SerialLite MegaCore Function

April 2005, MegaCore Version 1.0.0 Errata Sheet

Introduction

This document addresses known errata and documentation changes for version 1.0.0 of the SerialLite MegaCore® function.

Errata are design functional defects or errors. Errata may cause the SerialLite MegaCore function to deviate from published specifications.

Documentation changes include typos, errors, unclear descriptions or omissions from current published specifications or product documents. These documentation changes or clarifications will be incorporated into an upcoming release of the SerialLite MegaCore function.

SerialLite MegaCore Function v1.0.0 Issues

Altera has identified the following issues that affect the SerialLite MegaCore function v1.0.0:

The Link Does Not Come Up Over Long Cable Distances

An updated **LSM_LINKSM** file, including the timeout logic, is available. The size of the counter is determined by the `idle_count_size` parameter. In the **LSM_LINKSM** file, the `idle_count_size` parameter is set to 8. If that is too small of a value, change it, and recompile your variation.

Affected Configurations

This issue affects all variations for which the wire delay causes the link to time out.

Design Impact

For long cable distances, 2 miles for example, the link never comes up for the default `idle_count_size` of 8.

Workaround

Submit a **mySupport** request. Altera will e-mail you an updated **LSM_LINKSM** file with a parameterized IDLE counter. Use this file to replace the file in the **lib** sub-directory of the SerialLite MegaCore function installation directory.

The size of the counter is determined by the `idle_count_size` parameter. By default, the parameter is set to create an 8-bit IDLE counter. If that counter is too small still, increase the value of the parameter and recompile.

Solution Status

This issue will be fixed by adding a parameter to the next release of the SerialLite MegaCore function.

Flow Control Issue for Regular User Data Port

Flow control link management packets (LMPs) are not transmitted after the regular user data FIFO buffer exceeds the fill level threshold, causing the FIFO buffer to overflow over time.

The problem is due to incorrect parameterization of the regular user data port SCFIFO megafunction, used in the MegaCore function. Originally, the FIFO buffers were only meant to support depths of 128 elements.

Affected Configurations

This issue affects variations that have the flow control and regular user data port parameters enabled, and for which the receiver regular user data port FIFO buffers have depths greater than 128 elements.

Design Impact

The FIFO buffer thresholds used to generate flow control are compared to a bus that has the upper bits mismatched with the `SCFIFO` megafunction, causing the flow control generation signal to be 'x' stated, and thus flow control to never happen.

Workaround

✓ Reduce the size of the regular data port FIFO buffers to be 128-elements deep.

or

1. Submit a **mySupport** request. Altera will e-mail you two files that include RTL fixes for this issue.
2. The directory where you installed the SerialLite MegaCore function includes a sub-directory called **lib.** Use the two files to overwrite the files in that sub-directory.
3. Regenerate the variation.

Solution Status

This issue will be fixed in the next release of the SerialLite MegaCore function.

The Quartus II Software Does Not Merge Transmit Phase-Lock Loops (PLLs) for Multiple Function Instantiations

The Quartus® II software reports a *no-fit* for designs where more than one SerialLite MegaCore function is instantiated in each quad bank of serial inputs and outputs (I/Os). The Quartus II software consumes one ALTGXB transmitter PLL per instantiation, and does not merge the transmit PLLs even if the same reference clock is connected to each instantiation.

Affected Configurations

This issue affects single-lane and dual-lane variations that include more instantiations than the device ALTGXB transmitter PLLs.

Design Impact

Only a single instantiation can be used per quad bank of serial I/Os.

Workaround

The SerialLite wrapper can be modified to instantiate multiple SerialLite protocol core engines connected to a single ALTGXB transmitter PLL. The ALTGXB transmitter PLL can be configured to match the desired number of lanes. Each protocol core instantiation then connects to the associated bits on the ALTGXB transmitter PLL. Submit a **mySupport** request for help with this workaround.

Solution Status

This issue will be fixed in the next release of the SerialLite MegaCore function, requiring the Quartus II software version 5.1.

A Parameter Is Set to an Invalid Value by IP Toolbench

IP Toolbench sets the praif_almost_full parameter to a negative value, instead of a positive value, when a large number of channels is selected.

