Final Project

Project (3)-SPI-Wrapper:

1. Verification plan

a) Reset functionality should clear the flag, counter, rx_valid and MISO.
b) Check Starting communication phase when SS_n gets low.
c) Testing next state choice after current state is check command according to flag & MOSI.
d) Verify the serial to parallel conversion operation after 10 clock cycles after its start.
e) Checking the address is updated after the rx_valid is asserted in WRITE and READ_ADD cases.
f) Verifying the Read_data process.
g) Check End communication transition when master makes SS_n high.

Note:

- In Constraining and debugging the possibility of the master delaying the end communication signal (SS_n) was not taken in consideration for example if the master is writing in the RAM the high SS_n signal will come right after 11 clock cycles from starting the communication.

- In this DUT: dividing the inter tasks (counter, PISO, SIPO) in the SPI module resulted in inevitable delays that are not specified in the specs to correct these delays we would have to tear down the whole design. So, we just test for basic functionality neglecting timing specially in READ_DATA case as it clearly has a problem.

- We adjusted the testbench to match the design delays in order to check for the delayed output each clock cycle.

2. Verification Document

Label	Description	Stimulus Generation	Functional Coverage	Functionality Check
SLAVE_1	in case of rst_n the all rx_valid,rx_data,MISO will be equal zero	Randomization under constraints that reset_n is high most of the time	included the cover point of rst_n (rst_n_cp)and cross coverage with rx_valid (rx_valid_with_rst)and also with rx_data(rx_data_with_rst)	checking by using behavioral moddel for spi protocol
SLAVE_2	In case that the first 3 values of mosi after beginning communication are (000 or 001 or 110 or 111) the only valid squences to make a communication which is represent read (address(110) /data(111)) and write (address (000)/data(001))	randmisation of the mosi squence constrainting that the first 3 values inside the valid squences	coverpoint of rx_data(rx_data_cp) and cross coverage with the reset(rx_data_with_rst) and with SS_n(rx_data_with_SS_N)	checking by using behavioral moddel for spi protocol
SLAVE_3	SS_n is the key which strartts and ends the communication which means if I have raised SS_n during the anny part at the FSM the slave return to the idle state and not continue the communication	randomization constrainting that SS_n is low the most of time at the first simulation and at the end of the 10 clok cyle I forced ts value to 1 to end communication as expected	coverpoint of SS_n(SS_n_cp) and cross coverage with rx_valid(rx_valid_with_SS_N) and rx_data(rx_data_with_SS_N)	checking by using behavioral moddel for spi protocol
wrapper	the wapper must be act accroding spectification if it have an correct squences of mosies entered in the expected clock cycles in case that read data squence entered I expect the MISO give me data serially in 8 clk cycles after the squence is entered	randomisation of rst_n constrainting that its high most of time and the mosi squences to get the valid squences only	coverpoint mosi squence ((the_command_cover))and the rst_n(reset_coverage) which is called (reset_and_instructions)	checking by using behavioral moddel for spi wrapper

3. Coverage reports:

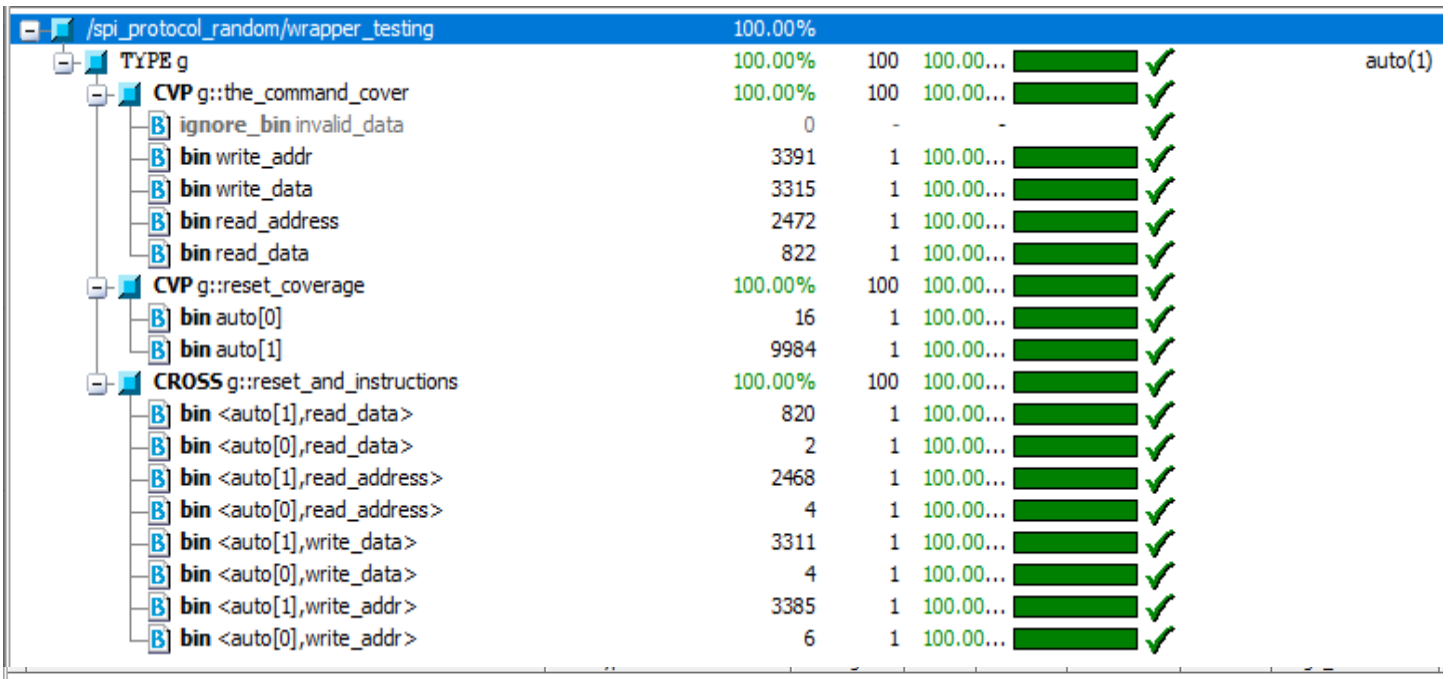
```
================================================================================
=== Instance: /\wrapper_tb#DUT /spislave/counter
=== Design Unit: work.up_counter
================================================================================
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         2         2         0   100.00%

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


Coverage Report by instance with details

================================================================================
=== Instance: /\wrapper_tb#DUT /spislave/shift_reg
=== Design Unit: work.SIPO
================================================================================
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         2         2         0   100.00%

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


Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                       2         2         0   100.00%

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


Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         46        46         0   100.00%

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

Toggle Coverage for instance /\wrapper_tb#DUT /spislave/shift_reg --

                                    Node      1H->0L      0L->1H                           "Coverage"
                                    ----------------------------------------------------------------
                                    MOSI           1           1                              100.00
                                    SS_n           1           1                              100.00
                                     clk           1           1                              100.00
                              rx_data[0-9]         1           1                              100.00
                                  tmp[9-0]         1           1                              100.00
```

```
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         20        20         0   100.00%

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

Toggle Coverage for instance /\wrapper_tb#DUT /spislave/counter --

                                          Node      1H->0L      0L->1H                          "Coverage"
                                          ----------------------------------------------------------------
                                           clk           1           1                             100.00
                                    counter[0-3]         1           1                             100.00
                                  counter_up[3-0]        1           1                             100.00
                                           rst_n         1           1                             100.00


================================================================================
=== Instance: /\wrapper_tb#DUT /spislave/shift_regII
=== Design Unit: work.PISO
================================================================================
Branch Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Branches                         4         4         0   100.00%

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


Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                       4         4         0   100.00%

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


Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         46        46         0   100.00%

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

Toggle Coverage for instance /\wrapper_tb#DUT /spislave/shift_regII --

                                          Node      1H->0L      0L->1H                          "Coverage"
                                          ----------------------------------------------------------------
                                           clk           1           1                             100.00
                                    counter[0-3]         1           1                             100.00
                                          dout           1           1                             100.00
                                       temp[7-0]         1           1                             100.00
                                    tx_data[0-7]         1           1                             100.00
                                        tx_valid         1           1                             100.00
```

