

# FIFO\_Project

**Submitted to: ENG. Kareem Waseem**

## Design code:

```
9
10 module FIFO(FIFO_interface.DUT FIFO_IC);
11
12 localparam max_fifo_addr = $clog2(FIFO_IC.FIFO_DEPTH);
13
14 reg [FIFO_IC.FIFO_WIDTH-1:0] mem [FIFO_IC.FIFO_DEPTH-1:0];
15 reg [max_fifo_addr-1:0] wr_ptr, rd_ptr;
16 reg [max_fifo_addr:0] count;
17
18 always @(posedge FIFO_IC.clk or negedge FIFO_IC.rst_n) begin
19     if (!FIFO_IC.rst_n) begin
20         wr_ptr <= 0;
21         FIFO_IC.overflow<=0; // was added
22     end
23     else if (FIFO_IC.wr_en && count < FIFO_IC.FIFO_DEPTH) begin
24         mem[wr_ptr] <= FIFO_IC.data_in;
25         FIFO_IC.wr_ack <= 1;
26         wr_ptr <= wr_ptr + 1;
27     end
28     else begin
29         FIFO_IC.wr_ack <= 0;
30         if (FIFO_IC.full & FIFO_IC.wr_en)
31             FIFO_IC.overflow <= 1;
32         else
33             FIFO_IC.overflow <= 0;
34     end
35 end
36
37 always @(posedge FIFO_IC.clk or negedge FIFO_IC.rst_n) begin
38     if (!FIFO_IC.rst_n) begin
39         rd_ptr <= 0;
40         FIFO_IC.underflow<=0; //was added
41     end
42     else if (FIFO_IC.rd_en && count != 0) begin
43         FIFO_IC.data_out <= mem[rd_ptr];
44         rd_ptr <= rd_ptr + 1;
45     end
46     else begin
47         if (FIFO_IC.empty && FIFO_IC.rd_en)begin
48             FIFO_IC.underflow <= 1;
49         end
50         else begin
51             FIFO_IC.underflow <= 0; // underflow was modified to be sequential instead of combinational
52         end
53     end
54 end
55
56 always @(posedge FIFO_IC.clk or negedge FIFO_IC.rst_n) begin
57     if (!FIFO_IC.rst_n) begin
58         count <= 0;
59     end
60     else begin
61         if (((FIFO_IC.wr_en, FIFO_IC.rd_en) == 2'b10) && !FIFO_IC.full)
62             count <= count + 1;
63         else if (((FIFO_IC.wr_en, FIFO_IC.rd_en) == 2'b01) && !FIFO_IC.empty)
64             count <= count - 1;
65         else if ((FIFO_IC.wr_en, FIFO_IC.rd_en) == 2'b11) begin // if condition was added to handle if the wr_en , rd_en are asserted at the same time
66             if (FIFO_IC.full) begin
67                 count <= count - 1;
68             end
69             else if (FIFO_IC.empty) begin
70                 count <= count + 1;
71             end
72         end
73     end
74 end
75
76 assign FIFO_IC.full = (count == FIFO_IC.FIFO_DEPTH)? 1 : 0;
77 assign FIFO_IC.empty = (count == 0)? 1 : 0;
78 assign FIFO_IC.almostfull = (count == FIFO_IC.FIFO_DEPTH-1)? 1 : 0; // corrected to be -1 instead of -2
79 assign FIFO_IC.almostempty = (count == 1)? 1 : 0;
80
```

```

84 property p_count_wr;
85     @(posedge FIFO_IC.clk) disable iff (!FIFO_IC.rst_n)
86     (FIFO_IC.wr_en && !FIFO_IC.full && !FIFO_IC.rd_en) | => (count == $past(count) + 1'b1);
87 endproperty
88
89 property p_count_rd;
90     @(posedge FIFO_IC.clk) disable iff (!FIFO_IC.rst_n)
91     (FIFO_IC.rd_en && !FIFO_IC.empty && !FIFO_IC.wr_en) | => (count == $past(count) - 1'b1);
92 endproperty
93
94 property p_w_ack;
95     @(posedge FIFO_IC.clk) disable iff (!FIFO_IC.rst_n)
96     (FIFO_IC.wr_en && !FIFO_IC.full) | => (FIFO_IC.wr_ack == 1);
97 endproperty
98
99 property p_full;
100     @(posedge FIFO_IC.clk)
101     (count == FIFO_IC.FIFO_DEPTH) | -> (FIFO_IC.full == 1);
102 endproperty
103
104 property p_empty;
105     @(posedge FIFO_IC.clk)
106     (count == 0) | -> (FIFO_IC.empty == 1);
107 endproperty
108
109 property p_almostfull;
110     @(posedge FIFO_IC.clk)
111     (count == FIFO_IC.FIFO_DEPTH - 1) | -> (FIFO_IC.almostfull == 1);
112 endproperty
113
114 property p_almostempty;
115     @(posedge FIFO_IC.clk)
116     (count == 1) | -> (FIFO_IC.almostempty == 1);
117 endproperty
118
119 property p_underflow;
120     @(posedge FIFO_IC.clk) disable iff (!FIFO_IC.rst_n)
121     (FIFO_IC.empty && FIFO_IC.rd_en) | => (FIFO_IC.underflow == 1);
122 endproperty
123
124 property p_overflow;
125     @(posedge FIFO_IC.clk) disable iff (!FIFO_IC.rst_n)
126     (FIFO_IC.full && FIFO_IC.wr_en) | => (FIFO_IC.overflow == 1);
127 endproperty
128
129 property p_count_check;
130     @(posedge FIFO_IC.clk)
131     (count <= FIFO_IC.FIFO_DEPTH);
132 endproperty
133
134 property p_wr_ptr_check;
135     @(posedge FIFO_IC.clk)
136     (wr_ptr < FIFO_IC.FIFO_DEPTH);
137 endproperty
138

