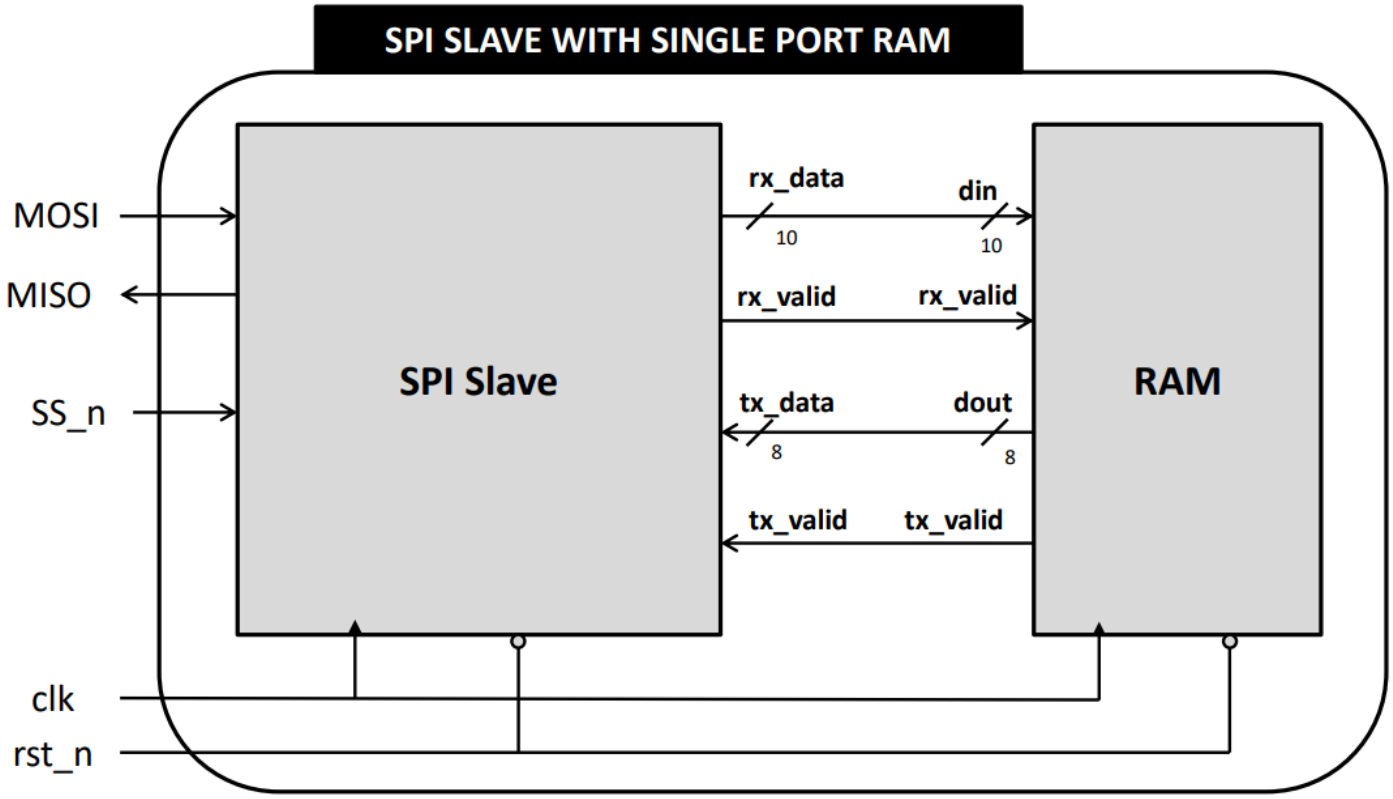
2025

SPI Slave with Single Port RAM

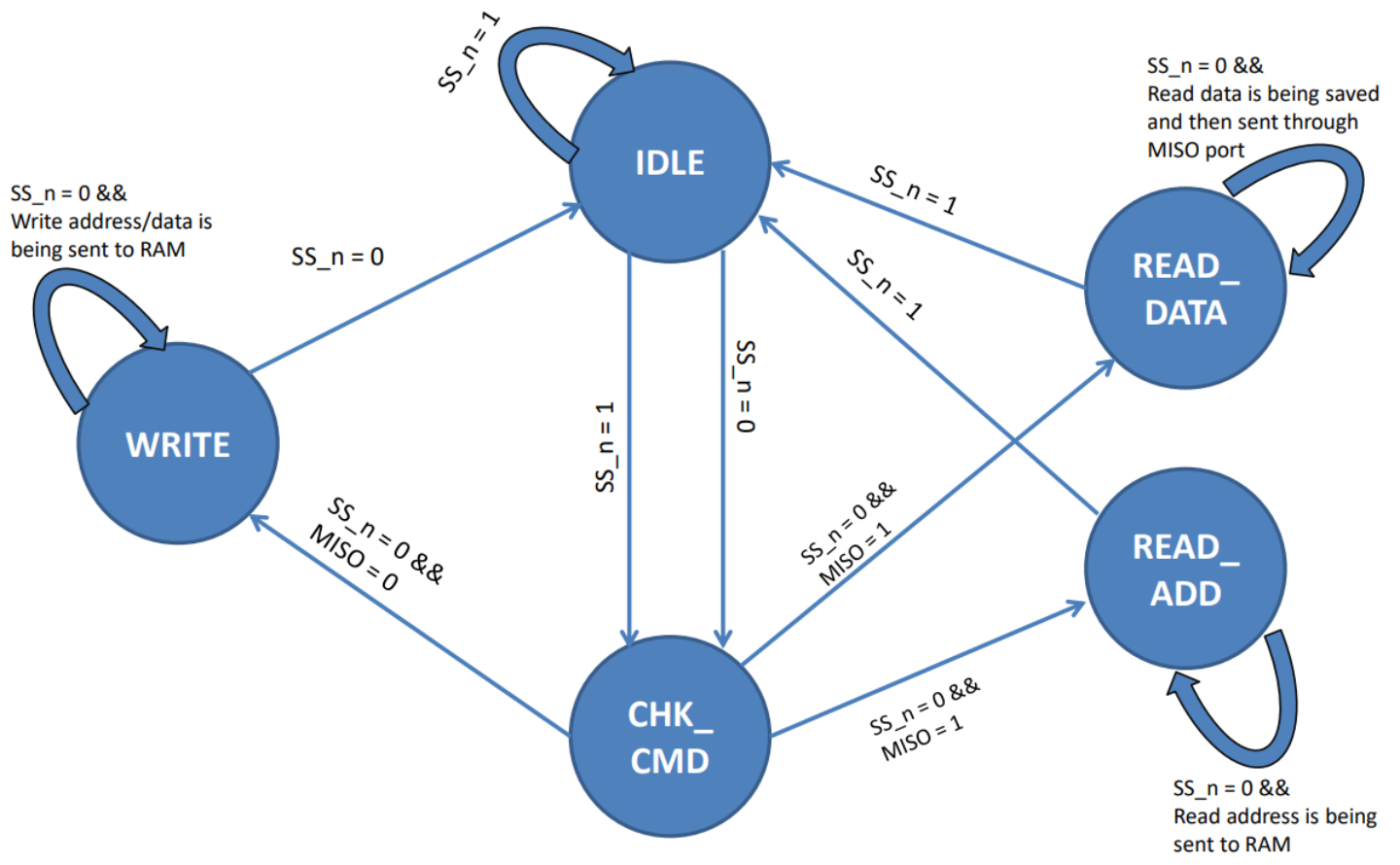
CONTENTS

SPI Slave with Single Port RAM

- Create a constraint file where the rst_n , SS_n & MOSI are connected to 3 switches, and the MISO to a led.
- The SPI slave implementation is done using FSM, we shall try out three different encoding (gray, one_hot or seq)
- We wish to operate at the highest frequency possible and so you shall choose the encoding based on the best timing report that gives the highest setup/hold slack after implementation.
- After choosing the best encoding, add a debug core such that all internals (MISO, MOSI, SS_n, rst_n & clk) can be analyzed and then generate a bitstream file.



SPI slave Transition Diagram.



Single Port Ram

```
module RAM(din, clk, rst_n, rx_vaild, dout, tx_valid);

parameter MEM_DEPTH = 256;
parameter ADDR_SIZE = 8;

input [9:0] din;
input clk, rst_n, rx_vaild;

output reg [7:0] dout;
output reg tx_valid;

reg [7:0] mem [MEM_DEPTH-1:0];

reg [7:0] address;

always @(posedge clk) begin
    if (!rst_n)begin
        dout <= 0;
        tx_valid <= 0;
        address<=0;
    end
    else begin

            case(din[9:8])
                2'b00: begin
                    if(rx_vaild)begin
                        address <= din[7:0];
                        tx_valid <= 0;
                    end
                end
                2'b01: begin
                    if(rx_vaild)begin
                        mem[address] <= din[7:0];
                        tx_valid <= 0;
                    end
                end
                2'b10:begin
                    if(rx_vaild)begin
                        address <= din[7:0];
                    end
                    tx_valid <= 1;
                end
                2'b11: begin
                    dout <= mem[address];
                    tx_valid <= 1;
                end

            endcase

    end
end


endmodule
```

SPI slave

```
module SPI_slave(
    input [7:0] tx_data,
    input ss_n, rst_n, clk, tx_valid, MOSI,
    output reg [9:0] rx_data,
    output reg rx_valid, MISO
);
    reg check_read=0;

    //counter declaration
    reg[3:0] counter;

    //flip_flops
    reg[2:0] cs, ns;

    //states assignment
    parameter IDLE      = 3'b000,
              CHK_CMD   = 3'b001,
              WRITE     = 3'b010,
              READ_DATA = 3'b011,
              READ_ADD  = 3'b100;

    //Dff model
    always @(posedge clk)
        if(!rst_n)begin
            cs <= IDLE;

        end
        else begin
            cs <= ns;
        end

    //next state logic
    always@(*)begin
        case(cs)
            IDLE: begin
                if(ss_n == 1)
                    ns = IDLE;
                else
                    ns = CHK_CMD;
            end
            CHK_CMD: begin
                case({check_read, MOSI})
                    2'b00,2'b10:
                        ns = WRITE;
                    2'b01:
```

```
                    ns = READ_ADD;
                2'b11:
                    ns = READ_DATA;
            endcase
        end
        WRITE: begin
            if(ss_n==0)
                ns = WRITE;
            else
                ns = IDLE;
        end
        READ_ADD: begin
            if(ss_n==0)
                ns = READ_ADD;
            else
                ns = IDLE;
        end
        READ_DATA: begin
            if(ss_n==0)
                ns = READ_DATA;
            else
                ns = IDLE;
        end
        default: ns = IDLE;
    endcase
end



//output logic
always@(posedge clk)begin
    if (!rst_n)begin
        rx_data <= 0;
        rx_valid <= 0;
        check_read <= 0;
        MISO <= 0;
        counter <= 0;
    end
    else begin
        case(cs)
            IDLE:
                begin
                    rx_data <= 0;
                    rx_valid<=0;
```