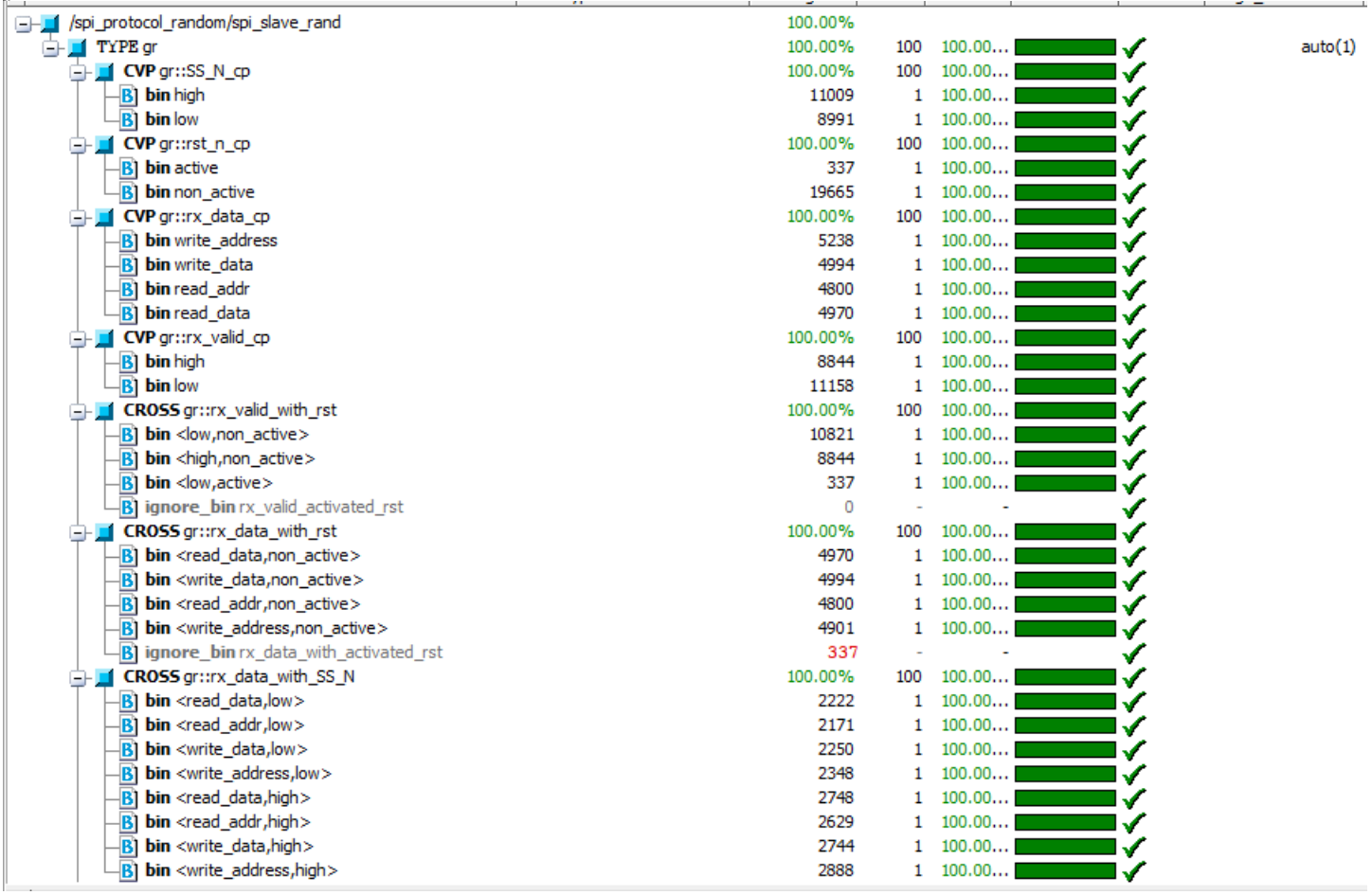
```
    Summary                       Bins      Hits    Misses  Coverage
    -------                       ----      ----    ------  --------
        FSM States                   5         5         0   100.00%
        FSM Transitions              8         8         0   100.00%
Statement Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Statements                   35        35         0   100.00%

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

Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                      96        96         0   100.00%

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

================================================================================
=== Instance: /\wrapper_tb#DUT
=== Design Unit: work.spi_wrapper
================================================================================
Toggle Coverage:
    Enabled Coverage              Bins      Hits    Misses  Coverage
    ----------------              ----      ----    ------  --------
    Toggles                         50        50         0   100.00%

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

Toggle Coverage for instance /\wrapper_tb#DUT  --

                                        Node      1H->0L      0L->1H                          "Coverage"
                                        ----------------------------------------------------------------
                                         clk           1           1                             100.00
                                        miso           1           1                             100.00
                                        mosi           1           1                             100.00
                                       rst_n           1           1                             100.00
                                rx_data[0-9]           1           1                             100.00
                                    rx_valid           1           1                             100.00
                                        ss_n           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)


Total Coverage By Instance (filtered view): 96.52%
```