```

```

139     property p_rd_ptr_check;
140         @(posedge FIFO_IC.clk)
141         (rd_ptr < FIFO_IC.FIFO_DEPTH);
142     endproperty
143
144
145     A_count_wr: assert property (p_count_wr);
146     C_count_wr: cover property (p_count_wr);
147
148     // Assertions and Cover for Read Count
149
150     A_count_rd: assert property (p_count_rd);
151     C_count_rd: cover property (p_count_rd);
152
153     // Assertions and Cover for Write Acknowledge
154
155     A_w_ack: assert property (p_w_ack);
156     C_w_ack: cover property (p_w_ack);
157
158     // Assertions and Cover for Full Condition
159
160     A_full: assert property (p_full);
161     C_full: cover property (p_full);
162
163     // Assertions and Cover for Empty Condition
164
165     A_empty: assert property (p_empty);
166     C_empty: cover property (p_empty);
167
168     // Assertions and Cover for Almost Full Condition
169
170     A_almostfull: assert property (p_almostfull);
171     C_almostfull: cover property (p_almostfull);
172
173     // Assertions and Cover for Almost Empty Condition
174
175     A_almostempty: assert property (p_almostempty);
176     C_almostempty: cover property (p_almostempty);
177
178     // Assertions and Cover for Underflow Condition
179
180     A_underflow: assert property (p_underflow);
181     C_underflow: cover property (p_underflow);
182
183     // Assertions and Cover for Overflow Condition
184
185     A_overflow: assert property (p_overflow);
186     C_overflow: cover property (p_overflow);
187
188     // Assertions and Cover for Counter Check
189
190     A_count_check: assert property (p_count_check);
191     C_count_check: cover property (p_count_check);
192

```

```

190  A_count_check: assert property (p_count_check);
191  C_count_check: cover property (p_count_check);
192
193  // Assertions and Cover for Write Pointer Check
194
195  A_wr_ptr_check: assert property (p_wr_ptr_check);
196  C_wr_ptr_check: cover property (p_wr_ptr_check);
197
198  // Assertions and Cover for Read Pointer Check
199
200  A_rd_ptr_check: assert property (p_rd_ptr_check);
201  C_rd_ptr_check: cover property (p_rd_ptr_check);
202
203  always_comb begin
204      if (!FIFO_IC.rst_n) begin
205
206          p_wr_ptr_reset: assert final (wr_ptr == 0);
207          C_wr_ptr_reset: cover (wr_ptr == 0);
208
209          p_rd_ptr_reset: assert final (rd_ptr == 0);
210          C_rd_ptr_reset: cover (rd_ptr == 0);
211
212          p_count_reset: assert final (count == 0);
213          C_count_reset: cover (count == 0);
214
215      end
216  end
217
218  endmodule
219

```

## Verification Plan:

	Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
1	FIFO_1	When the reset is asserted, the output data_out value should be low	Directed at the start of the sim, then randomized with constraint that drive the reset to be off most of the simulation time.	-	Immediate assertion to check the async reset functionality
2	FIFO_2	When wr_en is asserted and full is low wr_ack should be high	Randomization under constraints on the wr_en signal to be on 70% of the time and on the rd_en signal to be off 70% of the time	Cover all combination values of rd_en , wr_en and wr_ack	Concurrent assertion to check for it
3	FIFO_3	When wr_en is asserted and full is high then overflow should be high	Randomization under constraints on the wr_en signal to be on 70% of the time and on the rd_en signal to be off 70% of the time	Cover all combination values of rd_en , wr_en and overflow	Concurrent assertion to check for it
4	FIFO_4	When count is equal to FIFO depth, full value should be high	Randomization under constraints on the wr_en signal to be on 70% of the time and on the rd_en signal to be off 70% of the time	Cover all combination values of rd_en , wr_en and full	Concurrent assertion to check for it
5	FIFO_5	When count is equal to zero, empty value should be high	Randomization under constraints on the wr_en signal to be on 70% of the time and on the rd_en signal to be off 70% of the time	Cover all combination values of rd_en , wr_en and empty	Concurrent assertion to check for it
6	FIFO_6	When count is equal to FIFO depth -1, almostfull value should be high	Randomization under constraints on the wr_en signal to be on 70% of the time and on the rd_en signal to be off 70% of the time	Cover all combination values of rd_en , wr_en and almostfull	Concurrent assertion to check for it
7	FIFO_7	When count is equal to 1, almostempty value should be high	Randomization under constraints on the wr_en signal to be on 70% of the time and on the rd_en signal to be off 70% of the time	Cover all combination values of rd_en , wr_en and almostempty	Concurrent assertion to check for it
8					

9	FIFO_8	When rd_en is asserted and empty is high then underflow should be high	Randomization under constraints on the wr_en signal to be on 70% of the time and on the rd_en signal to be off 70% of the time	Cover all combination values of rd_en, wr_en and underflow	Concurrent assertion to check for it
10	FIFO_9	According to the given inputs the data_out should be correct	Randomization under constraints on the wr_en signal to be on 70% of the time and on the rd_en signal to be off 70% of the time		Check data function with reference model function to check the data_out
11	FIFO_10				

