

# Verification plan template

## Signal layer

In this section an analysis to the DUV is done, list the most important features and properties of the DUV.

- 1)
- 2)
- 3)
- 4)

Definition of all DUV input/output ports:

Table 1: Input/Output ports list of the DUV

Signal	Direction	Length (bits)	Description

Intercommunication interface:

In this section it is specified how the verification framework and the DUV are interconnected. For this purpose it is normally used a glue logic such APB bus or Wishbone bus, but it depends of the DUV. Besides, all input and output pots of this interface should be specified as it was done with the DUV.

Table 2: Intercommunication interface input/output ports

Signal	Direction	Length (bits)	Description

The handshake protocol implemented by the intercommunication interface must be specified in order to define whether or not it is required an adapter and what consideration should be taken.

Table 3: High-level handshake protocol of the intercommunication interface

Order	Involved signals	Description

If it is determined that an adapter must be implemented, then the adapter design should be described in this section, taking into account the synchronization signals and data transmission. It is advisable to include a graphic containing the interconnection between the DUV and the verification framework using the adapter.

## Stimuli sequences

To determined how many stimuli sequences are required, it should be considered the following:

- Number of working modes, for a example a memory can receive commands for writing and reading, those are the two modes it implements.
- Number of corner cases which refer to cases where the DUV is more likely to fail, for example a multiplier can present some critical cases when a number is multiplied by zero or when two relatively large or small numbers multiplied since an error, an overflow or underflow can be triggered in those cases.

Once the number of stimuli sequences was determined, then those sequences should be described in detail to exercise the cases previously described.

Table 4: Stimuli sequences

Sequence code	Description

## Validation mechanism

In this section it should be described the golden model that is intended to obtain the expected results in every stimulation. This golden model can be written in any programming language compatible with SystemVerilog, such as C/C++. Also, the intercommunication tool to connect

this golden model with the verification framework should be described. Normally, the DPI is used for this purpose, but in some cases other options can be used such as a SystemVerilog model or a design previously validated. Moreover, the consideration of this golden model should be listed, which refer to some differences or disadvantages of its employment.

- 1)
- 2)
- 3)

## Coverage metrics

In this section all coverage is defined. The first step is to list all requirements of the DUV considering aspects such as synchronization, data validity, data stability, signal timing, sequences and data states consistency. All design requirement must have a code to identify the requirement, a description, an specific type and a weight to generate the grade once the simulation is complete.

Table 5: Requirement specifications of the DUV

Requirement code	Description	Validation type	Priority

As default, the code coverage metrics (Expression, block, toggle, fsm) must reach the percentages listed in Table 6 in order to fulfill industrial standards.

Table 6: Code coverage metrics

Type	Description	Expected ratio (%)
Expression	How many statements were covered	100
Block	How many blocks have been covered	95
Toggle	How many times signals and ports are toggled during simulation	95
FSM	Indicates whether the FSMs in the design reach all possible states or not	95