FIFO

SV Project

By: Khaled Ahmed Hamed

Under supervision of Eng. Kareem Waseem

For more details please visit: Github repo

Design has 4 bugs:

1. Reset signals overflow, wr_ack and underflow (data_out not to be included)
2. Unhandled 2 cases:
   - If a read and write enables were high and the FIFO was empty, only writing will take place.
   - If a read and write enables were high and the FIFO was full, only reading will take place.
3. underflow is sequential not combinational
4. almostfull flag: FIFO_DEPTH-2 corrected to FIFO_DEPTH-1

Verification plan:

Label	Description	Stimulus Generation	Functional Coverage	Functionality Check
LABEL1	When the rst_n is asserted, All flags & internal signals should equal 0	Randomized less often 95 off & 5% on during the simulation	.	Immediate assertion to check async rst_n functionality as it is async
LABEL2	When rst_n is deactivated & the FIFO is full of elements (conut=depth), The full flag should be high	Randomized, Write enable to be high with distribution of the value WR_EN_ON_DIST and to be low with 100-WR_EN_ON_DISTduring the simulation	Cross coverage: WRITE_READ_FULL_CROSS, covers write & read enables and full flag	Immediate assertion to check full flag functionality as it is combinational
LABEL3	When rst_n is deactivated & the FIFO has one element left(conut=depth-1), The almostfull flag should be high	Randomized, Write enable to be high with distribution of the value WR_EN_ON_DIST and to be low with 100-WR_EN_ON_DISTduring the simulation	Cross coverage: WRITE_READ_ALMOST_FULL_CROSS, covers write & read enables and almostfull flag	Immediate assertion to check almostfull flag functionality as it is combinational
LABEL4	When rst_n is deactivated & the FIFO has no element inside (conut=0), The empty flag should be high	Randomized, Read enable to be high with distribution of the value RD_EN_ON_DIST and to be low with 100-RD_EN_ON_DIST during the simulation	Cross coverage: WRITE_READ_EMPTY_CROSS, covers write & read enables and empty flag	Immediate assertion to check empty flag functionality as it is combinational
LABEL5	When rst_n is deactivated & the FIFO has only one element inside (conut=1), The empty flag should be high	Randomized, Read enable to be high with distribution of the value RD_EN_ON_DIST and to be low with 100-RD_EN_ON_DIST during the simulation	Cross coverage: WRITE_READ_ALMSOT_EMPTY_CROSS, covers write & read enables and almostempty flag	Immediate assertion to check almostempty flag functionality as it is combinational
LABEL6	When rst_n is deactivated, You want to read & the FIFO is not full, The wr_ack flag should be high and wr_ptr should increment	Randomized, Write enable to be high with distribution of the value WR_EN_ON_DIST and to be low with 100-WR_EN_ON_DISTduring the simulation	Cross coverage: WRITE_READ_WR_ACK_CROSS, covers write & read enables and wr_ack flag	Concurrent assertion to check wr_ptr & wr_ack flag functionality as they are sequential
LABEL7	When rst_n is deactivated, You want to read & the FIFO is full, The overflow flag should be high	Randomized, Write enable to be high with distribution of the value WR_EN_ON_DIST and to be low with 100-WR_EN_ON_DISTduring the simulation	Cross coverage: WRITE_READ_OVERFLOW_CROSS, covers write & read enables and overflow flag	Concurrent assertion to check overflow flag functionality as it is sequential
LABEL8	When rst_n is deactivated, You want to read & the FIFO is not empty rd_ptr should increment and data_out should equal mem[rd_ptr]	Randomized, Read enable to be high with distribution of the value RD_EN_ON_DIST and to be low with 100-RD_EN_ON_DIST during the simulation	.	Concurrent assertion to check rd_ptr functionality as it is sequential, data_out checked against refernce model to check functionality
LABEL9	When rst_n is deactivated, You want to read & the FIFO is empty underflow flag should be high	Randomized, Read enable to be high with distribution of the value RD_EN_ON_DIST and to be low with 100-RD_EN_ON_DIST during the	Cross coverage: WRITE_READ_UNDERFLOW_CROSS, covers write & read enables and overflow flag	Concurrent assertion to check underflow flag functionality as it is sequential

RTL code snippets with assertions:

```
1    //////////////////////////////////////////////////////////////////////////
2    // Author: Kareem Waseem
3    // Verification Engineer: Khaled A. Hamed 🔶
4    // Course: Digital Verification using SV & UVM
5    // Description: FIFO Design
6    //////////////////////////////////////////////////////////////////////////
7    module FIFO (FIFO_interface.DUT FIFO_IF);
8
9    reg [FIFO_IF.max_fifo_addr-1:0] wr_ptr, rd_ptr;
10   reg [FIFO_IF.max_fifo_addr:0] count;
11
12   reg [FIFO_IF.FIFO_WIDTH-1:0] mem [FIFO_IF.FIFO_DEPTH-1:0];
13
14   always @(posedge FIFO_IF.clk or negedge FIFO_IF.rst_n) begin
15       if (!FIFO_IF.rst_n) begin
16           wr_ptr <= 0;
17           // Bug detected: Reset signals FIFO_IF.overflow & FIFO_IF.wr_ack
18           FIFO_IF.overflow <= 0;
19           FIFO_IF.wr_ack <= 0;
20       end
21       else if (FIFO_IF.wr_en && count < FIFO_IF.FIFO_DEPTH) begin
22           mem[wr_ptr] <= FIFO_IF.data_in;
23           FIFO_IF.wr_ack<=1;
24           wr_ptr <= wr_ptr + 1;
25       end
26       else begin
27           FIFO_IF.wr_ack <= 0;
28           if (FIFO_IF.full & FIFO_IF.wr_en)
29               FIFO_IF.overflow <= 1;
30           else
31               FIFO_IF.overflow <= 0;
32       end
33   end
34
35   always @(posedge FIFO_IF.clk or negedge FIFO_IF.rst_n) begin
36       if (!FIFO_IF.rst_n) begin
37           rd_ptr <= 0;
38           // Bug detected: Reset signals FIFO_IF.underflow
39           FIFO_IF.underflow <= 0;
40       end
41       else if (FIFO_IF.rd_en && count != 0) begin
42           FIFO_IF.data_out <= mem[rd_ptr];
43           rd_ptr <= rd_ptr + 1;
44       end
45       // Handled FIFO_IF.underflow behaviour when turned from combinational to sequential
46       else begin
47           if (FIFO_IF.empty & FIFO_IF.rd_en)
48               FIFO_IF.underflow <= 1;
49           else
50               FIFO_IF.underflow <= 0;
51       end
52   end
53
```

```
54   always @(posedge FIFO_IF.clk or negedge FIFO_IF.rst_n) begin
55       if (!FIFO_IF.rst_n) begin
56           count <= 0;
57       end
58       else begin
59           if  ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b10) && !FIFO_IF.full)
60               count <= count + 1;
61           else if ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b01) && !FIFO_IF.empty)
62               count <= count - 1;
63           // Bug detected: Unhandled case,  If a read and write enables were high and the FIFO was FIFO_IF.empty, only writing will take place.
64           else if ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b11) && FIFO_IF.empty)
65               count <= count + 1;
66           // Bug detected: Unhandled cases,  If a read and write enables were high and the FIFO was FIFO_IF.full, only reading will take place.
67           else if ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b11) && FIFO_IF.full)
68               count <= count - 1;
69       end
70   end
71
72   assign FIFO_IF.full = (count == FIFO_IF.FIFO_DEPTH)? 1 : 0;
73   assign FIFO_IF.empty = (count == 0)? 1 : 0;
74   assign FIFO_IF.almostfull = (count == FIFO_IF.FIFO_DEPTH-1)? 1 : 0; // Bug detected: FIFO_IF.FIFO_DEPTH-2 --> FIFO_IF.FIFO_DEPTH-1
75   assign FIFO_IF.almostempty = (count == 1)? 1 : 0;
76
```

```verilog
77     // Guarded assertions
78   `ifdef SIM
79       // Properties, Assertions & Covers
80       always_comb begin
81           if(!FIFO_IF.rst_n)
82           reset_1_assertion: assert final ((!FIFO_IF.wr_ack)&&(!FIFO_IF.overflow)&&(!FIFO_IF.underflow)&&(!wr_ptr)&&(!rd_ptr)&&(!count));
83           reset_1_cover: cover final ((!FIFO_IF.wr_ack)&&(!FIFO_IF.overflow)&&(!FIFO_IF.underflow)&&(!wr_ptr)&&(!rd_ptr)&&(!count));
84       end
85
86       always_comb begin
87           if((FIFO_IF.rst_n)&&(count == FIFO_IF.FIFO_DEPTH))
88           full_assertion: assert final (FIFO_IF.full);
89           full_cover: cover (FIFO_IF.full);
90       end
91
92       always_comb begin
93           if((FIFO_IF.rst_n)&&(count == 0))
94           empty_assertion: assert final (FIFO_IF.empty);
95           empty_cover: cover (FIFO_IF.empty);
96       end
97
98       always_comb begin
99           if((FIFO_IF.rst_n)&&(count == FIFO_IF.FIFO_DEPTH-1))
100          almostfull_assertion: assert final (FIFO_IF.almostfull);
101          almostfull_cover: cover (FIFO_IF.almostfull);
102      end
103
104      always_comb begin
105          if((FIFO_IF.rst_n)&&(count == 1))
106          almostempty_assertion: assert final (FIFO_IF.almostempty);
107          almostempty_cover: cover (FIFO_IF.almostempty);
108      end
109
110      property P1;
111          @(posedge FIFO_IF.clk or negedge FIFO_IF.rst_n)
112          (!FIFO_IF.rst_n) |-> ((!FIFO_IF.wr_ack)&&(!FIFO_IF.overflow)&&(!FIFO_IF.underflow)&&(!wr_ptr)&&(!rd_ptr)&&(!count));
113      endproperty
114
115      property P2;
116          @(posedge FIFO_IF.clk) disable iff(!FIFO_IF.rst_n)
117          (FIFO_IF.wr_en && !FIFO_IF.full ) |=> ((FIFO_IF.wr_ack)&&((wr_ptr==$past(wr_ptr)+1)||(wr_ptr==0 && $past(wr_ptr) +1 == 8)));
118      endproperty
119
120      property P3;
121          @(posedge FIFO_IF.clk) disable iff(!FIFO_IF.rst_n)
122          (FIFO_IF.full & FIFO_IF.wr_en) |=> (FIFO_IF.overflow);
123      endproperty
124
125
126      property P4;
127          @(posedge FIFO_IF.clk) disable iff(!FIFO_IF.rst_n)
128          (FIFO_IF.rd_en && count != 0) |=> ((rd_ptr==$past(rd_ptr)+1)||(rd_ptr==0 && $past(rd_ptr) +1 == 8));
129      endproperty
130

131      property P5;
132          @(posedge FIFO_IF.clk) disable iff(!FIFO_IF.rst_n)
133          (FIFO_IF.empty & FIFO_IF.rd_en) |=> (FIFO_IF.underflow);
134      endproperty
135
136      property P6;
137          @(posedge FIFO_IF.clk) disable iff(!FIFO_IF.rst_n)
138          (({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b10) && !FIFO_IF.full)   |=> ((count==$past(count)+1));
139      endproperty
140
141      property P7;
142          @(posedge FIFO_IF.clk) disable iff(!FIFO_IF.rst_n)
143          ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b01) && !FIFO_IF.empty)  |=> ((count==$past(count)-1));
144      endproperty
145
146      property P8;
147          @(posedge FIFO_IF.clk) disable iff(!FIFO_IF.rst_n)
148          ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b11) && FIFO_IF.empty)  |=> ((count==$past(count)+1));
149      endproperty
150
151      property P9;
152          @(posedge FIFO_IF.clk) disable iff(!FIFO_IF.rst_n)
153          ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b11) && FIFO_IF.full)  |=> ((count==$past(count)-1));
154      endproperty
155
156      reset_2_assertion: assert property(P1);
157      reset_2_cover: cover property(P1);
158
159      write_assertion: assert property(P2);
160      write_cover: cover property(P2);
161
162      overflow_assertion: assert property(P3);
163      overflow_cover: cover property(P3);
164
165      read_assertion: assert property(P4);
166      read_cover: cover property(P4);
167
168      underflow_assertion: assert property(P5);
169      underflow_cover: cover property(P5);
170
171      write_not_full_assertion: assert property(P6);
172      write_not_full_cover: cover property(P6);
173
174      read_not_empty_assertion: assert property(P7);
175      read_not_empty_cover: cover property(P7);
176
177      read_write_empty_assertion: assert property(P8);
178      read_write_empty_cover: cover property(P8);
179
180      read_write_full_assertion: assert property(P9);
181      read_write_full_cover: cover property(P9);
182  `endif
183  endmodule
```

