

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

DEVICE INTEROPERABILITY LEVEL 2 TEST DESCRIPTION

This document and all of its contents are proprietary to RIOLAB. No part of its contents may be used, copied, disclosed or conveyed to any party in any manner whatsoever without prior written permission from RIOLAB. Information is subject to change without notice. All trademarks are the property of their respective owners.

Copyright 2007. All rights reserved.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

1 REVISION HISTORY

Release Status	Rev. #	Date	Author/ Reviser	Group	Description of Changes
Draft	0.80	28 May 2006	FET	Engineering	Initial Draft.
Draft	0.82	15 June 2006	FET	Engineering	Internal review of test description methodology
Draft	0.83	21 June 2006	FET	Engineering	Completed test descriptions
Draft	0.84	23 June 2006	FET	Engineering	Internal review feedback
First release	0.90	26 June 2006	FET	Engineering	First release for review
Updated	1.0	29 June 2006	FET	Engineering	Updated with comments
Updated	1.1	29 June 2006	FET	Engineering	Removed redundant tests
Updated	1.2	4 July 2006	FET	Engineering	Fixed typo
Updated	1.3	7 July 2006	FET	Engineering	Updated internal comments
Updated	1.4	9 July 2006	FET	Engineering	Updated typo
Updated	1.5	4 August 2006	FET	Engineering	Tundra feedback
Updated	1.6	28 Oct 2006	FET	Engineering	Test feedback
Update	1.7	27 Nov 2006	FET	Engineering	Cover 1.3 spec in TID7
Update	1.8	31 Jan 2007	FET	Engineering	Update
Update	1.9	22 Feb 2007	FET	Engineering	Update
Update	2.0	12 June 2009	FET	Engineering	T435 and 438 had the 257byte transactions removed.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

TABLE OF CONTENTS

1	REVISION HISTORY	2
2	INTRODUCTION.....	6
	2.1 Purpose.....	6
	2.2 Related Documents	6
	2.3 Terms and Definitions	6
3	TEST LEVEL DEFINITION.....	8
	3.1 Overview	8
	3.2 Test Set-up.....	8
	3.3 Other Processing Elements Tested Against.....	8
	3.4 Test Case Descriptions and Checklists.....	8
	3.5 User controlled inputs.....	12
4	TEST CASES	13
	4.1 RTA Checklist Testing.....	13
	4.1.1 Reads to reserved CAR bits (ID=7).....	13
	4.1.2 Port n Error and Status CSR (ID=144)	14
	4.1.3 Port n Error and Status CSR (ID=149)	14
	4.1.4 ackID Transmitted (ID=188)	14
	4.1.5 ackID Received (ID=216).....	15
	4.1.6 Acknowledge Control Symbol (ID=217)	15
	4.1.7 Acknowledge Control Symbol (ID=225)	16
	4.1.8 Response packets (ID=375).....	16
	4.1.9 MAINTENANCE requests (ID=377)	16
	4.1.10 CARs can be read (ID=391).....	17
	4.1.11 Device Identity CAR (ID=392)	17
	4.1.12 Device Information CAR (ID=393).....	17
	4.1.13 Assembly Identity CAR (ID=394).....	18
	4.1.14 Assembly Information CAR (ID=395).....	18
	4.1.15 Processing Element Features CAR (ID=396).....	18
	4.1.16 Switch Port Information CAR (ID=397).....	18
	4.1.17 Source Operations CAR (ID=398).....	19
	4.1.18 Destination Operations CAR (ID=399)	19
	4.1.19 All necessary CSRs exist and can be read (ID=400).....	19

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

4.1.20	Processing Element Logical Layer Control CSR (ID=401).....	19
4.1.21	Local Configurations Space Base Address 0 CSR (ID=402).....	20
4.1.22	Local Configurations Space Base Address 1 CSR (ID=403).....	20
4.1.23	Destination Operations CAR(ID=427)	20
4.1.24	NREAD operations (ID=428).....	21
4.1.25	NWRITE operations (ID=429).....	21
4.1.26	SWRITE operations (ID=430)	21
4.1.27	NWRITE_R operations (ID=431).....	22
4.1.28	NREAD Response (ID=434).....	22
4.1.29	Maximum NREAD CAR (ID=435).....	22
4.1.30	Maximum NWRITE (ID=438)	23
4.1.31	Maximum SWRITE (ID=443)	23
4.1.32	Maximum NWRITE_R (ID=448)	23
4.1.33	Source Operations CAR (ID=467).....	24
4.1.34	NREAD operations (ID=468).....	24
4.1.35	NWRITE operations (ID=469).....	24
4.1.36	NWRITE_R operations CAR (ID=470)	24
4.1.37	SWRITE operations (ID=471)	25
4.1.38	NREAD transaction (ID=472).....	25
4.1.39	NREAD packet (ID=473).....	25
4.1.40	NREAD packet size (ID=474).....	25
4.1.41	NWRITE transaction (ID=475).....	26
4.1.42	NWRITE packets – double words (ID=476)	26
4.1.43	NWRITE packet size (ID=477).....	26
4.1.44	NWRITE packet size (ID=478).....	27
4.1.45	NWRITE_R transaction (ID=479).....	27
4.1.46	NWRITE_R packet payload (ID=480)	27
4.1.47	NWRITE_R packet size (ID=481)	28
4.1.48	NWRITE_R packet size (ID=482)	28
4.1.49	SWRITE transaction (ID=483).....	29
4.1.50	SWRITE packet payloads (ID=484).....	29
4.1.51	All necessary CARs exist (ID=566).....	29

