Group 3 SystemVerilog Testbench

2025 AUG 19th

Tom
Aiman
Ang

Components

Sequence

Fixed sequence + Randomised sequence

Scoreboard

Queue to store expected data at send and compare it to actual data when finish sending

Coverage

Dynamic coverage points that changes to number of operating bits

Assertion

Checks: MSB transfers, done flags, reset, no operation

Fixed Sequence

```
        $display("\n------------------------ FIXED SEQUENCE INPUT -------------------\n");
        $display("Initial reset complete. Starting test sequences.");

        // Test 2.2 TX Data MSB -> LSB
        reset();
        #100;
        $display("TEST: TX Data MSB -> LSB (TEST ID 2.2)");
        assert (gen.randomize()) else $fatal ("Randomization failed!");
        req             = 2'b01;
        din_master      = gen.din_master;
        din_slave       = gen.din_slave;
        wait_duration   = 8'd0;

        @(negedge sclk)
        for (int i = `SPI_TRF_BIT-1; i >= 0; i--) begin
            @(negedge sclk)
            assert (din_master[i] === dout_slave[0])
                else $error("Bit mismatch: Expected MSB = %b, Got = %b", din_master[i], dout_slave[0]);
        end

        // Test 7.1 Reset on transfer
        reset();
        #100;
        $display("TEST: Reset on transfer (TEST ID 7.1)");
        assert (gen.randomize()) else $fatal ("Randomization failed!");
        req             = 2'b01;
        din_master      = gen.din_master;
        din_slave       = gen.din_slave;
        wait_duration   = 8'd0;
        #1000;
        @(posedge sclk);
        reset();
```

Randomised Sequence

Create class for Randomization sequence

```
200
201     class tx_rx_rand;
202
203                 rand logic [(`SPI_TRF_BIT-1):0] din_master;
204                 rand logic [(`SPI_TRF_BIT-1):0] din_slave;
205                 rand logic [1:0] req;
206                 rand logic [7:0] wait_duration;
207                 constraint ran_range{
208                         din_master      inside {[0:(2**`SPI_TRF_BIT)-1]};
209                         din_slave       inside {[0:(2**`SPI_TRF_BIT)-1]};
210                         req             inside {[0:3]};
211                         wait_duration   inside {[0:256]};
212                         }
213
214     endclass
215     tx_rx_rand gen;
216
```

Contructing the class instantiate

```
    initial begin
        bit cs_went_high        = 0;
        clk                     = 0;
        req                     = 0;
        din_master              = 0;
        din_slave               = 0;
        wait_duration           = 0;

        scoreboard_inst         = new();
        gen                     = new();
```

Randomised Sequence

- Randomised request
- Randomised din_master din_slave inputs
- Randomised wait_duration

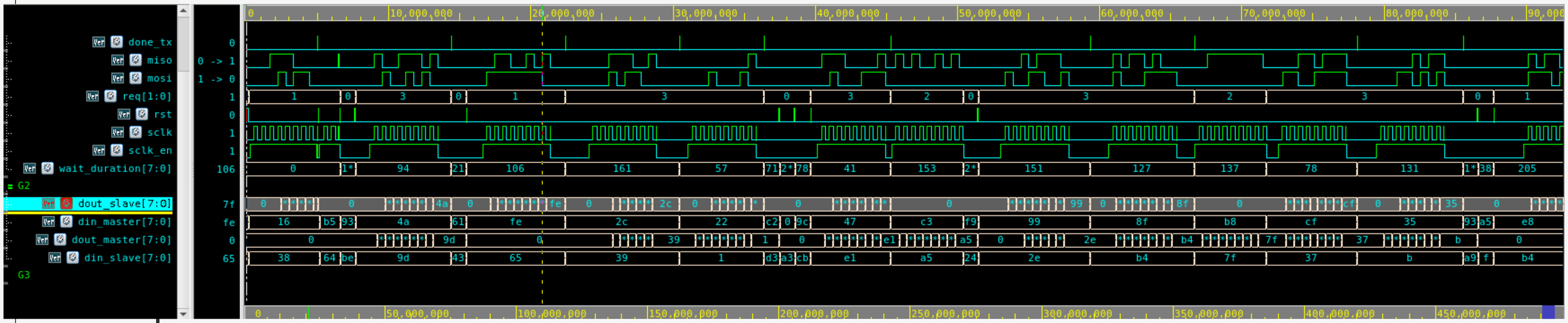
```
            @(posedge sclk);
            reset();

            // Test 6.5
            $display("\n------------------------- RANDOMIZED INPUT --------------------\n");
            repeat (100) begin
                    assert (gen.randomize()) else $fatal ("Randomization failed!");

                    req                             = gen.req;

                    @(posedge clk);
                    din_master              <= gen.din_master;
                    din_slave               <= gen.din_slave;
                    wait_duration   <= gen.wait_duration;

                    if (req == 2'b01) begin
                            @(posedge clk);
                            scoreboard_inst.push_tx_data(req, din_master);
                            @(posedge done_tx);
                            scoreboard_inst.check_tx_data();
                    end else if (req == 2'b10) begin
                            @(posedge clk);
                            scoreboard_inst.push_rx_data(req, din_slave);
                            @(posedge done_rx);
                            scoreboard_inst.check_rx_data();
                    end else if (req == 2'b11) begin
                            @(posedge clk);
                            scoreboard_inst.push_tx_data(req, din_master);
                            scoreboard_inst.push_rx_data(req, din_slave);
                            fork
                                    @(posedge done_tx);
                                    @(posedge done_rx);
                            join
                            scoreboard_inst.check_tx_data();
                            scoreboard_inst.check_rx_data();
                    end else if (req == 2'b00) begin
                            #1000;
                            reset();
                            continue;
                    end
            end

        // Finalize test and report
        #100;
        scoreboard_inst.summary();
        $display("TEST: All sequences completed.");
        $finish;
    end
```

