NetSpeed Orion Streaming Bridge Microarchitecture Specification

Customer Document

NetSpeed Orion Streaming Bridge Microarchitecture Specification

About This Document

This document describes the micro-architecture specification of Streaming Bridge. This document includes feature set, block diagram, micro-architecture description, pinout and parameters used in Streaming Bridge design

Audience

This document is intended for users of NetSpeed's Orion IP:

- NoC Architects
- NoC Designers
- NoC Verification Engineers

Prerequisite

Before proceeding, you should generally understand:

- Basics of Network on Chip technology
- NetSpeed Streaming Interface Specification

Related Documents

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

- NocStudio User Manual
- NetSpeed Streaming Interface Specification

Customer Support

For technical support about this product, please contact your local NetSpeed sales office, representative, or distributor.

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

Contents

List of Figures

List of Tables

Acronyms

NoC	Network on Chip
SoC	System on Chip
Host	An IP core, component, or device sitting in an SoC
Hostport	A port of a host that connects to NoC router's injection and ejection port via a bridge to be able to inject traffic into NoC or eject traffic from NoC
Interface	Sets of signals to receive or transmit transaction messages of a hostport; a hostport may contain multiple interfaces; each interface may be uni-directional i.e. it sends or receives transaction messages or bidirectional i.e. it both receives and transmits transaction messages
Router	A hardware switch at the cross point of a mesh connecting to up to 4 of its neighboring routers and one or more bridges to connect to one or more hostports
Bridge	Sits between a router's port (often injection/ejection) and a hostport of a host to convert the hostports signal protocol (such as AMBA AXI-4) to NoC packet format and vice-versa and additional operations needed by the signaling protocol such as width conversion, etc.
NetSpeed Streaming Interface/Protocol	Signaling protocol provided by NetSpeed NoC streaming bridges; it is a simple credit based bidirectional interface for hosts to inject and eject messages into/from NoC
NetSpeed Streaming Bridge	Sits between a router's port (often injection/ejection) and a NetSpeed streaming hostport of a host to convert the hostports signal to NoC packet format and vice-versa and additional operations needed by the protocol such as width conversion, etc.; there are 4 physical bidirectional NetSpeed streaming interfaces available per bridge
Streaming Interfaces	Streaming bridge has four bidirectional interfaces named a, b, c and d

Link/Port	Physical channel between two routers or between a router and a bridge
Channel	Physical or virtual channel between two routers or between a router and a bridge
Virtual Channel (VC)	Virtual channel between two routers or between a router and a bridge
Injection channel	Incoming (with respect to NoC) virtual or physical channel at a router at which a hostport is connected to via a bridge and at which the router receives traffic from the hostport
Ejection channel	Outgoing (with respect to NoC) virtual or physical channel at a router at which a hostport is connected to via a bridge and at which the router sends traffic to the hostport
Cell	A node in a 2D grid or mesh; A NoC router is associated with every cell
Node	A router or cell of the 2D NoC grid
Virtual node	A boundary router or cell of the 2D NoC grid; in nxm mesh, there are n virtual nodes each at the top boundary and the bottom boundary in the mesh, and m at the right and the left boundary of the mesh
Virtual router	A router at a virtual node; virtual router is connected to an adjacent internal (non-boundary) router; there is no real hardware associated with a virtual router; the hostport bridge connected to a virtual router is directly connected to the internal router's port at which the virtual router is connected to the internal router
Multi-layer NoC	Multiple parallel mesh NoCs each forming a layer; routers in each layer operate independently of each other; two NoC layers have no connection between their routers; a bridge at a cell connects the injection/ejection port to a router in each layer and transmits each hostport transaction message to one of the layer's routers, and receives transaction messages from all layer's routers delivering them to the hostport
NoC layer	An independent NoC layer in a multi-layer NoC
Packetization	Encoding of hostport signals into NoC packet

	format before they are delivered to the NoC; bridges perform packetization of hostport transaction messages into NoC packets and de-packetization of NoC packets into hostport transaction messages
Beat	A single cycle of data part of a transaction message at an AXI-4 or streaming hostport interface
Flit	Part of a packet that is transmitted or received at a router's port in a single cycle
Packet	A transaction message packetized into NoC message; a packet may contain one or many flits
Transaction	A sequence of inter-dependent messages between various source and destination hosts/ports/interfaces.
Hostport id	The globally unique numerical id of a hostport bridge; this is assigned by NocStudio or can be assigned by user during hostport addition; injecting transaction messages must have an associated destination in form of hostport id
hostport name	Name of the hostport in form of hostname/portname
interface id	This is a, b, c or d if used in context of name in streaming bridge or ld, st, st_resp and ld_data in context of name of AXI bridge; In streaming, they corresponds to 0, 1, 2 or 3 values respectively; in streaming bridge, injecting transaction messages must have an associated destination in form of interface id
QoS id	4-bit QoS id; each transaction has a single QoS id
priority	2-bit priority of a transaction; there is a 1-1 mapping between QoS id and priority which can be set in NocStudio
weight	weight of a QoS id; this 8-bit value

1 Overview

Orion NoC supports NetSpeed streaming bridge, which provides a relatively simple interface to the hostports and packetizes data to be transferred into NoC packets.

On a host port Streaming bridge either interfaces with the TX host interface or the RX host interface. Streaming bridge that interfaces with TX host interface is called Streaming TX Bridge and the one that interfaces with RX host interface is called Streaming RX Bridge. Streaming TX Bridge can have up to 4 host interfaces on the ingress side and up to 8 layer interfaces on the egress side. Streaming RX Bridge can have up to 8 layer interfaces on the ingress side and up to 4 host interfaces on the egress side. Each layer can further have up to 4 VCs each. Streaming bridges support same or different data widths on every interface.

Shown below is the simplified block diagram of Orion NoC.

