SPI SLAVE WITH SINGLE PORT RAM

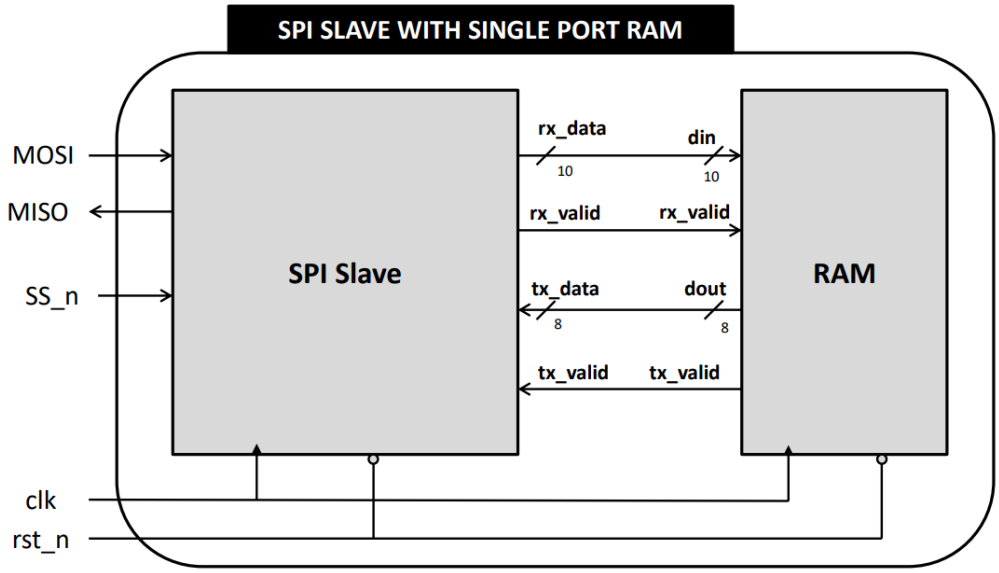
Prepared By :

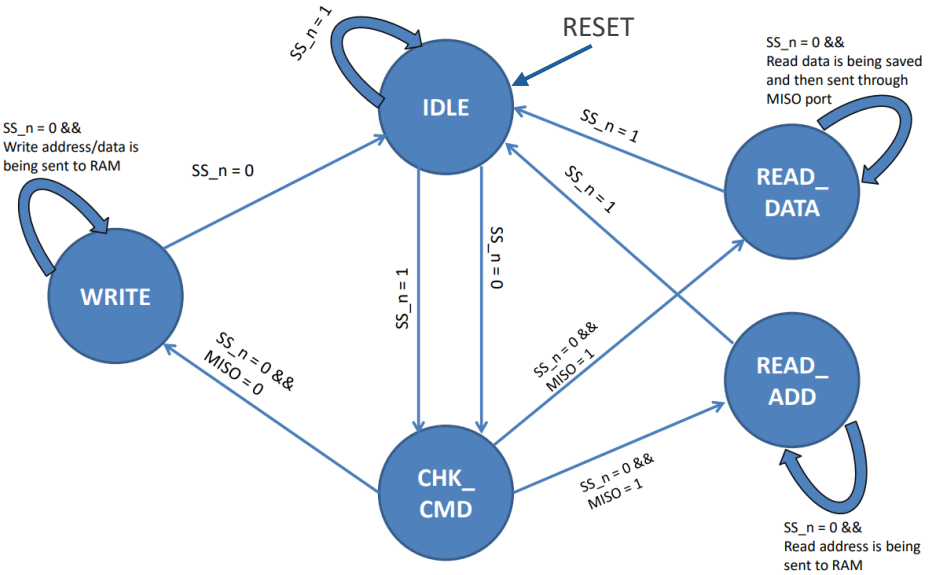
Mohamed Karam

Contents

OVERVIEW

This project creates a SPI Slave interface that acts as a communication bridge between a master device and RAM memory. The SPI Slave receives serial commands from the master through the MOSI and slave select to know when communication starts. The MISO is used to send data back to the master during read operations. The system converts the serial MOSI data to parallel format and translates commands for the RAM using 2-bit command codes to distinguish between write operations (sending address then data to store) and read operations (sending address then receiving stored data back). A state machine manages the communication flow, enabling external devices to access memory through the SPI protocol.





RTL CODE

SPI_top_module.v

```
module SPI_top_module(clk, rst_n, SS_n, MOSI, MISO);

    parameter MEM_DEPTH = 256;
    parameter ADDR_SIZE = 8;
    input clk, rst_n, SS_n, MOSI;
    output MISO;
    wire [9:0] rx_data;
    wire [7:0] tx_data;
    wire rx_valid, tx_valid;

    // SPI slave module
    SPI_slave SPI_slave_inst (
        .clk(clk),
        .rst_n(rst_n),
        .SS_n(SS_n),
        .MOSI(MOSI),
        .tx_data(tx_data),
        .tx_valid(tx_valid),
        .MISO(MISO),
        .rx_data(rx_data),
        .rx_valid(rx_valid)
    );

    // Single Port RAM module
    RAM #(
        .MEM_DEPTH(MEM_DEPTH),
        .ADDR_SIZE(ADDR_SIZE)
    ) RAM_inst (
        .clk(clk),
        .rst_n(rst_n),
        .din(rx_data),
        .rx_valid(rx_valid),
        .dout(tx_data),
        .tx_valid(tx_valid)
    );

endmodule
```

SPI_slave.v

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

    // States
    parameter IDLE     = 3'b000;
    parameter CHK_CMD  = 3'b001;
    parameter WRITE    = 3'b011;
    parameter READ_ADD = 3'b010;
    parameter READ_DATA = 3'b110;

    input clk;
    input rst_n;
    input SS_n;
    input MOSI;
    input tx_valid;
    input [7:0] tx_data;

    output reg MISO;
    output reg rx_valid;
    output reg [9:0] rx_data;

    reg [9:0] shift_reg; // For serial-to-parallel conversion
    reg [7:0] tx_data_reg; // Holds data for transmission
    reg read_addr_received; // To track if read address received
    reg [4:0] counter; // Counts bits for receive/transmission

    (* fsm_encoding = "gray" *)
    reg [2:0] cs, ns; // Current and next state

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

```
// Next state logic
always @(*) begin
    case (cs)
        IDLE : begin
            if (SS_n)
                ns = IDLE;
            else
                ns = CHK_CMD;
        end
        CHK_CMD: begin
            if (SS_n)
                ns = IDLE;
            else begin
                if (MOSI) begin
                    if (read_addr_received)
                        ns = READ_DATA;
                    else
                        ns = READ_ADD;
                end
                else
                    ns = WRITE;
            end
        end
        WRITE : begin
                if (SS_n)
                    ns = IDLE;
                else
                    ns = WRITE;
        end
        READ_ADD : begin
                if (SS_n)
                    ns = IDLE;
                else
                    ns = READ_ADD;
        end
        READ_DATA : begin
                if (SS_n)
                    ns = IDLE;
                else
                    ns = READ_DATA;
        end

        default : ns = IDLE;
    endcase
end
```

```
// Output logic
always @(posedge clk) begin
    if (~rst_n) begin
        rx_data <= 0;
        rx_valid <= 0;
        counter <= 0;
        MISO <= 0;
        read_addr_received <= 0;
        tx_data_reg <= 0;
        shift_reg <= 0;
    end

    else if ((cs == IDLE) || (cs == CHK_CMD)) begin
        rx_valid <= 0;
        shift_reg <= 0;
        counter <= 0;
        MISO <= 0;
        tx_data_reg <= 0;
    end

    else if ((cs == WRITE) || (cs == READ_ADD)) begin
        // 10 counts to receive 10 MOSI bits, 1 count to store shift_reg in rx_data
        if (counter < 11) begin
            counter <= counter + 1;
            if (counter < 10) begin
                shift_reg <= {shift_reg[8:0], MOSI}; // Shift in MOSI data
            end
            else begin
                rx_valid <= 1;
                rx_data <= shift_reg;
                if (cs == READ_ADD)
                    read_addr_received <= 1;
            end
        end
        else begin
            rx_valid <= 0;
        end
    end
```

```
        else if (cs == READ_DATA) begin
            if (tx_valid)
                tx_data_reg <= tx_data; // Load data for transmission
            else begin
                if (counter < 11) begin
                    counter <= counter + 1;
                    if (counter < 10) begin
                        shift_reg <= {shift_reg[8:0], MOSI}; // Shift in MOSI data
                    end
                    else begin
                        rx_valid <= 1;
                        rx_data <= shift_reg;
                    end
                end
                else begin
                    // counter >= 11
                    counter <= counter + 1;
                    {MISO,tx_data_reg} <= {tx_data_reg,1'b0}; // Shift out data on MISO
                    rx_valid <= 0;

                end
            end
        end

        else begin
            rx_valid <= 0;
        end
    end

endmodule
```

RAM.v

```
// Single Port Synchronous RAM
module RAM (clk, rst_n, rx_valid, din, tx_valid, dout);

    parameter MEM_DEPTH = 256;
    parameter ADDR_SIZE = 8;

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

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

    reg [ADDR_SIZE-1:0] write_addr, read_addr;
    reg [7:0] mem [MEM_DEPTH-1:0];

    always @(posedge clk) begin

        if (~rst_n) begin
            dout <= 0;
            tx_valid <= 0;
            write_addr <= 0;
            read_addr <= 0;
        end

        else begin
            tx_valid <= 0; // Default value

            if (rx_valid) begin
                case (din[9:8]) // Check Command bits
                    2'b00 : write_addr <= din[7:0];
                    2'b01 : mem[write_addr] <= din[7:0];
                    2'b10 : read_addr <= din[7:0];
                    2'b11 : begin
                            dout <= mem[read_addr];
                            tx_valid <= 1;
                    end
                endcase
            end

        end
    end
endmodule
```

TESTBENCH CODE

```
module SPI_master_tb();

    reg clk, rst_n;
    reg MOSI;
    reg SS_n;
    wire MISO;

    integer i;
    reg [9:0] temp_data;

    SPI_top_module DUT(clk, rst_n, SS_n, MOSI, MISO);

    // Clock generation
    initial begin
        clk = 0;
        forever #1 clk = ~clk;
    end

    // Test
    initial begin

        // Initialize memory
        $readmemh("mem.dat", DUT.RAM_inst.mem);

        // Reset and Initialize signals
        rst_n = 0;
        SS_n = 1;
        MOSI = 0;
        temp_data = 0;
        @(negedge clk);
        rst_n = 1;

        // ================================================
        // Test 1: address 0xff , data 0x7d
        // ================================================
        // Write address 0xff (10'b00_1111_1111)
        SS_n = 0; @(negedge clk);
        MOSI = 0; @(negedge clk); // Write command
        temp_data = 10'b00_1111_1111;
        for (i = 9; i >= 0; i = i - 1) begin
            MOSI = temp_data[i];
            @(negedge clk);
        end
        @(negedge clk);
        SS_n = 1; @(negedge clk);
```

```
// Write data 0x7d (10'b01_0111_1101)
SS_n = 0; @(negedge clk);
MOSI = 0; @(negedge clk); // Write command
temp_data = 10'b01_0111_1101;
for (i = 9; i >= 0; i = i - 1) begin
    MOSI = temp_data[i];
    @(negedge clk);
end
@(negedge clk);
SS_n = 1; @(negedge clk);

// Read address 0xff (10'b10_1111_1111)
SS_n = 0; @(negedge clk);
MOSI = 1; @(negedge clk); // Read command
temp_data = 10'b10_1111_1111;
for (i = 9; i >= 0; i = i - 1) begin
    MOSI = temp_data[i];
    @(negedge clk);
end
@(negedge clk);
SS_n = 1; @(negedge clk);

// Read data (expecting 0111_1101 on MISO)
SS_n = 0; @(negedge clk);
MOSI = 1; @(negedge clk); // Read command
MOSI = 1; @(negedge clk);
MOSI = 1; @(negedge clk);
// Send dummy bits
for (i = 0; i < 8; i = i + 1) begin
    MOSI = 0; @(negedge clk);
end
// Extra clocks to observe MISO
repeat(10) @(negedge clk);
SS_n = 1; @(negedge clk);
```