Interface code snippet:

```systemverilog
1   interface FIFO_interface (input bit clk);
2
3       // Parameters
4       parameter FIFO_WIDTH = 16;
5       parameter FIFO_DEPTH = 8;
6       localparam max_fifo_addr = $clog2(FIFO_DEPTH);
7
8       // Signals
9       logic [FIFO_WIDTH-1:0] data_in;
10      logic rst_n, wr_en, rd_en;
11      logic [FIFO_WIDTH-1:0] data_out;
12      logic wr_ack, overflow, underflow;
13      logic full, empty, almostfull, almostempty;
14
15      // Modports
16      modport DUT (input clk, data_in, rst_n, wr_en, rd_en, output data_out, wr_ack, overflow, underflow, full, empty, almostfull, almostempty);
17
18      modport TEST (output data_in, rst_n, wr_en, rd_en, input clk, data_out, wr_ack, overflow, underflow, full, empty, almostfull, almostempty);
19
20      modport MONITOR (input clk, data_in, rst_n, wr_en, rd_en, data_out, wr_ack, overflow, underflow, full, empty, almostfull, almostempty);
21
22  endinterface : FIFO_interface
23
```

Monitor code snippet:

```systemverilog
1   import shared_pkg::*;
2   import FIFO_transaction_pkg::*;
3   import FIFO_coverage_pkg::*;
4   import FIFO_scoreboard::*;
5
6   module FIFO_monitor(FIFO_interface.MONITOR FIFO_IF);
7
8       FIFO_transaction trans_obj;
9       FIFO_coverage cvg_obj;
10      FIFO_scoreboard score_obj;
11
12      initial begin
13          trans_obj=new(); cvg_obj=new(); score_obj=new();
14
15          forever begin
16              @(negedge FIFO_IF.clk);
17              // Sampling IO signals
18              trans_obj.rd_en=FIFO_IF.rd_en;
19              trans_obj.wr_en=FIFO_IF.wr_en;
20              trans_obj.rst_n=FIFO_IF.rst_n;
21              trans_obj.data_in=FIFO_IF.data_in;
22              trans_obj.data_out=FIFO_IF.data_out;
23              trans_obj.wr_ack=FIFO_IF.wr_ack;
24              trans_obj.overflow=FIFO_IF.overflow;
25              trans_obj.underflow=FIFO_IF.underflow;
26              trans_obj.full=FIFO_IF.full;
27              trans_obj.almostfull=FIFO_IF.almostfull;
28              trans_obj.empty=FIFO_IF.empty;
29              trans_obj.almostempty=FIFO_IF.almostempty;
30
31              fork
32              // First process
33              begin
34                  cvg_obj.sample_data(trans_obj);
35              end
36              // Second process
37              begin
38                  score_obj.check_data(trans_obj);
39              end
40              join
41
42          if (test_finished) begin
43              $display("The test has finished, Correct counts: %d, Error counts: %d",correct_counter, error_counter);
44              $display("The FIFO contents at the end is: %p", score_obj.fifo_queue);
45              $stop;
46          end
47          end
48
49      end
50  endmodule
```

Testbench code snippet:

```systemverilog
1   import shared_pkg::*;
2   import FIFO_transaction_pkg::*;
3   import FIFO_coverage_pkg::*;
4   import FIFO_scoreboard::*;
5
6
7   module FIFO_tb (FIFO_interface.TEST FIFO_IF);
8
9       FIFO_transaction trans_obj_tb = new();
10      initial begin
11          // Initialize signals
12          FIFO_IF.rst_n=0; FIFO_IF.rd_en=0; FIFO_IF.wr_en=0; FIFO_IF.data_in=0;
13          @(negedge FIFO_IF.clk);
14          FIFO_IF.rst_n=1;
15          @(negedge FIFO_IF.clk);
16
17          // Testing
18          repeat(10_000) begin
19              assert(trans_obj_tb.randomize());
20              FIFO_IF.rst_n=trans_obj_tb.rst_n;
21              FIFO_IF.rd_en=trans_obj_tb.rd_en;
22              FIFO_IF.wr_en=trans_obj_tb.wr_en;
23              FIFO_IF.data_in=trans_obj_tb.data_in;
24              @(negedge FIFO_IF.clk);
25          end
26
27          test_finished =1;// Raise flag to stop testing
28      end
29
30  endmodule : FIFO_tb
```

Top module code snippet:

```systemverilog
1   module FIFO_top ();
2
3       bit clk;
4       // Clock generation
5       initial begin
6           clk=0;
7           forever
8               #1 clk = ~clk;// Clock period = 2ns
9       end
10
11      // Instantiations
12      FIFO_interface FIFO_IF (clk);
13
14      FIFO FIFO_DUT (FIFO_IF);
15
16      FIFO_tb FIFO_TB (FIFO_IF);
17
18      FIFO_monitor FIFO_MON (FIFO_IF);
19  endmodule
20
```

Transaction package code snippet:

```systemverilog
1   package FIFO_transaction_pkg;
2       class FIFO_transaction ;
3           // Parameters
4           parameter FIFO_WIDTH = 16;
5           parameter FIFO_DEPTH = 8;
6           localparam max_fifo_addr = $clog2(FIFO_DEPTH);
7
8           // Signals
9           rand Logic [FIFO_WIDTH-1:0] data_in;
10          rand Logic rst_n, wr_en, rd_en;
11          Logic [FIFO_WIDTH-1:0] data_out;
12          Logic wr_ack, overflow, underflow;
13          Logic full, empty, almostfull, almostempty;
14          /*************Added signals as integers**************/
15          integer RD_EN_ON_DIST,WR_EN_ON_DIST;
16
17          // Constructor, let the default of RD_EN_ON_DIST be 30 and WR_EN_ON_DIST be 70
18          function new(integer RD_EN_ON_DIST = 30, integer WR_EN_ON_DIST =70);
19              this.RD_EN_ON_DIST=RD_EN_ON_DIST;
20              this.WR_EN_ON_DIST=WR_EN_ON_DIST;
21          endfunction : new
22
23          // Constraints
24          constraint rst_constraint {
25              rst_n dist {0:/5, 1:/95};//Assert reset less often
26          }
27          constraint write_constraint {
28              // Write enable to be high with distribution of the value WR_EN_ON_DIST and to be low with 100-WR_EN_ON_DIST
29              wr_en dist {1:/WR_EN_ON_DIST, 0:/(100-WR_EN_ON_DIST)};
30          }
31          constraint read_constraint {
32              // Read enable to be high with distribution of the value RD_EN_ON_DIST and to be low with 100-RD_EN_ON_DIST
33              rd_en dist {1:/RD_EN_ON_DIST, 0:/(100-RD_EN_ON_DIST)};
34          }
35      endclass : FIFO_transaction
36  endpackage : FIFO_transaction_pkg
```

Coverage package code snippet:

```systemverilog
1   package FIFO_coverage_pkg;
2   import FIFO_transaction_pkg::*;
3       class FIFO_coverage;
4
5           FIFO_transaction F_cvg_txn = new();
6
7           covergroup cg;
8
9               // Coverpoints
10              wr_en_cp: coverpoint F_cvg_txn.wr_en;
11              rd_en_cp: coverpoint F_cvg_txn.rd_en;
12              wr_ack_cp: coverpoint F_cvg_txn.wr_ack;
13              overflow_cp: coverpoint F_cvg_txn.overflow;
14              underflow_cp: coverpoint F_cvg_txn.underflow;
15              full_cp: coverpoint F_cvg_txn.full;
16              almostfull_cp: coverpoint F_cvg_txn.almostfull;
17              empty_cp: coverpoint F_cvg_txn.empty;
18              almostempty_cp: coverpoint F_cvg_txn.almostempty;
19
20              // Crosses
21              WRITE_READ_WR_ACK_CROSS: cross wr_en_cp, rd_en_cp, wr_ack_cp{
22                  ignore_bins WRITE0_READ1_WR_ACK1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{1} && binsof(wr_ack_cp)intersect{1};
23                  ignore_bins WRITE0_READ0_WR_ACK1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{0} && binsof(wr_ack_cp)intersect{1};
24              }
25              WRITE_READ_OVERFLOW_CROSS: cross wr_en_cp, rd_en_cp, overflow_cp{
26                  ignore_bins WRITE0_READ1_OVERFLOW1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{1} && binsof(overflow_cp)intersect{1};
27                  ignore_bins WRITE0_READ0_OVERFLOW1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{0} && binsof(overflow_cp)intersect{1};
28              }
29              WRITE_READ_UNDERFLOW_CROSS: cross wr_en_cp, rd_en_cp, underflow_cp {
30                  ignore_bins WRITE1_READ0_UNDERFLOW1 = binsof(wr_en_cp)intersect{1} && binsof(rd_en_cp)intersect{0} && binsof(underflow_cp)intersect{1};
31                  ignore_bins WRITE0_READ0_UNDERFLOW1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{0} && binsof(underflow_cp)intersect{1};
32              }
33              WRITE_READ_FULL_CROSS: cross wr_en_cp, rd_en_cp, full_cp {
34                  ignore_bins WRITE1_READ1_FULL1 = binsof(wr_en_cp)intersect{1} && binsof(rd_en_cp)intersect{1} && binsof(full_cp)intersect{1};
35                  ignore_bins WRITE0_READ1_FULL1 = binsof(wr_en_cp)intersect{0} && binsof(rd_en_cp)intersect{1} && binsof(full_cp)intersect{1};
36              }
37              WRITE_READ_EMPTY_CROSS: cross wr_en_cp, rd_en_cp, empty_cp;
38              WRITE_READ_ALMOST_FULL_CROSS: cross wr_en_cp, rd_en_cp, almostfull_cp;
39              WRITE_READ_ALMOST_EMPTY_CROSS: cross wr_en_cp, rd_en_cp, almostempty_cp;
40
41          endgroup : cg
42
43          function void sample_data(input FIFO_transaction F_txn);
44          F_cvg_txn = F_txn;
45          cg.sample();
46          endfunction : sample_data
47
48          function new();
49              cg=new();
50          endfunction : new
51
52      endclass : FIFO_coverage
53  endpackage : FIFO_coverage_pkg
```

Shared package code snippet:

```
1    package shared_pkg;
2
3        logic test_finished;
4
5        // Counters
6        integer error_counter=0;
7        integer correct_counter=0;
8
9    endpackage : shared_pkg
10
```

Scoreboard package code snippets:

```
1    package FIFO_scoreboard;
2        import FIFO_transaction_pkg::*;
3        import shared_pkg::*;
4
5        FIFO_transaction FIFO_transaction_object = new();
6
7        class FIFO_scoreboard;
8            // Reference FIFO
9            bit [FIFO_transaction_object.FIFO_WIDTH-1:0] fifo_queue [$];
10           int fifo_count = 0;
11
12           // Reference Signals
13           logic [FIFO_transaction_object.FIFO_WIDTH-1:0] data_out_ref;
14
15           // check_data Function
16           function void check_data(input FIFO_transaction obj_one);
17
18               reference_model(obj_one);
19
20               // Compare the data_out with the reference
21               if (obj_one.data_out !== data_out_ref) begin
22                   $display("Error!!, At time %t, data_out %d doesn't equal data_out_ref %d !!", $time, obj_one.data_out, data_out_ref);
23                   error_counter++;
24               end
25               else begin
26                   $display("Success, At time %t, data_out= %d equals data_out_ref= %d", $time, obj_one.data_out, data_out_ref);
27                   correct_counter++;
28               end
29           endfunction : check_data
30
31           // reference_model Function
32           function void reference_model(input FIFO_transaction F_txn);
33               // Reset logic
34               if (!F_txn.rst_n) begin
35                   fifo_queue <= {};
36                   fifo_count <= 0;
37               end
38               else begin
39                   // Write operation if not full
40                   if (F_txn.wr_en && fifo_count < FIFO_transaction_object.FIFO_DEPTH) begin
41                       fifo_queue.push_back(F_txn.data_in);
42                       fifo_count <= fifo_queue.size();
43                   end
44
45                   // Read operation if not empty
46                   if (F_txn.rd_en && fifo_count != 0) begin
47                       data_out_ref <= fifo_queue.pop_front();
48                       fifo_count <= fifo_queue.size();
49                   end
50
51               end
52           endfunction : reference_model
53
54       endclass : FIFO_scoreboard
55   endpackage : FIFO_scoreboard
```

DO File:

```
1    vlib work
2    vlog -f FIFO.list  +cover -covercells +define+SIM
3    vsim -voptargs=+acc work.FIFO_top -cover -sv_seed random -l sim.FIFO_log
4    add wave *
5    add wave -position insertpoint  \
6    sim:/FIFO_top/FIFO_IF/almostempty \
7    sim:/FIFO_top/FIFO_IF/almostfull \
8    sim:/FIFO_top/FIFO_IF/clk \
9    sim:/FIFO_top/FIFO_IF/data_in \
10   sim:/FIFO_top/FIFO_IF/data_out \
11   sim:/FIFO_top/FIFO_IF/empty \
12   sim:/FIFO_top/FIFO_IF/FIFO_DEPTH \
13   sim:/FIFO_top/FIFO_IF/FIFO_WIDTH \
14   sim:/FIFO_top/FIFO_IF/full \
15   sim:/FIFO_top/FIFO_IF/max_fifo_addr \
16   sim:/FIFO_top/FIFO_IF/overflow \
17   sim:/FIFO_top/FIFO_IF/rd_en \
18   sim:/FIFO_top/FIFO_IF/rst_n \
19   sim:/FIFO_top/FIFO_IF/underflow \
20   sim:/FIFO_top/FIFO_IF/wr_ack \
21   sim:/FIFO_top/FIFO_IF/wr_en
22   coverage save FIFO.ucdb -onexit
23   run -all
24   quit -sim
25   vcover report FIFO.ucdb -details -annotate -all -output coverage_FIFO_rpt.txt
26
```

