ES_LPC3220

Errata sheet LPC3220

Rev. 9 — 1 June 2011 **Errata sheet**

Document information

Info	Content
Keywords	LPC3220 errata
Abstract	This errata sheet describes both the known functional problems and any deviations from the electrical specifications known at the release date of this document. Each deviation is assigned a number and its history is tracked in a table.

Revision history

Rev	Date	Description
9	20110601	• Added USB.1.
8	20110201	• Added HSUART.1.
7	20100624	• Added DMA.1. • Added NOR.1.
6	20100318	• Added clarification regarding /01 Revision 'A' parts.
5	20100205	• The format of this errata sheet has been redesigned to comply with the new identity guidelines of NXP Semiconductors. • Added DDR write set-up time.

Contact information

For more information, please visit: **http://www.nxp.com**

For sales office addresses, please send an email to: **salesaddresses@nxp.com**

1. Product identification

The LPC3220 devices typically have the following top-side marking:

LPC3220FET296

xxxxxxx

xxYYWWR

The last letter in the last line (field 'R') will identify the device revision. This Errata Sheet covers the following revisions of the LPC3220:

Table 1. Device revision table

Revision identifier (R)	Revision description
'A'[1]	Second device revision
'-'[2]	Initial device revision

[1] Revision 'A' parts with and without the /01 suffix are identical. For example, LPC3220FET296 Revision 'A' is identical to LPC3220FET296/01 Revision 'A'.

[2] Does not apply to /01 parts.

Field 'YY' states the year the device was manufactured. Field 'WW' states the week the device was manufactured during that year.

2. Errata overview

Table 2. Functional problems table

Functional problems	Short description	Revision identifier	Detailed description
DMA.1	Single burst DMA memory-to-memory transfers have additional memory cycles when the DMA source memory is on the EMC bus.	'-', 'A'	Section 3.1 on page 4
NOR.1	When booting from NOR flash, SDRAM devices will not release the data bus, preventing the LPC3220 from booting correctly	'-', 'A'	Section 3.2 on page 6
DDR.2	DDR EMC_D[15:0] to EMC_DQS[1:0] data output set-up time, $t_{su(Q)}$, for MCU write to DDR provides limited timing margin	'-', 'A'	Section 3.3 on page 8
DDR.1	DDR interface has > 1.2 ns clock skew	'-', 'A'	Section 3.4 on page 10
RTC.1	An RTC match doesn't drive the ONSW pin active (HIGH).	'-', 'A'	Section 3.5 on page 11
INT.1	GPI_08 does not generate in interrupt signal.	'-'	Section 3.6 on page 11
MCPWM.1	Input pins (MCI0-2) on the Motor Control PWM peripheral are not functional	'-', 'A'	Section 3.7 on page 12
HSUART.1	High speed UART receive FIFO and status can freeze	'-', 'A'	Section 3.8 on page 13
USB.1	USB host controller hangs on a dribble bit	'-', 'A'	Section 3.9 on page 14

Table 3. AC/DC deviations table

AC/DC deviations	Short description	Revision identifier	Detailed description
ESD.1	Weak ESD protection on Reset_N pin (pin M14)	'-'	Section 4.1 on page 14

3. Functional problems detail

3.1 DMA.1: Single burst DMA memory-to-memory transfers have additional memory cycles when the DMA source memory is on the EMC bus

Introduction:

The DMA controller is an AHB master that can transfer blocks of data between peripheral-to-memory, memory-to-peripheral, peripheral-to-peripheral, and memory-to-memory. In addition to transferring data between memories, a DMA memory-to-memory flow can be used to transfer blocks of data to / from an FPGA or external peripheral chip connected to an EMC static memory chip select. When a memory, FPGA or external peripheral chip does not support burst transfers (i.e. multiple reads for each active chip select or read strobe) the burst size for that memory-to-memory flow must be set for one transfer per burst.