Randomised Sequence

- Randomised request
- Randomised din_master din_slave inputs
- Randomised wait_duration

```
------------------------ 15 BIT RECEIVED -------------------


------------------------ FIXED SEQUENCE INPUT -------------------

Initial reset complete. Starting test sequences.
Initial reset complete. Starting test sequences.
TEST: TX Data MSB -> LSB (Test ID 2.1)
TEST: Reset on transfer

------------------------ RANDOMIZED INPUT -------------------

[11655000][SEQ][REQ=3] Pushed TX data 0x2524 to queue.
[11655000][SEQ][REQ=3] Pushed RX data 0x4ecb to queue.
[22245000][SCB][PASS] TX data matched! Sent: 0x2524, Received: 0x2524
[22245000][SCB][PASS] RX data matched! Sent: 0x4ecb, Received: 0x4ecb
[23365000][SEQ][REQ=1] Pushed TX data 0x7f6a to queue.
[34195000][SCB][PASS] TX data matched! Sent: 0x7f6a, Received: 0x7f6a
[34215000][SEQ][REQ=3] Pushed TX data 0x1639 to queue.
[34215000][SEQ][REQ=3] Pushed RX data 0x1c8f to queue.
[46145000][SCB][PASS] TX data matched! Sent: 0x1639, Received: 0x1639
[46145000][SCB][PASS] RX data matched! Sent: 0x1c8f, Received: 0x1c8f
[46165000][SEQ][REQ=3] Pushed TX data 0x1141 to queue.
[46165000][SEQ][REQ=3] Pushed RX data 0x00b4 to queue.
[56015000][SCB][PASS] TX data matched! Sent: 0x1141, Received: 0x1141
[56015000][SCB][PASS] RX data matched! Sent: 0x00b4, Received: 0x00b4
[59335000][SEQ][REQ=3] Pushed TX data 0x2396 to queue.
[59335000][SEQ][REQ=3] Pushed RX data 0x70d8 to queue.
[68865000][SCB][PASS] TX data matched! Sent: 0x2396, Received: 0x2396
[68865000][SCB][PASS] RX data matched! Sent: 0x70d8, Received: 0x70d8
[68885000][SEQ][REQ=2] Pushed RX data 0x52e2 to queue.
[77855000][SCB][PASS] RX data matched! Sent: 0x52e2, Received: 0x52e2
[78975000][SEQ][REQ=3] Pushed TX data 0x4c83 to queue.
[78975000][SEQ][REQ=3] Pushed RX data 0x1704 to queue.
[90705000][SCB][PASS] TX data matched! Sent: 0x4c83, Received: 0x4c83
[90705000][SCB][PASS] RX data matched! Sent: 0x1704, Received: 0x1704
[90725000][SEQ][REQ=3] Pushed TX data 0x47f8 to queue.
[90725000][SEQ][REQ=3] Pushed RX data 0x5a10 to queue.
[101975000][SCB][PASS] TX data matched! Sent: 0x47f8, Received: 0x47f8
[101975000][SCB][PASS] RX data matched! Sent: 0x5a10, Received: 0x5a10
[101995000][SEQ][REQ=2] Pushed RX data 0x3fda to queue.
[110965000][SCB][PASS] RX data matched! Sent: 0x3fda, Received: 0x3fda
[110985000][SEQ][REQ=3] Pushed TX data 0x678d to queue.
```

Randomised Sequence Waveform



- Randomised request
- Randomised din_master
  din_slave inputs
- Randomised wait_duration

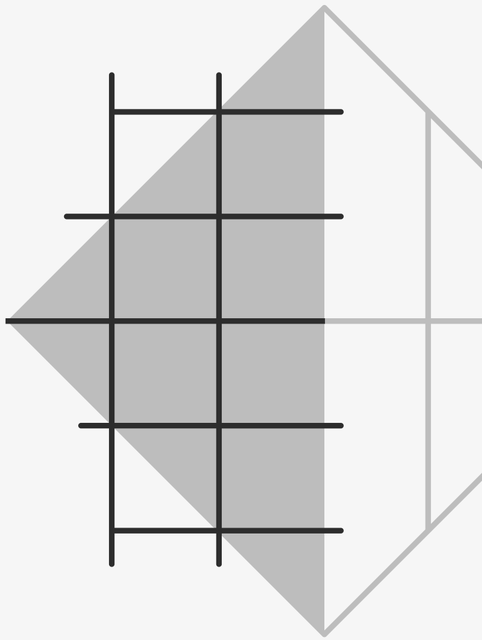
Scoreboard

```
 91    class spi_scoreboard;
 92        logic [(`SPI_TRF_BIT-1):0] tx_data_q [$];
 93        logic [(`SPI_TRF_BIT-1):0] rx_data_q [$];
 94
 95        int total_checks;
 96        int pass_count;
 97        int fail_count;
 98
 99        function new();
100            total_checks    = 0;
101            pass_count      = 0;
102            fail_count      = 0;
103            cg              = new();
104        endfunction
```

```
193          function summary();
194              $display("\n==================== SCOREBOARD SUMMARY ====================");
195              $display("Total Checks: %0d", total_checks);
196              $display("Pass Count  : %0d", pass_count);
197              $display("Fail Count  : %0d", fail_count);
198              $display("============================================================\n");
199          endfunction
```

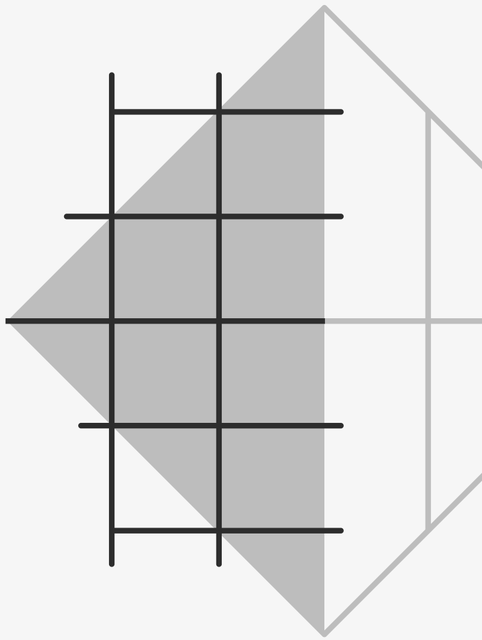
Scoreboard

```
147         function void push_tx_data(int req, logic [(`SPI_TRF_BIT-1):0] data);
148             tx_data_q.push_back(data);
149             $display("[%0t][SEQ][REQ=%0d] Pushed TX data 0x%h to queue.", $time, req, data);
150         endfunction
151 
152         function void push_rx_data(int req, logic [(`SPI_TRF_BIT-1):0] data);
153             rx_data_q.push_back(data);
154             $display("[%0t][SEQ][REQ=%0d] Pushed RX data 0x%h to queue.", $time, req, data);
155         endfunction
156 
157         function check_tx_data();
158             logic [(`SPI_TRF_BIT-1):0] expected_data, actual_data;
159             if (tx_data_q.size() > 0) begin
160                 expected_data = tx_data_q.pop_front();
161                 actual_data   = dout_slave;
162                 total_checks++;
163                 if (actual_data === expected_data) begin
164                     pass_count++;
165                     $display("[%0t][SCB][PASS] TX data matched! Sent: 0x%h, Received: 0x%h",
166                              $time, expected_data, actual_data);
167                 end else begin
168                     fail_count++;
169                     $error("[%0t][SCB][FAIL] TX data mismatch! Sent: 0x%h, Received: 0x%h",
170                            $time, expected_data, actual_data);
171                 end
172             end
173         endfunction
174 
175         function check_rx_data();
176             logic [(`SPI_TRF_BIT-1):0] expected_data, actual_data;
177             if (rx_data_q.size() > 0) begin
178                 expected_data = rx_data_q.pop_front();
179                 actual_data   = dout_master;
180                 total_checks++;
181                 if (actual_data === expected_data) begin
182                     pass_count++;
183                     $display("[%0t][SCB][PASS] RX data matched! Sent: 0x%h, Received: 0x%h",
184                              $time, expected_data, actual_data);
185                 end else begin
186                     fail_count++;
187                     $error("[%0t][SCB][FAIL] RX data mismatch! Sent: 0x%h, Received: 0x%h",
188                            $time, expected_data, actual_data);
189                 end
190             end
191         endfunction
```

