RapidIO™ Interconnect Specification Part 11: Multicast Extensions Specification

Rev. 1.3.1, 06/2005

RapidIO Trade Association

Revision History

Revision	Description	Date
1.3	First release	06/09/2004
1.3.1	Technical changes: the following new features showings: 04-08-00015.003 Converted to ISO-friendly templates	02/23/2005
1.3.1	Removed confidentiality markings for public release	06/07/2005

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

Suite 325, 3925 W. Braker Lane
Austin, TX 78759
512-305-0070 Tel.
512-305-0009 FAX.

Table of Contents

Table of Contents

List of Figures

List of Figures

Blank page

List of Tables

List of Tables

Blank page

Chapter 1 Overview

1.1 Introduction

This chapter provides an overview of the *RapidIO Part 11: Multicast Extensions Specification*. The goal of this specification is to add a simple mechanism to the existing RapidIO specifications that provides multicast functionality to a system. This specification assumes that the reader has a working understanding of the other RapidIO specifications.

1.1 Overview

The concept of duplicating a single message and sending it to multiple selected destinations is known as ‘multicast’, and is found to be useful in many computing systems. This can be accomplished by a variety of means. The most efficient and highest performance method is to have hardware support for the duplication of messages.

Within a RapidIO system, the ability to duplicate messages should scale with the number of end points in a system. Since the number of end points scales with the number of switches in the system, the multicast extensions are defined for switches only and end points are largely unaffected. Possible end point design considerations are described in Annex A.

The multicast specification is limited to request transactions that do not require responses, for example, *RapidIO Part 1: Input/Output Logical Specification* SWRITE transactions. This is because implementing support for collecting the response transactions within a switch device, which are typically not aware of RapidIO logical layer protocols, is problematic and complex.

The ability for a switch to send a single message to a variety of destinations can be implemented in a wide variety of ways, depending on system needs. There are two reasons, however, that motivate definition of a common interface and behavior for multicast in a system. Without a standard interface and behavioral definition, the wide variety of possible implementations would not allow a common multicast software driver to exist. The second reason is that without a standard definition for interface and behavior it is impossible to guarantee inter-operability of different components which support multicast.

In defining a common interface for a wide variety of implementations, it is necessary

to define the standard interface with some level of abstraction in order to avoid limiting implementation flexibility. Therefore, several examples of the use of the interface have been included.

1.2 Requirements

The multicast mechanism shall fulfill the following goals:

- Simple - excess complexity will not gain acceptance
- Compact - Does not cost excessive silicon area in a switch
- Robust - same level of protection and recovery as the rest of RapidIO
- Scalable - must be able to extend to multi-layer switch systems
- Compatibility with all physical layers

Chapter 2 Multicast Extensions Behavior

2.1 Introduction

This chapter describes the multicast extensions rules of operation in a RapidIO system. A RapidIO switch which does not support multicast can co-exist in a RapidIO fabric with other switches that do support multicast. The only requirement is that the switch be capable of routing the destination IDs used for multicast transactions.

2.2 Packet Replication

A RapidIO multicast operation consists of the replication of a single packet so that it can be received by multiple end points. This replication is performed by the switch devices in the fabric rather than by the end point itself, so that the capability to replicate packets expands with the number of switches (and hence possible end points) in a system. Each switch may be individually programmed to control which egress ports of the switch the replicated packets are sent to, and thus indirectly which specific set of end point devices receive the replicated packet. The packets themselves are not modified by the replication process, merely transmitted out through the appropriate ports.

This specification only addresses multicasting request packets for transactions which do not require responses. This greatly simplifies multicast support for RapidIO switches, which will therefore have no need to aggregate responses from other types of RapidIO operations. Examples of transactions which can be multicast are I/O logical specification NWRITE and SWRITE transactions. Multicasting transactions which require responses have implementation defined behavior.

2.3 Multicast Operation

Multicast operations have two control value types - **multicast masks** and **multicast groups**. The set of target end points which all receive a particular multicast packet is known as a **multicast group**. Each multicast group is associated with a unique destination ID. The destination ID of a received packet allows a RapidIO switch device to determine that a packet is to be replicated for a multicast.

A **multicast mask** is a value that controls which egress ports one or more multicast groups are associated with. Conceptually, a multicast mask is a register with one

enable bit for each possible switch egress port. There is one set of multicast masks for the entire switch. All multicast masks in a switch are assigned unique sequential ID numbers beginning with 0. Figure 2-1 shows an example of the use of multicast in a RapidIO system.

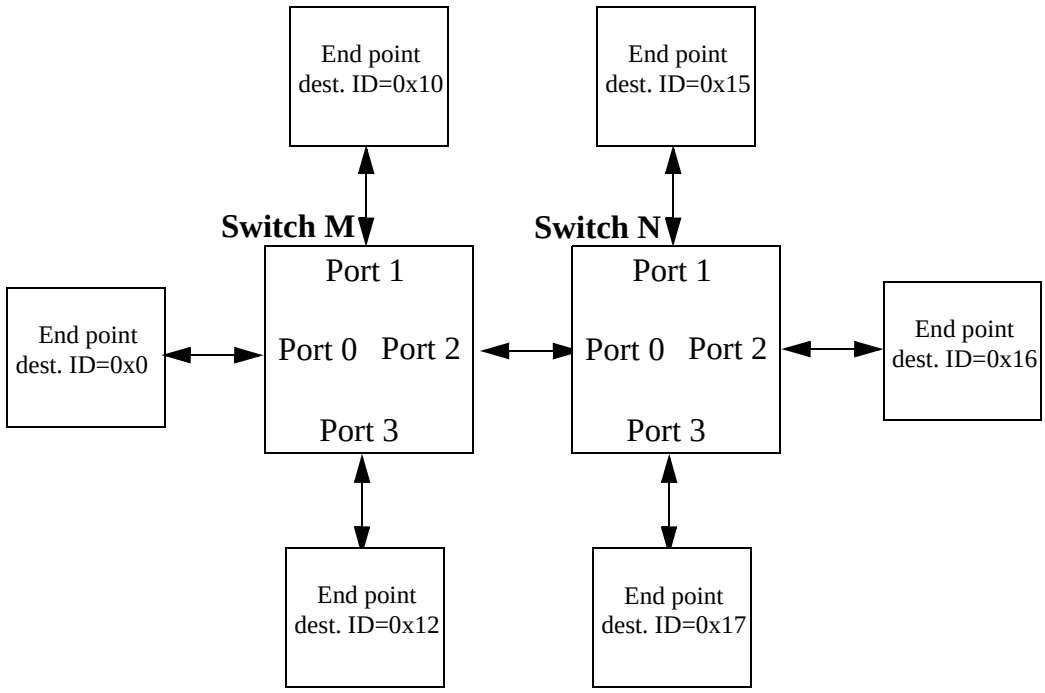

Figure 2-1. Multicast System Example

In this example, the end point assigned destination ID 0x0 uses destination ID 0x80 to perform multicast operations to the multicast group comprised of end points 0x10, 0x15, 0x16, and 0x17, arbitrarily called group A. Software configures the switch devices in the fabric to **associate** the destination IDs that represent multicast groups with multicast masks. For Figure 2-1 switch M associates destination ID 0x80 with egress ports 1 and 2, and switch N associates destination ID 0x80 with ports 1, 2, and 3. Figure 2-2 shows a possible relationship between the multicast group, the multicast masks for the switches, and the global system address map.

