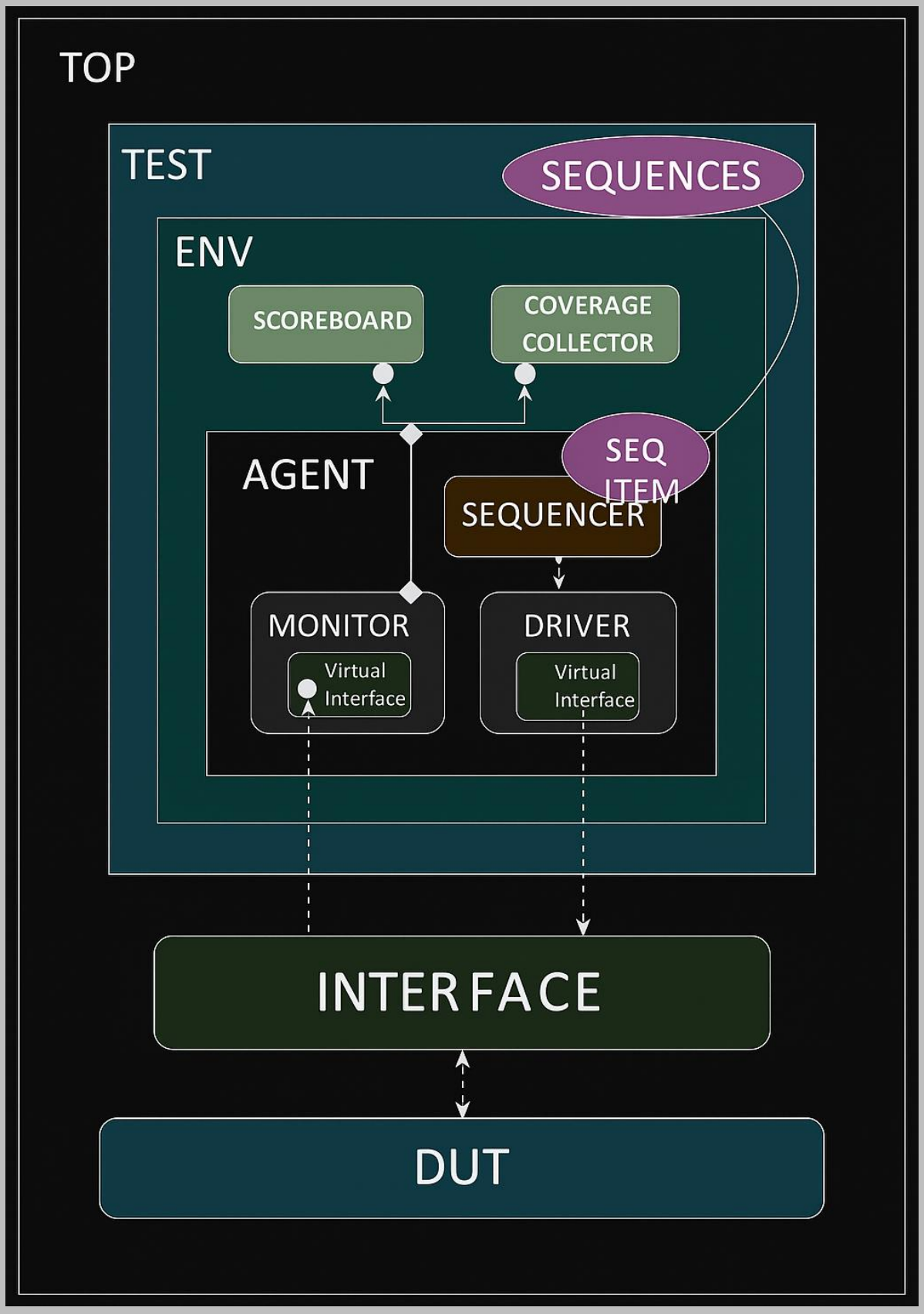
SPI Slave Verification

UVM Project

Muhammad Naim

UVM Structure



- ***TOP:*** *instantiates the DUT and the UVM via uvm_config_db and factory. contains the physical interface instances that connect to DUT.*
- ***uvm_test****: top-level test class that configures the environment, sets up sequences (or virtual sequences), and raises/drops objections to control simulation end.*
- ***uvm_env****: container for verification components (agents, scoreboard, coverage collectors, reference models, utilities). built during the build phase.*
- ***Agent****: contains a sequencer, driver, and monitor. Agents can be active (drive DUT) or passive (monitor only).*
- ***Sequencer****: receives sequence items from sequences and arbitrate sequences for its driver.*
- ***Driver****: converts seq_item into pin-level or TLM stimulus and drives DUT via virtual interface.*
- ***Monitor****: samples DUT signals and converts them into transactions, forwards to analysis ports.*
- ***Scoreboard****: subscribes to analysis ports from monitors and reference model(s) and performs checking (self-checking).*
- ***coverage collectors****: functional coverage and covergroups.*
- ***config_db / uvm_resource****: used to pass virtual interfaces, config parameters.*
- ***Factory****: enables test-time substitution of components and objects for flexibility.*

Design

```
module SLAVE (MOSI,MISO,SS_n,clk,rst_n,rx_data,rx_valid,tx_data,tx_valid);

    localparam IDLE      = 3'b000;
    localparam WRITE     = 3'b001;
    localparam CHK_CMD   = 3'b010;
    localparam READ_ADD  = 3'b011;
    localparam READ_DATA = 3'b100;

    input            MOSI, clk, rst_n, SS_n, tx_valid;
    input      [7:0] tx_data;
    output reg [9:0] rx_data;
    output reg       rx_valid, MISO;

    reg [3:0] counter;
    reg       received_address;

    reg [2:0] cs, ns;

    always @(posedge clk) begin
        if (~rst_n) begin
            cs <= IDLE;
        end
        else begin
            cs <= ns;
        end
    end

    always @(*) begin
        case (cs)
            IDLE : begin
                if (SS_n)
                    ns = IDLE;
                else
                    ns = CHK_CMD;
            end
            CHK_CMD : begin
                if (SS_n)
                    ns = IDLE;
                else begin
                    if (~MOSI)
                        ns = WRITE;
                    else begin
                        if (~received_address)//////not
                            ns = READ_ADD;
                        else
                            ns = READ_DATA;
                    end
                end
            end
            WRITE : begin
                if (SS_n)
                    ns = IDLE;
                else
                    ns = WRITE;
            end
            READ_ADD : begin
                if (SS_n)
                    ns = IDLE;
                else
                    ns = READ_ADD;
            end
            READ_DATA : begin
                if (SS_n)
                    ns = IDLE;
                else
                    ns = READ_DATA;
            end
        endcase
    end

```

```
always @(posedge clk) begin
    if (~rst_n) begin
        rx_data <= 0;
        rx_valid <= 0;
        received_address <= 0;
        MISO <= 0;
        counter <= 0; /////////rest
    end
    else begin
        case (cs)
            IDLE : begin
                rx_valid <= 0;
            end
            CHK_CMD : begin
                //////// read data counter = 8
                if (!SS_n && received_address) counter <= 8;
                else counter <= 10;
            end
            WRITE : begin
                if (counter > 0) begin
                    rx_data[counter-1] <= MOSI;
                    counter <= counter - 1;
                end
                else begin
                    rx_valid <= 1;
                end
            end
            READ_ADD : begin
                if (counter > 0) begin
                    rx_data[counter-1] <= MOSI;
                    counter <= counter - 1;
                end
                else begin
                    rx_valid <= 1;
                    received_address <= 1;
                end
            end
            READ_DATA : begin
                if (tx_valid) begin
                    rx_valid <= 0;
                    if (counter > 0) begin
                        MISO <= tx_data[counter-1];
                        counter <= counter - 1;
                    end
                    else begin
                        received_address <= 0;
                    end
                end
                else begin
                    if (counter > 0) begin
                        rx_data[counter-1] <= MOSI;
                        counter <= counter - 1;
                    end
                    else begin
                        rx_valid <= 1;
                        counter  <= 10; //////10
                    end
                end
            end
        endcase
    end
end

