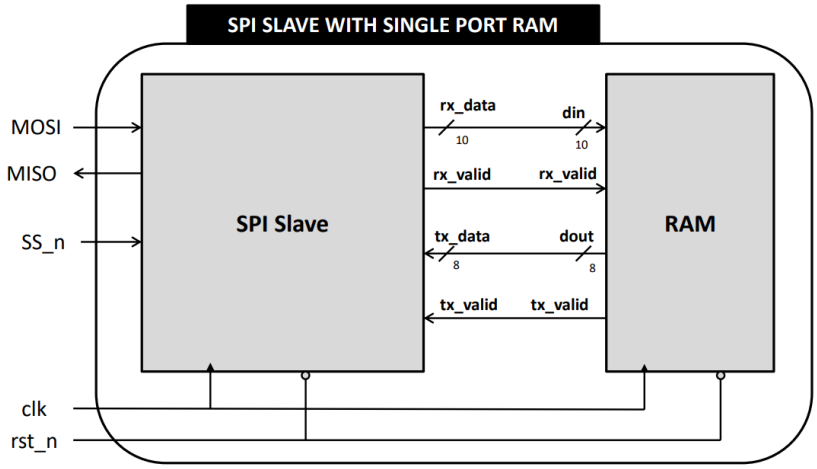
SPI Slave with Single Port RAM

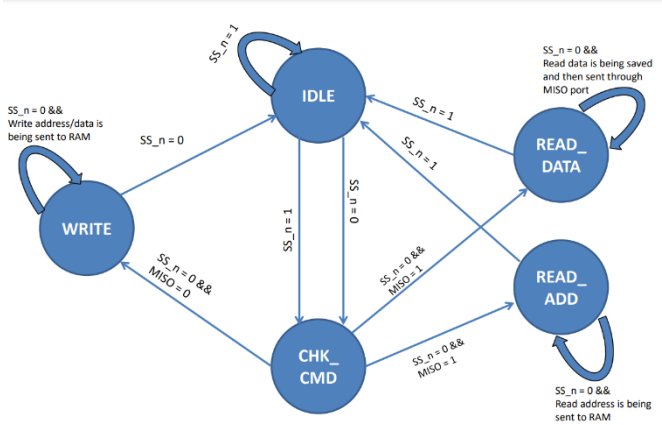
Broken clk team

Zeyad Ashraf Abdulaziz
Yousef Elbadry Haroun Mohamed

Introduction

The SPI (Serial Peripheral Interface) function plays a critical role in enabling communication between digital devices in embedded systems. SPI is a synchronous serial communication protocol that allows for fast and efficient data exchange between a master device and one or more slave devices. In the context of this project, the SPI function is responsible for facilitating the transfer of data between an SPI master and a slave device, which, in this case, interfaces with a single-port RAM, The SPI function operates using four key signals: Master Out Slave In (MOSI), Master In Slave Out (MISO), Serial Clock (SCLK), and Slave Select (SS). The MOSI line carries data from the master to the slave, while the MISO line is used by the slave to send data back to the master. The SCLK line synchronizes data transfer, ensuring that data is correctly sampled and shifted on both the master and slave sides. The SS line enables the slave device when asserted, allowing it to communicate with the master, by implementing the SPI function in this project, the slave device can accurately receive, store, and transmit data in coordination with the master device. This reliable and high-speed communication link is essential for applications where rapid data exchange and precise control are required, such as in sensor networks, memory interfacing, and real-time data processing systems.





RTL Codes:

Ram

```
module Ram (din,clk,rst_n,rx_valid,dout,tx_valid);
parameter MEM_DEPTH = 256;
parameter ADDR_SIZE = 8;
input clk,rst_n,rx_valid;
input [9:0] din;
output reg [ADDR_SIZE-1:0] dout;
output reg tx_valid;
reg [ADDR_SIZE-1:0] mem [MEM_DEPTH-1:0];
reg [ADDR_SIZE-1:0] hold_read_write;

always @(posedge clk) begin
if (~rst_n)
dout <= 0;
else
case (din[9:8])
0 : if (rx_valid) begin
    hold_read_write <= din[7:0];
    tx_valid <= 0;
end

1 : if (rx_valid) begin
    mem[hold_read_write] <= din[7:0];
    tx_valid <= 0;
end

2 : begin
    hold_read_write <= din[7:0];
    tx_valid <= 1;
end

3 : begin
    dout <= mem[hold_read_write];
    tx_valid <= 1;
end
endcase
end
endmodule
```

SPI SLAVE

```
module SPI_SLAVE (MOSI,MISO,SS_n,clk,rst_n,rx_data,rx_valid,tx_data,tx_valid);
parameter IDLE = 3'b000,
          CHK_CMD = 3'b001,
          READ_ADD = 3'b010,
          READ_DATA = 3'b011,
          WRITE = 3'b100;
input MOSI,clk,rst_n,tx_valid,SS_n;
input [7:0] tx_data; reg [7:0] shift_value;
output reg MISO,rx_valid;
output reg [9:0] rx_data;
(*fsm_encoding="sequential"*)
reg [2:0] cs,ns;
reg [3:0] counter_write;
reg [4:0] counter_read;
reg [9:0] serial_2_parallel;


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


//next stat logic
always @(cs,SS_n) begin
case (cs)
IDLE : if (SS_n)
            ns = IDLE;
       else
            ns = CHK_CMD;

CHK_CMD:begin
     if (SS_n)
            ns = IDLE;
        else if (MOSI)
        begin
            if (tx_valid)
                ns = READ_DATA;
            else
                ns = READ_ADD;
```

```
        end
        else
            ns = WRITE;
        end

READ_ADD : if (SS_n)
                ns = IDLE;
           else if (counter_write < 10)
                ns = READ_ADD;

READ_DATA :if (SS_n)
                ns = IDLE;
           else if (counter_read < 18)
                ns = READ_DATA;

WRITE : if (SS_n)
            ns = IDLE;
        else if (counter_write < 10)
            ns = WRITE;
endcase
end

//output logic
always @(posedge clk)
begin
if (~rst_n)
begin
    counter_write <= 0;
    counter_read <= 0;
    rx_data <= 0;
    serial_2_parallel <= 0;
end

else if ((cs == WRITE) || (cs == READ_ADD) || ((cs == READ_DATA) && (counter_read
<= 10)) && (~SS_n))
begin
serial_2_parallel <= {serial_2_parallel[8:0],MOSI};
counter_write <= counter_write + 1;

if (cs == READ_DATA)
counter_read <= counter_read + 1;

if (counter_write == 10) begin
rx_valid <= 1;
rx_data <= serial_2_parallel;
```

```
counter_write <= 0;
serial_2_parallel <= 0;
end
end

else if ((cs == (READ_DATA)) && (~SS_n) && (counter_read > 10))
begin
    if (tx_valid)
    begin
shift_value <= counter_read-11;
MISO <= tx_data >> shift_value;
counter_read <= counter_read + 1;
    end
    if (counter_read == 18)
        counter_read <= 0;
end

else begin
counter_read <= 0;
counter_write <= 0;
end
end
endmodule
```