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

4.1.52 Device Identity CAR (ID=567)29

4.1.53 Assembly Identity CAR (ID=569).....30

4.1.54 MAINTENANCE read response (ID=580).....30

4.1.55 MAINTENANCE read response (ID=581).....31

4.1.56 MAINTENANCE write response (ID=585)31

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

2 INTRODUCTION

2.1 Purpose

This document describes Device Interoperability Level 2 (DIL2) tests. This is a living document that will be revised each time the test description is changed. The revision number of this document will form a basis of a formal interoperability report.

DIL2 tests represent the second level of interoperability. Once DIL1 has been confirmed, DIL2 Interoperability tests will be performed.

DIL2 testing verifies fundamental packet protocol and maintenance register behaviour. DIL2 testing utilizes both in-band and logic analyzer methods of verifying checklist tests.

As an Interoperability Test Lab, the repeatability of test set-ups, method and sequence is necessary. This document ensures that the test method and sequence is repeatable.

2.2 Related Documents

The following documents were used in the creation of, or are referenced in this document:

- RapidIO™ Interconnect Specification Rev. 1.2, 6/2002
- RapidIO™ Interconnect Specification Part VII: System and Device Interoperability Specification Rev. 1.2, 06/2002
- RapidIO™ Interconnect Specification Annex 1: Software/System Bring Up Specification Rev. 1.3, 02/2005
- RapidIO™ Interconnect Specification Device Inter-operability and Certification Checklists Rev. 1.0.a, 09/2004

Other related documents:

- DIL2_Test_Setup
- Lab_Hardware_Library
- Command_Interpreter_Description
- DIL2_Test_Scripts

2.3 Terms and Definitions

DUT	Device Under Test. The DUT may be the source of transactions, the target of transactions, and/or a conduit for transactions.
DIL1	Device Interoperability Level 1 testing where the DUT is tested against <u>all</u> other vendor devices in the Hardware Library for both request and response level testing for all combinations of A:B, B:A, A:C, C:A, A:B:C, C:B:A, etc. Focus is on the interaction between devices.
DIL2/DIL3	Device Interoperability Level 2 and 3 testing where the DUT is monitored for proper request and response protocol adherence with

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

a limited set of Qualified endpoints and/or switches. Level 3 quantitatively tests deeper into the protocol than that of 2. Focus is on the behaviour of the DUT.

Hardware Library	RIOLAB library of qualified devices
LA	Logic Analyzer
Qualified Device	A device that has successfully passed DIL1 where it has been directly tested against all other vendor devices in the hardware library.
Qualified Endpoint	A qualified device capable of running scripts or receiving transactions for a device running a script.
Scripts	RapidIO transactions defined using a specific syntax that identifies the transaction type, and the parameters required to fully define it.
Target	A destination for RapidIO transactions issued from the DUT, through the DUT or that represents the DUT.
User Input	Parameters required to be specified by the person conducting the test such as Destination ID, Hop Count, Offset address, number of bytes etc.
User Actions	Interaction required by the person conducting the test such as resetting hardware, Connecting LA probes, Analyzing packet content, etc

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

3 TEST LEVEL DEFINITION

3.1 Overview

Testing will fall into two basic categories:

- Initiate transactions:
 - For Endpoints this will entail basic I/O reads and writes as well as basic maintenance transactions to a target endpoint.
 - Successful transactions will be validated by either in-band read transactions or by using a logic analyzer.
 - For switches – n/a.
- Respond to transactions:
 - For Endpoints this will entail basic I/O read and write responses as applicable.
 - Read responses will be validated by comparing data read from a test target to the data written to the test target or by using a logic analyzer.
 - For switches, this will entail propagating packets to and from endpoints appropriately or responding to maintenance transactions.

Note: PE’s within the RIOLAB hardware library will be referred to as “qualified” devices.

3.2 Test Set-up

Refer to document: DIL2_Test_Setup.doc.

3.3 Other Processing Elements Tested Against

All of the tests run can be done between multiple devices as shown below in the Devices table.

Refer to document: Lab_Hardware_Library.doc.

