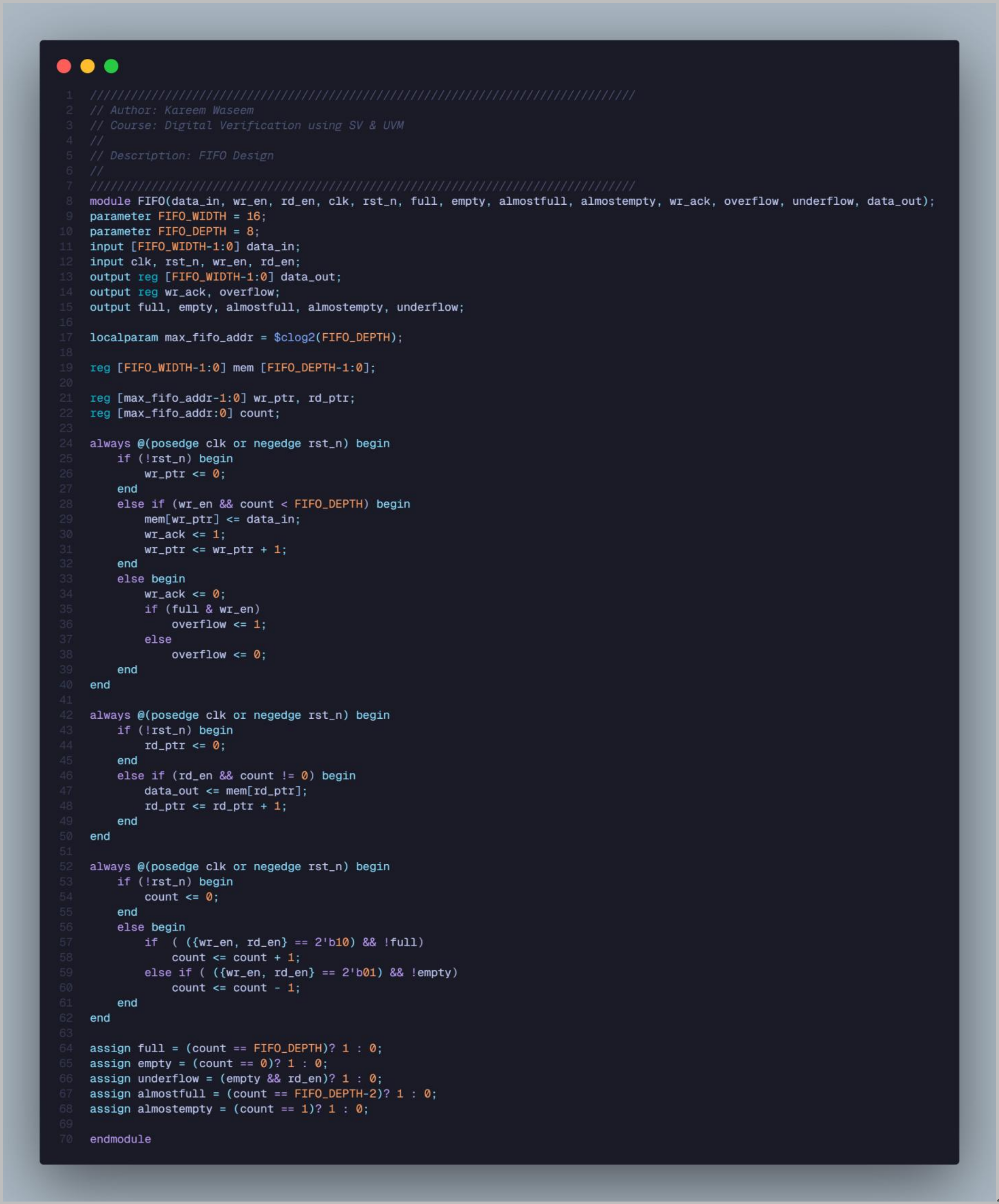
SYNC FIFO

SV Project

Muhammad Naim

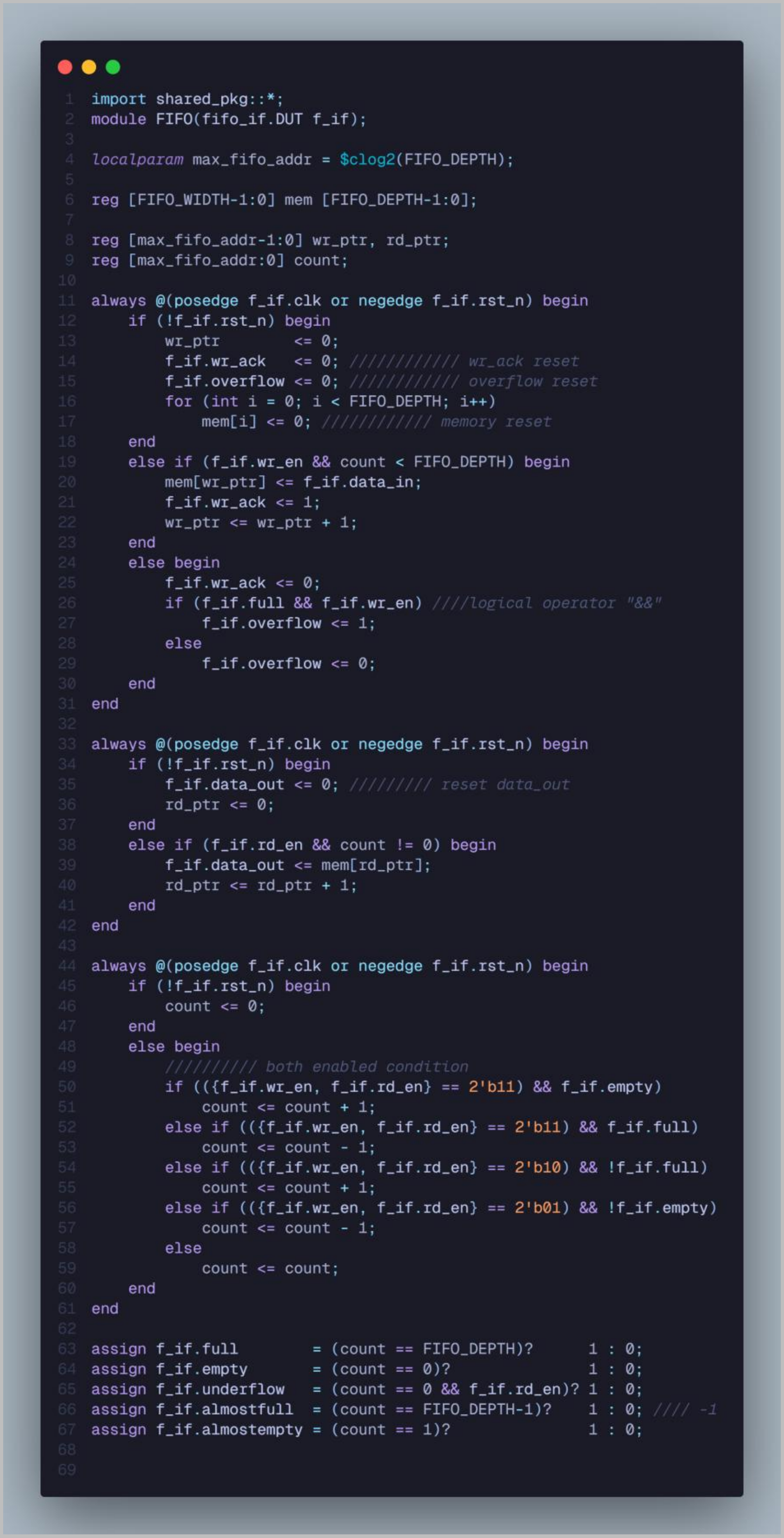
SYNC FIFO

Original design



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

localparam max_fifo_addr = $clog2(FIFO_DEPTH);

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

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

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

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

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

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

endmodule
```

After fixing



```
import shared_pkg::*;
module FIFO(fifo_if.DUT f_if);

localparam max_fifo_addr = $clog2(FIFO_DEPTH);

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

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

always @(posedge f_if.clk or negedge f_if.rst_n) begin
    if (!f_if.rst_n) begin
        wr_ptr        <= 0;
        f_if.wr_ack   <= 0; //////////// wr_ack reset
        f_if.overflow <= 0; //////////// overflow reset
        for (int i = 0; i < FIFO_DEPTH; i++)
            mem[i] <= 0; //////////// memory reset
    end
    else if (f_if.wr_en && count < FIFO_DEPTH) begin
        mem[wr_ptr] <= f_if.data_in;
        f_if.wr_ack <= 1;
        wr_ptr <= wr_ptr + 1;
    end
    else begin
        f_if.wr_ack <= 0;
        if (f_if.full && f_if.wr_en) ////logical operator "&&"
            f_if.overflow <= 1;
        else
            f_if.overflow <= 0;
    end
end

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

always @(posedge f_if.clk or negedge f_if.rst_n) begin
    if (!f_if.rst_n) begin
        count <= 0;
    end
    else begin
        ////////// both enabled condition
        if (({f_if.wr_en, f_if.rd_en} == 2'b11) && f_if.empty)
            count <= count + 1;
        else if (({f_if.wr_en, f_if.rd_en} == 2'b11) && f_if.full)
            count <= count - 1;
        else if (({f_if.wr_en, f_if.rd_en} == 2'b10) && !f_if.full)
            count <= count + 1;
        else if (({f_if.wr_en, f_if.rd_en} == 2'b01) && !f_if.empty)
            count <= count - 1;
        else
            count <= count;
    end
end

