fifo_addr:0] count;


class FIFO_scoreboard ;

  task check_data (input FIFO_transaction F_chk_txn);
    Reference_model (F_chk_txn);
    if ((data_out_ref !== F_chk_txn.data_out)) begin
          error_count ++ ;
          $display("   [ERROR] Mismatch at time %0t:", $time);
          $display("  data_out     => Expected: %0h, Actual: %0h", data_out_ref,    F_chk_txn.data_out);
    end
    else begin
      correct_count ++;
    end

  endtask

  function void Reference_model (input FIFO_transaction F_ref_txn);
      // Reset logic
      if (!F_ref_txn.rst_n) begin
        Queue <= {};
        count <= 0;
      end
      else begin
        // Write operation if not full
        if (F_ref_txn.wr_en && count <  FIFO_DEPTH) begin
          Queue.push_back(F_ref_txn.data_in);
          count <= Queue.size();
        end

        // Read operation if not empty
        if (F_ref_txn.rd_en && count != 0) begin
          data_out_ref <= Queue.pop_front();
          count <= Queue.size();
        end

      end

      // Update reference flags
      full_ref = (count == FIFO_DEPTH);
      empty_ref = (count == 0);
    endfunction

  endclass
endpackage
```

9) Do file :



```
vlib work
vlog -f src_files.list +cover -covercells +define+SIM
vsim -voptargs=+acc work.top  -cover -sv_seed random -l sim.FIFO_log
add wave /top/fifoif/*
run 0
add wave -position insertpoint  \
sim:/top/DUT/wr_ptr \
sim:/top/DUT/rd_ptr \
sim:/top/DUT/count\
add wave -position insertpoint  \
sim:/top/TEST/test_txn.RD_EN_ON_DIST \
sim:/top/TEST/test_txn.WR_EN_ON_DIST
add wave -position insertpoint  \
sim:/FIFO_scoreboard_pkg::data_out_ref
coverage save FIFO.ucdb -onexit -du FIFO
##run -all
##quit -sim
##vcover report FIFO_top.ucdb -details -annotate -all -output Coverage_FIFO_SV.txt
```

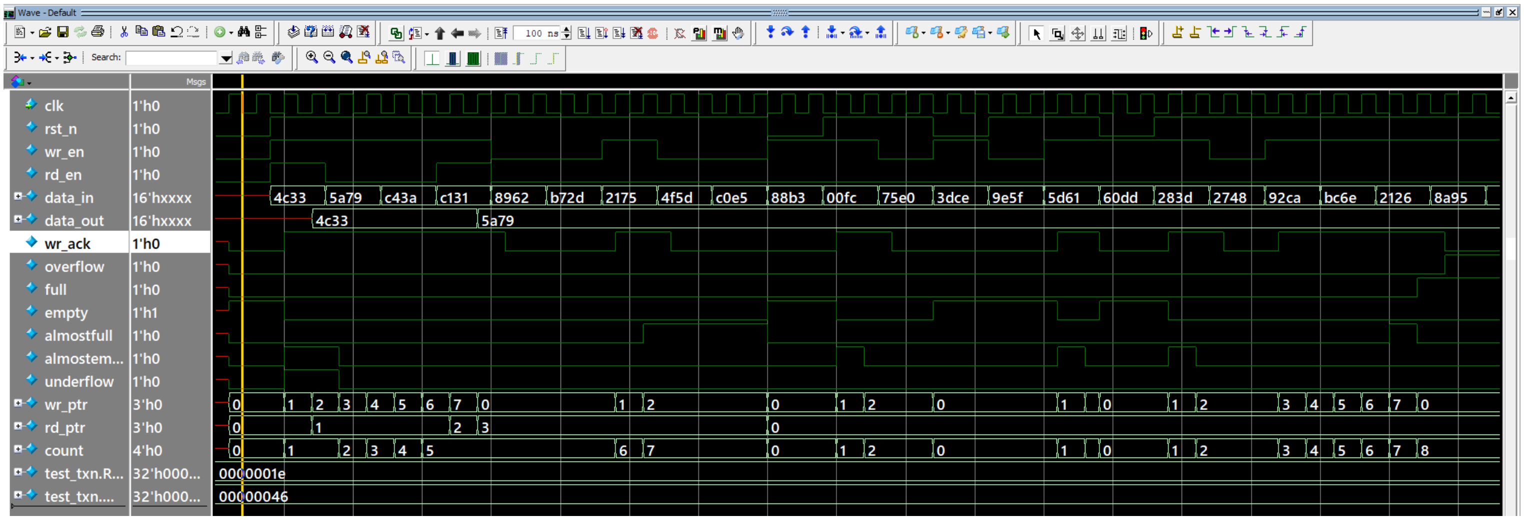
10) Source file :



