

# **Interoperability Testing and the RapidIO® Standard: The Need for Repeatable and Quantitative Assessment**

*A White Paper by Jim Parisien, President*

*Fabric Embedded Tools Corporation &*

*Chair of the RapidIO® Trade Association Interoperability Working Group*

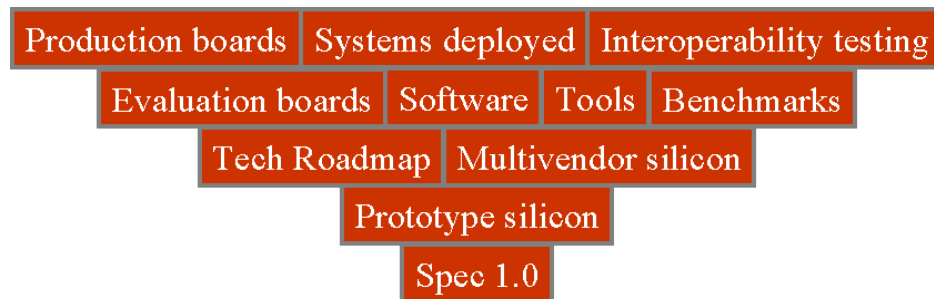
February, 2008

# Interoperability Testing and the RapidIO® Standard: The Need for Repeatable and Quantitative Assessment

By Jim Parisien, President, Fabric Embedded Tools Corporation

There are several key milestones a new technology goes through after it is first introduced. There is, of course, the initial specification and the race for first prototype silicon, followed by technology roadmaps and choices of multivendor silicon. Eventually, as a new technology matures, the focus begins to shift toward the health of the entire ecosystem.

The ecosystem encompasses everything a vendor needs to successfully design a product for a given technology. The ecosystem begins with a good selection of semiconductors but it doesn't end there. It includes development and prototype boards, systems, software and tools. One of the key ecosystem milestones in any technology is the appearance of a body that can provide formal unbiased interoperability testing.



That is not to say that interoperability testing doesn't begin soon after more than one semiconductor vendor has prototype silicon, but these results are typically neither unbiased nor formal. The kind of interoperability testing discussed in this white paper is completed by a third party independent of the semiconductor vendors. The results are produced in a formal report that can facilitate direct "apples to apples" comparison across different devices.

This milestone is an indication that there is both a sufficient "supply" of semiconductor devices in the ecosystem to choose from and that there is a sufficient "demand" for quantitative independent studies. There must also be sufficient software, tools and test criteria in place to create a repeatable and quantitative assessment of interoperability. In short, the existence of an independent interoperability facility is key evidence of a technology milestone as it transition from "new" status to "maturing" status.

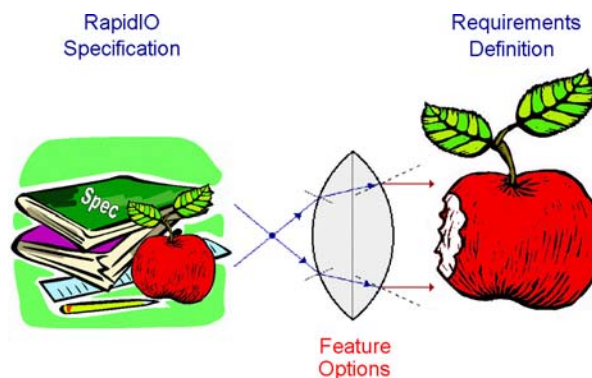
In 2007, the RapidIO® standard made this transition. RapidIO technology has, after all, been around since 2000, with the first parallel RapidIO devices debuting in 2004. While the physical interface changed in 2002 to include 10Gbps SERDES, the protocol remained largely unchanged and in 2005 the first serial RapidIO devices reached the market. Given this, it should be no surprise that RapidIO has achieved this maturity.

Not all interoperability testing is created equal, however. So let's take a closer look at Interoperability testing and the RapidIO standard.

