Freescale Semiconductor
Device Errata

MSE5121E_0M36P
Rev. 3, 12/2011

MPC5121e (M36P) Errata

by: Microcontroller Division

This document identifies implementation differences between the MPC5121e, M36P (Rev. 1), and the processor's description contained in the *MPC5121e Microcontroller Reference Manual* and the *MPC5121e Microcontroller Data Sheet*. Refer to http://www.freescale.com for the latest updates.

Contents

1 Overview

Table 1. MPC5121e Errata Summary

Module	ID	Date Added	Title
Clock	421	02/11/2009	Section 2.1, Fractional divider cannot support ratio change from 8 to 12
e300 Core	467	12/9/2008	Section 3.1, e300 — DTLB (Data Translation Lookaside Buffer) LRU performance issue
	ERR 003383	12/01/2011	Section 3.2, If critical interrupt and normal interrupt are used in a system, the e300 core may hang
ESD	473	02/11/2009	Section 4.1, ESD CDM — PCI pins do not meet the 500 V specification
	474	02/19/2009	Section 4.2, ESD HBM — USB V_{DDA} power supply pins do not meet the 2 kV specification
I/O	425	10/8/2008	Section 5.1, Pull-Up resistors are missing on PSC_3_4, PSC4_4, and PSC5_4
NFC	367	10/8/2008	Section 6.1, NFC RAM buffer access limitation
NFC	510	12/01/2011	Section 6.2, 8-bit ECC mode does not work with 16-bit wide NAND flash
PCI	30	08/14/2007	Section 7.1, STOP assertion by target device on last beat of PCI memory write transaction can cause a hang
	71	08/14/2007	Section 7.2, Data corruption by PCI_DMA when using destination address hold (DAHE)
	257	08/14/2007	Section 7.3, PCI controller hangs when returning from PCI pins low status
PMC	364	10/8/2008	Section 8.1, PMC deasserts wakeup interrupt automatically after e300 core is awake
RTC	471	02/02/2009 (updated)	Section 9.1, The Tamper (TAMP) bit in the SRTC is not reliable
	509	01/21/2010	Section 9.2, RTC Oscillator may not start properly
SATA	337	11/6/2007	Section 10.1, SATA module is non-functional
TEMPSENSE	393	02/11/2009	Section 11.1, Temperature sensor does not provide specified accuracy
USB	42	08/14/2007	Section 12.1, Read of PERIODICLISTBASE after successive writes may return wrong data
	410	10/8/2008	Section 12.2, Priming ISO over SOF creates a bad packet with correct CRC
	411	10/8/2008	Section 12.3, Port Change interrupt not generated
	465	02/11/2009	Section 12.4, The USB controller can issue transactions that lock up the AHB bus under certain conditions
	477	08/04/2010	Section 12.5, In Host mode, an RX FIFO overflow can generate a false babble error
	478	08/04/2010	Section 12.6, No interrupt generated on error condition in ISO Mult 3 mode
	479	08/04/2010	Section 12.7, Device ISO IN transactions underrun on ISO Mult 3 transfers
	480	03/17/2010	Section 12.8, Port change detect bit is set when a resume is forced by SW write to FPR bit in PORTSC register
	481	08/04/2010	Section 12.9, Incorrect value read from TXPBURST/RXPBURST
	497	08/04/2010	Section 12.10, EHCI NYET Response to a Complete Split Non-compliance
	498	08/04/2010	Section 12.11, CERR counter not reset during a ACK or NAK response to a PING
	499	08/04/2010	Section 12.12, RX Endpoint Flush when receiving a token OUT

Table 1. MPC5121e Errata Summary (continued)

Module	ID	Date Added	Title
USB	501	03/17/2010	Section 12.13, OTG Controller as Host does not support Data-line Pulsing SRP
	503	03/17/2010	Section 12.14, Device does not respond to INs after receiving corrupted handshake from previous IN transaction
	504	03/17/2010	Section 12.15, Last read of the current dTD done after USB interrupt

2 Clock

2.1 Fractional divider cannot support ratio change from 8 to 12

2.1.1 Description

Once the SCFR1[DIU_DIV] and the SCFR2[SDHC_DIV] bits are programmed to a divide ratio of 2:1 (CSB_CLK:DIU_CLK or CSB_CLK:SDHC_CLK), the frequency divider cannot be programmed to divide by any other divide ratio. That is, the divide ratio cannot be changed once the divide ratio is programmed to 2:1.

