UVM Final Project

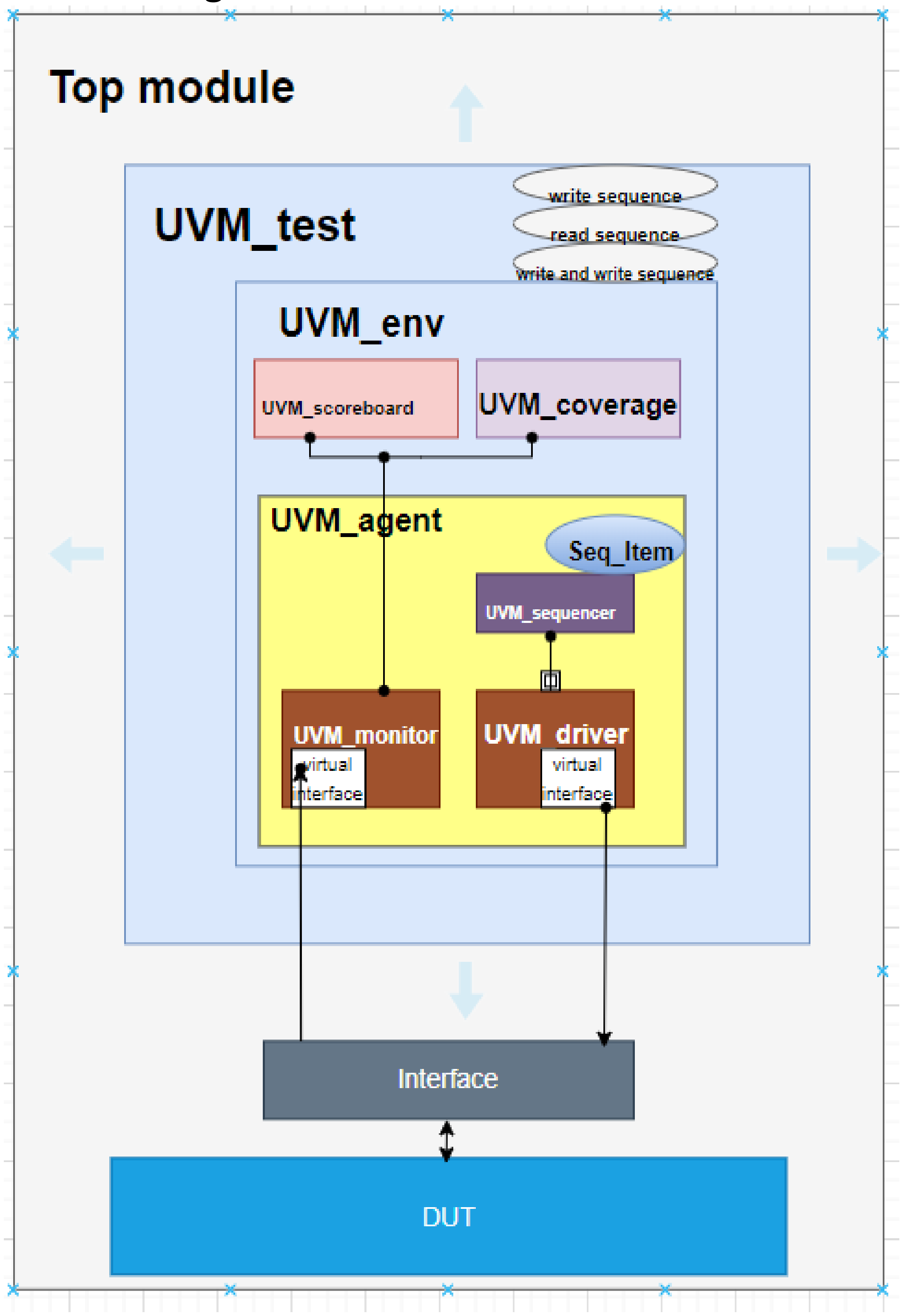
Synchronous FIFO

By: Amr Ahmed Abdelmoneim
Group 2

1. Verification plan

Section	Details
1. Introduction	**Design**: FIFO with configurable depth and width. **Objective**: Verify FIFO behavior (read/write, flags, overflow/underflow) using UVM.
2. Features to Verify	- Write/Read operations. - Full, empty, almostfull, almostempty flags. - Overflow and underflow. - Depth management.
3. Testbench Architecture	- **UVM Agent**: **Driver**: Drives transactions. **Monitor**: Monitors signals for coverage. **Sequencer**: • write_only_sequence • read_only_sequence • write_read_sequence. - **Scoreboard**: Compares DUT with reference model. - **Coverage collector** - **UVM Interface**: Connects DUT to UVM components.
4. Stimulus Strategy	- **Constrained-random stimulus**: Random write/read signals and data inputs. - **Directed stimulus**: Write until full (verify overflow). Read until empty (verify underflow).
5. Checker Strategy	- **Scoreboard**: Compare actual vs. expected (golden model). - **Assertions**: Flag assertions (full, empty). Overflow/underflow assertions. - **Error checking**: Ensure no data loss.
6. Coverage Strategy	- **Functional coverage**: All `wr_en` and `rd_en` combinations. Flag transitions (full, empty, etc.). - **Code coverage**: statment, branch, Toggle coverage.
7. Assertions	- Full/empty flag assertions. - Overflow/underflow based on pointers and control signals.
8. Test Plan	- **Basic tests**: Write and read sequence. Write only sequence Read only sequence Verify flags with directed tested and assertions
9. Simulation Plan	- UVM simulation with scoreboard and reference model. - Run for random, directed, and corner case scenarios.

