

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

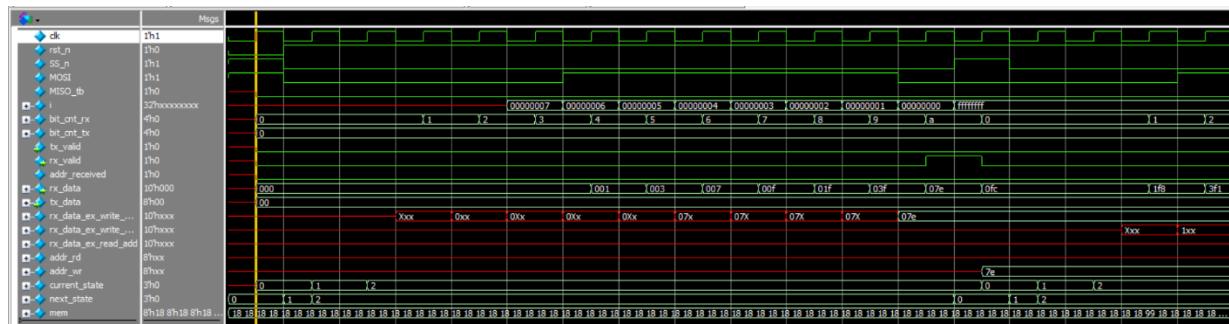
TEAM : SENSITIVITY\_LIST

NAMES:

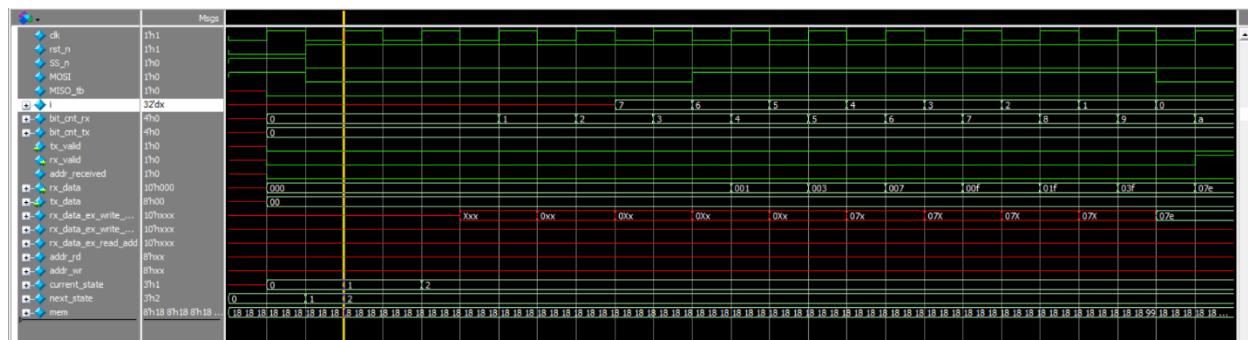
Radwa Yasser Ahmed  
Sherif Ahmed

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

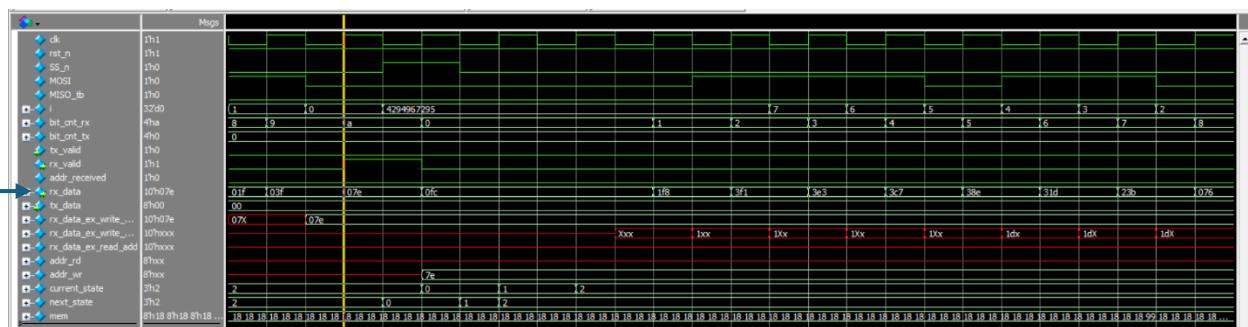
1. THE SYSTEM IS HELD IN RESET ( $RST_N = 0$ ), ENSURING ALL INTERNAL STATES ARE INITIALIZED BEFORE STARTING SPI COMMUNICATION.



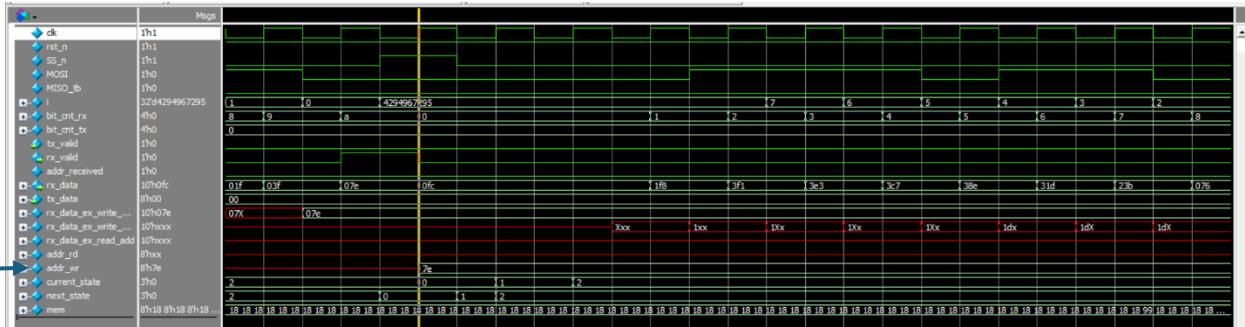
**2. AFTER RESET, THE SPI MODULE TRANSITIONS TO CHK\_CMD, RECEIVING THE FIRST MOSI BIT AS 0, CORRECTLY IDENTIFYING THE COMMAND AS A WRITE OPERATION IN THE NEXT CYCLE.**



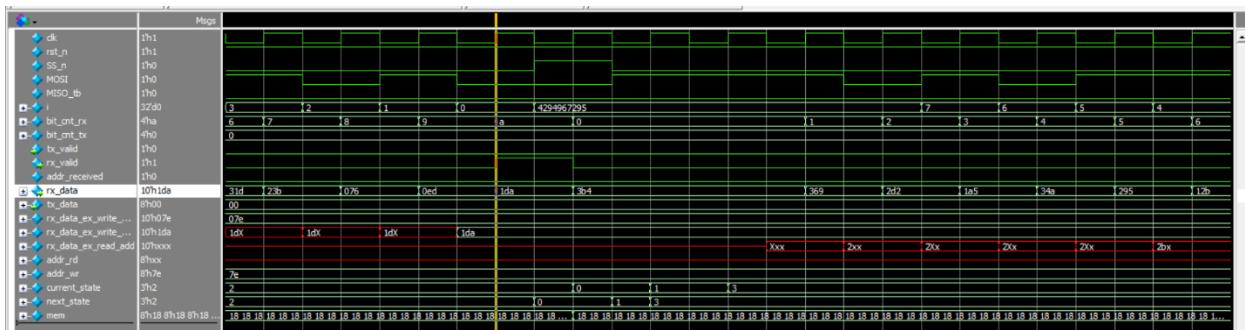
**3. SERIAL INPUT VIA MOSI COMPLETES, FORMING THE FULL 10-BIT RX\_DATA WHICH MATCHES THE EXPECTED WRITE ADDRESS.**



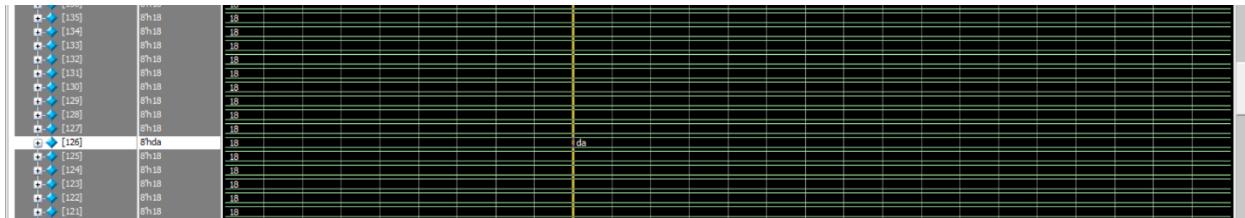
**4. AFTER ONE CLOCK CYCLE, THE WRITE ADDRESS IS UPDATED.**



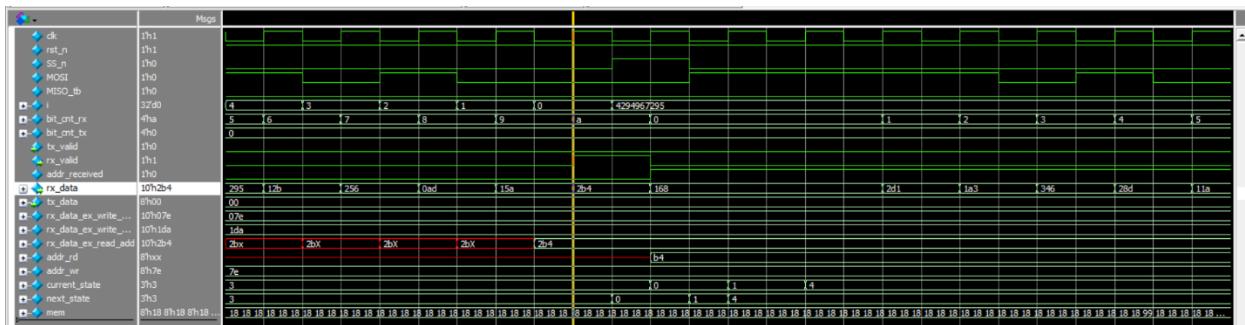
**5. ANOTHER SERIAL MOSI TRANSFER FORMS THE DATA TO BE WRITTEN. ONCE RECEIVED, RX\_DATA MATCHES THE EXPECTED VALUE THIS TIME THE LAST TWO BIT IS ONE SO IT'S WRITE DATA.**



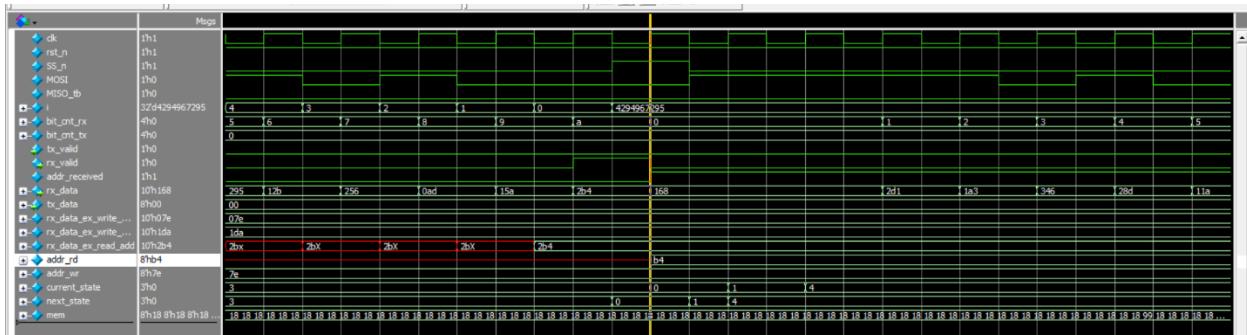
**6. AFTER A CLOCK CYCLE, MEMORY AT THE PREVIOUSLY RECEIVED WRITE ADDRESS (8'H7E=8'D126) IS UPDATED WITH THE NEW DATA .**



