Design

Errors

1- Almost full flag is raised one cycle before the FIFO is full by 2 elements not 1.
2- Not all cases of updating count according to wr_en and rd_en are covered (the cases where wr_en and rd_en are both high).
3- Reset must reset all flags, counter, and pointers.
4- Removed the redundant full from the if condition updating the overflow to get 100% conditional coverage.

Corrected Design Code With Assertions

```
////////////////////////////////////////////////////////////////////////////////
// Author: Kareem Waseem
// Course: Digital Verification using SV & UVM
//
// Description: FIFO Design
//
////////////////////////////////////////////////////////////////////////////////
module FIFO(IFIFO.DUT fifo_if);
    localparam max_fifo_addr = $clog2(fifo_if.FIFO_DEPTH);

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

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

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

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

```
            fifo_if.underflow <= 1;
        end else begin
            fifo_if.underflow <= 0;
        end
    end

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

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

    `ifdef SIM
        property IsFullEmpty;
            @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) !(fifo_if.full && fifo_if.empty);
        endproperty

        property IsEmpty;
            @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count == 0 |-> fifo_if.empty == 1;
        endproperty

        property IsNotEmpty;
            @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count != 0 |-> fifo_if.empty == 0;
        endproperty

        property IsAlmostEmpty;
            @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count == 1 |-> fifo_if.almostempty == 1;
        endproperty

        property IsFull;
            @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count == fifo_if.FIFO_DEPTH |-> fifo_if.full == 1;
        endproperty

        property IsNotFull;
            @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count != fifo_if.FIFO_DEPTH |-> fifo_if.full == 0;
        endproperty

        property IsAlmostFull;
            @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count == fifo_if.FIFO_DEPTH - 1 |-> fifo_if.almostfull == 1;
        endproperty

        property IsOverflow;
```

```systemverilog
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) (fifo_if.full && fifo_if.wr_en) |=> fifo_if.overflow == 1;
    endproperty

    property IsUnderflow;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) (fifo_if.empty && fifo_if.rd_en) |=> fifo_if.underflow == 1;
    endproperty

    property IsWriteAck;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) (!fifo_if.full && fifo_if.wr_en) |=> fifo_if.wr_ack == 1;
    endproperty

    property wr_ptr_wrap;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n)
            (!fifo_if.full && fifo_if.wr_en && wr_ptr == fifo_if.FIFO_DEPTH - 1) |=> wr_ptr == 0;
    endproperty

    property rd_ptr_wrap;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n)
            (!fifo_if.empty && fifo_if.rd_en && rd_ptr == fifo_if.FIFO_DEPTH - 1) |=> rd_ptr == 0;
    endproperty

    always_comb begin
        if (!fifo_if.rst_n) begin
            a_reset_empty: assert final(fifo_if.empty == 1);
        end
    end

    always_comb begin
        if (!fifo_if.rst_n) begin
            a_reset_full: assert final(fifo_if.full == 0);
        end
    end

    always_comb begin
        if (!fifo_if.rst_n) begin
            a_reset_almostempty: assert final(fifo_if.almostempty == 0);
        end
    end

    always_comb begin
        if (!fifo_if.rst_n) begin
            a_reset_almostfull: assert final(fifo_if.almostfull == 0);
        end
    end

    always_comb begin
        if (!fifo_if.rst_n) begin
            a_reset_overflow: assert final(fifo_if.overflow == 0);
        end
    end

    always_comb begin
        if (!fifo_if.rst_n) begin
            a_reset_underflow: assert final(fifo_if.underflow == 0);
        end
    end

    always_comb begin
        if (!fifo_if.rst_n) begin
            a_reset_wr_ack: assert final(fifo_if.wr_ack == 0);
        end
    end

    always_comb begin
        if (!fifo_if.rst_n) begin
            a_reset_wr_ptr: assert final(wr_ptr == 0);
        end
    end

    always_comb begin
        if (!fifo_if.rst_n) begin
            a_reset_rd_ptr: assert final(rd_ptr == 0);
        end
    end

    always_comb begin
        if (!fifo_if.rst_n) begin
            a_reset_count: assert final(count == 0);
        end
    end

    always_comb begin
        a_wr_ptr_threshold: assert final(wr_ptr < fifo_if.FIFO_DEPTH);
    end

    always_comb begin
        a_rd_ptr_threshold: assert final(rd_ptr < fifo_if.FIFO_DEPTH);
    end

    always_comb begin
        a_count_threshold: assert final(count <= fifo_if.FIFO_DEPTH);
    end

    a_IsFullEmpty: assert property (IsFullEmpty);
    c_IsFullEmpty: cover property (IsFullEmpty);

    a_IsEmpty: assert property (IsEmpty);
    c_IsEmpty: cover property (IsEmpty);

    a_IsNotEmpty: assert property (IsNotEmpty);
    c_IsNotEmpty: cover property (IsNotEmpty);

    a_IsAlmostEmpty: assert property (IsAlmostEmpty);
    c_IsAlmostEmpty: cover property (IsAlmostEmpty);

    a_IsFull: assert property (IsFull);
    c_IsFull: cover property (IsFull);

    a_IsNotFull: assert property (IsNotFull);
    c_IsNotFull: cover property (IsNotFull);

    a_IsAlmostFull: assert property (IsAlmostFull);
    c_IsAlmostFull: cover property (IsAlmostFull);

    a_IsOverflow: assert property (IsOverflow);
    c_IsOverflow: cover property (IsOverflow);

    a_IsUnderflow: assert property (IsUnderflow);
    c_IsUnderflow: cover property (IsUnderflow);

    a_IsWriteAck: assert property (IsWriteAck);
    c_IsWriteAck: cover property (IsWriteAck);

    a_wr_ptr_wrap: assert property (wr_ptr_wrap);
    c_wr_ptr_wrap: cover property (wr_ptr_wrap);

    a_rd_ptr_wrap: assert property (rd_ptr_wrap);
    c_rd_ptr_wrap: cover property (rd_ptr_wrap);
`endif
endmodule

```

Interface

```
interface IFIFO(clk);
    parameter FIFO_WIDTH = 16;
    parameter FIFO_DEPTH = 8;

    input bit 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;

    event start_to_sample;

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

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

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

```

Shared Package

```
package shared_pkg;
    bit test_finished;
    int error_count, correct_count;
endpackage

```

Verification Plan

Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
FIFO_1	When the reset is asserted, the output empty value should be high, all other flags should be low	Directed at the start of the simulation, and randomized with 5% to be asserted in the simulation time	-	Immediate assertion to check functionality
FIFO_2	When there is no elements in the fifo the empty output value should be high	Randomization	make sure that all the combinations of wr_en, wr_en and empty occurred	Concurrent assertion to check functionality
FIFO_3	When there is only one element in the fifo, almost empty value should be high	Randomization	make sure that all the combinations of wr_en, wr_en and almost empty occurred	Concurrent assertion to check functionality
FIFO_4	When there is a full fifo, the output full value should be high	Randomization	make sure that all the combinations of wr_en, wr_en and full occurred	Concurrent assertion to check functionality
FIFO_5	When the fifo is one element far to be full, the output almost full value should be high	Randomization	make sure that all the combinations of wr_en, wr_en and almost full occurred	Concurrent assertion to check functionality
FIFO_6	When writing in a full fifo, the output overflow should be high	Randomization under constraints on the wr_en to be high 70% of the simulation time	make sure that all the combinations of wr_en, wr_en and overflow occurred	Concurrent assertion to check functionality
FIFO_7	When reading an empty fifo, the output underflow should be high	Randomization under constraints on the rd_en to be high 30% of the simulation time	make sure that all the combinations of wr_en, wr_en and underflow occurred	Concurrent assertion to check functionality
FIFO_8	When writing in a not full fifo, the output write ack should be high	Randomization under constraints on the wr_en to be high 70% of the simulation time	make sure that all the combinations of wr_en, wr_en and write ack occurred	Concurrent assertion to check functionality
FIFO_9	the fifo can't be full and empty at the same time	Randomization	-	Immediate assertion to check functionality
FIFO_10	When reading a not empty fifo, the dout output should have the value written in the fifo first	Randomization under constraints on the rd_en to be high 30% of the simulation time	-	checker to check functionality

FIFO Transaction Class

```
package FIFO_transactions_pkg;
    parameter FIFO_WIDTH = 16;
    parameter FIFO_DEPTH = 8;

    class FIFO_transactions;
        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;
        int WR_EN_ON_DIST;

        function new(int rd_en_on_dist = 30, int wr_en_on_dist = 70);
            RD_EN_ON_DIST = rd_en_on_dist;
            WR_EN_ON_DIST = wr_en_on_dist;
        endfunction

        constraint FIFO_1 {
            rst_n dist {1 :/ 95, 0 :/ 5};
        }

        constraint FIFO_6_8 {
            wr_en dist {1 :/ WR_EN_ON_DIST, 0 :/ 100 - WR_EN_ON_DIST};
        }

        constraint FIFO_7_10 {
            wr_en dist {1 :/ RD_EN_ON_DIST, 0 :/ 100 - RD_EN_ON_DIST};
        }
    endclass
endpackage
```