2. UVM Structure using draw.io:

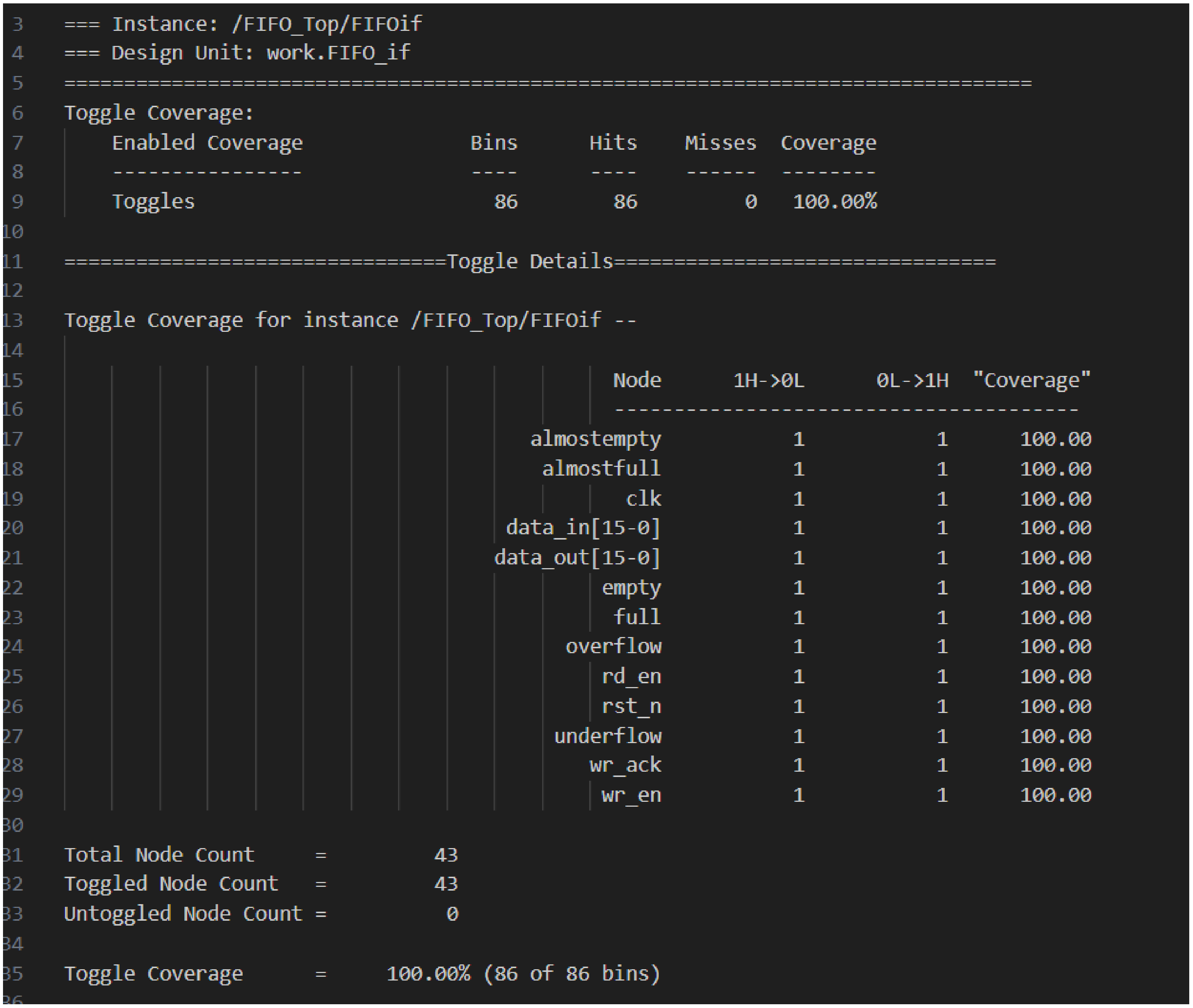


1. **Top Module** instantiates the DUT and connects the **interface**.
2. **UVM Test** generates sequences (stimulus) and passes them to the **UVM Environment**.
3. The **UVM Sequencer** controls the flow of sequence items to the **UVM Driver**.
4. The **UVM Driver** converts sequence items into pin-level signals to drive the DUT through the **interface**.
5. The **UVM Monitor** captures the signal activities and converts them into transactions.
6. **UVM Scoreboard** compares actual outputs against expected results to check for correctness.
7. **UVM Coverage** collects metrics to ensure all scenarios are tested.

Each component works together in the testbench to verify the functionality of the DUT, flagging any errors and providing comprehensive coverage metrics for verification completeness.