Files list:

```
1    shared_pkg.sv
2    FIFO_transaction_pkg.sv
3    FIFO_coverage_pkg.sv
4    FIFO_scoreboard.sv
5    FIFO_top.sv
6    FIFO_interface.sv
7    FIFO.sv
8    FIFO_tb.sv
9    FIFO_monitor.sv
```

Code coverage:

Toggle coverage:

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


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

Toggle Coverage for instance /FIFO_top/FIFO_IF --

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

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

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


FIRST IN
FIRST OUT
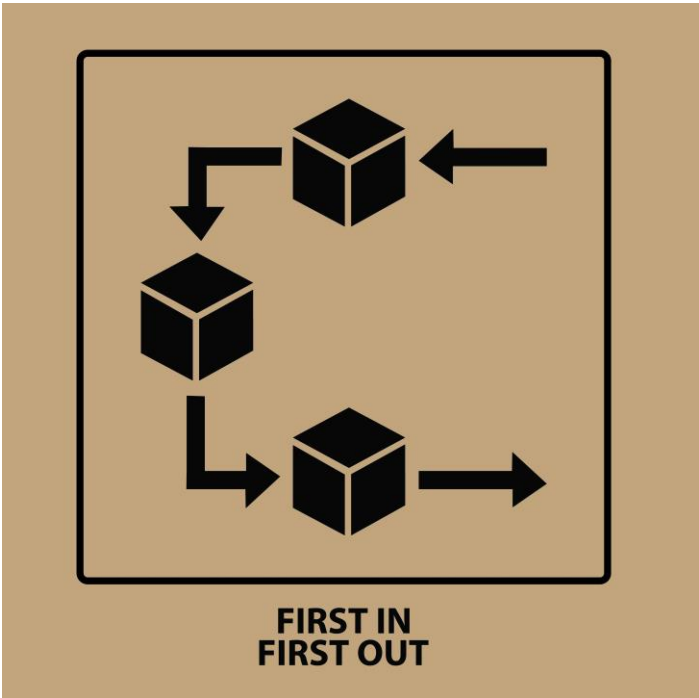
Branch coverage:

```
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                       35        35        0    100.00%

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

Branch Coverage for instance /FIFO_top/FIFO_DUT

    Line       Item                   Count    Source
    ----       ----                   -----    ------
  File FIFO.sv
---------------------------------IF Branch-----------------------------------
    15                              10455    Count coming in to IF
    15            1                   926        if (!FIFO_IF.rst_n) begin
    21            1                  4993        else if (FIFO_IF.wr_en && count < FIFO_IF.FIFO_DEPTH) begin
    26            1                  4536        else begin
 Branch totals: 3 hits of 3 branches = 100.00%


---------------------------------IF Branch-----------------------------------
    28                               4536    Count coming in to IF
    28            1                  1709            if (FIFO_IF.full & FIFO_IF.wr_en)
    30            1                  2827            else
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    36                              10455    Count coming in to IF
    36            1                   926        if (!FIFO_IF.rst_n) begin
    41            1                  2653        else if (FIFO_IF.rd_en && count != 0) begin
    46            1                  6876        else begin
 Branch totals: 3 hits of 3 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    47                               6876    Count coming in to IF
    47            1                   207            if (FIFO_IF.empty & FIFO_IF.rd_en)
    49            1                  6669            else
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    55                               9166    Count coming in to IF
    55            1                   921        if (!FIFO_IF.rst_n) begin
    58            1                  8245        else begin
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    59                               8245    Count coming in to IF
    59            1                  3469            if      ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b10) && !FIFO_IF.full)
    61            1                   781            else if ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b01) && !FIFO_IF.empty)
    64            1                   144            else if ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b11) && FIFO_IF.empty)
    67            1                   492            else if ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b11) && FIFO_IF.full)
                                    3359        All False Count
 Branch totals: 5 hits of 5 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    72                               5318    Count coming in to IF
    72            1                   848        assign FIFO_IF.full = (count == FIFO_IF.FIFO_DEPTH)? 1 : 0;
    72            2                  4470        assign FIFO_IF.full = (count == FIFO_IF.FIFO_DEPTH)? 1 : 0;
 Branch totals: 2 hits of 2 branches = 100.00%
```

```
---------------------------------IF Branch-----------------------------------
    72                               5318    Count coming in to IF
    72            1                   848    assign FIFO_IF.full = (count == FIFO_IF.FIFO_DEPTH)? 1 : 0;
    72            2                  4470    assign FIFO_IF.full = (count == FIFO_IF.FIFO_DEPTH)? 1 : 0;
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    73                               5318    Count coming in to IF
    73            1                   509    assign FIFO_IF.empty = (count == 0)? 1 : 0;
    73            2                  4809    assign FIFO_IF.empty = (count == 0)? 1 : 0;
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    74                               5318    Count coming in to IF
    74            1                  1072    assign FIFO_IF.almostfull = (count == FIFO_IF.FIFO_DEPTH-1)? 1 : 0; // Bug detected: FIFO_IF.FIFO_DEPTH-2 --> FIFO_IF.FIFO_DEPTH-1
    74            2                  4246    assign FIFO_IF.almostfull = (count == FIFO_IF.FIFO_DEPTH-1)? 1 : 0; // Bug detected: FIFO_IF.FIFO_DEPTH-2 --> FIFO_IF.FIFO_DEPTH-1
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    75                               5318    Count coming in to IF
    75            1                   577    assign FIFO_IF.almostempty = (count == 1)? 1 : 0;
    75            2                  4741    assign FIFO_IF.almostempty = (count == 1)? 1 : 0;
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    81                               9657    Count coming in to IF
    81            1                   900            if(!FIFO_IF.rst_n)
                                     8757    All False Count
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    87                               6224    Count coming in to IF
    87            1                   848            if((FIFO_IF.rst_n)&&(count == FIFO_IF.FIFO_DEPTH))
                                     5376    All False Count
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    93                               6224    Count coming in to IF
    93            1                   530            if((FIFO_IF.rst_n)&&(count == 0))
                                     5694    All False Count
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    99                               6224    Count coming in to IF
    99            1                  1072            if((FIFO_IF.rst_n)&&(count == FIFO_IF.FIFO_DEPTH-1))
                                     5152    All False Count
 Branch totals: 2 hits of 2 branches = 100.00%

---------------------------------IF Branch-----------------------------------
    105                              6224    Count coming in to IF
    105           1                   577            if((FIFO_IF.rst_n)&&(count == 1))
                                     5647    All False Count
 Branch totals: 2 hits of 2 branches = 100.00%
```

