RapidIO™ Interconnect Specification Part 12: Virtual Output Queueing Extensions Specification

Rev. 2.2, 06/2011

RapidIO Trade Association

Revision History

Revision	Description	Date
2.0	First release	06/14/2007
2.0	Public release	03/06/2008
2.1	No technical changes	MM/DD/200Y
2.1	Removed confidentiality markings for public release	08/13/2009
2.2	No technical changes	05/05/2011
2.2	Removed confidentiality markings for public release	06/06/2011

Questions regarding the RapidIO Trade Association, specifications, or membership should be forwarded to:

RapidIO Trade Association
12343 Hymeadow, Suite 2-R
(non-US mail deliveries to Suite 3-E)
Austin, TX 78750
512-401-2900 Tel.
512-401-2902 FAX.

Table of Contents

Table of Contents

Blank page

List of Figures

List of Figures

Blank page

List of Tables

List of Tables

Blank page

Chapter 1 Introduction

1.1 Problem Illustration

In the basic switch model shown below, head-of-line blocking occurs when a packet for port 3 cannot be transmitted on the link because of congestion in port 2. The link effectively stalls causing the congestion in port 2 to spread to traffic on other ports. The backpressure method described here helps alleviate congestion spreading caused by transient blockages of queueing structures.

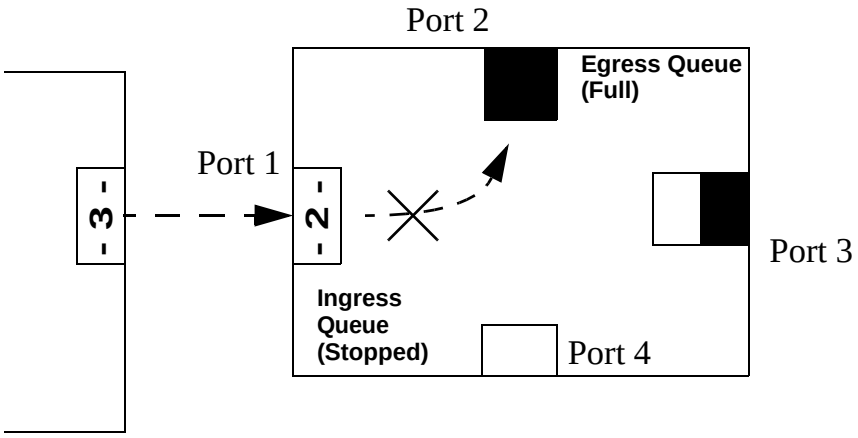


Figure 1-1. Basic Head-of-Line Blocking

The example is simplistic, but any queueing mechanism can become congested, head-of-line block, and stall the link to the upstream device. The VoQ Backpressure Process defines a *congestion message* that informs the upstream port about the congestion, allowing traffic to be sidelined (in a virtual output queue) in favor of traffic with a clear path ahead.

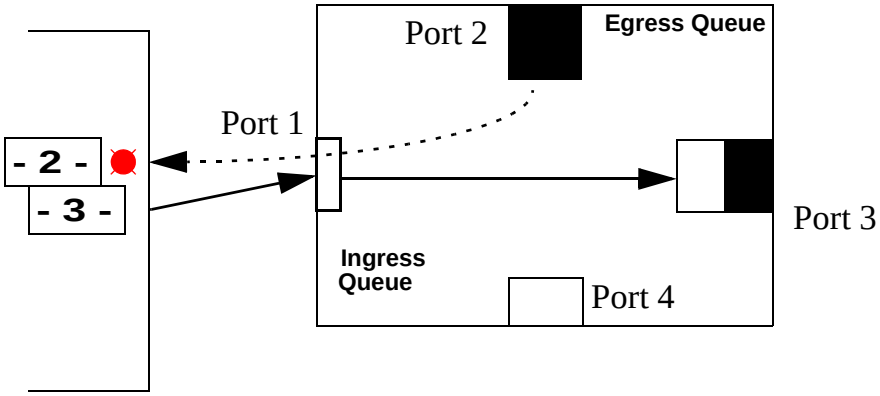


Figure 1-2. Effective Backpressure

Effective backpressure is achieved when the following elements exist:

a) The congested device can communicate congestion information to upstream devices.

b) The upstream device can segregate traffic and allow traffic to re-order based on that congestion information.