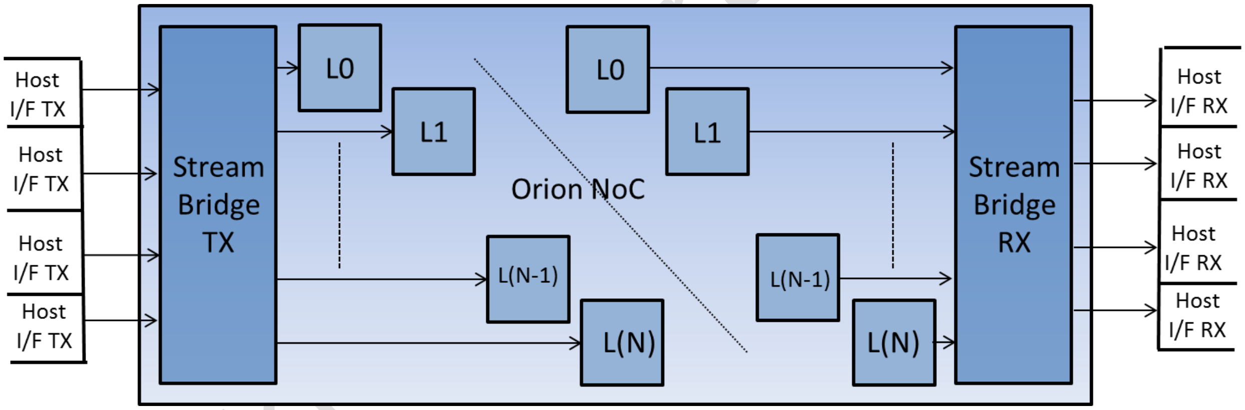

Figure 1: Orion NoC block diagram

The major components that are shown in the NoC are Streaming TX Bridge, Router layers and Streaming RX Bridge. Block also show the connectivity between Host port TX and RX interfaces.

2 Feature Summary

2.1 Streaming TX Bridge

A high level summary of features for the Streaming TX Bridge are provided in this section

- Supports up to 4 Host interfaces. Each interface implements
    - SOP, EOP, 4 QoS bits, Data beat and Data Valid, Target Interface ID, Target Host Port ID
    - Credit based flow control
    - Implements a power of 2 (e.g. 2, 4, 8) deep input FIFO on the Host side. Depth of the FIFO can be different for each host interface
- Connects to maximum of 8 layers with up to 4 VCs per layer.
    - Number of layer and number of VCs might not be power of 2. E.g. number of layers can be 1, 2, 3, 4, 5, 6, 7, 8 and VCs can be 1, 2, 3, 4.
    - Implements Credit based flow control.
- Implements fully parameterized Inputs/Outputs on Host and Router sides
- Implements 16 QoS profiles.
    - These are indexed by input QoS interface bus
    - Each Host interface accesses the same QoS profile table but with its own QoS value
    - Each QoS profile has 8-bit Weight field
    - QoS value with Interface ID and Host port ID are used to access Layer/VC/Route table
    - Weight field introduces a barrier type of packet when the weight counter expires
    - Egress packet to the router interface is marked as barrier packet from SOP, thru all data flits till EOP
- Implements a parameterized flow mapping table which provides forwarding Layer/VC/Route information
    - This is an associative table. The key {QoS value, Source interface ID, Destination interface ID, Host port ID } is compared with all the keys stored in the table.
        - If the key matches the corresponding outgoing Layer, VC and Router information is read from the table
        - If the key does not match then illegal destination flag is set and an Interrupt is raised. Illegal destination flag can be reset using the Register Bus which also resets the Interrupt
    - Each Host interface accesses the table in parallel
    - Route information from the mapping table is forwarded to the Router when the flit leaves the TX bridge.
- In each layer, host interface with a valid beat arbitrates for the layer

  - The winning host interface locks the VC of a particular layer on a SOP and unlocks it on a EOP
  - For a layer if a VC is locked by a particular host interface, another VC can be used for another host interface. Hence VC interleaving is allowed.
  - When a beat is granted on a VC, corresponding host FIFO is pop'ed and credit is returned
  - Arbiter is requested only when there is a host request, the corresponding VC is not locked and credits are available for the requested VC
- Implements Data width upsizing
  - The number of bits or width of the outgoing and incoming interfaces are individually configurable through parameters and must be in power of two multiple of cell_size. For testing minimum limit is 32/36 (depending on cell size)
  - Upsizing (data packing) is done according to Host interface data width and corresponding outgoing VC data width
  - Maximum ratio of 1:4 is supported i.e. Host interface to VC datawidth can be 1:1, 1:2, 1:3 or 1:4

2.2 Streaming RX Bridge

A high level summary of features for the Streaming RX Bridge is provided in this section.

- Connects to maximum of 8 layers. Each layer implements
  - Up to 4 VCs.
  - Credit based flow control
  - Implements a power of 2 (e.g. 2, 4, 8) deep input FIFO per VC. Depth of the FIFO can be different for each VC
- Supports up to 4 RX Host interfaces. Each interface implements
  - SOP, EOP, Data beat and Data beat valid.
  - Credit based flow control
- Implements fully parameterized Inputs/Ouputs on Router and Host interfaces
- Implements two sets of arbiters, VC arbiter and Layer arbiter
  - VC arbiter is instantiated per layer
    - This is a priority round robin arbiter. The priorities are programmed by NocStudio
    - The winning VC also forwards the priority to the layer
    - This arbiter does not update the pointers till layer arbiter grants the VC winning request
  - Layer arbiter is instantiated per host interface
    - This is also a priority round robin arbiter. The priorities are given by VC arbiter and they are per layer
- All Hosts interface credits are maintained. At the time of reset RX Bridge starts with credits programmed by NocStudio. The credit counters are incremented when host interface returns the credits and are decremented when a valid data beat is provided to the host interface.

- Layer that wins arbitration locks the Host interface to the winning VC. The lock is enabled on SOP and released of on an EOP. On a given layer, 4 interfaces can be locked to 4 different VCs which allow VC interleaving.
- When a beat is granted for a VC of a particular layer, corresponding VC FIFO is pop'ed and credit is returned
- Implements Data width downsizing
  - The number of bits or width of the outgoing and incoming interfaces are individually configurable through parameters and must be in power of two multiple of cell_size. For testing minimum limit is 32/36 (depending on cell size)
  - Downsizing (data unpacking) is done according to corresponding incoming VC data width and outgoing Host interface data width
  - Maximum ratio of 4:1 is supported i.e. VC data width to Host interface data width can be 1:1, 2:1, 3:1 or 4:1

3 Description

Bi-directional Streaming Bridge is instantiated in NetSpeed NoC IP to provide a simple interface to the SoC IP hosts. A bi-directional streaming bridge "ns_strbrdg" instantiates a streaming TX bridge "ns_strtxbrdg" and a RX bridge "ns_strrxbrdg". Each of these bridges (TX or RX) can be disabled using parameters P_STR_TX_ENABLE and P_STR_RX_ENABLE respectively.