## Top file:

```

1 module top ();
2
3     bit clk;
4
5     initial begin
6         clk = 0;
7         forever
8             #2 clk = ~clk;
9
10    end
11
12    FIFO_interface FIFO_IC (clk);
13    FIFO_tb tb (FIFO_IC);
14    FIFO DUT (FIFO_IC);
15    monitor MONITOR (FIFO_IC);
16
17 endmodule

```

## Interface code:

```

1 interface FIFO_interface (clk);
2
3     parameter FIFO_WIDTH = 16;
4     parameter FIFO_DEPTH = 8;
5
6     input clk;
7
8
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;
13    logic full, empty, almostfull, almostempty, underflow;
14
15
16    modport TEST (output rst_n, data_in, wr_en, rd_en,
17                  input clk, data_out, wr_ack, overflow, full, empty, almostfull, almostempty, underflow);
18
19    modport DUT (input rst_n, data_in, wr_en, rd_en, clk,
20                output data_out, wr_ack, overflow, full, empty, almostfull, almostempty, underflow);
21
22    modport MONITOR (input rst_n, data_in, wr_en, rd_en, clk,
23                    data_out, wr_ack, overflow, full, empty, almostfull, almostempty, underflow);
24
25 endinterface

```

Test bench code:

```
1  ▼ module FIFO_tb(FIFO_interface.TEST FIFO_IC);
2
3      import shared_pkg::*;
4      import trans_pkg::*;
5
6      FIFO_transaction trans = new;
7
8  ▼ initial begin
9      test_finished = 0;
10     FIFO_IC.rst_n = 0;
11     @(negedge FIFO_IC.clk);
12     FIFO_IC.rst_n = 1;
13
14     @(negedge FIFO_IC.clk);
15
16  ▼     for (int i = 0; i < 10000; i++) begin
17
18         assert(trans.randomize());
19
20         FIFO_IC.rst_n = trans.rst_n;
21         FIFO_IC.wr_en = trans.wr_en;
22         FIFO_IC.rd_en = trans.rd_en;
23         FIFO_IC.data_in = trans.data_in;
24
25
26         @(negedge FIFO_IC.clk);
27     end
28
29     test_finished = 1;
30 end
31
32
33 endmodule
```

## Monitor code:

```
1  module monitor (FIFO_interface.MONITOR FIFO_IC);
2
3      import coverage_pkg::*;
4      import scoreboard_pkg::*;
5      import trans_pkg::*;
6      import shared_pkg::*;
7
8      FIFO_transaction transaction=new;
9      FIFO_scoreboard score=new;
10     FIFO_coverage coverage=new;
11
12     initial begin
13
14         forever begin
15
16             @(negedge FIFO_IC.clk);
17
18             transaction.rd_en = FIFO_IC.rd_en;
19             transaction.wr_en = FIFO_IC.wr_en;
20             transaction.wr_ack = FIFO_IC.wr_ack;
21             transaction.full = FIFO_IC.full;
22             transaction.empty = FIFO_IC.empty;
23             transaction.almostfull = FIFO_IC.almostfull;
24             transaction.almostempty = FIFO_IC.almostempty;
25             transaction.overflow = FIFO_IC.overflow;
26             transaction.underflow = FIFO_IC.underflow;
27             transaction.data_in = FIFO_IC.data_in;
28             transaction.rst_n = FIFO_IC.rst_n;
29             transaction.data_out = FIFO_IC.data_out;
30
31             fork
32                 begin
33                     coverage.sample_data(transaction);
34                 end
35                 begin
36                     score.check_data(transaction);
37                 end
38             join
39
40             if (test_finished) begin
41                 $display("Error count = %0d, correct count = %0d",Error_count, Correct_count);
42                 $stop;
43             end
44         end
45     end
46 endmodule
47
```



## Transaction package:

```
1 package trans_pkg;
2
3     parameter FIFO_WIDTH = 16;
4     parameter FIFO_DEPTH = 8;
5
6     class FIFO_transaction;
7
8         rand logic [FIFO_WIDTH-1:0] data_in;
9         logic clk;
10        rand logic rst_n, wr_en, rd_en;
11        logic [FIFO_WIDTH-1:0] data_out;
12        logic wr_ack, overflow;
13        logic full, empty, almostfull, almostempty, underflow;
14
15        int RD_EN_ON_DIST, WR_EN_ON_DIST;
16
17        function new(int x = 30, int y = 70);
18
19            RD_EN_ON_DIST = x;
20            WR_EN_ON_DIST = y;
21
22        endfunction
23
24        constraint A {rst_n dist {1:/90, 0:/10};}
25        constraint B {wr_en dist {1:/WR_EN_ON_DIST, 0:/100-WR_EN_ON_DIST};}
26        constraint C {rd_en dist {1:/RD_EN_ON_DIST, 0:/100-RD_EN_ON_DIST};}
27
28
29
30    endclass
31
32    endpackage : trans_pkg
```

## Coverage package:

```
1 package coverage_pkg;
2
3 import trans_pkg::*;
4
5 class FIFO_coverage;
6
7     FIFO_transaction F_cvg_txn;
8
9     covergroup cg ;
10
11         cross_0: cross F_cvg_txn.wr_en, F_cvg_txn.rd_en, F_cvg_txn.wr_ack;
12         cross_1: cross F_cvg_txn.wr_en, F_cvg_txn.rd_en, F_cvg_txn.overflow;
13         cross_2: cross F_cvg_txn.wr_en, F_cvg_txn.rd_en, F_cvg_txn.full;
14         cross_3: cross F_cvg_txn.wr_en, F_cvg_txn.rd_en, F_cvg_txn.empty;
15         cross_4: cross F_cvg_txn.wr_en, F_cvg_txn.rd_en, F_cvg_txn.almostfull;
16         cross_5: cross F_cvg_txn.wr_en, F_cvg_txn.rd_en, F_cvg_txn.almostempty;
17         cross_6: cross F_cvg_txn.wr_en, F_cvg_txn.rd_en, F_cvg_txn.underflow;
18
19     endgroup
20
21     function void sample_data(FIFO_transaction F_txn);
22         F_cvg_txn = F_txn;
23         cg.sample();
24     endfunction
25
26
27     function new();
28         cg=new();
29     endfunction
30
31 endclass
32
33 endpackage : coverage_pkg
```

