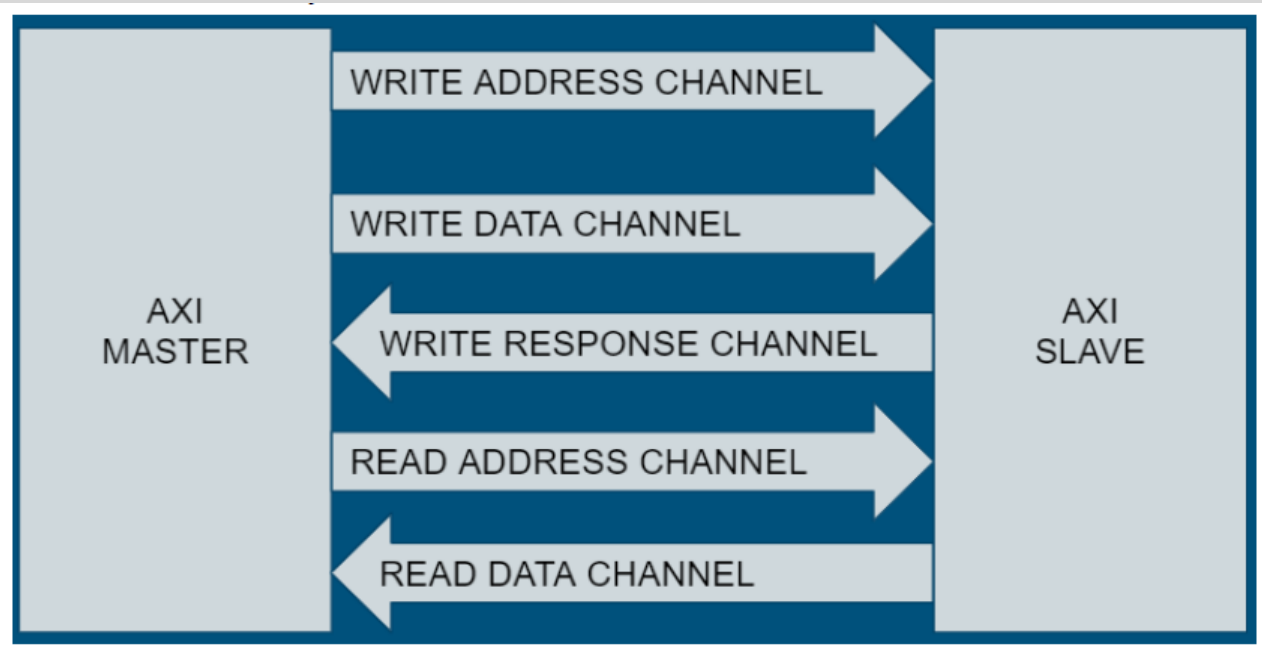
VERIFICATION OF AMBA AXI4-LITE PROTOCOL

Fall 2021
ECE 571 Project Group1
Narendra Srinivas R
Pradeep Reddy M
Naveen Manivannan

Introduction to AXI4-Lite

- The ARM Advanced Microcontroller Bus Architecture (AMBA)
- 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 (single address, single data) and supports only 32 or 64 bits data width.
- Read and writes are non buffer-able, non modifiable.

Block Diagram



- Master slave communicates using the following slides.
- Uses unidirectional communication.
- Master is a processor and slave is a memory system.

Channels used

- We used five Channels and each is used for a specific functionality.
- Write Address Channel
- Write Data Channel
- Write Response Channel
- Read Address Channel
- Read Data Channel

Handshaking (communication b/w master and slave)

Each channel has its own VALID/READY and uses the same for handshake, in order to transfer control and data information. When the data or control information is available, the source asserts the VALID signal to indicate. The destination asserts the READY signal to indicate that it is available to accept the data or control information. The transfer happens only if both the VALID and READY signals are ASSERTED. Source sends the next DATA or de-asserts VALID. Destination de-asserts READY if it is no longer able to accept DATA.

Write and Read Transaction

- Explanation of signals asserted and communication between master and slave.
- Uses control signals like Ready, Valid.
- Each channel uses its own signals.
- Different for reads and writes.
- If possible explain for an example with block diagram.

Verification Plan

- Verify all possible scenarios to identify any bugs in the broken code.
- Using testcases and Assertions to identify the bugs in design.
- Also keep in mind that reset and control signal takes priority.
- Ex: Writing to location that is not in the memory array. Writing to successive locations and see its is successful.
- Ex: AWADDR_STABLE_a

Test cases …

- To test if the design is working as expected.
- Test case to verify the following.
- Write
- Read
- Read followed by a write.
- Write followed by a read.
- Reset during write/ Read.
- Read and write to a location which is not in memory.

Assertions used…

- Consider Write address channel
- AWADDR_STABLE_a - AWADDR remains stable when AWVALID is asserted and AWREADY is LOW.
- AWVALID_STABLE_a –when Advalid is high it should remain high until adready is high.
- AWADDR_xcheck_a – to make sure there are no unknown values in address.
- We have written similar assertions in all other channels as well.
- Code walkthrough if possible to show other assertions.

Code walk-through..

- A quick overview of Design code.
- Open code and walk through the code to show the overall hierarchy of Test bench
- Explain Assertions used.
- Other possible testcases.

BUGS DETECTED

ASSERTION ERRORS FOUND

- RDATA_xcheck_a:- The RDATA contains x values when RVALID was high.
- AWADDR_stable_a :- AWADDR was not stable, when AWVALID was asserted and AWREADY was low
- WDATA_stable_a:- WDATA was not stable, when WVALID was asserted and WREADY was low.

DATA ERRORS FOUND

- The Data fetched was not matched with expected data(TBMEM).The data was returned was following a pattern like double the expected data.

Work split-up

Responsibilities	Members
Environment Setup	Pradeep,Srinivas,Naveen
Simulation	Pradeep,Srinivas
Unit level testing and Assertions for Write Address,Write data Channel	Srinivas,Naveen
Unit level testing and Assertions for Read Address,Read data Channel	Pradeep,Srinivas
Unit level testing and Assertions for Write Response Channel and Global Coverage	Naveen,Pradeep

Suggestions / Questions

Thank You