This project implements an SPI (Serial Peripheral Interface) Slave module designed to communicate with a master device and interface with a single-port RAM. The system uses a finite state machine (FSM) to decode SPI commands and perform read/write operations to memory. A testbench is used to validate functionality and correct MISO output behavior.

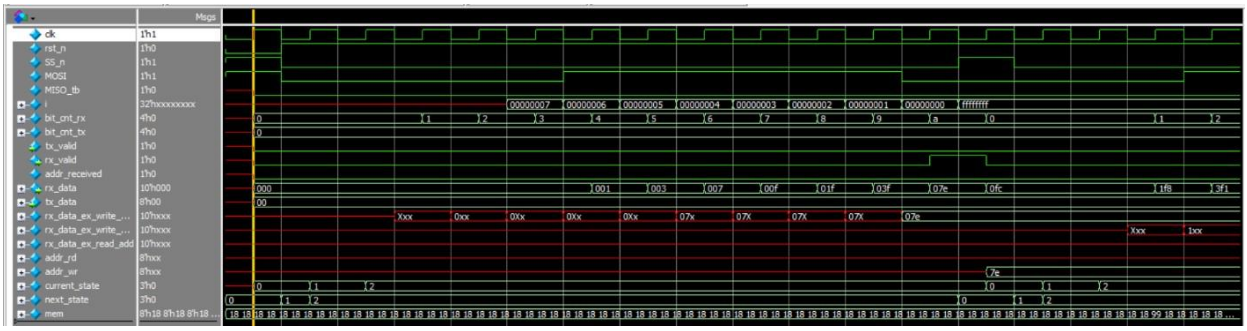
Project 2

SPI Slave with Single Port RAM

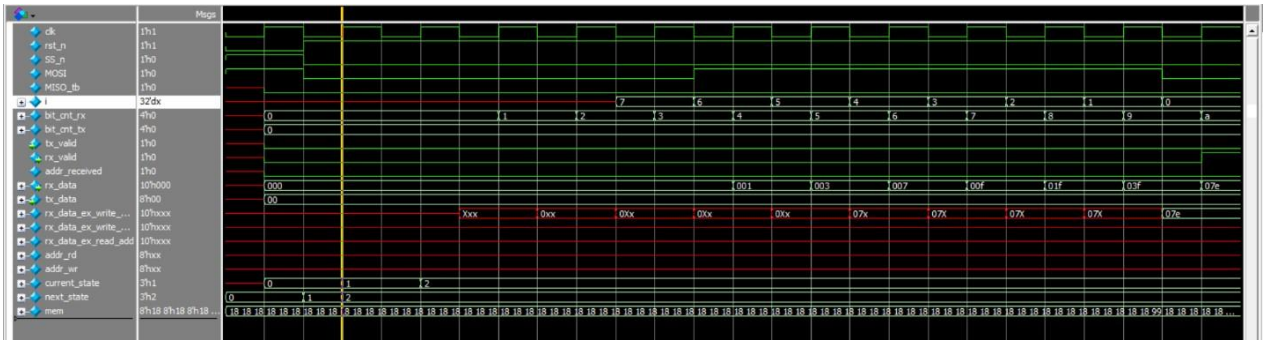
TEAM : SENSITIVITY_LIST
NAMES:
Sherif Ahmed Abdelfatah
Radwa Yasser Ahmed

THE WAVEFORMS SHOWN BELOW ARE GENERATED FROM QUESTA_SIM USING A .DO FILE FOR AUTOMATED SIMULATION SETUP:

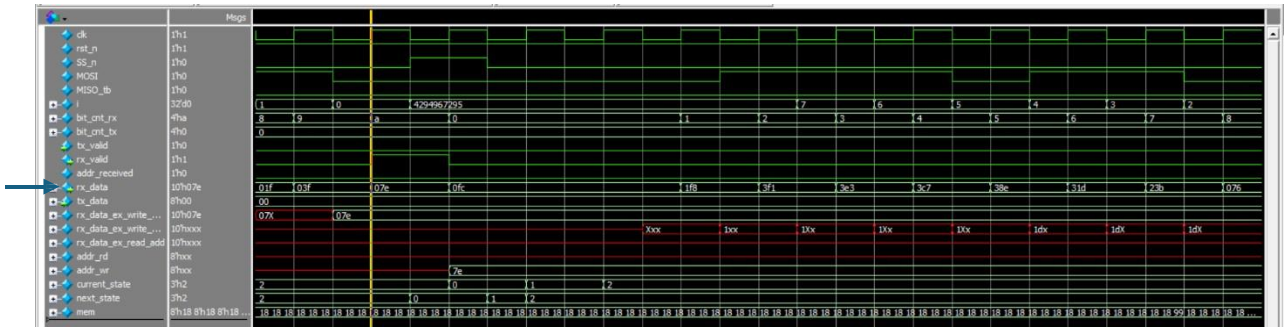
1. **THE SYSTEM IS HELD IN RESET (RST_N = 0), ENSURING ALL INTERNAL STATES ARE INITIALIZED BEFORE STARTING SPI COMMUNICATION.**



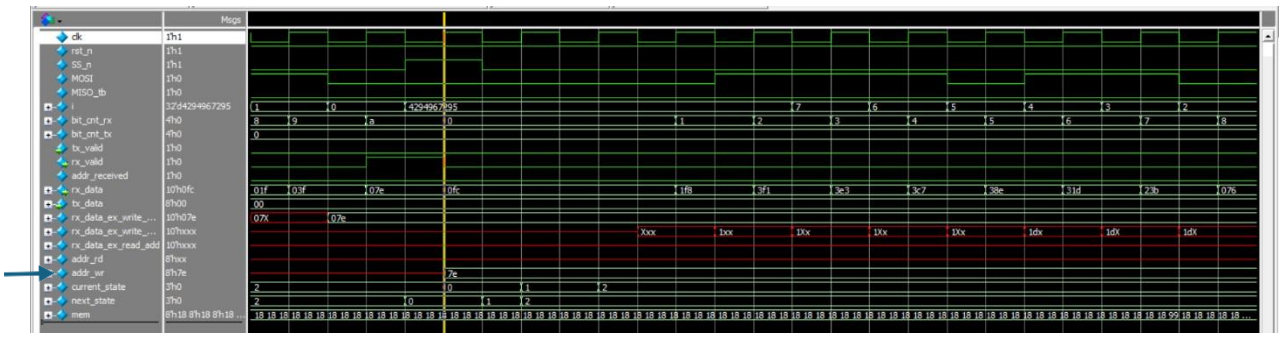
2. **AFTER RESET, THE SPI MODULE TRANSITIONS TO CHK_CMD, RECEIVING THE FIRST MOSI BIT AS 0, CORRECTLY IDENTIFYING THE COMMAND AS A WRITE OPERATION IN THE NEXT CYCLE.**



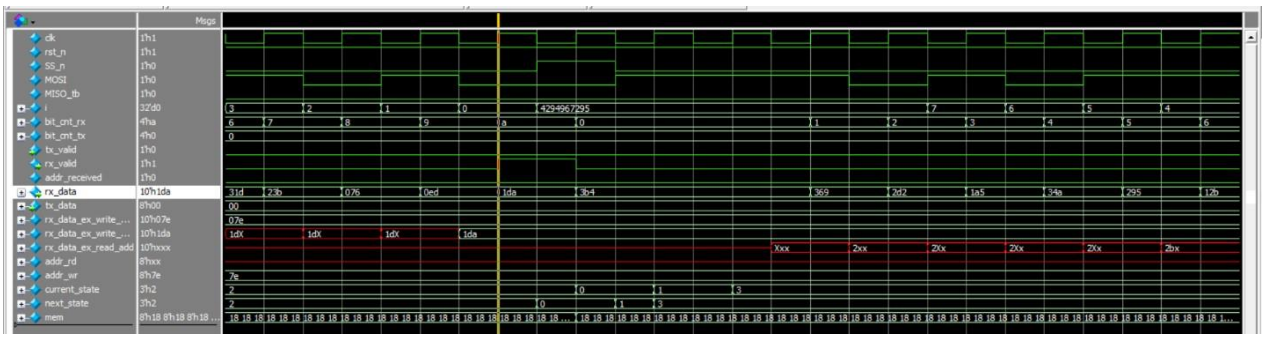
3. **SERIAL INPUT VIA MOSI COMPLETES, FORMING THE FULL 10-BIT RX_DATA WHICH MATCHES THE EXPECTED WRITE ADDRESS.**



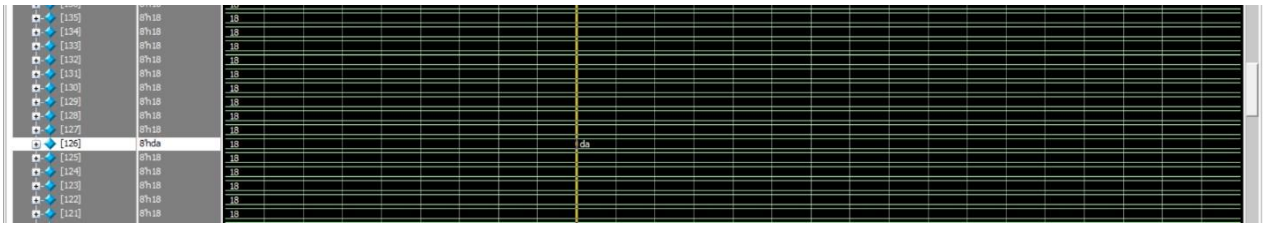
4. AFTER ONE CLOCK CYCLE, THE WRITE ADDRESS IS UPDATED.



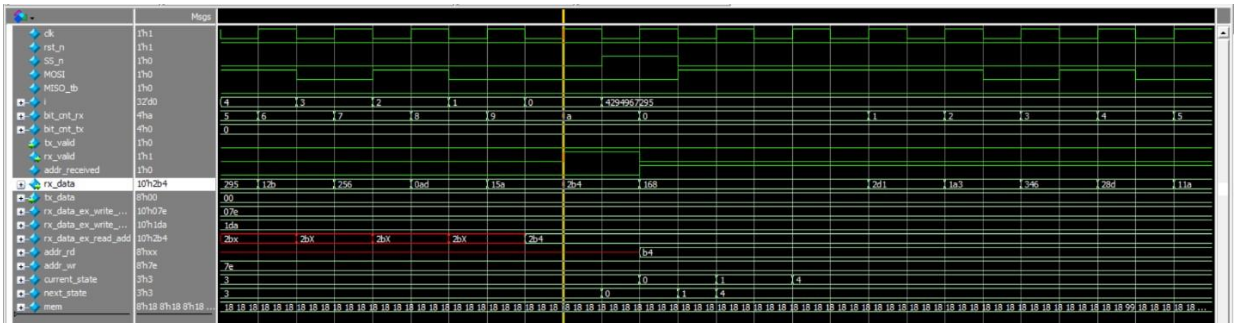
5. ANOTHER SERIAL MOSI TRANSFER FORMS THE DATA TO BE WRITTEN. ONCE RECEIVED, RX_DATA MATCHES THE EXPECTED VALUE THIS TIME THE LAST TWO BIT IS ONE SO IT'S WRITE DATA.



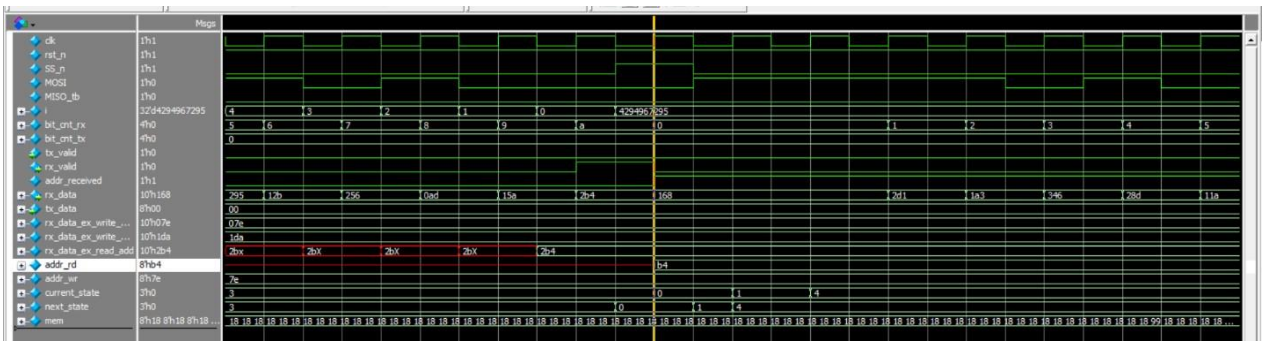
6. AFTER A CLOCK CYCLE, MEMORY AT THE PREVIOUSLY RECEIVED WRITE ADDRESS (8'H7E=8'D126) IS UPDATED WITH THE NEW DATA .



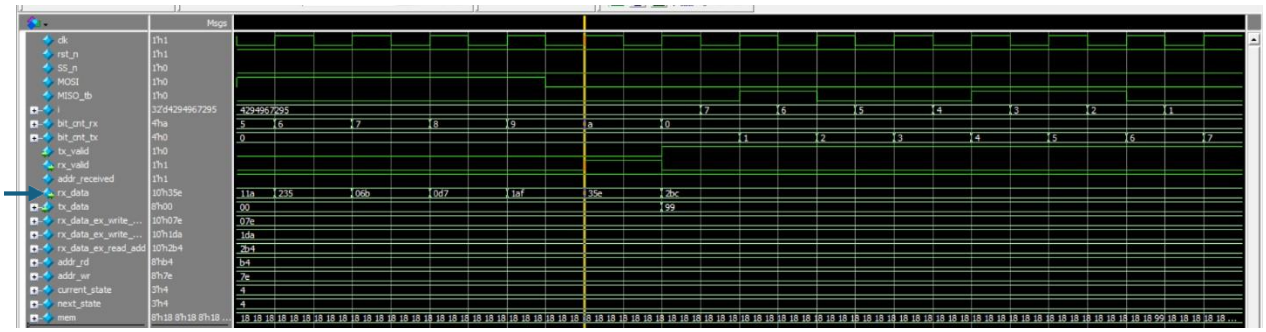
7. A 10-BIT READ ADDRESS IS RECEIVED SERIALLY THROUGH MOSI AND STORED IN RX_DATA.



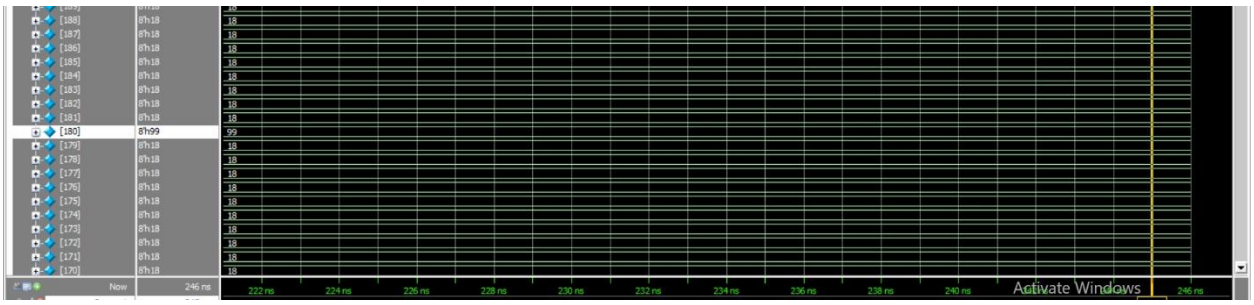
8. AFTER A CLOCK CYCLE, THE READ ADDRESS IS UPDATED .

