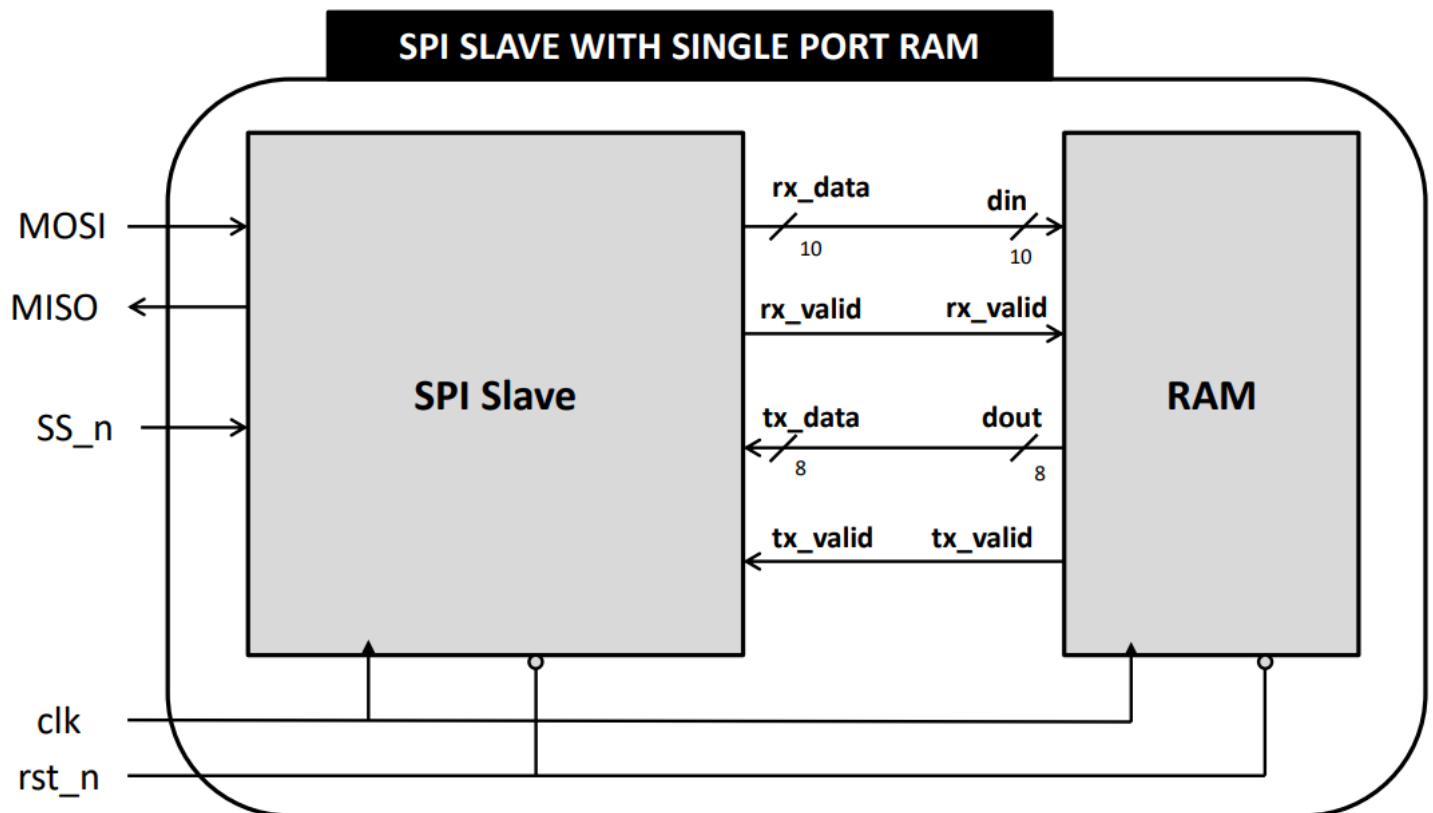


FINAL PROJECT

SPI SLAVE Interface

By: Sarah Abdelatty Ibrahim

Under Supervision of Eng. Kareem Waseem



Single Port Async. RAM

The single port asynchronous RAM module implements a memory block with a single data port.

It has the following parameters:

- MEM_DEPTH (default: 256): Depth of the memory.
- ADDR_SIZE (default: 8): Size of the memory address.

Ports:

Name	Type	Size	Description
clk	Input	1 bit	Clock signal
rst_n	Input	1 bit	Active low reset signal
din	Input	10 bit	Data input
rx_valid	Input	1 bit	If HIGH, accepts din[7:0] to save the write/read address internally or writes a memory word depending on the most significant 2 bits din[9:8]
dout	Output	8 bit	Data output
tx_valid	Output	1 bit	Whenever the command is a memory read, tx_valid should be HIGH

The most significant bits of din (din[9:8]) determine the operation to be performed:

Port	din[9:8]	Command	Description
din	00	Write	Holds din[7:0] internally as a write address
din	01	Write	Writes din[7:0] to the memory with the write address held previously
din	10	Read	Holds din[7:0] internally as a read address
din	11	Read	Reads the memory with the read address held previously. tx_valid should be HIGH, and dout holds the word read from the memory. din[7:0] is ignored

RTL Code Snippet:

```
module SYN_RAM (din, clk, rst_n, rx_valid, dout, tx_valid);
// Define Parameter
parameter MEM_DEPTH = 256;
parameter ADDR_SIZE = 8;

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

// Define output ports
output reg [7:0] dout;
output reg tx_valid;

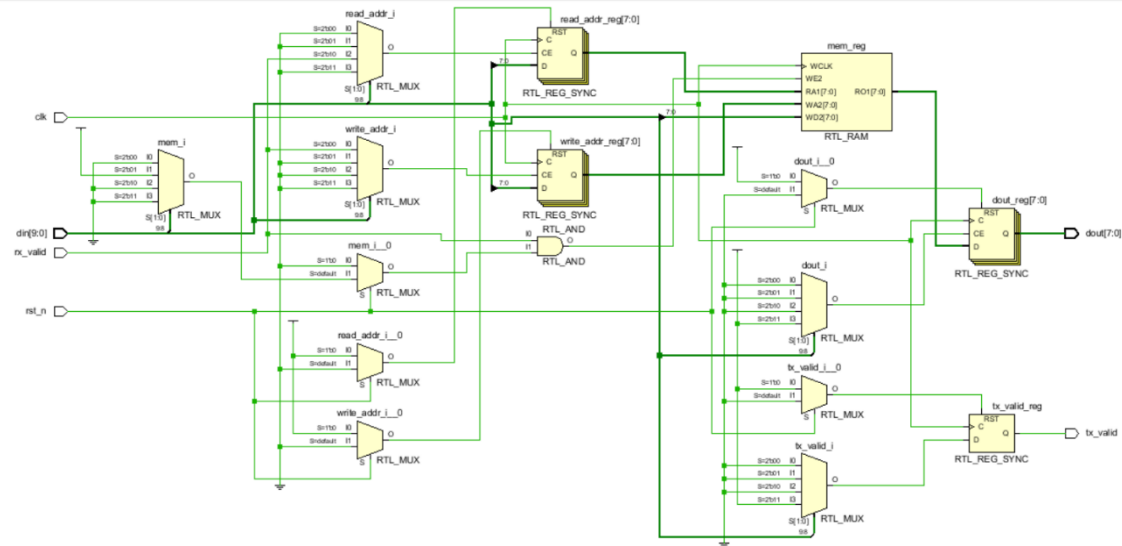
// Define Memory with width size = 8
reg [7:0] mem [MEM_DEPTH-1:0];

// Define addresses
reg [7:0] write_addr, read_addr;

// Define Data input
reg [7:0] data;

always @(posedge clk) begin
    if (~rst_n) begin
        dout <= 0;
        tx_valid <= 0;
        write_addr <= 0;
        read_addr <= 0;
    end
    else begin
        case (din[9:8])
            2'b00: begin
                tx_valid <= 0;
                if (rx_valid)
                    write_addr <= din[7:0];
            end
            2'b01: begin
                tx_valid <= 0;
                if (rx_valid)
                    mem[write_addr] <= din[7:0];
            end
            2'b10: begin
                tx_valid <= 0;
                if (rx_valid)
                    read_addr <= din[7:0];
            end
            2'b11: begin
                dout <= mem[read_addr];
                tx_valid <= 1;
            end
        endcase
    end
end
endmodule
```

Elaboration: Schematic:



Messages Tab:

Vivado Commands (4 Infos)

- General Messages** (4 Infos)
 - [ProjectBase 1-489] The host OS only allows 260 characters in a normal path. The project is stored in a path with more than 80 characters. If you experience issues with IP, Block Designs, or files not being found, please consider moving the project to a location with a shorter path. Alternately consider using the OS subst command to map part of the path to a drive letter. Current project path is 'G:/Sarah/Digital_Course/digital_course_models/ram/project/ram/Final_project_RAM'
 - [IP_Flow 19-234] Refreshing IP repositories
 - [IP_Flow 19-1704] No user IP repositories specified
 - [IP_Flow 19-2313] Loaded Vivado IP repository 'G:/vado3/Vivado/Vivado/2018.2/data/ip'
- Elaborated Design** (2 Infos)
 - General Messages** (2 Infos)
 - [Project 1-570] Preparing netlist for logic optimization
 - [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).