```
 
```

FIFO Coverage Class

```
package FIFO_coverage_pkg;
    import FIFO_transactions_pkg::*;

    class FIFO_coverage;
        FIFO_transactions F_cvg_txn;

        covergroup CovCode;
            wr_en_cp: coverpoint F_cvg_txn.wr_en iff (F_cvg_txn.rst_n) {
                bins high = {1};
                bins low = {0};
            }

            rd_en_cp: coverpoint F_cvg_txn.rd_en iff (F_cvg_txn.rst_n) {
                bins high = {1};
                bins low = {0};
            }

            full_cp: coverpoint F_cvg_txn.full iff (F_cvg_txn.rst_n) {
                bins high = {1};
                bins low = {0};
            }

            almostfull_cp: coverpoint F_cvg_txn.almostfull iff (F_cvg_txn.rst_n) {
                bins high = {1};
                bins low = {0};
            }

            empty_cp: coverpoint F_cvg_txn.empty iff (F_cvg_txn.rst_n) {
                bins high = {1};
                bins low = {0};
            }

            almostempty_cp: coverpoint F_cvg_txn.almostempty iff (F_cvg_txn.rst_n) {
                bins high = {1};
                bins low = {0};
            }

            overflow_cp: coverpoint F_cvg_txn.overflow iff (F_cvg_txn.rst_n) {
                bins high = {1};
                bins low = {0};
            }

            underflow_cp: coverpoint F_cvg_txn.underflow iff (F_cvg_txn.rst_n) {
                bins high = {1};
                bins low = {0};
            }

            wr_ack_cp: coverpoint F_cvg_txn.wr_ack iff (F_cvg_txn.rst_n) {
                bins high = {1};
                bins low = {0};
            }

            full_cr: cross wr_en_cp, rd_en_cp, full_cp iff (F_cvg_txn.rst_n) {
                ignore_bins ignr = binsof(rd_en_cp.high) && binsof(full_cp.high);
            }

            almostfull_cr: cross wr_en_cp, rd_en_cp, almostfull_cp iff (F_cvg_txn.rst_n);

            empty_cr: cross wr_en_cp, rd_en_cp, empty_cp iff (F_cvg_txn.rst_n) {
                ignore_bins ignr = binsof(wr_en_cp.high) && binsof(empty_cp.high);
            }

            almostempty_cr: cross wr_en_cp, rd_en_cp, almostempty_cp iff (F_cvg_txn.rst_n);

            overflow_cr: cross wr_en_cp, rd_en_cp, overflow_cp iff (F_cvg_txn.rst_n) {
                ignore_bins ignr = binsof(wr_en_cp.low) && binsof(overflow_cp.high);
            }

            underflow_cr: cross wr_en_cp, rd_en_cp, underflow_cp iff (F_cvg_txn.rst_n) {
                ignore_bins ignr = binsof(rd_en_cp.low) && binsof(underflow_cp.high);
            }

            wr_ack_cr: cross wr_en_cp, rd_en_cp, wr_ack_cp iff (F_cvg_txn.rst_n) {
                ignore_bins ignr = binsof(wr_en_cp.low) && binsof(wr_ack_cp.high);
            }
        endgroup

        function new();
            CovCode = new;
        endfunction

        function void sample_data(FIFO_transactions F_txn);
            F_cvg_txn = F_txn;
            CovCode.sample();
        endfunction
    endclass
endpackage

```

FIFO Scoreboard Class

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

    class FIFO_scoreboard;
        logic [FIFO_WIDTH - 1 : 0] data_out_ref;
        logic [FIFO_WIDTH - 1 : 0] q_ref[$];

        task check_data(FIFO_transactions f_trans);
            reference_model(f_trans);

            if (f_trans.data_out != data_out_ref) begin
                error_count++;

                $display("error, rst_n: %b, wr_en: %b, rd_en: %b, data_in: %b, data_out_dut: %h, data_out_ref: %h",
                                f_trans.rst_n, f_trans.wr_en, f_trans.rd_en, f_trans.data_in, f_trans.data_out, data_out_ref);
            end else begin
                correct_count++;
            end
        endtask

        task reference_model(FIFO_transactions f_trans);
            if (!f_trans.rst_n) begin
                q_ref.delete();
            end else begin
```

```
                bit is_wr, is_rd;

                if (f_trans.wr_en && f_trans.rd_en) begin
                    if (q_ref.size() == 0) begin
                        is_wr = 1;
                        is_rd = 0;
                    end else if (q_ref.size() == FIFO_DEPTH) begin
                        is_wr = 0;
                        is_rd = 1;
                    end else begin
                        is_wr = 1;
                        is_rd = 1;
                    end
                end else begin
                    is_wr = f_trans.wr_en;
                    is_rd = f_trans.rd_en;
                end

                if (is_wr) begin
                    q_ref.push_back(f_trans.data_in);
                end

                if (is_rd) begin
                    data_out_ref = q_ref.pop_front();
                end
            end
        endtask
    endclass
endpackage

```

FIFO Testbench Module

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

module FIFO_tb (IFIFO.DUT fifo_if);
    FIFO_transactions fifo_trans = new;
    FIFO_scoreboard fifo_score = new;
    FIFO_coverage fifo_cov = new;

    initial begin
        fifo_trans.rst_n = 0;
        fifo_if.rst_n = fifo_trans.rst_n;

        @(negedge fifo_if.clk);

        ->fifo_if.start_to_sample;

        fifo_score.check_data(fifo_trans);

        fifo_trans.rst_n = 1;
        fifo_if.rst_n = fifo_trans.rst_n;

        repeat (1000) begin
            fifo_trans.randomize();

            fifo_if.rst_n = fifo_trans.rst_n;
            fifo_if.data_in = fifo_trans.data_in;
            fifo_if.wr_en = fifo_trans.wr_en;
            fifo_if.rd_en = fifo_trans.rd_en;

            @(negedge fifo_if.clk);

            ->fifo_if.start_to_sample;
        end

        test_finished = 1;

        @(negedge fifo_if.clk);

        ->fifo_if.start_to_sample;
    end
endmodule

```

FIFO Monitor Module

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

module FIFO_mon(IFIFO.MONITOR fifo_if);
    FIFO_transactions fifo_trans = new;
    FIFO_scoreboard fifo_score = new;
    FIFO_coverage fifo_cov = new;

    initial begin
        forever begin
            @fifo_if.start_to_sample;
            @(negedge fifo_if.clk);

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

            fork
                begin
                    fifo_cov.sample_data(fifo_trans);
                end

                begin
                    fifo_score.check_data(fifo_trans);
                end
            join

            if (test_finished) begin
                $display("error: %d, correct: %d", error_count, correct_count);

                $stop;
            end
        end
    end
endmodule

```

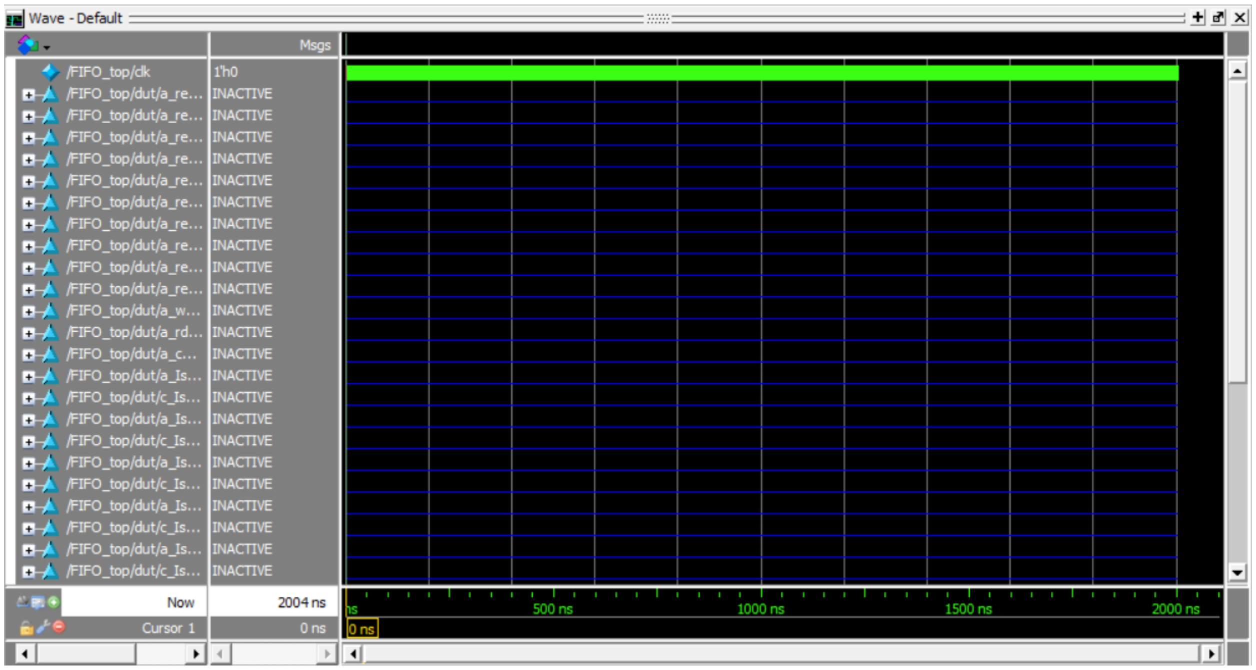
FIFO Top Module

```
module FIFO_top();
    bit clk;

    always #1 clk = ~clk;

    IFIFO fifo_if(clk);

    FIFO dut(fifo_if);
    FIFO_tb tb(fifo_if);
    FIFO_mon mon(fifo_if);
endmodule

```

Do File

```
vlib work
vlog *v  +cover +define+SIM
vsim -voptargs=+acc FIFO_top -cover
add wave *
add wave /FIFO_top/dut/a_reset_empty /FIFO_top/dut/a_reset_full /FIFO_top/dut/a_reset_almostempty
/FIFO_top/dut/a_reset_almostfull /FIFO_top/dut/a_reset_overflow /FIFO_top/dut/a_reset_underflow
/FIFO_top/dut/a_reset_wr_ack /FIFO_top/dut/a_reset_wr_ptr /FIFO_top/dut/a_reset_rd_ptr
/FIFO_top/dut/a_reset_count /FIFO_top/dut/a_wr_ptr_threshold /FIFO_top/dut/a_rd_ptr_threshold
/FIFO_top/dut/a_count_threshold /FIFO_top/dut/a_IsFullEmpty /FIFO_top/dut/c_IsFullEmpty
/FIFO_top/dut/a_IsEmpty /FIFO_top/dut/c_IsEmpty /FIFO_top/dut/a_IsNotEmpty
/FIFO_top/dut/c_IsNotEmpty /FIFO_top/dut/a_IsAlmostEmpty /FIFO_top/dut/c_IsAlmostEmpty
/FIFO_top/dut/a_IsFull /FIFO_top/dut/c_IsFull /FIFO_top/dut/a_IsNotFull /FIFO_top/dut/c_IsNotFull
/FIFO_top/dut/a_IsAlmostFull /FIFO_top/dut/c_IsAlmostFull /FIFO_top/dut/a_IsOverflow
/FIFO_top/dut/c_IsOverflow /FIFO_top/dut/a_IsUnderflow /FIFO_top/dut/c_IsUnderflow
/FIFO_top/dut/a_IsWriteAck /FIFO_top/dut/c_IsWriteAck /FIFO_top/dut/a_wr_ptr_wrap
/FIFO_top/dut/c_wr_ptr_wrap /FIFO_top/dut/a_rd_ptr_wrap /FIFO_top/dut/c_rd_ptr_wrap
coverage save FIFO.ucdb -onexit
run -all

