VERIFICATION OF AXI4-LITE

ECE 593 Fundamentals of Pre-Silicon Validation
VERIFICATION PLAN

Amrutha Anil (amrutha@pdx.edu)
Manjari Rajasekharan (manjari@pdx.edu)
Durganila Anandhan (anandhan@pdx.edu)

Table of Contents

1. Introduction

AXI-4 Lite Protocol, The Advanced eXtensible Interface (AXI), part of the ARM Advanced Microcontroller Bus Architecture is a parallel high-performance, synchronous, high-frequency, multi-master, multi-slave communication interface, mainly designed for on-chip communication.

The AMBA specification defines three AXI4 protocols:

- AXI4: A high performance memory mapped data and address interface. Capable of Burst access to memory mapped devices.
- AXI4-Lite: A subset of AXI, lacking burst access capability. Has a simpler interface than the full AXI4 interface. Main features are that the address is a traditional Address/Data - Does not support burst (single address, single data) and supports only 32 or 64 bits data width.
- AXI4-Stream: A fast unidirectional protocol for transferring data from master to slave.

There are 5 different channels between the master and the slave.

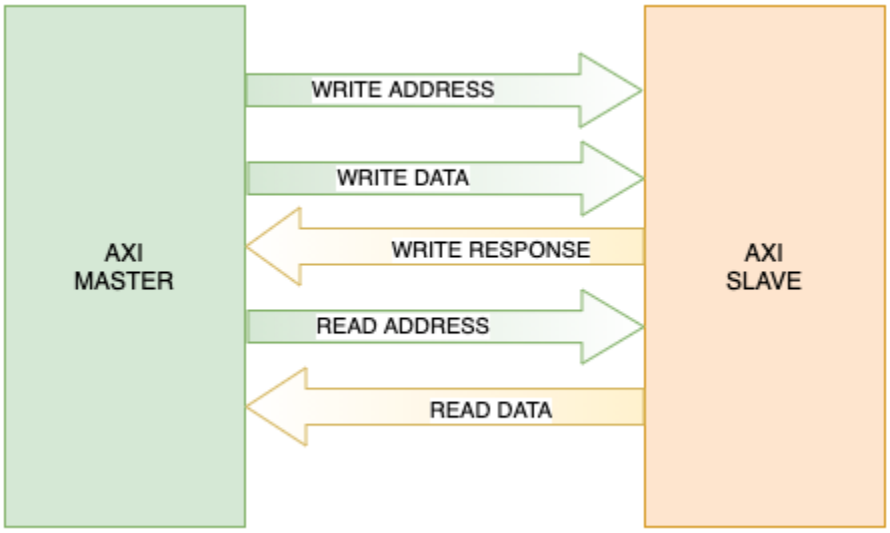

Fig 1: Channels between AXI4-Lite Master and Slave

In AXI4 Lite protocol, each channel uses the same VALID/READY handshake to transfer control and data information. The source generates the VALID signal to indicate when the data or control information is available. The destination generates the READY signal to indicate that it accepts the data or control information. Transfer occurs only when both the VALID and READY signals are HIGH.

Source sends the next DATA or de-asserts VALID. Destination de-asserts READY if it is no longer able to accept DATA.

We would be considering implementing multiple masters and multiple slaves, which would be connected by an interconnect.

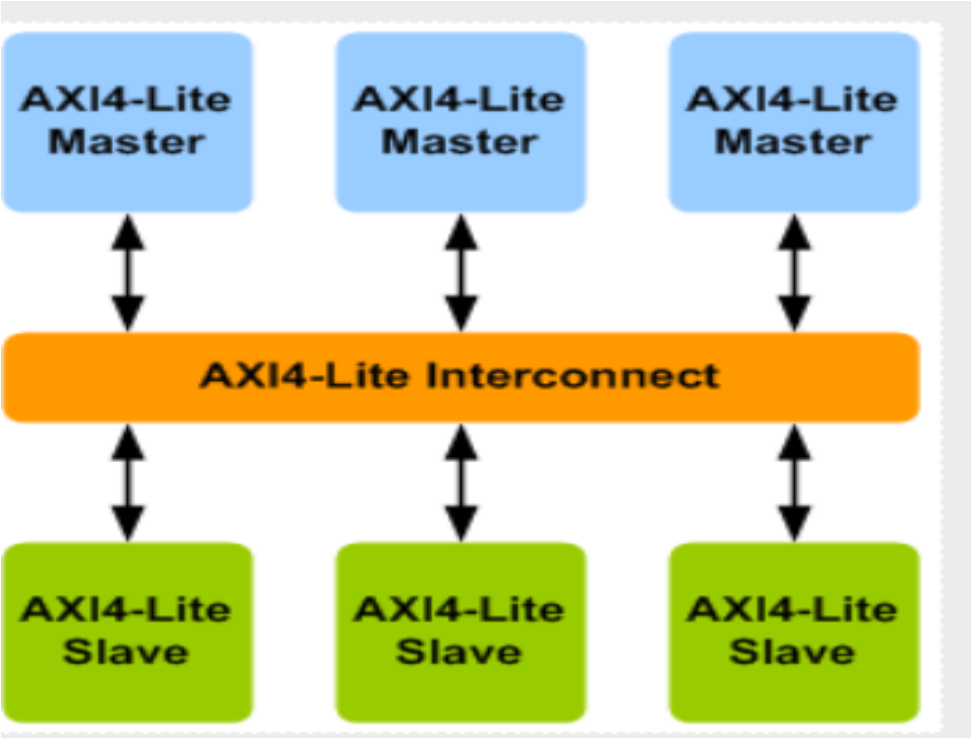

Fig.2: *AXI Interconnect Multi-Master/Multi-Slave*

2. Verification Requirements

2.1. Verification Levels

AXI4-Lite has a simple interface and the design consists of a master and slave unit. We will be verifying the master and slave individually at unit levels. The overall design will be verified at the System Level to test the protocol of AXI4-Lite between Master and Slave.

