Synchronous FIFO

Digital Verification Diploma

Under Supervision of **Eng. Kareem Waseem**

Prepared by **Muhammed Yasser**

Sept 30, 2025

Introduction

Reliable data buffering is foundational to the stability and performance of modern digital systems. The Synchronous First-In, First-Out (FIFO) buffer is a ubiquitous component used to safely manage data flow between blocks operating under a unified clock domain.

This project focused exclusively on the Verification of a provided, parameterizable Synchronous FIFO RTL implementation. The primary objective was to ensure the design's functional correctness and robustness to specifications through a rigorous, industry-standard verification methodology.

To achieve this, a SystemVerilog (SV) environment was developed. This environment facilitated the implementation of a Coverage-Driven Verification (CDV) flow, featuring constrained random stimulus generation, a golden reference model (Scoreboard) for checking data integrity, and a comprehensive set of functional and code coverage metrics. The verification effort was successfully executed to achieve a target of 100% functional and code coverage, rigorously testing all data path, control logic, boundary conditions, and error states (e.g., overflow and underflow) of the FIFO design.

RTL

Interface Code Snippets

FIFO_if.sv

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

    // Monitor modport
    modport monitor (
        input clk, rst_n, wr_en, rd_en,
        input data_in, data_out,
        input wr_ack, overflow,
        input full, empty, almostfull, almostempty, underflow
    );

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

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

RTL Code Snippets

FIFO.sv

```
////////////////////////////////////////////////////////////////////////////////
// Author: Kareem Waseem
// Course: Digital Verification using SV & UVM
//
// Description: FIFO Design
//
////////////////////////////////////////////////////////////////////////////////
module FIFO(FIFO_if f_if);

localparam max_fifo_addr = $clog2(f_if.FIFO_DEPTH);

reg [f_if.FIFO_WIDTH-1:0] mem [f_if.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.overflow <= 0;
        f_if.wr_ack <= 0;
    end
    else if (f_if.wr_en && count < f_if.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)
            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
        rd_ptr <= 0;
        f_if.underflow <= 0;
    end
    else if (f_if.rd_en && count != 0) begin
```

FIFO.sv

```
always @(posedge f_if.clk or negedge f_if.rst_n) begin
    end
    else if (f_if.rd_en && count != 0) begin
        f_if.data_out <= mem[rd_ptr];
        rd_ptr <= rd_ptr + 1;
    end
    else begin
        if (f_if.empty & f_if.rd_en)
            f_if.underflow <= 1;
        else
            f_if.underflow <= 0;
    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
        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'b11) && f_if.empty)
            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;
    end
end

assign f_if.full = (count == f_if.FIFO_DEPTH)? 1 : 0; //first mistake
assign f_if.empty = (count == 0)? 1 : 0;
assign f_if.almostfull = (count == f_if.FIFO_DEPTH-1)? 1 : 0;
assign f_if.almostempty = (count == 1)? 1 : 0;

`ifdef SIM
//Assertions
//a
a_resetBehaviour: assert property (@(posedge f_if.clk) !f_if.rst_n |-> !wr_ptr && !rd_ptr && !count);
a_resetBehaviour_cvr: cover property (@(posedge f_if.clk) !f_if.rst_n |-> !wr_ptr && !rd_ptr && !count);
//b
a_wr_ack: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n) f_if.wr_en && !f_if.full  |-> ##1 f_if.wr_ack);
```

FIFO.sv

```
`ifdef SIM
//Assertions
//a
//FIFO_1
a_resetBehaviour: assert property (@(posedge f_if.clk) !f_if.rst_n |-> !wr_ptr && !rd_ptr && !count);
a_resetBehaviour_cvr: cover property (@(posedge f_if.clk) !f_if.rst_n |-> !wr_ptr && !rd_ptr && !count);
//b
//FIFO_3//FIFO_4//FIFO_5//FIFO_10//FIFO_11//FIFO_12
a_wr_ack: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n) f_if.wr_en && !f_if.full  |-> ##1 f_if.wr_ack);
a_wr_ack_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n) f_if.wr_en && !f_if.full  |-> ##1 f_if.wr_ack);
//c
//FIFO_5, FIFO_12
a_overflow: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)  f_if.full && f_if.wr_en |-> ##1 f_if.overflow);
a_overflow_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)   f_if.full && f_if.wr_en |-> ##1 f_if.overflow);
//d
//FIFO_9, FIFO_11
a_underflow: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)     f_if.empty && f_if.rd_en |->##1 f_if.underflow);
a_underflow_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)  f_if.empty && f_if.rd_en |->##1 f_if.underflow);
//e
//FIFO_8, FIFO_9
a_empty: assert property (@(posedge f_if.clk)   !count |-> f_if.empty);
a_empty_cvr: cover property (@(posedge f_if.clk)    !count |-> f_if.empty);
//f
//FIFO_4, FIFO_5
a_full: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)  count == f_if.FIFO_DEPTH |-> f_if.full);
a_full_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)   count == f_if.FIFO_DEPTH |-> f_if.full);
//g
//FIFO_3, FIFO_12
a_almostfull: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)    count == f_if.FIFO_DEPTH-1 |-> f_if.almostfull);
a_almostfull_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)     count == f_if.FIFO_DEPTH-1 |-> f_if.almostfull);
//h
//FIFO_11, FIFO_7
a_almostempty: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)   count == 1 |-> f_if.almostempty);
a_almostempty_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)    count == 1 |-> f_if.almostempty);

//i
//read pointer
a_rd_ptr_wrap: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n) rd_ptr == f_if.FIFO_DEPTH-1 && f_if.rd_en && !f_if.empty |-> ##1 rd_ptr == 0);
a_rd_ptr_wrap_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n) rd_ptr == f_if.FIFO_DEPTH-1 && f_if.rd_en && !f_if.empty |-> ##1 rd_ptr == 0);
//write pointer
a_wr_ptr_wrap: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n) wr_ptr == f_if.FIFO_DEPTH-1 && f_if.wr_en && !f_if.full |-> ##1 wr_ptr == 0);
a_wr_ptr_wrap_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n) wr_ptr == f_if.FIFO_DEPTH-1 && f_if.wr_en && !f_if.full |-> ##1 wr_ptr == 0);

//j
always_comb begin
    if (f_if.rst_n) begin
        a_rd_ptr_thr: assert final (rd_ptr < f_if.FIFO_DEPTH);
        a_wr_ptr_thr: assert final (wr_ptr < f_if.FIFO_DEPTH);
        a_count_thr: assert final (count <= f_if.FIFO_DEPTH);
    end
end

`endif
endmodule
```

Bug Fixed #1

- The almostfull flag condition was wrong

Before

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

After

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

Bug Fixed #2

- Not handling the counter value when the wr_en = 1 and rd_en = 1

Before

```
45  always @(posedge f_if.clk or negedge f_if.rst_n) begin
46      if (!f_if.rst_n) begin
47          count <= 0;
48      end
49      else begin
50          if  ( ({f_if.wr_en, f_if.rd_en} == 2'b10) && !f_if.full)
51              count <= count + 1;
52          else if ( ({f_if.wr_en, f_if.rd_en} == 2'b01) && !f_if.empty)
53              count <= count - 1;
54      end
55  end
```

After

```
54  always @(posedge f_if.clk or negedge f_if.rst_n) begin
55      if (!f_if.rst_n) begin
56          count <= 0;
57      end
58      else begin
59          if  ( ({f_if.wr_en, f_if.rd_en} == 2'b11) && f_if.full)
60              count <= count - 1;
61          else if ( ({f_if.wr_en, f_if.rd_en} == 2'b11) && f_if.empty)
62              count <= count + 1;
63          else if ( ({f_if.wr_en, f_if.rd_en} == 2'b10) && !f_if.full)
64              count <= count + 1;
65          else if ( ({f_if.wr_en, f_if.rd_en} == 2'b01) && !f_if.empty)
66              count <= count - 1;
67      end
68  end
```

Bug Fixed #3

- The underflow flag logic was wrong, it should be sequential not combinational

Before

```
57  assign f_if.full = (count == f_if.FIFO_DEPTH)? 1 : 0;
58  assign f_if.empty = (count == 0)? 1 : 0;
59  assign f_if.underflow = (f_if.empty && f_if.rd_en)? 1 : 0;
60  assign f_if.almostfull = (count == f_if.FIFO_DEPTH-2)? 1 : 0;
61  assign f_if.almostempty = (count == 1)? 1 : 0;
```

After

```
37  always @(posedge f_if.clk or negedge f_if.rst_n) begin
38      if (!f_if.rst_n) begin
39          rd_ptr <= 0;
40          f_if.underflow <= 0;
41      end
42      else if (f_if.rd_en && count != 0) begin
43          f_if.data_out <= mem[rd_ptr];
44          rd_ptr <= rd_ptr + 1;
45      end
46      else begin
47          if (f_if.empty & f_if.rd_en)
48              f_if.underflow <= 1;
49          else
50              f_if.underflow <= 0;
51      end
52  end
```

Bug Fixed #4

- Not reseting the sequential flags when the reset is asserted

Before

```
17  always @(posedge f_if.clk or negedge f_if.rst_n) begin
18      if (!f_if.rst_n) begin
19          wr_ptr <= 0;
20      end
```

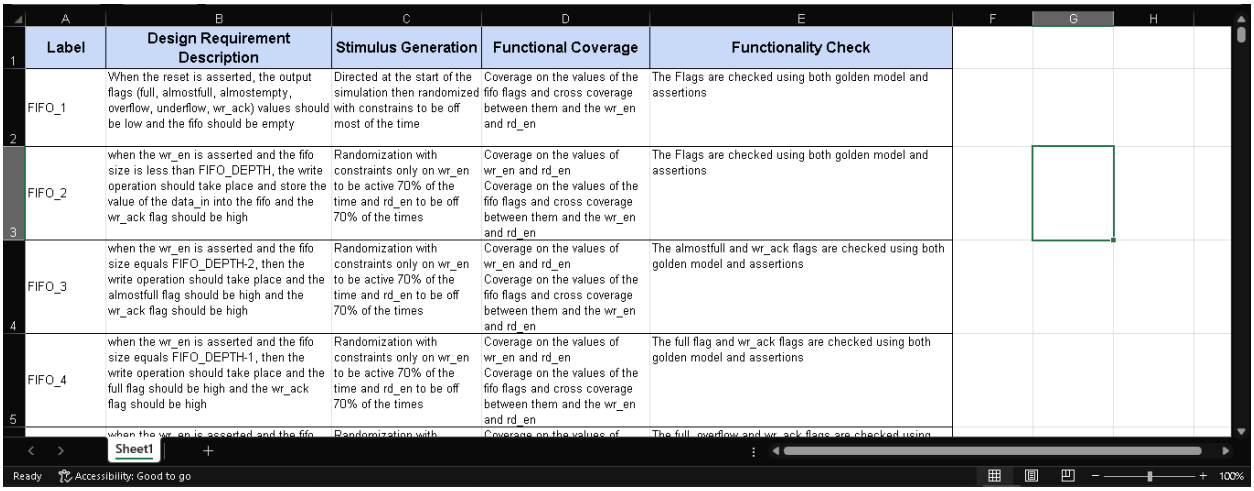
```
35  always @(posedge f_if.clk or negedge f_if.rst_n) begin
36      if (!f_if.rst_n) begin
37          rd_ptr <= 0;
38      end
```

After

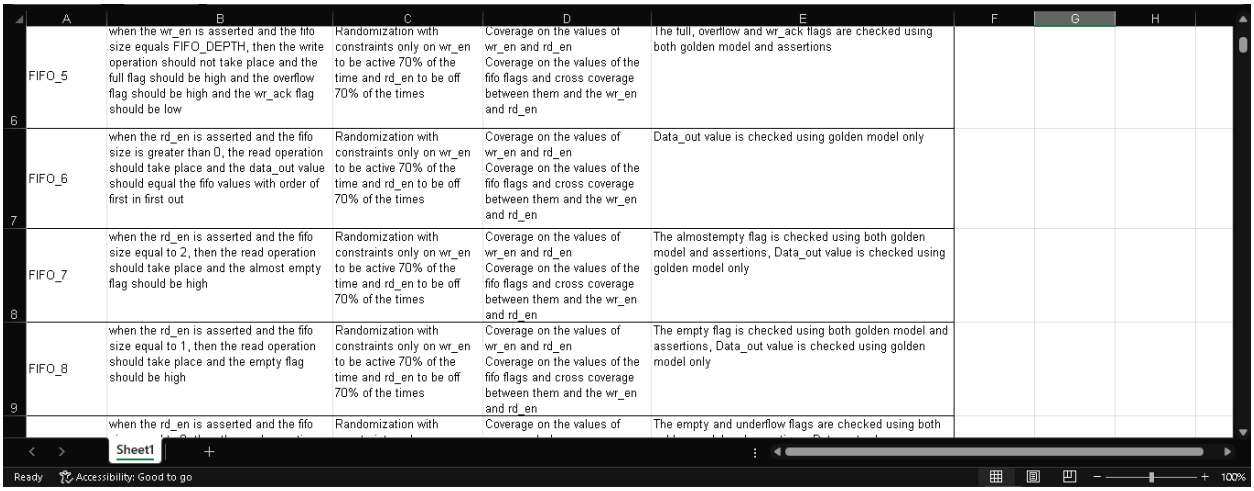
```
17  always @(posedge f_if.clk or negedge f_if.rst_n) begin
18      if (!f_if.rst_n) begin
19          wr_ptr <= 0;
20          f_if.overflow <= 0;
21          f_if.wr_ack <= 0;
22      end
```

```
37  always @(posedge f_if.clk or negedge f_if.rst_n) begin
38      if (!f_if.rst_n) begin
39          rd_ptr <= 0;
40          f_if.underflow <= 0;
41      end
```