3.1 Streaming TX Bridge

Figure below shows simplified schematic block diagram of TX Bridge. The configuration shown in the block diagram is of a TX bridge with 4 Host interfaces on the ingress side and 4 layers on the egress side.

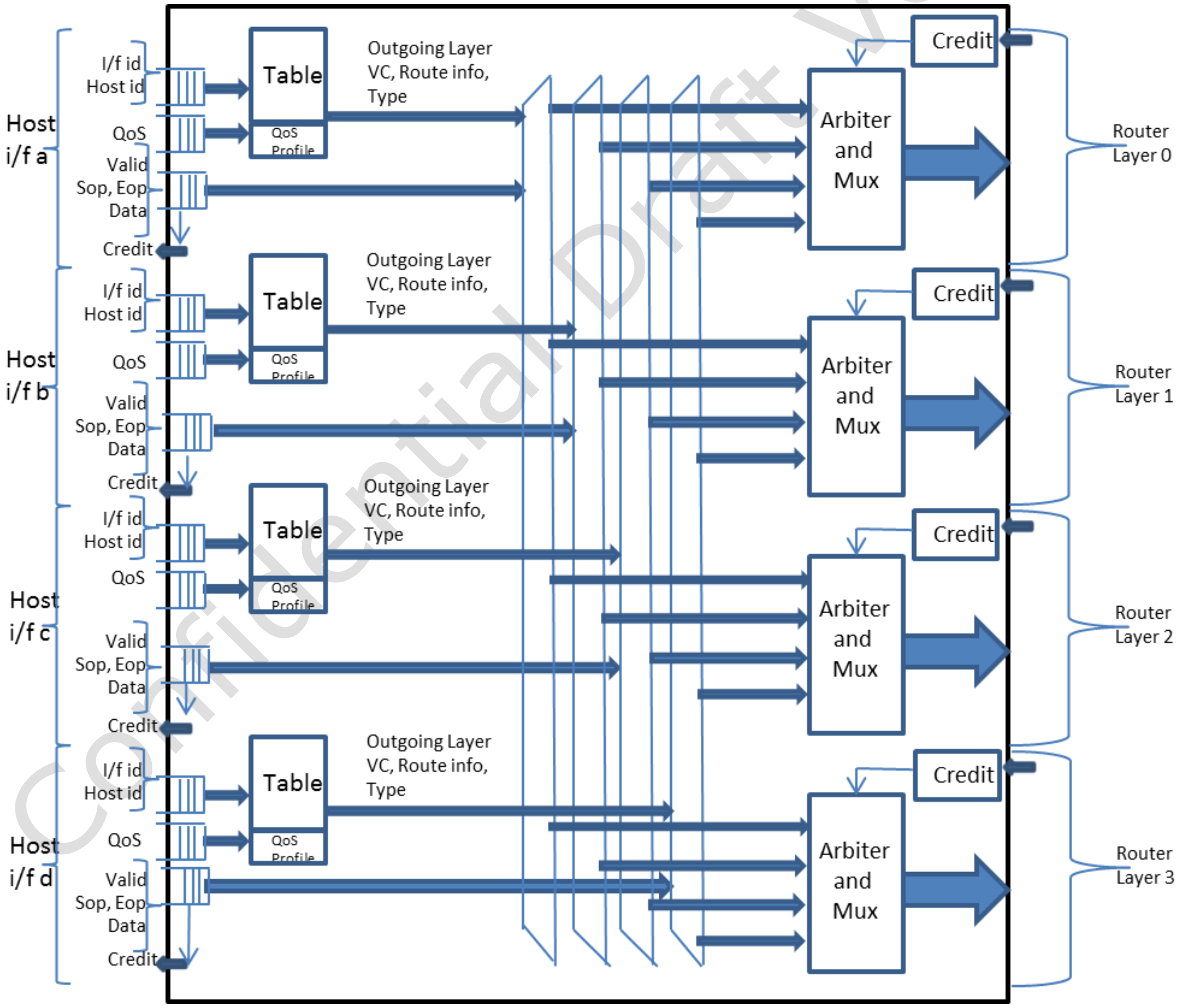

Figure 2: Streaming TX Bridge 4 host interface 4 layer block diagram

TX Bridge interfaces with the Streaming TX host port on the ingress side and interfaces with Router layers on the egress side. Streaming hostports can use up to four separate physical interfaces supported by the bridge. The interfaces are named a, b, c and d. On the egress side, TX bridge interfaces to up to 8 physical outgoing layers (block diagram only shows 4). Each layer can have up to 4 VCs each. Section **Error! Reference source not found.** hows the interface signals for host interfaces as well as layer interfaces.

3.1.1Input FIFO

Host interface signals get flopped using an input FIFO on every valid beat. There is a input FIFO for every Host Interface. The depth of each FIFO is controlled by NocStudio. The input FIFO can be optimized out if the Host interface is not enabled. In the current implementation the depth of the FIFO is a Log 2 function, therefore the depth of the FIFO can be 2, 4 or 8.

3.1.2Mapping Table

TX Bridge maps the hostport interface messages to VC and NoC layer. A mapping table is used for this purpose; the index to this table is QoS (priority) (4 bits), Source interface (2 bits), Destination interface (2 bits), and Destination hostport id (8 bits) of the message. The mapping table provides VC and NoC layer over which this message will be sent. In addition, the mapping table also provides the route information, which is used to route the resulting packet from this message.

On a valid transaction, the output of the FIFO accesses this table to get the forwarding Layer, VC and routing information. This information along with the beat valid, SoP, EoP and the data beat is then de-multiplexed to the egress layer interfaces

3.1.3Per Layer logic

De-multiplexers deliver the message (Beat valid, SoP, EoP and the Data) from all four incoming host interfaces to each layer. A round robin arbiter arbitrates between the host requests. The wining host interface acquires and locks the VC. When a multi-beat message is sent, the SOP of the message locks a VC and the EOP releases the VC. Multiple packets cannot be interleaved on a VC; however multiple VCs may be interleaved on the injection port to the router. For example, the possible scenario can be that Host interface “a” starts sending a multi beat packet and it wins the arbitration and acquires VC0 of Layer 0 i.e. Layer 0/VC0 will be locked on SOP and will be released on EOP. In parallel, Host interface “b” starts sending a multi beat packet and it acquires and locks VC1 of Layer 0, in the similar way “c” and “d” can lock VC2 and VC3 of Layer 0 respectively.

3.1.4Width conversion

TX Streaming Bridge packetizes the message beats into packet flits based on the interface width and the VC width. VC width is guaranteed to be equal to or wider than the interface width by construction. The NoC channel's widths are configured by NocStudio based on then source and destination interface widths whose messages go over the channel. If interface width and VC width are identical, then a single message beat becomes a single packet flit. Otherwise, multiple beats of a message are combined into a single packet flit at the TX Bridge; this process is called packing or upsizing. Each hostport interface's data width must be power of two multiple of cell_size. The maximum upsizing that can be done in TX is 1:4. There are 4 upsizing buffers per layer, which is one upsizing buffer per host interface.

3.1.5Flow control

There is a credit based flow control at both the hostport side and the router side.

3.1.6Output Registering

Output registering is NOT done in the TX Bridge, Router will register all the incoming signals from TX Bridge.

3.2 Streaming RX Bridge

Figure below shows simplified schematic block diagram of RX Bridge

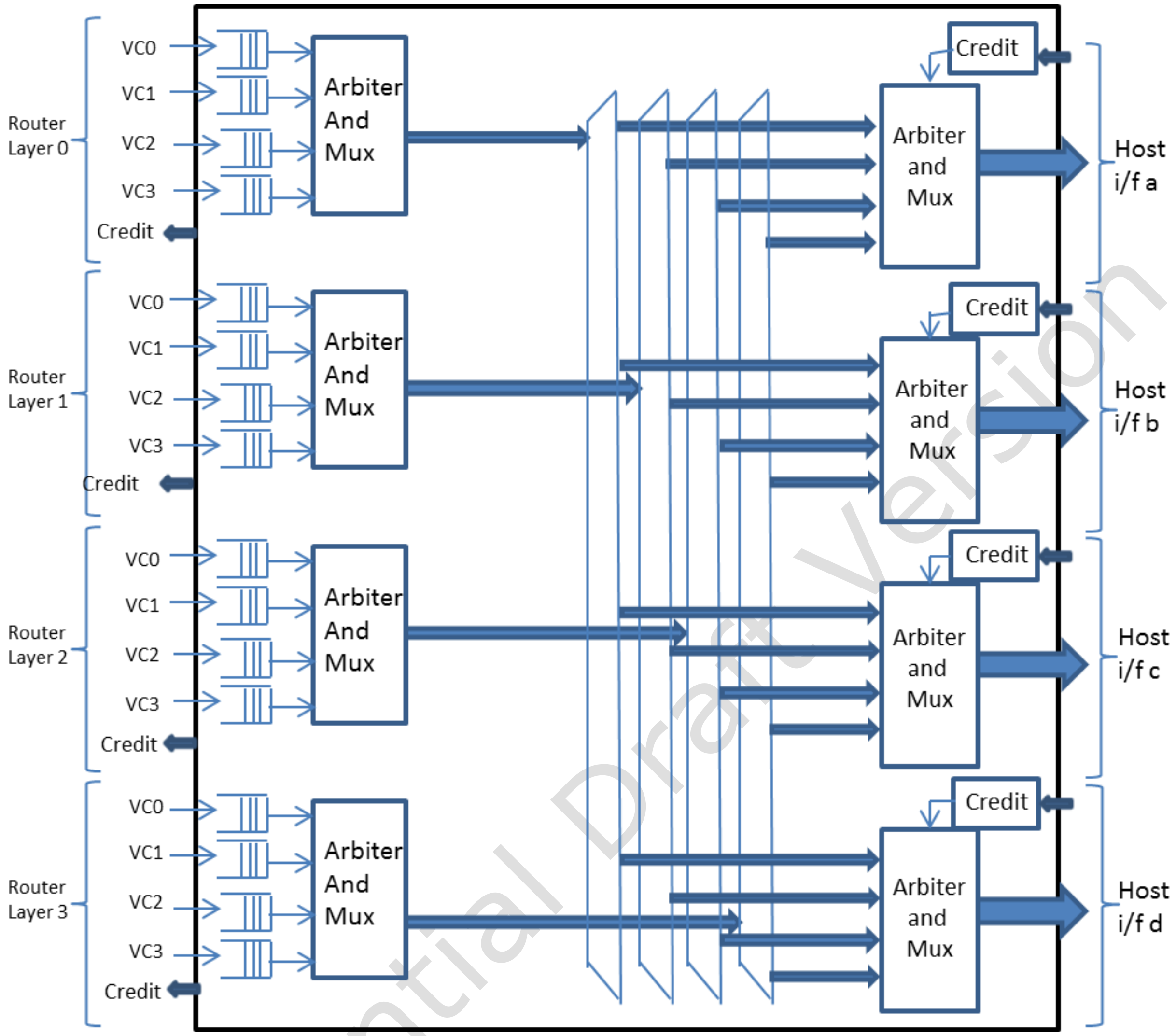

Figure 3: Streaming RX Bridge 4 layer 4 host interface block diagram

RX Bridge interfaces Router layers on the ingress side and interfaces with the Streaming RX host port on the egress side. There can be up to 8 physical incoming layers (block diagram only shows 4) on the ingress side. Each layer can have up to 4 VCs each. Streaming RX host ports can use up to four separate physical interfaces supported by the bridge. The interfaces are named a, b, c and d.

