The Serial Peripheral Interface Bus –VLSI Lab:   
Top Architecture

Contents

[Table of Changes 2](#_Toc299889206)

[Scope 2](#_Toc299889207)

[Top Architecture 2](#_Toc299889208)

[Master Host 3](#_Toc299889209)

[Slave Host 4](#_Toc299889210)

[SPI Interface 5](#_Toc299889211)

[Message Structure 6](#_Toc299889212)

List of Tables

[Table 1 - Table of Changes 2](#_Toc299889229)

List of Figures

[Figure 1 - Top Architecture 2](#_Toc299889289)

[Figure 2 - Master Host 3](#_Toc299889290)

[Figure 3 - Slave Host 4](#_Toc299889291)

[Figure 4 - SPI Interface 5](#_Toc299889292)

[Figure 5 - Message Structure 6](#_Toc299889293)

# Table of Changes

|  |  |  |
| --- | --- | --- |
| Version | Date | Reason of change |
| 1.0 | 31.07.2011 | Creation |

Table - Table of Changes

# Scope

This document describes the Top Architecture of the SPI Project, which is created in the VLSI Lab.

# Top Architecture

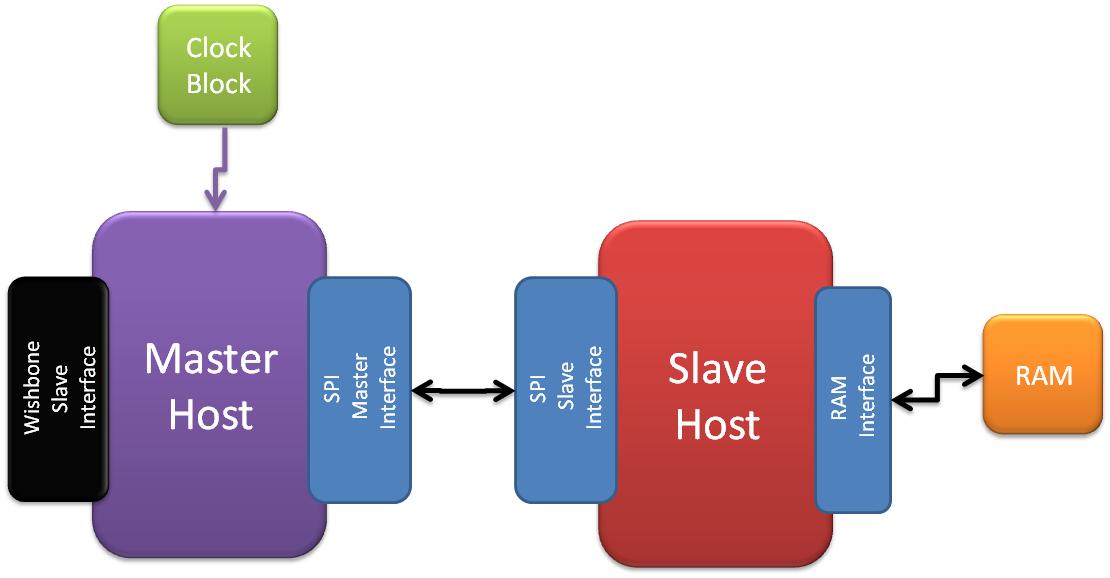


Figure - Top Architecture

1. **Clock Block** – SPI have 4 working methods, based on the CPOL and CPHA values. Instead of re-write every process twice (one for falling edge and one for rising edge), the Clock Block will connect the required clock to the Master Host.
2. **Master Host** – Connects between the outside world to the SPI world, through the Wishbone interface.
3. **Slave Host** – Connects between the SPI world to the outside world – here it is a RAM, which data can be read from and written to.

# Master Host

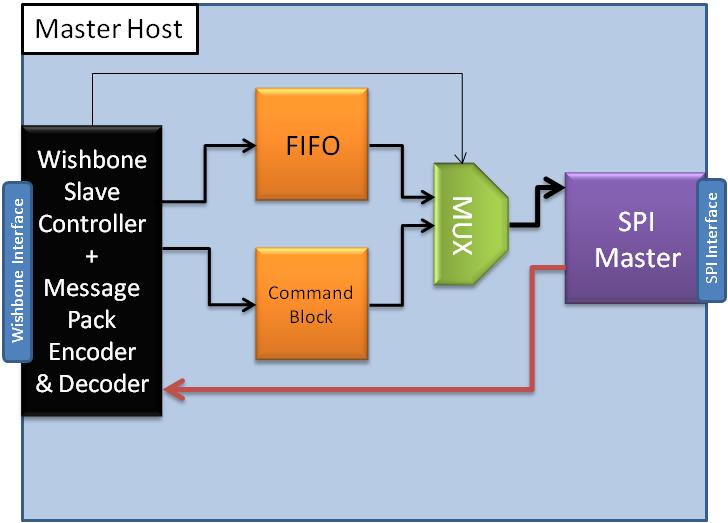


Figure - Master Host

The Master Host receives a Wishbone command from the WBS (Wishbone Slave) Interface. When a new command ("Write" / "Read") is received, the *Command Block* transmits the command, through the MUX to the SPI Master, which transmit this command to the SPI Host as a SPI message.