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


```

Bugs:

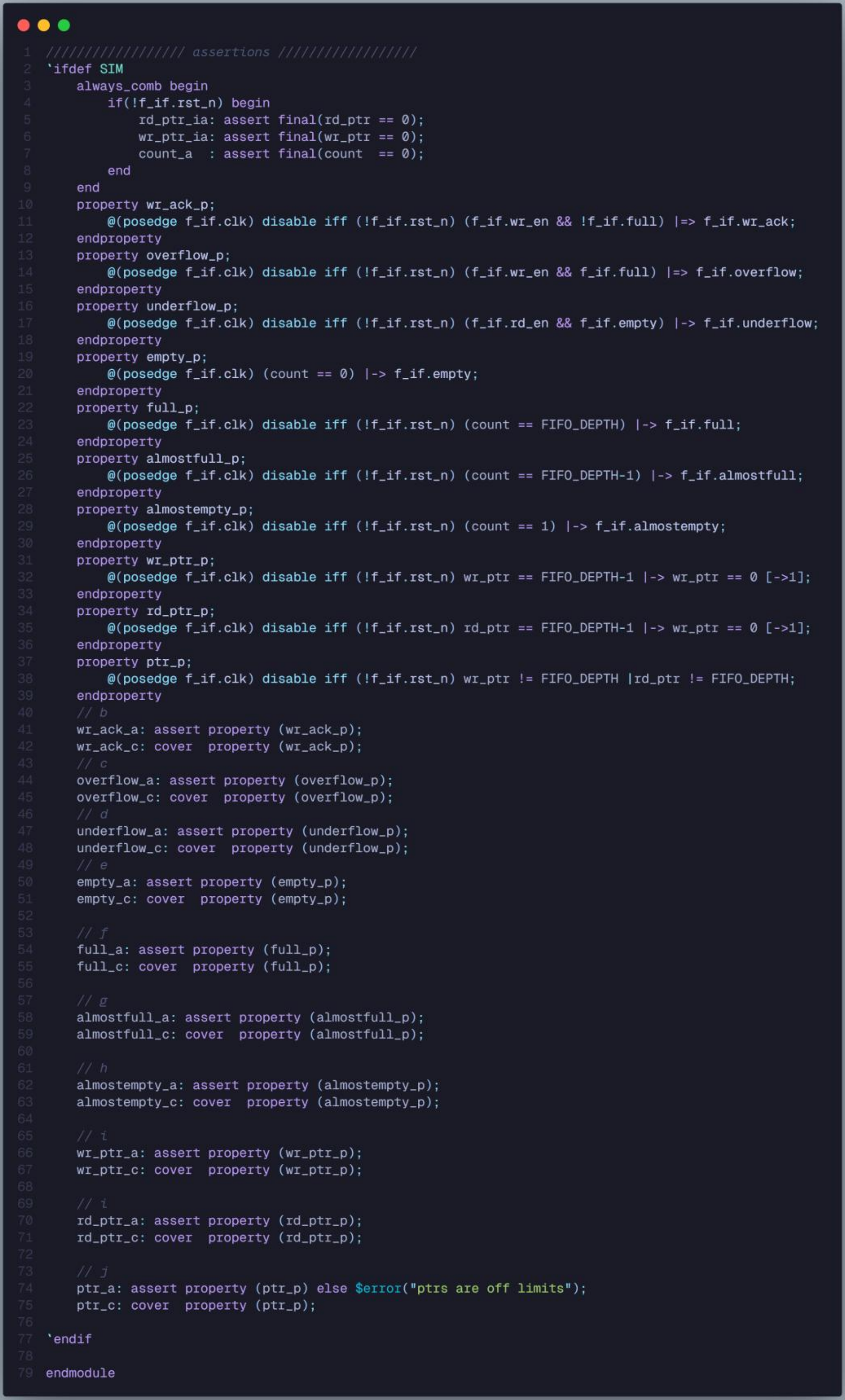
1. *FIFO "data_out", "mem", "overflow", and "wr_ack" signals are not in reset.*

2. *Using reduction operator "&" instead of a logical operator "&&" in overflow condition (line 26).*

3. *Missing "count" condition if both "wr_en" and "rd_en" signals are high.*

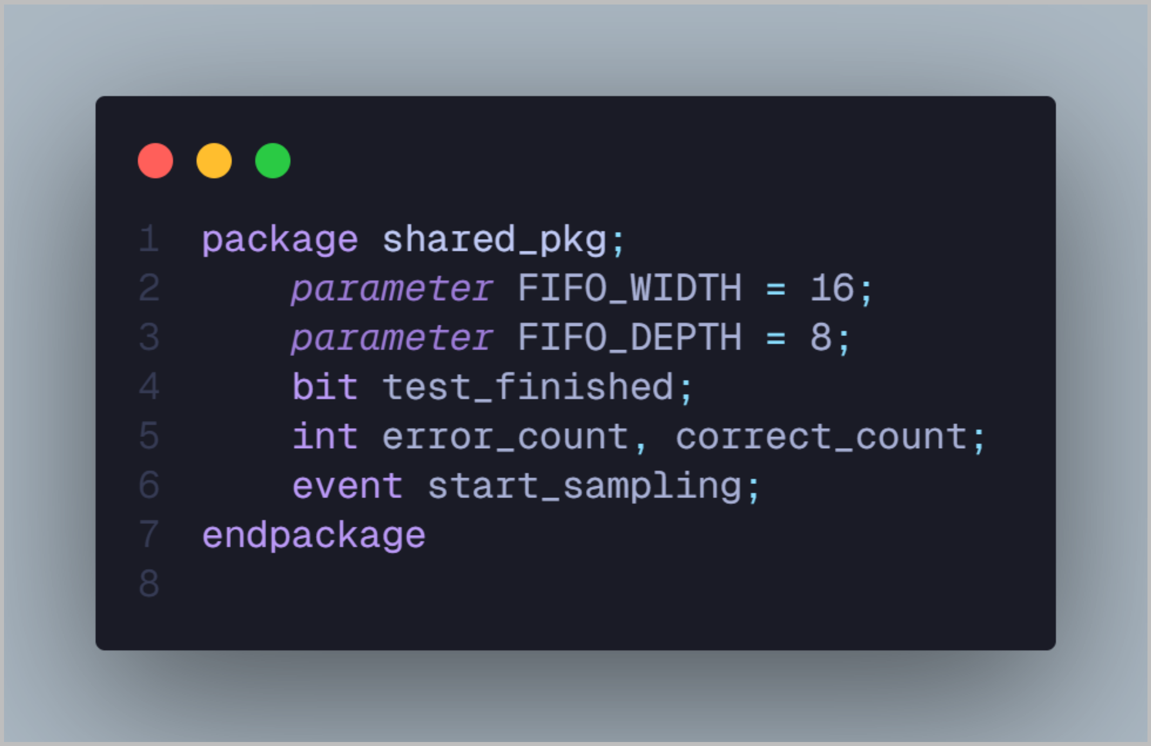
4. *"**-2**" instead of "**-1**" in the "almostfull" condition*

Design Assertions



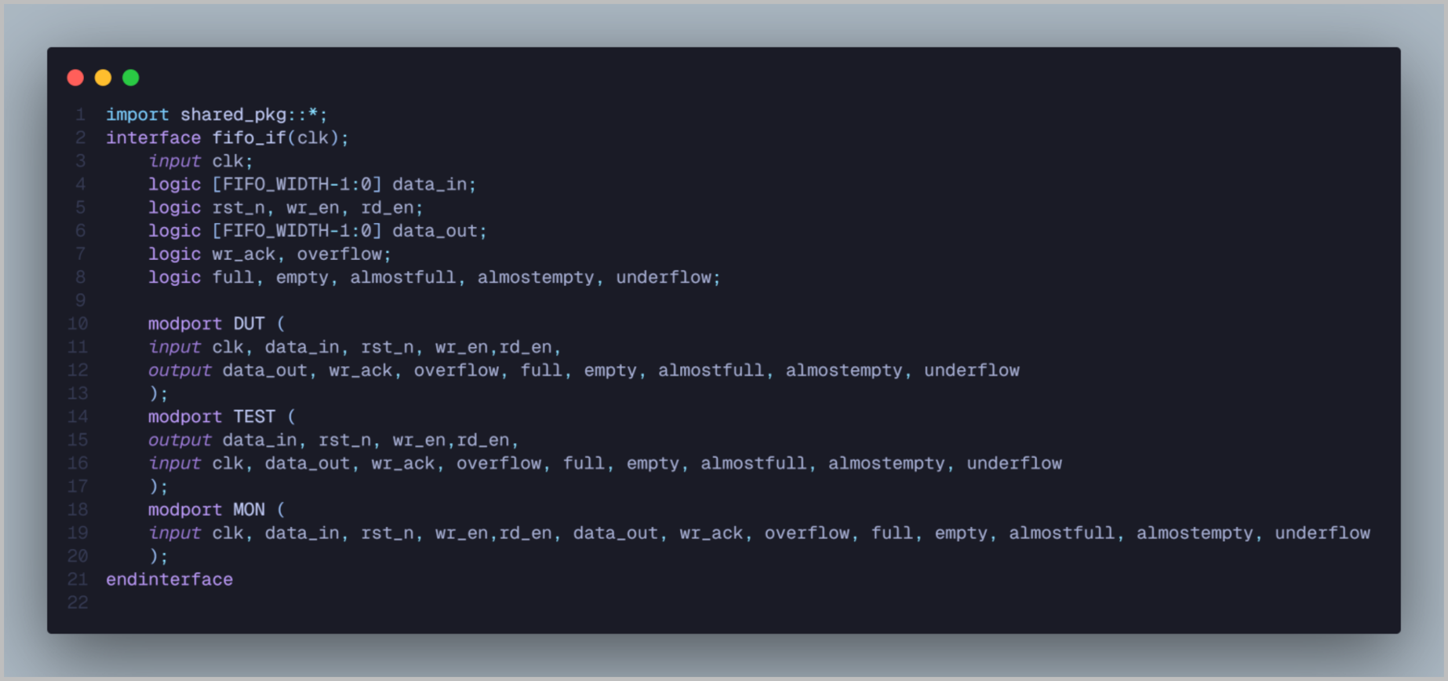
```
////////////////// assertions //////////////////
`ifdef SIM
    always_comb begin
        if(!f_if.rst_n) begin
            rd_ptr_ia: assert final(rd_ptr == 0);
            wr_ptr_ia: assert final(wr_ptr == 0);
            count_a  : assert final(count  == 0);
        end
    end
    property wr_ack_p;
        @(posedge f_if.clk) disable iff (!f_if.rst_n) (f_if.wr_en && !f_if.full) |=> f_if.wr_ack;
    endproperty
    property overflow_p;
        @(posedge f_if.clk) disable iff (!f_if.rst_n) (f_if.wr_en && f_if.full) |=> f_if.overflow;
    endproperty
    property underflow_p;
        @(posedge f_if.clk) disable iff (!f_if.rst_n) (f_if.rd_en && f_if.empty) |-> f_if.underflow;
    endproperty
    property empty_p;
        @(posedge f_if.clk) (count == 0) |-> f_if.empty;
    endproperty
    property full_p;
        @(posedge f_if.clk) disable iff (!f_if.rst_n) (count == FIFO_DEPTH) |-> f_if.full;
    endproperty
    property almostfull_p;
        @(posedge f_if.clk) disable iff (!f_if.rst_n) (count == FIFO_DEPTH-1) |-> f_if.almostfull;
    endproperty
    property almostempty_p;
        @(posedge f_if.clk) disable iff (!f_if.rst_n) (count == 1) |-> f_if.almostempty;
    endproperty
    property wr_ptr_p;
        @(posedge f_if.clk) disable iff (!f_if.rst_n) wr_ptr == FIFO_DEPTH-1 |-> wr_ptr == 0 [->1];
    endproperty
    property rd_ptr_p;
        @(posedge f_if.clk) disable iff (!f_if.rst_n) rd_ptr == FIFO_DEPTH-1 |-> wr_ptr == 0 [->1];
    endproperty
    property ptr_p;
        @(posedge f_if.clk) disable iff (!f_if.rst_n) wr_ptr != FIFO_DEPTH |rd_ptr != FIFO_DEPTH;
    endproperty
    // b
    wr_ack_a: assert property (wr_ack_p);
    wr_ack_c: cover  property (wr_ack_p);
    // c
    overflow_a: assert property (overflow_p);
    overflow_c: cover  property (overflow_p);
    // d
    underflow_a: assert property (underflow_p);
    underflow_c: cover  property (underflow_p);
    // e
    empty_a: assert property (empty_p);
    empty_c: cover  property (empty_p);

    // f
    full_a: assert property (full_p);
    full_c: cover  property (full_p);

    // g
    almostfull_a: assert property (almostfull_p);
    almostfull_c: cover  property (almostfull_p);

    // h
    almostempty_a: assert property (almostempty_p);
    almostempty_c: cover  property (almostempty_p);

    // i
    wr_ptr_a: assert property (wr_ptr_p);
    wr_ptr_c: cover  property (wr_ptr_p);

    // i
    rd_ptr_a: assert property (rd_ptr_p);
    rd_ptr_c: cover  property (rd_ptr_p);

    // j
    ptr_a: assert property (ptr_p) else $error("ptrs are off limits");
    ptr_c: cover  property (ptr_p);

`endif

endmodule
```

Shared package



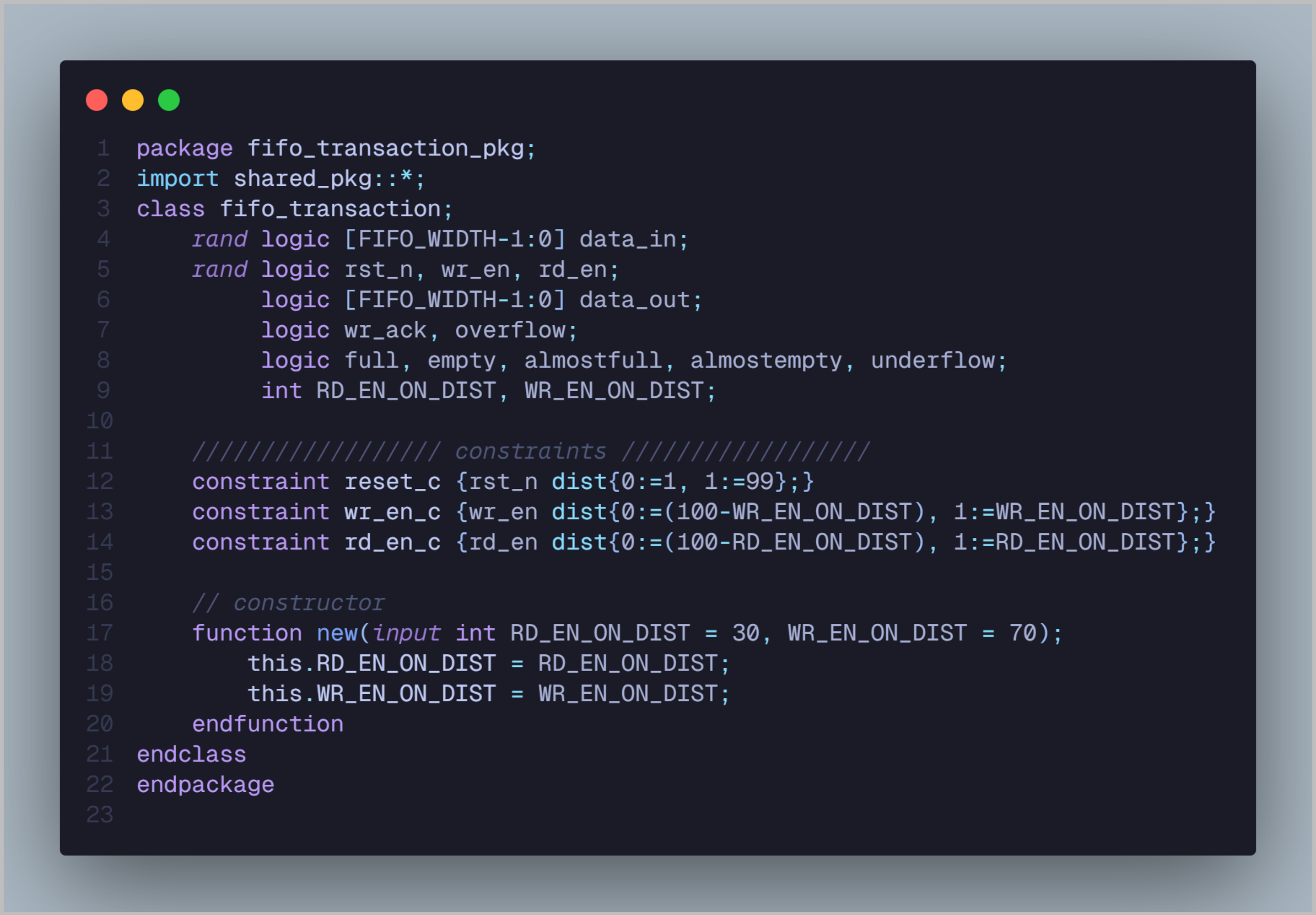
```
package shared_pkg;
    parameter FIFO_WIDTH = 16;
    parameter FIFO_DEPTH = 8;
    bit test_finished;
    int error_count, correct_count;
    event start_sampling;
endpackage
```

Interface



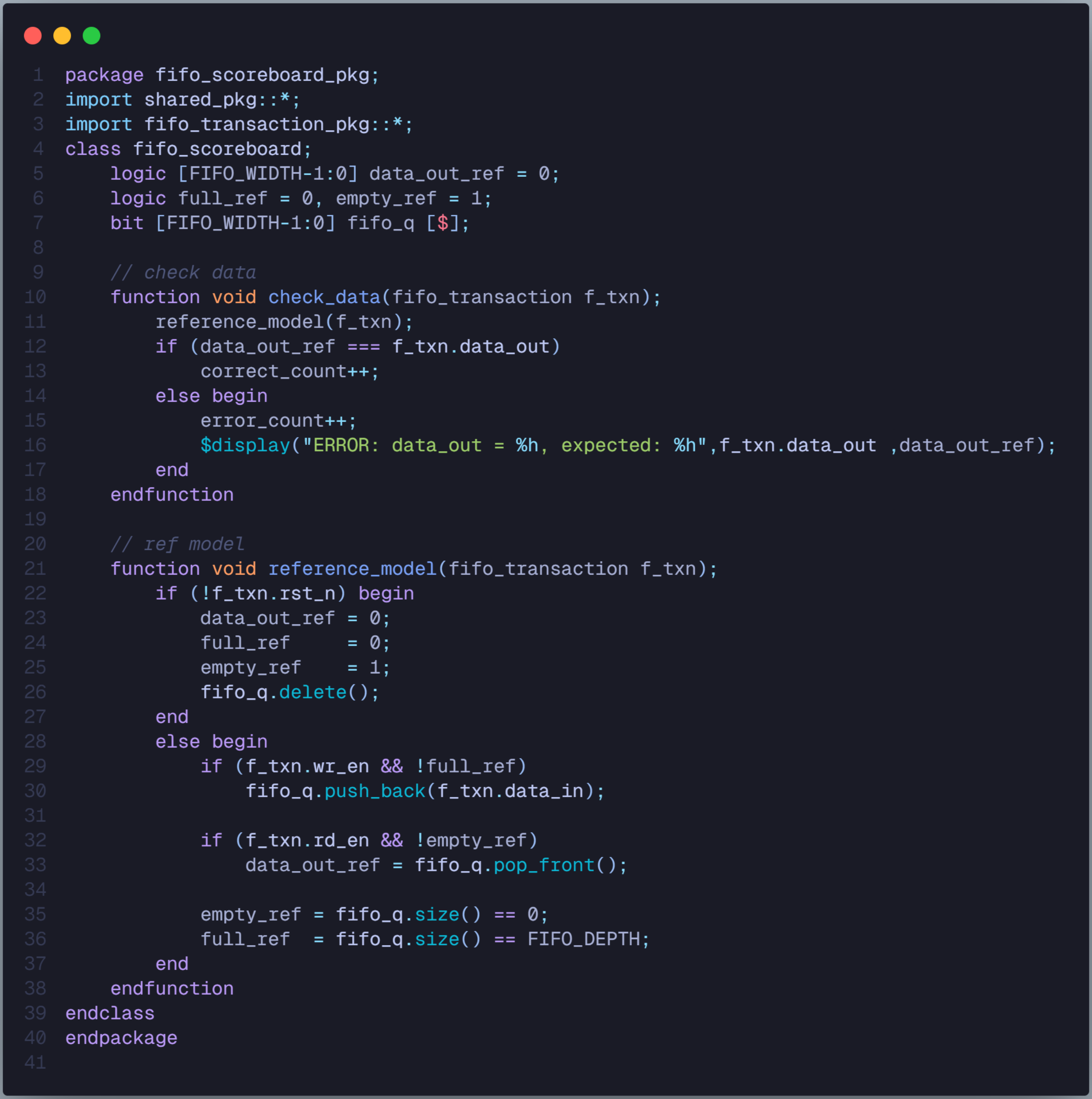
```
import shared_pkg::*;
interface fifo_if(clk);
    input clk;
    logic [FIFO_WIDTH-1:0] data_in;
    logic rst_n, wr_en, rd_en;
    logic [FIFO_WIDTH-1:0] data_out;
    logic wr_ack, overflow;
    logic full, empty, almostfull, almostempty, underflow;