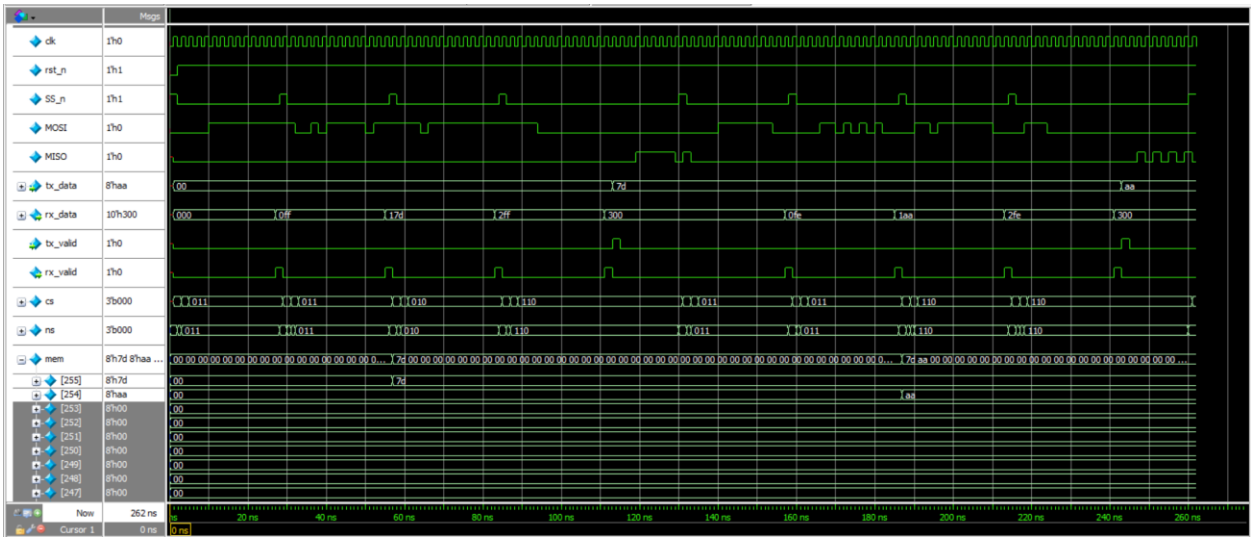
```
        // ================================================
        // Test 2: address 0xfe , data 0xaa
        // ================================================

        // Write address 0xfe (10'b00_1111_1110)
        SS_n = 0; @(negedge clk);
        MOSI = 0; @(negedge clk); // Write command
        temp_data = 10'b00_1111_1110;
        for (i = 9; i >= 0; i = i - 1) begin
            MOSI = temp_data[i];
            @(negedge clk);
        end
        @(negedge clk);
        SS_n = 1; @(negedge clk);

        // Write data 0xaa (10'b01_1010_1010)
        SS_n = 0; @(negedge clk);
        MOSI = 0; @(negedge clk); // Write command
        temp_data = 10'b01_1010_1010;
        for (i = 9; i >= 0; i = i - 1) begin
            MOSI = temp_data[i];
            @(negedge clk);
        end
        @(negedge clk);
        SS_n = 1; @(negedge clk);

        // Read address 0xfe (10'b10_1111_1110)
        SS_n = 0; @(negedge clk);
        MOSI = 1; @(negedge clk); // Read command
        temp_data = 10'b10_1111_1110;
        for (i = 9; i >= 0; i = i - 1) begin
            MOSI = temp_data[i];
            @(negedge clk);
        end
        @(negedge clk);
        SS_n = 1; @(negedge clk);

        // Read data (expecting 1010_1010 on MISO)
        SS_n = 0; @(negedge clk);
        MOSI = 1; @(negedge clk); // Read command
        MOSI = 1; @(negedge clk);
        MOSI = 1; @(negedge clk);
        // Send dummy bits
        for (i = 0; i < 8; i = i + 1) begin
            MOSI = 0; @(negedge clk);
        end
        // Extra clocks to observe MISO
        repeat(10) @(negedge clk);
        SS_n = 1; @(negedge clk);

        $stop;
    end
endmodule
```

DO FILE

```
vlib work
vlog RAM.v SPI_slave.v SPI_top_module.v SPI_master_tb.v
vsim -voptargs=+acc work.SPI_master_tb
add wave -position insertpoint  \
sim:/SPI_master_tb/clk \
sim:/SPI_master_tb/rst_n \
sim:/SPI_master_tb/SS_n \
sim:/SPI_master_tb/MOSI \
sim:/SPI_master_tb/MISO \
sim:/SPI_master_tb/DUT/SPI_slave_inst/tx_data \
sim:/SPI_master_tb/DUT/SPI_slave_inst/rx_data \
sim:/SPI_master_tb/DUT/SPI_slave_inst/tx_valid \
sim:/SPI_master_tb/DUT/SPI_slave_inst/rx_valid \
sim:/SPI_master_tb/DUT/SPI_slave_inst/cs \
sim:/SPI_master_tb/DUT/SPI_slave_inst/ns \
sim:/SPI_master_tb/DUT/RAM_inst/mem
run -all
#quit -sim
```

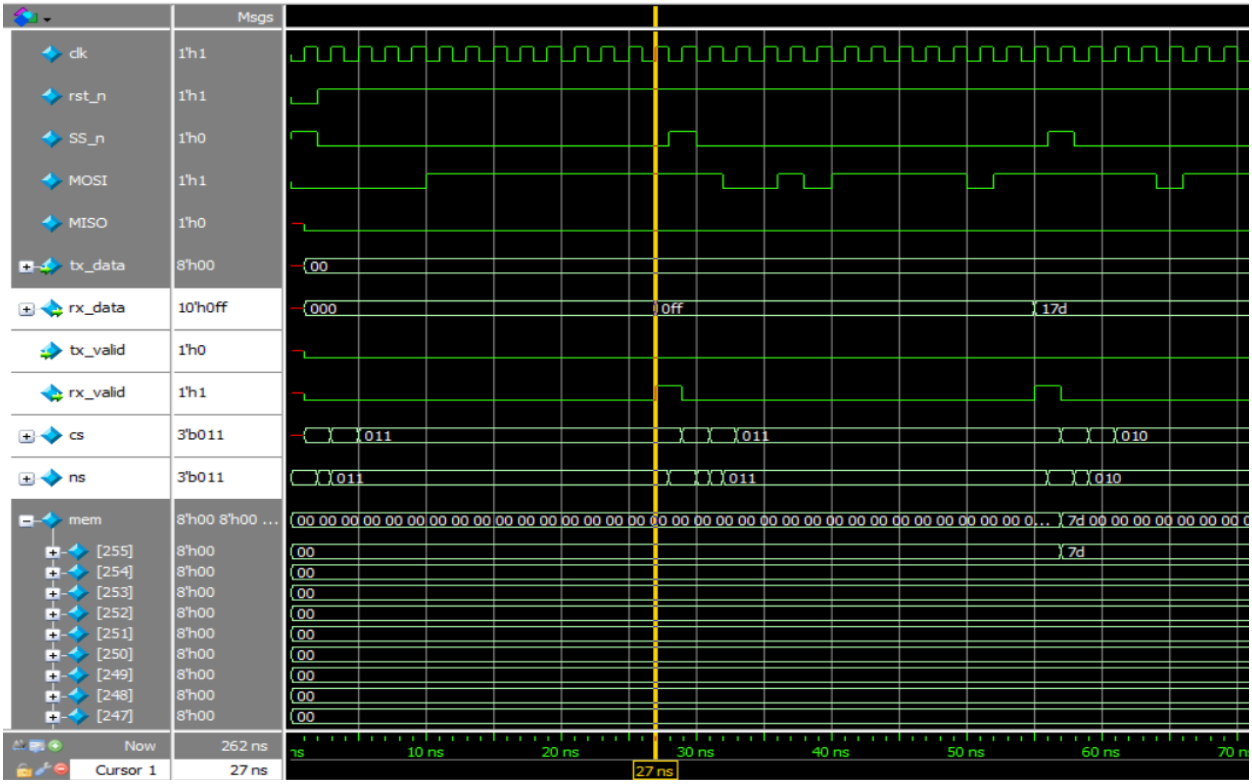
QUESTASIM SNIPPETS

Full Wave

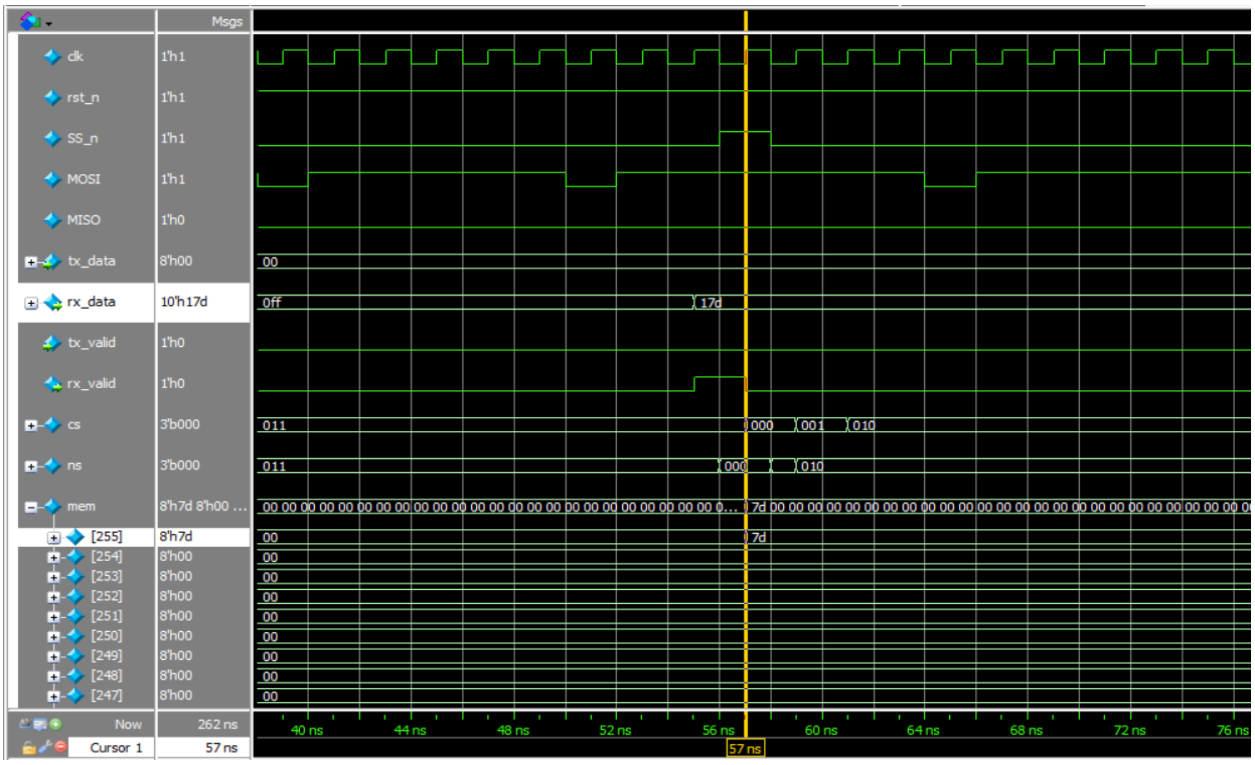


- In the first test case, the testbench writes the value 0x7D to memory address 0xFF by first sending a write address command (00_1111_1111), followed by a write data command (01_0111_1101). It then verifies the write operation by performing a read sequence - first sending a read address command (10_1111_1111), then executing a read data operation where the slave returns the stored value 0x7D via the MISO line. The second test case follows the identical pattern but uses address 0xFE and data value 0xAA.