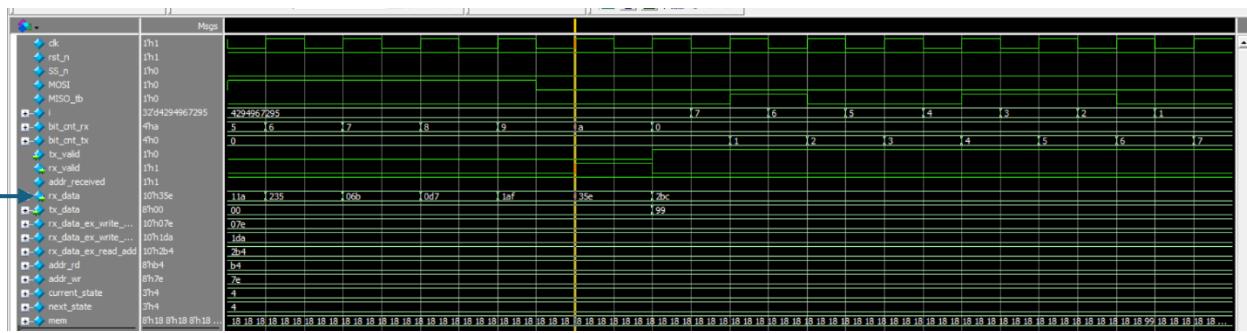
**7. A 10-BIT READ ADDRESS IS RECEIVED SERIALLY THROUGH MOSI AND STORED IN RX\_DATA.**



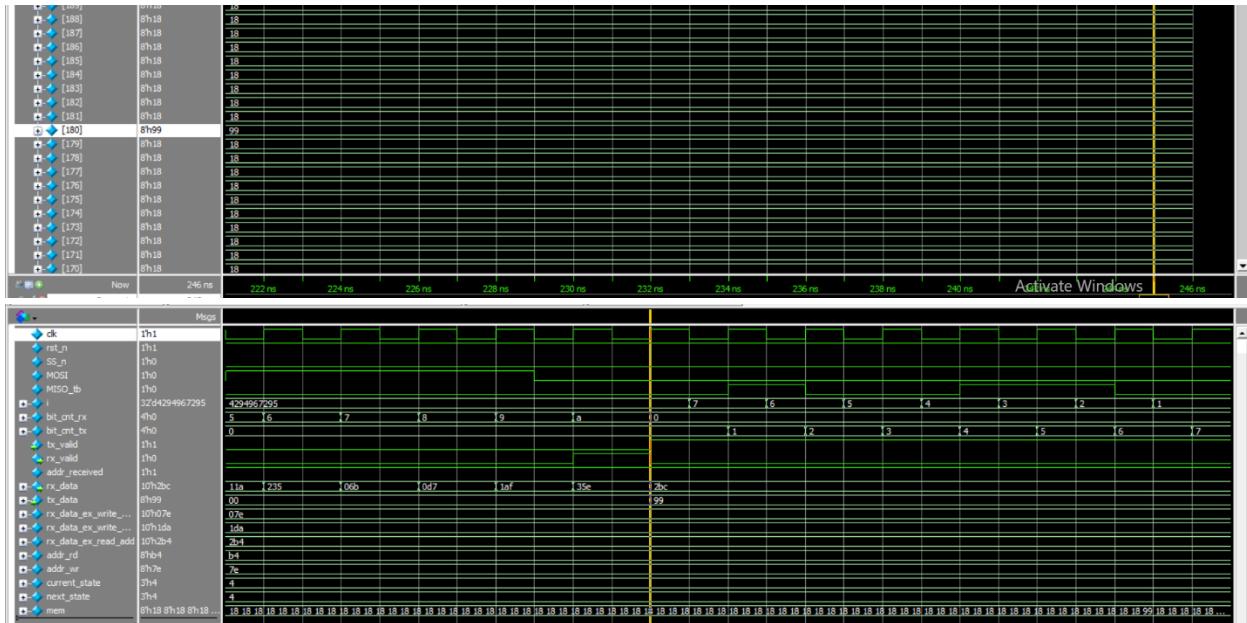
## 8. AFTER A CLOCK CYCLE, THE READ ADDRESS IS UPDATED .



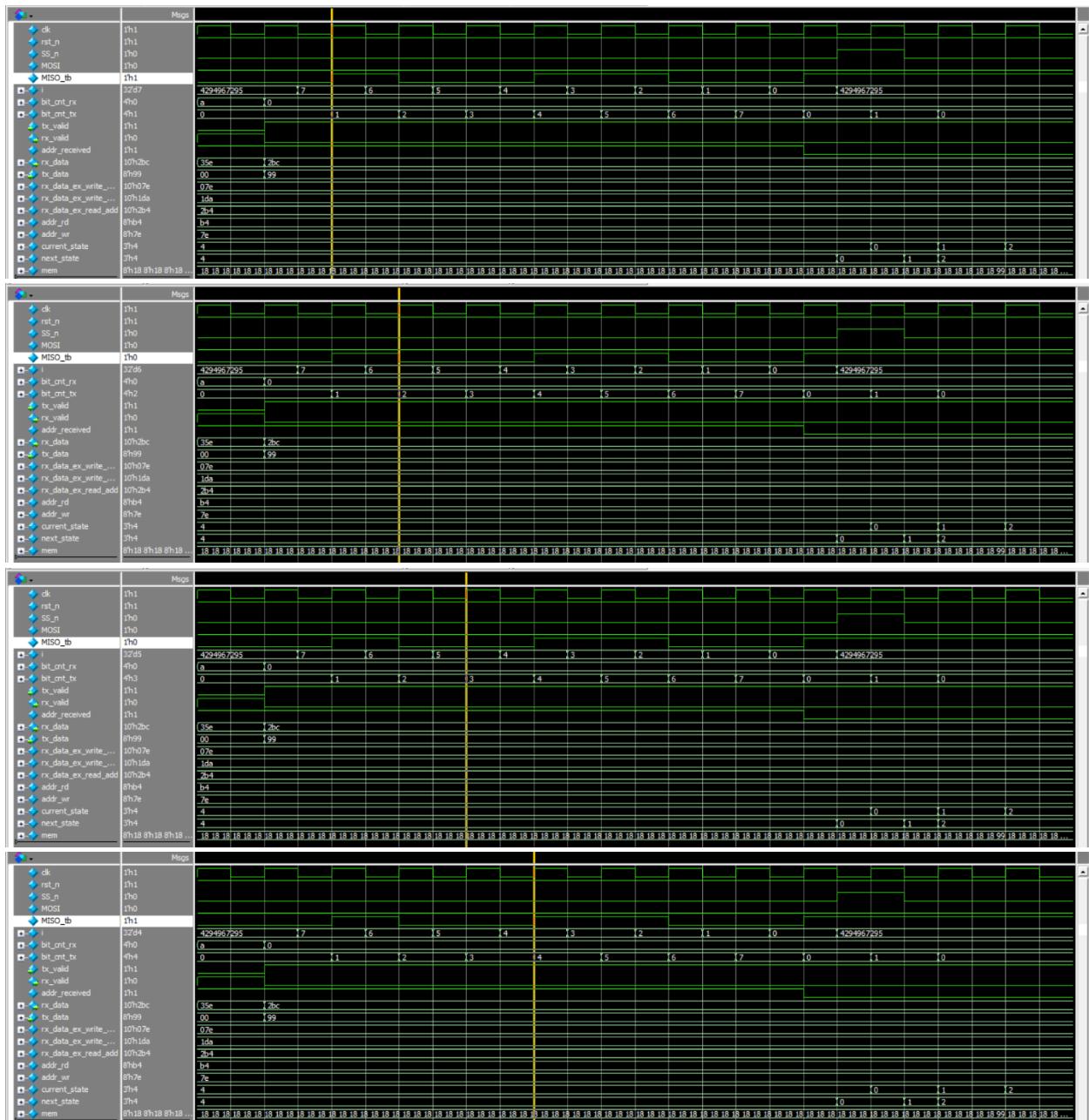
## 9. ANOTHER SERIAL INPUT MOSI IS RECEIVED THE FIRST 2 BITS IS 3 SO THAT MEANS WE WILL READ FROM THE MEM.

