

QorIQ Serial RapidIO Debug Tips

This application note outlines some common bring-up issues that customers may face when using the Serial RapidIO (SRIO) protocol on NXP QorIQ devices. The document covers issues related to device errata, hardware design, and software or configuration that may affect SRIO operation or performance. These guidelines aim to help with debugging problems and speed up the bring-up process. The debugging tips apply to QorIQ and Qonverge devices that implement the SRIO controller, including P-, T-, and B- series devices.

Contents

1. Errors due to device errata	2
2. Hardware issues	3
3. Memory map issues	6
4. Software issues	8
5. Revision history	15



1 Errors due to device errata

Several SRIO and SerDes related errata exist that impact QorIQ devices. Some of the errata affect SRIO link operation that could cause failure to complete link initialization at the desired port width or cause unrecoverable bit errors. For these errors, workaround requires performing a software re-training and/or overriding of certain registers. Some errata affect SRIO operation at certain frequencies; therefore, the minimum safe platform frequency requirement must be met. Other SerDes related errata restrict which SRIO protocol speeds are available for use. [Table 1](#) lists some common SRIO errata that affect QorIQ and Qonverge devices.

When link training errors are observed, the user should:

1. Implement all errata workarounds for errata that affect the revision of the device being used.
2. Remove workarounds for errata that have been fixed on the revision of the device being used.
3. Refer to the device errata document for a complete list of SRIO related errata and the impacted silicon revisions.

Table 1. Errata affecting SRIO

Errata	Description	Workaround	Impacted devices
SRIO A-007837	SRIO downtrains from 4x to 2x port width mode	Disable 2x training	T4240 T2080 B4860
SRIO A-008093	SRIO 5G x1 is not supported at platform frequencies below 525 MHz	Select SRIO 5G x4 or x2 if platform frequency is below 525MHz	T4240 T2080 (see Data Sheet) B4860 (see Data Sheet)
SRIO A-004034	SRIO downtrains to 2x or 1x, or may fail to complete link initialization	Perform software re-training sequence	P2040/P2041 P4080 P3041 P5020
SRIO A-004042	An SRIO port enabled for 2x operation may fail to train when connected to a device which supports 2x, but is operating in 1x mode	Perform software re-training sequence	P2040/P2041 P3041 P5020
SRIO-A006	SRIO 1x and 2x operation may see excess errors on Rx with low platform frequencies	Set the platform frequency to the minimum safe platform frequency of operation	P2040/P2041 P4080 Rev 2.0 P3041 P5020 Rev 1.0
SerDes A-007186	SerDes Ring VCO does not maintain lock throughout specified temperature range	For SRIO 2.5 Gbaud or 5 Gbaud protocols that use the Ring VCO, use the alternate protocols that use the LC VCO instead. For SRIO 3.125 Gbaud protocols that use Ring VCO, override the SerDes calibration values.	T4240 T2080 B4860

Table 1. Errata affecting SRIO

Errata	Description	Workaround	Impacted devices
SerDes A-007809	SerDes 2 LC VCO does not work for SRDS_PRTCL_S2=0x29 (SRIO1 and SRIO2 3.125 Gbaud)	Override LC VCO in SerDes PLL register	T2080
SerDes 8	Interfering SerDes PLLs can result in jitter increase	Use only SerDes protocol combinations supported on P4080 (see AN4065)	P4080
SerDes 9	Intermittent SerDes lane training failure when a differential signal from a connected device is not received on SD_RXn/ $\overline{\text{SD_RXn}}$ pins	Override SerDes bank register and perform software re-training sequence	P4080
SerDes A-004580	Internal tracking loop can falsely lock causing unrecoverable bit errors	Override default settings in SerDes bank registers	P2040/P2041 P3041 P4080 P5020
SerDes A-006	SerDes 3.125 Gbaud is not supported	Use the higher voltage alternate device part number to operate at 3.125 Gbaud	P2040/P2041 Rev 1.0 P3041 Rev 1.0 P5020 rev 1.0

2 Hardware issues

2.1 Clocking

SerDes clocking restrictions allow each SRIIO protocol a set of valid SerDes reference clocks that depend on the protocol speed. For the T- and B-series devices, reference clock selection is made in the RCW field SRDS_PLL_REF_CLK_SEL_Sn. Additionally, selection between 5 Gbaud and 2.5 Gbaud is made in the SRDS_DIV_SRIO_Sn field. For the P-series devices, the desired SRIIO speed is based on the SRDS_RATIO_Bn and SRDS_DIV_Bn fields, which determine the SerDes clock ratio and divider. SerDes clocking restriction examples for the T4240 and P4080 are shown in [Figure 1](#) and [Figure 2](#).