Problem:

When using memory-to-memory DMA with the EMC static chip select (EMC_CS[x]_N) as the DMA source and the DMA channel source burst size is set for a single transfer (DMACCxControl:SBSIZE = 0), each DMA source read should be a single bus-wide access. The access should be similar to reading the EMC_CS[x]_N static memory with an ARM LDR instruction, as shown in Figure 1. Note the EMC signal timing for the read is controlled by the EMCSTATICx registers. In all example scope shots the EMCSTATICWAITx registers are set to the maximum value.

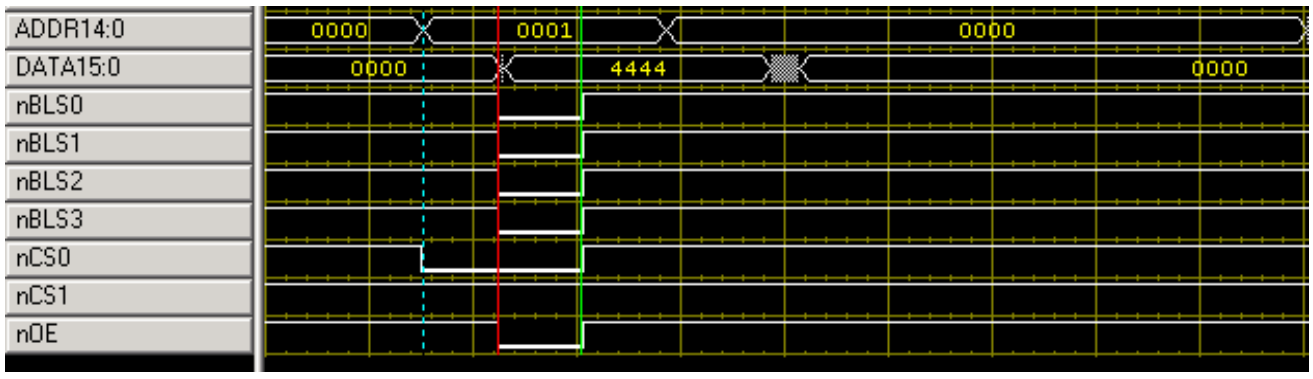

Fig 1. Scope shot 1 - expected read timing

However, the actual EMC timing for the source DMA read is a double wide chip select with a burst of two reads (notice how the address increments near the halfway point of nCS0 active), see Figure 2. The second data read during the burst is discarded, as the DMA destination write (also to nCS0 in Figure 2) following each read, always writes the first value read during the read burst. When the DMA source address is set to auto-increment, the last DMA read transfer will address the last address of the source buffer and the last source buffer address +1. This behavior only happens during the read part of the DMA transfer. Memory-to-memory DMA destination writes to the EMC static chip select work as expected.

DMA register values used in Figure 2:

DMACConfig = 0x01

DMACCxSrcAddr = 0xe0000000; EMC_CS0

DMACCxDestAddr = 0xe0000040; EMC_CS0

DMACCxLLI = 0x0

DMACCxControl = 0x0c480004; Dest & Src addrs increment, Dest & Src 32-bit; Dburst & Sburst size 1; transfer size 4

