Project Report:
SPI Master Core Design

Rohan Joshi
(21BML0131)

Vellore Institute of Technology,
Vellore

Abstract

This report presents the design and implementation of an SPI (Serial Peripheral Interface) Master Core for embedded systems. Synchronous serial interfaces are widely used to provide economical board-level interfaces between different devices such as microcontroller, DAcs, ADCs and FPGA-based applications, enabling seamless communication with peripheral devices. The report provides insights into the SPI protocol, the wishbone master, SPI slave and functionality of the clock generator sub-block and the shift register sub-block of the SPI Master core.

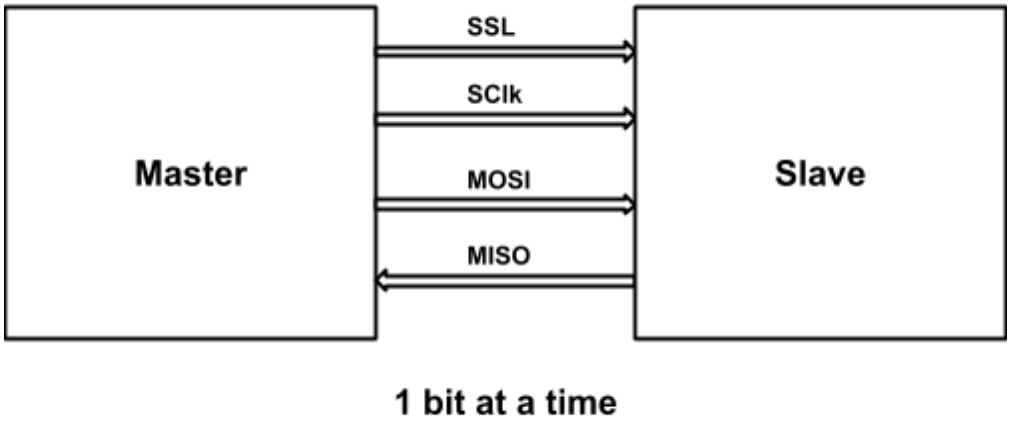
SPI Protocol

Serial Peripheral Interface is a synchronous protocol that allows a master device to initiate communication with a slave device. It is an interface used between microcontroller and peripheral ICs such as sensors, ADCs, DACs, shift registers, SRAM and others.

SPI Transfer Formats

SPI uses four single bit wires to establish communication:

1) Serial Clock (SClk)
2) Master Output Slave Input (MOSI)
3) Master Input Slave Output (MISO)
4) Device Select Pin (SSL)

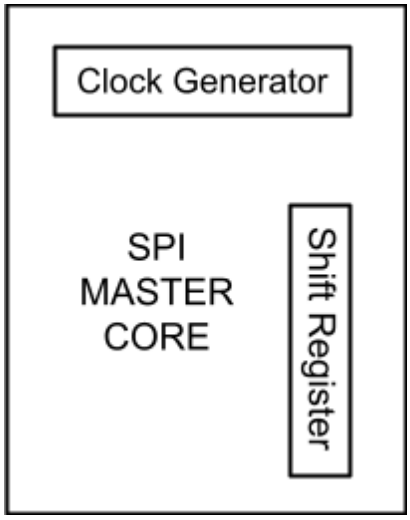


SPI Master Core Features

- Full duplex synchronous serial data.
- Variable length of transfer word up to 128 bits.
- MSB or LSB first data transfer.

- Rx and Tx on both rising or falling edges of the serial clock independently.
- Up to 32 slave select lines (but only one can communicate).
- Auto slave select.
- Interrupt enable.
- Clock divider.

