

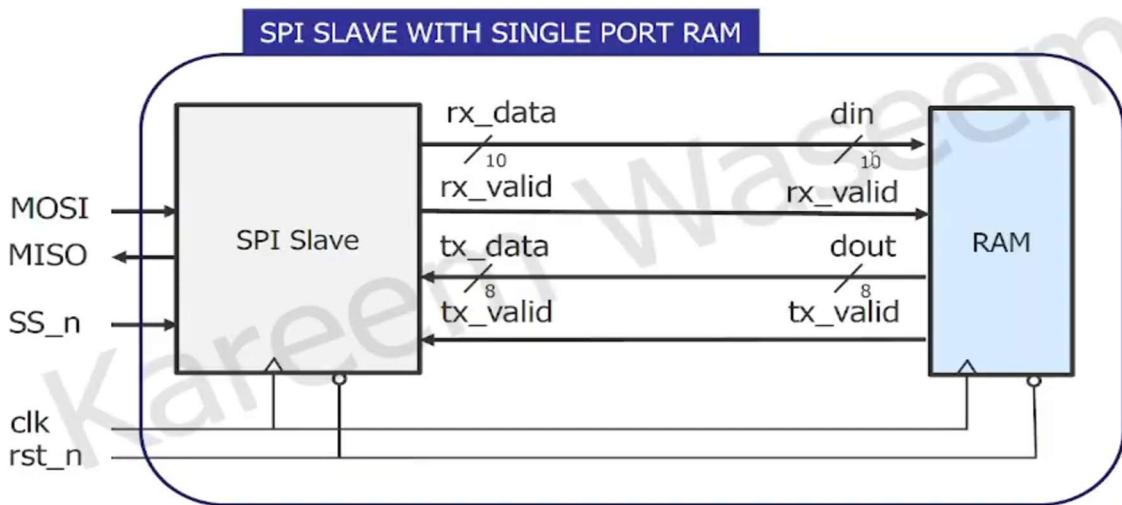
SPI SLAVE WITH SINGLE PORT RAM

Name : Ahmed Farag

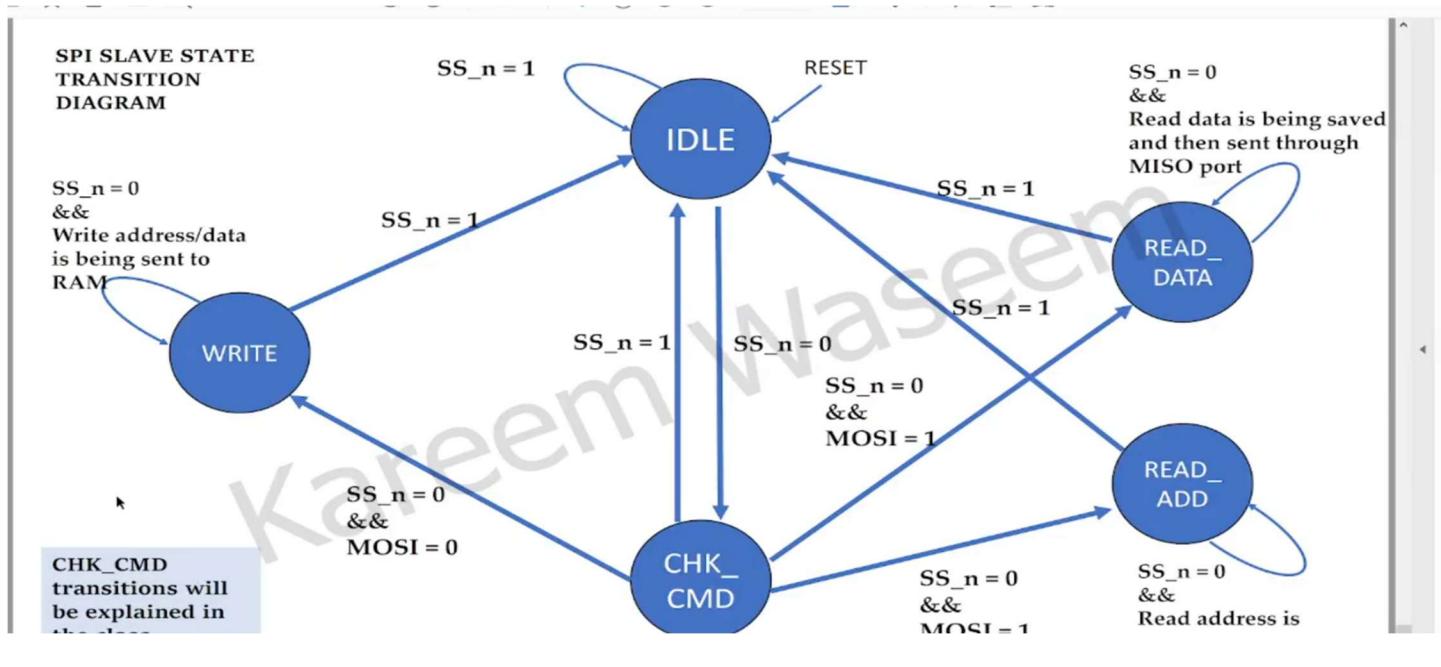
Overview: SPI Slave with Single-Port RAM

The **SPI Slave with Single-Port RAM** module is a digital system designed to facilitate serial communication between a master device (such as a microcontroller) and a memory storage block. This design uses the **Serial Peripheral Interface (SPI)** protocol and an internal **single-port RAM** for temporary data storage.

System Architecture:



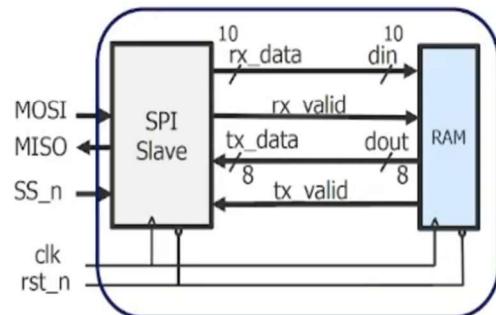
State transition diagram:



Operations:

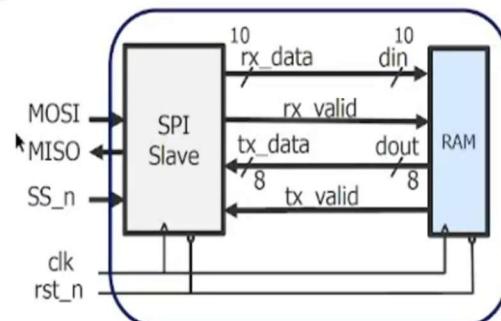
RAM Write Command – Write Address

1. Master will start the write command by sending the write address value, $rx_data[9:8] = din[9:8] = 2'b00$
2. $SS_n = 0$ to tell the SPI Slave that the master will begin communication
3. SPI Slave check the first received bit on MOSI port '0' which is a control bit to let the slave determine which operation will take place "write in this case". SPI Slave then expects to receive 10 more bits, the first 2 bits are "00" on two clock cycles and then the wr_address will be sent on 8 more clock cycles
4. Now the data is converted from serial "MOSI" to parallel after writing the $rx_data[9:0]$ bus
5. rx_valid will be HIGH to inform the RAM that it should expect data on din bus
6. din takes the value of rx_data
7. RAM checks on $din[9:8]$ and find that they hold "00"
8. RAM stores $din[7:0]$ in the internal write address bus
9. $SS_n = 1$ to end communication from Master side



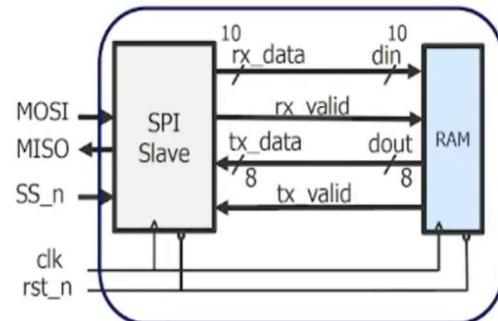
RAM Write Command – Write Data

1. Master will continue the write command by sending the write data value, $rx_data[9:8] = din[9:8] = 2'b01$
2. $SS_n = 0$ to tell the SPI Slave that the master will begin communication
3. SPI Slave check the first received bit on MOSI port '0' which is a control bit to let the slave determine which operation will take place "write in this case". SPI Slave then expects to receive 10 more bits, the first 2 bits are "01" on two clock cycles and then the wr_data will be sent on 8 more clock cycles
4. Now the data is converted from serial "MOSI" to parallel after writing the $rx_data[9:0]$ bus
5. rx_valid will be HIGH to inform the RAM that it should expect data on din bus
6. din takes the value of rx_data
7. RAM checks on $din[9:8]$ and find that they hold "01"
8. RAM stores $din[7:0]$ in the RAM with wr_address previously held
9. $SS_n = 1$ to end communication from Master side



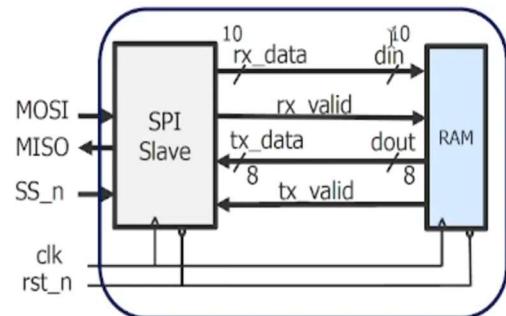
RAM Read Command – Read Address

1. Master will start the write command by sending the read address value, $rx_data[9:8] = din[9:8] = 2'b10$
2. $SS_n = 0$ to tell the SPI Slave that the master will begin communication
3. SPI Slave check the first received bit on MOSI port '1' which is a control bit to let the slave determine which operation will take place "read in this case". SPI Slave then expects to receive 10 more bits, the first 2 bits are "10" on two clock cycles and then the $rd_address$ will be sent on 8 more clock cycles
4. Now the data is converted from serial "MOSI" to parallel after writing the $rx_data[9:0]$ bus
5. rx_valid will be HIGH to inform the RAM that it should expect data on din bus
6. din takes the value of rx_data
7. RAM checks on $din[9:8]$ and find that they hold "10"
8. RAM stores $din[7:0]$ in the internal read address bus
9. $SS_n = 1$ to end communication from Master side



RAM Read Command – Read Data

1. Master will start the write command by sending the read address value, $rx_data[9:8] = din[9:8] = 2'b11$
2. $SS_n = 0$ to tell the SPI Slave that the master will begin communication
3. SPI Slave check the first received bit on MOSI port '1' which is a control bit to let the slave determine which operation will take place "read in this case". SPI Slave then expects to receive 10 more bits, the first 2 bits are "11" on two clock cycles and then dummy data will be sent and ignored since the master is waiting for the data to be sent from slave side
4. Now the data is converted from serial "MOSI" to parallel after writing the $rx_data[9:0]$ bus
5. din takes the value of rx_data
6. RAM reads $din[9:8]$ and find that they hold "11"
7. RAM will read from the memory with $rd_address$ previously held
8. RAM will assert tx_valid to inform slave that data out is ready
9. Slave reads tx_data and convert it into serial out data on MISO port
10. $SS_n = 1$, Master ends communication after receiving data
"8 clock cycles"



1-RTL CODE:

1-RAM module:

```
V SPI_PROJECT.V > RAM
1  module RAM #(parameter MEM_DEPTH=256,ADDR_SIZE=8)
2    (input [9:0]din
3     ,input clk,rst_n,rx_valid,
4     output reg tx_valid,
5     output reg [7:0]dout);
6    reg [ADDR_SIZE-1:0] mem [0:MEM_DEPTH-1];
7    reg [7:0]addr_RD,addr_WR;
8
9    always@(posedge clk)begin
10      tx_valid<=0;
11      if(~rst_n)begin
12        dout<=0;
13        tx_valid<=0;
14        addr_RD<=0;
15        addr_WR<=0;
16      end
17      else if (rx_valid) begin
18        case(din[9:8])
19          2'h0 : addr_WR<=din[7:0];
20          2'h1 : mem[addr_WR]<=din[7:0];
21          2'h2 : addr_RD<=din[7:0];
22          2'h3 :begin
23            dout<=mem[addr_RD];
24            tx_valid<=1;
25          end
26
27        endcase
28      end
29    end
30  endmodule
```

2-SPI SLAVE:

```
32 module SPI_SLAVE #(parameter IDLE=0,READ_DATA=1,READ_ADD=2,CHK_CMD=3,WRITE=4)
33   (input clk,rst_n,tx_valid,SS_n,MOSI,
34   input [7:0] tx_data,
35   output reg rx_valid,MISO,
36   output reg[9:0]rx_data);
37   (* fsm_encoding = "gray" *)
38   reg [2:0]cs,ns;
39   reg READ_FLAG;
40   reg [3:0] count;
41   reg[3:0] count2;
42   reg [7:0] MISO_BUS;
43   reg [9:0] MOSI_BUS;
44   always@(posedge clk)begin
45     if(~rst_n)begin
46       cs<=IDLE;
47     end
48     else
49       cs<=ns;
50   end
51
52   always@(*)begin
53     case (cs)
54       IDLE : begin
55         if(SS_n==0)
56           ns=CHK_CMD;
57         else
58           ns=IDLE;
59       end
60       CHK_CMD : begin
61         if(SS_n)
62           ns=IDLE;
63         else if(SS_n==0&&MOSI==0)begin
64           ns=WRITE;
65         end
66         else if (SS_n==0&&MOSI==1&&READ_FLAG==0)begin
67           ns=READ_ADD;
68         end
69       end
70
71       else if(SS_n==0&&MOSI==1&&READ_FLAG==1)begin
72         ns=READ_DATA;
```

```
72         |     ns=READ_DATA;
73     end
74     else
75         |     ns=CHK_CMD;
76     end
77     WRITE : begin
78         if(SS_n)
79             |     ns=IDLE;
80         else
81             |     ns=WRITE;
82     end
83     READ_ADD : begin
84         if(SS_n)
85             |     ns=IDLE;
86         else
87             |     ns=READ_ADD;
88     end
89     READ_DATA : begin
90         if(SS_n)
91             |     ns=IDLE;
92         else
93             |     ns=READ_DATA;
94     end
95     endcase
96 end
97
98 always@(posedge clk)begin
99     if(~rst_n)begin
100         |     count<=0;
101         |     rx_valid<=0;
102         |     rx_data<=0;
103         |     MISO<=0;
104         |     count2<=7;
105         |     READ_FLAG<=0;
106         |     MISO_BUS<=0;
107         |     MOSI_BUS<=0;
108     end
109     case(cs)
110         |     IDLE : begin
111             |         count<=0;
```

indexed objects

```
109         case(cs)
110             IDLE : begin
111                 count<=0;
112                 rx_valid<=0;
113                 count2<=7;
114                 MISO<=0;
115             end
116
117             WRITE : begin
118                 if(count<10)begin
119                     MOSI_BUS<={MOSI_BUS[9:0],MOSI};
120                     rx_valid<=0;
121                     count<=count+1;
122                 end
123                 else begin
124                     rx_data<=MOSI_BUS;
125                     rx_valid<=1;
126                 end
127             end
128             READ_ADD : begin
129                 if(count<10)begin
130                     MOSI_BUS<={MOSI_BUS[9:0],MOSI};
131                     rx_valid<=0;
132                     count<=count+1;
133                 end
134                 else begin
135                     rx_data<=MOSI_BUS;
136                     rx_valid<=1;
137                     READ_FLAG<=1;
138                 end
139             end
140
141             READ_DATA : begin
142                 if(count<10)begin
143                     MOSI_BUS<= {MOSI_BUS[9:0],MOSI};
144                     count<=count+1;
145                 end
146                 else if (count==10)begin
147                     rx_valid<=1;
148                     rx_data<=MOSI_BUS;
```

```

146         else if (count==10)begin
147             rx_valid<=1;
148             rx_data<=MOSI_BUS;
149             count<=11;
150         end
151         else if(tx_valid==1 && count<12)begin
152             MISO_BUS<=tx_data;
153             count<=count+1;
154             rx_valid<=0;
155         end
156         else if (count2>=0 && count2<4'b1111) begin
157             MISO<=MISO_BUS[count2];
158             count2<=count2-1;
159         end
160         else
161             READ_FLAG<=0;      //FLAG ADD / DATA
162     end
163 endcase
164
165 end
166 endmodule
167

```

3-SPI Wrapper:

```

169 module SPI_wrapper(input clk,rst_n,MOSI,SS_n,output MISO);
170 wire [9:0]rx_data;
171 wire rx_valid;
172 wire [7:0]tx_data;
173 wire tx_valid;
174
175 RAM RAM1(.clk(clk),.rst_n(rst_n),.din(rx_data),.rx_valid(rx_valid),.dout(tx_data),.tx_valid(tx_valid));
176
177 SPI_SLAVE SPI(.clk(clk),.rst_n(rst_n),.MOSI(MOSI),.SS_n(SS_n),.MISO(MISO),.tx_data(tx_data),.tx_valid(tx_valid)
178 ,.rx_data(rx_data),.rx_valid(rx_valid));
179 endmodule
180