DMACCxConfig = 0x01

EMCStaticConfig0 = 0x00000082; 32-bit width, Byte lane state 1

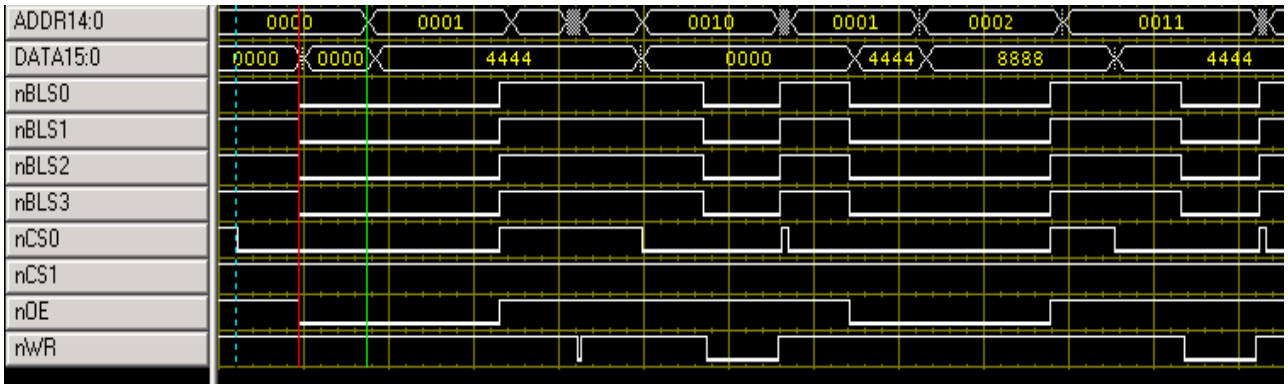

Fig 2. Scope shot 2 - actual read timing

Results of this behavior:

1. DMA reads from an external memory will have lower performance than a software read loop. The source read burst of two, to get one transfer, will significantly increase the time to complete all transfers in the memory-to-memory DMA, therefore decreasing the overall throughput possible on the EMC_CSx_N static memory interface.
2. Potential unintended consequence when the last DMA read accesses the address beyond the DMA source buffer address in the FPGA or external peripheral chip. This extra address is the second access during the last DMA source read.

Work-around:

When interfacing an external peripheral device that does not support burst mode access through the EMC Static Memory interface the following work-arounds are recommended:

1. Avoid using DMA to transfer read blocks of data from the external device. Instead use a software loop with LDR instruction to read blocks of data from the external device.
2. If DMA can't be avoided, ensure there is at least one unused address between the highest address used for the external device DMA data buffer and any status or control register in the device that will initiate any unwanted action just by reading from the register (i.e. clear an interrupt or status).

3.2 NOR.1: When booting from NOR flash, SDRAM devices will not release the data bus, preventing the LPC3220 from booting correctly

Introduction:

In systems that use SDRAM and boot from NOR FLASH, an issue can occur on system reset that will prevent the SDRAM devices from releasing the data bus. This will prevent normal operation of NOR FLASH due to data bus contention and prevent the LPC3220 from booting correctly. This applies to systems using either Single Data Rate (SDR) or Double Data Rate (DDR) SDRAM devices.

Problem:

If the LPC3220 is reset during an SDRAM access, the SDRAM clock and clock enable will be immediately de-asserted. If the de-assertion occurs during the period of time the SDRAM is driving the data bus, the SDRAM will hold that state until the next clock occurs at the SDRAM clock input when the clock enable is active. However, the LPC3220 won't deliver the clock and clock enables until software actually sets up the EMC state to do this, so the SDRAM will remain in the data assertion state on the data bus while the LPC3220 tries to boot.

When the chip attempts to load boot code from NOR FLASH after reset, the correct signals are asserted to the NOR FLASH device and the NOR FLASH device places its data on the data bus. But if the SDRAM is still driving the bus, the NOR FLASH device and SDRAM device are in contention and the data will not be read correctly into the LPC3220. In this situation, the LPC3220 will fail to boot.

Work-around:

Since this issue only occurs with NOR FLASH, using one of the other boot methods such as NAND or SPI FLASH boot is a good workaround for the issue.

If booting from NOR FLASH is a requirement, the simple circuit shown in Figure 3 can be used to clear the SDRAM state at system reset. This will not change the normal functioning of the LPC3220 EMC or SDRAM operations. If SDRAM devices are also present on the 2nd SDRAM chip select, a similar circuit will be needed for those devices using EMC_CKE1.