To ensure proper SerDes clocking, the user must:

1. Provide a valid SerDes clock input for the selected SRIIO speed.
2. Program the RCW fields for the selected SRIIO speed.
3. Ensure the actual SerDes clock input frequency is the same as selected SerDes clock frequency.

Table 18-9. Valid SerDes Reference Clocks and RCW Encodings

SerDes Protocol (given lane)	Valid reference clock frequency	Valid setting as determined by SRDS_PRTCL_Sn	Valid setting as determined by SRDS_PLL_REF_CLK_SEL_Sn	Valid setting as determined by SRDS_DIV_[prot]_Sn
Networking Protocols (SerDes 1 and SerDes 2)				
SGMII (1.25 Gbaud)	100 MHz	SGMII @ 1.25 Gbaud	0: 100 MHz	Don't Care
	125 MHz		1: 125 MHz	
2.5x SGMII (3.125 Gbaud)	125 MHz	SGMII @ 3.125 Gbaud	0: 125 MHz	Don't Care
	156.25 MHz		1: 156.25 MHz	
QSGMII (5.0 Gbps)	100 MHz	Any QSGMII	0: 100 MHz	Don't Care
	125 MHz		1: 125 MHz	
XAUI (3.125 Gbaud)	125 MHz	XAUI @ 3.125 Gbaud	0: 125 MHz	Don't Care
	156.25 MHz		1: 156.25 MHz	
HiGig (3.125 Gbaud)	125 MHz	HiGig @ 3.125 Gbaud	0: 125 MHz	Don't Care
	156.25 MHz		1: 156.25 MHz	
HiGig[2] (3.75 Gbaud)	125 MHz	HiGig[2] @ 3.75 Gbaud	0: 125 MHz	Don't Care
	156.25 MHz		1: 156.25 MHz	
XFI (10.3125 Gbaud) or 10GBASE-KR	156.25 Mhz	XFI @ 10.3125 Gbaud	0: 156.25 MHz	Don't Care
Non-networking Protocols (SerDes 3 and SerDes 4)				
PCI Express 2.5 GT/s (doesn't negotiate upwards)	100 MHz ¹	Any PCIe	0: 100 MHz	inter
	125 MHz ¹		1: 125 MHz	2'b10: 2.5 G
PCI Express 5 GT/s (can negotiate up to 5 Gbps)	100 MHz ¹	Any PCIe	0: 100 MHz	2'b01: 5.0 G
	125 MHz ¹		1: 125 MHz	
PCI Express 8 GT/s (can negotiate up to 8 Gbps)	100 MHz ¹	Any PCIe	0: 100 MHz	2'b00: 8.0 G
	125 MHz ¹		1: 125 MHz	
Serial RapidIO 2.5 Gbaud	100 MHz	SRIO @ 2.5/5 Gbaud	0: 100 MHz	1: 2.5 G
	125 MHz		1: 125 MHz	
Serial RapidIO 3.125 Gbaud	125 MHz	SRIO @ 3.125 Gbaud	0: 125 MHz	Don't Care
	156.25 MHz		1: 156.25 MHz	
Serial RapidIO 5 Gbaud	100 MHz	SRIO @ 2.5/5 Gbaud	0: 100 MHz	0: 5.0 G
	125 MHz		1: 125 MHz	

Figure 1. T4240 valid SerDes reference clocks

Table 4-39. Valid SerDes RCW Encodings and Reference Clocks (continued)