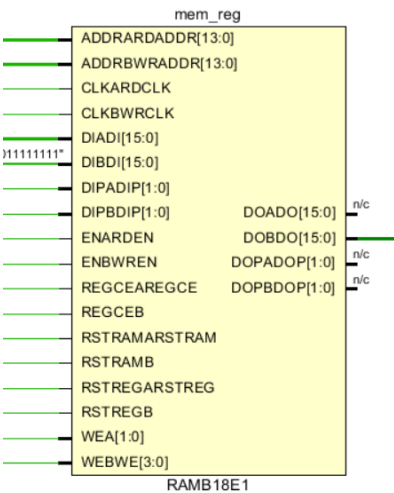
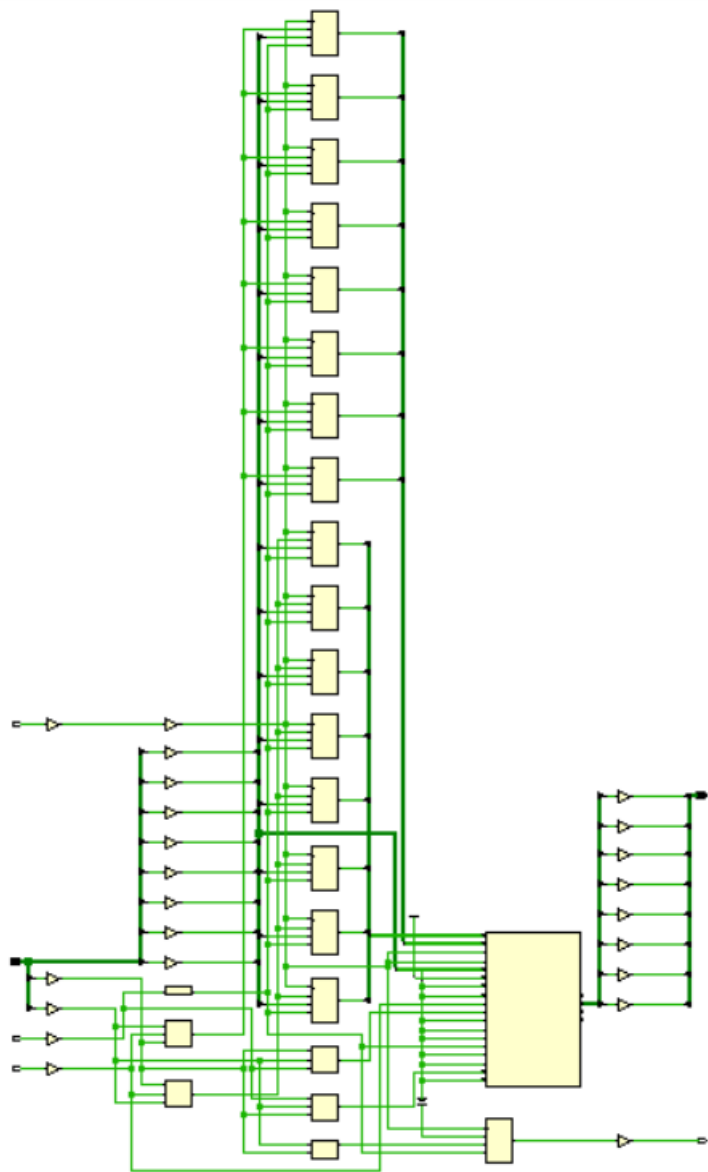
Synthesis: Messages Tab:

Warning (1) Info (26) Status (14) Show All

Synthesis (1 warning, 16 Infos)

- [Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a35f'
- [Synth 9-6157] synthesizing module 'SYN_RAM' [ASYN_RAM.v]
- [Synth 9-6156] done synthesizing module 'SYN_RAM' [1#1] [ASYN_RAM.v]
- [Device 21-403] Loading part xc7a35fcpq236-1L
- [Synth 9-4460] The timing for the instance L2mem_reg (implemented as a block RAM) might be sub-optimal as no optional output register could be merged into the block ram. Providing additional output register may help in improving timing. [1 more like this]
- [Project 1-571] Translating synthesized netlist
- [Netlist 29-17] Analyzing 14 Unisim elements for replacement
- [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
- [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.
- [Common 17-63] Releasing license: Synthesis
- [Constraints 18-5210] No constraint will be written out.
- [Common 17-1381] The checkpoint 'G:/Sarah/Digital_Course/digital_course_models/ram/project/ram/Final_project_RAM/ram/Final_project_RAM/synth_1/SYN_RAM.dcp' has been generated.
- [runtd-4] Executing: report_utilization -file SYN_RAM_utilization_synth.rpt -pb SYN_RAM_utilization_synth.pb
- [Common 17-206] Exiting Vivado at Mon Aug 5 00:49:32 2024.
- Synthesized Design** (6 Infos)
 - General Messages** (6 Infos)
 - [Netlist 29-17] Analyzing 14 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.

Schematic:



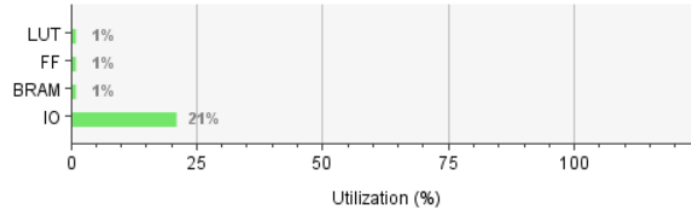
Timing Report Summary:

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): inf	Worst Hold Slack (WHS): inf	Worst Pulse Width Slack (WPWS): NA
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): NA
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: NA
Total Number of Endpoints: 88	Total Number of Endpoints: 88	Total Number of Endpoints: NA

There are no user specified timing constraints.

Utilization Report:

Resource	Utilization	Available	Utilization %
LUT	6	20800	0.03
FF	17	41600	0.04
BRAM	0.50	50	1.00
IO	22	106	20.75