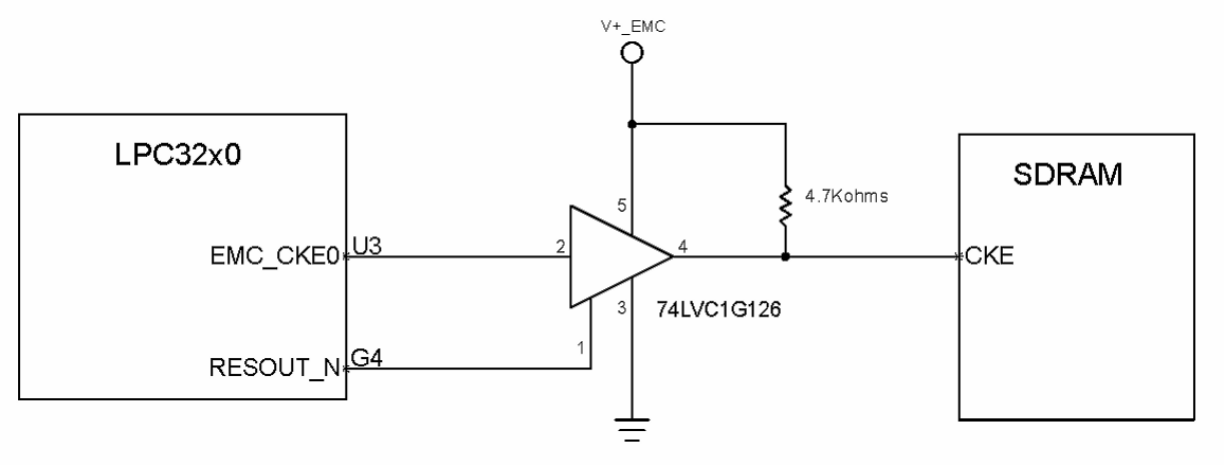

Fig 3. 128 MB DDR SDRAM example

3.3 DDR.2: DDR EMC_D[15:0] to EMC_DQS[1:0] data output set-up time, $t_{su(Q)}$, for MCU write to DDR provides limited timing margin

Remark: This affects both 1.8 V mobile and 2.5 V DDR SDRAM system implementations.

Introduction:

DDR memory interface signal EMC_DQS[1:0] is source synchronous, defined to be driven by the MCU center aligned to the data EMC_D[15:0] for writes, while driven by the DDR memory edge aligned to the EMC_D[15:0] for reads. The basic DDR write timing is shown in the data sheet Fig 1.

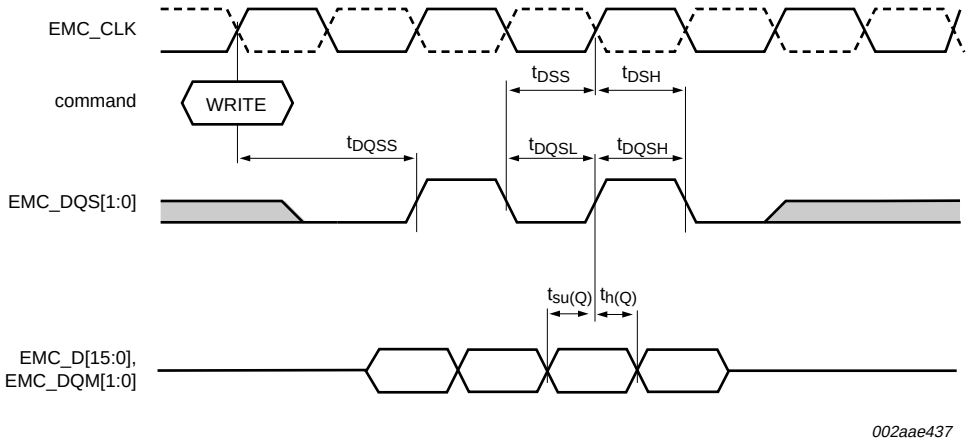

Fig 4. Basic DDR write timing

Problem:

For DDR writes the LPC3220 drives the EMC_DQS[1:0] earlier in the data valid window than center aligned. With the EMC_CLK at 133 MHz this produces a minimum set-up time between the EMC_D[15:0] and EMC_DQS[1:0] of 600 ps across silicon process, voltage and temperature. Test conditions are with the EMC buffers set to fast slew rate driving 2 inches of 50 Ω transmission line and 10 pF load capacitance. DDR memories specify EMC_D[15:0] to EMC_DQS[1:0] set-up time minimum as 400 ps. This leaves 200 ps set-up time margin due to customer specific load and PCB layout implementation. See the LPC3220_30_40_50 data sheet for the complete range of DDR data output set-up time, $t_{su(Q)}$, and data output hold time $t_{h(Q)}$ times.

Work-around:

To get the most DDR set-up time margin, the following is recommended:

1. The DDR initialization software should set the SDRAMCLK_CTRL register (0x4000 4068) SDRAM_PIN_SPEED[3:1] bits = 0 (fast slew rate). This is for both 1.8 V mobile and 2.5 V DDR memories.
2. Systems requiring 128 MB or less of DDR should be implemented using a single EMC_DYCSx_N for DDR. The single chip select system may be constructed with a single 16-bit wide DDR or two 8-bit wide DDR SDRAMs using up to the maximum supported 512 Mbit DDR density. Using two 8-bit wide DDRs will have less capacitive loading and facilitate simple point-to-point routing of EMC_D[15:0] and EMC_DQS[1:0] signals over using two 16-bit DDRs and two EMC_DYCSx banks.

3. Series termination resistors are not needed for the LPC3220 EMC outputs. If series termination resistors are used they should be placed as close to the DDR EMC_D[15:0] and EMC_DQS[1:0] pins as possible.
4. If the data bus EMC_D[15:0] is shared with additional devices (i.e., NOR flash, buffers, etc.) the board should be routed with a daisy chain topology, where the LPC3220 is placed at one extreme of the data bus and the DDR(s) at the other extreme. Other device(s) should be placed between the LPC3220 and DDR memory (closer to the DDR).
5. The PCB trace length of EMC_DQS[1:0] should be at least 2 inches (but not more than 4 inches) longer than EMC_DQ[15:0] and EMC_DQM[1:0]. On a typical FR4 PCB this adds at least 334 ps to set-up time margin for DDR writes. For reads from DDR the increased trace length of EMC_DQS[1:0] will be automatically compensated for by the software initialization function find_ddr_dqsin_delay() which sets the optimal value DDR_DQSIN_DELAY(SDRAMCLK_CTRL[6:2]). The function find_ddr_dqsin_delay() can be found in the "DDR SDRAM setup code for the LPC32x0 series" on the NXP web site.

Example 128 MB system DDR SDRAM using a single EMC_DYCSx_N:

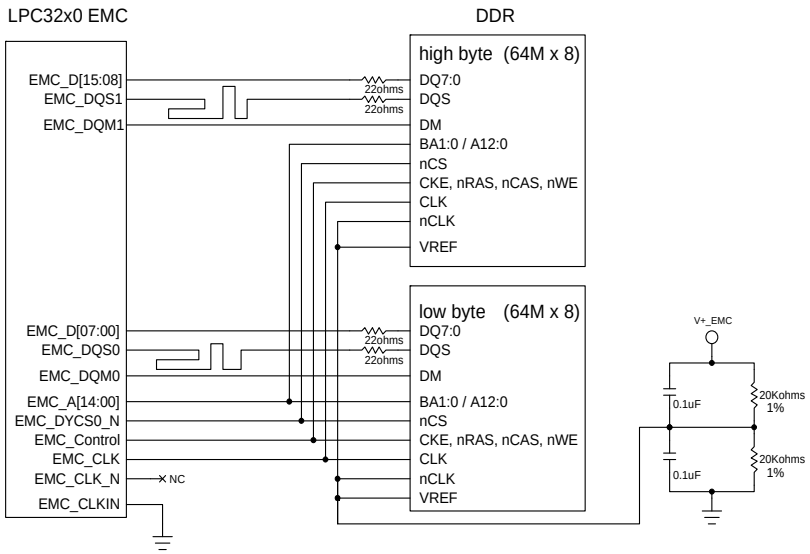

Fig 5. 128 MB DDR SDRAM example

3.4 DDR.1: DDR interface has >1.2 ns clock skew

Introduction:

DDR memory uses a differential clock which is generated by the LPC3220. The differential clock consists of two clock signals: EMC_CLK is the positive clock and DDR_nCLK is the negative clock.