These two properties are essential. To keep the operation at the physical layer, the method described uses port identification for both the staging of traffic and the congestion status. Implementation of this specification is optional.

1.2 Terminology

Upstream Device - A device ahead of another device in the traffic flow. The upstream device is the recipient of the backpressure messages.

Downstream Device - The device later in the traffic flow. The downstream device is the originator of backpressure messages.

Port - a port is a local value associated with a specific physical interface. Every device with more than 1 port is responsible for mapping the destination ID to a local port.

Congestion Message - A bit, or group of bits indicating the congestion status of one or more ports.

Backpressure Symbol - A specific field in a RapidIO control symbol that contains the congestion message.

1.3 Conventions

All fields and message formats are described using big endian format.

|| Concatenation, used to indicate that two fields are physically associated as consecutive bits

italics	Book titles in text are set in italics.
REG[FIELD]	Abbreviations or acronyms for registers are shown in uppercase text. Specific bits, fields, or ranges appear in brackets.
TRANSACTION	Transaction types are expressed in all caps.
operation	Device operation types are expressed in plain text.
n	A decimal value.
[*n*-*m*]	Used to express a numerical range from *n* to *m.*
0b*nn*	A binary value, the number of bits is determined by the number of digits.
0x*nn*	A hexadecimal value, the number of bits is determined by the number of digits or from the surrounding context; for example, 0x*nn* may be a 5, 6, 7, or 8 bit value.
x	This value is a don’t care.
<variable>	Identifies a logical variable that may be a specific field of a register or packet or data structure.

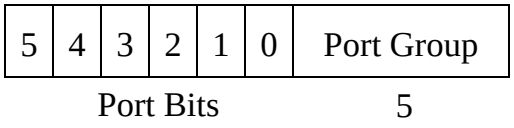
Blank page

Chapter 2 Overview

The purpose of this backpressure method is to maintain system performance during temporary congestion caused when statistical peaks in traffic flow oversubscribe the ability of a port and its associated buffering to handle the peak load. The backpressure avoids blocking of crossing traffic that competes for common resources. As such, the scope of the message is limited to a device and its immediately adjacent upstream devices. The system may be designed such that sustained congestion will cause a cascade of backpressure messages, but the ability to avoid degredation of performance drops as the radius of affected devices increases. RapidIO has other flow control methods to manage more systemic traffic impediments. Implementation of this specification is optional.

2.1 Congestion Message

Two key considerations for the message format described here are the efficiency and latency of the message, and independence for the two devices involved in the exchange from implementation differences. The congestion message uses a packed format to convey the status of multiple ports in a single message:



Port = Port Group * 6 + Offset
Port Bit = 1 = Congestion
Port Bit = 0 = No Congestion

Figure 2-1. Congestion Message for a Group of Ports

By combining 6 status bits with a port group, the number of total messages is reduced for larger port count devices. The format allows for devices up to 192 ports to be supported. Smaller devices my be able to communicate port status in one or two messages. A congestion message is typically transmitted when one at least of the ports' status changes. The congestion message is embedded in a field in the RapidIO control symbol. The symbol containing this message is defined as the backpressure symbol.

2.2 Traffic Staging

For the congestion message to be useful, the upstream device must segregate or stage traffic prior to committing it to the RapidIO link to the downstream device, or any critical resource (like a buffer) that could block other traffic. To stage the

packets, the upstream device must have knowledge about the routing configuration of the downstream device. A typical RapidIO switch will lookup incoming traffic and switch it to a port based on its destination ID. To support this backpressure method, that lookup must produce the egress port for the current device as well as the egress port for the next device. It is straightforward to align the routing tables, but it does require additional entries.