```

2-TESTBENCH:

```
177
178 module tb_SPI();
179 reg clk,rst_n,MOSI,SS_n;
180 wire MISO;
181
182 SPI_wrapper dut (.clk(clk),.rst_n(rst_n),.MOSI(MOSI),.SS_n(SS_n),.MISO(MISO));
183
184 initial begin
185     clk=0;
186     forever
187         #1 clk=~clk;
188 end
189 integer i;
190 initial begin
191     for (i=0; i<256; i=i+1) begin
192         dut.RAM1.mem[i]=8'h0;
193     end
194     dut.RAM1.mem[15]=8'h14;
195 end
196
197 initial begin
198     rst_n=0;
199     MOSI=0;
200     SS_n=1;
201     repeat(2)@(negedge clk);           //rst_chk
202
203     rst_n=1;
204     SS_n=0;
205     @(negedge clk);
206
207     MOSI=0;
208     repeat(3)@(negedge clk);
209     MOSI=1;
210     @(negedge clk);
211     MOSI=0;
212     repeat(7)@(negedge clk);      //WRITE ADDRESS MEM[128]
213     SS_n=1;
214
215     @(negedge clk);
216     SS_n=0;
217     @(negedge clk);
218     MOSI=0;
219     @(negedge clk);
220     MOSI=0;
221     @(negedge clk);
222     MOSI=1;
223     @(negedge clk);
224     MOSI=1;
225     repeat(8)@(negedge clk);      //WRITE    FF
--
```

```
    @(negedge clk);
    MOSI=1;
    repeat(8)@(negedge clk);      //WRITE      FF
    SS_n=1;
    @(negedge clk);

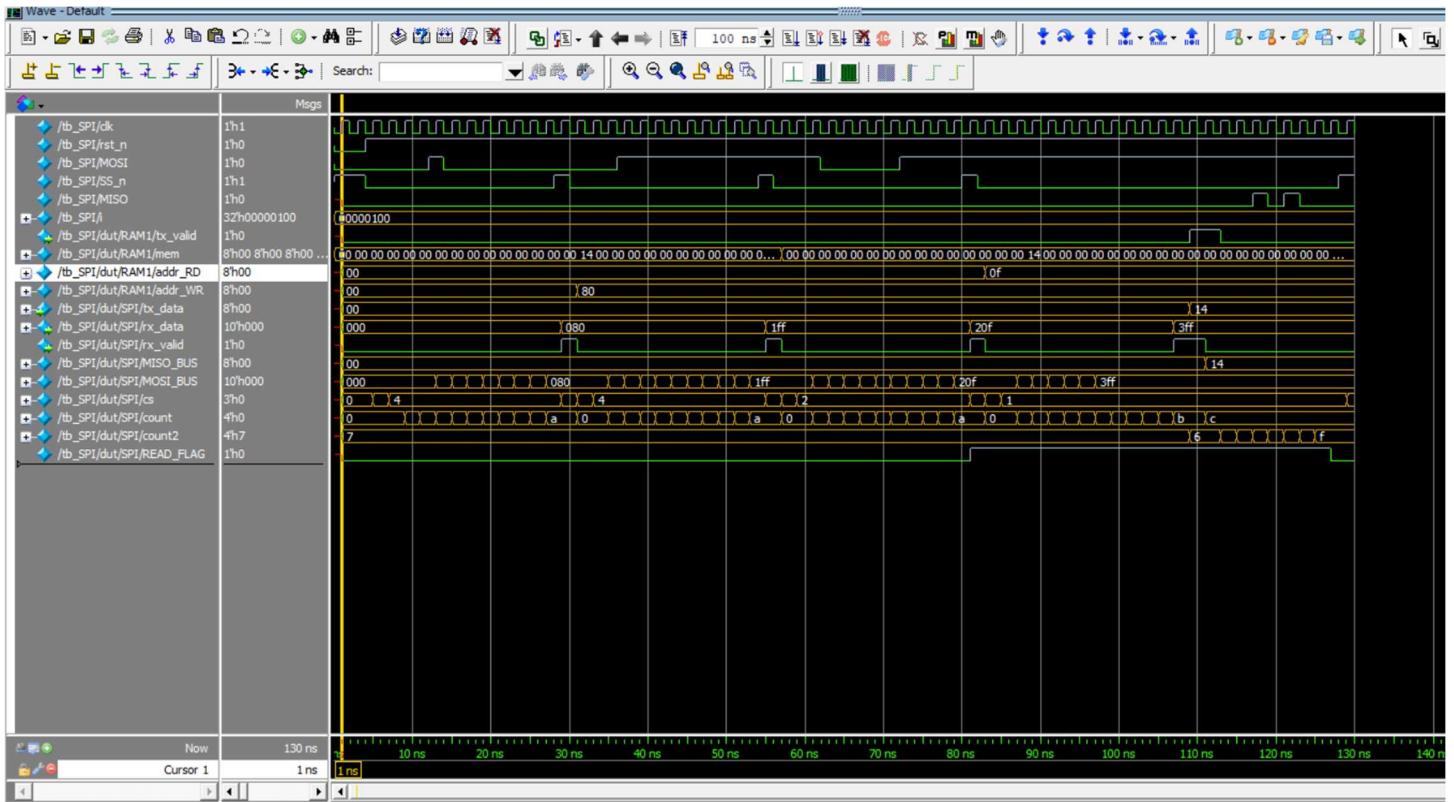
    SS_n=0;
    @(negedge clk);
    MOSI=1;
    repeat(2)@(negedge clk);
    MOSI=0;
    @(negedge clk);
    MOSI=0;
    repeat(4)@(negedge clk);
    MOSI=1;
    repeat(4)@(negedge clk);    //READ ADDRESS    MEM[15]
    SS_n=1;
    @(negedge clk);

    SS_n=0;
    @(negedge clk);
    MOSI=1;
    @(negedge clk);
    MOSI=1;
    repeat(21)@(negedge clk);   //READ DATA H'14
    SS_n=1;
    @(negedge clk);

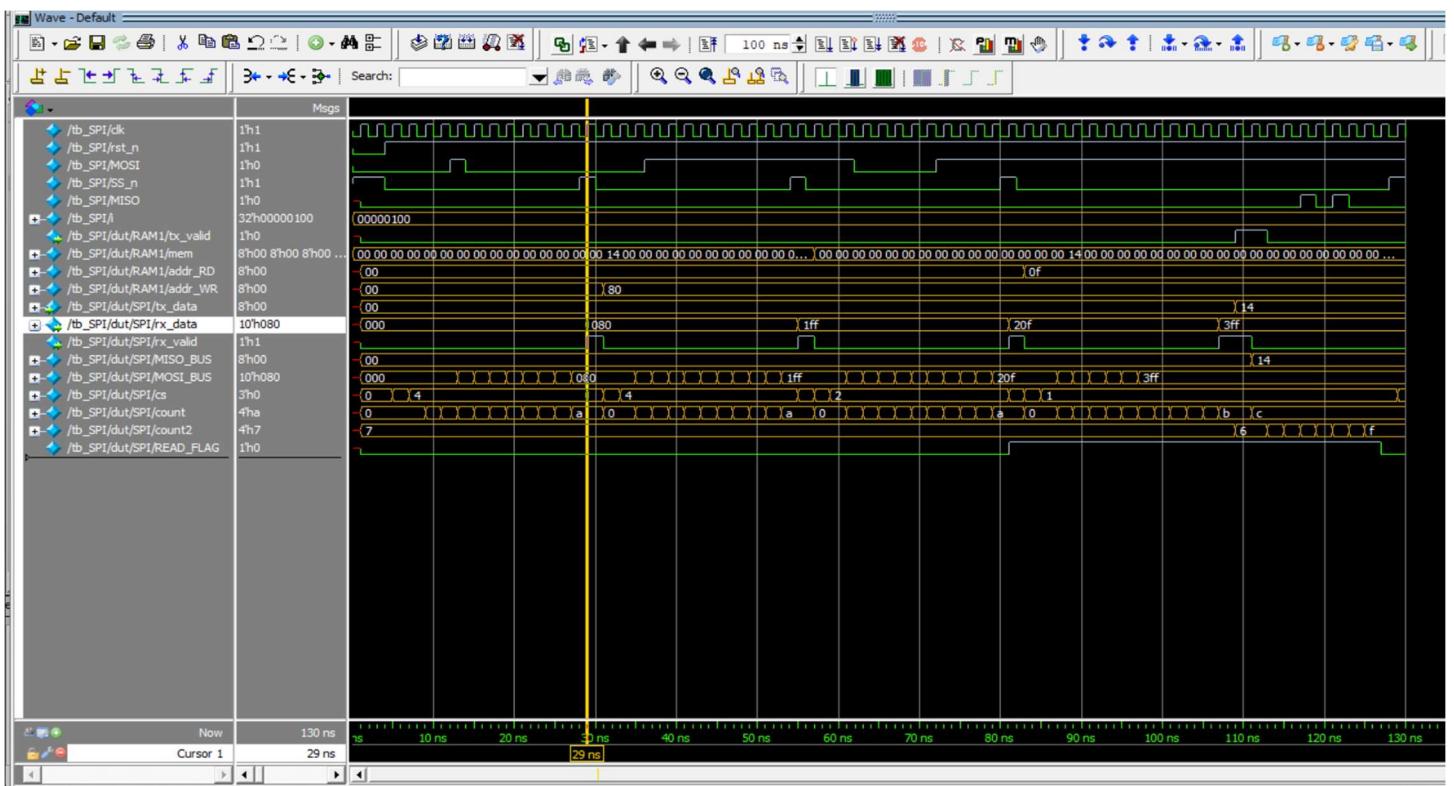
    $stop;
end
endmodule
```

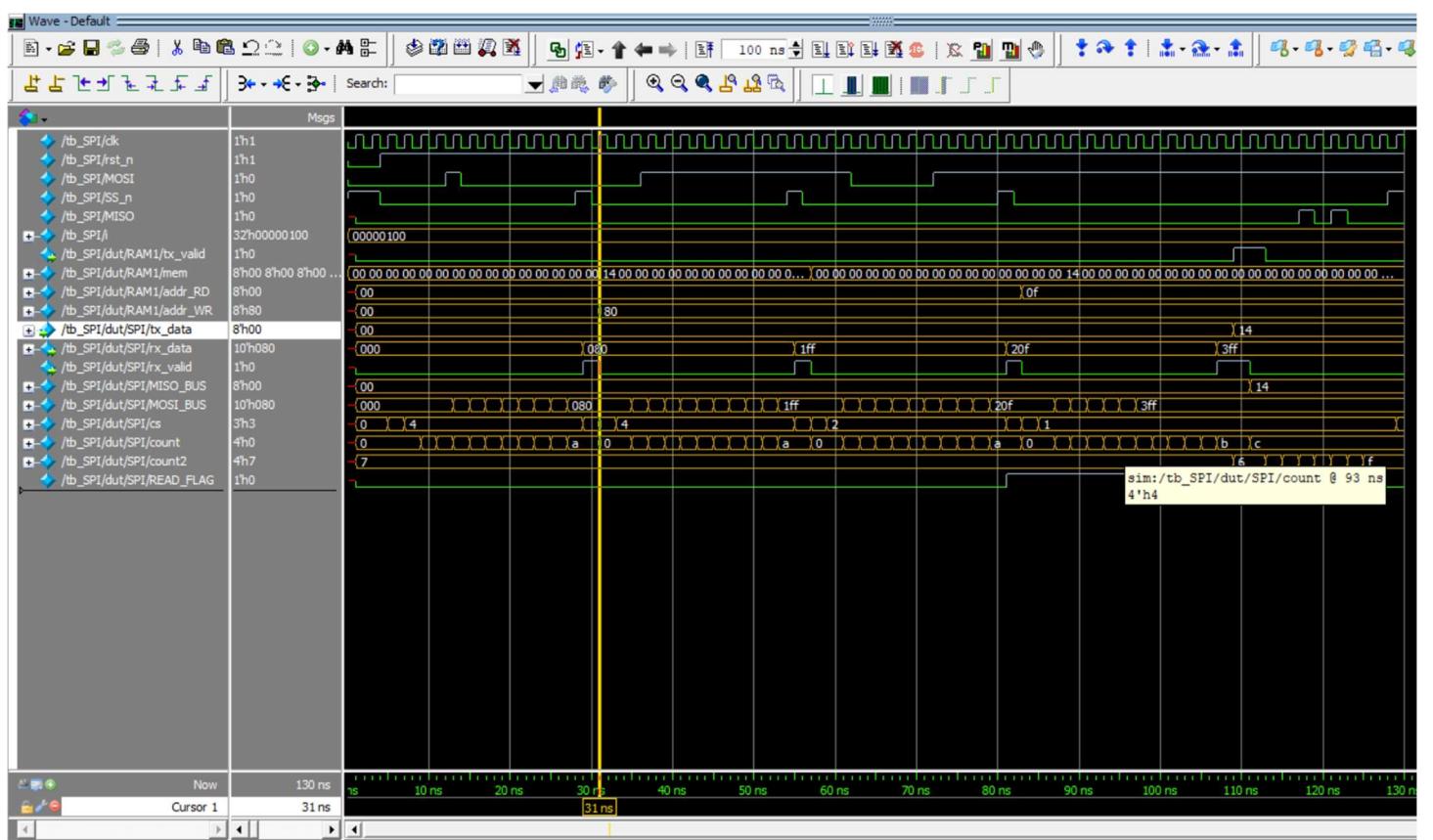
2-WAVE:

1rst check:

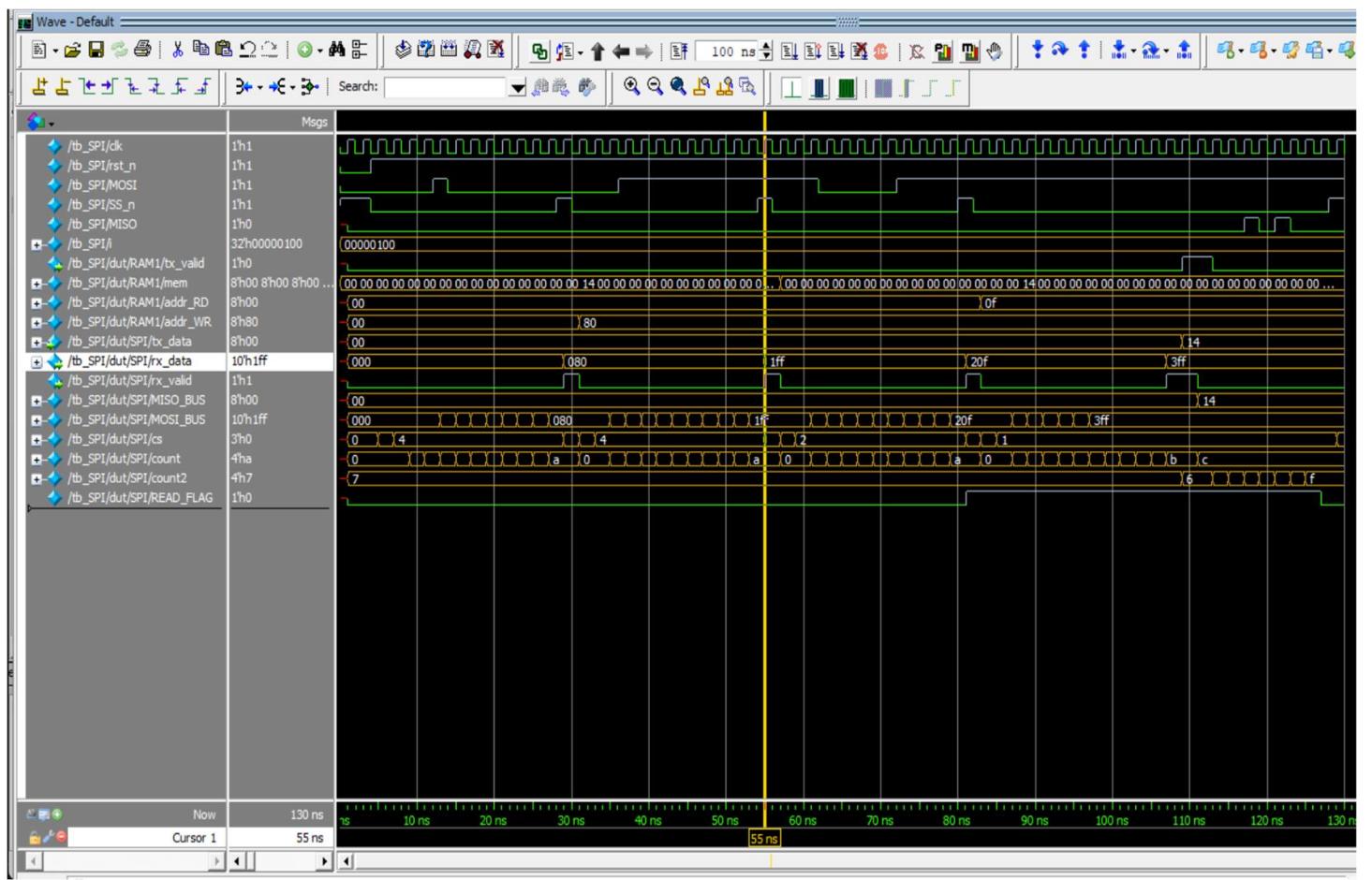


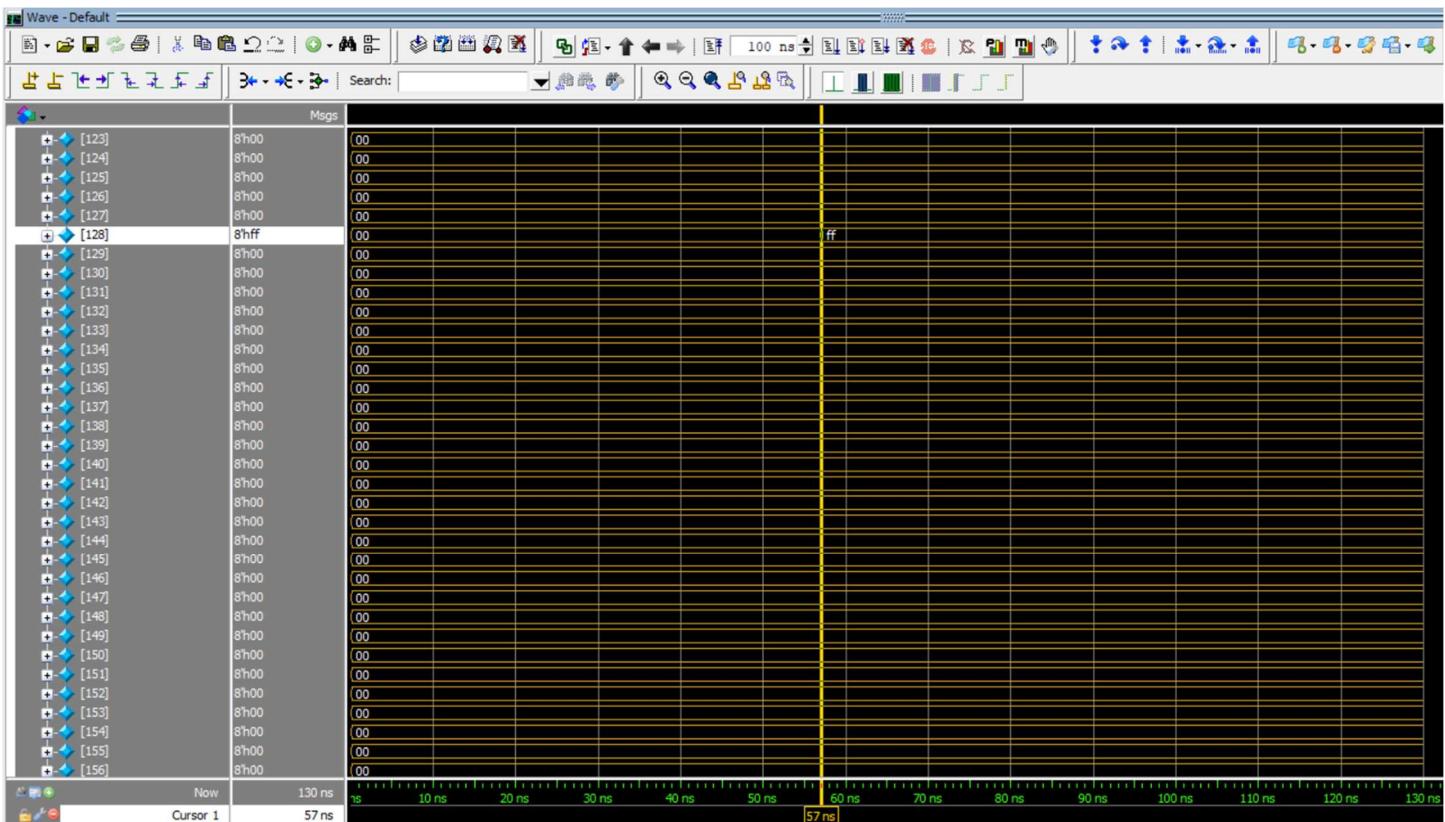
2-write address:



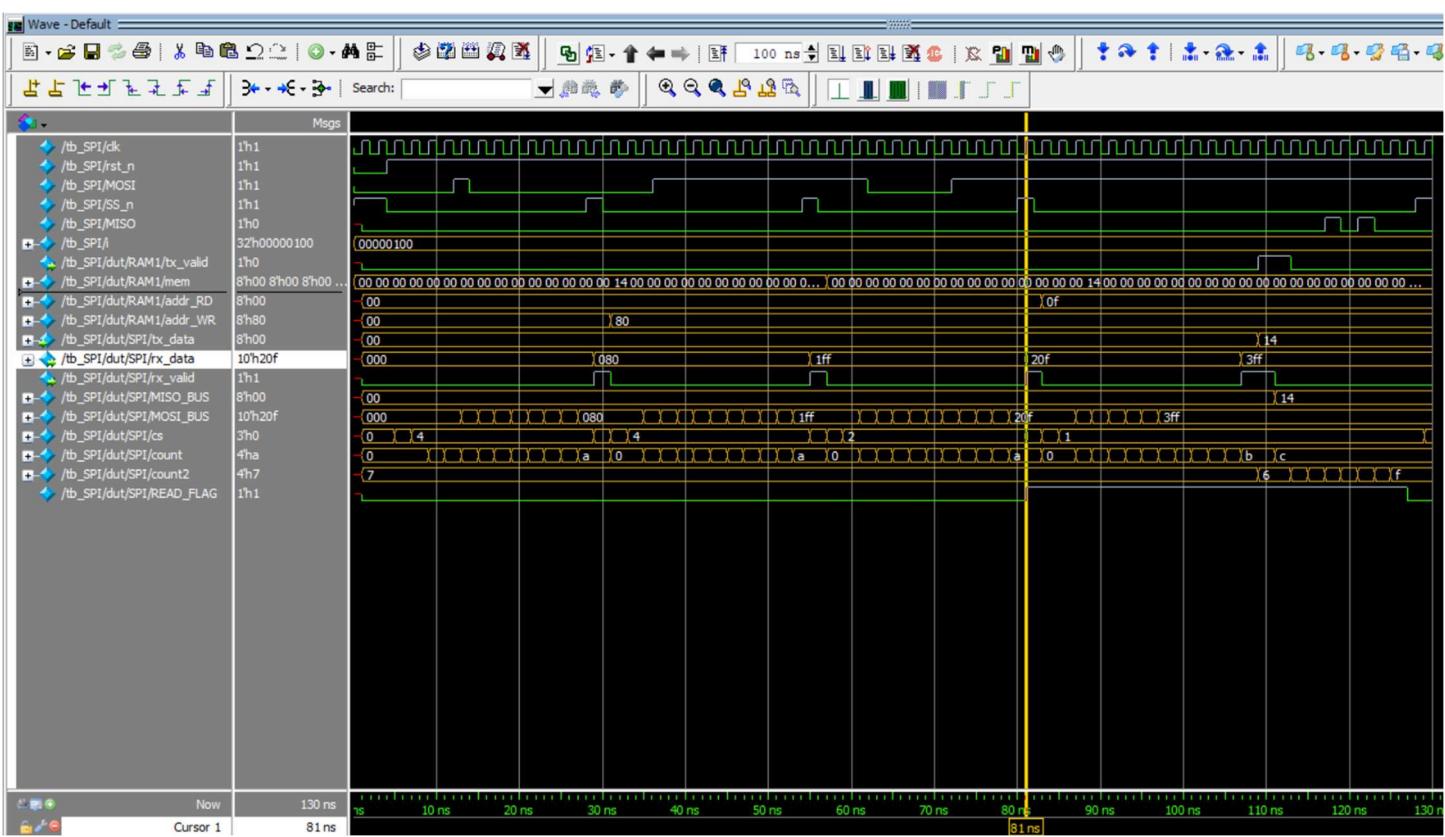


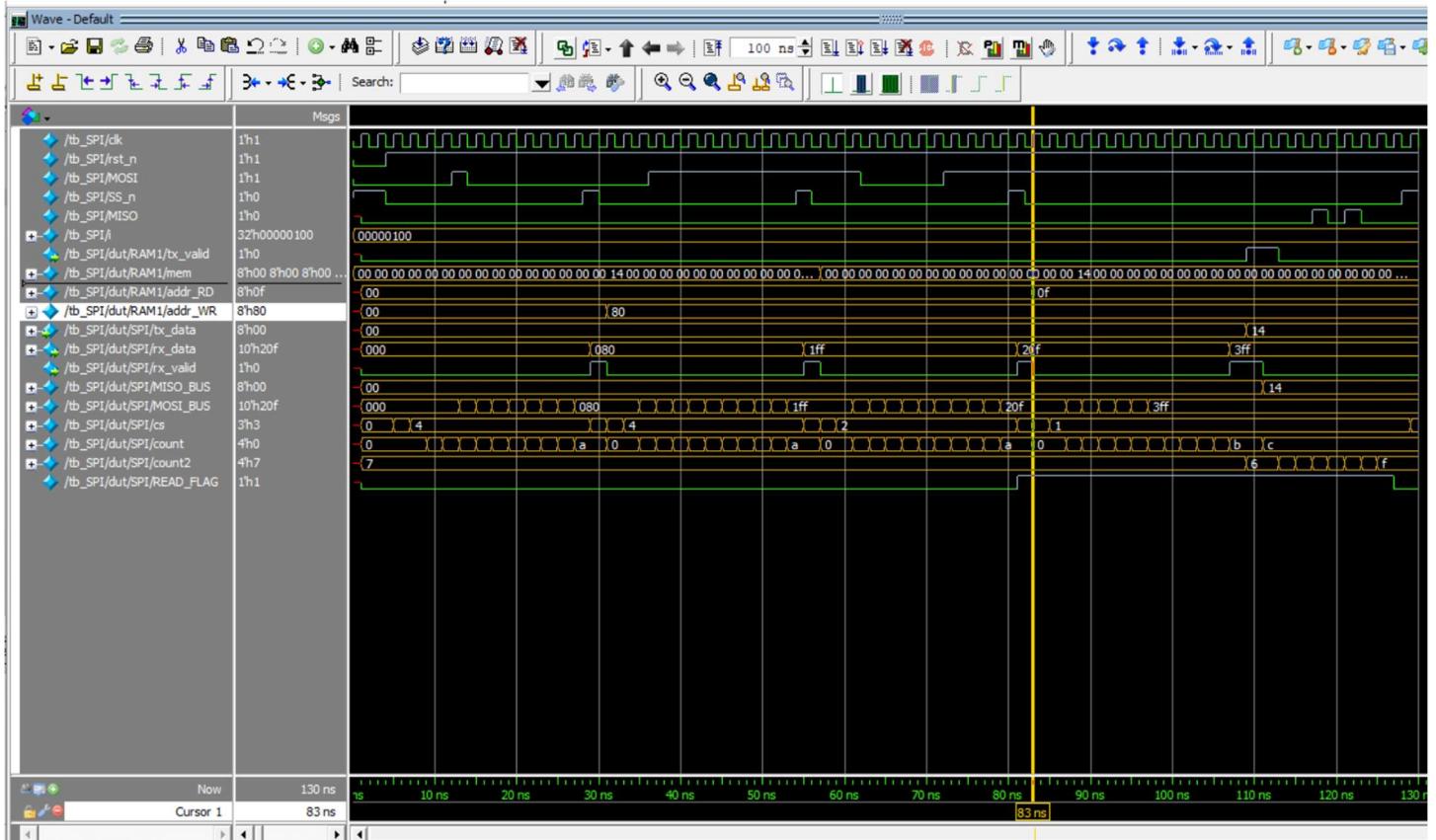
3-write data:



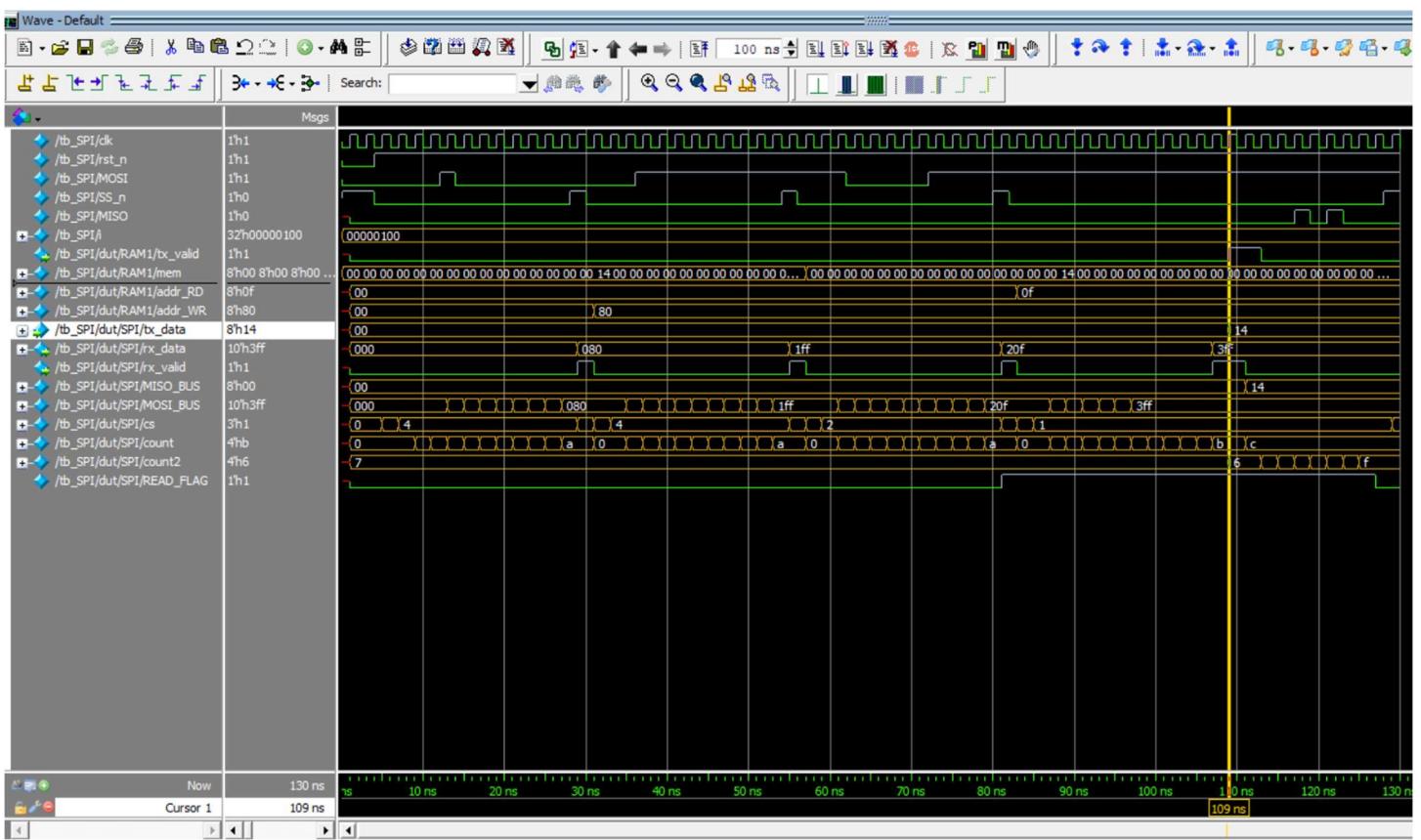


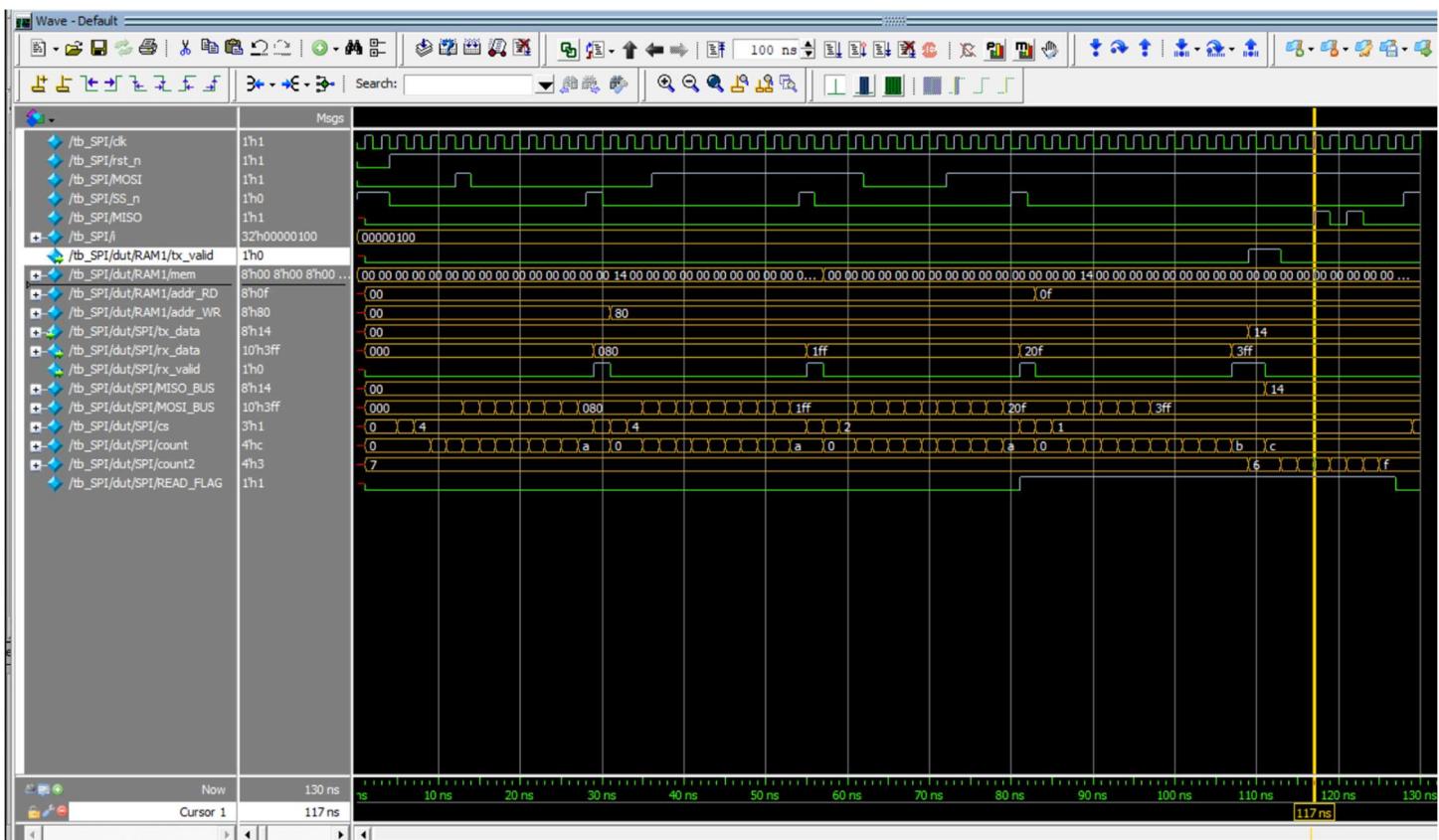
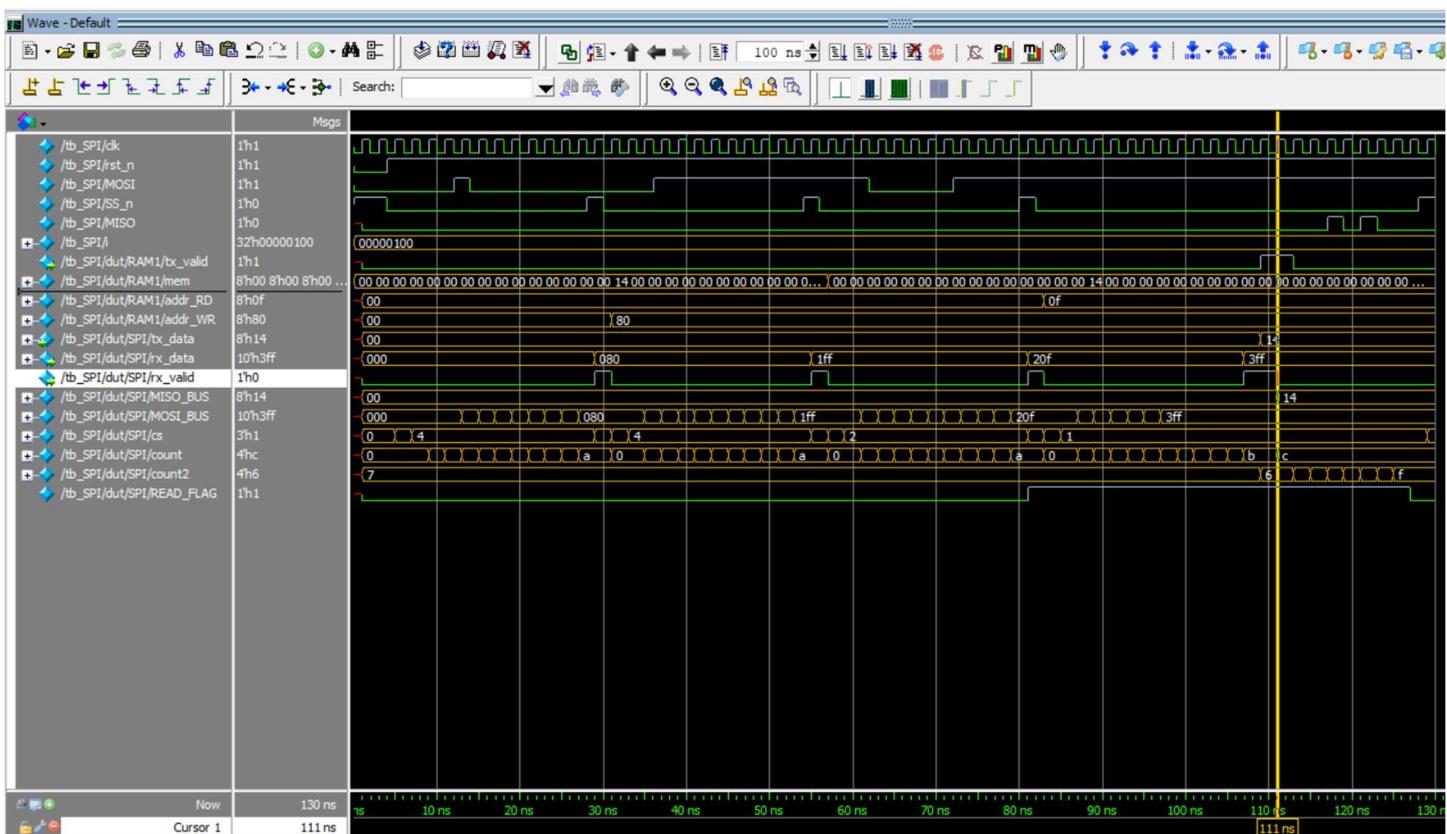
3-read address:

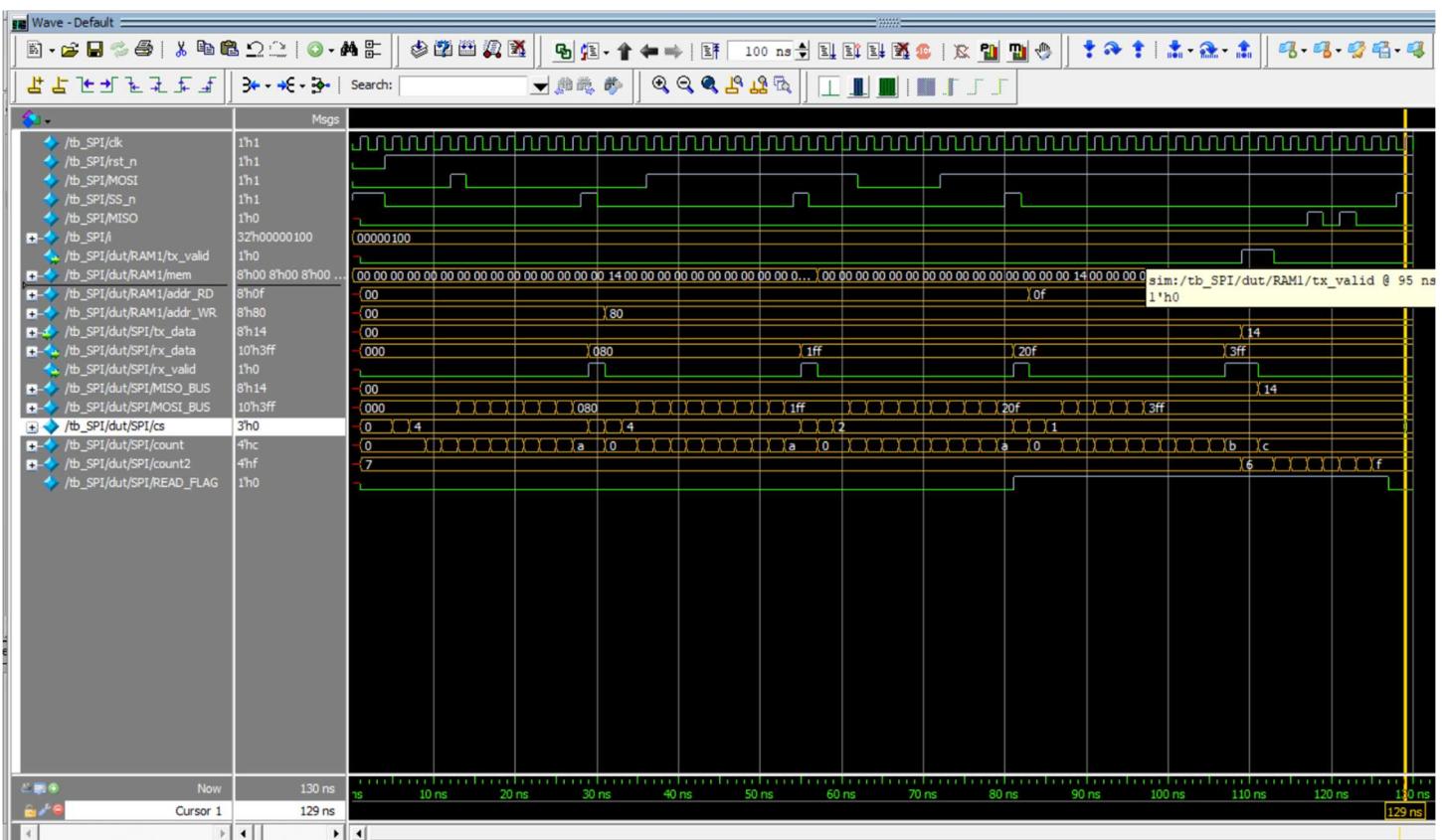
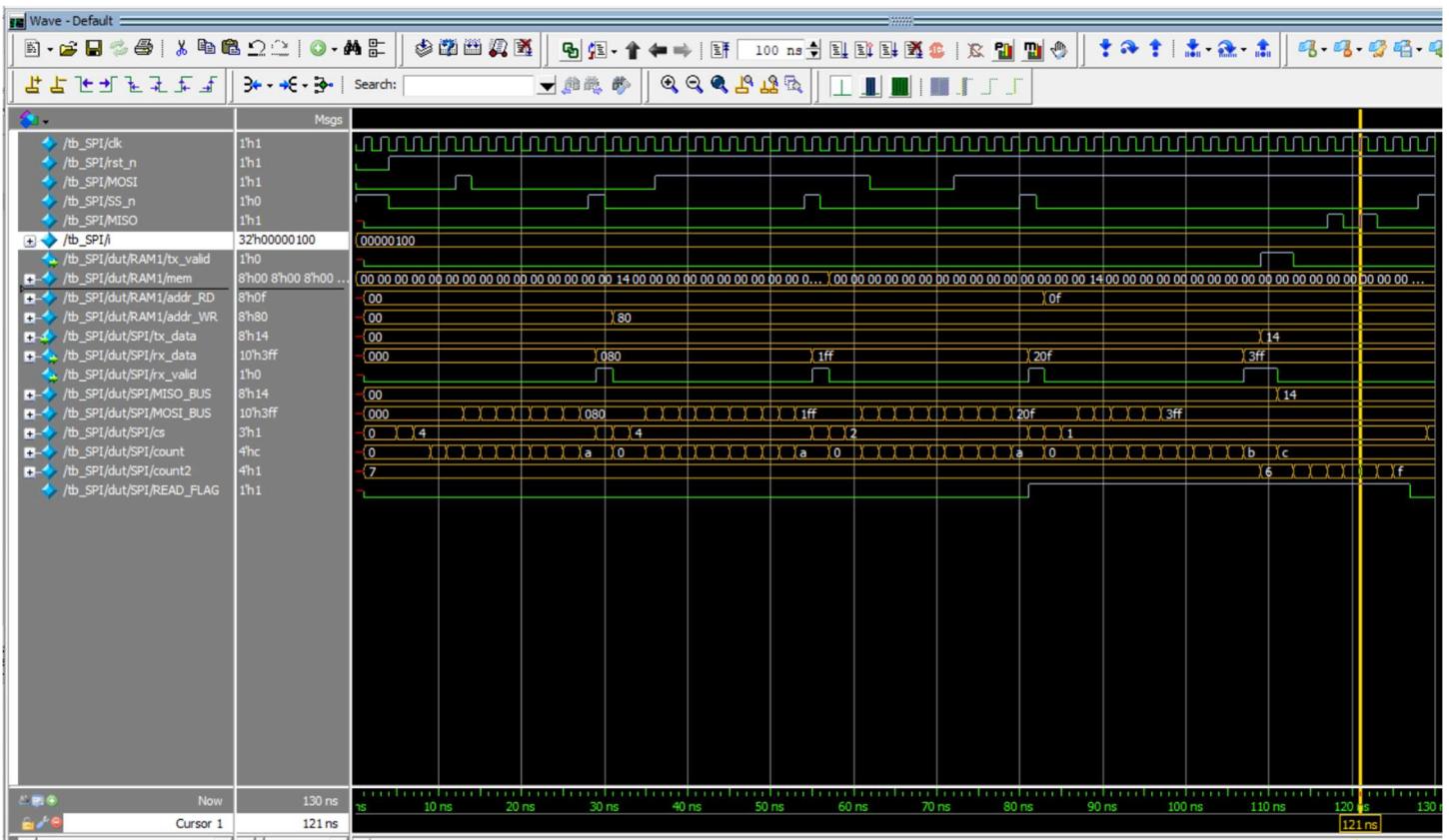




4-read data:







3-linting tool:

V Questa Lint 2021.1 (E:\Temp\lint.db)

File Edit View Schematic Window Help

Design Search: Type Search... Show Collated View

Instance Module

C:/Users/af455/Desktop/digital_diploma/SPI_PROJECT.V [SPI_wrapper]

162 endmodule
163
164
165
166 module SPI_wrapper(input clk,rst_n,MOSI,SS_n,output MISO);
167 wire [9:0]rx_data;
168 wire rx_valid;
169 wire [7:0]tx_data;
170 wire tx_valid;
171 RAM RAM1(.clk(clk),.rst_n(rst_n),.din(rx_data),.rx_valid(rx_valid))
172
173 SPI_SLAVE SPI(.clk(clk),.rst_n(rst_n),.MOSI(MOSI),.SS_n(SS_n),.MIS
174 ,.rx_data(rx_data),.rx_valid(rx_valid));
175 endmodule
176
177 module tb_SPI();
178 reg clk,rst_n,MOSI,SS_n;
179 wire MISO;
180
181 SPI_wrapper dut (.clk(clk),.rst_n(rst_n),.MOSI(MOSI),.SS_n(SS_n),.
182
183 initial begin
184 clk=0;
185 forever
186 #1 clk=~clk;
187 end
188 integer i;
189 initial begin

Flow Navigator Design Lint Checks

Filter: Type here Waived Fixed Pending Uninspected Bug Verified Total : 10 Selected : 0

Schematic 1

RAM1 (RAM) SPI_SLAVE (SPI)

MOSI SS_n CLK DIN DOUT RX_VALID TX_VALID MISO

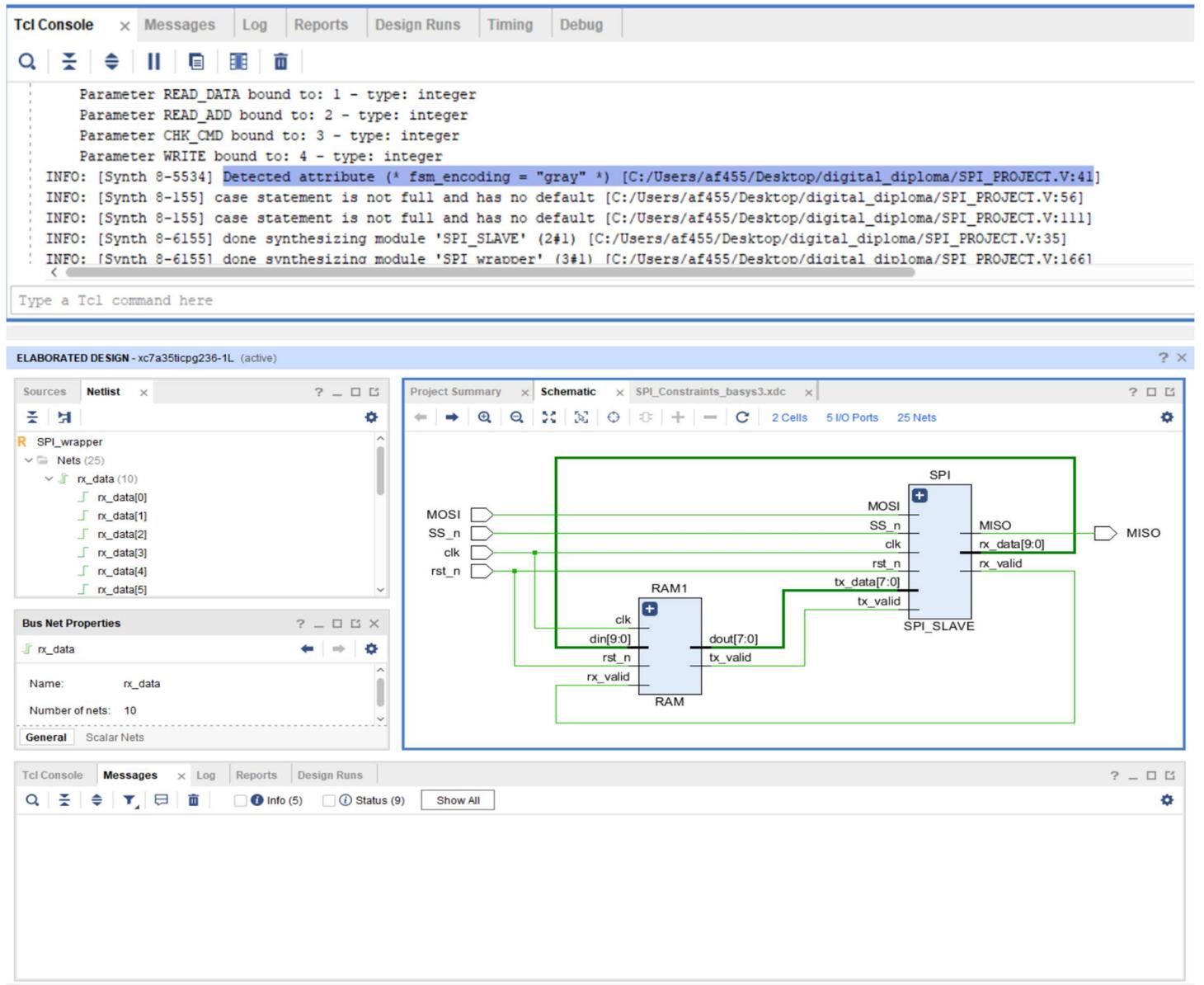
Initial Lint Summary Schematic 1

| Severity | Status | Check | Alias | Message | Module | Category | State | Owner | STARC Reference |
|----------|--------|----------------------------|-------|---|-----------|-----------------------|------------|-------|--|
| Info | ? | flop_output_in_initial | | Flop output is assigned a value in the initial construct. ... | RAM | Rtl Design Style open | unassigned | | 2.3.4.1 |
| Info | ? | seq_block_has_duplicate... | | Signal is assigned more than once in a sequential bloc... | SPI_SLAVE | Rtl Design Style open | unassigned | | 2.2.3.3 |
| Info | ? | seq_block_has_duplicate... | | Signal is assigned more than once in a sequential bloc... | SPI_SLAVE | Rtl Design Style open | unassigned | | 2.2.3.3 |
| Info | ? | seq_block_has_duplicate... | | Signal is assigned more than once in a sequential bloc... | SPI_SLAVE | Rtl Design Style open | unassigned | | 2.2.3.3 |
| Info | ? | seq_block_has_duplicate... | | Signal is assigned more than once in a sequential bloc... | SPI_SLAVE | Rtl Design Style open | unassigned | | 2.2.3.3 |
| Info | ? | assign_width_overflow | | Width of assignment RHS is greater than width of LH... | SPI_SLAVE | Rtl Design Style open | unassigned | | 2.1.3.2, 2.10.3.3, 2.10.3.4, 2.10.4.3... |

