Project 2

Group members
Ahmed Awad-Allah Mohamed Ali
Ibrahim Khaled Mohamed
Farah Ahmed Fathy

SPI

Design:

```
module RAM(din,rx_valid,clk,rst_n,dout,tx_valid);
parameter MEM_DEPTH=256;
parameter ADDR_SIZE=8;
input [9:0] din;
input rx_valid,clk,rst_n;
output reg [7:0] dout;
output reg tx_valid;
reg [ADDR_SIZE-1:0] ram [MEM_DEPTH-1:0];
reg [7:0] write_addr,read_addr;
always @(posedge clk or negedge rst_n) begin
    if (!rst_n) begin
        dout <= 0;
        tx_valid <= 0;
    end
    else if (din[9:8] == 2'b00 && rx_valid == 1) begin
        write_addr <= din[7:0];
        tx_valid <= 0;
    end
    else if (din[9:8] == 2'b01 && rx_valid == 1) begin
        ram[write_addr] <= din[7:0];
        tx_valid <= 0;
    end
    else if (din[9:8] == 2'b10 && rx_valid == 1) begin
        read_addr <= din[7:0];
        tx_valid <= 0;
    end
    else if (din[9:8] == 2'b11 && rx_valid == 1) begin
        dout <= ram[read_addr];
        tx_valid <= 1;
    end
end
endmodule
```

```
module SPI_SLAVE(MOSI,MISO,SS_n,clk,rst_n,rx_data,rx_valid,tx_data,tx_valid);
parameter IDLE=3'b000;
parameter CHK_CMD=3'b001;
parameter WRITE=3'b010;
parameter READ_ADD=3'b011;
parameter READ_DATA=3'b100;
input MOSI,SS_n,clk,rst_n,tx_valid;
input [7:0] tx_data;
output reg [9:0] rx_data;
output reg MISO,rx_valid;
reg check_addr=0;
reg [3:0] count;
reg [2:0] count2;
(* fsm_encoding = "one_hot" *)
reg [2:0] cs,ns;

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

always @(cs,MOSI,SS_n,tx_valid) begin
    case(cs)
    IDLE: begin
        if(SS_n == 1) begin
            ns = IDLE;
        end
        else begin
            ns = CHK_CMD;
        end
    end
    CHK_CMD: begin
        if (SS_n == 1) begin
            ns = IDLE;
        end
        else if (SS_n == 0 && MOSI == 0) begin
            ns = WRITE;
        end
        else if (SS_n == 0 && MOSI == 1) begin
            if (check_addr == 0) begin
                ns = READ_ADD;
            end
            else begin
                ns = READ_DATA;
            end
        end
```

```
    end
    WRITE: begin
        if (SS_n == 1) begin
            ns = IDLE;
        end
        else begin
            ns = WRITE;
        end
    end
    READ_ADD: begin
        if (SS_n == 1) begin
            ns = IDLE;
        end
        else begin
            ns = READ_ADD;
        end
    end
    READ_DATA: begin
        if (SS_n == 1) begin
            ns = IDLE;
        end
        else begin
            ns = READ_DATA;
        end
    end
    endcase
end

always @(posedge clk) begin
    case(cs)
    IDLE: begin
        rx_valid <= 0;
        rx_data <= 0;
        count <=  0;
        count2 <= 0;
    end
    CHK_CMD: begin
        rx_valid <= 0;
        rx_data <= 0;
        count <= 0;
        count2 <= 0;
    end
    WRITE: begin
        if(count == 10) begin
            rx_valid <= 1;
        end
        else begin
            count <= count+1;
            rx_data[9-count] <= MOSI;
        end
```

```
    end
    READ_ADD: begin
        if(count == 10) begin
            rx_valid <= 1;
            check_addr <=1;
        end
        else begin
            count <= count+1;
            rx_data[9-count] <= MOSI;
        end
    end
    READ_DATA: begin
        if(count == 10) begin
            rx_valid <= 1;
            check_addr <= 0;
            if(tx_valid == 1) begin
                MISO <= tx_data[7-count2];
                count2 <= count2 + 1;
            end
        end
        else begin
            count <= count+1;
            rx_data[9-count] <= MOSI;
        end
    end
    endcase
end

endmodule
```

```
module project_2(MOSI,MISO,SS_n,clk,rst_n);
input MOSI,SS_n,clk,rst_n;
output MISO;
wire [9:0] rx_data;
wire [7:0] tx_data;
wire rx_valid,tx_valid;
RAM
RAM1(.din(rx_data),.rx_valid(rx_valid),.clk(clk),.rst_n(rst_n),.dout(tx_data),.tx_valid(tx_va
lid));
SPI_SLAVE
SPI1(.MOSI(MOSI),.MISO(MISO),.SS_n(SS_n),.clk(clk),.rst_n(rst_n),.rx_data(rx_data),.rx_valid(
rx_valid),.tx_data(tx_data),.tx_valid(tx_valid));
endmodule
```

testbench:

```
module project_2_tb();
reg MOSI,SS_n,clk,rst_n;
wire MISO;
integer i;
integer j;
project_2 SPI(MOSI,MISO,SS_n,clk,rst_n);
initial begin
    for(i=0;i<256;i=i+1) begin
        SPI.RAM1.ram[i]=i;
    end
    clk = 0;
    forever begin
        #1 clk = ~clk;
    end
end
initial begin
    for(j=0;j<2;j=j+1) begin
        $display("start rst_n case = ",$time);
        rst_n = 0;
        MOSI = 1;
        SS_n=1;
        #10;
        $display("end rst_n case = ",$time);
        rst_n = 1;
        #20;
        MOSI=0;
        #20;
        MOSI=1;
        SS_n=0;
        #2;
        MOSI=0;
        #2;
        MOSI=0;
        #4;
        repeat(8) begin
            MOSI=$random;
            #2;
        end
        SS_n=1;
        #10;
        MOSI=0;
        SS_n=0;
        #6;
        MOSI=1;
        #2;
        repeat(8) begin
            MOSI=$random;
            #2;
```

```
        end
        SS_n=1;
        #20;
        SS_n=0;
        #2;
        MOSI=1;
        #4;
        MOSI=0;
        #2;
        repeat(8) begin
            MOSI=$random;
            #2;
        end
        SS_n=1;
        #10;
        SS_n=0;
        #2;
        MOSI=1;
        #6;
        repeat(8) begin
            MOSI=$random;
            #2;
        end
        #18;
        SS_n=1;
        #20;
    end
    rst_n=0;
    #6;
    rst_n=1;
    SS_n=0;
    #2;
    SS_n=1;
    #20;
    SS_n=0;
    #2;
    MOSI=0;
    #6;
    repeat(6) begin
        MOSI=$random;
        #2;
    end
    SS_n=1;
    #4;
    SS_n=0;
    #2;
    MOSI=0;
    #4;
    MOSI=1;
    #2;
```

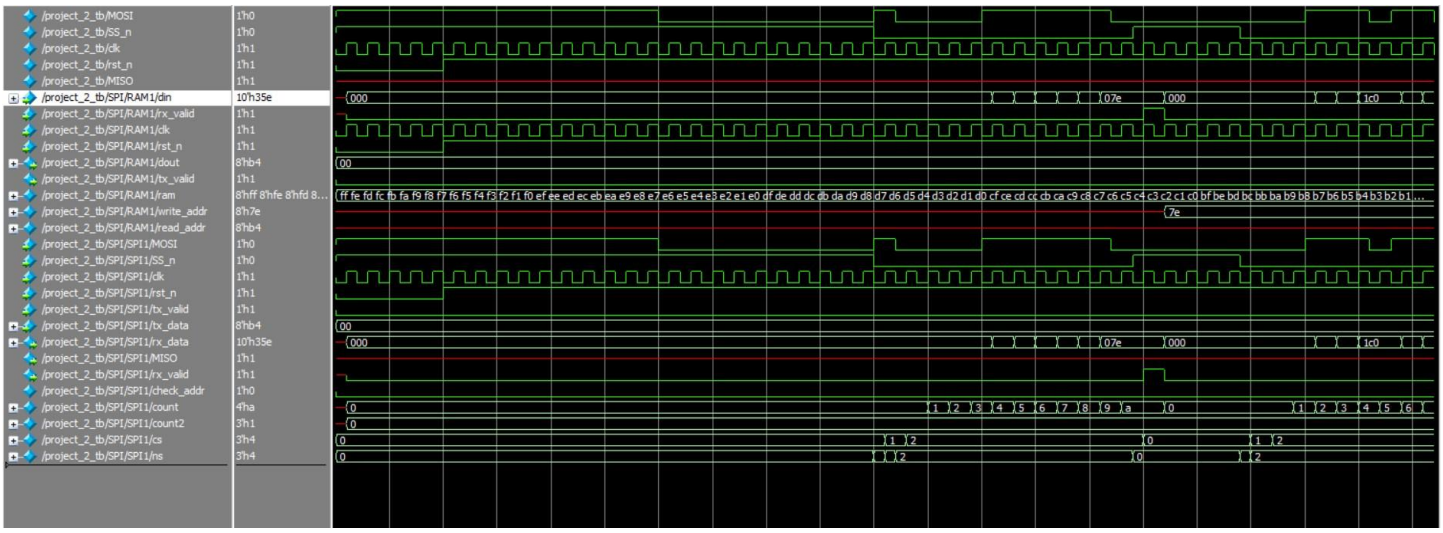
```
    repeat(6) begin
        MOSI=$random;
        #2;
    end
    SS_n=1;
    #4;
    SS_n=0;
    #2;
    MOSI=1;
    #4;
    MOSI=0;
    #2;
    repeat(6) begin
        MOSI=$random;
        #2;
    end
    SS_n=1;
    #4;
    SS_n=0;
    #2;
    MOSI=1;
    #6;
    repeat(6) begin
        MOSI=$random;
        #2;
    end
    SS_n=1;
    #4;
    for(i=0;i<1000;i=i+1) begin
        SS_n=0;
        #2;
        MOSI=1;
        #4;
        MOSI=0;
        #2;
        repeat(8) begin
            MOSI=$random;
            #2;
        end
        SS_n=1;
        #20;
        SS_n=0;
        #2;
        MOSI=1;
        #6;
        repeat(8) begin
            MOSI=$random;
            #6;
        end
        SS_n=1;
```

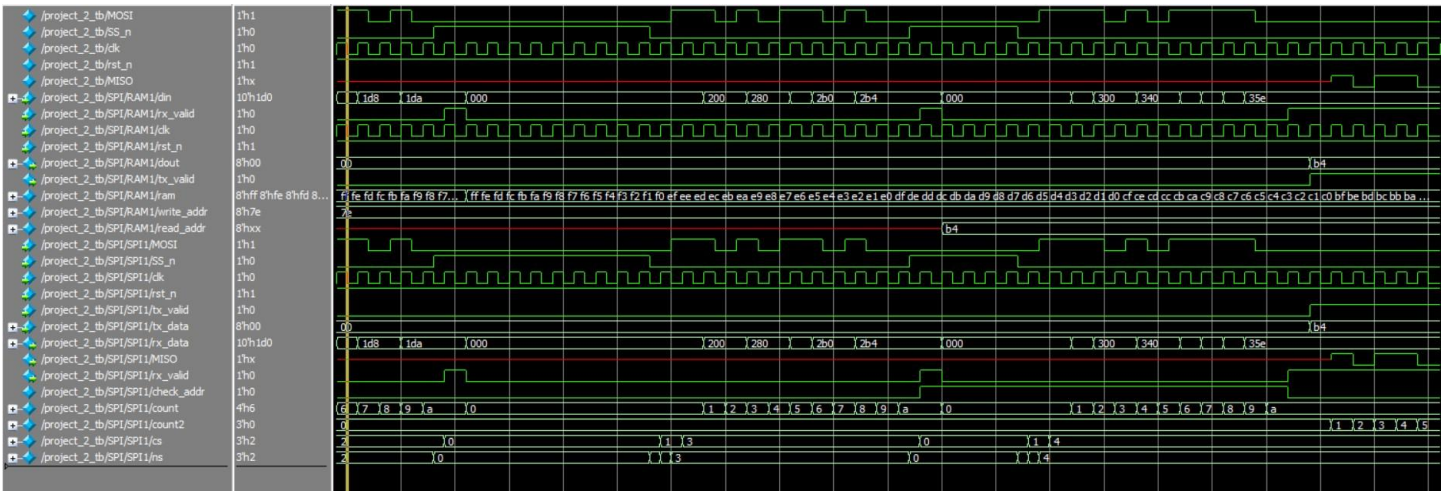
```
        #20;
        SS_n=0;
        #2;
        MOSI=0;
        #6;
        repeat(8) begin
            MOSI=$random;
            #2;
        end
        SS_n=1;
        #20;
        SS_n=0;
        #2;
        MOSI=0;
        #4;
        MOSI=1;
        #2;
        repeat(8) begin
            MOSI=$random;
            #2;
        end
        SS_n=1;
        #20;
    end

    $stop;
end
endmodule
```