Testbench Code Snippet:

```
module SYN_RAM_tb ();
    parameter MEM_DEPTH = 256;
    parameter ADDR_SIZE = 8;
    reg [9:0] din_tb;
    reg clk_tb, rst_n_tb, rx_valid_tb;
    wire [7:0] dout_dut;
    wire tx_valid_dut;
    integer i;

    SYN_RAM DUT(
        .din(din_tb), .clk(clk_tb),
        .rst_n(rst_n_tb), .rx_valid(rx_valid_tb),
        .dout(dout_dut), .tx_valid(tx_valid_dut)
    );

    initial begin
        clk_tb = 1;
        forever
            #1 clk_tb = ~clk_tb;
    end

    initial begin
        // Initialize all control signals
        din_tb = 0;
        rst_n_tb = 1;
        rx_valid_tb = 0;

        // Initialize the memory
        for (i=0; i<256; i=i+1)
            DUT.mem[i] = 0;

        // Test reset
        rst_n_tb = 0;
        @(negedge clk_tb);

        // Release reset and activate write (first write address)
        rst_n_tb = 1;
        rx_valid_tb = 1;
        din_tb[9:8] = 2'b00;
        din_tb[7:0] = 8'd181; // address of write = 181
        @(negedge clk_tb);
        // Activate write (then write data)
        din_tb[9:8] = 2'b01;
        din_tb[7:0] = 8'd230; // data input = 230
        @(negedge clk_tb);

        // Activate read (first read address)
        din_tb[9:8] = 2'b10;
        din_tb[7:0] = 8'd233; // addresses that i want to read = 233
        @(negedge clk_tb);
        // Activate read (first read address)
        din_tb[9:8] = 2'b11; // read zero
        @(negedge clk_tb);
    end
endmodule
```

```
// Activate read (read the address that have been written 181)
din_tb[9:8] = 2'b10;
din_tb[7:0] = 8'd181; // addresses that i want to read = 181
@(negedge clk_tb);
din_tb[9:8] = 2'b11; // read 230
@(negedge clk_tb);

// Activate write
din_tb[9:8] = 2'b00;
din_tb[7:0] = 8'd250; // address of write = 250
repeat (5) @(negedge clk_tb);
din_tb[9:8] = 2'b01;
din_tb[7:0] = 8'd156; // data input = 156
@(negedge clk_tb);

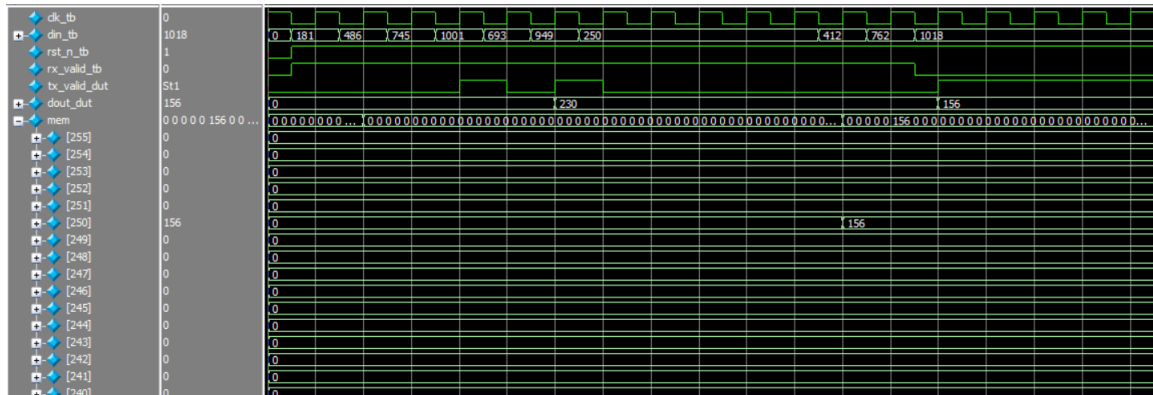
// Activate read (read the address that have been written 250)
din_tb[9:8] = 2'b10;
din_tb[7:0] = 8'd250; // addresses that i want to read = 250
@(negedge clk_tb);
rx_valid_tb = 0; // Test that the fourth condition doesn't depend on rx_valid
din_tb[9:8] = 2'b11; // read 156
repeat (5) @(negedge clk_tb);
$stop;

end

initial begin
    $monitor("din = %d, rx_valid = %b, dout = %d, tx_valid = %b",
            din_tb, rx_valid_tb, dout_dut, tx_valid_dut);
end

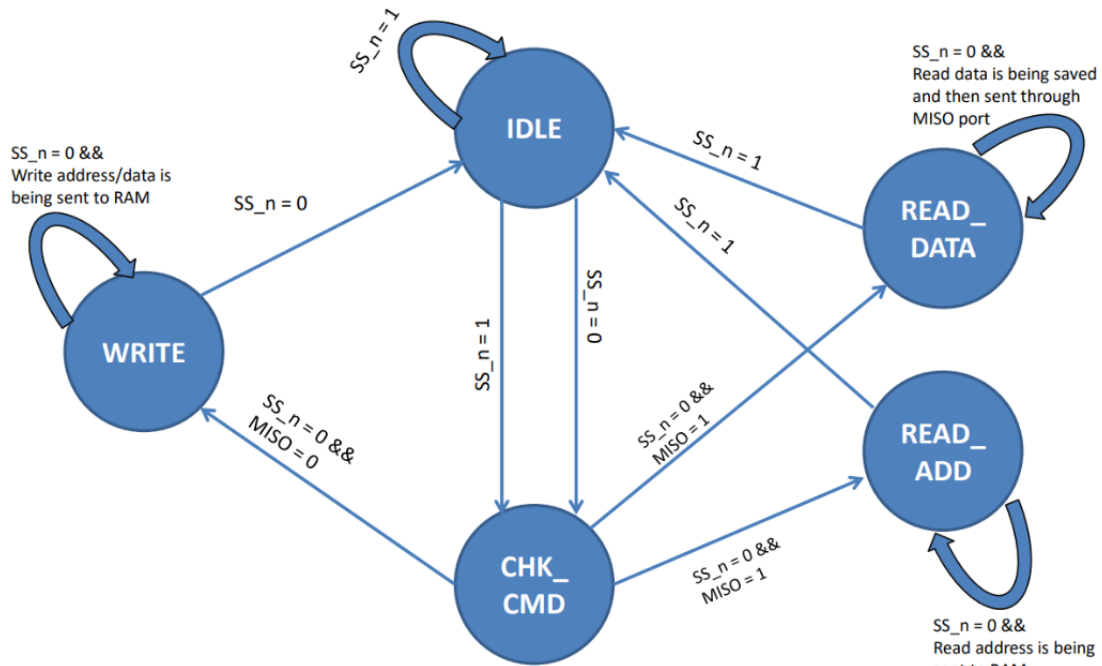
endmodule
```

Simulation Waveform Snippet:



SPI SLAVE

SPI SLAVE State Transition Diagram



Introduction to SPI SLAVE Module:

The SPI slave module effectively manages data transfer between a master and a slave device. It uses state machines to control the flow of operations, ensuring proper read and write transactions based on the SPI protocol. The counters and flags aid in converting data formats and ensuring synchronization between the master and slave devices.

RTL Code Snippets:

```
module FSM_SPI (MOSI, MISO, ss_n, clk, rst_n, rx_data, rx_valid, tx_data, tx_valid);
// Parameters for states
parameter IDEL = 3'b000;
parameter CHK_CMD = 3'b001;
parameter WRITE = 3'b010;
parameter READ_DATA = 3'b011;
parameter READ_ADD = 3'b100;

// Input ports
input MOSI, ss_n, tx_valid, clk, rst_n;
input [7:0] tx_data;

// Output ports
output reg MISO, rx_valid;
output reg [9:0] rx_data;

// Define current and next states
(*fsm_encoding="sequential"*)
reg [2:0] cs, ns;
reg rd_sig; // Define internal signal to select read(address or data)
reg [3:0] counter_par, counter_ser; // Counters to convert from series to parallel or vice versa
// state memory block
always @(posedge clk) begin
    if (~rst_n)
        cs <= IDEL;
    else
        cs <= ns;
    end

// next state logic block
always @(*) begin
    case(cs)
        IDEL:
            if (~ss_n)
                ns = CHK_CMD;
            else
                ns = IDEL;
        CHK_CMD:
            if (ss_n)
                ns = IDEL;
            else if (~ss_n) begin
                if (~MOSI)
                    ns = WRITE;
                else if (MOSI) begin
                    if (rd_sig)
                        ns = READ_DATA;
                    else if (~rd_sig)
                        ns = READ_ADD;
                end
            end
        WRITE:
            if (ss_n)
                ns = IDEL;
            else
                ns = WRITE;
        READ_DATA:
            if (ss_n)
                ns = IDEL;
            else
                ns = READ_DATA;
        READ_ADD:
            if (ss_n)
                ns = IDEL;
            else
                ns = READ_ADD;
    endcase
end

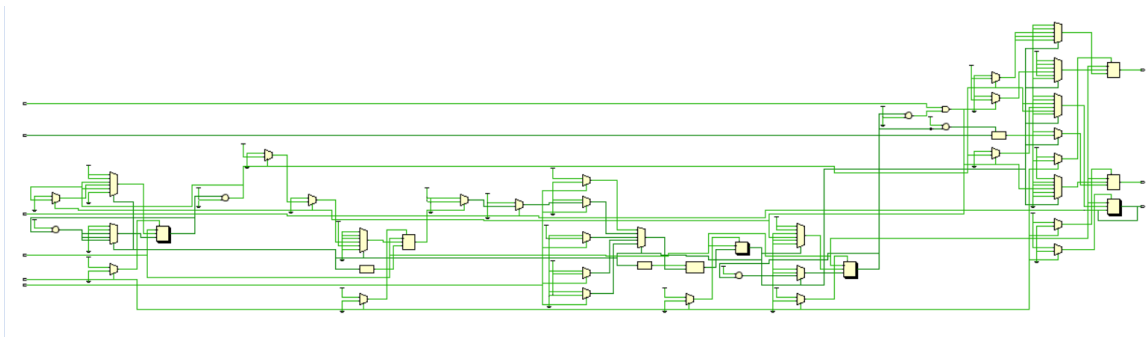
/// output logic block
always @(posedge clk) begin
    if (~rst_n) begin
        rx_data <= 0;
        rx_valid <= 0;
        MISO <= 0;
        rd_sig <= 0;
        counter_ser <= 3'b000;
        counter_par <= 3'b000;
    end
    else begin
        case(cs)
            IDEL: begin
                rx_valid <= 0;
                MISO <= 0;
                counter_ser <= 3'b000;
                counter_par <= 3'b000;
            end
            CHK_CMD: begin
                rx_valid <= 0;
                counter_ser <= 3'b000;
                counter_par <= 3'b000;
            end
            WRITE: begin
                // Convert Data series to parallel(write data or address)
                if (counter_ser < 10) begin
                    rx_valid <= 0;
                    rx_data <= {rx_data[8:0], MOSI};
                    counter_ser <= counter_ser + 1;
                end
                else
                    rx_valid <= 1;
            end
            READ_DATA: begin
                // Convert Data parallel to series(read data)
                if (tx_valid && counter_par < 8) begin
                    MISO <= tx_data[7-counter_par];
                    counter_par <= counter_par + 1;
                end
            end
        endcase
    end
end
```

```

else begin
    // Convert Data series to parallel(read address that I want its data)
    if (counter_ser<10) begin
        rx_valid <= 0;
        rx_data <= {rx_data[8:0], MOSI};
        counter_ser <= counter_ser + 1;
    end
    else begin
        rx_valid <= 1;
        rd_sig <= 0;
    end
end
end
endcase
end
endmodule

```

Elaboration:
Schematic:



Messages Tab:

- Vivado Commands (4 Infos)
 - General Messages (4 Infos)
 - [ProjectBase 1-489] The host OS only allows 260 characters in a normal path. The project is stored in a path with more than 80 characters. If you experience issues with IP, Block Designs, or files not being found, please consider moving the project to a location with a shorter path. Alternately consider using the OS subst command to map part of the path to a drive letter. Current project path is 'G:/Sarah/Digital_Course/digital_course_models\sim\final project\SPI\Final_project_SPI'
 - [IP_Flow 19-234] Refreshing IP repositories
 - [IP_Flow 19-1704] No user IP repositories specified
 - [IP_Flow 19-2313] Loaded Vivado IP repository 'G:/vado3/Vivado/Vivado/2018.2\data\ip'
- Elaborated Design (2 Infos)
 - General Messages (2 Infos)
 - [Project 1-570] Preparing netlist for logic optimization
 - [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).

