|  |
| --- |
| NetSpeed Orion AMBA Lint Waivers  Version: ORION-AMBA-15.08  August 15, 2015 |

NetSpeed Orion AMBA Lint Waivers

About This Document

This document describes Lint waivers for NetSpeed Orion.

Audience

This document is intended for users of NocStudio:

* NoC Designers
* NoC Verification Engineers
* SoC Designers
* SoC Verification Engineers

Prerequisite

Before proceeding, you should generally understand:

* Basics of Lint

Related Documents

The following documents can be used as a reference to this document.

* NetSpeed NocStudio Orion AMBA User Manual
* NetSpeed Orion AMBA Physical Design Guidelines
* NetSpeed Orion IP Integration Spec

Customer Support

For technical support about this product, please contact [support@netspeedsystems.com](mailto:support@netspeedsystems.com)

For general information about NetSpeed products refer to: [www.netspeedsystems.com](http://www.netspeedsystems.com)

**Contents**

[About This Document 2](#_Toc417381118)

[1 Waivers for Cadence HAL RTL lint check 4](#_Toc417381119)

# Waivers for Cadence HAL RTL lint check

Following rules are waived based on review of occurrences in the design.

**CONSTC**: "Constant conditional expression" There are instance in the code where a parameter may be directly used in a logic expression. Also there are several  localparam definitions using expressions based on other parameters.

**MEMSIZ**: "Memory declaration for '%s' defines a single bit memory word". Some parameterized vector arrays may become array of 1-bit vectors based on the parameter value in that configuration

**SHFTNC**: "Shift by non-constant" As a coding practice we allow use of non-constant shifts to represent multiplexers

**CLKUCL**: "The clock '%s' drives a combinational logic"There are clock gating modules built into the design. These are allowed to violate this rule

**UNCONO**: "Port '%s' (which is being used as an output) of entity/module ' %s' is being driven inside the design but not connected (either partially or completely) in its instance '%s'"Design modules are parameterized. In some configurations of these parameterized blocks, some output ports may be left unconnected when there is no functional path downstream of that port. These are usually driven by constants in the module.

**BSINTT**: "Bit/part select of integer or time variable '%s' encountered" Range select of integer variable used as loop index is allowed in our designs.

**IPRTEX**: "Integer is used in port expression" Subset configurations may be derived from parameterized template designs by driving unused inputs with constants and leaving used outputs unloaded.

**URDWIR**: "Wire '%s' defined in module '%s' does not drive any object but is assigned at least once" Some internal signals are defined as tap points for functional checkers. Some internal logic clouds may also be unused when certain parameter values remove logic using them. Synthesis tools are allowed to optimize these.

**USEPRT**: "Input port '%s' defined in %s '%s' is unused" Some modules have multiple clock ports defined to support ASYNC mode of operation. When configured by parameters to be in SYNC mode, these clock ports will remain unused.

**URAWIR**: Wire '%s' defined in module '%s' is unused (neither read nor assigned)" Some software generated files may contain unused wires

**URDREG**: "Local register variable '%s' is not read but assigned at least once in the module '%s'" Due to parameterized code for fine grained logic optimization. Some registers may remain unused when certain parameter values remove or disable logic using them. Synthesis tools are allowed to optimize these.

**VERREP**: "Repeated usage of identifier or label name '%s'" local scope loop variables and genvars are allowed to be reused

**USEPAR**: "Parameter '%s' is unused" Parameters to be used by certain code sections may remain unused if those code sections are removed/disabled due to parmeters

**TIELOG**: "The output/inout '%s' is assigned a constant logic value" For logic optimization, an output port is driven with a constant value when it is unused based on parameter configurations

**FTNNAS**: "Last statement in function '%s' in module '%s' does not assign to the function" Default value assignment before computaion inside the fucntion

**TPOUNR**: "Output '%s' of top level module is not a register" Allowed by our design practice

**FDTHRU**: "Feedthrough detected from input '%s' to output '%s'" In certain parameterized configuration of blocks there may be an input to output feedthrough

**POOBID**: "Variable index/range selection of '%s' is potentially outside of the defined range" Based on values of parameters, non power-of-two arrays may exist in the design. When indexed with pointers, this warning may occur. However fucntionally it is ensured that the pointers do not take values outside the defined range of the array

2670 Seely Ave

Building 11

San Jose, CA 95134

(408) 914-6962

<http://www.netspeedsystems.com>