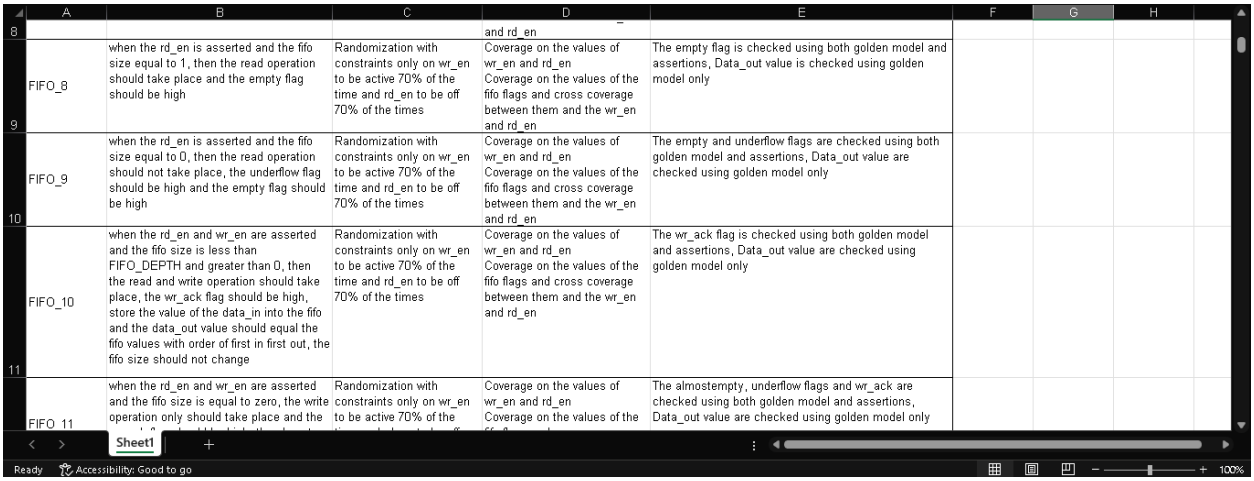
Verification Plan



Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
FIFO_1	When the reset is asserted, the output flags (full, almostfull, almostempty, overflow, underflow, wr_ack) values should be low and the fifo should be empty	Directed at the start of the simulation then randomized with constrains to be off most of the time	Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The Flags are checked using both golden model and assertions
FIFO_2	when the wr_en is asserted and the fifo size is less than FIFO_DEPTH, the write operation should take place and store the value of the data_in into the fifo and the wr_ack flag should be high	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The Flags are checked using both golden model and assertions
FIFO_3	when the wr_en is asserted and the fifo size equals FIFO_DEPTH-2, then the write operation should take place and the almostfull flag should be high and the wr_ack flag should be high	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The almostfull and wr_ack flags are checked using both golden model and assertions
FIFO_4	when the wr_en is asserted and the fifo size equals FIFO_DEPTH-1, then the write operation should take place and the full flag should be high and the wr_ack flag should be high	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The full flag and wr_ack flags are checked using both golden model and assertions



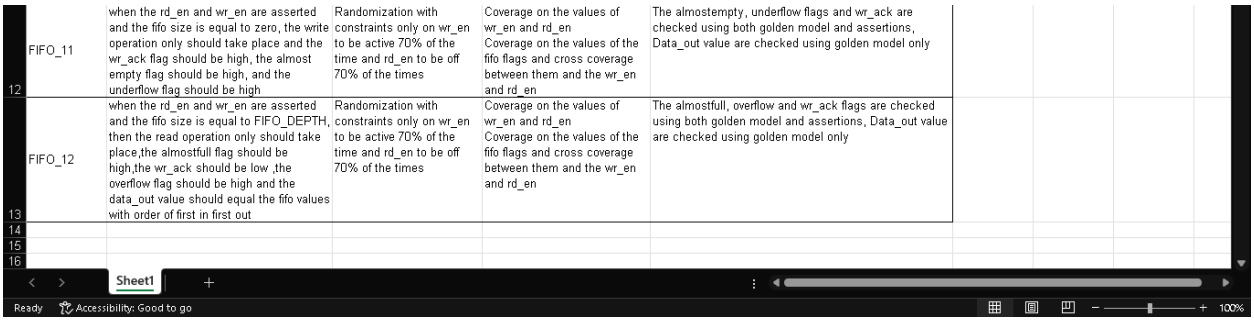
FIFO_5	when the wr_en is asserted and the fifo size equals FIFO_DEPTH, then the write operation should not take place and the full flag should be high and the overflow flag should be high and the wr_ack flag should be low	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The full, overflow and wr_ack flags are checked using both golden model and assertions
FIFO_6	when the rd_en is asserted and the fifo size is greater than 0, the read operation should take place and the data_out value should equal the fifo values with order of first in first out	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	Data_out value is checked using golden model only
FIFO_7	when the rd_en is asserted and the fifo size equal to 2, then the read operation should take place and the almost empty flag should be high	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The almostempty flag is checked using both golden model and assertions, Data_out value is checked using golden model only
FIFO_8	when the rd_en is asserted and the fifo size equal to 1, then the read operation should take place and the empty flag should be high	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The empty flag is checked using both golden model and assertions, Data_out value is checked using golden model only



	A	B	C	D	E
8				and rd_en	
9	FIFO_8	when the rd_en is asserted and the fifo size equal to 1, then the read operation should take place and the empty flag should be high	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The empty flag is checked using both golden model and assertions, Data_out value is checked using golden model only
10	FIFO_9	when the rd_en is asserted and the fifo size equal to 0, then the read operation should not take place, the underflow flag should be high and the empty flag should be high	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The empty and underflow flags are checked using both golden model and assertions, Data_out value are checked using golden model only
11	FIFO_10	when the rd_en and wr_en are asserted and the fifo size is less than FIFO_DEPTH and greater than 0, then the read and write operation should take place, the wr_ack flag should be high, store the value of the data_in into the fifo and the data_out value should equal the fifo values with order of first in first out, the fifo size should not change	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The wr_ack flag is checked using both golden model and assertions, Data_out value are checked using golden model only
	FIFO_11	when the rd_en and wr_en are asserted and the fifo size is equal to zero, the write operation only should take place and the	Randomization with constraints only on wr_en to be active 70% of the	Coverage on the values of wr_en and rd_en Coverage on the values of the	The almostempty, underflow flags and wr_ack are checked using both golden model and assertions, Data_out value are checked using golden model only

Sheet1

Ready Accessibility: Good to go 100%



	FIFO_11	when the rd_en and wr_en are asserted and the fifo size is equal to zero, the write operation only should take place and the wr_ack flag should be high, the almost empty flag should be high, and the underflow flag should be high	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The almostempty, underflow flags and wr_ack are checked using both golden model and assertions, Data_out value are checked using golden model only
12	FIFO_12	when the rd_en and wr_en are asserted and the fifo size is equal to FIFO_DEPTH, then the read operation only should take place,the almostfull flag should be high,the wr_ack should be low ,the overflow flag should be high and the data_out value should equal the fifo values with order of first in first out	Randomization with constraints only on wr_en to be active 70% of the time and rd_en to be off 70% of the times	Coverage on the values of wr_en and rd_en Coverage on the values of the fifo flags and cross coverage between them and the wr_en and rd_en	The almostfull, overflow and wr_ack flags are checked using both golden model and assertions, Data_out value are checked using golden model only
13					
14					
15					
16					

Sheet1

Ready Accessibility: Good to go 100%

Testbench

Shared Package Code Snippets

shared_pkg.sv

```
1  package shared_pkg;
2      logic test_finished;
3      int error_cnt, correct_cnt;
4      event emonitor;
5  endpackage
6
```

Top Code Snippets

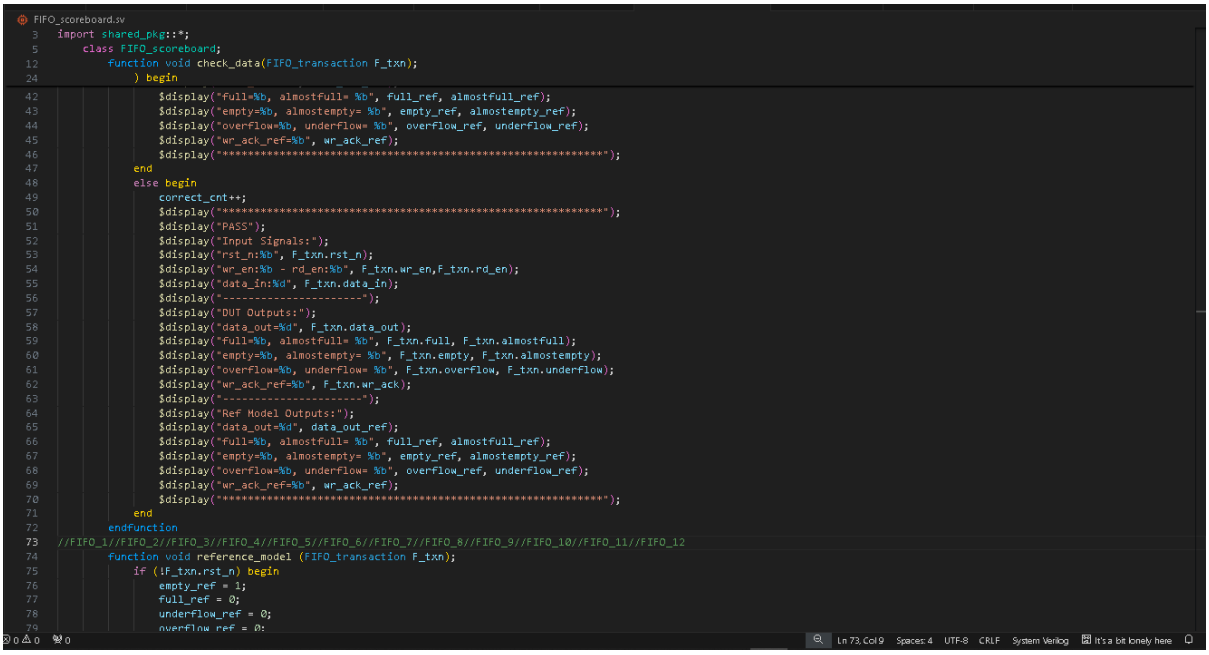
FIFO_top.sv

```
1  module top;
2      bit clk = 0;
3
4      initial forever #1 clk = ~clk;
5
6      FIFO_if f_if (clk);
7
8      FIFO DUT (f_if);
9
10     FIFO_tb TB (f_if);
11
12     FIFO_monitor MONITOR (f_if);
13
14 //FIFO_1
15     always_comb begin
16         if (!f_if.rst_n) begin
17             a_reset_1: assert final (f_if.empty && !f_if.full);
18             a_reset_2: assert final (!f_if.almostempty && !f_if.almostfull);
19             a_reset_3: assert final (!f_if.overflow && !f_if.underflow);
20         end
21     end
22
23 endmodule
```