Synthesis:
Encoding:

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

Messages Tab:

Warning (2)
 Info (34)
 Status (14)
 Show All

Synthesis (2 warnings, 24 infos)

- [Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a35t'

[Synth 8-6157] synthesizing module FSM_SPT [FSM_SPT.v1]

[Synth 8-5534] Deleted attribute (" fsm_encoding = "sequential") [FSM_SPT.v19]
- [Synth 8-155] case statement is not full and has no default [FSM_SPT.v32] (1 more like this)

[Synth 8-6155] done synthesizing module FSM_SPT (181) [FSM_SPT.v1]

[Device 21-403] Loading part xc7a35t-cpg236-1L
- [Synth 8-802] inferred FSM for state register 'cs_reg' in module FSM_SPT
- [Synth 8-5544] ROM "ns" won't be mapped to Block RAM because address size (3) smaller than threshold (5) (4 more like this)

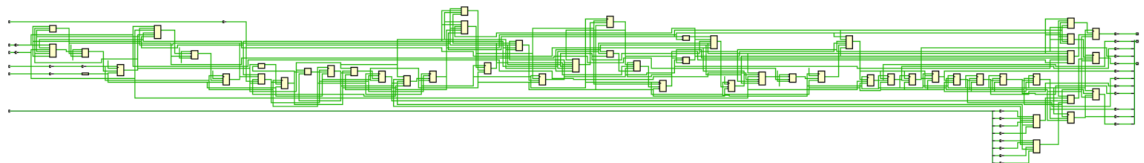
[Synth 8-3354] encoded FSM with state register 'cs_reg' using encoding 'sequential' in module FSM_SPT

[Synth 8-327] inferring latch for variable 'FSM_sequential_ns_reg' [FSM_SPT.v35]
- [Project 1-671] Translating synthesized netlist
- [Netlist 29-17] Analyzing 16 Unisim elements for replacement
- [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
- [Project 1-570] Preparing netlist for logic optimization
- [Opt 31-138] Pushed 0 inverters(s) to 0 load pin(s).
- [Project 1-111] Unisim Transformation Summary:
4 out of 3 instances were transformed.
LD => LDCOE: 3 instances
- [Common 17-833] Releasing license: Synthesis
- Constraints 18-5210: No constraint will be written out.
- [Common 17-1381] The checkpoint "Q:\Sarah\Digital_Course\digital_course_models\final project\SPi\Final_project_SPi\Final_project_SPi.runs\synth_1\FSM_SPT.dcp" has been generated.
- [Junit4-4] Executing :report_utilization -file FSM_SPT_utilization_synth.rpt -pb FSM_SPT_utilization_synth.pb
- [Common 17-206] Exiting Vivado at Mon Aug 5 01:02:08 2024...

Synthesized Design (6 infos)

- [Netlist 29-17] Analyzing 13 Unisim elements for replacement
-

Schematic:



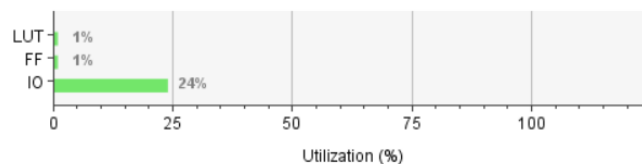
Timing Report Summary:

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): inf	Worst Hold Slack (WHS): inf	Worst Pulse Width Slack (WPWS): NA
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): NA
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: NA
Total Number of Endpoints: 83	Total Number of Endpoints: 83	Total Number of Endpoints: NA

There are no user specified timing constraints.

Utilization Report:

Resource	Utilization	Available	Utilization %
LUT	25	20800	0.12
FF	27	41600	0.06
IO	25	106	23.58



Testbench Code Snippets:

```
module FSM_SPI_tb ();
reg MOSI_tb, ss_n_tb, tx_valid_tb, clk_tb, rst_n_tb;
reg [7:0] tx_data_tb;
wire MISO_dut, rx_valid_dut;
wire [9:0] rx_data_dut;

FSM_SPI DUT(
.MOSI(MOSI_tb), .MISO(MISO_dut), .ss_n(ss_n_tb),
.clk(clk_tb), .rst_n(rst_n_tb), .rx_data(rx_data_dut),
.rx_valid(rx_valid_dut), .tx_data(tx_data_tb), .tx_valid(tx_valid_tb)
);

initial begin
    clk_tb = 1;
    forever
        #1 clk_tb = ~clk_tb;
end

initial begin
    // Initialize all control signals
    MOSI_tb = 0;
    ss_n_tb = 1;
    tx_data_tb = 0;
    rst_n_tb = 1;
    tx_valid_tb = 0;

    // Test Reset
    rst_n_tb = 0; // In the IDLE state
    repeat(5) @(negedge clk_tb);

    // Release reset and move to CHK_CMD state
    rst_n_tb = 1;
    ss_n_tb = 0;
    repeat(5) @(negedge clk_tb);

    // Check the WRITE state (Write address)
    MOSI_tb = 0;
    repeat(2) @(negedge clk_tb);
    repeat (8) begin
        MOSI_tb = $random;
        @(negedge clk_tb);
    end

    // Check the WRITE state (Write data)
    MOSI_tb = 0;
    @(negedge clk_tb);
    MOSI_tb = 1;
    @(negedge clk_tb);
    repeat (8) begin
        MOSI_tb = $random;
        @(negedge clk_tb);
    end
end
```

```
    // Back to IDLE state
    ss_n_tb = 1;
    @(negedge clk_tb);

    // Move to CHK_CMD state
    ss_n_tb = 0;
    @(negedge clk_tb);

    // Check the READ_ADD state (Read address)
    MOSI_tb = 1;
    @(negedge clk_tb);
    MOSI_tb = 0;
    @(negedge clk_tb);
    repeat (8) begin
        MOSI_tb = $random;
        @(negedge clk_tb);
    end

    // Back to IDLE state
    ss_n_tb = 1;
    @(negedge clk_tb);

    // Move to CHK_CMD state
    ss_n_tb = 0;
    @(negedge clk_tb);

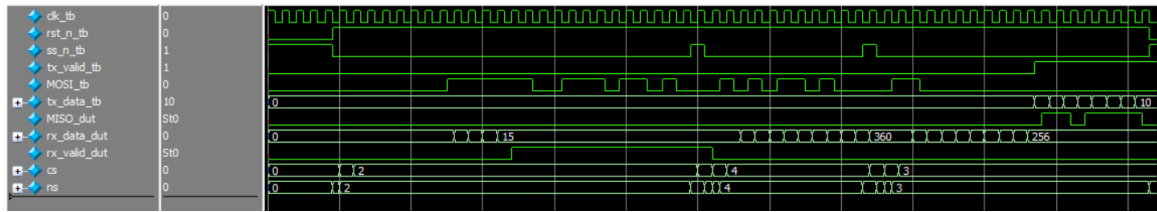
    // Check the READ_DATA state (Read data)
    MOSI_tb = 1;
    repeat(2) @(negedge clk_tb);
    repeat (8) begin
        MOSI_tb = 0;
        @(negedge clk_tb);
    end
    tx_valid_tb = 1;
    repeat (8) begin
        tx_data_tb = $random;
        @(negedge clk_tb);
    end

    // Reset all signals
    ss_n_tb = 1;
    rst_n_tb = 0;
    @(negedge clk_tb);
    $stop;
end

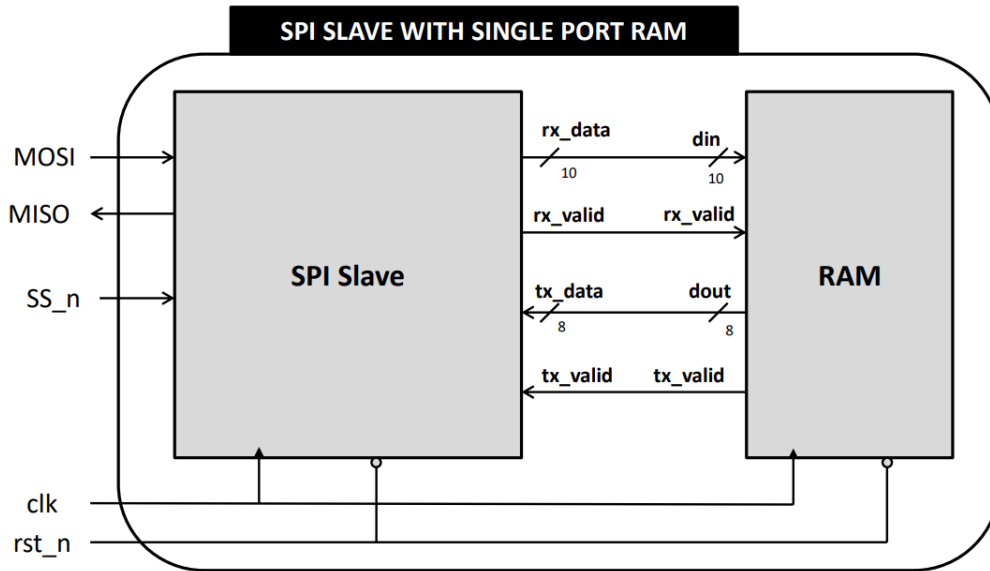
initial begin
    $monitor("MOSI = %b, MISO = %b, ss_n = %b, rx_data = %b, rx_valid = %b, tx_data = %b, tx_valid = %b",
    MOSI_tb, MISO_dut, ss_n_tb, rx_data_dut, rx_valid_dut, tx_data_tb, tx_valid_tb);
end

endmodule
```

Simulation Waveform Snippet:



SPI Wrapper



Introduction to Wrapper:

The Wrapper module integrates the Slave and RAM modules to enable SPI communication with a memory block. The Slave manages the SPI protocol and data transfer with the master device, while the RAM module handles data storage and retrieval. This setup allows the master device to read from and write to the memory block via the SPI interface, with the Slave ensuring proper data transfer and the RAM maintaining the data integrity.

