**AHB Protocol Verification Environment Specification**

Table of Contents

[Document history 2](#_Toc109746558)

[1.Introduction 4](#_Toc109746559)

[1.1 Scope 4](#_Toc109746560)

[1.2 Abbreviations 4](#_Toc109746561)

[1.3 References 4](#_Toc109746562)

[2. AHB Protocol Overview 5](#_Toc109746563)

[2.1. AHB Protocol Features 5](#_Toc109746564)

[2.2. AHB Protocol Interfaces 5](#_Toc109746565)

[3. Verification Environment 5](#_Toc109746566)

[3.1 Environment Overview 5](#_Toc109746567)

[3.2 Components description 6](#_Toc109746568)

[3.4 Checkers 6](#_Toc109746569)

[3.5 Coverage 6](#_Toc109746570)

[3.6 Tests 6](#_Toc109746571)

[4. Simulation flow 6](#_Toc109746572)

[5. How to run a test 7](#_Toc109746573)

[6. Limitations and assumptions 7](#_Toc109746574)

# Document history

| Version | Date | Approved by | Created by | Description |
| --- | --- | --- | --- | --- |
| 0.1 | 8.11.2022 | Liviu Ababei | Ioana Ailenei Andreea Haldan | Document created |
| 1.0 | {Date} | {Trainer} | {Trainee} | Revised by {Name} |

# 1.Introduction

## 1.1 Scope

This document describes the verification environment for the AHB Protocol.

## 1.2 Abbreviations

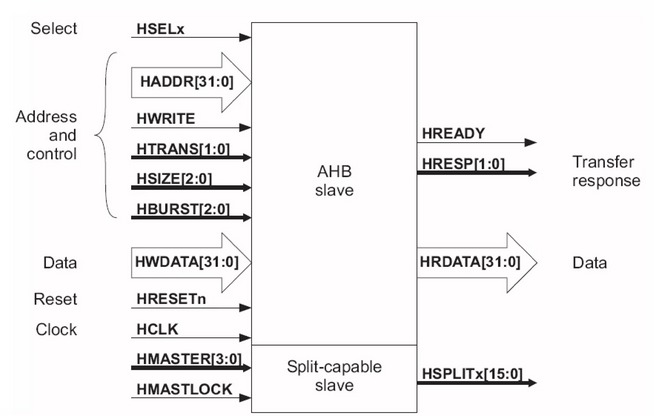
**DUT =>** Device Under Test

**AHB**  **=>** Advanced High-permormance Bus

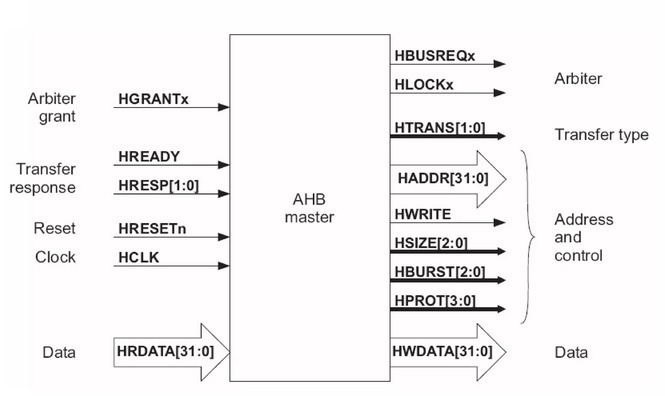
## 1.3 References

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Name** | **Revision** | **Description** |
| 1. | AHB Metric plan.xlsx | 1.0 | Coverage, Checkers and Test plans |
| 2. | 1. IHI0033C\_amba\_ahb\_protocol\_spec.pdf | x.x | Protocol AHB specification |

# 2. AHB Overview

* 1. 

1. Figure 1. The MASTER AHB input/output interfaces & architecture diagram



1. Figure 2. The SLAVE AHB input/output interfaces & architecture diagram

## 2.1. AHB Features

* Burst transfers
* Single clock-edge operation
* Non-tristate implementation
* Configurable data bus widths
* Configurable address bus widths

## 2.2. AHB Interfaces

1. The data\_buffer has the following interfaces

* *global*

|  |  |  |
| --- | --- | --- |
| 1. **Name** | 1. **Direction** | 1. **Width** |
| 1. clk | 1. input |  |
| 1. rst\_n | 1. input |  |

* *response*

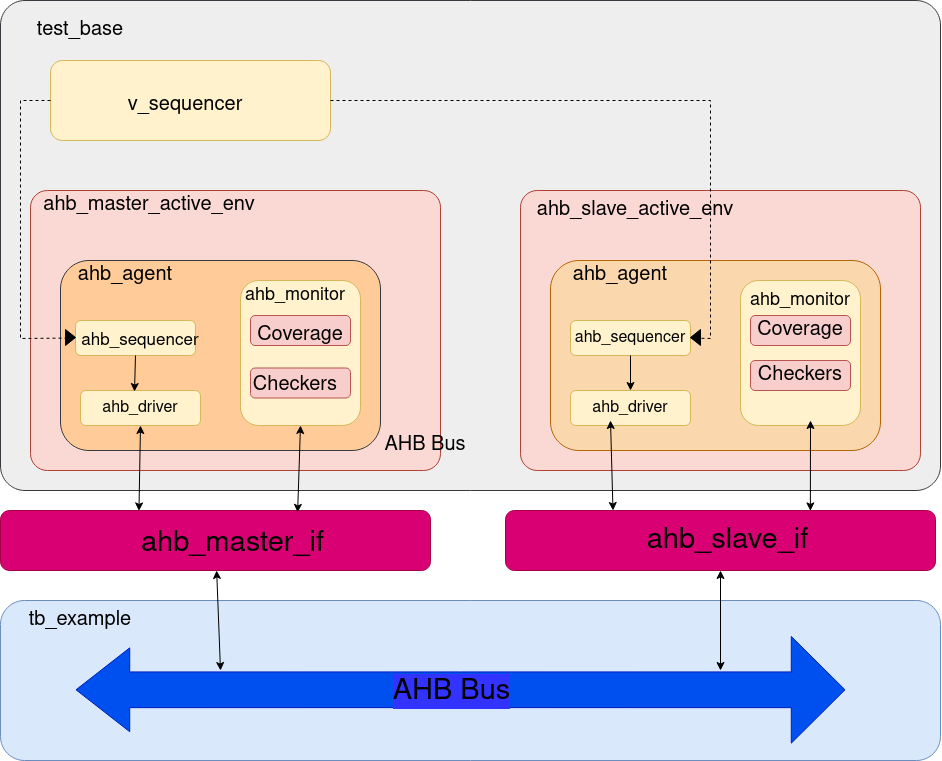
|  |  |  |
| --- | --- | --- |
| 1. **Name** | 1. **Direction** | 1. **Width** |
| 1. hrdata | 1. output | 1. [31:0] |
| 1. hreadyout | 1. output |  |
| 1. hresp | 1. output |  |