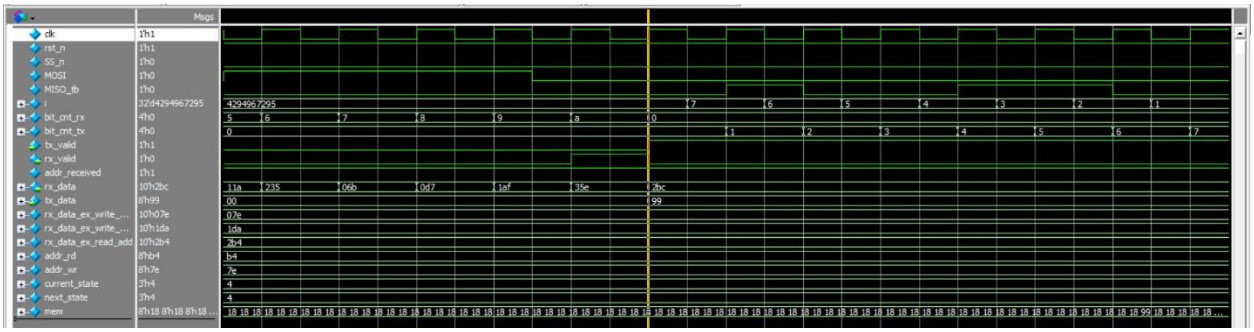


G. ANOTHER SERIAL INPUT MOSI IS RECEIVED THE FIRST 2 BITS IS 3 SO THAT MEANS WE WILL READ FROM THE MEM.

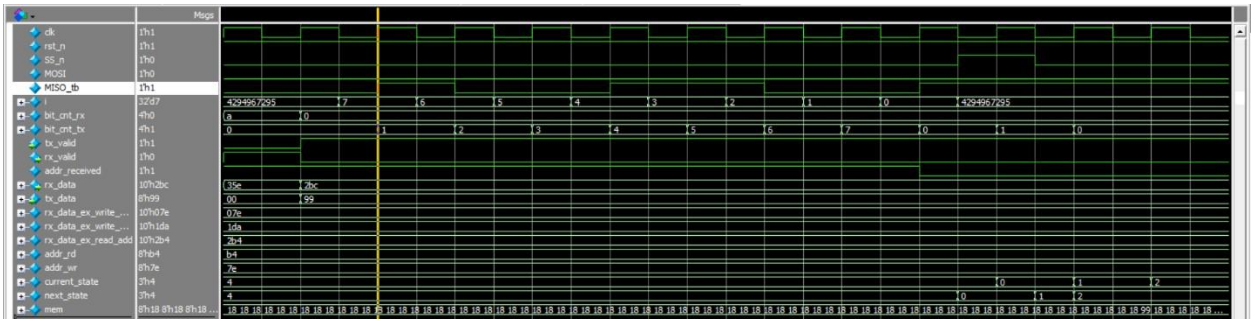


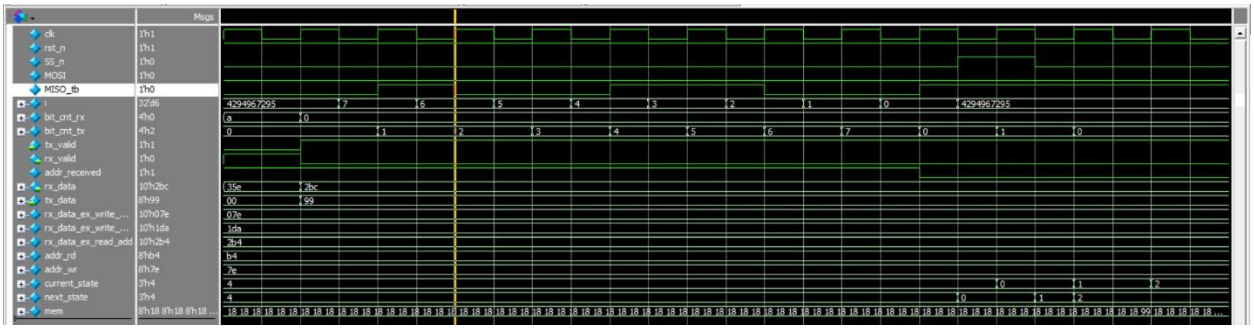
10. AFTER A CLOCK CYCLE, THE CONTENT OF MEMORY AT THE PREVIOUSLY RECEIVED READ ADDRESS (8'HB4=8'D180) UPDATES TX_DATA.

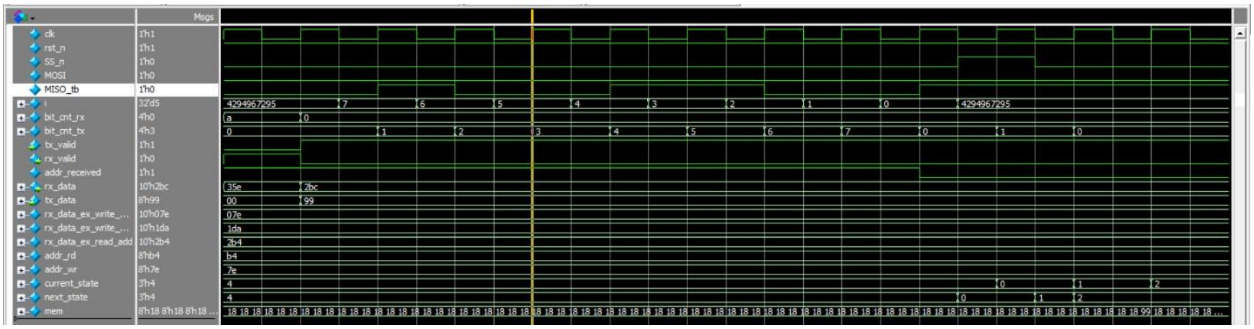


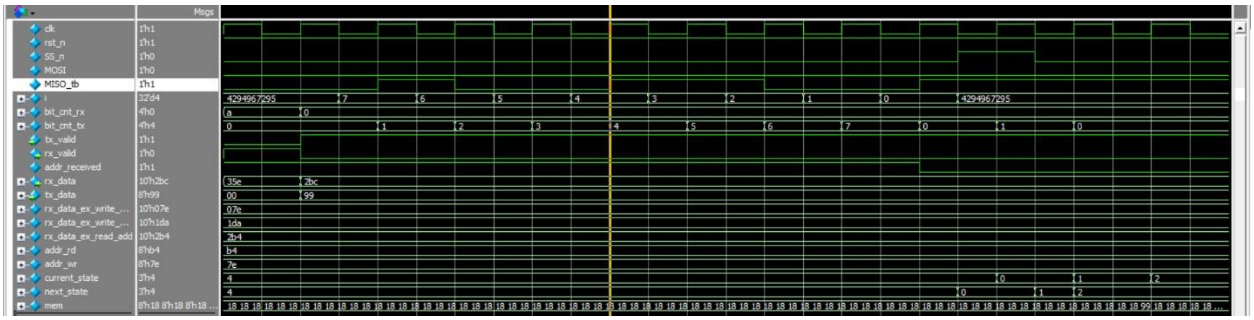


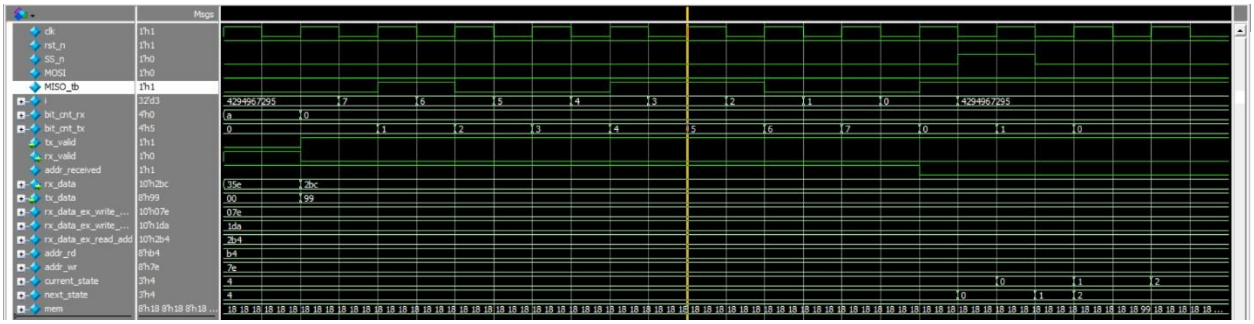
11. BITS ARE THEN SHIFTED OUT SERIALLY THROUGH **MISO**, MATCHING THE EXPECTED DATA BIT-BY-BIT. (8'HGG=8'B10011001).

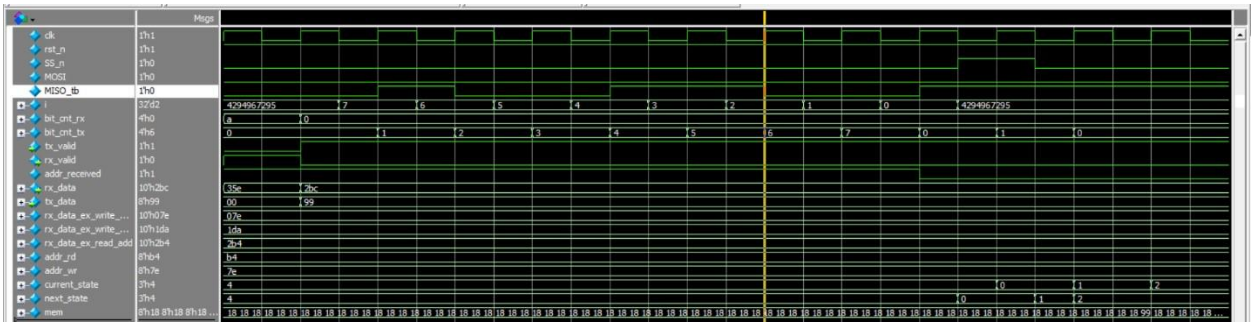


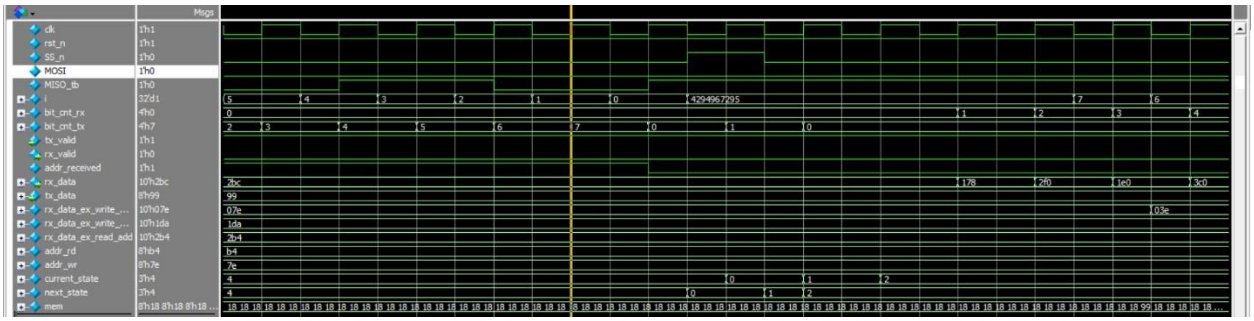


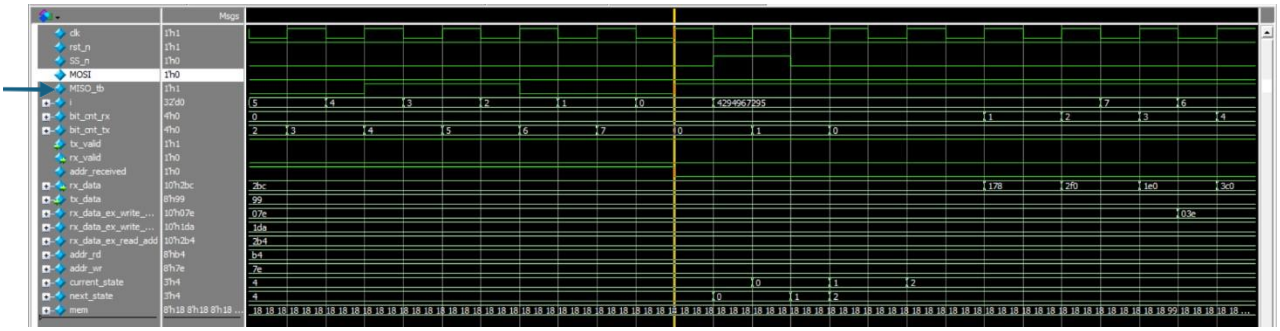




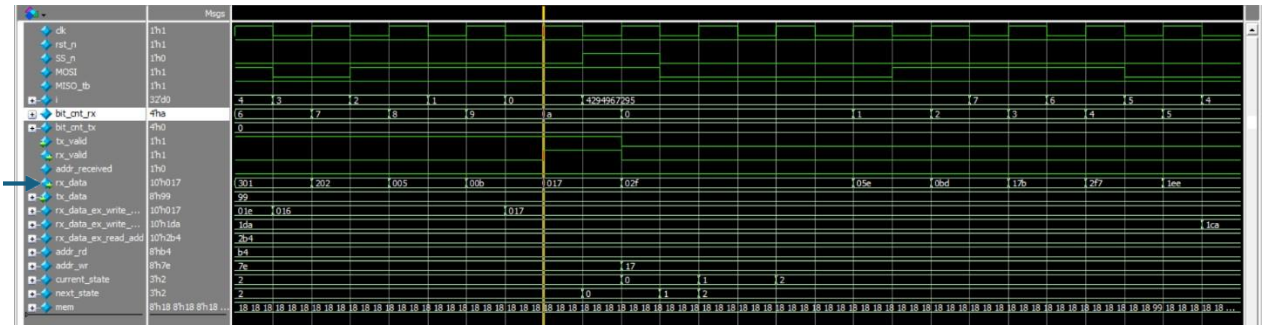

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h0
MISO_tb 1'h1
i 32'd3
bit_cnt_rx 4'h0
bit_cnt_tx 4'h5
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h2bc
tx_data 8'h99
rx_data_ex_write_... 10'h07e
rx_data_ex_write_... 10'h1da
rx_data_ex_read_add 10'h2b4
addr_rd 8'hb4
addr_wr 8'h7e
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