3.4 Test Case Descriptions and Checklists

The following RTA Interoperability checklist items are covered within this test:

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009


Test ID	Description	Test Method
7.00	Reads to reserved CAR bits return logic 0s.	SCRIPT
144.00	In the Port n Error and Status CSR, the "Port OK" bit will be 0 and the "Port Un-initialized" bit will be 1 after device reset.	Script
149.00	After the device reset, a 1x / 4x device shall have the Port n Error and Status CSR "Port OK" bit is 0 and the "Port Uninitialized" bit is 1.	Script
188.00	Start with ackID=0 after reset	LA
216.00	Start with ackID=0 after reset.	LA
217.00	Every packet causes an acknowledge control symbol with the corresponding ackID field.	LA
225.00	Device must send acknowledge control symbols in same order as packets are received (ackIDs are issued sequentially).	LA
375.00	Response packets are sent with the target and destination fields reversed from the corresponding request packet.	LA
377.00	Endpoint free switch responds to maintenance requests with hop_count=0 when received.	LA
391.00	All necessary CARs exist and can be read.	Script
392.00	Device Identity CAR	Script
393.00	Device Information CAR	Script
394.00	Assembly Identity CAR	Script
395.00	Assembly Information CAR	Script
396.00	Processing Element Features CAR	Script
397.00	Switch Port Information CAR	Script
398.00	Source Operations CAR	Script
399.00	Destination Operations CAR	Script
400.00	All necessary CSRs exist and can be read.	Script
401.00	Processing Element Logical Layer Control CSR	Script
402.00	Local Configurations Space Base Address 0 CSR	Script
403.00	Local Configurations Space Base Address 1 CSR	Script
427.00	Destination Operations CAR must indicate target support for	Script
428.00	NREAD operations	Script
429.00	NWRITE operations	Script
430.00	SWRITE operations	Script
431.00	NWRITE_R operations	Script
434.00	NREAD response data payload is of the requested size and alignment.	Script
435.00	Maximum NREAD request can be 256 bytes.	LA
438.00	Maximum NWRITE data payload can be 256	LA

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

	bytes.	
443.00	Maximum SWRITE data payload can be 256 bytes.	LA
448.00	Maximum NWRITE_R data payload can be 256 bytes.	LA
467.00	Source Operations CAR must indicate source support for ...	Script
468.00	NREAD operations	Script
469.00	NWRITE operations	Script
470.00	NWRITE_R operations	Script
471.00	SWRITE operations	Script
472.00	NREAD transaction	LA
473.00	NREAD packet never has a data payload.	LA
474.00	NREAD packet must properly specify the requested size and alignment.	LA
475.00	NWRITE transaction	LA
476.00	NWRITE packets always contain data payloads of one or more double-words, although sub-double-word data may be specified.	
477.00	NWRITE packet must properly specify the data payload size and alignment.	
478.00	NWRITE packet actual data payload size may be less than the specified size (multiple double-word only).	
479.00	NWRITE_R transaction	LA
480.00	NWRITE_R packets always contain data payloads of one or more double-words, although sub-double-word data may be specified.	
481.00	NWRITE_R packet must properly specify the data payload size and alignment.	
482.00	NWRITE_R packet actual data payload size may be less than the specified size (multiple double-word only).	
483.00	SWRITE transaction	LA
484.00	SWRITE packets always contain data payloads of one or more double-words.	
566.00	All necessary CARs exist and can be read.	SCRIPT
567.00	Device Identity CAR; bits 0-15: value is implementation dependent, bits 16-31: value is vendor specific as assigned by the RapidIO Trade Association.	SCRIPT
569.00	Assembly Identity CAR; bits 0-15: value is implementation dependent, bits 16-31: value is vendor specific as assigned by the RapidIO Trade Association.	SCRIPT
580.00	MAINTENANCE read response payload size matches requested size.	SCRIPT
581.00	Response data for a MAINTENANCE packet consists of one or more double words, although sub-double-word data may have been requested.	SCRIPT

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

585.00

MAINTENANCE write response does not contain a data payload. LA 

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

3.5 User controlled inputs

User controlled inputs are required to control how RapidIO transactions are issued from a given source device to a target device where the device under test may be the source, the target, or a switch or bridge in the data path.

RIOLAB provides test scripts that define the detailed RapidIO transactions issued for any given test. User controlled inputs such as Destination ID, payload size, hop count, etc. are also defined for each of these scripts. The syntax for these scripts are defined in the document titled "DIL2 Test Scripts.doc". A simple Script Command Interpreter may be written in support of any processor endpoint to leverage the published scripts. Alternatively, an application called RapidFET™ from Fabric Embedded Tools Corporation may be used to perform various interoperability related tests before or after coming to RIOLAB.

User controlled inputs are defined for each checklist item in section 5.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

4 TEST CASES

There is one primary stage associated with DIL2 testing. It is based on the checklist defined in section 4.4.