```

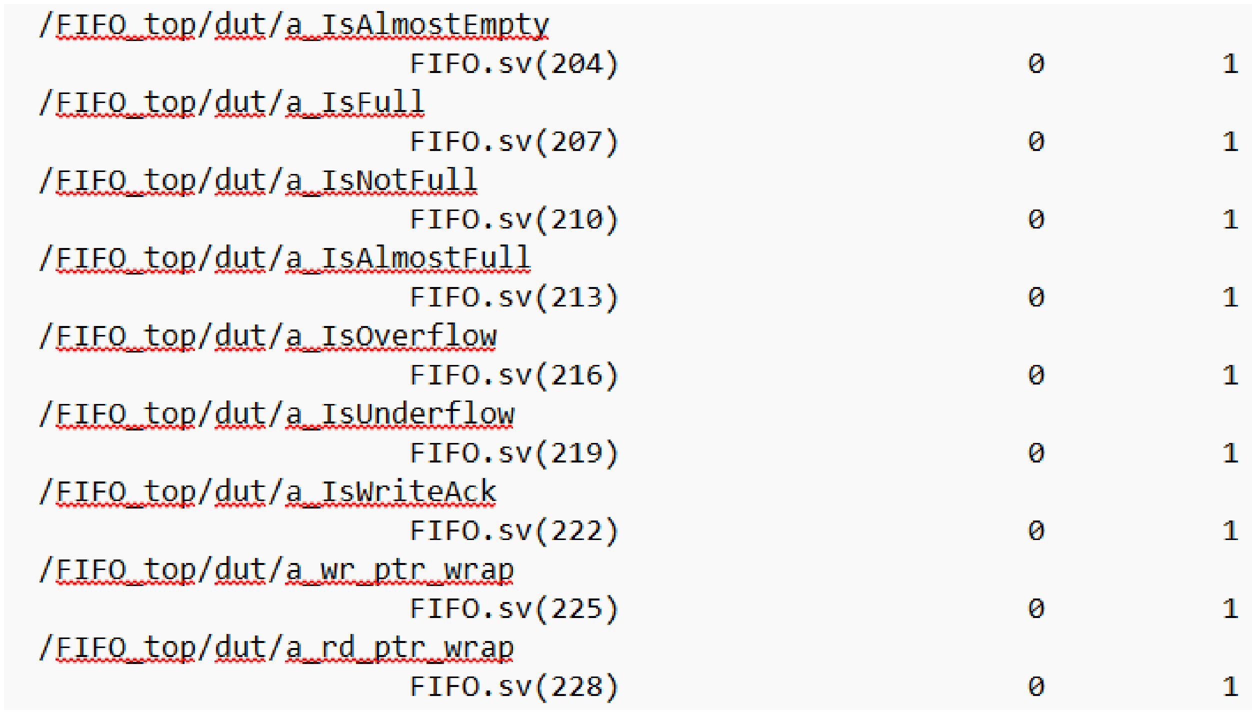
Waveform



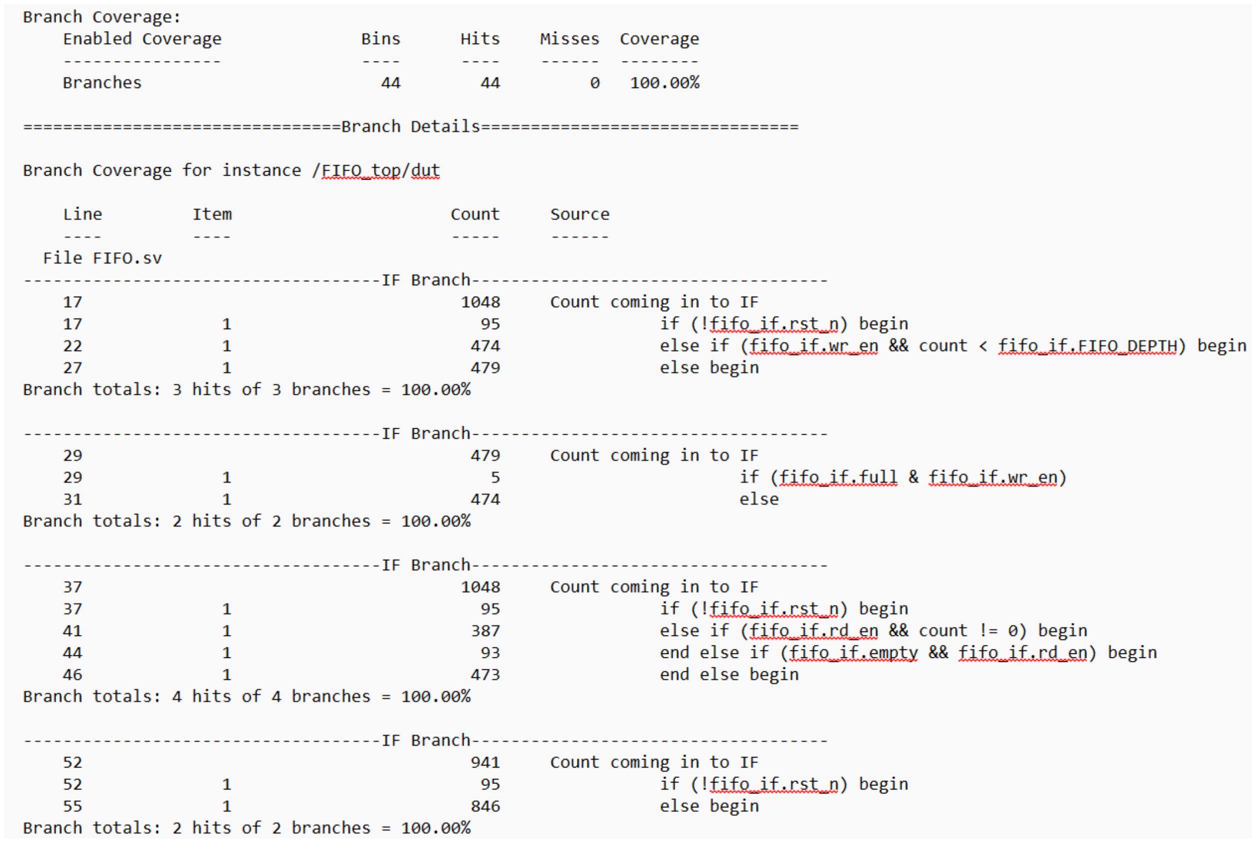
```
/FIFO_top/clk            1'h0
/FIFO_top/dut/a_re...    INACTIVE
/FIFO_top/dut/a_re...    INACTIVE
/FIFO_top/dut/a_re...    INACTIVE
/FIFO_top/dut/a_re...    INACTIVE
/FIFO_top/dut/a_re...    INACTIVE
/FIFO_top/dut/a_re...    INACTIVE
/FIFO_top/dut/a_re...    INACTIVE
/FIFO_top/dut/a_re...    INACTIVE
/FIFO_top/dut/a_re...    INACTIVE
/FIFO_top/dut/a_re...    INACTIVE
/FIFO_top/dut/a_w...     INACTIVE
/FIFO_top/dut/a_rd...    INACTIVE
/FIFO_top/dut/a_c...     INACTIVE
/FIFO_top/dut/a_Is...    INACTIVE
/FIFO_top/dut/c_Is...    INACTIVE
/FIFO_top/dut/a_Is...    INACTIVE
/FIFO_top/dut/c_Is...    INACTIVE
/FIFO_top/dut/a_Is...    INACTIVE
/FIFO_top/dut/c_Is...    INACTIVE
/FIFO_top/dut/a_Is...    INACTIVE
/FIFO_top/dut/c_Is...    INACTIVE
/FIFO_top/dut/a_Is...    INACTIVE
/FIFO_top/dut/c_Is...    INACTIVE
Now                      2004 ns
Cursor 1                 0 ns
```

