**AMBA 3 APB VIP**

**“AMBA 3 APB configurable VIP”**

Functional Verification Plan

Revision 0.1

EITRA

Crafting Preferred Engineers

Contents

[1 Document Details 4](#_Toc419795954)

[1.1 Revision History 4](#_Toc419795955)

[1.2 Definition, Acronyms, and Abbreviations 5](#_Toc419795956)

[1.3 References 6](#_Toc419795957)

[1.4 Open Items 6](#_Toc419795958)

[2 Overview 7](#_Toc419795959)

[2.1 Scope 7](#_Toc419795960)

[2.2 Verification Flow 7](#_Toc419795961)

[2.3 Verification Objectives 7](#_Toc419795962)

[2.4 Verification Methodology 7](#_Toc419795963)

[2.5 Verification Strategy 7](#_Toc419795964)

[3 Verification Environment Architecture 8](#_Toc419795965)

[3.1 Overview of <Project Name> Verification Architecture 8](#_Toc419795966)

[4 DUV Considerations Relevant To Verification 9](#_Toc419795967)

[5 Verification component 10](#_Toc419795968)

[6 Verification setup 11](#_Toc419795969)

[6.1 Tool Versions 11](#_Toc419795970)

[6.2 VIP Versions 11](#_Toc419795971)

[6.3 Directory Structure 11](#_Toc419795972)

[6.4 Project Setup 11](#_Toc419795973)

[6.5 Simulation Script Usage 11](#_Toc419795974)

[6.5.1 Running Single Testcase 11](#_Toc419795975)

[6.5.1 Running Regressions 12](#_Toc419795976)

[6.5.2 Compilation define and Simulation Argument Switch 12](#_Toc419795977)

[7 Coverage & Regression 12](#_Toc419795978)

[7.1 Running Regression 12](#_Toc419795979)

[7.2 Code coverage 12](#_Toc419795980)

[7.3 Assertion coverage 12](#_Toc419795981)

[7.4 Functional coverage 12](#_Toc419795982)

[7.4.1 Directed-Random Testing 12](#_Toc419795983)

[7.4.1.1 Objectives of Directed Random Testing 12](#_Toc419795984)

[7.4.1.2 Why Functional Coverage? 12](#_Toc419795985)

[7.4.2 How do we decided functional coverage bins / What to randomize ? 12](#_Toc419795986)

[7.4.2.1 When / Where /How to sample data to be covered? 12](#_Toc419795987)

[7.4.3 Regression with Functional Coverage 13](#_Toc419795988)

[7.4.3.1 To achieve functional coverage 13](#_Toc419795989)

[7.4.4 Directed-Random Testing : Self Checking 13](#_Toc419795990)

[7.4.4.1 How to check correctness of test with directed-random testing 13](#_Toc419795991)

[8 Assumptions and Limitation 13](#_Toc419795992)

[9 Final Release checklist 13](#_Toc419795993)

[10 Appendix A: References 13](#_Toc419795994)

[10.1 Revision History 13](#_Toc419795995)

[10.2 Items Out Of Scope of Verification 13](#_Toc419795996)

**Figures**

**No table of figures entries found.**

**Tables**

[Table 1‑1 Revision History 4](#_Toc419796106)

[Table 1-2 Definition, Acronyms and Abbreviations 5](#_Toc419796107)

[Table 1-3 References 6](#_Toc419796108)

[Table 10‑1 References 13](#_Toc419796109)

[Table 10‑2 Revision History 13](#_Toc419796110)

# Document Details

## Revision History

Table 1‑1 Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Revision | Author | Description |
| 06/03/2019 | Rev 0.1 | Dipesh Makwana | Initial Draft |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Definition, Acronyms, and Abbreviations

Table 1-2 Definition, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Term | Description |
| DS | **D**esign **S**pecification document |
| DUV | **D**esign **U**nder **V**erification |
| DUT | **D**esign Under **T**est |
| UVM | **U**niversal **V**erification **M**ethodology |
| VE | **V**erification **E**nvironment |
| TB | **T**est **B**ench |
| Expected | The word “Expected” is used to indicate that the transaction value under consideration is prepared by VE or is reference value. The transaction received from RTL is then compared with this expected transaction for data integrity checks. |
| Actual | The word “Actual” is used to indicate that transaction value under consideration is driven by DUV. This driven transaction is sampled by VE and then compared with the expected transaction for data integrity checks. |