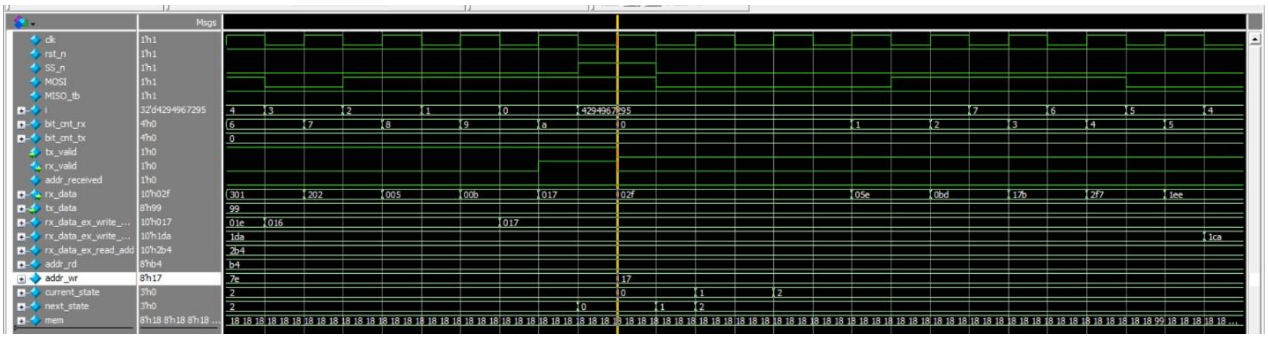

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h0
MISO_tb 1'h0
i 32'd2
bit_cnt_rx 4'h0
bit_cnt_tx 4'h6
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h2bc
tx_data 8'h99
rx_data_ex_write_... 10'h07e
rx_data_ex_write_... 10'h1da
rx_data_ex_read_add 10'h2b4
addr_rd 8'hb4
addr_wr 8'h7e
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

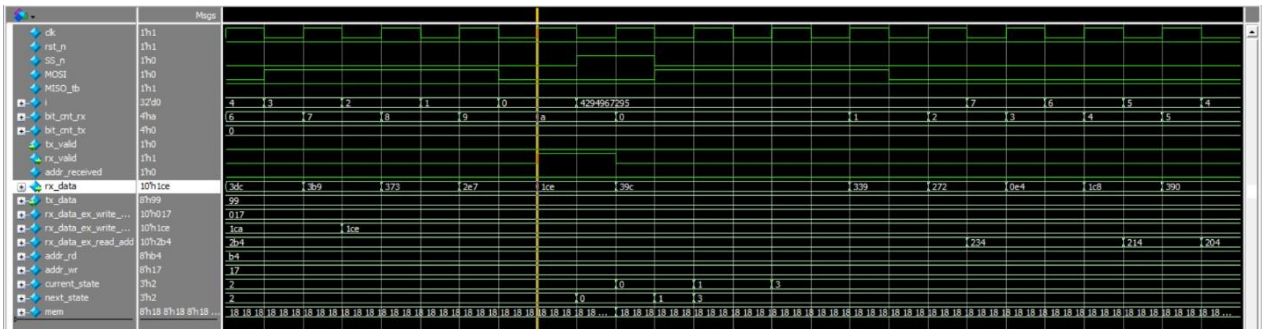

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h0
MISO_tb 1'h0
i 32'd1
bit_cnt_rx 4'h0
bit_cnt_tx 4'h7
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h2bc
tx_data 8'h99
rx_data_ex_write_... 10'h07e
rx_data_ex_write_... 10'h1da
rx_data_ex_read_add 10'h2b4
addr_rd 8'hb4
addr_wr 8'h7e
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

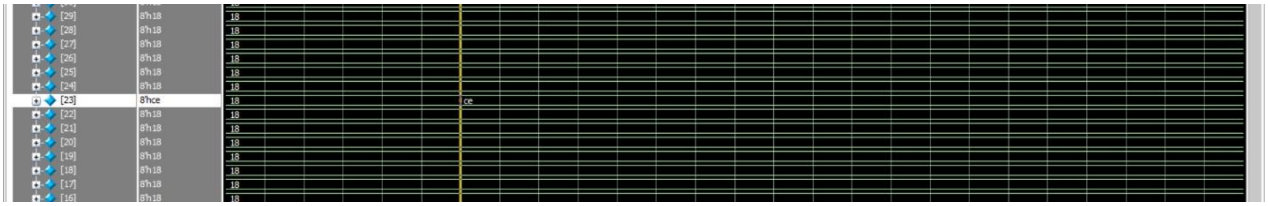

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h0
MISO_tb 1'h1
i 32'd0
bit_cnt_rx 4'h0
bit_cnt_tx 4'h0
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h0
rx_data 10'h2bc
tx_data 8'h99
rx_data_ex_write_... 10'h07e
rx_data_ex_write_... 10'h1da
rx_data_ex_read_add 10'h2b4
addr_rd 8'hb4
addr_wr 8'h7e
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

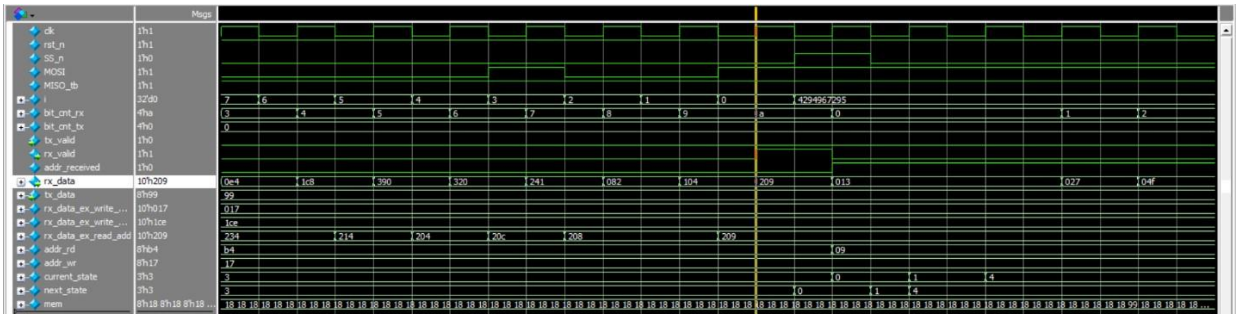
THAT'S THE SAME FUNCTIONS BUT WITH DIFFERENT INPUTS .

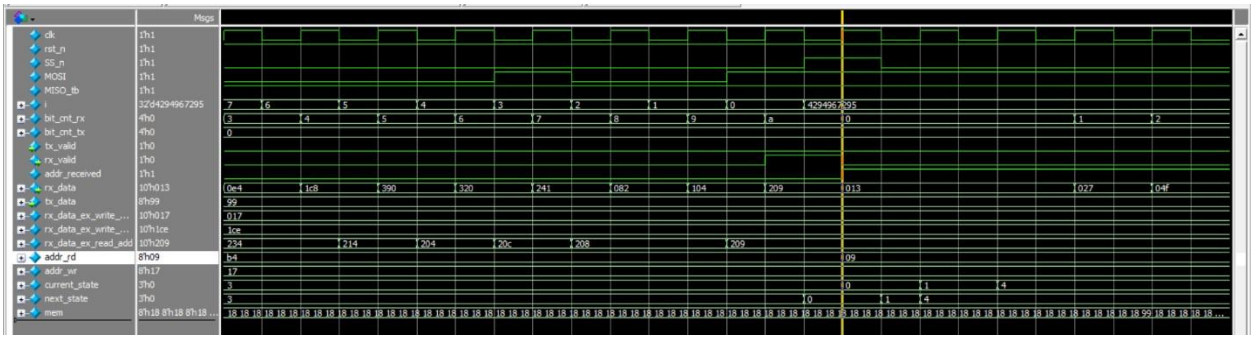


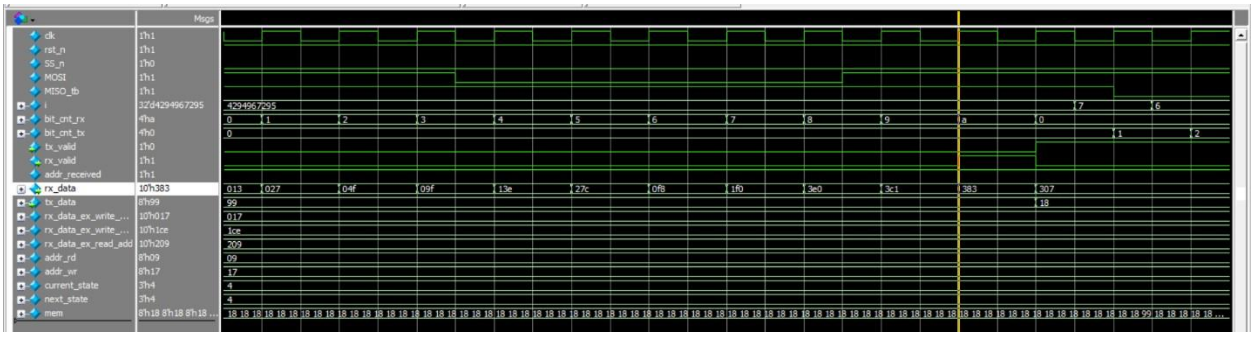




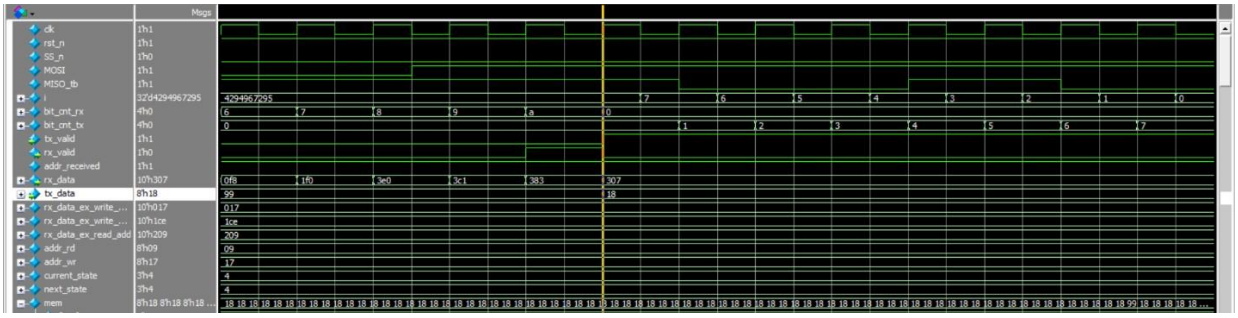


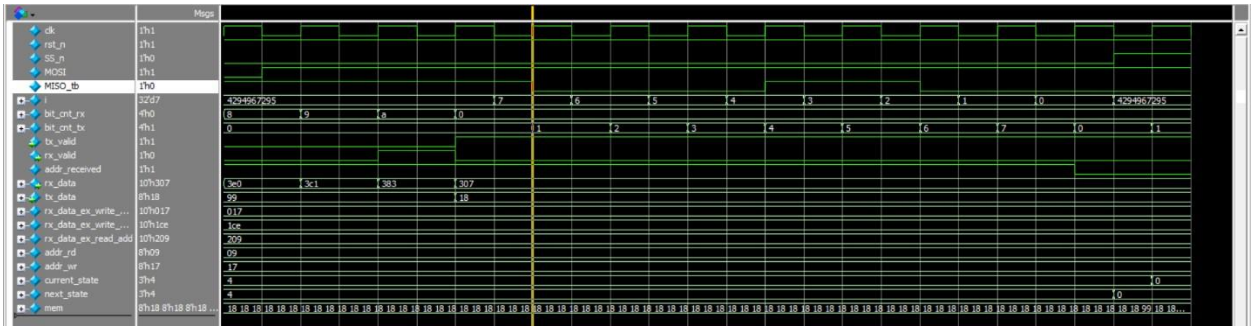


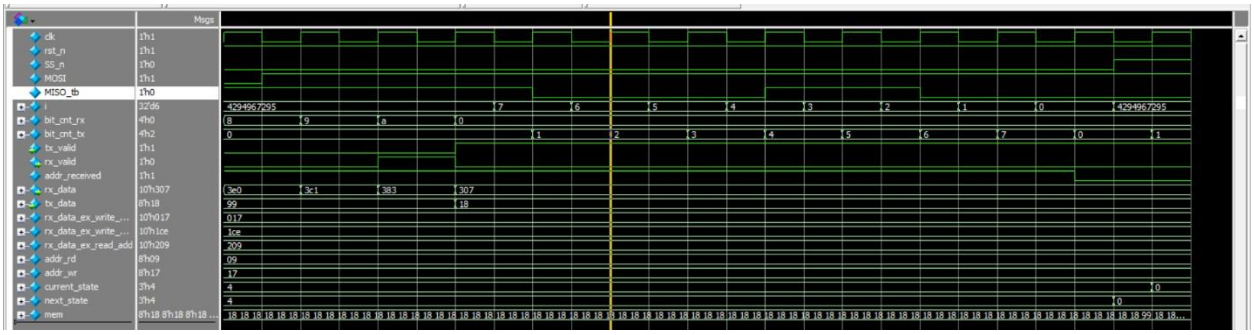

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h1
MOSI 1'h1
MISO_tb 1'h1
i 32'd4294967295
bit_cnt_rx 4'h0
bit_cnt_tx 4'h0
tx_valid 1'h0
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h013
tx_data 8'h99
rx_data_ex_write_...
rx_data_ex_write_...
rx_data_ex_read_add
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h0
next_state 3'h0
mem 8'h18 8'h18 8'h18 ...