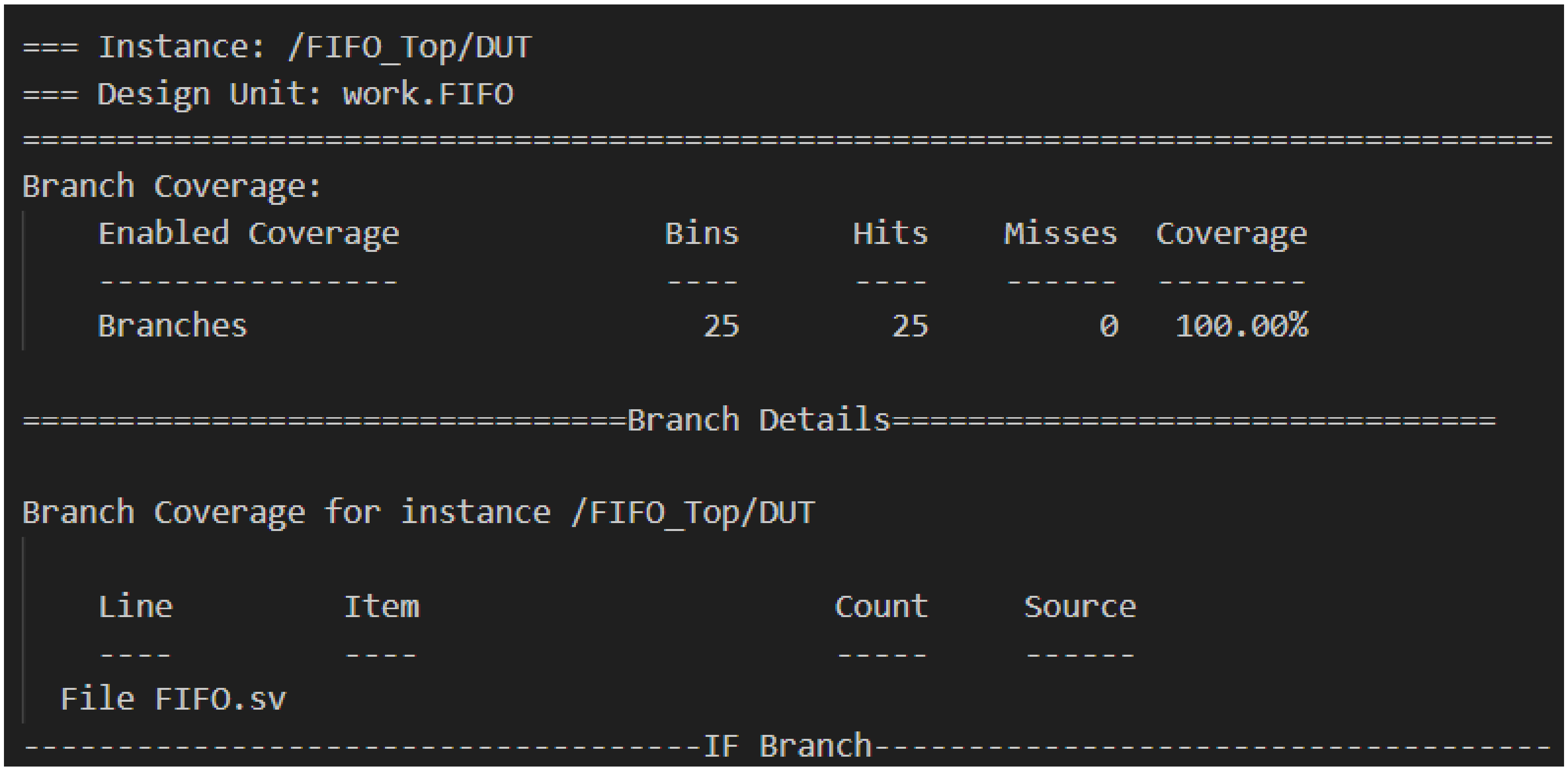
3. Code Coverage

1.Toogle Coverage:



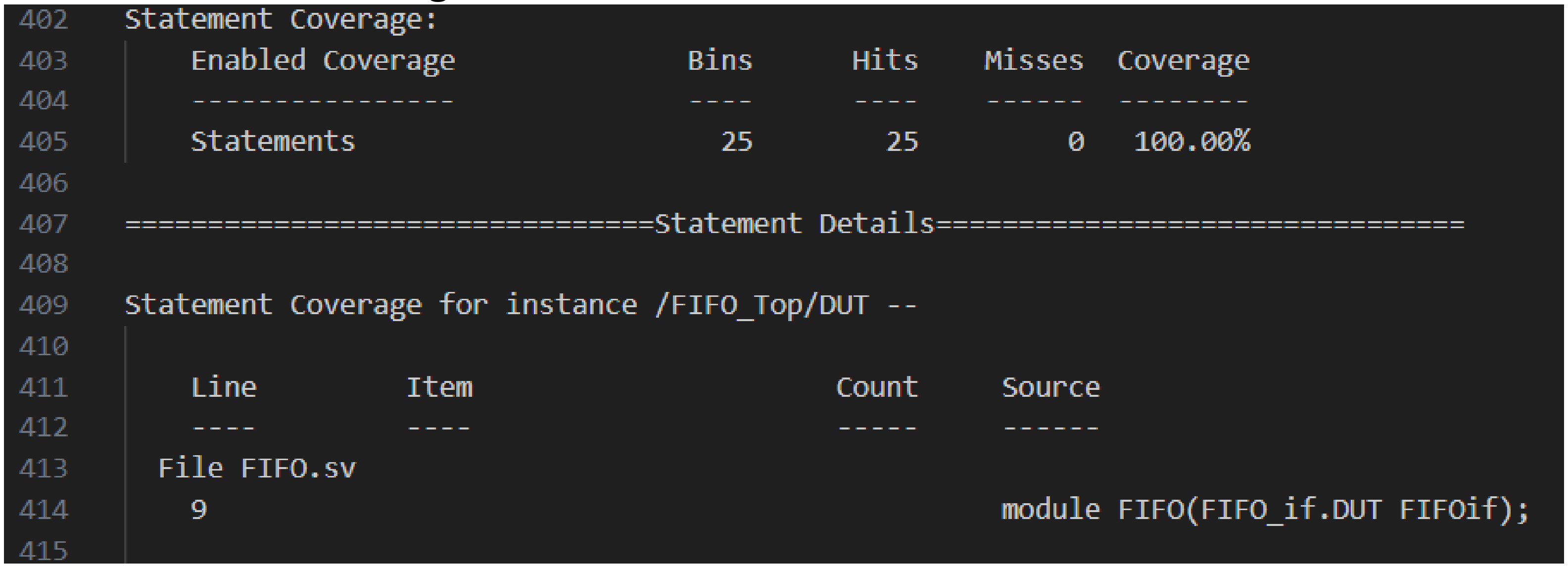
```
=== Instance: /FIFO_Top/FIFOif
=== Design Unit: work.FIFO_if
==================================================================================
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         86        86         0   100.00%

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

Toggle Coverage for instance /FIFO_Top/FIFOif --

                                              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)
