

Introduction

The SerialLite II MegaCore® function is a lightweight, chip-to-chip protocol suitable for both packet and streaming data in chip-to-chip, board-to-board, and backplane applications. It offers low protocol overhead, low gate count, and minimal data transfer latency. It provides reliable, high-speed transfers of packets between devices over serial links. The SerialLite II protocol defines packet encapsulation at the link layer, and data encoding at the physical layer. The protocol integrates transparently with existing networks, without software support.

The SerialLite II MegaCore function provides a simple and lightweight way to move data from one point to another reliably at high speeds. It comprises a serial link of up to 16 bonded lanes with logic to provide a number of basic and optional link support functions. The Atlantic interface is the primary access for delivering and receiving data.

Link Error Classifications

Table 1 shows the protocol error types classified by the SerialLite II protocol.

Table 1. Protocol Error Types

Error Type	Description
Catastrophic	A catastrophic error is an unrecoverable error caused by the initialization state machines.
Link	A link error results when the link is not able to transmit or receive data and it triggers the initialization process.
Data	A data error happens when there are bit errors at the physical layer. Physical layer bit errors most likely involve 8b/10b coding violations and may affect one or multiple lanes. Bit errors at the physical layer result in link layer protocol errors or CRC errors.

Most difficulties associated with the SerialLite II MegaCore function involve link initialization and link disconnection. It is important to understand why a SerialLite II link fails to initialize and why a SerialLite II link gets disconnected. The following sections explain these subjects.

Link Initialization Problem

The SerialLite II MegaCore function transitions through several initialization stages that the state machine must complete before the SerialLite II link is up. However, when the link initialization fails, one of the following problems is generally the cause:

- [“Inability to Achieve PLL or Frequency Lock in the Transceiver” on page 2](#)
- [“Inability to Detect Comma Character \(k28.5\)” on page 3](#)
- [“Reversed Polarity Violations” on page 3](#)
- [“Invalid Lane Order” on page 4](#)

- “Deskew FIFO Overflow and Inability to Align Lanes in Multiple Lanes Configuration” on page 5
- “Duration to Hold `ctrl_tc_force_train` is Not Long Enough for Self-synchronizing Link State Machine Mode” on page 5
- “Mismatched SerialLite II Configurations” on page 5

Inability to Achieve PLL or Frequency Lock in the Transceiver

Signals `stat_tc_pll_locked` and `stat_rr_freqlock` detect whether the link initialization problem is caused by PLL or frequency lock issues in the transceiver.

`stat_tc_pll_locked` signal indicates that the transceiver Tx PLL is locked to the `trefclk`; whereas `stat_rr_freqlock` signal indicates whether the transceiver block receiver channel is locked to the data mode in the `rxin` port. In a multi-lane design, this status signal is a bus with a width equivalent to the number of receiver lanes (where bit 0 corresponds to channel 0, and so on).


Another useful signal that determines the transceiver-locked status is `stat_tc_rst_done`. If the transceiver is successfully reset by the SerialLite II MegaCore (frequency locked obtained), then this signal is asserted. The SerialLite II link starts the training sequences to establish the link after this signal is asserted. If this signal is never asserted, the transceiver may not lock the frequency for one or more of the following reasons:

- Incorrect frequency value.
- Improper reset given to the core.

 For more information, refer to the Initialization and Restart section in the *SerialLite II MegaCore Function User Guide*.

- Noise on the line.
- Incorrect transceiver settings; such as incorrect **Pre-emphasis**, incorrect **Equalizer**, insufficient I/O voltage levels (**VOD**) or incorrect **VCCH**.
- The PPM difference to reference clock exceeds the PPM setting.

If the transmitter and receiver have separate clock sources, measure each clock and ensure that it is within the supported PPM. If the PPM setting is exceeded, you may not be able to lock to data.

 For more information on the lock mode of the transceiver, refer to the Automatic Lock Mode section in the *Stratix II GX Transceiver Architecture Overview* chapter of the *Stratix II GX Device Handbook*.

The inability to achieve PLL or frequency lock in the transceiver causes a catastrophic error.

