RapidIO™ Specification, Revision 3.0
RapidIO™ Specification, Revision 3.1

Errata Revision 2.1

Released 2/2016

RapidIO.org Confidential

Revision History

Revision 1.0	Description	Date
1.0	First Release	01/20/2014
2.0	Correct Errata 3 equation Correct Errata 5 to refer to transmit_enable instead of transit_enable_tw Add Errata Overview Section Add Errata 11, “Corrections to Typical Data Flow Diagrams	02/19/2015
2.1	Add Errata 12, “Transmit Emphasis Timeout Control Clarification Add Errata 13, “Dev32 Routing Table Examples Correction Add Errata 14, “Seed Control Word Bit Ordering Clarification Add Errata 15, “Descrambler Seed Ordered Sequence Spacing Clarification	2/1/2016

Questions regarding the RapidIO Trade Association, specifications, or membership should be forwarded to:
RapidIO Trade Association
Suite 325, 3925 W. Braker Lane
Austin, TX 78759
512-305-0070 Tel.
512-305-0009 FAX.

RapidIO Specification Revisions 3.0 and 3.1 Errata Revision 2.1

Publicly available RapidIO specifications are identified with a major and minor revision number of the form X.Y, where 'X' is the major revision number and 'Y' is the minor revision number. For example, specification revision 2.1 indicates that it is a minor revision of the 2.0 specification. Minor revisions may include new functionality, and may also include errata resolutions from previous major and/or minor revisions.

Privately available RapidIO specifications are identified with a major, minor, and tertiary revision number of the form X.Y.Z, where 'Z' is the tertiary revision number. The tertiary revision number is the revision number of the errata document for a publicly available specification. The errata documents existence is publicly visible. The errata documents contents are only available to RapidIO.org members. All versions of the errata documents are available to members.

This is Revision 2 of the errata for the RapidIO Specification Revision 3.0 and Revision 3.1. RapidIO specification revisions 3.0.2 and 3.1.2 are formed by applying the errata in this document to these specifications, as specified in the table below.

Errata	Revision 3.0	Revision 3.1
1, “Change “lost_valid_cs_reception”	Yes	
2, “Correct “Lane Alignment State Machine”	Yes	
3, “Retraining Equations for 1x/2x Ports	Yes	
4, “Receive Width Value Sent During Recovery States	Yes	
5, “Transmit Width Management Clarification	Yes	
6, “Maximum Packet Size Clarification	Yes	
7, “Link Request Control Symbol Transmission	Yes	
8, “Standard Route Cfg Port Select CSR Implementation Specific Bits	Yes	
9, “Port n Link Uninit Discard Timer CSR Change	Yes	Yes
10, “Change to Retrain/Xmt_Width_Control State Machine	Yes	Yes
11, “Corrections to Typical Data Flow Diagrams	Yes	Yes
12, “Transmit Emphasis Timeout Control Clarification	Yes	Yes
13, “Dev32 Routing Table Examples Correction	Yes	Yes
14, “Seed Control Word Bit Ordering Clarification	Yes	Yes
15, “Descrambler Seed Ordered Sequence Spacing Clarification	Yes	Yes

1 Change “lost_valid_cs_reception”

Make the following change to RapidIO Specification Revision 3.0, Part 6, *Section 5.18.1.3 State Machine Variables:*

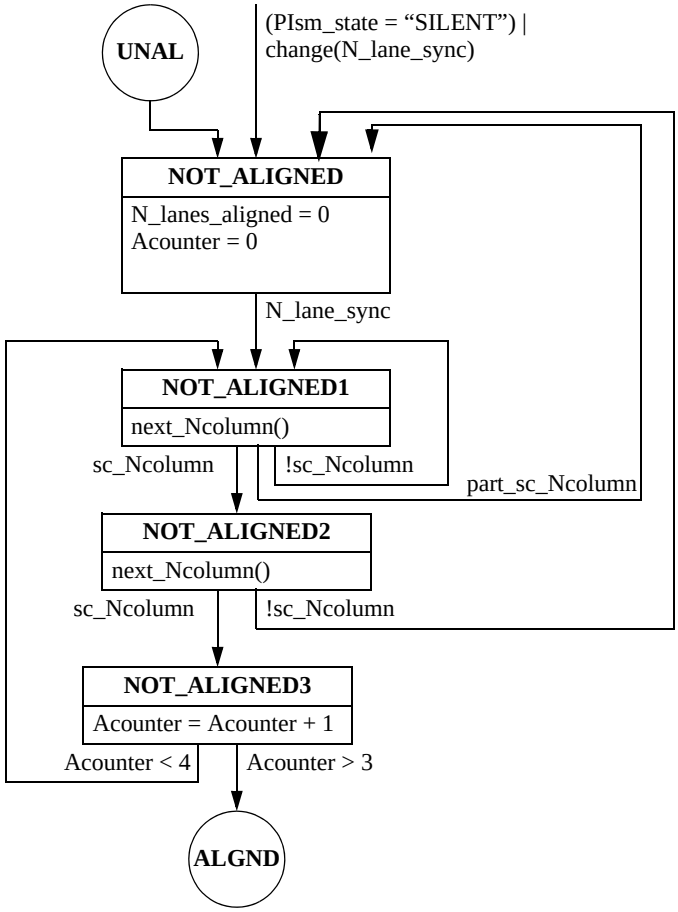
1. Change the definition of “lost_valid_cs_reception” to the following:

   “Asserted when receive_enable has been continuously asserted for the last 2048 columns and no valid control symbol or Status/Control Ordered Sequence has been received during that time.”

2 Correct “Lane Alignment State Machine”

Make the following change to the RapidIO Specification Revision 3.0 Part 6, Section 5.18.6, *Figure 5-29, Part 1*:

1. The 10xN lane alignment state machine must enter the ALIGNED state only after four consecutive SC ordered sequences have been received. The existing state machine must have a transition labelled “part_sc_Ncolumn” added from the NOT_ALIGNED1 state to the NOT_ALIGNED state. Change the 10xN state machine diagram as follows:



3 Retraining Equations for 1x/2x Ports

Make the following change to the RapidIO Specification Revision 3.0, Part 6, *Section 5.18.7.1*:

1. All 10xN state machine equations must operate as intended when a port capable of operating as a 4x, 2x, or 1x is configured to operate as a 1x port, a 2x port, or a 1x/4x port. As a result, change the 10xN 1xRtoSL equation to the following:

   1xRtoSL =
   !lane_sync[0] &
   !(lane_sync[1] & 2x_mode_supported) &
   !(lane_sync[2] & 4x_mode_supported) | recovery_tmr_done

4 Receive Width Value Sent During Recovery States

Make the following changes to the RapidIO Specification Revision 3.0, Part 6:

1. Change the definition of bits 17-19 of the status_control codeword found in *table 5-3* Status_control field content to the following:

   Receive width

   The width at which the port is currently receiving control symbols and packets (destriping width)

   0b000 - None
   0b001 - 1x mode, lane 0
   0b010 - 2x mode
   0b011 - 4x mode
   0b100 - 8x mode
   0b101 - 16x mode
   0b110 - 1x mode, lane 1
   0b111 - 1x mode, lane 2

   The receive width field shall retain the value it held prior to the Port Initialization State Machine entering the 1x_RECOVERY, 2x_RECOVERY or Nx_RECOVERY states for the duration of those recovery states.

2. Change the definition of the rcv_width state machine variable in *section 5.18.1.3* to the following:

   rcv_width (receive width)

   A three bit field indicating the width mode at which the port is currently receiving control symbols and packets from the link. Also the source of the value placed in the “Receive width” field of Status/Control control codewords transmitted by the port.

   The current receive width shall be encoded as follows:

   0b000 - None; the port has not completed initialization
   0b001 - 1x mode
   0b010 - 2x mode
   0b011 - 4x mode
   0b100 - 8x mode
   0b101 - 16x mode
   0b110 - 1x mode, lane 1
   0b111 - 1x mode, lane 2

   The rcv_width variable shall retain the value it held prior to the Port Initialization State Machine entering the 1x_RECOVERY, 2x_RECOVERY or Nx_RECOVERY states for the duration of those recovery states."

5 Transmit Width Management Clarification

Make the following changes to the RapidIO Specification Revision 3.0, Part 6:

1. Change the definition of transmit_enable_tw in *Section 5.18.1.3* to the following:

   A local transmit enable control used in the Transmit_Width state machine. Control symbols and packets may be transmitted when transmit_enable_tw is asserted. When transmit_enable_tw is deasserted, the port shall complete transmission of packets or control symbols in progress and then stop transmitting further packets or control symbols until transmit_enable_tw is asserted. Completion of in progress packet and control symbol transmission shall be signaled by the assertion of the xmting_idle variable.

2. Change the definition of xmting_idle in *Section 5.18.1.3* to the following:

   Asserted when the port has stopped transmitting control symbols and packets in response to the deassertion of transmit_enable. The port shall transmit only the IDLE3 sequence when xmting_idle is asserted. Xmting_idle is deasserted when transmit_enable is asserted.

3. Change the second paragraph of *Section 5.18.9.2* to the following:

   When an executable transmit width port command is received by a port, the Transmit_Width state machine attempts to switch to the requested transmit width. The state machine begins by starting the transmit width timer (xmt_width_tmr_en asserted) and output enabling the drivers for the lanes needed for requested width. When transitioning from a narrower to a wider width, 64b/67b compliant data sufficient to allow the link partner to achieve frame lock and lane alignment shall be sent on the newly enabled lanes. Some examples of 64b/67b compliant data are the IDLE3 sequence, and the data pattern sent on Lane 0 of the port.

4. Change the fourth paragraph of *Section 5.18.9.2* to the following:

   When the lanes needed for the new width become ready, the state machine halts the transmission of control symbols and packets by deasserting transmit_enable_tw and waiting for control symbol and packet transmission to end as indicated by xmting_idle.

5. Change the sixth paragraph of *Section 5.18.9.2* to the following:

   If the connected port ACKs the receive width link command and the receive width it reports in the "Receive width" of Status/Control control codewords that it transmits matches the requested receive width, the state machine switches to the new transmit width, changes the receive width link command to "hold", ACKs the transmit width port command, output disables any lanes not needed for the new transmit width, and re-enables control symbol and packet transmission (transmit_enable_tw asserted).

6 Maximum Packet Size Clarification

Make the following changes to the RapidIO Specification Revision 3.0, Part 7, *Section 3.2.1.1*:

1. Replace the ninth bullet with the following:
   - Support for maximum size packets for switch devices:
     — Switch devices which support Dev8 and/or Dev16 device IDs shall support a maximum packet size of 276 bytes. See *RapidIO Part 6: 1x/4x LP-Serial Physical Layer Specification*, Section 2.5
     — Switch devices which support Dev32 device IDs shall support a maximum packet size of 280 bytes. See *RapidIO Part 6: 1x/4x LP-Serial Physical Layer Specification*, Section 2.5

7 Link Request Control Symbol Transmission

Link Request control symbols must always begin in lane 0 of a multilane port. Replace the last paragraph of RapidIO Specification Revision 3.0, Part 6, Section 5.7.2 Packet Termination Delimiters, with:

If a packet is cancelled with a link-request control symbol, a Seed ordered sequence shall be transmitted between the end of the packet and the link-request control symbol on all lanes. Bytes of 0x00 shall be used to pad the space between the end of the packet and the beginning of the Seed ordered sequence. The link-request control symbol shall immediately follow the Seed ordered sequence. The link-request control symbol shall begin transmission in Lane 0 of a multi-lane port. Since the link-request control symbol also functions as the “restart-from-error” control symbol, the transmission of the Seed ordered sequence is needed to allow the receiver's descrambler(s) to recover synchronization with the input data stream(s) in the case the receiving port has lost descrambler sync.

8 Standard Route Cfg Port Select CSR Implementation Specific Bits

The most significant 4 bits of the Standard Route Cfg Select CSR are reserved for implementation specific functionality to be consistent with the Standard Route Default Port CSR. In RapidIO Specification Revision 3.0 Part 3, Section 3.5.6, change the definition of bits 0-7 in Table 3-10. “Bit Settings for Standard Route Configuration Port Select CSR” to the following:

Bits	Name	Reset Value	Description
0-3	Cop3_msb_or_imp_spec	0x00	Cop3_msb_or_imp_spec: This field shall be reserved if extended route table mechanism is not enabled and bit 19 of the Processing Element Features CAR is clear. When bit 19 of the Processing Element Features CAR is clear, and the extended route table mechanism is enabled, this field contains the most significant 4 bits of the configuration output port3 value. When bit 19 of the Processing Element Features CAR is set, this field optionally controls implementation-specific routing functionality: • Bits in this field that do not control implementation specific routing functionality shall be read only, with a fixed value of 0. • Implementation-specific routing functionality may be active if any bit in this field is set. • Implementation-specific routing functionality shall not be active if all bits in this field are clear.
4-7	Config_output_port3_lsb	0x00	Configuration output port3 - This field shall be reserved if the extended route table mechanism is not enabled. If the extended route table mechanism is enabled, this field contains the least significant 4 bits of the config output port3 value.

9 Port *n* Link Uninit Discard Timer CSR Change

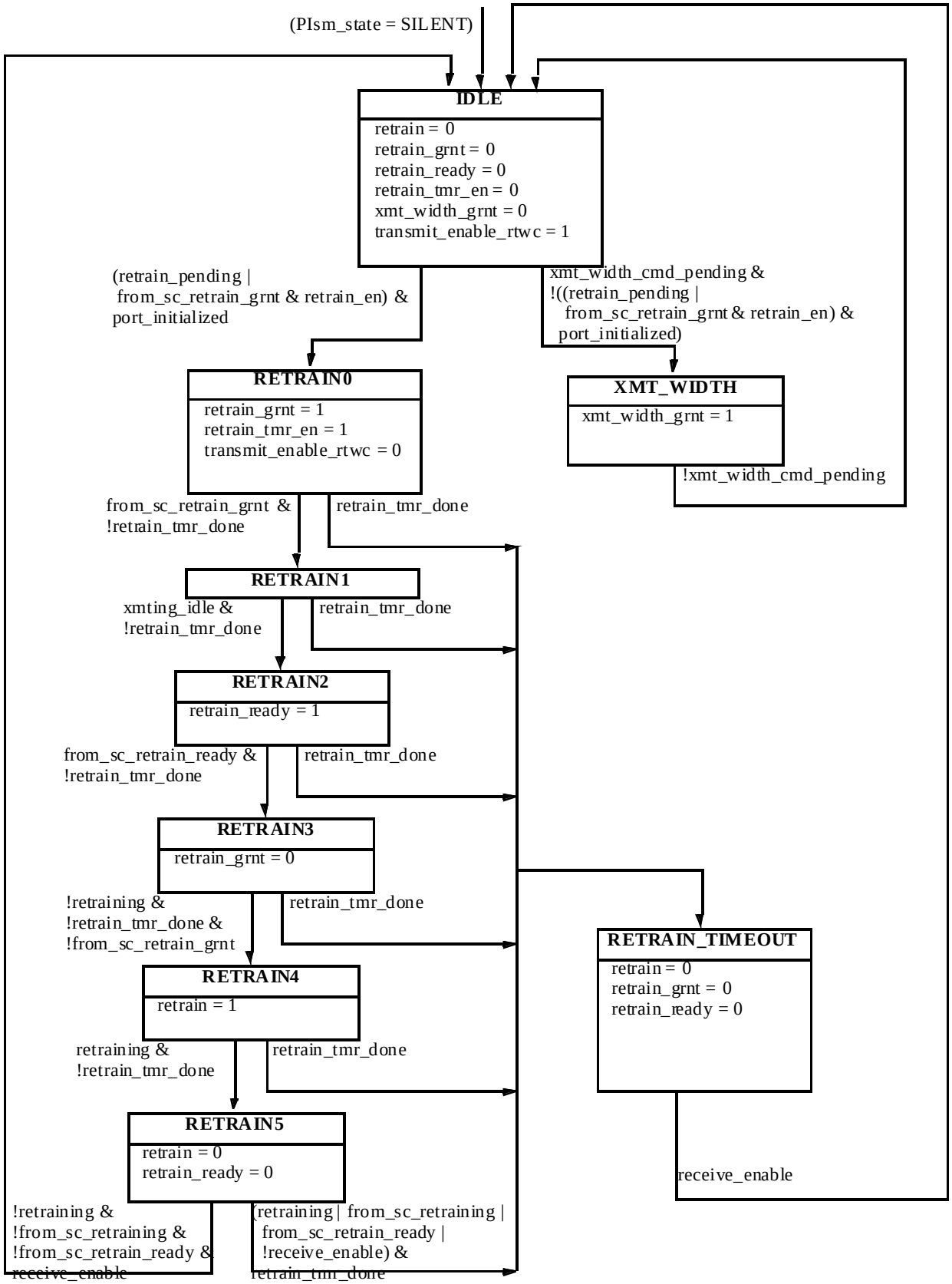
The Link Uninit Discard timer should expire when it has not been possible to exchange packets on the link for the programmed period. In RapidIO Specification Revision 3.0 and 3.1, Part 8, section 2.5.25 Port n Link Uninit Discard Timer CSR, the description of the Link Uninit Timeout Field is changed to refer to the link_initialized variable. This fixes a defect found in IDLE1 and IDLE2.

Bits	Name	Reset Value	Description
0-23	Link Uninit Timeout	0x00	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.

10 Change to Retrain/Xmt_Width_Control State Machine

One of the objectives of the RapidIO standard is to ensure deterministic operation. This includes the operation of state machines dependent on values found in the Status/Control Ordered Sequence(SC OS). However, the definition of the Retrain/Xmt_Width_Control FSM is not deterministic when retraining is preemptive. The operation of this state machine is dependent on the exchange SC OS values but SC OS cannot be transmitted when near continuous traffic is being transmitted on a link. A “near continuous” traffic pattern that includes IDLE3 sequences which are always 18 codewords would never send an SC OS, based on the example implementation shown in RapidIO Specification Revision 3.0 Part 6, 5.10.2 Idle Sequence 3 Generation, Figure 5-18 “Example of a Pseudo-Random Idle Codeword Generator".

To ensure SC OS will be transmitted deterministically to operate the Retrain/Xmt_Width_Control FSM, in RapidIO Specification Revision 3.0 and 3.1, Part 6, Figure 5-34 Retrain/Xmt_Width_Control state machine is changed to the following:

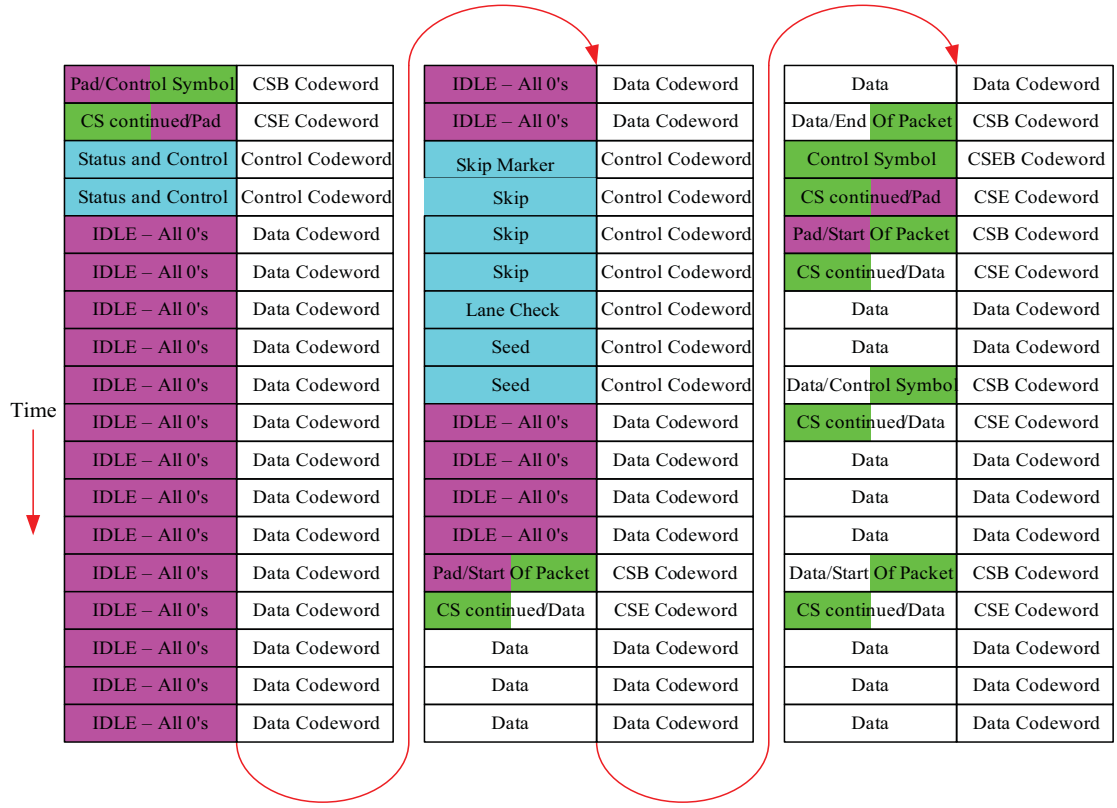

(PIsm_state = SILENT)
IDLE
retrain = 0
retrain_grnt = 0
retrain_ready = 0
retrain_tmr_en = 0
xmt_width_grnt = 0
transmit_enable_rtwc = 1
(retrain_pending |
from_sc_retrain_grnt & retrain_en) &
port_initialized
xmt_width_cmd_pending &
!((retrain_pending |
from_sc_retrain_grnt & retrain_en) &
port_initialized)
RETRAIN0
retrain_grnt = 1
retrain_tmr_en = 1
transmit_enable_rtwc = 0
XMT_WIDTH
xmt_width_grnt = 1
!xmt_width_cmd_pending
from_sc_retrain_grnt &
!retrain_tmr_done
retrain_tmr_done
RETRAIN1
xmting_idle &
!retrain_tmr_done
retrain_tmr_done
RETRAIN2
retrain_ready = 1
from_sc_retrain_ready &
!retrain_tmr_done
retrain_tmr_done
RETRAIN3
retrain_grnt = 0
!retraining &
!retrain_tmr_done &
!from_sc_retrain_grnt
retrain_tmr_done
RETRAIN_TIMEOUT
retrain = 0
retrain_grnt = 0
retrain_ready = 0
RETRAIN4
retrain = 1
retraining &
!retrain_tmr_done
retrain_tmr_done
RETRAIN5
retrain = 0
retrain_ready = 0
receive_enable
!retraining &
!from_sc_retraining &
!from_sc_retrain_ready &
receive_enable
(retraining | from_sc_retraining |
from_sc_retrain_ready |
!receive_enable) &
retrain_tmr_done

11 Corrections to Typical Data Flow Diagrams

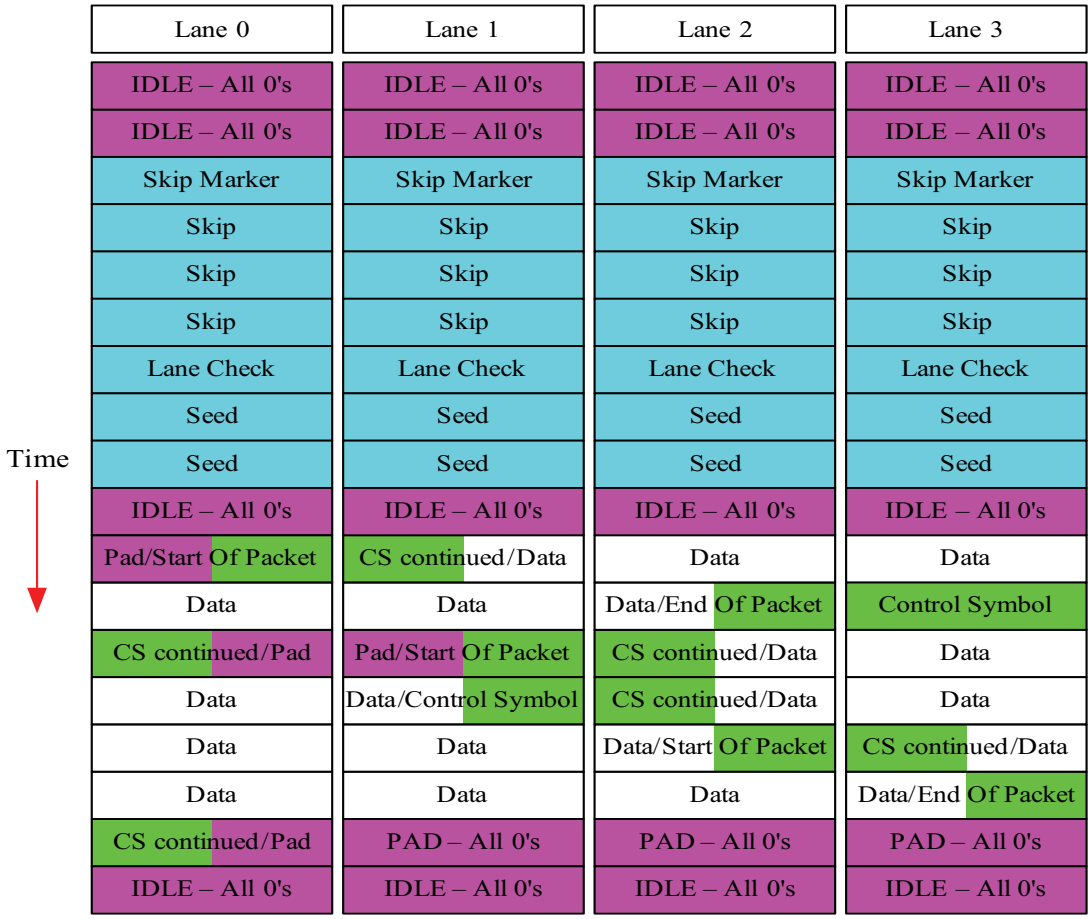
The definition of a skip ordered sequence is found in RapidIO Specification Revision 3.0 and 3.1, Part 6, section 5.9.3 Skip Ordered Sequence. It consists of the following sequence of control codewords:

- skip marker
- a number of skips
- lane check
- seed ordered sequence.

RapidIO Specification Revision 3.0 and 3.1, Part 6, figure 5-19. 1x Typical Data Flow with Control Symbol 64 is changed to be consistent with this definition:



RapidIO Specification Revision 3.0 and 3.1, Part 6, figure 5-20. Typical 4x Data Flow with Control Symbol 64 is also changed to be consistent with this definition:



12 Transmit Emphasis Timeout Control Clarification

In Part 6, the 7.6.16 Port n Link Timers Control CSRs Emphasis Command Timeout field was added to control the timeout for transmit emphasis commands found in IDLE2, IDLE3 and DME training. Some portions of the RapidIO Revision 3.0 and 3.1 specifications were not updated to reflect the presence of this register field. To that end, change the text found in Part 6 Section 4.7.4.1.4, description of handshake to:

"If, for any reason, the connected port fails to assert ACK or NACK the assertion of CMD within the timeout period configured in 7.6.16 Port n Link Timers Control CSRs Emphasis Command Timeout field, CMD shall be deasserted. Once deasserted, CMD shall remain deasserted for at least the timeout period configured in the Emphasis Command Timeout field before being reasserted."

Similarly, change the text in Part 6section 5.11.2.2, description of response timeout for IDLE3 short reach, point 4 to:

"4. Once a "Transmit equalizer command" is asserted, it shall remain asserted and unchanged in value until the value of "Transmit equalizer status" is different from "not_updated" or the command has been asserted for the timeout period configured in 7.6.16 Port n Link Timers Control CSRs Emphasis Command Timeout field. At that point, the command shall be de-asserted within 5 usec of whichever of the two events occurred first. If the command timed out, the command shall be deasserted for the timeout period configured in the Emphasis Command Timeout field."

Add the following text to the end of 5.11.2.1 Long run 10.3125 Gbaud training:
"Implementations shall implement a timeout on DME transmit emphasis requests. The timeout shall be controlled by 7.6.16 Port n Link Timers Control CSRs Emphasis Command Timeout field."

Change the definition of the Port n Link Timers Control CSR Emphasis Command Timeout field to:

Table 12-1. Bit Settings for Port *n* Link Timers Control CSRs

Bit	Name	Reset Value	Description
24-31	Emphasis Command Timeout	See Description	Controls the length of time allowed for transmit emphasis command to be acknowledged during DME training, CW training, CS Field training, and retraining. The Maximum Period for this timeout shall be 256 microseconds +/- 34%. The programmed period for this timeout is computed by: (Emphasis Command Timeout) * (Maximum Period/256). A value of 0 shall disable this timer. The reset value of this timer is the implementation specific value which results in a Emphasis Command timeout period that is: - at least 100 microseconds and - is as close to 100 microseconds as possible A value of 0 shall disable this timer. NOTE: The Maximum Period of this timeout is specified loosely (+/- 34%) to allow implementation flexibility and innovation. The reset value of the timeout is specified more tightly (+ 0 to 1/256%) to ensure consistent, interoperable behavior during link initialization. This register field is reserved when the port is operating with IDLE1.

13 Dev32 Routing Table Examples Correction

Change Part 3, Annex A to the following:

0.1 “Dev32 Configuration Examples

This chapter provides several examples of how to use the Dev32 routing table programming interface. The given examples build upon each other while proceeding through the sections. References to the order of operations within the examples run from the top of a list to the bottom unless otherwise stated.

Initially assume a switch with 16 ports which supports Dev32 device IDs. Assume that the switch must support the following routing hierarchy, where “**” means “All Values”:

- Device ID 0x00_11_20_** must be routed to port 14.
- Device IDs 0x00_11_0X_** must be routed to port X, where X is 0 to 13.
- Device IDs 0x00_ZZ_**_** must be routed to port 15 when ZZ is 0 to 0x10.
- All other packets must be dropped.

Further assume that Port 7 must be programmed to support the above hierarchy, and has initial register values as follows:

Table 0-1. Example Port 7 Routing Table Register Block Registers

Register Name	Register Address	Register Value
Switch Routing Table Register Block Header	0x8000	N/A
Port 7 Routing Table Control CSR	0x8120	0x8000_0000
Port 7 Level 0 Info CSR	0x8130	0x0107_0000
Port 7 Level 1 Info CSR	0x8134	0x0307_0400
Port 7 Level 2 Info CSR	0x8138	0x0407_1000

0.1.1 Example 1: Routing 0x00_11_20_** to Port 14

To route the Dev32 destination IDs 0x00_11_20_** to Port 14, make use of routing table group 0 for Level 0, and routing table group 1 for Level 1 and Level 2. Specific entries for each level must be programmed.

Table 0-2. Example 1 Accesses

Register Name	Register Address	Register Value	Description
Port 7 Level 0 Group 0 Entry 0 Routing Table Entry CSR	0x0007_0000	0x0000_0201	Map Level 0 Group 0 index 0x00 to Level 1 Group 1
Port 7 Level 1 Group 1 Entry 0x11 Routing Table Entry CSR	0x0007_0844	0x0000_0201	Map Level 1 Group 1 index 0x11 to Level 2 Group 1
Port 7 Level 2 Group 1 Entry 0x20 Routing Table Entry CSR	0x0007_1480	0x0000_000E	Map Level 2 Group 1 index 0x20 to Port 14.

0.1.2 Example 2: Routing 0x00_11_0X_** to Port X

This example builds upon the configuration put in place by Example 1. Routing configuration is therefore complete for Level 0 and Level 1, so what remains is to complete the Level 2 programming.

Table 0-3. Example 2 Accesses

Register Name	Register Address	Register Value	Description
Port 7 Level 2 Group 1 Entry 0x00 Routing Table Entry CSR	0x0007_1400	0x0000_0000	Map Level 2 Group 1 index 0x00 to Port 0.
Port 7 Level 2 Group 1 Entry 0x01 Routing Table Entry CSR	0x0007_1404	0x0000_0001	Map Level 2 Group 1 index 0x01 to Port 1.
Port 7 Level 2 Group 1 Entry 0x02 Routing Table Entry CSR	0x0007_1408	0x0000_0002	Map Level 2 Group 1 index 0x02 to Port 2.
Port 7 Level 2 Group 1 Entry 0x03 Routing Table Entry CSR	0x0007_140C	0x0000_0003	Map Level 2 Group 1 index 0x03 to Port 3.
Port 7 Level 2 Group 1 Entry 0x04 Routing Table Entry CSR	0x0007_1410	0x0000_0004	Map Level 2 Group 1 index 0x04 to Port 4.
Port 7 Level 2 Group 1 Entry 0x05 Routing Table Entry CSR	0x0007_1414	0x0000_0005	Map Level 2 Group 1 index 0x05 to Port 5.
Port 7 Level 2 Group 1 Entry 0x06 Routing Table Entry CSR	0x0007_1418	0x0000_0006	Map Level 2 Group 1 index 0x06 to Port 6.
Port 7 Level 2 Group 1 Entry 0x07 Routing Table Entry CSR	0x0007_141C	0x0000_0007	Map Level 2 Group 1 index 0x07 to Port 7.
Port 7 Level 2 Group 1 Entry 0x08 Routing Table Entry CSR	0x0007_1420	0x0000_0008	Map Level 2 Group 1 index 0x08 to Port 8.
Port 7 Level 2 Group 1 Entry 0x09 Routing Table Entry CSR	0x0007_1424	0x0000_0009	Map Level 2 Group 1 index 0x09 to Port 9.
Port 7 Level 2 Group 1 Entry 0x0A Routing Table Entry CSR	0x0007_1428	0x0000_000A	Map Level 2 Group 1 index 0x0A to Port A.
Port 7 Level 2 Group 1 Entry 0x0B Routing Table Entry CSR	0x0007_142C	0x0000_000B	Map Level 2 Group 1 index 0x0B to Port B.
Port 7 Level 2 Group 1 Entry 0x0C Routing Table Entry CSR	0x0007_1430	0x0000_000C	Map Level 2 Group 1 index 0x0C to Port C.
Port 7 Level 2 Group 1 Entry 0x0D Routing Table Entry CSR	0x0007_1434	0x0000_000D	Map Level 2 Group 1 index 0x0D to Port D.

0.1.3 Example 3: Routing 0x00_ZZ_**_** to Port 15, ZZ=[0,0x10]

This example builds upon the configuration put in place by Example 1. Routing configuration is

complete for Level 0, so what remains is to program the Level 1 registers.

Table 0-4. Example 3 Accesses

Register Name	Register Address	Register Value	Description
Port 7 Level 1 Group 1 Entry 0x00 Routing Table Entry CSR	0x0007_0800	0x0000_000F	Map Level 1 Group 1 index 0x00 to Port 15.
Port 7 Level 1 Group 1 Entry 0x01 Routing Table Entry CSR	0x0007_0804	0x0000_000F	Map Level 1 Group 1 index 0x01 to Port 15.
Port 7 Level 1 Group 1 Entry 0x02 Routing Table Entry CSR	0x0007_0808	0x0000_000F	Map Level 1 Group 1 index 0x02 to Port 15.
Port 7 Level 1 Group 1 Entry 0x03 Routing Table Entry CSR	0x0007_080C	0x0000_000F	Map Level 1 Group 1 index 0x03 to Port 15.
Port 7 Level 1 Group 1 Entry 0x04 Routing Table Entry CSR	0x0007_0810	0x0000_000F	Map Level 1 Group 1 index 0x04 to Port 15.
Port 7 Level 1 Group 1 Entry 0x05 Routing Table Entry CSR	0x0007_0814	0x0000_000F	Map Level 1 Group 1 index 0x05 to Port 15.
Port 7 Level 1 Group 1 Entry 0x06 Routing Table Entry CSR	0x0007_0818	0x0000_000F	Map Level 1 Group 1 index 0x06 to Port 15.
Port 7 Level 1 Group 1 Entry 0x07 Routing Table Entry CSR	0x0007_081C	0x0000_000F	Map Level 1 Group 1 index 0x07 to Port 15.
Port 7 Level 1 Group 1 Entry 0x08 Routing Table Entry CSR	0x0007_0820	0x0000_000F	Map Level 1 Group 1 index 0x08 to Port 15.
Port 7 Level 1 Group 1 Entry 0x09 Routing Table Entry CSR	0x0007_0824	0x0000_000F	Map Level 1 Group 1 index 0x09 to Port 15.
Port 7 Level 1 Group 1 Entry 0x0A Routing Table Entry CSR	0x0007_0828	0x0000_000F	Map Level 1 Group 1 index 0x0A to Port 15.
Port 7 Level 1 Group 1 Entry 0x0B Routing Table Entry CSR	0x0007_082C	0x0000_000F	Map Level 1 Group 1 index 0x0B to Port 15.
Port 7 Level 1 Group 1 Entry 0x0C Routing Table Entry CSR	0x0007_0830	0x0000_000F	Map Level 1 Group 1 index 0x0C to Port 15.
Port 7 Level 1 Group 1 Entry 0x0D Routing Table Entry CSR	0x0007_0834	0x0000_000F	Map Level 1 Group 1 index 0x0D to Port 15.
Port 7 Level 1 Group 1 Entry 0x0E Routing Table Entry CSR	0x0007_0838	0x0000_000F	Map Level 1 Group 1 index 0x0E to Port 15.
Port 7 Level 1 Group 1 Entry 0x0F Routing Table Entry CSR	0x0007_083C	0x0000_000F	Map Level 1 Group 1 index 0x0F to Port 15.
Port 7 Level 1 Group 1 Entry 0x10 Routing Table Entry CSR	0x0007_0840	0x0000_000F	Map Level 1 Group 1 index 0x10 to Port 15.

0.1.4 Example 4: All Other Packets Must Be Dropped

This example builds upon the configuration put in place by Example 3. Routing for Dev32 device IDs has been configured. Dev16 and Dev8 deviceIDs are routed using Group 0 of Level 1 and Level 2, respectively. The default values for all entries is to drop packets. Nothing more needs to be programmed to drop all Dev16 and Dev8 deviceIDs.

0.1.5 Example 5: Flat Routing Table Operation

This example illustrates the "flat" programming model, in which device IDs are supported sequentially by the routing tables. Dev16 device IDs of the form 0x00** are treated as Dev8 device IDs.

Initially assume a switch with 16 ports which supports Dev32 device IDs. Assume that the switch must support the following routing hierarchy, where "**" means "All Values":

- Device ID 0x01_20 must be routed to port 14.
- Device IDs 0x00_0X must be routed to port X, where X is 0 to 13.
- Device IDs 0x02_00 and 0x03_00 must be routed to port 15.
- All other packets must be dropped.

Further assume that Port 7 must be programmed to support the above hierarchy, and has initial register values as follows:

Table 0-5. Example 5 Port 7 Routing Table Register Block Registers

Register Name	Register Address	Register Value
Switch Routing Table Register Block Header	0x8000	N/A
Port 7 Routing Table Control CSR	0x8120	0x8000_0000
Port 7 Level 0 Info CSR	0x8130	0x0107_0000
Port 7 Level 1 Info CSR	0x8134	0x0307_0400
Port 7 Level 2 Info CSR	0x8138	0x0407_1000

The following register accesses must be performed:

Table 0-6. Example 5 Accesses

Register Name	Register Address	Register Value	Description
Port 7 Routing Table Control CSR	0x0000_8120	0x0000_0000	Change to Flat Routing Table Model
Port 7 Level 0 Info CSR	0x0000_8130	0x0407_0000	Read Level 0 Info to determine how many DeviceIDs are supported. Four groups are supported, or destIDs 0x0000 through 0x03FF.
Port 7 Level 1 Info CSR	0x0000_8134	0x0000_0000	Read Level 1 Info, confirm register is reserved
Port 7 Level 2 Info CSR	0x0000_8138	0x0000_0000	Read Level 2 Info, confirm register is reserved
Port 7 Level 0 Group 1 Entry 0x20	0x0007_0480	0x0000_000E	Route DestID 0x0120 to port 14.
Port 7 Level 0 Group 0 Entry 0x00	0x0007_0000	0x0000_0000	Route DestID 0x0000 to port 0.
Port 7 Level 0 Group 0 Entry 0x01	0x0007_0004	0x0000_0001	Route DestID 0x0001 to port 1.
Port 7 Level 0 Group 0 Entry 0x02	0x0007_0008	0x0000_0002	Route DestID 0x0002 to port 2.

Table 0-6. Example 5 Accesses

Register Name	Register Address	Register Value	Description
Port 7 Level 0 Group 0 Entry 0x03	0x0007_000C	0x0000_0003	Route DestID 0x0003 to port 3.
Port 7 Level 0 Group 0 Entry 0x04	0x0007_0010	0x0000_0004	Route DestID 0x0004 to port 4.
Port 7 Level 0 Group 0 Entry 0x05	0x0007_0014	0x0000_0005	Route DestID 0x0005 to port 5.
Port 7 Level 0 Group 0 Entry 0x06	0x0007_0018	0x0000_0006	Route DestID 0x0006 to port 6.
Port 7 Level 0 Group 0 Entry 0x07	0x0007_001C	0x0000_0007	Route DestID 0x0007 to port 7.
Port 7 Level 0 Group 0 Entry 0x08	0x0007_0020	0x0000_0008	Route DestID 0x0008 to port 8.
Port 7 Level 0 Group 0 Entry 0x09	0x0007_0024	0x0000_0009	Route DestID 0x0009 to port 9.
Port 7 Level 0 Group 0 Entry 0x0A	0x0007_0028	0x0000_000A	Route DestID 0x000A to port 10.
Port 7 Level 0 Group 0 Entry 0x0B	0x0007_002C	0x0000_000B	Route DestID 0x000B to port 11.
Port 7 Level 0 Group 0 Entry 0x0C	0x0007_0030	0x0000_000C	Route DestID 0x000C to port 12.
Port 7 Level 0 Group 0 Entry 0x0D	0x0007_0034	0x0000_000D	Route DestID 0x000D to port 13.
Port 7 Level 0 Group 2 Entry 0x00	0x0007_0800	0x0000_000F	Route DestID 0x0200 to port 15.
Port 7 Level 0 Group 3 Entry 0x00	0x0007_0C00	0x0000_000F	Route DestID 0x0300 to port 15.

14 Seed Control Word Bit Ordering Clarification

In Revison 3.0 and 3.1, Part 6, section 5.5.3.3 Descrambler Seed Control Codeword, append the following to the first paragraph:

The seed[0:57] bits map to the scrambling polynomial coefficients discussed in 5.5.4 Scrambling as follows:

- Seed[0] = x^1
- Seed[1] = x^2
- ....
- Seed[57] = x^{58}

Change the second paragraph of Revison 3.0 and 3.1, Part 6, section 5.5.4.2 Descrambler Synchronization to:

The first Descrambler Seed control codeword of the Seed ordered sequence shall be used to re-initialize the state of the descrambler. Refer to 5.5.3.3 Descrambler Seed Control Codeword for the mapping of seed[0:57] to the descrambler coefficients. A Descrambler Seed control codeword shall be determined to be the first in a Seed ordered sequence if the preceding codeword is not a Descrambler Seed control codeword. Based on this definition, only the first Seed ordered sequence of a sequence of consecutive Seed ordered sequences will trigger descrambler re-initialization.

15 Descrambler Seed Ordered Sequence Spacing Clarification

In Revison 3.0 and 3.1, make the following changes:

Part 6, section 5.9.1Seed Ordered Sequence, last paragraph, change the sentence:

“As part of the IDLE3 sequence the Seed ordered sequence shall be transmitted at least once for every 49 codewords transmitted per lane.”
to:
“As part of the IDLE3 sequence the Seed ordered sequence shall be transmitted at least once for every 52 codewords transmitted per lane.”
Part 6, section 5.10.1, change the bullets:
“A Status/Control ordered sequence shall be transmitted once every 18 to 49 codewords transmitted per lane.
A Seed ordered sequence shall be transmitted at least once every 49 codewords transmitted per lane.”
to:
“A Status/Control ordered sequence shall be transmitted once every 18 to 53 codewords transmitted per lane.
A Seed ordered sequence shall be transmitted at least once every 53 codewords transmitted per lane.”