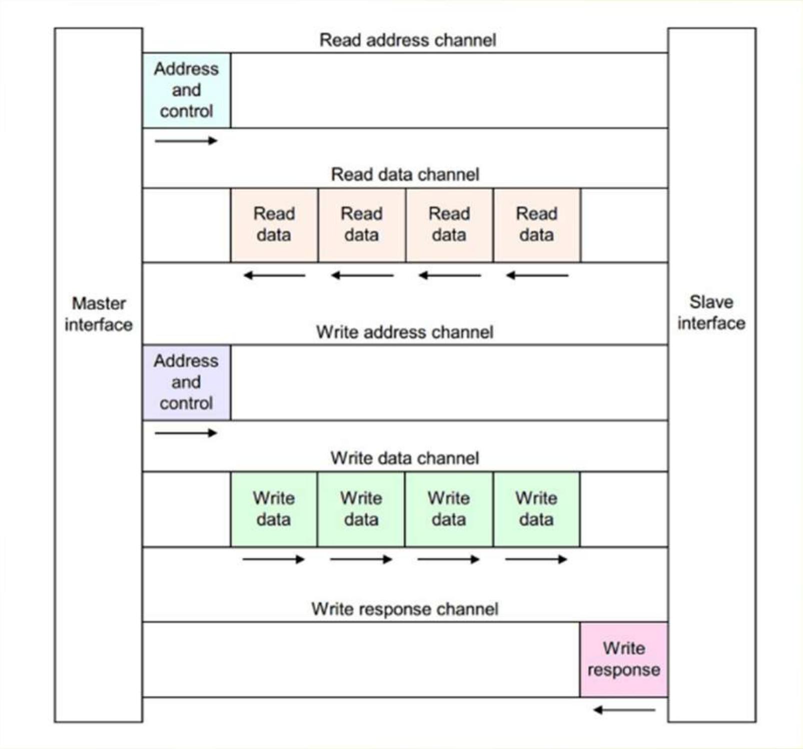
AXI Introduction

- AXI, or Advanced extensible Interface, is a high-performance interface protocol developed by ARM for connecting components in System-on-Chip (SoC) designs. It is part of the AMBA (Advanced Microcontroller Bus Architecture) specification, which aims to standardize communication between various blocks in a chip.

Overview of AXI

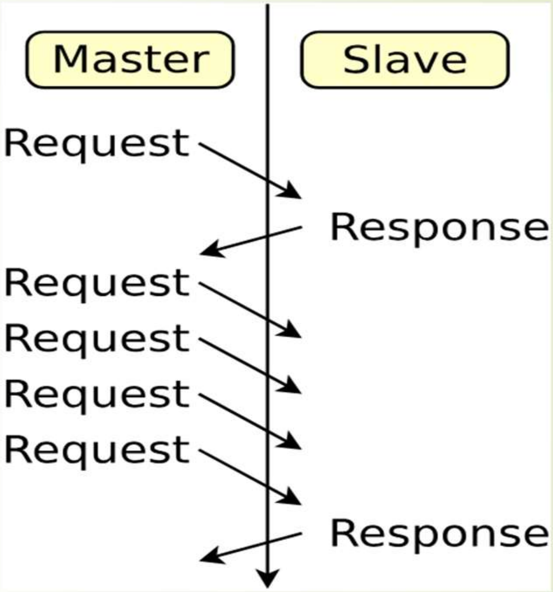
The AXI utilizes dedicated channels for data, address, and control signals to facilitate efficient communication between masters and slaves.

- Write Address Channel
- Write Data Channel
- Write Response Channel
- Read Address Channel
- Read Data/Response Channel