When the host returns a "ready" status, the data is being sent through the Message Pack Encoder (it will be explained later), to the FIFO, through the MUX, into the SPI Master.

Received data from the SPI Slave is encoded in the Message Pack Decoder, In case of "Read" mode (WBS\_WE\_I = '0', which means that the wishbone interface has received a read command), the data will be decoded, and transmitted through the Wishbone Interface.

# Slave Host

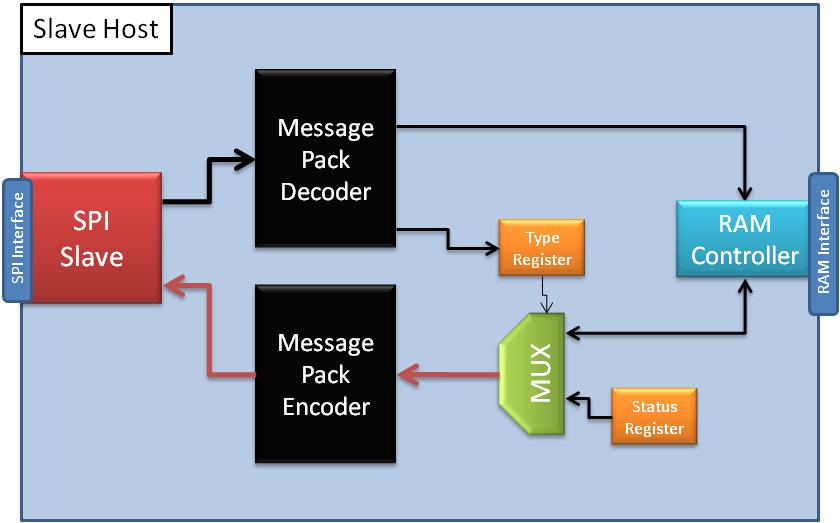


Figure - Slave Host

Data is received from the SPI Master, decoded in the Message Pack Decoder, and in case of write command - is saved in the RAM, through the RAM Controller.

The current command is saved in the Type Register.

When command "read" is received, data from the RAM is encoded in the Message Pack Encoder, and transmitted through the SPI Slave.

Status register will be read upon a start of write / read transaction, or when required (Master Host request).

# SPI Interface

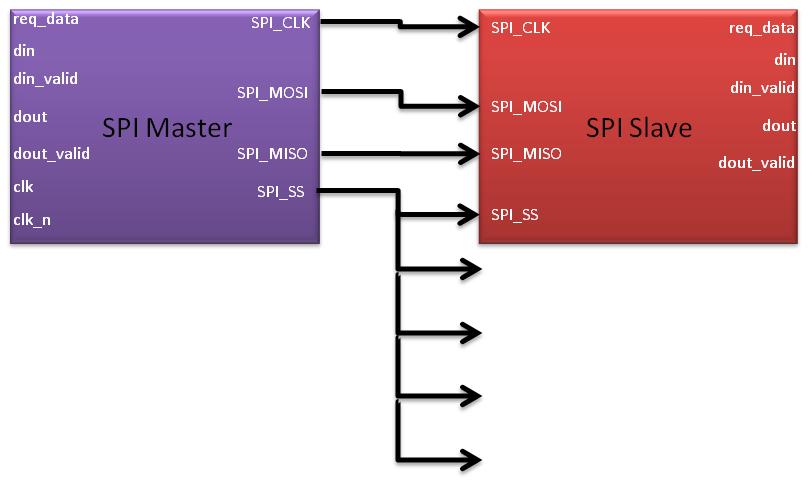


Figure - SPI Interface

SPI Master Interface contains interface to FIFO, and to clock (both the real clock and the inverse clock – Clock Negative).

SPI Slave Interface contains interface to RAM.

**SPI\_SS** is a bus, which its size is determined using a generic parameter.

# Message Structure

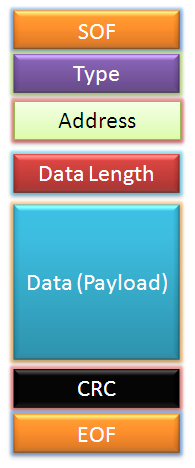


Figure - Message Structure

Message structure is consumed of the following blocks:

1. **SOF** – Start of Frame (Head)
2. **Type** – Message type (command / data…)
3. **Address** – To where the message will sent to (which address in RAM)
4. **Data length** – Data (Payload) length
5. **Data (Payload)** – the data itself
6. **CRC** – Here Checksum will be used, which will be calculated from the Type block to the Data block
7. **EOF** – End of frame (Tail)