FIFO Design

Design bugs

- Counter bugs:

```
52  always @(posedge clk or negedge rst_n) begin
53      if (!rst_n) begin
54          count <= 0;
55      end
56      else begin
57          if  ( ({wr_en, rd_en} == 2'b10) && !full)
58              count <= count + 1;
59          else if ( ({wr_en, rd_en} == 2'b01) && !empty)
60              count <= count - 1;
61      end
62  end
```

this code does not handle when wr_en and rd_en are high and the FIFO Full or Empty.

Handling:

```
79  always @(posedge clk or negedge rst_n) begin
80      if (!rst_n) begin
81          count <= 0;
82      end
83      else begin
84          if  ( ({wr_en, rd_en} == 2'b10) && !full)
85              count <= count + 1;
86          else if ( ({wr_en, rd_en} == 2'b01) && !empty)
87              count <= count - 1;
88          else if (({wr_en, rd_en} == 2'b11) && empty)
89              count <= count + 1;
90          else if (({wr_en, rd_en} == 2'b11) && full)
91              count <= count - 1;
92      end
93  end
```

- Output signals bugs:

```
24  always @(posedge clk or negedge rst_n) begin
25      if (!rst_n) begin
26          wr_ptr <= 0;
27      end
28      else if (wr_en && count < FIFO_DEPTH) begin
29          mem[wr_ptr] <= data_in;
30          wr_ack <= 1;
31          wr_ptr <= wr_ptr + 1;
32      end
33      else begin
34          wr_ack <= 0;
35          if (full & wr_en)
36              overflow <= 1;
37          else
38              overflow <= 0;
39      end
40  end

42  always @(posedge clk or negedge rst_n) begin
43      if (!rst_n) begin
44          rd_ptr <= 0;
45      end
46      else if (rd_en && count != 0) begin
47          data_out <= mem[rd_ptr];
48          rd_ptr <= rd_ptr + 1;
49      end
50  end

64  assign full = (count == FIFO_DEPTH)? 1 : 0;
65  assign empty = (count == 0)? 1 : 0;
66  assign underflow = (empty && rd_en)? 1 : 0;
67  assign almostfull = (count == FIFO_DEPTH-2)? 1 : 0;
68  assign almostempty = (count == 1)? 1 : 0;
```

Almost full signal will be wrong because must high when count = FIFO_DEPTH-1 not -2, and underflow signal must get its value in always block of Reading not with assign statement and when reset asserted overflow and underflow must be zeros.

Handling:

```
38  always @(posedge clk or negedge rst_n) begin
39      if (!rst_n) begin
40          wr_ptr <= 0;
41          overflow <=0;
42      end
43      else if (wr_en ) begin
44          if (full==0)begin
45              mem[wr_ptr] <= data_in;
46              wr_ack <= 1;
47              wr_ptr <= wr_ptr + 1;
48              overflow <=0;
49          end
50          else begin
51              wr_ack  <= 0;
52              overflow<= 1;
53          end
54      end
55      else begin
56          overflow  <= 0;
57          wr_ack    <= 0;
58      end
59  end
61  always @(posedge clk or negedge rst_n) begin
62      if (!rst_n) begin
63          rd_ptr <= 0;
64          underflow <=0;
65      end
66      else if (rd_en ) begin
67          if (empty==0)begin
68              data_out <= mem[rd_ptr];
69                rd_ptr <= rd_ptr + 1;
70             underflow <=0;
71          end
72          else
73              underflow <=1;
74      end
75      else
76          underflow <=0;
77  end
95  always@(count)begin
96      if (count==FIFO_DEPTH )
97          full=1;
98      else
99          full=0;
100     if (count==0)
101         empty=1;
102     else
103         empty=0;
104     if (count==(FIFO_DEPTH-1))
105         almostfull=1;
106     else
107         almostfull=0;
108     if (count == 1)
109         almostempty=1;
110     else
111         almostempty=0;
112 end
```

Design code

```
////////////////////////////////////////////////////////////////////////////////
// Author: Kareem Waseem
// Course: Digital Verification using SV & UVM
//
// Description: FIFO Design
//
////////////////////////////////////////////////////////////////////////////////
module FIFO #(parameter FIFO_DEPTH = 8, FIFO_WIDTH = 16)(interface_FIFO.DUT_Design inst_interface);


logic [inst_interface.FIFO_WIDTH-1:0] data_in;
```

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


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

localparam max_fifo_addr = $clog2(FIFO_DEPTH);

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

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

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

```
        overflow  <= 0;
        wr_ack    <= 0;
    end
end

always @(posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        rd_ptr <= 0;
        underflow <=0;
    end
    else if (rd_en ) begin
        if (empty==0)begin
            data_out <= mem[rd_ptr];
              rd_ptr <= rd_ptr + 1;
           underflow <=0;
        end
        else
            underflow <=1;
    end
    else
        underflow <=0;
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;
        else if (({wr_en, rd_en} == 2'b11) && empty)
            count <= count + 1;
        else if (({wr_en, rd_en} == 2'b11) && full)
            count <= count - 1;
    end
end

always@(count)begin
    if (count==FIFO_DEPTH )
        full=1;
    else
        full=0;
    if (count==0)
```

```
        empty=1;
    else
        empty=0;
    if (count==(FIFO_DEPTH-1))
        almostfull=1;
    else
        almostfull=0;
    if (count == 1)
        almostempty=1;
    else
        almostempty=0;
end
// Assertion guarded by the SIM macro
    `ifdef SIM
    // assertion here
    property EMPTY;
    @(posedge clk) disable iff (!rst_n)
     (count==0) |-> empty |-> (!full);
    endproperty

     property ALMOSTEMPTY;
    @(posedge clk) disable iff (!rst_n)
     (count==1) |-> (!empty)|-> almostempty;
    endproperty

     property ALMOSTFULL ;
    @(posedge clk) disable iff (!rst_n)
     (count==7) |-> almostfull ;
    endproperty

    property FULL ;
    @(posedge clk) disable iff (!rst_n)
     (count==8) |-> (full) |-> (!almostfull);
    endproperty

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

    property UNDERFLOW ;
    @(posedge clk) disable iff (!rst_n)
     // (rd_en && empty) |=> $past(underflow) ;
     (rd_en && empty) |=> underflow ;
    endproperty
```

```
    property WR_ACK_HIGH ;
    @(posedge clk) disable iff (!rst_n)
     // (wr_en && (!full) ) |=> ($past(wr_ack));
     (wr_en && (!full) ) |=> wr_ack;
    endproperty

    property WR_ACK_LOW ;
    @(posedge clk) disable iff (!rst_n)
     // (wr_en && full ) |=> (!($past(wr_ack)));
     (wr_en && full ) |=> (!wr_ack);
    endproperty

    property COUNTER_0 ;
    @(posedge clk ) !rst_n |=> (count==0);
    endproperty

    property COUNTER_INC_10 ;
    @(posedge clk) disable iff (!rst_n)
    ( ({wr_en, rd_en} == 2'b10) && !full)  |=> (count ==( $past(count)+1))   ;
    endproperty

    property COUNTER_INC_01 ;
    @(posedge clk) disable iff (!rst_n)
    ( ({wr_en, rd_en} == 2'b01) && !empty)  |=> (count ==( $past(count)-1))   ;
    endproperty

    property COUNTER_INC_11_WR ;
    @(posedge clk) disable iff (!rst_n)
    (({wr_en, rd_en} == 2'b11) && empty)  |=> (count ==( $past(count)+1))   ;
    endproperty

    property COUNTER_INC_11_RD ;
    @(posedge clk) disable iff (!rst_n)
    (({wr_en, rd_en} == 2'b11) && full)  |=> (count ==( $past(count)-1))   ;
    endproperty


    property COUNTER_STUN ;
    @(posedge clk) disable iff (!rst_n)
    // (((rd_en && empty) && (!wr_en))||((wr_en && full) && (!rd_en)) ) |=>
($past($past(count))==($past(count)));
    (((rd_en && empty) && (!wr_en))||((wr_en && full) && (!rd_en)) ) |=> ($past(count) == (count));
    endproperty
```

```
        FULL_assertion             : assert property (FULL)          else $display("FULL_assertion
fail"          );
        EMPTY_assertion            : assert property (EMPTY)         else
$display("EMPTY_assertion fail"         );
        ALMOSTFULL_assertion       : assert property (ALMOSTFULL)    else
$display("ALMOSTFULL_assertion fail"    );
        ALMOSTEMPTY_assertion      : assert property (ALMOSTEMPTY)   else
$display("ALMOSTEMPTY_assertion fail"   );
        OVERFLOW_assertion         : assert property (OVERFLOW)      else
$display("OVERFLOW_assertion"           );
        UNDERFLOW_assertion        : assert property (UNDERFLOW)     else
$display("UNDERFLOW_assertion"          );
        WR_ACK_HIGH_assertion      : assert property (WR_ACK_HIGH)   else
$display("WR_ACK_HIGH_assertion"        );
        WR_ACK_LOW_assertion       : assert property (WR_ACK_LOW)    else
$display("WR_ACK_LOW_assertion"         );
        COUNTER_0_assertion        : assert property (COUNTER_0)     else
$display("COUNTER_0_assertion"          );
        COUNTER_INC_10_assertion   : assert property (COUNTER_INC_10) else
$display("COUNTER_INC_WR_assertion fail");
        COUNTER_INC_01_assertion   : assert property (COUNTER_INC_01)   else
$display("COUNTER_INC_WR_assertion fail");
        COUNTER_INC_11_WR_assertion: assert property (COUNTER_INC_11_WR) else
$display("COUNTER_INC_WR_assertion fail");
        COUNTER_INC_11_RD_assertion: assert property (COUNTER_INC_11_RD) else
$display("COUNTER_INC_WR_assertion fail");
        COUNTER_STUN_assertion     : assert property (COUNTER_STUN)     else
$display("COUNTER_STUN_assertion fail"  );

        FULL_cover          : cover property (FULL)           $display("FULL_assertion
Pass"          );
        EMPTY_cover         : cover property (EMPTY)          $display("EMPTY_assertion
Pass"        );
        ALMOSTFULL_cover    : cover property (ALMOSTFULL)     $display("ALMOSTFULL_assertion
Pass"    );
        ALMOSTEMPTY_cover   : cover property (ALMOSTEMPTY)    $display("ALMOSTEMPTY_assertion
Pass"   );
        OVERFLOW_cover      : cover property
(OVERFLOW)        $display("OVERFLOW_assertion"          );
        UNDERFLOW_cover     : cover property
(UNDERFLOW)       $display("UNDERFLOW_assertion"         );
        WR_ACK_HIGH_cover   : cover property
(WR_ACK_HIGH)     $display("WR_ACK_HIGH_assertion"       );
```

```
        WR_ACK_LOW_cover    : cover property
(WR_ACK_LOW)      $display("WR_ACK_LOW_assertion"         );
        COUNTER_0_cover     : cover property
(COUNTER_0)       $display("COUNTER_0_assertion"          );
        COUNTER_INC_10_cover: cover property (COUNTER_INC_10)  $display("COUNTER_INC_WR_assertion
Pass");
        COUNTER_INC_01_cover: cover property (COUNTER_INC_01)  $display("COUNTER_INC_WR_assertion
Pass");
        COUNTER_INC_11_WR_cover: cover property
(COUNTER_INC_11_WR)  $display("COUNTER_INC_WR_assertion Pass");
        COUNTER_INC_11_RD_cover: cover property
(COUNTER_INC_11_RD)  $display("COUNTER_INC_WR_assertion Pass");
        COUNTER_STUN_cover  : cover property (COUNTER_STUN)    $display("COUNTER_STUN_assertion
Pass"  );
    `endif

endmodule
```

