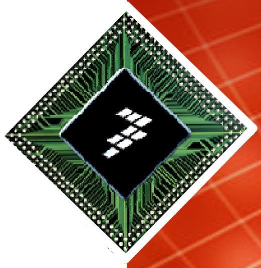
freescale™

Notification # 16317

USB Errata Information



Confidential and Proprietary

USB A-003845: Frame scheduling robustness-Host may issue token too close to uframe boundary

- **Devices: MPC837x, MPC8314/15, P1010/14 Rev 1.0 & 2.0, P1022/13 & P102x Family**
- **Description:** When the USB host encounters an under-run while sending a Bulk OUT packet, it issues a CRC error according to the specification. However, the retry never occurs on the USB and the host appears to hang; it does not send any further transactions including SOF packets. The device ultimately detects a suspend condition and defaults to full speed mode. This can also happen for IN transactions where the device encountered an under-run and sent BAD CRC. The host will retry in this case without checking for the time left in the current uFrame. The response from the device will cause frame babble in this case.
- **Impact: The host appears to be hang as it does not send any further packets.**
- System Impact: The host port is disabled, the halt bit is set, and the frame babble error is set in the associated data structure. This behavior complies with the EHCI specification for handling a frame babble error. The host controller driver handles the recovery by clearing the error conditions and re-queuing the transfer which should occur normally. When a frame babble occurs, there may be a loss of bandwidth because the application has to intervene and requeue the transfer and re-enable the port.
- **Workaround:** For OUT transactions: If the host controller TX under-runs can be avoided then the problem will not occur for OUT transaction. Using a larger value for TXSCHOH can avoid this issue for OUT transactions.
- For IN transactions: Insure the USB device side does not into an under-run condition. There is no workaround from the host side. The host controller driver (software) should handle the recovery by clearing the error conditions and re-queuing the transfer which should occur normally and re-enabling the port. The software driver can get the system restarted in this case. However, it cannot prevent the frame babble from occurring.
- **Fix plan: No plans to fix**

freescale™

USB A-003829: Host detects frame babble but does not halt the port or generate an interrupt (Page 1 of 2)

- **Devices: MPC837x, MPC8314/15, P1010/14 Rev 1.0 & 2.0, and P1020 Family**
- **Description:** A high speed ISO Device, connected downstream to a high speed hub connected to the USB host, babbled in to the uframe boundary EOF1 time and the hub disabled the propagation of traffic to the pstream root host. Inside the host controller, the ehci_ctrl state machine issues a request to the protocol engine to initiate the next transaction but this transaction is not sent to the USB as the port enable bit has been cleared. The result is that the ehci_crl state machine waits for the transaction to complete (which does not occur).
- Eventually the software application times out without any frame babble error information. Just the iTD transaction error is issued. The failure was seen with a high speed ISO device but a device babble error could occur on bulk or control or interrupt transactions for a failing device.
- The final state is that the port has transitioned to full speed and the port enable bit is de-asserted while the DMA does not know that there is a problem and the data structure shows only the occurrence of a transaction error.
- This bug can occur for host IN transaction where the sending device under runs and error injects on the data packet. In this case the host may retry the IN immediately and it does not consider that the protocol engine may not be ready to issue the IN to the USB. The protocol engine issues the IN up to 5 useconds later than the EHCI Control state machine has issued the request to send the packet so it could result in a frame babble.

USB A-003829: Host detects frame babble but does not halt the port or generate an interrupt (Page 2 of 2)

- **Impact:** The host port is disabled, the halt bit is set, and the frame babble error is set in the associated data structure. This behavior complies with the EHCI specification for handling a frame babble error. The host controller driver handles the recovery by clearing the error conditions and re-queuing the transfer which should occur normally. When a frame babble occurs, there may be a loss of bandwidth because the application has to intervene and re-queue the transfer and re-enable the port.
- **Workaround:** There is no workaround because the control of the retry is under hardware control so for non ISO transfers the hardware will retry so long as it determines that there is enough time but it
- does not account for the added delay due to the host protocol state machine being in the bus timeout state. Having a large TX FIFO and a good fill level (TXFIFOTHRES) will mean that there will be no under runs to host OUT transactions. This will significantly reduce the probability of occurrence of this issue for OUT Transactions. However please note that this bug can occur for host IN transaction where the sending device under runs and error injects on the data packet. In this case the host may retry the IN immediately.
- Recovery:
- The host will not get any response. The recovery from this condition will depend on the software, but host it will eventually time out and reset the device. This is not critical as this issue will only occur if the device downstream of a HS HUB is out of spec. and generates a frame babble.
- **Fix plan:** No plans to fix

freescale™