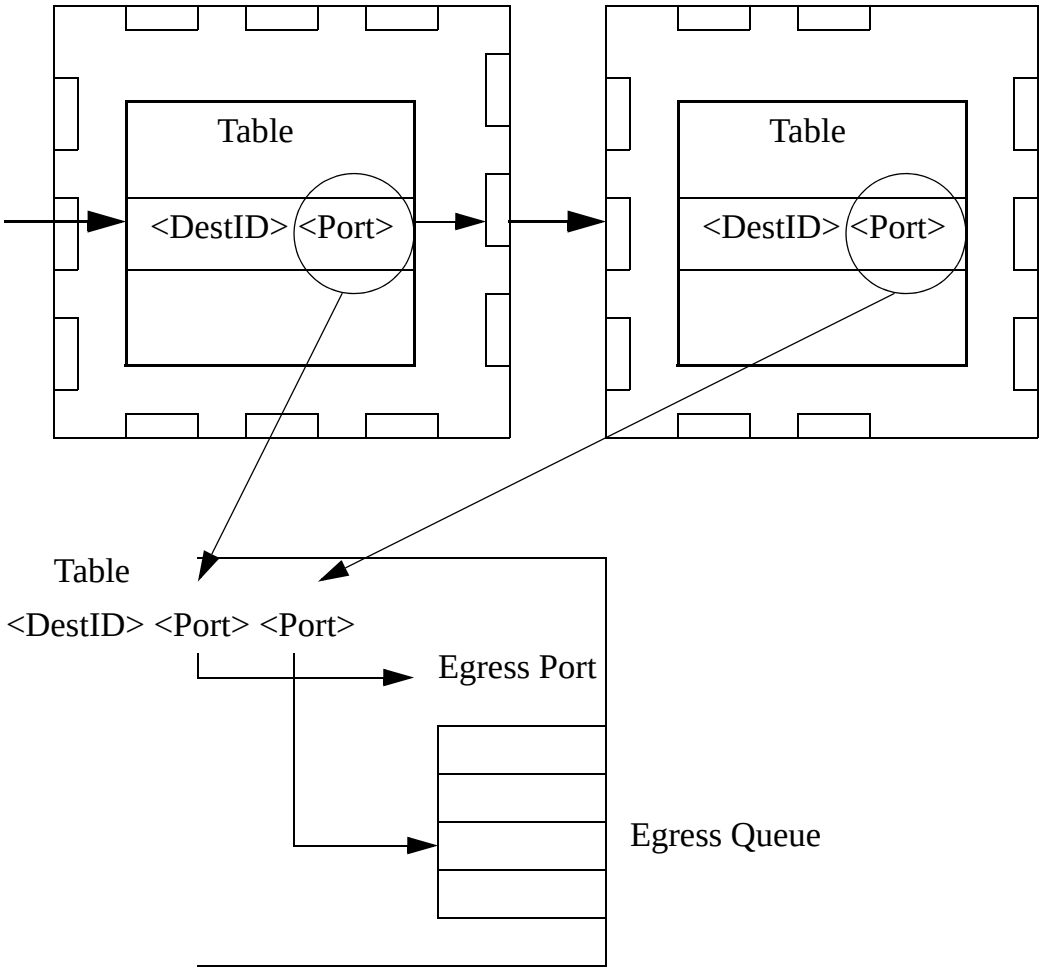


Figure 2-2. Adding Egress Staging

In the figure above, output staging is created by adding a second parameter to the routing table for the next hop port value. That value is the same value that has to be put in the downstream device's routing table. In the upstream device it is used to identify what queue to stage the traffic in, and what queue to act on when a congestion message is received. The value is specified in an implementation-dependant fashion.

For end points, there is normally no routing table. To use this form of backpressure, the end point would need a method to associate a destination ID with the downstream egress port.

2.3 Adding Device Independence

In the basic structure above, the traffic is staged in a queue that corresponds to a port in the downstream device. A difficulty arises to match the number of queues to the number of possible ports that might exist in the next device. It is inconsistent with RapidIO's goals to allow devices to implement cost and complexity as needed by their market to require every device to have 192 queues to support a maximum sized device, so an additional abstraction is required.

A device may combine traffic into fewer queues by reverse mapping the incoming port congestion message. In the figure below, the egress port supports 4 staging queues. The downstream device has 16 ports. So traffic for several ports are staged in a common queue. When a port specific message is received it is reverse mapped to the same queue that the forward lookup used to stage the data.