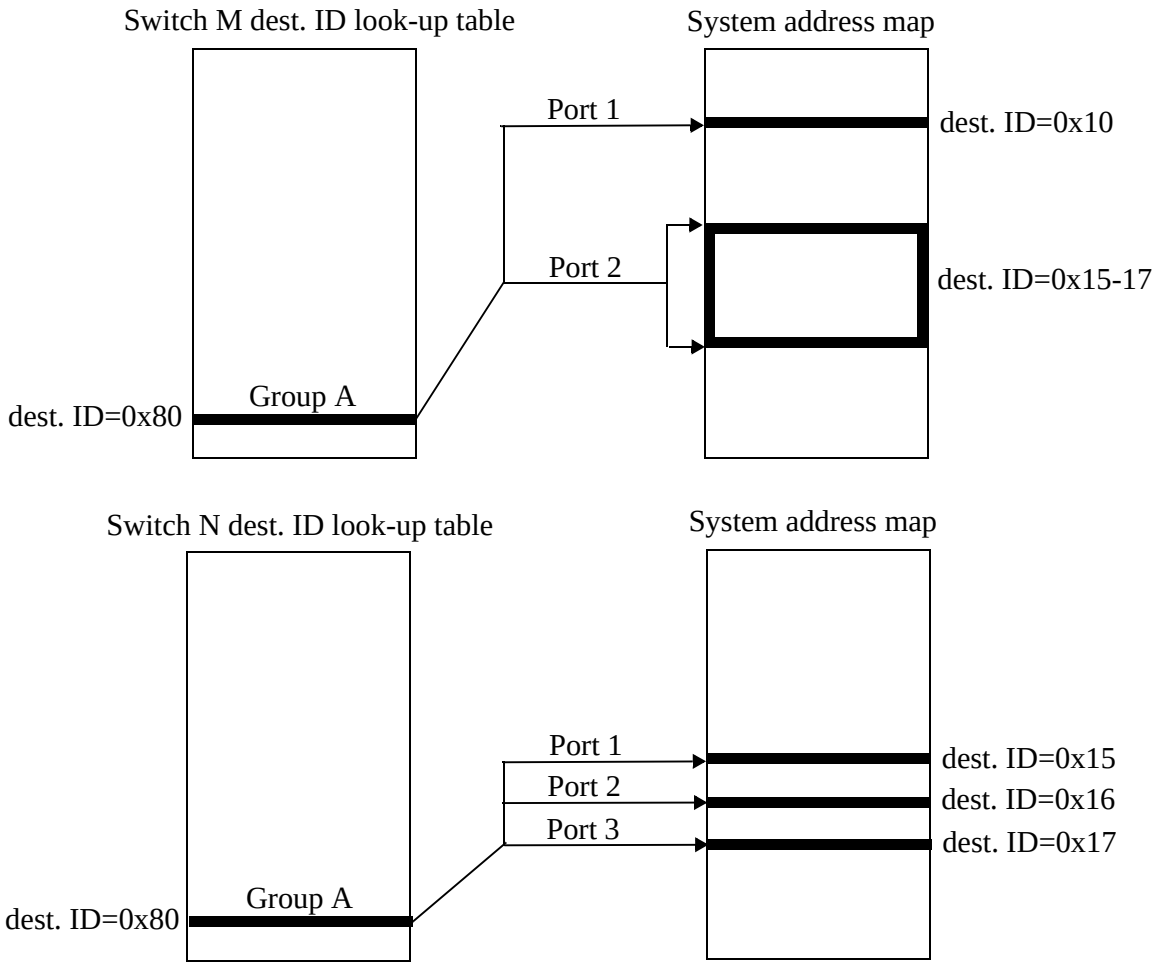

Figure 2-2. Multicast Association Example

Configuring a RapidIO switch to replicate packets for a multicast group is a two-step process. First, a list of egress ports is set in a multicast mask list. Second, one or more destination IDs which represent the multicast groups are associated with the multicast mask in the switch. During normal system operation, any time a switch receives a packet with a destination ID which has been associated with a multicast mask it will send that packet to all egress ports enabled by that multicast mask.

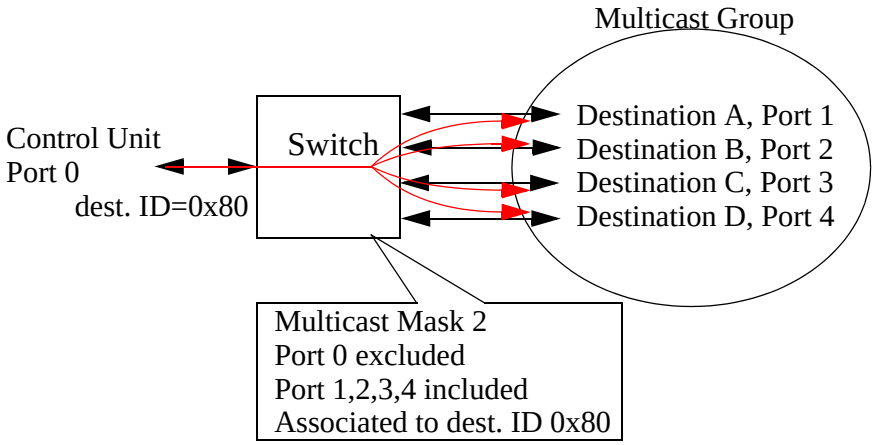

Figure 2-3. Multicast Configuration Example

Figure 2-3 shows a control unit connected to switch port 0 which needs to multicast to destinations A, B, C and D. A multicast mask, in this case arbitrarily picked as multicast mask 2, is set up to select which ports in the switch are part of the multicast group of destinations A, B, C, and D. A destination ID, in this case arbitrarily assigned 0x80, is associated with multicast mask 2 as the destination ID that the control unit should use to multicast to the multicast group. The associate operation is done using the CSRs defined in Chapter 3, “Multicast Extensions Registers”.

The defined CSRs allow a switch to associate destination IDs with multicast masks using a small number of maintenance write operations. The number of unique destination IDs that can be associated with a multicast mask is also defined in a CSR.

While each destination ID is associated with a unique multicast group, the programming model allows a destination ID to be mapped to a different multicast mask for each port on the switch. However, for each port a destination ID can be associated with at most one multicast mask. The last association operation performed for a specific port and destination ID dictates which multicast mask the destination ID is associated with. It is also possible to map a given destination ID to the same multicast mask for all ports.

A RapidIO switch may be capable of supporting large numbers of multicast groups by dedicating a sequential range of destination ID’s to an equal number of sequentially numbered multicast masks. A switch may also be designed which does not require all multicast destination IDs to be sequential. The programming model supports both of these implementations.

A packet will never be multicast back out of the port it was received on even if it is included in the multicast mask for that destination ID. This allows a group of end points which need to multicast to each other to share the same multicast mask. Packets using a multicast mask which has no egress ports selected will be dropped without error notification. A device may have implementation specific error notification in this situation, depending on system requirements.

The default state after a reset for multicast masks is that all multicast masks have no ports selected. Additionally, after reset no associations exist between any multicast group/destination ID and the multicast masks. However, implementation specific capabilities may modify the multicast mask values and associations after reset without software intervention.

For more information and examples on the use of the programming model for multicast refer to Annex B, “Multicast Applications (Informative)”.

2.4 Multicast Transaction Ordering Requirements

RapidIO packets which are in the same multicast group (the same destination ID) with the same flowID and are received on the same ingress port must be multicast on the egress ports in the same order that they were received. There are no ordering requirements between multicast packets and non-multicast packets, or between multicast packets in different multicast groups. Maintaining ordering between transactions in the same transaction request flow for a multicast group allows an application to multicast a completion flag at the end of a potentially large data transfer which was sent to the same multicast group.

Blank page

Chapter 3 Multicast Extensions Registers

3.1 Introduction

This section describes the Multicast Extensions CAR and CSR registers that allow an external processing element to determine if a switch supports the multicast extensions defined in this specification, and to manage the configuration of multicast groups for a switch processing element. This chapter only describes registers or register bits defined by this specification. Refer to the other RapidIO logical, transport, physical, and extension specifications of interest to determine a complete list of registers and bit definitions for a device. All registers are 32-bits and aligned to a 32-bit boundary. The behavior of reserved register bits and register offsets and access rules and requirements are described in the *RapidIO Part 1: Input/Output Logical Specification*.

Table 3-1. Multicast Register Map

Configuration Space Byte Offset	Register Name
0x0-C	Reserved
0x10	Processing Element Features CAR
0x14-2C	Reserved
0x30	Switch Multicast Support CAR
0x34	Reserved
0x38	Switch Multicast Information CAR
0x3C-7C	Reserved
0x80	Multicast Mask Port CSR
0x84	Multicast Associate Select CSR
0x88	Multicast Associate Operation CSR
0x8C–FC	Reserved
0x100–FFFC	Extended Features Space
0x10000–FFFFFC	Implementation-defined Space

3.2 Processing Elements Features CAR (Configuration Space Offset 0x10)

The Processing Elements Features CAR contains 31 processing elements features bits defined in various RapidIO specifications, as well as the Multicast Support bit, defined here.

Table 3-2. Bit Settings for Processing Elements Features CAR

Bit	Name	Reset Value	Description
0-20	-		Reserved (defined elsewhere)
21	Multicast Support	*	Support for multicast extensions 0b0 - Does not support multicast extensions 0b1 - Supports multicast extensions
22-31	-		Reserved (defined elsewhere)

* Implementation dependant

3.3 Switch Multicast Support CAR (Configuration Space Offset 0x30)

The Switch Multicast Support CAR defines support for a simple multicast model and the additional limits on multicast mask resources.

Table 3-3. Bit Settings for Switch Multicast Support CAR

Bit	Name	Reset Value	Description
0	Simple_Assoc	*	Support for a simple multicast association model 0b0 - Does not support simple association 0b1 - Supports simple association If this bit is set, the Block_Assoc bit in the Switch Multicast Information CAR must also be set.
1-31	-		Reserved (defined elsewhere)

* Implementation dependant

3.4 Switch Multicast Information CAR (Configuration Space Offset 0x38)

The Switch Multicast Information CAR defines the methods for associating destination IDs with multicast masks supported by a RapidIO switch device. It also defines the limits on multicast mask resources.

Table 3-4. Bit Settings for Switch Multicast Information CAR

Bits	Name	Description
0	Block_Assoc	Block association support - allows equal sized blocks of destination IDs and multicast masks to be associated with each other with a single operation rather than one at a time. 0b0 - block association is not supported 0b1 - block association is supported If the Simple_Assoc bit in the Switch Multicast Support CAR is set, this bit must also be set.
1	Per_Port_Assoc	Per ingress port association support - allows a destination ID to be associated with a multicast mask on a per-ingress port basis rather than a single association for the entire switch. 0b0 - per port association is not supported 0b1 - per port association is supported
2-15	MaxDestIDAssoc	The maximum number of destination IDs associations per multicast mask 0x0000 - 1 destination ID 0x0001 - 2 destination IDs ... 0x3FFF - 16384 destination IDs
16-31	MaxMcastMasks	The number of multicast egress port masks available. This field also defines the largest block of destination IDs that can be block associated. 0x0000 - [reserved] 0x0001 - 1 multicast mask 0x0002 - 2 multicast masks ... 0xFFFF - 65535 multicast masks

3.5 Multicast Mask Port CSR (Configuration Space Offset 0x80)

The Multicast Mask Port CSR allows configuration of the egress port list for each of the switch's multicast masks.

Writing the Write_to_Verify command sets up a Mcast_Mask and Egress_Port_Num pair to verify. The presence of the specified egress port in the specified multicast mask is indicated by the Port_Present bit on a subsequent read of the register.

Writing the Add_Port or Delete_Port commands adds or deletes the specified egress port to or from the specified multicast mask.

Writing the Add_All_Ports or Delete_All_Ports commands adds or deletes all of the egress ports in the specified multicast mask.

The result of illegal values or combinations for an operation is implementation dependent. For examples of how to use this register, refer to Section 4.2, "Configuring Multicast Masks**".**

Table 3-5. Bit Settings for Multicast Mask Port CSR

Bits	Name	Reset Value	Description
0-15	Mcast_Mask	0x0000	Specifies the multicast mask which is to be modified or queried as determined by the Mask_Cmd field.
16-23	Egress_Port_Num	0x00	Specifies the port number to be added, deleted, or queried with the Mask_Cmd field.
24	-	0b0	Reserved
25-27	Mask_Cmd	0b000	Specifies the mask action on a write. 0b000 - Write_to_Verify 0b001 - Add_Port 0b010 - Delete_Port 0b011 - reserved 0b100 - Delete_All_Ports 0b101 - Add_All_Ports 0b110 - reserved 0b111 - reserved
28–30	-	0b000	Reserved
31	Port_Present	0b0	Indicates the existence of the egress port and multicast mask pair as a result of the last preceding Write_to_Verify command. 0b0 - Port was not enabled as an egress port in the specified multicast mask 0b1 - Port was enabled as an egress port in the specified multicast mask. This bit is reserved on a write.

3.6 Multicast Associate Select CSR (Configuration Space Offset 0x84)

This register specifies the destination ID and multicast mask number for a subsequent associate operation controlled with the Multicast Associate Operation CSR. If block association is supported, this register specifies the start of the block to associate. For examples of how this register is used, refer to Section 4.4, “Configuring Associations”.

Table 3-6. Bit Settings for Multicast Associate Select CSR

Bits	Name	Reset Value	Description
0-7	Large_DestID	0x00	Selects the most significant byte of a large transport destination ID for an association operation
8-15	DestID	0x00	Selects the destination ID for an association operation
16-31	Mcast_Mask_Num	0x0000	Selects the multicast mask number for an association operation

3.7 Multicast Associate Operation CSR (Configuration Space Offset 0x88)

The Multicast Associate Operation CSR specifies three operations for associating destination IDs with multicast masks. The affected destination ID and multicast mask is specified in the Multicast Associate Select CSR. The specified operation is executed when this register is written. When this register is read and the Assoc_Cmd field it set to Write_to_Verify the specified operation is executed and the updated register state is returned. If this register is read and the Assoc_Cmd field is not set to Write_to_Verify the resulting behavior is implementation dependent. Block association operations assign associations sequentially starting with the destination ID and multicast mask specified in the Multicast Associate Select CSR.

Writing the Write_To_Verify command checks for an association between the destination ID and multicast mask specified in the Multicast Associate Select CSR. The result of the check is indicated by the state of the Assoc_Present bit on a read of this register. This command cannot be executed on a block.

Writing the Add_Assoc or Delete_Assoc command adds or deletes the association between the destination ID and the multicast mask (or block of associations, if block association is supported) specified in the Multicast Associate Select CSR.

The result of illegal values or field combinations for an association operation is implementation dependent. For examples of how this register is used, refer to Section 4.4, “Configuring Associations.

Table 3-7. Bit Settings for Multicast Associate Operation CSR

Bits	Name	Reset Value	Description
0-15	Assoc_Blksize	0x0000	This field specifies the number of sequential DestinationIDs to be associated with an equal number of sequential multicast mask numbers if block association is supported. This field is ignored on a Write_to_Verify command. 0x0000 - one association 0x0001 - two sequential associations ... 0xFFFF - 65536 sequential associations
16-23	Ingress_Port	0x00	This field specifies the ingress port association to affect if per-port ingress association is supported
24	Large_Transport	0b0	0b0 - the association is for small transport destination IDs 0b1 - the association is for large transport destination IDs
25-26	Assoc_Cmd	0b00	This field specifies the command to execute when this register is written. 0b00 - Write_To_Verify 0b01 - reserved 0b10 - Delete_Assoc 0b11 - Add_Assoc

Table 3-7. Bit Settings for Multicast Associate Operation CSR (Continued)

Bits	Name	Reset Value	Description
27-30	-	0b0000	reserved
31	Assoc_Present	0b0	This bit contains the result of the last Write_to_Verify command executed. 0b0 - no association present 0b1 - association present This bit is reserved on write.

Chapter 4 Configuration Examples

4.1 Introduction

This chapter provides several examples of how to use the multicast programming interface. The given examples build upon each other while proceeding through the sections. References to the order of operations within the examples run from the top of a list to the bottom unless otherwise stated.

Initially assume a switch with 8 ports which supports 4 or more multicast masks with two or more destination IDs allowed per multicast group so that a total of 8 destination IDs minimum can be associated with the multicast masks. The system has the following requirements:

- Three sources of traffic (ports 0, 1, and 2) must be multicast to two destinations (ports 6 and 7).
- Three ports (ports 3, 4 and 5) need to multicast signals between each other.
- All ports occasionally need to multicast to every other port.

Assume that the switch does not require any other multicast functions and therefore multicast masks 0, 1, and 2 will be used.

4.2 Configuring Multicast Masks

This section discusses assigning an egress port list to a multicast mask.

4.2.1 Clearing Multicast Masks

Suppose that the state of the multicast masks is unknown, and therefore the masks must be cleared before being configured. In order to clear the masks the following register accesses are made. (The accesses to the Multicast Mask Port CSR can be performed in any order.)

- Remove all ports from multicast mask 0
  - — Write the value 0x0000_0040 to the Multicast Mask Port CSR
- Remove all ports from multicast mask 1
  - — Write the value 0x0001_0040 to the Multicast Mask Port CSR
- Remove all ports from multicast mask 2
  - — Write the value 0x0002_0040 to the Multicast Mask Port CSR

4.2.2 Assigning Ports to Multicast Masks

To configure mask 0 to multicast to ports 6 and 7, mask 1 to multicast to ports 3, 4 and 5, and mask 2 to multicast to every port, requires the following series of register accesses. (The accesses to the Multicast Mask Port CSR can be performed in any order.)

- Add port 6 to multicast mask 0
  - — Write the value 0x0000_0610 to the Multicast Mask Port CSR
- Add port 7 to multicast mask 0
  - — Write the value 0x0000_0710 to the Multicast Mask Port CSR
- Add port 3 to multicast mask 1
  - — Write the value 0x0001_0310 to the Multicast Mask Port CSR
- Add port 4 to multicast mask 1
  - — Write the value 0x0001_0410 to the Multicast Mask Port CSR
- Add port 5 to multicast mask 1
  - — Write the value 0x0001_0510 to the Multicast Mask Port CSR
- Add all ports to multicast mask 2
  - — Write the value 0x0002_0050 to the Multicast Mask Port CSR

4.2.3 Removing a Port from a Multicast Mask

Suppose that the device attached to port 4 needs to be removed from the system. The following register accesses are used to modify multicast masks 1 and 2 to stop port 4 from being a multicast destination. (The accesses may be performed in any order.)

- Remove port 4 from multicast mask 1
  - — Write the value 0x0001_0420 to the Multicast Mask Port CSR
- Remove port 4 from multicast mask 2
  - — Write the value 0x0002_0420 to the Multicast Mask Port CSR

4.2.4 Querying a Multicast Mask

In this section suppose that a system designer needs to determine which of the 8 ports are included in multicast mask 2. The following accesses are to be performed to provide this information. (In each case, the write operation setting up the ‘Write to Verify’ operation must be performed before the subsequent read to check the Port Present bit status. The individual multicast masks may be queried in any order.)

- Verify that port 0 is included in mask 2
  - — Write the value 0x0002_0000 to the Multicast Mask Port CSR
  - — Read the value 0x0002_0001 from the Multicast Mask Port CSR

- Verify that port 1 is included in mask 2
  - — Write the value 0x0002_0100 to the Multicast Mask Port CSR
  - — Read the value 0x0002_0101 from the Multicast Mask Port CSR.
- Verify that port 2 is included in mask 2
  - — Write the value 0x0002_0200 to the Multicast Mask Port CSR
  - — Read the value 0x0002_0201 from the Multicast Mask Port CSR
- Verify that port 3 is included in mask 2
  - — Write the value 0x0002_0300 to the Multicast Mask Port CSR
  - — Read the value 0x0002_0301 from the Multicast Mask Port CSR
- Verify that port 4 is not included in mask 2
  - — Write the value 0x0002_0400 to the Multicast Mask Port CSR
  - — Read the value 0x0002_0400 from the Multicast Mask Port CSR
- Verify that port 5 is included in mask 2
  - — Write the value 0x0002_0500 to the Multicast Mask Port CSR
  - — Read the value 0x0002_0501 from the Multicast Mask Port CSR
- Verify that port 6 is included in mask 2
  - — Write the value 0x0002_0600 to the Multicast Mask Port CSR
  - — Read the value 0x0002_0601 from the Multicast Mask Port CSR
- Verify that port 7 is included in mask 2
  - — Write the value 0x0002_0700 to the Multicast Mask Port CSR
  - — Read the value 0x0002_0701 from the Multicast Mask Port CSR

4.3 Simple Association

If the Simple_Assoc bit is set in the Switch Multicast Support CAR, the device supports the simple multicast programming model. This model allows for basic multicast support for devices with a limited number of multicast masks, and requires a fixed relationship between those masks and sequential multicast groups.

4.3.1 Restrictions on Block Size

If the Simple_Assoc bit is set the device has a limited number of masks. Therefore, the number of sequential associations equals the maximum number of masks.

The Assoc_BlkSize field in the Multicast Associate Operation CSR must be set to the value of (MaxMCastMasks - 1). The MaxMCastMasks field is in the Switch Multicast Information CAR.

4.3.2 Restrictions on Block Associate

If the Simple_Assoc bit is set, non-block associations are precluded.

4.3.3 Restrictions on Associations

If the Simple_Assoc bit is set the device requires a fixed relationship between the sequential mask numbers and sequential destination IDs. This must be taken into account when the masks are associated.

The Multicast Associate Select CSR is set with the Mcast_Mask_num value set to 0x0000 and the Large_DestID and DestID fields set to an integer multiple of the MaxMCastMasks value.

Hardware that sets the new Simple_Assoc CAR bit could implement a single block associate for all of the masks that it supports with the requirement that they all be sequential destination IDs.

4.4 Configuring Associations

This section describes how to associate destination IDs with multicast masks, including examples of how to use the block association and per-port association functions.

4.4.1 Basic Association

For the assumed system it is now necessary to associate a destination ID with each multicast mask from the preceding examples. How this can be accomplished may vary depending on the capabilities of the switch. For this section, assume that neither block association nor per-ingress-port association is supported by the switch.

Following upon the previous example, assume the following additional system requirements.

- the 16 bit destination ID 0x1234 needs to be associated with multicast mask 0.
- the 8 bit destination ID 0x44 needs to be associated with multicast mask 1.
- the 16 bit destination ID 0xFEED needs to be associated with multicast mask 2.

In order to accomplished the desired associations, the following register accesses are required. (The individual association operations can be performed in any order.)

- Set up the operation to associate destination ID 0x1234 with multicast mask 0
  - — Write the value 0x1234_0000 to the Multicast Associate Select CSR
- Associate destination ID 0x1234 with multicast mask 0
  - — Write the value 0x0000_00E0 to the Multicast Associate Operation CSR
- Set up the operation to associate destination ID 0x44 with multicast mask 1

— Write the value 0x0044_0001 to the Multicast Associate Select CSR

- Associate destination ID 0x44 with multicast mask 1

  — Write the value 0x0000_0060 to the Multicast Associate Operation CSR

- Set up the operation to associate destination ID 0xFEED with multicast mask 2

  — Write the value 0xFEED_0002 to the Multicast Associate Select CSR

- Associate destination ID 0xFEED with multicast mask 2

  — Write the value 0x0000_00E0 to the Multicast Associate Operation CSR

4.4.2 Using Per-Ingress Port Association

For the associations discussed in the preceding section, if the switch supports per-ingress-port association (destination IDs are associated with multicast masks on a per ingress port basis), the required programming operations change. The associations for each multicast mask are grouped into a write to the Multicast Associate Select CSR, followed by a write to the Multicast Associate Operation CSR for each ingress port that must be aware of the association. (The writes to the Multicast Associate Operation CSR can occur in any order but must occur after the related writes to the Multicast Associate Select CSR. The individual association operations can be performed in any order.)

- Set up the operation to associate destination ID 0x1234 with multicast mask 0

  — Write the value 0x1234_0000 to the Multicast Associate Select CSR

- Associate destination ID 0x1234 with multicast mask 0 on ingress port 0

  — Write the value 0x0000_00E0 to the Multicast Associate Operation CSR

- Associate destination ID 0x1234 with multicast mask 0 on ingress port 1

  — Write the value 0x0000_01E0 to the Multicast Associate Operation CSR

- Associate destination ID 0x1234 with multicast mask 0 on ingress port 2

  — Write the value 0x0000_02E0 to the Multicast Associate Operation CSR

- Set up the operation to associate destination ID 0x44 with multicast mask 1

  — Write the value 0x0044_0001 to the Multicast Associate Select CSR

- Associate destination ID 0x44 with multicast mask 1 on ingress port 3

  — Write the value 0x0000_0360 to the Multicast Associate Operation CSR

- Associate destination ID 0x44 with multicast mask 1 on ingress port 4

  — Write the value 0x0000_0460 to the Multicast Associate Operation CSR

- Associate destination ID 0x44 with multicast mask 1 on ingress port 5

  — Write the value 0x0000_0560 to the Multicast Associate Operation CSR

- Set up the operation to associate destination ID 0xFEED with multicast mask 2

  — Write the value 0xFEED_0002 to the Multicast Associate Select CSR

- Associate destination ID 0xFEED with multicast mask 2 on ingress port 0

— Write the value 0x0000_00E0 to the Multicast Associate Operation CSR

- Associate destination ID 0xFEED with multicast mask 2 on ingress port 1

  — Write the value 0x0000_01E0 to the Multicast Associate Operation CSR

- Associate destination ID 0xFEED with multicast mask 2 on ingress port 2

  — Write the value 0x0000_02E0 to the Multicast Associate Operation CSR

- Associate destination ID 0xFEED with multicast mask 2 on ingress port 3

  — Write the value 0x0000_03E0 to the Multicast Associate Operation CSR

- Associate destination ID 0xFEED with multicast mask 2 on ingress port 4

  — Write the value 0x0000_04E0 to the Multicast Associate Operation CSR

- Associate destination ID 0xFEED with multicast mask 2 on ingress port 5

  — Write the value 0x0000_05E0 to the Multicast Associate Operation CSR

- Associate destination ID 0xFEED with multicast mask 2 on ingress port 6

  — Write the value 0x0000_06E0 to the Multicast Associate Operation CSR

- Associate destination ID 0xFEED with multicast mask 2 on ingress port 7

  — Write the value 0x0000_07E0 to the Multicast Associate Operation CSR

4.4.3 Using Block Association

In this section assume that the switch supports block association rather than per-ingress-port association. With this feature sequential destination IDs can be quickly associated to sequential multicast masks. In order to take advantage of this feature, different destination IDs assignments are required for the system than for the preceding examples. The starting destination 0xFF00 is arbitrarily selected.

- the 16 bit destination ID 0xFF00 is used to multicast from ports 0, 1 and 2 to ports 6 and 7, so destination ID 0xFF00 needs to be associated with multicast mask 0.
- the 16 bit destination ID 0xFF01 identifies the multicast group including ports 3, 4 and 5, so destination ID 0xFF01 needs to be associated with multicast mask 1.
- the 16 bit destination ID 0xFF02 identifies the multicast group that includes all ports, so destination ID 0xFF02 needs to be associated with multicast mask 2.

Note that the number of accesses needed to accomplish the desired associations is reduced to two. (The accesses must be performed in the order given.)

- Set up the associate operation starting with destination ID 0xFF00 and multicast mask 0

  — Write the value 0xFF00_0000 to the Multicast Associate Select CSR

- Associate three sequential destination IDs starting at 0xFF00 with three sequential multicast masks starting at 0

  — Write the value 0x0002_00E0 to the Multicast Associate Operation CSR

4.4.4 Using Per-Ingress Port and Block Association

Next, if both block association and per-ingress port association are supported by the switch, then the following sequence of operations is required. (The write to the Multicast Associate Select CSR must occur before the corresponding write to the Multicast Associate Operation CSR. The individual association operations can be performed in any order.)

- Set up the associate operations starting with destination ID 0xFF00 and multicast mask 0
  - — Write the value 0xFF00_0000 to the Multicast Associate Select CSR
- Associate three sequential destination IDs with three sequential multicast masks for ingress port 0
  - — Write the value 0x0002_00E0 to the Multicast Associate Operation CSR
- Associate three sequential destination IDs with three sequential multicast masks for ingress port 1
  - — Write the value 0x0002_01E0 to the Multicast Associate Operation CSR
- Associate three sequential destination IDs with three sequential multicast masks for ingress port 2
  - — Write the value 0x0002_02E0 to the Multicast Associate Operation CSR
- Associate three sequential destination IDs with three sequential multicast masks for ingress port 3
  - — Write the value 0x0002_03E0 to the Multicast Associate Operation CSR
- Associate three sequential destination IDs with three sequential multicast masks for ingress port 4
  - — Write the value 0x0002_04E0 to the Multicast Associate Operation CSR
- Associate three sequential destination IDs with three sequential multicast masks for ingress port 5
  - — Write the value 0x0002_05E0 to the Multicast Associate Operation CSR
- Associate three sequential destination IDs with three sequential multicast masks for ingress port 6
  - — Write the value 0x0002_06E0 to the Multicast Associate Operation CSR
- Associate three sequential destination IDs with three sequential multicast masks for ingress port 7
  - — Write the value 0x0002_07E0 to the Multicast Associate Operation CSR

For this example, suppose that ingress port 4 needs a second destination ID to be mapped to each of the three multicast masks and the switch also has this capability. The second destination would be added to port 4 with the following association operation. (The write to the Multicast Associate Select CSR must occur before the write to the Multicast Associate Operation CSR.