Affected Configurations

This issue affects variations that have the Number of Lanes parameter set to a value greater than five, and that use small priority packet sizes.

Design Impact

Model generation and synthesis fail when the `Number of Lanes` parameter is set to six or greater.

The `praif_almost_full` parameter is derived from the priority port FIFO buffer size minus eight, and is intended to compensate for pipeline delay. This parameter represents the highest fill value after which, if a new transaction is received, the link goes down. However, the formula used by IP Toolbench to set the `praif_almost_full` parameter generates a FIFO buffer that is too small (negative value) and that cannot handle the worst-case receiver latency, leading to overflows and loss of data.

Workaround

For priority packets, follow these steps:

1. Open the IP Toolbench-produced wrapper file in a text editor.
2. Find the line that sets the `praif_almost_full` parameter.
3. Change the IP Toolbench-computed value by adding four to the current value.

For data packets, follow these steps:

1. Enable the priority port.
2. Change the maximum packet length from 32 (default) to 256.
3. Disable the priority port. The 256 stays in the text box, but is grayed out.

Solution Status

This issue will be fixed in the next release of the SerialLite MegaCore function.

Slow Link Synchronization with VHDL Models

The IP Functional Simulation model is intended to accelerate synchronization. However, the VHDL model does not include this enhancement and synchronizes after approximately 250 µs. A counter is used during link synchronization and is set to 20 internal cycles for

accelerated link synchronization. The netlist or hardware takes 20,000 internal cycles to allow for the PLL lock to occur. The VHDL model synchronization period matches results obtained in hardware testing.

Affected Configurations

This issue affects all variations for which VHDL was chosen as the language during the creation of the IP Functional Simulation model.

Design Impact

Simulation takes an extended period of time to synchronize.

Workaround

Submit a **mySupport** request, and include the Quartus II archive containing the VHDL model produced by IP Toolbench. Our support team will generate a new VHDL IP Functional Simulation model with accelerated link synchronization.

Solution Status

This issue will be fixed in the next release of the SerialLite MegaCore function.

The NumPriorityPackets Parameter Is Missing in the Demonstration Testbench Parameter Text File

The demonstration testbench included with the SerialLite MegaCore function features configuration parameters, and these parameters are listed in a testbench text file produced by IP Toolbench. However, the `NumPriorityPackets` parameter does not appear in the text file when both regular and priority data ports are active.

Affected Configurations

This issue affects all variations that have both the regular data port and the priority data port parameters enabled.

Design Impact

This issue does not impact the design. However, because the parameter is missing from the text file, its value internally defaults to one—causing the testbench to transmit only a single priority packet.

Workaround

If you wish to send more than one priority packet, you must manually add the `NumPriorityPackets` value to the <*variation*>**_tb_params.txt** file.

Solution Status

This issue will be fixed in the next release of the SerialLite MegaCore function.

User-Defined Payload Mode Does Not Support Packets Greater than 256 Bytes

In the demonstration testbench, the packets transmitted and received can contain incrementing payload or user-controlled payload types. If user-controlled payload is manually turned on in the testbench, the packet array must be filled with payload data. However, a variable in the **agen.v** and **amon.v** files is improperly sized; it should be a 16-bit vector to access the 64 Kbyte array, but instead it is an 8-bit vector meaning that only 256 bytes can be accessed. Thus, packets over 256 bytes are corrupted when transmitted, and are not checked properly when received.

Affected Configurations

This issue affects all variations that use the regular data port parameter.

Design Impact

If the user-controlled payload option in the demonstration testbench is used, and the packet sizes are greater than 256 bytes, data errors will occur in the packet being transmitted.

Workaround

Do not use the user-controlled payload option in the demonstration testbench for packet sizes over 256 bytes.

Solution Status

This issue will be fixed in the next release of the SerialLite MegaCore function.

The Demonstration Testbench Parameter List in the User Guide Does Not Match the Text File

The demonstration testbench included with the SerialLite MegaCore function features configuration parameters. However, the parameter names in the testbench text file, produced by IP Toolbench, do not match exactly the names given in the *SerialLite MegaCore Function User Guide*. For example, the user guide uses `NUM_PRIORITY_PACKETS`, while the text file uses `NumPriorityPackets`.