Golden model code

```
module FIFO_GOLDEN (interface_FIFO.GOLDEN_REF inst_interface);
// module Golden (data_in , clk , rst_n , wr_en , rd_en , data_out_expect , wr_ack_expect , overflow_expect ,
full_expect ,
//  empty_expect , almostfull_expect , almostempty_expect , underflow_expect);
// parameter FIFO_WIDTH=16;
// parameter FIFO_DEPTH=8 ;
typedef logic [15:0] data_bus; // Define a 16-bit data bus type
// logic [inst_interface.FIFO_WIDTH-1:0] data_in;
 data_bus data_in;
 bit clk;
 logic  rst_n, wr_en, rd_en;
// logic [inst_interface.FIFO_WIDTH-1:0] data_out_expect;
 data_bus data_out_expect;
 logic wr_ack_expect, overflow_expect;
 logic full_expect, empty_expect, almostfull_expect, almostempty_expect, underflow_expect;
bit  deleted_queue;

assign inst_interface.wr_ack_expect     =wr_ack_expect;
assign inst_interface.overflow_expect   =overflow_expect;
assign inst_interface.underflow_expect  =underflow_expect;
assign inst_interface.full_expect       =full_expect;
assign inst_interface.empty_expect      =empty_expect;
assign inst_interface.almostfull_expect =almostfull_expect;
assign inst_interface.almostempty_expect=almostempty_expect;
assign inst_interface.data_out_expect   =data_out_expect;

assign clk= inst_interface.clk;
```

```
assign rst_n=inst_interface.rst_n;
assign wr_en=inst_interface.wr_en;
assign rd_en=inst_interface.rd_en;
assign data_in=inst_interface.data_in;

data_bus data_queue[$]; // Declare a queue of 16-bit data buses
always@(*)begin
    if (data_queue.size()==7)
        almostfull_expect=1;
    else
        almostfull_expect=0;
    if (data_queue.size()==1)
        almostempty_expect=1;
    else
        almostempty_expect=0;
    if (data_queue.size()==8)
        full_expect =1;
    else
        full_expect =0;
    if (data_queue.size()==0)
        empty_expect=1;
    else
        empty_expect=0;

    if (deleted_queue){almostempty_expect,almostfull_expect,
    full_expect,empty_expect,overflow_expect,underflow_expect}=6'b000_100;
end

always @(posedge clk or negedge rst_n) begin
    if(~rst_n) begin
        data_queue.delete();
        deleted_queue=1;
    end
    else if (wr_en) begin
         deleted_queue=0;
        if (full_expect==0)begin
            wr_ack_expect=1;
            overflow_expect=0;
            data_queue.push_back(data_in);
        end
        else begin
            overflow_expect=1;
            wr_ack_expect  =0;
        end
      end
```

```
    else begin
        wr_ack_expect  =0;
        overflow_expect=0;
    end

end
always @(posedge clk or negedge rst_n) begin
    if(~rst_n)
        data_queue.delete();
    else if (rd_en)begin
         deleted_queue=0;
        if (empty_expect==0)begin
            underflow_expect=0;
            data_out_expect=data_queue.pop_front();
        end
        else
            underflow_expect=1;
    end
    else
        underflow_expect=0;
    end

endmodule
```

Top code

```
module top();
    bit clk;
    always #10 clk=~clk;
    interface_FIFO inst_interface(clk);
    FIFO DUT_Design (inst_interface);
    // FIFO_Assertion_sva DUT_Assertion (inst_interface);
    FIFO_GOLDEN GOLDEN_REF (inst_interface);
    tb_FIFO TESTBENCH (inst_interface);
    monitor MONITOR(inst_interface);
    // bind FIFO  FIFO_Assertion_sva sva_inst (inst_interface);
endmodule
```

Interface code

```
interface interface_FIFO (clk);

input bit  clk;
parameter FIFO_WIDTH=16;
parameter FIFO_DEPTH=8 ;
logic [FIFO_WIDTH-1:0] data_in , data_out,data_out_expect ;
logic  wr_en;
logic  rd_en;
logic  rst_n;
logic  full,full_expect  ;
```

```
logic  empty,empty_expect;
logic  almostfull,almostfull_expect  ;
logic  almostempty,almostempty_expect;
logic  overflow,overflow_expect      ;
logic  underflow,underflow_expect    ;
logic  wr_ack,wr_ack_expect          ;

    modport DUT_Design (input clk,rst_n,data_in,wr_en,rd_en,
                output data_out,full,empty,almostempty,almostfull,overflow,underflow,wr_ack);
    // modport DUT_Assertion (input clk,rst_n,data_in,wr_en,rd_en,
    //           output data_out,full,empty,almostempty,almostfull,overflow,underflow,wr_ack);
    modport TESTBENCH(input data_out,full,empty,almostempty,almostfull,overflow,underflow,wr_ack,
clk,
                      data_out_expect,full_expect,empty_expect,almostempty_expect,almostfull_expect
                ,overflow_expect,underflow_expect,wr_ack_expect,
                output rst_n,data_in,wr_en,rd_en);
    modport MONITOR (input clk,rst_n,data_in,wr_en,rd_en,data_out,
                     full,empty,almostempty,almostfull,overflow,underflow,wr_ack,
                     data_out_expect,full_expect,empty_expect,almostempty_expect,almostfull_expect
                ,overflow_expect,underflow_expect,wr_ack_expect);
    modport GOLDEN_REF(input clk,rst_n,data_in,wr_en,rd_en,
                output
data_out_expect,full_expect,empty_expect,almostempty_expect,almostfull_expect
                ,overflow_expect,underflow_expect,wr_ack_expect);
endinterface
```

Packages code

Shared package

```
package shared_package;
    bit test_finished;
    integer Correct_counts = 0 ;
    integer Error_counts   = 0 ;
endpackage
```

Transaction package

```
package package_transaction ;
    parameter FIFO_WIDTH=16;
    parameter FIFO_DEPTH=8 ;
class FIFO_transaction;
        bit  clk;
        rand logic [FIFO_WIDTH-1:0] data_in ;
             logic [FIFO_WIDTH-1:0] data_out ,data_out_expect ;
        rand logic  wr_en;
        rand logic  rd_en;
        rand logic  rst_n;
        logic  full  , full_expect ;
        logic  empty , empty_expect;
        logic  almostfull , almostfull_expect ;
        logic  almostempty, almostempty_expect;
```

```
        logic  overflow   , overflow_expect   ;
        logic  underflow  , underflow_expect  ;
        logic  wr_ack     , wr_ack_expect     ;

        integer WR_EN_ON_DIST=70;
        integer RD_EN_ON_DIST=30;
constraint General {
    rst_n dist {1:/70,0:/30};
    wr_en dist {1:/WR_EN_ON_DIST,0:/(100 - WR_EN_ON_DIST)};
    rd_en dist {1:/RD_EN_ON_DIST,0:/(100 - RD_EN_ON_DIST)};
        }
endclass
endpackage
```

Scoreboard package

```
package package_scoreboard;

import shared_package::*;
import package_transaction::*;
class FIFO_scoreboard;

bit clk_ref ;
logic [FIFO_WIDTH-1:0]  data_out_ref ;
logic  wr_en_ref;
logic  rd_en_ref;
logic  rst_n_ref;
logic  full_ref ;
logic  empty_ref;
logic  almostfull_ref ;
logic  almostempty_ref;
logic  overflow_ref   ;
logic  underflow_ref  ;
logic  wr_ack_ref     ;


function void reference_model(FIFO_transaction values_object);
// calculate the expected value
  clk_ref        = values_object.clk               ;

  full_ref       = values_object.full_expect       ;
  empty_ref      = values_object.empty_expect      ;
  almostfull_ref = values_object.almostfull_expect ;
  almostempty_ref= values_object.almostempty_expect;
  overflow_ref   = values_object.overflow_expect   ;
  underflow_ref  = values_object.underflow_expect  ;
  wr_ack_ref     = values_object.wr_ack_expect     ;
```

```
  data_out_ref   = values_object.data_out_expect   ;

endfunction
function void check_data(FIFO_transaction values_object);
    reference_model(values_object);

    if(data_out_ref===values_object.data_out)
        Correct_counts++;
   else begin
    $display("There is an error in %0t ns , data_out_ref !=
data_out ,data_out_ref=%0d ,values_object.data_out =%0d
",$time(),data_out_ref,values_object.full);
    Error_counts++;
   end

   if (full_ref === values_object.full)
    Correct_counts++;
   else begin
    $display("There is an error in %0t ns , full_ref !=
full ,full_ref=%0d ,values_object.full =%0d ",$time(),full_ref,values_object.full);
    Error_counts++;
   end

   if (almostfull_ref === values_object.almostfull)
    Correct_counts++;
   else begin
    $display("There is an error in %0t ns , almostfull_ref !=
almostfull ,almostfull_ref=%0d ,almostfull =%0d
",$time(),almostfull_ref,values_object.almostfull);
    Error_counts++;
   end

   if (empty_ref === values_object.empty)
    Correct_counts++;
   else begin
    $display("There is an error in %0t ns , empty_ref !=
empty ,empty_ref=%0d ,empty =%0d ",$time(),empty_ref,values_object.empty);
    Error_counts++;
   end

   if (almostempty_ref === values_object.almostempty)
    Correct_counts++;
   else begin
```

