

# Verification of a SPI Protocol

(Serial Peripheral Interface)

# Description:

The spi\_serializer module is responsible for serializing data from a FIFO buffer into a SPI-compatible serial data. The module operates as follows:

## Data Serialization:

- During the SHIFT state, the module shifts out the data from the shift register bit by bit, synchronized with the SPI clock (sclk). The most significant bit (MSB) of the data is typically shifted out first. The mosi output provides the serialized data stream compatible with the SPI protocol.

## Clock Generation:

- clk\_div is a counter that divides by 2 the main clock (clk) to generate the slower SPI clock (sclk).
- sclk is toggled at a rate determined by ClockDivider.
- sclk\_enable signal will be used to control when the sclk should be toggled. It should be active during the SHIFT state.

## Done Signal:

- The done signal indicates the completion of the serialization process. It is set high when all data bits have been shifted out and the serialization process is complete, so we can move to IDLE state.

## Reset Logic:

- The module includes logic to handle resets (rst) to ensure proper initialization of internal states and signals.

## MOSI Signal:

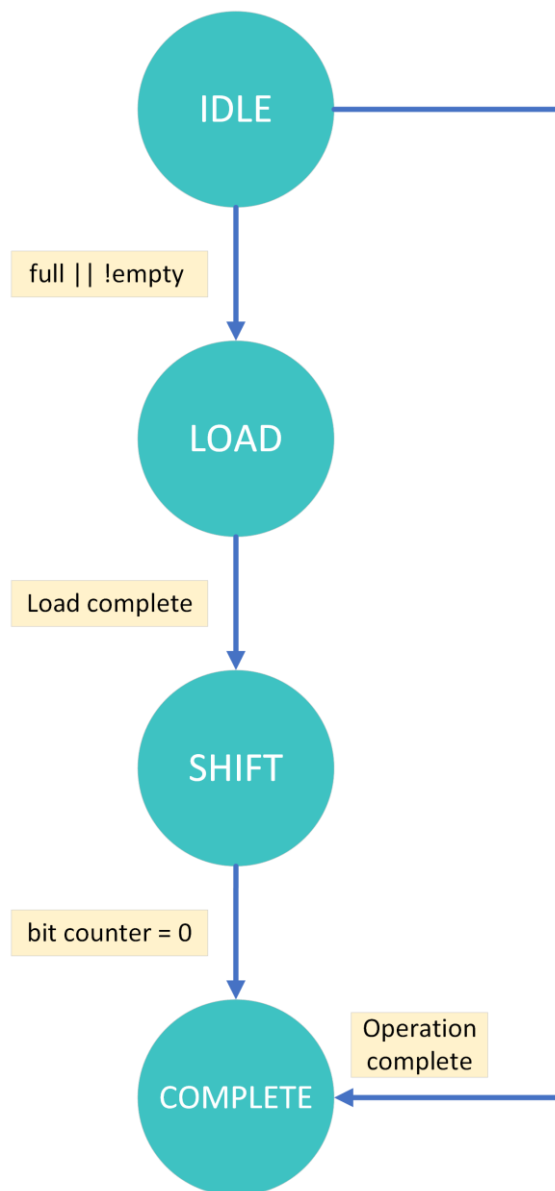
- Data Serialization: The mosi signal transmits the serialized bits of readData from the FIFO as an output.
- Bit-by-Bit Transmission: Data is sent out one bit at a time, synchronized with the SPI clock signal on the falling edge of sclk so that the data is stable (and can be sampled by a possible slave module on the rising edge of sclk).