Standard transaction checklist testing for DIL1 requires tests be issued from a number of qualified endpoints to or through the DUT and visa versa. DIL2 tests are all specific to DUT behaviour and therefore tests can be driven to and from only one other endpoint. Where RapidIO transactions are validated with a logic analyzer, only packets initiated by the DUT will be analyzed. Please refer to the Device Interoperability Level 2 Test Procedures document for details.

Note: At the DIL2 level, some of the tests require a logic analyzer to validate packet type, content and/or format. However, as the tests conducted are RapidIO interoperability tests and not device verification tests, RapidIO in-band verification will be utilized wherever possible.

4.1 RTA Checklist Testing

DIL2 testing for an Endpoint shall involve:

- **Phase A:** Test Scripts shall be run from a qualified endpoint, using the DUT as the target for transactions responses.
- **Phase B:** A different set of Test Scripts shall be run from the Device Under Test (DUT) to a qualified endpoint or switch.

DIL2 testing for a switch shall involve:

- **Phase A:** Test scripts shall be run from a qualified endpoint for maintenance transactions checklist items only, using the DUT as the target.
- **Phase B:** Other Test Scripts shall be run from one of the qualified endpoints to one of the other qualified endpoints through the DUT.

4.1.1 Reads to reserved CAR bits (ID=7)

Test ID 7: Reads to reserved CAR bits return logic 0s.

Using a qualified endpoint, issue maintenance read packets to the following CARs and confirm the bit fields specified return logic 0's.

CAR	Reserved bits	Comment
Processing Element Features	4:20, 22	
Switch Port Information	0:15	Switches only
Source Operations	10:12	
Destination Operations	10:12	

User input: Destination ID, Hop Count

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

User actions: none

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID7.txt.

4.1.2 Port n Error and Status CSR (ID=144)

Test ID 144: In the Port n Error and Status CSR, the "Port OK" bit will be 0 and the "Port Un-initialized" bit will be 1 after device reset.

Issue a maintenance read transaction from or through the DUT to the Port n Error and Status CSR of the DUT following a device reset. Verify that the "Port OK" bit is 0 and the "Port Un-initialized" bit is 1.

User input: Destination ID, Hop Count

User actions: Power up the DUT

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID144.txt.

4.1.3 Port n Error and Status CSR (ID=149)

Test ID 149: After the device reset, a 1x / 4x device shall have the Port n Error and Status CSR "Port OK" bit is 0 and the "Port Un-initialized" bit is 1.

Issue a maintenance read transaction from or through the DUT to the Port n Error and Status CSR of the DUT following a device reset. Verify that the "Port OK" bit is 0 and the "Port Un-initialized" bit is 1.

User input: Destination ID, Hop Count

User actions: Power up the DUT

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID149.txt.

4.1.4 ackID Transmitted (ID=188)

Test ID 188: Start with ackID=0 after reset (Transmission) - LA.

Issue a maintenance read transaction from or through the DUT of the Component Tag register of another endpoint as the first transaction following a device reset. Verify using a logic analyzer that the DUT transmitting the transaction uses an ackID=0 for the first transaction following the reset.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

User input: Destination ID, Hop Count.

User actions: Reset the system, Capture the maintenance read packet and interpret the ackID value and validate that the first transaction issue following a reset uses an ackID=0.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID188.txt.

4.1.5 ackID Received (ID=216)

Test ID 216: Start with ackID=0 after reset (Reception)- LA.

Using an endpoint, issue a maintenance read of the Component Tag register to the DUT as the first transaction following a device reset. Verify using a logic analyzer that the device receiving the transaction uses/expects an ackID=0 in the response packet.

User input: Destination ID, Hop Count

User actions: Reset the system, Capture the maintenance read response packet and interpret the ackID value and verify that the device receiving the transaction uses and ackID=0 in the response packet.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID216.txt.

4.1.6 Acknowledge Control Symbol (ID=217)

Test ID 217: Every packet causes an acknowledge control symbol with the corresponding ackID field - LA.

Using a qualified endpoint, issue 5 maintenance reads and 5 maintenance writes to the Component Tag Register in the DUT. Capture both the maintenance transactions and the acknowledge control symbols and verify that the acknowledge control symbols for each packet has a matching ackID to that of the maintenance transaction.

User input: Destination ID, Hop Count

User actions: Capture both the maintenance transactions and the acknowledge control symbols using a Logic Analyzer and verify that the acknowledge control symbols for each packet has a matching ackID to that of the maintenance transaction.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID217.txt.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

4.1.7 Acknowledge Control Symbol (ID=225)

Test ID 225: Device must send acknowledge control symbols in same order as packets are received (ackIDs are issued sequentially) - LA.

Using a qualified endpoint, issue 5 maintenance reads and 5 maintenance writes to the Component Tag Register in the DUT. Capture both the maintenance transactions and the acknowledge control symbols and verify that the acknowledge control symbols are issued in the same order as the corresponding packets are received with sequentially issued ackIDs.