wrapper

```
module Wrapper (MOSI,MISO,SS_n,clk,rst_n);
input MOSI,SS_n,clk,rst_n;
output MISO;
wire [9:0] rx_data; wire rx_valid; wire [7:0] tx_data; wire tx_valid;
SPI_SLAVE m1 (MOSI,MISO,SS_n,clk,rst_n,rx_data,rx_valid,tx_data,tx_valid);
Ram m2 (rx_data,clk,rst_n,rx_valid,tx_data,tx_valid);
endmodule
```

Test bench

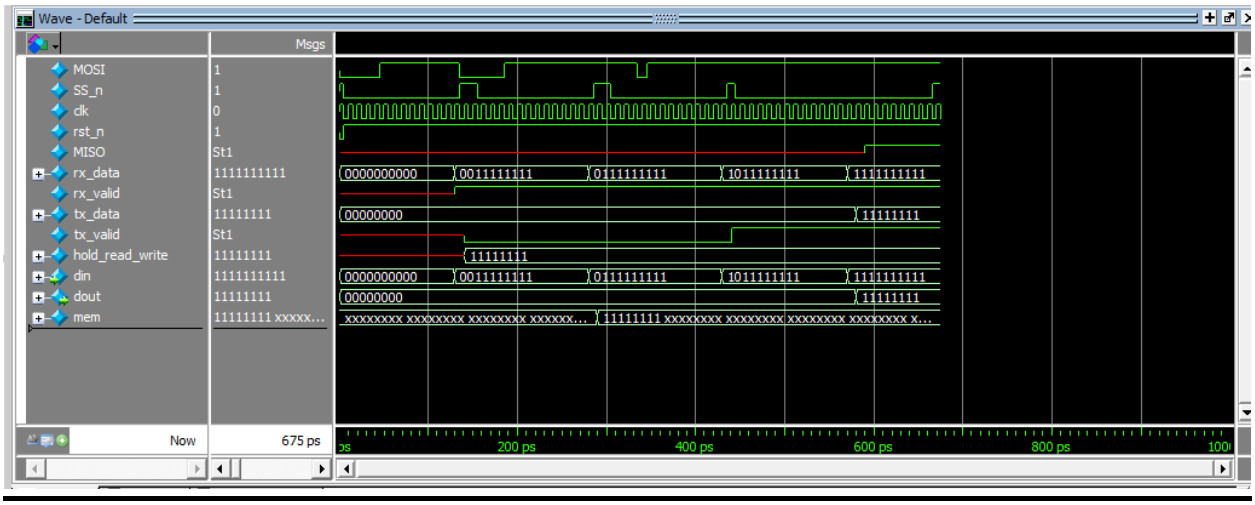
```
module Wrapper_tb();

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

    Wrapper Dut (
        .MOSI(MOSI),
        .MISO(MISO),
        .SS_n(SS_n),
        .clk(clk),
        .rst_n(rst_n)
    );



    always
    begin
        clk= 1;
        forever
            #5 clk = ~clk;
    end



    initial begin
        rst_n = 0;
        MOSI = 0;
        SS_n = 1;
        @(negedge clk)
        rst_n = 1;
        MOSI = 0;
        SS_n = 0;
        @(negedge clk);
        @(negedge clk);
        //*****************************************************************
***
        // Write Address to RAM
        MOSI = 0;
        @(negedge clk)
        MOSI = 0;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
```

```
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        repeat(2) @(negedge clk);

        MOSI = 0;
        SS_n = 1;
        @(negedge clk);
        @(negedge clk);


        SS_n = 0;
        @(negedge clk);
        @(negedge clk);
        // Write Data to RAM
        MOSI = 0;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        repeat(2) @(negedge clk);
```

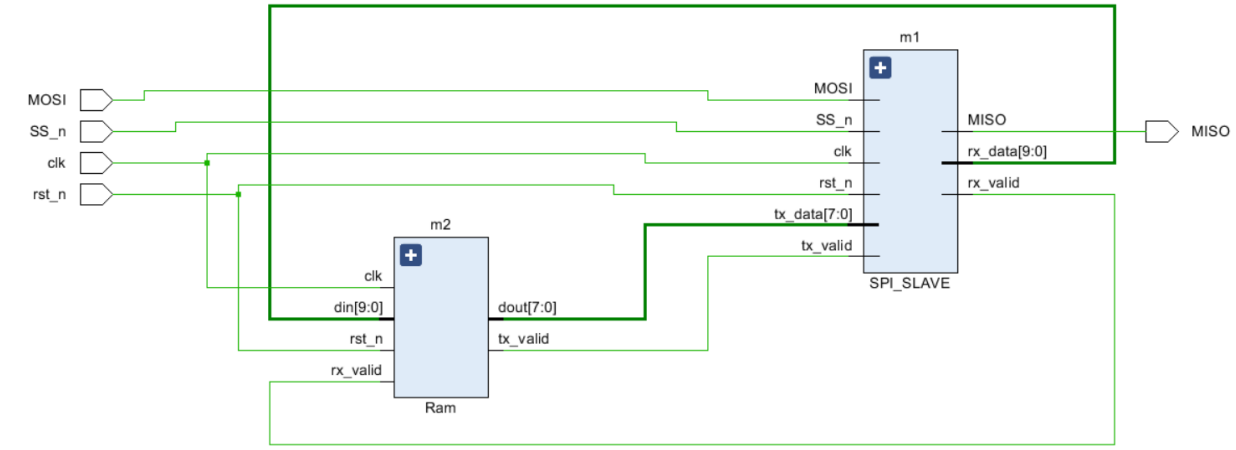
```
        SS_n = 1;
        @(negedge clk)
        @(negedge clk);

        SS_n = 0;
        @(negedge clk)
        @(negedge clk);
        // Read Address to RAM
        MOSI = 1;
        @(negedge clk)
        MOSI = 0;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        repeat(2) @(negedge clk);


        SS_n = 1;
        MOSI = 1;
        @(negedge clk)

        SS_n = 0;
        @(negedge clk)
        @(negedge clk)

        // Read Data to RAM
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
```