Coverage Report

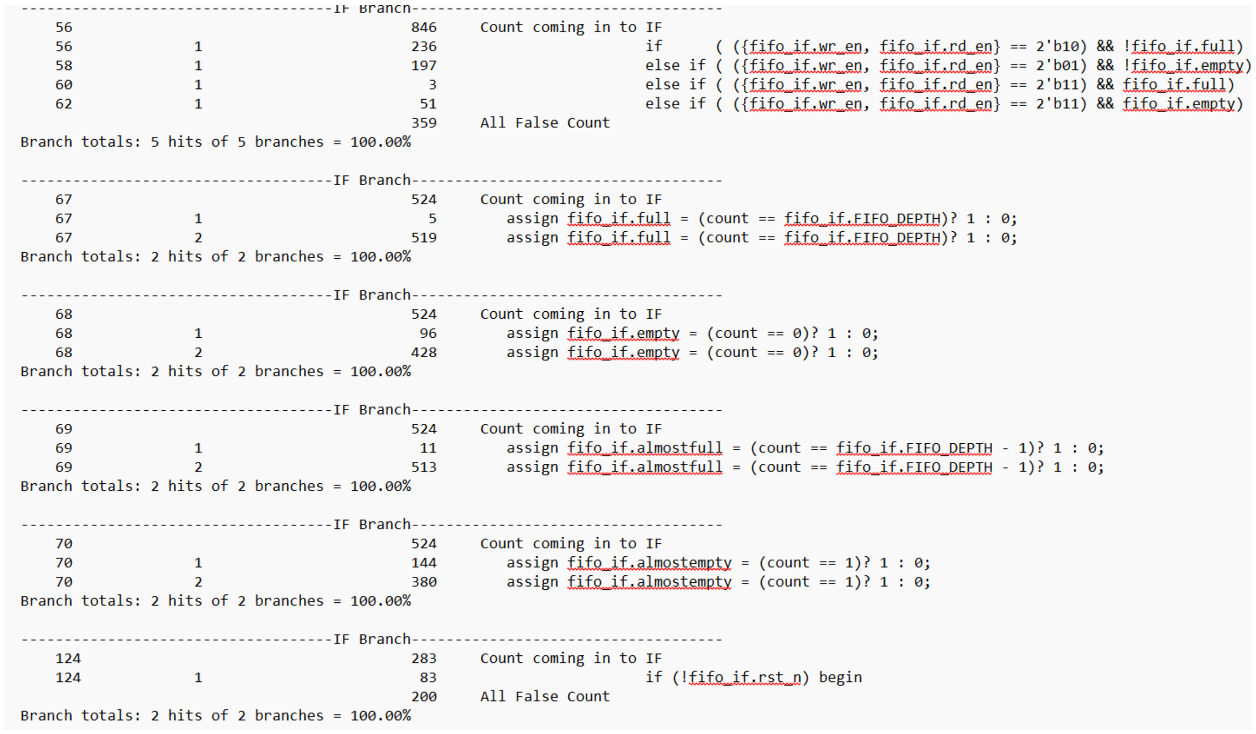
```
Assertion Coverage:
    Assertions                      25         25           0    100.00%
--------------------------------------------------------------------
Name                     File(Line)                  Failure      Pass
                                                     Count        Count
--------------------------------------------------------------------
/FIFO_top/dut/a_reset_empty
                         FIFO.sv(125)                      0          1
/FIFO_top/dut/a_reset_full
                         FIFO.sv(131)                      0          1
/FIFO_top/dut/a_reset_almostempty
                         FIFO.sv(137)                      0          1
/FIFO_top/dut/a_reset_almostfull
                         FIFO.sv(143)                      0          1
/FIFO_top/dut/a_reset_overflow
                         FIFO.sv(149)                      0          1
/FIFO_top/dut/a_reset_underflow
                         FIFO.sv(155)                      0          1
/FIFO_top/dut/a_reset_wr_ack
                         FIFO.sv(161)                      0          1
/FIFO_top/dut/a_reset_wr_ptr
                         FIFO.sv(167)                      0          1
/FIFO_top/dut/a_reset_rd_ptr
                         FIFO.sv(173)                      0          1
/FIFO_top/dut/a_reset_count
                         FIFO.sv(179)                      0          1
/FIFO_top/dut/a_wr_ptr_threshold
                         FIFO.sv(184)                      0          1
/FIFO_top/dut/a_rd_ptr_threshold
                         FIFO.sv(188)                      0          1
/FIFO_top/dut/a_count_threshold
                         FIFO.sv(192)                      0          1
/FIFO_top/dut/a_IsFullEmpty
                         FIFO.sv(195)                      0          1
/FIFO_top/dut/a_IsEmpty
                         FIFO.sv(198)                      0          1
/FIFO_top/dut/a_IsNotEmpty
                         FIFO.sv(201)                      0          1
```



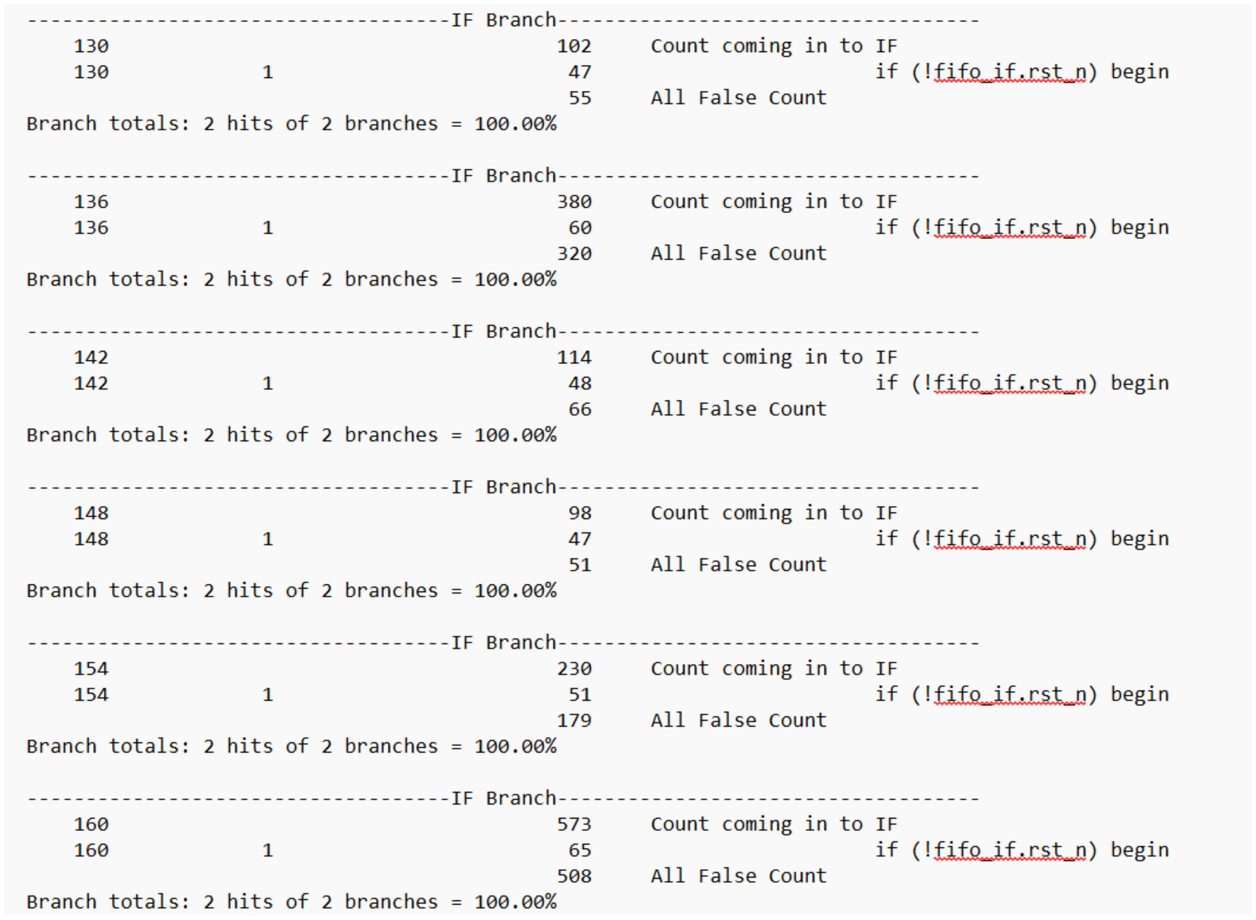
```
 /FIFO_top/dut/a_IsAlmostEmpty
                    FIFO.sv(204)                     0          1
 /FIFO_top/dut/a_IsFull
                    FIFO.sv(207)                     0          1
 /FIFO_top/dut/a_IsNotFull
                    FIFO.sv(210)                     0          1
 /FIFO_top/dut/a_IsAlmostFull
                    FIFO.sv(213)                     0          1
 /FIFO_top/dut/a_IsOverflow
                    FIFO.sv(216)                     0          1
 /FIFO_top/dut/a_IsUnderflow
                    FIFO.sv(219)                     0          1
 /FIFO_top/dut/a_IsWriteAck
                    FIFO.sv(222)                     0          1
 /FIFO_top/dut/a_wr_ptr_wrap
                    FIFO.sv(225)                     0          1
 /FIFO_top/dut/a_rd_ptr_wrap
                    FIFO.sv(228)                     0          1
```



```
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        44        44         0   100.00%

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

Branch Coverage for instance /FIFO_top/dut

    Line         Item                      Count     Source 
    ----         ----                      -----     ------ 
  File FIFO.sv
------------------------------------IF Branch------------------------------------
    17                                      1048     Count coming in to IF
    17              1                         95                if (!fifo_if.rst_n) begin
    22              1                        474                else if (fifo_if.wr_en && count < fifo_if.FIFO_DEPTH) begin
    27              1                        479                else begin
Branch totals: 3 hits of 3 branches = 100.00%

------------------------------------IF Branch------------------------------------
    29                                       479     Count coming in to IF
    29              1                          5                        if (fifo_if.full & fifo_if.wr_en)
    31              1                        474                        else
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    37                                      1048     Count coming in to IF
    37              1                         95                if (!fifo_if.rst_n) begin
    41              1                        387                else if (fifo_if.rd_en && count != 0) begin
    44              1                         93                end else if (fifo_if.empty && fifo_if.rd_en) begin
    46              1                        473                end else begin
Branch totals: 4 hits of 4 branches = 100.00%

------------------------------------IF Branch------------------------------------
    52                                       941     Count coming in to IF
    52              1                         95                if (!fifo_if.rst_n) begin
    55              1                        846                else begin
Branch totals: 2 hits of 2 branches = 100.00%
```