```

Design Assertions

```
`ifdef SIM


property idle_chk_p;
    @(posedge clk) disable iff (!rst_n) (cs == IDLE) |=> cs == CHK_CMD [-> 1]
endproperty
idle_chk_a: assert property (idle_chk_p);
idle_chk_c: cover  property (idle_chk_p);

property chk_wr_p;
    @(posedge clk) disable iff (!rst_n) (cs == CHK_CMD) |=> cs == (WRITE || READ_ADD || READ_DATA) [-> 1]
endproperty
chk_wr_a: assert property (chk_wr_p);
chk_wr_c: cover  property (chk_wr_p);

property wr_idle_p;
    @(posedge clk) disable iff (!rst_n) (cs == WRITE) |=> cs == (IDLE) [-> 1]
endproperty
wr_idle_a: assert property (wr_idle_p);
wr_idle_c: cover  property (wr_idle_p);

property rd_add_idle_p;
    @(posedge clk) disable iff (!rst_n) (cs == READ_ADD) |=> cs == (IDLE) [-> 1]
endproperty
rd_add_idle_a: assert property (rd_add_idle_p);
rd_add_idle_c: cover  property (rd_add_idle_p);

property rd_data_idle_p;
    @(posedge clk) disable iff (!rst_n) (cs == READ_DATA) |=> cs == (IDLE) [-> 1]
endproperty
rd_data_idle_a: assert property (rd_data_idle_p);
rd_data_idle_c: cover  property (rd_data_idle_p);


`endif

endmodule
```

Bugs:

1. *Counter is not in reset.*
2. *Not “received_address” instead of “received_address” in read_data condition in chk_cmd state.*
3. *Read_data counter = 8 not 10 in the chk_cmd state.*
4. *Counter equals 10 not 8 in the read_data state.*

Assertions

Feature	Assertion
Whenever reset is asserted, MISO is low	@(posedge clk) !rst_n |=> MISO == 0
Whenever reset is asserted, rx_valid is low	@(posedge clk) !rst_n |=> rx_valid == 0
Whenever reset is asserted, rx_data is low	@(posedge clk) !rst_n |=> rx_data == 0

UVM Code

```
interface slave_if(clk);
    input clk;
    logic MOSI, rst_n, SS_n, tx_valid;
    logic [7:0] tx_data;
    logic [9:0] rx_data;
    logic       rx_valid, MISO;
    logic [9:0] rx_data_golden;
    logic       rx_valid_golden, MISO_golden;
endinterface

```

```
module sva(
    input clk,
    logic MOSI, rst_n, SS_n, tx_valid,
    logic [7:0] tx_data,
    logic [9:0] rx_data,
    logic       rx_valid, MISO
);

// reset
miso_a    : assert property (miso_p);
miso_c    : cover  property (miso_p);
rx_valid_a: assert property(rx_valid_p);
rx_valid_c: cover  property(rx_valid_p);
rx_data_a : assert property(rx_data_p);
rx_data_c : cover  property(rx_data_p);

property miso_p;
    @(posedge clk) !rst_n |=> MISO == 0
endproperty
property rx_valid_p;
    @(posedge clk) !rst_n |=> rx_valid == 0
endproperty
property rx_data_p;
    @(posedge clk) !rst_n |=> rx_data == 0
endproperty

endmodule

```

```
package shared_pkg;
    int correct, error;
endpackage

```

```
package config_pkg;
    import uvm_pkg::*;
    `include "uvm_macros.svh"
    class slave_config extends uvm_object;
        `uvm_object_utils(slave_config)

        virtual slave_if slave_vif;

        function new (string name = "slave_config");
            super.new (name);
        endfunction
    endclass
endpackage

```

```
package seq_item_pkg;
import uvm_pkg::*;
import shared_pkg::*;
`include "uvm_macros.svh"
class slave_seq_item extends uvm_sequence_item;
`uvm_object_utils(slave_seq_item)
rand bit       rst_n, SS_n, MOSI, tx_valid;
rand bit [7:0] tx_data;
     bit [9:0] rx_data;
     bit       rx_valid, MISO;
// golden
     bit [9:0] rx_data_golden;
     bit       rx_valid_golden, MISO_golden;
// vars
rand   bit [10:0] mosi_arr; //   randomized array
static bit [10:0] mosi_bus; // unrandomized array
static bit [ 4:0] cycles;   // static counter
       bit [ 4:0] mosi_ptr;
// constructor
function new(string name = "slave_seq_item");
    super.new(name);
endfunction
// string
function string convert2string();
    return $sformatf("%s rst = 0b%0b, direction = 0b%0b", super.convert2string, rst_n, rx_data);
endfunction
function string convert2string_stimulus();
    return $sformatf("rst = 0b%0b, direction = 0b%0b", rst_n, rx_data);
endfunction

////////////////// constraints ////////////////////
constraint reset_c {rst_n dist{0:=1, 1:=99};}
constraint mosi_bits_c {
    if (~SS_n) (mosi_arr[10:8] inside {3'b000, 3'b001, 3'b110, 3'b111});
}
constraint ss_n_c {
    if (cycles == 0)
        SS_n == 1;
    else
        SS_n == 0;
}
function void post_randomize();
    if (SS_n) mosi_bus = mosi_arr;
    MOSI = mosi_bus[mosi_ptr];
    if (cycles > 0)
        cycles--;
    else begin
        if (mosi_bus[10:8] == 3'b111) cycles = 23;
        else                          cycles = 13;
    end
    if (mosi_ptr > 0) begin
        mosi_ptr--;
    end
    else begin
        mosi_ptr = 5'd11;
    end
    // tx_valid constraint
    if (mosi_bus[10:8] == 3'b111)
        tx_valid = 1;
    else
        tx_valid = 0;
endfunction
endclass
endpackage
```

```
package main_seq_pkg;
    import uvm_pkg::*;
    import seq_item_pkg::*;
    `include "uvm_macros.svh"

    class slave_main_seq extends uvm_sequence #(slave_seq_item);
        `uvm_object_utils(slave_main_seq)
        slave_seq_item seq_item;
        function new(string name = "slave_main_seq");
            super.new(name);
        endfunction

        task body;
            repeat(10000) begin
            seq_item = slave_seq_item::type_id::create("seq_item");
            start_item(seq_item);
            assert(seq_item.randomize());
            finish_item(seq_item);
            end
        endtask
    endclass
endpackage
```

```
package reset_seq_pkg;
    import uvm_pkg::*;
    import seq_item_pkg::*;
    import shared_pkg::*;
    `include "uvm_macros.svh"
    class slave_reset_seq extends uvm_sequence #(slave_seq_item);
        `uvm_object_utils(slave_reset_seq)
        slave_seq_item seq_item;
        function new(string name = "slave_reset_seq");
            super.new(name);
        endfunction

        task body;
            seq_item = slave_seq_item::type_id::create("seq_item");
            start_item(seq_item);
            seq_item.MOSI     = 0;
            seq_item.rst_n    = 0;
            seq_item.SS_n     = 0;
            seq_item.tx_valid = 0;
            seq_item.tx_data  = 0;
            finish_item(seq_item);
        endtask
    endclass
endpackage
```

```
package sequencer_pkg;
    import uvm_pkg::*;
    import seq_item_pkg::*;
    `include "uvm_macros.svh"
    class slave_sequencer extends uvm_sequencer #(slave_seq_item);
        `uvm_component_utils(slave_sequencer)
        function new(string name = "slave_seq_item", uvm_component parent = null);
            super.new(name, parent);
        endfunction
    endclass
endpackage

```

```
package driver_pkg;
    import uvm_pkg::*;
    import seq_item_pkg::*;
    `include "uvm_macros.svh"
    class slave_driver extends uvm_driver #(slave_seq_item);
        `uvm_component_utils(slave_driver)
        virtual slave_if slave_vif;
        slave_seq_item stim_seq_item;
        function new (string name = "slave_driver", uvm_component parent = null);
            super.new(name, parent);
        endfunction

        task run_phase (uvm_phase phase);
            super.run_phase (phase);
            forever begin
            stim_seq_item = slave_seq_item::type_id::create("stim_seq_item");
            seq_item_port.get_next_item(stim_seq_item);
            slave_vif.MOSI     = stim_seq_item.MOSI;
            slave_vif.rst_n    = stim_seq_item.rst_n;
            slave_vif.SS_n     = stim_seq_item.SS_n;
            slave_vif.tx_valid = stim_seq_item.tx_valid;
            slave_vif.tx_data  = stim_seq_item.tx_data;
            @(negedge slave_vif.clk);
            seq_item_port.item_done();
            `uvm_info("run_phase", stim_seq_item.convert2string_stimulus(), UVM_HIGH)
        end
    endtask
endclass
endpackage