Monitor Code Snippets

Scoreboard

```
FIFO_scoreboard.sv
package FIFO_scoreboard_pkg;
import FIFO_transaction_pkg::*;
import shared_pkg::*;

    class FIFO_scoreboard;
        logic [FIFO_WIDTH-1:0] data_out_ref;
        logic wr_ack_ref, overflow_ref;
        logic full_ref, empty_ref, almostfull_ref, almostempty_ref, underflow_ref;

        logic [FIFO_WIDTH-1:0] FIFO_refmodel [$];

        function void check_data(FIFO_transaction F_txn);
            reference_model(F_txn);

            if (
                (F_txn.data_out != data_out_ref) ||
                (F_txn.wr_ack != wr_ack_ref) ||
                (F_txn.overflow != overflow_ref) ||
                (F_txn.underflow != underflow_ref) ||
                (F_txn.full != full_ref) ||
                (F_txn.almostfull!= almostfull_ref) ||
                (F_txn.empty != empty_ref) ||
                (F_txn.almostempty != almostempty_ref)
            ) begin
                error_cnt++;
                $display("*************************************************************");
                $display("ERROR - DUT Out not equal Ref Model Out");
                $display("Input Signals:");
                $display("rst_n:%b", F_txn.rst_n);
                $display("wr_en:%b - rd_en:%b", F_txn.wr_en,F_txn.rd_en);
                $display("data_in:%d", F_txn.data_in);
                $display("----------------------");
                $display("DUT Outputs:");
                $display("data_out=%d", F_txn.data_out);
                $display("full=%b, almostfull= %b", F_txn.full, F_txn.almostfull);
                $display("empty=%b, almostempty= %b", F_txn.empty, F_txn.almostempty);
                $display("overflow=%b, underflow= %b", F_txn.overflow, F_txn.underflow);
                $display("wr_ack_ref=%b", F_txn.wr_ack);
                $display("----------------------");
                $display("Ref Model Outputs:");
                $display("data_out=%d", data_out_ref);
                $display("full=%b, almostfull= %b", full_ref, almostfull_ref);
```



```
FIFO_scoreboard.sv
import shared_pkg::*;
    class FIFO_scoreboard;
        function void check_data(FIFO_transaction F_txn);
            ) begin
                $display("full=%b, almostfull= %b", full_ref, almostfull_ref);
                $display("empty=%b, almostempty= %b", empty_ref, almostempty_ref);
                $display("overflow=%b, underflow= %b", overflow_ref, underflow_ref);
                $display("wr_ack_ref=%b", wr_ack_ref);
                $display("*************************************************************");
            end
            else begin
                correct_cnt++;
                $display("*************************************************************");
                $display("PASS");
                $display("Input Signals:");
                $display("rst_n:%b", F_txn.rst_n);
                $display("wr_en:%b - rd_en:%b", F_txn.wr_en,F_txn.rd_en);
                $display("data_in:%d", F_txn.data_in);
                $display("----------------------");
                $display("DUT Outputs:");
                $display("data_out=%d", F_txn.data_out);
                $display("full=%b, almostfull= %b", F_txn.full, F_txn.almostfull);
                $display("empty=%b, almostempty= %b", F_txn.empty, F_txn.almostempty);
                $display("overflow=%b, underflow= %b", F_txn.overflow, F_txn.underflow);
                $display("wr_ack_ref=%b", F_txn.wr_ack);
                $display("----------------------");
                $display("Ref Model Outputs:");
                $display("data_out=%d", data_out_ref);
                $display("full=%b, almostfull= %b", full_ref, almostfull_ref);
                $display("empty=%b, almostempty= %b", empty_ref, almostempty_ref);
                $display("overflow=%b, underflow= %b", overflow_ref, underflow_ref);
                $display("wr_ack_ref=%b", wr_ack_ref);
                $display("*************************************************************");
            end
        endfunction
//FIFO_1//FIFO_2//FIFO_3//FIFO_4//FIFO_5//FIFO_6//FIFO_7//FIFO_8//FIFO_9//FIFO_10//FIFO_11//FIFO_12
        function void reference_model (FIFO_transaction F_txn);
            if (!F_txn.rst_n) begin
                empty_ref = 1;
                full_ref = 0;
                underflow_ref = 0;
                overflow_ref = 0;
```

```
FIFO_scoreboard.sv
 3  import shared_pkg::*;
 5      class FIFO_scoreboard;
72          endfunction
73  //FIFO_1//FIFO_2//FIFO_3//FIFO_4//FIFO_5//FIFO_6//FIFO_7//FIFO_8//FIFO_9//FIFO_10//FIFO_11//FIFO_12
74          function void reference_model (FIFO_transaction F_txn);
75              if (!F_txn.rst_n) begin
76                  empty_ref = 1;
77                  full_ref = 0;
78                  underflow_ref = 0;
79                  overflow_ref = 0;
80                  almostempty_ref = 0;
81                  almostfull_ref = 0;
82                  wr_ack_ref = 0;
83                  FIFO_refmodel.delete();
84              end
85              else begin
86
87                  wr_ack_ref = (F_txn.wr_en && FIFO_refmodel.size() < FIFO_DEPTH) ? 1 : 0;
88
89                  //Write
90                  if (F_txn.wr_en && !full_ref) begin
91                      FIFO_refmodel.push_back(F_txn.data_in);
92                  end
93                  else begin
94                      overflow_ref = (F_txn.wr_en && full_ref)? 1 : 0;
95                  end
96                  //Read
97                  if (F_txn.rd_en && !empty_ref) begin
98                      data_out_ref = FIFO_refmodel.pop_front();
99                  end
100                 else begin
101                     underflow_ref = (F_txn.rd_en && empty_ref) ? 1 : 0;
102                 end
103
104                 full_ref = (FIFO_refmodel.size() == FIFO_DEPTH) ? 1 : 0;
105                 empty_ref = (FIFO_refmodel.size() == 0) ? 1 : 0;
106                 almostfull_ref = (FIFO_refmodel.size() == FIFO_DEPTH-1) ? 1 : 0;
107                 almostempty_ref = (FIFO_refmodel.size() == 1) ? 1 : 0;
108             end
109
110         endfunction
111     endclass
```


Coverage

```
FIFO_coverage.sv
 1  package FIFO_coverage_pkg;
 2  import FIFO_transaction_pkg::*;
 3  class FIFO_coverage;
 4      FIFO_transaction F_cvg_txn;
 5      covergroup FIFO_cvr;
 6
 7          //FIFO_2//FIFO_3//FIFO_4//FIFO_5//FIFO_6//FIFO_7//FIFO_8//FIFO_9//FIFO_10//FIFO_11//FIFO_12
 8          write_en: coverpoint F_cvg_txn.wr_en;
 9          read_en: coverpoint F_cvg_txn.rd_en;
10          full: coverpoint F_cvg_txn.full;
11          almostfull: coverpoint F_cvg_txn.almostfull;
12          empty: coverpoint F_cvg_txn.empty;
13          almostempty: coverpoint F_cvg_txn.almostempty;
14          overflow: coverpoint F_cvg_txn.overflow;
15          underflow: coverpoint F_cvg_txn.underflow;
16          write_ack: coverpoint F_cvg_txn.wr_ack;
17
18          cross_full: cross write_en, read_en, full{
19              ignore_bins full1 = binsof (write_en) intersect {1} && binsof (read_en) intersect {1} && binsof (full) intersect {1};
20              ignore_bins full2 = binsof (write_en) intersect {0} && binsof (read_en) intersect {1} && binsof (full) intersect {1};
21          }
22          cross_almostfull: cross write_en, read_en, almostfull;
23          cross_empty:cross write_en, read_en, empty;
24          cross_almostempty:cross write_en, read_en, almostempty;
25          cross_overflow:cross write_en, read_en, overflow{
26              ignore_bins overflow1 = binsof (write_en) intersect {0} && binsof (read_en) intersect {1} && binsof (overflow) intersect {1};
27              ignore_bins overflow2 = binsof (write_en) intersect {0} && binsof (read_en) intersect {0} && binsof (overflow) intersect {1};
28          }
29          cross_underflow:cross write_en, read_en, underflow{
30              ignore_bins underflow1 = binsof (write_en) intersect {1} && binsof (read_en) intersect {0} && binsof (underflow) intersect {1};
31              ignore_bins underflow2 = binsof (write_en) intersect {0} && binsof (read_en) intersect {0} && binsof (underflow) intersect {1};
32          }
33          cross_write_ack:cross write_en, read_en, write_ack{
34              ignore_bins wr_ack1 = binsof (write_en) intersect {0} && binsof (read_en) intersect {1} && binsof (write_ack) intersect {1};
35              ignore_bins wr_ack2 = binsof (write_en) intersect {0} && binsof (read_en) intersect {0} && binsof (write_ack) intersect {1};
36          }
37
38      endgroup
39
40      function new();
41          FIFO_cvr = new;
42      endfunction
```



```
40      function new();
41          FIFO_cvr = new;
42      endfunction
43
44      function void sample_data(FIFO_transaction F_txn);
45          this.F_cvg_txn = F_txn;
46          FIFO_cvr.sample();
47      endfunction
48  endclass
49  endpackage
```

Monitor

FIFO_monitor.sv

```sv
import FIFO_transaction_pkg::*;
import shared_pkg::*;
module FIFO_monitor (FIFO_if fifo_if);

    FIFO_transaction F_txn;
    FIFO_scoreboard F_scrbrd;
    FIFO_coverage F_cvg;

    initial begin
        F_txn = new;
        F_scrbrd = new;
        F_cvg = new;
        forever begin
            wait (emonitor.triggered);
            @(negedge fifo_if.clk);
            F_txn.rst_n = fifo_if.rst_n;
            F_txn.wr_en = fifo_if.wr_en;
            F_txn.rd_en = fifo_if.rd_en;
            F_txn.data_in = fifo_if.data_in;
            F_txn.data_out = fifo_if.data_out;
            F_txn.full = fifo_if.full;
            F_txn.almostfull = fifo_if.almostfull;
            F_txn.empty = fifo_if.empty;
            F_txn.almostempty = fifo_if.almostempty;
            F_txn.overflow = fifo_if.overflow;
            F_txn.underflow = fifo_if.underflow;
            F_txn.wr_ack = fifo_if.wr_ack;
            fork
                begin
                    F_cvg.sample_data(F_txn);
                end
                begin
                    F_scrbrd.check_data(F_txn);
                end
            join
            if (test_finished) begin
                $display("Correct Tests = %d, Error Tests = %d", correct_cnt, error_cnt);
                $stop;
            end
        end
    end
endmodule
```

Testbench Code Snippets

FIFO_tb.sv

```sv
import shared_pkg::*;
import FIFO_transaction_pkg::*;
module FIFO_tb (FIFO_if f_if);


    logic [FIFO_WIDTH-1:0] FIFO_refmodel [$];
    logic [FIFO_WIDTH-1:0] data_out_ref;
    logic wr_ack_ref, overflow_ref;
    logic full_ref, empty_ref, almostfull_ref, almostempty_ref, underflow_ref;
    FIFO_transaction F_txn;

    int error_cnt_tb = 0, correct_cnt_tb = 0;

    initial begin
        F_txn = new;
        correct_cnt = 0;
        error_cnt = 0;
        test_finished = 0;
        //FIFO_1
        f_if.rst_n = 0;
        f_if.wr_en = 0;
        f_if.rd_en = 0;
        f_if.data_in = 0;
        ->emonitor;
        repeat(4)@(negedge f_if.clk);
        check_data();
        //FIFO_2//FIFO_3//FIFO_4//FIFO_5//FIFO_6//FIFO_7//FIFO_8//FIFO_9//FIFO_10//FIFO_11//FIFO_12
        repeat(3000) begin
            assert(F_txn.randomize());
            f_if.rst_n = F_txn.rst_n;
            f_if.wr_en = F_txn.wr_en;
            f_if.rd_en = F_txn.rd_en;
            f_if.data_in = F_txn.data_in;
            ->emonitor;
            @(negedge f_if.clk);
            check_data();
        end
        test_finished = 1;
        ->emonitor;
        $display("Correct Test @TB = %d, Error Test @TB = %d", correct_cnt_tb, error_cnt_tb);
    end
```

FIFO_tb.sv

```
module FIFO_tb (FIFO_if f_if);

    task check_data;
        ref_model();
        if (data_out_ref != f_if.data_out) begin
            error_cnt_tb++;
            $display("Error - DUT out: %h != Ref Model out: %h", f_if.data_out, data_out_ref);
        end
        else begin
            correct_cnt_tb ++;
        end
    endtask

    task ref_model;
            if (!f_if.rst_n) begin
                empty_ref = 1;
                full_ref = 0;
                underflow_ref = 0;
                overflow_ref = 0;
                almostempty_ref = 0;
                almostfull_ref = 0;
                wr_ack_ref = 0;
                FIFO_refmodel.delete();
            end
            else begin

                wr_ack_ref = (f_if.wr_en && FIFO_refmodel.size() < FIFO_DEPTH) ? 1 : 0;

                //Write
                if (f_if.wr_en && !full_ref) begin
                    FIFO_refmodel.push_back(f_if.data_in);
                end
                else begin
                    overflow_ref = (f_if.wr_en && full_ref)? 1 : 0;
                end
                //Read
                if (f_if.rd_en && !empty_ref) begin
                    data_out_ref = FIFO_refmodel.pop_front();
                end
                else begin
                    underflow_ref = (f_if.rd_en && empty_ref) ? 1 : 0;
                end
```

```
                    underflow_ref = (f_if.rd_en && empty_ref) ? 1 : 0;
                end

                full_ref = (FIFO_refmodel.size() == FIFO_DEPTH) ? 1 : 0;
                empty_ref = (FIFO_refmodel.size() == 0) ? 1 : 0;
                almostfull_ref = (FIFO_refmodel.size() == FIFO_DEPTH-1) ? 1 : 0;
                almostempty_ref = (FIFO_refmodel.size() == 1) ? 1 : 0;
            end
    endtask


endmodule
```

DO

run.do

```
# 1. Setup and Compile with coverage enabled
vlib work
vlog -f src_files.list +cover -covercells +define+SIM

# 2. Start simulation
vsim -voptargs=+acc work.top -classdebug -cover

# 3. Add waves for debugging
add wave /top/f_if/*
add wave -position insertpoint  \
    sim:/top/DUT/mem \
    sim:/top/DUT/wr_ptr \
    sim:/top/DUT/rd_ptr \
    sim:/top/DUT/count
add wave -position insertpoint  \
    sim:/top/MONITOR/F_txn
add wave -position end  sim:/top/MONITOR/F_scrbrd
add wave -position end  sim:/top/MONITOR/F_cvg
add wave -position insertpoint  \
    sim:/shared_pkg::error_cnt \
    sim:/shared_pkg::correct_cnt

# Add Assertion Waves for debugging
add wave /top/a_reset_1 /top/a_reset_2 /top/a_reset_3 /top/DUT/a_resetBehaviour /top/DUT/a_wr_ack /top/DUT/a_overflow /top/DUT/a_underflow /top/DUT/a_empty /top/DUT/a_full /top/DUT/

# 4. Run the simulation to completion
run -all

# 5. Save the collected coverage data to a file
coverage save FIFO.ucdb -onexit

coverage exclude -cvgpath {/FIFO_coverage_pkg/FIFO_coverage/FIFO_cvr/cross_full/<auto[1],auto[1],auto[1]>} {/FIFO_coverage_pkg/FIFO_coverage/FIFO_cvr/cross_full/<auto[0],auto[1],aut
coverage exclude -cvgpath {/FIFO_coverage_pkg/FIFO_coverage/FIFO_cvr/cross_overflow/<auto[0],auto[1],auto[1]>} {/FIFO_coverage_pkg/FIFO_coverage/FIFO_cvr/cross_overflow/<auto[0],aut
coverage exclude -cvgpath {/FIFO_coverage_pkg/FIFO_coverage/FIFO_cvr/cross_underflow/<auto[1],auto[0],auto[1]>} {/FIFO_coverage_pkg/FIFO_coverage/FIFO_cvr/cross_underflow/<auto[0],a
coverage exclude -cvgpath {/FIFO_coverage_pkg/FIFO_coverage/FIFO_cvr/cross_write_ack/<auto[0],auto[1],auto[1]>} {/FIFO_coverage_pkg/FIFO_coverage/FIFO_cvr/cross_write_ack/<auto[0],a
```

