|  |
| --- |
| NetSpeed Gemini CDC Waivers  Version: GEMINI-16.04  April 15, 2016 |

NetSpeed Gemini AMBA CDC Waivers

About This Document

This document describes CDC waivers for NetSpeed Gemini.

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 Clock Domain Crossings

Related Documents

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

* NetSpeed NocStudio Gemini User Manual
* NetSpeed Gemini Physical Design Guidelines
* NetSpeed Gemini 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](#_Toc427492531)

[Audience 2](#_Toc427492532)

[Prerequisite 2](#_Toc427492533)

[Related Documents 2](#_Toc427492534)

[Customer Support 2](#_Toc427492535)

[1 Waivers for Cadence CDC Tool 4](#_Toc427492536)

[1.1 Tool Details 4](#_Toc427492537)

[1.2 False CDC Violations – Design Intent 4](#_Toc427492538)

[1.2.1 False Violation #1 4](#_Toc427492539)

[1.2.2 False Violation #2 4](#_Toc427492540)

[1.2.3 False Violation #3 5](#_Toc427492541)

[1.2.4 False Violation #4 5](#_Toc427492542)

[1.2.5 False Violation #5 6](#_Toc427492543)

[1.3 False CDC Violations – Tool Issues 6](#_Toc427492544)

[1.3.1 False Violation #1 6](#_Toc427492545)

[1.4 Known CDC Violations 7](#_Toc427492546)

[1.4.1 Violation #1 7](#_Toc427492547)

[1.4.2 Violation #2 7](#_Toc427492548)

[1.4.3 Violation #3 7](#_Toc427492549)

[1.4.4 Violation #4 7](#_Toc427492550)

# Waivers for Cadence CDC Tool

## Tool Details

* Tool: Cadence Conformal Constraint Designer CDC
* Version: Ver15.10-D182

## False CDC Violations – Design Intent

### False Violation #1

The tool reports false CDC violations on following clock cross modules, when *fast\_clk* and *slow\_clk* signals are specified in different clock domains by NetSpeed auto-generated SDC files. The leaf level module is ns\_clkcross\_buffer, instanced in all four of these modules where the false CDC violations are reported

* ns\_clkcross\_fast\_to\_slow
* ns\_clkcross\_slow\_to\_fast
* ns\_clkcross\_rwack\_slow\_to\_fast.v
* ns\_clkcross\_rwack\_fast\_to\_slow.v

Resolution**:** *fast\_clk* and *slow\_clk* clocks need to be specified to be in the same clock domain manually in the sdc constraint file**.** The clock cross modules are phase-aligned synchronous clock crossers with an N:1 or 1:N ratio

Work around**:** The customer can post-edit the auto-generated SDC files to specify *fast\_clk* and *slow\_clk* clocks in the same clock domain.

Example:

set clock\_clk\_A\_sync\_group [get\_clocks [list fast\_clk slow\_clk ]]

set\_clock\_groups -name async\_clock\_group -asynchronous \

-group clock\_clk\_A\_sync\_group \

–group [get\_clocks clk\_C]

### False Violation #2

There are CDC violations on input signals to ns\_pwrsup.

* sleep\_ack\_asserted\_n
* sleep\_ack\_deasserted
* fence\_ack\_asserted\_n
* fence\_ack\_deasserted\_n
* wake\_req\_asserted

Resolution**:** These CDC violations should be waived. These signals by protocol definition ensure that the convergence issue is not seen and these should be waived. These are defined as level signals which transition to same value over a period of time. There is no chance of stray glitches being captured as valid signal transition on these signals. Please refer to NetSpeed NocStudio Low Power User Manual for further explanation.

Work around**:** Not applicable

### False Violation #3

The tool reports CDC violation for reset paths from ns\_aceslvbrdg, for asynchronous interface with the NoC, with following end point destination registers.

from\_instance : u\_ns\_fabric/\*/u\_ns\_axiliteslvbrdg\_\*/u\_ns\_axiliteslvbrdg/u\_axislv\_inst/u\_ns\_aceslvbrdg/\*\_clk\_rstsync/u\_ns\_demet/demet\_stage\_q\_reg\*

to\_instance : u\_ns\_fabric/\*/u\_ns\_c\_ppln\_\_\*/u\_ns\_noc\_credit\_ppln/G\_CREDIT\_PPLN[1].u\_ns\_credit\_ppln\_stage/credit\_q\_reg\*

Resolution**:** This violation should be waived. Reset is synchronized in *noc\_clk\_rstsync/* *pdr\_noc\_clk\_rstsync* using *noc\_clk* and further synchronized in   
*asclk\_rstsync* using *rx\_fifo\_async\_clk.* So the async path is only on reset assertion. The   
reset synchronizer is synchronizing this path anyway, so this is not a concern.

