NetSpeed Orion

Release Notes

Version: ORION-17.04

May 25, 2017

NetSpeed Orion 17.04 Release Notes

About This Document

This document lists the release notes for NetSpeed Orion. Using NetSpeed NocStudio, users can define NoC architectures, describe specifications and requirements, optimize the NoC design and finally generate the NoC IP files such as RTL, testbench, synthesis scripts, NoC IP documentation etc.

Audience

This document is intended for users of NocStudio:

- NoC Designers
- NoC Architects
- SoC Architects

Prerequisite

Before proceeding, you should generally understand:

- Basics of NetSpeed Orion IP Technology

Related Documents

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

- NetSpeed NocStudio User Manual

Customer Support

For technical support about this product, please contact support@netspeedsystems.com

For general information about NetSpeed products refer to: www.netspeedsystems.com

NETSPEED SYSTEMS

Contents

1 Deliverables

- NetSpeed NocStudio Package and one of the license options:
  - N7 version supporting 16 layers and 256 bridges
  - N5 version supporting 4 layers and 60 bridges
  - N3 version supporting 1 layers and 12 bridges
- NocStudio executable with interactive GUI.
- Verification checkers to be used in the DV environment.
- Sanity Test Bench.
- Documentation
  a. NocStudio User Manual: The User Guide describes how to set up a system using NocStudio and how to use it to generate NetSpeed IP.
  b. IP Integration Spec: The Integration Manual describes how to integrate a configured network into a larger subsystem.
  c. Technical Reference Manual: The Technical Reference Manual describes how the functionality of the various NoC elements, the features and functions available, and how to dynamically change the functions using the programmer's mode.

2 Installation

- NocStudio uses FlexLM based licensing.
  - Linux CentOS 5.5 or higher
  - For node-locked license file, copy over the license file under NocStudio installation directory and renamed it as “license.dat”. If the license file resides in a separated folder, please set environment variable LM_LICENSE_FILE with the proper path.
  - For floating licensing scheme, please download and extract netspeed.flexlmpkg.tar.gz for 32- or 64-bit license daemon and follow FlexLM documentation.

  NOTE: When untarring Linux files, ensure it is done on a Linux machine. Untarring Linux files on a Windows machine causes problems with symbolic links.

- The release makes use of Qt libraries covered under LGPL:
  - http://qt-project.org/downloads

3 Feature Updates: System Interconnect

3.1 IEEE Std 1364™-2005 format

Interconnect RTL format has been updated to be compliant with IEEE Std 1364™-2005 format.

3.2 TCL Interface support

NetSpeed adds support in NocStudio to use TCL interpreter to to process input. All NocStudio commands will be registered in the *netspeed::namespace*, and all commands that don't overlap builtin commands will be registered without namespace. The text output of NocStudio commands will be available as the result of running them in TCL mode. When the TCL command being run returns a value, that value will be printed in the console. Within TCL mode, $() and $[] interpolation is not available, and defcmd-defined commands will not be registered in the TCL interpreter. NocStudio commands that use '[', ']', in their syntax will need special escaping to be executed in TCL mode, while '{' and '}' will normally be interpreted properly without escaping. TCL's stdout is not yet redirected into the NocStudio console, so this output will appear in the terminal that NocStudio runs in.

Examples:

enter_tcl_mode

info tclversion

Prints the version of TCL built into NocStudio

enter_tcl_mode

source json.tcl; [dict keys [dict get [json::json2dict [json_dump] hosts]

Prints the list of hosts reported from json_dump (requires json.tcl library)

3.3 Enhanced Performance counter support

Enhanced configurable performance counters have been added for

- Counting cycles on an RX host interface where the interface is out of parity
- Counting cycles where NoC VCs have valid flits for an RX host interface
- Cycles with credit stalls (valid request but no credit) on output VCs of router output port

In Addition, destination bridge ID, interface ID and QoS fields are logged when a route looking failure occurs on TX host interface.

4 Feature Updates: Non-Coherent Components

4.1 Functional Safety - Interface Parity support

NetSpeed enhances its Functional Safety offering by adding interface parity support its host interfaces. With this, the host IP can generate and detect parity, and thus extends the reliability features further into the SoC.

- On control interfaces, single bit parity for all the fields, or byte parity for every byte group of signals on the interface is supported.
- On data channels parity for each byte of data is available.

Transmitting end generates parity for every beat on the interface, and receiving end checks parity and logs any error.

Limitations: For more details and limitations, please refer to Chapter "Safety and Reliability" in the TRM.