2.1.2 Workaround

When selecting the clock frequencies for the DIU and the SDHC, if the clocks must be dynamically changed, be sure not to select a clock ratio of 2:1.

3 e300 Core

3.1 e300 — DTLB (Data Translation Lookaside Buffer) LRU performance issue

3.1.1 Description

A performance degradation may occur for the e300 core under some specific load and store memory copy operations. This issue is related to performance only and the core always produces the correct and expected result. The issue relates to LRU during a DTLB miss with a load and a store operation within a 128 KB range (the EA[15:19] DTLB index) relationship between the source and the destination of a memory to memory copy. If the load *and* the store operations (going to a different address) are such that the EA[15:19] remain unchanged, then the condition exists, and the DTLB LRU remains pointing to a single way.

The effect of the issue is that the DTLB only uses a single way entry for all operations. This results in the Data TLB DSI exception being taken for *every* load and store within the range, not just the exception needed to load the initial page.

3.1.2 Workaround

The software workaround is to implement a Least Recently Written (LRW) in software, since the Least Recently Used (LRU) answer from the hardware will always point to way 1 after the first entry. Currently, when a Data Translation Miss exception occurs, the SRR1 bit 14 (DTLB replacement way) is automatically set to indicate the way selected for replacement. However, this can be overwritten by altering that bit with an LRW setting in software. The DTLB implementation is a 2-way set associative implementation with 32 entries per way. This means one word (32 bits) can be kept in software to manage the LRW settings on every new load to the DTLB. The LRU hardware algorithm is changed to an LRW software algorithm.

3.2 If critical interrupt and normal interrupt are used in a system, the e300 core may hang

3.2.1 Description

If critical interrupt and normal interrupt are being used in a system, the e300 core may hang.

3.2.2 Workaround

Instead of using an RFI at the end of every exception handler, replace the RFI with the following:

Disable critical interrupts by setting MSR[CE] to 0 with **mtspr**.

Copy SRR0 and SRR1 to CSRR0 and CSRR1, respectively.

Execute an RFCI. This enables MSR[CE] and any other bits that original RFI would have set, including MSR[EE]

Sample code:

```
// Disable MSR[CE]
mfmsr   r2
lis     r3, 0xffff
ori     r3, r3, 0xff7f
  and     r2, r2, r3
  sync
  mtmsr   r2
  isync

// Copy SRR0, SRR1 to CSRR0 and CSRR1
  mfspr   r2, srr0
  mfspr   r3, srr1
  mtspr   csrr0, r2
  mtspr   csrr1, r3

… restore GPRs

rfci
```

4 ESD

4.1 ESD CDM — PCI pins do not meet the 500 V specification

4.1.1 Description

The PCI pins do not meet the 500 V ESD CDM AEC specification.

4.1.2 Workaround

Be sure that the manufacturing system used for placing this device has its ESD maximum well below the 500 V CDM specification.

4.2 ESD HBM — USB V_{DDA} power supply pins do not meet the 2 kV specification

4.2.1 Description

The V_{DDA} power supply pins (C21, D20) do not meet the 2 kV AEC specification. These pins pass at 1.5 kV.

4.2.2 Workaround

If ESD protection greater than 1.5 kV is required on these pins, add external circuitry to protect these pins.

5 I/O

5.1 Pull-Up resistors are missing on PSC_3_4, PSC4_4, and PSC5_4

5.1.1 Description

Pull-up resistors are missing on PSC_3_4, PSC4_4 and PSC5_4. An internal pull-up resistor should prevent an external NAND flash chip or any chip connected to LPC_CS6 from driving the EMB_AD lines during the PORESET cycle or during boot time.

The reset configuration latching and the boot up instruction could be corrupted.