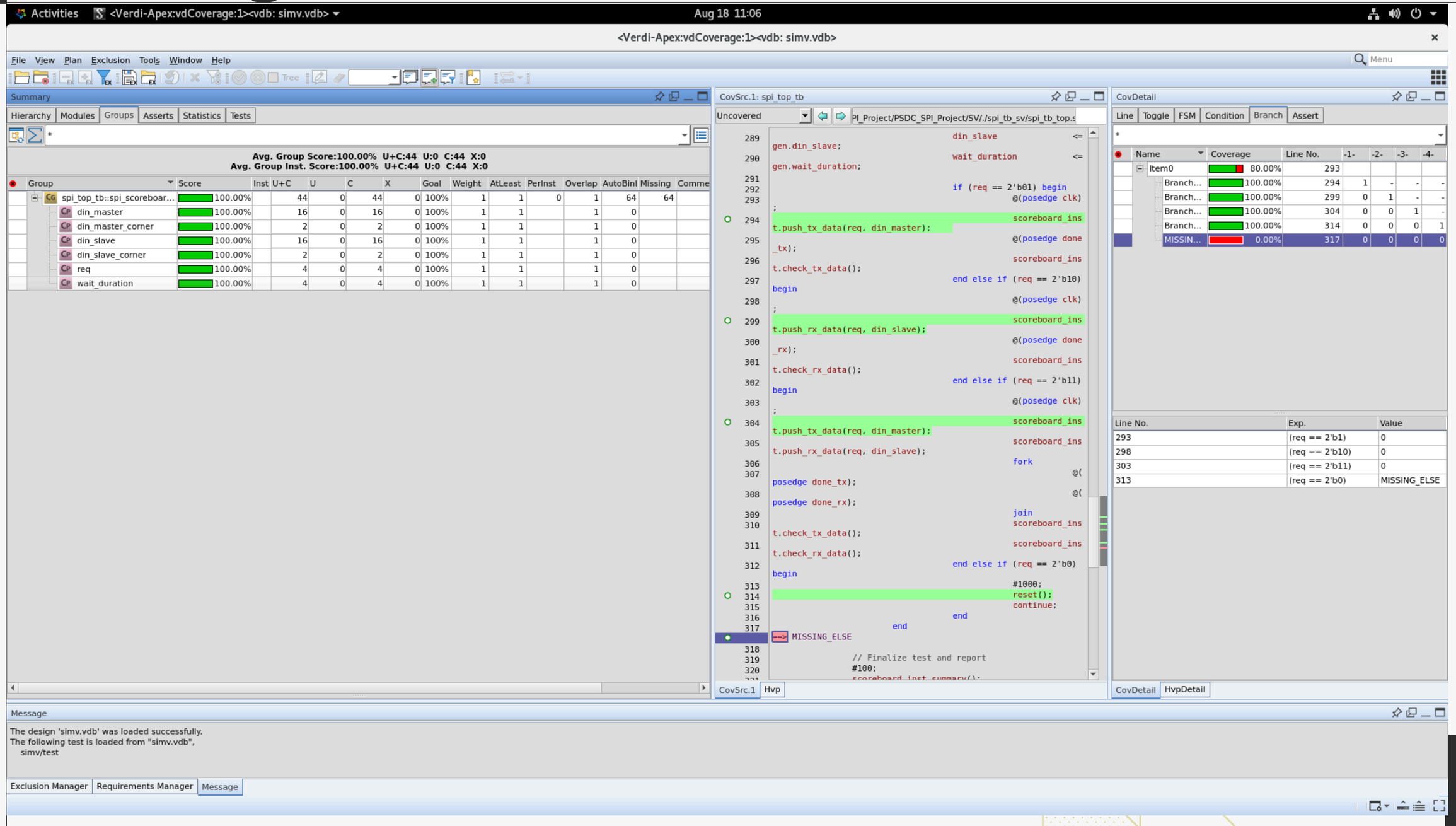


Coverage

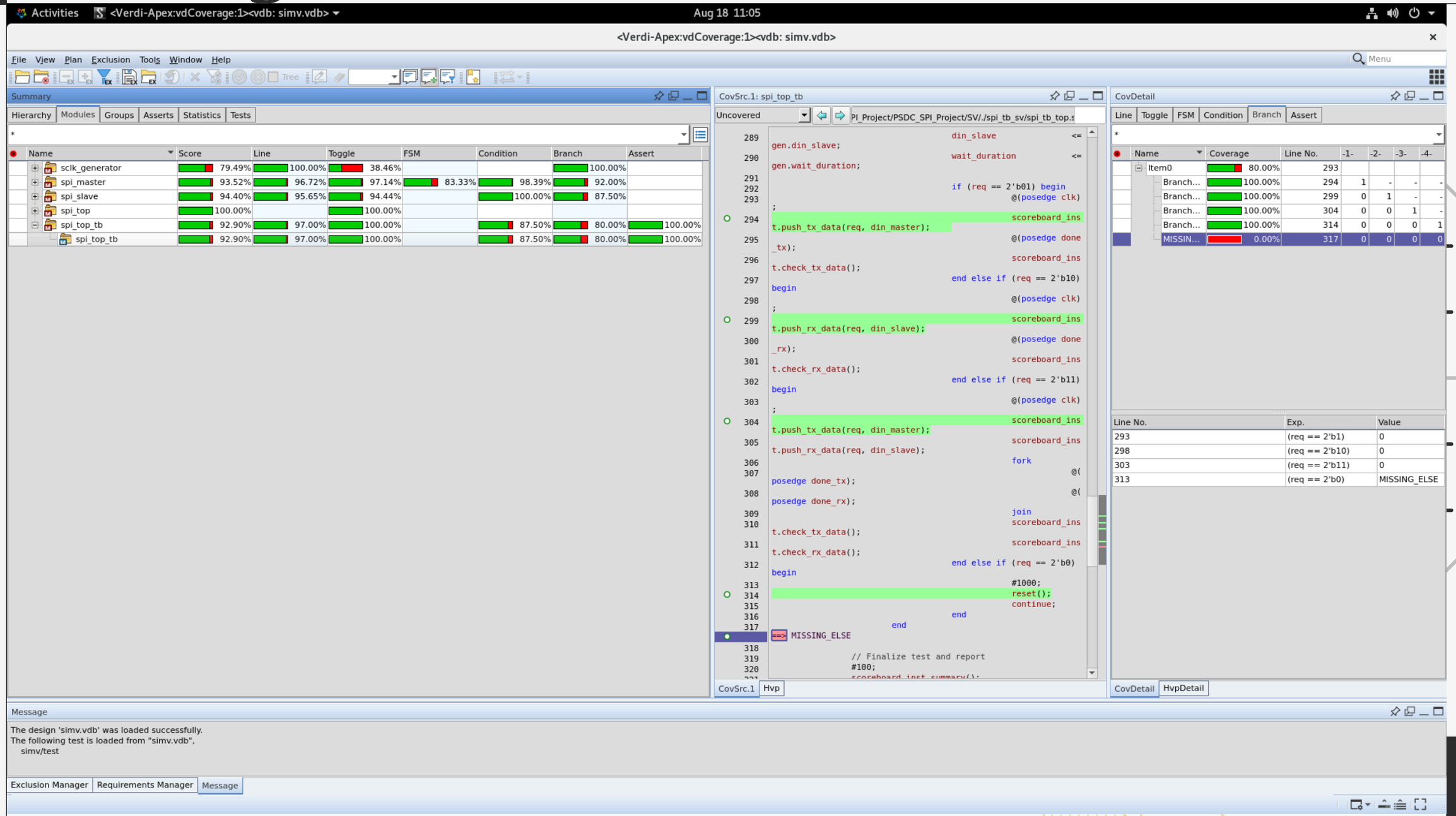
```
106         covergroup cg @(posedge clk);
107
108             coverpoint req {
109                 bins no_operation = {2'b0};
110                 bins tx_only      = {2'b01};
111                 bins rx_only      = {2'b10};
112                 bins full_duplex  = {2'b11};
113             }
114
115             coverpoint wait_duration {
116                 bins zero   = {0};
117                 bins short  = {[1:5]};
118                 bins med    = {[6:15]};
119                 bins long   = {[16:255]};
120             }
121
122             // Full range of values
123             coverpoint din_master {
124                 bins all_values[] = {[0 : (1<<`SPI_TRF_BIT)-1]};
125             }
126
127             coverpoint din_slave {
128                 bins all_values[] = {[0 : (1<<`SPI_TRF_BIT)-1]};
129             }
130
131             // Corner cases for din_master
132             din_master_corner : coverpoint din_master {
133                 bins zero     = {0};
134                 bins all_ones = { (1<<`SPI_TRF_BIT) - 1 };
135             }
136
137             // Corner cases for din_slave
138             din_slave_corner : coverpoint din_slave {
139                 bins zero     = {0};
140                 bins all_ones = { (1<<`SPI_TRF_BIT) - 1 };
141             }
142
143         endgroup
```



Coverage


Activities
<Verdi-Apex:vdCoverage:1><vdb: simv.vdb>
Aug 18 11:06
<Verdi-Apex:vdCoverage:1><vdb: simv.vdb>
File View Plan Exclusion Tools Window Help
Menu
Summary
Hierarchy Modules Groups Asserts Statistics Tests
Avg. Group Score:100.00% U+C:44 U:0 C:44 X:0
Avg. Group Inst. Score:100.00% U+C:44 U:0 C:44 X:0
Group Score Inst U+C U C X Goal Weight AtLeast PerInst Overlap AutoBinI Missing Comme