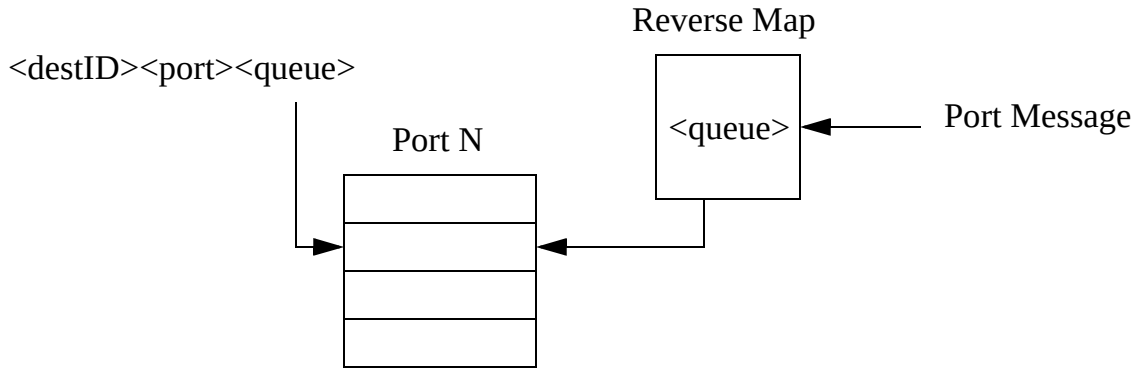


Figure 2-3. Mapping Staging Queues

In this example, as long as the forward/reverse mapping corresponds to the right downstream device's egress port, any implementation can be used, as it is all internal to a single device. This mapping requires that only enough bits for the number of queues be added to the forward table, which can be very large. In the reverse direction, the mapping can be RAM based for maximum flexibility (in the above example requiring a 256 x 2 bit RAM), or a straight decoder. This specification does not prescribe a specific method.

The only other requirement, when mapping multiple ports to a single queue, is that the queue must be shut down when any of those ports are congested. This will reduce the benefits of this backpressure method, but a significant amount of benefit is achieved with just a few queues.

2.4 Relationship With Virtual Channels

The staging method illustrated above uses queues in the output stage of the upstream device. Devices implementing multiple Virtual Channels will have additional queueing structures for the VCs. The staging queues may be before the traffic is sorted into its VC, or each VC may have a set of staging queues. When the output

queueing is not tied to the implementation of VCs, the congestion message is not linked to VC operation, and the backpressure symbol can be combined with any valid combinations of RapidIO symbols.

If the output queueing is embedded within the VC structure, the VoQ congestion message may be linked to VC operation. Both message formats are described in the next section. The congestion message may be associated with a specific VC. A CSR bit is provided to enable or disable the linking of the VC_Status and congestion messages into a common control symbol.

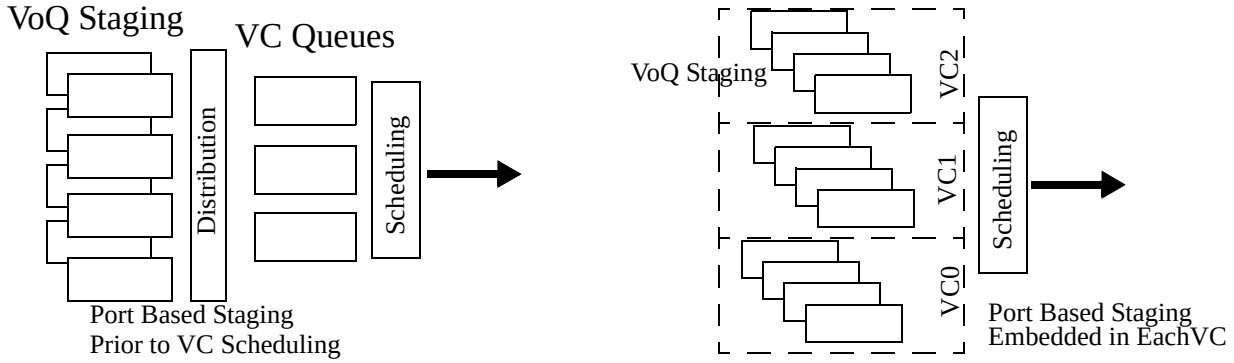


Figure 2-4. Linking VCs to VoQ Backpressure

2.5 Additional Queueing Considerations

If the downstream (originating) device is using queues in its output stage to determine congestion, then congestion messages will have to be sent to all ports that might be sources of incoming traffic.

