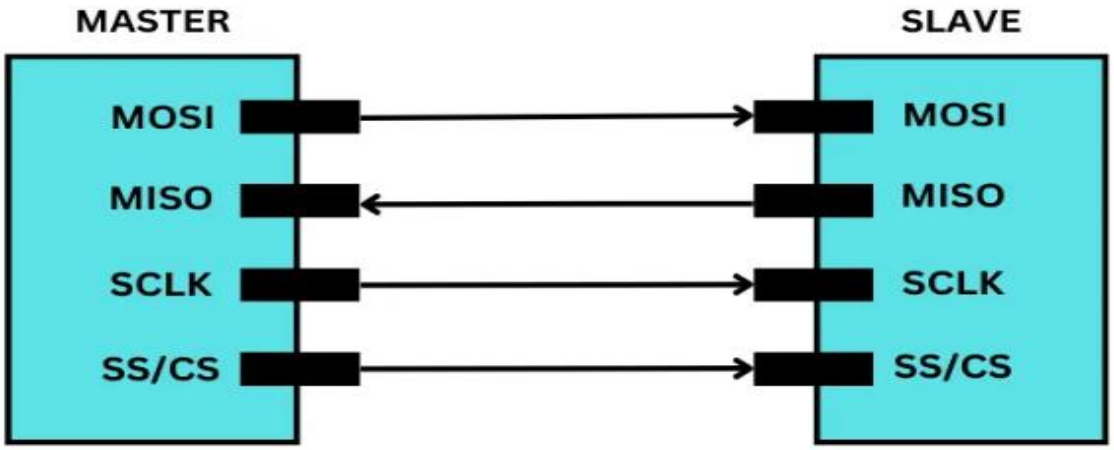
Created By	Ismail Raafat Fawzy

SPI_Interface



Introduction to SPI Communication Protocol:

The Serial Peripheral Interface (SPI) is a synchronous serial communication protocol developed by Motorola, widely used for short-distance communication in embedded systems. SPI allows for high-speed data transfer between a master device and one or more peripheral devices. It is particularly valued for its simplicity, efficiency, and low pin count, making it an essential protocol in digital design and embedded systems.

Definition of SPI Communication Protocol

SPI operates in a master-slave configuration, where the master device initiates communication and controls the clock signal. The key signals in SPI communication are:

- **MOSI (Master Out Slave In):** Data line for transmission from master to slave.
- **MISO (Master In Slave Out):** Data line for transmission from slave to master.
- **SCLK (Serial Clock):** Clock signal generated by the master to synchronize data transfer.
- **SS (Slave Select):** Control line for selecting the slave device.

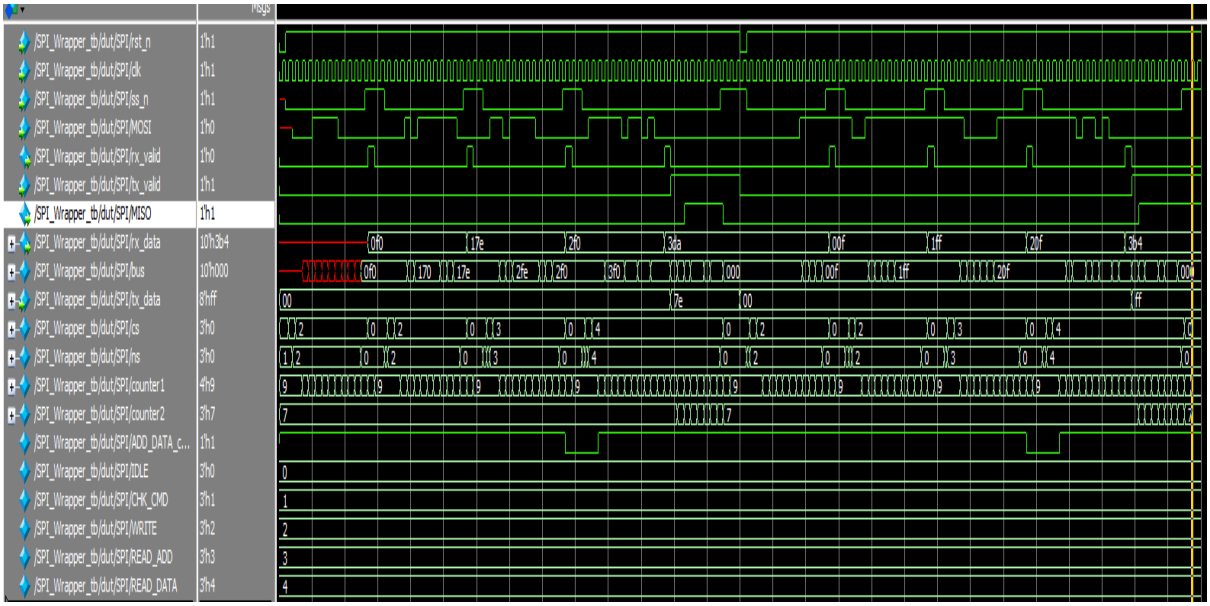
Data transfer in SPI is full-duplex, meaning data can be sent and received simultaneously. The protocol supports different modes of operation, determined by the clock polarity (CPOL) and clock phase (CPHA) settings, allowing flexibility in communication.

Importance of SPI Communication Protocol

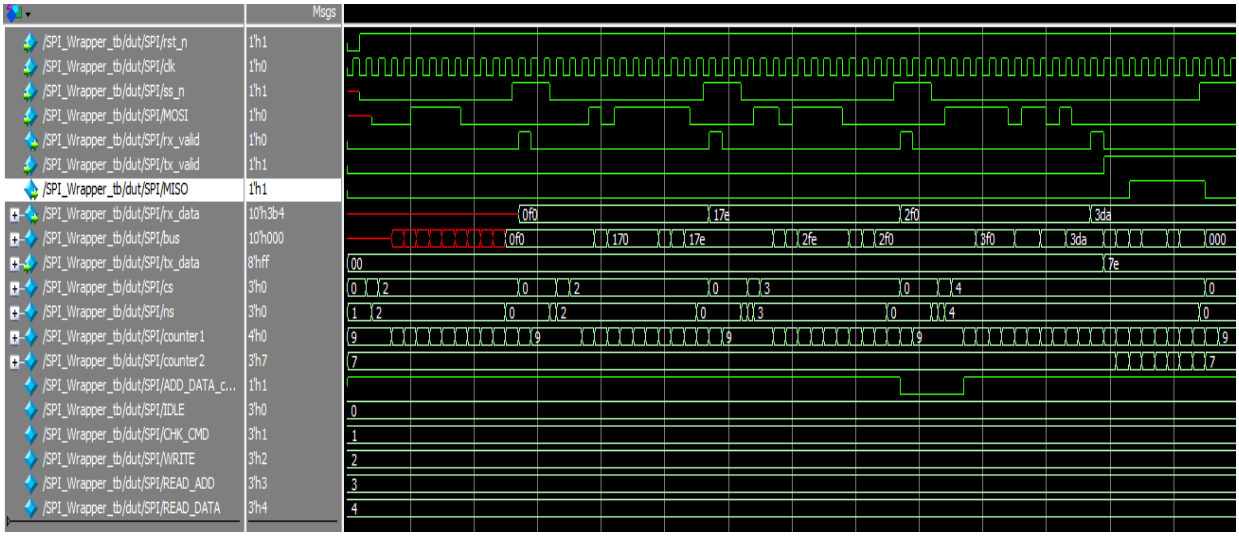
SPI is crucial in embedded systems and digital design for several reasons:

1. **Speed:** SPI supports high-speed data transfers, making it suitable for applications requiring quick data exchanges, such as sensors, memory devices, and display controllers.
2. **Simplicity:** The protocol's straightforward design and minimal overhead make it easy to implement and debug, reducing development time and complexity.
3. **Efficiency:** SPI's full-duplex communication and streamlined signal lines contribute to efficient data transmission, minimizing latency and maximizing throughput.
4. **Scalability:** SPI can connect multiple slave devices to a single master, using individual slave select lines, allowing for scalable and expandable system designs.
5. **Versatility:** SPI is compatible with a wide range of devices, from simple sensors to complex microcontrollers, making it a versatile choice for various applications.