3.2.1Input FIFO

Router layer interface signals get flopped using an input FIFO on every valid flit. Since the bridge supports up to 4 VCs, there can be up to 4 input FIFOs. The depth of these FIFO is independently controlled by NocStudio. The input VC FIFO can be optimized out if the VC is not enabled. In the current implementation the depth of the FIFO is a Log 2 function, therefore the depth of the FIFO can be 2, 4 or 8.

3.2.2VC Arbiter

VC requests from the FIFO goes to a VC arbiter that is placed in each layer. This is a priority round robin arbiter. The priorities of each layer/VCs are programmed by NocStudio. The wining VC request along with the Priority, Beat valid, SoP, EoP and the Data is then de-multiplexed to all egress host interface logic.

3.2.3Egress per Host Interface logic

De-multiplexers deliver the message (Priority, Beat valid, SoP, EoP and the Data) from all incoming layers. A priority round robin arbiter arbitrates between the layer requests. The wining layer request acquires and locks host interface. When a multi-beat message is sent, the SOP of the message locks the host interface and the EOP releases the host interface.

3.2.4Width conversion

RX Streaming Bridge de-packetizes the packet flits into data beats for delivery based on the interface width and the VC width. VC width is guaranteed to be equal to or wider than the interface width by construction. If interface width and VC width are identical, then each packet flit becomes a single message beat. Otherwise, a single packet flit is divided into multiple message beats for delivery to the interface; this process is called unpacking or downsizing. Each hostport interface's data width must be power of two multiple of cell_size. The maximum downsizing that can be done in RX is 4:1

3.2.5Flow control

There is a credit based flow control at both the router side and the host port side

3.2.6Output Registering

Output registering is NOT done in the RX Bridge, Host interfaces have to register all the incoming signals from TX Bridge.

4 Coarse Clock Gating

The idea behind coarse clock gating of Streaming Bridge is to save power when it is in idle condition for substantially long period of time.

4.1 Streaming Tx Bridge Clock Gating

There are two aspects of coarse clock gating in Streaming Tx Bridge

- Generating ifce_busy signal per NoC layer for the Noc element it is attached to
- Self clock gating based on the idle condition of the Streaming Bridge

4.1.1 Interface busy signal from Streaming Tx Bridge

On each NoC layer through which it sends traffic, the Streaming Tx bridge is attached to the host port of a router. A dedicated busy/idle signal is required to be sent to each one of these routers for coarse clock gating purpose. Thus the busy signal is going to be one per NoC layer. The busy signal can be set, once the outgoing layer information corresponding to an incoming message beat is obtained. The outgoing layer information is obtained through look up of a mapping table. The look up is based on Source Interface and *QoS value, Destination Interface Id, Destination Host port Id signals* provided by the host through that particular Source Interface. If the output of the look up matches to a particular layer, busy signal for that particular layer will be set. An "*Outgoing Layer X Match*" signal will be generated for each Noc Layer. The "*Outgoing Layer X Match*" is set by the message entry at the head of the input interface Fifo, doing a lookup of the mapping table and the look up output matching a particular Noc Layer X. An "*Outgoing Layer X Status*" is maintained for each Noc Layer and is set high a cycle after "*Outgoing Layer X Match*" is asserted. "*Outgoing Layer X Busy*" signal is OR function of "*Outgoing Layer X Match*" and "*Outgoing Layer X Status*".

The register "*Output Layer X Status*" is required to let a message (at the head of the interface FIFO) know that corresponding host port and logic in Router on Noc Layer X is already clock enabled. A message can only arbitrate for Output Layer X in Streaming Bridge, if "*Output Layer X Status*" is set to high.

The "*Output Layer X Status*" register will be cleared after a fixed number of cycles, after all the transactions across all the host input interfaces have been exhausted and the credits for all transactions to the NoC have been returned.

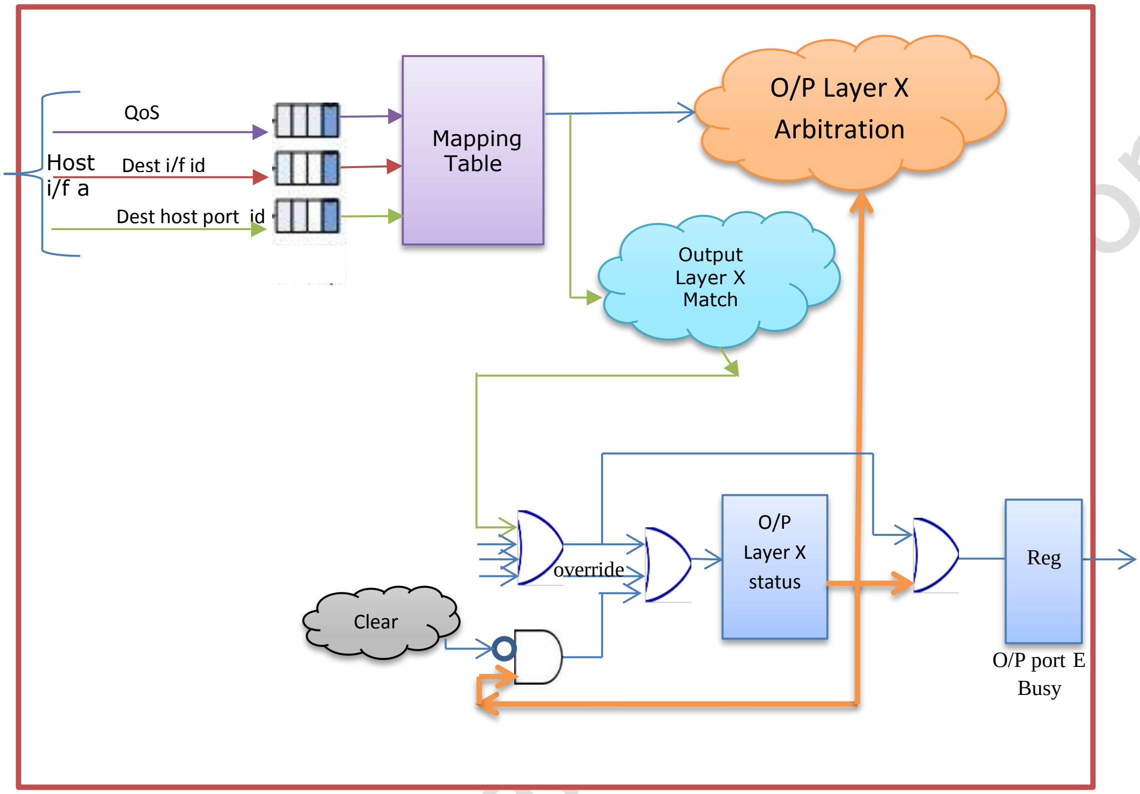

Figure 4 Interface Busy signal generation per layer from Streaming Bridge

4.1.2 Self clock gating of Streaming Tx Bridge

The whole Streaming Tx Bridge block is clock gated off once there are no outstanding messages in the per interface VC Fifos or currently being processed, and all credits from NoC have been returned.