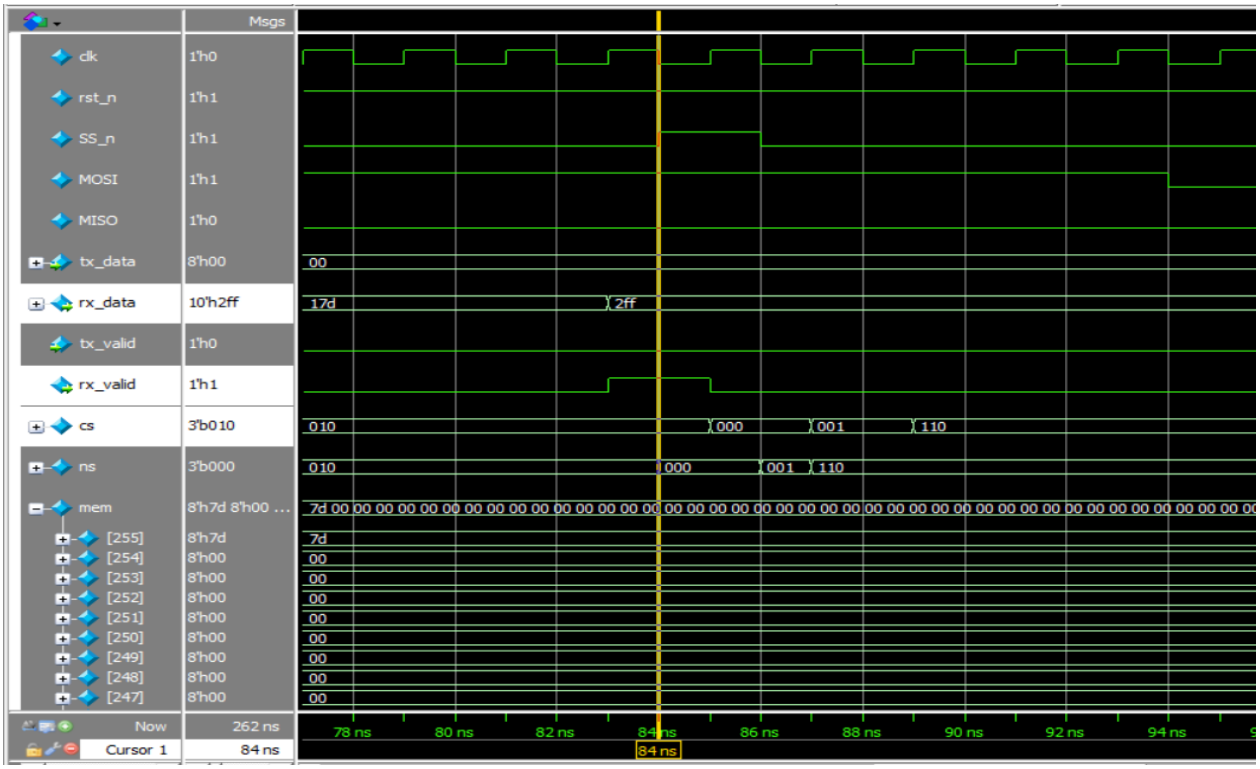
Write Address 0xff



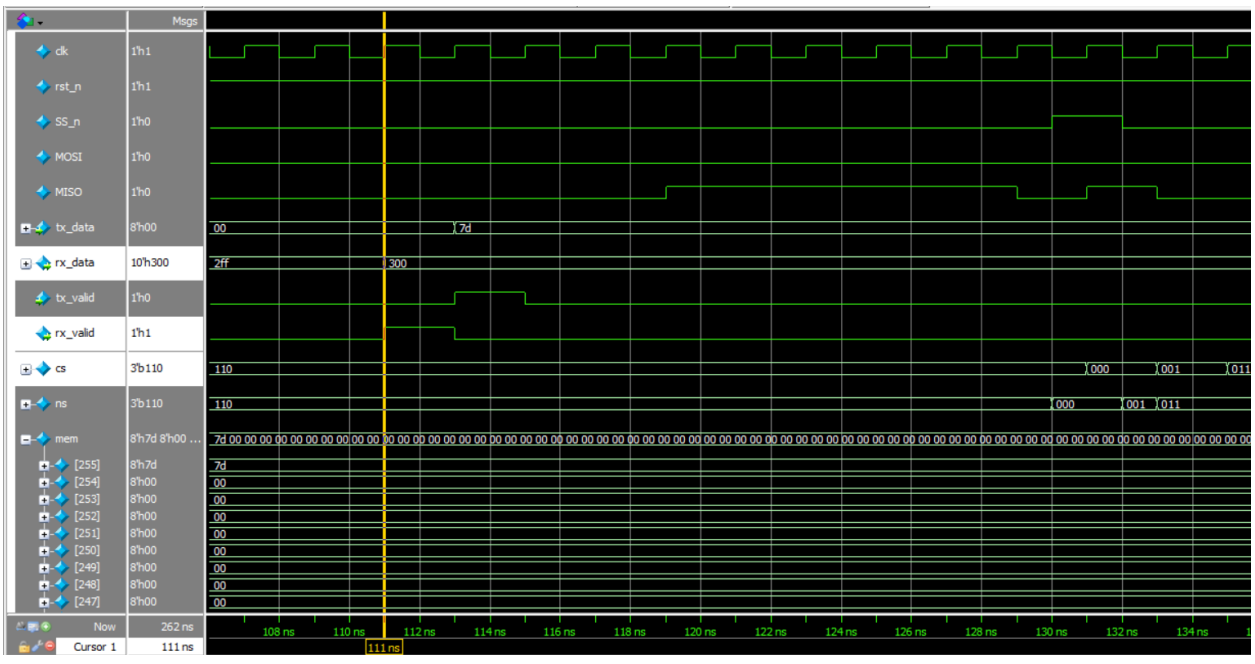
Write Data 0x7d



Read Address 0xff



Read Data

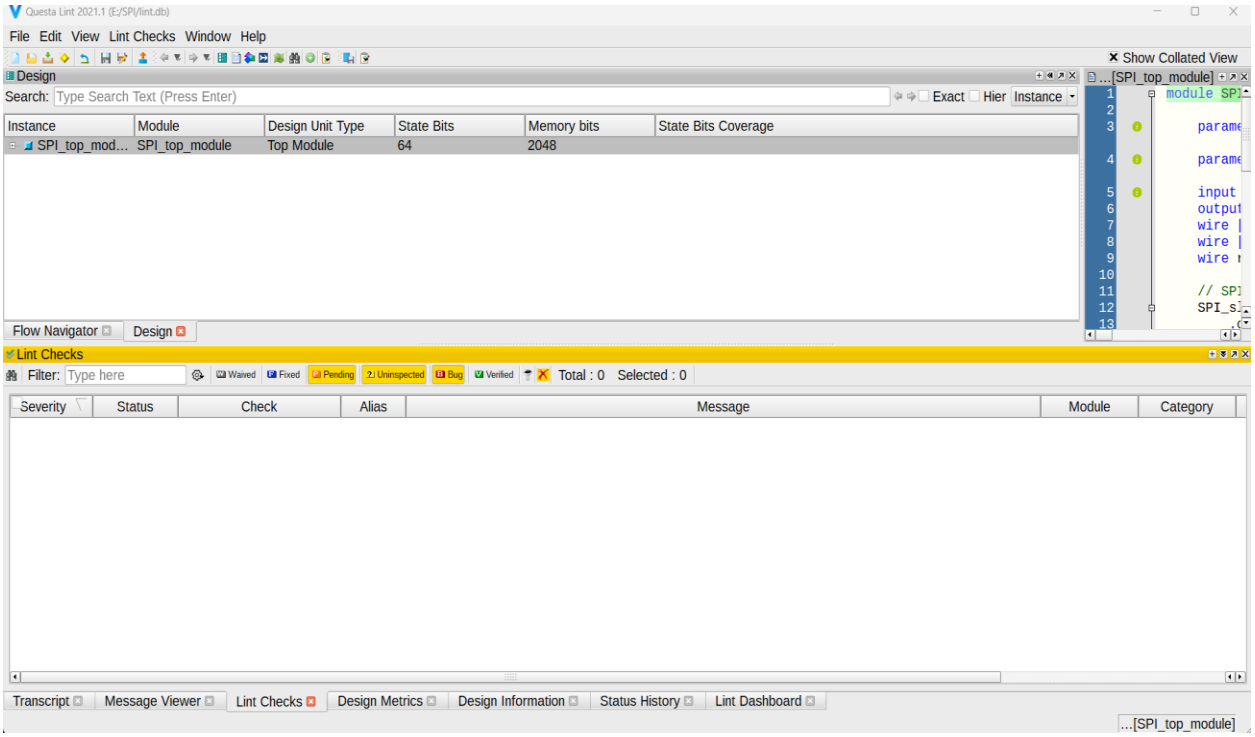


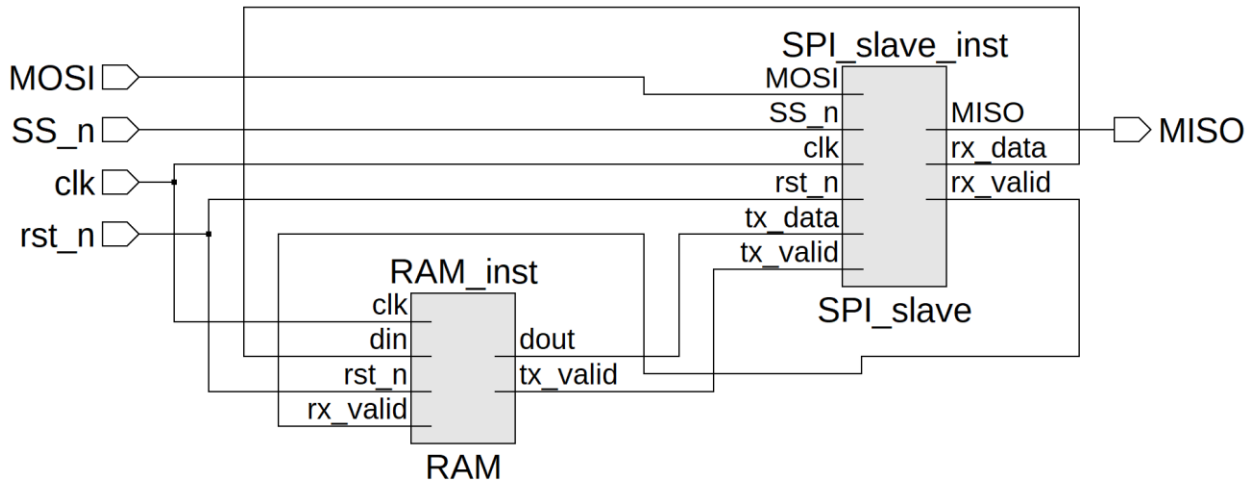
- The value 0111_1101 appears on MISO as expected.

CONSTRAINT FILE

```
## Clock signal
set_property -dict { PACKAGE_PIN W5   IOSTANDARD LVCMOS33 } [get_ports clk]
create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]

## Switches
set_property -dict { PACKAGE_PIN V17   IOSTANDARD LVCMOS33 } [get_ports {rst_n}]
set_property -dict { PACKAGE_PIN V16   IOSTANDARD LVCMOS33 } [get_ports {SS_n}]
set_property -dict { PACKAGE_PIN W16   IOSTANDARD LVCMOS33 } [get_ports {MOSI}]

## LEDs
set_property -dict { PACKAGE_PIN U16   IOSTANDARD LVCMOS33 } [get_ports {MISO}]

## Configuration options, can be used for all designs
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBVS VCCO [current_design]

## SPI configuration mode options for QSPI boot, can be used for all designs
set_property BITSTREAM.GENERAL.COMPRESS TRUE [current_design]
set_property BITSTREAM.CONFIG.CONFIGRATE 33 [current_design]
set_property CONFIG_MODE SPIx4 [current_design]
```

LINTING


Questa Lint 2021.1 (E:/SPI/lint.db)
File Edit View Lint Checks Window Help
Show Collated View
Design
Search: Type Search Text (Press Enter)
Exact Hier Instance
Instance Module Design Unit Type State Bits Memory bits State Bits Coverage
SPI_top_mod... SPI_top_module Top Module 64 2048
...[SPI_top_module]
Flow Navigator Design
Lint Checks
Filter: Type here
Waived Fixed Pending Uninspected Bug Verified
Total : 0 Selected : 0
Severity Status Check Alias Message Module Category
Transcript Message Viewer Lint Checks Design Metrics Design Information Status History Lint Dashboard
...[SPI_top_module]

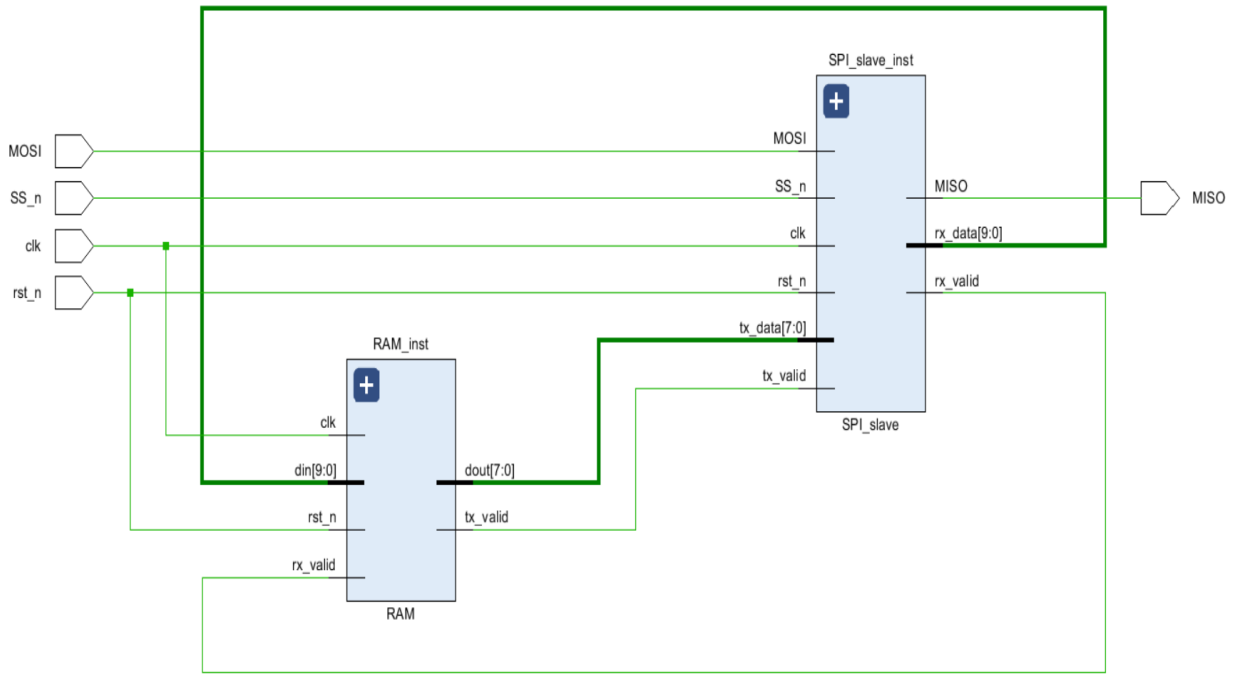

MOSI
SS_n
clk
rst_n
SPI_slave_inst
MOSI
SS_n
clk
rst_n
tx_data
tx_valid
MISO
rx_data
rx_valid
SPI_slave
MISO
RAM_inst
clk
din
rst_n
rx_valid
dout
tx_valid
RAM