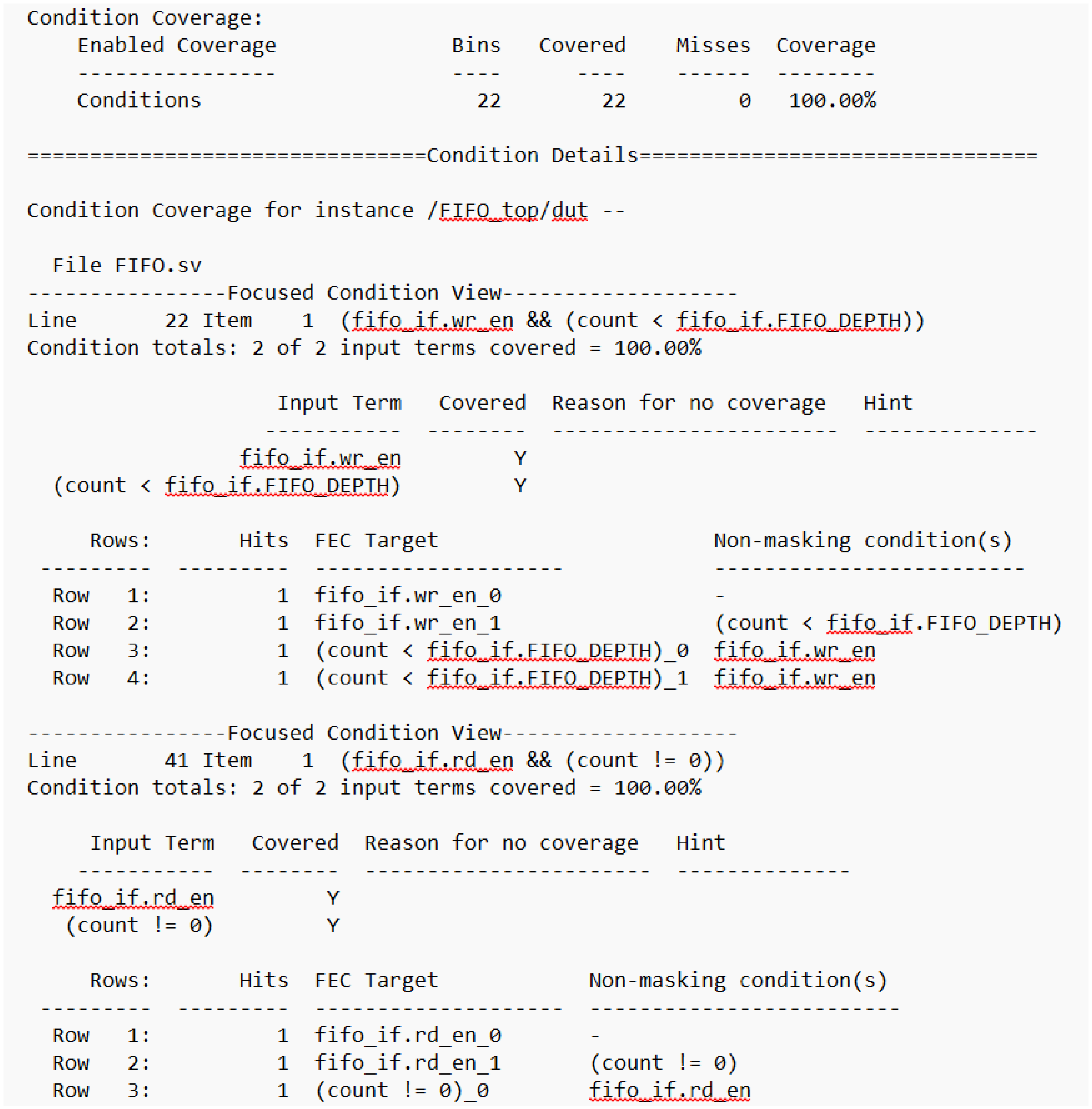


```
-----------------------------------IF Branch-----------------------------------
    56                                846     Count coming in to IF
    56              1                 236                        if      ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b10) && !fifo_if.full)
    58              1                 197                        else if ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b01) && !fifo_if.empty)
    60              1                   3                        else if ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b11) && fifo_if.full)
    62              1                  51                        else if ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b11) && fifo_if.empty)
                                      359     All False Count
Branch totals: 5 hits of 5 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    67                                524     Count coming in to IF
    67              1                   5        assign fifo_if.full = (count == fifo_if.FIFO_DEPTH)? 1 : 0;
    67              2                 519        assign fifo_if.full = (count == fifo_if.FIFO_DEPTH)? 1 : 0;
Branch totals: 2 hits of 2 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    68                                524     Count coming in to IF
    68              1                  96        assign fifo_if.empty = (count == 0)? 1 : 0;
    68              2                 428        assign fifo_if.empty = (count == 0)? 1 : 0;
Branch totals: 2 hits of 2 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    69                                524     Count coming in to IF
    69              1                  11        assign fifo_if.almostfull = (count == fifo_if.FIFO_DEPTH - 1)? 1 : 0;
    69              2                 513        assign fifo_if.almostfull = (count == fifo_if.FIFO_DEPTH - 1)? 1 : 0;
Branch totals: 2 hits of 2 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    70                                524     Count coming in to IF
    70              1                 144        assign fifo_if.almostempty = (count == 1)? 1 : 0;
    70              2                 380        assign fifo_if.almostempty = (count == 1)? 1 : 0;
Branch totals: 2 hits of 2 branches = 100.00%

-----------------------------------IF Branch-----------------------------------
    124                               283     Count coming in to IF
    124             1                  83                        if (!fifo_if.rst_n) begin
                                      200     All False Count
Branch totals: 2 hits of 2 branches = 100.00%
```



```
------------------------------------IF Branch------------------------------------
    130                                       102     Count coming in to IF
    130              1                         47                       if (!fifo_if.rst_n) begin
                                               55     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    136                                       380     Count coming in to IF
    136              1                         60                       if (!fifo_if.rst_n) begin
                                              320     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    142                                       114     Count coming in to IF
    142              1                         48                       if (!fifo_if.rst_n) begin
                                               66     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    148                                        98     Count coming in to IF
    148              1                         47                       if (!fifo_if.rst_n) begin
                                               51     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    154                                       230     Count coming in to IF
    154              1                         51                       if (!fifo_if.rst_n) begin
                                              179     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    160                                       573     Count coming in to IF
    160              1                         65                       if (!fifo_if.rst_n) begin
                                              508     All False Count
Branch totals: 2 hits of 2 branches = 100.00%
```

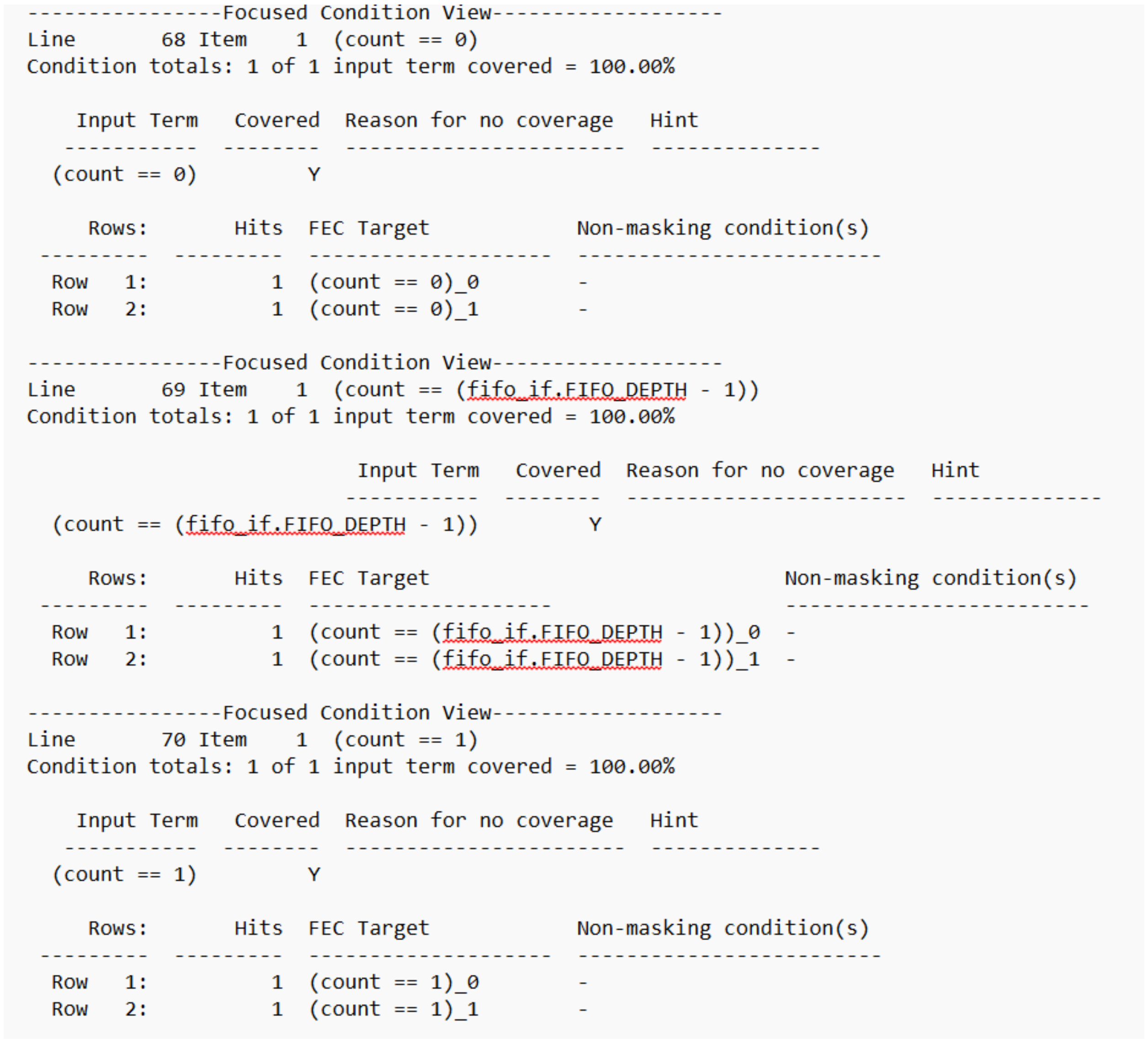
```
------------------------------------IF Branch------------------------------------
    166                                      603     Count coming in to IF
    166              1                        84                         if (!fifo_if.rst_n) begin
                                             519     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    172                                      512     Count coming in to IF
    172              1                        80                         if (!fifo_if.rst_n) begin
                                             432     All False Count
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    178                                      615     Count coming in to IF
    178              1                        83                         if (!fifo_if.rst_n) begin
                                             532     All False Count
Branch totals: 2 hits of 2 branches = 100.00%
```