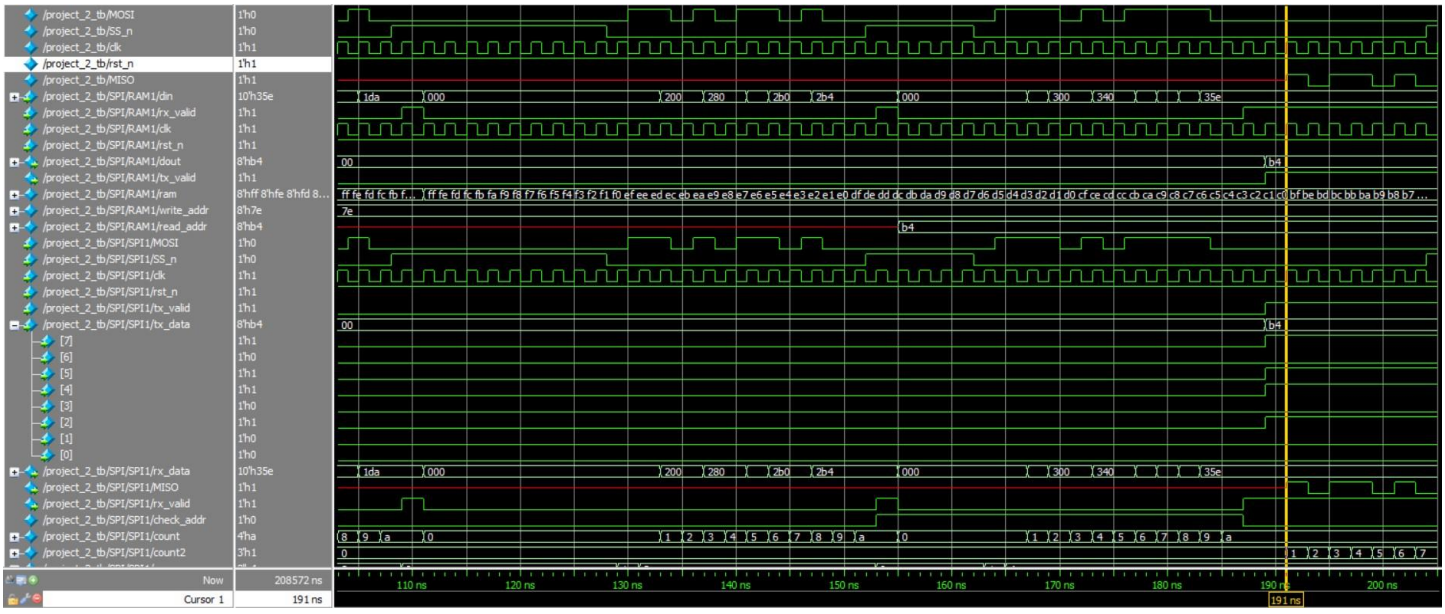
DO FILE:

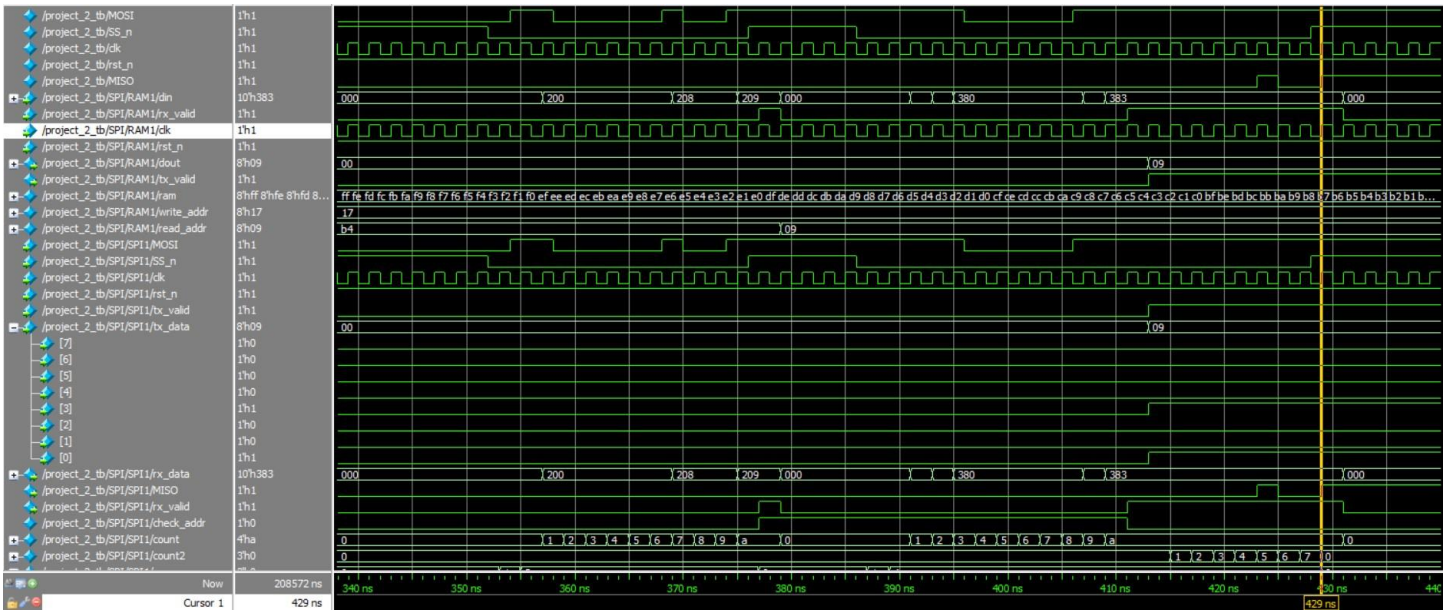
```
quit -sim
vlib work
vlog RAM.v SPI.v project2.v project_2 _tb.v
vsim -voptargs=+acc work.project_2_tb
add wave *
run -all
```

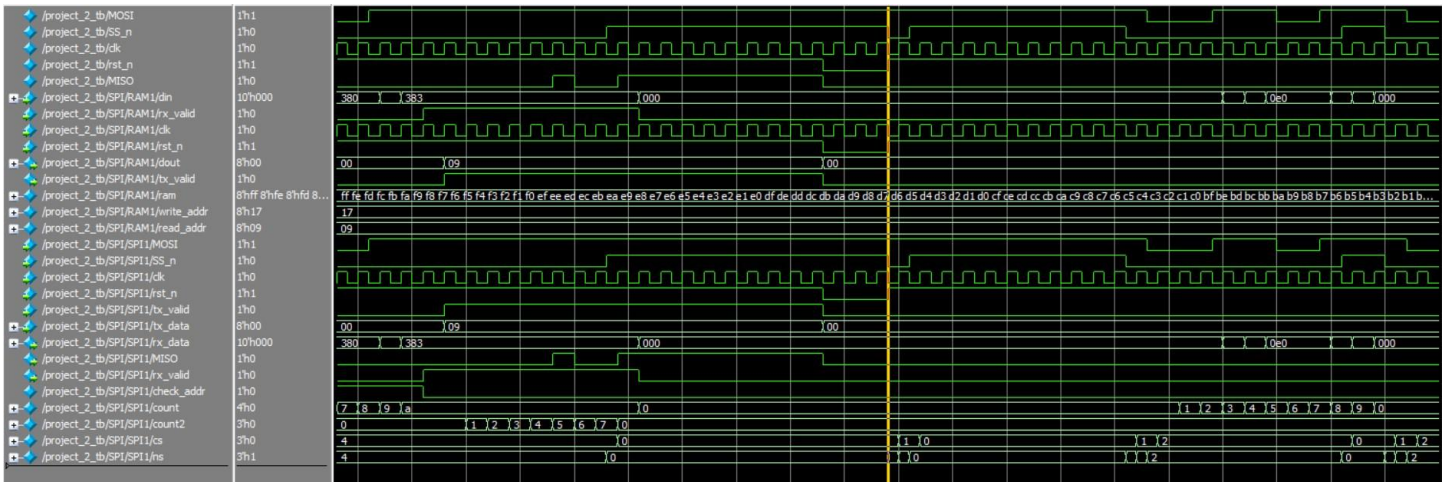
simulation:

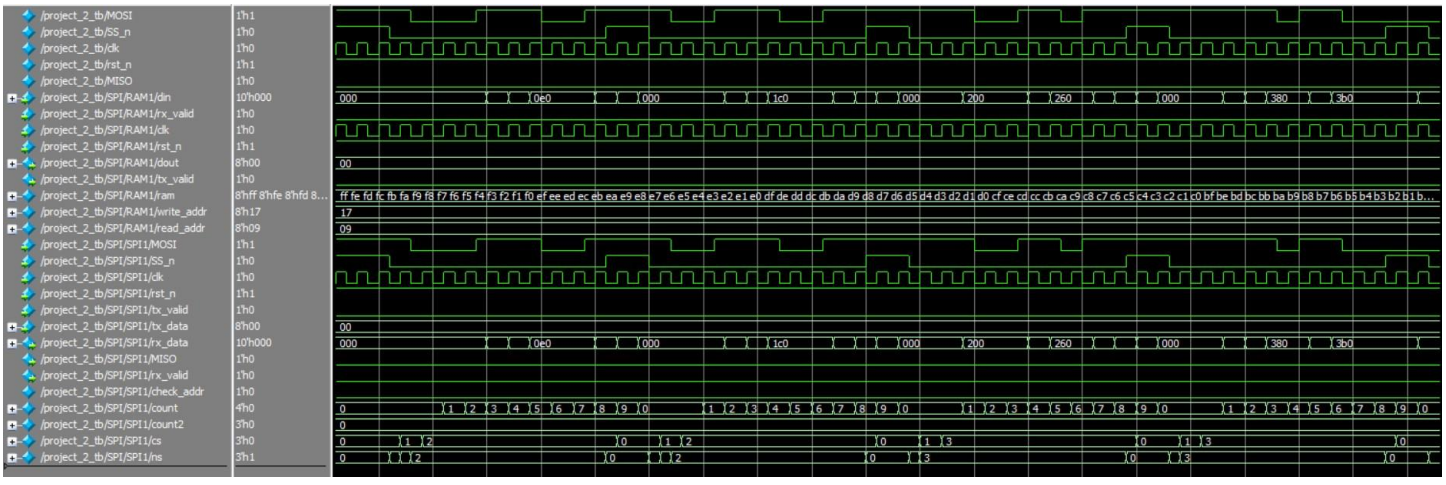

/project_2_tb/MOSI
/project_2_tb/SS_n
/project_2_tb/clk
/project_2_tb/rst_n
/project_2_tb/MISO
/project_2_tb/SPI/RAM1/din
/project_2_tb/SPI/RAM1/rx_valid
/project_2_tb/SPI/RAM1/clk
/project_2_tb/SPI/RAM1/rst_n
/project_2_tb/SPI/RAM1/dout
/project_2_tb/SPI/RAM1/tx_valid
/project_2_tb/SPI/RAM1/ram
/project_2_tb/SPI/RAM1/write_addr
/project_2_tb/SPI/RAM1/read_addr
/project_2_tb/SPI/SPI1/MOSI
/project_2_tb/SPI/SPI1/SS_n
/project_2_tb/SPI/SPI1/clk
/project_2_tb/SPI/SPI1/rst_n
/project_2_tb/SPI/SPI1/tx_valid
/project_2_tb/SPI/SPI1/tx_data
/project_2_tb/SPI/SPI1/rx_data
/project_2_tb/SPI/SPI1/MISO
/project_2_tb/SPI/SPI1/rx_valid
/project_2_tb/SPI/SPI1/check_addr
/project_2_tb/SPI/SPI1/count
/project_2_tb/SPI/SPI1/count2
/project_2_tb/SPI/SPI1/cs
/project_2_tb/SPI/SPI1/ns


/project_2_tb/MOSI
/project_2_tb/SS_n
/project_2_tb/clk
/project_2_tb/rst_n
/project_2_tb/MISO
/project_2_tb/SPI/RAM1/din
/project_2_tb/SPI/RAM1/rx_valid
/project_2_tb/SPI/RAM1/clk
/project_2_tb/SPI/RAM1/rst_n
/project_2_tb/SPI/RAM1/dout
/project_2_tb/SPI/RAM1/tx_valid
/project_2_tb/SPI/RAM1/ram
/project_2_tb/SPI/RAM1/write_addr
/project_2_tb/SPI/RAM1/read_addr
/project_2_tb/SPI/SPI1/MOSI
/project_2_tb/SPI/SPI1/SS_n
/project_2_tb/SPI/SPI1/clk
/project_2_tb/SPI/SPI1/rst_n
/project_2_tb/SPI/SPI1/tx_valid
/project_2_tb/SPI/SPI1/tx_data
/project_2_tb/SPI/SPI1/rx_data
/project_2_tb/SPI/SPI1/MISO
/project_2_tb/SPI/SPI1/rx_valid
/project_2_tb/SPI/SPI1/check_addr
/project_2_tb/SPI/SPI1/count
/project_2_tb/SPI/SPI1/count2
/project_2_tb/SPI/SPI1/cs
/project_2_tb/SPI/SPI1/ns


/project_2_tb/MOSI
/project_2_tb/SS_n
/project_2_tb/clk
/project_2_tb/rst_n
/project_2_tb/MISO
/project_2_tb/SPI/RAM1/din
/project_2_tb/SPI/RAM1/rx_valid
/project_2_tb/SPI/RAM1/clk
/project_2_tb/SPI/RAM1/rst_n
/project_2_tb/SPI/RAM1/dout
/project_2_tb/SPI/RAM1/tx_valid
/project_2_tb/SPI/RAM1/ram
/project_2_tb/SPI/RAM1/write_addr
/project_2_tb/SPI/RAM1/read_addr
/project_2_tb/SPI/SPI1/MOSI
/project_2_tb/SPI/SPI1/SS_n
/project_2_tb/SPI/SPI1/clk
/project_2_tb/SPI/SPI1/rst_n
/project_2_tb/SPI/SPI1/tx_valid
/project_2_tb/SPI/SPI1/tx_data
[7]
[6]
[5]
[4]
[3]
[2]
[1]
[0]
/project_2_tb/SPI/SPI1/rx_data
/project_2_tb/SPI/SPI1/MISO
/project_2_tb/SPI/SPI1/rx_valid
/project_2_tb/SPI/SPI1/check_addr
/project_2_tb/SPI/SPI1/count
/project_2_tb/SPI/SPI1/count2
Now
Cursor 1
110 ns
120 ns
130 ns
140 ns
150 ns
160 ns
170 ns
180 ns
190 ns
200 ns