		$display("There is an error in %0t ns , almostempty_ref != almostempty ,almostempty_ref=%0d ,values_object.almostempty =%0d ",$time(),almostempty_ref,values_object.almostempty);
		Error_counts++;
		end
		
		if (overflow_ref === values_object.overflow)
		Correct_counts++;
		else begin
		$display("There is an error in %0t ns , overflow_ref != overflow ,overflow_ref=%0d ,values_object.overflow =%0d ",$time(),overflow_ref,values_object.overflow);
		Error_counts++;
		end
		
		if (underflow_ref === values_object.underflow)
		Correct_counts++;
		else begin
		$display("There is an error in %0t ns , underflow_ref != underflow ,underflow_ref=%0d ,values_object.underflow =%0d ",$time(),underflow_ref,values_object.underflow);
		Error_counts++;
		end
		
		if (wr_ack_ref === values_object.wr_ack)
		Correct_counts++;
		else begin
		$display("There is an error in %0t ns , wr_ack_ref != wr_ack ,wr_ack_ref=%0d ,values_object.wr_ack =%0d ",$time(),wr_ack_ref,values_object.wr_ack);
		Error_counts++;
		end
		
		endfunction
		endclass
		endpackage

	Coverage package	package package_coverage; import package_transaction::*; FIFO_transaction F_cvg_txn =new(); class FIFO_coverage ;

```
function void sample_data (FIFO_transaction F_txn);

            F_cvg_txn.clk         = F_txn.clk         ;
            F_cvg_txn.data_in     = F_txn.data_in     ;
            F_cvg_txn.data_out    = F_txn.data_out    ;
            F_cvg_txn.wr_en       = F_txn.wr_en       ;
            F_cvg_txn.rd_en       = F_txn.rd_en       ;
            F_cvg_txn.full        = F_txn.full        ;
            F_cvg_txn.empty       = F_txn.empty       ;
            F_cvg_txn.almostfull  = F_txn.almostfull  ;
            F_cvg_txn.almostempty = F_txn.almostempty ;
            F_cvg_txn.overflow    = F_txn.overflow    ;
            F_cvg_txn.underflow   = F_txn.underflow   ;
            F_cvg_txn.wr_ack      = F_txn.wr_ack      ;

            cover_group.sample ();
endfunction
//  which are write enable, read enable and each output control signals (outputs
except data_out) to make sure that all combinations of write and read enable took
place in all state of the FIFO.
covergroup cover_group;

    write_enable    :coverpoint F_cvg_txn.wr_en{
    bins write_1  = {1};
    bins write_0  = {0};
    }
    read_enable     :coverpoint F_cvg_txn.rd_en{
    bins read_1  = {1};
    bins read_0  = {0};
    }
    full_flag       :coverpoint F_cvg_txn.full {
    bins full_1  = {1};
    bins full_0  = {0};
    }
    empty_flag      :coverpoint F_cvg_txn.empty{
    bins empty_1  = {1};
    bins empty_0  = {0};
    }
    almostfull_flag :coverpoint F_cvg_txn.almostfull{
    bins almostfull_1  = {1};
    bins almostfull_0  = {0};
    }
    almostempty_flag:coverpoint F_cvg_txn.almostempty {
    bins almostempty_1  = {1};
    bins almostempty_0  = {0};
```

```
    }
    overflow_flag   :coverpoint F_cvg_txn.overflow    {
    bins overflow_1  = {1};
    bins overflow_0  = {0};
    }
    underflow_flag  :coverpoint F_cvg_txn.underflow   {
    bins underflow_1  = {1};
    bins underflow_0  = {0};
    }
    wr_ack_flag     :coverpoint F_cvg_txn.wr_ack  {
    bins wr_ack_1  = {1};
    bins wr_ack_0  = {0};
    }

    cross write_enable,read_enable,full_flag {
    ignore_bins ignored_1_will_not_happend =binsof (read_enable.read_1) && binsof
(full_flag.full_1);
    }
    cross write_enable,read_enable,empty_flag      ;
    cross write_enable,read_enable,almostfull_flag ;
    cross write_enable,read_enable,almostempty_flag;
    cross write_enable,read_enable,overflow_flag{
    ignore_bins ignored_2_will_not_happend =binsof (write_enable.write_0) && binsof
(overflow_flag.overflow_1);
    }
    cross write_enable,read_enable,underflow_flag{
    ignore_bins ignored_3_will_not_happend =binsof (read_enable.read_0) && binsof
(underflow_flag.underflow_1);
    }
    cross write_enable,read_enable,wr_ack_flag     ;
    endgroup
    cover_group=new();
        endclass
endpackage
```

Monitor code

```
//import the shared package
import shared_package::*;
//import class packages
import package_transaction::*;
import package_coverage   ::*;
import package_scoreboard ::*;

module monitor (interface_FIFO.MONITOR inst_interface);
    //creat objects of different classes
FIFO_transaction class_transaction =new();
```

```
FIFO_scoreboard  class_scoreboard  =new();
FIFO_coverage    class_coverage    =new();

     initial begin
forever begin
    @(negedge inst_interface.clk)begin

        class_transaction.clk         =inst_interface.clk    ;
        class_transaction.rst_n       =inst_interface.rst_n  ;
        class_transaction.wr_en       =inst_interface.wr_en  ;
        class_transaction.rd_en       =inst_interface.rd_en  ;
        class_transaction.data_in     =inst_interface.data_in;

        class_transaction.wr_ack      =inst_interface.wr_ack   ;
        class_transaction.overflow    =inst_interface.overflow ;
        class_transaction.full        =inst_interface.full     ;
        class_transaction.almostfull  =inst_interface.almostfull ;
        class_transaction.almostempty =inst_interface.almostempty;
        class_transaction.data_out    =inst_interface.data_out   ;
        class_transaction.underflow   =inst_interface.underflow  ;
        class_transaction.empty       =inst_interface.empty  ;

        class_transaction.data_out_expect   =inst_interface.data_out_expect;
        class_transaction.full_expect       =inst_interface.full_expect;
        class_transaction.empty_expect      =inst_interface.empty_expect;
        class_transaction.almostfull_expect =inst_interface.almostfull_expect;
        class_transaction.almostempty_expect=inst_interface.almostempty_expect;
        class_transaction.overflow_expect   =inst_interface.overflow_expect;
        class_transaction.underflow_expect  =inst_interface.underflow_expect;
        class_transaction.wr_ack_expect     =inst_interface.wr_ack_expect;

               fork
         // process 1
         begin
            class_coverage.sample_data(class_transaction);
         end

         // process 2

         begin
            class_scoreboard.check_data(class_transaction);
         end

        join
```

```
          end

    if (test_finished==1)begin
     $display("Total Correct counts = %0d and total Error counts
= %0d",Correct_counts/8,Error_counts);
     $stop;
    end

end
    // $monitor("wr_ack = %b, overflow = %b, almostfull = %b, full = %b, almostempty
= %b,
    //  data_out = %b ,clk = %b, rst_n = %b, wr_en = %b,rd_en = %b, data_in = %b",
wr_ack,overflow,
    //  almostfull,full,almostempty,data_out,rst_n,wr_en,rd_en,data_in );
  end
endmodule
```

Testbench code

```
import package_coverage::*;
import package_scoreboard::*;
import package_transaction::*;
import shared_package::*;

module tb_FIFO (interface_FIFO.TESTBENCH inst_interface );

FIFO_transaction class_random_ports =new();
bit  clk;
logic [FIFO_WIDTH-1:0] data_in , data_out,data_out_expect ;
logic  wr_en;
logic  rd_en;
logic  rst_n;
logic  full ,full_expect ;
logic  empty,empty_expect;
logic  almostfull,almostfull_expect ;
logic  almostempty,almostempty_expect;
logic  overflow ,overflow_expect    ;
logic  underflow, underflow_expect  ;
logic  wr_ack   ,wr_ack_expect      ;

assign wr_ack_expect      = inst_interface.wr_ack_expect     ;
assign empty_expect       = inst_interface.empty_expect      ;
assign overflow_expect    = inst_interface.overflow_expect   ;
assign full_expect        = inst_interface.full_expect       ;
assign almostfull_expect  = inst_interface.almostfull_expect ;
assign almostempty_expect = inst_interface.almostempty_expect;
```

```
assign data_out_expect    = inst_interface.data_out_expect   ;
assign underflow_expect   = inst_interface.underflow_expect  ;

assign wr_ack      = inst_interface.wr_ack     ;
assign empty       = inst_interface.empty      ;
assign overflow    = inst_interface.overflow   ;
assign full        = inst_interface.full       ;
assign almostfull  = inst_interface.almostfull ;
assign almostempty = inst_interface.almostempty;
assign data_out    = inst_interface.data_out   ;
assign underflow   = inst_interface.underflow  ;


assign clk         = inst_interface.clk        ;
assign inst_interface.rst_n      = rst_n  ;
assign inst_interface.wr_en      = wr_en  ;
assign inst_interface.rd_en      = rd_en  ;
assign inst_interface.data_in    = data_in;

task inst_values();
    rst_n  =class_random_ports.rst_n  ;
    @(negedge clk)begin
    data_in=class_random_ports.data_in;
    wr_en  =class_random_ports.wr_en  ;
    rd_en  =class_random_ports.rd_en  ;
    end
endtask


initial begin
    test_finished=0;
    for (int i=0;i<5000;i++)begin
    assert(class_random_ports.randomize());
        if (i==0)
         class_random_ports.rst_n=0;
        inst_values();
    end

    test_finished=1;
end
endmodule
```