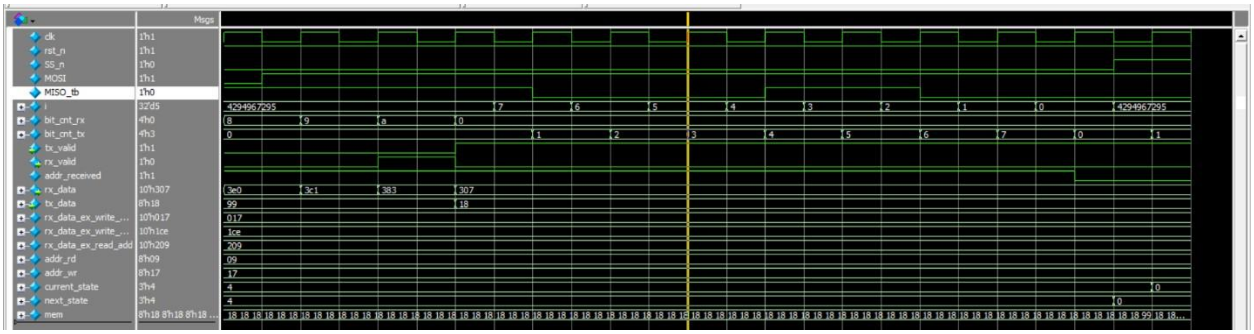

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h1
MISO_tb 1'h1
i 32'd4294967295
bit_cnt_rx 4'ha
bit_cnt_tx 4'h0
tx_valid 1'h0
rx_valid 1'h1
addr_received 1'h1
rx_data 10'h383
tx_data 8'h99
rx_data_ex_write_...
rx_data_ex_write_...
rx_data_ex_read_add
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

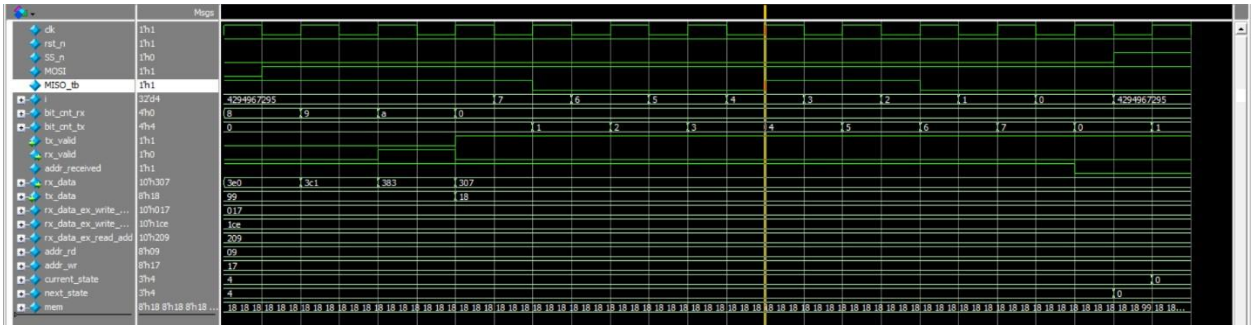
[17] 8'h18
[16] 8'h18
[15] 8'h18
[14] 8'h18
[13] 8'h18
[12] 8'h18
[11] 8'h18
[10] 8'h18
[9] 8'h18
[8] 8'h18
[7] 8'h18
[6] 8'h18
[5] 8'h18
[4] 8'h12
[3] 8'h12
[2] 8'h12
[1] 8'h12
[0] 8'h12

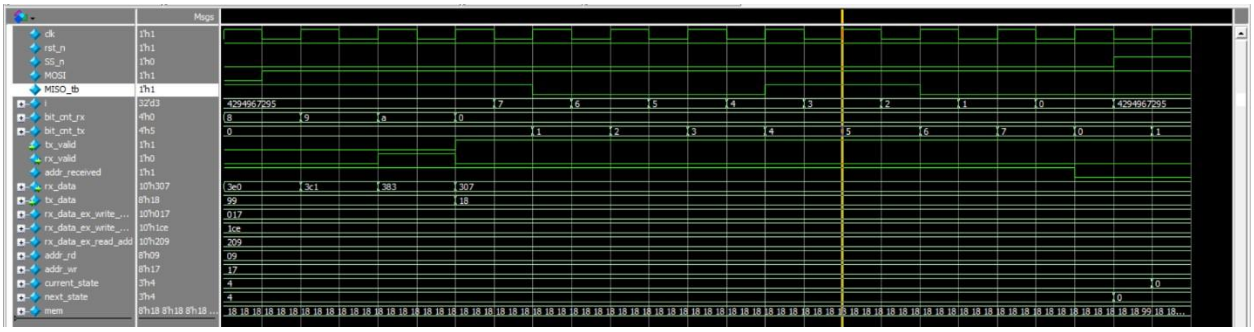

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h1
MISO_tb 1'h1
i 32'd4294967295
bit_cnt_rx 4'h0
bit_cnt_tx 4'h0
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h307
tx_data 8'h18
rx_data_ex_write_...
rx_data_ex_write_...
rx_data_ex_read_add
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

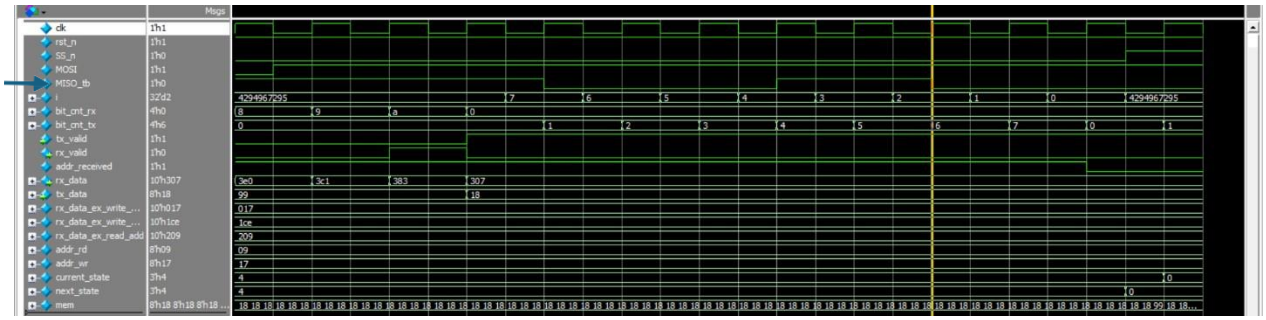

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h1
MISO_tb 1'h0
i 32'd7
bit_cnt_rx 4'h0
bit_cnt_tx 4'h1
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h307
tx_data 8'h18
rx_data_ex_write_... 10'h017
rx_data_ex_write_... 10'h1ce
rx_data_ex_read_add 10'h209
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

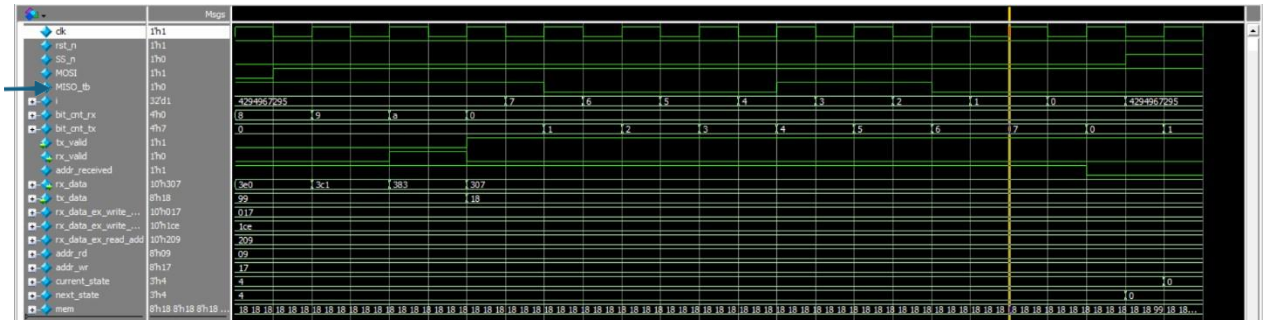

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h1
MISO_tb 1'h0
i 32'd6
bit_cnt_rx 4'h0
bit_cnt_tx 4'h2
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h307
tx_data 8'h18
rx_data_ex_write_... 10'h017
rx_data_ex_write_... 10'h1ce
rx_data_ex_read_add 10'h209
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

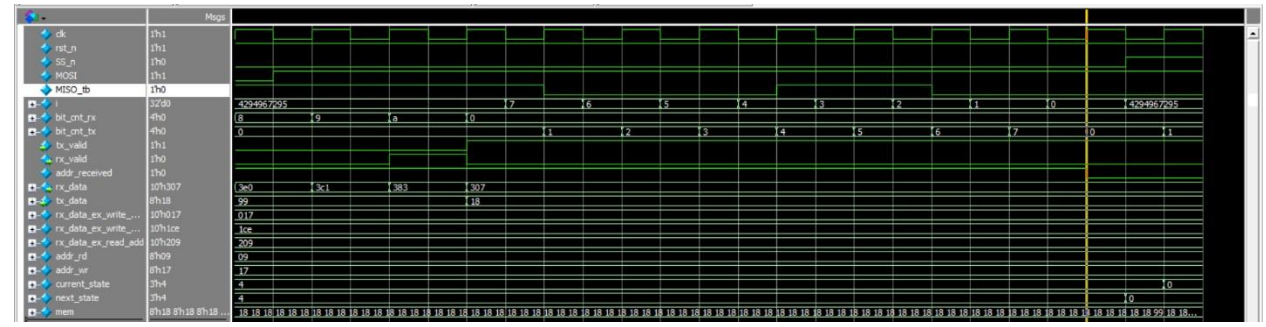

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h1
MISO_tb 1'h0
i 32'd5
bit_cnt_rx 4'h0
bit_cnt_tx 4'h3
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h307
tx_data 8'h18
rx_data_ex_write_... 10'h017
rx_data_ex_write_... 10'h1ce
rx_data_ex_read_add 10'h209
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...


Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h1
MISO_tb 1'h1
i 32'd4
bit_cnt_rx 4'h0
bit_cnt_tx 4'h4
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h307
tx_data 8'h18
rx_data_ex_write_... 10'h017
rx_data_ex_write_... 10'h1ce
rx_data_ex_read_add 10'h209
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

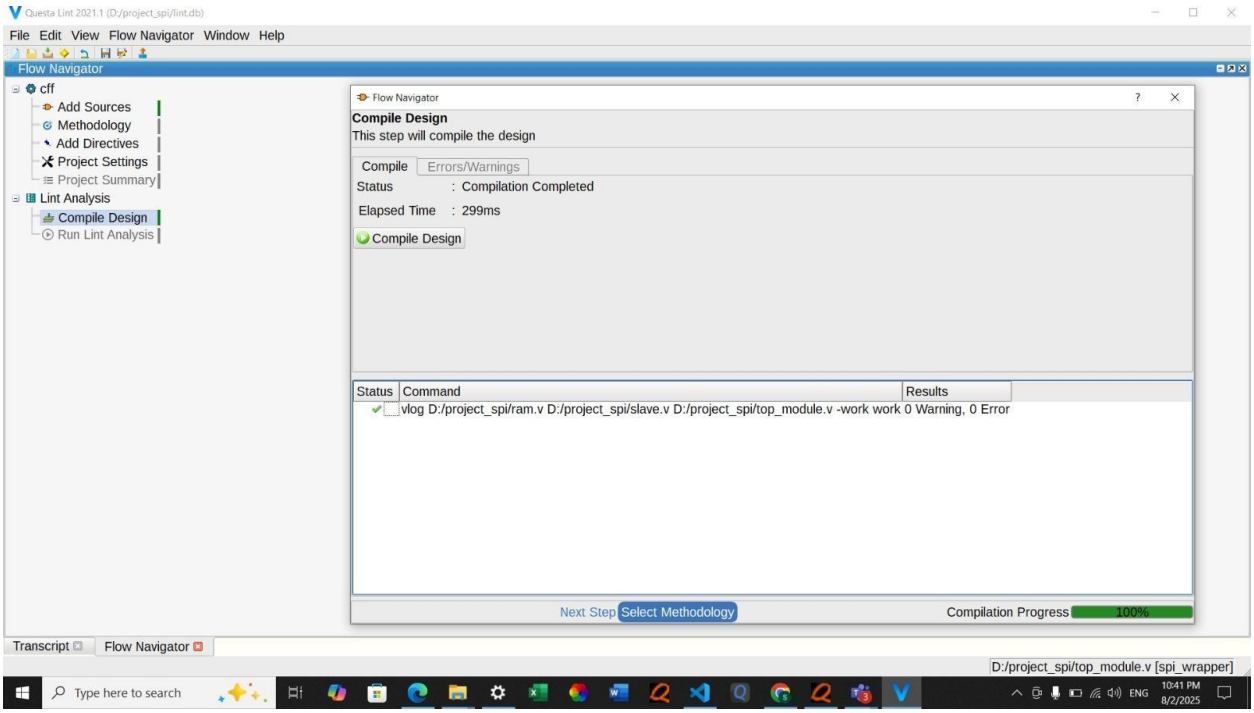

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h1
MISO_tb 1'h1
i 32'd3
bit_cnt_rx 4'h0
bit_cnt_tx 4'h5
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h307
tx_data 8'h18
rx_data_ex_write_... 10'h017
rx_data_ex_write_... 10'h1ce
rx_data_ex_read_add 10'h209
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

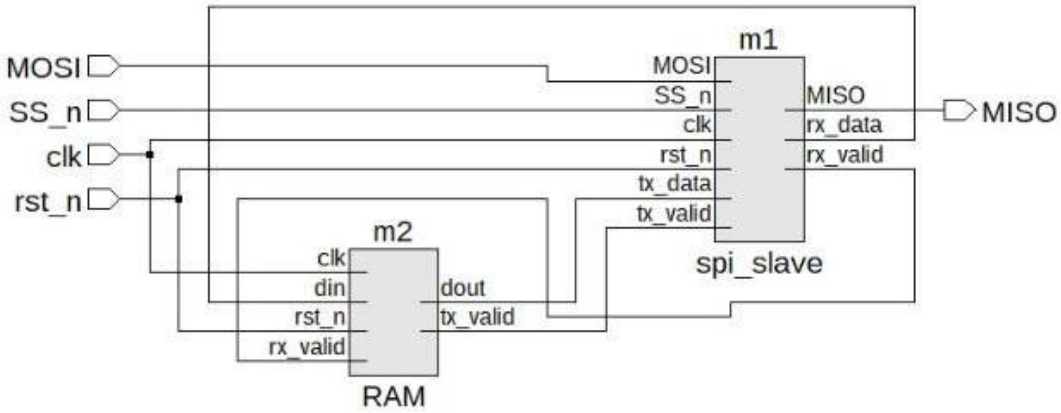

Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h1
MISO_tb 1'h0
i 32'd2
bit_cnt_rx 4'h0
bit_cnt_tx 4'h6
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h307
tx_data 8'h18
rx_data_ex_write_... 10'h017
rx_data_ex_write_... 10'h1ce
rx_data_ex_read_add 10'h209
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...


Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h1
MISO_tb 1'h0
i 32'd1
bit_cnt_rx 4'h0
bit_cnt_tx 4'h7
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h1
rx_data 10'h307
tx_data 8'h18
rx_data_ex_write_... 10'h017
rx_data_ex_write_... 10'h1ce
rx_data_ex_read_add 10'h209
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...


Msgs
clk 1'h1
rst_n 1'h1
SS_n 1'h0
MOSI 1'h1
MISO_tb 1'h0
i 32'd0
bit_cnt_rx 4'h0
bit_cnt_tx 4'h0
tx_valid 1'h1
rx_valid 1'h0
addr_received 1'h0
rx_data 10'h307
tx_data 8'h18
rx_data_ex_write_... 10'h017
rx_data_ex_write_... 10'h1ce
rx_data_ex_read_add 10'h209
addr_rd 8'h09
addr_wr 8'h17
current_state 3'h4
next_state 3'h4
mem 8'h18 8'h18 8'h18 ...