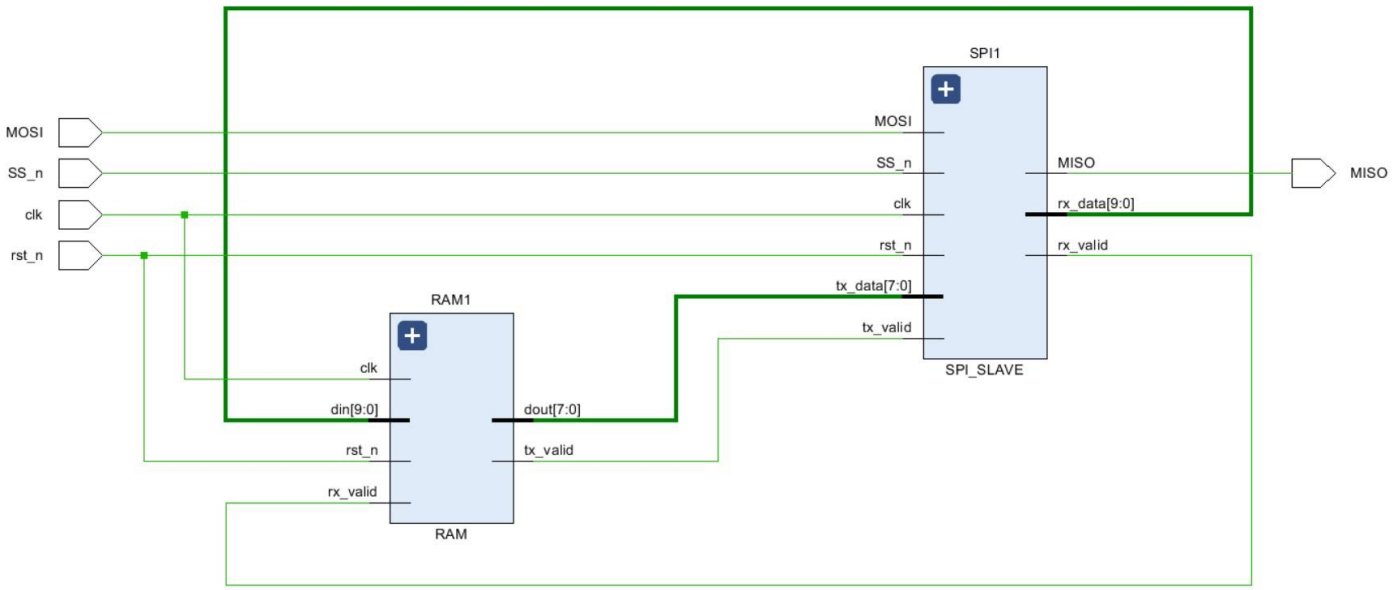

/project_2_tb/MOSI
/project_2_tb/SS_n
/project_2_tb/clk
/project_2_tb/rst_n
/project_2_tb/MISO
/project_2_tb/SPI/RAM1/din
/project_2_tb/SPI/RAM1/rx_valid
/project_2_tb/SPI/RAM1/clk
/project_2_tb/SPI/RAM1/rst_n
/project_2_tb/SPI/RAM1/dout
/project_2_tb/SPI/RAM1/tx_valid
/project_2_tb/SPI/RAM1/ram
/project_2_tb/SPI/RAM1/write_addr
/project_2_tb/SPI/RAM1/read_addr
/project_2_tb/SPI/SPI1/MOSI
/project_2_tb/SPI/SPI1/SS_n
/project_2_tb/SPI/SPI1/clk
/project_2_tb/SPI/SPI1/rst_n
/project_2_tb/SPI/SPI1/tx_valid
/project_2_tb/SPI/SPI1/tx_data
/project_2_tb/SPI/SPI1/rx_data
/project_2_tb/SPI/SPI1/MISO
/project_2_tb/SPI/SPI1/rx_valid
/project_2_tb/SPI/SPI1/check_addr
/project_2_tb/SPI/SPI1/count
/project_2_tb/SPI/SPI1/count2
Now
Cursor 1
208572 ns
429 ns
340 ns
350 ns
360 ns
370 ns
380 ns
390 ns
400 ns
410 ns
420 ns
429 ns

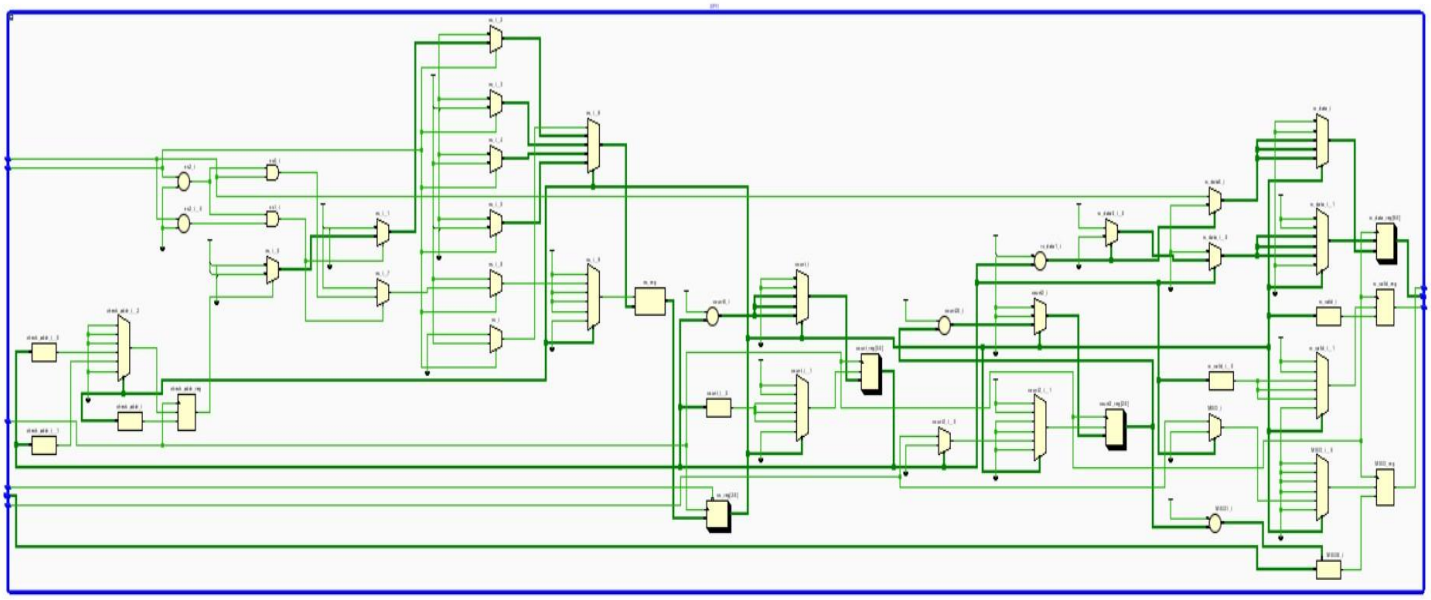

/project_2_tb/MOSI
/project_2_tb/SS_n
/project_2_tb/clk
/project_2_tb/rst_n
/project_2_tb/MISO
/project_2_tb/SPI/RAM1/din
/project_2_tb/SPI/RAM1/rx_valid
/project_2_tb/SPI/RAM1/clk
/project_2_tb/SPI/RAM1/rst_n
/project_2_tb/SPI/RAM1/dout
/project_2_tb/SPI/RAM1/tx_valid
/project_2_tb/SPI/RAM1/ram
/project_2_tb/SPI/RAM1/write_addr
/project_2_tb/SPI/RAM1/read_addr
/project_2_tb/SPI/SPI1/MOSI
/project_2_tb/SPI/SPI1/SS_n
/project_2_tb/SPI/SPI1/clk
/project_2_tb/SPI/SPI1/rst_n
/project_2_tb/SPI/SPI1/tx_valid
/project_2_tb/SPI/SPI1/tx_data
/project_2_tb/SPI/SPI1/rx_data
/project_2_tb/SPI/SPI1/MISO
/project_2_tb/SPI/SPI1/rx_valid
/project_2_tb/SPI/SPI1/check_addr
/project_2_tb/SPI/SPI1/count
/project_2_tb/SPI/SPI1/count2
/project_2_tb/SPI/SPI1/cs
/project_2_tb/SPI/SPI1/ns


/project_2_tb/MOSI
/project_2_tb/SS_n
/project_2_tb/clk
/project_2_tb/rst_n
/project_2_tb/MISO
/project_2_tb/SPI/RAM1/din
/project_2_tb/SPI/RAM1/rx_valid
/project_2_tb/SPI/RAM1/clk
/project_2_tb/SPI/RAM1/rst_n
/project_2_tb/SPI/RAM1/dout
/project_2_tb/SPI/RAM1/tx_valid
/project_2_tb/SPI/RAM1/ram
/project_2_tb/SPI/RAM1/write_addr
/project_2_tb/SPI/RAM1/read_addr
/project_2_tb/SPI/SPI1/MOSI
/project_2_tb/SPI/SPI1/SS_n
/project_2_tb/SPI/SPI1/clk
/project_2_tb/SPI/SPI1/rst_n
/project_2_tb/SPI/SPI1/tx_valid
/project_2_tb/SPI/SPI1/tx_data
/project_2_tb/SPI/SPI1/rx_data
/project_2_tb/SPI/SPI1/MISO
/project_2_tb/SPI/SPI1/rx_valid
/project_2_tb/SPI/SPI1/check_addr
/project_2_tb/SPI/SPI1/count
/project_2_tb/SPI/SPI1/count2
/project_2_tb/SPI/SPI1/cs
/project_2_tb/SPI/SPI1/ns

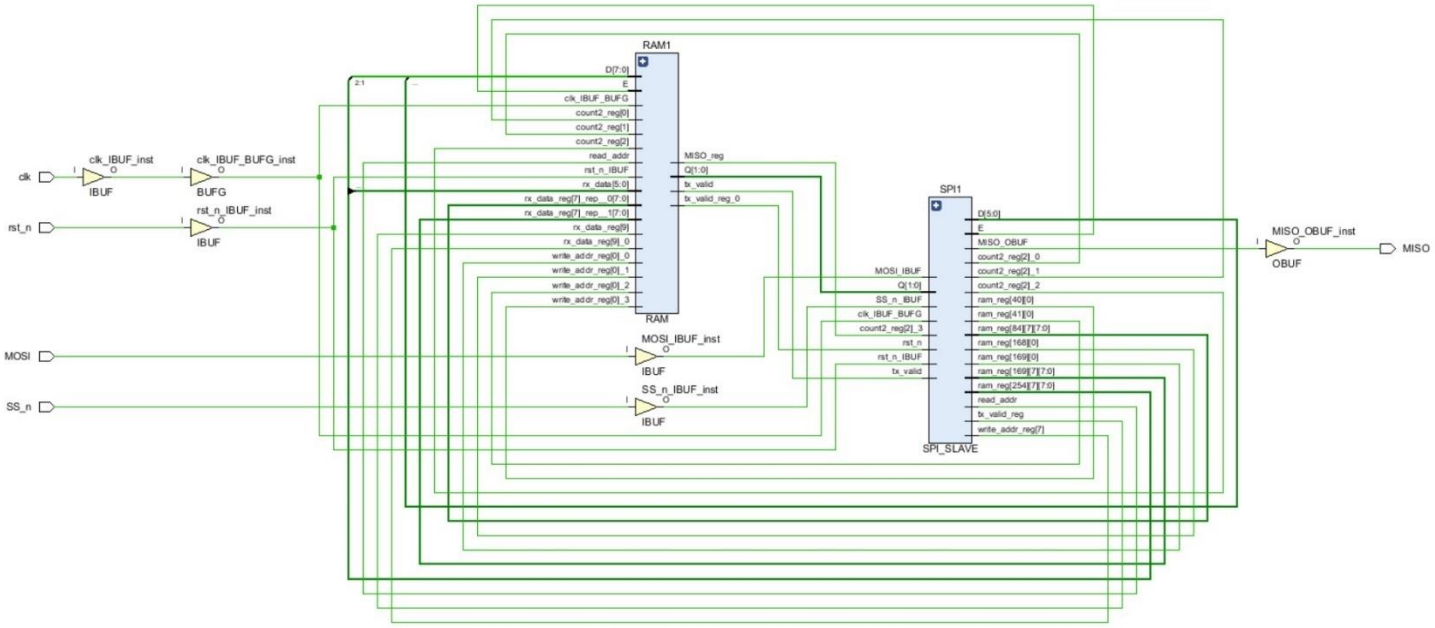
Gray_encoding:

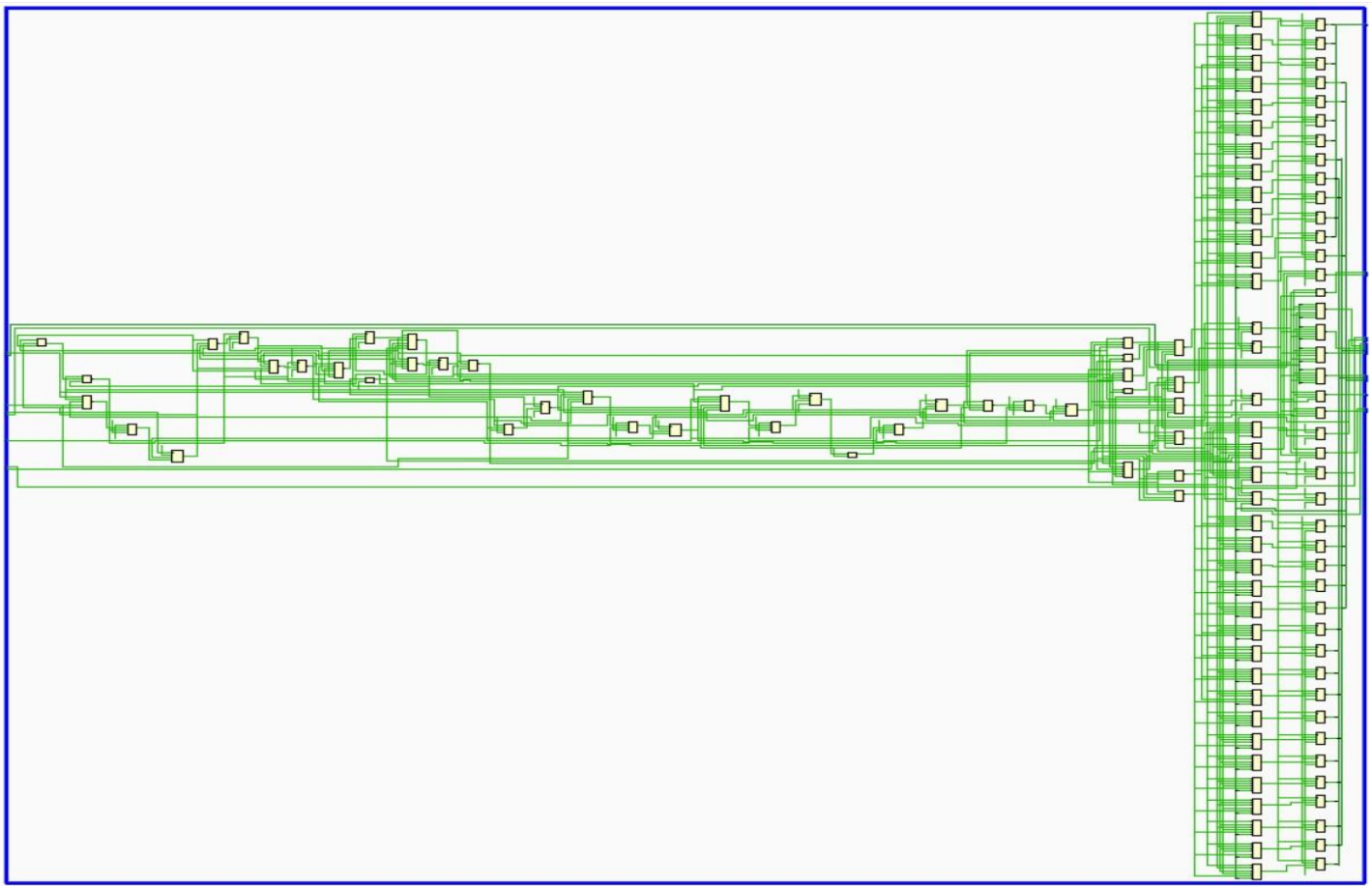
schematic after the elaboration:





schematic after the synthesis:





Synthesis report:

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

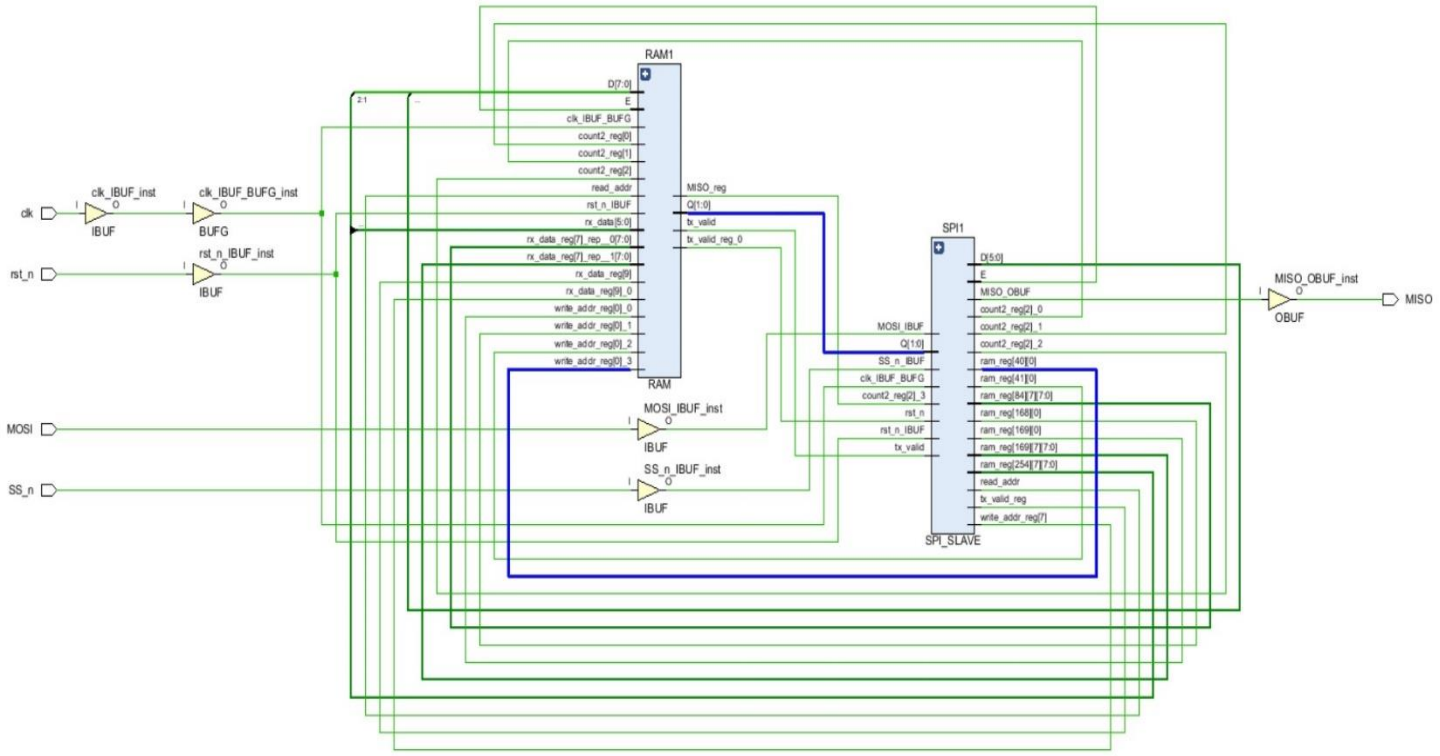
Timing report after the synthesis:

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	6.143 ns	Worst Hold Slack (WHS):	0.147 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:	4246	Total Number of Endpoints:	4246	Total Number of Endpoints:	2146

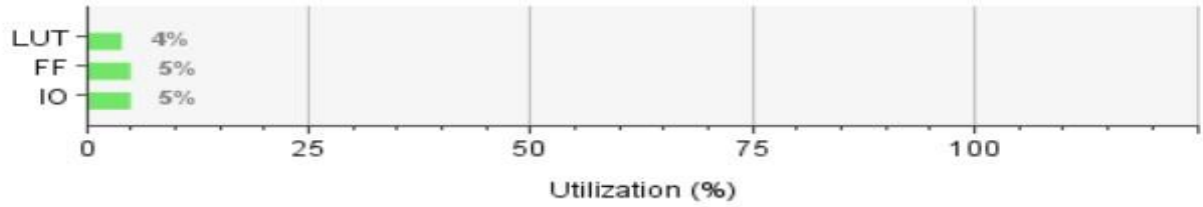
All user specified timing constraints are met.

Critical path:



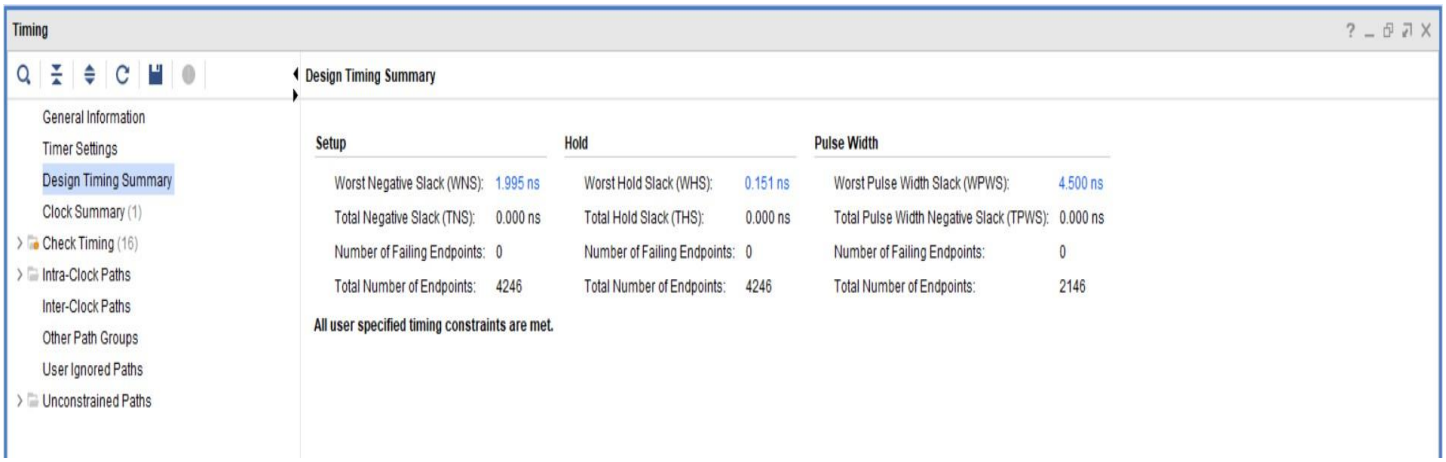
Utilization report (implementation):

Resource	Utilization	Available	Utilization %
LUT	873	20800	4.20
FF	2148	41600	5.16
IO	5	106	4.72

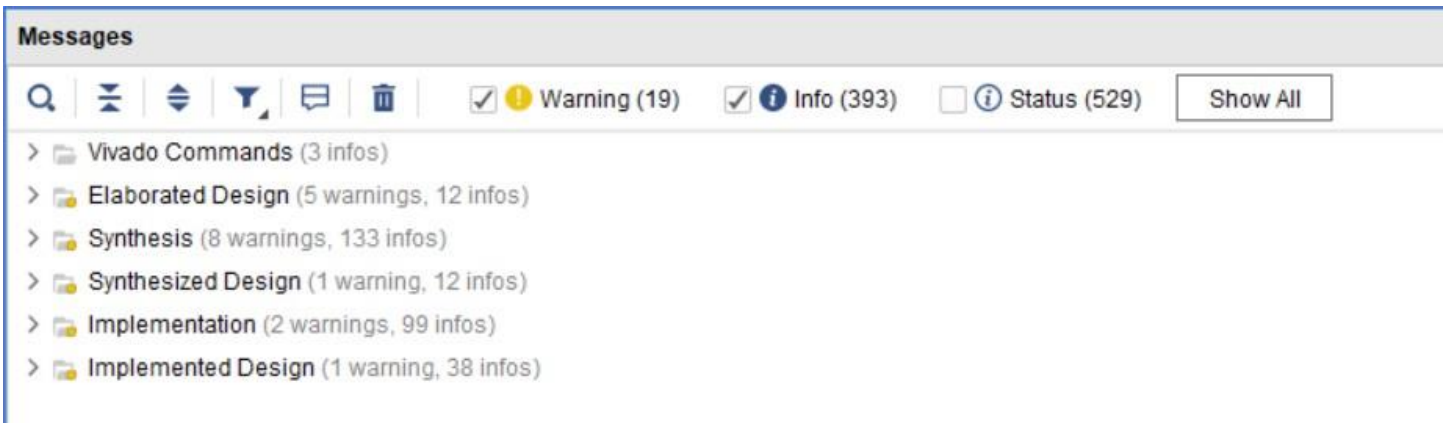


Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	F8 Muxes (8150)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
project_2	873	2148	273	136	827	873	52	5	1
RAM1 (RAM)	811	2100	273	136	811	811	8	0	0
SPI1 (SPI_SLAVE)	62	48	0	0	20	62	42	0	0