2.2. Functions

The following functions will be verified:

1. Write transactions between master and slave
2. Read transaction between master and slave
3. Different transfer types
4. Transactions from 2 masters to 2 slaves

The following functionalities will not be verified:

1. Protection types will not be verified
2. Data width will be restricted to 32 bits

2.3. Specific Tests & Methods

2.3.1. Type of Verification

A blackbox approach is chosen to verify the design thoroughly. The test stimulus will be driven into the DUT and an automated scoreboard will be used to verify the outputs.

2.3.2. Verification Strategy

We have decided to use deterministic and constrained random strategies to verify the design. In order to check the correctness of the design deterministic tests will be used to verify corner/edge cases as well as to test the basic functionality. More constrained tests will be added after analysing the coverage of existing tests. The stimulus send will be captured in a model

2.3.3. Abstraction Level

The verification will be transaction level and the lower level driver will drive cycle specific values to the DUV interface.

2.3.4. Checking

We are planning to use golden vector for unit level testing and a reference model for system level testing. At lower levels it is easier to use the golden vector as the testing will be deterministic. Since it will be difficult to maintain a golden vector for a more complex system level testing, a reference model is preferred.

Some of the checking are as follows:

1. All signals must be at their default when a reset is asserted
2. The order of write from a master is the same received by slave
3. Burst is always maintained as one
4. The read/write from a master is assigned to the correct slave (a write request from master 1 is send to slave 1)
5. Write address is X only when in reset
6. Write address value is X only when in reset

2.4. Coverage

Covergroups for each of the channels between Master and Slave:

- Write Address
- Write Data
- Read Address
- Read Data
- Write Response

Covergroups for Master & Slave reads and writes:

- Master Read
- Master Write
- Slave Read
- Slave Write
- Master Read with reset
- Master Write with reset
- Slave Read with reset
- Slave Write with reset

Covergroups for each of the channels with reset.

- Ready and valid signals of Write Address with reset
- Ready and valid signals of Write Data with reset
- Ready and valid signals of Read Address with reset
- Ready and valid signals of Read Data with reset
- Ready and valid signals of Write Response with reset

2.5. Scenarios

Single operations	
1	Write 4 bytes to a location
2	Write 3 bytes to a location
3	Write 2 bytes to a location
4	Write to two locations
5	Read 4 bytes from a location
6	Read 3 bytes from a location
7	Read 2 bytes from a location

Multiple Operations:	
1	Two writes to the same location and then Read
2	Write and then Read from a location
3	Two Writes to and then read from the 2nd location
4	Read and then write from a location
5	Write to 2-3 locations and read from the written location
6	Read and then write at different locations
7	Write after reset
8	Reset after write
9	Read after reset
10	Reset after Read
11	Provide address in one cycle and data in the next for a write
12	Random reads and writes
13	Send a write with strobe on byte 1
14	Send a write with strobe on byte 2
15	Send a write with strobe on byte 3
16	Send a write with strobe on byte 4
17	send a write with strobe on all bytes
18	send a write with strobe deasserted for all 4 bytes
19	multiple masters attempting to communicate with a single slave
20	One master trying to write to different slaves

Error Scenarios:	
1	Write to an invalid address
2	Read from an invalid address

Tests for interconnects:	
1	The bus protocol must be obeyed
2	No more than one master can be connected to any particular slave.
3	No more than one slave can be connected to any particular masters
4	Any unconnected slave should neither receive requests nor send replies.
5	Should check if the requested address belongs to one of the slaves, else should return error
6	Should generate error, if slave does not respond within a specified time
7	Grant for a master doesn't get revoked when waiting for slave to respond
8	Slaves should not send data/ set ready signal unless a master sends a request
9	Arbitration method used is maintained throughout

3. Project Management

3.1. Tools

1. QuestaSIM
2. Modelsim PE Student Edition

3.2. Risks/Dependencies

1. The short schedule is a risk factor for the completion of project
2. Development of the environment using OOP is new to the team members and is a potential risk
3. The plan does not have any other potential risks or dependencies.

3.3. Resources

Team consists of three members:

1. Amrutha Anil (amrutha@pdx.edu)
2. Durganila Anandhan (anandhan@pdx.edu)
3. Manjari Rajasekharan (manjari@pdx.edu)

3.4. Schedule

05/14/2021	Unit level testing
05/19/2021	System level testing
05/23/2021	Analysis of verification plan and implement missing cases
Future	Convert the testbench to UVM based test