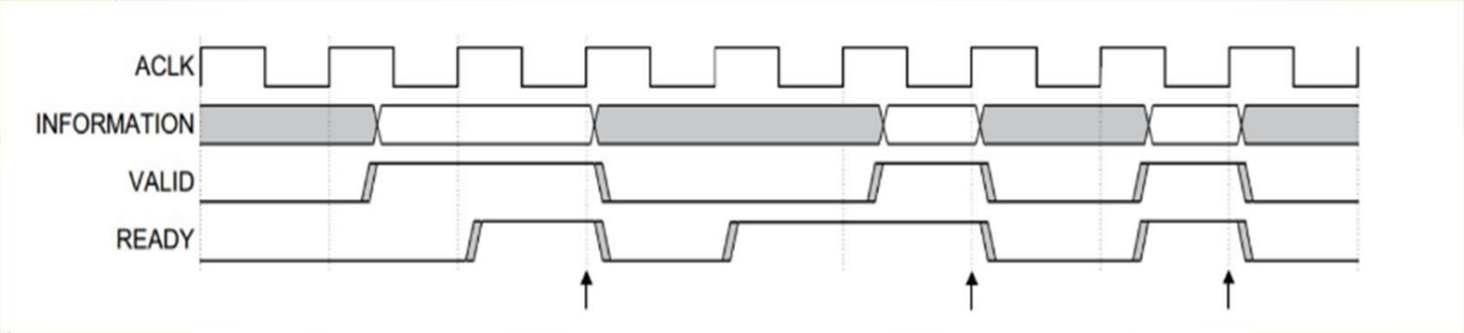


Key Features of AXI

- Burst based transaction with start-address
- Five channel Architecture
- Multiple outstanding Addresses(Multiple request)
- Separate channel for read and write operations
- Aligned and Non-Aligned Address support
- Higher Bandwidth



Handshake Mechanism



- **VALID**: Indicates that the sender (master or slave) has valid data or an address to send.
- **READY**: Indicates that the receiver (slave or master) is ready to accept data or the address.
- The data transfer happens only when both the VALID and READY signals are asserted (set to high).

AXI Signals

WRITE ADDRESS CHANNEL

- AWID
- AWVALID
- AWADDR
- AWREADY
- AWBURST
- AWSIZE
- AWLEN

WRITE DATA CHANNEL

- WID
- WVALID
- WDATA
- WREADY
- WSTRB
- WLAST

WRITE RESPONSE CHANNEL

- BID
- BVALID
- BRESP
- BREADY

READ ADDRESS CHANNEL

- ARID
- ARVALID
- ARADDR
- ARREADY
- ARBURST
- ARSIZE
- ARLEN

READ DATA CHANNEL

- RID
- RVALID
- RDATA
- RREADY
- RRESP
- RLAST

AXI Burst Transfer

AXI supports burst transfers, which allow for multiple data transfers with a single address and control signal, reducing overhead and enhancing efficiency