```

2. Branch Coverage:



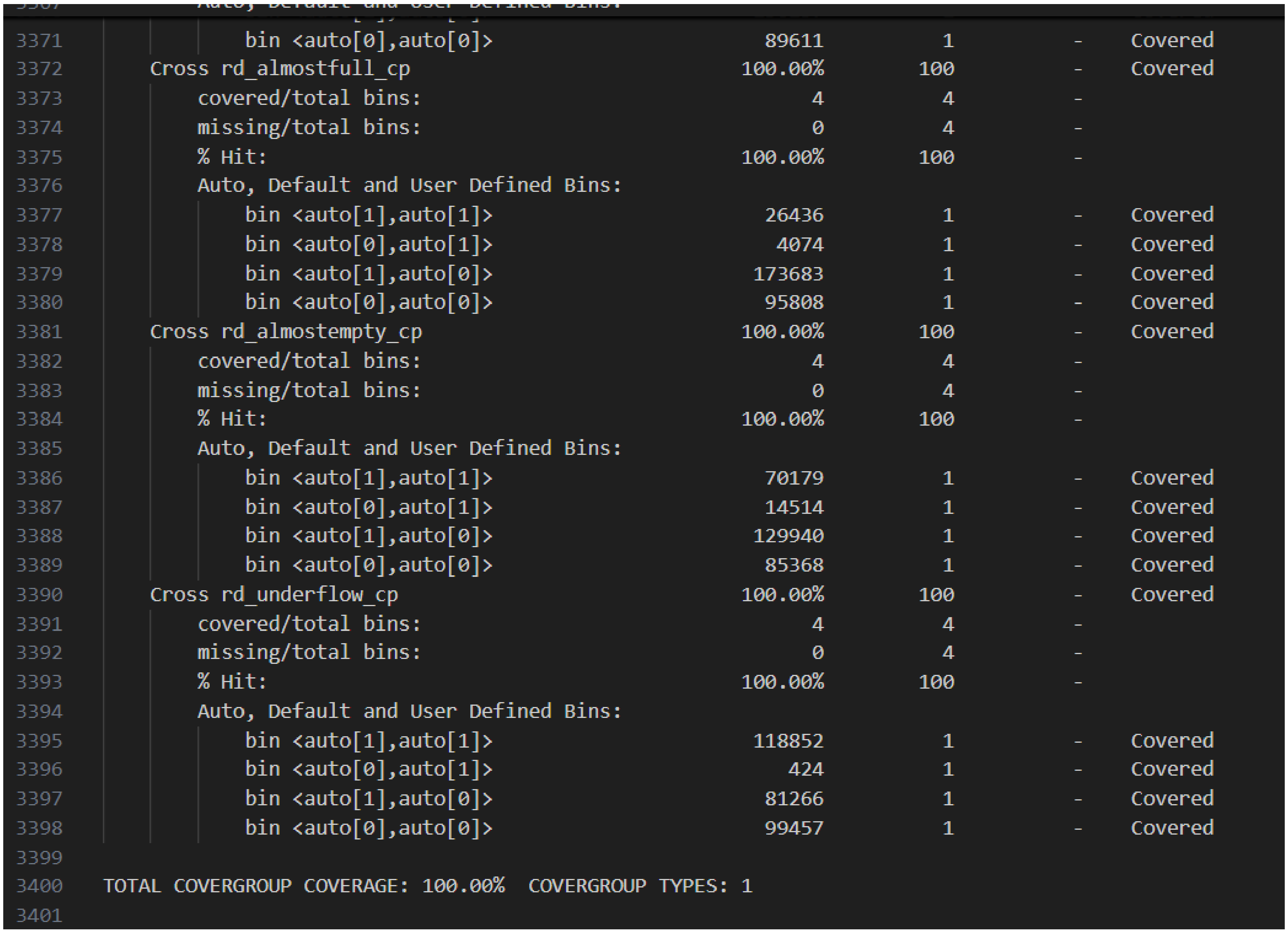
```
=== Instance: /FIFO_Top/DUT
=== Design Unit: work.FIFO
=====================================================================================
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        25        25         0   100.00%

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

Branch Coverage for instance /FIFO_Top/DUT

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File FIFO.sv
------------------------------------IF Branch------------------------------------
```