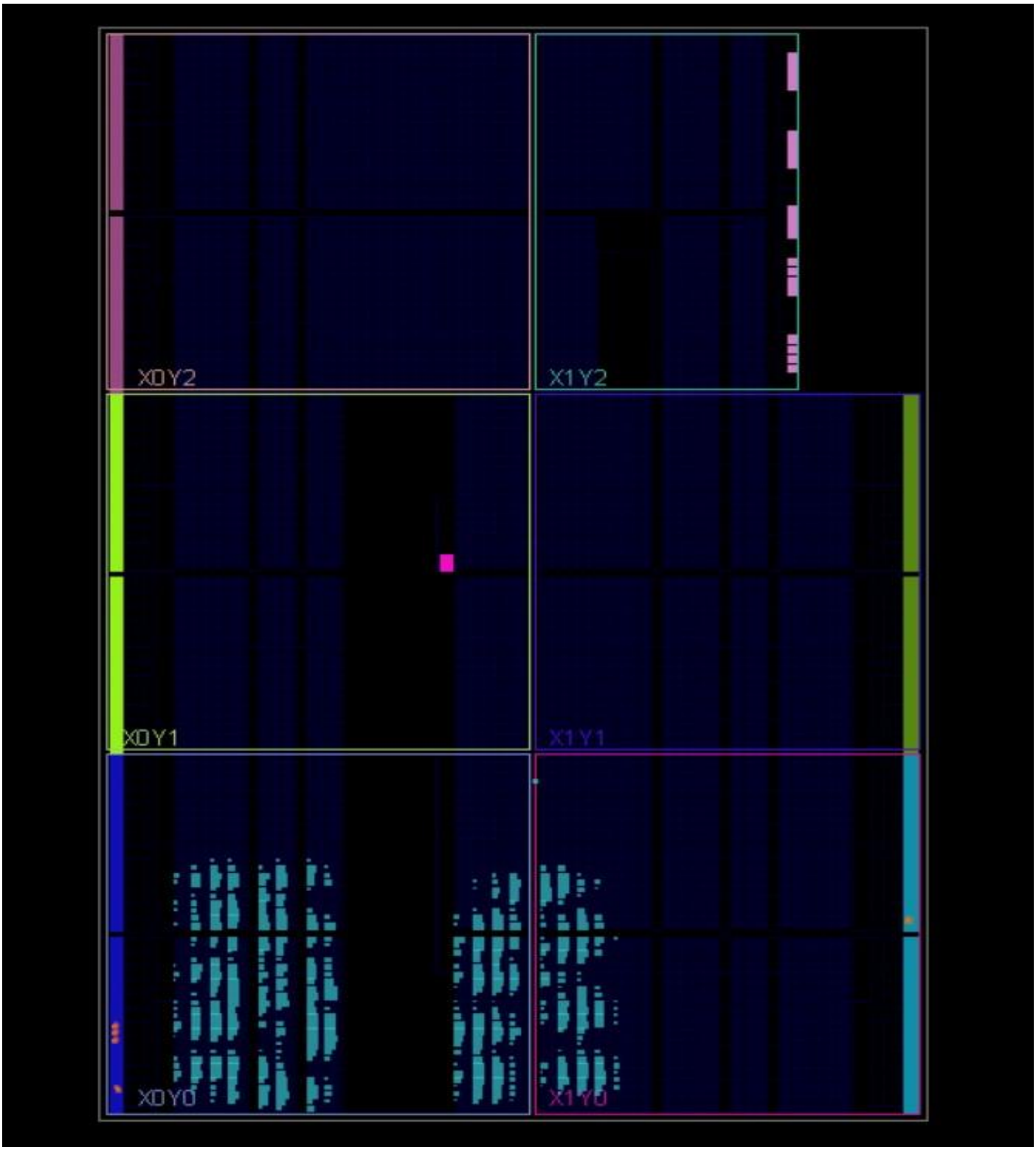
Timing report(implementation):



Messages:

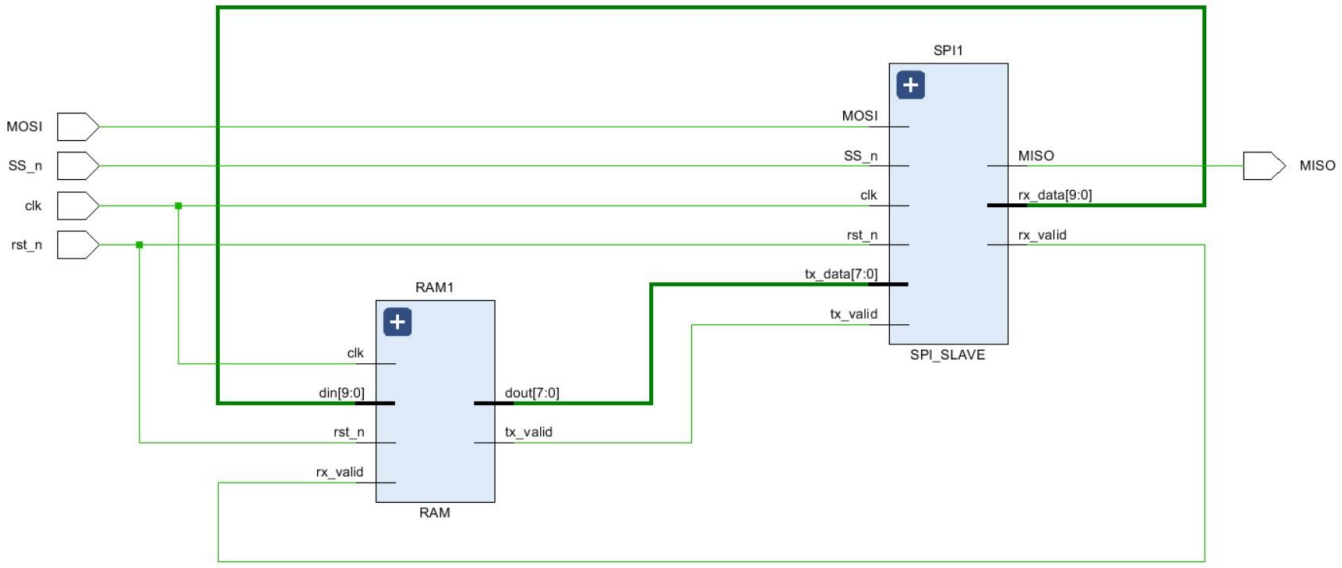


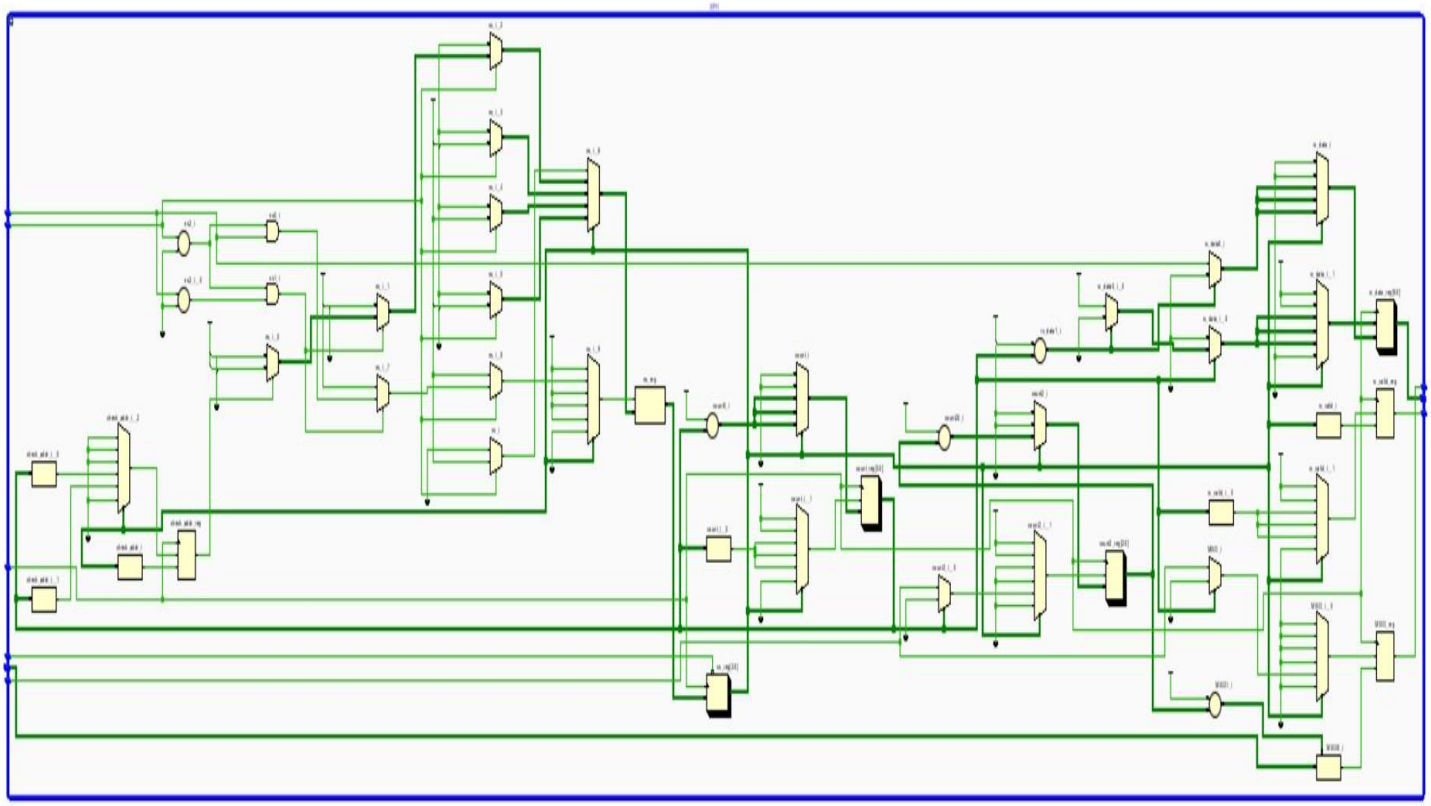
Device:



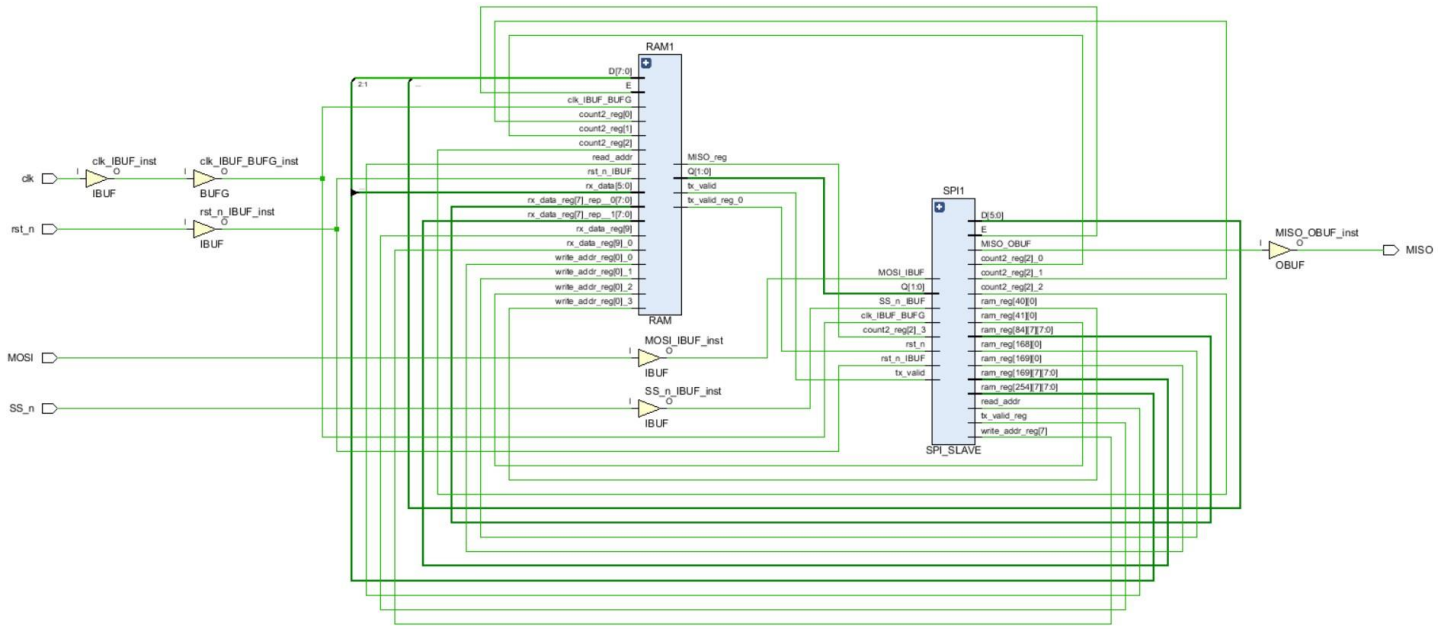
one_hot:

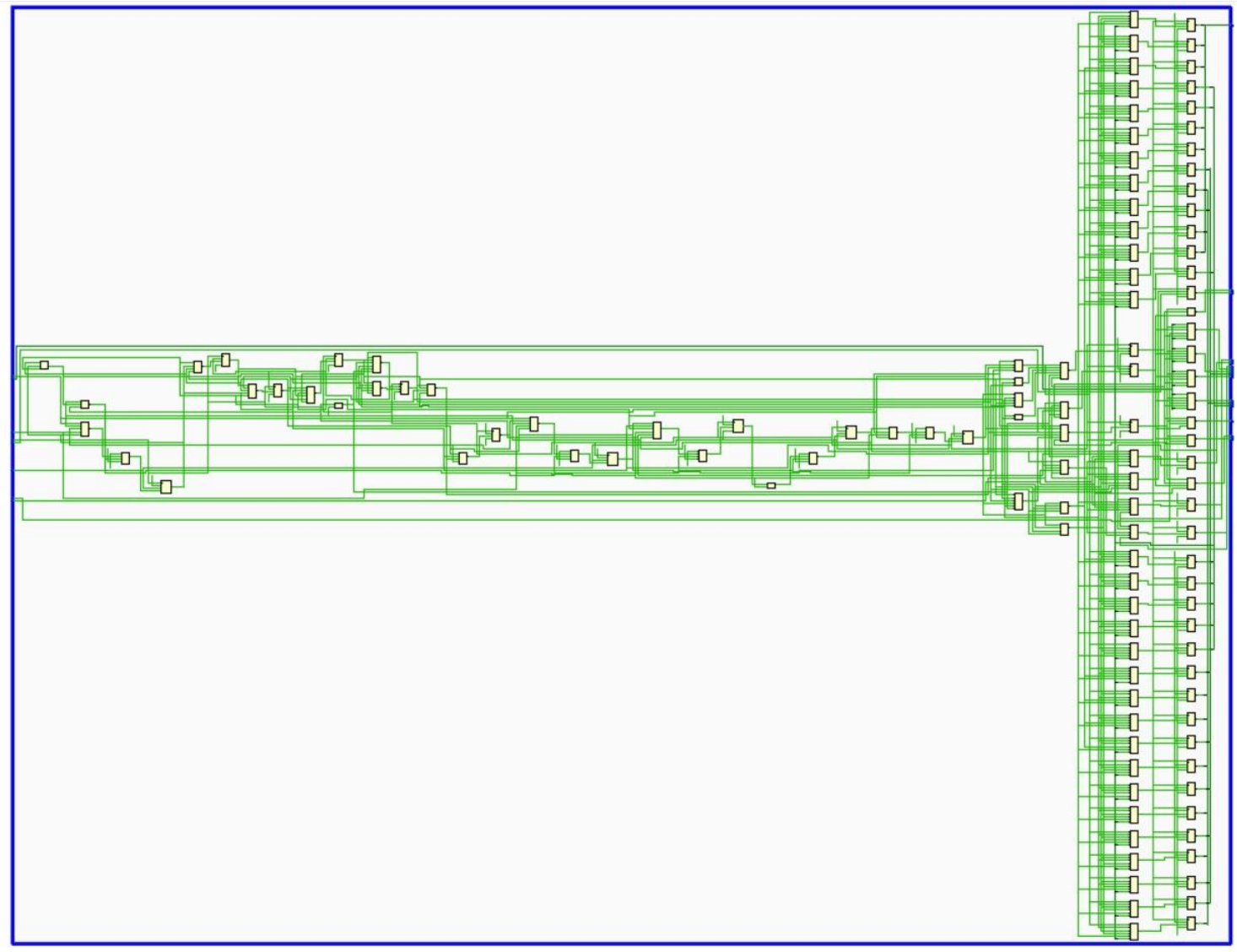
schematic after the elaboration:





schematic after the synthesis:





Synthesis report:

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

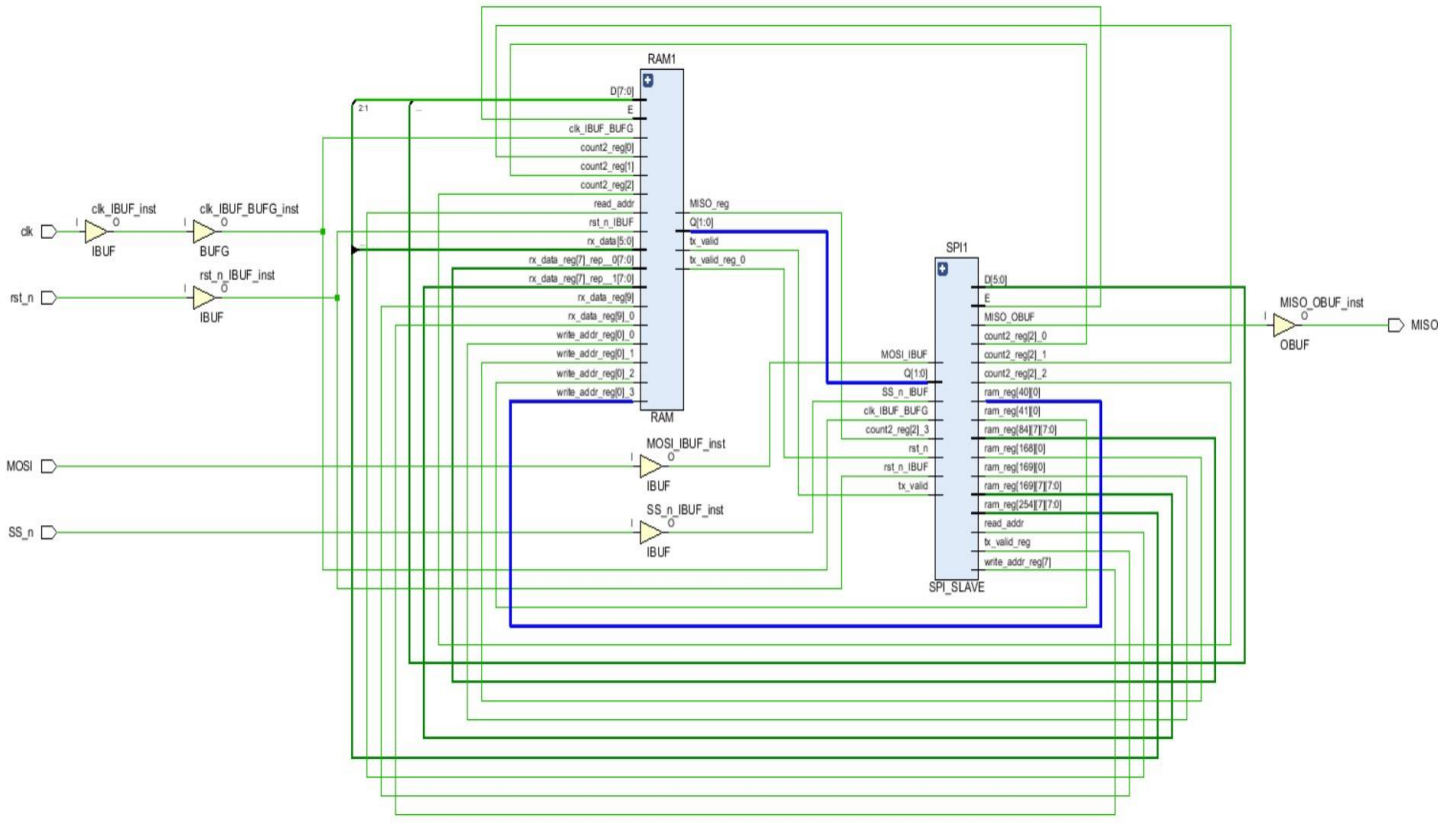
Timing report after the synthesis:

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	6.143 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:	4246	Total Number of Endpoints:	4246	Total Number of Endpoints:	2148

All user specified timing constraints are met.

Critical path:

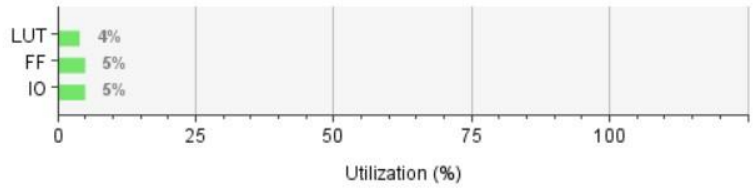


Utilization report (implementation):

Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	F8 Muxes (8150)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
project_2	874	2152	273	136	714	874	55	5	1
RAM1 (RAM)	811	2100	273	136	695	811	8	0	0
SPI1 (SPI_SLAVE)	63	52	0	0	23	63	46	0	0

Summary

Resource	Utilization	Available	Utilization %
LUT	874	20800	4.20
FF	2152	41600	5.17
IO	5	106	4.72



Timing report(implementation):

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	2.633 ns	Worst Hold Slack (WHS):	0.192 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:	4246	Total Number of Endpoints:	4246	Total Number of Endpoints:	2148

All user specified timing constraints are met.

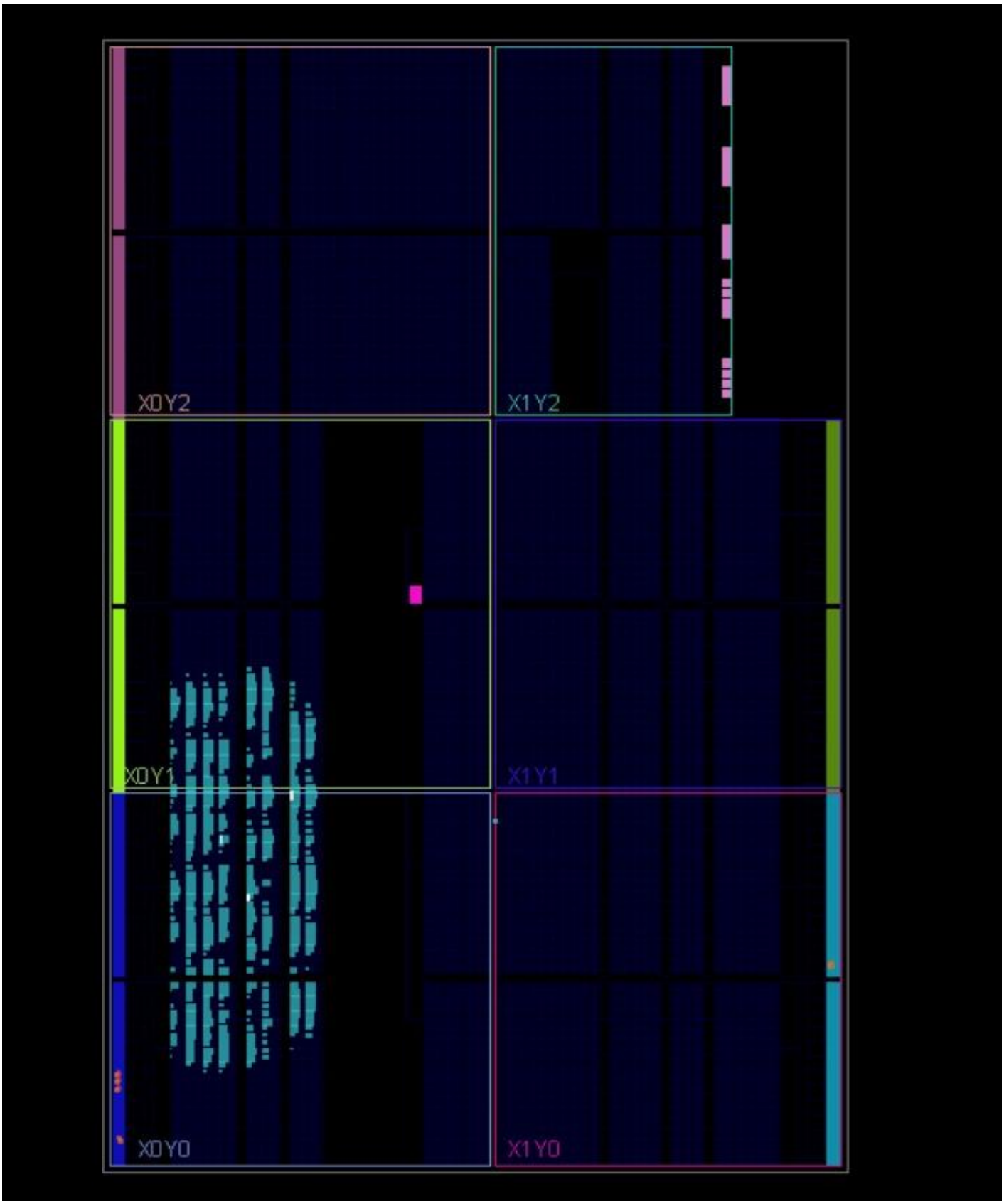
Messages:

Messages

Warning (19) Info (370) Status (526) Show All

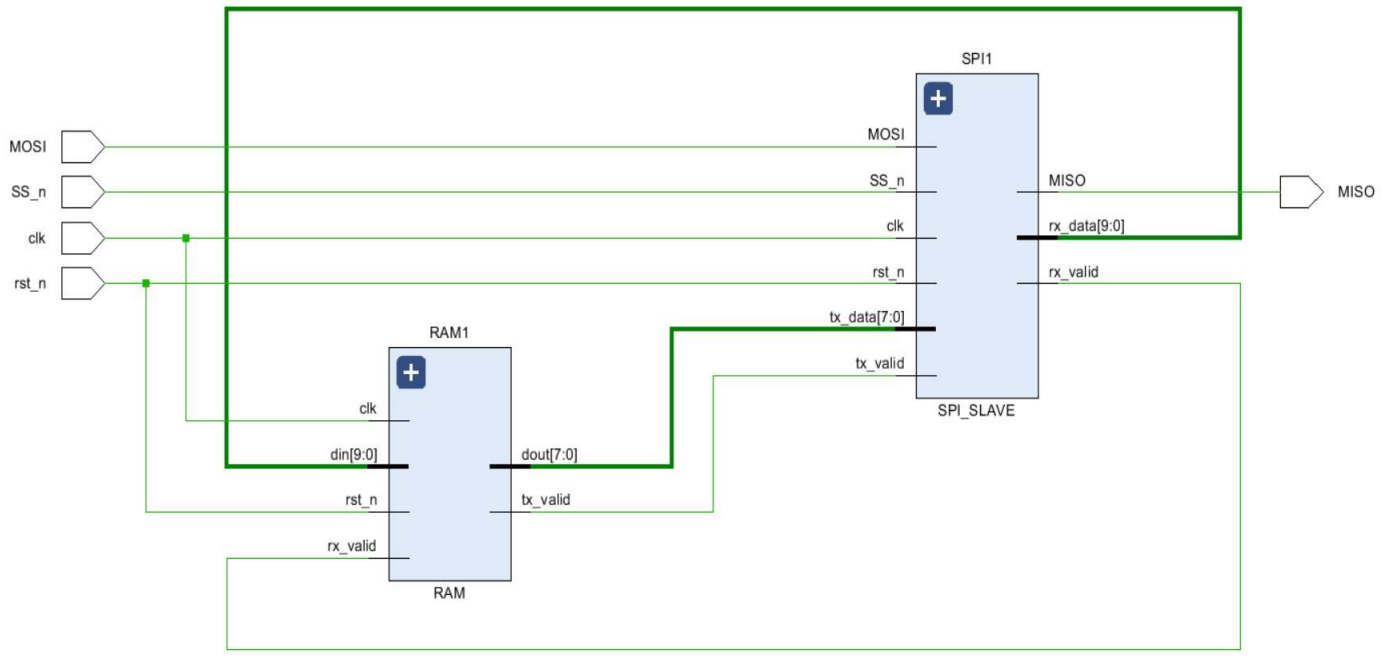
- Vivado Commands (3 infos)
- Elaborated Design (5 warnings, 12 infos)
- Synthesis (8 warnings, 133 infos)
- Synthesized Design (1 warning, 10 infos)
- Implementation (2 warnings, 99 infos)
- Implemented Design (1 warning, 16 infos)