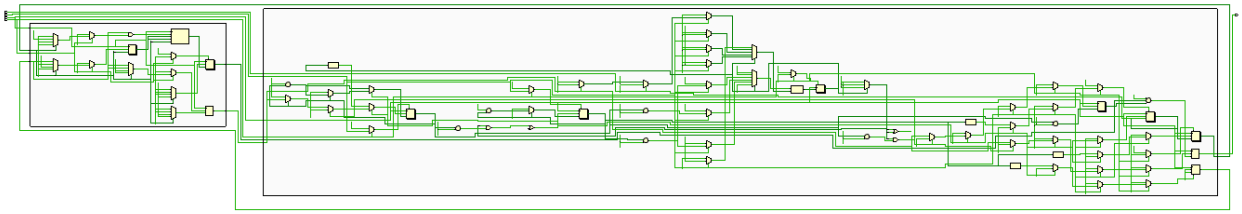
```
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        MOSI = 1;
        @(negedge clk)
        repeat(10) @(negedge clk);

        SS_n = 1;
        @(negedge clk)

        $display("MISO = %b", MISO);
        //*****************************************************************
***
        $stop;
    end

endmodule
```

Wave form snippets

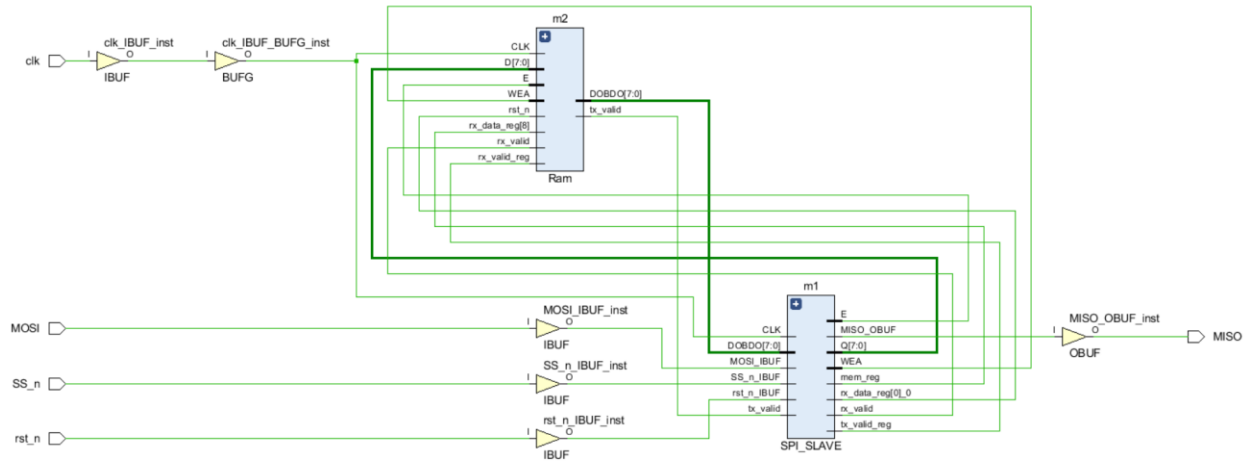


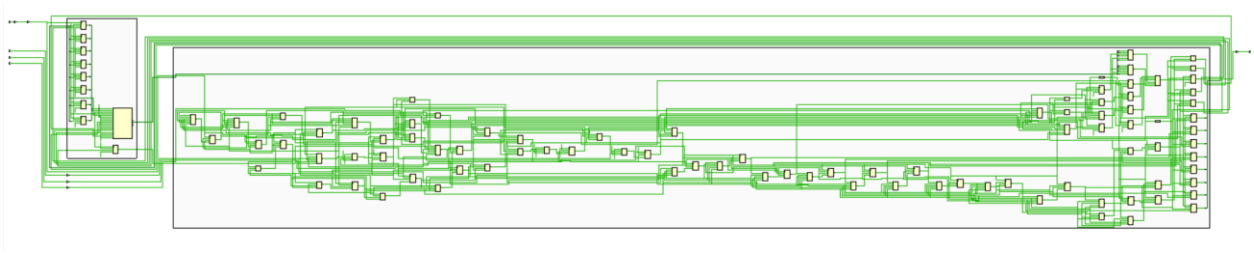
Elaboration schematic

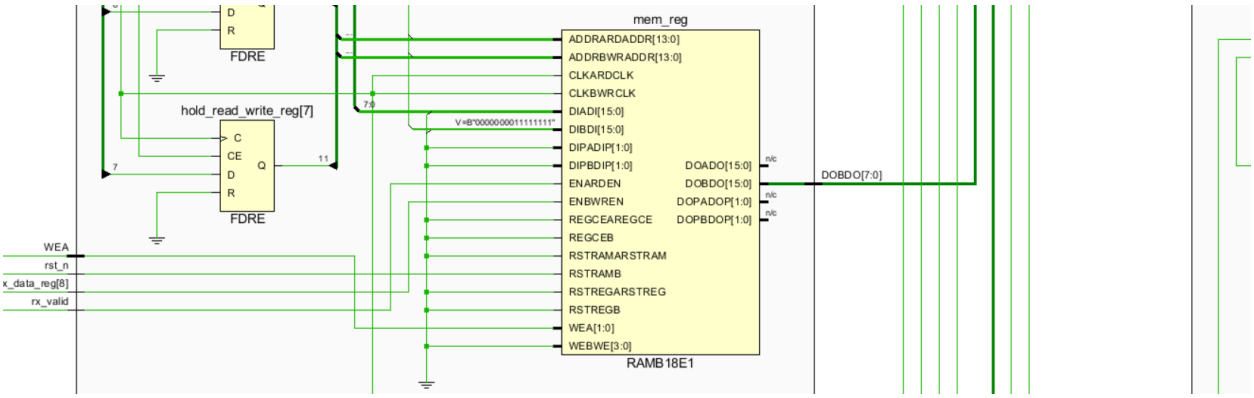


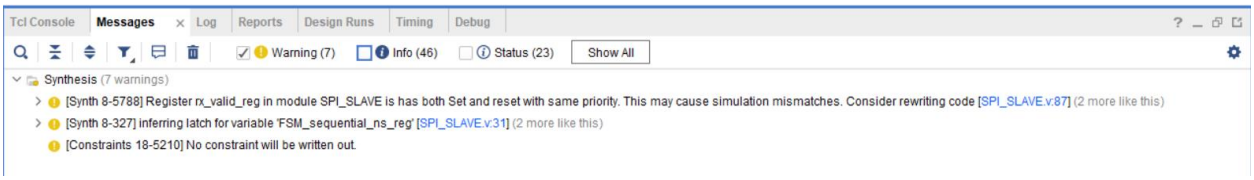


Synthesis

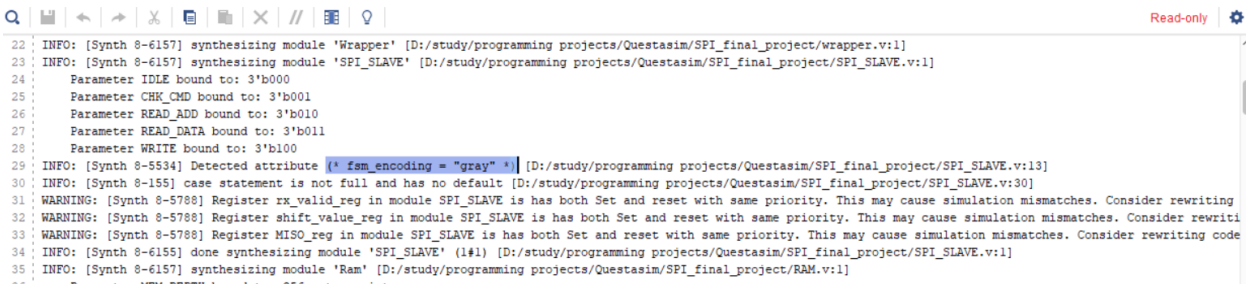

clk
clk_IBUF_inst
IBUF
clk_IBUF_BUFG_inst
BUFG
m2
CLK
D[7:0]
E
WEA
rst_n
rx_data_reg[8]
rx_valid
rx_valid_reg
DOBDO[7:0]
tx_valid
Ram
MOSI
MOSI_IBUF_inst
IBUF
SS_n
SS_n_IBUF_inst
IBUF
rst_n
rst_n_IBUF_inst
IBUF
m1
CLK
DOBDO[7:0]
MOSI_IBUF
SS_n_IBUF
rst_n_IBUF
tx_valid
E
MISO_OBUF
O[7:0]
WEA
mem_reg
rx_data_reg[0]_0
rx_valid
tx_valid_reg
SPI_SLAVE
MISO_OBUF_inst
OBUF
MISO




FDRE
hold_read_write_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
DOBDO[7:0]
WEA