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

Transaction package



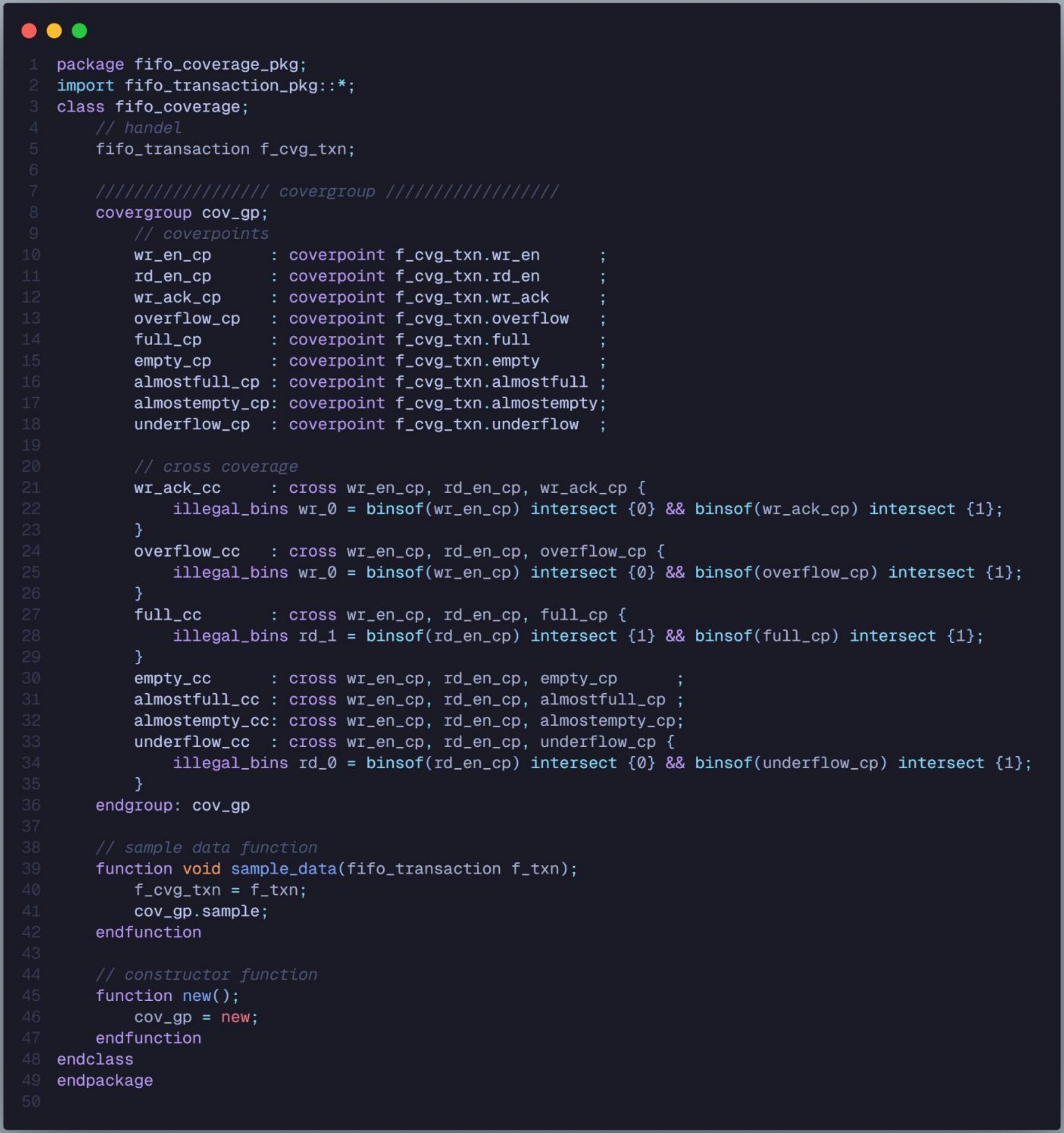
```
package fifo_transaction_pkg;
import shared_pkg::*;
class fifo_transaction;
    rand logic [FIFO_WIDTH-1:0] data_in;
    rand logic rst_n, wr_en, rd_en;
         logic [FIFO_WIDTH-1:0] data_out;
         logic wr_ack, overflow;
         logic full, empty, almostfull, almostempty, underflow;
         int RD_EN_ON_DIST, WR_EN_ON_DIST;

    ////////////////// constraints //////////////////
    constraint reset_c {rst_n dist{0:=1, 1:=99};}
    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};}

    // constructor
    function new(input int RD_EN_ON_DIST = 30, WR_EN_ON_DIST = 70);
        this.RD_EN_ON_DIST = RD_EN_ON_DIST;
        this.WR_EN_ON_DIST = WR_EN_ON_DIST;
    endfunction
endclass
endpackage
```

Scoreboard package



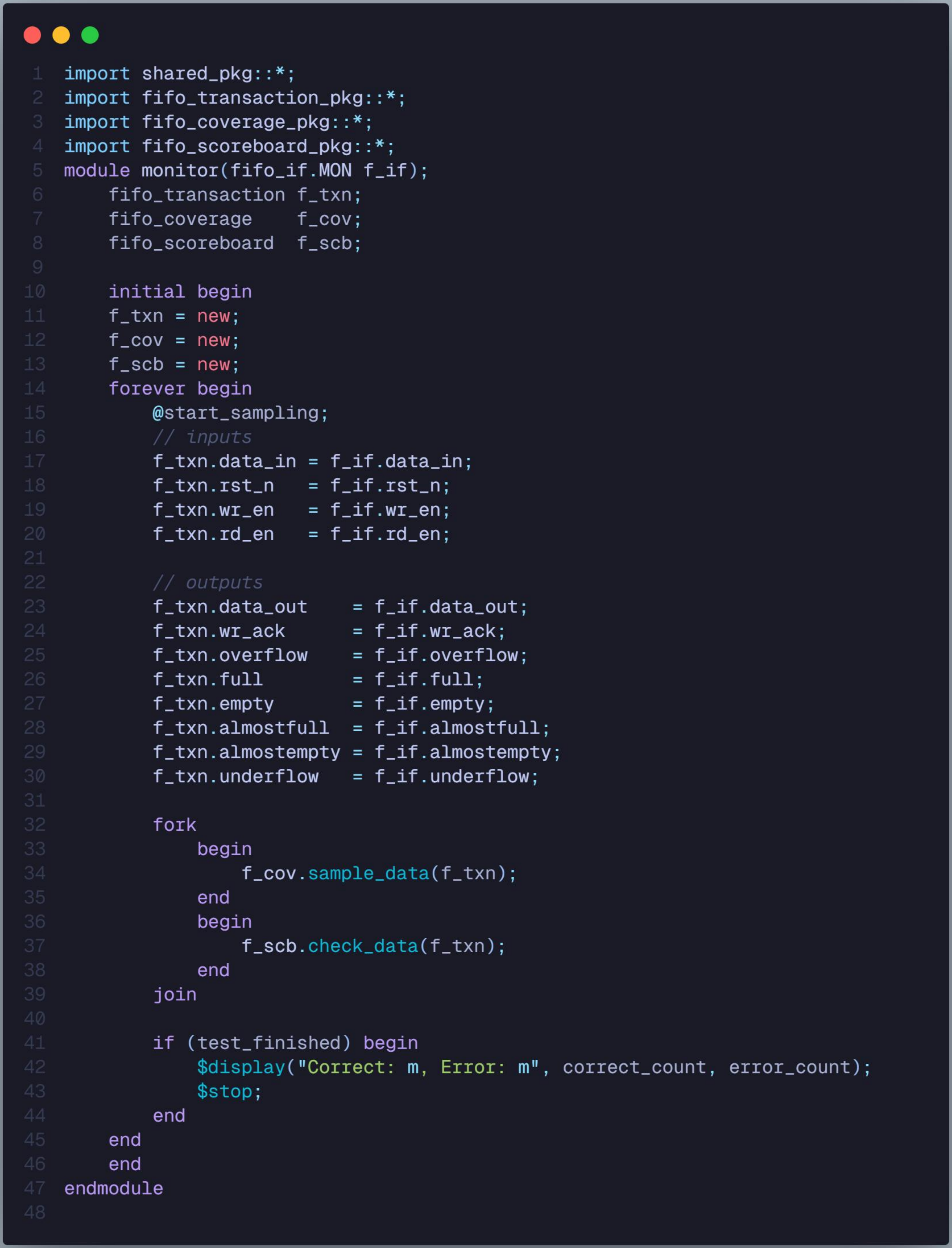
```systemverilog
package fifo_scoreboard_pkg;
import shared_pkg::*;
import fifo_transaction_pkg::*;
class fifo_scoreboard;
    logic [FIFO_WIDTH-1:0] data_out_ref = 0;
    logic full_ref = 0, empty_ref = 1;
    bit [FIFO_WIDTH-1:0] fifo_q [$];

    // check data
    function void check_data(fifo_transaction f_txn);
        reference_model(f_txn);
        if (data_out_ref === f_txn.data_out)
            correct_count++;
        else begin
            error_count++;
            $display("ERROR: data_out = %h, expected: %h",f_txn.data_out ,data_out_ref);
        end
    endfunction

    // ref model
    function void reference_model(fifo_transaction f_txn);
        if (!f_txn.rst_n) begin
            data_out_ref = 0;
            full_ref     = 0;
            empty_ref    = 1;
            fifo_q.delete();
        end
        else begin
            if (f_txn.wr_en && !full_ref)
                fifo_q.push_back(f_txn.data_in);

            if (f_txn.rd_en && !empty_ref)
                data_out_ref = fifo_q.pop_front();

            empty_ref = fifo_q.size() == 0;
            full_ref  = fifo_q.size() == FIFO_DEPTH;
        end
    endfunction
endclass
endpackage
```

Coverage package



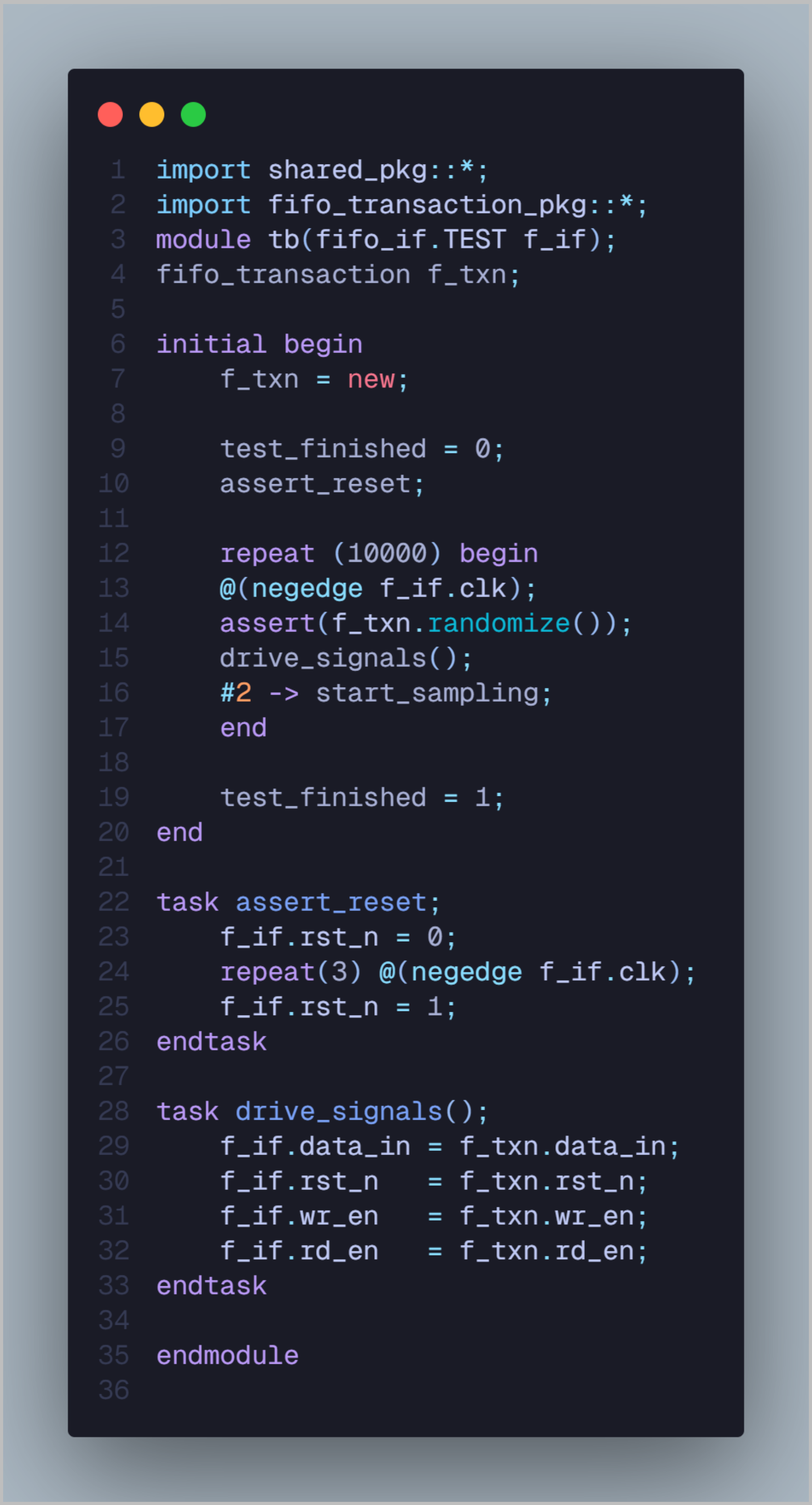
```systemverilog
package fifo_coverage_pkg;
import fifo_transaction_pkg::*;
class fifo_coverage;
    // handel
    fifo_transaction f_cvg_txn;

    ///////////////// covergroup /////////////////
    covergroup cov_gp;
        // coverpoints
        wr_en_cp      : coverpoint f_cvg_txn.wr_en      ;
        rd_en_cp      : coverpoint f_cvg_txn.rd_en      ;
        wr_ack_cp     : coverpoint f_cvg_txn.wr_ack     ;
        overflow_cp   : coverpoint f_cvg_txn.overflow   ;
        full_cp       : coverpoint f_cvg_txn.full       ;
        empty_cp      : coverpoint f_cvg_txn.empty      ;
        almostfull_cp : coverpoint f_cvg_txn.almostfull ;
        almostempty_cp: coverpoint f_cvg_txn.almostempty;
        underflow_cp  : coverpoint f_cvg_txn.underflow  ;

        // cross coverage
        wr_ack_cc     : cross wr_en_cp, rd_en_cp, wr_ack_cp {
            illegal_bins wr_0 = binsof(wr_en_cp) intersect {0} && binsof(wr_ack_cp) intersect {1};
        }
        overflow_cc   : cross wr_en_cp, rd_en_cp, overflow_cp {
            illegal_bins wr_0 = binsof(wr_en_cp) intersect {0} && binsof(overflow_cp) intersect {1};
        }
        full_cc       : cross wr_en_cp, rd_en_cp, full_cp {
            illegal_bins rd_1 = binsof(rd_en_cp) intersect {1} && binsof(full_cp) intersect {1};
        }
        empty_cc      : cross wr_en_cp, rd_en_cp, empty_cp      ;
        almostfull_cc : cross wr_en_cp, rd_en_cp, almostfull_cp ;
        almostempty_cc: cross wr_en_cp, rd_en_cp, almostempty_cp;
        underflow_cc  : cross wr_en_cp, rd_en_cp, underflow_cp {
            illegal_bins rd_0 = binsof(rd_en_cp) intersect {0} && binsof(underflow_cp) intersect {1};
        }
    endgroup: cov_gp

    // sample data function
    function void sample_data(fifo_transaction f_txn);
        f_cvg_txn = f_txn;
        cov_gp.sample;
    endfunction

    // constructor function
    function new();
        cov_gp = new;
    endfunction
endclass
endpackage
```