### False Violation #4

The tool reports CDC violation for reset paths from ns\_aceslvbrdg, for asynchronous interface with the NoC, with the following end point destination registers.

from\_instance :  
u\_ns\_fabric/\*/u\_ns\_axiliteslvbrdg\_\*/u\_ns\_axiliteslvbrdg/u\_axislv\_inst/u\_ns\_aceslvbrdg/\*\_clk\_rstsync/u\_ns\_demet/demet\_stage\_q\_reg\*  
  
to\_instance :  
u\_ns\_fabric/\*/u\_ns\_axiliteslvbrdg\_largehost\_\*/u\_ns\_axiliteslvbrdg/u\_axislv\_inst/u\_ns\_aceslvbrdg/u\_ns\_strrxswitch/ns\_strrxbrdg\_layeriflogic0/strrxbrdg\_lay[\*].strrxfifologicvc\*.IVCFIFO\_ASYNC\_\*.vc\*\_fifo\_async/\*

Resolution**:** This violation should be waived. Reset is synchronized in   
*noc\_clk\_rstsync/* *pdr\_noc\_clk\_rstsync* using *noc\_clk* and further synchronized in   
*asclk\_rstsync* using *rx\_fifo\_async\_clk.* So the async path is only on reset assertion. The   
reset synchronizer is synchronizing this path anyway, so this is not a concern.

### False Violation #5

The tool reports CDC violation for reset paths from ns\_acemstrbrdg\_core, for asynchronous interface with the NoC, with various end points destination registers.

from\_instance : u\_ns\_fabric/\*/\*/u\_ns\_acemstrbrdg\_core/noc\_clk\_rstsync/u\_ns\_demet/demet\_stage\_q\_reg\*  
  
Resolution**:** This violation should be waived. Reset is synchronized in   
*noc\_clk\_rstsync/* *pdr\_noc\_clk\_rstsync* using *noc\_clk* and further synchronized in   
*asclk\_rstsync* using *rx\_fifo\_async\_clk.* So the async path is only on reset assertion. The   
reset synchronizer is synchronizing this path anyway, so this is not a concern.

## False CDC Violations – Tool Issues

### False Violation #1

False violation paths due to tool not being able to identify asynchronous FIFOs across two clock domains

from\_instance : u\_ns\_fabric/\*/u\_ns\_router\_\*/u\_ns\_router/G\_\*\_INBLK\_ENB.u\_\*\_inblk/G\_IVCBUF\_IVCCTRL[\*].G\_IVC\_ENB.G\_IVCBUF\_ASYNC.u\_ivcbuf\_async/reg\_array\_reg\*

Resolution: CDC violations from above starting path should be waived. The CDC tool fails to identify *ns\_ivcbuf\_1rp\_a\_c* module inside ns\_router as asynchronous FIFO. The *to\_instance* of these paths are various destination registers in the design.

from\_instance : u\_ns\_fabric/\*/\*/\*/\*/\*/u\_ns\_strrxswitch/ns\_strrxbrdg\_layeriflogic0/strrxbrdg\_lay[\*].strrxfifologicvc\*.IVCFIFO\_ASYNC\_\*.vc\*\_fifo\_async/reg\_array\_reg\*  
  
Resolution: CDC violations from above starting path should be waived. The CDC tool fails to identify *ns\_ivcbuf\_1rp\_a\_c* module inside ns\_strrxswitch as asynchronous FIFO. The *to\_instance* of these paths are various destination registers in the design.

## Known CDC Violations

### Violation #1

ns\_aceslvbrdg module does not have a synchronizer on input *noc\_acsb\_cg\_busy* signal, when it has an ASYNC interface with the NoC layer. This is applicable  
when P\_NOC\_RX\_ASYNC\_MODE for the layer is set to 1

**Resolution:** To be fixed in future release

**Work around:** To bypass this CDC violation, coarse grained clock gating feature for the design has to be statically turned off. This is done by specifying **“mesh\_prop coarse\_clock\_gating\_enabled no”** in the NS config file

### Violation #2

ns\_acemstrbrdg\_core module does not have a synchronizer on input *noc\_acmb\_cg\_busy* signal, when it has an ASYNC interface with the NoC layer. This is applicable when P\_NOC\_RX\_ASYNC\_MODE for the layer is set to 1

**Resolution:** To be fixed in future release

**Work around:** To bypass this CDC violation, coarse grained clock gating feature for the design has to be statically turned off. This is done by specifying **“mesh\_prop coarse\_clock\_gating\_enabled no”** in the NS config file

### Violation #3

ns\_strrxbrdg\_layeriflogic module has incorrect clock connections in synchronizer instances vc0\_full\_sync, vc1\_full\_sync, vc2\_full\_sync, vc3\_full\_sync. The “*clk*” pin of these instances should be connected to *fifo\_clk* signal instead of *fifo\_async\_clk.* This CDC violation shows up when P\_NOC\_RX\_ASYNC\_MODE for the layer is set to 1, ie ns\_strrxswitch has ASYNC interface with the NoC layer.

**Resolution:** To be fixed in future release

**Work around:** Set the router clock domain the same as their attached bridge, where this violaton is taking place.

### Violation #4

ns\_rbm\_tunnel module does not have synchronizer on its input signal *pd\_active\_bridge*. The CDC tool will report violation when low power mode is enabled.

**Resolution:** To be fixed in future release

**Work around**: To bypass this CDC violation, disable Low power mode in NocStudio

2670 Seely Ave

Building 11

San Jose, CA 95134

(408) 914-6962

<http://www.netspeedsystems.com>