rst_n
rx_data_reg[8]
rx_valid



Gray encoding

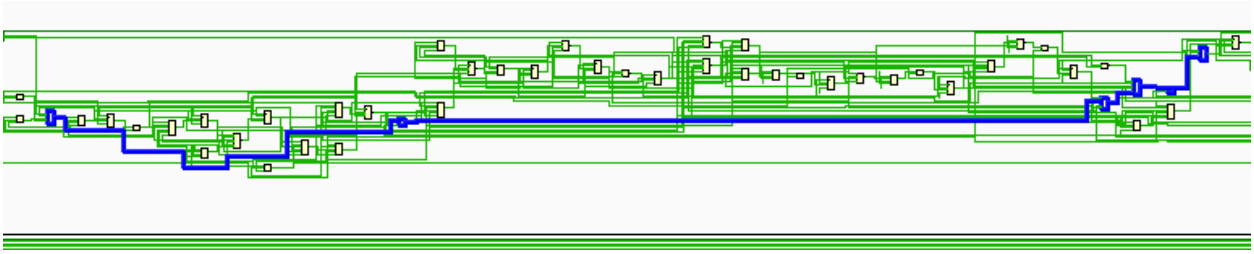


State	New Encoding	Previous Encoding
IDLE	001	000
CHK_CMD	111	001
READ_DATA	011	011
READ_ADD	010	010
WRITE	000	100

Design Timing Summary

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.223 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:	124	Total Number of Endpoints:	124	Total Number of Endpoints:	51

Name	Slack	Levels	High Fanout	From	To	Total Delay	Logic Delay	Net D
Path 1	5.223	4	14	m1/counter_read_reg[2]/C	m1/serial_2_parallel_reg[0]/R	4.164	1.149	
Path 2	5.223	4	14	m1/counter_read_reg[2]/C	m1/serial_2_parallel_reg[1]/R	4.164	1.149	



Log | Reports | Design Runs | **Utilization** | Timing | Debug

Hierarchy

Name	Slice LUTs (20800)	Slice Registers (41600)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
Wrapper	38	51	0.5	5	1
dbg_hub (dbg_hub_CV)	0	0	0	0	0
m1 (SPI_SLAVE)	38	42	0	0	0
m2 (Ram)	0	9	0.5	0	0

Tcl Console | **Messages** | Log | Reports | Design Runs | Debug

Warning (7) | Info (46) | Status (29) | Show All

- 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.