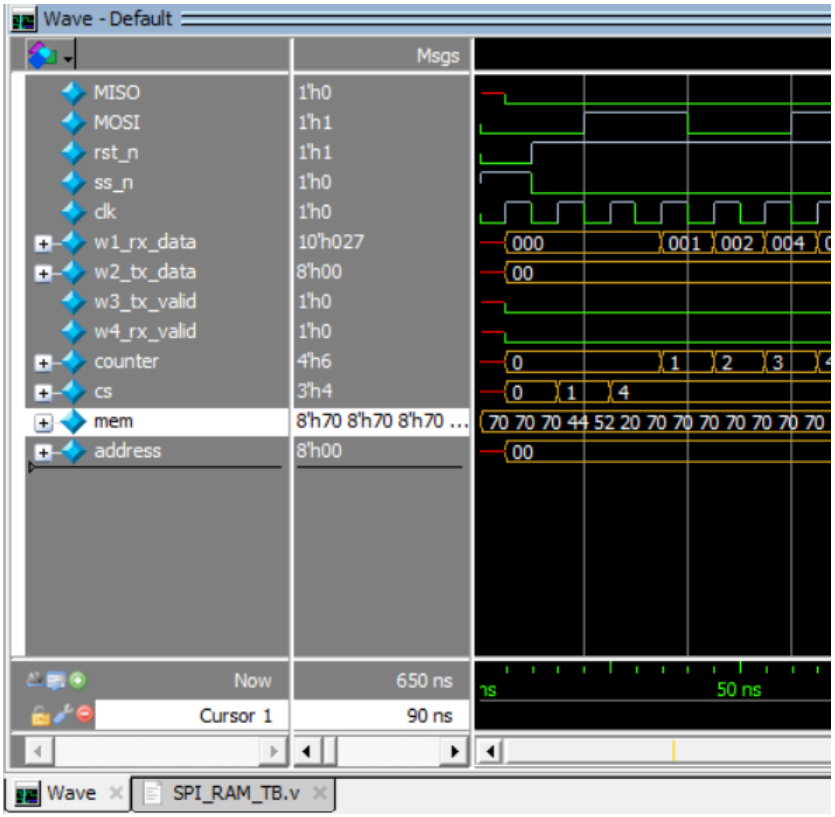
```
                    rx_valid<=0;
                    MISO <= 0;
                end

            CHK_CMD:
                begin
                    rx_data <= 0;
                    rx_valid <= 0;
                    MISO <= 0;
                end

            WRITE: begin
                rx_data <= {rx_data[8:0],MOSI};
                counter <= counter+1;
                if(counter == 10)begin
                    rx_valid <= 1;
                    counter <= 0;
                end
                else
                    rx_valid <= 0;
            end

            READ_ADD: begin
                rx_data <= {rx_data[8:0],MOSI};
                counter <= counter+1;
                if(counter == 10)begin
                    rx_valid <= 1;
                    counter <= 0;
                    check_read <=  1; /// flag to know if it's ( address or data )
                end
                else
                    rx_valid <= 0;
            end

            READ_DATA: begin
                rx_data <= {rx_data[8:0],MOSI};
                counter <= counter+1;
                if(counter == 10)begin
                    rx_valid <= 1;
                    counter <= 0;
                    check_read <=  1; /// flag to know if it's ( address or data )
                end
                else
                    rx_valid <= 0;

                if((tx_valid==1)&&(counter<7||counter==7))
                    MISO <= tx_data[7-counter];
                else
                    MISO<=0;
                end
        endcase
    end
end

endmodule
```

SPI Wrapper

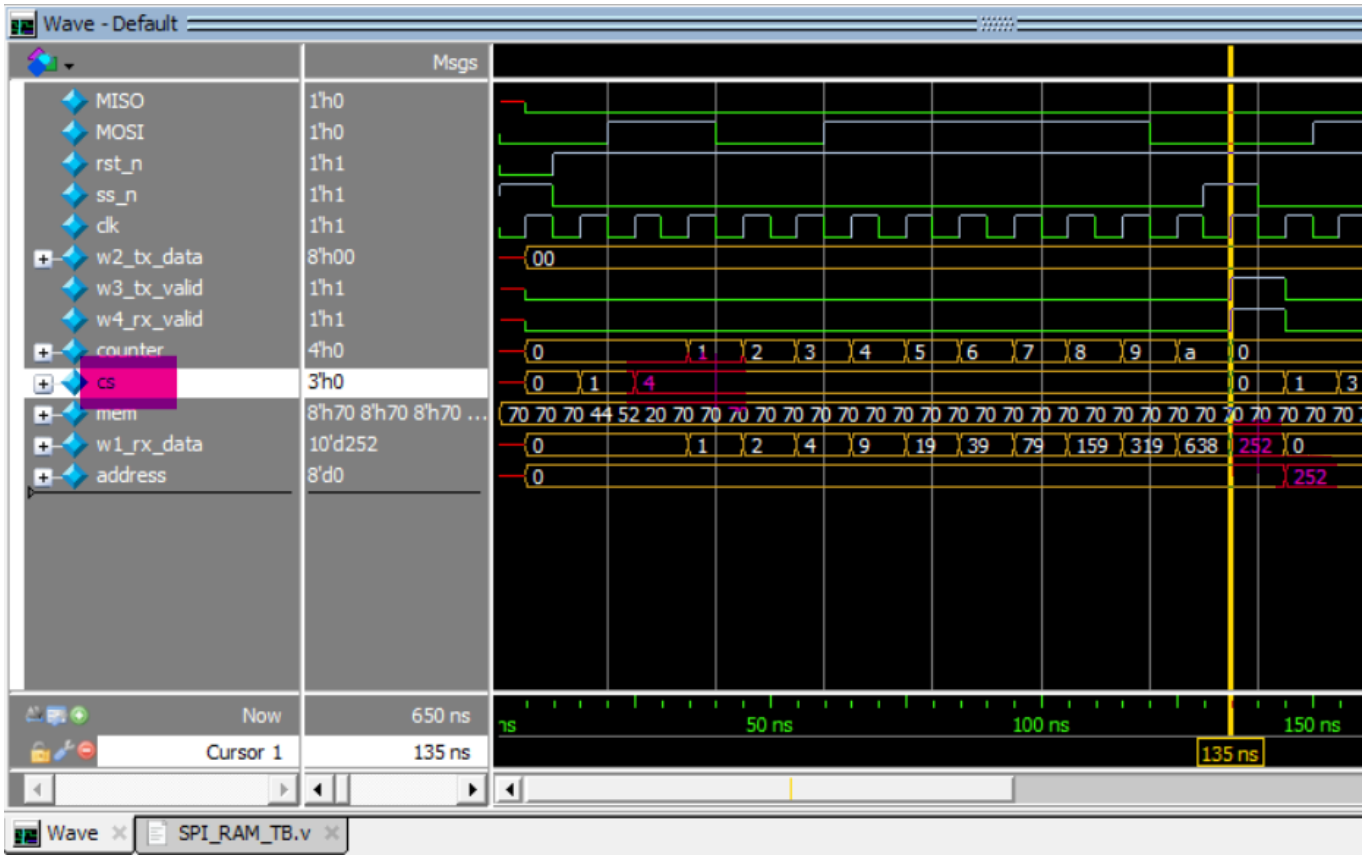
```
module SPI_w_RAM #(
    parameter MEM_DEPTH = 256,
    parameter ADDR_SIZE = 8
)(
    output MISO,
    input MOSI,
    input rst_n , ss_n , clk
);

wire[9:0]w1_rx_data;
wire[7:0]w2_tx_data;
wire w3_tx_valid , w4_rx_valid;

//connecting modules

SPI_slave S1(
    .tx_data(w2_tx_data),
    .ss_n(ss_n),
    .rst_n(rst_n),
    .clk(clk),
    .tx_valid(w3_tx_valid),
    .MOSI(MOSI),
    .rx_data(w1_rx_data),
    .rx_valid(w4_rx_valid),
    .MISO(MISO)
);

RAM #(
    .MEM_DEPTH(MEM_DEPTH),
    .ADDR_SIZE(ADDR_SIZE)
)R1(
    .din(w1_rx_data),
    .clk(clk),
    .rst_n(rst_n),
    .rx_vaild(w4_rx_valid),
    .dout(w2_tx_data),
    .tx_valid(w3_tx_valid)
);

endmodule
```