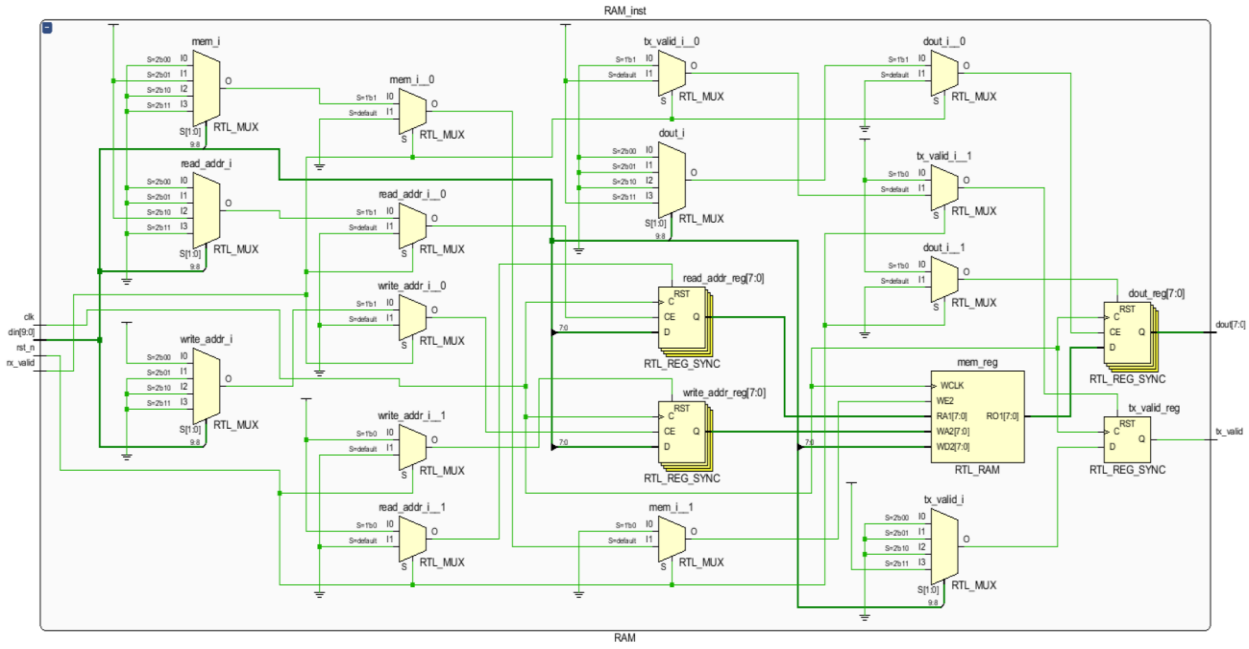
ELABORATION “GRAY ENCODING”

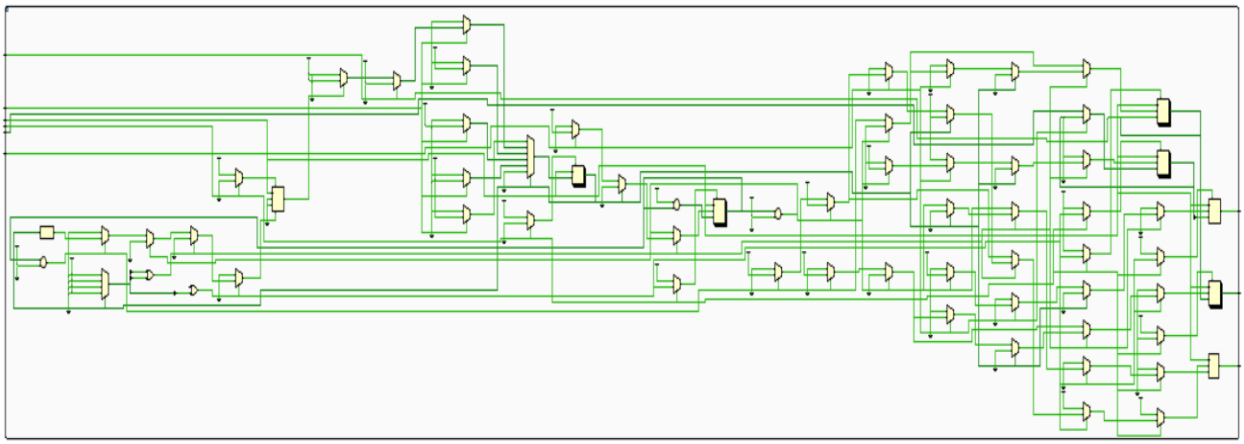
Messages

- Elaborated Design (11 infos)
  - General Messages (11 infos)
    - [Synth 8-6157] synthesizing module 'SPI_top_module' [SPI_top_module.v:1] (2 more like this)
    - [Synth 8-5534] Detected attribute (* fsm_encoding = "gray" *) [SPI_slave.v:29]
    - [Synth 8-6155] done synthesizing module 'SPI_slave' (1#1) [SPI_slave.v:1] (2 more like this)
    - [Device 21-403] Loading part xc7a35ticpg236-1L
    - [Project 1-570] Preparing netlist for logic optimization
    - [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
    - [Project 1-111] Unisim Transformation Summary:
      No Unisim elements were transformed.

Schematic




RAM_inst
mem_i
mem_i__0
tx_valid_i__0
dout_i__0
read_addr_i
read_addr_i__0
dout_i
tx_valid_i__1
write_addr_i__0
dout_i__1
read_addr_reg[7:0]
dout_reg[7:0]
clk
din[9:0]
rst_n
rx_valid
write_addr_i
write_addr_i__1
write_addr_reg[7:0]
mem_reg
tx_valid_reg
dout[7:0]
tx_valid
read_addr_i__1
mem_i__1
tx_valid_i
RTL_MUX
RTL_REG_SYNC
RTL_RAM
RAM



SYNTHESIS “GRAY ENCODING”

Messages

- Synthesized Design (9 infos)
  - General Messages (9 infos)
    - [Netlist 29-17] Analyzing 5 Unisim elements for replacement
    - [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
    - [Project 1-479] Netlist was created with Vivado 2018.2
    - [Project 1-570] Preparing netlist for logic optimization
    - [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
    - [Project 1-111] Unisim Transformation Summary:
      No Unisim elements were transformed.
    - [Timing 38-35] Done setting XDC timing constraints.
    - [Timing 38-91] UpdateTimingParams: Speed grade: -1L, Delay Type: min_max.
    - [Timing 38-191] Multithreading enabled for timing update using a maximum of 2 CPUs

Utilization report

Name	Slice LUTs (20800)	Slice Registers (41600)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
SPI_top_module	26	56	0.5	5	1
RAM_inst (RAM)	5	17	0.5	0	0
SPI_slave_inst (SPI_sl...	21	39	0	0	0

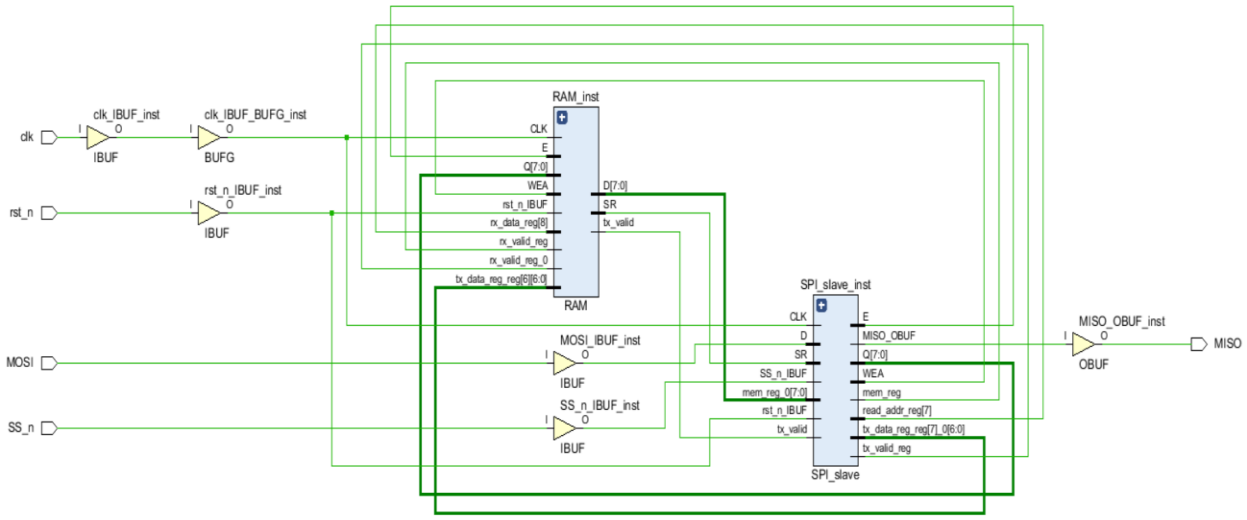
timing report

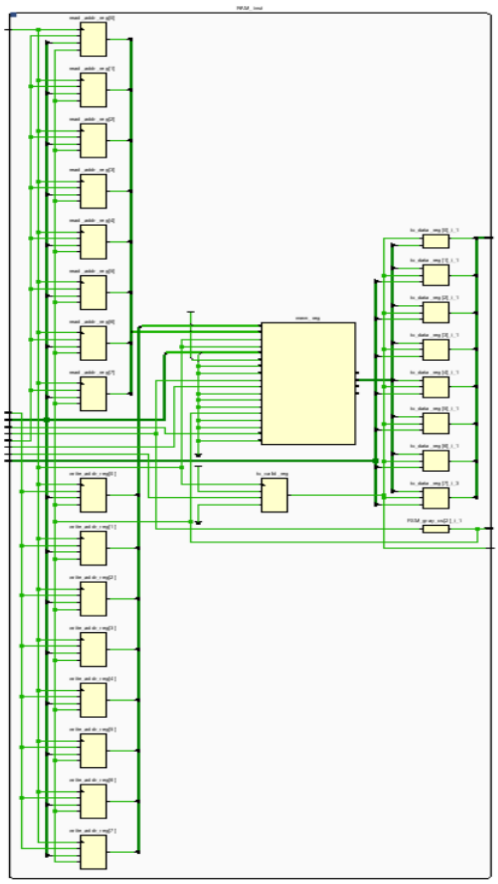
Design Timing Summary

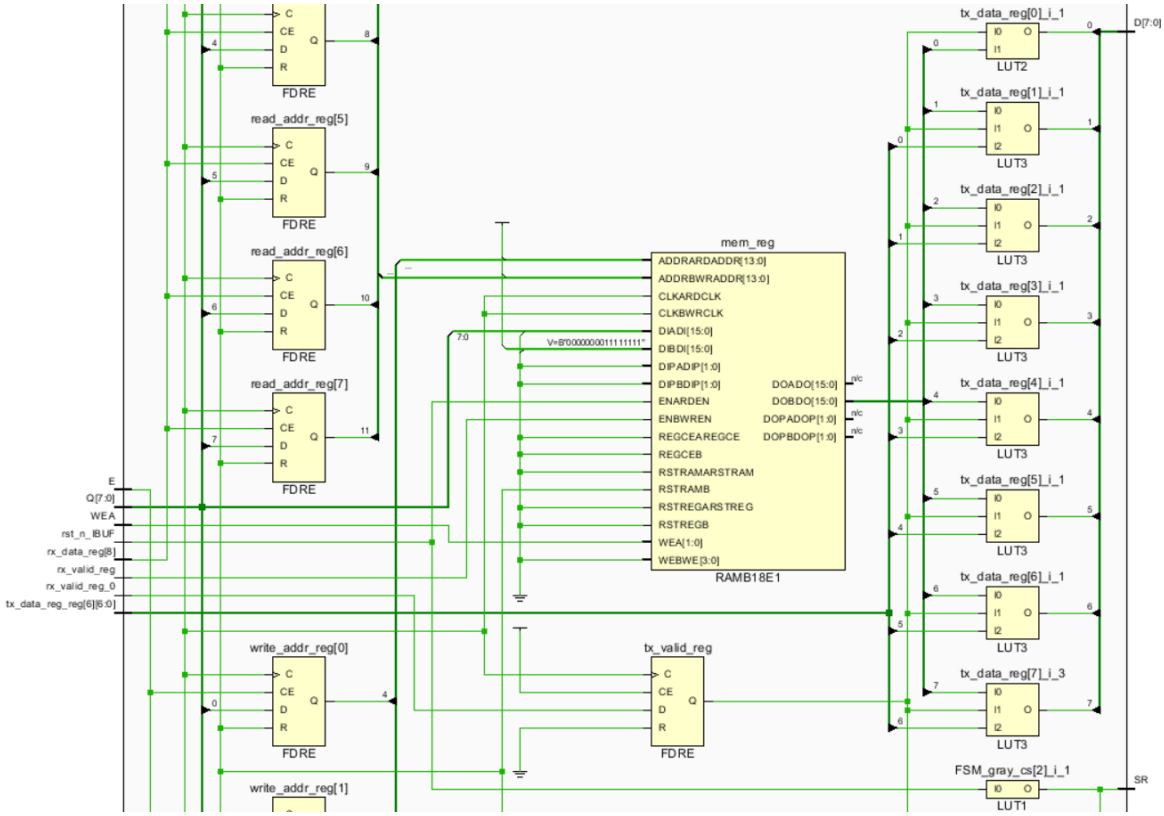
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	6.445 ns	Worst Hold Slack (WHS):	0.134 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	156	Total Number of Endpoints:	156	Total Number of Endpoints:	59

All user specified timing constraints are met.