Coverage

Code Coverage

Branch

FIFO_cvr.txt - Notepad

File Edit Format View Help

```
                    FIFO.sv(114)                       0          1
/top/DUT/a_count_thr FIFO.sv(115)                      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------------------------------------
    18                                     3270     Count coming in to IF
    18              1                       554        if (!f_if.rst_n) begin

    23              1                      1701        else if (f_if.wr_en && count < f_if.FIFO_DEPTH) begin

    28              1                      1015        else begin

Branch totals: 3 hits of 3 branches = 100.00%

------------------------------------IF Branch------------------------------------
    30                                     1015     Count coming in to IF
    30              1                       240                if (f_if.full & f_if.wr_en)

    32              1                       775                else

Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    38                                     3270     Count coming in to IF
    38              1                       554        if (!f_if.rst_n) begin

    42              1                       684        else if (f_if.rd_en && count != 0) begin

    46              1                      2032        else begin

Branch totals: 3 hits of 3 branches = 100.00%

------------------------------------IF Branch------------------------------------
    47                                     2032     Count coming in to IF
    47              1                       112                if (f_if.empty & f_if.rd_en)

    49              1                      1920                else

Branch totals: 2 hits of 2 branches = 100.00%
```

Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8

FIFO_cvr.txt - Notepad

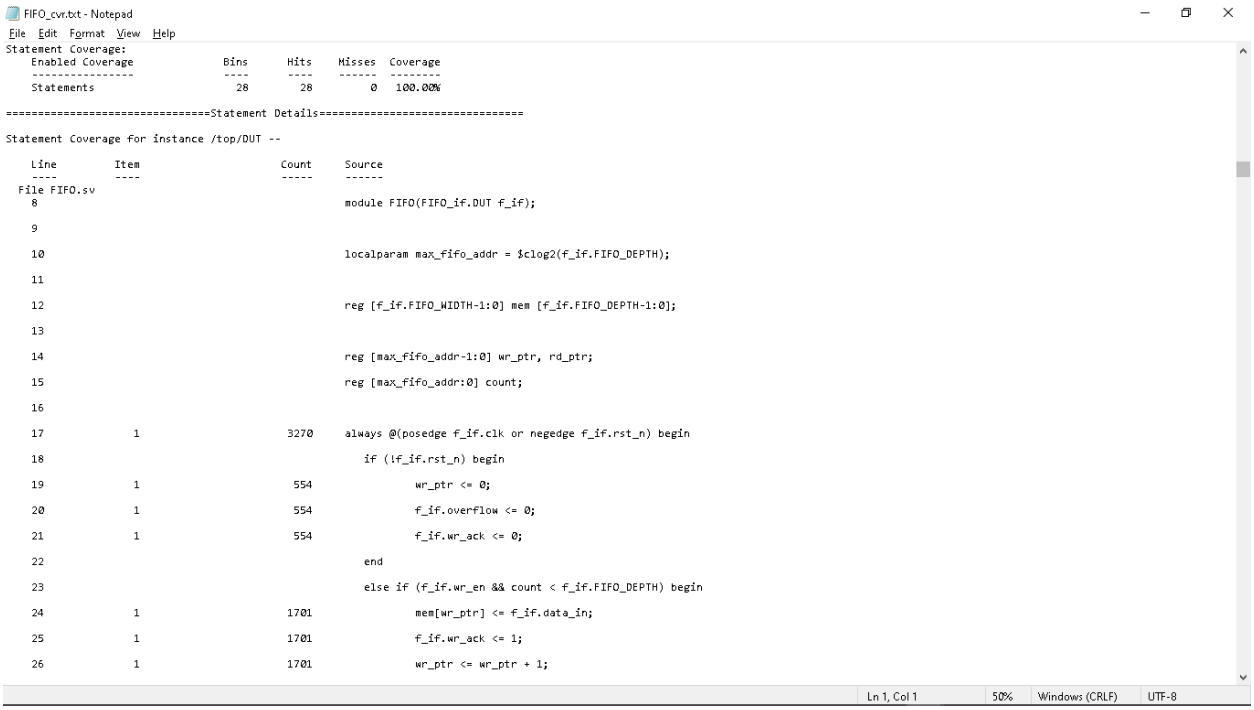
File Edit Format View Help

```
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    55                                     3005     Count coming in to IF
    55              1                       547        if (!f_if.rst_n) begin

    58              1                      2458        else begin

Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    59                                     2458     Count coming in to IF
    59              1                        76                if      ( ({f_if.wr_en, f_if.rd_en} == 2'b11) && f_if.full)

    61              1                        83                else if ( ({f_if.wr_en, f_if.rd_en} == 2'b11) && f_if.empty)

    63              1                      1212                else if ( ({f_if.wr_en, f_if.rd_en} == 2'b10) && !f_if.full)

    65              1                       202                else if ( ({f_if.wr_en, f_if.rd_en} == 2'b01) && !f_if.empty)

                                            885     All False Count
Branch totals: 5 hits of 5 branches = 100.00%

------------------------------------IF Branch------------------------------------
    70                                     1817     Count coming in to IF
    70              1                       138     assign f_if.full = (count == f_if.FIFO_DEPTH)? 1 : 0; //first mistake

    70              2                      1679     assign f_if.full = (count == f_if.FIFO_DEPTH)? 1 : 0; //first mistake

Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    71                                     1817     Count coming in to IF
    71              1                       283     assign f_if.empty = (count == 0)? 1 : 0;

    71              2                      1534     assign f_if.empty = (count == 0)? 1 : 0;

Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    72                                     1817     Count coming in to IF
    72              1                       187     assign f_if.almostfull = (count == f_if.FIFO_DEPTH-1)? 1 : 0;

    72              2                      1630     assign f_if.almostfull = (count == f_if.FIFO_DEPTH-1)? 1 : 0;

Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    73                                     1817     Count coming in to IF
    73              1                       319     assign f_if.almostempty = (count == 1)? 1 : 0;
```

Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8

```
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    112                                    2760     Count coming in to IF
    112             1                      2244        if (f_if.rst_n) begin

                                            516     All False Count
Branch totals: 2 hits of 2 branches = 100.00%
```

Statement

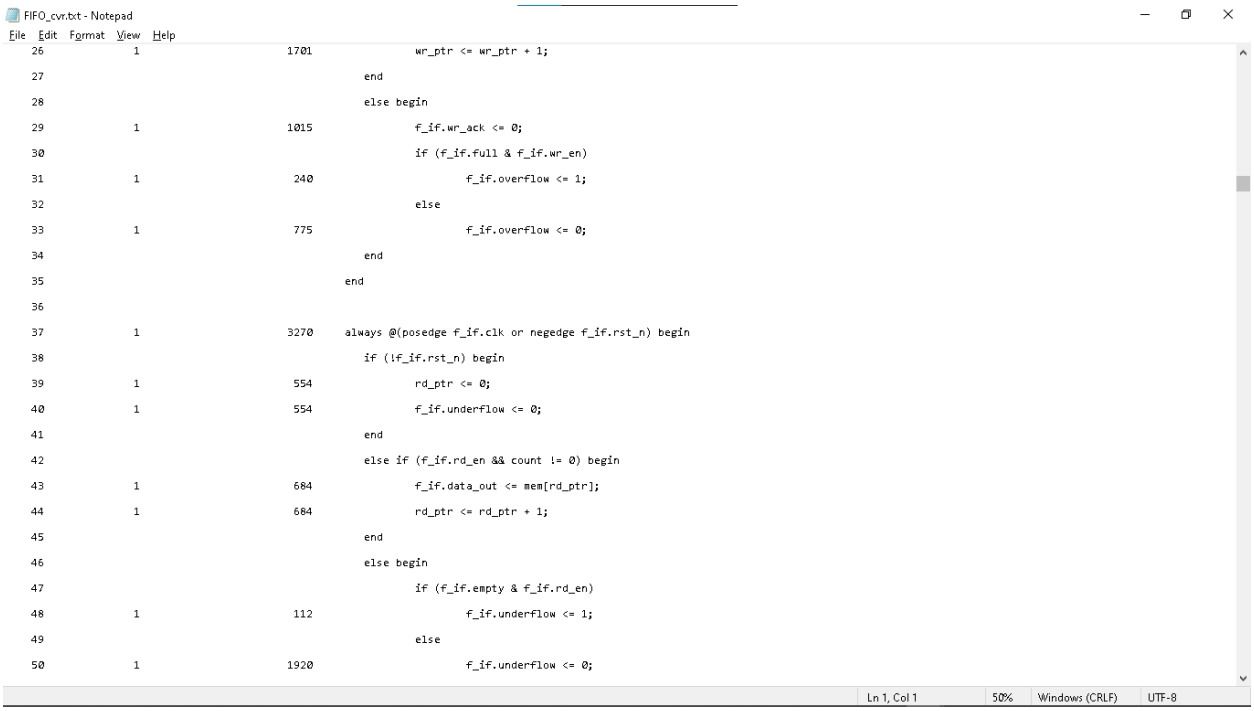


FIFO_cvr.txt - Notepad

File Edit Format View Help

```
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                      28        28         0   100.00%

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

Statement Coverage for instance /top/DUT --

    Line         Item                      Count     Source
    ----         ----                      -----     ------
  File FIFO.sv
    8                                      	module FIFO(FIFO_if.DUT f_if);
    9
    10                                     	localparam max_fifo_addr = $clog2(f_if.FIFO_DEPTH);
    11
    12                                     	reg [f_if.FIFO_WIDTH-1:0] mem [f_if.FIFO_DEPTH-1:0];
    13
    14                                     	reg [max_fifo_addr-1:0] wr_ptr, rd_ptr;
    15                                     	reg [max_fifo_addr:0] count;
    16
    17             1                  3270     always @(posedge f_if.clk or negedge f_if.rst_n) begin
    18                                        	if (!f_if.rst_n) begin
    19             1                   554     		wr_ptr <= 0;
    20             1                   554     		f_if.overflow <= 0;
    21             1                   554     		f_if.wr_ack <= 0;
    22                                        	end
    23                                        	else if (f_if.wr_en && count < f_if.FIFO_DEPTH) begin
    24             1                  1701     		mem[wr_ptr] <= f_if.data_in;
    25             1                  1701     		f_if.wr_ack <= 1;
    26             1                  1701     		wr_ptr <= wr_ptr + 1;
```

Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8



FIFO_cvr.txt - Notepad

File Edit Format View Help

```
    26             1                  1701     		wr_ptr <= wr_ptr + 1;
    27                                        	end
    28                                        	else begin
    29             1                  1015     		f_if.wr_ack <= 0;
    30                                        		if (f_if.full & f_if.wr_en)
    31             1                   240     			f_if.overflow <= 1;
    32                                        		else
    33             1                   775     			f_if.overflow <= 0;
    34                                        	end
    35                                        end
    36
    37             1                  3270     always @(posedge f_if.clk or negedge f_if.rst_n) begin
    38                                        	if (!f_if.rst_n) begin
    39             1                   554     		rd_ptr <= 0;
    40             1                   554     		f_if.underflow <= 0;
    41                                        	end
    42                                        	else if (f_if.rd_en && count != 0) begin
    43             1                   684     		f_if.data_out <= mem[rd_ptr];
    44             1                   684     		rd_ptr <= rd_ptr + 1;
    45                                        	end
    46                                        	else begin
    47                                        		if (f_if.empty & f_if.rd_en)
    48             1                   112     			f_if.underflow <= 1;
    49                                        		else
    50             1                  1920     			f_if.underflow <= 0;
```

Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8

FIFO_cvr.txt - Notepad

File Edit Format View Help

```
49                                                      else
50              1                       1920                    f_if.underflow <= 0;
51                                                  end
52                                              end
53
54              1                       3005    always @(posedge f_if.clk or negedge f_if.rst_n) begin
55                                                  if (!f_if.rst_n) begin
56              1                        547                count <= 0;
57                                                  end
58                                                  else begin
59                                                      if      ( ({f_if.wr_en, f_if.rd_en} == 2'b11) && f_if.full)
60              1                         76                    count <= count - 1;
61                                                      else if ( ({f_if.wr_en, f_if.rd_en} == 2'b11) && f_if.empty)
62              1                         83                    count <= count + 1;
63                                                      else if ( ({f_if.wr_en, f_if.rd_en} == 2'b10) && !f_if.full)
64              1                       1212                    count <= count + 1;
65                                                      else if ( ({f_if.wr_en, f_if.rd_en} == 2'b01) && !f_if.empty)
66              1                        202                    count <= count - 1;
67                                                  end
68                                              end
69
70              1                       1818    assign f_if.full = (count == f_if.FIFO_DEPTH)? 1 : 0; //first mistake
71              1                       1818    assign f_if.empty = (count == 0)? 1 : 0;
72              1                       1818    assign f_if.almostfull = (count == f_if.FIFO_DEPTH-1)? 1 : 0;
73              1                       1818    assign f_if.almostempty = (count == 1)? 1 : 0;
```

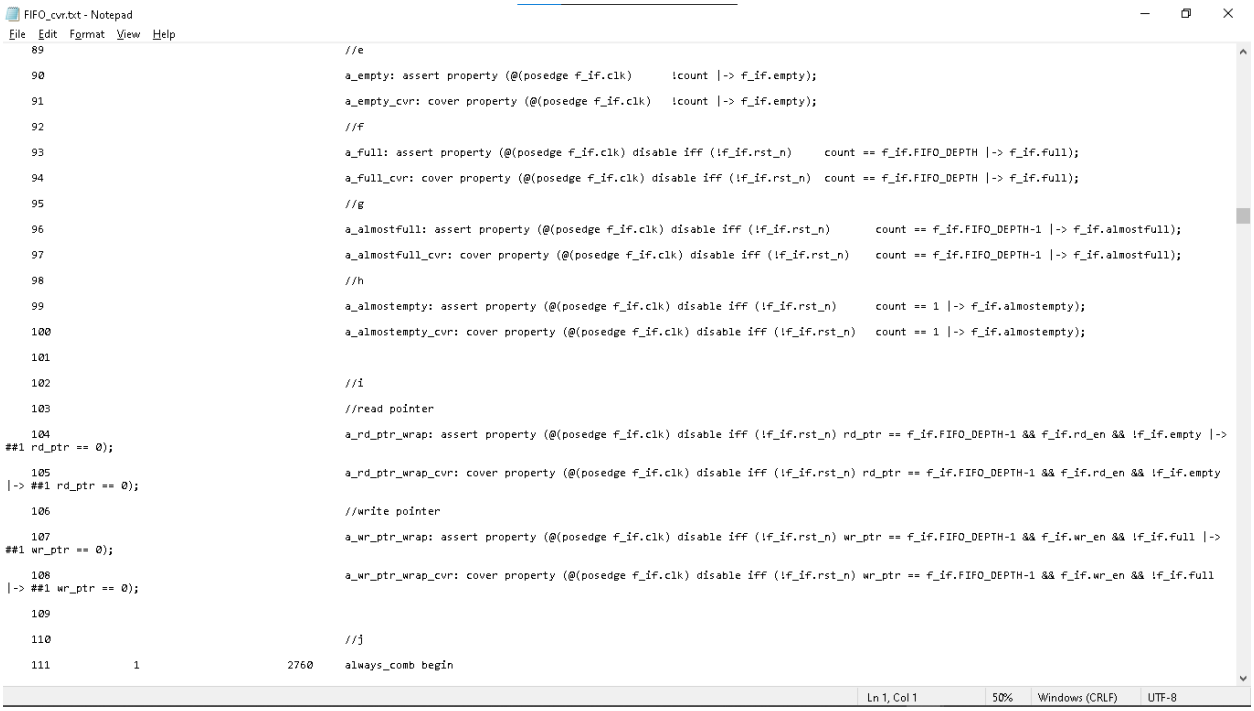
Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8

FIFO_cvr.txt - Notepad

File Edit Format View Help

```
70              1                       1818    assign f_if.full = (count == f_if.FIFO_DEPTH)? 1 : 0; //first mistake
71              1                       1818    assign f_if.empty = (count == 0)? 1 : 0;
72              1                       1818    assign f_if.almostfull = (count == f_if.FIFO_DEPTH-1)? 1 : 0;
73              1                       1818    assign f_if.almostempty = (count == 1)? 1 : 0;
74
75                                              `ifdef SIM
76                                              //Assertions
77                                              //a
78                                              a_resetBehaviour: assert property (@(posedge f_if.clk) !f_if.rst_n |-> !wr_ptr && !rd_ptr && !count);
79                                              a_resetBehaviour_cvr: cover property (@(posedge f_if.clk) !f_if.rst_n |-> !wr_ptr && !rd_ptr && !count);
80                                              //b
81                                              a_wr_ack: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n) f_if.wr_en && !f_if.full  |-> ##1 f_if.wr_ack);
82                                              a_wr_ack_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n) f_if.wr_en && !f_if.full  |-> ##1 f_if.wr_ack);
83                                              //c
84                                              a_overflow: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)         f_if.full && f_if.wr_en |-> ##1 f_if.overflow);
85                                              a_overflow_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)      f_if.full && f_if.wr_en |-> ##1 f_if.overflow);
86                                              //d
87                                              a_underflow: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)        f_if.empty && f_if.rd_en |->##1 f_if.underflow);
88                                              a_underflow_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)     f_if.empty && f_if.rd_en |->##1 f_if.underflow);
89                                              //e
90                                              a_empty: assert property (@(posedge f_if.clk)      !count |-> f_if.empty);
91                                              a_empty_cvr: cover property (@(posedge f_if.clk)   !count |-> f_if.empty);
92                                              //f
93                                              a_full: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)     count == f_if.FIFO_DEPTH |-> f_if.full);
94                                              a_full_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)  count == f_if.FIFO_DEPTH |-> f_if.full);
```

Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8



FIFO_cvr.txt - Notepad

File Edit Format View Help

```
    89                                          //e
    90                                          a_empty: assert property (@(posedge f_if.clk)      !count |-> f_if.empty);
    91                                          a_empty_cvr: cover property (@(posedge f_if.clk)   !count |-> f_if.empty);
    92                                          //f
    93                                          a_full: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)     count == f_if.FIFO_DEPTH |-> f_if.full);
    94                                          a_full_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)  count == f_if.FIFO_DEPTH |-> f_if.full);
    95                                          //g
    96                                          a_almostfull: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)       count == f_if.FIFO_DEPTH-1 |-> f_if.almostfull);
    97                                          a_almostfull_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)    count == f_if.FIFO_DEPTH-1 |-> f_if.almostfull);
    98                                          //h
    99                                          a_almostempty: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n)      count == 1 |-> f_if.almostempty);
    100                                         a_almostempty_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n)   count == 1 |-> f_if.almostempty);
    101
    102                                         //i
    103                                         //read pointer
    104                                         a_rd_ptr_wrap: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n) rd_ptr == f_if.FIFO_DEPTH-1 && f_if.rd_en && !f_if.empty |->
##1 rd_ptr == 0);
    105                                         a_rd_ptr_wrap_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n) rd_ptr == f_if.FIFO_DEPTH-1 && f_if.rd_en && !f_if.empty
|-> ##1 rd_ptr == 0);
    106                                         //write pointer
    107                                         a_wr_ptr_wrap: assert property (@(posedge f_if.clk) disable iff (!f_if.rst_n) wr_ptr == f_if.FIFO_DEPTH-1 && f_if.wr_en && !f_if.full |->
##1 wr_ptr == 0);
    108                                         a_wr_ptr_wrap_cvr: cover property (@(posedge f_if.clk) disable iff (!f_if.rst_n) wr_ptr == f_if.FIFO_DEPTH-1 && f_if.wr_en && !f_if.full
|-> ##1 wr_ptr == 0);
    109
    110                                         //j
    111             1                    2760    always_comb begin
```

Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8

Toggle

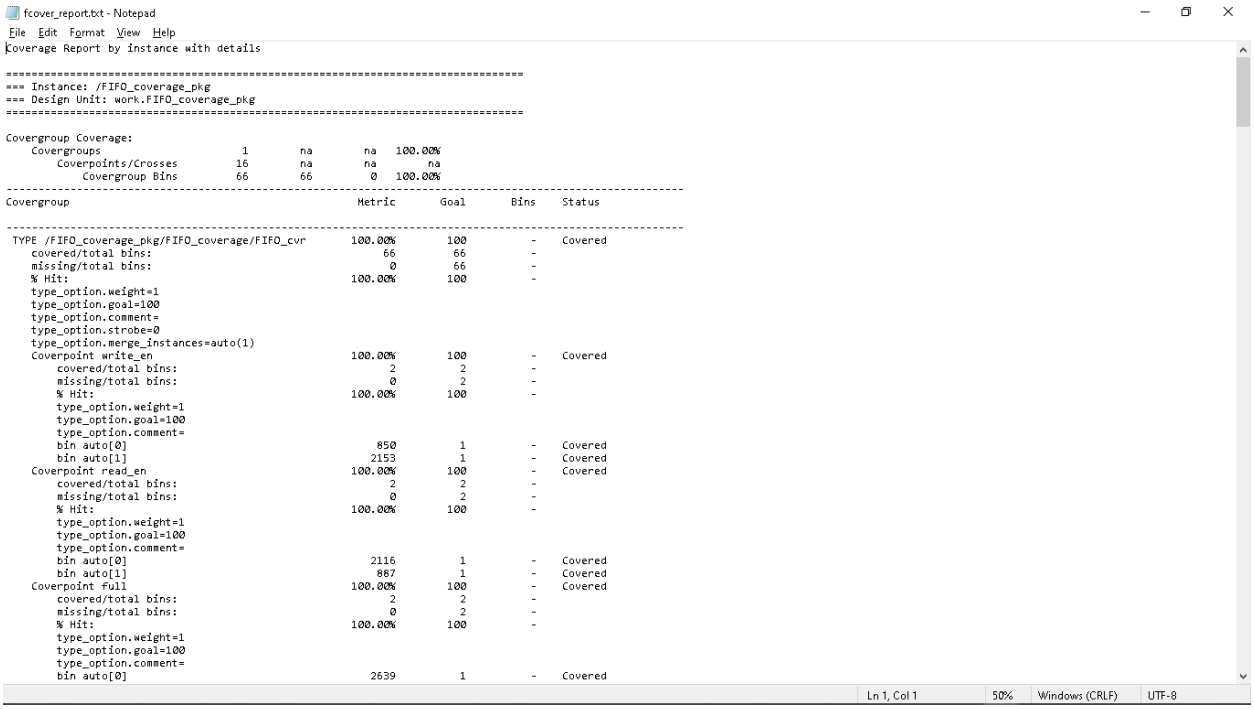
```
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         20        20         0   100.00%

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

Toggle Coverage for instance /top/DUT --

                                        Node      1H->0L      0L->1H  "Coverage"
                                        ----------------------------------------
                                   count[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 /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[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)
```

Functional Coverage

Note: Some Exclusion have been done to achieve the 100% functional coverage

1- Exclusion of bin wr_en = 1, rd_en = 1 and full = 1 because that cannot happen
2- Exclusion of bin wr_en = 0, rd_en = 1 and full = 1 because that cannot happen
3- Exclusion of bin wr_en = 0, rd_en=1 and overflow =1 because that cannot happen
4- Exclusion of bin wr_en = 0, rd_en = 0 and overflow = 1 because that cannot happen
5- Exclusion of bin wr_en = 1, rd_en = 0 and underflow = 1 because that cannot happen
6- Exclusion of bin wr_en = 0, rd_en = 0 and underflow = 1 because that cannot happen
7- Exclusion of bin wr_en = 0, rd_en = 1 and wr_ack = 1 because that cannot happen
8- Exclusion of bin wr_en = 0, rd_en = 0 and wr_ack = 1 because that cannot happen



fcover_report.txt - Notepad

File Edit Format View Help

```
Coverage Report by instance with details

================================================================================
=== 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/FIFO_cvr       100.00%        100          -    Covered
    covered/total bins:                                    66         66          -
    missing/total bins:                                     0         66          -
    % Hit:                                             100.00%        100          -
    type_option.weight=1
    type_option.goal=100
    type_option.comment=
    type_option.strobe=0
    type_option.merge_instances=auto(1)
    Coverpoint write_en                                100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                         100.00%        100          -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                       850          1          -    Covered
        bin auto[1]                                      2153          1          -    Covered
    Coverpoint read_en                                 100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                         100.00%        100          -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                      2116          1          -    Covered
        bin auto[1]                                       887          1          -    Covered
    Coverpoint full                                    100.00%        100          -    Covered
        covered/total bins:                                 2          2          -
        missing/total bins:                                 0          2          -
        % Hit:                                         100.00%        100          -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                      2639          1          -    Covered
```

Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8

fcover_report.txt - Notepad

File Edit Format View Help

```
        % Hit:                                  100.00%         100         -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                2639           1         -    Covered
        bin auto[1]                                 363           1         -    Covered
    Coverpoint almostfull                       100.00%         100         -    Covered
        covered/total bins:                            2           2         -
        missing/total bins:                            0           2         -
        % Hit:                                  100.00%         100         -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                2709           1         -    Covered
        bin auto[1]                                 293           1         -    Covered
    Coverpoint empty                            100.00%         100         -    Covered
        covered/total bins:                            2           2         -
        missing/total bins:                            0           2         -
        % Hit:                                  100.00%         100         -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                2565           1         -    Covered
        bin auto[1]                                 437           1         -    Covered
    Coverpoint almostempty                      100.00%         100         -    Covered
        covered/total bins:                            2           2         -
        missing/total bins:                            0           2         -
        % Hit:                                  100.00%         100         -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                2492           1         -    Covered
        bin auto[1]                                 510           1         -    Covered
    Coverpoint overflow                         100.00%         100         -    Covered
        covered/total bins:                            2           2         -
        missing/total bins:                            0           2         -
        % Hit:                                  100.00%         100         -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                2704           1         -    Covered
        bin auto[1]                                 298           1         -    Covered
    Coverpoint underflow                        100.00%         100         -    Covered
        covered/total bins:                            2           2         -
        missing/total bins:                            0           2         -
        % Hit:                                  100.00%         100         -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                2871           1         -    Covered
```

Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8

fcover_report.txt - Notepad

File Edit Format View Help

```
        % Hit:                                  100.00%         100         -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                2871           1         -    Covered
        bin auto[1]                                 131           1         -    Covered
    Coverpoint write_ack                        100.00%         100         -    Covered
        covered/total bins:                            2           2         -
        missing/total bins:                            0           2         -
        % Hit:                                  100.00%         100         -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        bin auto[0]                                1301           1         -    Covered
        bin auto[1]                                1701           1         -    Covered
    Cross cross_full                            100.00%         100         -    Covered
        covered/total bins:                            6           6         -
        missing/total bins:                            0           6         -
        % Hit:                                  100.00%         100         -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[0]>              632           1         -    Covered
            bin <auto[0],auto[1],auto[0]>              255           1         -    Covered
            bin <auto[1],auto[0],auto[1]>              302           1         -    Covered
            bin <auto[1],auto[0],auto[0]>             1219           1         -    Covered
            bin <auto[0],auto[0],auto[1]>               61           1         -    Covered
            bin <auto[0],auto[0],auto[0]>              533           1         -    Covered
        Illegal and Ignore Bins:
            ignore_bin full2                             0                     -    ZERO
            ignore_bin full1                             0                     -    ZERO
    Cross cross_almostfull                      100.00%         100         -    Covered
        covered/total bins:                            8           8         -
        missing/total bins:                            0           8         -
        % Hit:                                  100.00%         100         -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>              128           1         -    Covered
            bin <auto[0],auto[1],auto[1]>               21           1         -    Covered
            bin <auto[1],auto[0],auto[1]>               90           1         -    Covered
            bin <auto[0],auto[0],auto[1]>               54           1         -    Covered
            bin <auto[1],auto[1],auto[0]>              504           1         -    Covered
            bin <auto[0],auto[1],auto[0]>              234           1         -    Covered
            bin <auto[1],auto[0],auto[0]>             1431           1         -    Covered
            bin <auto[0],auto[0],auto[0]>              540           1         -    Covered
    Cross cross_empty                           100.00%         100         -    Covered
        covered/total bins:                            8           8         -
```

Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8

fcover_report.txt - Notepad

File Edit Format View Help

```
    Cross cross_empty                                  100.00%        100          -    Covered
        covered/total bins:                                  8          8          -
        missing/total bins:                                  0          8          -
        % Hit:                                         100.00%        100          -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>                   67          1          -    Covered
            bin <auto[0],auto[1],auto[1]>                   92          1          -    Covered
            bin <auto[1],auto[0],auto[1]>                  145          1          -    Covered
            bin <auto[0],auto[0],auto[1]>                  133          1          -    Covered
            bin <auto[1],auto[1],auto[0]>                  565          1          -    Covered
            bin <auto[0],auto[1],auto[0]>                  163          1          -    Covered
            bin <auto[1],auto[0],auto[0]>                 1376          1          -    Covered
            bin <auto[0],auto[0],auto[0]>                  461          1          -    Covered
    Cross cross_almostempty                            100.00%        100          -    Covered
        covered/total bins:                                  8          8          -
        missing/total bins:                                  0          8          -
        % Hit:                                         100.00%        100          -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>                  179          1          -    Covered
            bin <auto[0],auto[1],auto[1]>                   37          1          -    Covered
            bin <auto[1],auto[0],auto[1]>                  199          1          -    Covered
            bin <auto[0],auto[0],auto[1]>                   95          1          -    Covered
            bin <auto[1],auto[1],auto[0]>                  453          1          -    Covered
            bin <auto[0],auto[1],auto[0]>                  218          1          -    Covered
            bin <auto[1],auto[0],auto[0]>                 1322          1          -    Covered
            bin <auto[0],auto[0],auto[0]>                  499          1          -    Covered
    Cross cross_overflow                               100.00%        100          -    Covered
        covered/total bins:                                  6          6          -
        missing/total bins:                                  0          6          -
        % Hit:                                         100.00%        100          -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>                   91          1          -    Covered
            bin <auto[1],auto[0],auto[1]>                  207          1          -    Covered
            bin <auto[1],auto[1],auto[0]>                  541          1          -    Covered
            bin <auto[1],auto[0],auto[0]>                 1314          1          -    Covered
            bin <auto[0],auto[1],auto[0]>                  255          1          -    Covered
            bin <auto[0],auto[0],auto[0]>                  594          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin overflow2                             0                     -    ZERO
            ignore_bin overflow1                             0                     -    ZERO
    Cross cross_underflow                              100.00%        100          -    Covered
```

Ln 1, Col 1 | 50% | Windows (CRLF) | UTF-8

fcover_report.txt - Notepad

File Edit Format View Help

```
            bin <auto[1],auto[1],auto[0]>                  541          1          -    Covered
            bin <auto[1],auto[0],auto[0]>                 1314          1          -    Covered
            bin <auto[0],auto[1],auto[0]>                  255          1          -    Covered
            bin <auto[0],auto[0],auto[0]>                  594          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin overflow2                             0                     -    ZERO
            ignore_bin overflow1                             0                     -    ZERO
    Cross cross_underflow                              100.00%        100          -    Covered
        covered/total bins:                                  6          6          -
        missing/total bins:                                  0          6          -
        % Hit:                                         100.00%        100          -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>                   97          1          -    Covered
            bin <auto[1],auto[0],auto[0]>                  535          1          -    Covered
            bin <auto[0],auto[1],auto[1]>                   34          1          -    Covered
            bin <auto[0],auto[1],auto[0]>                  221          1          -    Covered
            bin <auto[1],auto[0],auto[0]>                 1521          1          -    Covered
            bin <auto[0],auto[0],auto[0]>                  594          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin underflow2                            0                     -    ZERO
            ignore_bin underflow1                            0                     -    ZERO
    Cross cross_write_ack                              100.00%        100          -    Covered
        covered/total bins:                                  6          6          -
        missing/total bins:                                  0          6          -
        % Hit:                                         100.00%        100          -
        type_option.weight=1
        type_option.goal=100
        type_option.comment=
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>                  489          1          -    Covered
            bin <auto[1],auto[0],auto[1]>                 1212          1          -    Covered
            bin <auto[1],auto[1],auto[0]>                  143          1          -    Covered
            bin <auto[1],auto[0],auto[0]>                  309          1          -    Covered
            bin <auto[0],auto[1],auto[0]>                  255          1          -    Covered
            bin <auto[0],auto[0],auto[0]>                  594          1          -    Covered
        Illegal and Ignore Bins:
            ignore_bin wr_ack2                               0                     -    ZERO
            ignore_bin wr_ack1                               0                     -    ZERO


            -    -      -

TOTAL COVERGROUP COVERAGE: 100.00%  COVERGROUP TYPES: 1

Total Coverage By Instance (filtered view): 100.00%
```

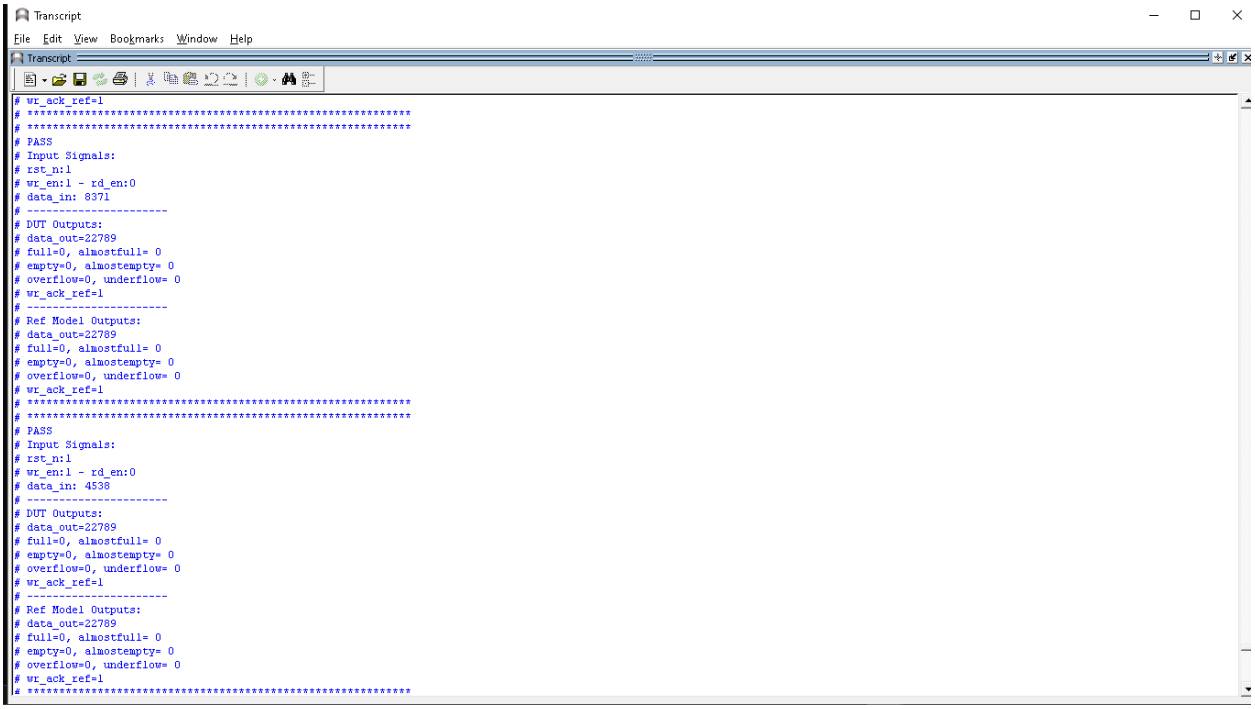
Assertion Coverage

```
DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                    Design Design   Lang File(Line)      Hits Status
                                        Unit   UnitType
--------------------------------------------------------------------------------------------
/top/DUT/a_resetBehaviour_cvr           FIFO   Verilog  SVA  FIFO.sv(80)     288 Covered
/top/DUT/a_wr_ack_cvr                   FIFO   Verilog  SVA  FIFO.sv(84)    1546 Covered
/top/DUT/a_overflow_cvr                 FIFO   Verilog  SVA  FIFO.sv(88)     211 Covered
/top/DUT/a_underflow_cvr                FIFO   Verilog  SVA  FIFO.sv(92)      97 Covered
/top/DUT/a_empty_cvr                    FIFO   Verilog  SVA  FIFO.sv(96)     682 Covered
/top/DUT/a_full_cvr                     FIFO   Verilog  SVA  FIFO.sv(100)    322 Covered
/top/DUT/a_almostfull_cvr               FIFO   Verilog  SVA  FIFO.sv(104)    269 Covered
/top/DUT/a_almostempty_cvr              FIFO   Verilog  SVA  FIFO.sv(108)    463 Covered
/top/DUT/a_rd_ptr_wrap_cvr              FIFO   Verilog  SVA  FIFO.sv(113)     21 Covered
/top/DUT/a_wr_ptr_wrap_cvr              FIFO   Verilog  SVA  FIFO.sv(116)    110 Covered

TOTAL DIRECTIVE COVERAGE: 100.00%  COVERS: 10

ASSERTION RESULTS:
---------------------------------------------------------------------
Name                File(Line)                   Failure      Pass
                                                 Count        Count
---------------------------------------------------------------------
/top/a_reset_1      FIFO_top.sv(17)                    0          1
/top/a_reset_2      FIFO_top.sv(18)                    0          1
/top/a_reset_3      FIFO_top.sv(19)                    0          1
/top/DUT/a_resetBehaviour
                    FIFO.sv(79)                        0          1
/top/DUT/a_wr_ack   FIFO.sv(83)                        0          1
/top/DUT/a_overflow FIFO.sv(87)                        0          1
/top/DUT/a_underflow FIFO.sv(91)                       0          1
/top/DUT/a_empty    FIFO.sv(95)                        0          1
/top/DUT/a_full     FIFO.sv(99)                        0          1
/top/DUT/a_almostfull
                    FIFO.sv(103)                       0          1
/top/DUT/a_almostempty
                    FIFO.sv(107)                       0          1
/top/DUT/a_rd_ptr_wrap
                    FIFO.sv(112)                       0          1
/top/DUT/a_wr_ptr_wrap
                    FIFO.sv(115)                       0          1
/top/DUT/a_rd_ptr_thr
                    FIFO.sv(121)                       0          1
/top/DUT/a_wr_ptr_thr
                    FIFO.sv(122)                       0          1
/top/DUT/a_count_thr FIFO.sv(123)                      0          1
/top/TB/#ublk#182146786#28/immed__29
```