Monitor



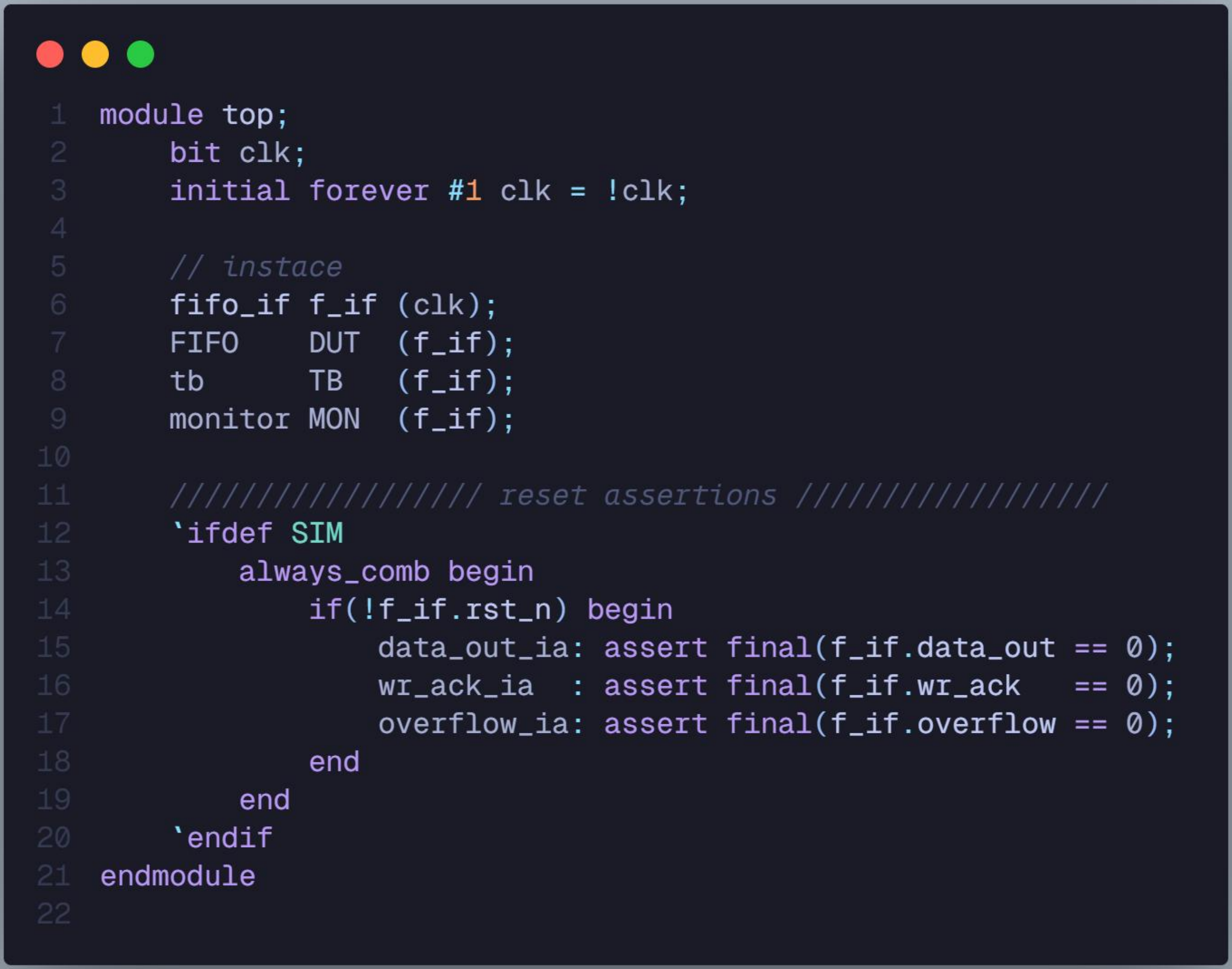
```
import shared_pkg::*;
import fifo_transaction_pkg::*;
import fifo_coverage_pkg::*;
import fifo_scoreboard_pkg::*;
module monitor(fifo_if.MON f_if);
    fifo_transaction f_txn;
    fifo_coverage    f_cov;
    fifo_scoreboard  f_scb;

    initial begin
    f_txn = new;
    f_cov = new;
    f_scb = new;
    forever begin
        @start_sampling;
        // inputs
        f_txn.data_in = f_if.data_in;
        f_txn.rst_n   = f_if.rst_n;
        f_txn.wr_en   = f_if.wr_en;
        f_txn.rd_en   = f_if.rd_en;

        // outputs
        f_txn.data_out    = f_if.data_out;
        f_txn.wr_ack      = f_if.wr_ack;
        f_txn.overflow    = f_if.overflow;
        f_txn.full        = f_if.full;
        f_txn.empty       = f_if.empty;
        f_txn.almostfull  = f_if.almostfull;
        f_txn.almostempty = f_if.almostempty;
        f_txn.underflow   = f_if.underflow;

        fork
            begin
                f_cov.sample_data(f_txn);
            end
            begin
                f_scb.check_data(f_txn);
            end
        join

        if (test_finished) begin
            $display("Correct: m, Error: m", correct_count, error_count);
            $stop;
        end
    end
    end
endmodule
```

Testbench



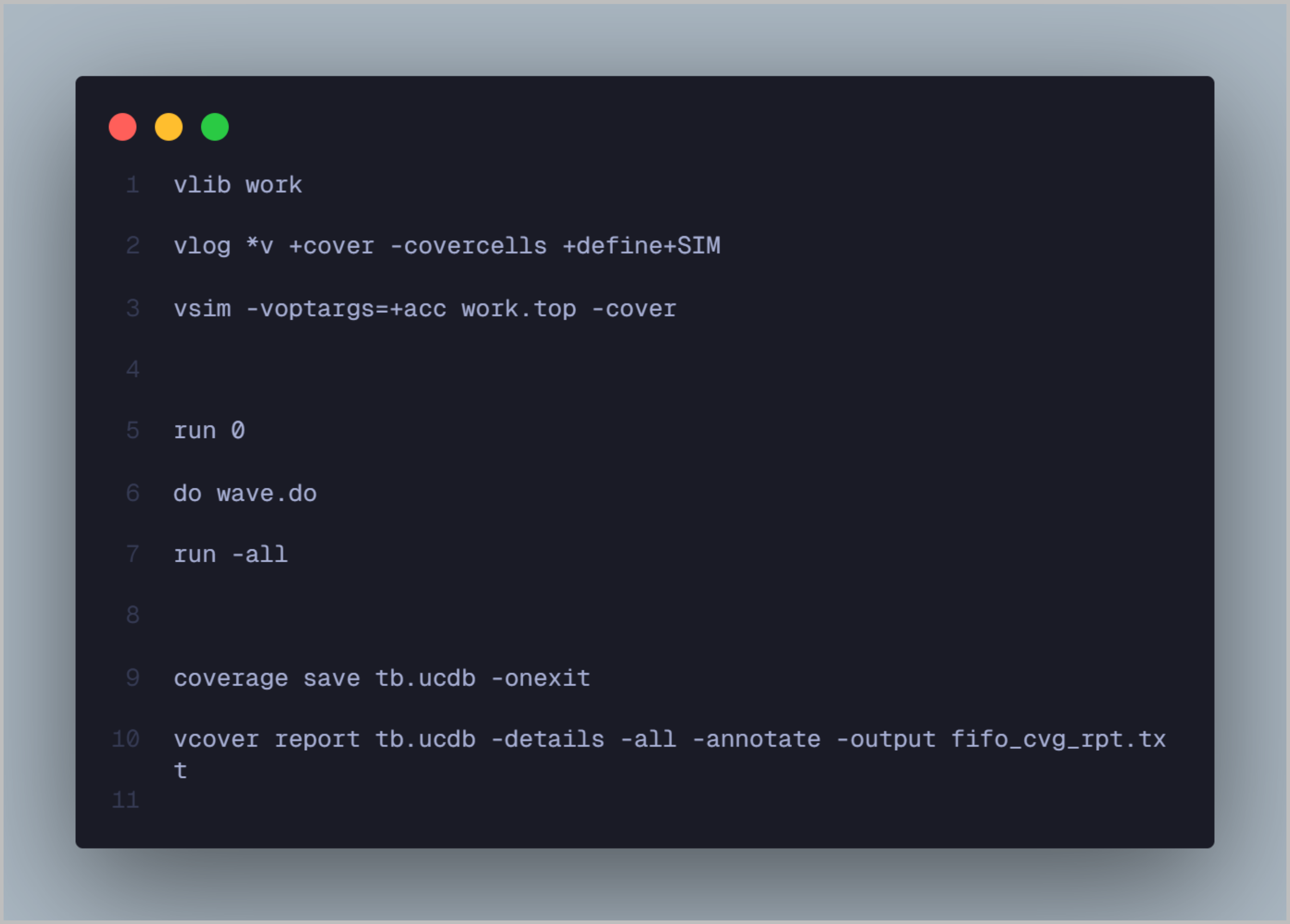
```
import shared_pkg::*;
import fifo_transaction_pkg::*;
module tb(fifo_if.TEST f_if);
fifo_transaction f_txn;

initial begin
    f_txn = new;

    test_finished = 0;
    assert_reset;

    repeat (10000) begin
    @(negedge f_if.clk);
    assert(f_txn.randomize());
    drive_signals();
    #2 -> start_sampling;
    end

    test_finished = 1;
end

task assert_reset;
    f_if.rst_n = 0;
    repeat(3) @(negedge f_if.clk);
    f_if.rst_n = 1;
endtask

task drive_signals();
    f_if.data_in = f_txn.data_in;
    f_if.rst_n   = f_txn.rst_n;
    f_if.wr_en   = f_txn.wr_en;
    f_if.rd_en   = f_txn.rd_en;
endtask

endmodule
```

Top



```
module top;
    bit clk;
    initial forever #1 clk = !clk;

    // instace
    fifo_if f_if (clk);
    FIFO    DUT  (f_if);
    tb      TB   (f_if);
    monitor MON  (f_if);

    ////////////////// reset assertions //////////////////
    `ifdef SIM
        always_comb begin
            if(!f_if.rst_n) begin
                data_out_ia: assert final(f_if.data_out == 0);
                wr_ack_ia  : assert final(f_if.wr_ack   == 0);
                overflow_ia: assert final(f_if.overflow == 0);
            end
        end
    `endif
endmodule
```