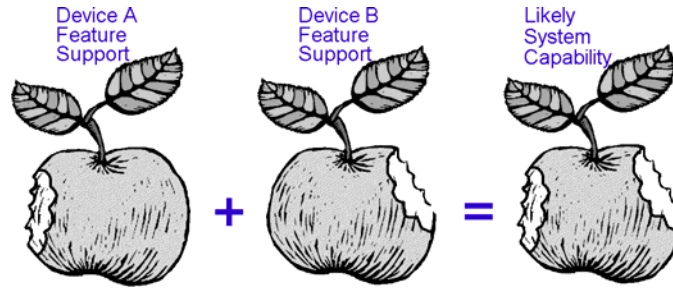
So what is interoperability? It comes in many flavours – device interoperability, board interoperability, software interoperability, and protocol interoperability. In truth, interoperability is a vast subject that covers the entire spectrum, but it always begins with device interoperability. Device interoperability is a prerequisite to all of the other types of interoperability. Without device interoperability, the rest is meaningless.

So why can't you just buy two devices from two different semiconductor manufacturers and have them work together? Isn't it enough if both devices are designed to the same specification revision? The simple answer is that while you can purchase two such RapidIO devices, they are not guaranteed to be interoperable even if first order testing shows they can pass traffic. The truth is that while the RapidIO specification is common starting point, there are three key potential implementation "distortions" that can affect device interoperability.

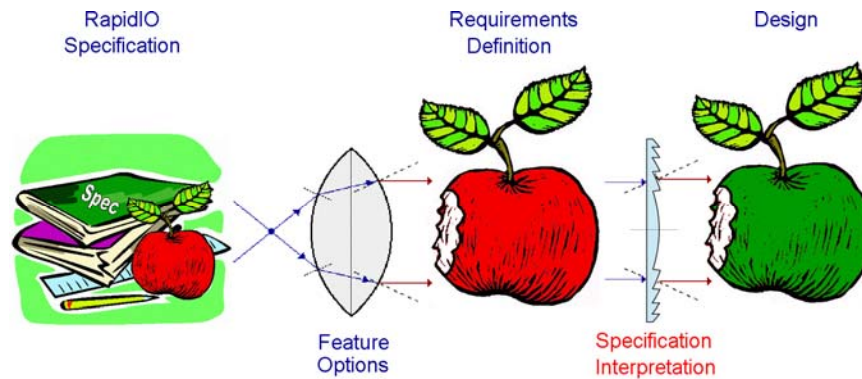
First, if a device is said to support version 1.3 of the RapidIO specification, it does not necessarily mean that *all* features defined in the 1.3 specifications have been implemented. It likely means that *some* 1.3 specification features have been implemented. This practice is not intended to be deceitful in any way; it is just that many features in the specification are optional. This is the first distortion that can affect interoperability.



It is imperative that designers understand which features are important because certain features may not work in a system unless they are supported by all, or most of, the devices within a network – such as multicast, large transport, messaging, doorbells, atomic operations, etc. The result of having a non-homogeneous system is that system capability often reduces to the lowest common denominator of features.

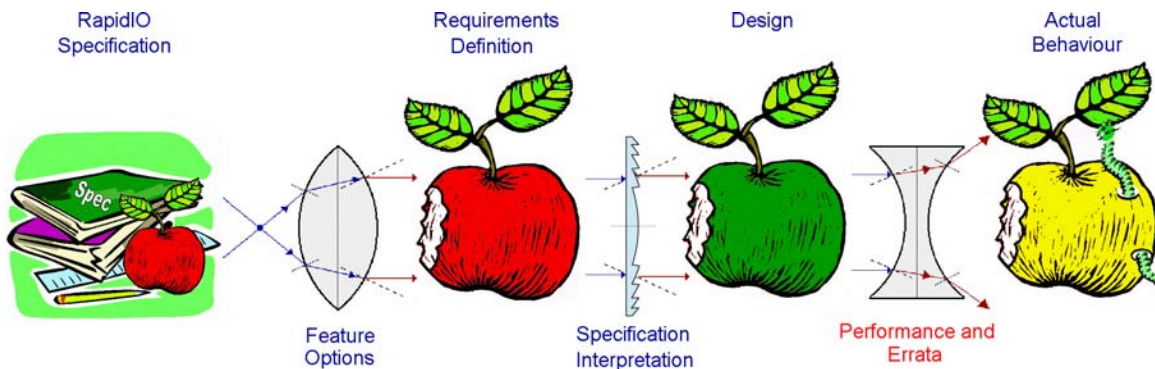


The second possible distortion that occurs is a function of “specmanship”. Regardless of how carefully a requirement is defined, the English language never seems precise enough to rule out multiple interpretations. The result is that designers can implement the same feature slightly differently. Often, this can affect interoperability. Even in cases where there are software “work-arounds” available, it is important to know about the issue before it can be addressed properly at the system level – not just for a specific device. So sometimes, the ideal intended design that is defined is not actually to spec.



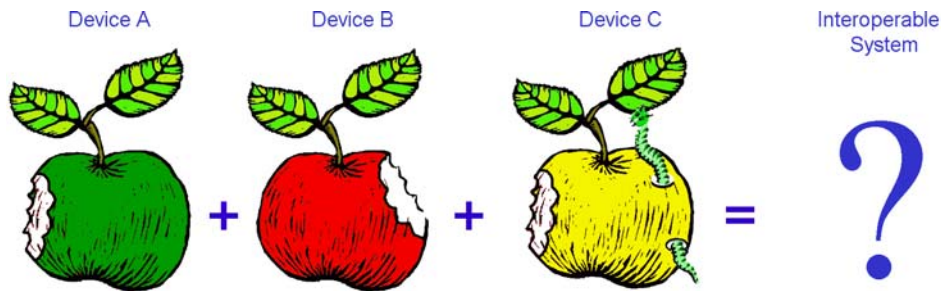
(Optional picture)

The third possible distortion that occurs is that the final implemented design does not always function as intended, especially in today’s highly complex semiconductor devices. Typically, this is due to architectural/performance limitations in the design itself, or implementation errors, i.e. bugs.



The result of all of this is that the actual behaviour of a device is not always likely to be fully representative of the intended specification. This is common for any device

that is designed to any specification and not unique to RapidIO devices. The only way to validate complete and true interoperability is to very carefully test each device against the RapidIO specification itself, and to test each device against every other device to ensure there are no surprises.



Certainly, this kind of testing begins early on by the semiconductor companies themselves. Typically two vendors who have synergistic markets and devices will do private testing to ensure their devices largely work together. This is certainly necessary, but not sufficient. The primary shortcoming here is that there is a different proprietary test suite used from vendor to vendor. Additionally, there isn't necessarily complete visibility of what parts of the specification have or have not been tested. Furthermore, systems today are not often built homogeneously from one semiconductor vendor, so it is not enough that only some vendors informally test together. In addition, in systems where blades are often used, it may not always be possible to control which semiconductor manufacturers are going to be plugged in over time. So not only should competitors test against each other, but ideally all devices should be tested against all other devices wherever possible.

What is required to do proper device interoperability testing?

First and foremost, a quantitative and repeatable test methodology is required. Then, to avoid conflicts of interest of vendors providing their own test results, testing through an independent third party must be completed.

Why is quantitative testing so important? Simply put, quantitative testing is the first step to gaining repeatability, and this is a critical step to allowing consistent "Apples to Apples" comparisons across complex devices, thereby letting designers make informed choices about the devices they select and those that are less desirable and should be left behind.



Qualitative testing in a generic software environment just doesn't cut it, as it is critical to know which processor or switch is the best choice for your application. Furthermore, it is not enough to have a specification checklist that is manually filled out by a semiconductor vendor to state intended compliance. A specification checklist is used as a starting point from which tests are derived to quantitatively confirm compliance. A clear set of procedures must also be defined so that tests can be repeated consistently. Finally, results need to be tabulated in a common format so that it is clear what tests have been performed, how they have been performed, and how well a device has performed against these tests.

While it is possible to cut corners and lean towards a more traditional qualitative assessment of interoperability, the costs of the problems that result from this are simply passed on to the end customer. Experience shows that the earlier on in the ecosystem chain that testing is completed, and the more thoroughly it's conducted, the less expensive it is in the long run. This has never been truer or more critical as interconnect technologies have become so much more complex and run at much higher speeds.

What is happening with the RapidIO standard and device interoperability?



At the very beginning, the RapidIO Trade Association recognized a need for both a common hardware interoperability platform (HIP) and a specification checklist from which to test from. It was clear that these elements were required to achieve "Repeatable and Quantitative" results.

The RapidIO Trade Association also encouraged the development of an independent third party RapidIO interoperability lab which received significant support from the RapidIO semiconductor community. In early 2006, the RapidIO Interoperability Lab (RIOLAB) formally opened and the existence of the HIP and specification checklist provided the foundation of what is today used in RIOLAB to perform interoperability testing. RIOLAB developed test scripts against the specification checklist and documented the procedures required to consistently run them. The Software infrastructure was developed such that the same test scripts can be run from any processor using any RTOS environment thereby ensuring consistency in testing all RapidIO elements (Processors, Bridges, Switches, and Memory). Testing is

completed for a given device against the checklist and across every other device in the qualified library.

Each device is tested against three device interoperability checklist levels (DIL1, DIL2, and DIL3) and the results are formally documented to facilitate comparison across different devices. A summary of these results are available at the beginning of each report allowing for a quick assess performance against other devices across all checklists. Also, detailed results of each phase of testing are recorded, including a checklist summary sheet, test script log, logic analyzer screen captures where appropriate, and register/routing table content at the time of testing. All of these data facilitate post failure analysis of device behaviour.

### Excerpt of the Table of Contents


			
		Page 2 of 39	
Title:	████████████████████	Revision:	1.2
Status:	Proprietary	Revision Date:	6 June 2007

**TABLE OF CONTENTS**

- 1 OVERVIEW.....4**
- 1.1 Terms and Definitions .....4
- 1.2 Vendor Information.....5
- 1.3 Test Lab Information .....5
- 1.4 Qualified Vendor Partner Program.....6
- 2 TEST SUMMARY INFORMATION .....6**
- 3 TEST RESULTS .....1**
- 3.1 Phase A Results .....1
  - 3.1.1 Part 1 – After Reset .....1
    - Test phase summary sheet .....2
    - Test checklist results sheet .....3
    - DIL1 Test Script Log.....3
    - Register Report .....3
    - Routing Table report .....3
    - Processing Element (PE) report.....3
    - Logic Analyzer Report.....4
  - 3.1.2 Part 2 .....4
    - Test phase summary sheet .....5
    - Test checklist results sheet .....6
    - DIL1 Test Script Log.....7
    - Register Report .....7
    - Routing Table report .....7
    - Processing Element (PE) report.....7
    - Logic Analyzer Report.....7
- 3.2 Phase B Results .....7
  - 3.2.1 Before Training .....7
    - Test phase summary sheet .....8
    - Test checklist results sheet .....9

## Example Checklist Summary Sheet



**SCORE: 17/17 100%**

**DIL3**

Bypass writing out LUT Script

DIL3 Checklist Test	
MCU Count	0
Destination ID	5
Node ID	2
Memory Address	

RapidIO Version 2.2.0 Build 0103

**Project Path:** Demonstration

**Test Date/Time:** 4/5/2007 11:09

Run Checklist Tests Now

Enable	ID	Section	Description	Max Score	Score	Result	Comments
	1	3.2.1	Device assigns reserved packet fields to logic 0s.	1			
	47	3.4.12.A.1	Port Maintenance Block Header 0 CSR has proper Extended Features Block ID in the EF_ID field.	3			
Yes	48	3.4.12.A.2	Port Maintenance Block Header 0 EF_PTR field reset value is implementation dependent.	1	1	Pass	Extended Features Pointer bit-field: (Bits[3-15] = 0x1000)
Yes	49	3.4.12.B	Port Link Time-out Control CSR, reset value of time-out value field 0xFFFFFFFF	1	1	Pass	
	50	3.4.12.B.1	Port Link Time-out Control CSR Bits 0-23 are writable, and time-out value varies with value written.	2			
	57	3.4.12.D.2	Verify Master Enable field accurately reflects the capabilities of the device.	1			
Yes	58	3.4.12.D.3	Verify Discovered field is set to 1 as soon as a maintenance write request is received to this register bit and if Discovered bit is 0 after reset for this device.	1	1	Pass	Port General Control CSR Discovered field (Bit[2]) = 0
	59	3.4.12.E	Port n Error and Status CSR, reset value is 0x00000001.	3	3	Pass	Ports tested: 4, 6, 12 Port 4: PASS Port 6: PASS Port 12: PASS
Yes	60	3.4.12.E.1	Verify that fields in the Port n Error and Status CSR are set on the appropriate conditions.	3			
	63	3.4.12.F	Port n Control CSR is compliant with the following.				
Yes	64	3.4.12.F.1	Reset value of the following fields is implementation dependent: Port_Width, Initialized_Port_Width, Output_Port_Enable, Input_Port_Enable, Multicast_Event_Participant.	2	2	Pass	Port n Control CSR Port 2_Address Oct19C Port_Width = 0x1 Initialized_Port_Width = 0x0 Output_Port_Enable = 0x1 Input_Port_Enable = 0x1 Multicast_Event_Participant = 0x0
	65	3.4.12.F.2	Reset value of the following fields is logic 0s: Port_Width_Override, Port_Disable, Error_Checkin_Disable.	2	2	Pass	Ports tested: 4, 6, 12 Port 4: PASS Port 6: PASS Port 12: PASS
Yes	66	3.4.12.F.3	Reset value of the Port_Type field is implementation defined.	1	1	Pass	Port n Control CSR Port 4_Address Oct10C Port_Type = 0x1
	67	3.4.12.F.4	Bits 13-33 are reserved (always logic 0s).	1			
	71	3.4.12.G.2	A read of the Port n Link Maintenance Request CSR returns the last command sent.	3			
	72	3.4.12.G.3	The reset value of the Port n Link Maintenance Request CSR is 0.	2	2	Pass	Ports tested: 4, 6, 12 Port 4: PASS Port 6: PASS Port 12: PASS
Yes	75	3.4.12.H.2	Subsequent reads of the Port n Link Maintenance Response CSR register have the response_valid field cleared.	3			
	76	3.4.12.H.3	The reset value of the Port n Link Maintenance Response CSR register is 0.	2	2	Pass	Ports tested: 4, 6, 12 Port 4: PASS Port 6: PASS Port 12: PASS
Yes	77	3.4.12.H.4	All bits in the Port n Link Maintenance Response CSR register are read only.	1			
	83	3.4.12.I.5	The reset value of the Port n Local ackID CSR register is 0.	2	2	Pass	Ports tested: 4, 6, 12 Port 4: PASS Port 6: PASS Port 12: PASS
Yes	84	3.4.12.I.6	The Inbound_ackID, Outstanding_ackID, and Outbound_ackID fields may be written to.	2			
	85	3.4.12.I.7	Bits 0-2, 0-16, and 24-26 are reserved (always logic 0s).	1			