## Scoreboard package:

```
1 package scoreboard_pkg;
2 import trans_pkg::*;
3 import shared_pkg::*;
4
5 Logic [FIFO_WIDTH-1:0] fifo_queue[$];
6
7 class FIFO_scoreboard;
8     Logic [FIFO_WIDTH-1:0] data_out_ref;
9     Logic wr_ack_ref;
10    Logic [2:0] wr_ptr, rd_ptr;
11    Logic [3:0] fifo_count;
12    Logic full_ref;
13    Logic empty_ref;
14
15
16    function void check_data(input FIFO_transaction F_txn );
17
18        reference_model(F_txn);
19        if (data_out_ref != F_txn.data_out) begin
20            $display("Error with data_out_ref = %0d and F_txn.data_out = %0d", data_out_ref,F_txn.data_out);
21            Error_count++;
22        end
23        else begin
24            Correct_count++;
25            $display("correct with data_out_ref = %0d and F_txn.data_out = %0d", data_out_ref,F_txn.data_out);
26        end
27    endfunction
28
29    function reference_model(input FIFO_transaction F_txn);
30
31        if (!F_txn.rst_n) begin
32            fifo_queue <= {};
33            fifo_count <= 0;
34        end
35        else begin
36            if (F_txn.wr_en && fifo_count < FIFO_DEPTH) begin
37                fifo_queue.push_back(F_txn.data_in);
38                fifo_count <= fifo_queue.size();
39            end
40
41            if (F_txn.rd_en && fifo_count != 0) begin
42                data_out_ref <= fifo_queue.pop_front();
43                fifo_count <= fifo_queue.size();
44            end
45        end
46
47        full_ref = (fifo_count == FIFO_DEPTH);
48        empty_ref = (fifo_count == 0);
49
50    endfunction
51 endclass
52
53 endpackage : scoreboard_pkg
```

## Shared package:

```
1 package shared_pkg;
2
3     int Error_count, Correct_count;
4     bit test_finished;
5
6 endpackage
```

Do file:

```
1 vlib work
2 vlog interface.sv pkg_trans.sv pkg_coverage.sv pkg_scoreboard.sv shared_pkg.sv FIFO.sv monitor.sv FIFO_tb.sv top.sv +cover -covercells
3 vsim -voptargs+=acc work.top -cover
4 add wave *
5 coverage save top.ucdb -onexit
6 run -all
```

Coverage report:

Name	Class Type	Coverage	Goal	% of Goal	Status	Included	Merge_instances	Get_inst_coverage	Comment
<b>CROSS cg::cross_0</b>		100.00%	100	100.00...		✓			
bin <auto[1],auto[1],auto[1]>		1309	1	100.00...		✓			
bin <auto[0],auto[1],auto[1]>		568	1	100.00...		✓			
bin <auto[1],auto[0],auto[1]>		3034	1	100.00...		✓			
bin <auto[0],auto[0],auto[1]>		1314	1	100.00...		✓			
bin <auto[1],auto[1],auto[0]>		802	1	100.00...		✓			
bin <auto[0],auto[1],auto[0]>		368	1	100.00...		✓			
bin <auto[1],auto[0],auto[0]>		1805	1	100.00...		✓			
bin <auto[0],auto[0],auto[0]>		801	1	100.00...		✓			
<b>CROSS cg::cross_1</b>		100.00%	100	100.00...		✓			
bin <auto[1],auto[1],auto[1]>		223	1	100.00...		✓			
bin <auto[0],auto[1],auto[1]>		126	1	100.00...		✓			
bin <auto[1],auto[0],auto[1]>		515	1	100.00...		✓			
bin <auto[0],auto[0],auto[1]>		217	1	100.00...		✓			
bin <auto[1],auto[1],auto[0]>		1888	1	100.00...		✓			
bin <auto[0],auto[1],auto[0]>		810	1	100.00...		✓			
bin <auto[1],auto[0],auto[0]>		4324	1	100.00...		✓			
bin <auto[0],auto[0],auto[0]>		1898	1	100.00...		✓			
<b>CROSS cg::cross_2</b>		100.00%	100	100.00...		✓			
bin <auto[1],auto[1],auto[1]>		239	1	100.00...		✓			
bin <auto[0],auto[1],auto[1]>		123	1	100.00...		✓			
bin <auto[1],auto[0],auto[1]>		507	1	100.00...		✓			
bin <auto[0],auto[0],auto[1]>		240	1	100.00...		✓			
bin <auto[1],auto[1],auto[0]>		1872	1	100.00...		✓			
bin <auto[0],auto[1],auto[0]>		813	1	100.00...		✓			
bin <auto[1],auto[0],auto[0]>		4332	1	100.00...		✓			
bin <auto[0],auto[0],auto[0]>		1875	1	100.00...		✓			
<b>CROSS cg::cross_3</b>		100.00%	100	100.00...		✓			
bin <auto[1],auto[1],auto[1]>		323	1	100.00...		✓			
bin <auto[0],auto[1],auto[1]>		146	1	100.00...		✓			
bin <auto[1],auto[0],auto[1]>		757	1	100.00...		✓			
bin <auto[0],auto[0],auto[1]>		320	1	100.00...		✓			
bin <auto[1],auto[1],auto[0]>		1788	1	100.00...		✓			