```

```
package monitor_pkg;
import uvm_pkg::*;
import seq_item_pkg::*;
import shared_pkg::*;
`include "uvm_macros.svh"
class slave_monitor extends uvm_monitor;
    `uvm_component_utils(slave_monitor)
    virtual slave_if slave_vif;
    slave_seq_item rsp_seq_item;
    uvm_analysis_port #(slave_seq_item) mon_ap;

    function new(string name = "slave_monitor", uvm_component parent = null);
        super.new(name, parent);
    endfunction

    function void build_phase(uvm_phase phase);
        super.build_phase(phase);
        mon_ap = new("mon_ap", this);
    endfunction

    task run_phase(uvm_phase phase);
        super.run_phase(phase);
        forever begin
        rsp_seq_item = slave_seq_item::type_id::create("rsp_seq_item");
        @(negedge slave_vif.clk);
        rsp_seq_item.MOSI     = slave_vif.MOSI;
        rsp_seq_item.rst_n    = slave_vif.rst_n;
        rsp_seq_item.SS_n     = slave_vif.SS_n;
        rsp_seq_item.tx_valid = slave_vif.tx_valid;
        rsp_seq_item.tx_data  = slave_vif.tx_data;
        rsp_seq_item.rx_data  = slave_vif.rx_data;
        rsp_seq_item.rx_valid = slave_vif.rx_valid;
        rsp_seq_item.MISO     = slave_vif.MISO;
        // golden
        rsp_seq_item.rx_data_golden  = slave_vif.rx_data_golden;
        rsp_seq_item.rx_valid_golden = slave_vif.rx_valid_golden;
        rsp_seq_item.MISO_golden     = slave_vif.MISO_golden;
        mon_ap.write(rsp_seq_item);
        `uvm_info("run_phase", rsp_seq_item.convert2string(), UVM_HIGH)
    end
endtask
endclass
endpackage

```

```
package agent_pkg;
    import uvm_pkg::*;
    import seq_item_pkg::*;
    import driver_pkg::*;
    import sequencer_pkg::*;
    import monitor_pkg::*;
    import config_pkg::*;
    `include "uvm_macros.svh"

    class slave_agent extends uvm_agent;
        `uvm_component_utils(slave_agent)
        slave_sequencer sqr;
        slave_driver drv;
        slave_monitor mon;
        slave_config slave_cfg;
        uvm_analysis_port #(slave_seq_item) agt_ap;

        function new(string name = "slave_agent", uvm_component parent = null);
            super.new(name, parent);
        endfunction

        function void build_phase(uvm_phase phase);
            super.build_phase(phase);
            if(!uvm_config_db#(slave_config)::get(this, "", "cfg", slave_cfg))
            `uvm_fatal("build_phase", "errrrrrorrrrrr")

            sqr = slave_sequencer::type_id::create("sqr", this);
            drv = slave_driver   ::type_id::create("drv", this);
            mon = slave_monitor  ::type_id::create("mon", this);
            agt_ap = new("agt_ap", this);
        endfunction

        function void connect_phase(uvm_phase phase);
            super.connect_phase(phase);
            drv.slave_vif = slave_cfg.slave_vif;
            mon.slave_vif = slave_cfg.slave_vif;
            drv.seq_item_port.connect(sqr.seq_item_export);
            mon.mon_ap.connect(agt_ap);
        endfunction
    endclass
endpackage

```

```
package cov_col_pkg;
    import uvm_pkg::*;
    import seq_item_pkg::*;
    import shared_pkg::*;
    `include "uvm_macros.svh"
    class slave_coverage extends uvm_component;
        `uvm_component_utils(slave_coverage)
        uvm_analysis_export #(slave_seq_item) cov_export;
        uvm_tlm_analysis_fifo #(slave_seq_item) cov_fifo;
        slave_seq_item seq_item_cov;

        ////////////////// cover group /////////////////
        covergroup cov_gp;
            rx_data_cp : coverpoint seq_item_cov.rx_data[9:8];
            rx_trans_cp: coverpoint seq_item_cov.rx_data[9:8] {
                bins t_00_00 = (2'b00 => 2'b00);
                bins t_00_01 = (2'b00 => 2'b01);
                bins t_00_10 = (2'b00 => 2'b10);
                bins t_01_00 = (2'b01 => 2'b00);
                bins t_01_01 = (2'b01 => 2'b01);
                bins t_01_11 = (2'b01 => 2'b11);
                bins t_10_00 = (2'b10 => 2'b00);
                bins t_10_11 = (2'b10 => 2'b11);
                bins t_11_00 = (2'b11 => 2'b00);
                bins t_11_01 = (2'b11 => 2'b01);
                bins t_11_11 = (2'b11 => 2'b11);
                }
            ss_n_cp : coverpoint seq_item_cov.SS_n {
                bins normal_tr   = (1 => 0[*13] => 1);
                bins extended_tr = (1 => 0[*23] => 1);
            }
            mosi_cp : coverpoint seq_item_cov.MOSI {
                bins write_addr = (0 => 0 => 0);
                bins write_data = (0 => 0 => 1);
                bins read_addr  = (1 => 1 => 0);
                bins read_data  = (1 => 1 => 1);
            }
            patterns_by_protocol : cross ss_n_cp, mosi_cp {
                ignore_bins wr_addr = binsof(ss_n_cp.extended_tr) && binsof(mosi_cp.write_addr);
                ignore_bins wr_data = binsof(ss_n_cp.extended_tr) && binsof(mosi_cp.write_data);
                ignore_bins rd_addr = binsof(ss_n_cp.extended_tr) && binsof(mosi_cp.read_addr);
            }

        endgroup

        function new(string name = "slave_coverage", uvm_component parent = null);
            super.new(name, parent);
            // covergroups inst.
            cov_gp = new;
        endfunction

        function void build_phase(uvm_phase phase);
            super.build_phase(phase);
            cov_export = new("cov_export", this);
            cov_fifo = new("cov_info", this);
        endfunction

        function void connect_phase(uvm_phase phase);
            super.connect_phase(phase);
            cov_export.connect(cov_fifo.analysis_export);
        endfunction

        task run_phase(uvm_phase phase);
            super.run_phase(phase);
            forever begin
            cov_fifo.get(seq_item_cov);
            cov_gp.sample();
            end
        endtask
    endclass
endpackage

```

```
package scoreboard_pkg;
    import uvm_pkg::*;
    import seq_item_pkg::*;
    import shared_pkg::*;
    `include "uvm_macros.svh"
    class slave_scoreboard extends uvm_scoreboard;
        `uvm_component_utils(slave_scoreboard)
        uvm_analysis_export #(slave_seq_item) sb_export;
        uvm_tlm_analysis_fifo #(slave_seq_item) sb_fifo;
        slave_seq_item seq_item_sb;

        function new(string name = "slave_scoreboard", uvm_component parent = null);
            super.new(name, parent);
        endfunction

        function void build_phase(uvm_phase phase);
            super.build_phase(phase);
            sb_export = new("sb_export", this);
            sb_fifo   = new("sb_fifo",   this);
        endfunction

        function void connect_phase(uvm_phase phase);
            super.connect_phase(phase);
            sb_export.connect(sb_fifo.analysis_export);
        endfunction

        task run_phase(uvm_phase phase);
            super.run_phase(phase);
            forever begin
            sb_fifo.get(seq_item_sb);
            if (seq_item_sb.MISO_golden !== seq_item_sb.MISO) begin
                `uvm_error("run_phase", $sformatf("errrrrrrrrrrorrrrrrrr"))
                error++;
            end
            else begin
                `uvm_info("run_phase", $sformatf("corrrrreccccccccccccct"), UVM_HIGH)
                correct++;
            end
            end
        endtask

    endclass
endpackage
```

```
////////////////////////////////////////////////////////////////////////////////
// Author: Kareem Waseem
// Course: Digital Verification using SV & UVM
//
// Description: UVM Example
//
////////////////////////////////////////////////////////////////////////////////
package slave_env_pkg;
import uvm_pkg::*;
import agent_pkg::*;
import scoreboard_pkg::*;
import cov_col_pkg::*;
`include "uvm_macros.svh"
class slave_env extends uvm_env;
    `uvm_component_utils(slave_env)
    slave_agent      agt;
    slave_scoreboard sb;
    slave_coverage   cov;

    function new (string name = "slave_env", uvm_component parent = null);
        super.new(name, parent);
    endfunction

    function void build_phase (uvm_phase phase);
        super.build_phase(phase);
        agt = slave_agent     ::type_id::create("agt", this);
        sb  = slave_scoreboard::type_id::create("sb",  this);
        cov = slave_coverage  ::type_id::create("cov", this);
    endfunction

    function void connect_phase(uvm_phase phase);
        super.connect_phase(phase);
        agt.agt_ap.connect(sb.sb_export);
        agt.agt_ap.connect(cov.cov_export);
    endfunction
endclass
endpackage