Snippets from the waveforms captured from QuestaSim

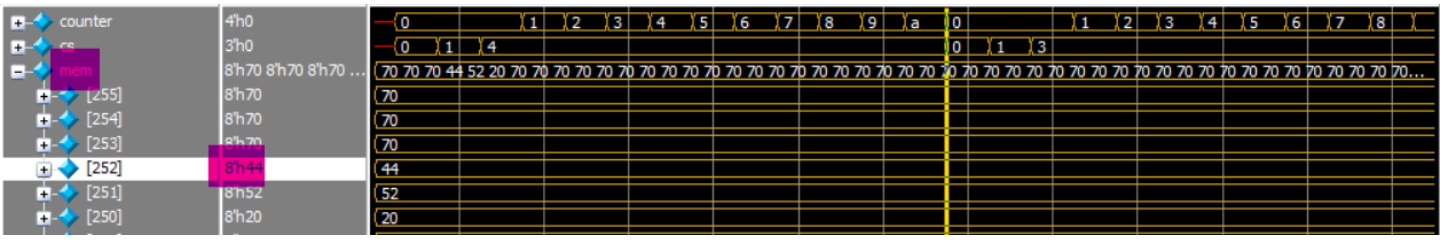
1_testing synchronous reset



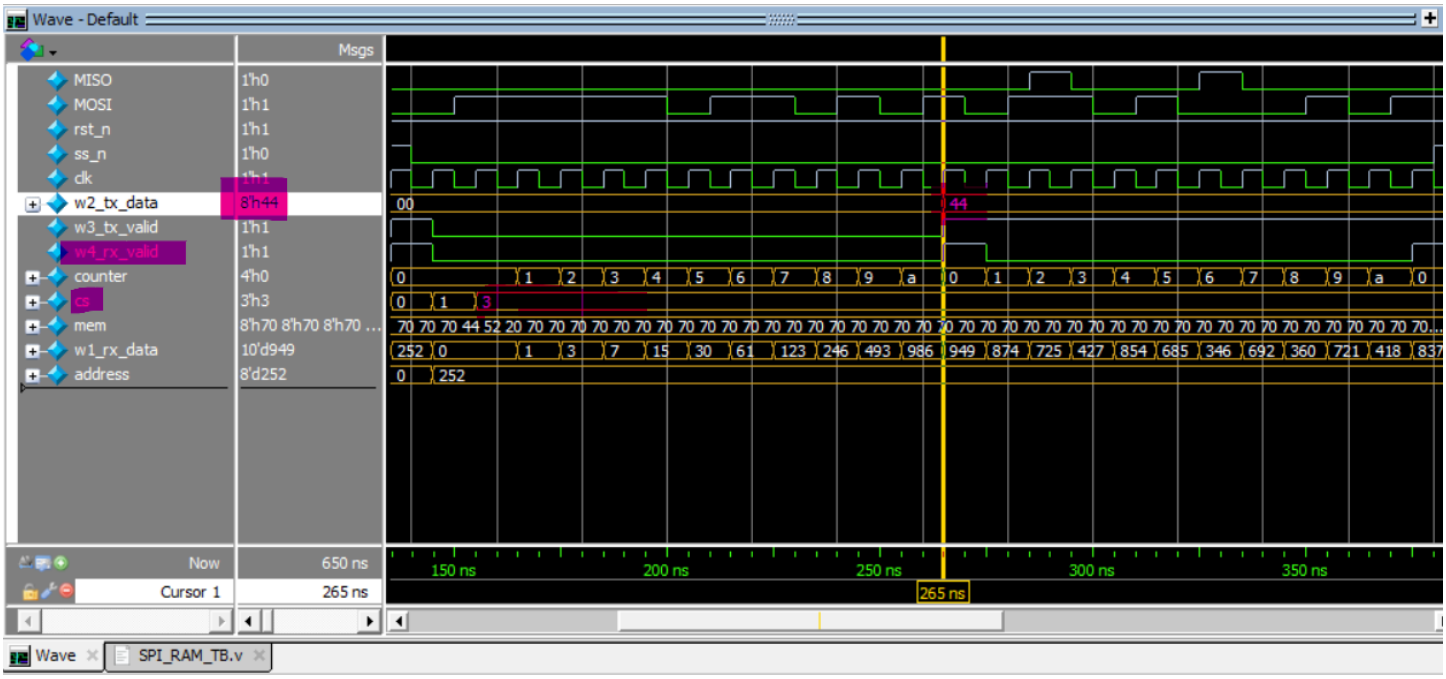
2_testing read address



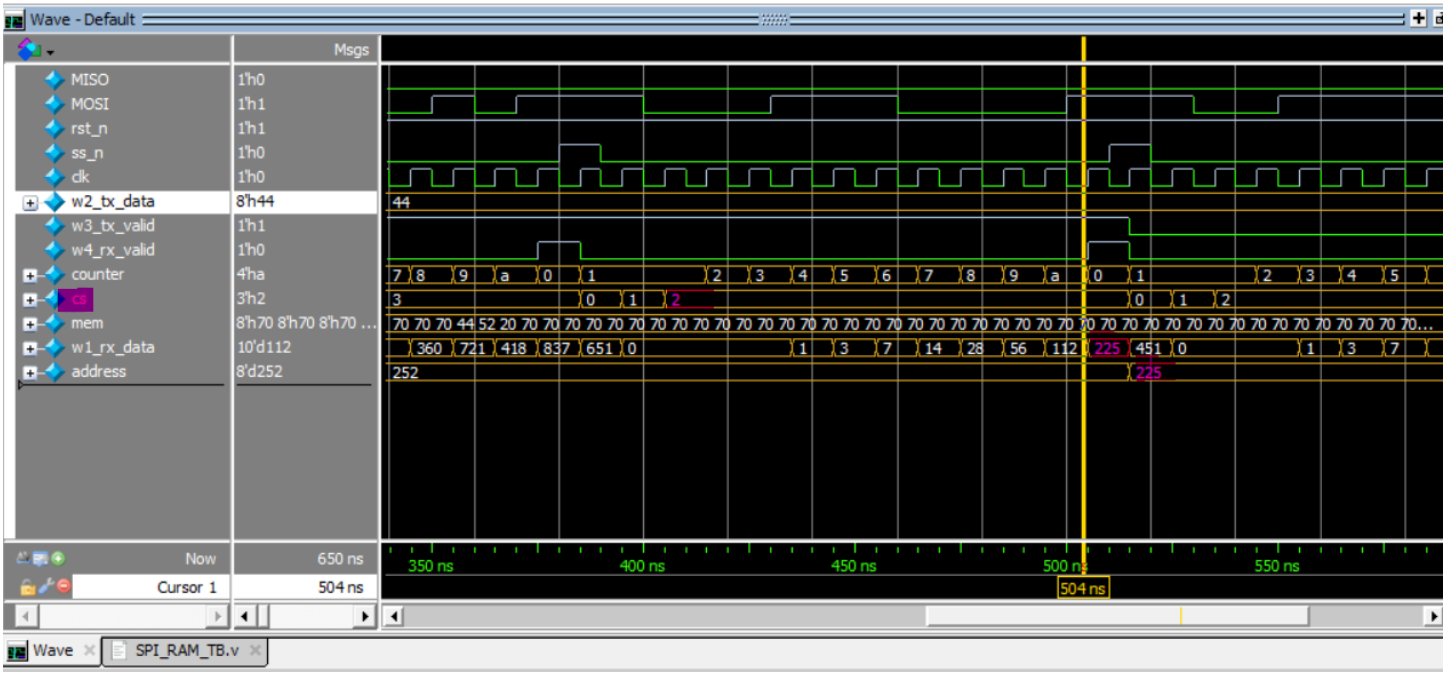
3_what was in the address



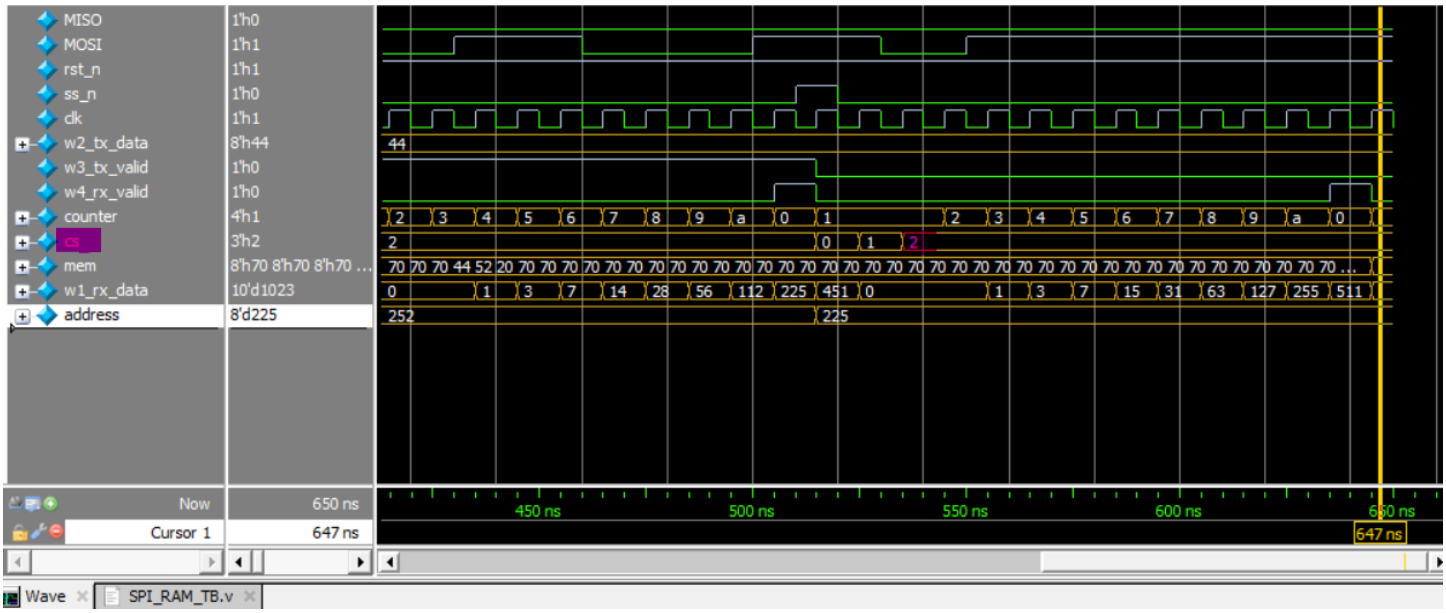
4_testing read data



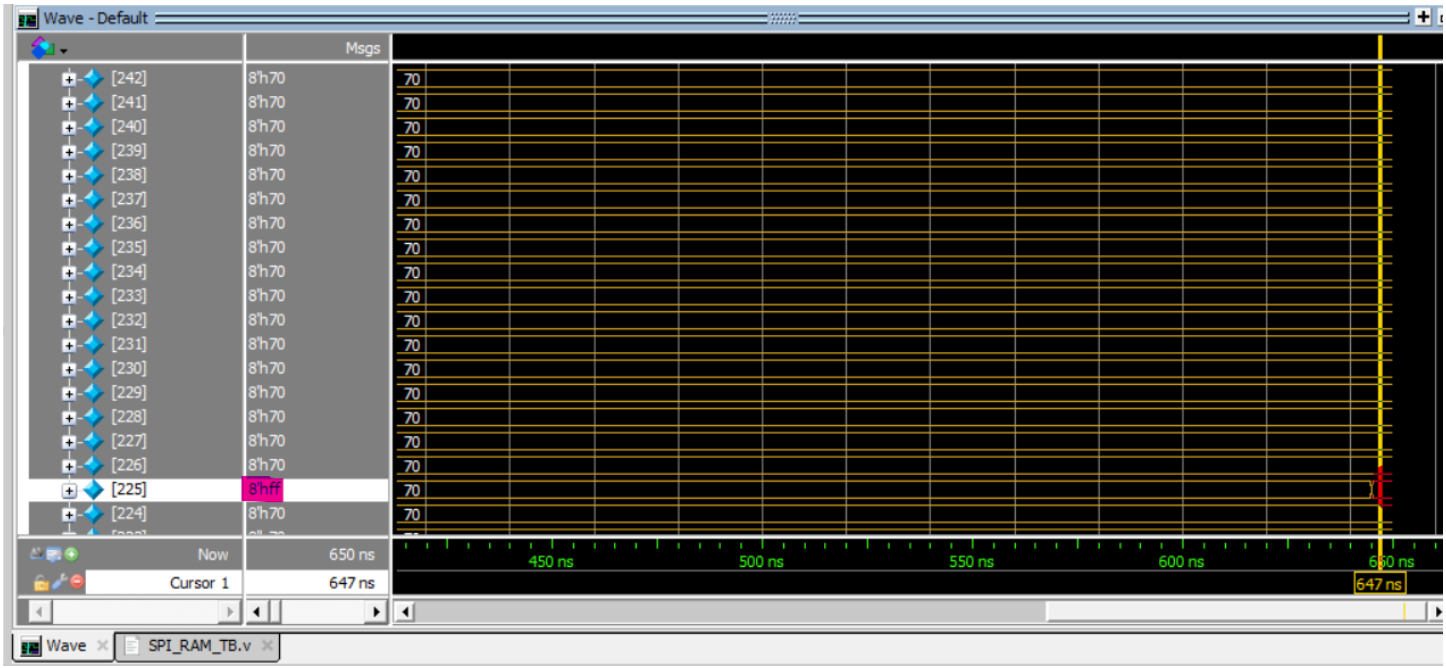
