SL811HS/SL811S Errata

Embedded USB Host/Slave Controller

Cypress Semiconductor Corporation 3901 North First Street San Jose CA 95134 408-943-2600

April 04, 2002

Glossary of Terms

Below is a list of sections that may be found under each issue discussed. If any one of the following sections isn't listed then it doesn't apply.

Background: Any required background knowledge of the subject will be listed here. Things that may be found in this category include, USB spec quotes, history behind the feature change or issue, etc...

Reference#: Internal reference number for the issue.

Category: Here the issue is categorized as a bug, feature change, or other.

Revision: The revision section lists the revision of the chip that the issue applies to.

Description: Provides a descriptive summary of the issue.

Cause: Lists the cause of the issue, problem, or feature.

Implication: Describes the implication of the issue, problem, or feature.

Workaround: If any workarounds are possible they will be listed in this field.

Status: Describes the state of the issue. If a problem has been resolved, the revision of silicon with the fix will be listed here.

Table 1. Quick Reference Table

#	Issue Title	Chip Rev Number 1.2	1.3	1.4	1.5
1	USB A/B Buffer Toggle Problem	X			
2	Speed Switching Problem in Low Speed Hub Operation	X	X		
3	SE1 (Single-end 1)	X	X		
4	Missing SETUP/IN/OUT packet around SOF generation	X	X		
5	Missing ACK during SETUP packet	X	X		
6	No EOP Interrupt for low speed	X	X		
7	Can't recognize PID_IN packet when multiple devices are connected			X	
8	12MHz		X	X	
9	X2 clock was not disable during the suspend mode			X	
10	Babble detection bit (bit 2) in the Interrupt Status Reg. (0DH) has been removed				X
11	Crossover voltage in Transceiver is more than 2v			X	X
12	Series resistors for data lines		X	X	X
13	SE0 problem in Low Speed Hub Operation			X	X
14	Sync to SOF does not apply to low speed mode	X	X	X	X

Note: If chip revision number box is marked then the issue applies, otherwise ignore.

1. Host Mode: USB A/B Buffer Toggle Problem

Category: Bug

Revision: Revision 1.2

Description: The SL811HS operation of USB A and B Bank toggle is sometimes compromised when operating the SL811HS with auto SOF generation enabled. An error occurs which results in the ARM bit failing to set.

Cause: Internal logic error

Implication: When the ARM bit isn't set USB communication will stop.

Workaround: The recommended software workaround is to only use the USB_A bank or the USB_B bank and never use both banks at the same time. It is still possible to do ping-pong operation with a single bank, either A or B.

Status: Fixed in Revision 1.3 and beyond.

2. Host Mode: Speed Switching Problem in Low Speed Hub Operation

Category: Bug

Revision: Revision 1.2 and 1.3

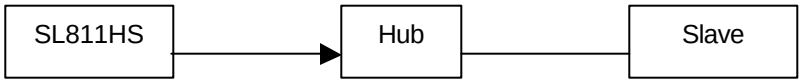
Description: The SL811HS generates erroneous packets after it tries to communicate with low speed devices through USB Hub.

In some instances when operating with low speed devices attached to a Hub, the SL811HS fails to switch back to Full Speed at the end of packet time (EOP). This results in the next tokens being transmitted erroneously at Low Speed.

"Erroneous packets" are low speed token packets,"3C". The SL811HS starts generating them after sending a low speed ACK over USB Hub. This problem occurs for low speed PID_IN transactions only.

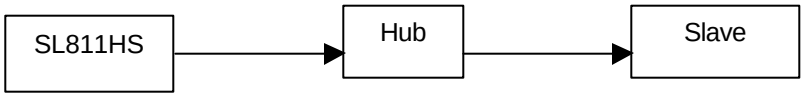
1) Sending Full Speed-PRE packet.

PRE



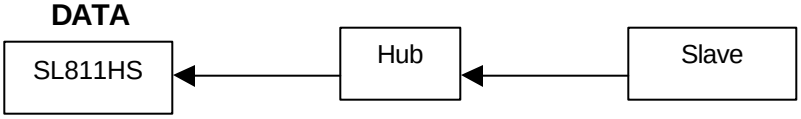
2) Sending Low Speed -IN packet.