Affected Configurations

This issue affects all variations of the MegaCore function.

Solution Status

The parameter names will be corrected in the next release of the *SerialLite MegaCore Function User Guide*.

The Demonstration Testbench Fails When Generated by the Quartus II 5.0 Software

The ModelSim SE 6.0c software produces an error when running the demonstration testbench. The IP Functional Simulation model created by the Quartus II software version 5.0 contains new additional parameters on the `ALTGX` macro. When simulated with an older macro file, the simulation generates an error because the new parameter is missing from the older Altera® simulation library.

Affected Configurations

This issue affects all demonstration testbenches generated by the Quartus II 5.0 software that are run with the ModelSim SE 6.0c software. The testbench works when generated by the Quartus II 4.2 software. The ModelSim AE 6.0c software has the Altera libraries precompiled within the simulation, and does not experience any problems with the Quartus II 5.0-generated files.

Design Impact

The simulation produces the following error:

```
** Error: (vsim-3043) slite_broadcast.vo(36699):
Unresolved reference to 'allow_gxb_merging' in
ni0O1i.allow_gxb_merging.
```

Workaround

Change all references to the Altera libraries, **C:/MegaCore/slite-v1.0.0/lib/altera**, to **$QUARTUS_ROOTDIR\eda\sim_lib** in the generated **_vlog_arg.txt** file.

Solution Status

This issue will be fixed in the next release of the SerialLite MegaCore function.

The Demonstration Testbench Does not Achieve Link Synchronization

Hardware testing uses initial reset conditions; IP Functional Simulation models randomly assign a power-up level to all flops instead of a reset path. In hardware, the initial value of a flop after power up is always zero. In simulation, the value of the flop after power up can be one. A power-up level of one affects the link state machine, and the link fails to synchronize.

Affected Configurations

This issue could affect all IP Functional Simulation models.

Design Impact

The simulation experiences an extended initialization sequence that results in a timeout. The hardware is not affected, and works as intended.

Workaround

Submit a **mySupport** request, and include the Quartus II archive containing your MegaCore variation files generated by IP Toolbench. Our support team will generate a new IP Functional Simulation model in which randomization of power-up levels is disabled.

Solution Status

This issue will be fixed in the next release of the SerialLite MegaCore function.

The Demonstration Testbench Fails in Unbalanced Configurations

The Verilog HDL demonstration testbench included with the SerialLite MegaCore function loops the high-speed serial interface from the transmitter (`TX_OUT[]`) to the receiver instantiation (`RX_IN[]`). If the receiver and transmitter MegaCore functions do not have matching configurations (for example: CRC-32 payload protection for the transmitter, but no CRC payload protection for the receiver), the testbench fails.

Affected Configurations

This issue affects all variations of the MegaCore function.

Design Impact

The demonstration testbench reports an error, and indicates that the testbench has failed.

Workaround

No workaround is available for this issue. For unbalanced configurations, you must create an environment that uses two independent SerialLite MegaCore functions.

Solution Status

This issue will be fixed in the next release of the SerialLite MegaCore function.

The ModelSim 6.0c Software Produces a Compilation Error When Loading the Demonstration Testbench

The ModelSim 6.0c software produces an error when loading the demonstration testbench. The command `vlog` is used to compile a file called **sl_param.v**, found in the MegaCore library directory. This file is then called by the testbench using: `` `include "sl_param.v" ``. The reference to the actual MegaCore directory is not included as it could be different in the user's environment. The `vlog` command allows the directory to not be directly specified within the testbench when using the ModelSim 5.8d software, but the directory is now specified in the ModelSim 6.0c software.

Affected Configurations

This issue affects all demonstration testbenches that are run with the ModelSim SE 6.0c and ModelSim AE 6.0c software. The testbench works with the ModelSim 5.8d software.

Design Impact

The simulation produces the following error:

```
# ** Error:
C:/MegaCore/slite-v1.0.0/lib/src/sl_param.v(31):
near "parameter":  expecting: LIBRARY CONFIG
# ** Error: C:/Modeltech_60c/win32/vlog failed.
```

Workaround

Copy the **sl_param.v** file from the MegaCore installation directory, **C:/MegaCore/slite-v1.0.0/lib/src/sl_param.v**, to your directory and remove the file reference from the MegaCore generated **_vlog_arg.txt** file.