Do file



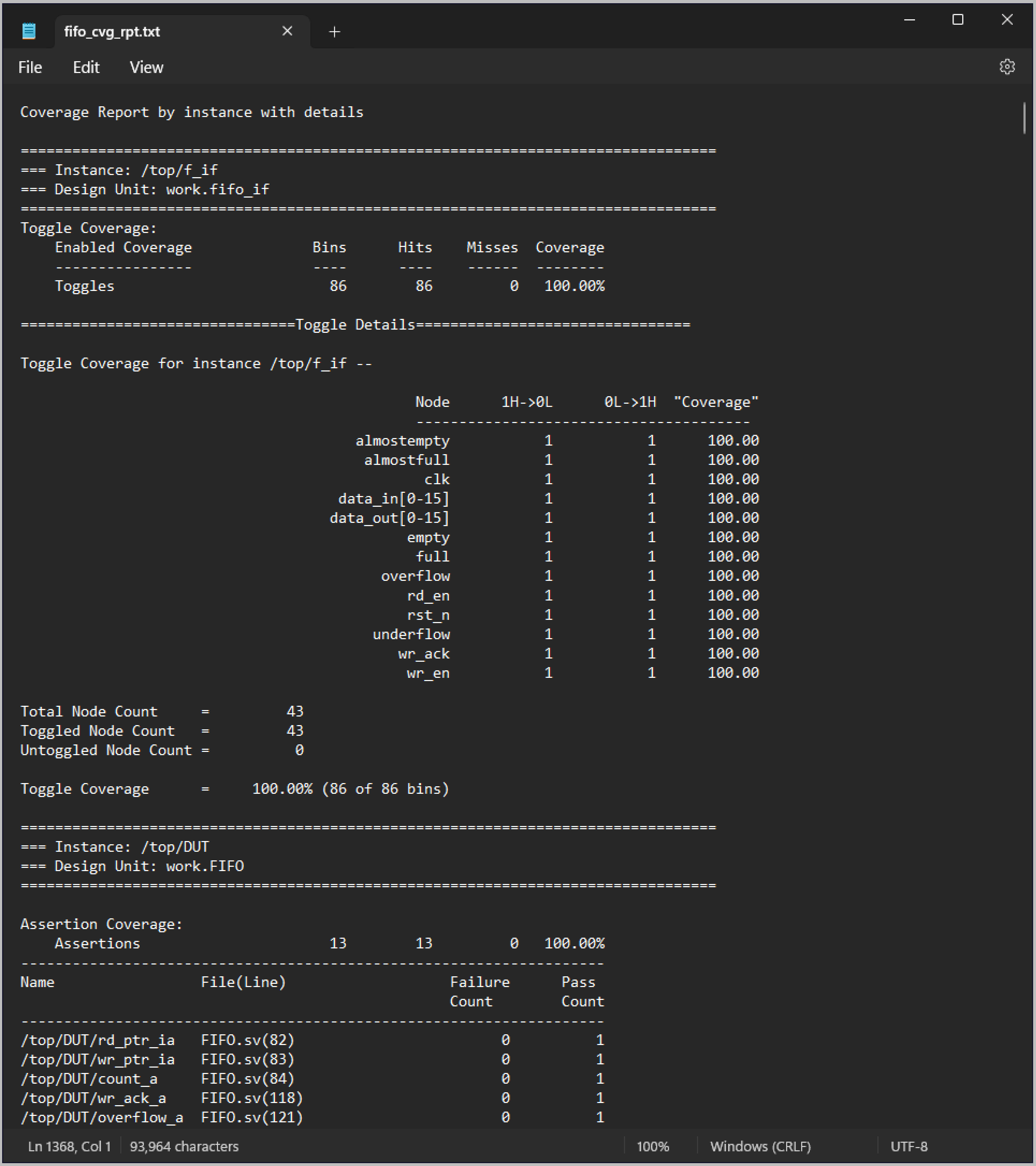
```
vlib work
vlog *v +cover -covercells +define+SIM
vsim -voptargs=+acc work.top -cover

run 0
do wave.do
run -all

coverage save tb.ucdb -onexit
vcover report tb.ucdb -details -all -annotate -output fifo_cvg_rpt.txt
```

Verification plan

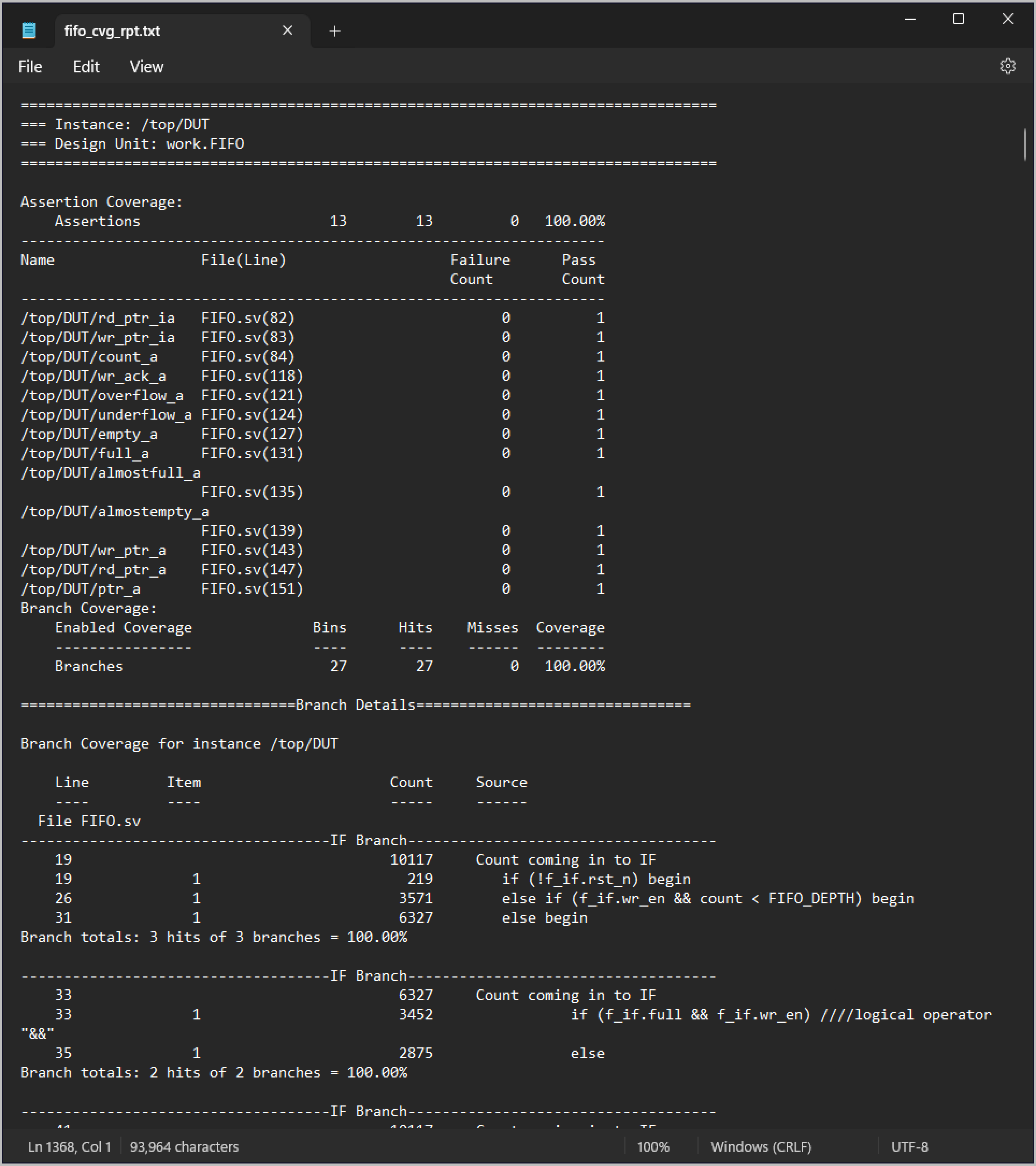
Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
Reset	When the reset is asserted the output value should be zero	Directed at the start of the simulation	-	A checker in the test bench to make sure the output is correct.
Check_data	Verifying FIFO output in case of reading	Randomized during the simulation under the constraints of "rd_en" signal to be on 30% of the time	-	A checker in the scoreboard to make sure the output is correct compared to the golden model.
Reset_ia	When the reset is asserted the data_out, wr_ack, and overflow signals should be zero	Randomized during the simulation under the constraints of reset to be off 99% or the time	-	A checker in the design module to make sure the output is correct. (assertion)
b	Wr_ack signal is high when the writing completed successfully.	Randomized during the simulation under the constraint of the wr_en to be on 70% of the time	Cross coverage with write and read enable	A checker in the design module to make sure the output is correct. (assertion)
c	Overflow signal is high when the writing occurs with fifo memory full.	Randomized during the simulation under the constraint of the wr_en to be on 70% of the time	Cross coverage with write and read enable	A checker in the design module to make sure the output is correct. (assertion)
d	underflow signal is high when the reding occurs with fifo memory empty.	Randomized during the simulation under the constraints of "rd_en" signal to be on 30% of the time	Cross coverage with write and read enable	A checker in the design module to make sure the output is correct. (assertion)
e	empty signal is high when the fifo memory is empty.	Randomized during the simulation	Cross coverage with write and read enable	A checker in the design module to make sure the output is correct. (assertion)
f	full signal is high when the fifo memory is full.	Randomized during the simulation	Cross coverage with write and read enable	A checker in the design module to make sure the output is correct. (assertion)
g	almostfull signal is high when the fifo memory has one empty spot.	Randomized during the simulation	Cross coverage with write and read enable	A checker in the design module to make sure the output is correct. (assertion)

h	almostempty signal is high when the fifo memory has one full spot.	Randomized during the simulation	Cross coverage with write and read enable	A checker in the design module to make sure the output is correct. (assertion)
i	Fifo read and write pointers should wrap from max to zero	Randomized during the simulation	-	A checker in the design module to make sure the output is correct. (assertion)
j	Fifo pointers don’t exceed the max limit.	Randomized during the simulation	-	A checker in the design module to make sure the output is correct. (assertion)
Reset	When the reset is asserted the data_out, wr_ack, and overflow signals should be zero	Randomized during the simulation under the constraints of reset to be off 99% or the time	-	A checker in the top module to make sure the output is correct. (assertion)

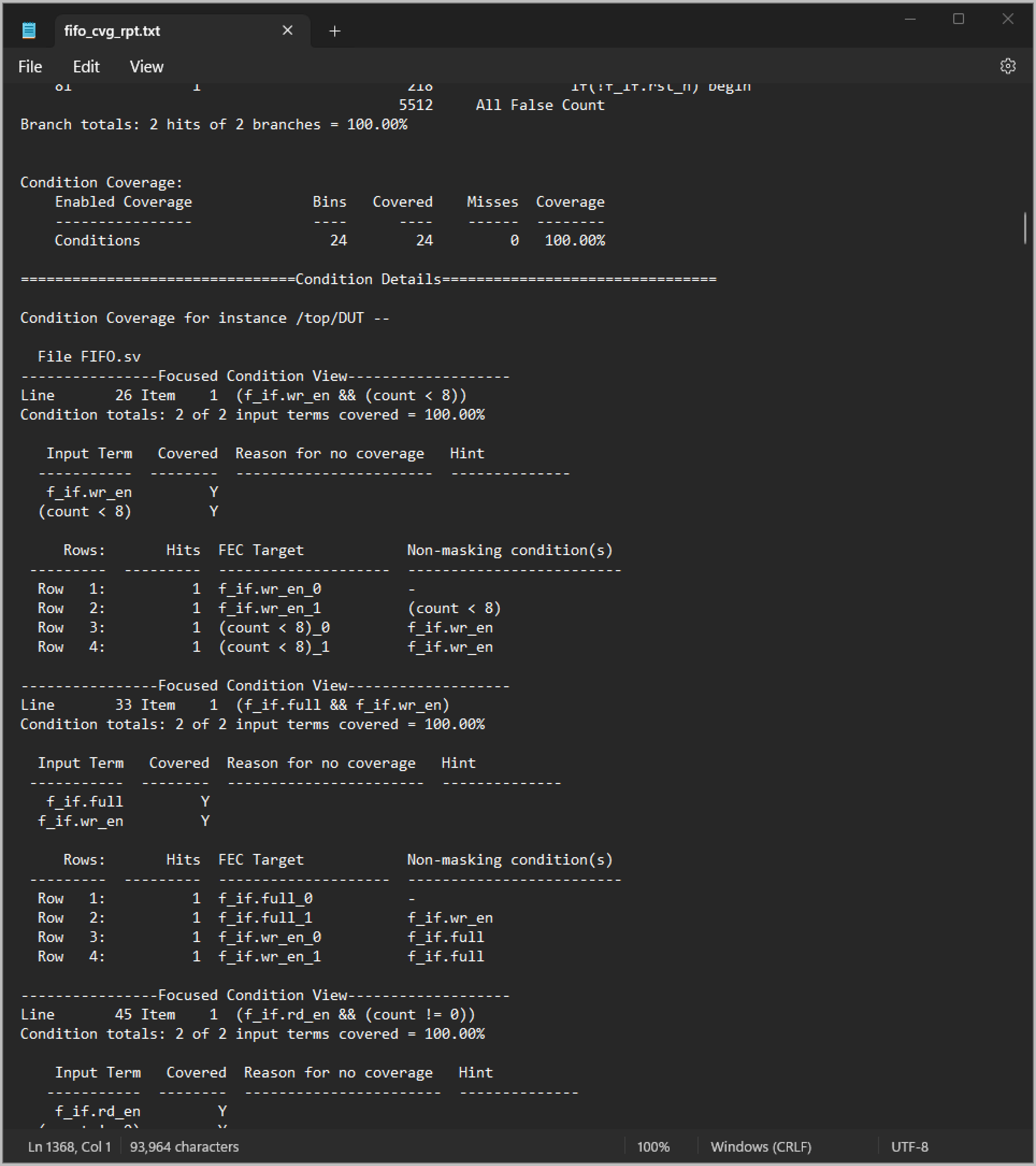
Coverage Report



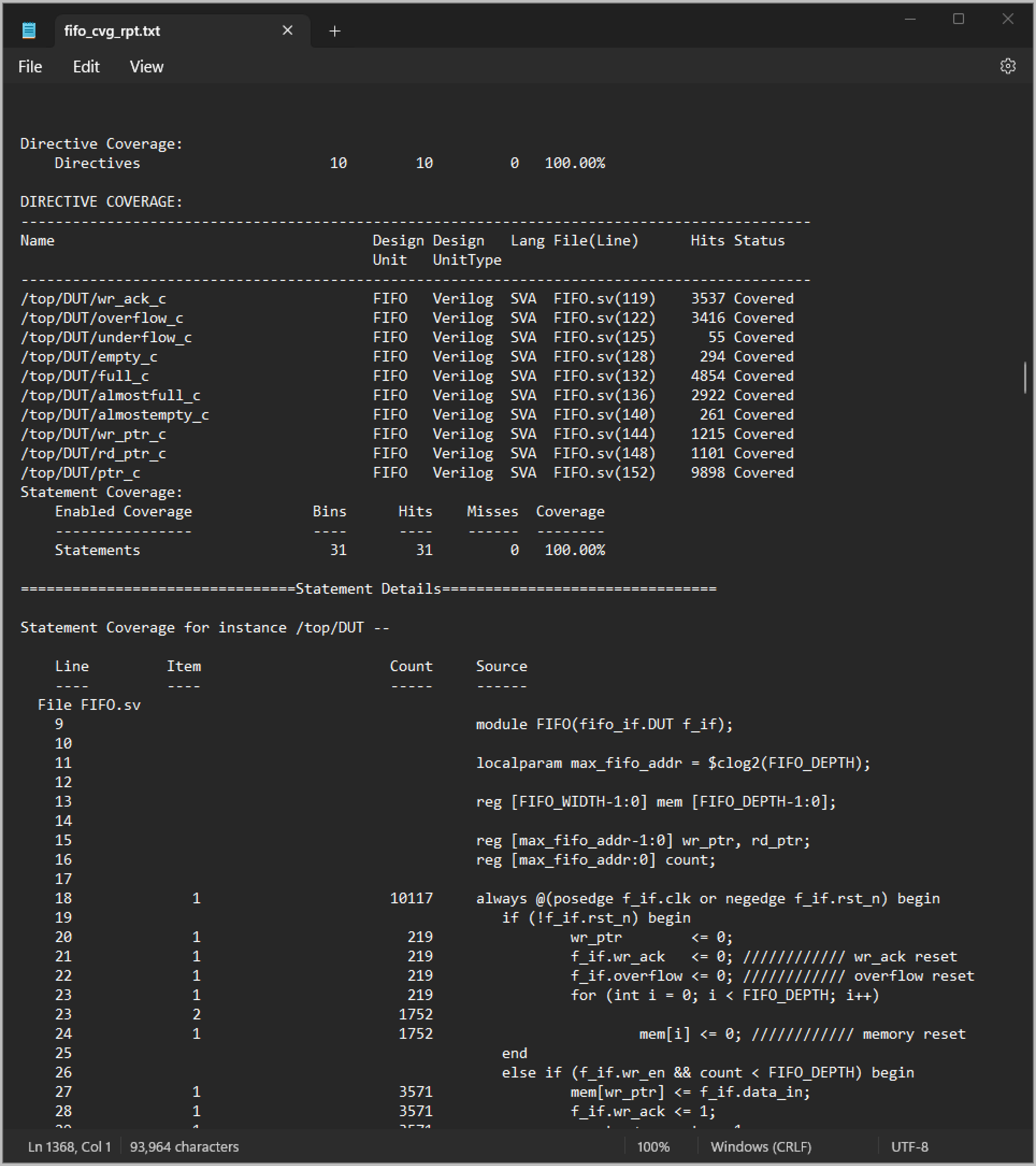
```
Coverage Report by instance with details

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

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

Toggle Coverage for instance /top/f_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)

=================================================================================
=== Instance: /top/DUT
=== Design Unit: work.FIFO
=================================================================================

Assertion Coverage:
    Assertions                      13        13         0   100.00%
--------------------------------------------------------------------
Name                 File(Line)                    Failure      Pass
                                                   Count        Count
--------------------------------------------------------------------
/top/DUT/rd_ptr_ia   FIFO.sv(82)                         0          1
/top/DUT/wr_ptr_ia   FIFO.sv(83)                         0          1
/top/DUT/count_a     FIFO.sv(84)                         0          1
/top/DUT/wr_ack_a    FIFO.sv(118)                        0          1
/top/DUT/overflow_a  FIFO.sv(121)                        0          1
```