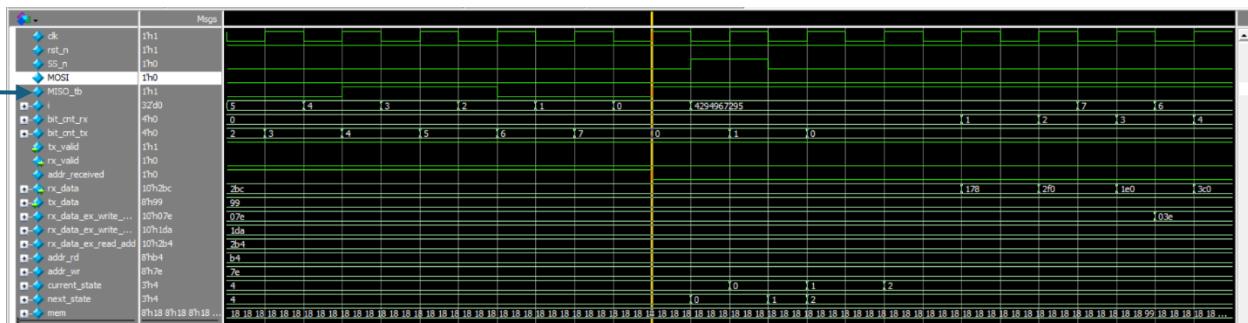
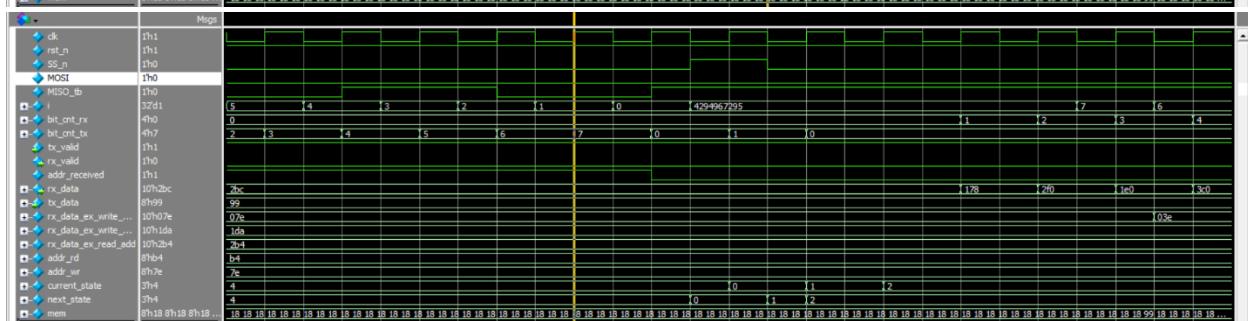
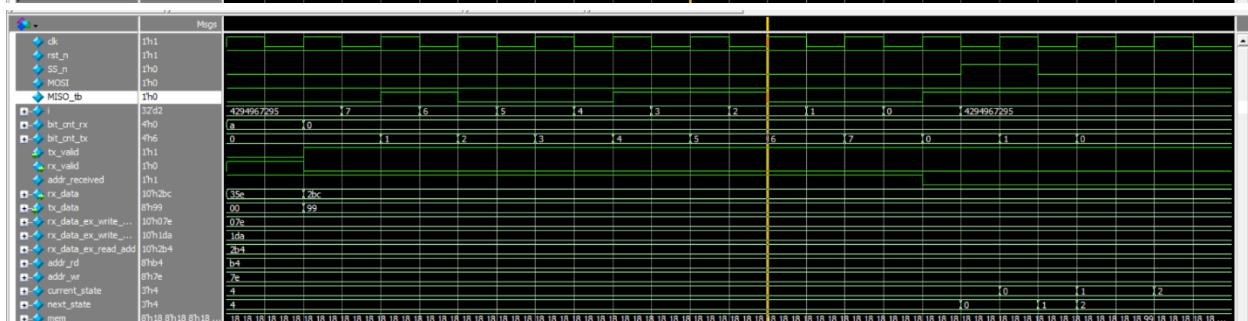
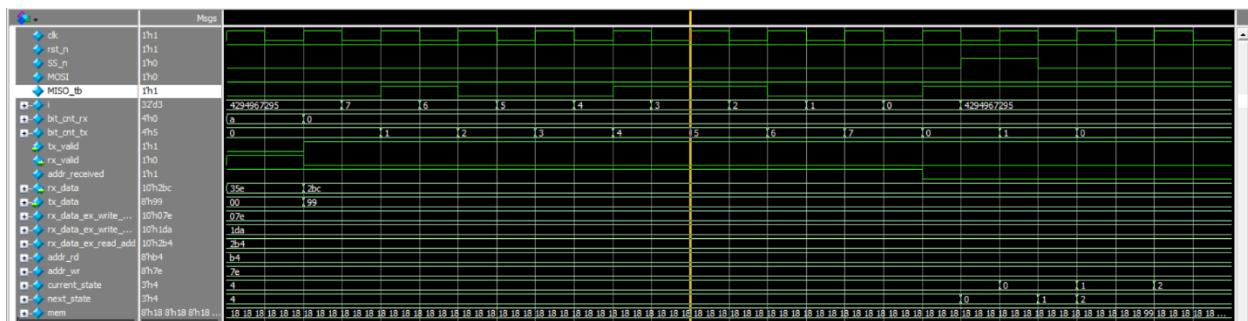


## 10. AFTER A CLOCK CYCLE, THE CONTENT OF MEMORY AT THE PREVIOUSLY RECEIVED READ ADDRESS (8'HB4=8'D180) UPDATES TX\_DATA.

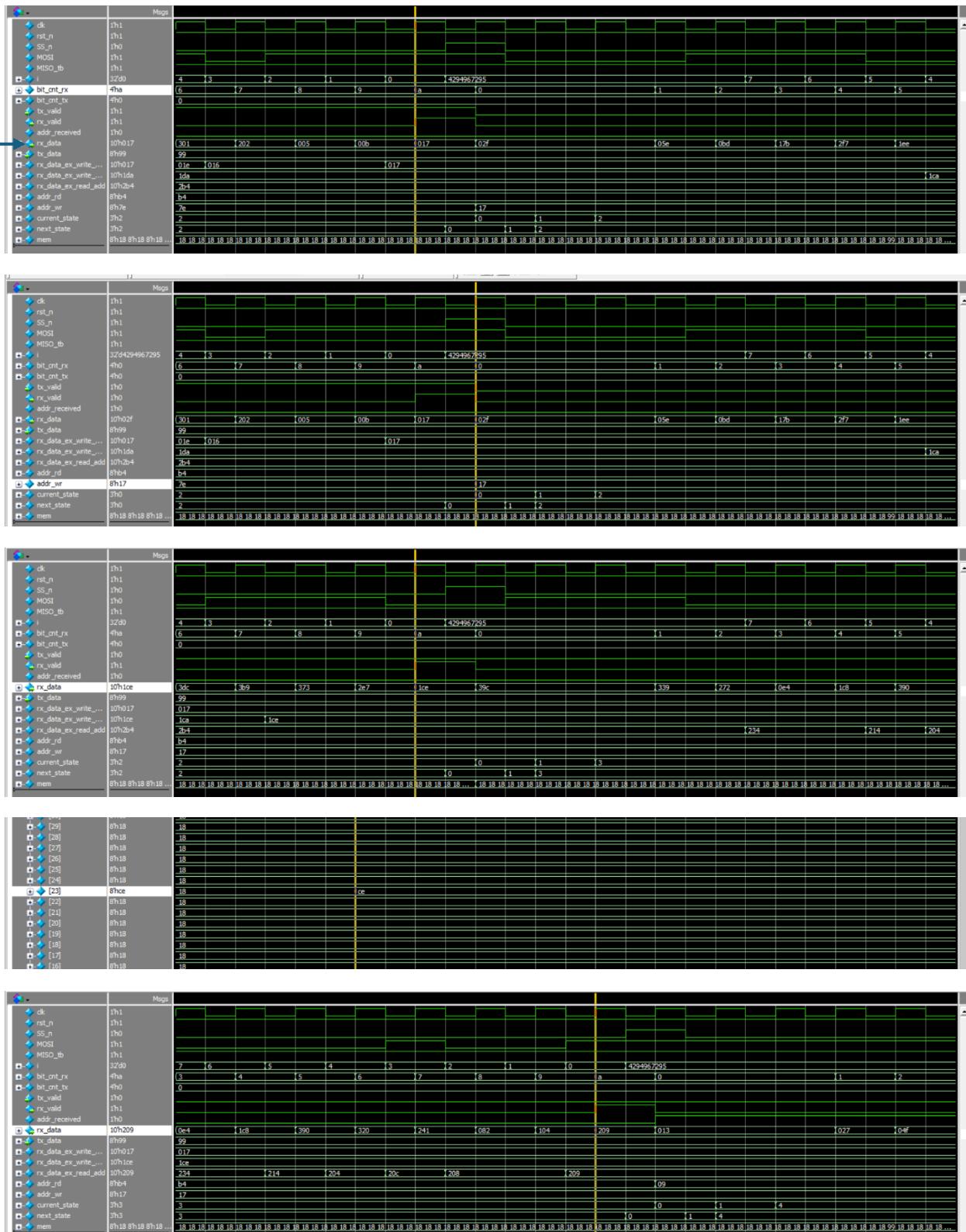


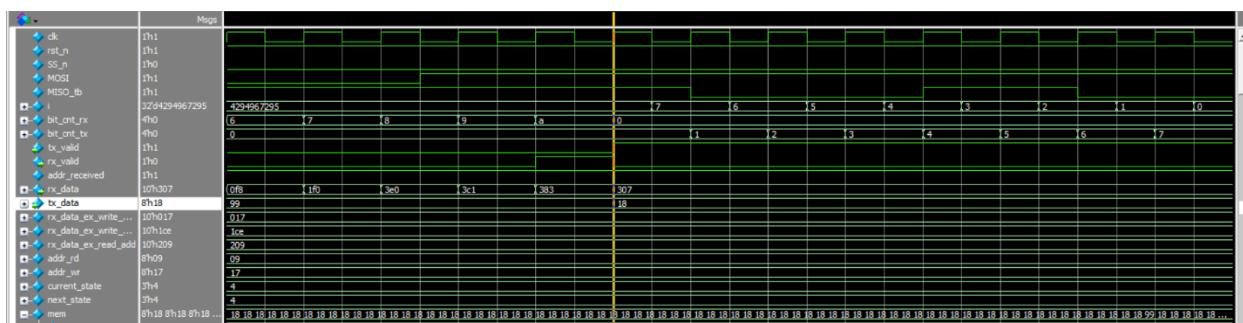
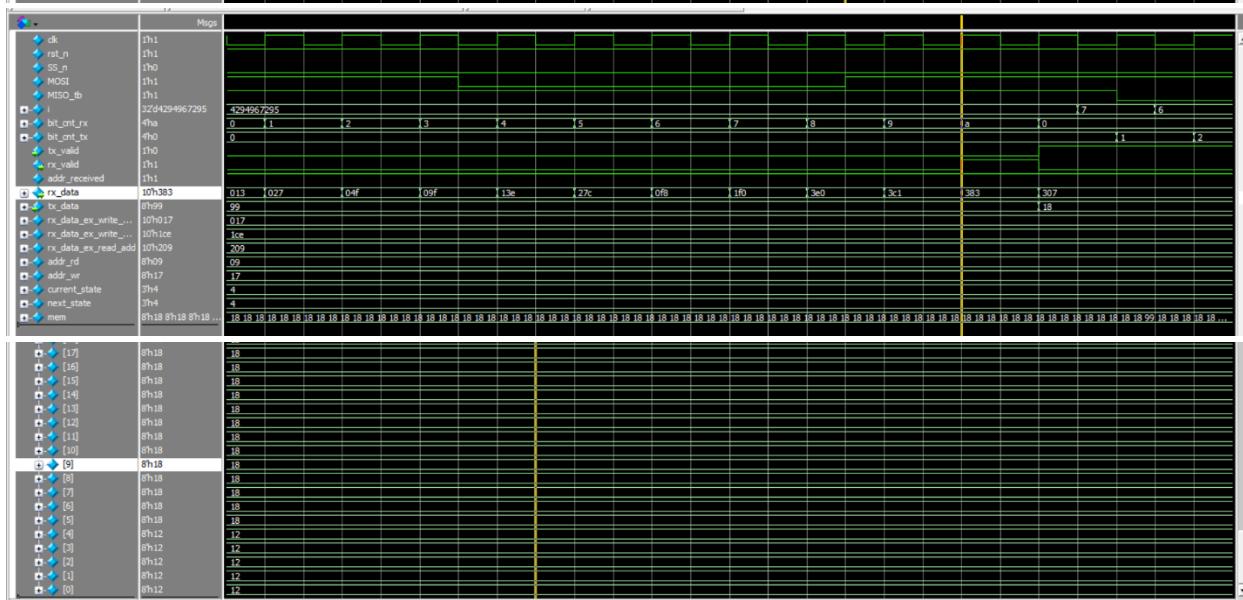
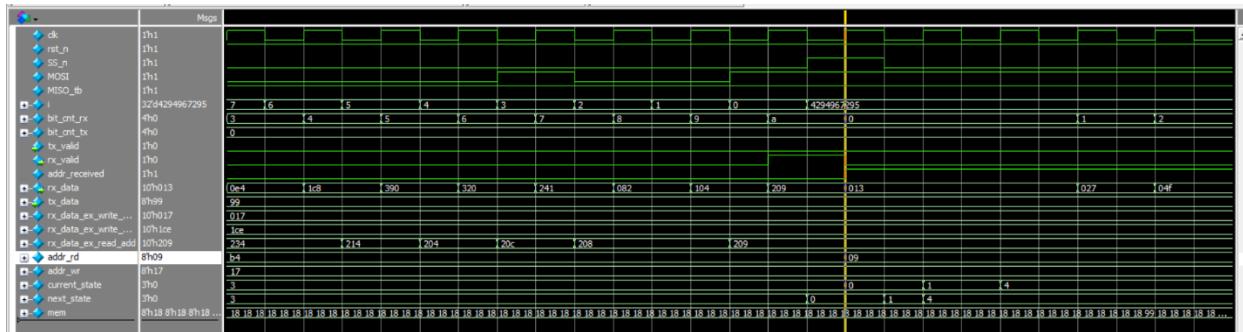
**11. BITS ARE THEN SHIFTED OUT SERIALLY THROUGH MISO, MATCHING THE EXPECTED DATA BIT-BY-BIT.  
(8'H99=8'B10011001).**