## References

Table 1-3 References

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Document | Version Date | Remarks |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 10 |  |  |  |

## Open Items

# Overview

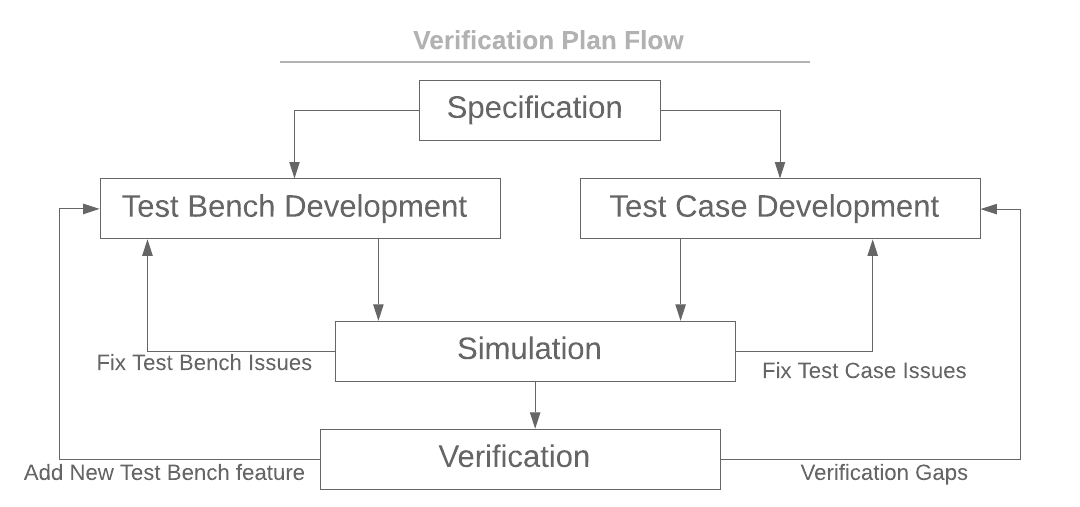
## Scope

This document specifies the plan for verifying the front-end design of <Project Name> for functional accuracy using simulations. It describes the flow, objectives, methodology, and strategy of execution for functional verification.

## Verification Flow

Functional verification approach is coverage driven constrained random verification.

Verification plan primarily driven by Test plan, Coverage plan, Checker plan.



Verification Flow

## Verification Objectives

|  |  |
| --- | --- |
| **Feature Name** | **Feature Description** |
| Write Transfer | write Transfer check |
| Read Transfer | Read Transfer check |
| Write Strobe | Write Strobe functionality Check |
| Transfer Faliure check | Check transfer faliure from slave: error responce |
| Extend Transfer | check wait state |

## Verification Methodology

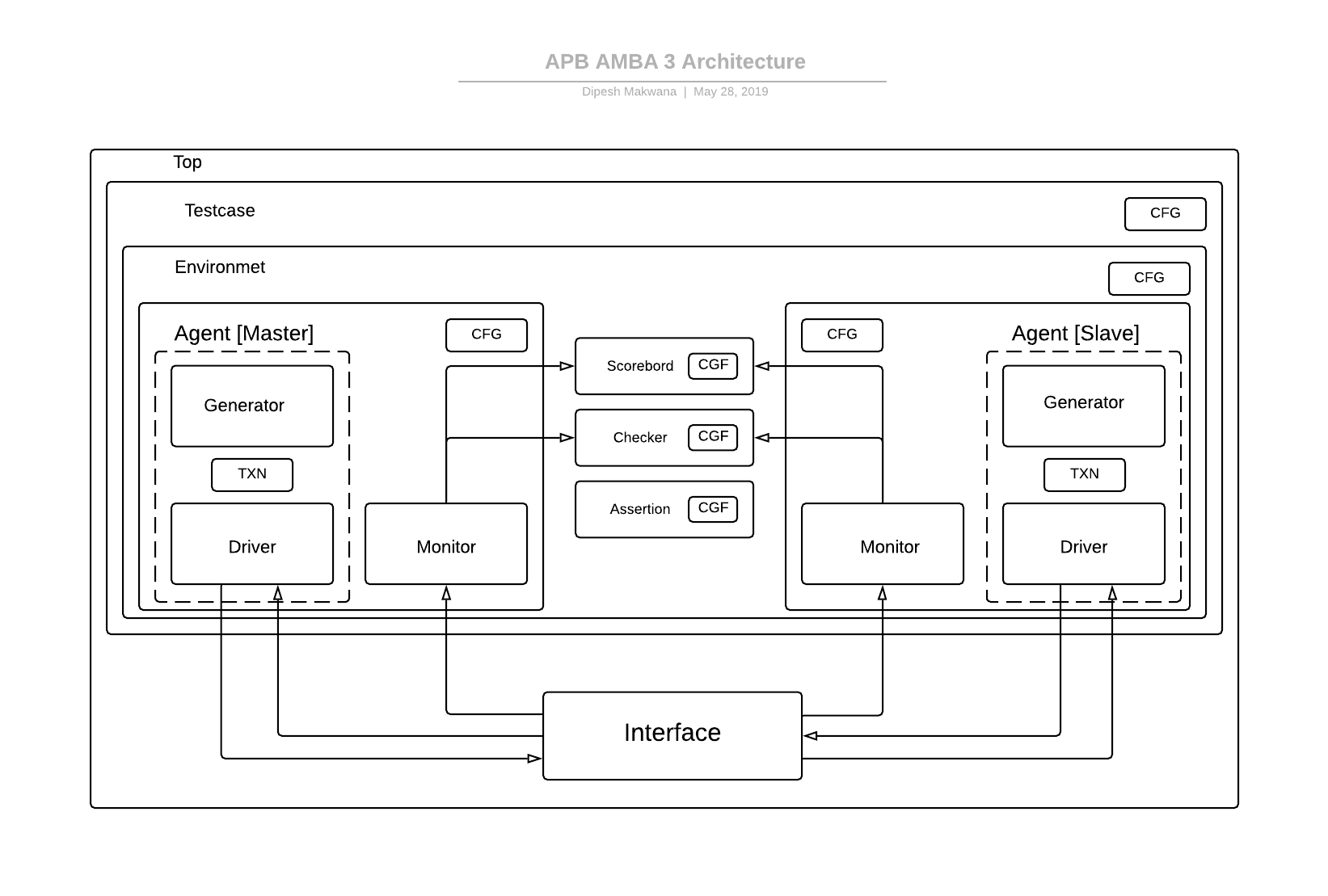
Coverage driven constrained random verification

## Verification Strategy

1. Preparing test bench and test cases
2. Preparing Assertion and coverage
3. Preparing Checker

# Verification Environment Architecture

## Overview of <Project Name> Verification Architecture



Detailed description of each component/object of verification environment. Explain with as much details possible. If require, then add multiple figures to demonstrate better.

Wherever applicable, add brief/detailed description, diagram, flow chart, tables, etc.

Cover following points per block/IP:

* Verification architecture diagram and description
* Verification component list and description
* Test Plan link
* Test Flow
* Features (To Be) Verified
* Functional Coverage Plan and Details
* Test case execution

# DUV Considerations Relevant To Verification

Brief description of design

List down block/feature list to be verified

# Verification component

SLAVE VIP:-

1.apb3\_s\_txn\_c

Description: Transaction Class

penable, pselx, pwrite = Type:- Single bit

paddr,pwdata, prdata= Type:- 32 bit.

Method name : txn\_do\_copy

Parameters passed : None

Returned parameters : None

Description : Deep copy.

2.apb3\_slv\_gen\_c

Description: Slave Generator Class

tr :-object of transaction class.

gen2driv = Type : mailbox

ended = Type : event

repeat\_count = Type:int

Method name : new

Parameters passed : gen2driv, ended

Returned parameters : None

Description : assigning memory to mailbox and event.