```

```
////////////////////////////////////////////////////////////////////////////////
// Author: Kareem Waseem
// Course: Digital Verification using SV & UVM
//
// Description: UVM Example
//
////////////////////////////////////////////////////////////////////////////////
package slave_test_pkg;
import slave_env_pkg::*;
import uvm_pkg::*;
import config_pkg::*;
import main_seq_pkg::*;
import reset_seq_pkg::*;
import slave_env_pkg::*;
`include "uvm_macros.svh"

class slave_test extends uvm_test;
  // Do the essentials (factory register & Constructor)
    `uvm_component_utils(slave_test)

    slave_env env;
    slave_config slave_cfg;
    virtual slave_if slave_vif;
    slave_main_seq main_seq;
    slave_reset_seq reset_seq;

    function new (string name = "slave_test", uvm_component parent = null);
        super.new (name, parent);
    endfunction

    function void build_phase (uvm_phase phase);
        super.build_phase (phase);
        env       = slave_env      ::type_id::create("env", this);
        slave_cfg  = slave_config   ::type_id::create("slave_cfg");
        main_seq  = slave_main_seq ::type_id::create("main_seq");
        reset_seq = slave_reset_seq::type_id::create("reset_seq");

        if (!uvm_config_db#(virtual slave_if)::get(this, "", "ifk", slave_cfg.slave_vif))
            `uvm_fatal ("build_phase", "errrrrrrrrrrrrrorrrrrrrrrrrrrrrrr");

        uvm_config_db#(slave_config)::set(this, "*", "cfg", slave_cfg);
    endfunction

    task run_phase (uvm_phase phase);
        super.run_phase (phase);
        phase.raise_objection(this);
        `uvm_info ("run_phase", "welcommmmmmmmmmmmmmmmmmmmmmmme", UVM_MEDIUM)
        reset_seq.start(env.agt.sqr);
        main_seq.start(env.agt.sqr);
        phase.drop_objection(this);
    endtask

endclass: slave_test
endpackage
```

```
////////////////////////////////////////////////////////////////////////////////
// Author: Kareem Waseem
// Course: Digital Verification using SV & UVM
//
// Description: UVM Example
//
////////////////////////////////////////////////////////////////////////////////
import uvm_pkg::*;
import slave_test_pkg::*;
`include "uvm_macros.svh"

module top();
  // Clock generation
    bit clk;
    initial forever #1 clk = !clk;
  // Instantiate the interface and DUT
    slave_if a_if (clk);
    SLAVE dut (a_if.MOSI,a_if.MISO,a_if.SS_n,clk,a_if.rst_n,a_if.rx_data,a_if.rx_valid,a_if.tx_data,a_if.tx_valid);
    golden_model golden (a_if.MOSI,a_if.MISO_golden,a_if.SS_n,clk,a_if.rst_n,a_if.rx_data_golden,a_if.rx_valid_golden,a_if.tx_data,a_if.tx_valid);
    bind SLAVE sva sva_inst(clk,a_if.MOSI,a_if.rst_n,a_if.SS_n,a_if.tx_valid,a_if.tx_data,a_if.rx_data,a_if.rx_valid,a_if.MISO);
    initial begin
        uvm_config_db #(virtual slave_if)::set(null, "uvm_test_top", "ifk", a_if);
        run_test("slave_test");
    end
endmodule

```

Golden Model

```
module golden_model (MOSI,MISO,SS_n,clk,rst_n,rx_data,rx_valid,tx_data,tx_valid);

localparam IDLE      = 3'b000;
localparam WRITE     = 3'b001;
localparam CHK_CMD   = 3'b010;
localparam READ_ADD  = 3'b011;
localparam READ_DATA = 3'b100;

input           MOSI, clk, rst_n, SS_n, tx_valid;
input      [7:0] tx_data;
output reg [9:0] rx_data;
output reg       rx_valid, MISO;


reg [2:0] cs , ns;
reg ADD_DATA_checker;
reg [3:0] counter1;
reg [3:0] counter2;

    //state memory
    always @(posedge clk)
    begin
        if(~rst_n)
            cs <= IDLE;
        else
            cs <= ns ;
    end

    //next state logic
    always @(*) begin
        case(cs)
            IDLE : begin
                if(SS_n)
                    ns = IDLE;
                else
                    ns = CHK_CMD;
            end
            CHK_CMD : begin
                if(SS_n)
                    ns = IDLE;
                else begin
                    if((MOSI == 0))
                        ns = WRITE;
                    else begin
                        if (ADD_DATA_checker)
                            ns = READ_ADD;
                        else
                            ns = READ_DATA;
                        end
                    end
            end
            WRITE : begin
                if(SS_n)
                    ns = IDLE;
                else
                    ns = WRITE;
            end
            READ_ADD : begin
                if(SS_n)
                    ns = IDLE;
                else
                    ns = READ_ADD;
            end
            READ_DATA : begin
                if(SS_n)
                    ns = IDLE;
                else
                    ns = READ_DATA;
            end
        endcase
    end

    //output logic
    always @(posedge clk) begin
        if (~rst_n) begin
            counter1 <= 10;
            counter2 <= 8;
            ADD_DATA_checker <= 1;
            rx_data  <= 0;
            rx_valid <= 0;
            MISO     <= 0;
        end
        //IDLE state
        else begin
            if(cs == IDLE) begin
                rx_valid <= 0 ;
                counter1 <= 10;
                counter2 <= 8 ;
            end
            //WRITE state
            else if(cs == WRITE) begin
                if (counter1 > 0)begin
                    rx_data[counter1-1] <= MOSI;
                    counter1 <= counter1 - 1;
                end
                else begin
                    rx_valid <= 1;
                end
            end
            else if (cs == READ_ADD) begin
                if (counter1 > 0)begin
                    rx_data[counter1-1] <= MOSI;
                    counter1 <= counter1 - 1;
                end
                else begin
                    rx_valid <= 1;
                    ADD_DATA_checker <= 0;
                end
            end
        //READ_DATA state
            else if (cs == READ_DATA) begin
                if(tx_valid) begin
                    rx_valid <= 0;
                    if (counter2 > 0) begin
                        MISO <= tx_data[counter2-1];
                        counter2 <= counter2 - 1;
                    end
                    else
                        ADD_DATA_checker <= 1;
                end
                else begin
                    if (counter1 > 0)begin
                        rx_data[counter1-1] <= MOSI;
                        counter1 <= counter1 - 1;
                    end
                    else begin
                        rx_valid <= 1;
                        counter1 <= 10;
                    end
                end
            end
        end

    end
endmodule
```

Do file

```
vlib work
vlog -f src_files.list +cover -covercells +define+SIM
vsim -voptargs=+acc work.top -classdebug -uvmcontrol=all -cover

run 0
do wave.do
run -all

#coverage save UVM.ucdb -onexit
#vcover report UVM.ucdb -details -all -annotate -output coverage_rpt.txt

```

```
slave_if.sv
SPI_slave.sv
golden_model.v
sva.sv
shared_pkg.sv
config.sv
seq_item.sv
main_seq.sv
reset_seq.sv
sequencer.sv
driver.sv
monitor.sv
agent.sv
cov_col.sv
scoreboard.sv
env.sv
test.sv
top.sv

