SV-FIFO

Name:Moataz ballah Mohsen Osman

Design without BUGS:

```
module FIFO(FIFO_if.DUT f1);

localparam max_fifo_addr = $clog2(f1.FIFO_DEPTH);

reg [f1.FIFO_WIDTH-1:0] mem [f1.FIFO_DEPTH-1:0];

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

always @(posedge f1.clk or negedge f1.rst_n) begin
    if (!f1.rst_n) begin
        wr_ptr <= 0;
        f1.wr_ack <= 0;
        f1.overflow <= 0;
    end
    else if (f1.wr_en && count < f1.FIFO_DEPTH) begin
        mem[wr_ptr] <= f1.data_in;
        f1.wr_ack <= 1;
        wr_ptr <= wr_ptr + 1;
    end
    else begin
        f1.wr_ack <= 0;
        if (f1.full & f1.wr_en)
            f1.overflow <= 1;
        else
            f1.overflow <= 0;
    end
end

always @(posedge f1.clk or negedge f1.rst_n) begin
    if (!f1.rst_n) begin
        rd_ptr <= 0;
        f1.data_out <= 0;
        f1.underflow <= 0;
    end
    else if (f1.rd_en && count != 0) begin
        f1.data_out <= mem[rd_ptr];
        rd_ptr <= rd_ptr + 1;
    end
```

```
    else begin

        if (f1.empty & f1.rd_en)
            f1.underflow <= 1;
        else
            f1.underflow <= 0;
    end

end

always @(posedge f1.clk or negedge f1.rst_n) begin
    if (!f1.rst_n) begin
        count <= 0;
    end
    else begin
        if  ( ({f1.wr_en, f1.rd_en} == 2'b10) && !f1.full)
            count <= count + 1;
        else if ( ({f1.wr_en, f1.rd_en} == 2'b01) && !f1.empty)
            count <= count - 1;
            else if ( ({f1.wr_en, f1.rd_en} == 2'b11) && f1.empty)
            count <= count + 1;
            else if ( ({f1.wr_en, f1.rd_en} == 2'b11) && f1.full)
            count <= count - 1;

    end
end

assign f1.full = (count == f1.FIFO_DEPTH)? 1 : 0;
assign f1.empty = (count == 0)? 1 : 0;
assign f1.almostfull = (count == f1.FIFO_DEPTH-1)? 1 : 0;
assign f1.almostempty = (count == 1)? 1 : 0;
```

Plan:

	A	B	C	D	E
1	Label	design requirment description	stimulas generation	Functional coverage	functionality check
2	FIFO_1	if reset asserted the design output goes low also all the counter and pointers	first its directed then it can appper in randomization	-	withe both golden_model and assertions
3	FIFO_2	write enabeld in most of time and read is low most of time write to the fifo if not full and if it will be overflowed	randomiztion and with constraint of write to be asserted most of time	cover fo wr_en and crossed with wr_ack also rd_en,full,empty,overflow	withe both golden_model and assertions
4	FIFO_3	read enabeld in most of time and write is low most of time read from the fifo if not empty and if it will be underflowed	randomiztion and with constraint of read to be asserted most of time	cover fo rd_en and crossed with wr_ack also wr_en,full,empty,underflow	withe both golden_model and assertions
5	FIFO_4	read and write can be asserted at the same time and can be on of the time	randomiztion and with	cover for all signals	withe both golden_model and assertions

