**FIFO VERIFICATION**

**Test Plan:**

1. reset (0-1)

2. Check on underflow flag

3. Check on empty flag

4. Fill up the whole memory and while filling up the memory check on almost full flag

5. After FIFO is filled check overflow flag

6. Empty the memory and check on the almost empty flag

7. After FIFO is emptied check underflow

8. Randomize values to check the behavior of the flags

**Code Coverage Report:**

**=================================================================================**

**# === Instance: /FIFO\_top/f\_if**

**# === Design Unit: work.FIFO\_if**

**# =================================================================================**

**# Toggle Coverage:**

**# Enabled Coverage Bins Hits Misses Coverage**

**# ---------------- ---- ---- ------ --------**

**# Toggles 86 86 0 100.00%**

**#**

**# ================================Toggle Details================================**

**#**

**# Toggle Coverage for instance /FIFO\_top/f\_if --**

**#**

**# Node 1H->0L 0L->1H "Coverage"**

**# ---------------------------------------**

**# almostempty 1 1 100.00**

**# almostfull 1 1 100.00**

**# clk 1 1 100.00**

**# data\_in[15-0] 1 1 100.00**

**# data\_out[15-0] 1 1 100.00**

**# empty 1 1 100.00**

**# full 1 1 100.00**

**# overflow 1 1 100.00**

**# rd\_en 1 1 100.00**

**# rst\_n 1 1 100.00**

**# underflow 1 1 100.00**

**# wr\_ack 1 1 100.00**

**# wr\_en 1 1 100.00**

**#**

**# Total Node Count = 43**

**# Toggled Node Count = 43**

**# Untoggled Node Count = 0**

**#**

**# Toggle Coverage = 100.00% (86 of 86 bins)**

**#**

**# =================================================================================**

**# === Instance: /FIFO\_top/tb**

**# === Design Unit: work.FIFOtb**

**# =================================================================================**

**#**

**# Assertion Coverage:**

**# Assertions 9 2 7 22.22%**

**# --------------------------------------------------------------------**

**# Name File(Line) Failure Pass**

**# Count Count**

**# --------------------------------------------------------------------**

**# /FIFO\_top/tb/immed\_\_98**

**# tb.sv(98) 1 1**

**# /FIFO\_top/tb/#ublk#78493346#79/immed\_\_80**

**# tb.sv(80) 0 1**

**# /FIFO\_top/tb/write\_when\_full**

**# tb.sv(91) 11 1**

**# /FIFO\_top/tb/read\_when\_empty**

**# tb.sv(92) 7 1**

**# /FIFO\_top/tb/write\_when\_almostfull**

**# tb.sv(93) 5 0**

**# /FIFO\_top/tb/read\_when\_almostempty**

**# tb.sv(94) 1 1**

**# /FIFO\_top/tb/write tb.sv(95) 4 1**

**# /FIFO\_top/tb/reset\_assert**

**# tb.sv(96) 0 1**

**# /FIFO\_top/tb/read\_when\_empty2**

**# tb.sv(97) 4 0**

**# Branch Coverage:**

**# Enabled Coverage Bins Hits Misses Coverage**

**# ---------------- ---- ---- ------ --------**

**# Branches 3 3 0 100.00%**

**#**

**# ================================Branch Details================================**

**#**

**# Branch Coverage for instance /FIFO\_top/tb**

**#**

**# Line Item Count Source**

**# ---- ---- ----- ------**

**# File tb.sv**

**# ------------------------------------IF