```

Verification plan

Label	Design Requirement Description	Stimulus Generation	Functional Coverage	Functionality Check
Reset	When the reset is asserted the output value should be zero	Directed at the start of the simulation	-	A checker in the test bench to make sure the output is correct.
MISO	Verifying output in case of reading data	Randomized during the simulation	Covering all valid transitions	A checker in the scoreboard to make sure the output is correct compared to the golden model.
Reset_ia	When the reset is asserted the MISO, rx_valid, and rx_data signals should be zero	Randomized during the simulation under the constraints of reset to be off 99% of the time	-	A checker in the design module to make sure the output is correct. (assertion)
idle_chk	**check correct FSM transitions**	Randomized during the simulation	-	A checker in the design module to make sure the output is correct. (assertion)
chk_wr	**check correct FSM transitions**	Randomized during the simulation	-	A checker in the design module to make sure the output is correct. (assertion)
wr_idle	**check correct FSM transitions**	Randomized during the simulation	-	A checker in the design module to make sure the output is correct. (assertion)
rd_add_idle	**check correct FSM transitions**	Randomized during the simulation	-	A checker in the design module to make sure the output is correct. (assertion)
rd_data_idle	**check correct FSM transitions**	Randomized during the simulation	-	A checker in the design module to make sure the output is correct. (assertion)

Coverage Report

coverage_rpt.txt

File Edit View

```
Coverage Report by instance with details

================================================================================
=== Instance: /top/a_if
=== Design Unit: work.slave_if
================================================================================
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         74        74         0   100.00%

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

Toggle Coverage for instance /top/a_if --

                                          Node      1H->0L      0L->1H  "Coverage"
                                          ----------------------------------------
                                          MISO           1           1      100.00
                                    MISO_golden          1           1      100.00
                                          MOSI           1           1      100.00
                                          SS_n           1           1      100.00
                                           clk           1           1      100.00
                                         rst_n           1           1      100.00
                                  rx_data[0-9]           1           1      100.00
                             rx_data_golden[0-9]           1           1      100.00
                                      rx_valid          1           1      100.00
                                rx_valid_golden          1           1      100.00
                                   tx_data[0-7]          1           1      100.00
                                       tx_valid          1           1      100.00

Total Node Count     =         37
Toggled Node Count   =         37
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (74 of 74 bins)

================================================================================
=== Instance: /top/dut/sva_inst
=== Design Unit: work.sva
================================================================================

Assertion Coverage:
    Assertions                       3         3         0   100.00%
--------------------------------------------------------------------
Name                 File(Line)                   Failure      Pass
                                                  Count        Count
--------------------------------------------------------------------
/top/dut/sva_inst/miso_a
                     sva.sv(10)                         0          1
/top/dut/sva_inst/rx_valid_a
                     sva.sv(12)                         0          1
/top/dut/sva_inst/rx_data_a
                     sva.sv(14)                         0          1
```

Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8

coverage_rpt.txt

File Edit View

```
Coverage Report by instance with details

================================================================================
=== Instance: /top/a_if
=== Design Unit: work.slave_if
================================================================================
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         74        74         0   100.00%

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

Toggle Coverage for instance /top/a_if --

                                          Node      1H->0L      0L->1H  "Coverage"
                                          ----------------------------------------
                                          MISO           1           1      100.00
                                  MISO_golden            1           1      100.00
                                          MOSI           1           1      100.00
                                          SS_n           1           1      100.00
                                           clk           1           1      100.00
                                         rst_n           1           1      100.00
                                  rx_data[0-9]           1           1      100.00
                             rx_data_golden[0-9]           1           1      100.00
                                      rx_valid           1           1      100.00
                                rx_valid_golden           1           1      100.00
                                   tx_data[0-7]           1           1      100.00
                                       tx_valid           1           1      100.00

Total Node Count     =         37
Toggled Node Count   =         37
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (74 of 74 bins)

================================================================================
=== Instance: /top/dut/sva_inst
=== Design Unit: work.sva
================================================================================

Assertion Coverage:
    Assertions                       3         3         0   100.00%
--------------------------------------------------------------------
Name                 File(Line)                   Failure      Pass
                                                  Count        Count
--------------------------------------------------------------------
/top/dut/sva_inst/miso_a
                     sva.sv(10)                          0          1
/top/dut/sva_inst/rx_valid_a
                     sva.sv(12)                          0          1
/top/dut/sva_inst/rx_data_a
                     sva.sv(14)                          0          1
```

Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8

coverage_rpt.txt

File Edit View

```
================================================================================
=== Instance: /top/dut/sva_inst
=== Design Unit: work.sva
================================================================================

Assertion Coverage:
    Assertions                       3         3         0   100.00%
----------------------------------------------------------------
Name                 File(Line)                   Failure      Pass
                                                  Count        Count
----------------------------------------------------------------
/top/dut/sva_inst/miso_a
                     sva.sv(10)                         0          1
/top/dut/sva_inst/rx_valid_a
                     sva.sv(12)                         0          1
/top/dut/sva_inst/rx_data_a
                     sva.sv(14)                         0          1

Directive Coverage:
    Directives                       3         3         0   100.00%

DIRECTIVE COVERAGE:
--------------------------------------------------------------------------------------------
Name                                    Design Design   Lang File(Line)      Hits Status
                                        Unit   UnitType
--------------------------------------------------------------------------------------------
/top/dut/sva_inst/miso_c                sva    Verilog  SVA  sva.sv(11)       106 Covered
/top/dut/sva_inst/rx_valid_c            sva    Verilog  SVA  sva.sv(13)       106 Covered
/top/dut/sva_inst/rx_data_c             sva    Verilog  SVA  sva.sv(15)       106 Covered
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         50        50         0   100.00%

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

Toggle Coverage for instance /top/dut/sva_inst --

                                          Node      1H->0L      0L->1H  "Coverage"
                                          ----------------------------------------
                                          MISO           1           1      100.00
                                          MOSI           1           1      100.00
                                          SS_n           1           1      100.00
                                           clk           1           1      100.00
                                         rst_n           1           1      100.00
                                  rx_data[0-9]           1           1      100.00
                                      rx_valid           1           1      100.00
                                  tx_data[0-7]           1           1      100.00
                                      tx_valid           1           1      100.00

Total Node Count     =         25
Toggled Node Count   =         25
Untoggled Node Count =          0
```

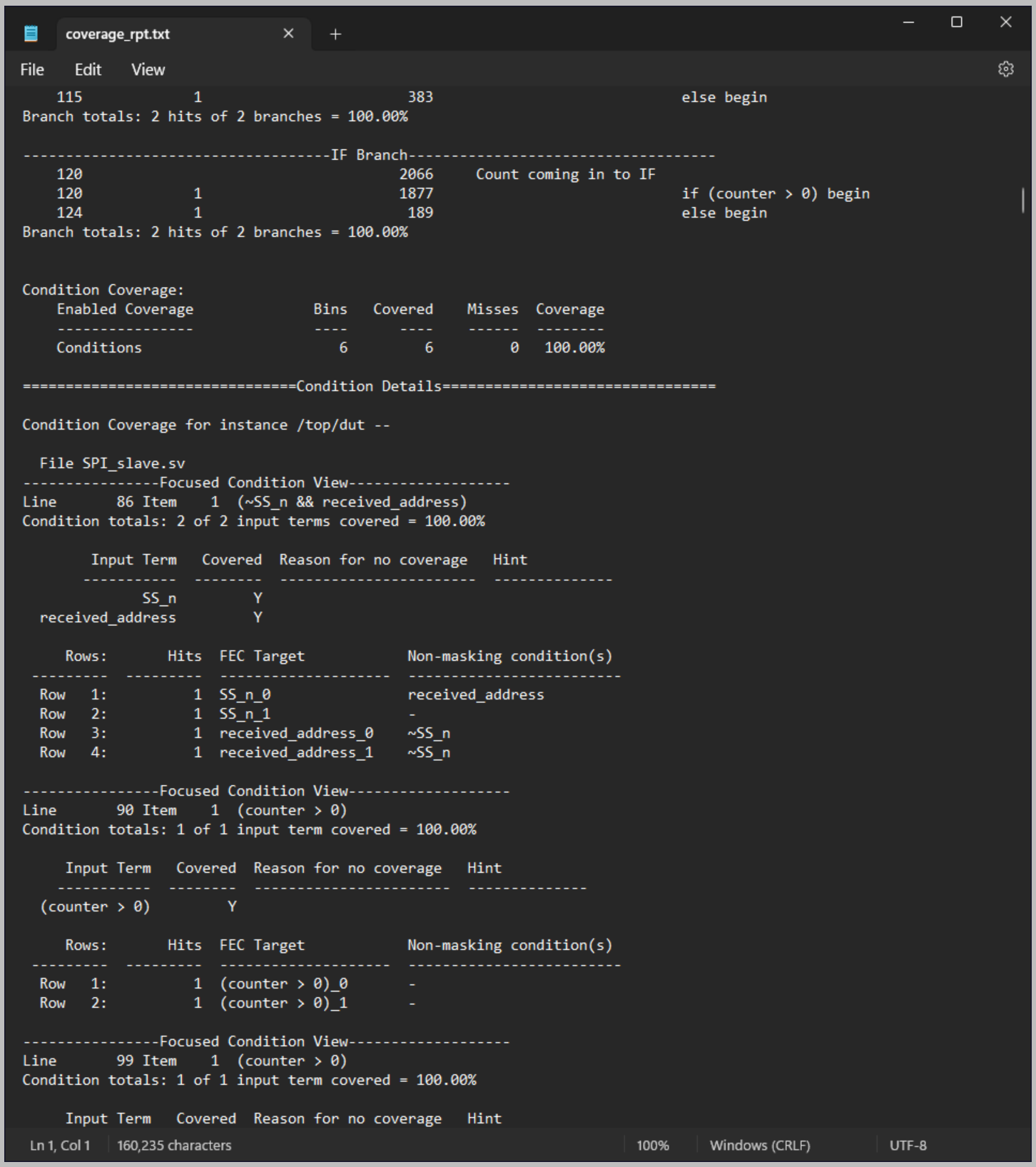
Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8