SerDes Protocol (given lane)	Valid Reference Clock Frequency	Valid Setting as Determined by SRDS_RATIO_Bn	Valid Setting as Determined by SRDS_DIV_Bn
PCI Express (5 Gbps) ¹	100 MHz	50:1	/1
	125 MHz	40:1	/1
Serial RapidIO (2.5 Gbps)	100 MHz	25:1	/1
		50:1	/2
	125 MHz	20:1	/1
		40:1	/2
Serial RapidIO (3.125 Gbps)	125 MHz	25:1	/1

Figure 2. P4080 valid SerDes reference clocks

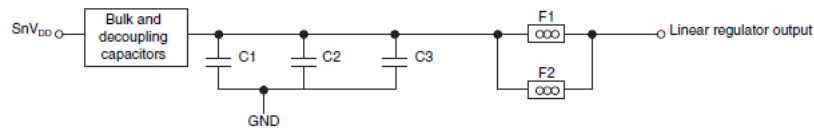
2.2 Board design issues

Poor layout issues are often the cause of data corruption, poor data eye, or a high bit error rate. The SerDes block requires a clean, tightly regulated source of power to ensure low jitter on transmit and reliable recovery of data in the receiver. Proper power supply decoupling capacitors must be in place for the main SerDes core logic supply/pad power supply for the SerDes receiver (SnV_{DD}) and the pad power supply for the SerDes transmitter (XnV_{DD}) pins. Figure 3 shows the recommended decoupling scheme for the T4240.

S_nV_{DD} may be supplied by a linear regulator or sourced by a filtered V_{DD} . Systems may design in both options to allow flexibility to address system noise dependencies. However, for initial system bring-up, the linear regulator option is highly recommended. An example solution for S_nV_{DD} filtering, where S_nV_{DD} is sourced from a linear regulator, shown in this figure. The component values in this example filter are system-dependent and are still under characterization, component values may need adjustment based on the system or environment noise.

Where:

- $C1 = 0.003 \mu\text{F} \pm 10\%$, X5R, with $\text{ESL} \leq 0.5 \text{ nH}$
- $C2$ and $C3 = 2.2 \mu\text{F} \pm 10\%$, X5R, with $\text{ESL} \leq 0.5 \text{ nH}$
- $F1$ and $F2 = 120 \Omega$ at 100 MHz 2A 25% 0603 Ferrite (for example, Murata BLM18PG121SH1)
- Bulk and decoupling capacitors are added, as needed, per power supply design.



X_nV_{DD} may be supplied by a linear regulator or sourced by a filtered G_nV_{DD} . Systems may design-in both options to allow flexibility to address system noise dependencies. However, for initial system bring-up, the linear regulator option is highly recommended. An example solution for X_nV_{DD} filtering, where X_nV_{DD} is sourced from a linear regulator, is shown in the following figure. The component values in this example filter are system-dependent and are still under characterization, so component values may need adjustment based on the system or environment noise.

Where:

- $C1 = 0.003 \mu\text{F} \pm 10\%$, X5R, with $\text{ESL} \leq 0.5 \text{ nH}$
- $C2$ and $C3 = 2.2 \mu\text{F} \pm 10\%$, X5R, with $\text{ESL} \leq 0.5 \text{ nH}$
- $F1$ and $F2 = 120 \Omega$ at 100 MHz 2A 25% 0603 Ferrite (for example, Murata BLM18PG121SH1)
- Bulk and decoupling capacitors are added, as needed, per power supply design.

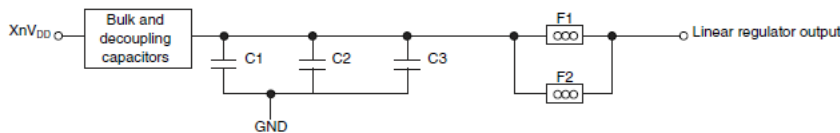


Figure 3. T4240 SerDes power supply decoupling

The SerDes PLL supply provides power to the analog portions of the SerDes PLL. To ensure stability of the internal clock, the power supplied to the SerDes PLL must be filtered. In general, T-series devices require the $AVDD_SDn_PLLn$ to be a filtered version of X_nV_{DD} , while P-series devices require the SerDes PLL supply voltage ($AVDD_SRDSn$) to be a filtered version of S_nV_{DD} . [Figure 4](#) shows an example PLL filtering scheme.

To ensure stability of the internal clock, ensure the power supplied to the PLL is filtered using a circuit similar to the one shown in this figure.

- $AV_{DD_SDn_PLLn}$ should be a filtered version of XnV_{DD} .
- Signals on the SerDes interface are fed from the XnV_{DD} power plane.
- It is recommended that an area fill or power plane split be provided for both AV_{DD} and $AGND$ to provide a low-impedance profile, which helps keep nearby crosstalk noise from inducing unwanted noise.
- Voltage for $AV_{DD_SDn_PLLn}$ is defined at the PLL supply filter and not the pin of $AV_{DD_SDn_PLLn}$.
- A 47 μF 0805 XR5 or XR7, 4.7 μF 0603, and 0.003 μF 0402 capacitor are recommended. The size and material type are important. A 0.33 Ω \pm 1% resistor is recommended.
- **Caution:** These filters are a necessary extension of the PLL circuitry and are compliant with the device specifications. Any deviation from the recommended filters is done at the user's risk.

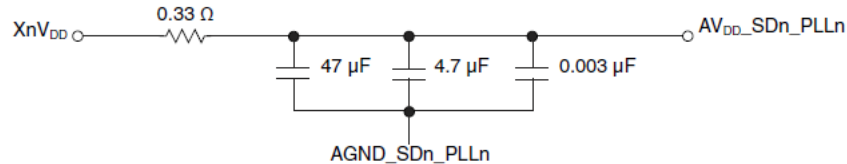


Figure 4. T4240 SerDes PLL filtering

For proper hardware design, the user must:

1. Ensure the board design implements the device-specific SerDes power supply decoupling recommendations in the device data sheet or the design checklist document.
2. Ensure the board design implements the device-specific SerDes PLL filtering recommendations in the device data sheet or the design checklist document.

3 Memory map issues

Accesses to the SRIO interface are memory-mapped, so it is important to ensure memory accesses are redirected to the SRIO interface correctly. Before SRIO transactions are attempted, there must be a TLB entry to cover the area of the memory map used for SRIO transactions. Additionally, all addresses used by the system, except the configuration space mapped by CCSRBAR, must be mapped by a local access window (LAW). A LAW entry must cover the region of the memory map used for SRIO. Finally, the SRIO address translation and mapping unit (ATMU) must initialize the outbound windows to translate an address from the local space to the SRIO space and the inbound windows to translate an address from the external SRIO space to the local address space.

Incorrect address translation, a system hang, or undefined behavior can result from misconfiguration of the LAW/TLB/ATMU windows.

3.1 TLB/LAW/ATMU mapping issues

The starting addresses of the TLB/LAW/ATMU must be aligned to the window size. The relevant registers and fields are shown in [Table 2](#).

Table 2. Window address alignment to size

Window	Starting address	Size
TLB	Effective page number in MAS2[EPN]	MAS1[TSIZE]
	Real page number in MAS7[RPN] and MAS3[RPN]	
Local access window base address	LAWBARH[BASE_ADDR_HIGH] and LAWBARL[BASE_ADDR_LOW]	LAWAR[SIZE]
Outbound window	Base address in ROWBAR[BADD]	ROWAR[SIZE]
	Translation address in ROWTAR[TRAD]	
Inbound window	Base address in RIWBAR[BADD]	RIWAR[SIZE]
	Translation address in RIWTAR[TRAD]	

In the example below, an IO error response will be generated on an outbound NREAD request with the following settings:

Example 1. Misaligned ATMU window

- SRIO outbound window base address 0xAC000000
- SRIO outbound translation address 0x10800000
- SRIO outbound window size 16 MB

In this example, the ATMU translation address is not properly aligned to the window size. The address 0x10800000 is not evenly divisible by 16 MB. An example of a properly aligned translation address is 0x14000000.

To avoid memory mapping issues, the user must:

1. Ensure the TLB effective and real addresses are aligned to the window size.
2. Ensure the LAW addresses are aligned to the window size.
3. Ensure the ATMU base and translation starting addresses are aligned to the window size.
4. Ensure there is no overlap between the TLB/LAW/ATMU windows.
5. Ensure each ATMU and TLB space is covered by a LAW entry.

3.2 TLB memory/cache attributes

The TLB entry for SRIO must set the memory/cache attributes to be cache-inhibited and guarded. All loads and stores to the SRIO interface should bypass the caches. The SRIO should be marked as guarded to prevent speculative reads, which could potentially hang the processor.

To ensure proper TLB settings, the user must:

1. Set the TLB settings for the SRIO entry with WIMGE=0b01010 for cache-inhibited and guarded cache and memory attributes.

4 Software issues

4.1 Errors during initial training sequence

During initialization, both sides of the link send the idle sequences to align the recovered sampling clock. Once a sufficient number of error-free characters is received, the inbound link becomes operational. The port is allowed to transmit after a number of status control symbols have been received from the connected device. When both sides can reliably receive error-free characters, the link is considered operational.

During the initial training sequence when a device powers up or is reset, it is quite common for some errors to occur. These errors are expected while the SerDes clock and data recovery loop locks on the incoming data stream. After the ports are initialized, link training errors should be cleared before beginning normal operation.

After initial link training completes, the user should:

1. Confirm the Error and Status Command and Status Register ESCSR[PO] = 1 and ESCSR[PU] = 0 to indicate the input and output ports have been initialized and are exchanging error-free control symbols with the connected device.
2. Clear the errors in the registers shown in [Table 3](#):

Table 3. Registers to clear after link initialization

Register	Action
ESCSR	Write to clear all port error bits. Only PO should be set to indicate input and output ports are OK.
EDCSR	Clear all transmission errors.
IECSR	Write 1 to clear the retry error threshold exceeded bit.
ECACSR	Clear the captured attributes errors.
PCSECCSR0	Clear the captured packet/control symbol error.
PECCSR1:3	Clear the captured packet errors.
LTLEDCSR	Clear all logical/transport layer errors.
LTLACCSR	Clear the captured address associated with the logical/transport layer error.

Table 3. Registers to clear after link initialization

Register	Action
LTLDIDCCSR	Clear the captured device ID associated with the logical/transport layer error.
LTLCCCSR	Clear the captured address associated with the logical/transport layer error.

- Resume normal operation.

4.2 Software re-training

If the link configuration needs to be updated, that is, change the link width, software can perform link re-training using the following sequence:

- Software on the host must ensure that all RapidIO transactions have completed and link activity has stopped.
- Set CCSR[PD] on the host to disable the port and force the initialization state machines to reset.
- Set PCR[OB DEN] on the host to enable the discarding of any pending packets.
- Clear PCR[OB DEN] on the host.
- Configure new operating width (via CCSR[PWO]) or any other new configurations.
- Clear CCSR[PD] on the host to re-enable the drivers.
- Poll ESCSR[PO] on the host until it is set, which indicates the link has attained port and link initialization.
- Poll ESCSR[PO] on the agent until it is set, which indicates the link has attained port and link initialization.
- Poll ESCSR[OES] on the agent until it is clear, which indicates the link has completed the error recovery sequence initiated from the port disable.
- Poll ESCSR[OES] on the host until it is clear, which indicates the link has completed the error recovery sequence initiated from the port disable.
- Clear the agent's error and status registers.
- Clear the host's error and status registers.
- Begin normal packet transfer.

4.3 Disabled error detection and reporting

Physical and logical/transport layer error detection can be enabled to detect and report errors, and detailed error information can be captured in registers to help identify the cause of the error. By enabling error detection, software can be notified of the error event so it can determine the source of the error and take action based on the information locked in the error capture registers.

For example, if an NREAD request fails to be acknowledged by the connected device, a link time-out error eventually occurs, causing the sending device to enter the output error stopped state. If error detection/reporting is enabled, software is notified of the error reported in the EDCSR with error

information locked in the capture registers ECACSR, PCSECCSR0, and PECCSR1:3. After the error is detected and reported, software can then determine the source of the error, make the appropriate adjustments or fixes to remedy the issue, and recover from the error so normal operation can resume.

Error rate counters and threshold registers are also available to report errors when the error count reaches a level that the system considers to be unacceptable. For example, if poor performance is observed, the counting of the number of packet-not-accepted control symbols can be enabled in the ERECSR[PNA]. This would allow software to determine if there is a high rate of packet-not-accepted returned by the connected device.

To enable error detection/reporting, the user should configure the registers with suggested settings shown in [Table 4](#).

Table 4. Registers to enable error detection and reporting

Register	Recommendation	Error Detection/Reporting
ERECSR	Enable all port errors to be detected and reported.	Check EDCSR for the reported errors. Check ECACSR, PCSECCSR0, PECCSR1:3 for the captured error information.
ERCSR	Set the ERR field to 0b11 to not limit incrementing the port error rate count.	Check the ERCSR[ERC] field for the error count of transmission errors.
ERTCSR	Set the ERFTT and ERDTR fields to 0x01 to set the failed threshold trigger and degraded threshold triggers, respectively, to 1.	When the ERFTT is exceeded, the EDCSR[OFE] bit is set to indicate output port encountered a failed condition. When the ERDTR is exceeded, the EDCSR[ODE] bit is set to indicate the output port has encountered a degraded condition.
PRETCR	Set the RET field to 0xF to generate an error interrupt when the number of consecutive physical retry errors exceed this threshold.	When the RET is exceeded, the EDCSR[RETE] bit is set.
LTLLECSR	Enable all logical layer errors to be detected and reported.	Check LTLLEDCSR for the reported errors. Check LTLACCSR, LTLIDCCSR, LTLCCCSR for the captured error information.
LRETCR	Set the RET field to 0xF to generate an error interrupt when the number of consecutive logical retry errors exceed this threshold.	When RET is exceeded, the LTLLEDCSR[RETE] bit is set.

4.4 Incorrect target ID or device ID

If a packet is sent by a device but the target endpoint does not receive the packet, the user should check that the target ID of the outbound ATMU matches the target endpoint's device ID. Packets with invalid target IDs are dropped at the physical layer. The common transport system size determines how the target ID is defined in the outbound ATMU. [Table 5](#) shows where the target ID and the device ID information are indicated.

Table 5. Target ID and device ID

Transport size	Outbound target ID	Device ID
Small	ROWTAR[TREXAD0-7]	BDIDCSR[BDID]
Large	Bits 0:5 in ROWTEAR[LTGTID] Bits 6:7 in ROWTAR[LTGTID] Bits 8:15 in ROWTAR[TREXAD0-7]	BDIDCSR[LBDID]

The user must:

1. Ensure there is no mismatch in the destination ID and the intended target endpoint's device ID.

4.5 Out-of-sync AckIDs

The acknowledge ID (AckID) is the packet identifier for acknowledgments back to the packet sender. If the link partner is reset when its expected AckID is non-zero, then an error occurs when the link partner receives the next transmitted packet because the link partner's expected AckID has been reset to zero. This causes a mismatch between the transmitted AckID and the expected AckID. The local AckID and the link partner's local AckID values should be re-synchronized to match the expected and transmitted AckID values.

To re-synchronize the AckIDs:

1. Issue a link request/input-status request by setting the Link Maintenance Request Command and Status Register (LMREQCSR) = 0x00000004.
2. Poll the Link Maintenance Response Command Status Register (LMRESPCSR[RV]) until the response valid bit is set.
3. Read the link status in LMRESPCSR[LS].
If LS = 0b10000, the port is accepting packets. Continue to step 4.
If LS != 0b10000, the connected device must be reset.
4. Read the connected device's AckID status in LMRESPCSR[AS].
5. Read the outbound next transmitted AckID value in the Local AckID Status Command and Status Register (LASCSCR[OBA]).
6. Compare the AckIDs from LMRESPCSR[AS] and LASCSCR[OBA]. If the AckIDs do not match, set the outbound next transmitted AckID in LASCSCR[OBA] and the inbound next expected AckID value in LASCSCR[IA] equal to the value read from LMRESPCSR[AS].
7. Continue normal operation.

4.6 Time-out errors

Time-out counters are used to detect certain types of errors such as a lost request or response packet or a lost acknowledgment. When these timers expire before the expected response or acknowledge is received, a time-out error is generated. The QorIQ devices support three time-out counters:

1. The Port Link Time-Out Control Command and Status Register (PLTOCCSR) contains the time-out value for link events, such as sending a packet to receive the corresponding acknowledge or sending a link request to receive the corresponding link response. The timer starts when the request packet is transmitted out on the link. The timer stops when the request is acknowledged with a packet accept by the link partner.
2. The Port Response Time-Out Control Command and Status Register (PRTOCCSR) contains the time-out value for sending a request packet to receive the corresponding response packet. The timer starts when the request packet is transmitted out on the link. The timer stops when the corresponding response packet is received.
3. The Logical Outbound Packet Time-to-Live Configuration Register (LOPTTLCR) contains the time-out value that a packet is allowed to exist within the device. The timer starts when the request is received in the logical layer. The request is then forwarded to the physical layer and gets transmitted out on the link. The timer stops when a packet accept is received on the link.

Figure 5 shows when these time-out counters start and stop.

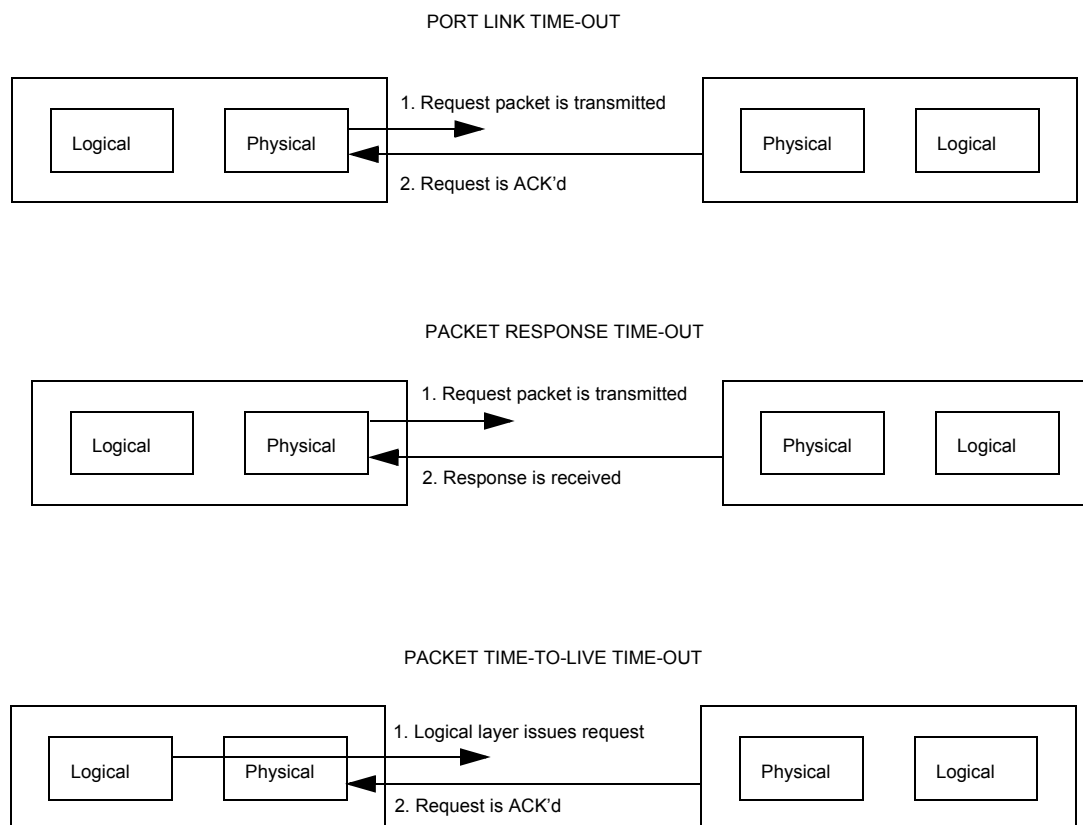


Figure 5. Time-out counters

For proper usage of the time-out counters for detection of lost packets, the user should follow these guidelines:

1. Enable error reporting of port link time-out errors in the Error Rate Enable Command and Status Register (ERECSR[LTO]). This allows detection of an acknowledge or link-response control symbol that is not received within the time-out interval in PLTOCCSR. The link time-out event is indicated in the Error Detect Command and Status Register (EDCSR).
 - a) Read the first four bytes of the captured control character and control symbol information in the Packet/Control Symbol Error Capture Command and Status Register (PCSECCSR0).
 - b) Read bytes 4-15 in the Packet Error Capture Command and Status Registers (PECCSR[1:3]).

2. Enable error reporting of packet response time-out errors in the Logical/Transport Layer Error Enable Command and Status Register (LTLEECSR[PRT]). This allows detection of a response packet that is not received within the time-out interval in PRTOCCSR. The packet response time-out event is indicated in the Logical/Transport Layer Error Detect Command and Status Register (LTLEDCSR[PRT]).
 - a) Read the captured address associated with the error in the Logical/Transport Layer Address Capture Command and Status Register (LTLACCSR).
 - b) Read the captured source and destination IDs associated with the error in the Logical/Transport Layer Device ID Capture Command and Status Register (LTLDIDCCSR).
 - c) Read the captured format and transaction type in the Logical/Transport Layer Control Command and Status Register (LTLCCCSR).
3. Enable error reporting of packet time-to-live errors in LTLLEECSR[PTTL]. This allows detection of a packet that is not received within the time-out interval in LOPTTLCR. The packet time-to-live time-out event is indicated in LTLEDCSR[PTTL].
 - a) Read the captured error information in LTLACCSR, LTLDIDCCSR, and LTLCCCSR.
4. Set the time-out intervals long enough to prevent false time-out errors to be signaled.
5. Ensure the LOPTTLCR value is larger than the PLTOCCSR value.
6. Do not change the time-out counters while there are transmit packets in the SRIO pipeline.
7. Clear the PCR[OBDEN] output buffer drain bit after the LOPTTLCR expires. The OBDEN is automatically set when the packet time-to-live timer expires, causing the packets from the outbound buffer to be drained and not be transmitted. Packets will not be sent out until software clears the OBDEN.

4.7 Insufficient buffer space

If buffer space is not available, the receiving device rejects the packet and sends a packet-retry control symbol to its link partner. A packet-retry control symbol indicates that the receiving device was unable to accept the packet due to a temporary resource conflict, that is, insufficient buffering for packets of priority less than or equal to the retried packet, and a retry is requested for the sender to retransmit the rejected packet. The receiving device enters the input retry-stopped state. The sender of the rejected packet enters the output retry-stopped state and sends a restart-from-retry control symbol, which enables the receiver to start accepting packets after a packet-retry situation. After the receiver exits the input retry-stopped state, it can resume packet reception.

The time associated with resending packets and recovering from the retry-stopped states degrades performance. If excessive packet-retry control symbols are received:

- The link partner should ensure there is sufficient inbound buffer space to receive packets and avoid packet-retry situations. The amount of buffering provided is implementation dependent.
- If possible, the transmitting device should increase the packet priority level if the destination buffers are not available to accept packets of a particular priority. The packet priority is configured in the transaction flow level field (ROWARx[TFLOWLV]).
- Increase time-out values to avoid time-out errors that could result from system congestion when buffers are not available to accept packets.

5 Revision history

This table provides a revision history for this document.

Table 6. Document revision history

Rev. number	Date	Substantive change(s)
0	05/2016	Initial public release

How to Reach Us:

Home Page:

nxp.com

Web Support:

nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address:
nxp.com/SalesTermsandConditions.

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE,

JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, Altivec, C-5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C-Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. Oracle and

Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

© 2016 NXP B.V.

Document Number: AN5295

Rev. 0

05/2016