5_test write address



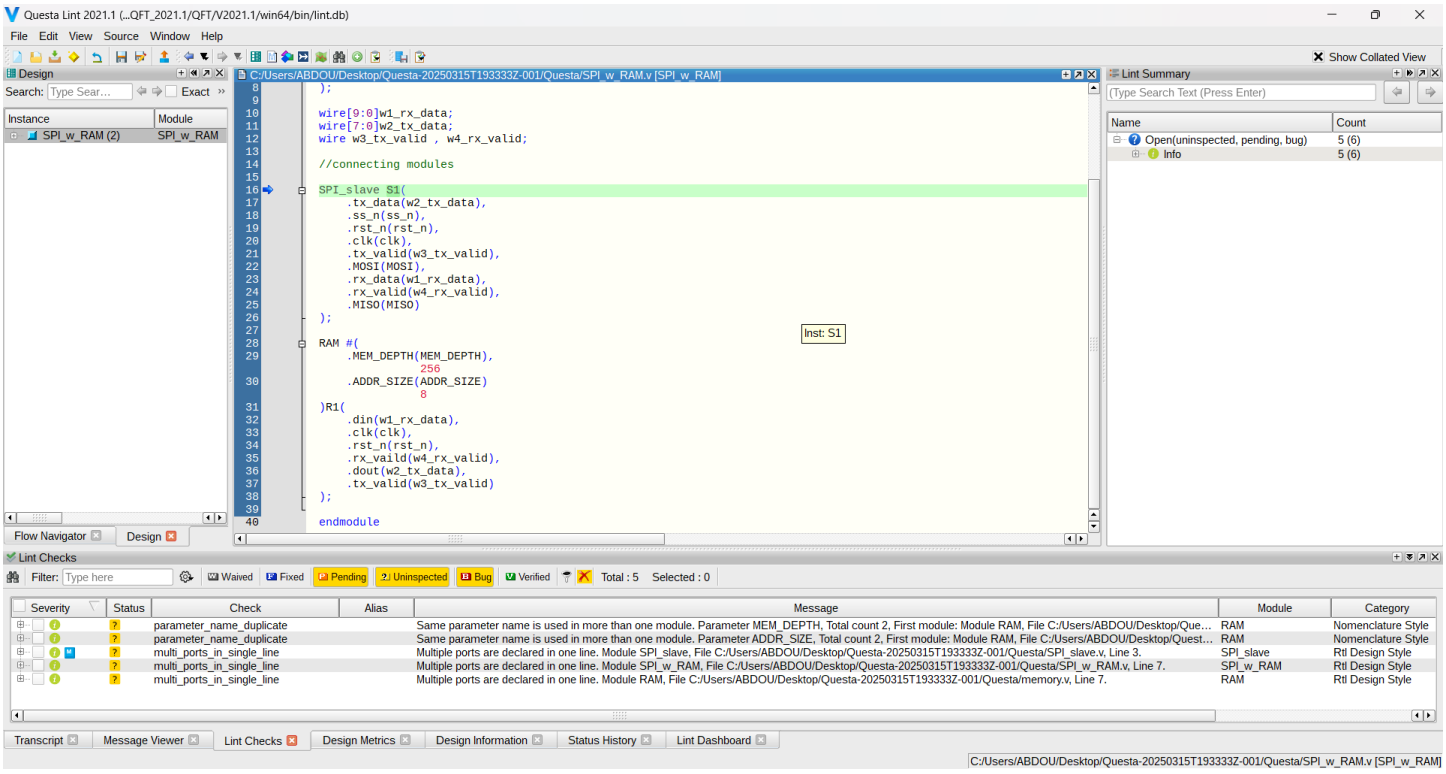
6_testing write data



7_what we wrote in TB is written in the address we determined



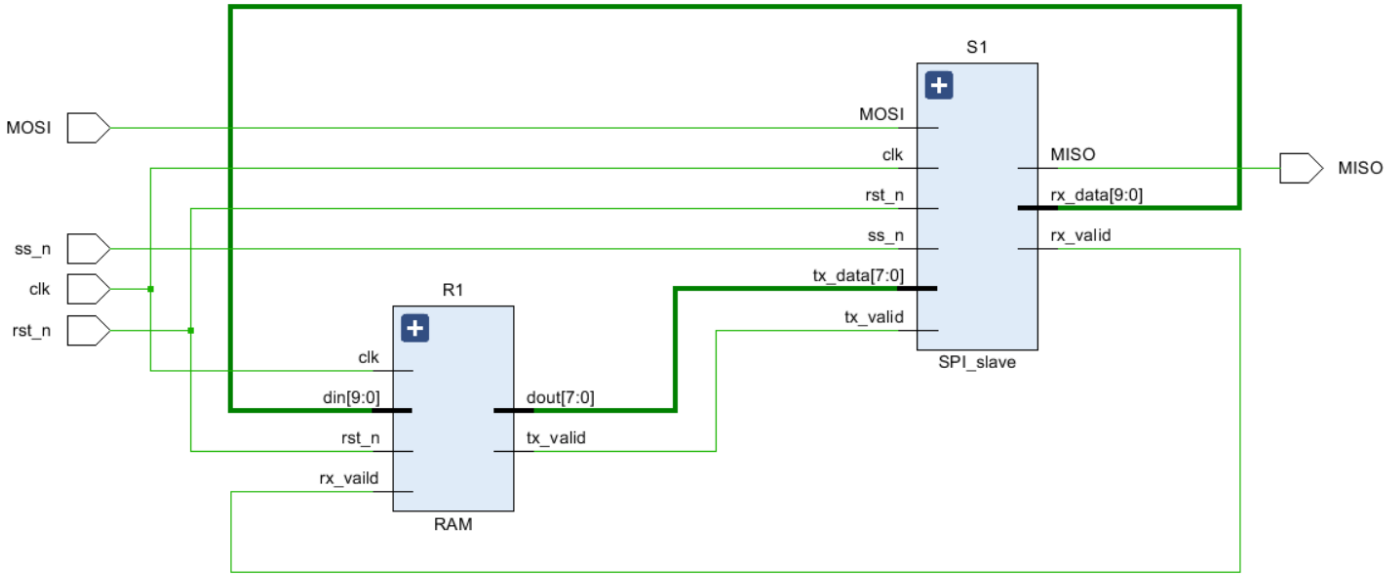
Questa Lint



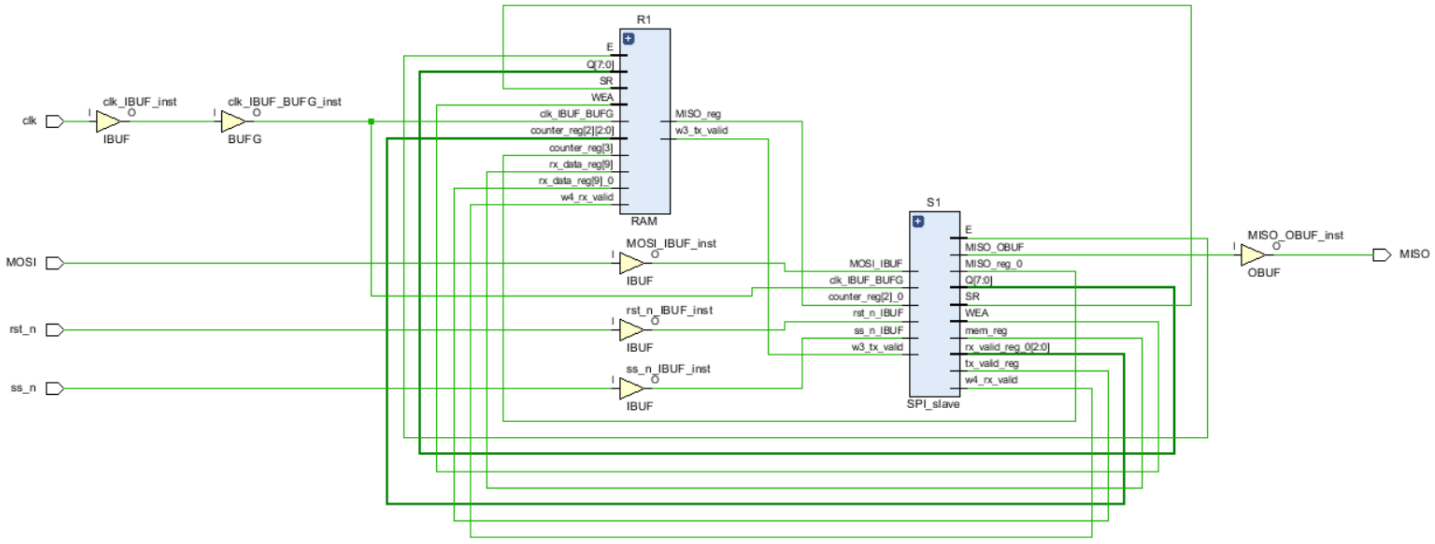
VIVADO

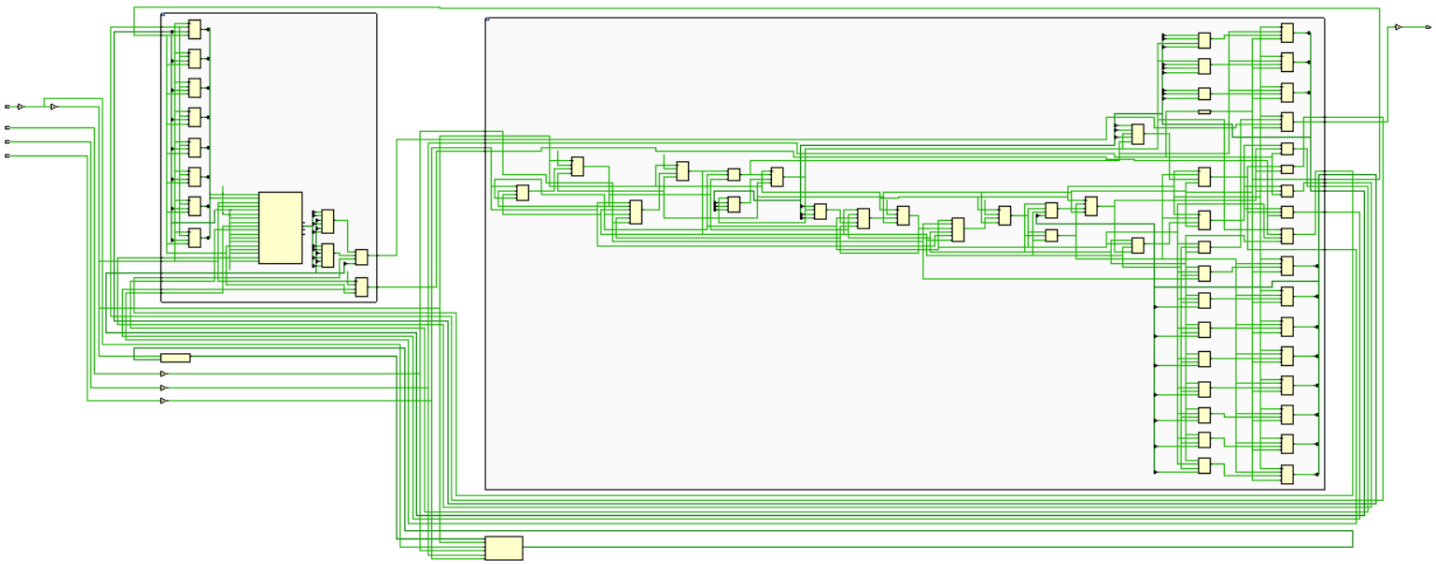
(* fsm_encoding = "gray" *)