- Set up the associate operations starting with destination ID 0xFF03 and multicast mask 0
  - — Write the value 0xFF03_0000 to the Multicast Associate Select CSR
- Associate three sequential destination IDs with three sequential multicast masks for ingress port 4
  - — Write the value 0x0002_04E0 to the Multicast Associate Operation CSR

4.4.5 Removing a Destination ID to Multicast Mask Association

Now assume that packets from destination ID 0xFF02 on port 4 should no longer be allowed to multicast to all nodes (multicast mask 2). To remove destination ID 0xFF02 from being associated with multicast mask 2 on port 4, the following register accesses need to be performed in order.

- Set up the operation to remove the association between destination ID 0xFF02 and multicast mask 2
  - — Write the value 0xFF02_0002 to the Multicast Associate Select CSR
- Remove the association between destination ID 0xFF02 and multicast mask 2 on ingress port 4
  - — Write the value 0x0000_04C0 to the Multicast Associate Operation CSR

4.4.6 Querying an Association

There are three scenarios for querying destination ID to multicast mask associations in a switch. For the first scenario assume that a system designer wants to know which multicast masks are associated with destination ID 0xFF01 on port 4. Note that since a read of the Multicast Associate Operation CSR causes the last command written to be executed, that register is only written at the beginning of the sequence. (The individual associations can be queried in any order.)

- Set up the associate operations for destination ID 0xFF01 and multicast mask 0
  - — Write the value 0xFF01_0000 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF01 is not associated with multicast mask 0 for port 4
  - — Write the value 0x0000_0480 to the Multicast Associate Operation CSR
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR
- Set up the associate operations for destination ID 0xFF01 and multicast mask 1
  - — Write the value 0xFF01_0001 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF01 is not associated with multicast mask 1 for port 4
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR

- Set up the associate operations for destination ID 0xFF01 and multicast mask 2
  - — Write the value 0xFF01_0002 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF01 is associated with multicast mask 2 for port 4
  - — Read the value 0x0000_0481 from the Multicast Associate Operation CSR
- Set up the associate operations for destination ID 0xFF01 and multicast mask 3
  - — Write the value 0xFF01_0003 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF01 is not associated with multicast mask 3 for port 4
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR

For the second scenario assume that the system designer wants to know which destination IDs from 0xFF00 through 0xFF07 are associated with multicast mask 0 on Port 4. (The individual associations may be queried in any order.)

- Set up the associate operations for destination ID 0xFF00 and multicast mask 0
  - — Write the value 0xFF00_0000 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF00 is associated with multicast mask 0 for port 4
  - — Write the value 0x0000_0480 to the Multicast Associate Operation CSR
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR
- Set up the associate operations for destination ID 0xFF00 and multicast mask 0
  - — Write the value 0xFF01_0000 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF01 is not associated with multicast mask 0 for port 4
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR
- Set up the associate operations for destination ID 0xFF00 and multicast mask 0
  - — Write the value 0xFF02_0000 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF02 is not associated with multicast mask 0 for port 4
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR
- Set up the associate operations for destination ID 0xFF00 and multicast mask 0
  - — Write the value 0xFF03_0000 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF03 is not associated with multicast mask 0 for port 4
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR

- Set up the associate operations for destination ID 0xFF00 and multicast mask 0
  - — Write the value 0xFF04_0000 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF04 is not associated with multicast mask 0 for port 4
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR
- Set up the associate operations for destination ID 0xFF00 and multicast mask 0
  - — Write the value 0xFF05_0000 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF05 is associated with multicast mask 0 for port 4
  - — Read the value 0x0000_0481 from the Multicast Associate Operation CSR
- Set up the associate operations for destination ID 0xFF00 and multicast mask 0
  - — Write the value 0xFF06_0000 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF06 is not associated with multicast mask 0 for port 4
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR
- Set up the associate operations for destination ID 0xFF00 and multicast mask 0
  - — Write the value 0xFF07_0000 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF07 is not associated with multicast mask 0 for port 4
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR

For the last scenario assume that the system designer now wants to know whether or not destination ID 0xFF03 is mapped to multicast mask 3 on all ports. (The individual associations may be queried in any order.)

- Set up the associate operations for destination ID 0xFF03 and multicast mask 3
  - — Write the value 0xFF03_0003 to the Multicast Associate Select CSR
- Verify that destination ID 0xFF03 is not associated with multicast mask 3 for port 0
  - — Write the value 0x0000_0080 to the Multicast Associate Operation CSR
  - — Read the value 0x0000_0080 from the Multicast Associate Operation CSR
- Verify that destination ID 0xFF03 is not associated with multicast mask 3 for port 1
  - — Write the value 0x0000_0180 to the Multicast Associate Operation CSR
  - — Read the value 0x0000_0180 from the Multicast Associate Operation CSR

- Verify that destination ID 0xFF03 is not associated with multicast mask 3 for port 2
  - — Write the value 0x0000_0280 to the Multicast Associate Operation CSR
  - — Read the value 0x0000_0280 from the Multicast Associate Operation CSR
- Verify that destination ID 0xFF03 is associated with multicast mask 3 for port 3
  - — Write the value 0x0000_0380 to the Multicast Associate Operation CSR
  - — Read the value 0x0000_0381 from the Multicast Associate Operation CSR
- Verify that destination ID 0xFF03 is not associated with multicast mask 3 for port 4
  - — Write the value 0x0000_0480 to the Multicast Associate Operation CSR
  - — Read the value 0x0000_0480 from the Multicast Associate Operation CSR
- Verify that destination ID 0xFF03 is not associated with multicast mask 3 for port 5
  - — Write the value 0x0000_0580 to the Multicast Associate Operation CSR
  - — Read the value 0x0000_0580 from the Multicast Associate Operation CSR
- Verify that destination ID 0xFF03 is not associated with multicast mask 3 for port 6
  - — Write the value 0x0000_0680 to the Multicast Associate Operation CSR
  - — Read the value 0x0000_0680 from the Multicast Associate Operation CSR
- Verify that destination ID 0xFF03 is not associated with multicast mask 3 for port 7
  - — Write the value 0x0000_0780 to the Multicast Associate Operation CSR
  - — Read the value 0x0000_0780 from the Multicast Associate Operation CSR

Blank page

Annex A End Point Considerations (Informative)

A.1 Introduction

This appendix provides implementation considerations for end points that are intended to be used in a multicast RapidIO system.

A.2 Multicast Destination ID

If an end point does validation of the destination ID of a received packet against it's own deviceID or IDs, then that end point must be able to disable the comparison or have a deviceID assignment that allows validation of the multicast packet.

