SL811HS

January 7, 2011

Errata for SL811HS Embedded USB Host/Slave Controller

This document describes the errata for the SL811HS. Details include errata trigger conditions, available workarounds, and silicon revision applicability. Compare this document to the device data sheet for a complete functional description.

Contact your local Cypress Sales Representative if you have questions.

Part Numbers Affected

Part Number	Package Type	Operating Range
SL811HS	All	Commercial

SL811HS/SL811 Qualification Status

Product status: In production - Qual Report: 014401

SL811HS/SL811 Errata Summary

The following table defines the errata applicability to available SL811HS/SL811 family of devices.

Note Errata titles in this table are hyperlinked. Click an entry to go to its description.

Items	SL811HS/SL811S	Rev Letter/Number	Fix Status
1. Host Mode: SE0 problem in low-speed hub operation	X	1.5	Use workaround
2. Host Mode: Sync to SOF does not apply to low-speed mode	X	1.5	Use workaround
3. Host/Peripheral Mode: 12 MHz operation with sensitive internal PLL	X	1.5	Use workaround
4. Peripheral Mode: Unreliable DMA interface	X	1.5	Use workaround
5. Peripheral Mode: SL811HS can miss packets in a noisy environment	X	1.5	Use workaround
6. Host/Peripheral Mode: Auto-increment feature results in corrupt data	X	1.5	Use workaround

1. Host Mode: SEO problem in low-speed hub operation

- **PROBLEM DEFINITION**

  Some hubs that send SE0s upstream during the EOF1 time frame may cause the SL811HS to stop sending SOFs. This problem occurs when operating with low-speed devices attached downstream of such a hub. This is not a problem with full-speed devices. According to the USB Specification, hubs are permitted to transmit SE0s during the EOF1 time frame. This is done to eliminate potential babble conditions on the bus and is an optional feature implemented in some hubs.

- **PARAMETERS AFFECTED**

  SOFs

- **TRIGGER CONDITION(S)**

  Attaching hub that sends SE0s upstream during the EOF1 time frame.

- **SCOPE OF IMPACT**

  The SL811HS can not host a low-speed device downstream of a hub that generates SE0s during EOF1.

- **WORKAROUND**

  The only complete workaround is to use a hub that does not transmit SE0s upstream during EOF1. Some hubs, including all Cypress hubs, have the option to disable SE0s from being generated during EOF1.

For a list of hubs that do not generate SE0s upstream during EOF1, or for more information on disabling this feature in Cypress hubs, contact Cypress USB support.

- **FIX STATUS**

  Use workaround.

2. Host Mode: Sync to SOF does not apply to low-speed mode

- **PROBLEM DEFINITION**

  The SYNC to SOF bit (bit 5) of the USB Host Control Registers [00H, 08H], is only designed for full-speed support. However, all other full-speed SOF bits and registers do apply to low-speed EOPs as well. In full-speed mode, this bit should only be used when the software cannot fit a packet within the remaining 1 ms frame. Setting this bit automatically delays sending the packet until the next SOF.

- **PARAMETERS AFFECTED**

  SYNC to SOF

- **TRIGGER CONDITION(S)**

  Full-speed support.

- **SCOPE OF IMPACT**

  If the SOF bit is set when operating in low-speed mode, packets may not get sent from the SL811HS.

- **WORKAROUND**

  Do not set the SOF bit when operating in low-speed mode. Instead, if a packet does not fit within the remaining 1 ms frame, firmware needs to delay sending it until after the next EOP. Using a simple delay loop or using the SOF timer interrupt (also EOP timer interrupt in low-speed mode) are two possible ways of doing this.

- **FIX STATUS**

  Use workaround.

3. Host/Peripheral Mode: 12 MHz operation with sensitive internal PLL

- **PROBLEM DEFINITION**

  The internal PLL is very sensitive. The PLL causes any high frequency noise on the V_{DD} pins to result in clock jitter.

- **PARAMETERS AFFECTED**

  USB data signaling at full-speed and improper timing of SOF packets.

- **TRIGGER CONDITION(S)**

  Operation at 12 MHz with high frequency noise on the V_{DD} pins.

- **SCOPE OF IMPACT**

  When operating the SL811HS at 12 MHz, high frequency noise on the V_{DD} pins can result in clock jitter. The clock jitter results in different symptoms depending on the severity of the jitter. Most notable is improper USB data signaling at full speed and improper timing of SOF packets.

- **WORKAROUND**

  The best workaround is to use 48 MHz to eliminate using the PLL. If 12 MHz is required, take these steps to reduce any jitter output of the PLL.

  1. Reduce high frequency noise on all SL811HS V_{DD} pins. This can be accomplished by adding proper decoupling capacitors directly on the V_{DD} pins. The value of 0.1 µF can be too large, depending on the inductivity of the traces on the PCB; experiment with values of 0.01 µF or even 1000 pF. In addition, ceramic capacitors are recommended.

  2. Use a 12 MHz oscillator instead of a crystal. An oscillator produces much sharper edge rates, which allow more tolerance for jitter.

  3. Careful layout can minimize this PLL jitter significantly:

  a. Use the shortest traces possible for decoupling capacitors.

  b. Use ground and VCC planes.