User input: Destination ID, Hop Count

User actions: Capture both the maintenance transactions and the acknowledge control symbols using a Logic Analyzer and verify that the acknowledge control symbols are issued in the same order as the corresponding packets are received with sequentially issued ackIDs.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID225.txt.

4.1.8 Response packets (ID=375)

Test ID 375: Response packets are sent with the target and destination fields reversed from the corresponding request packet - LA.

Using a qualified endpoint, issue a maintenance write and a maintenance read to the Component TAG register of the DUT. Using a logic analyzer, ensure that the response packet target and destination fields are reversed from that of the maintenance write transaction.

User input: Destination ID, Hop Count

User actions: Using a logic analyzer, ensure that the response packet target and destination fields are reversed from that of the maintenance write transaction.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID375.txt.

4.1.9 MAINTENANCE requests (ID=377)

Test ID 377: Endpoint free switch responds to maintenance requests with hop_count=0 when received - LA.

Using a qualified endpoint, issue a maintenance write to the Component TAG register of the DUT. Verify using a maintenance read that the Component tag register was successfully written to and that the Device ID and Vendor ID registers match that of the

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

expected DUT. Using a logic analyzer, ensure that the maintenance write and read packets contains a hop_count= 0.

User input: Destination ID, Hop Count, DeviceID, VendorID

User actions: Using a logic analyzer, ensure that the maintenance packets contains a hop_count=0.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID377.txt.

4.1.10 CARs can be read (ID=391)

Test ID 391: All necessary CARs exist and can be read - Title.

Note: This is a checklist heading. It is included here for completeness and does not represent a checklist test in itself.

4.1.11 Device Identity CAR (ID=392)

Test ID 392: Device Identity CAR.

Using a qualified endpoint, issue a maintenance read to the Device Identity CAR. Verify that it successfully returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID392.txt.

4.1.12 Device Information CAR (ID=393)

Test ID 393: Device Information CAR

Using a qualified endpoint, issue a maintenance read to the Device Information CAR. Verify that it successfully returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID393.txt.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

4.1.13 Assembly Identity CAR (ID=394)

Test ID 394: Assembly Identity CAR.

Using a qualified endpoint, issue a maintenance read to the Assembly Identity CAR. Verify that it successfully returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID394.txt.

4.1.14 Assembly Information CAR (ID=395)

Test ID 395: Assembly Information CAR.

Using a qualified endpoint, issue a maintenance read to the Assembly Information CAR. Verify that it successfully returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID395.txt.

4.1.15 Processing Element Features CAR (ID=396)

Test ID 396: Processing Element Features CAR.

Using a qualified endpoint, issue a maintenance read to the Processing Element Features CAR. Verify that it successfully returns an expected value.

User input: Destination ID, Hop Count, and Processing Element Features CAR expected value

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID396.txt.

4.1.16 Switch Port Information CAR (ID=397)

Test ID 397: Switch Port Information CAR.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

Using a qualified endpoint, issue a maintenance read to the Switch Port Information CAR. Verify that it successfully returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID397.txt.

4.1.17 Source Operations CAR (ID=398)

Test ID 398: Source Operations CAR.

Using a qualified endpoint, issue a maintenance read to the Source Operations CAR. Verify that it successfully returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID398.txt.

4.1.18 Destination Operations CAR (ID=399)

Test ID 399: Destination Operations CAR.

Using a qualified endpoint, issue a maintenance read to the Destination Operations CAR. Verify that it successfully returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID399.txt.

4.1.19 All necessary CSRs exist and can be read (ID=400)

Test ID 400: All necessary CSRs exist and can be read - Title.

Note: This is a checklist heading. It is included here for completeness and does not represent a checklist test in itself.

4.1.20 Processing Element Logical Layer Control CSR (ID=401)

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

Test ID 401: Processing Element Logical Layer Control CSR.

Using a qualified endpoint, issue a maintenance read to the Processing Element Logical Layer Control CSR of an endpoint. Verify that it contains the value 0x00000001, or 0x00000002, or 0x00000004.

User input: Destination ID, Hop Count

User actions: none

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID401.txt.

4.1.21 Local Configurations Space Base Address 0 CSR (ID=402)

Test ID 402: Local Configurations Space Base Address 0 CSR.

Using a qualified endpoint, issue a maintenance read to the Local Configurations Space Base Address 0 CSR of an endpoint. Verify that it returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID402.txt.

4.1.22 Local Configurations Space Base Address 1 CSR (ID=403)

Test ID 403: Local Configurations Space Base Address 1 CSR.

Using a qualified endpoint, issue a maintenance read to the Local Configurations Space Base Address 1 CSR of an endpoint. Verify that it returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID403.txt.