Branch------------------------------------**

**# 53 108 Count coming in to IF**

**# 53 1 48 if(wr\_ack)**

**#**

**# 55 1 27 else if(rd\_en)**

**#**

**# 33 All False Count**

**# Branch totals: 3 hits of 3 branches = 100.00%**

**#**

**#**

**#**

**# Covergroup Coverage:**

**# Covergroups 1 na na 100.00%**

**# Coverpoints/Crosses 2 na na na**

**# Covergroup Bins 4 4 0 100.00%**

**# ----------------------------------------------------------------------------------------------------------**

**# Covergroup Metric Goal Bins Status**

**#**

**# ----------------------------------------------------------------------------------------------------------**

**# TYPE /FIFO\_top/tb/fiforand/wr\_rd\_cov 100.00% 100 - Covered**

**# covered/total bins: 4 4 -**

**# missing/total bins: 0 4 -**

**# % Hit: 100.00% 100 -**

**# Coverpoint num\_of\_rd 100.00% 100 - Covered**

**# covered/total bins: 2 2 -**

**# missing/total bins: 0 2 -**

**# % Hit: 100.00% 100 -**

**# bin rd\_1 1 1 - Covered**

**# bin rd\_0 1 1 - Covered**

**# Coverpoint num\_of\_wr 100.00% 100 - Covered**

**# covered/total bins: 2 2 -**

**# missing/total bins: 0 2 -**

**# % Hit: 100.00% 100 -**

**# bin wr\_1 1 1 - Covered**

**# bin wr\_0 1 1 - Covered**

**#**

**# Directive Coverage:**

**# Directives 9 7 2 77.77%**

**#**

**# DIRECTIVE COVERAGE:**

**# --------------------------------------------------------------------------------------------**

**# Name Design Design Lang File(Line) Hits Status**

**# Unit UnitType**

**# --------------------------------------------------------------------------------------------**

**# /FIFO\_top/tb/cover\_\_reset FIFOtb Verilog SVA tb.sv(139) 3 Covered**

**# /FIFO\_top/tb/cover\_\_write\_whilenotfull FIFOtb Verilog SVA tb.sv(138) 38 Covered**

**# /FIFO\_top/tb/cover\_\_read\_while\_almostempty**

**# FIFOtb Verilog SVA tb.sv(137) 2 Covered**

**# /FIFO\_top/tb/cover\_\_write\_while\_almostfull**

**# FIFOtb Verilog SVA tb.sv(136) 0 ZERO**

**# /FIFO\_top/tb/cover\_\_read\_while\_empty2 FIFOtb Verilog SVA tb.sv(135) 0 ZERO**

**# /FIFO\_top/tb/cover\_\_read\_while\_empty FIFOtb Verilog SVA tb.sv(134) 9 Covered**

**# /FIFO\_top/tb/cover\_read\_1 FIFOtb Verilog SVA tb.sv(133) 29 Covered**

**# /FIFO\_top/tb/cover\_\_write\_while\_full FIFOtb Verilog SVA tb.sv(132) 9 Covered**

**# /FIFO\_top/tb/cover\_\_read FIFOtb Verilog SVA tb.sv(131) 29 Covered**

**# Statement Coverage:**

**# Enabled Coverage Bins Hits Misses Coverage**

**# ---------------- ---- ---- ------ --------**

**# Statements 39 39 0 100.00%**

**#**

**# ================================Statement Details================================**

**#**

**# Statement Coverage for instance /FIFO\_top/tb --**

**#**

**# Line Item Count Source**

**# ---- ---- ----- ------**

**# File tb.sv**

**# 1 module FIFOtb(FIFO\_if.TEST f\_if);**

**#**

**# 2 class fiforand;**

**#**

**# 3 rand bit rd\_enable,wr\_enable;**

**#**

**# 4 rand bit [15:0] data;**

**#**

**# 5 rand bit reset;**

**#**

**# 6 constraint rst{**

**#**

**# 7 reset dist{1:=98,0:=2};**

**#**

**# 8 }**

**#**

**# 9 constraint ram{**

**#**

**# 10 wr\_enable!