4.2 Streaming Rx Bridge clock gating

The Streaming Bridge Receive block (traffic from NoC) is clock gated off based on traffic condition on a layer by layer basis. This scheme is slightly different than clock gating scheme for Transmit Block. For each NoC layer, there will be associated coarse clock gating logic. Once the input VC Fifos for a particular NoC layer have been drained, the clock gating logic will wait for the "*Input Layer X Busy*" signal from its attached router on that layer to be de-asserted. Once the condition is reached that the input VC Fifos for that NoC layer X are empty (credits returned back to the router) and "*Input Layer X Busy*" signal have been de-asserted, the associated logic could be clock gated off.

The whole Receive Block can be clock gated off, once all the input logic for all NoC Layers have been clock gated and all credits have been returned from the host to the Streaming Bridge.

5 Streaming Bridge Interfaces

Following are the interface signals of bi- directional streaming bridge

Signal name	Width	Input/ Output	Description
noc_clk	Single	Input	Clock
noc_reset	Single	Input	Reset
System_clk_en	Single	Input	System clock enable
System_cg_or	Single	Input	System Clock Gate override
Scan_mode	Single	Input	Scan mode override for clock gating cell
Host to/from Streaming TX bridge Signals			
hst_strtxbrdg_ifid	[P_STR_TX_HST_PORTS*P_STR_TX_IFID_WIDTH-1:0]	Input	Output Interface ID
hst_strtxbrdg_hpid	[P_STR_TX_HST_PORTS*P_STR_TX_HPID_WIDTH-1:0]	Input	Output Host Port ID
hst_strtxbrdg_qos	[P_STR_TX_HST_PORTS*P_STR_TX_QOS_WIDTH-1:0]	Input	Incoming QoS value
hst_strtxbrdg_sop	[P_STR_TX_HST_PORTS-1:0]	Input	Start of Packet
hst_strtxbrdg_eop	[P_STR_TX_HST_PORTS-1:0]	Input	End of Packet
hst_strtxbrdg_dbeat	[P_CELL_SIZE*P_STR_TX_TNO_OF_HSTCELLS-1:0]	Input	Data Beat
hst_strtxbrdg_dbeat_vld	[P_STR_TX_HST_PORTS-1:0]	Input	Data Beat Valid
strtxbrdg_hst_credit_inc	[P_STR_TX_HST_PORTS-1:0]	Output	Credit info
Router to/from Streaming TX bridge Signals			
rt_strtxbrdg_credit_inc	[P_STR_TX_RT_NOOFLAYERS*P_STR_TX_RT_NOOFVC*P_STR_TX_RT_CR_WIDTH-1:0]	Input	Credit info
strtxbrdg_rt_flit_sop	[P_STR_TX_RT_NOOFLAYERS-1:0]	Output	Start of Packet
strtxbrdg_rt_flit_eop	[P_STR_TX_RT_NOOFLAYERS-1:0]	Output	End of Packet
strtxbrdg_rt_flit_bv	[P_TX_TNOBV_WIDTH-1:0]	Output	Byte Valid in the flit with eop
strtxbrdg_rt_flit_valid	[P_STR_TX_RT_NOOFLAYERS*P_STR_TX_RT_NOOFVC-1:0]	Output	Flit valid
strtxbrdg_rt_flit_data	[P_CELL_SIZE*P_STR_TX_TNO_OF_LAYCELLS-1:0]	Output	Flit data
strtxbrdg_rt_flit_type	[P_STR_TX_RT_NOOFLAYERS-1:0]	Output	Flit type 0: Normal; 1:barrier
strtxbrdg_rt_flit_sb	[P_STR_TX_RT_NOOFLAYERS*P_STR_TX_RT_SB_WIDTH-1:0]	Output	Side band signals
strtxbrdg_rt_flit_outp	[P_STR_TX_RT_NOOFLAYERS*P_STR_TX_RT_OP_WIDTH-1:0]	Output	Router out port
Strtxbrdg_rt_cg_busy	[P_STR_TX_RT_NOOFLAYERS-1:0]	Output	Clock Gating : Output NoC Layer Busy - Flits outstanding for