3. Statment Coverage:



```
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                      25        25         0   100.00%

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

Statement Coverage for instance /FIFO_Top/DUT --

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File FIFO.sv
    9                                                module FIFO(FIFO_if.DUT FIFOif);
```

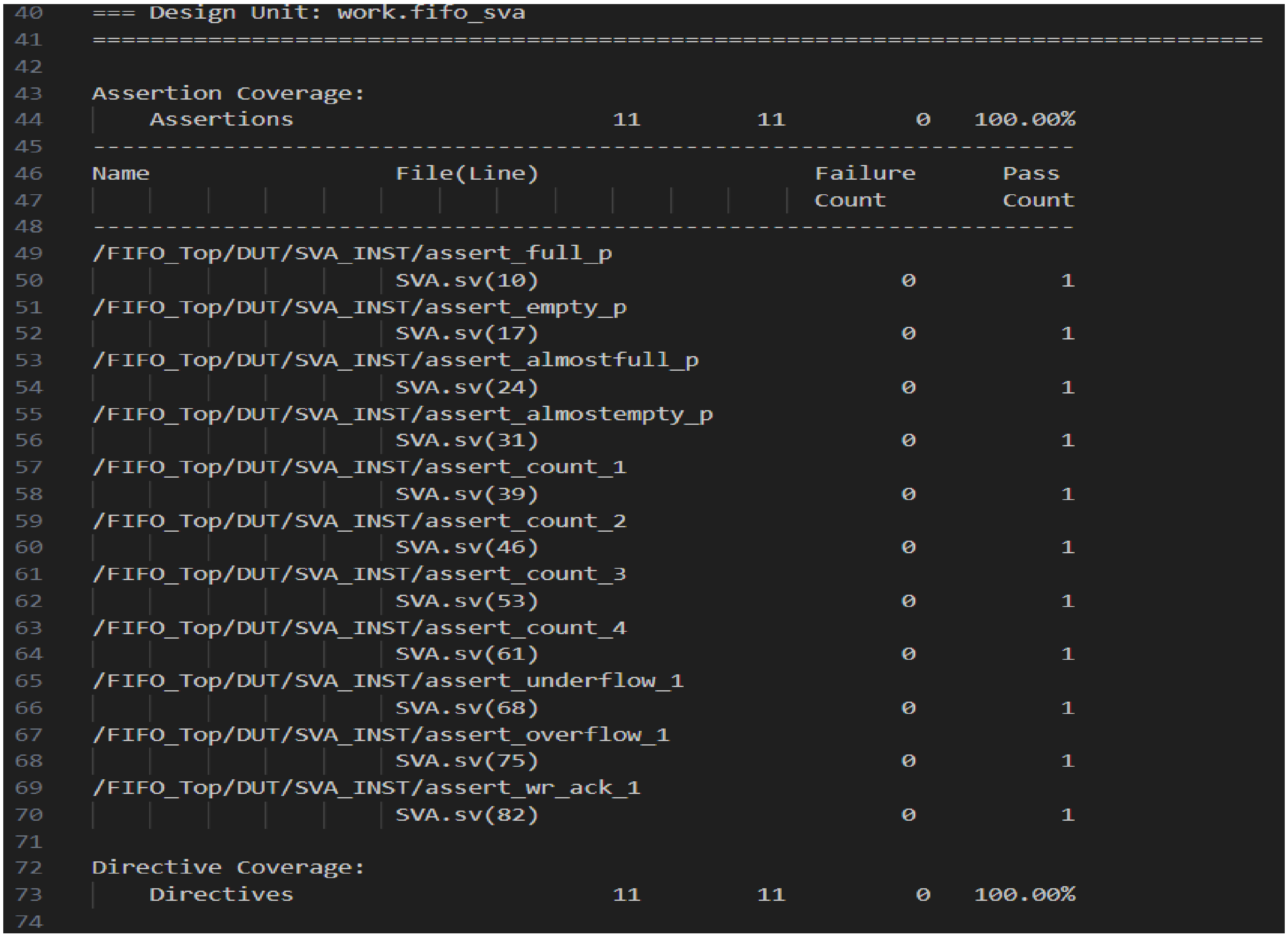
4. Functional Coverage



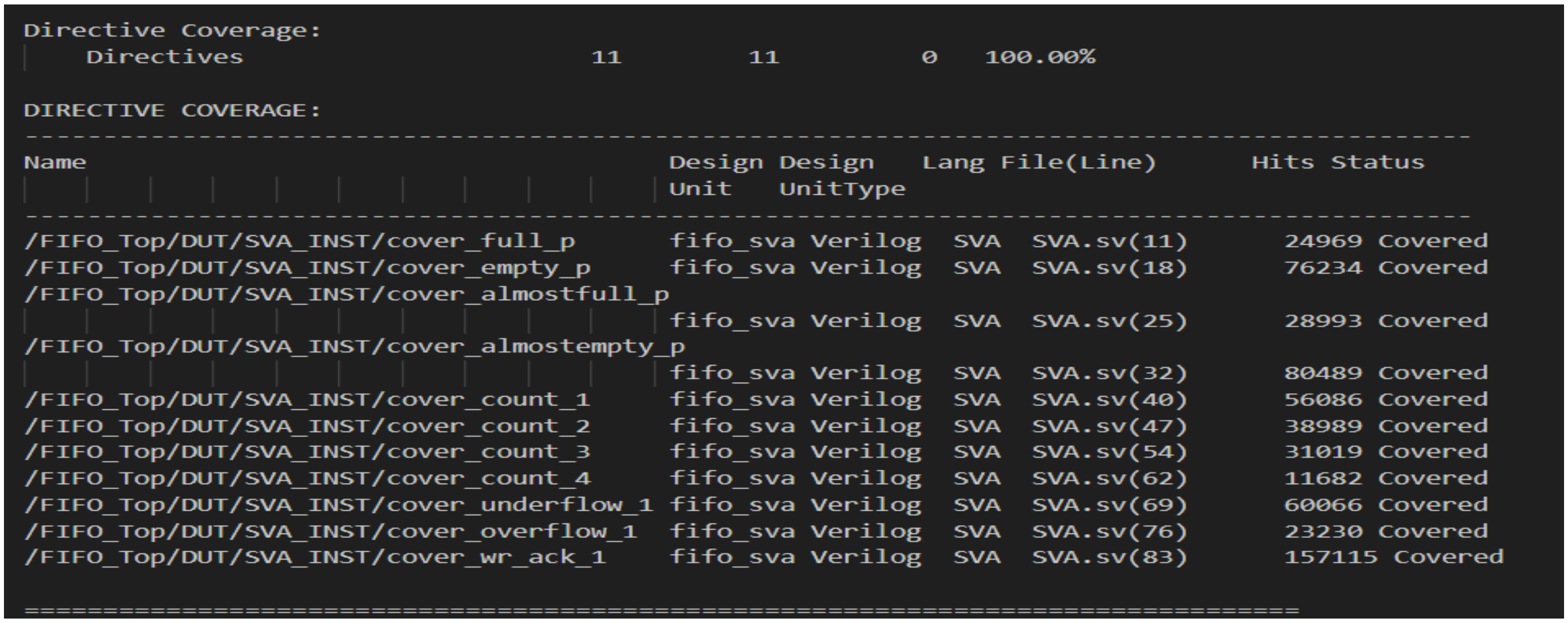
```
            bin <auto[0],auto[0]>                      89611          1          -    Covered
    Cross rd_almostfull_cp                            100.00%        100         -    Covered
        covered/total bins:                                4          4          -
        missing/total bins:                                0          4          -
        % Hit:                                        100.00%        100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1]>                      26436          1          -    Covered
            bin <auto[0],auto[1]>                       4074          1          -    Covered
            bin <auto[1],auto[0]>                     173683          1          -    Covered
            bin <auto[0],auto[0]>                      95808          1          -    Covered
    Cross rd_almostempty_cp                           100.00%        100         -    Covered
        covered/total bins:                                4          4          -
        missing/total bins:                                0          4          -
        % Hit:                                        100.00%        100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1]>                      70179          1          -    Covered
            bin <auto[0],auto[1]>                      14514          1          -    Covered
            bin <auto[1],auto[0]>                     129940          1          -    Covered
            bin <auto[0],auto[0]>                      85368          1          -    Covered
    Cross rd_underflow_cp                             100.00%        100         -    Covered
        covered/total bins:                                4          4          -
        missing/total bins:                                0          4          -
        % Hit:                                        100.00%        100         -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1]>                     118852          1          -    Covered
            bin <auto[0],auto[1]>                        424          1          -    Covered
            bin <auto[1],auto[0]>                      81266          1          -    Covered
            bin <auto[0],auto[0]>                      99457          1          -    Covered

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1
```

5. **Assertions**:

Feature	**Assertion**
Whenever the FIFO count equals FIFO_DEPTH, the FIFO is full	@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) (count == FIFO_DEPTH)
Whenever the FIFO count is 0, the FIFO is empty	@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) (count == 0).
Whenever the FIFO count equals FIFO_DEPTH-1, almost full is high	@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) (count == FIFO_DEPTH-1)
Whenever the FIFO count equals 1, almost empty is high	@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) (count == 1)
On write enable, count increments if not full	@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) ({FIFOif.wr_en, FIFOif.rd_en} == 2'b10 && !FIFOif.full).
On read enable, count decrements if not empty	`@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) ({FIFOif.wr_en, FIFOif.rd_en} == 2'b01 && !FIFOif.empty).
On simultaneous write and read with FIFO empty, count increments	@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) ({FIFOif.wr_en, FIFOif.rd_en} == 2'b11 && FIFOif.empty).
On simultaneous write and read with FIFO full, count decrements	@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) ({FIFOif.wr_en, FIFOif.rd_en} == 2'b11 && FIFOif.full)
FIFO underflow when attempting to read from an empty FIFO	@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) (FIFOif.empty && FIFOif.rd_en && count == 0).
FIFO overflow when attempting to write to a full FIFO	@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) (FIFOif.full && FIFOif.wr_en && !(count < FIFO_DEPTH))
Write acknowledgment is asserted when write enable and FIFO not full	@(posedge FIFOif.clk) disable iff (!FIFOif.rst_n) (FIFOif.wr_en && count < FIFO_DEPTH)