```
fifo_if.sv
shared_pkg.sv
FIFO_transaction_pkg.sv
FIFO_scoreboard_pkg.sv
FIFO_coverage_pkg.sv
FIFO.sv
fifo_tb.sv
fifo_monitor.sv
top.sv

```

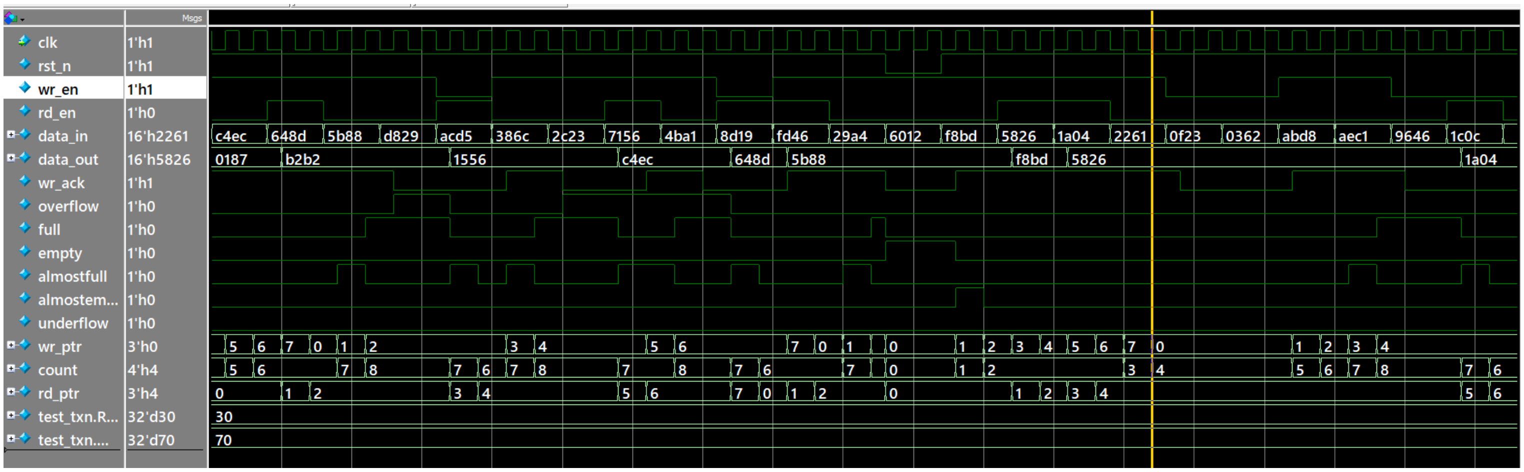
11) Questasim Snippets :

FIFO_1 : (When the reset is asserted the FIFO outputs (data_out & other flags) should be low .)


Wave - Default
Search:
100 ns
Msgs
data_in: 4c33 5a79 c43a c131 8962 b72d 2175 4f5d c0e5 88b3 00fc 75e0 3dce 9e5f 5d61 60dd 283d 2748 92ca bc6e 2126 8a95
data_out: 4c33 5a79
wr_ptr: 0 1 2 3 4 5 6 7 0 1 2 0 1 2 0 1 0 1 2 3 4 5 6 7 0
rd_ptr: 0 1 2 3 0
count: 0 1 2 3 4 5 6 7 0 1 2 0 1 0 1 2 3 4 5 6 7 8
0000001e
00000046

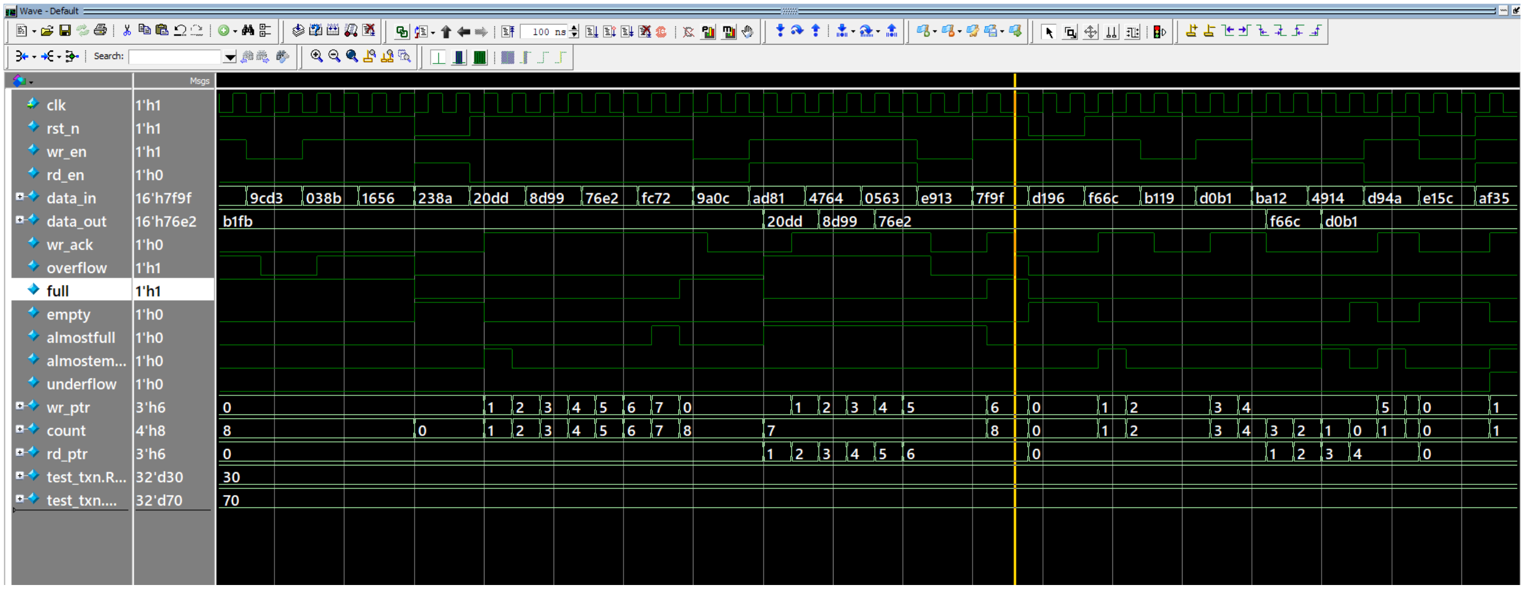
clk	1'h0
rst_n	1'h0
wr_en	1'h0
rd_en	1'h0
data_in	16'hxxxx
data_out	16'hxxxx
wr_ack	1'h0
overflow	1'h0
full	1'h0
empty	1'h1
almostfull	1'h0
almostem...	1'h0
underflow	1'h0
wr_ptr	3'h0
rd_ptr	3'h0
count	4'h0
test_txn.R...	32'h000...
test_txn....	32'h000...

FIFO_2 : (When the wr_en is asserted, the FIFO is not Full , Writing Operation takes place withl data_in input , wr_ack should be activated)


Msgs
data_in: c4ec 648d 5b88 d829 acd5 386c 2c23 7156 4ba1 8d19 fd46 29a4 6012 f8bd 5826 1a04 2261 0f23 0362 abd8 aec1 9646 1c0c
data_out: 0187 b2b2 1556 c4ec 648d 5b88 f8bd 5826 1a04
wr_ptr: 5 6 7 0 1 2 3 4 5 6 7 0 1 0 1 2 3 4 5 6 7 0 1 2 3 4
count: 5 6 7 8 7 6 7 8 7 8 7 6 7 0 1 2 3 4 5 6 7 8 7 6
rd_ptr: 0 1 2 3 4 5 6 7 0 1 2 0 1 2 3 4 5 6
30
70

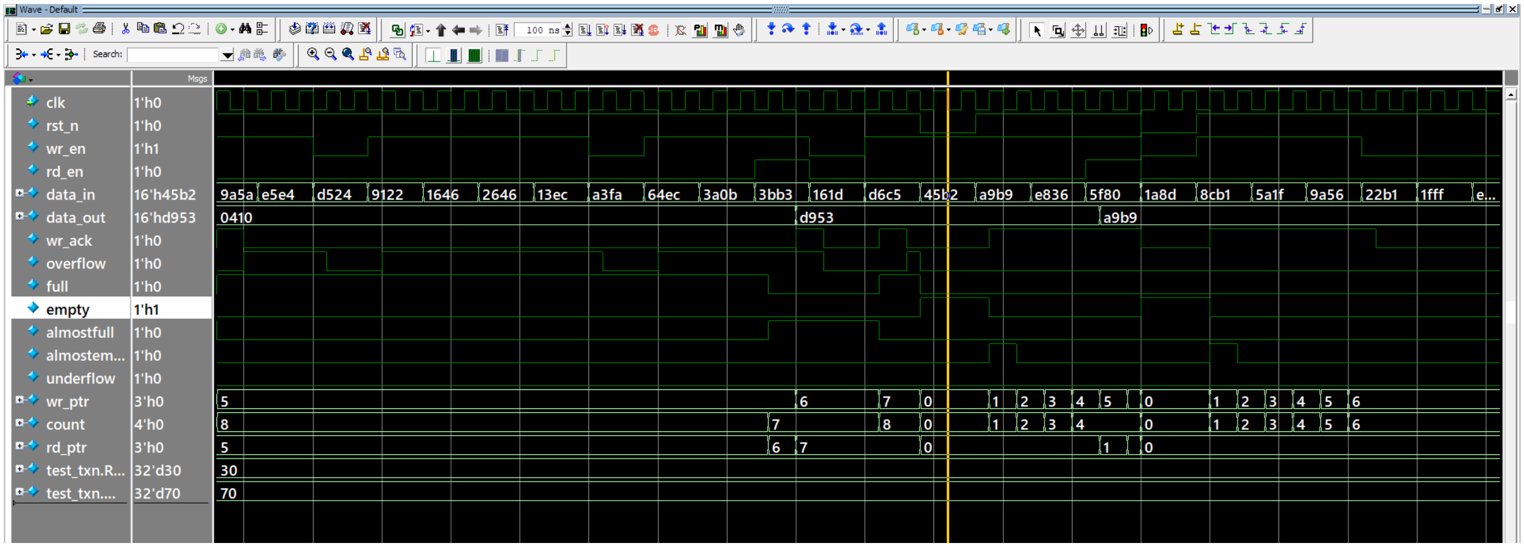
clk	1'h1
rst_n	1'h1
wr_en	1'h1
rd_en	1'h0
data_in	16'h2261
data_out	16'h5826
wr_ack	1'h1
overflow	1'h0
full	1'h0
empty	1'h0
almostfull	1'h0
almostem...	1'h0
underflow	1'h0
wr_ptr	3'h0
count	4'h4
rd_ptr	3'h4
test_txn.R...	32'd30
test_txn....	32'd70

FIFO_3: (When the FIFO is full , the full flag should be activated)

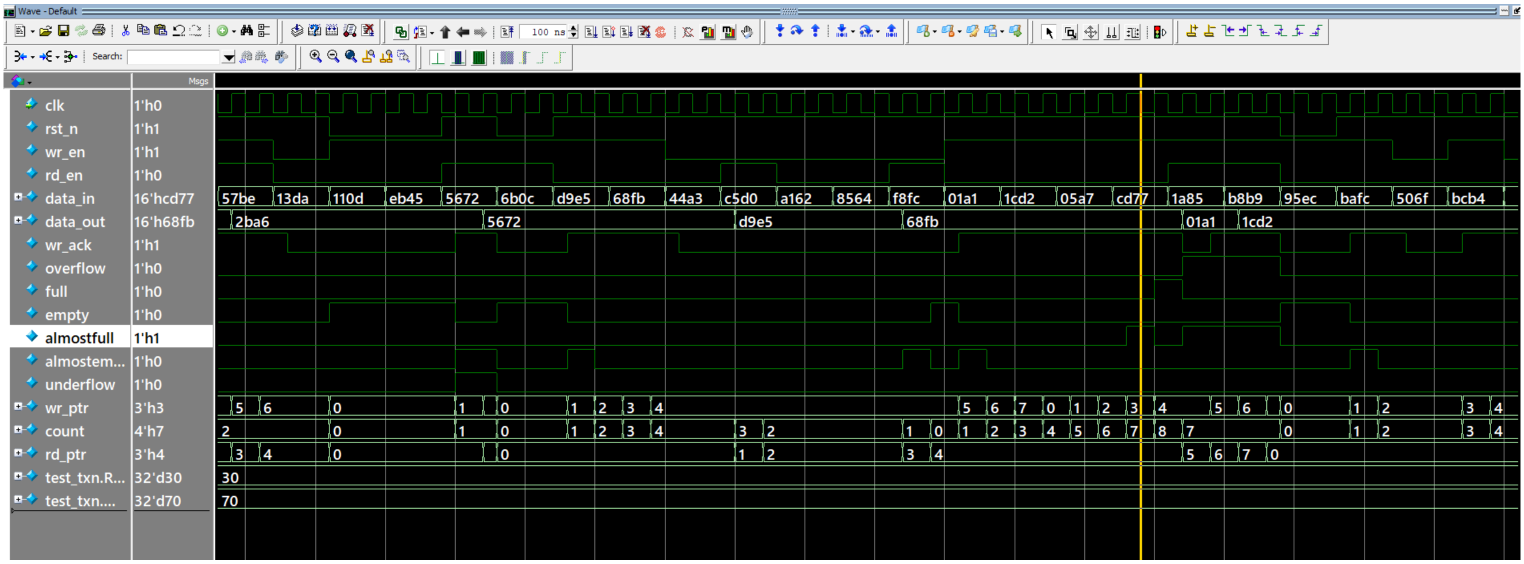

Wave - Default
Search:
100 ns
Msgs
data_in: 9cd3 038b 1656 238a 20dd 8d99 76e2 fc72 9a0c ad81 4764 0563 e913 7f9f d196 f66c b119 d0b1 ba12 4914 d94a e15c af35
data_out: b1fb 20dd 8d99 76e2 f66c d0b1
wr_ptr: 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 0 1 2 3 4 5 0 1
count: 8 0 1 2 3 4 5 6 7 8 7 8 0 1 2 3 4 3 2 1 0 1 0 1
rd_ptr: 0 1 2 3 4 5 6 0 1 2 3 4 0
30
70

clk	1'h1
rst_n	1'h1
wr_en	1'h1
rd_en	1'h0
data_in	16'h7f9f
data_out	16'h76e2
wr_ack	1'h0
overflow	1'h1
full	1'h1
empty	1'h0
almostfull	1'h0
almostem...	1'h0
underflow	1'h0
wr_ptr	3'h6
count	4'h8
rd_ptr	3'h6
test_txn.R...	32'd30
test_txn....	32'd70

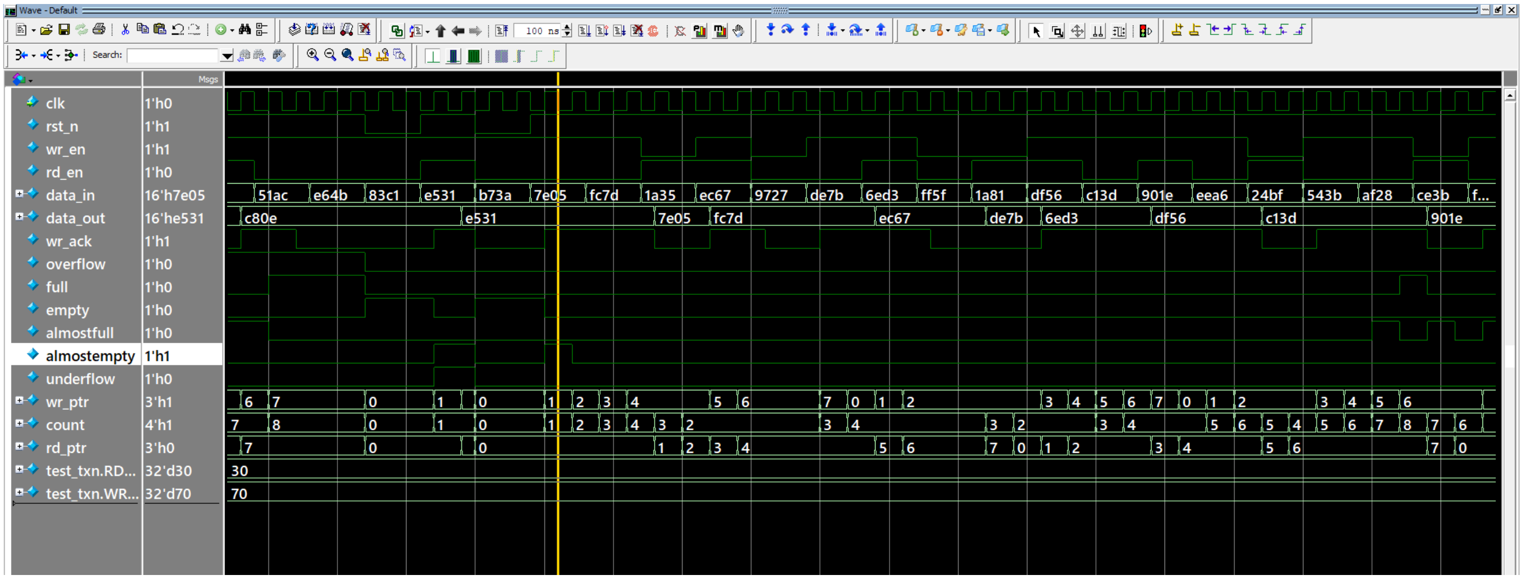
FIFO_4 : (When the FIFO is empty , the empty flag should be activated)



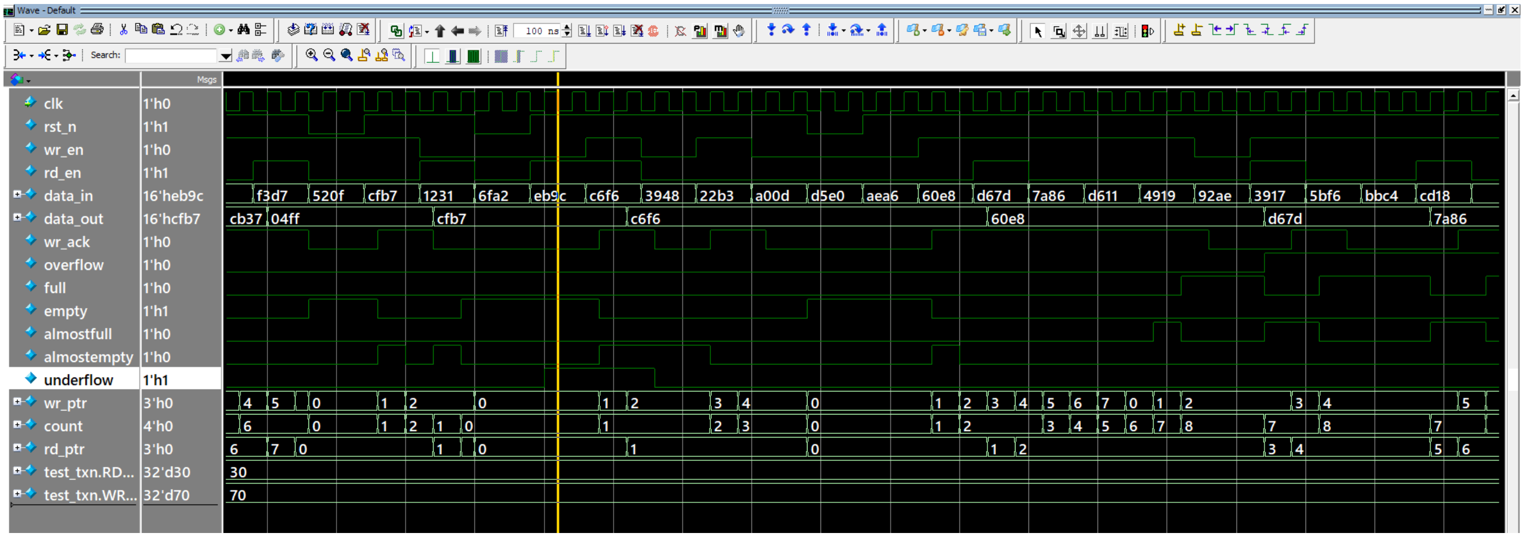
FIFO_5 : (When the FIFO has only 1 space left , the almostfull flag should be activated)



FIFO_6 : (When the FIFO has only 1 space occupied , the almostempty flag should be activated)



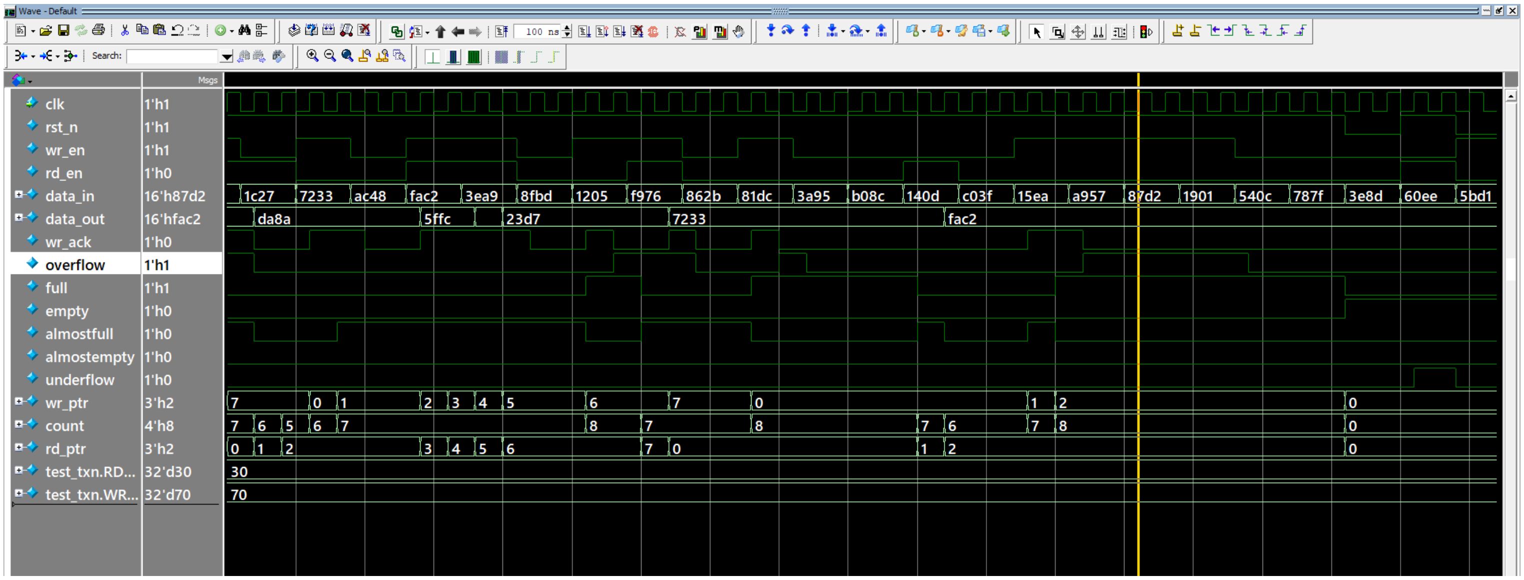
FIFO_7 : (When the FIFO is empty and the rd_en signal is high , underflow signal should be activated)



Wave - Default	Msgs
clk	1'h0
rst_n	1'h1
wr_en	1'h0
rd_en	1'h1
data_in	16'heb9c
data_out	16'hcfb7
wr_ack	1'h0
overflow	1'h0
full	1'h0
empty	1'h1
almostfull	1'h0
almostempty	1'h0
underflow	1'h1
wr_ptr	3'h0
count	4'h0
rd_ptr	3'h0
test_txn.RD...	32'd30
test_txn.WR...	32'd70

f3d7 520f cfb7 1231 6fa2 eb9c c6f6 3948 22b3 a00d d5e0 aea6 60e8 d67d 7a86 d611 4919 92ae 3917 5bf6 bbc4 cd18
cb37 04ff cfb7 c6f6 60e8 d67d 7a86
4 5 0 1 2 0 1 2 3 4 0 1 2 3 4 5 6 7 0 1 2 3 4 5
6 0 1 2 1 0 1 2 3 0 1 2 3 4 5 6 7 8 7 8 7
6 7 0 1 0 1 0 1 2 3 4 5 6
30
70

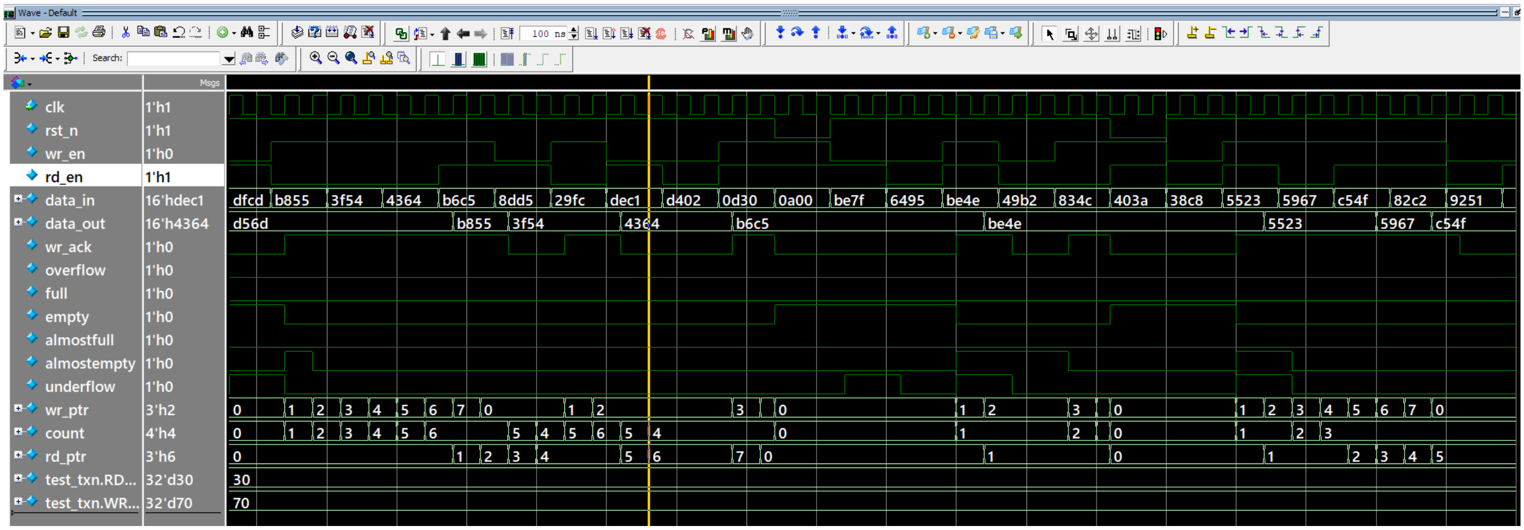
FIFO_8 : (When the FIFO is full and the wr_en signal is high , overflow signal should be activated)



Wave - Default	Msgs
clk	1'h1
rst_n	1'h1
wr_en	1'h1
rd_en	1'h0
data_in	16'h87d2
data_out	16'hfac2
wr_ack	1'h0
overflow	1'h1
full	1'h1
empty	1'h0
almostfull	1'h0
almostempty	1'h0
underflow	1'h0
wr_ptr	3'h2
count	4'h8
rd_ptr	3'h2
test_txn.RD...	32'd30
test_txn.WR...	32'd70

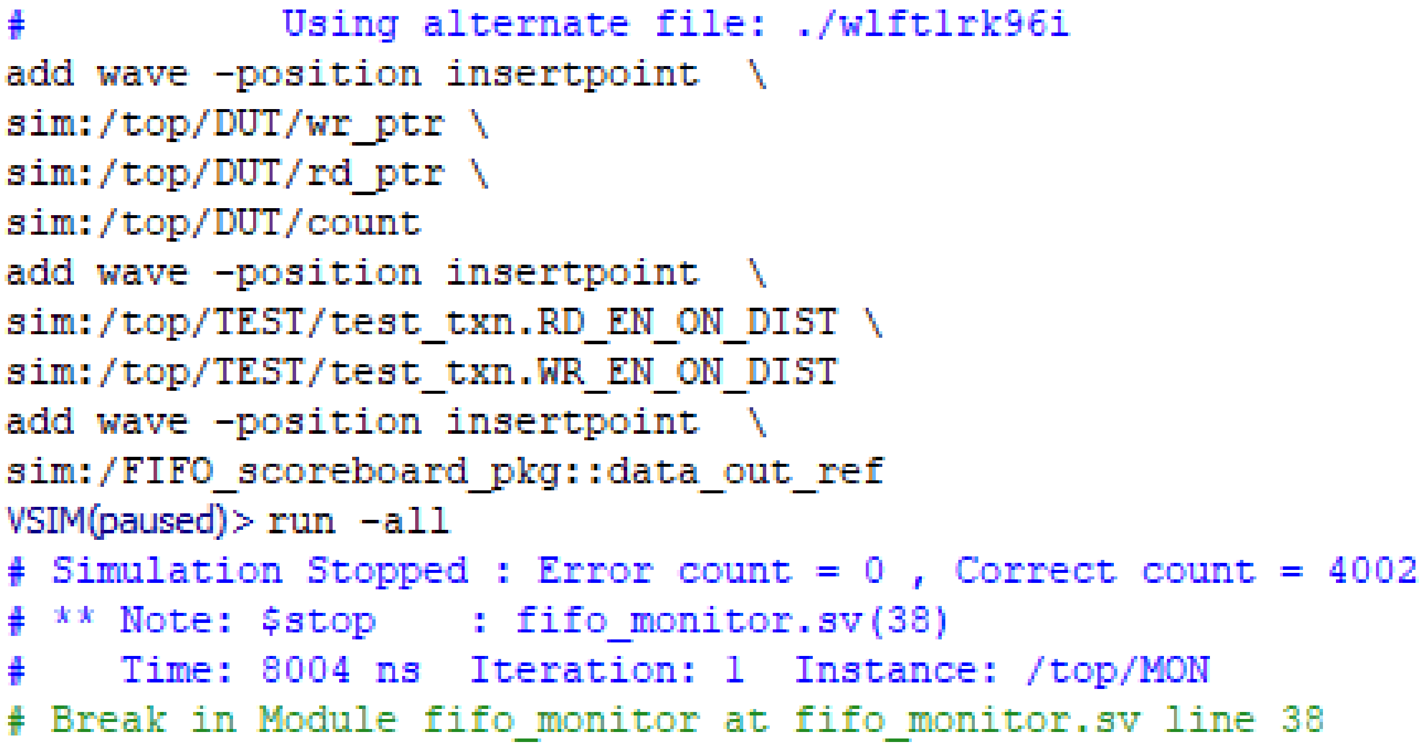
1c27 7233 ac48 fac2 3ea9 8fbd 1205 f976 862b 81dc 3a95 b08c 140d c03f 15ea a957 87d2 1901 540c 787f 3e8d 60ee 5bd1
da8a 5ffc 23d7 7233 fac2
7 0 1 2 3 4 5 6 7 0 1 2 0
7 6 5 6 7 8 7 8 7 6 7 8 0
0 1 2 3 4 5 6 7 0 1 2 0
30
70

FIFO_9 : (When the FIFO is not empty & rd_en signal is high the data_out should take the value of the FIFO element with the rd_ptr address and the rd_ptr is incremented)



Wave - Default	Msgs
clk	1'h1
rst_n	1'h1
wr_en	1'h0
rd_en	1'h1
data_in	16'hdec1
data_out	16'h4364
wr_ack	1'h0
overflow	1'h0
full	1'h0
empty	1'h0
almostfull	1'h0
almostempty	1'h0
underflow	1'h0
wr_ptr	3'h2
count	4'h4
rd_ptr	3'h6
test_txn.RD...	32'd30
test_txn.WR...	32'd70

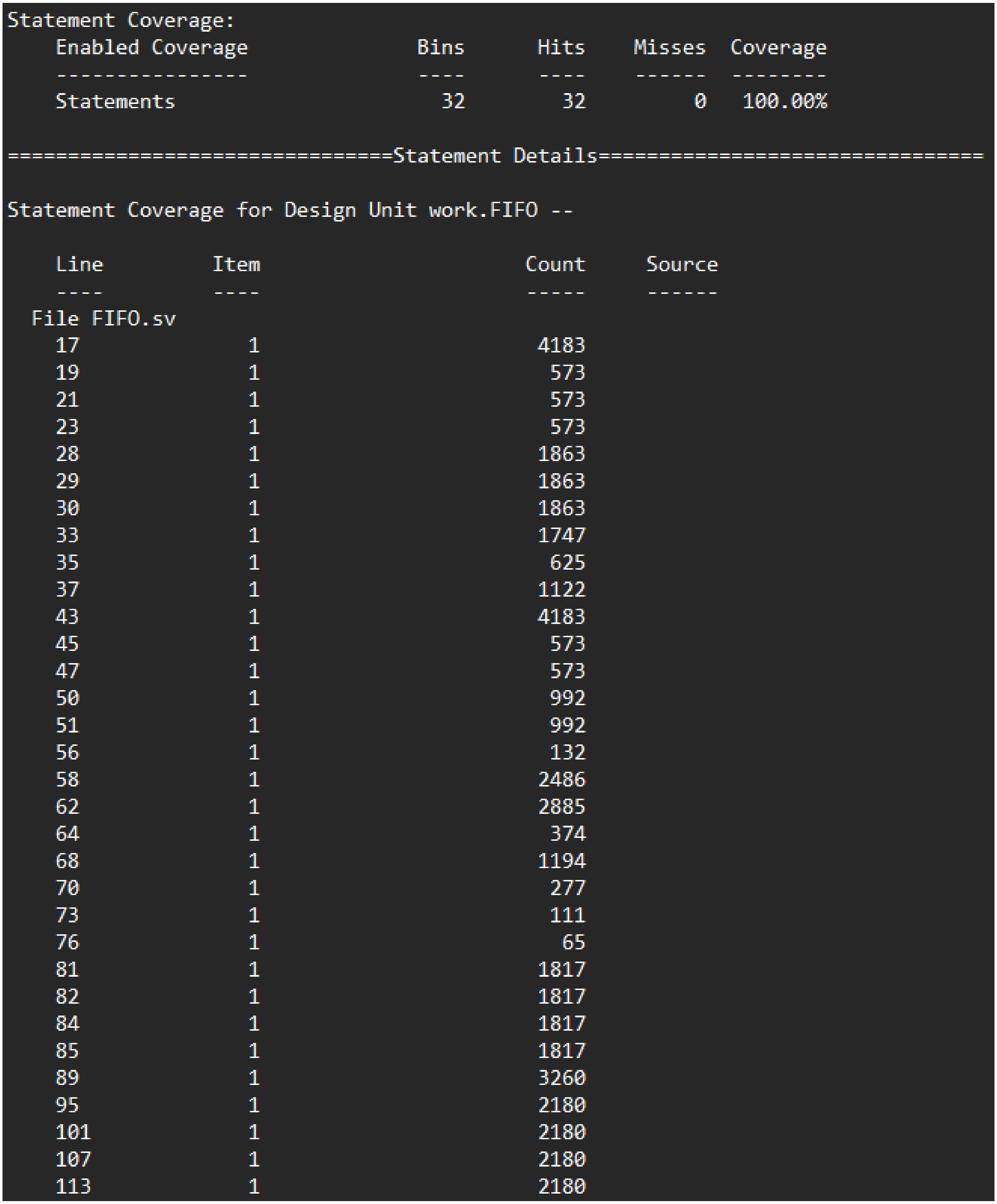
dfcd b855 3f54 4364 b6c5 8dd5 29fc dec1 d402 0d30 0a00 be7f 6495 be4e 49b2 834c 403a 38c8 5523 5967 c54f 82c2 9251
d56d b855 3f54 4364 b6c5 be4e 5523 5967 c54f
0 1 2 3 4 5 6 7 0 1 2 3 0 1 2 3 0 1 2 3 4 5 6 7 0
0 1 2 3 4 5 6 5 4 5 6 5 4 0 1 2 0 1 2 3
0 1 2 3 4 5 6 7 0 1 0 1 2 3 4 5
30
70