- **FIX STATUS**

  Use workaround.

[+] Feedback

4. **Peripheral Mode: Unreliable DMA interface**
   - **PROBLEM DEFINITION**
     The DMA interface can be unreliable in slave mode.
   - **PARAMETERS AFFECTED**
     DMA transfers to or from the SL811HS internal RAM.
   - **TRIGGER CONDITION(S)**
     Use of the DMA interface to move data to or from SL811HS internal RAM.
   - **SCOPE OF IMPACT**
     When performing DMA writes, data may get corrupted. This problem has only been seen for DMA write operations, but can also occur for read operations as well.
   - **WORKAROUND**
     Use the standard Data Port interface instead of the DMA interface for writing to or reading from the SL811HS RAM space. The DMA interface is not a recommended interface for the SL811HS due to this issue.
   - **FIX STATUS**
     Use workaround.
5. **Peripheral Mode: SL811HS can miss packets in a noisy environment**
   - **PROBLEM DEFINITION**
     In a noisy environment, the SL811HS has the potential to occasionally miss a packet. Occasionally missed packets are anticipated and dealt with in USB 2.0 Specification Section 10.2.6, where the following applies "It is recommended that the error count not be incremented when there is an error due to host specific reasons (buffer underrun or overrun), and that whenever a transaction does not encounter a transmission error, the error count be reset to zero." In other words if an individual packet is missed and the next packet is processed properly, the recommendation is that the error counter be reset to ‘0’. When drivers are written with this in mind, they can avoid issues that cause the transfer to be retired due to three errors in a transaction.
   - **PARAMETERS AFFECTED**
     Error count.
   - **TRIGGER CONDITION(S)**
     Electrically noisy environments.
   - **SCOPE OF IMPACT**
     If the SL811HS is used in an electrically noisy environment that may corrupt three requests within that transaction, the transaction will be retired by the host.
   - **WORKAROUND**
     1) The workaround for this issue is to write the driver according to the guidelines specified in section 10.2.6 of the USB 2.0 Specification, to prevent the driver from retrying the transfer.
     2) Board layout is the major reason for electrical noise that can aggravate this issue. When doing layout for the USB chip, use guidelines provided in a Cypress application note titled, *High-speed USB PCB Layout Recommendations* found on the Cypress web site.
   - **FIX STATUS**
     Use workaround.

[+] Feedback

6. Host/Peripheral Mode: Auto-increment feature results in corrupt data

- **PROBLEM DEFINITION**

  The SL811HS has a feature called auto-increment used to read or write blocks of the data buffer. This feature is used to speed up the time it takes to write blocks of data because an address location write is not required between data writes or reads. In some cases, the auto-increment feature can intermittently fail, causing the RAM location to be corrupted or the read buffer to provide incorrect data to the system processor. This type of error is very infrequent.

- **PARAMETERS AFFECTED**

  Any RAM location where auto-increment feature is used to access data. This includes both register and buffer space.

- **TRIGGER CONDITION(S)**

  Use of the auto-increment feature.

- **SCOPE OF IMPACT**

  When using the auto-increment feature for writes or reads, it is possible for the data to become corrupt. The following table demonstrates a typical error when it occurs. The error condition is shown in red.

  If an error occurs during writes using auto-increment, an address location can be written with the value of the previous address; each subsequent write will also be incorrect until the end of the block write. In the following example, note that the value of 0x01 from address 0x11 is incorrectly written to address 0x12 instead of the expected value of 0x02. After this error, the write to each subsequent address is also incorrectly written with the value that was intended to be in the previous address location.

  If an error occurs during a read using auto-increment, a single location can be incorrectly read as the previous addresses value. If the data is read again, it will show that the data in RAM is correct..

Auto-increment error during a write								
Address	**0x10**	**0x11**	**0x12**	**0x13**	**0x14**	**.....**	**0x1E**	**0x1F**
Data intended to be written to RAM	0x00	0x01	0x02	0x03	0x04	.....	0x0E	0x0F
Data actually written to RAM	0x00	0x01	**0x01**	**0x02**	**0x03**	**.....**	**0x0D**	**0x0E**
Auto-increment error during a read								
Address	**0x10**	**0x11**	**0x12**	**0x13**	**0x14**	**.....**	**0x1E**	**0x1F**
Data actually in RAM	0x00	0x01	0x02	0x03	0x04	.....	0x0E	0x0F
Data read back from RAM	0x00	0x01	**0x01**	0x03	0x04	.....	0x0E	0x0F

- **WORKAROUND**

  The easiest way to work around this issue is to not use the auto-increment feature. This affects performance because the address must be written prior to each write or read.

- **FIX STATUS**

  Use workaround.

References

1. SL811HS Embedded USB Host/Slave Controller Data Sheet (Document # 38-08008)

[+] Feedback

Document History Page

Document Title: Errata for SL811HS Embedded USB Host/Slave Controller Document Number: 38-17030				
Rev.	**ECN No.**	**Submission Date**	**Orig. of Change**	**Description of Change**
**	296874	See ECN	VCS	New Release
*A	3130821	01/07/2011	NMMA	Updated template Removed reference to SL811S/T data sheet in the References section

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com

© Cypress Semiconductor Corporation, 2005-2011. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Any Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.