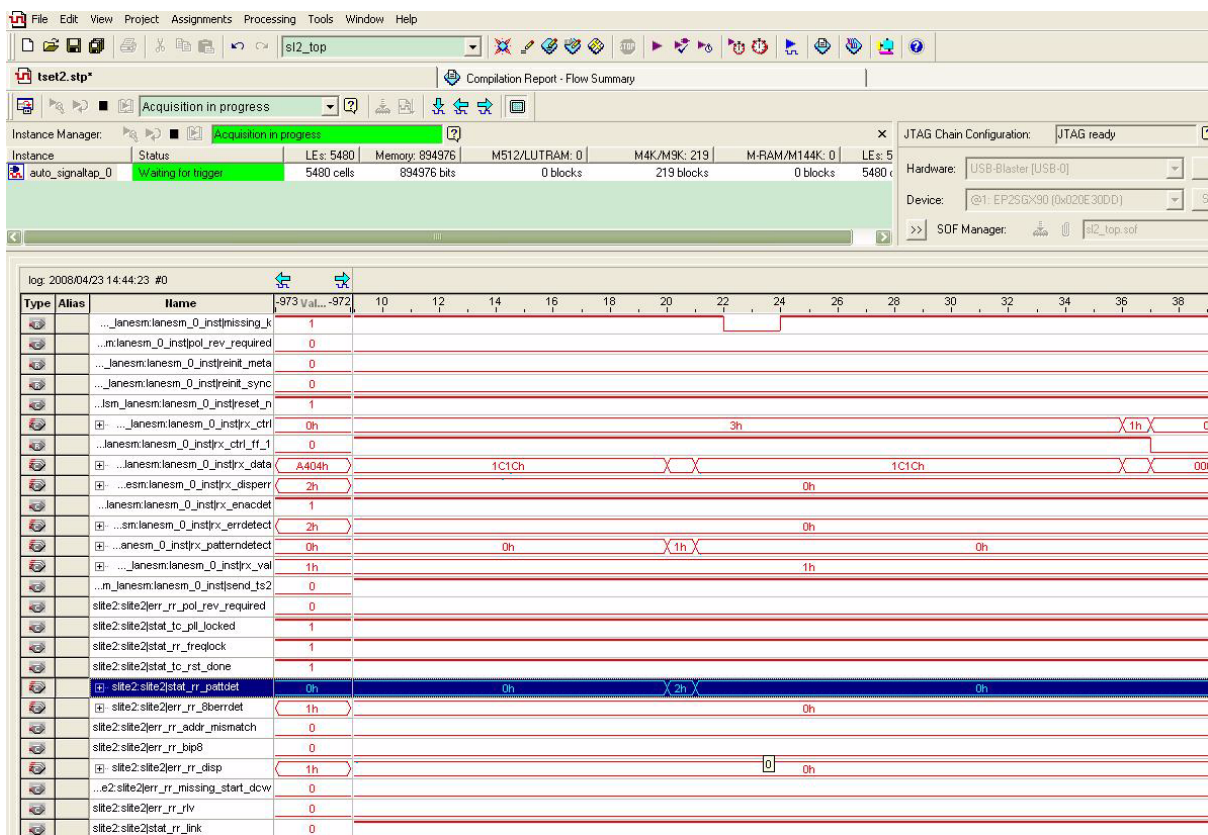
Inability to Detect Comma Character (k28.5)

After the transceiver reset sequences are complete, the transceiver hunts for k28.5 character to initialize the link state machine training sequences. The assertion of signal `stat_rr_pattdet` indicates the detection of the comma character. If `stat_rr_pattdet` does not toggle, the k28.5 character has not been detected, which means the SerialLite II link is never initialized. This situation normally occurs when one core comes out from reset but the core at the adjacent device does not come out from reset due to unlocked frequency problems.

The inability to detect k28.5 character causes a catastrophic error.

The Quartus® II SignalTap® II Logic Analyzer debugging shows the behavior of this signal as shown in Figure 1. The `stat_rr_pattdet` signal is asserted once a comma character is detected on the lane.

Figure 1. SignalTap II Logic Analyzer showing `stat_rr_pattdet` Signal



Reversed Polarity Violations

The SerialLite II MegaCore function offers automatic detection and correction of a reversed polarity lane (per lane detection). This feature is not supported by the Stratix® GX family devices. For Stratix GX, automatic polarity lane correction feature is not supported.