Statement coverage:

```
Statement Coverage:
    Enabled Coverage          Bins    Hits    Misses  Coverage
    ----------------          ----    ----    ------  --------
    Statements                 32      32        0    100.00%

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

Statement Coverage for instance /FIFO_top/FIFO_DUT --

    Line      Item              Count    Source
    ----      ----              -----    ------
  File FIFO.sv
    7                                    module FIFO (FIFO_interface.DUT FIFO_IF);
    8
    9                                    reg [FIFO_IF.max_fifo_addr-1:0] wr_ptr, rd_ptr;
    10                                   reg [FIFO_IF.max_fifo_addr:0] count;
    11
    12                                   reg [FIFO_IF.FIFO_WIDTH-1:0] mem [FIFO_IF.FIFO_DEPTH-1:0];
    13
    14         1               10484     always @(posedge FIFO_IF.clk or negedge FIFO_IF.rst_n) begin
    15                                       if (!FIFO_IF.rst_n) begin
    16         1                 991             wr_ptr <= 0;
    17                                           // Bug detected: Reset signals FIFO_IF.overflow & FIFO_IF.wr_ack
    18         1                 991             FIFO_IF.overflow <= 0;
    19         1                 991             FIFO_IF.wr_ack <= 0;
    20                                       end
    21                                       else if (FIFO_IF.wr_en && count < FIFO_IF.FIFO_DEPTH) begin
    22         1                4931             mem[wr_ptr] <= FIFO_IF.data_in;
    23         1                4931             FIFO_IF.wr_ack<=1;
    24         1                4931             wr_ptr <= wr_ptr + 1;
    25                                       end
    26                                       else begin
    27         1                4562             FIFO_IF.wr_ack <= 0;
    28                                           if (FIFO_IF.full & FIFO_IF.wr_en)
    29         1                1694                 FIFO_IF.overflow <= 1;
    30                                           else
    31         1                2868                 FIFO_IF.overflow <= 0;
    32                                       end
    33                                   end
    34
    35         1               10484     always @(posedge FIFO_IF.clk or negedge FIFO_IF.rst_n) begin
    36                                       if (!FIFO_IF.rst_n) begin
    37         1                 991             rd_ptr <= 0;
    38                                           // Bug detected: Reset signals FIFO_IF.underflow
    39         1                 991             FIFO_IF.underflow <= 0;
    40                                       end
    41                                       else if (FIFO_IF.rd_en && count != 0) begin
    42         1                2578             FIFO_IF.data_out <= mem[rd_ptr];
    43         1                2578             rd_ptr <= rd_ptr + 1;
    44                                       end
    45                                       // Handled FIFO_IF.underflow behaviour when turned from combinational to sequential
    46                                       else begin
    47                                           if (FIFO_IF.empty & FIFO_IF.rd_en)
    48         1                 208                 FIFO_IF.underflow <= 1;
    49                                           else
    50         1                6707                 FIFO_IF.underflow <= 0;
    51                                       end
    52                                   end
    53
    54         1                9257     always @(posedge FIFO_IF.clk or negedge FIFO_IF.rst_n) begin
    55                                       if (!FIFO_IF.rst_n) begin
    56         1                 982             count <= 0;
    57                                       end
    58                                       else begin
    59                                           if     ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b10) && !FIFO_IF.full)
    60         1                3486                 count <= count + 1;
    61                                           else if ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b01) && !FIFO_IF.empty)
    62         1                 766                 count <= count - 1;
    63                                           // Bug detected: Unhandled case,  If a read and write enables were high and the FIFO was FIFO_IF.empty, only writing will take place.
    64                                           else if ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b11) && FIFO_IF.empty)
    65         1                 143                 count <= count + 1;
    66                                           // Bug detected: Unhandled cases,  If a read and write enables were high and the FIFO was FIFO_IF.full, only reading will take place.
    67                                           else if ( ({FIFO_IF.wr_en, FIFO_IF.rd_en} == 2'b11) && FIFO_IF.full)
    68         1                 510                 count <= count - 1;
    69                                       end
    70                                   end
    71
    72         1                5368     assign FIFO_IF.full = (count == FIFO_IF.FIFO_DEPTH)? 1 : 0;
    73         1                5368     assign FIFO_IF.empty = (count == 0)? 1 : 0;
    74         1                5368     assign FIFO_IF.almostfull = (count == FIFO_IF.FIFO_DEPTH-1)? 1 : 0; // Bug detected: FIFO_IF.FIFO_DEPTH-2 --> FIFO_IF.FIFO_DEPTH-1
    75         1                5368     assign FIFO_IF.almostempty = (count == 1)? 1 : 0;
    76
    77                                   // Guarded assertions
    78                                   `ifdef SIM
    79                                       // Properties, Assertions & Covers
    80         1                9729     always_comb begin
    81                                       if(!FIFO_IF.rst_n)
    82                                           reset_1_assertion: assert final ((!FIFO_IF.wr_ack)&&(!FIFO_IF.overflow)&&(!FIFO_IF.underflow)&&(!wr_ptr)&&(!rd_ptr)&&(!count));
    83                                           reset_1_cover: cover final ((!FIFO_IF.wr_ack)&&(!FIFO_IF.overflow)&&(!FIFO_IF.underflow)&&(!wr_ptr)&&(!rd_ptr)&&(!count));
```

```
    77                                   // Guarded assertions
    78                                   `ifdef SIM
    79                                       // Properties, Assertions & Covers
    80         1                9729     always_comb begin
    81                                       if(!FIFO_IF.rst_n)
    82                                           reset_1_assertion: assert final ((!FIFO_IF.wr_ack)&&(!FIFO_IF.overflow)&&(!FIFO_IF.underflow)&&(!wr_ptr)&&(!rd_ptr)&&(!count));
    83                                           reset_1_cover: cover final ((!FIFO_IF.wr_ack)&&(!FIFO_IF.overflow)&&(!FIFO_IF.underflow)&&(!wr_ptr)&&(!rd_ptr)&&(!count));
    84                                   end
    85
    86         1                6331     always_comb begin
    87                                       if((FIFO_IF.rst_n)&&(count == FIFO_IF.FIFO_DEPTH))
    88                                           full_assertion: assert final (FIFO_IF.full);
    89                                           full_cover: cover (FIFO_IF.full);
    90                                   end
    91
    92         1                6331     always_comb begin
    93                                       if((FIFO_IF.rst_n)&&(count == 0))
    94                                           empty_assertion: assert final (FIFO_IF.empty);
    95                                           empty_cover: cover (FIFO_IF.empty);
    96                                   end
    97
    98         1                6331     always_comb begin
    99                                       if((FIFO_IF.rst_n)&&(count == FIFO_IF.FIFO_DEPTH-1))
    100                                          almostfull_assertion: assert final (FIFO_IF.almostfull);
    101                                          almostfull_cover: cover (FIFO_IF.almostfull);
    102                                  end
    103
    104        1                6331     always_comb begin
```

Condition coverage:

```
Condition Coverage:
    Enabled Coverage              Bins    Covered   Misses  Coverage
    ----------------              ----    -------   ------  --------
    Conditions                      32       32        2    100%

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

Condition Coverage for instance /FIFO_top/FIFO_DUT --

  File FIFO.sv
-----------------Focused Condition View-------------------
Line      21 Item    1  (FIFO_IF.wr_en && (count < FIFO_IF.FIFO_DEPTH))
Condition totals: 2 of 2 input terms covered = 100.00%

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

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

-----------------Focused Condition View-------------------
Line      28 Item    1  (FIFO_IF.full & FIFO_IF.wr_en)
Condition totals: 1 of 2 input terms covered = 50.00%

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

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

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

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

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

-----------------Focused Condition View-------------------
Line      47 Item    1  (FIFO_IF.empty & FIFO_IF.rd_en)
Condition totals: 1 of 2 input terms covered = 50.00%

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

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

```
----------------Focused Condition View--------------------
Line      59 Item    1  ((~FIFO_IF.rd_en && FIFO_IF.wr_en) && ~FIFO_IF.full)
Condition totals: 3 of 3 input terms covered = 100.00%

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

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

----------------Focused Condition View--------------------
Line      61 Item    1  ((FIFO_IF.rd_en && ~FIFO_IF.wr_en) && ~FIFO_IF.empty)
Condition totals: 3 of 3 input terms covered = 100.00%

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

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

----------------Focused Condition View--------------------
Line      64 Item    1  ((FIFO_IF.rd_en && FIFO_IF.wr_en) && FIFO_IF.empty)
Condition totals: 3 of 3 input terms covered = 100.00%

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

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

----------------Focused Condition View--------------------
Line      67 Item    1  ((FIFO_IF.rd_en && FIFO_IF.wr_en) && FIFO_IF.full)
Condition totals: 3 of 3 input terms covered = 100.00%

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

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

```
  Row   5:         1 FIFO_IF.full_0      (FIFO_IF.rd_en && FIFO_IF.wr_en)
  Row   6:         1 FIFO_IF.full_1      (FIFO_IF.rd_en && FIFO_IF.wr_en)

----------------Focused Condition View-------------------
Line      72 Item   1 (count == FIFO_IF.FIFO_DEPTH)
Condition totals: 1 of 1 input term covered = 100.00%

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

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

----------------Focused Condition View-------------------
Line      73 Item   1 (count == 0)
Condition totals: 1 of 1 input term covered = 100.00%

     Input Term   Covered  Reason for no coverage   Hint
    ----------  --------  ----------------------   --------------
    (count == 0)          Y

       Rows:        Hits  FEC Target                   Non-masking condition(s)
    ---------  ---------  --------------------         -------------------------
    Row   1:         1 (count == 0)_0                   -
    Row   2:         1 (count == 0)_1                   -

----------------Focused Condition View-------------------
Line      74 Item   1 (count == (FIFO_IF.FIFO_DEPTH - 1))
Condition totals: 1 of 1 input term covered = 100.00%

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

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

----------------Focused Condition View-------------------
Line      75 Item   1 (count == 1)
Condition totals: 1 of 1 input term covered = 100.00%

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

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

----------------Focused Condition View-------------------
Line      87 Item   1 (FIFO_IF.rst_n && (count == FIFO_IF.FIFO_DEPTH))
Condition totals: 2 of 2 input terms covered = 100.00%

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

       Rows:        Hits  FEC Target                   Non-masking condition(s)
    ---------  ---------  --------------------         -------------------------
    Row   1:         1 FIFO_IF.rst_n_0                  -
    Row   2:         1 FIFO_IF.rst_n_1                  (count == FIFO_IF.FIFO_DEPTH)
    Row   3:         1 (count == FIFO_IF.FIFO_DEPTH)_0  FIFO_IF.rst_n
    Row   4:         1 (count == FIFO_IF.FIFO_DEPTH)_1  FIFO_IF.rst_n

----------------Focused Condition View-------------------
Line      93 Item   1 (FIFO_IF.rst_n && (count == 0))
Condition totals: 2 of 2 input terms covered = 100.00%

     Input Term   Covered  Reason for no coverage   Hint
    ----------  --------  ----------------------   --------------
    FIFO_IF.rst_n        Y
    (count == 0)         Y

       Rows:        Hits  FEC Target           Non-masking condition(s)
    ---------  ---------  --------------------  -------------------------
    Row   1:         1 FIFO_IF.rst_n_0          -
    Row   2:         1 FIFO_IF.rst_n_1          (count == 0)
    Row   3:         1 (count == 0)_0           FIFO_IF.rst_n
    Row   4:         1 (count == 0)_1           FIFO_IF.rst_n

----------------Focused Condition View-------------------
Line      99 Item   1 (FIFO_IF.rst_n && (count == (FIFO_IF.FIFO_DEPTH - 1)))
Condition totals: 2 of 2 input terms covered = 100.00%

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

       Rows:        Hits  FEC Target                   Non-masking condition(s)
    ---------  ---------  --------------------         -------------------------
    Row   1:         1 FIFO_IF.rst_n_0                  -
    Row   2:         1 FIFO_IF.rst_n_1                  (count == (FIFO_IF.FIFO_DEPTH - 1))
    Row   3:         1 (count == (FIFO_IF.FIFO_DEPTH - 1))_0  FIFO_IF.rst_n
    Row   4:         1 (count == (FIFO_IF.FIFO_DEPTH - 1))_1  FIFO_IF.rst_n

----------------Focused Condition View-------------------
Line     105 Item   1 (FIFO_IF.rst_n && (count == 1))
Condition totals: 2 of 2 input terms covered = 100.00%

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

       Rows:        Hits  FEC Target           Non-masking condition(s)
    ---------  ---------  --------------------  -------------------------
    Row   1:         1 FIFO_IF.rst_n_0          -
    Row   2:         1 FIFO_IF.rst_n_1          (count == 1)
    Row   3:         1 (count == 1)_0           FIFO_IF.rst_n
    Row   4:         1 (count == 1)_1           FIFO_IF.rst_n
```