Note: Coverage didn't reach 100% because of conditional coverage the all-false term.

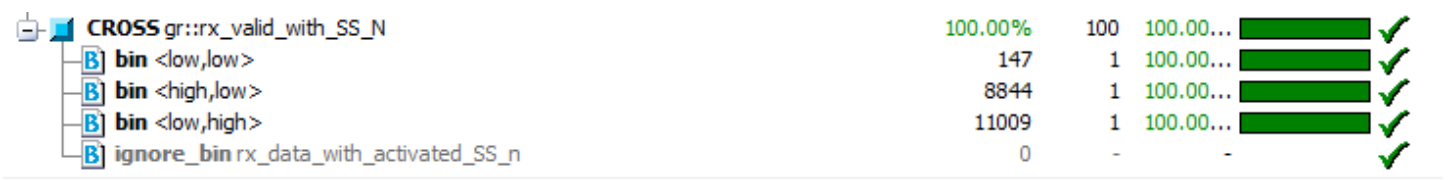
➢ Function Coverage:



/spi_protocol_random/wrapper_testing	100.00%			
TYPE g	100.00%	100	100.00...	auto(1)
CVP g::the_command_cover	100.00%	100	100.00...	
ignore_bin invalid_data	0	-	-	
bin write_addr	3391	1	100.00...	
bin write_data	3315	1	100.00...	
bin read_address	2472	1	100.00...	
bin read_data	822	1	100.00...	
CVP g::reset_coverage	100.00%	100	100.00...	
bin auto[0]	16	1	100.00...	
bin auto[1]	9984	1	100.00...	
CROSS g::reset_and_instructions	100.00%	100	100.00...	
bin <auto[1],read_data>	820	1	100.00...	
bin <auto[0],read_data>	2	1	100.00...	
bin <auto[1],read_address>	2468	1	100.00...	
bin <auto[0],read_address>	4	1	100.00...	
bin <auto[1],write_data>	3311	1	100.00...	
bin <auto[0],write_data>	4	1	100.00...	
bin <auto[1],write_addr>	3385	1	100.00...	
bin <auto[0],write_addr>	6	1	100.00...	



/spi_protocol_random/spi_slave_rand	100.00%			
TYPE gr	100.00%	100	100.00...	auto(1)
CVP gr::SS_N_cp	100.00%	100	100.00...	
bin high	11009	1	100.00...	
bin low	8991	1	100.00...	
CVP gr::rst_n_cp	100.00%	100	100.00...	
bin active	337	1	100.00...	
bin non_active	19665	1	100.00...	
CVP gr::rx_data_cp	100.00%	100	100.00...	
bin write_address	5238	1	100.00...	
bin write_data	4994	1	100.00...	
bin read_addr	4800	1	100.00...	
bin read_data	4970	1	100.00...	
CVP gr::rx_valid_cp	100.00%	100	100.00...	
bin high	8844	1	100.00...	
bin low	11158	1	100.00...	
CROSS gr::rx_valid_with_rst	100.00%	100	100.00...	
bin <low,non_active>	10821	1	100.00...	
bin <high,non_active>	8844	1	100.00...	
bin <low,active>	337	1	100.00...	
ignore_bin rx_valid_activated_rst	0	-	-	
CROSS gr::rx_data_with_rst	100.00%	100	100.00...	
bin <read_data,non_active>	4970	1	100.00...	
bin <write_data,non_active>	4994	1	100.00...	
bin <read_addr,non_active>	4800	1	100.00...	
bin <write_address,non_active>	4901	1	100.00...	
ignore_bin rx_data_with_activated_rst	337	-	-	
CROSS gr::rx_data_with_SS_N	100.00%	100	100.00...	
bin <read_data,low>	2222	1	100.00...	
bin <read_addr,low>	2171	1	100.00...	
bin <write_data,low>	2250	1	100.00...	
bin <write_address,low>	2348	1	100.00...	
bin <read_data,high>	2748	1	100.00...	
bin <read_addr,high>	2629	1	100.00...	
bin <write_data,high>	2744	1	100.00...	
bin <write_address,high>	2888	1	100.00...	