Schematic elaboration



Schematic Synthesis





Encoding from report

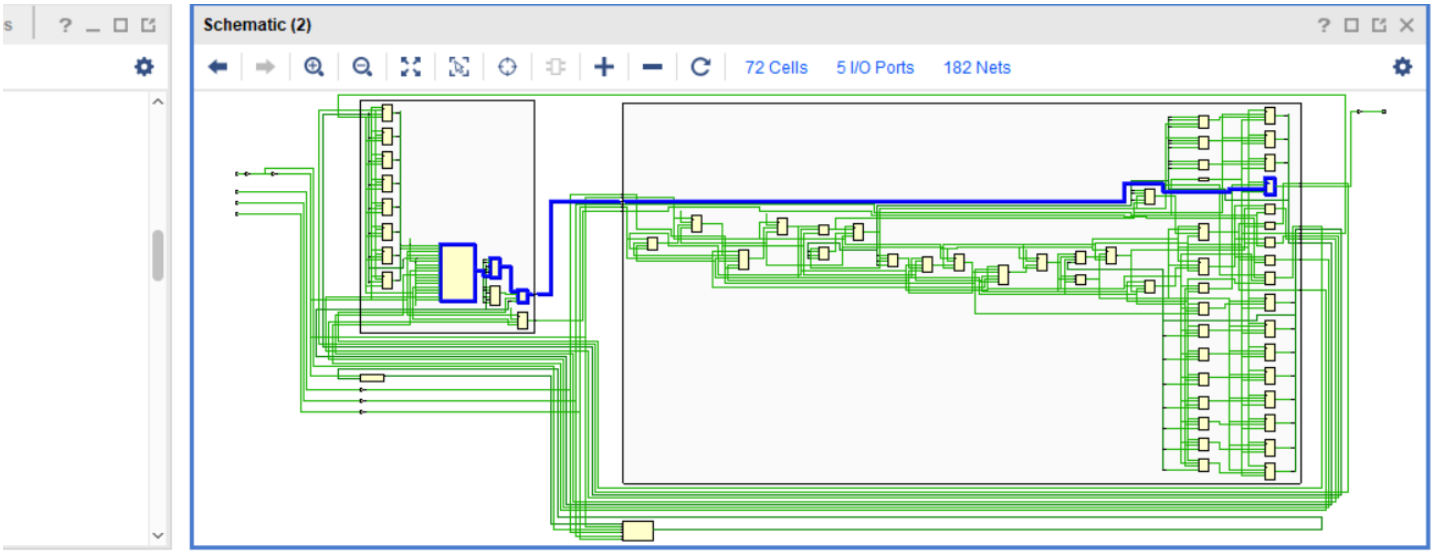
State	New Encoding	Previous Encoding
IDLE	000	000
CHK_CMD	001	001
WRITE	011	010
READ_ADD	010	100
READ_DATA	111	011

Timing report

rts | Design Runs | Timing ×

Design Timing Summary

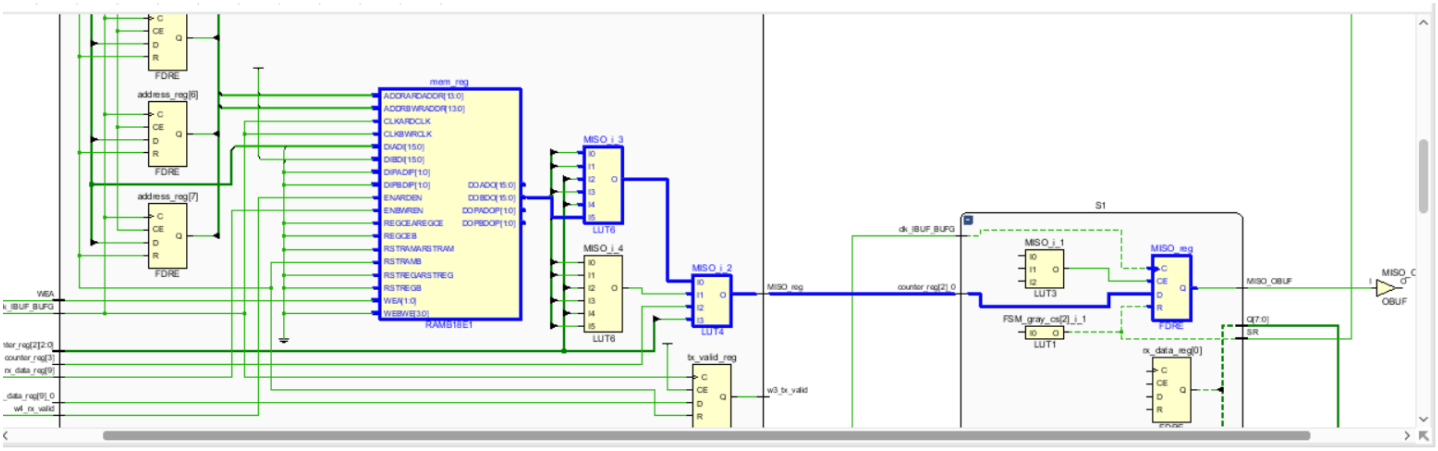
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.445 ns	Worst Hold Slack (WHS):	0.142 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0



Design Runs | Timing

Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirement
Path 1	5.445	2	3	1	R1/mem_reg/CLKBWRCLK	S1/MISO_reg/D	4.404	2.702	1.702	10.0



Tcl Console | Messages | Log | Reports | Design Runs | Timing

Intra-Clock Paths - sys_clk_pin - Setup

General Information
Timer Settings
Design Timing Summary

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requirem
Path 1	5.445	2	3	1	R1/mem_reg/CLKBWRCLK	S1/MISO_reg/D	4.404	2.702	1.702	
Path 2	6.978	1	2	3	S1/rx_data_reg[9]/C	R1/mem_reg/WEA[0]	2.310	0.751	1.559	

Implementation

Hierarchy

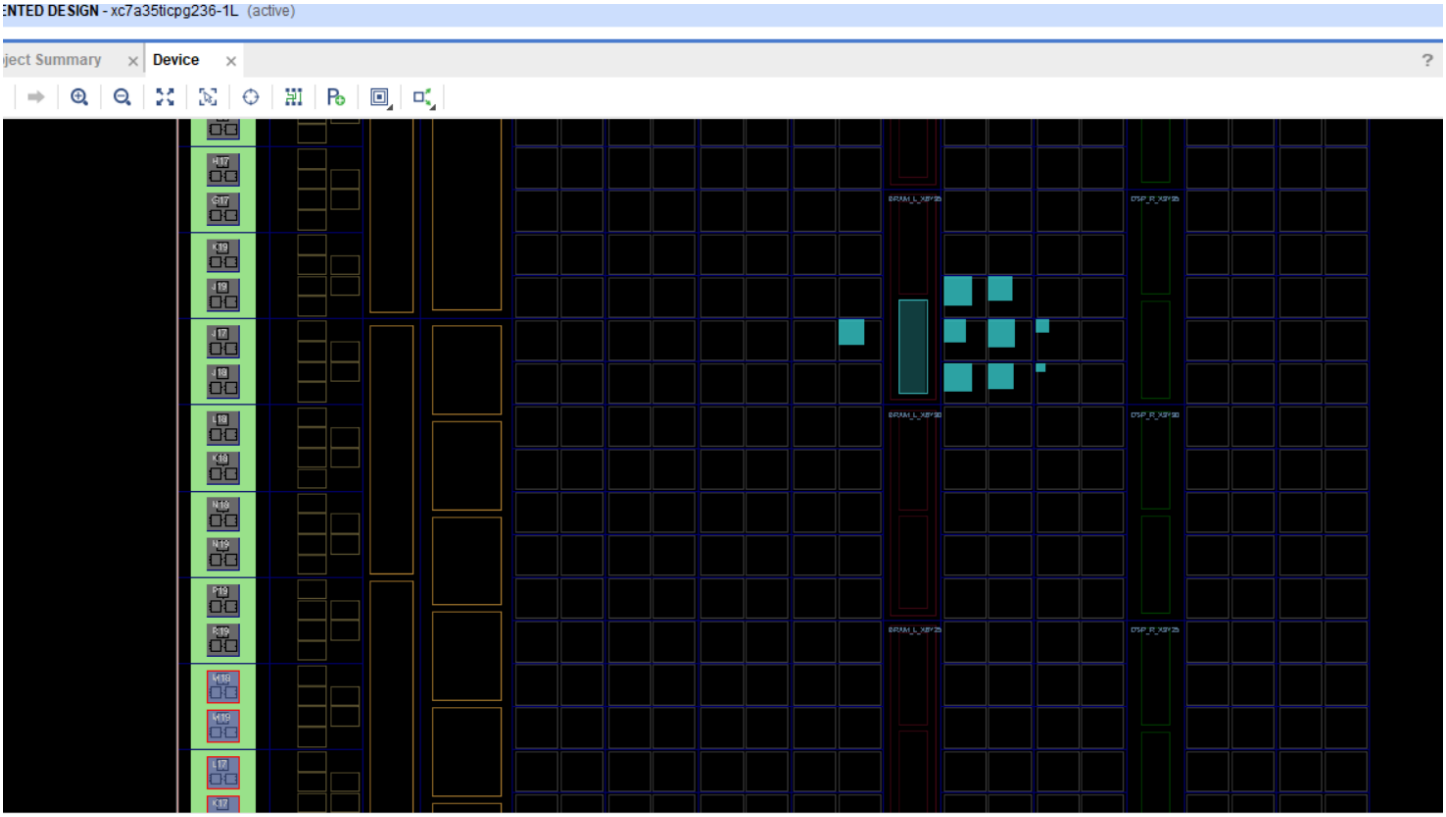
Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
SPI_w_RAM	28	29	9	28	17	0.5	5	1
R1 (RAM)	3	9	3	3	0	0.5	0	0
S1 (SPI_slave)	25	20	9	25	16	0	0	0

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.578 ns	Worst Hold Slack (WHS):	0.055 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:	81	Total Number of Endpoints:	81	Total Number of Endpoints:	32

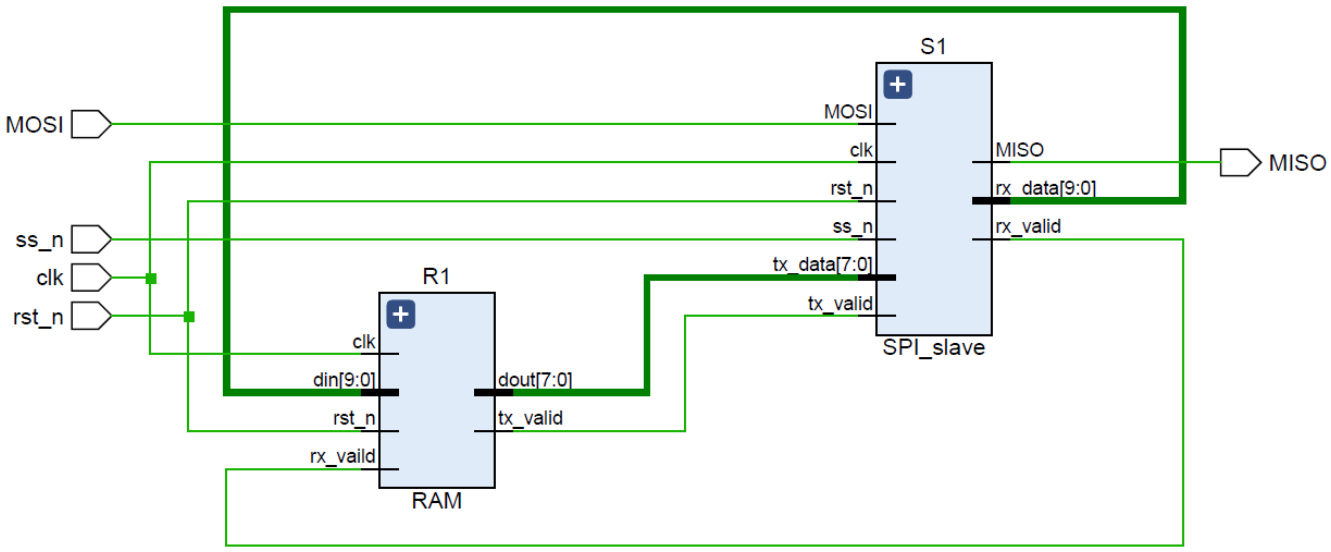
All user specified timing constraints are met.

Device:

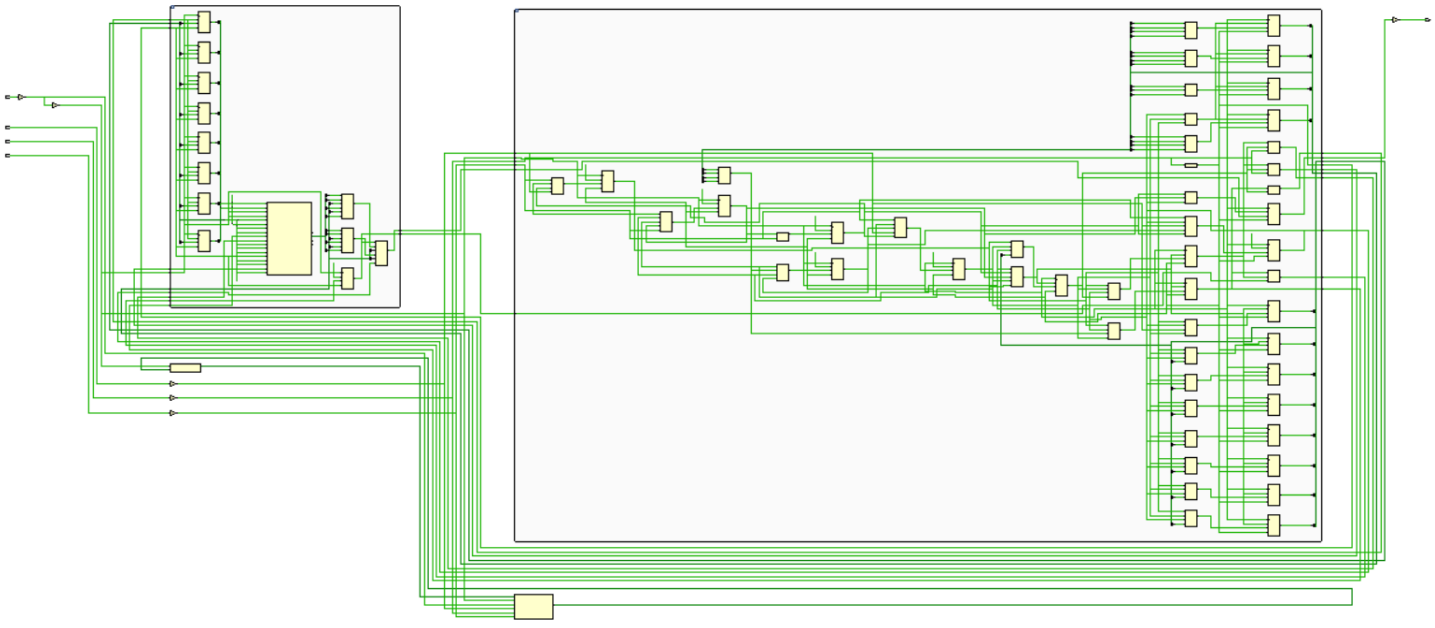


(* fsm_encoding = "one_hot" *)

Elaborator



Schematic Synthesis:



Encoding Report:

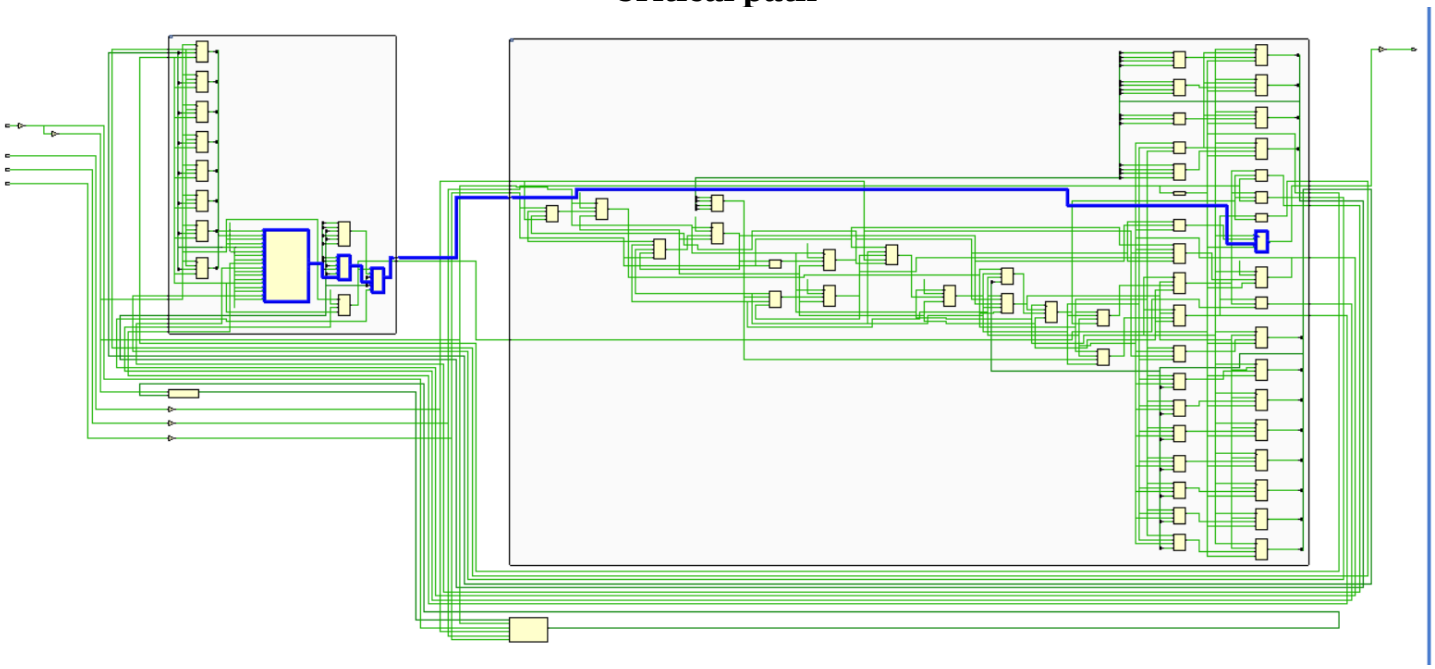
State	New Encoding	Previous Encoding
IDLE	00001	000
CHK_CMD	00010	001
WRITE	00100	010
READ_ADD	01000	100
READ_DATA	10000	011

Timing report:

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.898 ns	Worst Hold Slack (WHS):	0.142 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

Critical path



Implementation:

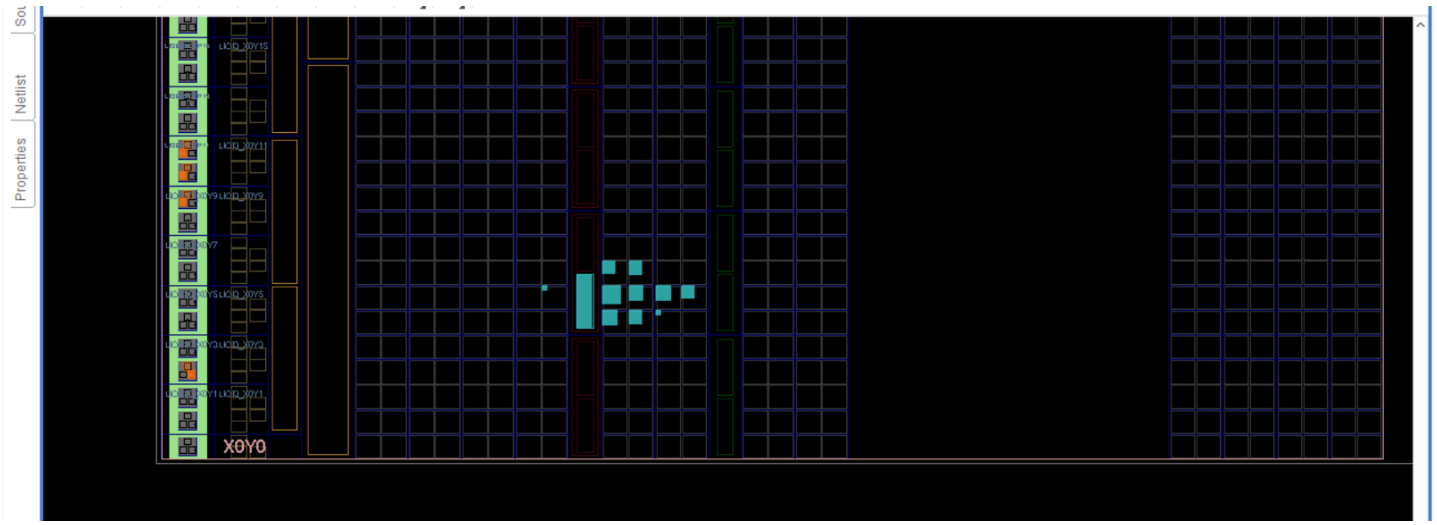
Reports | Design Runs | Power | Methodology | Timing | **Utilization**

Hierarchy

Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
SPI_w_RAM	30	31	10	30	20	0.5	5	1
R1 (RAM)	3	9	4	3	0	0.5	0	0
S1 (SPI_slave)	27	22	9	27	19	0	0	0

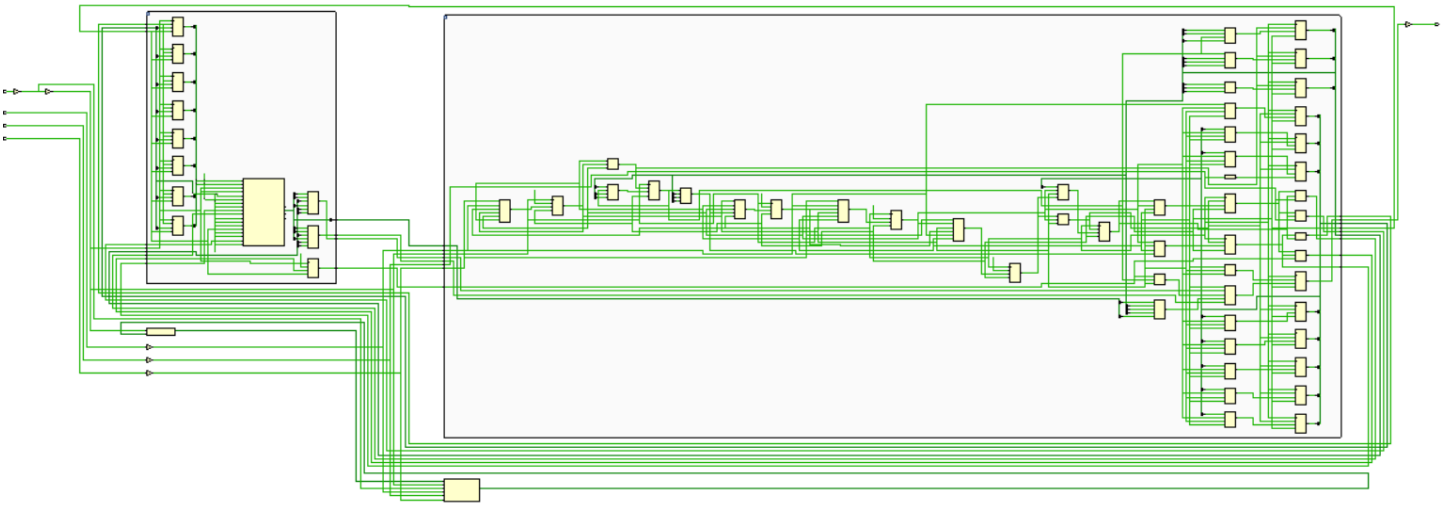
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.414 ns	Worst Hold Slack (WHS):	0.047 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



(* fsm_encoding = "sequential" *)

Synthesis Schema:



Encoding report

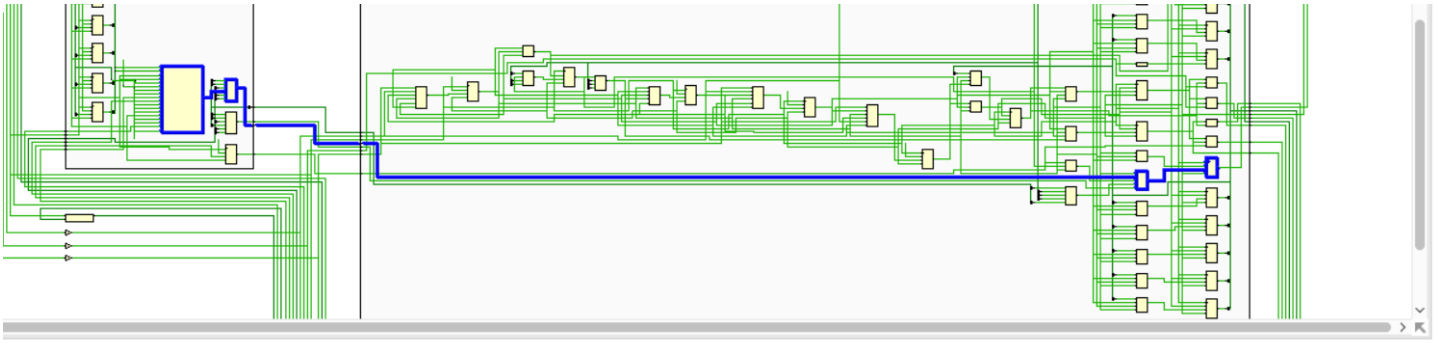
State	New Encoding	Previous Encoding
IDLE	000	000
CHK_CMD	001	001
WRITE	010	010
READ_ADD	011	100
READ_DATA	100	011

Timing report:

Design Runs | Timing

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.682 ns	Worst Hold Slack (WHS):	0.142 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



Messages | Log | Reports | Design Runs | Timing

Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requ
Path 1	5.682	2	3	1	R1/mem_reg/CLKBWRCLK	S1/MISO_reg/D	4.167	2.702	1.465	
Path 2	6.978	1	2	3	S1/rx_data_reg[9]/C	R1/mem_reg/WEA[0]	2.310	0.751	1.559	
Path 3	6.978	1	2	3	S1/rx_data_reg[9]/C	R1/mem_reg/WEA[1]	2.310	0.751	1.559	
Path 4	7.067	1	2	3	S1/rx_data_reg[9]/C	R1/mem_reg/ENBWREN	2.310	0.751	1.559	

Implementation

Reports | Design Runs | Power | Methodology | Timing | Utilization

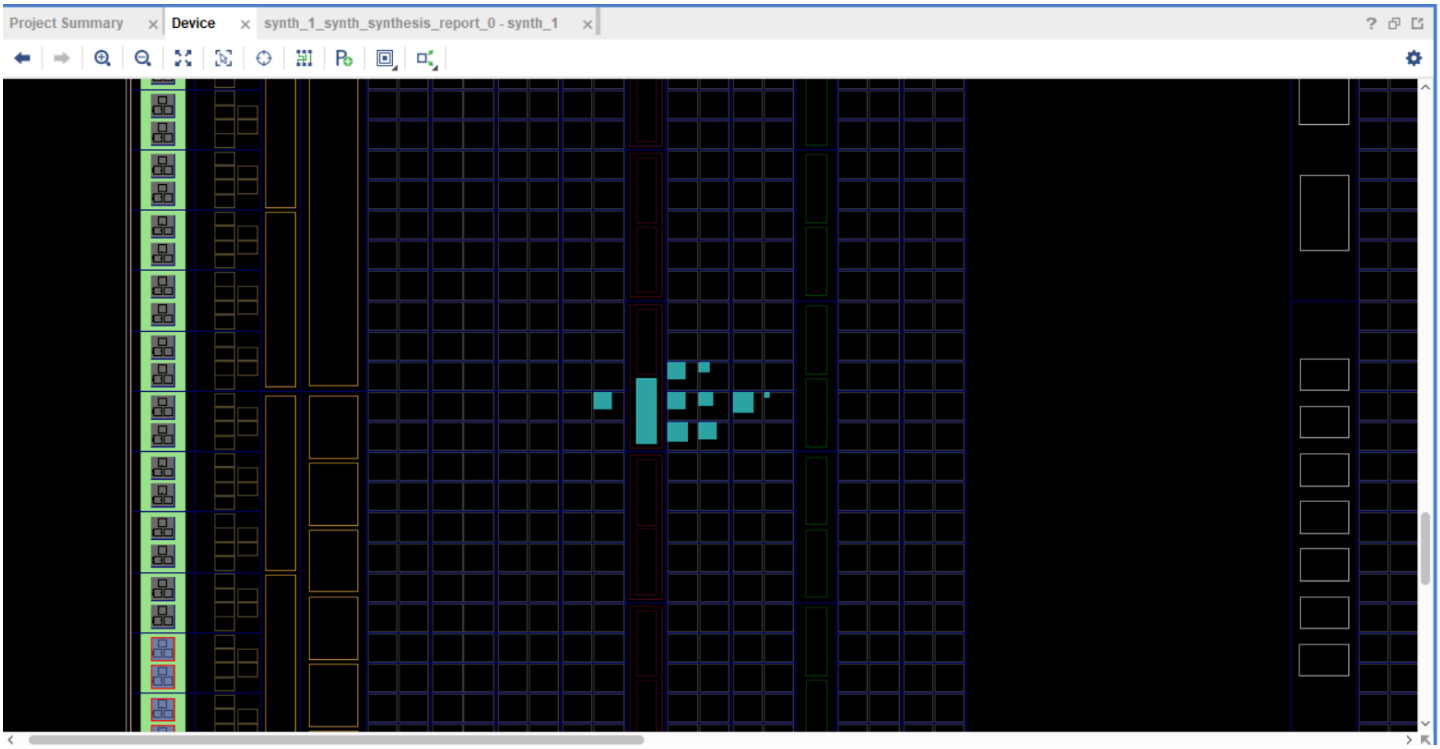
Hierarchy

Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
SPI_w_RAM	29	29	9	29	19	0.5	5	1
R1 (RAM)	2	9	3	2	0	0.5	0	0
S1 (SPI_slave)	27	20	8	27	19	0	0	0

Design Runs | Power | Methodology | Timing | Utilization

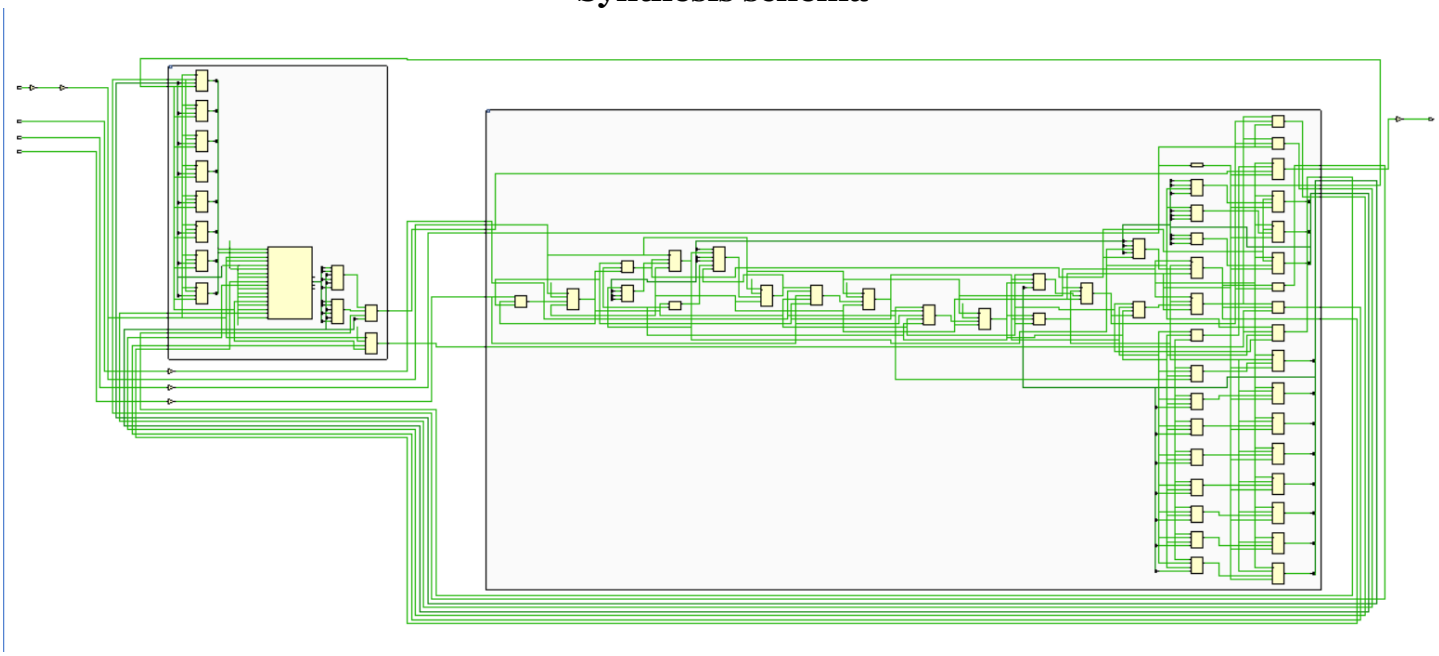
Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.503 ns	Worst Hold Slack (WHS):	0.044 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



(* fsm_encoding = "johnson" *)

Synthesis schema



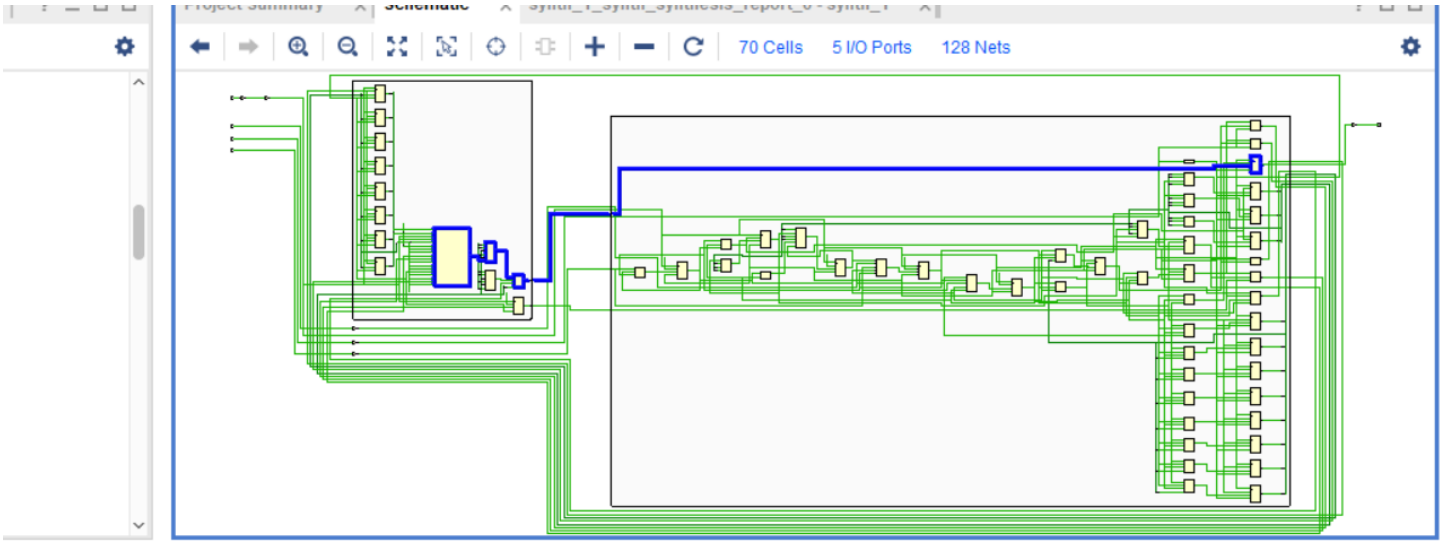
Encoding Report:

State	New Encoding	Previous Encoding
IDLE	000	000
CHK_CMD	100	001
WRITE	110	010
READ_ADD	111	100
READ_DATA	011	011

Timing report

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.445 ns	Worst Hold Slack (WHS):	0.142 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



Design Runs | Timing

Intra-Clock Paths - sys_clk_pin - Setup

Name	Slack	Levels	Routes	High Fanout	From	To	Total Delay	Logic Delay	Net Delay	Requireme
Path 1	5.445	2	3	1	R1/mem_reg/CLKBWRCLK	S1/MISO_reg/D	4.404	2.702	1.702	10
Path 2	6.978	1	2	3	S1/rx_data_reg[9]/C	R1/mem_reg/WEA[0]	2.310	0.751	1.559	10

Implementation

Reports | Design Runs | Power | Methodology | Timing | Utilization

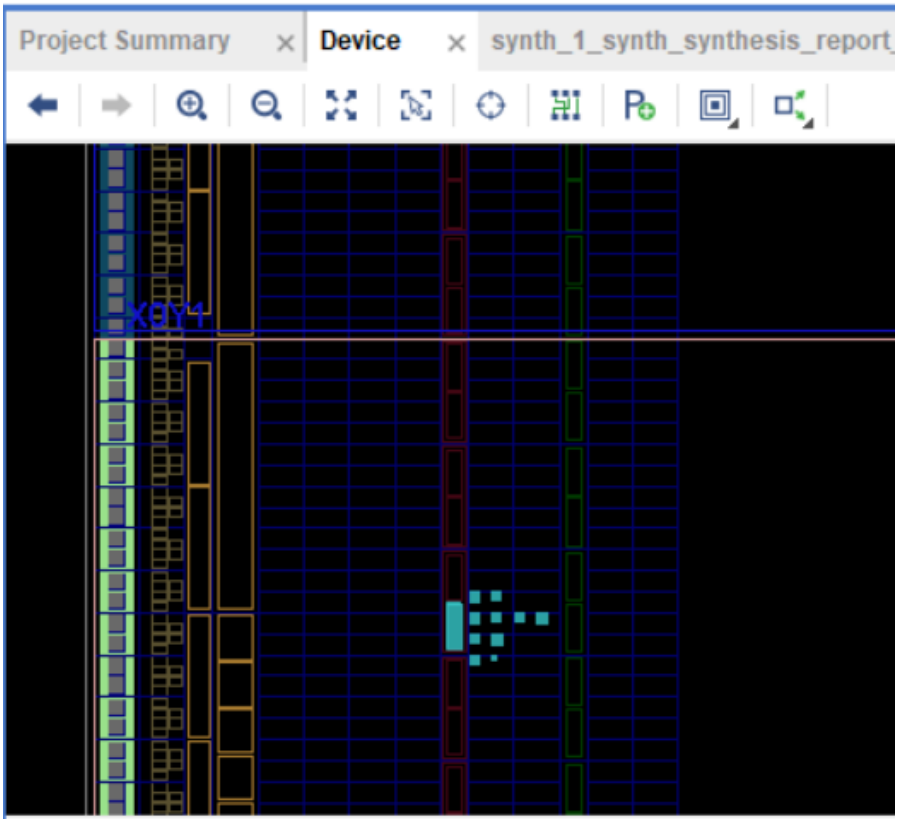
Hierarchy

Name	Slice LUTs (20800)	Slice Registers (41600)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
SPI_w_RAM	29	29	10	29	18	0.5	5	1
R1 (RAM)	3	9	4	3	0	0.5	0	0
S1 (SPI_slave)	26	20	9	26	17	0	0	0

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.635 ns	Worst Hold Slack (WHS):	0.044 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

Project Summary | Device | synth_1_synth_synthesis_report



Massages:

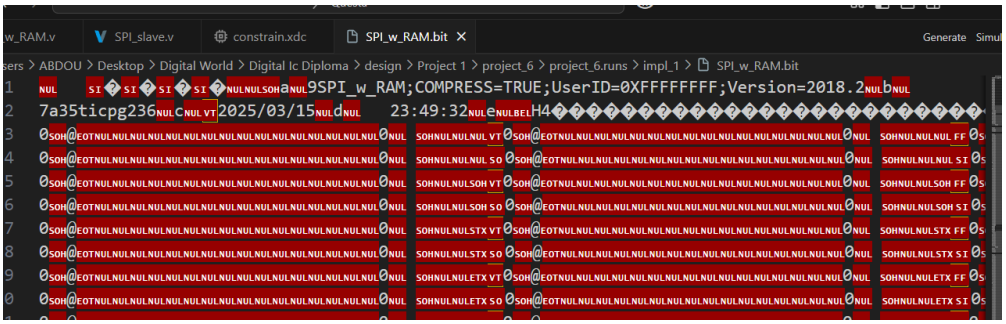
Tcl Console | Messages | Log | Reports | Design Runs | Power | Methodology | Timing | Utilization

Warning (1) | Info (239) | Status (490) | Show All

Synthesis (1 warning)

[Constraints 18-5210] No constraint will be written out.

write_bitstream Complete ✓



THANK YOU