Assertions coverage:

```
Assertion Coverage:
    Assertions                        14        14         0   100.00%
--------------------------------------------------------------------
Name                   File(Line)              Failure      Pass
                                               Count        Count
--------------------------------------------------------------------
/FIFO_top/FIFO_DUT/reset_1_assertion
                       FIFO.sv(82)                 0            1
/FIFO_top/FIFO_DUT/full_assertion
                       FIFO.sv(88)                 0            1
/FIFO_top/FIFO_DUT/empty_assertion
                       FIFO.sv(94)                 0            1
/FIFO_top/FIFO_DUT/almostfull_assertion
                       FIFO.sv(100)                0            1
/FIFO_top/FIFO_DUT/almostempty_assertion
                       FIFO.sv(106)                0            1
/FIFO_top/FIFO_DUT/reset_2_assertion
                       FIFO.sv(156)                0            1
/FIFO_top/FIFO_DUT/write_assertion
                       FIFO.sv(159)                0            1
/FIFO_top/FIFO_DUT/overflow_assertion
                       FIFO.sv(162)                0            1
/FIFO_top/FIFO_DUT/read_assertion
                       FIFO.sv(165)                0            1
/FIFO_top/FIFO_DUT/underflow_assertion
                       FIFO.sv(168)                0            1
/FIFO_top/FIFO_DUT/write_not_full_assertion
                       FIFO.sv(171)                0            1
/FIFO_top/FIFO_DUT/read_not_empty_assertion
                       FIFO.sv(174)                0            1
/FIFO_top/FIFO_DUT/read_write_empty_assertion
                       FIFO.sv(177)                0            1
/FIFO_top/FIFO_DUT/read_write_full_assertion
                       FIFO.sv(180)                0            1
Branch Coverage:
    Enabled Coverage            Bins      Hits    Misses  Coverage
    ----------------            ----      ----    ------  --------
    Branches                      35        35         0   100.00%
```

Directive coverage:

```
Directive Coverage:
    Directives                        14        14         0   100.00%

DIRECTIVE COVERAGE:
-----------------------------------------------------------------------------------
Name                          Design Design  Lang File(Line)      Hits Status
                              Unit   UnitType
-----------------------------------------------------------------------------------
/FIFO_top/FIFO_DUT/reset_1_cover       FIFO   Verilog  SVA  FIFO.sv(83)     974 Covered  |
/FIFO_top/FIFO_DUT/full_cover          FIFO   Verilog  SVA  FIFO.sv(89)     943 Covered
/FIFO_top/FIFO_DUT/empty_cover         FIFO   Verilog  SVA  FIFO.sv(95)    1047 Covered
/FIFO_top/FIFO_DUT/almostfull_cover    FIFO   Verilog  SVA  FIFO.sv(101)   1126 Covered
/FIFO_top/FIFO_DUT/almostempty_cover   FIFO   Verilog  SVA  FIFO.sv(107)    691 Covered
/FIFO_top/FIFO_DUT/reset_2_cover       FIFO   Verilog  SVA  FIFO.sv(157)    509 Covered
/FIFO_top/FIFO_DUT/write_cover         FIFO   Verilog  SVA  FIFO.sv(160)   4695 Covered
/FIFO_top/FIFO_DUT/overflow_cover      FIFO   Verilog  SVA  FIFO.sv(163)   1619 Covered
/FIFO_top/FIFO_DUT/read_cover          FIFO   Verilog  SVA  FIFO.sv(166)   2446 Covered
/FIFO_top/FIFO_DUT/underflow_cover     FIFO   Verilog  SVA  FIFO.sv(169)    198 Covered
/FIFO_top/FIFO_DUT/write_not_full_cover FIFO  Verilog  SVA  FIFO.sv(172)   3316 Covered
/FIFO_top/FIFO_DUT/read_not_empty_cover FIFO  Verilog  SVA  FIFO.sv(175)    717 Covered
/FIFO_top/FIFO_DUT/read_write_empty_cover
                              FIFO   Verilog  SVA  FIFO.sv(178)    137 Covered
/FIFO_top/FIFO_DUT/read_write_full_cover FIFO Verilog  SVA  FIFO.sv(181)    487 Covered
Statement Coverage:
    Enabled Coverage            Bins      Hits    Misses  Coverage
    ----------------            ----      ----    ------  --------
    Statements                    32        32         0   100.00%
```