A.3 End Point Multicast Channels

It may be valuable for an end point to have support for one or more multicast channels. Multicast channels are address ranges in RapidIO address space for which an end point may accept a multicast packet and possibly translate the RapidIO write address to another local address region. This is necessary if the recipient of a multicast transaction does not have valid address space at the address received. The size and quantity of multicast channels depend on the requirements of the application. It may also be necessary to link multicast channels to particular multicast groups.

A multicast channel valid bit can be implemented to control whether an address out-of-range error occurs for a received address which falls inside a multicast channel address range. A multicast channel enable bit can control whether an end point silently ignores the packet when an address is received which falls inside the channel address range. The enable bit allows software finer control over which end points for a particular multicast ID will actually process the multicast write without modifying switch settings in the fabric.

Blank page

Annex B Multicast Applications (Informative)

B.1 Introduction

In a multi-switch RapidIO fabric, each switch which supports multicast in the fabric will have it's own set of multicast masks. The particular multicast mask in each switch device associated with a multicast group is very likely to have a different egress port pattern enabled, depending upon where that switch is in the switch fabric topology.

As an example, refer to the following system, where data streams entering switch A1 need to be sent to a set of destinations. There are several possible approaches to implementing this system. The first example is based on the number of different multicast groups that must be supported. Destinations are linked to a destination ID which in turn is associated with static multicast mask values. In the second example, a specific multicast mask in each switch is associated with each possible destination. The destinations are linked statically to destination IDs.

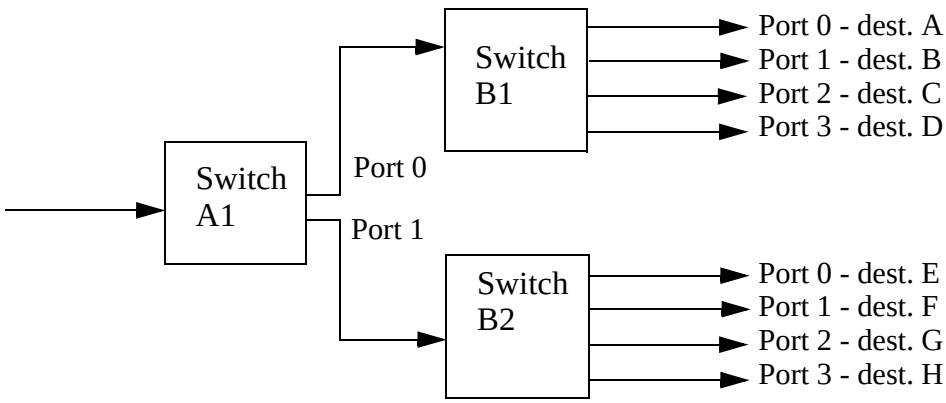

Figure 4-1. Example System using Multicast

B.2 Example 1 - Static Multicast Masks

If there are 256 combinations of destinations to receive a data stream, multicast requires 256 multicast groups, associated with 256 destination IDs. This means that an 8 bit destination ID could be used, but then there would be no destination IDs left over for control traffic in the system. As a result, this example assumes that the system needs to use 16 bit destination IDs in order to support multicast.

It is possible to use the least significant 4 bits of the 16 bit destination ID to identify which ports in Switch B1 need to be multicast to, and the next most significant 4 bits for the ports on B2. Arbitrarily selecting the value of 0x04 for the upper byte of the destination ID, then all multicast destination IDs have a format of 0x04XY, where X selects the ports in switch B2 and Y selects the ports in switch B1.

Switch A1 therefore needs two multicast masks as shown in Table 4-1.

Table 4-1. Multicast Masks for Switch A1

Multicast Mask Index	Egress Ports	Description
0	None	Associated with destination ID 0x0400, indicating that no destination is to receive this data stream. Packets multicast with destination ID of 0x0400 are dropped without notification.
1	Port 0 and port 1	Associated with destination IDs 0x04XY, where both X and Y is not 0. These represent all destination IDs which need only be multicast to both Switch B1 and switch B2.

Destination IDs of the form 0x040Y, where Y is non-zero, or 0x04X0, where X is non zero, do not have to be replicated. They can be routed directly to either port 0 (for 0x040Y) or port 1 (0x04X0) using the standard switch routing tables since there is only a single egress port.

Because Multicast Mask 1 must have 223 ((256 total) - (16 for X) - (16 for Y) - (1 for none)) destination IDs associated with it, the Switch Multicast Information CAR MaxDestIDAssociations field must contain a value of at least 222. In this particular case, the easiest internal implementation for the selection of packets to be multicast may be the use of a non-existent port in the routing table. For example, since Switch A1 has three ports, make use of a non-existent port value in the routing table to signify that the packet is subject to multicast.

Switches B1 and B2 must have 16 multicast masks, each associated with a particular combination of their egress ports 0 through 3. Each multicast mask may have 16 destination IDs associated with it, so the Switch Multicast Information CAR MaxDestIDAssociations field must contain a value of at least 15.

Table 4-2 describes which destination IDs must be associated with each multicast group for Switches B1. Note that for index 0, if the routing tables in Switch A1 are set up correctly, no packets with those multicast groups should reach switch B1.

Table 4-2. Multicast Masks for Switch B1

Multicast Mask Index	Egress Ports	Description
0	None	Associated with the following destination IDs: 0x0400 0x0410 0x0420 ... 0x04E0 0x04F0
1	Port 0	Associated with the following destination IDs: 0x0401 0x0411 0x0421 ... 0x04E1 0x04F1
2	Port 1	Associated with the following destination IDs: 0x0402 0x0412 0x0422 ... 0x04E2 0x04F2
3	Ports 1and 0	Associated with the following destination IDs: 0x0403 0x0413 ... 0x04E3 0x04F3
4	Port 2	Associated with the following destination IDs: 0x0404 0x0414 0x0424 ... 0x04E4 0x04F4
5	Ports2 and 0	Associated with the following destination IDs: 0x0405 0x0415 0x0425 ... 0x04E5 0x04F5
6	Ports2 and 1	Associated with the following destination IDs: 0x0406 0x0416 0x0426 ... 0x04E6 0x04F6

Table 4-2. Multicast Masks for Switch B1

Multicast Mask Index	Egress Ports	Description
7	Ports 2, 1 and 0	Associated with the following destination IDs: 0x0407 0x0417 0x0427 ... 0x04E7 0x04F7
8	Port 3	Associated with the following destination IDs: 0x0408 0x0418 0x0428 ... 0x04E8 0x04F8
9	Ports 3 and 0	Associated with the following destination IDs: 0x0409 0x0419 0x0429 ... 0x04E9 0x04F9
10	Ports 3 and 1	Associated with the following destination IDs: 0x040A 0x041A 0x042A ... 0x04EA 0x04FA
11	Ports 3, 1 and 0	Associated with the following destination IDs: 0x040B 0x041B 0x042B ... 0x04EB 0x04FB
12	Ports3 and2	Associated with the following destination IDs: 0x040C 0x041C 0x042C ... 0x04EC 0x04FC
13	Ports 3, 2 and 0	Associated with the following destination IDs: 0x040D 0x041D 0x042D ... 0x04ED 0x04FD

Table 4-2. Multicast Masks for Switch B1

Multicast Mask Index	Egress Ports	Description
14	Ports 3, 2 and 1	Associated with the following destination IDs: 0x040E 0x041E 0x042E ... 0x04EE 0x04FE
15	Ports 3, 2, 1 and 0	Associated with the following destination IDs: 0x040F 0x041F 0x042F ... 0x04EF 0x04FF

Table 4-2 describes which destination IDs must be associated with each multicast group for Switches B1. Note that for index 0, if the routing tables in Switch A1 are set up correctly, no packets with those multicast groups should reach switch B2.