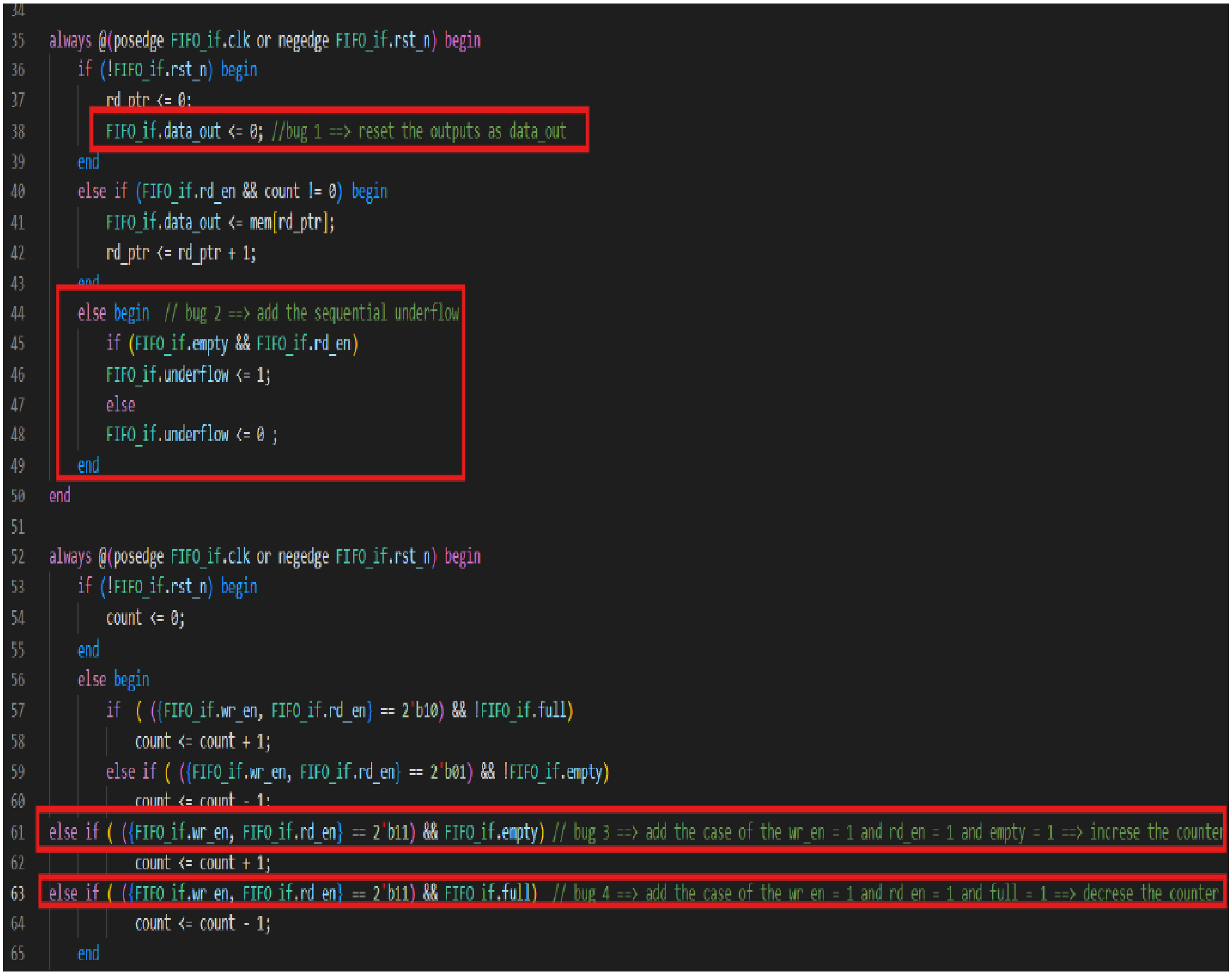


```
=== Design Unit: work.fifo_sva
=====================================================================================

Assertion Coverage:
    Assertions                      11         11          0    100.00%
--------------------------------------------------------------------
Name                    File(Line)                   Failure      Pass
                                                     Count        Count
--------------------------------------------------------------------
/FIFO_Top/DUT/SVA_INST/assert_full_p
                        SVA.sv(10)                         0            1
/FIFO_Top/DUT/SVA_INST/assert_empty_p
                        SVA.sv(17)                         0            1
/FIFO_Top/DUT/SVA_INST/assert_almostfull_p
                        SVA.sv(24)                         0            1
/FIFO_Top/DUT/SVA_INST/assert_almostempty_p
                        SVA.sv(31)                         0            1
/FIFO_Top/DUT/SVA_INST/assert_count_1
                        SVA.sv(39)                         0            1
/FIFO_Top/DUT/SVA_INST/assert_count_2
                        SVA.sv(46)                         0            1
/FIFO_Top/DUT/SVA_INST/assert_count_3
                        SVA.sv(53)                         0            1
/FIFO_Top/DUT/SVA_INST/assert_count_4
                        SVA.sv(61)                         0            1
/FIFO_Top/DUT/SVA_INST/assert_underflow_1
                        SVA.sv(68)                         0            1
/FIFO_Top/DUT/SVA_INST/assert_overflow_1
                        SVA.sv(75)                         0            1
/FIFO_Top/DUT/SVA_INST/assert_wr_ack_1
                        SVA.sv(82)                         0            1

Directive Coverage:
    Directives                      11         11          0    100.00%

```



```
Directive Coverage:
    Directives                      11         11          0    100.00%

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                        Design Design   Lang File(Line)      Hits Status
                                            Unit   UnitType
--------------------------------------------------------------------------------------------
/FIFO_Top/DUT/SVA_INST/cover_full_p         fifo_sva Verilog  SVA  SVA.sv(11)      24969 Covered
/FIFO_Top/DUT/SVA_INST/cover_empty_p        fifo_sva Verilog  SVA  SVA.sv(18)      76234 Covered
/FIFO_Top/DUT/SVA_INST/cover_almostfull_p
                                            fifo_sva Verilog  SVA  SVA.sv(25)      28993 Covered
/FIFO_Top/DUT/SVA_INST/cover_almostempty_p
                                            fifo_sva Verilog  SVA  SVA.sv(32)      80489 Covered
/FIFO_Top/DUT/SVA_INST/cover_count_1        fifo_sva Verilog  SVA  SVA.sv(40)      56086 Covered
/FIFO_Top/DUT/SVA_INST/cover_count_2        fifo_sva Verilog  SVA  SVA.sv(47)      38989 Covered
/FIFO_Top/DUT/SVA_INST/cover_count_3        fifo_sva Verilog  SVA  SVA.sv(54)      31019 Covered
/FIFO_Top/DUT/SVA_INST/cover_count_4        fifo_sva Verilog  SVA  SVA.sv(62)      11682 Covered
/FIFO_Top/DUT/SVA_INST/cover_underflow_1 fifo_sva Verilog  SVA  SVA.sv(69)      60066 Covered
/FIFO_Top/DUT/SVA_INST/cover_overflow_1  fifo_sva Verilog  SVA  SVA.sv(76)      23230 Covered
/FIFO_Top/DUT/SVA_INST/cover_wr_ack_1       fifo_sva Verilog  SVA  SVA.sv(83)      157115 Covered

================================================================================
```