Coverage:

```
package FIFO_coverage_pkg;
import FIFO_transaction_pkg::*;
class FIFO_coverage;
    FIFO_transaction F_cvg_txn;
covergroup cvr_gp;
        wr_en: coverpoint F_cvg_txn.wr_en{
            bins high = {1};
            bins low = {0};
        }
        rd_en: coverpoint F_cvg_txn.rd_en{
            bins high = {1};
            bins low = {0};
        }
        wr_ack: coverpoint F_cvg_txn.wr_ack{
            bins high = {1};
            bins low = {0};
        }
        overflow: coverpoint F_cvg_txn.overflow{
            bins high = {1};
            bins low = {0};
        }
        full: coverpoint F_cvg_txn.full {
            bins high = {1};
            bins low = {0};
        }
        underflow: coverpoint F_cvg_txn.underflow{
            bins high = {1};
            bins low = {0};
        }
        empty: coverpoint F_cvg_txn.empty;
        almostfull: coverpoint F_cvg_txn.almostfull;
        almostempty: coverpoint F_cvg_txn.almostempty;

        wr_re_wr_ack: cross wr_en,rd_en,wr_ack {
            ignore_bins B_0x1 = binsof(wr_en.low) && binsof(rd_en) && binsof(wr_ack.high);
        }
        wr_re_overflow: cross wr_en,rd_en,overflow {
            ignore_bins B_0x1 = binsof(wr_en.low) && binsof(rd_en) && binsof(overflow.high);
        }
        wr_re_full: cross wr_en,rd_en,full {
            ignore_bins B_x11 = binsof(wr_en) && binsof(rd_en.high) && binsof(full.high);
        }
        wr_re_underflow: cross wr_en,rd_en,underflow {
            ignore_bins B_x01 = binsof(wr_en) && binsof(rd_en.low) && binsof(underflow.high);
        }
```

```
        wr_re_empty: cross wr_en,rd_en,empty;
        wr_re_almostfull: cross wr_en,rd_en,almostfull;
        wr_re_almostempty: cross wr_en,rd_en,almostempty;
    endgroup

    function new();
        cvr_gp = new();
    endfunction //new()

    function void sample_data(FIFO_transaction F_txn);
        F_cvg_txn =  F_txn;
        cvr_gp.sample();
    endfunction
endclass //FIFO_coverage
endpackage
```

Interface:

```
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;

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

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

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

endinterface : FIFO_if
```

monitor:

```
import FIFO_transaction_pkg::*;
import FIFO_coverage_pkg::*;
import FIFO_scoreboard_pkg::*;
import shared_pkg::*;
module FIFO_monitor(FIFO_if.MONITOR f_if);

FIFO_transaction fxtn;
FIFO_coverage cov;
FIFO_scoreboard sb;

initial begin
    cov = new();
    sb = new();
    fxtn = new();
    forever begin
        wait(t);      begin
            @(negedge f_if.clk);
            fxtn.data_in = f_if.data_in;
            fxtn.wr_en = f_if.wr_en;
            fxtn.rd_en = f_if.rd_en;
            fxtn.rst_n = f_if.rst_n;
            fxtn.data_out = f_if.data_out;
            fxtn.wr_ack = f_if.wr_ack;
            fxtn.full = f_if.full;
            fxtn.empty = f_if.empty;
            fxtn.almostempty = f_if.almostempty;
            fxtn.almostfull = f_if.almostfull;
            fxtn.overflow = f_if.overflow;
            fxtn.underflow = f_if.underflow;

        fork
        begin
            cov.sample_data(fxtn);
        end
        begin
            sb.check_data(fxtn);
        end
        join
        if (test_finished==1) begin

            $display("test finished, error_count = %0d , correct_count = %0d",error_count,correct_count);
            $stop;

        end
        end
```

Scoreboard:

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


    class FIFO_scoreboard;
    parameter FIFO_WIDTH = 16;
    parameter FIFO_DEPTH = 8;
    parameter max_fifo_addr = $clog2(FIFO_DEPTH);

    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;



    logic [FIFO_WIDTH-1:0] mem [FIFO_DEPTH-1:0];
    logic [max_fifo_addr-1:0] wr_ptr, rd_ptr;
    logic [max_fifo_addr:0] count;


    task check_data(FIFO_transaction F_txn);
        reference_model(F_txn);
        if (F_txn.data_out != data_out_ref) begin
            error_count++;
            $display("Error at %t : data_out = %h, data_out_ref = %h", $time, F_txn.data_out, data_out_ref);
        end
        else begin
            correct_count++;
        end
    endtask
```

```
task reference_model(FIFO_transaction F_ref);

if (!F_ref.rst_n) begin
    wr_ptr = 0;
    wr_ack_ref = 0;
    underflow_ref = 0;
    rd_ptr = 0;
    data_out_ref = 0;
    overflow_ref = 0;
    count = 0;

end
else begin

    if (F_ref.wr_en && count < F_ref.FIFO_DEPTH) begin
    mem[wr_ptr] = F_ref.data_in;
    wr_ack_ref = 1;
    wr_ptr = wr_ptr + 1;
end
else begin
    wr_ack_ref = 0;
    if (full_ref & F_ref.wr_en)
        overflow_ref = 1;
    else
        overflow_ref = 0;
end

 if (F_ref.rd_en && count != 0) begin
    data_out_ref = mem[rd_ptr];
    rd_ptr = rd_ptr + 1;
end