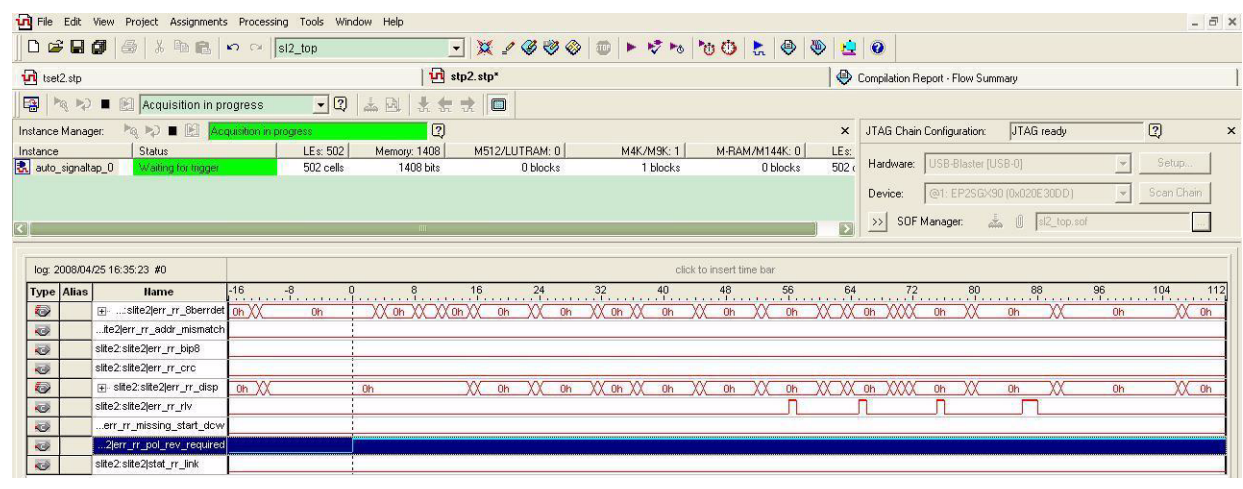
You can use the `err_rr_pol_rev_required` signal as a flag to indicate reversed polarity violations. The signal asserts the following instances:

- For Stratix GX, this signal is asserted when the polarity lane needs to be reversed; but the core is unable to perform the reversal because the feature is unavailable.
- For Arria® GX, Stratix II GX and Stratix IV devices, this signal is only asserted if polarity reversal is required, but no longer needed once the transceiver is put into reversed-polarity mode and the core is not reset. Once the transceiver is put into reversed polarity mode, the SerialLite II core stays in this mode until the core is reset. If the core determines that polarity needs to be changed again, the `err_rr_pol_rev_required` signal is asserted. The polarity is not flipped, and the link is not established.

In both cases, reversed polarity violations cause a catastrophic error. However, only the second case may be recovered through manual reset.

The Quartus II SignalTap II Logic Analyzer debugging shows the behavior of the `err_rr_pol_rev_required` signal as shown in Figure 2.

Figure 2. SignalTap II Logic Analyzer showing `err_rr_pol_rev_required` Signal



In the SignalTap II Logic Analyzer, you can see that when `err_rr_pol_rev_required` is asserted, `stat_rr_link` never goes high.

Invalid Lane Order

The SerialLite II logic checks the lane order, and attempts to reverse the case where the most significant lane of one end of the link is connected to the least significant lane of the other end. For example, a 4-lane system where lane 0 is connected to pin 3, lane 1 to pin 2, lane 2 to pin 1, and lane 3 to pin 0. If the lane order is scrambled (for example, lane 0 is connected to pin 2, lane 1 to pin 3, lane 2 to pin 0, lane 3 to pin 1), the receiving end cannot unscramble it.

There is no signal at the top level that indicates an invalid lane order. However, you can trace the lane order success through the internal signals with the bus name `order` in the `deskews_inst` instance by using the SignalTap II Logic Analyzer.

Invalid lane order causes an unrecoverable catastrophic error.

Deskew FIFO Overflow and Inability to Align Lanes in Multiple Lanes Configuration

Signal `err_dskfifo_oflw` indicates when deskew FIFO buffer overflows. When this signal is asserted, the deskew logic determines that the SerialLite II lanes are outside deskew tolerance (more than 15, 6, or 2 code words between high speed serial lanes [TSIZE 1, 2, 4 respectively]).

Deskew FIFO overflow and the inability to align lanes in multiple lanes configuration cause an unrecoverable catastrophic error.

Duration to Hold `ctrl_tc_force_train` is Not Long Enough for Self-synchronizing Link State Machine Mode

The self-synchronizing link state machine (LSM) is a light-weight implementation of the SerialLite II LSM that is especially useful for data streaming. The `ctrl_tc_force_train` signal must be asserted for the training patterns to be sent. The LSM links up after receiving 64 consecutive valid, error-free training patterns.



Refer to the Self Synchronized Link Up section in the *SerialLite II MegaCore Function User Guide* for information on the time duration required for this signal.

This causes catastrophic error that can be corrected by extending the duration of the `ctrl_tc_force_train` signal in the adjacent device. Typically, the reason for extending the duration is a result of an adjacent device requiring a longer time duration to complete the start up reset sequence for the transceiver.

Mismatched SerialLite II Configurations

A SerialLite II link is unable to initialize if the following parameters are different in the two cores:

- **Data rate**
- **Transfer size**
- **Self-Synchronized Link-up**
- Tx Num Lanes (Core 1) vs Rx Num Lanes (Core 2)
- Rx Num Lanes (Core 1) vs Tx Num Lanes (Core 2)
- **Data Type (Packets or Streaming)**
- **Priority packets and Data packets** port configurations (if non-streaming mode)
For example: Core 1 has priority port talking to Core 2 with only the data port enabled. However, FIFO size differences will not cause any mismatched configurations.
- **Retry-on-error**
- **Segment size** (if **Retry-on-error** is enabled)
- **Enable flow control**
- Tx CRC setting (Core 1) vs Rx CRC setting (Core 2)
- Rx CRC setting (Core 1) vs Tx CRC setting (Core 1)

The difference in parameters between the two cores causes a catastrophic error.

Link Down Problem

Once the link is initialized, data will be sent through the link. However, some of the problems listed below may cause the SerialLite II link to come down.

- “Receives 8 TS1s”
- “ROE Resends 4 Segments without ACK”
- “Clock Compensation Removal FIFO Overflows”
- “Leaky Bucket “Overflows”” on page 7
- “TDS Sequence Received with |ALN| Not Aligned” on page 8
- “Frequency Lost Lock in the Transceiver” on page 8
- “Broadcast Link Problem” on page 8



In some of the cases above, the feature is optional and may not be applicable.

In all cases, when the link gets disconnected, the SerialLite II core will attempt to reestablish the link.

Receives 8 TS1s

Once the SerialLite II link is initialized, if the link state machine receives eight consecutive TS1 or TS2 training sequences, the SerialLite II link gets disconnected. This indicates that there was a link problem or reinitialization request being generated from another SerialLite II core.

The internal signals `k_count_ge_7` in each `lanesm_X_inst`, where X is a lane number, indicate this issue.

ROE Resends 4 Segments without ACK

Retry-on-error (ROE) is an option for priority segments in the link layer. The near-end transmitter will resend the segment of data when the near-end receiver receives a NACK or when it does not receive any ACK after a certain period of time. If the transmitter needs to resend the same segment four times without the reception of the ACK for that segment, the SerialLite II link will get disconnected.

The internal signal that is useful to detect this scenario is `txroemsg_lsm_reinit` in `tx_core_inst`.

This problem causes a link error.

Clock Compensation Removal FIFO Overflows



Clock compensation removal FIFO overflows if there is any incorrect clock PPM settings or when the clocks are outside the supported PPM limits. The internal signal that indicates that the clock compensation removal FIFO overflows is the `icre_lsm_reinit` in `lsm0` instance.

When clock compensation removal FIFO overflows, it causes a catastrophic error that requires the PPM settings to change in the core (and recompile), or replacing the clock crystals on board to meet the maximum supported PPM difference.

Leaky Bucket "Overflows"

The leaky bucket will overflow when there are too many errors occurring in a window of time. The contributing errors (all data errors) are as shown in [Table 2](#).

Table 2. Errors that Cause Leaky Bucket to Overflow

Errors	Description
Disparity Errors (err_rr_disp)	The error signal is asserted when the receiver detects an 8b/10b disparity error. It may be caused by single or multiple bit error.  This bit error may occur due to incorrect transceiver settings (Equalizer , Pre-emphasis setting, VOD , transceiver Bandwith mode).
8b10b Errors (err_rr_8b10b)	The error signal is asserted when the receiver detects an invalid 8b/10b code character. It may be caused by single or multiple bit error.  This bit error may occur due to incorrect transceiver settings (Equalizer , Pre-emphasis setting, VOD , transceiver Bandwith mode).
CRC Errors (err_rr_crc)	This signal is asserted when a CRC error is detected in the received packet.
BIP-8 Errors (err_rr_bip8)	This signal is asserted when a BIP-8 error is detected in the received link management packet.