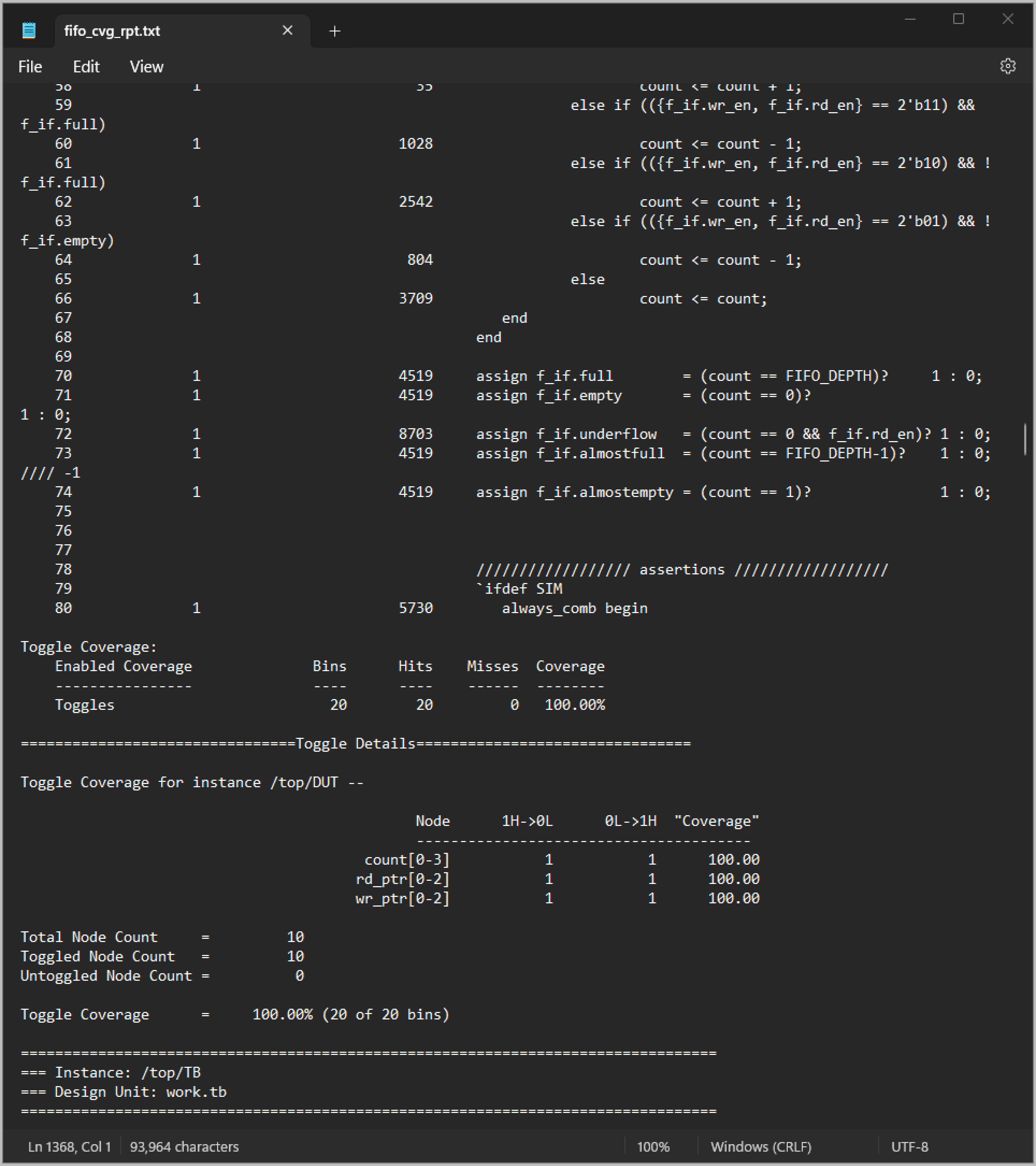
```
=================================================================================
=== Instance: /top/DUT
=== Design Unit: work.FIFO
=================================================================================

Assertion Coverage:
    Assertions                      13        13         0   100.00%
--------------------------------------------------------------------
Name                 File(Line)                   Failure      Pass
                                                  Count        Count
--------------------------------------------------------------------
/top/DUT/rd_ptr_ia   FIFO.sv(82)                        0          1
/top/DUT/wr_ptr_ia   FIFO.sv(83)                        0          1
/top/DUT/count_a     FIFO.sv(84)                        0          1
/top/DUT/wr_ack_a    FIFO.sv(118)                       0          1
/top/DUT/overflow_a  FIFO.sv(121)                       0          1
/top/DUT/underflow_a FIFO.sv(124)                       0          1
/top/DUT/empty_a     FIFO.sv(127)                       0          1
/top/DUT/full_a      FIFO.sv(131)                       0          1
/top/DUT/almostfull_a
                     FIFO.sv(135)                       0          1
/top/DUT/almostempty_a
                     FIFO.sv(139)                       0          1
/top/DUT/wr_ptr_a    FIFO.sv(143)                       0          1
/top/DUT/rd_ptr_a    FIFO.sv(147)                       0          1
/top/DUT/ptr_a       FIFO.sv(151)                       0          1
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        27        27         0   100.00%

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

Branch Coverage for instance /top/DUT

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File FIFO.sv
------------------------------------IF Branch------------------------------------
    19                                     10117     Count coming in to IF
    19              1                        219        if (!f_if.rst_n) begin
    26              1                       3571        else if (f_if.wr_en && count < FIFO_DEPTH) begin
    31              1                       6327        else begin
Branch totals: 3 hits of 3 branches = 100.00%

------------------------------------IF Branch------------------------------------
    33                                      6327     Count coming in to IF
    33              1                       3452             if (f_if.full && f_if.wr_en) ////logical operator
"&&"
    35              1                       2875             else
Branch totals: 2 hits of 2 branches = 100.00%

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



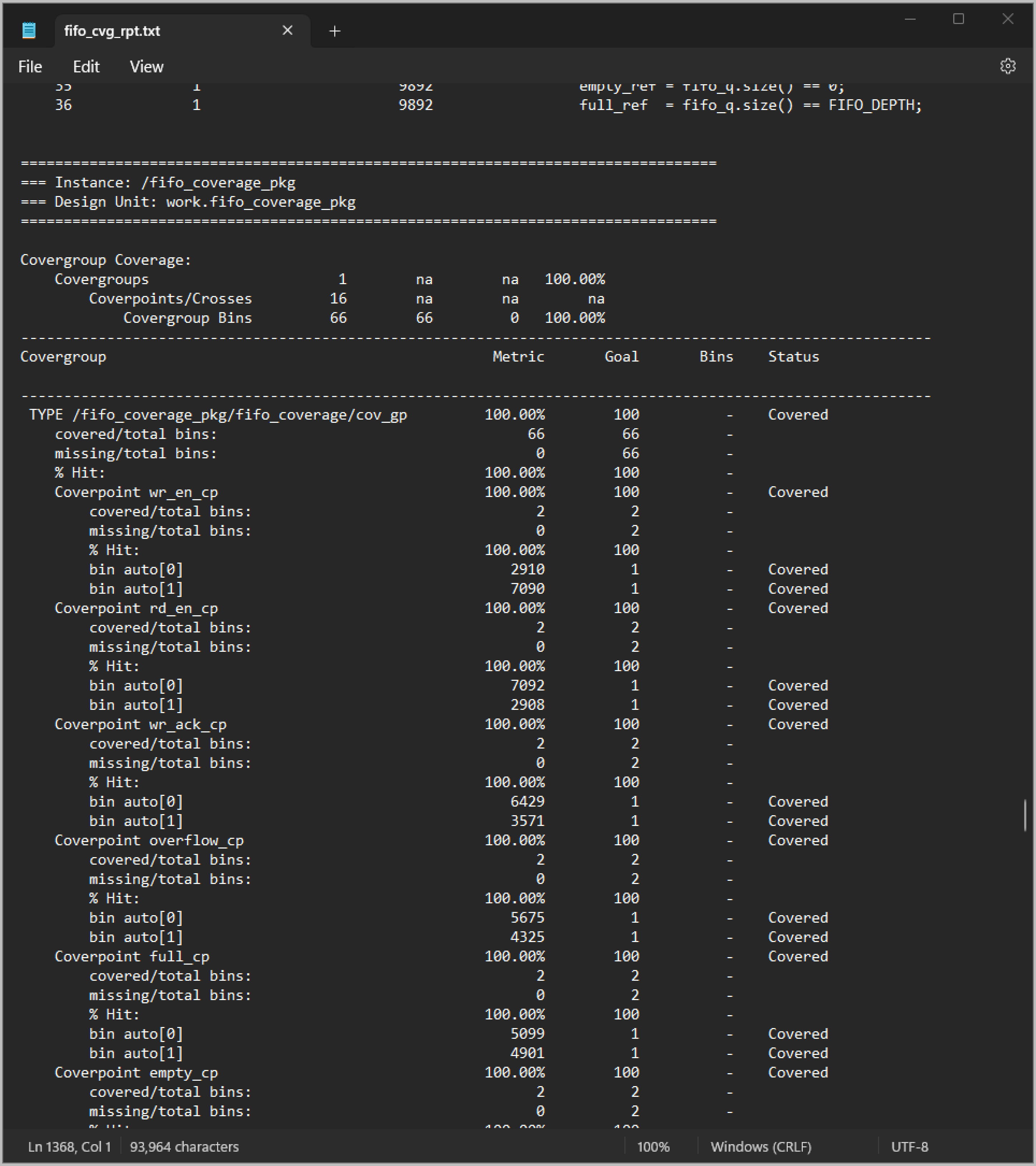
```
    81              1                        218              if(!f_if.rst_n) begin
                                            5512     All False Count
Branch totals: 2 hits of 2 branches = 100.00%


Condition Coverage:
    Enabled Coverage              Bins   Covered    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Conditions                      24        24         0   100.00%

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

Condition Coverage for instance /top/DUT --

  File FIFO.sv
----------------Focused Condition View-------------------
Line       26 Item    1  (f_if.wr_en && (count < 8))
Condition totals: 2 of 2 input terms covered = 100.00%

   Input Term   Covered  Reason for no coverage   Hint
  -----------  --------  -----------------------  --------------
   f_if.wr_en         Y
  (count < 8)         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  f_if.wr_en_0          -
  Row   2:          1  f_if.wr_en_1          (count < 8)
  Row   3:          1  (count < 8)_0         f_if.wr_en
  Row   4:          1  (count < 8)_1         f_if.wr_en

----------------Focused Condition View-------------------
Line       33 Item    1  (f_if.full && f_if.wr_en)
Condition totals: 2 of 2 input terms covered = 100.00%

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

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  f_if.full_0           -
  Row   2:          1  f_if.full_1           f_if.wr_en
  Row   3:          1  f_if.wr_en_0          f_if.full
  Row   4:          1  f_if.wr_en_1          f_if.full

----------------Focused Condition View-------------------
Line       45 Item    1  (f_if.rd_en && (count != 0))
Condition totals: 2 of 2 input terms covered = 100.00%