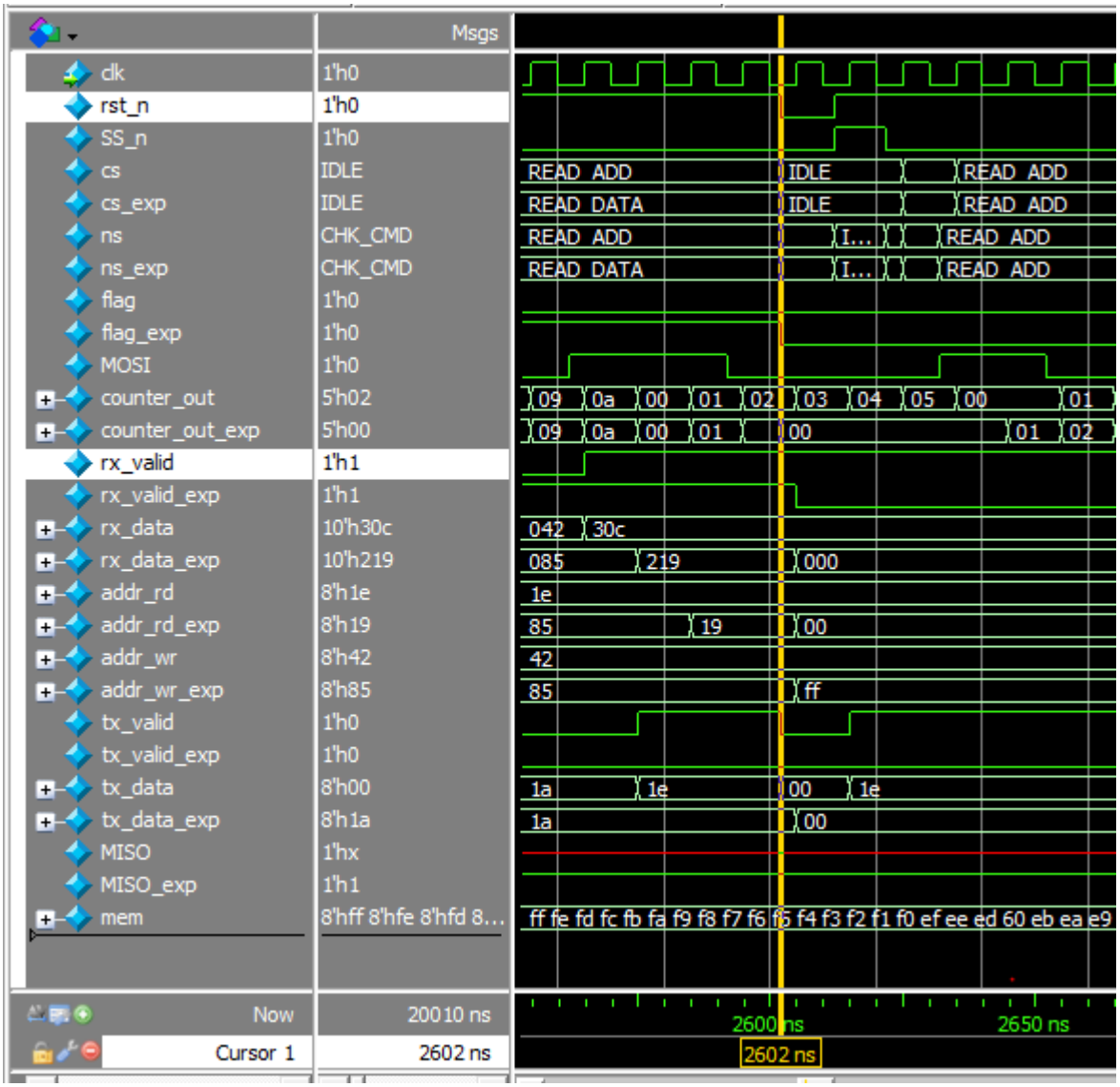


CROSS gr::rx_valid_with_SS_N	100.00%	100	100.00...	
bin <low,low>	147	1	100.00...	
bin <high,low>	8844	1	100.00...	
bin <low,high>	11009	1	100.00...	
ignore_bin rx_data_with_activated_SS_n	0	-	-	