Ln 2344, Col 69 | 50% | Windows (CRLF) | UTF-8

Simulation

Transcript


Transcript
File Edit View Bookmarks Window Help
Transcript
wr_ack_ref=1
**
**
PASS
Input Signals:
rst_n:1
wr_en:1 - rd_en:0
data_in: 8371

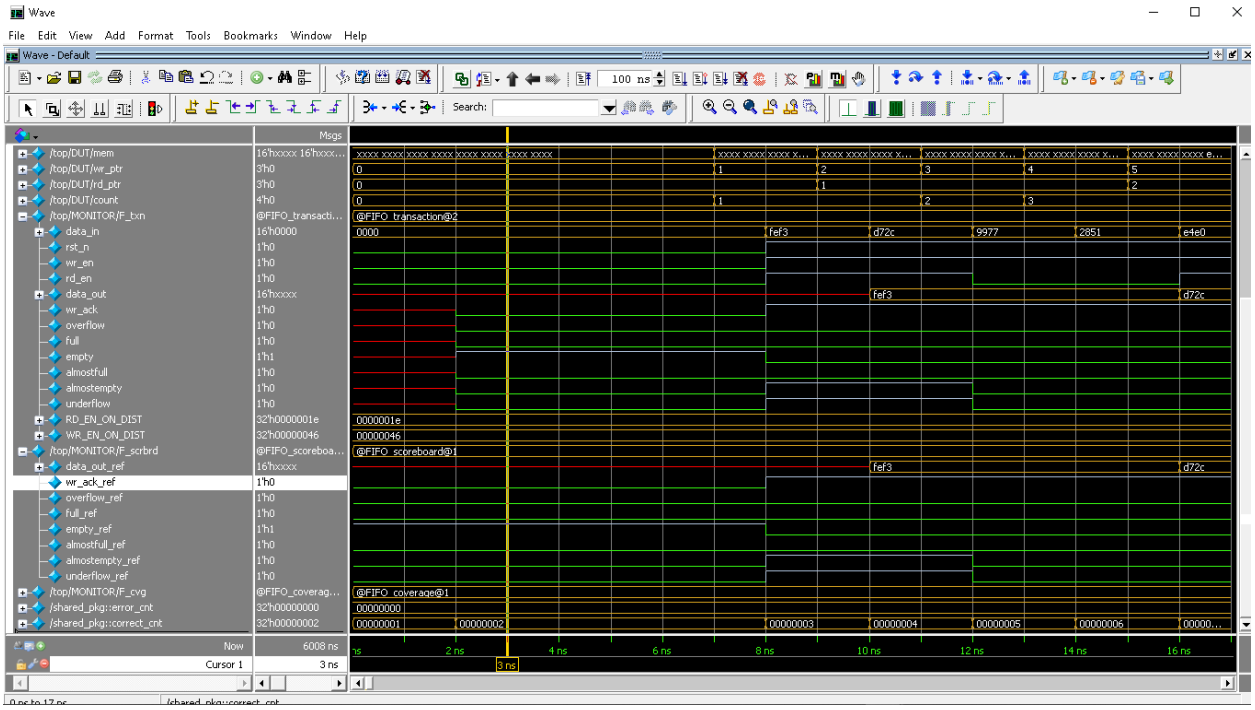
DUT Outputs:
data_out=22789
full=0, almostfull= 0
empty=0, almostempty= 0
overflow=0, underflow= 0
wr_ack_ref=1

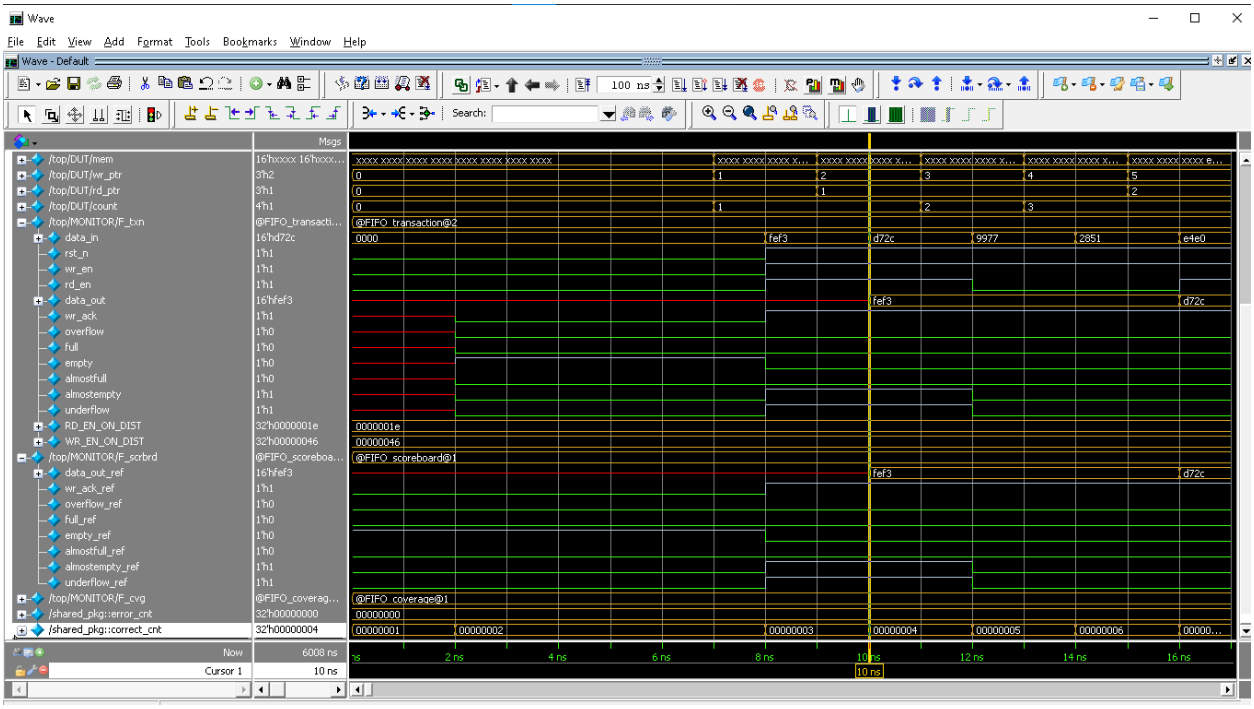
Ref Model Outputs:
data_out=22789
full=0, almostfull= 0
empty=0, almostempty= 0
overflow=0, underflow= 0
wr_ack_ref=1
**
**
PASS
Input Signals:
rst_n:1
wr_en:1 - rd_en:0
data_in: 4538

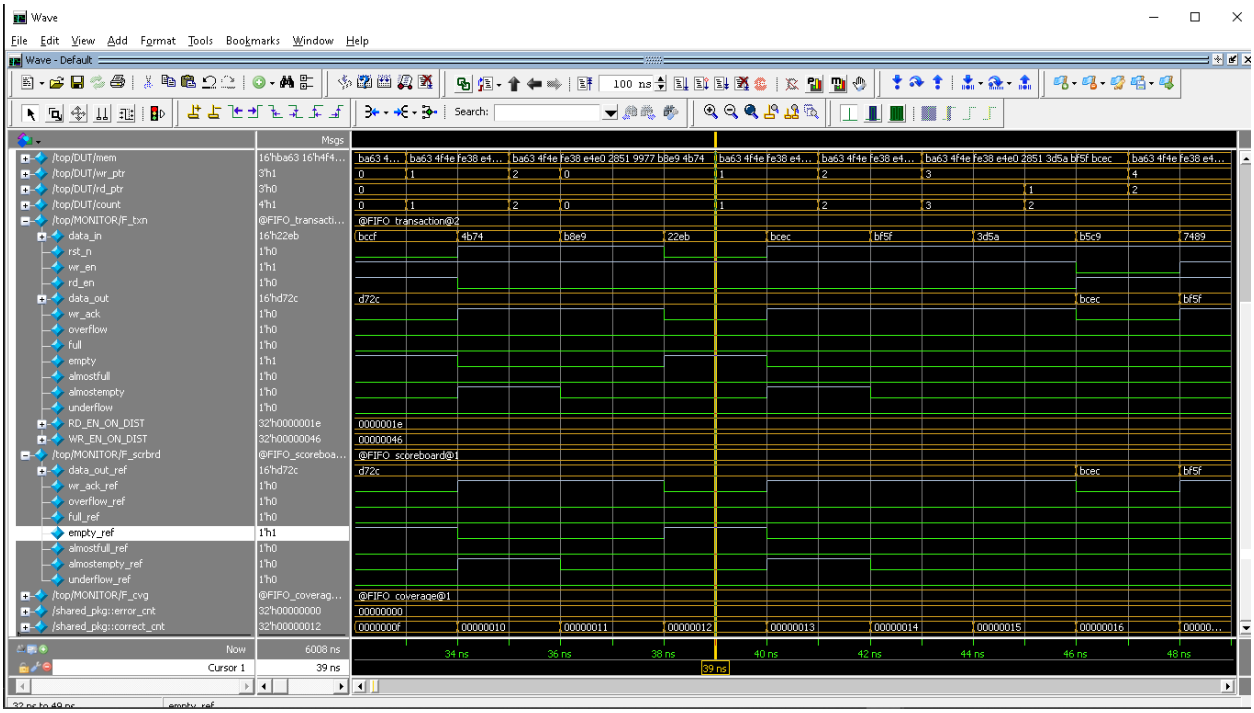
DUT Outputs:
data_out=22789
full=0, almostfull= 0
empty=0, almostempty= 0
overflow=0, underflow= 0
wr_ack_ref=1

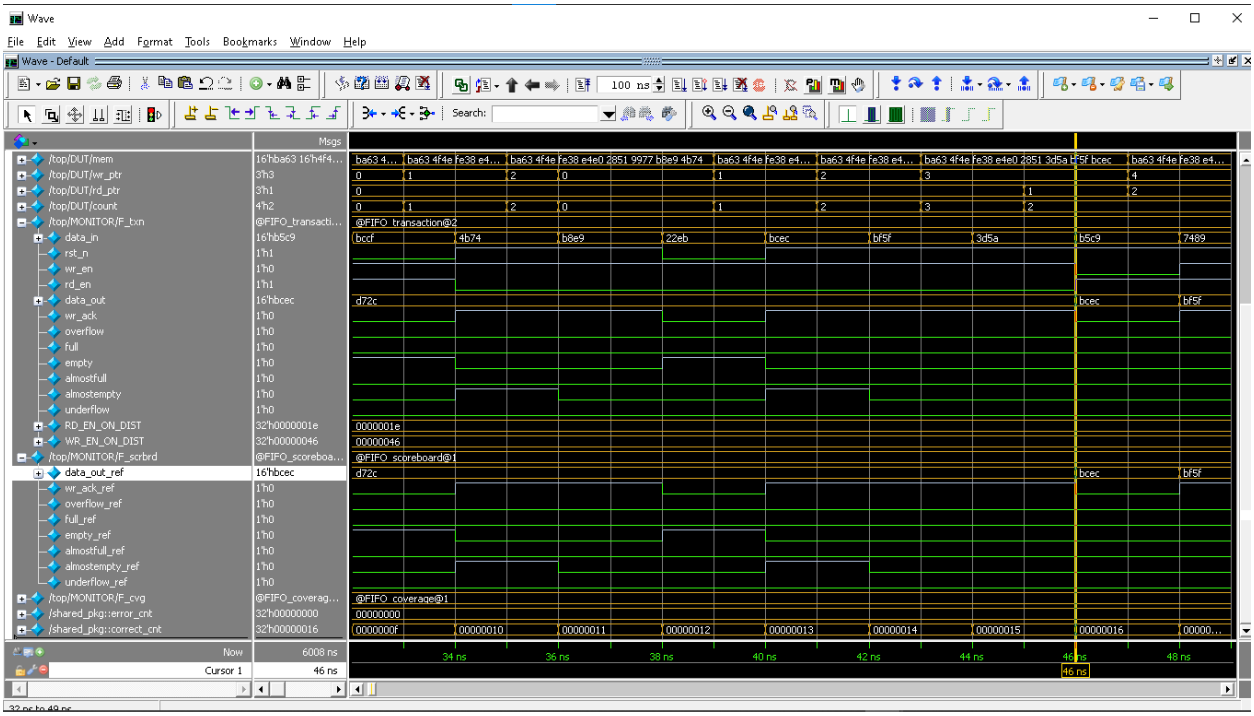
Ref Model Outputs:
data_out=22789
full=0, almostfull= 0
empty=0, almostempty= 0
overflow=0, underflow= 0
wr_ack_ref=1
**