5.1.2 Workaround

Use external pull-up resistors on PSC3_4 (LP_CS6), PSC4_4 (NFC_CE2), and PSC5_4 (NFC_CE3) to prevent NAND flash chips or any chip connected to LPC_CS6 from driving the reset configuration word during the PORESET sequence. Pull-ups are required if NFC_CE2, NFC_CE3, or LPC_CS6 are used.

6 NFC

6.1 NFC RAM buffer access limitation

6.1.1 Description

Accessing the NFC RAM with the e300 core or DMA engine during an external NAND flash data transfer (FDI/FDO command) can lead to loss of (write) data or wrong (read) data.

6.1.2 Workaround

- Do not access NFC RAM during FDI/FDI command execution.
- Do not start FDO/FDI commands until DMA read/write RAM buffer is finished.

6.2 8-bit ECC mode does not work with 16-bit wide NAND flash

6.2.1 Description

The NAND flash controller does not correctly detect or generate 8-bit ECC codes for 16-bit wide NAND flash memory. Therefore, the following boot configurations are invalid:

- RST_CONF_ROMLOC[1:0] (EMB_AD[1:0]) = 11. (boot from NAND)
- RST_CONF_NFC_DBW (EMB_AD[21]) = 1 (16-bit data bus)
- RST_CONF_NFC_PS (EMB_AD[20]) = 1. (4 KB page, 218 byte spare, 8-bit ECC)

It is also not possible to configure 8-bit ECC mode with 16-bit NAND flash memory after boot using the NAND Flash Operation Configuration (NF_CFG1) register's ECC_MODE field set to 0.

6.2.2 Workaround

8-bit ECC mode must be avoided for 16-bit NAND flash devices. 4-bit ECC mode may be used with 16-bit NAND flash memory. 4-bit and 8-bit ECC modes may be used with 8-bit NAND flash memory devices. If not booting from the NAND flash memory device, hardware ECC may be bypassed and calculated in software.

7 PCI

7.1 STOP assertion by target device on last beat of PCI memory write transaction can cause a hang

7.1.1 Description

As a master, the PCI Controller can combine a memory write to the last PCI double word (4 bytes) of a 32-byte cache line with a 4-byte memory write to the first double word of the subsequent cache line. This only occurs if the second memory write arrives at the PCI controller before the de-assertion of PCI_FRAME for the first write transaction. If the writes are combined, the PCI controller masters a single memory write transaction on the PCI bus. If, for this transaction, the PCI target asserts PCI_STOP during the last data beat of the transaction (PCI_FRAME is de-asserted, but PCI_TRDY and PCI_IRDY are asserted), the transaction completes correctly. A subsequent write transaction other than an 8-byte write transaction causes a hang on the bus. Two different hang conditions can occur:

- If the target disconnects with data on the first beat of this last write transaction, the PCI controller de-asserts PCI_IRDY on the same cycle as it de-asserts PCI_FRAME (PCI protocol violation), and no more transactions are mastered by the PCI controller.
- If the target does not disconnect with data on the first beat of the last write transaction, PCI_IRDY is de-asserted after the first beat is transferred and is not asserted any more after that, causing a hang.

This affects 32-bit PCI target devices that arbitrarily assert PCI_STOP on memory write transactions without detecting that the data beat being transferred is the last data beat of the transaction. If the PCI transaction is a one data beat transaction and the target asserts PCI_STOP during the transfer of that beat, no impact occurs.

7.1.2 Workaround

7.1.2.1 Hardware Workaround

A PCI target device could avoid asserting PCI_STOP during the last beat of a PCI memory write transaction greater than one data beat and crosses a cache line boundary. It could assert PCI_STOP during the last data beat of the 32-byte cacheline or not assert PCI_STOP at all.

7.1.2.2 Software Workaround

Set the PCI latency timer register (offset 0x0D) to zero. A value of zero is the reset value for this register. Keeping this register unmodified after reset prevents the PCI controller from ever combining writes.