Design Blocks of SPI Master Core



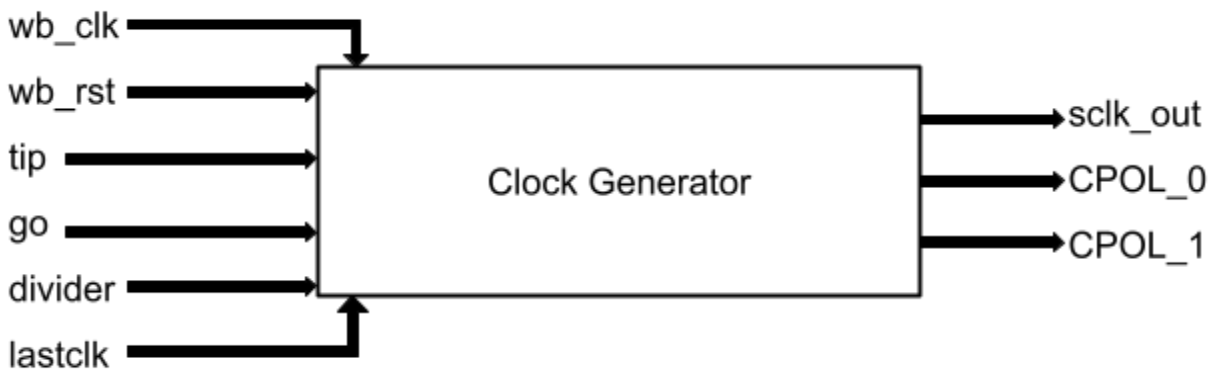
Clock Generator

The clock generator sub-block is a critical component of the SPI Master Core, responsible for generating the clock signal used for synchronization between the master and slave devices. It allows for configuration of clock frequency, clock polarity, and clock phase to support different SPI modes.

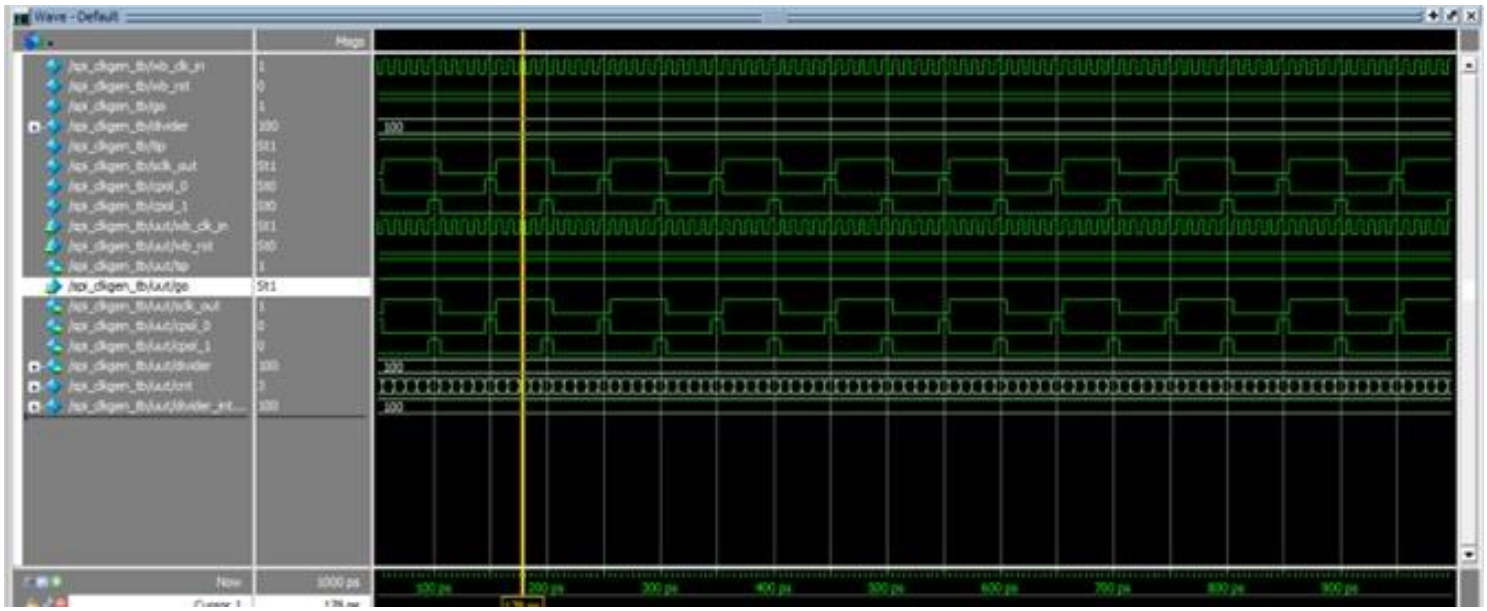
Primary Functions

- Clock Frequency Generation: Generates a clock signal at a specific frequency, which determines which data is transmitted. This frequency is usually configurable to suit the needs of SPI slave devices.
- Clock Polarity and Phase Control: SPI supports multiple clock modes (0, 1, 2, and 3), which define the polarity (CPOL) and phase control (CPHA) of the clock signal. The clock generator section can be configured to produce a clock signal in the required mode for compatibility with SPI devices.
- Clock Distribution: The clock signal generated by the clock generator block is distributed to all connected slave devices on the SPI bus, ensuring that they all operate in sync. This synchronization is essential for reliable data transfer.
- Clock Enable/Disable: Some SPI cores include the ability to enable or disable the clock signal. This feature can be useful for controlling power or terminating communication with slave devices when necessary.