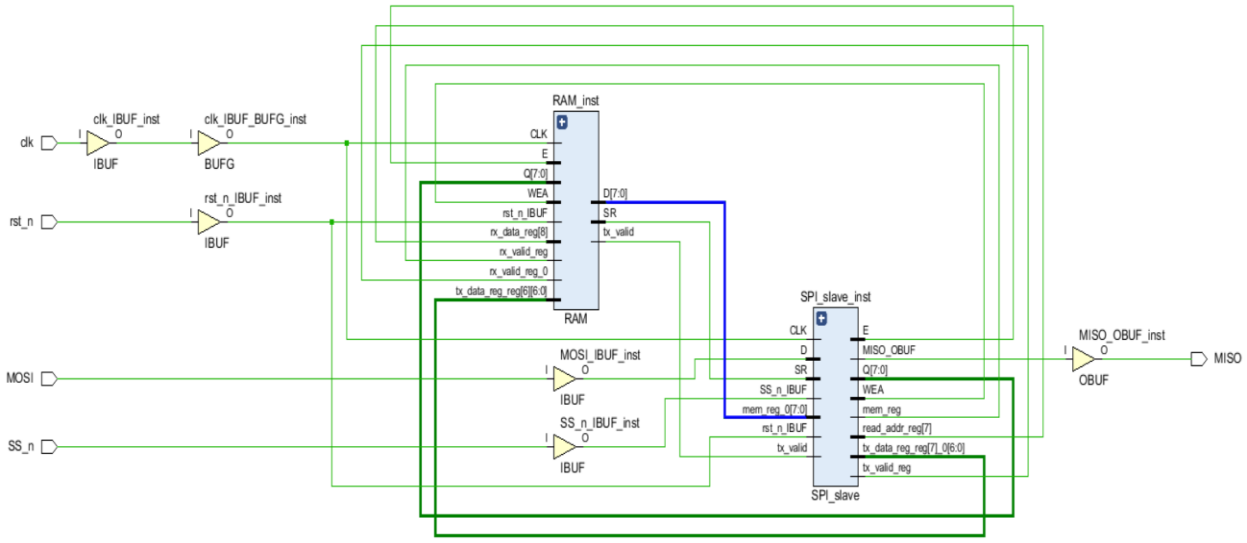
Schematic






FDRE
read_addr_reg[5]
read_addr_reg[6]
read_addr_reg[7]
mem_reg
ADDRARDADDR[13:0]
ADDRBWRADDR[13:0]
CLKARDCLK
CLKBWRCLK
DIADI[15:0]
DIBDI[15:0]
DIPADIP[1:0]
DIPBDIP[1:0]
ENARDEN
ENBWREN
REGCEAREGCE
REGCEB
RSTRAMARSTRAM
RSTRAMB
RSTREGARSTREG
RSTREGB
WEA[1:0]
WEBWE[3:0]
DOADO[15:0]
DOBDO[15:0]
DOPADOP[1:0]
DOPBDOP[1:0]
RAMB18E1
E
Q[7:0]
WEA
rst_n_IBUF
rx_data_reg[8]
rx_valid_reg
rx_valid_reg_0
tx_data_reg_reg[6][6:0]
write_addr_reg[0]
write_addr_reg[1]
tx_valid_reg
tx_data_reg[0]_i_1
tx_data_reg[1]_i_1
tx_data_reg[2]_i_1
tx_data_reg[3]_i_1
tx_data_reg[4]_i_1
tx_data_reg[5]_i_1
tx_data_reg[6]_i_1
tx_data_reg[7]_i_3
LUT2
LUT3
FSM_gray_cs[2]_i_1
LUT1
D[7:0]
SR

Critical path



Synthesis report

```
-----------------------------------------------------------------------------------------------
                State |                New Encoding |               Previous Encoding
-----------------------------------------------------------------------------------------------
                 IDLE |                         000 |                              000
              CHK_CMD |                         111 |                              001
            READ_DATA |                         001 |                              110
             READ_ADD |                         011 |                              010
                WRITE |                         010 |                              011
-----------------------------------------------------------------------------------------------
INFO: [Synth 8-3354] encoded FSM with state register 'cs_reg' using encoding 'gray' in module 'SPI_slave'
-----------------------------------------------------------------------------
Finished RTL Optimization Phase 2 : Time (s): cpu = 00:00:27 ; elapsed = 00:00:29 . Memory (MB): peak = 765.062 ; gain = 507.637
-----------------------------------------------------------------------------
```

IMPLEMENTATION “GRAY ENCODING”

Messages

- Implemented Design (9 infos)
  - General Messages (9 infos)
    - [Netlist 29-17] Analyzing 5 Unisim elements for replacement
    - [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
    - [Project 1-479] Netlist was created with Vivado 2018.2
    - [Project 1-570] Preparing netlist for logic optimization
    - [Timing 38-478] Restoring timing data from binary archive.
    - [Timing 38-479] Binary timing data restore complete.
    - [Project 1-856] Restoring constraints from binary archive.
    - [Project 1-853] Binary constraint restore complete.
    - [Project 1-111] Unisim Transformation Summary:
      No Unisim elements were transformed.

Utilization report

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)
N SPI_top_module	27	56	12	27	13	0.5	5	1
RAM_inst (RAM)	6	17	5	6	0	0.5	0	0
SPI_slave_inst (SPI_sl...	21	39	10	21	9	0	0	0

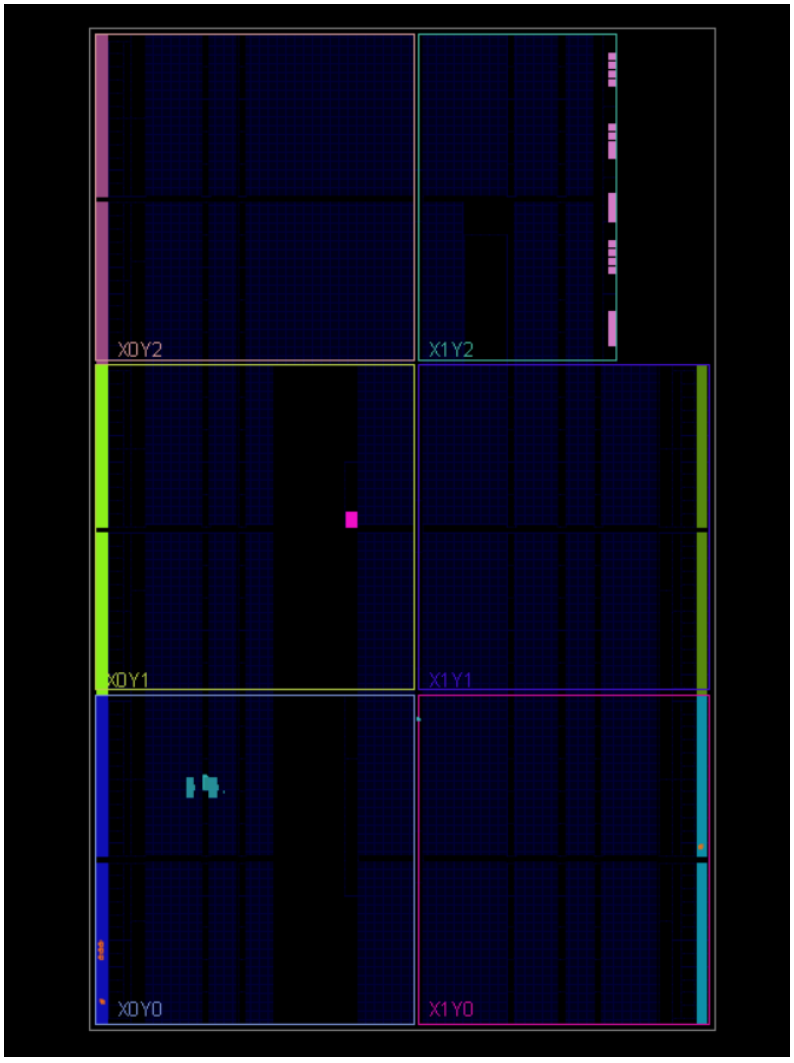
timing report

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	6.382 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
Total Number of Endpoints:	157	Total Number of Endpoints:	157	Total Number of Endpoints:	59

All user specified timing constraints are met.

Device

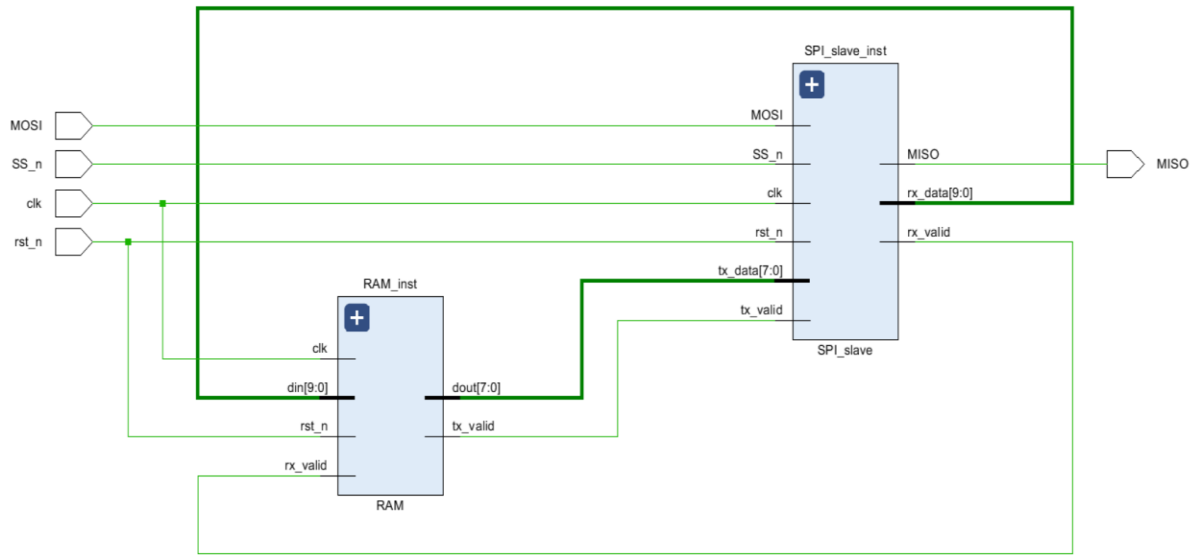


ELABORATION “SEQUENTIAL ENCODING”

Messages

- Elaborated Design (10 infos)
  - General Messages (10 infos)
    - [Synth 8-6157] synthesizing module 'SPI_top_module' [SPI_top_module.v:1] (2 more like this)
    - [Synth 8-5534] Detected attribute (* fsm_encoding = "sequential" *) [SPI_slave.v:30]
    - [Synth 8-6155] done synthesizing module 'SPI_slave' (1#1) [SPI_slave.v:1] (2 more like this)
    - [Project 1-570] Preparing netlist for logic optimization
    - [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
    - [Project 1-111] Unisim Transformation Summary:
      No Unisim elements were transformed.

Schematic



SYNTHESIS “SEQUENTIAL ENCODING”

Messages

- Synthesized Design (6 infos)
  - General Messages (6 infos)
    - [Netlist 29-17] Analyzing 5 Unisim elements for replacement
    - [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
    - [Project 1-479] Netlist was created with Vivado 2018.2
    - [Project 1-570] Preparing netlist for logic optimization
    - [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
    - [Project 1-111] Unisim Transformation Summary:
      No Unisim elements were transformed.