4. Bugs report:

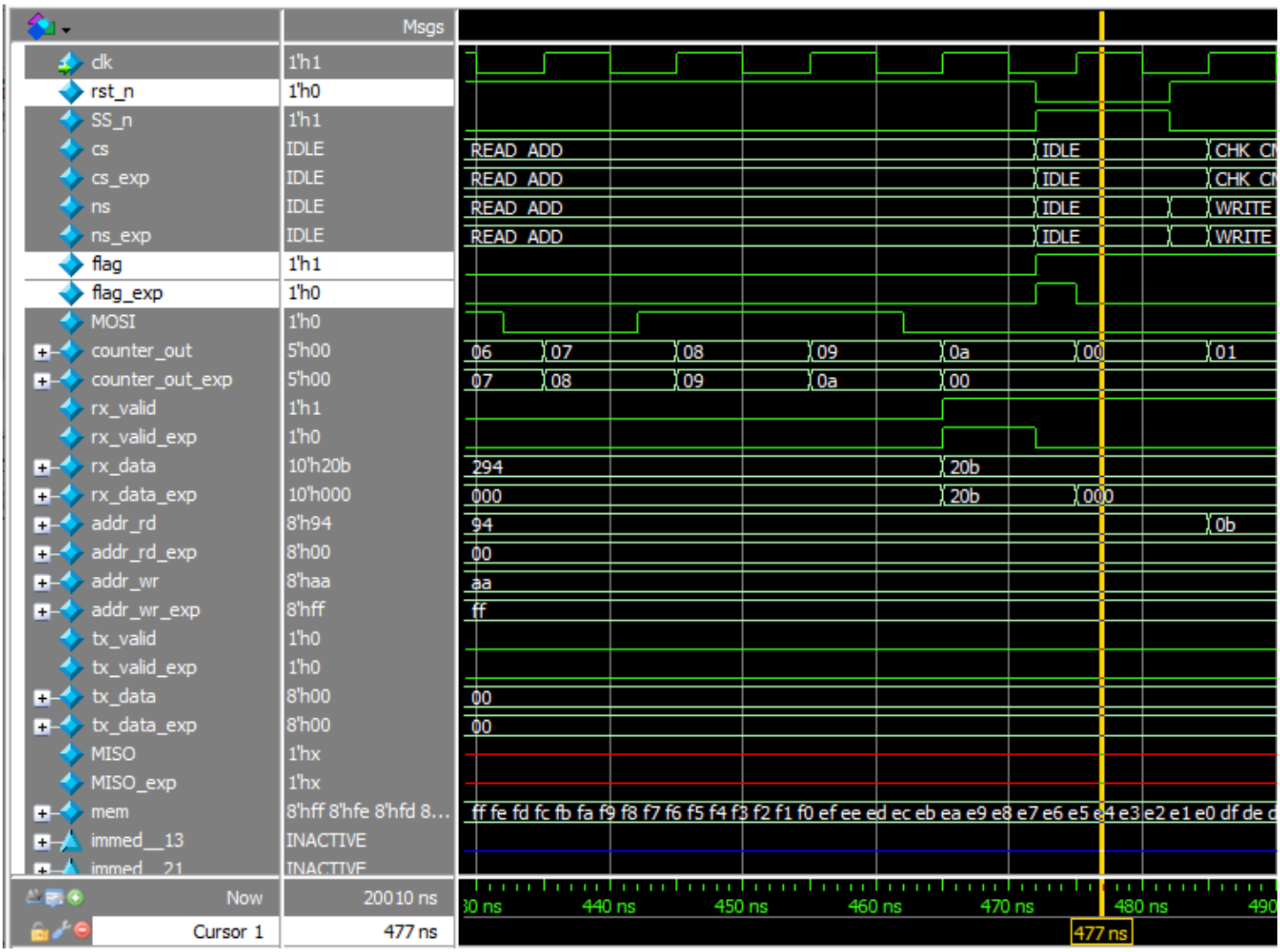
a) Timing delays in the SPI wrapper operations specially in READ_DATA case.

b) In case of active reset rx_valid does not change (remains the same).