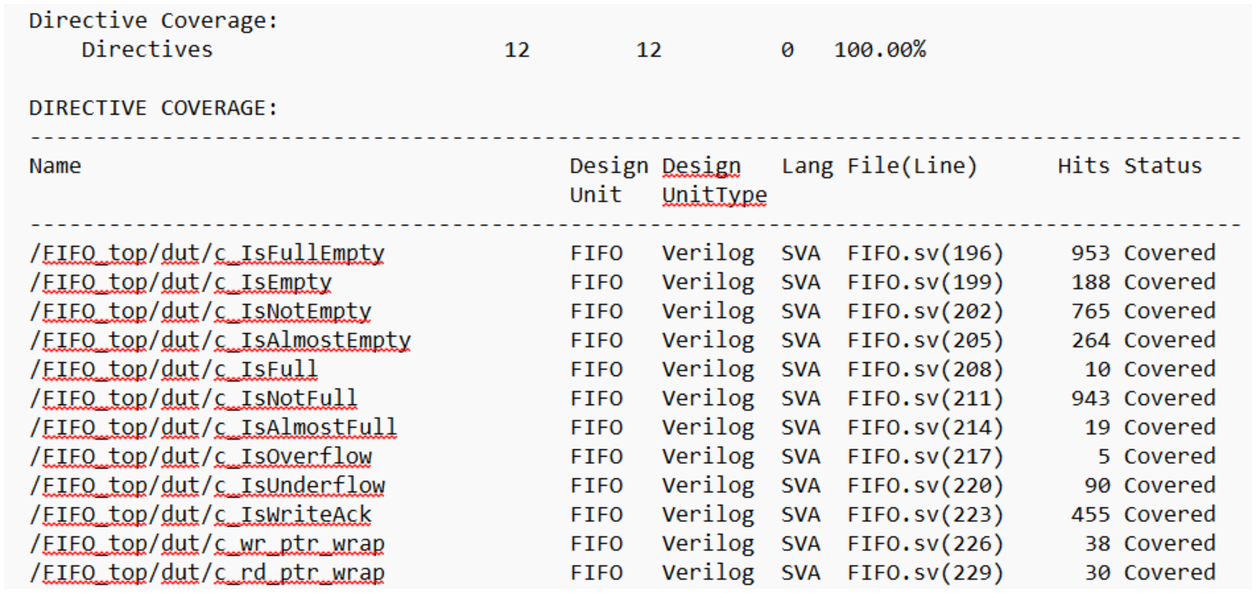
```
Condition Coverage:
    Enabled Coverage              Bins    Covered    Misses  Coverage
    ----------------              ----       ----    ------  --------
    Conditions                      22         22         0   100.00%

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

Condition Coverage for instance /FIFO_top/dut --

  File FIFO.sv
----------------Focused Condition View-------------------
Line       22 Item    1  (fifo_if.wr_en && (count < fifo_if.FIFO_DEPTH))
Condition totals: 2 of 2 input terms covered = 100.00%

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

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

----------------Focused Condition View-------------------
Line       41 Item    1  (fifo_if.rd_en && (count != 0))
Condition totals: 2 of 2 input terms covered = 100.00%

     Input Term   Covered  Reason for no coverage   Hint
    -----------  --------  -----------------------  --------------
  fifo_if.rd_en         Y
   (count != 0)         Y

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  fifo_if.rd_en_0       -
  Row   2:          1  fifo_if.rd_en_1       (count != 0)
  Row   3:          1  (count != 0)_0        fifo_if.rd_en
  Row   4:          1  (count != 0)_1        fifo_if.rd_en

----------------Focused Condition View-------------------
Line       44 Item    1  (fifo_if.empty && fifo_if.rd_en)
Condition totals: 2 of 2 input terms covered = 100.00%

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

     Rows:       Hits  FEC Target            Non-masking condition(s)
 ---------  ---------  --------------------  -------------------------
  Row   1:          1  fifo_if.empty_0       -
  Row   2:          1  fifo_if.empty_1       fifo_if.rd_en
  Row   3:          1  fifo_if.rd_en_0       fifo_if.empty
  Row   4:          1  fifo_if.rd_en_1       fifo_if.empty

----------------Focused Condition View-------------------
Line       56 Item    1  ((~fifo_if.rd_en && fifo_if.wr_en) && ~fifo_if.full)
Condition totals: 3 of 3 input terms covered = 100.00%

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

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

```
----------------Focused Condition View-------------------
Line       58 Item    1  ((fifo_if.rd_en && ~fifo_if.wr_en) && ~fifo_if.empty)
Condition totals: 3 of 3 input terms covered = 100.00%

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

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

----------------Focused Condition View-------------------
Line       60 Item    1  ((fifo_if.rd_en && fifo_if.wr_en) && fifo_if.full)
Condition totals: 3 of 3 input terms covered = 100.00%

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

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

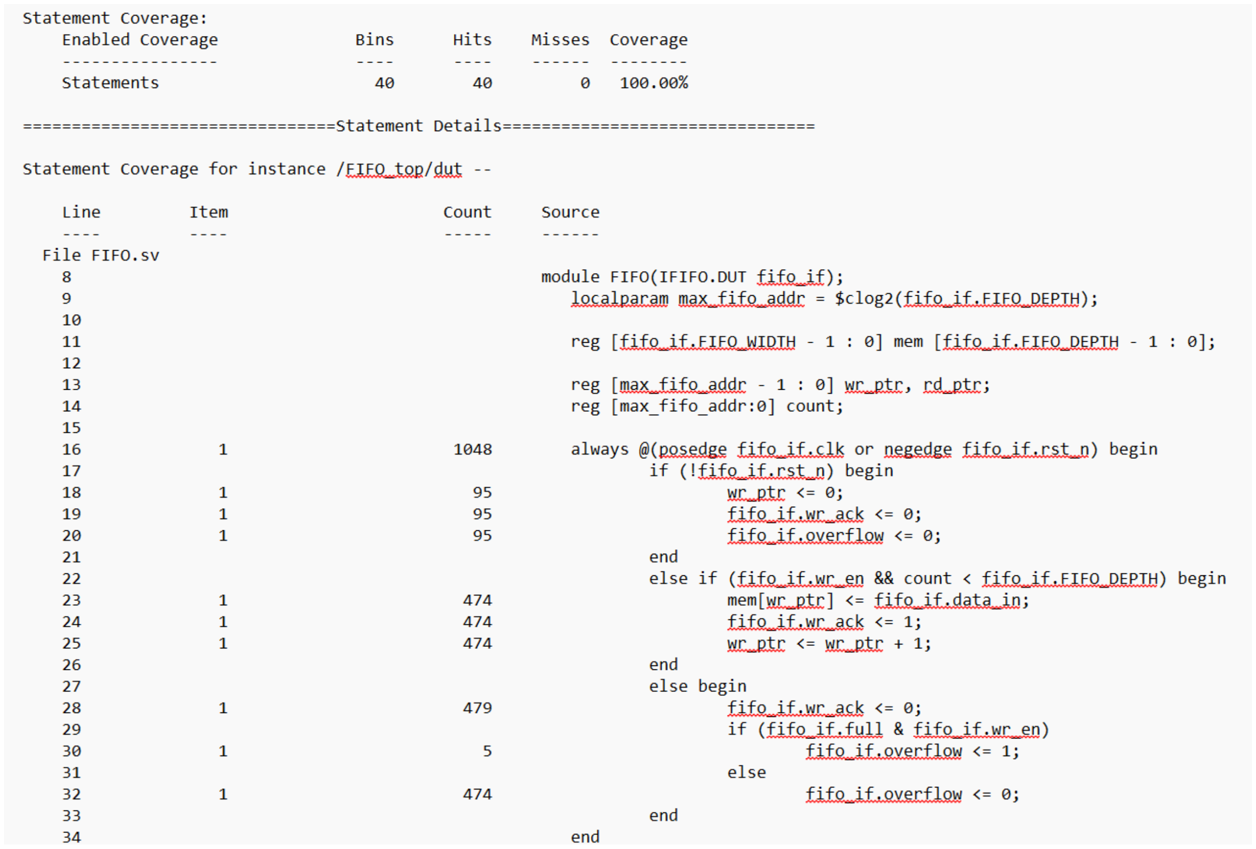
```
----------------Focused Condition View-------------------
Line       62 Item    1  ((fifo_if.rd_en && fifo_if.wr_en) && fifo_if.empty)
Condition totals: 3 of 3 input terms covered = 100.00%

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

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

----------------Focused Condition View-------------------
Line       67 Item    1  (count == fifo_if.FIFO_DEPTH)
Condition totals: 1 of 1 input term covered = 100.00%

                     Input Term   Covered  Reason for no coverage   Hint
                    -----------  --------  ----------------------  --------------
  (count == fifo_if.FIFO_DEPTH)         Y

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



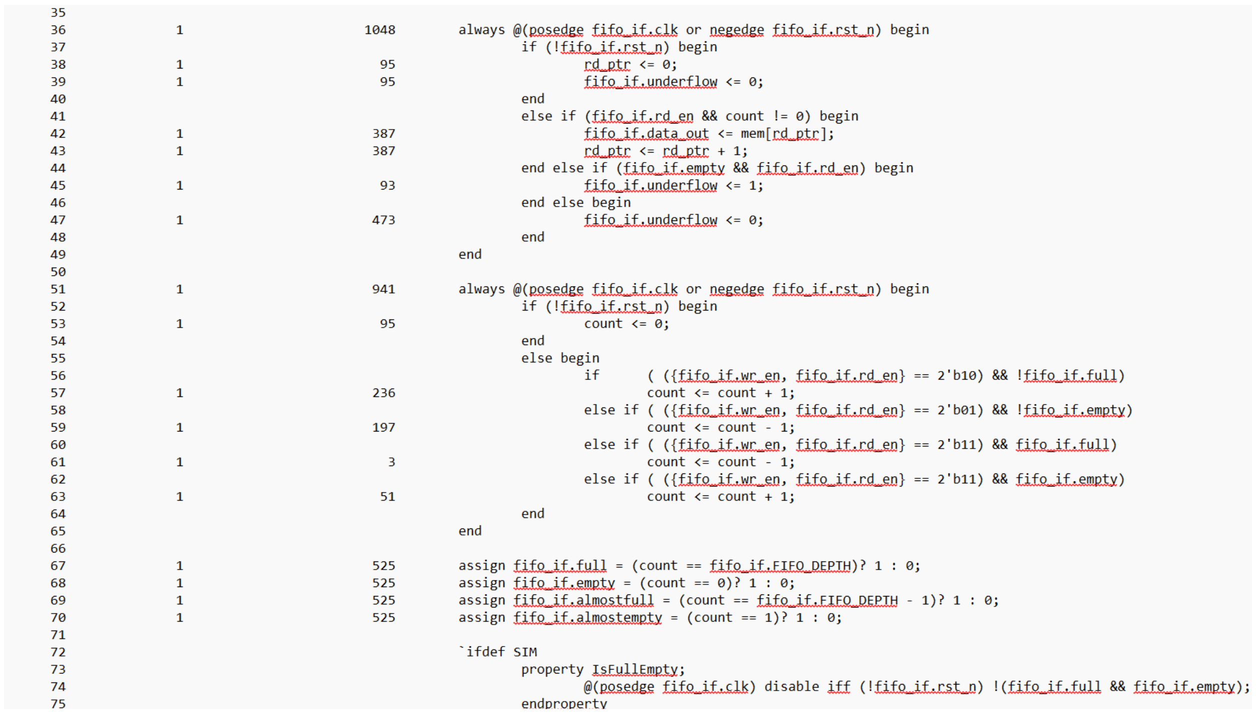
```
----------------Focused Condition View-------------------
Line       68 Item    1  (count == 0)
Condition totals: 1 of 1 input term covered = 100.00%