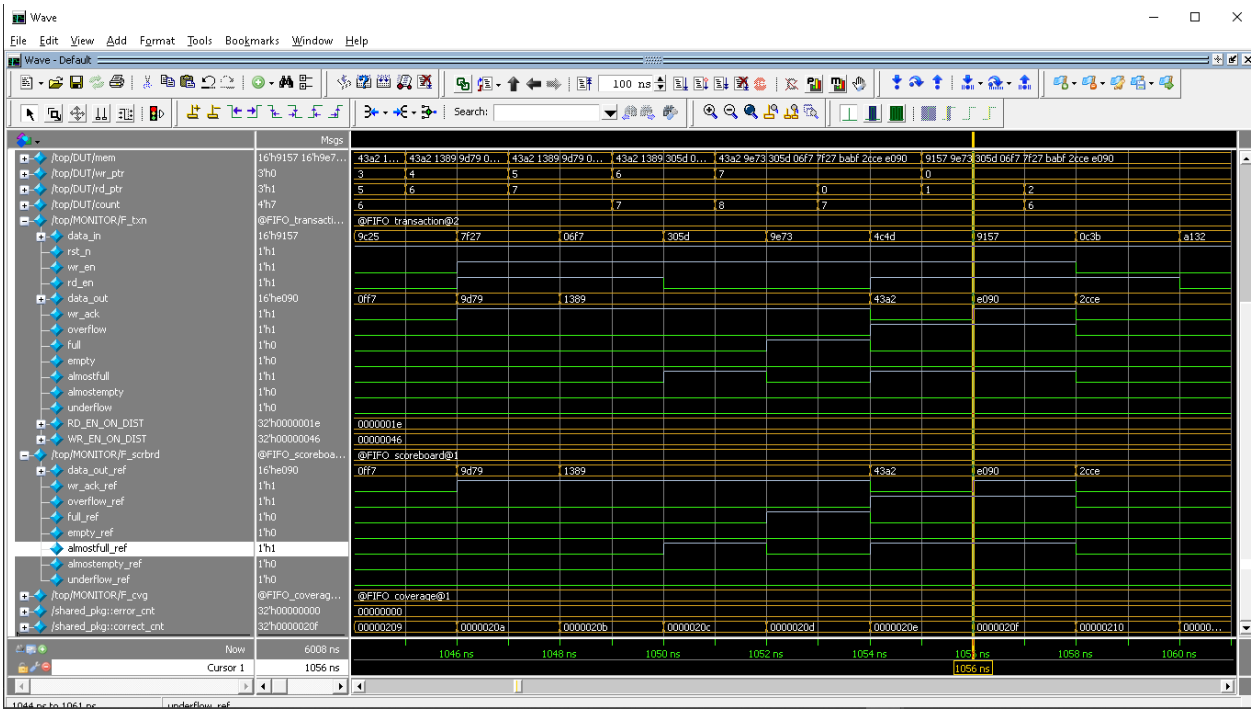
Waveform

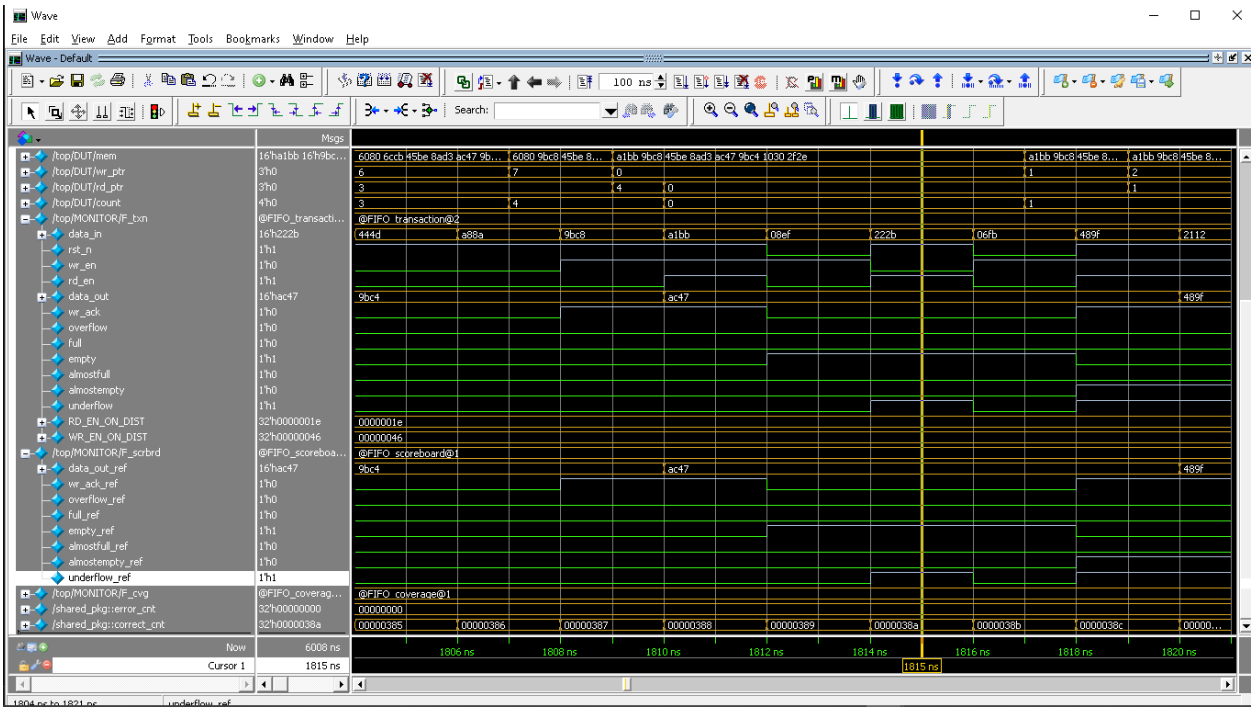

Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Search:
Msgs
/top/DUT/mem
/top/DUT/wr_ptr
/top/DUT/rd_ptr
/top/DUT/count
/top/MONITOR/F_txn
data_in
rst_n
wr_en
rd_en
data_out
wr_ack
overflow
full
empty
almostfull
almostempty
underflow
RD_EN_ON_DIST
WR_EN_ON_DIST
/top/MONITOR/F_scrbrd
data_out_ref
wr_ack_ref
overflow_ref
full_ref
empty_ref
almostfull_ref
almostempty_ref
underflow_ref
/top/MONITOR/F_cvg
/shared_pkg::error_cnt
/shared_pkg::correct_cnt
Now
Cursor 1
6008 ns
3 ns
2 ns
4 ns
6 ns
8 ns
10 ns
12 ns
14 ns
16 ns

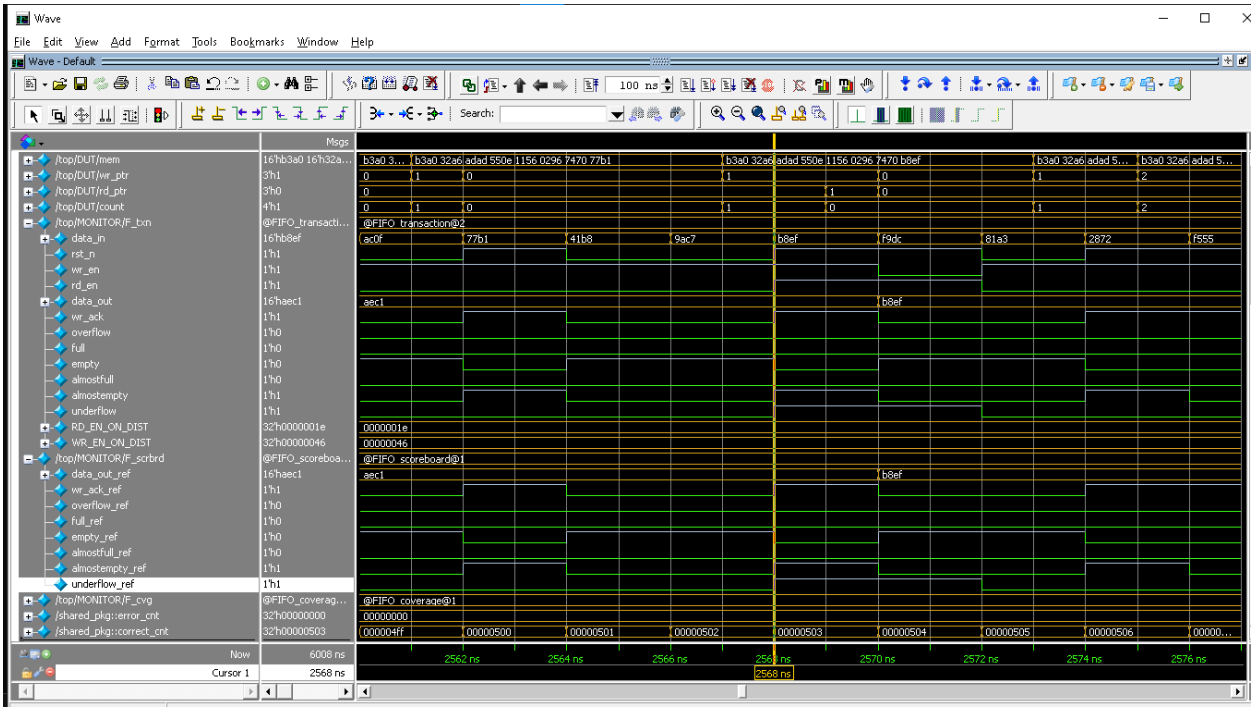

Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Msgs
/top/DUT/mem
/top/DUT/wr_ptr
/top/DUT/rd_ptr
/top/DUT/count
/top/MONITOR/F_txn
data_in
rst_n
wr_en
rd_en
data_out
wr_ack
overflow
full
empty
almostfull
almostempty
underflow
RD_EN_ON_DIST
WR_EN_ON_DIST
/top/MONITOR/F_scrbrd
data_out_ref
wr_ack_ref
overflow_ref
full_ref
empty_ref
almostfull_ref
almostempty_ref
underflow_ref
/top/MONITOR/F_cvg
/shared_pkg::error_cnt
/shared_pkg::correct_cnt
Now
Cursor 1


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Msgs
/top/DUT/mem
/top/DUT/wr_ptr
/top/DUT/rd_ptr
/top/DUT/count
/top/MONITOR/F_txn
data_in
rst_n
wr_en
rd_en
data_out
wr_ack
overflow
full
empty
almostfull
almostempty
underflow
RD_EN_ON_DIST
WR_EN_ON_DIST
/top/MONITOR/F_scrbrd
data_out_ref
wr_ack_ref
overflow_ref
full_ref
empty_ref
almostfull_ref
almostempty_ref
underflow_ref
/top/MONITOR/F_cvg
/shared_pkg::error_cnt
/shared_pkg::correct_cnt
Now
Cursor 1


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Msgs
/top/DUT/mem
/top/DUT/wr_ptr
/top/DUT/rd_ptr
/top/DUT/count
/top/MONITOR/F_txn
data_in
rst_n
wr_en
rd_en
data_out
wr_ack
overflow
full
empty
almostfull
almostempty
underflow
RD_EN_ON_DIST
WR_EN_ON_DIST
/top/MONITOR/F_scrbrd
data_out_ref
wr_ack_ref
overflow_ref
full_ref
empty_ref
almostfull_ref
almostempty_ref
underflow_ref
/top/MONITOR/F_cvg
/shared_pkg::error_cnt
/shared_pkg::correct_cnt
Now
Cursor 1
34 ns
36 ns
38 ns
40 ns
42 ns
44 ns
46 ns
48 ns


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Msgs
/top/DUT/mem
/top/DUT/wr_ptr
/top/DUT/rd_ptr
/top/DUT/count
/top/MONITOR/F_txn
data_in
rst_n
wr_en
rd_en
data_out
wr_ack
overflow
full
empty
almostfull
almostempty
underflow
RD_EN_ON_DIST
WR_EN_ON_DIST
/top/MONITOR/F_scrbrd
data_out_ref
wr_ack_ref
overflow_ref
full_ref
empty_ref
almostfull_ref
almostempty_ref
underflow_ref
/top/MONITOR/F_cvg
/shared_pkg::error_cnt
/shared_pkg::correct_cnt
Now
Cursor 1
1046 ns
1048 ns
1050 ns
1052 ns
1054 ns
1056 ns
1058 ns
1060 ns


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Msgs
/top/DUT/mem
/top/DUT/wr_ptr
/top/DUT/rd_ptr
/top/DUT/count
/top/MONITOR/F_txn
data_in
rst_n
wr_en
rd_en
data_out
wr_ack
overflow
full
empty
almostfull
almostempty
underflow
RD_EN_ON_DIST
WR_EN_ON_DIST
/top/MONITOR/F_scrbrd
data_out_ref
wr_ack_ref
overflow_ref
full_ref
empty_ref
almostfull_ref
almostempty_ref
underflow_ref
/top/MONITOR/F_cvg
/shared_pkg::error_cnt
/shared_pkg::correct_cnt
Now
Cursor 1
1806 ns
1808 ns
1810 ns
1812 ns
1814 ns
1816 ns
1818 ns
1820 ns
1815 ns


Wave
File Edit View Add Format Tools Bookmarks Window Help
Wave - Default
Msgs
/top/DUT/mem
/top/DUT/wr_ptr
/top/DUT/rd_ptr
/top/DUT/count
/top/MONITOR/F_txn
data_in
rst_n
wr_en
rd_en
data_out
wr_ack
overflow
full
empty
almostfull
almostempty
underflow
RD_EN_ON_DIST
WR_EN_ON_DIST
/top/MONITOR/F_scrbrd
data_out_ref
wr_ack_ref
overflow_ref
full_ref
empty_ref
almostfull_ref
almostempty_ref
underflow_ref
/top/MONITOR/F_cvg
/shared_pkg::error_cnt
/shared_pkg::correct_cnt
Now
Cursor 1
2562 ns
2564 ns
2566 ns
2568 ns
2570 ns
2572 ns
2574 ns
2576 ns
2568 ns