else begin
```

```
    else begin

        if (empty_ref & F_ref.rd_en)
            underflow_ref = 1;
        else
            underflow_ref = 0;
    end

    full_ref = (count == F_ref.FIFO_DEPTH)? 1 : 0;
empty_ref = (count == 0)? 1 : 0;
almostfull_ref = (count == F_ref.FIFO_DEPTH-1)? 1 : 0;
almostempty_ref = (count == 1)? 1 : 0;

        if  ( ({F_ref.wr_en, F_ref.rd_en} == 2'b10) && !full_ref)
            count = count + 1;
        else if ( ({F_ref.wr_en, F_ref.rd_en} == 2'b01) && !empty_ref)
            count = count - 1;
            else if ( ({F_ref.wr_en, F_ref.rd_en} == 2'b11) && empty_ref)
            count = count + 1;
            else if ( ({F_ref.wr_en, F_ref.rd_en} == 2'b11) && full_ref)
            count = count - 1;

    end
    endtask
    endclass
endpackage
```

Testbench:

```
import FIFO_transaction_pkg::*;
import shared_pkg::*;

module FIFO_tb(FIFO_if.TEST f_if);

FIFO_transaction F_txnWrite;
FIFO_transaction F_txnRead;
FIFO_transaction F_txnWriteRead;

task INITIALIZE();
   f_if.rst_n = 1;
   f_if.data_in = 0;
   f_if.wr_en = 0;
   f_if.rd_en = 0;
   test_finished = 0;
endtask

   task RESET();
   @(negedge f_if.clk);
   f_if.rst_n = 0;
   @(negedge f_if.clk);
   f_if.rst_n = 1;

   endtask

initial begin
   F_txnWrite = new();
F_txnRead = new(70,30);
F_txnWriteRead = new(50,50);
   INITIALIZE();
   RESET();
   @(negedge f_if.clk);
   @(negedge f_if.clk);
```

```
    repeat(10000) begin
        @(negedge f_if.clk);
        assert (F_txnWrite.randomize()) ;
        f_if.data_in = F_txnWrite.data_in;
        f_if.wr_en = F_txnWrite.wr_en;
        f_if.rd_en = F_txnWrite.rd_en;
        f_if.rst_n = F_txnWrite.rst_n;
        ->t;
    end

    repeat(10000) begin
        @(negedge f_if.clk);
        assert (F_txnRead.randomize()) ;
        f_if.data_in = F_txnRead.data_in;
        f_if.wr_en = F_txnRead.wr_en;
        f_if.rd_en = F_txnRead.rd_en;
        f_if.rst_n = F_txnRead.rst_n;
        ->t;

    end

    repeat(10000) begin
        @(negedge f_if.clk);
        assert (F_txnWriteRead.randomize()) ;
        f_if.data_in = F_txnWriteRead.data_in;
        f_if.wr_en = F_txnWriteRead.wr_en;
        f_if.rd_en = F_txnWriteRead.rd_en;
        f_if.rst_n = F_txnWriteRead.rst_n;
        ->t;

    end

    test_finished = 1;

end
endmodule
```

Transaction:

```
package FIFO_transaction_pkg;

class FIFO_transaction;
    parameter FIFO_WIDTH = 16;
    parameter FIFO_DEPTH = 8;
    parameter max_fifo_addr = $clog2(FIFO_DEPTH);
    bit clk;
    rand logic [FIFO_WIDTH-1:0] data_in;
    rand logic rst_n, wr_en, rd_en;
    logic [FIFO_WIDTH-1:0] data_out;
    logic wr_ack, overflow;
    logic full, empty, almostfull, almostempty, underflow;
    integer RD_EN_ON_DIST,WR_EN_ON_DIST;


    function new(integer RD_EN_ON_DIST = 30 , integer WR_EN_ON_DIST = 70);
        this.RD_EN_ON_DIST = RD_EN_ON_DIST;
        this.WR_EN_ON_DIST = WR_EN_ON_DIST;
    endfunction

    constraint rst_n_dist {
        rst_n dist {0:=10 , 1:=90};
    }

    constraint wr_en_dist {
        wr_en dist {0:=100-WR_EN_ON_DIST , 1:=WR_EN_ON_DIST};
    }

    constraint rd_en_dist {
        rd_en dist {0:=100-RD_EN_ON_DIST , 1:=RD_EN_ON_DIST};
    }
endclass //FIFO_transaction
endpackage
```

Assertions:

```
always_comb begin
    if (!f1.rst_n) begin
       p1:assert final(f1.full == 0 && f1.empty == 1 && f1.almostfull == 0 && f1.almostempty == 0 &&
       f1.underflow == 0 && f1.overflow == 0 && f1.wr_ack == 0 && rd_ptr == 0 && wr_ptr == 0 && count == 0);
    end
end


property p2;
@(posedge f1.clk) disable iff (!f1.rst_n) (f1.wr_en && !f1.full) |=> ##1 (f1.wr_ack == 1);
endproperty

assert property (p2);
cover property (p2);


property p3;
@(posedge f1.clk) disable iff (!f1.rst_n) (f1.wr_en && f1.full) |=> ##1 (f1.overflow == 1);
endproperty

assert property (p3);
cover property (p3);


property p4;
@(posedge f1.clk) disable iff (!f1.rst_n) (f1.rd_en && f1.empty) |=> ##1 (f1.underflow == 1);
endproperty

assert property (p4);
cover property (p4);


property p5;
@(posedge f1.clk) disable iff (!f1.rst_n) (count == 0) |=> (f1.empty == 1);
endproperty

assert property (p5);
cover property (p5);
```

```
property p6;
@(posedge f1.clk) disable iff (!f1.rst_n) (count == f1.FIFO_DEPTH) |=> (f1.full == 1);
endproperty

assert property (p6);
cover property (p6);


property p7;
@(posedge f1.clk) disable iff (!f1.rst_n) (count == (f1.FIFO_DEPTH - 'b1)) |=> (f1.almostfull == 1);
endproperty

assert property (p7);
cover property (p7);

property p8;
@(posedge f1.clk) disable iff (!f1.rst_n) (count == 1) |=> (f1.almostempty == 1);
endproperty

assert property (p8);
cover property (p8);


property p9;
@(posedge f1.clk) disable iff (!f1.rst_n) ((!f1.rd_en throughout f1.wr_en[->8]) ##0 (wr_ptr == 0)) |=> (wr_ptr == 0);
endproperty

assert property (p9);
cover property (p9);


property p10;
@(posedge f1.clk) disable iff (!f1.rst_n) ((!f1.wr_en throughout f1.rd_en[->8]) ##0 (rd_ptr == 0)) |=> (rd_ptr == 0);
endproperty

assert property (p10);
cover property (p10);

always @(posedge f1.clk) begin
    p11:assert final(rd_ptr < (f1.FIFO_DEPTH) && wr_ptr < (f1.FIFO_DEPTH) && count <= (f1.FIFO_DEPTH));
end

`endif
endmodule
```

Shared package:

```
package shared_pkg;


    bit test_finished;
    int error_count=0;
    int correct_count=0;
    event t;
endpackage
```

TOP:

```
module top_FIFO();

bit clk ;
always  #1 clk = ~clk ;

FIFO_if f_if(clk);

FIFO DUT(f_if);

FIFO_tb tb (f_if);

FIFO_monitor mon (f_if);



endmodule
```

DO:

```
vlib work
vlog *.*v +cover -covercells +define+SIM
vsim -voptargs=+acc work.top_FIFO -cover
coverage save FIFO.ucdb -onexit
run 0
add wave /top_FIFO/f_if/*
add wave -position insertpoint  \
sim:/top_FIFO/DUT/mem \
sim:/top_FIFO/DUT/wr_ptr \
sim:/top_FIFO/DUT/rd_ptr \
sim:/top_FIFO/DUT/count
run -all
#quit -sim
#vcover report FIFO.ucdb -details -annotate -all
```

Code coverage:

```
# ================================================================================
# === File: FIFO.sv
# ================================================================================
# Statement Coverage:
#     Enabled Coverage             Active      Hits    Misses % Covered
#     ----------------             ------      ----    ------ ---------
#     Stmts                            30        30         0     100.0
#
# ================================Statement Details================================
#

Branch Coverage:
    Enabled Coverage            Active      Hits    Misses % Covered
    ----------------            ------      ----    ------ ---------
    Branches                        27        27         0     100.0

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

Branch Coverage for file FIFO.sv --
NOTE: The modification timestamp for source file 'FIFO.sv' has been altered since compilation.