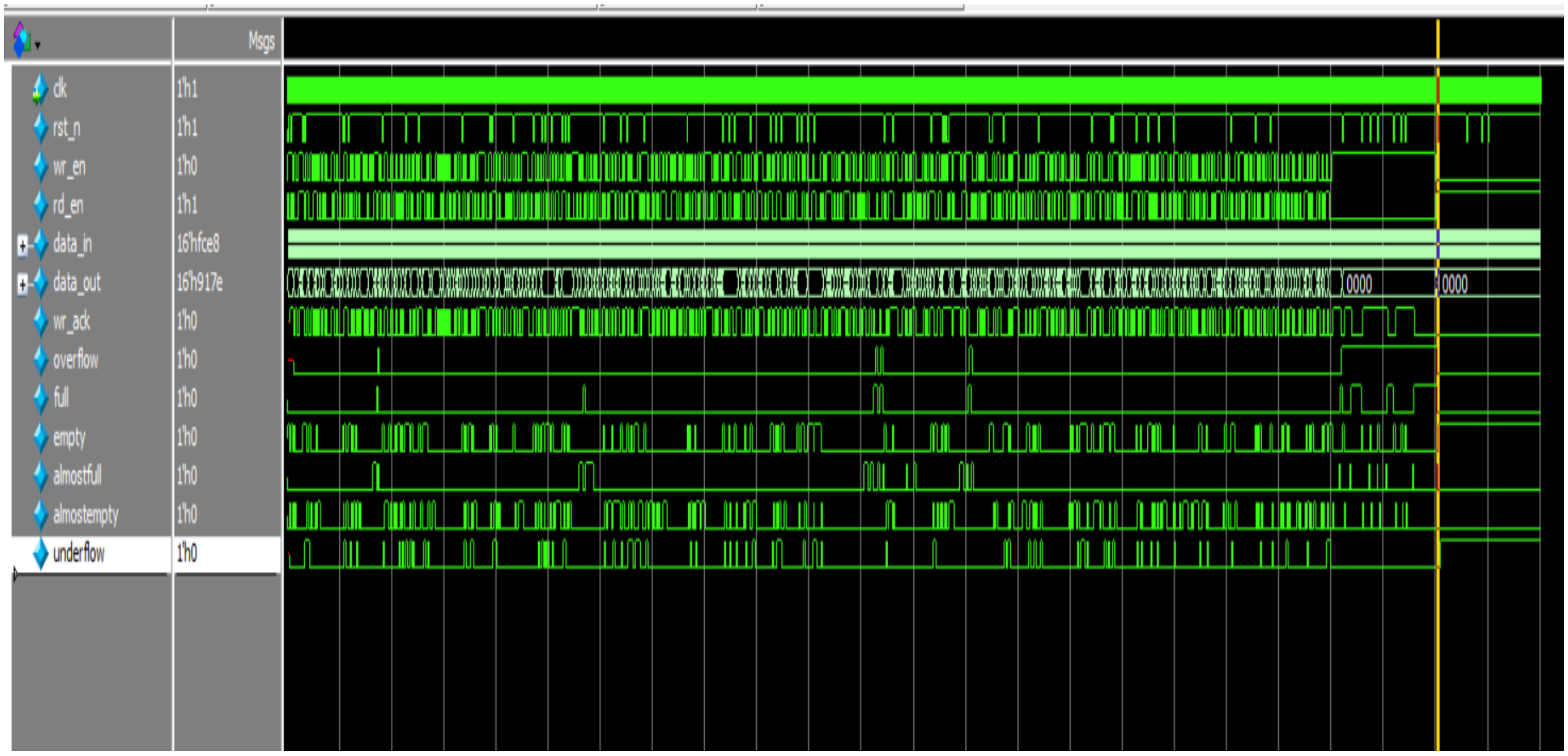
6. BUG Report:



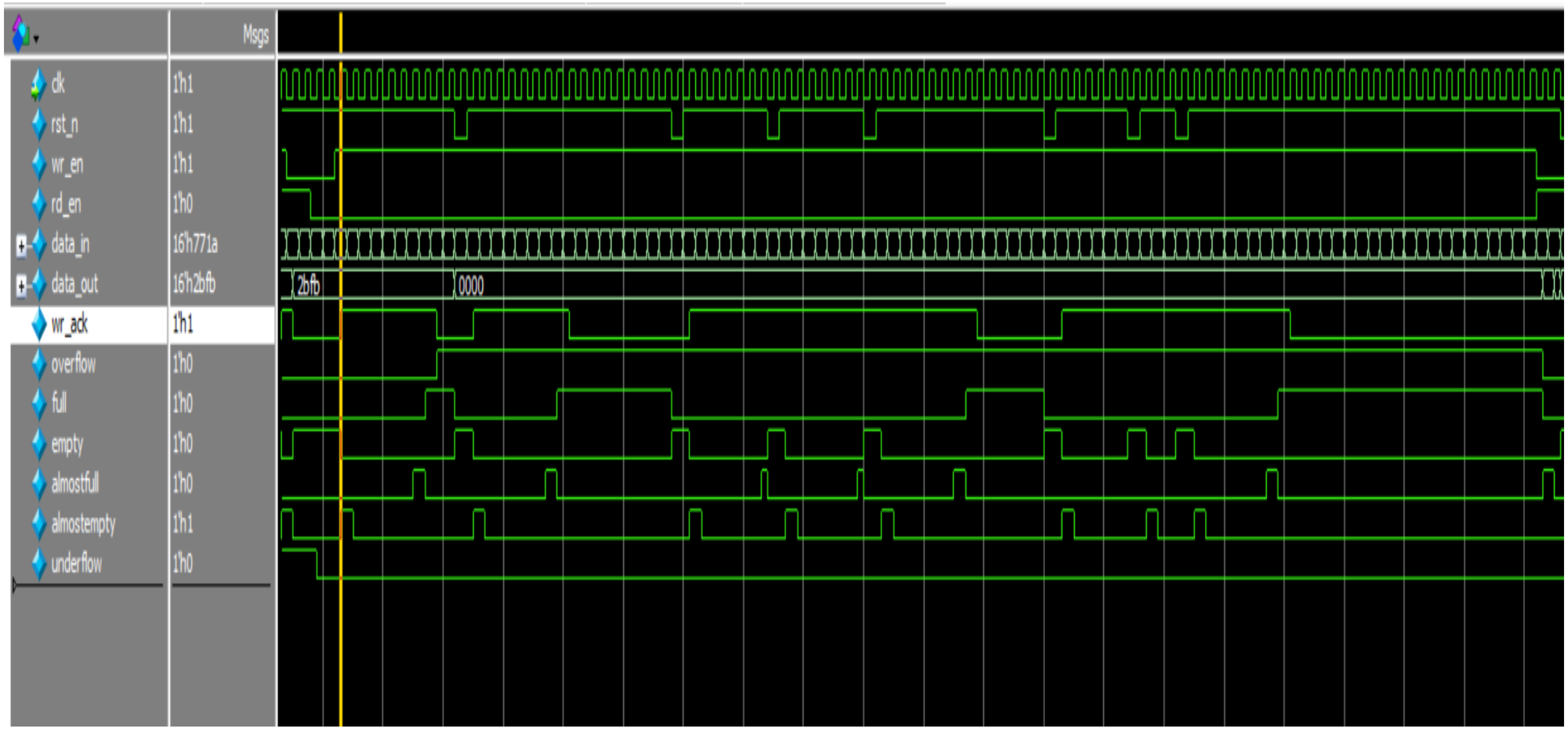
```
always @(posedge FIFO_if.clk or negedge FIFO_if.rst_n) begin
    if (!FIFO_if.rst_n) begin
        rd_ptr <= 0;
        FIFO_if.data_out <= 0; //bug 1 ==> reset the outputs as data_out
    end
    else if (FIFO_if.rd_en && count != 0) begin
        FIFO_if.data_out <= mem[rd_ptr];
        rd_ptr <= rd_ptr + 1;
    end
    else begin  // bug 2 ==> add the sequential underflow
        if (FIFO_if.empty && FIFO_if.rd_en)
        FIFO_if.underflow <= 1;
        else
        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.empty) // bug 3 ==> add the case of the wr_en = 1 and rd_en = 1 and empty = 1 ==> increse the counter
            count <= count + 1;
else if ( ({FIFO_if.wr_en, FIFO_if.rd_en} == 2'b11) && FIFO_if.full)  // bug 4 ==> add the case of the wr_en = 1 and rd_en = 1 and full = 1 ==> decrese the counter
            count <= count - 1;
    end
```