Functional coverage:

```
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/cg     100.00%    100      -    Covered
    covered/total bins:                            66     66      -
    missing/total bins:                             0     66      -
    % Hit:                                    100.00%    100      -
    Coverpoint wr_en_cp                       100.00%    100      -    Covered
        covered/total bins:                         2      2      -
        missing/total bins:                         0      2      -
        % Hit:                                100.00%    100      -
        bin auto[0]                              3011      1      -    Covered
        bin auto[1]                              6991      1      -    Covered
    Coverpoint rd_en_cp                       100.00%    100      -    Covered
        covered/total bins:                         2      2      -
        missing/total bins:                         0      2      -
        % Hit:                                100.00%    100      -
        bin auto[0]                              7079      1      -    Covered
        bin auto[1]                              2923      1      -    Covered
    Coverpoint wr_ack_cp                      100.00%    100      -    Covered
        covered/total bins:                         2      2      -
        missing/total bins:                         0      2      -
        % Hit:                                100.00%    100      -
        bin auto[0]                              5071      1      -    Covered
        bin auto[1]                              4931      1      -    Covered
    Coverpoint overflow_cp                    100.00%    100      -    Covered
        covered/total bins:                         2      2      -
        missing/total bins:                         0      2      -
        % Hit:                                100.00%    100      -
        bin auto[0]                              7878      1      -    Covered
        bin auto[1]                              2124      1      -    Covered
    Coverpoint underflow_cp                   100.00%    100      -    Covered
        covered/total bins:                         2      2      -
        missing/total bins:                         0      2      -
        % Hit:                                100.00%    100      -
        bin auto[0]                              9745      1      -    Covered
        bin auto[1]                               257      1      -    Covered
    Coverpoint full_cp                        100.00%    100      -    Covered
        covered/total bins:                         2      2      -
        missing/total bins:                         0      2      -
        % Hit:                                100.00%    100      -
        bin auto[0]                              7468      1      -    Covered
        bin auto[1]                              2534      1      -    Covered
    Coverpoint almostfull_cp                  100.00%    100      -    Covered
        covered/total bins:                         2      2      -
        missing/total bins:                         0      2      -
        % Hit:                                100.00%    100      -
        bin auto[0]                              8260      1      -    Covered
        bin auto[1]                              1742      1      -    Covered
    Coverpoint empty_cp                       100.00%    100      -    Covered
        covered/total bins:                         2      2      -
        missing/total bins:                         0      2      -
        % Hit:                                100.00%    100      -
        bin auto[0]                              9206      1      -    Covered
        bin auto[1]                               796      1      -    Covered
    Coverpoint almostempty_cp                 100.00%    100      -    Covered
        covered/total bins:                         2      2      -
        missing/total bins:                         0      2      -
        % Hit:                                100.00%    100      -
        bin auto[0]                              8934      1      -    Covered
        bin auto[1]                              1068      1      -    Covered
    Cross WRITE_READ_WR_ACK_CROSS             100.00%    100      -    Covered
        covered/total bins:                         6      6      -
        missing/total bins:                         0      6      -
        % Hit:                                100.00%    100      -
        Auto, Default and User Defined Bins:
            bin <auto[1],auto[1],auto[1]>        1031      1      -    Covered
            bin <auto[1],auto[0],auto[1]>        2411      1      -    Covered
            bin <auto[1],auto[1],auto[0]>        1030      1      -    Covered
            bin <auto[1],auto[0],auto[0]>        2519      1      -    Covered
```

```
          bin <auto[1],auto[0],auto[0]>            2519        1       -    Covered    |
          bin <auto[0],auto[1],auto[0]>             443        1       -    Covered
          bin <auto[0],auto[0],auto[0]>            1079        1       -    Covered
       Illegal and Ignore Bins:
          ignore_bin WRITE0_READ0_WR_ACK1          1070                -    Occurred
          ignore_bin WRITE0_READ1_WR_ACK1           419                -    Occurred
  Cross WRITE_READ_OVERFLOW_CROSS               100.00%      100       -    Covered
       covered/total bins:                           6        6       -
       missing/total bins:                           0        6       -
       % Hit:                                   100.00%      100       -
       Auto, Default and User Defined Bins:
          bin <auto[1],auto[1],auto[1]>             439        1       -    Covered
          bin <auto[1],auto[0],auto[1]>            1039        1       -    Covered
          bin <auto[1],auto[1],auto[0]>            1622        1       -    Covered
          bin <auto[1],auto[0],auto[0]>            3891        1       -    Covered
          bin <auto[0],auto[1],auto[0]>             687        1       -    Covered
          bin <auto[0],auto[0],auto[0]>            1678        1       -    Covered
       Illegal and Ignore Bins:
          ignore_bin WRITE0_READ0_OVERFLOW1         471                -    Occurred
          ignore_bin WRITE0_READ1_OVERFLOW1         175                -    Occurred
  Cross WRITE_READ_UNDERFLOW_CROSS              100.00%      100       -    Covered
       covered/total bins:                           6        6       -
       missing/total bins:                           0        6       -
       % Hit:                                   100.00%      100       -
       Auto, Default and User Defined Bins:
          bin <auto[1],auto[1],auto[1]>              57        1       -    Covered
          bin <auto[1],auto[1],auto[0]>            2004        1       -    Covered
          bin <auto[0],auto[1],auto[1]>              15        1       -    Covered
          bin <auto[0],auto[1],auto[0]>             847        1       -    Covered
          bin <auto[1],auto[0],auto[0]>            4801        1       -    Covered
          bin <auto[0],auto[0],auto[0]>            2093        1       -    Covered
       Illegal and Ignore Bins:
          ignore_bin WRITE0_READ0_UNDERFLOW1         56                -    Occurred
          ignore_bin WRITE1_READ0_UNDERFLOW1        129                -    Occurred
  Cross WRITE_READ_FULL_CROSS                   100.00%      100       -    Covered
       covered/total bins:                           6        6       -
       missing/total bins:                           0        6       -
       % Hit:                                   100.00%      100       -
       Auto, Default and User Defined Bins:
          bin <auto[1],auto[1],auto[0]>            1525        1       -    Covered
          bin <auto[0],auto[1],auto[0]>             662        1       -    Covered
          bin <auto[1],auto[0],auto[1]>            1243        1       -    Covered
          bin <auto[1],auto[0],auto[0]>            3687        1       -    Covered
          bin <auto[0],auto[0],auto[1]>             555        1       -    Covered
          bin <auto[0],auto[0],auto[0]>            1594        1       -    Covered
       Illegal and Ignore Bins:
          ignore_bin WRITE0_READ1_FULL1             200                -    Occurred
          ignore_bin WRITE1_READ1_FULL1             536                -    Occurred
  Cross WRITE_READ_EMPTY_CROSS                  100.00%      100       -    Covered
       covered/total bins:                           8        8       -
       missing/total bins:                           0        8       -
       % Hit:                                   100.00%      100       -
       Auto, Default and User Defined Bins:
          bin <auto[1],auto[1],auto[1]>             153        1       -    Covered
          bin <auto[0],auto[1],auto[1]>              68        1       -    Covered
          bin <auto[1],auto[0],auto[1]>             428        1       -    Covered
          bin <auto[0],auto[0],auto[1]>             147        1       -    Covered
          bin <auto[1],auto[1],auto[0]>            1908        1       -    Covered
          bin <auto[0],auto[1],auto[0]>             794        1       -    Covered
          bin <auto[1],auto[0],auto[0]>            4502        1       -    Covered
          bin <auto[0],auto[0],auto[0]>            2002        1       -    Covered
  Cross WRITE_READ_ALMOST_FULL_CROSS            100.00%      100       -    Covered
       covered/total bins:                           8        8       -
       missing/total bins:                           0        8       -
       % Hit:                                   100.00%      100       -
       Auto, Default and User Defined Bins:
          bin <auto[1],auto[1],auto[1]>             339        1       -    Covered
          bin <auto[0],auto[1],auto[1]>             157        1       -    Covered
          bin <auto[1],auto[0],auto[1]>             862        1       -    Covered
          bin <auto[0],auto[0],auto[1]>             384        1       -    Covered
          bin <auto[1],auto[1],auto[0]>            1722        1       -    Covered
          bin <auto[0],auto[1],auto[0]>             705        1       -    Covered
          bin <auto[1],auto[0],auto[0]>            4068        1       -    Covered
          bin <auto[0],auto[0],auto[0]>            1765        1       -    Covered
  Cross WRITE READ ALMOST EMPTY CROSS           100.00%      100       -    Covered
```

```
       Auto, Default and User Defined Bins:
          bin <auto[1],auto[1],auto[1]>             339        1       -    Covered
          bin <auto[0],auto[1],auto[1]>             157        1       -    Covered
          bin <auto[1],auto[0],auto[1]>             862        1       -    Covered
          bin <auto[0],auto[0],auto[1]>             384        1       -    Covered
          bin <auto[1],auto[1],auto[0]>            1722        1       -    Covered
          bin <auto[0],auto[1],auto[0]>             705        1       -    Covered
          bin <auto[1],auto[0],auto[0]>            4068        1       -    Covered
          bin <auto[0],auto[0],auto[0]>            1765        1       -    Covered
  Cross WRITE_READ_ALMOST_EMPTY_CROSS           100.00%      100       -    Covered
       covered/total bins:                           8        8       -
       missing/total bins:                           0        8       -
       % Hit:                                   100.00%      100       -
       Auto, Default and User Defined Bins:
          bin <auto[1],auto[1],auto[1]>             229        1       -    Covered
          bin <auto[0],auto[1],auto[1]>              83        1       -    Covered
          bin <auto[1],auto[0],auto[1]>             515        1       -    Covered
          bin <auto[0],auto[0],auto[1]>             241        1       -    Covered
          bin <auto[1],auto[1],auto[0]>            1832        1       -    Covered
          bin <auto[0],auto[1],auto[0]>             779        1       -    Covered
          bin <auto[1],auto[0],auto[0]>            4415        1       -    Covered
          bin <auto[0],auto[0],auto[0]>            1908        1       -    Covered
```

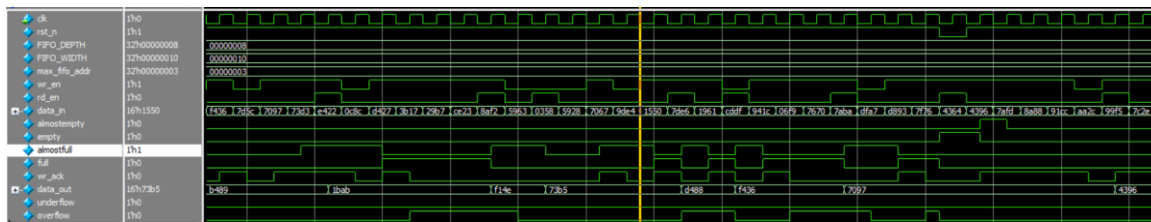# QuestaSim simulation waveform & transcript snippets:

Label1(When the rst_n is asserted, All flags & internal signals should equal 0):
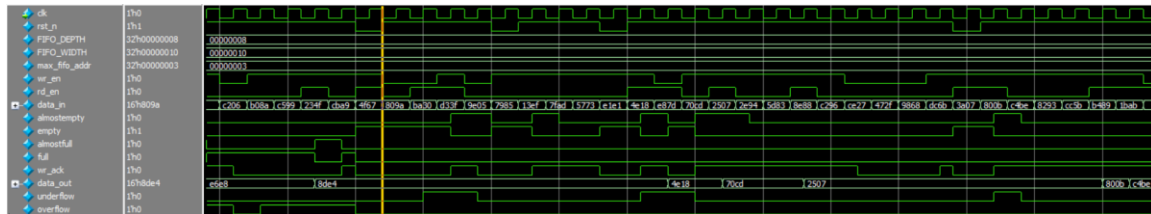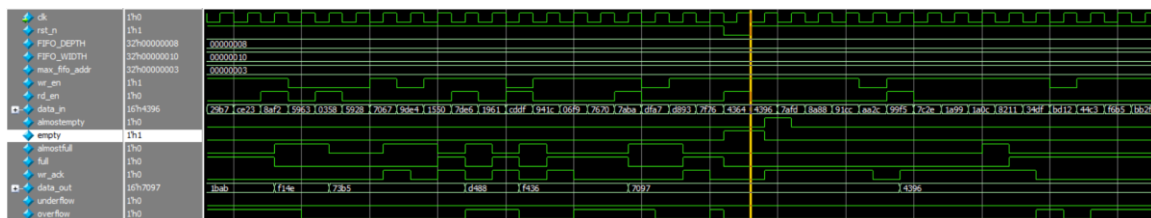