THE IMAGES BELOW SHOW THE RESULTS OF THE LINTING TOOL WITH NO DETECTED ERRORS, AND THE SUCCESSFUL SYNTHESIS OF THE DESIGN:

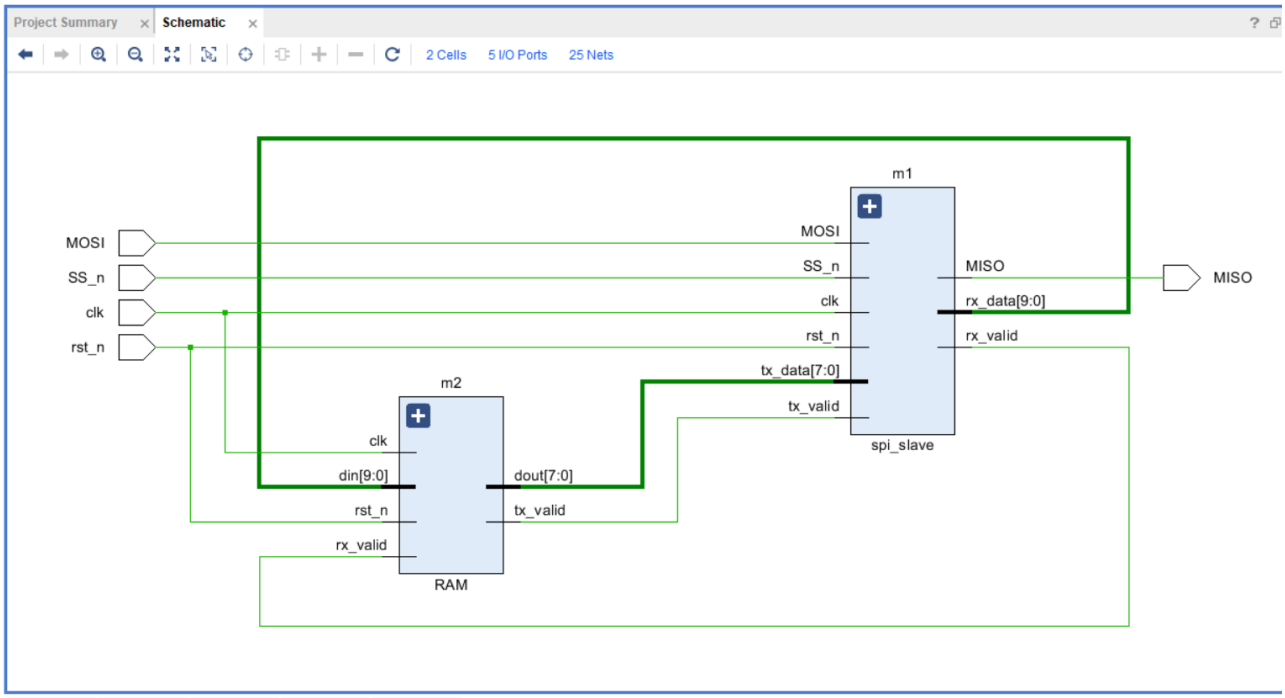




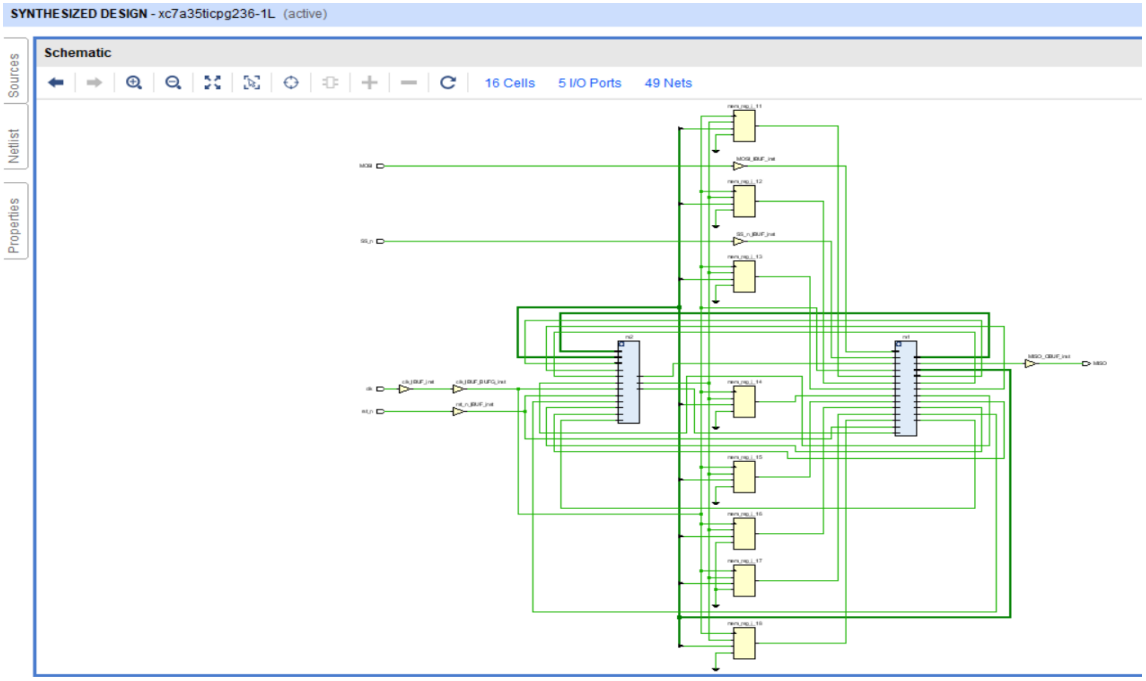
VIVADO

1-SCHEMATIC AND SYNTHESIS

SCHEMATIC AFTER THE ELABORATION



SCHEMATIC AFTER THE SYNTHESIS



Synthesis report showing the encoding used

Info : in line 2G synthesis detect (*fsm_ecoding=gray*)

Schematic | synth_1_synth_synthesis_report_0 - synth_1

D:/project_spi/project_SPI/project_SPI.runs/synth_1/spi_wrapper.vds

Read-only

```
INFO: [Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a35ti'
INFO: Launching helper process for spawning children vivado processes
INFO: Helper process launched with PID 8180
---------------------------------------------------------------------------------
Starting RTL Elaboration : Time (s): cpu = 00:00:02 ; elapsed = 00:00:02 . Memory (MB): peak = 409.445 ; gain = 97.855
---------------------------------------------------------------------------------
INFO: [Synth 8-6157] synthesizing module 'spi_wrapper' [D:/project_spi/top_module.v:1]
INFO: [Synth 8-6157] synthesizing module 'spi_slave' [D:/project_spi/slave.v:1]
    Parameter IDLE bound to: 3'b000
    Parameter CHK_CMD bound to: 3'b001
    Parameter WRITE bound to: 3'b010
    Parameter READ_ADD bound to: 3'b011
    Parameter READ_DATA bound to: 3'b100
INFO: [Synth 8-5534] Detected attribute (* fsm_encoding = "gray" *) [D:/project_spi/slave.v:22]
INFO: [Synth 8-6155] done synthesizing module 'spi_slave' (1#1) [D:/project_spi/slave.v:1]
INFO: [Synth 8-6157] synthesizing module 'RAM' [D:/project_spi/ram.v:1]
    Parameter MEM_DEPTH bound to: 256 - type: integer
```

UTILIZATION REPORT, TIMING REPORT

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

Hierarchy

Hierarchy
Summary
Slice Logic
Slice LUTs (<1%)
LUT as Logic (<1%)
Slice Registers (<1%)
Register as Flip Flop

Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
spi_wrapper	33	40	1	0.5	5	1
m1 (spi_slave)	29	23	0	0	0	0
m2 (RAM)	4	9	1	0.5	0	0

utilization_1

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

Design Timing Summary

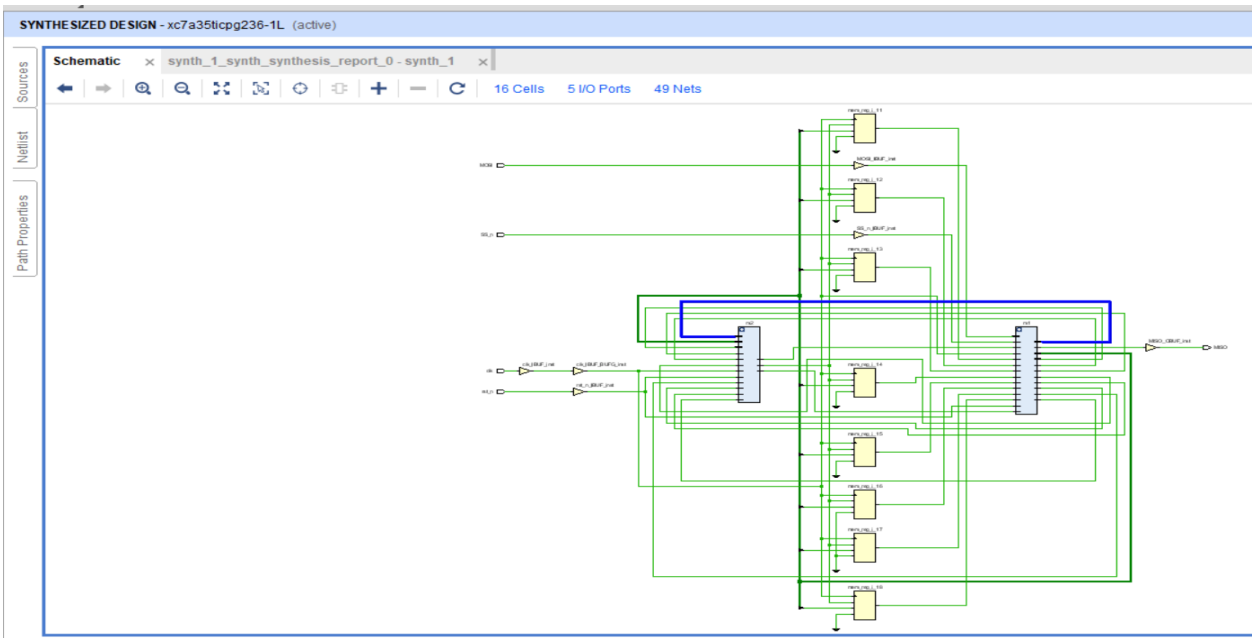
General Information
Timer Settings
Design Timing Summary
Clock Summary (1)
Check Timing (4)
Intra-Clock Paths
Inter-Clock Paths
Other Path Groups

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	6.710 ns	Worst Hold Slack (WHS):	0.146 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:	97	Total Number of Endpoints:	97	Total Number of Endpoints:	43

All user specified timing constraints are met.

Timing Summary - timing_1

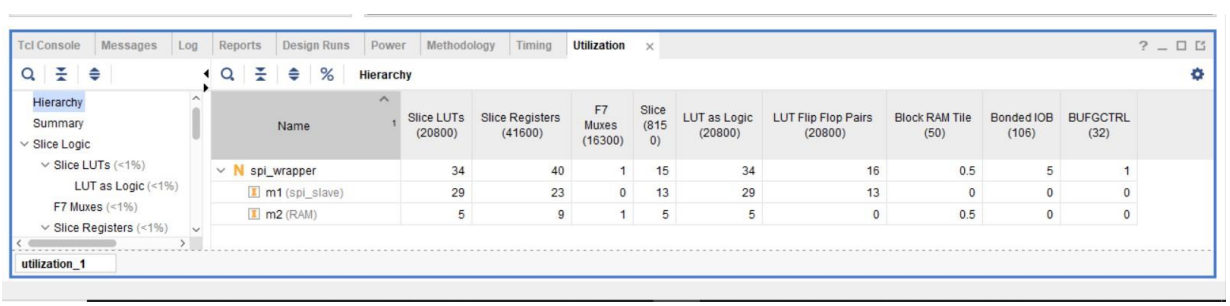
SNIPPET OF THE CRITICAL PATH HIGHLIGHTED IN THE SCHEMATIC



2- IMPLEMENTATION SNIPPETS FOR EACH ENCODING

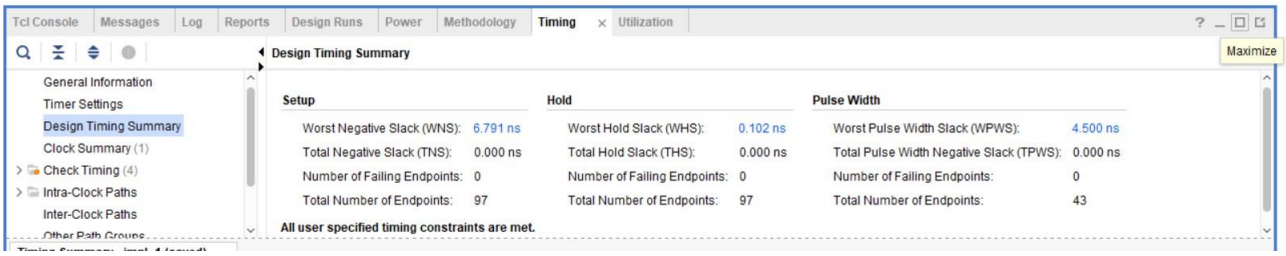
Let's start with gray_encoding

UTILIZATION

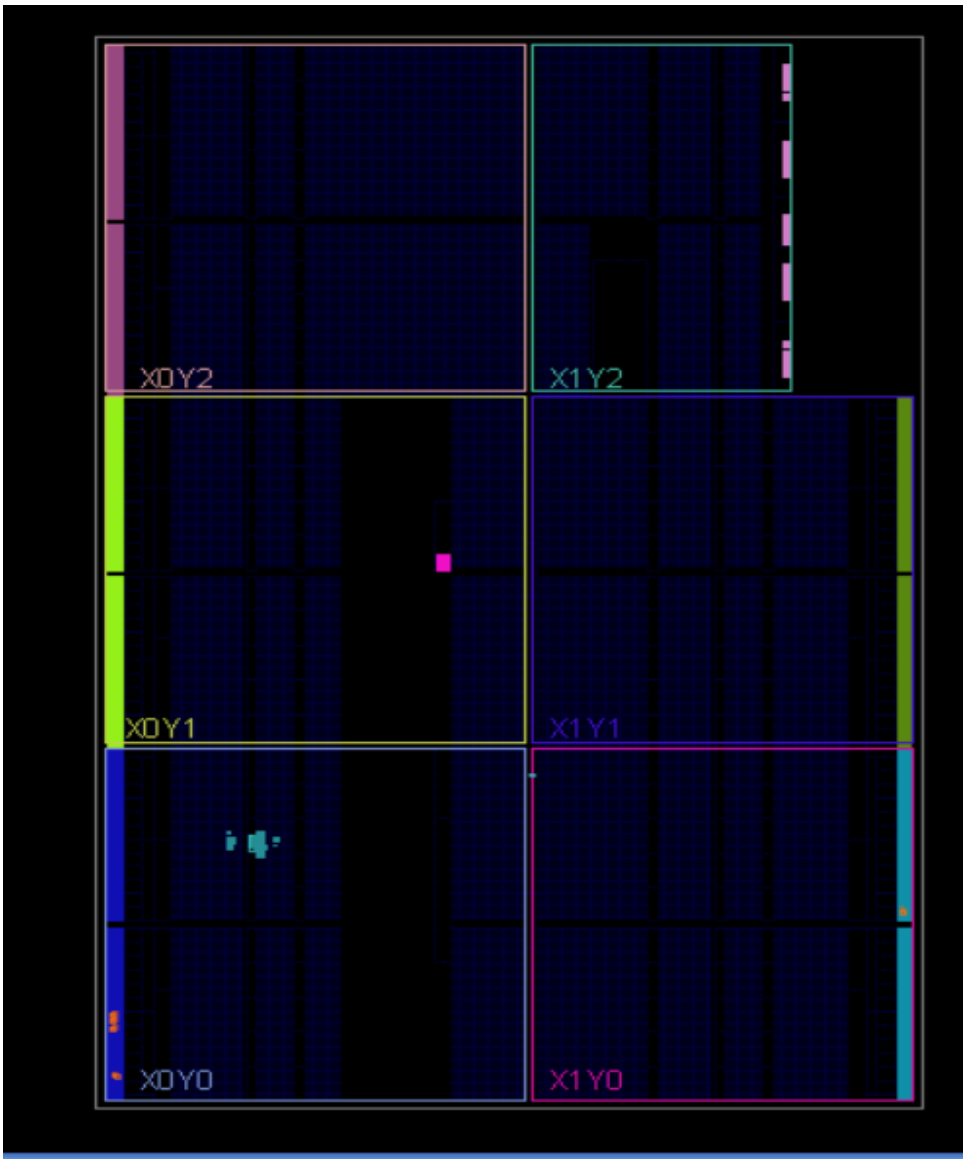


Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
spi_wrapper	34	40	1	15	34	16	0.5	5	1
m1 (spi_slave)	29	23	0	13	29	13	0	0	0
m2 (RAM)	5	9	1	5	5	0	0.5	0	0