Low Speed-IN



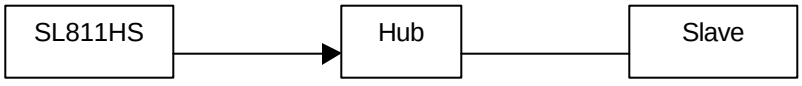
3) Receiving Low Speed -DATA packet.

DATA



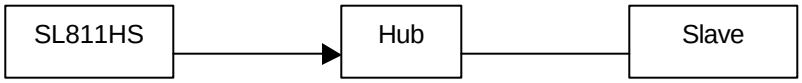
4) Sending Full Speed -PRE packet.

PRE



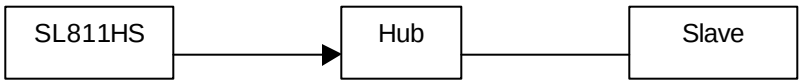
5) Sending Low Speed -ACK packet.

ACK



6) Sending Bad packets.

Low Speed -"3C"



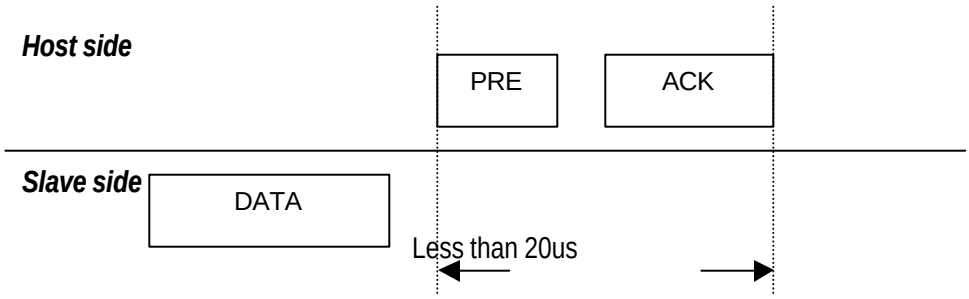
At step, (6), SL811HS will generate the bad packets constantly.

Cause: Internal logic error.

Implication: The SL811HS generates erroneous packets after trying to communicate with low speed devices through a USB Hub.

Workaround: In order to stop the generation of bad packets and to recover the SL811HS' internal logic, developer should generate an additional Full Speed PID_IN packet after sending the Low Speed ACK packet.

Full Speed PID_IN packets must be generated right after Low Speed ACKs to prevent the generation of bad packets. Please see the timing chart below.



The USB_Done interrupt is set after receiving the DATA packet requested by PID_IN packet. There is about a 16-17 us delay time between USB_Done interrupt and the end of the Low Speed ACK packet.

Developer should prepare a Full Speed PID_IN packet and set arm bit within 16-17us.

Here is the sample code:

This code should be placed in an interrupt handler for USB_Done.

Revision 1.2:

```
#ifdef REV12
result=SL11Read(EP0Status);
if (pid==PID_IN && !(result&0x40)) (*See Note 1)
{
   SL11Write(IntStatus,0xff);
   SL11Write(EP0Control,1); (*See Note 2 )
   Delayms(1,1);
}
#endif
```

*1… In case of NAK condition, SL811HS will not generate low speed NAK. This mean there is no issue on the USB Bus.

*2… For quick response, A developer can skip preparations of Full Speed PID_IN because LOW SPEED PID_IN has the same values in register 03H and 04H.

Revision 1.3:

```
#ifdef REV13
   if (dbug && pid==PID_IN) // low speed work
arround
   {
      SL11Write(EP1Status, 0xc0); // Insert PRE
Token
      SL11Write(EP1XferLen, 0);   // Zero len
   }
#endif
```

* For quick response, developer can skip preparations of Full Speed PRE packet.

Status: This problem has been fixed in revision 1.4 and beyond.

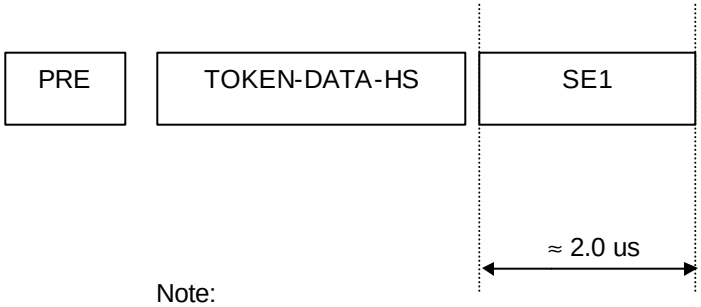
3. Host Mode: SE1 (Single-end 1)

Background: The SE1 has been defined in the USB2.0 as both D+ and D- being at a voltage above V_{OSE1} min, which is 0.8V.

Category: Bug

Revision: Revision 1.2 and 1.3

Description: SL811HS generated a SE1 condition after the PRE Token is enabled. This corrupts SOF generation.



Note:
- TOKEN: SETUP/IN/OUT
- DATA: Data Payload
- HS: Hand Shaking (ACK/NAK/STALL)

Cause: Internal logic error

Implication: SE1 is an illegal condition defined by the USB spec.

Workaround: There is no software work-around. The SE1 causes slower USB transfer rate after a HUB.

Status: This problem has been fixed in revision 1.4 and beyond.

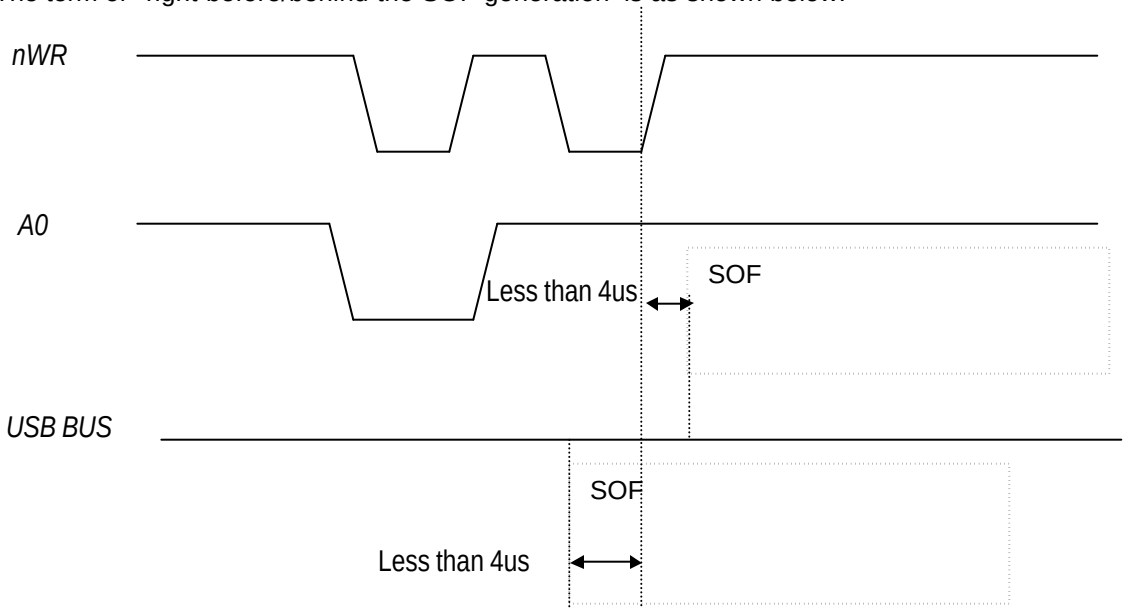
4. Host Mode: Missing SETUP/IN/OUT packet around SOF generation

Category: Bug

Revision: Revision 1.2 and 1.3

Description: Sometimes a prepared packet in the SL811HS buffer will not go out if the Arm command is written to the register 00H/08H right before/behind the SOF generation.

The term of “right before/behind the SOF generation” is as shown below.



If nWR strobe is asserted with enough margin (greater than 4us between nWR and SOF), this problem does not occur.

Cause: Internal logic error

Implication: When the problem does occur the next scheduled packet will not be generated, so developers need to re-arm for the next packet to be sent.

Workaround: Normally, writing the Arm command right before/behind the SOF will cause a “Babble” condition. Developers should track the SOF counter value to avoid “Babble” condition.

The SL811HS has a “SOF” bit in the register 00H/08H which when set, prevents a packet from overriding the HW SOF generation. However, Developers must not write any Arm command right before or close to the SOF generation.

Basic concepts of workaround are:

1) Developers must not write the Arm command if the SOF counter value is too small.
2) Developers are recommended to check if a previous packet has completed with USB_Done interrupt.

Here is the sample code:

```
while( loop-- )
{       short sofc;

        int_cnt++;
        sofc = SL11Read(0xf);
        SL11Write(IntStatus,0xff);
        if (sofc > (BYTE)pl)
                SL11Write(EP0Control,Cmd); // Enable ARM
        else if (sofc > 1)
                SL11Write(EP0Control,Cmd|0x20);
                // Enable ARM and Wait after SOF
        else
        {
                while (SL11Read(0xf)<2) {};
                // Wait for
                SL11Write(EP0Control,Cmd|0x20);
                // Enable ARM and Wait after SOF
        }
        Delay1ms(2,0);
        // just for example only.  If CPU uses the
Interrupt
        // should check the interrupt variable "int_cnt"
        if ((SL11Read(IntStatus)&1)==0)
            printf("Missing Interrupt\n");
        else int_cnt--;
}
if (int_cnt)

        printf("Missing interrupt\n");
```

1) The code selects transfer method based on the SOF counter.
If (the SOF counter indicates enough bandwidth), then arm.
If (the SOF counter > 1), then arm with SOF bit.
If (the SOF counter == 0), then wait.

Status: This problem has been fixed in revision 1.4 and beyond

5. Slave Mode: Missing ACK during SETUP packet

Background: By USB specification 1.1 definition, Inter-packet delays are measured from the SE0-to-J transition at the end of the EOP to the J-to-K transition that starts the next packet.

The host must provide at least two bit times of J after the SE0 and the start of frame of a new packet.

Category: Bug

Revision: Revision 1.2 and 1.3

Description: On some OHCI controllers, the SL811S fails to return an ACK after the host sends a SETUP packet.

PC	USB Chip	TIPD	Result
Freeway	Intel(UHCI)	3 bits	OK
Dynabook	NEC(OHCI)	5 bits	OK
Fujitsu	Acer (OHCI)	2 bits	NG
NEC	Acer (OHCI)	2 bits	NG
Sotec	Sis (OHCI)	2 bits	OK
USB card	Opti (OHCI)	2 bits	OK
IMAC	Opti (OHCI)	2 bits	NG

Cause: Inter-packet delay.

Implication: A timeout will occur and the host controller may not continue communication with that SL811S endpoint.

Workaround: There is no software workaround for USB OHCI control with an inter-packet delay of 2 bits.

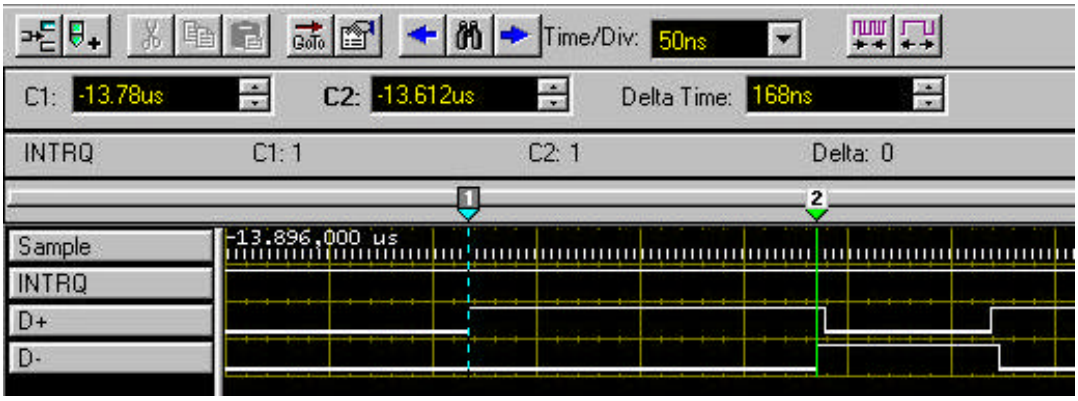


Figure 1: 2-bit inter-packet delay from Opti (OHCI)

Status: This problem has been fixed in Revision 1.4 and beyond

6. Slow speed Slave Mode: No EOP Interrupt for low speed

Background: USB hosts generate EOP packets every 1ms as a keep-alive for low speed USB device. The issue concerns low speed operation when it goes into suspend/resume state.

Category: Bug

Revision: Revision 1.2 and 1.3

Description: SL811S does not generate an interrupt for keep-alive packets (or EOP) generate by a USB Host.

Cause: Function not implemented in revision 1.2 and 1.3

Implication: When operating in low speed, an idle bus state cannot be monitored to determine when to enter suspend.

Workaround: No software workaround.

Status: This problem has been fixed in Revision 1.4 and beyond

7. Slave Mode: Can't recognize PID_IN packet when multiple devices are connected.

Category: Bug

Revision: Revision 1.4

Description: SL811S does not recognize the PID_IN packet when there are PID_OUT packets that are sent to other USB Devices connected on the same Root HUB or HUB.

Cause: This bug was introduced during the release from Revision 1.3 to Revision 1.4 due to the net-list editing.

It has been verified in the ModelSim simulation, which confirms this bug existed. The ModelSim showed that if either a PID_SETUP or PID_OUT is sent to other devices on the same USB Bus, this will cause the SL811S to hang on the next PID_IN package that is sent.

For example of the failure:
PID_OUT sent to device 2. PID_IN sent to device 3 (3 is the SL811S), then the PID_IN will be ignored by the SL811S.

Note:
This problem will not happened if only one device (i.e. SL811S) is connected to the Root-HUB or the HUB.

This problem will not occur on the PID_OUT, PID_SETUP.

For example:
PID_OUT sent to device 2. PID_OUT or PID_SETUP sent to device 3 (3 is the SL811S), then the PID_OUT for PID_SETUP will be OK for the SL811S.

Implication: This will cause the SL811HS slave mode to hang.

Workaround: Application can retry any command that uses the PID_IN. For example "ReadFile" command might need to retry couple time incase of failure on reading the data.

Note: ReadFile must be retried within 3ms, which should allow at least 1 SOF packet to clear the error condition.

Status: This problem has been fixed in Revision 1.5.

8. Slave/Host Mode: X2 clock was not disable during the suspend mode.

Category: Feature change

Revision: Revision 1.4

Description: The X2 should be disable during the suspend mode to reduce power consumption. This will be applied in both Host and Slave mode.

Implication: If X2 is enabled during suspend more power will be consumed.

Status: X2 is disabled during suspend in revision 1.5

9. Host/Slave Mode: 12MHz

Category: Bug

Revision: Revision 1.3 and 1.4

Description: History of clocking problems.

Silicon rev 1.2: No problem.
Silicon rev 1.3: Can not support 12MHz clock must use 48MHz.
Silicon rev 1.4: Failure rate of using clock 12MHz is 1%
Silicon rev 1.5: Fixed this problem.

Implication: The SL811HS will not operate.

Status: This problem has been fixed in revision 1.5.

10. Host Mode: Babble detection bit (bit 2) in the Interrupt Status Reg. (0DH) has been removed.

Category: Feature change

Revision: Revision 1.5

Description: Revision 1.5 removed the Babble detection due to the gate level net-list editing to save gate count. The babble detection bit2 in register 0DH is no longer supported. Any read from bit2 returns zero.

Status: The Babble detection bit is no longer supported in revision 1.5.

11. Slave Mode: Crossover voltage in Transceiver is more than 2v.

Category: Bug

Revision: Revision 1.4 and 1.5

Description: The Crossover voltage in the transceiver is more than 2V. The USB2.0 spec requires that the crossover voltage be under 2V.

Measurement on the D+ and D- during any IN packet (i.e. Host Send PID_IN) or the SL811HS send ACK or NAK packet to the host.

Note: There was an improvement from Revision 1.4 to Revision 1.5. However the Revision 1.5 will only fail at 3.6v.

Workaround: To balance the crossover voltage, a 12pF capacitor connected from D- to GND at the USB connector is required. See figure 2 below.

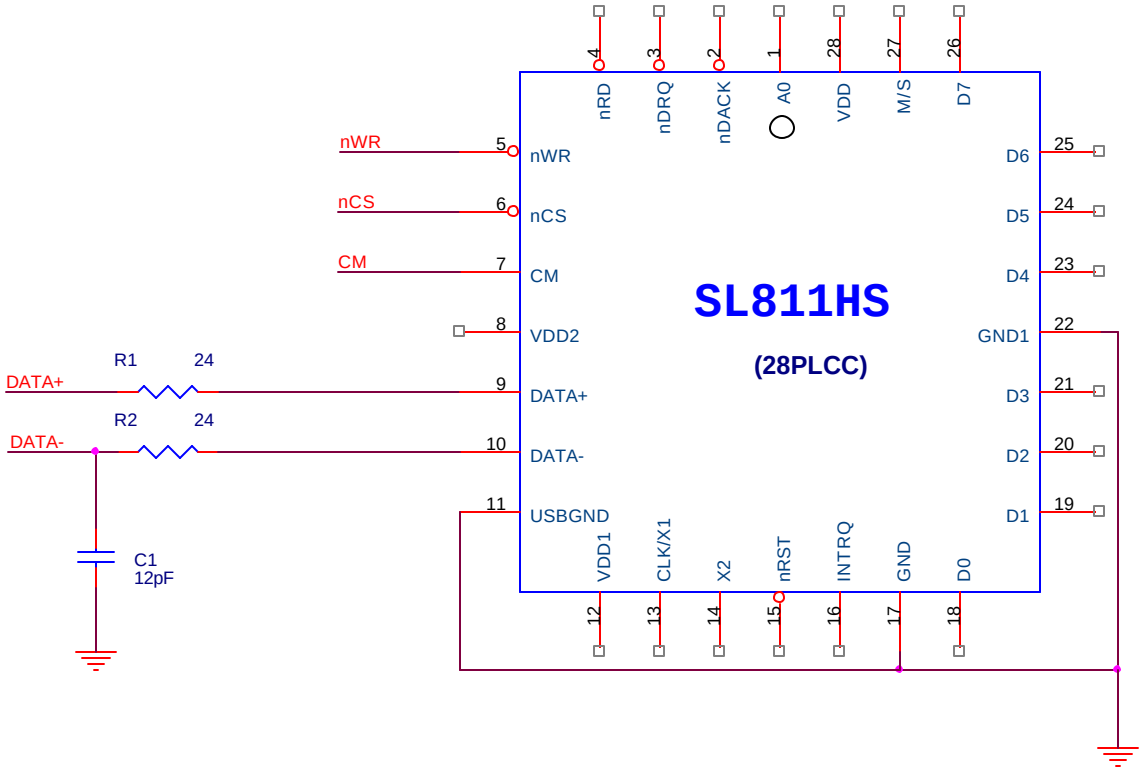


Figure 2: Series resistors and correction capacitor

Status: Under investigation.

12. Slave Mode: Series resistors for data lines

Category: Feature change

Revision: Revision 1.3 and beyond

Description: Up to (and including) silicon rev 1.2, the output impedance of the transceivers (D+, D-) was 36 to 42 ohms. This required series resistors (R1 and R2 in figure 2) of 33 ohms.

Revision 1.3 and on have different transceivers. These transceivers are rated at 12 to 22 ohms output impedance. Series resistors have to be in the 18 to 24 ohms range for optimal impedance match.

Cause: Different internal impedance of the transceivers.

13. Host Mode: SE0 problem in Low Speed Hub Operation

Background: 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.

Category: Bug

Revision: Revision 1.4, 1.5

Description: 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.

Cause: The SE0 will cause the SIE receiving clock to be halted.

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.

Note: Hub support is not required for OTG dual role devices (DRDs).

Status: Closed

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

Category: Clarification

Revision: Revision 1.2, 1.3, 1.4, 1.5

Description: 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.

Implication: 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 pulling the SOF Timer interrupt (also EOP Timer interrupt in low speed mode) are two possible ways of doing this.

REVISION HISTORY

Name and Revision	Date Issue	Description	Author/Revise
Errata 1.2	03/27/2001	Created for chip Revision 1.2	Bernie
Revised	6/5/2001	Silicon 1.3 (Chip is not released)	TBN
Revised	07/24/01	Update Errata 1.2 &1.3	TBN
Revised	07/24/01	Silicon 1.4	TBN
Errata 1.4	09/17/01	Bugs in Rev 1.4	SBN
Errata 1.5	11/02/01	Silicon 1.5	RL
Errata 1.5	11/14/01	Silicon 1.5	SBN
Errata 1.6	04/04/02	Added SOF SYNC Issue 14, modified Errata Item 13, and reformatted.	MUL