spi_top_tb::spi_scoreboar... 100.00% 44 0 44 0 100% 1 1 0 1 64 64
din_master 100.00% 16 0 16 0 100% 1 1 1 0
din_master_corner 100.00% 2 0 2 0 100% 1 1 1 0
din_slave 100.00% 16 0 16 0 100% 1 1 1 0
din_slave_corner 100.00% 2 0 2 0 100% 1 1 1 0
req 100.00% 4 0 4 0 100% 1 1 1 0
wait_duration 100.00% 4 0 4 0 100% 1 1 1 0
CovSrc.1: spi_top_tb
Uncovered
PI_Project/PSDC_SPI_Project/SV/./spi_tb_sv/spi_tb_top.s
289 din_slave <= gen.din_slave;
290 wait_duration <= gen.wait_duration;
291
292 if (req == 2'b01) begin
293 @(posedge clk);
294 scoreboard_inst.push_tx_data(req, din_master);
295 @(posedge done_tx);
296 scoreboard_inst.check_tx_data();
297 end else if (req == 2'b10) begin
298 @(posedge clk);
299 scoreboard_inst.push_rx_data(req, din_slave);
300 @(posedge done_rx);
301 scoreboard_inst.check_rx_data();
302 end else if (req == 2'b11) begin
303 @(posedge clk);
304 scoreboard_inst.push_tx_data(req, din_master);
305 scoreboard_inst.push_rx_data(req, din_slave);
306 fork
307 @(posedge done_tx);
308 @(posedge done_rx);
309 join
310 scoreboard_inst.check_tx_data();
311 scoreboard_inst.check_rx_data();
312 end else if (req == 2'b0) begin
313 #1000;
314 reset();
315 continue;
316 end
317 end
==> MISSING_ELSE
318
319 // Finalize test and report
320 #100;
CovSrc.1 Hvp
CovDetail
Line Toggle FSM Condition Branch Assert
Name Coverage Line No. -1- -2- -3- -4-
Item0 80.00% 293
Branch... 100.00% 294 1 - - -
Branch... 100.00% 299 0 1 - -
Branch... 100.00% 304 0 0 1 -
Branch... 100.00% 314 0 0 0 1
MISSIN... 0.00% 317 0 0 0 0
Line No. Exp. Value
293 (req == 2'b1) 0
298 (req == 2'b10) 0
303 (req == 2'b11) 0
313 (req == 2'b0) MISSING_ELSE
CovDetail HvpDetail
Message
The design 'simv.vdb' was loaded successfully.
The following test is loaded from "simv.vdb",
simv/test
Exclusion Manager Requirements Manager Message

