SPI Controller Verification IP

Functional Verification Using UVM RAL Methodology

(For APB/Wb-SPI Master Core)

Overview

The SPI Master Controller is designed to perform full-duplex synchronous serial communication between a master and multiple slave devices, following the SPI protocol. This Verification IP (VIP) enables comprehensive, configurable, and reusable verification of the APB/Wb-SPI Master core using a robust UVM-based environment and register abstraction via RAL methodology.

Key Features

- Full-duplex SPI communication
- Supports **up to 128-bit** word length transfers
- **MSB/LSB first** configurable transmission
- **Rx and Tx** edges configurable (rising/falling)
- 8 programmable **Slave Select (SS)** lines
- Register-programmable via **APB interface**
- **Technology-independent Verilog** design
- Fully synthesizable
- Verified using **UVM Testbench** with built-in **SPI slave VI**

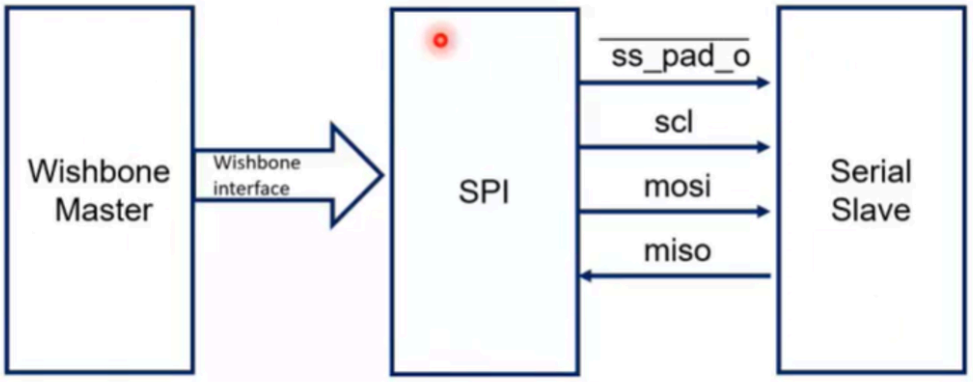
IO Interfaces

APB/WB Interface Signals:

Signal	Width	Direction	Description

PCLK	1	Input	Clock
PRESET	1	Input	Active-high reset
PADDR	32	Input	Address input
PWDATA	32	Input	Data input
PRDATA	32	Output	Data output
PENABLE	1	Input	Transfer enable
PSEL	1	Input	Peripheral select
PWRITE	1	Input	Write control
PREADY	1	Output	Transfer complete status

SPI Master Core Architecture



SPI External Connections

Signal	Width	Direction	Description
/ss_pad_o	8	Output	Active-low slave select lines
sclk_pad_o	1	Output	Serial clock
mosi_pad_o	1	Output	Master-out, slave-in data
miso_pad_i	1	Input	Master-in, slave-out data

Core Register Map

Register Name	Address	Width	Access	Description
RX0–RX3	0x00–0x0C	32	Read	Data Receive Registers
TX0–TX3	0x00–0x0C	32	R/W	Data Transmit Registers
CTRL	0x10	32	R/W	Control and Status Register
DIVIDER	0x14	32	R/W	Clock Divider
SS	0x18	32	R/W	Slave Select Register

Register Highlights

CTRL Register (0x10)

- `CHAR_LEN[6:0]`: Data bit length (1–128 bits)
- `GO_BSY`: Starts the SPI transfer
- `IE`: Enables interrupt post-transfer
- `ASS`: Auto Slave Select enable
- `LSB`: Configures LSB or MSB first
- `Tx_NEG`/`Rx_NEG`: Configures clock edge for transmit and receive

DIVIDER Register (0x14)