Name	Class Type	Coverage	Goal	% of Goal	Status	Included	Merge_instances	Get_inst_coverage	Comment
bin <auto[1],auto[0],auto[1]>		757	1	100.00...		✓			
bin <auto[0],auto[0],auto[1]>		320	1	100.00...		✓			
bin <auto[1],auto[1],auto[0]>		1788	1	100.00...		✓			
bin <auto[0],auto[1],auto[0]>		790	1	100.00...		✓			
bin <auto[1],auto[0],auto[0]>		4082	1	100.00...		✓			
bin <auto[0],auto[0],auto[0]>		1795	1	100.00...		✓			
<b>CROSS cg::cross_4</b>		100.00%	100	100.00...		✓			
bin <auto[1],auto[1],auto[1]>		207	1	100.00...		✓			
bin <auto[0],auto[1],auto[1]>		85	1	100.00...		✓			
bin <auto[1],auto[0],auto[1]>		481	1	100.00...		✓			
bin <auto[0],auto[0],auto[1]>		216	1	100.00...		✓			
bin <auto[1],auto[1],auto[0]>		1904	1	100.00...		✓			
bin <auto[0],auto[1],auto[0]>		851	1	100.00...		✓			
bin <auto[1],auto[0],auto[0]>		4358	1	100.00...		✓			
bin <auto[0],auto[0],auto[0]>		1899	1	100.00...		✓			
<b>CROSS cg::cross_5</b>		100.00%	100	100.00...		✓			
bin <auto[1],auto[1],auto[1]>		379	1	100.00...		✓			
bin <auto[0],auto[1],auto[1]>		157	1	100.00...		✓			
bin <auto[1],auto[0],auto[1]>		844	1	100.00...		✓			
bin <auto[0],auto[0],auto[1]>		310	1	100.00...		✓			
bin <auto[1],auto[1],auto[0]>		1732	1	100.00...		✓			
bin <auto[0],auto[1],auto[0]>		779	1	100.00...		✓			
bin <auto[1],auto[0],auto[0]>		3995	1	100.00...		✓			
bin <auto[0],auto[0],auto[0]>		1805	1	100.00...		✓			
<b>CROSS cg::cross_6</b>		100.00%	100	100.00...		✓			
bin <auto[1],auto[1],auto[1]>		136	1	100.00...		✓			
bin <auto[0],auto[1],auto[1]>		73	1	100.00...		✓			
bin <auto[1],auto[0],auto[1]>		291	1	100.00...		✓			
bin <auto[0],auto[0],auto[1]>		107	1	100.00...		✓			
bin <auto[1],auto[1],auto[0]>		1975	1	100.00...		✓			
bin <auto[0],auto[1],auto[0]>		863	1	100.00...		✓			
bin <auto[1],auto[0],auto[0]>		4548	1	100.00...		✓			
bin <auto[0],auto[0],auto[0]>		2008	1	100.00...		✓			

Covergroups										
Name	Class Type	Coverage	Goal	% of Goal	Status	Included	Merge_instances	Get_inst_coverage	Comment	
/coverage_pkg/FIFO_coverage		100.00%								
TYF8 cg		100.00%	100	100.00...					auto(1)	
CVP cg::({#F_cvg_bkn.wr_en_0#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.rd_en_1#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.underflow_2#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.wr_en_3#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.rd_en_4#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.almostempty_5#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.wr_en_6#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.rd_en_7#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.almostfull_8#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.wr_en_9#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.rd_en_10#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.empty_11#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.wr_en_12#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.rd_en_13#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.full_14#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.wr_en_15#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.rd_en_16#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.overflow_17#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.wr_en_18#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.rd_en_19#})		100.00%	100	100.00...						
CVP cg::({#F_cvg_bkn.wr_ack_20#})		100.00%	100	100.00...						
CROSS cg::cross_0		100.00%	100	100.00...						
B bin <auto[1],auto[1],auto[1]>		1309	1	100.00...						
B bin <auto[0],auto[1],auto[1]>		568	1	100.00...						
B bin <auto[1],auto[0],auto[1]>		3034	1	100.00...						
B bin <auto[0],auto[0],auto[1]>		1314	1	100.00...						
B bin <auto[1],auto[1],auto[0]>		802	1	100.00...						
B bin <auto[0],auto[1],auto[0]>		368	1	100.00...						
B bin <auto[1],auto[0],auto[0]>		1805	1	100.00...						
B bin <auto[0],auto[0],auto[0]>		801	1	100.00...						
CROSS cg::cross_1		100.00%	100	100.00...						