Problem:

There is approximately 1.27 ns of skew between the low transition of the DDR_nCLK and the high transition of the EMC_CLK. This can cause two problems: 1) Some DDR devices use this clock transition to drive a digital lock loop (DLL) in the DDR device. The DDR clock skew can cause the DDR device's internal DLL to loose lock, resulting in the wrong data being latched. 2) The DDR clock skew can also cause a reduced Data Valid Window (also called Data-Out Window) from a DDR device. However, the LPC3220 has a programmable DQS delay to achieve center alignment for accurate data reads.

Work-around:

Connecting the DDR device negative clock input (DDR_nCLK from the LPC3220) to the DDR Reference Voltage (Vref - the midpoint of the DDR signal voltage swing, which is generally VDDQ/2) avoids the clock skew problem, though it also eliminates the advantages of differential signaling. The LPC3220 DDR_nCLK output should be left unconnected. DDR Reference Voltage can be generated with a divide-by-two voltage divider. Standard DDR memories usually require a Vref input, so this DDR reference voltage should already be available. Mobile DDR devices typically do not have a Vref input, so the external voltage divider may need to be added to the design for this work-around.

It is also possible to compensate for the 1.27 ns clock skew by adding an additional 7 inches of pcb trace length to the EMC_CLK signal. However, this could have unintentional consequences; such as increased Electro-Magnetic Interference.

3.5 RTC.1: An RTC match doesn't drive the ONSW pin active (HIGH)

Introduction:

An ONSW output pin (M15) is included in the LPC3220 to assist in waking up the chip after power is removed from all functions except the RTC and Battery RAM. When there is an active match condition the RTC will drive the ONSW pin HIGH. The RTC only drives the ONSW pin while the match is active, and after 1 second of active match, if the software has not accessed the RTC block, the ONSW pin will go low when the match is no longer active.

Problem:

When power is removed from all functions except the RTC and Battery RAM, the RTC does NOT drive the ONSW pin HIGH when there is an active match condition.

Work-around:

There is no work-around for this problem.

3.6 INT.1: GPI_08 does not generate an interrupt signal

Introduction:

The LPC3220 contains 12 pins (GPI_00 - GPI_09, GPI_19, GPI_28) that function as dedicated General Purpose Inputs. Each of these pins can generate an individual interrupt for the input pin. Sub Interrupt Controller Register 1 (SIC1_ER) and Sub Interrupt Controller Register 2 (SIC2_ER) contains bits that allow enabling or disabling the interrupt for the associated pin.

Problem:

When bit nine is set to one in the Sub Interrupt Controller 2 Enable register (SIC2_ER[9]) it does not enable the interrupt for the GPI_08 pin. All other General Purpose Input pins (GPI_00 - GPI_07, GPI_09, GPI_19, GPI_28) interrupts work correctly.

Work-around:

There is no work-around for this problem.

3.7 MCPWM.1: Input pins (MCI0-2) on the Motor Control PWM peripheral are not functional

Introduction:

On the LPC3220, the Motor Control PWM (MCPWM) peripheral is optimized for three-phase AC and DC motor control applications and can also be used in applications which require timing, counting, capture, and comparison. The MCPWM contains three input pins (MCI0-2) for PWM channels 0, 1, and 2. The inputs can be used as feedbacks for controlling brushless DC motors with Hall sensors, and also can be used to trigger a Timer/Counter's (TC) capture or increment a channel's TC when MCPWM is configured as a timer/counter.

Note: MCI0-2 pins are also called MCFB0-2 (refer to LPC32x0 User manual for more details).

Problem:

The input pins (MCI0-2) are not functional.

Work-around:

The GPIO interrupts[1] need to be used instead of the MCPWM MCI0-2 pins. On the LPC3220, the GPIO interrupts can only be set to either trigger on the rising edge or on the falling edge. Therefore, in order to detect all six states of the connected hall sensor through an interrupt, the state of the pin needs to be determined and switch to rising or falling edge interrupt accordingly.

1. Available GPIO interrupt pins: GPIO_00 to GPIO_05, GPI_00 to GPI_09, GPI_19, GPI_28, and all port 0 and port 1 pins.

3.8 HSUART.1: High speed UART receive FIFO and status can freeze

Introduction:

The three high speed UART's (HSUART) receive (RX) FIFOs can sometimes enter a state where they no longer accept received data. When this state occurs, the HSUART's RX FIFO will no longer accept data regardless of RX FIFO fill status. The receive state of the HSUART may indicate a number of possible, but invalid, receive statuses. These invalid statuses may include RX FIFO or timeout interrupts pending with no receive data in the RX FIFO, invalid RX FIFO status, stuck RX interrupts, or other possible RX statuses. Once the HSUART enters this state, the state can only be cleared by a chip reset. This applies to the U1_RX, U2_RX, and U7_RX pins. The HSUART's transmit side is not affected by this issue and will work as normal when the HSUART receive side stops. The four standard UARTs do not exhibit this behavior.

Problem:

It has been determined that this failed receive state can be entered by receiving a burst of high frequency noise into the HSUART RX pin. High frequency noise consists of pulsed or random toggling of the HSUART RX line at about 2.5 MHz or greater. The chance of the HSUART entering the state increases with the number of pulses and frequency of the pulses received. Generally, a single pulse won't cause the state to occur.

During normal data transfer with transfer rates 2400 bps (416 uS) to 921.6 Kbps (1.085 uS), this state won't occur. However, conditions outside the transfer itself may cause the state to occur. It has been observed in some systems that insertion of the serial cable into the board's serial connector can cause connection noise or oscillations on the transceiver. This noise is driven onto the HSUART RX pin from the transceiver as a series of random pulses.

Work-around:

If all 7 UARTs aren't needed or 921.6 Kbps transfer rate isn't needed, use the standard UARTs instead of the high speed UARTs to avoid the issue altogether. For systems that require the HSUARTs, care must be taken to limit the exposure of the HSUART RX signal for the type of signal conditions that can cause the state to occur. There are several possible solutions that can help reduce the state from occurring.

Whenever the HSUART is not in use, place the HSUART into loopback mode. When in loopback, the RX pin is connected internally to the HSUARTs TX pin and is isolated from the external RX input. While in this loopback state, the condition won't occur on the HSUART regardless of the signal on the RX input. The HSUART TX pin will remain in the idle state in loopback mode when no data is being sent from the HSUART.

Optionally, if the HSUART is connected to a transceiver that supports enabling and disabling of the input signal from the transceiver RX input to the transceiver RX output to the HSUART RX input, disable it when not expecting a transmission. Regardless of how the transceiver is connected to the RX pin, the RX pin should be prevented from floating at power-up, reset, or when the transceiver is disabled. This can be done by adding a pull-up resistor to the HSUART RX pin.

If using a system where the HSUART always need to be enabled, consider adding the capability to sense when the cable has been plugged into the connector and switch the HSUART out of loopback mode only once the cable has been installed to prevent cable insertion noise.

3.9 USB.1: USB host controller hangs on a dribble bit

Introduction:

Full-/low-speed signaling uses bit stuffing throughout the packet without exception. If the receiver sees seven consecutive ones anywhere in the packet, then a bit stuffing error has occurred and the packet should be ignored.