Utilization report

Name	Slice LUTs (20800)	Slice Registers (41600)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
N SPI_top_module	26	56	0.5	5	1
RAM_inst (RAM)	5	17	0.5	0	0
SPI_slave_inst (SPI_sl...	21	39	0	0	0

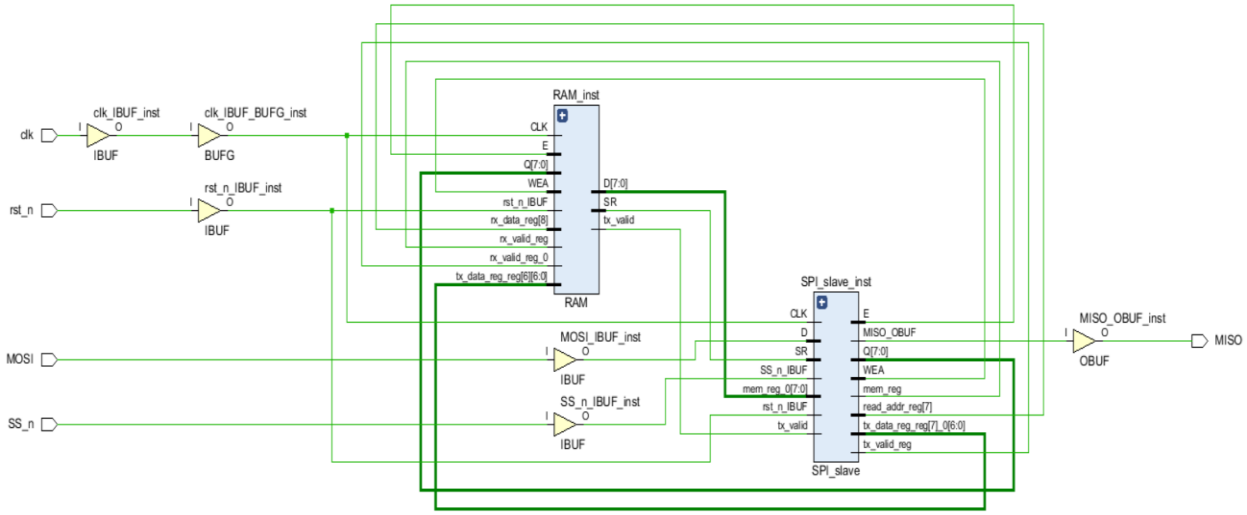
timing report

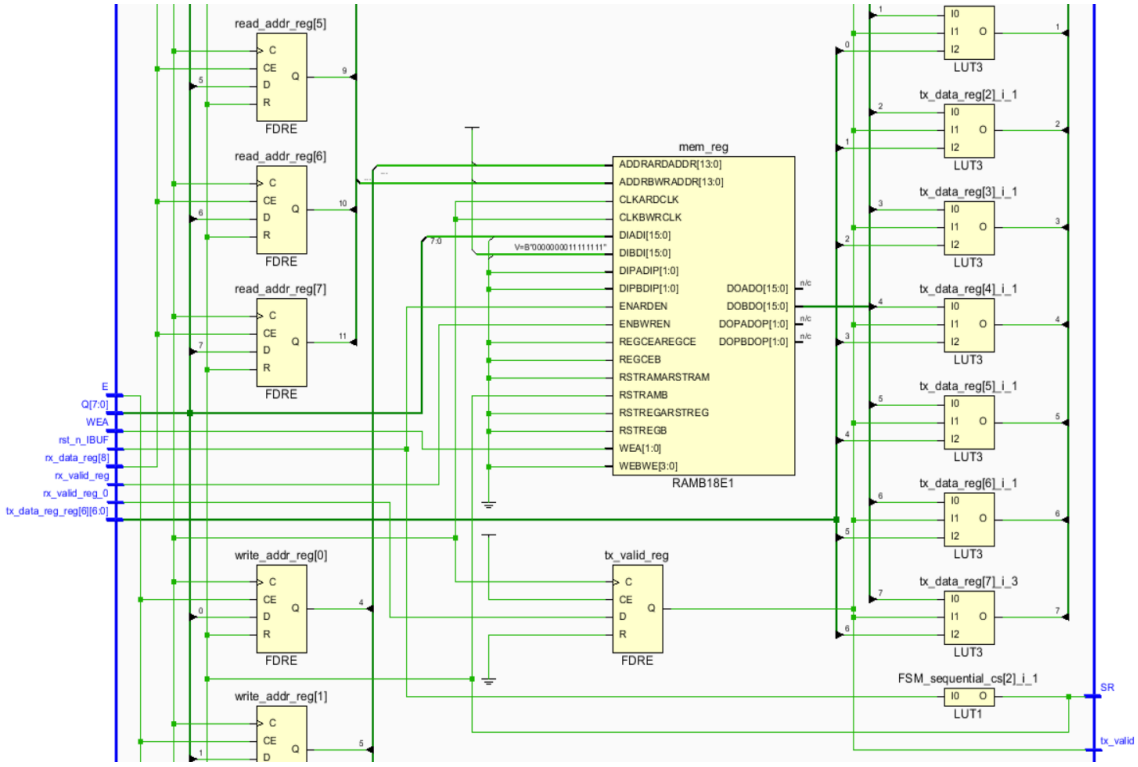
Design Timing Summary

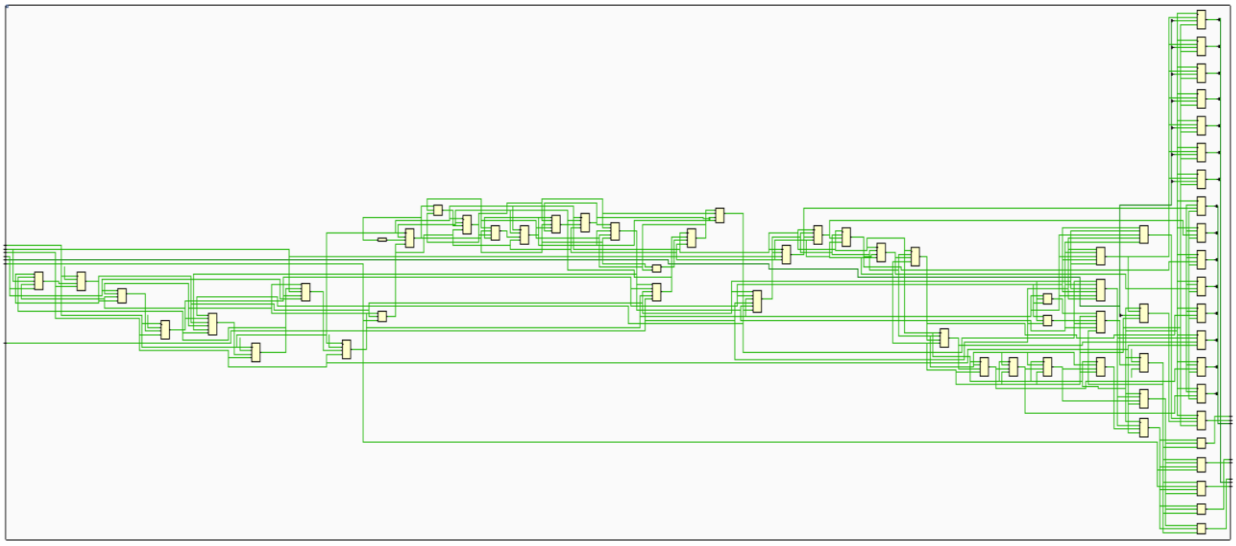
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	6.445 ns	Worst Hold Slack (WHS):	0.134 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	156	Total Number of Endpoints:	156	Total Number of Endpoints:	59

All user specified timing constraints are met.

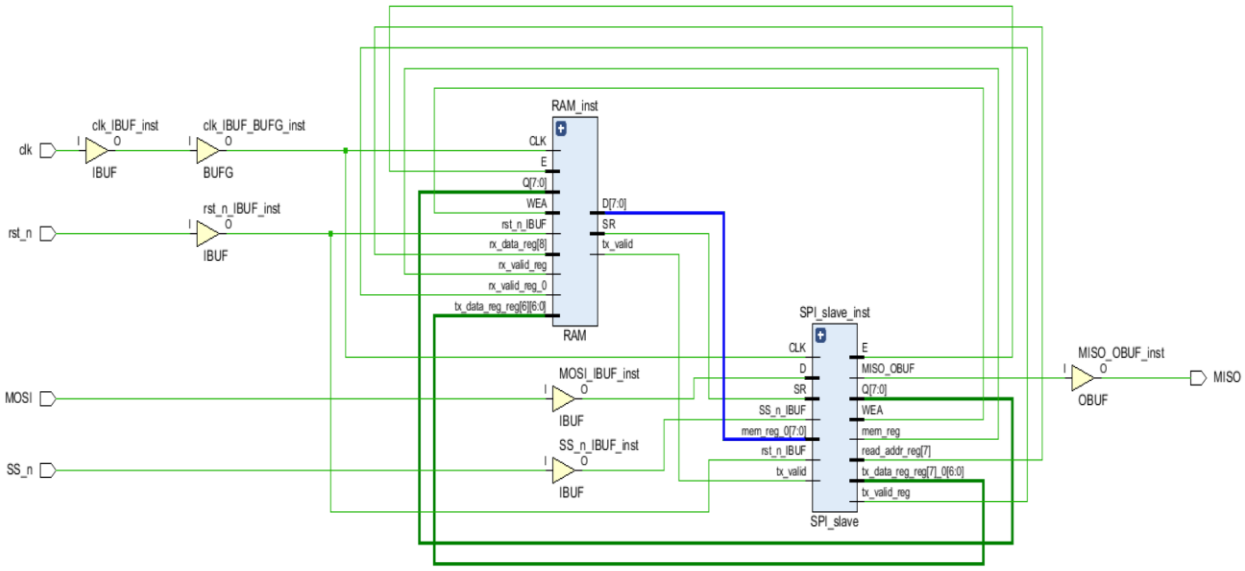
Schematic


clk
clk_IBUF_inst
IBUF
clk_IBUF_BUFG_inst
BUFG
rst_n
rst_n_IBUF_inst
IBUF
RAM_inst
CLK
E
Q[7:0]
WEA
rst_n_IBUF
rx_data_reg[8]
rx_valid_reg
rx_valid_reg_0
tx_data_reg_reg[6][6:0]
D[7:0]
SR
tx_valid
RAM
MOSI
MOSI_IBUF_inst
IBUF
SS_n
SS_n_IBUF_inst
IBUF
SPI_slave_inst
CLK
D
SR
SS_n_IBUF
mem_reg_0[7:0]
rst_n_IBUF
tx_valid
E
MISO_OBUF
Q[7:0]
WEA
mem_reg
read_addr_reg[7]
tx_data_reg_reg[7]_0[6:0]
tx_valid_reg
SPI_slave
MISO_OBUF_inst
OBUF
MISO


read_addr_reg[5]
read_addr_reg[6]
read_addr_reg[7]
FDRE
mem_reg
ADDRARDADDR[13:0]
ADDRBWRADDR[13:0]
CLKARDCLK
CLKBWRCLK
DIADI[15:0]
DIBDI[15:0]
DIPADIP[1:0]
DIPBDIP[1:0]
ENARDEN
ENBWREN
REGCEAREGCE
REGCEB
RSTRAMARSTRAM
RSTRAMB
RSTREGARSTREG
RSTREGB
WEA[1:0]
WEBWE[3:0]
DOADO[15:0]
DOBDO[15:0]
DOPADOP[1:0]
DOPBDOP[1:0]
RAMB18E1
LUT3
tx_data_reg[2]_i_1
tx_data_reg[3]_i_1
tx_data_reg[4]_i_1
tx_data_reg[5]_i_1
tx_data_reg[6]_i_1
tx_data_reg[7]_i_3
write_addr_reg[0]
write_addr_reg[1]
tx_valid_reg
FSM_sequential_cs[2]_i_1
LUT1
SR
tx_valid



Critical path



Synthesis report

```
---------------------------------------------------------------------------------------------------
               State |                     New Encoding |               Previous Encoding
---------------------------------------------------------------------------------------------------
                IDLE |                              000 |                              000
             CHK_CMD |                              100 |                              001
           READ_DATA |                              001 |                              110
            READ_ADD |                              010 |                              010
               WRITE |                              011 |                              011
---------------------------------------------------------------------------------------------------
INFO: [Synth 8-3354] encoded FSM with state register 'cs_reg' using encoding 'sequential' in module 'SPI_slave'
---------------------------------------------------------------------------------------------------
Finished RTL Optimization Phase 2 : Time (s): cpu = 00:00:24 ; elapsed = 00:00:26 . Memory (MB): peak = 764.781 ; gain = 507.711
---------------------------------------------------------------------------------------------------
```

IMPLEMENTATION “SEQUENTIAL ENCODING”

Messages

- Implemented Design (9 infos)
  - General Messages (9 infos)
    - [Netlist 29-17] Analyzing 5 Unisim elements for replacement
    - [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
    - [Project 1-479] Netlist was created with Vivado 2018.2
    - [Project 1-570] Preparing netlist for logic optimization
    - [Timing 38-478] Restoring timing data from binary archive.
    - [Timing 38-479] Binary timing data restore complete.
    - [Project 1-856] Restoring constraints from binary archive.
    - [Project 1-853] Binary constraint restore complete.
    - [Project 1-111] Unisim Transformation Summary:
      No Unisim elements were transformed.