------------------------------------IF Branch------------------------------------
    18                                  32659     Count coming in to IF
    18              1                    5634       mem[wr_ptr] <= fl.data_in;
    23              1                   12584       fl.wr_ack <= 0;
    28              1                   14441      end
Branch totals: 3 hits of 3 branches = 100.0%

------------------------------------IF Branch------------------------------------
    30                                  14441     Count coming in to IF
    30              1                     701
    32              1                   13740      if (!fl.rst_n) begin
Branch totals: 2 hits of 2 branches = 100.0%

------------------------------------IF Branch------------------------------------
    38                                  32659     Count coming in to IF
    38              1                    5634       fl.data_out <= mem[rd_ptr];
    43              1                    7473
    48              1                   19552      end
Branch totals: 3 hits of 3 branches = 100.0%
```

```
------------------------------------IF Branch------------------------------------
    30                                  14441     Count coming in to IF
    30              1                     701
    32              1                   13740      if (!fl.rst_n) begin
Branch totals: 2 hits of 2 branches = 100.0%

------------------------------------IF Branch------------------------------------
    38                                  32659     Count coming in to IF
    38              1                    5634       fl.data_out <= mem[rd_ptr];
    43              1                    7473
    48              1                   19552      end
Branch totals: 3 hits of 3 branches = 100.0%

------------------------------------IF Branch------------------------------------
    50                                  19552     Count coming in to IF
    50              1                    5918     end
    52              1                   13634     always @(posedge fl.clk or negedge fl.rst_n) begin
Branch totals: 2 hits of 2 branches = 100.0%

------------------------------------IF Branch------------------------------------
    59                                  28860     Count coming in to IF
    59              1                    5539       else if ( ({fl.wr_en, fl.rd_en} == 2'b01) && !fl.empty)
    62              1                   23321         count <= count + 1;
Branch totals: 2 hits of 2 branches = 100.0%

------------------------------------IF Branch------------------------------------
    63                                  23321     Count coming in to IF
    63              1                    6917       else if ( ({fl.wr_en, fl.rd_en} == 2'b11) && fl.full)
    65              1                    3775
    67              1                    2179     end
    69              1                     210     assign fl.full = (count == fl.FIFO_DEPTH)? 1 : 0;
                                          10240     All False Count
Branch totals: 5 hits of 5 branches = 100.0%

------------------------------------IF Branch------------------------------------
    75                                  14928     Count coming in to IF
    75              1                     420
    75              2                   14508
Branch totals: 2 hits of 2 branches = 100.0%


#
# ------------------------------------IF Branch------------------------------------
#     75                                  14928     Count coming in to IF
#     75              1                     420
#     75              2                   14508
# Branch totals: 2 hits of 2 branches = 100.0%
#
# ------------------------------------IF Branch------------------------------------
#     76                                  14928     Count coming in to IF
#     76              1                    4099
#     76              2                   10829
# Branch totals: 2 hits of 2 branches = 100.0%
#
# ------------------------------------IF Branch------------------------------------
#     77                                  14928     Count coming in to IF
#     77              1                     607     always_comb begin
#     77              2                   14321     always_comb begin
# Branch totals: 2 hits of 2 branches = 100.0%
#
# ------------------------------------IF Branch------------------------------------
#     78                                  14928     Count coming in to IF
#     78              1                    4789         if (!fl.rst_n) begin
#     78              2                   10139         if (!fl.rst_n) begin
# Branch totals: 2 hits of 2 branches = 100.0%
#
# ------------------------------------IF Branch------------------------------------
#     84                                  30352     Count coming in to IF
#     84              1                    5190
#                                         25162     All False Count
# Branch totals: 2 hits of 2 branches = 100.0%
#
#
# Condition Coverage:
#     Enabled Coverage            Active   Covered    Misses % Covered
#     ----------------            ------      ----    ------ ---------
#     FEC Condition Terms              20        18         2      90.0
#
```