Block Diagram



Output Waveform: (Testbench of clock generator)



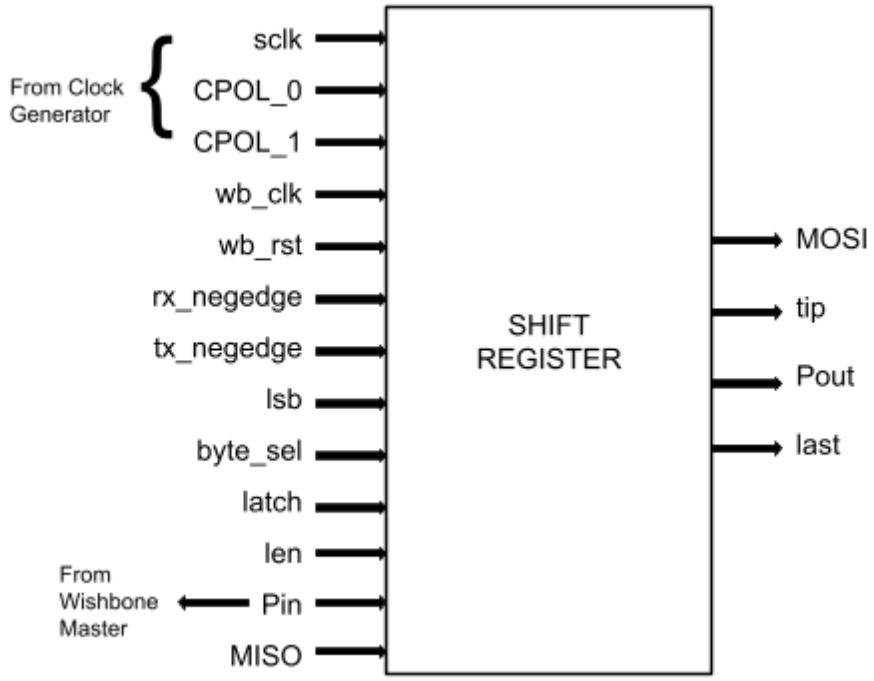
Shift Register

The shift register sub-block plays a pivotal role in serializing outgoing data and deserializing incoming data during SPI communication.