```
#            Using alternate file: ./wlftlrk96i
add wave -position insertpoint  \
sim:/top/DUT/wr_ptr \
sim:/top/DUT/rd_ptr \
sim:/top/DUT/count
add wave -position insertpoint  \
sim:/top/TEST/test_txn.RD_EN_ON_DIST \
sim:/top/TEST/test_txn.WR_EN_ON_DIST
add wave -position insertpoint  \
sim:/FIFO_scoreboard_pkg::data_out_ref
VSIM(paused)> run -all
# Simulation Stopped : Error count = 0 , Correct count = 4002
# ** Note: $stop    : fifo_monitor.sv(38)
#    Time: 8004 ns  Iteration: 1  Instance: /top/MON
# Break in Module fifo_monitor at fifo_monitor.sv line 38
```

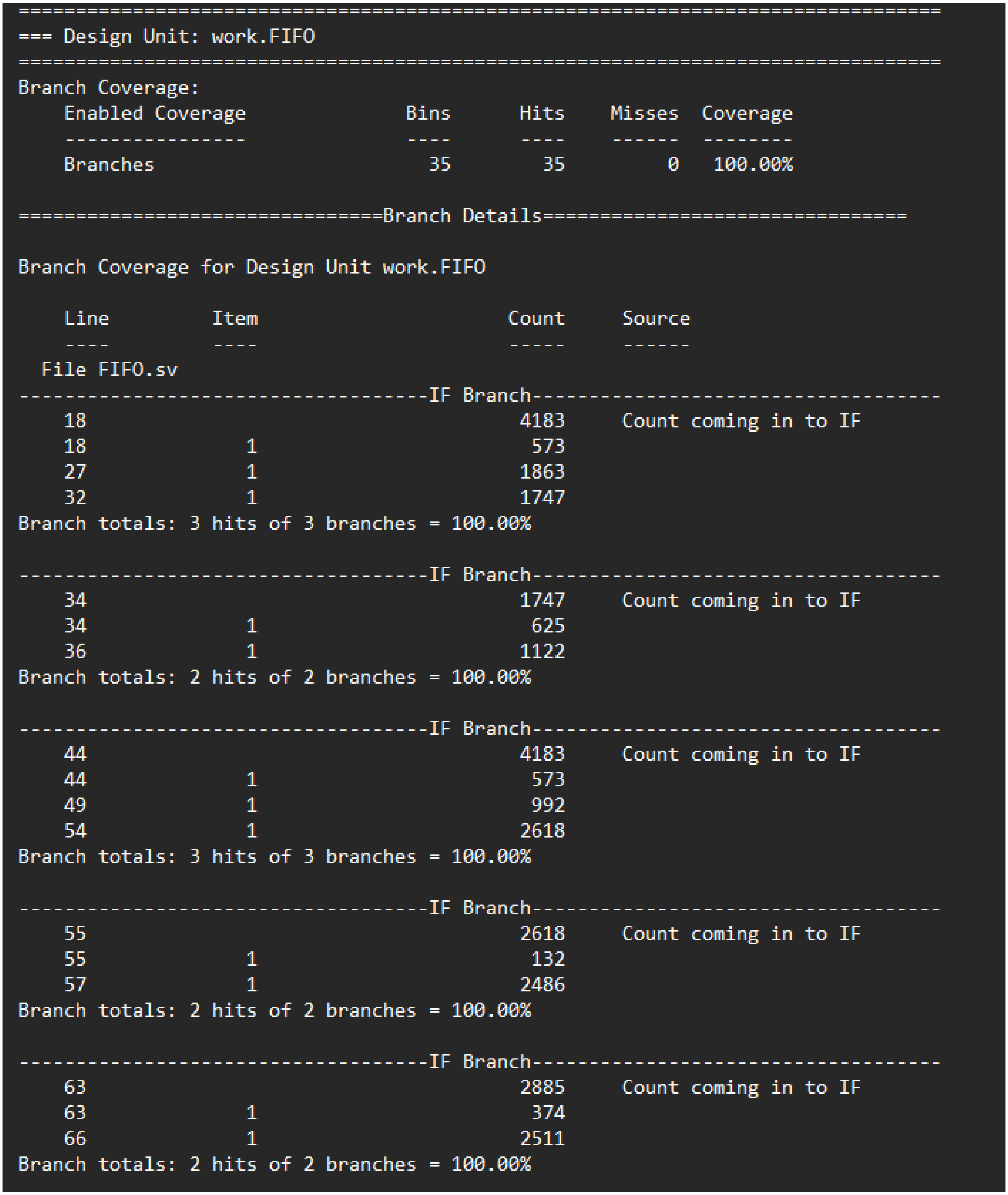
12) Code Coverage :

- **Statement Coverage :**

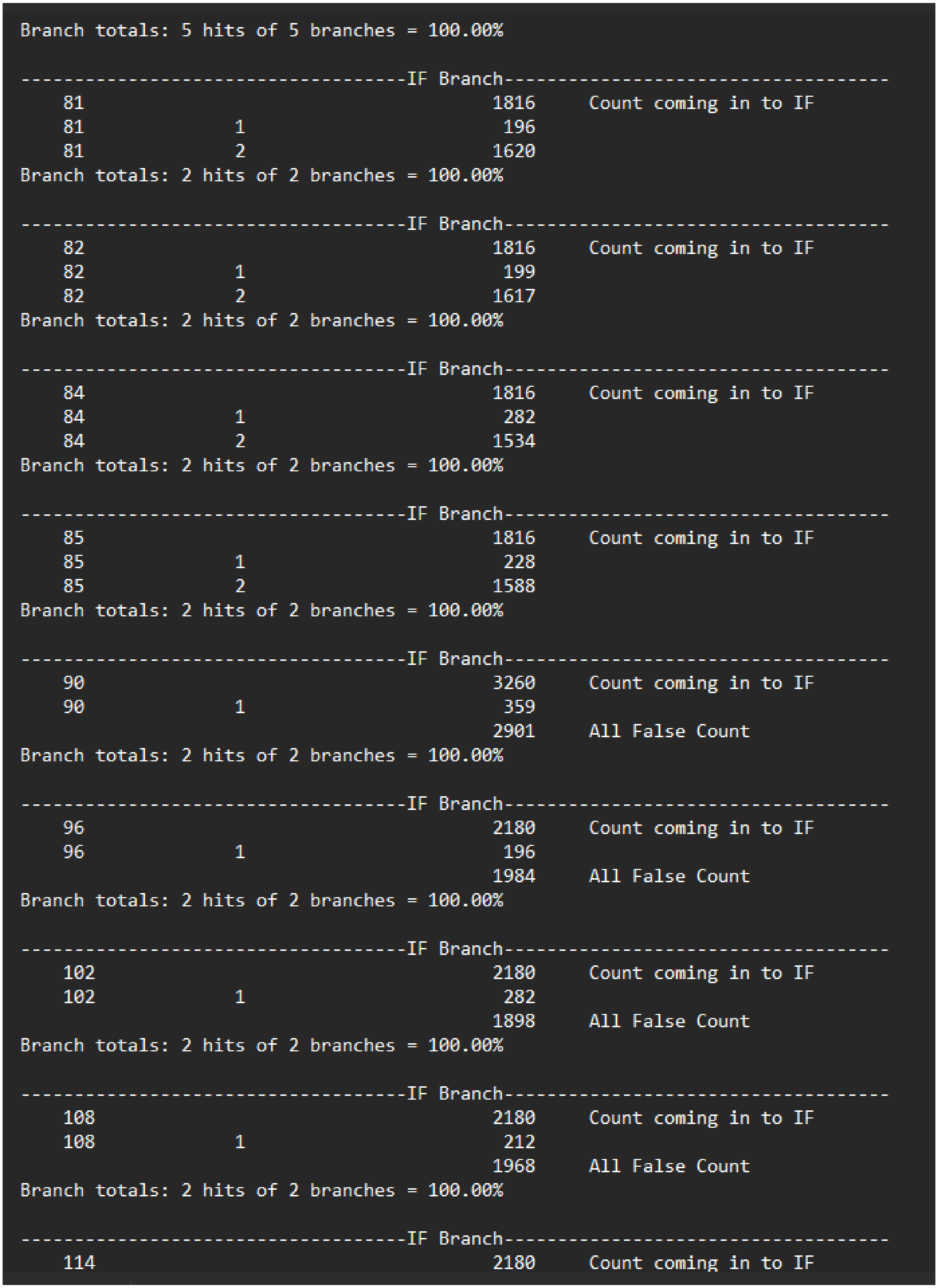


```
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                      32        32         0   100.00%

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

Statement Coverage for Design Unit work.FIFO --

    Line         Item                      Count     Source 
    ----         ----                      -----     ------ 
  File FIFO.sv
    17              1                       4183
    19              1                        573
    21              1                        573
    23              1                        573
    28              1                       1863
    29              1                       1863
    30              1                       1863
    33              1                       1747
    35              1                        625
    37              1                       1122
    43              1                       4183
    45              1                        573
    47              1                        573
    50              1                        992
    51              1                        992
    56              1                        132
    58              1                       2486
    62              1                       2885
    64              1                        374
    68              1                       1194
    70              1                        277
    73              1                        111
    76              1                         65
    81              1                       1817
    82              1                       1817
    84              1                       1817
    85              1                       1817
    89              1                       3260
    95              1                       2180
    101             1                       2180
    107             1                       2180
    113             1                       2180
```

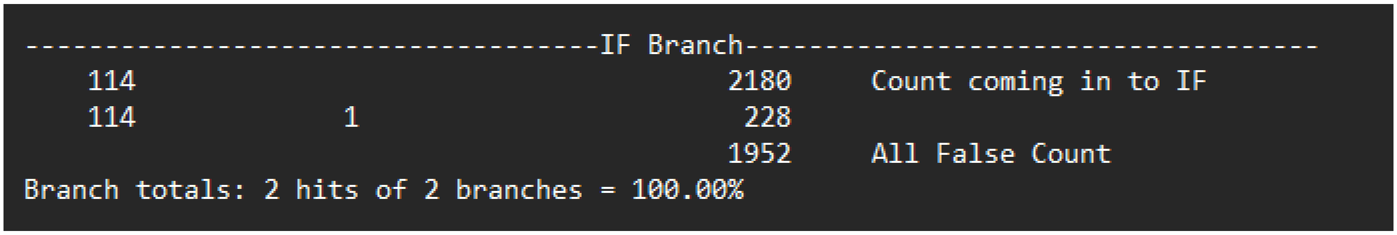
- **Branch Coverage :**