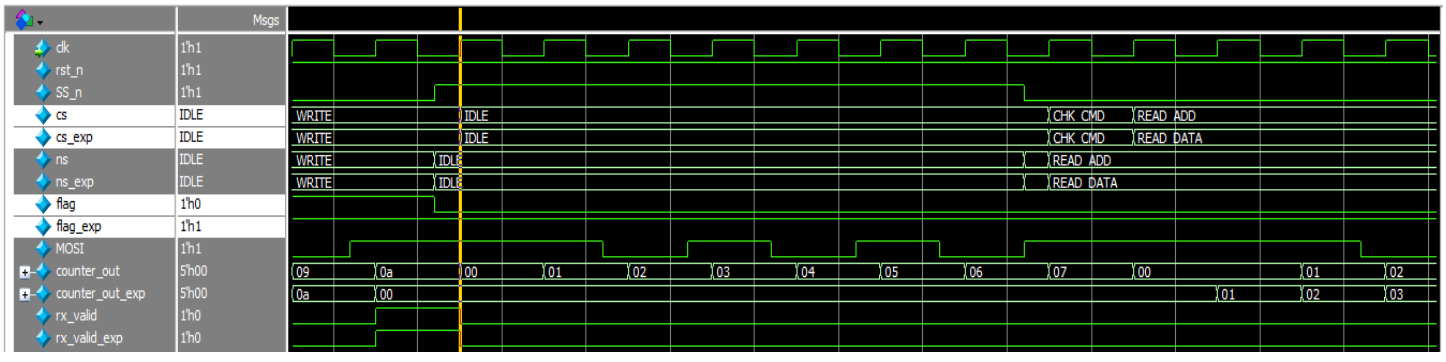
Snippet shows the stability of the rx_valid flag even during active reset

c) During asserted reset the flag which indicates that an address was sent does not get cleared.



Snippet shows the stability of the address flag even during active reset

d) If the current state is WRITE and master ends the communication the flag is cleared. Which causes a problem in the next READ operation induces wrong READ_ADD operation.



Snippet shows the false clear of the flag after WRITE operation

e) Checking to clear the counter in the design takes place on the current state which makes the counter clears but the next clock cycle instead it should check for the next state if it is equal CHK_CMD to clear the counter right away and be able to assert rx_valid and access memory before the communication ends.

5. Code snippets

a) Coverage & cover points:

```
   covergroup gr;
   SS_N_cp:coverpoint SS_n{
      bins high={1};
      bins low={0};
   }
   rst_n_cp:coverpoint rst_n{
      bins active={0};
      bins non_active={1};
   }
   rx_data_cp:coverpoint rx_data[9:8]
   {
bins write_address={2'b00};
bins write_data={2'b01};
bins read_addr={2'b10};
bins read_data={2'b11};
   }

   rx_valid_cp:coverpoint rx_valid{
      bins high={1};
      bins low={0};
   }
rx_valid_with_rst: cross rx_valid_cp,rst_n_cp{   ignore_bins rx_valid_activated_rst=binsof(rx_valid_cp.high)&&binsof(rst_n_cp.active);

}

rx_data_with_rst: cross  rx_data_cp,rst_n_cp{
      ignore_bins rx_data_with_activated_rst=binsof(rx_data_cp)&&binsof(rst_n_cp.active);

}
rx_data_with_SS_N: cross rx_data_cp,SS_N_cp;
rx_valid_with_SS_N: cross rx_valid_cp,SS_N_cp;
   endgroup
```

```
covergroup g;
the_command_cover:coverpoint data_holder[10:8] {
bins write_addr={3'b000};
bins write_data={3'b001};
bins read_address={3'b110};
bins read_data={3'b111};
ignore_bins invalid_data={3'b101,3'b100,3'b010,3'b011};
}
reset_coverage: coverpoint rst_n;
reset_and_instructions: cross reset_coverage,the_command_cover;
endgroup
```