4.2 Asymmetric RD/WR widths

Traditionally the read and writes data widths match, but based on customer and industry trends, there is a need to support interfaces that require different data widths between the read and write channels. NetSpeeds adds this feature that allows a master to specify independent AXI_DATA_WIDTHS widths of the read (RDATA) and write (WDATA) paths.

4.3 REGFILE support

In order to support various design methodologies and techniques, NetSpeed has added the configurable option to clearly separate register-file structures so that the user can provide a custom implementation. Additionally, NocStudio reporting has been updated to breakout the register-file storage bits into a separate reporting line.

For more details, please refer to Integration Specification.

Limitations:

- ECC support is not available

4.4 Protection against Slave dependency behavior

NetSpeed has added support for specific slaves that exhibit a dependency between their read and write responses. Specifically, some slaves send write reponses in the middle of a multibeat

read data response, with the expectation that write response cannot be blocked, and pauses sending the read data until then. This introduces dependency between the channels, and NetSpeed has added support to automatically size the write responses FIFOs to avoid this. This is done by adding a configurable property to the slave bridge, which the user can enabled based on their slave behavior.

4.5 REORDER BRIDGE SUPPORT (*beta FEATURE*)

The reorder bridge acts as a proxy for a collection of masters to allow sharing of the reorder buffer resource across those masters. In systems where the traffic flows are very bursty, this can result in a lower area cost compared to each master bridge having a dedicated reorder buffer.

Limitations:

- Gemini Coherency support cannot be used in the same NoC as a reorder bridge
- Pegasus last-level cache cannot be used in the same NoC as a reorder bridge
- Netspeed Power-management fencing/draining cannot be used for components (routers, bridges) that are accessed through a reorder bridge
- All *add_range* for slaves that go through ROB must have full permission (rd/write, secure/nonsecure (base[5:0] and mask[5:0] fully enabled)).

5 EDA Tool Compatibility

- Cadence EDA tools were used for verification and synthesis of this product.
  - Incisive RTL Simulator 15.22.012
  - Genus RTL Synthesis 15.20-p004_1
  - HAL Linting tool 13.20.036
  - Confirmal 15.10.120
- Compatibility testing has been done with VCS J-2014.12-SP3-3. Please refer to IP Integration specification to enable/disable specific NetSpeed checker in order to resolve or workaround any verification related issues, if any. Contact your NetSpeed or Synopsys support team for assistance.

6 Errata: System Interconnect

None

7 Errata: Non-Coherent Components

7.1 AHB

There could be a deadlock between AHB master bridge and the AHB master if master is waiting for HREADY to be asserted before removing the BUSY command. Workaround: remove BUSY as soon as the next command is available.

7.2 Priority Address Map

The Priority Address Map has a potential issue when some agents do not have access to a slave in a foreground range. Instead of getting a decode error when they attempt to access those ranges, they can hit against the background range and send the request to that slave.

8 Changes to Commands and Properties

8.1 Command Changes

Command Name	Comment
set_ivc_ovc_mapping	New command to set the mapping between an input VC at a router port and an output port on the same router.
reset_ivc_ovc_mapping	New command to reset the mapping between an input VC at a router port and an output port
list_ivc_ovc_mapping	New command to to list the input VC to output VC mapping on a router.
assert_group_clocks	New command to check that the specified rtl group has no clocks other than the listed clocks
list_curves	New command to list the points of a curve/probability mass function or show all curves/probability mass functions
set_curve	New command to set the piecewise linear curve associated with a name.
reset_map	New command to reset the mapping of traffic.
enter_tcl_mode	New command to enable use of the TCL interpreter to process further input
exit_tcl_mode	New command to return to processing NCF input format from TCL mode
ml_build	New command to build NoCs using machine learning
add_reorder_block	New command to add a reorder block to the NoC

8.2 Default Property Changes

Property Name	**Default Value**	**Comment**
read_burstiness	1	This property has been replaced by new property burstiness

burstiness	1	This property is meant to replace read_burstiness
noc_injection_queue_depth	0	This property has been replaced by niq_depth
niq_depth	0	This property is meant to replace noc_injection_queue_depth
noc_ejection_queue_depth	0	This property has been replaced by neq_depth
neq_depth	0	This property is meant to replace noc_ejection_queue_depth
host_processing_queue_depth	32	This property has been renamed to hpq_depth
hpq_depth	32	This property has replaced host_processing_queue_depth
trace_enable	no	This property has been deprecated
axi4_ac_parity_enb	no	New property to enable parity on all AC channels
axi4_acaddr_parity_enb	no	New property to enable parity for addresses on all AC channels
axi4_ar_parity_enb	no	New property to enable parity on all AR channels
axi4_araddr_parity_enb	no	New property to enable parity for addresses on all AR channels
axi4_aw_parity_enb	no	New property to enable parity on all AW channels
axi4_awaddr_parity_enb	no	New property to enable parity for addresses on all AW channels
axi4_bresp_parity_enb	no	New property to enable parity on all B response channels
axi4_cd_parity_enb	no	New property to enable parity for data on all CD channels
axi4_r_parity_enb	no	New property to enable parity for data on all R channels
axi4_w_parity_enb	no	New property to enable parity for data on all W channels
axi4_rresp_parity_enb	no	New property to enable parity on all R response channels
ppln_in_node_id_pd	no	New property to enable the use of node PDs for pipeline stages
prefer_shortest_path_routes	yes	New property to prefer shortest path routes with more turns over longer routes with fewer turns in route computation between 2 points