7.2 Data corruption by PCI_DMA when using destination address hold (DAHE)

7.2.1 Description

There can be corruption of the PCI_DMA data under the following conditions:

- DMAMR[DAHE] = 0b1 (destination address hold)
- DMAMR[DAHTS] = 0b10 (4 bytes) or 0b11 (8 bytes)
- PCI_DMA source address is not aligned to the transaction size specified by DAHTS
- The source port width is smaller than the destination transaction size OR the source port returns valid read data only in the valid byte lanes
- An access to a PCI_DMA register happens during the transfer

Example of error condition:

- DAHTS is 8 bytes and the source port is a 32-bit PCI bus
- The source memory space is on the PCI bus and is not prefetchable

7.2.2 Workaround

Use a source address aligned to the destination transaction size.

7.3 PCI controller hangs when returning from PCI pins low status

7.3.1 Description

During the inactivity period (low power) of the PCI bus, the PCI pins are pulled low by setting the GCR[PPL] bit.

The PCI controller state machine remains active during this period and is actively tracking the bus signals. It assumes the external PCI pins are meeting the AC timing specifications.

When GCR[PPL] =0b0, the bus signals start to rise to 1 logic, pulled by the bus pull-up resistors.

The AC timing spec of the signals rising at this moment may not always be met. Therefore, a timing violation may occur and cause the state machines to enter into an illegal state (stuck).

7.3.2 Workaround

After setting GCR[PPL] = 0b0, read back the GCR[PPL] to ensure the operation was completed. Immediately turn off the clocks to the PCI controller (SCCR[PCICM]) to ensure that the state machines of the PCI controller do not hang due to slow rise time of the PCI signals and metastability. Wait until the PCI signals have returned to defined logic levels. Then, turn on the PCI clocks to continue operation. The wait time must be determined by system characterization.

Configure PCI frame pin to GPIO17 functionality. Make sure that GPIO17 is an input. Clear PCI_GCR[PPL]. Wait until the PCI signals have returned to a steady state. The wait time must be determined by system characterization. Configure PCI_FRAME pin back to PCI_FRAME functionality.

Do not use the push-pin-low feature if it is not critical for the system.

8 PMC

8.1 PMC deasserts wakeup interrupt automatically after e300 core is awake

8.1.1 Description

To exit core PLL change mode and PRE-DIV copy mode, the PMC generates a wakeup interrupt to the e300 core. The e300 core is awaked and signals this to the PMC module. The PMC deasserts the wakeup interrupt while the e300 core is jumping into the corresponding exception vector. Thus, the interrupt handler will not see the interrupt in the IPIC SIVCR or SIPNR_L registers that indicates the PMC was the initiator of the interrupt.

A similar sequence happens when the e300 core is awakened by CSB bus activity.

8.1.2 Workaround

- Software should ignore this interrupt and should continue to execute the application.
- Turn off the wakeup interrupt generation caused by CSB activities by clearing the PMCMR[PMCIE] bit.

9 RTC

9.1 The Tamper (TAMP) bit in the SRTC is not reliable

9.1.1 Description

The TAMP bit in the SRTC does not reliably provide an indication that the RTC power supply or the RTC oscillator has been interrupted.

9.1.2 Workaround

In your system’s design, do not rely on the tamper indication to be reliable.

9.2 RTC Oscillator may not start properly

9.2.1 Description

The Real-Time Clock (RTC) oscillator may not start properly when the V_{BAT_RTC} power domain is active and V_{DD_CORE} is transitioning from power-off to power-on.

When the V_{BAT_RTC} power domain is active and the V_{DD_CORE} transitions from off to on, the RTC time base becomes unreliable. The functionality of the oscillator and clock feeding the RTC counter is in one of three states:

- Normal operation
- Oscillator circuit disabled
  - — The crystal-based circuit is not operational, preventing a clock to the RTC time base so the RTC cannot continue counting.
- The clock signal into the RTC is running at 32 kHz instead of 1 Hz.

If the RTC clock signal is not operational, it affects the following chip features:

- The RTC will not keep accurate time.
- The chip cannot enter or exit Hibernate mode. All Hibernate exit conditions are affected:

- — CAN activity
- — External GPIOs: GPIO28, GPIO29, GPIO30, and GPIO31
- — RTC alarm
- The chip cannot enter Deep Sleep mode.