The time interval just before an EOP is a special case. The last data bit before the EOP can become stretched by hub switching skews. This is known as dribble and can lead to a situation where dribble introduces a sixth bit that does not require a bit stuff. Therefore, the receiver must accept a packet for which there are up to six full bit times at the port with no transitions prior to the EOP.

Problem:

The USB host controller will hang indefinitely if it sees a dribble bit on the USB bus. It will hang the first time a dribble bit is seen. Once it is in this state there is no recovery other than a hard chip reset. This problem has no effect on the USB device controller.

Work-around:

None.

4. AC/DC deviations detail

4.1 ESD.1: Weak ESD protection on Reset_N pad

Introduction:

The LPC3220 was designed to withstand electrostatic discharges up to 2000 V using the Human Body Model.

Problem:

The RESET_N pad (pin M14) does not pass ESD tests above 1000 V.

Work-around:

Observe proper ESD handling precautions for the RESET_N pin

5. Legal information

5.1 Definitions

Draft — The document is a draft version only. The content is still under internal review and subject to formal approval, which may result in modifications or additions. NXP Semiconductors does not give any representations or warranties as to the accuracy or completeness of information included herein and shall have no liability for the consequences of use of such information.

5.2 Disclaimers

Limited warranty and liability — Information in this document is believed to be accurate and reliable. However, NXP Semiconductors does not give any representations or warranties, expressed or implied, as to the accuracy or completeness of such information and shall have no liability for the consequences of use of such information.

In no event shall NXP Semiconductors be liable for any indirect, incidental, punitive, special or consequential damages (including - without limitation - lost profits, lost savings, business interruption, costs related to the removal or replacement of any products or rework charges) whether or not such damages are based on tort (including negligence), warranty, breach of contract or any other legal theory.

Notwithstanding any damages that customer might incur for any reason whatsoever, NXP Semiconductors' aggregate and cumulative liability towards customer for the products described herein shall be limited in accordance with the *Terms and conditions of commercial sale* of NXP Semiconductors.

Right to make changes — NXP Semiconductors reserves the right to make changes to information published in this document, including without limitation specifications and product descriptions, at any time and without notice. This document supersedes and replaces all information supplied prior to the publication hereof.

Suitability for use — NXP Semiconductors products are not designed, authorized or warranted to be suitable for use in life support, life-critical or safety-critical systems or equipment, nor in applications where failure or malfunction of an NXP Semiconductors product can reasonably be expected to result in personal injury, death or severe property or environmental damage. NXP Semiconductors accepts no liability for inclusion and/or use of NXP Semiconductors products in such equipment or applications and therefore such inclusion and/or use is at the customer's own risk.

Applications — Applications that are described herein for any of these products are for illustrative purposes only. NXP Semiconductors makes no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Customers are responsible for the design and operation of their applications and products using NXP Semiconductors products, and NXP Semiconductors accepts no liability for any assistance with applications or customer product design. It is customer's sole responsibility to determine whether the NXP Semiconductors product is suitable and fit for the customer's applications and products planned, as well as for the planned application and use of customer's third party customer(s). Customers should provide appropriate design and operating safeguards to minimize the risks associated with their applications and products.

NXP Semiconductors does not accept any liability related to any default, damage, costs or problem which is based on any weakness or default in the customer's applications or products, or the application or use by customer's third party customer(s). Customer is responsible for doing all necessary testing for the customer's applications and products using NXP Semiconductors products in order to avoid a default of the applications and the products or of the application or use by customer's third party customer(s). NXP does not accept any liability in this respect.

Export control — This document as well as the item(s) described herein may be subject to export control regulations. Export might require a prior authorization from national authorities.

5.3 Trademarks

Notice: All referenced brands, product names, service names and trademarks are the property of their respective owners.

6. Contents

For more information, please visit: http://www.nxp.com
For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 1 June 2011
Document identifier: ES_LPC3220