Main Wires between RAM & Slave

1. `rx_data` in the SPI slave module is connected to the `din` port in the RAM module.
2. `rx_valid` in the SPI slave module is connected to `rx_valid` in the RAM module.
3. `dout` in the RAM module is connected to `tx_data` in the SPI slave module.
4. `tx_valid` in the RAM module is connected to `tx_valid` in the SPI slave module.

RTL Code Snippet:

```
module WRAPPER (MOSI, MISO, ss_n, clk, rst_n);
// Input ports
input MOSI, clk, rst_n, ss_n;

// Output ports
output MISO;

// Internal signals
wire [9:0] rx_data;
wire rx_valid, tx_valid;
wire [7:0] tx_data;

// Instantiate Master Slave SPI
FSM_SPI SLAVE_SPI(
    .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)
);

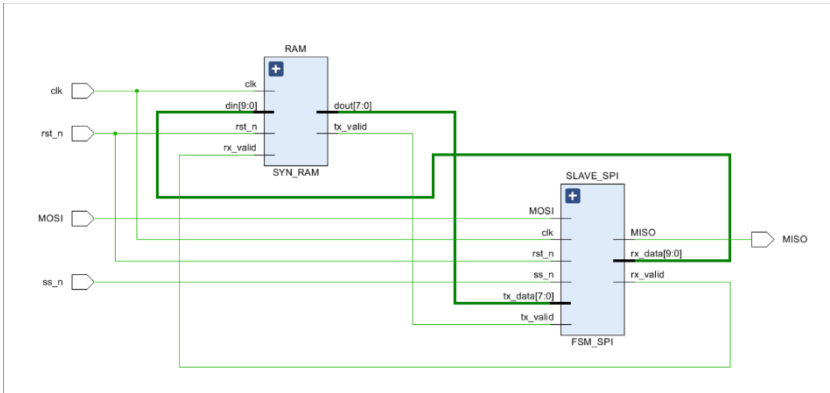
// Instantiate Memory "RAM"
SYN_RAM #(
    .MEM_DEPTH(256), .ADDR_SIZE(8)
) RAM(
    .din(rx_data), .clk(clk),
    .rst_n(rst_n), .rx_valid(rx_valid),
    .dout(tx_data), .tx_valid(tx_valid)
);

endmodule
```

Gray Encoding

1. Elaboration:

Schematic:



Messages Tab:

Vivado Commands (3 infos)

General Messages (3 infos)

- [IP_Flow 19-234] Refreshing IP repositories
- [IP_Flow 19-1704] No user IP repositories specified
- [IP_Flow 19-2313] Loaded Vivado IP repository 'G:/vivado3/Vivado/Vivado/2018.2/data/ip'.

Elaborated Design (3 infos)

General Messages (3 infos)

- [Project 1-570] Preparing netlist for logic optimization
- [Project 1-236] Implementation specific constraints were found while reading constraint file [G:/Final_project_dig/Constraints_final_project.xdc]. These constraints will be ignored for synthesis but will be used in implementation. Impacted constraints are listed in the file [X:/WRAPPER_propimpl.xdc]. Resolution: To avoid this warning, move constraints listed in [X:/WRAPPER_propimpl.xdc] to another XDC file and exclude this new file from synthesis with the used_in_synthesis property (File Properties dialog in GUI) and re-run elaboration/synthesis.
- [Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).

2. Synthesis:

Encoding:

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

[illegible][illegible]

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
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 100	Total Number of Endpoints: 100	Total Number of Endpoints: 44
All user specified timing constraints are met.		

Resource	Utilization	Available	Utilization %
LUT	29	20800	0.14
FF	44	41600	0.11
BRAM	0.50	50	1.00
IO	5	106	4.72

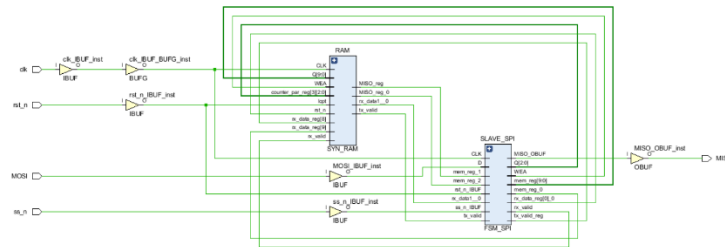


3. Implementation:

Messages Tab:

- ▼ 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.

Schematic:



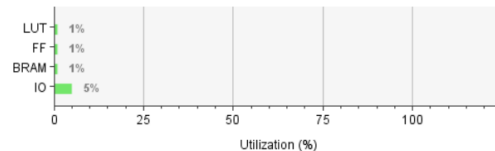
Timing Report Summary:

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.446 ns	Worst Hold Slack (WHS): 0.042 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: 100	Total Number of Endpoints: 100	Total Number of Endpoints: 44

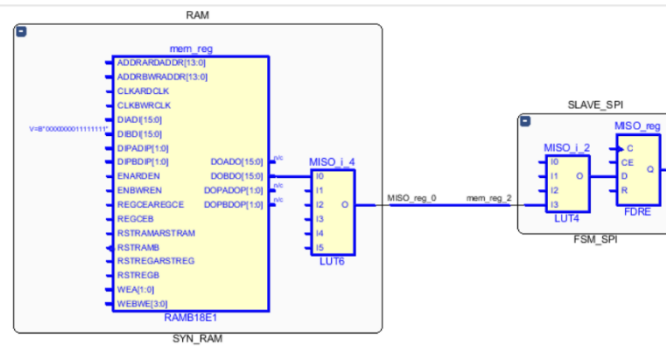
All user specified timing constraints are met.

Utilization Report:

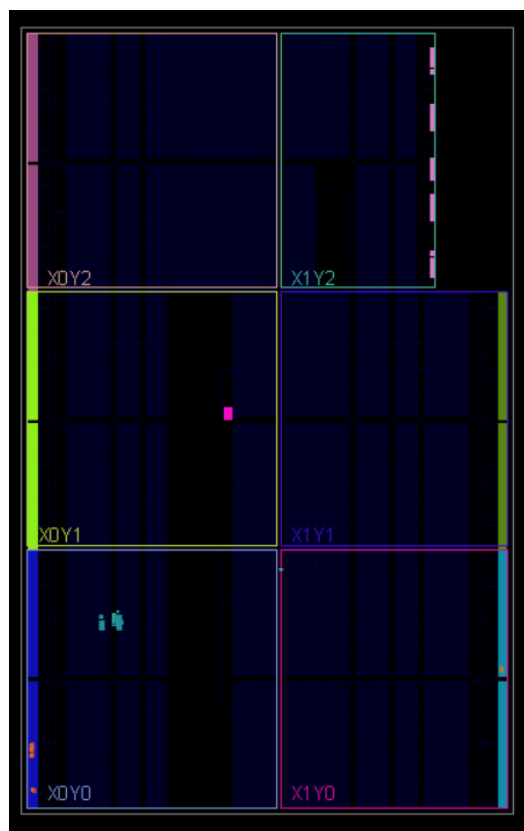
Resource	Utilization	Available	Utilization %
LUT	29	20800	0.14
FF	44	41600	0.11
BRAM	0.50	50	1.00
IO	5	106	4.72



Critical Path:



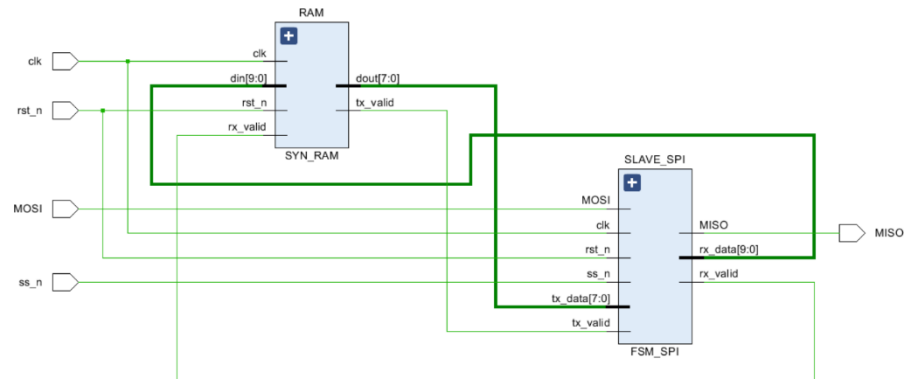
Device:



Sequential Encoding

4. Elaboration:

Schematic:



Messages Tab:

Vivado Commands (3 infos)

General Messages (3 infos)

- [IP_Flow 19-234] Refreshing IP repositories
- [IP_Flow 19-1704] No user IP repositories specified
- [IP_Flow 19-2313] Loaded Vivado IP repository 'G:\Mvado3\Vivado\Nvado\2018.2\data\ip'.