Do file

```
# First Section (Assertions and Simulation)
vlib work
vlog shared_package.sv package_coverage.sv top.sv package_fifo_scoreboard.sv package_transaction.sv
FIFO_design_sv.sv Golden_Ref.sv interface.sv Monitor.sv test_check.sv +cover
vlog -work work -vopt -sv -stats=none +incdir+path_to_assertions_dir +define+SIM FIFO_design_sv.sv
vsim -voptargs=+acc work.top -cover
run -all
coverage save top.ucdb -du FIFO -onexit
coverage report -detail -assert -cvg -directive -comments -output Assertion_Fcoverage_reports.txt {}
quit -sim

# Second Section (Code Coverage)
vlog shared_package.sv package_coverage.sv top.sv package_fifo_scoreboard.sv package_transaction.sv
FIFO_design_sv.sv Golden_Ref.sv interface.sv Monitor.sv test_check.sv +cover
vsim -voptargs=+acc work.top -cover
run -all
coverage save top.ucdb -du FIFO -onexit
quit -sim
vcover report top.ucdb -details -annotate -all -output Code_coverage_reports.txt
```

Code Coverage

```
Coverage Report by instance with details

================================================================================
=== Instance: /\top#DUT_Design
=== Design Unit: work.FIFO
================================================================================
Branch Coverage:
  Enabled Coverage          Bins     Hits    Misses  Coverage
  ----------------          ----     ----    ------  --------
  Branches                    25       25         0  100.00%

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

Branch Coverage for instance /\top#DUT_Design

  Line       Item                  Count     Source
  ----       ----                  -----     ------
 File FIFO_design_sv.sv
------------------------------------IF Branch------------------------------------
  39                               6019     Count coming in to IF
  39         1                     2475              if (!rst_n) begin
  43         1                     2443              else if (wr_en ) begin
  55         1                     1101              else begin
Branch totals: 3 hits of 3 branches = 100.00%

------------------------------------IF Branch------------------------------------
  44                               2443     Count coming in to IF
  44         1                     2431                                  if (full==0)begin
  50         1                       12                                  else begin
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
  62                               5407     Count coming in to IF
  62         1                     2225              if (!rst_n) begin
  66         1                     1078              else if (rd_en ) begin
  75         1                     2104              else
Branch totals: 3 hits of 3 branches = 100.00%

------------------------------------IF Branch------------------------------------
  67                               1078     Count coming in to IF
  67         1                      658                                  if (empty==0)begin
  72         1                      420                                  else
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
  80                               5665     Count coming in to IF
  80         1                     2327              if (!rst_n) begin
  83         1                     3338              else begin
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
  84                               3338     Count coming in to IF
  84         1                     1684                                  if                ( ({wr_en, rd_en} == 2'b10) && !full)
  86         1                      204                                  else if ( ({wr_en, rd_en} == 2'b01) && !empty)
  88         1                      299                                  else if (({wr_en, rd_en} == 2'b11) && empty)
  90         1                        6                                  else if (({wr_en, rd_en} == 2'b11) && full)
                                   1145     All False Count
Branch totals: 5 hits of 5 branches = 100.00%

------------------------------------IF Branch------------------------------------
```

```
   96                        3074    Count coming in to IF
   96          1               14                if (count==FIFO_DEPTH )
   98          1             3060                else
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
   100                       3074    Count coming in to IF
   100         1              983                if (count==0)
   102         1             2091                else
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
   104                       3074    Count coming in to IF
   104         1               26                if (count==(FIFO_DEPTH-1))
   106         1             3048                else
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
   108                       3074    Count coming in to IF
   108         1             1035                if (count == 1)
   110         1             2039                else
Branch totals: 2 hits of 2 branches = 100.00%


Condition Coverage:
   Enabled Coverage          Bins   Covered    Misses  Coverage
   ----------------          ----   -------    --------
   Conditions                  16        16         0  100.00%

================================Condition Details================================

Condition Coverage for instance /\top#DUT_Design  --

  File FIFO_design_sv.sv
----------------Focused Condition View-------------------
Line      84 Item    1  ((~rd_en && wr_en) && ~full)
Condition totals: 3 of 3 input terms covered = 100.00%

  Input Term   Covered  Reason for no coverage   Hint
 -----------  --------  -----------------------  --------------
     rd_en        Y
     wr_en        Y
     full         Y

    Rows:      Hits  FEC Target           Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  rd_en_0              (~full && wr_en)
  Row   2:          1  rd_en_1              -
  Row   3:          1  wr_en_0              ~rd_en
  Row   4:          1  wr_en_1              (~full && ~rd_en)
  Row   5:          1  full_0               (~rd_en && wr_en)
  Row   6:          1  full_1               (~rd_en && wr_en)

----------------Focused Condition View-------------------
Line      86 Item    1  ((rd_en && ~wr_en) && ~empty)
Condition totals: 3 of 3 input terms covered = 100.00%

  Input Term   Covered  Reason for no coverage   Hint
 -----------  --------  -----------------------  --------------
     rd_en        Y
     wr_en        Y
     empty        Y

    Rows:      Hits  FEC Target           Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  rd_en_0              -
  Row   2:          1  rd_en_1              (~empty && ~wr_en)
  Row   3:          1  wr_en_0              (~empty && rd_en)
  Row   4:          1  wr_en_1              rd_en
  Row   5:          1  empty_0              (rd_en && ~wr_en)
  Row   6:          1  empty_1              (rd_en && ~wr_en)

----------------Focused Condition View-------------------
Line      88 Item    1  ((rd_en && wr_en) && empty)
Condition totals: 3 of 3 input terms covered = 100.00%

  Input Term   Covered  Reason for no coverage   Hint
 -----------  --------  -----------------------  --------------
     rd_en        Y
     wr_en        Y
     empty        Y

    Rows:      Hits  FEC Target           Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  rd_en_0              -
  Row   2:          1  rd_en_1              (empty && wr_en)
  Row   3:          1  wr_en_0              rd_en
  Row   4:          1  wr_en_1              (empty && rd_en)
```

```
  Row   5:        1  empty_0            (rd_en && wr_en)
  Row   6:        1  empty_1            (rd_en && wr_en)

----------------Focused Condition View-------------------
Line       90 Item    1  ((rd_en && wr_en) && full)
Condition totals: 3 of 3 input terms covered = 100.00%

  Input Term   Covered  Reason for no coverage   Hint
 -----------  --------  ----------------------  --------------
      rd_en        Y
      wr_en        Y
       full        Y

    Rows:      Hits  FEC Target          Non-masking condition(s)
 ---------  ---------  -------------------  -------------------------
  Row   1:        1  rd_en_0             -
  Row   2:        1  rd_en_1             (full && wr_en)
  Row   3:        1  wr_en_0             rd_en
  Row   4:        1  wr_en_1             (full && rd_en)
  Row   5:        1  full_0              (rd_en && wr_en)
  Row   6:        1  full_1              (rd_en && wr_en)

----------------Focused Condition View-------------------
Line       96 Item    1  (count == 8)
Condition totals: 1 of 1 input term covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
  -----------  --------  ----------------------  --------------
  (count == 8)        Y

    Rows:      Hits  FEC Target          Non-masking condition(s)
 ---------  ---------  -------------------  -------------------------
  Row   1:        1  (count == 8)_0       -
  Row   2:        1  (count == 8)_1       -

----------------Focused Condition View-------------------
Line       100 Item    1  (count == 0)
Condition totals: 1 of 1 input term covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
  -----------  --------  ----------------------  --------------
  (count == 0)        Y

    Rows:      Hits  FEC Target          Non-masking condition(s)
 ---------  ---------  -------------------  -------------------------
  Row   1:        1  (count == 0)_0       -
  Row   2:        1  (count == 0)_1       -

----------------Focused Condition View-------------------
Line       104 Item    1  (count == (8 - 1))
Condition totals: 1 of 1 input term covered = 100.00%

        Input Term   Covered  Reason for no coverage   Hint
       -----------  --------  ----------------------  --------------
  (count == (8 - 1))        Y

    Rows:      Hits  FEC Target          Non-masking condition(s)
 ---------  ---------  -------------------  -------------------------
  Row   1:        1  (count == (8 - 1))_0  -
  Row   2:        1  (count == (8 - 1))_1  -

----------------Focused Condition View-------------------
Line       108 Item    1  (count == 1)
Condition totals: 1 of 1 input term covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
  -----------  --------  ----------------------  --------------
  (count == 1)        Y

    Rows:      Hits  FEC Target          Non-masking condition(s)
 ---------  ---------  -------------------  -------------------------
  Row   1:        1  (count == 1)_0       -
  Row   2:        1  (count == 1)_1       -


Statement Coverage:
    Enabled Coverage          Bins     Hits    Misses  Coverage
    ----------------          ----     ----    ------  --------
    Statements                  39       39         0   100.00%

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

Statement Coverage for instance /\top#DUT_Design  --

    Line        Item                  Count     Source
    ----        ----                  -----     ------
  File FIFO_design_sv.sv
    8                                 module FIFO #(parameter FIFO_DEPTH = 8, FIFO_WIDTH = 16)(interface_FIFO.DUT_Design inst_interface);
```

```
9
10                          logic [inst_interface.FIFO_WIDTH-1:0] data_in;
11                          logic clk, rst_n, wr_en, rd_en;
12                          logic [inst_interface.FIFO_WIDTH-1:0] data_out;
13                          logic wr_ack, overflow;
14                          logic full, empty, almostfull, almostempty, underflow;
15
16
17                          assign inst_interface.wr_ack     = wr_ack;
18                          assign inst_interface.overflow   = overflow;
19                          assign inst_interface.underflow  = underflow;
20                          assign inst_interface.full       = full;
21                          assign inst_interface.empty      = empty;
22                          assign inst_interface.almostfull  = almostfull;
23                          assign inst_interface.almostempty = almostempty;
24                          assign inst_interface.data_out   = data_out;
25      1       10003    assign clk= inst_interface.clk;
26      1       2037     assign rst_n=inst_interface.rst_n;
27      1       2120     assign wr_en=inst_interface.wr_en;
28      1       2116     assign rd_en=inst_interface.rd_en;
29      1       5001     assign data_in=inst_interface.data_in;
30
31                          localparam max_fifo_addr = $clog2(FIFO_DEPTH);
32
33                          reg [FIFO_WIDTH-1:0] mem [FIFO_DEPTH-1:0];
34
35                          reg [max_fifo_addr:0] count ;
36                          reg [max_fifo_addr-1:0] wr_ptr,rd_ptr ;
37
38      1       6019     always @(posedge clk or negedge rst_n) begin
39                               if (!rst_n) begin
40      1       2475                     wr_ptr <= 0;
41      1       2475                     overflow <=0;
42                               end
43                               else if (wr_en ) begin
44                                        if (full==0)begin
45      1       2431                        mem[wr_ptr] <= data_in;
46      1       2431                        wr_ack <= 1;
47      1       2431                        wr_ptr <= wr_ptr + 1;
48      1       2431                        overflow <=0;
49                                       end
50                                       else begin
51      1        12                         wr_ack  <= 0;
52      1        12                                  overflow<= 1;
53                                       end
54                               end
55                               else begin
56      1       1101                     overflow  <= 0;
57      1       1101                     wr_ack   <= 0;
58                               end
59                          end
60
61      1       5407     always @(posedge clk or negedge rst_n) begin
62                               if (!rst_n) begin
63      1       2225                     rd_ptr <= 0;
64      1       2225                     underflow <=0;
65                               end
66                               else if (rd_en ) begin
67                                        if (empty==0)begin
68      1        658                                 data_out <= mem[rd_ptr];
69      1        658                            rd_ptr <= rd_ptr + 1;
70      1        658                          underflow <=0;
71                                       end
72                                       else
73      1        420                                 underflow <=1;
74                               end
75                               else
76      1       2104                     underflow <=0;
77                          end
78
79      1       5665     always @(posedge clk or negedge rst_n) begin
80                               if (!rst_n) begin
81      1       2327                     count <= 0;
82                               end
83                               else begin
84                                       if               ( ({wr_en, rd_en} == 2'b10) && !full)
85      1       1684                             count <= count + 1;
86                                       else if ( ({wr_en, rd_en} == 2'b01) && !empty)
87      1        204                             count <= count - 1;
88                                       else if (({wr_en, rd_en} == 2'b11) && empty)
89      1        299                             count <= count + 1;
90                                       else if (({wr_en, rd_en} == 2'b11) && full)
91      1         6                              count <= count - 1;
92                               end
93                          end
94
95      1       3074     always@(count)begin
96                               if (count==FIFO_DEPTH )
```

```
   97          1                14                              full=1;
   98                                                  else
   99          1              3060                              full=0;
  100                                                  if (count==0)
  101          1               983                              empty=1;
  102                                                  else
  103          1              2091                              empty=0;
  104                                                  if (count==(FIFO_DEPTH-1))
  105          1                26                              almostfull=1;
  106                                                  else
  107          1              3048                              almostfull=0;
  108                                                  if (count == 1)
  109          1              1035                              almostempty=1;
  110                                                  else
  111          1              2039                              almostempty=0;
```