=rd\_enable;**

**#**

**# 11 rd\_enable dist{1:=50,0:=50};**

**#**

**# 12 wr\_enable dist{1:=50,0:=50};**

**#**

**# 13 }**

**#**

**# 14 covergroup wr\_rd\_cov();**

**#**

**# 15 num\_of\_rd : coverpoint rd\_enable iff (!reset){**

**#**

**# 16 bins rd\_1={1};**

**#**

**# 17 bins rd\_0={0};**

**#**

**# 18 }**

**#**

**# 19 num\_of\_wr : coverpoint wr\_enable iff (!reset){**

**#**

**# 20 bins wr\_1={1};**

**#**

**# 21 bins wr\_0={0};**

**#**

**# 22 }**

**#**

**# 23 endgroup**

**#**

**# 24 function new();**

**#**

**# 25 1 1 wr\_rd\_cov=new();**

**#**

**# 26 endfunction**

**#**

**# 27**

**#**

**# 28 endclass**

**#**

**# 29 parameter WIDTH=16;**

**#**

**# 30 logic wr\_en,rd\_en,rst\_n,//i/p**

**#**

**# 31 full,almostfull,empty,almostempty,overflow,underflow,wr\_ack;**

**#**

**# 32 logic [WIDTH-1:0] data\_in,data\_out;**

**#**

**# 33 1 240 assign clk=f\_if.clk;**

**#**

**# 34 1 10 assign data\_out = f\_if.data\_in;**

**#**

**# 35 1 24 assign full = f\_if.full;**

**#**

**# 36 1 26 assign almostfull = f\_if.almostfull;**

**#**

**# 37 1 13 assign empty =f\_if.empty ;**

**#**

**# 38 1 20 assign almostempty = f\_if.almostempty;**

**#**

**# 39 1 11 assign overflow = f\_if.overflow ;**

**#**

**# 40 1 15 assign underflow = f\_if.underflow;**

**#**

**# 41 1 46 assign wr\_ack=f\_if.wr\_ack;**

**#**

**# 42 assign f\_if.rst\_n = rst\_n;**

**#**

**# 43 assign f\_if.data\_in = data\_in;**

**#**

**# 44 assign f\_if.wr\_en = wr\_en;**

**#**

**# 45 assign f\_if.rd\_en = rd\_en;**

**#**

**# 46 logic [WIDTH-1:0] FIFOQ[$];**

**#**

**# 47 int i;**

**#**

**# 48 logic [15:0] check;**

**#**

**# 49 1 1 fiforand wr\_rd=new();**

**#**

**# 50**

**#**

**# 51**

**#**

**# 52 1 108 always\_comb begin**

**#**

**# 53 if(wr\_ack)**

**#**

**# 54 1 48 FIFOQ.push\_front(data\_in);**

**#**

**# 55 else if(rd\_en)**

**#**

**# 56 1 27 check=FIFOQ.pop\_back;**

**#**

**# 57 end**

**#**

**# 58**

**#**

**# 59**

**#**

**# 60 initial begin**

**#**

**# 61 1 1 rst\_n =0;**

**#**

**# 62 1 1 #2**

**#**

**# 63 1 1 rst\_n=1;**

**#**

**# 64 end**

**#**

**# 65 initial begin**

**#**

**# 66 1 1 @(posedge clk)**

**#**

**# 67 1 1 rd\_en=1;**

**#**

**# 68 1 1 #2**

**#**

**# 69 1 1 rd\_en=0;**

**#**

**# 70 1 1 wr\_en=1;**

**#**

**# 71 1 1 for (i=0;i<9;i++)begin**

**#**

**# 71 2 9**

**# 72 1 9 data\_in=$random();**

**#**

**# 73 1 9 #2;**

**#**

**# 74 end**

**#**

**# 75 1 1 rd\_en=1;**

**#**

**# 76 1 1 wr\_en=0;**

**#**

**# 77 1 1 repeat(9) #2;**

**#**

**# 78 1 1 $display("starting randomization");**

**#**

**# 79 1 1 for (i=0;i<100;i++)begin**

**#**

**# 79 2 100**

**# 80 assert(wr\_rd.randomize());**

**#**

**# 81 1 100 rst\_n=wr\_rd.reset;**

**#**

**# 82 1 100 wr\_rd.wr\_rd\_cov.sample();**

**#**

**# 83 1 100 wr\_en=wr\_rd.wr\_enable;**

**#**

**# 84 1 100 rd\_en=wr\_rd.rd\_enable;**

**#**

**# 85 1 100 #2;**

**#**

**# 86 end**

**#**

**# 87 1 1 $stop;**

**#**

**# 88 