If the downstream device is using virtual output queues, presorting traffic by egress port at the input, then it has the ability to reflect congestion status on only those ports that are receiving traffic for the congested output. Input queued switches do require N^2 queues (where N is the number of ports).

Devices using some combination of input and output queueing, or shared memory architectures, may make congestion decisions based on whatever resource allocation algorithm is being employed. It is important to consider some of the following boundary conditions:

- If the congestion message is issued with too little room in the port's egress to account for packets that might be in flight, head-of-line blocking can still occur.
- If very small queueing structures are used, a lot of on/off chatter can occur. This is not necessarily bad as long as the additional utilization of link bandwidth is accounted for.

The generation and application of the congestion message defined in this specification will be highly dependent on the queueing and switch design of the device, and as such, is left to the implementer.

Chapter 3 Control Symbol Format

The VoQ congestion message adds a control symbol to the *RapidIO Interconnect Specification Part 6: LP-Serial Physical Layer Specification*. Refer to that specification for the definitions of control symbols, packet delimiting, and the definitions of the fields not defined here. The VoQ backpressure symbol uses the extended control symbol defined for use with the 5/6G baud link speeds. This method may be used with the 1.25/2.5/3.125G baud link speeds but those links will have to be designed to support the extended control symbol.

3.1 Stype2 Control Symbol

The long control symbol is defined as follows in the *RapidIO Interconnect Specification Part 6: LP-Serial Physical Layer Specification*:

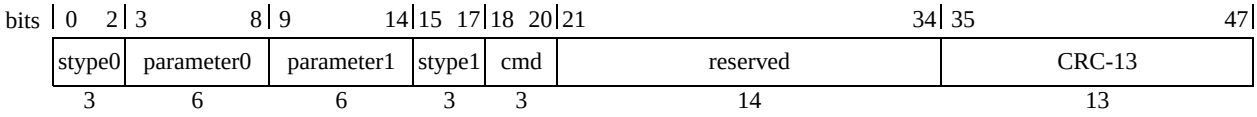


Figure 3-1. Long Control Symbol Format

The stype2 field uses the 14 reserved long control symbol bits and has an operation code (CMD) field and a parameter field. The VoQ backpressure symbol defines the follow usage for the stype2 symbol:

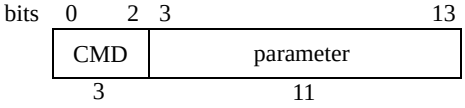


Figure 3-2. Stype2 Control Symbol Format

Table 3-1 shows the stype2 control symbol format definitions.

Table 3-1. Stype2 Symbol Format

Function	CMD	Parameter	
	Bits 0 - 2	Bits 3 - 13	
NOP	0b000	0b000 0000 0000	
VoQ Backpressure	0b001	Bits 3 - 8, Port Status Bits	Bits 9 - 13, Port Group
Reserved	0b010 - 0b111	0b000 0000 0000	

Table 3-2 shows the VoQ port status bit definitions.

Table 3-2. VoQ Port Status Bits

Port Status Bits	Offset
Bit 3	Port Offset 5
Bit 4	Port Offset 4
Bit 5	Port Offset 3
Bit 6	Port Offset 2
Bit 7	Port Offset 1
Bit 8	Port Offset 0
Status: 0b1 = Port is Congested 0b0 = Port is not Congested	

Table 3-3 shows the port group bit definitions.

Table 3-3. Port Group Bits

Port Group Bits 9 - 13	Base (Port Group * 6)
0b00000	0x00
0b00001	0x06
0b00010	0x0C
|	|
0b11111	0xBA
Base = Port group x 6, Port = Base + Offset	

3.2 VC_Status Symbol Linking

To specify a specific port within a VC for flow control, the VCID in the VC_Status symbol is used to identify the VC. This is done by combining the VoQ backpressure symbol and the VC_Status symbols in a single symbol. For VC0, the Status Symbol is used.

