SL811HS

Errata Revision: [**]

[December 23, 2004]
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. This document should be used to compare to the datasheet for this device to fully describe the device functionality.

Please contact your local Cypress Sales Representative if you have further 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 devices.

Note: Errata titles are hyperlinked. Click on table entry to jump to 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: 12MHz Operation with Sensitive Internal PLL	X	1.5	Use workaround
4. Peripheral Mode: 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 may result 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 in some instances when operating with low speed devices attached downstream of such a Hub. This is not a problem with full speed devices. According to USB spec, 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, such as all of 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, please 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 can not fit a packet within the remaining 1ms frame. Setting this bit will automatically delay 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 doesn't fit within the remaining 1ms 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: 12MHz Operation with Sensitive Internal PLL

- **PROBLEM DEFINITION**

  The internal PLL is very sensitive. The PLL will cause any high frequency noise on the VDD 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 12MHz with high frequency noise on the VDD pins.

- **SCOPE OF IMPACT**

  When operating the SL811HS at 12MHz, high frequency noise on the VDD pins could result in clock jitter. The clock jitter could result in a number of different symptoms depending on the severity of the jitter. Most notably will be improper USB data signaling at full speed and improper timing of SOF packets.

- **WORKAROUND**

  The best workaround is to use 48MHz to eliminate using the PLL. If 12MHz is required there are a number of things that can be done to reduce any jitter output of the PLL.

  1. Reduce high frequency noise on all SL811HS VDD pins. This can be accomplished by adding proper decoupling capacitors directly on the VDD pins. The value of 0.1uF might be too large, depending on the inductivity of the traces on the PCB and values of 0.01uF or even 1000pF should be experimented with. In addition ceramic capacitors are recommended.

  2. Use a 12MHz oscillator rather then a crystal. An oscillator produces much sharper edge rates, which will 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.

4. Peripheral Mode: 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)**

  Usage of the DMA interface for moving 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 may 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 language 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 will cause the transfer to be retired due to 3 errors in a transaction. This is discussed in detail in a Cypress application note titled *USB Error Handling for Electrically Noisy Environments* found on the Cypress web site.

  - **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 correct work around 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 exacerbate 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.

6. Host/Peripheral Mode: Auto-increment feature may result 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)**

Usage 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 be 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 and thus each subsequent write will also be incorrect until the end of the block write. Note in the following example the value of 0x01 from address 0x11 was incorrectly written to address 0x12 instead of the value 0x02 as expected. 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 the data would be correct and show that the data in RAM was 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 will cause a performance degradation due to the fact that 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)
2. SL811S/T USB Dual Speed Slave Controller Data Sheet (Document # 38-08009)

Document History Page

Document Title: Errata for SL811HS Embedded USB Host/Slave Controller Document Number: 38-17030				
REV.	**ECN NO.**	**Issue Date**	**Orig. of Change**	**Description of Change**
**	296874	See ECN	VCS	New Release