TIMING



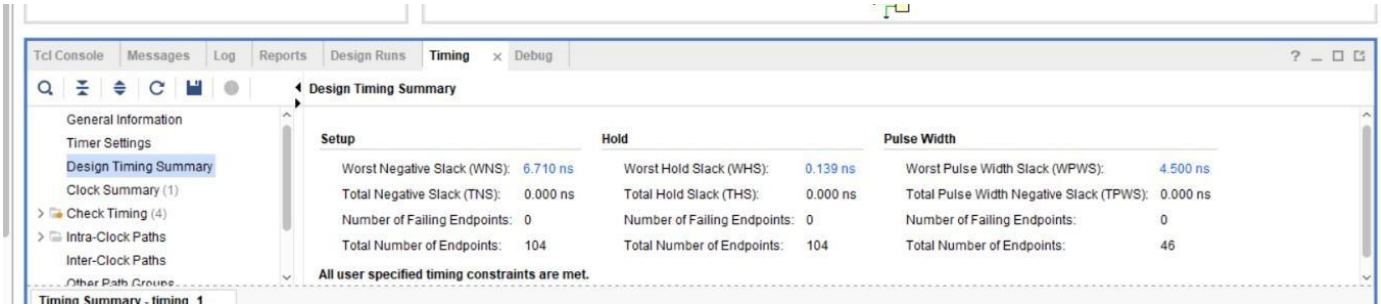
DEVICE

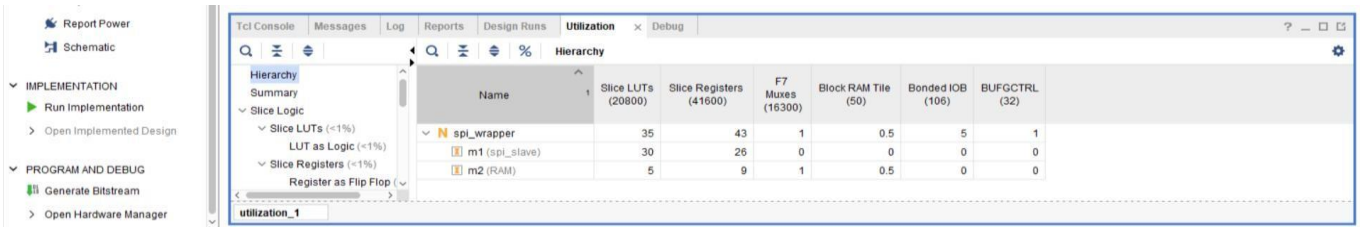


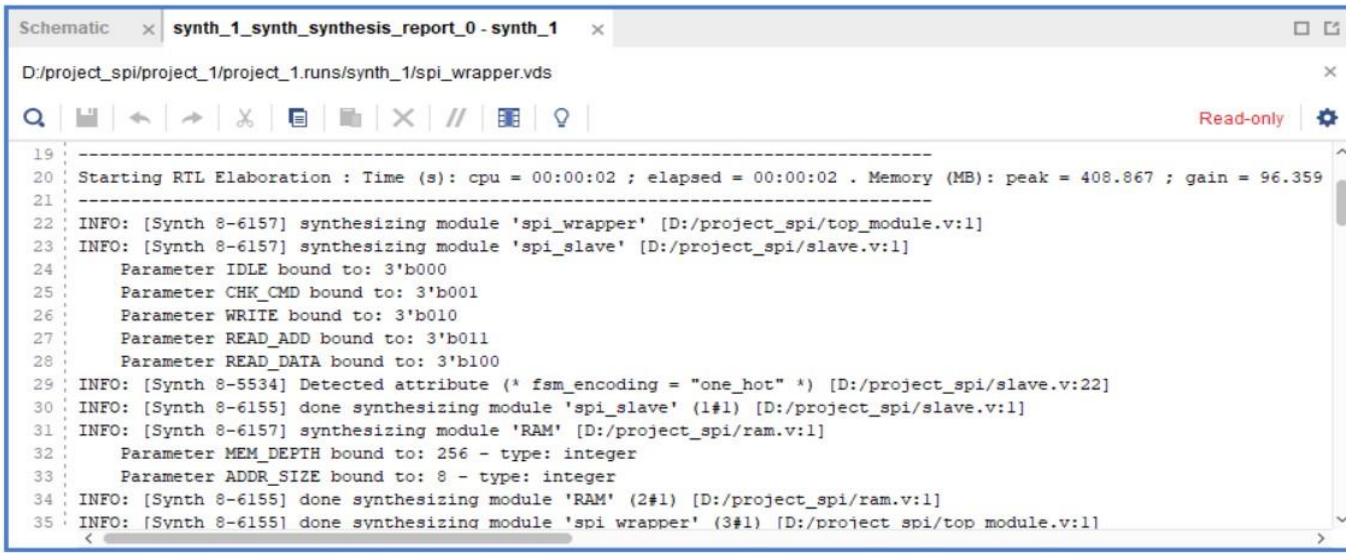
Zero errors and zero critical warnings

SECOND : ONE_HOT ENCODING

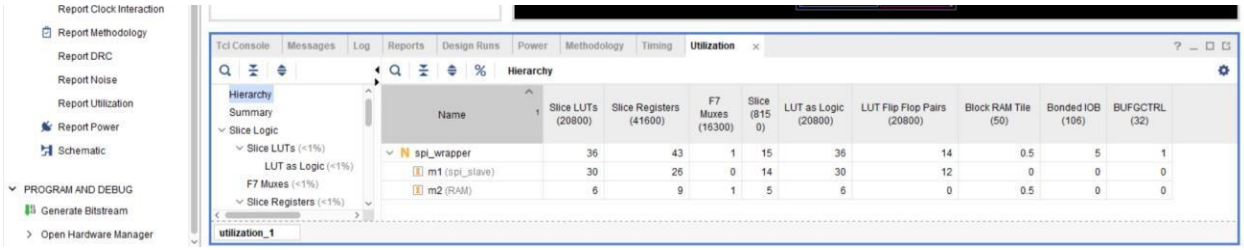
AFTER SYNTHESIS:



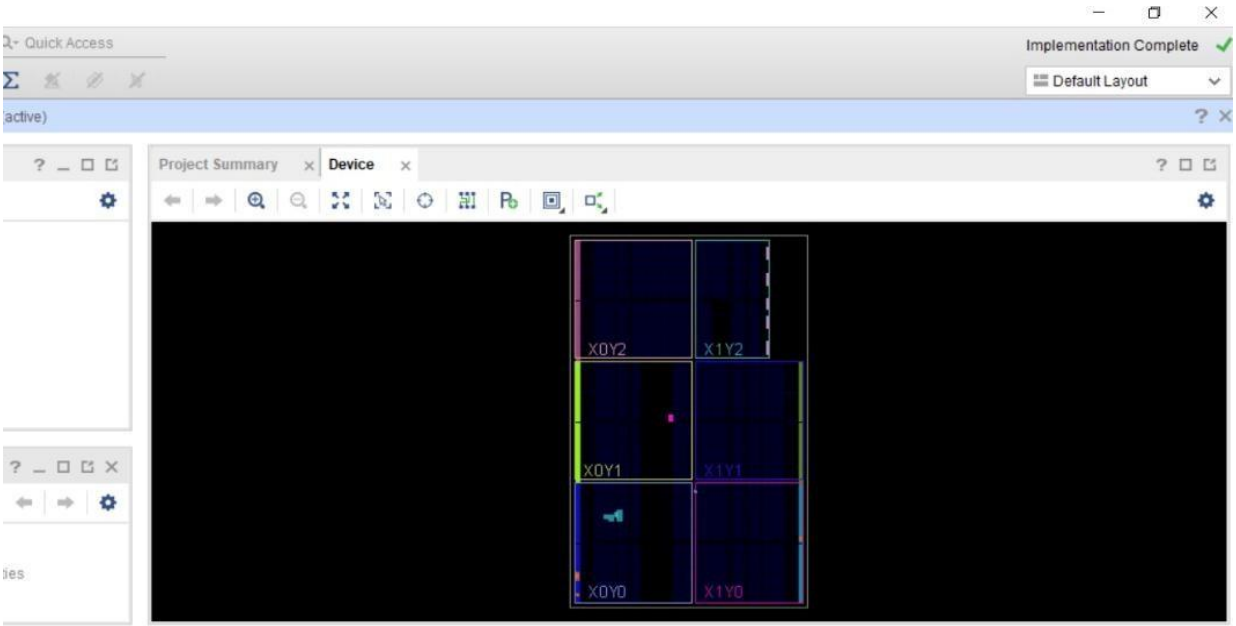




After implementation:



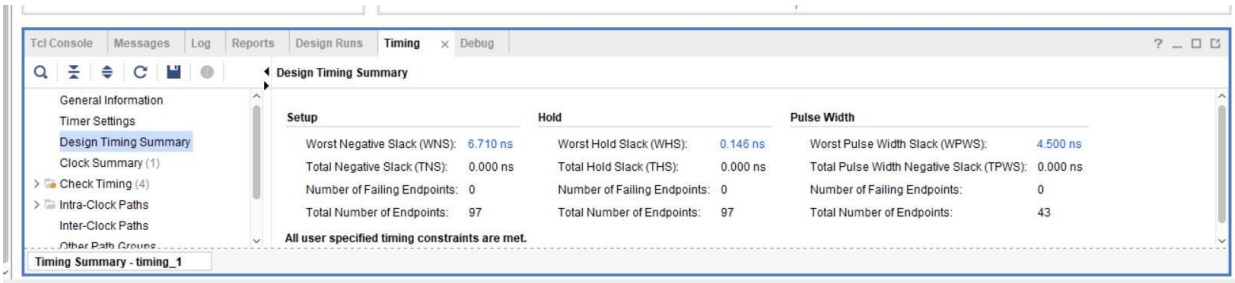
Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
spi_wrapper	36	43	1	15	36	14	0.5	5	1
m1 (spi_slave)	30	26	0	14	30	12	0	0	0
m2 (RAM)	6	9	1	5	6	0	0.5	0	0