Toggle Coverage:

Enabled Coverage	Bins	Hits	Misses	Coverage
Toggles	106	106	0	100.00%

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

Toggle Coverage for instance /\top#DUT_Design --

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

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

Toggle Coverage = 100.00% (106 of 106 bins)

Total Coverage By Instance (filtered view): 100.00%

Function coverage

Coverage Report by instance with details

```
================================================================================
=== Instance: /top/DUT_Design
=== Design Unit: work.FIFO
================================================================================
```

Assertion Coverage:

Assertions				
Assertions	**14**	**14**	**0**	**100.00%**

Name	File(Line)	Failure Count	Pass Count
/top/DUT_Design/FULL_assertion	FIFO_design_sv.sv(191)	0	1
/top/DUT_Design/EMPTY_assertion	FIFO_design_sv.sv(192)	0	1
/top/DUT_Design/ALMOSTFULL_assertion	FIFO_design_sv.sv(193)	0	1
/top/DUT_Design/ALMOSTEMPTY_assertion	FIFO_design_sv.sv(194)	0	1
/top/DUT_Design/OVERFLOW_assertion	FIFO_design_sv.sv(195)	0	1
/top/DUT_Design/UNDERFLOW_assertion	FIFO_design_sv.sv(196)	0	1
/top/DUT_Design/WR_ACK_HIGH_assertion	FIFO_design_sv.sv(197)	0	1
/top/DUT_Design/WR_ACK_LOW_assertion	FIFO_design_sv.sv(198)	0	1

```
/top/DUT_Design/COUNTER_0_assertion
                FIFO_design_sv.sv(199)          0       1
/top/DUT_Design/COUNTER_INC_10_assertion
                FIFO_design_sv.sv(200)          0       1
/top/DUT_Design/COUNTER_INC_01_assertion
                FIFO_design_sv.sv(201)          0       1
/top/DUT_Design/COUNTER_INC_11_WR_assertion
                FIFO_design_sv.sv(202)          0       1
/top/DUT_Design/COUNTER_INC_11_RD_assertion
                FIFO_design_sv.sv(203)          0       1
/top/DUT_Design/COUNTER_STUN_assertion
                FIFO_design_sv.sv(204)          0       1

Directive Coverage:
    Directives                  14      14       0   100.00%

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                Design Design   Lang File(Line)     Hits Status
                                    Unit   UnitType
--------------------------------------------------------------------------------------------
/top/DUT_Design/FULL_cover              FIFO   Verilog  SVA  FIFO_design_sv.sv(206)
                                                          18 Covered
/top/DUT_Design/EMPTY_cover             FIFO   Verilog  SVA  FIFO_design_sv.sv(207)
                                                        1404 Covered
/top/DUT_Design/ALMOSTFULL_cover        FIFO   Verilog  SVA  FIFO_design_sv.sv(208)
                                                          27 Covered
/top/DUT_Design/ALMOSTEMPTY_cover       FIFO   Verilog  SVA  FIFO_design_sv.sv(209)
                                                        1014 Covered
/top/DUT_Design/OVERFLOW_cover          FIFO   Verilog  SVA  FIFO_design_sv.sv(210)
                                                          11 Covered
/top/DUT_Design/UNDERFLOW_cover         FIFO   Verilog  SVA  FIFO_design_sv.sv(211)
                                                         292 Covered
/top/DUT_Design/WR_ACK_HIGH_cover       FIFO   Verilog  SVA  FIFO_design_sv.sv(212)
                                                        1729 Covered
/top/DUT_Design/WR_ACK_LOW_cover        FIFO   Verilog  SVA  FIFO_design_sv.sv(213)
                                                          11 Covered
/top/DUT_Design/COUNTER_0_cover         FIFO   Verilog  SVA  FIFO_design_sv.sv(214)
                                                        1457 Covered
/top/DUT_Design/COUNTER_INC_10_cover    FIFO   Verilog  SVA  FIFO_design_sv.sv(215)
                                                        1208 Covered
/top/DUT_Design/COUNTER_INC_01_cover    FIFO   Verilog  SVA  FIFO_design_sv.sv(216)
                                                         143 Covered
/top/DUT_Design/COUNTER_INC_11_WR_cover FIFO   Verilog  SVA  FIFO_design_sv.sv(217)
                                                         203 Covered
/top/DUT_Design/COUNTER_INC_11_RD_cover FIFO   Verilog  SVA  FIFO_design_sv.sv(218)
                                                           6 Covered
/top/DUT_Design/COUNTER_STUN_cover      FIFO   Verilog  SVA  FIFO_design_sv.sv(219)
                                                          94 Covered

================================================================================
=== Instance: /top/TESTBENCH
=== Design Unit: work.tb_FIFO
================================================================================


Assertion Coverage:
    Assertions                  1       1       0   100.00%
--------------------------------------------------------------------
Name                File(Line)              Failure     Pass
                                            Count       Count
--------------------------------------------------------------------
/top/TESTBENCH/#anonblk#176401887#71#4#/#ublk#176401887#71/immed__72
                    test_check.sv(72)               0       1

================================================================================
=== Instance: /package_coverage
=== Design Unit: work.package_coverage
================================================================================

Covergroup Coverage:
    Covergroups                 1       na      na   100.00%
        Coverpoints/Crosses     16      na      na      na
            Covergroup Bins     68      68      0   100.00%
--------------------------------------------------------------------------------------------
```