    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       69 Item    1  (count == (fifo_if.FIFO_DEPTH - 1))
Condition totals: 1 of 1 input term covered = 100.00%

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

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

----------------Focused Condition View-------------------
Line       70 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        -
```



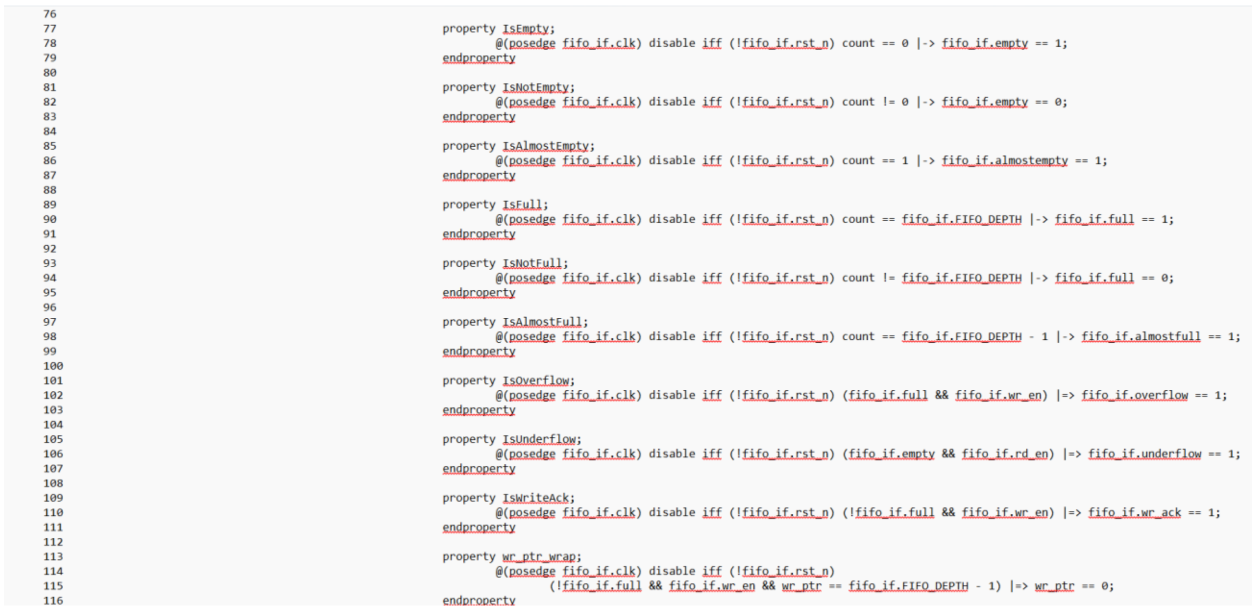
```
Directive Coverage:
    Directives                          12        12         0   100.00%

DIRECTIVE COVERAGE:
------------------------------------------------------------------------------------------
Name                                    Design Design   Lang File(Line)      Hits Status
                                        Unit   UnitType
------------------------------------------------------------------------------------------
/FIFO_top/dut/c_IsFullEmpty             FIFO   Verilog  SVA  FIFO.sv(196)     953 Covered
/FIFO_top/dut/c_IsEmpty                 FIFO   Verilog  SVA  FIFO.sv(199)     188 Covered
/FIFO_top/dut/c_IsNotEmpty              FIFO   Verilog  SVA  FIFO.sv(202)     765 Covered
/FIFO_top/dut/c_IsAlmostEmpty           FIFO   Verilog  SVA  FIFO.sv(205)     264 Covered
/FIFO_top/dut/c_IsFull                  FIFO   Verilog  SVA  FIFO.sv(208)      10 Covered
/FIFO_top/dut/c_IsNotFull               FIFO   Verilog  SVA  FIFO.sv(211)     943 Covered
/FIFO_top/dut/c_IsAlmostFull            FIFO   Verilog  SVA  FIFO.sv(214)      19 Covered
/FIFO_top/dut/c_IsOverflow              FIFO   Verilog  SVA  FIFO.sv(217)       5 Covered
/FIFO_top/dut/c_IsUnderflow             FIFO   Verilog  SVA  FIFO.sv(220)      90 Covered
/FIFO_top/dut/c_IsWriteAck              FIFO   Verilog  SVA  FIFO.sv(223)     455 Covered
/FIFO_top/dut/c_wr_ptr_wrap             FIFO   Verilog  SVA  FIFO.sv(226)      38 Covered
/FIFO_top/dut/c_rd_ptr_wrap             FIFO   Verilog  SVA  FIFO.sv(229)      30 Covered
```



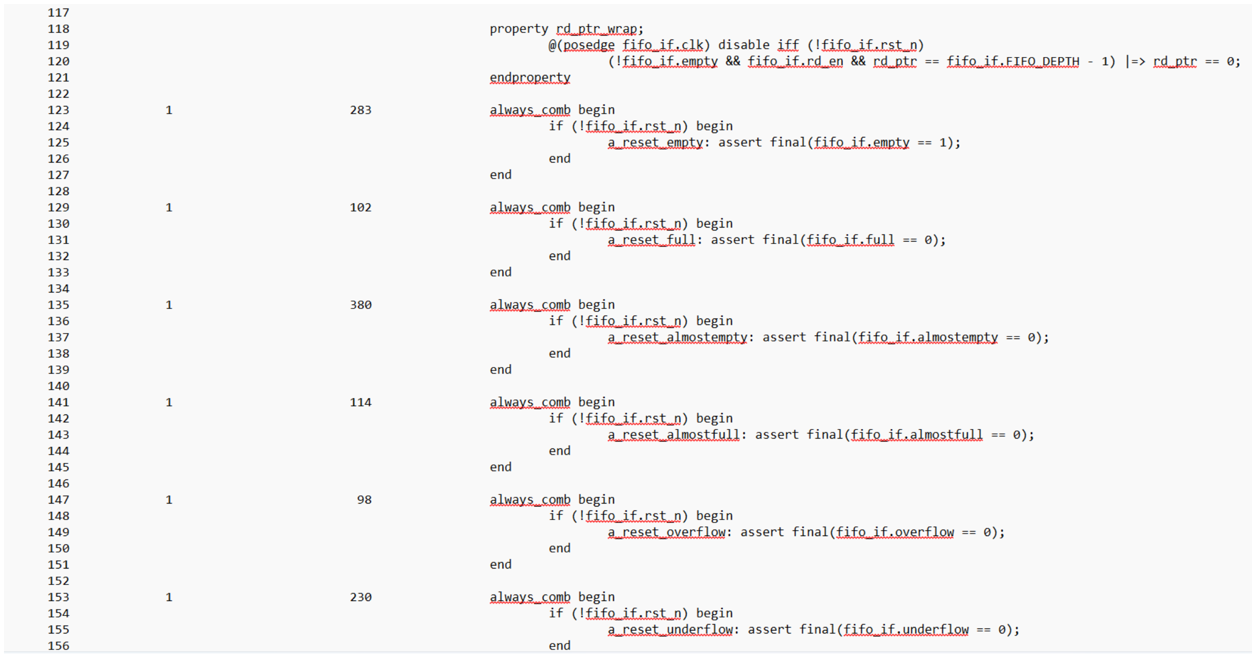
```
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                      40        40         0   100.00%