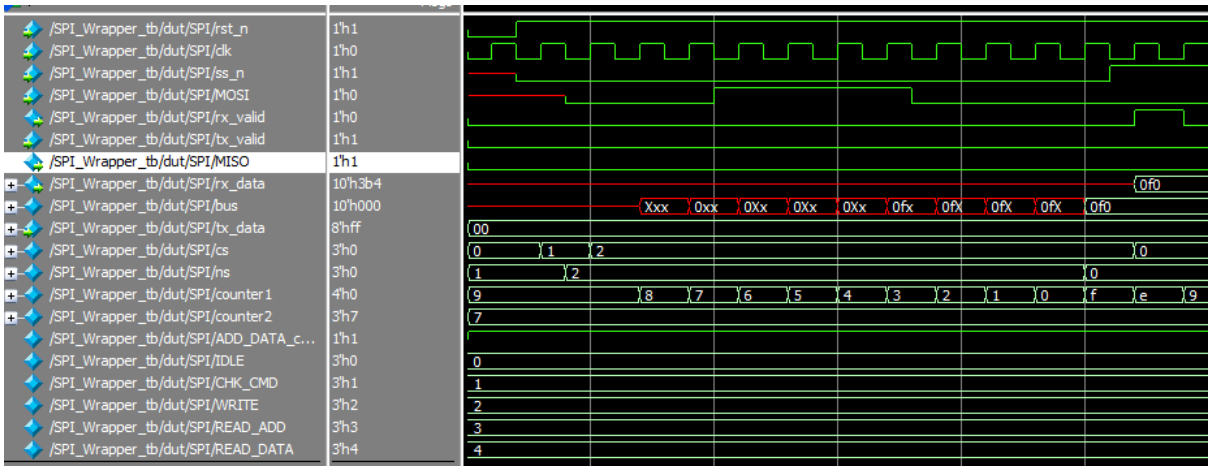
Snapshot of the whole wave form:



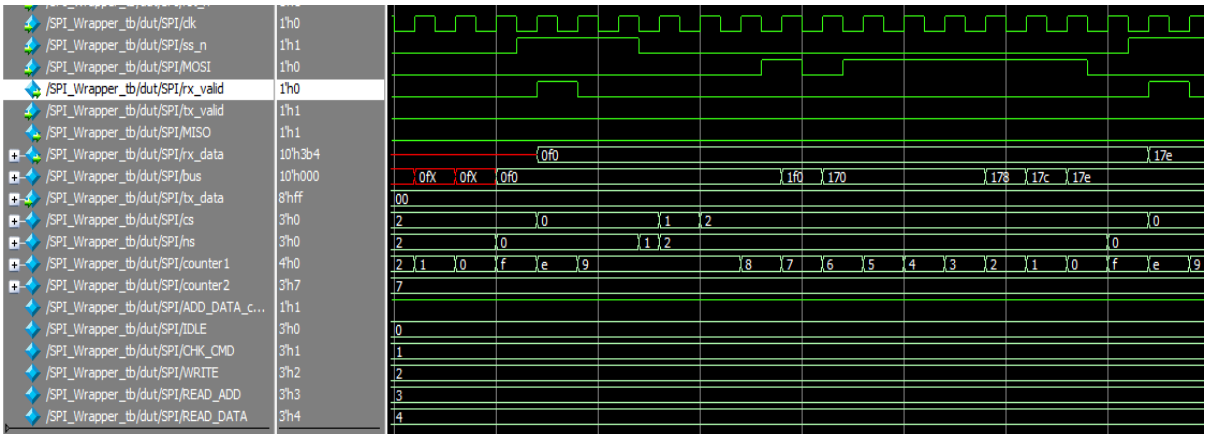
First scenario:



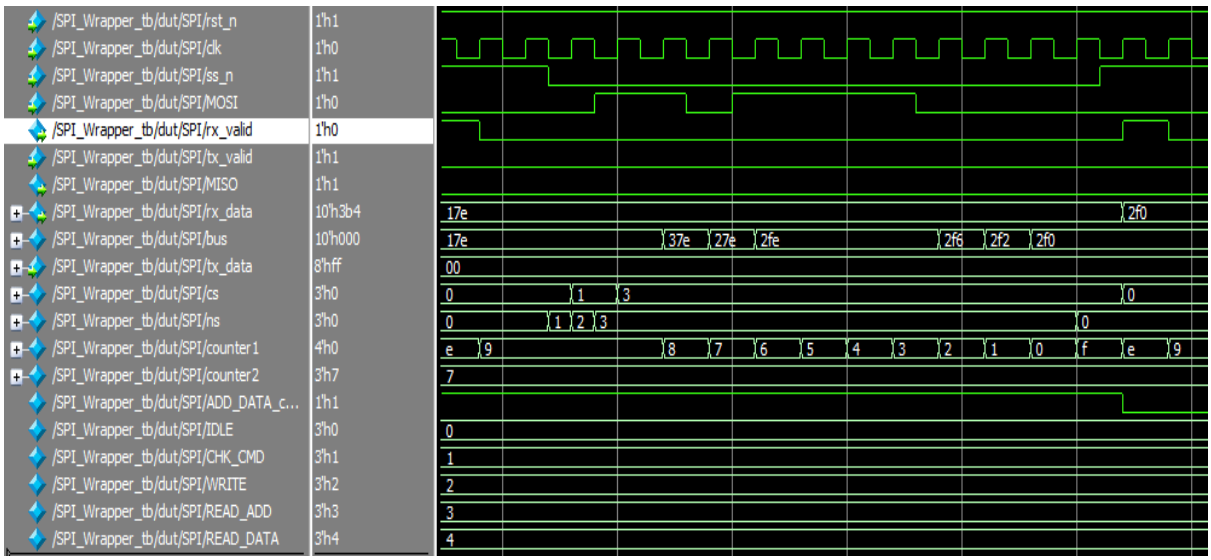
- WRITE state (Address):



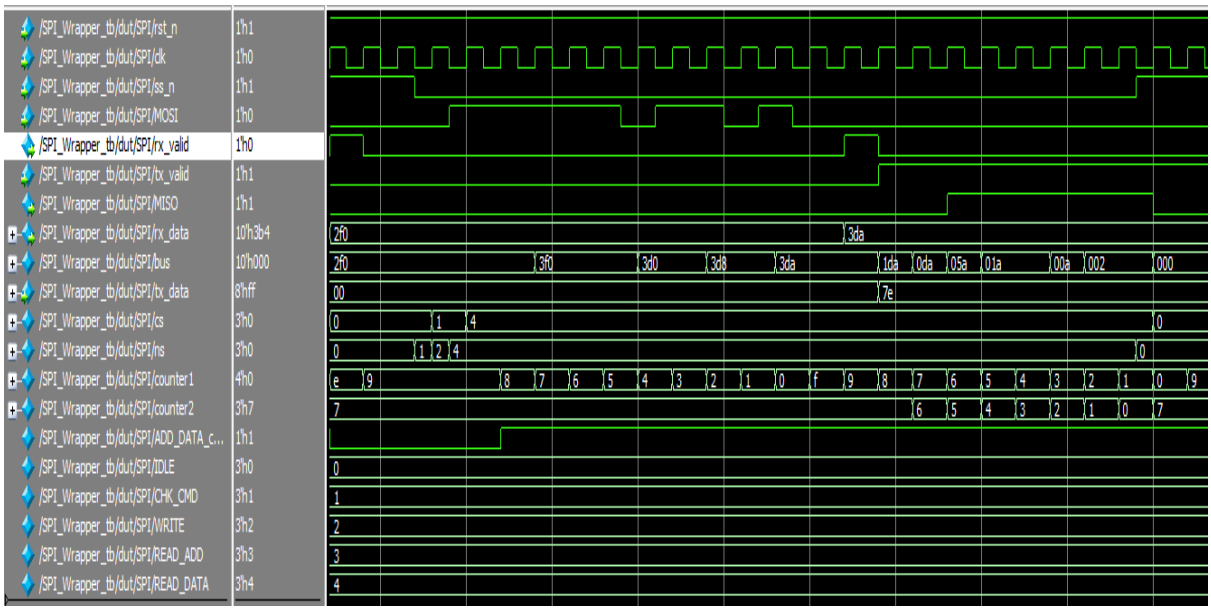
- WRITE state (Data):



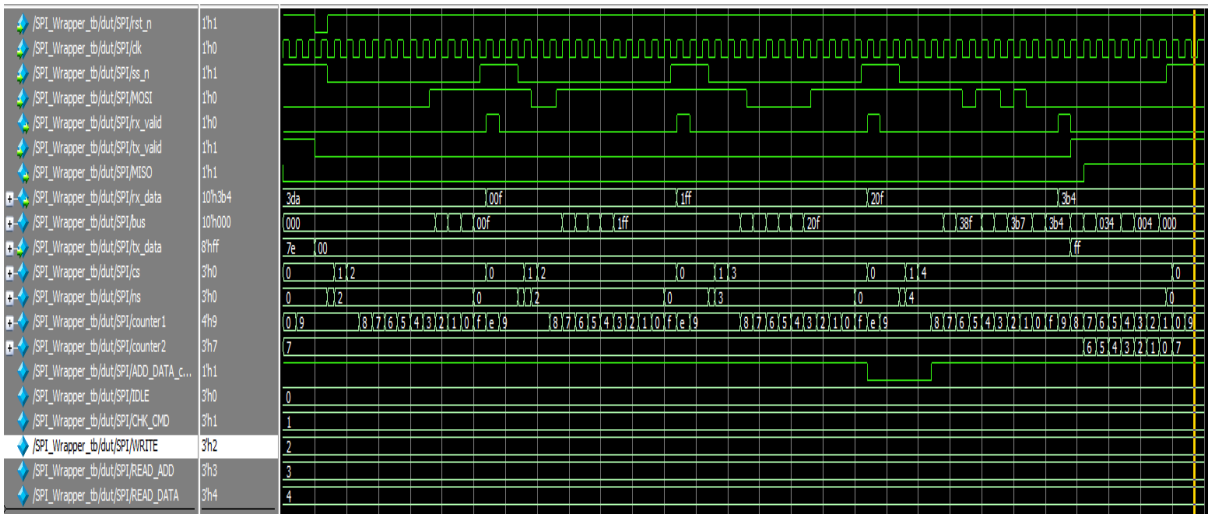
- READ_ADD state:



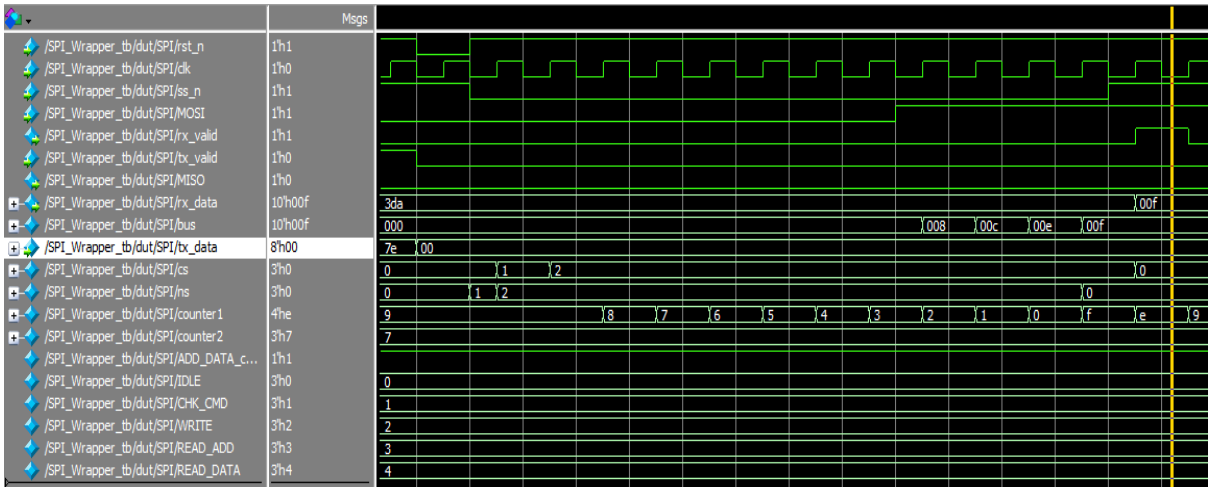
- READ_DATA state:



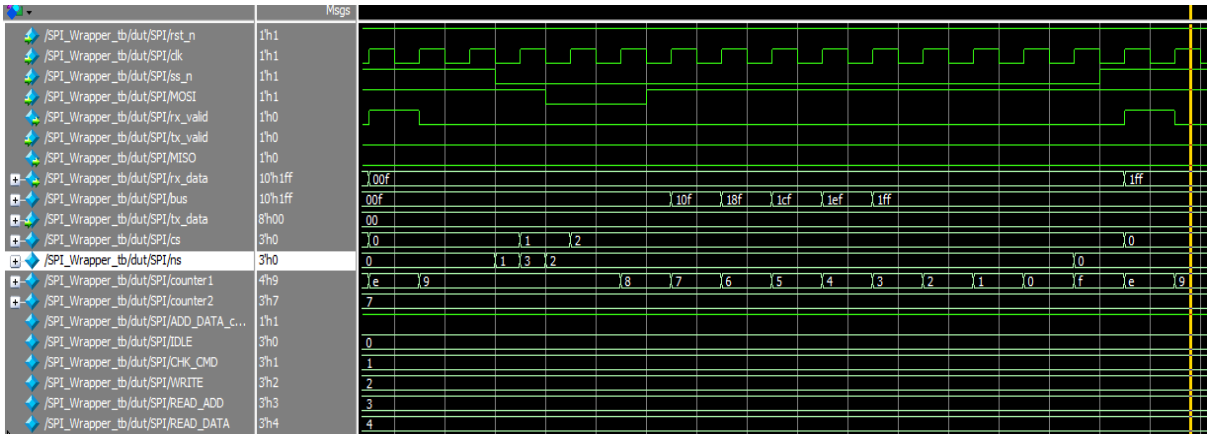
Second scenario:



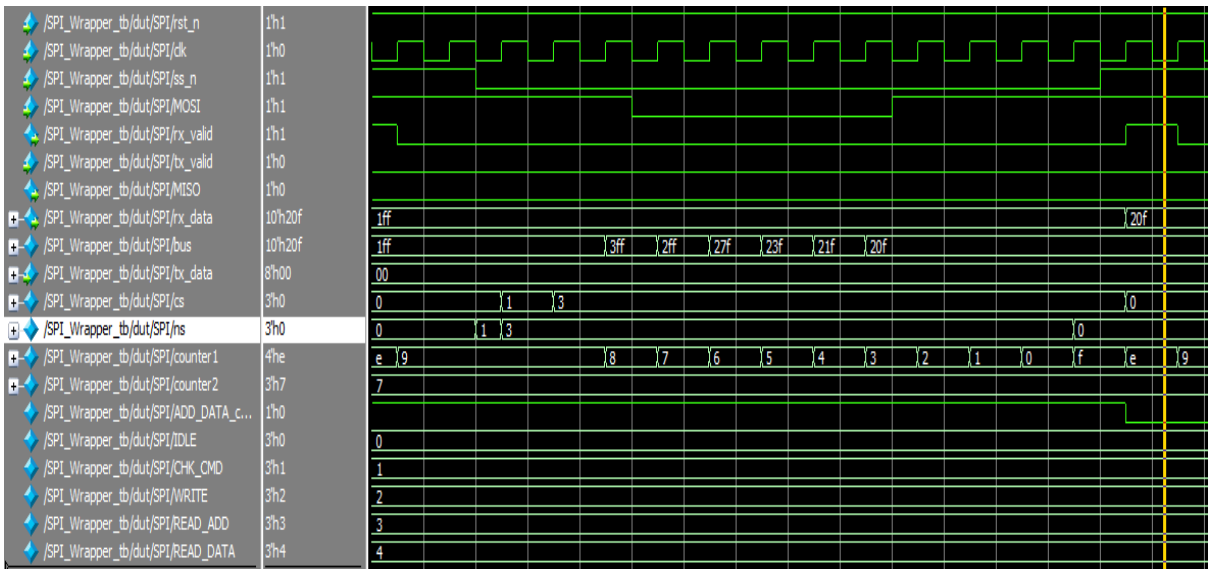
- WRITE state (Address):



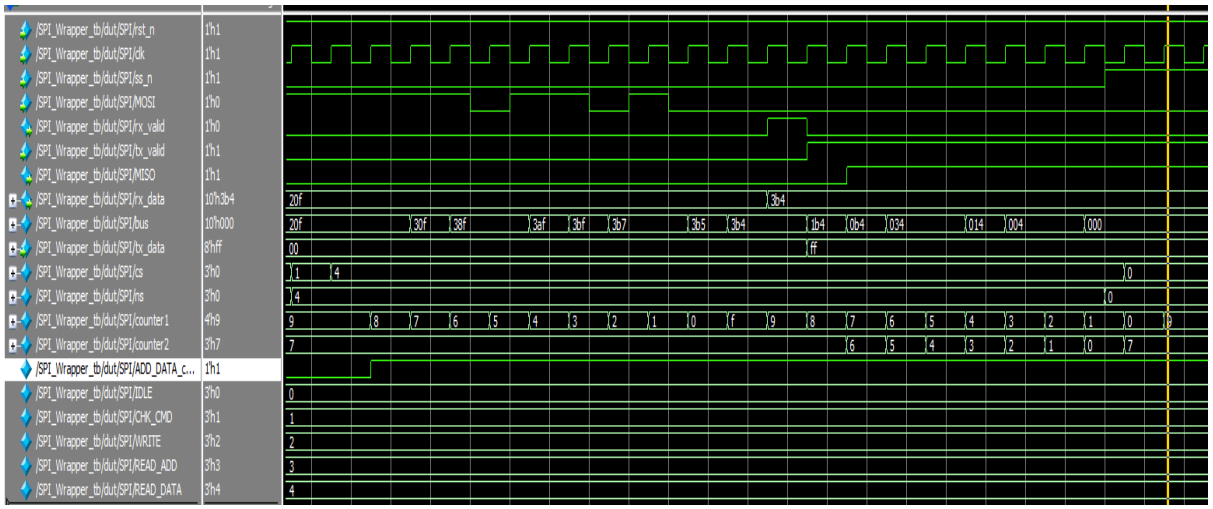
- WRITE state (Data):



- READ_ADD state:

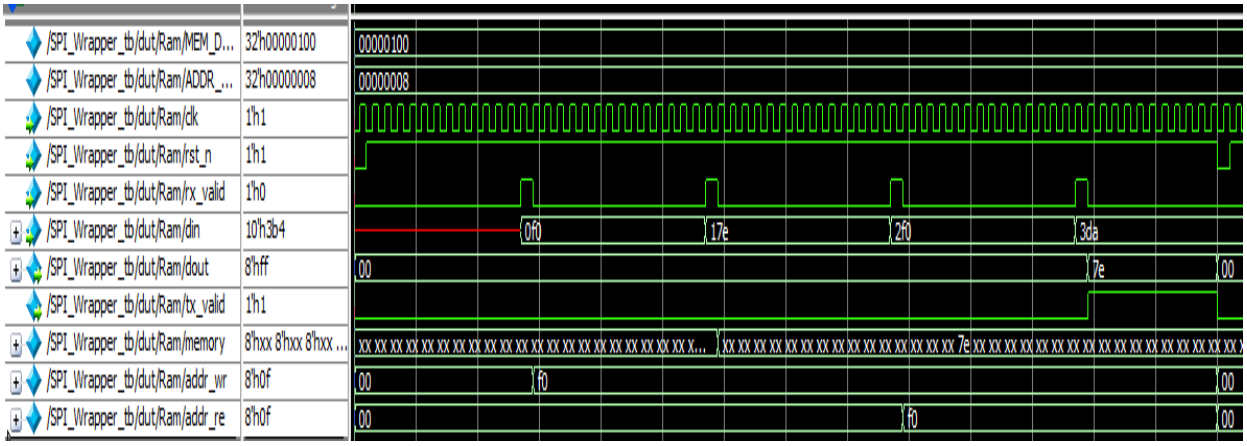


- READ_DATA state:

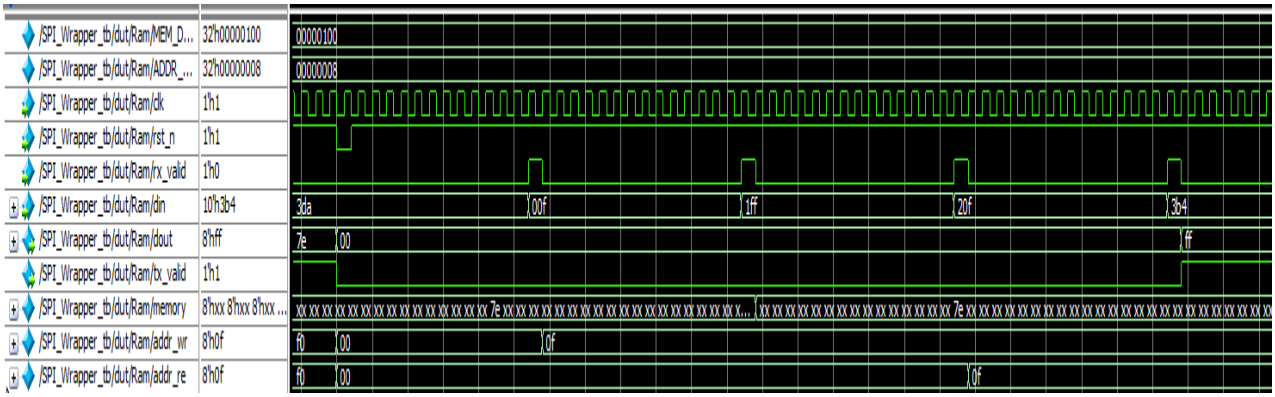


Snapshots of RAM:

- First scenario:

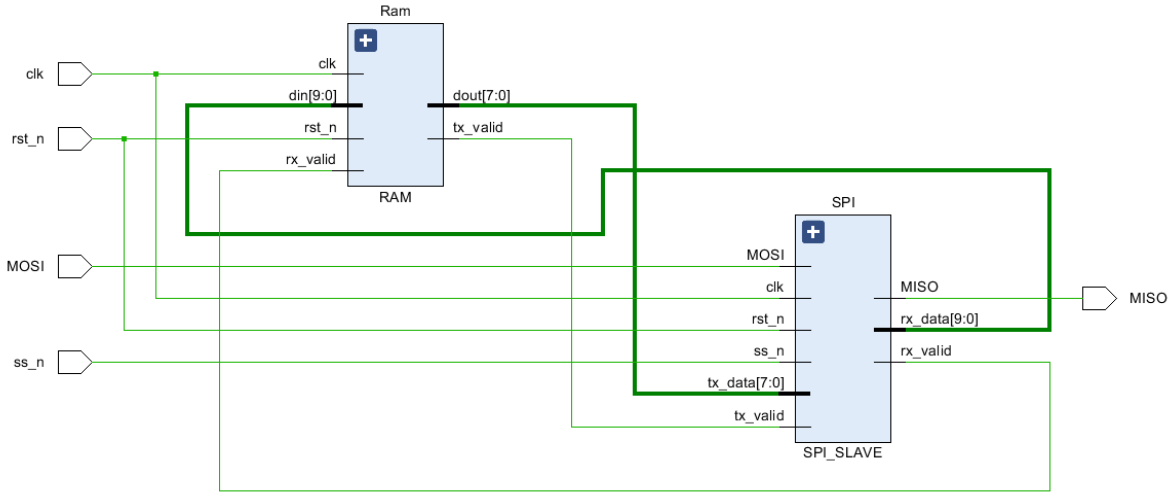


- Second scenario:

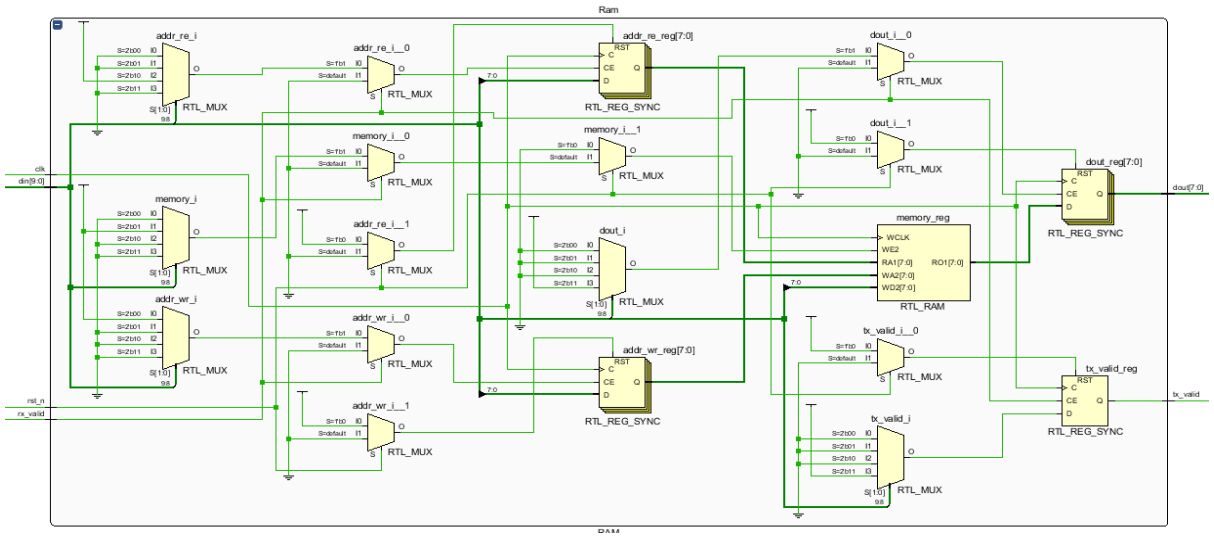


Vivado:

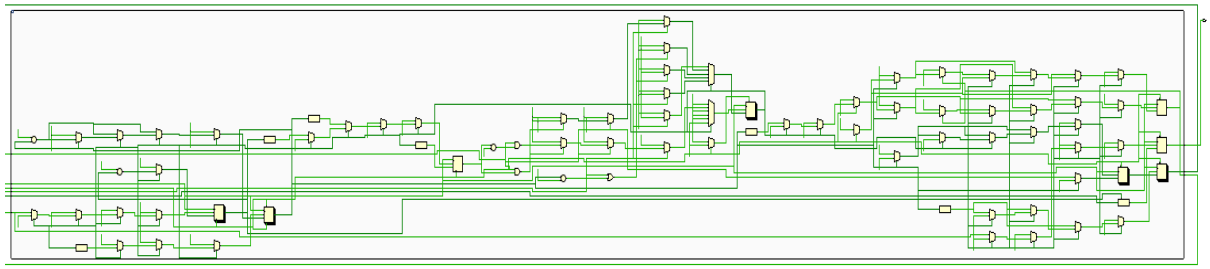
- Elaboration:
  - System:



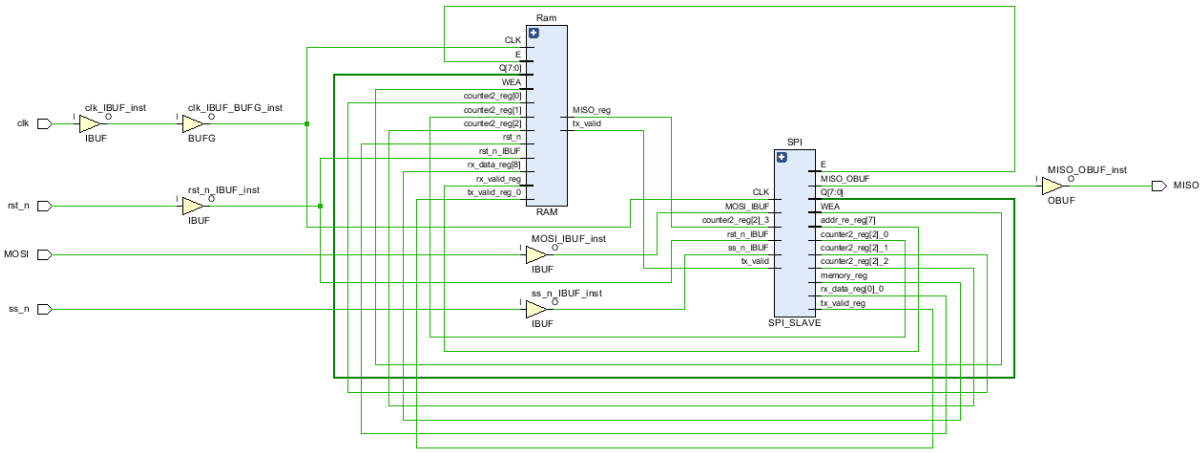
  - Ram:



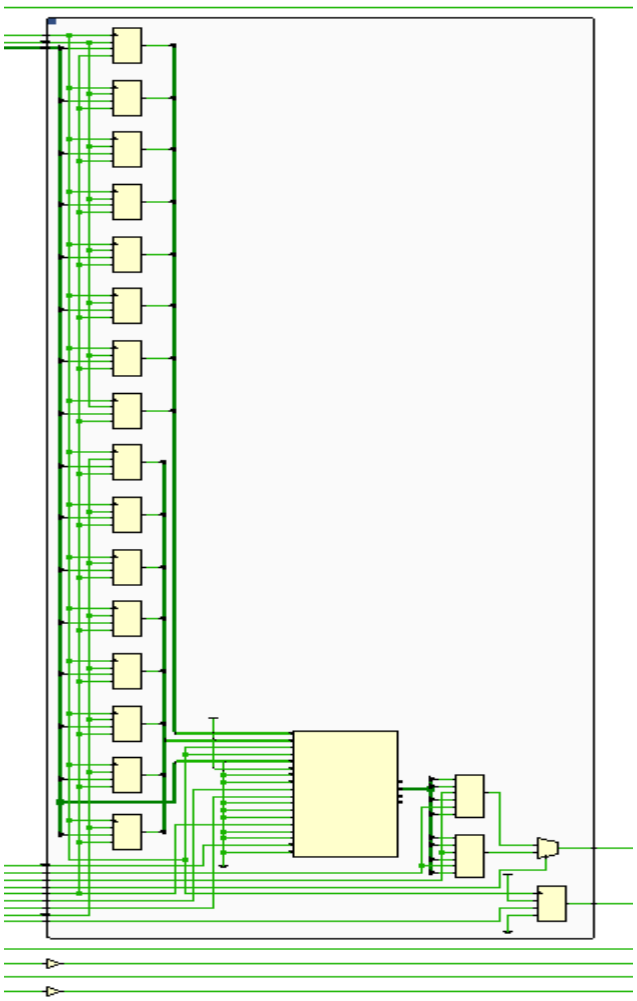
  - SPI:



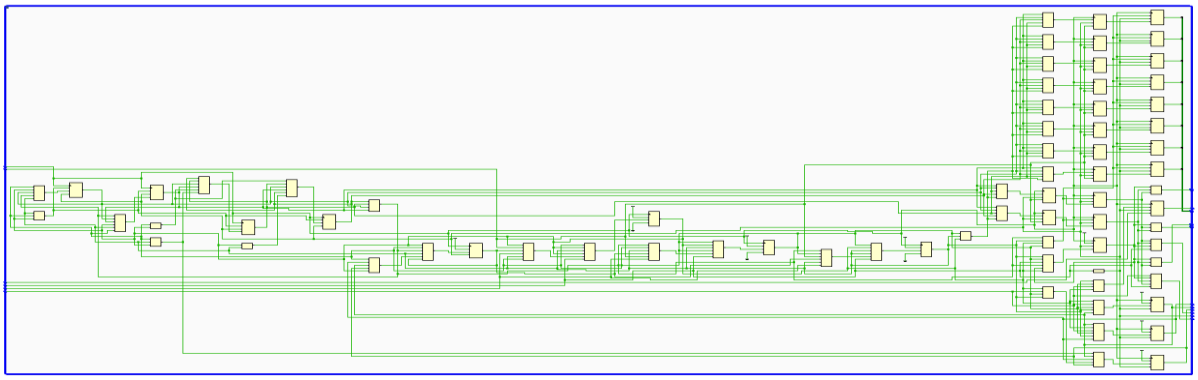
- **Synthesis:**
  - **Seq:**
    - Schematic:
      - System:



      - Ram:



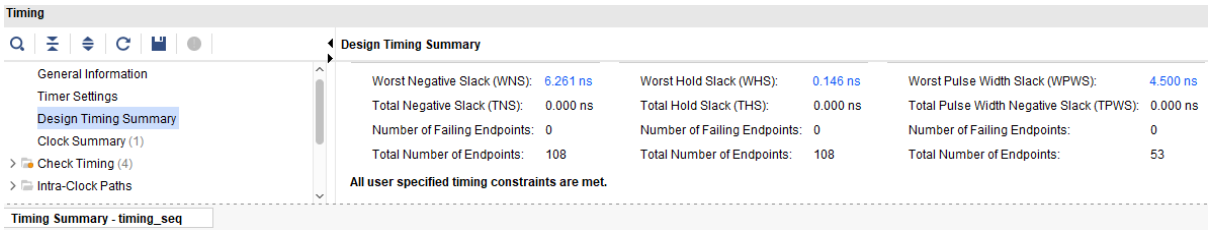
- SPI:



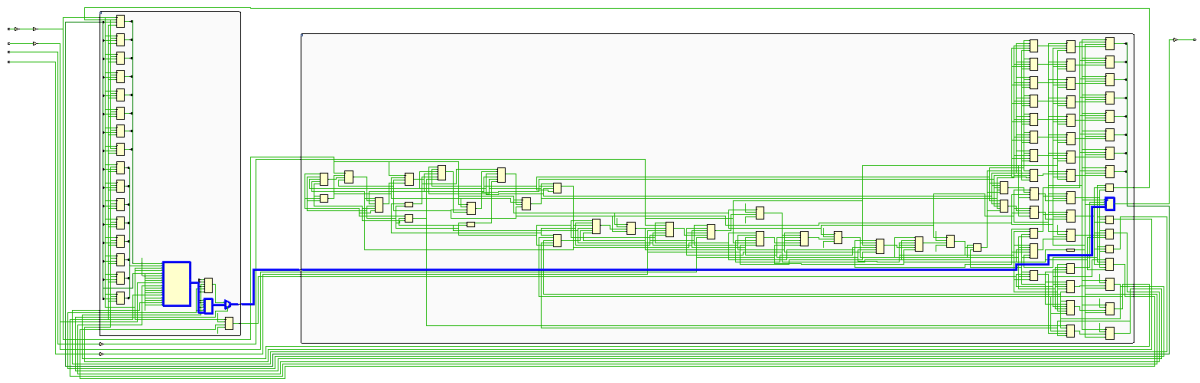
  - Encoding Report:

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

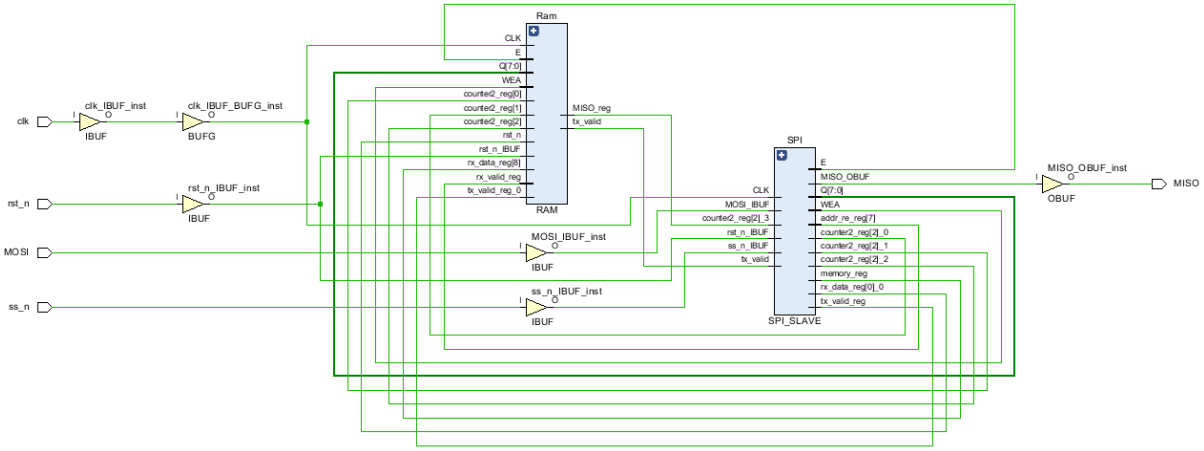
  - Timing Summary:



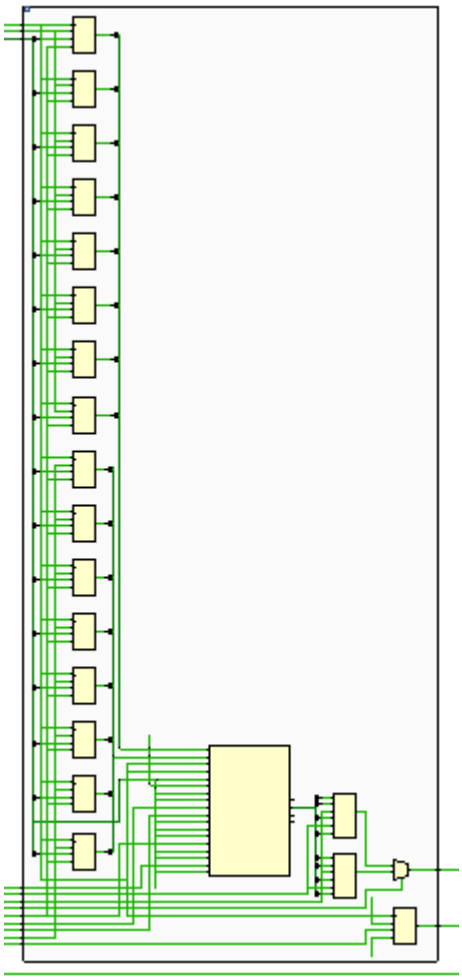
  - Critical Path:



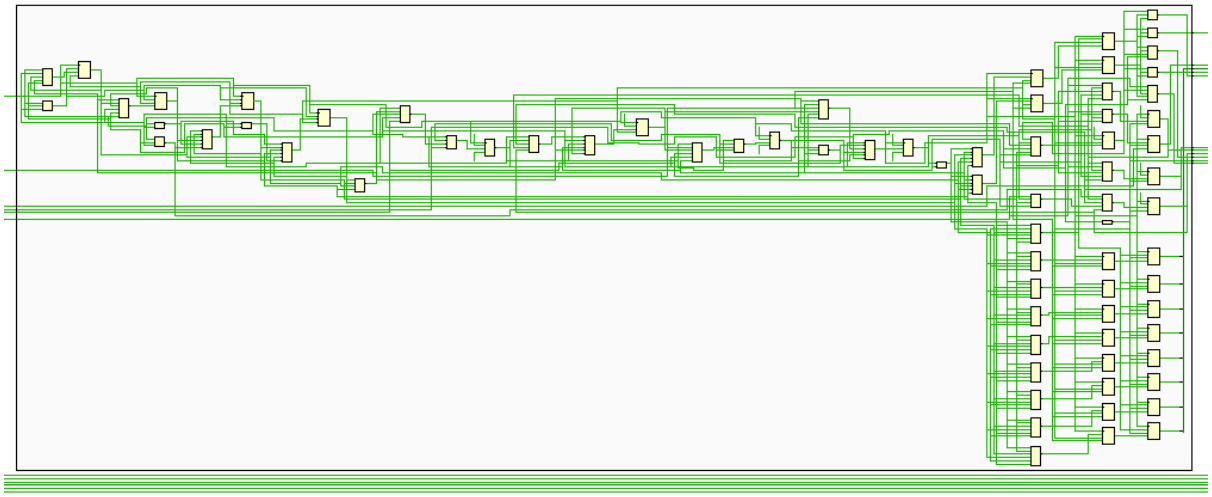
- **Gray:**
  - Schematic:
    - System:



    - Ram:



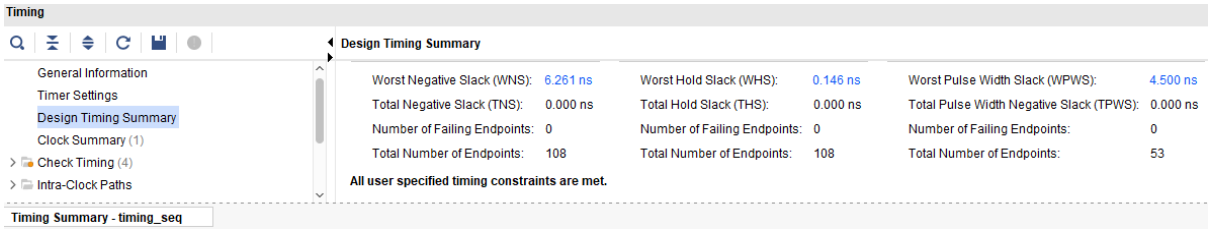
- SPI:



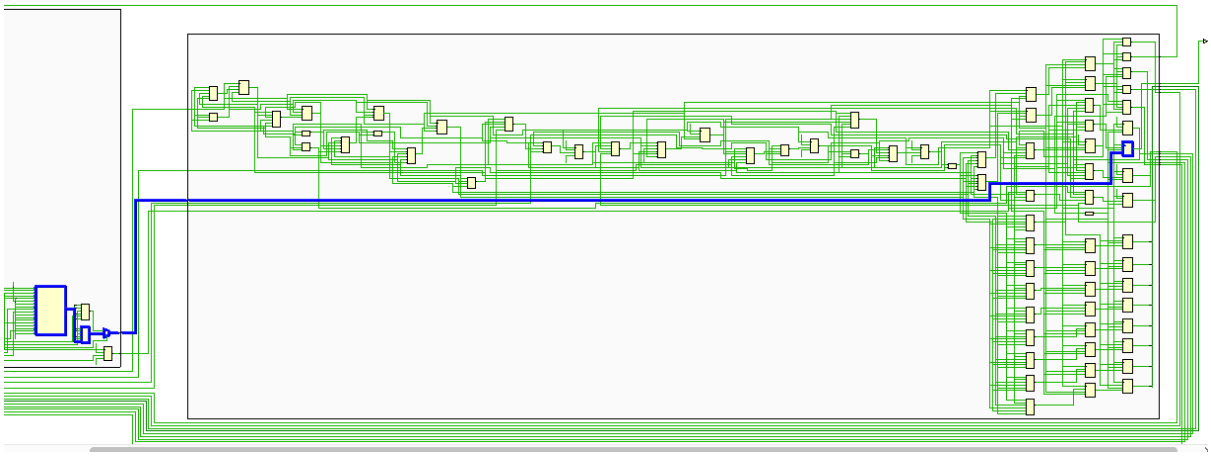
- Encoding Report:

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

- Timing Summary:



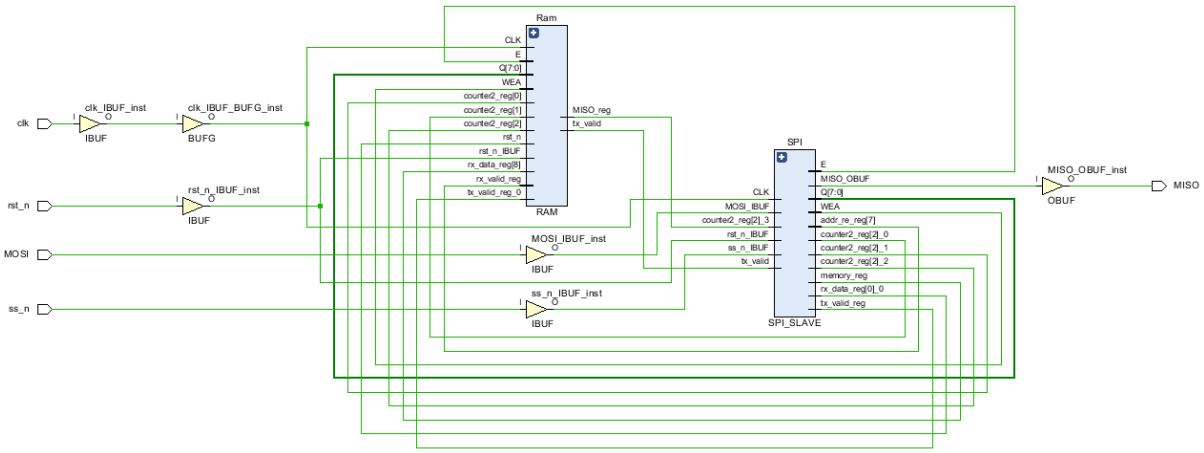
- Critical Path:



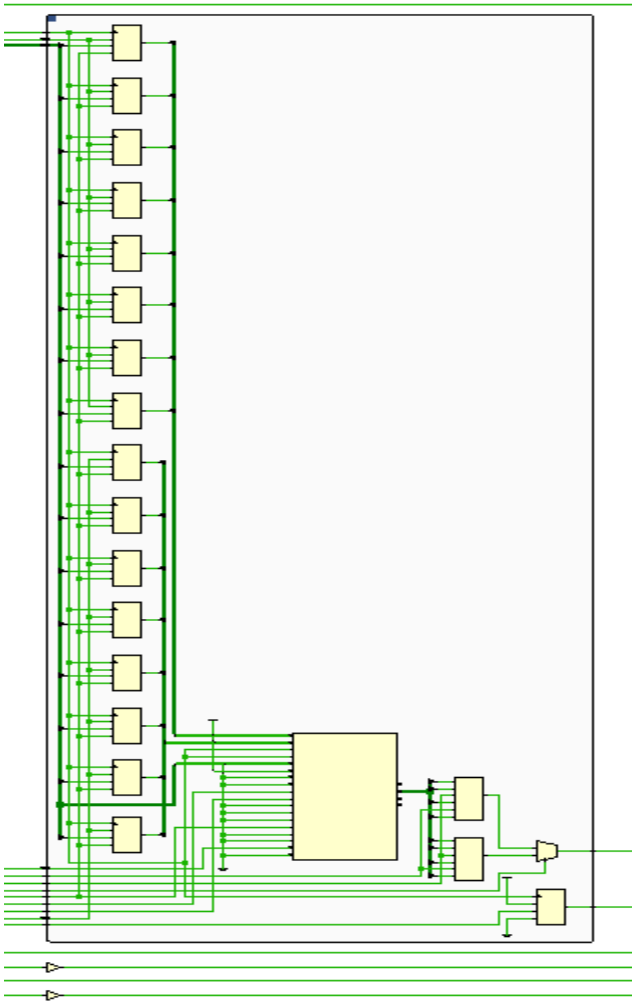
One_Hot:

Schematic:

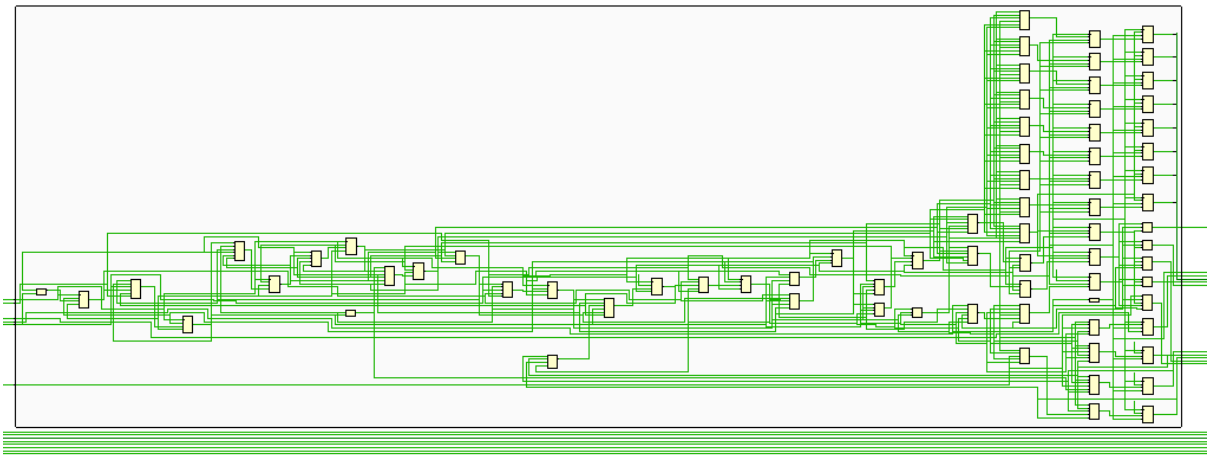
- System:



- Ram:



- SPI:

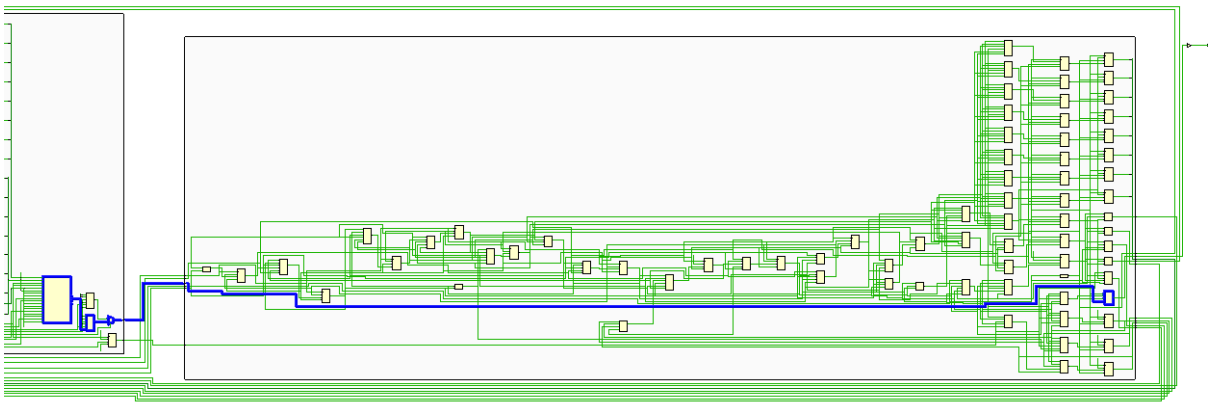


- Encoding Report:

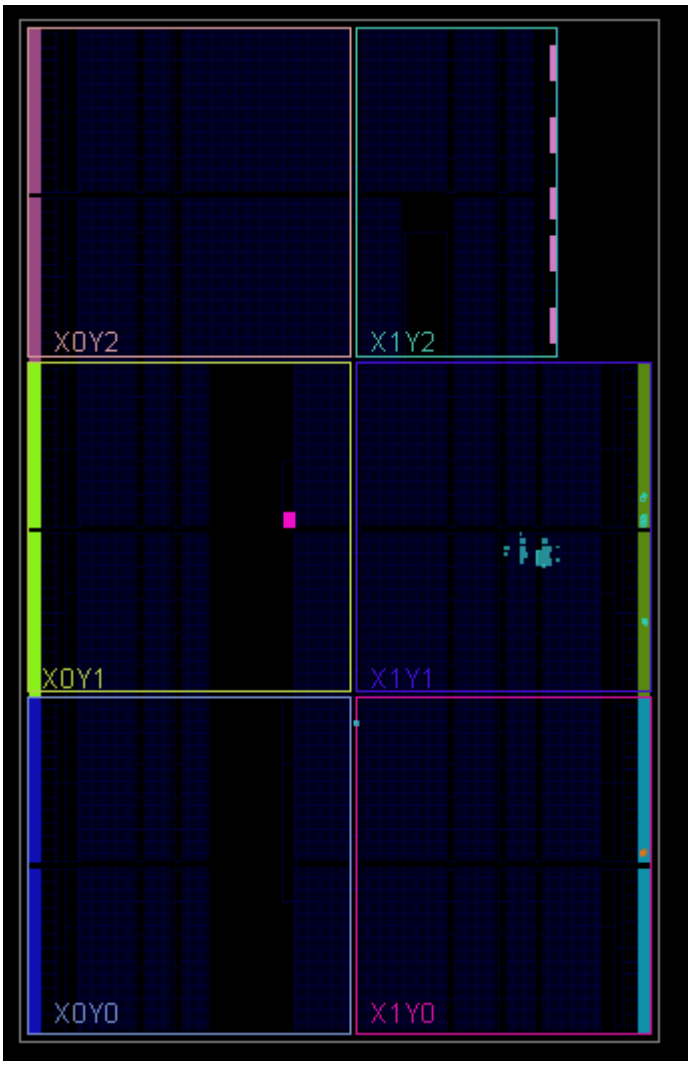
State	New Encoding	Previous Encoding
IDLE	00001	000
CHK_CMD	00010	001
WRITE	00100	010
READ_ADD	01000	011
READ_DATA	10000	100

- Timing Summary:

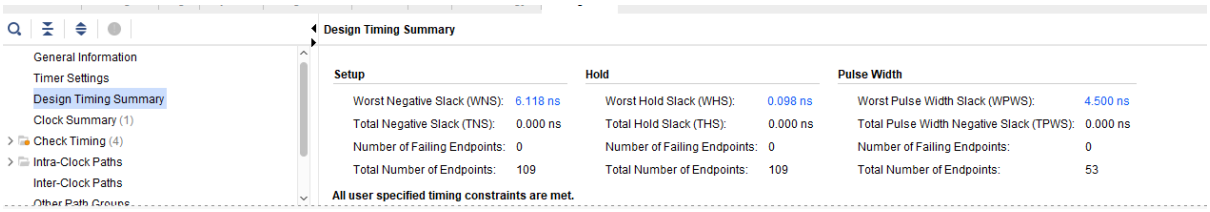
- Critical Path:



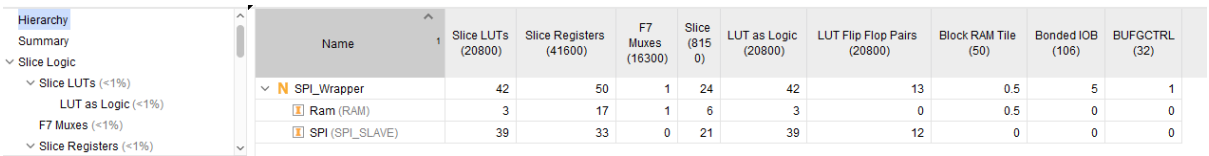
- # Implementation:
  - ## Seq:
    - ### Schematic:



    - ### Timing:

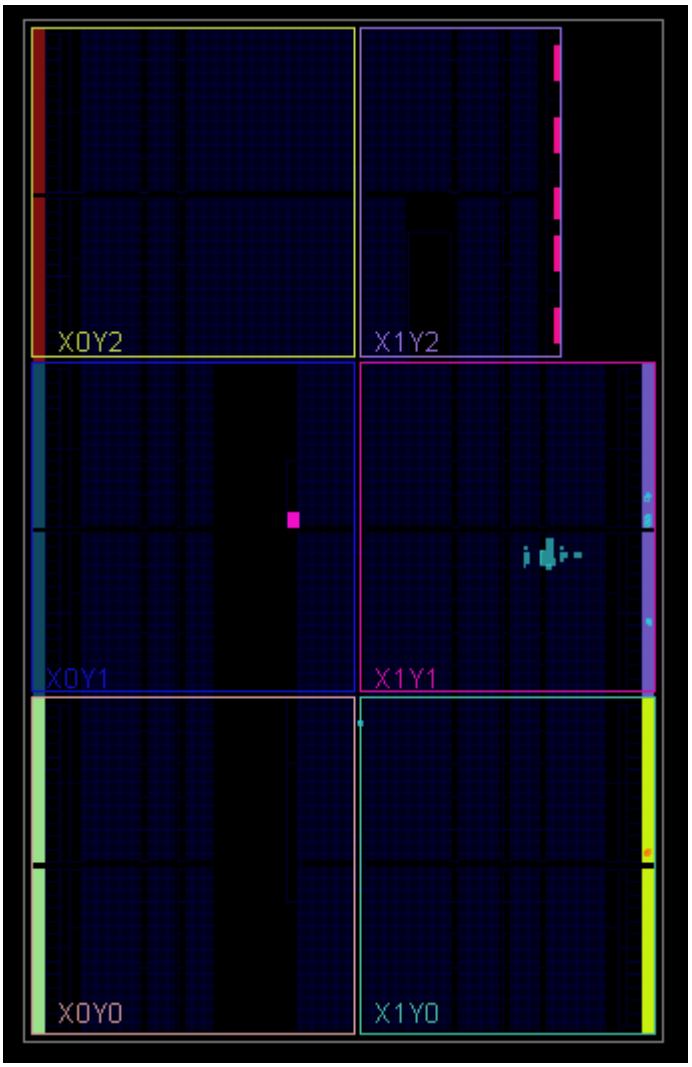


    - ### Utilization:



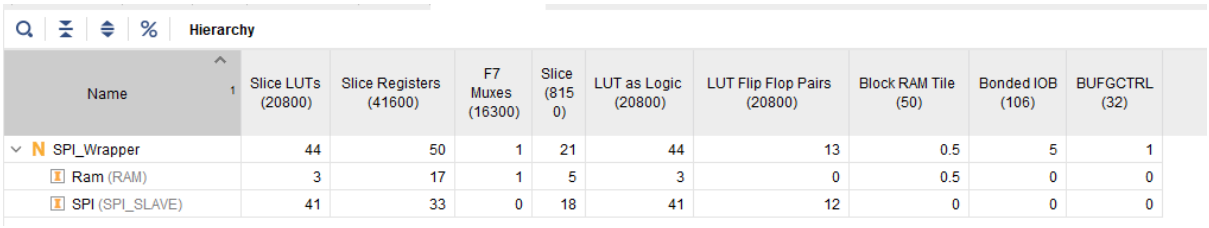
Name	Slice LUTs (20800)	Slice Registers (41600)	F7 Muxes (16300)	Slice (8150)	LUT as Logic (20800)	LUT Flip Flop Pairs (20800)	Block RAM Tile (50)	Bonded IOB (106)	BUFGCTRL (32)
SPI_Wrapper	42	50	1	24	42	13	0.5	5	1
Ram (RAM)	3	17	1	6	3	0	0.5	0	0
SPI (SPI_SLAVE)	39	33	0	21	39	12	0	0	0

- **Gray:**
  - Schematic:



  - Timing:

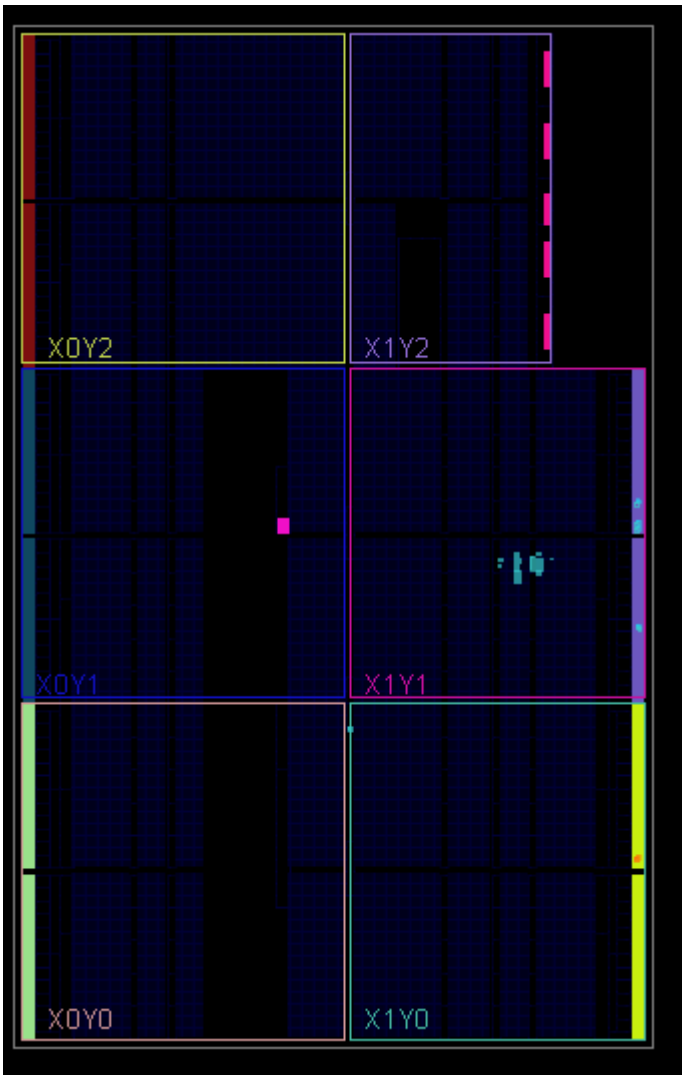
  - Utilization:



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)
N SPI_Wrapper	44	50	1	21	44	13	0.5	5	1
Ram (RAM)	3	17	1	5	3	0	0.5	0	0
SPI (SPI_SLAVE)	41	33	0	18	41	12	0	0	0

One_Hot:

Schematic:



Timing:

Utilization:

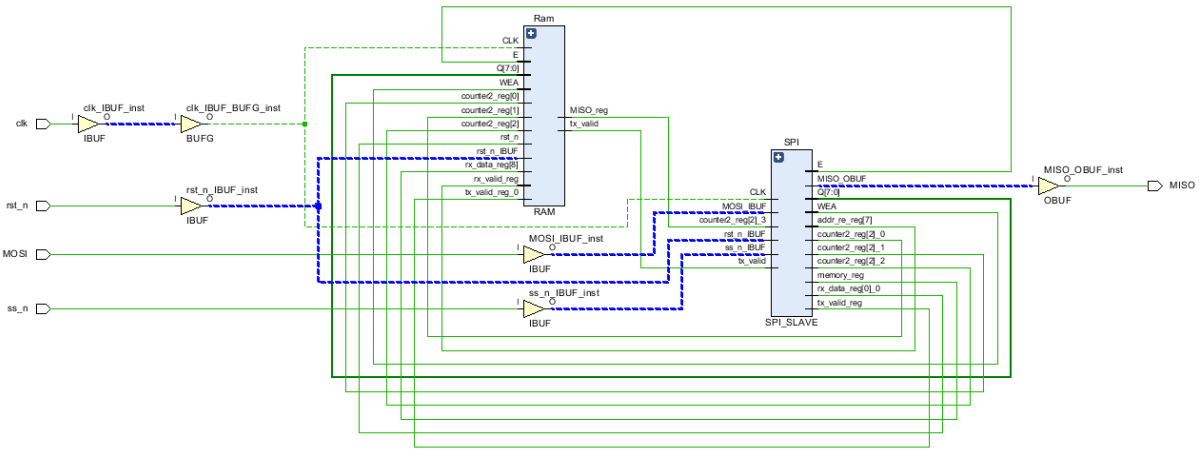
Hierarchy

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)
N SPI_Wrapper	42	52	1	21	42	15	0.5	5	1
Ram (RAM)	3	17	1	5	3	0	0.5	0	0
SPI (SPI_SLAVE)	39	35	0	19	39	14	0	0	0

From the results we found that the best encoding is seq.

- ## Messages:

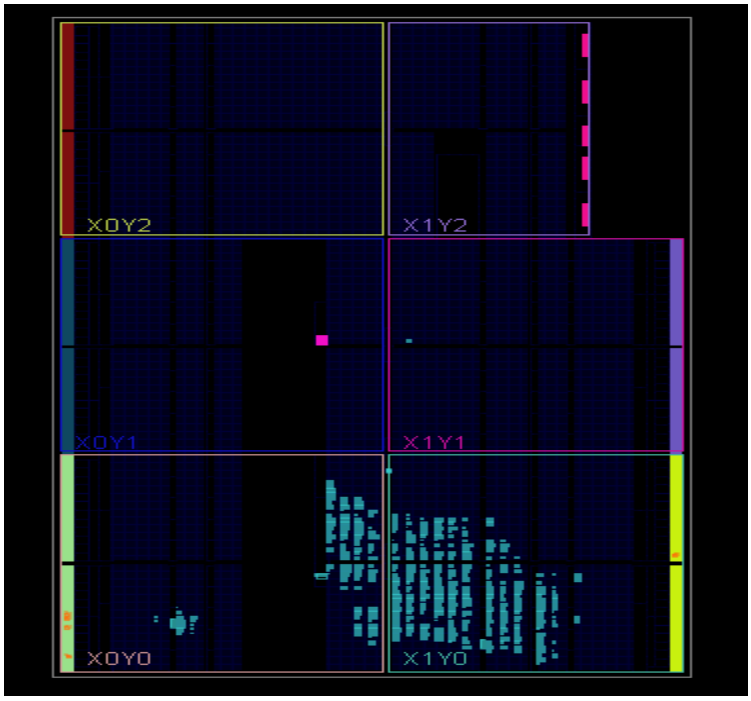
- ## Debugged Design:



- ## Timing after debug:

- ## Messages after debug:

- **Implementation of Device after Debug:**



- **Bitstream Successfull:**