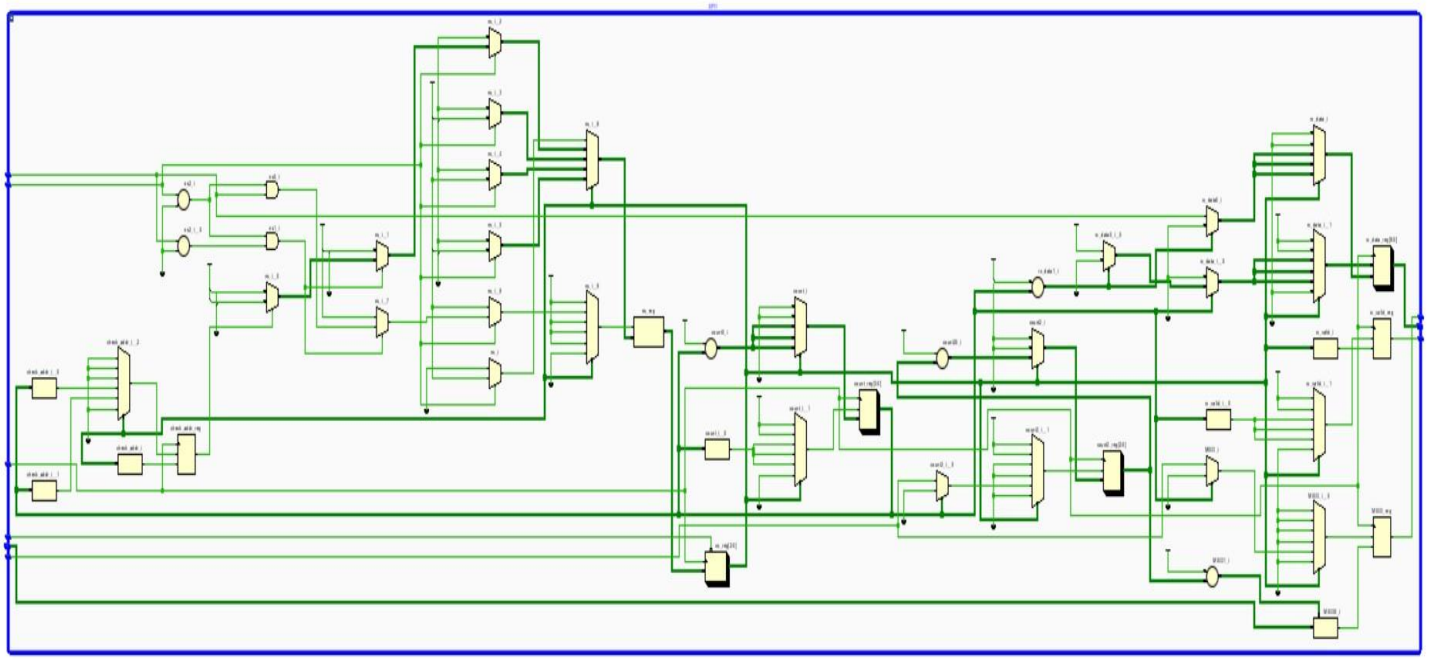
Device:



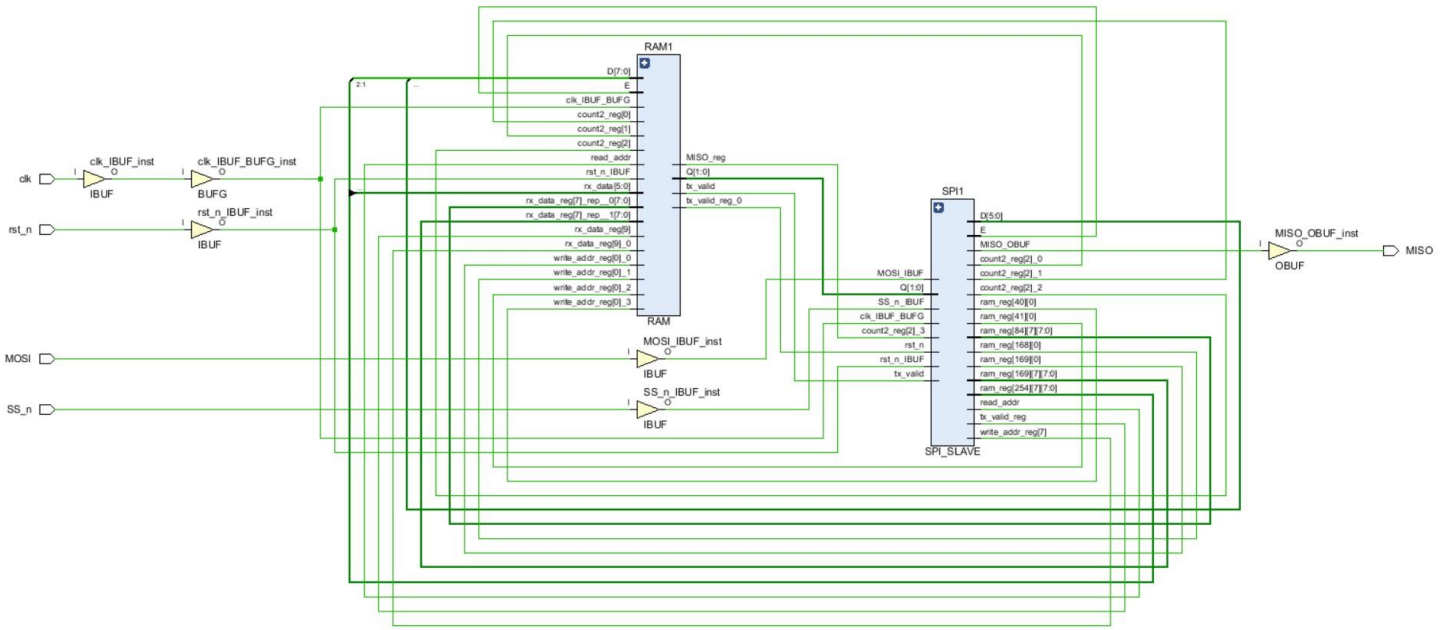
sequential:

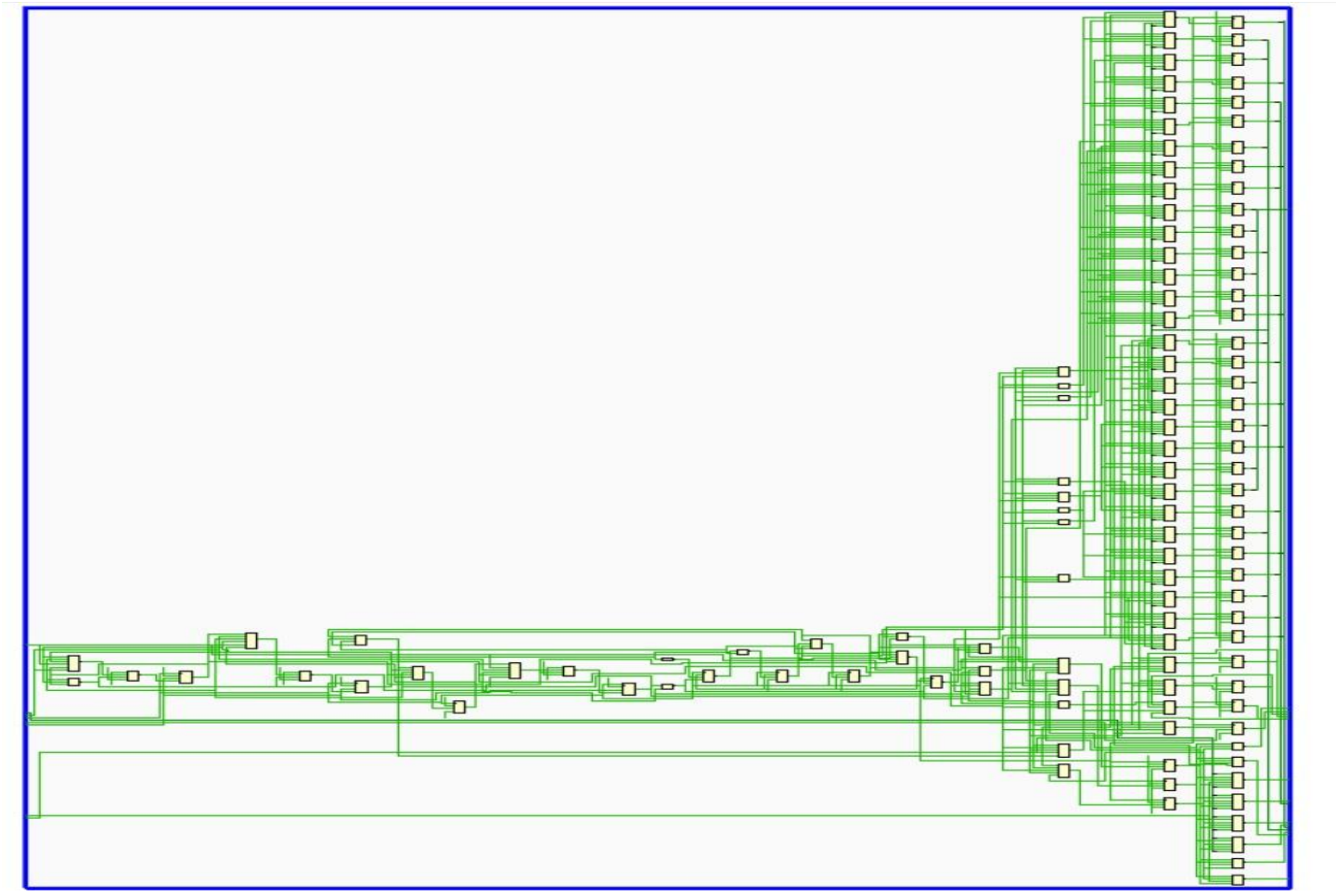
schematic after the elaboration:





schematic after the synthesis:





Synthesis report:

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

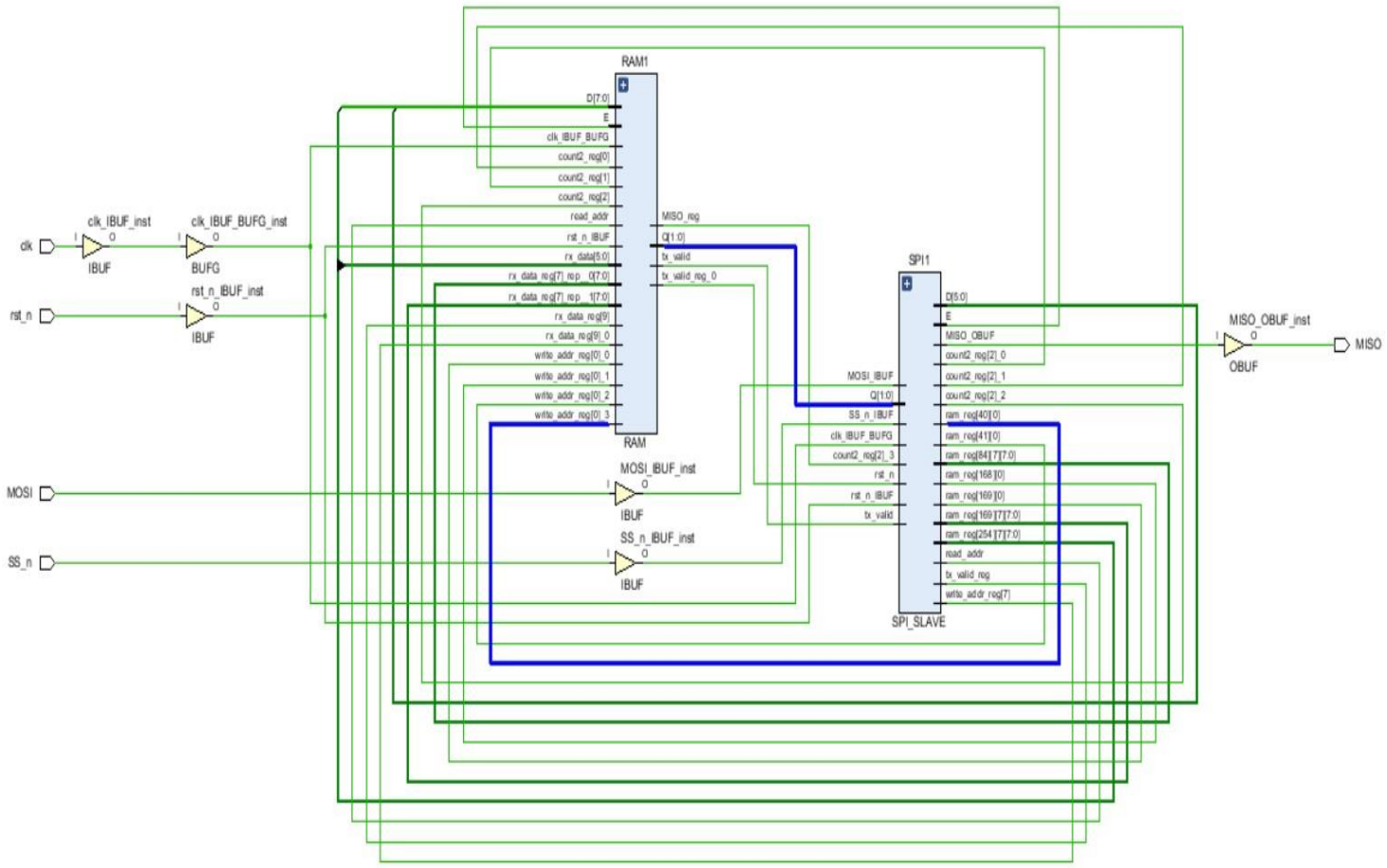
Timing report after the synthesis:

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	6.143 ns	Worst Hold Slack (WHS):	0.147 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:	4250	Total Number of Endpoints:	4250	Total Number of Endpoints:	2146

All user specified timing constraints are met.