			that NoC layer.
Router to/from Streaming RX bridge Signals			
rt_strrxbrdg_flit_sop	[P_STR_RX_RT_NOOFLAYERS-1:0]	Input	Start of Packet
rt_strrxbrdg_flit_eop	[P_STR_RX_RT_NOOFLAYERS-1:0]	Input	End of Packet
rt_strrxbrdg_flit_bv	[P_RX_TNOBV_WIDTH-1:0]	Input	Byte Valid in the flit with eop
rt_strrxbrdg_flit_valid	[P_STR_RX_RT_NOOFLAYERS*P_STR_RX_RT_NOOFVC-1:0]	Input	Flit valid
rt_strrxbrdg_flit_data	[P_CELL_SIZE*P_STR_RX_TNO_OF_LAYCELLS-1:0]	Input	Flit data
rt_strrxbrdg_flit_type	[P_STR_RX_RT_NOOFLAYERS-1:0]	Input	Flit type 0: Normal; 1:barrier
rt_strrxbrdg_flit_sb	[P_STR_RX_RT_NOOFLAYERS*P_STR_RX_RT_SB_WIDTH-1:0]	Input	Side band signals (2 bits USED)
rt_strrxbrdg_flit_outp	[P_STR_RX_RT_NOOFLAYERS*P_STR_RX_RT_OP_WIDTH-1:0]	Input	Router out port (NOT USED)
strrxbrdg_rt_credit_inc	[P_STR_RX_RT_NOOFLAYERS*P_STR_RX_RT_NOOFVC*P_STR_RX_RT_CR_WIDTH-1:0]	Output	Credit info
rt_strrxbrdg_cg_busy	[P_STR_TX_RT_NOOFLAYERS-1:0]	Input	Clock Gating : Input Layer Busy – Flits outstanding from a NoC Layer into the Streaming Rx bridge
Host to/from Streaming TX bridge Signals			
hst_strrxbrdg_credit_inc	[P_STR_RX_HST_PORTS-1:0]	Input	Credit info
strrxbrdg_hst_sop	[P_STR_RX_HST_PORTS-1:0]	Output	Start of Packet
strrxbrdg_hst_eop	[P_STR_RX_HST_PORTS-1:0]	Output	End of Packet
strrxbrdg_hst_dbeat_vld	[P_STR_RX_HST_PORTS-1:0]	Output	Data beat valid
strrxbrdg_hst_dbeat	[P_CELL_SIZE*P_STR_RX_TNO_OF_HSTCELLS-1:0]	Output	Data beat

Table 1: Streaming Bridge Inputs/Outputs

6 Streaming Bridge Parameters

Following are the parameters used by the bi- directional streaming bridge

Parameter	Description
Streaming TX bridge parameters	
parameter P_STR_TX_ENABLE	Parameter to enable/disable TX bridge in a bi-directional instance 0: TX Disabled 1: TX Enabled
parameter P_STR_TX_L2_HST_INTERFACES	Log 2 of TX Host Interfaces
parameter P_STR_TX_HST_ INTERFACES	Total number of TX Host Interfaces
parameter P_STR_L2_INFIFODEPTH_WIDTH	Width of Log 2 of input FIFO depth
parameter P_STR_TX_ L2_INFIFODEPTH_WIDTH (array) [P_STR_TX_HST_INTERFACES*P_STR_L2_INFIFODEPTH_WIDTH-1:0]	Array of parameters indicating Log 2 of input FIFO depth
parameter P_STR_TX_HST_SINGLEBEAT	Not used in this release
parameter P_STR_IFID_WIDTH	Interface ID width
parameter P_STR_HPID_WIDTH	Host port ID width
parameter P_STR_QOS_WIDTH	QoS profile index width
parameter P_STR_TX_RT_SB_WIDTH	Router side band signal width of the TX bridge
parameter P_STR_TX_RT_OP_WIDTH	Router out port width of the TX bridge
parameter P_STR_TX_RT_CR_WIDTH	Router credit counter width for each VC
parameter P_STR_TX_L2_NOOFLY	Log 2 of number of layers of the TX bridge
parameter P_STR_TX_RT_NOOFLAYERS	Total number of layers of the TX bridge
parameter P_STR_TX_L2_NOOFVC	Log 2 of max number of VCs per layer of the TX bridge (Fixed to 2)
parameter P_STR_TX_RT_NOOFVC	Max number of VCs per layer of the TX bridge (Fixed to 4)
parameter P_CELL_SIZE	Parameter to indicate CELL_SIZE width. This width stays the same throughout the NoC
parameter P_STR_TX_MAX_LAYDWIDTH	Max Layer data width can be calculated using localparam P_MAXLAY_DATA_WIDTH = P_CELL_SIZE*2**P_STR_MAX_LAYDWIDTH;
parameter P_STR_TX_TNO_OF_HSTCELLS	This parameter is used to define the data bus width for all interfaces that are connected to host TX port. Total data width is P_CELL_SIZE * P_STR_TX_TNO_OF_HSTCELLS
parameter P_STR_TX_TNO_OF_LAYCELLS	This parameter is used to define the data bus width for all layer physical interfaces that are connected to host TX port. Total data width is P_CELL_SIZE * P_STR_TX_TNO_OF_LAYCELLS
localparam P_STR_TX_MAXNOOF_L2_CELLS	Width of Log 2 of maximum number of cells

parameter P_STR_TX_HST_DWIDTH (array) [P_STR_TX_HST_PORTS*P_STR_TX_MAXNOOF_L2_CELLS -1:0]	Data width of each host interface
parameter P_STR_TX_LAY_DWIDTH (array) [P_STR_TX_RT_NOOFLAYERS*P_STR_TX_MAXNOOF_L2_CELLS-1:0]	Physical data width of each layer
parameter P_STR_TX_LAYVC_DWIDTH (array) [P_STR_TX_RT_NOOFLAYERS*P_STR_TX_RT_NOOFVC*P_STR_TX_MAXNOOF_L2_CELLS-1:0]	Data width of each VC per layer
parameter P_STR_TX_MAXDATAWIDTH	Max number of bits needed to specify data width
parameter P_STR_TX_HST_DATAWIDTH [P_STR_HST_PORTS*P_STR_MAXDATAWIDTH-1:0]	Host Data widths
parameter P_STR_TX_LAY_DATAWIDTH [P_STR_RT_NOOFLAYERS*P_STR_MAXDATAWIDTH-1:0]	Layer Data widths
parameter P_STR_TX_MAXWCWIDTH	Max number of bits needed to specify number of cells valid in a flit information
parameter P_STR_TX_RT_WC_WIDTH (array) [P_STR_TX_RT_NOOFLAYERS*P_STR_TX_MAXWCWIDTH-1:0]	Hold information of number of cells valid per flit per layer
parameter P_STR_TX_VC_INITIAL_CREDITS (array) [P_STR_TX_RT_NOOFLAYERS*P_STR_TX_RT_NOOFVC*P_STR_TX_RT_CR_WIDTH-1:0]	Initial VC credits per layer
parameter P_STR_TX_ VC_ENB (array)	Parameter used to optimize out logic when VC(s) are not enabled in TX bridge.
parameter P_STR_TX_ HSTIF_ENB (array)	Parameter used to optimize out logic when Host Interfaces are not enabled in TX bridge.
parameter P_STR_BARRIER_EN	Barrier enable for the TX bridge
parameter P_STR_QOSWT_WIDTH	Width of an individual weight counter
localparam P_STR_QOSTBL_DEPTH (2**P_STR_QOS_WIDTH)	Depth of QoS profile table
parameter P_STR_QOSWT (array) [P_STR_QOSTBL_DEPTH*P_STR_QOSWT_WIDTH-1:0]	Contents of QoS profile table (weight counters)
localparam P_STR_ROUTE_INFO = P_STR_RT_OP_WIDTH+P_STR_RT_SB_WIDTH-2	Total width of Routing table info
parameter P_STR_TBL_NOOFENT	Defines number of entries in the lookup associative table that store forwarding information
parameter P_STR_TBL_KEY (array) [P_STR_KEY_WIDTH*P_STR_TBL_NOOFENT-1:0]	Array of keys stored in the lookup associative table that store forwarding information
parameter P_STR_TBL_VAL (array) [P_STR_OPLYVC_WIDTH*P_STR_TBL_NOOFENT-1:0]	Array of forwarding LAYER/VC information stored in the lookup associative table
parameter P_STR_RTTBL_VAL (array) [P_STR_ROUTE_INFO *P_STR_TBL_NOOFENT-1:0]	Array of forwarding routing information stored in the lookup associative table
Streaming RX bridge parameters	
parameter P_STR_RX_ENABLE	Parameter to enable/disable RX bridge in a bi-directional instance 0: RX Disabled 1: RX Enabled
parameter P_STR_RX_L2_HST_INTERFACES	Log 2 of RX Host Interfaces
parameter P_STR_RX_HST_ INTERFACES	Total number of RX Host Interfaces
parameter P_STR_RX_RT_SB_WIDTH	Router side band signal width of the RX bridge
parameter P_STR_RX_CR_WIDTH	Host credit counter width for each host

