**VLSI TECHNOLOGY PAGE**

**AXI4VIP SPECIFICATION:**

**DESIGN DESCRIPTION**

Author:

Nguyen Hung Quan

Nguyen Huy

Le Quang Vu

CONTENT

[CONTENT ii](#_Toc92103536)

[LIST OF FIGURES iii](#_Toc92103537)

[LIST OF TABLES iv](#_Toc92103538)

[CONVENTION, ACRONYMS, AND ABBREVIATIONS v](#_Toc92103539)

[1 AXI4VIP OVERVIEW 1](#_Toc92103540)

[1.1 Features 2](#_Toc92103541)

[1.1.1 Common 2](#_Toc92103542)

[1.1.2 AXI master and slave mode 2](#_Toc92103543)

[1.1.3 Active and passive mode 2](#_Toc92103544)

[1.1.4 Checker and assertion 3](#_Toc92103545)

[1.2 Integration 3](#_Toc92103546)

[1.3 Directory and file structure 6](#_Toc92103547)

[BIBLIOGRAPHY 10](#_Toc92103548)

LIST OF FIGURES

[Figure 1‑1: AXI4VIP master integration 3](#_Toc92103599)

[Figure 1‑2: AXI4VIP slave integration 4](#_Toc92103600)

[Figure 1‑3: Full AXI4VIP master/slave integration block diagram 5](#_Toc92103601)

LIST OF TABLES

[Table 1‑1: Directory and file structure 7](#_Toc92103550)

# ****CONVENTION, ACRONYMS, AND ABBREVIATIONS****

| **Convention, acronyms, and abbreviations** | **Description** |
| --- | --- |
| API | Application Programming Interface |
| DUT | Design Under Test |
| AMBA | Advanced Microcontroller Bus Architecture |
| AXI | Advanced eXtensible Interface.  It is replaced to AXI3 or AXI4 |
| AXI3 | AMBA 3 AXI |
| AXI4 | AMBA 4 AXI |
| IP | Intellectual Property |
| RTL | Register Transfer Level |
| TLM | Transaction Level Modeling |
| UVM | Universal Verification Methodology |
| VIP | Verification IP |
|  |  |

# AXI4VIP OVERVIEW

...

## Features

### Common

AXI4VIP is the VIP is compatible with AMBA 4 AXI (AXI4) protocol, and a part of AMBA 3 AXI (AXI3)

* Support both AMBA AXI4 master and slave mode
* Support both active and passive mode
* Configure the bit width of address and data bus
* Configure the transfer type, response type
* Support the AXI protocol checker and assertion
* Support the debug report
* Support outstanding and out-of-order feature
* Drive WID which is supported only in AXI3
* Do not support “send a write data before sending the write address” that is accepted only in AXI3

### AXI master and slave mode

In AXI master mode, VIP is an AXI master, which initializes the bus transactions. VIP connects to an interface of AXI slave.

In AXI slave mode, VIP is an AXI slave, which responses the AXI requests, read or write request. VIP connects to an interface of AXI master.

### Active and passive mode

An AXI master or slave agent can be configured as the active or passive agent. In passive mode. Agent has the driver to drive the AXI interface. In passive mode, agent is only the monitor to capture the transfers on the AXI interface.

### Checker and assertion

Checker or assertion is implemented to check the state of AXI interface, and AXI protocol rules. User can enable or disable VIP assertions.

## Integration

AXI4VIP can be used to verify a AXI master, AXI slave, or interconnect (bus).



Figure ‑: AXI4VIP master integration



Figure ‑: AXI4VIP slave integration



Figure ‑: Full AXI4VIP master/slave integration block diagram

## Directory and file structure

<https://github.com/nguyenquanicd/AXI4VIP>

Table 1‑1: Directory and file structure

| **Directory and File** | **Description** | **Parent class** | **Priority** |
| --- | --- | --- | --- |
| rtl | RTL code (DUT) | - | 1 |
| vip | UVM components | - |  |
| |-- vt\_axi\_mst\_agent.sv | AXI master agent | uvm\_agent | 7 |
| |-- vt\_axi\_mst\_driver.sv | AXI master driver | uvm\_driver | 5 |
| |-- vt\_axi\_mst\_sequencer.sv | AXI master sequencer | uvm\_sequencer | 6 |
| |-- vt\_axi\_mst\_transaction.sv | AXI master transaction | vt\_axi\_transaction | 4 |
| |-- vt\_axi\_mst\_cfg.sv | AXI master configuration | vt\_axi\_cfg | 4 |
| |-- vt\_axi\_slv\_agent.sv | AXI slave agent | uvm\_agent | 7 |
| |-- vt\_axi\_slv\_driver.sv | AXI slave driver | uvm\_driver | 5 |
| |-- vt\_axi\_slv\_sequencer.sv | AXI slave sequencer | uvm\_sequencer | 6 |
| |-- vt\_axi\_slv\_transaction.sv | AXI slave transaction | vt\_axi\_transaction | 4 |
| |-- vt\_axi\_slv\_cfg.sv | AXI slave configuration | vt\_axi\_cfg | 4 |
| |-- vt\_axi\_monitor.sv | AXI monitor is used for both master and slave VIP | uvm\_monitor | 5 |
| |-- vt\_axi\_transaction.sv | Base AXI transaction | uvm\_sequence\_item | 3 |
| |-- vt\_axi\_typedef.sv | Data type defines | - | 2 |
| |-- vt\_axi\_defines.svh | VIP defines | - | 2 |
| |-- vt\_axi\_pkg.sv | Package definition | - | 8 |
| |-- vt\_axi\_cfg.sv | VIP top configuration | uvm\_config\_db | 3 |
| |-- vt\_axi\_interface.sv | AXI interface | - | 1 |
| |-- vt\_axi\_func\_cov.sv | Function coverage definition | - | 8 |
| |-- vt\_axi\_assertion.sv | Protocol assertion | - | 8 |
| env | Environment components | - |  |
| |-- vt\_axi\_environment.sv | Hierarchical container of components | uvm\_env | 9 |
| |-- vt\_axi\_testbench.sv | Base test | uvm\_test | 9 |
| |-- vt\_axi\_scoreboard.sv | Default scoreboard | uvm\_scoreboard |  |
| |-- vt\_axi\_env\_pkg.sv | Top environment package definition | - |  |
| |-- vt\_axi\_vir\_sequencer.sv | Virtual sequencer wraps agents in the multi-agent environment | uvm\_sequencer |  |
| |-- vt\_axi\_error\_catcher.sv | Catch the error and warning for regression | - |  |
| test | Testcase files | - |  |
| |-- vt\_axi\_base\_test.sv | Base testcase (test) | uvm\_sequence | 10 |
| |-- vt\_axi\_test\_pkg.sv | Test package | - | 10 |
| seq | Sequence files |  |  |
| |-- vt\_axi\_mst\_base\_seq.sv | AXI master sequence | uvm\_sequence | 11 |
| |-- vt\_axi\_slv\_base\_seq.sv | AXI slave sequence | uvm\_sequence | 11 |
| |-- vt\_axi\_seq\_pkg.sv | Sequence package |  | 11 |
| run | Working directory and the testcase execution | - |  |
| |-- run\_sim.py | Core simulation script | - | 13 |
| |-- compile.f | Source file list | - | 12 |
| |-- Makefile | User interface commands | - | 13 |
| project.env | System environment variables | - | 12 |
| img | Project images for Github README | - |  |
| doc | Specification documents | - |  |
| tmp | Template source code (unused) | - |  |

# ****BIBLIOGRAPHY****

1. ...