FIXED LENGTH

Fixed length burst transfer a predefined Number

Data words

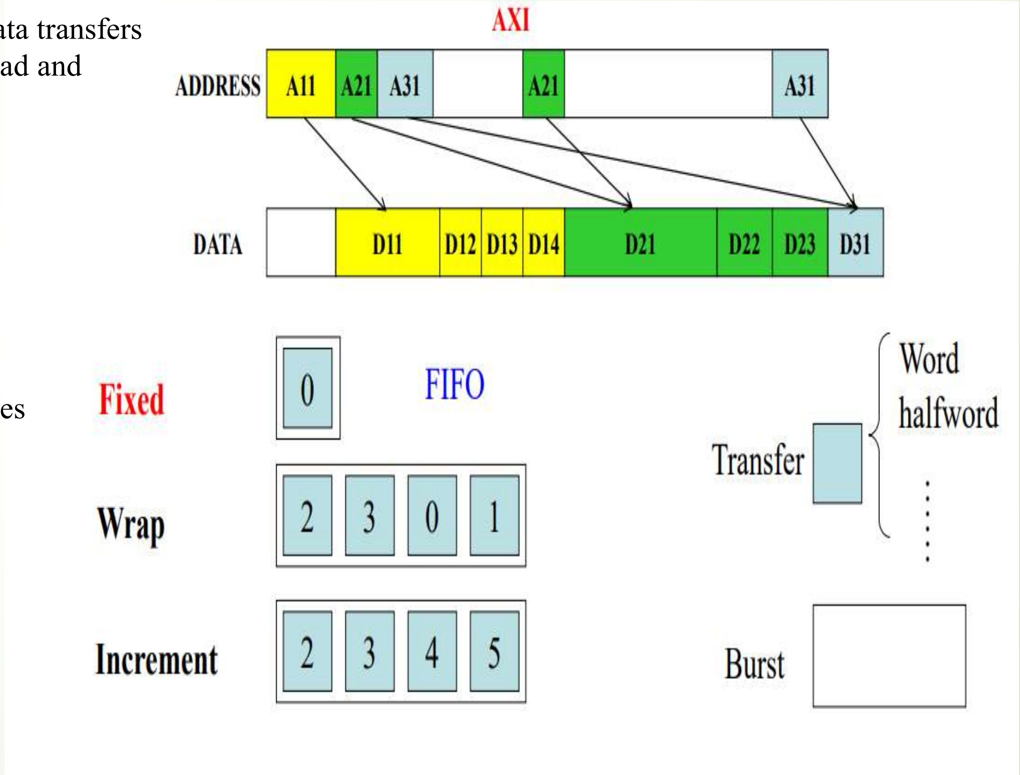
INCREMENT

Increment burst transfer data words at sequential addresses

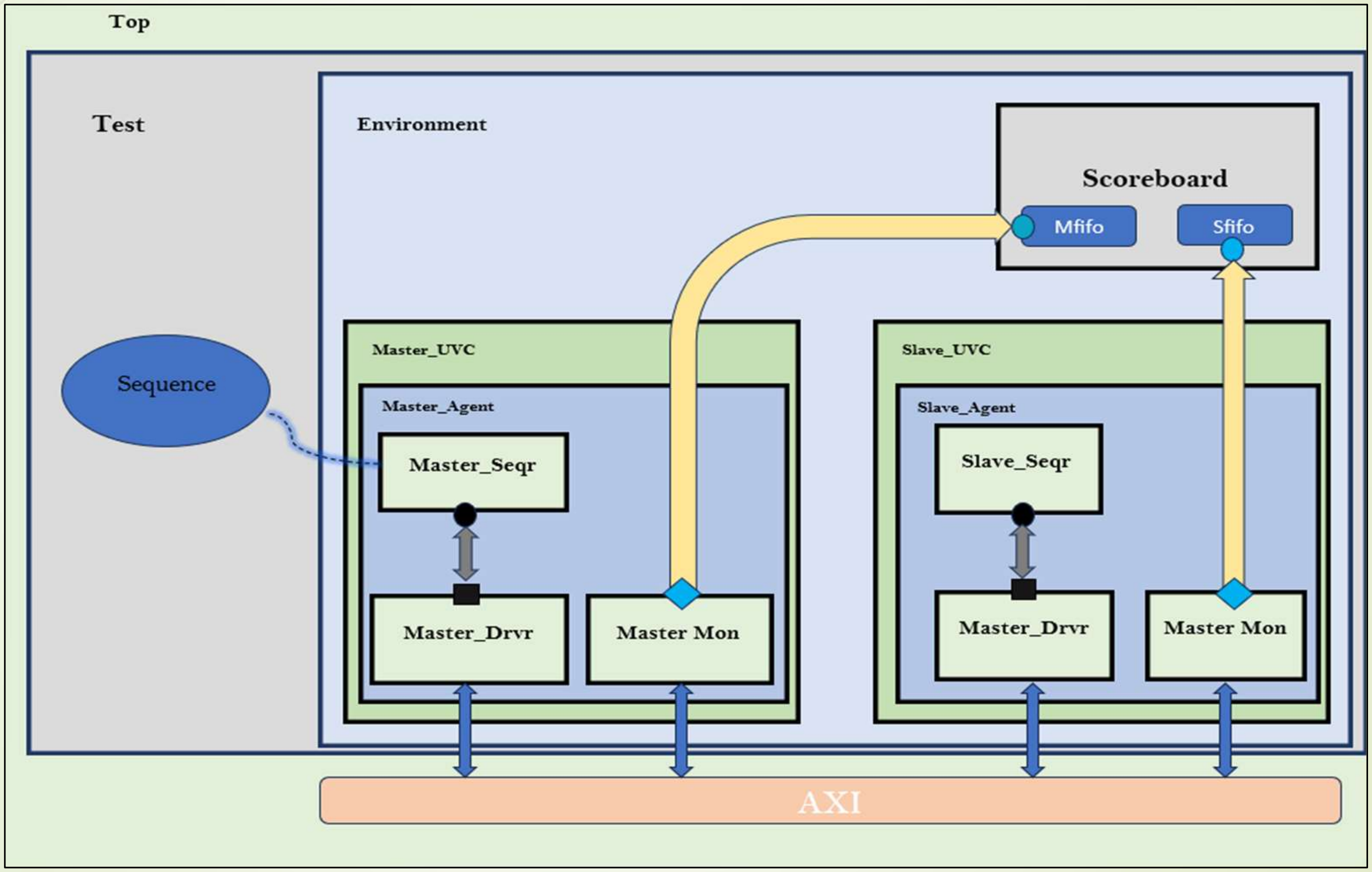
WRAP

Wrap burst transfer data words with Addresses

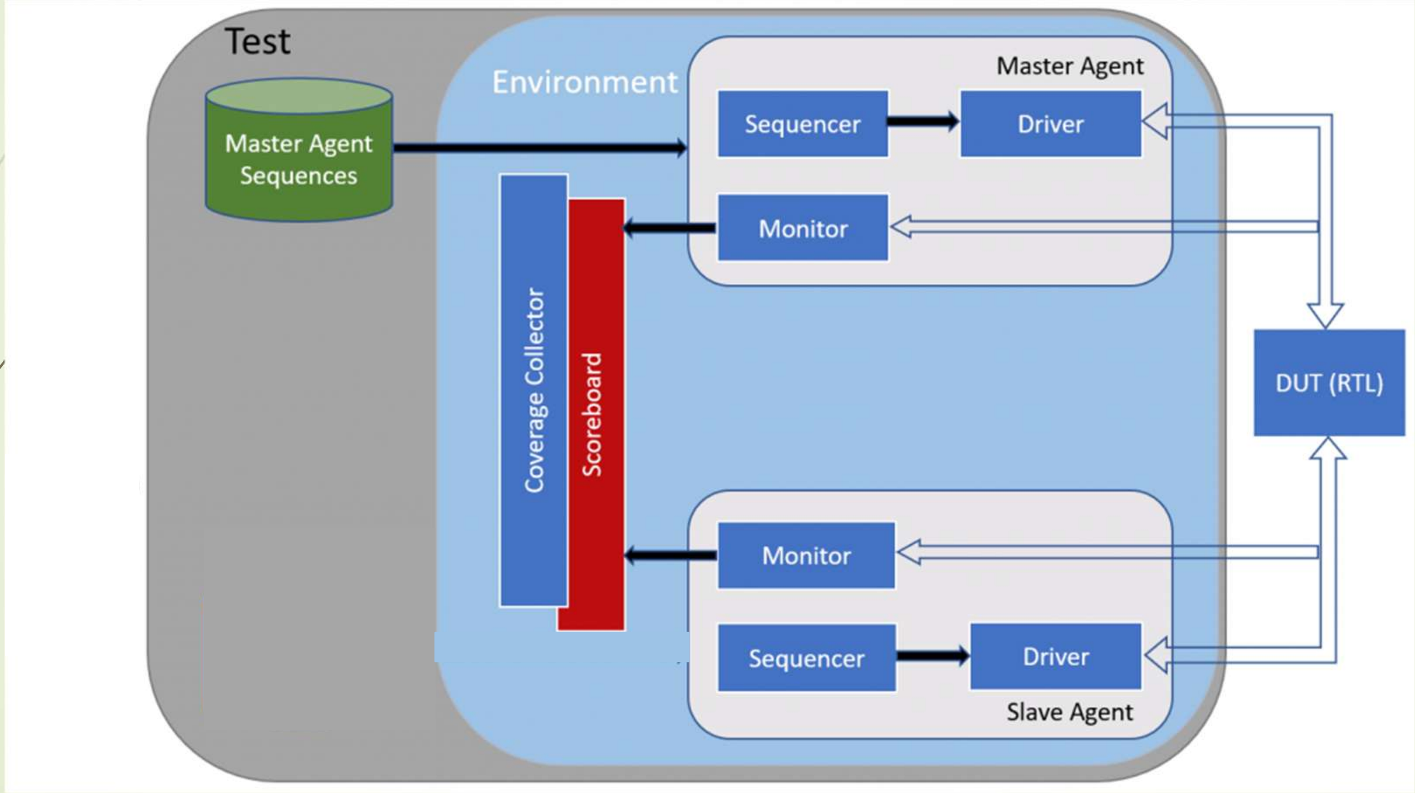
That wrap around a defined boundary



TB Architecture



Scoreboard


Test
Master Agent Sequences
Environment
Master Agent
Sequencer
Driver
Monitor
Coverage Collector
Scoreboard
DUT (RTL)
Monitor
Sequencer
Driver
Slave Agent

Conclusion

- ✓ AXI verification is critical for ensuring high-performance and reliable data transfer in SoCs.
- ✓ Using UVM, System Verilog, and industry-standard tools helps achieve thorough verification and high functional coverage.
- ✓ Robust AXI verification is essential for meeting the demands of modern digital systems

Thank You

Any Questions?