Elaborated Design (13 infos)

General Messages (13 infos)

- [Synth 8-6157] synthesizing module 'WRAPPER' [WRAPPER.v:1] (2 more like this)
- [Synth 8-5534] Detected attribute (* fsm_encoding = "sequential" *) [FSM_SPI.v:19]
- [Synth 8-155] case statement is not full and has no default [FSM_SPI.v:32] (1 more like this)
- [Synth 8-6155] done synthesizing module 'FSM_SPI' (1#1) [FSM_SPI.v:1] (2 more like this)
- [Device 21-403] Loading part xc7a35lcpq236-1L
- [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.

5. Synthesis:

Encoding:

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

Messages Tab:

Synthesis (2 warnings, 33 infos)

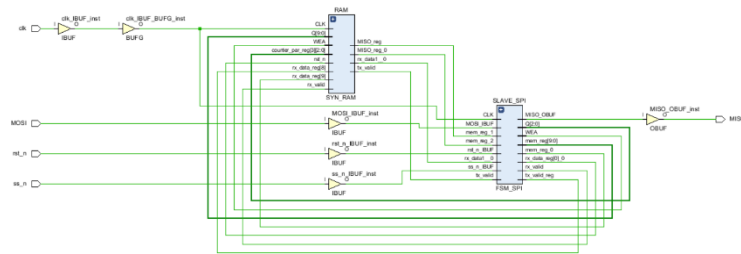
- [Common 17-348] Got license for feature "Synthesis" and/or device "xc7a35f"
- [Synth 8-6157] synthesizing module 'WRAPPER' [WRAPPER.v:1] (2 more like this)
- [Synth 8-5534] Detected attribute (* fsm_encoding = "sequential" *) [FSM_SPI.v:19]
- [Synth 8-155] case statement is not full and has no default [FSM_SPI.v:32] (1 more like this)
- [Synth 8-155] done synthesizing module 'FSM_SPI' (1#1) [FSM_SPI.v:1] (2 more like this)
- [Device 21-403] Loading part xc7a35lcpq236-1L
- [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.

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.

Schematic:

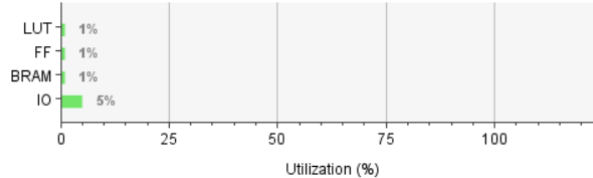


Timing Report Summary:

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
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 100	Total Number of Endpoints: 100	Total Number of Endpoints: 44
All user specified timing constraints are met.		

Utilization Report:

Resource	Utilization	Available	Utilization %
LUT	29	20800	0.14
FF	44	41600	0.11
BRAM	0.50	50	1.00
IO	5	106	4.72

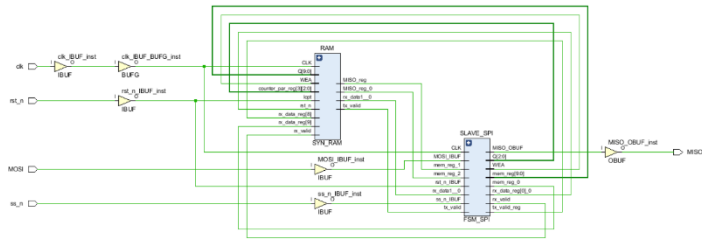


6. Implementation:

Messages Tab:

- ▼ 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.

Schematic:



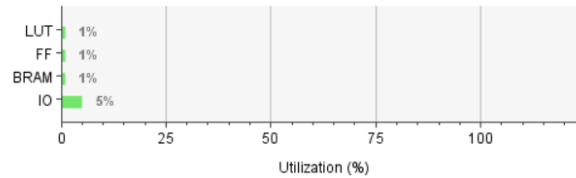
Timing Report Summary:

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.465 ns	Worst Hold Slack (WHS): 0.071 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: 100	Total Number of Endpoints: 100	Total Number of Endpoints: 44

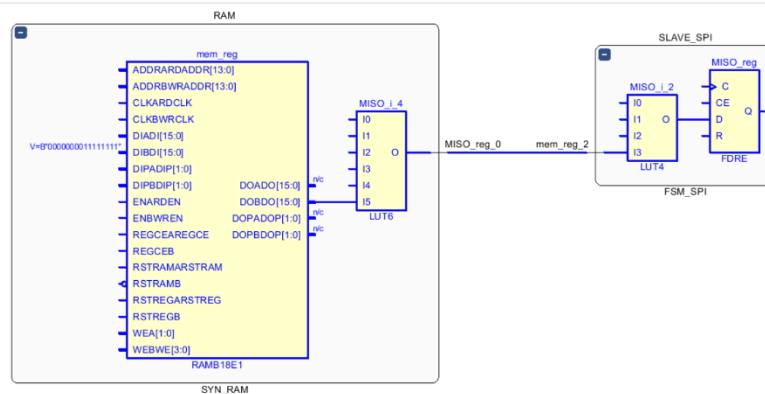
All user specified timing constraints are met.

Utilization Report:

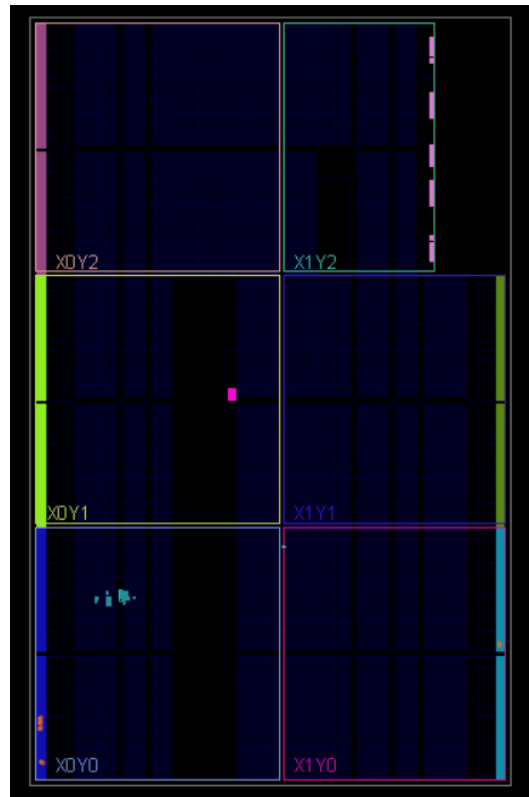
Resource	Utilization	Available	Utilization %
LUT	29	20800	0.14
FF	44	41600	0.11
BRAM	0.50	50	1.00
IO	5	106	4.72



Critical Path:



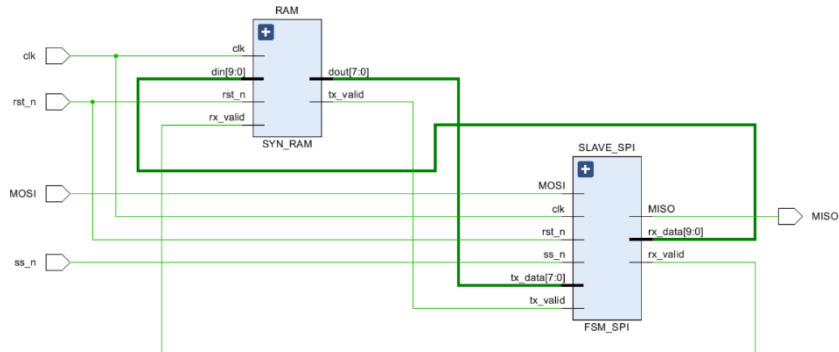
Device:



One Hot Encoding

7. Elaboration:

Schematic:



Messages Tab:

Vivado Commands (3 infos)

General Messages (3 infos)

[IP_Flow 19-234] Refreshing IP repositories

[IP_Flow 19-1704] No user IP repositories specified

[IP_Flow 19-2313] Loaded Vivado IP repository 'G:\Vivado3\Vivado\Vivado\2018.2\data\ip'.

Elaborated Design (2 Infos)

General Messages (2 infos)

[Project 1-570] Preparing netlist for logic optimization

[Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).

