RapidIO™ Interconnect Specification Part 6: LP-Serial Physical Layer Specification

Rev. 2.2, 06/2011

RapidIO Trade Association

Revision History

Revision	Description	Date
1.1	First release	12/17/2001
1.2	Technical changes: incorporate Rev. 1.1 errata rev. 1.1.1, errata 3	06/26/2002
1.3	Technical changes: incorporate Rev 1.2 errata 1 as applicable, the following errata showings: 03-03-00004.002, 03-07-00002.001, 03-12-00000.002, 03-12-00002.004, 04-02-00000.001, 04-05-00000.003, 04-05-00006.002 (partial), 04-05-00007.001 and the following new features showings: 02-03-0003.004, 02-06-00001.004, 04-08-00013.002, 04-09-00022.002 Converted to ISO-friendly templates	02/23/2005
1.3	Removed confidentiality markings for public release	06/07/2005
2.0	Significant editorial changes Technical changes: errata showings 04-11-00031.001, 06-04-00000.003, 06-07-00001.001, 07-03-00000.002, 07-03-00001.001 new features showings 05-04-00001.005, 05-04-00003.004, new speed bin and width definitions with supporting protocol	06/14/2007
2.0.1	Very minor editorial changes	08/29/2007
2.0.1	Removed confidentiality markings for public release	03/06/2008
2.1	Significant editorial changes Technical changes: errata showings 07-09-00000.003, 07-06-00000.010, 08-05-00001.003, 08-03-00000.001, 08-02-00000.008, 08-06-00001.004, 07-11-00001.010, 08-10-00000.003, 08-11-00001.000	MM/DD/200Y
2.1	Removed confidentiality markings for public release	08/13/2009
2.2	Significant editorial changes Technical changes: errata showings 09-08-00000.005, 09-09-00000.004, 10-02-00000.001, 10-03-00000.004, 10-01-00003.006, 10-08-00000.003, 10-08-00001.005, 10-10-00000.002, 10-10-00001.004, 10-11-00002.001, 11-02-00000.002, Consolidated Comments on 11-01-00000.000	05/05/2011
2.2	Removed confidentiality markings for public release	06/06/2011

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

RapidIO Trade Association
12343 Hymeadow, Suite 2-R
(non-US mail deliveries to Suite 3-E)
Austin, TX 78750
512-401-2900 Tel.
512-401-2902 FAX.

Table of Contents

Chapter 1 Overview

Chapter 2 Packets

Chapter 3 Control Symbols

Table of Contents

Table of Contents

Chapter 5 LP-Serial Protocol

Table of Contents

Chapter 6 LP-Serial Registers

Table of Contents

Chapter 7 Signal Descriptions

Chapter 8 Common Electrical Specifications

Table of Contents

Table of Contents

Table of Contents

Chapter 10 5Gbaud and 6.25Gbaud LP-Serial Links

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Blank page

List of Figures

List of Figures

List of Figures

List of Figures

Blank page

List of Tables

List of Tables

List of Tables

List of Tables

Blank page

Chapter 1 Overview

1.1 Introduction

The *RapidIO Part 6: LP-Serial Physical Layer Specification* addresses the physical layer requirements for devices utilizing an electrical serial connection medium. This specification defines a full duplex serial physical layer interface (link) between devices. The links are comprised of one or more lanes, each lane being a pair of unidirectional serial signaling paths with one path in each direction. Further, it allows ganging of up to sixteen serial lanes for applications requiring higher link performance. It also defines a protocol for link management and packet transport over a link.

RapidIO systems are comprised of end point processing elements and switch processing elements. The RapidIO interconnect architecture is partitioned into a layered hierarchy of specifications which includes the Logical, Common Transport, and Physical layers. The Logical layer specifications define the operations and associated transactions by which end point processing elements communicate with each other. The Common Transport layer defines how transactions are routed from one end point processing element to another through switch processing elements. The Physical Layer defines how adjacent processing elements electrically connect to each other. RapidIO packets are formed through the combination of bit fields defined in the Logical, Common Transport, and Physical Layer specifications.

The RapidIO LP-Serial specification defines a protocol for packet delivery between serial RapidIO devices including packet and control symbol transmission, flow control, error management, and other device to device functions. A particular device may not implement all of the mode selectable features found in this document. See the appropriate user's manual or implementation specification for specific implementation details of a device.

The LP-Serial physical layer specification has the following properties:

- Embeds the transmission clock with data using an 8B/10B encoding scheme.
- Supports links with from one lane, up to sixteen ganged lanes where each lane is a pair of unidirectional serial paths with one path in each direction.
- Allows switching packets between RapidIO LP-Serial ports and *RapidIO Part 4: 8/16 LP-LVDS Physical Layer Specification* ports without requiring packet manipulation.

- Employs similar retry and error recovery protocols as the RapidIO 8/16 LP-LVDS physical layer specification.
- Supports transmission rates of 1.25, 2.5, 3.125, 5 and 6.25Gbaud (data rates of 1, 2, 2.5, 4 and 5 Gbps) per lane.
- Supports division of the physical layer bandwidth into up to 9 virtual channels with independent flow control.

This specification first defines the individual elements that make up the link protocol such as packets, control symbols, and the serial bit encoding scheme. This is followed by a description of the link protocol. Finally, the control and status registers, signal descriptions, and electrical specifications are specified.

The virtual channel features are optional. This specification defines a single virtual channel mode of operation that is fully compatible with previous RapidIO specifications.

1.2 Contents

Following are the contents of the *RapidIO Part 6: LP-Serial Physical Layer Specification:*

- Chapter 1, "Overview", (this chapter) provides an overview of the specification
- Chapter 2, "Packets", defines how a RapidIO LP-Serial packet is formed by prefixing a 10-bit physical layer header to the combined RapidIO transport and logical layer bit fields followed by an appended 16-bit CRC field.
- Chapter 3, "Control Symbols", defines the format of the two control symbols (short and long) used for packet acknowledgment, link utility functions, link maintenance, packet delineation, packet acknowledgement and to convey flow control information. They may be transmitted between packets and some may be embedded within a packet.
- Chapter 4, "PCS and PMA Layers", describes the Physical Coding Sublayer (PCS) functionality as well as the Physical Media Attachment (PMA) functionality. The PCS layer functionality includes 8B/10B encoding scheme for embedding clock with data. It also gives transmission rules for the 1x-Nx interfaces and defines the link initialization sequence for clock synchronization. Among other things, the PMA (Physical Medium Attachment) function is responsible for serializing the 10-bit code-groups to and from the serial bitstream(s).
- Chapter 5, "LP-Serial Protocol", describes in detail how packets, control symbols, and the PCS/PMA layers are used to implement the physical layer protocol. This includes topics such as link initialization, link maintenance, error detection and recovery, flow control, bandwidth division, and transaction delivery ordering.

- Chapter 6, "LP-Serial Registers", describes the physical layer control and status register set. By accessing these registers a processing element may query the capabilities and status and configure another LP-Serial RapidIO processing element.
- Chapter 7, "Signal Descriptions", contains the signal pin descriptions for a RapidIO LP-Serial port and shows connectivity between processing elements.
- Chapter 8, "Common Electrical Specifications", Chapter 9, "1.25Gbaud, 2.5Gbaud, and 3.125Gbaud LP-Serial Links", and Chapter 10, "5Gbaud and 6.25Gbaud LP-Serial Links", describe the electrical specifications for a RapidIO LP-Serial device.
- Annex A, “Transmission Line Theory and Channel Information (Informative)", contains a discussion to aid in applying the AC specifications to a system design.
- Annex B, “BER Adjustment Methodology (Informative)", provides recommendations for measuring link error rates.
- Annex C, "Interface Management (Informative)", contains information pertinent to interface management in a RapidIO system, including error recovery, link initialization, and packet retry state machines.
- Annex D, "Critical Resource Performance Limits (Informative)", contains a discussion on outstanding transactions and their relationship to transmission distance capability.
- Annex E, “Manufacturability and Testability (Informative)", recommends implementing to IEEE standard 1149.6 for improved manufacturing and manufacturing test.
- Annex F, “Multiple Port Configuration Example (Informative)", describes an example of a port configuration scenario.

1.3 Terminology

Refer to the Glossary at the back of this document.

1.4 Conventions

||	Concatenation, used to indicate that two fields are physically associated as consecutive bits
ACTIVE_HIGH	Names of active high signals are shown in uppercase text with no overbar. Active-high signals are asserted when high and not asserted when low.
$\overline{\text{ACTIVE\_LOW}}$	Names of active low signals are shown in uppercase text with an overbar. Active low signals are asserted when low and not asserted when high.

italics	Book titles in text are set in italics.
REG[FIELD]	Abbreviations or acronyms for registers are shown in uppercase text. Specific bits, fields, or ranges appear in brackets.
TRANSACTION	Transaction types are expressed in all caps.
operation	Device operation types are expressed in plain text.
n	A decimal value.
[*n*-*m*]	Used to express a numerical range from *n* to *m*.
0b*nn*	A binary value, the number of bits is determined by the number of digits.
0x*nn*	A hexadecimal value, the number of bits is determined by the number of digits or from the surrounding context; for example, 0x*nn* may be a 5, 6, 7, or 8 bit value.
x	This value is a don’t care

Chapter 2 Packets

2.1 Introduction

This chapter specifies the LP-Serial packet format and the fields that are added by LP-Serial physical layer. These packets are fed into the PCS function explained in Chapter 4, "PCS and PMA Layers".

2.2 Packet Field Definitions

This section specifies the bit fields added to a packet by the LP-Serial physical layer. These fields are required to implement the flow control, error management, and other specified system functions of the LP-Serial specification. The fields are specified in Table 2-1.

Table 2-1. Packet Field Definitions

Field	Description
ackID	Acknowledge ID is the packet identifier for link level packet acknowledgment—see Section 5.6.2, "Acknowledgment Identifier" for details concerning ackID functionality. The length of the ackID value depends on the length of the control symbol being used on the link. When the short control symbol is being used, the ackID value shall be 5 bits long and shall be left justified in the ackID field (ackID[0-4]) with the right most bit of the field (ackID[5]) set to 0b0. When the long control symbol is being used, the ackID value shall be 6 bits long which fills the ackID field.
VC	The VC bit specifies the usage of the PRIO and CRF fields. When VC = 0, the PRIO and CRF fields contain the priority bits for a virtual channel 0 packet. When VC = 1 the PRIO and CRF fields contain the Virtual Channel ID for a VC 1-8 packet. See Table 2-2.
prio	Depending on the value of the VC field, PRIO specifies packet priority or contains the most significant bits of the Virtual Channel ID (VCID). See Table 2-2. See Section 5.6.3, "Packet Priority and Transaction Request Flows" for an explanation of prioritizing packets. See Section 5.4, "Virtual Channels" for an explanation of virtual channels.
CRF	Depending on the value of the VC field, CRF differentiates between virtual channel 0 flows of equal priority or contains the least significant bit of the Virtual Channel ID. If VC=0 and Critical Request Flow is not supported, this bit is reserved. See Table 2-2. See Section 5.6.3, "Packet Priority and Transaction Request Flows" for an explanation of prioritizing packets. See Section 5.4, "Virtual Channels" for an explanation of virtual channels.
CRC	Cyclic Redundancy Code used to detect transmission errors in the packet. See Section 2.4.1 for details on the CRC error detection scheme.

Table 2-2 describes the use of the VC, prio, and CRF fields.

Table 2-2. Use of VC, PRIO and CRF Fields

VC	Description
Single VC mode:	
VC = 0	when CRF is RSVD = 0, PRIO sets packet priority as follows: 00 - lowest priority 01 - medium priority 10 - high priority 11 - highest priority
VC = 0	when CRF is supported, PRIO||CRF sets packet priority: 00 0 - lowest priority 00 1 - critical flow lowest priority 01 0 - medium priority 01 1 - critical flow medium priority 10 0 - high priority 10 1 - critical flow high priority 11 0 - highest priority 11 1 - critical flow high priority
Multiple VC Mode:	
VC = 0 VC = 1	VC||PRIO||CRF Channel 0 XX X - VC0 (PRIO, CRF = Priority, same as single VC mode) * 1 00 0 - VC1 (PRIO, CRF = VCID) 1 00 1 - VC2 1 01 0 - VC3 1 01 1 - VC4 1 10 0 - VC5 1 10 1 - VC6 1 11 0 - VC7 1 11 1 - VC8 * Note: VC0 is the backwards-compatibility channel
When Fewer than 8 VCs are supported (in addition to VC0)	
VC = 1	VC||PRIO||CRF Channel 1 00 X - VC1 (VC0 + 4 VCs) 1 01 X - VC3 1 10 X - VC5 1 11 X - VC7 1 0X X - VC1 (VC0 + 2VCs) 1 1X X - VC5 1 XX X - VC1 (VC0 + 1VC)

2.3 Packet Format

This section specifies the format of LP-Serial packets. Figure 2-1 shows the format of the LP-Serial packet and how the physical layer ackID, VC, CRF, and prio fields are prefixed at the beginning of the packet and a 16-bit CRC field is appended to the end of the packet. An additional CRC may be included within the packet (see Section 2.4.1, "Packet CRC Operation", below).

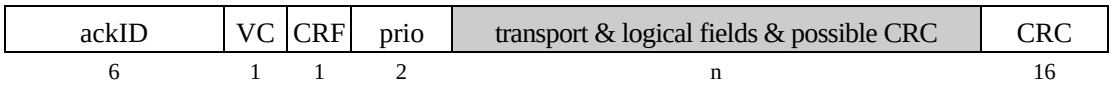


Figure 2-1. Packet Format

The unshaded fields are the fields added by the physical layer. The shaded field is the combined logical and transport layer bits and fields that are passed to the physical layer (also including the possible Section 2.4.1 additional CRC).

LP-Serial packets shall have a length that is an integer multiple of 32 bits. This sizing simplifies the design of port logic whose internal data paths are an integer multiple of 32 bits in width. Packets, as defined in the appropriate logical and transport layer specifications, have a length that is an integer multiple of 16 bits. This is illustrated in Figure 2-2. If the length of a packet defined by the above combination of specifications is an odd multiple of 16 bits, a 16-bit pad whose value is 0 (0x0000) shall be appended at the end of the packet such that the resulting padded packet is an integer multiple of 32 bits in length.

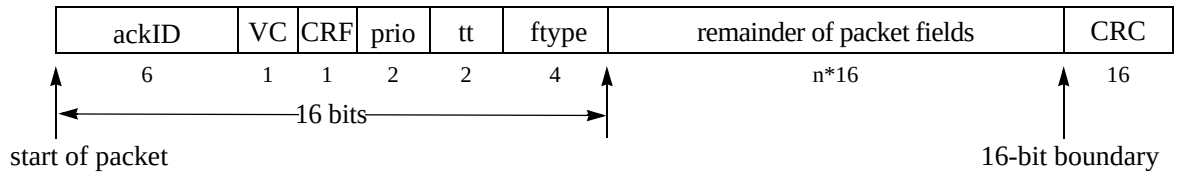


Figure 2-2. Packet Alignment

2.4 Packet Protection

A 16-bit CRC code is added to each packet by the LP-Serial physical layer to provide error detection. The code covers the entire packet except for the ackID field, which is considered to be zero for the CRC calculations. Figure 2-3 shows the CRC coverage for the first 16 bits of the packet which contain the bits not covered by the code.

This structure allows the ackID value to be changed on a link-by-link basis as the packet is transported across the fabric without requiring that the CRC be recomputed for each link. Since ackID values on each link are assigned sequentially for each subsequent transmitted packet, an error in the ackID field is easily detected.

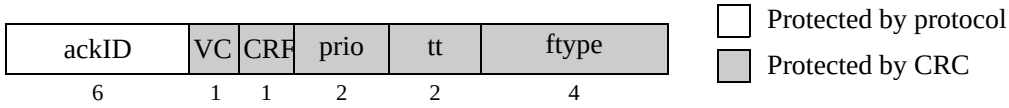


Figure 2-3. Error Coverage of First 16 Bits of Packet Header

2.4.1 Packet CRC Operation

The CRC is appended to a packet in one of two ways. For a packet whose length, exclusive of CRC, is 80 bytes or less, a single CRC is appended at the end of the logical fields. For packets whose length, exclusive of CRC, is greater than 80 bytes, a CRC is added after the first 80 bytes and a second CRC is appended at the end of the logical layer fields.

The second CRC value is a continuation of the first. The first CRC is included in the running calculation, meaning that the running CRC value is not re-initialized after it is inserted after the first 80 bytes of the packet. This allows intervening devices to regard the embedded CRC value as two bytes of packet payload for CRC checking purposes. If the CRC appended to the end of the logical layer fields does not cause the end of the resulting packet to align to a 32-bit boundary, a two byte pad of all logic 0s is postpended to the packet. The pad of logic 0s allows the CRC check to always be done at the 32-bit boundary. A corrupt pad may or may not cause a CRC error to be detected, depending upon the implementation.

The early CRC value can be used by the receiving processing element to validate the header of a large packet and start processing the data before the entire packet has been received, freeing up resources earlier and reducing transaction completion latency.

NOTE:

While the embedded CRC value can be used by a processing element to start processing the data within a packet before receiving the entire packet, it is possible that upon reception of the end of the packet the final CRC value for the packet is incorrect. This would result in a processing element that has processed data that may have been corrupted. Outside of the error recovery mechanism described in Section 5.13.2, "Link Behavior Under Error", the *RapidIO Interconnect Specification* does not address the occurrence of such situations nor does it suggest a means by which a processing element would handle such situations. Instead, the mechanism for handling this situation is left to be addressed by the device manufacturers for

devices that implement the functionality of early processing of packet data.

Figure 2-4 is an example of an unpadded packet of length less than or equal to 80 bytes.

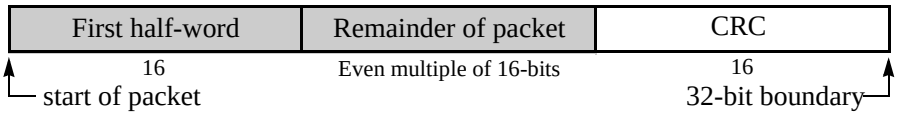


Figure 2-4. Unpadded Packet of Length 80 Bytes or Less

Figure 2-5 is an example of a padded packet of length less than or equal to 80 bytes.

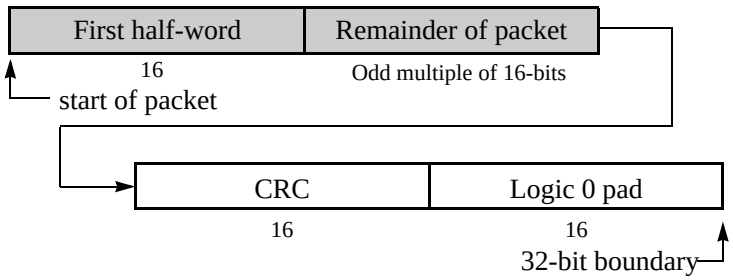


Figure 2-5. Padded Packet of Length 80 Bytes or Less

Figure 2-6 is an example of an unpadded packet of length greater than 80 bytes.

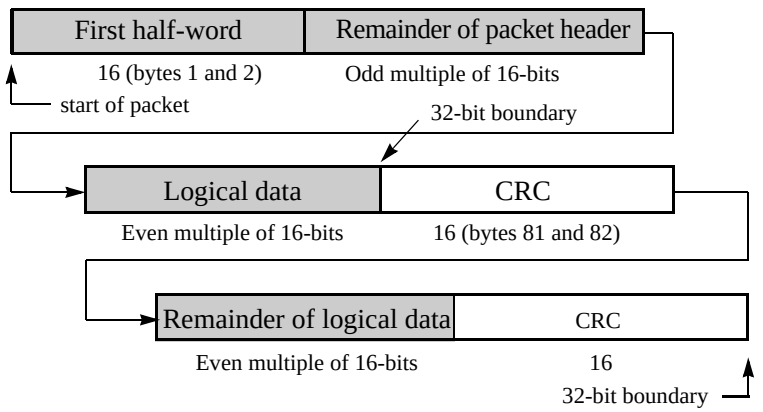


Figure 2-6. Unpadded Packet of Length Greater than 80 Bytes

Figure 2-7 is an example of a padded packet of length greater than 80 bytes.

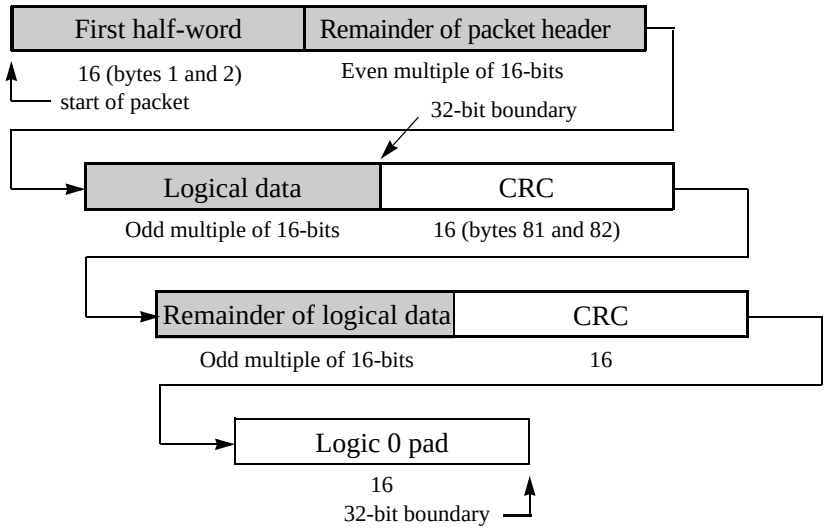


Figure 2-7. Padded Packet of Length Greater than 80 Bytes

2.4.2 CRC-16 Code

The ITU polynomial $X^{16}+X^{12}+X^5+1$ shall be used to generate the 16-bit CRC for packets. The value of the CRC shall be initialized to 0xFFFF (all logic 1s) at the beginning of each packet. For the CRC calculation, the uncovered six bits are treated as logic 0s. As an example, a 16-bit wide parallel calculation is described in the equations in Table 2-3. Equivalent implementations of other widths can be employed.

Table 2-3. Parallel CRC-16 Equations

Check Bit	e00	e01	e02	e03	e04	e05	e06	e07	e08	e09	e10	e11	e12	e13	e14	e15
C00					x	x			x				x			
C01						x	x			x				x		
C02							x	x			x				x	
C03	x							x	x			x				x
C04	x	x			x	x				x						
C05		x	x			x	x				x					
C06	x		x	x			x	x				x				
C07	x	x		x	x			x	x				x			
C08	x	x	x		x	x			x	x				x		
C09		x	x	x		x	x			x	x				x	
C10			x	x	x		x	x			x	x				x
C11	x			x				x				x				
C12	x	x			x				x				x			

Table 2-3. Parallel CRC-16 Equations (Continued)

Check Bit	e00	e01	e02	e03	e04	e05	e06	e07	e08	e09	e10	e11	e12	e13	e14	e15
C13		x	x			x				x				x		
C14			x	x			x				x				x	
C15				x	x			x				x				x

where:

C00–C15 contents of the new check symbol

e00–e15 contents of the intermediate value symbol
e00 = d00 XOR c00
e01 = d01 XOR c01
through
e15 = d15 XOR c15

d00–d15 contents of the next 16 bits of the packet

c00–c15 contents of the previous check symbol

assuming the pipeline described in Figure 2-8

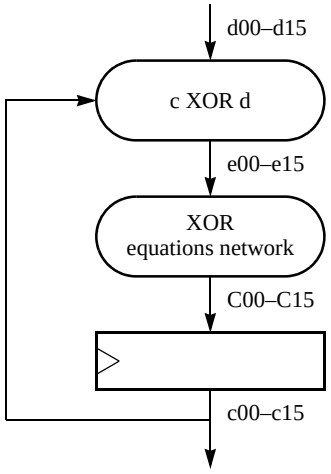


Figure 2-8. CRC Generation Pipeline

2.5 Maximum Packet Size

The RapidIO Specification does not contain an overall specification for the maximum size of a packet that a logical layer may pass to the transport layer or the transport layer may pass to a physical layer. Maximum sizes can only be determined by examining the format of each packet type at the logical layer and the operation of the transport and physical layers.

The longest packets are those containing an operand address within the destination device, an operand size and a maximum length payload (256 bytes). Currently the largest packet format is the type 5 (write class) format, defined in the I/O Logical specification. The sizes of the components of the maximum packet are shown in more detail in Table 2-4.

Table 2-4. Maximum Packet Size

Field	Size (bytes)	Layer	Notes
Header	2	Logical, Transport, Physical	See Figure 2-2
Source ID	2	Transport	Large transport
Destination ID	2	Transport	Large transport
Trans/wrsize	1	Logical	Type 5 (write class)
srcTID	1	Logical	Type 5 (write class)
Address	8	Logical	Type 5 (write class); includes Extended_address, Address, Wdptr, and Xambs
Payload	256	Logical	Maximum data payload
CRC	4	Physical	Includes two CRC additional bytes for packets greater than 80 bytes
Total	276		

The maximum transmitted packet size permitted by the LP-Serial specification is 276 bytes. This includes all packet logical, transport, and physical layer header information, data payload, and required CRC bytes, but does not include any packet delimiting control symbols or other necessary physical layer control information.

Chapter 3 Control Symbols

3.1 Introduction

This chapter specifies RapidIO physical layer control symbols. Control symbols are the message elements used by ports connected by a LP-Serial link to manage all aspects of LP-Serial link operation. They are used for link maintenance, packet delimiting, packet acknowledgment, error reporting, and error recovery.

Two control symbols are defined. The first one is three (3) characters long and is referred to as the “short” control symbol. The second one is six (6) characters long and is referred to as the “long” control symbol.

The short control symbol was the first control symbol defined for LP-Serial links. It was designed for links operating at less than 5.5 GBaud per lane and receivers that do not employ decision feedback equalization (DFE). It provides the functionality needed for the basic link protocol plus some extensions to the link protocol.

The long control symbol is an extension of the short control symbol. It was designed for links operating at greater than 5.5 GBaud per lane and receivers employing DFE. The additional characters are required in part to provide stronger error detection for burst errors that are characteristic of receivers using DFE. The additional characters are also available to provide support for link protocol extensions beyond those supported by the short control symbol.

When use of the long control symbol is supported by both ends of a LP-Serial link operating at less than 5.5 GBaud per lane, it may be used instead of the short control symbol to provide more control symbol functionality.

LP-Serial control symbols carry at least two independent functions. Each function is assigned one or more control symbol fields for its use. One of the fields assigned to a function specifies the primary function type. The other fields assigned to the function may, depending on the primary function type, further specify the function type, contain information required for the execution of the function, contain “supplemental information” that is not required for the execution of the function and whose value does not affect the behavior of the receiving port, or be unused. Fields that specify the function type or contain data required for the execution of the function are called “functional” fields. Fields that contain “supplemental information” are called “informational” fields. All fields are functional unless specified otherwise.

For forward compatibility, a control symbol function received by a port with an encoding in one or more of the fields assigned to the function that the port does not understand or support shall be handled as follows. If an encoding that the port does not understand or support occurs in a functional field, the control symbol function shall be ignored. If an encoding that the port does not understand or support occurs only in an informational field, the control symbol function shall be executed. In either case, no error shall be reported.

3.2 Control Symbol Field Definitions

This section describes the fields that make up the control symbols.

Table 3-1. Control Symbol Field Definitions

Field	Definition
stype0	Encoding for control symbols that make use of parameter0 and parameter1. Eight encodings are defined in Table 3-2.
parameter0	Used in conjunction with stype0 encodings. Reference Table 3-2 for the description of parameter0 encodings.
parameter1	Used in conjunction with stype0 encodings. Reference Table 3-2 for the description of parameter1 encodings.
stype1	Encoding for control symbols which make use of the cmd field. The eight encodings are defined in Table 3-7.
cmd	Used in conjunction with the stype1 field to define the link maintenance commands. Refer to Table 3-7 for the cmd field descriptions.
reserved	Set to logic 0s on transmission and ignored on reception
CRC-5	5-bit code used to detect transmission errors in short control symbols. See Section 3.6, "Control Symbol Protection" for details on the CRC error detection scheme.
CRC-13	13-bit code used to detect transmission errors in long control symbols. See Section 3.6, "Control Symbol Protection" for details on the CRC error detection scheme.

3.3 Control Symbol Format

This section describes the general formats of the LP-Serial control symbols. All short control symbols shall have the 24 data bit format shown in Figure 3-1.

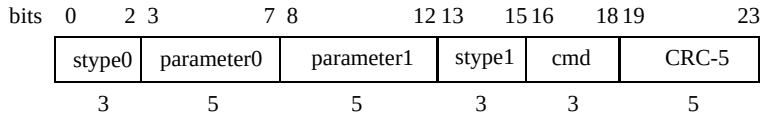


Figure 3-1. Short Control Symbol Format

Short control symbols can carry two functions, one encoded in the stype0 field and one encoded in the stype1 field. The fields parameter0 and parameter1 are used by the functions encoded in the stype0 field. The cmd field is a modifier for the functions encoded in the stype1 field.

The functions encoded in stype0 are “status” functions that convey some type of status about the port transmitting the control symbol. The functions encoded in stype1 are requests to the receiving port or transmission delimiters.

All long control symbols shall have the 48 data bit format shown in Figure 3-2.

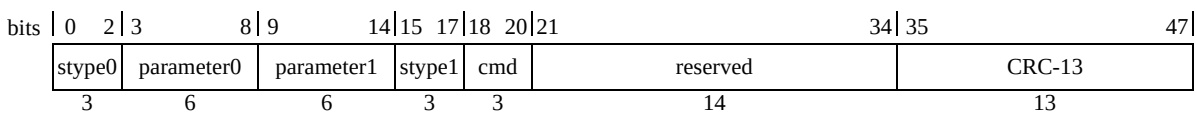


Figure 3-2. Long Control Symbol Format

With one exception, the stype0, parameter0, parameter1, stype1, and cmd fields in the long control symbol have exactly the same function, encoding, and size as the same named fields in the short control symbol. The exception is that parameter0 and parameter1 are 5-bit fields in the short control symbol and 6-bit fields in the long control symbol.

Control symbols are defined with the ability to carry at least two functions so that a packet acknowledgment and a packet delimiter can be carried in the same control symbol. Packet acknowledgment and packet delimiter control symbols constitute the vast majority of control symbol traffic on a busy link. Carrying an acknowledgment (or status) and a packet delimiter whenever possible in a single control symbol allows a significant reduction in link overhead traffic and an increase in the link bandwidth available for packet transmission.

A control symbol carrying one function is referred to using the name of the function it carries. A control symbol carrying more than one functions may be referred to using the name of any function that it carries. For example, a control symbol with stype0 set to packet-accepted and stype1 set to NOP is referred to a packet-accepted control symbol. A control symbol with stype0 set to packet-accepted and stype1 set to restart-from-retry is referred to as either a packet-accepted control symbol or a restart-from-retry control symbol depending on which name is appropriate for the context.

3.4 Stype0 Control Symbols

The encoding and function of stype0 and the information carried in parameter0 and parameter1 for each stype0 encoding shall be as specified in Table 3-2.

Table 3-2. Stype0 Control Symbol Encoding

stype0	Function	Contents of		Reference
		Parameter0	Parameter1	
0b000	packet-accepted	packet_ackID	buf_status	Section 3.4.1
0b001	packet-retry	packet_ackID	buf_status	Section 3.4.2
0b010	packet-not-accepted	arbitrary	cause	Section 3.4.3
0b011	reserved	—	—	—
0b100	status	ackID_status	buf_status	Section 3.4.4
0b101	VC_status	VCID	buf_status	Section 3.4.5
0b110	link-response	ackID_status	port_status	Section 3.4.6
0b111	implementation-defined *	implementation-defined	implementation-defined	—

* While implementation-defined control symbols are allowed, their use can result in inter-operability problems and is not recommended. There is no registration facility for implementation-defined control symbols. As a result, two implementations may assign different

meanings to the same encoding of the Parameter0 and/or Parameter1 fields which could result in undefined and/or inconsistent behavior, data corruption, or system failure.

The packet-accepted, packet-retry and packet-not-accepted control symbols are collectively referred to as “packet acknowledgment” control symbols.

“Status” (0b100) is the default stype0 encoding and is used when a control symbol does not convey another stype0 function.

Table 3-3 defines the parameters valid for stype0 control symbols and that are used for more than one value of stype0.

Table 3-3. Stype0 Parameter Definitions

Parameter	Definition
packet_ackID	The ackID of the packet being acknowledged.
ackID_status	The value of the ackID field expected in the next packet the port receives. This value is 1 greater than the ackID of the last packet accepted by the port exclusive of CT mode packets accepted after the port entered an Input-stopped state. For example, a value of 0b00001 (short control symbol) or 0b000001 (long control symbol) indicates that the ackID of the last packet accepted by the port exclusive of CT mode packets accepted after the port entered an Input-stopped state was 0 and that the port is expecting to receive a packet with an ackID field value of 1.
buf_status	The number of maximum length packet buffers the port has available for packet reception on the specified virtual channel (VC) at the time the control symbol containing the field is generated. The value of the buf_status field in a packet-accepted control symbol is inclusive of the receive buffer consumption of the packet being accepted. The field is used in transmitter controlled flow control to control the rate at which packets are transmitted to prevent loss of packets at the receiver due to a lack of packet buffers. **For short control symbols:** Value 0-29: The encoded value is the number of maximum sized packet buffers the port has available for reception on the specified VC. The value 0, for example, signifies that the port has no packet buffers available for the specified VC (thus is not able to accept any new packets for that VC). Value 30: The value 30 indicates that the port has at least 30 maximum length packet buffers available for reception on the specified VC. Value 31: The port has an undefined number of maximum sized packet buffers available for packet reception, and relies on retry for flow control. **For long control symbols:** Value 0-61: The encoded value is the number of maximum sized packet buffers the port has available for reception on the specified VC. The value 0, for example, signifies that the port has no packet buffers available for the specified VC (thus is not able to accept any new packets for that VC). Value 62: The value 62 indicates that the port has at least 62 maximum length packet buffers available for reception on the specified VC. Value 63: The port has an undefined number of maximum sized packet buffers available for packet reception, and relies on retry for flow control

NOTE:

The following sections depict various control symbols. Since control symbols can contain one or more functions, shading in the figures is used to indicate which fields are applicable to that specific control symbol function.

3.4.1 Packet-Accepted Control Symbol

The packet-accepted control symbol indicates that the port sending the control symbol has taken responsibility for sending the packet to its final destination and that resources allocated to the packet by the port receiving the control symbol can be released. This control symbol shall be generated only after the entire packet has been received and found to be free of detectable errors. The packet-accepted control symbol formats are displayed in Figure 3-3.

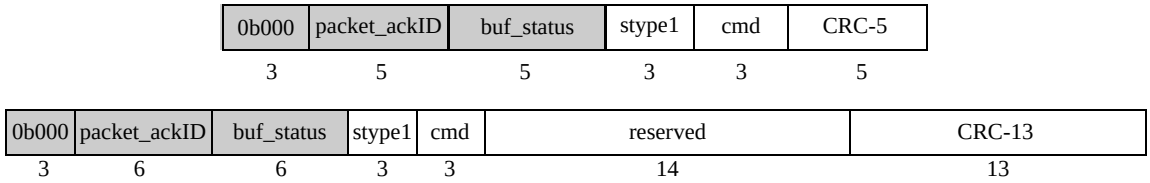


Figure 3-3. Packet-Accepted Control Symbol Formats

The buf_status value in the control symbol is for the VC of the packet being accepted. Since the VC of the packet is not carried in the control symbol, the port receiving the control symbol must reassociate the ackID in the packet_ackID field with the VC of the accepted packet to determine the VC to which the buf_status applies.

3.4.2 Packet-Retry Control Symbol

A packet-retry control symbol indicates that the port sending the control symbol was not able to accept the packet due to some temporary resource conflict such as insufficient buffering and the packet must be retransmitted. The control symbol formats are displayed in Figure 3-4.

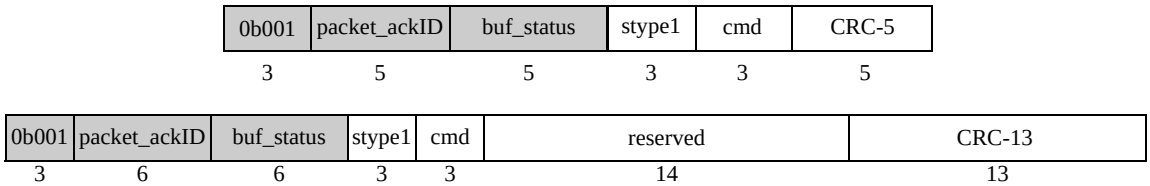


Figure 3-4. Packet-Retry Control Symbol Formats

The packet-retry control symbol is only used in singleVC mode for compatibility with Rev. 1.x RapidIO devices. Packet retry is replaced with error recovery when multiple VCs are active. See Chapter 5, "LP-Serial Protocol", for more information.

3.4.3 Packet-Not-Accepted Control Symbol

The packet-not-accepted control symbol indicates that the port sending the control symbol has either detected an error in the received character stream or, when operating in multiple VC mode, has insufficient buffer resources and as a result may have rejected a packet or control symbol. The control symbol contains a “arbitrary” field and a “cause” field. The control symbol formats are shown in Figure 3-5.

Figure 3-5. Packet-Not-Accepted Control Symbol Formats

The “arbitrary” field contains an arbitrary number and conveys no useful information.

The “cause” field is used to provide information about the type of error that was detected for diagnostics and debug use. The content of the cause field is informational only.

The cause field shall be encoded as specified in Table 3-4 which lists a number of common faults and their encodings. If the port issuing the control symbol is not able to specify the fault, or the fault is not one of those listed in the table, the general error encoding shall be used.

Table 3-4. Cause Field Definition

Cause	Definition
0b00000	Reserved
0b00001	Received packet with an unexpected ackID
0b00010	Received a control symbol with bad CRC
0b00011	Non-maintenance packet reception is stopped
0b00100	Received packet with bad CRC
0b00101	Received invalid character, or valid but illegal character
0b00110	Packet not accepted due to lack of resources
0b00111	Loss of descrambler sync
0b01000 - 0b11110	Reserved
0b11111	General error

3.4.4 Status Control Symbol

The status control symbol indicates receive status information about the port sending the control symbol. The control symbol contains the ackID_status and the buf_status fields. The ackID_status field allows the receiving port to determine if it and the sending port are in sync with respect to the next ackID value the sending port expects to receive. The buf_status field indicates to the receiving port the number of maximum length packet buffers the sending port has available for reception on VC0 as defined in Table 3-3.

“Status” is the default stype0 encoding and is used when the control symbol does not convey another stype0 function.

The status control symbol formats are shown in Figure 3-6.

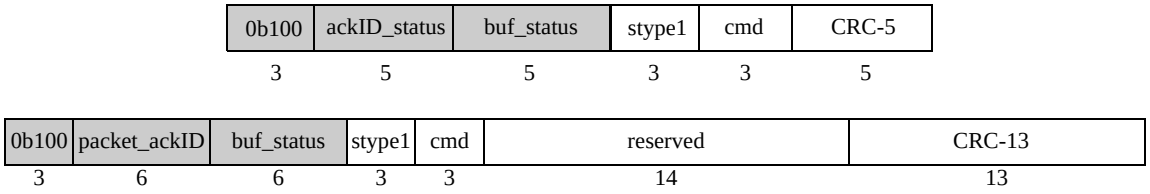


Figure 3-6. Status Control Symbol Formats

3.4.5 VC-Status Control Symbol

The VC-status control symbol indicates to the receiving port the available buffer space that the sending port has available for packet reception on the virtual channel (VC) specified in the control symbol. The VC-status control symbol is used only for virtual channels 1 through 8 (VC1 through VC8) and may be transmitted only when the specified VC is implemented and enabled. (The status control symbol described in Section 3.4.4, "Status Control Symbol" provides this function for VC0.)

The VCID field specifies the VC to which the control symbol applies. VCID is 3-bit field that is right justified in the Parameter0 field of the control symbol. The remaining bits of the parameter0 field are reserved, set to 0 on transmission and ignored on reception. The buf_status field indicates to the receiving port the number of maximum length packet buffers the sending port has available for reception on the specified VC as defined in Table 3-3.

The VC-status control symbol may be transmitted at any time and should be transmitted whenever the number of maximum length packet buffers available for reception on a VC has changed and has not been otherwise communicated to the connected port.

The VC-status control symbol formats are shown in Figure 3-7.

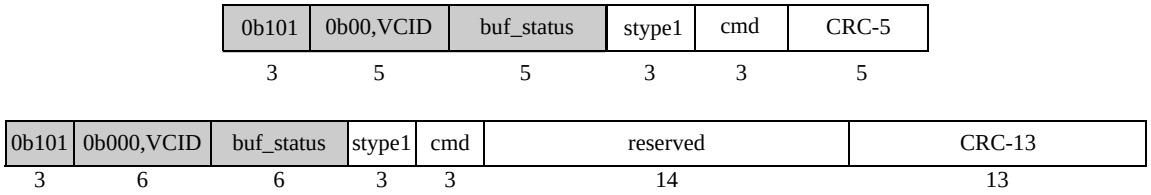


Figure 3-7. VC_Status Control Symbol Formats

The encoding of the VCID field is specified in Table 3-5. The VCID corresponds to the VCID in the physical layer format as described in Chapter 2, "Packets".

Table 3-5. VCID Definition

8 Optional VCs Active	
VCID	**Definition**
0b000	VC1
0b001	VC2
0b010	VC3
0b011	VC4
0b100	VC5
0b101	VC6
0b110	VC7
0b111	VC8

4 Optional VCs Active	
VCID	**Definition**
0b00x	VC1
0b01x	VC3
0b10x	VC5
0b11x	VC7

2 Optional VCs Active	
VCID	**Definition**
0b0xx	VC1
0b1xx	VC5

1 Optional VC Active	
VCID	**Definition**
0bxxx	VC1

Active VCs are in addition to VC0

Formats for 4, 2, and 1 active VCs are shown in the three right hand columns of the table. When using fewer than 8 VCs, bits in the VCID are ignored starting from the LSB, consistent with the bit usage in the packet format. For example, with one optional VC active, all bit patterns in the VCID are interpreted as pertaining to VC1.

3.4.6 Link-Response Control Symbol

The link-response control symbol is used by a port to respond to a link-request control symbol as described in the link maintenance protocol described in Section 5.7, "Link Maintenance Protocol". The status reported in the status field is the status of the port at the time the associated input-status link-request control symbol was received and is informational only.

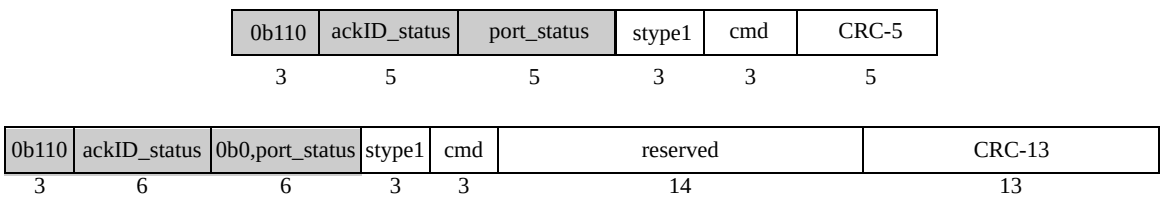


Figure 3-8. Link-Response Control Symbol Formats

The encoding of the link-response control symbol port_status field is defined in Table 3-6.

Table 3-6. Port_status Field Definitions

Port_status	Status	Description
0b00000 - 0b00001	—	Reserved
0b00010	Error	The port has encountered an unrecoverable error and is unable to accept packets.
0b00011	—	Reserved
0b00100	Retry-stopped	The port has retried a packet and is waiting in the input retry-stopped state to be restarted.
0b00101	Error-stopped	The port has encountered a transmission error and is waiting in the input error-stopped state to be restarted.
0b00110 - 0b01111	—	Reserved
0b10000	OK	The port is accepting packets
0b10001 - 0b11111	—	Reserved

3.5 Stype1 Control Symbols

The encoding of stype1 and the function of the cmd field are defined in Table 3-7.

Table 3-7. Stype1 Control Symbol Encoding

stype1	stype1 Function	cmd	cmd Function	Packet Delimiter	Reference
0b000	Start-of-packet	0b000	Start-of-packet	yes	Section 3.5.1
		0b001 - 0b111	Reserved	No	
0b001	Stomp	0b000	Stomp	yes	Section 3.5.2
		0b001 - 0b111	Reserved	No	
0b010	End-of-packet	0b000	End-of-packet	yes	Section 3.5.3
		0b001 - 0b111	Reserved	No	
0b011	Restart-from-retry	0b000	Restart-from-retry	*	Section 3.5.4
		0b001 - 0b111	Reserved	No	
0b100	Link-request	0b000 - 0b010	Reserved	*	-
		0b011	Reset-device		Section 3.5.5.1
		0b100	Input-status		Section 3.5.5.2
		0b101- 0b111	Reserved		-
0b101	Multicast-event	0b000	Multicast-event	No	Section 3.5.6
		0b001 - 0b111	Reserved	No	
0b110	Reserved	0b000 - 0b111	Reserved	No	-
0b111	NOP (Ignore) **	0b000	NOP (Ignore) **	No	-
		0b001 - 0b111	Reserved	No	

Note: * denotes that restart-from-retry and link-request control symbols may only be packet delimiters if a packet is in progress.

Note: ** NOP (Ignore) is not defined as a control symbol, but is the default value when the control symbol does not convey another stype1 function.

NOTE:

The following sections depict various control symbols. Since control symbols can contain one or two functions, shading in the figures is used to indicate which fields are applicable to that specific control symbol function.

3.5.1 Start-of-Packet Control Symbol

The start-of-packet control symbol is used to delimit the beginning of a packet. The control symbol formats are shown in Figure 3-9.

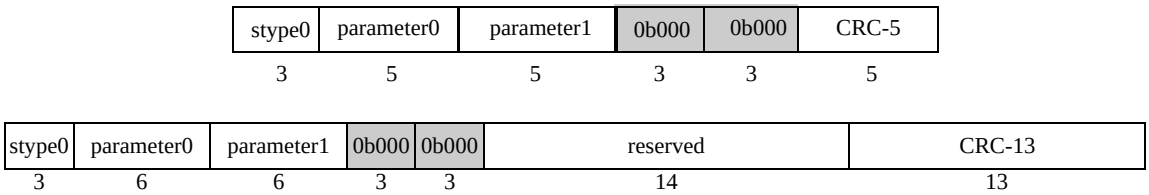


Figure 3-9. Start-of-Packet Control Symbol Formats

3.5.2 Stomp Control Symbol

The stomp control symbol is used to cancel a partially transmitted packet. The protocol for packet cancellation is specified in Section 5.10, "Canceling Packets". The stomp control symbol formats are shown in Figure 3-10 below.

Figure 3-10. Stomp Control Symbol Formats

3.5.3 End-of-Packet Control Symbol

The end-of-packet control symbol is used to delimit the end of a packet. The control symbol formats are shown in Figure 3-11.

Figure 3-11. End-of-Packet Control Symbol Formats

3.5.4 Restart-From-Retry Control Symbol

This control symbol is used to mark the beginning of packet retransmission, so that the receiver knows when to start accepting packets after the receiver has requested a packet to be retried. The restart-from-retry control symbol cancels a current packet and may also be transmitted on an idle link. The control symbol formats are shown in Figure 3-12 below.

Figure 3-12. Restart-From-Retry Control Symbol Formats

3.5.5 Link-Request Control Symbol

A link-request control symbol is used by a port to either issue a command to the connected port or request its input port status. A link-request control symbol always cancels a packet whose transmission is in progress and can also be sent between packets. Under error conditions, a link-request/input-status control symbol acts as a link-request/restart-from-error control symbol as described in Section 5.7, "Link Maintenance Protocol". The control symbol formats are displayed in Figure 3-13.

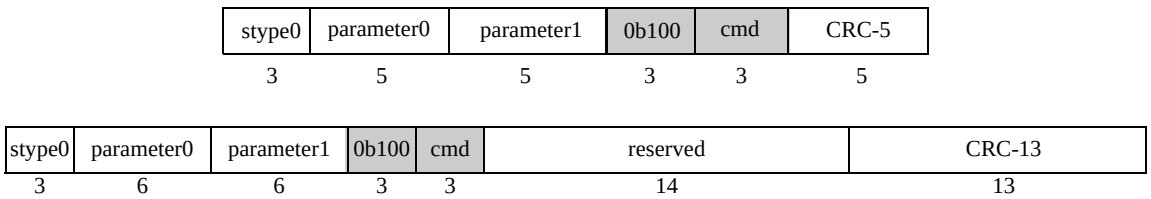


Figure 3-13. Link-Request Control Symbol Formats

The use of the “cmd” field in the link-request control symbol format is defined in Table .

Table 3-8. Cmd Field Definitions

cmd Encoding	Command Name	Description	Reference
0b000-0b010	—	Reserved	
0b011	Reset-device	Reset the receiving device	Section 3.5.5.1
0b100	Input-status	Return input port status; functions as a link request (restart-from-error) control symbol under error conditions	Section 3.5.5.2
0b101-0b111	—	Reserved	

3.5.5.1 Reset-Device Command

The reset-device command causes the receiving device to go through its reset or power-up sequence. All state machines and the configuration registers reset to the original power on states. The reset-device command does not generate a link-response control symbol.

Due to the undefined reliability of system designs it is necessary to put a safety lockout on the reset function of the link-request control symbol. A port receiving a reset-device command in a link-request control symbol shall not perform the reset function unless it has received four reset-device commands in a row without any other intervening packets or control symbols, except status control symbols. This will prevent spurious reset commands from inadvertently resetting a device.

When issuing a reset with four consecutive reset commands, care must be taken to account for all effects associated with the reset event. Consult *RapidIO Part 8: Error Management Extensions Specification* for more information.

3.5.5.2 Input-Status Command

The input-status command requests the receiving port to return a link-response containing the ackID value it expects to next receive on its input port and the current input port operational status for informational purposes. This command causes the receiver to flush its output port of all control symbols generated by packets received before the input-status command. Flushing the output port is implementation dependent and may result in either discarding the contents of the receive buffers or sending the control symbols on the link. The receiver then responds with a link-response control symbol.

3.5.6 Multicast-Event Control Symbol

The multicast-event control symbol allows the occurrence of a user-defined system event to be multicast throughout a system. The multicast-event control symbol differs from other control symbols in that it carries information not related to the link carrying the control symbol. Refer to Section 5.5.3.4, "Multicast-Event Control Symbols" for more details on Multicast-Events.

The multicast-event control symbol formats are shown in Figure 3-14 below.

Figure 3-14. Multicast-Event Control Symbol Formats

3.6 Control Symbol Protection

Control symbol error detection is provided by a cyclic redundancy check (CRC) code.

A 5-bit CRC is used for the short control symbol. It provides detection of a single burst error of 5 bits or less in the 24 data bits of the 8B/10B decoded control symbol. A single 5 bit burst error is the longest burst error that can be caused by a single bit transmission error at the 8B/10B code-group level.

A 13-bit CRC is used for the long control symbol. It provides detection of any set of errors in the 48 data bits of the 8B/10B decoded control symbol that can be caused by a burst error on one lane of 11 bits or less at the 8B/10B code-group level. An 11 bit error at the code-group level can corrupt at most two code-groups.

3.6.1 CRC-5 Code

The ITU polynomial $X^5+X^4+X^2+1$ shall be used to generate the 5-bit CRC for short control symbols.

The 5-bit CRC shall be computed over 20 bits comprised of control symbol bits 0 through 18 plus a 20^{th} bit that is appended after bit 18 of the control symbol. The added bit shall be set to logic 0 (0b0). The 20^{th} bit is added in order to provide maximum implementation flexibility for all types of designs. The CRC shall be computed beginning with control symbol bit 0. Before the CRC is computed, the CRC shall be set to all 1's (0b11111).

The CRC check bits c[0:4] occupy short control symbol bits [19:23] respectively.

The 5-bit CRC shall be generated by each transmitter and verified by each receiver using the short control symbol.

3.6.2 CRC-5 Parallel Code Generation

Since it is often more efficient to implement a parallel CRC algorithm rather than a serial, examples of the equations for a complete, 19-bit single-stage parallel implementation is shown in shown in Table 3-9. Since only a single stage is used,

the effect of both setting the initial CRC to all 1's (0b11111) and a 20th bit set to logic 0 (0b0) have been included in the equations.

In Table 3-9, an "x" means that the data input should be an input to the Exclusive-OR necessary to compute that particular bit of the CRC. A "!x", means that bit 18 being applied to the CRC circuit must be inverted.

Table 3-9. Parallel CRC-5 Equations

Control Symbol	CRC Checksum Bits				
Data for CRC	**C0**	**C1**	**C2**	**C3**	**C4**
D18	x	!x	!x	!x	x
D17		x		x	x
D16	x		x	x	
D15	x	x			x
D14			x	x	x
D13		x	x	x	
D12	x	x	x		
D11		x	x		x
D10	x	x		x	
D9					x
D8				x	
D7			x		
D6		x			
D5	x				
D4	x		x		x
D3	x	x	x	x	x
D2		x		x	x
D1	x		x	x	
D0	x	x			x

Figure 3-15 shows the 19-bits that the CRC covers and how they should be applied to the circuit. As seen in Figure 3-15, bits are labeled with 0 on the left and 18 on the right. Bit 0, from the stype0 field, would apply to D0 in Table 3-9 and bit 18, from the cmd field, would apply to D18 in Table 3-9. Once completed, the 5-bit CRC is appended to the control symbol.

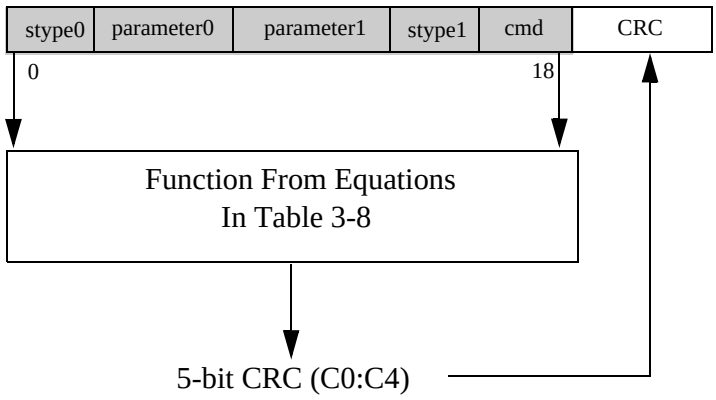


Figure 3-15. CRC-5 Implementation

3.6.3 CRC-13 Code

The polynomial $x^{13} + x^{10} + x^8 + x^5 + x^2 + 1$ shall be used to generate the 13-bit CRC for long control symbols.

The 13-bit CRC shall be computed over control symbol bits 0 through 34 beginning with control symbol bit 0. Before the 13-bit CRC is computed, the CRC shall be set to all 0's (0b0_0000_0000_0000).

The CRC check bits c[0:12] shall occupy long control symbol bits [35:47] respectively.

The 13-bit CRC shall be generated by each transmitter and verified by each receiver using the long control symbol.

3.6.4 CRC-13 Parallel Code Generation

For the CRC-13 parallel code generation, the equations are shown in Figure 3-10, using rules as for the CRC-5 parallel generation.

Table 3-10. Parallel CRC-13 Equations

Control Symbol	CRC Checksum Bits												
Data For CRC	C0	C1	C2	C3	C4	C5	C6	C7	C8	C9	C 10	C 11	C 12
D34			x		x			x			x		x
D33		x		x			x			x		x	
D32	x		x			x			x		x		
D31		x	x							x	x		x
D30	x	x							x	x		x	
D29	x		x		x				x				x

Control Symbol	CRC Checksum Bits												
D28		x	x	x	x						x	x	x
D27	x	x	x	x						x	x	x	
D26	x	x			x			x	x	x			x
D25	x		x	x	x		x		x		x	x	x
D24		x		x	x	x				x		x	x
D23	x		x	x	x				x		x	x	
D22		x		x	x					x			x
D21	x		x	x					x			x	
D20		x			x								x
D19	x			x								x	
D18					x			x					x
D17				x			x					x	
D16			x			x					x		
D15		x			x					x			
D14	x			x					x				
D13					x						x		x
D12				x						x		x	
D11			x						x		x		
D10		x						x		x			
D9	x						x		x				
D8			x		x	x					x		x
D7		x		x	x					x		x	
D6	x		x	x					x		x		
D5		x			x					x	x		x
D4	x			x					x	x		x	
D3					x				x				x
D2				x				x				x	
D1			x				x				x		
D0		x				x				x			

Chapter 4 PCS and PMA Layers

4.1 Introduction

This chapter specifies the functions provided by the Physical Coding Sublayer (PCS) and Physical Media Attachment (PMA) sublayer. (The PCS and PMA terminology is adopted from IEEE 802.3). The topics include character representation, scrambling, lane striping, 8B/10B encoding, serialization of the data stream, code-groups, columns, link transmission rules, idle sequences, and link initialization.

The concept of lanes is used to describe the width of a LP-Serial link. A lane is a single unidirectional signal path between two LP-Serial ports. Five widths are defined for LP-Serial links, 1, 2, 4, 8 and 16 lanes per direction. A link with N lanes in each direction is referred to as a Nx link, e.g. a link with 4 lanes in each direction is referred to as a 4x link.

4.2 PCS Layer Functions

The Physical Coding Sublayer (PCS) function is responsible for idle sequence generation, lane striping, scrambling and encoding for transmission and decoding, lane alignment, descrambling and destriping on reception. The PCS uses an 8B/10B encoding for transmission over the link.

The PCS layer also provides mechanisms for determining the operational mode of the port as Nx or 1x operation, and means to detect link states. It provides for clock difference tolerance between the sender and receiver without requiring flow control.

The PCS layer performs the following transmit functions:

- Dequeues packets and delimited control symbols awaiting transmission as a character stream.
- Scrambles packet and control symbol data if required.
- Stripes the transmit character stream across the available lanes.
- Generates the idle sequence and inserts it into the transmit character stream for each lane when no packets or delimited control symbols are available for transmission.
- Encodes the character stream of each lane independently into 10-bit parallel code-groups.

- Passes the resulting 10-bit parallel code-groups to the PMA.

The PCS layer performs the following receive functions:

- Decodes the received stream of 10-bit parallel code-groups for each lane independently into characters.
- Marks characters decoded from invalid code-groups as invalid.
- If the link is using more than one lane, aligns the character streams to eliminate the skew between the lanes and reassembles (destripes) the character stream from each lane into a single character stream.
- Descrambles packet and control symbol data if required.
- Delivers the decoded character stream of packets and delimited control symbols to the higher layers.

4.3 PMA Layer Functions

The PM (Physical Medium Attachment) layer is responsible for serializing 10-bit parallel code-groups to/from a serial bitstream on a lane-by-lane basis. Upon receiving data, the PMA function provides alignment of the received bitstream to 10-bit code-group boundaries, independently on a lane-by-lane basis. It then provides a continuous stream of 10-bit code-groups to the PCS, one stream for each lane. The 10-bit code-groups are not observable by layers higher than the PCS.

If a LP-Serial port support supports either baud rate discovery or adaptive equalization, these functions are also performed in the PMA layer.

4.4 Definitions

Definitions of terms used in this specification are provided below.

Byte: An 8-bit unit of information. Each bit of a byte has the value 0 or 1.

Character: A 9-bit entity comprised of an information byte and a control bit that indicates whether the information byte contains data or control information. The control bit has the value D or K indicating that the information byte contains respectively data or control information.

D-character: A character whose control bit has the value “D”. Also referred to as a data character.

K-character: A character whose control bit has the value “K”. Also referred to as a special character.

Code-group: A 10-bit entity that is the result of 8B/10B encoding a character.

Column: The group of N characters that are transmitted at nominally the same time by a LP-Serial port operating in Nx mode.

Comma: A 7-bit pattern, unique to certain 8B/10B special code-groups, that is used by a receiver to determine code-group boundaries. See more in Section 4.5.7.4, "Sync (/K/)" and Table 4-2.

Idle sequence: The sequence of characters (code-groups after 8B/10B encoding) that is transmitted by a port on each of its active output lanes when the port is not transmitting a packet or control symbol. The idle sequence allows the receiver to maintain bit synchronization, code-group alignment and, if applicable, adaptive equalization settings in between packets and control symbols.

Lane: A single unidirectional signal path, typically a differential pair, between two LP-Serial ports.

Nx port: A LP-Serial port that supports links with up to a maximum of N lanes in each direction.

Lane Alignment: The process of eliminating the skew between the lanes of a LP-Serial link operating in Nx mode such that the characters transmitted as a column by the sender are output by the alignment process of receiver as a column. Without lane alignment, the characters transmitted as a column might be scattered across several columns output by the receiver. The alignment process uses the columns of “A” special characters transmitted as part of the idle sequence.

1x mode: A LP-Serial port mode of operation in which the port transmits on a single lane and receives on a single lane.

Nx mode: A LP-Serial port mode of operation in which the port both transmits and receives on multiple lanes. A LP-Serial port operating in Nx mode transmits on N lanes and receives on N lanes where N has a value greater then 1. The transmit data stream is distributed across the N transmit lanes and the receive data stream is distributed across the N receive lanes.

Striping: The method used on a link operating in Nx mode to distribute data across the N lanes simultaneously. For each direction of the link, the character stream is *striped* across the lanes, on a character-by-character basis, beginning with lane 0, continuing in incrementing lane number order across the lanes, and wrapping back to lane 0 for character N+1.

4.5 8B/10B Transmission Code

The 8B/10B transmission code used by the PCS encodes 9-bit characters (8 bits of information and a control bit) into 10-bit code-groups for transmission and reverses the process on reception. Encodings are defined for 256 data characters and 12 special characters.

The code-groups comprising the 8B/10B code have either an equal number of ones and zeros (balanced) or the number of ones differs from the number of zeros by two (unbalanced). This eases the task of maintaining 0/1 balance. The selection of code-groups also guarantees a minimum of three transitions, 0 to 1 or 1 to 0, within each code-group. For encoding, unbalanced code-groups are grouped in pairs with one member of the pair having more ones than zeros and the other member of the

pair having more zeros than ones. This allows the encoder, when selecting an unbalanced code-group, to select a code-group unbalanced toward ones or unbalanced toward zeros, depending on which is required to maintain the 0/1 balance of the encoder output code-group stream.

The 8B/10B code has the following properties.

- Sufficient bit transition density (3 to 8 transitions per code-group) to allow clock recovery by the receiver.
- Special code-groups that are used for establishing the receiver synchronization to the 10-bit code-group boundaries, delimiting control symbols and maintaining receiver bit and code-group boundary synchronization.
- 0/1 balanced. (can be AC coupled)
- Detection of all single and some multiple-bit errors.

4.5.1 Character and Code-Group Notation

The description of 8B/10B encoding and decoding uses the following notation for characters, code-group and their bits.

The information bits ([0-7]) of an unencoded character are denoted with the letters “A” through “H” where the letter “H” denotes the most significant information bit (RapidIO bit 0) and the letter “A” denotes the least significant information bit (RapidIO bit 7). This is shown in Figure 4-1.

Each data character has a representation of the form Dx.y where x is the decimal value of the least significant 5 information bits EDCBA, and y is the decimal value of the most significant 3 information bits HGF as shown in Figure 4-1. Each special character has a similar representation of the form Kx.y.

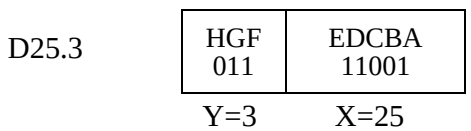


Figure 4-1. Character Notation Example (D25.3)

The output of the 8B/10B encoding process is a 10-bit code-group. The bits of a code-group are denoted with the letters “a” through “j”. The bits of a code-group are all of equal significance, there is no most significant or least significant bit. The ordering of the code-group bits is shown in Figure 4-2.

The code-groups corresponding to the data character Dx.y is denoted by /Dx.y/. The code-groups corresponding to the special character Kx.y is denoted by /Kx.y/.

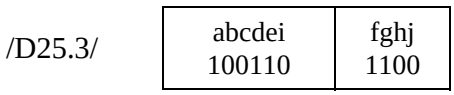


Figure 4-2. Code-Group Notation Example (/D25.3/)

4.5.2 Running Disparity

The 8B/10B encoding and decoding functions use a binary variable called running disparity. The variable can have a value of either positive (RD+) or negative (RD-). The encoder and decoder each have a running disparity variable for each lane which are all independent of each other.

The primary use of running disparity in the encoding process is to keep track of whether the decoder has output more ones or more zeros. The current value of encoder running disparity is used to select the which unbalanced code-group will be used when the encoding for a character requires a choice between two unbalanced code-groups.

Another use of running disparity in the decoding process is to detect errors. Given a value of decoder running disparity, only (256 + 12) = 268 of the 1024 possible code-group values have defined decodings. The remaining 756 possible code-group values have no defined decoding and represent errors, either in that code-group or in an earlier code-group.

4.5.3 Running Disparity Rules

After power-up and before the port is operational, both the transmitter (encoder) and receiver (decoder) must establish current values of running disparity.

The transmitter shall use a negative value as the initial value for the running disparity for each lane.

The receiver may use either a negative or positive initial value of running disparity for each lane.

The following algorithm shall be used for calculating the running disparity for each lane. In the encoder, the algorithm operates on the code-group that has just been generated by the encoder. In the receiver, the algorithm operates on the received code-group that has just been decoded by the decoder.

Each code-group is divided to two sub-blocks as shown in Figure 4-2, where the first six bits (abcdei) form one sub-block (6-bit sub-block) and the second four bits (fghj) form a second sub-block (4-bit sub-block). Running disparity at the beginning of the 6-bit sub-block is the running disparity at the end of the preceding code-group. Running disparity at the beginning of the 4-bit sub-block is the running disparity at

the end of the preceding 6-bit sub-block. Running disparity at the end of the code-group is the running disparity at the end of the 4-bit sub-block.

The sub-block running disparity shall be calculated as follows:

1. The running disparity is positive at the end of any sub-block if the sub-block contains more 1s than 0s. It is also positive at the end of a 4-bit sub-block if the sub-block has the value 0b0011 and at the end of a 6-bit sub-block if the sub-block has the value 0b000111.
2. The running disparity is negative at the end of any sub-block if the sub-block contains more 0s than 1s. It is also negative at the end of a 4-bit sub-block if the sub-block has the value 0b1100 and at the end of a 6-bit sub-block if the sub-block has the value 0b111000.
3. In all other cases, the value of the running disparity at the end of the sub-block is running disparity at the beginning of the sub-block (the running disparity is unchanged).

4.5.4 8B/10B Encoding

The 8B/10B encoding function encodes 9-bit characters into 10-bit code-groups.

The encodings for the 256 data characters (Dx.y) are specified in Table 4-1. The encodings for the 12 special characters (Kx.y) are specified in Table 4-2. Both tables have two columns of encodings, one marked RD- and one marked RD+. When encoding a character, the code-group in the RD- column is selected if the current value of encoder running disparity is negative and the code-group in the RD+ column is selected if the current value of encoder running disparity is positive.

Data characters (Dx.y) shall be encoded according to Table 4-1 and the current value of encoder running disparity. Special characters (Kx.y) shall be encoded according to Table 4-2 and the current value of encoder running disparity. After each character is encoded, the resulting code-group shall be used by the encoder to update the running disparity according to the rules in Section 4.5.3, "Running Disparity Rules".

4.5.5 Transmission Order

The parallel 10-bit code-group output of the encoder shall be serialized and transmitted with bit “a” transmitted first and a bit ordering of “abcdeifghj”. This is shown in Figure 4-3.

Figure 4-3 gives an overview of a character passing through the encoding, serializing, transmission, deserializing, and decoding processes. The left side of the figure shows the transmit process of encoding a character stream using 8B/10B encoding and the 10-bit serialization. The right side shows the reverse process of the receiver deserializing and using 8B/10B decoding on the received code-groups.

The dotted line shows the functional separation between the PCS layer, that provides 10-bit code-groups, and the PMA layer that serializes the code-groups.

The drawing also shows on the receive side the bits of a special character containing the comma pattern that is used by the receiver to establish 10-bit code-boundary synchronization.

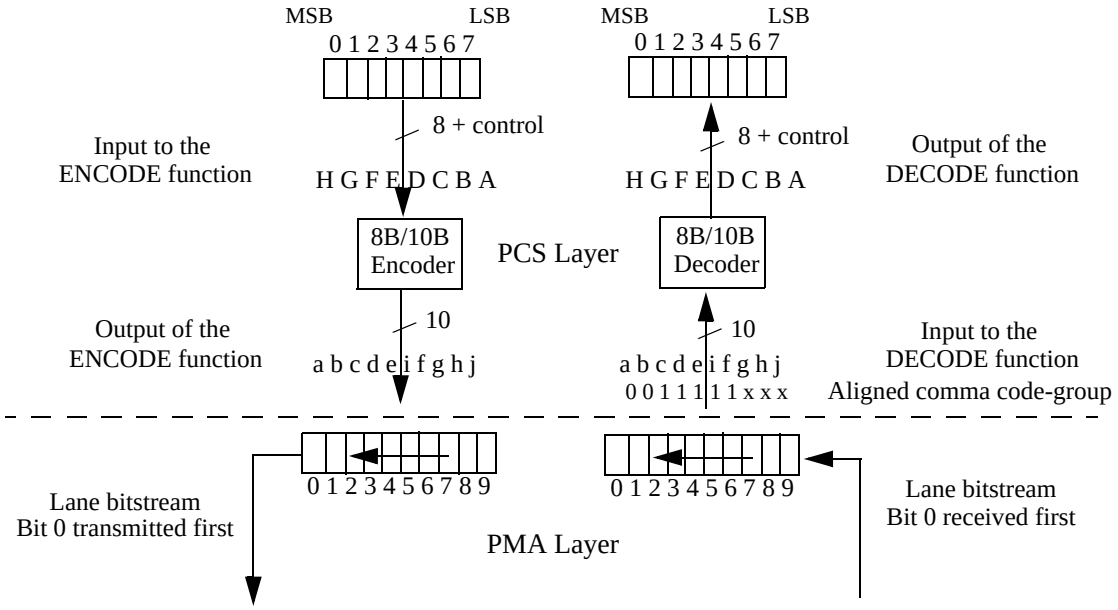


Figure 4-3. Lane Encoding, Serialization, Deserialization, and Decoding Process

4.5.6 8B/10B Decoding

The 8B/10B decoding function decodes received 10-bit code-groups into 9-bit characters and detects and reports received code-groups that have no defined decoding due to one or more transmission errors.

The decoding function uses Table 4-1, Table 4-2 and the current value of the decoder running disparity. To decode a received code-group, the decoder shall select the RD- column of Table 4-1 and Table 4-2 if the current value of the decoder running disparity is negative or shall select the RD+ column if the value is positive. The decoder shall then compare the received code-group with the code-groups in the selected column of both tables. If a match is found in one of the tables, the code-group is defined to be a “valid” code-group and is decoded to the associated character. If no match is found, the code-group is defined to be an “invalid” code-group and is decoded to a character that is flagged in some manner as INVALID. After each code-group is decoded, the decoded code-group shall be used by the decoder to update the decoder running disparity according to the rules in Section 4.5.3, "Running Disparity Rules".

Table 4-1. Data Character Encodings

Character Name	Character Value (hex)	Character Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D0.0	00	000 00000	100111 0100	011000 1011
D1.0	01	000 00001	011101 0100	100010 1011
D2.0	02	000 00010	101101 0100	010010 1011
D3.0	03	000 00011	110001 1011	110001 0100
D4.0	04	000 00100	110101 0100	001010 1011
D5.0	05	000 00101	101001 1011	101001 0100
D6.0	06	000 00110	011001 1011	011001 0100
D7.0	07	000 00111	111000 1011	000111 0100
D8.0	08	000 01000	111001 0100	000110 1011
D9.0	09	000 01001	100101 1011	100101 0100
D10.0	0A	000 01010	010101 1011	010101 0100
D11.0	0B	000 01011	110100 1011	110100 0100
D12.0	0C	000 01100	001101 1011	001101 0100
D13.0	0D	000 01101	101100 1011	101100 0100
D14.0	0E	000 01110	011100 1011	011100 0100
D15.0	0F	000 01111	010111 0100	101000 1011
D16.0	10	000 10000	011011 0100	100100 1011
D17.0	11	000 10001	100011 1011	100011 0100
D18.0	12	000 10010	010011 1011	010011 0100
D19.0	13	000 10011	110010 1011	110010 0100
D20.0	14	000 10100	001011 1011	001011 0100
D21.0	15	000 10101	101010 1011	101010 0100
D22.0	16	000 10110	011010 1011	011010 0100
D23.0	17	000 10111	111010 0100	000101 1011
D24.0	18	000 11000	110011 0100	001100 1011
D25.0	19	000 11001	100110 1011	100110 0100
D26.0	1A	000 11010	010110 1011	010110 0100
D27.0	1B	000 11011	110110 0100	001001 1011
D28.0	1C	000 11100	001110 1011	001110 0100
D29.0	1D	000 11101	101110 0100	010001 1011
D30.0	1E	000 11110	011110 0100	100001 1011
D31.0	1F	000 11111	101011 0100	010100 1011
D0.1	20	001 00000	100111 1001	011000 1001
D1.1	21	001 00001	011101 1001	100010 1001
D2.1	22	001 00010	101101 1001	010010 1001

Table 4-1. Data Character Encodings

Character Name	Character Value (hex)	Character Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D3.1	23	001 00011	110001 1001	110001 1001
D4.1	24	001 00100	110101 1001	001010 1001
D5.1	25	001 00101	101001 1001	101001 1001
D6.1	26	001 00110	011001 1001	011001 1001
D7.1	27	001 00111	111000 1001	000111 1001
D8.1	28	001 01000	111001 1001	000110 1001
D9.1	29	001 01001	100101 1001	100101 1001
D10.1	2A	001 01010	010101 1001	010101 1001
D11.1	2B	001 01011	110100 1001	110100 1001
D12.1	2C	001 01100	001101 1001	001101 1001
D13.1	2D	001 01101	101100 1001	101100 1001
D14.1	2E	001 01110	011100 1001	011100 1001
D15.1	2F	001 01111	010111 1001	101000 1001
D16.1	30	001 10000	011011 1001	100100 1001
D17.1	31	001 10001	100011 1001	100011 1001
D18.1	32	001 10010	010011 1001	010011 1001
D19.1	33	001 10011	110010 1001	110010 1001
D20.1	34	001 10100	001011 1001	001011 1001
D21.1	35	001 10101	101010 1001	101010 1001
D22.1	36	001 10110	011010 1001	011010 1001
D23.1	37	001 10111	111010 1001	000101 1001
D24.1	38	001 11000	110011 1001	001100 1001
D25.1	39	001 11001	100110 1001	100110 1001
D26.1	3A	001 11010	010110 1001	010110 1001
D27.1	3B	001 11011	110110 1001	001001 1001
D28.1	3C	001 11100	001110 1001	001110 1001
D29.1	3D	001 11101	101110 1001	010001 1001
D30.1	3E	001 11110	011110 1001	100001 1001
D31.1	3F	001 11111	101011 1001	010100 1001
D0.2	40	010 00000	100111 0101	011000 0101
D1.2	41	010 00001	011101 0101	100010 0101
D2.2	42	010 00010	101101 0101	010010 0101
D3.2	43	010 00011	110001 0101	110001 0101
D4.2	44	010 00100	110101 0101	001010 0101
D5.2	45	010 00101	101001 0101	101001 0101
D6.2	46	010 00110	011001 0101	011001 0101

Table 4-1. Data Character Encodings

Character Name	Character Value (hex)	Character Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D7.2	47	010 00111	111000 0101	000111 0101
D8.2	48	010 01000	111001 0101	000110 0101
D9.2	49	010 01001	100101 0101	100101 0101
D10.2	4A	010 01010	010101 0101	010101 0101
D11.2	4B	010 01011	110100 0101	110100 0101
D12.2	4C	010 01100	001101 0101	001101 0101
D13.2	4D	010 01101	101100 0101	101100 0101
D14.2	4E	010 01110	011100 0101	011100 0101
D15.2	4F	010 01111	010111 0101	101000 0101
D16.2	50	010 10000	011011 0101	100100 0101
D17.2	51	010 10001	100011 0101	100011 0101
D18.2	52	010 10010	010011 0101	010011 0101
D19.2	53	010 10011	110010 0101	110010 0101
D20.2	54	010 10100	001011 0101	001011 0101
D21.2	55	010 10101	101010 0101	101010 0101
D22.2	56	010 10110	011010 0101	011010 0101
D23.2	57	010 10111	111010 0101	000101 0101
D24.2	58	010 11000	110011 0101	001100 0101
D25.2	59	010 11001	100110 0101	100110 0101
D26.2	5A	010 11010	010110 0101	010110 0101
D27.2	5B	010 11011	110110 0101	001001 0101
D28.2	5C	010 11100	001110 0101	001110 0101
D29.2	5D	010 11101	101110 0101	010001 0101
D30.2	5E	010 11110	011110 0101	100001 0101
D31.2	5F	010 11111	101011 0101	010100 0101
D0.3	60	011 00000	100111 0011	011000 1100
D1.3	61	011 00001	011101 0011	100010 1100
D2.3	62	011 00010	101101 0011	010010 1100
D3.3	63	011 00011	110001 1100	110001 0011
D4.3	64	011 00100	110101 0011	001010 1100
D5.3	65	011 00101	101001 1100	101001 0011
D6.3	66	011 00110	011001 1100	011001 0011
D7.3	67	011 00111	111000 1100	000111 0011
D8.3	68	011 01000	111001 0011	000110 1100
D9.3	69	011 01001	100101 1100	100101 0011
D10.3	6A	011 01010	010101 1100	010101 0011

Table 4-1. Data Character Encodings

Character Name	Character Value (hex)	Character Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D11.3	6B	011 01011	110100 1100	110100 0011
D12.3	6C	011 01100	001101 1100	001101 0011
D13.3	6D	011 01101	101100 1100	101100 0011
D14.3	6E	011 01110	011100 1100	011100 0011
D15.3	6F	011 01111	010111 0011	101000 1100
D16.3	70	011 10000	011011 0011	100100 1100
D17.3	71	011 10001	100011 1100	100011 0011
D18.3	72	011 10010	010011 1100	010011 0011
D19.3	73	011 10011	110010 1100	110010 0011
D20.3	74	011 10100	001011 1100	001011 0011
D21.3	75	011 10101	101010 1100	101010 0011
D22.3	76	011 10110	011010 1100	011010 0011
D23.3	77	011 10111	111010 0011	000101 1100
D24.3	78	011 11000	110011 0011	001100 1100
D25.3	79	011 11001	100110 1100	100110 0011
D26.3	7A	011 11010	010110 1100	010110 0011
D27.3	7B	011 11011	110110 0011	001001 1100
D28.3	7C	011 11100	001110 1100	001110 0011
D29.3	7D	011 11101	101110 0011	010001 1100
D30.3	7E	011 11110	011110 0011	100001 1100
D31.3	7F	011 11111	101011 0011	010100 1100
D0.4	80	100 00000	100111 0010	011000 1101
D1.4	81	100 00001	011101 0010	100010 1101
D2.4	82	100 00010	101101 0010	010010 1101
D3.4	83	100 00011	110001 1101	110001 0010
D4.4	84	100 00100	110101 0010	001010 1101
D5.4	85	100 00101	101001 1101	101001 0010
D6.4	86	100 00110	011001 1101	011001 0010
D7.4	87	100 00111	111000 1101	000111 0010
D8.4	88	100 01000	111001 0010	000110 1101
D9.4	89	100 01001	100101 1101	100101 0010
D10.4	8A	100 01010	010101 1101	010101 0010
D11.4	8B	100 01011	110100 1101	110100 0010
D12.4	8C	100 01100	001101 1101	001101 0010
D13.4	8D	100 01101	101100 1101	101100 0010
D14.4	8E	100 01110	011100 1101	011100 0010

Table 4-1. Data Character Encodings

Character Name	Character Value (hex)	Character Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D15.4	8F	100 01111	010111 0010	101000 1101
D16.4	90	100 10000	011011 0010	100100 1101
D17.4	91	100 10001	100011 1101	100011 0010
D18.4	92	100 10010	010011 1101	010011 0010
D19.4	93	100 10011	110010 1101	110010 0010
D20.4	94	100 10100	001011 1101	001011 0010
D21.4	95	100 10101	101010 1101	101010 0010
D22.4	96	100 10110	011010 1101	011010 0010
D23.4	97	100 10111	111010 0010	000101 1101
D24.4	98	100 11000	110011 0010	001100 1101
D25.4	99	100 11001	100110 1101	100110 0010
D26.4	9A	100 11010	010110 1101	010110 0010
D27.4	9B	100 11011	110110 0010	001001 1101
D28.4	9C	100 11100	001110 1101	001110 0010
D29.4	9D	100 11101	101110 0010	010001 1101
D30.4	9E	100 11110	011110 0010	100001 1101
D31.4	9F	100 11111	101011 0010	010100 1101
D0.5	A0	101 00000	100111 1010	011000 1010
D1.5	A1	101 00001	011101 1010	100010 1010
D2.5	A2	101 00010	101101 1010	010010 1010
D3.5	A3	101 00011	110001 1010	110001 1010
D4.5	A4	101 00100	110101 1010	001010 1010
D5.5	A5	101 00101	101001 1010	101001 1010
D6.5	A6	101 00110	011001 1010	011001 1010
D7.5	A7	101 00111	111000 1010	000111 1010
D8.5	A8	101 01000	111001 1010	000110 1010
D9.5	A9	101 01001	100101 1010	100101 1010
D10.5	AA	101 01010	010101 1010	010101 1010
D11.5	AB	101 01011	110100 1010	110100 1010
D12.5	AC	101 01100	001101 1010	001101 1010
D13.5	AD	101 01101	101100 1010	101100 1010
D14.5	AE	101 01110	011100 1010	011100 1010
D15.5	AF	101 01111	010111 1010	101000 1010
D16.5	B0	101 10000	011011 1010	100100 1010
D17.5	B1	101 10001	100011 1010	100011 1010
D18.5	B2	101 10010	010011 1010	010011 1010

Table 4-1. Data Character Encodings

Character Name	Character Value (hex)	Character Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D19.5	B3	101 10011	110010 1010	110010 1010
D20.5	B4	101 10100	001011 1010	001011 1010
D21.5	B5	101 10101	101010 1010	101010 1010
D22.5	B6	101 10110	011010 1010	011010 1010
D23.5	B7	101 10111	111010 1010	000101 1010
D24.5	B8	101 11000	110011 1010	001100 1010
D25.5	B9	101 11001	100110 1010	100110 1010
D26.5	BA	101 11010	010110 1010	010110 1010
D27.5	BB	101 11011	110110 1010	001001 1010
D28.5	BC	101 11100	001110 1010	001110 1010
D29.5	BD	101 11101	101110 1010	010001 1010
D30.5	BE	101 11110	011110 1010	100001 1010
D31.5	BF	101 11111	101011 1010	010100 1010
D0.6	C0	110 00000	100111 0110	011000 0110
D1.6	C1	110 00001	011101 0110	100010 0110
D2.6	C2	110 00010	101101 0110	010010 0110
D3.6	C3	110 00011	110001 0110	110001 0110
D4.6	C4	110 00100	110101 0110	001010 0110
D5.6	C5	110 00101	101001 0110	101001 0110
D6.6	C6	110 00110	011001 0110	011001 0110
D7.6	C7	110 00111	111000 0110	000111 0110
D8.6	C8	110 01000	111001 0110	000110 0110
D9.6	C9	110 01001	100101 0110	100101 0110
D10.6	CA	110 01010	010101 0110	010101 0110
D11.6	CB	110 01011	110100 0110	110100 0110
D12.6	CC	110 01100	001101 0110	001101 0110
D13.6	CD	110 01101	101100 0110	101100 0110
D14.6	CE	110 01110	011100 0110	011100 0110
D15.6	CF	110 01111	010111 0110	101000 0110
D16.6	D0	110 10000	011011 0110	100100 0110
D17.6	D1	110 10001	100011 0110	100011 0110
D18.6	D2	110 10010	010011 0110	010011 0110
D19.6	D3	110 10011	110010 0110	110010 0110
D20.6	D4	110 10100	001011 0110	001011 0110
D21.6	D5	110 10101	101010 0110	101010 0110
D22.6	D6	110 10110	011010 0110	011010 0110

Table 4-1. Data Character Encodings

Character Name	Character Value (hex)	Character Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D23.6	D7	110 10111	111010 0110	000101 0110
D24.6	D8	110 11000	110011 0110	001100 0110
D25.6	D9	110 11001	100110 0110	100110 0110
D26.6	DA	110 11010	010110 0110	010110 0110
D27.6	DB	110 11011	110110 0110	001001 0110
D28.6	DC	110 11100	001110 0110	001110 0110
D29.6	DD	110 11101	101110 0110	010001 0110
D30.6	DE	110 11110	011110 0110	100001 0110
D31.6	DF	110 11111	101011 0110	010100 0110
D0.7	E0	111 00000	100111 0001	011000 1110
D1.7	E1	111 00001	011101 0001	100010 1110
D2.7	E2	111 00010	101101 0001	010010 1110
D3.7	E3	111 00011	110001 1110	110001 0001
D4.7	E4	111 00100	110101 0001	001010 1110
D5.7	E5	111 00101	101001 1110	101001 0001
D6.7	E6	111 00110	011001 1110	011001 0001
D7.7	E7	111 00111	111000 1110	000111 0001
D8.7	E8	111 01000	111001 0001	000110 1110
D9.7	E9	111 01001	100101 1110	100101 0001
D10.7	EA	111 01010	010101 1110	010101 0001
D11.7	EB	111 01011	110100 1110	110100 1000
D12.7	EC	111 01100	001101 1110	001101 0001
D13.7	ED	111 01101	101100 1110	101100 1000
D14.7	EE	111 01110	011100 1110	011100 1000
D15.7	EF	111 01111	010111 0001	101000 1110
D16.7	F0	111 10000	011011 0001	100100 1110
D17.7	F1	111 10001	100011 0111	100011 0001
D18.7	F2	111 10010	010011 0111	010011 0001
D19.7	F3	111 10011	110010 1110	110010 0001
D20.7	F4	111 10100	001011 0111	001011 0001
D21.7	F5	111 10101	101010 1110	101010 0001
D22.7	F6	111 10110	011010 1110	011010 0001
D23.7	F7	111 10111	111010 0001	000101 1110
D24.7	F8	111 11000	110011 0001	001100 1110
D25.7	F9	111 11001	100110 1110	100110 0001
D26.7	FA	111 11010	010110 1110	010110 0001

Table 4-1. Data Character Encodings

Character Name	Character Value (hex)	Character Bits HGF EDCBA	Current RD –	Current RD +
			abcdei fghj	abcdei fghj
D27.7	FB	111 11011	110110 0001	001001 1110
D28.7	FC	111 11100	001110 1110	001110 0001
D29.7	FD	111 11101	101110 0001	010001 1110
D30.7	FE	111 11110	011110 0001	100001 1110
D31.7	FF	111 11111	101011 0001	010100 1110

Table 4-2. Special Character Encodings

Character Name	Character Value (hex)	Character Bits HGF EDCBA	Current RD –	Current RD +	Notes
			abcdei fghj	abcdei fghj	
K28.0	1C	000 11100	001111 0100	110000 1011	
K28.1	3C	001 11100	001111 1001	110000 0110	2,3
K28.2	5C	010 11100	001111 0101	110000 1010	1
K28.3	7C	011 11100	001111 0011	110000 1100	
K28.4	9C	100 11100	001111 0010	110000 1101	1
K28.5	BC	101 11100	001111 1010	110000 0101	2
K28.6	DC	110 11100	001111 0110	110000 1001	1
K28.7	FC	111 11100	001111 1000	110000 0111	1,2
K23.7	F7	111 10111	111010 1000	000101 0111	1
K27.7	FB	111 11011	110110 1000	001001 0111	
K29.7	FD	111 11101	101110 1000	010001 0111	
K30.7	FE	111 11110	011110 1000	100001 0111	1

Notes
1. Reserved code-group.
2. The code-group contain a comma.
3. A Reserved code-group for Idle Sequence 1

The “comma” is an important element of 8B/10B encoding. A comma is a pattern of 7 bits that is used by receivers to acquire code-group boundary alignment. Two commas patterns are defined, 0b0011111 (comma+) and 0b1100000 (comma-). The pattern occurs in bits **abcdeif** of the special characters K28.1, K28.5 and K28.7. Within the code-group set, it is a singular bit pattern, which, in the absence of transmission errors, cannot appear in any other location of a code-group and cannot be generated across the boundaries of any two adjacent code-groups with the following exception:

The /K28.7/ special code-group when followed by any of the data

code-groups /D3.y/, /D11.y/, /D12.y/, /D19.y/, /D20.y/, /D28.y/, or /K28.y/, where y is an integer in the range 0 through 7, may (depending on the value of running disparity) cause a comma to be generated across the boundary of the two code-groups. A comma that is generated across the boundary between two adjacent code-groups may cause the receiver to change the 10-bit code-group alignment. As a result, the /K28.7/ special code-group may be used for test and diagnostic purposes only.