Coverage


Activities
<Verdi-Apex:vdCoverage:1><vdb: simv.vdb>
Aug 18 11:05
<Verdi-Apex:vdCoverage:1><vdb: simv.vdb>
File View Plan Exclusion Tools Window Help
Menu
Summary
Hierarchy Modules Groups Asserts Statistics Tests
Name Score Line Toggle FSM Condition Branch Assert
sclk_generator 79.49% 100.00% 38.46% 100.00%
spi_master 93.52% 96.72% 97.14% 83.33% 98.39% 92.00%
spi_slave 94.40% 95.65% 94.44% 100.00% 87.50%
spi_top 100.00% 100.00%
spi_top_tb 92.90% 97.00% 100.00% 87.50% 80.00% 100.00%
spi_top_tb 92.90% 97.00% 100.00% 87.50% 80.00% 100.00%
CovSrc.1: spi_top_tb
Uncovered
PI_Project/PSDC_SPI_Project/SV/./spi_tb_sv/spi_tb_top.s
289 din_slave <= gen.din_slave;
290 wait_duration <= gen.wait_duration;
291
292 if (req == 2'b01) begin
293 @(posedge clk);
294 scoreboard_inst.push_tx_data(req, din_master);
295 @(posedge done_tx);
296 scoreboard_inst.check_tx_data();
297 end else if (req == 2'b10) begin
298 @(posedge clk);
299 scoreboard_inst.push_rx_data(req, din_slave);
300 @(posedge done_rx);
301 scoreboard_inst.check_rx_data();
302 end else if (req == 2'b11) begin
303 @(posedge clk);
304 scoreboard_inst.push_tx_data(req, din_master);
305 scoreboard_inst.push_rx_data(req, din_slave);
306 fork
307 @(posedge done_tx);
308 @(posedge done_rx);
309 join
310 scoreboard_inst.check_tx_data();
311 scoreboard_inst.check_rx_data();
312 end else if (req == 2'b0) begin
313 #1000;
314 reset();
315 continue;
316 end
317 end
==> MISSING_ELSE
318
319 // Finalize test and report
320 #100;
CovSrc.1 Hvp
CovDetail
Line Toggle FSM Condition Branch Assert
Name Coverage Line No. -1- -2- -3- -4-
Item0 80.00% 293
Branch... 100.00% 294 1 - - -
Branch... 100.00% 299 0 1 - -
Branch... 100.00% 304 0 0 1 -
Branch... 100.00% 314 0 0 0 1
MISSIN... 0.00% 317 0 0 0 0
Line No. Exp. Value
293 (req == 2'b1) 0
298 (req == 2'b10) 0
303 (req == 2'b11) 0
313 (req == 2'b0) MISSING_ELSE
CovDetail HvpDetail
Message
The design 'simv.vdb' was loaded successfully.
The following test is loaded from "simv.vdb",
simv/test
Exclusion Manager Requirements Manager Message

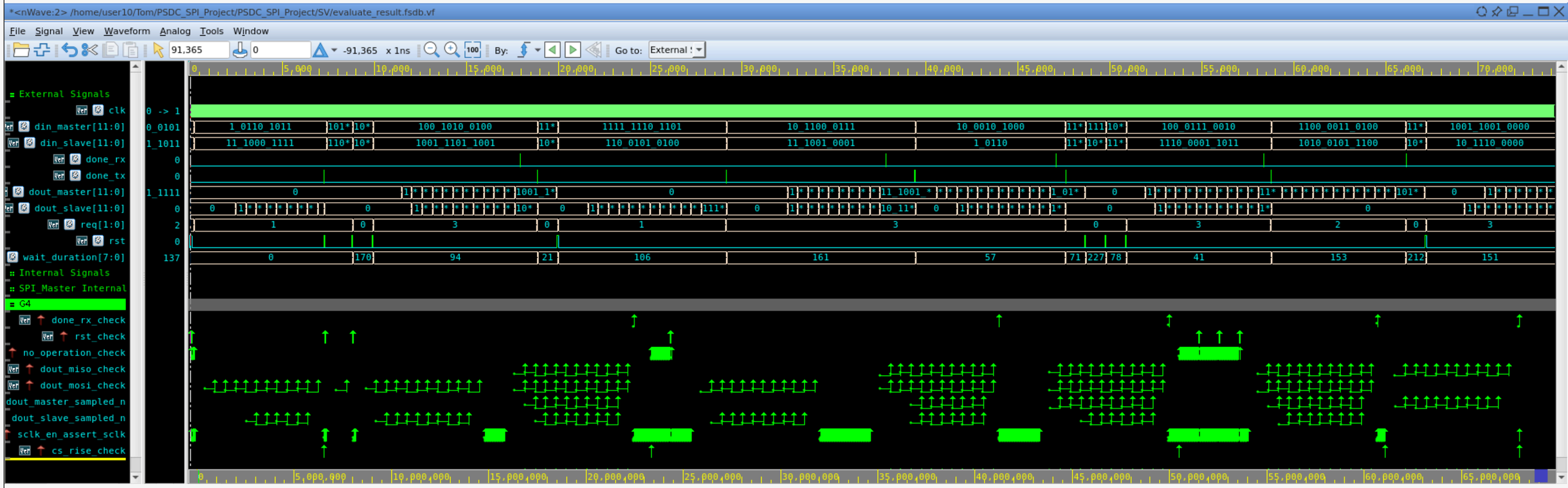
Assertion

```
334     done_rx_check: assert property (@(posedge clk)
335     disable iff (rst || ($past(req) == 2'b01 || $past(req) == 2'b00))
336         ($rose (dut.spi_master_inst.sclk_negedge) && (din_slave === dout_master)) |=> done_rx
337     ) else $error("%t [FAIL][Done RX] RX did not aserrt when din_slave=%h dout_master=%h",
338               $time, din_slave, dout_master);
339
340     rst_check: assert property (@(posedge clk)
341         (rst) |->
342         (dout_master == '0) && (dout_slave == '0) && (done_tx == 1'b0) && (done_rx == 1'b0)
343     ) else $error("%t [FAIL][rst_check] rst=%b", $time, rst);
344
345     no_operation_check: assert property (@(posedge clk)
346     disable iff (rst)
347         (req == 2'b00) |-> ##2 ($stable(dout_master)) && ($stable(dout_slave)) && (done_tx == 1'b0) && (done_rx == 1'b0)
348     ) else $error("%t [FAIL][no_operation_check] dout_master=0x%h, dout_slave=0x%h, done_tx=%b, done_rx=%b", $time, dout_master, dout_slave, done_tx, done_rx);
349
350     dout_miso_check: assert property (@(negedge sclk)
351     disable iff (rst || (din_slave == dout_master))
352         (req == 2'b10 || req == 2'b11) |=> ($past(miso) == dout_master[0])
353     ) else $error("%t [FAIL][dout_miso_check] dout_master LSB != miso in RX/Full-Duplex mode", $time);
354
355     dout_mosi_check: assert property (@(negedge sclk)
356     disable iff (rst || (din_master == dout_slave))
357         (req == 2'b01 || req == 2'b11) |=> ($past(mosi) == dout_slave[0])
358     ) else $error("%t [FAIL][dout_mosi_check] dout_slave LSB != mosi in TX/Full-Duplex mode", $time);
```