8. Synthesis:

Encoding:

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

Messages Tab:

Warning (2)

Info (53)

Status (26)

Show All

Synthesis (2 warnings, 33 infos)

[Common 17-348] Got license for feature 'Synthesis' and/or device 'xc7a300'

[Synth 8-6157] synthesizing module 'VWRAPPER' (VWRAPPER.v1) (2 more like this)

[Synth 8-5534] Detected attribute 'ram_encoding = 'one_hot'' for 'FSM_SPI' (1)

[Synth 8-155] case statement is not full and has no default 'FSM_SPI' (32) (1 more like this)

[Synth 8-6155] done synthesizing module 'FSM_SPI' (1W1) (FSM_SPI.v1) (2 more like this)

[Device 21-403] Loading part xc7a300pg236-1L

[Project 1-219] Implementation specific constraints were found while reading constraint file [G:\Final_project_dig\Constraints_Final_project.vdc]. These constraints will be ignored for synthesis but will be used in implementation. Impacted constraints are listed in the file [X:\VWRAPPER_project\pdc]. Resolution: To avoid this warning, move constraints listed in [Undefined] to another XDC file and exclude this new file from synthesis with the used_in_synthesis property (File Properties dialog in GUI) and re-run elaboration/synthesis.

[Synth 8-802] inferred FSM for state register 'cs_reg' in module 'FSM_SPI'

[Synth 8-8544] ROM 'rs' won't be mapped to Block RAM because address size (3) smaller than threshold (5) (4 more like this)

[Synth 8-3354] encoded FSM with state register 'cs_reg' using encoding 'one-hot' in module 'FSM_SPI'

[Synth 8-3277] inferred path for variable 'FSM_stateOut_reg' (FSM_SPI.v2)

[Synth 8-4480] The timing for the instance 'L_RAMMem_reg' (implemented as a block RAM) might be sub-optimal as no optional output register could be merged into the block ram. Providing additional output register may help in improving timing. (1 more like this)

[Project 1-570] Translating synthesized netlist

[Netlist 29-17] Analyzing 10 Unisim elements for replacement

[Netlist 29-26] Unisim Transformation completed in 0 CPU seconds

[Project 1-570] Preparing netlist for logic optimization (1 more like this)

[Opt 31-138] Pushed 0 inverter(s) to 0 load pin(s).

[Project 1-115] Unisim Transformation Summary
No Unisim elements were transformed. (1 more like this)

[Common 17-83] Releasing license: Synthesis

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

[Common 17-1385] The checkpoint G:\Final_project_dig\project_Final\project_Final_xm31synth_110VWRAPPER.dcp has been generated.

[Synth 4-4] Encoding_report_utilization.de VWRAPPER_utilization_synth.rpt-go VWRAPPER_utilization_synth.go

[Common 17-2096] Exiting Vivado at Mon Aug 5 6:22:47 2024...

Synthesized Design (8 infos)

General Messages (3 infos)

[Netlist 29-17] Analyzing 10 Unisim elements for replacement

[Netlist 29-26] 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 1-115] Unisim Transformation Summary
No Unisim elements were transformed.

22 | Page

[illegible]

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.898 ns	Worst Hold Slack (WHS): 0.144 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: 100	Total Number of Endpoints: 100	Total Number of Endpoints: 46
All user specified timing constraints are met.		

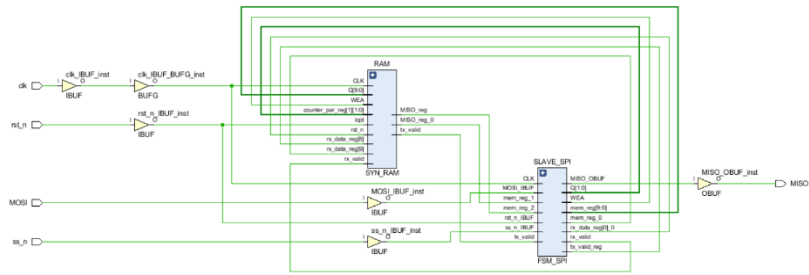
Resource	Utilization	Available	Utilization %
LUT	31	20800	0.15
FF	48	41600	0.12
BRAM	0.50	50	1.00
IO	5	106	4.72

A horizontal bar chart titled 'Utilization (%)' on the x-axis. The x-axis has major grid lines at 0, 25, 50, 75, and 100. The y-axis lists the resources: LUT, FF, BRAM, and IO. The bars represent the utilization percentage for each resource: LUT is 1%, FF is 1%, BRAM is 1%, and IO is 5%.

Resource	Utilization (%)
LUT	1%
FF	1%
BRAM	1%
IO	5%

- ▼ Implemented Design (12 infos)
- ▼ General Messages (12 infos)
 - 1 [Netlist 29-17] Analyzing 5 Unisim elements for replacement
 - 1 [Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
 - 1 [Project 1-479] Netlist was created with Vivado 2018.2
 - 1 [Project 1-570] Preparing netlist for logic optimization
 - 1 [Timing 38-478] Restoring timing data from binary archive.
 - 1 [Timing 38-479] Binary timing data restore complete.
 - 1 [Project 1-856] Restoring constraints from binary archive.
 - 1 [Project 1-853] Binary constraint restore complete.
 - > 1 [Timing 38-91] UpdateTimingParams: Speed grade: -1L, Delay Type: min_max. (1 more like this)
 - > 1 [Timing 38-191] Multithreading enabled for timing update using a maximum of 2 CPUs (1 more like this)

Schematic:

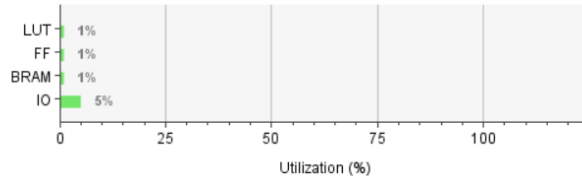


Timing Report Summary:

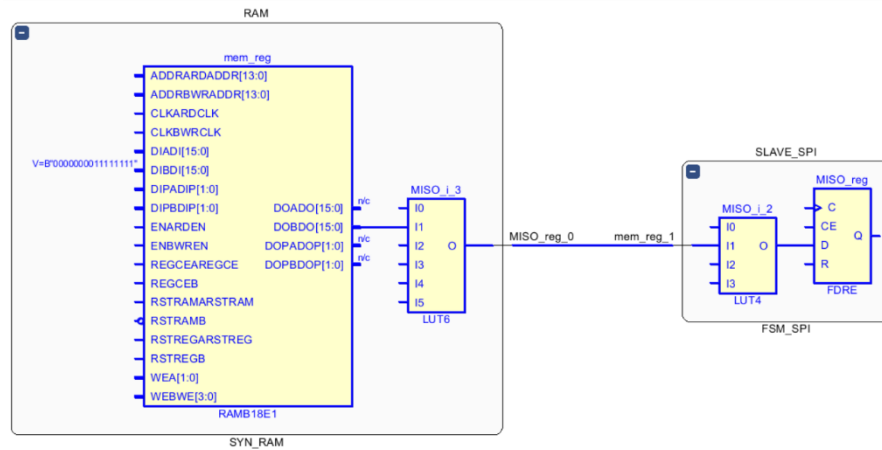
Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 5.415 ns	Worst Hold Slack (WHS): 0.046 ns	Worst Pulse Width Slack (WPWS): 4.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 100	Total Number of Endpoints: 100	Total Number of Endpoints: 46

Utilization Report:

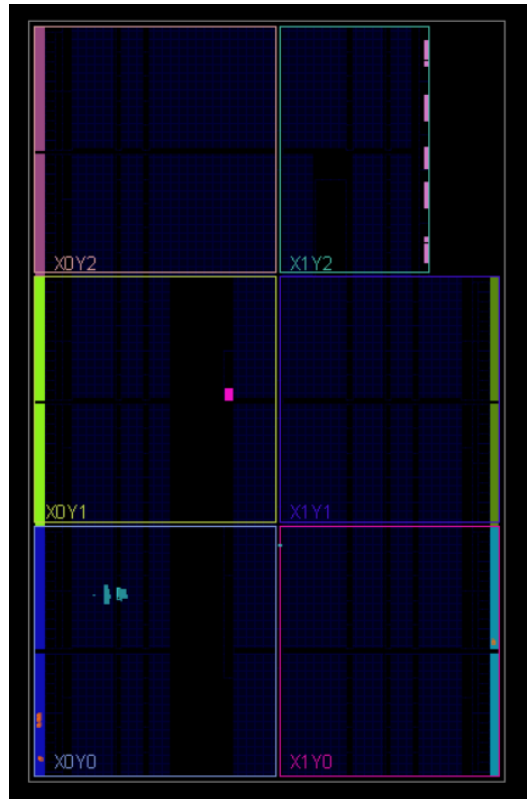
Resource	Utilization	Available	Utilization %
LUT	31	20800	0.15
FF	48	41600	0.12
BRAM	0.50	50	1.00
IO	5	106	4.72



Critical Path:



Device:



The Best Encoding

based on the best timing report that gives the high setup/hold slack after implementation.