	interface
parameter P_STR_RX_L2_NOOFLY	Log 2 of number of layers of the RX bridge
parameter P_STR_RX_RT_NOOFLAYERS	Total number of layers of the RX bridge
parameter P_STR_RX_L2_NOOFVC	Log 2 of max number of VCs per layer of the RX bridge (Fixed to 2)
parameter P_STR_RX_RT_NOOFVC	Max number of VCs per layer of the RX bridge (Fixed to 4)
parameter P_STR_RX_ L2_INFIFODEPTH_WIDTH (array) [P_STR_RX_RT_NOOFLAYERS*P_STR_RX_RT_NOOFVC*P_STR_L2_INFIFODEPTH_WIDTH-1:0]	Array of parameters indicating Log 2 of input FIFO depth
parameter P_STR_RX_MAX_LAYDWIDTH	Max Layer data width can be calculated using localparam P_MAXLAY_DATA_WIDTH = P_CELL_SIZE*2**P_STR_RX_MAX_LAYDWIDTH;
parameter P_STR_RX_TNO_OF_HSTCELLS	This parameter is used to define the data bus width for all interfaces that are connected to host RX port. Total data width is P_CELL_SIZE * P_STR_TX_TNO_OF_HSTCELLS
parameter P_STR_RX_TNO_OF_LAYCELLS	This parameter is used to define the data bus width for all layer physical interfaces that are connected to host RX port. Total data width is P_CELL_SIZE * P_STR_RX_TNO_OF_LAYCELLS
parameter P_STR_RX_HST_INITIAL_CREDITS (array) [P_STR_RX_HST_ INTERFACES *P_STR_RX_CR_WIDTH-1:0]	Initial host credits per interfaces
parameter P_STR_RX_ VC_ENB (array)	This parameter used to optimize out logic when VC(s) are not enabled in RX bridge.
parameter P_STR_RX_ HSTIF_ENB (array)	This parameter used to optimize out logic when Host Interfaces are not enabled in RX bridge.
parameter P_STR_QOSPRI_WIDTH	This parameter used to define the width of priority bits so that we can index into Layer/VC priority table
Parameter P_STR_LAYVC_PRI (array) [P_STR_RX_RT_NOOFLAYERS*P_STR_RX_RT_NOOFVC*P_STR_RX_QOSPRI_WIDTH-1:0]	Layer/VC priority array/table
localparam P_STR_RX_MAXNOOF_L2_CELLS	Width of Log 2 of maximum number of cells
parameter P_STR_RX_HST_DWIDTH (array) [P_STR_RX_HST_PORTS*P_STR_RX_MAXNOOF_L2_CELLS-1:0]	Data width of each host interface
parameter P_STR_RX_LAY_DWIDTH (array) [P_STR_RX_RT_NOOFLAYERS*P_STR_RX_MAXNOOF_L2_CELLS-1:0]	Physical data width of each layer
parameter P_STR_RX_LAYVC_DWIDTH (array) [P_STR_RX_RT_NOOFLAYERS*P_STR_RX_RT_NOOFVC*P_STR_RX_MAXNOOF_L2_CELLS-1:0]	Data width of each VC per layer
parameter P_STR_RX_MAXDATAWIDTH	Max number of bits needed to specify data width
parameter P_STR_RX_HST_DATAWIDTH	Host Data widths

[P_STR_RX_HST_PORTS*P_STR_RX_MAXDATAWIDTH-1:0]	
parameter P_STR_RX_LAY_DATAWIDTH [P_STR_RX_RT_NOOFLAYERS*P_STR_RX_MAXDATAWIDTH-1:0]	Layer Data widths
parameter P_STR_RX_MAXWCWIDTH	Max number of bits needed to specify number of cells valid in a flit information
parameter P_STR_RX_RT_WC_WIDTH (array) [P_STR_RX_RT_NOOFLAYERS*P_STR_RX_MAXWCWIDTH-1:0]	Hold information of number of cells valid per flit per layer

Table 2: Streaming Bridge parameters

Glossary

Term

Definition

Document Changes/Revisions

Documentation Changes include additions, deletions, and modifications made to this document. This section identifies the changes made in each release of the document.

Document Revision A

Update 1. NetSpeed Orion Streaming Bridge Microarchitecture spec - initial version

2670 Seely Ave
Building 11
San Jose, CA 95134
(408) 914-6962
www.netspeedsystems.com