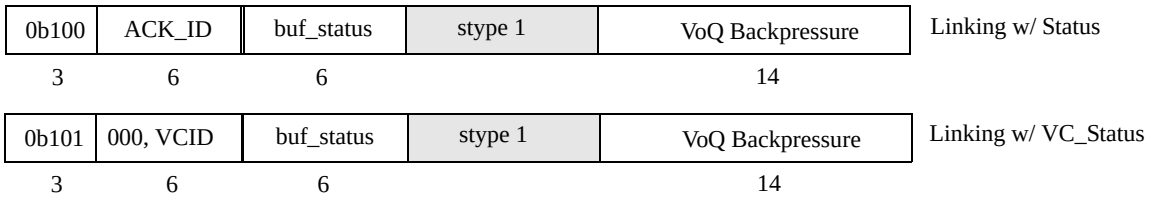


Figure 3-3. Linking VCs to VoQ Backpressure

When VoQ Link is a 1 (in the VoQ CSR), and the stype2 field contains a VoQ Backpressure Symbol, the upstream device may flow control just the queue within the corresponding VC. When linking is enabled, the VoQ backpressure symbol may only be combined with a VC_Status or Status symbol.

Linking only works if both upstream and downstream devices use port staging within VC structures. If either device does not have a corresponding capability, the linking capability shall be disabled in the CSR. A device with queueing within VCs shall flow control corresponding port queues in all VCs when linking is disabled.

When the enable VC linking bit is not set, the VoQ backpressure symbol is not associated with any particular VC, even if the message is happens to be combined with a VC_Status or Status symbol.

Blank page

Chapter 4 Rules

4.1 Implementation Rules

a) Implementation of VoQ backpressure is entirely optional.

b) Devices may implement VoQ backpressure on a port by port basis.

c) Devices may support only the generation of backpressure messages without the ability to honor messages, or vice versa.

d) Backpressure symbols use the stype2 region of the extended control symbol, therefore devices using this method must have the link initialize with the extended control symbol active.

4.2 Rules for Generating Backpressure Control Symbols

a) The VoQ backpressure symbol shall only be transmitted to an upstream device if generation is enabled for a given port. If a congested port requests a symbol be sent to all upstream devices, only ports enabled for this feature shall actually transmit the symbol.

b) The backpressure symbol may be included with any other valid combination of stype 0 and stype 1 symbols if VC linking is not enabled.

c) The backpressure symbol shall only be included with a VC_status or Status control symbol to have the message linked to queues specific to VCs 0 through 8 (VC linking enabled).

d) Ports shall be grouped in order, with the first 6 ports of the device occupying the first port group in the backpressure message (port group 0 encompasses ports 0 - 5, port group 1 covers ports 6 - 11, etc.).

e) The backpressure symbol shall be generated anytime the status of at least one of the ports in the group changes. It is up to the implementer to define what constitutes a status change.

f) The backpressure symbol may be generated at arbitrary intervals, based on a timer. The timer may be the same timer used for VC_status, or it may be a separate timer. Use of a timer is implementation specific.

g) The backpressure symbol may be generated after link recovery to avoid orphaned congestion states.

4.3 Rules for Interpreting Backpressure Control Symbols

a) Devices shall have a mechanism to associate traffic with the downstream device's egress port.

b) Devices shall have a mechanism to associate the incoming congestion message with traffic destined for the indicated port. All traffic identified for that port shall not be committed to a critical resource when that port is identified as congested, allowing other traffic to pass. A critical resource is any resource that can block other traffic like a link or a buffer in a VC.

c) Traffic that is still eligible for transmission is still subject to existing RapidIO ordering rules.

d) Traffic that has been segregated shall be re-introduced to the data stream with the same ordering requirements that existed when it was segregated.

e) Devices may deliberately co-mingle traffic (traffic destined to different ports) to simplify implementations. When such co-mingling loses the ability to further discriminate among the ports, any congestion for any of the ports associated with the co-mingled traffic results in all that traffic being stopped. Co-mingled traffic may only be committed to the link if all ports represented by the traffic are not congested.

Chapter 5 Register Definitions

5.1 VoQ Backpressure Extended Features Block

This section describes the registers for all RapidIO LP-Serial devices supporting virtual channels. This Extended Features register block is assigned Extended Features block ID=0x000B.

5.1.1 Register Map

Table 5-1 shows the VoQ backpressure register map for all RapidIO LP-Serial devices. The Block Offset is the offset relative to the 16-bit Extended Features Pointer (EF_PTR) that points to the beginning of the block.