Table 4-3. Multicast Masks for Switch B2

Multicast Mask Index	Egress Ports	Description
0	None	Associated with the following destination IDs: 0x0400 0x0401 0x0402 ... 0x040E 0x040F
1	Port 0	Associated with the following destination IDs: 0x0410 0x0411 0x0412 ... 0x041E 0x041F
2	Port 1	Associated with the following destination IDs: 0x0420 0x0421 0x0422 ... 0x042E 0x042F
3	Ports 1and 0	Associated with the following destination IDs: 0x0430 0x0431 0x0432 ... 0x043E 0x043F

Table 4-3. Multicast Masks for Switch B2

Multicast Mask Index	Egress Ports	Description
4	Port 2	Associated with the following destination IDs: 0x0440 0x0441 0x0442 ... 0x044E 0x044F
5	Ports2 and 0	Associated with the following destination IDs: 0x0450 0x0451 0x0452 ... 0x045E 0x045F
6	Ports2 and 1	Associated with the following destination IDs: 0x0460 0x0461 0x0462 ... 0x046E 0x046F
7	Ports 2, 1 and 0	Associated with the following destination IDs: 0x0470 0x0471 0x0472 ... 0x047E 0x047F
8	Port 3	Associated with the following destination IDs: 0x0480 0x0481 0x0482 ... 0x048E 0x048F
9	Ports 3 and 0	Associated with the following destination IDs: 0x0490 0x0491 0x0492 ... 0x049E 0x049F
10	Ports 3 and 1	Associated with the following destination IDs: 0x04A0 0x04A1 0x04A2 ... 0x04AE 0x04AF

Table 4-3. Multicast Masks for Switch B2

Multicast Mask Index	Egress Ports	Description
11	Ports 3, 1 and 0	Associated with the following destination IDs: 0x04B0 0x04B1 0x04B2 ... 0x04BE 0x04BF
12	Ports3 and2	Associated with the following destination IDs: 0x04C0 0x04C1 0x04C2 ... 0x04CE 0x04CF
13	Ports 3, 2 and 0	Associated with the following destination IDs: 0x04D0 0x04D1 0x04D2 ... 0x04DE 0x04DF
14	Ports 3, 2 and 1	Associated with the following destination IDs: 0x04E0 0x04E1 0x04E2 ... 0x04EE 0x04EF
15	Ports 3, 2, 1 and 0	Associated with the following destination IDs: 0x04F0 0x04F1 0x04F2 ... 0x04FE 0x04FF

It is up to the application whether either of the switch routing tables should be used for the destination IDs associated with multicast masks 1, 2, 4, and 8, as packets for these destination IDs do not have to be replicated.

Configuring each of the 16 multicast masks for switches B1 and B2 should require a maximum of 2 writes to the Multicast Mask Load CSR. Multicast masks with one or two ports require a number of register writes equal to the number of ports. Multicast masks with three egress ports to be selected should add all of the ports and then remove the port which doesn't belong in the multicast mask, thus requiring a maximum of two register writes. The multicast mask with all ports selected requires 1 register write. Thus, to configure all 16 of the multicast masks requires a maximum of (0 + (5*1) + (10*2))=25 register write operations.

For the destination ID to multicast mask association operations for Switch B1, it

would make sense to implement block association operations since this would greatly reduce the amount of effort required to associate destination IDs with multicast masks. This feature makes possible in this example to associate a sequential block of 16 destination IDs with the 16 multicast masks with only 32 register writes. Refer to Table 4.4.3, “Using Block Association,” on page 30 for details of the pair of writes required for each block of 16 destination IDs.

For the destination ID to multicast mask association operations for Switch B2 there is no pattern that leverages the programming model to speed the association of destination IDs to multicast masks. In Switch B2, it would make sense to use the regular switch routing tables rather than a multicast mask for the destination IDs associated with multicast masks 1, 2, 4 and 8 in order to minimize the number of writes required. The remaining 12 multicast groups each require 32 register write operations to complete their associations with the appropriate destination IDs, for a total of 384 writes. Designers who prefer speed of initialization over reliability may reduce this to 352 register writes by ignoring the destination IDs associated with multicast mask 0.

For switch B2, it may make sense in some systems to implement application specific configuration registers to reduce the number of operations required for configuration.

There can be significant limitations to using static multicast masks. Assume, of the 8 destinations, destinations A, B, C, D, and E are receiving one data stream using destination ID 0x041F, and destinations F, G, and H are receiving a second data stream using destination ID 0x0420.

If destination E switches wishes to change to the second data stream, two things must happen. The destination ID for the first data stream must change from 0x041F to 0x040F in order to have the proper multicast mask for switches B1 and B2, and the destination ID for the second data stream must correspondingly change from 0x0420 to 0x0430.

Because the destination IDs have changed, the switches are now allowed to reorder packets sent to destination IDs 0x041F, 0x040F, 0x0420 and 0x0430, which may change the behavior of the system in unexpected or undesirable ways.

Another issue with static multicast masks is that the latency difference for a data stream between different destinations depends upon whether the data stream is routed using the regular switch routing table or multicast through a particular switch. The different destinations will see different performance characteristics.

These characteristics could have undesirable side effects for latency and jitter sensitive applications like Voice over Internet Protocol (VoIP).

B.3 Example 2 - Linking Multicast Masks to Destination IDs

As an alternative implementation, again suppose that there are 256 possible destinations which need to be multicast, numbered 0 through 255. Each destination has a number of data streams it can receive, up to 256, which is always associated with a 16 bit destination ID of the form 0x04<destination stream>. This requires 256 multicast masks in switches A1, B1, and B2.

When a destination changes the data stream it wants to receive, the multicast masks for that data stream need to be changed. First, the multicast mask in each switch associated with the stream currently being received needs to be modified to stop multicasting to this destination. Next, the multicast mask for the new data stream needs to be modified in each switch to enable multicast to that destination.

Depending on system requirements, there are many ways to implement the multicast capabilities in this system. For example, switch A1 could always multicast all data streams to both switch B1 and switch B2. In this case, switch A1 would require 1 multicast mask that could have all 256 destination IDs associated with it. Switch B1 and B2 may receive a lot of undesired traffic in this case.

Initial programming of the multicast masks is not a requirement as with example 1. No ports should be selected in any mask after reset. Multicast masks will be modified during system operation as destinations request to receive a particular data stream. Removing the data streams from one multicast mask and adding a data stream to a multicast mask can be performed in two register writes for each switch.

The destination ID associations with multicast masks can be done far more effectively in this example if the switch devices support block associate operations. Refer to Section 3.4, “Switch Multicast Information CAR (Configuration Space Offset 0x38)”, and the programming examples in Section 4.4, “Configuring Associations” for more information and examples.

Blank page

Glossary of Terms and Abbreviations

The glossary contains an alphabetical list of terms, phrases, and abbreviations used in this book.

A

Associate, Association. A defined relationship between a destination ID and a group of end point devices, or, in a switch, a defined relationship between a destination ID and a multicast mask.

M

Multicast. The concept of sending a single message to multiple destinations in a system.

Multicast group. The group of end point devices in a system that is the target of a multicast operation.

Multicast mask. The group of egress ports in a switch that are the targets of a replicated multicast packet.

Blank page

Blank page

Blank page