One hot encoding

Schematic | **synth_1_synth_synthesis_report_0 - synth_1**

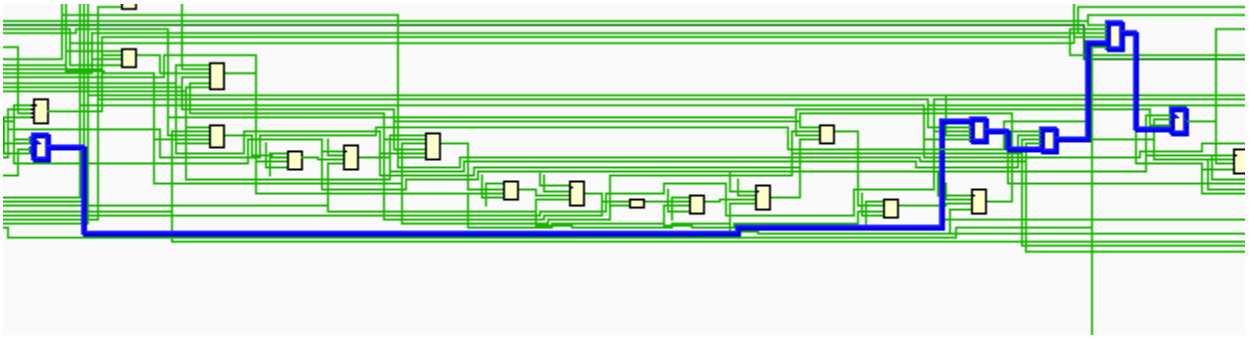
D:/study/programming projects/Vivado/SPI_SLAVE_WITH_SINGLE_PORT_RAM/SPI_SLAVE_WITH_SINGLE_PORT_RAM.runs/synth_1/Wrapper.vds

Read-only

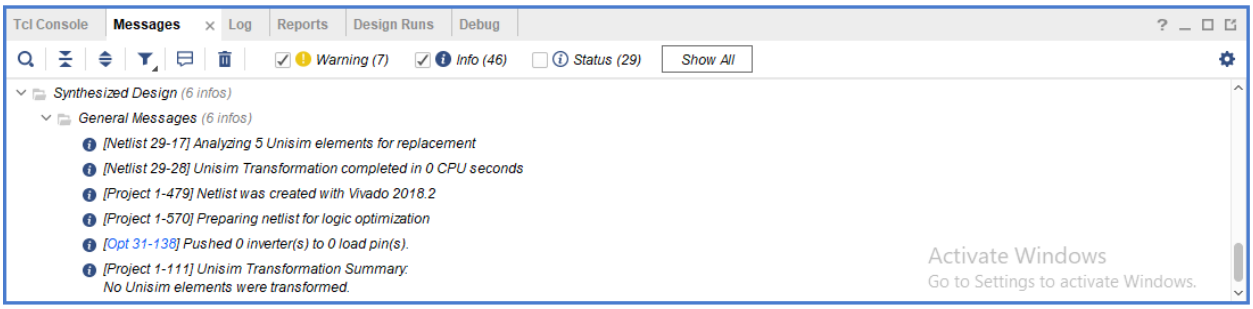
```
28     Parameter WRITE bound to: 3'b100
29  INFO: [Synth 8-5534] Detected attribute (* fsm_encoding = "one_hot" *) [D:/study/programming projects/Questasim/SPI_final_project/SPI_SLAVE.v:13]
30  INFO: [Synth 8-155] case statement is not full and has no default [D:/study/programming projects/Questasim/SPI_final_project/SPI_SLAVE.v:30]
```

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

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.719 ns	Worst Hold Slack (WHS):	0.148 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:	133	Total Number of Endpoints:	133	Total Number of Endpoints:	53



Name	Slice LUTs (20800)	Slice Registers (41600)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
N Wrapper	33	55	0.5	5	1
m1 (SPI_SLAVE)	33	46	0	0	0
m2 (Ram)	0	9	0.5	0	0



Sequential encoding

Project Summary | Device | Schematic | SPI_SLAVE.v | synth_1_synth_synthesis_report_0 - synth_1

D:/study/programming projects/Vivado/SPI_SLAVE_WITH_SINGLE_PORT_RAM/SPI_SLAVE_WITH_SINGLE_PORT_RAM.runs/synth_1/Wrapper.vds

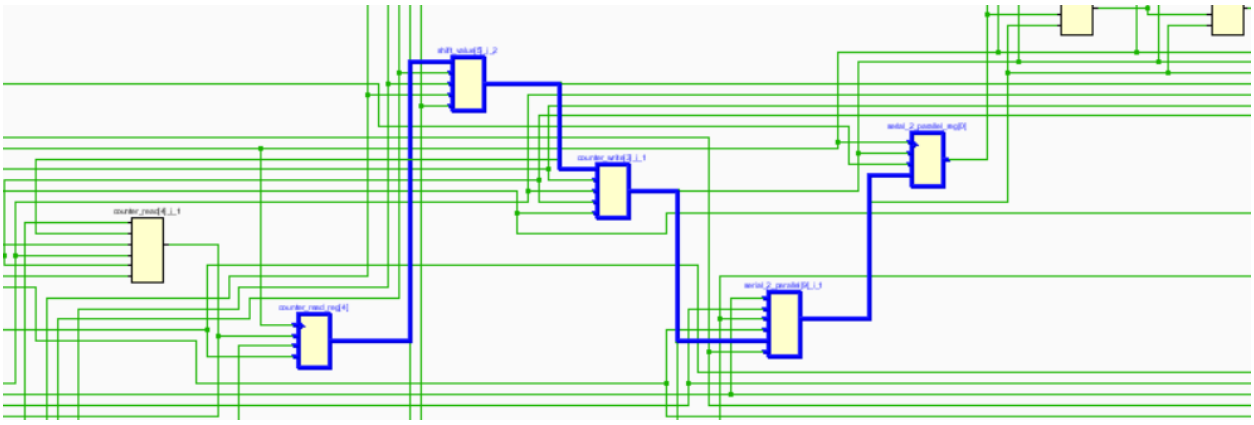
Read-only

```
28     Parameter WRITE bound to: 3'b100
29  INFO: [Synth 8-5534] Detected attribute (* fsm_encoding = "sequential" *) [D:/study/programming projects/Questasim/SPI_final_project/SPI_SLAVE.v:13]
30  INFO: [Synth 8-155] case statement is not full and has no default [D:/study/programming projects/Questasim/SPI_final_project/SPI_SLAVE.v:30]
31  INFO: [Synth 8-6155] done synthesizing module 'SPI_SLAVE' (1#1) [D:/study/programming projects/Questasim/SPI_final_project/SPI_SLAVE.v:1]
```