The address of a byte in the block is calculated by adding the block byte offset to EF_PTR that points to the beginning of the block. This is denoted as [EF_PTR+xx] where xx is the block byte offset in hexadecimal.

Table 5-1. VoQ Register Block

Block Byte Offset	Register Name
0x0	LP-Serial Port - VoQ Backpressure Register Block Header
0xC	Port 0 VoQ Control Register
0x10	Port 1 VoQ Control Register
0x14	Port 2 VoQ Control Register
0x18	Port 3 VoQ Control Register
0x1C-0x304	Port *n* VoQ Control Registers
0x308	Port 191 VoQ Control Register

5.1.2 VoQ Backpressure Control Block Registers

Multiport devices implementing VoQ backpressure shall implement one register per port, even if the port does not support backpressure. Single port end points implement the port 0 register only

5.1.2.1 LP-Serial VC Register Block Header (Block Offset 0x0)

The LP-Serial VC register block header register contains the EF_PTR to the next extended features block and the EF_ID that identifies this as the LP-Serial virtual channel register block header.

Table 5-2. Bit Settings for LP-Serial Register Block Header

Bit	Name	Reset Value	Description
0-15	EF_PTR		Hard wired pointer to the next block in the data structure, if one exists
16-31	EF_ID	0x000B	Hard wired Extended Features ID

5.1.2.2 Port *n* VoQ Control Status Register (Block Offset - Variable, see Section 5.1.1)

This register is used by each port to set up VoQ backpressure operation.

Table 5-3. Port n VoQ Backpressure CSR

Bit	Name	Reset Value	Description
0	VoQ Backpressure Symbol Generation Supported	see footnote[1]	0b0 = generation of VoQ backpressure is not supported by this port 0b1 = generation of VoQ backpressure supported (read-only)
1	VoQ Backpressure Symbol Reception Supported	see footnote[2]	0b0 = reception of VoQ backpressure is not supported by this port 0b1 = reception of VoQ backpressure supported (read-only)
2	Linking with VCs supported	see footnote[3]	0b0 = linking of VoQ backpressure with virtual channels is not supported by this port 0b1 = linking of VoQ backpressure with virtual channels is supported (read-only)
3-7	reserved	0b0	
8	Enable VoQ Symbol Generation	0b0	0b0 = No VoQ symbols will be transmitted 0b1 = VoQ symbol generation is enabled
9	Enable VoQ Participation	0b0	0b0 = this port's status will not be included in any VoQ symbols transmitted, nor cause symbols to be generated. (the port's status will always be reflected as enabled). 0b1 = this port's status will be reflected in VoQ backpressure symbols and will cause symbols to be generated
10	Port XOFF	0b0	0b0 = Port status will reflect current state of the port. 0b1 = Port status will always reflect congested (= 0b1)
11	Enable VC linking	0b0	0b0 = Linking VoQ backpressure with VC Status is disabled 0b1 = Linking VoQ backpressure with VC Status is enabled
12-31	reserved	0x0_0000	

[1]The reset value is implementation dependent

[2]The reset value is implementation dependent

[3]The reset value is implementation dependent

Symbol Generation by a port must be enabled only when the device at the other end of the link supports reception.

VC linking should only be enabled if both connected devices support it. Support is defined as being able to both generate and receive VC linked messages. Either requires an underlying queueing structure that can segregate traffic by both VC and port.

Bits 9 and 10 combine as shown in Table 5-4.

Table 5-4. Port Status Control

Bit 9	Bit 10	Status Reflected in VoQ Backpressure Messages
0	0	Port Status is always 0b0 (will not cause symbol to be generated)
0	1	Port Status is always 0b1 (will not cause symbol to be generated)

Table 5-4. Port Status Control

Bit 9	Bit 10	Status Reflected in VoQ Backpressure Messages
1	0	Normal operation, state transitions cause symbols to be generated and the status is reflected in the symbol
1	1	Port Status is always 0b1 (will cause a symbol to be generated if changing from normal operation causes a state change).

With bit 9 = 0b0, toggling bit 10 will change the ports reported state, but will not trigger any new symbols. With bit 9 = 0b1, changing from normal operation to congested or congested to normal operation will cause a symbol to be transmitted only if the state of the port changed.

Note that changing the status of the port does not necessarily imply traffic will change. That depends on the configuration of the upstream device.