Method name : apb3\_slv\_gen\_run\_t

Parameters passed : none

Returned parameters : None

Description : randomization ,putting values in mailbox and after complete repeatation trigger the event

3. apb3\_slv\_drv\_c

Description: Slave Driver class.

no\_transaction => Type :- int

Description : to check number of transaction

apb3\_ms\_vir\_intf => Type :- virtual

Description: virtual interface

gen2driv => Type:- mailbox

Method name : new

Parameters passed : virtual interface and mailbox

Returned parameters : None

Description : Constructing memory for virtual interface and mailbox

Method name : apb3\_slv\_drv\_reset\_t

Parameters passed : None

Returned parameters : None

Description : to reset all the variables

Method name : apb3\_slv\_drv\_main\_t

Parameters passed : None

Returned parameters : None

Description : read write functionality

Method name : apb3\_slv\_drv\_run\_t

Parameters passed : None

Returned parameters : None

Description : Drive the signals.

4. apb3\_s\_intf

Description: Slave Interface

clocking driver

clocking monitor

modport : DRIVER

ports : clocking driver, clock, reset

modport : MONITOR

ports : clocking monitor, clock, reset

5. apb3\_slv\_mon\_c

apb3\_ms\_vif => Type :- virtual

Description: virtual interface

mon2scb => Type :- mailbox

Method name : new

Parameters passed : virtual interface and mailbox

Returned parameters : None

Description : Constructing memory for virtual interface and mailbox

Method name : apb3\_slv\_mon\_run\_t

Parameters passed : none

Returned parameters : None

Description : monitors data and transfer to scoreboard through mailbox.

6. apb3\_slv\_agent\_c

Description: Slave agent class.

s\_drv : Type => object for slave driver class

s\_gen : Type => object for slave generator class

s\_mon : Type => object for slave monitor class

slv\_gen\_ended : Type => event

gen2driv : Type => mailbox generator to driver

mon2scb : Type => mailbox monitor to scoreboard

apb3\_ms\_intf : Type => virtual interface.

Method name : new

Parameters passed : virtual interface

Returned parameters : None

Description : constructing memory .

Method name : pre\_build

Parameters passed : none

Returned parameters : None

Description :

Method name : build

Parameters passed : none

Returned parameters : None

Description : calling run task of driver, generator and monitor.

Method name : post\_build

Parameters passed : none

Returned parameters : None

Description :

Method name : apb3\_slv\_agnt\_run\_t

Parameters passed : none

Returned parameters : None

Description : calling pre\_build , build, post\_build.

MASTER VIP:-

1. apb3\_m\_txn\_c

Description: Master Transaction Class

penable, pselx, pwrite = Type:- Single bit

paddr,pwdata, prdata= Type:- 32 bit.

Method name : txn\_do\_copy

Parameters passed : None

Returned parameters : None

Description : Deep copy.

2. apb3\_mst\_gen\_c

Description: Master Generator Class

trans :-object of transaction class.

gen2driv = Type : mailbox

ended = Type : event

repeat\_count = Type:int

Method name : new

Parameters passed : mailbox, event

Returned parameters : None

Description : assigning memory to mailbox and event.

Method name : apb3\_mst\_gen\_run\_t

Parameters passed : none

Returned parameters : None

Description : randomization ,putting values in mailbox and after complete repeatation trigger the event

3. apb3\_mst\_drv\_c

Description: Master Driver class.

no\_transaction => Type :- int

Description : to check number of transaction

mem\_vif => Type :- virtual

Description: virtual interface

gen2driv => Type:- mailbox

Method name : new

Parameters passed : virtual interface and mailbox

Returned parameters : None

Description : Constructing memory for virtual interface and mailbox

Method name : apb3\_mst\_drv\_reset\_t

Parameters passed : None

Returned parameters : None

Description : to reset all the variables

Method name : apb3\_mst\_drv\_driver\_t

Parameters passed : None

Returned parameters : None

Description : read write functionality

Method name : apb3\_mst\_drv\_run\_t

Parameters passed : None

Returned parameters : None

Description : Drive the signals.

4. apb3\_m\_intf

Description: Master Interface

clocking driver

clocking monitor

modport : DRIVER

ports : clocking driver, clock, reset

modport : MONITOR

ports : clocking monitor, clock, reset

5. apb3\_mst\_mon\_c

mem\_vif => Type :- virtual

Description: virtual interface

mon2scb => Type :- mailbox

Method name : new

Parameters passed : virtual interface and mailbox

Returned parameters : None

Description : Constructing memory for virtual interface and mailbox

Method name : apb3\_mst\_mon\_run\_t

Parameters passed : none

Returned parameters : None

Description : monitors data and transfer to scoreboard through mailbox.