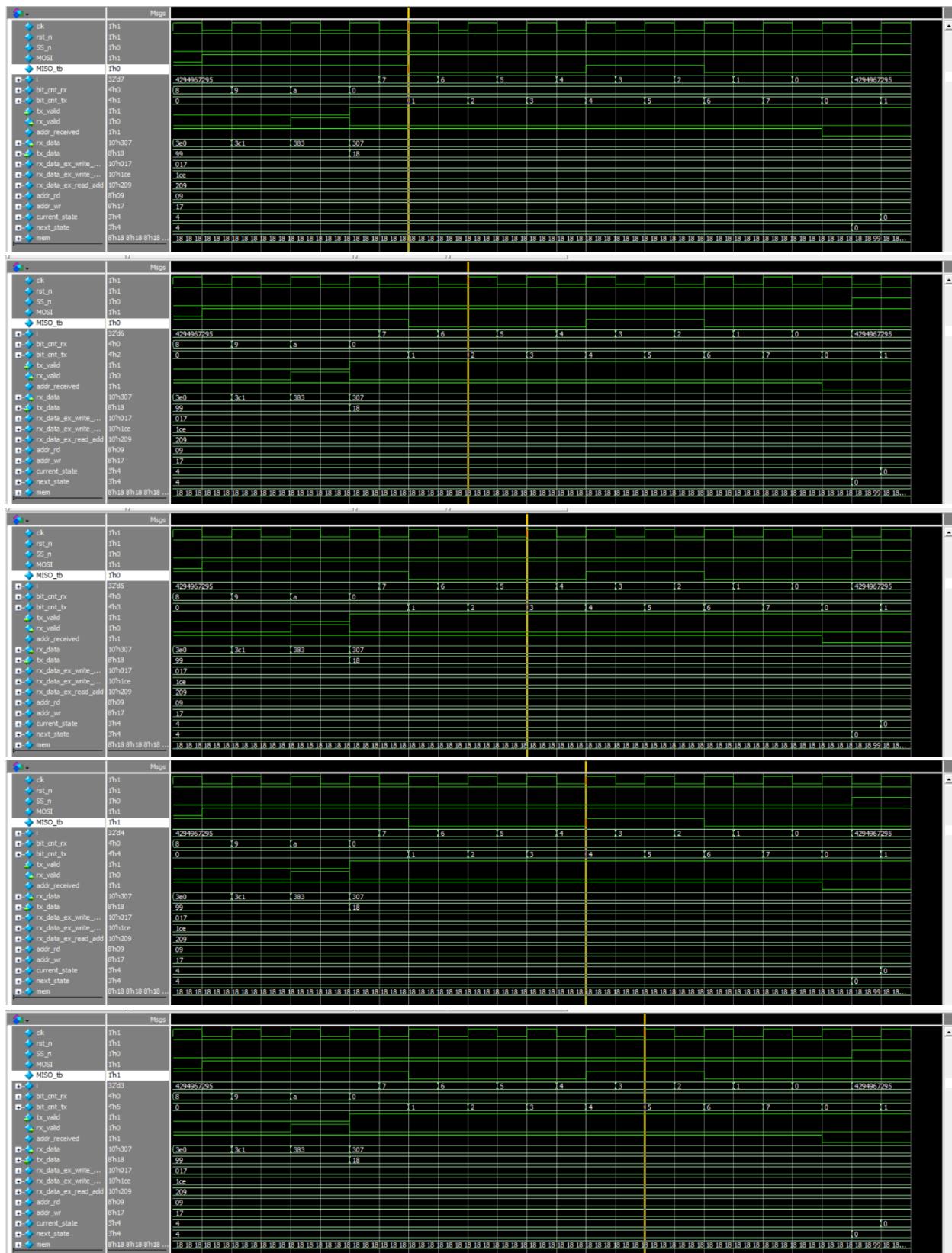


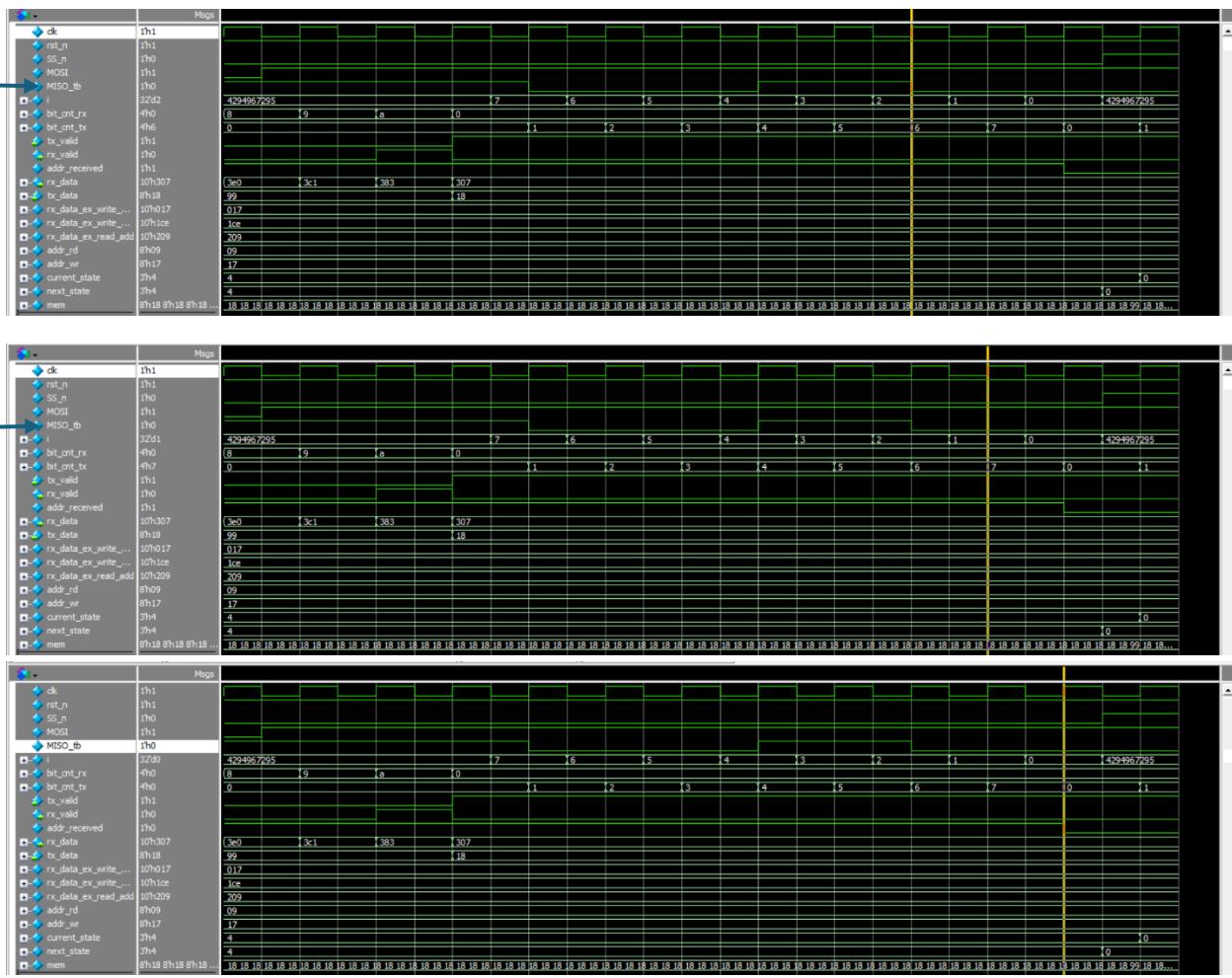


## THAT'S THE SAME FUNCTIONS BUT WITH DIFFERENT INPUTS .

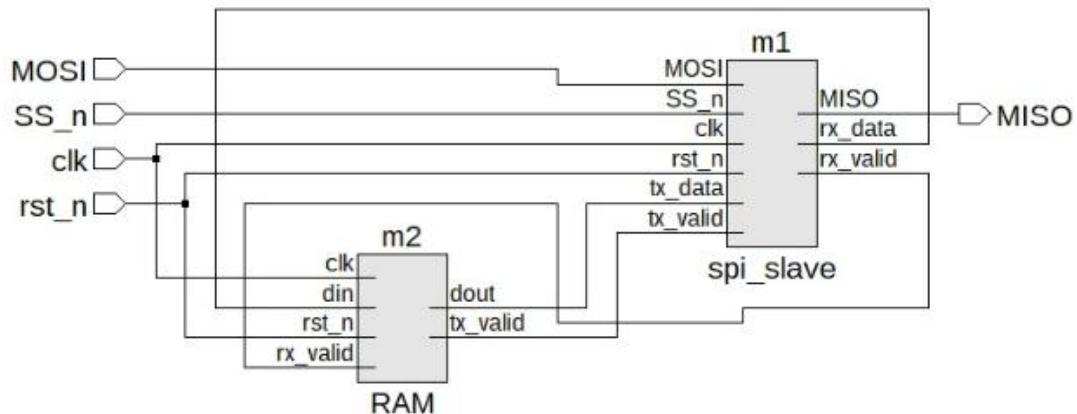
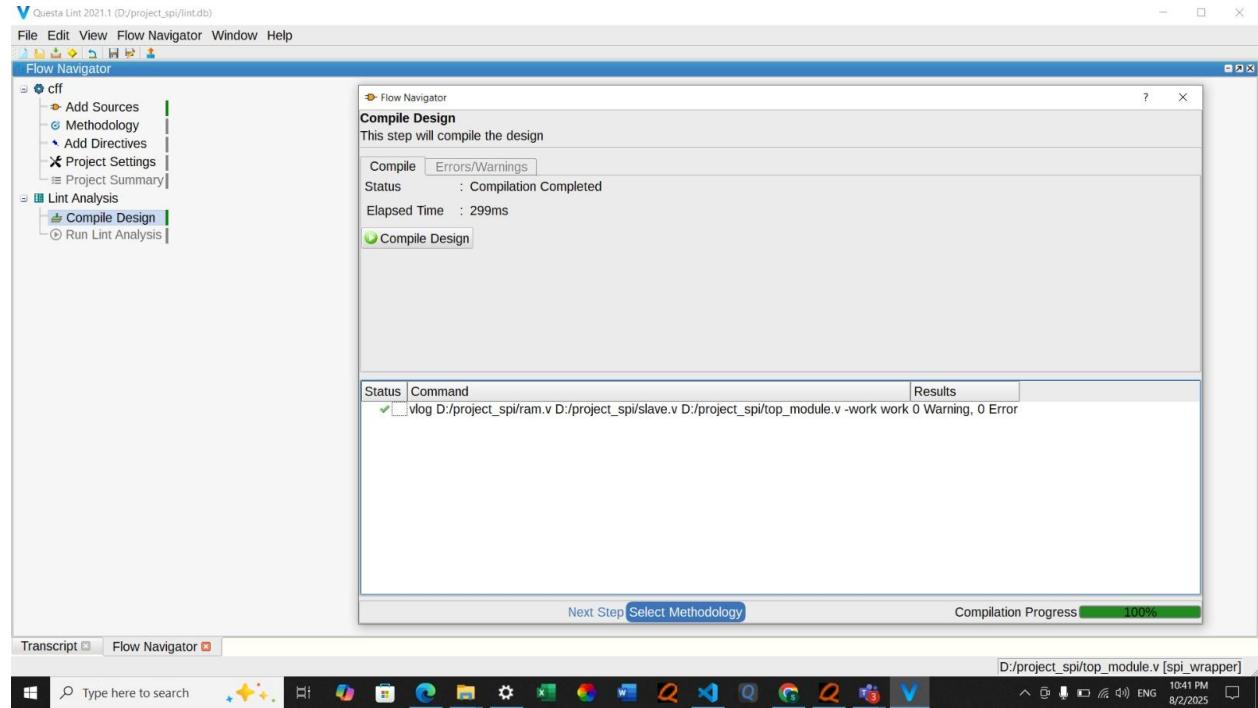








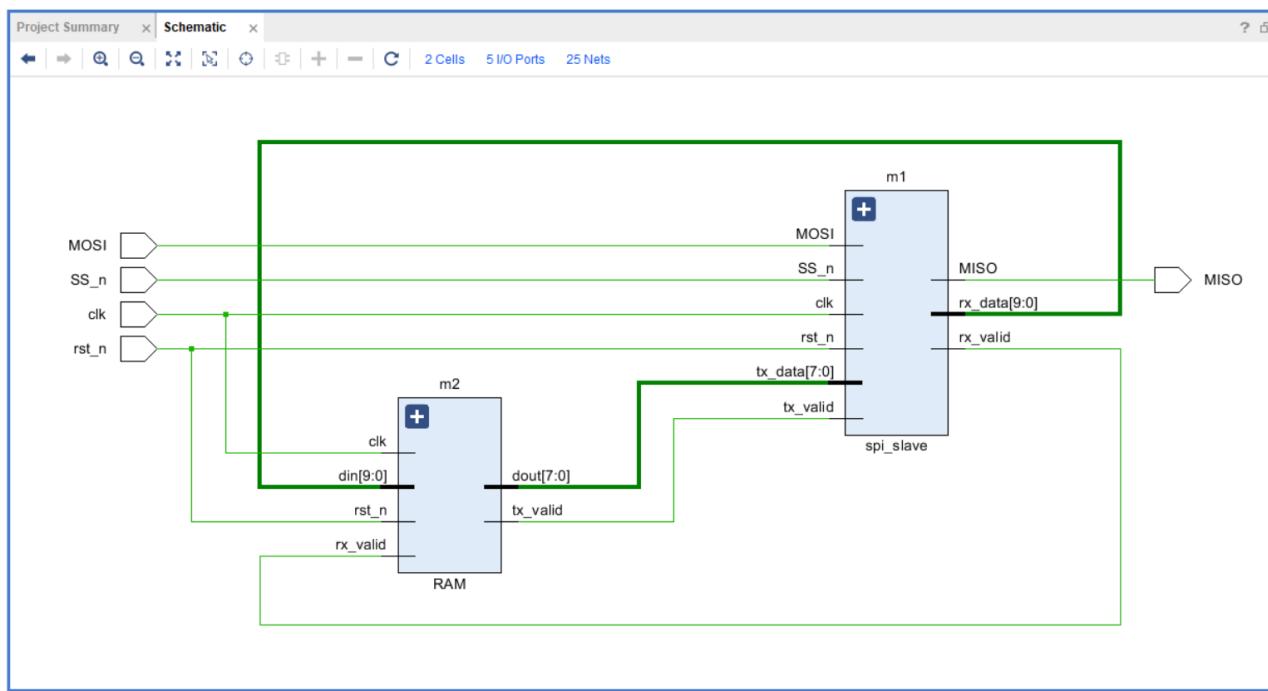
THE IMAGES BELOW SHOW THE RESULTS OF THE LINTING TOOL WITH NO DETECTED ERRORS, AND THE SUCCESSFUL SYNTHESIS OF THE DESIGN:



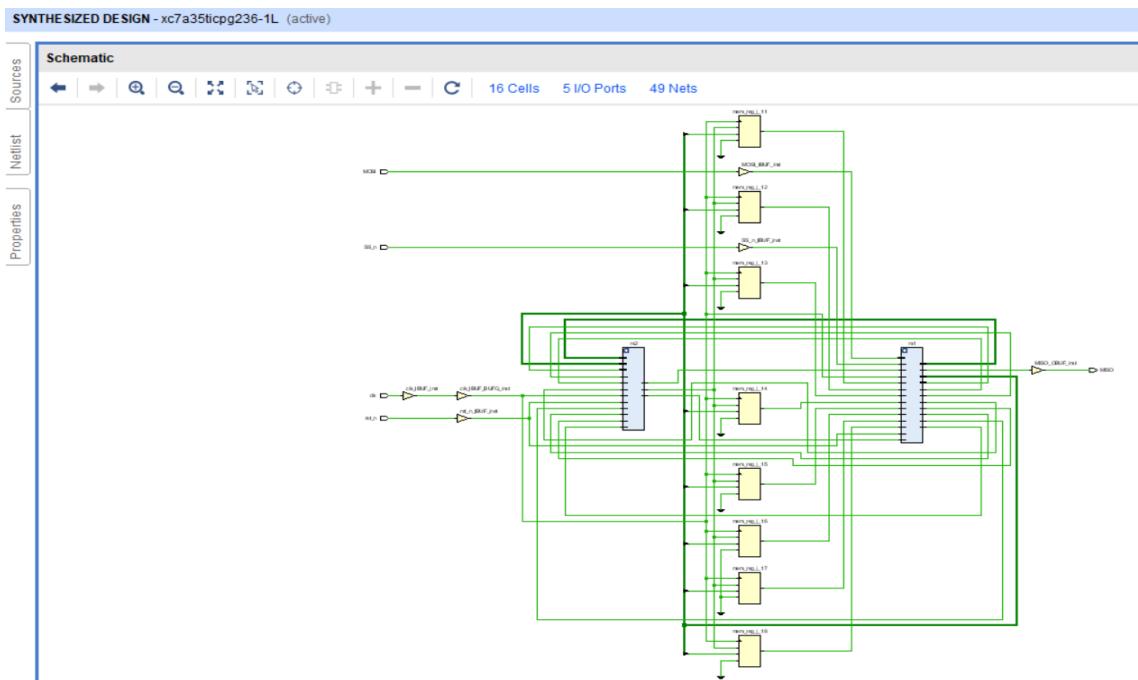
# VIVADO

## 1-SCHEMATIC AND SYNTHESIS

### SCHEMATIC AFTER THE ELABORATION

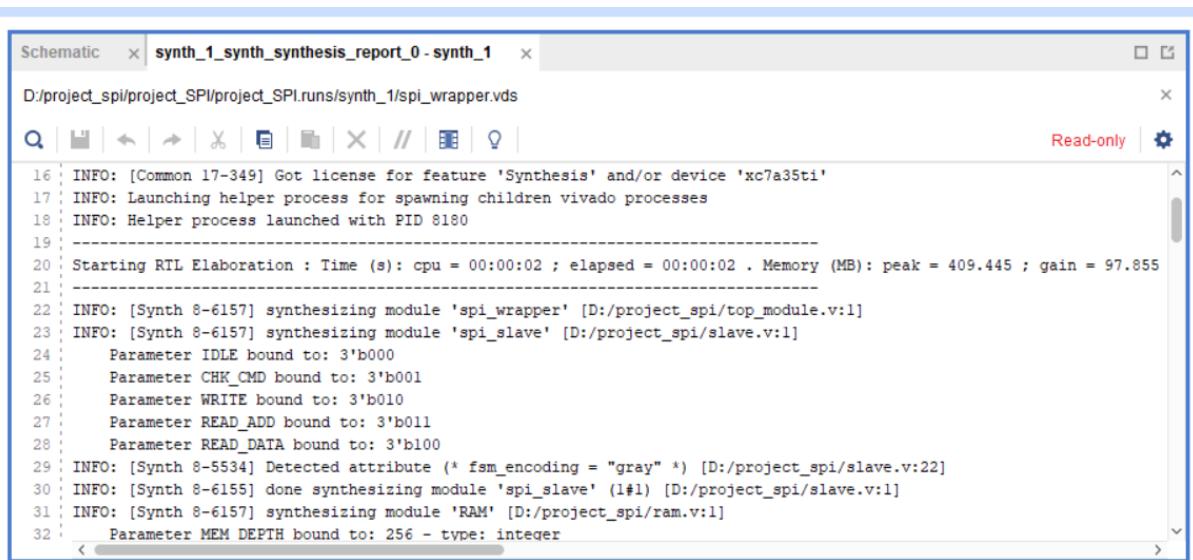