end**

**#**

**# 89**

**#**

**# 90 //readerror: assert property(read);**

**#**

**# 91 write\_when\_full:assert property(write\_while\_full);**

**#**

**# 92 read\_when\_empty:assert property(read\_while\_empty);**

**#**

**# 93 write\_when\_almostfull:assert property(write\_while\_almostfull);**

**#**

**# 94 read\_when\_almostempty:assert property(read\_while\_almostempty);**

**#**

**# 95 write:assert property(write\_whilenotfull);**

**#**

**# 96 reset\_assert:assert property(reset);**

**#**

**# 97 read\_when\_empty2:assert property(read\_while\_empty2);**

**#**

**# 98 1 119 always @(posedge clk) assert (!(wr\_en&&rd\_en))**

**#**

**#**

**# Toggle Coverage:**

**# Enabled Coverage Bins Hits Misses Coverage**

**# ---------------- ---- ---- ------ --------**

**# Toggles 182 131 51 71.97%**

**#**

**# ================================Toggle Details================================**

**#**

**# Toggle Coverage for instance /FIFO\_top/tb --**

**#**

**# Node 1H->0L 0L->1H "Coverage"**

**# ---------------------------------------**

**# almostempty 1 1 100.00**

**# almostfull 1 1 100.00**

**# check[0-15] 1 1 100.00**

**# clk 1 1 100.00**

**# data\_in[0-15] 1 1 100.00**

**# data\_out[0-15] 1 1 100.00**

**# empty 1 1 100.00**

**# full 1 1 100.00**

**# i[0-5] 1 1 100.00**

**# i[6] 0 1 50.00**

**# i[7-31] 0 0 0.00**

**# 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 = 91**

**# Toggled Node Count = 65**

**# Untoggled Node Count = 26**

**#**

**# Toggle Coverage = 71.97% (131 of 182 bins)**

**#**

**# =================================================================================**

**# === Instance: /FIFO\_top**

**# === Design Unit: work.FIFO\_top**

**# =================================================================================**

**# Statement Coverage:**

**# Enabled Coverage Bins Hits Misses Coverage**

**# ---------------- ---- ---- ------ --------**

**# Statements 4 4 0 100.00%**

**#**

**# ================================Statement Details================================**

**#**

**# Statement Coverage for instance /FIFO\_top --**

**#**

**# Line Item Count Source**

**# ---- ---- ----- ------**

**# File top.sv**

**# 1 module FIFO\_top();**

**#**

**# 2 bit clk;**

**#**

**# 3**

**#**

**# 4 //clock generation**

**#**

**# 5 initial begin**

**#**

**# 6 1 1 clk = 0;**

**#**

**# 7 1 1 forever**

**#**

**# 8 1 239 #1 clk = ~clk;**

**#**

**# 8 2 238**

**#**

**# Toggle Coverage:**

**# Enabled Coverage Bins Hits Misses Coverage**

**# ---------------- ---- ---- ------ --------**

**# Toggles 2 2 0 100.00%**

**#**

**# ================================Toggle Details================================**

**#**

**# Toggle Coverage for instance /FIFO\_top --**

**#**

**# Node 1H->0L 0L->1H "Coverage"**

**# ---------------------------------------**

**# clk 1 1 100.00**

**#**

**# Total Node Count = 1**

**# Toggled Node Count = 1**

**# Untoggled Node Count = 0**

**#**

**# Toggle Coverage = 100.00% (2 of 2 bins)**

**#**

**# COVERGROUP COVERAGE:**