Assertions												
Name	Assertion Type	Language	Enable	Failure Count	Pass Count	Active Count	Memory	Peak Memory	Peak Memory Time	Cumulative Threads	ATV	Assertion Expression
/top/bt/anoribk#182146786#16#4#/#ubk#182146786#16...	Immediate	SVA	on	0	1	-	-	-	-	-	off	assert (/randomize(...))
/top/DUT/A_count_wr	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) di...
/top/DUT/A_count_rd	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) di...
/top/DUT/A_wr_ack	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) di...
/top/DUT/A_full	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) (...
/top/DUT/A_empty	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) (...
/top/DUT/A_almostfull	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) (...
/top/DUT/A_almostempty	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) (...
/top/DUT/A_underflow	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) di...
/top/DUT/A_overflow	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) di...
/top/DUT/A_count_check	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) co...
/top/DUT/A_wr_ptr_check	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) wr...
/top/DUT/A_rd_ptr_check	Concurrent	SVA	on	0	1	-	0B	0B	0 ns	0	off	assert( @(posedge FIFO_IC.clk) rd...
/top/DUT/p_wr_ptr_reset	Immediate	SVA	on	0	1	-	-	-	-	-	off	assert (wr_ptr==0)
/top/DUT/p_rd_ptr_reset	Immediate	SVA	on	0	1	-	-	-	-	-	off	assert (rd_ptr==0)
/top/DUT/p_count_reset	Immediate	SVA	on	0	1	-	-	-	-	-	off	assert (count==0)

Cover Directives														
Name	Language	Enabled	Log	Count	AtLeast	Limit	Weight	Cmplt %	Cmplt graph	Included	Memory	Peak Memory	Peak Memory Time	Cumulative Threads
/top/DUT/C_count...	SVA	✓	Off	3487	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_count...	SVA	✓	Off	630	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_w_ack	SVA	✓	Off	5001	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_full	SVA	✓	Off	998	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_empty	SVA	✓	Off	2368	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_almost...	SVA	✓	Off	903	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_almost...	SVA	✓	Off	1513	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_underf...	SVA	✓	Off	365	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_overfl...	SVA	✓	Off	619	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_count...	SVA	✓	Off	10002	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_wr_ptr...	SVA	✓	Off	10002	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/C_rd_ptr...	SVA	✓	Off	10002	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/c_wr_ptr...	SVA	✓	Off	951	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/c_rd_ptr...	SVA	✓	Off	1134	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0
/top/DUT/c_count...	SVA	✓	Off	912	1	Unli...	1	100%	<div></div>	✓	0	0	0 ns	0

```

=== Instance: /top/FIFO IC
=== Design Unit: work.FIFO_interface

```

Toggle Coverage:

Enabled Coverage	Bins	Hits	Misses	Coverage
-----	----	----	----	-----
Toggles	86	86	0	100.00%

```

=====
=== Instance: /top/tb
=== Design Unit: work.FIFO_tb
=====

Assertion Coverage:
  Assertions          1          1          0  100.00%
-----
Name                  File(Line)                  Failure      Pass
                        Count              Count
-----
/top/tb/#anonblk#182146786#16#4#/#ublk#182146786#16/immed_18
                        FIFO_tb.sv(18)              0            1
Statement Coverage:
  Enabled Coverage    Bins      Hits      Misses  Coverage
-----
Statements           15        15        0      100.00%

```

```

=====
=== Instance: /top/DUT
=== Design Unit: work.FIFO
=====

Assertion Coverage:
  Assertions          15          15          0  100.00%
-----
Name                  File(Line)                  Failure      Pass
                        Count              Count
-----
/top/DUT/A_count_wr    FIFO.sv(147)              0            1
/top/DUT/A_count_rd    FIFO.sv(152)              0            1
/top/DUT/A_w_ack       FIFO.sv(157)              0            1
/top/DUT/A_full        FIFO.sv(162)              0            1
/top/DUT/A_empty       FIFO.sv(167)              0            1
/top/DUT/A_almostfull  FIFO.sv(172)              0            1
/top/DUT/A_almostempty FIFO.sv(177)              0            1
/top/DUT/A_underflow   FIFO.sv(182)              0            1
/top/DUT/A_overflow    FIFO.sv(187)              0            1
/top/DUT/A_count_check FIFO.sv(192)              0            1
/top/DUT/A_wr_ptr_check FIFO.sv(197)              0            1
/top/DUT/A_rd_ptr_check FIFO.sv(202)              0            1
/top/DUT/p_wr_ptr_reset FIFO.sv(208)              0            1
/top/DUT/p_rd_ptr_reset FIFO.sv(211)              0            1
/top/DUT/p_count_reset FIFO.sv(214)              0            1
Branch Coverage:
  Enabled Coverage    Bins      Hits      Misses  Coverage
-----
Branches             29        29        0      100.00%

```

```

Branch Coverage:
  Enabled Coverage    Bins      Hits      Misses  Coverage
-----
Branches             29        29        0      100.00%

```

Directive Coverage:  
Directives                    15            15            0   100.00%

DIRECTIVE COVERAGE:

Name	Design Unit	Design UnitType	Lang	File(Line)	Hits	Status
/top/DUT/C_count_wr	FIFO	Verilog	SVA	FIFO.sv(148)	3487	Covered
/top/DUT/C_count_rd	FIFO	Verilog	SVA	FIFO.sv(153)	630	Covered
/top/DUT/C_w_ack	FIFO	Verilog	SVA	FIFO.sv(158)	5001	Covered
/top/DUT/C_full	FIFO	Verilog	SVA	FIFO.sv(163)	998	Covered
/top/DUT/C_empty	FIFO	Verilog	SVA	FIFO.sv(168)	2368	Covered
/top/DUT/C_almostfull	FIFO	Verilog	SVA	FIFO.sv(173)	903	Covered
/top/DUT/C_almostempty	FIFO	Verilog	SVA	FIFO.sv(178)	1513	Covered
/top/DUT/C_underflow	FIFO	Verilog	SVA	FIFO.sv(183)	365	Covered
/top/DUT/C_overflow	FIFO	Verilog	SVA	FIFO.sv(188)	619	Covered
/top/DUT/C_count_check	FIFO	Verilog	SVA	FIFO.sv(193)	10002	Covered
/top/DUT/C_wr_ptr_check	FIFO	Verilog	SVA	FIFO.sv(198)	10002	Covered
/top/DUT/C_rd_ptr_check	FIFO	Verilog	SVA	FIFO.sv(203)	10002	Covered
/top/DUT/c_wr_ptr_reset	FIFO	Verilog	SVA	FIFO.sv(209)	951	Covered
/top/DUT/c_rd_ptr_reset	FIFO	Verilog	SVA	FIFO.sv(212)	1134	Covered
/top/DUT/c_count_reset	FIFO	Verilog	SVA	FIFO.sv(215)	912	Covered