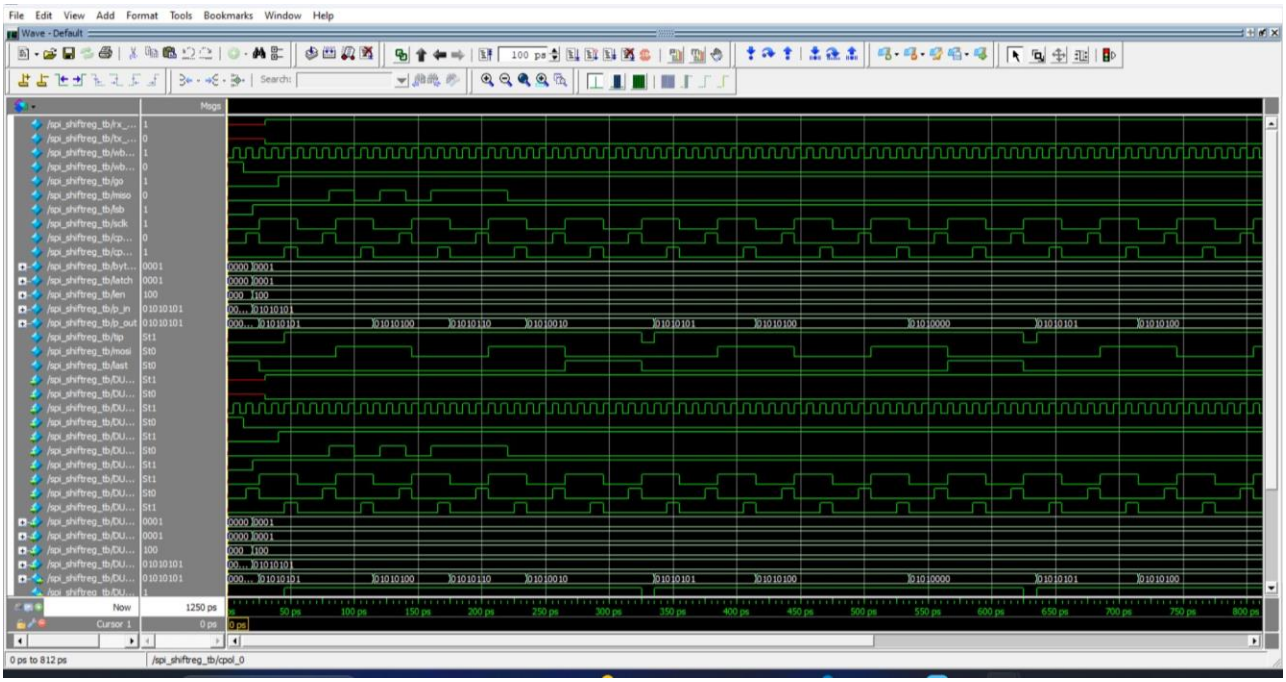
Primary Functions

- Data Serialization: When the SPI master transmits data to a slave device, the shift register block takes parallel data and serializes it into a bitstream. It shifts out these bits one at a time, synchronized with the clock signal.
- Data Deserialization: When receiving data from a slave device, the shift register block deserializes the incoming data, converting it from a serial bitstream back into parallel data. The data is then made available on the data bus for further processing.
- Buffering: The shift register block may include data buffers to temporarily store data as it is shifted in or out. This buffering helps ensure that data is transferred reliably and at the correct timing.

Block Diagram



Output Waveform: (Testbench of shift register)



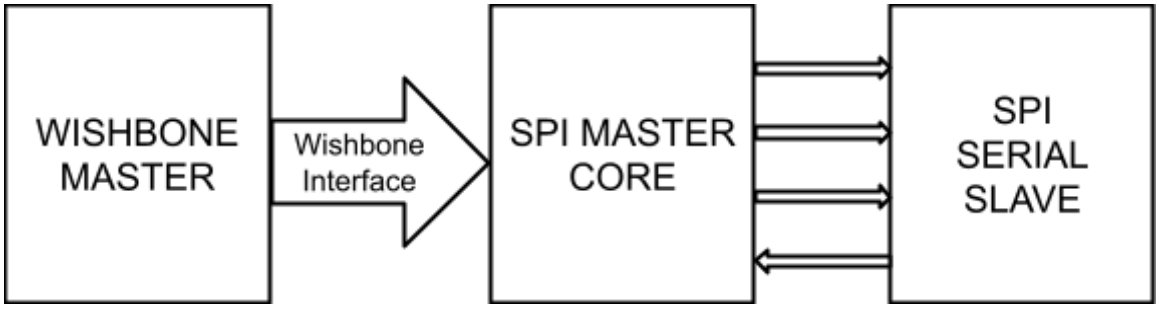
Top Module of the SPI Communication Core

This module refers to the highest-level module that orchestrates the entire SPI Communication Core. It serves as the central control unit, connecting all the blocks within a large system, such as an FPGA or microcontroller. It consists of the wishbone master, the SPI master core and the SPI slave.

Primary Functions

- External Interface: It interfaces with the external world, connecting to pins or connectors for data, clock, and control signals. These connections facilitate communication between the SPI core and the devices it communicates with, both as a master and a slave.
- Slave Device Selection: The top module often includes logic for selecting the specific slave device with which the SPI master will communicate. This selection can involve enabling or disabling the Slave Select (SS) lines corresponding to various slave devices.
- Data Handling: It manages the transfer of data to and from the SPI core. The data may be sourced from or delivered to other components in the system through the data bus.
- Control and Configuration: The top module provides control over the operation of the SPI core. It may include registers for configuring settings like clock frequency, clock polarity, clock phase, and other mode-specific parameters.
- Clock Generation: In some designs, the top module is responsible for initiating the clock signal generation by interacting with the clock generator block within the SPI core.
- Status and Error Handling: It may incorporate logic for monitoring the status of the SPI communication, detecting errors, and taking corrective actions when necessary.

SPI Communication Architecture



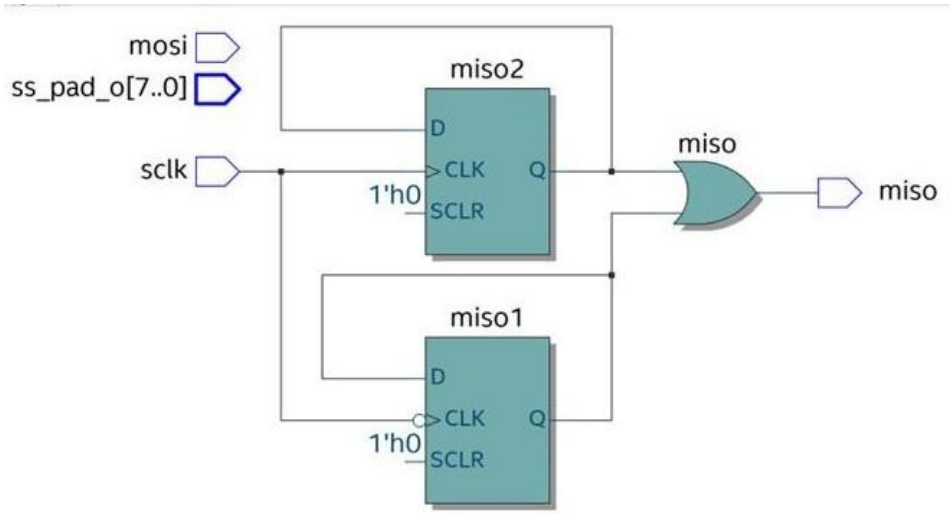
Wishbone Master : A bus that drives the inputs.