**# ----------------------------------------------------------------------------------------------------------**

**# Covergroup Metric Goal Bins Status**

**#**

**# ----------------------------------------------------------------------------------------------------------**

**# TYPE /FIFO\_top/tb/fiforand/wr\_rd\_cov 100.00% 100 - Covered**

**# covered/total bins: 4 4 -**

**# missing/total bins: 0 4 -**

**# % Hit: 100.00% 100 -**

**# Coverpoint num\_of\_rd 100.00% 100 - Covered**

**# covered/total bins: 2 2 -**

**# missing/total bins: 0 2 -**

**# % Hit: 100.00% 100 -**

**# bin rd\_1 1 1 - Covered**

**# bin rd\_0 1 1 - Covered**

**# Coverpoint num\_of\_wr 100.00% 100 - Covered**

**# covered/total bins: 2 2 -**

**# missing/total bins: 0 2 -**

**# % Hit: 100.00% 100 -**

**# bin wr\_1 1 1 - Covered**

**# bin wr\_0 1 1 - Covered**

**#**

**# TOTAL COVERGROUP COVERAGE: 100.00% COVERGROUP TYPES: 1**

**#**

**# DIRECTIVE COVERAGE:**

**# --------------------------------------------------------------------------------------------**

**# Name Design Design Lang File(Line) Hits Status**

**# Unit UnitType**

**# --------------------------------------------------------------------------------------------**

**# /FIFO\_top/tb/cover\_\_reset FIFOtb Verilog SVA tb.sv(139) 3 Covered**

**# /FIFO\_top/tb/cover\_\_write\_whilenotfull FIFOtb Verilog SVA tb.sv(138) 38 Covered**

**# /FIFO\_top/tb/cover\_\_read\_while\_almostempty**

**# FIFOtb Verilog SVA tb.sv(137) 2 Covered**

**# /FIFO\_top/tb/cover\_\_write\_while\_almostfull**

**# FIFOtb Verilog SVA tb.sv(136) 0 ZERO**

**# /FIFO\_top/tb/cover\_\_read\_while\_empty2 FIFOtb Verilog SVA tb.sv(135) 0 ZERO**

**# /FIFO\_top/tb/cover\_\_read\_while\_empty FIFOtb Verilog SVA tb.sv(134) 9 Covered**

**# /FIFO\_top/tb/cover\_read\_1 FIFOtb Verilog SVA tb.sv(133) 29 Covered**

**# /FIFO\_top/tb/cover\_\_write\_while\_full FIFOtb Verilog SVA tb.sv(132) 9 Covered**

**# /FIFO\_top/tb/cover\_\_read FIFOtb Verilog SVA tb.sv(131) 29 Covered**

**#**

**# TOTAL DIRECTIVE COVERAGE: 77.77% COVERS: 9**

**#**

**# ASSERTION RESULTS:**

**# --------------------------------------------------------------------**

**# Name File(Line) Failure Pass**

**# Count Count**

**# --------------------------------------------------------------------**

**# /FIFO\_top/tb/immed\_\_98**

**# tb.sv(98) 1 1**

**# /FIFO\_top/tb/#ublk#78493346#79/immed\_\_80**

**# tb.sv(80) 0 1**

**# /FIFO\_top/tb/write\_when\_full**

**# tb.sv(91) 11 1**

**# /FIFO\_top/tb/read\_when\_empty**

**# tb.sv(92) 7 1**

**# /FIFO\_top/tb/write\_when\_almostfull**

**# tb.sv(93) 5 0**

**# /FIFO\_top/tb/read\_when\_almostempty**

**# tb.sv(94) 1 1**

**# /FIFO\_top/tb/write tb.sv(95) 4 1**

**# /FIFO\_top/tb/reset\_assert**

**# tb.sv(96) 0 1**

**# /FIFO\_top/tb/read\_when\_empty2**

**# tb.sv(97) 4 0**

**#**

**# Total Coverage By Instance (filtered view): 80.18%**

**Assertion:**

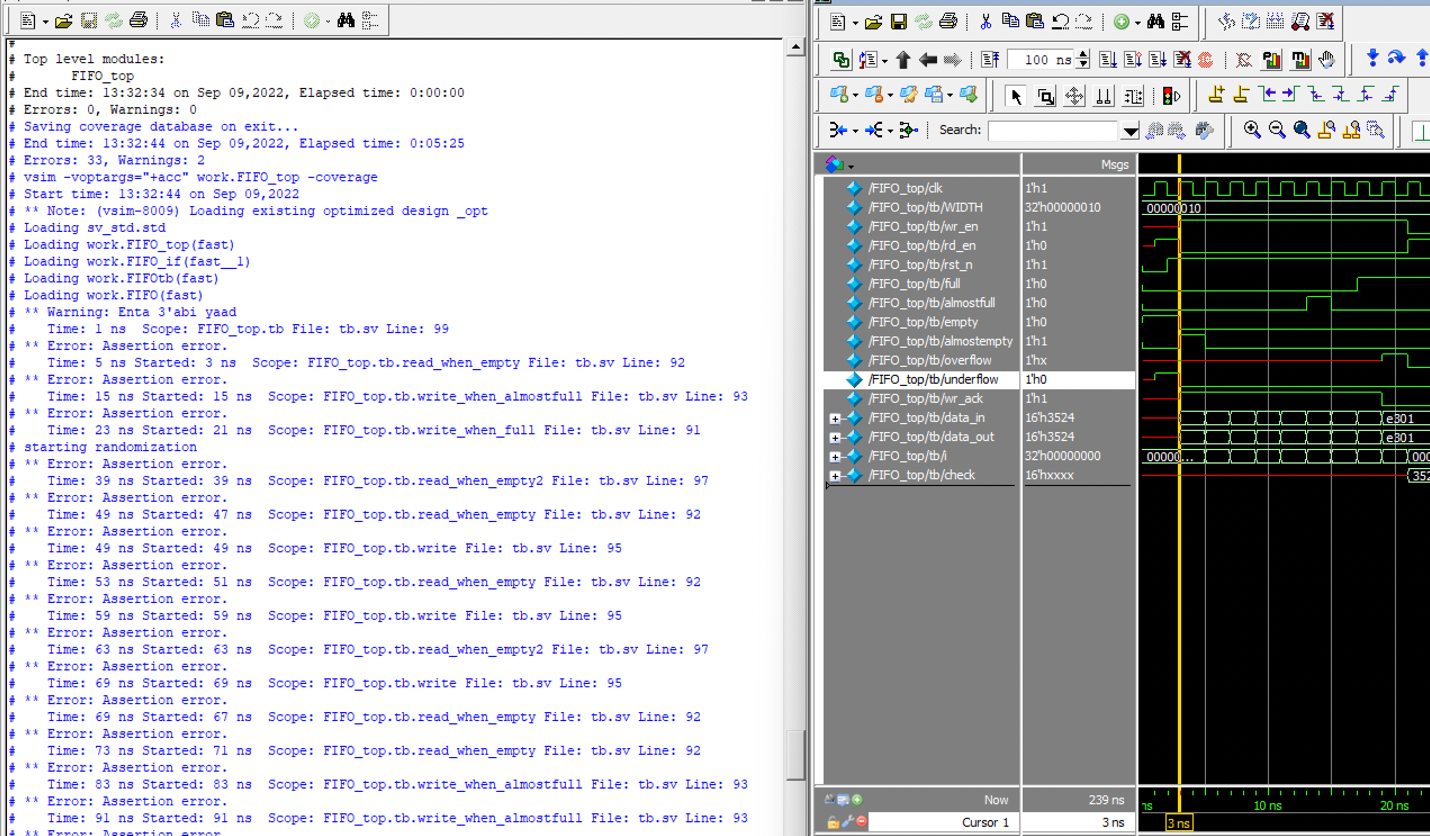
For the following assertion I used a queue array and popped the right value of dout into a variable named ‘checka’