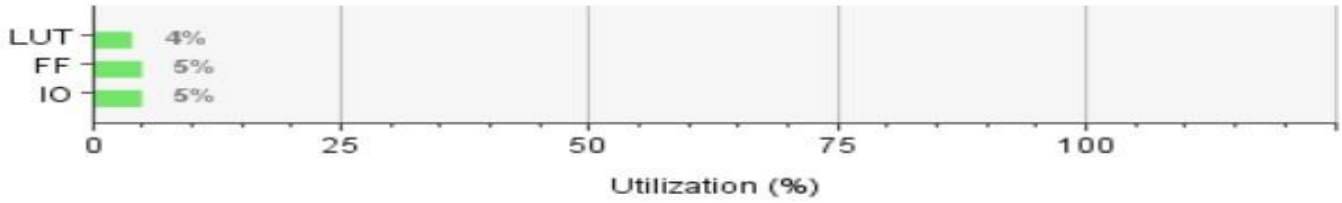
Critical path:



Utilization report (implementation):

Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	F8 Muxes (8150)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Bonded IOB (106)	BUFGCTRL (32)
project_2	872	2148	273	136	791	872	53	5	1
RAM1 (RAM)	811	2100	273	136	773	811	8	0	0
SPI1 (SPI_SLAVE)	61	48	0	0	21	61	43	0	0

Resource	Utilization	Available	Utilization %
LUT	872	20800	4.19
FF	2148	41600	5.16
IO	5	106	4.72



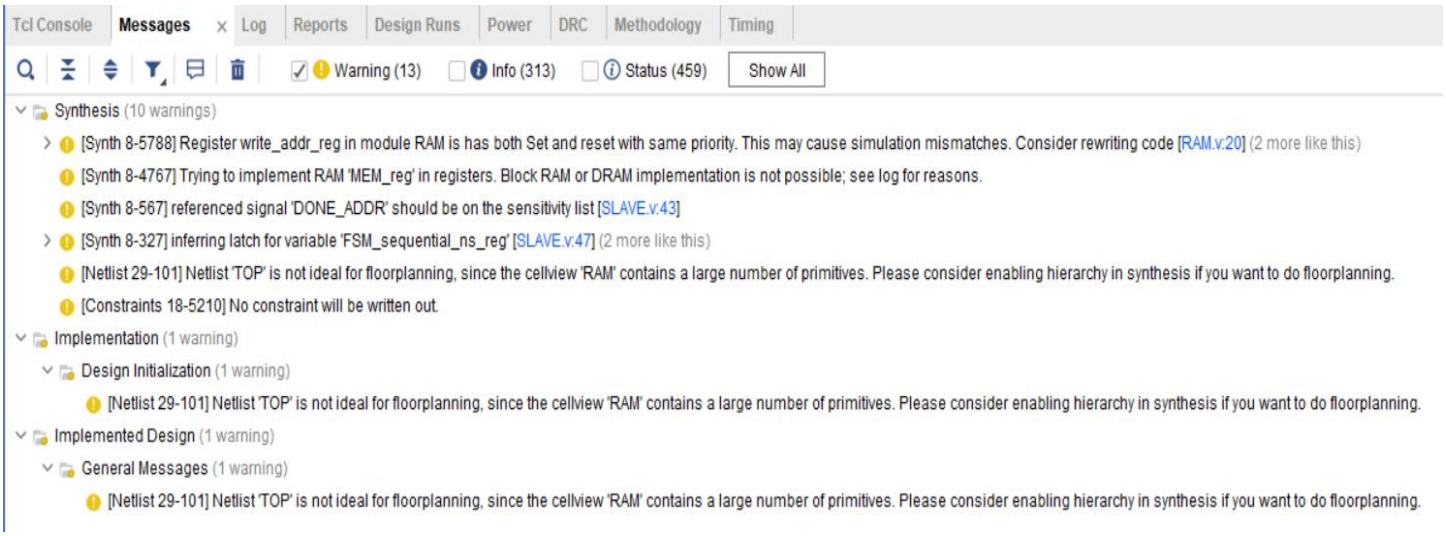
Timing report(implementation):

Design Timing Summary

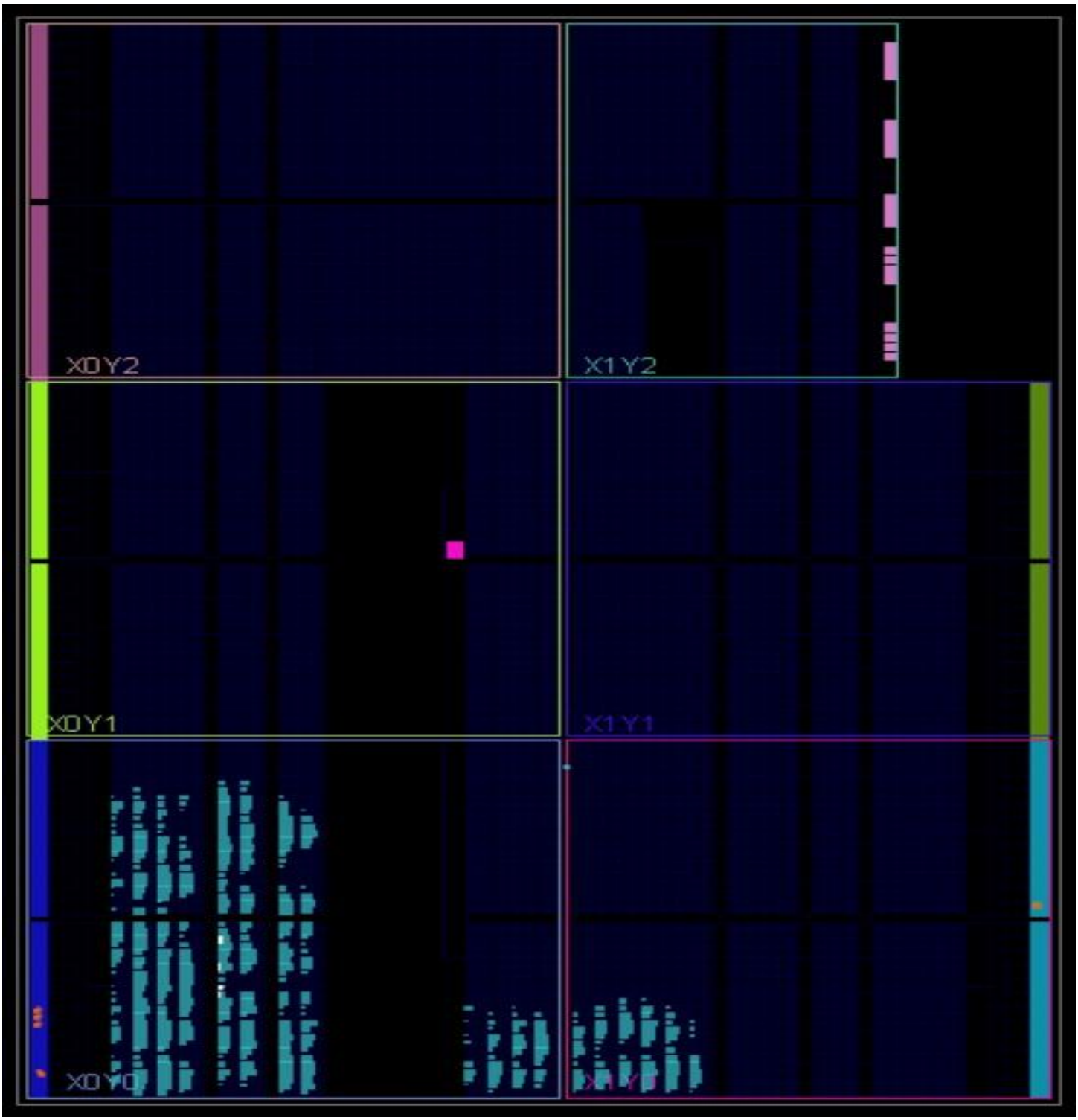
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	2.069 ns	Worst Hold Slack (WHS):	0.197 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	4250	Total Number of Endpoints:	4250	Total Number of Endpoints:	2146

All user specified timing constraints are met.

Messages:

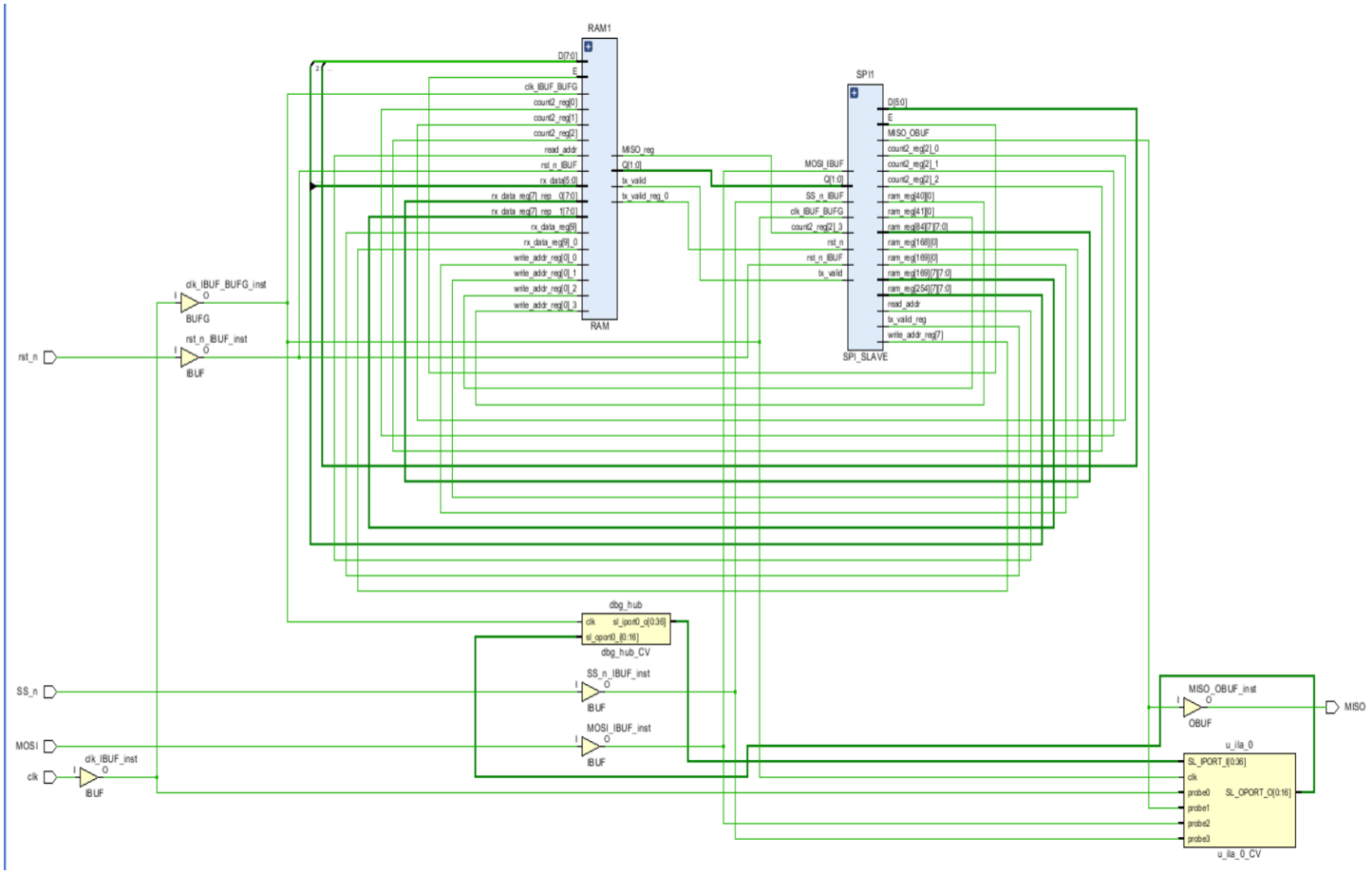


Device:



❖ Best Setup Time Slack (Tss) for -> One_hot encoding.

Setup debug



❖ **In case of using synchronous rst_n we notice that Vivado Read the RAM as a RAM Block in the FBGA board.**