Solution Status

This issue will be fixed in the next release of the SerialLite MegaCore function.

The ModelSim SE 6.0c Software Produces a Fatal Error With Condition Coverage of the altera_mf.v File

The ModelSim SE 6.0c software produces a fatal error when loading the demonstration testbench. The command `vlog` is used to compile a file called **altera_mf.v**, found in the MegaCore library directory.

Affected Configurations

This issue affects all demonstration testbenches that are run with the ModelSim SE 6.0c software. The testbench works with the ModelSim AE 6.0c software because the **altera_mf.v** does not need to be compiled. Also, older versions of the software—such the ModelSim 5.8d version—are not affected.

Design Impact

The simulation produces the following error:

```
Model Technology ModelSim SE vlog 6.0c Compiler 2005.02 Feb  2 2005
-- Compiling module lcell
```

```
-- Compiling module global
-- Compiling module carry
-- Compiling module cascade
-- Compiling module carry_sum
-- Compiling module exp
-- Compiling module soft
-- Compiling module opndrn
-- Compiling module row_global
-- Compiling module prim_gdff
-- Compiling module dffea
-- Compiling module dffeas
-- Compiling module ALTERA_DEVICE_FAMILIES
** Fatal: Unexpected signal: 11.
** Error:
/opt/cad/quartus_50_B140/eda/sim_lib/altera_mf.v(588)
: Verilog Compiler exiting
```

Workaround

Edit the tool command language (Tcl) script to remove the conditional coverage, `-cover c`, from the `vlog` command. Change line 649 of the **_tb.do** Tcl script to:

```
vlog -O0 -cover bs $VLOGDefineClause -f $VlogArgFile
```

Solution Status

This issue will be fixed in the ModelSim SE 6.0d software.

The Serial Loopback Pin Is Not Documented

Serial loopback is dynamically enabled on a channel-by-channel basis using the `RX_SLPBK` port. When the `RX_SLPBK` signal is high, all blocks that are active when the signal is low remain active. Although serial loopback is enabled, data is still being output on the `TX_OUT[]` port.

When this pin is set, the data on `TX_OUT[]` is looped back onto the `RX_IN[]` port, overriding any user data.

Affected Configurations

This issue affects all variations of the MegaCore function.

Design Impact

The serial loopback pin may be inadvertently set to a value that does not match your design intent.

Workaround

Once you understand the purpose of this pin, set it to a value based on your design requirements.

Solution Status

This pin will be described in the next release of the *SerialLite MegaCore Function User Guide*.

Limitations of the DAV Signal on the Priority Port

The *SerialLite MegaCore Function User Guide*'s description of the DAV signal on the priority port implies that the user logic must stop writing to the SerialLite MegaCore function when the THDAV signal is deasserted. The THDAV signal is asserted once the buffer is empty, pushing the store-and-forward behavior to the user logic, resulting in much lower bandwidth.

If the user logic strictly follows the THDAV, it cannot begin to load a second packet until the first has been transmitted, even though there are two buffers. This requirement limits the priority packet bandwidth of that port. The interface is not optimized for the kind of performance you may need.

Affected Configurations

This issue affects all variations that have the priority port parameter enabled.

Design Impact

If you tie the THDAV port to the THENA port (a common setup), the effective bandwidth on the priority port may hover around 50%.

Workaround

The user logic can continue writing after the negation of the DAV signal (when ROE is not enabled) up to the EOP mark (the EOP must be stalled until the DAV is reasserted).

Solution Status

The DAV port's description will be clarified in the next release of the *SerialLite MegaCore Function User Guide*.

Contact Information

For more information, go to Altera's mySupport website at **www.altera.com/mysupport** and click **Create New Service Request**. Choose the **Product Related Request** form.

Revision History

Table 1 shows the revision history.

Table 1. Revision History

Version	Date	Details of Change
1.0	April 2005	First release of the SerialLite MegaCore Function errata sheet for v1.0.0.

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
www.altera.com
Applications Hotline:
(800) 800-EPLD
Literature Services:
literature@altera.com