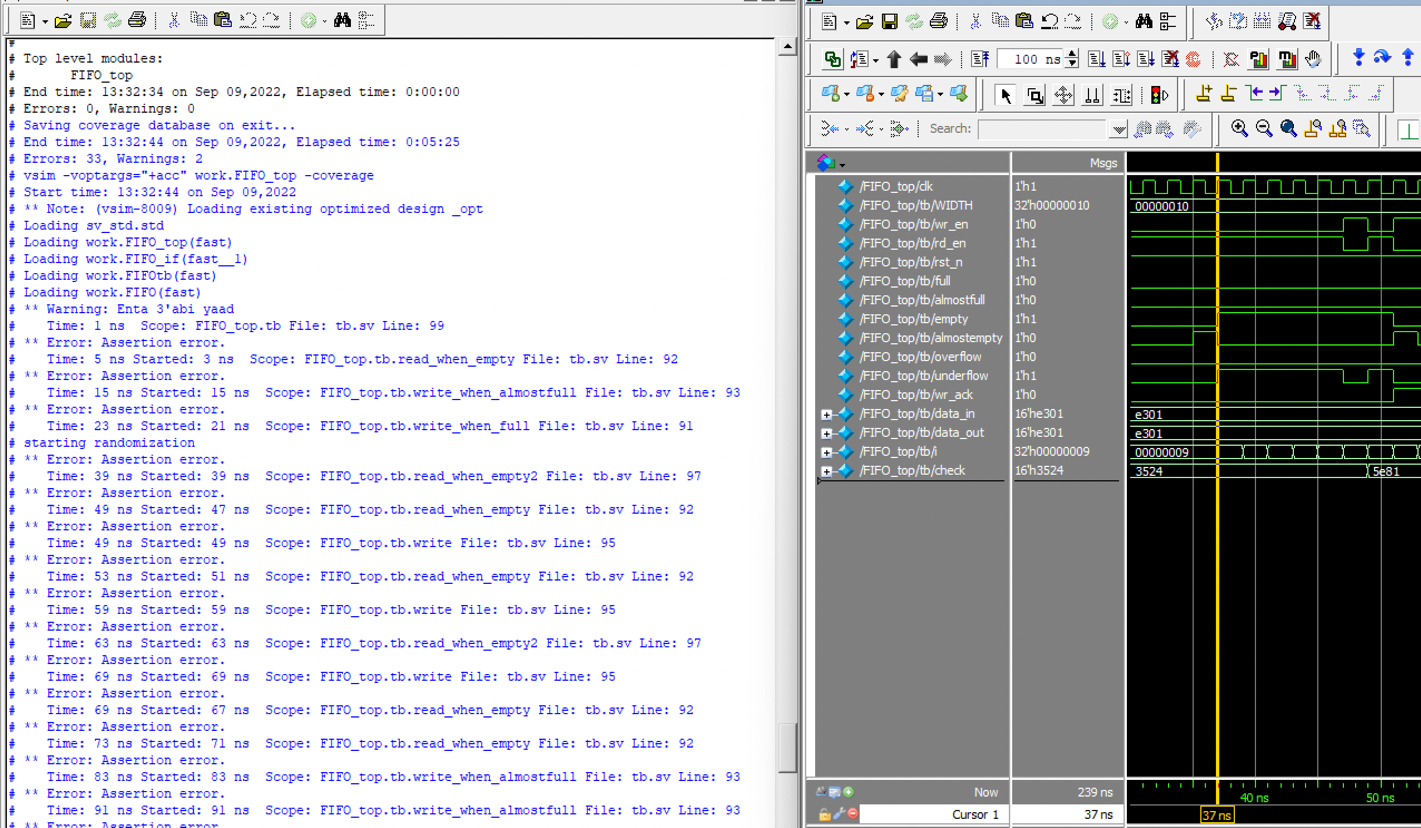
|  |  |
| --- | --- |
| **Feature** | **Assertion** |
| When full flag is on and there is a write signal overflow flag should be on after 1 clock cycle | @(posedge clk) full&&wr\_en|>##1(!wr\_ack&&overflow); |
| When empty flag is on and there is a read signal underflow flag should be on after 1 clock cycle | @(posedge clk) empty&&rd\_en|->##1 (underflow); |
| Checking if underflow flag is raised early or not | @(posedge clk) $rose(empty)&&rd\_en|->(!underflow); |
| When almost full flag is raised and write enable is on full flag should be on after 1 clock cycle | @(posedge clk) almostfull&&wr\_en|->##1(full); |
| When almost empty flag is raised and read signal is on, empty flag should be on after 1 clock cycle | @(posedge clk) almostempty&&rd\_en|->##1(empty); |
| Checking if write acknowledge is on after write signal is recieved | @(posedge clk) !full&&wr\_en|->(wr\_ack); |
| When reset is received FIFO should be emptied | @(posedge clk) !rst\_n|->(empty); |