### SCHEMATIC AFTER THE SYNTHESIS



## Synthesis report showing the encoding used

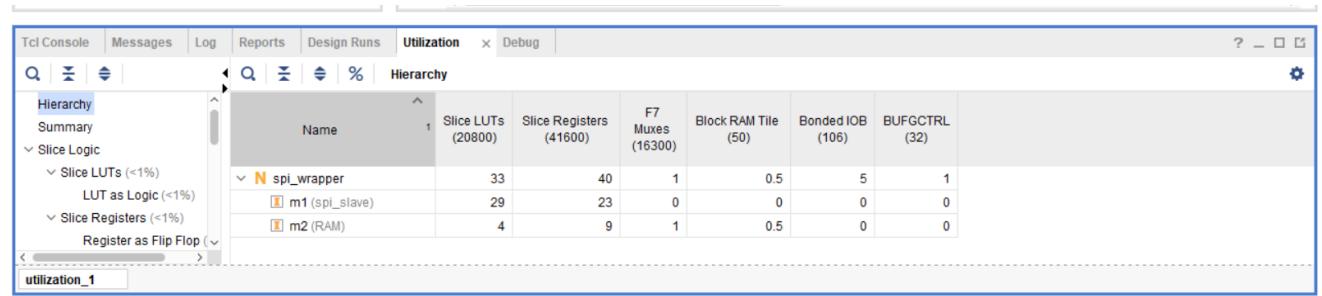
Info : in line 29 synthesis detect (\*fsm\_ecoding=gray\*)



The screenshot shows the Vivado Schematic editor interface. A tab labeled "synth\_1\_synth\_synthesis\_report\_0 - synth\_1" is active. The window displays a synthesis log for the module "spi\_wrapper.v". The log includes various informational messages such as license acquisition, process launching, parameter definitions, and detection of the "fsm\_encoding = "gray"" attribute. The log ends with the message "INFO: [Synth 8-6157] synthesizing module 'spi\_slave' [D:/project\_spi/slave.v:1]".

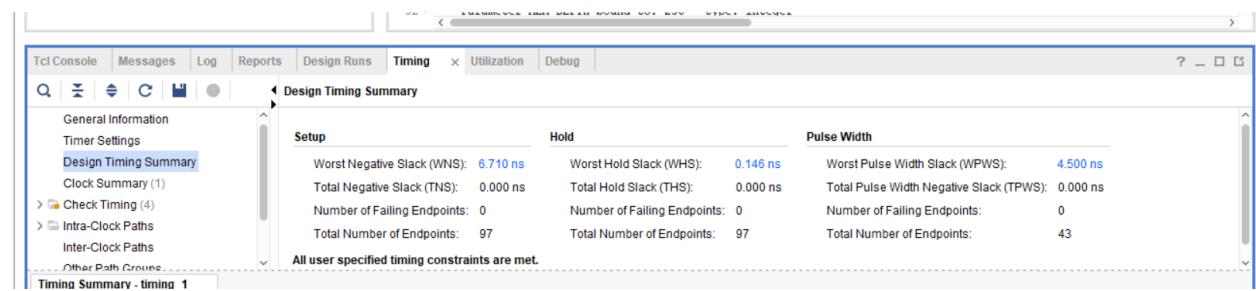
```
INFO: [Common 17-349] Got license for feature 'Synthesis' and/or device 'xc7a35ti'
INFO: Launching helper process for spawning children vivado processes
INFO: Helper process launched with PID 8180
-----
Starting RTL Elaboration : Time (s): cpu = 00:00:02 ; elapsed = 00:00:02 . Memory (MB): peak = 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' (l#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



The screenshot shows the Vivado Utilization Report. The "Utilization" tab is selected. On the left, a hierarchical tree view shows the design structure, with "Slice Logic" expanded to show "Slice LUTs" and "Slice Registers". The main table lists resources for the "spi\_wrapper" module and its sub-modules "m1 (spi\_slave)" and "m2 (RAM)".

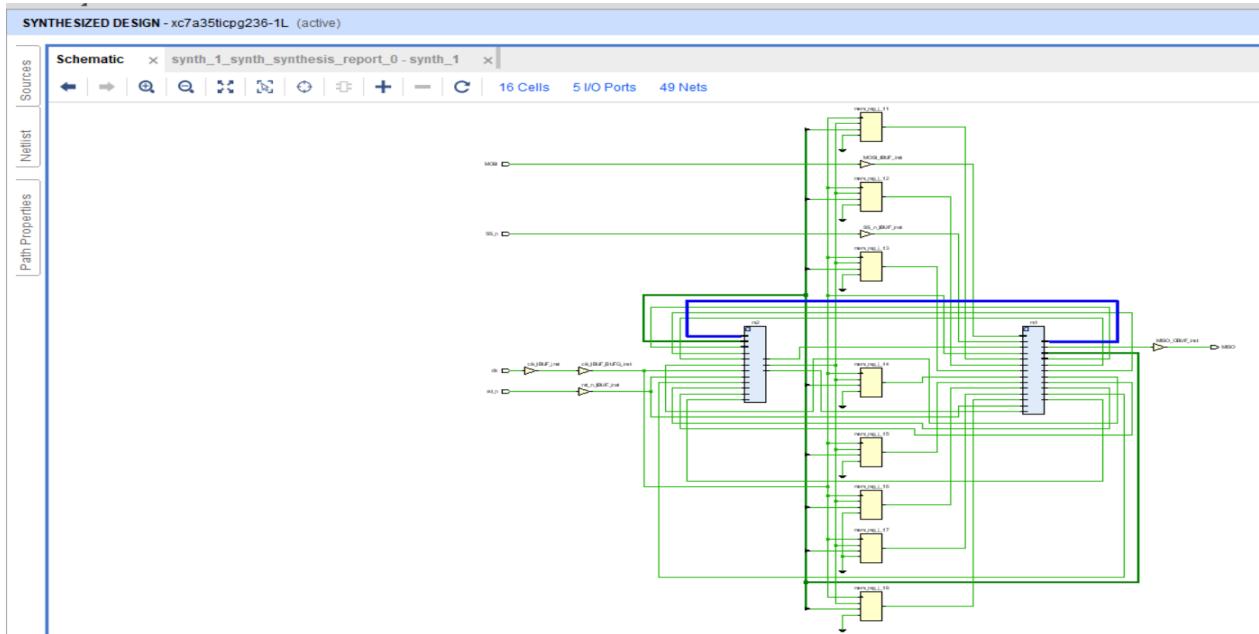
Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
spi_wrapper	33	40	1	0.5	5	1
m1 (spi_slave)	29	23	0	0	0	0
m2 (RAM)	4	9	1	0.5	0	0



The screenshot shows the Vivado Timing Report. The "Timing" tab is selected. The "Design Timing Summary" section displays timing constraints for the design. It shows setup times (WNS: 6.710 ns), hold times (WHS: 0.146 ns), and pulse widths (WPWS: 4.500 ns). The summary states: "All user specified timing constraints are met."

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

## SNIPPET OF THE CRITICAL PATH HIGHLIGHTED IN THE SCHEMATIC



## 2-IMPLEMENTATION SNIPPETS FOR EACH ENCODING

Let's start with gray\_encoding

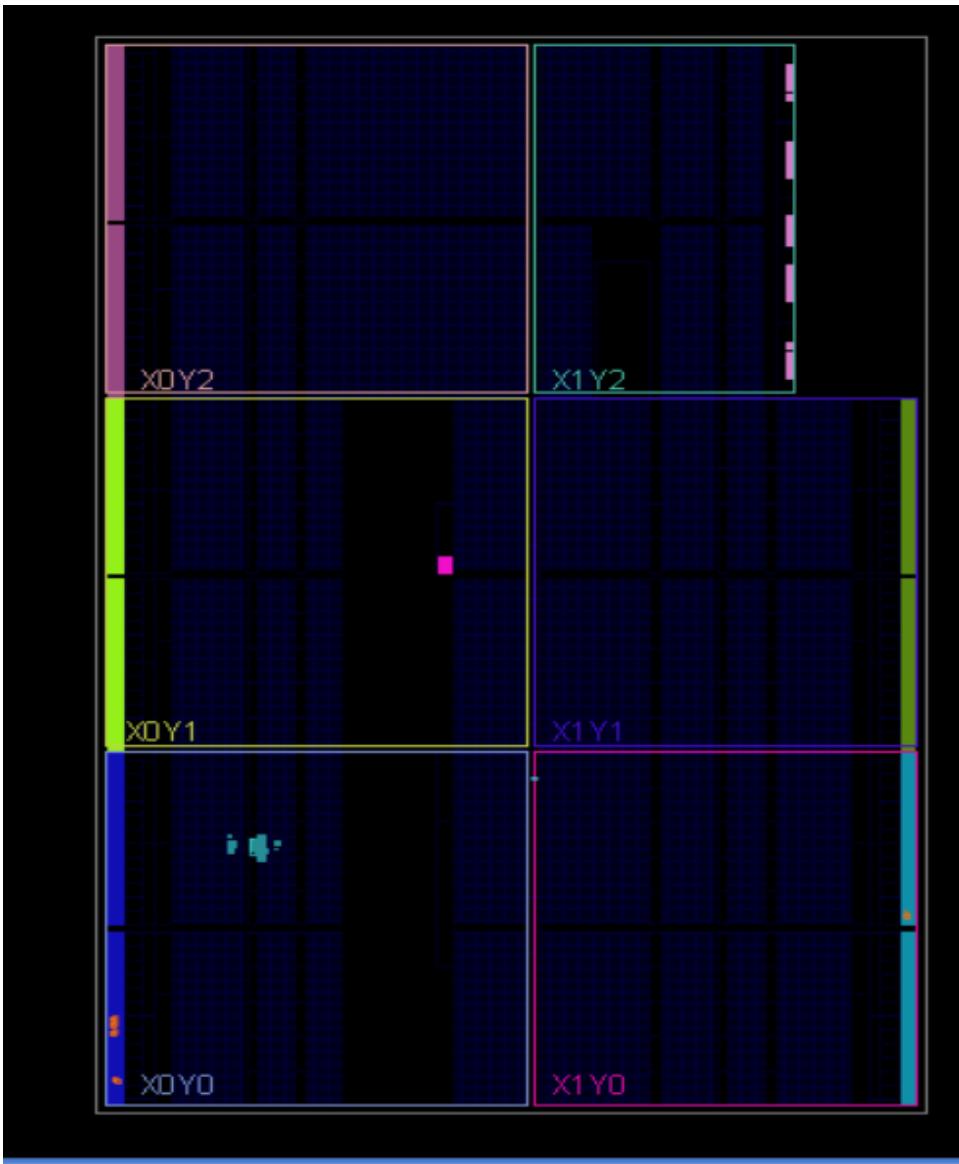
### UTILIZATION

Utilization										
	Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Slice (815 0)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
Hierarchy	spi_wrapper	34	40	1	15	34	16	0.5	5	1
Summary	m1 (spi_slave)	29	23	0	13	29	13	0	0	0
Slice Logic	m2 (RAM)	5	9	1	5	5	0	0.5	0	0

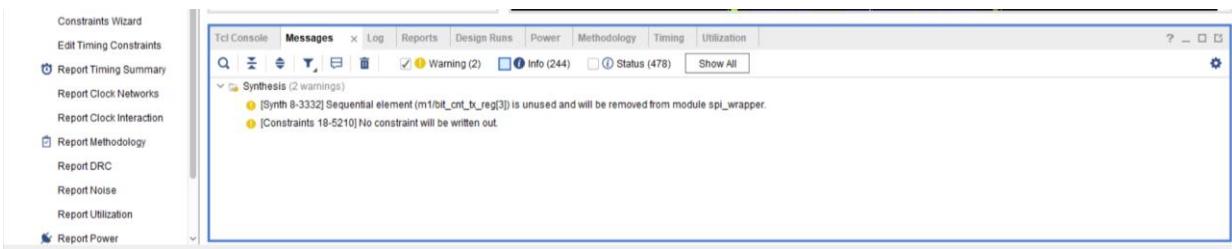
### TIMING

Timing											
	Design Timing Summary			Setup			Hold			Pulse Width	
General Information											
Timer Settings				Worst Negative Slack (WNS):	6.791 ns		Worst Hold Slack (WHS):	0.102 ns		Worst Pulse Width Slack (WPWS):	4.500 ns
Design Timing Summary				Total Negative Slack (TNS):	0.000 ns		Total Hold Slack (THS):	0.000 ns		Total Pulse Width Negative Slack (TPWS):	0.000 ns
Clock Summary (1)				Number of Failing Endpoints:	0		Number of Failing Endpoints:	0		Number of Failing Endpoints:	0
> Check Timing (4)				Total Number of Endpoints:	97		Total Number of Endpoints:	97		Total Number of Endpoints:	43
> Intra-Clock Paths				All user specified timing constraints are met.							
Inter-Clock Paths											
> Other Path Groups											

## DEVICE



**Zero errors and zero critical warnings**



## SECOND : ONE\_HOT ENCODING

### AFTER SYNTHESIS:

The screenshot displays three main windows from the Xilinx Vivado IDE:

- Timing Summary - timing\_1**: Shows Design Timing Summary with Setup, Hold, and Pulse Width sections. All user-specified timing constraints are met.
- Utilization Report - utilization\_1**: Shows resource usage for the implemented design. The table includes columns for Name, Slice LUTs (20800), Slice Registers (41600), F7 Muxes (16300), Block RAM Tile (50), Bonded IOB (106), and BUFCTRL (32). Key entries include spi\_wrapper, m1 (spi\_slave), and m2 (RAM).
- Schematic - synth\_1\_synth\_synthesis\_report\_0 - synth\_1**: Shows the synthesis report for the module synth\_1. The log output is as follows:

```
19: Starting RTL Elaboration : Time (s): cpu = 00:00:02 ; elapsed = 00:00:02 . Memory (MB): peak = 408.867 ; gain = 96.359
20:
21:
22: INFO: [Synth 8-6157] synthesizing module 'spi_wrapper' [D:/project_spi/top_module.v:1]
23: INFO: [Synth 8-6157] synthesizing module 'spi_slave' [D:/project_spi/slave.v:1]
24:     Parameter IDLE bound to: 3'b000
25:     Parameter CHK_CMD bound to: 3'b001
26:     Parameter WRITE bound to: 3'b010
27:     Parameter READ_ADD bound to: 3'b011
28:     Parameter READ_DATA bound to: 3'b100
29: INFO: [Synth 8-5534] Detected attribute (* fsm_encoding = "one_hot" *) [D:/project_spi/slave.v:22]
30: INFO: [Synth 8-6155] done synthesizing module 'spi_slave' (l#1) [D:/project_spi/slave.v:1]
31: INFO: [Synth 8-6157] synthesizing module 'RAM' [D:/project_spi/ram.v:1]
32:     Parameter MEM_DEPTH bound to: 256 - type: integer
33:     Parameter ADDR_SIZE bound to: 8 - type: integer
34: INFO: [Synth 8-6155] done synthesizing module 'RAM' (2#1) [D:/project_spi/ram.v:1]
35: INFO: [Synth 8-61551] done synthesizing module 'spi_wrapper' (3#1) [D:/project_spi/top_module.v:1]
```

## After implementation:

Report Clock Interaction  
 Report Methodology  
 Report DRC  
 Report Noise  
 Report Utilization  
 Report Power  
 Schematic  
 PROGRAM AND DEBUG  
 Generate Bitstream  
 Open Hardware Manager

Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Slices (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
<b>spi_wrapper</b>	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

Implementation Complete ✓  
 Default Layout

Project Summary | Device

Design Timing Summary

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 6.684 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: 104	Total Number of Endpoints: 104	Total Number of Endpoints: 46

All user specified timing constraints are met.

## THIRD : SEQUENTIAL

### AFTER SYNTHESIZING

#### timing and utilization

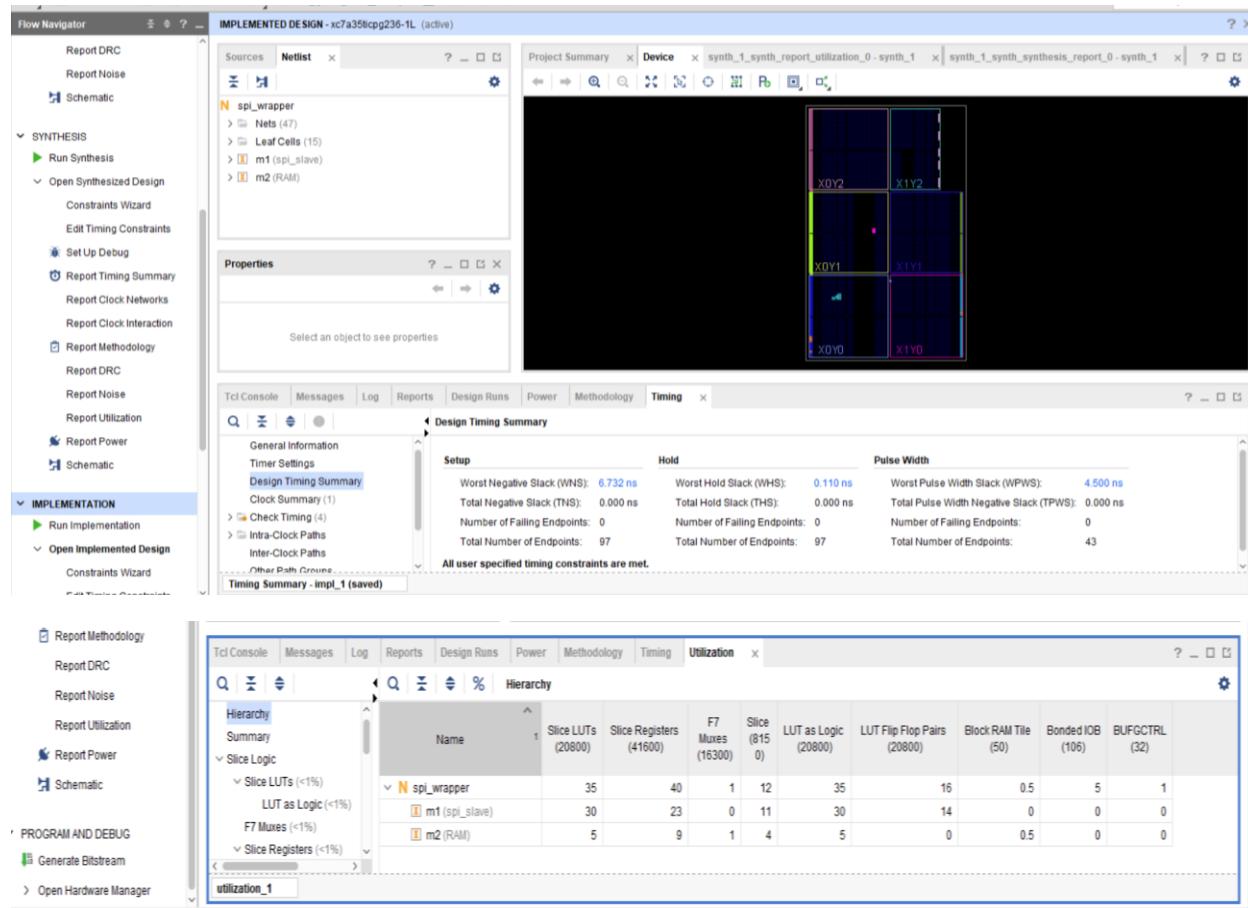
The image displays three separate windows from the Vivado IDE, each showing synthesis results for a project named "synth\_1".

- Timing Summary Window:** This window is titled "Design Timing Summary". It shows timing constraints across three categories: Setup, Hold, and Pulse Width. Key values include Worst Negative Slack (WNS) at 6.710 ns, Worst Hold Slack (WHS) at 0.146 ns, and Worst Pulse Width Slack (WPWS) at 4.500 ns. A message at the bottom states "All user specified timing constraints are met."
- Utilization Report Window:** This window is titled "Report Utilization". It displays a detailed utilization report for the "spi\_wrapper" module. The table below shows the distribution of resources across various components.

Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
spi_wrapper	34	40	1	0.5	5	1
m1 (spi_slave)	30	23	0	0	0	0
m2 (RAM)	4	9	1	0.5	0	0

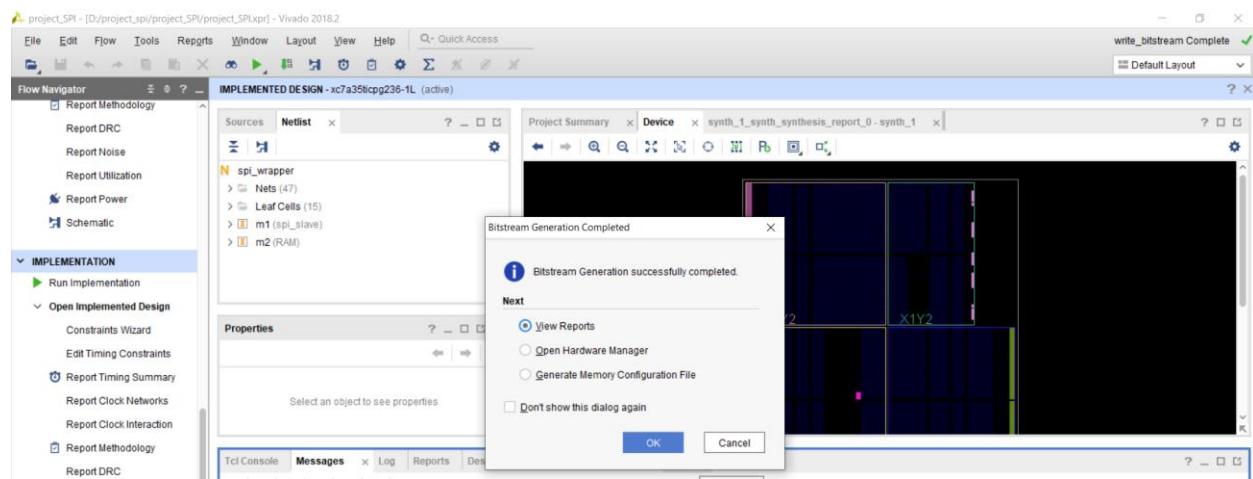
- Synthesis Log Window:** This window is titled "synth\_1\_synth\_report\_utilization\_0 - synth\_1" and shows the command-line output of the synthesis process. The log includes messages about license acquisition, helper process launching, and the start of RTL elaboration. It also lists the synthesis of modules like "spi\_wrapper", "spi\_slave", and "RAM".

## After implementation:

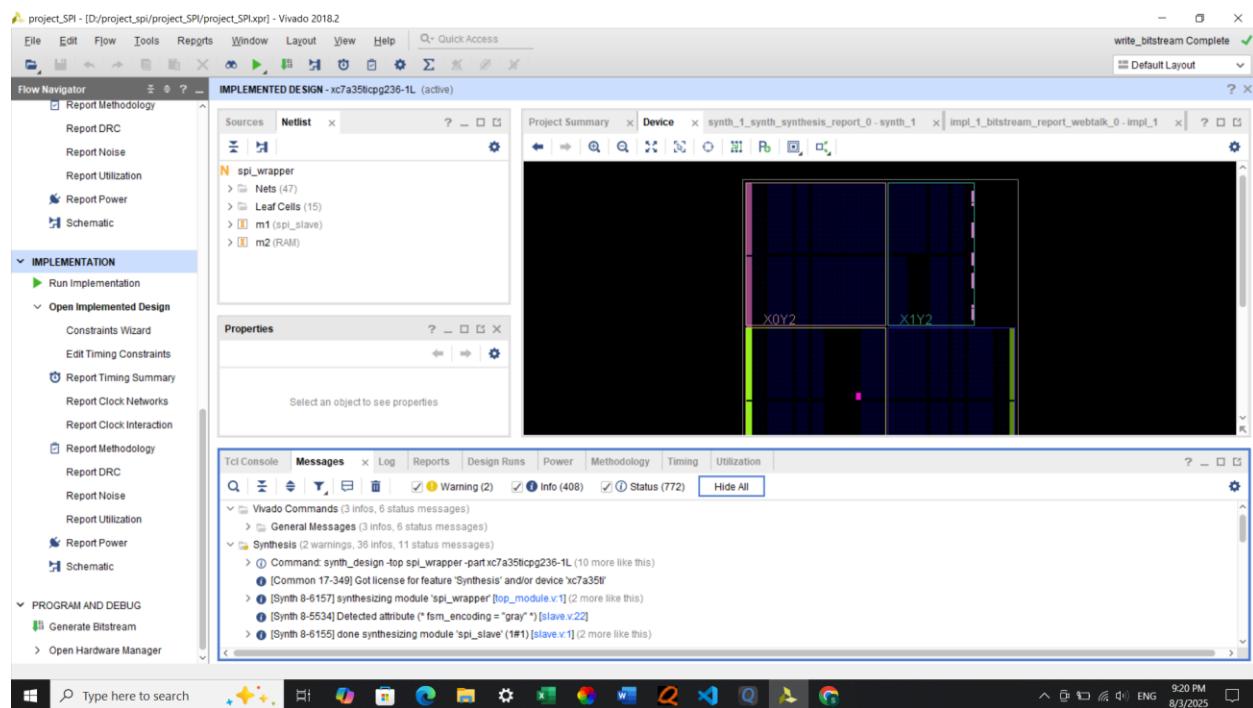


# BASED ON TIME WE FOUND THAT GRAY\_ENCODING IS BETTER

NOW LETS GENERATE BITSTREAM FOR GRAY\_ENCODING:

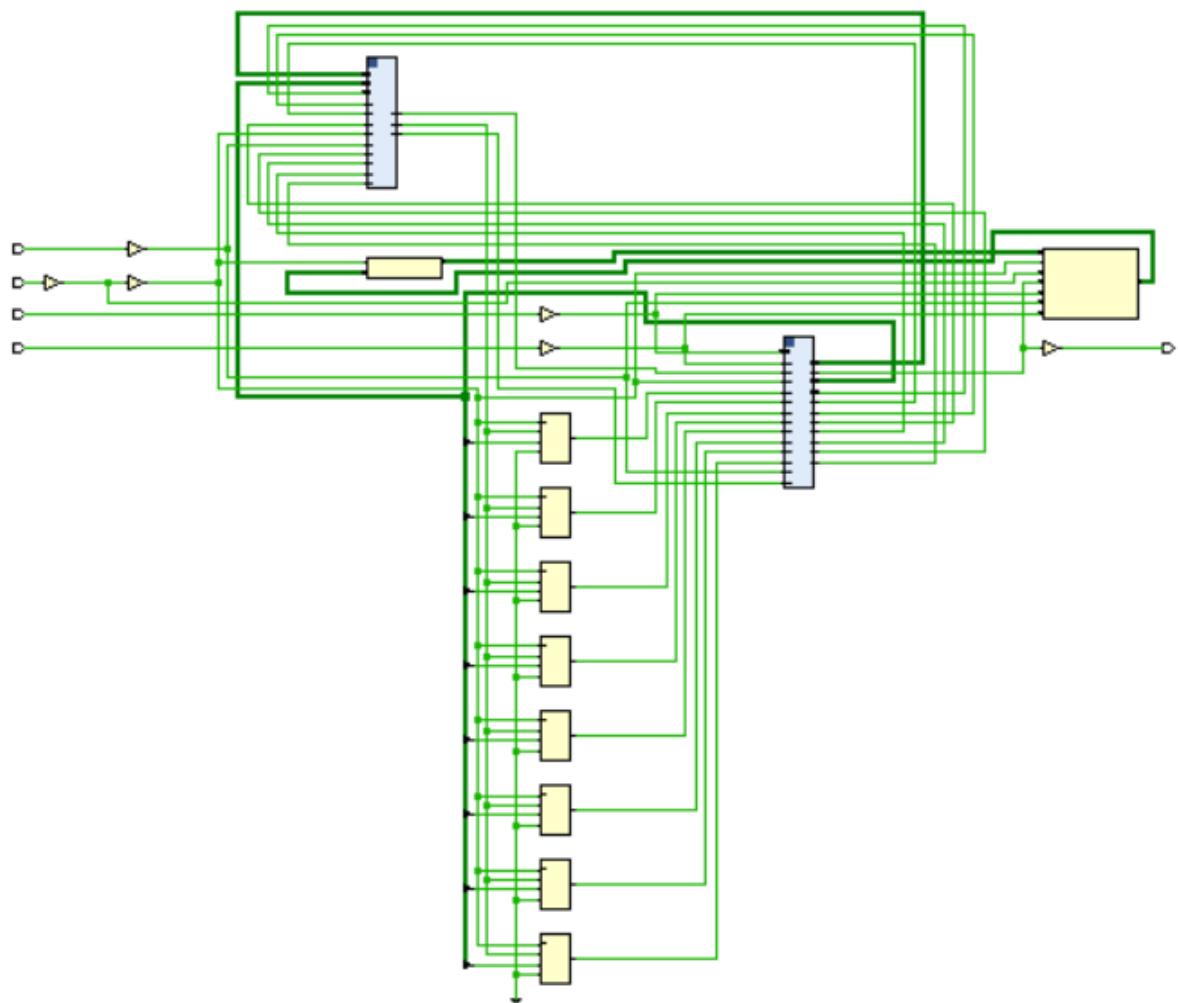


## ZERO ERRORS AND ZERO CRITICAL WARNINGS

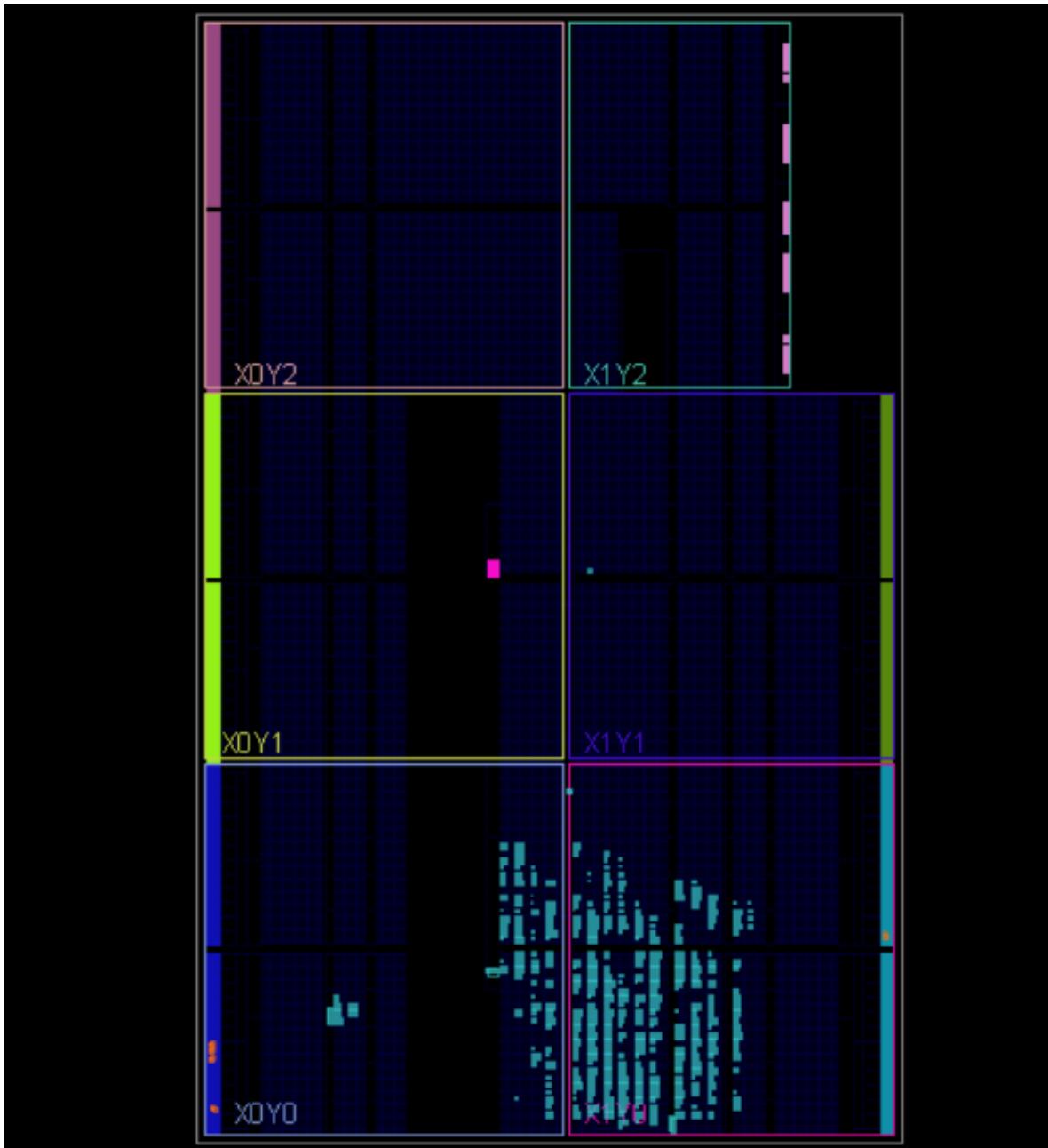


**DEBUG**

**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]

```