Wishbone Interface: Allows the wishbone master to read from and write to registers within the SPI core.

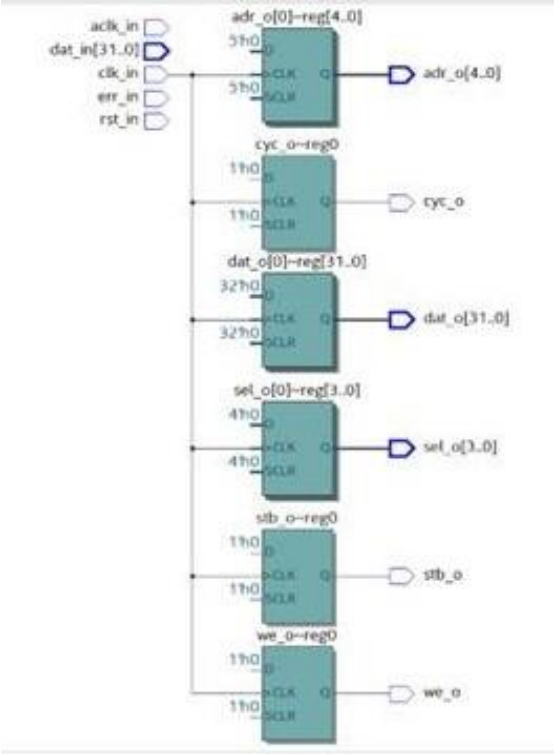
SPI Master Core: It is responsible for controlling and coordinating communication with the connected slave devices.

SPI Slave: Plays a passive role in the communication, responding to commands and data requests from the SPI master.