Transcript Message Viewer Lint Checks Design Metrics Design Information Status History Lint Dashboard

4-elaboration:

-gray:



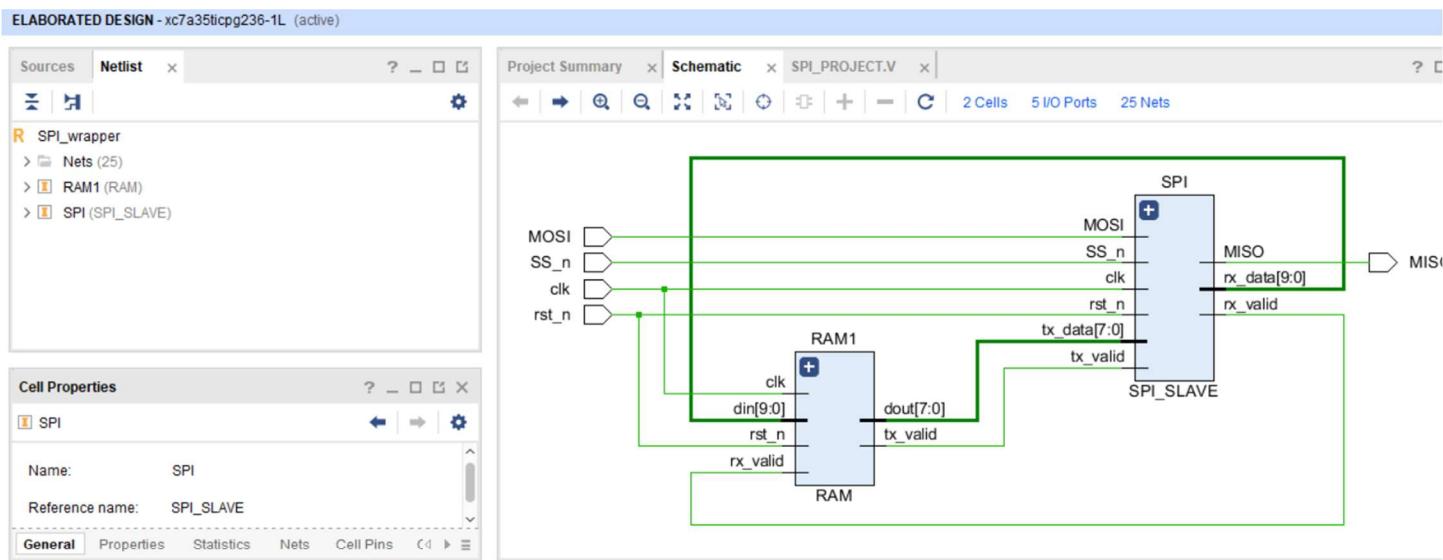
-one hot:

Tcl Console x Messages Log Reports Design Runs Timing Debug

Q | X | D | H | M | S | B |

```
Parameter IDLE bound to: 0 - type: integer
Parameter READ_DATA bound to: 1 - type: integer
Parameter READ_ADD bound to: 2 - type: integer
Parameter CHK_CMD bound to: 3 - type: integer
Parameter WRITE bound to: 4 - type: integer
INFO: [Synth 8-5534] Detected attribute (* fsm_encoding = "one_hot" *) [C:/Users/af455/Desktop/digital_diploma/SPI_PROJECT.V:41]
INFO: [Synth 8-155] case statement is not full and has no default [C:/Users/af455/Desktop/digital_diploma/SPI_PROJECT.V:56]
INFO: [Synth 8-155] case statement is not full and has no default [C:/Users/af455/Desktop/digital_diploma/SPI_PROJECT.V:111]
INFO: [Synth 8-61551] done synthesizing module 'SPI_SLAVE' (2#1) [C:/Users/af455/Desktop/digital_diploma/SPI_PROJECT.V:351]
```

Type a Tcl command here



Tcl Console Messages x Log Reports Design Runs

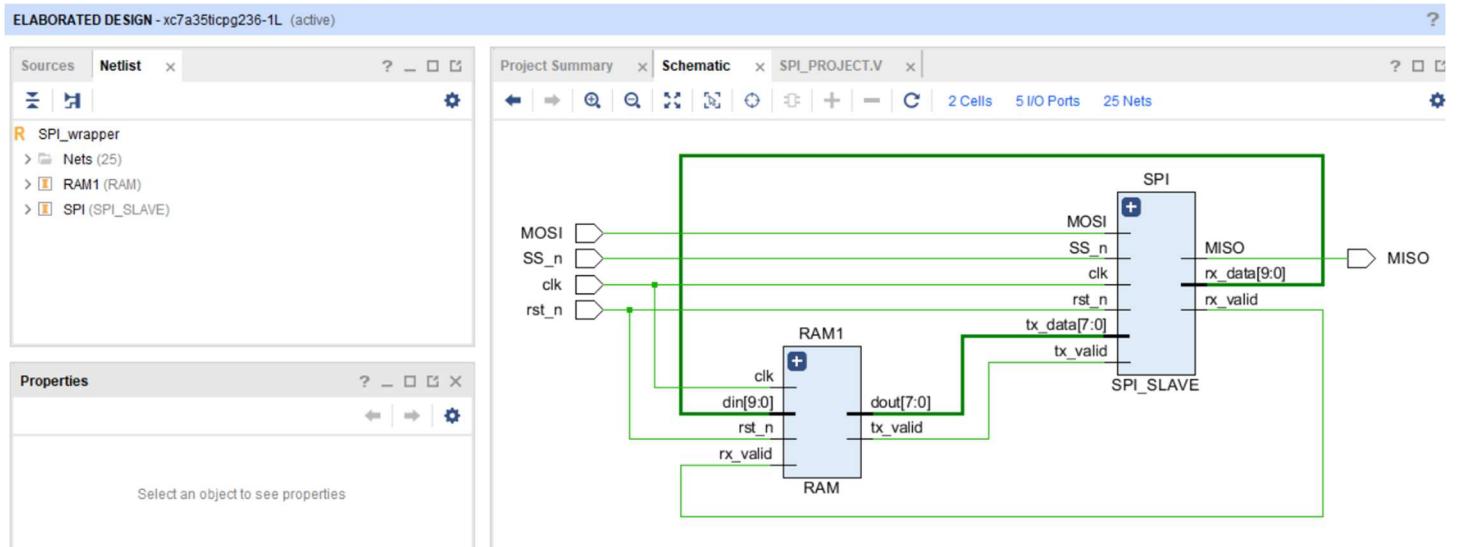
Q | X | D | H | M | S | B |

W Warning (2) I Info (48) S Status (54) Show All

Synthesis (2 warnings)

- [Synth 8-327] inferring latch for variable 'FSM_onehot_ns_reg' [SPI_PROJECT.V:59]
- [Constraints 18-5210] No constraint will be written out.

-seq:



Tcl Console Messages Log Reports Design Runs

Parameter READ_DATA bound to: 1 - type: integer

Parameter READ_ADD bound to: 2 - type: integer

Parameter CHK_CMD bound to: 3 - type: integer

Parameter WRITE bound to: 4 - type: integer

INFO: [Synth 8-5534] Detected attribute (* fsm_encoding = "sequential" *) [C:/Users/af455/Desktop/digital_diploma/SPI_PROJECT.V:41]

INFO: [Synth 8-155] case statement is not full and has no default [C:/Users/af455/Desktop/digital_diploma/SPI_PROJECT.V:56]

INFO: [Synth 8-155] case statement is not full and has no default [C:/Users/af455/Desktop/digital_diploma/SPI_PROJECT.V:111]

INFO: [Synth 8-6155] done synthesizing module 'SPI_SLAVE' (2#1) [C:/Users/af455/Desktop/digital_diploma/SPI_PROJECT.V:35]

INFO: [Synth 8-61551] done synthesizing module 'SPI wrapper' (3#1) [C:/Users/af455/Desktop/digital_diploma/SPI_PROJECT.V:166]

Type a Tcl command here

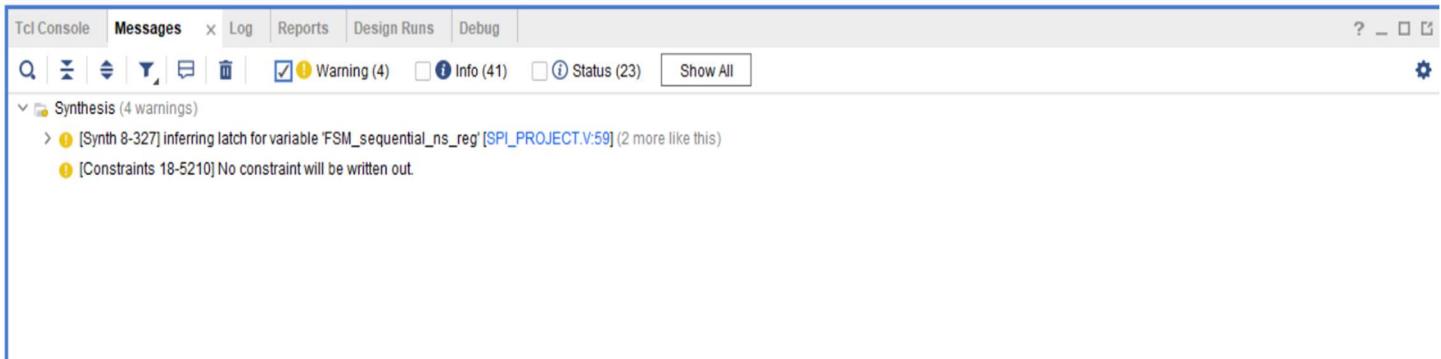
Tcl Console Messages Log Reports Design Runs

Info (215) Status (517) Show All

5-synthesis:

1-messages:

-gray:



-one hot:

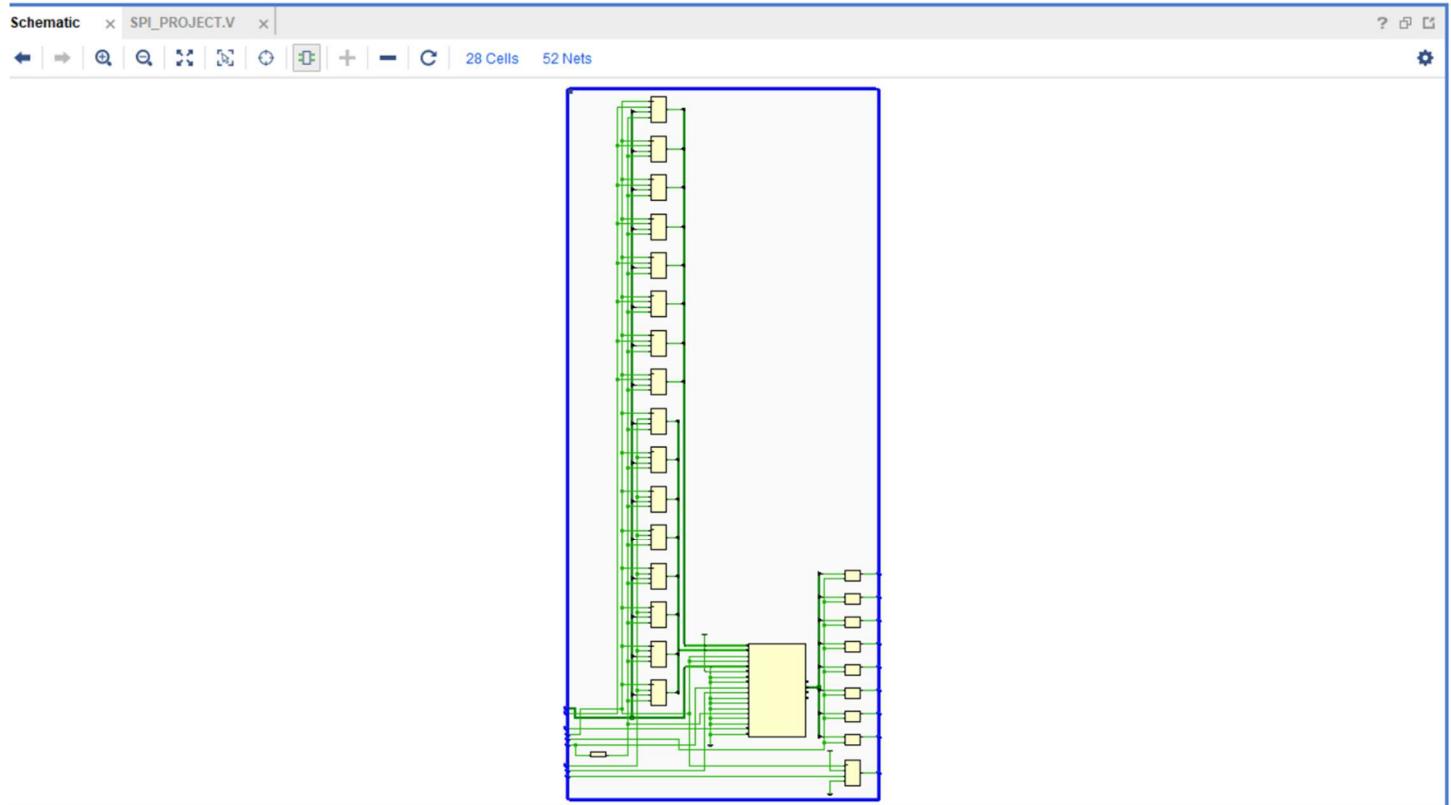
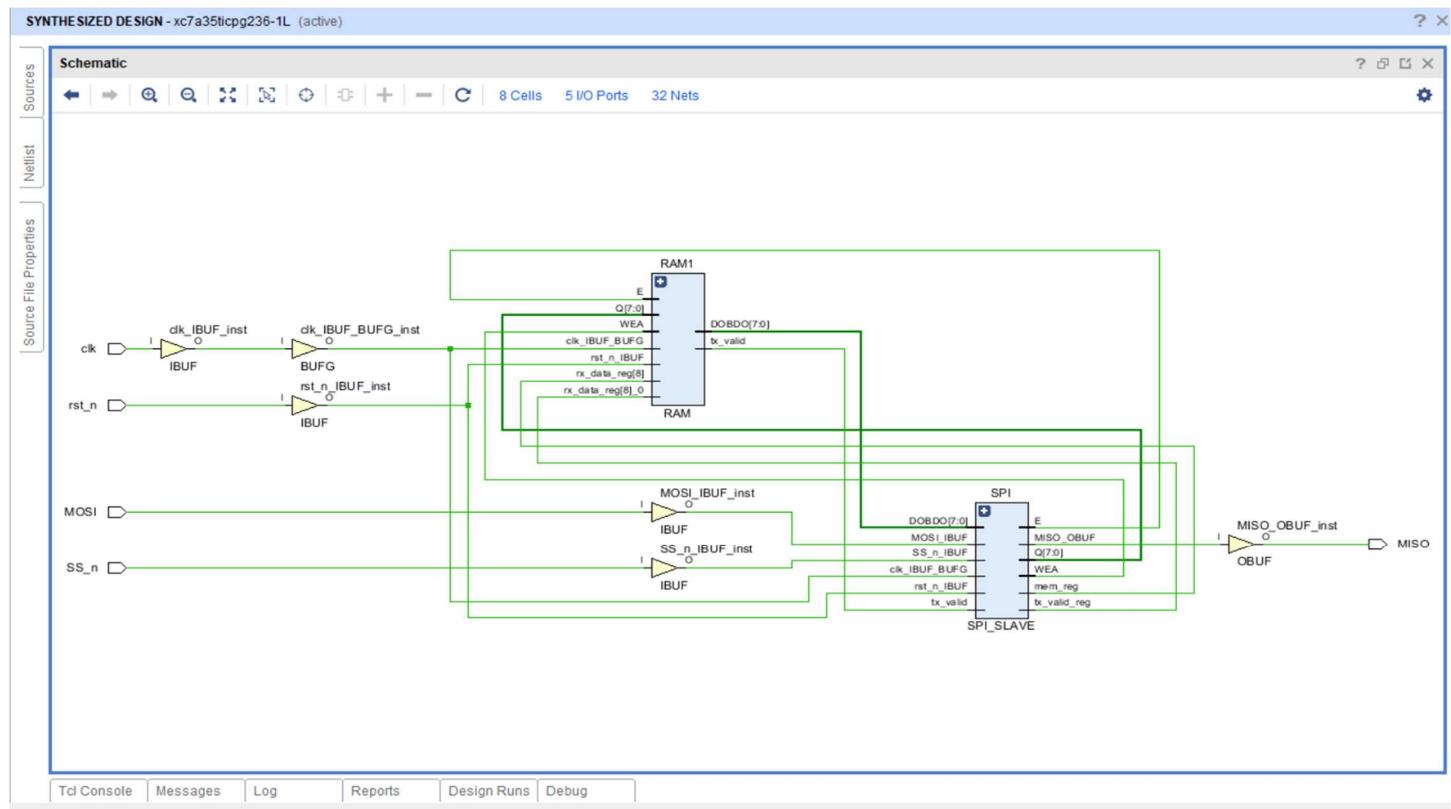


-seq:



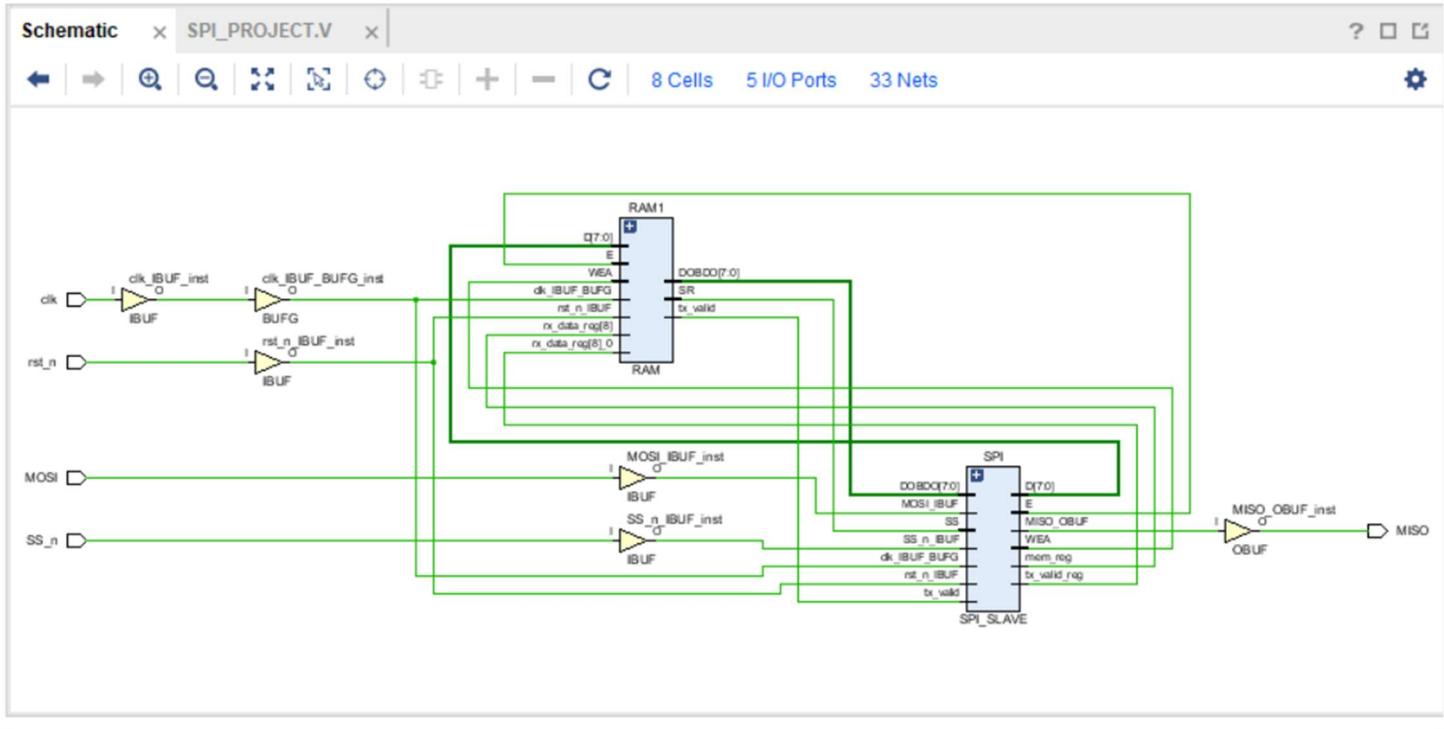
2-schematics:

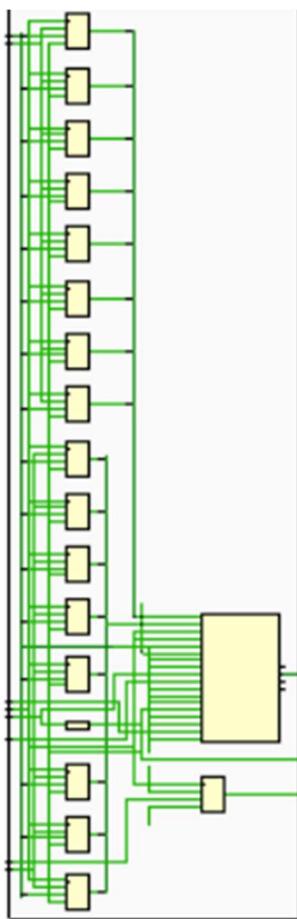
-gray:





-One hot:

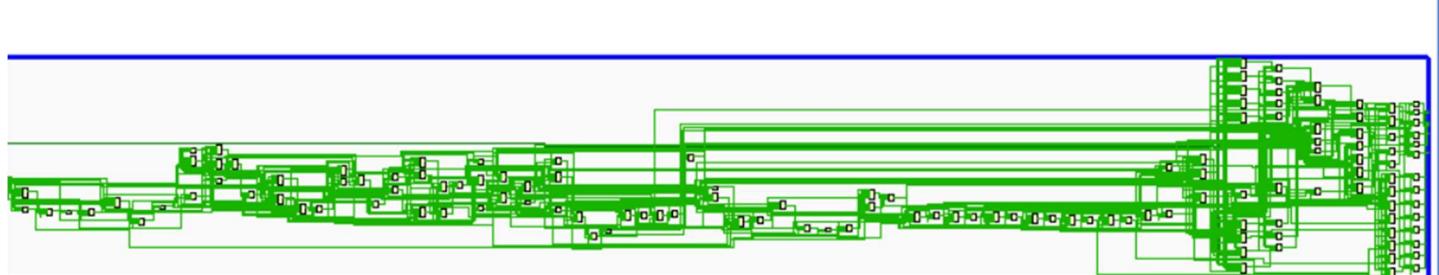
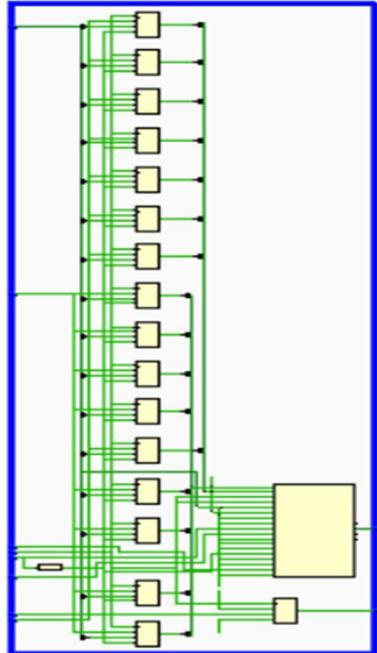
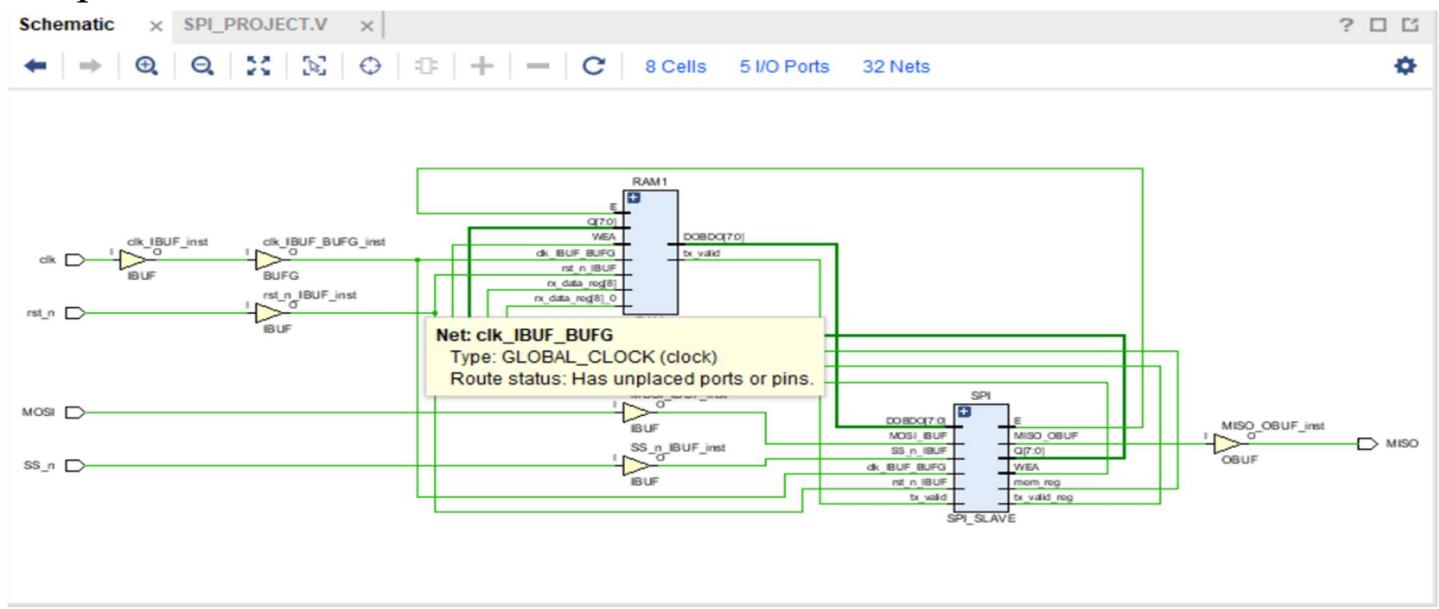




Schematic x SPI_PROJECT.V x synth_1_synth_synthesis_report_0 - synth_1 x ? □ ☰
◀ ▶ Q Q X X + - C 136 Cells 153 Nets



-seq:



3-utilization report:

-gray:

The screenshot shows the Utilization tab in the Vivado interface. The left sidebar has 'utilization_1' selected. The main area displays utilization data for various components. The table includes columns for Name, Slice LUTs (20800), Slice Registers (41600), Block RAM Tile (50), Bonded IOB (106), and BUFGCTRL (32). The SPI_wrapper component is expanded, showing its internal components: RAM1 (RAM) and SPI (SPI_SLAVE).

| Name | Slice LUTs (20800) | Slice Registers (41600) | Block RAM Tile (50) | Bonded IOB (106) | BUFGCTRL (32) |
|-----------------|--------------------|-------------------------|---------------------|------------------|---------------|
| N SPI_wrapper | 72 | 62 | 0.5 | 5 | 1 |
| RAM1 (RAM) | 5 | 17 | 0.5 | 0 | 0 |
| SPI (SPI_SLAVE) | 67 | 45 | 0 | 0 | 0 |

-one hot:

The screenshot shows the Utilization tab in the Vivado interface. The left sidebar has 'utilization_1' selected. The main area displays utilization data for various components. The table includes columns for Name, Slice LUTs (20800), Slice Registers (41600), F7 Muxes (16300), Block RAM Tile (50), Bonded IOB (106), and BUFGCTRL (32). The SPI_wrapper component is expanded, showing its internal components: RAM1 (RAM) and SPI (SPI_SLAVE). The SPI component is highlighted with a blue selection bar.

| Name | Slice LUTs (20800) | Slice Registers (41600) | F7 Muxes (16300) | Block RAM Tile (50) | Bonded IOB (106) | BUFGCTRL (32) |
|-----------------|--------------------|-------------------------|------------------|---------------------|------------------|---------------|
| N SPI_wrapper | 71 | 66 | 1 | 0.5 | 5 | 1 |
| RAM1 (RAM) | 1 | 17 | 0 | 0.5 | 0 | 0 |
| SPI (SPI_SLAVE) | 70 | 49 | 1 | 0 | 0 | 0 |

-seq:

The screenshot shows the Utilization tab in the Vivado interface. The left sidebar has 'utilization_1' selected. The main area displays utilization data for various components. The table includes columns for Name, Slice LUTs (20800), Slice Registers (41600), Block RAM Tile (50), Bonded IOB (106), and BUFGCTRL (32). The SPI_wrapper component is expanded, showing its internal components: RAM1 (RAM) and SPI (SPI_SLAVE).

| Name | Slice LUTs (20800) | Slice Registers (41600) | Block RAM Tile (50) | Bonded IOB (106) | BUFGCTRL (32) |
|-----------------|--------------------|-------------------------|---------------------|------------------|---------------|
| N SPI_wrapper | 87 | 62 | 0.5 | 5 | 1 |
| RAM1 (RAM) | 1 | 17 | 0.5 | 0 | 0 |
| SPI (SPI_SLAVE) | 86 | 45 | 0 | 0 | 0 |

4-time report:

-Gray encoding:

Tcl Console | Messages | Log | Reports | Design Runs | **Timing** | Debug | ? | □

Design Timing Summary

General Information

Timer Settings

Design Timing Summary

Clock Summary (1)

> Check Timing (16)

> Intra-Clock Paths

Inter-Clock Paths

Other Path Groups

Timing Summary - timing_1

| Setup | Hold | Pulse Width |
|---|---|---|
| Worst Negative Slack (WNS): 6.099 ns | Worst Hold Slack (WHS): 0.149 ns | Worst Pulse Width Slack (WPWS): 4.500 ns |
| Total Negative Slack (TNS): 0.000 ns | Total Hold Slack (THS): 0.000 ns | Total Pulse Width Negative Slack (TPWS): 0.000 ns |
| Number of Failing Endpoints: 0 | Number of Failing Endpoints: 0 | Number of Failing Endpoints: 0 |
| Total Number of Endpoints: 146 | Total Number of Endpoints: 146 | Total Number of Endpoints: 62 |

All user specified timing constraints are met.

-One hot:

Tcl Console | Messages | Log | Reports | Design Runs | **Timing** | Utilization | Debug | ? | □

Design Timing Summary

General Information

Timer Settings

Design Timing Summary

Clock Summary (1)

> Check Timing (34)

> Intra-Clock Paths

Inter-Clock Paths

Other Path Groups

Timing Summary - timing_1

| Setup | Hold | Pulse Width |
|---|---|---|
| Worst Negative Slack (WNS): 5.739 ns | Worst Hold Slack (WHS): 0.161 ns | Worst Pulse Width Slack (WPWS): 4.500 ns |
| Total Negative Slack (TNS): 0.000 ns | Total Hold Slack (THS): 0.000 ns | Total Pulse Width Negative Slack (TPWS): 0.000 ns |
| Number of Failing Endpoints: 0 | Number of Failing Endpoints: 0 | Number of Failing Endpoints: 0 |
| Total Number of Endpoints: 164 | Total Number of Endpoints: 164 | Total Number of Endpoints: 64 |

All user specified timing constraints are met.

-seq:

Tcl Console | Messages | Log | Reports | Design Runs | **Timing** | Debug | ? | □

Design Timing Summary

General Information

Timer Settings

Design Timing Summary

Clock Summary (1)

> Check Timing (16)

> Intra-Clock Paths

Inter-Clock Paths

Other Path Groups

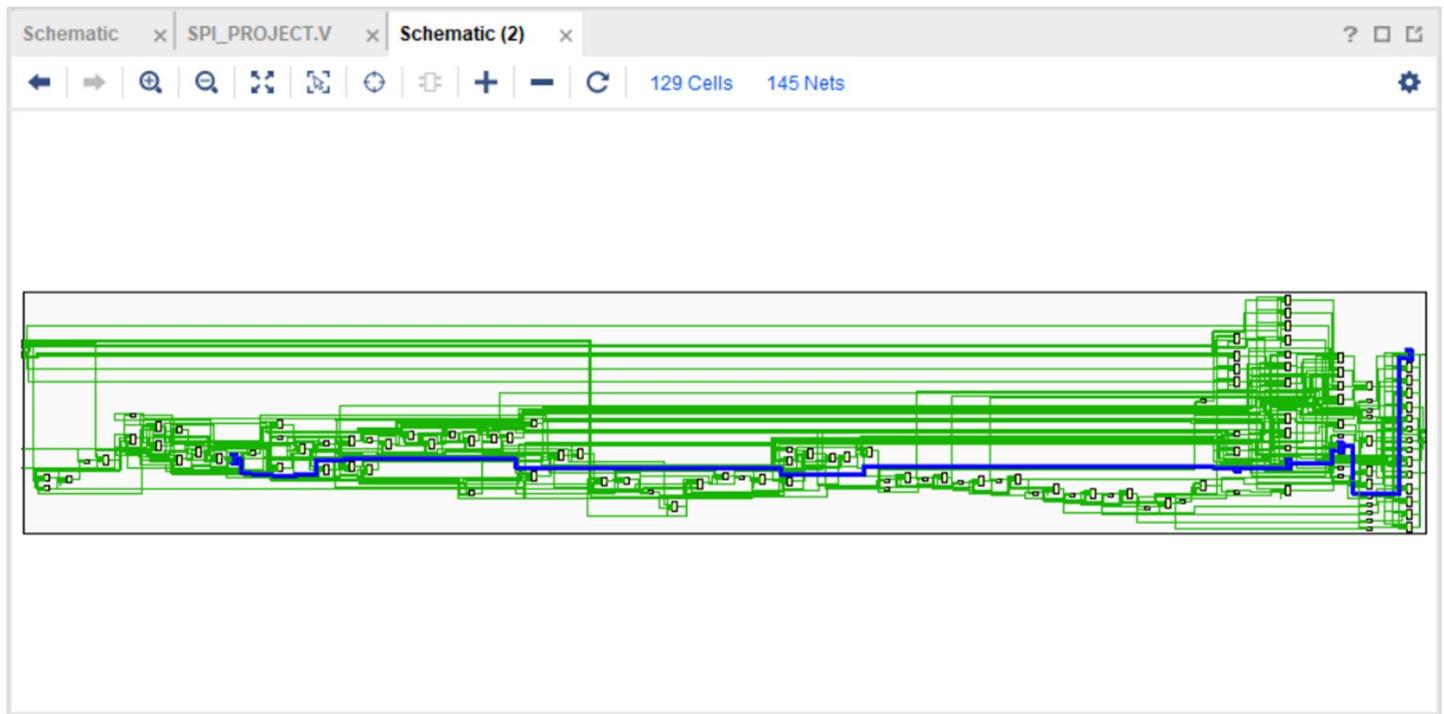
Timing Summary - timing_1

| Setup | Hold | Pulse Width |
|---|---|---|
| Worst Negative Slack (WNS): 5.798 ns | Worst Hold Slack (WHS): 0.161 ns | Worst Pulse Width Slack (WPWS): 4.500 ns |
| Total Negative Slack (TNS): 0.000 ns | Total Hold Slack (THS): 0.000 ns | Total Pulse Width Negative Slack (TPWS): 0.000 ns |
| Number of Failing Endpoints: 0 | Number of Failing Endpoints: 0 | Number of Failing Endpoints: 0 |
| Total Number of Endpoints: 140 | Total Number of Endpoints: 140 | Total Number of Endpoints: 62 |

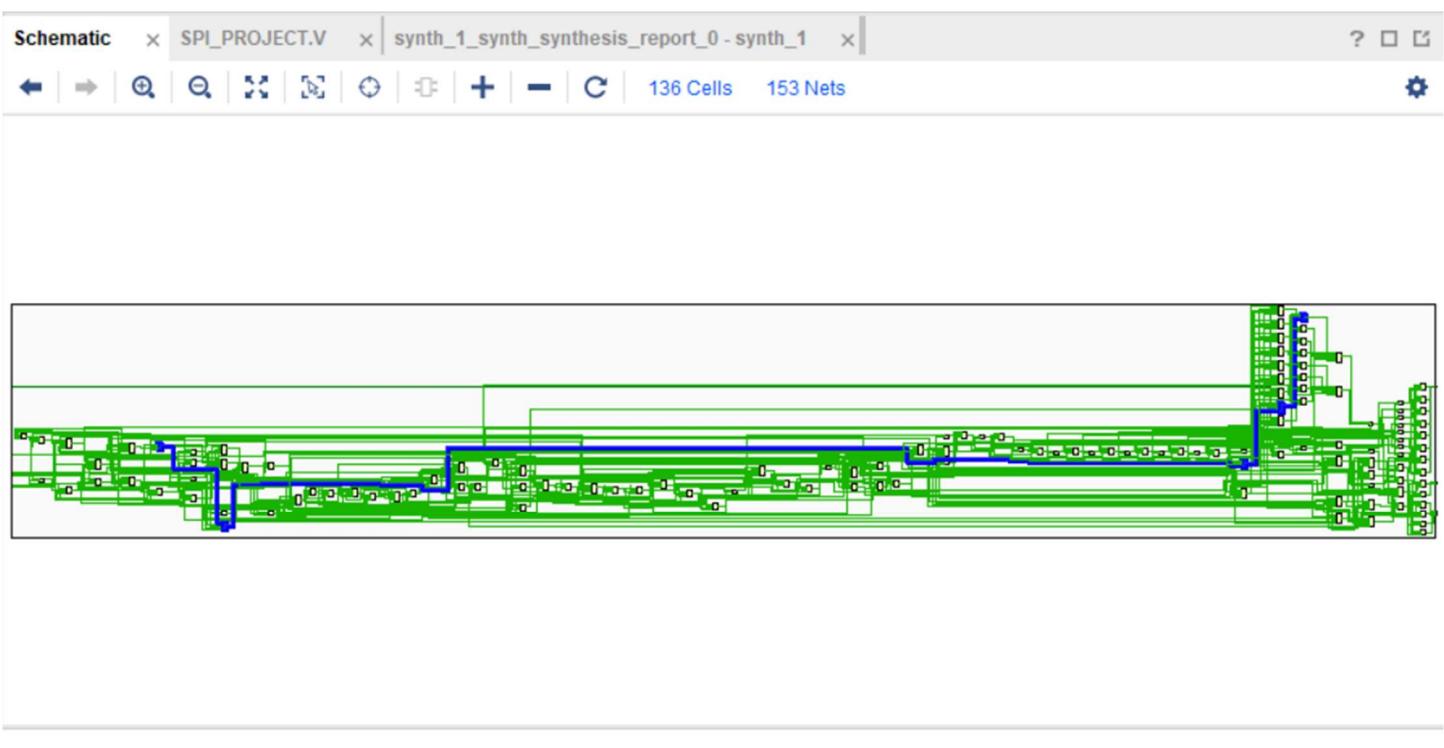
All user specified timing constraints are met.

5-critical path:

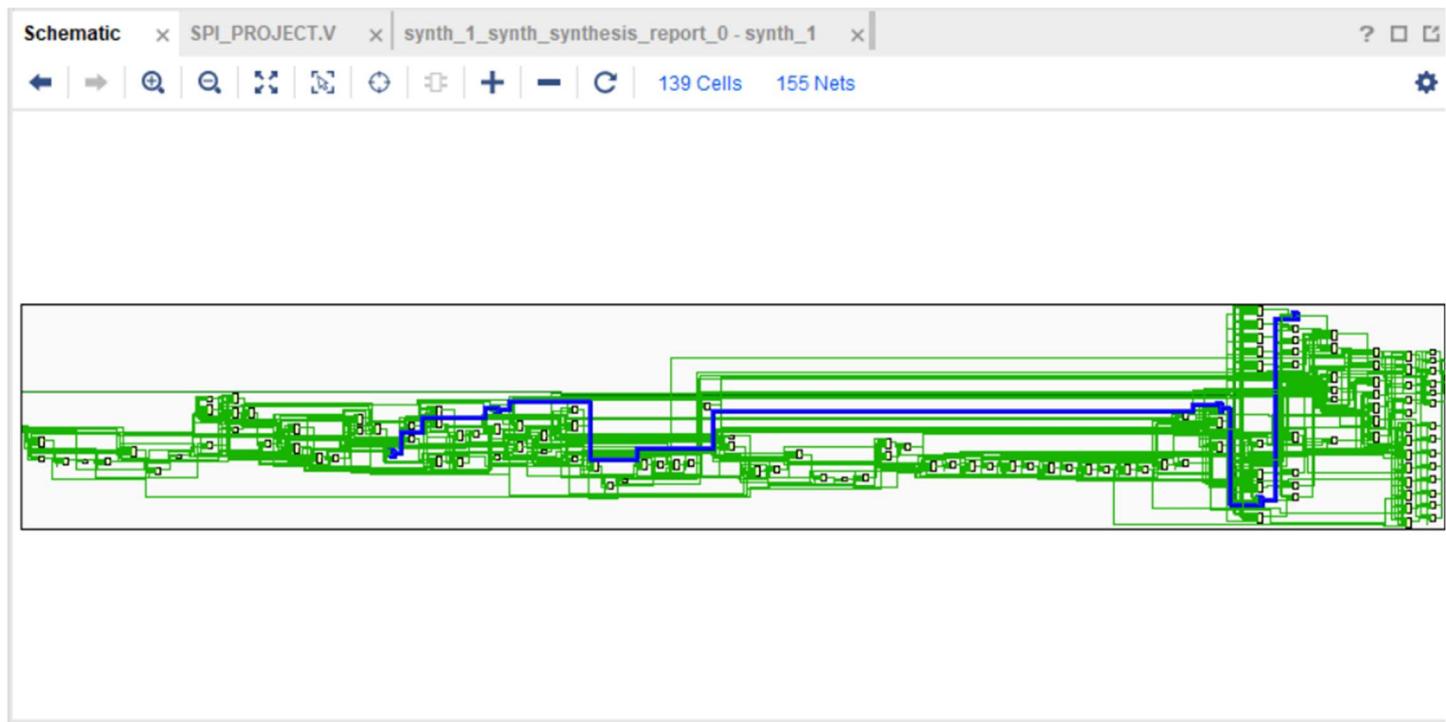
-gray:



-one hot:



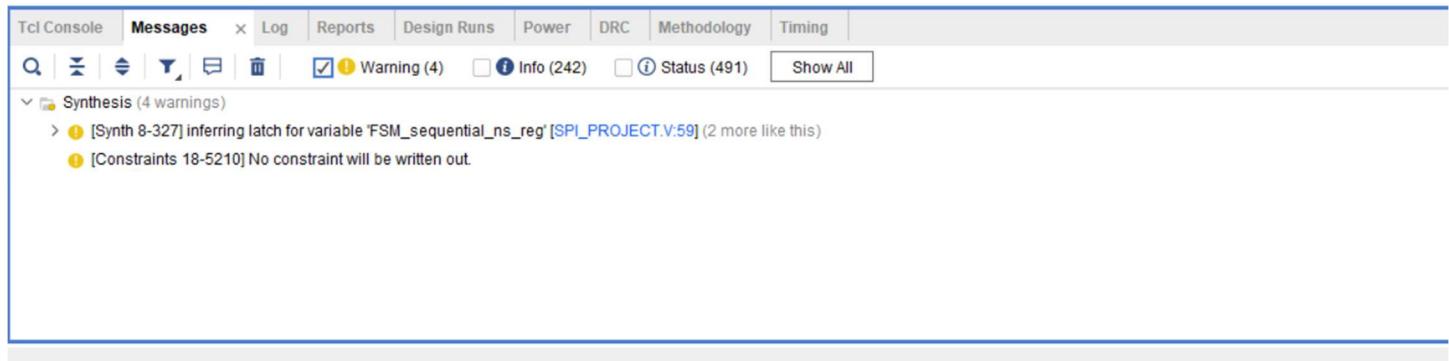
-seq:



6-implementation report:

1-messages:

-gray:



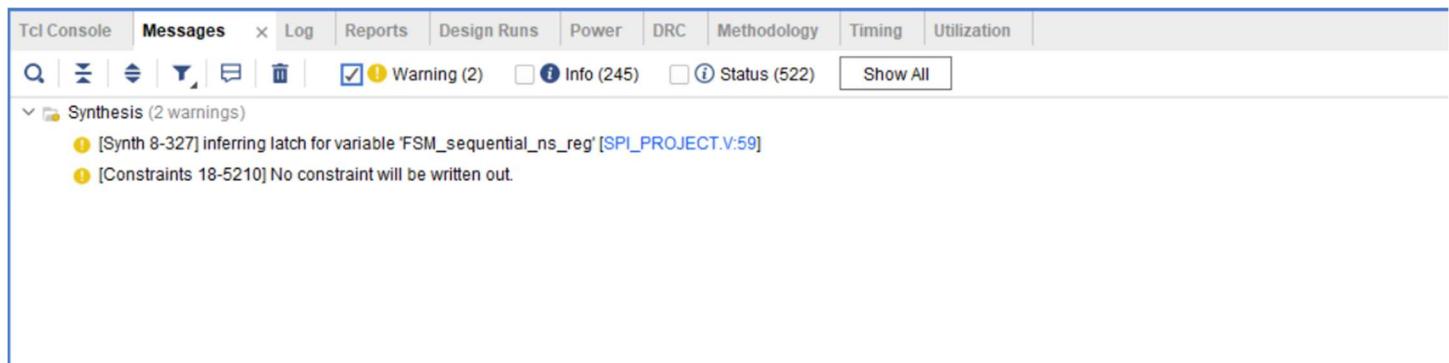
Tcl Console | **Messages** | Log | Reports | Design Runs | Power | DRC | Methodology | Timing | Utilization

Search | Filter | Show All | Warning (4) | Info (242) | Status (491)

Synthesis (4 warnings)

- > ! [Synth 8-327] inferring latch for variable 'FSM_sequential_ns_reg' [SPI_PROJECT.V:59] (2 more like this)
- ! [Constraints 18-5210] No constraint will be written out.

-one hot:



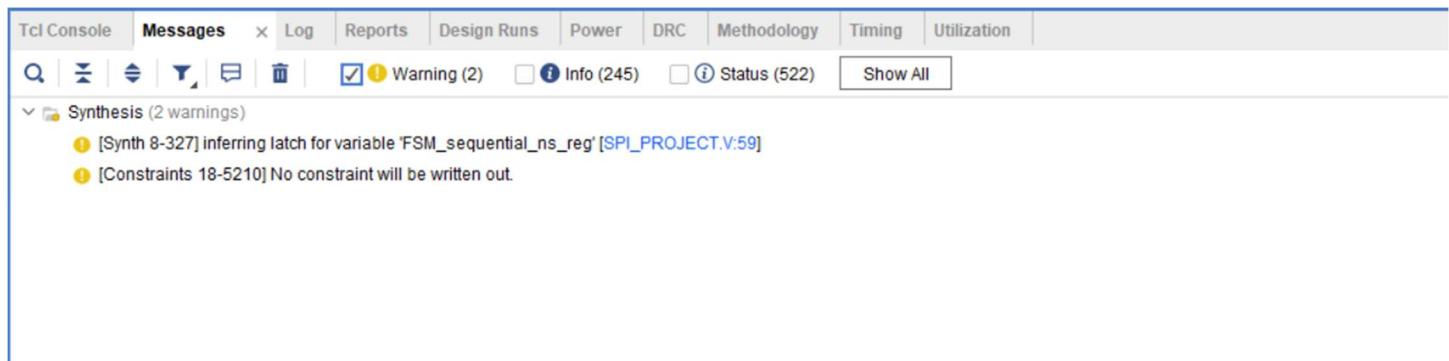
Tcl Console | **Messages** | Log | Reports | Design Runs | Power | DRC | Methodology | Timing | Utilization

Search | Filter | Show All | Warning (2) | Info (245) | Status (522)

Synthesis (2 warnings)

- ! [Synth 8-327] inferring latch for variable 'FSM_sequential_ns_reg' [SPI_PROJECT.V:59]
- ! [Constraints 18-5210] No constraint will be written out.

-seq:



Tcl Console | **Messages** | Log | Reports | Design Runs | Power | DRC | Methodology | Timing | Utilization

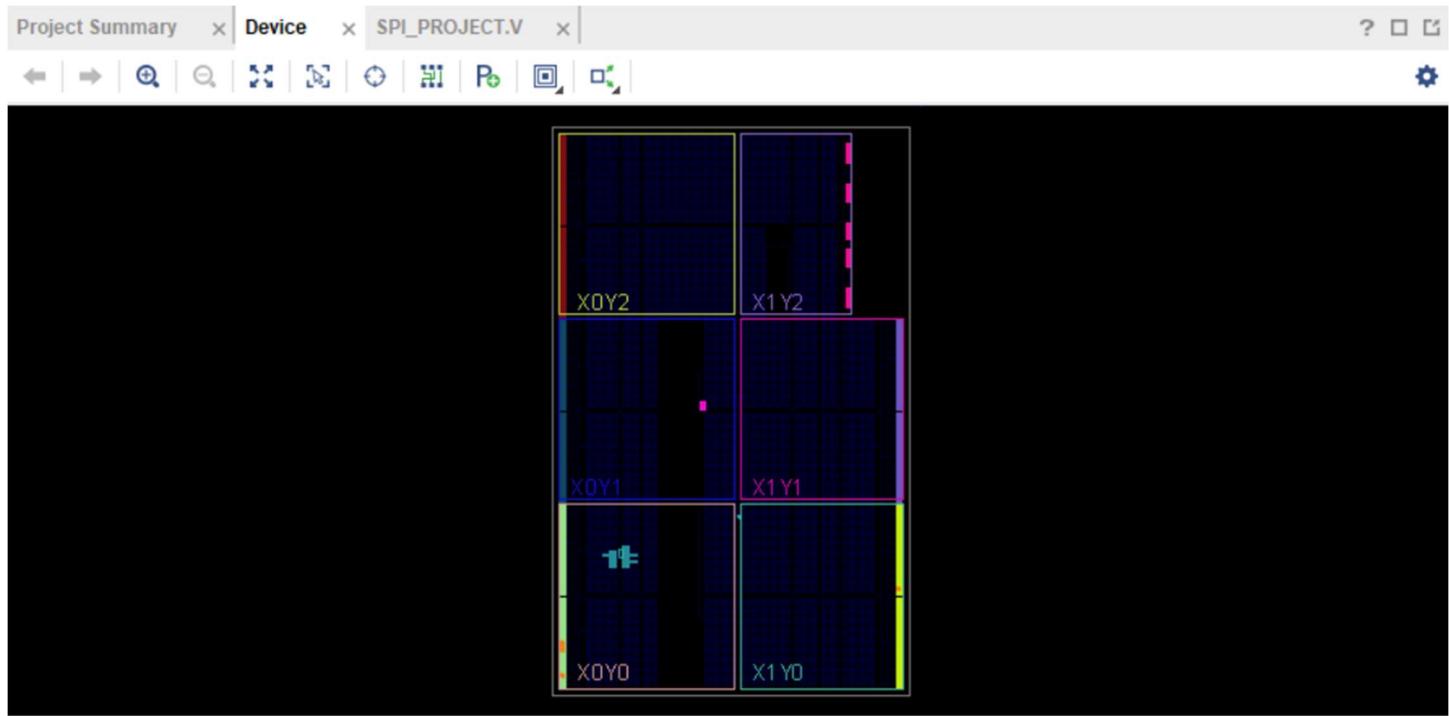
Search | Filter | Show All | Warning (2) | Info (245) | Status (522)

Synthesis (2 warnings)

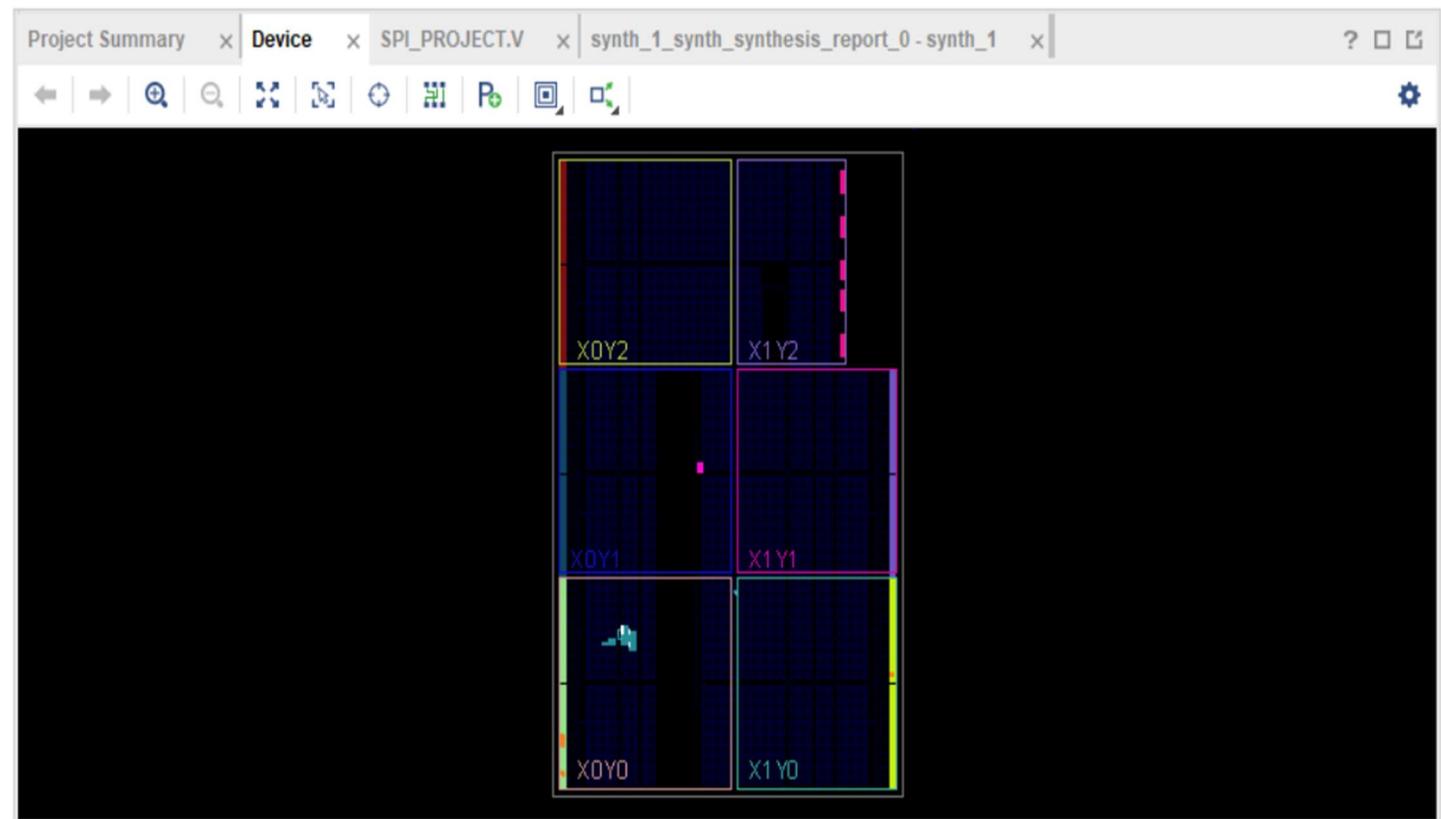
- ! [Synth 8-327] inferring latch for variable 'FSM_sequential_ns_reg' [SPI_PROJECT.V:59]
- ! [Constraints 18-5210] No constraint will be written out.

2-device:

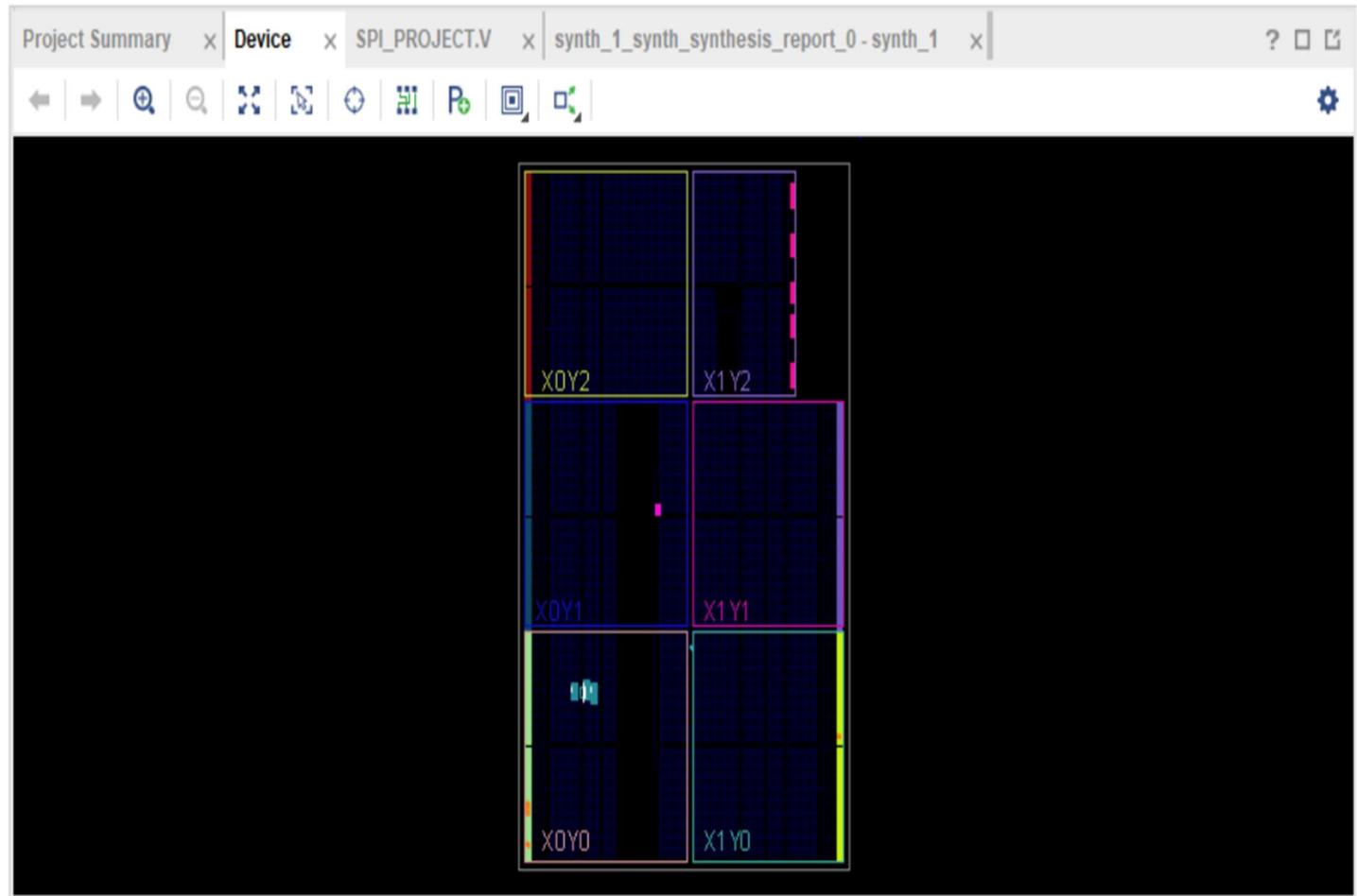
-gray:



-one hot:

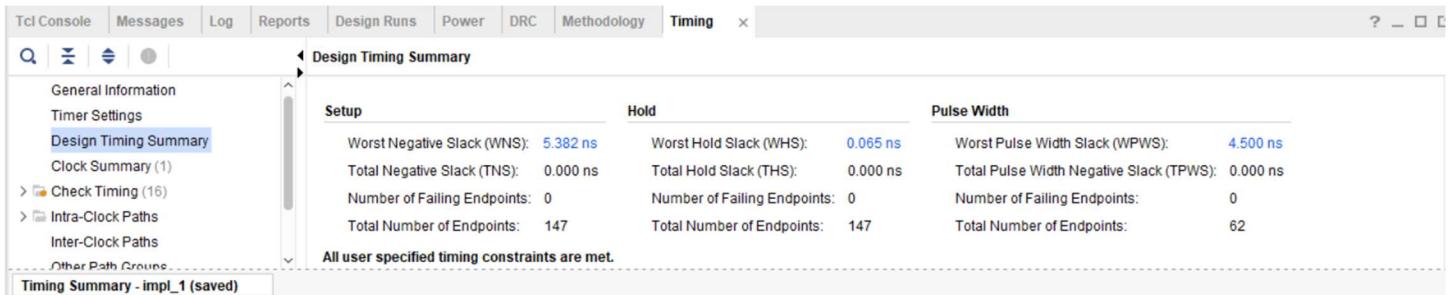


-seq:



3-Time report:

-gray: (fastest design)



Tcl Console Messages Log Reports Design Runs Power DRC Methodology **Timing** × ? _ □

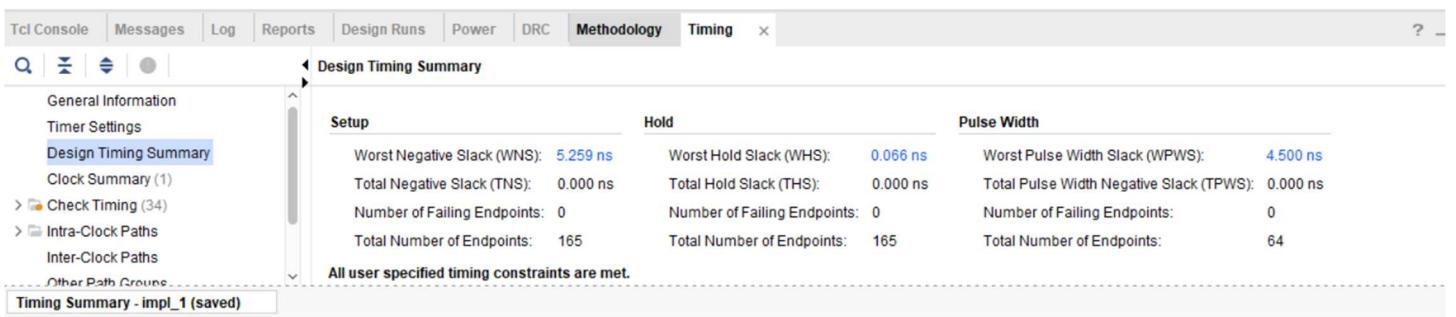
Design Timing Summary

General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (16)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
Timing Summary - impl_1 (saved)

| Setup | | | Hold | | | Pulse Width | | |
|------------------------------|-----------------|--|------------------------------|-----------------|--|--|-----------------|--|
| Worst Negative Slack (WNS): | 5.382 ns | | Worst Hold Slack (WHS): | 0.065 ns | | Worst Pulse Width Slack (WPWS): | 4.500 ns | |
| Total Negative Slack (TNS): | 0.000 ns | | Total Hold Slack (THS): | 0.000 ns | | Total Pulse Width Negative Slack (TPWS): | 0.000 ns | |
| Number of Failing Endpoints: | 0 | | Number of Failing Endpoints: | 0 | | Number of Failing Endpoints: | 0 | |
| Total Number of Endpoints: | 147 | | Total Number of Endpoints: | 147 | | Total Number of Endpoints: | 62 | |

All user specified timing constraints are met.

-One hot:



Tcl Console Messages Log Reports Design Runs Power DRC Methodology **Timing** × ? _ □

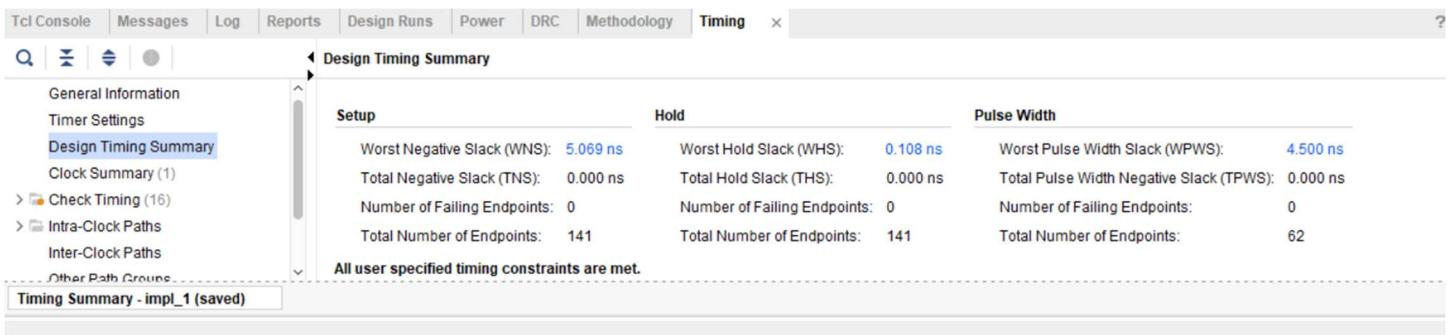
Design Timing Summary

General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (34)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
Timing Summary - impl_1 (saved)

| Setup | | | Hold | | | Pulse Width | | |
|------------------------------|-----------------|--|------------------------------|-----------------|--|--|-----------------|--|
| Worst Negative Slack (WNS): | 5.259 ns | | Worst Hold Slack (WHS): | 0.066 ns | | Worst Pulse Width Slack (WPWS): | 4.500 ns | |
| Total Negative Slack (TNS): | 0.000 ns | | Total Hold Slack (THS): | 0.000 ns | | Total Pulse Width Negative Slack (TPWS): | 0.000 ns | |
| Number of Failing Endpoints: | 0 | | Number of Failing Endpoints: | 0 | | Number of Failing Endpoints: | 0 | |
| Total Number of Endpoints: | 165 | | Total Number of Endpoints: | 165 | | Total Number of Endpoints: | 64 | |

All user specified timing constraints are met.

-seq:



Tcl Console Messages Log Reports Design Runs Power DRC Methodology **Timing** × ? _ □

Design Timing Summary

General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (16)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups
Timing Summary - impl_1 (saved)

| Setup | | | Hold | | | Pulse Width | | |
|------------------------------|-----------------|--|------------------------------|-----------------|--|--|-----------------|--|
| Worst Negative Slack (WNS): | 5.069 ns | | Worst Hold Slack (WHS): | 0.108 ns | | Worst Pulse Width Slack (WPWS): | 4.500 ns | |
| Total Negative Slack (TNS): | 0.000 ns | | Total Hold Slack (THS): | 0.000 ns | | Total Pulse Width Negative Slack (TPWS): | 0.000 ns | |
| Number of Failing Endpoints: | 0 | | Number of Failing Endpoints: | 0 | | Number of Failing Endpoints: | 0 | |
| Total Number of Endpoints: | 141 | | Total Number of Endpoints: | 141 | | Total Number of Endpoints: | 62 | |

All user specified timing constraints are met.

4-utilization report:

-gray:

| Hierarchy | Name | 1 | Slice LUTs (20800) | Slice Registers (41600) | Slice (815 0) | LUT as Logic (20800) | LUT Flip Flop Pairs (20800) | Block RAM Tile (50) | Bonded IOB (106) | BUFGCTRL (32) |
|-------------------------|-----------------|----|-----------------------|----------------------------|---------------------|-------------------------|--------------------------------|------------------------|---------------------|------------------|
| Summary | | | | | | | | | | |
| Slice Logic | | | | | | | | | | |
| Slice LUTs (1%) | | | | | | | | | | |
| LUT as Logic (1%) | | | | | | | | | | |
| Slice Registers (<1%) | | | | | | | | | | |
| Register as Latch (<1%) | | | | | | | | | | |
| | N SPI_wrapper | 73 | | 62 | 25 | | 73 | | 29 | |
| | RAM1 (RAM) | 6 | | 17 | 6 | | 6 | | 0 | |
| | SPI (SPI_SLAVE) | 67 | | 45 | 24 | | 67 | | 25 | |

-one hot:(best area)

| Hierarchy | Name | 1 | Slice LUTs (20800) | Slice Registers (41600) | F7 Muxes (16300) | Slice (815 0) | LUT as Logic (20800) | LUT Flip Flop Pairs (20800) | Block RAM Tile (50) | Bonded IOB (106) | BUFGCTRL (32) |
|-----------------------|-----------------|----|-----------------------|----------------------------|------------------------|---------------------|-------------------------|--------------------------------|------------------------|---------------------|------------------|
| Summary | | | | | | | | | | | |
| Slice Logic | | | | | | | | | | | |
| Slice LUTs (1%) | | | | | | | | | | | |
| LUT as Logic (1%) | | | | | | | | | | | |
| F7 Muxes (<1%) | | | | | | | | | | | |
| Slice Registers (<1%) | | | | | | | | | | | |
| | N SPI_wrapper | 70 | | 66 | 1 | 24 | | 70 | | 32 | |
| | RAM1 (RAM) | 2 | | 17 | 0 | 4 | | 2 | | 0 | |
| | SPI (SPI_SLAVE) | 69 | | 49 | 1 | 22 | | 69 | | 31 | |

-seq:

| Hierarchy | Name | 1 | Slice LUTs (20800) | Slice Registers (41600) | Slice (815 0) | LUT as Logic (20800) | LUT Flip Flop Pairs (20800) | Block RAM Tile (50) | Bonded IOB (106) | BUFGCTRL (32) |
|-------------------------|-----------------|----|-----------------------|----------------------------|---------------------|-------------------------|--------------------------------|------------------------|---------------------|------------------|
| Summary | | | | | | | | | | |
| Slice Logic | | | | | | | | | | |
| Slice LUTs (1%) | | | | | | | | | | |
| LUT as Logic (1%) | | | | | | | | | | |
| Slice Registers (<1%) | | | | | | | | | | |
| Register as Latch (<1%) | | | | | | | | | | |
| | N SPI_wrapper | 87 | | 62 | 25 | | 87 | | 45 | |
| | RAM1 (RAM) | 2 | | 17 | 4 | | 2 | | 0 | |
| | SPI (SPI_SLAVE) | 86 | | 45 | 24 | | 86 | | 44 | |

7-bitstream:

