RapidIO™ Interconnect Specification Part 8: Error Management/Hot Swap Extensions Specification

4.1, 6/2017

RapidIO.org

Revision History

Revision	Description	Date
1.2	First public release	09/13/2002
1.3	Technical changes: the following errata showings: 04-02-00002.001 and the following new features showings: 04-09-00022.002 Converted to ISO-friendly templates	02/23/2005
2.0	Technical changes: errata showing 06-02-00001.005	06/14/2007
2.1	Technical changes: errata showing 07-07-00000.004	07/09/2009
2.2	Technical changes: errata showing 10-08-00000.003	05/05/2011
3.0	Changed RTA contact information. Technical Changes: Standardized support for hot extraction and hot insertion, including register refinement/ additions. Register additions/changes to support Dev32. Register additions/changes to allow reuse of registers to latch information for packet types previously unsupported by this part of the RapidIO standard, and to enable latching control symbol information encoded using 64b/67b.	10/11/2013
3.1	Minor editorial changes. Technical changes: Added support for multiple error capture FIFO for physical layer and logical/transport layer errors. Register additions/changes for multiple error capture FIFO support. Register field additions for MECS and SMECS time synchronization error reporting.	9/18/2014
3.2	Inclusion of errata 9: Port n Link Uninit Discard Timer CSR Change	01/28/2016
4.0	No technical changes.	06/15/2016
4.1	No technical changes. Editorial change to Appendix A “Hot-insertion/extraction Discussion” on page 50, clarifying that hot swap/hot plug control mechanism is system specific.	06/30/2017

Questions regarding RapidIO.org, specifications, or membership should be forwarded to:

RapidIO.org
8650 Spicewood Springs #145-515
Austin, TX 78759
512-827-7680 Tel.

Table of Contents

Chapter 1 Error Management Extensions

Chapter 2 Error Management Registers

Table of Contents

List of Tables

List of Tables

Blank page

Chapter 1 Error Management Extensions

1.1 Introduction

The error management extensions describe added requirements in all physical and logical layers. These extensions add definitions to bits that were previously reserved in the Port *n* Control CSRs and add new registers that are contained within the Error Management Extended Features Block. This chapter describes the behavior of a device when an error is detected and how the new registers and bits are managed by software and hardware. Implementation of this specification is optional.

1.2 Physical Layer Extensions

The following registers and register bit extensions allow software to monitor and control the reporting of physical layer errors:

- (Extensions to the) Port n Control CSRs defined in Section 2.2
- (Extensions to the) Port n Error and Status CSRs defined in Section 2.2
- Port-Write Target deviceID CSR defined in Section 2.5.11
- Port n Error Detect CSR defined in Section 2.5.15
- Port n Error Rate Enable CSR defined in Section 2.5.16
- Port n Attributes Capture CSR defined in Section 2.5.17
- Port n Capture 0 CSR through Port n Capture 4 CSR defined in Section 2.5.18 through Section 2.5.22
- Port n Error Rate CSR defined in Section 2.5.23
- Port n Error Rate Threshold CSR defined in Section 2.5.24

The Hot Swap Extensions consist of the following registers and register bit extensions, which allow software to be notified of the addition and removal of processing elements:

- Port-Write Target deviceID CSR defined in Section 2.5.11
- (Extensions to the) Port n Error Detect CSR defined in Section 2.5.15
- (Extensions to the) Port n Error Rate Enable CSR defined in Section 2.5.16
- Port n Link Uninit Discard Timer CSR defined in Section 2.5.25

1.2.1 Port *n* Error Detect, Enable, and Capture CSRs

Each detected occurrence of a physical layer error shall be logged by hardware in the Port *n* Error Detect CSRs by setting the appropriate error indication bit. Each detected error occurrence should set no more than one error indication bit, the bit that most specifically identifies the detected error. The Port *n* Error Detect CSRs does not lock when a detected error bit is set allowing each subsequent detected error to also be logged in the register. By reading the register, software may see the types of physical layer errors that have occurred since the register was last cleared.

Physical layer errors are enabled for error capture and error counting when the bit corresponding to the error has been set in the Port *n* Error Rate Enable CSRs by software. Error information is captured in the Port *n* Attributes Capture CSRs and the Port *n* Capture 0-4 CSRs. The Capture Valid Info bit in the Port *n* Attributes Capture CSRs indicates whether the error information in the capture CSRs is valid.

When the Capture Valid Info status bit is not set in the Port *n* Attributes Capture CSRs, information about the next enabled physical layer error shall be saved to the Port *n* Capture 0-4 CSRs. The Info Type and Error Type fields of the Port *n* Attributes Capture CSRs shall be updated and the register's Capture Valid Info status bit shall be set by hardware to lock the error capture registers. Typically, the first 16 or 20 bytes of a packet, the 4 bytes of a delimited Control Symbol 24, or the 8 bytes of a delimited Control Symbol 48 or Control Symbol 64 that have a detected error are saved in the Port *n* Capture 0-4 CSRs. Packets smaller than 16 bytes are captured in their entirety. The Port *n* Capture 0-4 CSRs and the Port *n* Attributes Capture CSRs are not overwritten by hardware with error capture information for subsequent errors until software writes a zero to the Capture Valid Info bit.

The characters used for data transfer by the 8b/10b encoded LP-Serial physical layer protocol are 9 bit entities, but the specified formats for the Port *n* Capture 0-4 CSRs allocate only 8 bits per character (4 characters per 32-bit CSR) for recording the characters of a corrupted control symbol or the first 16 characters of a corrupted packet. Therefore it is not possible to unambiguously capture all possible LP-Serial control symbol and packet corruptions using the specified Port *n* Capture 0-4 CSRs format. Examples of this are when a code-group encoding a data character is changed by a transmission error into a code-group encoding a special character or a code-group with no 8b/10b decoding (an invalid character).

1.2.2 Error Reporting Thresholds

Physical layer errors are normally hidden from system software since they may be recovered with no loss of data and without software intervention. Two thresholds are defined in the Port *n* Error Rate Threshold CSRs which can be set to force a report to system software when the physical layer error rate reaches a level that is deemed by the system to be either degraded or unacceptable. The two thresholds are respectively the Degraded Threshold and the Failed Threshold. These thresholds are

used as follows.

When the error rate counter is incremented, the Error Rate Degraded Threshold Trigger provides a threshold value that, when equal to or exceeded by the value in the Error Rate Counter in the Port *n* Error Rate register, shall cause the error reporting logic to set the Output Degraded-encountered bit in the Port *n* Error and Status CSRs, and notify the system software as described in Section 1.4.

The Error Rate Failed Threshold Trigger, if enabled, shall be larger than the degraded threshold trigger. It provides a threshold value that, when equal to or exceeded by the value in the Error Rate Counter, shall trigger the error reporting logic to set the Output Failed-encountered bit in the Port *n* Error and Status CSRs, and notify system software as described in Section 1.4.

No action shall be taken if the Error Rate Counter continues to exceed either threshold value after initial notification when additional errors are detected. No action shall be taken when the Error Rate Counter drops below either threshold.

1.2.3 Error Rate Control and Status

The fields in the Port *n* Error Rate CSRs are used to monitor the error rate of the port *n* physical layer.

The Error Rate Counter field contains the 8-bit Error Rate Counter. The Error Rate Counter shall increment when a physical layer error is detected whose associated enable bit is set in the Port *n* Error Rate Enable CSRs. The Error Rate Counter shall decrement at the rate specified by the Error Rate Bias field of the Port *n* Error Rate CSRs. The Error Rate Counter shall not underflow (shall not decrement when equal to 0x00) and shall not overflow (shall not increment when equal to 0xFF). The incrementing and decrementing of the Error Rate Counter are in no way affected by the values in the Degraded and Failed threshold fields. Software may reset the Error Rate Counter at any time.

The rate at which events are counted by the Error Rate Counter depends on the error rates and the bits set in the Port *n* Error Rate Enable CSRs. If bit 11 “Received packet-not-accepted control symbol enable” of the Port *n* Error Rate Enable CSRs is not set, only errors detected by Port *n* and whose counting is enabled are counted. If bit 11 is set, then errors detected by the connected port are also counted as the reception of a packet-not-accepted control symbol, while not an error in itself, is an indication that the connected port has detected a physical layer error. If in addition to bit 11 being set, one or more of virtual channels 1-8 are enabled and are operating in reliable transmission (RT) mode, packet retries requested by the connected port will also be counted because packet-not-accepted control symbols are used in this case to signal the rejection of an RT packet by the connected port due to a lack of buffer space.

The Error Rate Bias field determines the rate at which the Error Rate Counter is decremented and defines the acceptable error rate of the physical layer for error

reporting purposes. In the absence of additional counted physical layer errors, this mechanism allows the system to recover from both Failed and Degraded levels of operation without a software reset of the Error Rate Counter. If the error rate of the physical layer errors being counted is less than the decrement rate specified in the Error Rate Bias field, the value of the Error Rate counter will rarely be greater than 0x01 or 0x02.

The Error Rate Recovery field defines how far above the Error Rate Failed Threshold Trigger in the Port *n* Error Rate Threshold Register the Error Rate Counter is allowed to count. In the absence of additional counted errors, this allows software to control the length of time required for the value of the Error Rate Counter to drop below both the Failed and Degraded Thresholds.

The Peak Error Rate field shall contain the largest value encountered by the Error Rate Counter since the field was last reset. This field is loaded whenever the current value of the Peak Error Rate field is exceeded by the value of the Error Rate Counter.

1.2.4 Port Behavior When Error Rate Failed Threshold is Reached

The behavior of a port when the Error Rate Counter in the Port *n* Error Rate CSRs reaches the Error Rate Failed Threshold and the threshold is enabled depends upon the values of the Stop on Port Failed-encountered Enable and the Drop Packet Enable bits in the Port *n* Control CSRs. The Table 1-1 below defines the required behavior.

Table 1-1. Port Behavior when Error Rate Failed Threshold has been hit

Stop on Port Failed Encountered Enable	Drop Packet Enable	Port Behavior	Comments
0	0	The port shall continue to attempt to transmit packets to the connected device if the Output Failed-encountered bit is set and/or if the Error Rate Failed threshold has been met or exceeded.	All devices
0	1	The port shall discard packets that receive a Packet-not-accepted control symbol when the Error Rate Failed Threshold has been met or exceeded. Upon discarding a packet, the port shall set the Output Packet-dropped bit in the Port *n* Error and Status CSRs. If the output port “heals”, the Error Rate Counter falls below the Error Rate Failed Threshold, the output port shall continue to attempt to forward all packets.	Switch Device Only

Table 1-1. Port Behavior when Error Rate Failed Threshold has been hit

Stop on Port Failed Encountered Enable	Drop Packet Enable	Port Behavior	Comments
1	0	The port shall stop attempting to send packets to the connected device when the Output Failed-encountered bit is set. The output port will congest.	All devices.
1	1	The port shall discard all output packets without attempting to send when the port's Output Failed-encountered bit is set. Upon discarding a packet, the port shall set Output Packet-dropped bit in the Port *n* Error and Status CSRs.	All devices.

1.2.5 Packet Timeout Mechanism in a Switch Device

In some systems, it is either desirable or necessary to bound the length of time a packet can remain in a switch. To enable this functionality, a switch shall monitor the length of time each packet accepted by one of its ports has been in the switch. The acceptance of a packet by a port is signaled by the port issuing a packet-accepted control symbol for the packet. The timing begins when the port accepts the packet.

If a packet remains in a switch longer than the Time-to-Live time specified by the Time-to-Live field of the Packet Time-to-live CSR as defined in Section 2.5.12, the packet shall be discarded rather than forwarded, the Output Packet-Dropped bit shall be set in the Port *n* Error and Status CSRs, and the system shall be notified as described in Section 1.4.

1.2.6 Hot Swap Extensions

When a Field Replaceable Unit (FRU) is inserted into a running system, it may be necessary to immediately inform system software. Similarly, when an FRU is removed from a running system, it may be necessary to immediately inform system software. The Link Uninit to OK Transition event can be used to inform system software of the insertion or removal of an FRU.

In the event that an FRU is removed from a system unexpectedly, the number of physical layer errors detected is uncertain. It is not possible to set the Physical Layer Error Management extensions thresholds, as described in sections 1.2.2 and 1.2.3, to differentiate between an expected bit error rate and FRU removal. The Hot Swap Extensions uses a timeout period for link reinitialization, the Port *n* Link Uninit Discard Timer CSRs, to detect when a link has been unavailable for a period of time deemed excessive by the system. When the Port *n* Link Uninit Discard Timer period expires, packets are discarded to avoid system congestion. Depending on the system design, the congestion could prevent system software from handling the unexpected FRU removal, which could lead to system failure.

A port-write may be sent to inform system software of a Hot Swap Extensions event. The Hot Swap Extensions events are incorporated into the Port *n* Error Detect CSRs, as the contents of this CSR are sent in a port-write. The Hot Swap Extensions events are also included in the Port *n* Error Rate Enable CSRs, as this is the standard register that controls notification and information capture for physical layer events.

However, unlike the Error Management Extensions physical layer events, the removal or insertion of an FRU is not a correctable error. For this reason, unlike Error Management Extensions events, Hot Swap Extension events shall not contribute to the error reporting thresholds described in section 1.2.2/1.2.3, shall not cause any error information to be latched, and shall not cause the Port *n* Capture 0-4 CSRs to lock.

1.2.7 Physical Layer Multiple Event Capture

Some fault tolerant applications require capture of multiple events to understand the sequence of events that lead to a failure. It is possible to implement the capture of multiple errors as a First-In-First-Out (FIFO) queue of events, where each entry in the FIFO represents the six registers starting with the Port n Attributes Capture CSR at the time of the entry. The Port n FIFO Error Detect CSR, also part of each FIFO entry, captures information similar to the Port n Error Detect CSR. The Port n Error Detect CSR behavior is constant, whether or not a FIFO is implemented.

The FIFO shall provide a queue of at least two entries, with the oldest entry at the front of the queue and new entries added at the end of the queue.

A FIFO entry becomes occupied when that entry is added to the end of the FIFO queue. The six registers starting with the Port n Attributes Capture CSR, and the Port n FIFO Error Detect CSR, shall occupy the oldest unoccupied entry in the FIFO.

The oldest occupied FIFO entry shall become unoccupied when software writes a zero to the Capture Valid Info bit. The FIFO shall be considered full if all entries are occupied. The FIFO shall be considered empty if all entries are unoccupied.

The FIFO error capture function supplies the value that occupies a FIFO entry. The FIFO error capture function operates on enabled and disabled events, which are defined in 1.2.1, “Port n Error Detect, Enable, and Capture CSRs”. The FIFO error capture function shall operate as follows:

- The FIFO error capture function value for the Port n FIFO Error Detect CSR shall consist of all events detected since the last time the FIFO error capture function value occupied an entry in the FIFO.
- The FIFO error capture function value for the Port n Attributes Capture CSR shall be updated for every detected enabled event, and shall consist of attributes information for the detected enabled event.
- The FIFO error capture function value for the five registers starting with the Port n Capture 0 CSR shall be updated for every detected enabled event.

If the FIFO is not full and an enabled event has been detected, the current FIFO error capture function value shall occupy the next entry in the FIFO. If multiple enabled events occur simultaneously, at least one event shall occupy an entry in the FIFO.

The Capture Valid Info bit in the Port n Attributes Capture CSR shall be 1 when at least one FIFO entry is occupied. The Capture Valid Info bit in the Port n Attributes Capture CSR shall be 0 when all FIFO entries are unoccupied. The value of other fields in the Port n Attributes Capture CSR, and the value of the Port n FIFO Error Detect CSR and the five registers starting with the Port n Capture 0 CSR, is undefined when all FIFO entries are unoccupied.

1.3 Logical and Transport Layer Extensions

While the RapidIO physical layer may be working properly, an end point processing element may encounter logical or transport layer errors, or other errors unrelated to its RapidIO ports, while trying to complete a transaction. The “ERROR” status response transaction is the mechanism for the target device to indicate to the source that there is a problem completing the request. Experiencing a timeout waiting for a response is also a symptom of an end point or switch fabric with a problem. These types of errors are logged and reporting enabled with a set of registers that are separate from those used for the Physical Layer errors:

- Logical/Transport Layer Error Detect CSR defined in Section 2.5.3
- Logical/Transport Layer Error Enable CSR defined in Section 2.5.4
- Logical/Transport Layer Capture CSRs defined in Section 2.5.5 to Section 2.5.8

1.3.1 Logical/Transport Error Detect, Enable, and Capture CSRs

When a logical or transport layer error is detected, the appropriate error bit shall be set by the hardware in the Logical/Transport Layer Error Detect CSR. If the corresponding bit is also set in the Logical/Transport Layer Error Enable CSR, the detect register shall lock, the appropriate information is saved in the Logical/Transport Layer Capture registers, all resources held by the transaction are freed, and system software is notified of the error as described in Section 1.4. If multiple enabled errors occur during the same clock cycle, multiple bits will be set in the detect register and the contents of the Logical/Transport Layer Capture registers are implementation dependent. Once locked, subsequent errors will not set another error detect bit. The contents of the Logical/Transport Capture CSRs are valid if the bitwise AND of the Logical/Transport Layer Error Detect CSR and the Logical/Transport Layer Error Detect Enable CSR is not equal to zero (0x00000000).

Software shall write the Logical/Transport Detect register with all logic 0s to clear the error detect bits or a corresponding enable bit to unlock the register. Any other recovery actions associated with these types of errors are system dependent and outside the scope of this specification.

1.3.2 Message Passing Error Detection

Message passing is a special case of logical layer error recovery requiring error detection at both the source and destination ends of the message. The source of the message has the request-to-response timeout (defined in the Port Response Timeout Control CSR in the RapidIO Physical Layer specifications) to detect lost request or response packets in the switch fabric. However, in order to not hang the recipient mailbox in the case of a lost request packet for a multiple packet message, the recipient mailbox shall have an analogous response-to-request timeout. This timeout is for sending a response packet to receiving the next request packet of a given message operation, and has the same value as the request-to-response timeout that is already specified. The Logical/Transport Layer Control Capture CSR contains the ‘msg info’ field to capture the critical information of the last received (or sent) message segment before timeout.

1.3.3 Other Logical Layer Errors

The RapidIO specification contains many logical layer packet types beyond the Logical I/O and Messaging types, which can be used in systems with greatly varying complexity. The capabilities necessary to find the root cause of defects for these packet types and systems also vary with customer requirements and the implementation technology. While the need to find defects in these systems is constant, it is not necessary to define standard methods of debugging them, as this

does not affect interoperability.

RapidIO devices should consider their users needs for defect determination, and capture information appropriate to the scale and complexity of the system. Debug needs should be considered both as the source of a request and the target. Debug capabilities for customer field use may also be needed.

The basis of defect determination is information about detected errors. The Logical/Transport layer capture registers may be used to capture such information.

1.3.4 Logical/Transport Layer Multiple Event Capture

Some fault tolerant applications require capture of multiple events to understand the sequence of events that lead to a failure. It is possible to implement the capture of multiple errors as a First-In-First-Out (FIFO) queue of events, where each entry in the FIFO consists of the Logical/Transport Layer Error Detect CSR, and all Logical/Transport Layer Capture CSRs, at the time of the entry.

The FIFO shall provide a queue of at least two entries, with the oldest entry at the front of the queue and new entries added at the end of the queue.

A FIFO entry becomes occupied when that entry is added to the end of the FIFO queue. The Logical/Transport Layer Error Detect CSR, and all Logical/Transport Layer Capture CSRs shall access the oldest occupied entry in the FIFO.

The oldest occupied FIFO entry shall become unoccupied when software writes 0x00000000 to the Logical/Transport Layer Error Detect CSR. The FIFO shall be considered full if all entries are occupied. The FIFO shall be considered empty if all entries are unoccupied.

The FIFO error capture function supplies the value that occupies a FIFO entry. The FIFO error capture function operates on enabled and disabled events, as defined in 1.3.1, “Logical/Transport Error Detect, Enable, and Capture CSRs”. The FIFO error capture function shall operate as follows:

- The FIFO error capture function value for the Logical/Transport Layer Error Detect CSR shall consist of all events detected since the last time the FIFO error capture function value occupied an entry in the FIFO.
- The FIFO error capture function value for all Logical/Transport Layer Capture CSRs shall be updated for every detected enabled event.

If the FIFO is not full and an enabled event has been detected, the current FIFO error capture function value shall occupy the next entry in the FIFO. If multiple enabled events occur simultaneously, at least one event shall occupy an entry in the FIFO.

The Logical/Transport Layer Error Detect CSR shall not be 0 when the FIFO is not empty. The Logical/Transport Layer Error Detect CSR shall be 0 when the FIFO is empty. The value for all Logical/Transport Layer Capture CSRs is undefined when the FIFO is empty.

1.4 System Software Notification of Error

System software is notified of logical, transport, and physical layer errors in two ways. An interrupt is issued to the local system by a device, the method of which is not defined in this specification, or a Maintenance port-write operation is issued by a device. Maintenance port-write operations are sent to a predetermined system host (defined in the Port-write Target deviceID CSR in Section 2.5.11). The sending device sets the Port-write Pending status bit in the Port *n* Error and Status CSRs. A 16 byte data payload of the Maintenance Port-write packet contains the contents of several CSRs, the port on the device that encountered the error condition (for port-based errors), and some optional implementation specific additional information as shown in Table 1-2. Software indicates that it has seen the port-write operation by clearing the Port-write Pending status bit.

The Component Tag CSR is defined in the *RapidIO Part 3: Common Transport Specification*, and is used to uniquely identify the reporting device within the system. The Port ID field, the Logical/Transport Layer Detect CSR defined in Section 2.5.3, and the Port *n* Error Detect CSRs defined in Section 2.5.15, are used to describe the encountered error condition.

Table 1-2. Port-Write Packet Data Payload for Error Reporting

Data Payload Byte Offset	Word	
0x0	Component Tag CSR	
0x4	Port *n* Error Detect CSRs	
0x8	Implementation specific	Port ID (byte)
0xC	Logical/Transport Layer Error Detect CSR	

1.5 Mechanisms for Software Debug

In most systems, it is difficult to verify the error handling software. The Error management extensions make some registers writable for easier debug.

The Logical/Transport Layer Error Detect CSR and the six error capture registers starting with Logical/Transport Layer High Address Capture CSR are writable by software to allow software debug of the system error recovery mechanisms. For software debug, software must write the desired information into the six error capture registers starting with Logical/Transport Layer High Address Capture CSR. If the Logical/Transport Layer Error Capture FIFO Implemented bit is set, the FIFO error capture function value is updated by the register writes. The next step is to write the Logical/Transport Layer Error Detect CSR to set an enabled error bit. If the Logical/Transport Layer Error Capture FIFO Implemented bit is cleared, this will lock the registers. If the Logical/Transport Layer Error Capture FIFO Implemented bit is set and the FIFO is not full, this will cause the FIFO error capture function

value to occupy a FIFO entry. When an error detect bit is set, the hardware will inform the system software of the error using its standard error reporting mechanism. After the error has been reported, the system software may read and clear registers as necessary to complete its error handling protocol testing.

The Port n Error Detect CSR and the five registers starting with the Port n Capture 0 CSR are also writable by software to allow software debug of the system error recovery and thresholding mechanism.

For debug when the Physical Layer Error Capture FIFO Implemented bit is cleared, software must write the Port n Attributes Capture CSR to set the Capture Valid Info bit and then the packet/control symbol information in the five registers starting with the Port n Capture 0 CSR.

For debug when the Physical Layer Error Capture FIFO Implemented bit is set, software must write the five registers starting with the Port n Capture 0 CSR, and then write to the Port n Error Detect CSR to update the FIFO error capture function values for these registers. When the Port n Attributes Capture CSR is written with a value that sets the Capture Valid Info bit, and the FIFO is not full, the FIFO error capture function value shall occupy a FIFO entry.

Each write of a non-zero value to the Port n Error Detect CSR shall cause the Error Rate Counter to increment if the corresponding error bit is enabled in the Port n Error Rate Enable CSR. When a threshold is reached, the hardware will inform the system software of the error using its standard error reporting mechanism. After the error has been reported, the system software may read and clear registers as necessary to complete its error handling protocol testing.

1.6 IDLE3 Port_Status Extension

The Error Management and Hot Swap functions can affect the operation of a port's input and output directions. Some of these conditions prevent acceptance or transmission of any packets by a processing element, causing errors to be detected by the link partner. In response to these error conditions, the link partner will initiate the standard error recovery protocol as defined in Part 6. Depending on the state of both link partners, software intervention may be necessary to resume packet exchange.

To enable software interrogation of link partner Error Management/Hot Swap status without using packets, the link-response port_status field is extended for IDLE3 devices as defined in Table 1-3. IDLE1 and IDLE2 devices shall not use port_status values other than those defined in Part 6.

Table 1-3. Port_status Field Definitions

Port_status bit number	Description
0-2	Reserved
3	Input Port Enabled The conditions defined for this bit to be set are extended as follows: - The Port *n* Control CSRs “Port Lockout” bit is cleared.
4-6	Reserved
7	Output Port Enabled The conditions defined for this bit to be set are extended as follows: - The Port *n* Control CSRs “Port Lockout” bit is cleared.
8	Reserved
9	Output Port Failed. This bit shall be asserted when at least one of the following conditions is true, otherwise de-asserted: - The Port *n* Error and Status CSRs “Output Failed-Encountered” bit is set - The Port *n* Error Detect CSRs “Link Uninit Packet Discard Active” bit is set - An implementation specific condition exists which forces continuous output port packet discard
10-11	Reserved

Chapter 2 Error Management Registers

2.1 Introduction

This section describes the Error Management Extended Features block, and adds a number of new bits to the existing standard physical layer registers. ‘End-point only’ and ‘switch only’ register bits shall be considered reserved when the registers are implemented on devices for which these bits are not required.

2.2 Additions to Existing Registers

2.2.1 Port *n* Control CSRs

The following bits are added to the RapidIO Part 6: LP-Serial Physical Layer Specification Port *n* Control CSRs.

Table 2-1. Bit Settings for Port *n* Control CSRs

Bit	Name	Reset Value	Description
28	Stop on Port Failed-encountered Enable	0b0	This bit is used with the Drop Packet Enable bit to force certain behavior when the Error Rate Failed Threshold has been met or exceeded. See Section 1.2.4 of the Part 8: Error Management Extensions for detailed requirements.
29	Drop Packet Enable	0b0	This bit is used with the Stop on Port Failed-encountered Enable bit to force certain behavior when the Error Rate Failed Threshold has been met or exceeded. See Section 1.2.4 of the Part 8: Error Management Extensions for detailed requirements.
30	Port Lockout	0b0	When this bit is cleared, the packets that may be received and issued are controlled by the state of the Output Port Enable and Input Port Enable bits in the Port *n* Control CSR. When this bit is set, this port is stopped and is not enabled to issue or receive any packets; the input port can still follow the training procedure and can still send and respond to link-requests; all received packets return packet-not-accepted control symbols to force an error condition to be signaled by the sending device

2.2.2 Port *n* Error and Status CSRs

The following bits are added to the RapidIO Part 6: LP-Serial Physical Layer Specification Port *n* Error and Status CSRs.

Table 2-2. Bit Settings for Port *n* Error and Status CSRs

Bit	Name	Reset Value	Description
5	Output Packet-dropped	0b0	Output port has discarded a packet. Once set remains set until written with a logic 1 to clear.
6	Output Failed-encountered	0b0	Output port has encountered a failed condition, meaning that the port's failed error threshold has been reached in the Port *n* Error Rate Threshold register. Once set remains set until written with a logic 1 to clear. Receipt of a reset-port request shall clear this bit to 0. State machines associated with this bit shall be reset to their power-up state.
7	Output Degraded-encountered	0b0	Output port has encountered a degraded condition, meaning that the port's degraded error threshold has been reached in the Port *n* Error Rate Threshold register. Once set remains set until written with a logic 1 to clear.

2.3 New Error Management Registers

This section describes the Extended Features block (EF_ID=0x0007 or EF_ID=0x0017) that allows an external processing element to manage the error status and reporting for a processing element. This chapter only describes registers or register bits defined by this extended features block. All registers are 32-bits and aligned to a 32-bit boundary.

Table 2-3 describes the required behavior for accesses to reserved register bits and reserved registers for the RapidIO Extended Features register space,

Table 2-3. Extended Feature Space Reserved Access Behavior

Byte Offset	Space Name	Item	Initiator behavior	Target behavior
0x100–FFFC	Extended Features Space	Reserved bit	read - ignore returned value[1]	read - return logic 0
			write - preserve current value[2]	write - ignored
		Implementation-defined bit	read - ignore returned value unless implementation-defined function understood	read - return implementation-defined value
			write - preserve current value if implementation-defined function not understood	write - implementation-defined
		Reserved register	read - ignore returned value	read - return logic 0s
			write -	write - ignored

[1]Do not depend on reserved bits being a particular value; use appropriate masks to extract defined bits from the read value.

[2]All register writes shall be in the form: read the register to obtain the values of all reserved bits, merge in the desired values for defined bits to be modified, and write the register, thus preserving the value of all reserved bits.

2.4 Register Map

The register map for the error management registers shall be as specified by Table 2-4. This register map is currently only defined for devices with up to 16 RapidIO ports, but can be extended or shortened if more or less port definitions are required for a device. For example, a device with four RapidIO ports is only required to use register map space corresponding to offsets [EF_PTR+0x00] to [EF_PTR+0x13C]. Register map offset [EF_PTR+0x140] can be used for another Extended Features block.

The registers that appear in the Error Management/Hot Swap Extensions Register Block vary based on the functionality indicated in the Error Management/Hot Swap Extensions Block CAR. Table 2-4 describes what registers shall be implemented based on the value of the Error Management/Hot Swap Extensions Block CAR. The register offsets and names are listed, along with three columns that indicate which registers must be implemented. An "X" in the column means that the register shall be implemented.

Table 2-4. Error Management/Hot Swap Extensions Register Requirements

	Block Byte Offset	Register Name	Error Mgmt Only	Hot Swap & Error Mgmt	Hot Swap Only
General	0x0	Error Management/Hot Swap Extensions Block Header	X	X	X
	0x4	Error Management/Hot Swap Extensions Block CAR	X	X	X
	0x8	Logical/Transport Layer Error Detect CSR	X	X	-
	0xC	Logical/Transport Layer Error Enable CSR	X	X	-
	0x10	Logical/Transport Layer High Address Capture CSR	X	X	-
	0x14	Logical/Transport Layer Address Capture CSR	X	X	-
	0x18	Logical/Transport Layer Device ID Capture CSR	X	X	-
	0x1C	Logical/Transport Layer Control Capture CSR	X	X	-
	0x20	Logical/Transport Layer Dev32 Destination ID Capture CSR	X	X	-
	0x24	Logical/Transport Layer Dev32 Source ID Capture CSR	X	X	-
	0x28	Port-write Target deviceID CSR	X	X	X
	0x2C	Packet Time-to-live CSR	X	X	X
	0x30	Port-write Dev32 Target deviceID CSR	X	X	X
	0x34	Port-Write Transmission Control CSR	X	X	X
	0x38-3C	Reserved			

Table 2-4. Error Management/Hot Swap Extensions Register Requirements

	Block Byte Offset	Register Name	Error Mgmt Only	Hot Swap & Error Mgmt	Hot Swap Only
Port 0	0x40	Port 0 Error Detect CSR	X	X	X
	0x44	Port 0 Error Rate Enable CSR	X	X	X
	0x48	Port 0 Attributes Capture CSR	X	X	-
	0x4C	Port 0 Capture 0 CSR	X	X	-
	0x50	Port 0 Capture 1 CSR	X	X	-
	0x54	Port 0 Capture 2 CSR	X	X	-
	0x58	Port 0 Capture 3 CSR	X	X	-
	0x5C	Port 0 Capture 4 CSR	X	X	-
	0x60-64	Reserved			
	0x68	Port 0 Error Rate CSR	X	X	-
	0x6C	Port 0 Error Rate Threshold CSR	X	X	-
	0x70	Port 0 Link Uninit Discard Timer CSR	-	X	X
	0x74-78	Reserved			
	0x7C	Port 0 Error Detect FIFO CSR	X	X	
Port 1	0x80	Port 1 Error Detect CSR	X	X	X
	0x84	Port 1 Error Rate Enable CSR	X	X	X
	0x88	Port 1 Attributes Capture CSR	X	X	-
	0x8C	Port 1 Capture 0 CSR	X	X	-
	0x90	Port 1 Capture 1 CSR	X	X	-
	0x94	Port 1 Capture 2 CSR	X	X	-
	0x98	Port 1 Capture 3 CSR	X	X	-
	0x9C	Port 1 Capture 4 CSR	X	X	-
	0xA0-A4	Reserved			
	0xA8	Port 1 Error Rate CSR	X	X	-
	0xAC	Port 1 Error Rate Threshold CSR	X	X	-
	0xB0	Port 1 Link Uninit Discard Timer CSR	-	X	X
	0xB4-B8	Reserved			
	0xBC	Port 1 Error Detect FIFO CSR	X	X	
Ports 2-14	0xC0–3FC	Assigned to Port 2-14 CSRs Register implementation requirements are the same as for port 0			

Table 2-4. Error Management/Hot Swap Extensions Register Requirements

	Block Byte Offset	Register Name	Error Mgmt Only	Hot Swap & Error Mgmt	Hot Swap Only
Port 15	0x400	Port 15 Error Detect CSR	X	X	X
	0x404	Port 15 Error Rate Enable CSR	X	X	X
	0x408	Port 15 Attributes Capture CSR	X	X	-
	0x40C	Port 15 Capture 0 CSR	X	X	-
	0x410	Port 15 Capture 1 CSR	X	X	-
	0x414	Port 15 Capture 2 CSR	X	X	-
	0x418	Port 15 Capture 3 CSR	X	X	-
	0x41C	Port 15 Capture 4 CSR	X	X	-
	0x420-424	Reserved			
	0x428	Port 15 Error Rate CSR	X	X	-
	0x42C	Port 15 Error Rate Threshold CSR	X	X	-
	0x430	Port 15 Link Uninit Discard Timer CSR	-	X	X
	0x434-438	Reserved			
	0x43C	Port 15 Error Detect FIFO CSR	X	X	

2.5 Command and Status Registers (CSRs)

Refer to Table 2-3 for the required behavior for access to reserved registers and register bits.

2.5.1 Error Management Extensions Block Header (Block Offset 0x0)

This register contains the EF_PTR to the next EF_BLK and the EF_ID that identifies this as the error management extensions block header. The use and meaning of the bits shall be as specified in Table 2-5. The register is read-only.

Table 2-5. Bit Settings for Error Management Extensions Block Header

Bit	Name	Reset Value	Description
0-15	EF_PTR		Hard-wired pointer to the next block in the data structure, if one exists
16-31	EF_ID	0x0007 or 0x0017	Devices which implement the standard Error Management Registers, with or without Hot Swap Extensions, shall use an EF_ID value of 0x07. Devices which only implement the Hot Swap Extensions, shall use an EF_ID value of 0x17.

2.5.2 Error Management/Hot Swap Extension Block CAR (Block Offset 0x4)

This register indicates the supported Error Management Extension and Hot Swap Extension functionality. The use and meaning of the bits shall be as specified in Table 2-6. The register is read-only.

Table 2-6. Bit Settings for Error Management/Hot Swap Extension Block CAR

Bit	Name	Reset Value	Description
0	Error Management Extensions Not Implemented	Implementation Specific	Indicates whether or not Error Management Extensions functionality (and registers) is supported. 0b0 - all registers and bit fields specific to Error Management Extensions shall be supported. 0b1 - all registers and/or bit fields specific to Error Management Extensions may not be supported.
1	Hot Swap Extensions Implemented	Implementation Specific	Indicates whether or not Hot Swap functionality and registers are supported. 0b0 - all registers and bit fields specific to Hot Swap Extensions support may not be supported. 0b1 - all registers and bit fields specific to Hot Swap Extensions support shall be supported.

Table 2-6. Bit Settings for Error Management/Hot Swap Extension Block CAR

Bit	Name	Reset Value	Description
2	Physical Layer Error Capture FIFO Implemented	Implementation Specific	Indicates whether or not a FIFO for capture of multiple physical layer events is implemented. 0b0 - all registers, bit fields and behavior specific to FIFO capture of multiple physical layer events may not be supported 0b1 - all registers, bit fields and behavior specific to FIFO capture of multiple physical layer events shall be supported by all ports on the device
3	Logical/Transport Layer Error Capture FIFO Implemented	Implementation Specific	Indicates whether or not a FIFO for capture of multiple logical/transport layer events is implemented. 0b0 - all registers, bit fields and behavior specific to FIFO capture of multiple logical/transport layer events may not be supported 0b1 - all registers, bit fields and behavior specific to FIFO capture of multiple logical/transport layer events shall be supported
4-31	—		Reserved

2.5.3 Logical/Transport Layer Error Detect CSR (Block Offset 0x08)

This register indicates the error that was detected by the Logical or Transport logic layer. Multiple bits may get set in the register if simultaneous errors are detected during the same clock cycle that the errors are logged. The use and meaning of the bits shall be as specified in Table 2-7. Unless otherwise specified, the bits in this register are read/write.

Table 2-7. Bit Settings for Logical/Transport Layer Error Detect CSR

Bit	Name	Reset Value	Description
0	IO error response	0b0	Received a response of ‘ERROR’ for an IO Logical Layer Request. (end point device only)
1	Message error response	0b0	Received a response of ‘ERROR’ for an MSG Logical Layer Request. (end point device only)
2	GSM error response	0b0	Received a response of ‘ERROR’ for a GSM Logical Layer Request. (end point device only)
3	Message Format Error	0b0	Received MESSAGE packet data payload with an invalid size or segment (MSG logical) (end point device only)
4	Illegal transaction decode	0b0	Received a supported request/response packet with undefined field values (IO/MSG/GSM logical) (switch or endpoint device)
5	Illegal transaction target error	0b0	Received a packet that contained a destination ID that is not defined for this processing element. End points with multiple ports and a built-in switch function may not report this as an error (Transport) (optional) (switch or end point device)

Table 2-7. Bit Settings for Logical/Transport Layer Error Detect CSR

Bit	Name	Reset Value	Description
6	Message Request Timeout	0b0	A required message request has not been received within the specified timeout interval (MSG logical) (end point device only)
7	Packet Response Timeout	0b0	A required response has not been received within the specified time out interval (IO/MSG/GSM logical) (end point device only)
8	Unsolicited Response	0b0	An unsolicited/unexpected Response packet was received (IO/MSG/GSM logical; only Maintenance response for switches) (switch or endpoint device)
9	Unsupported Transaction	0b0	A transaction is received that is not supported in the Destination Operations CAR (IO/MSG/GSM logical; only Maintenance port-write for switches) (switch or endpoint device)
10-21	—		Reserved
22	Lost Tick Error Status	0b0	Indicates the current status of the MECS Tick Interval CSR Lost Tick Error Status bit. The Lost Tick Error Status bit must be cleared by writing to the MECS Tick Interval CSR. If the Lost Tick Error Status bit is not defined, this bit is reserved. (end point device only)
23	Lost TSG Sync Error Status	0b0	Indicates the current status of the MECS Tick Interval CSR Lost TSG Sync Error Status bit. The Lost TSG Sync Error Status bit must be cleared by writing to the MECS Tick Interval CSR. If the Lost TSG Sync Error Status bit is not defined, this bit is reserved. (end point device only)
24-31	Implementation Specific error	0x00	An implementation specific error has occurred. (switch or end point device)

2.5.4 Logical/Transport Layer Error Enable CSR (Block Offset 0x0C)

This register contains the bits that control if an error condition locks the Logical/Transport Layer Error Detect and Capture registers and is reported to the system host. Without exception, bit “b” of this register controls the capture and reporting of the error whose occurrence is indicated by bit “b” of the Logical/Transport Layer Error Detect CSR. The use and meaning of the bits shall be as specified in Table 2-8. Unless otherwise specified, the bits in this register are read/write.

Table 2-8. Bit Settings for Logical/Transport Layer Error Enable CSR

Bit	Name	Reset Value	Description
0	IO error response enable	0b0	Enable reporting of an IO 'ERROR' response. Save and lock original request transaction information in all Logical/Transport Layer Capture CSRs. (end point device only)
1	Message error response enable	0b0	Enable reporting of a Message 'ERROR' response. Save and lock original request transaction information in all Logical/Transport Layer Capture CSRs. (end point device only)
2	GSM error response enable	0b0	Enable reporting of a GSM 'ERROR' response. Save and lock original request transaction capture information in all Logical/Transport Layer Capture CSRs. (end point device only)
3	Message Format Error enable	0b0	Enable reporting of a MESSAGE packet data payload with an invalid size or segment (MSG logical). Save and lock transaction capture information in Logical/Transport Layer Device ID and Control Capture CSRs. (end point device only)
4	Illegal transaction decode enable	0b0	Enable reporting of a supported request/response packet with undefined field values (IO/MSG/GSM logical). Save and lock transaction capture information in Logical/Transport Layer Device ID and Control Capture CSRs. (switch or end-point device)
5	Illegal transaction target error enable	0b0	Enable reporting of a packet that contains a destination ID that is not defined for this processing element. Save and lock transaction capture information in Logical/Transport Layer Device ID and Control Capture CSRs. (optional) (switch or end point device)
6	Message Request timeout error enable	0b0	Enable reporting of a Message Request timeout error. Save and lock transaction capture information in Logical/Transport Layer Device ID and Control Capture CSRs for the last Message request segment packet received. (end point device only)
7	Packet Response Timeout error enable	0b0	Enable reporting of a packet response timeout error. Save and lock original request address in Logical/Transport Layer Address Capture CSRs. Save and lock original request Destination ID in Logical/Transport Layer Device ID Capture CSR. (end point device only)
8	Unsolicited Response error enable	0b0	Enable reporting of receiving an unsolicited/unexpected Response packet (IO/MSG/GSM logical; only Maintenance responses for switches). Save and lock transaction capture information in Logical/Transport Layer Device ID and Control Capture CSRs. (switch or end-point device)
9	Unsupported Transaction error enable	0b0	Enable reporting of an unsupported transaction error. Save and lock transaction capture information in Logical/Transport Layer Device ID and Control Capture CSRs. (switch or end-point device)
10-21	—		Reserved
22	Lost Tick Error Enable	0b0	Enable reporting of the current status of the MECS Tick Interval CSR Lost Tick Error Status bit. The Logical/Transport Layer Device ID and Control Capture CSRs shall not lock when this error is detected. If the Lost Tick Error Status bit is not defined, this bit is reserved. (end point device only)

Table 2-8. Bit Settings for Logical/Transport Layer Error Enable CSR

Bit	Name	Reset Value	Description
23	Lost TSG Sync Error Enable	0b0	Enable reporting of the MECS Tick Interval CSR Lost TSG Sync Error Status bit. The Logical/Transport Layer Device ID and Control Capture CSRs shall not lock when this error is detected. If the Lost TSG Sync Error Status bit is not defined, this bit is reserved. (end point device only)
24-31	Implementation Specific error enable	0x00	Enable reporting of an implementation specific error has occurred. Save and lock capture information in appropriate Logical/Transport Layer Capture CSRs.

2.5.5 Logical/Transport Layer High Address Capture CSR (Block Offset 0x10)

This register contains error information. It is locked when a Logical/Transport error is detected and the corresponding enable bit is set. This register is required for end point devices that support 66 or 50 bit addresses, and for all switch devices which support operation with IDLE3. The use and meaning of the bits shall be as specified in Table 2-9. Unless otherwise specified, the bits in this register are read/write.

Table 2-9. Bit Settings for Logical/Transport Layer High Address Capture CSR

Bit	Name	Reset Value	Description
0-31	address[0-31]	All 0s	For switch devices, this field may capture implementation specific information for detected logical/transport layer errors. When an endpoint detects an error related to a Logical I/O packet, this field may capture implementation specific information when the address size is 34 bits or less. When an endpoint detects an error related to a Logical I/O packet and the address size is 50 bits, bits 0 to 15 of this field may capture implementation specific information and bits 16 to 31 of this field shall contain address bits 2-17. When an endpoint detects an error related to a Logical I/O packet and the address size is 66 bits, this field shall capture address bits 2-33. For all other logical/transport layer errors, this field may contain implementation specific information.

2.5.6 Logical/Transport Layer Address Capture CSR (Block Offset 0x14)

This register contains error information. It is locked when a Logical/Transport error is detected and the corresponding enable bit is set. The use and meaning of the bits shall be as specified in Table 2-10. Unless otherwise specified, the bits in this register are read/write.

Table 2-10. Bit Settings for Logical/Transport Layer Address Capture CSR

Bit	Name	Reset Value	Description
0-28	address[32-60]	All 0s	When a logical/transport error is detected for a maintenance packet which has an offset, the offset is found in the least significant 21 bits of this field. Otherwise, when a Logical/transport layer error is detected by a switch device, this field may contain implementation specific information. When an endpoint detects an error with a Logical I/O transaction, this field contains the least significant 29 bits of the address field. When an endpoint detects and error for any other transaction type, this field may contain implementation specific information.
29	Implementation Specific	0b0	This field may contain implementation specific information.
30-31	xamsbs	0b00	When an endpoint detects an error with a logical I/O transaction, this field shall contain the extended address bits of the address associated with the error (for requests, for responses if available). For all other errors detected by an endpoint, and in switch devices, this field may contain implementation specific information.

2.5.7 Logical/Transport Layer Device ID Capture CSR (Block Offset 0x18)

This register contains error information. It is locked when an error is detected and the corresponding enable bit is set. The use and meaning of the bits shall be as specified in Table 2-11. Unless otherwise specified, the bits in this register are read/write.

Table 2-11. Bit Settings for Logical/Transport Layer Device ID Capture CSR

Bit	Name	Reset Value	Description
0-7	MSB destinationID	0x00	Most significant byte of the destinationID associated with the error (Dev16 systems only)
8-15	destinationID	0x00	The destinationID associated with the error (Dev8 and Dev16 systems only)
16-23	MSB sourceID	0x00	Most significant byte of the sourceID associated with the error (Dev16 systems only)
24-31	sourceID	0x00	The sourceID associated with the error (Dev8 and Dev16 systems only)

2.5.8 Logical/Transport Layer Control Capture CSR (Block Offset 0x1C)

This register contains error information. It is locked when a Logical/Transport error is detected and the corresponding enable bit is set. The use and meaning of the bits shall be as specified in Table 2-12. Unless otherwise specified, the bits in this register are read/write.

Table 2-12. Bit Settings for Logical/Transport Layer Control Capture CSR

Bit	Name	Reset Value	Description
0-3	ftype	0x0	Format type associated with the error
4-7	ttype	0x0	Transaction type associated with the error. If the format type does not include a ttype field, this field may contain implementation specific information.
8-15	msg info	0x00	letter, mbox, and msgseg for the last Message request received for the mailbox that had an error (Message errors only). For non-Message errors, this field may contain implementation specific information.
16-31	Implementation specific	0x0000	Implementation specific information associated with the error

2.5.9 Logical/Transport Layer Dev32 Destination ID Capture CSR (Block Offset 0x20)

This register contains error information. It is locked when a Logical/Transport error is detected and the corresponding enable bit is set. This register shall be implemented for devices that have bit 19 (Dev32 Support) set in the Processing Element Features CAR. The use and meaning of the bits shall be as specified in Table 2-13. This register is read/write.

Table 2-13. Bit Settings for Logical/Transport Layer Dev32 Destination ID Capture CSR

Bit	Name	Reset Value	Description
0-31	Dev32 DestID	All 0’s	The Dev32 destination ID associated with the error.

2.5.10 Logical/Transport Layer Dev32 Source ID Capture CSR (Block Offset 0x24)

This register contains error information. It is locked when a Logical/Transport error is detected and the corresponding enable bit is set. This register shall be implemented for devices that have bit 19 (Dev32 Support) set in the Processing Element Features CAR. The use and meaning of the bits shall be as specified in Table 2-14. This register is read/write.

Table 2-14. Bit Settings for Logical/Transport Layer Dev32 Source ID Capture CSR

Bit	Name	Reset Value	Description
0-31	Dev32 SrcID	All 0's	The Dev32 source ID associated with the error.

2.5.11 Port-Write Target deviceID CSR (Block Offset 0x28)

This register contains the target Dev8 or Dev16 deviceID to be used when a device generates a Maintenance port-write operation to report errors to a system host. The use and meaning of the bits shall be as specified in Table 2-15. Unless otherwise specified, the bits in this register are read/write.

Table 2-15. Bit Settings for Port-Write Target deviceID CSR

Bit	Name	Reset Value	Description
0-7	Dev16_deviceID_msb	0x00	This is the most significant byte of the port-write target deviceID (Dev16 deviceIDs only).
8-15	Dev8_deviceID	0x00	This is the port-write target Dev8 deviceID, or least significant byte of the Dev16 deviceID.
16	Dev8_or_16	0b0	Dev8 or Dev16 deviceID size to use for a port-write 0b0 - Port-writes originated by this device shall use Dev8 deviceIDs 0b1 - Port-writes originated by this device shall use Dev16 deviceIDs This bit field controls the deviceID size to use for a port-write when Dev32_PW is 0.
17	Dev32_PW	0b0	This bit field shall be implemented for devices that have bit 19 (Dev32 Support) set in the Processing Element Features CAR. This bit field controls the use of Dev32 deviceID size for a port-write 0b0 - Port-write deviceID size shall be controlled by the Dev8_or_16 field 0b1 - Port-writes originated by this device shall use the Dev32 deviceID defined in the 2.5.13, "Port-write Dev32 Target deviceID CSR
18-31	—		Reserved

2.5.12 Packet Time-to-live CSR (Block Offset 0x2C)

The Packet Time-to-live register specifies the length of time that a packet is allowed to exist within a switch device. The maximum value of the Time-to-live variable (0xFFFF) shall correspond to 100 msec. +/-34%. The resolution (minimum step size) of the Time-to-live variable shall be (maximum value of Time-to-live)/(2^{16}-1). The reset value is all logic 0s, which disables the Time-to-live function so that a packet never times out. This register is not required for devices without switch functionality. The use and meaning of the bits shall be as specified in Table 2-16. Unless otherwise specified, the bits in this register are read/write.

Table 2-16. Bit Settings for Packet Time-to-live CSR

Bit	Name	Reset Value	Description
0-15	Time-to-live value	0x0000	Maximum time that a packet is allowed to exist within a switch device
16-31	—		Reserved

2.5.13 Port-write Dev32 Target deviceID CSR (Block Offset 0x30)

This register contains the Dev32 target deviceID to be used when a device generates a Maintenance port-write operation to report errors to a system host and the Dev32_PW bit is set. This register shall be implemented for devices that have bit 19 (Dev32 Support) set in the Processing Element Features CAR. The use and meaning of the bits shall be as specified in Table 2-17. Unless otherwise specified, the bits in this register are read/write.

Table 2-17. Bit Settings for Port-Write Dev32 Target deviceID CSR

Bit	Name	Reset Value	Description
0-31	Dev32_deviceID	All 0's	The port-write Dev32 target device ID.

2.5.14 Port-Write Transmission Control CSR (Block Offset 0x34)

The Port-Write transmission control CSR determines whether port-write notification is enabled or disabled for the device. The use and meaning of the bits shall be as specified in Table 2-18. Unless otherwise specified, the bits in this register are read/write.

Table 2-18. Bit Settings for Port-Write Transmission Control CSR

Bit	Name	Reset Value	Description
0-30	—		Reserved
31	Port-write Transmission Disable	0b0	0 - Enabled events for the device shall cause port-writes to be generated 1 - Enabled events for the device shall not cause new port-writes to be generated. Previously generated port-writes may be transmitted after this bit is set.

2.5.15 Port *n* Error Detect CSR (Block Offset 0x40, 80,..., 400)

The Port *n* Error Detect Register indicates the physical layer errors that have been detected by the Port *n* hardware since the register was last cleared. The register is cleared by software writing the register with the data 0x0000_0000.

The use and meaning of the bits shall be as specified in Table 2-19. Unless otherwise specified, the bits in this register are read/write.

The two right-most columns in Table 2-19 indicate which bit fields must be implemented for Error Management Extensions and Hot Swap Extensions. An “X” in these columns means that the bit shall be implemented for that extension.

Table 2-19. Bit Settings for Port *n* Error Detect CSRs

Bit	Name	Reset Value	Description	Error Mgmt	Hot Swap
0	Implementation specific error	0b0	An implementation specific error has been detected	X	-
1	Link OK to Uninit Transition	0b0	The link has transitioned from a link initialized to link uninitialized state.	-	X
2	Link Uninit Packet Discard Active	0b0	The Link Uninit Discard Timer CSR period has expired.	-	X
3	Link Uninit to OK Transition	0b0	The link has transitioned from a link uninitialized to link initialized state.	-	X
4-7	—		Reserved		
8	Deprecated	0b0	Deprecated	X	-
9	Received corrupt control symbol	0b0	Received a control symbol with a bad CRC value or Received an incorrect sequence of control symbol codewords (for example, CSE or CSEB without a preceding CSB, or a CSB or CSEB that is not followed by a CSE or CSEB (IDLE3)	X	-
10	Received acknowledge control symbol with unexpected ackID	0b0	Received a packet-accepted or packet-retry control symbol with an unexpected ackID. "Error Recovery with ackID in PNA Enabled" is set in the Port n Latency Optimization CSR and a packet-not-accepted control symbol is received with an unexpected ackID.	X	-

Table 2-19. Bit Settings for Port *n* Error Detect CSRs

Bit	Name	Reset Value	Description	Error Mgmt	Hot Swap
11	Received packet-not-accepted control symbol	0b0	Received packet-not-accepted control symbol	X	-
12	Received packet with unexpected ackID	0b0	Received packet with unexpected ackID value - out-of-sequence ackID	X	-
13	Received packet with bad CRC	0b0	Received packet with a bad CRC value	X	-
14	Received packet exceeds maximum allowed size	0b0	Received packet that exceeds the maximum allowed size	X	-
15	Received illegal or invalid character	0b0	Received an 8b/10b code-group that is invalid (a code-group that does not have a 8b/10b decode given the current running disparity) or illegal (a code-group that is valid but whose use is not allowed by the LP-Serial protocol). This bit may be set in conjunction with bit 29, Delineation error. The implementation of this bit is optional, but strongly recommended. (IDLE1 and IDLE2) or Bit Interleaved Parity (BIP) check failed on at least one lane (IDLE3)	X	-
16	Received data character in IDLE1 sequence	0b0	Received a data character in an IDLE1 sequence. This bit may be set in conjunction with bit, 29 Delineation error. The implementation of this bit is optional, but strongly recommended.	X	-
17	Loss of descrambler synchronization	0b0	Loss of receiver descrambler synchronization while receiving scrambled control symbol and packet data. This bit shall be implemented only if port n supports descrambling of packet and control symbol data.	X	-
18	Invalid Ordered Sequence	0b0	Received an invalid ordered sequence on at least one lane (IDLE3)	X	-
19-25	—		Reserved		
26	Non-outstanding ackID	0b0	When there are outstanding ackIDs, a link_response was received with an ackID that is not outstanding	X	-
27	Protocol error	0b0	An unexpected control symbol was received	X	-
28	Deprecated	0b0	Deprecated	X	-
29	Delineation error	0b0	Received an 8b/10b code-group that is invalid (a code-group that does not have a 8b/10b decode given the current running disparity), illegal (a code-group that is valid, but whose use is not allowed by the LP-Serial protocol) or that is in a position in the received code-group stream that is not allowed by the LP-Serial protocol (IDLE1 and IDLE2) or Received a 64b/67b codeword whose type and !type bit values are the same (IDLE3)	X	-
30	Unsolicited acknowledgement control symbol	0b0	An unsolicited packet acknowledgement control symbol was received	X	-

Table 2-19. Bit Settings for Port *n* Error Detect CSRs

Bit	Name	Reset Value	Description	Error Mgmt	Hot Swap
31	Link timeout	0b0	A packet acknowledgement or link-response control symbol was not received within the specified timeout interval	X	-

2.5.16 Port *n* Error Rate Enable CSR (Block Offset 0x44, 84,..., 404)

The two right-most columns of Table 2-20 indicate which bit fields must be implemented for Error Management Extensions and Hot Swap Extensions. An “X” in these columns means that the bit shall be implemented for that extension.

If Hot Swap Extension is supported in this register, the Hot Swap bits enable the event notification and packet discard for the detected event.

When the Error Management Extension is supported, the Error Management bits when set cause specific detected errors to increment the error rate counter in the Port *n* Error Rate Threshold Register and capture information about the error in, and then lock, the Port *n* Capture 0-4 CSRs. Without exception, bit “b” of this register controls the capture and counting of the detected error whose occurrence is indicated by bit “b” of the Port *n* Error Detect CSR.

The use and meaning of the bits shall be as specified in Table 2-20. Unless otherwise specified, the bits in this register are read/write.

Table 2-20. Bit Settings for Port *n* Error Rate Enable CSRs

Bit	Name	Reset Value	Description	Error Mgmt	Hot Swap
0	Implementation specific error enable	0b0	Enable error rate counting of implementation specific errors	X	-
1	Link OK to Uninit Transition Enable	0b0	Enable event notification for when the link has transitioned from a link initialized to link uninitialized state.	-	X
2	Link Uninit Packet Discard Active Enable	0b0	Enable event notification for Link Uninit Packet Discard Timer events.	-	X
3	Link Uninit to OK Transition Enable	0b0	Enable event notification for when the link has transitioned from a link uninitialized to link initialized state.	-	X
4-7	—		Reserved		
8	Deprecated	0b0	Deprecated	X	-
9	Received corrupt control symbol enable	0b0	Enable error rate counting of a corrupt control symbol	X	-

Table 2-20. Bit Settings for Port *n* Error Rate Enable CSRs

Bit	Name	Reset Value	Description	Error Mgmt	Hot Swap
10	Received acknowledge control symbol with unexpected ackID enable	0b0	Enable error rate counting of: - packet-accepted or packet-retry control symbol is received with an unexpected ackID - If "Error Recovery with ackID in PNA Enabled" is set in the Port n Latency Optimization CSR and a packet-not-accepted control symbol is received with an unexpected ackID	X	-
11	Received packet-not-accepted control symbol enable	0b0	Enable error rate counting of received packet-not-accepted control symbols.	X	-
12	Received packet with unexpected ackID enable	0b0	Enable error rate counting of packet with unexpected ackID value - out-of-sequence ackID	X	-
13	Received packet with bad CRC enable	0b0	Enable error rate counting of packet with a bad CRC value	X	-
14	Received packet exceeds maximum allowed size enable	0b0	Enable error rate counting of packet that exceeds the maximum allowed size	X	-
15	Received illegal or invalid character enable	0b0	Enable error rate counting of reception of illegal or invalid characters (IDLE1 or IDLE2), or failed BIP check (IDLE3). This bit shall be implemented only if bit 15 (Received illegal or invalid character) of the Port *n* Error Detect CSR is implemented.	X	-
16	Received data character in an IDLE1 sequence enable	0b0	Enable error rate counting of reception of a data character in an IDLE1 sequence. This bit shall be implemented only if bit 16 (Received data character in IDLE1 sequence) of the Port *n* Error Detect CSR is implemented.	X	-
17	Loss of descrambler synchronization enable	0b0	Enable error rate counting of loss of receiver descrambler synchronization when scrambled control symbol and packet data is being received. This bit shall be implemented only if bit 17 (Loss of descrambler synchronization) of the Port *n* Error Detect CSR is implemented.	X	-
18	Invalid ordered sequence enable	0b0	Enable error rate counting of invalid ordered sequence reception for all enabled lanes. This bit shall be implemented only if bit 18 (Invalid ordered sequence) of the Port *n* Error Detect CSR is implemented. (IDLE3)	X	-
19-25	—		Reserved		
26	Non-outstanding ackID enable	0b0	Enable error rate counting of link-responses received with an ackID that is not outstanding when there are outstanding ackIDs	X	-
27	Protocol error enable	0b0	Enable error rate counting of received unexpected control symbol symbols	X	-
28	Deprecated	0b0	Deprecated	X	-

Table 2-20. Bit Settings for Port *n* Error Rate Enable CSRs

Bit	Name	Reset Value	Description	Error Mgmt	Hot Swap
29	Delineation error enable	0b0	Enable error rate counting of:- Reception of an 8b/10b code-group that is invalid, illegal, or is in a position in the received code-group stream that is not allowed by the LP-Serial protocol (IDLE1 and IDLE2) - Reception of 64b/67b codeword whose type and !type bits are the same (IDLE3)	X	-
30	Unsolicited acknowledgement control symbol enable	0b0	Enable error rate counting of received unsolicited packet acknowledgement control symbols	X	-
31	Link timeout enable	0b0	Enable error rate counting of link timeout errors	X	-

2.5.17 Port *n* Attributes Capture CSR (Block Offset 0x48, 88,..., 408)

This register indicates the type of information captured in the Port *n* Capture 0-4 CSRs. If multiple errors are detected during the same clock cycle, one of the detected errors shall be selected for capture and the captured error shall be indicated in the Error type field. When multiple errors are detected during the same clock cycle, the error selected for capture is implementation specific.

The use and meaning of the bits shall be as specified in Table 2-21. Bits 0-30 of this register shall be valid and locked when the Capture valid info bit is set. Unless otherwise specified, the bits in this register are read/write.

Table 2-21. Bit Settings for Port *n* Attributes Capture CSRs

Bit	Name	Reset Value	Description
0-2	Info type	0b000	Type of information logged 0b000 - packet 0b001 - reserved 0b010 - Control Symbol 24 0b011 - Control Symbol 48 0b100 - implementation specific (capture register contents are implementation specific) 0b101 - Control Symbol 64 0b110 - deprecated 0b111 - reserved
3-7	Error type	0x00	The encoded value of the bit in the Port *n* Error Detect CSR that describes the error captured in the Port *n* Capture 0-4 CSRs.
8-27	Implementation Dependent	All 0s	Implementation specific error information. If port *n* is operating with IDLE1 or IDLE2, the following should be implemented: • If the Info_type is “packet”, the “control” bits of packet characters 0-15 should be captured in bits 8-23, respectively. • If the Info_type is “Control Symbol 24”, the “control” bits of delimited control symbol characters 0-3 should be captured in bits 8-11, respectively. • If the Info_type is “Control Symbol 48 symbol”, the “control” bits of delimited control symbol characters 0-7 should be captured in bits 8-15, respectively. If port *n* is operating with IDLE3, the following should be implemented: • The inverted, !type and type bits of the codewords that are captured in the Port *n* Capture 0-4 CSRs should be captured here.
28-30	—		Reserved
31	Capture valid info	0b0	This bit is set by hardware to indicate that the Port *n* Capture 0-4 CSRs and the other bits in this register contain valid information and are locked. This bit is cleared and the Port *n* Capture 0-4 CSRs and the other bits in this register are unlocked when software writes 0b0 to the bit.

2.5.18 Port *n* Capture 0 CSR (Block Offset 0x4C, 8C,..., 40C)

The use and meaning of the bits shall be as specified in Table 2-22. The contents of the register are valid and locked when the Capture Valid Info bit of the Port *n* Attributes Capture CSR is set (0b1). Unless otherwise specified, the bits in this register are read/write.

Table 2-22. Bit Settings for Port *n* Capture 0 CSRs

Bit	Name	Reset Value	Description
0-31	Capture 0	All 0s	If the info_type field of the Port n Attributes Capture CSR is "Control Symbol 24", "Control Symbol 48" or "Control Symbol 64", Delimited Control Symbol 24, Delimitted Control Symbol 48 bytes 0-3, or Control Symbol 64 bytes 0-3. If the info_type field of the Port *n* Attributes Capture CSR is "packet", packet bytes 0-3. Otherwise, implementation specific.

2.5.19 Port *n* Capture 1 CSR (Block Offset 0x50, 90,..., 410)

The use and meaning of the bits shall be as specified in Table 2-23. The contents of the register are valid and locked when the Capture Valid Info bit of the Port *n* Attributes Capture CSR is set (0b1). Unless otherwise specified, the bits in this register are read/write.

Table 2-23. Bit Settings for Port *n* Capture 1 CSRs

Bit	Name	Reset Value	Description
0-31	Capture 1	All 0s	If the info_type field of the Port n Attributes Capture CSR is "Control Symbol 48" or "Control Symbol 64", delimited control symbol Bytes 4-7. If the info_type field of the Port n Attributes Capture CSR is "packet", packet Bytes 4-7. Otherwise, implementation specific.

2.5.20 Port *n* Capture 2 CSR (Block Offset 0x54, 94,..., 414)

The use and meaning of the bits shall be as specified in Table 2-24. The contents of the register are valid and locked when the Capture Valid Info bit of the Port *n* Attributes Capture CSR is set (0b1). Unless otherwise specified, the bits in this register are read/write.

Table 2-24. Bit Settings for Port *n* Capture 2 CSRs

Bit	Name	Reset Value	Description
0-31	Capture 2	All 0s	If the info_type field of the Port n Attributes Capture CSR is "packet", packet Bytes 8-11. Otherwise, implementation specific.

2.5.21 Port *n* Capture 3 CSR (Block Offset 0x58, 98,..., 418)

The use and meaning of the bits shall be as specified in Table 2-25. The contents of the register are valid and locked when the Capture Valid Info bit of the Port *n* Attributes Capture CSR is set (0b1). Unless otherwise specified, the bits in this register are read/write.

Table 2-25. Bit Settings for Port *n* Capture 3 CSRs

Bit	Name	Reset Value	Description
0-31	Capture 3	All 0s	If the info_type field of the Port n Attributes Capture CSR is "packet", packet Bytes 12-15. Otherwise, implementation specific.

2.5.22 Port *n* Capture 4 CSR (Block Offset 0x5C, 9C,..., 41C)

The use and meaning of the bits shall be as specified in Table 2-26. This register shall be implemented for devices that have bit 19 (Dev32 Support) set in the Processing Element Features CAR. The contents of the register are valid and locked when the Capture Valid Info bit of the Port *n* Attributes Capture CSR is set. Unless otherwise specified, the bits in this register are read/write.

Table 2-26. Bit Settings for Port *n* Capture 4 CSRs

Bit	Name	Reset Value	Description
0-31	Capture 4	All 0s	If the info_type field of the Port n Attributes Capture CSR is "packet", packet Bytes 16-19. Otherwise, implementation specific.

2.5.23 Port *n* Error Rate CSR (Block Offset 0x68, A8,..., 428)

This register is used in conjunction with the Port *n* Error Rate Threshold register to monitor and control the reporting of Port *n* physical layer errors. The use and meaning of the bits shall be as specified in Table 2-27. Unless otherwise specified, the bits in this register are read/write.

Table 2-27. Bit Settings for Port *n* Error Rate CSRs

Bit	Name	Reset Value	Description
0-7	Error Rate Bias	0x80	This field specifies the rate at which the Error Rate Counter is decremented (the error rate bias value) 0x00 - do not decrement the error rate counter 0x01 - decrement every 1ms (+/-34%) 0x02 - decrement every 10ms (+/-34%) 0x04 - decrement every 100ms (+/-34%) 0x08 - decrement every 1s (+/-34%) 0x10 - decrement every 10s (+/-34%) 0x20 - decrement every 100s (+/-34%) 0x40 - decrement every 1000s (+/-34%) 0x80 - decrement every 10000s (+/-34%) other values are reserved
8-13	—		Reserved
14-15	Error Rate Recovery	0b00	The value of this field limits the incrementing of the Error Rate Counter above the failed threshold trigger. 0b00 - only count 2 errors above 0b01 - only count 4 errors above 0b10 - only count 16 error above 0b11 - do not limit incrementing the error rate count
16-23	Peak Error Rate	0x00	This field contains the peak value attained by the error rate counter since the field was last reset.
24-31	Error Rate Counter	0x00	This field contains a count of the number of physical layer errors that have been detected by the port, decremented by the Error Rate Bias mechanism, to create an indication of the physical layer error rate. Receipt of a reset-port request shall clear this field to 0x00. State machines associated with this field shall be reset to their power-up state.

2.5.24 Port *n* Error Rate Threshold CSR (Block Offset 0x6C, AC, ..., 42C)

This register controls the reporting of the link status to the system host. The use and meaning of the bits shall be as specified in Table 2-28. Unless otherwise specified, the bits in this register are read/write.

Table 2-28. Bit Settings for Port *n* Error Rate Threshold CSRs

Bit	Name	Reset Value	Description
0-7	Error Rate Failed Threshold Trigger	0xFF	This field contains the threshold value for reporting an error condition due to a possible broken link. 0x00 - Disable the Error Rate Failed Threshold Trigger 0x01 - Set the error reporting threshold to 1 0x02 - Set the error reporting threshold to 2 ... 0xFF - Set the error reporting threshold to 255
8-15	Error Rate Degraded Threshold Trigger	0xFF	This field contains the threshold value for reporting an error condition due to a degrading link. 0x00 - Disable the Error Rate Degraded Threshold Trigger 0x01 - Set the error reporting threshold to 1 0x02 - Set the error reporting threshold to 2 ... 0xFF - Set the error reporting threshold to 255
16-31	—		Reserved

2.5.25 Port *n* Link Uninit Discard Timer CSR (Block Offset 0x70, 0xB0, ..., 0x430)

The maximum value of the Link Uninit Timeout variable (0xFFFFFF) shall correspond to 6 to 12 seconds. The resolution of the Link Uninit Timeout variable shall be (maximum Link Uninit Timeout interval)/(2^{24}-1). The use and meaning of the bits shall be as specified in Table 2-29. Unless otherwise specified, the bits in this register shall be readable and writable.

Table 2-29. Bit Settings for Port n Link Uninit Discard Timer CSRs

Bit	Name	Reset Value	Description
0-23	Link Uninit Timeout	0x000000	On IDLE1 and IDLE2 links, this timer shall start counting when link_initialized is deasserted, and shall continue counting until link_initialized is asserted. On IDLE3 links this timer shall start when at least one of link_initialized, receive_enable, or transmit_enable is deasserted, and shall continue counting until link_initialized, receive_enable and transmit enable are all asserted. When this timer expires, all packets directed to this port from inside the device shall be discarded, and a "Link Uninit Packet Discard Active" event shall be detected. Packet discard shall occur until the "Link Uninit Packet Discard Active" status bit is cleared. The Link Uninit Discard Timer shall be disabled when Link Uninit Timeout is 0.
24-31	—		Reserved

2.5.26 Port *n* FIFO Error Detect CSR (Block Offset 0x7C, BC,..., 43C)

The Port *n* FIFO Error Detect Register shall be implemented when bit 2 of the Error Management/Hot Swap Extension Block CAR is set. The behavior of the register shall be as defined in Section 1.2.7 on page 14.

The use and meaning of the bits shall be as specified in Table 2-30. Unless otherwise specified, the bits in this register are read/write.

The two right-most columns in Table 2-30 indicate which bit fields must be implemented for Error Management Extensions and Hot Swap Extensions. An "X" in these columns means that the bit shall be implemented for that extension.

Table 2-30. Bit Settings for Port *n* FIFO Error Detect CSRs

Bit	Name	Reset Value	Description	Error Mgmt	Hot Swap
0	Implementation specific error	0b0	An implementation specific error has been detected	X	-
1	Link OK to Uninit Transition	0b0	The link has transitioned from a link initialized to link uninitialized state.	-	X

Table 2-30. Bit Settings for Port *n* FIFO Error Detect CSRs

Bit	Name	Reset Value	Description	Error Mgmt	Hot Swap
2	Link Uninit Packet Discard Active	0b0	The Link Uninit Discard Timer CSR period has expired.	-	X
3	Link Uninit to OK Transition	0b0	The link has transitioned from a link uninitialized to link initialized state.	-	X
4-7	—		Reserved		
8	Deprecated	0b0	Deprecated	X	-
9	Received corrupt control symbol	0b0	Received a control symbol with a bad CRC value or Received an incorrect sequence of control symbol codewords (for example, CSE or CSEB without a preceding CSB, or a CSB or CSEB that is not followed by a CSE or CSEB (IDLE3)	X	-
10	Received acknowledge control symbol with unexpected ackID	0b0	Received a packet-accepted or packet-retry control symbol with an unexpected ackID. "Error Recovery with ackID in PNA Enabled" is set in the Port n Latency Optimization CSR and a packet-not-accepted control symbol is received with an unexpected ackID.	X	-
11	Received packet-not-accepted control symbol	0b0	Received packet-not-accepted control symbol	X	-
12	Received packet with unexpected ackID	0b0	Received packet with unexpected ackID value - out-of-sequence ackID	X	-
13	Received packet with bad CRC	0b0	Received packet with a bad CRC value	X	-
14	Received packet exceeds maximum allowed size	0b0	Received packet that exceeds the maximum allowed size	X	-
15	Received illegal or invalid character	0b0	Received an 8b/10b code-group that is invalid (a code-group that does not have a 8b/10b decode given the current running disparity) or illegal (a code-group that is valid but whose use is not allowed by the LP-Serial protocol). This bit may be set in conjunction with bit 29, Delineation error. The implementation of this bit is optional, but strongly recommended. (IDLE1 and IDLE2) or Bit Interleaved Parity (BIP) check failed on at least one lane (IDLE3)	X	-
16	Received data character in IDLE1 sequence	0b0	Received a data character in an IDLE1 sequence. This bit may be set in conjunction with bit, 29 Delineation error. The implementation of this bit is optional, but strongly recommended.	X	-
17	Loss of descrambler synchronization	0b0	Loss of receiver descrambler synchronization while receiving scrambled control symbol and packet data. This bit shall be implemented only if port n supports descrambling of packet and control symbol data.	X	-
18	Invalid Ordered Sequence	0b0	Received an invalid ordered sequence on at least one lane (IDLE3)	X	-
19-25	—		Reserved		

Table 2-30. Bit Settings for Port *n* FIFO Error Detect CSRs

Bit	Name	Reset Value	Description	Error Mgmt	Hot Swap
26	Non-outstanding ackID	0b0	When there are outstanding ackIDs, a link_response was received with an ackID that is not outstanding	X	-
27	Protocol error	0b0	An unexpected control symbol was received	X	-
28	Deprecated	0b0	Deprecated	X	-
29	Delineation error	0b0	Received an 8b/10b code-group that is invalid (a code-group that does not have a 8b/10b decode given the current running disparity), illegal (a code-group that is valid, but whose use is not allowed by the LP-Serial protocol) or that is in a position in the received code-group stream that is not allowed by the LP-Serial protocol (IDLE1 and IDLE2) or Received a 64b/67b codeword whose type and !type bit values are the same (IDLE3)	X	-
30	Unsolicited acknowledgement control symbol	0b0	An unsolicited packet acknowledgement control symbol was received	X	-
31	Link timeout	0b0	A packet acknowledgement or link-response control symbol was not received within the specified timeout interval	X	-

Annex A Error Management Discussion (Informative)

A.1 Introduction

This section is intended to provide useful information/background on the application of the error management capabilities. This section is a guideline, not part of the specification.

A.2 Limitations of Error Management Discussion

The RapidIO hardware that implements the Error Management extensions is able to log physical layer errors and errors that occur at a higher level. Some error scenarios require no software intervention and recovery procedures are done totally by the hardware.

Some error scenarios detected require fault management software for recovery to be successful. For example, some types of logical layer errors on a Read or Write operation may be recoverable by killing the software process using the affected memory space and removing the memory space from the available system resource pool. It may also be possible for software to retry the operation, possibly through a different path in the switch fabric. Since such fault management software is typically tightly coupled to a particular system and/or implementation, it is considered outside of the scope of this specification.

Another area of fault recovery that requires fault management software to be implemented is correcting of system state after an error during an atomic operation. The swap style Atomic operations are possibly recoverable through software and require software convention to uniquely identify attempts to take locks. For example, if the request is lost and times out, software can examine the current lock value to determine if the request or the associated response was the transaction that was lost in the switch fabric. For all other Atomic operations (such as the Atomic set operation), it is impossible to correct the system state in the presence of a ‘lost packet’ type of error.

The use of RapidIO message packets relies on the use of higher layer protocols for error management. Since end points that communicate via messaging are typically running a variety of higher layer protocols, error reporting of both request and response timeouts is done locally by the message queue management controller.

Note that side effect errors can occur, for example, ERROR responses or RETRY responses during an active (partially completed) message, which may complicate the recovery procedure. The recovery strategies for messages lost in this manner are outside of the scope of this specification.

Globally Shared Memory systems that encounter a logical or transport layer error are typically not recoverable by any mechanism as this usually means that the processor caches are no longer coherent with the main memory system. Historically, recovery from such errors requires a complete reboot of the machine after the component that caused the error is repaired or replaced.

A.3 Hot-insertion/extraction Discussion

Hot-insertion can be regarded as an error condition in which a new part of the system is detected, therefore, hot-insertion of a Field Replaceable Unit (FRU) can be handled utilizing the above described mechanisms. This section describes two approaches for hot insertion. The first generally applies to high availability systems, or systems where FRUs need to be brought into the system in a controlled manner. The control mechanism is system specific. The second generally applies to systems where availability is less of a concern, for example, a trusted system or a system without a system host.

At system boot time, the system host identifies all of the unattached links in the machine through system discovery and puts them in a locked mode, whereby all incoming packets are to be rejected, leaving the drivers and receivers enabled. This is done by setting the Discovered bit in the Port General Control CSR and the Port Lockout bit in the Port *n* Control CSR. Note that whenever an FRU is removed, the port lockout bit should be used to ensure that whatever new FRU is inserted cannot access the system until the system host allows it. When a FRU is hot-inserted connecting to a switch device, the now connected link will automatically start the training sequence. When training is complete (the Port OK bit in the Port *n* Error and Status CSR is now set), the locked port generates a Maintenance port-write operation to notify the system host of the new connection, and sets the Port-write Pending bit.

On receipt of the port-write, the system host is responsible for bringing the inserted FRU into the system in a controlled manner. The system host can communicate with the inserted FRU using Maintenance operations after clearing all error conditions, if any, clearing the Port Lockout bit and clearing the Output and Input Port Enable bits in the Port *n* Control CSR. This procedure allows the system host to access the inserted FRU safely, without exposing itself to incorrect behavior by the inserted FRU.

In order to issue Maintenance operations to the inserted FRU, the system host must first make sure that the ackID values for both ends are consistent. Since the inserted FRU has just completed a power-up reset sequence, both it’s transmit and receive

ackID values are the reset value of 0x00. The system host can set the switch device's transmit and receive ackID values to also be 0x00 through the Port *n* Local ackID Status CSR if they are not already in that state, and can then issue packets normally.

The second method for hot insertion would allow the replaced FRU to bring itself into the system, which is necessary for a system in which the FRU is the system host itself. In this approach, the Port Lockout bit is not set and instead the Output and Input Port Enable bits are set for any unconnected port, allowing inserted FRUs free access to the system without reliance on a system host. Also, a port-write operation is not generated when the training sequence completes and the link is active, so a host is not notified of the event. However, this method leaves the system vulnerable to corruption from a misbehaving hot-inserted FRU.

As with the first case, the inserted FRU must make the ackID values for both link partners match in order to begin sending packets. In order to accomplish this, the inserted FRU may send a reset-port request to reinitialize the connected port. If the connected port does not support the reset-port request, the inserted FRU may recover the link by generating a link-request/link-status to the attached device to obtain it's expected receiver value using the Port *n* Link Maintenance Request and Response CSRs. It can then set its transmit ackID value to match. Next, the inserted FRU generates a Maintenance write operation to set the attached device's Port *n* Local ackID Status CSR to set the transmit ackID value to match the receive ackID value in the system host. Upon receipt of the maintenance write, the attached device sets it's transmit ackID value as instructed, and generates the maintenance response using the new value. Packet transmission can now proceed normally.

Hot extraction from a port's point of view behaves identically to a very rapidly failing link and therefore can utilize the above described error reporting mechanism. Hot extraction is ideally done in a controlled fashion by taking the FRU to be removed out of the system as a usable resource through the system management software so that extraction does not cause switch fabric congestion or result in a loss of data. Note that the Port n Link Uninit Discard Timer CSR can be used to prevent congestion in the case of an FRU removal which is not coordinated by the system host.

The required mechanical aspects of hot-insertion and hot-extraction are not addressed in this specification.

A.4 Port-write Discussion

The error management specification includes only one destination for port-write operations, while designers of reliable systems would assume that two is the minimum number. This section explains the rationale for only having one port-write destination.

It is assumed that in the event of an error on a link that both ends of the link will see the error. Thus, there are two parties who can be reporting on any error. In the case

that the sole link between an end point and a switch fails completely, the switch is expected to see and report the error. When one of a set of redundant links between an end point and a switch device fails, it is expected that the switch and possibly the end point will report the failure.

When a link between two switches fails, it is assumed that there are multiple paths to the controlling entity available for the port-write to travel. The switches will be able to send at least one, and possibly two, reports to the system host. It is assumed that it is possible to set up a switch's routing parameters such that the traffic to the system host will follow separate paths from each switch.

In some reliable systems, the system host is implemented as multiple redundant subsystems. It is assumed in RapidIO that only one subsystem is actually in control at any one time, and so should be the recipient of all port-writes. If the subsystem that should be in control is detected to be insane, it is the responsibility of the rest of the control subsystem to change the destination for port-writes to be the new subsystem that is in control.

A.5 Physical Layer Fatal Error Recovery Discussion

Recovery from a fatal error under software control at the physical layer may be possible under certain circumstances. An example of this would be if the transmitter and receiver have lost synchronization of their ackIDs. This could occur if one end of the link experienced a spurious reset. In this case a loss of packets may occur as there may be outstanding unacknowledged packets between the transmitter and the receiver.

Such an event would cause an error to be detected given the appropriate initial conditions at the transmitter, and, eventually a port-write to the system host to be generated if the system is properly configured:

- The reset state of the Input Port Enable bit in the Port *n* Control CSR set to disabled throughout the system.
- All defined errors in the Port *n* Error Detect CSRs are enabled and will increment the error rate counter throughout the system.

If a device experiences a reset event, numerous errors will be detected by the transmitter over time, and eventually an error threshold is reached as described in Section 1.2.2, "Error Reporting Thresholds", and the system host is notified as described in Section 1.4, "System Software Notification of Error". The most likely errors that will be detected are bits 12 (Received packet-not-accepted control symbol) and 26 (Non-outstanding ackID) in the Port *n* Error Detect CSRs, but others could be encountered depending upon the state of the link at the time of the reset event.

Re-synchronizing the ackIDs must be done from the transmitter side as it is not possible to communicate with the receiver with maintenance transactions in this

situation. Re-synchronizing the ackIDs can be done by sending a reset-port request to those devices which support it. ackID re-synchronization can also be done by resetting some of the physical layer state by writing the Port *n* Local ackID Status CSR with the appropriate ackID values. It may be necessary to have the transmitter drop outstanding packets using that CSR as well, depending upon the situation. It may not be desirable, or it might not be possible, to resend the packets, depending upon the state of the overall system and the transmitter implementation.

Therefore, the following sequence of events occur:

1. The system is configured as described above and is operating.
2. The receiver of a transmitter/receiver pair experiences a reset.
3. The transmitter enters error recovery mode and attempts to re-train the link.
4. Eventually the receiver comes back and link re-training completes.
5. The transmitter starts the error recovery sequence and begins to encounter large numbers of errors due to a bad ackID for a link-response (which may immediately cause a port-write transaction to be sent to the system host to report the condition) or having all packets receive packet-not-accepted control symbols. As noted earlier, other errors may also be detected.
6. At some point, an error threshold is reached and the system host is sent a port-write maintenance transaction to report the condition, if one has not already been sent.
7. The system host cleans up the machine using reset-port requests and/or maintenance transactions, including resetting ackIDs in the transmitter and rediscovering and reconfiguring the lost portion of the machine. This may be a very complex and time-consuming task.

Note that it may be useful to implement resetting the ackIDs and restarting the link in hardware for lab debug or for applications where frequent resets are expected and software intervention is not required. This could leverage the standard port-reset request function.

Blank page

Glossary of Terms and Abbreviations

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

D

Degraded threshold. Bits 8-15 of the Port n Error Rate Threshold CSR. An application-specific level that indicates an unacceptable error rate resulting in degraded throughput, when equal to the error rate count.

F

Failed threshold. Bits 0-7 of the Port n Error Rate Threshold CSR. An application-specific level that indicates an error rate due to a broken link, when equal to the error rate count.

H

Hot-insertion. Hot-insertion is the insertion of a processing element into a powered-up system.

Hot-extraction. Hot-extraction is the removal of a processing element from a powered-up system.

L

Logical/Transport error. A logical/transport error is one that cannot be resolved using the defined transmission error recovery sequence, results in permanent loss of data or causes system corruption. Recovery may possible under software control.

N

Non-reporting processing element. A non-reporting processing element depends upon an attached device (usually a switch) to report its logged errors to the system host on its behalf.

O

Operation. A set of transactions between end point devices in a RapidIO system (requests and associated responses) such as a read or a write.

Ownership. A processing element has the only valid copy of a coherence granule and is responsible for returning it to home memory.

P

Physical error. A physical error occurs only in the physical layer.

Port healing. The process whereby software resets the error rate count, or allows it to decrement as required by the error rate bias field of the Port n Error Rate CSR.

R

Read operation. An operation used to obtain a globally shared copy of a coherence granule.

Reporting processing element. A reporting processing element is capable of reporting its logged errors to the system host.

S

Switch processing element. One of three processing elements, a switch processing element, or switch, is capable of logging and reporting errors to the host system.

T

Transmission error. A transmission error is one that can be resolved using the defined transmission error recovery sequence, results in no permanent loss of data and does not cause system corruption. Recovery may also be possible under software control using mechanisms outside of the scope of this specification.