4.5.7 Special Characters and Columns

Table 4-3 defines the special characters and columns of special characters used by LP-Serial links. Special characters are used for the following functions:

1. Alignment to code-group (10-bit) boundaries on lane-by-lane basis.
2. Alignment of the receive data stream across N lanes.
3. Marking the start of the IDLE2 CS field
4. Clock rate compensation between receiver and transmitter.
5. Control symbol delimiting.

Table 4-3. Special Characters and Columns

Code-Group/Column Designation	Code-Group/Column Use	Number of Code-groups	Encoding
/PD/	Packet_Delimiter Control Symbol	1	/K28.3/
/SC/	Start_of_Control_Symbol	1	/K28.0/
/K/	Sync	1	/K28.5/
/R/	Skip	1	/K29.7/
/A/	Align	1	/K27.7/
/M/	Mark	1	/K28.1/
/**I**/	Idle	1	
||K||	Sync column	N	a column of /K28.5/
||R||	Skip column	N	a coulmn of /K29.7/
||A||	Align column	N	a column of /K27.7/
||M|	Mark column	N	a column of /K28.1/
||**I**||	Idle column	N	a column of Idle

4.5.7.1 Packet Delimiter Control Symbol (/PD/)

PD and /PD/ are aliases for respectively the K28.3 character and the /K28.3/ code-group which are used to delimit a control symbol that contains a packet delimiter.

4.5.7.2 Start of Control Symbol (/SC/)

SC and /SC/ are aliases for respectively the K28.0 character and the /K28.0/ code-group which are used to delimit a control symbol that does not contain a packet delimiter.

4.5.7.3 Idle (/I/)

I and /I/ are aliases for respectively any of the idle sequence characters and idle sequence code-groups.

4.5.7.4 Sync (/K/)

K and /K/ are aliases for respectively the K28.5 character and the /K28.5/ code-group which are used in idle sequences to provide the receiver with the information it requires to achieve and maintain bit and 10-bit code-group boundary synchronization. /K28.5/ was selected as the Sync character as it contains the comma pattern in bits **abcdeif** which is required to locate the code-group boundaries and it provides the maximum number of transitions in bits **ghj**.

4.5.7.5 Skip (/R/)

R and /R/ are aliases for respectively the K29.7 character and the /K29.7/ code-group which are used in the idle sequences and in the clock compensation sequence.

4.5.7.6 Align (/A/)

A and /A/ are aliases for respectively the K27.7 character and the /K27.7/ code-group which are used in idle sequences and for lane alignment on links operating in Nx mode.

4.5.7.7 Mark (/M/)

M and /M/ are aliases for respectively the K28.1 character and the /K28.1/ code-group which are used in Idle Sequence 2 to provide the receiver with the information it requires to achieve and maintain 10-bit code-group boundary synchronization and to mark the location of the Idle frame CS field.

4.5.7.8 Illegal

A special character and its associated code-group that is defined by the 8B/10B code, but not specified for use by the LP-Serial protocol are declared to be an “illegal” character and “illegal” code-group respectively. The special characters K23.7, K28.2, K28.4, K28.6, K28.7 and K30.7 are illegal characters, and if a link is operating with Idle Sequence 1, K28.1 is also an illegal character.

4.5.8 Effect of Single Bit Code-Group Errors

Except in receivers using decision feedback equalization (DFE), single bit code-group errors will be the dominant code-group error by many orders of

magnitude. It is therefore useful to know the variety of code-group corruptions that can be caused by a single bit error.

Table 4-4 lists all possible code-group corruptions that can be caused by a single-bit error. The notation /X/ => /Y/ means that the code-group for the character X has been corrupted by a single-bit error into the code-group for the character Y. If the corruption results in a code-group that is invalid for the current receiver running disparity, the notation /X/ => /INVALID/ is used. The table provides the information required to deterministically detect all isolated single bit transmission errors on links operating with idle sequence 1 and short control symbols.

Table 4-4. Code-Group Corruption Caused by Single Bit Errors

Corruption	Detection on links using idle sequence 1 and short control symbols
/SC/ => /INVALID/	Detectable as an error when decoding the code-group. When this error occurs within a packet, it is indistinguishable from a /Dx.y/ => /INVALID/. When this error occurs outside of a packet, the type of error can be inferred from whether the /INVALID/ is followed by the three /Dx.y/ that comprise the control symbol data.
/PD/ => /INVALID/	Detectable as an error when decoding the code-group. When this error occurs within a packet, it is indistinguishable from a /Dx.y/ => /INVALID/. When this error occurs outside of a packet, the type of error can be inferred from whether the /INVALID/ is followed by the three /Dx.y/ that comprise the control symbol data.
/A/, /K/ or /R/ => /Dx.y/	Detectable as an error as /Dx.y/ is illegal outside of a packet or control symbol and /A/, /K/ and /R/ are illegal within a packet or control symbol.
/A/, /K/ or /R/ => /INVALID/	Detectable as an error when decoding the code-group.
/Dx.y/ => /A/, /K/ or /R/	Detectable as an error as /A/, /K/ and /R/ are illegal within a packet or control symbol and /Dx.y/ is illegal outside of a packet or control symbol.
/Dx.y/ => /INVALID/	Detectable as an error when decoding the code-group.
/Dx.y/ => /Du.v/	Detectable as an error by the packet or control symbol CRC. The error will also result in a subsequent unerrored code-group being decoded as INVALID, but that resulting INVALID code-group may occur an arbitrary number of code-groups after the errored code-group.

4.6 LP-Serial Link Widths

LP-Serial links may have 1, 2, 4, 8, or 16 lanes per direction. All LP-Serial ports shall support operation on links with one lane per direction (1x mode) and may optionally support operation over links with 2, 4, 8 and/or 16 lanes per direction (respectively 2x mode, 4x mode, 8x mode and 16x mode). For example, a port that supports operation over 8 lanes per direction (8x mode) must also support operation over one lane per direction (1x mode) and may optionally also support operation over 2 and/or 4 lanes per direction (2x mode and/or 4x mode). The requirement that all LP-Serial ports support 1x mode is to ensure that any pair of LP-Serial ports that

are capable of operating at the same baud rate also support a common link width over which they can always communicate with each other.

LP-Serial ports that support operation over two or more lanes per direction shall support 1x mode operation over two of those lanes, lane 0 and lane R (the redundancy lane). If the port supports operation over at most two lanes per direction (2x mode), lane R shall be lane 1. If the port supports operation over more than two lanes, lane R shall be lane 2. Requiring ports that support operation over links with two or more lanes per direction to also support 1x mode over two lanes per direction provides a redundant fallback capability that allows communication over the link at reduced bandwidth in the presence of lane failure, regardless of the lane that fails.

4.7 Idle Sequence

An idle sequence is a sequence of characters, code-groups after 8B/10B encoding, that is transmitted by a LP-Serial port on each of its active output lanes when the port is not initialized and, when the port is initialized, there is nothing else to transmit. At a minimum, an idle sequence provides the information required by a LP-Serial receiver to acquire and retain bit, code-group and lane alignment and contains clock compensation sequences to support retimers.

Two idles sequences are defined, Idle Sequence 1, which is referred to as IDLE1, and Idle sequence 2, which is referred to as IDLE2. Both sequences contain the /K/, /A/ and /R/ special code-groups that are required respectively for establishing code-group and lane alignment in the LP-Serial receiver and providing clock compensation for retimers.

IDLE1 was the first idle sequence defined for LP-Serial links and is unchanged from the IDLE specified in Rev. 1.3 of this specification. It is based on and is very similar to the idle sequence used by XAUI, an interconnect that is defined in Clause 47 of IEEE Standard 802.3. IDLE1 was designed for LP-Serial links operating at less than 5.5 GBaud per lane and transmitters and receivers that do not used adaptive equalization. IDLE1 provides only the minimum idle sequence functionality.

IDLE2 was designed for LP-Serial links operating at greater than 5.5 GBaud and transmitters and receivers using adaptive equalization. In addition to the minimum idle sequence functionality, IDLE2 provides link width, lane identification and lane polarity information, randomized data for equalizer training and a command and status channel for receiver control of the transmit equalizer.

When idle is transmitted by a LP-Serial port, an idle sequence shall be transmitted on each of the port's active output lanes. Ports operating in Nx mode shall not stripe the idle sequence across the active lanes; there is an idle sequence for each of the N lanes.

An uninitialized LP-Serial port (state variable port_initialized not asserted) shall continuously transmit an idle sequence on all active output lanes. An initialized

LP-Serial port (state variable port_initialized asserted) shall transmit an idle sequence on each of its active output lanes when there is nothing else to transmit. An idle sequence may not be inserted in a packet or control symbol. An initialized LP-Serial port that becomes uninitialized while transmitting a packet or control symbol may transmit several code-groups per lane of the packet and/or control symbol before beginning the transmission of an idle sequence.

On links operating in 1x mode, the first code-group of the idle sequence shall immediately follow the last code-group of the preceding control symbol. When a link is operating in Nx mode, the first column of N idle code-groups shall immediately follow the column containing the last code-groups of the preceding control symbol.

4.7.1 Clock Compensation Sequence

The “clock compensation sequence” is four character sequence comprised of a K special character immediately followed by three R special characters (K,R,R,R). Clock compensation sequences are transmitted as part of idle sequences.

A port shall transmit a clock compensation sequence on each of its active output lanes at least once every 5000 characters transmitted per lane by the port. When a clock compensation sequence is transmitted, the entire 4 character sequence shall be transmitted. When transmitted by a port operating in Nx mode, the clock compensation sequence shall be transmitted in parallel on all N lanes resulting in the column sequence ||K||R||R||R||.

Since a packet or delimited control symbol may not be interrupted by an idle sequence, it is recommended that a port transmit a clock compensation sequence on each of its active output lanes at least once every 4096 characters transmitted per lane by the port. This requirement implies that the flow of packets and delimited control symbols available from the upper layers can be interrupted long enough to transmit an idle sequence containing a clock compensation sequence.

The compensation sequence allows retimers (discussed in Section 4.11) to compensate for up to a +/- 200 ppm difference between input bit rate and output bit rate. Both rates have a +/-100 ppm tolerance. It may also be used to allow the input side of an end point port to compensate for up to a +/-200 ppm difference between the input bit rate and the bit rate of the device core which may be running off a different clock. This is done by dropping or adding an /R/ immediately following a /K/ in 1x mode or an ||R|| immediately following a ||K|| in Nx mode as needed to avoid overrun or underrun.

4.7.2 Idle Sequence 1 (IDLE1)

Idle Sequence 1 is a sequence of the special characters A, K and R. The sequence is 8B/10B encoded before transmission, yielding a sequence of the special code-groups /A/, /K/ and /R/ that is transmitted on the link.

The IDLE1 sequence shall comply with the following requirements:

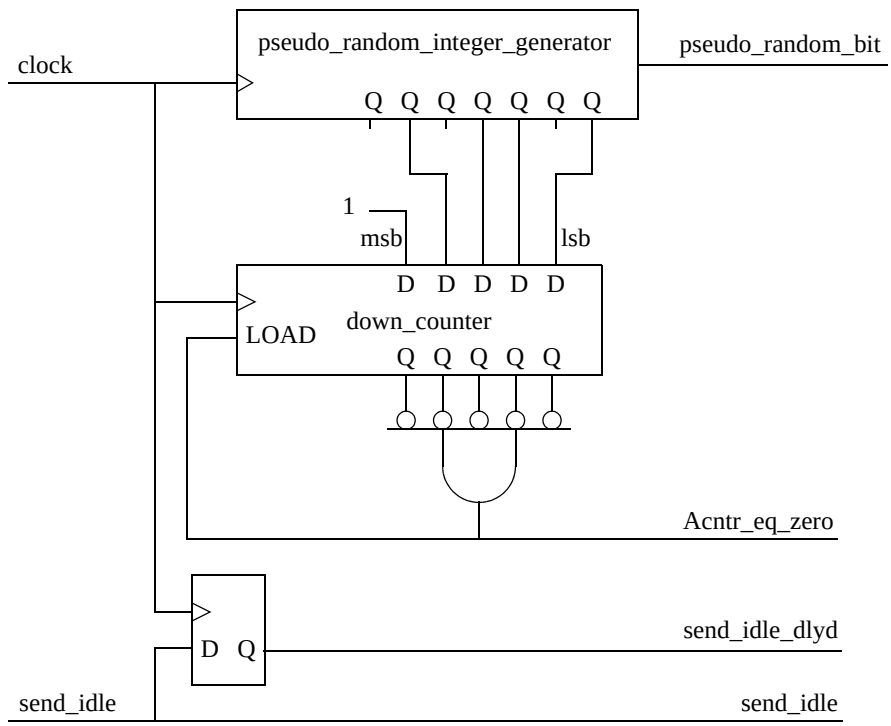
1. Each instance of an IDLE1 sequence shall begin with the K special character.
2. The second, third and fourth characters of each IDLE1 sequence may be the R special character. This allows the first four characters of an IDLE1 sequence to be K,R,R,R, the “clock compensation sequence”.
3. Except when generating the clock compensation sequence, all characters following the first character of an IDLE1 shall be a randomly selected sequence of A, K and R special characters that is based on a pseudo-random sequence generator of 7th degree or greater and subject to minimum and maximum requirements on the spacing of the A special characters. The pseudo-random selection of characters in the idle sequence results in a sequence code-groups whose spectrum has no discrete lines which helps control the EMI of long idle sequences.
4. The number of non-A special characters between A special characters within an IDLE1 sequence shall be no less than 16 and no more than 31. The number shall be pseudo-randomly selected based on a pseudo-random sequence generator of 7th degree of greater. Ideally, the number of non-A characters separating A characters should be uniformly distributed across the range of 16 through 31. However, the IDLE1 spectrum appears to be relatively insensitive to the actual distribution.
5. The requirement on the number of characters between successive A special characters should be maintained between successive IDLE1 sequences to ensure that two successive A special characters are always separated by at least 16 non-A characters.
6. Except when transmitting a clock compensation sequence, an IDLE1 sequence may be of any length and may be terminated after any code-group.
7. Each instance of IDLE1 shall be a new IDLE1 sequence that is unrelated to any previous IDLE1 sequence. Once transmission of an IDLE1 sequence has begun, the sequence may only be terminated. It may not be interrupted or stalled and then continued later.
8. When a port transmitting IDLE1 is operating in Nx mode, the port shall transmit the identical sequence of A, K and R special characters in parallel on each of the N lanes and the N idle sequences shall be aligned across the lanes such that the initial /K/ of the N sequences shall all occur in the same column and the last code-group of the N sequences shall all occur in the same column. As a result, the IDLE1 sequence will appear as a sequence of the columns ||K||, ||R|| and ||A|| at the transmitter output.

4.7.3 Idle Sequence 1 Generation

A primitive polynomial of at least 7th degree is recommended as the generating polynomial for the pseudo-random sequence that is used in the generation of the idle sequence. The polynomials $x^7 + x^6 + 1$ and $x^7 + x^3 + 1$ are examples of primitive 7th

degree polynomials which may be used as generator polynomials. The pseudo-random sequence generator is clocked (generates a new pseudo-random sequence value) once per idle sequence code-group (column). Four of the pseudo-random sequence generator state bits may be selected to generate the pseudo-random value for /A/ spacing. The selection of the state bits and their weighting has a significant effect of the distribution of values for /A/ spacing. Any other state bit or logical function of state bits may be selected as the /K/ vs. /R/ selector.

Figure 4-4 shows an example circuit illustrating how this may be done. The clock ticks whenever a code-group or column is transmitted. Send_idle is asserted whenever an idle sequence begins. The equations indicate the states in which to transmit the indicated idle code-group, except when the compensation sequence is being transmitted. Any equivalent method is acceptable.



send_K = send_idle & (!send_idle_dlyd | send_idle_dlyd & !Acntr_eq_zero & pseudo_random_bit)
send_A = send_idle & send_idle_dlyd & Acntr_eq_zero
send_R = send_idle & send_idle_dlyd & !Acntr_eq_zero & !pseudo_random_bit

Figure 4-4. Example of a Pseudo-Random Idle Code-Group Generator

4.7.4 Idle Sequence 2 (IDLE2)

IDLE 2 is a sequence of data characters and the special characters A, K, M and R. The character sequence is 8B/10B encoded before transmission, yielding a sequence of data code-groups and the special /A/, /K/, /M/ and /R/ code-groups that are transmitted on the link.

The IDLE sequence 2 shall be comprised of a continuous sequence of idle frames and clock compensation sequences. Subject to the following requirements, the exact order of idle frames and clock compensation sequences in an IDLE 2 sequence is implementation dependent.

1. The minimum clock compensation sequence density (clock compensation sequences per characters transmitted per lane) shall comply with the requirements specified in Section 4.7.1, "Clock Compensation Sequence".
2. Each clock compensation sequence shall be followed by an idle frame.
3. Each idle frame shall be followed by either a clock compensation sequence or another idle frame.
4. When a port is operating in Nx mode, the sequence of clock compensation sequences and idle frames shall be the same for all N lanes.

After a port using IDLE2 is initialized (the port initialization state variable port_initialized is asserted), the port may terminate an IDLE2 sequence after any character of an idle frame to transmit a control symbol or a SYNC sequence immediately followed by a link-request control symbol subject to the following requirements:

1. Each M special character transmitted that is part of the idle frame random data field shall be followed by a minimum of four (4) random data field random data characters.
2. The sequence of four (4) M special characters at the beginning of a CS field marker shall not be truncated.
3. A port operating in Nx mode shall terminate an IDLE2 sequence at exactly the same character position in the sequence for each of the N lanes.

Each instance of IDLE2 shall be a new IDLE2 sequence that is unrelated to any previous IDLE2 sequence. Once transmission of an IDLE2 sequence has begun, the sequence may only be terminated. It may not be interrupted or stalled and then continued later.

When a port transmitting IDLE2 is operating in Nx mode, the port shall transmit IDLE2 sequences in parallel on each of the N lanes. The sequences will be similar, but not identical because the information carried in the CS Field Marker will differ from lane to lane and the information carried in the CS Field may also differ from lane to lane. The IDLE2 sequences transmitted on each of the N lanes shall be aligned across the lanes such that the first character of the N idle sequences shall all occur in the same column and the last character of the N idle sequences shall all

occur in the same column. As a result, the IDLE2 sequence will appear at the transmitter output as a sequence of the columns ||K||, ||R||, ||M|| and ||A|| and columns containing only data code-groups.

4.7.4.1 Idle Frame

Each idle frame shall be composed of three parts, a random data field, a command and status (CS) field marker and an encoded CS field as shown in Figure 4-5.

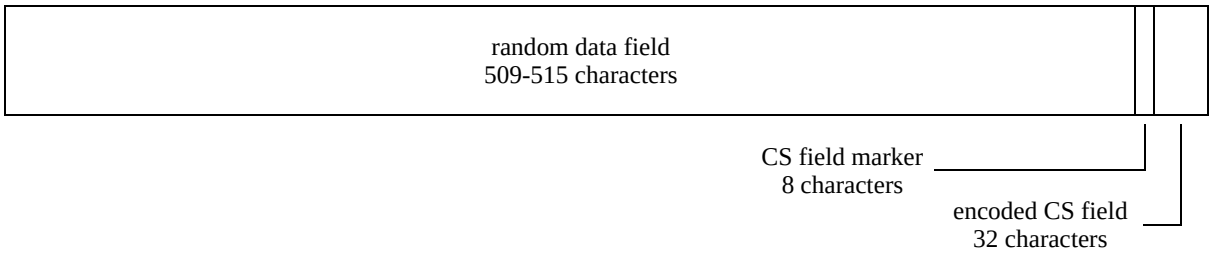


Figure 4-5. Idle Sequence 2 Idle Frame

4.7.4.1.1 IDLE Sequence 2 Random Data Field

The IDLE2 random data field shall contain pseudo-random data characters and the A and M special characters. The total length of the random data field shall be no less than 509 and no more than 515 characters. The idle field shall comply with the following requirements.

1. Unless otherwise specified, the characters comprising the random data field shall be pseudo-random data characters.
2. The random data field of an idle frame that immediately follows a clock compensation sequence shall begin with a M special character. Otherwise, the random data field of an idle frame shall begin with a pseudo-random data character.
3. Unless otherwise specified, the pseudo-random data characters in the random data field shall occur in contiguous sequences of not less than 16 and no more than 31 pseudo-random characters. The length of each contiguous sequence shall be pseudo-randomly selected. The lengths of the contiguous sequences should be uniformly distributed across the range of 16 to 31 characters. Adjacent contiguous sequences shall be separated by a single A or M special character. Each separator shall be pseudo-randomly selected. The probability of selecting the A or M special character for a given separator should be equal. The last four (4) characters of the random data field shall be pseudo-random data characters. The length of the first contiguous sequence of pseudo-random characters in the random data field shall be no less than 16 and no more than 35 characters. The length of the last contiguous sequence of pseudo-random characters in the random data field shall be no less than 4 and no more than 35 characters.

4. Each random data field that is transmitted on a given lane of a link shall be generated by first generating a prototype random data field using the above rules, but with a D0.0 character in the place of each pseudo-random data character, and then scrambling the prototype random data field with the transmit scrambler for that lane. The scrambling shall be done in exactly the same manner as packet and control symbol data characters are scrambled. The scrambler, the scrambling method and the scrambling rules are specified in Section 4.8.1, "Scrambling Rules".
5. When a port is operating in Nx mode, the location A or M special characters in a random data field shall be identical for all N lanes. If the k^{th} character of a random data field transmitted on lane 0 is an A (M) special character, the k^{th} character of the random data fields transmitted on lanes 1 through N-1 is also an A (M) special character.

Generating the random data field pseudo-random data characters by scrambling D0.0 characters results in the output serial random data bit stream being the scrambling sequence. This allows the receiver to recover the descrambler seed from the received idle frame random data field. It also allows the receiver to verify that the lane descramblers are synchronized to the incoming data stream. If a lane descrambler is correctly synchronized, the pseudo-random data characters in the idle frame random data field will all descramble to D0.0 characters.

4.7.4.1.2 IDLE Sequence 2 CS Field Marker

The CS field marker indicates the beginning of the command and status (CS) field and provides information about the link polarity, link width and lane numbering.

The CS field marker shall be the 8 character sequence

M, M, M, M, D21.5, Dx.y, D21.5, $\overline{\text{Dx.y}}$

where

x, the least significant 5 bits of Dx.y, encodes lane_number[0-4], the number of the lane within the port,

y, the most significant 3 bits of Dx.y, encodes active_link_width[0-2], the active width of the port and

$\overline{\text{Dx.y}}$ is the bit wise complement of Dx.y.

As shown above, the CS frame marker characters shall be transmitted from left to right. The first character transmitted is M, the last character transmitted is $\overline{\text{Dx.y}}$.

The “M, M, M, M” sequence that begins the CS field marker is unique and is used to locate the start of the CS data field. The sequence occurs only between the Idle Sequence 2 idle frame random data and CS fields. It never occurs in control symbols or packet data and can not be created by an isolated burst error of 11 bits or less at the code-group level.

The character D21.5 provides lane polarity indication. The 8B/10B encoding of D21.5 is independent of running disparity. If the lane polarity is inverted, the character will decode as D10.2.

The active_port_width field shall be encoded as specified in Table 4-5.

Table 4-5. Active Port Width Field Encodings

y	active_link_width[0-2]	Link mode	Notes
0	0b000	1x	
1	0b001	2x	
2	0b010	4x	
3	0b011	8x	
4	0b100	16x	
5	0b101	1x on lanes 0, 1 and 2	3
6	0b110	1x on both lanes 0 and 1	1
7	0b111	1x on both lanes 0 and 2	2

Notes
1. Used when a 2x port is operating in 1x mode.
2. Used when a 4x, 8x, or 16x port is operating in 1x mode.
3. Used when a 1x/2x/Nx port is operating in 1x mode. Some early implementations may report this mode as an active_link_width of 0b110 or 0b111 instead of 0b101.

The lane_number field shall be encoded as specified in Table 4-6.

Table 4-6. Lane Number Field Encodings

x	lane_number[0-4]	lane number
0	0b00000	0
1	0b00001	1
2	0b00010	2
3	0b00011	3
4	0b00100	4
5	0b00101	5
6	0b00110	6
7	0b00111	7
8	0b01000	8
9	0b01001	9
10	0b01010	10
11	0b01011	11
12	0b01100	12
13	0b01101	13
14	0b01110	14
15	0b01111	15
16-31	0b10000 - 0b11111	Reserved

A CS field marker whose first four characters are not all M special characters, fifth and seventh characters are not both D21.5 or D10.2 or sixth and eight character are not the bit wise complements of each other shall be determined to be corrupted. A received CS field marker that is determined to be truncated and/or corrupted shall be ignored and discarded. Any error detected in a truncated and/or corrupted CS field marker that is determined to be the result of a transmission error and not the result of truncation, such as an “invalid” or “illegal” character, shall be reported as an input error.

4.7.4.1.3 IDLE2 Command and Status Field (CS field)

The CS field allows a port to provide certain status information about itself to the connected port and to control the transmit emphasis settings of the connected port if the connected port supports adaptive transmit emphasis.

The CS field shall have 32 information bits, cs_field[0-31], and 32 check bits, cs_field[32-63]. The check bits cs_field[32-63] shall be the bit wise complement of the information bits cs_field[0-31] respectively.

The CS field bits are defined in Table 4-7.

Table 4-7. Command and Status Field Encodings

CS_field bit(s)	Definition
0	CMD - Command This bit indicates to the connected port when an emphasis update command is present 0b0 - no request present 0b1 - request present
1	Implementation defined
2	Receiver trained When the lane receiver controls any transmit or receive adaptive equalization, this bit indicates whether or not all adaptive equalizers controlled by the lane receiver are trained 0b0 - One or more adaptive equalizers are controlled by the lane receiver and at least one of those adaptive equalizers is not trained 0b1 - The lane receiver controls no adaptive equalizers or all of the adaptive equalizers controlled by the receiver are trained
3	Data scrambling/descrambling enabled This bit indicates whether control symbol and packet data characters are being scrambled before transmission and descrambled upon reception This bit indicates whether or not the transmitter is scrambling control symbol and packet data characters. 0b0: scrambling/descrambling disabled 0b1: scrambling/descrambling enabled
4-5	Tap(-1) status - Transmit emphasis tap(-1) status These bits indicate the status of transmit emphasis tap(-1). 0b00: not implemented 0b01: at minimum emphasis 0b10: at maximum emphasis 0b11: at intermediate emphasis setting

Table 4-7. Command and Status Field Encodings

CS_field bit(s)	Definition
6-7	Tap(+1) status - Transmit emphasis tap(+1) status. These bits indicate the status of transmit emphasis tap(+1). 0b00: not implemented 0b01: at minimum emphasis 0b10: at maximum emphasis 0b11: at intermediate emphasis setting
8-23	Reserved
24-25	Tap(-1) Command - Transmit emphasis tap(-1) update command This bit is used in conjunction with the “CMD” bit to change or retain the emphasis setting of tap(-1). 0b00: hold 0b01: decrease emphasis by one step 0b10: increase emphasis by one step 0b11: reserved
26-27	Tap(+1) Command - Transmit emphasis tap(+1) update command This bit is used in conjunction with the “CMD” bit to change or retain the emphasis setting of tap(+1). 0b00: hold 0b01: decrease emphasis by one step 0b10: increase emphasis by one step 0b11: reserved
28	Reset emphasis This bit is used in conjunction with the “CMD” bit to force the transmit emphasis settings in the connected transmitter to no emphasis 0b0: Ignore 0b1: Reset all transmit emphasis taps to no emphasis
29	Preset emphasis This bit is used in conjunction with the “CMD” bit to force the transmit emphasis settings in the connected transmitter to initial or preset values 0b0: Ignore 0b1: Set all transmit emphasis setting to their preset values.
30	ACK This bit indicates when a transmit emphasis update command from the connected port has been accepted. 0b0: command not accepted 0b1: command accepted
31	NACK This bit indicates when a transmit emphasis update command from the connected port has been refused. 0b0: command not refused 0b1: command refused

The 64 cs_field bits shall be encoded in pairs as specified in Table 4-8.

Table 4-8. CS Field 8/10 Bit Encodings

CS_field[n,n+1] n even	Encoding
0,0	D7.3
0,1	D24.3

Table 4-8. CS Field 8/10 Bit Encodings

CS_field[n,n+1] n even	Encoding
1,0	D30.3
1,1	D24.7

This encoding has the property that after 8B/10B encoding, the resulting transmit signal has a minimum run length of 2 except at the boundary between code-groups when a /D24.7/ is immediately followed by a /D30.3/. The minimum run length of 2 reduces the effective bandwidth of the transmitted signal which improves the reliability of transmission over an unequalized or partially equalized lane.

The characters encoding the CS channel shall be transmitted in the order of the bits they encode beginning with the character encoding CS field bits [0,1] and ending with the character encoding bits [62-63].

A CS field whose bits [32-63] are not the bit wise complement of bits [0-31] respectively shall be determined to be corrupted. A received CS field that is determined to be truncated and/or corrupted shall be ignored and discarded. Any error detected in a truncated and/or corrupted CS field that is determined to be the result of a transmission error and not the result of truncation, such as an “invalid” or “illegal” character, shall be reported as an input error.

4.7.4.1.4 IDLE2 CS Field Use

The transmit emphasis status and update commands supported by the CS Field are based on a reference model for the transmitter emphasis network that is a transversal filter with K taps with baud period tap spacing. A 5-tap transversal filter is shown in Figure 4-6. The filter taps are named according to their position relative to the “main” tap which is designated tap(0). As the signal propagates through the filter, taps that are reached by the signal before it reaches the main tap are designated with negative integers. Taps that are reached by the signal after it has passed the main tap are designated with positive integers. For example, the tap immediately before the main tap is designated tap(-1), the tap immediately following the main tap is designated tap(+1) and the second tap after the main tap is designated tap(+2). The output signal of a transversal filter is formed by multiplying the voltage of each tap by a tap coefficient and summing the products together. The coefficient for tap(n) is designated k_n. The main tap, tap(0), has the property that its coefficient (k_0) is always positive. When all emphasis is disabled, the main tap coefficient is 1 and all of the other tap coefficients are 0.

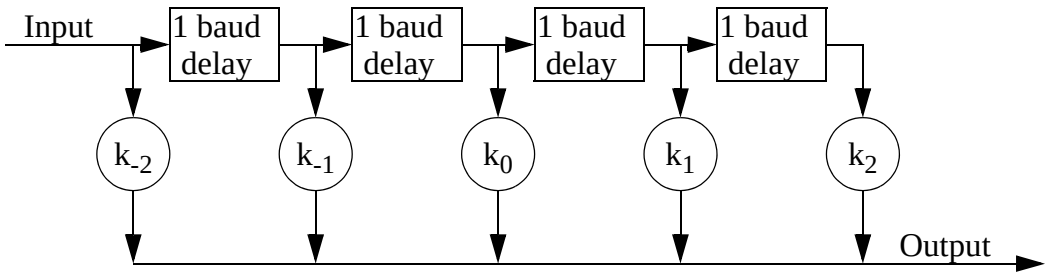


Figure 4-6. 5-tap Transversal Filter

The structure of the transmit emphasis transversal filter in a given port is conveyed to the connected port by the Tap(n) status fields in the CS fields transmitted by the port.

The intended use for transmit emphasis is to allow at least partial compensation for the transmission losses of links implemented with differential printed circuit board (PCB) trace pairs which increase with increasing frequency. Compensation is achieved by emphasizing the higher frequency portion of the transmit spectrum before transmission. A transversal filter for this purpose typically has two or three taps. The two tap filter has a main tap, tap(0), and either a tap(-1) or a tap(+1). The three tap filter has a main tap and both a tap(-1) and a tap(+1). When adjusted for transmit emphasis, the coefficients of tap(-1) and tap(+1) will be negative with emphasis increasing as the coefficients become more negative.

The CS fields exchanged between connected LP-Serial ports provides a command and acknowledgement path that allows a LP-Serial receiver to control the transmit emphasis of the connected transmitter. The issuing and acknowledgement of transmit emphasis commands is control by a handshake that uses the CS field signals CMD, ACK and NACK.

A receiver may issue the following commands. Only one of these commands may be issued at a time.

reset emphasis

preset emphasis

modify the emphasis provided by tap(-1), if tap(-1) is implemented

modify the emphasis of tap(+1), if tap(+1) is implemented

CS field commands shall be issued and acknowledged using the following rules. References to specific command bits and to the CMD bit refer to the specific command bits and the CMD bit in CS fields transmitted by the port issuing the command. References to the ACK and NACK bits refer to the ACK and NACK bits

in CS fields received from the connected port. An example of this handshake is shown in Figure 4-7.

Specific command bits may be changed only when the ACK and NACK bits are both de-asserted and the CMD bit is either de-asserted or transitioning from de-asserted to asserted.

Once the CMD bit is asserted, the connected port will either assert ACK after accepting and executing the command or assert NACK if the command cannot be executed. The assertion of ACK or NACK shall occur no more than 250usec after the assertion of CMD. ACK and NACK shall never be asserted at the same time.

Once ACK or NACK is asserted in a CS field received by the port issuing the command, the CMD bit is de-asserted.

ACK or NACK, whichever is asserted, shall be de-asserted within 250usec of receipt of a CS field with the CMD bit deasserted.

If, for any reason, the connected port fails to assert ACK or NACK within 250usec of the assertion of CMD, CMD may be deasserted. Once deasserted, CMD shall remain deasserted for at least 250usec before being reasserted.

A CS field command to increase the emphasis of tap(n) by one step shall cause the tap(n) coefficient to be made more negative by one step. A command to decrease the emphasis of tap(n) by one step shall cause the tap(n) coefficient to be made more positive by one step. The transmit emphasis step sizes are implementation dependant. The adjustment of the tap(n) coefficient value may result in the coefficient value of one or more of the other taps to be modified by the transmitter to maintain certain specifications such as the minimum or maximum transmit amplitude.

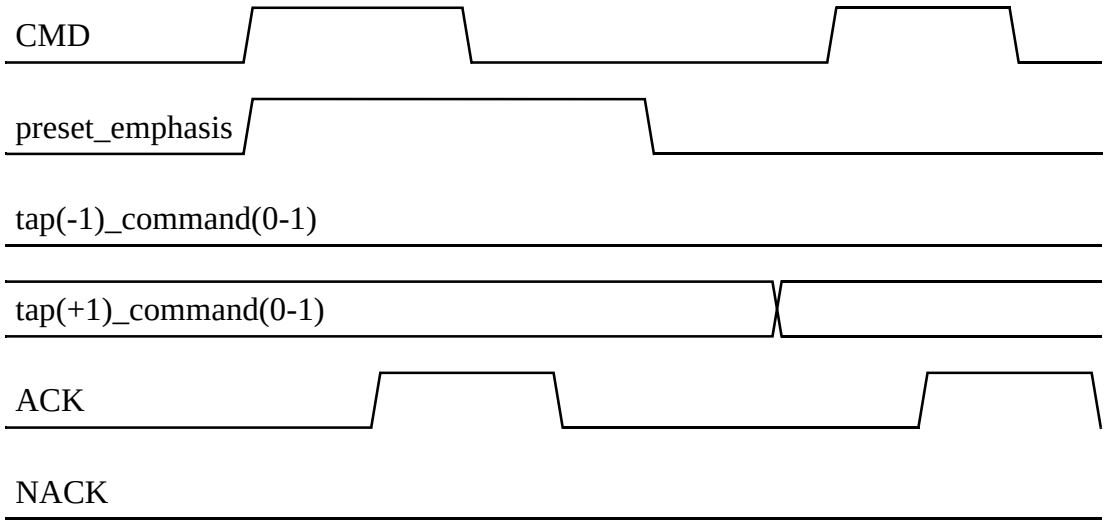


Figure 4-7. Example of CS Field CMD, ACK, NACK Handshake

4.7.5 Idle Sequence Selection

LP-Serial links operating at greater than 5.5 GBaud per lane shall always use the IDLE2 sequence. LP-Serial links operating a less than 5.5 GBaud per lane shall support use of the IDLE1 sequence and may support use of the IDLE2 sequence.

If a LP-Serial port is operating at less than 5.5 GBaud per lane and both ports on the link support both IDLE1 and IDLE2, the port shall determine which idle sequence to use on the link by using the following algorithm during the port initialization process.

> If a LP-Serial port is operating at less than 5.5 GBaud per lane, supports the IDLE2 sequence and its configuration allows it to use the IDLE2 sequence, the port shall transmit the IDLE2 sequence when it enters the SEEK state of the port initialization process. (The port initialization process is specified in Section 4.12.) Otherwise, a LP-Serial port operating at less than 5.5 GBaud per lane shall transmit the IDLE1 sequence when entering the SEEK state and shall use the IDLE1 on the link until the port reenters the SEEK state.
>
> A LP-Serial port transmitting the IDLE2 sequence shall monitor the idle sequence it is receiving from the connected port. The port shall determine the idle sequence being received from the connected port using a lane for which lane_sync is asserted. The techniques and algorithms used by a port supporting both IDLE1 and IDLE2 to determine which idle sequence it is receiving are implementation specific and outside the scope of this specification.

If the LP-Serial port that is transmitting the IDLE2 sequence receives IDLE2 from the connected port, IDLE2 shall be the idle sequence used on the link until the port reenters the SEEK state. If the port receives IDLE1 from the connected port, the port shall switch to transmitting IDLE1 and IDLE1 shall be the idle sequence used on the link until the port reenters the SEEK state.

There are restrictions on the type of equalizers and, if any of the equalization is adaptive, on the adaptive equalizer training algorithms that may be used by ports operating at less than 5.5 GBaud. These restrictions are specified in Section 9.2, "Equalization" and in Section 10.2, "Equalization".

4.8 Scrambling

Scrambling smooths the spectrum of a port's transmit signal and reduces the spectrum's peak values. This is most important when long strings of the same character or of a repeating character sequence are transmitted. The result is a reduction in the amount of electromagnetic interference (EMI) generated by the link and easier design of adaptive equalizer training algorithms. Scrambling of packet and control symbol data characters is used only on links operating with idle sequence 2 (IDLE2). It is not used on links operating with idle sequence 1 (IDLE1) for backwards compatibility with early revisions of this specification.

4.8.1 Scrambling Rules

The use of control symbol and packet data character scrambling on a LP-Serial link is determined by the idle sequence being used on the link.

If the idle sequence selection process specified in Section 4.7.5 has selected idle sequence 1 (IDLE1) for use on the link, no characters shall be scrambled before transmission on the link.

If the idle sequence selection process has selected idle sequence 2 (IDLE2), control symbol and packet data characters shall be scrambled by the transmitter before transmission on the link and descrambled in the receiver upon reception. (The per lane scramblers are also used to generate the pseudo-random data characters in the IDLE2 random data field as specified in Section 4.7.4.1.1). Special characters, CS field marker data characters, and CS field data characters shall not be scrambled before transmission.

Scrambling and descrambling of control symbol and packet data characters can be disabled for test purposes by setting the Data scrambling disable bit in the Port *n* Control 2 CSR. Scrambling and descrambling of control symbol and packet data characters shall not be disabled for normal link operation. Setting the Data scrambling disable bit does not disable the use of the lane scramblers for the

generation of pseudo-random data characters for the IDLE2 random data field. (See Section 6.6.10, "Port n Control 2 CSRs (Block Offset 0x54, 74, ... , 234)").

Scrambling and descrambling shall be done at the lane level. Nx ports shall have a transmit scrambling and receive descrambling function for each of the N lanes. In the transmitter, scrambling shall occur before 8B/10B encoding, and if the port is operating in Nx mode, after lane striping. In the receiver, descrambling shall occur after 8B/10B decoding, and if the port is operating in Nx mode, before lane destriping.

The polynomial $x^{17}+x^8+1$ shall be used to generate the pseudo-random sequences that are used for scrambling and descrambling. This polynomial is not primitive, but when the sequence generator is initialized to all 1s or other appropriate values, the polynomial produces a sequence with a repeat length of 35,805 bits. The bit serial output of the pseudo-random sequence generator shall be taken from the output of the register holding x^{17}. The pseudo random sequence generator is shown in Figure 4-8.

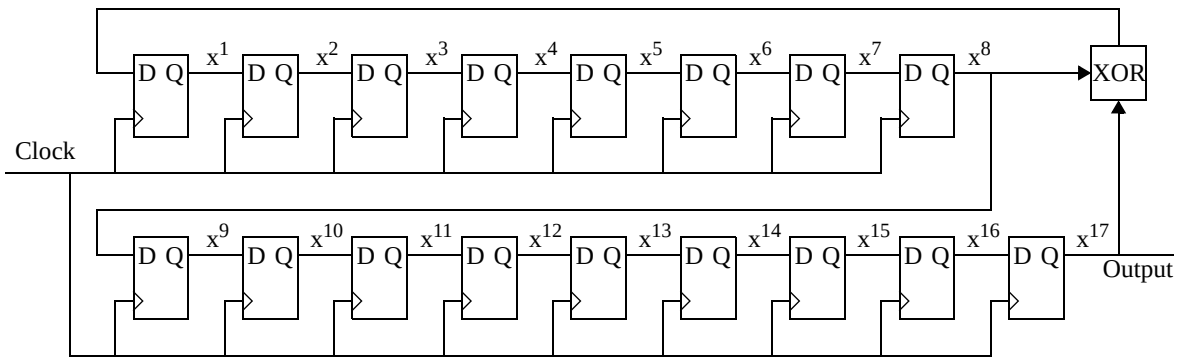


Figure 4-8. Scrambling Sequence Generator

Control symbol and packet data characters shall be scrambled and descrambled by XORing the bits of each character with the output of the pseudo-random sequence generator. The bits of each data character are scrambled/descrambled in order of decreasing significance. The most significant bit (bit 0) is scrambled/descrambled first, the least significant bit (bit 7) is scrambled/descrambled last.

The transmitter and receiver scrambling sequence generators shall step during all characters except R special characters. This is to prevent loss of sync between transmit and receive scramblers when an /R/ or ||R|| is added or removed by a retimer.

To minimize any correlation between lanes when a port is transmitting on multiple lanes, the scrambling sequence applied to a given output lane of the port shall be offset from the scrambling sequence applied to any other output lane of the port by at least 64 bits. If separate scrambling sequence generators are used for each lane, the offset requirement can be achieved by initializing the scramblers to the values specified in Table 4-9, which provide an offset of 64.

Table 4-9. Scrambler Initialization Values

Lane	Initialization value [x^1-x^{17}]
0	0b1111 1111 1111 1111 1
1	0b1111 1111 0000 0110 1
2	0b0000 0000 1000 0110 1
3	0b0000 0110 0111 1010 0
4	0b1000 0000 1011 1001 0
5	0b1111 1010 1000 0111 0
6	0b0100 0011 1001 1011 1
7	0b1100 0100 1010 0101 0
8	0b0101 1111 0100 1001 0
9	0b1111 1010 0111 1001 1
10	0b1011 0011 0111 0010 1
11	0b1100 1010 1011 0011 0
12	0b1011 1000 0101 0011 1
13	0b0000 1011 0110 1111 0
14	0b0101 1000 1001 1010 1
15	0b0011 0111 1010 1000 1

4.8.2 Descrambler Synchronization

Since the pseudo-random data characters of the random data field of the idle sequence 2 idle frame are generated by scrambling D0.0 characters, the pseudo-random characters of the random data field contain the pseudo-random sequence used by the transmitter to scramble control symbol and packet data characters. A sequence of at least four (4) contiguous pseudo-random data characters immediately follow each M special character in the random data field.

Each lane descrambler shall synchronize itself to the scrambled data stream it is receiving by using the scrambling sequence extracted from the pseudo-random data characters received by the lane to re-initialize the state of the descrambler.

After a lane descrambler has been re-initialized, the next two descrambler sync tests, which are defined in Section 4.8.3, shall be used to verify descrambler synchronization. If the result of both lane descrambler sync tests is “pass”, the descrambler shall be determined to be “in sync”. Otherwise, the lane descrambler shall be determined to be “out of sync” and the resynchronization process shall be repeated.

To ensure that a port that may have lost descrambler sync is able to recover descrambler sync before it is sent a link maintenance protocol link-request control symbol, a LP-Serial port that is operating with IDLE2 shall transmit a SYNC sequence (described below) before transmitting any link-request control symbol.

The SYNC sequence shall be transmitted in parallel on each of the N active lanes of a link operating in Nx mode and shall immediately precede the link-request control symbol. If the link is operating in 1x mode, the last character of the SYNC sequence is immediately followed by the first character of the link-request. If the link is operating in Nx mode, the last column of the SYNC sequence is immediately followed by the column containing the first characters of the link-request.

The SYNC sequence shall be comprised of four contiguous repetitions of a five character sequence that begins with a M special character immediately followed by 4 pseudo-random data characters, i.e. the SYNC sequence is MDDDD MDDDD MDDDD MDDDD. The pseudo-random data characters shall be generated in the same way as the pseudo-random data characters in the random data field of the IDLE2 idle frame are generated. The SYNC sequence will appear as four repetitions of ||M||D||D||D||D|| on a link operating in Nx mode.

4.8.3 Descrambler Synchronization Verification

Each active lane of a LP-Serial port that is descrambling received control symbol and packet data characters shall, with the one exception stated below, perform a descrambler synchronization state check (descrambler sync check) whenever a descrambler sync check trigger event is detected in the received character stream of the lane.

A descrambler sync check trigger event is defined as the occurrence of one of the following character sequences in the received character stream of an active lane.

1. A single K, M or R special character that is not part of a contiguous sequence of K, M and/or R special characters.
2. A contiguous sequence of K and/or R special characters possibly followed by a M special character.

The descrambler sync check shall consist of inspecting the descrambled values of the four contiguous characters following the trigger sequence. These four characters are designated the descrambler sync “check field”. The characters comprising the check field shall be determined as follows.

The check field for the first type of trigger event shall be the four characters immediately following the K, M or R special character.

The check field for the second type of trigger event that does not end with a M special character shall be the four characters immediately following the contiguous sequence of K and/or R special characters.

The check field for the second type of trigger event that ends with a M special character shall be the four characters immediately following the M special character.

When the descrambler is in sync and in the absence of transmission errors, the "check field" will contain four data characters that are all D0.0s after descrambling.

The exception to the rule stated above that each descrambler sync check trigger sequence shall cause the receiving lane to execute a descrambler sync check is when the descrambler check trigger sequence begins in the four character check field of a previous trigger sequence. When this occurs, the trigger sequence shall not trigger a descrambler sync check. For example, the RM in the sequence KRXRMDDDD, where X is neither a K nor R, shall not trigger a descrambler sync check as it begins in the four character check field used by the descrambler sync check triggered by the KR sequence.

If the descrambled value of each of the four characters in a check field is D0.0, the result of the descrambler sync test shall be "pass". Otherwise, the result of the descrambler sync test shall be "fail" and the descrambler shall be determined to be "out of sync".

A sync test can fail because of either a loss of descrambler sync or a data transmission error(s) in either the sync trigger sequence or the check field. While this will result is some false sync test "failures", it is preferable to allowing false sync test "passes" which can result in undetected control symbol or packet corruption.

If a descrambler sync test fails, the port shall immediately enter the Input Error-stopped state if it is not already in that state and resynchronize the descrambler. All control symbols and packet received while a lane descrambler is out of sync shall be ignored and discarded. The cause field in the packet-not-accepted control symbol issued by the port on entering the Input Error-stopped state due to a sync check failure shall indicate "loss of descrambler sync".

4.9 1x Mode Transmission Rules

4.9.1 1x Ports

A 1x LP-Serial port shall 8B/10B encode and transmit the character stream of delimited control symbols and packets received from the upper layers in the order the characters were received from the upper layers. When neither control symbols nor packets are available from the upper layers for transmission, an idle sequence shall be fed to the input of the 8B/10B encoder for encoding and transmission.

On reception, the code-group stream is 8B/10B decoded and the resulting character stream of error free delimited control symbols and packets shall be passed to the upper layers in the order the characters were received from the link.

If the link is operating with idle sequence 2, control symbol and packet data characters shall be scrambled before transmission and descrambled after reception as specified in Section 4.8.

Figure 4-9 shows the encoding and transmission order for a short control symbol transmitted over a LP-Serial link operating in 1x mode.

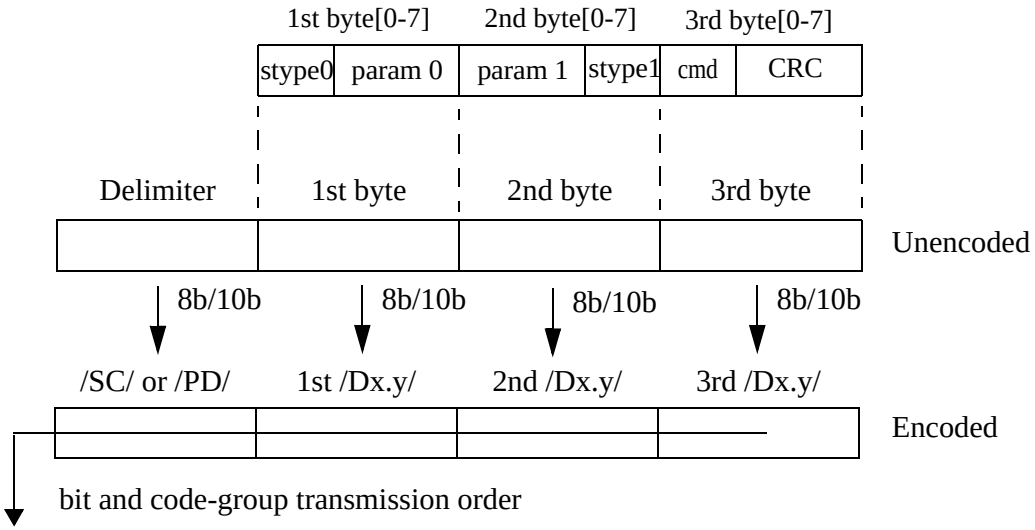


Figure 4-9. 1x Mode Short Control Symbol Encoding and Transmission Order

Figure 4-10 shows the encoding and transmission order for a packet transmitted over a 1x LP-Serial link.

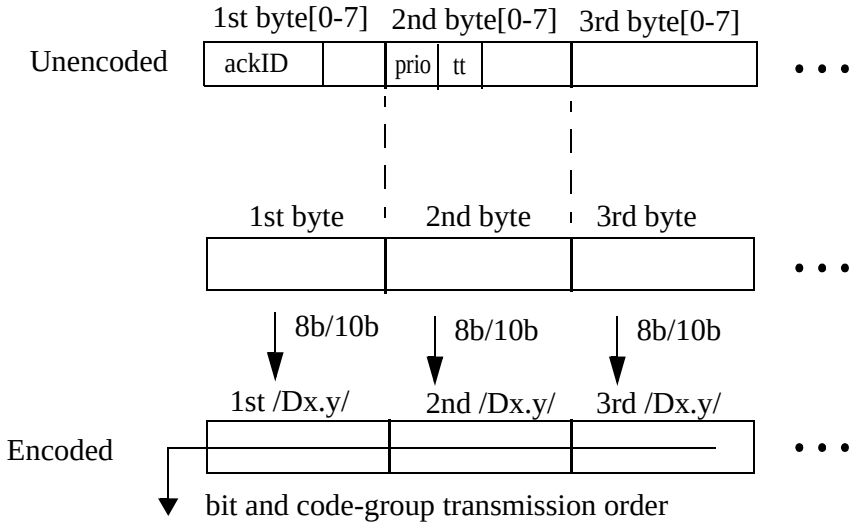


Figure 4-10. 1x Mode Packet Encoding and Transmission Order

Figure 4-11 shows an example of idle sequence 1, short control symbol and packet transmission on a 1x LP-Serial link.

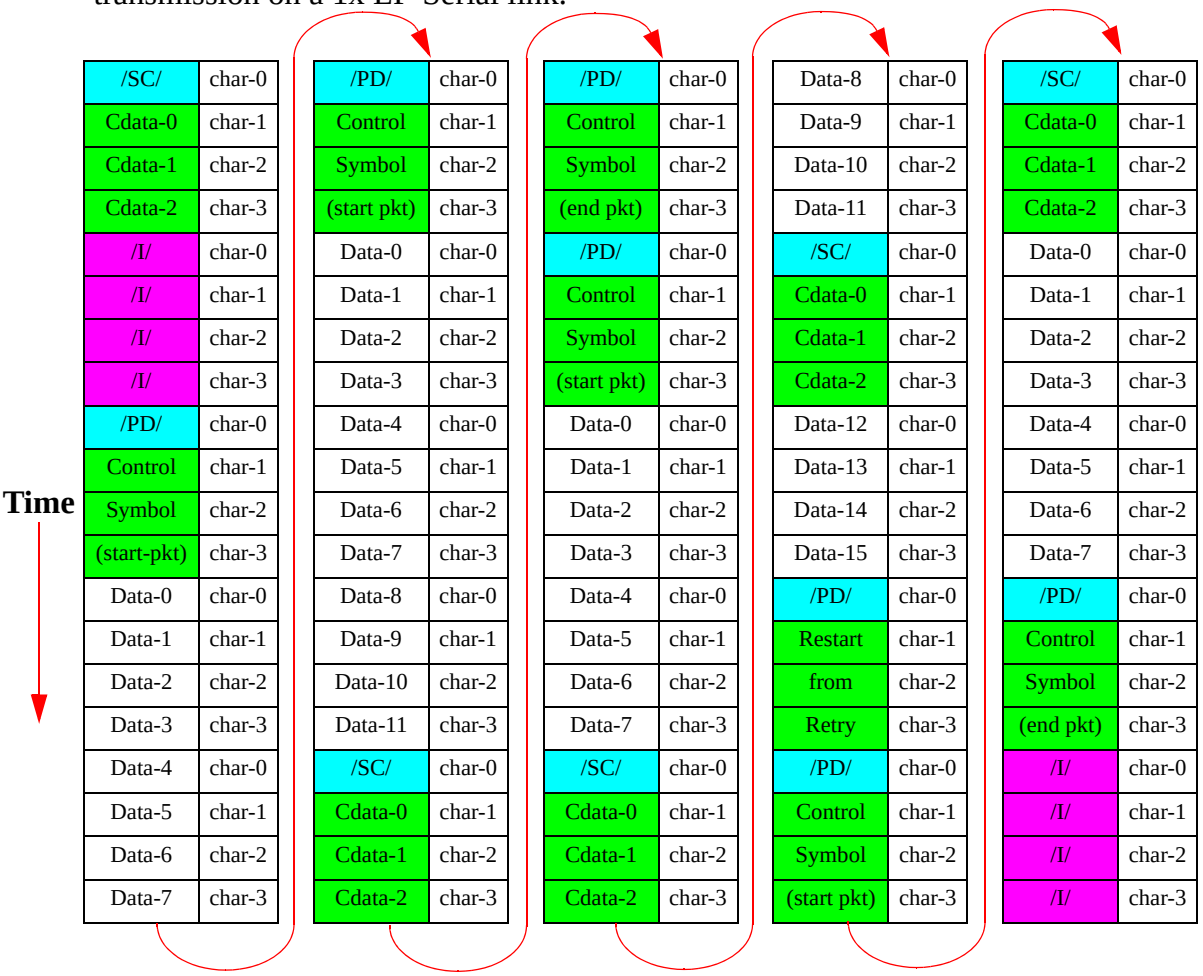


Figure 4-11. 1x Typical Data Flow with Short Control Symbol

4.9.2 Nx Ports Operating in 1x Mode

When a Nx port is operating in 1x mode, the character stream of delimited control symbols and packets received from the upper layers shall be fed in parallel to both lanes 0 and R for encoding and transmission in the order the characters were received from the upper layers. (The character stream is not striped across the lanes before encoding as is done when operating in Nx mode.) When neither delimited control symbols nor packets are available from the upper layers for transmission, an idle sequence shall be fed in parallel to both lane 0 and lane R for 8B/10B encoding and transmission on lanes 0 and R.

On reception, the code-group stream from either lane 0 or R shall be selected according to the state of the 1x/Nx_Initialization state machine (Section 4.12.4.5), decoded and the error free delimited control symbols and packets passed to the upper layers.

When a port that optionally supports and is enabled for both 2x mode and a wider Nx mode is operating in 1x, the port shall support both lanes 1 and 2 as redundancy lanes. The port shall transmit the 1x mode data stream on lanes 0, 1 and 2 and attempt to receive 1x mode data stream on lanes 0, 1 and 2. The port shall select between using the data received on lane 0 or the data received on the redundancy lane which may be either lane 1 or lane 2 depending on the connected port. Unless forced to use the redundancy lane, the port shall use the data stream received on lane 0 if it is available. The 1x/Nx_Initialization state machine specified in Section 4.12.4.6 shall be modified for a port supporting both 2x and a wider Nx mode to comply with the above requirements.

If the link is operating with idle sequence 2, control symbol and packet data characters shall be scrambled before transmission and descrambled after reception as specified in Section 4.8.

Once a Nx port is initialized to a 1x mode, the port may elect to disable the output driver of the lane which was not selected for reception by the initialization state machine of the connected port. Since the ports connected by the link may not be receiving on the same lane (one port could be receiving on lane 0 and the other port receiving on lane R), the connected port must be interrogated to determine which lane can be output disabled. It is recommended that the mechanism for disabling the output driver be under software control.

4.10 Nx Link Striping and Transmission Rules

A LP-Serial port operating in Nx mode shall stripe the character stream of delimited control symbols and packets received from the upper layers across the N active output lanes in the order the characters were received from the upper layers. Each lane shall then 8B/10B encode and transmit the characters assigned to it. When neither control symbols nor packets are available from the upper layers for transmission, an idle sequence shall be fed to each of the N lanes for 8B/10B encoding and transmission.

Packets and delimited control symbols shall be striped across the N active lanes beginning with lane 0. The first character of each packet, or delimited control symbol, shall be placed in lane K where K modulo 4 = 0. The second character shall be placed in lane (K + 1), and the n^{th} character shall be placed in lane (K + (n - 1)) which wraps around to lane 0 when (K + (n - 1)) modulo N = 0.

The lengths of control symbols and packets in the LP-Serial Physical Layer are positive integer multiples of 4 characters. As a result, when N, the width of the link, is greater than 4, occasions will occur when there are not enough packets and/or control symbols available for transmission to fill a column. For example, lanes 0-3 of a link operating in 8x mode contain a delimited short control symbol or the last 4 characters of a delimited long control symbol, but there is nothing available to put in lanes 4-7. When this occurs, all remaining characters in the column shall be filled (padded) with pseudo-random data characters. The first pseudo-random data pad

character shall occur in a lane whose lane_number modulo 4 = 0. The number of pseudo-random data pad characters in a column shall be a positive integer multiple of 4. If the link is operating with idle sequence 2, the pseudo-random data characters may be generated by using the lane scramblers to scramble D0.0 characters. Padding characters shall not be inserted between packet delimiting control symbols and the packet(s) they delimit.

After striping, each of the N streams of characters shall be independently 8B/10B encoded and transmitted.

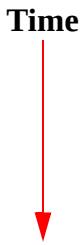
On reception, each lane shall be 8B/10B decoded.

If the link is operating with idle sequence 2, control symbol and packet data characters shall be scrambled before transmission and descrambled after reception as specified in Section 4.8.

After decoding, the N lanes shall be aligned. The ||A|| columns transmitted as part of an idle sequence provide the information needed to perform alignment. After alignment, the columns are destriped into a single character stream and passed to the upper layers.

The lane alignment process eliminates the skew between lanes so that after destriping, the ordering of characters in the received character stream is the same as the ordering of characters before striping and transmission. Since the minimum number of non ||A|| columns between ||A|| columns is 16, the maximum lane skew that can be unambiguously corrected is the time it takes to transmit 7 code-groups on a lane.

Figure 4-12 shows an example of Idle Sequence 1, short control symbol and packet transmission on a 4x link.



Lane-0	Lane-1	**Lane-2**	**Lane-3**
/SC/	Cdata-0	Cdata-1	Cdata-2
/I/	/I/	/I/	/I/
/PD/	Control Symbol (Start-of-packet)		
Data-0	Data-1	Data-2	Data-3
Data-4	Data-5	Data-6	Data-7
/PD/	Control Symbol (Start-of-packet)		
Data-0	Data-1	Data-2	Data-3
Data-4	Data-5	Data-6	Data-7
Data-8	Data-9	Data-10	Data-11
/SC/	Cdata-0	Cdata-1	Cdata-2
/PD/	Control Symbol (End-of-packet)		
/PD/	Control Symbol (Start-of-packet)		
Data-0	Data-1	Data-2	Data-3
Data-4	Data-5	Data-6	Data-7
/SC/	Cdata-0	Cdata-1	Cdata-2
Data-8	Data-9	Data-10	Data-11
/SC/	Cdata-0	Cdata-1	Cdata-2
Data-12	Data-13	Data-14	Data-15
/PD/	Control Symbol (Restart-from-retry)		
/PD/	Control Symbol (Start-of-packet)		
/SC/	Cdata-0	Cdata-1	Cdata-2
Data-0	Data-1	Data-2	Data-3
Data-4	Data-5	Data-6	Data-7
/PD/	Control Symbol (End-of-packet)		
/I/	/I/	/I/	/I/

Figure 4-12. Typical 4x Data Flow with Short Control symbol

4.11 Retimers and Repeaters

The LP-Serial Specification allows “retimers” and “repeaters”. Retimers amplify a weakened signal, but do not transfer jitter to the next segment because they use a local transmit clock. Repeaters also amplify a weakened signal, but transfer jitter to the next segment because they use a transmit clock derived from the received data stream. Retimers allow greater distances between end points at the cost of additional latency. Repeaters support less distance between end points than retimers and only add a small amount of latency.

4.11.1 Retimers

A retimer shall comply with all applicable AC specifications found in Chapter 8, "Common Electrical Specifications", Chapter 9, "1.25Gbaud, 2.5Gbaud, and 3.125Gbaud LP-Serial Links", and Chapter 10, "5Gbaud and 6.25Gbaud LP-Serial Links". This includes resetting the jitter budget thus extending the transmission distance for the link. The retimer repeats the received code-groups after performing code-group synchronization and serializes the bitstream again on transmission, based on a local clock reference. Up to two retimers are allowed between two end nodes.

A retimer is not RapidIO protocol-aware or addressable in any way. The only awareness a retimer has is to the synchronization on the /K/ code-group and the function of /R/ insertion and removal. A retimer may insert up to one /R/ code-group immediately following a /K/ code-group, or remove one /R/ code-group that immediately follows a /K/ code-group. Since the /R/ code-group is disparity neutral, its insertion or deletion does not affect the running disparity.

A N-lane retimer must perform lane synchronization and deskew, in exactly the same way a RapidIO device implementing this physical layer does when synchronizing inputs during initialization and startup. A Nx mode retimer will synchronize and align all lanes that are driven to it. Therefore, such a retimer allows for the degradation of an input Nx link to a 1x link on either lane 0 or R. If any link drops out, the retimer must merely continue to pass the active links, monitoring for the compensation sequence and otherwise passing through whatever code-groups appear on its inputs. A retimer may optionally not drive any outputs whose corresponding inputs are not active.

Any insertion or removal of a /R/ code-groups in a N-lane retimer must be done on a full column. A retimer may retime links operating at the same width only (i.e. cannot connect a link operating at 1x to a link operating at Nx). A retimer may connect a 1x link to a Nx link that is operating in 1x mode. Retimers perform clock tolerance compensation between the receive and transmit clock. The transmit clock is usually derived from a local reference.

Retimers do not check for code violations. Code-groups received on one port are transmitted on the other regardless of code violations or running disparity errors.

4.11.2 Repeaters

A repeater is used to amplify the signal, but does not retime the signal, and therefore can add additional jitter to the signal. It does not compensate for clock rate variation. The repeater repeats the received code-groups as the bits are received by sampling the incoming bits with a clock derived from the bit stream, and then retransmitting them based on that clock. Repeaters may be used with Nx links but lane-to-lane skew may be amplified. Repeaters do not interpret or alter the bit stream in any way.

4.12 Port Initialization

This section specifies the port initialization process. The process includes detecting the presence of a partner at the other end of the link (a link partner), establishing bit synchronization and code-group boundary alignment and if present, adjusting any adaptive equalizers. The process also includes determining if the connected port supports an Nx mode in addition to 1x mode and selecting 1x or Nx mode operation, then, if 1x mode is selected, selecting lane 0 or lane R (the redundancy lane, lane 1 for 2x ports and lane 2 for 4x, 8x or 16x ports) for link reception.

Port initialization may optionally include baud rate discovery.

The initialization process is controlled by several state machines. The number and type of state machines depends on whether the port supports only 1x mode (a 1x port) or supports both 1x and one or more Nx modes (a 1x/Nx port). In either case, there is a primary state machine and one or more secondary state machines. The use of multiple state machines results in a simpler overall design. As might be expected, the initialization process for a 1x port is simpler than and is a subset of the initialization process for a 1x/Nx port.

The port initialization process supports an optional test mode that allows ports that support more than one multi-lane mode of operation to enable and monitor the operation of the inactive lanes when the port is operating at less than maximum width. The performance of inactive lanes can be monitored only if the inactive lanes are connected to and supported by the connected port and the test mode is implemented and enabled in both ports. The test mode is enabled with the “Enable inactive lanes” bit defined in Section 6.7.2.2. The initiation, implementation and interpretation of tests conducted using this test mode is outside of this specification.

The initialization process for 1x, 1x/Nx ports, and ports supporting 1x mode and multiple Nx modes is both described in text and specified with state machine diagrams. **In the case of conflict between the text and a state machine diagram, the state machine diagram takes precedence.**

4.12.1 1x Mode Initialization

The initialization process for ports that support only 1x mode shall be controlled by two state machines, 1x_Initialization and Lane_Synchronization. 1x_Initialization is the primary state machine and Lane_Synchronization is the secondary state machine. The operation of these state machines is described and specified in Section 4.12.4.4 and Section 4.12.4.2 respectively.

4.12.2 1x/Nx Mode Initialization

The initialization process for ports that support both 1x and a Nx mode is controlled by a primary state machine and four or more secondary state machines. The primary state machine is the 1x/Nx_Initialization state machine. Lane_Synchronization[0]

through Lane_Synchronization[N-1] (one for each of the N lanes), Lane_Alignment (one for each supported Nx mode) and 1x/2x_Mode_Detect (used only in the 1x/2x_Initialization state machine) are the secondary state machines. The operation of the secondary state machines is described and specified in Section 4.12.4.2 through Section 4.12.4.4 respectively.

The 1x/Nx_Initialization state machine provides a degree of LP-Serial link width auto-negotiation. The goal of the auto-negotiation is to ensure that any connected combination of 1x, 1x/2x, 1x/4x, 1x/8x or 1x/16x LP-Serial ports that are configured in some manner to operate at the same baud rate will automatically find a link width over which they can communicate. For example if a 1x/4x port is connected to a 1x/8x port, they will auto-negotiate to operate in 1x mode. If however the 1x/8x port optionally also supports 4x mode (making it a 1x/4x/8x port) and its 1x/Nx_Initialization state machine has been modified as shown in Figure 4-22 to be a 1x/4x/8x_Initialization state machine, then the ports will auto-negotiate to operate in 4x mode.

In most configurations, the auto-negotiation also ensures that a pair of connected multi-lane LP-Serial ports configured in some manner to operate at the same baud rate will find a link width over which they can communicate in the presence of a lane failure. For example, if two 1x/4x ports are connected and lane 0 is broken in one direction, the ports will auto-negotiate to operate in 1x mode using lane 0 in the direction that lane 0 is operational and lane 2 in the direction that lane 0 is broken. This feature works only for pairs of ports that support the same redundancy lane. It does not work when a 1x/2x port is connected to a 1x/4x or wider port.

4.12.3 Baud Rate Discovery

Baud rate discovery is optional. If implemented, baud rate discovery occurs during the SEEK state of the 1x_Initialization and 1x/Nx_Initialization state machines. Ports that implement baud rate discovery shall use the following algorithm.

1. When the port enters the SEEK state, it begins transmitting an idle sequence on lane 0 and, if the port supports a Nx mode, on lane R, the 1x mode redundancy lane. The idle sequence shall be transmitted at the highest lane baud rate that is supported by the port and that is enabled for use.
2. The port shall then look for an inbound signal on lane 0 or lane R of the link from a connected port. The method of detecting the presence of an inbound signal from a connected port is implementation specific and outside the scope of this specification.
3. Once an inbound signal is detected, the port shall determine the baud rate of the signal. The method of detecting the baud rate of the signal is implementation specific and outside the scope of this specification.
4. If the baud rate of the inbound signal is the same as the baud rate at which the port is transmitting, the link shall operate at that per lane baud rate until the port reenters the SEEK state and the baud rate discovery process is complete.

5. If the baud rate on the inbound signal is less than the baud rate of the idle sequence being transmitted by the port, the port shall reduce the baud rate at which it is transmitting to the next lowest baud rate that it supports and that is enabled for use and go to step 2.
6. If the baud rate on the inbound signal is greater than the baud rate of the idle sequence being transmitted by the port, the port shall continue transmitting at the current baud rate go to step 2.

An informational state diagram for the Baudrate_Discovery state machine is shown in Figure 4-13.

The techniques and algorithms used to compare the baud rates of the signals being transmitted and received are implementation specific and beyond the scope of this specification.

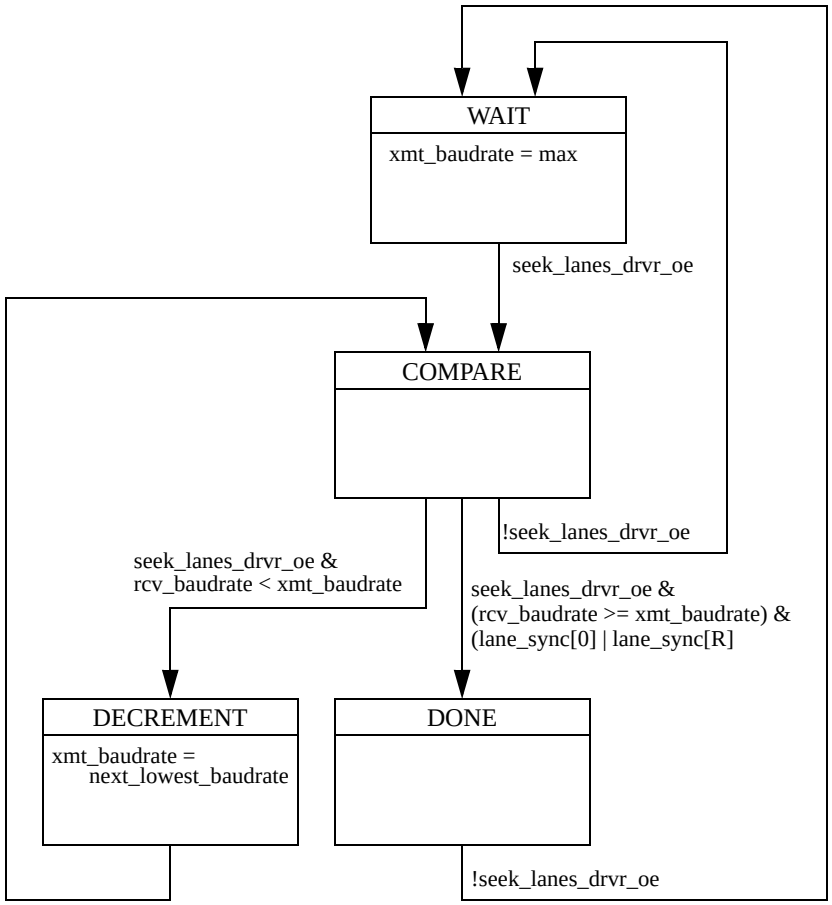


Figure 4-13. Baudrate_Discovery state machine (Informational)

4.12.4 State Machines

4.12.4.1 State Machine Conventions, Functions and Variables

4.12.4.1.1 State Machine Conventions

The conventions used in state machine specification are as follows.

A state machine state is persistent until an exit condition occurs.

A state machine variable that is listed in the body of a state but is not part of an assignment statement is asserted for the duration of that state only.

A state machine variable that is assigned a value in the body of a state retains that value until assigned a new value in another state.

A state machine function that is listed in the body of a state is executed once during the state.

A state machine variable is asserted when its value is 1 and de-asserted when it value is 0.

Except when otherwise directed by parentheses, the order of precedence of logic operations when evaluating a logic expression is, in order of decreasing precedence, negation/compliment (!) followed by intersection (&) and union (|).

Logic expressions within paired parentheses are evaluated before the rest of a logic expression is evaluated with the operations within the innermost pair of parentheses evaluated first.

4.12.4.1.2 State Machine Functions

The functions used in the state machines are defined as follows.

change( )

Asserted when the variable on which it operates changes state.

next_code_group( )

Gets the next 10 bit code-group for the lane when it becomes available.

next_Ncolumn( )

Gets the next column of N code-groups or characters, as appropriate, from lanes 0 to N-1 when it becomes available.

4.12.4.1.3 State Machine Variables

The variables used in the state machines are defined as follows.

1x_mode_delimiter

Asserted when a column of two characters from lanes 0 and 1 contains two SC or two PD special characters. Otherwise de-asserted.

1x_mode_detected

Asserted by the 1x/2x_Mode_Detect state machine when it determines that the link receiver input signals on lanes 0 and 1 are in 1x mode. Otherwise, de-asserted.

2x_mode_delimiter

Asserted when a column of two characters from lanes 0 and 1 contains one SC or PD special character and one data character. Otherwise de-asserted.

||A||

Asserted when the current column contains all /A/s. Otherwise de-asserted.

Acounter

A counter used in the Lane Alignment state machine to count received alignment columns (||A||s).

align_error

Asserted when the current column contains at least one /A/, but not all /A/s. Otherwise, de-asserted.

/COMMA/

If Idle Sequence 1 is being used on the link to which the port is connected, asserted when the current code-group is /K28.5/. Otherwise, de-asserted.

If Idle Sequence 2 is being used on the link to which the port is connected, asserted when the current code-group is either /K28.1/ or /K28.5/. Otherwise, de-asserted.

Dcounter

A 2-bit synchronous saturating up/down counter with the behavior specified in Table 4-10. The counter is used in the 1x/2x_Mode_Detect state machine.

Table 4-10. Dcounter Definition

Counter Value	(count_up,count_down)			
	0,0	0,1	1,0	1,1
0x0	0x0	0x0	0x1	0x0
0x1	0x1	0x0	0x2	0x1
0x2	0x2	0x1	0x3	0x2

Table 4-10. Dcounter Definition

Counter Value	(count_up,count_down)			
	0,0	0,1	1,0	1,1
0x3	0x3	0x2	0x3	0x3

disc_tmr_done (discovery timer done)

Asserted when disc_tmr_en has been continuously asserted for 28 +/- 4 msec and the state machine is in the DISCOVERY or a RECOVERY state. The assertion of disc_tmr_done causes disc_tmr_en to be de-asserted. When the state machine is in a state other than the DISCOVERY or a RECOVERY state, disc_tmr_done is de-asserted.

disc_tmr_en (discovery timer enable)

When asserted, the discovery timer (disc_tmr) runs. When de-asserted, the discovery timer is reset to and maintains its initial value.

force_1x_mode

Asserted when all Nx (multi-lane) modes are disabled. When asserted, forces the 1x/Nx Initialization state machine to use 1x mode.

force_laneR

When force_1x_mode is asserted, force_laneR controls whether lane 0 or lane R, the redundancy lane, is preferred for 1x mode reception. If force_laneR is asserted, lane R is the preferred lane. If force_laneR is deasserted, lane 0 is the preferred lane. If the preferred lane is functional, it is selected by the port initialization state machine for 1x mode reception. If the preferred lane is not functional, the non-preferred lane, if functional, is selected for 1x mode reception.

If force_1x_mode is not asserted, the state of force_laneR has no effect on the initialization state machine.

force_reinit

When asserted, forces the port Initialization state machine to re-initialize. The signal is set under software control and is cleared by the Initialization state machine.

Icounter

Counter used in the Lane_Synchronization state machine to count INVALID received code-groups. There is one Icounter for each lane in a Nx mode receiver.

idle_selected

When asserted, indicates that the IDLE sequence for use on the link has been selected by the Idle Sequence Selection process specified in Section 4.7.5.

If the port supports only one IDLE sequence at the current baud rate, the bit is always asserted.

If the port supports multiple IDLE sequences at the current baud rate, the bit is de-asserted when the Initialization state machine is in the SILENT state and is otherwise controlled by the Idle Sequence Selection process. The Idle Sequence Selection process runs when the Initialization state machine is in the SEEK state and lane_sync has been asserted for lane 0, 1 and/or 2. The bit is asserted when the Idle Sequence Selection process completes.

/INVALID/

When asserted, /INVALID/ indicates that the current code-group is an invalid code-group.

Kcounter

Counter used in the Lane_Synchronization state machine to count received code-groups that contain a comma pattern. There is one Kcounter for each lane in a Nx mode receiver.

lane_ready[n]

lane_ready[n] = lane_sync[n] & rcvr_trained[n]

lane_sync

Asserted by the Lane_Synchronization state machine when it determines that the lane it is monitoring is in bit synchronization and code-group boundary alignment. Otherwise de-asserted.

lane_sync[n]

The lane_sync signal for lane n.

lanes01_drvr_oe

When asserted, the output drivers for lanes 0 and 1 are enabled

lanes02_drvr_oe

When asserted, the output drivers for lanes 0 and 2 are enabled

lanes13_drvr_oe

When asserted, the output drivers for lanes 1 and 3 are enabled

link_drvr_oe (link driver output enable)

When asserted, the output link driver of a 1x port is enabled.

Mcounter

Mcounter is used in the Lane_Alignment state machine to count columns received that contain at least one /A/, but not all /A/s.

N_lanes_aligned

Asserted by the Lane_Alignment state machine when it determines that lanes 0 through N-1 are in sync and aligned.

N_lanes_drvr_oe

The output enable for the lanes 0 through N - 1.

N_lanes_ready

N_lanes_ready = N_lanes_aligned & lane_ready[0] & ... & lane_ready[N-1]

N_lane_sync

Indicates when lanes 0 through N-1 of a receiver operating in Nx mode are in bit synchronization and code-group boundary alignment.

N_lane_sync = lane_sync[0] & ... & lane_sync[N-1]

Nx_mode

Asserted when the port is initialized and operating in Nx mode

port_initialized

When asserted, port_initialized indicates that the port is initialized. Otherwise the port is not initialized. The state of port_initialized affects what the port may transmit on and accept from the link.

receive_lane1

In a 2x port that is initialized and is operating in 1x mode (2x_mode de-asserted), receive_lane1 indicates which lane the port has selected for input. When asserted, the port input is taken from lane 1. When de-asserted the port input is taken from lane 0. When the port is operating in 2x mode (2x_mode asserted), receive_lane1 is undefined and shall be ignored.

receive_lane2

In a Nx port that is initialized and is operating in 1x mode (Nx_mode de-asserted for all N > 2), receive_lane2 indicates which lane the port has selected for input. When asserted, the port input is taken from lane 2. When de-asserted the port input is taken from lane 0. When the port is operating in Nx mode (some Nx_mode asserted), receive_lane2 is undefined and shall be ignored.

rcvr_trained[n]

De-asserted when the local lane[n] receiver controls adaptive equalization in the receiver and/or the connected lane[n] transmitter and the training of the equalization in either the lane[n] receiver or the connected lane[n] transmitter has not been

completed. Otherwise, asserted.

seek_lanes_drvr_oe

The output enable for the lane 0 and the lane R output drivers of a 1x/Nx port.

signal_detect

Asserted when a lane receiver is enabled and a signal meeting an implementation defined criteria is present at the input of the receiver. The use of signal_detect is implementation dependent. It may be continuously asserted or it may be used to require that some implementation defined additional condition be met before the Lane_Synchronization state machine is allowed to exit the NO_SYNC state. Signal_detect might for example be used to ensure that the input signal to a lane receiver meet some minimum AC input power requirement to prevent the receiver from locking on to crosstalk.

silence_timer_done

Asserted when silence_timer_en has been continuously asserted for 120 +/- 40 µs and the state machine is in the SILENT state. The assertion of silence_timer_done causes silence_timer_en to be de-asserted. When the state machine is not in the SILENT state, silence_timer_done is de-asserted.

silence_timer_en

When asserted, the silence_timer runs. When de-asserted, the silence_timer is reset to and maintains its initial value.

/VALID/

When asserted, /VALID/ indicates that the current code-group is a valid code-group given the current running disparity.

Vcounter

Vcounter is used in the Lane_Synchronization state machine to count VALID received code-groups. There is one Vcounter for each lane in a Nx mode receiver.

4.12.4.2 Lane Synchronization State Machine

The Lane_Synchronization state machine monitors the bit synchronization and code-group boundary alignment for a lane receiver. A port that supports only 1x mode (1x port) has one Lane_Synchronization state machine. A port that supports Nx mode has N Lane_Synchronization state machines, one for each lane (Lane_Synchronization[0] through Lane_Synchronization[N-1]).

The Lane_Synchronization state machine is specified in Figure 4-14

The state machine determines the bit synchronization and code-group boundary alignment state of a lane receiver by monitoring the received code-groups and looking for code-groups containing the “comma” pattern, other valid code-groups and invalid code-groups. The “comma” pattern is the bit sequence that is used to

establish code-group boundary alignment. When a lane is error free the “comma” pattern occurs only in the /K28.1/ and /K28.5/ code-groups. Several counters are used to provide hysteresis so that occasional bit errors do not cause spurious lane_sync state changes.

The state machine does not specify how bit synchronization and code-group boundary alignment is to be achieved. The methods used by a lane receiver to achieve bit synchronization and code-group boundary alignment are implementation dependent. However, an isolated single bit or burst error shall not cause the code-group boundary alignment mechanism to change alignment. For example, a single bit or burst error that results in a “comma” pattern across a code-group boundary shall not cause the code-group boundary alignment mechanism to change alignment.

The state machine starts in the NO_SYNC state and sets the variables Kcounter[n], Vcounter[n], and lane_sync[n] to 0 (lane n is out of code-group boundary sync). It then looks for a /COMMA/ code-group. When it finds one and the signal signal_detect[n] is asserted, the machine moves to the NO_SYNC_1 state.

The NO_SYNC_1 state in combination with the NO_SYNC_2 and NO_SYNC_3 states looks for the reception of 127 /COMMA/ and Vmin /VALID/ code-groups without any intervening /INVALID/ code-groups. When this condition is achieved, state machine goes to state SYNC. If an intervening /INVALID/ code-group is detected, the machine goes back to the NO_SYNC state.

The values of 127 and Vmin are selected such that it is highly unlikely that SYNC would be falsely reported and that the bit error rate (BER) is low enough that it is highly unlikely that once asserted, lane_sync will “flicker” ON and OFF while the training of the receiver timing recovery and any adaptive equalization is completed. Vmin shall have a minimum value of 0 and is implementation dependent. When Vmin = 0, the behavior of this Lane_Synchronization state machine is identical to that of the Lane_Synchronization state machine specified in Rev. 1.3 of this specification.

Table 4-11 shows the approximate maximum probability of lane_sync “flicker” for some values of Vmin and over the BER range of $1*10^{-2}$ to $1*10^{-12}$. It is recommended that Vmin be at least $2^{12} - 1$.

Table 4-11. lane_sync “Flicker” Probability

Vmin	Approximate maximum probability of lane_sync flicker
0	0.24
$2^{12} - 1$	0.021
$2^{13} - 1$	0.011

Table 4-11. lane_sync "Flicker" Probability

Vmin	Approximate maximum probability of lane_sync flicker
$2^{14} - 1$	0.0056
$2^{15} - 1$	0.0028
$2^{16} - 1$	0.0014

When Vmin = 0 and IDLE1 is being received, something more than 256 code-groups must be received after the first /COMMA/ to achieve the 128 /COMMA/ code-groups without error criteria to transition to the SYNC state because the /COMMA/ code-group comprises slightly less than half of the code-groups in the IDLE1 sequence.

When Vmin = 0 and IDLE2 is being received, something more than 9 Idle Frames must be received after the first /COMMA/ to achieve the 127 /COMMA/ code-groups without error to transition to the SYNC state because there are on average about 14 /COMMA/ code-groups per Idle Frame.

In the SYNC state, the machine sets the variable lane_sync[n] to 1 (lane n is in code-group boundary sync), sets the variable Icounter[n] to 0 and begins looking for /INVALID/ code-groups. If an /INVALID/ code-group is detected, the machine goes to state SYNC_1.

The SYNC_1 state in combination with the SYNC_2, SYNC_3, and SYNC_4 states looks for 255 consecutive /VALID/ code-groups without any /INVALID/ code-groups. When 255 /VALID/ symbols are received, the Icounter[n] value is decremented in the transition through the SYNC_4 state. If it does not, it increments Icounter[n]. If Icounter[n] is decremented back to 0, the state machine returns to the SYNC state. If Icounter[n] is incremented to Imax, the state machine goes to the NO_SYNC state and starts over. Imax is an integer and shall have a value of 3 or greater for receivers not using DFE and a value of 4 or greater for receivers using DFE. This algorithm tolerates isolated single bit or burst errors in that an isolated single bit or burst error will not cause the machine to change the variable lane_sync[n] from 1 to 0 (in sync to out of sync).

A single bit error at the code-group level can cause two INVALID characters to be reported, one due to a corrupted code-group and one due to corrupted running disparity with causes a subsequent code-group to be reported as INVALID. A burst error no longer than 11 bits in length can cause three INVALID characters to be reported, two due to two corrupted code-groups and one due to corrupted running disparity which causes a subsequent code-group to be reported as INVALID.

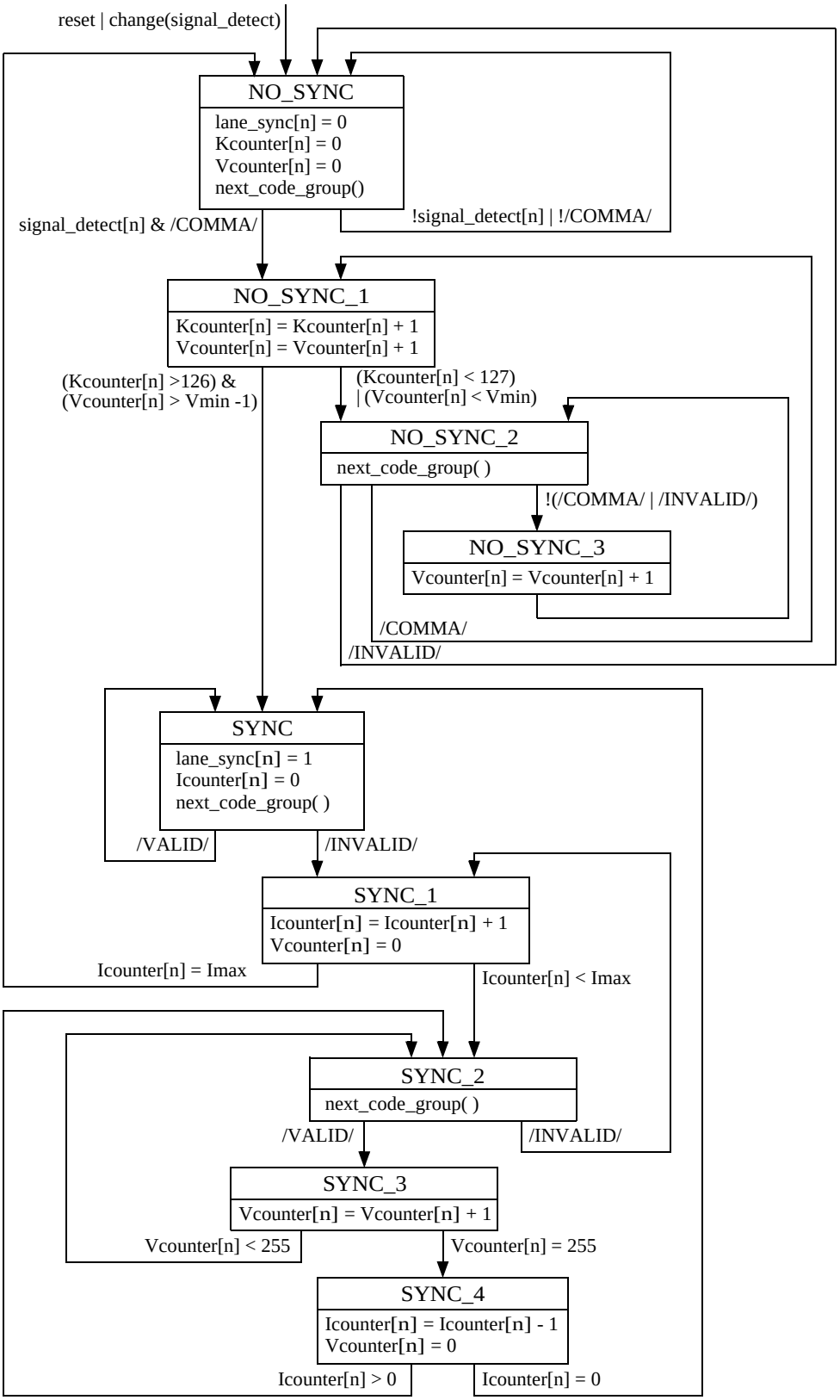


Figure 4-14. Lane_Synchronization State Machine

4.12.4.3 Lane Alignment State Machine

The Lane_Alignment state machine monitors the alignment of the output of the N lane receivers in a port operating in Nx mode. A port supporting one or more multi-lane modes has one Lane_Alignment state machine for each supported Nx mode. A port supporting only 1x mode does not have a Lane_Alignment state machine. Lane alignment is required in a Nx port receiver to compensate for unequal propagation delays through the N lanes.

The Lane_Alignment state machine is specified in Figure 4-15.

The state machine determines the alignment state by monitoring the N lanes for columns containing all /A/s (||A||), columns containing at least one but not all /A/s and columns containing no /A/s. Several counters are used to provide hysteresis so that isolated single bit or burst errors do not cause spurious lanes_aligned state changes.

The state machine does not specify how lane alignment is to be achieved. The methods used by a port receiver to achieve lane alignment are implementation dependent. However, isolated single bit or burst errors shall not cause the lane alignment mechanism to change lane alignment. For example, an isolated single bit or burst error that results in a column that contains at least one /A/ but not all /A/s shall not cause the lane alignment mechanism to change the lane alignment.

The state machine starts in the NOT_ALIGNED state where the variables Acounter and N_lanes_aligned are set to 0 (all N lanes are not aligned). The machine then waits for all N lanes to achieve code-group boundary alignment (N_lanes_sync asserted) and the reception of an ||A|| (a column of all /A/s). When this occurs, the machine goes to NOT_ALIGNED_1 state.

The NOT_ALIGNED_1 state in combination with the NOT_ALIGNED_2 state looks for the reception of four ||A||s without the intervening reception of a misaligned column (a column with at least one /A/ but not all /A/s which causes the signal align_error to be asserted). When this occurs, the machine goes to the ALIGNED state. If an intervening misaligned column is received, the machine goes back to the NOT_ALIGNED state.

In the ALIGNED state, the machine sets the variable N_lanes_aligned to 1 (all N lanes are aligned) and the variable Mcounter to 0 and looks for a misaligned column (align_error asserted). If a misaligned column is detected, the machine goes to the ALIGNED_1 state.

The ALIGNED_1 state in combination with the ALIGNED_2 and ALIGNED_3 states look for the reception of four ||A||s without the intervening reception of more than Mmax - 1 additional misaligned columns. If this condition occurs, the state machine returns to the ALIGNED state. If Mmax - 1 additional intervening misaligned columns occurs, the machine goes to the NOT_ALIGNED state and

starts over. Mmax is an integer and shall have a value of 2 or greater for receivers not using DFE and a value of 3 or greater for receivers using DFE.

This algorithm tolerates an isolated single bit or burst error in that such an error will not cause the machine to change the variable N_lanes_aligned from 1 to 0 (in lane alignment to out of lane alignment).

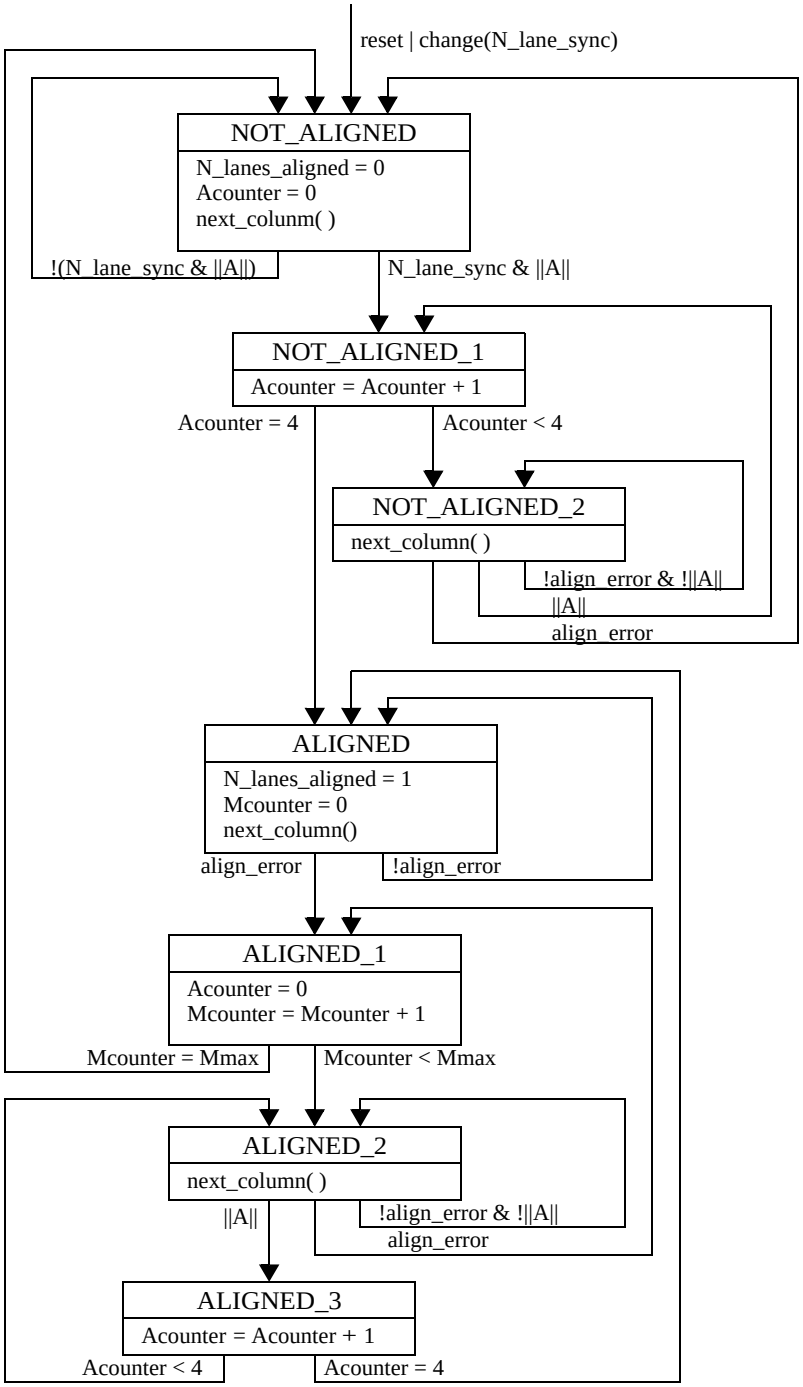


Figure 4-15. Lane_Alignment State Machine

4.12.4.4 1x/2x Mode Detect State Machine

The 1x/2x_Mode_Detect state machine monitors the columns formed from aligned characters received on lanes 0 and 1 of a port that supports 2x mode. When such a port is receiving an input signal on only lanes 0 and 1, the state machine is used to determine whether the connected port is transmitting in 1x mode or in 2x mode. A port that supports 2x mode shall have one 1x/2x_Mode_Detect state machine.

The 1x/2x_Mode_Detect state machine is specified in Figure 4-16.

Except for the case of N = 2, a 1x/Nx receiver can tell whether the connected port is operating in 1x mode or Nx mode by observing the number of active lanes it is receiving (the number of lanes for which lane_sync[n] is asserted). This follows from the fact that a 1x/Nx port operating in 1x mode transmits only on lanes 0 and R. In the case of N = 2, the port transmits on both lanes regardless of whether it is operating in 1x or 2x mode making it impossible for a 1x/2x receiver to determine the mode of the connected port based on the number of active lanes it is receiving. The 1x/2x_Mode_Detect state machine provides mode detection for the 1x/2x receiver.

The 1x/2x_Mode_Detect state machine enters the INITIALIZE state whenever the port is reset or the state of 2_lanes_aligned changes state. The machine initializes the 1x_mode_detected and Dcounter variables (the connected port is initially assumed to be operating in 2x mode) and waits for the lanes to become aligned. Once the two lanes are aligned, the machine goes to the GET_COLUMN state to get the next available column.

In the GET_COLUMN state, each column is examined as it becomes available to determine whether it contains any control symbol delimiter special characters (SC or PD characters). If no SC or PD characters are found, no action is taken and the state machine remains in the GET_COLUMN state. If the column contains a single SC or PD special character, the column is determined to be a 2x mode delimiter and the state machine enters the 2x_DELIMITER state. If the column contains a two SC or two PD special characters, the column is determined to be a 1x mode delimiter and the state machine enters the 1x_DELIMITER state.

In the 1x_DELIMITER state, the Dcounter is decremented by 1 and the value of the Dcounter is tested. If the Dcounter is > 0, the state machine goes to the GET_COLUMN state. If the Dcounter is $= 0$, the state machine goes to the SET_1x_MODE state where 1x_mode_detected is set to 1. The state machine then goes to the GET_COLUMN state.

In the 2x_DELIMITER state, the Dcounter is incremented by 1 and the value of the Dcounter is tested. If the Dcounter is < 3, the state machine goes to the GET_COLUMN state. If the Dcounter is $= 3$, the state machine goes to the SET_2x_MODE state where 1x_mode_detected is set to 0. The state machine then goes to the GET_COLUMN state.

The Dcounter is used to prevent transmission errors from erroneously changing the state of 1x_mode_detected.

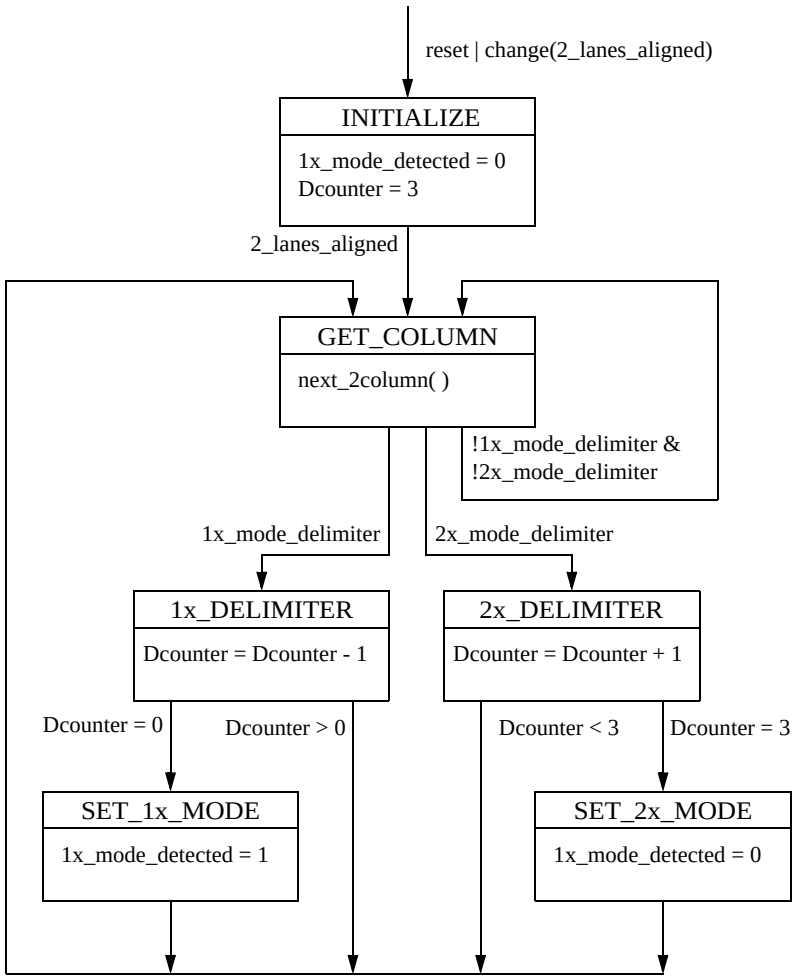


Figure 4-16. 1x/2x_Mode_Detect State Machine

4.12.4.5 1x Mode Initialization State Machine

The 1x_Initialization state machine specified in this section shall be used by ports that support only 1x mode (1x ports). The state machine is specified in Figure 4-17.

The machine starts in the SILENT state. The link output driver is disabled to force the link partner to initialize regardless of its current state. The duration of the SILENT state is controlled by the silence_timer. The duration must be long enough to ensure that the link partner detects the silence (as a loss of lane_sync) and is forced to initialize but short enough that it is readily distinguished from a link break. When the silent interval is complete, the SEEK state is entered.

In the SEEK state, the link output driver is enabled, an idle sequence is transmitted, and the port waits for lane_ready to be asserted indicating the presence of a link partner. While lane_ready as defined indicates the bit and code-group boundary

alignment state of the link receiver, it is used by the state machine to indicate the presence of a link partner. When lane_ready and idle_selected are both asserted, the 1X_MODE state is entered.

The input signal force_reinit allows the port to force link initialization at any time.

The variable port_initialized is asserted only in the 1X_MODE state.

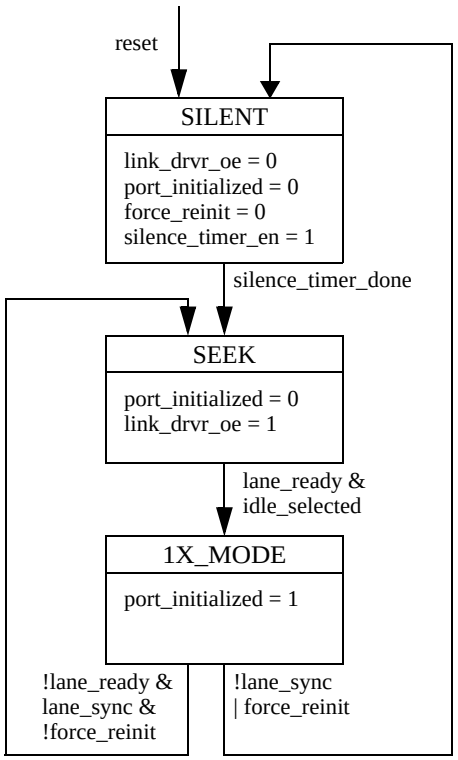


Figure 4-17. 1x_Initialization State Machine

4.12.4.6 1x/Nx Mode Initialization State Machine for N = 4, 8, 16

The 1x/Nx_Initialization state machines specified in this section shall be used by ports that support both 1x mode and an Nx mode (1x/Nx ports) for N = 4, 8 or 16. The initialization state machine for 1x/2x ports is specified in Section 4.12.4.7. 1x/8x and 1x/16x ports shall use the 1x/Nx_Initialization state machine specified in Figure 4-18. 1x/4x ports should use the 1x/Nx_Initialization state machine specified in Figure 4-18, but may use the 1x/4x_Initialization state machine specified in Figure 4-19. The 1x/4x_Initialization state machine of Figure 4-19 shall not be used in new designs.

The 1x/Nx_Initialization state machine controls port initialization and determines when the port is initialized. The state machine also controls whether the port receiver operates in 1x or Nx mode and in 1x mode whether lane 0 or lane 2, the 1x mode redundancy lane, is selected for control symbol and packet reception.

The 1x/Nx_Initialization state machine starts in SILENT state. All N lane output drivers are disabled to force the link partner to re-initialize regardless of its current state. The duration of the SILENT state is controlled by the silence_timer. The duration must be long enough to ensure that the link partner detects the silence (as a loss of lane_sync) and is forced to re-initialize. When the silent interval is complete, the state machine enters the SEEK state.

In the SEEK state, a 1x/Nx port transmits an idle sequence on lanes 0 and 2 (the other output drivers remain disabled to save power) and waits for an indication that a link partner is present. While lane_sync as defined indicates the bit and code-group boundary alignment state of a lane receiver, it is used by the state machine to indicate the presence of a link partner. A link partner is declared to be present when either lane_sync[0] or lane_sync[2] is asserted. The assertion of idle_selected and either lane_sync[0] or lane_sync[2] causes the state machine to enter the DISCOVERY state.

In the DISCOVERY state, the port enables the output drivers for all N lanes and transmits an idle sequence on all N lanes if Nx mode is enabled. The discovery timer (disc_tmr) is started. The discovery timer allows time for the link partner to enter its DISCOVERY state and if Nx mode is enabled in the link partner, for all N local lane receivers to acquire bit synchronization and code-group boundary alignment and to complete the training of any adaptive equalization that is present and for all N lanes to be aligned.

While waiting for the end of the discovery period (disc_tmr_en asserted but disc_tmr_done de-asserted), if Nx_mode is enabled, all N lanes become ready and lane alignment is achieved (N_lanes_ready asserted), the machine enters the Nx_MODE state. If force_1x_mode is asserted (Nx_mode_enabled is deasserted), force_laneR is not asserted and lane 0 becomes ready (lane_ready[0] asserted), the machine enters the 1x_MODE_LANE0 state. If both force_1x_mode and force_laneR are asserted and lane 2 becomes ready (lane_ready[2] asserted), the machine enters the 1x_MODE_LANE2 state.

At the end of the discovery period (disc_tmr_done asserted), if the state machine has not entered the Nx_mode or one of the 1x modes and at least one of lane 0 or lane 2 is ready, the machine will enter one of the 1x mode states. If lane 0 is ready and either force_1x_mode and force_laneR are asserted but lane 2 is not ready or Nx mode is enabled but N_lanes_ready is deasserted, the machine enters the 1X_MODE_LANE0 state. If lane 2 is ready, lane 0 is not ready and either force_1x_mode is asserted and force_laneR is not asserted or neither force_1x_mode nor N_lanes_ready are asserted, the machine enters the 1X_MODE_LANE2 state. If neither lane_ready[0] nor lane_ready[2] is asserted, the machine enters the SILENT state and restarts the port initialization process.

If lane synchronization for both lane 0 and lane R is lost (both lane_sync[0] and lane_sync[2] de-asserted) during the DISCOVERY state, the state machine enters the SILENT state and restarts the port initialization process.

When in the Nx_MODE state, port_initialized is asserted. If N_lanes_ready is lost (N_lanes_ready de-asserted), the state machine transitions to either the SILENT state if both lane_sync[0] and lane_sync[2] are de-asserted or the DISCOVERY state if either lane_sync[0] or lane_sync[2] is asserted. This allows a 1x/Nx port in the Nx_MODE state to recover to Nx_MODE if N_lanes_ready was de-asserted due to multi-bit reception error or the need to retrain some of the adaptive equalization, but also allows the port to switch to 1x mode if the port is no longer able to receive in Nx mode or if the connected 1x/Nx port is not able to receive in Nx mode and has switched to 1x mode.

When in the 1x_MODE_LANE0 state, port_initialized is asserted. If lane_ready[0] is de-asserted but lane_sync[0] is still asserted, the machine transitions to the 1x_RECOVERY state to attempt recovery to the 1x_MODE_LANE0 state. If lane_sync[0] is de-asserted the state machine enters the SILENT state.

When in the 1x_MODE_LANE2 state, port_initialized is asserted. If lane_ready[2] is de-asserted but lane_sync[2] is still asserted, the machine transitions to the 1x_RECOVERY state to attempt recovery to the 1x_MODE_LANER state. If lane_sync[2] is de-asserted, the state machine enters the SILENT state.

When the 1x_RECOVERY state is entered, the discovery timer (disc_tmr_en asserted) is started. The port reenters the 1x_MODE_LANE0 state if lane_ready[0] is reasserted and the port was in the 1x_MODE_LANE0 state immediately before entering the 1x_RECOVERY state. The port reenters the 1x_MODE_LANE2 state if lane_ready[2] is reasserted and the port was in the 1x_MODE_LANE2 state immediately before entering the 1x_RECOVERY state. If both lane_sync[0] and lane_sync[2] are lost (both lane_sync[0] and lane_sync[R] de-asserted), the SILENT state in entered. To prevent that state machine from possibly being stuck in the 1x_RECOVERY state, if the appropriate lane_ready[ ] is not asserted before the discovery time is up (disc_tmr_done asserted), the SILENT state is entered.

The state machine does not support recovery from a 1x mode state to Nx_MODE or the other 1x mode without going through the SILENT state.

The input signals force_1x_mode and force_laneR allow the state of the machine to be forced during initialization into 1x mode, and in 1x mode to be forced to receive on lane 2.

The input signal force_reinit allows the port to force port n link re-initialization at any time.

The variable port_initialized is asserted only in the 1x_MODE_LANE0, 1x_MODE_LANE2 and Nx_MODE states.

NOTE:

The name and specified function of the state machine variable N_lanes_drvr_oe is potentially confusing. As specified, its assertion causes the drivers for all N lanes of an Nx link to be output enabled.

However, N_lanes_drvr_oe is only asserted when the state machine variable lanes02_drvr_oe is also asserted. (The assertion of lanes02_drvr_oe causes the drivers for lane 0 and 2 to be output enabled). As a consequence, the net effect of the assertion or de-assertion of N_lanes_drvr_oe is that the drivers of all of the N lanes except the lanes 0 and 2 are output enabled or disabled respectively. The operation of an implementation that uses lanes02_drvr_oe as the output enable for the seek lane drivers and N_lanes_drvr_oe as the output enable for the remaining N-2 lanes will be operationally indistinguishable from an implementation that uses (lanes02_drvr_oe OR N_lanes_drvr_oe) as the output enable for the seek lane drivers and N_lanes_drvr_oe as the output enable for the remaining N-2 lanes.

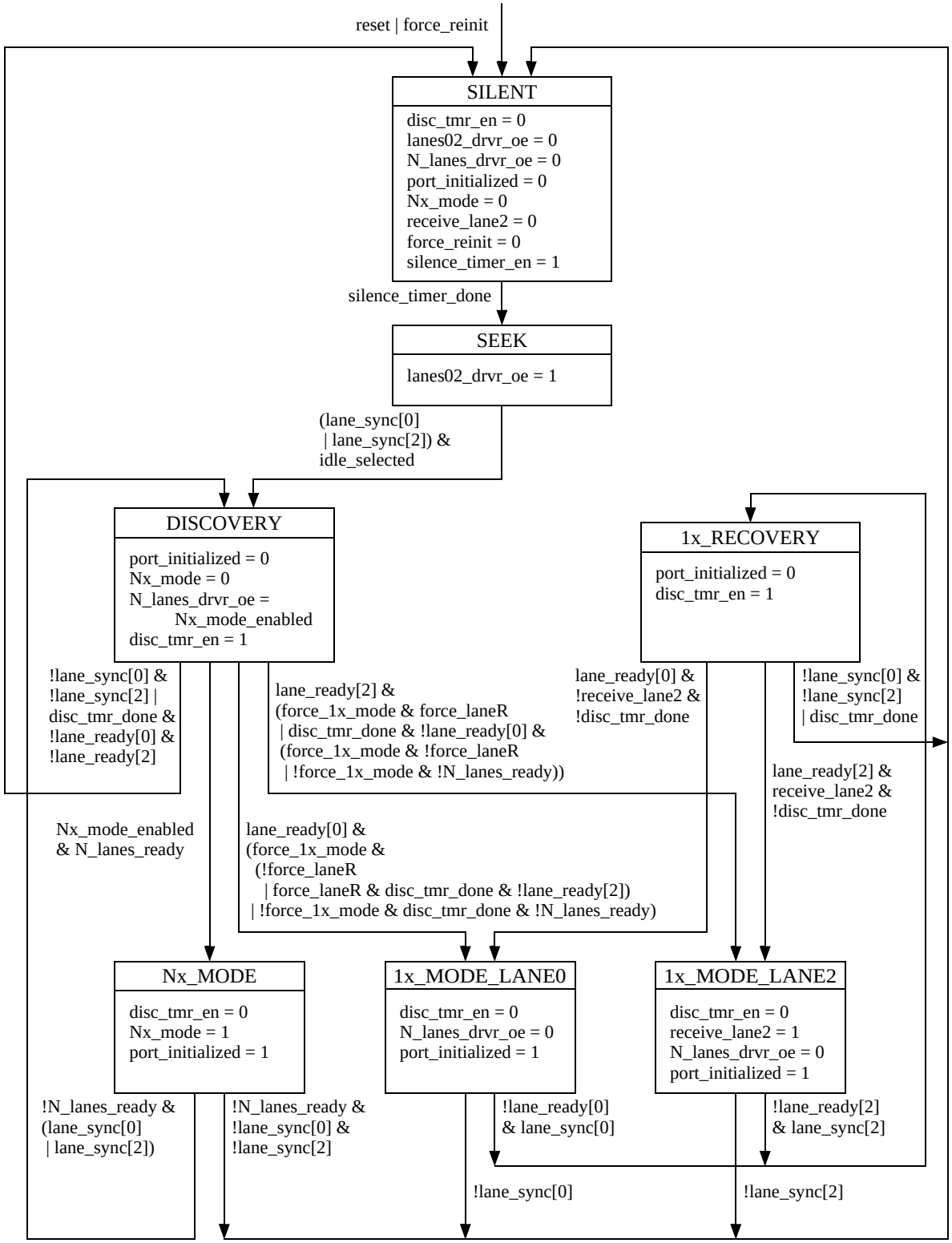


Figure 4-18. 1x/Nx_Initialization State Machine for N = 4, 8, 16

The following Initialization state machine may be used for 1x/4x ports that support only the IDLE1 idle sequence. The only difference between the 1x/Nx Initialization state machine of Figure 4-18 and the 1x/4x_Initialization state machine of Figure 4-19 is that the 1x/4x_Initialization machine does not have the 1x_RECOVERY state. As a consequence, the machines have different behavior when force_1x_mode is asserted. Unlike the 1x/Nx machine, the 1x/4x machine does not have a bias for the 1x_MODE_LANE0 state when force_1x_mode is not asserted.

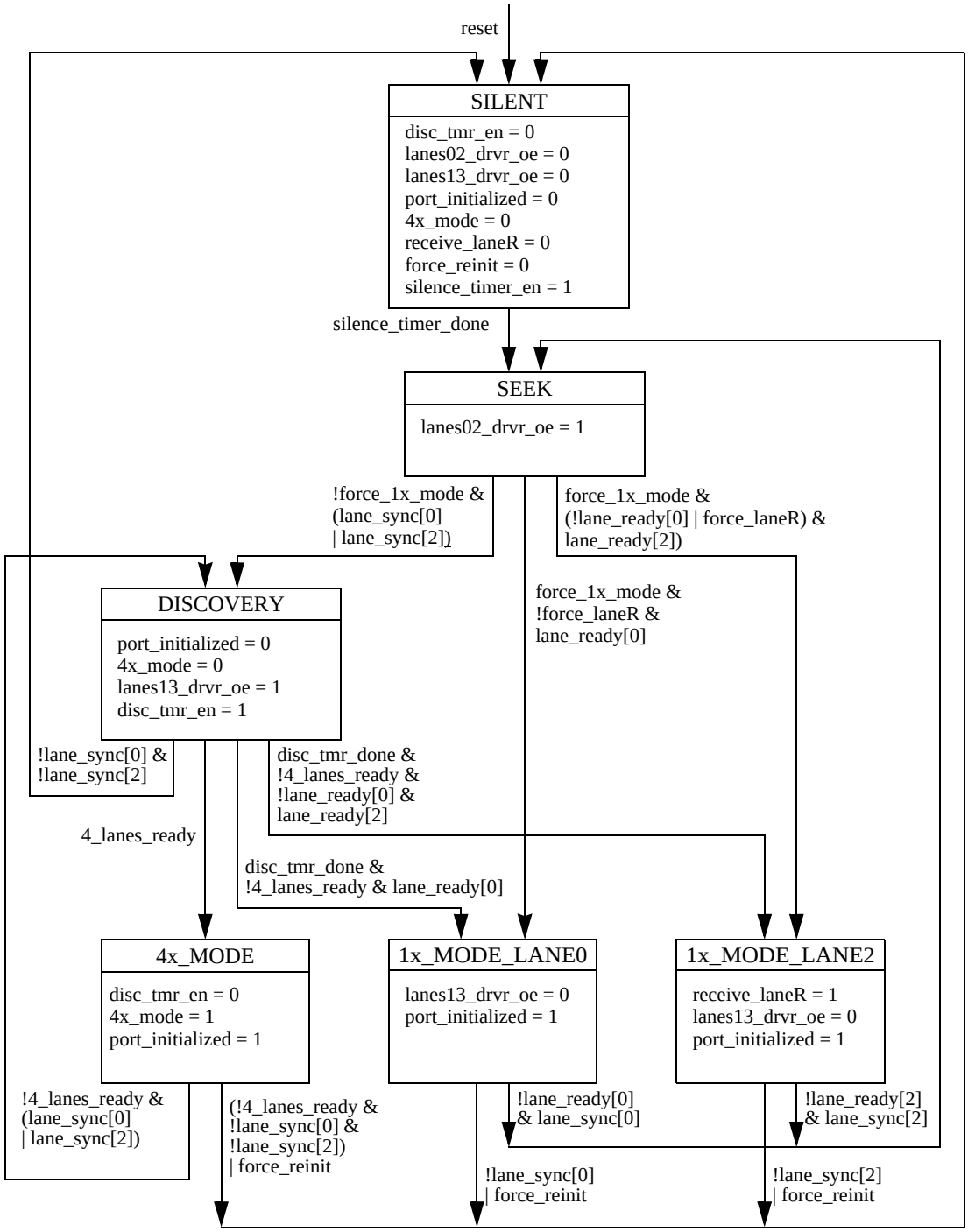


Figure 4-19. Alternate 1x/4x_Initialization State Machine

4.12.4.7 1x/2x Mode Initialization State Machine

The 1x/2x_Initialization state machine specified in this section shall be used by 1x/2x ports. Except for the method it uses to decide whether to operate in 1x or 2x

mode and the use of lane 1 as the redundancy lane, this state machine is identical to the 1x/Nx_Initialization state machine specified in Figure 4-18 with N = 2.

Ports that support more than 2 lanes disable all lanes except lanes 0 and R when operating in 1x mode. This allows the Initialization state machine for a port supporting more than 2 lanes to use the number of active lanes the port is receiving to determine whether to operate in 1x or Nx mode. 1x/2x ports transmit on both lanes regardless of whether they are operating in 1x or 2x mode. As a result, 1x/2x ports need a mechanism other than the number of active lanes being received to determine whether to operate in 1x or 2x mode. The 1x/2x_Mode_Detect state machine specified in Section 4.12.4.4 provides this mechanism.

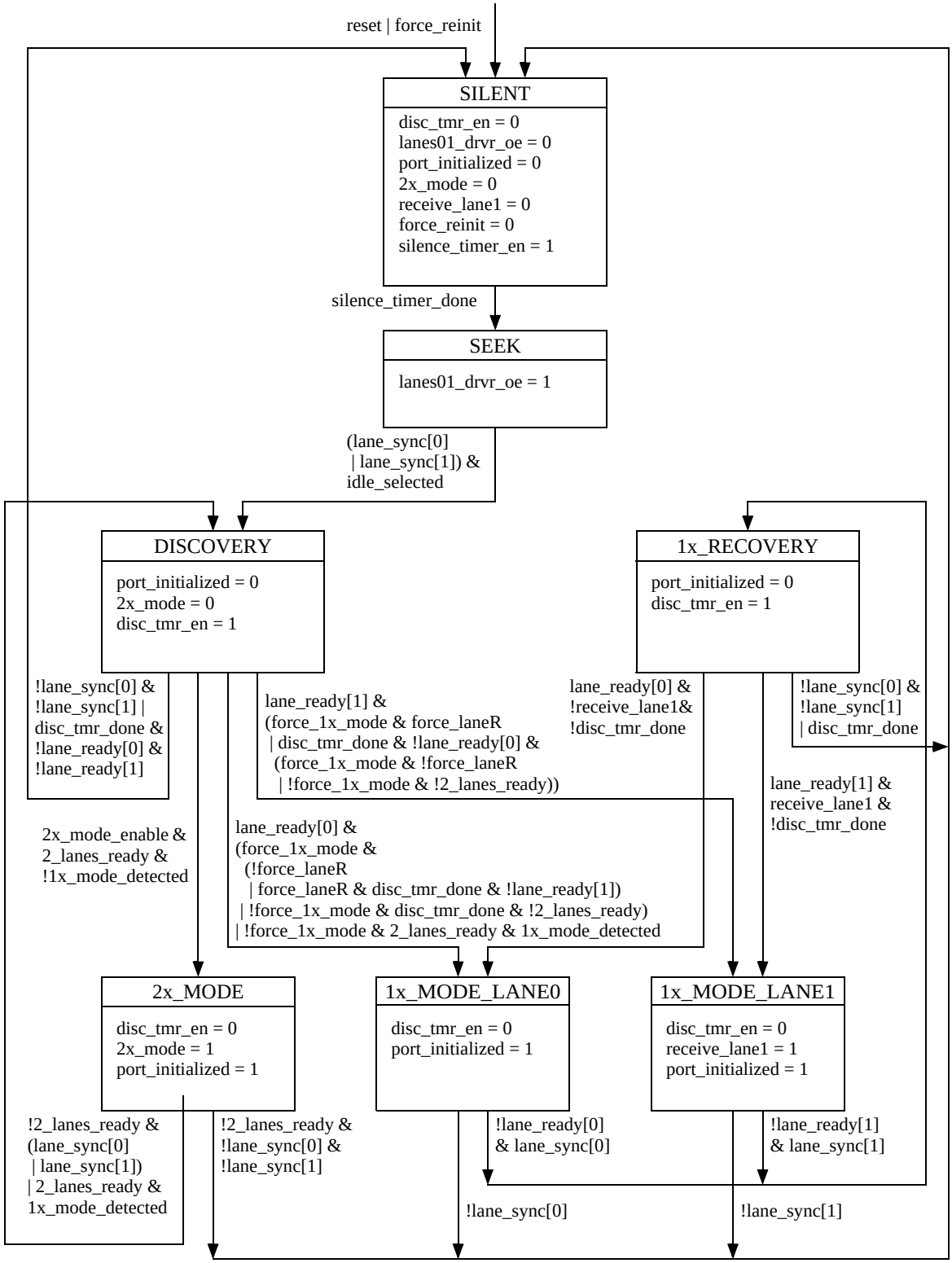


Figure 4-20. 1x/2x_Initialization State Machine

4.12.4.8 1x/Mx/Nx Mode Initialization State Machines

A Nx port may optionally support more than one multi-lane mode of operation. For example, an 8x port may support 4x mode in addition to the 8x mode and 1x modes. A port supporting more than one multi-lane mode is referred to as a 1x/Mx/ .... /Nx port where 1 < M < ... < N.

The initialization state machine for a port that supports multiple multi-lane modes of operation requires two or three additional states for each additional supported mode of multi-lane operation.

Like the 1x/Nx_Initialization state machine, the 1x/Mx/Nx_Initialization state machines support link width negotiation. The negotiation algorithm implemented by the state machine attempts to select the greatest link width supported by both ports of a connected port pair. However, once a link width is selected, a wider link width can be selected only if the state machine enters the SILENT state which restarts the selection algorithm.

4.12.4.8.1 1x/2x/Nx Initialization State Machine

The 1x/2x/Nx_Initialization state machine is specified in Figure 4-21 and shall be used by 1x/2x/Nx ports. Because the redundancy lane, lane R, differs for a 1x/2x port and a 1x/Nx port (N = 4, 8 or 16), the Initialization state machine for a 1x/2x/Nx port is the most complicated of the possible 1x/Mx/Nx_Initialization state machines.

The 1x/2x/Nx_Initialization state machine has three more states than a 1x/Nx_Initialization state machine, the 2x_MODE, 2x_RECOVERY and the 1x_MODE_LANE1 states

The operation of the 1x/2x/Nx_Initialization state machine is essentially the same as that of a 1x/2x_Initialization state machine for the 1x and 2x modes operation and that of a 1x/Nx_Initialization state machine for Nx mode operation. The differences between the 1x/2x/Nx_Initialization state machine and the 1x/2x_Initialization and 1x/Nx_Initialization state machines are as follows.

In the SEEK state, the lanes whose drivers are output enabled depend on the modes that are enabled. Lanes 0 and 1 are output enabled if the 2x mode is enabled. Lanes 0 and 2 are output enabled if the Nx mode is enabled or the 2x mode is disabled. And if both modes are enabled, lanes 0, 1 and 2 are output enabled. The state machine enters the DISCOVERY state when lane_sync is asserted for lanes 0, 1 or 2.

In the DISCOVERY state, the lane selection priority for 1x mode is lane 0 first, lane 2 second and lane 1 third. This priority is to bias the selection to lane 0 and to ensure that lane 2, not lane 1, is selected when 4x mode or wider is enabled in the connected port.

In the 2x_MODE state, the state machine transitions to the 2x_RECOVERY state if 1x_mode_detected is asserted. The state machine goes to the 2x_RECOVERY state rather than directly to the 1x_MODE_LANE0 state so that the port_initialized bit is

de-asserted indicating that the port is no longer in the normal operational state and that the link must be re-initialized before packet transmission can be resumed. Once in the 2x_RECOVERY state, the state machine then transitions to the 1x_MODE_LANE0 state if both 2_lanes_ready and 1x_mode_detected are still asserted.

The 2x_RECOVERY state is used to prevent the port from recovering to Nx mode once 2x mode has been selected.

In the 1x_MODE_LANE2 state, the state machine is allowed to transition to the 1x_MODE_LANE1 *state* via the 1x_RECOVERY state in the event that the connected port is a 1x/2x/Nx port and the connected port switches to 2x_MODE.

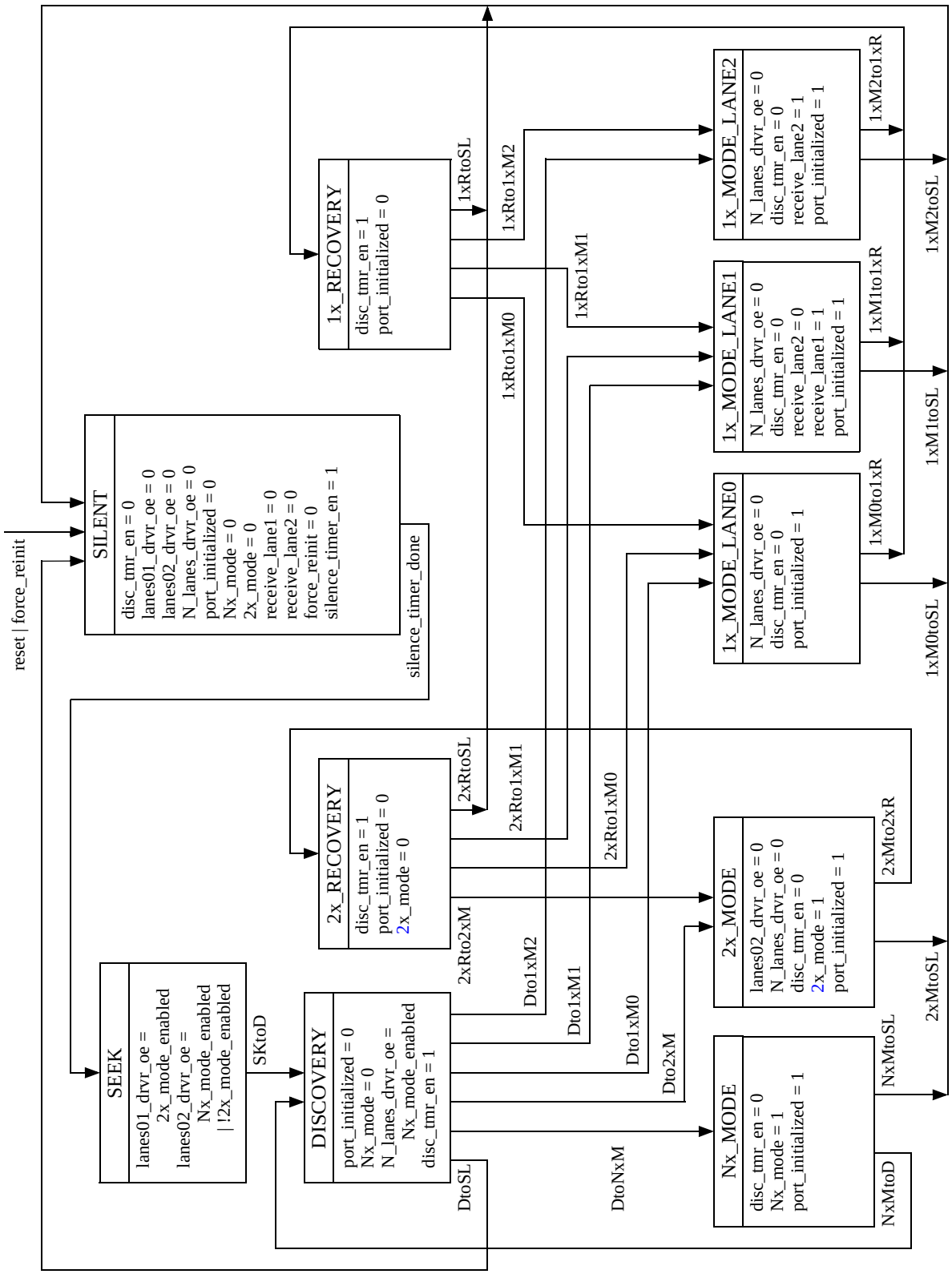


Figure 4-21. 1x/2x/Nx_Initialization State Machine

The variables that are local to the 1x/2x/Nx_Initialization state machine shown in Figure 4-21 are defined as follows.

```
1xM0to1xR = !lane_ready[0] & lane_sync[0]

1xM0toSL = !lane_sync[0]

1xM1to1xR = !lane_ready[1] & lane_sync[1]

1xM1toSL = !lane_sync[1]

1xM2to1xR = !lane_ready[2] & (lane_sync[1] | lane_sync[2])

1xM2toSL = !lane_sync[2] & !lane_sync[1]

1xRto1xM0 = !disc_tmr_done & !receive_lane1 & !receive_lane2 & lane_ready[0]

1xR to1xM1 = !disc_tmr_done &
             (receive_lane1 | receive_lane2 & !lane_ready[2]) & lane_ready[1]

1xRto1xM2 = !disc_tmr_done & receive_lane2 & lane_ready[2]

1xRtoSL = !lane_sync[0] & !lane_sync[1] & !lane_sync[2]
             | disc_tmr_done

2xMto2xR = !2_lanes_ready & (lane_sync[0] | lane_sync[1])
             | 2_lanes_ready & 1x_mode_detected

2xMtoSL = !lane_sync[0] & !lane_sync[1]

2xRto1xM0 = disc_tmr_done & !2_lanes_ready & lane_ready[0]
             | 2_lanes_ready & 1x_mode_detected

2xRto1xM1 = disc_tmr_done & !2_lanes_ready & !lane_ready[0] & lane_ready[1]

2xRto2xM = 2_lanes_ready & !1x_mode_detected

2xRtoSL = !lane_sync[0] & !lane_sync[1]
             | disc_tmr_done & !lane_ready[0] & !lane_ready[1]

Dto1xM0 = lane_ready[0] &
             (  force_1x_mode &
                (!force_laneR
                 | force_laneR & disc_tmr_done & !lane_ready[1] & !lane_ready[2]
                )
              | !force_1x_mode & disc_tmr_done &
                (!Nx_mode_enabled | !N_lanes_ready) &
                (!2x_mode_enabled | !2_lanes_ready)
             )
```

```
Dto1xM1 = disc_tmr_done & lane_ready[1] & !lane_ready[2] &
                ( force_1x_mode &
                  (force_laneR | !force_laneR & disc_tmr_done & !lane_ready[0])
                | !force_1x_mode & !lane_ready[0] &
                  (!Nx_mode_enabled | !N_lanes_ready) &
                  (!2x_mode_enabled | !2_lanes_ready)
                )

Dto1xM2 = lane_ready[2] &
                ( force_1x_mode &
                  (force_laneR | !force_laneR & disc_tmr_done & !lane_ready[0])
                | !force_1x_mode & disc_tmr_done & !lane_ready[0] &
                   (!Nx_mode_enabled | !N_lanes_ready) &
                   (!2x_mode_enabled | !2_lanes_ready)
                )

Dto2xM = 2x_mode_enabled & 2_lanes_ready &
                (!Nx_mode_enabled | disc_tmr_done & !N_lanes_ready)

DtoNxM = Nx_mode_enabled & N_lanes_ready

DtoSL = !lane_sync[0] & !lane_sync[1] & !lane_sync[2]
                | disc_tmr_done & !lane_ready[0] & !lane_ready[1] & !lane_ready[2]

NxMtoD = !N_lanes_ready & (lane_sync[0] | lane_sync[2])

NxMtoSL = !lane_sync[0] & !lane_sync[2]

SKtoD = (lane_sync[0] | lane_sync[1] | lane_sync[2]) & idle_selected
```

4.12.4.8.2 1x/Mx/Nx Initialization State Machine (N > M > 2)

The 1x/Mx/Nx_Initialization state machine for N > M > 2 is specified in Figure 4-22 and shall be used by 1x/Mx/Nx ports.

The 1x/Nx/Nx_Initialization state machine has two more states than a 1x/Nx_Initialization state machine, the Mx_MODE and Mx_RECOVERY states, but one less state than the 1x/2x/Nx_Initialization state machine, the 1x_MODE_LANE1 state. Its operation is most similar to that of the 1x/2x/Nx_Initialization state machine, but is less complex as the redundancy lane R is the same for all N and M > 2.

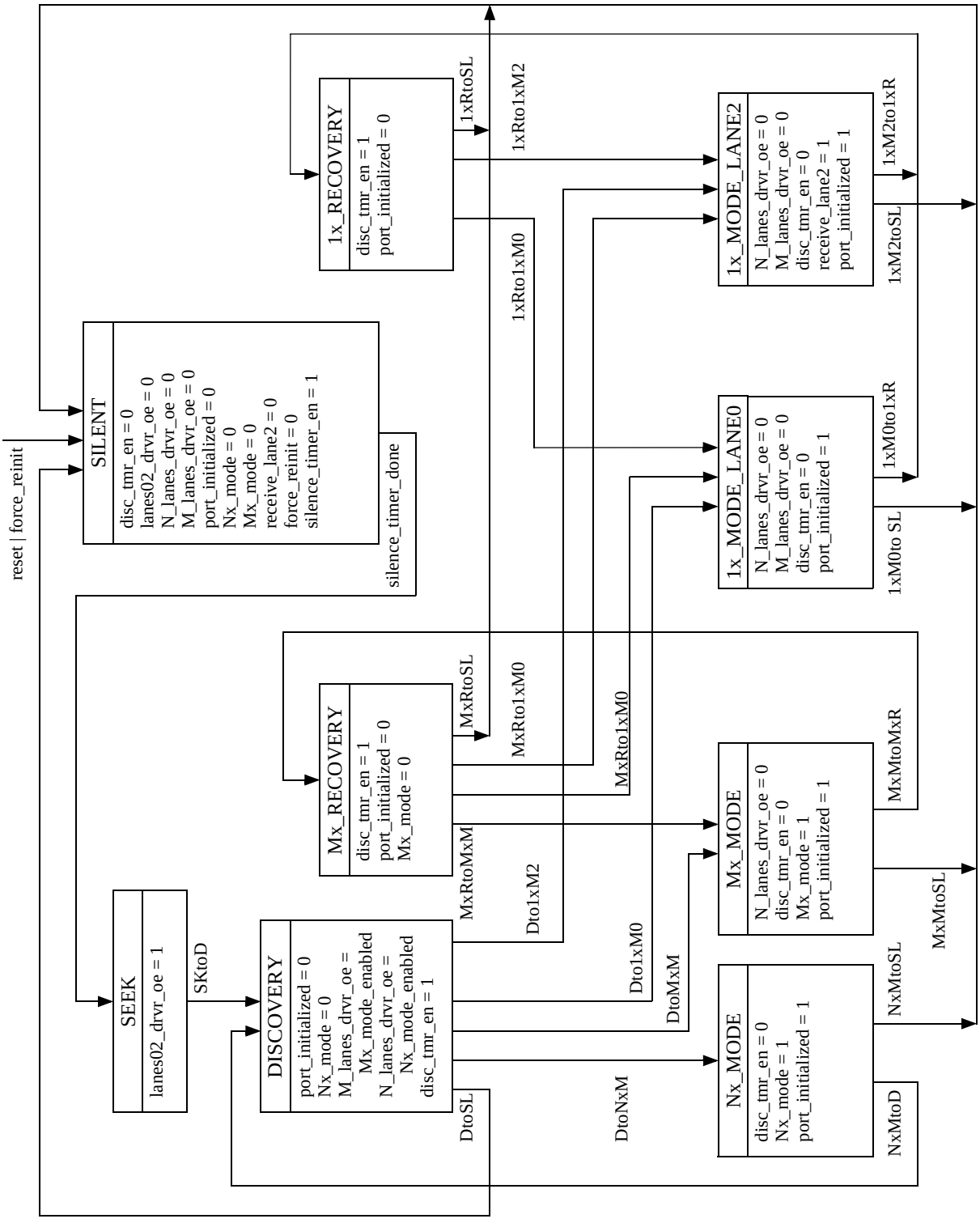


Figure 4-22. 1x/Mx/Nx_Initialization State Machine for N > M > 2

The variables that are local to the 1x/Mx/Nx_Initialization state machine shown in Figure 4-22 are defined as follows.

```
1xM0to1xR = !lane_ready[0] & lane_sync[0]

1xM0toSL = !lane_sync[0]

1xM2to1xR = !lane_ready[2] & lane_sync[2]

1xM2toSL = !lane_sync[2]

1xRto1xM0 = !disc_tmr_done & !receive_lane2 & lane_ready[0]

1xRto1xM2 = !disc_tmr_done & receive_lane2 & lane_ready[2]

1xRtoSL = !lane_sync[0] & !lane_sync[2]
                | disc_tmr_done

Dto1xM0 = lane_ready[0] &
                (  force_1x_mode &
                   (!force_laneR | force_laneR & disc_tmr_done & !lane_ready[2])
                  | !force_1x_mode & disc_tmr_done &
                    (!Nx_mode_enabled | !N_lanes_ready) &
                    (!Mx_mode_enabled | !M_lanes_ready)
                )

Dto1xM2 = lane_ready[2] &
                (  force_1x_mode &
                   (force_laneR | !force_laneR & disc_tmr_done & !lane_ready[0])
                  | !force_1x_mode & disc_tmr_done & !lane_ready[0] &
                    (!Nx_mode_enabled | !N_lanes_ready) &
                    (!Mx_mode_enabled | !M_lanes_ready)
                 )

DtoMxM = Mx_mode_enabled & M_lanes_ready &
                (!Nx_mode_enabled | disc_tmr_done & !N_lanes_ready)

DtoNxM = Nx_mode_enabled & N_lanes_ready

DtoSL = !lane_sync[0] & !lane_sync[2]
                | disc_tmr_done & !lane_ready[0] & !lane_ready[2]

MxMtoMxR = !M_lanes_ready & (lane_sync[0] | lane_sync[2])

MxMtoSL = !lane_sync[0] & !lane_sync[2]

MxRto1xM0 = disc_tmr_done & !M_lanes_ready & lane_ready[0]

MxRto1xM2 = disc_tmr_done & !M_lanes_ready & !lane_ready[0] & lane_ready[2]

MxRtoMxM = !disc_tmr_done & M_lanes_ready

MxRtoSL = !lane_sync[0] & !lane_sync[2]
                | disc_tmr_done & !lane_ready[0] & !lane_ready[2]

NxMtoD = !N_lanes_ready & (lane_sync[0] | lane_sync[2])

NxMtoSL = !lane_sync[0] & !lane_sync[2]
```

SKtoD = (lane_sync[0] | lane_sync[2]) & idle_selected

Chapter 5 LP-Serial Protocol

5.1 Introduction

This chapter specifies the LP-serial protocol which is the link level protocol for LP-serial links. The chapter covers traffic types, virtual channels (VCs), virtual channel queue management, packet priority, the mapping of transaction request flows onto packet priority, buffer management, and the use of control symbols in managing the delivery of packets between two devices connected by a LP-Serial link.

The protocol defines two types of traffic and provides a method for exchanging packets of each traffic type. The first type of traffic, called “reliable traffic” (RT), is the type of traffic RapidIO was originally designed to support. RT mode provides reliable delivery of packets between two devices that are connected by a RapidIO LP-Serial link. The second type of traffic, called “continuous traffic” (CT), provides unreliable delivery of packets that are “time sensitive”.

The protocol supports up to nine (9) virtual channels (VC0-VC8). Virtual Channel 0 (VC0) is always active and always operates in reliable traffic mode. It provides backward compatibility with Revision 1.3 RapidIO LP-Serial links. When only VC0 is active, a link is said to be operating in single VC mode. VCs 1-8 are optional, and if implemented, may be disabled for backward compatibility with Rev. 1.3 LP-Serial links.

5.2 Packet Exchange Protocol

As originally designed, the LP-Serial specification defines a protocol for devices connected by a LP-Serial link in which each packet transmitted by one device is acknowledged by control symbols transmitted by the other device. If a packet cannot be accepted for any reason, an acknowledgment control symbol indicates the reason and that the original packet and any transmitted subsequent packets must be resent. This behavior provides a flow control and error control mechanism between connected processing elements. This is the protocol for reliable traffic (RT).

Figure 5-1 shows an example of transporting a RT request and response packet pair across an interconnect fabric with acknowledgments between the link transmitter/receiver pairs along the way. This allows flow control and error handling to be managed between each electrically connected device pair rather than between the original source and final target of the packet. An end point device shall transmit

an acknowledgment control symbol for a request packet before transmitting the response packet corresponding to that request.

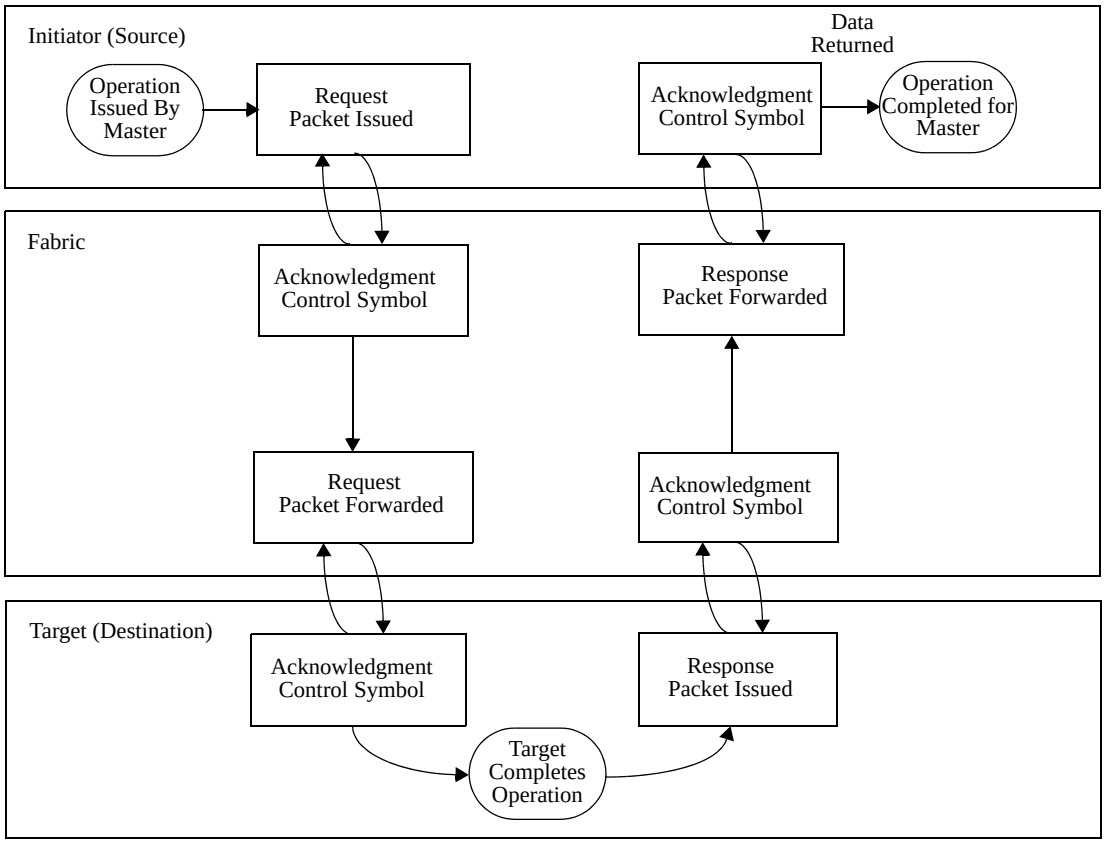


Figure 5-1. Example Transaction with Acknowledgment

The protocol for continuous traffic (CT) is very similar to the protocol for reliable traffic (RT). The primary differences are that some CT packets are not acknowledged and CT packets are subject to loss due to errors or insufficient buffer resources at the receiver.

5.3 Traffic types

The LP-Serial protocol provides support for transporting two types of traffic, “reliable traffic” (RT) and “continuous traffic” (CT). Reliable Traffic is guaranteed to be lossless by using packet retransmission whenever packet corruption is detected or receiver buffer resources are overrun. Continuous Traffic is subject to packet loss when packet corruption is detected or receiver buffer resources are overrun, but does not incur any additional latency, by continuing its packet flow without retransmission of unacknowledged packets.

5.4 Virtual Channels

Virtual channels provides a mechanism that allows the bandwidth of a link to be allocated amongst various unrelated “streams” and types of traffic in a manner that ensures that each stream, or group of streams, receives a guaranteed minimum fraction of the link bandwidth. This is done by allocating one or more streams of a given traffic type to each VC and then allocating each VC a specified fraction of the link bandwidth.

The LP-Serial protocol supports up to nine (9) virtual channels (VC0-VC8).

5.4.1 Virtual channel 0 (VC0)

VC0 shall be supported by all LP-Serial ports. VC0 shall always be active, operate in RT mode and support packet priority rules. VC0 provides the packet transport service specified in Rev. 1.3 of this specification and is backward compatible with Rev. 1.3.

5.4.2 Virtual Channels 1-8 (VC1-8)

Support for VC1 through VC8 by LP-Serial ports is optional. Any of VC1 through VC8 that are implemented shall support operation in RT mode and may optionally support and be configured for operation in CT mode. CT VCs operate independent of each other. RT VCs operate as a “RT Group”. That is to say, when the error recovery protocol is used to recover a damaged packet, the unacknowledged packets for all VCs in RT mode are retransmitted. See Section 5.13, "Error Detection and Recovery" for more on the error recovery process of RT and CT VCs.

The number of VCs implemented is up to the implementer. VC0 is always implemented. The number of channels for VCs 1-8 may be 0, 1, 2, 4, or 8. The binary configuration allows traffic to be combined (by ignoring bits in the VC field) in a predictable manner. Implementations with fewer than the full number of VCs should ignore, but must not modify, any ignored VC bits. That way traffic can fan back out into a larger set of VCs on subsequent links. The hierarchy for combining VCs is as follows:

Table 5-1. Additional VC Combinations

8 VCs	4 VCs	2 VCs	1 VC
VC1	VC1	VC1	VC1
VC2			
VC3	VC3		
VC4			
VC5	VC5	VC5	
VC6			
VC7	VC7		
VC8			

In systems implementing one or more of VCs 1 through 8 and in which the number of VCs 1 through 8 that are implemented varies from one LP-Serial link to another, care needs to exercised in assignment of VC numbers so that the desired RT or CT characteristic of a virtual channel is maintained as the channel passes from one link to another link that implements fewer virtual channels.

5.4.3 Virtual Channel Utilization

Packets are transmitted from one or more virtual channels according to the weighted distribution of bandwidth for each channel. The weighting is such that under demand for full utilization of the link's bandwidth, each active VC is guaranteed a certain portion of that bandwidth. This is the minimum that each VC can achieve. When the demand for bandwidth is less than the allocation for any VC, the extra bandwidth may be distributed among the other VCs giving them more than their allotment. The algorithm for scheduling traffic is up to the implementer as long as the rules (see Section 5.11, "Transaction and Packet Delivery Ordering Rules") are met.

There are no packet ordering guarantees between VCs. Packets within a VC in VCs 1 - 8 are equally weighted and must be kept in order. Only packets within VC0 have additional ordering rules based on priority.

5.5 Control Symbols

Control Symbols are the message elements used by ports connected by a LP-Serial link to manage all aspects of LP-Serial link operation. They are used for link maintenance, packet delimiting, packet acknowledgment, error reporting, and error recovery.

5.5.1 Control Symbol Selection

The control symbol used on a LP-Serial link is determined by the idle sequence being used on the link. Idle sequence selection occurs during the port initialization

process. If the link is operating with idle sequence 1 (IDLE1), the short control symbol shall be used. If the link is operating with idle sequence 2 (IDLE2), the long control symbol shall be used.

5.5.2 Control Symbol Delimiting

LP-Serial control symbols are delimited for transmission by 8B/10B special characters.

Short control symbols are delimited by a single 8B/10B special character that marks the beginning of the control symbol and immediately precedes the first character of the control symbol. Since control symbol length is constant and known, an end delimiters is neither needed nor provided.

Long control symbols are delimited by two 8B/10B special characters. The first special character marks the beginning of the control symbol (the start delimiter) and immediately precedes the first character of the control symbol. The second special character marks the end of the control symbol (the end delimiter) and immediately follows the last character of the control symbol. The end delimiter special character replicates the value of the start delimiter special character. The end delimiter is provided for error detection in a burst error environment.

One of two special characters is used to delimit a control symbol. If the control symbol contains a packet delimiter, the special character PD (K28.3) is used. If the control symbol does not contain a packet delimiter, the special character SC (K28.0) is used. This use of different special characters provides the receiver with an "early warning" of the content of the control symbol.

The control symbol delimiting special character(s) shall be added to the control symbol before the control symbol is passed to the PCS sublayer for 8B/10B encoding and, if applicable, lane striping.

The combination of a control symbol and its delimiting special character(s) is referred to as a “delimited control symbol”.

5.5.3 Control Symbol Use

5.5.3.1 Link Initialization

An LP-Serial port needs be initialized and the link to which it is connected also needs be initialized before the port may begin the normal transmission of packets and control symbols. The port is initialized when the port’s Initialization state machine variable port_initialized is asserted. The link is initialized after the port has successfully completed the following link initialization process and entered the link_initialized state (link_initialized variable asserted).

When a port is in the port_initialized state, but not in the link_initialized state, the port shall transmit only a idle sequences, status, VC-status, link-request and

link-response control symbols and, if IDLE2 is the idle sequence in use on the link, SYNC sequences.

After a LP-Serial port is initialized, the port shall complete the following sequence of actions to enter the link_initialized state (normal operational state).

1. The initialized port shall transmit idle and at least one status control symbol per 1024 code-groups transmitted per lane until the port has received an error free status control symbol from the connected port. The transmission of status control symbols indicates to the connected port that the port has completed initialization. The transmission of an idle sequence is required for the connected port to complete initialization.
2. After the initialized port has received an error free status control symbol from the connected port, the port shall transmit idle and at least 15 additional status control symbols. This group of control symbols may be sent more rapidly than the minimum rate of one status control symbol every 1024 code-groups transmitted per lane.
3. After the initialized port has received an error free status control symbol, the port shall wait until it has received a total of seven error free status control symbols with no intervening errors. This requirement provides a degree of link verification before packets and other control symbols are transmitted.
4. If any VC other than VC0 is implemented and enabled, the port shall transmit a single VC_Status control symbol for each such VC. This initializes the flow control status for each implemented and enabled VC other than VC0.
5. The port enters the link_initialized state.

Once a port is in the link_initialized state, loss of port initialization (port_initialized variable deasserted) shall cause the port to exit the link_initialized state (link_initialized variable deasserted). The link is then uninitialized from the point of view of that port. Once the port has exited the link_initialized state, the port shall not resume the normal transmission of packets and control symbols until the port has re-entered both the port_initialized and link_initialized states.

A port that is not in the port_initialized state shall ignore and discard any packet or control symbol that it receives from the connected port. A port that is in the port_initialized state but not in the link_initialized state shall ignore and discard any packet or any control symbol, other than status, VC-status, link-request or link-response control symbols, that it receives from the connected port.

A LP-Serial port shall not enter the Input error-stopped state or the Output error-stopped state unless the port is in the link_initialized state. The lost of link initialization (the state machine link_initialized variable is deasserted) shall not cause a port already in the Input error-stopped state or the Output error-stopped state to exit either of those states.

5.5.3.2 Buffer Status Maintenance

When a LP-Serial port is in the normal operational state, it shall transmit a control symbol containing the buf_status field for VC0 at least once every 1024 code-groups transmitted per lane. To comply with this requirement, the port shall transmit a status control symbol if no other control symbol containing the buf_status field for VC0 is available for transmission.

When a LP-Serial port is in the normal operational state and any VC other than VC0 is active (VCs 1-8), the port shall transmit a control symbol containing the buf_status field for each active VC at least once every VC refresh period. To comply with this requirement, the port shall transmit a VC_status control symbol for each active VC, other than VC0, if no other control symbol containing the buf_status field for that VC is available for transmission during the VC refresh interval. VC_status may be transmitted at any time, triggered by changes in VC conditions according to implementation specific algorithms.

The VC refresh period can be configured through the VC Refresh Interval register field defined in Chapter 6, "LP-Serial Registers". The shortest VC refresh period is 1024 code-groups and the longest required VC refreshing period is 1024 x 16 = 16K code groups. The VC refresh period must be implemented supporting 16K code groups.

NOTE:VC Refresh Period

The VC Refresh Interval register field contains space for up to 8 bits to be used, so based on implementation, the maximum refresh period may be 256K code groups. See section 6.4.2.1.

5.5.3.3 Embedded Control Symbols

Any control symbol that does not contain a packet delimiter may be embedded in a packet. An embedded control symbol may contain any defined encoding of stype0 and an stype1 encoding of "multicast-event" or "NOP". Control symbols with stype1 encodings of start-of-packet, end-of-packet, stomp, restart-from-retry, or link-request cannot be embedded as they would terminate the packet.

When a control symbol is embedded in a packet, the delimited control symbol shall begin on a 4-character boundary of the packet. That is, the number of packet characters between the end of the delimited start-of-packet control symbol and the start of the embedded delimited control symbol shall be a non-negative integer multiple of 4.

The manner and degree to which control symbol embedding is used on a link impacts both link and system performance. For example, embedding multicast-event control symbols allows their propagation delay and delay variation through switch processing elements to be minimized and is highly desirable for some multicast-event applications. Embedding packet acknowledgment control symbols reduces the delay in freeing packet buffers in the transmitting port which

can increase packet throughput and reduce packet propagation delay in some situations, which can be desirable. On the other hand, embedding all packet acknowledgment control symbols rather than combining as many of them as possible with packet delimiter control symbols reduces the link bandwidth available for packet transmission and may be undesirable.

5.5.3.4 Multicast-Event Control Symbols

The Multicast-Event control symbol provides a mechanism through which end points are notified that some system defined event has occurred. This event can be selectively multicast through the system. Refer to Section 3.5.6 for the format of the multicast-event control symbol.

When a switch processing element receives a Multicast-Event control symbol, the switch shall forward the Multicast-Event by issuing a Multicast-Event control symbol from each port that is designated in the port's CSR as a Multicast-Event output port. A switch port shall never forward a Multicast-Event control symbol back to the device from which it received a Multicast-Event control symbol regardless of whether the port is designated a Multicast-Event output or not.

It is intended that at any given time, Multicast-Event control symbols will be sourced by a single device. However, the source device can change (in case of failover, for example). In the event that two or more Multicast-Event control symbols are received by a switch processing element close enough in time that more than one is present in the switch at the same time, at least one of the Multicast-Event control symbols shall be forwarded. The others may be forwarded or discarded (device dependent).

The system defined event whose occurrence Multicast-Event gives notice of has no required temporal characteristics. It may occur randomly, periodically, or anything in between. For instance, Multicast-Event may be used for a heartbeat function or for a clock synchronization function in a multiprocessor system.

In an application such as clock synchronization in a multiprocessor system, both the propagation time of the notification through the system and the variation in propagation time from Multicast-Event to Multicast-Event are of concern. For these reasons and the need to multicast, control symbols are used to convey Multicast-Events as control symbols have the highest priority for transmission on a link and can be embedded in packets.

While this specification places no limits on Multicast-Event forwarding delay or forwarding delay variation, switch functions should be designed to minimize these characteristics. In addition, switch functions shall include in their specifications the maximum value of Multicast-Event forwarding delay (the maximum value of Multicast-Event forwarding delay through the switch) and the maximum value of Multicast-Event forwarding delay variation (the maximum value of Multicast-Event forwarding delay through the switch minus the minimum value of Multicast-Event forwarding delay through the switch).

5.6 Packets

5.6.1 Packet Delimiting

LP-Serial packets are delimited for transmission by control symbols. Since packet length is variable, both start-of-packet and end-of-packet delimiters are required. The start-of-packet delimiter immediately precedes the first character of the packet or an embedded delimited control symbol. The control symbol marking the end of a packet (packet termination) immediately follows the last character of the packet or the end of an embedded delimited control symbol.

The following control symbols are used to delimit packets.

- Start-of-packet
- End-of-packet
- Stomp
- Restart-from-retry
- Any link-request

5.6.1.1 Packet Start

The beginning of a packet (packet start) shall be marked by a start-of-packet control symbol.

5.6.1.2 Packet Termination

A packet shall be terminated in one of the following three ways:

- The end of a packet is marked with an end-of-packet control symbol.
- The end of a packet is marked with a start-of-packet control symbol that also marks the beginning of a new packet.
- The packet is canceled by a restart-from-retry, stomp, or any link-request control symbol

5.6.2 Acknowledgment Identifier

Each packet requires an identifier to uniquely identify its acknowledgment control symbol. This identifier, the acknowledge ID (ackID) is 5 bits long when using short control symbols and 6 bits long when using long control symbols. This allows up to 2^N outstanding unacknowledged request and/or response packets where N is the number of bits in the ackID field. To eliminate the ambiguity between 0 and 2^N outstanding packets, a maximum of 2^N-1 outstanding unacknowledged packets shall be allowed at any one time.

The value of ackID assigned to the first packet transmitted after a reset shall be 0. The values of ackID assigned to subsequent packets shall be in increasing numerical order, wrapping back to 0 on overflow. The ackID assigned to a

packet indicates the order of the packet transmission and is independent of the virtual channel assignment of the packet.

The acknowledgment control symbols are defined in Chapter 3, "Control Symbols". When acknowledgement control symbols are received containing VC specific information (e.g., buf_status), the transmitter side of the port must reassociate that information with the correct VC based on the returned ackID.

5.6.3 Packet Priority and Transaction Request Flows

Within VC0 each packet has a priority, and optionally a critical request flow, that is assigned by the end point processing element that is the source of (initiates) the packet. The priority is carried in the prio field of the packet and has four possible values: 0, 1, 2, or 3. Packet priority increases with the priority value with 0 being the lowest priority and 3 being the highest. Packet priority is used in RapidIO for several purposes which include transaction ordering and deadlock prevention. The critical request flow is carried in the CRF bit. It allows a flow to be designated as a critical or preferred flow with respect to other flows of the same priority. Support for critical request flows is strongly encouraged.

When a transaction is encapsulated in a packet for transmission, the transaction request flow indicator (flowID) of the transaction is mapped into the prio field (and optionally the CRF bit) of the packet. If the CRF bit is not supported, transaction request flows A and B are mapped to priorities 0 and 1 respectively and transaction request flows C and above are mapped to priority 2 as specified in Table below.

Table 5-2. VC0 Transaction Request Flow to Priority Mapping

Flow	System Priority	Request Packet Priority	Response Packet Priority
C or higher	Highest	2	3
B	Next	1	2 or 3
A	Lowest	0	1, 2, or 3

If the CRF bit is supported, the transaction request flows are mapped similarly as specified in Table 5-3 below. Devices that do not support the CRF bit treat it as reserved, setting it to logic 0 on transmit and ignoring it on receive.

Table 5-3. VC0 Transaction Request Flow to Priority and Critical Request Flow Mapping

Flow	System Priority	Request CRF Bit Setting	Request Packet Priority	Response CRF Bit Setting	Response Packet Priority
F or higher	Highest	1	2	1	3
E	Higher than A, B, C, D	0	2	0	3
D	Higher than A, B, C	1	1	1	2 or 3
C	Higher than A, B	0	1	0	2 or 3

Flow	System Priority	Request CRF Bit Setting	Request Packet Priority	Response CRF Bit Setting	Response Packet Priority
B	Higher than A	1	0	1	1, 2, or 3
A	Lowest	0	0	0	1, 2, or 3

The mapping of transaction request flows allows a RapidIO transport fabric to maintain transaction request flow ordering without the fabric having any knowledge of transaction types or their interdependencies. This allows a RapidIO fabric to be forward compatible as the types and functions of transactions evolve. A fabric can maintain transaction request flow ordering by simply maintaining the order of packets with the same priority and critical request flow for each path through the fabric and can maintain transaction request flow priority by never allowing a lower priority packet to pass a higher priority packet taking the same path through the fabric. In the case of congestion or some other restriction, a set CRF bit indicates that a flow of a priority can pass a flow of the same priority without the CRF bit set.

For VC0, flows identified as A - F (or higher) are synonymous with 0A - 0F, etc. Flows for VCs 1-8 (A and higher) are identified as 1A, 2A,...8A. All traffic in flows 1A-8A are transaction requests which do not require a response. Transaction requests that require responses, and their corresponding responses, must use VC0 with the appropriate priority.

Table 5-4. Flow IDs for VCs

Transaction Request Flow	VC	Transaction Request Flow	VC
1A and higher	VC1	5A and higher	VC5
2A and higher	VC2	6A and higher	VC6
3A and higher	VC3	7A and higher	VC7
4A and higher	VC4	8A and higher	VC8

5.7 Link Maintenance Protocol

The link maintenance protocol involves a request and response pair between ports connected by a LP-Serial link. For software management, the request is generated through ports in the configuration space of the sending device. An external host write of a command to the link-request register with an I/O logical specification maintenance write transaction causes a link-request control symbol to be issued onto the output port of the device, but only one link-request can be outstanding on a link at a time.

The device that is linked to the sending device shall respond with an link-response control symbol if the link-request command required it to do so. The external host retrieves the link-response by polling the link-response register with I/O logical maintenance read transactions. A device with multiple RapidIO interfaces has a link-request and a link-response register pair for each corresponding RapidIO interface.

The automatic error recovery mechanism relies on the hardware generating link-request/input-status control symbols under the transmission error conditions described in Section 5.13.2.1, "Recoverable Errors" and using the corresponding link-response information to attempt recovery.

Due to the undefined reliability of system designs, it is necessary to put a safety lockout on the reset function of the link-request/reset-device control symbol. A device receiving a link-request/reset-device control symbol shall not perform the reset function unless it has received four link-request/reset-device control symbols in a row without any intervening packets or other control symbols, except status control symbols. This will prevent spurious reset-device commands inadvertently resetting a device. The link-request/reset-device control symbol does not require a response.

The input-status command of the link-request/input-status control symbol is used by the hardware to recover from transmission errors. If the input port had stopped due to a transmission error that generated a packet-not-accepted control symbol back to the sender, the link-request/input-status control symbol acts as a link-request/restart-from-error control symbol, and the receiver is re-enabled to receive new packets after generating the link-response control symbol. The link-request/input-status control symbol may also be used to restart the receiving device if it is waiting for a restart-from-retry control symbol after retrying a packet. This situation can occur if transmission errors are encountered while trying to resynchronize the sending and receiving devices after the retry.

The link-request/input-status control symbol requires a response. A port receiving a link-request/input-status control symbol returns a link-response control symbol containing two pieces of information:

- port_status
- ackID_status

These status indicators are described in Table 3-6.

The retry-stopped state indicates that the port has retried a packet and is waiting to be restarted. This state is cleared when a restart-from-retry (or a link-request/input-status) control symbol is received. The error-stopped state indicates that the port has encountered a transmission error and is waiting to be restarted. This state is cleared when a link-request/input-status control symbol is received.

5.8 Packet Transmission Protocol

The LP-Serial protocol for packet transmission provides link level flow and error detection and recovery.

The protocol uses control symbols to delimit packets when they are transmitted across a LP-Serial link as specified in Section 5.6.1, "Packet Delimiting".

The link protocol uses acknowledgment to monitor packet transmission. With one exception, each packet transmitted across a LP-Serial link shall be acknowledged by the receiving port with a packet acknowledgment control symbol. Packets shall be acknowledged in the order in which they were transmitted (ackID order). The exception is when an event has occurred that caused a port to enter the Input Error-stopped state. CT mode packets accepted by a port after the port entered the Input Error-stopped state and before the port receives a link-request/input-status control symbol shall not be acknowledged.

To associate packet acknowledgment control symbols with transmitted packets, each packet shall be assigned an ackID value according to the rules of Section 5.6.2, "Acknowledgment Identifier" that is carried in the ackID field of the packet and the packet_ackID field of the associated acknowledgment control symbol. The ackID value carried by a packet indicates its order of transmission and the order in which it is acknowledged.

The LP-Serial link RT protocol uses retransmission to recover from packet transmission errors or a lack of receive buffer resources. To enable packet retransmission, a copy of each RT packet transmitted across a LP-Serial link shall be kept by the sending port until either a packet-accepted control symbol is received for the packet or the sending port determines that the packet has encountered an unrecoverable error condition.

The LP-Serial link CT protocol does not use packet retransmission. CT mode packets that are corrupted by transmission errors or that are not accepted because of a lack of receive buffer resources are discarded and lost. Therefore, a port need not retain a copy of a CT mode packet whose transmission has been completed.

The LP-Serial link protocol uses the ackID value carried in each packet to ensure that no RT mode packets are lost due to transmission errors. With one exception, a port shall accept packets from a LP-Serial link only in sequential ackID order, i.e. if the ackID value of the last packet accepted was N, the ackID value of the next packet that is accepted must be (N+1) modulo2^n where n is the number of bits in the ackID field. The exception is when an event has occurred that caused a port to enter the Input Error-stopped state. A CT mode packet received by a port after the port entered the Input Error-stopped state and before the port receives a link-request/input-status control symbol shall be accepted by the port without regard to the value of the packet's ackID field if the packet is otherwise error free and there are adequate receive buffer resources to accept the packet. The value that is maintained by the port of the ackID expected in the next packet shall not be changed by the acceptance of CT packets during this period.

A LP-Serial port accepts or rejects each error free packet that it receives with the expected ackID depending on whether the port has input buffer space available for the VC and/or priority level of the packet. The use of the packet-accepted, packet-retry, packet-not-accepted and restart-from-retry control symbols and the

buf_status field in packet acknowledgment control symbols to control the flow of packets across a LP-Serial link is covered in Section 5.9, "Flow Control".

The LP-Serial link protocol allows a packet that is being transmitted to be canceled at any point during its transmission. Packet cancellation is covered in Section 5.10, "Canceling Packets".

The LP-Serial link protocol provides detection and recovery processes for both transmission errors and protocol violations. The enumeration of detectable errors, the detection of errors and the associated error recovery processes are covered in Section 5.13, "Error Detection and Recovery".

In order to prevent switch processing element internal errors, such as SRAM soft bit errors, from silently corrupting a packet and the system, switch processing elements shall maintain packet error detection coverage while a packet is passing through the switch. The simplest method for maintaining packet error detection coverage is to pass the packet CRC through the switch as part of the packet. This works well for all non-maintenance packets whose CRC does not change as the packets are transported from source to destination through the fabric. Maintaining error detection coverage is more complicated for maintenance packets as their hop_count and CRC change every time they pass through a switch. However, passing the packet CRC through the switch as part of the packet does not prevent packet loss due to soft errors within the switch. Recovery from soft errors within a switch requires that each packet passing through the switch be covered by some type of error correction of adequate strength.

In order to support transaction ordering requirements of the logical layer specifications, the LP-Serial protocol imposes packet delivery ordering requirements within the physical layer and transaction delivery ordering requirements between the physical layer and the transport layer in end point processing elements. These requirements are covered in Section 5.11, "Transaction and Packet Delivery Ordering Rules".

In order to prevent deadlock, the LP-Serial protocol imposes a set of deadlock prevention rules. These rules are covered in Section 5.12, "Deadlock Avoidance".

This specification provides both bandwidth reservation and priority based channels. Priority scheduling may or may not be included in the reservation of bandwidth. Whatever allocation of bandwidth is used for priority traffic, higher level flows will reduce the bandwidth available for lower level flows. It is possible that traffic associated with higher flow levels can starve traffic associated with lower flow levels. It is important to use the available flows properly for the transaction type, to insure the rules in Section 5.11, "Transaction and Packet Delivery Ordering Rules" and Section 5.12, "Deadlock Avoidance" are met. The actual mechanisms used to schedule traffic are beyond the scope of this specification.

5.9 Flow Control

This section defines RapidIO LP-Serial link level flow control. The flow control operates between each pair of ports connected by a LP-Serial link. The purpose of link level flow control is to prevent the loss of packets due to a lack of buffer space in a link receiver.

The LP-Serial protocol defines two methods or modes of flow control. These are named receiver-controlled flow control and transmitter-controlled flow control. Every RapidIO LP-Serial port shall support receiver-controlled flow control. LP-Serial ports may optionally support transmitter-controlled flow control.

5.9.1 Receiver-Controlled Flow Control

Receiver-controlled flow control is the simplest and basic method of flow control. In this method, the input side of a port controls the flow of packets from its link partner by accepting or rejecting packets on a packet by packet basis. The receiving port provides no information to its link partner about the amount of buffer space it has available for packet reception.

As a result, its link partner transmits packets with no *a priori* expectation as to whether a given packet will be accepted or rejected. A port signals its link partner that it is operating in receiver-controlled flow control mode by setting the buf_status field to all 1's in every control symbol containing the field that the port transmits. This method is named receiver-controlled flow control because the receiver makes all of the decisions about how buffers in the receiver are allocated for packet reception.

A port operating in receiver-controlled flow control mode accepts or rejects each inbound error free packet based on whether the receiving port has enough buffer space available for the VC and the priority level of the packet. If there is enough buffer space available, the port accepts the packet and transmits a packet-accepted control symbol to its link partner that contains the ackID of the accepted packet in its packet_ackID field. This informs the port's link partner that the packet has been received without detected errors and that it has been accepted by the port. On receiving the packet-accepted control symbol, the link partner discards its copy of the accepted packet freeing buffer space in the partner.

The remaining behavior is a function of the mode of the VC.

5.9.1.1 Reliable Traffic VC Receivers

If buffer space is not available, the port rejects the packet. If multiple VCs are active, and the VC is in reliable traffic mode, the rejected packet shall be acknowledged with the packet-not-accepted control symbol. The cause field of the control symbol should be set to "packet not accepted due to lack of resources". This causes the entire "RT Group" to go through the same process used in error recovery to resequence and

retransmit the RT packets. See Section 5.13, "Error Detection and Recovery" for details.

If the port is operating in single VC mode, the port may use the Packet Retry protocol described in Section 5.9.1.3, "Single VC Retry Protocol", or it may continue to use the packet-not-accepted protocol described above.

5.9.1.2 Continuous Traffic VC Receivers

If buffer space is not available, and the VC is in CT mode, the packet is acknowledged as accepted, and the packet is discarded. This preserves the order of the normal link response and does not impact performance. Receiver based flow control for CT channels will result in packet loss due to receiver overruns depending on bandwidth and buffering conditions. See Section 5.9.2, "Transmitter-Controlled Flow Control" for transmitter based flow control options.

5.9.1.3 Single VC Retry Protocol

When operating with a single VC (VC0), the receiver may use the retry protocol for handling receiver overruns. It is a requirement that implementers include this functionality in the channel design to be backward compatible with existing RapidIO interfaces.

When a port rejects a packet, it immediately enters the Input Retry-stopped state and follows the Input Retry-stopped recovery process specified in Section 5.9.1.4, "Input Retry-Stopped Recovery Process". As part of the Input Retry-stopped recovery process, the port sends a packet-retry control symbol to its link partner indicating that the packet whose ackID is in the packet_ackID field of the control symbol and all packets subsequently transmitted by the port have been discarded by the link partner and must all be retransmitted. The control symbol also indicates that the link partner is temporarily out of buffers for packets of priority less than or equal to the priority of the retried packet.

A port that receives a packet-retry control symbol immediately enters the Output Retry-stopped state and follows the Output Retry-stopped recovery process specified in Section 5.9.1.5, "Output Retry-Stopped Recovery Process". As part of the Output Retry-stopped recovery process, the port receiving the packet-retry control symbol sends a restart-from-retry control symbol which causes its link partner to exit the Input Retry-stopped state and resume packet reception. The ackID assigned to that first packet transmitted after the restart-from-retry control symbol is the ackID of the packet that was retried.

Figure 5-2 shows an example of single VC receiver-controlled flow control operation. In this example the transmitter is capable of sending packets faster than the receiver is able to absorb them. Once the transmitter has received a retry for a packet, the transmitter may elect to cancel any packet that is presently being transmitted since it will be discarded anyway. This makes bandwidth available for any higher priority packets that may be pending transmission.

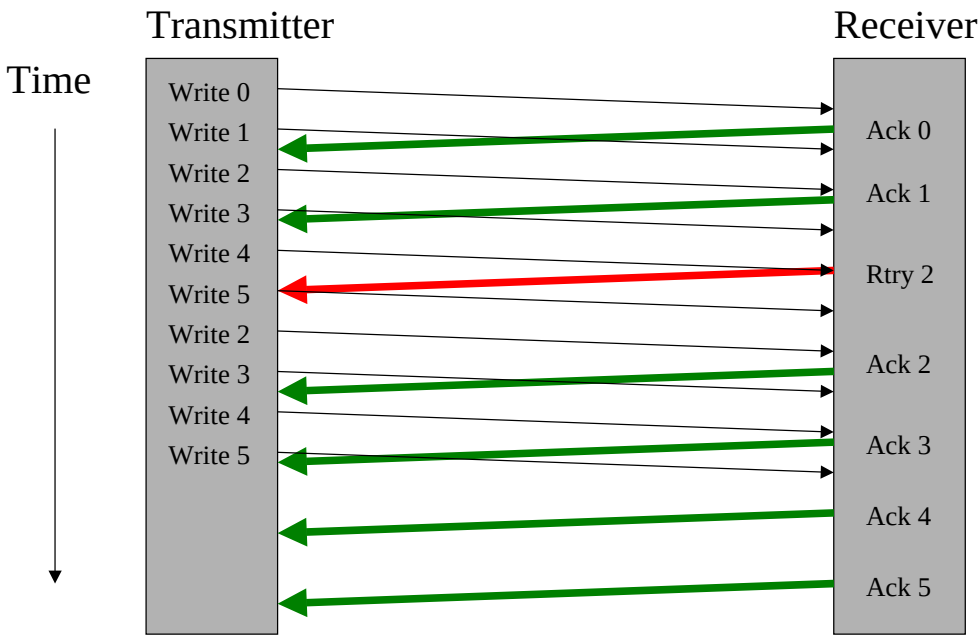


Figure 5-2. Single VC Mode Receiver-Controlled Flow Control

5.9.1.4 Input Retry-Stopped Recovery Process

When the input side of a port operating with only VC0 active (single VC mode) retries a packet, it immediately enters the Input Retry-stopped state. To recover from this state, the input side of the port takes the following actions.

- Discards the rejected or canceled packet without reporting a packet error and ignores all subsequently received packets while the port is in the Input Retry-stopped state.
- Causes the output side of the port to issue a packet-retry control symbol containing the ackID value of the retried packet in the packet_ackID field of the control symbol. (The packet-retry control symbol causes the output side of the link partner to enter the Output Retry-stopped state and send a restart-from-retry control symbol.)
- When a restart-from-retry control symbol is received, exit the Input Retry-stopped state and resume packet reception.

An example state machine with the behavior described in this section is included in Section C.2, "Packet Retry Mechanism".

5.9.1.5 Output Retry-Stopped Recovery Process

To recover from the Output Retry-stopped state, the output side of a port takes the following actions.

- Immediately stops transmitting new packets.

- Resets the link packet acknowledgment timers for all transmitted but unacknowledged packets. (This prevents the generation of spurious timeout errors.)
- Transmits a restart-from-retry control symbol.
- Backs up to the first unaccepted packet (the retried packet) which is the packet whose ackID value is specified by the packet_ackID value contained in the packet-retry control symbol. (The packet_ackID value is also the value of ackID field the port retrying the packet expects in the first packet it receives after receiving the restart-from-retry control symbol.)
- Exits the Output Retry-stopped state and resumes transmission with either the retried packet or a higher priority packet which is assigned the ackID value contained in the packet_ackID field of the packet-retry control symbol.

An example state machine with the behavior described in this section is included in Section C.2, "Packet Retry Mechanism".

5.9.2 Transmitter-Controlled Flow Control

In transmitter-controlled flow control, the receiving port provides information to its link partner about the amount of buffer space it has available for packet reception. With this information, the sending port can allocate the use of the receiving port's receive buffers according to the number and priority of packets that the sending port has waiting for transmission without concern that one or more of the packets shall be forced to retry.

A port signals its link partner that it is operating in transmitter-controlled flow control mode by setting the buf_status field to a value different from all 1's in every control symbol containing the field that the port transmits. This method is named transmitter-controlled flow control because the transmitter makes almost all of the decisions about how the buffers in the receiver are allocated for packet reception.

The number of free buffers that a port has available for packet reception is conveyed to its link partner by the value of the buf_status field in the control symbols that the port transmits. The value conveyed by the buf_status field is the number of maximum length packet buffers currently available for packet reception up to the limit that can be reported in the field. If a port has more buffers available than the maximum value that can be reported in the buf_status field, the port sets the field to that maximum value. A port may report a smaller number of buffers than it actually has available, but it shall not report a greater number.

A port informs its link partner when the number of free buffers available for packet reception changes. The new value of buf_status is conveyed in the buf_status field of a packet-accepted, packet-retry, status, or VC_status control symbol. Each change in the number of free buffers a port has available for packet reception need not be conveyed to the link partner. However, a port shall send a control symbol

containing the buf_status field to its link partner no less often than the minimum rate specified in Section 5.5.3.2, "Buffer Status Maintenance".

When a port implements more than VC0, the value of buf_status is kept on a per VC basis by the receiving port. When a packet-accepted symbol is returned, the buf_status field is filled with the status for the specific VC that the packet was sent to. When sending buf_status asynchronously (not in response to any specific packet), the status control symbol is used for VC0, and the VC_status control symbol is used for VC's 1-8.

A port whose link partner is operating in transmitter-control flow control mode should never receive a packet-not-accepted (or packet-retry control symbol if operating in single VC mode) from its link partner unless the port has transmitted more packets than its link partner has receive buffers, has violated the rules that all input buffers may not be filled with low priority packets or there is some fault condition. A receiver overrun is handled according to the rules in 5.9.1, "Receiver-Controlled Flow Control".

If a port, operating in single VC mode, for whose link partner is operating in transmitter-control flow control mode, receives a packet-retry control symbol, the output side of the port immediately enters the Output Retry-stopped state and follows the Output Retry-stopped recovery process specified in Section 5.9.1.5, "Output Retry-Stopped Recovery Process".

A simple example of single VC transmitter-controlled flow control is shown in Figure 5-3.

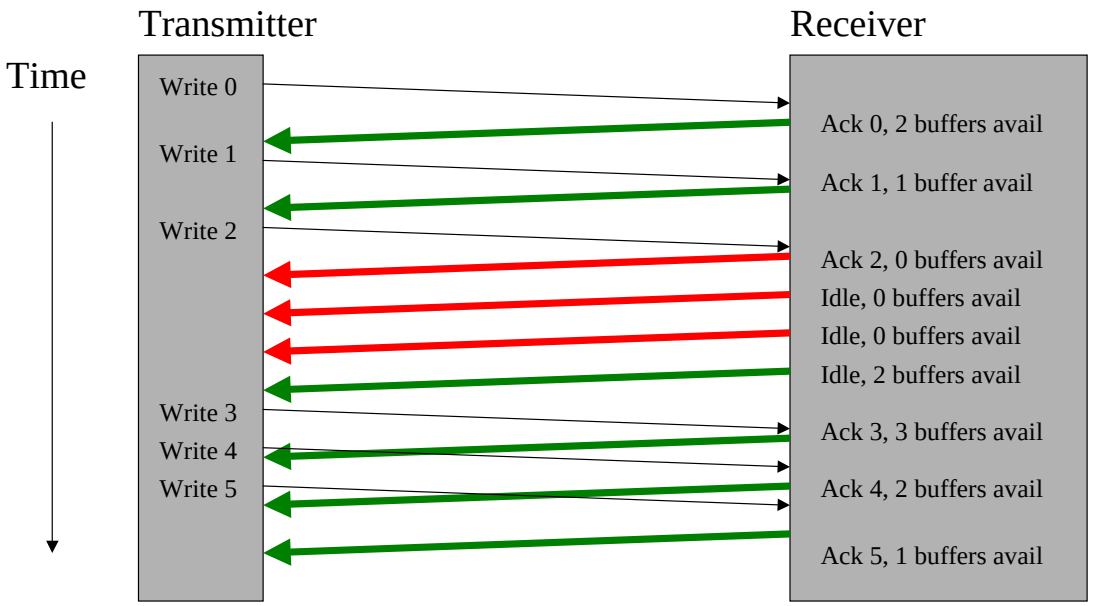


Figure 5-3. Single VC Mode Transmitter-Controlled Flow Control

5.9.2.1 Receive Buffer Management

In transmitter-controlled flow control, the transmitter manages the packet receive buffers in the receiver. This may be done in a number of ways, but the selected method shall not violate the rules in Section 5.12, "Deadlock Avoidance" concerning the acceptance of packets by ports.

For VC0, it is important to manage buffers in a way that reserves room for high priority packets. One possible implementation to organize the buffers is to establish watermarks and use them to progressively limit the packet priorities that can be transmitted as the effective number of free VC0 buffers in the receiver decreases. For example, VC0 has four priority levels. Three non-zero watermarks are needed to progressively limit the packet priorities that may be transmitted as the effective number of free VC0 buffers decreases. Designate the three watermarks as WM0, WM1, and WM2 where WM0 > WM1 > WM2 > 0 and employ the following rules.

If $free\_buffer\_count_0$ >= WM0, all priority packets may be transmitted.

If WM0 > $free\_buffer\_count_0$ >= WM1, only priority 1, 2, and 3 packets may be transmitted.

If WM1 > $free\_buffer\_count_0$ >= WM2, only priority 2 and 3 packets may be transmitted.

If WM2 > $free\_buffer\_count_0$, only priority 3 packets may be transmitted.

If this method is implemented, the initial values of the watermarks may be set by the hardware at reset as follows.

WM0 = 4

WM1 = 3

WM2 = 2

These initial values may be modified by hardware or software. The modified watermark values shall be based on the number of free buffers reported in the buf_status field of status control symbols received by the port following link initialization and before the start of packet transmission.

The three watermark values and the number of free buffers reported in the buf_status field of status control symbols received by the port following link initialization and before the start of packet transmission may be stored in a CSR. Since the maximum value of each of these four items is 62, each will fit in an 8-bit field and all four will fit in a single 32-bit CSR. If the watermarks are software setable, the three watermark fields in the CSR should be writable. For the greatest flexibility, a watermark register should be provided for each port on a device.

For VCs 1-8, packets within the same VC are equal in priority and always kept in order. The free buffers in the receiver can be partitioned between VCs in any number

of ways: they can be equally divided among the VCs, they can be statically partitioned based on the bandwidth allocation percentages, or they may be dynamically allocated from a larger pool. The only requirement is that once a given amount of buffers is reported by the receiver to the transmitter those buffers shall remain available for packets for that VC. Buffers may be deallocated once they are used, by removing the data, but not reporting the buffer available to that VC. At that time, the buffer may be reallocated to another VC. The specific method of buffer allocation is beyond the scope of this specification.

5.9.2.2 Effective Number of Free Receive Buffers

The number of buffers available in a link partner for packet reception on a given VC is typically less than the value of the buf_status field most recently received for that VC from the link partner. The value in the buf_status field does not account for packets that have been transmitted by the VC but not acknowledged by its link partner. The variable $free\_buffer\_count_N$ is defined to be the effective number of free buffers available in the link partner for packet reception on VC N. The recommended way for a port to compute and maintain these "free buffer counts" is to implement the following rules.

1. Each active VC maintains a variable "$free\_buffer\_count_{VC}$" whose value shall be the effective number of free buffers available to that VC in the link partner for packet reception.
2. Each active VC maintains a variable "$outstanding\_packet\_count_{VC}$" whose value is number of packets that have been transmitted on that VC, but that have not been acknowledged by its link partner.
3. After link initialization and before the start of packet transmission,

```
If {[(control_symbol = short) & (received_buf_status < 31)] |
        [(control_symbol = long) & (received_buf_status < 63)]} {
        flow_control_mode = transmitter;
        free_buffer_count0 = received_buf_status0;
        outstanding_packet_count0 = 0;
        for VC 1 through 8 {
            free_buffer_countVC =
                        received_VC_buffer_statusVC
            outstanding_packet_countVC = 0
        }
}
else
        flow_control_mode = receiver;
```

4. When a status or VC_Status control symbol is received by the port,

```
free_buffer_countVC =
        received_buf_statusVC - outstanding_packet_countVC;
```

5. When a packet is transmitted by the VC,

   $outstanding\_packet\_count_{VC}$ = $outstanding\_packet\_count_{VC} + 1$

   $free\_buffer\_count_{VC} = free\_buffer\_count_{VC} - 1$

6. When a packet-accepted control symbol is received by the port indicating that a packet has been accepted by the link partner, the buf_status field of the control symbol is reassociated with the originating VC:

   $Outstanding\_packet\_count_{VC}$ = $Outstanding\_packet\_count_{VC} - 1$;

   $free\_buffer\_count_{VC}$ = $received\_buf\_status_{VC} - outstanding\_packet\_count_{VC}$;

7. When a packet-retry control symbol is received by the port indicating that a packet has been forced by the link partner to retry,

   $Outstanding\_packet\_count_0 = 0$;

   $free\_buffer\_count_0 = received\_buf\_status_0$;

8. When a packet-not-accepted control symbol is received by the port indicating that a packet has been rejected by the link partner because of one or more detected errors or a lack of buffer resources,

   $Outstanding\_packet\_count_{VC} = 0$;

   $free\_buffer\_count_{VC}$ = $free\_buffer\_count_{VC}$ (remains unchanged);

9. When a link-response control symbol is received,

   $free\_buffer\_count_0 = received\_buf\_status$;

5.9.2.3 Speculative Packet Transmission

A port whose link partner is operating in transmitter-controlled flow control mode may send more packets on a given VC than the number of free buffers indicated by the link partner as being available for that VC. Packets transmitted in excess of the free_buffer_count are transmitted on a speculative basis and are subject to retry by the link partner. The link partner accepts or rejects these packets on a packet by packet basis in exactly the same way it would if operating in receiver-controlled flow control mode. A port may use such speculative transmission in an attempt to maximize the utilization of the link. However, speculative transmission that results in a significant number of retries and discarded packets can reduce the effective bandwidth of the link.

When the link has multiple operating VCs, speculative packet transmission may increase the CT packet loss rate and how frequently the link runs the error-recovery process.

5.9.3 Flow Control Mode Negotiation

Immediately following the initialization of a link, each port begins sending status control symbols to its link partner. The value of the buf_status field in these control symbols indicates to the link partner the flow control mode supported by the sending port.

The flow control mode negotiation rule is as follows:

> If the port and its link partner both support transmitter-controlled flow control, then both ports shall use transmitter-controlled flow control. Otherwise, both ports shall use receiver-controlled flow control.

If multiple VCs are used, then a port shall have either all channels in receiver based flow control or all channels in transmitter based flow control. All status and VC_status control symbols shall be consistent in their buf_status reporting in this regard.

5.10 Canceling Packets

When a port becomes aware of some condition that will require the packet it is currently transmitting to be retransmitted, the port may cancel the packet. This allows the port to avoid wasting bandwidth by not completing the transmission of a packet that the port knows must be retransmitted. Alternatively, the sending port may choose to complete transmission of the packet normally.

A port may cancel a packet if the port detects a problem with the packet as it is being transmitted or if the port receives a packet-retry or packet-not-accepted control symbol for a packet that is still being transmitted or that was previously transmitted. A packet-retry or packet-not-accepted control symbol can be transmitted by a port for a packet at any time after the port begins receiving the packet.

The sending device shall use the stomp control symbol, the restart-from-retry control symbol (in response to a packet-retry control symbol), or any link request control symbol to cancel a packet.

A port receiving a canceled packet shall drop the packet. The cancellation of a packet shall not result in the generation or report of any errors. If the packet was canceled because the sender received a packet-not-accepted control symbol, the error that caused the packet-not-accepted to be sent shall be reported in the normal manner.

The behavior of a port that receives a canceled packet depends on the control symbol that canceled the packet. A port that is not in an input stopped state (Retry-stopped or Error-stopped) while receiving the canceled packet and has not previously acknowledged the packet shall have the following behavior.

> If the packet is canceled by a link-request/input-status control symbol, the port shall drop the packet without reporting a packet error.

If the packet is canceled by a restart-from-retry control symbol a protocol error has occurred and the port shall immediately enter the Input Error-stopped state and follows the Input Error-stopped recovery process specified in Section 5.13.2.6, "Input Error-Stopped Recovery Process".

If the packet was canceled by other than a restart-from-retry or link-request/input-status control symbol and the port is operating in single VC mode (only VC0 is active), the port shall immediately enter the Input Retry-Stopped state and follow the Input Retry-Stopped recovery process specified in Section 5.9.1.4, "Input Retry-Stopped Recovery Process". If the packet was canceled before the packet ackID field was received by the port, the packet_ackID field of the associated packet-retry control symbol acknowledging the packet shall be set to the ackID the port expected in the canceled packet

If the packet was canceled by other than a restart-from-retry or link-request/input-status control symbol and the port is operating in multiple VC mode (at least one of VC1-8 is active), the port shall immediately enter the Input Error-Stopped state and follow the Input Error-Stopped recovery process specified in Section 5.13.2.6, "Input Error-Stopped Recovery Process".

A packet whose transmission is canceled shall be considered to be an untransmitted packet.

5.11 Transaction and Packet Delivery Ordering Rules

The rules specified in this section are required for the physical layer to support the transaction ordering rules specified in the logical layer specifications.

Transaction Delivery Ordering Rules:

1. **The physical layer of an end point processing element port shall encapsulate in packets and forward to the RapidIO fabric transactions comprising a given transaction request flow in the same order that the transactions were received from the transport layer of the processing element.**
2. **The physical layer of an end point processing element port shall ensure that a higher priority request transaction that it receives from the transport layer of the processing element before a lower priority request transaction with the same sourceID and the same destinationID is forwarded to the fabric before the lower priority transaction.**
3. **The physical layer of an end point processing element port shall deliver transactions to the transport layer of the processing element in the same order that the packetized transactions were received by the port.**

Packet Delivery Ordering Rules:

1. **A packet initiated by a processing element shall not be considered committed to the RapidIO fabric and does not participate in the packet delivery ordering rules until the packet has been accepted by the device at the other end of the link. (RapidIO does not have the concept of delayed or deferred transactions. Once a packet is accepted into the fabric, it is committed.)**
2. **A switch shall not alter the priority, critical request flow or VC of a packet.**
3. **Packet forwarding decisions made by a switch processing element shall provide a consistent output port selection which is based solely on the value of the destinationID field carried in the packet.**
4. **A switch processing element shall not change the order of packets comprising a transaction request flow (packets with the same sourceID, the same destinationID, the same priority, same critical request flow, same VC bit, and ftype != 8) as the packets pass through the switch.**
5. **A switch processing element shall not allow lower priority non-maintenance packets (ftype != 8) to pass higher priority non-maintenance packets with the same sourceID and destinationID as the packets pass through the switch.**
6. **A switch processing element shall not allow a priority N maintenance packet (ftype = 8) to pass another maintenance packet of priority N or greater that takes the same path through the switch (same switch input port and same switch output port).**

Rules for Scheduling Among VCs:

The whole link bandwidth is evenly divided into 'N' portions and each portion is 1/N of the whole link bandwidth. Each VC is configured to have guaranteed bandwidth. The method among VCs is also vendor dependent, as long as it satisfies the following three rules:

1. If the total guaranteed bandwidth for all the supported VCs is more than 100%, then the received bandwidth for each supported VC cannot be guaranteed.
2. If the total guaranteed bandwidth for all the supported VCs is less than or equal to 100%, demand for more than its guaranteed bandwidth shall not cause any other VCs to receive less than their guaranteed bandwidth.
3. If one VC demands less bandwidth than its guaranteed bandwidth, the extra bandwidth may be distributed among other VCs.

If VC0 participates in the bandwidth reservation process, then all VCs will receive their expected minimum bandwidth. However, VC0 may be treated as a special case. VC0 may be treated with strict priority, getting whatever bandwidth is required when it has traffic to transport. In this condition, the remaining VCs will divide up

whatever portion of bandwidth remains. If VC0's utilization is significant, compared with the traffic on the other VCs, then the other VCs bandwidth will still be proportional to each other, but will vary as the available bandwidth is modified by VC0.

The implementer may also choose to implement some priorities within VC0 with strict priority, and schedule the rest with reserved bandwidth. This specification does not require any particular treatment as there are application cases for any of the above. Chapter 6, "LP-Serial Registers" defines a standard control register should the implementer decide to make this a programmable feature.

5.12 Deadlock Avoidance

Request transactions requiring responses shall only use VC0. The response packet shall only use VC0. The following requirements apply to prioritized traffic within VC0.

To allow a RapidIO protocol to evolve without changing the switching fabric, switch processing elements are not required, with the sole exception of ftype 8 maintenance transactions, to discern between packet types, their functions or their interdependencies. Switches, for instance, are not required to discern between packets carrying request transactions and packets carrying response transactions. As a result, it is possible for two end points, A and B to each fill all of their output buffers, the fabric connecting them and the other end point's input buffers with read requests. This would result in an input to output dependency loop in each end point in which there would be no buffer space to hold the responses necessary to complete any of the outstanding read requests.

To break input to output dependencies, end point processing elements must have the ability to issue outbound response packets even if outbound request packets awaiting transmission are congestion blocked by the connected device. Two techniques are provided to break input to output dependencies. First, a response packet (a packet carrying a response transaction) is always assigned an initial priority one priority level greater than the priority of the associated request packet (the packet carrying the associated request transaction).

This requirement is specified in Table 5-2 and Table 5-3. It breaks the dependency cycle at the request flow level. Second, the end point processing element that is the source of the response packet may additionally raise the priority of the response packet to a priority higher than the minimum required by Table 5-2 and Table 5-3 if necessary for the packet to be accepted by the connected device. This additional increase in response packet priority above the minimum required by Table 5-2 and Table 5-3 is called promotion. An end point processing element may promote a response packet only to the degree necessary for the packet to be accepted by the connected device.

The following rules define the deadlock prevention mechanism:

Deadlock Prevention Rules:

1. **A RapidIO fabric shall be dependency cycle free for all operations that do not require a response. (This rule is necessary as there are no mechanisms provided in the fabric to break dependency cycles for operations not requiring responses.)**
2. **A packet carrying a request transaction that requires a response shall not be issued at the highest priority. (This rule ensures that an end point processing element can issue a response packet at a priority higher than the priority of the associated request. This rule in combination with rule 3 are basis for the priority assignments in Table 5-2 and Table 5-3)**
3. **A packet carrying a response shall have a priority at least one priority level higher than the priority of the associated request. (This rule in combination with rule 2 are basis for the priority assignments in Table 5-2 and Table 5-3)**
4. **A switch processing element port shall accept an error-free packet of priority N if there is no packet of priority greater than or equal to N that was previously received by the port and is still waiting in the switch to be forwarded. (This rule has multiple implications which include but are not limited to the following. First, a switch processing element port must have at least as many maximum length packet input buffers as there are priority levels. Second, a minimum of one maximum length packet input buffer must be reserved for each priority level. A input buffer reserved for priority N might be restricted to only priority N packets or might be allowed to hold packets of priority greater than or equal to N, either approach complies with the rule.)**
5. **A switch processing element port that transmits a priority N packet that is forced to retry by the connected device shall select a packet of priority greater than N, if one is available, for transmission. (This guarantees that packets of a given priority will not block higher priority packets.)**
6. **An end point processing element port shall accept an error-free packet of priority N if the port has enough space for the packet in the input buffer space of the port allocated for packets of priority N. (Lack of input buffer space is the only reason an end point may retry a packet.)**
7. **The decision of an end point processing element to accept or retry an error-free packet of priority N shall not depend on the ability of the end point to issue request packets of priority less than or equal to N from any of its ports. (This rule works in conjunction with rule 6. It prohibits a device's inability to issue packets of priority less than or equal to N, due to congestion in the connected device, from resulting in a lack of buffers to receive inbound packets of priority greater than or equal to N which in turn would result in packets of priority greater than or equal to N being forced to retry. The implications and some ways of complying with this rule are presented in the following paragraphs.)**

One implication of Rule 7 is that a port may not fill all of its buffers that can be used to hold packets awaiting transmission with packets carrying request transactions. If this situation was allowed to occur and the output was blocked due to congestion in the connected device, read transactions could not be processed (no place to put the response packet), input buffer space would become filled and all subsequent inbound request packets would be forced to retry violating Rule 7.

Another implication is that a port must have a way of preventing output blockage at priority less than or equal to N, due to congestion in the connected device, from resulting in a lack of input buffer space for inbound packets of priority greater than or equal to N. There are multiple ways of doing this.

One way is to provide a port with input buffer space for at least four maximum length packets and reserve input buffer space for higher priority packets in a manner similar to that required by Rule 4 for switches. In this case, output port blockage at priority less than or equal to N will not result is blocking inbound packets of priority greater than or equal to N as any responses packets they generate will be of priority greater than N which is not congestion blocked. The port must however have the ability to select packets of priority greater than N for transmission from the packets awaiting transmission. This approach does not require the use of response packet priority promotion.

A port can use the promotion mechanism to increase the priority of response packets until they are accepted by the connected device. This allows output buffer space containing response packets to be freed even though all request packets awaiting transmission are congestion blocked.

As an example, suppose an end point processing element has a blocked input port because all available resources are being used for a response packet that the processing element is trying to send. If the response packet is retried by the downstream processing element, raising the priority of the response packet until it is accepted allows the processing element’s input port to unblock so the system can make forward progress.

It should be noted that implementing response priority promotion in a device may help with its link partner’s input buffer congestion, not its own input buffer congestion. It should also be noted that response priority promotion may not be able to guarantee forward progress in the system unless the link partner has implemented priority based input buffer reservation.

5.13 Error Detection and Recovery

Error detection and recovery is becoming a more important issue for many systems. The LP-Serial specification provides extensive error detection and recovery by combining retry protocols with cyclic redundancy codes, the selection of delimiter special characters and response timers.

One feature of the error protection strategy is that with the sole exception of maintenance packets, the CRC value carried in a packet remains unchanged as the packet moves through the fabric. The CRC carried in a maintenance packet must be regenerated at each switch as the hop count changes.

5.13.1 Lost Packet Detection

Some types of errors, such as a lost request or response packet or a lost acknowledgment, result in a system with hung resources. To detect this type of error there shall be timeout counters that expire when sufficient time has elapsed without receiving the expected response from the system. Because the expiration of one of these timers should indicate to the system that there is a problem, this time interval should be set long enough so that a false timeout is not signaled. The response to this error condition is implementation dependent.

The RapidIO specifications require timeout counters for the physical layer, the port link timeout counters, and counters for the logical layer, the port response timeout counters. The interpretation of the counter values is implementation dependent, based on a number of factors including link clock rate, the internal clock rate of the device, and the desired system behavior.

The physical layer timeout occurs between the transmission of a packet and the receipt of an acknowledgment control symbol. This timeout interval is likely to be comparatively short because the packet and acknowledgment pair must only traverse a single link.

The logical layer timeout occurs between the issuance of a request packet that requires a response packet and the receipt of that response packet. This timeout is counted from the time that the logical layer issues the packet to the physical layer to the time that the associated response packet is delivered from the physical layer to the logical layer. Should the physical layer fail to complete the delivery of the packet, the logical layer timeout will occur. This timeout interval is likely to be comparatively long because the packet and response pair have to traverse the fabric at least twice and be processed by the target. Error handling for a response timeout is implementation dependent.

Certain GSM operations may require two response transactions, and both must be received for the operation to be considered complete. In the case of a device implementation with multiple links, one response packet may be returned on the same link where the operation was initiated and the other response packet may be returned on a different link. If this behavior is supported by the issuing processing element, the port response timeout implementation must look for both responses, regardless on which links they are returned.

5.13.2 Link Behavior Under Error

The LP-Serial link uses error detection and retransmission to protect RT packets against loss or corruption due to transmission errors. Transmission error detection is done at the input port, and all transmission error recovery is also initiated at the input port.

The packet transmission protocol requires that each RT packet transmitted by a port be acknowledged by the receiving port and that a port retain a copy of each RT packet that it transmits until the port receives a packet-accepted control symbol acknowledgment for the packet or the sending port determines that the packet has encountered an unrecoverable error. If the receiving port detects a transmission error in a packet, the port sends a packet-not-accepted control symbol acknowledgment back to the sender indicating that the packet was corrupted as received. After a link-request/input-status and link-response control symbol exchange, the sender begins retransmission with the next packet according to the priority/bandwidth scheduling rules. The RT VCs retransmit all packets that were unacknowledged at the time of the error. CT VCs continue with the next untransmitted packet.

All RT packets corrupted in transmission are retransmitted. The number of times a packet may be retransmitted before the sending port determines that the packet has encountered an unrecoverable condition is implementation dependent.

5.13.2.1 Recoverable Errors

The following five basic types of errors are detected by a LP-Serial port:

- An idle sequence error
- A control symbol error
- A packet error
- A column padding error
- A timeout waiting for an acknowledgment or link-response control symbol

5.13.2.2 Idle Sequence Errors

The detectable idle sequence errors depend on the idle sequence being used on the link. Links operating with short control symbols use the IDLE1 sequence. Links operating with long control symbols use the IDLE2 sequence.

To limit input port complexity, the port is not required to determine the specific error that resulted in an idle sequence error.

5.13.2.2.1 IDLE1 Sequence Errors

The IDLE1 sequence is comprised of A, K, and R (8B/10B special) characters. If an input port detects an invalid character or any valid character other then A, K, or R in an IDLE1 sequence and the port is not in the Input Error-stopped state, the port shall immediately enter the Input Error-stopped state and follow the Input Error-stopped

recovery process specified in Section 5.13.2.6, "Input Error-Stopped Recovery Process".

Following are several examples of idle sequence errors.

- A single bit transmission error can change an /A/, /K/, or /R/ code-group into a /Dx.y/ (data) code-group which is illegal in an idle sequence.
- A single bit transmission error can change an /A/, /K/, or /R/ code-group into an invalid code-group.
- A single bit transmission error can change an /SP/ or /PD/ (control symbol delimiters) into an invalid code-group.

5.13.2.2.2 IDLE2 Sequence Errors

The IDLE2 sequence is comprised of A, K, M and R special characters and data characters. If an input port detects any of the following errors in an IDLE2 sequence and the port is not in the Input Error-stopped state, the port shall immediately enter the Input Error-stopped state and follow the Input Error-stopped recovery process specified in Section 5.13.2.6, "Input Error-Stopped Recovery Process".

- An invalid character or any special character other than A, K, M or R
- After lane alignment is achieved,

  a column that contains an A, but is not all As,

  a column that contains a K, but is not all Ks,

  a column that contains a M, but is not all Ms,

  a column that contains a R, but is not all Rs or

  a column that contains a data character, but is not all data characters.

5.13.2.3 Control Symbol Errors

There are two types of detectable control symbol errors

- An uncorrupted control symbol that violates the link protocol
- A corrupted control symbol

5.13.2.3.1 Link Protocol Violations

The reception of a control symbol with no detected corruption that violates the link protocol shall cause the receiving port to immediately enter the appropriate Error-stopped state. Stype1 control symbol protocol errors shall cause the receiving port to immediately enter the Input Error-stopped state if not already in the Input Error-stopped state and follow the Input Error-stopped recovery process specified in Section 5.13.2.6, "Input Error-Stopped Recovery Process". Stype0 control symbol protocol errors shall cause the receiving port to immediately enter the Output Error-stopped state if not already in the Output Error-stopped state and follow the Output Error-stopped recovery process specified in Section 5.13.2.7, "Output Error-Stopped Recovery Process". If both stype0 and stype1 control symbols

contain protocol errors, then the receiving port shall enter both Error-stopped states and follow both error recovery processes.

Link protocol violations include the following:

- Unexpected packet-accepted, packet-retry, or packet-not-accepted control symbol
- Packet acknowledgment control symbol with an unexpected packet_ackID value
- Link timeout while waiting for an acknowledgment or link-response control symbol
- Receipt of a packet-retry symbol when operating in multi-VC mode

The following does not constitute a protocol violation:

- Receipt of a VC_status symbol when operating in single VC mode. Unexpected VC_status symbols are discarded.

The following is an example of a link protocol violation and recovery. A sender transmits RT mode packets labeled ackID 2, 3, 4, and 5. It receives acknowledgments for packets 2, 4, and 5, indicating a probable error associated with ackID 3. The sender then stops transmitting new packets and sends a link-request/input-status (restart-from-error) control symbol to the receiver. The receiver then returns a link- response control symbol indicating which packets it has received properly. These are the possible responses and the sender's resulting behavior:

- expecting ackID = 3 - sender must retransmit packets 3, 4, and 5
- expecting ackID = 4 - sender must retransmit packets 4 and 5
- expecting ackID = 5 - sender must retransmit packet 5
- expecting ackID = 6 - receiver got all packets, resume operation
- expecting ackID = anything else - fatal (non-recoverable) error

5.13.2.3.2 Corrupted Control symbols

The reception of a control symbol with detected corruption shall cause the receiving port to immediately enter the Input Error-stopped state if not already in the Input Error-stopped state and follow the Input Error-stopped recovery process specified in Section 5.13.2.6, "Input Error-Stopped Recovery Process".

Input ports detect the following types of control symbol corruption.

- A control symbol containing invalid characters or valid but non-data characters
- A control symbol with an incorrect CRC value
- A control symbol whose start delimiter (SC or PD) occurs in a lane whose lane_number mod4 != 0
- A long control symbol that does not have a end delimiter in the seventh character position after its start delimiter and with the same value as the start delimiter

5.13.2.4 Packet Errors

Each packet received by a port shall be checked for the following types of errors:

- Packet with an unexpected ackID value
- Packet with an incorrect CRC value
- Packet containing invalid characters or valid non-data characters
- Packet that exceeds the maximum packet size.

With one exception, the reception of a packet with any of the above errors shall cause the receiving port to immediately enter the Input Error-stopped state if not already in the Input Error-stopped state and follow the Input Error-stopped recovery process specified in Section 5.13.2.6, "Input Error-Stopped Recovery Process". The exception occurs when the link to which the port is connected is operating with the IDLE2 idle sequence, the packet in which one or more errors were detected was canceled by a link-request control symbol, and the only errors detected in the packet were the presence of one or more M special characters and may cause excessive packet length. In this case, the errors detected in the packet shall be ignored and the packet handled as a canceled packet as specified in Section 5.10, "Canceling Packets".

5.13.2.5 Link Timeout

A link timeout while waiting for an acknowledgment or link-response control symbol is handled as a link protocol violation as described in Section 5.13.2.3.1, "Link Protocol Violations".

5.13.2.6 Input Error-Stopped Recovery Process

When the input side of a port detects a transmission error, it immediately enters the Input Error-stopped state. To recover from this state, the input side of the port takes the following actions.

- Record the condition(s) that caused the port to enter the Input Error-stopped state.
- If an error(s) was detected in a control symbol or packet, ignore and discard the corrupted control symbol or packet.
- Cause the output side of the port to issue a packet-not-accepted control symbol. (The packet-not-accepted control symbol causes the output side of the receiving port to enter the Output Error-stopped state and send a link-request/input-status control symbol.)
- Subsequent to the event that caused the port to enter the Input Error-stopped state and prior to the reception of a link-request/input-status control symbol,

  discard without acknowledgement or error report all packets that are received for VCs operating in RT mode,

accept without acknowledgement (accept silently) all error free packets that are received for VCs operating in CT mode for which the VC specified in the packet has buffer space available and

discard without acknowledgement all packets that are received for VCs operating in CT mode which are not error free or for which the VC specified in the packet does not have buffer space available.

- When a link-request/input-status control symbol is received from the connected port, cause the output side of the port to transmit a link-response control symbol and if the transmitter-controlled flow control is in use on the link, to also transmit a VC_Status control symbol for each of VC1-8 that is active. The transmission of a VC_Status control symbol for each of VC1-8 that is active is optional if receiver-controlled flow control in use on the link. The input side of the port should also cause the output side of the port to transmit a status control symbol (for VC0). The input side of the port then exits the Input Error-stopped state and resumes normal packet reception. The actual transmission of the link-response, VC-status, and status control symbols may occur after the input side of the port exits the Input Error-stopped state and resumes normal packet reception.
- The transmission of the link-response, status and VC-status control symbols is subject to the following requirements.

  The link-response control symbol shall be transmitted either before any of the status and VC-status control symbols are transmitted or after all of the status and VC-status control symbols are transmitted.

  The status and VC-status control symbols that are transmitted shall be transmitted in the following order. If a status control symbol is transmitted it shall be transmitted first before any of the VC-status control symbols. Any VC-status control symbols that are transmitted shall be transmitted after the status control symbol and in order of increasing VCID.

  The link-response control symbol shall not be transmitted until the input side of the port is ready to resume packet reception and either the buffer consumption of all packets received by the port before the link-request/input-status control symbol has been determined or the port is able to maintain the distinction after packet reception resumes between packets received before the reception of the link-request/input-status control symbol and packets received after the reception of the link-request/input-status control symbol (as the processing of packets received before the link-request/input-status control symbol differs from the processing of packets received after the link-request/input-status control symbol).

  The status or VC-status control symbol for a VC operating in RT mode shall indicate the number of receive buffers available

for that VC inclusive of the buffer consumption of all packets received and accepted by the port for that VC before the event that caused the port entered the Input Error-stopped state.

The VC-status control symbol for a VC operating in CT mode shall indicate the number of receive buffers available for that VC inclusive of the buffer consumption of all packets received and accepted by the port for that VC before the link-request/input-status control symbol was received.

The status and VC-status control symbols shall be transmitted before any packet acknowledgment control symbols are transmitted for packets received after the link-request/input-status control symbol was received.

An example state machine with the behavior described in this section is included in Section C.3, "Error Recovery".

5.13.2.7 Output Error-Stopped Recovery Process

To recover from the Output Error-stopped state, the output side of a port takes the following actions.

- Immediately stops transmitting new packets.
- Resets the link packet acknowledgment timers for all transmitted but unacknowledged packets. (This prevents the generation of spurious timeout errors.)
- Transmits an input-status link-request/input-status (restart-from-error) control symbol. (The input status link-request/input-status control symbol causes the connected port to transmit a link-response control symbol that contains the input_status and ackID_status of the input side of the port. The ackID_status field contains the ackID value that is expected in the next packet that the connected port receives.)
- When the link-response is received, VCs operating in RT mode back up to the first unaccepted packet in each VC. VCs operating in CT mode silently assume the unacknowledged packets were accepted and adjust their state accordingly.
- The port exits the Output Error-stopped state and resumes transmission with the next RT or CT packet according to the bandwidth allocation algorithm using the ackID value contained in the link-response control symbol.

An example state machine with the behavior described in this section is included in Section C.3, "Error Recovery".

5.14 Power Management

Power management is currently beyond the scope of this specification and is implementation dependent. A device that supports power management features can

make these features accessible to the rest of the system using the device's local configuration registers.

Chapter 6 LP-Serial Registers

6.1 Introduction

This chapter describes the visible register set that allows an external processing element to determine the capabilities, configuration, and status of a processing element using this physical layer specification. This chapter only describes registers or register bits defined by this specification. Refer to the other RapidIO logical, transport, and physical specifications of interest to determine a complete list of registers and bit definitions.

There are four types of LP-Serial devices, an end point device, an end point device with additional software recovery registers, an end point free (or switch) device, and an end point free device with additional software recovery registers. Each has a different set of CSRs, specified in Section 6.5.1, Section 6.5.2, Section 6.5.3, and Section 6.5.4, respectively. All four device types have the same CARs, specified in Section 6.4.

Devices supporting Virtual Channels contain an additional register block for configuring VC support for each port. That block is added on after the above register block using a separate EF_PTR, as described in Section 6.8.

6.2 Register Map

These registers utilize the Extended Features blocks and can be accessed using *RapidIO Part 1: Input/Output Logical Specification* maintenance operations. Any register offsets not defined are considered reserved for this specification unless otherwise stated. Other registers required for a processing element are defined in other applicable RapidIO specifications and by the requirements of the specific device and are beyond the scope of this specification. Read and write accesses to reserved register offsets shall terminate normally and not cause an error condition in the target device.

The Extended Features pointer (EF_PTR) defined in the RapidIO logical specifications contains the offset of the first Extended Features block in the Extended Features data structure for a device. The LP-Serial physical features block may exist in any position in the Extended Features data structure and may exist in any portion of the Extended Features Space in the register address map for the device.

Register bits defined as reserved are considered reserved for this specification only. Bits that are reserved in this specification may be defined in another RapidIO specification.

Table 6-1. LP-Serial Register Map

Configuration Space Byte Offset	Register Name
0x0-F	Reserved
0x10-13	Processing Element Features CAR
0x14-0xFF	Reserved
0x100–FFFF	Extended Features Space
0x10000–FFFFFF	Implementation-defined Space

6.3 Reserved Register, Bit and Bit Field Value Behavior

Table 6-2 describes the required behavior for accesses to reserved register bits and reserved registers for the RapidIO register space,

Table 6-2. Configuration Space Reserved Access Behavior

Byte Offset	Space Name	Item	Initiator behavior	Target behavior
0x0–3F	Capability Register Space (CAR Space - this space is read-only)	Reserved bit	read - ignore returned value[1]	read - return logic 0
			write -	write - ignored
		Implementation-defined bit	read - ignore returned value unless implementation-defined function understood	read - return implementation-defined value
			write -	write - ignored
		Reserved register	read - ignore returned value	read - return logic 0s
			write -	write - ignored
0x40–FF	Command and Status Register Space (CSR Space)	Reserved bit	read - ignore returned value	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

Table 6-2. Configuration Space Reserved Access Behavior (Continued)

Byte Offset	Space Name	Item	Initiator behavior	Target behavior
0x100–FFFF	Extended Features Space	Reserved bit	read - ignore returned value	read - return logic 0
			write - preserve current value	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
0x10000–FFFFFF	Implementation-defined Space	Reserved bit and register	All behavior implementation-defined	

[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.

When a writable bit field is set to a reserved value, device behavior is implementation specific.

6.4 Capability Registers (CARs)

Every processing element shall contain a set of registers that allows an external processing element to determine its capabilities using the I/O logical maintenance read operation. All registers are 32 bits wide and are organized and accessed in 32-bit (4 byte) quantities, although some processing elements may optionally allow larger accesses. CARs are read-only. Refer to Table 6-2 for the required behavior for accesses to reserved registers and register bits.

CARs are big-endian with bit 0 the most significant bit.

6.4.1 Processing Element Features CAR (Configuration Space Offset 0x10)

The processing element features CAR identifies the major functionality provided by the processing element. The bit settings are shown in Table 6-3.

Table 6-3. Bit Settings for Processing Element Features CAR

Bits	Name	Description
0–3	—	Reserved
4	Multiport	The bit shall be implemented by devices that support the LP-Serial IDLE2 sequence, but is optional for devices that do not support the LP-Serial IDLE2 sequence. If this bit is not implemented it is Reserved. If this bit is implemented, the Switch Port Information CAR at Configuration Space Offset 0x14 (see *RapidIO Part 1: I/O Logical Specification*) shall be implemented regardless of the state of bit 3 of the Processing Element Features CAR. Indicates whether the PE implements multiple external RapidIO ports 0b0 - PE does not implement multiple external RapidIO ports 0b1 - PE implements multiple external RapidIO ports
5–24	—	Reserved
25	Implementation-defined	Implementation-defined
26	CRF Support	PE supports the Critical Request Flow (CRF) indicator 0b0 - Critical Request Flow is not supported 0b1 - Critical Request Flow is supported
27–31	—	Reserved

6.5 LP-Serial Extended Feature Blocks

This section describes the LP-Serial Extended Features Blocks. There is a separate Extended Features block for each of the following types of processing elements.

Generic end point devices

Generic end point devices, software assisted error recovery option

Generic end point free devices

Generic end point free devices, software assisted error recovery option

All registers in the LP-Serial Extended Feature Blocks are 32 bits in length and aligned to 32 bit boundaries. The details of the registers used are specified in Section 6.6.

6.5.1 Generic End Point Devices

This section specifies the LP-Serial registers comprising the LP-Serial Extended Features Block for a generic end point device. This Extended Features register block is assigned Extended Features block ID=0x0001.

Table 6-4 shows the register map of the RapidIO LP-Serial Extended Features Block for generic end point devices. The register map specifies the registers that comprise this Extended Features Block.

The Block Offset is the offset relative to the 16-bit Extended Features Pointer (EF_PTR) that points to the beginning of the block. The address of a byte in the block is calculated by adding the block byte offset to EF_PTR that points to the beginning of the block. This is denoted as [EF_PTR+xx] where xx is the block byte offset in hexadecimal.

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] through [EF_PTR + 0xBC]. Register map offset [EF_PTR + 0xC0] can be used for another Extended Features block.

Table 6-4. LP-Serial Register Map - Generic End Point Devices

	Block Byte Offset	Register Name
General	0x0	LP-Serial Register Block Header
	0x4–1C	Reserved
	0x20	Port Link Timeout Control CSR
	0x24	Port Response Timeout Control CSR
	0x28-38	Reserved
	0x3C	Port General Control CSR
Port 0	0x40-50	Reserved
	0x54	Port 0 Control 2 CSR
	0x58	Port 0 Error and Status CSR
	0x5C	Port 0 Control CSR
Port 1	0x60-70	Reserved
	0x74	Port 1 Control 2 CSR
	0x78	Port 1 Error and Status CSR
	0x7C	Port 1 Control CSR
Ports 2-14	0x80–218	Assigned to Port 2-14 CSRs
Port 15	0x220-230	Reserved
	0x234	Port 15 Control 2 CSR
	0x238	Port 15 Error and Status CSR
	0x23C	Port 15 Control CSR

The structure and use of the registers comprising the LP-Serial Generic End Point Extended Features Block is specified in Section 6.6.

The required behavior for accesses to reserved registers and register bits is specified in Table 6-2.

6.5.2 Generic End Point Devices, software assisted error recovery option

This section specifies the LP-Serial registers comprising the LP-Serial Extended Features Block for a generic end point device that supports software assisted error recovery. This is most useful for devices that for whatever reason do not want to implement error recovery in hardware and to allow software to generate link-request control symbols and see the results of the responses and for device debug. This Extended Features register block is assigned Extended Features block ID=0x0002.

Table 6-5 shows the register map for generic RapidIO LP-Serial end point devices with software assisted error recovery. The register map specifies the registers that comprise this Extended Features Block.

The Block Offset is the offset relative to the 16-bit Extended Features Pointer (EF_PTR) that points to the beginning of the block. The address of a byte in the block is calculated by adding the block byte offset to EF_PTR that points to the beginning of the block. This is denoted as [EF_PTR+xx] where xx is the block byte offset in hexadecimal.

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] through [EF_PTR + 0xBC]. Register map offset [EF_PTR + 0xC0] can be used for another Extended Features block.

Table 6-5. LP-Serial Register Map - Generic End Point Devices (SW assisted)

	Block Byte Offset	Register Name
General	0x0	LP-Serial Register Block Header
	0x4–1C	Reserved
	0x20	Port Link Timeout Control CSR
	0x24	Port Response Timeout Control CSR
	0x28-38	Reserved
	0x3C	Port General Control CSR
Port 0	0x40	Port 0 Link Maintenance Request CSR
	0x44	Port 0 Link Maintenance Response CSR
	0x48	Port 0 Local ackID Status CSR
	0x4C-50	Reserved
	0x54	Port 0 Control 2 CSR
	0x58	Port 0 Error and Status CSR
	0x5C	Port 0 Control CSR

Table 6-5. LP-Serial Register Map (Continued)- Generic End Point Devices (SW assisted)

	Block Byte Offset	Register Name
Port 1	0x60	Port 1 Link Maintenance Request CSR
	0x64	Port 1 Link Maintenance Response CSR
	0x68	Port 1 Local ackID Status CSR
	0x6C-70	Reserved
	0x74	Port 1 Control 2 CSR
	0x78	Port 1 Error and Status CSR
	0x7C	Port 1 Control CSR
Ports 2-14	0x80–218	Assigned to Port 2-14 CSRs
Port 15	0x220	Port 15 Link Maintenance Request CSR
	0x224	Port 15 Link Maintenance Response CSR
	0x228	Port 15 Local ackID Status CSR
	0x22C-230	Reserved
	0x234	Port 15 Control 2 CSR
	0x238	Port 15 Error and Status CSR
	0x23C	Port 15 Control CSR

The structure and use of the registers comprising the LP-Serial Generic End Point, software assisted error recovery option Extended Features Block is specified in Section 6.6.

The required behavior for accesses to reserved registers and register bits is specified in Table 6-2.

6.5.3 Generic End Point Free Devices

This section specifies the LP-Serial registers comprising the LP-Serial Extended Features Block for a generic device that does not contain end point functionality (i.e. switches). This Extended Features register block uses extended features block ID=0x0003.

Table 6-6 shows the register map for generic RapidIO LP-Serial end point-free devices. The register map specifies the registers that comprise this Extended Features Block.

The Block Offset is the offset relative to the 16-bit Extended Features Pointer (EF_PTR) that points to the beginning of the block. The address of a byte in the block is calculated by adding the block byte offset to EF_PTR that points to the beginning of the block. This is denoted as [EF_PTR+xx] where xx is the block byte offset in hexadecimal.

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] through [EF_PTR + 0xBC]. Register map offset [EF_PTR + 0xC0] can be used for another Extended Features block.

Table 6-6. LP-Serial Register Map - Generic End Point Free Devices

	Block Byte Offset	Register Name
General	0x0	LP-Serial Register Block Header
	0x4–1C	Reserved
	0x20	Port Link Timeout Control CSR
	0x24-38	Reserved
	0x3C	Port General Control CSR
Port 0	0x40-50	Reserved
	0x54	Port 0 Control 2 CSR
	0x58	Port 0 Error and Status CSR
	0x5C	Port 0 Control CSR
Port 1	0x60-70	Reserved
	0x74	Port 1 Control 2 CSR
	0x78	Port 1 Error and Status CSR
	0x7C	Port 1 Control CSR

Table 6-6. LP-Serial Register Map (Continued) - Generic End Point Free Devices

	Block Byte Offset	Register Name
Ports 2-14	0x80–218	Assigned to Port 2-14 CSRs
Port 15	0x220-230	Reserved
	0x234	Port 15 Control 2 CSR
	0x238	Port 15 Error and Status CSR
	0x23C	Port 15 Control CSR

The structure and use of the registers comprising the LP-Serial Generic End Point Free Extended Features Block is specified in Section 6.6.

The required behavior for accesses to reserved registers and register bits is specified in Table 6-2.

6.5.4 Generic End Point Free Devices, software assisted error recovery option

This section specifies the LP-Serial registers comprising the LP-Serial Extended Features Block for a generic device that does not contain end point functionality but that does support software assisted error recovery. Typically these devices are switches. This is most useful for devices that for whatever reason do not want to implement error recovery in hardware and to allow software to generate link-request control symbols and see the results of the responses and for device debug. This Extended Features register block is assigned Extended Features block ID=0x0009.

Table 6-7 shows the register map for generic RapidIO LP-Serial end point-free devices with software assisted error recovery. The register map specifies the registers that comprise this Extended Features Block.

The Block Offset is the offset relative to the 16-bit Extended Features Pointer (EF_PTR) that points to the beginning of the block. The address of a byte in the block is calculated by adding the block byte offset to EF_PTR that points to the beginning of the block. This is denoted as [EF_PTR+xx] where xx is the block byte offset in hexadecimal.

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] through [EF_PTR + 0xBC]. Register map offset [EF_PTR + 0xC0] can be used for another Extended Features block.

Table 6-7. LP-Serial Register Map - Generic End Point-free Devices (SW assisted)

	Block Byte Offset	Register Name
General	0x0	LP-Serial Register Block Header
	0x4–1C	Reserved
	0x20	Port Link Timeout Control CSR
	0x24-38	Reserved
	0x3C	Port General Control CSR
Port 0	0x40	Port 0 Link Maintenance Request CSR
	0x44	Port 0 Link Maintenance Response CSR
	0x48	Port 0 Local ackID Status CSR
	0x4C-50	Reserved
	0x54	Port 0 Control 2 CSR
	0x58	Port 0 Error and Status CSR
	0x5C	Port 0 Control CSR

Table 6-7. LP-Serial Register Map (Continued)- Generic End Point-free Devices (SW assisted)

	Block Byte Offset	Register Name
Port 1	0x60	Port 1 Link Maintenance Request CSR
	0x64	Port 1 Link Maintenance Response CSR
	0x68	Port 1 Local ackID Status CSR
	0x6C-70	Reserved
	0x74	Port 1 Control 2 CSR
	0x78	Port 1 Error and Status CSR
	0x7C	Port 1 Control CSR
Ports 2-14	0x80–218	Assigned to Port 2-14 CSRs
Port 15	0x220	Port 15 Link Maintenance Request CSR
	0x224	Port 15 Link Maintenance Response CSR
	0x228	Port 15 Local ackID Status CSR
	0x22C-230	Reserved
	0x234	Port 15 Control 2 CSR
	0x238	Port 15 Error and Status CSR
	0x23C	Port 15 Control CSR

The structure and use of the registers comprising the LP-Serial Generic End Point Free, software assisted error recovery option Extended Features Block is specified in Section 6.6.

The required behavior for accesses to reserved registers and register bits is specified in Table 6-2.

6.6 LP-Serial Command and Status Registers (CSRs)

All Command and Status registers are 32 bits in length and are aligned to 32 bit boundaries. All CSRs are accessed as 4 byte entities. CSRs are big endian with bit 0 the most significant bit.

Refer to Table 6-2 for the required behavior for accesses to reserved register bits.

6.6.1 LP-Serial Register Block Header (Block Offset 0x0)

The LP-Serial register block header register contains the EF_PTR to the next extended features block and the EF_ID that identifies LP-Serial Extended Feature Block for which this is the register block header.

Table 6-8. Bit Settings for LP-Serial Register 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		Hard wired Extended Features Block ID

6.6.2 Port Link Timeout Control CSR (Block Offset 0x20)

The port link timeout control register contains the timeout timer value for all ports on a device. This timeout is for link events such as sending a packet to receiving the corresponding acknowledge and sending a link-request to receiving the corresponding link-response. The reset value is the maximum timeout interval, and represents between 3 and 6 seconds.

Table 6-9. Bit Settings for Port Link Timeout Control CSR

Bit	Name	Reset Value	Description
0–23	timeout value	All 1s	timeout interval value
24-31	—		Reserved

6.6.3 Port Response Timeout Control CSR (Block Offset 0x24)

The port response timeout control register contains the timeout timer count for all ports on a device. This timeout is for sending a request packet to receiving the corresponding response packet. The reset value is the maximum timeout interval, and represents between 3 and 6 seconds.

Table 6-10. Bit Settings for Port Response Timeout Control CSR

Bit	Name	Reset Value	Description
0–23	timeout value	All 1s	timeout interval value
24-31	—		Reserved

6.6.4 Port General Control CSR (Block Offset 0x3C)

The bits accessible through the Port General Control CSR are bits that apply to all ports in a device. There is a single copy of each such bit per device. These bits are also accessible through the Port General Control CSR of any other physical layers implemented on a device.

The structure and bit definitions of the Port General Control CSR depend on whether or not the device contains an end point. The register bit definitions for a generic end point device with or without the software assisted error recovery option are specified in Table 6-11.

Table 6-11. Bit Settings for Port General Control CSR, Generic End Point Devices

Bit	Name	Reset Value	Description
0	Host	see footnote[1]	A Host device is a device that is responsible for system exploration, initialization, and maintenance. Agent or slave devices are initialized by Host devices. 0b0 - agent or slave device 0b1 - host device
1	Master Enable	see footnote[2]	The Master Enable bit controls whether or not a device is allowed to issue requests into the system. If the Master Enable is not set, the device may only respond to requests. 0b0 - processing element cannot issue requests 0b1 - processing element can issue requests
2	Discovered	see footnote[3]	This device has been located by the processing element responsible for system configuration 0b0 - The device has not been previously discovered 0b1 - The device has been discovered by another processing element
3-31	—		Reserved

[1]The Host reset value is implementation dependent

[2]The Master Enable reset value is implementation dependent

[3]The Discovered reset value is implementation dependent

The register bit definitions for a generic end point free device with or without the software assisted error recovery option are specified in Table 6-12.

Table 6-12. Bit Settings for General Port Control CSR, Generic End Point Free Device

Bit	Name	Reset Value	Description
0-1	—		Reserved
2	Discovered	see footnote[1]	This device has been located by the processing element responsible for system configuration 0b0 - The device has not been previously discovered 0b1 - The device has been discovered by another processing element
3-31	—		Reserved

[1]The Discovered reset value is implementation dependent

6.6.5 Port *n* Link Maintenance Request CSRs (Block Offsets 0x40, 60, ... , 220)

The port link maintenance request registers are accessible both by a local processor and an external device. A write to one of these registers generates a link-request control symbol on the corresponding RapidIO port interface.

Table 6-13. Bit Settings for Port *n* Link Maintenance Request CSRs

Bit	Name	Reset Value	Description
0–28	—		Reserved
29-31	Command	0b000	Command to be sent in the link-request control symbol. If read, this field returns the last written value.

6.6.6 Port *n* Link Maintenance Response CSRs (Block Offsets 0x44, 64, ... , 224)

The port link maintenance response registers are accessible both by a local processor and an external device. A read to this register returns the status received in a link-response control symbol. The ackID_status and port_status fields are defined in Table 3-3 and Table 3-6. This register is read-only.

Table 6-14. Bit Settings for Port *n* Link Maintenance Response CSRs

Bit	Name	Reset Value	Description
0	response_valid	0b0	If the link-request causes a link-response, this bit indicates that the link-response has been received and the status fields are valid. If the link-request does not cause a link-response, this bit indicates that the link-request has been transmitted. This bit automatically clears on read.
1-20	—		Reserved
21-26	ackID_status	0b000000	ackID status field from the link-response control symbol. Bit 21 is only valid for long control symbols.
27-31	port_status	0b00000	port status field from the link-response control symbol

6.6.7 Port *n* Local ackID CSRs (Block Offsets 0x48, 68, ... , 228)

The port link local ackID status registers are accessible both by a local processor and an external device. A read to this register returns the local ackID status for both the output and input sides of the ports.

Table 6-15. Bit Settings for Port *n* Local ackID Status CSRs

Bit	Name	Reset Value	Description
0	Clr_outstanding_ackIDs	0b0	Writing 0b1 to this bit causes all outstanding unacknowledged packets to be discarded. This bit should only be written when trying to recover a failed link. This bit is always logic 0 when read.
1	—		Reserved
2-7	Inbound_ackID	0b000000	Input port next expected ackID value. Bit 2 is only valid for long control symbols.
8-17	—		Reserved
18-23	Outstanding_ackID	0b000000	Output port unacknowledged ackID status. Next expected acknowledge control symbol ackID field that indicates the ackID value expected in the next received acknowledge control symbol. Bit 18 is only valid for long control symbols.
24-25	—		Reserved
26-31	Outbound_ackID	0b000000	Output port next transmitted ackID value. Software writing this value can force retransmission of outstanding unacknowledged packets in order to manually implement error recovery. Bit 26 is only valid for long control symbols.

6.6.8 Port *n* Error and Status CSRs (Block Offset 0x58, 78, ... , 238)

These registers are accessed when a local processor or an external device wishes to examine the port error and status information.

Table 6-16. Bit Settings for Port *n* Error and Status CSRs

Bit	Name	Reset Value	Description
0	Idle Sequence 2 Support	see footnote[1]	Indicates whether the port supports idle sequence 2 for baudrates of less than 5.5 GBaud. 0b0 - idle sequence 2 not supported for baudrates < 5.5 GBaud. 0b1 - idle sequence 2 supported for baudrates < 5.5 GBaud.
1	Idle Sequence 2 Enable	see footnote[2]	Controls whether idle sequence 2 is enabled for baudrates of less than 5.5 GBaud. 0b0 - idle sequence 2 disabled for baudrates < 5.5 GBaud. 0b1 - idle sequence 2 enabled for baudrates < 5.5 GBaud. The port shall not allow this bit to be set unless idle sequence 2 is supported and shall not allow this bit to be cleared if only idle sequence 2 is supported.
2	Idle Sequence	see footnote[1]	Indicates which idle is active. 0b0 - idle sequence 1 is active. 0b1 - idle sequence 2 is active.
3	—		Reserved
4	Flow Control Mode	0b0	Indicates which flow control mode is active (read only). 0b0 - receiver-controlled flow control is active. 0b1 - transmitter-controlled flow control is active.
5-10	—		Reserved
11	Output Retry-encountered	0b0	Output port has encountered a retry condition.This bit is set when bit 13 is set. Once set, remains set until written with a logic 1 to clear.
12	Output Retried	0b0	Output port has received a packet-retry control symbol and can not make forward progress. This bit is set when bit 13 is set and is cleared when a packet-accepted or a packet-not-accepted control symbol is received (read-only).
13	Output Retry-stopped	0b0	Output port has received a packet-retry control symbol and is in the “output retry-stopped” state (read-only).
14	Output Error-encountered	0b0	Output port has encountered (and possibly recovered from) a transmission error. This bit is set when bit 15 is set. Once set, remains set until written with a logic 1 to clear.
15	Output Error-stopped	0b0	Output is in the “output error-stopped” state (read-only).
16-20	—		Reserved
21	Input Retry-stopped	0b0	Input port is in the “input retry-stopped” state (read-only).
22	Input Error-encountered	0b0	Input port has encountered (and possibly recovered from) a transmission error. This bit is set when bit 23 is set. Once set, remains set until written with a logic 1 to clear.
23	Input Error-stopped	0b0	Input port is in the “input error-stopped” state (read-only).
24-26	—		Reserved

Table 6-16. Bit Settings for Port *n* Error and Status CSRs

Bit	Name	Reset Value	Description
27	Port-write Pending	0b0	Port has encountered a condition which required it to initiate a Maintenance Port-write operation This bit is only valid if the device is capable of issuing a maintenance port-write transaction. Once set remains set until written with a logic 1 to clear.
28	Port Unavailable	see footnote[3]	Indicates whether or not the port is available (read only). The port's resources may have been merged with another port to support wider links. 0b0 - The port is available for use. 0b1 - The port is not available for use.
29	Port Error	0b0	Input or output port has encountered an error from which hardware was unable to recover. Once set, remains set until written with a logic 1 to clear.
30	Port OK	0b0	The input and output ports are initialized and the port is exchanging error-free control symbols with the attached device (read-only).
31	Port Uninitialized	0b1	Input and output ports are not initialized. This bit and bit 30 are mutually exclusive (read-only).

[1]The reset value is implementation dependent

[2]The reset value is 0b1 if feature is supported, otherwise 0b0

[3]The Port Unavailable reset value is implementation dependent

6.6.9 Port *n* Control CSRs (Block Offsets 0x5C, 7C, ... , 23C)

The port *n* control registers contain control register bits for individual ports on a processing element.

Table 6-17. Bit Settings for Port *n* Control CSRs

Bit	Name	Reset Value	Description
0-1	Port Width Support	see footnote[1]	Indicates port width modes supported by the port (read-only). This field is used in conjunction with the Extended Port Width Support field of this register. The bits of these two fields collectively indicate the port width modes supported by the port in addition to 1x mode which is supported by all ports Bit 0: 0b0 - 2x mode not supported 0b1 - 2x mode supported Bit 1: 0b0 - 4x mode not supported 0b1 - 4x mode supported
2-4	Initialized Port Width	see footnote[2]	Width of the ports after initialized (read only): 0b000 - Single-lane port 0b001 - Single-lane port, lane R 0b010 - Four-lane port 0b011 - Two-lane port 0b100 - Eight-lane port 0b101 - Sixteen-lane port 0b110 - 0b111 - Reserved

Bit	Name	Reset Value	Description
5 - 7	Port Width Override	0b000	Soft port configuration to control the width modes available for port initialization. The bits in this field are used and defined in conjunction with the bits of the Extended Port Width Override field (bits 16-17). When bit [5] = 0b0, bits 16-17 are Reserved When bit [5] = 0b1, bit 6 controls the enabling of 4x mode, bit 7 controls the enabling of 2x mode, bit 16 controls the enabling of 8x mode and bit 17 controls the enabling of 16x mode. Port n Control CSR bits [5-7,16-17] 0b000xx - All lanes widths supported by the port are enabled 0b001xx - Reserved 0b010xx - Force 1x mode, lane R not forced 0b011xx - Force 1x mode, force lane R 0b10000 - Implementation specific behavior 0b10001 - 16x mode enabled; 2x, 4x and 8x modes disabled 0b10010 - 8x mode enabled; 2x, 4x and 16x modes disabled 0b10011 - 8x and 16x modes enabled; 2x and 4x modes disabled 0b10100 - 2x mode enabled; 4x, 8x and 16x modes disabled 0b10101 - 2x and 16x modes enabled; 4x and 8x modes disabled 0b10110 - 2x and 8x modes enabled; 4x and 16x modes disabled 0b10111 - 2x, 8x and 16x modes enabled; 4x mode disabled 0b11000 - 4x mode enabled; 2x, 8x and 16x modes disabled 0b11001 - 4x and 16x modes enabled; 2x and 8x modes disabled 0b11010 - 4x and 8x modes enabled; 2x and 16x modes disabled 0b11011 - 4x, 8x and 16x modes enabled; 2x mode disabled 0b11100 - 2x and 4x modes enabled; 8x and 16x modes disabled 0b11101- 2x, 4x and 16x modes enabled; 8x mode disabled 0b11110 - 2x, 4x and 8x modes enabled; 16x mode disabled 0b11111- 2x, 4x, 8x and 16 x modes enabled The port shall not allow the enabling of a link width mode that is not supported by the port. A change in the value of the Port Width Override or Extended Port Width Override field shall cause the port to re-initialize using the new field value(s).
8	Port Disable	0b0	Port disable: 0b0 - port receivers/drivers are enabled 0b1 - port receivers/drivers are disabled and are unable to receive/transmit any packets or control symbols
9	Output Port Enable	see footnote[3]	Output port transmit enable: 0b0 - port is stopped and not enabled to issue any packets except to route or respond to I/O logical MAINTENANCE packets. Control symbols are not affected and are sent normally. This is the recommended state after device reset. 0b1 - port is enabled to issue any packets

Bit	Name	Reset Value	Description
10	Input Port Enable	see footnote[4]	Input port receive enable: 0b0 - port is stopped and only enabled to route or respond I/O logical MAINTENANCE packets. Other packets generate packet-not-accepted control symbols to force an error condition to be signaled by the sending device. Control symbols are not affected and are received and handled normally. This is the recommended state after device reset. 0b1 - port is enabled to respond to any packet
11	Error Checking Disable	0b0	This bit disables all RapidIO transmission error checking 0b0 - Error checking and recovery is enabled 0b1 - Error checking and recovery is disabled Device behavior when error checking and recovery is disabled and an error condition occurs is undefined
12	Multicast-event Participant	see footnote[5]	Send incoming Multicast-event control symbols to this port (multiple port devices only)
13	—		Reserved
14	Enumeration Boundary	see footnote[6]	An enumeration boundary aware system enumeration algorithm shall honor this flag. The algorithm, on either the ingress or the egress port, shall not enumerate past a port with this bit set. This provides for software enforced enumeration domains within the RapidIO fabric.
15	—		Reserved
16-17	Extended Port Width Override	0b00	Extended soft port configuration to control the width modes available for port initialization. The bits in this field are used and defined in conjunction with the bits in the Port Width Override field. See the Description of the Port Width Override field for the specification of these bits.
18-19	Extended Port Width Support	see footnote[7]	Indicates additional port width modes supported by the port (read-only). This field is used in conjunction with the Port Width Support field of this register. The bits of these two fields collectively indicate the port width modes supported by the port in addition to 1x mode which is supported by all ports Bit 18: 0b0 - 8x mode not supported 0b1 - 8x mode supported Bit 19: 0b0 - 16x mode not supported 0b1 - 16x mode supported
20-27	Implementation-defined		Implementation-defined
28-30	—		Reserved
31	Port Type		This indicates the port type (read only) 0b0 - Reserved 0b1 - Serial port

[1]The Port Width reset value is implementation dependent

[2]The Initialized Port Width reset value is implementation dependent

[3]The Output Port Enable reset value is implementation dependent

[4]The Input Port Enable reset value is implementation dependent

[5]The Multicast-event Participant reset value is implementation dependent

[6]The Enumeration Boundary reset value is implementation dependent; provision shall be made to allow the reset value to be configurable if this feature is supported

[7]The Extended Port Width Support reset value is implementation dependent

6.6.10 Port *n* Control 2 CSRs (Block Offset 0x54, 74, ... , 234)

These registers are accessed when a local processor or an external device wishes to examine the port baudrate information.

Table 6-18. Bit Settings for Port *n* Control 2 CSRs

Bit	Name	Reset Value	Description
0-3	Selected Baudrate	0b0000	Indicates the initialized baudrate of the port 0b0000 - no rate selected 0b0001 - 1.25 GBaud 0b0010 - 2.5 GBaud 0b0011 - 3.125 GBaud 0b0100 - 5.0 GBaud 0b0101 - 6.25 GBaud 0b0110 - 0b1111 - Reserved
4	Baudrate Discovery Support	see footnote[1]	Indicates whether automatic baudrate discovery is supported (read-only) 0b0 - Automatic baudrate discovery not supported 0b1 - Automatic baudrate discovery supported
5	Baudrate Discovery Enable	see footnote[2]	Controls whether automatic baudrate discovery is enabled 0b0 - Automatic baudrate discovery disabled 0b1 - Automatic baudrate discovery enable The port shall not allow this bit to be set unless it supports baudrate discovery.
6	1.25 GBaud Support	see footnote[1]	Indicates whether port operation at 1.25 GBaud is supported (read only) 0b0 - 1.25 GBaud operation not supported 0b1 - 1.25 GBaud operation supported
7	1.25 GBaud Enable	see footnote[2]	Controls whether port operation at 1.25 GBaud is enabled 0b0 - 1.25 GBaud operation disabled 0b1 - 1.25 GBaud operation enabled The port shall not allow this bit to be set unless it supports 1.25 GBaud.
8	2.5 GBaud Support	see footnote[1]	Indicates whether port operation at 2.5 GBaud is supported (read only) 0b0 - 2.5 GBaud operation not supported 0b1 - 2.5 GBaud operation supported
9	2.5 GBaud Enable	see footnote[2]	Controls whether port operation at 2.5 GBaud is enabled 0b0 - 2.5 GBaud operation disabled 0b1 - 2.5 GBaud operation enabled The port shall not allow this bit to be set unless it supports 2.5 GBaud.
10	3.125 GBaud Support	see footnote[1]	Indicates whether port operation at 3.125 GBaud is supported (read only) 0b0 - 3.125 GBaud operation not supported 0b1 - 3.125 GBaud operation supported
11	3.125 GBaud Enable	see footnote[2]	Controls whether port operation at 3.125 GBaud is enabled 0b0 - 3.125 GBaud operation disabled 0b1 - 3.125 GBaud operation enabled The port shall not allow this bit to be set unless it supports 3.125 GBaud.
12	5.0 GBaud Support	see footnote[1]	Indicates whether port operation at 5.0 GBaud is supported (read only) 0b0 - 5.0 GBaud operation not supported 0b1 - 5.0 GBaud operation supported

Table 6-18. Bit Settings for Port *n* Control 2 CSRs

Bit	Name	Reset Value	Description
13	5.0 GBaud Enable	see footnote[2]	Controls whether port operation at 5.0 GBaud is enabled 0b0 - 5.0 GBaud operation disabled 0b1 - 5.0 GBaud operation enabled The port shall not allow this bit to be set unless it supports 5.0 GBaud.
14	6.25 GBaud Support	see footnote[1]	Indicates whether port operation at 6.25 GBaud is supported (read only) 0b0 - 6.25 GBaud operation not supported 0b1 - 6.25 GBaud operation supported
15	6.25 GBaud Enable	see footnote[2]	Controls whether port operation at 6.25 GBaud is enabled 0b0 - 6.25 GBaud operation disabled 0b1 - 6.25 GBaud operation enabled The port shall not allow this bit to be set unless it supports 6.25 GBaud
16-27	—		Reserved

Table 6-18. Bit Settings for Port *n* Control 2 CSRs

Bit	Name	Reset Value	Description
28	Enable Inactive Lanes	0b0	The implementation of this bit is optional. When implemented, this bit allows software to force the lanes of the port that are not currently being used to carry traffic, the “inactive lanes”, to be enabled for testing while the “active lanes” continue to carry traffic. If this bit is not implemented it is reserved. When a 1x/Nx or 1x/Mx/Nx port is operating in 1x mode where 1<M<N and N = 4, 8 or 16, lanes 0 and 2 are the active lanes and lane 1 and lanes 3 through N-1 are the inactive lanes. When a 1x/Mx/Nx port is operating in Mx mode where 1<M<N and N = 4, 8 or 16, lanes 0 through M-1 are the active lanes and lanes M through N-1 are the inactive lanes. The test mode enabled by the implementation of this bit only allows the testing of the inactive lanes that are supported by both of the connected ports. For example, if a 1x/4x/8x port is connected to a 1x/4x/16x port and the link is operating in 4x mode, only lanes 4 though 7 can be monitored using this test mode. Use of the test mode enabled by the implementation of this bit to monitor the behavior of the inactive lanes requires that this bit must be set in both ports and that all link width modes wider than the desired Mx mode must be disabled in the Port n Control CSR of both ports. Failure to meet these requirements will result in unspecified link behavior. (Modes wider than the desired Mx mode must be disabled so that the Initialization state machine ignores the asserted lane_sync signals from the lanes with forced output enables and does not attempt to enter a mode wider than Mx). **If implemented, this bit shall not be asserted when the port is connected to a link that includes retimers as defined in Section 4.11.1, "Retimers".** 0b0: The output enables of all of the lanes controlled by the port are controlled solely by the port’s Initialization state machine 0b1: The port’s receivers for the inactive lanes are enabled. The port’s drivers for the inactive lanes are output enabled if and only if the port’s Initialization state machine is not in the SILENT or SEEK state. A continuous IDLE sequence of the same type as is in use on the active lanes shall be transmitted on the inactive lanes when their transmitters are output enabled. The IDLE sequences transmitted on the inactive lanes shall comply with all rules for that type of IDLE sequence including alignment across the inactive lanes, but they are not required to use the same bit sequences or be aligned in any way relative to the IDLE sequences transmitted on the active lanes. If IDLE2 is being used on the active lanes of the port, the inactive lanes of the port shall report their lane number and port width in the CS Field Marker and handle commands carried in the CS Field as if they were active lanes.

Table 6-18. Bit Settings for Port *n* Control 2 CSRs

Bit	Name	Reset Value	Description
29	Data scrambling disable	0b0	Read-write 0b0: transmit scrambler and receive descrambler are enabled. 0b1: The transmit scrambler and receive descrambler are disabled for control symbol and packet data characters. Control symbol and packet data characters are neither scrambled in the transmitter before transmission nor descrambled in the receiver upon reception. The transmit scrambler remains enabled for the generation of pseudo-random data characters for the IDLE2 random data field. This bit is for test use only and shall not be asserted during normal operation.
30	Remote Transmit Emphasis Control Support	see footnote[3]	Indicates whether the port is able to transmit commands to control the transmit emphasis in the connected port 0b0 - The port does not support transmit emphasis adjustment in the connected port 0b - The port supports transmit emphasis adjustment in the connected port
31	Remote Transmit Emphasis Control Enable	see footnote[4]	Controls whether the port may adjust the transmit emphasis in the connected port 0b0 - Remote transmit emphasis control is disabled 0b1 - Remote transmit emphasis control is enabled The port shall not let this bit be set unless remote transmit emphasis control is supported and the link to which the port is connect is using idle sequence 2 (IDLE2).

[1]The reset value is implementation dependent

[2]The reset value is 0b1 if feature is supported, otherwise 0b0

[3]The Remote Transmit Emphasis Control Support reset value is implementation dependent

[4]The Remote Transmit Emphasis Control Enable reset value is implementation dependent

6.7 LP-Serial Lane Extended Features Block

This section specifies the LP-Serial Lane Extended Features Block. All registers in this block are 32 bits in length, aligned to a 32-bit (4-byte) boundary and accessed as 4 byte entities, although some processing elements may optionally allow larger accesses. This Extended Features register block is assigned Extended Features block ID=0x000D.

6.7.1 Register Map

Table 6-19 shows the register map of the RapidIO LP-Serial Lane Extended Features Block. The register map specifies the registers that comprise this Extended Features Block.

The Block Offset is the offset relative to the 16-bit Extended Features Pointer (EF_PTR) that points to the beginning of the block. The address of a byte in the block is calculated by adding the block byte offset to EF_PTR that points to the beginning of the block. This is denoted as [EF_PTR+xx] where xx is the block byte offset in hexadecimal.

This register map is currently only defined for devices with up to 32 LP-Serial lanes, but can be extended or shortened if more or less lane definitions are required for a device. For example, a device with four LP-Serial lanes is only required to use register map space corresponding to offsets [EF_PTR + 0x00] through [EF_PTR + 0x8C]. Register map offset [EF_PTR + 0x90] can be used for another Extended Features block.

Table 6-19. LP-Serial Lane Register Map

	Block Byte Offset	Register Name
General	0x0	LP-Serial Lane Register Block Header
	0x4–C	Reserved
Lane 0	0x10	Lane 0 Status 0 CSR
	0x14	Lane 0 Status 1 CSR
	0x18	Lane 0 Status 2 CSR
	0x1C	Lane 0 Status 3 CSR
	0x20	Lane 0 Status 4 CSR
	0x24	Lane 0 Status 5 CSR
	0x28	Lane 0 Status 6 CSR
	0x2C	Lane 0 Status 7 CSR

Table 6-19. LP-Serial Lane Register Map

	Block Byte Offset	Register Name
Lane 1	0x30	Lane 1 Status 0 CSR
	0x34	Lane 1 Status 1 CSR
	0x38	Lane 1 Status 2 CSR
	0x3C	Lane 1 Status 3 CSR
	0x40	Lane 1 Status 4 CSR
	0x44	Lane 1 Status 5 CSR
	0x48	Lane 1 Status 6 CSR
	0x4C	Lane 1 Status 7 CSR
Ports 2-30	0x50–3EC	Registers for lanes 2-30
Port 31	0x3F0	Lane 31 Status 0 CSR
	0x3F4	Lane 31 Status 1 CSR
	0x3F8	Lane 31 Status 2 CSR
	0x3FC	Lane 31 Status 3 CSR
	0x400	Lane 31 Status 4 CSR
	0x404	Lane 31 Status 5 CSR
	0x408	Lane 31 Status 6 CSR
	0x40C	Lane 31 Status 7 CSR

The structure and use of the registers comprising the LP-Serial Lane Extended Features Block is specified in Section 6.7.2.

The required behavior for accesses to reserved registers and register bits is specified in Table 6-2.

6.7.2 LP-Serial Lane Command and Status Registers (CSRs)

6.7.2.1 LP-Serial Register Block Header (Block Offset 0x0)

The LP-Serial register block header register contains the EF_PTR to the next extended features block and the EF_ID that identifies LP-Serial Lane Extended Feature Block for which this is the register block header.

Table 6-20. Bit Settings for LP-Serial Register 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	0x000D	Hard wired Extended Features Block ID

6.7.2.2 Lane *n* Status 0 CSRs (Block Offsets 0x10, 30, ... , 3F0)

This register shall always be implemented. It contains status information about the local lane transceiver, i.e. the lane *n* transceiver in the device implementing this register. Unless otherwise specified, all bits in this register are read-only (RO).

Table 6-21. Bit Settings for Lane *n* Status 0 CSRs

Bit	Name	Reset Value	Description
0-7	Port Number		The number of the port within the device to which the lane is assigned
8-11	Lane Number		The number of the lane within the port to which the lane is assigned
12	Transmitter type		Transmitter type 0b0 - short run 0b1 - long run
13	Transmitter mode		Transmitter operating mode 0b0 - short run 0b1 - long run
14-15	Receiver type		Receiver type 0b00 - short run 0b01 - medium run 0b10 - long run 0b11 - Reserved
16	Receiver input inverted		This bit indicates whether the lane receiver has detected that the polarity of its input signal is inverted and has inverted its receiver input to correct the polarity. 0b0 - receiver input not inverted 0b1 - receiver input inverted
17	Receiver trained		When the lane receiver controls any transmit or receive adaptive equalization, this bit indicates whether or not all adaptive equalizers controlled by the lane receiver are trained. If the lane supports the IDLE2 sequence, the value of this bit shall be the same as the value in the “Receiver trained” bit in the CS Field transmitted by the lane. 0b0 - One or more adaptive equalizers are controlled by the lane receiver and at least one of those adaptive equalizers is not trained 0b1 - The lane receiver controls no adaptive equalizers or all of the adaptive equalizers controlled by the lane receiver are trained
18	Receiver lane sync		This bit indicates the state of the lane’s lane_sync signal. 0b0: lane_sync FALSE 0b1: lane_sync TRUE
19	Receiver lane ready		This bit indicates the state of the lane’s lane_ready signal 0b0 - lane_ready FALSE 0b1 - lane_ready TRUE

Bit	Name	Reset Value	Description
20-23	8B/10B decoding errors	0x0	This field indicates the number of 8B/10B decoding errors that have been detected for this lane since this register was last read. The field is reset to 0x0 when the register is read. 0x0: No 8B/10B decoding errors have been detected since this register was last read. 0x1: One 8B/10B decoding error has been detected since this register was last read. 0x2: Two 8B/10B decoding errors have been detected since this register was last read. ... 0xD: Thirteen 8B/10B decoding errors have been detected since this register was last read. 0xE: Fourteen 8B/10B decoding errors have been detected since this register was last read. 0xF: At least fifteen 8B/10B decoding errors have been detected since this register was last read.
24	Lane_sync state change	0b0	Indicates whether the lane_sync signal for this lane has changed state since the bit was last read. This bit is reset to 0b0 when the register is read. This bit provides an indication of the burstiness of the transmission errors detected by the lane receiver. 0b0 - The state of lane_sync has not changed since this register was last read 0b1 - The state of lane_sync has changed since this register was last read
25	Rcvr_trained state change	0b0	Indicates whether the rcvr_trained signal for this lane has changed state since the bit was last read. This bit is reset to 0b0 when the register is read. A change in state of rcvr_trained indicates that the training state of the adaptive equalization under the control of this receiver has changed. Frequent changes of the training state suggest a problem with the lane. 0b0 - The state of rcvr_trained has not changed since this register was last read 0b1 - The state of rcvr_trained has changed since this register was last read
26-27	—		Reserved
28	Status 1 CSR implemented		This bit indicates whether or not the Status 1 CSR is implemented for this lane 0b0 - The Status 1 CSR is not implemented for this lane 0b1 - The Status 1 CSR is implemented for this lane
29-31	Status 2-7 CSRs implemented		This field indicates the number of implementation specific Status 2-7 CSRs that are implemented for this lane 0b000 - None of the Status 2-7 CSRs are implemented for this lane 0b001 - The Status 2 CSR is implemented for this lane 0b010 - The Status 2 and 3 CSRs are implemented for this lane 0b011 - The Status 2 through 4 CSRs are implemented for this lane 0b100 - The Status 2 through 5 CSRs are implemented for this lane 0b101 - The Status 2 through 6 CSRs are implemented for this lane 0b110 - The Status 2 through 7 CSRs are implemented for this lane 0b111 - Reserved

6.7.2.3 Lane *n* Status 1 CSRs (Block Offsets 0x14, 34, ... , 3F4)

This register shall be implemented if and only if the lane supports the IDLE2 sequence. The register contains information about the connected port that is collected from the CS markers and CS fields of the IDLE2 sequence received by the local lane *n* receiver. Only information from error free CS markers and CS fields shall be reported in this register. Unless otherwise specified, all bits in this register are read-only (RO).

Table 6-22. Bit Settings for Lane *n* Status 1 CSRs

Bit	Name	Reset Value	Description
0	IDLE2 received	0b0	This bit indicates whether an IDLE2 has been received by the lane since the bit was last reset. The bit is R/W. This bit can be reset by writing the bit with the value 0b1. Writing the bit with the value 0b0 does not change the value of the bit. 0b0 - No IDLE2 sequence has been received since the bit was last reset 0b1 - An IDLE2 sequence has been received at some time since the bit was last reset
1	IDLE2 information current	0b0	This bit indicates whether the information in this register that is collected from the received IDLE2 sequence is current. When asserted, this bit indicates that the information is from the last IDLE2 CS Marker and CS Field that were received by the lane without detected errors, and that the lane's lane_sync signal has remained asserted since the last CS Marker and CS Field were received. 0b0 - The IDLE2 information is not current 0b1 - The IDLE2 information is current
2	Values changed	0b1	This bit indicates whether the values of any of the other 31 bits in this register have changed since the register was last read. This bit is reset when the register is read. 0b0 - The values have not changed 0b1 - One or more values have changed
3	Implementation defined		Implementation defined
4	Connected port lane receiver trained		Connected port lane receiver trained 0b0 - Receiver not trained 0b1 - Receiver trained
5-7	Received port width		Received port width 0b000 - 1 lane 0b001 - 2 lanes 0b010 - 4 lanes 0b011 - 8 lanes 0b100 - 16 lanes 0b101-0b111 - Reserved
8-11	Lane number in connected port		The number of the lane (0-15) within the connected port 0b0000 - Lane 0 0b0001 - Lane 1 ... 0b1111 - Lane 15

Bit	Name	Reset Value	Description
12-13	Connected port transmit emphasis Tap(-1) status		Tap(-1) status 0b00 - Tap(-1) not implemented 0b01 - Tap(-1) at minimum emphasis 0b10 - Tap(-1) at maximum emphasis 0b11 - Tap(-1) at intermediate emphasis setting
14-15	Connected port transmit emphasis Tap(+1) status		Tap(+1) status 0b00 - Tap(+1) not implemented 0b01 - Tap(+1) at minimum emphasis 0b10 - Tap(+1) at maximum emphasis 0b11 - Tap(+1) at intermediate emphasis setting
16	Connected port scrambling/descrambling enabled		Connected port scrambling/descrambling 0b0 - Scrambling/descrambling not enabled 0b1 - Scrambling/descrambling enabled
17-31	—		Reserved

6.7.2.4 Implementation Specific CSRs

6.7.2.4.1 Lane n Status 2 CSR (Block Offsets 0x18, 38, ..., 3F8)

6.7.2.4.2 Lane n Status 3 CSR (Block Offsets 0x1C, 3C, ..., 3FC)

6.7.2.4.3 Lane n Status 4 CSR (Block Offsets 0x20, 40, ..., 400)

6.7.2.4.4 Lane n Status 5 CSR (Block Offsets 0x24, 44, ..., 404)

6.7.2.4.5 Lane n Status 6 CSR (Block Offsets 0x28, 48, ..., 408)

6.7.2.4.6 Lane n Status 7 CSR (Block Offsets 0x2C, 4C, ..., 40C)

The implementation of these registers is optional and when implemented their contents and format are implementation specific. The registers shall be implemented in increasing numerical order beginning with the Lane *n* Status 2 CSR. For example, if only one of the registers is implemented it shall be the Status 2 CSR. If three registers are implemented they shall be the Status 2 through 4 CSRs, and if five of the registers are implemented, they shall be the Status 2 through 6 CSRs.

6.8 Virtual Channel Extended Features Block

This section describes the registers for RapidIO LP-Serial devices supporting virtual channels. This Extended Features register block is assigned Extended Features block EF_ID=0x000A.

6.8.1 Register Map

Table 6-23 shows the virtual channel register map for RapidIO LP-Serial devices. The Block Offset is the offset relative to the 16-bit Extended Features Pointer (EF_PTR) that points to the beginning of the block.

The address of a byte in the block is calculated by adding the block byte offset to EF_PTR that points to the beginning of the block. This is denoted as [EF_PTR+xx] where xx is the block byte offset in hexadecimal.

Table 6-23. Virtual Channel Registers

	Block Byte Offset	Register Name
General	0x0	VC Register Block Header
	0x4-1C	Reserved
Port 0	0x20	Port 0 VC Control and Status Register
	0x24	Port 0 VC0 BW Allocation Register
	0x28	Port 0 VC 5, VC 1 BW Allocation Register
	0x2C	Port 0 VC 7, VC 3 BW Allocation Register
	0x30	Port 0 VC 6, VC 2 BW Allocation Register
	0x34	Port 0 VC 8, VC 4 BW Allocation Register
	0x38-3C	Reserved
Port 1	0x40	Port 1 VC Control and Status Register
	0x44	Port 1 VC0 BW Allocation Register
	0x48	Port 1 VC 5, VC 1 BW Allocation Register
	0x4C	Port 1 VC 7, VC 3 BW Allocation Register
	0x50	Port 1 VC 6, VC 2 BW Allocation Register
	0x54	Port 1 VC 8, VC 4 BW Allocation Register
	0x58-5C	Reserved
Port *n*	[(0x20 * (n + 1)] to [0x20 * (n + 1) + 0x1C]	Additional Port Registers

The registers are paired according to the VCs as they are implemented. In the second example, with VCs Supported 0x01, the upper portion (VC5 portion) of the register would be non-functioning.

NOTE:

There are no provisions in this specification to provide for dynamic reconfiguration of the VCs. A vendor is not prohibited from implementing dynamic reconfiguration, it is just beyond the scope of this specification. Both ends of the channel need to be configured alike, or unexpected behavior may result, also beyond the scope of this specification. The default method is to configure VC operation when the channel is quiescent either by protocol, or by holding the master enable in the disabled state.

6.8.2 Virtual Channel Control Block Registers

This section contains register descriptions that define the bandwidth allocation configuration for the virtual channels.

6.8.2.1 VC Register Block Header (Block Offset 0x0)

The LP-Serial VC register block header register contains the EF_PTR to the next extended features block and the EF_ID that identifies this as the Virtual Channel Extended Features Block.

Table 6-24. Bit Settings for VC Register 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	0x000A	Hard wired Extended Features Block ID

6.8.2.2 Port *n* VC Control and Status Registers (Block Offset ((port number) + 1) * 0x20))

This register is used by each port to set up VC operation.

Table 6-25. Port *n* VC Control and Status Registers

Bit	Name	Reset Value	Description
0 - 7	VC Refresh Interval	0x00	The number of 1024 code group intervals over which the VC status must be refreshed. Refresh Interval: 0x0 - 1K code groups, 0xF - 16K code groups, 0xFF - 256K code groups Implementers are required to support a maximum VC refreshing period of at least 1024 x 16 = 16K code groups in size. The maximum possible VC refreshing period that can be supported is 1024 x 256 = 256K code groups. Writing to this field with a value greater than the maximum supported value by the port will set the field to the maximum value supported by the port
8 - 15	CT Mode	0x00	Enables VCs to operate in CT mode beginning with VC8: 0x00 - all VCs in RT mode For 8 VCs: 0x01 - VC8 in CT mode 0x03 - VC8, VC7 in CT mode 0x07 - VC8, VC7, VC6, VC 5 in CT mode 0x0F - VC8 - VC1 in CT mode For 4 VCs: 0x01 - VC7 in CT mode 0x03 - VC7, VC5 in CT mode 0x07 - VC7, VC5, VC3, VC1 in CT mode For 2 VCs: 0x01 - VC5 in CT mode 0x03 - VC5, VC1 in CT mode For 1 VC: 0x01 - VC1 in CT mode Implementers may support CT mode on a portion of the available VCs. CT mode must be implemented in the highest VCs first to allow this simplified programming model. VCs not supporting CT operation are indicated by not allowing the programmed bits to set. Example: 8VCs enabled, VC8 and VC7 only support CT mode. Writing a 0x07 would result in a register value of 0x03 when read back.

Table 6-25. Port *n* VC Control and Status Registers

Bit	Name	Reset Value	Description
16 - 23	VCs Support	see footnote[1]	Number of Virtual Channels Supported (Read Only) 0x00 - Only VC0 is supported 0x01 - VC0, VC1 Supported 0x02 - VC0, VC1, VC5 supported 0x04 - VC0, VC1, VC3, VC5, VC7 supported 0x08 - VC0, VC1-VC8
24 - 31	VCs Enable	0x00	0x00 - Enable Only VC0 0x01 - Enable VC0, VC1 0x02 - Enable VC0, VC1, VC5 0x04 - Enable VC0, VC1, VC3, VC5, VC7 0x08 - Enable VC0, VC1-VC8 Note: Bits 24-27, and any bits associated with unimplemented VCs need not be writable, but must return 0 when read. Setting this field to a value larger than the number of VCs supported as indicated in bits 16-23. will result in only VC0 being enabled.

[1]The VCs Supported reset value is implementation dependent

6.8.2.3 Port *n* VC0 BW Allocation Registers (Block Offset (((port number) + 1) * 0x20) + 0x04))

This register is used to enable and configure VC0's participation in the bandwidth reservation scheduling.

Table 6-26. Port *n* VC0 BW Allocation CSR

Bit	Name	Reset Value	Description
0	VC0 Bandwidth Reservation Capable	see footnote[1]	0b0 - VC0 is strict priority, and has priority over the other VCs. It will utilize bandwidth without regard to bandwidth reservation. The bandwidth reservation algorithm will divide up what bandwidth is remaining after VC0 has no outstanding requests. 0b1 - VC0 is capable of being allocated bandwidth This bit is read only
1	VC0 BW Res Enable	0b0	0b0 - VC0 is strict priority, does not participate in bandwidth reservation 0b1 - VC0 will be allocated bandwidth according to BW Allocation Registers
2 - 7	—		Reserved
8 - 15	Bandwidth Reservation Precision	see footnote[2]	Indicates the number of bits used in the bandwidth reservation precision for all VCs in this port. (read only) 0x00 - 8 bits 0x01 - 9 bits 0x02 - 10 bits 0x04 - 11 bits 0x08 - 12 bits 0x10 - 13 bits 0x20 - 14 bits 0x40 - 15 bits 0x80 - 16 bits
16-31	Bandwidth Allocation	0x00	The contents of this register determines the minimum bandwidth reserved for this VC (see below) The bandwidth allocation value is left justified based on precision. Bits, are ignored based on the precision value: 0bnnnn_nnnn_xxxx_xxxx (8 bit precision) where 'x' represents ignored bits 0bnnnn_nnnn_nxxx_xxxx (9 bit precision) 0bnnnn_nnnn_nnnn_xxxx (12 bit precision), etc.

[1] The VC0 Bandwidth Reservation Capable reset value is implementation dependent
[2] The Bandwidth Reservation Precision reset value is implementation dependent

VC0 may or may not participate in the bandwidth reservation scheduling for the link. The required implementation is for VC0 to be strict priority. Traffic on VC0 is serviced before any of the other VCs in this mode. The remaining bandwidth is then divided according to the percentages in the bandwidth allocations. This will result in the bandwidth allocations being variable if VC0's utilization is significant when compared with the activity on the other VCs.

Optionally, VC0 may be included in the bandwidth reservation scheduling. In this case, the priorities within VC0 are serviced when VC0 is allocated bandwidth on the

link. VC0 activity cannot cause the other VCs to receive less than their allocation of bandwidth.

The Bandwidth Reservation Precision field is used to indicate the granularity of bandwidth scheduling for the port. The value in this register applies to the subsequent BW Allocation Registers as well.

The value programmed in the BW Allocation Registers is a binary fraction based on the percentage of the overall total bandwidth. 100% bandwidth is represented by a value of 1.000:

Table 6-27. BW Allocation Register Bit Values

Bit / Value							
0	1	2	3	4	5	6	7
2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}
Bit / Value							
8	9	10	11	12	13	14	15
2^{-9}	2^{-10}	2^{-11}	2^{-12}	2^{-13}	2^{-14}	2^{-15}	2^{-16}

Example: 33% bandwidth is allocated as:

33/100 = 0.0101010101010101b, so the BW allocation register value is:
0101010101010101b, and would be rounded down to:
01010101xxxxxxxxb if 8 bit precision is being used.

The value may be programmed as is into the left justified register, with the unused bits being ignored, but that might cause some precision errors. Also, if the percentage results in a value smaller than the precision, a value of 0 could result in a VC getting no service. The precision value allows the bandwidth allocation algorithm to round up or down based on the dividing point, and to detect and round up a zero value to allocate at least a minimal increment of bandwidth.

The total of all the allocations should not exceed 100%. The result, by definition, will not be as programmed. The actual behavior will depend on the method used to schedule the activity. The implementation of the scheduler, and thus its behavior when not programmed correctly is outside the scope of this specification.

6.8.2.4 Port *n* VC*x* BW Allocation Registers (Block Offset ((((port number) + 1) * 0x20) + (offset based on VC #, see Table 6-23)))

This register is used to enable and program VCs 1-8 participation in the bandwidth reservation scheduling. Each register supports 2 VCs, ordered as described in Section 6.8.1, "Register Map".

Table 6-28. Port *n* VC*x* BW Allocation CSR

Bit	Name	Reset Value	Description
0 - 15	Bandwidth Allocation	0x0000	The contents of this register determines the minimum bandwidth reserved for this VC (see below) The bandwidth allocation value is left justified based on precision. Bits, are ignored based on the precision value: 0bnnnn_nnnn_xxxx_xxxx (8 bit precision) where 'x' represents ignored bits 0bnnnn_nnnn_nxxx_xxxx (9 bit precision) 0bnnnn_nnnn_nnnn_xxxx (12 bit precision), etc.
16-31	Bandwidth Allocation	0x0000	The contents of this register determines the minimum bandwidth reserved for this VC (see below) The bandwidth allocation value is left justified based on precision. Bits, are ignored based on the precision value: 0bnnnn_nnnn_xxxx_xxxx (8 bit precision) where 'x' represents ignored bits 0bnnnn_nnnn_nxxx_xxxx (9 bit precision) 0bnnnn_nnnn_nnnn_xxxx (12 bit precision), etc.

In the instance where VC1 is supported, but VC5 is not, bits 0 - 15 are reserved.

The Bandwidth Allocation is as described previously for VC0.

A value of '0' for bandwidth allocation results in no service being given to that VC. VCs initialize with a value of zero and remain inactive until allocated bandwidth. It is recommended that the bandwidth allocations be made before enabling the VCs, but the actual implementation is beyond the scope of this specification.

Chapter 7 Signal Descriptions

7.1 Introduction

This chapter contains the signal pin descriptions for a RapidIO LP-Serial port. The interface is defined either as a 1x, 2x, 4x, 8x, or 16x lane, full duplex, point-to-point interface using differential signaling. A lane implementation consists of Nx4 wires with two for the egress and two for the ingress direction. The electrical details are described in Chapter 8, "Common Electrical Specifications".

7.2 Signal Definitions

Table 7-1 provides a summary of the RapidIO LP-Serial signal pins as well as a short description of their functionality.

Table 7-1. LP-Serial Signal Description

Signal Name	I/O	Signal Meaning	Timing Comments
TD[0-(N-1)][1]	O	Transmit Data - The transmit data is a unidirectional point to point bus designed to transmit the packet information. The TD bus of one device is connected to the RD bus of the receiving device. TD[0] is used in 1x mode.	Clocking is embedded in data using 8B/10B encoding.
$\overline{\text{TD}}$[0-(N-1)][1]	O	Transmit Data complement—These signals are the differential pairs of the TD signals.	
RD[0-(N-1)][1]	I	Receive Data - The receive data is a unidirectional point to point bus designed to receive the packet information. The RD bus of one device is connected to the TD bus of the receiving device. RD[0] is used in 1x mode.	
$\overline{\text{RD}}$[0-(N-1)][1]	I	Receive Data complement—These signals are the differential pairs of the RD signals.	

NOTES:
1. N has legal values of 1, 2, 4, 8, and 16

7.3 Serial RapidIO Interface Diagrams

Figure 7-1 shows the signal interface diagram connecting two 1x devices together with the RapidIO LP-Serial interconnect.

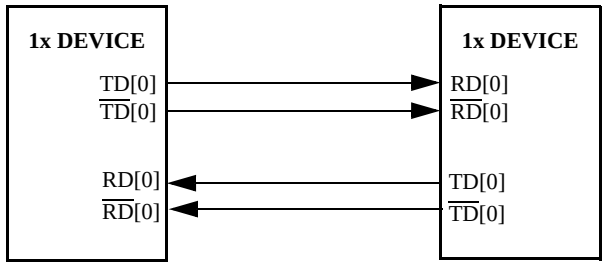


Figure 7-1. RapidIO 1x Device to 1x Device Interface Diagram

Figure 7-2 shows the signal interface diagram connecting two Nx devices together with the RapidIO LP-Serial interconnect.

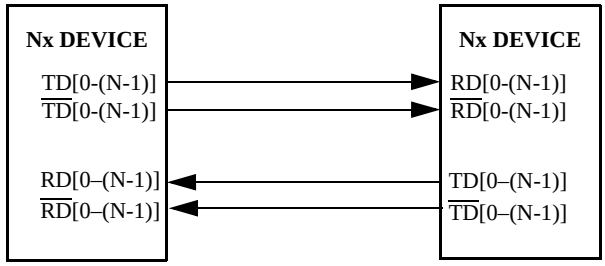


Figure 7-2. RapidIO Nx Device to Nx Device Interface Diagram

Figure 7-3 shows the connections between a Nx LP-Serial device and a 1x LP-Serial device.

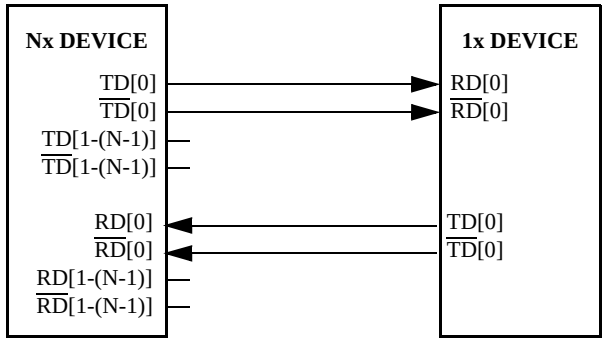


Figure 7-3. RapidIO Nx Device to 1x Device Interface Diagram

Chapter 8 Common Electrical Specifications

8.1 Introduction

The chapter defines the common electrical specifications for the LP-Serial physical layer. Chapter 9, "1.25Gbaud, 2.5Gbaud, and 3.125Gbaud LP-Serial Links" defines Level I links compatible with the 1.3 version of the Physical Layer Specification, that supports baud rates of 1.25, 2.5, and 3.125Gbaud. Chapter 10, "5Gbaud and 6.25Gbaud LP-Serial Links" defines Level II links that support baud rates of 5Gbaud and 6.25Gbaud.

A Level I link shall:

- allow 1.25Gbaud, 2.5Gbaud, or 3.125Gbaud baud rates
- supports AC coupling
- supports hot plug
- supports short run (SR) and long run (LR) links achieved with two transmitters
- support single receiver specification that will accept signals from both the short run and long run transmitter specifications
- achieve Bit Error Ratio of lower than 10^{-12} per lane

A Level II link shall:

- allow 5 or 6.25 Gbaud baud rates
- supports AC coupling and optional DC coupling
- supports hot plug
- supports short run (SR), medium run (MR), and long run (LR) links achieved with two transmitters and two receivers
- achieves Bit Error Ratio of lower than 10^{-15} per lane but test requirements will be verified to 10^{-12} per lane.

Together, these specifications allow for solutions ranging from simple chip-to-chip interconnect to board-to-board interconnect driving two connectors across a backplane. The faster and wider electrical interfaces specified here are required to provide higher density and/or lower cost interfaces.

The short run defines a transmitter and a receiver that should be used mainly for chip-to-chip connections on either the same printed circuit board or across a single connector. This covers the case where connections are made to a mezzanine

(daughter) card. The smaller swings of the short run specification reduces the overall power used by the transceivers.

The long run defines a transmitter and receiver that use larger voltage swings and channel equalization that allows a user to drive signals across two connectors and backplanes.

The two transmitter specifications allows for a medium run specification that also uses larger voltage swings that are capable of driving signals across a backplane but simplifies the receiver requirements to minimize power and complexity. This option has been included to allow the system integrator to deploy links that take advantage of either channel materials and/or construction techniques that reduce channel loss to achieve lower power systems.

It is also a goal of this specification to enable the inter-operability of Level I and Level II links to allow newer devices to be used with existing legacy devices.

All unit intervals are specified with a tolerance of ±100ppm. The worst case frequency difference between any transmit and receive clock is 200ppm.

The electrical specifications are based on loss, jitter, and channel cross-talk budgets and defines the characteristics required to communicate between a transmitter and a receiver using nominally 100Ω differential copper signal traces on a printed circuit board. Rather than specifying materials, channel components, or configurations, this specification focuses on effective channel characteristics. Hence a short length of poorer material should be equivalent to a longer length of premium material. A 'length' is effectively defined in terms of its attenuation rather than physical distance.

The RapidIO specification defines applicable data characteristics (e.g. DC balance, transition density, maximum run length), channel models and compliance points/parameters supporting the physical run and conditions.

Finally it is assumed that the link designer has taken care to minimize reflections and crosstalk so that the link can be sufficiently equalized with the transmitter and receiver chosen.

8.2 References

1. IEEE Standard 802.3ae-2002. “Standard for Information technology-Telecommunications and information exchange between systems-Local and metropolitan area networks-Special Requirements. Part 3: Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specification. Amendment: Media Access Control (MAC) Parameters, Physical Layers, and Management Parameters for 10 Gb/s Operation”, IEEE Std. 802.3ae-2002, August 30, 2002.
2. Optical Internetworking Forum “Common Electrical I/O (CEI) - Electrical and Jitter Interoperability Agreements for 6G+ bps and 11G+ bps I/O”, IA # OIF-CEI-02.0, January 28, 2005.

3. ITU-T Recommendation O.150 May 1996 and corrigendum May 2002. General requirements for instrumentation for performance measurements on digital transmission equipment.
4. Low Voltage Differential Swing (LVDS), ANSI/TIA/EIA-644-A-2001
5. Optical Internetworking Forum, OIF 2002.507.01 - High Speed Backplane (HSB) Interface Electrical Specification for 5-6.375Gbps Baud Rates over Currently Existing Communications Backplanes.

8.3 Abbreviations

Table 8-1. Abbreviations

Abbreviation	Meaning
BER	Bit Error Ratio
BERT	Bit Error Ratio Test or Tester
BUJ	Bounded Uncorrelated Jitter
CBGJ	Correlated Bounded Gaussian Jitter
CBHPJ	Correlated Bounded High Probability Jitter
CEI	Common Electrical I/O
CDF	Cumulative Distribution Function
CDR	Clock Data Recovery
CID	Consecutive Identical Digits
CML	Current Mode Logic
Cn	Cursor number
DCD	Duty Cycle Distortion
dB	Decibel
DDJ	Data Dependent Jitter
DFE	Decision Feedback Equalizer
DJ	Deterministic Jitter
DUT	Device Under Test
EMI	Electro-Magnetic Interference
erf	error function
erfinv	inverse error function
ESD	Electro-Static Discharge
FEXT	Far End Cross Talk
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
FR-4	Fire Retardant 4 Glass Reinforce Epoxy Laminate PCB material
Gbps	Giga bits per second
GJ	Gaussian Jitter
Gbaud	Giga symbols per second

Table 8-1. Abbreviations

Abbreviation	Meaning
HF	High Frequency
HPF	High Pass Filter
HPJ	High Probability Jitter
IA	Implementation Agreement
ISI	Inter-Symbol Interference
LMS	Least Mean Square
LPF	Low Pass Filter
LVDS [4]	Low Voltage Differential Signal
LR	Long Run
mA	milli-Amp
MR	Medium Run
mV	milli-Volt
NEXT	Near End Cross Talk
NRZ	Non Return to Zero
PCB	Printed Circuit Board
PDF	Probability Distribution Function
PECL	Positive Emitter Coupled Logic
PJ	Periodic Jitter
pp	Peak to Peak
ppd	Peak to Peak Differential (as in 300mVppd)
PLL	Phase Locked Loop
ps	pico second
PRBS	Pseudo Random Bit Stream
Q	Inverse error function
RJ	Random Jitter
RV	Random Variable
RX	Receiver
R_Zvtt	Resistance of termination to Vtt
S11 and S22	reflection coefficient
S21	transmission coefficient
SCC11 and SCC22	Common mode reflection coefficients
SCD11 and SCD22	Differential to common mode conversion coefficient
SDD11 and SDD22	Differential reflection coefficients
SDC11 and SDC22	Common mode to differential conversion coefficient
SJ	Sinusoidal Jitter
SR	Short Run
sym/s	symbols/second

Table 8-1. Abbreviations

Abbreviation	Meaning
TJ	Total Jitter
TDM	Time Division Multiplexed data
TFI	TDM Fabric to Framer Interface
TX	Transmitter
UBHPJ	Uncorrelated Bounded High Probability Jitter
UI	Unit Interval = 1/(baud rate)
UUGJ	Uncorrelated Unbounded Gaussian Jitter
Vtt	Termination Voltage
XAUI	10 Gigabit Attachment Unit Interface

8.4 Definitions

Table 8-2. General Definitions

Parameter	Description
Bit Error Ratio	A parameter that reflects the quality of the serial transmission and detection scheme. The Bit Error Ratio is calculated by counting the number of erroneous bits output by a receiver and dividing by the total number of transmitted bits over a specified transmission period.
Baud rate	Is a measure of the number of times per second a signal in a communications channel changes state. The state is usually voltage level, frequency, or phase angle. It is named after Émile Baudot, the inventor of the Baudot code for telegraphy.
Channel	In this specification Channel shall mean electrical differential channel. The channel is combination of electrical interconnects that together form the signal path from reference points T to R - see Figure 8-11. The channel will typically consist of PCB traces, via holes, component attachment pads and connectors. A characteristic of a signal channel is the complex characteristic impedance Z.
Common Mode Voltage	Average of the Vhigh and Vlow voltage levels - see Figure 8-1.
Confidence level	The use of this definition shall be understood as being with reference to a Gaussian distribution
Differential Termination Resistance mismatch	The difference in the DC termination resistance with respect to ground of any two signals forming a differential pair. Usually due to large process spread the absolute termination resistance is specified relatively loose, e.g. 20% where the relative difference of resistors of the same device will be much less, e.g 5%. This parameter is used to specify the relative difference tighter than the overall resistance for the purpose of minimizing differential signal mode conversion
Gaussian	A statistical distribution (also termed “normal”) characterized by populations that are not bound in value and have well defined “tails”. The term “random” in this document always refers to a Gaussian distribution.
Golden PLL	Refers to a defined clock extraction unit which phase tracks the inherent clock present in a data signal. The phase tracking bandwidth is usually defined in terms of a corner frequency and if not defined with a corner frequency of baud/1667, a roll off of 20 dB/dec and <0.1 dB peaking
Golden Channel	Refers to an electrical channel which is usually identified using a channel compliancy methodology and is used in the testing of transmitters and receivers

Table 8-2. General Definitions

Parameter	Description
Intersymbol Interference	Data dependent deterministic jitter caused by the time differences required for the signal to arrive at the receiver threshold when starting from different places in bit sequences (symbols). For example when using media that attenuates the peak amplitude of the bit sequence consisting of alternating 0, 1, 0, 1... more than peak amplitude of the bit sequence consisting of 0, 0, 0, 0, 1, 1, 1, 1... the time required to reach the receiver threshold with the 0, 1, 0, 1... is less than required from the 0, 0, 0, 0, 1, 1, 1, 1... The run length of 4 produces a higher amplitude which takes more time to overcome when changing bit values and therefore produces a time difference compared to the run length of 1 bit sequence. When different run lengths are mixed in the same transmission the different bit sequences (symbols) therefore interfere with each other. Intersymbol Interference is expected whenever any bit sequence has frequency components that are propagated at different rates by the transmission media.
Lane	A single RapidIO Channel
Link	A functional connection between the Tx and Rx ports of 2 components, that can be multiple or parallel RapidIO Lanes defined as 1:N. The definition a Link does not imply duplex operation.
Non-transparent applications	Defines an application where the high frequency transmit jitter of a device is defined independently to the high frequency jitter present at any data input of the same device
Skew	The constant portion of the difference in the arrival time between the data of any two in-band signals.
Stressed Signal (or) Stressed Eye	In order to test the tolerance of a receiver a stressed signal or eye is defined which when applied to the receiver must be received with the defined Bit Error Rate. The stressed signal or eye is defined in terms of its horizontal closure or jitter and amplitude normally in conjunction with an eye-mask.
Transparent applications	Defines an application where the high frequency transmit jitter of a device is dependent on the high frequency jitter present at one or more of the data inputs of the same device
Symbol	Unit of information conveyed by a single state transition in the medium
Symbol spaced	Describes a time difference equal to the nominal period of the data signal
Unit Interval	One nominal bit period for a given signaling speed. It is equivalent to the shortest nominal time between signal transitions. UI is the reciprocal of Symbol.

Table 8-3. Jitter and Wander Definitions

Parameter	Description
Correlated Bounded Gaussian Jitter	Jitter distribution where the value of the jitter shows a correlation to the signal level being transmitted. The distribution is quantified, using a Gaussian approximation, as the gradient of the bathtub linearization at the Bit Error Rate of interest. R_RJ = R_GJ
Correlated Bounded High Probability Jitter	Jitter distribution where the value of the jitter shows a strong correlation to the signal level being transmitted. This jitter may considered as being equalizable due to its correlation to the signal level. Was called Data Dependent Jitter in earlier specification revisions.
Correlated Wander	Components of wander that are common across all applicable in band signals.
Duty Cycle Distortion	The absolute value of the difference in the average width of a'1' symbol or a '0' symbol and the ideal periodic time in a clock-like repeating 0,1,0,1 sequence. Duty Cycle Distortion is part of the CBHPJ distribution and is measured at the time-averaged signal level.
Gaussian Jitter	An overall term that defines a jitter distribution that at the BER of interest e.g. 1e-15 still shows a Gaussian distribution. Unless otherwise specified Gaussian Jitter is the RMS sum of CBGJ and UUGJ. Was called Random Jitter in earlier specification revisions.

Table 8-3. Jitter and Wander Definitions

Parameter	Description
High Probability Jitter	Jitter distribution that at the BER of interest is approximated by a dual dirac. Unless otherwise specified High Probability Jitter is the sum of UBHPJ, CBHPJ, PJ, SJ, DCD. The distribution is quantified, using a dual dirac approximation, as the offset of the bathtub linearization at the Bit Error Rate of interest. Was called Deterministic Jitter in earlier specification revisions.
Jitter	Jitter is deviation from the ideal timing of an event at the mean amplitude of the signal population. Low frequency deviations are tracked by the clock recovery circuit, and do not directly affect the timing allocations within a bit interval. Jitter that is not tracked by the clock recovery circuit directly affects the timing allocations in a bit interval. Jitter is phase variations in a signal (clock or data) after filtering the phase with a single pole high pass filter with the -3 dB point at the jitter corner frequency.
Jitter Generation	Jitter generation is the process whereby jitter appears at the output port in the absence of applied input jitter at the input port.
Jitter RMS	The root mean square value or standard deviation of jitter. See clause 2 for more information.
Jitter Transfer	The ratio of the jitter output and jitter input for a component, device, or system often expressed in dB. A negative dB jitter transfer indicates the element removed jitter. A positive dB jitter transfer indicates the element added jitter. A zero dB jitter transfer indicates the element had no effect on jitter. The ratio should be applied separately to deterministic components and Gaussian (random) jitter components.
Peak-to-Peak Jitter	For any type of jitter, Peak to Peak Jitter is the full range of the jitter distribution that contributes within the specified BER.
Periodic Jitter	A sub form of HPJ that defines a jitter which has a single fundamental harmonic plus possible multiple even and odd harmonics.
Relative Wander	Components of wander that are uncorrelated between any two in band signals (See Figure 8-6)
Sigma	Refers to the standard deviation of a random variable modelled as a Gaussian Distribution. When used in reference to jitter, it refers to the standard deviation of the Gaussian Jitter component(s). When used in reference to confidence levels of a result refers to the probability that the result is correct given a Gaussian Mode, e.g. a measured result with 3 sigma confidence level would imply that 99.9% of the measurements are correct.
Sinusoidal Jitter	A sub form of HPJ that defines a jitter which has a single frequency harmonic.
Total Jitter	Sum of all jitter components.
Total Wander	The sum of the correlated and uncorrelated wander. (See Figure 8-7)
Unbounded Gaussian Jitter	Jitter distribution that shows a true Gaussian distribution where the observed peak to peak value has an expected value that grows as a function of the measurement time. This form of jitter is assumed to arise from phase noise random processes typically found in VCO structures or clock sources. It is usually quantified as either the Root Mean Square (RMS) or Sigma of the Gaussian distribution, or as the expected peak value for a given measurement population. (Formally defined as T_RJ)
Uncorrelated Bounded High Probability Jitter.	Jitter distribution where the value of the jitter show no correlation to any signal level being transmitted. Formally defined as T_DJ.
Uncorrelated Wander	Components of wander that are not correlated across all applicable in band signals.
Wander	The peak to peak variation in the phase of a signal (clock or data) after filtering the phase with a single pole low pass filter with the -3db point at the wander corner frequency. Wander does not include skew.

8.4.1 Definition of Amplitude and Swing

LP-Serial links use differential signaling. This section defines the terms used in the description and specification of these differential signals. Figure 8-1 shows how these signals are defined and sets out the relationship between absolute and differential voltage amplitude. The figure shows waveforms for either the transmitter output (TD and $\overline{TD}$) or a receiver input (RD and $\overline{RD}$).

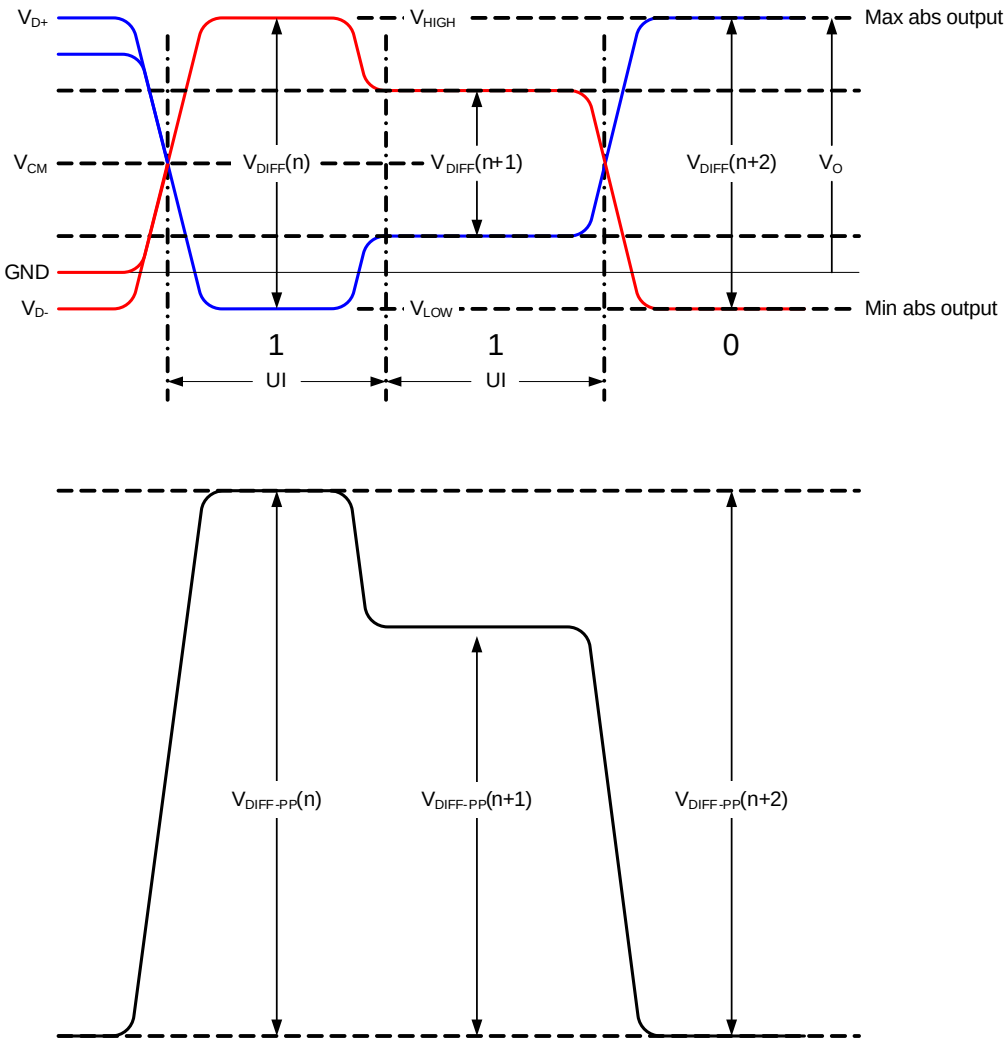


Figure 8-1. Definition of Transmitter Amplitude and Swing

Each signal swings between the voltages V_{HIGH} and V_{LOW} where

$$\mathbf{V_{HIGH} > V_{LOW}}$$

The differential voltage, V_{DIFF} , is defined as

$$\mathbf{V_{DIFF} = V_{D+} - V_{D-}}$$

where V_{D+} is the voltage on the positive conductor and V_{D-} is the voltage on the negative conductor of a differential transmission line. V_{DIFF} represents either the differential output signal of the transmitter, V_{OD}, or the differential input signal of the receiver, V_{ID} where

$$\mathbf{V_{OD} = V_{TD} - V_{\overline{TD}}}$$

and

$$\mathbf{V_{ID} = V_{RD} - V_{\overline{RD}}}$$

The common mode voltage, V_{CM}, is defined as the average or mean voltage present on the same differential pair. Therefore

$$\mathbf{V_{CM} = |V_{D+} + V_{D-}|/2}$$

The maximum value, or the peak-to-peak differential voltage, is calculated on a per unit interval and is defined as

$$\mathbf{V_{DIFFp-p} = 2 \times max|V_{D+} - V_{D-}|}$$

because the differential signal ranges from $V_{D+} - V_{D-}$ to $-(V_{D+} - V_{D-})$

To illustrate these definitions using real values, consider the case of a CML (Current Mode Logic) transmitter and each of its outputs, TD and $\overline{TD}$, has a swing that goes between V_{HIGH} = 2.5V and V_{LOW} = 2.0V, inclusive. Using these values the common mode voltage is calculated to be 2.25 V and the single-ended peak voltage swing of the signals TD and $\overline{TD}$ is 500mVpp. The differential output signal ranges between 500mV and -500mV, inclusive. therefore the peak-to-peak differential voltage is 1000mVppd.

8.4.2 Transmitter (Near-End) Template

For each baud rate at which the LP-Serial transmitter is specified to operate, the output eye pattern for transition symbols shall fall entirely within the unshaded portion of the Transmitter (near-end) Output Compliance Mask defined in Figure 8-2. Specific parameter values are called out in the sections that follow.

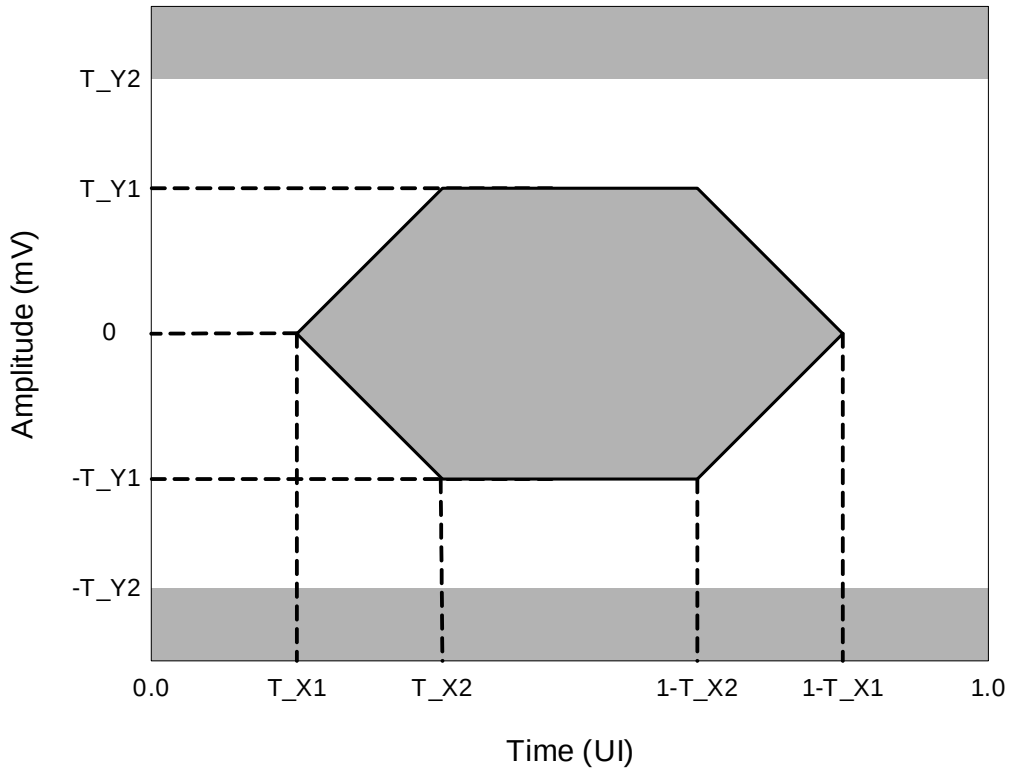


Figure 8-2. Transition Symbol Transmit Eye Mask

The output eye pattern of a LP-Serial transmitter that implements de-emphasis (to equalize the link and reduce intersymbol interference) need only comply with the Transition Transmitter Output Compliance Mask when there is a symbol transition from 1 to 0 or 1 to 0 or when pre-emphasis is disabled or minimized

For 5Gbaud and 6.25Gbaud links the Transmitters eye mask will also be evaluated during the steady-state where there are no symbol transitions, e.g a 1 followed by a 1 or a 0 followed by a 0, and the signal has been de-emphasized. This additional transmitter eye mask constraint is shown in Figure 8-3.

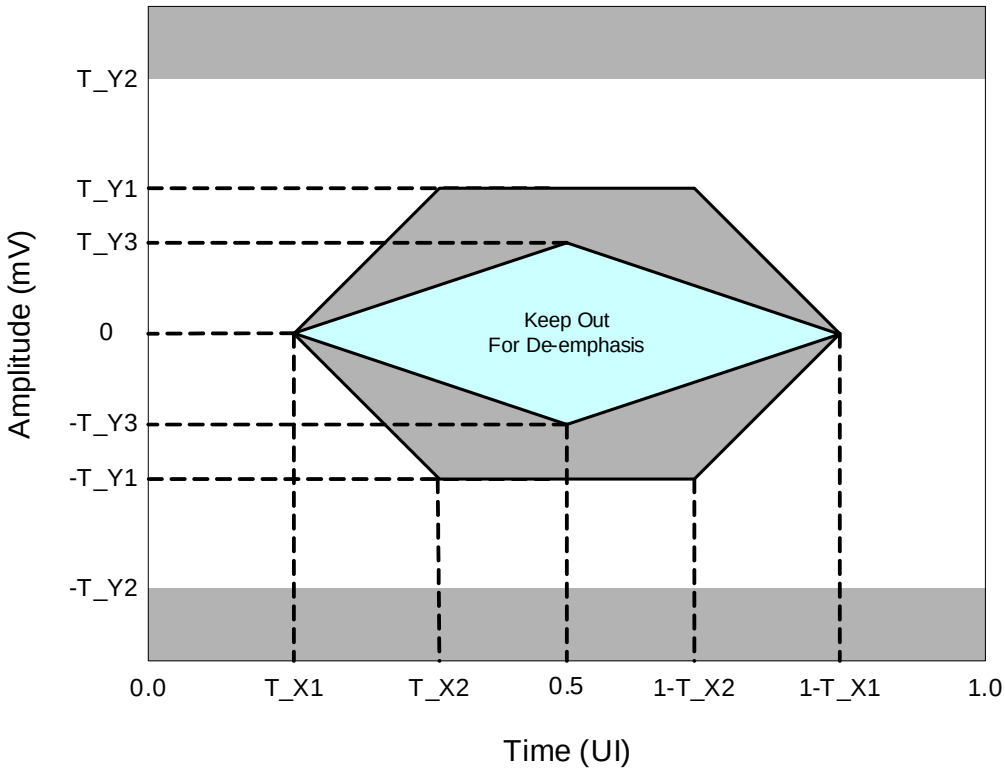


Figure 8-3. Transition and Steady State Symbol Eye Mask

During the steady-state the eye mask prevents the transmitter from de-emphasizing the low frequency content of the data too much and limiting the available signal-to-noise at the receiver.

The de-emphasis introduces a jitter artifact that is not accounted for in this eye mask. This additional jitter is the result of the finite rise/fall time of the transmitter and the non-uniform voltage swing between the transitions. This additional deterministic jitter must be accounted for as part of the high probability jitter.

Table 8-4 defines the standard parameters that will be specified for every transmitter.

Table 8-4. Transmitter Output Jitter Specification

Characteristic	Symbol	Conditions	Min	Typ	Max	Units
Total Jitter	T_TJ					UIpp
Eye Mask	T_X1					UI
Eye Mask	T_X2					UI
Eye Mask	T_Y1					mV

NOTES:
Uncorrelated Unbounded Gaussian Jitter must be defined with respect to specified BER of 10^{-12}, Q=7.03 for 1.25Gbaud, 2.5Gbaud, and 3.125Gbaud links
Uncorrelated Unbounded Gaussian Jitter must be defined with respect to specified BER of 10^{-15}, Q=7.94 for 5GBaud and 6.25Gbaud links

Table 8-4. Transmitter Output Jitter Specification

Characteristic	Symbol	Conditions	Min	Typ	Max	Units
Eye Mask	T_Y2					mV
Eye Mask (5Gbaud and 6.25Gbaud only)	T_Y3					mV
NOTES: Uncorrelated Unbounded Gaussian Jitter must be defined with respect to specified BER of 10^{-12}, Q=7.03 for 1.25Gbaud, 2.5Gbaud, and 3.125Gbaud links Uncorrelated Unbounded Gaussian Jitter must be defined with respect to specified BER of 10^{-15}, Q=7.94 for 5GBaud and 6.25Gbaud links						

Note: In previous versions of the RapidIO LP-Serial specification different symbols names were used to define the time and voltage points on eye masks. Table 8-5 can be used as a cross reference for the transmitter eye mask symbol names.

Table 8-5. Transmitter Eye Mask Cross Reference

Current Version	1.3 Version
T_Y1	V_{DIFF} min
T_Y2	V_{DIFF} max
T_Y3	N/A
T_X1	A
T_X2	B

8.4.3 Receiver (Far-End) Template

The receiver (far-end) template has two definitions based on Level I and Level II links.

8.4.3.1 Level I Receiver Template

Figure 8-4 illustrates the definition in a Level I receiver eye template.

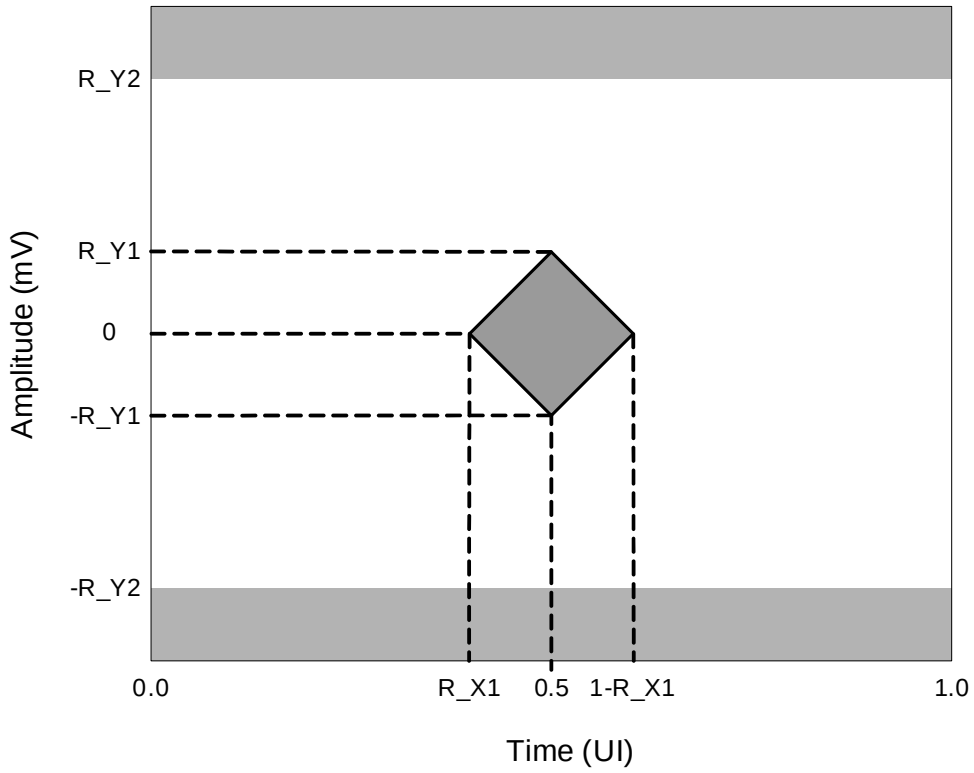


Figure 8-4. Level I Receiver Input Mask

Table 8-8 defines the standard parameters that will be specified for Level I receivers which have an open eye at the far-end. The termination conditions used to measure the received eye are defined Section 8.5.13.

Table 8-6. Level I Receiver Jitter Specification

Characteristic	Symbol	Conditions	Min	Typ	Max	Units
Total Jitter	R_TJ					UIpp
Eye Mask	R_X1					UI
Eye Mask	R_X2					UI
Eye Mask	R_Y1					mV
NOTES: Uncorrelated Unbounded Gaussian Jitter must be defined with respect to specified BER of 10^{-12}, Q=7.03 for 1.25Gbaud, 2.5Gbaud, and 3.125Gbaud links						

Also in the previous versions of the RapidIO LP-Serial specification different symbols names were used to define the time and voltage points on eye masks. Table 8-8 can be used as a cross reference for the receiver eye mask.

Table 8-7. Receiver Eye Mask Cross Reference

Current Version	1.3 Version
R_Y1	V_{DIFF} min
R_Y2	V_{DIFF} max
R_X1	A
R_X2	B

8.4.3.2 Level II Receiver Template

For a Level II link the receiver mask it is defined as is defined in Figure 8-5. Specific parameter values for both masks are called out in the sections that follow.

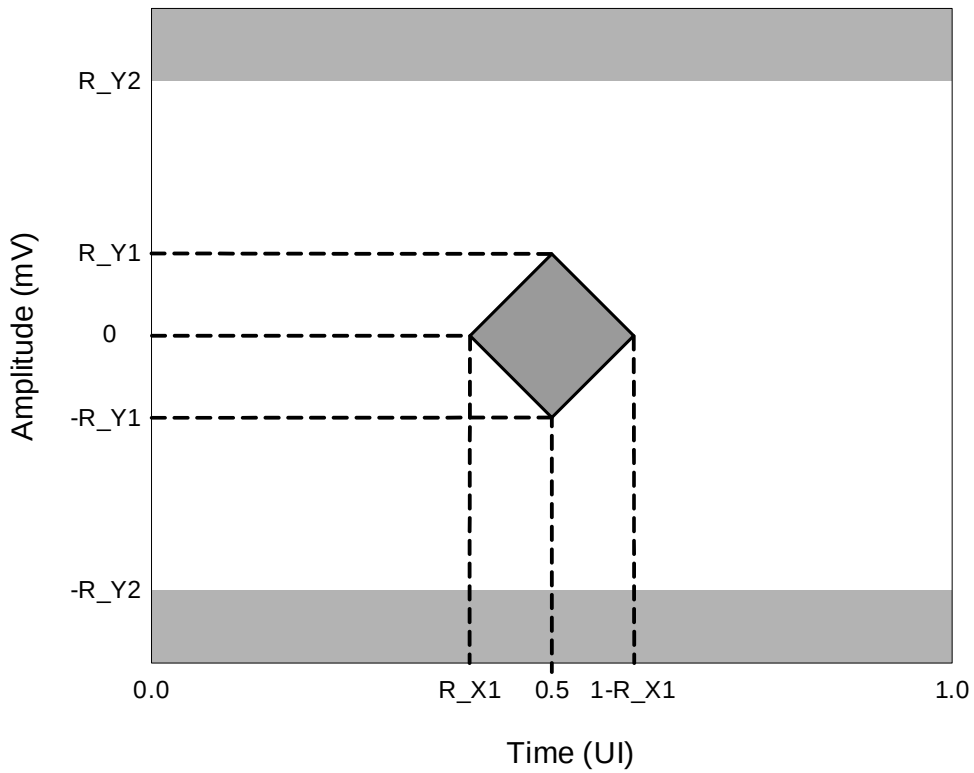


Figure 8-5. Receiver Input Mask

Table 8-8 defines the standard parameters that will be specified for receivers that have an open eye at the far-end. The termination conditions used to measure the received eye are defined Section 8.5.13.

Table 8-8. Level II Receiver Jitter Specification

Characteristic	Symbol	Conditions	Min	Typ	Max	Units
Total Jitter	R_TJ					UIpp
Eye Mask	R_X1					UI
Eye Mask	R_Y1					mV
NOTES: Uncorrelated Unbounded Gaussian Jitter must be defined with respect to specified BER of 10^{-15}, Q=7.94 for 5Gbaud and 6.25Gbaud links						

8.4.4 Definition of Skew and Relative Wander

See Figure 8-6 for an illustration of skew and relative wander. The definitions appear in Table 8-3.

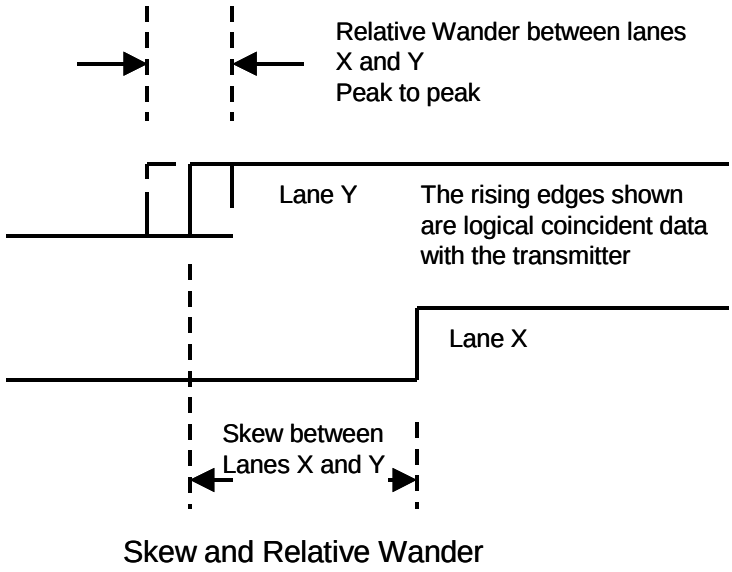


Figure 8-6. Skew and Relative Wander Between in Band Signals

See Figure 8-7 for an illustration of total wander in a signal. The definition appears in Table 8-3.

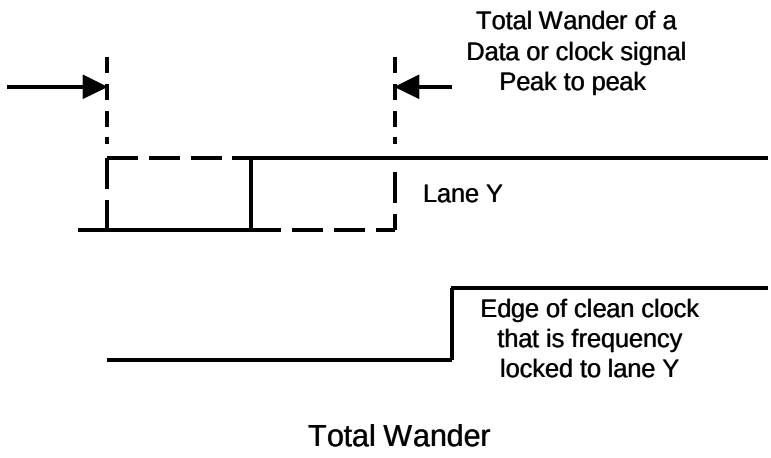


Figure 8-7. Total Wander of a Signal

8.4.5 Total Wander Mask

Total wander specifications should be considered as accumulated low frequency jitter. As modern CDRs are digitally based they show a corner tracking frequency plus slew limitation which has been guaranteed, therefore for jitter tolerance testing the total wander needs to be spectrally defined to ensure correct operation.

To this end, for jitter tolerance testing, the wander is considered a sinusoidal jitter source as shown in Figure 8-8 below.

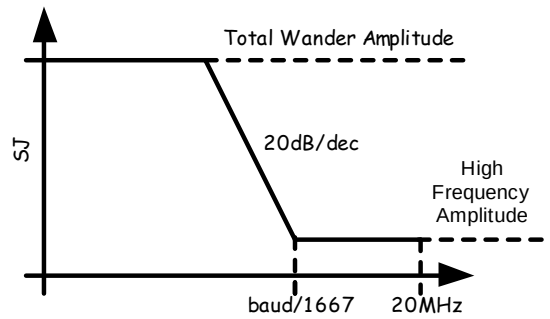


Figure 8-8. Total Wander Mask

At higher frequency this jitter source is used to ensure margin in the high frequency jitter tolerance of the receiver. At lower frequencies the higher SJ should then be tracked by the CDR.

8.4.6 Relative Wander Mask

Specifically for interfaces defining relative wander, Figure 8-9 is also defined in terms of a sinusoidal jitter sources as shown below.

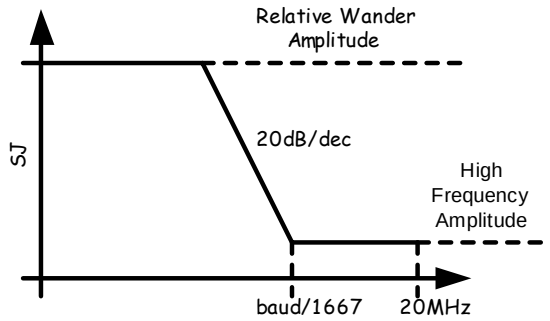


Figure 8-9. Relative Wander Mask

8.4.7 Random Jitter Mask

To ensure that the random jitter modulation of stressed signals is above the CDR bandwidth and therefore untracked, the filter mask shown in Figure 8-10 shall be applied where necessary.

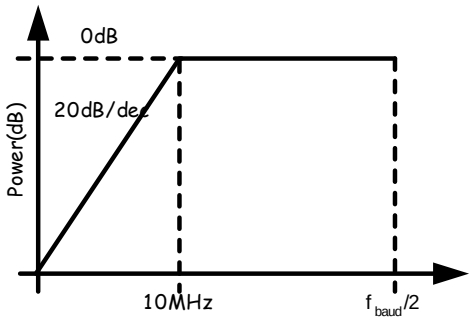


Figure 8-10. Random Jitter Spectrum

8.4.8 Defined Test Patterns

The data test patterns are unique to the two levels of link and will be defined in the sections specific to these.

8.4.9 Reference Model

The LP-Serial electrical reference model is defined in Figure 8-11. Note that the RX and TX blocks include all off-chip components associated with the respective function. Thus the reference points T and R are defined to be the component edge of the transmitter and receiver respectively.

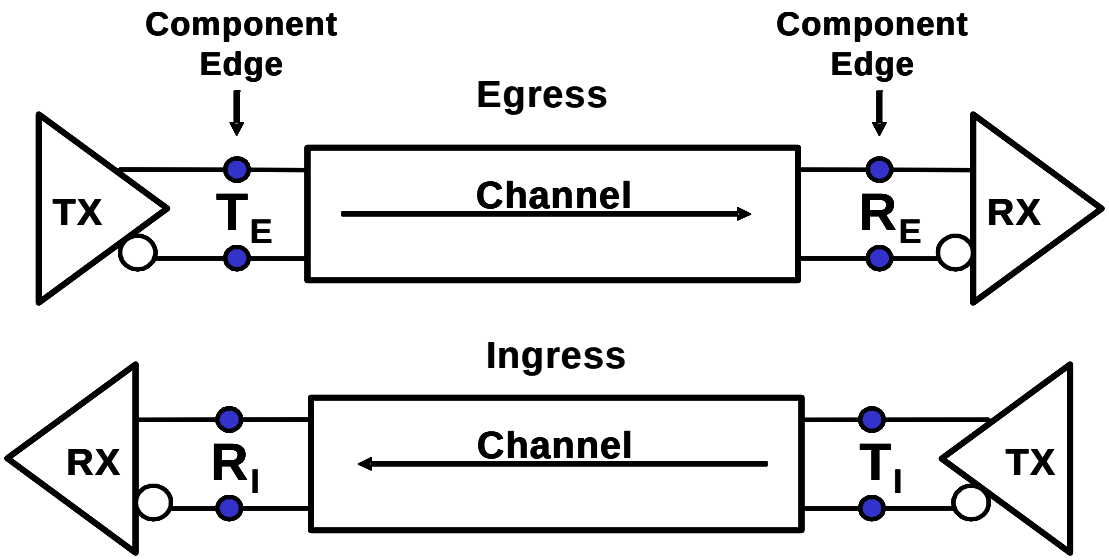


Figure 8-11. Reference Model

Note: Through out this specification the terms ‘near’ and ‘far’ are used to describe aspects of the channel. **Near-end** will always be used to refer to the end of the channel attached to the transmitter, e.g. T_E or T_I, independent of if it is the egress or ingress channel. **Far-end** will be used to refer to the end of the channel attached to the receiver, e.g. R_I or R_E.

8.5 Common Electrical Specification

8.5.1 Introduction

This section specifies electrical parameters and attributes common to all links. In the event of a difference between an individual link and these general requirements, the respective individual link shall prevail.

The LP-Serial 1.25Gbaud, 2.5Gbaud, and 3.125Gbaud Electrical specifications are guided by the XAUI electrical interface specified in Clause 47 of IEEE 802.3ae-2002.[1]

The LP-Serial 5Gbaud and 6.25Gbaud Electrical specifications are based upon the Optical Internetworking Forum's Common Electrical Interface [2], referred to henceforth as CEI.

CEI includes the following sections that are the basis for the LP-Serial RapidIO 5Gbaud and 6.25Gbaud interfaces:

- CEI-6G-SR clause 6 specification for data lane(s) that support bit rates from 4.976 to 6.375Gbaud over Printed Circuit Boards with physical runs from 0 to 20cm and up to 1 connector. CEI-6G-SR forms the basis for the LP-Serial 5Gbaud and 6.25Gbaud Short Run Interface electrical specifications. RapidIO has enhanced this electrical specification to include a continuous-time equalizer with one zero and one pole.
- CEI-6G-LR Clause 7 specification for data lane(s) that support bit rates from 4.976 to 6.375Gbaud over Printed Circuit Boards with physical runs from 0 to 100cm and up to 2 connectors. CEI-6G-LR forms the basis for the LP-Serial 5Gbaud and 6.25Gbaud Long Run Interface electrical specifications.
- RapidIO has added a specification for data lane(s) that supports bit rates from 5 to 6.25Gbaud over Printed Circuit Boards and physical runs from 0 to 60cm and up 2 connectors. The CEI-6G-LR transmitter and a continuous-time receiver with one zero and one pole form the basis for the LP-Serial 5Gbaud and 6.25Gbaud Medium Run Interface electrical specifications.

  Note: The OIF CEI documentation uses the term “reach” to describe the length of the channel. Here “run” is used to maintain consistency with the RapidIO 1.3 interconnect specification.

  While the OIF CEI documentation defines support for 4.976 to 6.375Gbaud RapidIO only supports 5.0 and 6.25 Gbaud data rates

8.5.2 Data Patterns

There is a requirement that the link data follow 8B/10B encoding rules and when specified raw data scrambling requirements as defined in Chapter 4, "PCS and PMA Layers", to ensure proper operation. The predicted BER performance and jitter requirements are only valid when this assumption is satisfied. If all of these conditions are not met, then the link may not work to the full distance, or meet the BER, or in fact work at all.

8.5.3 Signal Levels

The signal is a low swing differential interface. This implies that the receiver has a wide common mode range (within the maximum absolute input voltages). All devices must support load type 0 defined in Table 8-9. Level II SR devices can optionally support any or all of the other 3 load types while Level II MR and LR devices can optionally support load type 1.

Table 8-9. Definition of Load Types

Characteristics	Load Type 0	Load Type 1	Load Type 2	Load Type 3	Units
R_Zvtt	>1k	<30	<30	<30	Ω
Nominal Vtt	undefined	1.2	1.0	0.8	V

This type of differential interface allows for inter-operability between components operating from different supply voltages and different I/O types (CML, LVDS-like, PECL, etc.). Low swing differential signaling provides noise immunity and improved electromagnetic interference (EMI). Differential signal swings are defined in following sections and depend on several factors such as transmitter pre-equalization, receiver equalization, and transmission line losses.

8.5.4 Bit Error Ratio

8.5.4.1 Level I Bit Error Ratio

The LP-Serial 1.25Gbaud, 2.5Gbaud, and 3.125Gbaud interface lanes will operate with a Bit Error Ratio (BER) of 10^{-12}.

It should be noted that most modern system are capable of achieving the improved BER required in Level II links.

8.5.4.2 Level II Bit Error Ratio

The LP-Serial 5Gbaud and 6.25Gbaud interface lanes will operate with a Bit Error Ratio (BER) of 10^{-15} (with a test requirement to verify 10^{-12}). See Clause 2 of CEI for more information on the jitter model and how to measure BER.

8.5.5 Ground Differences

The maximum ground difference between the transmitter and the receiver shall be ±50mV for SR links and ±100mV for MR and LR links. This will affect the absolute maximum voltages at compliance point 'R'. If transmitter and receiver are on the same PCB with no intervening connectors, then the ground difference is approximately 0mV.

8.5.6 Cross Talk

Cross talk arises from coupling within the connectors, on the PCB, the package and the die. Cross talk can be categorized as either Near-End or Far-End cross talk (NEXT and FEXT). In either of these categories, the amount of cross talk is dependent upon signal amplitudes, signal spectrum, and trace/cable length. There can be many aggressor channels onto one victim channel, however typically only a few are dominant.

Further consideration of cross talk can be found in Annex A, “Transmission Line Theory and Channel Information (Informative)".

8.5.7 Transmitter Test Load

All transmitter characteristics should be implemented and measured to a differential impedance of 100Ω ±1% at DC with a return loss of better than 20dB from baud rate divided by 1667 to 1.5 times the baud rate, unless otherwise noted.

8.5.8 Transmitter Lane-to-Lane Skew

While the protocol layer will control some of the lane to lane skew, the electrical level for the lane-to-lane skew caused by the transmitter circuitry and associated routing is allowed up to be:

- less than 1000 ps for port widths less than or equal to 4 lanes
- less than 2UI + 1000 ps for port width with greater than 4 lanes

Hence, the total output (i.e. measured) lane-to-lane skew is to be specified in the protocol standards with the above skew taken into account. The transmitter lane-to-lane skew is only for the SerDes TX and does not include any effects of the channel.

8.5.9 Receiver Input Lane-to-Lane Skew

The maximum amount of lane-to-lane skew at the input pins of the receiver is determined by the ability of the receiver to resolve the difference between two successive ||A|| columns. Since the minimum number of non-||A|| columns between ||A|| columns is 16, the maximum lane skew that can be unambiguously corrected is the time it takes to transmit 7 code groups per lane. Therefore, the maximum lane-to-lane skew at the input pins of a receiver is calculated as:

(7 code groups) x (10 bits/code-group) x (1 UI/bit) x (ns/UI)

It is important to note that the total lane-to-lane skew specification includes the skew caused by the transmitter’s PCS and PMA (SerDes), the channel, the receivers’ PMA (SerDes) and PCS and any logic that is needed to create the aligned column of ||A|| at the receiving device.

8.5.10 Transmitter Short Circuit Current

The max DC current into or out of the transmitter pins when either shorted to each other or to ground shall be ±100mA when the device is fully powered up. From a hot swap point of view, the ±100mA limit is only valid after 10µs.

8.5.11 Differential Resistance and Return Loss, Transmitter and Receiver

The DC differential resistance shall be between 80 and 120Ω, inclusive.

The differential return loss shall be better than A0 from f0 to f1 and better than

A0 + Slope*log10(f/f1)

where f is the frequency from f1 to f2 (see Figure 8-12). Differential return loss is measured at compliance points T and R. If AC coupling is used, then all components (internal or external) are to be included in this requirement. The reference impedance for the differential return loss measurements is 100Ω.

Common mode return loss measurement shall be better than -6dB between a minimum frequency of 100MHz and a maximum frequency of 0.75 times the baud rate. The reference impedance for the common mode return loss is 25Ω.

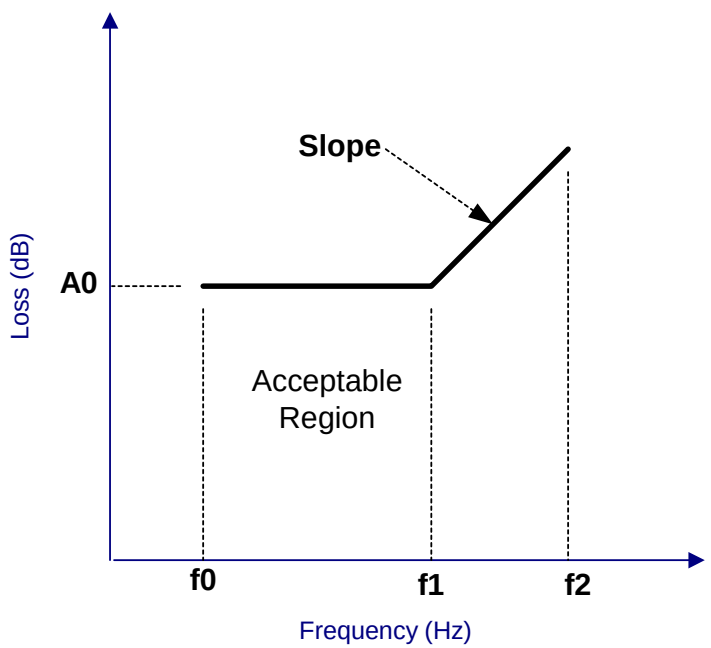


Figure 8-12. Transmitter and Input Differential Return Loss

8.5.12 Baud Rate Tolerance

The baud rates are defined to be 1.25Gbaud, 2.5Gbaud, 3.125Gbaud, 5Gbaud and 6.25Gbaud. Each interface is required to operate asynchronously with a tolerance of ±100ppm from the nominal baud rate.

Note: The minimum and maximum baud rates can be calculated as:

Baudrate∗(1 ± 100E-6)

8.5.13 Termination and DC Blocking

Each link requires a nominal 100Ω differential source termination at the transmitter and a nominal 100Ω differential load termination at the receiver. The terminations shall provide both differential and common mode termination to effectively absorb differential or common mode noise and reflections. Receivers and transmitters shall support AC coupling and may also optionally support DC coupling. AC Coupled receivers require a differential termination >1kΩ at DC (by blocking capacitors in or near receivers as shown in Figure 8-13 or by circuit means within the receiver). DC Coupled devices shall meet additional electrical parameters T_Vcm, R_Vrcm, R_Vtt, R_Zvtt. All termination components are included within the RX and TX blocks as shown in the reference model as defined in Section 8.4.9.

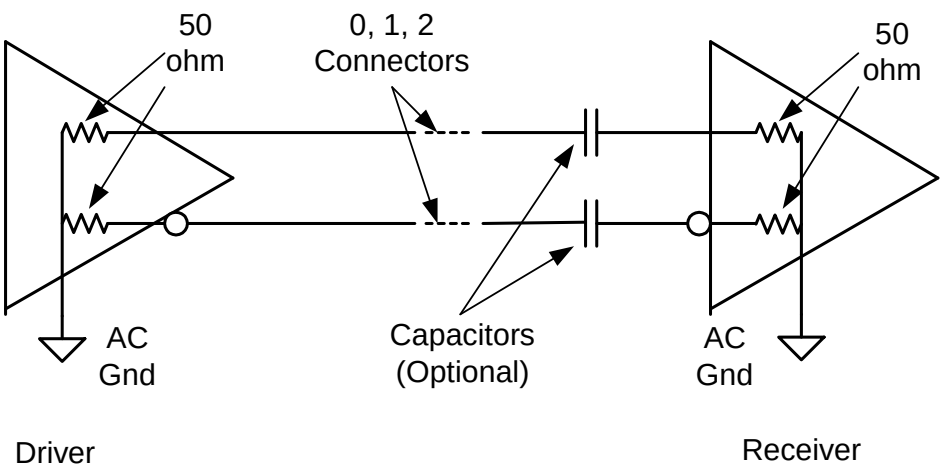


Figure 8-13. Termination Example

8.6 Pulse Response Channel Modelling

This section shall describe the theoretical background for channel modelling.

8.6.1 Generating a Pulse Response

Knowing the spectral transfer function for a channel allows the pulse response of the channel can be calculated using tools such as MATLAB®

The Pulse Response of the channel is the received pulse for an ideal square wave and is calculated by either

- convolving the pulse with the impulse response of the channel or
- multiplying the Fourier spectrum of the ideal transmitted square wave with the channel response and taking the inverse Fourier transform, where

f_{max} is difference between the maximum positive and minimum negative frequency

P is the number of equally space points in the frequency array

$tx(t)$ is the transmit signal pulse

$tx(\omega)$ is the transmit signal pulse in the frequency domain

$Tr(\omega)$ is the transfer function of the channel

$rx(t)$ is the resulting pulse response of the channel

$$t_{step} = \frac{1}{f_{max}}$$

$$t = t_{step} \cdot n$$

$$n = [1,P]$$

$$tx(t) = H(0) \cdot H(t_{period} - t)$$

$$rx(\omega) = tx(\omega) \cdot Tr(\omega)$$

$$rx(t) = ifft(rx(\omega))$$

8.6.2 Basic Pulse Response Definitions

A receive pulse response as calculated is graphically represented in Figure 8-14.

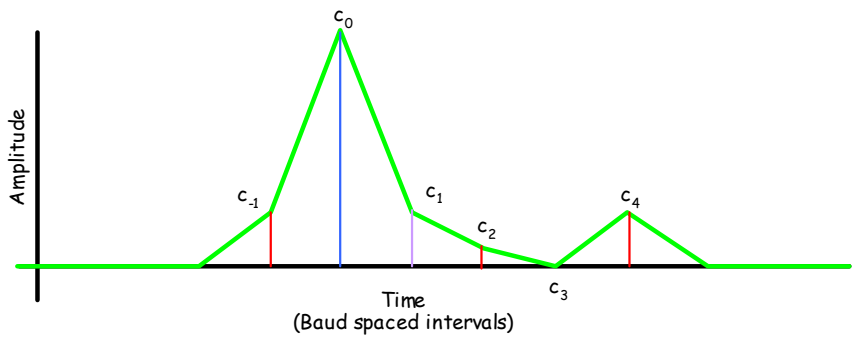


Figure 8-14. Graphical Representation of Receiver Pulse

Cursors are defined as being the amplitude of the received pulse at symbol spaces from the maximum signal energy at c_0, and extend to infinity in both negative and

positive time. The exact position of c_0 is arbitrary and is defined specifically by the various methodologies.

A precursor is defined as a cursor that occurs before the occurrence of the main signal c_0, i.e. c_n where n<0, usually convergences to zero within a small number of bits

A post cursor is defined as a cursor that occurs after the occurrence of the main signal c_0, i.e. c_n where n>0, and usually convergences to zero within twice the propagation time of the channel.

Given a deterministic data stream travelling across the channel, the superposition of the channel pulses give rise to Inter-Symbol Interference (ISI). This ISI has a maximum occurring for a worst case pattern, which for a channel response where all cursors are positive would be a single *1* or *0* in the middle of a long run of *0*s or *1*s respectively. This maximum is referred to Total Distortion.

$$\Theta = \sum_{(n=-\infty),(n\neq 0)}^{n=\infty} |c_n|$$

Due to ISI an enclosure in the time domain also occurs which can be determined by either running exhaustive simulations or simulations with determined worst case patterns. For the case where the ISI is so large that the eye is closed, Inherent Channel Jitter has no meaning.

8.6.3 Transmitter Pulse Definition

A transmitter is defined by its ability to generate a transmit pulse. A single *1* transmit symbol has different amplitudes at symbol space intervals, t_n, where post taps have n>0, and pre-taps have n<0.

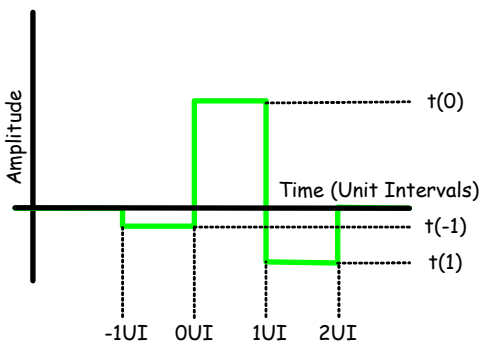


Figure 8-15. Transmit Pulse

When a pulse train is transmitted the exact transmitted amplitude is therefore the superposition of the pulses from the previous and to be transmitted pulses, such as in a FIR filter.

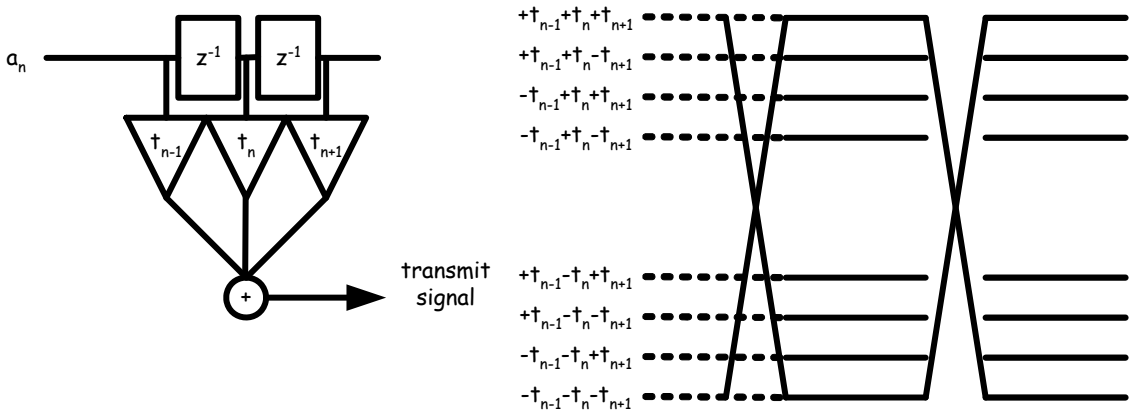


Figure 8-16. Transmitter FIR Filter Function

This superposition can be understood by referring to the amplitudes depicted for various bit sequences in Figure 8-16.

The transmit emphasis can be defined to have certain limits of maximum transmit amplitude or ratios of emphasis as defined below.

$$P_{post} = \frac{t_1}{t_0}$$

$$E = 20log\frac{1 + P_{post}}{1 - P_{post}}$$

$$\sum |t_n| < V_{tx}|_{min}$$

where

P_{post} is the first coefficient of the transmit FIR

E is the emphasis of the transmit emphasis

$V_{tx}|_{min}$ is the maximum transmit amplitude

8.6.4 Receiver Pulse Response

Given an emphasized transmitter the pulse response of the receiver should be recalculated using the emphasized transmit pulse as opposed to a simple NRZ pulse.

The receiver pulse cursors are defined in Figure 8-17.

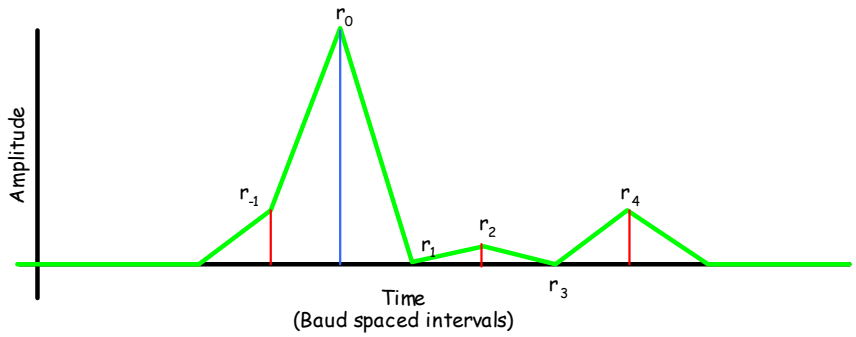


Figure 8-17. Receiver Pulse Definition

8.6.5 Crosstalk Pulse Response

The crosstalk pulse response is analogous to the receiver pulse response as defined in Section 8.6.4 but using the crosstalk channel, i.e. NEXT or FEXT network analysis measurement. The transmit signal as seen in the system should be used for the calculation of the resulting crosstalk pulse response, e.g. an emphasized transmitter from above, or XAUI transmit NRZ pulse.

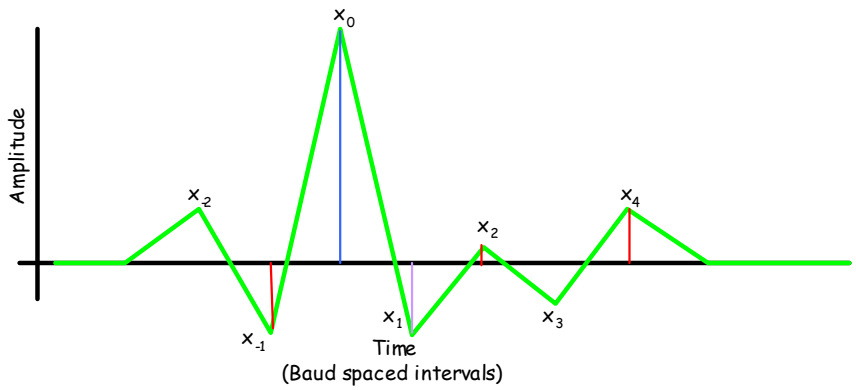


Figure 8-18. Crosstalk Pulse Definition

The Crosstalk pulse response is then defined as above in Figure 8-18 as being a set of cursors x_n usually oscillatory in form. The position of x_0 is defined as being at the maximum amplitude of the pulse response.

8.6.6 Decision Feedback Equalizer

The following filter function can be used to verify the capability of the channel to be used in such an application.

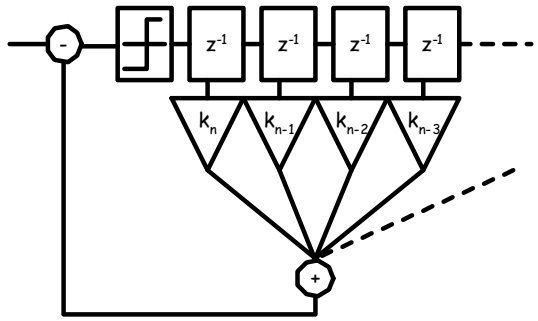


Figure 8-19. Decision Feedback Equalizer

The value of the coefficients are calculated directly from the channel pulse response or the receiver pulse using an emphasized transmitter.

$k_n = c_n\big|_{n = [1,m]}$ for unemphasized transmitters, or

$k_n = r_n\big|_{n = [1,m]}$ for emphasized transmitters

This equalizer is capable of equalizing a finite number of post cursors, whose individual values may be limited.

8.6.7 Time Continuous Transverse Filter

A.k.a. Feed forward Filter, Finite Input Response or Comb Structure, the Transverse Filter, Figure 8-20 consists of a finite number of coefficients, k. The sum of the continuous value of symbol spaced delayed samples multiplied by these coefficients then gives the resulting signal.

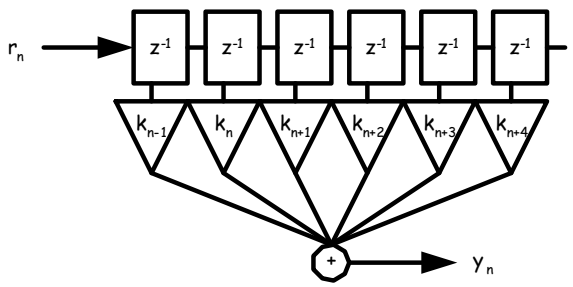


Figure 8-20. Feed Forward Filter

8.6.7.1 Time Continuous Zero-Pole Equalizer Adaption

The pole-zero algorithm takes the SDD21 magnitude response for the through channel and inverts it to produce a desired CTE filter response curve. From a set of initial conditions for p_n poles and z_n zeros, the squared differences are minimized between the CTE response and the inverse channel response curve. The minimization is done using a simplex method, specifically the Nelder-Mead

Multidimensional Unconstrained Non-Linear Minimization Method. The Nelder-Mead method provides a local minimization of the square of the difference between the two curves by descending along the gradient of the difference function. Once the optimization result is obtained, it is compared to a specified threshold. If the threshold exceeds the target tolerance, an incrementally offset seed point is generated from a 6-dimensional grid of seed points, and the process is iterated until the correct curve is obtained within the target tolerance.

8.6.8 Time Continuous Zero/Pole

The Zero/Pole Filter is defined, in the frequency domain by

$$H(f) = \frac{p}{z} \cdot \frac{(z + j2\pi f)}{(p + j2\pi f)}$$

and consists of a single zero, z, and single pole, p.

8.6.9 Degrees of Freedom

8.6.9.1 Receiver Sample Point

A receiver shall be allowed to either position the centre sampling point fully independently to the signal transitions or exactly in between the mean crossover of the receiver signal.

8.6.9.2 Transmit Emphasis

Transmit emphasis and receiver filter coefficients must be optimized with the defined resolution to give the best achievable results. Unless otherwise stated it shall be assumed that the coefficients are defined using floating point variables.

8.7 Jitter Modelling

This section describes the theoretical background of the methodology used for jitter budgeting and jitter measurement. To avoid fundamental issues with the additional of jitter using the dual dirac model through a band limited channel, a fundamental methodology call “stateye” is defined in Section 8.7.5, which uses only convolution of the jitter distribution for the calculation of the jitter at the receiver.

8.7.1 High Frequency Jitter vs. Wander

Jitter is defined as the deviation of the signal transition from an origin, usually its mean. This deviation has an amplitude and an associated spectrum. High frequency jitter is defined by a 1st order high pass phase filter with a corner frequency equal to the ideal CDR bandwidth. The low frequency Jitter or Wander is defined by a 1st order low pass phase filter with a corner frequency equal to the bandwidth.

8.7.2 Total Wander vs. Relative Wander

Generation of Total and Relative Wander can be achieved using a “Common” and “AntiPhase” Sinusoidal Source, where the total and relative wander are then related as defined below.

$$A_{total} = A_{common} + A_{antiphase}$$
$$A_{relative} = 2A_{antiphase}$$

By adding sinusoidal frequencies of slightly differing frequencies the maximum total and relative wander is achieved at various phase relationship like shown in Figure 8-21.

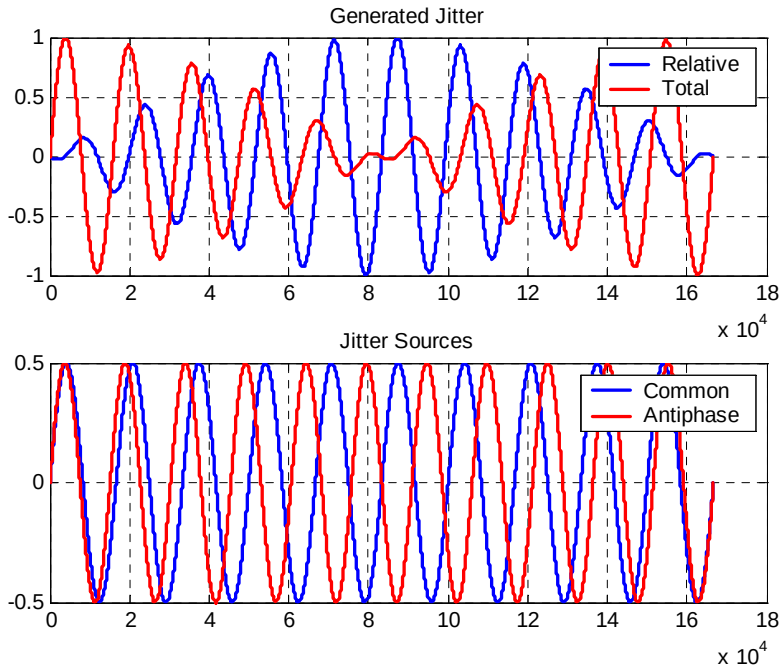


Figure 8-21. Generation of Total and Relative Wander

8.7.3 Correlated vs. Uncorrelated Jitter

If a correlation exists between the amplitude of the jitter and the current, past, and future signal level of a data channel, this type of jitter is deemed correlated. Typically this is encountered when band limitation and inter-symbol interference occurs. Due to amplitude to phase conversion of the ISI, a jitter is observed which has a direct correlation to the data pattern being transmitter.

8.7.4 Jitter Distributions

High frequency jitter is traditionally measured and described using probability density functions which describe the probability of the data signal crossing a decision threshold, as shown in Figure 8-22.

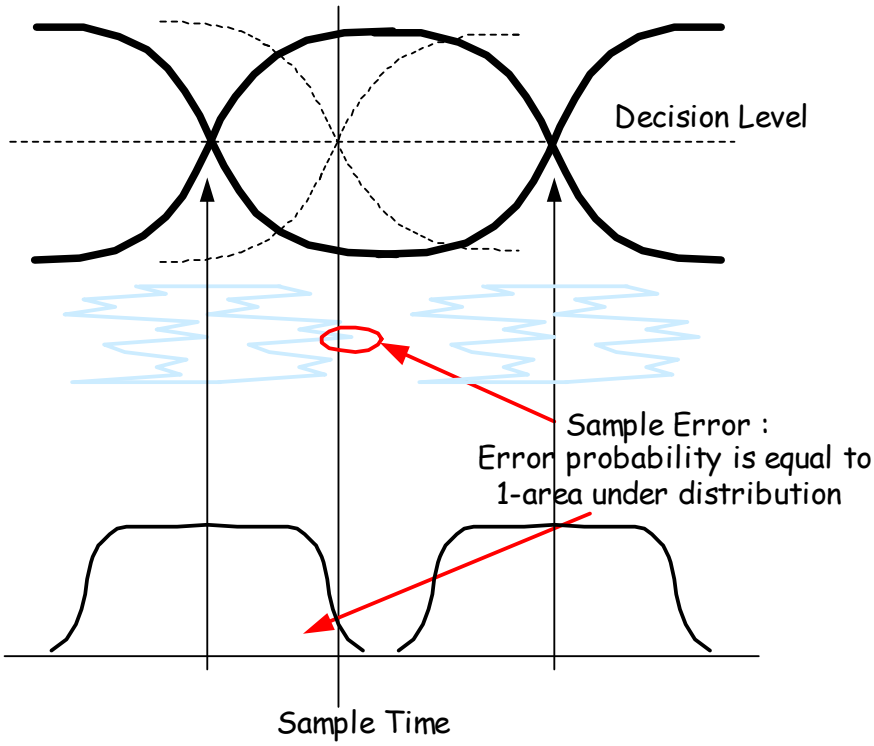


Figure 8-22. Jitter Probability Density Functions

The low probability part of the jitter distribution can be described by two components, mathematically described in the following sections.

8.7.4.1 Unbounded and Bounded Gaussian Distribution

We define a Unbounded Gaussian distribution function in terms of sigma as below.

$$GJ(\tau, \sigma) = \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sigma} \cdot e^{-\frac{\tau^2}{2\sigma^2}}$$

For every offset τ, there exists a finite and non-zero probability.

8.7.4.2 Bounded Gaussian Distribution

We define a Bounded Gaussian Distribution function[1] in terms of sigma and a maximum value as below.

$$GJ(\tau, \sigma) = \begin{cases} \frac{1}{\sqrt{2\pi}} \cdot \frac{1}{\sigma} \cdot e^{\frac{\tau^2}{2\sigma^2}} & if \ \tau \le \tau_{max} \\ 0 & \tau > \tau_{max} \end{cases}$$

For random processes consisting of a finite number of random variables there exists a finite non-zero probability only if $\tau \le \tau_{max}$. For example, a band limited channel is bounded but shows a Gaussian Distribution below its maximum. See Section 8.7.4.8, "Example of Bounded Gaussian" for an explanation concerning extrapolation.

8.7.4.3 High Probability Jitter

We define a dual dirac distribution function for a High Probability jitter (W) as below.

$$HPJ(\tau, W) = \frac{\delta(\tau - \frac{W}{2})}{2} + \frac{\delta(\tau + \frac{W}{2})}{2}$$

8.7.4.4 Total Jitter

We define the convolution of the High Probability and Gaussian jitter as being the total jitter and define it as below.

$$TJ(\tau, W, \sigma) = \frac{1}{2\sqrt{2\pi}} \cdot \frac{1}{\sigma} \cdot \left[e^{-\frac{\delta\left(\tau - \frac{W}{2}\right)^2}{2\sigma^2}} + e^{-\frac{\delta\left(\tau + \frac{W}{2}\right)^2}{2\sigma^2}} \right]$$

8.7.4.5 Probability Distribution Function vs. Cumulative Distribution Function

An example of the convolution of GJ (magenta), HPJ (green) to give TJ (red) can be seen Figure 8-23.

[1]Due to the bounded function the function does not comply to the requirements that the integral of the pdf from minus infinity to infinity is one. This small inaccuracy is recognized and acceptance in this context.

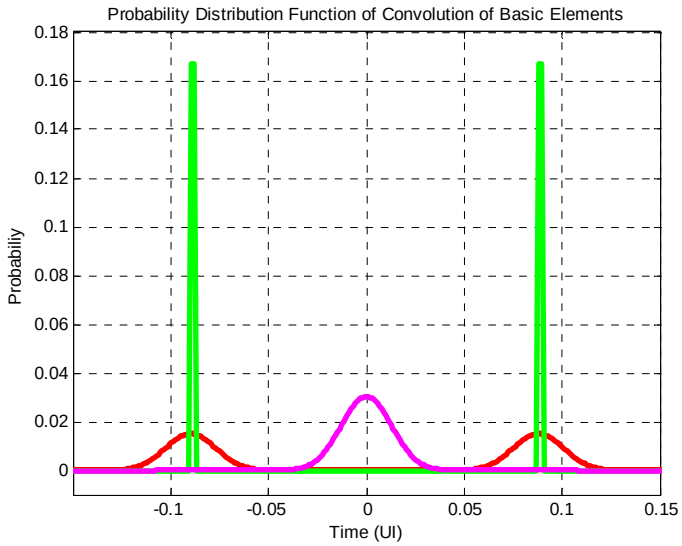


Figure 8-23. Example of Total Jitter PDF

When integrating the probability distribution functions, same colors, we obtain the cumulative distribution function or half the bathtub, shown in Figure 8-24.

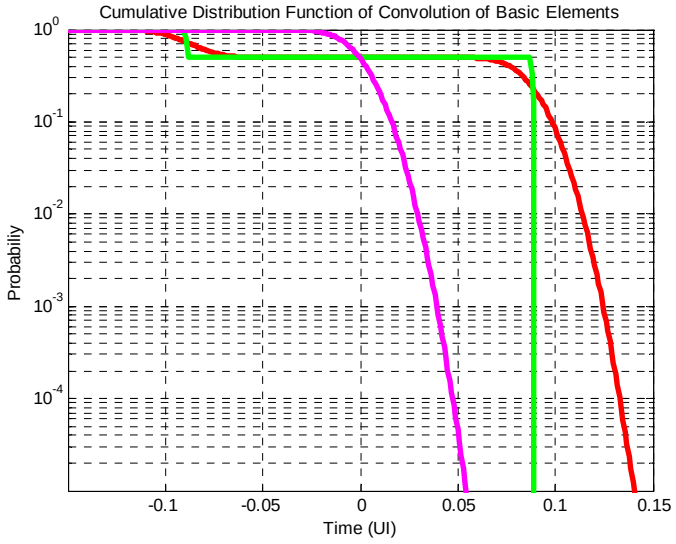


Figure 8-24. Example of Total Jitter CDF

8.7.4.6 BathTub Curves

Given a measured bathtub curve consisting of measured BER for various sampling offsets, the defined Gaussian and High Probability Distributions can be used to describe the important features of the distribution.

Initially the BER axis should be converted to Q as defined below, e.g. a BER of 10^{-12} is a Q=7.04, and a BER of 10^{-15} a Q=7.94[1].

$$Q = \sqrt{2} \cdot erf^{-1}(2 \cdot (1 - BER) - 1)$$

where $\mathrm{erf}^{-1}(x)$ is the inverse function of the error function $\mathrm{erf}(x)$.

$$erf(z) = \frac{2}{\sqrt{\pi}} \cdot \int_0^z e^{-t^2} dt$$

Note: this conversion from BER to Q is only valid given a large time offset from the optimal sampling point. The use of the nomenclature BER in this reference should therefore be carefully used. Any accurate prediction of the BER towards the center of the eye should be done using Marcum's Q function, and is outside the scope of this document.

By linearizing the bathtub, as shown in Figure 8-25, we can describe the function of the left and right hand linear parts of the bathtub in terms of an offset (HPJ) and gradient (1/GJ).

$$Q_{left}(\tau_{offset}) = (\tau_{offset} - HPJ_{left}) \cdot \frac{1}{GJ_{left}}$$

$$Q_{right}(\tau_{offset}) = (HPJ_{left} - \tau_{offset}) \cdot \frac{1}{GJ_{right}}$$

The conversion to a linearized bathtub from a measurement should be calculated using a polynomial fit algorithm for parts of the measurement made at low BERs or high Q.

[1]It is assumed that when measuring the jitter bathtub that the left and right parts of the bathtub are independent to each other, e.g. the tail of the right hand part of the bathtub and negligible effect on the left hand side of the bathtub.

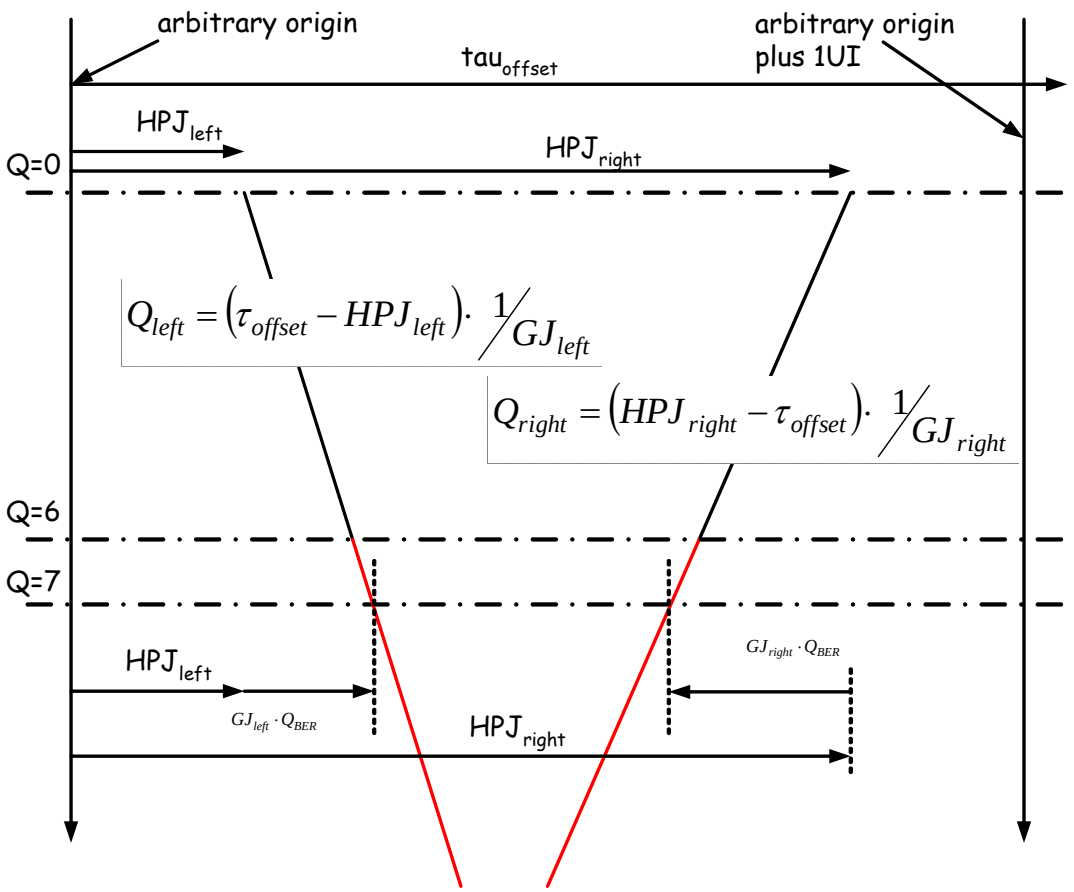


Figure 8-25. Bathtub Definition

8.7.4.7 Specification of GJ and HPJ

In this specification the left and right hand terms are combined to give a single definition as below where Q_{BER} is the Q for the BER of interest, e.g Q=7.49 for a $BER = 10^{-15}$.

$$HPJ_{total} = 1 - (HPJ_{right} - HPJ_{left})$$

$$GJ_{total} = GJ_{left} \cdot Q_{BER} + GJ_{right} \cdot Q_{BER} = 2Q_{BER} \cdot GJ_{rms}$$

$$GJ_{rms} = \frac{GJ_{left} + GJ_{right}}{2}$$

$$J_{total} = GJ_{total} + HPJ_{total}$$

8.7.4.8 Example of Bounded Gaussian

Assuming that the Cumulative Distribution Function of the jitter could be measured to the probabilities shown, Figure 8-26 shows an example of when a jitter should be classified as Correlated High Probability or Correlated Bounded Gaussian.

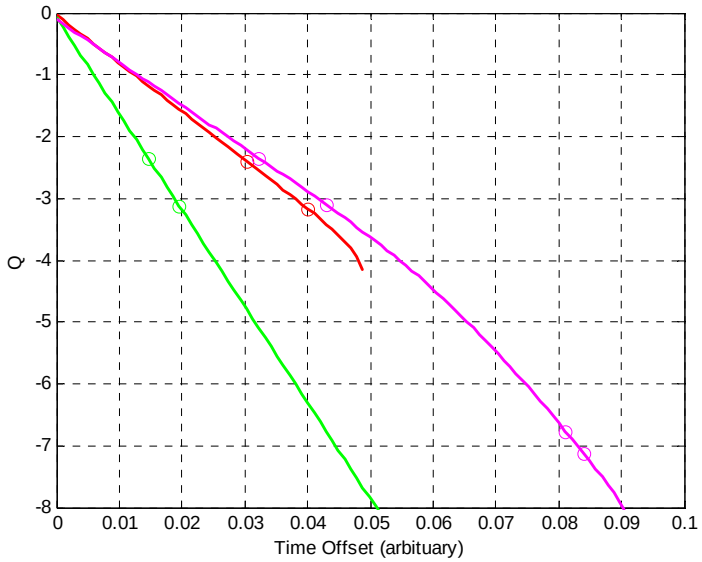


Figure 8-26. Example of Bounded Gaussian

The convolution of a true Unbounded Gaussian Jitter (green) with a Bounded Gaussian Jitter (Red) can be seen (Magenta). It can be clearly seen and measured that at a Q of -3 the Bounded Jitter is still Gaussian and the resulting convolution can be calculated using RMS addition. Below a Q of -5 the Bounding effect can be seen, and if we linearize the Bathtub we measure a non-zero High Probability Jitter and Gaussian component.

8.7.5 Statistical Eye Methodology

The following section describes the fundamental underlying the StatEye methodology. For a golden implementation please refer to the scripts on the OIF website, which are published separately.

8.7.5.1 Derivation of Cursors and Calculation of PDF

The Statistical Eye Methodology uses a channel pulse response and crosstalk pulse response in conjunction with a defined sampling jitter to generate an equivalent eye which represents the eye opening as seen by the receiver for a given probability of occurrence. This is shown in Figure 8-27.

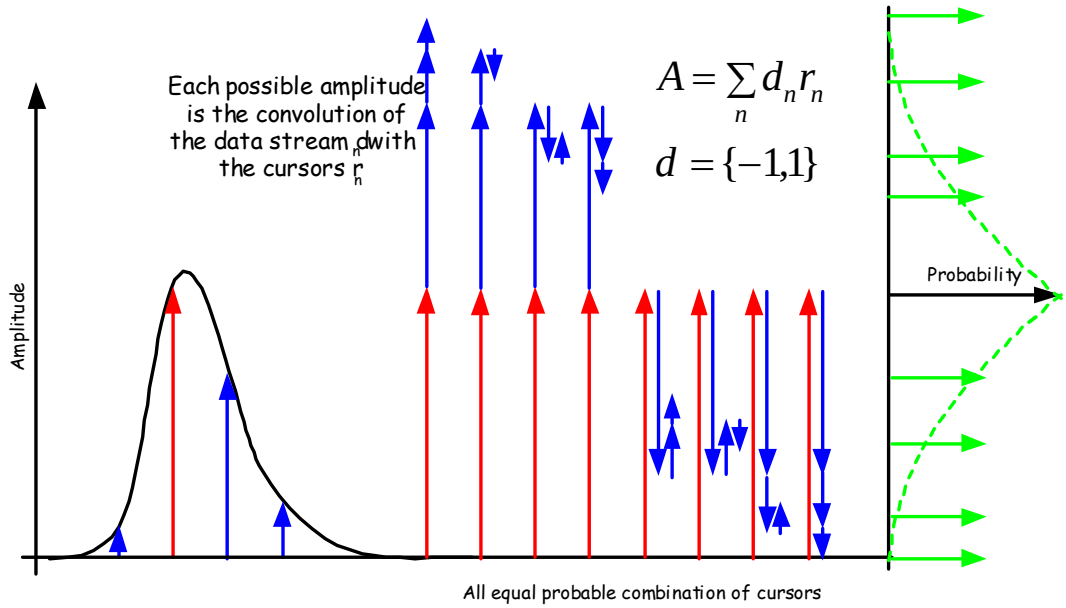


Figure 8-27. Statistics of Pulse Response Cursor

Given a pulse response (black left) we locate c_0 at an arbitrary point (red arrow) and measure the symbol space cursors (blue arrows),

Given a DFE the post cursors should be adjusted by negating the measured post cursors by the appropriate static coefficient of the DFE, up to the maximum number of cursors specified.

According to the exact data pattern these cursors superimpose to Inter-symbol Interference. Each possible combination of these cursors is calculated and from these combinations a histogram is generated to form the probability density function (PDF) (green).

By varying the reference sampling point for c0 as shown in Figure 8-28, the previous function is repeated and family of conditional PDFs build up.

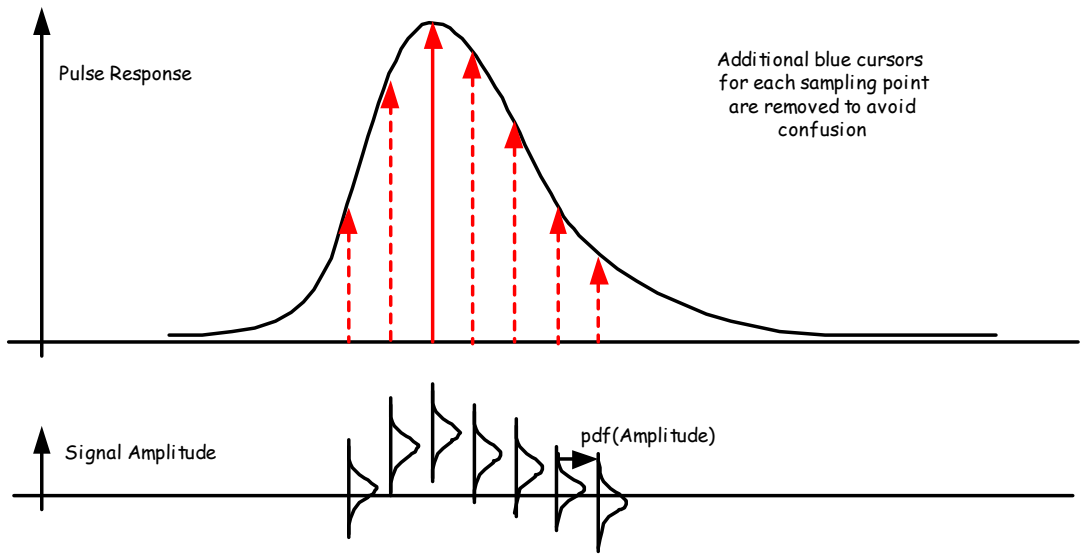


Figure 8-28. Variation of the c0 Sampling Time

This can be represented mathematically below.

Given,

$r_n(\tau)$ are the cursors of the pulse response at sampling τ

e_b is the ideal static equalization coefficients of the b tap DFE

$c(\tau)$ is the set of equalization cursors at sampling τ

$\delta(\tau) = \lim_{\varepsilon \to 0} \varepsilon |x|^{\varepsilon - 1}$ is the dirac or delta function

$d_{n,b}$ are all the possible combinations of the data stream and is either 1 or 0

$p(ISI, \tau)$ is the probability density function of the ISI for a given sample time

A similar family of PDFs are generated for the crosstalk pulse response and any other aggressors in the system using the cursor set below, noting that the entire pulse response is used.

$$c(\tau) = \left[r_{-\frac{m}{2}}(\tau) \ldots r_{-1}(\tau)\ r_0(\tau)\ r_1(\tau) \ldots r_{\frac{m}{2}}(\tau) \right]$$

8.7.5.2 Inclusion of Sampling Jitter

In a real system the sampling point c0 is defined by the CDR and is jittered, for the sake of standardization, by the transmitter. This jitter has a probability density function which is centred at the receiver CDR sampling point and defined the probability of each of the previous conditional PDFs occurring[1], as shown in Figure 8-29.

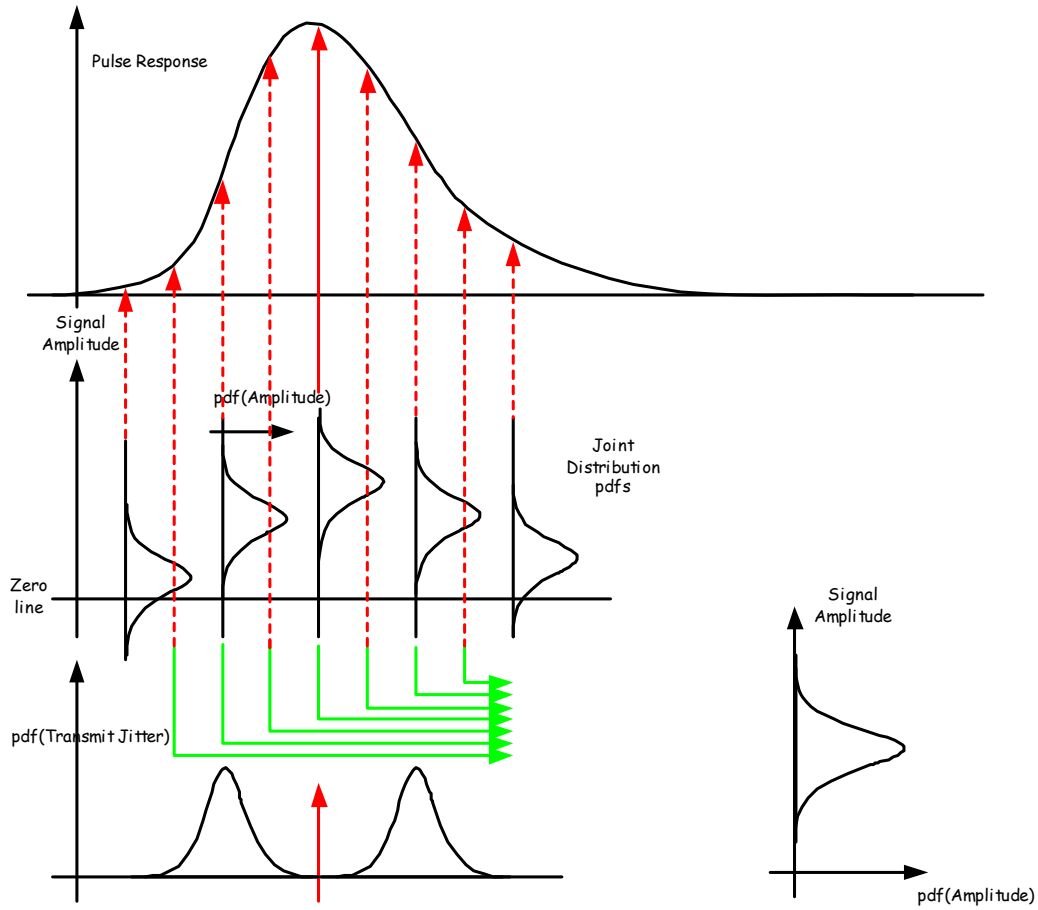


Figure 8-29. Varying the Receiver Sampling Point

By multiplying each the conditional PDFs by its associated sampling jitter probability and summing their results together, the joint probability density function at the given receiver CDR sample point can be calculated.

Given,

$p_{jitter}(\tau, w, \sigma)$ is the dual dirac probability density function of the sampling jitter in the system, as defined in Section 8.7.4.4, "Total Jitter"

[1]Currently DCD effects are not taken into account

$p_{crosstalk}(ISI, \tau)$ is the probability density function of the crosstalk

$p_{forward}(ISI, \tau)$ is the probability density function of the ISI of the forward channel

$a \otimes b$ is the convolution operative

$$p_{average}(ISI, \tau) =$$

$$\int_{-\infty}^{\infty} \{[p_{crosstalk}(ISI, \tau + \upsilon + w) \otimes p_{forward}(ISI, \tau + \upsilon)] \cdot p_{jitter}(\upsilon, w, \sigma)\} d\upsilon$$

8.7.5.3 Generation of Statistical Eye

By varying the receiver CDR sampling point a new joint probability density function, Figure 8-29 can be generated.

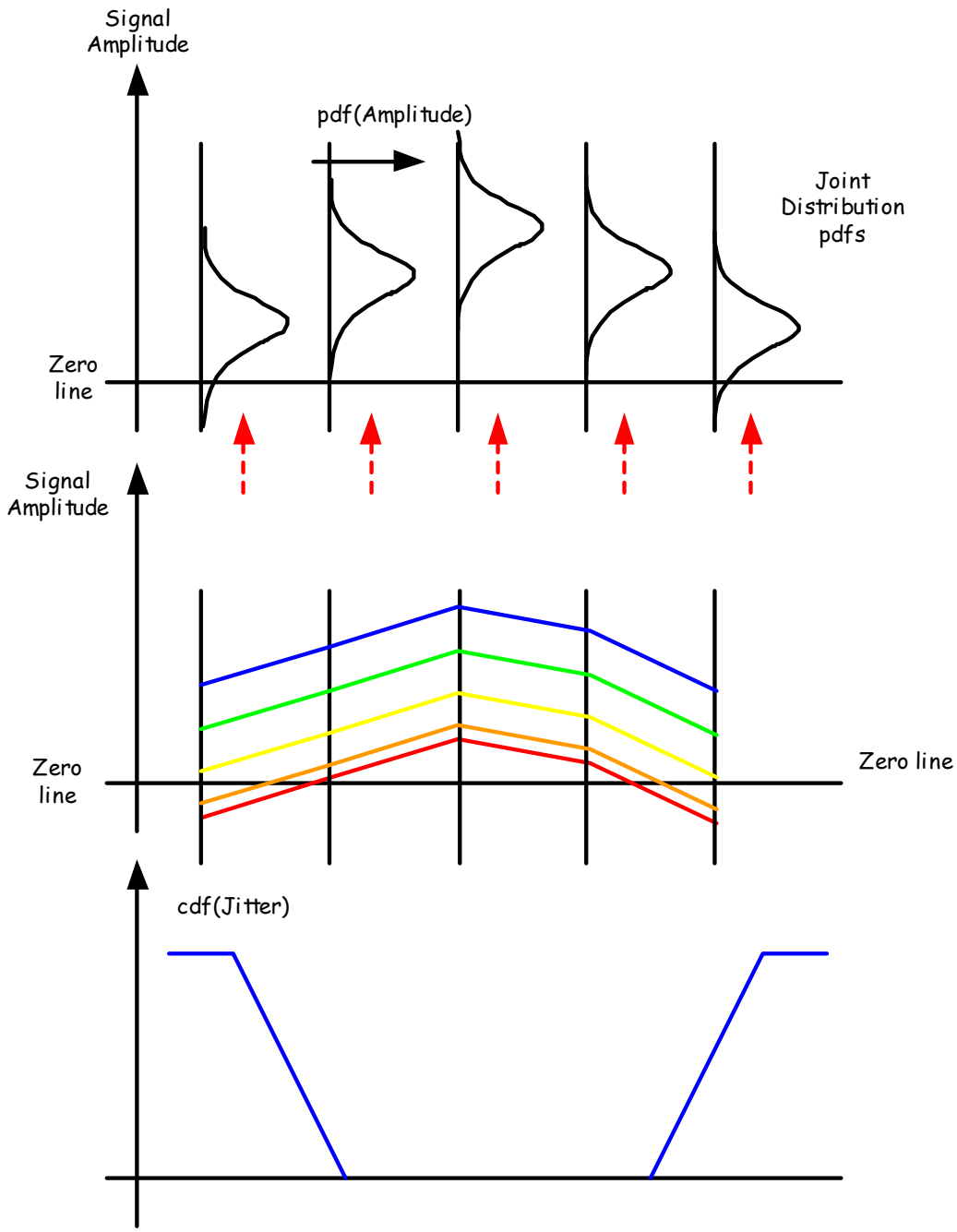


Figure 8-30. Generation of the Data Eye and Bathtub

By integrating the Joint Probability Density Function to give the Cumulative Distribution function, and creating a contour plot an equivalent of the receiver eye can be generated which shows the exact probability of obtaining a given amplitude, shown in Figure 8-30, this equivalent eye is termed the statistical eye, shown in Figure 8-31.

By only plotting the probability against time by cutting the statistical eye along the decision threshold axis can a bathtub of the jitter can be generated.

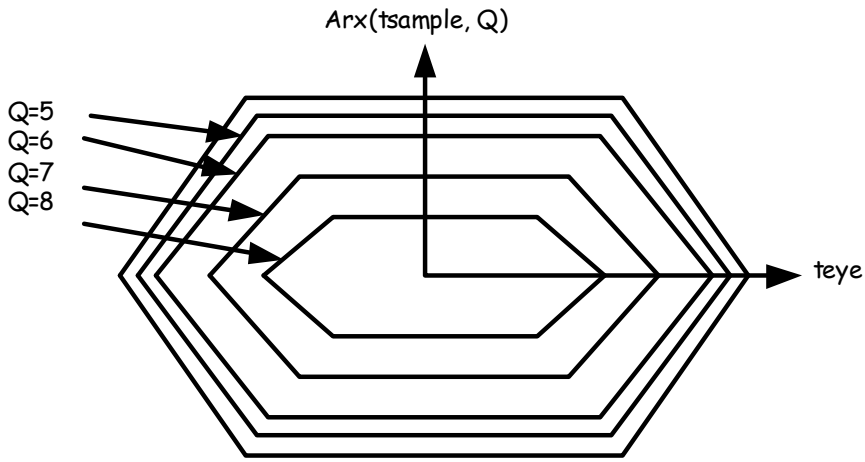


Figure 8-31. Statistical Eye

Chapter 9 1.25Gbaud, 2.5Gbaud, and 3.125Gbaud LP-Serial Links

This chapter details the requirements for Level I RapidIO LP-Serial short and long run electrical interfaces of nominal baud rates of 1.25Gbaud, 2.5Gbaud, and 3.125Gbaud using NRZ coding (hence 1 bit per symbol at the electrical level). A compliant device must meet all of the requirements listed below. The electrical interface is based on a high speed, low voltage logic with a nominal differential impedance of 100Ω. Connections are point-to-point balanced differential pair and signalling is unidirectional.

The level of links defined in this section are identical to those defined in revision 1.3 of the 1x/4x LP-Serial electrical specification. The terminology has been updated to be consistent with the new level links defined in Section 8.1, "Introduction".

9.1 Level I Application Goals

The following are application requirements common to short run and long run at 1.25Gbaud, 2.5Gbaud, and 3.125Gbaud:

- The electrical specifications shall support lane widths options of 1x to Nx where N=2, 4, 8, and 16.
- AC coupling at the receiver shall be specified to ensure inter-operability between transmitters and receivers of different vendors and technologies.
- A compliant device may implement any subset of baud rates contained in this section.
- A compliant device may implement either a short run transmitter, a long run transmitter, or both, at each of the baud rates that it supports.
- The clock frequency tolerance requirement for transmit and receive are ±100ppm. The worst case frequency differences between any transmit and receive clock is 200ppm.
- The Bit Error Ratio (BER) shall be better than 10^{-12} per lane.
- The transmitter pins shall be capable of surviving short circuit either to each other, to supply voltages, and to ground.
- The short run interface shall be capable of spanning at least 20cm of PCB material with up to a single connector.

- The long run interface shall be capable of spanning at least 50cm of PCB material with up to two connectors.

9.2 Equalization

At the high baud rates used by Level I LP-Serial links, the signals transmitted over a link are degraded by losses and characteristic impedance discontinuities in the interconnect media. The losses increase with increasing baud rate and interconnect media length and cause signal attenuation and inter-symbol interference that degrade the opening of the eye pattern at both the receiver input and the data decoder decision point. Depending on the baud rate and interconnect length, the degradation can be greater than that allowed by the specification.

The signal degradation can be partially negated by the use of equalization in the transmitter and/or receiver. Equalization in the transmitter can improve the eye pattern at both the receiver input and the data decoder decision point. Equalization in the receiver can only improve the eye pattern at the data decoder decision point. Equalization is likely to be required only for longer Level I interconnects and higher Level I baud rates.

The types of equalizers and, if the equalizers are adaptive, the adaptive equalizer training algorithms that may be used in Level I transmitter or receiver are subject to the following restrictions.

Equalizers that can convert a single bit error into a multiple bit burst error, such as decision feedback equalizers (DFEs), shall not be used when IDLE1 has been selected for use on the link.

The training algorithm for any adaptive equalization used by a Level I transmitter and/or receiver shall consistently train the equalizer and retain the equalizer's training when IDLE1 is the training signal and shall consistently retain the equalizer's training when IDLE1 has been selected for use on the link and the signal on the link is a continuous sequence of maximum length packets whose payload is either all ONES or all ZEROS.

9.3 Explanatory Note on Level I Transmitter and Receiver Specifications

AC electrical specifications are given for the transmitter and receiver. Long run and short run interfaces at three baud rates are described.

The parameters for the AC electrical specifications are guided by the XAUI electrical interface specified in Clause 47 of IEEE 802.3ae-2002.[1]

XAUI has similar application goals as serial RapidIO Level I devices as described in Section 8.5, "Common Electrical Specification". The goal of this standard is that

electrical designs for Level I electrical designs can reuse XAUI, suitably modified for applications at the baud intervals and runs described herein.

9.4 Level I Electrical Specification

9.4.1 Level I Short Run Transmitter Characteristics

The key transmitter electrical specifications at compliance point T are summarized in Table 9-1 and Table 9-2 while the following sections fully detail all of the requirements.

Table 9-1. Level I SR Transmitter AC Timing Specifications

Characteristics	Symbol	Conditions	Min	Typ	Max	Units
Baud Rate	T_Baud	Section 9.4.1.2	1.25		3.125	Gbaud
Absolute Output Voltage	V_O	Section 9.4.1.3	-0.40		2.30	Volts
Output Differential Voltage (into floating load Rload = 100Ω	T_Vdiff	Section 9.4.1.3	500		1000	mVppd
Differential Resistance	T_Rd	Section 9.4.1.5	80	100	120	W
Recommended output rise and fall times (20% to 80%)	T_tr, T_tf	Section 9.4.1.4	60			ps
Differential Output Return Loss (T_baud/10 ≤ f < T_Baud/2)	T_SDD22	Section 9.4.1.6				dB
Differential Output Return Loss (T_baud/2 ≤ f ≤ T_baud)						dB
Common Mode Return Loss (625 MHz ≤ f ≤ T_baud)	T_SCC22	Section 9.4.1.6			Note 3	dB
Transmitter Common Mode Noise[1]	T_Ncm				Note 4	mVppd
Output Common Mode Voltage	T_Vcm	Load Type 0[2]	0		2.1	V
Multiple output skew, N<=4	S_{MO}	Section 9.4.1.7			1000	ps
Multiple output skew, N>4	S_{MO}	Section 9.4.1.7			2UI +1000	ps
Unit Interval	UI		320		800	ps

NOTES:
1. For all Load Types: R_Rdin = 100Ω ± 20Ω. For Vcm definition, see Figure 8-1
2. Load Type 0 with min. T_Vdiff, AC-Coupling or floating load.
3. It is suggested that T_SCCC22 be -6 dB to be compatible with Level II transmitter requirements
4. It is suggested that T_Ncm be limited to 5% of T_Vdiff to be compatible with Level II transmitter requirements

Table 9-2. Level I SR Transmitter Output Jitter Specifications

Characteristic	Symbol	Condition	Min	Typ	Max	Units
Uncorrelated High Probability Jitter	T_UHPJ	Section 9.4.1.9			0.17	UIpp
Duty Cycle Distortion	T_DCD	Section 9.4.1.9			0.05	UIpp
Total Jitter	T_TJ	Section 9.4.1.9			0.35	UIpp
Eye Mask	T_X1	Section 9.4.1.9			0.17	UI
Eye Mask	T_X2	Section 9.4.1.9			0.39	UI
Eye Mask	T_Y1	Section 9.4.1.9	250			mV
Eye Mask	T_Y2	Section 9.4.1.9			500	mV

9.4.1.1 Level I SR Transmitter Test Load

All transmitter characteristics should be implemented and measured to a differential impedance of 100Ω ± 5% at DC with a return loss of better than 20dB from the baud rate divided by 1667 to 1.5 times the baud rate, unless otherwise noted.

9.4.1.2 Level I SR Transmitter Baud Rate

The baud rates are 1.25Gbaud, 2.5Gbaud, and 3.125Gbaud with a tolerance of ±100ppm.

9.4.1.3 Level I SR Transmitter Amplitude and Swing

Transmitter differential amplitude shall be between 500 to 1000mVppd, inclusive, either with or without transmit emphasis. Absolute driver output voltage shall be between -0.4V and 2.4V, inclusive, with respect to the local ground. See Figure 8-1 for an illustration of absolute driver output voltage and definition of differential peak-to-peak amplitude.

9.4.1.4 Level I SR Transmitter Rise and Fall Times

The recommended minimum differential rise and fall time is 60ps as measured between the 20% and 80% of the maximum measured levels; the maximum differential rise and fall times are defined by the Tx eye diagram (Figure 8-2 and Table 9-4). Shorter rise and falls may result in excessive high frequency components and increase EMI and cross talk.

9.4.1.5 Level I SR Transmitter Differential Pair Skew

It is recommended that the timing skew at the output of a LP-Serial transmitter between the two signals that comprise a differential pair not exceed 25ps at 1.25Gbaud, 20ps at 2.5Gbaud, and 15ps at 3.125Gbaud.

9.4.1.6 Level I SR Transmitter Output Resistance and Return Loss

Refer to Section 8.5.11, "Differential Resistance and Return Loss, Transmitter and Receiver" for the reference model for return loss. See Table 9-3 for Level I short and long run transmitter parameters. Definitions for these parameters are in Figure 8-12.

Table 9-3. Level I SR Transmitter Return Loss Parameters

Parameter	Value	Units
A0	-10	dB
f0	T_Baud/10	Hz
f1	625	MHz
f2	T_Baud	Hz
Slope	10.0	dB/dec

9.4.1.7 Level I SR Transmitter Lane-to-Lane Skew

The electrical level of lane-to-lane skew caused by the transmitter circuitry and associated routing must be less than 1000 ps for links of 4 lanes or less. Links with greater than 4 lanes must have lane-to-lane skew of less than 2UI + 1000 ps. The transmitter lane-to-lane skew is only for the serdes Tx and does not include any effects of the channel.

9.4.1.8 Level I SR Transmitter Short Circuit Current

It is recommended that the max DC current into or out of the transmitter pins when either shorted to each other or to ground be ±100mA when the device is fully powered up. From a hot swap point of view, the ±100mA limit is only valid after 10 μs.

9.4.1.9 Level I SR Transmitter Template and Jitter

For each baud rate at which a transmitter is specified to operate, the output eye pattern of the transmitter shall fall entirely within the unshaded portion of the Transmitter Output Compliance Mask shown in Figure 8-2 with the parameters specified in Table 9-4. The output eye pattern of a LP-Serial transmitter that implements pre-emphasis (to equalize the link and reduce inter-symbol interference) need only comply with the Transmitter Output Compliance Mask when pre-emphasis is disabled or minimized.

Table 9-4. Level I SR Near-End (Tx) Template Intervals

Characteristics	Symbol	Near-End Value	Units
Eye Mask	T_X1	0.17	UI
Eye Mask	T_X2	0.39	UI
Eye Mask	T_Y1	250	mV
Eye Mask	T_Y2	500	mV

Table 9-4. Level I SR Near-End (Tx) Template Intervals

Characteristics	Symbol	Near-End Value	Units
Eye Mask	T_Y3	N/A	mV
Uncorrelated Bounded High Probability Jitter	T_UBHPJ	0.17	UIpp
Duty Cycle Distortion	T_DCD	0.05	UIpp
Total Jitter	T_TJ	0.35	UIpp

9.4.2 Level I Long Run Transmitter Characteristics

The key transmitter electrical specifications at compliance point T are summarized in Table 9-5 and Table 9-6 while the following sub-clauses fully detail all of the requirements.

Table 9-5. Level I LR Transmitter AC Timing Specifications

Characteristics	Symbol	Conditions	Min	Typ	Max	Units
Baud Rate	T_Baud	Section 9.4.2.2	1.25		3.125	Gbaud
Absolute Output Voltage	V_O	Section 9.4.2.3	-0.40		2.30	Volts
Output Differential Voltage (into floating load Rload=100Ω)	T_Vdiff	Section 9.4.2.3	800		1600	mVppd
Differential Resistance	T_Rd	Section 9.4.1.5	80	100	120	W
Recommended output rise and fall times (20% to 80%)	T_tr, T_tf		60			
Differential Output Return Loss (T_baud/10 £ f < T_Baud/2)	T_SDD22	Section 9.4.1.6				dB
Differential Output Return Loss (T_baud/2 ≤ f ≤ T_baud)						dB
Common Mode Return Loss (625 MHz ≤ f ≤ T_baud)	T_SCC22	Section 9.4.1.6			Note 3	dB
Transmitter Common Mode Noise[1]	T_Ncm				Note 4	mVppd
Output Common Mode Voltage	T_Vcm	Load Type 0[2]	0		2.1	V
Multiple output skew, N<=4	S_{MO}				1000	ps
Multiple output skew, N>4	S_{MO}				2UI+ 1000	ps
Unit Interval	UI		320		800	ps

NOTES:
1. For all Load Types: R_Rdin = 100Ω ± 20Ω. For Vcm definition, see Figure 8-1.
2. Load Type 0 with min. T_Vdiff, AC-Coupling or floating load.
3. It is suggested that T_SCCC22 be -6 dB to be compatible with Level II transmitter requirements
4. It is suggested that T_Ncm be limited to 5% of T_Vdiff to be compatible with Level II transmitter requirements

Table 9-6. Level I LR Transmitter Output Jitter Specifications

Characteristic	Symbol	Condition	Min	Typ	Max	Units
Uncorrelated High Probability Jitter	T_UHPJ	Section 9.4.1.9			0.17	UIpp
Duty Cycle Distortion	T_DCD	Section 9.4.1.9			0.05	UIpp
Total Jitter	T_TJ	Section 9.4.1.9			0.35	UIpp
Eye Mask	T_X1	Section 9.4.1.9			0.17	UI
Eye Mask	T_X2	Section 9.4.1.9			0.39	UI
Eye Mask	T_Y1	Section 9.4.1.9	400			mV
Eye Mask	T_Y2	Section 9.4.1.9			800	mV

9.4.2.1 Level I LR Transmitter Test Load

All transmitter characteristics should be implemented and measured to a differential impedance of 100Ω ± 5% at DC with a return loss of better than 20 dB from the baud rate divided by 1667 to 1.5 times the baud rate, unless otherwise noted.

9.4.2.2 Level I LR Transmitter Baud Rate

The baud rates are 1.25Gbaud, 2.5Gbaud, and 3.125Gbaud with a tolerance of ±100ppm.

9.4.2.3 Level I LR Transmitter Amplitude and Swing

Transmitter differential amplitude shall be between 400 to 1600mVppd, inclusive, either with or without transmit emphasis. Absolute driver output voltage shall be between -0.4V and 2.4V, inclusive, with respect to the local ground. See Figure 8-1 for an illustration of absolute driver output voltage and definition of differential peak-to-peak amplitude.

9.4.2.4 Level I LR Transmitter Rise and Fall Times

The recommended minimum differential rise and fall time is 60ps as measured between the 20% and 80% of the maximum measured levels; the maximum differential rise and fall times are defined by the Tx eye diagram (Figure 8-2 and Table 9-8). Shorter rise and falls may result in excessive high frequency components and increase EMI and cross talk.

9.4.2.5 Level I LR Transmitter Differential Pair Skew

It is recommended that the timing skew at the output of a LP-Serial transmitter between the two signals that comprise a differential pair not exceed 25ps at 1.25Gbaud, 20ps at 2.5Gbaud and 15ps at 3.125 Gbaud.

9.4.2.6 Level I LR Transmitter Output Resistance and Return Loss

Refer to Section 8.5.11 for the reference model for return loss. See Table 9-3 for Level I short and long run transmitter parameters. Definitions for these parameters are in Figure 8-12.

Table 9-7. Level I LR Transmitter Return Loss Parameters

Parameter	Value	Units
A0	-8	dB
f0	T_Baud/10	Hz
f1	T_Baud/2	MHz
f2	T_Baud	Hz
Slope	16.6	dB/dec

9.4.2.7 Level I LR Transmitter Lane-to-Lane Skew

The electrical level of lane-to-lane skew caused by the transmitter circuitry and associated routing must be less than 1000 ps for links of 4 lanes or less. Links with greater than 4 lanes must have lane-to-lane skew of less than 2UI + 1000 ps. The transmitter lane-to-lane skew is only for the serdes Tx and does not include any effects of the channel.

9.4.2.8 Level I LR Transmitter Short Circuit Current

It is recommended that the max DC current into or out of the transmitter pins when either shorted to each other or to ground be ±100mA when the device is fully powered up. From a hot swap point of view, the ±100mA limit is only valid after 10μs.

9.4.2.9 Level I LR Transmitter Template and Jitter

For each baud rate at which a LP-Serial transmitter is specified to operate, the output eye pattern of the transmitter shall fall entirely within the unshaded portion of the Transmitter Output Compliance Mask shown in Figure 8-2 with the parameters specified in Table 9-4. The output eye pattern of a LP-Serial transmitter that implements pre-emphasis (to equalize the link and reduce inter-symbol interference) need only comply with the Transmitter Output Compliance Mask when pre-emphasis is disabled or minimized.

Table 9-8. Level I LR Near-End (Tx) Template Intervals

Characteristics	Symbol	Near-End Value	Units
Eye Mask	T_X1	0.17	UI
Eye Mask	T_X2	0.39	UI
Eye Mask	T_Y1	400	mV
Eye Mask	T_Y2	800	mV

Table 9-8. Level I LR Near-End (Tx) Template Intervals

Characteristics	Symbol	Near-End Value	Units
Eye Mask	T_Y3	N/A	mV
Uncorrelated Bounded High Probability Jitter	T_UBHPJ	0.17	UIpp
Duty Cycle Distortion	T_DCD	0.05	UIpp
Total Jitter	T_TJ	0.35	UIpp

9.4.3 Level I Receiver Specifications

Level I LP-Serial receiver electrical and timing specifications are stated in the text and tables of this section.

Table 9-9. Level I Receiver Electrical Input Specifications

Characteristic	Symbol	Conditions	Min	Typ	Max	Units
Rx Baud Rate (1.25 Gbaud)	R_Baud			1.250		Gbaud
Rx Baud Rate (2.5 Gbaud)				2.500		Gbaud
Rx Baud Rate (3.125 Gbaud)				3.125		Gbaud
Absolute Input Voltage	R_Vin	Section 9.4.3.4				
Input Differential voltage	R_Vdiff	Section 9.4.3.3	200		1600	mVppd
Differential Resistance	R_Rdin	Section 9.4.3.7	80	100	120	W
Differential Input Return Loss (100 MHz ≤ f ≤ R_Baud/2)	R_SDD11	Section 9.4.3.7				dB
Differential Input Return Loss (R_Baud/2 ≤ f ≤ R_Baud)						
Common mode Input Return Loss (100 MHz to 0.8 *R_Baud)	R_SCC11	Section 9.4.3.7				dB
Termination Voltage[1,2]	R_Vtt	R_Vtt floating[4]	Not Specified			V
Input Common Mode Voltage[1,2]	R_Vrcm	R_Vtt floating[3,4,]	-0.05		1.85	V
Wander divider (in Figure 8-8 & Figure 8-8)	n			10		

NOTES:
1. Input common mode voltage for AC-coupled or floating load input with min. T_Vdiff,
2. Receiver is required to implement at least one of specified nominal R_Vtt values, and typically implements only one of these values. Receiver is only required to meet R_Vrcm parameter values that correspond to R_Vtt values supported.
3. Input common mode voltage for AC-coupled or floating load input with min. T_Vdiff.
4. For floating load, input resistance must be ≥ 1kΩ.

Table 9-10. Level I Receiver Input Jitter Tolerance Specifications

Characteristic	Symbol	Conditions	Min	Typ	Max	Units
Bit Error Ratio	BER				10^{-12}	
Bounded High Probability Jitter	R_BHPJ	Section 9.4.3.8			0.37	UIpp
Sinusoidal Jitter, maximum	R_SJ-max	Section 9.4.3.8			8.5	UIpp
Sinusoidal Jitter, High Frequency	R_SJ-hf	Section 9.4.3.8			0.1	UIpp
Total Jitter (Does not include Sinusoidal Jitter)	R_TJ	Section 9.4.3.8			0.55	UIpp
Total Jitter Tolerance[1]	R_JT				0.65	UIpp
Eye Mask	R_X1	Section 9.4.3.8			0.275	UI
Eye Mask	R_Y1	Section 9.4.3.8			100	mV
Eye Mask	R_Y2	Section 9.4.3.8			800	mV

NOTES:
1. Total jitter is composed of three components, deterministic jitter, random jitter and single frequency sinusoidal jitter. The sinusoidal jitter may have any amplitude and frequency in the unshaded region of Figure 9-1. The sinusoidal jitter component is included to ensure margin for low frequency jitter, wander, noise, crosstalk and other variable system effects.

9.4.3.1 Level I Receiver Input Baud Rate

All devices shall work at either 1.25Gbaud, 2.5Gbaud, or 3.125Gbaud or any combination of these baud rates with the baud rate tolerance as per Section 9.4.1.2.

9.4.3.2 Level I Receiver Reference Input Signals

Reference input signals to the receiver have the characteristics determined by compliant transmitter. The reference input signal must satisfy the transmitter near-end template and jitter given in Figure 8-2 and Table 9-1 (Table 9-5), Table 9-2 (Table 9-6), and Table 9-3 (Table 9-7) for short run (long run) as well as the far-end eye template and jitter given in Figure 8-5 and Table 9-13, with the differential load impedance of 100Ω ± 1% at DC with a return loss of better than 20 dB from baud rate divided by 1667 to 1.5 times the baud rate. Note that the input signal might not meet either of these templates when the actual receiver replaces this load.

9.4.3.3 Level I Receiver Input Signal Amplitude

The receiver shall accept differential input signal amplitudes produced by compliant transmitters connected without attenuation to the receiver. This may be larger than the 1600mVppd maximum of the transmitter due to output/input impedances and reflections.

The minimum input amplitude is defined by the far-end transmitter template, the actual receiver input impedance and the loss of the actual PCB. Note that the far-end transmitter template is defined using a well controlled load impedance, however the real receiver is not, which can leave the receiver input signal smaller than the minimum 200mVppd.

9.4.3.4 Level I Receiver Absolute Input Voltage

The voltage levels at the input of an AC coupled receiver (if AC coupling is done within the receiver) or at the Tx side of the external AC coupling cap (if AC coupling is done externally) shall be between -0.40 to 2.30V, inclusive, with respect to local ground.

9.4.3.5 Level I Receiver Input Common Mode Impedance

AC coupling is to be considered part of the receiver for the purposes of this specification unless explicitly stated otherwise. It should be noted that various methods for AC coupling are allowed (for example, internal to the chip or done externally). See Section 8.5.13 for more information.

9.4.3.6 Level I Receiver Input Lane-to-Lane Skew

Refer to Section 8.5.9.

9.4.3.7 Level I Receiver Input Resistance and Return Loss

Refer to Section 8.5.11 for the reference model for return loss. See Table 9-11 for Level I receiver parameters. Definitions for these parameters are in Figure 8-12.

Table 9-11. Level I Input Return Loss Parameters

Parameter	Value	Units
A0	-8	dB
f0	100	MHz
f1	$RBaud \times \frac{1}{2}$	Hz
f2	R_Baud	Hz
Slope	16.6	dB/dec

Receiver input impedance shall result in a differential return loss better that -8dB and a common mode return loss better than -6dB from 100MHz to (0.5)*(R_Baud Frequency). This includes contributions from on-chip circuitry, the chip package and any off-chip components related to the receiver. AC coupling components are included in this requirement. The reference impedance for return loss measurements is 100Ω resistive for differential return loss and 25Ω resistive for common mode.

9.4.3.8 Level I Receiver Input Jitter Tolerance

The DUT shall be measured to have a BER better than specified for stressed signal with a confidence level of three sigma. Therefore the receiver shall tolerate at least the far-end eye template and jitter requirements as given in Figure 8-5 and Table 9-10 with an additional SJ with any frequency and amplitude defined by the mask of Figure 9-1 where the minimum and maximum total wander amplitude are 0.1UIpp and 8.5UIpp respectively. This additional SJ component is intended to

ensure margin for wander, hence is over and above any high frequency jitter from Table 9-13.

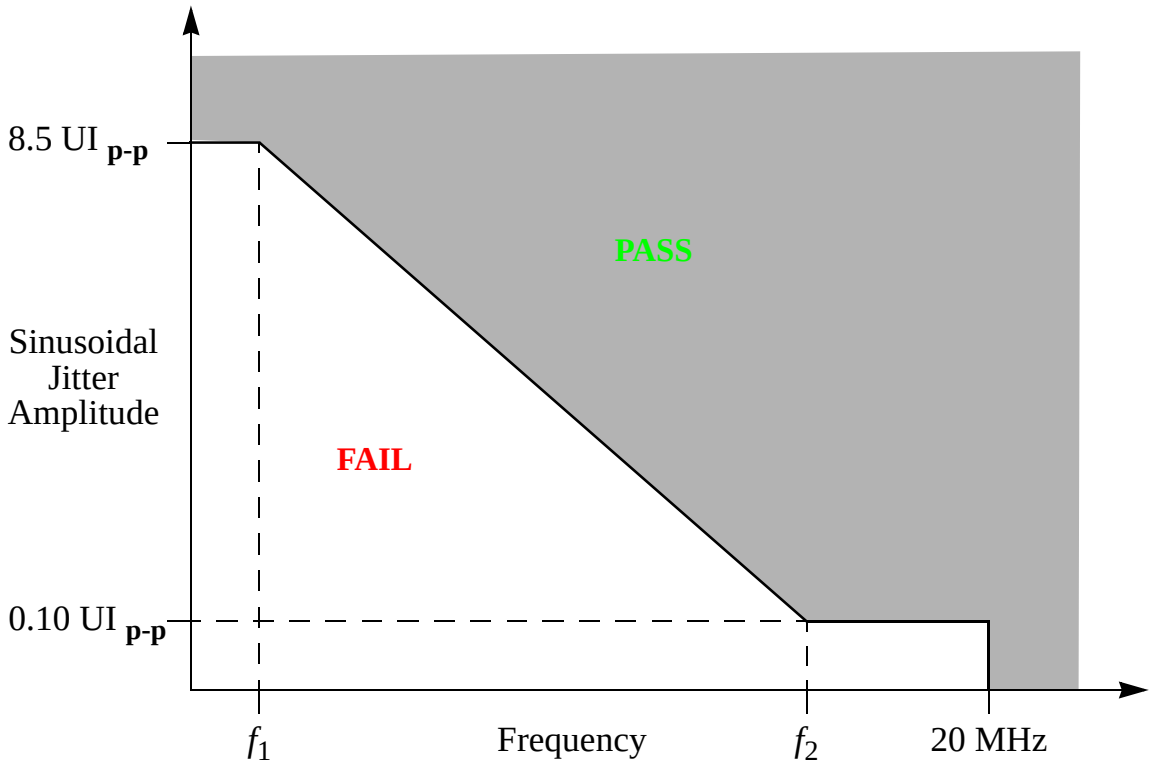


Figure 9-1. Single Frequency Sinusoidal Jitter Limits

Table 9-12 defines the low and high knee frequency for Level I links where the baud rates are defined as in Section 9.4.3.1.

Table 9-12. Level I Single Frequency Sinusoidal Jitter Limits Knee Frequencies

Receiver Data Baud Rate (Gbaud)	f_1 (kHz)	f_2 (kHz)
1.25	8.82	750
2.5	17.6	1500
3.125	22.1	1875

For each baud rate at which a LP-Serial receiver is specified to operate, the receiver shall meet the corresponding Bit Error Ratio specification in Table 9-10 when the eye pattern of the receiver test signal (exclusive of sinusoidal jitter) falls entirely within the unshaded portion of the Receiver Input Compliance Mask shown in Figure 8-5 with the parameters specified in Table 9-13. The eye pattern of the receiver test signal is measured at the input pins of the receiving device with the device replaced with a 100Ω ± 5% differential resistive load.

Table 9-13. Level I Far-End (Rx) Template Intervals

Characteristics	Symbol	Far-End Value	Units
Eye Mask	R_X1	0.275	UI
Eye Mask	R_Y1	100	mV
Eye Mask	R_Y2	800	mV
High Probability Jitter	R_HPJ	0.37	UIpp
Total Jitter (Does not include Sinusoidal Jitter)	R_TJ	0.55	UIpp

9.5 Level I Measurement and Test Requirements

Since the LP-Serial electrical specification is guided by the XAUI electrical interface specified in Clause 47 of IEEE 802.3ae-2002, the measurement and test requirements defined here are similarly guided by Clause 47. In addition, the CJPAT test pattern defined in Annex 48A of IEEE802.3ae-2002 is specified as the test pattern for use in transmitter eye pattern and jitter measurements. Annex 48B of IEEE802.3ae-2002 is recommended as a reference for additional information on jitter test methods.

9.5.1 Level I Transmitter Measurements

9.5.1.1 Level I Eye Template Measurements

For the purpose of transmitter eye template measurements, the effects of a single-pole high pass filter with a 3dB point at (Baud Frequency)/1667 is applied to the jitter. The data pattern for template measurements is the Continuous Jitter Test Pattern (CJPAT) defined in Annex 48A of IEEE802.3ae. All lanes of the LP-Serial link shall be active in both the transmit and receive directions, and opposite ends of the links shall use asynchronous clocks. N lane implementations shall use CJPAT as defined in Annex 48A. Single lane implementations shall use the CJPAT sequence specified in Annex 48A for transmission on lane 0. The amount of data represented in the eye shall be adequate to ensure that the bit error ratio is less than 10^{-12}. The eye pattern shall be measured with AC coupling and the compliance template centered at 0 Volts differential. The left and right edges of the template shall be aligned with the mean zero crossing points of the measured data eye. The load for this test shall be 100Ω resistive ± 5% differential to 2.5GHz.

9.5.1.2 Level I Jitter Test Measurements

For the purpose of transmitter jitter measurement, the effects of a single-pole high pass filter with a 3dB point at (Baud Frequency)/1667 is applied to the jitter. The data pattern for jitter measurements is the Continuous Jitter Test Pattern (CJPAT) pattern defined in Annex 48A of IEEE802.3ae. All lanes of the LP-Serial link shall be active in both the transmit and receive directions, and opposite ends of the links shall use asynchronous clocks. N lane implementations shall use CJPAT as defined

in Annex 48A. Single lane implementations shall use the CJPAT sequence specified in Annex 48A for transmission on lane 0. Jitter shall be measured with AC coupling and at 0 Volts differential. Jitter measurement for the transmitter (or for calibration of a jitter tolerance setup) shall be performed with a test procedure resulting in a BER curve such as that described in Annex 48B of IEEE802.3ae.

9.5.1.3 Level I Transmit Jitter Load

Transmit jitter is measured at the driver output when terminated into a load of 100Ω resistive ± 5% differential to 2.5GHz.

9.5.2 Level I Receiver Jitter Tolerance

Jitter tolerance is measured at the receiver using a jitter tolerance test signal. This signal is obtained by first producing the sum of deterministic and random jitter defined in Section 9.4.3 and then adjusting the signal amplitude until the data eye contacts the 4 points of the minimum eye opening of the receive template shown in Table 8-4 and Table 9-13. Note that for this to occur, the test signal must have vertical waveform symmetry about the average value and have horizontal symmetry (including jitter) about the mean zero crossing. Eye template measurement requirements are as defined above. Random jitter is calibrated using a high pass filter with a low frequency corner at 20MHz and a 20dB/decade rolloff below this. The required sinusoidal jitter specified in Section 9.4.3 is then added to the signal and the test load is replaced by the receiver being tested.

Chapter 10 5Gbaud and 6.25Gbaud LP-Serial Links

This chapter details the requirements for Level II RapidIO LP-Serial short, medium, and long run electrical interfaces of nominal baud rates of 5Gbaud and 6.25Gbaud using NRZ coding (hence 1 bit per symbol at the electrical level). A compliant device must meet all of the requirements listed below. The electrical interface is based on a high speed low voltage logic with a nominal differential impedance of 100Ω. Connections are point-to-point balanced differential pair and signaling is unidirectional.

10.1 Level II Application Goals

10.1.1 Common to Level II Short run, Medium run and Long run

The following are application requirements common to short run, medium run and long run Level II links at 5Gbaud and 6.25Gbaud:

- The electrical specifications shall support lane widths options of 1x, 2x, 4x, 8x and 16x.
- Both AC coupled and DC coupled links options shall be specified. A compliant device must implement AC coupling and may implement DC coupling as an option.
- A compliant device may implement any subset of baud rates contained in this chapter.
- A compliant device may implement either a short run transmitter, a long run transmitter, or both, at each of the baud rates that it supports.
- A compliant device may implement either a short run receiver or a long run receiver at each of the baud rates that it supports.
- The clock frequency tolerance requirement for transmit and receive are ±100ppm. The worst case frequency differences between any transmit and receive clock is 200ppm.
- The Bit Error Ratio (BER) shall be better than 10^{-15} per lane but the test requirements will be to verify 10^{-12} per lane.

- Transmitters and receivers used on short, medium and long run links shall inter-operate for path lengths up to 20cm.
- Transmitters and receivers used on medium and long run links shall inter-operate for path lengths up to 60cm.
- The transmitter pins shall be capable of surviving short circuit either to each other, to supply voltages, and to ground.

10.1.2 Application Goals for Level II Short Run

- The short run interface shall be capable of spanning at least 20cm of PCB material with up to a single connector.

10.1.3 Application Goals for Level II Medium Run

- The medium run interface shall be capable of spanning at least 60cm of PCB material with up to two connectors.
- An AC coupled receiver used for a medium run shall be inter-operable with an AC coupled short run transmitter
- An AC coupled transmitter used for a medium run shall be inter-operable with an AC coupled short run receiver, provided that the signal swing values are lowered. This implies that the signal swing is configurable.
- The medium run PHY may use techniques such as increased signal swing and linear equalization to accommodate medium run backplane applications, where the receiver eye may be closed.

10.1.4 Application Goals for Long Run

- The long run interface shall be capable of spanning at least 100cm of PCB material with up to two connectors.
- An AC coupled long run receiver shall be inter-operable with an AC coupled short or medium run transmitter
- An AC coupled long run transmitter shall be inter-operable with an AC coupled short run receiver provided that the signal swing values are lowered. This implies that the signal swing is configurable.
- The long run PHY may use techniques such as increased signal swing, linear equalization, and Decision Feedback Equalizer, designed to accommodate longer run backplane applications, where the receiver eye may be closed.
- A long run transmitter and receiver is intended to accommodate ‘legacy’ long run RapidIO 1.3 backplanes of at least 60cm with up to two connectors that can operate at data rates up to 6.25Gbaud.

10.1.5 Explanatory Note on Transmitter and Receiver Specifications

AC electrical specifications are given for transmitters and receivers. Long run, medium run and short run interfaces at two baud rates are described.

The parameters for the AC electrical specifications are guided by the OIF CEI Electrical and Jitter Inter-operability agreement for CEI-6G-SR and CEI-6G-LR[Reference 2].

OIF CEI-6G-SR and CEI-6G-LR have similar application goals to serial RapidIO, as described in Section 10.1, "Level II Application Goals". The goal of this standard is that electrical designs for serial RapidIO can reuse electrical designs for OIF CEI-6G, suitably modified for applications at the baud intervals and runs described herein.

10.2 Equalization

At the high baud rates used by Level II LP-Serial links, the signals transmitted over a link are degraded by losses and characteristic impedance discontinuities in the interconnect media. The losses increase with increasing baud rate and interconnect media length and cause signal attenuation and inter-symbol interference that degrade the opening of the eye pattern at both the receiver input and the data decoder decision point. Depending on the baud rate and interconnect length, the degradation can be greater than that allowed by the specification.

The signal degradation can be partially negated by the use of equalization in the transmitter and/or receiver. Equalization in the transmitter can improve the eye pattern at both the receiver input and the data decoder decision point. Equalization in the receiver can only improve the eye pattern at the data decoder decision point. Some degree of equalization is required by most Level II interconnects.

The types of equalizers and, if the equalizers are adaptive, the adaptive equalizer training algorithms that may be used in a Level II 5.0 GBaud transmitter or receiver are subject to the following restrictions.

Equalizers that can convert a single bit error into a multiple bit burst error, such as decision feedback equalizers (DFEs), shall not be used when IDLE1 has been selected for use on the link.

The training algorithm for any adaptive equalization used by a Level II transmitter and/or receiver shall consistently train the equalizer and retain the equalizer’s training when IDLE1 is the training signal and shall consistently retain the equalizer’s training when IDLE1 has been selected for use on the link and the signal on the link is a continuous sequence of maximum length packets whose payload is either all ONES or all ZEROS.

The above restrictions on the types of equalizers and adaptive equalizer training algorithms do not apply to Level II transmitters and receivers operating at greater than 5.5 GBaud.

10.3 Link Compliance Methodology

10.3.1 Overview

A serial link is comprised of a transmitter, a receiver, and a channel which connects them. Typically, two of these are normatively specified, and the third is informatively specified. In this specification, the transmitter and channel are normatively specified, while the receiver is informatively specified.

This specification follows the OIF inter-operability or compliance methodology and is based on using transmitter and receiver reference models, measured channel S-parameters, eye masks, and calculated “statistical eyes”. These “statistical eyes” are determined by the reference models and measured channel S-parameters using publicly available StatEye MATLAB® scripts and form the basis for identifying compliant transmitters and channels. Compliant receivers are identified through a BER test.

Reference models are used extensively because at 5 and 6.25Gbaud data rates the incoming eye at the receiver may be closed. This prevents specifying receiver compliance through receiver eye masks as is typically done at lower data rates.

10.3.2 Reference Models

The OIF serial link reference model is shown in Figure 10-1. The reference models are simple models of the transmitter and receiver equalization with the effects of amplitude, return loss, and bandwidth included. These models do not include any other aspects of transmitter or receiver performance.

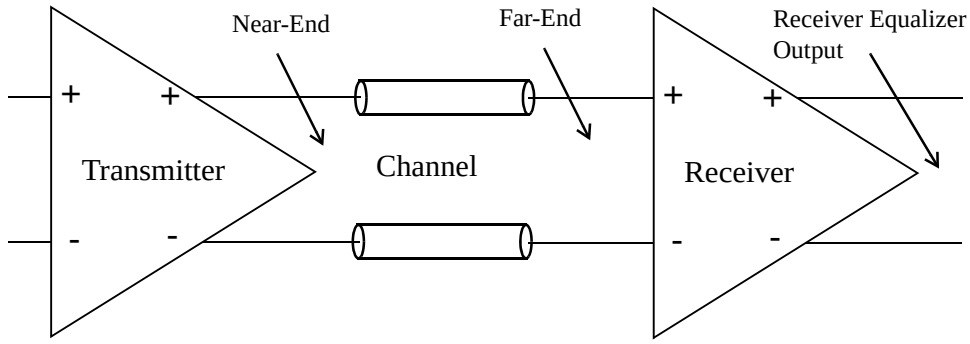


Figure 10-1. OIF Reference Model

There are three target channel run goals in this specification which require various amounts of equalization. These different goals can be met using two transmitter and two receiver reference models. The run goals are short (20cm), medium (60cm), and long (100cm). The reference models for each of the run goals are based on combining short and long run transmitter and receiver models as shown in Table 10-1.

Table 10-1. Reference Models

Run	Tx Reference Model	Rx Reference Model
Short	Short	Short
Medium	Long	Short
Long	Long	Long

NOTES:
Transmitter Reference Models
Short: 1 tap with ≤ 3 dB post cursor emphasis
Long: 1 tap with ≤ 6 dB of either pre or post cursor emphasis
Receiver Reference Model
Short: Single pole, Single zero with ≤ 4 dB max gain
Long: 5 tap DFE

10.3.3 Channel Compliance

A compliant channel is determined using the appropriate transmitter and receiver reference model, measured S-parameters for the channel under consideration, and the StatEye script. A compliant channel is one that produces a receiver equalizer output "statistical eye" which meets a BER $\leq 10^{-15}$ using StatEye.

10.3.4 Transmitter Compliance

The experimental setup for transmitter compliance is shown in Figure 10-2. The shown setup consists of the transmitter under test connected to a compliant channel terminated with a 100Ω differential load. OIF requires the compliant channel used in verifying transmitter compliance use at least half of the available transmitter emphasis to produce an open eye at the far-end of the channel.

Using the shown setup, the following three conditions shall be met for compliant transmitters:

1. After optimally adjusting the transmitter amplitude and emphasis to produce the most open far-end eye (given the transmitter emphasis constraint), the measured far-end eye must be equal or better than the calculated far-end eye as produced by StatEye.
2. The high frequency transmit jitter measured at the near-end must meet specification.
3. The measured near-end transmit eye mask must meet the specified near-end eye mask.

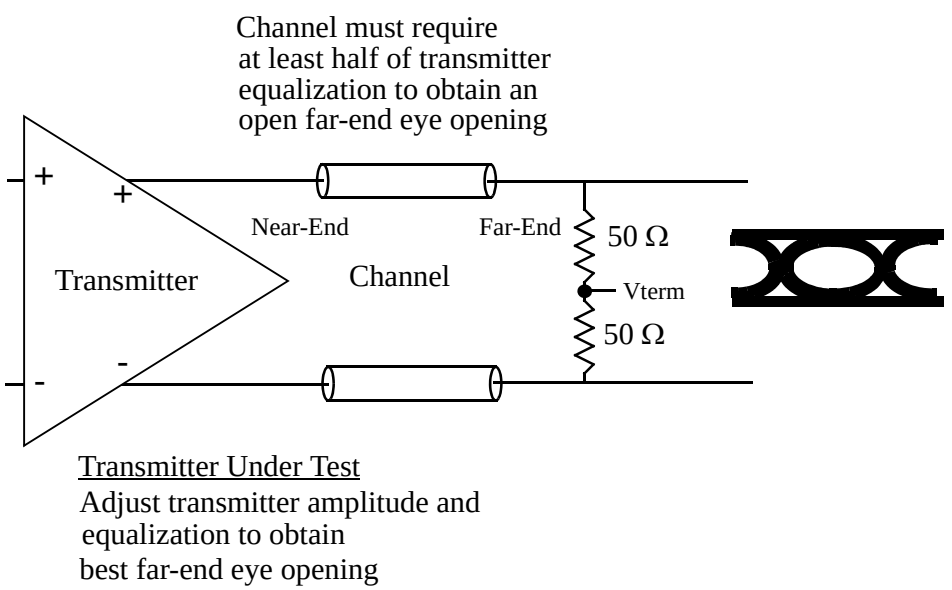


Figure 10-2. Transmitter Compliance Setup

10.3.5 Receiver Compliance

The experimental setup for receiver compliance is shown in Figure 10-3. The shown setup consists of a compliant channel connected to the receiver under test. To verify the receiver under test, the receiver must meet a BER < 10^{-12} with a stressed input eye mask. OIF does not place any requirements on the channel used in this measurement other than it must be compliant.

The input stressed eye used in this measurement includes sinusoidal, high probability, and Gaussian jitter as defined in the appropriate sections of this specification, along with any necessary additive crosstalk. Additive crosstalk is used to insure that the receiver under test is adequately stressed if a low loss channel is used in the measurement.

The additive input crosstalk signal is determined using the channel S-parameters, receiver reference model, and the StatEye script. It must be of amplitude such that the resulting receiver equalizer output eye, given the channel, jitter, and crosstalk, is as close as feasible in amplitude when compared to the defined minimum amplitude used for channel compliance.

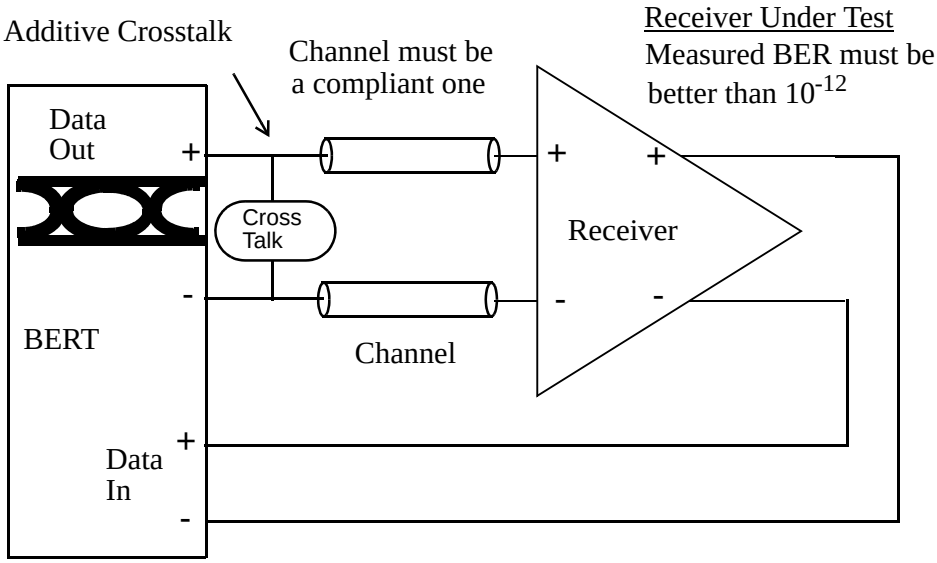


Figure 10-3. Receiver Compliance Setup

10.4 Level II Short Run Interface - General Requirements

10.4.1 Jitter and Inter-operability Methodology

This section describes the requirements for inter-operability testing of the electrical interfaces used to implement a Short Run link. The LP-Serial 5Gbaud and 6.25Gbaud short run interfaces use Method C, described in CEI sub-clause 2.2. This sub-clause defines the inter-operability methodology specifically for interfaces where transmit emphasis may be used and the receiver eye requires Linear Continuous Time equalization (from channel inter-operability point of view) to be open to within the BER of interest.

10.4.1.1 Level II SR Defined Test Patterns[1]

A free running PRBS31 polynomial [ITU-T 0.150] shall be used for the testing of jitter tolerance and output jitter compliance.

10.4.1.2 Level II SR Channel Compliance

The following steps shall be made to identify which channels are to be considered compliant:

1. The forward channel and significant crosstalk channels shall be measured using a network analyzer for the specified baud rate (see Section 10.7.4.5, "Network Analysis Measurement" for a suggested method). Differential S-parameters will be used to represent the characteristics of this channel.
2. The reference transmitter shall be a single post tap transmitter, with ≤ 3dB of emphasis and infinite precision accuracy.
3. A Tx edge rate filter: a single pole 20dB/dec low pass at 75% of baud rate, this is to emulate a Tx -3dB bandwidth at $^3/_4$ baud rate.
4. A transmit amplitude of 400mVppd shall be used.
5. Additional Uncorrelated Bounded High Probability Jitter of 0.15UIpp (emulating part of the Tx jitter).
6. Additional Uncorrelated Unbounded Gaussian Jitter of 0.15UIpp (emulating part of the Tx jitter)
7. The baud rate shall be 5Gbaud or 6.25Gbaud.
8. The reference transmitter shall use the worst case transmitter return loss at the baud frequency. In order to construct the worse case transmitter return loss, the reference transmitter should be considered to be a parallel R and C, where R is the defined maximum allowed DC resistance of the interface and C is increased until the defined maximum Return Loss at the baud frequency is reached. The transmitter return loss is specified in Section 10.4.2.1.6, "Level II SR Transmitter Output Resistance and Return Loss".
9. An ideal receiver filter of the form in CEI Section 8.6.7, "Time Continuous Transverse Filter". The reference receiver uses a continuous-time equalizer with 1 zero and 1 pole in the region of baudrate/100 to baudrate. Additional parasitic zeros and poles must be considered part of the receiver vendor’s device and be dealt with as they are for the reference receiver. Pole and Zero values have infinite precision accuracy. Maximum required gain/attenuation shall be less than or equal to 4dB.
10. The reference receiver shall use a sampling point defined at the midpoint between the average zero crossings of the differential signal.

[1] All descriptions of PRBS31 imply the standard polynomial as described in [Reference 3]

11. The reference receiver shall use the worst case receiver return loss at the baud frequency. In order to construct the worse case receiver return loss, the reference receiver should be considered to be a parallel R and C, where R is the defined maximum allowed DC resistance of the interface and C is increased until the defined maximum Return Loss at the baud frequency is reached. The receiver return loss is specified in Section 10.4.2.2.7, "Level II SR Receiver Input Resistance and Return Loss".
12. The opening of the eye shall be calculated using Statistical Eye Analysis methods, as per Section 8.7.5, "Statistical Eye Methodology", and confirmed to be within the requirements as specified in Table 10-9 at the required BER, 10^{-15}.

10.4.1.3 Level II SR Transmitter Inter-operability

The following step shall be made to identify which transmitters are to be considered compliant:

1. It shall be verified that the measured eye is equal or better than the calculated eye for the given measurement probability Q (see Annex B.3, “Eye Mask Adjustment for Sampling Oscilloscopes") for a suggested method of calculating Q given a measurement population), given:
   - A “compliance” channel as per Section 10.4.1.2, "Level II SR Channel Compliance" that required at least half the maximum transmit emphasis.
   - Using this channel the transmitter shall be then be optimally adjusted and the resulting eye measured (see Section 10.7.4.6, "Eye Mask Measurement Setup" for a suggested method).
   - Using this channel the statistical eye shall then be calculated, as per CEI Section 8.7.5, "Statistical Eye Methodology", using the maximum defined transmit jitter and the actual transmitter's amplitude and emphasis.
2. The high frequency transmit jitter shall be within that specified (see Section 10.7.1, "High Frequency Transmit Jitter Measurement"for suggested methods)
3. The specified transmit eye mask shall not be not violated (see Section 10.7.4.6, "Eye Mask Measurement Setup" for a suggested method) after adjusting the horizontal time positions for the measured time with a confidence level of 3 sigma (see Annex B.3, “Eye Mask Adjustment for Sampling Oscilloscopes" for a suggested method).

10.4.1.4 Level II SR Receiver Inter-operability

The following step shall be made to identify which receivers are to be considered compliant:

1. The DUT shall be measured to have a BER[1] better than 10^{-12} for a stressed signal (see Section 10.7.4.2, "Jitter Tolerance with no Relative Wander Lab Setup" for a suggested method) with a confidence level of three sigma (see Annex B.2, “Confidence Level of Errors Measurement" for a suggested method), given:
   - The defined sinusoidal jitter mask for total and relative wander as per Section 10.4.2.2.8, "Level II SR Receiver Input Jitter Tolerance" with a high frequency total/relative wander and a maximum total/relative wander as defined in the CEI IA.
   - The specified amount of High Probability Jitter and Gaussian jitter per Section 10.4.2.2.8, "Level II SR Receiver Input Jitter Tolerance".
   - An additive crosstalk signal of amplitude such that the resulting statistical eye, given the channel, jitter and crosstalk, is as close as feasible in amplitude when compared to the defined minimum amplitude for channel compliance.

10.4.2 Level II SR Electrical Characteristics

The electrical interface is based on high speed, low voltage logic with nominal differential impedance of 100Ω. Connections are point-to-point balanced differential pair and signalling is unidirectional.

10.4.2.1 Level II SR Transmitter Characteristics

The key transmitter characteristics are summarized in Table 10-2 and Table 10-3 while the following sections fully detail all the requirements.

Table 10-2. Level II SR Transmitter Output Electrical Specifications

Characteristic	Symbol	Condition	Min	Typ	Max	Units
Baud Rate (5Gbaud)	T_Baud	Section 10.4.2.1.2	5.00 -0.01%	5.00	5.00 +0.01%	Gbaud
Baud Rate (6.25Gbaud)			6.25 -0.01%	6.25	6.25 +0.01%	Gbaud
Absolute Output Voltage	V_O	Section 10.4.2.1.3	-0.40		2.30	Volts
Output Differential voltage (into floating load Rload=100Ω)	T_Vdiff	Section 10.4.2.1.3	400		750	mVppd
Differential Resistance	T_Rd	Section 10.4.2.1.6	80	100	120	W

NOTES:
1. Load Type 0 with min T_Vdiff, AC-Coupling or floating load
2. For Load Types 1 through 3: R_Zvtt ≤ 30Ω; Vtt is defined for each load type as follows: Load Type 1 R_Vtt = 1.2V +5%/-8%; Load Type 2 R_Vtt = 1.0V +5%/-8%; Load Type 3 R_Vtt = 0.8V +5%/-8%.
3. DC Coupling compliance is optional (Type 1 through 3). Only Transmitters that support DC coupling are required to meet this parameter. It is acceptable for a transmitter to restrict the range of T_Vdiff in order to comply with the specified T_Vcm range. For a transmitter which supports multiple T_Vdiff levels, it is acceptable for a transmitter to claim DC Coupling Compliance if it meets the T_Vcm ranges for at least one of its T_Vdiff setting as long as those setting(s) that are compliant are indicated.

[1] if the defined measurement BER is different to system required BER, adjustments to applied stressed eye TJ are necessary

Table 10-2. Level II SR Transmitter Output Electrical Specifications

Characteristic	Symbol	Condition	Min	Typ	Max	Units
Recommended output rise and fall times (20% to 80%)	T_tr, T_tf	Section 10.4.2.1.4	30			ps
Skew between signals comprising a differential pair	T_SKEW_{diff}	Section 10.4.2.1.5			15	ps
Differential Output Return Loss (100 MHz to 0.5*T_Baud)	T_SDD22	Section 10.4.2.1.6			-8	dB
Differential Output Return Loss (0.5*T_Baud to T_Baud)						
Common Mode Return Loss (100 MHz to 0.75 *T_Baud)	T_SCC22	Section 10.4.2.1.6			-6	dB
Transmitter Common Mode Noise	T_Ncm				5% of T_Vdiff	mVppd
Output Common Mode Voltage	T_Vcm	Load Type 0[1] Section 8.5.3	100		1700	mV
		Load Type 1[2,3] Section 8.5.3	630		1100	mV

NOTES:
1. Load Type 0 with min T_Vdiff, AC-Coupling or floating load
2. For Load Types 1 through 3: R_Zvtt ≤ 30Ω; Vtt is defined for each load type as follows: Load Type 1 R_Vtt = 1.2V +5%/-8%; Load Type 2 R_Vtt = 1.0V +5%/-8%; Load Type 3 R_Vtt = 0.8V +5%/-8%.
3. DC Coupling compliance is optional (Type 1 through 3). Only Transmitters that support DC coupling are required to meet this parameter. It is acceptable for a transmitter to restrict the range of T_Vdiff in order to comply with the specified T_Vcm range. For a transmitter which supports multiple T_Vdiff levels, it is acceptable for a transmitter to claim DC Coupling Compliance if it meets the T_Vcm ranges for at least one of its T_Vdiff setting as long as those setting(s) that are compliant are indicated.

Table 10-3. Level II SR Transmitter Output Jitter Specifications

Characteristic	Symbol	Conditions	Min	Typ	Max	Units
Uncorrelated High Probability Jitter	T_UHPJ	Section 10.4.2.1.8			0.15	UIpp
Duty Cycle Distortion	T_DCD	Section 10.4.2.1.8			0.05	UIpp
Total Jitter	T_TJ	Section 10.4.2.1.8			0.30	UIpp
Eye Mask	T_X1	Section 10.4.2.1.8			0.15	UI
Eye Mask	T_X2	Section 10.4.2.1.8			0.40	UI
Eye Mask	T_Y1	Section 10.4.2.1.8	200			mV
Eye Mask	T_Y2	Section 10.4.2.1.8			375	mV

10.4.2.1.1 Level II SR Transmitter Test Load

All transmitter characteristics should be implemented and measured to a differential impedance of 100W ± 1% at DC with a return loss of better than 20dB from baud rate divided by 1667 to 1.5 times the baud rate, unless otherwise noted.

10.4.2.1.2 Level II SR Transmitter Baud Rate

The baud rates are 5Gbaud and 6.25Gbaud with a tolerance of ±100ppm.

10.4.2.1.3 Level II SR Transmitter Amplitude and Swing

Transmitter differential output amplitude shall be between 400 and 750mVppd, inclusive, either with or without any transmit emphasis. Absolute transmitter output voltage shall be between -0.1V and 1.9V, inclusive, with respect to local ground. See Figure 8-1 for an illustration of absolute transmitter output voltage limits and definition of differential peak-to-peak amplitude.

10.4.2.1.4 Level II SR Transmitter Rise and Fall Times

The recommended minimum differential rise and fall times are 30ps as measured between the 20% and 80% of the maximum measured levels; the maximum differential rise and fall times are defined by the Tx eye diagram Figure 8-2 and Table 10-5. Shorter rise and fall times may result in excessive high frequency components and increase EMI and cross talk.

10.4.2.1.5 Level II SR Transmitter Differential Pair Skew

The timing skew at the output of a Level II SR transmitter between the two signals that comprise a differential pair shall not exceed 15 ps at 5.0 Gbaud and 6.25 Gbaud.

10.4.2.1.6 Level II SR Transmitter Output Resistance and Return Loss

Refer to Section 8.5.11, "Differential Resistance and Return Loss, Transmitter and Receiver" for the reference model for return loss. See Table 10-4 for 5Gbaud and 6.25Gbaud short run transmitter parameters. Definitions for these parameters are in Figure 8-12.

Table 10-4. Level II SR Transmitter Return Loss Parameters

Parameter	Value	Units
A0	-8	dB
f0	100	MHz
f1	T_Baud/2	Hz
f2	T_Baud	Hz
Slope	16.6	dB/dec

10.4.2.1.7 Level II SR Transmitter Lane-to-Lane Skew

The electrical level of lane-to-lane skew caused by the transmitter circuitry and associated routing must be less than 1000 ps for links of 4 lanes or less. Links with greater than 4 lanes must have lane-to-lane skew of less than 2UI + 1000 ps. The transmitter lane-to-lane skew is only for the serdes Tx and does not include any effects of the channel.

10.4.2.1.8 Level II SR Transmitter Template and Jitter

As per Section 10.4.1.3, "Level II SR Transmitter Inter-operability" the transmitter shall satisfy both the near-end and far-end eye template and jitter requirements as

given in Figure 8-2, Table 10-5, Figure 8-5, and Table 10-9 either with or without any transmit emphasis.

The maximum near-end duty cycle distortion (T_DCD) shall be less than 0.05UIpp.

It should be noted that it is assumed the Uncorrelated High Probability Jitter component of the transmitter jitter is not Inter-symbol Interference (ISI). This is only assumed from a receiver point of view and does not in any way put any restrictions on the real transmitter HPJ.

Table 10-5. Level II SR Near-End (Tx) Template Intervals

Characteristics	Symbol	Near-End Value	Units
Eye Mask	T_X1	0.15	UI
Eye Mask	T_X2	0.40	UI
Eye Mask	T_Y1	200	mV
Eye Mask	T_Y2	375	mV
Eye Mask	T_Y3	125	mV
Uncorrelated Bounded High Probability Jitter	T_UBHPJ	0.15	UIpp
Duty Cycle Distortion	T_DCD	0.05	UIpp
Total Jitter	T_TJ	0.30	UIpp

10.4.2.2 Level II SR Receiver Characteristics

The key receiver characteristics are summarized in Table 10-6 and Table 10-7 while the following sections fully detail all the requirements.

Table 10-6. Level II SR Receiver Electrical Input Specifications

Characteristic	Symbol	Conditions	Min	Typ	Max	Units
Rx Baud Rate (5Gbaud)	R_Baud	Section 10.4.2.2.1	5.00 -0.01%	5.00	5.00 +0.01%	Gbaud
Rx Baud Rate (6.25Gbaud)			6.25 -0.01%	6.25	6.25 +0.01%	Gbaud
Absolute Input Voltage	R_Vin	Section 10.4.2.2.4				
Input Differential voltage	R_Vdiff	Section 10.4.2.2.3	125		1200	mVppd
Differential Resistance	R_Rdin	Section 10.4.2.2.7	80	100	120	Ω
Bias Voltage Source Impedance (load types 1 to 3)[1]	R_Zvtt				30	Ω

NOTES:
1. DC Coupling compliance is optional. For Vcm definition, see Figure 8-1.
2. Receiver is required to implement at least one of specified nominal R_Vtt values, and typically implements only one of these values. Receiver is only required to meet R_Vrcm parameter values that correspond to R_Vtt values supported.
3. Input common mode voltage for AC-coupled or floating load input with min. T_Vdiff.
4. For floating load, input resistance must be ≥ 1kΩ.

Table 10-6. Level II SR Receiver Electrical Input Specifications

Characteristic	Symbol	Conditions	Min	Typ	Max	Units
Differential Input Return Loss (100MHz to 0.5*R_Baud)	R_SDD11	Section 10.4.2.2.7			-8	dB
Differential Input Return Loss (0.5*R_Baud to R_Baud))						
Common mode Input Return Loss (100MHz to 0.5*R_Baud)	R_SCC11	Section 10.4.2.2.7			-6	dB
Termination Voltage[1,2]	R_Vtt	R_Vtt floating[4]	Not Specified			V
		R_Vtt = 1.2V Nominal	1.2 - 8%		1.2 + 5%	V
		R_Vtt = 1.0V Nominal	1.0 - 8%		1.0 + 5%	V
		R_Vtt = 0.8V Nominal	0.8 - 8%		0.8 + 5%	V
Input Common Mode Voltage	R_Vfcm	Load Type 0[2]	0		1800	mV
		Load Type 1[1,3]	595		R_Vtt - 60	mV
Wander divider (in Figure 8-8 & Figure 8-9)	n			10		

NOTES:
1. DC Coupling compliance is optional. For Vcm definition, see Figure 8-1.
2. Receiver is required to implement at least one of specified nominal R_Vtt values, and typically implements only one of these values. Receiver is only required to meet R_Vrcm parameter values that correspond to R_Vtt values supported.
3. Input common mode voltage for AC-coupled or floating load input with min. T_Vdiff.
4. For floating load, input resistance must be ≥ 1kΩ.

Table 10-7. Level II SR Receiver Input Jitter Tolerance Specifications

Characteristics	Symbol	Conditions	Min	Typ	Max	Units
Bounded High Probability Jitter	R_BHPJ	Section 10.4.2.2.8			0.45	UIpp
Sinusoidal Jitter, maximum	R_SJ-max	Section 10.4.2.2.8			5	UIpp
Sinusoidal Jitter, High Frequency	R_SJ-hf	Section 10.4.2.2.8			0.05	UIpp
Total Jitter (Does not include Sinusoidal Jitter)	R_TJ	Section 10.4.2.2.8			0.60	UIpp
Eye Mask	R_X1	Section 10.4.2.2.8			0.30	UI
Eye Mask	R_Y1	Section 10.4.2.2.8			62.5	mV
Eye Mask	R_Y2	Section 10.4.2.2.8			375	mV

10.4.2.2.1 Level II SR Receiver Input Baud Rate

All devices shall work at 5Gbaud, 6.25Gbaud or both baud rates with the baud rate tolerance as per Section 8.5.12, "Baud Rate Tolerance".

10.4.2.2.2 Level II SR Receiver Reference Input Signals

Reference input signals to the receiver have the characteristics determined by compliant transmitter. The reference input signal must satisfy the transmitter near-end template and jitter given in Figure 8-2 and Table 10-5, as well as the

far-end eye template and jitter given in Figure 8-5 and Table 10-9, with the differential load impedance of 100Ω ± 1% at DC with a return loss of better than 20 dB from baud rate divided by 1667 to 1.5 times the baud rate. Note that the input signal might not meet either of these templates when the actual receiver replaces this load.

10.4.2.2.3 Level II SR Receiver Input Signal Amplitude

The receiver shall accept differential input signal amplitudes produced by compliant transmitters connected without attenuation to the receiver. This may be larger than the 1200mVppd maximum of the transmitter due to output/input impedances and reflections.

The minimum input amplitude is defined by the far-end transmitter template, the actual receiver input impedance, and the loss of the actual PCB. Note that the far-end transmitter template is defined using a well controlled load impedance, however, the real receiver is not, which can leave the receiver input signal smaller than the minimum 125mVppd.

10.4.2.2.4 Level II SR Receiver Absolute Input Voltage

The absolute voltage levels with respect to the receiver ground at the input of the receiver are dependent on the transmitter implementation, the inter-ground difference, whether the receiver is AC or DC coupled, and (in the case of DC coupling load types 1 to 3) the nominal R_Vtt supported by the receiver. The voltage levels at the input of a DC coupled receiver shall be consistent with the R_Vrcm and R_Vdiff values defined in Table 10-6.

The voltage levels at the input of an AC coupled receiver (if AC coupling is done within the receiver) or at the Tx side of the external AC coupling cap (if AC coupling is done externally) shall be between -0.15 and 1.95V, inclusive, with respect to local ground.

10.4.2.2.5 Level II SR Receiver Input Common Mode Impedance

The input common mode impedance (R_Zvtt) at the input of the receiver is dependent on whether the receiver is AC or DC coupled. The value of R_Zvtt as measured at the input of an AC coupled receiver is undefined. The value of R_Zvtt as measured at the input of a DC coupled receiver is defined as per Table 10-6.

If AC coupling is used it is to be considered part of the receiver for the purposes of this specification unless explicitly stated otherwise. It should be noted that various methods for AC coupling are allowed (for example, internal to the chip or done externally). See Section 8.5.13, "Termination and DC Blocking" for more information.

10.4.2.2.6 Level II SR Receiver Input Lane-to-Lane Skew

Lane-to-lane skew at the input to the receiver shall not exceed 70UI peak. See Section 8.5.9, "Receiver Input Lane-to-Lane Skew".

10.4.2.2.7 Level II SR Receiver Input Resistance and Return Loss

Refer to Section 8.5.11, "Differential Resistance and Return Loss, Transmitter and Receiver" for the reference model for return loss. See Table 10-8 for 5Gbaud and 6.25Gbaud short run receiver parameters. Definitions for these parameters are in Figure 8-12.

Table 10-8. Level II SR Input Return Loss Parameters

Parameter	Value	Units
A0	-8	dB
f0	100	MHz
f1	R_Baud/2	Hz
f2	R_Baud	Hz
Slope	16.6	dB/dec

10.4.2.2.8 Level II SR Receiver Input Jitter Tolerance

As per Section 10.4.1.4, "Level II SR Receiver Inter-operability", the receiver shall tolerate at least the far-end eye template and jitter requirements as given in Figure 8-5 and Table 10-9 with an additional SJ with any frequency and amplitude defined by the mask of Figure 8-9 where the minimum and maximum total wander amplitude are 0.05UIpp and 5UIpp respectively. This additional SJ component is intended to ensure margin for wander, hence is over and above any high frequency jitter from Table 10-9.

Table 10-9. Level II SR Far-End (Rx) Template Intervals

Characteristics	Symbol	Far-End Value	Units
Eye Mask	R_X1	0.30	UI
Eye Mask	R_Y1	62.5	mV
Eye Mask	R_Y2	375	mV
Uncorrelated Bounded High Probability Jitter	R_UBHPJ	0.15	UIpp
Correlated Bounded High Probability Jitter	R_CBHPJ	0.30	UIpp
Total Jitter (Does not include Sinusoidal Jitter)	R_TJ	0.60	UIpp

10.4.2.3 Level II SR Link and Jitter Budgets

The primary intended application is as a point-to-point interface of up to approximately 20cm and up to one connector between integrated circuits using controlled impedance traces on low-cost printed circuit boards (PCBs). Informative loss and jitter budgets are presented in Table 10-10 to demonstrate the feasibility of legacy FR4 epoxy PCB’s. The jitter budget is given in Table 10-11. The performance of an actual transceiver interconnect is highly dependent on the implementation.

Table 10-10. Level II SR Informative Loss, Skew and Jitter Budget

Description	Loss (dB)	Differential Skew (ps)	Bounded High Probability (UIpp)	TJ (UIpp)
Driver	0	15	0.15	0.30
Interconnect (with Connector)	6.6	25	0.15	0.15
Other	3.5		0.15	0.15
Total	10.1	40	0.45	0.60

Table 10-11. Level II SR High Frequency Jitter Budget

CEI-6G-SR	Uncorrelated Jitter		Correlated Jitter		Total Jitter				Amplitude	
	Unbounded Gaussian	High Probability	Bounded Gaussian	Bounded High Probability	Gaussian	Sinusoidal	Bounded High Probability	Total		
Abbreviation	UUGJ	UHPJ	CBGJ	CBHPJ	GJ	SJ	HPJ	TJ	k	
Units	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp		mVppd
Transmitter	0.150	0.150		-0.200[1]	0.150		-0.050	0.100		400.0
Channel				0.500						
Receiver Input	0.150	0.150	0.000	0.300	0.150		0.450	0.600	0.25	125
Clock + Sampler	0.150	0.100		0.100						-50.0
Budget	0.212	0.250	0.000	0.400	0.212	0.050	0.650	0.912	0.13	75.0

NOTES:
1. Due to transmitter emphasis, it reduces the ISI as seen at the receiver. Thus this number is negative.

10.4.3 Level II SR StatEye.org Template

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% example template for setting up a standard, i.e. equalizer
% jitter and return loss

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

param.version = [param.version '_v1.0'];

% these are internal variables and should not be changed

param.scanResolution = 0.01;
param.binsize                          = 0.0005;
param.points                           = 2^13;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the transmitter and baud rate. The tx filter has two
% parameters defined for the corner frequency of the poles

param.bps                              = 6.25e9;
param.bitResolution                    = 1/(4*param.bps);
param.txFilter                         = 'singlepole';
param.txFilterParam                    = [0.75];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the return loss up. The return loss can be turned off
% using the appropriate option

param.returnLoss                       = 'on';
param.cpad                             = 1.0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the transmitter emphasis up. Some example setting are
% included which can be uncommented

% single tap emphasis
param.txpre                            = [];
param.signal                           = 1.0;
param.txpost                           = [-0.1];
param.vstart                           = [-0.3 -0.3];
param.vend                             = [+0.0 +0.0];
param.vstep                            = [0.1 0.05 0.025];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the de-emphasis of 4-point transmit pulse
```

```
% the de-emphasis run if param.txpre = [] and param.txpost = []

param.txdeemphasis = [1 1 1 1];            % de-emphasis is off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the data coding changing the transmit pulse spectrum
% the coding run if param.txpre = [] and param.txpost = []

param.datacoding = 1;        % the coding is off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set PAM amplitude and rate

param.PAM = 2;              % PAM is switched off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the rxsample point does not need to be changed as it is
% automatically adjusted by the optimization scripts.
% The number of DFE taps should be set, however, the initial
% conditions are irrelevant.

param.rxsample                      = -0.1;

% no DFE
param.dfe                           = [];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% sampling jitter in HPJpp and GJrms is defined here

param.txdj                          = 0.15;
param.txrj                          = 0.15/(2*7.94);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the following options are not yet implemented and should
% not be changed

param.user                          = [0.0];
param.useuser                       = 'no';
param.usesymbol                     = '';
param.xtAmp                         = 1.0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

param.TransmitAmplitude = 0.400; % mVppdif
param.MinEye            = 0.125; % mVppdif

param.Q                 = 2*7.94;
```

```
param.maxDJ             = 0.30;
param.maxTJ             = 0.60;
```

10.5 Level II Long Run Interface General Requirements

10.5.1 Long Run Jitter and Inter-operability Methodology

The LP-Serial 5Gbaud and 6.25Gbaud short run interfaces use Method D, described in CEI clause 2.4. This section defines the inter-operability methodology specifically for interfaces where transmit emphasis may be used and the receiver eye requires DFE equalization (from channel inter-operability point of view) to be open to within the BER of interest.

10.5.1.1 Level II LR Channel Compliance

The following steps shall be made to identify which channels are to be considered compliant:

1. The forward channel and significant crosstalk channels shall be measured using a network analyzer for the specified baud rate (see Section 10.7.4.5, "Network Analysis Measurement" for a suggested method).
2. A single pre or post tap transmitter with ≤ 6dB of emphasis, with infinite precision accuracy.
3. A Tx edge rate filter: a two-pole 40dB/dec low pass at 75% of baud rate, this is to emulate both Rx and Tx -3dB bandwidths at $^3/_4$ baud rate.
4. A transmit amplitude of 800mVppd.
5. Additional Uncorrelated Bounded High Probability Jitter of 0.15UIpp (emulating part of the Tx jitter).
6. Additional Uncorrelated Unbounded Gaussian Jitter of 0.15UIpp (emulating part of the Tx jitter).
7. The reference transmitter shall use the worst case transmitter return loss at the baud frequency. In order to construct the worse case transmitter return loss, the reference transmitter should be considered to be a parallel R and C, where R is the defined maximum allowed DC resistance of the interface and C is increased until the defined maximum Return Loss at the baud frequency is reached. The transmitter return loss is specified in Section 10.5.2.1.6, "Level II LR Transmitter Output Resistance and Return Loss".
8. An ideal receiver filter of the form in Section 8.6.6, "Decision Feedback Equalizer". The reference receiver uses a 5 tap DFE, with infinite precision accuracy and having the following restriction on the coefficient values:

   Let W[N] be sum of DFE tap coefficient weights from taps N through M where

N = 1 is previous decision (i.e. first tap)
M = oldest decision (i.e. last tap)
R_Y2 = T_Y2 = 400mV
Y = min(R_X1, (R_Y2 - R_Y1) / R_Y2) = 0.30
$Z = {}^{2}/_{3} = 0.66667$

Then $W[N] \leq Y * Z^{(N - 1)}$

For the channel compliance model the number of DFE taps (M) = 5. This gives the following maximum coefficient weights for the taps:

W[1] ≤ 0.2625 (sum of taps 1 to 5)
W[2] ≤ 0.1750 (sum of taps 2 to 5)
W[3] ≤ 0.1167 (sum of taps 3 to 5)
W[4] ≤ 0.0778 (sum of taps 4 and 5)
W[5] ≤ 0.0519 (tap 5)

Notes:
- These coefficient weights are absolute assuming a T_Vdiff of 1 Vppd
- For a real receiver the restrictions on tap coefficients would apply for the actual number of DFE taps implemented (M)

9. The reference receiver shall use the worst case receiver return loss at the baud frequency. In order to construct the worse case receiver return loss, the reference receiver should be considered to be a parallel R and C, where R is the defined maximum allowed DC resistance of the interface and C is increased until the defined maximum Return Loss at the baud frequency is reached. The receiver return loss is specified in Section 10.5.2.2.7, "Level II LR Receiver Input Resistance and Return Loss".

Table 10-12. Level II LR Receiver Equalization Output Eye Mask

Parameter	Symbol	Max	Units
Eye mask	R_X1	0.30	UI
Eye mask	R_Y1	50	mV
Bounded High Probability Jitter	R_BHPJ	0.325	UI

10. Any parameters that have degrees of freedom (e.g. filter coefficients or sampling point) shall be optimized against the amplitude, at the zero phase offset, as generated by the Statistical Eye Output, e.g. by sweeping all degrees of freedom and selecting the parameters giving the maximum amplitude. A receiver return loss, as defined by the reference receiver, shall be used.

11. The opening of the eye shall be calculated using Statistical Eye Analysis methods, as per Section 8.7.5, "Statistical Eye Methodology", and confirmed to be within the requirements of the equalized eye mask as specified in Table 10-12 at the required BER, 10^{-15}.

10.5.1.2 Level II LR Transmitter Inter-operability

The following step shall be made to identify which transmitters are to be considered compliant:

1. It shall be verified that the measured eye is equal or better than the calculated eye for the given measurement probability Q (see Annex B.3, “Eye Mask Adjustment for Sampling Oscilloscopes" for a suggested method of calculating Q given a measurement population), given:
   - A “compliance” channel as per Section 10.5.1.1, "Level II LR Channel Compliance" that required at least half the maximum transmit emphasis with no receiver filtering to give an open eye.
   - Using this channel the transmitter shall be then optimally adjusted and the resulting near-end eye measured (see Section 10.7.4.6, "Eye Mask Measurement Setup" for a suggested method).
   - Using this channel the statistical eye shall then be calculated, as per Section 8.7.5, "Statistical Eye Methodology", using the maximum defined transmit jitter and the actual transmitter's amplitude and emphasis.

If the transmit jitter or transmit eye mask is additionally defined then the following steps shall also be made to identify which transmitters are to be considered compliant:

1. The high frequency transmit jitter shall be within that specified (see Section 10.7.1, "High Frequency Transmit Jitter Measurement" for suggested methods).

The specified transmit eye mask shall not be violated (see Section 10.7.4.6, "Eye Mask Measurement Setup" for a suggested method) after adjusting the horizontal time positions for the measured time with a confidence level of 3 sigma (see Annex B.3, “Eye Mask Adjustment for Sampling Oscilloscopes" for a suggested method).

10.5.1.3 Level II LR Receiver Inter-operability

The following step shall be made to identify which receivers are to be considered compliant:

1. The DUT shall be measured to have a BER[1] better than specified for a stressed signal (see Section 10.7.4.3, "Jitter Tolerance with Defined ISI and no Relative Wander" for a suggested method) with a confidence level of three sigma (see Annex B.2, “Confidence Level of Errors Measurement" for a suggested method), given:
   - The defined sinusoidal jitter mask for relative wander as per Section 8.4.6, "Relative Wander Mask" with a high frequency relative wander and a maximum relative wander as defined in Section 10.5.2.2.8, "Level II LR Receiver Jitter Tolerance".
   - The specified amount of High Probability Jitter and Gaussian jitter as defined in Section 10.5.2.2.8, "Level II LR Receiver Jitter Tolerance".
   - A compliance channel or filter as identified by Section 10.5.1.1, "Level II LR Channel Compliance".
   - An additive crosstalk signal of amplitude such that the resulting statistical eye, given the channel, jitter, and crosstalk, is as close as feasible in amplitude when compared to the defined minimum amplitude for channel compliance.

10.5.2 Level II LR Interface Electrical Characteristics

The electrical interface is based on high speed, low voltage logic with nominal differential impedance of 100Ω. Connections are point-to-point balanced differential pair and signalling is unidirectional.

10.5.2.1 Level II LR Transmitter Characteristics

The key transmitter characteristics are summarized in Table 10-13 and Table 10-14 while the following sections fully detail all the requirements.

[1] if the defined measurement BER is different to system required BER, adjustments to applied stressed eye TJ are necessary

Table 10-13. Level II LR Transmitter Output Electrical Specifications

Characteristics	Symbols	Conditions	Min	Typ	Max	Units
Tx Baud Rate (5Gbaud)	T_Baud	Section 10.5.2.1.2	5.00 -0.01%	5.00	5.00 +0.01%	Gbaud
Tx Baud Rate (6.25Gbaud)			6.25 -0.01%	6.25	6.25 +0.01%	Gbaud
Absolute Output Voltage	V_O	Section 10.5.2.1.3	-0.40		2.30	Volts
Output Differential voltage (into floating load Rload=100Ω)	T_Vdiff	Section 10.5.2.1.3[1]	800		1200	mVppd
Differential Resistance	T_Rd	Section 10.5.2.1.6	80	100	120	Ω
Recommended output rise and fall times (20% to 80%)	T_tr, T_tf	Section 10.5.2.1.4	30			ps
Skew between signals comprising a differential pair	T_SKEW_{diff}	Section 10.5.2.1.5			15	ps
Differential Output Return Loss (100 MHz to 0.5*T_Baud)	T_SDD22	Section 10.5.2.1.6			-8	dB
Differential Output Return Loss (0.5*T_Baud to T_Baud)						
Common Mode Return Loss (100 MHz to 0.75 *T_Baud)	T_S11	Section 10.5.2.1.6			-6	dB
Transmitter Common Mode Noise	T_Ncm				5% of T_Vdiff	mVppd
Output Common Mode Voltage	T_Vcm	Load Type 0[2] Section 8.5.3	100		1700	mV
		Load Type 1[3,4] Section 8.5.3	630		1100	mV

NOTES:
1. The Transmitter must be capable of producing a minimum T_Vdiff greater than or equal to 800mVppd. In applications where the channel is better than the worst case allowed, a Transmitter device may be provisioned to produce T_Vdiff less than this minimum value, but greater than or equal to 400mVppd, and is still compliant with this specification.
2. Load Type 0 with min T_Vdiff, AC-Coupling or floating load.
3. For Load Type 1: R_Zvtt ≤ 30 Ω; T_Vtt & R_Vtt = 1.2V +5%/-8%.
4. DC Coupling compliance is optional (Load Type 1). Only Transmitters that support DC coupling are required to meet this parameter.

Table 10-14. Level II LR Transmitter Output Jitter Specifications

Characteristics	Symbol	Conditions	Min	Typ	Max	Units
Uncorrelated High Probability Jitter	T_UHPJ	Section 10.5.2.2.8			0.15	UIpp
Duty Cycle Distortion	T_DCD	Section 10.5.2.2.8			0.05	UIpp
Total Jitter	T_TJ	Section 10.5.2.2.8			0.30	UIpp
Eye Mask	T_X1	Section 10.5.2.2.8			0.15	UI
Eye Mask	T_X2	Section 10.5.2.2.8			0.40	UI
Eye Mask	T_Y1	Section 10.5.2.2.8	200			mV
Eye Mask	T_Y2	Section 10.5.2.2.8			600	mV

10.5.2.1.1 Level II LR Transmitter Test Load

All transmitter characteristics should be implemented and measured to a differential impedance of 100Ω ± 1% at DC with a return loss of better than 20dB from baud rate divided by 1667 to 1.5 times the baud rate, unless otherwise noted.

10.5.2.1.2 Level II LR Transmitter Baud Rate

The baud rates are 5Gbaud and 6.25Gbaud with a tolerance of ±100ppm.

10.5.2.1.3 Level II LR Transmitter Amplitude and Swing

Transmitter differential output amplitude shall be able to drive between 800 and 1200mVppd, inclusive, either with or without any transmit emphasis. DC referenced logic levels are not defined since the receiver must have high common mode impedance at DC. However, absolute transmitter output voltage shall be between -0.1V and 1.9V, inclusive, with respect to local ground. See Figure 8-1 for an illustration of absolute transmitter output voltage limits and definition of differential peak-to-peak amplitude.

10.5.2.1.4 Level II LR Transmitter Rise and Fall Times

The recommended minimum differential rise and fall time is 30ps as measured between the 20% and 80% of the maximum measured levels; the maximum differential rise and fall times are defined by the Tx eye diagram (Figure 8-2 and Table 10-16). Shorter rise and falls may result in excessive high frequency components and increase EMI and cross talk.

10.5.2.1.5 Level II LR Transmitter Differential Pair Skew

The timing skew at the output of a Level II LR transmitter between the two signals that comprise a differential pair shall not exceed 15 ps at 5.0 Gbaud and 6.25 Gbaud.

10.5.2.1.6 Level II LR Transmitter Output Resistance and Return Loss

Refer to Section 8.5.11, "Differential Resistance and Return Loss, Transmitter and Receiver" for the reference model for return loss. See Table 10-15 for 5Gbaud and 6.25Gbaud long run transmitter parameters. Definitions for these parameters are in Figure 8-12.

Table 10-15. Level II LR Transmitter Return Loss Parameters

Parameter	Value	Units
A0	-8	dB
f0	100	MHz
f1	T_Baud/2	Hz
f2	R_Baud	Hz
Slope	16.6	dB/dec

10.5.2.1.7 Level II LR Transmitter Lane-to-Lane Skew

The electrical level of lane-to-lane skew caused by the transmitter circuitry and associated routing must be less than 1000 ps for links of 4 lanes or less. Links with greater than 4 lanes must have lane-to-lane skew of less than 2UI + 1000 ps. The transmitter lane-to-lane skew is only for the serdes Tx and does not include any effects of the channel.

10.5.2.1.8 Level II LR Transmitter Short Circuit Current

The max DC current into or out of the transmitter pins when either shorted to each other or to ground shall be ±100mA when the device is fully powered up. From a hot swap point of view, the ±100mA limit is only valid after 10μs.

10.5.2.1.9 Level II LR Transmitter Template and Jitter

The transmitter shall satisfy both the near-end eye template and jitter requirements as given in Figure 8-2 and Table 10-16 either with or without any transmit emphasis.

The maximum near-end duty cycle distortion (T_DCD) shall be less than 0.05UIpp.

It should be noted that it is assumed the Uncorrelated High Probability Jitter component of the transmitter jitter is not Inter-symbol Interference (ISI). This is only assumed from a receiver point of view so that a receiver can't equalize it and does not in any way put any restrictions on the real transmitter HPJ.

Table 10-16. Level II LR Near-End Template Intervals

Characteristics	Symbol	Near-End Value	Units	Comments
Eye Mask	T_X1	0.15	UI	
Eye Mask	T_X2	0.40	UI	
Eye Mask	T_Y1	200	mV	For connection to short run Rx
		400		For connection to long run Rx
Eye Mask	T_Y2	375	mV	For connection to short run Rx
		600		For connection to long run Rx
Uncorrelated Bounded High Probability Jitter	T_UBHPJ	0.15	UIpp	
Duty Cycle Distortion	T_DCD	0.05	UIpp	
Total Jitter	T_TJ	0.30	UIpp	

10.5.2.2 Level II LR Receiver Characteristics

The key receiver characteristics are summarized in Table 10-17 while the following sections fully detail all the requirements.

Table 10-17. Level II LR Receiver Electrical Input Specifications

Characteristic	Symbol	Condition	Min	Typ	Max	Units
Rx Baud Rate (5Gbaud)	R_Baud	Section 10.5.2.1.2	5.00 -0.01%	5.00	5.00 +0.01%	Gbaud
Rx Baud Rate (6.25Gbaud)			6.25 -0.01%	6.25	6.25 +0.01%	Gbaud
Absolute Input Voltage	R_Vin	Section 10.5.2.2.4				
Input Differential voltage	R_Vdiff	Section 10.5.2.2.3			1200	mVppd
Differential Resistance	R_Rdin	Section 10.5.2.2.7	80	100	120	Ω
Bias Voltage Source Impedance (load type 1)[1]	R_Zvtt				30	Ω
Differential Input Return Loss (100MHz to 0.5*R_Baud)	R_SDD11	Section 10.5.2.2.7			-8	dB
Differential Input Return Loss (0.5*R_Baud to R_Baud))						
Common mode Input Return Loss (100MHz to 0.5*R_Baud)	R_SCC11	Section 10.5.2.2.7			-6	dB
Input Common Mode Voltage	R_Vfcm	Load Type 0[2]	0		1800	mV
		Load Type 1[1,3]	595		R_Vtt - 60	mV
Wander divider (in Figure 8-8 & Figure 8-9)	n			10		

NOTES:
1. DC Coupling compliance is optional (Load Type 1). Only receivers that support DC coupling are required to meet this parameter.
2. Load Type 0 with min T_Vdiff, AC-Coupling or floating load. For floating load, input resistance must be ≥ 1kΩ
3. For Load Type 1: T_Vtt & R_Vtt = 1.2V +5%/-8%.

10.5.2.2.1 Level II LR Receiver Input Baud Rate

All devices shall work at 5Gbaud, 6.25Gbaud or both baud rates with the baud rate tolerance as per Section 8.5.12.

10.5.2.2.2 Level II LR Receiver Reference Input Signals

Reference input signals to the receiver have the characteristics determined by the compliant transmitter. The reference input signal must satisfy the transmitter near-end template and jitter given in Figure 8-2 and Table 10-16, as well as the far-end eye jitter given in Table 10-20, with the differential load impedance of 100Ω ± 1% at DC with a return loss of better than 20dB from baud rate divided by 1667 to 1.5 times the baud rate. Note that the input signal might not meet either of these requirements when the actual receiver replaces this load.

10.5.2.2.3 Level II LR Receiver Input Signal Amplitude

The receiver shall accept differential input signal amplitudes produced by compliant transmitters connected without attenuation to the receiver. This may be larger than the 1200mVppd maximum of the transmitter due to output/input impedances and reflections.

The minimum input amplitude is defined by the far-end transmitter template, the actual receiver input impedance, and the loss of the actual PCB. Note that the far-end transmitter template is defined using a well controlled load impedance, however the real receiver is not, which can leave the receiver input signal smaller than expected.

10.5.2.2.4 Level II LR Receiver Absolute Input Voltage

The absolute voltage levels with respect to the receiver ground at the input of the receiver are dependent on the transmitter implementation and the inter-ground difference.

The voltage levels at the input of an AC coupled receiver (if the effective AC coupling is done within the receiver) or at the Tx side of the external AC coupling cap (if AC coupling is done externally) shall be between -0.15 and 1.95V, inclusive, with respect to local ground.

10.5.2.2.5 Level II LR Receiver Input Common Mode Impedance

The input common mode impedance (R_Zvtt) at the input of the receiver is dependent on whether the receiver is AC or DC coupled. The value of R_Zvtt as measured at the input of an AC coupled receiver is undefined. The value of R_Zvtt as measured at the input of a DC coupled receiver is defined as per Table 10-17.

If AC coupling is used it is to be considered part of the receiver for the purposes of this specification unless explicitly stated otherwise. It should be noted that various methods for AC coupling are allowed (for example, internal to the chip or done externally). See Section 8.5.13, "Termination and DC Blocking" for more information.

10.5.2.2.6 Level II LR Receiver Input Lane-to-Lane Skew

Lane-to-lane skew at the input to the receiver shall not exceed 70UI peak. See Section 8.5.9, "Receiver Input Lane-to-Lane Skew".

10.5.2.2.7 Level II LR Receiver Input Resistance and Return Loss

Refer to Section 8.5.11, "Differential Resistance and Return Loss, Transmitter and Receiver" for the reference model for return loss. See Table 10-18 for 5Gbaud and 6.25Gbaud short run receiver parameters. Definitions for these parameters are in Figure 8-12.

Table 10-18. Level II LR Input Return Loss Parameters

Parameter	Value	Units
A0	-8	dB
f0	100	MHz
f1	R_Baud/2	Hz
f2	R_Baud	Hz
Slope	16.6	dB/dec

10.5.2.2.8 Level II LR Receiver Jitter Tolerance

As per Section 10.5.1.3, "Level II LR Receiver Inter-operability", the receiver shall tolerate at least the far-end jitter requirements as given in Table 10-12 in combination with any compliant channel, as per Section 10.5.1.1, "Level II LR Channel Compliance", with an additional SJ with any frequency and amplitude defined by the mask of Figure 8-9 where the minimum and maximum total wander amplitude are 0.05UIpp and 5UIpp respectively. This additional SJ component is intended to ensure margin for wander, hence is over and above any high frequency jitter from Table 10-12.

10.5.3 Level II LR Link and Jitter Budgets

The primarily intended application is as a point-to-point interface of up to approximately 100cm and up to two connector between integrated circuits using controlled impedance traces on low-cost printed circuit boards (PCBs). Informative loss and jitter budgets are presented in Table 10-19 to demonstrate the feasibility of legacy FR4 epoxy PCB's. The jitter budget is given in Table 10-20. The performance of an actual transceiver interconnect is highly dependent on the implementation.

Table 10-19. Level II LR Informative Loss, Skew and Jitter Budget

Description	Loss (dB)	Differential Skew (ps)	Bounded High Probability (UIpp)	TJ (UIpp)
Transmitter	0	15	0.15	0.30
Interconnect (with Connector)	15.9	25	0.35	0.513
Other	4.5		0.10	0.262
Total	20.4	40	0.60	0.875

Table 10-20. Level II LR High Frequency Jitter Budget

CEI-6G-LR	Uncorrelated Jitter		Correlated Jitter		Total Jitter				Amplitude	
	Unbounded Gaussian	High Probability	Bounded Gaussian	Bounded High Probability	Gaussian	Sinusoidal	Bounded High Probability	Total		
Abbreviation	UUGJ	UHPJ	CBGJ	CBHPJ	GJ	SJ	HPJ	TJ	k	
Units	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp		mVppd
Transmitter	0.150	0.150			0.150		0.150	0.300		800.0
Channel			0.230	0.525						
Receiver Input	0.150	0.150	0.230	0.525	0.275		0.675	0.950	0.00	0.0[2]
Equalizer				-0.350[1]						
Post Equalization	0.150	0.150	0.230	0.175	0.275		0.325	0.60	0.20	100.0
DFE Penalties				0.100					-0.08	-45.0
Clock + Sampler	0.150	0.100		0.100						-45.0
Budget	0.212	0.250	0.230	0.375	0.313	0.050	0.625	0.988	0.06	10.0

NOTES:
1. Due to receiver equalization, it reduces the ISI as seen inside the receiver. Thus this number is negative.
2. It is assumed that the eye is closed at the receiver, hence receiver equalization is required as indicated below.

10.5.4 Level II LR StatEye.org Template

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% example template for setting up a standard, i.e. equalizer
% jitter and return loss

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

param.version = [param.version '_v1.0'];

% these are internal variables and should not be changed

param.scanResolution                = 0.01;
param.binsize                       = 0.0005;
param.points                        = 2^13;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the transmitter and baud rate. The tx filter has two
% parameters defined for the corner frequency of the poles

param.bps                           = 6.25e9;
param.bitResolution                 = 1/(4*param.bps);
param.txFilter                      = 'twopole';
param.txFilterParam                 = [0.75 0.75];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the return loss up. The return loss can be turned off
% using the appropriate option

param.returnLoss                    = 'on';
param.cpad                          = 1.00;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the transmitter emphasis up. Some example setting are
% included which can be uncommented

% single tap emphasis
param.txpre                         = [-0.1];
param.signal                        = 1.0;
param.txpost                        = [];
param.vstart                        = [-0.3 -0.3];
param.vend                          = [+0.0 +0.0];
param.vstep                         = [0.1 0.05 0.025];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the de-emphasis of 4-point transmit pulse
```

```
% the de-emphasis run if param.txpre = [] and param.txpost = []

param.txdeemphasis = [1 1 1 1];            % de-emphasis is off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the data coding changing the transmit pulse spectrum
% the coding run if param.txpre = [] and param.txpost = []

param.datacoding = 1;        % the coding is off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set PAM amplitude and rate

param.PAM = 2;              % PAM is switched off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the rxsample point does not need to be changed as it is
% automatically adjusted by the optimization scripts.
% The number of DFE taps should be set, however, the initial
% conditions are irrelevant.

param.rxsample                      = -0.1;

param.dfe                           = [0.3 0.1 0.1 0.1 0.1];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% sampling jitter in HPJpp and GJrms is defined here

param.txdj                          = 0.15;
param.txrj                          = 0.15/(2*7.94);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the following options are not yet implemented and should
% not be changed

param.user                          = [0.0];
param.useuser                       = 'no';
param.usesymbol                     = '';
param.xtAmp                         = 1.0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

param.TransmitAmplitude = 0.800; % mVppdif
param.MinEye            = 0.100; % mVppdif

param.Q                 = 2*7.94;
param.maxDJ             = 0.325;
```

```
param.maxTJ             = 0.60;
```

10.6 Level II Medium Run Interface General Requirements

10.6.1 Medium Run Jitter and Inter-operability Methodology

The LP-Serial 5Gbaud and 6.25Gbaud short run interfaces use Method C, described in CEI clause 2.4. This section defines the inter-operability methodology specifically for interfaces where transmit emphasis may be used and the receiver eye requires linear equalization (from channel inter-operability point of view) to be open to within the BER of interest.

10.6.1.1 Level II Medium Run Channel Compliance

The following steps shall be made to identify which channels are to be considered compliant:

1. The forward channel and significant crosstalk channels shall be measured using a network analyzer for the specified baud rate (see CEI Section 10.7.4.5, "Network Analysis Measurement" for a suggested method).
2. A single pre or post tap transmitter with <= 6dB of emphasis, with infinite precision accuracy.
3. A Tx edge rate filter: simple 40dB/dec low pass at 75% of baud rate, this is to emulate both Rx and Tx -3dB bandwidths at $^{3}/_{4}$ baud rate.
4. A transmit amplitude of 800mVppd.
5. Additional Uncorrelated Bounded High Probability Jitter of 0.15UIpp (emulating part of the Tx jitter).
6. Additional Uncorrelated Unbounded Gaussian Jitter of 0.15UIpp (emulating part of the Tx jitter).
7. The reference transmitter shall use the worst case transmitter return loss at the baud frequency. In order to construct the worse case transmitter return loss, the reference transmitter should be considered to be a parallel R and C, where R is the defined maximum allowed DC resistance of the interface and C is increased until the defined maximum Return Loss at the baud frequency is reached. The transmitter return loss is specified in Section 10.6.2.1.6, "Level II MR Transmitter Output Resistance and Return Loss".
8. An ideal receiver filter of the form in Section 8.6.8, "Time Continuous Zero/Pole". The reference receiver uses a continuous-time equalizer with 1 zero and 1 pole in the region of baudrate/100 to baudrate. Additional parasitic zeros and poles must be considered part of the receiver vendor’s device and

be dealt with as they are for the reference receiver. Pole and Zero values have infinite precision accuracy. Maximum required gain/attenuation shall be less than or equal to 4dB.

9. The reference receiver shall use the worst case receiver return loss at the baud frequency. In order to construct the worse case receiver return loss, the reference receiver should be considered to be a parallel R and C, where R is the defined maximum allowed DC resistance of the interface and C is increased until the defined maximum Return Loss at the baud frequency is reached. The receiver return loss is specified in Section 10.6.2.2.7, "Level II MR Receiver Input Resistance and Return Loss".

Table 10-21. Level II LR Receiver Equalization Output Eye Mask

Parameter	Symbol	Max	Units
Eye mask	R_X1	0.30	UI
Eye mask	R_Y1	50	mV
Bounded High Probability Jitter	R_BHPJ	0.325	UI

10. Any parameters that have degrees of freedom (e.g. filter coefficients or sampling point) shall be optimized against the amplitude, at the zero phase offset, as generated by the Statistical Eye Output, e.g. by sweeping all degrees of freedom and selecting the parameters giving the maximum amplitude. A receiver return loss, as defined by the reference receiver, shall be used.
11. The opening of the eye shall be calculated using Statistical Eye Analysis methods, as per Section 8.7.5, "Statistical Eye Methodology", and confirmed to be within the requirements of the equalized eye mask as specified in Table 10-12 at the required BER, 10^{-12}.

10.6.1.2 Level II MR Transmitter Inter-operability

The following step shall be made to identify which transmitters are to be considered compliant:

1. It shall be verified that the measured eye is equal or better than the calculated eye for the given measurement probability Q (see Annex B.3, “Eye Mask Adjustment for Sampling Oscilloscopes" for a suggested method of calculating Q given a measurement population), given:
   - A “compliance” channel as per Section 10.6.1.1, "Level II Medium Run Channel Compliance" that required at least half the maximum transmit emphasis with no receiver filtering to give an open eye.
   - Using this channel the transmitter shall be then optimally adjusted and the resulting near-end eye measured (see Section 10.7.4.6, "Eye Mask Measurement Setup" for a suggested method).
   - Using this channel the statistical eye shall then be calculated, as per Section 8.7.5, "Statistical Eye Methodology", using the maximum defined transmit jitter and the actual transmitter's amplitude and emphasis.

If the transmit jitter or transmit eye mask is additionally defined then the following steps shall also be made to identify which transmitters are to be considered compliant:

1. The high frequency transmit jitter shall be within that specified (see Section 10.7.1, "High Frequency Transmit Jitter Measurement" for suggested methods).

The specified transmit eye mask shall not be violated (see Section 10.7.4.6, "Eye Mask Measurement Setup" for a suggested method) after adjusting the horizontal time positions for the measured time with a confidence level of 3 sigma (see Annex B.3, “Eye Mask Adjustment for Sampling Oscilloscopes" for a suggested method).

10.6.1.3 Medium Receiver Inter-operability

The following step shall be made to identify which receivers are to be considered compliant:

1. The DUT shall be measured to have a BER[1] better than specified for a stressed signal (see Section 10.7.4.3, "Jitter Tolerance with Defined ISI and no Relative Wander" for a suggested method) with a confidence level of three sigma (see Annex B.2, “Confidence Level of Errors Measurement" for a suggested method), given:
   - The defined sinusoidal jitter mask for relative wander as per Section 8.4.5, "Total Wander Mask" with a high frequency relative wander and a maximum relative wander as defined in Section 10.5.2.2.8, "Level II LR Receiver Jitter Tolerance".
   - The specified amount of High Probability Jitter and Gaussian jitter as defined in Section 10.5.2.2.8, "Level II LR Receiver Jitter Tolerance".
   - A compliance channel or filter as identified by Section 10.5.1.1, "Level II LR Channel Compliance".
   - An additive crosstalk signal of amplitude such that the resulting statistical eye, given the channel, jitter, and crosstalk, is as close as feasible in amplitude when compared to the defined minimum amplitude for channel compliance.

10.6.2 Level II MR Interface Electrical Characteristics

The electrical interface is based on high speed low voltage logic with nominal differential impedance of 100Ω. Connections are point-to-point balanced differential pair and signalling is unidirectional.

10.6.2.1 Level II MR Transmitter Characteristics

The key transmitter characteristics are summarized in Table 10-22 and Table 10-23 while the following sections fully detail all the requirements.

[1] if the defined measurement BER is different to system required BER, adjustments to applied stressed eye TJ are necessary

Table 10-22. Level II MR Transmitter Output Electrical Specifications

Characteristics	Symbols	Conditions	Min	Typ	Max	Units
Tx Baud Rate (5Gbaud)	T_Baud	Section 10.6.2.1.2	5.00 -0.01%	5.00	5.00 +0.01%	Gbaud
Tx Baud Rate (6.25Gbaud)			6.25 -0.01%	6.25	6.25 +0.01%	Gbaud
Absolute Output Voltage	V_O	Section 10.6.2.1.3	-0.40		2.30	Volts
Output Differential voltage (into floating load Rload=100Ω)	T_Vdiff	Section 10.6.2.1.3[1]	800		1200	mVppd
Differential Resistance	T_Rd	Section 10.6.2.1.6	80	100	120	Ω
Recommended output rise and fall times (20% to 80%)	T_tr, T_tf	Section 10.6.2.1.4	30			ps
Skew between signals comprising a differential pair	T_SKEW_{diff}	Section 10.6.2.1.5			15	ps
Differential Output Return Loss (100 MHz to 0.5*T_Baud)	T_SDD22	Section 10.6.2.1.6			-8	dB
Differential Output Return Loss (0.5*T_Baud to T_Baud)						
Common Mode Return Loss (100MHz to 0.75 *T_Baud)	T_S11	Section 10.6.2.1.6			-6	dB
Transmitter Common Mode Noise	T_Ncm				5% of T_Vdiff	mVppd
Output Common Mode Voltage	T_Vcm	Load Type 0[2] Section 8.5.3	100		1700	mV
		Load Type 1[3,4] Section 8.5.3	630		1100	mV

NOTES:
1. The Transmitter must be capable of producing a minimum T_Vdiff greater than or equal to 800mVppd. In applications where the channel is better than the worst case allowed, a Transmitter device may be provisioned to produce T_Vdiff less than this minimum value, but greater than or equal to 400mVppd, and is still compliant with this specification.
2. Load Type 0 with min T_Vdiff, AC-Coupling or floating load.
3. For Load Type 1: R_Zvtt ≤ 30 Ω; T_Vtt & R_Vtt = 1.2V +5%/-8%
4. DC Coupling compliance is optional (Load Type 1). Only Transmitters that support DC coupling are required to meet this parameter.

Table 10-23. Level II MR Transmitter Output Jitter Specifications

Characteristics	Symbol	Conditions	Min	Typ	Max	Units
Uncorrelated High Probability Jitter	T_UHPJ	Section 10.6.2.2.8			0.15	UIpp
Duty Cycle Distortion	T_DCD	Section 10.6.2.2.8			0.05	UIpp
Total Jitter	T_TJ	Section 10.6.2.2.8			0.30	UIpp
Eye Mask	T_X1	Section 10.6.2.2.8			0.15	UI
Eye Mask	T_X2	Section 10.6.2.2.8			0.40	UI
Eye Mask	T_Y1	Section 10.6.2.2.8	200			mV
Eye Mask	T_Y2	Section 10.6.2.2.8			600	mV

10.6.2.1.1 Level II MR Transmitter Test Load

All transmitter characteristics should be implemented and measured to a differential impedance of 100Ω ±1% at DC with a return loss of better than 20dB from baud rate divided by 1667 to 1.5 times the baud rate, unless otherwise noted.

10.6.2.1.2 Level II MR Transmitter Baud Rate

The baud rates are 5Gbaud and 6.25Gbaud with a tolerance of ±100ppm.

10.6.2.1.3 Level II MR Transmitter Amplitude and Swing

Transmitter differential output amplitude shall be able to drive between 800 and 1200mVppd, inclusive, either with or without any transmit emphasis. DC referenced logic levels are not defined since the receiver must have high common mode impedance at DC. However, absolute transmitter output voltage shall be between -0.1V and 1.9V, inclusive, with respect to local ground. See Figure 8-1 for an illustration of absolute transmitter output voltage limits and definition of differential peak-to-peak amplitude.

10.6.2.1.4 Level II MR Transmitter Rise and Fall Times

The recommended minimum differential rise and fall time is 30ps as measured between the 20% and 80% of the maximum measured levels; the maximum differential rise and fall times are defined by the Tx eye diagram (Figure 8-2 and Table 10-16). Shorter rise and falls may result in excessive high frequency components and increase EMI and cross talk.

10.6.2.1.5 Level II MR Transmitter Differential Pair Skew

The timing skew at the output of a Level II MR transmitter between the two signals that comprise a differential pair shall not exceed 15 ps at 5.0 Gbaud and 6.25 Gbaud.

10.6.2.1.6 Level II MR Transmitter Output Resistance and Return Loss

Refer to Section 8.5.11, "Differential Resistance and Return Loss, Transmitter and Receiver" for the reference model for return loss. See Table 10-15 for 5Gbaud and 6.25Gbaud long run transmitter parameters. Definitions for these parameters are in Figure 8-12.

Table 10-24. Level II MR Transmitter Return Loss Parameters

Parameter	Value	Units
A0	-8	dB
f0	100	MHz
f1	T_Baud/2	Hz
f2	R_Baud	Hz
Slope	16.6	dB/dec

10.6.2.1.7 Level II MR Transmitter Lane-to-Lane Skew

The electrical level of lane-to-lane skew caused by the transmitter circuitry and associated routing must be less than 1000 ps for links of 4 lanes or less. Links with greater than 4 lanes must have lane-to-lane skew of less than 2UI + 1000 ps. The transmitter lane-to-lane skew is only for the serdes Tx and does not include any effects of the channel.

10.6.2.1.8 Level II MR Transmitter Short Circuit Current

The max DC current into or out of the transmitter pins when either shorted to each other or to ground shall be ±100mA when the device is fully powered up. From a hot swap point of view, the ±100mA limit is only valid after 10 μs.

10.6.2.1.9 Level II MR Transmitter Template and Jitter

The transmitter shall satisfy both the near-end eye template and jitter requirements as given in Figure 8-2, Figure 8-3, and Table 10-16 either with or without any transmit emphasis.

The maximum near-end duty cycle distortion (T_DCD) shall be less than 0.05UIpp.

It should be noted that it is assumed the Uncorrelated High Probability Jitter component of the transmitter jitter is not Inter-symbol Interference (ISI). This is only assumed from a receiver point of view so that a receiver can't equalize it and does not in any way put any restrictions on the real transmitter HPJ.

Table 10-25. Level II MR Near-End Template Intervals

Characteristics	Symbol	Near-End Value	Units	Comments
Eye Mask	T_X1	0.15	UI	
Eye Mask	T_X2	0.40	UI	
Eye Mask	T_Y1	200	mV	For connection to short run Rx
		400		For connection to long run Rx
Eye Mask	T_Y2	375	mV	For connection to short run Rx
		600		For connection to long run Rx
Uncorrelated Bounded High Probability Jitter	T_UBHPJ	0.15	UIpp	
Duty Cycle Distortion	T_DCD	0.05	UIpp	
Total Jitter	T_TJ	0.30	UIpp	

10.6.2.2 Level II MR Receiver Characteristics

The key receiver characteristics are summarized in Table 10-26 while the following sections fully detail all the requirements.

Table 10-26. Level II MR Receiver Electrical Input Specifications

Characteristic	Symbol	Condition	Min	Typ	Max	Units
Rx Baud Rate (5Gbaud)	R_Baud	Section 10.6.2.2.1	5.00 -0.01%	5.00	5.00 +0.01%	Gbaud
Rx Baud Rate (6.25Gbaud)			6.25 -0.01%	6.25	6.25 +0.01%	Gbaud
Absolute Input Voltage	R_Vin	Section 10.6.2.2.4				
Input Differential voltage	R_Vdiff	Section 10.6.2.2.3			1200	mVppd
Differential Resistance	R_Rdin	Section 10.5.2.2.7	80	100	120	Ω
Bias Voltage Source Impedance (load type 1)[1]	R_Zvtt				30	Ω
Differential Input Return Loss (100MHz to 0.5*R_Baud)	R_SDD11	Section 10.6.2.2.7			-8	dB
Differential Input Return Loss (0.5*R_Baud to R_Baud))						
Common mode Input Return Loss (100MHz to 0.5*R_Baud)	R_SCC11	Section 10.6.2.2.7			-6	dB
Input Common Mode Voltage	R_Vfcm	Load Type 0[2]	0		1800	mV
		Load Type 1[1,3]	595		R_Vtt - 60	mV
Wander divider (in Figure 8-8 & Figure 8-9)	n			10		

NOTES:
1. DC Coupling compliance is optional (Load Type 1). Only receivers that support DC coupling are required to meet this parameter.
2. Load Type 0 with min T_Vdiff, AC-Coupling or floating load. For floating load, input resistance must be ≥ 1kΩ
3. For Load Type 1: T_Vtt & R_Vtt = 1.2V +5%/-8%.

10.6.2.2.1 Level II MR Receiver Input Baud Rate

All devices shall work at 5Gbaud, 6.25Gbaud or both baud rates with the baud rate tolerance as per Section 8.5.12, "Baud Rate Tolerance".

10.6.2.2.2 Level II MR Receiver Reference Input Signals

Reference input signals to the receiver have the characteristics determined by the compliant transmitter. The reference input signal must satisfy the transmitter near-end template and jitter given in Figure 8-2, Figure 8-3, and Table 10-16, as well as the far-end eye jitter given in Table 10-20, with the differential load impedance of 100Ω ± 1% at DC with a return loss of better than 20dB from baud rate divided by 1667 to 1.5 times the baud rate. Note that the input signal might not meet either of these requirements when the actual receiver replaces this load.

10.6.2.2.3 Level II MR Receiver Input Signal Amplitude

The receiver shall accept differential input signal amplitudes produced by compliant transmitters connected without attenuation to the receiver. This may be larger than the 1200mVppd maximum of the transmitter due to output/input impedances and reflections.

The minimum input amplitude is defined by the far-end transmitter template, the actual receiver input impedance, and the loss of the actual PCB. Note that the far-end transmitter template is defined using a well controlled load impedance, however the real receiver is not, which can leave the receiver input signal smaller than expected.

10.6.2.2.4 Level II MR Receiver Absolute Input Voltage

The absolute voltage levels with respect to the receiver ground at the input of the receiver are dependent on the transmitter implementation and the inter-ground difference.

The voltage levels at the input of an AC coupled receiver (if the effective AC coupling is done within the receiver) or at the Tx side of the external AC coupling cap (if AC coupling is done externally) shall be between -0.15 and 1.95V, inclusive, with respect to local ground.

10.6.2.2.5 Level II MR Receiver Input Common Mode Impedance

The input common mode impedance (R_Zvtt) at the input of the receiver is dependent on whether the receiver is AC or DC coupled. The value of R_Zvtt as measured at the input of an AC coupled receiver is undefined. The value of R_Zvtt as measured at the input of a DC coupled receiver is defined as per Table 10-17.

If AC coupling is used it is to be considered part of the receiver for the purposes of this specification unless explicitly stated otherwise. It should be noted that various methods for AC coupling are allowed (for example, internal to the chip or done externally). See Section 8.5.13, "Termination and DC Blocking" for more information.

10.6.2.2.6 Level II MR Receiver Input Lane-to-Lane Skew

Lane-to-lane skew at the input to the receiver shall not exceed 70UI peak. See Section 8.5.9, "Receiver Input Lane-to-Lane Skew".

10.6.2.2.7 Level II MR Receiver Input Resistance and Return Loss

Refer to Section 8.5.11, "Differential Resistance and Return Loss, Transmitter and Receiver" for the reference model for return loss. See Table 10-27 for 5Gbaud and 6.25Gbaud short run receiver parameters. Definitions for these parameters are in Figure 8-12.

Table 10-27. Level II MR Input Return Loss Parameters

Parameter	Value	Units
A0	-8	dB
f0	100	MHz
f1	R_Baud/2	Hz
f2	R_Baud	Hz
Slope	16.6	dB/dec

10.6.2.2.8 Level II MR Receiver Jitter Tolerance

As per Section 10.5.1.3, "Level II LR Receiver Inter-operability", the receiver shall tolerate at least the far-end jitter requirements as given in Table 10-12 in combination with any compliant channel, as per Section 10.5.1.1, "Level II LR Channel Compliance", with an additional SJ with any frequency and amplitude defined by the mask of Figure 8-8 where the minimum and maximum total wander amplitude are 0.05UIpp and 5UIpp respectively. This additional SJ component is intended to ensure margin for wander, hence is over and above any high frequency jitter from Table 10-12.

10.6.3 Level II MR Link and Jitter Budgets

The primarily intended application is as a point-to-point interface of up to approximately 60cm and up to two connector between integrated circuits using controlled impedance traces on low-cost printed circuit boards (PCBs). Informative loss and jitter budgets are presented in Table 10-19 to demonstrate the feasibility of legacy FR-4 epoxy PCB's. The jitter budget is given in Table 10-20. The performance of an actual transceiver interconnect is highly dependent on the implementation.

Table 10-28. Level II MR Informative Loss, Skew and Jitter Budget

Description	Loss (dB)	Differential Skew (ps)	Bounded High Probability (UIpp)	TJ (UIpp)
Transmitter	0	15	0.15	0.30
Interconnect (with Connector)	15.9	25	0.35	0.513
Other	4.5		0.10	0.262
Total	20.4	40	0.60	0.875

Table 10-29. Level II MR High Frequency Jitter Budget

CEI-6G-LR	Uncorrelated Jitter		Correlated Jitter		Total Jitter				Amplitude	
	Unbounded Gaussian	High Probability	Bounded Gaussian	Bounded High Probability	Gaussian	Sinusoidal	Bounded High Probability	Total		
Abbreviation	UUGJ	UHPJ	CBGJ	CBHPJ	GJ	SJ	HPJ	TJ	k	
Units	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp	UIpp		mVppd
Transmitter	0.150	0.150			0.150		0.150	0.300		800.0
Channel			0.230	0.525						
Receiver Input	0.150	0.150	0.230	0.525	0.275		0.675	0.950	0.00	0.0[2]
Equalizer				-0.350[1]						
Post Equalization	0.150	0.150	0.230	0.175	0.275		0.325	0.60	0.20	100.0
DFE Penalties				0.100					-0.08	-45.0
Clock + Sampler	0.150	0.100		0.100						-45.0
Budget	0.212	0.250	0.230	0.375	0.313	0.050	0.625	0.988	0.06	10.0

NOTES:
1. Due to receiver equalization, it reduces the ISI as seen inside the receiver. Thus this number is negative.
2. It is assumed that the eye is closed at the receiver, hence receiver equalization is required as indicated below.

10.6.4 Level II MR StatEye.org Template

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% example template for setting up a standard, i.e. equalizer
% jitter and return loss

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

param.version = [param.version '_v1.0'];

% these are internal variables and should not be changed

param.scanResolution                = 0.01;
param.binsize                       = 0.0005;
param.points                        = 2^13;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the transmitter and baud rate. The tx filter has two
% parameters defined for the corner frequency of the poles

param.bps                           = 6.25e9;
param.bitResolution                 = 1/(4*param.bps);
param.txFilter                      = 'twopole';
param.txFilterParam                 = [0.75 0.75];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the return loss up. The return loss can be turned off
% using the appropriate option

param.returnLoss                    = 'on';
param.cpad                          = 1.0;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the transmitter emphasis up. Some example setting are
% included which can be uncommented

% single tap emphasis
param.txpre                         = [];
param.signal                        = 1.0;
param.txpost                        = [-0.1];
param.vstart                        = [-0.3 -0.3];
param.vend                          = [+0.0 +0.0];
param.vstep                         = [0.1 0.05 0.025];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the de-emphasis of 4-point transmit pulse
```

```
% the de-emphasis run if param.txpre = [] and param.txpost = []

param.txdeemphasis = [1 1 1 1];            % de-emphasis is off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set the data coding changing the transmit pulse spectrum
% the coding run if param.txpre = [] and param.txpost = []

param.datacoding = 1;        % the coding is off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% set PAM amplitude and rate

param.PAM = 2;              % PAM is switched off

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the rxsample point does not need to be changed as it is
% automatically adjusted by the optimization scripts.
% The number of DFE taps should be set, however, the initial
% conditions are irrelevant.

param.rxsample                      = -0.1;

% no DFE
param.dfe                           = [];

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% The CTE shall be controlled.

param.cte = 1; % CTE setting “0” = off; “1” = on;
param.ctethresh = 3; % max gain;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% sampling jitter in HPJpp and GJrms is defined here

param.txdj                          = 0.15;
param.txrj                          = 0.15/(2*7.94);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% the following options are not yet implemented and should
% not be changed

param.user                          = [0.0];
param.useuser                       = ’no’;
param.usesymbol                     = ’’;
param.xtAmp                         = 1.0;
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

param.TransmitAmplitude = 0.800; % mVppdif
param.MinEye            = 0.100; % mVppdif

param.Q                 = 2*7.94;
param.maxDJ             = 0.325;
param.maxTJ             = 0.60;
```

10.7 Level II Measurement and Test Requirements

All methodology described in this section is only relevant for verification of low level CDR functionality, and does not cover any required tests for protocol compliance, e.g. deskew. The methodology is based on the assumption that either an integrated BERT is present in the DUT or a loop or functionality for the attachment of external equipment.

10.7.1 High Frequency Transmit Jitter Measurement

The following section describes various methods for measuring high frequency jitter, which depending upon the baud rate can be applied for various levels of accuracy.

10.7.1.1 BERT Implementation

Referring to Figure 10-4, this section describes test methodology based on bathtub extraction, which relies on equipment being available for the given baud rate.

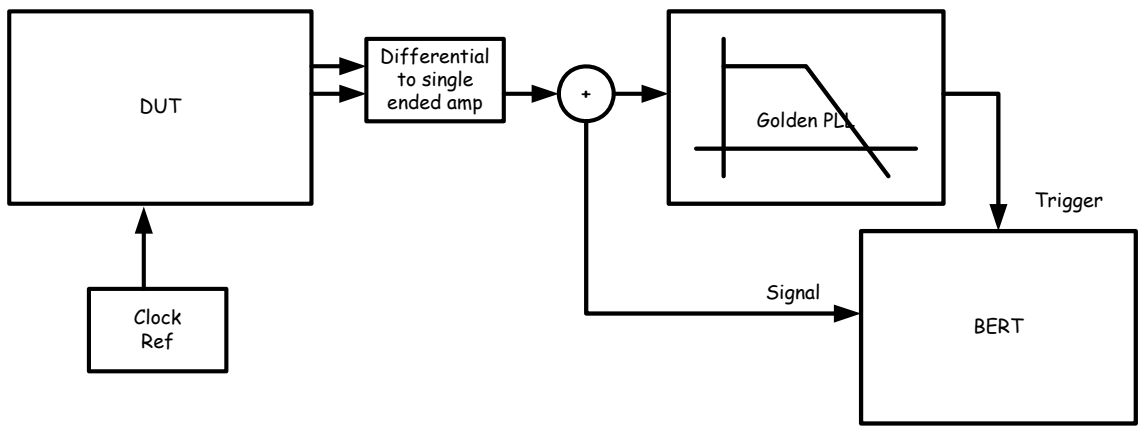


Figure 10-4. BERT with Golden PLL

- This same methodology can be used by equalized transmitters by initially turning the equalization off, or by performing the measurement at the end of a golden channel.

- The transmitter under test shall transmit the specified data pattern, while all other signals are active.
  - The other channels can transmit the same pattern if they have at least a 16 bit offset with the channel under test.
  - All links within a device under test to be active in both transmit and receive directions, and receive links are to use asynchronous clocks with respect to transmit links to maximum allowed ppm offset as specified in the protocol specifications.
- The data should be differentially analyzed using an external differential amp or differential input BERT and golden PLL.
  - Use of single ended signals will give an inaccurate measurement and should not be used.
  - The use of a balun will most likely degrade the signal integrity and is only recommended for 3Gbaud signalling when the balun is linear with a return loss of better than -15dB until three times the baud rate.
- Inherent bandwidth of clock reference inputs of the BERT should be verified, e.g. in the case of parBERTs. Additional bandwidth limitation of the BERT will lead to inaccurate results.
- The use of a golden PLL is required to eliminate inherent clock content (Wander) in transmitted data signals for long measurement periods.
  - The golden PLL should have at maximum a bandwidth of baud rate over 1667, with a maximum of 20dB/dec rolloff, until at least baud rate over 16.67, with no peaking around the corner frequency.
- The output jitter for the DUT is not defined as the contributed jitter from the DUT but as the total output jitter including the contributions from the reference clock. To this end, the reference clock of the DUT should be verified to have a performance similar to the real application.
- A confidence level of three sigma should be guaranteed in the measurement of BER for the Bathtub as per Annex B.2, “Confidence Level of Errors Measurement".[1]
- The High Probability and Gaussian Jitter components should be extracted from the bathtub measurement using the methodology defined in Section 8.7.4.6, "BathTub Curves".
- If not defined the maximum Gaussian jitter is equal to the maximum total jitter minus the actual High Probability jitter.

[1]It is assumed due to the magnitude of jitter present at the transmitter that the left and right hand parts of the bathtub are independent to each other

10.7.1.2 Spectrum Analyzer and Oscilloscope Methodology

10.7.1.2.1 Band Limited[1] Unbounded Gaussian Noise

Referring to Figure 10-5, blandishment or high frequency Gaussian noise can be measured at the transmitter of the DUT accurately using a high frequency 101010 pattern and measuring the spectral power[2]. In Figure 10-5 the clock reference is such that its power noise represents the typical power noise of the reference in the system.

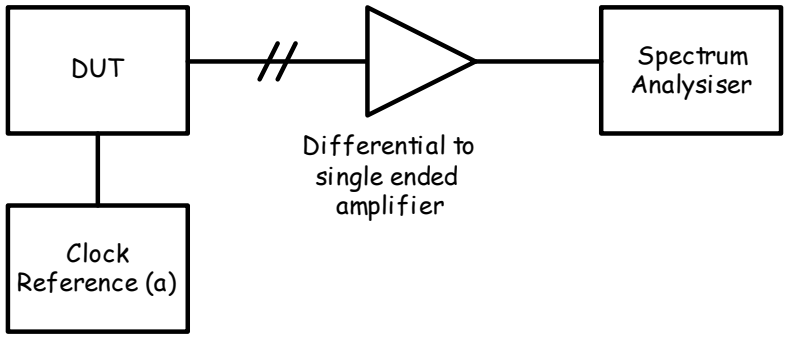


Figure 10-5. Spectral Measurement Setup

The spectral power is calculating by integrating over the frequency band of interest and converting into time jitter.

$$\tau_{rms} = \frac{1}{2\pi}\sqrt{2 \cdot \int_{f_1/100}^{100f_2} \left| \frac{1/f_1 \cdot j \cdot f}{(1 + j \cdot f/f_1)(1 + j \cdot f/f_2)} \right| \cdot 10^{\frac{P(f)}{10}}}$$

where

τ_{rms} is the time jitter

$P(f)$ is the measured spectral power for 1Hz Bandwidth

It should be noted that the measured Gaussian noise for a driver can usually be considered equivalent to that derived from a full bathtub jitter distribution.

10.7.1.2.2 Band Limited 60 Second Total Jitter Measurements

In certain CEI-11G-SR applications total jitter measurements of 60 seconds are required. The Gaussian Jitter, as measured above, should be multiplied by a Q of 6.96[3]. If spurs are present in the spectrum then these must be converted to time jitter

[1]Normal CEI application will integrate from the defined ideal CDR bandwidth to infinity, while some CEI-11G-SR application will integrate over a specific band

[2]The spectral power should be measured using averaging

[3]Traditional measurements are performed for 60 seconds using a demodulator and performing a real time peak to peak measurement of the jitter. Given this, the number of bits transmitter across the link in 60 seconds is calculated and the associated three sigma confidence level, peak to peak multiplication factor, Q, for the random jitter.

separately using an inverse of the Bessel function as per Figure 10-6, which describes the power spectrum for a given phase modulated signal

where

$F(P_n)$ is the inverse spectral SSB power to time modulation (below)

$$\tau_{pkpk} = 2Q\tau_{rms} + \sum_{n} F(P_n)$$

P_n is the relative SSB power of a spur.

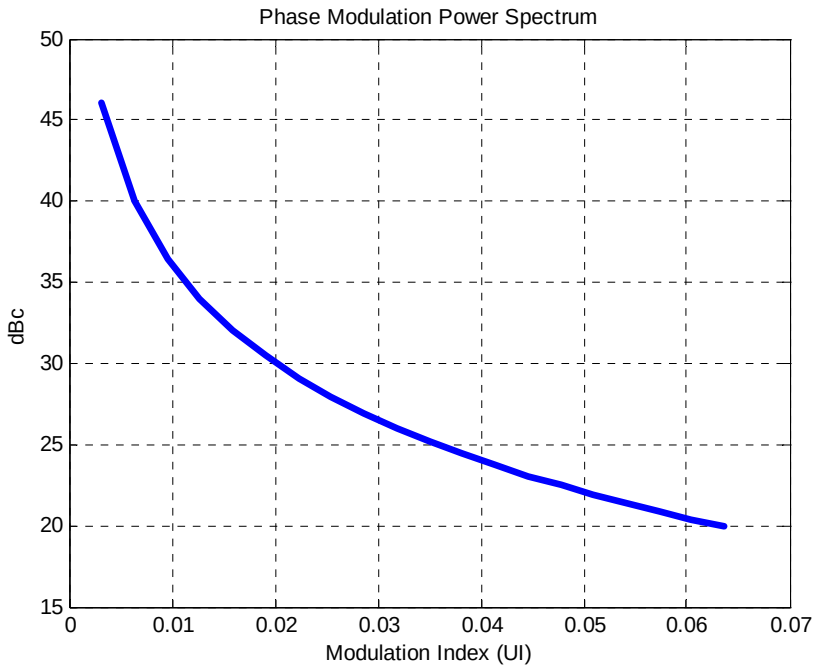


Figure 10-6. Single Side Band Relative Power Spectrum for Phase Modulated Signal

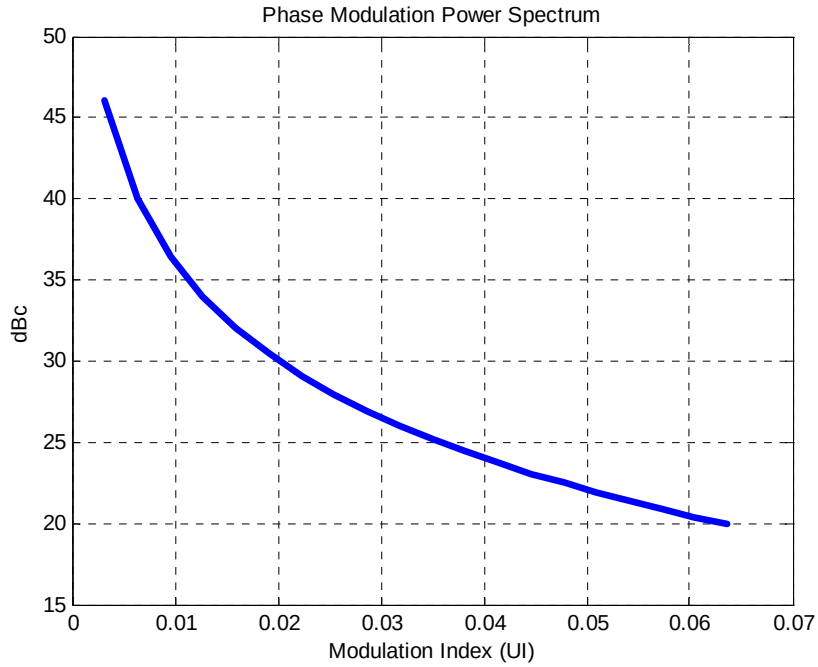


Figure 10-7. Single Side Band Relative Power Spectrum for Phase Modulated Signal

10.7.1.2.3 Uncorrelated High Probability Jitter

After measuring the Gaussian Jitter, as above, an oscilloscope measurement, as per Section 10.7.4.6, "Eye Mask Measurement Setup", of the peak to peak jitter should be performed using a 101010 pattern.

The Uncorrelated High Probability Jitter is then calculated by removing the accumulated Unbounded Gaussian jitter

$$\tau_{UBHJ} = \tau_{pkpk} - 2Q\tau_{rms}$$

using a Q calculated for a 3 sigma confidence level[1] as per Annex B.3, “Eye Mask Adjustment for Sampling Oscilloscopes".

10.7.1.2.4 Total High Probability Jitter

After measuring the Unbounded Gaussian Jitter, as above, an oscilloscope measurement, as per Section 10.7.4.6, "Eye Mask Measurement Setup", of the peak to peak jitter should be performed using the standard pattern e.g. PRBS31.

The Total High Probability Jitter is then calculated by removing the accumulated Gaussian jitter.

[1]It is recommended that enough samples on the oscilloscope should be made such that Q>4

$$\tau_{HPJ} = \tau_{pkpk} - 2Q\tau_{rms}$$

using a Q calculated for a 3 sigma confidence level[1] as per Annex B.3, “Eye Mask Adjustment for Sampling Oscilloscopes".

10.7.2 Total Transmit Wander Measurement

This section describes the total transmit wander of a simple non-equalized transmitter as depicted in Figure 10-8 below.

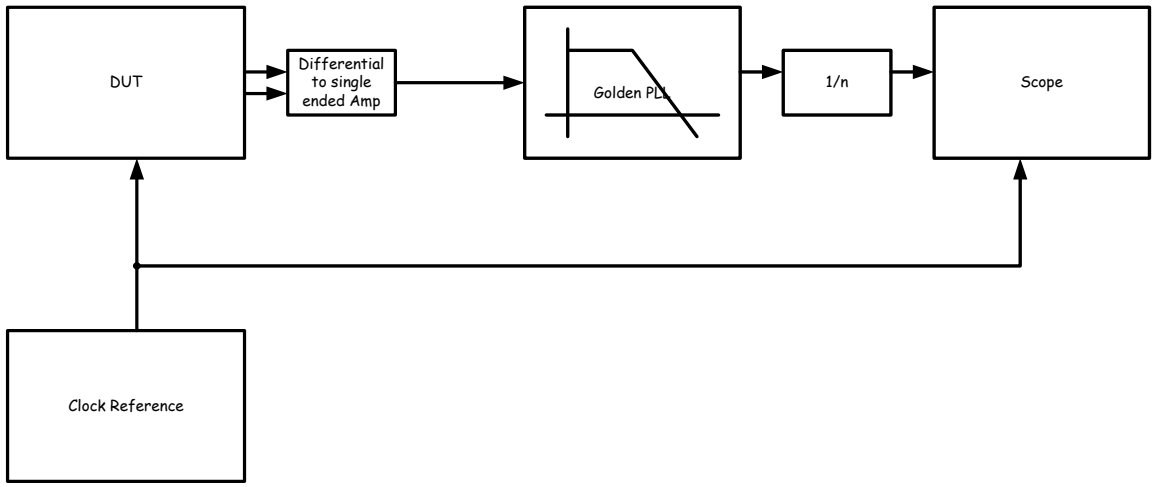


Figure 10-8. Transmit Wander Lab Setup

- The transmitter under test shall transmit the specified data pattern while all other signals are active.
  - The other channels can transmit the same pattern if they have at least a 16 bit offset with the channel under test.
  - All lanes to be active in both transmit and receive directions, and opposite ends of the link, i.e. transmit to receiver, are to use asynchronous clocks to maximum allowed ppm offset as specified in the protocol specifications.
- The transmitter can be tested single ended as high frequency jitter components are filtered by the golden PLL.
- Temperature and supply voltage should be cycled with a rate slower than baud rate over 166700Hz during test to exercise any delay components in the DUT.
- The inherent clock wander in signal shall be extracted using golden PLL and divided by the 1/n block, such as to limit the measured wander to 1UI at the divided frequency, and thus allowing it to be measured on an oscilloscope.
  - The golden PLL should have at a minimum bandwidth of baud rate over 1667, with a maximum of 20dB/dec rolloff, until at least baud rate over 16.67, and is

[1]It is recommended that enough samples on the oscilloscope should be made such that Q>4

suggested to have no peaking around the corner frequency.

- The peak to peak total wander of the extracted clock should be measured using a scope triggered by the reference clock. The measured peak to peak wander should be verified to be bounded by repeating the measurement for ever increasing periods of time until the measurement is constant.

10.7.3 Relative Transmit Wander Measurement

This section describes specifically for SxI-5 interfaces, where limitations are defined in terms of relative wander between data lanes and clocks, whose relative wander can be measured as depicted below.

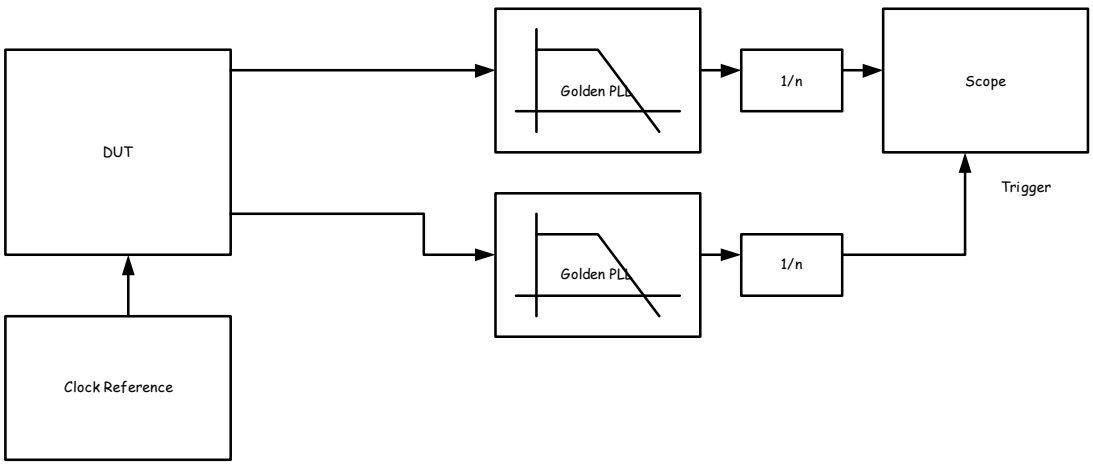


Figure 10-9. Relative Wander Lab Setup

- The transmitter under test shall transmit the specified data pattern while all other signals are active.
  - The other channels can transmit the same pattern if they have at least a 16 bit offset with the channel under test.
  - All lanes to be active in both transmit and receive directions, and opposite ends of the link, i.e. transmit to receiver, are to use asynchronous clocks to maximum allowed ppm. offset as specified in the protocol specifications.
  - The transmitters can be tested single ended as high frequency jitter components are filtered by the golden PLL.
  - Temperature and supply voltage should be cycled with a rate slower than baud rate over 166700Hz during test to exercise any delay components in the DUT.
  - The inherent clock wander in each signal shall be extracted using golden PLL and divided by the 1/n block, such as to limit the measured wander to 1UI at the divided frequency, and thus allowing it to be measured on an oscilloscope.
  - The golden PLL should have at a minimum bandwidth of baud rate over 1667, with a maximum of 20dB/dec rolloff, until at least baud rate over 16.67, and is

suggested to have no peaking around the corner frequency.

- The peak to peak relative wander between the extracted clocks should be measured using a scope triggered by one of the extracted clocks. The measured peak to peak wander should be verified to be bounded by repeating the measurement for ever increasing periods of time until the measurement is constant.

10.7.4 Jitter Tolerance

10.7.4.1 Jitter Tolerance with Relative Wander Lab Setup

The following section describes the required jitter tolerance methodology for devices where Relative Wander is applicable, e.g. SxI.5, and where no receive equalization is implemented.

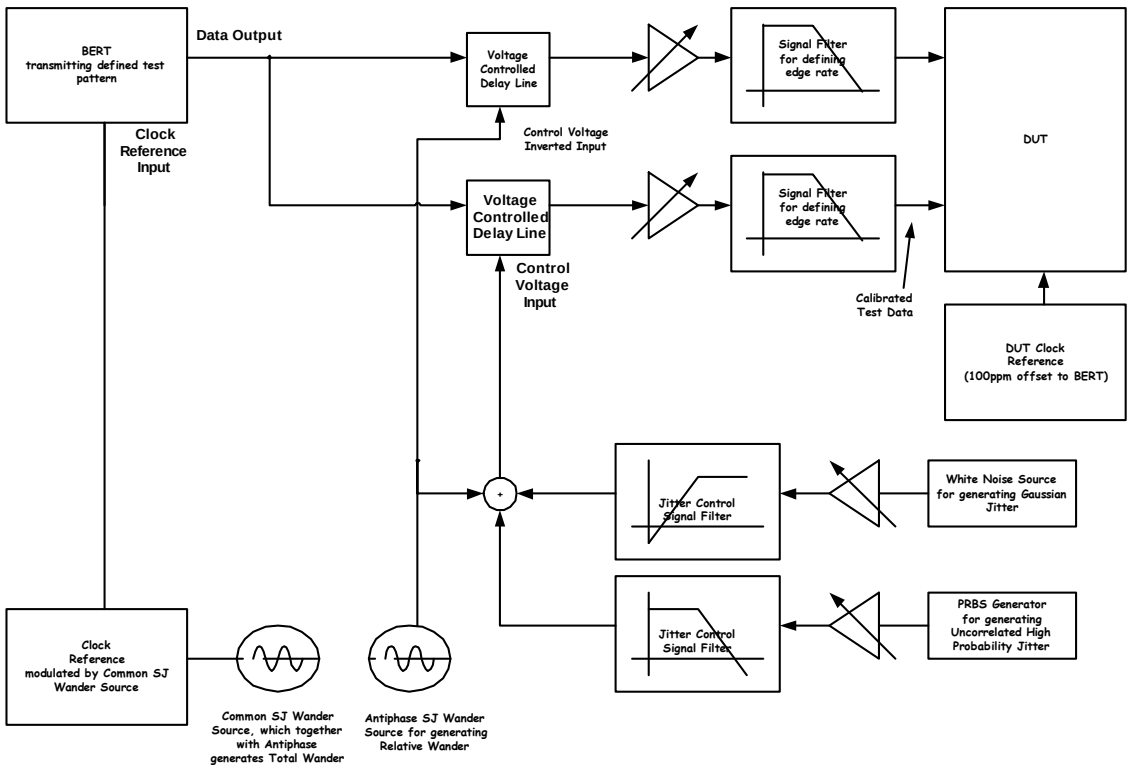


Figure 10-10. Jitter Tolerance with Relative Wander Lab Setup

10.7.4.1.1 General

The transmitter under test shall transmit the specified data pattern while all other signals are active.

– The other channels can transmit the same pattern if they have at least a 16 bit offset with the channel under test.

  - All lanes to be active in both transmit and receive directions, and opposite ends of the link, i.e. transmit to receiver, are to use asynchronous clocks to maximum allowed ppm offset as specified in the protocol specifications.

- The DUT shall be tested using an internal BERT or loop to have the defined BER performance.
- The confidence level of the BER measurement should be at least three sigma as per Annex B.2, “Confidence Level of Errors Measurement".

10.7.4.1.2 Synchronization

- All lanes are to be active in both transmit and receive directions.
- All reference clocks should have the maximum offset frequency, with respect to each other, as defined in the CEI IA.

10.7.4.1.3 Jitter

- The applied calibrated test signal shall have applied a calibrated amount of HF, GJ, and HPJ.
- The jitter control signal for generating High Probability Jitter should be filtered using at least a first order low pass filter with a corner frequency between 1/20 - 1/10 of the baud rate of the PRBS generator to ensure that high frequency components are removed. The distribution of the jitter after the filter must be reasonably even, symmetrical, and large spikes should be avoided. The order of the PRBS polynomial may be between 7 and 11, inclusive, to allow flexibility in meeting this objective. The rate of the PRBS generator should be between 1/10 - 1/3 of the data rate of the DUT, and their rates must be not harmonically related. The upper -3 dB frequency of the filtered HPJ should be at least 1/100 of the data rate of the DUT to represent transmitter jitter that is above the tracking frequencies of the DUT's CDR. Calibration of HPJ must be done with a golden PLL in place. Once these objectives are achieved, there is no need to vary these settings; any combination of settings that meets all the objectives is satisfactory.
- The jitter control signal for generating Unbounded Gaussian Jitter shall be filtered as per Figure 8-10 using the “Jitter Control Signal Filter”. However, the upper frequency of the Gaussian Jitter spectrum will be, acceptably, limited by the bandwidth of the voltage controlled delay line. The crest factor of the white noise generator should be better than 18dB.
- The calibrated test signal shall have a calibrated amount of Total Wander and Relative Wander as compared to the used clock by using the Common SJ Wander and Antiphase SJ Sources with 1% frequency offsets (note the use of the inverted input to the uppermost delay line) as per Section 8.7.2, "Total Wander vs. Relative Wander".
- The amplitude of the Total Wander and Relative Wander is defined by the sinusoidal masks defined in Section 8.4.5, "Total Wander Mask" and Section 8.4.6, "Relative Wander Mask" with the specified amplitudes from the CEI IA.

- Wander should be applied
  - from a frequency equivalent to 1UI of Total Jitter up to 20MHz modulation frequency.
  - at a maximum of 2MHz frequency steps above the corner frequency.
  - at a maximum of 200kHz frequency steps below the corner frequency.

10.7.4.1.4 Amplitude

- The calibrated data signals should be filtered using single pole low pass filter with a corner frequency of 0.7 times the baud rate to define the edge rate.
- The amplitude of the signal should be adjusted such that it just passes the defined receiver data eye sensitivity.
- For testing of DC coupled receivers either a pattern generator capable of generating differential signals and setting the common mode should be used, or a combined AC coupled signal together with a biased-T. Using this setup the common mode should be varied between the defined maximum and minimum.

10.7.4.2 Jitter Tolerance with no Relative Wander Lab Setup

The following section describes the required jitter tolerance methodology for devices where Relative Wander is not applicable and no receive equalization is implemented.

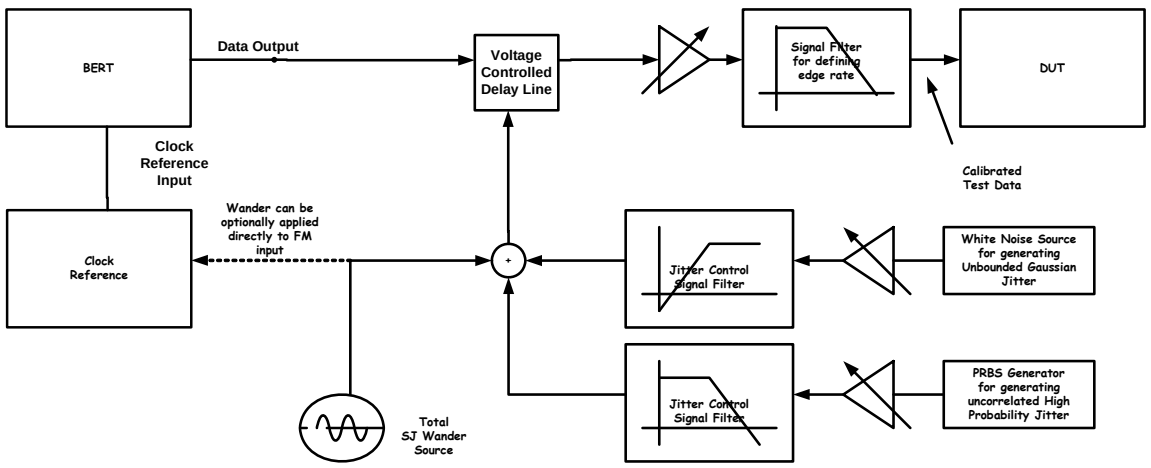


Figure 10-11. Jitter Tolerance with no Relative Wander

Referring to Figure 10-11, the DUT shall be tested as per the description in Section 10.7.4.1, "Jitter Tolerance with Relative Wander Lab Setup", omitting any requirements relating to relative wander and where only Total Wander is applied via the SJ Source shown.

10.7.4.3 Jitter Tolerance with Defined ISI and no Relative Wander

The following section describes the required jitter tolerance methodology for devices where Relative Wander is not applicable, e.g. SxI.5, and where receive equalization is implemented and the performance of the equalization must be verified.

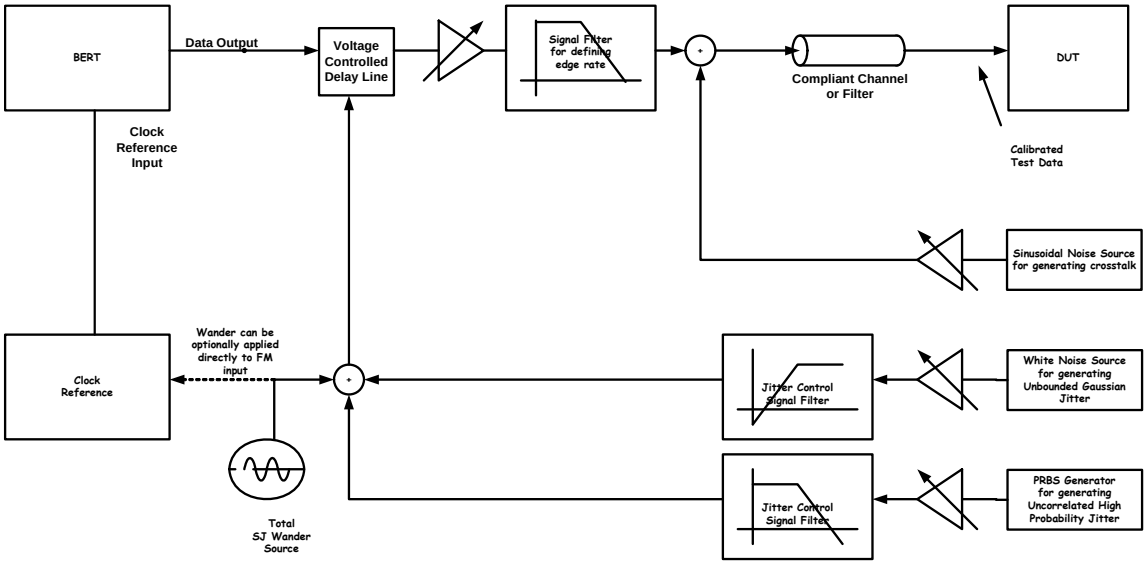


Figure 10-12. Jitter Tolerance with Defined ISI

Referring to Figure 10-12, the DUT shall be tested as per the description in Section 10.7.4.1, "Jitter Tolerance with Relative Wander Lab Setup", omitting any requirements relating to relative wander, and additionally:

- The transmit jitter and amplitude shall be initially calibrated as per Section 10.7.1, "High Frequency Transmit Jitter Measurement" at the output of the delay line.
- A compliance channel shall be added.
- The defined amount of uncorrelated additive noise shall be applied via a sinusoidal source differentially to the signal. The frequency used shall be between 100MHz and the lesser of 1/4 the data rate and 2GHz. There is no need to sweep the frequency.

10.7.4.4 Jitter Transfer

This section describes how jitter transfer relevant interfaces can be tested for compliance:

- The BERT shall generate a data pattern as defined by the CEI IA.
- The jitter present before the delay line should be minimized so as to maximize any transfer bandwidth function of the DUT.

- A sinusoidal jitter should be applied following the same defined SJ mask as used for jitter tolerance and with the same resolution as described in Section 10.7.4, "Jitter Tolerance".

The peak to peak jitter for a 60 second period measured on the scope should be compared before and after the application of the sinusoidal jitter. The ratio of the difference to the jitter applied is then defined as the jitter transfer function.

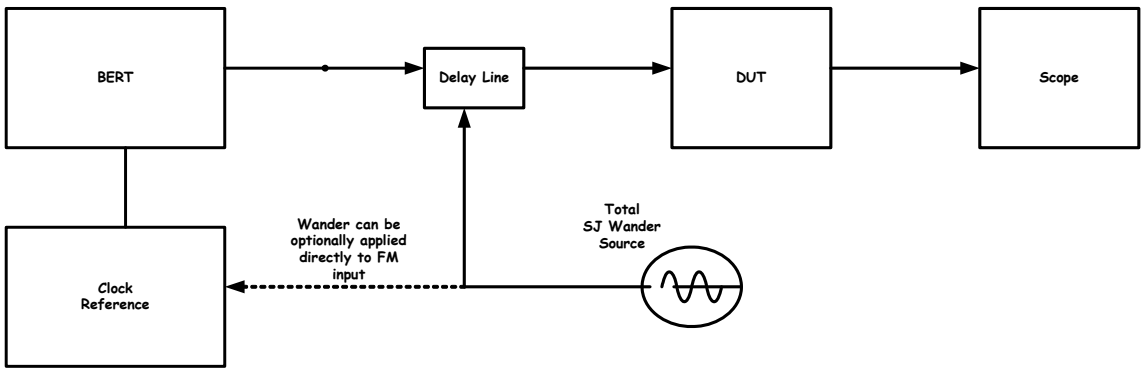


Figure 10-13. Jitter Transfer Lab Setup

10.7.4.5 Network Analysis Measurement

To enable accurate analysis of a channel the following methodology should be followed for the measurement and calculation of the effective channel transfer function.

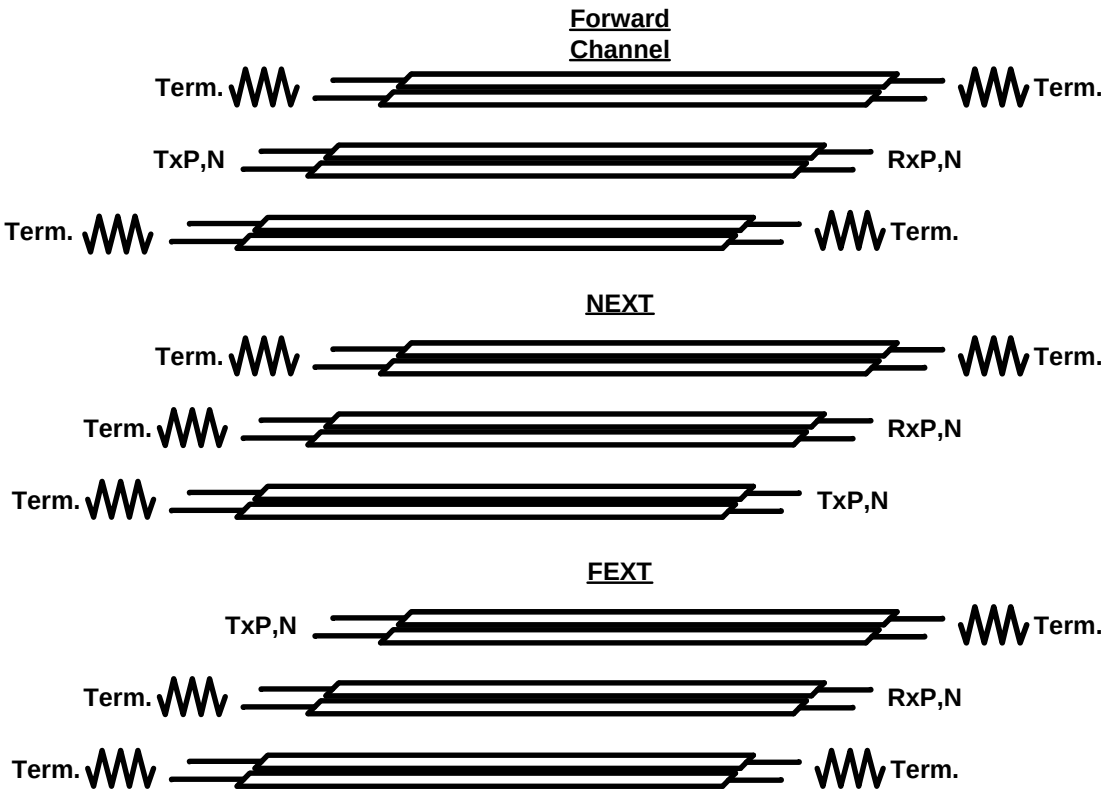


Figure 10-14. S-parameter Port Definitions

- Figure 10-14 shows an overview of the termination and port definitions typically used when measuring the forward channel and NEXT/FEXT crosstalk aggressors.
- The intermediate frequency (IF) bandwidth should be set to a maximum of 300Hz with 100Hz preferred. The launch power shall be specified to the highest available leveled output power not to exceed 0dBm.
- Either direct differential measurements of the channel S21 and S11 should be performed or multiple single ended measurements from which the differential modes can be calculated.[1]
- Linear frequency steps of the measurements shall be no larger than 12.5MHz.
- A frequency range from no higher than 100MHz to no lower than three times the fundamental frequency should be measured.
- Extrapolation towards DC should be performed linearly on magnitude part with the phase being extrapolated to zero at DC, i.e. only a real part is present at DC.

[1]Special care must be taken when performing multiple single ended measurements if the system is tightly coupled

- The channel response of the channel should be calculated by cascading the complete 4 port S-parameter matrix with a worst case transmitter and receiver. The transmitter/receiver should be described as a parallel R and C, where R is the defined maximum allowed DC resistance of the interface and C is increased until the defined maximum Return Loss at the defined frequency is reached.
- Any defined effective transmit or receiver filters should also be cascaded with the channel response.
- The time resolution should be increased by resampling the impulse response in the time domain.
- If required, interpolation of the frequency domain should be performed on the magnitude and unwrapped phase components of the channel response

$$Tr(\omega) = \begin{bmatrix} 1 & 1 \\ 1 & Tx_{22}(\omega) \end{bmatrix} \otimes \begin{bmatrix} S_{11}(\omega) & S_{21}(\omega) \\ S_{12}(\omega) & S_{22}(\omega) \end{bmatrix} \otimes \begin{bmatrix} Rx_{11}(\omega) & 1 \\ 1 & 1 \end{bmatrix}$$

where

$S_{m,n}$ is the measured 4 port differential data of the channel

Tx_{22} is the transmitter return loss

Rx_{11} is the receiver return loss

$Tr(\omega)$ is the receiver return loss.

Converting the original frequency range to time domain, we obtain

$$i(t_m) = ifft(Tr(\omega))$$

where

$$\omega = [-\frac{3}{4}f_{baud}, \frac{3}{4}f_{baud}]$$

10.7.4.6 Eye Mask Measurement Setup

The measurement of an eye mask is defined by the various CEI IAs in terms of a polygon for the probability of the required Bit Error Rate. This polygon may have to be altered given that the sample population of the scope is limited and must be adjusted as per Annex B.3, “Eye Mask Adjustment for Sampling Oscilloscopes". For the measurement of the signal the laboratory setup shown in Figure 10-15 should be used, including the recommendations list in Section 10.7.1, "High Frequency Transmit Jitter Measurement".

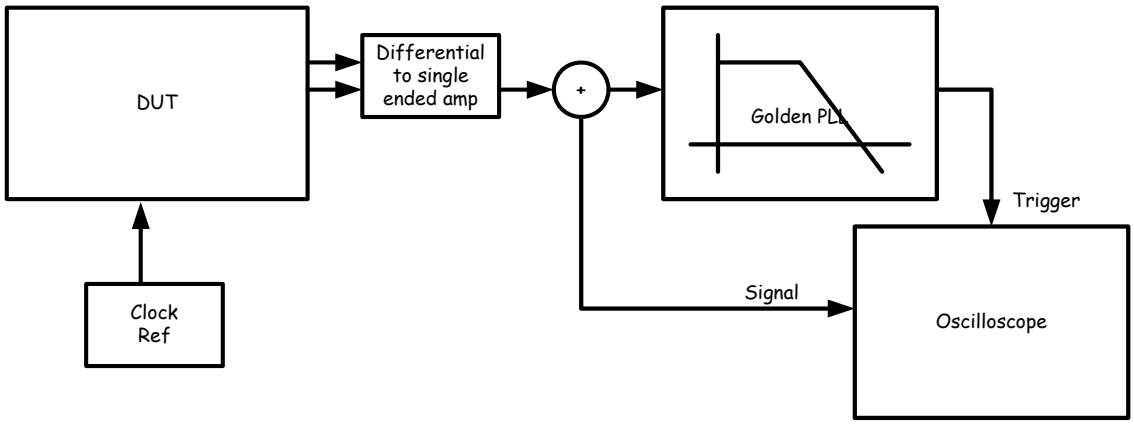


Figure 10-15. Mask Measurement with Golden PLL

Blank page

Annex A Transmission Line Theory and Channel Information (Informative)

A.1 Transmission Lines Theory

The performance of a high frequency transmission line is strongly affected by impedance matching, high frequency attenuation and noise immunity.

It is possible to design a high frequency transmission line using only a single conductor. Nevertheless, most high frequency signals use differential transmission lines (i.e. a pair of coupled conductors carrying signals of opposite polarity). Although differential signaling appears wasteful of both pins and signal traces it results in much better noise immunity. Differential signals produce less conducted noise because the opposite power and ground current flows cancel each other both in the line driver and in the transmission line. Differential signals produce less radiated noise because over a modest distance the opposite fields induced by the opposite currents cancel each other. Differential signals are less susceptible to noise because most sources of noise (common mode noise) tend to affect both signal lines identically, producing a variation in common mode voltage but not in differential voltage.

A.2 Impedance Matching

The AC impedance of a single conductor is determined by the trace geometry, distance to the nearest AC ground plane(s) and the dielectric constant of the material between the trace and the ground plane(s). If the distance between the signal trace and the nearest ground plane is significantly less than the distance to other signal traces the signal trace will behave as a single-ended transmission line. Its AC impedance does not vary with signal polarity although it may vary with frequency due to the properties of the dielectric material. This impedance is often called single ended impedance, Zse.

The AC impedance, Z of a differential transmission line is affected by the configuration of the pair of conductors and the relationship between their signal polarities, in addition to the trace geometry, distance to the nearest AC ground plane(s) and the dielectric constant of the material between the trace and the ground plane(s). If the paired conductors are close enough to interact (coupled), then the impedance for signals of opposite polarity (odd mode impedance, Zodd) will be lower than the impedance for signals of the same polarity (even mode impedance,

Zeven).

If there is minimal coupling between the paired conductors then *Zodd* = *Zeven* = *Zse*. Coupled transmission lines always produce *Zodd* < *Zse* < *Zeven*. The following equations relate effective differential impedance, *Zdiff*, to common mode impedance, *Zcm*, and single ended impedance, *Zse*, to even and odd mode impedances:

$$Zdiff = 2Zodd \qquad Zcm = \frac{Zeven}{2} \qquad Zse = \frac{Zeven + Zodd}{2}$$

Most differential data signals are designed with *zdiff* = 100Ω and 25Ω < *Zcm* < 50Ω.

There is a trade-off in the choice of *Zcm*. *Zcm* = 25Ω (no coupling) may reduce conducted noise for transmission lines with inadequate AC or DC grounding. *Zcm* = 50Ω (close coupling) may reduce radiated noise (crosstalk) which is more critical in backplanes. However close coupling requires careful ground construction to control common mode noise.

The reader may wonder why common mode impedance is meaningful in a differential transmission system. In a perfectly constructed system only odd mode (opposite polarity) signals propagate. However imperfections in the transmission system cause differential to common mode conversion. Once converted into common mode the energy may convert back to differential mode by the same imperfections. Thus, these imperfections convert some of the signal energy from opposite polarities to the same polarity and back.

The two main sources of mode conversion are impedance mismatches which cause part of the energy to be reflected, and differential skew which causes variations in forward signal propagation delay between the individual paths of the differential pair. Impedance mismatches typically occur at boundaries between transmission line segments, including wire bonds, solder joints, connectors, vias, and trace-to-via transitions. Often ignored sources of impedance mismatches at these boundaries are discontinuities within the AC ground itself as well as asymmetric coupling between the individual traces and the AC ground. Differential skew can occur at these same boundaries and also due to mismatched trace lengths in device packages and in PCBs.

A.3 Impedance Definition Details

Differential transmission lines consist of two conductors and a ground plane. The voltage-current relationships at one end of this line can be formulated in terms of a two-port as in Figure A-1.

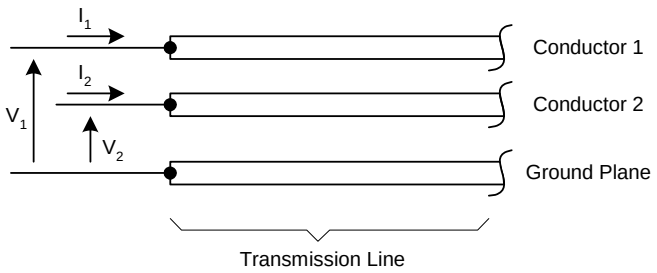


Figure A-1. Transmission Line as 2-port

The voltage current relationships are:

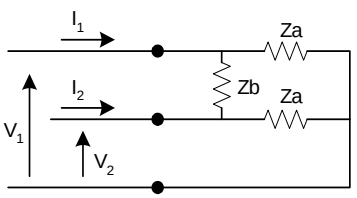
$$V_1 = Z_{11}I_1 + Z_{12}I_2 \qquad V_2 = Z_{21}I_1 + Z_{22}I_2$$

If the line is infinitely long or perfectly terminated, then these four impedance values are the characteristic impedance of the line. The characteristic impedance is a 2 x 2 matrix:

$$\hat{Z}_c = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix}$$

Generally, all four of the matrix entries are complex. But, at frequencies of interest, the inductance and capacitance per unit length dominate so that all four quantities are approximately real positive numbers. For engineering purposes it is common to speak of the impedances as though they are resistances with no imaginary part, keeping in mind that the imaginary part exists. Since the line is passive and symmetric, we have $Z_{11} = Z_{22}$ and $Z_{12} = Z_{21}$ so that the line is described by just two impedance values. If the line is to be perfectly terminated, then we must create a network that is equivalent to $\hat{Z}_c$. That is, we need a 3-terminal (2 nodes + ground) network that presents the same values of Z_{11} and Z_{12} as the line. A T or pi network could be used. The pi network is shown in Figure A-2, along with the impedance values in terms of Z_{11} and Z_{12}.

$$Za = Z_{11} + Z_{12} \qquad Zb = \frac{Z_{11}^{\,2} - Z_{12}^{\,2}}{Z_{12}}$$



$$Zodd = \frac{ZaZb}{2Za + Zb} = Z_{11} - Z_{12} \qquad Zeven = Za = Z_{11} + Z_{12}$$

Figure A-2. Network Terminations

The odd and even mode impedances, *Zodd* and *Zeven*, are other impedance

definitions that are more descriptive, referring to the polarity of the signal propagating the differential pair. In the case of opposite signal polarity in the two lines of the signal pair the odd mode impedance is used. In the case of same signal polarity the even mode is used. *Zodd* and *Zeven* are measured as shown in Figure A-3.

Figure A-3. Measurement of Zodd, Zeven

Zodd	*Zeven*
$V = V_1 = -V_2$	$V = V_1 = V_2$
$I = I_1 = -I_2$	$I = I_1 = I_2$
$Zodd = \frac{V}{I}$	$Zeven = \frac{V}{I}$

Odd mode impedance is the impedance measured when the two halves of the line are driven by equal voltage or current sources of opposite polarity. Even mode impedance is the impedance measured when the two halves of the line are driven by equal voltage or current sources of the same polarity.

From the above equations we see that *Zeven* is always greater than *Zodd* by $2Z_{12}$, where Z_{12} is a measure of the amount of coupling between the lines. This means that *Zeven* is larger than *Zodd* for coupled transmission lines.

A.4 Density considerations

The preceding section showed that, for two idealized forms of termination, *Zodd* is correctly terminated but *Zeven* is not. The first illustrated case, using a 50Ω resistor (or its equivalent) from either terminal to ground (or to AC ground), has become relatively standard. Because it has *ZoddT* = *ZevenT* = 50Ω, it provides correct differential termination and is often close to providing correct common-mode termination.

By increasing the conductor spacing in the transmission line we can decrease Zeven (decrease z_{12}) and bring it closer to 50Ω. But dense backplanes require a large number of transmission lines per unit cross-sectional area of the printed circuit board. This means that the two printed circuit traces comprising the differential transmission line are forced close together, which increases z_{12}. The backplane design is therefore, a compromise between the desire for high density of

transmission lines and a desire for correct common-mode termination.

Transmission lines act as low-pass filters due to skin effect and dielectric absorption. As the density of transmission lines increases, both the series resistance per unit length and the parallel conductance per unit length increase. This, in turn, results in greater attenuation at a given frequency. Thus, high speed backplane design is not just a compromise between density and common-mode matching. There is also a compromise between density and attenuation.

A.5 Common-Mode Impedance and Return Loss

It is demonstrated above that increasing the density of transmission lines in a backplane results in higher common-mode impedance, which is known as interference, and for high amplitudes the receiver is likely to be disrupted.

Common-mode interference arises from several sources. Among them are:

1. Imperfections in driver circuits
2. A difference in length between the two conductors of the transmission line
3. Imperfections in impedance matching across board boundaries, connectors, and vias causing mode conversion, from differential to common mode
4. EMI

The interference resulting from the driver probably has a spectrum that is the same as or similar to that of the signal. EMI arising from coupling into the printed circuit traces should be small, assuming that coupled stripline is used. However, connector pins may be exposed. EMI may have frequency components that are well below signal frequencies, which means that it won't necessarily be attenuated to the extent that signals are. But, at the same time, the lower frequencies are probably poorly coupled into the backplane circuit.

Earlier, two ideal forms of termination were presented based on either one or two resistors. These ideal terminating devices are helpful in examining the relationship between the parameters of the transmission line versus those of the device. Real devices, however, are not simple resistances. They contain parasitic components and a non-ideal path from package pins to die. There may also be a need to AC-couple the terminations.

The most that can be done in this situation is to make the package and the die appear as close to ideal as possible over as much of the signal spectrum as possible. The extent of the deviation from ideal is specified and measured as a function of frequency. The preferred measures are S_{11} (single-ended return loss) or S_{DD11} (differential return loss) as functions of frequency. (Sometimes S_{22} or S_{DD22} are used to indicate an output.) Ideally these return losses are 0 (no reflection) over the frequency range of interest. In dB this is $-\infty$.

Note: Sometimes a return loss is specified as a positive number, it being understood

that this still refers to the log of a reflection coefficient in the range of 0 to 1.

A.6 Crosstalk Considerations

This implementation assumes that the dominant cross talk can come from aggressors other than the transmitter associated with the receiver. Hence NEXT cancellation is not useful.

Crosstalk between channels should be minimized by good design practices. This includes the pin-out arrangement to the driving/receiving ICs, connectors and backplane tracking.

Optimum arrangement for minimizing crosstalk between channels at IC pins is illustrated in Figure A-4 below. Crosstalk between channels can be reduced by grouping TX and RX pins and avoiding close proximity between individual TX and Rx pins. This practice will minimize coupling of noise from TX drivers into RX inputs.

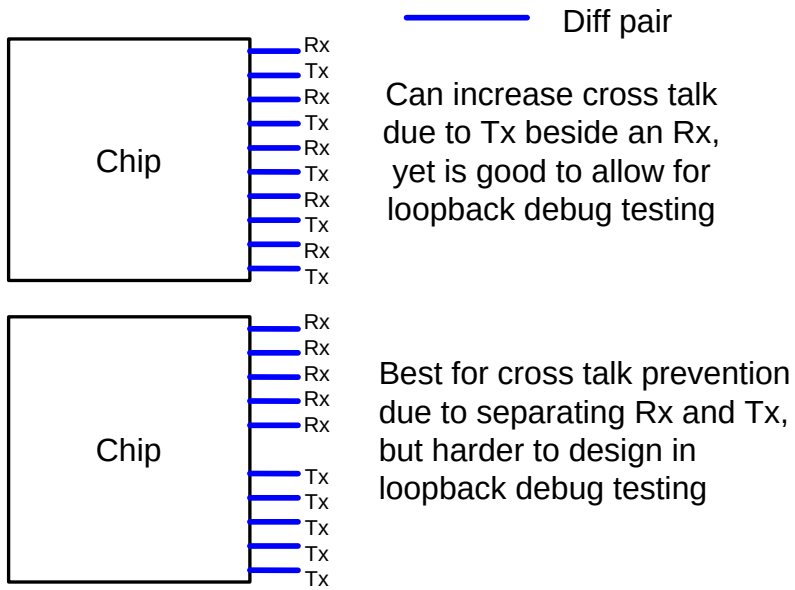


Figure A-4. Minimization of Crosstalk at IC Pins

Crosstalk at connector pins can be minimized by careful optimization of connections as shown in Figure A-5 below.

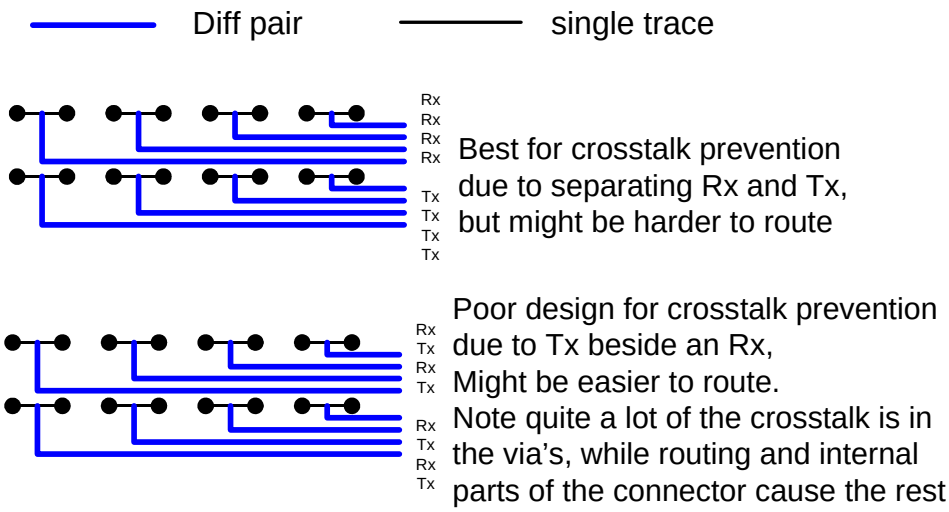


Figure A-5. Minimization of Crosstalk At Connector Pins

Crosstalk between channels over a backplane can be minimized by careful arrangement of tracking, avoiding coupling of noise into RX inputs and increasing spacing "d" between channels as far as possible as shown in Figure A-6 below.

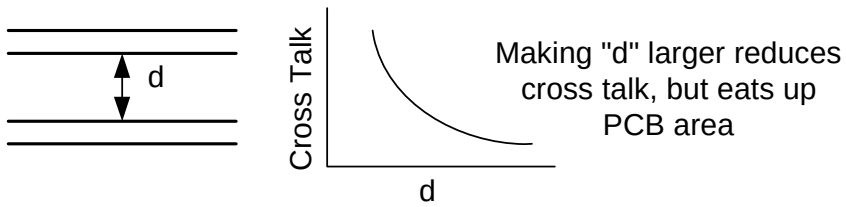


Figure A-6. Minimization of Crosstalk Over Backplane

A.7 Equation Based Channel Loss by Curve Fit

This section describes a technique with specific limitations. It does not include any phase data for the SDD_{21}, and includes no return loss information about SDD_{11} or SDD_{22}, information that is critical for the evaluation of a specific topology's performance. The preceding proposed statistical-eye characterization includes these effects by including the full 4-port s-parameter measurements. The following method is included for information only and is believed to be of relevance to the overall understanding of the channel transfer loss.

One way to specify the channel loss is to have an average or worst case "curve" fit to several real channels. This method includes effects of real vias and connectors. This method typically uses the equation below:

$$Att = -20 * \log(e) * \left(a_1 * \sqrt{f} + a_2 * f + a_3 * f^2\right)$$

Where ***f*** is frequency in Hz, a_1, a_2, and a_3 are the curve fit coefficients and ***Att*** is in dB.

Table A-1 gives some examples of these coefficients and Figure A-7 plots them along with the PCB model and a real 75cm backplane with 5cm paddle cards on both ends. These examples are representative of Level II LR applications but do not represent specifications that a RapidIO link is to comply with.

Table A-1. Curve fit Coefficients

Channel	a_1	a_2	a_3
sRIO[1] (50cm)	6.5e-6	2.0e-10	3.3e-20
75cm[2] "Worse"	6.5e-6	3.9e-10	6.5e-20
75cm[3] "Typical"	6.0e-6	3.9e-10	3.5e-20

[1]Chapter 8 reference 1
[2]Chapter 8 reference 5
[3]Chapter 8 reference 5

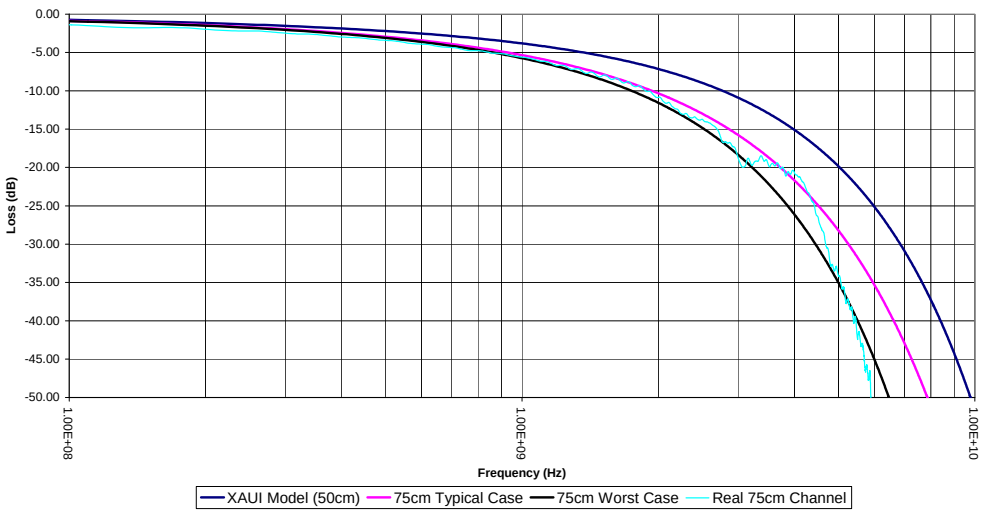


Figure A-7. Equations Based Channel Loss Curves

Annex B BER Adjustment Methodology (Informative)

B.1 Extrapolation of Correlated Bounded Gaussian Jitter to low BERs

For this specification, which has a BER requirement of 1×10^{-15} or lower, measurements to that level are very time consuming (or rely on averaging multi-links), hence it is more practical to only take measurements to Qs around 7 (BER around 1×10^{-12}).

B.1.1 Bathtub Measurements

CBGJ can appear as either GJ or CBHPJ depending upon the Q at which it is linearized.

If HPJ and GJ are measured using a bathtub there is no knowledge as to if the GJ is UUGJ or CBGJ. For system budgeting it is recommended that the bathtub GJ should be assumed to be all UUGJ.

If combined spectral oscilloscope methods are used then UUGJ, UBHPJ, and CBHPJ can be estimated. It is not possible to estimate the CBGJ as it has already become bounded and appears as CBHPJ. For system budgeting it is recommended that this peak value is valid for the extrapolated Q of interest.

B.2 Confidence Level of Errors Measurement

Assuming that a link with a given BER can be modelled as a Bernoulli random process, the following statistics can be assumed.

Given,

p is the probability of error

$q = (1-p)$ is the probability of not having an error

n is the number of bits received and measured

then

$m = np$ is the expected number of errors received

$\sigma = \sqrt{npq}$ is the sigma of the variation of the number of errors received

As an example process, for a 3 sigma confidential level

$$p = 10^{-12}$$

$$n = 100 \cdot 10^{12}$$

$$m = 100$$

$$\sigma = 10$$

$$m\Big|_{max}^{min} = [m + Q\sigma]\Big|_{Q=-3}^{Q=3}$$

$$m\Big|_{max}^{min} = \begin{matrix} 70 \\ 130 \end{matrix}$$

To assess the accuracy of such a measurement an equivalent process with a higher BER can be calculated that would show the same limit of error for the same confidence level and measured number of bits.

$$m\Big|_{max} = E[m] - Q\sigma$$
$$m\Big|_{max} = np - Q\sqrt{npq}$$
$$m\Big|_{max} = np - Q\sqrt{np(1-p)}$$

Solving the quadratic equation for p

$$p = 1.69 \times 10^{-12}$$

B.3 Eye Mask Adjustment for Sampling Oscilloscopes

The data mask is defined for the bit error rate of the link. Given that this bit error rate is very small, typical oscilloscope measurement will not sample enough points to be able to verify compliance to these mask.

B.3.1 Theory

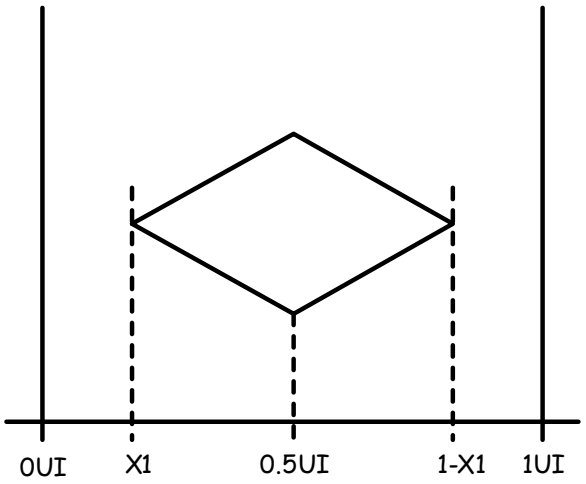


Figure B-1. Example Data Mask

Given an example eye mask, Figure B-1, the extremes of the mask, X1, are defined as a linear addition of a Gaussian and high probability jitter component.

$$X1 = \frac{HPJ}{2} + Q \cdot GJ_{rms}$$

where

HPJ is the high probability jitter

GJ_{rms} is the Gaussian distributed jitter

Q is the GJ multiplication factor

Given a low sample population and the requirements for mask verification to achieve a hit or no-hit result, X1 must be adjusted according to the sample population and the confidence level that a particular peak to peak is achieved.

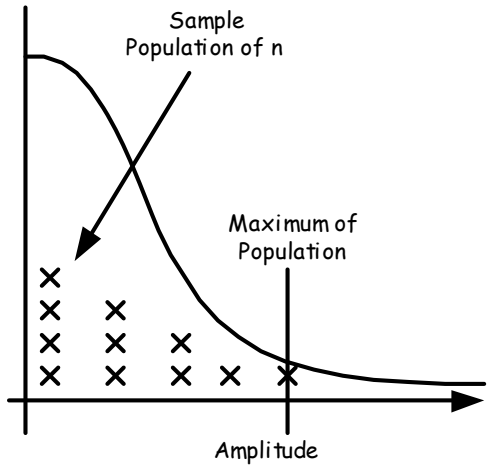


Figure B-2. Example Data Mask

Given a random process the probability of measuring a particular maximum amplitude on an oscilloscope requires one sample to lie on the maximum and all other samples to lie below this value. Referring this all to a half Gaussian distribution and a population of n, there are n different ways this can occur,

$$P(x_m) = nQ(x_m)\left(\int_0^{x_m} Q(x)dx\right)^{n-1}$$

where

x_m is the random variable of the maximum amplitude measured

x is the random variable of the underlying random jitter process

$Q(x)$ is the Q function of the Normal probability density function

n is the sample population

$P(x_m)$ is a probability density function

The equation above is solved and the probability of attaining a given maximum (normalized to the sigma) for various populations plotted, Figure B-3.

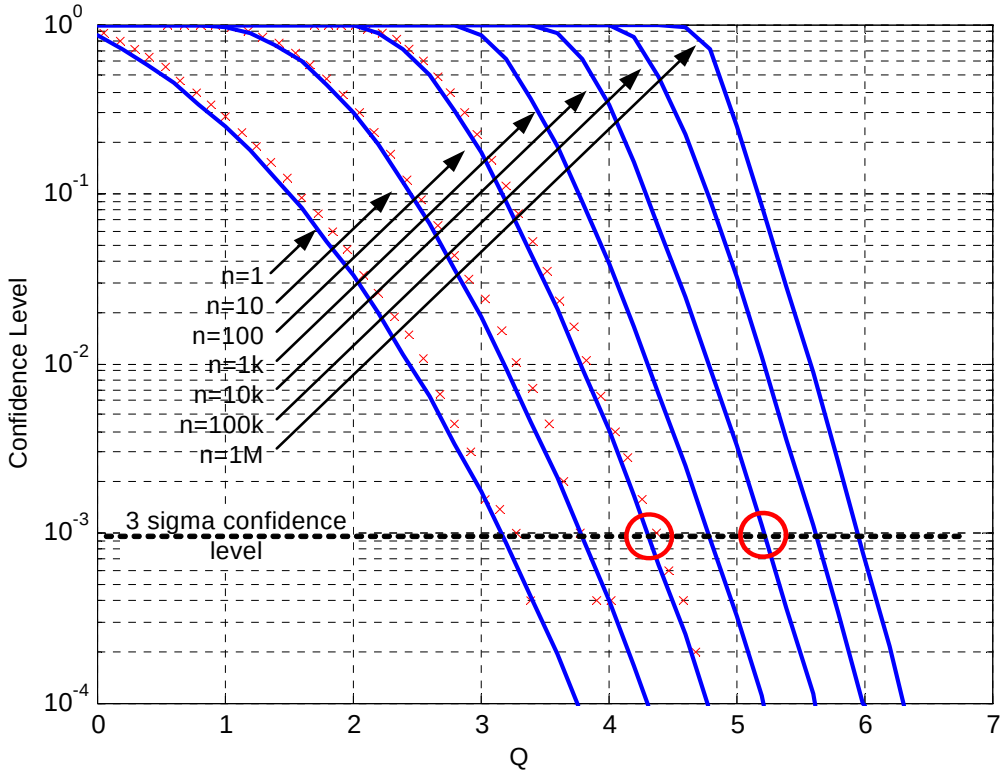


Figure B-3. Cumulative Distribution Function of Maximum Amplitude

B.3.2 Usage

Given a known sampling population, n, calculated from the measurement time, average transition density and sampling/collection frequency of the oscilloscope the three sigma confidence level (i.e. 1.3×10^{-3}) of the measured Gaussian jitter peak value can be read from Figure B-3. This value should be multiplied by 2 to give the full peak to peak value of the random jitter.

The three sigma confidence level should be understood as ensuring that 99.96% of all good devices do not violate the eye mask. To limit the number of bad devices that also pass the eye mask it is strongly recommended that the sample population be chosen as to give a Q larger than 5.

For example, referring to the red circled intersections Figure B-3, if we calculate that the sample population for an oscilloscope was 100 i.e. n=100, then for a 3 sigma confidence this equals a Q of 4.2. As the recommended Q value is 5 we should increase the sample population to 10k to give a Q of 5.2.

Blank page

Annex C Interface Management (Informative)

C.1 Introduction

This appendix contains state machine descriptions that illustrate a number of behaviors that are described in the *RapidIO Part 6: LP-Serial Physical Layer Specification*. They are included as examples and are believed to be correct, however, actual implementations should not use the examples directly.

C.2 Packet Retry Mechanism

This section contains the example packet retry mechanism state machine referred to in Section 5.8, “Packet Transmission Protocol".

Packet retry recovery actually requires two inter-dependent state machines in order to operate, one associated with the input port and the other with the output port on the two connected devices. The two state machines work together to attempt recovery from a retry condition.

C.2.1 Input port retry recovery state machine

If a packet cannot be accepted by a receiver for reasons other than error conditions, such as a full input buffer, the receiver follows the state sequence shown in Figure C-1.

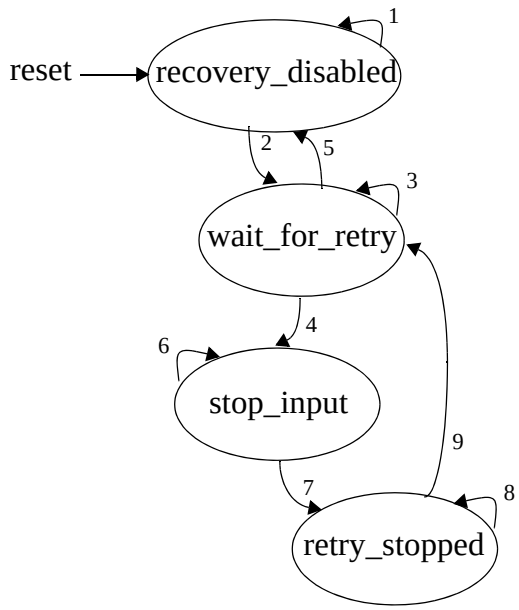


Figure C-1. Input Port Retry Recovery State Machine

Table C-1 describes the state transition arcs for Figure C-1. The states referenced in the comments in quotes are the RapidIO LP-Serial defined status states, not states in this state machine.

Table C-1. Input Port Retry Recovery State Machine Transition Table

Arc	Current State	Next state	Cause	Comments
1	recovery_disabled	recovery_disabled	Remain in this state until the input port is enabled to receive packets.	This is the initial state after reset. The input port can't be enabled before the initialization sequence has been completed, and may be controlled through other mechanisms as well, such as a software enable bit.
2	recovery_disabled	wait_for_retry	Input port is enabled.	
3	wait_for_retry	wait_for_retry	Remain in this state until a packet retry situation has been detected.	
4	wait_for_retry	stop_input	A packet retry situation has been detected.	Usually this is due to an internal resource problem such as not having packet buffers available for low priority packets.
5	wait_for_retry	recovery_disabled	Input port is disabled.	
6	stop_input	stop_input	Remain in this state until described input port stop activity is completed.	Send a packet-retry control symbol with the expected ackID, discard the packet, and don't change the expected ackID. This will force the attached device to initiate recovery starting at the expected ackID. Clear the "Port Normal" state and set the "Input Retry-stopped" state.
7	stop_input	retry_stopped	Input port stop activity is complete.	

Table C-1. Input Port Retry Recovery State Machine Transition Table (Continued)

Arc	Current State	Next state	Cause	Comments
8	retry_stopped	retry_stopped	Remain in this state until a restart-from-retry or link request (restart-from-error) control symbol is received or an input port error is encountered.	The "Input Retry-stopped" state causes the input port to silently discard all incoming packets and not change the expected ackID value.
9	retry_stopped	wait_for_retry	Received a restart-from-retry or a link request (restart-from-error) control symbol or an input port error is encountered.	Clear the "Input Retry-stopped" state and set the "Port Normal" state. An input port error shall cause a clean transition between the retry recovery state machine and the error recovery state machine.

C.2.2 Output port retry recovery state machine

On receipt of an error-free packet-retry control symbol, the attached output port follows the behavior shown in Figure C-2. The states referenced in the comments in quotes are the RapidIO 8/16 LP-LVDS defined status states, not states in this state machine.

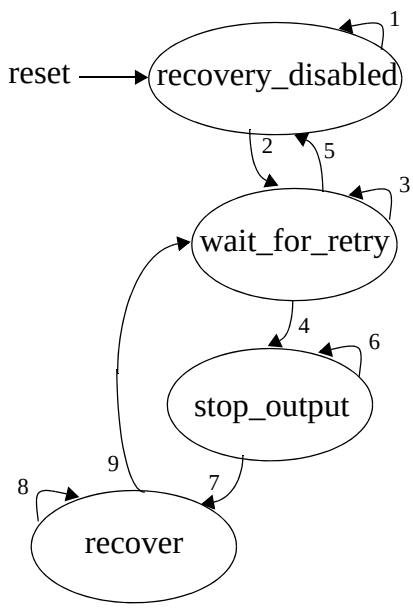


Figure C-2. Output Port Retry Recovery State Machine

Table C-2 describes the state transition arcs for Figure C-2.

Table C-2. Output Port Retry Recovery State Machine Transition Table

Arc	Current State	Next state	Cause	Comments
1	recovery_disabled	recovery_disabled	Remain in this state until the output port is enabled to receive packets.	This is the initial state after reset. The output port can't be enabled before the initialization sequence has been completed, and may be controlled through other mechanisms as well, such as a software enable bit.
2	recovery_disabled	wait_for_retry	Output port is enabled.	
3	wait_for_retry	wait_for_retry	Remain in this state until a packet-retry control symbol is received.	The packet-retry control symbol shall be error free.
4	wait_for_retry	stop_output	A packet-retry control symbol has been received.	Start the output port stop procedure.
5	wait_for_retry	recovery_disabled	Output port is disabled.	
6	stop_output	stop_output	Remain in this state until the output port stop procedure is completed.	Clear the "Port Normal" state, set the "Output Retry-stopped" state, and stop transmitting new packets.
7	stop_output	recover	Output port stop procedure is complete.	
8	recover	recover	Remain in this state until the internal recovery procedure is completed.	The packet sent with the ackID value returned in the packet-retry control symbol and all subsequent packets shall be retransmitted. Output port state machines and the outstanding ackID scoreboard shall be updated with this information, then clear the "Output Retry-stopped" state and set the "Port Normal" state to restart the output port. Receipt of a packet-not-accepted control symbol or other output port error during this procedure shall cause a clean transition between the retry recovery state machine and the error recovery state machine. Send restart-from-retry control symbol.
9	recover	wait_for_retry	Internal recovery procedure is complete.	Retransmission has started, so return to the wait_for_retry state to wait for the next packet-retry control symbol.

C.3 Error Recovery

This section contains the error recovery state machine referred to in Section 5.13.2, “Link Behavior Under Error."

Error recovery actually requires two inter-dependent state machines in order to operate, one associated with the input port and the other with the output port on the two connected devices. The two state machines work together to attempt recovery.

C.3.1 Input port error recovery state machine

There are a variety of recoverable error types described in detail in Section 5.13.2, “Link Behavior Under Error". The first group of errors are associated with the input port, and consists mostly of corrupt packet and control symbols. An example of a corrupt packet is a packet with an incorrect CRC. An example of a corrupt control symbol is a control symbol with error on the 5-bit CRC control symbol. The recovery state machine for the input port of a RapidIO link is shown in Figure C-3.

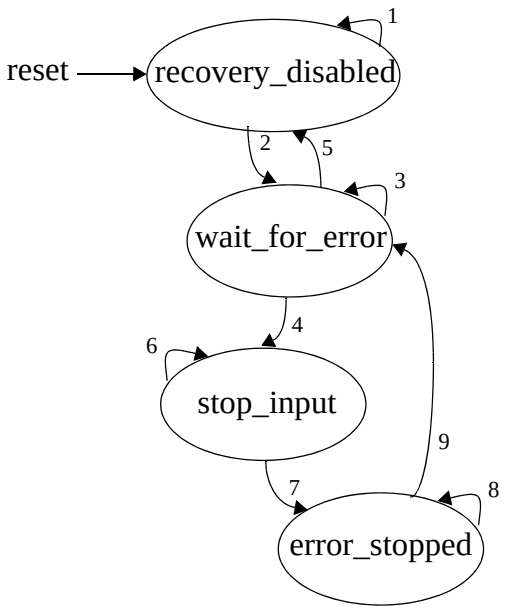


Figure C-3. Input Port Error Recovery State Machine

Table C-3 describes the state transition arcs for Figure C-3. The states referenced in the comments in quotes are the RapidIO LP-Serial defined status states, not states in this state machine.

Table C-3. Input Port Error Recovery State Machine Transition Table

Arc	Current State	Next state	Cause	Comments
1	recovery_disabled	recovery_disabled	Remain in this state until error recovery is enabled.	This is the initial state after reset. Error recovery can't be enabled before the initialization sequence has been completed, and may be controlled through other mechanisms as well, such as a software enable bit.
2	recovery_disabled	wait_for_error	Error recovery is enabled.	
3	wait_for_error	wait_for_error	Remain in this state until a recoverable error is detected.	Detected errors and the level of coverage is implementation dependent.
4	wait_for_error	stop_input	A recoverable error has been detected.	An output port associated error will not cause this transition, only an input port associated error.
5	wait_for_error	recovery_disabled	Error recovery is disabled.	
6	stop_input	stop_input	Remain in this state until described input port stop activity is completed.	Send a packet-not-accepted control symbol and, if the error was on a packet, discard the packet and don't change the expected ackID value. This will force the attached device to initiate recovery. Clear the "Port Normal" state and set the "Input Error-stopped" state.
7	stop_input	error_stopped	Input port stop activity is complete.	
8	error_stopped	error_stopped	Remain in this state until a link request (restart-from-error) control symbol is received.	The "Input Error-stopped" state causes the input port to silently discard all subsequent incoming packets and ignore all subsequent input port errors.
9	error_stopped	wait_for_error	Received a link request (restart-from-error) control symbol.	Clear the "Input Error-stopped" state and set the "Port Normal" state, which will put the input port back in normal operation.

C.3.2 Output port error recovery state machine

The second recoverable group of errors described in Section 5.13.2, "Link Behavior Under Error" is associated with the output port, and is comprised of control symbols that are error-free and indicate that the attached input port has detected a transmission error or some other unusual situation has occurred. An example of this situation is indicated by the receipt of a packet-not-accepted control symbol. The state machine for the output port is shown in Figure C-4.

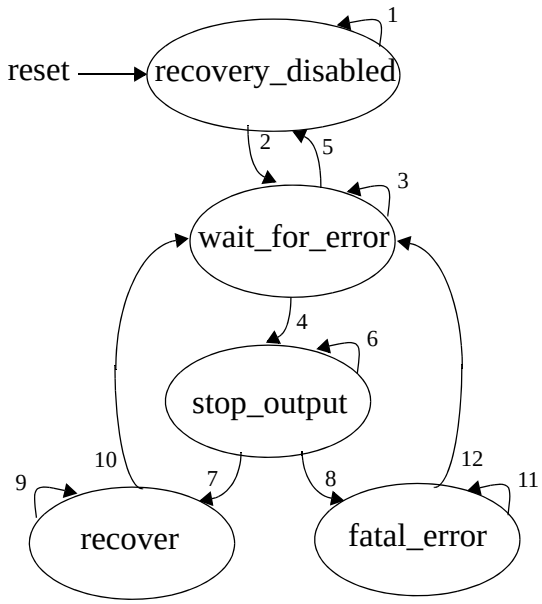


Figure C-4. Output Port Error Recovery State Machine

Table C-4 describes the state transition arcs for Figure C-4. The states referenced in the comments in quotes are the RapidIO 8/16 LP-LVDS defined status states, not states in this state machine.

Table C-4. Output Port Error Recovery State Machine Transition Table

Arc	Current State	Next state	Cause	Comments
1	recovery_disabled	recovery_disabled	Remain in this state until error recovery is enabled.	This is the initial state after reset. Error recovery can't be enabled before the initialization sequence has been completed, and may be controlled through other mechanisms as well, such as a software enable bit.
2	recovery_disabled	wait_for_error	Error recovery is enabled.	
3	wait_for_error	wait_for_error	Remain in this state until a recoverable error is detected.	Detected errors and the level of coverage is implementation dependent.
4	wait_for_error	stop_output	A recoverable error has been detected.	An input port associated error will not cause this transition, only an output port associated error.
5	wait_for_error	recovery_disabled	Error recovery is disabled.	

Table C-4. Output Port Error Recovery State Machine Transition Table (Continued)

Arc	Current State	Next state	Cause	Comments
6	stop_output	stop_output	Remain in this state until an exit condition occurs.	Clear the "Port Normal" state, set the "Output Error-stopped" state, stop transmitting new packets, and send a link-request/input-status control symbol. Ignore all subsequent output port errors. The input on the attached device is in the "Input Error-stopped" state and is waiting for a link-request/input-status in order to be re-enabled to receive packets. An implementation may wish to timeout several times before regarding a timeout as fatal using a threshold counter or some other mechanism.
7	stop_output	recover	The link-response is received and returned an outstanding ackID value	An outstanding ackID is a value sent out on a packet that has not been acknowledged yet. In the case where no ackID is outstanding the returned ackID value shall match the next expected/next assigned ackID value, indicating that the devices are synchronized. Recovery is possible, so follow recovery procedure.
8	stop_output	fatal_error	The link-response is received and returned an ackID value that is not outstanding, or timed out waiting for the link-response.	Recovery is not possible, so start error shutdown procedure.
9	recover	recover	Remain in this state until the internal recovery procedure is completed.	The packet sent with the ackID value returned in the link-response and all subsequent packets shall be retransmitted. All packets transmitted with ackID values preceding the returned value were received by the attached device, so they are treated as if packet-accepted control symbols have been received for them. Output port state machines and the outstanding ackID scoreboard shall be updated with this information, then clear the "Output Error-stopped" state and set the ‘Port Normal" state to restart the output port.
10	recover	wait_for_error	The internal recovery procedure is complete.	retransmission (if any was necessary) has started, so return to the wait_for_error state to wait for the next error.

Table C-4. Output Port Error Recovery State Machine Transition Table (Continued)

Arc	Current State	Next state	Cause	Comments
11	fatal_error	fatal_error	Remain in this state until error shutdown procedure is completed.	Clear the "Output Error-stopped" state, set the "Port Error" state, and signal a system error.
12	fatal_error	wait_for_error	Error shutdown procedure is complete.	Return to the wait_for_error state.

C.3.3 Changes in Error Recovery Behavior for CT

The basic states, as previously described, apply to the overall port. Each VC must carry some independent state:

Packets in a transmitter's VC queue are: pending transmission, sent-pending acknowledgement, or acknowledged (and subsequently removed from the queue). RT and CT VCs keep this same information, but behave slightly differently on error recovery. In RT queues, packets sent-pending acknowledgment, are returned to the pending transmission state. Packets pending acknowledgment in CT queues are moved to the acknowledged state. In this way, the sent packets in the CT queue are not resent.

(Note that it may not be necessary to keep the actual packet in a CT VC, only track the needed acknowledges to keep the credit balance for transmitter flow control accurate.)

Blank page

Annex D Critical Resource Performance Limits (Informative)

The RapidIO LP-Serial layer is intended for use over links whose length ranges from centimeters to tens of meters. The shortest length links will almost certainly use copper printed circuit board traces. The longer lengths will require the use of fiber optics (optical fiber and electro-optical converters) to overcome the high frequency losses of long copper printed circuit board traces or cable. The longer lengths will also have significant propagation delay which can degrade the usable bandwidth of a link.

The serial protocol is a handshake protocol. Each packet transmitted by a port is assigned an ID (the ackID) and a copy of the packet is retained by the port in a holding buffer until the packet is accepted by the port's link partner. The number of packets that a port can transmit without acknowledgment is limited to the lesser of the number of distinct ackIDs and the number of buffers available to hold unacknowledged packets. Which ever is the limiting resource, ackIDs or holding buffers, will be called the "critical resource".

The concern is the time between the assignment of a critical resource to a packet and the release of that resource as a consequence of the packet being accepted by the link partner. Call this time the resource_release_delay. When the resource_release_delay is less than the time it takes to transmit a number of packets equal to the number of distinct critical resource elements, there is no degradation of link performance. When the resource_release_delay is greater than the time it takes to transmit a number of packets equal to the number of distinct critical resource elements, the transmitter may have to stall from time to time waiting for a free critical resource. This will degraded the usable link bandwidth. The onset of degradation will depend on the average length of transmitted packets and the physical length of the link as reflected in the resource_release_delay.

The following example provides some idea of the impact on link performance of the interaction between link length and a critical resource. For purposes of this example, the following assumptions are made.

1. The link is a 4 lane (4x) link.
2. The link uses optical fiber and electro-optical transceivers to allow link lengths of tens of meters. The propagation delay of the optical fiber is 0.45c.
3. The width of the data path within the port is 4 bytes.

4. The data path and logic within the port run at a clock rate equal to the aggregate unidirectional data rate of the link divided by 32. This is referred to as the logic clock. One cycle of this clock is referred to a one logic clock cycle. (If the aggregate unidirectional baud rate of the link was used to compute the logic clock, the baud rate would be divided by 40. With 8B/10B encoding, the baud rate is 1.25 times the data rate.)
5. The minimum length packet header is used. Write request packets have a length of 12 bytes plus a payload containing an integer multiple of 8 bytes. Read request packets have a length of 12 bytes. Read response packets have a length of 8 bytes plus a payload containing an integer multiple of 8 bytes.
6. The beginning and end of each packet is delimited by a control symbol. A single control symbol may delimit both the end of one packet and the beginning of the next packet.
7. Packet acknowledgments are carried in packet delimiter control symbols when ever possible to achieve the efficiency provided by the dual stype control symbol. This implies that a packet acknowledgment must wait for an end-of-packet control symbol if packet transmission is in progress when the packet acknowledgment becomes available.
8. The logic and propagation delay in the packet transmission direction is comprised of the following components.

Table D-1. Packet Transmission Delay Components

Item	Time required
Generate start-of-packet control symbol (critical resource is available)	1 logic clock cycle
Generate start-of-packet control symbol CRC	1 logic clock cycle
8B/10B encode delimiter and start-of-packet control symbol	1 logic clock cycle
Serialize and transmit delimiter and start-of-packet control symbol	1 logic clock cycle
PCB copper and electro-optical transmitter delay	2 ns
Optical fiber delay	fiber_length/0.45c
Electro-optical receiver and pcb copper delay	2 ns
Receive and deserialize delimiter and start-of-packet control symbol	0.5 logic clock cycles
Receive and deserialize packet	depends on packet
Receive and deserialize delimiter and end-of-packet control symbol	1 logic clock cycle
8B/10B decode delimiter and end-of-packet control symbol	1 logic clock cycle
Check CRC of end-of-packet control symbol	1 logic clock cycle
Make packet acceptance decision	1 logic clock cycle

9. The logic and propagation delay in the packet acknowledgment direction is comprised of the following.

Table D-2. Packet Acknowledgment Delay Components

Item	Time required
Wait for end-of-packet if packet transmission is in progress, generate packet-acknowledgment control symbol and control symbol CRC	depends on packet >= 2 logic clock cycles
8B/10B encode delimiter and packet-acknowledgment control symbol	1 logic clock cycle
Serialize and transmit delimiter and packet-acknowledgment control symbol	1 logic clock cycle
PCB copper and electro-optical transmitter delay	2 ns
Optical fiber delay	fiber_length/0.45c
Electro-optical receiver and pcb copper delay	2 ns
Receive and deserialize delimiter and packet-acknowledgment control symbol	0.5 logic clock cycles
8B/10B decode delimiter and packet-acknowledgment control symbol	1 logic clock cycle
Check CRC of packet-acknowledgment control symbol	1 logic clock cycle
Make decision to free critical resource	1 logic clock cycle

The packet times in the above tables depend on packet length which in turn depends on packet type and payload size. Since packet traffic will typically involve a mixture of packet types and payload sizes, the traffic in each direction will be assumed to contain an equal number of read, write and response packets and average payloads of 8, 32, and 64 bytes.

The number of logic clock cycles required to transmit or receive a packet is given in the following table as a function of packet type and payload size.

Table D-3. Packet Delays

Packet Type	Packet Header bytes	Data Payload bytes	Transmit/Receive Time logic clock cycles
Read	12	0	3
Response	8	8	4
		32	10
		64	18
Write	12	8	5
		32	11
		64	19

Using the above table and the assumed equal number of read, write and response packets, the average number of logic clock cycles to transmit or received a packet is 4, 8, and 13.3 respectively for packet payloads of 8, 32, and 64 bytes. The average wait for the completion of a packet being transmitted is assumed to be 1/2 the transmit time.

The following table gives the maximum length of the optical fiber before the packet

transmission rate becomes limited by the critical resource for a 4x link operating at unidirectional data rates of 4.0, 8.0 and 10.0 Gb/s.

Table D-4. Maximum Transmission Distances

Number of Critical Resources Available	Data Payload (Bytes)	Maximum Fiber Length Before Critical Resource Limited (Meters)		
		4.0 Gb/s link	8.0 Gb/s link	10.0 Gb/s link
4	8	-	-	-
	32	4.3	1.9	1.4
	64	11.4	5.5	4.3
8	8	9.7	4.6	3.5
	32	23.6	11.5	9.1
	64	42.2	20.8	16.6
16	8	31.1	15.3	12.1
	32	62.2	30.8	24.6
	64	103.7	51.6	41.1
24	8	52.5	26.0	20.7
	32	100.8	50.2	40.0
	64	165.2	82.3	65.7
32	8	74.0	36.7	29.3
	32	139.5	69.5	55.5
	64	226.7	113.1	90.3

Annex E Manufacturability and Testability (Informative)

It is not possible in many cases for assembly vendors to verify the integrity of soldered connections between components and the printed circuit boards to which they are attached. Alternative methods to direct probing are needed to insure high yields for printed circuit assemblies which include LP-Serial RapidIO devices.

It is recommended that component vendors support IEEE Std. 1149.6 (commonly known as "AC-JTAG") on all connections to LP-Serial RapidIO links. (Note: IEEE Std. 1149.6 is needed, in addition to IEEE Std. 1149.1, due the fact that RapidIO LP-Serial lanes are AC-coupled.) This provides boundary scan capability on all TD, TDN, RD, and RDN pins on a component which supports one or more LP-Serial RapidIO ports.

The IEEE Std. 1149.6 is available from the IEEE.

Blank page

Annex F Multiple Port Configuration Example (Informative)

F.1 Introduction

This appendix contains flow-chart descriptions that illustrates the Port-Width negotiation process described in Chapter 4, “PCS and PMA Layers. They are included as examples and are believed to be correct, however, actual implementations and system design should not use the examples directly.

F.2 System with Different Port Width Capabilities

In a high-performance system, a high-bandwidth switch processing element is often used to aggregate traffic; while the connecting agents can be ones of lower bandwidth. Under this circumstance, the switch processing element has to identify the discrepancy of port-widths between link partners and set up accordingly. Figure shows a typical system with a switch processing element connected between the System Host and two connecting Agents. The system is set up as follows:

- System Host is connected to Switch A
- Switch A has a 8x port which is capable of multiple port configuration.
- Agent B has a 4x port connected to Switch A lanes 4-7
- Agent C has a 2x port connected to Switch A lanes 0-3.

The following example is used to illustrate the negotiation that will take place between Switch A and Agents B and C. It is assumed that the System Host and the Switch A have already established error-free communication.

1. By default, the 8x-port of Switch A looks for an 8x connection but fails to come up with its link partner; thus, falling to 1x mode on lane 0 or 2.
2. Agent B fails to establish 4x link with Switch A. It tries to fall back to 1x mode on its lane 0 or 2 but still fails. Its 4x port has failed.
3. Agent C fails to establish a 2x link with Switch A. When it tries to fall back to 1x mode, it succeeds in lane 0 and re-establish communication with Switch A on its lane 0.
4. System Host reads through the established 1x link between Switch A and Agent C. From the Vendor Port-Width CAR of Agent C, System Host discovers that Agent C can support 2x mode.

5. System Host checks Switch A for its support of 2x mode on its lower quad-link.
6. System Host writes to the Port Width Override CSR to force both Switch A lower quad-link.
7. System Host puts Agent C back to 2x mode.
8. A 2x-link is established between Switch A and Agent C.
9. System Host discovers from Vendor Port-Width CAR in Agent B (not through Switch A because the link was not established yet) that Agent B supports 4x mode. It also discovers that Switch A supports multiple port configuration (from its Vendor Specific registers) and its extra port is available (Vendor Port-Width CAR).[1]
10. System Host configures the new port on the upper-quad link of Switch A.
11. Agent B now recognizes a 4x link partner.
12. A 4x link is now established between Switch A and Agent B.

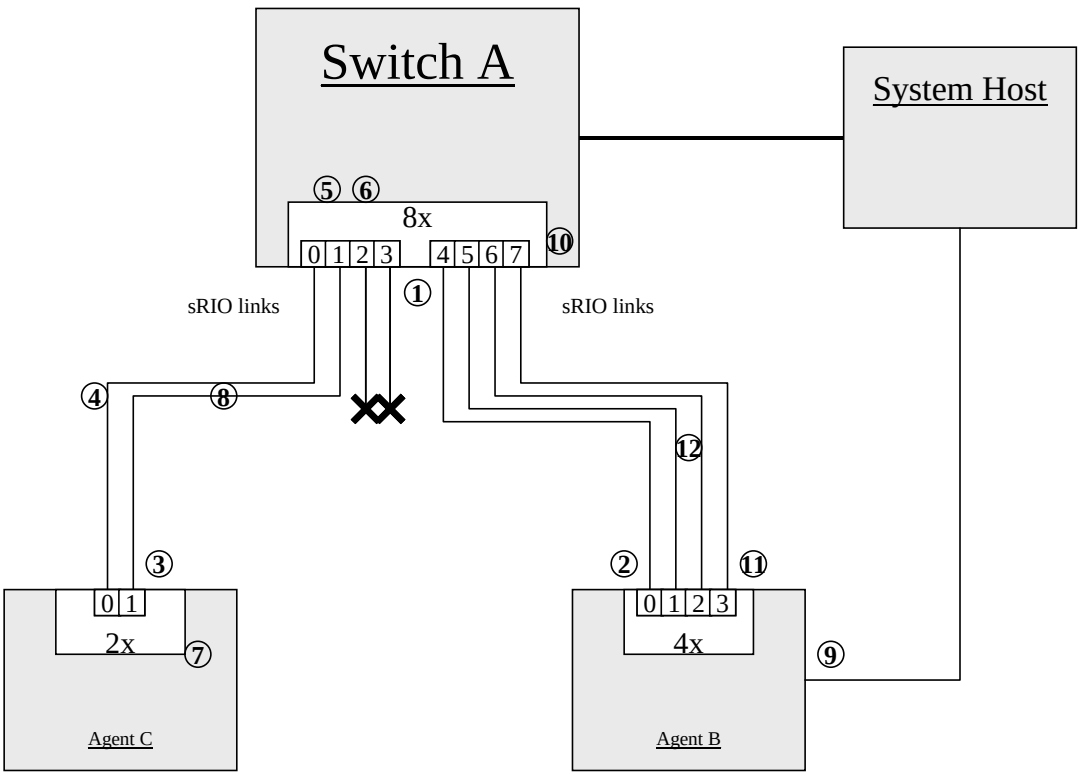


Figure F-1. Example system with asymmetric port-width capabilities

[1]Steps 9 to 12 are optional. Switch A is not required to support multiple-port configuration to be compliant.

Glossary of Terms and Abbreviations

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

A

AC Coupling. A method of connecting two devices together that does not pass DC.

Agent. A processing element that provides services to a processor.

ANSI. American National Standards Institute.

B

Big-endian. A byte-ordering method in memory where the address n of a word corresponds to the most significant byte. In an addressed memory word, the bytes are ordered (left to right) 0, 1, 2, 3, with 0 being the most significant byte.

Bridge. A processing element that connects one computer bus to another, allowing a processing element on one bus to access an processing element on the other.

C

Capability registers (CARs). A set of read-only registers that allow a processing element to determine another processing element's capabilities.

Code-group. A 10-bit entity produced by the 8B/10B encoding process and the input to the 8B/10B decoding process.

Command and status registers (CSRs). A set of registers that allow a processing element to control and determine the status of another processing element's internal hardware.

Continuous Transmission (CT). A mode of packet transmission that allows some packet loss to minimize latency by not retransmitting packets.

Control symbol. A quantum of information transmitted between two linked devices to manage packet flow between the devices.

CRC. Cyclic redundancy code

D

Deadlock. A situation in which two processing elements that are sharing resources prevent each other from accessing the resources, resulting in a halt of system operation.

Deferred or delayed transaction. The process of the target of a transaction capturing the transaction and completing it after responding to the the source with a retry.

Destination. The termination point of a packet on the RapidIO interconnect, also referred to as a target.

Device. A generic participant on the RapidIO interconnect that sends or receives RapidIO transactions, also called a processing element.

Device ID. The identifier of a processing element connected to the RapidIO interconnect.

Direct Memory Access (**DMA**). A process element that can independently read and write system memory.

Distributed memory. System memory that is distributed throughout the system, as opposed to being centrally located.

Double word. An eight byte quantity, aligned on eight byte boundaries.

E

EMI. Electromagnetic Interference.

End point. A processing element which is the source or destination of transactions through a RapidIO fabric.

End point device. A processing element which contains end point functionality.

End point free device. A processing element which does not contain end point functionality.

Ethernet. A common local area network (LAN) technology.

External processing element. A processing element other than the processing element in question.

F

Fabric. A series of interconnected switch devices, typically used in reference to a switch fabric.

Field or Field name. A sub-unit of a register, where bits in the register are named and defined.

FIFO. First in, first out.

Full-duplex. Data can be transmitted in both directions between connected processing elements at the same time.

G

Globally shared memory (GSM). Cache coherent system memory that can be shared between multiple processors in a system.

H

Half-word. A two byte or 16-bit quantity, aligned on two byte boundaries.

Header. Typically the first few bytes of a packet, containing control information.

I

Initiator. The origin of a packet on the RapidIO interconnect, also referred to as a source.

I/O. Input-output.

IP. Intellectual Property

ITU. International Telecommunication Union.

L

Little-endian. A byte-ordering method in memory where the address n of a word corresponds to the least significant byte. In an addressed memory word, the bytes are ordered (left to right) 3, 2, 1, 0, with 3 being the most significant byte.

Local memory. Memory associated with the processing element in question.

LP. Link Protocol

LSB. Least significant byte.

LVDS. Low voltage differential signaling.

M

Message passing. An application programming model that allows processing elements to communicate through special hardware instead of through memory as with the globally shared memory programming model.

MSB. Most significant byte.

N

Non-coherent. A transaction that does not participate in any system globally shared memory cache coherence mechanism.

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.

P

Packet. A set of information transmitted between devices in a RapidIO system.

Payload. The user data embedded in the RapidIO packet.

PCB. Printed circuit board.

PCS. Physical Coding Sublayer.

PMA. Physical Media Attachment.

Port-write. An address-less write operation.

Priority. The relative importance of a transaction or packet; in most systems a higher priority transaction or packet will be serviced or transmitted before one of lower priority.

Processing Element (**PE**). A generic participant on the RapidIO interconnect that sends or receives RapidIO transactions, also called a device.

Processor. The logic circuitry that responds to and processes the basic instructions that drive a computer.

R

Receiver. The RapidIO interface input port on a processing element.

Reliable Transmission (RT). A mode of operation that guarantees packet delivery by retransmitting packets when an error occurs.

S

Sender. The RapidIO interface output port on a processing element.

Semaphore. A technique for coordinating activities in which multiple processing elements compete for the same resource.

Serializer. A device which converts parallel data (such as 8-bit data) to a single bit-wide datastream.

Source. The origin of a packet on the RapidIO interconnect, also referred to as an initiator.

SRAM. Static random access memory.

Switch. A multiple port processing element that directs a packet received on one of its input ports to one of its output ports.

T

Target. The termination point of a packet on the RapidIO interconnect, also referred to as a destination.

Transaction. A specific request or response packet transmitted between end point devices in a RapidIO system.

Transaction request flow. A sequence of transactions between two processing elements that have a required completion order at the destination processing element. There are no ordering requirements between transaction request flows.

W

Word. A four byte or 32 bit quantity, aligned on four byte boundaries.

Write port. Hardware within a processing element that is the target of a port-write operation.

Blank page