The detailed results of each set of checklist tests are provided which include the detailed RapidIO transactions that took place along with Logic Analyzer packet analysis where appropriate. An overall percentage pass indication is given along with a score. Each checklist test is also weighted based on its level of importance – low 1, med 2, high 3 – so an overall score is reported. Furthermore, some tests are identified as being critical so that failing just one of these results in an overall failure.

In the end, devices that have passed are graded with a percentage and scored to facilitate quick comparison.

# Results Score

Example of part DIL1

ID	Section	Description	# Devices passed	8548	8455	6482	578	Score
253	3.9.15	A RapidIO device will accept packets of length up to 276 bytes. (Testing target device)	4/4	Pass	Pass	Pass	Pass	3
357	4.2.2.C	Component Tag CSR: reset value is 0x00000000.	4/4	Pass	Pass	Pass	Pass	1
360	4.2.3.A	Host Base Device ID CSR Host_base_deviceID field reset value is 0xFFFF.	4/4	Pass	Pass	Pass	Pass	2
361	4.2.3.B	When Host Base Device ID CSR Host_base_deviceID field value is 0xFFFF, the field value can be changed to any value.	4/4	Pass	Pass	Pass	Pass	2
362	4.2.3.C	When Host Base Device ID CSR Host_base_deviceID field value is not 0xFFFF, the field value will change to 0xFFFF when a value equal to the current field value is written.	4/4	Pass	Pass	Pass	Pass	2
363	4.2.3.D	When Host Base Device ID CSR Host_base_deviceID field value is not 0xFFFF, the field value does not change when a value not equal to the current field value is written.	4/4	Pass	Pass	Pass	Pass	2
364	4.2.3.E	When Host Base Device ID CSR Host_base_deviceID field value is 0xFFFF, and 0xFFFF is written to the field, subsequent writes of values not equal to 0xFFFF behave as per 4.2.2.B, 4.2.2.C, and 4.2.2.D.	4/4	Pass	Pass	Pass	Pass	2
365	4.2.3.F	The Component Tag CSR component_tag field can be written to any value.	4/4	Pass	Pass	Pass	Pass	2
412	5.3.1.B	MAINTENANCE read request generates a MAINTENANCE read response.	4/4	Pass	Pass	Pass	Pass	3
417	5.3.2.B	MAINTENANCE write request generates a MAINTENANCE write responses.	4/4	Pass	Pass	Pass	Pass	3
577	6.3.1	MAINTENANCE read transaction.	4/4	Pass	Pass	Pass	Pass	3
578	6.3.1.A	MAINTENANCE read request size of 4 bytes must be supported.	4/4	Pass	Pass	Pass	Pass	3
579	6.3.1.B	MAINTENANCE read request generates a MAINTENANCE read response.	4/4	Pass	Pass	Pass	Pass	3
582	6.3.2	MAINTENANCE write transaction.	4/4	Pass	Pass	Pass	Pass	3
583	6.3.2.A	MAINTENANCE write request may be for 4 bytes.	4/4	Pass	Pass	Pass	Pass	3
584	6.3.2.B	MAINTENANCE write request generates a MAINTENANCE write response.	4/4	Pass	Pass	Pass	Pass	3
	Data	Nwrite_R 8	4/4	Pass	Pass	Pass	Pass	3
	Data	Nwrite_R 32	4/4	Pass	Pass	Pass	Pass	3
	Data	Nwrite_R 256	4/4	Pass	Pass	Pass	Pass	3