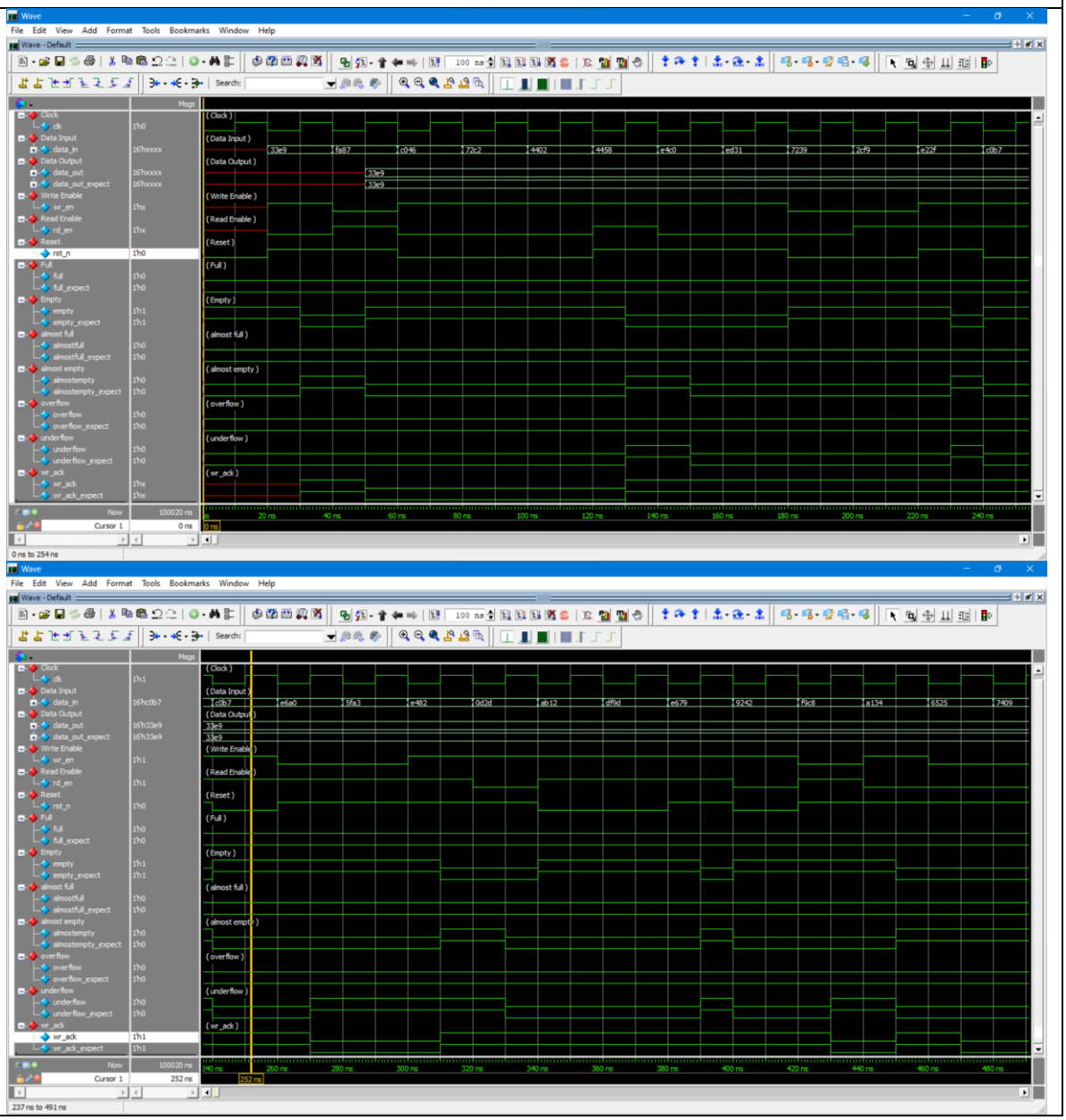
```
Covergroup                                Metric      Goal      Bins    Status

---------------------------------------------------------------------------------------------------------------
TYPE /package_coverage/FIFO_coverage/cover_group    100.00%       100         -    Covered
  covered/total bins:                                  68        68         -
  missing/total bins:                                   0        68         -
  % Hit:                                       100.00%       100         -
  Coverpoint write_enable                      100.00%       100         -    Covered
    covered/total bins:                                 2         2         -
    missing/total bins:                                 0         2         -
    % Hit:                                     100.00%       100         -
    bin write_1                                    3483         1         -    Covered
    bin write_0                                    1517         1         -    Covered
  Coverpoint read_enable                       100.00%       100         -    Covered
    covered/total bins:                                 2         2         -
    missing/total bins:                                 0         2         -
    % Hit:                                     100.00%       100         -
    bin read_1                                     1515         1         -    Covered
    bin read_0                                     3485         1         -    Covered
  Coverpoint full_flag                         100.00%       100         -    Covered
    covered/total bins:                                 2         2         -
    missing/total bins:                                 0         2         -
    % Hit:                                     100.00%       100         -
    bin full_1                                       24         1         -    Covered
    bin full_0                                     4977         1         -    Covered
  Coverpoint empty_flag                        100.00%       100         -    Covered
    covered/total bins:                                 2         2         -
    missing/total bins:                                 0         2         -
    % Hit:                                     100.00%       100         -
    bin empty_1                                    1980         1         -    Covered
    bin empty_0                                    3021         1         -    Covered
  Coverpoint almostfull_flag                   100.00%       100         -    Covered
    covered/total bins:                                 2         2         -
    missing/total bins:                                 0         2         -
    % Hit:                                     100.00%       100         -
    bin almostfull_1                                 37         1         -    Covered
    bin almostfull_0                               4964         1         -    Covered
  Coverpoint almostempty_flag                  100.00%       100         -    Covered
    covered/total bins:                                 2         2         -
    missing/total bins:                                 0         2         -
    % Hit:                                     100.00%       100         -
    bin almostempty_1                              1448         1         -    Covered
    bin almostempty_0                              3553         1         -    Covered
  Coverpoint overflow_flag                     100.00%       100         -    Covered
    covered/total bins:                                 2         2         -
    missing/total bins:                                 0         2         -
    % Hit:                                     100.00%       100         -
    bin overflow_1                                   12         1         -    Covered
    bin overflow_0                                 4989         1         -    Covered
  Coverpoint underflow_flag                    100.00%       100         -    Covered
    covered/total bins:                                 2         2         -
    missing/total bins:                                 0         2         -
    % Hit:                                     100.00%       100         -
    bin underflow_1                                 422         1         -    Covered
    bin underflow_0                                4579         1         -    Covered
  Coverpoint wr_ack_flag                       100.00%       100         -    Covered
    covered/total bins:                                 2         2         -
    missing/total bins:                                 0         2         -
    % Hit:                                     100.00%       100         -
    bin wr_ack_1                                   3443         1         -    Covered
    bin wr_ack_0                                   1557         1         -    Covered
  Cross #cross__0#                             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 <write_0,read_0,full_0>                   1056         1         -    Covered
      bin <write_0,read_0,full_1>                      4         1         -    Covered
      bin <write_1,read_0,full_0>                   2405         1         -    Covered
      bin <write_1,read_0,full_1>                     20         1         -    Covered
      bin <write_0,read_1,full_0>                    457         1         -    Covered
      bin <write_1,read_1,full_0>                   1058         1         -    Covered
    Illegal and Ignore Bins:
      ignore_bin ignored_1_will_not_happend            0                   -    ZERO
  Cross #cross__1#                             100.00%       100         -    Covered
```

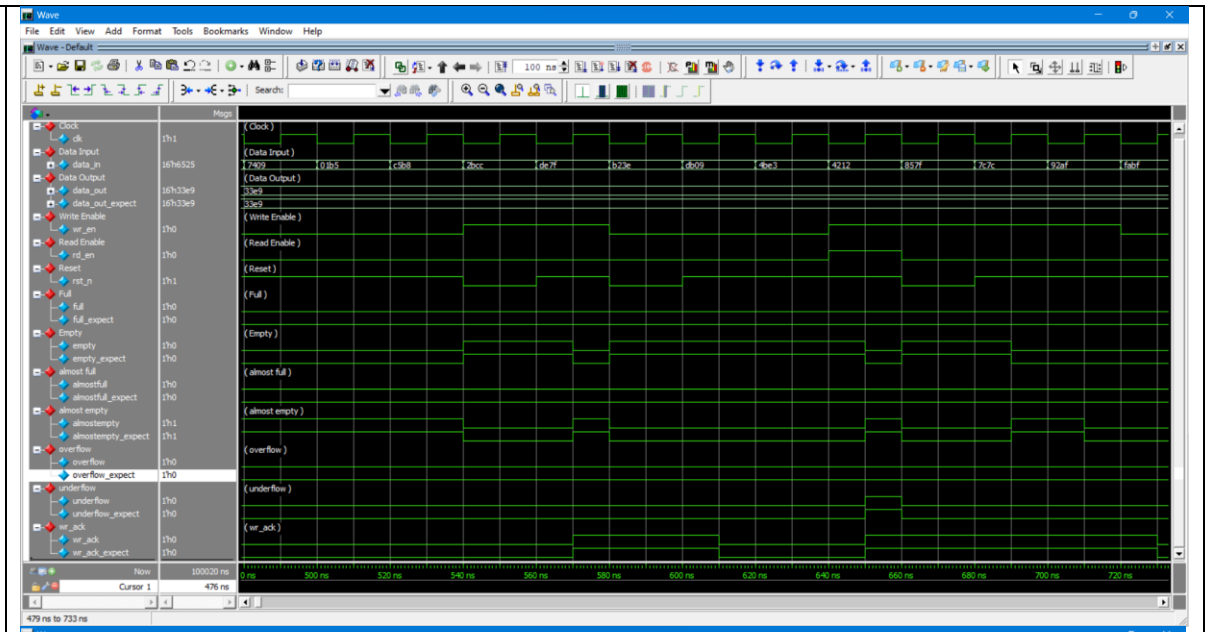
```
    covered/total bins:                    8      8      -
    missing/total bins:                    0      8      -
    % Hit:                           100.00%    100      -
    Auto, Default and User Defined Bins:
      bin <write_0,read_0,empty_0>           476      1      -  Covered
      bin <write_1,read_0,empty_0>          1690      1      -  Covered
      bin <write_0,read_1,empty_0>           102      1      -  Covered
      bin <write_1,read_1,empty_0>           753      1      -  Covered
      bin <write_0,read_0,empty_1>           584      1      -  Covered
      bin <write_1,read_0,empty_1>           735      1      -  Covered
      bin <write_0,read_1,empty_1>           355      1      -  Covered
      bin <write_1,read_1,empty_1>           305      1      -  Covered
  Cross #cross__2#                     100.00%    100      -  Covered
    covered/total bins:                    8      8      -
    missing/total bins:                    0      8      -
    % Hit:                           100.00%    100      -
    Auto, Default and User Defined Bins:
      bin <write_0,read_0,almostfull_0>      1054      1      -  Covered
      bin <write_1,read_0,almostfull_0>      2407      1      -  Covered
      bin <write_0,read_1,almostfull_0>       455      1      -  Covered
      bin <write_1,read_1,almostfull_0>      1047      1      -  Covered
      bin <write_0,read_0,almostfull_1>         6      1      -  Covered
      bin <write_1,read_0,almostfull_1>        18      1      -  Covered
      bin <write_0,read_1,almostfull_1>         2      1      -  Covered
      bin <write_1,read_1,almostfull_1>        11      1      -  Covered
  Cross #cross__3#                     100.00%    100      -  Covered
    covered/total bins:                    8      8      -
    missing/total bins:                    0      8      -
    % Hit:                           100.00%    100      -
    Auto, Default and User Defined Bins:
      bin <write_0,read_0,almostempty_0>       855      1      -  Covered
      bin <write_1,read_0,almostempty_0>      1741      1      -  Covered
      bin <write_0,read_1,almostempty_0>       405      1      -  Covered
      bin <write_1,read_1,almostempty_0>       551      1      -  Covered
      bin <write_0,read_0,almostempty_1>       205      1      -  Covered
      bin <write_1,read_0,almostempty_1>       684      1      -  Covered
      bin <write_0,read_1,almostempty_1>        52      1      -  Covered
      bin <write_1,read_1,almostempty_1>       507      1      -  Covered
  Cross #cross__4#                     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 <write_0,read_0,overflow_0>        1060      1      -  Covered
      bin <write_1,read_0,overflow_0>        2419      1      -  Covered
      bin <write_0,read_1,overflow_0>         457      1      -  Covered
      bin <write_1,read_1,overflow_0>        1052      1      -  Covered
      bin <write_1,read_0,overflow_1>           6      1      -  Covered
      bin <write_1,read_1,overflow_1>           6      1      -  Covered
    Illegal and Ignore Bins:
      ignore_bin ignored_2_will_not_happend        0              -  ZERO
  Cross #cross__5#                     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 <write_0,read_0,underflow_0>       1060      1      -  Covered
      bin <write_1,read_0,underflow_0>       2425      1      -  Covered
      bin <write_0,read_1,underflow_0>        334      1      -  Covered
      bin <write_0,read_1,underflow_1>        123      1      -  Covered
      bin <write_1,read_1,underflow_0>        759      1      -  Covered
      bin <write_1,read_1,underflow_1>        299      1      -  Covered
    Illegal and Ignore Bins:
      ignore_bin ignored_3_will_not_happend        0              -  ZERO
  Cross #cross__6#                     100.00%    100      -  Covered
    covered/total bins:                    8      8      -
    missing/total bins:                    0      8      -
    % Hit:                           100.00%    100      -
    Auto, Default and User Defined Bins:
      bin <write_0,read_0,wr_ack_0>          855      1      -  Covered
      bin <write_1,read_0,wr_ack_0>          237      1      -  Covered
      bin <write_0,read_1,wr_ack_0>          365      1      -  Covered
      bin <write_1,read_1,wr_ack_0>          100      1      -  Covered
      bin <write_0,read_0,wr_ack_1>          205      1      -  Covered
      bin <write_1,read_0,wr_ack_1>         2188      1      -  Covered
```

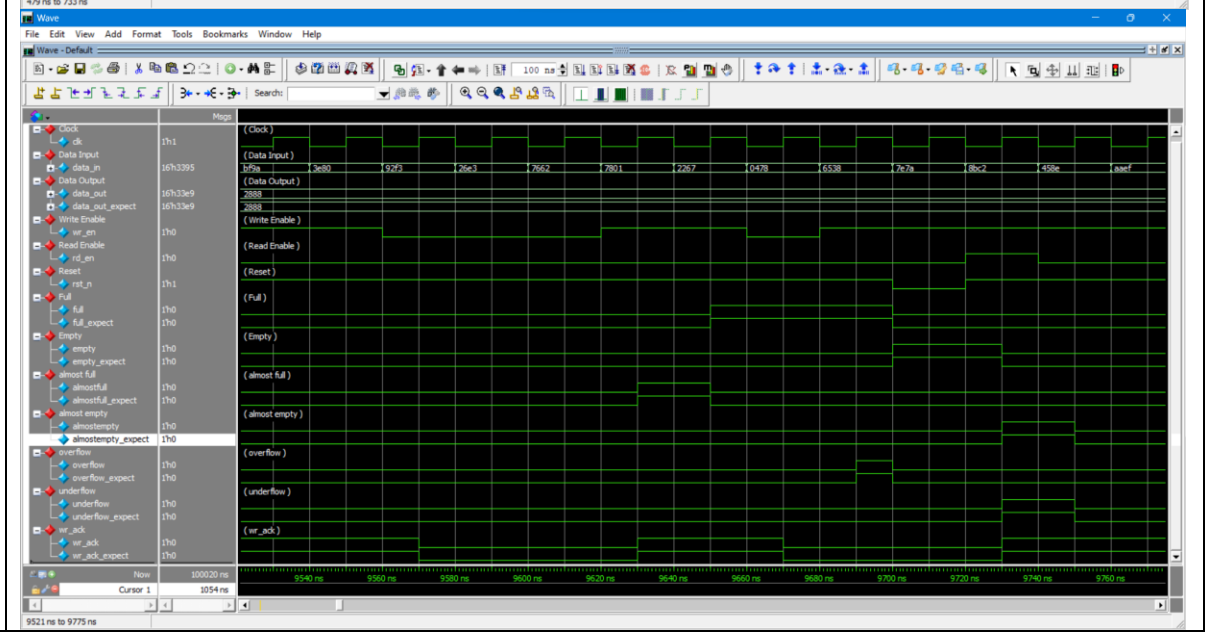
```
        bin <write_0,read_1,wr_ack_1>              92        1       -   Covered
        bin <write_1,read_1,wr_ack_1>              958       1       -   Covered

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

----------------------------------------------------------------------------------------------------------------
 TYPE /package_coverage/FIFO_coverage/cover_group    100.00%       100       -   Covered
   covered/total bins:                               68        68        -
   missing/total bins:                               0         68        -
   % Hit:                                 100.00%    100       -
   Coverpoint write_enable                           100.00%   100       -   Covered
     covered/total bins:                             2         2         -
     missing/total bins:                             0         2         -
     % Hit:                               100.00%    100       -
     bin write_1                           3483      1         -   Covered
     bin write_0                           1517      1         -   Covered
   Coverpoint read_enable                            100.00%   100       -   Covered
     covered/total bins:                             2         2         -
     missing/total bins:                             0         2         -
     % Hit:                               100.00%    100       -
     bin read_1                            1515      1         -   Covered
     bin read_0                            3485      1         -   Covered
   Coverpoint full_flag                              100.00%   100       -   Covered
     covered/total bins:                             2         2         -
     missing/total bins:                             0         2         -
     % Hit:                               100.00%    100       -
     bin full_1                            24        1         -   Covered
     bin full_0                            4977      1         -   Covered
   Coverpoint empty_flag                             100.00%   100       -   Covered
     covered/total bins:                             2         2         -
     missing/total bins:                             0         2         -
     % Hit:                               100.00%    100       -
     bin empty_1                           1980      1         -   Covered
     bin empty_0                           3021      1         -   Covered
   Coverpoint almostfull_flag                        100.00%   100       -   Covered
     covered/total bins:                             2         2         -
     missing/total bins:                             0         2         -
     % Hit:                               100.00%    100       -
     bin almostfull_1                      37        1         -   Covered
     bin almostfull_0                      4964      1         -   Covered
   Coverpoint almostempty_flag                       100.00%   100       -   Covered
     covered/total bins:                             2         2         -
     missing/total bins:                             0         2         -
     % Hit:                               100.00%    100       -
     bin almostempty_1                     1448      1         -   Covered
     bin almostempty_0                     3553      1         -   Covered
   Coverpoint overflow_flag                          100.00%   100       -   Covered
     covered/total bins:                             2         2         -
     missing/total bins:                             0         2         -
     % Hit:                               100.00%    100       -
     bin overflow_1                        12        1         -   Covered
     bin overflow_0                        4989      1         -   Covered
   Coverpoint underflow_flag                         100.00%   100       -   Covered
     covered/total bins:                             2         2         -
     missing/total bins:                             0         2         -
     % Hit:                               100.00%    100       -
     bin underflow_1                       422       1         -   Covered
     bin underflow_0                       4579      1         -   Covered
   Coverpoint wr_ack_flag                            100.00%   100       -   Covered
     covered/total bins:                             2         2         -
     missing/total bins:                             0         2         -
     % Hit:                               100.00%    100       -
     bin wr_ack_1                          3443      1         -   Covered
     bin wr_ack_0                          1557      1         -   Covered
   Cross #cross__0#                                  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 <write_0,read_0,full_0>                   1056      1         -   Covered
       bin <write_0,read_0,full_1>                   4         1         -   Covered
       bin <write_1,read_0,full_0>                   2405      1         -   Covered
       bin <write_1,read_0,full_1>                   20        1         -   Covered
```

```
        bin <write_0,read_1,full_0>              457       1       -   Covered
        bin <write_1,read_1,full_0>              1058      1       -   Covered
    Illegal and Ignore Bins:
        ignore_bin ignored_1_will_not_happend    0               -   ZERO
Cross #cross__1#                         100.00%      100       -   Covered
    covered/total bins:                      8        8         -
    missing/total bins:                      0        8         -
    % Hit:                           100.00%      100       -
    Auto, Default and User Defined Bins:
        bin <write_0,read_0,empty_0>             476       1       -   Covered
        bin <write_1,read_0,empty_0>             1690      1       -   Covered
        bin <write_0,read_1,empty_0>             102       1       -   Covered
        bin <write_1,read_1,empty_0>             753       1       -   Covered
        bin <write_0,read_0,empty_1>             584       1       -   Covered
        bin <write_1,read_0,empty_1>             735       1       -   Covered
        bin <write_0,read_1,empty_1>             355       1       -   Covered
        bin <write_1,read_1,empty_1>             305       1       -   Covered
Cross #cross__2#                         100.00%      100       -   Covered
    covered/total bins:                      8        8         -
    missing/total bins:                      0        8         -
    % Hit:                           100.00%      100       -
    Auto, Default and User Defined Bins:
        bin <write_0,read_0,almostfull_0>        1054      1       -   Covered
        bin <write_1,read_0,almostfull_0>        2407      1       -   Covered
        bin <write_0,read_1,almostfull_0>        455       1       -   Covered
        bin <write_1,read_1,almostfull_0>        1047      1       -   Covered
        bin <write_0,read_0,almostfull_1>        6         1       -   Covered
        bin <write_1,read_0,almostfull_1>        18        1       -   Covered
        bin <write_0,read_1,almostfull_1>        2         1       -   Covered
        bin <write_1,read_1,almostfull_1>        11        1       -   Covered
Cross #cross__3#                         100.00%      100       -   Covered
    covered/total bins:                      8        8         -
    missing/total bins:                      0        8         -
    % Hit:                           100.00%      100       -
    Auto, Default and User Defined Bins:
        bin <write_0,read_0,almostempty_0>       855       1       -   Covered
        bin <write_1,read_0,almostempty_0>       1741      1       -   Covered
        bin <write_0,read_1,almostempty_0>       405       1       -   Covered
        bin <write_1,read_1,almostempty_0>       551       1       -   Covered
        bin <write_0,read_0,almostempty_1>       205       1       -   Covered
        bin <write_1,read_0,almostempty_1>       684       1       -   Covered
        bin <write_0,read_1,almostempty_1>       52        1       -   Covered
        bin <write_1,read_1,almostempty_1>       507       1       -   Covered
Cross #cross__4#                         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 <write_0,read_0,overflow_0>          1060      1       -   Covered
        bin <write_1,read_0,overflow_0>          2419      1       -   Covered
        bin <write_0,read_1,overflow_0>          457       1       -   Covered
        bin <write_1,read_1,overflow_0>          1052      1       -   Covered
        bin <write_1,read_0,overflow_1>          6         1       -   Covered
        bin <write_1,read_1,overflow_1>          6         1       -   Covered
    Illegal and Ignore Bins:
        ignore_bin ignored_2_will_not_happend    0               -   ZERO
Cross #cross__5#                         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 <write_0,read_0,underflow_0>         1060      1       -   Covered
        bin <write_1,read_0,underflow_0>         2425      1       -   Covered
        bin <write_0,read_1,underflow_0>         334       1       -   Covered
        bin <write_0,read_1,underflow_1>         123       1       -   Covered
        bin <write_1,read_1,underflow_0>         759       1       -   Covered
        bin <write_1,read_1,underflow_1>         299       1       -   Covered
    Illegal and Ignore Bins:
        ignore_bin ignored_3_will_not_happend    0               -   ZERO
Cross #cross__6#                         100.00%      100       -   Covered
    covered/total bins:                      8        8         -
    missing/total bins:                      0        8         -
    % Hit:                           100.00%      100       -
    Auto, Default and User Defined Bins:
        bin <write_0,read_0,wr_ack_0>            855       1       -   Covered
```

```
        bin <write_1,read_0,wr_ack_0>          237       1       -   Covered
        bin <write_0,read_1,wr_ack_0>          365       1       -   Covered
        bin <write_1,read_1,wr_ack_0>          100       1       -   Covered
        bin <write_0,read_0,wr_ack_1>          205       1       -   Covered
        bin <write_1,read_0,wr_ack_1>         2188       1       -   Covered
        bin <write_0,read_1,wr_ack_1>           92       1       -   Covered
        bin <write_1,read_1,wr_ack_1>          958       1       -   Covered

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                 Design Design   Lang File(Line)     Hits Status
                                     Unit   UnitType
--------------------------------------------------------------------------------------------
/top/DUT_Design/FULL_cover           FIFO   Verilog  SVA  FIFO_design_sv.sv(206)
                                                                  18 Covered
/top/DUT_Design/EMPTY_cover          FIFO   Verilog  SVA  FIFO_design_sv.sv(207)
                                                                1404 Covered
/top/DUT_Design/ALMOSTFULL_cover     FIFO   Verilog  SVA  FIFO_design_sv.sv(208)
                                                                  27 Covered
/top/DUT_Design/ALMOSTEMPTY_cover    FIFO   Verilog  SVA  FIFO_design_sv.sv(209)
                                                                1014 Covered
/top/DUT_Design/OVERFLOW_cover       FIFO   Verilog  SVA  FIFO_design_sv.sv(210)
                                                                  11 Covered
/top/DUT_Design/UNDERFLOW_cover      FIFO   Verilog  SVA  FIFO_design_sv.sv(211)
                                                                 292 Covered
/top/DUT_Design/WR_ACK_HIGH_cover    FIFO   Verilog  SVA  FIFO_design_sv.sv(212)
                                                                1729 Covered
/top/DUT_Design/WR_ACK_LOW_cover     FIFO   Verilog  SVA  FIFO_design_sv.sv(213)
                                                                  11 Covered
/top/DUT_Design/COUNTER_0_cover      FIFO   Verilog  SVA  FIFO_design_sv.sv(214)
                                                                1457 Covered
/top/DUT_Design/COUNTER_INC_10_cover FIFO   Verilog  SVA  FIFO_design_sv.sv(215)
                                                                1208 Covered
/top/DUT_Design/COUNTER_INC_01_cover FIFO   Verilog  SVA  FIFO_design_sv.sv(216)
                                                                 143 Covered
/top/DUT_Design/COUNTER_INC_11_WR_cover FIFO  Verilog  SVA  FIFO_design_sv.sv(217)
                                                                 203 Covered
/top/DUT_Design/COUNTER_INC_11_RD_cover FIFO  Verilog  SVA  FIFO_design_sv.sv(218)
                                                                   6 Covered
/top/DUT_Design/COUNTER_STUN_cover   FIFO   Verilog  SVA  FIFO_design_sv.sv(219)
                                                                  94 Covered

TOTAL DIRECTIVE COVERAGE: 100.00%  COVERS: 14

ASSERTION RESULTS:
---------------------------------------------------------------------
Name              File(Line)              Failure     Pass
                                          Count       Count
---------------------------------------------------------------------
/top/DUT_Design/FULL_assertion
                  FIFO_design_sv.sv(191)          0       1
/top/DUT_Design/EMPTY_assertion
                  FIFO_design_sv.sv(192)          0       1
/top/DUT_Design/ALMOSTFULL_assertion
                  FIFO_design_sv.sv(193)          0       1
/top/DUT_Design/ALMOSTEMPTY_assertion
                  FIFO_design_sv.sv(194)          0       1
/top/DUT_Design/OVERFLOW_assertion
                  FIFO_design_sv.sv(195)          0       1
/top/DUT_Design/UNDERFLOW_assertion
                  FIFO_design_sv.sv(196)          0       1
/top/DUT_Design/WR_ACK_HIGH_assertion
                  FIFO_design_sv.sv(197)          0       1
/top/DUT_Design/WR_ACK_LOW_assertion
                  FIFO_design_sv.sv(198)          0       1
/top/DUT_Design/COUNTER_0_assertion
                  FIFO_design_sv.sv(199)          0       1
/top/DUT_Design/COUNTER_INC_10_assertion
                  FIFO_design_sv.sv(200)          0       1
/top/DUT_Design/COUNTER_INC_01_assertion
                  FIFO_design_sv.sv(201)          0       1
/top/DUT_Design/COUNTER_INC_11_WR_assertion
                  FIFO_design_sv.sv(202)          0       1
```