The SerialLite II MegaCore function uses the “leaky bucket algorithm” as a data error threshold mechanism to detect excessive incidence of data errors. The “leaky bucket algorithm” has the following rules:

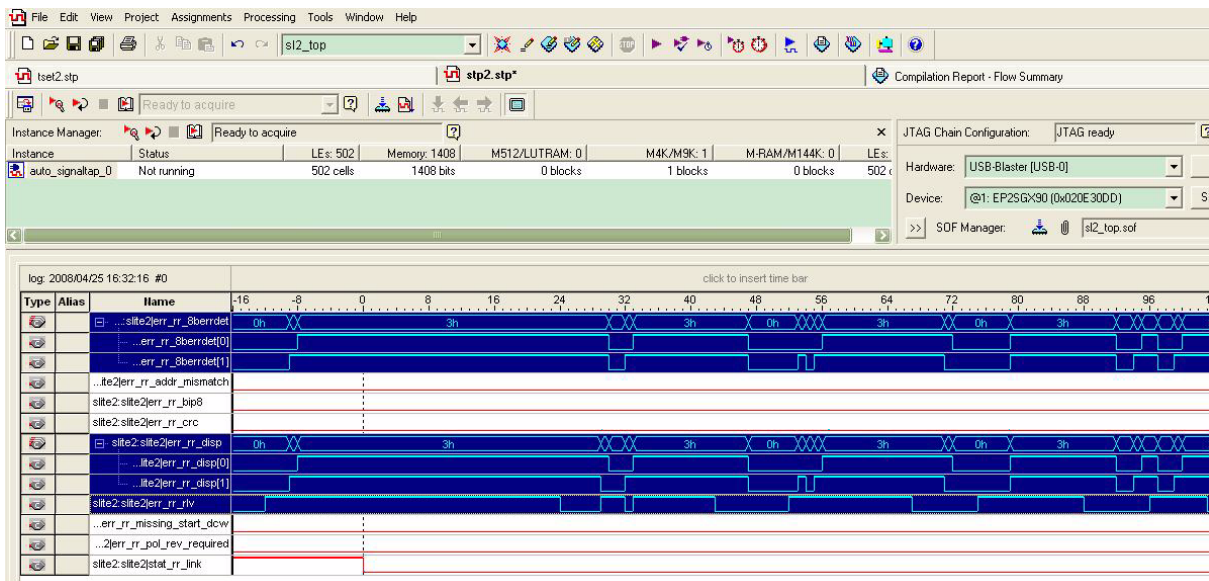
- When a data error is received, the error-threshold counter is incremented by one.
- The error threshold counter is decremented by one every sixteen columns until it reaches zero.
- The decrement event is free running and not synchronized to internal operation or link state.
- The data error threshold is exceeded when the error-threshold counter is greater than or equal to four.

The leaky bucket overflowing is a link error.

This section has many internal signals, but the important signal for debugging is bus `reinit` in `lsm_leaky` instance.

The Quartus II SignalTap II Logic Analyzer debugging shows some transceiver errors (`err_rr_disp` and `err_rr_8berrdet`) asserted causing the link to go down as shown in Figure 3.

Figure 3. SignalTap II Logic Analyzer showing Transceiver Errors



TDS Sequence Received with |ALN| Not Aligned

Once the core performs lane deskew, the align character `|ALN|` is present on all lanes at the same time. After the deskew process completes and the `|ALN|` character is not found on all lanes at the same time, the link will attempt to reinitialize. The internal signal that indicates this problem is `bad_column` in the `deskewsm_inst`.

If `|ALN|` is not aligned, this causes a link error.

Frequency Lost Lock in the Transceiver

There are some cases where the user finds the frequency locked signals are asserted and deasserted after a short period of time. The lost locked scenario is most probably caused by incorrect transceiver settings (**Equalizer**, **Pre-emphasis** setting, **VOD**, transceiver **Bandwidth mode**). These settings mainly depend on the overall system.

When the frequency is lost locked, `rx_freqlocked` signal/bus in the `xcvr_inst` instance is deasserted.

Frequency lost lock in the transceiver causes a link error.

Broadcast Link Problem

For broadcast mode, the SerialLite II MegaCore function is configured to use a single shared transmitter and multiple receivers in the master device. If one of the receivers has a link error, all the other receivers will also report a link error, as the SerialLite II core requires all receivers to reestablish the link.

This problem can be traced through `bcst_link_down` internal signal in each `lanesm_X_inst`, where X is a lane number.

Broadcast link problem causes a link error.

Revision History

Table 3 shows the revision history for this application note.

Table 3. Document Revision History

Date and Revision	Changes Made	Summary of Changes
August 2008, version 1.0	Initial Release.	—



101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Copyright © 2008 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.



I.S. EN ISO 9001