9.2.2 Workaround

Use combinations of the following measures to obtain different levels of normal operation:

1. Ensure proper RTC operation.

   Ensure that the V_{BAT_RTC} power domain does not lead the V_{DD_CORE} by more than 3 ms, or toggle V_{BAT_RTC} after V_{DD_CORE} has been turned on. The 3 ms is measured from V_{BAT_RTC} at 50% to V_{DD_CORE} at 90% of full values.

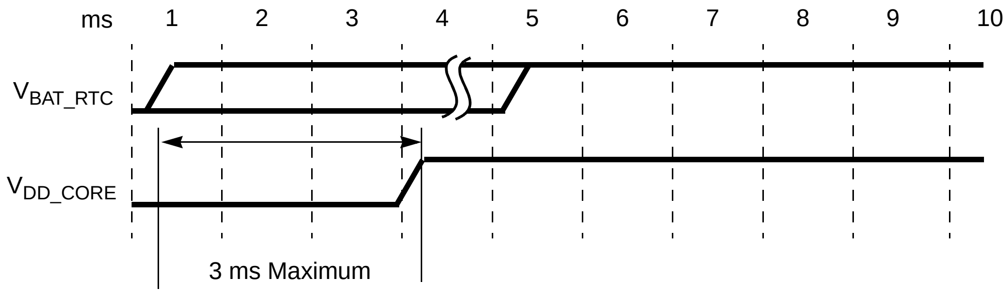

Figure 1. V_{BAT_RTC} to V_{DD_CORE} Timing

   V_{BAT_RTC} can be powered concurrently with or enabled by V_{DD_CORE}. Cycling V_{BAT_RTC} power will result in the RTC counter being reset. See item 4 below. Following this workaround measure eliminates the need for measures 2 and 3 below.

2. Ensure the RTC time base is running.

   To ensure that the RTC input clock is operational, the RTC_XTALI input must be driven by an oscillator signal rather than using a crystal. For oscillator signal specifications, please see the *MPC5121e Datasheet (MPC5121e)* specification number D3.20. A running RTC time base affects the ability to enter and exit Hibernation, and the ability to enter Deep Sleep power modes.

3. Ensure correct RTC time base.

   If V_{BAT_RTC} is kept active when V_{DD_CORE} is inactive, on each startup (e.g.,cold boot or resume from Hibernate), software must ensure that bits in the extended clock register (MBAR + 0x082C) are correct. To ensure the value is correct, software must write the following values, in sequence, into the extended clock register at address MBAR + 0x82C:

   a) 0x8000

   b) 0xC000

   c) 0x8000

   d) 0x0000

This sequence sets the extended register into normal operational mode, which will operate at the proper frequency (32 kHz). This accelerated time base will affect proper wakeup times from Hibernation and Deep Sleep power modes.

NOTE

> The Extended Register at MBAR + 0x82C is currently not documented in the *MPC5121e Reference Manual (MPC5121eRM)*.

4. Restore an accurate real time.

   If the real time is lost due to the RTC counter being reset which is required by the application, the system needs to use an alternate real time clock reference such as an RTC clock chip, or retrieve the time from an external source such as the internet.

10 SATA

10.1 SATA module is non-functional

10.1.1 Description

The SATA module does not function correctly.

10.1.2 Workaround

Do not use the on-chip SATA control in the design.

11 TEMPSENSE

11.1 Temperature sensor does not provide specified accuracy

11.1.1 Description

The temperature measurement accuracy is ±10 °C compared to the specification target of ±5 °C.

11.1.2 Workaround

When utilizing the temperature sensor, be sure to account for the accuracy in the system.

12 USB

12.1 Read of PERIODICLISTBASE after successive writes may return wrong data

12.1.1 Description

The USB controller allows a preset device address in DEVICEADDR register before the device is enumerated, using a shadow register, to assist slow processors. The problem is that this mechanism, which is supposed to be functional only in device mode, is not blocked in host mode. The DEVICEADDR register serves as PERIODICLISTBASE in host mode.

If PERIODICLISTBASE was set to some value, and later it is modified by software in such a way that bit 24 is set to a 1, then an incorrect (previous) value is read back.

12.1.2 Workaround

Write 0x0000 to the PERIODICLISTBASE two times in succession before setting PERIODICLISTBASE to the desired value. This clears the shadow register.

12.2 Priming ISO over SOF creates a bad packet with correct CRC

12.2.1 Description

When an IN token arrives during the prime process (before the endpoint is fully primed), the device sends an empty packet. Also, the data already in the FIFO is incorrectly flushed.

The next time an IN token is received, the endpoint is fully primed. However, previously flushed data is missing. Therefore, the packet is shorter than the size indicated and is flagged as an error.

12.2.2 Workaround

Moving the ISO priming away from the SOF boundary eliminates or reduces the problem. To do this, the priming of the ISO endpoint should be started after the IN token for the endpoint has been sent by the host (SOF interrupt). The time to start the prime depends on what other endpoints are serviced and the latency of the system.

12.3 Port Change interrupt not generated

12.3.1 Description

A Port Change interrupt may fail to be generated when the device controller (on self-powered devices) receives an IN token from the host and immediately after that, a disconnect event through an ULPI RXCMD occurs.

12.3.2 Workaround

An interrupt will be generated. Software needs to check the status of the OTGSC register to understand the cause of the interrupt (VBUS related flags will change). In the worst case, this interrupt is going to appear around 2 ms after the end of the specific session.

12.4 The USB controller can issue transactions that lock up the AHB bus under certain conditions

12.4.1 Description

The AHB bus can lock up if there are other pending transactions while the USB controller issues a specific bus transaction. These specific transactions can happen when the packet size modulo BURSTSIZE setting is 5, 6, or 7 bytes.

12.4.2 Workaround

In the system software set the USB SBUSCFG register to INCR8 (0x2) to avoid any NONSEQ-INC transfers.

12.5 In Host mode, an RX FIFO overflow can generate a false babble error

12.5.1 Description

When the USB interface is configured as a host (or A-device in OTG), an RX FIFO overflow can falsely generate a frame babble error. The frame babble error causes the offending port to be disabled.

12.5.2 Workaround

In such case, to enable the port again the software stack must reset the bus and re-enumerate the connected device.

12.6 No interrupt generated on error condition in ISO Mult 3 mode

12.6.1 Description

When an ISO Fulfillment Error occurs and Mult = 3, the status is not updated to indicate such error and an interrupt is not fired. If the USB interface is functioning in Device ISO IN mode, it does not force the retirement of the ISO-dTD, so it transmits such data in the next uFrame. All the transactions following this situation will be shifted by one in the Mult order until all dTDs are consumed.

12.6.2 Workaround

Use Mult = 2.

12.7 Device ISO IN transactions underrun on ISO Mult 3 transfers

12.7.1 Description

Device ISO IN transactions might underrun on high-speed high-bandwidth endpoints that transfer more than 2048 bytes per period (Mult = 3).

12.7.2 Workaround

Limit the transfer to 2048 bytes (Mult = 2).

12.8 Port change detect bit is set when a resume is forced by SW write to FPR bit in PORTSC register

12.8.1 Description

Working as host, when doing resume (software is driven by writing to bit 6 of PORTSC*x* register), a port change interrupt was generated at the end of resume. According to the EHCI specification, no interrupt should be generated.

12.8.2 Workaround

The extra interrupt should be taken into account when doing resume.

12.9 Incorrect value read from TXPBURST/RXPBURST

12.9.1 Description

The value of the USB register BURSTSIZE (base + 0x160) is incorrectly read when the field AHBRST of USB register SBUSCFG (base + 0x090) is set to any value between 0b000 and 0b100.

12.9.2 Workaround

Create a software register that shadows the value of BURSTSIZE. This shadow register should be used for reading back values.

12.10 EHCI NYET Response to a Complete Split Non-compliance

12.10.1 Description

The NYET response to CSPLIT Bulk is not EHCI-compliant as far as the NackCnt is concerned:

- The NackCnt field is not being decremented when NYET is received.

The NYET response to CSPLIT Interrupt is not EHCI-compliant as far as the Cerr is concerned.