	/top/DUT_Design/COUNTER_INC_11_RD_assertion FIFO_design_sv.sv(203) 0 1 /top/DUT_Design/COUNTER_STUN_assertion FIFO_design_sv.sv(204) 0 1 /top/TESTBENCH/#anonblk#176401887#71#4#/#ublk#176401887#71/immed__72 test_check.sv(72) 0 1 **Total Coverage By Instance (filtered view): 100.00%**
Questa Snippets	


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Clock
Data Input
Data Output
Write Enable
Read Enable
Reset
Full
Empty
almost full
almost empty
overflow
underflow
wr_ack
Now
Cursor 1
479 ns to 733 ns


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Clock
Data Input
Data Output
Write Enable
Read Enable
Reset
Full
Empty
almost full
almost empty
overflow
underflow
wr_ack
Now
Cursor 1
9521 ns to 9775 ns

Verification req document

Label	Description	Stimulus Generation	Functional Coverage	Functionality Check
FIFO_reset	incase reset is asserted	Under constrain for a randomize stimulus when rst_n = 0 then it will be high	included in coverpoint to hit and count the appearance of all the values bins bins empty_1 = {1}; bins full_0 = {0}; bins almostfull_0 = {0}; bins almostempty_0 = {0}; bins overflow_0 = {0}; bins underflow_0 = {0};	outputs will be checked to equal zero by comparing with Golden_model
Writing_State	incase wr_en is asserted and rd_en is not	under constrain with size = 8 for the FIFO with random data_in wr_en equal 1 and rd_en equal 0	included in coverpoint to hit and count the appearance of all the values bins bins write_1 = {1}; bins full_1 = {1}; bins empty_1 = {1}; bins empty_0 = {0}; bins full_0 = {0}; bins almostfull_1 = {1}; bins almostfull_0 = {0}; bins almostempty_1 = {1}; bins almostempty_0 = {0}; bins overflow_1 = {1}; bins overflow_0 = {0}; bins wr_ack_1 = {1}; bins wr_ack_0 = {0};	Output will be ched by comparing outputs to Golden_model outputs
Reading_state	incase rd_en is asserted and wr_en is not	under constrain with size = 8 for the FIFO with random data_in wr_en equal 0 and rd_en equal 1	included in coverpoint to hit and count the appearance of all the values bins bins read_1 = {1}; bins full_1 = {1}; bins empty_1 = {1}; bins empty_0 = {0}; bins full_0 = {0}; bins almostfull_1 = {1}; bins almostfull_0 = {0}; bins almostempty_1 = {1}; bins almostempty_0 = {0}; bins overflow_1 = {1}; bins overflow_0 = {0}; bins wr_ack_1 = {1}; bins wr_ack_0 = {0};	Output will be ched by comparing outputs to Golden_model outputs
Writing_Reading_state	in this case , write enable and read enable came together so it will have 3 states 1) the FIFO empty so no read but there is a write . 2) the FIFO full so not write but there is a read . 3) the FIFO is not empty and not full	under constrain with size = 8 for the FIFO with random data_in wr_en equal 1 and rd_en equal 1	included in coverpoint to hit and count the appearance of all the values bins bins read_1 = {1}; bins full_1 = {1}; bins empty_1 = {1}; bins empty_0 = {0}; bins full_0 = {0}; bins almostfull_1 = {1}; bins almostfull_0 = {0}; bins almostempty_1 = {1}; bins almostempty_0 = {0}; bins overflow_1 = {1}; bins overflow_0 = {0}; bins wr_ack_1 = {1}; bins wr_ack_0 = {0};	Output will be ched by comparing outputs to Golden_model outputs