SPI Slave Block Diagram

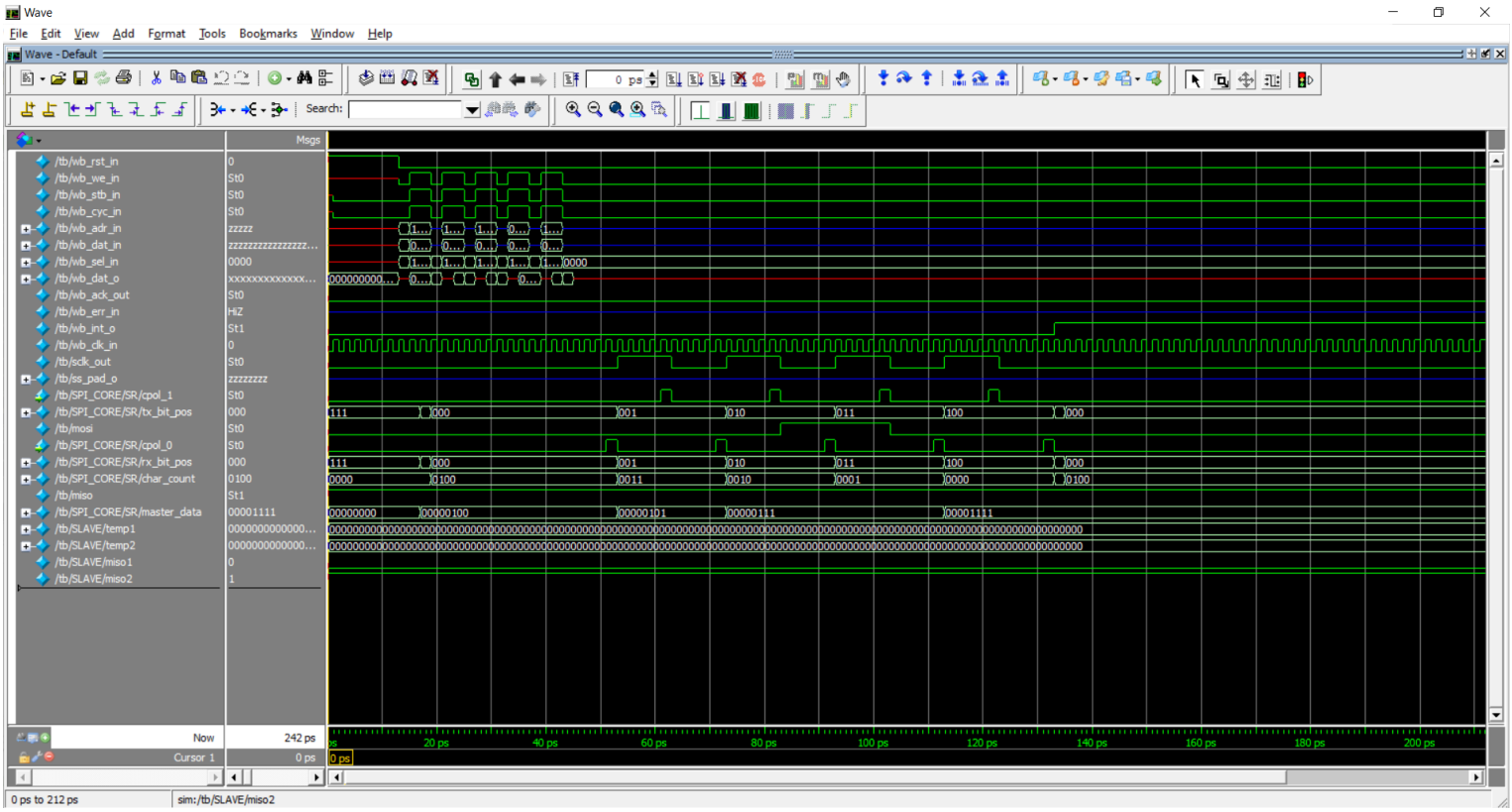


Wishbone Master Block Diagram

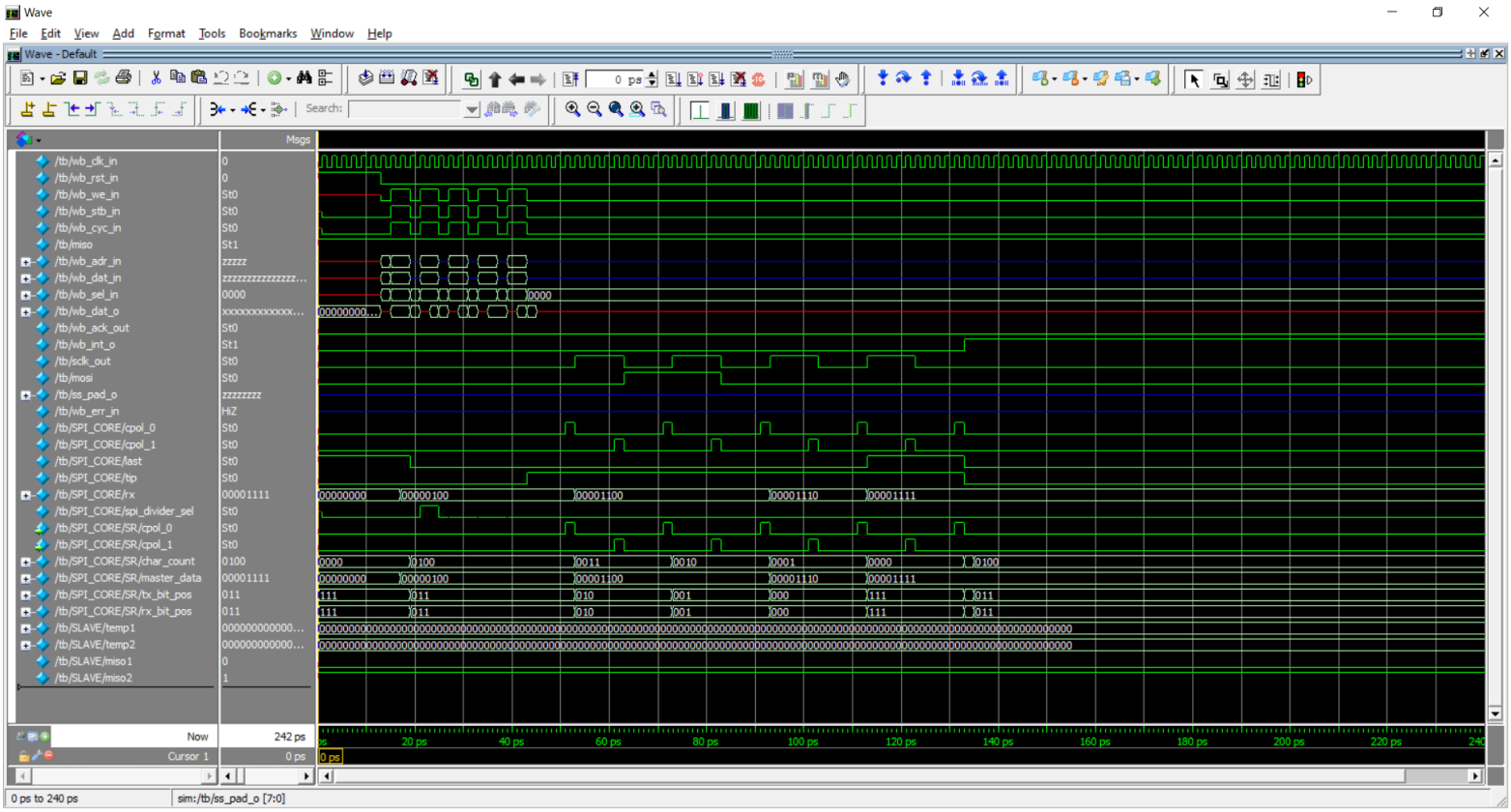


