HDL MODELS

MODEL USAGE GUIDELINES

MAY 2014

Version 1.0

Highlights

- Defines Model Usage Guidelines with compiler directives

Revision History

Date	Document Version	Commen
15-May-2014	1.0	First release

This page is intentionally left blank.

Table of Contents

This page is intentionally left blank.

1. General Description

In model, certain compiler directives / switches are defined to enable the user to perform faster simulations in digital domain. These switches, however, do not reflect the real design behavior. The users should use them at their own discretion.

2. Known Problems and Solutions (KPS)

2.1 Verilog Model Usage

2.1.1 Debug Level Usage

The display of messages is controlled by three methods. This is applicable to every compiled verilog module.

This can be achieved in three ways –

2.1.1.1 Message Control through compiler directive

The display of messages is controlled by providing a compiler directive from the command line during compilation:

+define+ST_NO_MSG_MODE to get no messages from the model.
+define+ST_ERROR_ONLY_MODE to get only ERROR messages from the model
+define+ST_ALL_MSG_MODE to get all messages i.e. ERROR + WARNING + INFO from the model.
By default, the model is set as to display ERROR + WARNING messages from the model.

2.1.1.2 Message Control by overriding a parameter (defparam)

The display of messages is controlled by overriding a parameter p_debug_level in the test bench to values 0, 1, 2, and 3 using the following method:

defparam <instance_name>.p_debug_level = <value>

2.1.1.3 Message Control by updating a model internal register

The display of messages can also be controlled by updating a model internal register to a certain value from the test environment:

<instance_name>.debug_level = <value>
The value to be overridden has the following definitions
2'b00: to get no messages from the model.
2'b01: to get only ERROR messages from the model.
2'b10: to get only ERROR + WARNING messages from the model.
2'b11: to get all messages (ERROR + WARNING +INFO) from the model.
By default, the debug_level in the model is set as 2'b10.

2.1.2 Message Display Control

There is another method for the controlling the display of messages, which is achieved by overriding a parameter available in the model, the same can be achieved by adapting any of the two methods mentioned below :

Compiler Directive

+define+ST_MSG_CONTROL_TIME=<desired_value in ns>

defparam

defparam <instance_name>.message_control_time = <desired_value in ns>

It is to be understood that this features 'only' enables the display of messages from the desired time, whereas the output behavior remains intact.

2.1.3 Timescale

The time scale used in the model is 1 ns / 1 ps. The timescale of the model can be changed by providing a compiler directive from the command line during compilation in the following way

+define+ST_TIMESCALE='\'timescale <time_scale>/<precision>' for single invoke commands
-define ST_TIMESCALE=\'timescale\ <time_scale>\/<precision> for multi command invoke

The model is qualified on 1ns / 1ps, if the time scale is changed the model may behave incorrectly.
This provision is given exclusively to improve the simulation time and doesn't reflect the design behavior. The user needs to use this feature with extreme caution.

2.1.4 Delay Mode Control

The model is developed and validated on **'delay_mode_path,** where as in the model there is a provision to make the model work without a delay mode. The same can be achieved by providing a compiler directive from the command line during simulation:

+define+ST_NODELAYMODE

The model is qualified on 'delay_mode_path. The user needs to use this feature with extreme caution

3. Contact Information

For more information about this document, please write to **HELPDESK.**