```
==================================================================================
=== Design Unit: work.FIFO
==================================================================================
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        35        35         0   100.00%

================================Branch Details================================

Branch Coverage for Design Unit work.FIFO

    Line         Item                      Count     Source 
    ----         ----                      -----     ------ 
  File FIFO.sv
------------------------------------IF Branch------------------------------------
    18                                      4183     Count coming in to IF
    18              1                        573     
    27              1                       1863     
    32              1                       1747     
Branch totals: 3 hits of 3 branches = 100.00%

------------------------------------IF Branch------------------------------------
    34                                      1747     Count coming in to IF
    34              1                        625     
    36              1                       1122     
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    44                                      4183     Count coming in to IF
    44              1                        573     
    49              1                        992     
    54              1                       2618     
Branch totals: 3 hits of 3 branches = 100.00%

------------------------------------IF Branch------------------------------------
    55                                      2618     Count coming in to IF
    55              1                        132     
    57              1                       2486     
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    63                                      2885     Count coming in to IF
    63              1                        374     
    66              1                       2511     
Branch totals: 2 hits of 2 branches = 100.00%
```

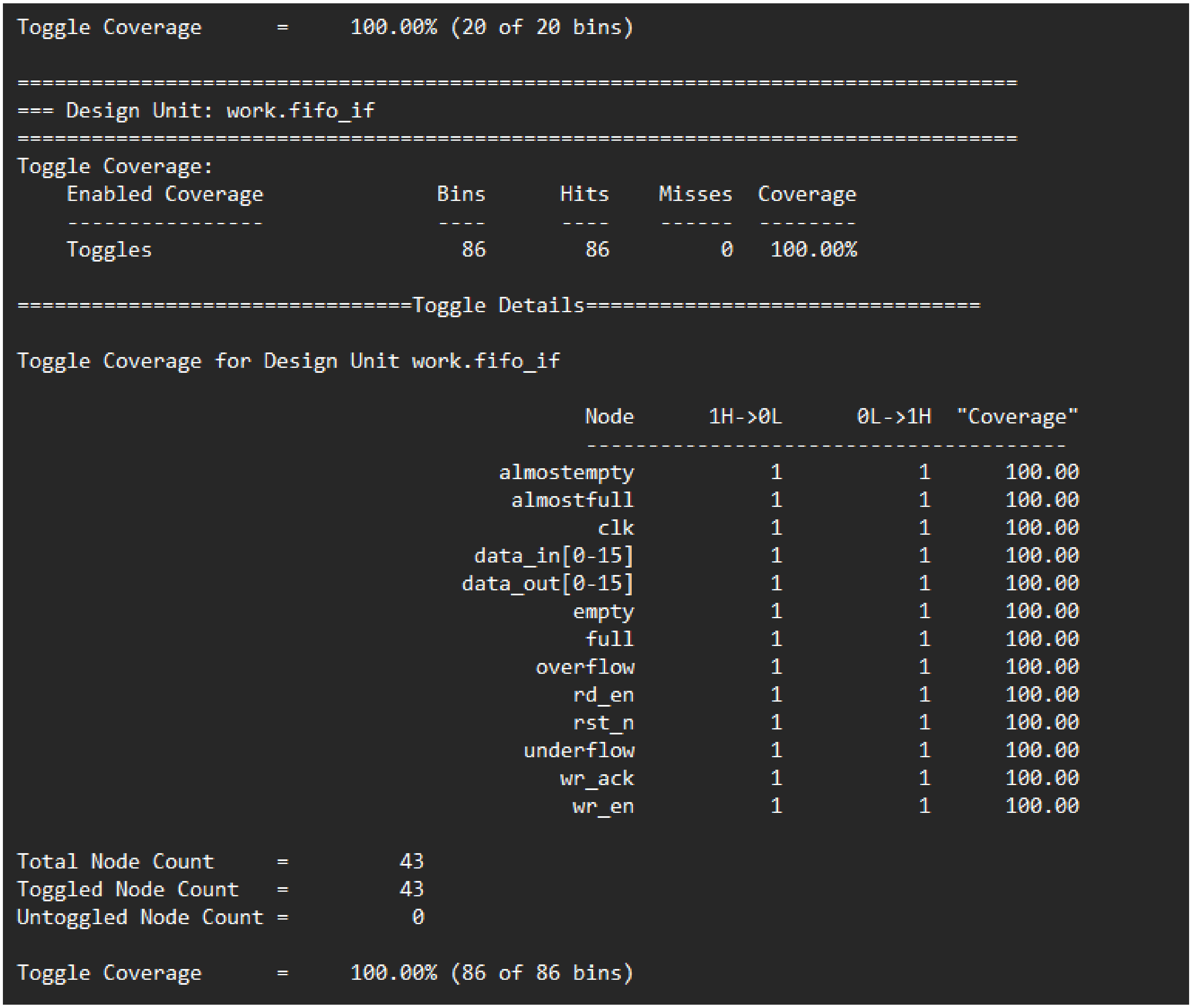


```
Branch totals: 5 hits of 5 branches = 100.00%

------------------------------------IF Branch------------------------------------
    81                                      1816     Count coming in to IF
    81              1                        196
    81              2                       1620
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    82                                      1816     Count coming in to IF
    82              1                        199
    82              2                       1617
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    84                                      1816     Count coming in to IF
    84              1                        282
    84              2                       1534
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    85                                      1816     Count coming in to IF
    85              1                        228
    85              2                       1588
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    90                                      3260     Count coming in to IF
    90              1                        359
                                            2901     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    96                                      2180     Count coming in to IF
    96              1                        196
                                            1984     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    102                                     2180     Count coming in to IF
    102             1                        282
                                            1898     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    108                                     2180     Count coming in to IF
    108             1                        212
                                            1968     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    114                                     2180     Count coming in to IF
```



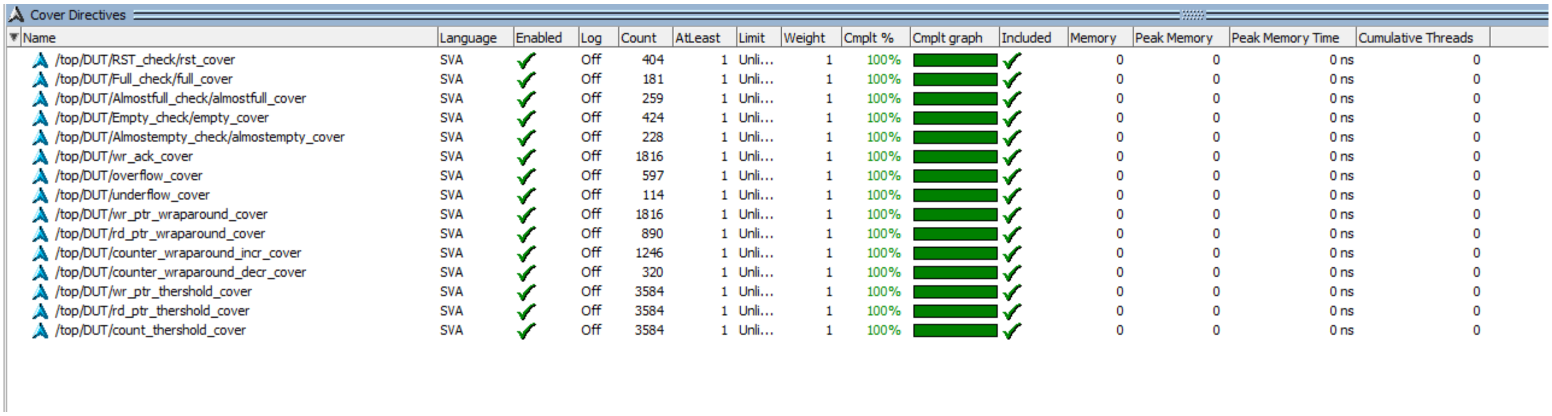
```
------------------------------------IF Branch------------------------------------
    114                                     2180     Count coming in to IF
    114             1                        228
                                            1952     All False Count
Branch totals: 2 hits of 2 branches = 100.00%
```

- **<u>Toggle Coverage :</u>**



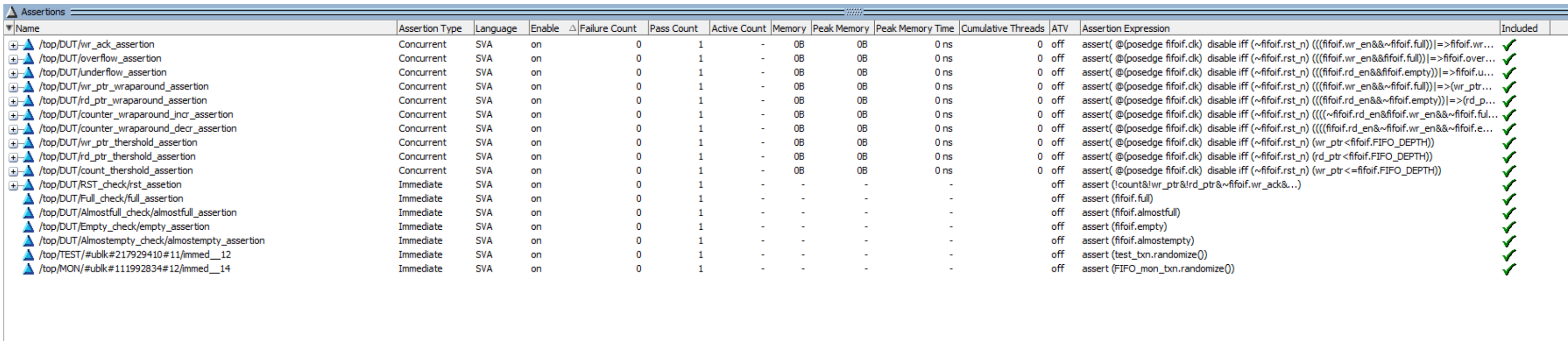
```
Toggle Coverage      =     100.00% (20 of 20 bins)

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

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

Toggle Coverage for Design Unit work.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[0-15]           1           1      100.00
                                    data_out[0-15]           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)
```

13) Assertions Coverage :



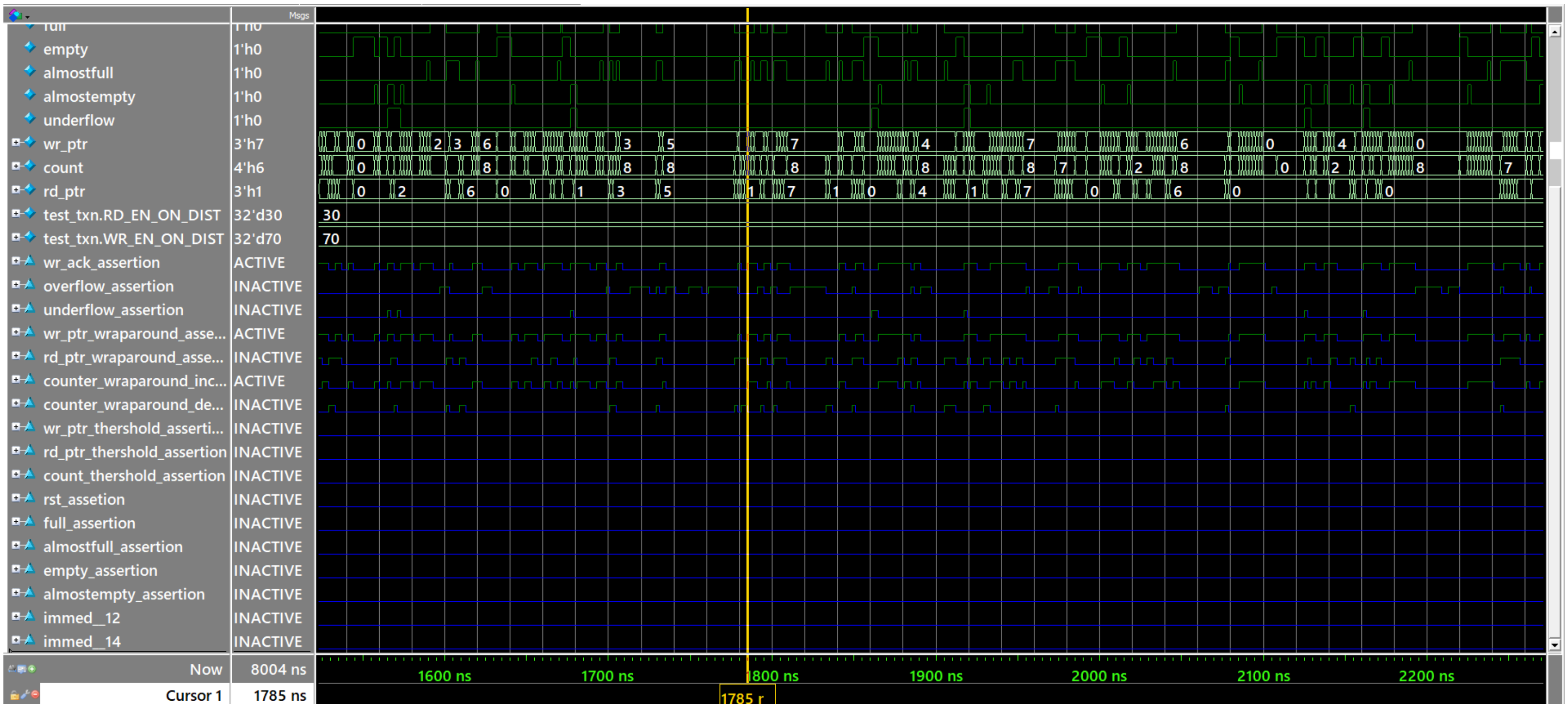
Cover Directives

Name	Language	Enabled	Log	Count	AtLeast	Limit	Weight	Cmplt %	Cmplt graph	Included	Memory	Peak Memory	Peak Memory Time	Cumulative Threads
/top/DUT/RST_check/rst_cover	SVA	✓	Off	404	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/Full_check/full_cover	SVA	✓	Off	181	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/Almostfull_check/almostfull_cover	SVA	✓	Off	259	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/Empty_check/empty_cover	SVA	✓	Off	424	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/Almostempty_check/almostempty_cover	SVA	✓	Off	228	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/wr_ack_cover	SVA	✓	Off	1816	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/overflow_cover	SVA	✓	Off	597	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/underflow_cover	SVA	✓	Off	114	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/wr_ptr_wraparound_cover	SVA	✓	Off	1816	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/rd_ptr_wraparound_cover	SVA	✓	Off	890	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/counter_wraparound_incr_cover	SVA	✓	Off	1246	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/counter_wraparound_decr_cover	SVA	✓	Off	320	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/wr_ptr_thershold_cover	SVA	✓	Off	3584	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/rd_ptr_thershold_cover	SVA	✓	Off	3584	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0
/top/DUT/count_thershold_cover	SVA	✓	Off	3584	1	Unli...	1	100%	🟩	✓	0	0	0 ns	0

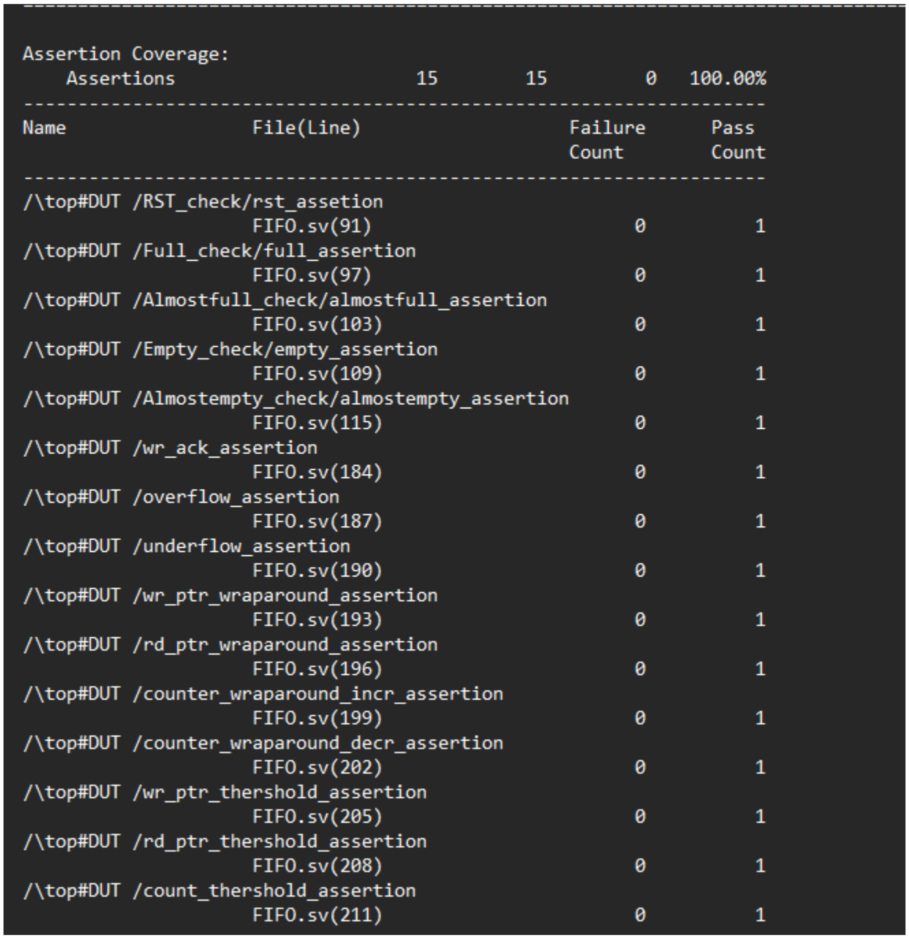


Assertions

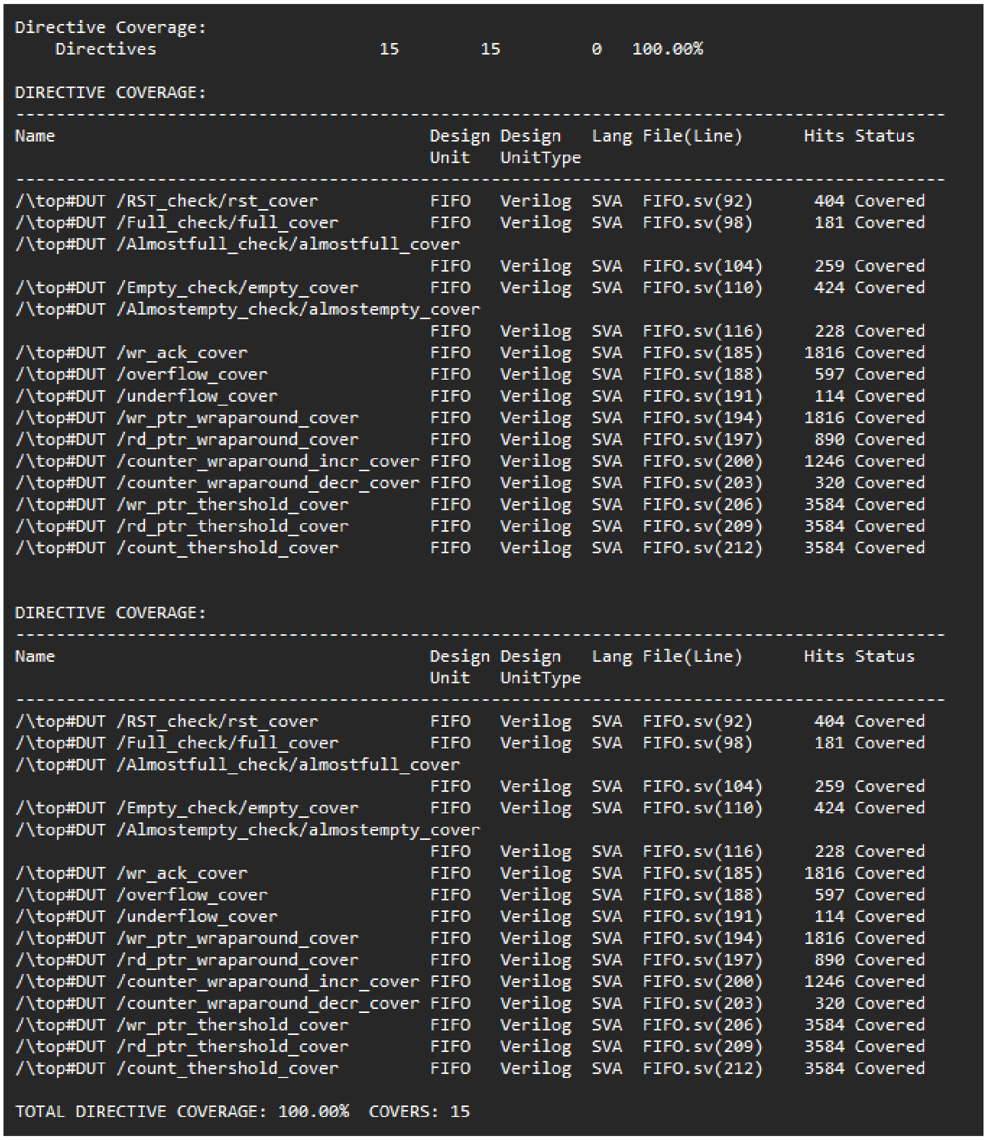
Name	Assertion Type	Language	Enable	Failure Count	Pass Count	Active Count	Memory	Peak Memory	Peak Memory Time	Cumulative Threads	ATV	Assertion Expression	Included
/top/DUT/wr_ack_assertion	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifoif.clk) disable iff (~fifoif.rst_n) (((fifoif.wr_en&&~fifoif.full))|=>fifoif.wr...	✓
/top/DUT/overflow_assertion	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifoif.clk) disable iff (~fifoif.rst_n) (((fifoif.wr_en&&fifoif.full))|=>fifoif.over...	✓
/top/DUT/underflow_assertion	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifoif.clk) disable iff (~fifoif.rst_n) (((fifoif.rd_en&&fifoif.empty))|=>fifoif.u...	✓
/top/DUT/wr_ptr_wraparound_assertion	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifoif.clk) disable iff (~fifoif.rst_n) (((fifoif.wr_en&&~fifoif.full))|=>(wr_ptr...	✓
/top/DUT/rd_ptr_wraparound_assertion	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifoif.clk) disable iff (~fifoif.rst_n) (((fifoif.rd_en&&~fifoif.empty))|=>(rd_p...	✓
/top/DUT/counter_wraparound_incr_assertion	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifoif.clk) disable iff (~fifoif.rst_n) ((((~fifoif.rd_en&fifoif.wr_en&&~fifoif.ful...	✓
/top/DUT/counter_wraparound_decr_assertion	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifoif.clk) disable iff (~fifoif.rst_n) ((((fifoif.rd_en&~fifoif.wr_en&&~fifoif.e...	✓
/top/DUT/wr_ptr_thershold_assertion	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifoif.clk) disable iff (~fifoif.rst_n) (wr_ptr<fifoif.FIFO_DEPTH))	✓
/top/DUT/rd_ptr_thershold_assertion	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifoif.clk) disable iff (~fifoif.rst_n) (rd_ptr<fifoif.FIFO_DEPTH))	✓
/top/DUT/count_thershold_assertion	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert(@(posedge fifoif.clk) disable iff (~fifoif.rst_n) (wr_ptr<=fifoif.FIFO_DEPTH))	✓
/top/DUT/RST_check/rst_assetion	Immediate	SVA	on	0	1	-	-	-	-		off	assert (!count&!wr_ptr&!rd_ptr&~fifoif.wr_ack&...)	✓
/top/DUT/Full_check/full_assertion	Immediate	SVA	on	0	1	-	-	-	-		off	assert (fifoif.full)	✓
/top/DUT/Almostfull_check/almostfull_assertion	Immediate	SVA	on	0	1	-	-	-	-		off	assert (fifoif.almostfull)	✓
/top/DUT/Empty_check/empty_assertion	Immediate	SVA	on	0	1	-	-	-	-		off	assert (fifoif.empty)	✓
/top/DUT/Almostempty_check/almostempty_assertion	Immediate	SVA	on	0	1	-	-	-	-		off	assert (fifoif.almostempty)	✓
/top/TEST/#ublk#217929410#11/immed__12	Immediate	SVA	on	0	1	-	-	-	-		off	assert (test_txn.randomize())	✓
/top/MON/#ublk#111992834#12/immed__14	Immediate	SVA	on	0	1	-	-	-	-		off	assert (FIFO_mon_txn.randomize())	✓



	Msgs
full	1'h0
empty	1'h0
almostfull	1'h0
almostempty	1'h0
underflow	1'h0
wr_ptr	3'h7
count	4'h6
rd_ptr	3'h1
test_txn.RD_EN_ON_DIST	32'd30
test_txn.WR_EN_ON_DIST	32'd70
wr_ack_assertion	ACTIVE
overflow_assertion	INACTIVE
underflow_assertion	INACTIVE
wr_ptr_wraparound_asse...	ACTIVE
rd_ptr_wraparound_asse...	INACTIVE
counter_wraparound_inc...	ACTIVE
counter_wraparound_de...	INACTIVE
wr_ptr_thershold_asserti...	INACTIVE
rd_ptr_thershold_assertion	INACTIVE
count_thershold_assertion	INACTIVE
rst_assetion	INACTIVE
full_assertion	INACTIVE
almostfull_assertion	INACTIVE
empty_assertion	INACTIVE
almostempty_assertion	INACTIVE
immed__12	INACTIVE
immed__14	INACTIVE
Now	8004 ns
Cursor 1	1785 ns

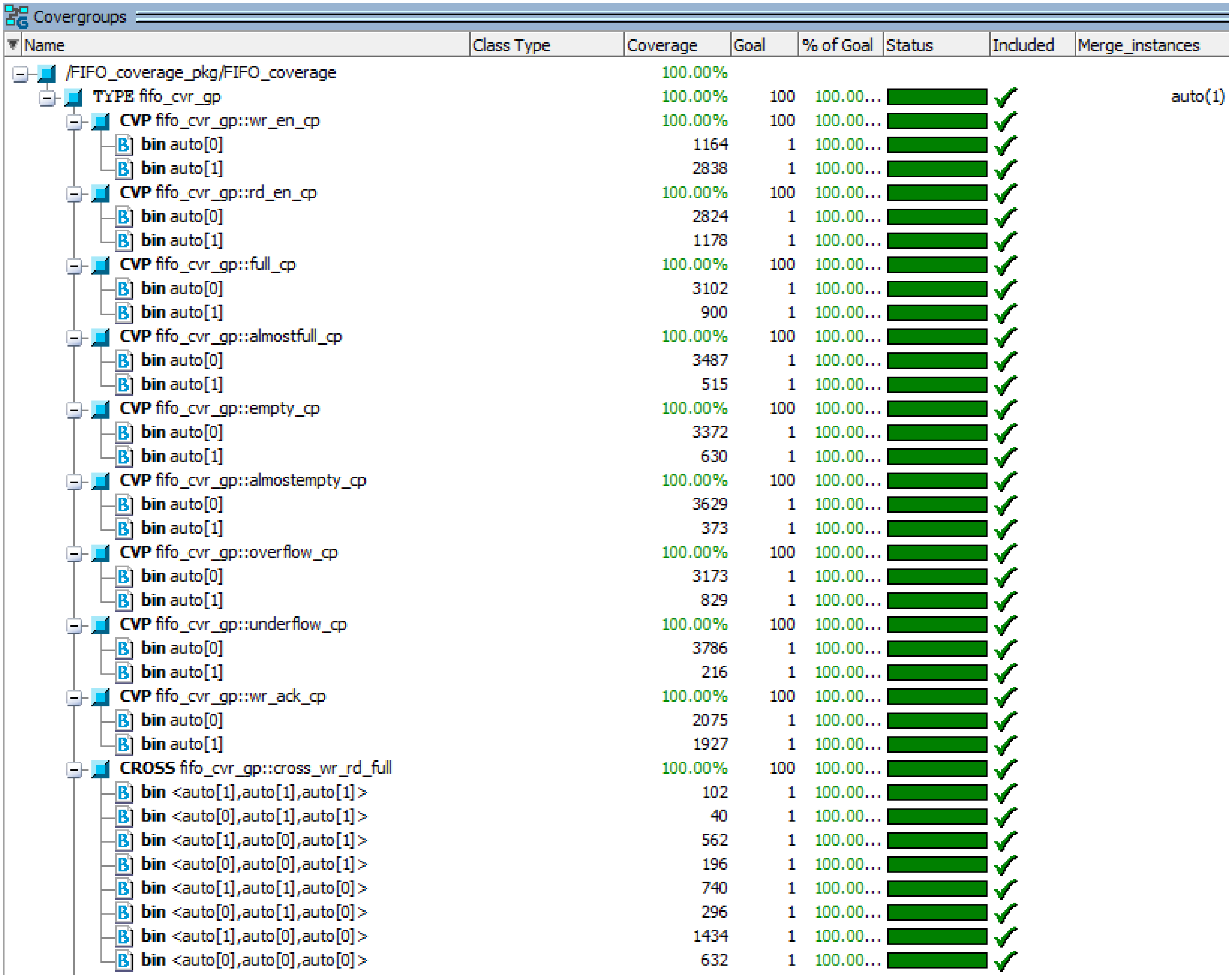