• For each device

• Weighted tests

• Critical failures cause automatic failure of an interoperability level

The last key criteria required for good interoperability testing is unbiased and independent reporting. Today, Fabric Embedded Tools (FET) Corporation, a producer of RapidIO Network Management and Diagnostic tools, owns and operates RIOLAB. FET has been a member of the RapidIO Trade Association and has been selling products into the RapidIO ecosystem since 2004. FET's tools are designed to support all RapidIO semiconductor devices, and as such it is a non-biased and independent entity in which to operate an interoperability lab. Since acquiring RIOLAB in February 2007, FET has completed the test infrastructure and successfully produced test reports for several key semiconductor vendors in the ecosystem. While not all RapidIO semiconductor vendors have reports for all three levels yet, these early reports are evidence of an important start and commitment to providing the end user with a complete device interoperability picture offered by DIL1, 2 and 3. Furthermore, RIOLAB is continuing to work with the semiconductor vendors in the ecosystem so additional reports are expected in the coming quarters.

RIOLAB is focused on providing vendors with unbiased and cost-effective interoperability testing and reports for RapidIO Silicon and Boards. Semiconductor vendors, or any vendor for that matter, can commission RIOLAB to do interoperability testing on specific silicon or boards. Vendors can also purchase interoperability reports that have already been commissioned. These reports provide a quantitative view of how a device or board performs against a RapidIO specification checklist and other devices. RIOLAB reports are available today either through your preferred RapidIO semiconductor vendor or from RIOLAB.

There is never a substitute for purchasing evaluation boards and prototyping software, and confirming that the devices work. However, successfully sending large

volumes of packets across a system hardly constitutes a rigorous quantitative assessment of interoperability. Interconnect technologies today are pushing 10Gbps data rates and address a variety of requirements which span high availability to flow control. The depth and breadth of compliance testing required for comprehensive coverage is significant and potentially daunting. Designers don't have time to do in-depth interoperability testing autonomously, yet can't take the risk that components will simply "all function and perform the same way" because they have been designed to the same interconnect standard.

The choices are clear: either re-invent the wheel and reproduce the RIOLAB test suite or save time and money by accessing RIOLAB's testing services and/or existing interoperability test reports. Interoperability testing is the corner stone of the due diligence required to make an informed decision when choosing the best Processor, Bridge, Memory or Switch for your next RapidIO design. Semiconductor vendors understand this and are working with RIOLAB to address the need.

---

**Jim Parisien**  
President

Fabric Embedded Tools Corporation  
300 Earl Grey Drive, Suite 211  
Kanata, Ontario, Canada, K2T 1C1  
Phone (office): 1 (613) 271-9636  
Fax: 1 (613) 248-5089  
[www.fetcorp.com](http://www.fetcorp.com)

**About RIOLAB:**

RIOLAB is a state-of-the-art, independent testing facility that provides the RapidIO eco-system with device interoperability and specification compliance testing to meet the growing needs of silicon vendors and OEMs. Originally launched in February 2006, RIOLAB is a division of Fabric Embedded Tools Corporation, and continues to receive strong support from key RapidIO eco-system vendors including Altera, Ericsson, Freescale Semiconductor, Integrated Device Technology, Lucent Technologies, Mercury Computer Systems, Texas Instruments, Tundra Semiconductors, Xilinx, and other members of the RapidIO Trade Association Steering Committee, industry experts, and OEMs.