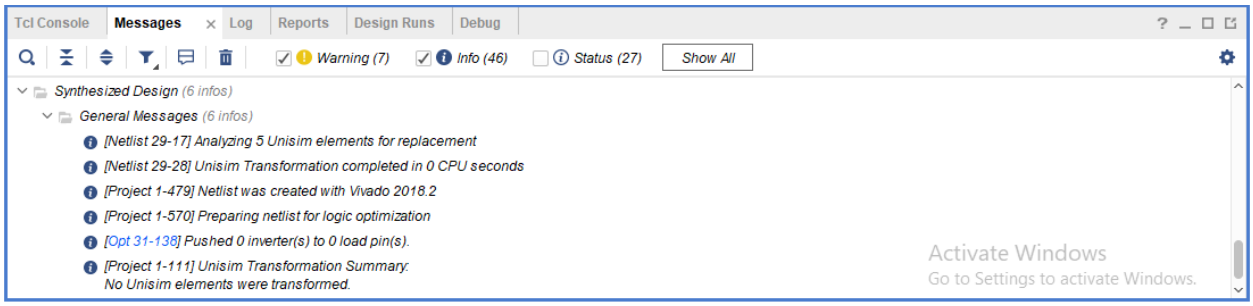
State	New Encoding	Previous Encoding
IDLE	001	000
CHK_CMD	100	001
READ_DATA	010	011
READ_ADD	011	010
WRITE	000	100

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.747 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:	124	Total Number of Endpoints:	124	Total Number of Endpoints:	51

All user specified timing constraints are met.



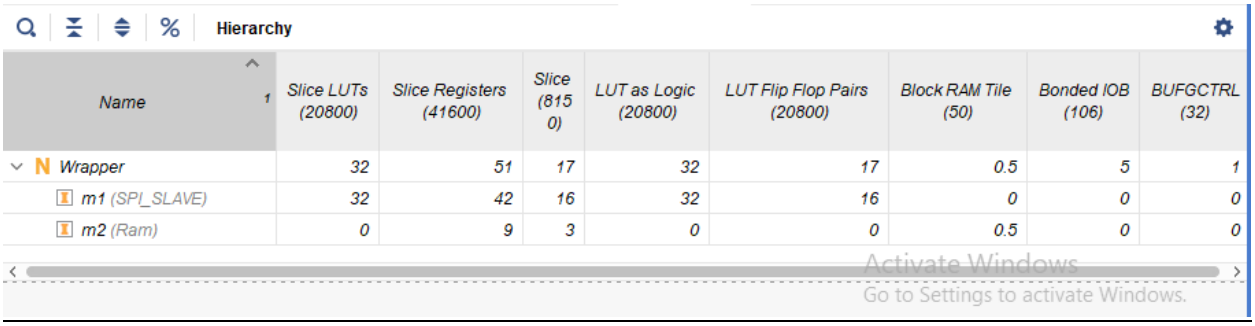
Name	Slice LUTs (20800)	Slice Registers (41600)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
N Wrapper	33	51	0.5	5	1
m1 (SPI_SLAVE)	33	42	0	0	0
m2 (Ram)	0	9	0.5	0	0