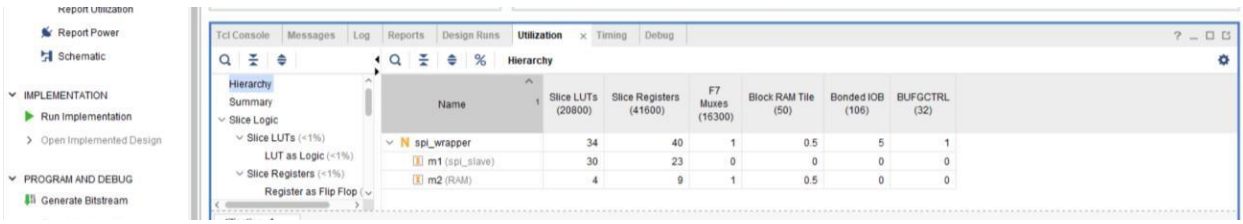


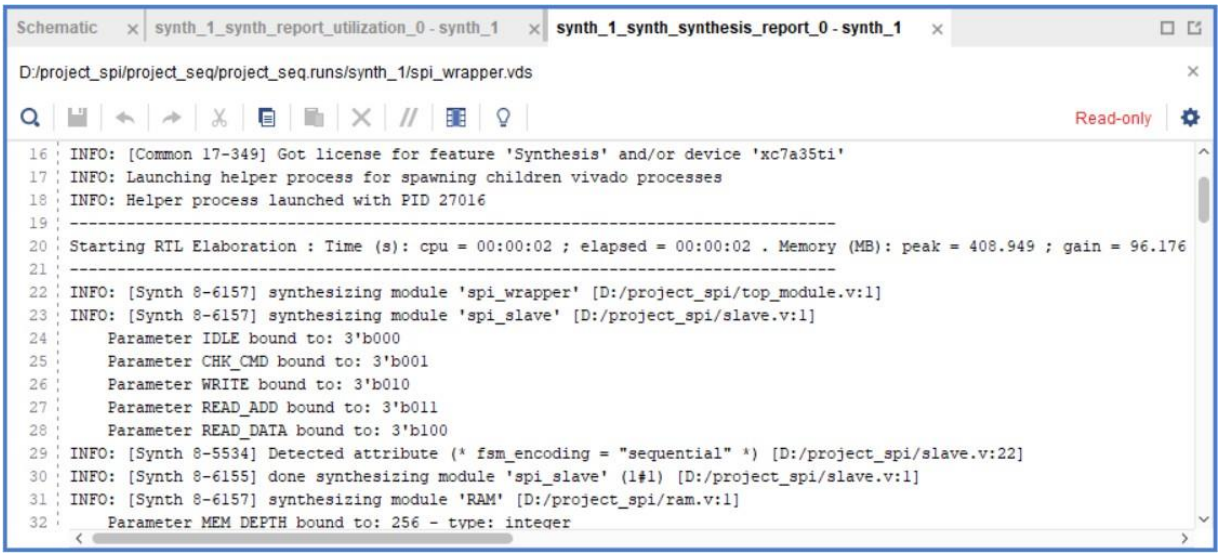
THIRD : SEQUENTIAL

AFTER SYNTHESIZING

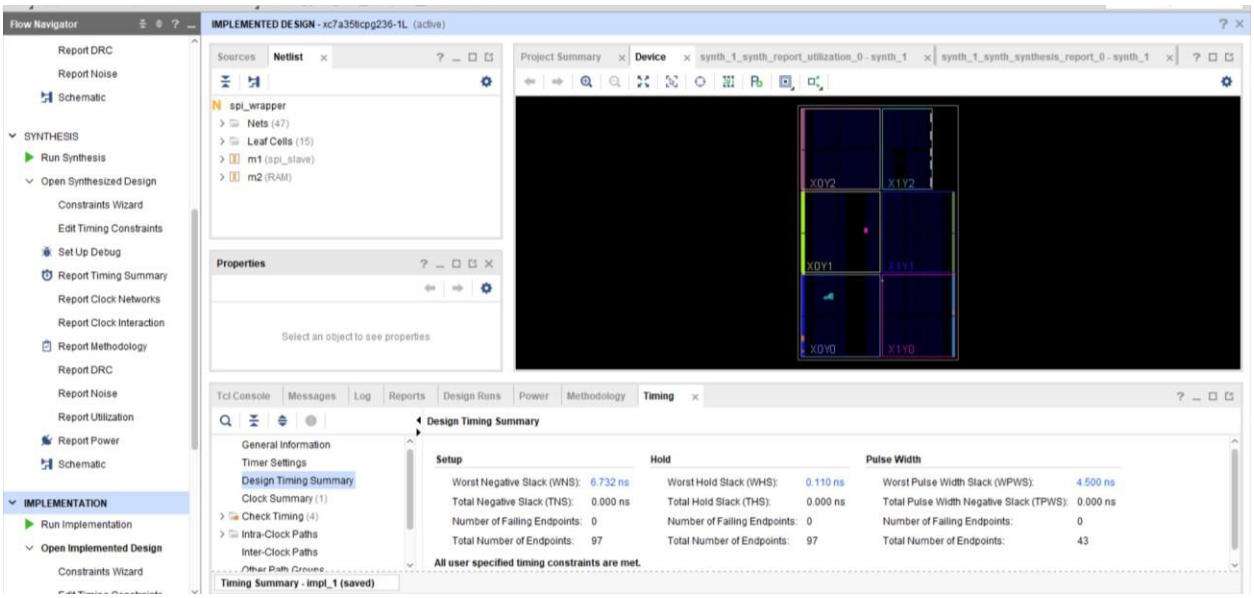
timing and uttilization

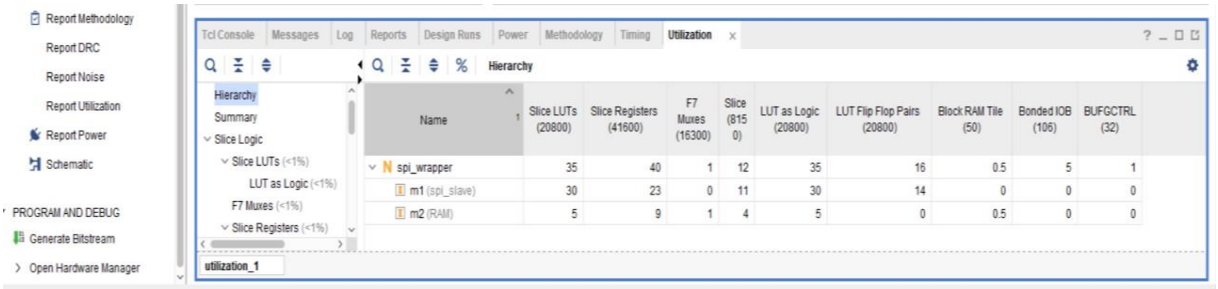






After implementation:

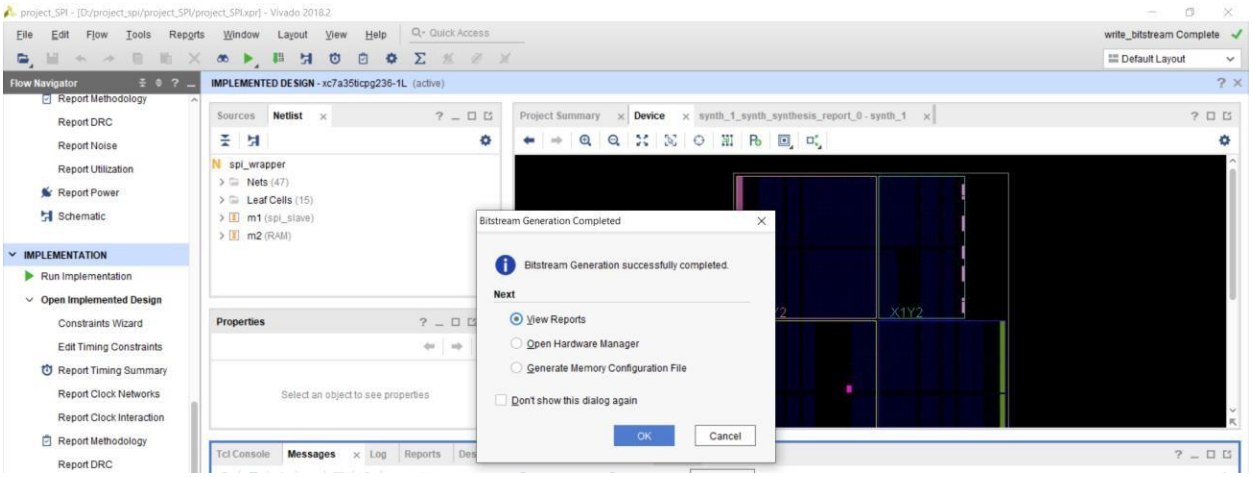




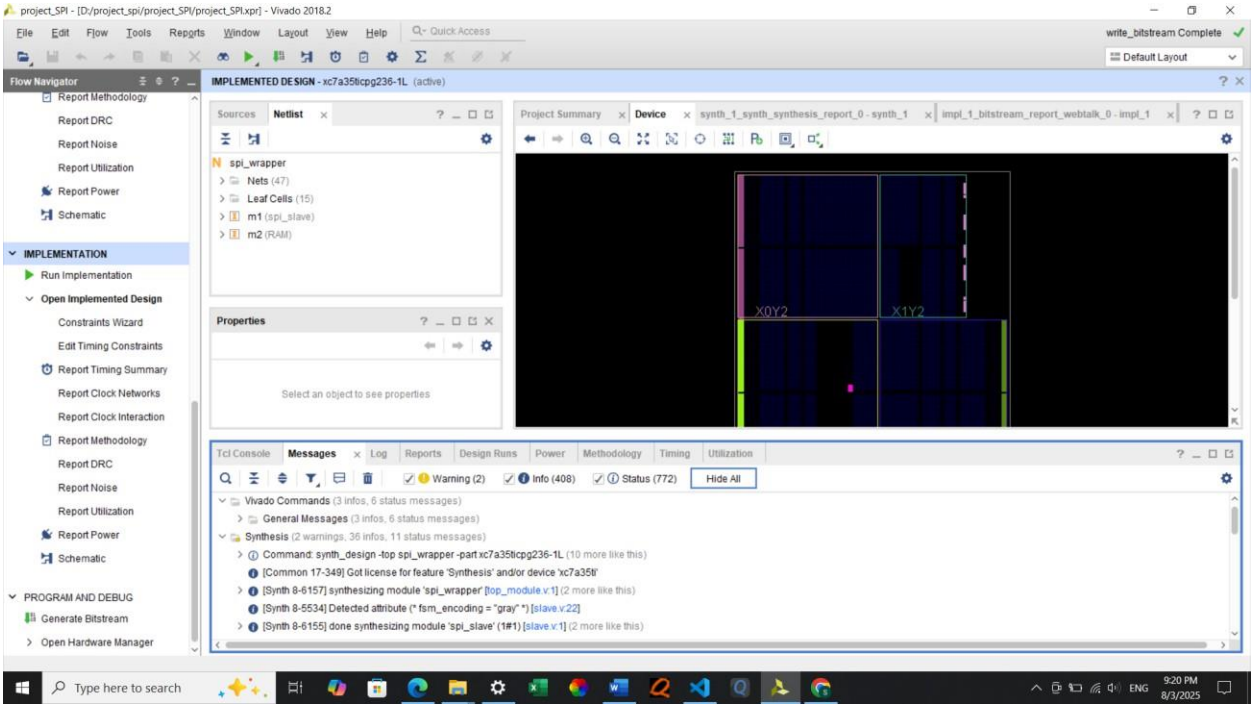
Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
spi_wrapper	35	40	1	12	35	16	0.5	5	1
m1 (spi_slave)	30	23	0	11	30	14	0	0	0
m2 (RAM)	5	9	1	4	5	0	0.5	0	0

BASED ON TIME WE FOUND THAT GRAY_ENCODING IS BETTER

NOW LETS GENERATE BITSTREAM FOR GRAY_ENCODING:

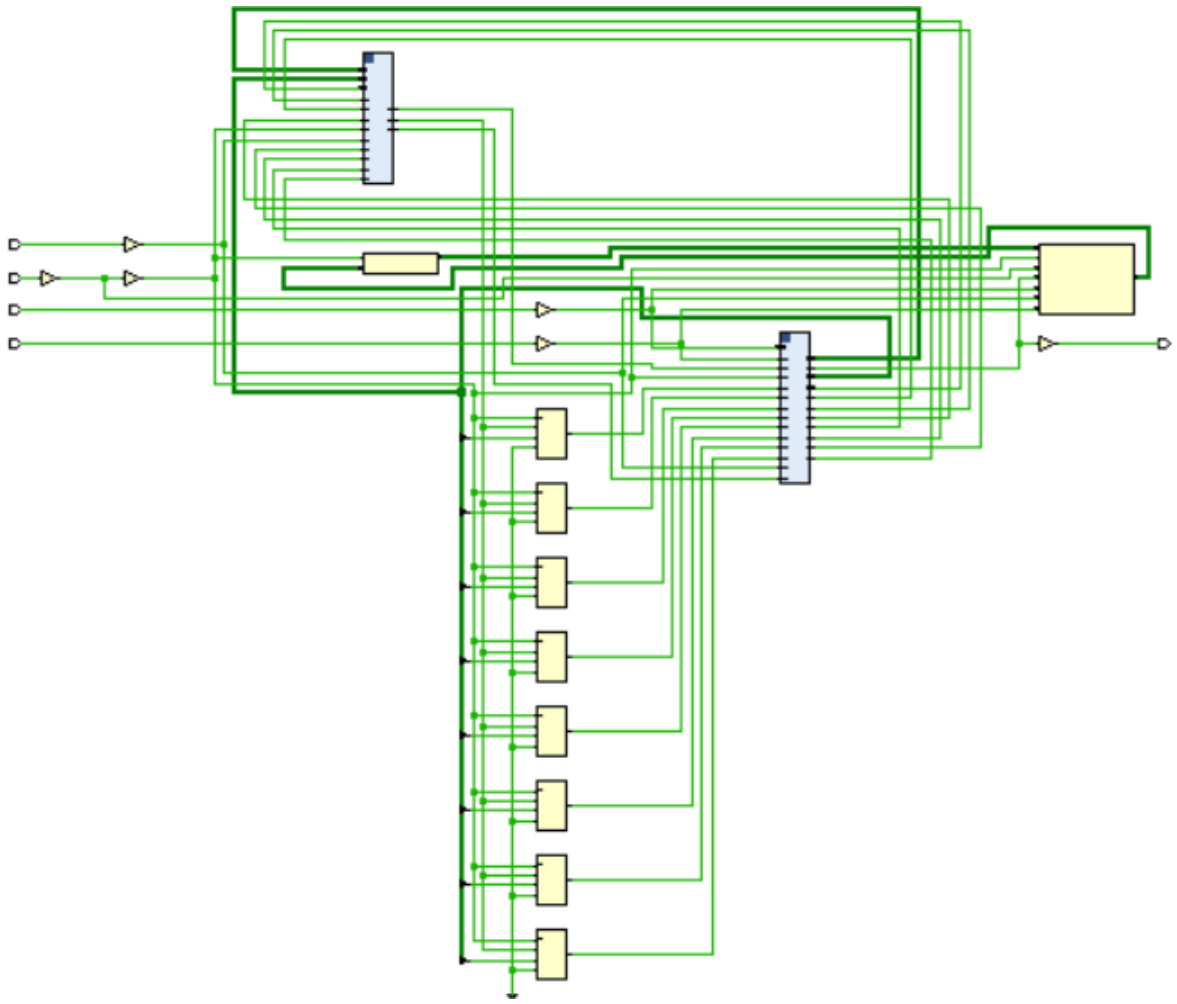


ZERO ERRORS AND ZERO CRITICAL WARNINGS

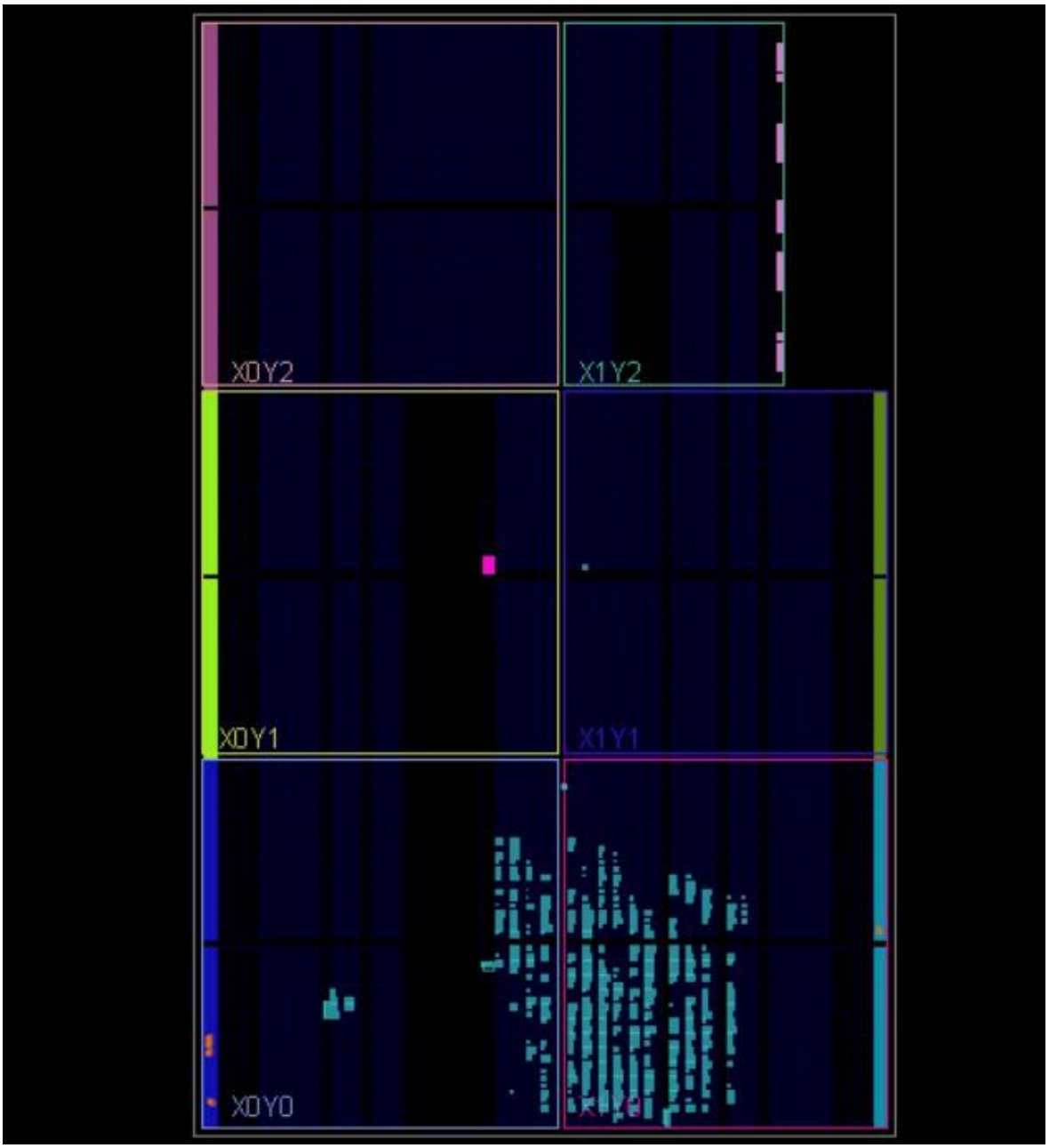


DEBUG

SCHEMATIC AFTER DEBUG



DEVICE



CONSTRAIN FILE AFTER DEBUG

```
## This file is a general .xdc for the Basys3 rev B board
## To use it in a project:
## - uncomment the lines corresponding to used pins
## - rename the used ports (in each line, after get_ports) according to the top
level signal names in the project

## 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]
#set_property -dict { PACKAGE_PIN W17   IOSTANDARD LVCMOS33 } [get_ports {sw[3]}]
#set_property -dict { PACKAGE_PIN W15   IOSTANDARD LVCMOS33 } [get_ports {sw[4]}]
#set_property -dict { PACKAGE_PIN V15   IOSTANDARD LVCMOS33 } [get_ports {sw[5]}]
#set_property -dict { PACKAGE_PIN W14   IOSTANDARD LVCMOS33 } [get_ports {sw[6]}]
#set_property -dict { PACKAGE_PIN W13   IOSTANDARD LVCMOS33 } [get_ports {sw[7]}]
#set_property -dict { PACKAGE_PIN V2    IOSTANDARD LVCMOS33 } [get_ports {sw[8]}]
#set_property -dict { PACKAGE_PIN T3    IOSTANDARD LVCMOS33 } [get_ports {sw[9]}]
#set_property -dict { PACKAGE_PIN T2    IOSTANDARD LVCMOS33 } [get_ports
{sw[10]}]
#set_property -dict { PACKAGE_PIN R3    IOSTANDARD LVCMOS33 } [get_ports
{sw[11]}]
#set_property -dict { PACKAGE_PIN W2    IOSTANDARD LVCMOS33 } [get_ports
{sw[12]}]
#set_property -dict { PACKAGE_PIN U1    IOSTANDARD LVCMOS33 } [get_ports
{sw[13]}]
#set_property -dict { PACKAGE_PIN T1    IOSTANDARD LVCMOS33 } [get_ports
{sw[14]}]
#set_property -dict { PACKAGE_PIN R2    IOSTANDARD LVCMOS33 } [get_ports
{sw[15]}]


## LEDs
set_property -dict {PACKAGE_PIN U16 IOSTANDARD LVCMOS33} [get_ports MISO]
#set_property -dict { PACKAGE_PIN E19   IOSTANDARD LVCMOS33 } [get_ports
{led[1]}]
#set_property -dict { PACKAGE_PIN U19   IOSTANDARD LVCMOS33 } [get_ports
{led[2]}]
```

```
#set_property -dict { PACKAGE_PIN V19   IOSTANDARD LVCMOS33 } [get_ports
{led[3]}]
#set_property -dict { PACKAGE_PIN W18   IOSTANDARD LVCMOS33 } [get_ports
{led[4]}]
#set_property -dict { PACKAGE_PIN U15   IOSTANDARD LVCMOS33 } [get_ports
{led[5]}]
#set_property -dict { PACKAGE_PIN U14   IOSTANDARD LVCMOS33 } [get_ports
{led[6]}]
#set_property -dict { PACKAGE_PIN V14   IOSTANDARD LVCMOS33 } [get_ports
{led[7]}]
#set_property -dict { PACKAGE_PIN V13   IOSTANDARD LVCMOS33 } [get_ports
{led[8]}]
#set_property -dict { PACKAGE_PIN V3    IOSTANDARD LVCMOS33 } [get_ports
{led[9]}]
#set_property -dict { PACKAGE_PIN W3    IOSTANDARD LVCMOS33 } [get_ports
{led[10]}]
#set_property -dict { PACKAGE_PIN U3    IOSTANDARD LVCMOS33 } [get_ports
{led[11]}]
#set_property -dict { PACKAGE_PIN P3    IOSTANDARD LVCMOS33 } [get_ports
{led[12]}]
#set_property -dict { PACKAGE_PIN N3    IOSTANDARD LVCMOS33 } [get_ports
{led[13]}]
#set_property -dict { PACKAGE_PIN P1    IOSTANDARD LVCMOS33 } [get_ports
{led[14]}]
#set_property -dict { PACKAGE_PIN L1    IOSTANDARD LVCMOS33 } [get_ports
{led[15]}]


##7 Segment Display
#set_property -dict { PACKAGE_PIN W7   IOSTANDARD LVCMOS33 } [get_ports {seg[0]}]
#set_property -dict { PACKAGE_PIN W6   IOSTANDARD LVCMOS33 } [get_ports {seg[1]}]
#set_property -dict { PACKAGE_PIN U8   IOSTANDARD LVCMOS33 } [get_ports {seg[2]}]
#set_property -dict { PACKAGE_PIN V8   IOSTANDARD LVCMOS33 } [get_ports {seg[3]}]
#set_property -dict { PACKAGE_PIN U5   IOSTANDARD LVCMOS33 } [get_ports {seg[4]}]
#set_property -dict { PACKAGE_PIN V5   IOSTANDARD LVCMOS33 } [get_ports {seg[5]}]
#set_property -dict { PACKAGE_PIN U7   IOSTANDARD LVCMOS33 } [get_ports {seg[6]}]

#set_property -dict { PACKAGE_PIN V7   IOSTANDARD LVCMOS33 } [get_ports dp]

#set_property -dict { PACKAGE_PIN U2   IOSTANDARD LVCMOS33 } [get_ports {an[0]}]
#set_property -dict { PACKAGE_PIN U4   IOSTANDARD LVCMOS33 } [get_ports {an[1]}]
#set_property -dict { PACKAGE_PIN V4   IOSTANDARD LVCMOS33 } [get_ports {an[2]}]
#set_property -dict { PACKAGE_PIN W4   IOSTANDARD LVCMOS33 } [get_ports {an[3]}]
```

```
##Buttons
#set_property -dict { PACKAGE_PIN U18   IOSTANDARD LVCMOS33 } [get_ports rst]
#set_property -dict { PACKAGE_PIN T18   IOSTANDARD LVCMOS33 } [get_ports btnU]
#set_property -dict { PACKAGE_PIN W19   IOSTANDARD LVCMOS33 } [get_ports btnL]
#set_property -dict { PACKAGE_PIN T17   IOSTANDARD LVCMOS33 } [get_ports btnR]
#set_property -dict { PACKAGE_PIN U17   IOSTANDARD LVCMOS33 } [get_ports btnD]


##Pmod Header JA
#set_property -dict { PACKAGE_PIN J1   IOSTANDARD LVCMOS33 } [get_ports
{JA[0]}];#Sch name = JA1
#set_property -dict { PACKAGE_PIN L2   IOSTANDARD LVCMOS33 } [get_ports
{JA[1]}];#Sch name = JA2
#set_property -dict { PACKAGE_PIN J2   IOSTANDARD LVCMOS33 } [get_ports
{JA[2]}];#Sch name = JA3
#set_property -dict { PACKAGE_PIN G2   IOSTANDARD LVCMOS33 } [get_ports
{JA[3]}];#Sch name = JA4
#set_property -dict { PACKAGE_PIN H1   IOSTANDARD LVCMOS33 } [get_ports
{JA[4]}];#Sch name = JA7
#set_property -dict { PACKAGE_PIN K2   IOSTANDARD LVCMOS33 } [get_ports
{JA[5]}];#Sch name = JA8
#set_property -dict { PACKAGE_PIN H2   IOSTANDARD LVCMOS33 } [get_ports
{JA[6]}];#Sch name = JA9
#set_property -dict { PACKAGE_PIN G3   IOSTANDARD LVCMOS33 } [get_ports
{JA[7]}];#Sch name = JA10

##Pmod Header JB
#set_property -dict { PACKAGE_PIN A14   IOSTANDARD LVCMOS33 } [get_ports
{JB[0]}];#Sch name = JB1
#set_property -dict { PACKAGE_PIN A16   IOSTANDARD LVCMOS33 } [get_ports
{JB[1]}];#Sch name = JB2
#set_property -dict { PACKAGE_PIN B15   IOSTANDARD LVCMOS33 } [get_ports
{JB[2]}];#Sch name = JB3
#set_property -dict { PACKAGE_PIN B16   IOSTANDARD LVCMOS33 } [get_ports
{JB[3]}];#Sch name = JB4
#set_property -dict { PACKAGE_PIN A15   IOSTANDARD LVCMOS33 } [get_ports
{JB[4]}];#Sch name = JB7
#set_property -dict { PACKAGE_PIN A17   IOSTANDARD LVCMOS33 } [get_ports
{JB[5]}];#Sch name = JB8
#set_property -dict { PACKAGE_PIN C15   IOSTANDARD LVCMOS33 } [get_ports
{JB[6]}];#Sch name = JB9
#set_property -dict { PACKAGE_PIN C16   IOSTANDARD LVCMOS33 } [get_ports
{JB[7]}];#Sch name = JB10

##Pmod Header JC
```

```
#set_property -dict { PACKAGE_PIN K17   IOSTANDARD LVCMOS33 } [get_ports
{JC[0]}];#Sch name = JC1
#set_property -dict { PACKAGE_PIN M18   IOSTANDARD LVCMOS33 } [get_ports
{JC[1]}];#Sch name = JC2
#set_property -dict { PACKAGE_PIN N17   IOSTANDARD LVCMOS33 } [get_ports
{JC[2]}];#Sch name = JC3
#set_property -dict { PACKAGE_PIN P18   IOSTANDARD LVCMOS33 } [get_ports
{JC[3]}];#Sch name = JC4
#set_property -dict { PACKAGE_PIN L17   IOSTANDARD LVCMOS33 } [get_ports
{JC[4]}];#Sch name = JC7
#set_property -dict { PACKAGE_PIN M19   IOSTANDARD LVCMOS33 } [get_ports
{JC[5]}];#Sch name = JC8
#set_property -dict { PACKAGE_PIN P17   IOSTANDARD LVCMOS33 } [get_ports
{JC[6]}];#Sch name = JC9
#set_property -dict { PACKAGE_PIN R18   IOSTANDARD LVCMOS33 } [get_ports
{JC[7]}];#Sch name = JC10

##Pmod Header JXADC
#set_property -dict { PACKAGE_PIN J3   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[0]}];#Sch name = XA1_P
#set_property -dict { PACKAGE_PIN L3   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[1]}];#Sch name = XA2_P
#set_property -dict { PACKAGE_PIN M2   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[2]}];#Sch name = XA3_P
#set_property -dict { PACKAGE_PIN N2   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[3]}];#Sch name = XA4_P
#set_property -dict { PACKAGE_PIN K3   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[4]}];#Sch name = XA1_N
#set_property -dict { PACKAGE_PIN M3   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[5]}];#Sch name = XA2_N
#set_property -dict { PACKAGE_PIN M1   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[6]}];#Sch name = XA3_N
#set_property -dict { PACKAGE_PIN N1   IOSTANDARD LVCMOS33 } [get_ports
{JXADC[7]}];#Sch name = XA4_N


##VGA Connector
#set_property -dict { PACKAGE_PIN G19   IOSTANDARD LVCMOS33 } [get_ports
{vgaRed[0]}]
#set_property -dict { PACKAGE_PIN H19   IOSTANDARD LVCMOS33 } [get_ports
{vgaRed[1]}]
#set_property -dict { PACKAGE_PIN J19   IOSTANDARD LVCMOS33 } [get_ports
{vgaRed[2]}]
#set_property -dict { PACKAGE_PIN N19   IOSTANDARD LVCMOS33 } [get_ports
{vgaRed[3]}]
```

```
#set_property -dict { PACKAGE_PIN N18   IOSTANDARD LVCMOS33 } [get_ports
{vgaBlue[0]}]
#set_property -dict { PACKAGE_PIN L18   IOSTANDARD LVCMOS33 } [get_ports
{vgaBlue[1]}]
#set_property -dict { PACKAGE_PIN K18   IOSTANDARD LVCMOS33 } [get_ports
{vgaBlue[2]}]
#set_property -dict { PACKAGE_PIN J18   IOSTANDARD LVCMOS33 } [get_ports
{vgaBlue[3]}]
#set_property -dict { PACKAGE_PIN J17   IOSTANDARD LVCMOS33 } [get_ports
{vgaGreen[0]}]
#set_property -dict { PACKAGE_PIN H17   IOSTANDARD LVCMOS33 } [get_ports
{vgaGreen[1]}]
#set_property -dict { PACKAGE_PIN G17   IOSTANDARD LVCMOS33 } [get_ports
{vgaGreen[2]}]
#set_property -dict { PACKAGE_PIN D17   IOSTANDARD LVCMOS33 } [get_ports
{vgaGreen[3]}]
#set_property -dict { PACKAGE_PIN P19   IOSTANDARD LVCMOS33 } [get_ports Hsync]
#set_property -dict { PACKAGE_PIN R19   IOSTANDARD LVCMOS33 } [get_ports Vsync]


##USB-RS232 Interface
#set_property -dict { PACKAGE_PIN B18   IOSTANDARD LVCMOS33 } [get_ports RsRx]
#set_property -dict { PACKAGE_PIN A18   IOSTANDARD LVCMOS33 } [get_ports RsTx]


##USB HID (PS/2)
#set_property -dict { PACKAGE_PIN C17   IOSTANDARD LVCMOS33   PULLUP true }
[get_ports PS2Clk]
#set_property -dict { PACKAGE_PIN B17   IOSTANDARD LVCMOS33   PULLUP true }
[get_ports PS2Data]


##Quad SPI Flash
##Note that CCLK_0 cannot be placed in 7 series devices. You can access it using
the
##STARTUPE2 primitive.
#set_property -dict { PACKAGE_PIN D18   IOSTANDARD LVCMOS33 } [get_ports
{QspiDB[0]}]
#set_property -dict { PACKAGE_PIN D19   IOSTANDARD LVCMOS33 } [get_ports
{QspiDB[1]}]
#set_property -dict { PACKAGE_PIN G18   IOSTANDARD LVCMOS33 } [get_ports
{QspiDB[2]}]
#set_property -dict { PACKAGE_PIN F18   IOSTANDARD LVCMOS33 } [get_ports
{QspiDB[3]}]
#set_property -dict { PACKAGE_PIN K19   IOSTANDARD LVCMOS33 } [get_ports QspiCSn]
```

```
## 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]

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 1024 [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]
```