**Bug Report:**

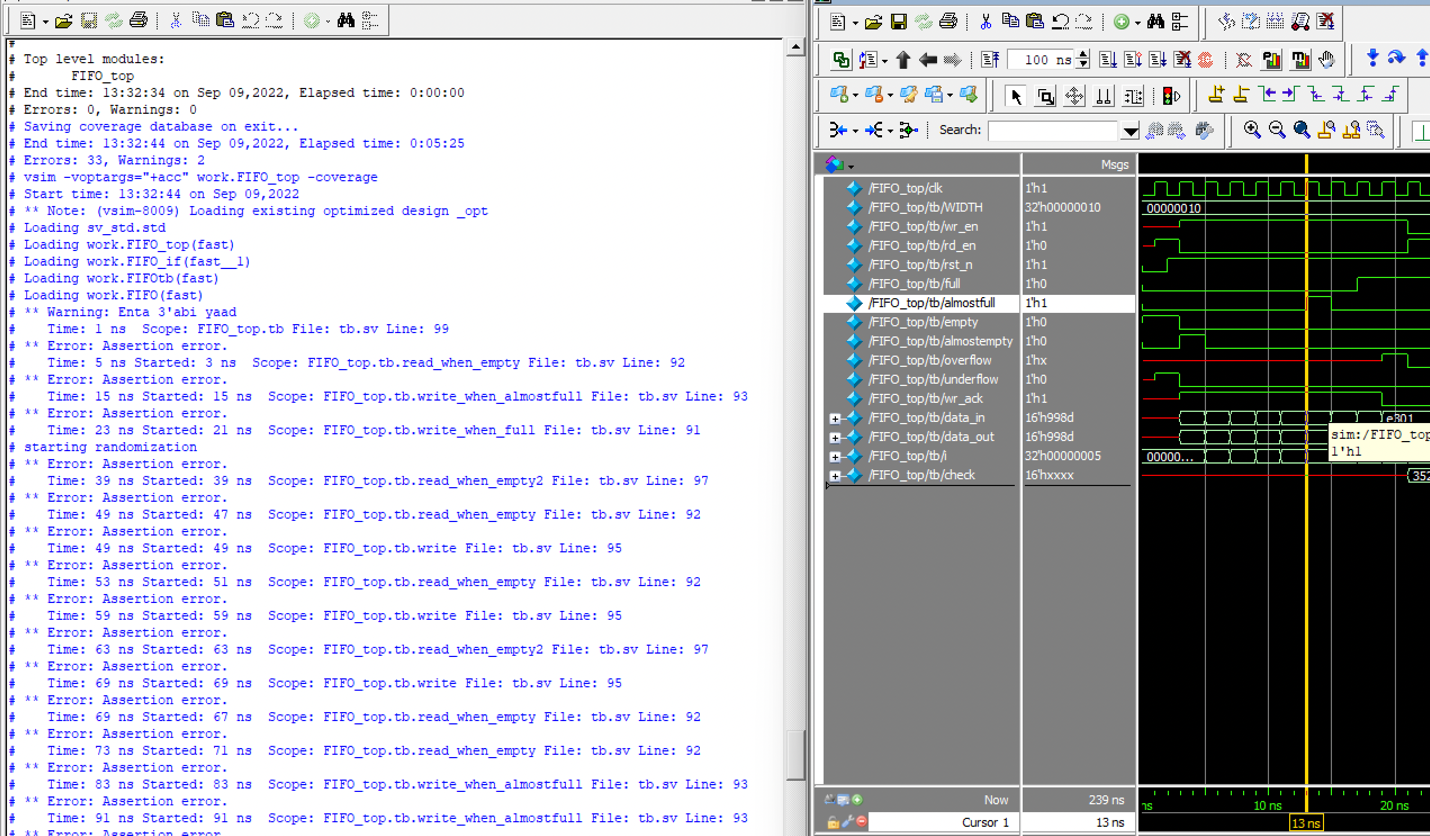
1. Fifo does not pop values on Dout
2. Din=Dout, Din is the same as Din
3. Underflow flag is raised in the same clock as when the empty flag was raised
4. Full flag is raised after the almost full flag by 2 clock cycles not 1
5. Overflow flag is not raised if write signal is high when the fifo is full