```
---------------------------------------------------------------------------------

Assertion Coverage:
    Assertions                      15         15          0   100.00%
--------------------------------------------------------------------
Name                 File(Line)                   Failure      Pass
                                                  Count        Count
--------------------------------------------------------------------
/\top#DUT /RST_check/rst_assetion
                     FIFO.sv(91)                        0           1
/\top#DUT /Full_check/full_assertion
                     FIFO.sv(97)                        0           1
/\top#DUT /Almostfull_check/almostfull_assertion
                     FIFO.sv(103)                       0           1
/\top#DUT /Empty_check/empty_assertion
                     FIFO.sv(109)                       0           1
/\top#DUT /Almostempty_check/almostempty_assertion
                     FIFO.sv(115)                       0           1
/\top#DUT /wr_ack_assertion
                     FIFO.sv(184)                       0           1
/\top#DUT /overflow_assertion
                     FIFO.sv(187)                       0           1
/\top#DUT /underflow_assertion
                     FIFO.sv(190)                       0           1
/\top#DUT /wr_ptr_wraparound_assertion
                     FIFO.sv(193)                       0           1
/\top#DUT /rd_ptr_wraparound_assertion
                     FIFO.sv(196)                       0           1
/\top#DUT /counter_wraparound_incr_assertion
                     FIFO.sv(199)                       0           1
/\top#DUT /counter_wraparound_decr_assertion
                     FIFO.sv(202)                       0           1
/\top#DUT /wr_ptr_thershold_assertion
                     FIFO.sv(205)                       0           1
/\top#DUT /rd_ptr_thershold_assertion
                     FIFO.sv(208)                       0           1
/\top#DUT /count_thershold_assertion
                     FIFO.sv(211)                       0           1
```



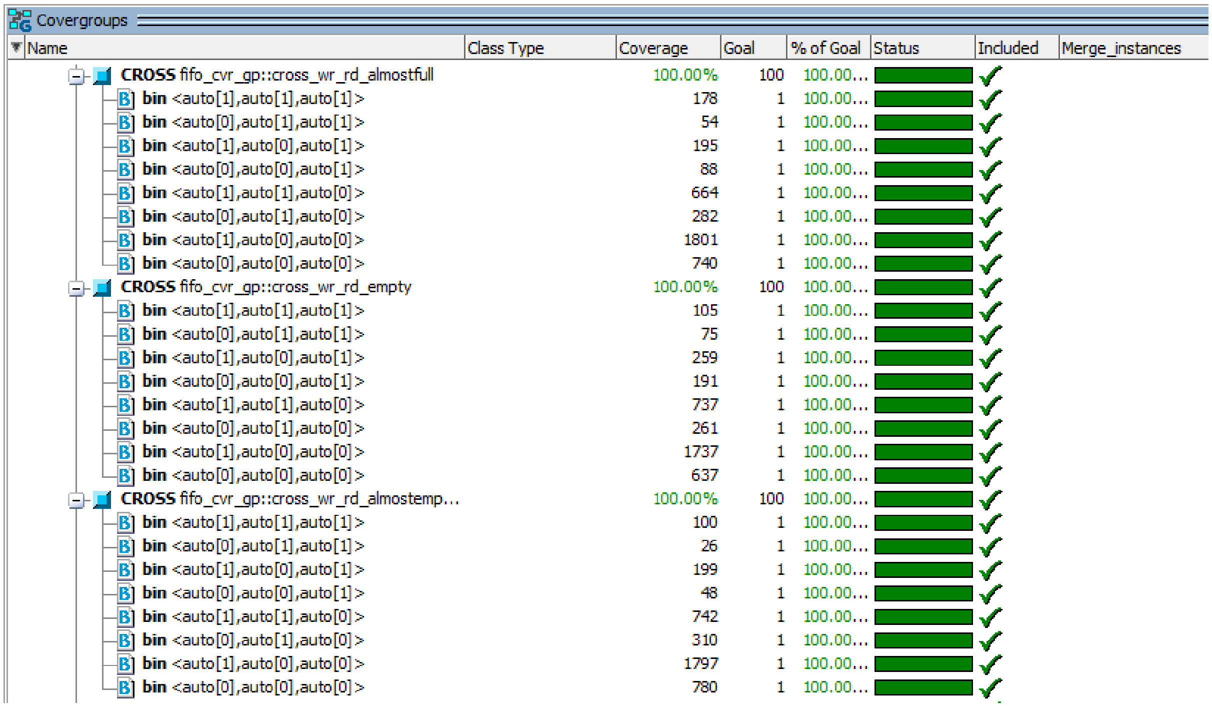
```
Directive Coverage:
    Directives                      15         15          0   100.00%

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                    Design Design   Lang File(Line)      Hits Status
                                        Unit   UnitType
--------------------------------------------------------------------------------------------
/\top#DUT /RST_check/rst_cover          FIFO   Verilog  SVA  FIFO.sv(92)      404 Covered
/\top#DUT /Full_check/full_cover        FIFO   Verilog  SVA  FIFO.sv(98)      181 Covered
/\top#DUT /Almostfull_check/almostfull_cover
                                        FIFO   Verilog  SVA  FIFO.sv(104)     259 Covered
/\top#DUT /Empty_check/empty_cover      FIFO   Verilog  SVA  FIFO.sv(110)     424 Covered
/\top#DUT /Almostempty_check/almostempty_cover
                                        FIFO   Verilog  SVA  FIFO.sv(116)     228 Covered
/\top#DUT /wr_ack_cover                 FIFO   Verilog  SVA  FIFO.sv(185)    1816 Covered
/\top#DUT /overflow_cover               FIFO   Verilog  SVA  FIFO.sv(188)     597 Covered
/\top#DUT /underflow_cover              FIFO   Verilog  SVA  FIFO.sv(191)     114 Covered
/\top#DUT /wr_ptr_wraparound_cover      FIFO   Verilog  SVA  FIFO.sv(194)    1816 Covered
/\top#DUT /rd_ptr_wraparound_cover      FIFO   Verilog  SVA  FIFO.sv(197)     890 Covered
/\top#DUT /counter_wraparound_incr_cover FIFO  Verilog  SVA  FIFO.sv(200)    1246 Covered
/\top#DUT /counter_wraparound_decr_cover FIFO  Verilog  SVA  FIFO.sv(203)     320 Covered
/\top#DUT /wr_ptr_thershold_cover       FIFO   Verilog  SVA  FIFO.sv(206)    3584 Covered
/\top#DUT /rd_ptr_thershold_cover       FIFO   Verilog  SVA  FIFO.sv(209)    3584 Covered
/\top#DUT /count_thershold_cover        FIFO   Verilog  SVA  FIFO.sv(212)    3584 Covered


DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                    Design Design   Lang File(Line)      Hits Status
                                        Unit   UnitType
--------------------------------------------------------------------------------------------
/\top#DUT /RST_check/rst_cover          FIFO   Verilog  SVA  FIFO.sv(92)      404 Covered
/\top#DUT /Full_check/full_cover        FIFO   Verilog  SVA  FIFO.sv(98)      181 Covered
/\top#DUT /Almostfull_check/almostfull_cover
                                        FIFO   Verilog  SVA  FIFO.sv(104)     259 Covered
/\top#DUT /Empty_check/empty_cover      FIFO   Verilog  SVA  FIFO.sv(110)     424 Covered
/\top#DUT /Almostempty_check/almostempty_cover
                                        FIFO   Verilog  SVA  FIFO.sv(116)     228 Covered
/\top#DUT /wr_ack_cover                 FIFO   Verilog  SVA  FIFO.sv(185)    1816 Covered
/\top#DUT /overflow_cover               FIFO   Verilog  SVA  FIFO.sv(188)     597 Covered
/\top#DUT /underflow_cover              FIFO   Verilog  SVA  FIFO.sv(191)     114 Covered
/\top#DUT /wr_ptr_wraparound_cover      FIFO   Verilog  SVA  FIFO.sv(194)    1816 Covered
/\top#DUT /rd_ptr_wraparound_cover      FIFO   Verilog  SVA  FIFO.sv(197)     890 Covered
/\top#DUT /counter_wraparound_incr_cover FIFO  Verilog  SVA  FIFO.sv(200)    1246 Covered
/\top#DUT /counter_wraparound_decr_cover FIFO  Verilog  SVA  FIFO.sv(203)     320 Covered
/\top#DUT /wr_ptr_thershold_cover       FIFO   Verilog  SVA  FIFO.sv(206)    3584 Covered
/\top#DUT /rd_ptr_thershold_cover       FIFO   Verilog  SVA  FIFO.sv(209)    3584 Covered
/\top#DUT /count_thershold_cover        FIFO   Verilog  SVA  FIFO.sv(212)    3584 Covered