Utilization report

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)
N SPI_top_module	27	56	14	27	11	0.5	5	1
I RAM_inst (RAM)	6	17	5	6	0	0.5	0	0
I SPI_slave_inst (SPI_sl...	21	39	12	21	7	0	0	0

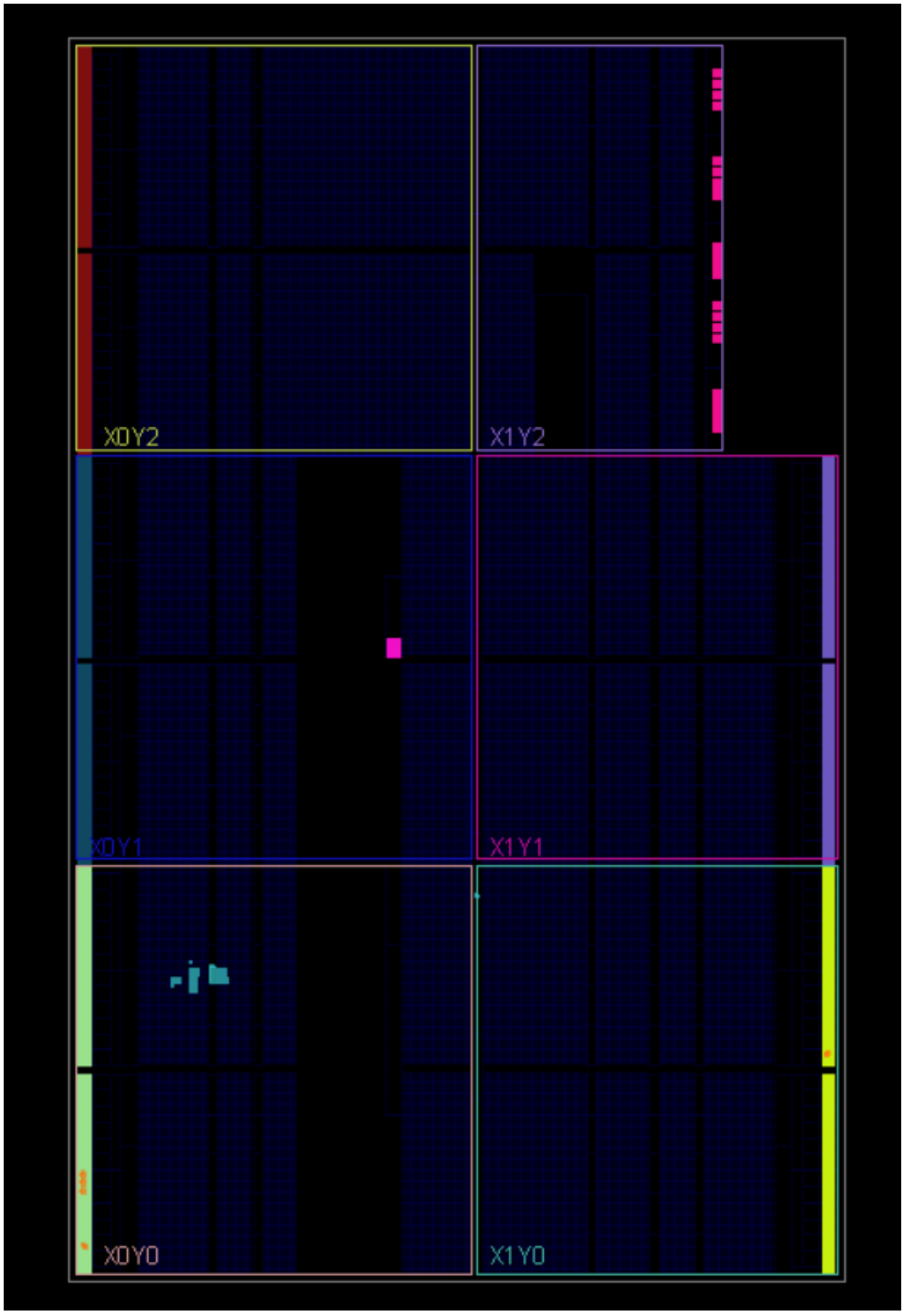
timing report

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	6.139 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
Total Number of Endpoints:	157	Total Number of Endpoints:	157	Total Number of Endpoints:	59

All user specified timing constraints are met.

Device

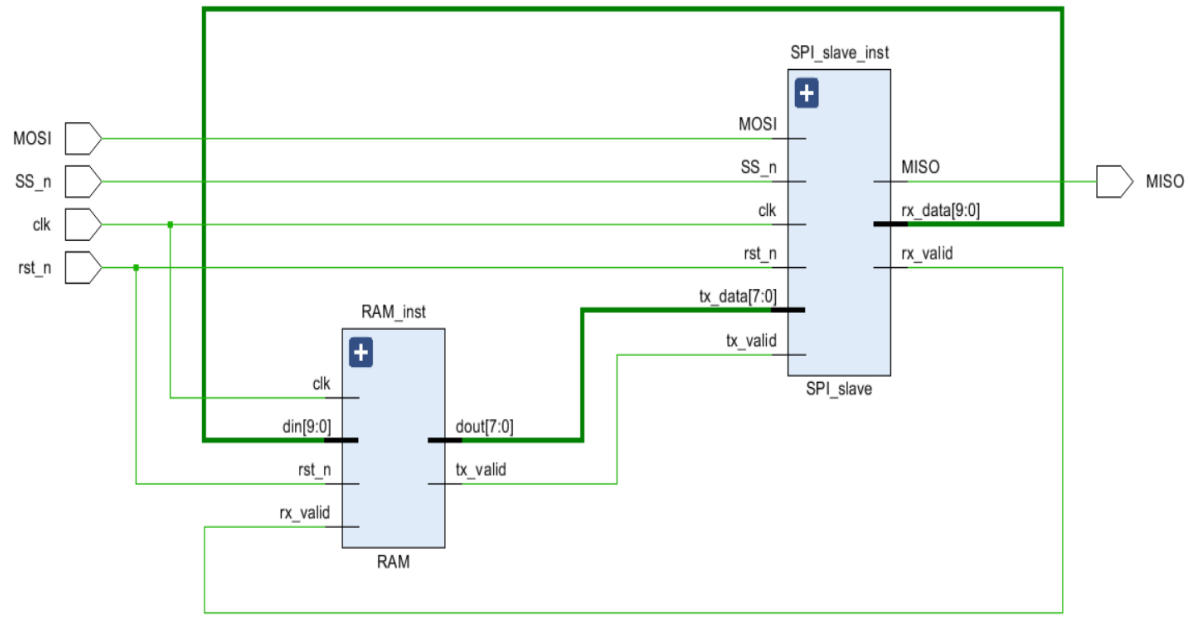


ELABORATION “ONE HOT ENCODING”

Messages

- Elaborated Design (10 infos)
  - General Messages (10 infos)
    - [Synth 8-6157] synthesizing module 'SPI_top_module' [SPI_top_module.v:1] (2 more like this)
    - [Synth 8-5534] Detected attribute (* fsm_encoding = "one_hot" *) [SPI_slave.v:30]
    - [Synth 8-6155] done synthesizing module 'SPI_slave' (1#1) [SPI_slave.v:1] (2 more like this)
    - [Project 1-570] Preparing netlist for logic optimization
    - [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
    - [Project 1-111] Unisim Transformation Summary:
      No Unisim elements were transformed.

Schematic



SYNTHESIS “ONE HOT ENCODING”

Messages

- Synthesized Design (9 infos)
  - General Messages (9 infos)
    - [Netlist 29-17] Analyzing 5 Unisim elements for replacement
    - [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
    - [Project 1-479] Netlist was created with Vivado 2018.2
    - [Project 1-570] Preparing netlist for logic optimization
    - [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).
    - [Project 1-111] Unisim Transformation Summary:
      No Unisim elements were transformed.
    - [Timing 38-35] Done setting XDC timing constraints.
    - [Timing 38-91] UpdateTimingParams: Speed grade: -1L, Delay Type: min_max.
    - [Timing 38-191] Multithreading enabled for timing update using a maximum of 2 CPUs

Utilization report

Name	Slice LUTs (20800)	Slice Registers (41600)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
SPI_top_module	29	58	0.5	5	1
RAM_inst (RAM)	5	17	0.5	0	0
SPI_slave_inst (SPI_sl...	24	41	0	0	0