6. apb3\_mst\_agent\_c

Description: Master agent class.

mst\_drv : Type => object for master driver class

mst\_gen : Type => object for master generator class

mst\_mon : Type => object for master monitor class

mst\_gen\_ended : Type => event

gen2driv : Type => mailbox generator to driver

mon2scb : Type => mailbox monitor to scoreboard

mem\_vif : Type => virtual interface.

Method name : new

Parameters passed : virtual interface

Returned parameters : None

Description : constructing memory .

Method name : pre\_build

Parameters passed : none

Returned parameters : None

Description :

Method name : build

Parameters passed : none

Returned parameters : None

Description : calling run task of driver, generator and monitor.

Method name : post\_build

Parameters passed : none

Returned parameters : None

Description :

Method name : apb3\_mst\_agnt\_run\_t

Parameters passed : none

Returned parameters : None

Description : calling pre\_build , build, post\_build.

7. apb3\_ms\_sb\_c

mon2sb => Type : mailbox

Method name : new

Parameters passed : mailbox

Returned parameters : None

Description : Constructing memory for mailbox and memory initialization

Method name : apb3\_ms\_sb\_run\_t

Parameters passed : none

Returned parameters : None

Description : compare actual and expected data.

8. apb3\_ms\_env\_c

mst\_agnt => Type : object of master agent class

slv\_agent => Type : object of slave agent class

mst\_vif => Type : master virtual interface

slv\_vif => Type : slave virtual interface

Method name : new

Parameters passed : master virtual interface, slave virtual interface

Returned parameters : None

Description : Constructing memory for master and slave virtual interface and master and slave agent class object

Method name : apb3\_ms\_env\_pre\_test\_t

Parameters passed : none

Returned parameters : None

Description : for initial check.

Method name : apb3\_ms\_env\_test\_t

Parameters passed : none

Returned parameters : None

Description :

Method name : apb3\_ms\_env\_post\_test\_t

Parameters passed : none

Returned parameters : None

Description :

Method name : apb3\_ms\_env\_run\_t

Parameters passed : none

Returned parameters : None

Description :

9. apb3\_ms\_top\_c

clk => Type : bit

intf => Type : interface instance

parameter passed : clk

# Verification setup

This section shall provide details of about the verification environment requirements. It describes the requirements on tools and their versions, the directory structure used, steps to set up the project independently, and how to execute the tests.

## Tool Versions

Table to list details of all tools required/used

* Tool name
* Source
* Tool Revision Used
* Comment

## VIP Versions

Table to list details of all VIPs required/used

* VIP name
* Source
* VIP Revision Used
* Comment

## Directory Structure

For directory structure along with file details

Also mention details about files/directories created after compiling/running tests

## Project Setup

The setup step for the above tool configuration will be as follows.

The following are the steps required to setup the verification environment.

* Set Environment Variables
* Source various files to setup tools

## Simulation Script Usage

Simulation will be run in simulation run directory already setup. Details of the simulation run script can be obtained with the help option of the run script. The simulation script is available at $PRJ\_ROOT/<path\_to\_simulation\_script>.

### Running Single Testcase

Provide use model and details of each switch used while sourcing simulation script.

### Running Regressions

The regression command for a specific test list file.

Provide use model and details of each switch used while sourcing simulation script.

### Compilation define and Simulation Argument Switch

Provide details of each compilation define and simulation argument

# Coverage & Regression

This section identifies different strategies for running regression, regression analysis and coverage collection and closure.

## Running Regression

Following strategy will be used for regression ……………

## Code coverage

## Assertion coverage

## Functional coverage

### Directed-Random Testing

#### Objectives of Directed Random Testing

#### Why Functional Coverage?

### How do we decided functional coverage bins / What to randomize ?

#### When / Where /How to sample data to be covered?

### Regression with Functional Coverage

#### To achieve functional coverage

### Directed-Random Testing : Self Checking

#### How to check correctness of test with directed-random testing

# Assumptions and Limitation

# Final Release checklist

# Appendix A: References

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Document | Version | Remarks |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |

Table 10‑4 References

## Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version No. | Date | Owner | Changes |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table 10‑5 Revision History

## Items Out Of Scope of Verification