    Input Term   Covered  Reason for no coverage   Hint
   -----------  --------  -----------------------  --------------
    f_if.rd_en         Y
```



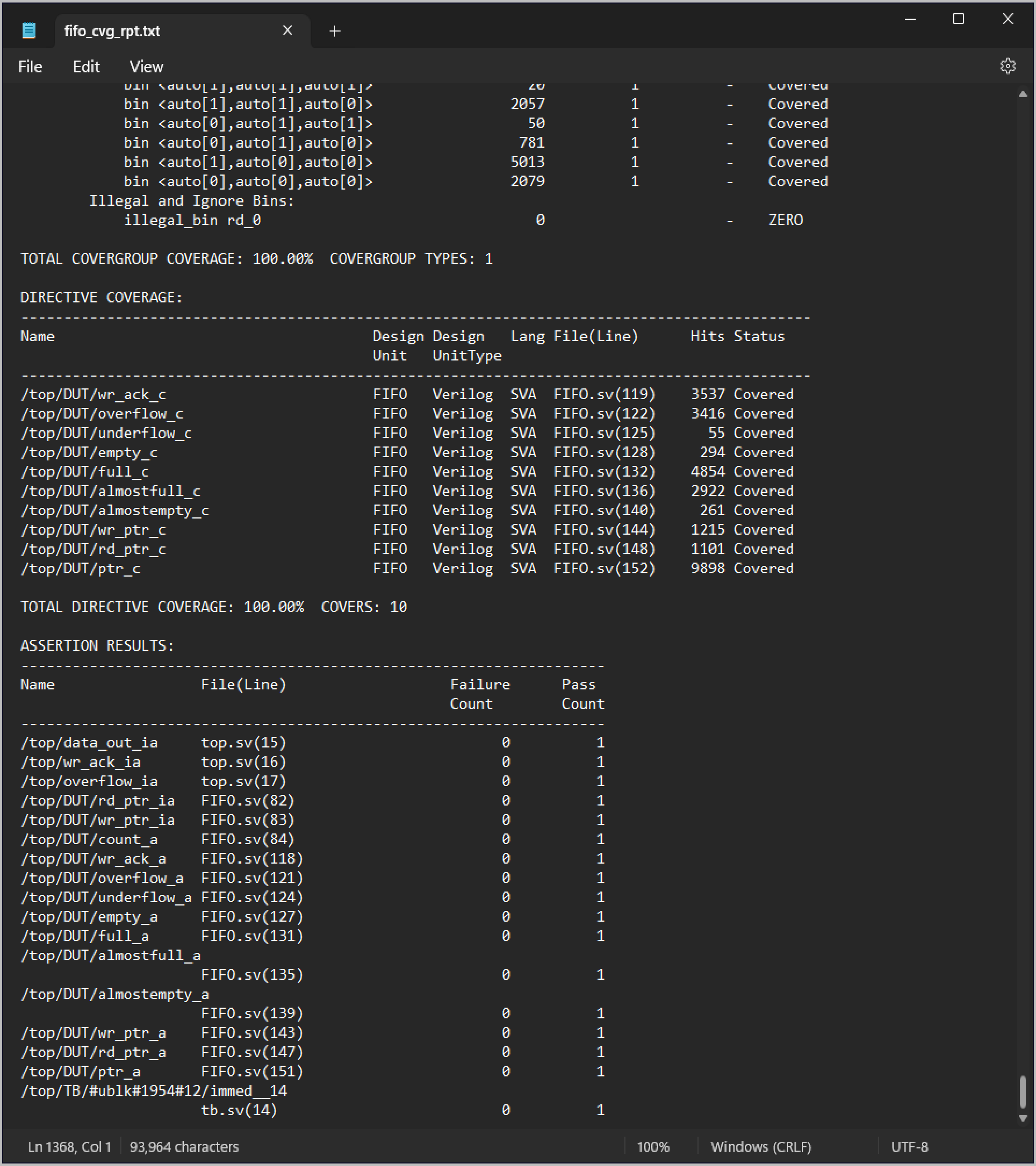
```
Directive Coverage:
    Directives                    10        10         0   100.00%

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                     Design Design   Lang File(Line)      Hits Status
                                         Unit   UnitType
--------------------------------------------------------------------------------------------
/top/DUT/wr_ack_c                        FIFO   Verilog  SVA  FIFO.sv(119)    3537 Covered
/top/DUT/overflow_c                      FIFO   Verilog  SVA  FIFO.sv(122)    3416 Covered
/top/DUT/underflow_c                     FIFO   Verilog  SVA  FIFO.sv(125)      55 Covered
/top/DUT/empty_c                         FIFO   Verilog  SVA  FIFO.sv(128)     294 Covered
/top/DUT/full_c                          FIFO   Verilog  SVA  FIFO.sv(132)    4854 Covered
/top/DUT/almostfull_c                    FIFO   Verilog  SVA  FIFO.sv(136)    2922 Covered
/top/DUT/almostempty_c                   FIFO   Verilog  SVA  FIFO.sv(140)     261 Covered
/top/DUT/wr_ptr_c                        FIFO   Verilog  SVA  FIFO.sv(144)    1215 Covered
/top/DUT/rd_ptr_c                        FIFO   Verilog  SVA  FIFO.sv(148)    1101 Covered
/top/DUT/ptr_c                           FIFO   Verilog  SVA  FIFO.sv(152)    9898 Covered
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                      31        31         0   100.00%

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

Statement Coverage for instance /top/DUT --

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File FIFO.sv
    9                                                module FIFO(fifo_if.DUT f_if);
    10
    11                                               localparam max_fifo_addr = $clog2(FIFO_DEPTH);
    12
    13                                               reg [FIFO_WIDTH-1:0] mem [FIFO_DEPTH-1:0];
    14
    15                                               reg [max_fifo_addr-1:0] wr_ptr, rd_ptr;
    16                                               reg [max_fifo_addr:0] count;
    17
    18              1                      10117     always @(posedge f_if.clk or negedge f_if.rst_n) begin
    19                                                  if (!f_if.rst_n) begin
    20              1                        219              wr_ptr        <= 0;
    21              1                        219              f_if.wr_ack   <= 0; //////////// wr_ack reset
    22              1                        219              f_if.overflow <= 0; //////////// overflow reset
    23              1                        219              for (int i = 0; i < FIFO_DEPTH; i++)
    23              2                       1752
    24              1                       1752                      mem[i] <= 0; //////////// memory reset
    25                                                  end
    26                                                  else if (f_if.wr_en && count < FIFO_DEPTH) begin
    27              1                       3571              mem[wr_ptr] <= f_if.data_in;
    28              1                       3571              f_if.wr_ack <= 1;
```



```
    58              1                       55                  count <= count + 1;
    59                                                  else if (({f_if.wr_en, f_if.rd_en} == 2'b11) &&
f_if.full)
    60              1                     1028                  count <= count - 1;
    61                                                  else if (({f_if.wr_en, f_if.rd_en} == 2'b10) && !
f_if.full)
    62              1                     2542                  count <= count + 1;
    63                                                  else if (({f_if.wr_en, f_if.rd_en} == 2'b01) && !
f_if.empty)
    64              1                      804                  count <= count - 1;
    65                                                  else
    66              1                     3709                  count <= count;
    67                                              end
    68                                           end
    69
    70              1                     4519   assign f_if.full        = (count == FIFO_DEPTH)?     1 : 0;
    71              1                     4519   assign f_if.empty       = (count == 0)?
1 : 0;
    72              1                     8703   assign f_if.underflow   = (count == 0 && f_if.rd_en)? 1 : 0;
    73              1                     4519   assign f_if.almostfull  = (count == FIFO_DEPTH-1)?    1 : 0;
//// -1
    74              1                     4519   assign f_if.almostempty = (count == 1)?               1 : 0;
    75
    76
    77
    78                                           ///////////////// assertions /////////////////
    79                                           `ifdef SIM
    80              1                     5730      always_comb begin

Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         20        20         0   100.00%

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

Toggle Coverage for instance /top/DUT --

                                              Node      1H->0L      0L->1H  "Coverage"
                                              ---------------------------------------
                                        count[0-3]           1           1      100.00
                                       rd_ptr[0-2]           1           1      100.00
                                       wr_ptr[0-2]           1           1      100.00

Total Node Count     =         10
Toggled Node Count   =         10
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (20 of 20 bins)

=================================================================================
=== Instance: /top/TB
=== Design Unit: work.tb
=================================================================================
```

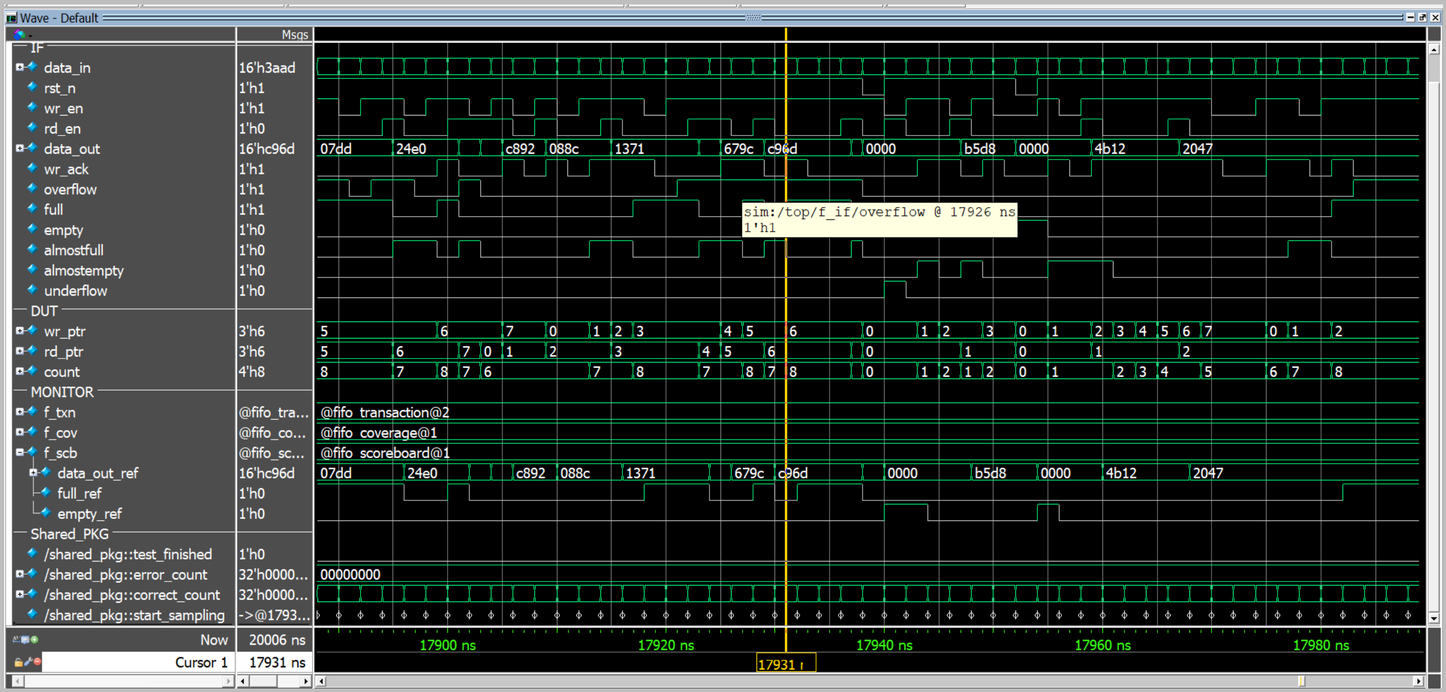


```
    35           1                  9892                empty_ref = fifo_q.size() == 0;
    36           1                  9892                full_ref  = fifo_q.size() == FIFO_DEPTH;


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

Covergroup Coverage:
    Covergroups                      1        na        na   100.00%
        Coverpoints/Crosses         16        na        na        na
            Covergroup Bins         66        66         0   100.00%
----------------------------------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status

----------------------------------------------------------------------------------------------------------
 TYPE /fifo_coverage_pkg/fifo_coverage/cov_gp         100.00%        100          -    Covered
    covered/total bins:                                    66         66          -
    missing/total bins:                                     0         66          -
    % Hit:                                            100.00%        100          -
    Coverpoint wr_en_cp                               100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      2910          1          -    Covered
        bin auto[1]                                      7090          1          -    Covered
    Coverpoint rd_en_cp                               100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      7092          1          -    Covered
        bin auto[1]                                      2908          1          -    Covered
    Coverpoint wr_ack_cp                              100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      6429          1          -    Covered
        bin auto[1]                                      3571          1          -    Covered
    Coverpoint overflow_cp                            100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      5675          1          -    Covered
        bin auto[1]                                      4325          1          -    Covered
    Coverpoint full_cp                                100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100          -
        bin auto[0]                                      5099          1          -    Covered
        bin auto[1]                                      4901          1          -    Covered
    Coverpoint empty_cp                               100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                        100.00%        100
```

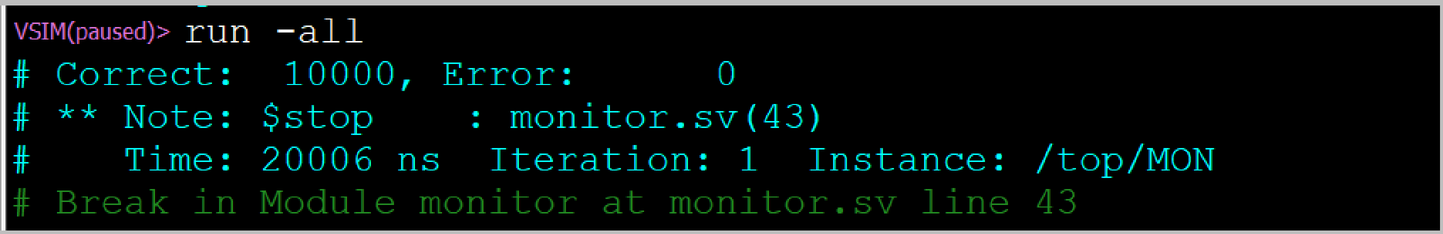


```
            bin <auto[1],auto[1],auto[1]>                  20          1          -    Covered
            bin <auto[1],auto[1],auto[0]>                2057          1          -    Covered
            bin <auto[0],auto[1],auto[1]>                  50          1          -    Covered
            bin <auto[0],auto[1],auto[0]>                 781          1          -    Covered
            bin <auto[1],auto[0],auto[0]>                5013          1          -    Covered
            bin <auto[0],auto[0],auto[0]>                2079          1          -    Covered
        Illegal and Ignore Bins:
            illegal_bin rd_0                                0                     -    ZERO

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                     Design Design   Lang File(Line)      Hits Status
                                         Unit   UnitType
--------------------------------------------------------------------------------------------
/top/DUT/wr_ack_c                        FIFO   Verilog  SVA  FIFO.sv(119)    3537 Covered
/top/DUT/overflow_c                      FIFO   Verilog  SVA  FIFO.sv(122)    3416 Covered
/top/DUT/underflow_c                     FIFO   Verilog  SVA  FIFO.sv(125)      55 Covered
/top/DUT/empty_c                         FIFO   Verilog  SVA  FIFO.sv(128)     294 Covered
/top/DUT/full_c                          FIFO   Verilog  SVA  FIFO.sv(132)    4854 Covered
/top/DUT/almostfull_c                    FIFO   Verilog  SVA  FIFO.sv(136)    2922 Covered
/top/DUT/almostempty_c                   FIFO   Verilog  SVA  FIFO.sv(140)     261 Covered
/top/DUT/wr_ptr_c                        FIFO   Verilog  SVA  FIFO.sv(144)    1215 Covered
/top/DUT/rd_ptr_c                        FIFO   Verilog  SVA  FIFO.sv(148)    1101 Covered
/top/DUT/ptr_c                           FIFO   Verilog  SVA  FIFO.sv(152)    9898 Covered