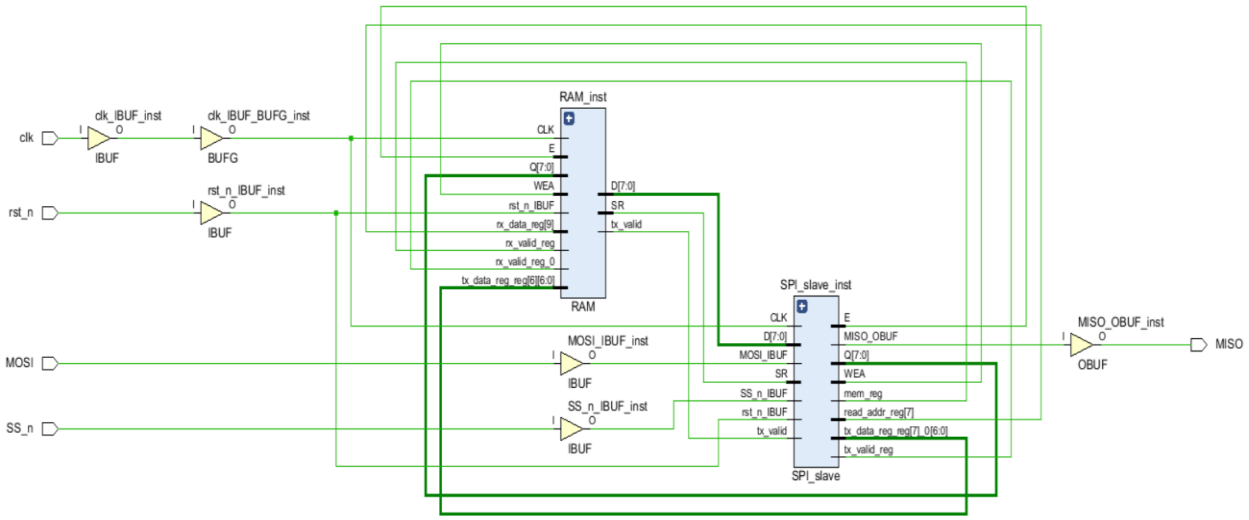
timing report

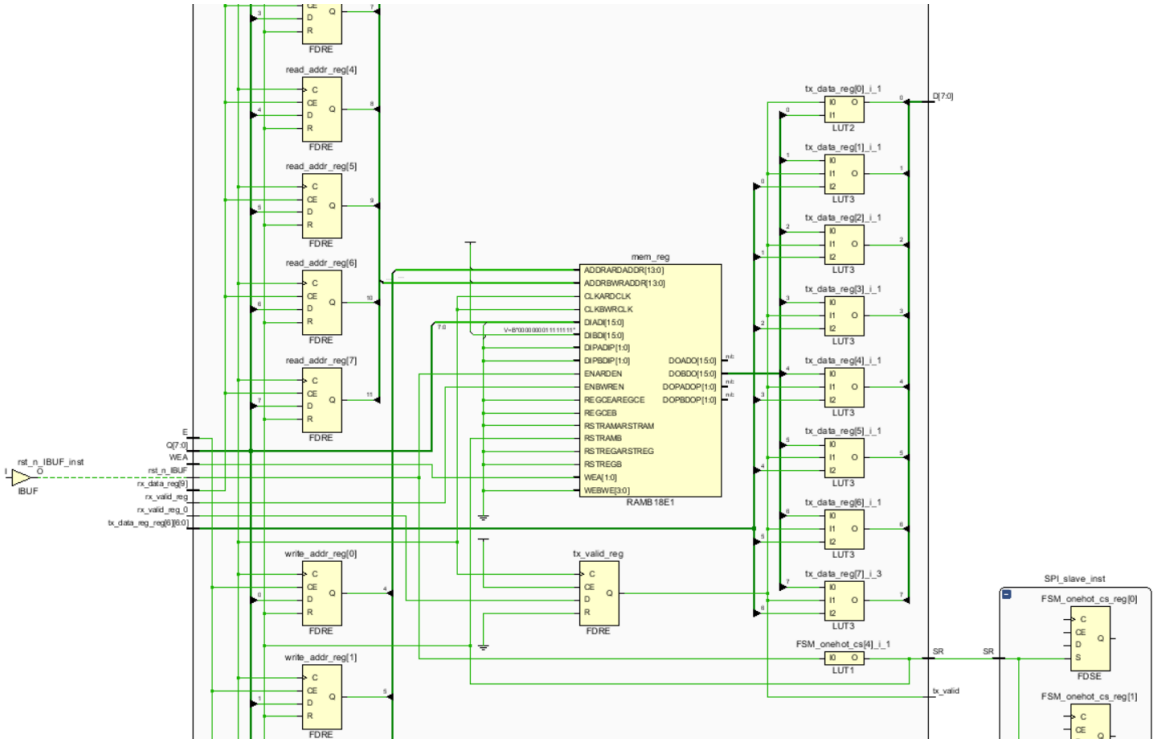
Design Timing Summary

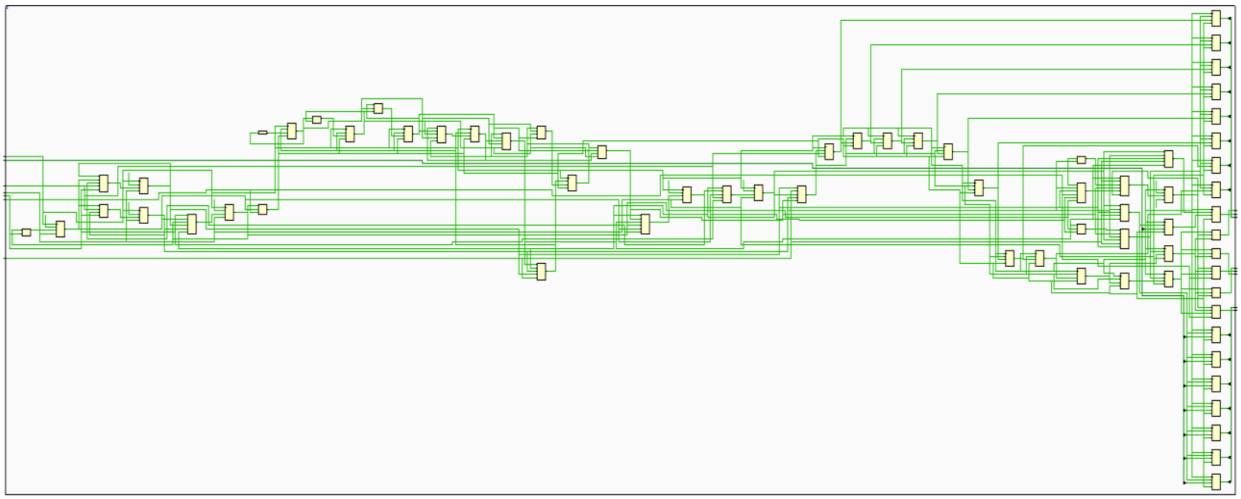
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	6.343 ns	Worst Hold Slack (WHS):	0.134 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	158	Total Number of Endpoints:	158	Total Number of Endpoints:	61

All user specified timing constraints are met.

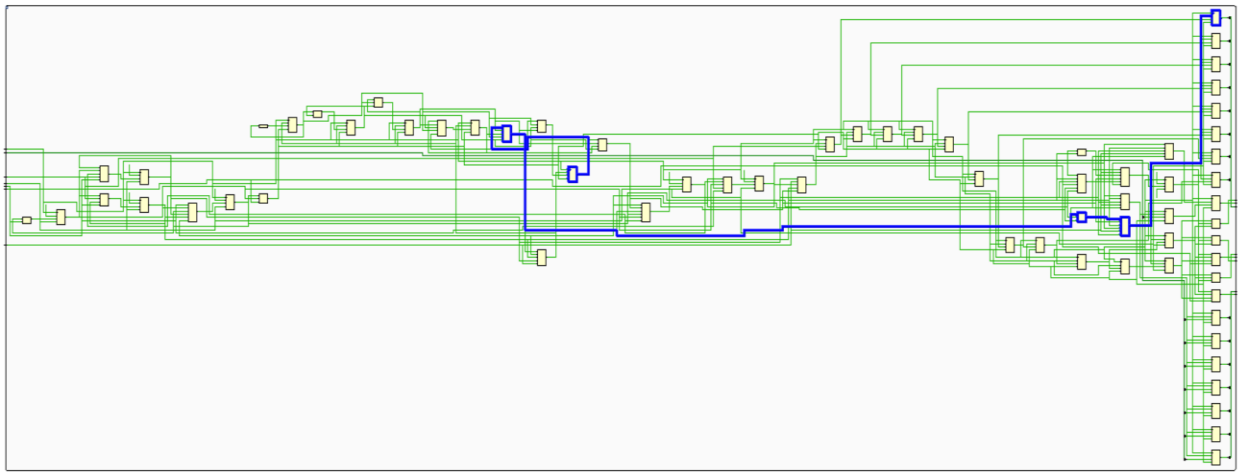
Schematic

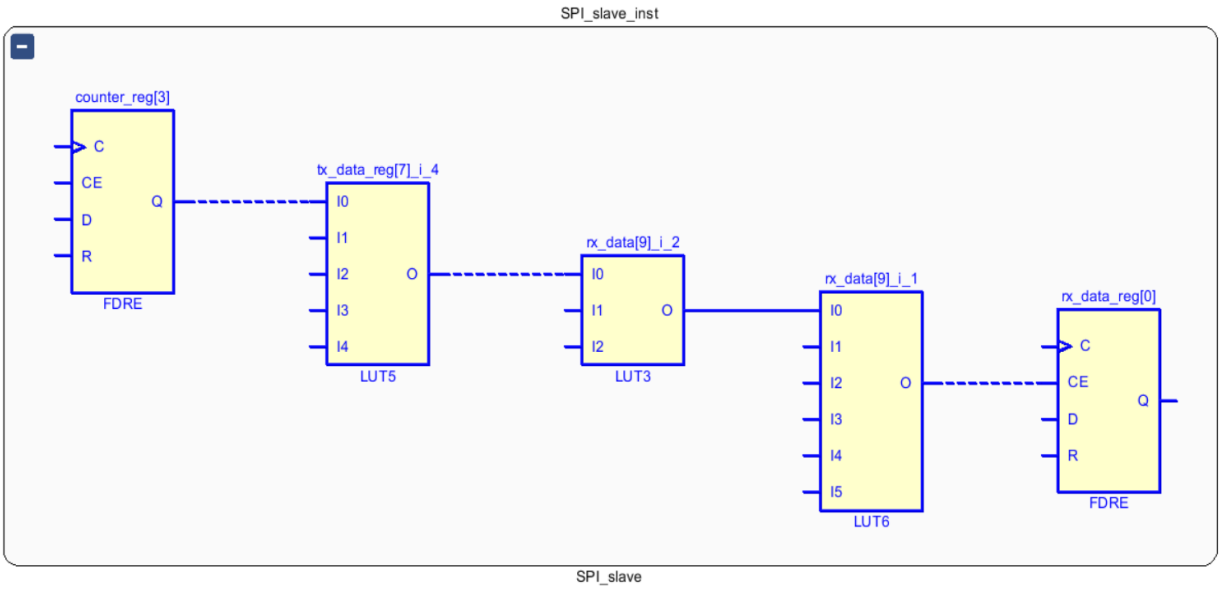



FDRE
read_addr_reg[4]
read_addr_reg[5]
read_addr_reg[6]
read_addr_reg[7]
write_addr_reg[0]
write_addr_reg[1]
mem_reg
RAMB18E1
ADDRARDADDR[13:0]
ADDRBWRADDR[13:0]
CLKARDCLK
CLKBWRCLK
DIADI[15:0]
DIBDI[15:0]
DIPADIP[1:0]
DIPBDIP[1:0]
ENARDEN
ENBWREN
REGCEAREGCE
REGCEB
RSTRAMARSTRAM
RSTRAMB
RSTREGARSTREG
RSTREGB
WEA[1:0]
WEBWE[3:0]
DOADO[15:0]
DOBDO[15:0]
DOPADOP[1:0]
DOPBDOP[1:0]
tx_valid_reg
tx_data_reg[0]_i_1
tx_data_reg[1]_i_1
tx_data_reg[2]_i_1
tx_data_reg[3]_i_1
tx_data_reg[4]_i_1
tx_data_reg[5]_i_1
tx_data_reg[6]_i_1
tx_data_reg[7]_i_3
LUT2
LUT3
FSM_onehot_cs[4]_i_1
LUT1
rst_n_IBUF_inst
IBUF
rst_n_IBUF
rx_data_reg[0]
rx_valid_reg
rx_valid_reg_0
tx_data_reg_reg[6][0:0]
E
WEA
Q[7:0]
SR
tx_valid
SPI_slave_inst
FSM_onehot_cs_reg[0]
FSM_onehot_cs_reg[1]
FDSE



Critical path





Synthesis report

```
-----------------------------------------------------------------------------------------------
               State |                     New Encoding |               Previous Encoding
-----------------------------------------------------------------------------------------------
                IDLE |                            00001 |                              000
              CHK_CMD |                            10000 |                              001
            READ_DATA |                            00010 |                              110
             READ_ADD |                            00100 |                              010
                WRITE |                            01000 |                              011
-----------------------------------------------------------------------------------------------
INFO: [Synth 8-3354] encoded FSM with state register 'cs_reg' using encoding 'one-hot' in module 'SPI_slave'
---------------------------------------------------------------------------------
Finished RTL Optimization Phase 2 : Time (s): cpu = 00:00:24 ; elapsed = 00:00:26 . Memory (MB): peak = 765.035 ; gain = 507.969
---------------------------------------------------------------------------------
```

IMPLEMENTATION “ONE HOT ENCODING”

Messages

- Implemented Design (9 infos)
  - General Messages (9 infos)
    - [Netlist 29-17] Analyzing 5 Unisim elements for replacement
    - [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
    - [Project 1-479] Netlist was created with Vivado 2018.2
    - [Project 1-570] Preparing netlist for logic optimization
    - [Timing 38-478] Restoring timing data from binary archive.
    - [Timing 38-479] Binary timing data restore complete.
    - [Project 1-856] Restoring constraints from binary archive.
    - [Project 1-853] Binary constraint restore complete.
    - [Project 1-111] Unisim Transformation Summary:
      No Unisim elements were transformed.

Utilization report

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)
N SPI_top_module	30	58	17	30	15	0.5	5	1
I RAM_inst (RAM)	6	17	6	6	0	0.5	0	0
I SPI_slave_inst (SPI_sl...	24	41	15	24	10	0	0	0

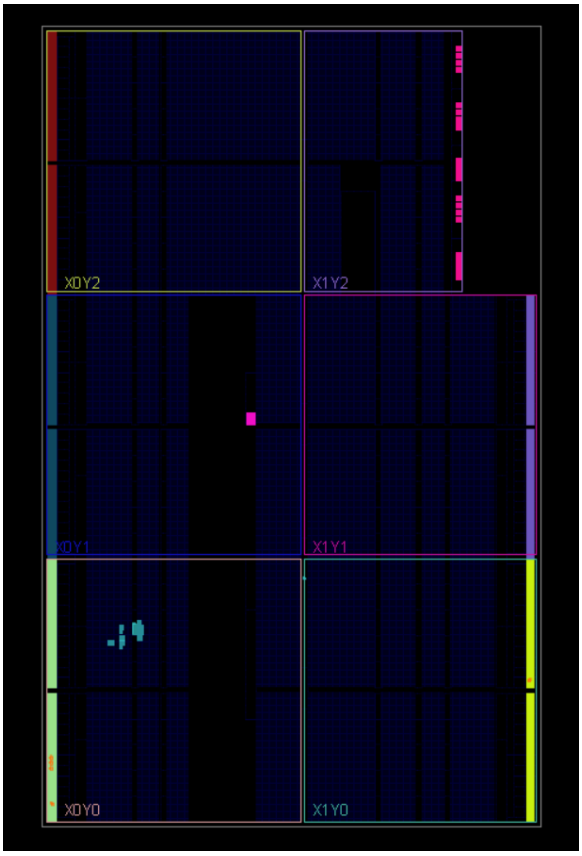
timing report

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.867 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
Total Number of Endpoints:	159	Total Number of Endpoints:	159	Total Number of Endpoints:	61

All user specified timing constraints are met.

Device



FSM Encoding Comparison Based on Timing Performance

Encoding Style	Setup Time Slack (WNS)	Hold Time Slack (WHS)
Gray	6.382 ns	0.044 ns
One-Hot	5.867 ns	0.044 ns
Sequential	6.139 ns	0.044 ns

- **Gray Encoding** provides the highest **Worst Negative Slack (WNS)** of **6.382 ns**, which implies it can run at the **highest clock frequency** compared to the others.