Statement Coverage:

Enabled Coverage	Bins	Hits	Misses	Coverage
Statements	25	25	0	100.00%

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

Toggle Coverage:

Enabled Coverage	Bins	Hits	Misses	Coverage
Toggles	20	20	0	100.00%

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

Covergroup Coverage:

Covergroups	1	na	na	100.00%
Coverpoints/Crosses	28	na	na	na
Covergroup Bins	98	98	0	100.00%

bin <auto[0],auto[0],auto[0]>                    2008            1            -            Cov

Statement Coverage:

Enabled Coverage	Bins	Hits	Misses	Coverage
Statements	3	3	0	100.00%

TOTAL COVERGROUP COVERAGE: 100.00% COVERGROUP TYPES: 1

DIRECTIVE COVERAGE:

Name	Design Unit	Design UnitType	Lang	File(Line)	Hits	Status
/top/DUT/C_count_wr	FIFO	Verilog	SVA	FIFO.sv(148)	3487	Covered
/top/DUT/C_count_rd	FIFO	Verilog	SVA	FIFO.sv(153)	630	Covered
/top/DUT/C_w_ack	FIFO	Verilog	SVA	FIFO.sv(158)	5001	Covered
/top/DUT/C_full	FIFO	Verilog	SVA	FIFO.sv(163)	998	Covered
/top/DUT/C_empty	FIFO	Verilog	SVA	FIFO.sv(168)	2368	Covered
/top/DUT/C_almostfull	FIFO	Verilog	SVA	FIFO.sv(173)	903	Covered
/top/DUT/C_almostempty	FIFO	Verilog	SVA	FIFO.sv(178)	1513	Covered
/top/DUT/C_underflow	FIFO	Verilog	SVA	FIFO.sv(183)	365	Covered
/top/DUT/C_overflow	FIFO	Verilog	SVA	FIFO.sv(188)	619	Covered
/top/DUT/C_count_check	FIFO	Verilog	SVA	FIFO.sv(193)	10002	Covered
/top/DUT/C_wr_ptr_check	FIFO	Verilog	SVA	FIFO.sv(198)	10002	Covered
/top/DUT/C_rd_ptr_check	FIFO	Verilog	SVA	FIFO.sv(203)	10002	Covered
/top/DUT/c_wr_ptr_reset	FIFO	Verilog	SVA	FIFO.sv(209)	951	Covered
/top/DUT/c_rd_ptr_reset	FIFO	Verilog	SVA	FIFO.sv(212)	1134	Covered
/top/DUT/c_count_reset	FIFO	Verilog	SVA	FIFO.sv(215)	912	Covered

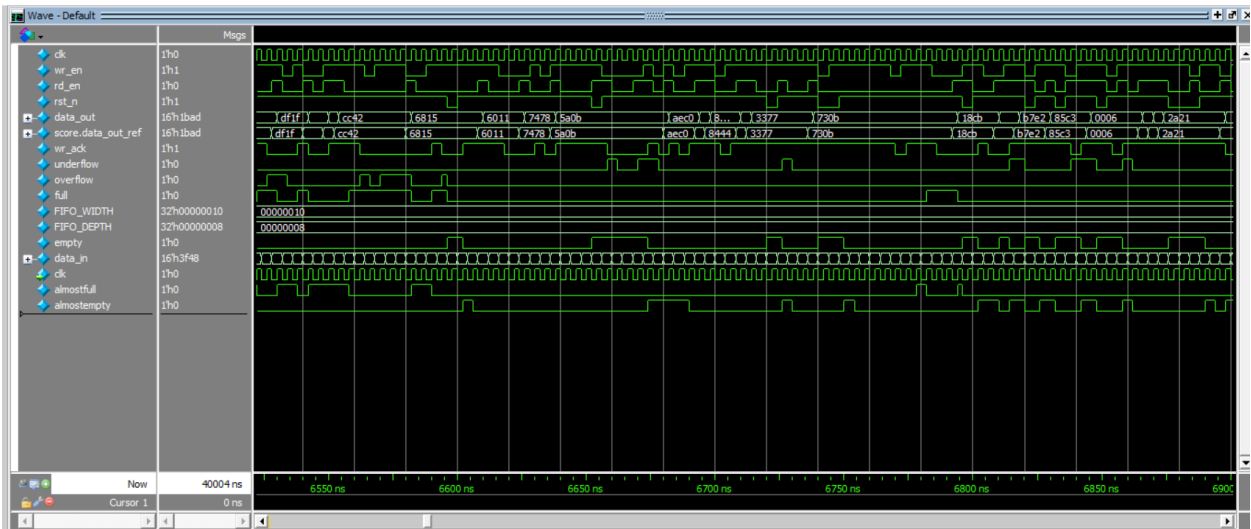
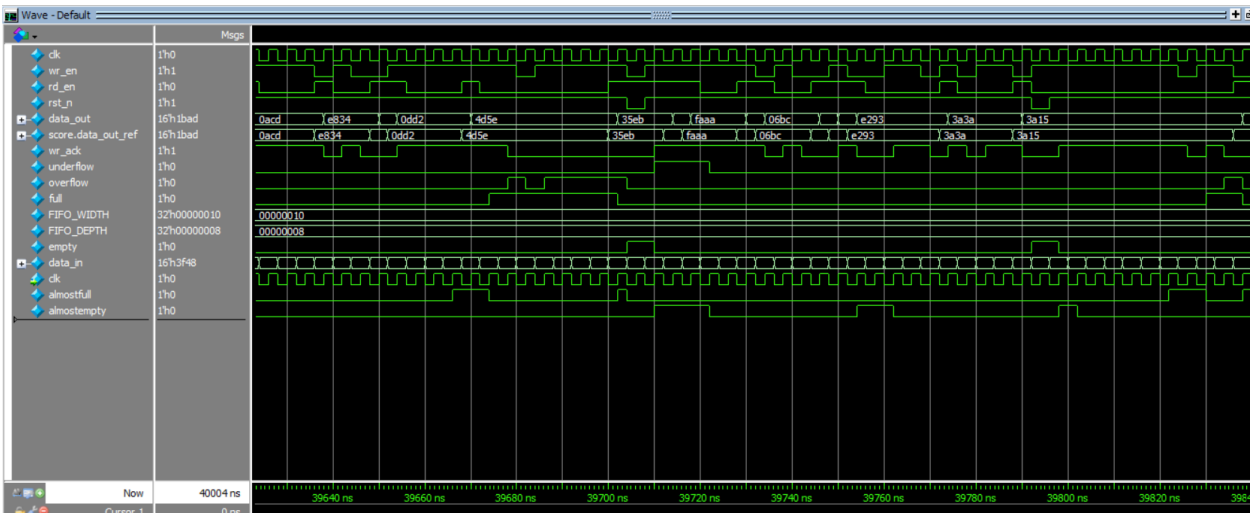
TOTAL DIRECTIVE COVERAGE: 100.00% COVERS: 15