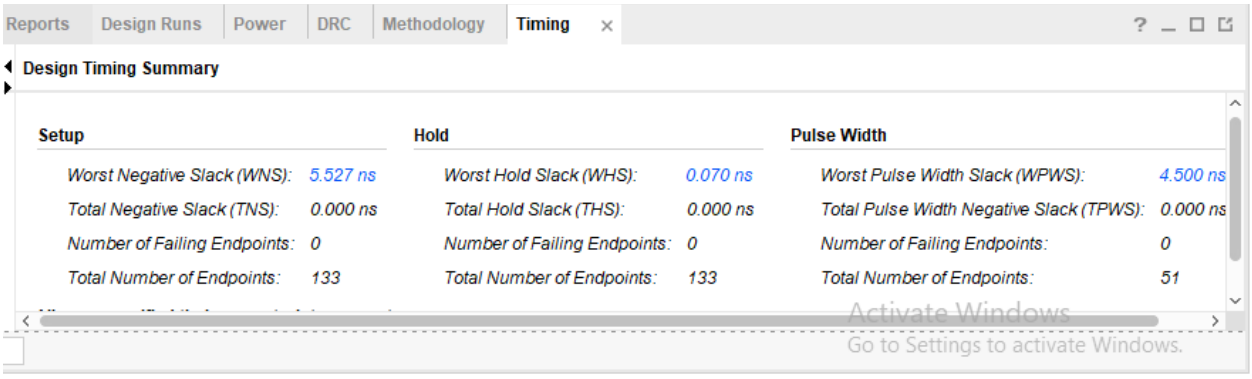


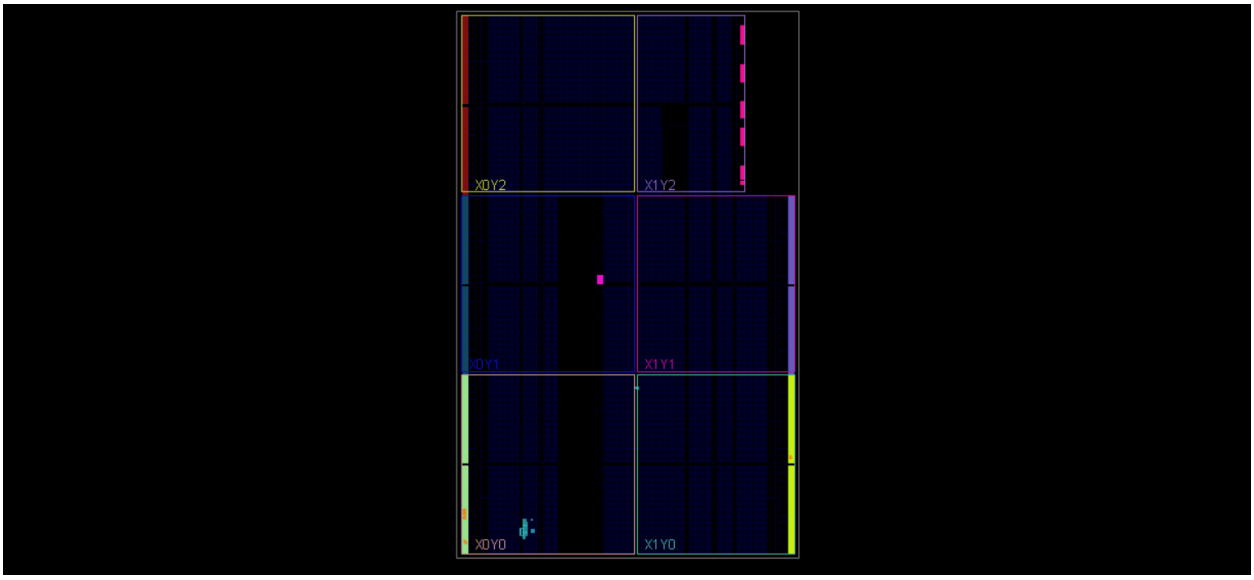
No critical warnings or errors

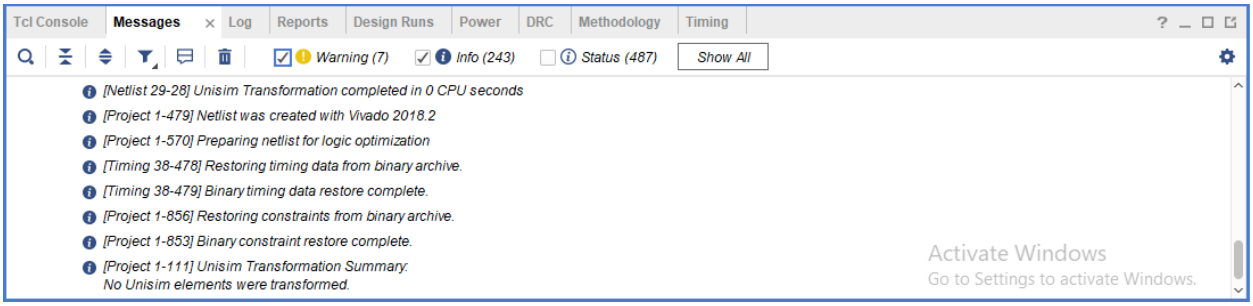
Implementation

Gray encoding

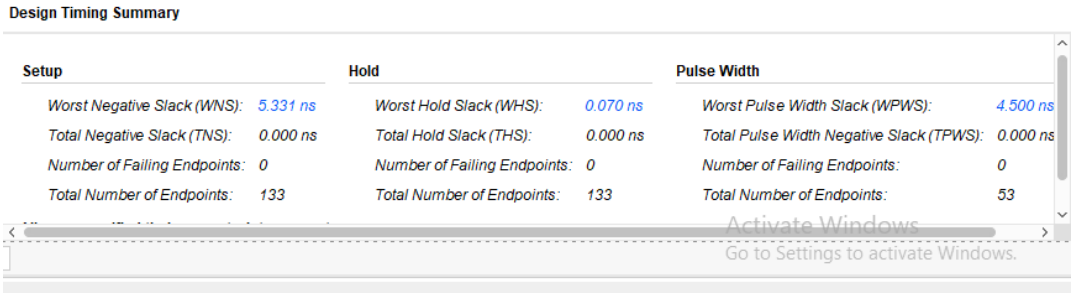




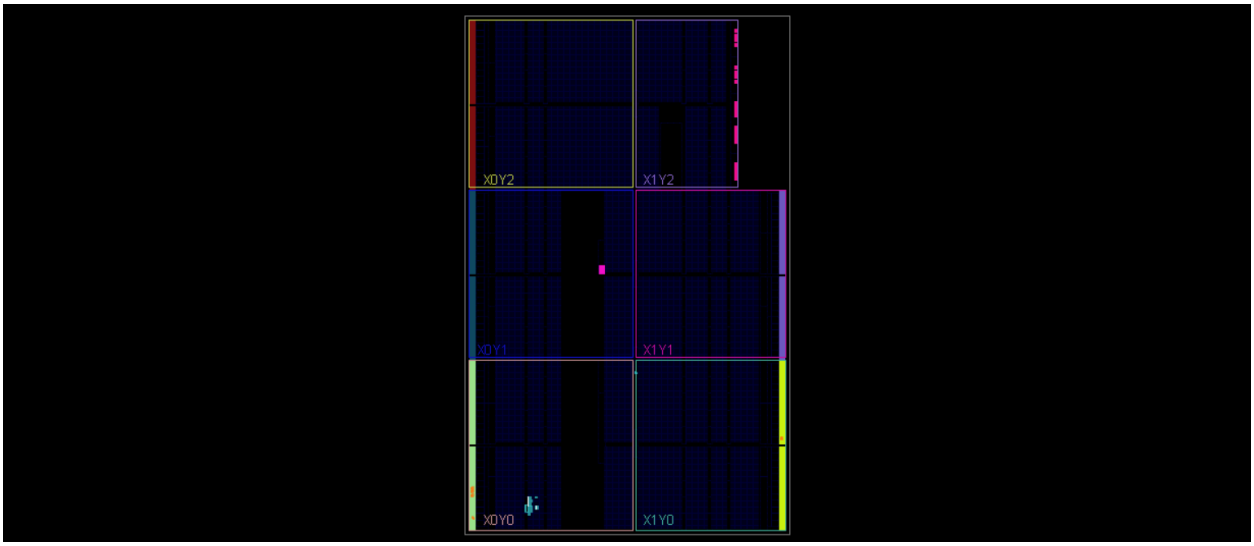




One hot encoding



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)
Wrapper	33	55	17	33	11	0.5	5	1
m1 (SPI_SLAVE)	33	46	15	33	11	0	0	0
m2 (Ram)	0	9	2	0	0	0.5	0	0



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

Warning (7) | Info (243) | Status (489) | Show All

[Synth 8-5534] Detected attribute (* fsm_encoding = "one_hot" *) [SPI_SLAVE.v:12]

[Synth 8-155] case statement is not full and has no default [SPI_SLAVE.v:29]

[Synth 8-567] referenced signal 'MOSI' should be on the sensitivity list [SPI_SLAVE.v:28] (3 more like this)

[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 1-236] Implementation specific constraints were found while reading constraint file [D:/Digital design/SPI_SLAVE_PROJECT/Constraints_basys3_SPI_SLAVE.xdc]. These constraints will be ignored for synthesis but will be used in implementation. Impacted constraints are listed in the file [.Xil/Wrapper_propImpl.xdc].
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 'SPI_SLAVE'