4.1.23 Destination Operations CAR(ID=427)

Test ID 427: Destination Operations CAR must indicate target support for:

Note: This is a checklist heading. It is included here for completeness and does not represent a checklist test in itself.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

4.1.24 NREAD operations (ID=428)

Test ID 428: NREAD operations.

Using a qualified endpoint, issue a maintenance read to the Destination Operations CAR of an endpoint. Confirm that the READ field in the CAR returns the correct capability value successfully.

User input: Destination ID, Hop Count.

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID428.txt.

4.1.25 NWRITE operations (ID=429)

Test ID 429: NWRITE operations.

Using a qualified endpoint, issue a maintenance read to the Destination Operations CAR of an endpoint. Confirm that the nWRITE field in the CAR returns the correct capability value successfully.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID429.txt.

4.1.26 SWRITE operations (ID=430)

Test ID 430: SWRITE operations.

Using a qualified endpoint, issue a maintenance read to the Destination Operations CAR of an endpoint. Confirm that the sWRITE field in the CAR returns the correct capability value successfully.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID430.txt.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

4.1.27 NWRITE_R operations (ID=431)

Test ID 431: NWRITE_R operations.

Using a qualified endpoint, issue a maintenance read to the Destination Operations CAR of an endpoint. Confirm that the nWRITE_R field in the CAR returns the correct capability value successfully.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID431.txt.

4.1.28 NREAD Response (ID=434)

Test ID 434: NREAD response data payload is of the requested size and alignment.

Using the DUT, issue an nWrite transaction of 32 bytes to a memory address ending in 0x000 using data incrementing from 1 to 32 - decimal. Issue an nREAD transaction to an endpoint for a payload size of 21 bytes using a memory address ending with "0x002". Confirm that the nREAD transaction response returns sequential data starting at 3 and ending with 23 - decimal.

User input: Destination ID, memory location

User actions: none.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID434.txt.

4.1.29 Maximum NREAD CAR (ID=435)

Test ID 435: Maximum NREAD request can be 256 bytes.

Using the DUT, issue an nREAD transaction to an endpoint for a payload size of 256 bytes. Using a logic analyzer, confirm that the nRead request specifies 246 bytes and that the response is correct.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the nRead request specifies 246 bytes and that the response is correct.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID435.txt.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

4.1.30 Maximum NWRITE (ID=438)

Test ID 438: Maximum NWRITE data payload can be 256 bytes.

Using a qualified endpoint, issue an nWrite transaction to an endpoint for a payload size of 256 bytes. Using a logic analyzer, confirm that the data payload from the first write transaction is 256.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the data payload from the first write transaction is 256.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID438.txt.

4.1.31 Maximum SWRITE (ID=443)

Test ID 443: Maximum SWRITE data payload can be 256 bytes.

Using a qualified endpoint, issue an sWrite transaction to an endpoint for a payload size of 256 bytes. Using a logic analyzer, confirm that the data payload from the write transaction is 256 bytes.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the data payload from the write transaction is 256 bytes.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID443.txt.

4.1.32 Maximum NWRITE_R (ID=448)

Test ID 448: Maximum NWRITE_R data payload can be 256 bytes.

Using a qualified endpoint, issue an nWrite_R transaction to an endpoint for a payload size of 256 bytes. Using a logic analyzer, confirm that the data payload from the write transaction is 256 bytes.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the data payload from the write transaction is 256 bytes.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID448.txt.

4.1.33 Source Operations CAR (ID=467)

Test ID 467: Source Operations CAR must indicate source support for: - Title

Note: This is a checklist heading. It is included here for completeness and does not represent a checklist test in itself.

4.1.34 NREAD operations (ID=468)

Test ID 468: NREAD operations.

Using a qualified endpoint, issue a maintenance read to the Source Operations CAR. Verify that it successfully returns the correct value for the nRead bit field.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID468.txt.

4.1.35 NWRITE operations (ID=469)

Test ID 469: NWRITE operations.

Using a qualified endpoint, issue a maintenance read to the Source Operations CAR. Verify that it successfully returns the correct value for the nWrite bit field.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID469.txt.

4.1.36 NWRITE_R operations CAR (ID=470)

Test ID 470: NWRITE_R operations.

Using a qualified endpoint, issue a maintenance read to the Source Operations CAR. Verify that it successfully returns the correct value for the nWrite_R bit field.

User input: Destination ID, Hop Count

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID470.txt.

4.1.37 SWRITE operations (ID=471)

Test ID 471: SWRITE operations.

Using a qualified endpoint, issue a maintenance read to the Source Operations CAR. Verify that it successfully returns the correct value for the sWrite bit field.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID471.txt.

4.1.38 NREAD transaction (ID=472)

Test ID 472: NREAD transaction - Title.

Note: This is a checklist heading. It is included here for completeness and does not represent a checklist test in itself.

4.1.39 NREAD packet (ID=473)

Test ID 473: NREAD packet never has a data payload.

Using the DUT, issue an nREAD transaction to an endpoint for a payload size of 256 bytes. Using a logic analyzer, confirm that the nREAD transaction does not contain a data payload.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the nREAD transaction does not contain a data payload.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID473.txt.

4.1.40 NREAD packet size (ID=474)

Test ID 474: NREAD packet must properly specify the requested size and alignment.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

Using the DUT, issue an nREAD transaction to an endpoint for a payload size of 21 bytes using an memory address ending with “2”. Using a logic analyzer, confirm that the nREAD transaction correctly specifies the requested size and alignment.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the nREAD transaction correctly specifies the requested size and alignment.

- 3 packets Payload Content wdptr rdsiz
- 1st 8 bytes 6 bytes 0b1 0b1001
- 2nd 8 bytes 8 bytes 0b0 0b1011
- 3rd 8 bytes 7 bytes 0b0 0b1010

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID474.txt.

4.1.41 NWRITE transaction (ID=475)

Test ID 475: NWRITE transaction - Title.

Note: This is a checklist heading. It is included here for completeness and does not represent a checklist test in itself.

4.1.42 NWRITE packets – double words (ID=476)

Test ID 476: NWRITE packets always contain data payloads of one or more double-words, although sub-double-word data may be specified.

Using the DUT, issue an nWRITE transaction to an endpoint for a payload size of 21 bytes to an address ending with “0”. Using a logic analyzer, confirm that the two nWRITE transactions are issued containing the correct number of double word or sub-double word data.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the nWrite transactions payload contains:

- 2 packets Payload Content wdptr rdsiz
- 1st 16 bytes 16 bytes 0b1 0b1011
- 2nd 8 bytes 5 bytes 0b0 0b0111

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID476.txt.

4.1.43 NWRITE packet size (ID=477)

Test ID 477: NWRITE packet must properly specify the data payload size and alignment.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

Using the DUT, issue an nWRITE transaction to an endpoint for a payload size of 20 bytes using an memory address ending with “3”. Using a logic analyzer, confirm that the nWrite transaction correctly specifies the requested size and alignment.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the nWrite transactions payload size and alignment are:

- 3 packets

	Payload	Content	wdptr	rdsiz
○ 1 st	8 bytes	5 bytes	0b1	0b0111
○ 2 nd	8 bytes	8 bytes	0b0	0b1011
○ 3 rd	8 bytes	7 bytes	0b0	0b1010

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID477.txt.

4.1.44 NWRITE packet size (ID=478)

Test ID 478: NWRITE packet actual data payload size may be less than the specified size (multiple double-word only).

Using the DUT, issue an nWRITE transaction to an endpoint for a payload size of 5 bytes using an memory address ending with “3”. Using a logic analyzer, confirm that the nWrite transaction packet size is 8 bytes.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the nWrite transaction packet size is 8 bytes:

- 1 packet

○ 1 st	8 bytes	5 bytes	0b1	0b0111
-------------------	---------	---------	-----	--------

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID478.txt.

4.1.45 NWRITE_R transaction (ID=479)

Test ID 479: NWRITE_R transaction - Title.

Note: This is a checklist heading. It is included here for completeness and does not represent a checklist test in itself.

4.1.46 NWRITE_R packet payload (ID=480)

Test ID 480: NWRITE_R packets always contain data payloads of one or more double-words, although sub-double-word data may be specified.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

Using the DUT, issue an nWRITE_R transaction to an endpoint for a payload size of 21 bytes to an address ending with “0”. Using a logic analyzer, confirm that the two nWRITE transactions are issued containing the correct number of double word or sub-double word data.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the nWrite_R transactions payload contains:

- 2 packets

	Payload	Content	wdptr	rdsiz
○ 1 st	16 bytes	16 bytes	0b1	0b1011
○ 2 nd	8 bytes	5 bytes	0b0	0b0111

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID480.txt.

4.1.47 NWRITE_R packet size (ID=481)

Test ID 481: NWRITE_R packet must properly specify the data payload size and alignment.

Using the DUT, issue an nWRITE_R transaction to an endpoint for a payload size of 20 bytes using an memory address ending with “3”. Using a logic analyzer, confirm that the nWrite transaction correctly specifies the requested size and alignment.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the nWrite_R transactions payload size and alignment are:

- 3 packets

	Payload	Content	wdptr	rdsiz
○ 1 st	8 bytes	5 bytes	0b1	0b0111
○ 2 nd	8 bytes	8 bytes	0b0	0b1011
○ 3 rd	8 bytes	7 bytes	0b0	0b1010

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID481.txt.

4.1.48 NWRITE_R packet size (ID=482)

Test ID 482: NWRITE_R packet actual data payload size may be less than the specified size (multiple double-word only).