ASSERTION RESULTS:

Name	File(Line)	Failure Count	Pass Count
/top/tb/#anonblk#182146786#16#4#/#ublk#182146786#16/immed__18			
	FIFO_tb.sv(18)	0	1
/top/DUT/A_count_wr	FIFO.sv(147)	0	1
/top/DUT/A_count_rd	FIFO.sv(152)	0	1
/top/DUT/A_w_ack	FIFO.sv(157)	0	1
/top/DUT/A_full	FIFO.sv(162)	0	1
/top/DUT/A_empty	FIFO.sv(167)	0	1
/top/DUT/A_almostfull	FIFO.sv(172)	0	1
/top/DUT/A_almostempty	FIFO.sv(177)	0	1
/top/DUT/A_underflow	FIFO.sv(182)	0	1
/top/DUT/A_overflow	FIFO.sv(187)	0	1
/top/DUT/A_count_check	FIFO.sv(192)	0	1
/top/DUT/A_wr_ptr_check	FIFO.sv(197)	0	1
/top/DUT/A_rd_ptr_check	FIFO.sv(202)	0	1
/top/DUT/p_wr_ptr_reset	FIFO.sv(208)	0	1
/top/DUT/p_rd_ptr_reset	FIFO.sv(211)	0	1
/top/DUT/p_count_reset	FIFO.sv(214)	0	1



Simulations:



---

```
# correct with data_out_ref = 42592 and F_txn.data_out = 42592
# correct with data_out_ref = 42592 and F_txn.data_out = 42592
# correct with data_out_ref = 42592 and F_txn.data_out = 42592
# correct with data_out_ref = 42592 and F_txn.data_out = 42592
# correct with data_out_ref = 42592 and F_txn.data_out = 42592
# correct with data_out_ref = 42592 and F_txn.data_out = 42592
# correct with data_out_ref = 29129 and F_txn.data_out = 29129
# correct with data_out_ref = 29129 and F_txn.data_out = 29129
# correct with data_out_ref = 4439 and F_txn.data_out = 4439
# correct with data_out_ref = 4439 and F_txn.data_out = 4439
# correct with data_out_ref = 4439 and F_txn.data_out = 4439
# correct with data_out_ref = 4439 and F_txn.data_out = 4439
# correct with data_out_ref = 4439 and F_txn.data_out = 4439
# correct with data_out_ref = 51761 and F_txn.data_out = 51761
# correct with data_out_ref = 51761 and F_txn.data_out = 51761
# correct with data_out_ref = 60277 and F_txn.data_out = 60277
# correct with data_out_ref = 60277 and F_txn.data_out = 60277
# correct with data_out_ref = 60277 and F_txn.data_out = 60277
# correct with data_out_ref = 60277 and F_txn.data_out = 60277
# correct with data_out_ref = 60277 and F_txn.data_out = 60277
# correct with data_out_ref = 60277 and F_txn.data_out = 60277
# correct with data_out_ref = 60277 and F_txn.data_out = 60277
# correct with data_out_ref = 5780 and F_txn.data_out = 5780
# correct with data_out_ref = 5780 and F_txn.data_out = 5780
# correct with data_out_ref = 5780 and F_txn.data_out = 5780
# correct with data_out_ref = 5780 and F_txn.data_out = 5780
# correct with data_out_ref = 5780 and F_txn.data_out = 5780
# correct with data_out_ref = 5780 and F_txn.data_out = 5780
# correct with data_out_ref = 5780 and F_txn.data_out = 5780
# correct with data_out_ref = 7085 and F_txn.data_out = 7085
# correct with data_out_ref = 7085 and F_txn.data_out = 7085
# correct with data_out_ref = 7085 and F_txn.data_out = 7085
# Error count = 0, correct count = 10002
```