TOTAL DIRECTIVE COVERAGE: 100.00%  COVERS: 10

ASSERTION RESULTS:
--------------------------------------------------------------------
Name                 File(Line)                   Failure      Pass
                                                  Count        Count
--------------------------------------------------------------------
/top/data_out_ia     top.sv(15)                         0          1
/top/wr_ack_ia       top.sv(16)                         0          1
/top/overflow_ia     top.sv(17)                         0          1
/top/DUT/rd_ptr_ia   FIFO.sv(82)                        0          1
/top/DUT/wr_ptr_ia   FIFO.sv(83)                        0          1
/top/DUT/count_a     FIFO.sv(84)                        0          1
/top/DUT/wr_ack_a    FIFO.sv(118)                       0          1
/top/DUT/overflow_a  FIFO.sv(121)                       0          1
/top/DUT/underflow_a FIFO.sv(124)                       0          1
/top/DUT/empty_a     FIFO.sv(127)                       0          1
/top/DUT/full_a      FIFO.sv(131)                       0          1
/top/DUT/almostfull_a
                     FIFO.sv(135)                       0          1
/top/DUT/almostempty_a
                     FIFO.sv(139)                       0          1
/top/DUT/wr_ptr_a    FIFO.sv(143)                       0          1
/top/DUT/rd_ptr_a    FIFO.sv(147)                       0          1
/top/DUT/ptr_a       FIFO.sv(151)                       0          1
/top/TB/#ublk#1954#12/immed__14
                     tb.sv(14)                          0          1
```

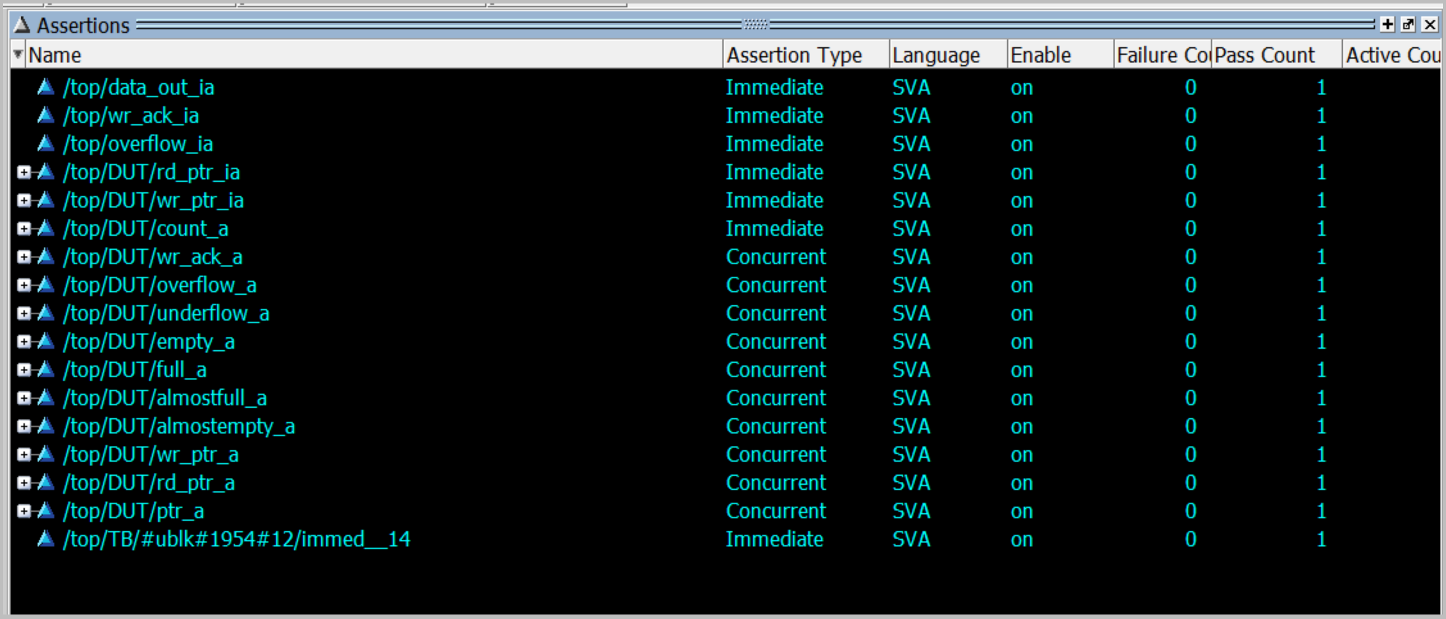
QuestaSim result



Wave - Default	Msgs
IF	
data_in	16'h3aad
rst_n	1'h1
wr_en	1'h1
rd_en	1'h0
data_out	16'hc96d
wr_ack	1'h1
overflow	1'h1
full	1'h1
empty	1'h0
almostfull	1'h0
almostempty	1'h0
underflow	1'h0
DUT	
wr_ptr	3'h6
rd_ptr	3'h6
count	4'h8
MONITOR	
f_txn	@fifo_tra...
f_cov	@fifo_co...
f_scb	@fifo_sc...
data_out_ref	16'hc96d
full_ref	1'h0
empty_ref	1'h0
Shared_PKG	
/shared_pkg::test_finished	1'h0
/shared_pkg::error_count	32'h0000...
/shared_pkg::correct_count	32'h0000...
/shared_pkg::start_sampling	->@1793...
Now	20006 ns
Cursor 1	17931 ns

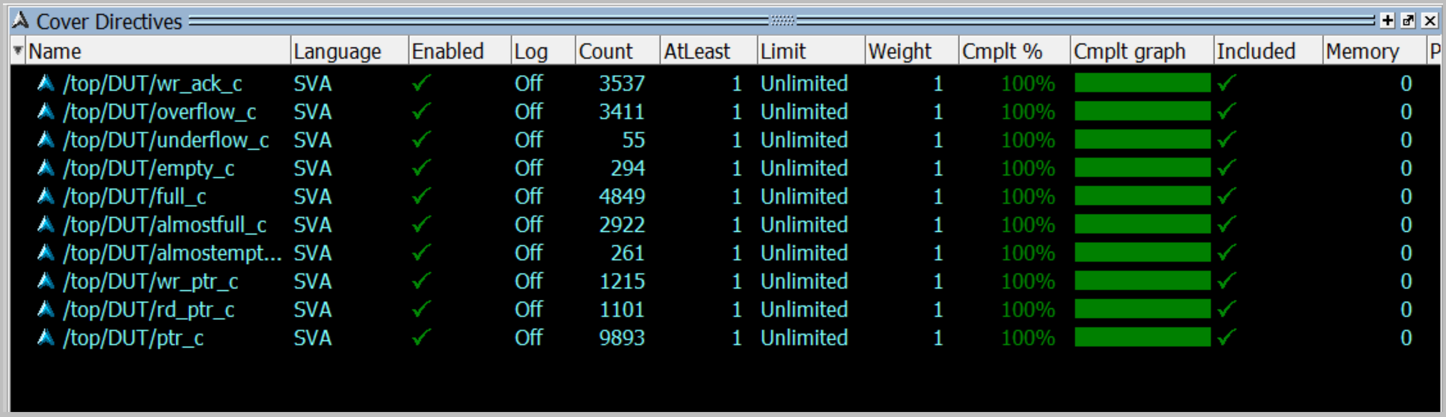


```
VSIM(paused)> run -all
# Correct:  10000, Error:       0
# ** Note: $stop    : monitor.sv(43)
#    Time: 20006 ns  Iteration: 1  Instance: /top/MON
# Break in Module monitor at monitor.sv line 43
```



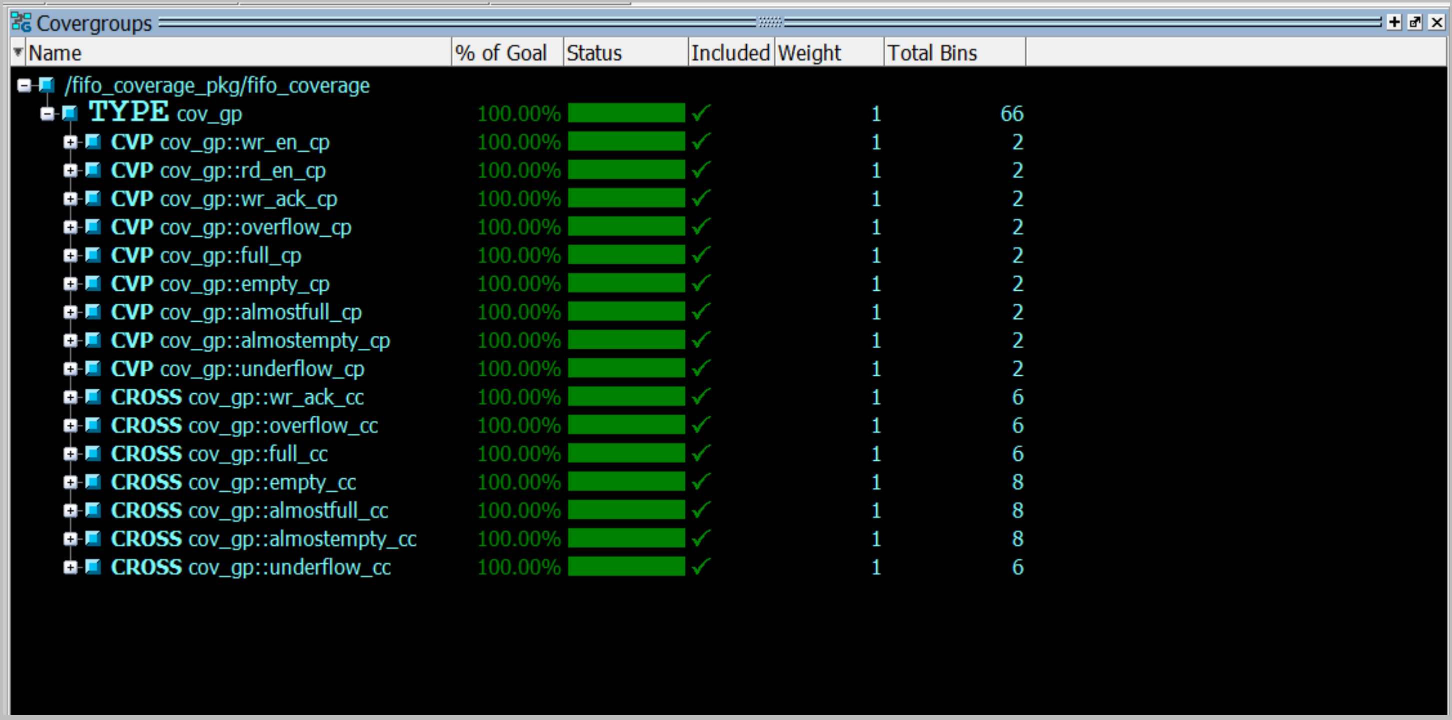
Assertions

Name	Assertion Type	Language	Enable	Failure Co	Pass Count	Active Cou
/top/data_out_ia	Immediate	SVA	on	0	1	
/top/wr_ack_ia	Immediate	SVA	on	0	1	
/top/overflow_ia	Immediate	SVA	on	0	1	
/top/DUT/rd_ptr_ia	Immediate	SVA	on	0	1	
/top/DUT/wr_ptr_ia	Immediate	SVA	on	0	1	
/top/DUT/count_a	Immediate	SVA	on	0	1	
/top/DUT/wr_ack_a	Concurrent	SVA	on	0	1	
/top/DUT/overflow_a	Concurrent	SVA	on	0	1	
/top/DUT/underflow_a	Concurrent	SVA	on	0	1	
/top/DUT/empty_a	Concurrent	SVA	on	0	1	
/top/DUT/full_a	Concurrent	SVA	on	0	1	
/top/DUT/almostfull_a	Concurrent	SVA	on	0	1	
/top/DUT/almostempty_a	Concurrent	SVA	on	0	1	
/top/DUT/wr_ptr_a	Concurrent	SVA	on	0	1	
/top/DUT/rd_ptr_a	Concurrent	SVA	on	0	1	
/top/DUT/ptr_a	Concurrent	SVA	on	0	1	
/top/TB/#ublk#1954#12/immed__14	Immediate	SVA	on	0	1	



Cover Directives

Name	Language	Enabled	Log	Count	AtLeast	Limit	Weight	Cmplt %	Cmplt graph	Included	Memory	P
/top/DUT/wr_ack_c	SVA	✓	Off	3537	1	Unlimited	1	100%	🟩	✓	0	
/top/DUT/overflow_c	SVA	✓	Off	3411	1	Unlimited	1	100%	🟩	✓	0	
/top/DUT/underflow_c	SVA	✓	Off	55	1	Unlimited	1	100%	🟩	✓	0	
/top/DUT/empty_c	SVA	✓	Off	294	1	Unlimited	1	100%	🟩	✓	0	
/top/DUT/full_c	SVA	✓	Off	4849	1	Unlimited	1	100%	🟩	✓	0	
/top/DUT/almostfull_c	SVA	✓	Off	2922	1	Unlimited	1	100%	🟩	✓	0	
/top/DUT/almostempt...	SVA	✓	Off	261	1	Unlimited	1	100%	🟩	✓	0	
/top/DUT/wr_ptr_c	SVA	✓	Off	1215	1	Unlimited	1	100%	🟩	✓	0	
/top/DUT/rd_ptr_c	SVA	✓	Off	1101	1	Unlimited	1	100%	🟩	✓	0	
/top/DUT/ptr_c	SVA	✓	Off	9893	1	Unlimited	1	100%	🟩	✓	0	



Covergroups

Name	% of Goal	Status	Included	Weight	Total Bins
/fifo_coverage_pkg/fifo_coverage					
TYPE cov_gp	100.00%	🟩	✓	1	66
CVP cov_gp::wr_en_cp	100.00%	🟩	✓	1	2
CVP cov_gp::rd_en_cp	100.00%	🟩	✓	1	2
CVP cov_gp::wr_ack_cp	100.00%	🟩	✓	1	2
CVP cov_gp::overflow_cp	100.00%	🟩	✓	1	2
CVP cov_gp::full_cp	100.00%	🟩	✓	1	2
CVP cov_gp::empty_cp	100.00%	🟩	✓	1	2
CVP cov_gp::almostfull_cp	100.00%	🟩	✓	1	2
CVP cov_gp::almostempty_cp	100.00%	🟩	✓	1	2
CVP cov_gp::underflow_cp	100.00%	🟩	✓	1	2
CROSS cov_gp::wr_ack_cc	100.00%	🟩	✓	1	6
CROSS cov_gp::overflow_cc	100.00%	🟩	✓	1	6
CROSS cov_gp::full_cc	100.00%	🟩	✓	1	6
CROSS cov_gp::empty_cc	100.00%	🟩	✓	1	8
CROSS cov_gp::almostfull_cc	100.00%	🟩	✓	1	8
CROSS cov_gp::almostempty_cc	100.00%	🟩	✓	1	8
CROSS cov_gp::underflow_cc	100.00%	🟩	✓	1	6

Thank you