* *request*

|  |  |  |
| --- | --- | --- |
| 1. **Name** | 1. **Direction** | 1. **Width** |
| 1. hsel | 1. input |  |
| 1. htrans | 1. input | 1. [1:0] |
| 1. hburst | 1. input | 1. [2:0] |
| 1. hsize | 1. input | 1. [2:0] |
| 1. haddr | 1. input | 1. [31 : 0] |
| 1. hwrite | 1. input |  |
| 1. hwdata | 1. input | 1. [31 : 0] |
| 1. hwstrobe | 1. input | 1. [3:0] |

# Verification Environment

## 3.1 Environment Overview

* 1. The AHB Protocol verification environment diagram is presented in Figure 2.
  2. 

1. Figure 2. The AHB Protocol Verification Environment Diagram

## 3.2 Components description

**Sequence**: series of transactions

**Sequencer**: generates the transactions and drives the sequences to the drives

**Driver**: drives the signals to reset state during the reset and when it is out of reset, drives AHB transactions and receives the transaction from sequencer via a TLM .

**Monitor**: monitors the AHB interface, collects data from the bus and encapsulates in an item to be sent to the higher level via an analysis port for further checking. Also, it collects coverage information and checks some protocol specifications (data and control signals are stable until hsel and hreadyout are asserted, hburst generates a correct transfer: correct number of address, htrans has the corrrect flow: e.g.: after IDLE comes NONSEQ).

**Agent**: instantiates sequencer, driver and monitor and connects the components via TLM interfaces. It can be active(drives and monitors) or passive(monitors).

## 3.4 Checkers

The checkers list can be found in AHB Protocol metric driven plan.xlsx, the “Checkers” sheet.

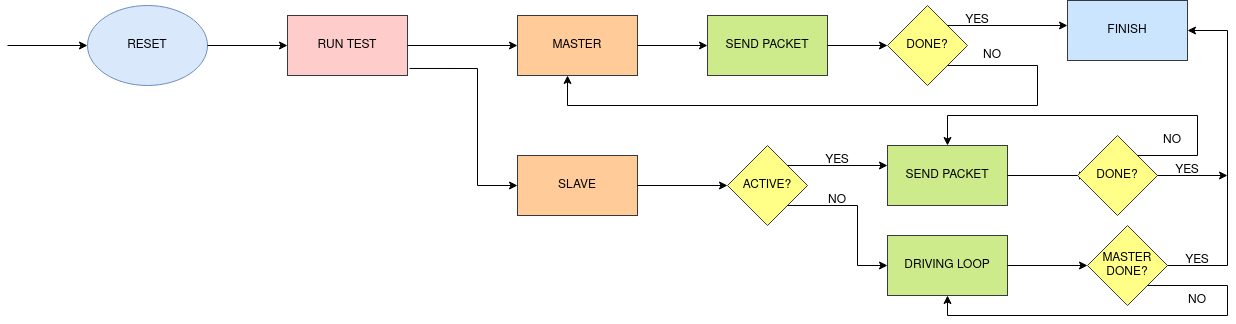
## 3.5 Coverage

The functional coverage description can be found in AHB Protocol metric driven plan.xlsx, the “Coverage” sheet.

## 3.6 Tests

The tests’ description can be found in AHB Protocol metric driven plan.xlsx, the “Tests” sheet.

# 4. Simulation flow



1. Reset the AHB Bus. Generate reset and clock in top, then drive the signals in idle state to interface

2. Start a Master Sequence and a Slave Sequence in a fork.

3. Master Sequence Send Packet: Generates the random transfer packet, send the packet to driver, then drive the signals to the interface accordingly with the protocol specifications. Monitors the interface forever and forms packets to send to the higher level, checks and collect coverage.

4. Slave Sequence Send Packet: If Active: Generates the random transfer packet, send the packet to driver, then drive the signals to the interface accordingly with the protocol specifications. Monitors the interface forever and forms packets to send to the higher level, checks and collect coverage. If Reactive: looks on interface forever and put random data on interface when is needed to. Also monitors.

5. Done?: If we have more sequences to run then step 2, 3.active will repeat, if not, we finish the test. 3.reactive will stop when the master will stop.

# 5. How to run a test

[Script to use]

1. ../example/run.ius

[From where to run]

1. ../example

[Example of command line parameters]

1. run\_ius test\_example\_1

# 6. Limitations and assumptions

[What doen’t work]

reset on fly

[What assumptions you took on DUT functionality]

[What is still to be verified]

[What is not supported in the VE from the DUT functionality]