Output Waveform: (Top Level Testbench)

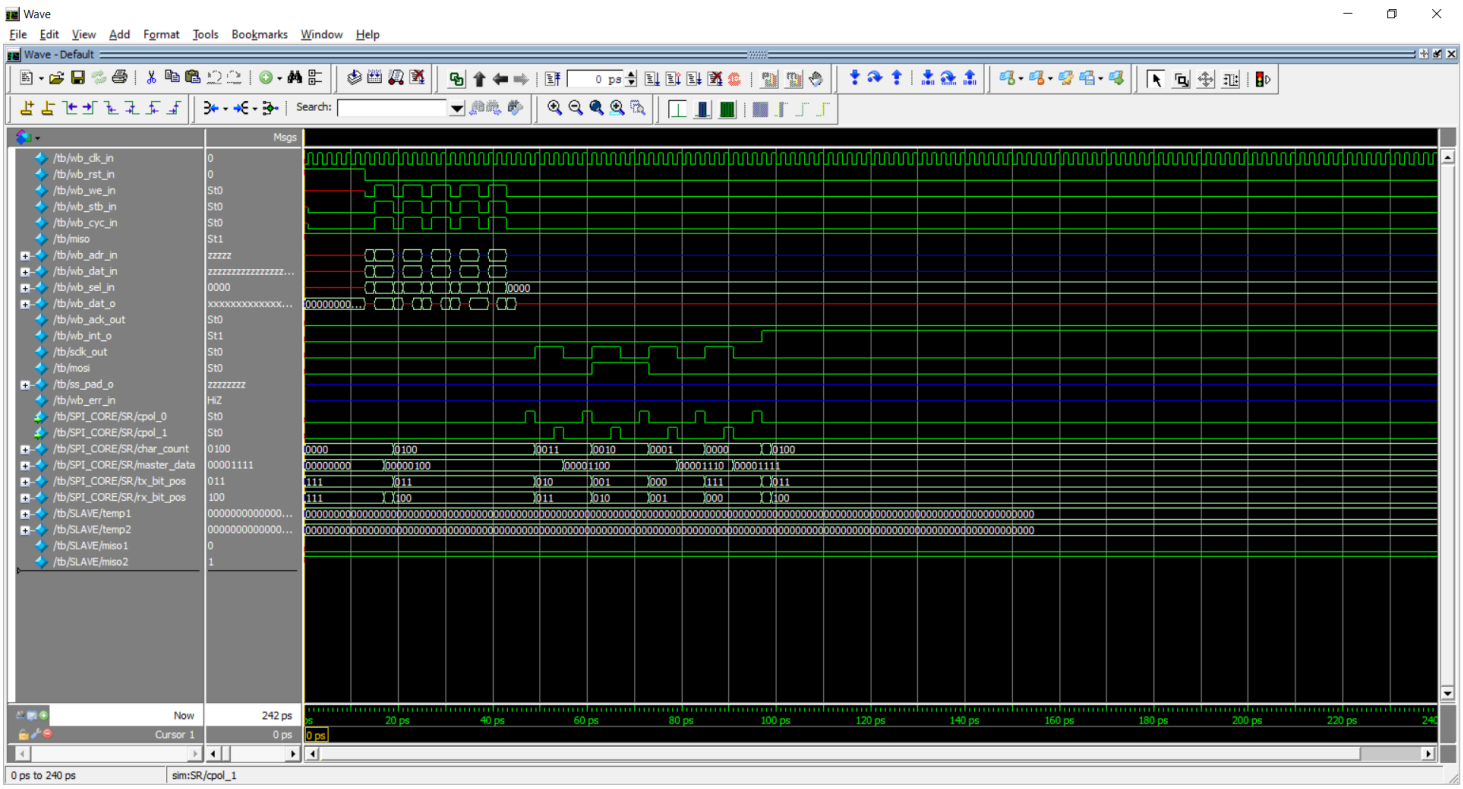
Case 1: tx_neg =1,rx_neg=0, LSB=1, char_len=4