```
# ================================Condition Details================================
#
# Condition Coverage for file FIFO.sv --
# NOTE: The modification timestamp for source file 'FIFO.sv' has been altered since compilation.
#
# ----------------Focused Condition View-------------------
# Line       23 Item    1  (fl.wr_en && (count < fl.FIFO_DEPTH))
# Condition totals: 2 of 2 input terms covered = 100.0%
#
#               Input Term   Covered  Reason for no coverage   Hint
#              -----------  --------  ----------------------  --------------
#                 fl.wr_en         Y
#   (count < fl.FIFO_DEPTH)        Y
#
#      Rows:       Hits  FEC Target                Non-masking condition(s)
#  ---------  ---------  --------------------      -------------------------
#   Row   1:          1  fl.wr_en_0                -
#   Row   2:          1  fl.wr_en_1                (count < fl.FIFO_DEPTH)
#   Row   3:          1  (count < fl.FIFO_DEPTH)_0  fl.wr_en
#   Row   4:          1  (count < fl.FIFO_DEPTH)_1  fl.wr_en
#
# ----------------Focused Condition View-------------------
# Line       30 Item    1  (fl.full & fl.wr_en)
# Condition totals: 1 of 2 input terms covered = 50.0%
#
#   Input Term   Covered  Reason for no coverage   Hint
#  -----------  --------  ----------------------  --------------
#      fl.full         N  '_0' not hit             Hit '_0'
#     fl.wr_en         Y
#
#      Rows:       Hits  FEC Target            Non-masking condition(s)
#  ---------  ---------  --------------------  -------------------------
#   Row   1:    ***0***  fl.full_0             fl.wr_en
#   Row   2:          1  fl.full_1             fl.wr_en
#   Row   3:          1  fl.wr_en_0            fl.full
#   Row   4:          1  fl.wr_en_1            fl.full
#
```

```
#   Input Term   Covered  Reason for no coverage   Hint
#  -----------  --------  ----------------------  --------------
#      f1.full         N  '_0' not hit             Hit '_0'
#     f1.wr_en         Y
#
#      Rows:       Hits  FEC Target              Non-masking condition(s)
#  ---------  ---------  --------------------  --------------------------
#   Row   1:     ***0***  f1.full_0             f1.wr_en
#   Row   2:           1  f1.full_1             f1.wr_en
#   Row   3:           1  f1.wr_en_0            f1.full
#   Row   4:           1  f1.wr_en_1            f1.full
#
# ----------------Focused Condition View-------------------
# Line       43 Item    1  (f1.rd_en && (count != 0))
# Condition totals: 2 of 2 input terms covered = 100.0%
#
#     Input Term   Covered  Reason for no coverage   Hint
#    -----------  --------  ----------------------  --------------
#       f1.rd_en         Y
#   (count != 0)         Y
#
#      Rows:       Hits  FEC Target              Non-masking condition(s)
#  ---------  ---------  --------------------  --------------------------
#   Row   1:           1  f1.rd_en_0            -
#   Row   2:           1  f1.rd_en_1            (count != 0)
#   Row   3:           1  (count != 0)_0        f1.rd_en
#   Row   4:           1  (count != 0)_1        f1.rd_en
#
# ----------------Focused Condition View-------------------
# Line       50 Item    1  (f1.empty & f1.rd_en)
# Condition totals: 1 of 2 input terms covered = 50.0%
#
#   Input Term   Covered  Reason for no coverage   Hint
#  -----------  --------  ----------------------  --------------
#     f1.empty         N  '_0' not hit             Hit '_0'
#     f1.rd_en         Y
#
#      Rows:       Hits  FEC Target              Non-masking condition(s)
#  ---------  ---------  --------------------  --------------------------
#   Row   1:     ***0***  f1.empty_0            f1.rd_en
#   Row   2:           1  f1.empty_1
#   Row   3:           1  f1.rd_en_0
#   Row   4:           1  f1.rd_en_1
#
# ================================Toggle Details================================
#
# Toggle Coverage for File FIFO.sv --
#
#       Line                                  Node      1H->0L      0L->1H  "Coverage"
# ----------------------------------------------------------------------------------
#         14                            wr_ptr[2]           1           1      100.00
#         14                            wr_ptr[1]           1           1      100.00
#         14                            wr_ptr[0]           1           1      100.00
#         14                            rd_ptr[2]           1           1      100.00
#         14                            rd_ptr[1]           1           1      100.00
#         14                            rd_ptr[0]           1           1      100.00
#         15                             count[3]           1           1      100.00
#         15                             count[2]           1           1      100.00
#         15                             count[1]           1           1      100.00
#         15                             count[0]           1           1      100.00
#
# Total Node Count     =         10
# Toggled Node Count   =         10
# Untoggled Node Count =          0
#
# Toggle Coverage      =      100.0% (20 of 20 bins)
#
```

Functional coverage:

COVERGROUP COVERAGE:

Covergroup	Metric	Goal	Status
TYPE /FIFO_coverage_pkg/FIFO_coverage/cvr_gp	100.0%	100	Covered
covered/total bins:	66	66	
missing/total bins:	0	66	
% Hit:	100.0%	100	
Coverpoint cvr_gp::wr_en	100.0%	100	Covered
covered/total bins:	2	2	
missing/total bins:	0	2	
% Hit:	100.0%	100	
bin high	14814	1	Covered
bin low	15186	1	Covered
Coverpoint cvr_gp::rd_en	100.0%	100	Covered
covered/total bins:	2	2	
missing/total bins:	0	2	
% Hit:	100.0%	100	
bin high	14818	1	Covered
bin low	15182	1	Covered
Coverpoint cvr_gp::wr_ack	100.0%	100	Covered
covered/total bins:	2	2	
missing/total bins:	0	2	
% Hit:	100.0%	100	
bin high	12584	1	Covered
bin low	17416	1	Covered
Coverpoint cvr_gp::overflow	100.0%	100	Covered
covered/total bins:	2	2	
missing/total bins:	0	2	
% Hit:	100.0%	100	
bin high	852	1	Covered
bin low	29148	1	Covered
Coverpoint cvr_gp::full	100.0%	100	Covered
covered/total bins:	2	2	
missing/total bins:	0	2	
% Hit:	100.0%	100	
bin high	1142	1	Covered
bin low	28858	1	Covered
Coverpoint cvr_gp::underflow	100.0%	100	Covered
covered/total bins:	2	2	
missing/total bins:	0	2	
% Hit:	100.0%	100	
bin high	7350	1	Covered
bin low	22650	1	Covered

Coverpoint cvr_gp::empty	100.0%	100	Covered
covered/total bins:	2	2	
missing/total bins:	0	2	
% Hit:	100.0%	100	
bin auto[0]	18696	1	Covered
bin auto[1]	11304	1	Covered
Coverpoint cvr_gp::almostfull	100.0%	100	Covered
covered/total bins:	2	2	
missing/total bins:	0	2	
% Hit:	100.0%	100	
bin auto[0]	29011	1	Covered
bin auto[1]	989	1	Covered
Coverpoint cvr_gp::almostempty	100.0%	100	Covered
covered/total bins:	2	2	
missing/total bins:	0	2	
% Hit:	100.0%	100	
bin auto[0]	21921	1	Covered
bin auto[1]	8079	1	Covered
Cross cvr_gp::wr_re_wr_ack	100.0%	100	Covered
covered/total bins:	6	6	
missing/total bins:	0	6	
% Hit:	100.0%	100	
bin <high,high,high>	5667	1	Covered
bin <high,low,high>	6917	1	Covered
bin <high,high,low>	873	1	Covered
bin <low,high,low>	8278	1	Covered
bin <high,low,low>	1357	1	Covered
bin <low,low,low>	6908	1	Covered
ignore_bin B_0x1	0		ZERO
Cross cvr_gp::wr_re_overflow	100.0%	100	Covered
covered/total bins:	6	6	
missing/total bins:	0	6	
% Hit:	100.0%	100	
bin <high,high,high>	254	1	Covered
bin <high,low,high>	598	1	Covered
bin <high,high,low>	6286	1	Covered
bin <low,high,low>	8278	1	Covered
bin <high,low,low>	7676	1	Covered
bin <low,low,low>	6908	1	Covered
ignore_bin B_0x1	0		ZERO

```
    Cross cvr_gp::wr_re_full                          100.0%        100    Covered
        covered/total bins:                                6          6
        missing/total bins:                                0          6
        % Hit:                                         100.0%        100
        bin <high,low,high>                              911          1    Covered
        bin <low,low,high>                               231          1    Covered
        bin <high,high,low>                             6540          1    Covered
        bin <high,low,low>                              7363          1    Covered
        bin <low,high,low>                              8278          1    Covered
        bin <low,low,low>                               6677          1    Covered
        ignore_bin B_x11                                   0               ZERO
    Cross cvr_gp::wr_re_underflow                     100.0%        100    Covered
        covered/total bins:                                6          6
        missing/total bins:                                0          6
        % Hit:                                         100.0%        100
        bin <high,high,high>                            2702          1    Covered
        bin <low,high,high>                             4648          1    Covered
        bin <high,high,low>                             3838          1    Covered
        bin <high,low,low>                              8274          1    Covered
        bin <low,high,low>                              3630          1    Covered
        bin <low,low,low>                               6908          1    Covered
        ignore_bin B_x01                                   0               ZERO
    Cross cvr_gp::wr_re_empty                         100.0%        100    Covered
        covered/total bins:                                8          8
        missing/total bins:                                0          8
        % Hit:                                         100.0%        100
        bin <high,high,auto[0]>                         5877          1    Covered
        bin <high,high,auto[1]>                          663          1    Covered
        bin <low,high,auto[0]>                          1523          1    Covered
        bin <low,high,auto[1]>                          6755          1    Covered
        bin <high,low,auto[0]>                          7408          1    Covered
        bin <high,low,auto[1]>                           866          1    Covered
        bin <low,low,auto[0]>                           3888          1    Covered
        bin <low,low,auto[1]>                           3020          1    Covered
    Cross cvr_gp::wr_re_almostfull                    100.0%        100    Covered
        covered/total bins:                                8          8
        missing/total bins:                                0          8
        % Hit:                                         100.0%        100
        bin <high,high,auto[0]>                         6153          1    Covered
        bin <high,high,auto[1]>                          387          1    Covered
        bin <low,high,auto[0]>                          8181          1    Covered
        bin <low,high,auto[1]>                            97          1    Covered
        bin <high,low,auto[0]>                          7974          1    Covered
        bin <high,low,auto[1]>                           300          1    Covered
        bin <low,low,auto[0]>                           6703          1    Covered
        bin <low,low,auto[1]>                            205          1    Covered
    Cross cvr_gp::wr_re_almostempty                   100.0%        100    Covered
        covered/total bins:                                8          8
        missing/total bins:                                0          8
        % Hit:                                         100.0%        100
        bin <high,high,auto[0]>                         2766          1    Covered
        bin <high,high,auto[1]>                         3774          1    Covered
        bin <low,high,auto[0]>                          7588          1    Covered
        bin <low,high,auto[1]>                           690          1    Covered
        bin <high,low,auto[0]>                          6354          1    Covered
        bin <high,low,auto[1]>                          1920          1    Covered
        bin <low,low,auto[0]>                           5213          1    Covered
        bin <low,low,auto[1]>                           1695          1    Covered
 CLASS FIFO_coverage

TOTAL COVERGROUP COVERAGE: 100.0%  COVERGROUP TYPES: 1
```

```
DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------------
Name                                    Design Design   Lang File(Line)      Count Status
                                        Unit   UnitType
--------------------------------------------------------------------------------------------------
/top_FIFO/DUT/cover__p10                FIFO   Verilog  SVA  FIFO.sv(153)       21 Covered
/top_FIFO/DUT/cover__p9                 FIFO   Verilog  SVA  FIFO.sv(145)       36 Covered
/top_FIFO/DUT/cover__p8                 FIFO   Verilog  SVA  FIFO.sv(137)     7298 Covered
/top_FIFO/DUT/cover__p7                 FIFO   Verilog  SVA  FIFO.sv(130)      894 Covered
/top_FIFO/DUT/cover__p6                 FIFO   Verilog  SVA  FIFO.sv(122)     1029 Covered
/top_FIFO/DUT/cover__p5                 FIFO   Verilog  SVA  FIFO.sv(114)    10176 Covered
/top_FIFO/DUT/cover__p4                 FIFO   Verilog  SVA  FIFO.sv(106)     5372 Covered
/top_FIFO/DUT/cover__p3                 FIFO   Verilog  SVA  FIFO.sv(98)       636 Covered
/top_FIFO/DUT/cover__p2                 FIFO   Verilog  SVA  FIFO.sv(90)      11369 Covered

TOTAL DIRECTIVE COVERAGE: 100.0%  COVERS: 9
```

/top_FIFO/DUT/cover__p10	SVA	✓	Off	21	1	Unli...	1	100%	✓	0	0
/top_FIFO/DUT/cover__p9	SVA	✓	Off	36	1	Unli...	1	100%	✓	0	0
/top_FIFO/DUT/cover__p8	SVA	✓	Off	7298	1	Unli...	1	100%	✓	0	0
/top_FIFO/DUT/cover__p7	SVA	✓	Off	894	1	Unli...	1	100%	✓	0	0
/top_FIFO/DUT/cover__p6	SVA	✓	Off	1029	1	Unli...	1	100%	✓	0	0
/top_FIFO/DUT/cover__p5	SVA	✓	Off	10176	1	Unli...	1	100%	✓	0	0
/top_FIFO/DUT/cover__p4	SVA	✓	Off	5372	1	Unli...	1	100%	✓	0	0
/top_FIFO/DUT/cover__p3	SVA	✓	Off	636	1	Unli...	1	100%	✓	0	0
/top_FIFO/DUT/cover__p2	SVA	✓	Off	11369	1	Unli...	1	100%	✓	0	0