Verification Using Cocotb

Test Plan v1
sumofN

Nishanth N

06/07/2021

Table of Contents

1. Introduction

1.1 Purpose

1.2 DUT Overview

2.Test Strategy

2.1 Test Objectives

2.2 Test Approach

3.Test Scope

3.1 In - Scope

3.2 Out of Scope

4.Testing Features

4.1 Datapath Test

4.2 ConfigurationTest

4.3 Input Data Test

5.Testing Environment

1. Introduction

1.1. Purpose

The main purpose of the test plan document is to maintain a record of test verification methods required for the task, to plan the appropriate approach for the test involving various aspects of cocotb, and to frame a timeline of the events during the test duration.

1.2 DUT Overview

The Test involves a DUT which takes in N 8-bit inputs and gives out the sum of N 8-bit inputs. The DUT has three interfaces :

- Input Data Interface
- Output Data Interface
- Configuration Interface

The Main purpose of the DUT is to sum all of the N 8-bit input.

2. Test Strategy

2.1 Test Objectives

The objective of the test is to verify the functionality of the RTL model that it works according to the given design specifications.

2.2 Test Approach

The testbench is written in python using cocotb library. The class ‘SumN_tb’ instantiates all drivers and monitors for the required transactions, along with the test models which provide the expected output with the given inputs from the drivers. The scoreboard takes in expected output and actual DUT output and compares the result. The @cocotb.test function is the main function that instantiates the class and implements the test. Testfactory generates tests with added options for configuration test and input data test.

3. Test Scope

3.1 In Scope

Verification of DUT ‘sumofN’ using Cocotb. The test will focus on datapath, configuration and input data entry.

3.2 Out Of Scope

Verification on sum overflow or control path. Specific test checks on timing restrictions, performance and speed of the DUT.

4. Testing Features

4.1 Datapath Test

Generic functional test of Dut. Checking correct flow of input and output data.

4.2 Configuration Test

Test on configuration address. Raise Interrupt for incorrect configuration of DUT.

4.3 Input Data Test

Put N-1 inputs and check the value of Sum.

5. Testing Environment

Required Software during the testing phase:

- Python v3.6
- GCC Compiler
- Cocotb Library
- Icarus Verilog Simulator
- GTKWave wave viewer
- Git version control system
- Linux OS
- VScode Editor