Using the DUT, issue an nWRITE_R transaction to an endpoint for a payload size of 5 bytes using an memory address ending with “3”. Using a logic analyzer, confirm that the nWrite transaction packet size is 8 bytes.

User input: Destination ID, memory location

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

User actions: Using a logic analyzer, confirm that the nWrite_R transaction packet size is 8 bytes:

- 1 packet Payload Content wdptr rdsiz
- 1st 8 bytes 5 bytes 0b1 0b0111

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID482.txt.

4.1.49 SWRITE transaction (ID=483)

Test ID 483: SWRITE transaction - Title.

Note: This is a checklist heading. It is included here for completeness and does not represent a checklist test in itself.

4.1.50 SWRITE packet payloads (ID=484)

Test ID 484: SWRITE packets always contain data payloads of one or more double-words..

Using the DUT, issue an sWRITE transaction to an endpoint for a payload size of 24 bytes to an address ending with “0”. Using a logic analyzer, confirm that the one sWRITE transaction is issued containing the correct number of double word data.

User input: Destination ID, memory location

User actions: Using a logic analyzer, confirm that the sWrite transactions payload contains:

- 1 packets Payload
- 1st 24 bytes

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID484.txt.

4.1.51 All necessary CARs exist (ID=566)

Test ID 566: All necessary CARs exist and can be read - Title.

Note: This is a checklist heading. It is included here for completeness and does not represent a checklist test in itself.

4.1.52 Device Identity CAR (ID=567)

Test ID 567: Device Identity CAR; bits 0-15: value is implementation dependent, bits 16-31: value is vendor specific as assigned by the RapidIO Trade Association.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

Using a qualified endpoint, issue a maintenance read to the Device Identity CAR. Verify that it successfully returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned in bit field 0-15 and 16-31.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID567.txt.

4.1.53 Assembly Identity CAR (ID=569)

Test ID 569: Assembly Identity CAR; bits 0-15: value is implementation dependent, bits 16-31: value is vendor specific as assigned by the RapidIO Trade Association.

Using a qualified endpoint, issue a maintenance read to the Assembly Identity CAR. Verify that it successfully returns an expected value.

User input: Destination ID, Hop Count

User actions: Confirm that the expected value was returned in bit field 0-15 and 16-31.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID569.txt.

4.1.54 MAINTENANCE read response (ID=580)

Test ID 580: MAINTENANCE read response payload size matches requested size.

Using a qualified endpoint, issue a maintenance read of:

- a) 64 bytes starting at the Assembly Identity CAR address - If the Source and DUT support greater than 4 byte maintenance transaction
- b) 4 bytes to the Assembly Identity CAR – if Source and/or DUT do not support greater than 4 byte maintenance transactions.

Using a logic analyzer, confirm that the maintenance read response packet payload size matches the requested size.

User input: Destination ID, Hop Count

User actions: When prompted, indicated if the Source and DUT support greater than 4 byte maintenance transactions. Using a logic analyzer, confirm that the maintenance read response packet payload size matches the requested size.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID580.txt.

Title:	Device Interoperability Level 2 Test Descriptions	File Number:	
Security Level:	Public	Release Status:	v 2.0
Owner Group:	RIOLAB	Revision Date:	12 June 2009

4.1.55 MAINTENANCE read response (ID=581)

Test ID 581: Response data for a MAINTENANCE packet consists of one or more double words, although sub-double-word data may have been requested.

If the Source and DUT support greater than 4 byte maintenance transactions: Using a qualified endpoint, issue a maintenance Read transaction to the Assembly Identity CAR in the DUT for a payload size of 21 bytes. Using a logic analyzer, confirm that the two maintenance read response packets are issued containing the correct number of double word or sub-double word data.

User input: Destination ID, Hop Count

User actions: **Prompt user if the Source and DUT support greater than 4 byte maintenance transactions. If “No” then skip the test. If “Yes” then:** Using a logic analyzer, confirm that the maintenance read response transactions payload’s contains:

- 2 packets

	Payload	Content	wdptr	rdsiz
○ 1 st	16 bytes	16 bytes	0b1	0b1011
○ 2 nd	8 bytes	5 bytes	0b0	0b0111

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID581.txt.

4.1.56 MAINTENANCE write response (ID=585)

Test ID 585: MAINTENANCE write response does not contain a data payload.

Using a qualified endpoint, issue a maintenance Write transaction to the Component Tag CSR in the DUT using a value of 0x12345678. Using a logic analyzer, confirm that the maintenance write response packet does not contain a data payload.

User input: Destination ID, Hop Count

User actions: Using a logic analyzer, confirm that the maintenance write response packet does not contain a data payload.

The scripts that are representative of the RapidIO transactions that reproduce this test can be found in document DIL2 Test Scripts ID585.txt.