Gray Encoding:

- **Worst Negative Setup Slack (WNSS):** 5.446 ns
- **Worst Hold Slack (WHS):** 0.042 ns
- **Worst Pulse Width Slack (WPWS):** 4.500 ns

Sequential Encoding:

- **Worst Negative Setup Slack (WNSS):** 5.465 ns
- **Worst Hold Slack (WHS):** 0.071 ns
- **Worst Pulse Width Slack (WPWS):** 4.500 ns

One-Hot Encoding:

- **Worst Negative Setup Slack (WNSS):** 5.415 ns
- **Worst Hold Slack (WHS):** 0.046 ns
- **Worst Pulse Width Slack (WPWS):** 4.500 ns

Analysis

- **Setup Slack:** Sequential encoding has the highest worst negative setup slack (5.465 ns), indicating it has the most timing margin for setup constraints.
- **Hold Slack:** Sequential encoding has the best worst hold slack (0.071 ns), providing the highest timing margin for hold constraints.
- **Pulse Width Slack:** All three encodings have the same worst pulse width slack (4.500 ns).

Comparison:

- **Setup Slack (Higher is better):**
 - Gray: 5.446 ns
 - Sequential: 5.465 ns
 - One-Hot: 5.415 ns
- **Hold Slack (Higher is better):**
 - Gray: 0.042 ns
 - Sequential: 0.071 ns
 - One-Hot: 0.046 ns

Conclusion

Given that Sequential encoding has the highest setup and hold slack values, it provides the best timing margin overall. Therefore, to operate at the highest frequency possible, you should choose **Sequential encoding**.

Constraints File:

```
## Clock signal
set_property -dict {PACKAGE_PIN W5 IOSTANDARD LVCMOS33} [get_ports clk]
create_clock -period 10.000 -name sys_clk_pin -waveform {0.000 5.000} -add [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]

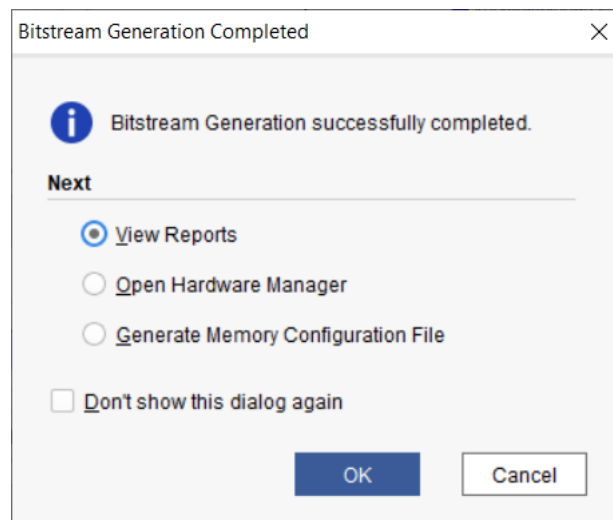
## LEDs
set_property -dict {PACKAGE_PIN U16 IOSTANDARD LVCMOS33} [get_ports MISO]

## Configuration options, can be used for all designs
set_property CONFIG_VOLTAGE 3.3 [current_design]
set_property CFGBV5 VCC0 [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]

create_debug_core u_ila_0 ila
set_property ALL_PROBE_SAME_MU true [get_debug_cores u_ila_0]
set_property ALL_PROBE_SAME_MU_CNT 1 [get_debug_cores u_ila_0]
set_property C_ADV_TRIGGER false [get_debug_cores u_ila_0]
set_property C_DATA_DEPTH 4096 [get_debug_cores u_ila_0]
set_property C_EN_STRG_QUAL false [get_debug_cores u_ila_0]
set_property C_INPUT_PIPE_STAGES 0 [get_debug_cores u_ila_0]
set_property C_TRIGIN_EN false [get_debug_cores u_ila_0]
set_property C_TRIGOUT_EN false [get_debug_cores u_ila_0]
set_property port_width 1 [get_debug_ports u_ila_0/clk]
connect_debug_port u_ila_0/clk [get_nets [list clk_IBUF_BUFG]]
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe0]
set_property port_width 1 [get_debug_ports u_ila_0/probe0]
connect_debug_port u_ila_0/probe0 [get_nets [list clk_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe1]
set_property port_width 1 [get_debug_ports u_ila_0/probe1]
connect_debug_port u_ila_0/probe1 [get_nets [list MISO_OBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe2]
set_property port_width 1 [get_debug_ports u_ila_0/probe2]
connect_debug_port u_ila_0/probe2 [get_nets [list MOSI_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe3]
set_property port_width 1 [get_debug_ports u_ila_0/probe3]
connect_debug_port u_ila_0/probe3 [get_nets [list rst_n_IBUF]]
create_debug_port u_ila_0 probe
set_property PROBE_TYPE DATA_AND_TRIGGER [get_debug_ports u_ila_0/probe4]
set_property port_width 1 [get_debug_ports u_ila_0/probe4]
connect_debug_port u_ila_0/probe4 [get_nets [list ss_n_IBUF]]
set_property C_CLK_INPUT_FREQ_HZ 300000000 [get_debug_cores dbg_hub]
set_property C_ENABLE_CLK_DIVIDER false [get_debug_cores dbg_hub]
set_property C_USER_SCAN_CHAIN 1 [get_debug_cores dbg_hub]
connect_debug_port dbg_hub/clk [get_nets clk_IBUF_BUFG]
```

Bitstream Generation Success:



Master (SPI SLAVE Interface TB) RTL Code Snippet:

```
module MASTER_tb();
//Signals Declaration
reg MOSI;
reg ss_n;
reg clk;
reg rst_n;
wire MISO;
//DUT Instantiation
WRAPPER DUT(.MOSI(MOSI),.MISO(MISO),.ss_n(ss_n),.clk(clk),.rst_n(rst_n));
//Clock Generation
initial begin
    clk = 1;
    forever
        #5 clk = ~clk;
end
//Test Stimulus Generator
initial begin
    //Activate Reset ,Un-Enable ss_n& Initialize all signals
    rst_n = 0;
    ss_n = 1;
    MOSI = 0;
    @(negedge clk);
    //De-Activate Reset& Start testing
    rst_n = 1;

    //Test Writing Address
    ss_n = 0;
    @(negedge clk);

    MOSI = 0;//Write
    @(negedge clk);

    MOSI = 0;
    @(negedge clk);
    MOSI = 0;
    @(negedge clk);
    //2'b00 >> To write address
    repeat(3) begin
        MOSI = 0;
        @(negedge clk);
        MOSI = 1;
        @(negedge clk);
        MOSI = 1;
        @(negedge clk);
        MOSI = 1;
        @(negedge clk);
    end
    //Address : 8'b01110111
    ss_n = 1;
    @(negedge clk);
end
```

```

//Test Writing Data
ss_n = 0;
@(negedge clk);
MOSI = 0;//Write
@(negedge clk);

MOSI = 0;
@(negedge clk);
MOSI = 1;
@(negedge clk);
//2'b01 >> To write Data
MOSI = 1;
@(negedge clk);
MOSI = 0;
@(negedge clk);
MOSI = 1;
@(negedge clk);
MOSI = 0;
@(negedge clk);
MOSI = 1;
@(negedge clk);
MOSI = 0;
@(negedge clk);
MOSI = 1;
@(negedge clk);
MOSI = 0;
@(negedge clk);
//Data:8'b10101010
ss_n = 1;
@(negedge clk);

//Test Reading Address
ss_n = 0;
@(negedge clk);

MOSI = 1;//Read
@(negedge clk);

MOSI = 1;
@(negedge clk);
MOSI = 0;
@(negedge clk);
//2'b10 >> To Read address
repeat(3) begin
    MOSI = 0;
    @(negedge clk);
    MOSI = 1;
    @(negedge clk);
    MOSI = 1;
    @(negedge clk);
    MOSI = 1;
    @(negedge clk);
end
//Address : 8'b01110111
ss_n = 1;

```

```

//Address : 8'b01110111
ss_n = 1;
@(negedge clk);

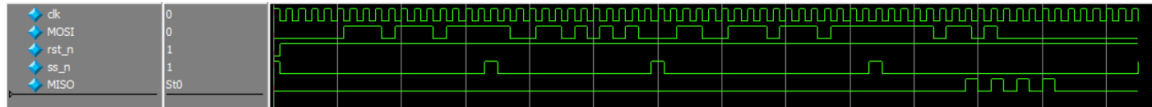
//Test Reading Data
ss_n = 0;
@(negedge clk);

MOSI = 1;//Read
@(negedge clk);

MOSI = 1;
@(negedge clk);
MOSI = 1;
@(negedge clk);
//2'b11 >> To Read Data
MOSI = 0;
@(negedge clk);
MOSI = 1;
@(negedge clk);
MOSI = 1;
@(negedge clk);
MOSI = 0;
@(negedge clk);
MOSI = 1;
@(negedge clk);
MOSI = 0;
@(negedge clk);
MOSI = 0;
@(negedge clk);
MOSI = 0;
@(negedge clk);
//Dummy Data
repeat(8) @(negedge clk);
ss_n = 1;
$stop;
end
//Test Monitor & Results
initial begin
    $monitor("MOSI = %b, MISO = %b, ss_n = %b",MOSI, MISO, ss_n);
end
endmodule

```

Simulation Waveform Snippet:



Do File:

```
vlib work
vlog SYN_RAM.v FSM_SPI.v WRAPPER.v MASTER_tb.v
vsim -voptargs=+acc work.MASTER_tb
add wave *
run -all
#quit -sim
```