## State Machine:

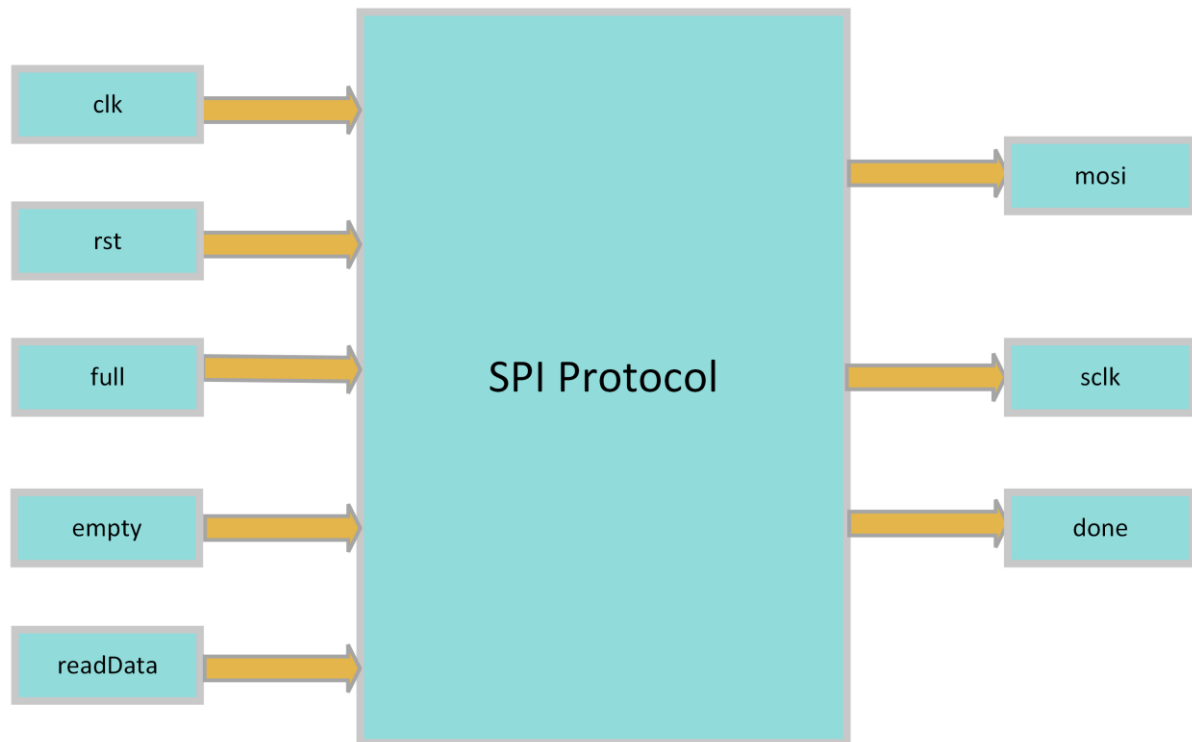
- A finite state machine (FSM) controls the serialization process. The FSM progresses through different states to manage the serialization process. FSM states are defined as follows:
- IDLE: Initial state where the module waits for the condition to start serialization, in this case, when the FIFO is full or not empty.

- **LOAD:** State where the module loads data from the FIFO.
- **SHIFT:** State where the module shifts out bits serially.
- **COMPLETE:** State indicating the completion of the serialization process, the done signal is activated in this state.

State Machine Diagram.



### Block Diagram:



### Input Definitions

- clk : Clock signal to synchronize the operations.
- rst : Reset signal to initialize the module.
- full: Signal indicating that the FIFO is full, triggering the start of serialization.
- empty: Signal indicating that the FIFO is empty, preventing serialization when active.
- readData ([DataWidth-1:0]): Data to be serialized.

### Output Definitions.

- sclk : SPI clock signal.
- mosi : SPI Master Out Slave In data line (serial data output).
- done: Signal indicating that the serialization process is complete.

### Internal Components:

- State Variables: state and next\_state used to track the state of the finite state machine.
- Shift Register: shift\_reg used to hold the data being serialized before being sent out.
- Bit Counter: bit\_counter used to keep track of the number of bits shifted out during serialization.
- SCLK Generator: Generates the SPI clock by toggling sclk.

# Formal Verification:

## Properties

N#	Name	Category	Status

**Verify the correct write operation.**

Properties defined to cover this spec:

Assumptions:

Assertions:

Covers:

**Verify the correct read operation:**

Properties defined to cover this spec:

Assumptions:

Assertions:

Covers

**Verify the correct full signal:**

Assertions:

Covers:

**Verify the correct empty signal:**

Assertions:

Covers:

**Verify the correct simultaneous write and read operation:**

Properties defined to cover this spec:

Covers:

Results:

Bug fixing:

Solution: