AXI4-Lite to Wishbone Bridge

Introduction

AXI Lite to Wishbone bridge IP can connect a Wishbone slave-based IP with AXI Lite master.

Features

- **Independent read and write channels:** The IP has independent read and write channels and IP can drop the write channel in case of READONLY scenario and drop the read channel in case of WRITEONLY scenario. These READONLY and WRITEONLY scenarios are indicated to the IP through parameterization.
- **Channel Arbiter:** The channel arbiter feature generally enables any IP to alternate between two masters. But here, it is arbitrating the read and write bus of the Wishbone bus. Because this IP uses separate channels for reads and writes, that is why an arbiter is required to alternate between the two channels.
    - **Priority based Arbitration:** If arbiter SCHEME is set to PRIORITY, then the channel given to arbiter as channel 'a' would be given priority over channel 'b'.
    - **Alternating Arbitration:** If arbiter SCHEME is set to ALTERNATING, then alternate channel is connected to output when other channel ends transaction.
- **Configurable AXI-lite data bus width:** AXI4 lite data bus has configurable width through parameterization. There is support available for 32-bit and 64-bit operation.
- **Configurable Wishbone data bus width:** Currently, Wishbone data bus width is parameterized and 8 bit and 32-bit bus widths are supported.
- **Zero on Idle:** When neither read nor write transactions are active, the Wishbone interface outputs are pulled down to zero to decrease power consumption.
- **Transaction timeout:** When Wishbone slave IP fails to respond back to the bridge till 10 clock cycles, then transaction is timed out and AXI lite master is brought out of stalled condition. This number of cycles for which the bridge waits for the response is parameterized.

Architecture

The bridge transforms AXI4 lite signals into Wishbone signals. Following are the signals involved in AXI and WB interfaces

AXI4-lite signals

There are 5 channels in AXI4-lite

1. Write address
2. Write data
3. Write response
4. Read address
5. Read data

These channels are having following signals in them. The starting character of these signals is ‘i' or ‘o’ that indicates this signal to be input or output of the bridge (AXI slave) respectively.

Global	Write address	Write data	Write response	Read address	Read data
i_clk	i_axi_awvalid	i_axi_wvalid	o_axi_bvalid	i_axi_arvalid	o_axi_rvalid
i_axi_reset_n	o_axi_awready	o_axi_wready	i_axi_bready	o_axi_arready	i_axi_rready
-	i_axi_awprot	i_axi_wdata	o_axi_bresp	i_axi_araddr	o_axi_rdata
-	-	i_axi_wstrb	-	i_axi_arprot	o_axi_rresp

Wishbone signals

Following input and output signals exist for Wishbone interface implemented in the bridge. Starting characters ‘i’ or ‘o’ indicate that either the signal is input or output:

Input Signals	Output Signals	Output Signals
i_wb_stall	o_reset	o_wb_addr
i_wb_ack	o_wb_cyc	o_wb_data
i_wb_data	o_wb_stb	o_wb_sel
i_wb_err	o_wb_we	

Module description

In AXI interface, there are total 5 channels. 3 for write and 2 for read. The design considers Read address and Read data channels as one channel of ‘Read’. And considers three write related channels: write address, write data and write response, as one ‘write’ channel.

There are a total of 6 modules present in the bridge design.

1. **axlite2wbsp:** this is the top module of the bridge design. It connects the read and write channel modules with the channel arbiter and generates the final outputs for Wishbone master.
2. **axilwr2wbsp:** this module implements the logic of write channel and translates AXI write channel signals into Wishbone write signals.
3. **axilrd2wbsp:** this module implements the logic of read channel and translates AXI read channel signals into Wishbone read signals.
4. **wbarbiter:** this module is necessary because AXI has two different buses for read address and write address, whereas Wishbone has a single address bus for both read and write operations. The

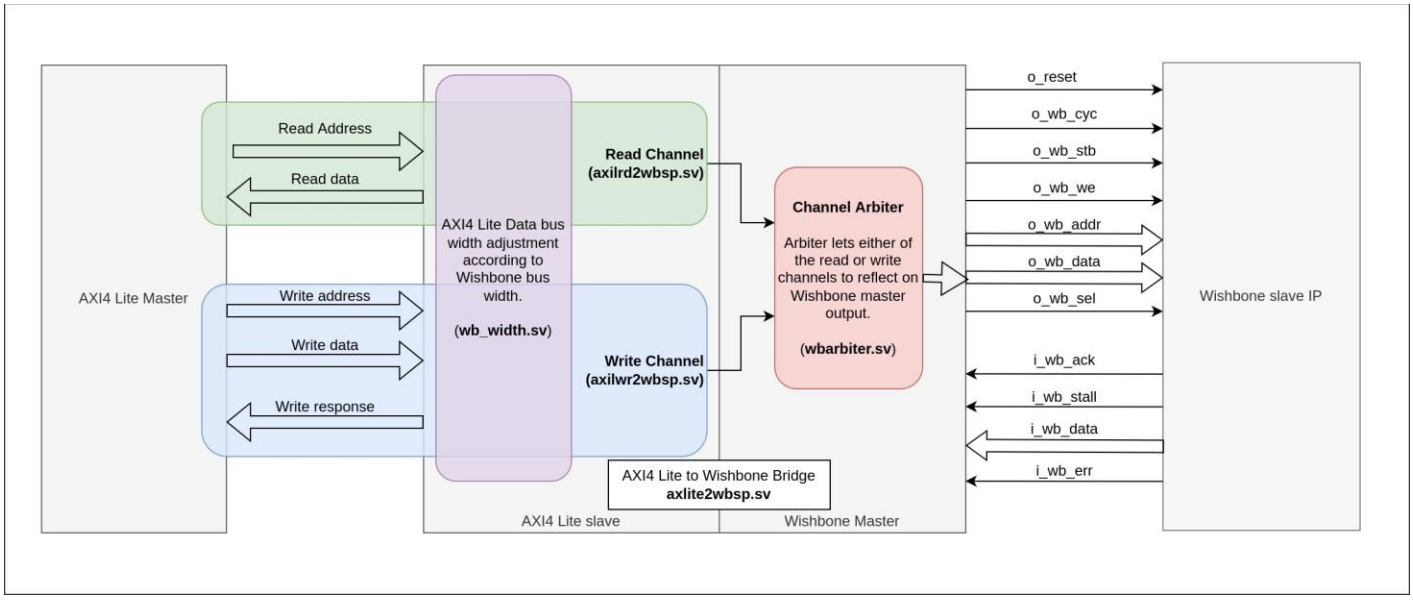
arbiter module lets only one of the read or write channels use the address and data buses of Wishbone. It uses multiple schemes like "Priority" and "Alternating".

5. **wb_width:** this module helps translate AXI data and address bus into Wishbone data and address bus where widths of AXI and Wishbone are different. This module converts byte address coming from AXI to word address that is required by Wishbone bus.
6. **num_ones:** this is a helper module of wb_width module.

Block Diagram

This section places blocks of modules, described in previous section, into a diagram to depict the functionality of these modules overall.

To avoid over complicated diagram, AXI4 lite channels are shown instead of individual signals.



Features Validation

Following is the validation status of features mentioned on top of the document:

Sr. #	Feature name	Validation status
1	Read operation	Passed
2	Write operation	Passed
3	AXI Lite width 32 and 64 bits	Passed
4	Wishbone 8-bit data width	Passed
5	Zero on idle	Passed
6	Transaction timeout	Passed