axi4s_drain_b_response	yes	New property that indicates whether axi4s devices have preallocated space for B response packets to drain into the bridge
axi4s_r_interleave	no	New property that indicates whether axi4s devices can send interleaved read data.
axi4_allow_different_data_widths	no	New property to allow read response and write request channels to have different data widths.
axi4m_ar_rob_ram_enable	no	New propety to enable use of RAMs instead of flops for the read reorder buffer
axi4m_ar_rob_ram_in_width	0	New property to set the number of bits of input for read reorder buffer RAMs.
axi4m_ar_rob_ram_out_width	0	New property to set the number of bits of output for read reorder buffer RAMs.

8.3 MESH PROPERTY CHANGES

Property Name	Default Value	Comment
intf_parity_addr_per_byte	yes	New property to choose between per byte or per word parity for addresses in the NoC
intf_parity_per_byte	no	New property to choose between per byte or per word parity in the NoC

8.4 BRIDGE PROPERTY CHANGES

Property Name	Comment
axi4m_ar_rob_ram_enable	New propety to enable use of RAMs instead of flops for the read reorder buffer
axi4m_ar_rob_ram_in_width	New property to set the number of bits of input for read reorder buffer RAMs.
axi4m_ar_rob_ram_out_width	New property to set the number of bits of output for read reorder buffer RAMs.
axi4s_drain_b_response	New property that indicates whether axi4s devices have preallocated space for B response packets to drain into the bridge

axi4_allow_different_data_widths	New property to allow read response and write request channels to have different data widths.
axi4_ar_parity_enb	New property to enable parity on the AR channel
axi4_araddr_parity_enb	New property to enable parity for addresses on the AR channel
axi4_aw_parity_enb	New property to enable parity on the AW channel
axi4_awaddr_parity_enb	New property to enable parity for addresses on the AW channel
axi4_bresp_parity_enb	New property to enable parity on the B response channel
axi4_r_parity_enb	New property to enable parity for data on the R channel
axi4_w_parity_enb	New property to enable parity for data on the W channel
axi4_rresp_parity_enb	New property to enable parity on the R response channel

8.5 Host Property Changes

None

8.6 Interface Property Changes

Property Name	**Comment**
noc_injection_queue_depth	This property has been replaced by niq_depth
niq_depth	This property is meant to replace noc_injection_queue_depth
noc_ejection_queue_depth	This property has been replaced by neq_depth
neq_depth	This property is meant to replace noc_ejection_queue_depth
host_processing_queue_depth	This property has been replaced by hpq_depth
hpq_depth	This property is meant to replace host_processing_queue_depth
hpq_latency_curve	This property is used to automatically control host processing latency based on the specified curve

hpq_latency	This property is used to automatically control host processing latency based on the specified constant value.
hpq_rate_curve	This property is used to automatically control host processing completion rate.
hpq_rate	This property is used to automatically control host processing completion rate based on a specified constant value.
shared_hpq	This property is used to control the host processing queue that requests arriving at this interface should go to.
guaranteed_sink	This property forces the interface to sink every packet that it receives without any backpressue into the NoC
atid	This property specifies the ATB ID of the element
trace_fifo_depth	This property specifies the size of the trace capture buffer for this interface.
ext_timestamp_value	This property is used to enable use of TSVALUE interface to produce timestamps from time source external to bridge.
timestamp_width	This property is used to specify the bit width of timestamp values captured in trace streams.
trace_req_capture_mask	This property is used to specify the bit mask that controls which fields are captured for this interface.
trace_resp_capture_mask	This property is used to specify the bit mask that controls which fields are captured for this interface.
atb_fifo_depth	This property is used to specify the size of the ATB data buffer for this interface

8.7 Link Property Changes

None

8.8 Router Property Changes

None

8.9 VC Property Changes

None

2870 Zanker Road,

Suite 210,

San Jose, CA 95134

(408) 617-5209

http://www.netspeedsystems.com