- Controls SPI clock generation
- `f_sclk = f_wbclk / (DIVIDER + 1)`

UVM-Based Verification Architecture

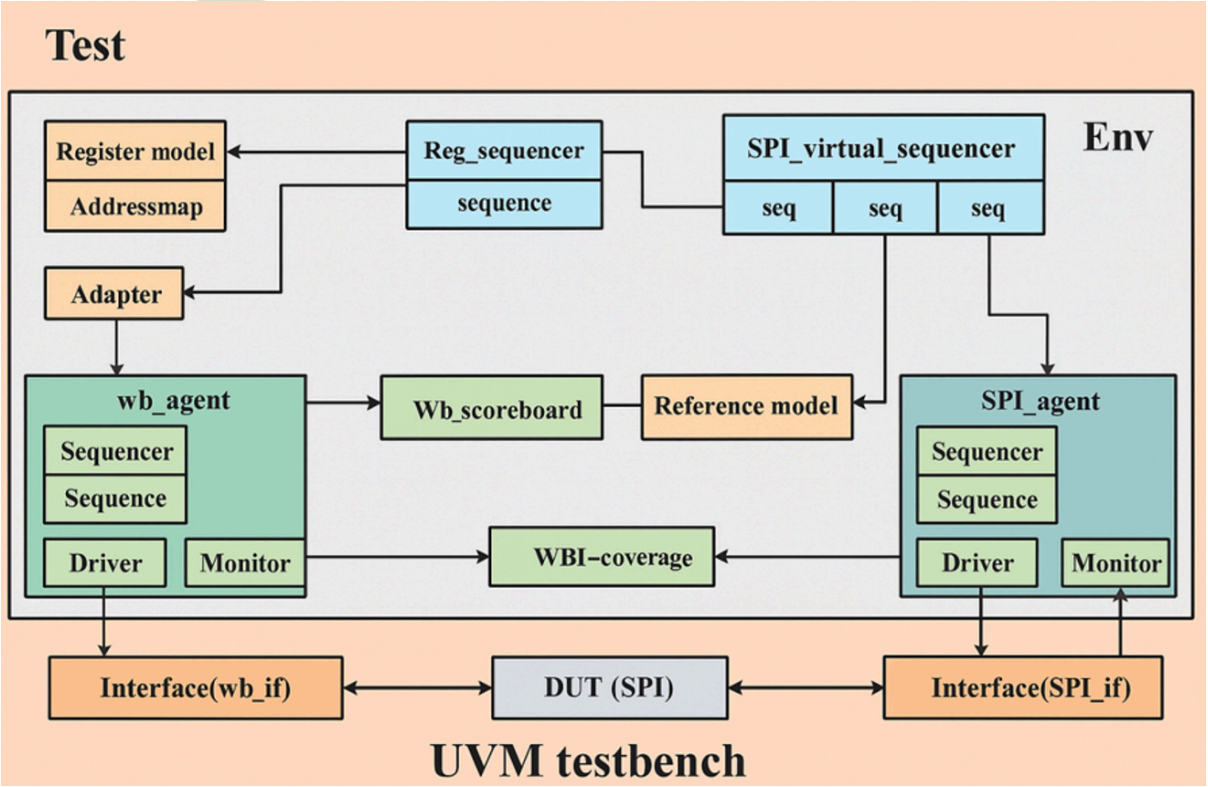
Testbench Hierarchy

- **SPI_TEST**: Top-level test extension from `uvm_test`

- **SPI_ENVIRONMENT**: Encapsulates agent and scoreboard
- **SPI_AGENT**: Configured as active, includes driver, monitor, sequencer
- **SPI_SEQUENCE/ITEM**: Parameterized transaction flow with constraints
- **SPI_DRIVER**: Drives transaction to DUT via APB

- **SPI_MONITOR**: Captures and forwards observed responses
- **SPI_SCOREBOARD**: Compares expected vs. actual data
- **COVERAGE**: Functional and code coverage monitored via Cadence IMC

"UVM Environment Architecture for SPI Verification"



Test Methodology

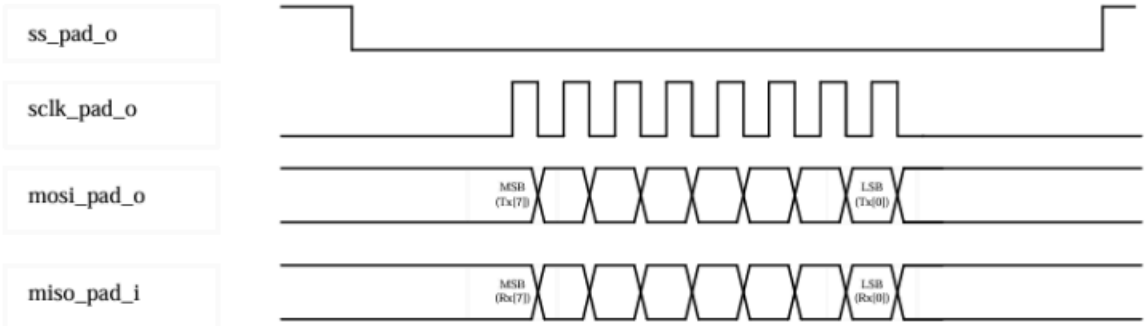
- **Reset Phase**: Initializes DUT and testbench state
- **Main Phase**: Triggers SPI sequences using UVM sequencer

- **Synchronization**: Handled via APB clocking block in interface
- **Coverage Goals**: Achieve 100% functional + code coverage

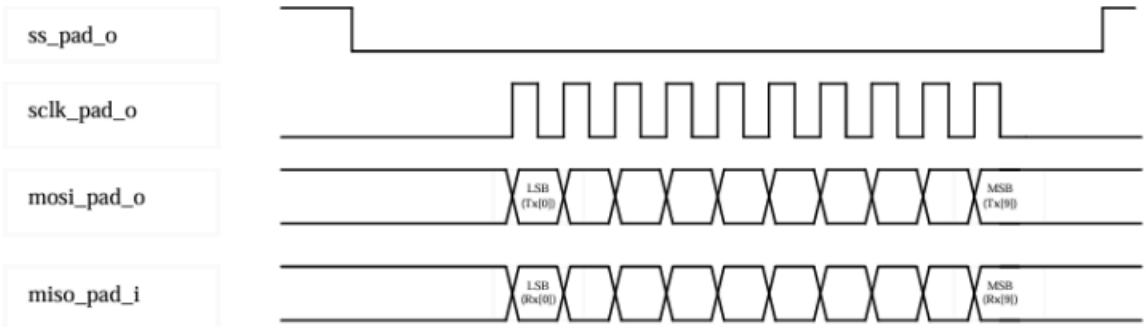
---

Timing Diagrams

- Write cycle with `PREADY` wait + no-wait conditions
- SPI transfer with varying Clock Polarity and Clock Phase combinations



CTRL[LSB] = 0, CTRL[CHAR_LEN] = 0x08, CTRL[TX_NEG] = 1, CTRL[RX_NEG] = 0



CTRL[LSB] = 1, CTRL[CHAR_LEN] = 0x0a, CTRL[TX_NEG] = 0, CTRL[RX_NEG] = 1

Test Case Scenarios

- **APB/Wb-Related**:
    - Read/write transactions with/without `PREADY` delays
    - Address/data/data-phase validation

- **SPI-Related**:
    - Transfer integrity with varying data lengths
    - Phase/polarity effects on data propagation

Conclusion

This Verification IP for the SPI Controller offers an exhaustive and configurable environment to ensure SPI Master core reliability and standard compliance. Its UVM architecture supports seamless integration with APB/Wb-based SoC designs and guarantees scalability across designs using SPI-based interfaces.