- The Cerr field is being adjusted to the maximum value '3' up on receipt of a NYET on every CSPLIT transaction.

The behavior of the Reclamation bit is not totally compliant with the EHCI specification.

- The reclamation bit was being set to 0 every time the host fetched the queue head with H bit set, giving rise to empty list detection, and the async. list was not empty yet. This situation was leading the host to the async sleep state repeated times because the host was not paying any attention to NackCnt and RL.

12.10.2 Workaround

There is no direct software workaround, but the system software can cancel the transfer that is not reaching to the end after some time, and retry it later.

12.11 CERR counter not reset during a ACK or NAK response to a PING

12.11.1 Description

When an ACK or NAK is sent from the device in response to a PING, the CERR counter value is not being reset to the initial value.

12.11.2 Workaround

A software workaround for this issue is possible. If a value of 0 is used in field CERR of the dTD when building the data structures, the Controller will retry the transaction continuously, and will not be retired due to consecutive errors. This change actually increases the chance for the transaction to complete. Using CERR = 0 is compliant with the EHCI spec.

12.12 RX Endpoint Flush when receiving a token OUT

12.12.1 Description

When receiving a Token OUT, if a RX flush command is issued at the same time, to the same endpoint, the packet will be lost (it will report ACK but the data won't be copy to memory). Additionally, if the controller is in the stream disable mode (SDIS bit set a '1') the endpoint will also be blocked and will NAK forever any token sent to that endpoint.

12.12.2 Workaround

Do not use the RX flush feature to clear/stop an endpoint if acting as a peripheral.

12.13 OTG Controller as Host does not support Data-line Pulsing SRP

12.13.1 Description

When the OTG core is acting as a Host and VBUS is turned off, and the attached Device attempts to perform a Session Request Protocol by using Data-line Pulsing, it will not be recognized by the Host. Also, when doing HNP and becoming a Host, a SE0 is forced in the line causing the OPT TD5.4 test to fail, without software workaround.

12.13.2 Workaround

A software workaround is possible for the HNP situation only. With this workaround, it is possible to pass OPT TD5.4 test. The software must assert core mode to the device (USBMODE.CM) and set the Run/Stop bit (USBCMD.RS) to 1 just after the controller ends reset. Wait until USBCMD.RST is 0 after setting it to 1.

12.14 Device does not respond to INs after receiving corrupted handshake from previous IN transaction

12.14.1 Description

The device does not respond to IN tokens after receiving a corrupted handshake if the next tokens arrive sooner than the BTO constant as defined on vusb_hs_pkg (BTO_TIME_HS_60MHZ / 30MHZ).

12.14.2 Workaround

No workaround is available.

12.15 Last read of the current dTD done after USB interrupt

12.15.1 Description

After executing a dTD, the device controller executes a final read of the dTD terminate bit to see if another dTD has been added to the linked list by software right at the last moment.

It was found that the last read of the current dTD is being done after the interrupt is issued and not before. There could be a race between this final dTD read and the interrupt handling routine servicing the interrupt on complete that may result on the software freeing the data structure memory location, prior to the last dTD read being performed.

12.15.2 Workaround

No workaround is available.

Table 2. Revision History Table

Rev. Number	Substantive Changes	Date of Release
1.0	• Initial release.	03/2009
2.0	• Added item 509, RTC.	08/2010
3.0	• Added items 477, 478. 479, 480, 481, 497, 498, 499, 501, 503, 504, USB (all). • Added item 510, NFC. • Removed item 469, PSC. • Added item ERR003383, e300 Core.	12/2011

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
www.freescale.com/support

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064
Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

Freescale Semiconductor Literature Distribution Center
1-800-441-2447 or +1-303-675-2140
Fax: +1-303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

RoHS-compliant and/or Pb-free versions of Freescale products have the functionality and electrical characteristics as their non-RoHS-compliant and/or non-Pb-free counterparts. For further information, see http://www.freescale.com or contact your Freescale sales representative.

For information on Freescale’s Environmental Products program, go to http://www.freescale.com/epp.

Document Number: MSE5121E_0M36P
Rev. 3
12/2011