coverage_rpt.txt

File Edit View

```
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         50        50         0   100.00%

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

Toggle Coverage for instance /top/dut/sva_inst --

                                          Node      1H->0L      0L->1H  "Coverage"
                                          ----------------------------------------
                                          MISO           1           1      100.00
                                          MOSI           1           1      100.00
                                          SS_n           1           1      100.00
                                           clk           1           1      100.00
                                         rst_n           1           1      100.00
                                  rx_data[0-9]           1           1      100.00
                                      rx_valid           1           1      100.00
                                  tx_data[0-7]           1           1      100.00
                                      tx_valid           1           1      100.00

Total Node Count     =         25
Toggled Node Count   =         25
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (50 of 50 bins)

================================================================================
=== Instance: /top/dut
=== Design Unit: work.SLAVE
================================================================================

Assertion Coverage:
    Assertions                       5         5         0   100.00%
--------------------------------------------------------------------
Name                 File(Line)                   Failure      Pass
                                                  Count        Count
--------------------------------------------------------------------
/top/dut/idle_chk_a  SPI_slave.sv(141)                  0          1
/top/dut/chk_wr_a    SPI_slave.sv(147)                  0          1
/top/dut/wr_idle_a   SPI_slave.sv(153)                  0          1
/top/dut/rd_add_idle_a
                     SPI_slave.sv(159)                  0          1
/top/dut/rd_data_idle_a
                     SPI_slave.sv(165)                  0          1
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        36        36         0   100.00%

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

Branch Coverage for instance /top/dut

    Line         Item                      Count     Source
```

Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8


coverage_rpt.txt
File Edit View
115 1 383 else begin
Branch totals: 2 hits of 2 branches = 100.00%
----------IF Branch----------
120 2066 Count coming in to IF
120 1 1877 if (counter > 0) begin
124 1 189 else begin
Branch totals: 2 hits of 2 branches = 100.00%
Condition Coverage:
Enabled Coverage Bins Covered Misses Coverage
Conditions 6 6 0 100.00%
==========Condition Details==========
Condition Coverage for instance /top/dut --
File SPI_slave.sv
----------Focused Condition View----------
Line 86 Item 1 (~SS_n && received_address)
Condition totals: 2 of 2 input terms covered = 100.00%
Input Term Covered Reason for no coverage Hint
SS_n Y
received_address Y
Rows: Hits FEC Target Non-masking condition(s)
Row 1: 1 SS_n_0 received_address
Row 2: 1 SS_n_1 -
Row 3: 1 received_address_0 ~SS_n
Row 4: 1 received_address_1 ~SS_n
----------Focused Condition View----------
Line 90 Item 1 (counter > 0)
Condition totals: 1 of 1 input term covered = 100.00%
Input Term Covered Reason for no coverage Hint
(counter > 0) Y
Rows: Hits FEC Target Non-masking condition(s)
Row 1: 1 (counter > 0)_0 -
Row 2: 1 (counter > 0)_1 -
----------Focused Condition View----------
Line 99 Item 1 (counter > 0)
Condition totals: 1 of 1 input term covered = 100.00%
Input Term Covered Reason for no coverage Hint
Ln 1, Col 1
160,235 characters
100%
Windows (CRLF)
UTF-8

coverage_rpt.txt

File Edit View

```
---------------Focused Condition View-------------------
Line       120 Item    1  (counter > 0)
Condition totals: 1 of 1 input term covered = 100.00%

     Input Term   Covered  Reason for no coverage   Hint
    -----------  --------  ----------------------  -------------
  (counter > 0)         Y

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



Directive Coverage:
    Directives                       5         5         0   100.00%

DIRECTIVE COVERAGE:
-----------------------------------------------------------------------------------------
Name                                 Design Design   Lang File(Line)      Hits Status
                                     Unit   UnitType
-----------------------------------------------------------------------------------------
/top/dut/idle_chk_c                  SLAVE  Verilog  SVA  SPI_slave.sv(142)
                                                                          734 Covered
/top/dut/chk_wr_c                    SLAVE  Verilog  SVA  SPI_slave.sv(148)
                                                                          607 Covered
/top/dut/wr_idle_c                   SLAVE  Verilog  SVA  SPI_slave.sv(154)
                                                                         3768 Covered
/top/dut/rd_add_idle_c               SLAVE  Verilog  SVA  SPI_slave.sv(160)
                                                                         1550 Covered
/top/dut/rd_data_idle_c              SLAVE  Verilog  SVA  SPI_slave.sv(166)
                                                                         2550 Covered
FSM Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    FSM States                       5         5         0   100.00%
    FSM Transitions                  8         8         0   100.00%

================================FSM Details================================

FSM Coverage for instance /top/dut --

FSM_ID: cs
    Current State Object : cs
    ---------------------
    State Value MapInfo :
    ---------------------
Line          State Name              Value
----          ----------              -----
  30               IDLE                 0
  36             CHK_CMD                2
  62           READ_DATA                4
  56            READ_ADD                3
```

Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8

coverage_rpt.txt

File Edit View

```
    Summary                     Bins      Hits    Misses  Coverage
    -------                     ----      ----    ------  --------
        FSM States                 5         5         0   100.00%
        FSM Transitions            8         8         0   100.00%
Statement Coverage:
    Enabled Coverage            Bins      Hits    Misses  Coverage
    ----------------            ----      ----    ------  --------
    Statements                   40        40         0   100.00%

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

Statement Coverage for instance /top/dut --

    Line         Item                     Count     Source
    ----         ----                     -----     ------
  File SPI_slave.sv
    1                                               module SLAVE
(MOSI,MISO,SS_n,clk,rst_n,rx_data,rx_valid,tx_data,tx_valid);
    2
    3                                                   localparam IDLE      = 3'b000;
    4                                                   localparam WRITE     = 3'b001;
    5                                                   localparam CHK_CMD   = 3'b010;
    6                                                   localparam READ_ADD  = 3'b011;
    7                                                   localparam READ_DATA = 3'b100;
    8
    9                                                   input            MOSI, clk, rst_n, SS_n, tx_valid;
    10                                                  input      [7:0] tx_data;
    11                                                  output reg [9:0] rx_data;
    12                                                  output reg       rx_valid, MISO;
    13
    14                                                  reg [3:0] counter;
    15                                                  reg       received_address;
    16
    17                                                  reg [2:0] cs, ns;
    18
    19              1                        2915       always @(posedge clk) begin
    20                                                      if (~rst_n) begin
    21              1                         105               cs <= IDLE;
    22                                                      end
    23                                                      else begin
    24              1                        2810               cs <= ns;
    25                                                      end
    26                                                  end
    27
    28              1                        3628       always @(*) begin
    29                                                      case (cs)
    30                                                          IDLE : begin
    31                                                              if (SS_n)
    32              1                         651                       ns = IDLE;
    33                                                              else
    34              1                         746                       ns = CHK_CMD;
    35                                                          end
    36                                                          CHK_CMD : begin
```

Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8

coverage_rpt.txt

File Edit View

```
    122             1                       1877                                counter <= counter - 1;
    123                                                                       end
    124                                                                       else begin
    125             1                        189                                rx_valid <= 1;
    126             1                        189                                counter  <= 10; //////10

Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         72        72         0   100.00%

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

Toggle Coverage for instance /top/dut --

                                          Node      1H->0L      0L->1H  "Coverage"
                                          ----------------------------------------
                                          MISO           1           1      100.00
                                          MOSI           1           1      100.00
                                          SS_n           1           1      100.00
                                           clk           1           1      100.00
                                   counter[0-3]          1           1      100.00
                                        cs[0-2]          1           1      100.00
                                        ns[0-2]          1           1      100.00
                                received_address         1           1      100.00
                                           rst_n         1           1      100.00
                                    rx_data[0-9]         1           1      100.00
                                        rx_valid         1           1      100.00
                                    tx_data[0-7]         1           1      100.00
                                        tx_valid         1           1      100.00

Total Node Count     =         36
Toggled Node Count   =         36
Untoggled Node Count =          0

Toggle Coverage      =     100.00% (72 of 72 bins)

================================================================================
=== Instance: /top/golden
=== Design Unit: work.golden_model
================================================================================
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                        39        39         0   100.00%

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

Branch Coverage for instance /top/golden

    Line         Item                     Count     Source
    ----         ----                     -----     ------
  File golden_model.v
----------------------------------IF Branch-------------------------------------
```

Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8

coverage_rpt.txt

File Edit View

```
------------------------------------IF Branch------------------------------------
    114                                 668     Count coming in to IF
    114             1                   285                         if (counter2 > 0) begin
    118             1                   383                         else
Branch totals: 2 hits of 2 branches = 100.00%

------------------------------------IF Branch------------------------------------
    122                                2066     Count coming in to IF
    122             1                  1901                         if (counter1 > 0)begin
    126             1                   165                         else begin
Branch totals: 2 hits of 2 branches = 100.00%


Condition Coverage:
    Enabled Coverage              Bins   Covered    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Conditions                       8         8         0   100.00%

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

Condition Coverage for instance /top/golden --

  File golden_model.v
----------------Focused Condition View-------------------
Line       85 Item    1  (cs == 0)
Condition totals: 1 of 1 input term covered = 100.00%

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

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

----------------Focused Condition View-------------------
Line       91 Item    1  (cs == 1)
Condition totals: 1 of 1 input term covered = 100.00%

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

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

----------------Focused Condition View-------------------
Line       92 Item    1  (counter1 > 0)
Condition totals: 1 of 1 input term covered = 100.00%

      Input Term   Covered  Reason for no coverage   Hint
```

Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8

coverage_rpt.txt

File Edit View

```
  (counter1 > 0)          Y

     Rows:       Hits  FEC Target           Non-masking condition(s)
 ---------  ---------  -------------------  -------------------------
  Row   1:          1  (counter1 > 0)_0      -
  Row   2:          1  (counter1 > 0)_1      -


FSM Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    FSM States                       5         5         0   100.00%
    FSM Transitions                  8         8         0   100.00%

================================FSM Details================================

FSM Coverage for instance /top/golden --

FSM_ID: cs
    Current State Object : cs
    ---------------------
    State Value MapInfo :
    ---------------------
Line          State Name              Value
----          ----------              -----
  32               IDLE                  0
  38             CHK_CMD                 2
  64           READ_DATA                 4
  58            READ_ADD                 3
  52               WRITE                 1
    Covered States :
    ----------------
                   State          Hit_count
                   -----          ---------
                    IDLE                750
                  CHK_CMD               738
                READ_DATA               449
                READ_ADD                315
                   WRITE                663
    Covered Transitions :
    --------------------
Line           Trans_ID          Hit_count          Transition
----           --------          ---------          ----------
  36                  0                738          IDLE -> CHK_CMD
  48                  1                225          CHK_CMD -> READ_DATA
  46                  2                159          CHK_CMD -> READ_ADD
  43                  3                334          CHK_CMD -> WRITE
  40                  4                 20          CHK_CMD -> IDLE
  66                  5                225          READ_DATA -> IDLE
  60                  6                158          READ_ADD -> IDLE
  54                  7                334          WRITE -> IDLE


    Summary                       Bins      Hits    Misses  Coverage
```

Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8

coverage_rpt.txt

File Edit View

```
  66                5               225          READ_DATA -> IDLE
  60                6               158          READ_ADD -> IDLE
  54                7               334          WRITE -> IDLE


    Summary                     Bins      Hits    Misses  Coverage
    -------                     ----      ----    ------  --------
        FSM States                 5         5         0   100.00%
        FSM Transitions            8         8         0   100.00%
Statement Coverage:
    Enabled Coverage            Bins      Hits    Misses  Coverage
    ----------------            ----      ----    ------  --------
    Statements                   41        41         0   100.00%

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

Statement Coverage for instance /top/golden --

    Line         Item                     Count     Source
    ----         ----                     -----     ------
  File golden_model.v
    1                                              module golden_model
(MOSI,MISO,SS_n,clk,rst_n,rx_data,rx_valid,tx_data,tx_valid);
    2
    3                                              localparam IDLE      = 3'b000;
    4                                              localparam WRITE     = 3'b001;
    5                                              localparam CHK_CMD   = 3'b010;
    6                                              localparam READ_ADD  = 3'b011;
    7                                              localparam READ_DATA = 3'b100;
    8
    9                                              input            MOSI, clk, rst_n, SS_n, tx_valid;
    10                                             input      [7:0] tx_data;
    11                                             output reg [9:0] rx_data;
    12                                             output reg       rx_valid, MISO;
    13
    14
    15                                             reg [2:0] cs , ns;
    16                                             reg ADD_DATA_checker;
    17                                             reg [3:0] counter1;
    18                                             reg [3:0] counter2;
    19
    20                                                 //state memory
    21             1                    2915        always @(posedge clk)
    22                                                 begin
    23                                                     if(~rst_n)
    24             1                     105                 cs <= IDLE;
    25                                                     else
    26             1                    2810                cs <= ns ;
    27                                                 end
    28
    29                                                 //next state logic
    30             1                    3628        always @(*) begin
    31                                                     case(cs)
    32                                                         IDLE : begin
```

Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8

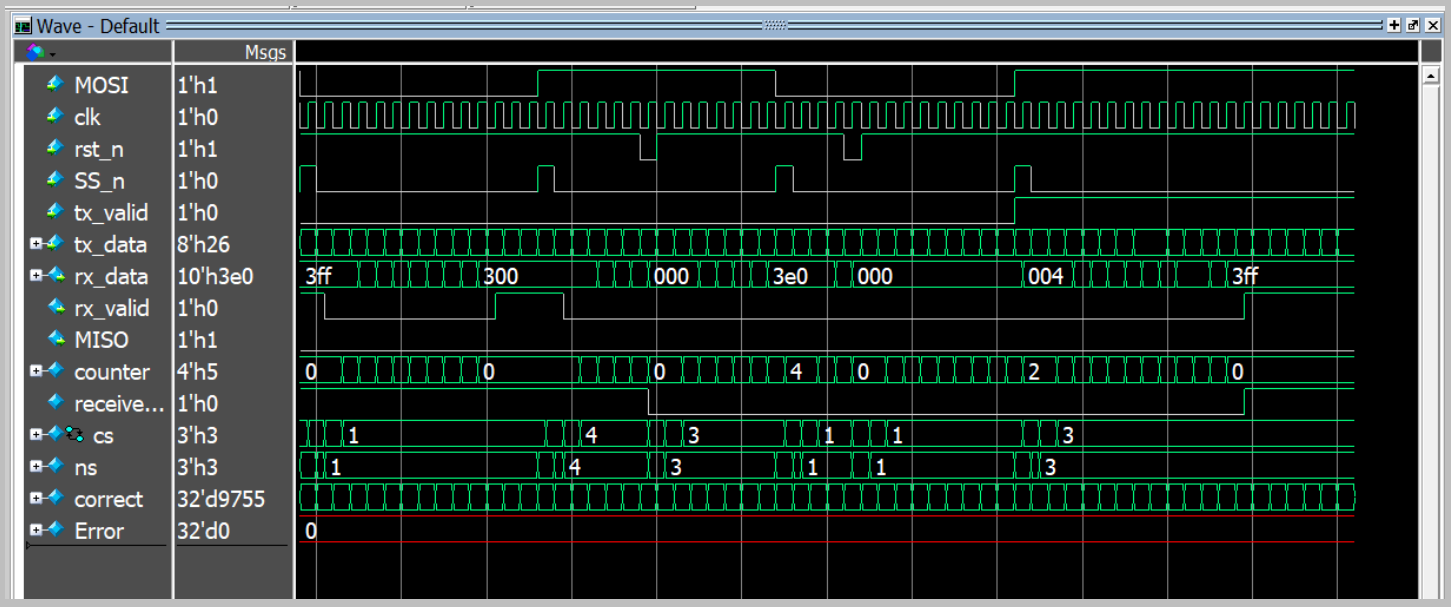
coverage_rpt.txt

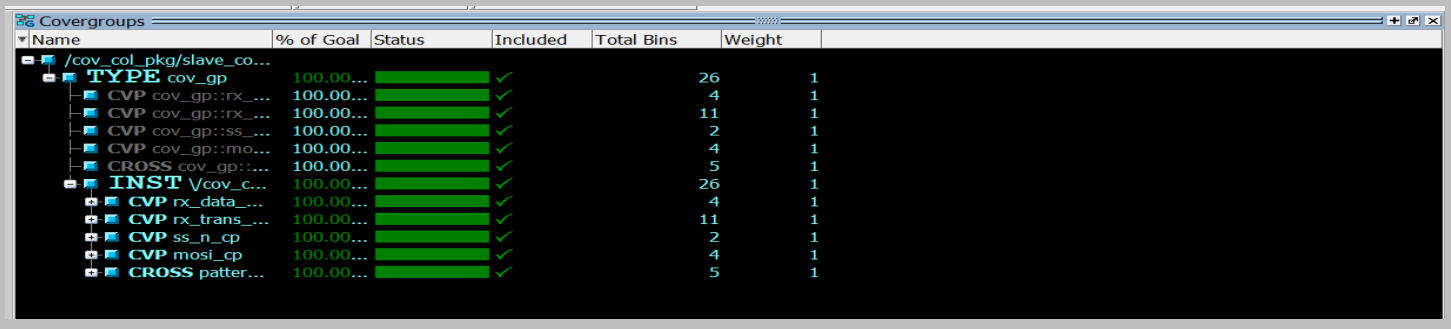
File Edit View

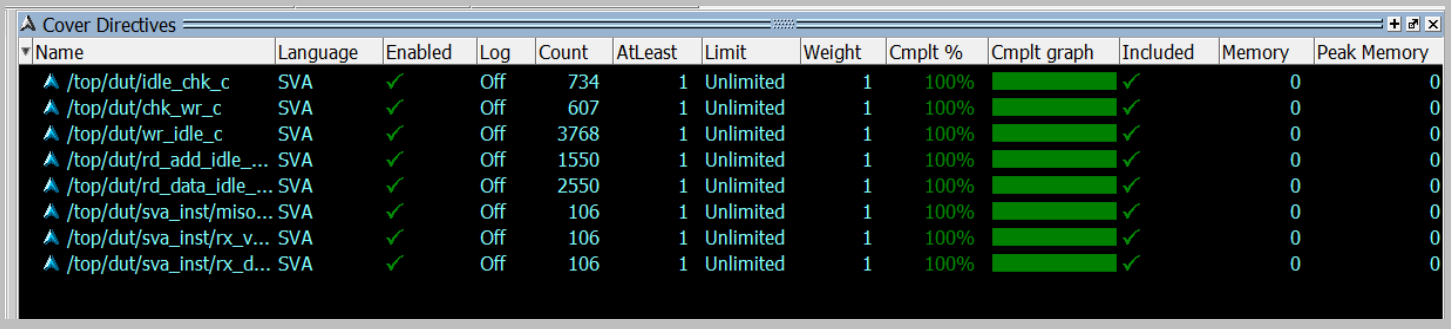
```
    14              1                       10000               repeat(10000) begin
    15              1                       10000               seq_item =
slave_seq_item::type_id::create("seq_item");
    16              1                       10000               start_item(seq_item);
    17                                                          assert(seq_item.randomize());
    18              1                       10000               finish_item(seq_item);


================================================================================
=== Instance: /cov_col_pkg
=== Design Unit: work.cov_col_pkg
================================================================================

Covergroup Coverage:
    Covergroups                     1        na        na   100.00%
        Coverpoints/Crosses         5        na        na        na
            Covergroup Bins        26        26         0   100.00%
----------------------------------------------------------------------------------------------------------------
Covergroup                                          Metric       Goal       Bins    Status

----------------------------------------------------------------------------------------------------------------
 TYPE /cov_col_pkg/slave_coverage/cov_gp             100.00%        100          -    Covered
    covered/total bins:                                  26         26          -
    missing/total bins:                                   0         26          -
    % Hit:                                           100.00%        100          -
    Coverpoint rx_data_cp                            100.00%        100          -    Covered
        covered/total bins:                               4          4          -
        missing/total bins:                               0          4          -
        % Hit:                                       100.00%        100          -
    Coverpoint rx_trans_cp                           100.00%        100          -    Covered
        covered/total bins:                              11         11          -
        missing/total bins:                               0         11          -
        % Hit:                                       100.00%        100          -
    Coverpoint ss_n_cp                               100.00%        100          -    Covered
        covered/total bins:                               2          2          -
        missing/total bins:                               0          2          -
        % Hit:                                       100.00%        100          -
    Coverpoint mosi_cp                               100.00%        100          -    Covered
        covered/total bins:                               4          4          -
        missing/total bins:                               0          4          -
        % Hit:                                       100.00%        100          -
    Cross patterns_by_protocol                       100.00%        100          -    Covered
        covered/total bins:                               5          5          -
        missing/total bins:                               0          5          -
        % Hit:                                       100.00%        100          -
 Covergroup instance \/cov_col_pkg::slave_coverage::cov_gp
                                                     100.00%        100          -    Covered
    covered/total bins:                                  26         26          -
    missing/total bins:                                   0         26          -
    % Hit:                                           100.00%        100          -
    Coverpoint rx_data_cp                            100.00%        100          -    Covered
        covered/total bins:                               4          4          -
        missing/total bins:                               0          4          -
        % Hit:                                       100.00%        100          -
```

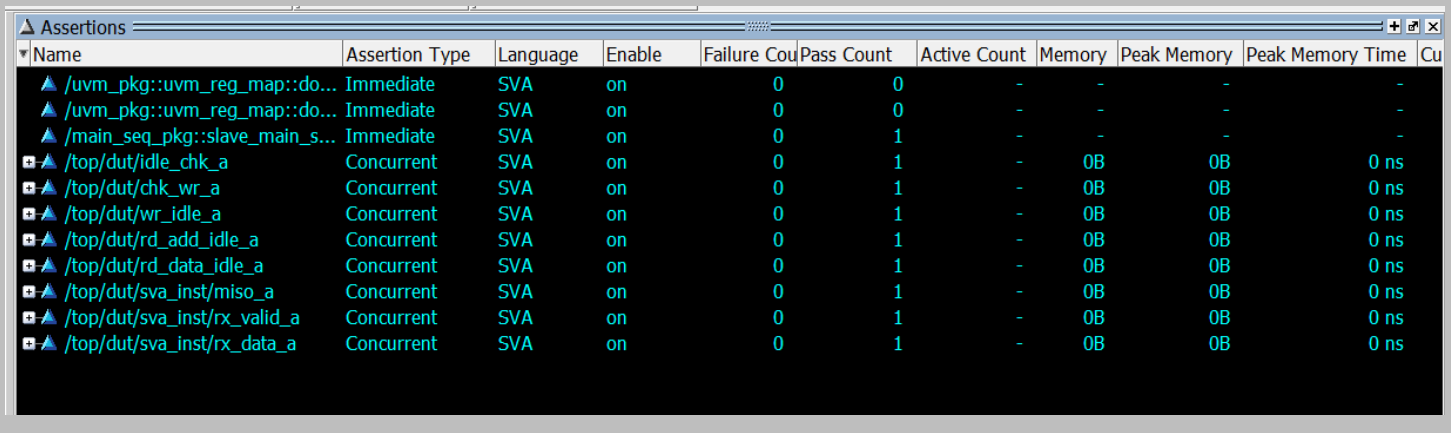
Ln 1, Col 1 | 160,235 characters | 100% | Windows (CRLF) | UTF-8

QuestaSim results









Thank you