b) Assertions file:

```
module spi_sva(wrapper_if.MONITOR wrap_if);

    always_comb begin
        if(!wrap_if.rst_n) begin
            P2: assert final (wrap_if.flag == 0) ;
            P3: assert final (wrap_if.cs ==IDLE) ;
        end
    end
    property valid_clear;
        @(posedge wrap_if.clk) !wrap_if.rst_n |=>  ( wrap_if.rx_valid ==0);
    endproperty

    property comm_end ;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n) (wrap_if.SS_n) |=> (wrap_if.cs == IDLE) ;
    endproperty

    property comm_start ;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n) (wrap_if.cs==IDLE && ! wrap_if.SS_n ) |=> (wrap_if.cs==CHK_CMD ) ;
    endproperty

    property Wr_transition ;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n || wrap_if.SS_n) (wrap_if.cs==CHK_CMD && !wrap_if.MOSI ) |=> (wrap_if.cs==WRITE && wrap_if.counter_out ==0) ;
    endproperty

    property write_inc ;
        logic [4:0] prev_count;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n || wrap_if.SS_n) (wrap_if.cs==WRITE &&wrap_if.counter_out!=0 , prev_count= wrap_if.counter_out) |=>
         (wrap_if.counter_out == (prev_count+4'b0001)) ;
    endproperty

    property write_valid;
        logic [9:0] word ;

        @(posedge wrap_if.clk) disable iff (!wrap_if.rst_n || wrap_if.SS_n)
          (
            (wrap_if.cs == WRITE && wrap_if.counter_out == 0,word= 10'b0) |=> (1,word = {word[8:0], $past(wrap_if.MOSI)}) [*0:$]
            ##2 (word == $past(wrap_if.rx_data))
          );
    endproperty
    property read_add_valid;
       logic [9:0] word_rd ;

       @(posedge wrap_if.clk) disable iff (!wrap_if.rst_n || wrap_if.SS_n)
         (
           (wrap_if.cs == WRITE && wrap_if.counter_out == 0,word_rd= 10'b0) |=> (1,word_rd = {word_rd[8:0], $past(wrap_if.MOSI)}) [*0:$]
           ##2 (word_rd == $past(wrap_if.rx_data))
         );
    endproperty

    property write_add ;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n) (wrap_if.cs==WRITE && wrap_if.rx_valid && !wrap_if.rx_data[9] && !wrap_if.rx_data[8]) |=> (wrap_if.addr_wr== wrap_if.rx_data[7:0]);
    endproperty

    property write_data;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n) (wrap_if.cs==WRITE && wrap_if.rx_valid && !wrap_if.rx_data[9] && wrap_if.rx_data[8]) |=>
         (DUT.mem.mem[wrap_if.addr_wr]== wrap_if.rx_data[7:0]) ;
    endproperty

    property read_add_trans ;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n || wrap_if.SS_n) (wrap_if.cs== CHK_CMD && wrap_if.MOSI && !wrap_if.flag) |=> (wrap_if.cs==READ_ADD);
    endproperty

    property read_add ;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n ) (wrap_if.cs==READ_ADD && wrap_if.rx_valid && wrap_if.rx_data[9] && !wrap_if.rx_data[8]) |=> (wrap_if.addr_rd== wrap_if.rx_data[7:0])
    endproperty

    property read_data_trans ;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n || wrap_if.SS_n) (wrap_if.cs== CHK_CMD && wrap_if.MOSI && wrap_if.flag) |=> (wrap_if.cs==READ_DATA);
    endproperty

    property read_data_seq;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n || wrap_if.SS_n) (wrap_if.cs== READ_DATA && wrap_if.counter_out==0 ) |=> ##[9:10]$rose(wrap_if.rx_valid) ##1 $rose(wrap_if.tx_valid)  ;
    endproperty

    property flag_set ;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n ) (wrap_if.cs == READ_ADD && wrap_if.rx_valid==1) |=> $rose($past(wrap_if.flag));
    endproperty

    property flag_clear;
        @(posedge wrap_if.clk) disable iff(!wrap_if.rst_n) (wrap_if.cs == READ_DATA && wrap_if.ns == IDLE ) |=> $fell($past(wrap_if.flag));
    endproperty
```