TOTAL DIRECTIVE COVERAGE: 100.00%  COVERS: 15
```

14) Functional Coverage :



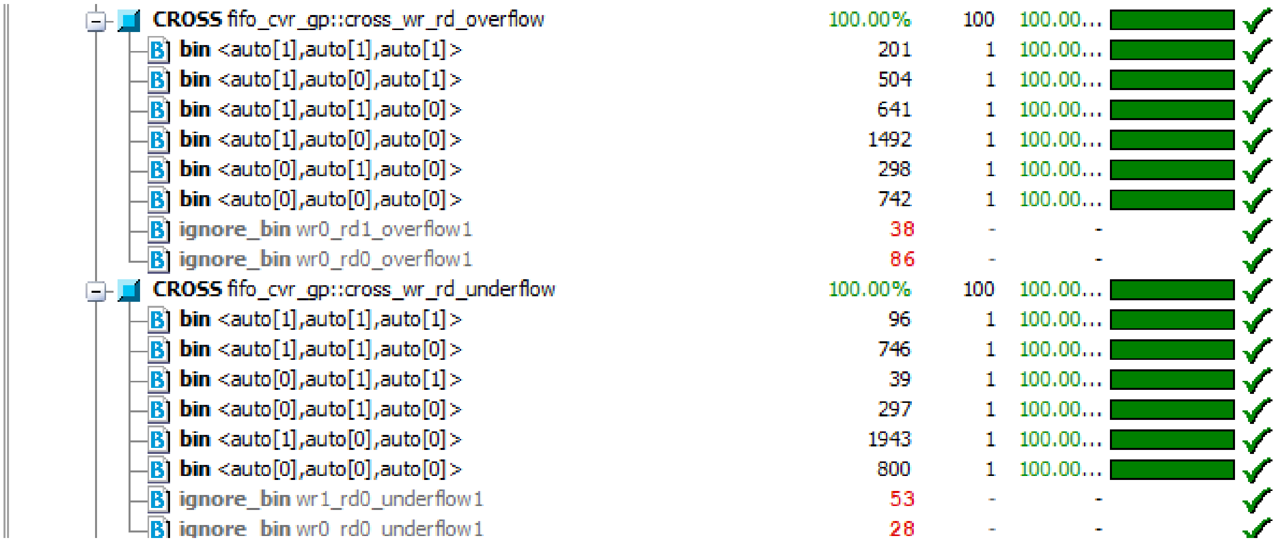
Covergroups

Name	Class Type	Coverage	Goal	% of Goal	Status	Included	Merge_instances
/FIFO_coverage_pkg/FIFO_coverage		100.00%					
TYPE fifo_cvr_gp		100.00%	100	100.00...	🟩	✓	auto(1)
CVP fifo_cvr_gp::wr_en_cp		100.00%	100	100.00...	🟩	✓	
bin auto[0]		1164	1	100.00...	🟩	✓	
bin auto[1]		2838	1	100.00...	🟩	✓	
CVP fifo_cvr_gp::rd_en_cp		100.00%	100	100.00...	🟩	✓	
bin auto[0]		2824	1	100.00...	🟩	✓	
bin auto[1]		1178	1	100.00...	🟩	✓	
CVP fifo_cvr_gp::full_cp		100.00%	100	100.00...	🟩	✓	
bin auto[0]		3102	1	100.00...	🟩	✓	
bin auto[1]		900	1	100.00...	🟩	✓	
CVP fifo_cvr_gp::almostfull_cp		100.00%	100	100.00...	🟩	✓	
bin auto[0]		3487	1	100.00...	🟩	✓	
bin auto[1]		515	1	100.00...	🟩	✓	
CVP fifo_cvr_gp::empty_cp		100.00%	100	100.00...	🟩	✓	
bin auto[0]		3372	1	100.00...	🟩	✓	
bin auto[1]		630	1	100.00...	🟩	✓	
CVP fifo_cvr_gp::almostempty_cp		100.00%	100	100.00...	🟩	✓	
bin auto[0]		3629	1	100.00...	🟩	✓	
bin auto[1]		373	1	100.00...	🟩	✓	
CVP fifo_cvr_gp::overflow_cp		100.00%	100	100.00...	🟩	✓	
bin auto[0]		3173	1	100.00...	🟩	✓	
bin auto[1]		829	1	100.00...	🟩	✓	
CVP fifo_cvr_gp::underflow_cp		100.00%	100	100.00...	🟩	✓	
bin auto[0]		3786	1	100.00...	🟩	✓	
bin auto[1]		216	1	100.00...	🟩	✓	
CVP fifo_cvr_gp::wr_ack_cp		100.00%	100	100.00...	🟩	✓	
bin auto[0]		2075	1	100.00...	🟩	✓	
bin auto[1]		1927	1	100.00...	🟩	✓	
CROSS fifo_cvr_gp::cross_wr_rd_full		100.00%	100	100.00...	🟩	✓	
bin <auto[1],auto[1],auto[1]>		102	1	100.00...	🟩	✓	
bin <auto[0],auto[1],auto[1]>		40	1	100.00...	🟩	✓	
bin <auto[1],auto[0],auto[1]>		562	1	100.00...	🟩	✓	
bin <auto[0],auto[0],auto[1]>		196	1	100.00...	🟩	✓	
bin <auto[1],auto[1],auto[0]>		740	1	100.00...	🟩	✓	
bin <auto[0],auto[1],auto[0]>		296	1	100.00...	🟩	✓	
bin <auto[1],auto[0],auto[0]>		1434	1	100.00...	🟩	✓	
bin <auto[0],auto[0],auto[0]>		632	1	100.00...	🟩	✓	

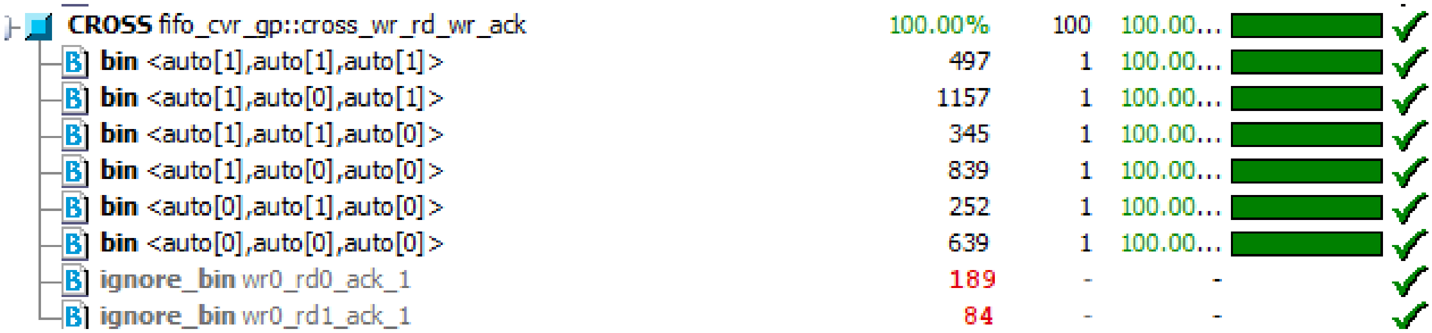


Covergroups

Name	Class Type	Coverage	Goal	% of Goal	Status	Included	Merge_instances
CROSS fifo_cvr_gp::cross_wr_rd_almostfull		100.00%	100	100.00...	🟩	✓	
bin <auto[1],auto[1],auto[1]>		178	1	100.00...	🟩	✓	
bin <auto[0],auto[1],auto[1]>		54	1	100.00...	🟩	✓	
bin <auto[1],auto[0],auto[1]>		195	1	100.00...	🟩	✓	
bin <auto[0],auto[0],auto[1]>		88	1	100.00...	🟩	✓	
bin <auto[1],auto[1],auto[0]>		664	1	100.00...	🟩	✓	
bin <auto[0],auto[1],auto[0]>		282	1	100.00...	🟩	✓	
bin <auto[1],auto[0],auto[0]>		1801	1	100.00...	🟩	✓	
bin <auto[0],auto[0],auto[0]>		740	1	100.00...	🟩	✓	
CROSS fifo_cvr_gp::cross_wr_rd_empty		100.00%	100	100.00...	🟩	✓	
bin <auto[1],auto[1],auto[1]>		105	1	100.00...	🟩	✓	
bin <auto[0],auto[1],auto[1]>		75	1	100.00...	🟩	✓	
bin <auto[1],auto[0],auto[1]>		259	1	100.00...	🟩	✓	
bin <auto[0],auto[0],auto[1]>		191	1	100.00...	🟩	✓	
bin <auto[1],auto[1],auto[0]>		737	1	100.00...	🟩	✓	
bin <auto[0],auto[1],auto[0]>		261	1	100.00...	🟩	✓	
bin <auto[1],auto[0],auto[0]>		1737	1	100.00...	🟩	✓	
bin <auto[0],auto[0],auto[0]>		637	1	100.00...	🟩	✓	
CROSS fifo_cvr_gp::cross_wr_rd_almostemp...		100.00%	100	100.00...	🟩	✓	
bin <auto[1],auto[1],auto[1]>		100	1	100.00...	🟩	✓	
bin <auto[0],auto[1],auto[1]>		26	1	100.00...	🟩	✓	
bin <auto[1],auto[0],auto[1]>		199	1	100.00...	🟩	✓	
bin <auto[0],auto[0],auto[1]>		48	1	100.00...	🟩	✓	
bin <auto[1],auto[1],auto[0]>		742	1	100.00...	🟩	✓	
bin <auto[0],auto[1],auto[0]>		310	1	100.00...	🟩	✓	
bin <auto[1],auto[0],auto[0]>		1797	1	100.00...	🟩	✓	
bin <auto[0],auto[0],auto[0]>		780	1	100.00...	🟩	✓	



CROSS fifo_cvr_gp::cross_wr_rd_overflow	100.00%	100	100.00...	🟩	✓
bin <auto[1],auto[1],auto[1]>	201	1	100.00...	🟩	✓
bin <auto[1],auto[0],auto[1]>	504	1	100.00...	🟩	✓
bin <auto[1],auto[1],auto[0]>	641	1	100.00...	🟩	✓
bin <auto[1],auto[0],auto[0]>	1492	1	100.00...	🟩	✓
bin <auto[0],auto[1],auto[0]>	298	1	100.00...	🟩	✓
bin <auto[0],auto[0],auto[0]>	742	1	100.00...	🟩	✓
ignore_bin wr0_rd1_overflow1	38	-	-		✓
ignore_bin wr0_rd0_overflow1	86	-	-		✓
CROSS fifo_cvr_gp::cross_wr_rd_underflow	100.00%	100	100.00...	🟩	✓
bin <auto[1],auto[1],auto[1]>	96	1	100.00...	🟩	✓
bin <auto[1],auto[1],auto[0]>	746	1	100.00...	🟩	✓
bin <auto[0],auto[1],auto[1]>	39	1	100.00...	🟩	✓
bin <auto[0],auto[1],auto[0]>	297	1	100.00...	🟩	✓
bin <auto[1],auto[0],auto[0]>	1943	1	100.00...	🟩	✓
bin <auto[0],auto[0],auto[0]>	800	1	100.00...	🟩	✓
ignore_bin wr1_rd0_underflow1	53	-	-		✓
ignore_bin wr0_rd0_underflow1	28	-	-		✓



CROSS fifo_cvr_gp::cross_wr_rd_wr_ack	100.00%	100	100.00...	🟩	✓
bin <auto[1],auto[1],auto[1]>	497	1	100.00...	🟩	✓
bin <auto[1],auto[0],auto[1]>	1157	1	100.00...	🟩	✓
bin <auto[1],auto[1],auto[0]>	345	1	100.00...	🟩	✓
bin <auto[1],auto[0],auto[0]>	839	1	100.00...	🟩	✓
bin <auto[0],auto[1],auto[0]>	252	1	100.00...	🟩	✓
bin <auto[0],auto[0],auto[0]>	639	1	100.00...	🟩	✓
ignore_bin wr0_rd0_ack_1	189	-	-		✓
ignore_bin wr0_rd1_ack_1	84	-	-		✓