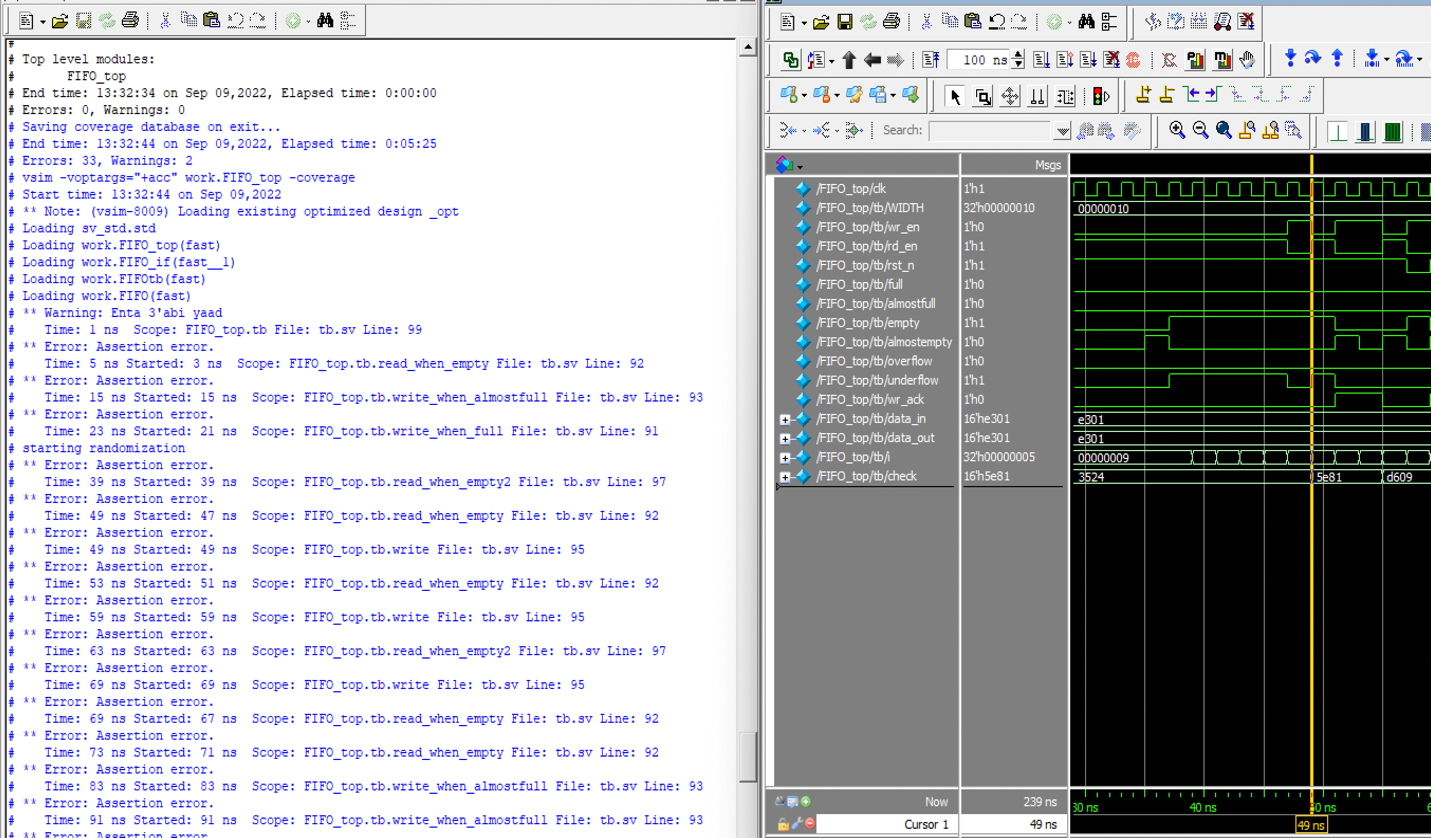
**Screenshots**

**Read when empty Read when empty 2**

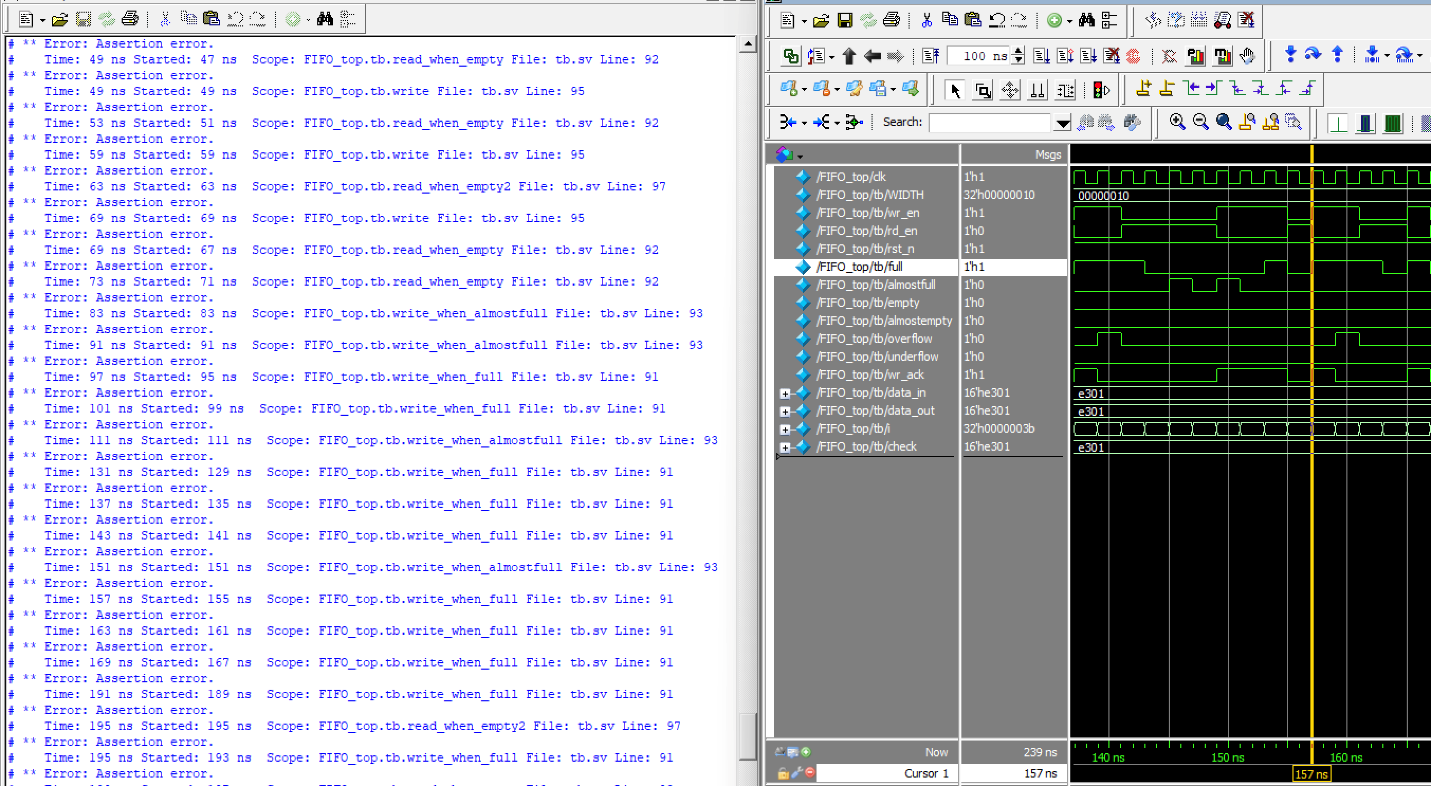
****

**Write when almost full**

****

**Write when there is a reset **

**Write when full**

****