Assertion

```
360     dout_master_sampled_n: assert property (@(negedge dut.spi_master_inst.sclk_negedge)
361     disable iff (rst || (din_slave == dout_master))
362         ((req == 2 || req ==3 ) && !cs && dout_master !== 0) |=> (dout_master !== $past(dout_master))
363     ) else $error("dout_master did not change at negedge sclk: Previous = %b, Current = %b", $past(dout_master), dout_master);
364
365     dout_slave_sampled_n: assert property (@(negedge dut.spi_master_inst.sclk_negedge)
366     disable iff (rst || (din_master == dout_slave))
367         ((req == 1 || req == 3) && !cs && dout_slave !== 0) |=> (dout_slave !== $past(dout_slave))
368     ) else $error("dout_slave did not change at negedge sclk: Previous = %b, Current = %b", $past(dout_slave), dout_slave);
369
370     // SCLK enable check on posedge main clock
371     sclk_en_assert_sclk: assert property (@(posedge clk)
372     disable iff (rst)
373         !sclk_en |-> ##2 $stable(sclk)
374     ) else $error("sclk does not toggle when not en");
375
376     // Test 7.2 CS Assert after TX/RX transfer
377     cs_rise_check: assert property (@(posedge clk)
378     disable iff (rst || req == 2'b11)
379         (done_tx || done_rx) |-> cs
380     ) else $error("%t [FAIL][CS Timing] CS did not rise with done_tx(%b) or done_rx(%b)", $time, done_tx, done_rx);
```

Assertion


*<nWave:2> /home/user10/Tom/PSDC_SPI_Project/PSDC_SPI_Project/SV/evaluate_result.fsdb.vf
File Signal View Waveform Analog Tools Window
External Signals
clk
din_master[11:0]
din_slave[11:0]
done_rx
done_tx
dout_master[11:0]
dout_slave[11:0]
req[1:0]
rst
wait_duration[7:0]
Internal Signals
SPI_Master Internal
G4
done_rx_check
rst_check
no_operation_check
dout_miso_check
dout_mosi_check
dout_master_sampled_n
dout_slave_sampled_n
sclk_en_assert_sclk
cs_rise_check

Innovation

Dynamic Operating Bits

```
13      `ifndef SPI_TRF_BIT
14      `endif
15      localparam int DATA_WIDTH = `SPI_TRF_BIT;  // Use macro from Makefile
```

The use of macro from Makefile allows the number of operating bit to be dynamically applied outside of the testbench.

Makefile

```
build:
        vcs $(SRC) $(VCS_OPTS) $(FUNC_CVR) $(INCLUDE) $(TIMESCALE) \
            +define+SPI_TRF_BIT=$(BIT) | tee build.log
```

```
ROOT := $(shell pwd)
export ROOT
SEED = 5
Bit?=12
```

Innovation

Dynamic Operating Bits

```
logic [1:0] req;
logic [7:0] wait_duration;
logic [(`SPI_TRF_BIT-1):0] din_master;
logic [(`SPI_TRF_BIT-1):0] din_slave;

logic [(`SPI_TRF_BIT-1):0] dout_master;
logic [(`SPI_TRF_BIT-1):0] dout_slave;
logic done_tx;
logic done_rx;
```

```
class spi_scoreboard;
            logic [(`SPI_TRF_BIT-1):0] tx_data_q [$];
            logic [(`SPI_TRF_BIT-1):0] rx_data_q [$];
```

The implementation of macro from Makefile allows the number of operating bit to be dynamically applied outside of the testbench.

```
class tx_rx_rand;

            rand logic [(`SPI_TRF_BIT-1):0] din_master;
            rand logic [(`SPI_TRF_BIT-1):0] din_slave;
            rand logic [1:0] req;
            rand logic [7:0] wait_duration;
            constraint ran_range{
                    din_master              inside {[0:(2**`SPI_TRF_BIT)-1]};
            din_slave               inside {[0:(2**`SPI_TRF_BIT)-1]};
                    req                     inside {[0:3]};
                    wait_duration   inside {[0:256]};
                    }

endclass
tx_rx_rand gen;
```

Innovation

Dynamic Operating Bits

Output from the macro when the BIT was set after launching “make build run BIT=15” and “make build run BIT=8”