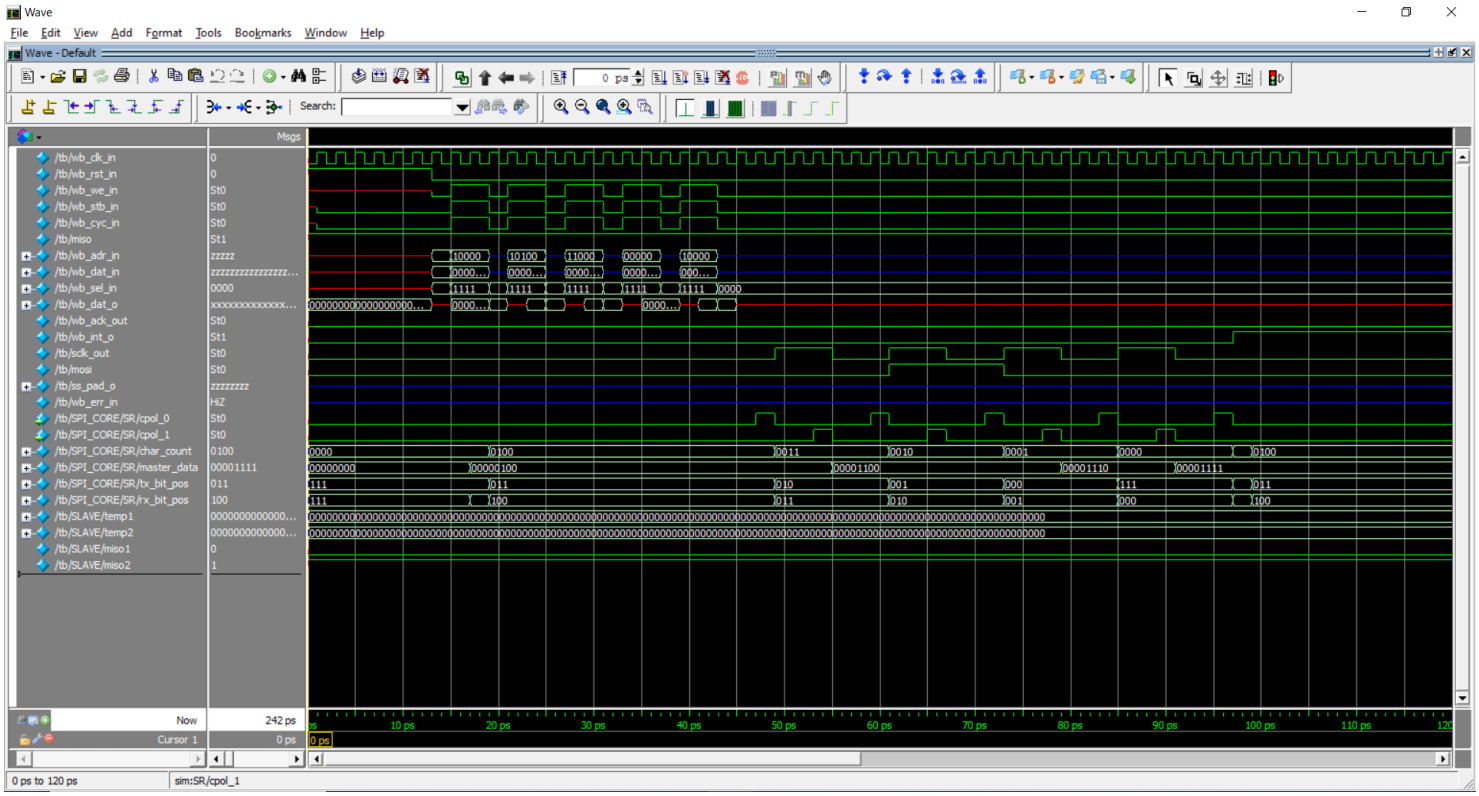
Case 2: tx_neg =1,rx_neg=0, LSB=0, char_len=4



Case 3: tx_neg =0,rx_neg=1, LSB=1, char_len=4



Case 4: tx_neg =0,rx_neg=1, LSB=0, char_len=4



Hence, the top module is implemented and the required waveforms have been received for all cases.