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

Statement Coverage for instance /FIFO_top/dut --

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File FIFO.sv
    8                                                module FIFO(IFIFO.DUT fifo_if);
    9                                                   localparam max_fifo_addr = $clog2(fifo_if.FIFO_DEPTH);
    10
    11                                                  reg [fifo_if.FIFO_WIDTH - 1 : 0] mem [fifo_if.FIFO_DEPTH - 1 : 0];
    12
    13                                                  reg [max_fifo_addr - 1 : 0] wr_ptr, rd_ptr;
    14                                                  reg [max_fifo_addr:0] count;
    15
    16             1                        1048        always @(posedge fifo_if.clk or negedge fifo_if.rst_n) begin
    17                                                          if (!fifo_if.rst_n) begin
    18             1                          95                        wr_ptr <= 0;
    19             1                          95                        fifo_if.wr_ack <= 0;
    20             1                          95                        fifo_if.overflow <= 0;
    21                                                          end
    22                                                          else if (fifo_if.wr_en && count < fifo_if.FIFO_DEPTH) begin
    23             1                         474                        mem[wr_ptr] <= fifo_if.data_in;
    24             1                         474                        fifo_if.wr_ack <= 1;
    25             1                         474                        wr_ptr <= wr_ptr + 1;
    26                                                          end
    27                                                          else begin
    28             1                         479                        fifo_if.wr_ack <= 0;
    29                                                                  if (fifo_if.full & fifo_if.wr_en)
    30             1                           5                                fifo_if.overflow <= 1;
    31                                                                  else
    32             1                         474                                fifo_if.overflow <= 0;
    33                                                          end
    34                                                  end
```



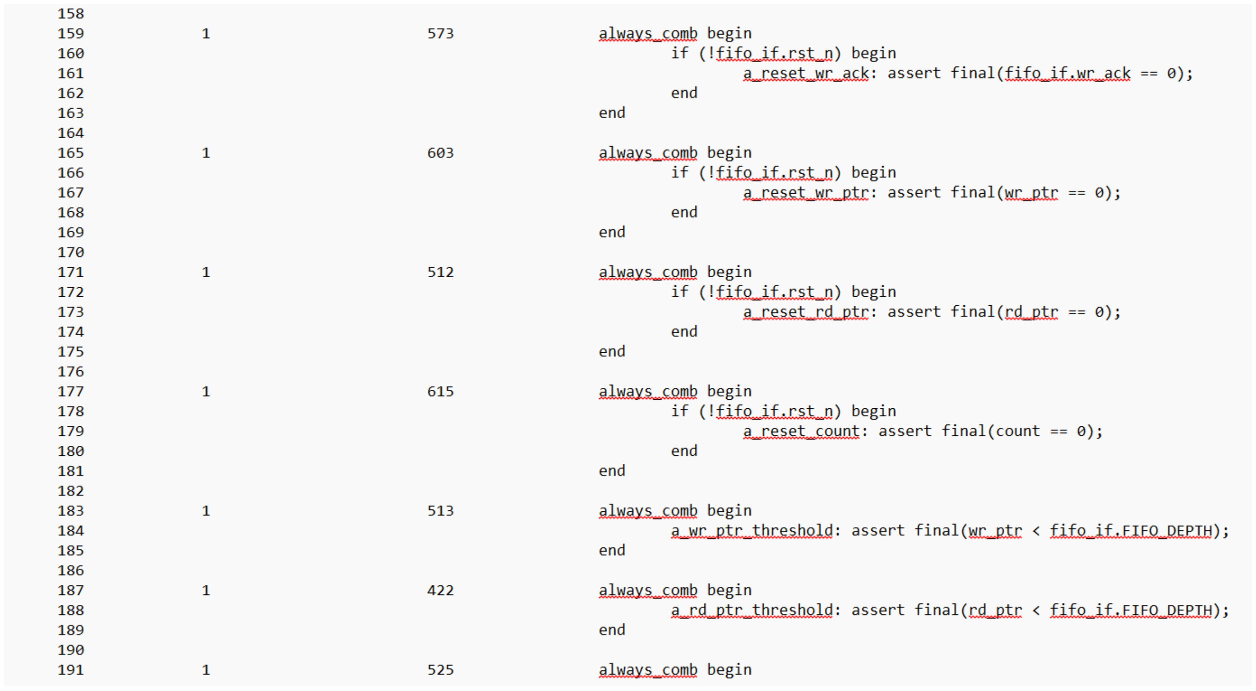
```
1    1048    always @(posedge fifo_if.clk or negedge fifo_if.rst_n) begin
                     if (!fifo_if.rst_n) begin
1      95                    rd_ptr <= 0;
1      95                    fifo_if.underflow <= 0;
                     end
                     else if (fifo_if.rd_en && count != 0) begin
1     387                    fifo_if.data_out <= mem[rd_ptr];
1     387                    rd_ptr <= rd_ptr + 1;
                     end else if (fifo_if.empty && fifo_if.rd_en) begin
1      93                    fifo_if.underflow <= 1;
                     end else begin
1     473                    fifo_if.underflow <= 0;
                     end
             end

1     941    always @(posedge fifo_if.clk or negedge fifo_if.rst_n) begin
                     if (!fifo_if.rst_n) begin
1      95                    count <= 0;
                     end
                     else begin
                             if      ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b10) && !fifo_if.full)
1     236                            count <= count + 1;
                             else if ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b01) && !fifo_if.empty)
1     197                            count <= count - 1;
                             else if ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b11) && fifo_if.full)
1       3                            count <= count - 1;
                             else if ( ({fifo_if.wr_en, fifo_if.rd_en} == 2'b11) && fifo_if.empty)
1      51                            count <= count + 1;
                     end
             end

1     525    assign fifo_if.full = (count == fifo_if.FIFO_DEPTH)? 1 : 0;
1     525    assign fifo_if.empty = (count == 0)? 1 : 0;
1     525    assign fifo_if.almostfull = (count == fifo_if.FIFO_DEPTH - 1)? 1 : 0;
1     525    assign fifo_if.almostempty = (count == 1)? 1 : 0;

             `ifdef SIM
                     property IsFullEmpty;
                             @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) !(fifo_if.full && fifo_if.empty);
                     endproperty
```



```

property IsEmpty;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count == 0 |-> fifo_if.empty == 1;
endproperty

property IsNotEmpty;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count != 0 |-> fifo_if.empty == 0;
endproperty

property IsAlmostEmpty;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count == 1 |-> fifo_if.almostempty == 1;
endproperty

property IsFull;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count == fifo_if.FIFO_DEPTH |-> fifo_if.full == 1;
endproperty

property IsNotFull;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count != fifo_if.FIFO_DEPTH |-> fifo_if.full == 0;
endproperty

property IsAlmostFull;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) count == fifo_if.FIFO_DEPTH - 1 |-> fifo_if.almostfull == 1;
endproperty

property IsOverflow;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) (fifo_if.full && fifo_if.wr_en) |=> fifo_if.overflow == 1;
endproperty

property IsUnderflow;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) (fifo_if.empty && fifo_if.rd_en) |=> fifo_if.underflow == 1;
endproperty

property IsWriteAck;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n) (!fifo_if.full && fifo_if.wr_en) |=> fifo_if.wr_ack == 1;
endproperty

property wr_ptr_wrap;
        @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n)
                (!fifo_if.full && fifo_if.wr_en && wr_ptr == fifo_if.FIFO_DEPTH - 1) |=> wr_ptr == 0;
endproperty
```



```
117
118                                             property rd_ptr_wrap;
119                                                     @(posedge fifo_if.clk) disable iff (!fifo_if.rst_n)
120                                                             (!fifo_if.empty && fifo_if.rd_en && rd_ptr == fifo_if.FIFO_DEPTH - 1) |=> rd_ptr == 0;
121                                             endproperty
122
123          1                283               always_comb begin
124                                                     if (!fifo_if.rst_n) begin
125                                                             a_reset_empty: assert final(fifo_if.empty == 1);
126                                                     end
127                                             end
128
129          1                102               always_comb begin
130                                                     if (!fifo_if.rst_n) begin
131                                                             a_reset_full: assert final(fifo_if.full == 0);
132                                                     end
133                                             end
134
135          1                380               always_comb begin
136                                                     if (!fifo_if.rst_n) begin
137                                                             a_reset_almostempty: assert final(fifo_if.almostempty == 0);
138                                                     end
139                                             end
140
141          1                114               always_comb begin
142                                                     if (!fifo_if.rst_n) begin
143                                                             a_reset_almostfull: assert final(fifo_if.almostfull == 0);
144                                                     end
145                                             end
146
147          1                 98               always_comb begin
148                                                     if (!fifo_if.rst_n) begin
149                                                             a_reset_overflow: assert final(fifo_if.overflow == 0);
150                                                     end
151                                             end
152
153          1                230               always_comb begin
154                                                     if (!fifo_if.rst_n) begin
155                                                             a_reset_underflow: assert final(fifo_if.underflow == 0);
156                                                     end
```



```
158
159          1                573               always_comb begin
160                                                     if (!fifo_if.rst_n) begin
161                                                             a_reset_wr_ack: assert final(fifo_if.wr_ack == 0);
162                                                     end
163                                             end
164
165          1                603               always_comb begin
166                                                     if (!fifo_if.rst_n) begin
167                                                             a_reset_wr_ptr: assert final(wr_ptr == 0);
168                                                     end
169                                             end
170
171          1                512               always_comb begin
172                                                     if (!fifo_if.rst_n) begin
173                                                             a_reset_rd_ptr: assert final(rd_ptr == 0);
174                                                     end
175                                             end
176
177          1                615               always_comb begin
178                                                     if (!fifo_if.rst_n) begin
179                                                             a_reset_count: assert final(count == 0);
180                                                     end
181                                             end
182
183          1                513               always_comb begin
184                                                     a_wr_ptr_threshold: assert final(wr_ptr < fifo_if.FIFO_DEPTH);
185                                             end
186
187          1                422               always_comb begin
188                                                     a_rd_ptr_threshold: assert final(rd_ptr < fifo_if.FIFO_DEPTH);
189                                             end
190
191          1                525               always_comb begin
```

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

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

Toggle Coverage for instance /FIFO_top/dut --

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

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

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

```
=================================================================================
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         86        86         0   100.00%

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

Toggle Coverage for instance /FIFO_top/fifo_if --

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

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

Toggle Coverage      =     100.00% (86 of 86 bins)
```