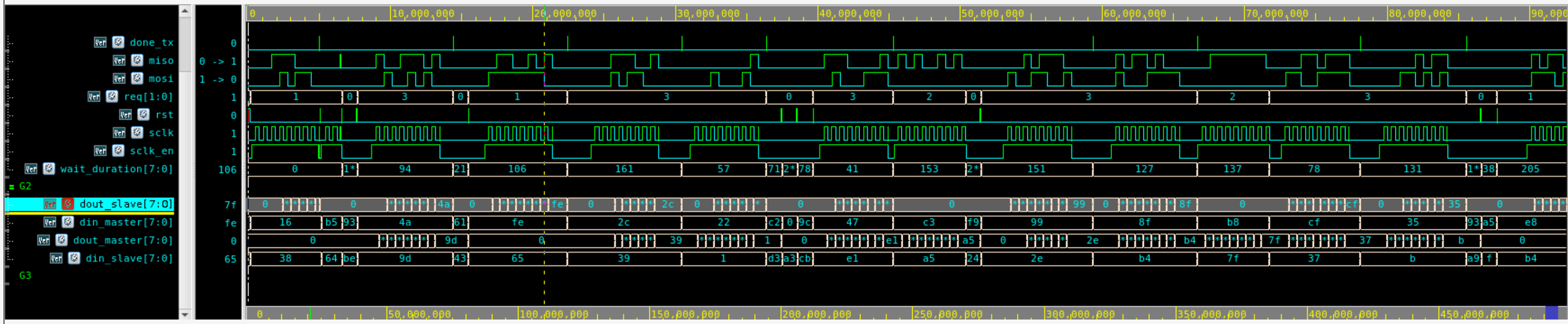
```
------------------------ 15 BIT RECEIVED --------------------


------------------------ FIXED SEQUENCE INPUT --------------------

Initial reset complete. Starting test sequences.
Initial reset complete. Starting test sequences.
TEST: TX Data MSB -> LSB (Test ID 2.1)
TEST: Reset on transfer

------------------------ RANDOMIZED INPUT --------------------

[11655000][SEQ][REQ=3] Pushed TX data 0x2524 to queue.
[11655000][SEQ][REQ=3] Pushed RX data 0x4ecb to queue.
[22245000][SCB][PASS] TX data matched! Sent: 0x2524, Received: 0x2524
[22245000][SCB][PASS] RX data matched! Sent: 0x4ecb, Received: 0x4ecb
[23365000][SEQ][REQ=1] Pushed TX data 0x7f6a to queue.
[34195000][SCB][PASS] TX data matched! Sent: 0x7f6a, Received: 0x7f6a
[34215000][SEQ][REQ=3] Pushed TX data 0x1639 to queue.
[34215000][SEQ][REQ=3] Pushed RX data 0x1c8f to queue.
[46145000][SCB][PASS] TX data matched! Sent: 0x1639, Received: 0x1639
[46145000][SCB][PASS] RX data matched! Sent: 0x1c8f, Received: 0x1c8f
[46165000][SEQ][REQ=3] Pushed TX data 0x1141 to queue.
[46165000][SEQ][REQ=3] Pushed RX data 0x00b4 to queue.
[56015000][SCB][PASS] TX data matched! Sent: 0x1141, Received: 0x1141
[56015000][SCB][PASS] RX data matched! Sent: 0x00b4, Received: 0x00b4
[59335000][SEQ][REQ=3] Pushed TX data 0x2396 to queue.
[59335000][SEQ][REQ=3] Pushed RX data 0x70d8 to queue.
[68865000][SCB][PASS] TX data matched! Sent: 0x2396, Received: 0x2396
[68865000][SCB][PASS] RX data matched! Sent: 0x70d8, Received: 0x70d8
[68885000][SEQ][REQ=2] Pushed RX data 0x52e2 to queue.
[77855000][SCB][PASS] RX data matched! Sent: 0x52e2, Received: 0x52e2
[78975000][SEQ][REQ=3] Pushed TX data 0x4c83 to queue.
[78975000][SEQ][REQ=3] Pushed RX data 0x1704 to queue.
[90705000][SCB][PASS] TX data matched! Sent: 0x4c83, Received: 0x4c83
[90705000][SCB][PASS] RX data matched! Sent: 0x1704, Received: 0x1704
[90725000][SEQ][REQ=3] Pushed TX data 0x47f8 to queue.
[90725000][SEQ][REQ=3] Pushed RX data 0x5a10 to queue.
[101975000][SCB][PASS] TX data matched! Sent: 0x47f8, Received: 0x47f8
[101975000][SCB][PASS] RX data matched! Sent: 0x5a10, Received: 0x5a10
[101995000][SEQ][REQ=2] Pushed RX data 0x3fda to queue.
[110965000][SCB][PASS] RX data matched! Sent: 0x3fda, Received: 0x3fda
[110985000][SEQ][REQ=3] Pushed TX data 0x678d to queue.
```

```
------------------------ 8 BIT RECEIVED --------------------


------------------------ FIXED SEQUENCE INPUT --------------------

Initial reset complete. Starting test sequences.
Initial reset complete. Starting test sequences.
TEST: TX Data MSB -> LSB (Test ID 2.1)
TEST: Reset on transfer

------------------------ RANDOMIZED INPUT --------------------

[7735000][SEQ][REQ=3] Pushed TX data 0x4a to queue.
[7735000][SEQ][REQ=3] Pushed RX data 0x9d to queue.
[14405000][SCB][PASS] TX data matched! Sent: 0x4a, Received: 0x4a
[14405000][SCB][PASS] RX data matched! Sent: 0x9d, Received: 0x9d
[15525000][SEQ][REQ=1] Pushed TX data 0xfe to queue.
[22435000][SCB][PASS] TX data matched! Sent: 0xfe, Received: 0xfe
[22455000][SEQ][REQ=3] Pushed TX data 0x2c to queue.
[22455000][SEQ][REQ=3] Pushed RX data 0x39 to queue.
[30465000][SCB][PASS] TX data matched! Sent: 0x2c, Received: 0x2c
[30465000][SCB][PASS] RX data matched! Sent: 0x39, Received: 0x39
[30485000][SEQ][REQ=3] Pushed TX data 0x22 to queue.
[30485000][SEQ][REQ=3] Pushed RX data 0x01 to queue.
[36415000][SCB][PASS] TX data matched! Sent: 0x22, Received: 0x22
[36415000][SCB][PASS] RX data matched! Sent: 0x01, Received: 0x01
[39735000][SEQ][REQ=3] Pushed TX data 0x47 to queue.
[39735000][SEQ][REQ=3] Pushed RX data 0xe1 to queue.
[45345000][SCB][PASS] TX data matched! Sent: 0x47, Received: 0x47
[45345000][SCB][PASS] RX data matched! Sent: 0xe1, Received: 0xe1
[45365000][SEQ][REQ=2] Pushed RX data 0xa5 to queue.
[50415000][SCB][PASS] RX data matched! Sent: 0xa5, Received: 0xa5
[51535000][SEQ][REQ=3] Pushed TX data 0x99 to queue.
[51535000][SEQ][REQ=3] Pushed RX data 0x2e to queue.
[59345000][SCB][PASS] TX data matched! Sent: 0x99, Received: 0x99
[59345000][SCB][PASS] RX data matched! Sent: 0x2e, Received: 0x2e
[59365000][SEQ][REQ=3] Pushed TX data 0x8f to queue.
[59365000][SEQ][REQ=3] Pushed RX data 0xb4 to queue.
```