7. Waveform for each sequence:

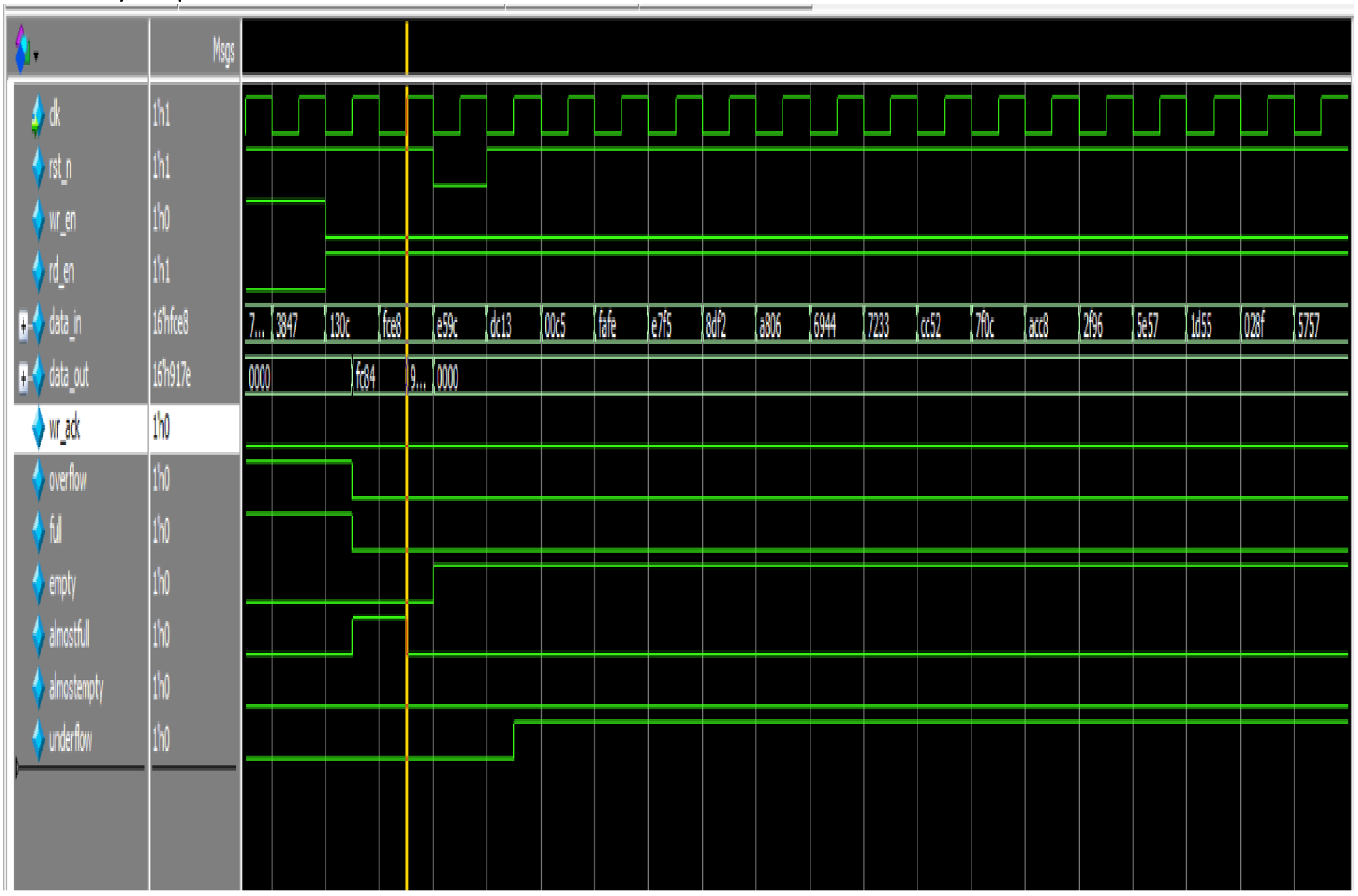
Wave form indicates that first sequence is write and read, in second sequence is write enable is always “1” and in third sequence read enable is always “1”:



Write only sequence waveform:



Read only sequence waveform:


Msgs
clk 1'h1
rst_n 1'h1
wr_en 1'h0
rd_en 1'h1
data_in 16'hfce8
7... 3847 130c fce8 e59c dc13 00c5 fafe e7f5 8df2 a806 6944 7233 cc52 7f0c acc8 2f96 5e57 1d55 028f 5757
data_out 16'h917e
0000 fc84 9... 0000
wr_ack 1'h0
overflow 1'h0
full 1'h0
empty 1'h0
almostfull 1'h0
almostempty 1'h0
underflow 1'h0