Label2(When rst_n is deactivated & the FIFO is full of elements (conut=depth), The full flag should be high):
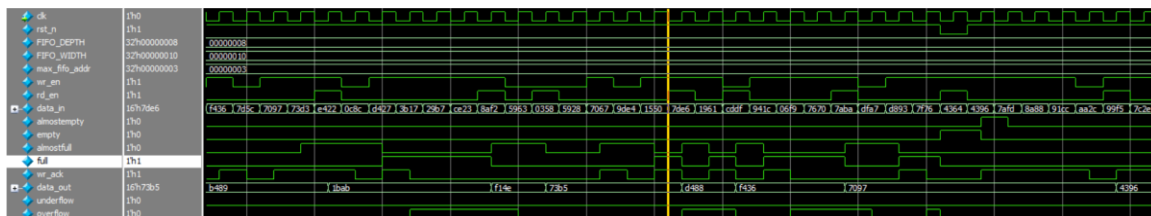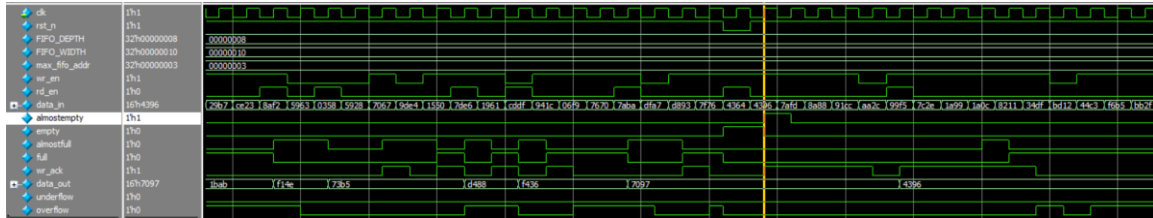

Label3(When rst_n is deactivated & the FIFO has one element left free(conut=depth-1), The almostfull flag should be high):
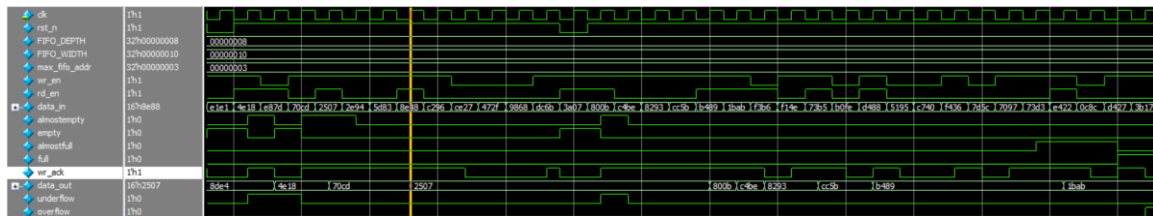

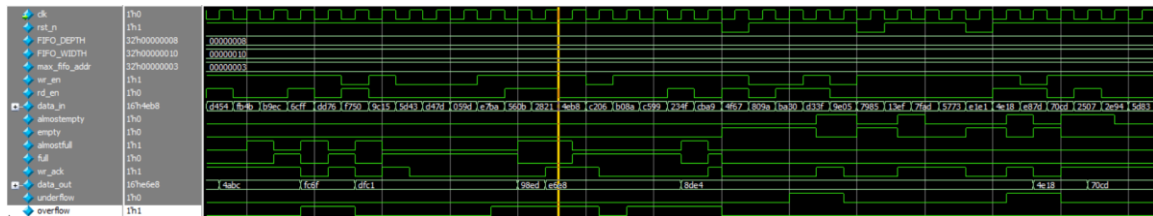Label4(When rst_n is deactivated & the FIFO has no element inside (conut=0), The empty flag should be high):

Label5(When rst_n is deactivated & the FIFO has only one element inside (conut=1), The empty flag should be high):

Label6(When rst_n is deactivated, You want to read & the FIFO is not full, The wr_ack flag should be high and wr_ptr should increment):

Label7(When rst_n is deactivated, You want to read & the FIFO is full, The overflow flag should be high):


Label8(When rst_n is deactivated, You want to read & the FIFO is not empty rd_ptr should increment and data_out should equal mem[rd_ptr]):

```
# Success, At time                    19972, data_out= 17168 equals data_out_ref= 17168
# Success, At time                    19974, data_out= 17168 equals data_out_ref= 17168
# Success, At time                    19976, data_out= 17168 equals data_out_ref= 17168
# Success, At time                    19978, data_out= 17168 equals data_out_ref= 17168
# Success, At time                    19980, data_out= 41860 equals data_out_ref= 41860
# Success, At time                    19982, data_out= 41860 equals data_out_ref= 41860
# Success, At time                    19984, data_out= 41860 equals data_out_ref= 41860
# Success, At time                    19986, data_out= 41860 equals data_out_ref= 41860
# Success, At time                    19988, data_out= 41860 equals data_out_ref= 41860
# Success, At time                    19990, data_out= 20926 equals data_out_ref= 20926
# Success, At time                    19992, data_out= 20926 equals data_out_ref= 20926
# Success, At time                    19994, data_out= 20926 equals data_out_ref= 20926
# Success, At time                    19996, data_out=  3376 equals data_out_ref=  3376
# Success, At time                    19998, data_out= 17121 equals data_out_ref= 17121
# Success, At time                    20000, data_out=  5133 equals data_out_ref=  5133
# Success, At time                    20002, data_out=  5133 equals data_out_ref=  5133
# Success, At time                    20004, data_out=  5133 equals data_out_ref=  5133
```

Label9(When rst_n is deactivated, You want to read & the FIFO is empty underflow flag should be high):
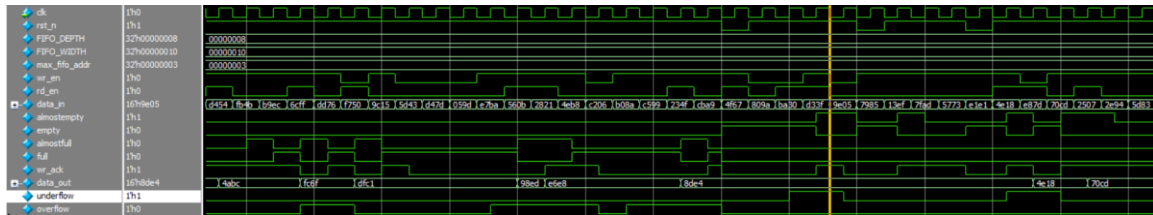
The FIFO is empty at the end of simulation
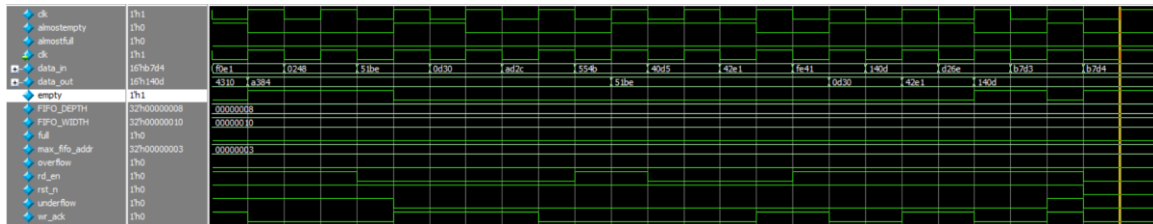
The test has finished, Correct counts: 10002, Error counts: 0
The FIFO contents at the end is: '{}

تم بحمد الله

سورة النجم

بِسْمِ اللَّهِ الرَّحْمَٰنِ الرَّحِيمِ

وَأَن لَّيْسَ لِلْإِنسَٰنِ إِلَّا مَا سَعَىٰ ﴿٣٩﴾

سورة التوبة

وَقُلِ ٱعْمَلُوا۟ فَسَيَرَى ٱللَّهُ عَمَلَكُمْ وَرَسُولُهُۥ وَٱلْمُؤْمِنُونَ ۖ وَسَتُرَدُّونَ إِلَىٰ عَٰلِمِ ٱلْغَيْبِ وَٱلشَّهَٰدَةِ فَيُنَبِّئُكُم بِمَا كُنتُمْ تَعْمَلُونَ ﴿١٠٥﴾