Innovation

Dynamic Operating Bits

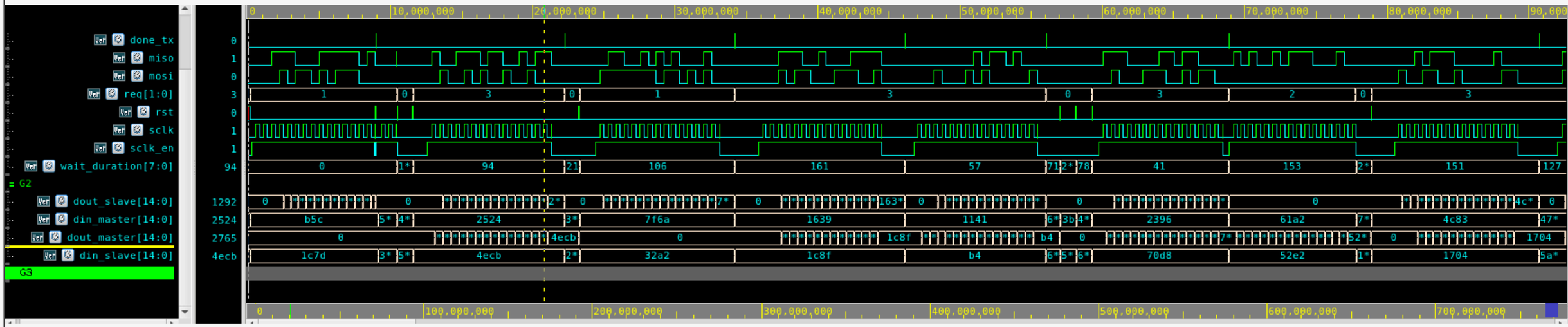
8-bits waveform



Innovation

Dynamic Operating Bits

15-bits waveform



Limitations

Incompatible Assertion

```
"./spi_tb_sv/spi_tb_top.sv", 331: spi_top_tb.done_rx_check: started at 313085000ps failed at 313095000ps
        Offending 'done_rx'
Error: "./spi_tb_sv/spi_tb_top.sv", 331: spi_top_tb.done_rx_check: at time 313095000 ps
           313095000 [FAIL][Done RX] RX did not aserrt when din_slave=0 dout_master=0
"./spi_tb_sv/spi_tb_top.sv", 331: spi_top_tb.done_rx_check: started at 313645000ps failed at 313655000ps
        Offending 'done_rx'
Error: "./spi_tb_sv/spi_tb_top.sv", 331: spi_top_tb.done_rx_check: at time 313655000 ps
           313655000 [FAIL][Done RX] RX did not aserrt when din_slave=0 dout_master=0
"./spi_tb_sv/spi_tb_top.sv", 331: spi_top_tb.done_rx_check: started at 314205000ps failed at 314215000ps
        Offending 'done_rx'
Error: "./spi_tb_sv/spi_tb_top.sv", 331: spi_top_tb.done_rx_check: at time 314215000 ps
           314215000 [FAIL][Done RX] RX did not aserrt when din_slave=0 dout_master=0
"./spi_tb_sv/spi_tb_top.sv", 331: spi_top_tb.done_rx_check: started at 314765000ps failed at 314775000ps
        Offending 'done_rx'
Error: "./spi_tb_sv/spi_tb_top.sv", 331: spi_top_tb.done_rx_check: at time 314775000 ps
           314775000 [FAIL][Done RX] RX did not aserrt when din_slave=0 dout_master=0
[316555000][SCB][PASS] TX data matched! Sent: 0xd, Received: 0xd
[316555000][SCB][PASS] RX data matched! Sent: 0x0, Received: 0x0
[317675000][SEQ][REQ=2] Pushed RX data 0x3 to queue.
[320485000][SCB][PASS] RX data matched! Sent: 0x3, Received: 0x3
[321605000][SEQ][REQ=1] Pushed TX data 0x8 to queue.
[328675000][SCB][PASS] TX data matched! Sent: 0x8, Received: 0x8
[328695000][SEQ][REQ=1] Pushed TX data 0x0 to queue.
[332645000][SCB][PASS] TX data matched! Sent: 0x0, Received: 0x0
```

Lower bit count has higher chance of having the input = output during mid transfer, which will falsely trigger the done flag assertion

Limitations

Pending Assertions

```
382 duplex_cs_rise_check: assert property (@(posedge clk)
383     disable iff (rst || (req == 2'b01 || req == 2'b10))
384     tx_rx_done |-> cs
385 ) else $error("%t [FAIL][CS Timing] CS did not rise after both done_tx(%b) and done_rx(%b)",
386                 $time, done_tx, done_rx);
387
388 int counter;
389 initial begin
390     counter = 0;
391     forever begin
392         @(negedge sclk)
393         $display("At negedge sclk: counter=%0d", counter);
394         if (counter >= 10) begin
395             counter = 0;
396         end else if (din_master === dout_slave) begin
397             for (counter = 1; counter <= wait_duration; counter++) begin
398                 @(posedge clk);
399                 $display("counter=%0d", counter);
400             end
401         end
402     end
403 end
404
405 done_tx_check: assert property (@(posedge clk)
406     disable iff (rst || ($past(req) == 2'b10 || $past(req) == 2'b00))
407     ($rose(dut.spi_master_inst.sclk_negedge) && (counter == wait_duration)) |=> done_rx
408 ) else $error("%t [FAIL][Done TX] TX did not assert when din_slave=%h dout_master=%h",
409             $time, din_slave, dout_master);
```

Unfinished assertions due to wait_duration being a variable rather than a fixed parameter

Thank you for listening

For Presentation

All-Purpose

2022 May 15

Full Name