Sequential encoding

Design Timing Summary

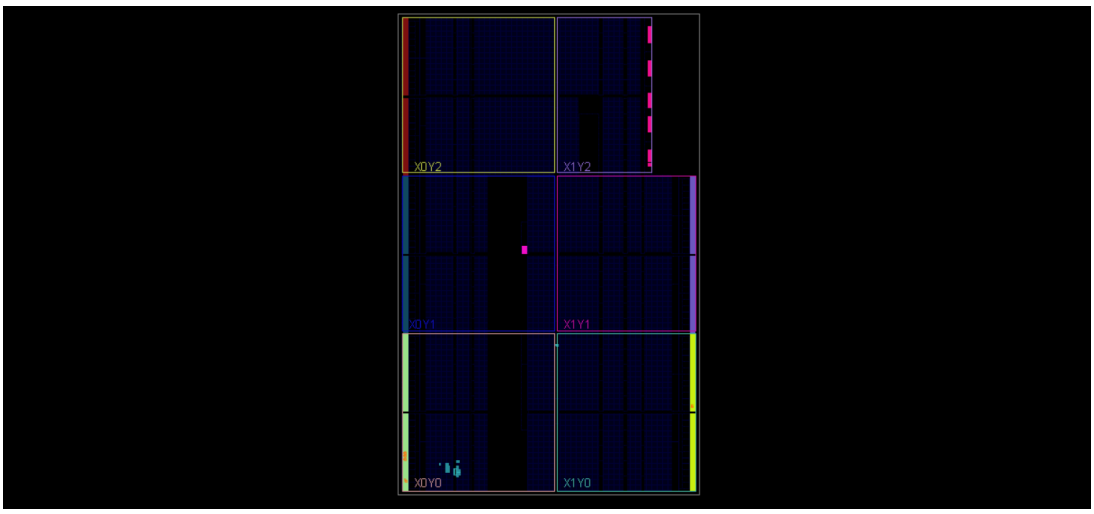
Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	5.527 ns	Worst Hold Slack (WHS):	0.070 ns	Worst Pulse Width Slack (WPWS):	4.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	133	Total Number of Endpoints:	133	Total Number of Endpoints:	51

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 Wrapper	33	51	17	33	15	0.5	5	1
m1 (SPI_SLAVE)	33	42	15	33	15	0	0	0
m2 (Ram)	0	9	3	0	0	0.5	0	0

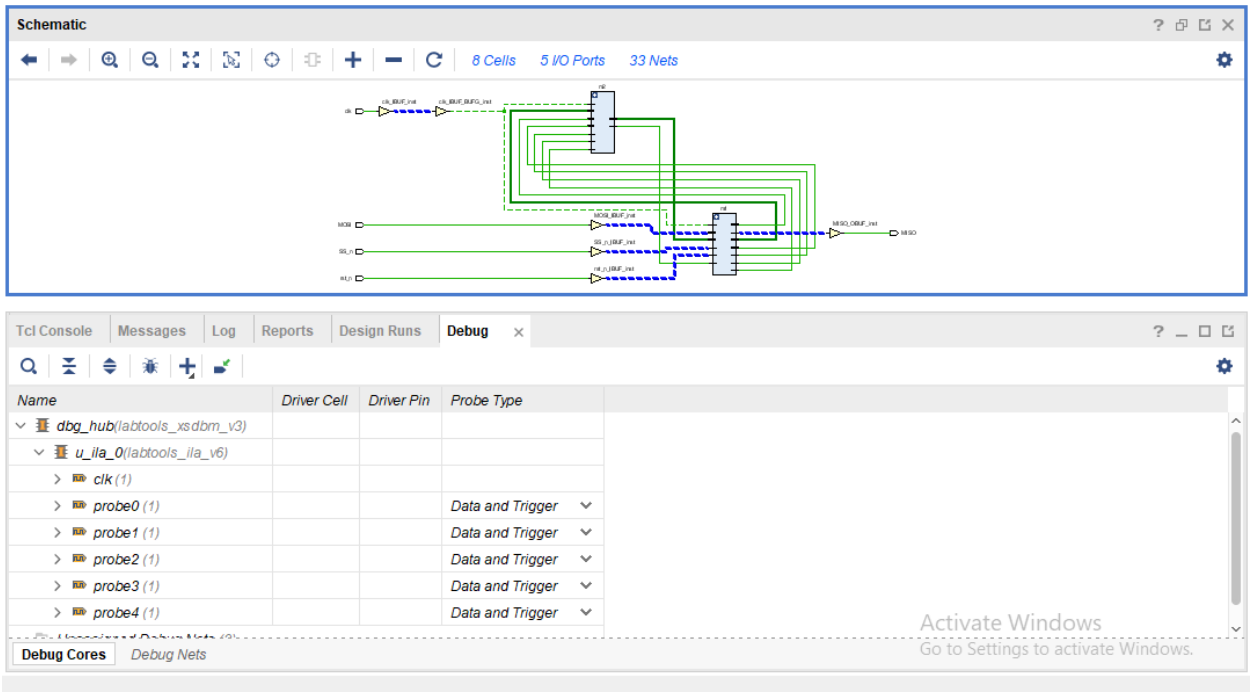
Tcl Console | Messages | Log | Reports | Design Runs | Power | DRC | Methodology | Timing

Warning (7) | Info (241) | Status (479) | Show All

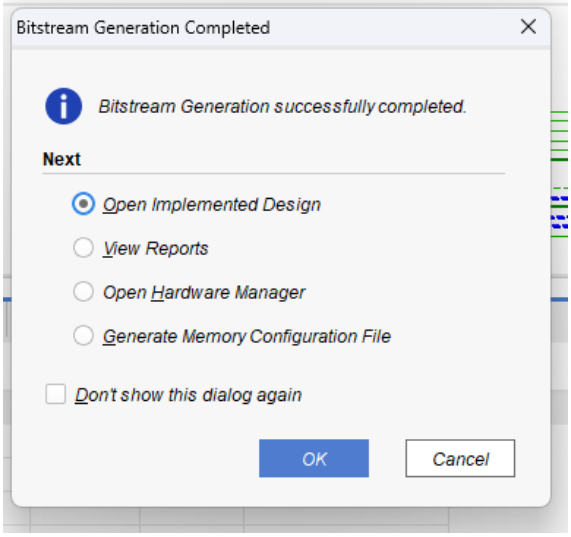
```
[Netlist 29-28] Unisim Transformation completed in 0 CPU seconds
[Project 1-479] Netlist was created with Vivado 2018.2
[Project 1-570] Preparing netlist for logic optimization
[Timing 38-478] Restoring timing data from binary archive.
[Timing 38-479] Binary timing data restore complete.
[Project 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.
```

So the sequential encoding has the best timing so we choose it



We made a setup debug for these nets



Bitstream was created successfully